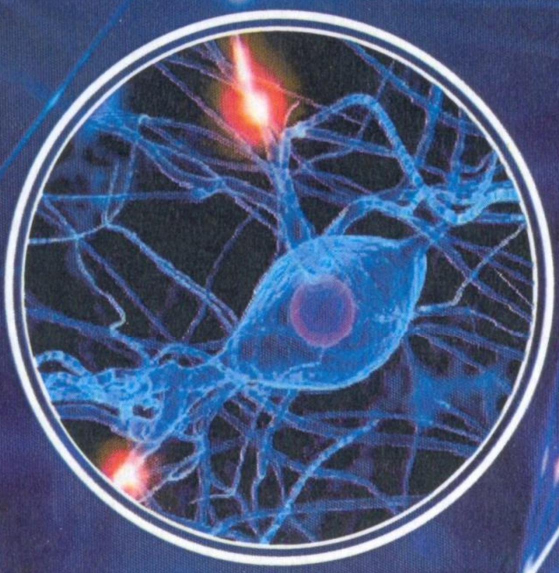
ШАРИПБАЙ А.А.

НЕЙРОННЫЕ СЕТИ

Учебное пособие



УДК 004. (075.8)
ББК 32.943 я 73
Ш 25

Рецензенты:
Адамов А.А. – заведующий кафедрой Математического и компьютерного моделирования, ЕНУ имени Л.Н.Гумилева, д.т.н., профессор;
Казиев Г.З. – директор исследовательского центра АО «Национальные информационные технологии», д.т.н., профессор;
Ускенбаева Р.К. – проректор Международного университета информационных технологий, д.т.н., профессор.

Шарипбай А.А – директор НИИ «Искусственный интеллект», профессор кафедры «Информатики и информационной безопасности» ЕНУ имени Л.Н. Гумилева, доктор технических наук, лауреат Государственной премии Республики Казахстан, академик Международной академии информатизации.

РЕКОМЕНДОВАНО
Евразийским национальным университетом имени Л.Н. Гумилева

Ш 25 НЕЙРОННЫЕ СЕТИ: **учебное пособие/ авт. Шарипбай А.А. –**
Алматы: Эверо, 2017. – 278 с.

ISBN 978-601-310-232-0

В учебном пособии рассматриваются искусственные нейронные сети, которые являются математическими моделями естественных (биологических) нейронных сетей и представляют собой совокупность связанных между собой нейронов, способные обрабатывать входную информацию и формировать выходные воздействия в процессе изменения своего внутреннего состояния во времени.

Содержание учебного пособия составлено так, чтобы применить кредитную технологию обучения и имеет четыре уровня, которые позволяют проводить текущие, промежуточные, рубежные и итоговые контроли и оценки знаний.

Учебным пособием могут пользоваться студенты, магистранты, докторанты и преподаватели специальностей Информатика, Системы информационной безопасности, Информационные системы, Вычислительная техника и программное обеспечение, Автоматизация и управление, Математическое и компьютерное моделирование, а также ученые и специалисты, которые интересуются искусственными нейронными сетями.

УДК 004. (075.8)
ББК 32.943 я 73

ISBN 978-601-310-232-0

© Шарипбай А.А., 2017.
© Эверо, 2017.

ОГЛАВЛЕНИЕ

НИИ Искусственный интеллект

НИИ искусственный интеллект

ПРЕДИСЛОВИЕ

В настоящее время теория нейронных сетей состоялась как наука, обладает собственным аппаратом и предметом исследований. Нейронные сети включают в себя большой спектр вопросов из разных областей науки: информатики, математики, физики, схемотехники и позволяют решать широкий круг различных задач.

Значительный интерес к нейронным сетям был инициирован после появления в 1982 году работы Хопфилда (Hopfield J.J.), который показал, что задача с нейронами может быть сведена к обобщениям ряда моделей, разработанных к тому моменту в физике неупорядоченных систем. Работа сети Хопфилда состоит в релаксации начального "спинового портрета" матрицы двоичных кодов к одному из стационарных состояний, определяемых правилом обучения Хебба.

В 1986 году появилась работа Румельхарта, Хинтона и Вильямса (Rumelhart D.E., Hinton G.E., Williams R.J., 1986), содержавшая ответ на вопрос, долго сдерживавший развитие нейроинформатики – как обучаются иерархические слоистые нейронные сети, для которых учеными "классиками" еще в 40-50-ых годах была установлена универсальность для широкого класса задач. В последующие годы предложенный Хинтоном алгоритм обратного распространения ошибок претерпел много вариаций и модификаций.

Искусственная нейронная сеть применяется как модель, средство и инструмент для аппроксимации многомерных функций и ассоциативной памяти, прогнозирования и диагностики процессов, для поиска по ассоциациям и поиска закономерностей в массивах данных, для адаптивного управления и статистического анализа, для идентификации и распознавания образов и др.

Нейронные сети играют важную роль в построении различных интеллектуальных систем и применяются для решения очень многих задач обработки изображений, управления роботами и

НИИ искусственный интеллект

непрерывными производствами, для распознавания и синтеза речи, для диагностики заболеваний людей и технических неполадок в машинах и приборах, для предсказания курсов валют и результатов скачек, а также для решения других динамических задач.

Предлагаемое учебное пособие написано на основе лекций, читавшихся автором в разные годы для студентов и магистрантов факультета информационных технологий Евразийского национального университета имени Л.Н. Гумилева по специальностям Информатика, Информационные системы, Вычислительная техника и программное обеспечение.

Причиной подготовки данного учебного пособия послужило отсутствие учебных изданий в данной сфере. Поэтому для полноценного обеспечения учебного процесса автору приходилось использовать различные источники и интернет ресурсы, в которых одни и те же понятия имели разные термины, определения и обозначения. Данное обстоятельство натолкнуло на идею написания данного учебного пособия с использованием единых терминов, определений и обозначений для одних и тех же понятий. Автор надеется, что предлагаемое учебное пособие исправит ситуацию с текущей нехваткой материлов и учебных изданий по изучаемому предмету, а также позволит унифицировать термины, определения и обозначения одних и тех же понятий.

Методика изложения учебного материала сводится к тому, чтобы преподнести читателю сведения о фундаментальных фактах в теории нейронных сетей, познакомить его с приемами доказательства утверждений в изучаемой области, а также дать ему определенный запас примеров и упражнений для закрепления изучаемого материала.

Учебный материал организован в соответствии с кредитной технологией, и состоит из трех уровней: модуль, блок, урок. Это позволит обучать и проводить контроль знаний на четырех уровнях: на уровне урока – текущий контроль, на уровне блока – промежуточный контроль, на уровне модуля – рубежный контроль, на уровне учебного пособия – итоговый контроль. Оценка знаний

НИИ интеллект

осуществляется с помощью тестов, составленных в соответствии с вышеперечисленными уровнями.

Учебное пособие, как было упомянуто выше, состоит из трех модулей, а также из списка использованных источников и предметных указателей. Первый модуль посвящен обсуждению математических основ нейронных сетей и состоит из трех блоков, в которых приводится описание элементов теории множеств, элементов теории функций, элементов векторного исчисления. Второй модуль охватывает вопросы теории нейронных сетей и состоит из трех блоков, в которых раскрываются основные понятия и виды нейронных сетей, функции активации и модели нейронов. В третьем модуле рассматривается обучение нейронных сетей, в котором имеется один блок, описывающий различные парадигмы и методы обучения. В списке литературы и интернет ресурсов приводятся источники, которые были использованы при формировании учебного материала учебного пособия. Среди них имеются широко распространенные монографии, учебники, учебные пособия и др. В связи с этим в тексте настоящего учебного пособия в начале каждого параграфа будут указаны ссылки на авторов тех или иных моделей нейронных сетей.

Учебным пособием могут пользоваться студенты, магистранты, докторанты, преподаватели, ученые по специальностей Информатика, Системы информационной безопасности, Вычислительная техника и программное обеспечение, Информационные системы, Автоматизация и управление, Математическое и компьютерное моделирование, а также ученые и специалисты, которые интересуются искусственными нейронными сетями и желают расширить знания в соответствующей области .

Автор:
Алтынбек Амирович ШАРИПБАЙ,
доктор технических наук, профессор,
лауреат Государственной премии Республики Казахстан.

НИИ Искусственный интеллект

I. МАТЕМАТИЧЕСКИЕ ОСНОВЫ НЕЙРОННЫХ СЕТЕЙ

II.1. Элементы теории множеств

I.1.1. Множества и операции над ними

В данном параграфе нами будут рассмотрены понятия множества, определены операции над ними и раскрыты их свойства, а также приведены примеры, предложены задания, заданы вопросы. При формировании учебного материала данного параграфа были использованы следующие источники [14, 17, 22, 27, 28, 33].

Определение I.1.1.1. *Множеством* называется совокупность элементов, различимых по некоторому типу (признаку).

Множества обозначаются прописными (большими) латинскими буквами, а их элементы – строчными (малыми) латинскими буквами или арабскими цифрами. Например, множество натуральных чисел обозначается буквой N, а его элементы цифрами – 1, 2, 3, … .

Множество задается двумя способами: перечислением всех элементов или описанием свойств элементов. При этом его элементы указываются в фигурных скобках { и }.

Определение I.1.1.2. Мощность множества A равна числу его элементов и обозначается через $|A|$.

Примеры I.1.1.1:

1. Все неотрицательные целые числа образуют множество натуральных чисел.

2. $D = \{0, 1\}$, где элементами множества D являются только перечисленные постоянные величины 0, 1.

3. $X = \{x: x>0\}$, где элементами множества X являются только положительные переменные величины x.

4. Если $A = \{a, b, c, d, e\}$, то $|A| = 5$.

Запись $a \in A$ ($a \notin A$) означает, что элемент a принадлежит (не принадлежит) множеству A.

Среди всех множеств выделяют два особых множества:

1. $\varnothing$ – *пустое множество*, не содержащее ни одного элемента.

НИИ Искусственный интеллект

2. U – *универсальное множество* (*универсум*), содержащее все элементы рассматриваемого типа (предметной области).

Относительно теории универсум это множество, содержащее в качестве элементов все объекты, рассматриваемые в этой теории.

Например, универсумом является:

1) в теории чисел – множество всех целых чисел;

2) в теории языков – множество всех слов в заданном алфавите;

3) в геометрии – множество всех точек n-мерного геометрического пространства.

Определения I.1.1.2. Пусть заданы два множества A и B, тогда над ними можно определить следующие операции:

1. *Объединение* состоит из элементов A или B, т.е.,

$$A \cup B = \{x \colon x \in A \vee x \in B\}.$$

2. *Пересечение* состоит из элементов A и B, т.е.,

$$A \cap B = \{x \colon x \in A \ \& \ x \in B\}.$$

3. *Дополнение* состоит из элементов универсума U и не включает элементов A, т.е.,

$$\overline{A} = \{x \colon x \in U \ \& \ x \notin A\}.$$

4. *Разность* состоит из элементов множества A и не включает элементов B, т.е.,

$$A \setminus B = \{x \colon x \in A \& x \notin B\}.$$

5. *Симметрическая разность* состоит только из элементов A или только из элементов B, т.е.,

$$A \triangle B = \{x \colon (x \in A \& x \notin B) \vee (x \in B \& x \notin A)\}$$

6. *Декартовое* (*прямое*) *произведение* состоит из всевозможных упорядоченных пар элементов A и B, т.е.,

$$A \times B = \{(a, b) \colon a \in A \ \& \ b \in B\}$$

Операции (1) – (3) могут быть представлены с помощью диаграммы Эйлера-Венна (Рис. *I.1.1*), в которой универсум U изображается прямоугольником, а множества A и B – окружностями. Для выделения результата применяется штриховка.

НИИ Искусственный интеллект

Здесь показано, что множества A и B являются подмножествами U, и они записываются как $A \subseteq U$ и $B \subseteq U$ (см. *I*.2.2.).

Операции (1) – (3) можно определить не только над двумя множествами, но и над n множествами $A_1, A_2, \ldots, A_n$, где $n \in N$ & $n > 2$.

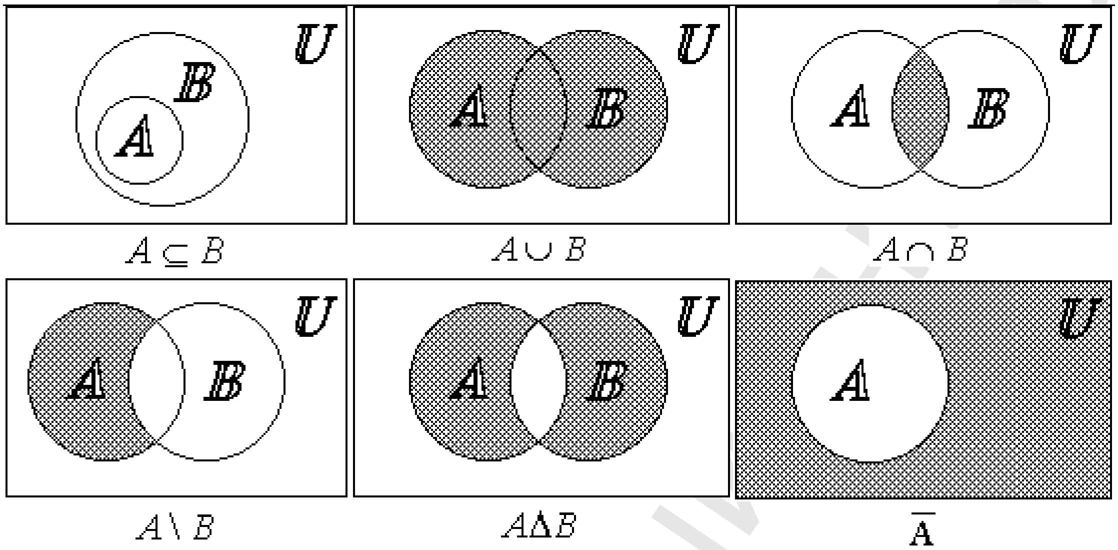


Рис. *I.1.1*. Диаграммы Эйлера-Венна.

Объединение над множествами $A_1, A_2, \ldots, A_n$ определяется как:

$$A_1 \cup A_2 \cup \ldots \cup A_n = \bigcup_{i=1}^{n} A_i.$$

Пересечение над множествами $A_1, A_2, \ldots, A_n$ определяется как:

$$A_1 \cap A_2 \cap \ldots \cap A_n = \bigcap_{i=1}^{n} A_i.$$

Прямое произведение над множествами A_1, A_2, ..., A_n определяется как множество кортежей вида $(a_1, a_2, \ldots, a_n)$, где элементы $a_1 \in A_1$, $a_2 \in A_2$, …, $a_n \in A_n$, т.е.

$$A_1 \times A_2 \times \ldots \times A_n = \{(a_1, a_2, \ldots, a_n): a_1 \in A_1 \;\&\; a_2 \in A_2 \;\&\ldots\&\; a_n \in A_n\}.$$

Здесь, если $A_1 = A_2 = \ldots = A_n = A$, то $\underbrace{A \times A \times \ldots \times A}_{n} = A^n$ – степень.

Примеры I.1.1.2. Пусть $A = \{a, b, c, d, e, f\}$, $B = \{c, d\}$, то:

1. $A \cup B = \{a, b, c, d, e, f\}$;
2. $A \cap B = \{c, d\}$;
3. $A \setminus B = \{a, b, e, f\}$;
4. $A \triangle B = \{a, b, e, f\}$
5. $B \times B = \{(c, c), (c, d), (d, c), (d, d)\}$;

6. $\overline{A}$ зависит от того, какой будет универсум *U*. Допустим, если $U = \{a, b, c, d, e, f, h\}$, то $\overline{A}=\{h\}$.

Теперь можно показать табличный способ задания множеств и операции над ними. Пусть заданы U, $A \subseteq U$ и $x \in U$.

***Определение* I.1.1.3.** *Индикаторной* (*характеристической*) функцией для множества *A* называется функция $I_A(x)$, заданная как:

$$I_A(x)=\begin{cases} 1, & \text{если} \quad x\in A \\ 0, & \text{если} \quad x\notin A \end{cases}$$

Таким образом: $I_A\colon\ U\to\{0,1\}$.

Для $A \subseteq U$ и $B \subseteq U$ имеют место следующие свойства:

$$I_A(x)=I_B(x)\Leftrightarrow A=B;$$
$$I_A(x)\le I_B(x)\Leftrightarrow A\subseteq B;$$
$$I_{\overline{A}}(x)=1-I_A(x);$$
$$I_{A\cup B}(x)=I_A(x)+I_B(x)-I_A(x)\cdot I_B(x);$$
$$I_{A\cap B}(x)=I_A(x)\cdot I_B(x);$$
$$I_{A\setminus B}(x)=I_A(x)-I_A(x)\cdot I_B(x);$$
$$I_{A\Delta B}(x)=I_A(x)+I_B(x)-2I_A(x)\cdot I_B(x).$$

Индикаторы удобно задавать с помощью таблицы *I.1.1*.

Таблица *I.1.1*. Индикаторы.

$x\in A$	$x\in B$	$x\in A\cup B$	$x\in A\cap B$	$x\in A\setminus B$	$x\notin A$	$x\in A\Delta B$
0	0	0	0	0	1	0
0	1	1	0	0	1	1
1	0	1	0	1	0	1
1	1	1	1	0	0	0

Операции над множествами обладают следующими свойствами:

I. ***Объединение***, ***пересечение и разность***:

1) $A\cup\varnothing = A$ – свойство нуля;
2) $A\cup A = A$ – идемпотентность;

НИИ искусственный интеллект

3) $A \cup B = B$, если все элементы A содержатся в B;

4) $A \cup B = B \cup A$ – коммутативность;

5) $(A \cup B) \cup C = A \cup (B \cup C) = A \cup B \cup C$ – ассоциативность;

6) $A \cap \varnothing = A$ – свойство нуля;

7) $A \cap A = A$ – идемпотентность;

8) $A \cap B = A$, если A все элементы A содержатся в B;

9) $A \cap B = B \cap A$– коммутативность;

10) $(A \cap B) \cap C = A \cap (B \cap C) = A \cap B \cap C$ – ассоциативность;

11) $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$ – дистрибутивность;

12) $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$ – дистрибутивность;

13) $A \cap (B \setminus C) = (A \cap B) \setminus (A \cap C)$ – дистрибутивность;

14) $A \cup \overline{A} = U$ – свойство дополнения;

15) $A \cap \overline{A} = \varnothing$ – свойство дополнения;

16) $\overline{A \cup B} = \overline{A} \cap \overline{B}$ – закон де Моргана

17) $\overline{A \cap B} = \overline{A} \cup \overline{B}$ – закон де Моргана

18) $\overline{\overline{A}} = A$– инволютивность;

19) $A \setminus \varnothing = A$ – свойство разности;

20) $A \setminus A = \varnothing$ – свойство разности;

21) $A \setminus B = A \cap \overline{A} = \varnothing$ – свойство разности;

22) $B \setminus A = B \cap \overline{A} = B \setminus (B \cap A)$– свойство разности;

II. Симметрическая разность и прямое произведение:

1) $A \triangle \varnothing = A$ – свойство нуля;

2) $A \triangle A = \varnothing$ – идемпотентность;

3) $A \triangle B = (A \cup B) \setminus (A \cap B)$ – свойство симметрической разности;

4) $A \triangle B = B \triangle A$ – коммутативность;

5) $(A \triangle B) \triangle C = A \triangle (B \triangle C) = A \triangle B \triangle C$ – ассоциативность;

6) $(A \cup B) \times C = (A \times C) \cup (B \times C)$ – дистрибутивность;

7) $A \times (B \cup C) = (A \times B) \cup (A \times C)$ – дистрибутивность;

8) $(A \cap B) \times C = (A \times C) \cap (B \times C)$ – дистрибутивность;

9) $A \times (B \cap C) = (A \times B) \cap (A \times C)$ – дистрибутивность;

10) $(A \setminus B) \times C = (A \times C) \setminus (B \times C)$ – дистрибутивность;

НИИ Искусственный интеллект

11) $A\times(B\backslash C) = (A\times B) \setminus (A\times C)$ – дистрибутивность;

Примеры I.1.1. Пусть $A=\{1, 2\}$, $B=\{a, b\}$, $C=\{+, -\}$. Тогда дистрибутивность $(A\cup B)\cup C = A\cup(B\cup C) = A\cup B\cup C$ определяется как $(\{1,2\}\cup\{a,b\})\cup\{+,-\}=\{1,2\}\cup(\{a,b\})\cup\{+,-\}) = \{1,2\}\cup\{a,b\}\cup\{+,-\}$.

Задания I.1.1. Пусть $A=\{1, 2, 4\}$, $B=\{3, 4, 5, 6\}$. Выполните:

1) $A\cup\varnothing$;
2) $A\cup A$;
3) $A\cap B$;
4) $A\times B$;
5) $A \setminus B$;
6) $A\cup\overline{A}$;
7) $A\triangle B$;
8) $(A\cup B)\times C$;
9) $A\times(B\cup C)$;
10) $(A\backslash B)\times C$;
11) $(A\cap B)\times C$;
12) $(A\triangle B)\triangle C$.

Вопросы I.1.1:

1. Как задаются множества?
2. Что такое универсальное множество?
3. Как определяется подмножества?
4. Что такое диаграмма Эйлера-Венна?
5. Как выглядит диаграмма Эйлера-Венна для объединения?
6. Как определяется прямое произведение множеств?
7. Как определяется разность множеств?
8. Как определяется симметричная разность множеств?
9. Как определяется индикаторная функция множества?
10. В чем заключается закон де Моргана?
11. Если $A=\{1,2,3\}$, $B=\{3,4,5\}$, то что означает $\{1,2,3,4,5\}$?
12. Если $A=\{1,2,4\}$, $B=\{4,3,2\}$, то что означает $\{2,4\}$?

НИУ Искусственный интеллект

13. Если A={1,2,3,4}, B={3,4,5,6}, то что означает {2,4}?
14. Что такое дистрибутивность?
15. Что такое ассоциативность?
16. Что такое коммутативность?

НИИ Искусственный интеллект

I.1.2. Отношения и способы их представления

В этом параграфе будут определены понятия отношения и даны их представления, а также приведены примеры, предложены задания, заданы вопросы. При формировании учебного материала данного параграфа были использованы следующие источники [14, 17, 22, 27, 28, 33].

Определения I.1.2.1. Если заданы два множества А и В одного и того же типа, то можно вести следующие отношения:

1) $A = B$: *A равно B*, если A и B состоят из одних и тех же элементов, т.е. A и B являются подмножествами друг друга;

2) $A \subseteq B$: *A содержится в B*, если все элементы A принадлежат B или A равно B, это означает, что A является подмножеством B;

3) $A \subset B$: *A строго содержится в B*, если все элементы A принадлежат B и A не равно B, т.е. некоторые элементы B не принадлежат A, это означает, что A является собственным подмножеством B.

Аналогично можно определить отношения включает $A \supseteq B$ и строго включает $A \supset B$.

Нетрудно заметить, что выше введенные отношения $=$, $\subseteq$ и $\subset$ являются подмножествами прямого произведения $A \times B$.

Таким образом, можно считать, что любое отношение – это некоторое подмножество в прямом произведении, выделяемое определенным законом.

Замечание I.1.2.1. Пустое множество $\varnothing$ является собственным подмножеством любого конечного множества.

Примеры I.1.2.1:

1) если $A=\{a,b,c\}$, $B=\{b,a,c\}$, то $A=B$;
2) если $A=\{1,2,3,4\}$, $B=\{3,1,4,2\}$, то $A \subseteq B$;
3) если $A=\{1,2,3\}$, $B=\{3,1,4,2\}$, то $A \subset B$

Определение I.1.2.2. Пусть A_1, A_2, ..., A_n − произвольные множества, не обязательно различные, $n \geq 1$. Тогда *n-арным отношением* на множествах A_1, A_2, ..., A_n является подмножество

$$R^n \subseteq A_1 \times A_2 \times \ldots \times A_n,$$

НИИ государственный интеллект

где R^1 – унарное отношение на A_1, R^2 – бинарное отношение на A_1, A_2, R^3 – тренарное отношение на A_1, A_2, A_3 и т.д.

Всякое *унарное отношение* на множестве A является характеристическим свойством некоторого его подмножества. Множество всех унарных отношений на A совпадает с множеством всех подмножеств множества A.

Примеры I.1.2.2:

1. Унарное отношение $R_1^1=\{n:\ n\in N \& n<100\}$ определяет множество натуральных чисел меньше числа 100;

2. Унарное отношение $R_2^1=\{n:\ \forall k\in N(n=2*k)\}$ определяет множество четных натуральных чисел;

3. Унарное отношение $R_3^1=\{n:\ \forall k\in Z(n=2*k+1)\}$ определяет множество нечетных целых чисел.

Определения I.1.2.3. *Бинарное отношение* определяется над парами множества и может представлено одним из трех способов:

1) *префиксная запись* – знак отношения вставляется перед участниками бинарного отношения;

2) *инфиксная запись* – знак отношения вставляется между участниками бинарного отношения;

3) *постфиксная запись* – знак отношения вставляется после участниками бинарного отношения.

Бинарные отношения над парами элементов часто представляют с помощью таблиц: строки соответствуют первым элементам пары, столбцы – вторым элементам пары, а наличие отношения между конкретными элементами строки и столбца отмечается специальным знаком, например, знаком «1» или др.

Примеры I.1.2.3:

1. Если $a\in A$ и $b\in B$ находятся в бинарном отношении R, то это можно записать как:

   Rab – префиксная запись;

   aRb – инфиксная запись;

   abR – постфиксная запись.

НИИ искусственный интеллект

2. Если множества $A = \{a_1, ..., a_r\}$ и $B = \{b_1, ..., b_s\}$ находятся в бинарном отношении R, то его можно представить с помощью таблицы I.1.2, в которой элементы a_i представляются строками, элементы b_j, – столбцами, а отношения a_iRb_j отмечается “1”:

Таблица I.1.2. Двоичное отношение.

R	b_1	b_2	...	b_{s-1}	b_s
a_1	1		...	1	
a_2		1	...		
...	...	...	...	...	...
a_{r-1}	1		...	1	
a_r			...		1

Определения I.1.2.4. Говорят, что бинарное отношение R на множестве S:

1) *рефлексивно*, если для каждого $s \in S$ имеет место sRs;

2) *транзитивно*, если для любых $s,t,u \in \mathrm{S}$ из sRt и tRu следует sRu;

3) *симметрично*, если для любых s, $t \in S$ из sRt следует tRs;

4) *антисимметрично*, если из aRb и bRa следует $a=b$.

Примеры I.1.2.4:

1. Отношение > над числами не является рефлексивным;
2. Отношения =, ≥, > над числами являются транзитивными;
3. Отношение = над числами является симметричным.

Определение **I.1.2.5.** Бинарное отношение R называется отношением *эквивалентности*, если оно удовлетворяет свойствам рефлексивности, транзитивности и симметричности.

Каждому отношению эквивалентности на множестве S соответствует единственное разбиение данного множества на смежные классы.

НИИ искусственный интеллект

Примеры I.1.2.5. Отношение = в любом числовом множестве будет отношением эквивалентности, т.е., для любых k, m, $n \in N$:

1. Рефлексивность: n = n.
2. Транзитивность: из $k <m$ и $m <n$ следует $k <n$.
3. Симметричность: из $k = m$ следует $m = k$.

Определение I.1.2.6. Бинарное отношение R на некотором множестве S, удовлетворяющее свойствам рефлексивности, транзитивности и антисимметричности, называется отношением *частичного порядка*.

Примеры I.1.2.6. Отношениями частичного порядка является:

1. Отношение ⊆ для подмножеств некоторого множества;
2. Отношение ⊇ для подмножеств некоторого множества;
3. Отношение = на множестве целых чисел;
4. Отношение ≤ на множестве натуральных чисел;
5. Отношение ≥ на множестве целых чисел.

Определения I.1.2.7:

1. *Частично упорядоченным множеством* называется множество A с определенным на нем отношением частичного порядка. Точнее говря, частично упорядоченным множеством называется пара $<A, R>$, где A – множество, а R – отношение частичного порядка на A.

2. Элементы a и b частично упорядоченного множества A называются *сравнимыми* относительно частичного порядка R на этом множестве, если имеет место aRb или bRa.

3. Частичный порядок на множестве A называется *линейным порядком*, если любые два элемента a и b из A сравнимы относительно частичного порядка R.

4. *Линейно-упорядоченным множеством* или *цепью* называется частично-упорядоченное множество, в котором элементы каждой пары сравнимы.

5. Элементов a и b частично упорядоченного множества A называют *несравнимыми*, если между ними не выполнено ни одно отношение частичного порядка. Возможность существования

несравнимых элементов объясняет смысл термина *«частично упорядоченное множество»*.

Примеры 1.1.2.7:

1. Множество натуральных чисел N с отношением $\leq$ является частично-упорядоченным, все натуральные числа будут сравнимыми относительно отношения $\leq$ и N есть цепь.

2. Множество всех действительных чисел с отношением $=$ является линейно упорядоченным множеством, если все действительные числа будут сравнимыми относительно $=$.

3. Пусть A – множество действительнозначных функций на отрезке $[0,1]$ с определенным на нем отношением частичного порядка $<, =, >$, то элементы $f(x)=x$ и $g(x)=1\text{-}x$ будут несравнимы.

Определения I.1.2.8. Пусть A – частично-упорядоченное множество, B – его подмножество, т.е., $A \supseteq B$. Тогда:

1) *нижней гранью* (*верхней гранью*) множества B во множестве A называется элемент $a \in A$, такой, что $a \leq b$ ($b \leq a$) для любого $b \in B$;

2) элемент $a \in A$ называется *наименьшим* (*наибольшим*) во множестве A, если a есть нижняя (верхняя) грань самого A;

3) элемент $a \in A$ называется *минимальным* (*максимальным*) во множестве A, если не существует $b \in A$, такого, что $b < a$ ($a < b$).

Наименьший (наибольший) элемент множества A является его единственным минимальным (максимальным) элементом.

Примеры I.1.2.8:

1. Множество всех подмножеств множества A имеет наименьший элемент $\varnothing$ и наибольший элемент само A.

2. Множество N натуральных чисел имеет наименьший элемент 1 и не имеет наибольшего элемента.

3. Множество Z всех целых чисел не имеет наименьшего, наибольшего, минимального и максимального элемента.

Элементы множеств A и B находятся во *взаимно-однозначном соответствии*, если каждому элементу $a \in A$ по некоторому закону сопоставлен один и тот же элемент $b \in B$, причем каждый $b \in B$ оказывается сопоставленным одному и тому же $a \in A$.

НИИ Искусственный интеллект

Множества A и B являются *эквивалентными* (*равномощными*), если можно установить взаимно-однозначное соответствие между их элементами.

Замечание I.1.2.2. Бинарное отношение может задаться тройкой множеств $<R, A, B>$, где $R \subseteq A \times B$ – график отношения и записываться $(a, \underline{b}) \in R$ или aRb. Тогда можно определить:

Область определения: Dom $R = \{x \in A: \exists y \in B (x, y) \in R\}$;

Область значения: Run $R = \{y \in B: \exists x \in A (x, y) \in R\}$;

Обратное отношение: $R^{-1} = \{(y, x) \in B \times A: (x, \underline{y}) \in R\}$;

Композиция отношения: $R \subseteq A \times B$, $S \subseteq B \times C$,

$R \cdot S = \{(x, z) \in A \times C: \exists y \in B\ [(xRy)\ \&(ySz)]\}$.

Определение I.1.2.9. Бинарное отношение $f \subseteq X \times Y$ называется *функцией* из X в Y, если Dom $R = X$ и $(x, y) \in f$, $(x, z) \in f \Rightarrow y = z$.

Функция $f: X \rightarrow Y$ называется:

1) *сюръективной*, если для любого $y \in Y$ существует $x \in X$ такой, что $y = f(x)$, т.е., $\forall y \in Y\ \exists x \in X (y = f(x))$;

2) *инъективной*, для любых x_1, $x_2 \in X$ из того, что $x_1 \neq x_2$ следует $f(x_1) \neq f(x_2)$, т.е., $\forall x_1 \in X\ \forall x_2 \in X\ (x_1 \neq x_2 \Rightarrow f(x_1) \neq f(x_2))$;

3) *биективной*, если она сюръективна и инъективна.

Любую бинарную функцию можно связать с тренарным отношением, например, если задана бинарная функция $f(x,y)$, то ее можно связать с тренарным отношением $R^3(x,y,z)$ так, чтобы $z=f(x,y)$.

Примеры I.1.2.9. Пусть x, y, $z \in N$ – натуральные числа и задана бинарная функция $f(x,y)=z$, тогда:

1) если x есть 3, y есть 5, z есть 8, а f есть операция сложение +, то вместо записи $f(x,y)=z$ можно писать 3+5=8 и с ней связать тренарное отношение *Add*(3, 5, 8), для которого справедливо 8 = 3+5.

2) если x есть 8, y есть 2, z есть 4, а f есть операция деление «:», то вместо записи $f(x,y)=z$ можно писать 8:2 = 4 и с ней связать тренарное отношение *Dev*(8, 2, 4), для которого справедливо 4 = 8:2.

НИИ Искусственный интеллект

Ясно, что для некоторых x, $y \in N$ результатом выполнения операции деления является не целое число и для них тренарное отношение $Dev(x, y, z)$ не выполняется. Поэтому для таких случаев необходимо определить дополнительные условия о результатах.

Определение I.1.2.10. *Транзитивным замыканием* отношения R на множестве A называется пересечение всех транзитивных отношений, содержащих R как подмножество (иначе, минимальное транзитивное отношение, содержащее R как подмножество).

Транзитивное замыкание существует для любого отношения. Для этого отметим, что пересечение любого множества транзитивных отношений транзитивно. Более того, обязательно существует транзитивное отношение, содержащее R как подмножество.

Транзитивное замыкание обладает следующими свойствами:

1) Транзитивное замыкание рефлексивного отношения рефлексивно, т.к. транзитивное отношение содержит исходное отношение;

2) Транзитивное замыкание симметричного отношения симметрично. Действительно, пусть имеется транзитивное отношение aRb, значит существуют x_1, x_2, ..., x_n такие, что aRx_1, x_1Rx_2, …, x_nRb. Но из симметричности отношения R следует bRx_n, x_nRx_{n-1}, …, x_1Ra, следовательно, bRa.

3) Транзитивное замыкание не сохраняет антисимметричность, например, для отношения $\{(a,b), (b,c), (c,a)\}$ на множестве $\{a, b, c\}$.

4) Транзитивное замыкание транзитивного отношения - оно само.

Отношение $R^*=R^+\cup R^0$, где $R^0=\{(\varepsilon, \varepsilon): \varepsilon \in A\}$ иногда называют рефлексивно-транзитивным замыканием, хотя часто под "транзитивным замыканием" подразумевается именно R^*. Обычно различия между этими отношениями не являются значительными.

Примеры I.1.2.10:

1. Для любых a, b, $c \in N$ из $a < b$ и $b < c$ следует $a < c$.
2. Для любых x, $y \in N$ из $x=y$ следует $y= x$.

НИИ искусственный интеллект

3. Если A – множество городов, и на них задано отношение xRy, означающее "существует автобусный маршрут из x в y", то транзитивным замыканием этого отношения будет отношение "существует возможность добраться автобусом из x в y".

Задания I.1.2. При $A=\{a,b,c,d\}$, $B=\{b,d\}$, $C=\{c\}$ определить:

1. $A=B$;
2. $A\supseteq C$;
3. $A\supseteq (B\cup C)$;
4. Дайте определение пустого множества;
5. Дайте определение бинарного отношения;
6. Дайте определение n-арного отношения;
7. Представьте префиксную запись бинарного отношения между множествами A и B;
8. Представьте инфиксную запись бинарного отношения между множествами A и B;
9. Представьте постфиксную запись бинарного отношения между множествами A и B;
10. Представьте бинарное отношение с помощью таблицы;
11. Дайте определение отношения эквивалентности;
12. Дайте определение линейно-упорядоченного множества;
13. Дайте определение нижней грани множества.

Вопросы I.1.2:

1. Что определяет унарное отношение на множестве A?
2. Сколько множеств участвуют в бинарном отношении?
3. Как определяется рефлексивность бинарного отношения?
4. Как определяется транзитивность бинарного отношения?
5. Как определяется симметричность бинарного отношения?
6. В чем заключается частичная упорядоченность множества?
7. Как определяется композиция отношений?
8. Из чего состоит мощность объединения двух множеств?
9. Как определяется эквивалентность двух множеств?
10. Какие множества являются равномощными?
11. Как определяется отношение частичного порядка?

НИИ искусственного интеллект

12. Какие множества называются несравнимыми?
13. Что такое транзитивное замыкание?

НИИ Искусственный интеллект

II.2. Элементы теории функций

I.2.1. Определения, значения и производная функции

В этом параграфе будут определены понятия простой функции, сложной функции и обратной функция, будут даны способы их задания и определения значений. Будут приведены примеры, предложены задания, заданы вопросы. При формировании учебного материала данного параграфа были использованы [14,17].

Пусть X, Y – множества некоторых (числовых, логических, символьных или других) значений. Тогда если каждому значению переменной величины x из множества X по определенному правилу f соответствует единственное значение величины y из множества Y, то говорят, что задается *простая функция* $y = f(x)$. Для простой функции множество X называется *областью задания* или *областью определения*, независимая переменная $x \in X$ называется *аргументом*, множество Y называется *областью значений* или *областью изменения*, а зависимая переменная $y \in Y$ – *значением.*

В общем случае, для простой функции f область определения обозначается как $D(f)$, а область значений – $E(f)$, т.е. можно писать $D(f) = X$, $E(f) = Y$.

Простую функцию можно задавать аналитически в виде некоторого алгебраического выражения, указывающего операции и последовательность их выполнения, которые необходимо совершить над значением независимой переменной, в результате которых получается значение зависимой переменной.

Функцию можно задавать в виде таблицы, указывая значения независимой переменной x в определенном порядке $x_1, x_2, \ldots, x_n$, а рядом с ними указывая соответствующие им значения зависимой переменной $y_1, y_2, \ldots, y_n$, например, в виде следующей таблицы.

Значение аргумента	x_1	x_2	…	x_n
Значение функции	y_1	y_2	…	y_n

Табличный способ задания функции дает возможность определить конкретные значения функции сразу, без каких-либо

НИИ Искусственный интеллект

дополнительных вычислений, но не дает наглядного изображения изменения функции в зависимости от аргумента.

Теперь покажем, как получается сложная функция от заданной простой функции.

Пусть задана простоя функция $u = \varphi(x)$ с областью определения $D(\varphi\ ) = X$ и областью значений $E(\varphi) = D_1$ и $y = \psi(u)$ с областью определения $D(\psi) = D_1$ и областью значений $E(\psi) = Y$. Тогда последовательное применение к каждому значению x соответствий φ и ψ определяет определенное значение y. Это записывают так:

$$y = \psi(\varphi(x)).$$

Значение функции $u=\varphi(x)$ называют *промежуточной переменной* или *промежуточным аргументом*, функция от функции $y = \psi(\varphi(x))$ называется *сложной функцией* независимой переменной x или *суперпозицией* или *композицией* двух функций: внутренней φ и внешней ψ. При этом область определения внешней функции должна содержать область значений внутренней функции, т.е. $D(\psi) \supset E(\varphi)$.

Обычно для суперпозиции используется запись $y = \psi(u)$, где $u=\varphi(x)$, а для композиции – обозначение $\psi \cdot \varphi$.

Очень важным является понятие обратной функции, которую можно определить следующим образом.

Если задана функция $y = g(x)$, то *обратной функцией* называется такая функция g^{-1}, которая каждому элементу y из области значений $E(g)$ ставит в соответствие все те элементы x из области определения $D(g)$, для которых выполняется $g(x) = y$, т.е. $g^{-1}(y) = \{ x \mid g(x) = y \}$.

Функции g и g^{-1} называются *взаимно обратными*. Для взаимно обратных функций выполняются соотношения $D(g) = E(g^{-1})$ и $D(g^{-1}) = E(g)$.

Если задана функция $y = g(x)$, то чтобы найти обратную ей функцию g^{-1}, необходимо решить уравнение относительно переменной y, поменяв местами переменные x и y.

Значение $A \in E(f)$ называется *пределом функции* $f(x)$ в точке

НИИ искусственный интеллект

$x_0 \in D(f)$ по Коши, если для любого наперёд взятого положительного числа $\varepsilon > 0$ найдётся отвечающее ему число $\delta(\varepsilon) > 0$ такое, что для всех аргументов x, удовлетворяющих условию $0 < |x - x_0| < \delta$, выполняется неравенство $|f(x) - A| < \varepsilon$.

$$\lim_{x \to x_0} f(x) = A \Leftrightarrow \forall \varepsilon > 0 \ \exists \delta(\varepsilon) > 0 \ \forall x\colon \ 0 < |x - x_0| < \delta \Rightarrow |f(x) - A| < \varepsilon.$$

Функция $f(x)$ *непрерывна в точке* $x_0 \in D(f)$, если для любого $\varepsilon > 0$ существует $\delta(\varepsilon) > 0$ такое, что для любого $x \in D(f)$ выполняется $0 < |x - x_0| < \delta \Rightarrow |f(x) - f(x_0)| < \varepsilon$.

Функция f *непрерывна на множестве* E, если она непрерывна в каждой точке данного множества.

Пусть $f(x)$ – некоторая числовая функция, определенная в интервале $]a; b[$ и x_0 – некоторая фиксированная точка (некоторое постоянное значение) в этом интервале. Для произвольного значения x из этого интервала составим разность $x - x_0 = \Delta x$ и назовем ее приращением независимой переменной (аргумента). Приращением функции $f(x)$ в точке x_0 называют разность между значением в точке $x = x_0 + \Delta x$ и значением в точке x_0 и обозначают через выражение

$$\Delta f(x_0) = f(x_0 + \Delta x) - f(x_0).$$

Поскольку x_0 считается фиксированной (постоянной), то приращение функции $\Delta f(x_0)$ является функцией приращения аргумента Δx. Составим отношение

$$\frac{f(x_0 + \Delta x) - f(x_0)}{\Delta x} = \frac{\Delta f(x_0)}{\Delta x},$$

которое также будет функцией приращения аргумента Δx, и рассмотрим предел этого выражения при Δx, стремящимся к нулю. Если этот предел существует, то говорят, что функция $f(x)$ имеет производную в точке x_0, и пишут

$$\lim_{\Delta x \to 0} \frac{f(x_0 + \Delta x) - f(x_0)}{\Delta x} = \frac{\Delta f(x_0)}{\Delta x} = f'(x_0).$$

НИИ Искусственный интеллект

Число $f'(x_0)$ называется *производной функции* в точке x_0. Производную функции $f(x)$ в точке x_0 также обозначают

$$f'(x)_{x=x_0} \text{ или } \frac{df(x_0)}{dx}.$$

Говорят, что функция $f(x)$ *дифференцируема* в точке x_0 , если она имеет производную в точке x_0 .

Если функция $f(x)$ *дифференцируема* в каждой точке интервала $]a; b[$, то говорят, что она дифференцируема в интервале $]a; b[$.

Производная функции $f(x)$, дифференцируема в интервале $]a; b[$ сама является функцией x.

Пусть заданы две функции $u(x)$ и $v(x)$. Тогда можно привести правила вычисления производных суммы, разности, произведения и деления функций:

$$1.\ (u \pm v)' = u' \pm v';$$

$$2.\ (u * v)' = u'v + v'u;$$

$$3.\left(\frac{1}{u}\right)' = \frac{u'}{u^2};$$

$$4.\left(\frac{u}{v}\right)' = \frac{u'v - v'u}{v^2}.$$

Пусть задана сложная функция $y = f(g(x))$. Тогда производная этой сложной функции вычисляется по формуле

$$y' = (f(g(x)))' = f'(g(x)) \cdot g'(x).$$

Значение аргумента $x_0 \in D(f)$ называется *точкой локального минимума* (или *локального максимума*) функции $f(x)$, если для этого значения x_0 и произвольной величины $\delta>0$ найдется такая окрестность $]x_0-\delta,\ x_0+\delta[\ \subset D(f)$, что для всех значений x из этой окрестности выполняется неравенство $f(x_0) < f(x)$ (или $f(x_0) > f(x)$).

Значение функции в точке локального минимума (или локального максимума) x_0 называется *локальным минимумом* (или *локальным максимумом*) функции $f(x)$.

НИИ Искусственный интеллект

Функция $y = f(x)$ при $x=x_0$ имеет локальный максимум, если значение $f(x_0)$ больше всех других ее значений, принимаемых вблизи точки x_0, т.е. $f(x_0)>f(x)$ для $x\in]x_0-\delta, x_0+\delta[\ \&\delta>0$.

Функция $y = f(x)$ при $x=x_0$ имеет локальный минимум, если значение $f(x_0)$ меньше всех других ее значений, принимаемых вблизи точки x_0, т.е. $f(x_0)<f(x)$ для $x\in]x_0-\delta, x_0+\delta[\ \&\delta>0$.

Максимум и минимум функции называют *экстремумом.* При этом если выполняется условия $f(x_0) \le f(x)$ (или $f(x_0) \ge f(x)$), то экстремум (минимум или максимум) называют *нестрогим*, т.е. в указанной δ окрестности могут быть совпадающие (равные) значения функции.

Пусть точка x_0 является точкой экстремума функции $f(x)$, определенной в некоторой окрестности точки x_0. Тогда либо производная $f'(x_0)$ не существует, либо $f'(x_0)=0$.

Пусть функция $f(x)$ дважды дифференцируема в точке x_0. Тогда при условии $f'(x_0)=0$ и $f''(x_0)<0$ точка x_0 является точкой локального максимума. А если $f'(x_0)=0$ и $f''(x_0)>0$, то x_0 является точкой локального минимума.

I.2.1. Задания:

1. Найдите обратную функцию функции $y=log_2x$.
2. Найдите области определения и значения функции $y=arctg^2x$.
3. Найдите минимум функции $y= x^2$.
4. Дайте определение непрерывности функции.

I.2.1. Вопросы:

1. Как определяются сложные функции?
2. Какие определяются обратные функции?
3. Что такое максимум и минимум функции?
4. Что такое дифференцируемость функции в точке x_0?

НИИ искусственный интеллект

I.2.2. Числовые функции и их свойства

В этом параграфе будут рассмотрены виды числовых функций, такие как, простая функция, сложная функция и обратная функция, будут даны способы их определения, а также приведены примеры, предложены задания, заданы вопросы. При формировании учебного материала данного параграфа были использованы следующие источники [14, 17, 22, 27, 28, 33].

При определении и изучении видов функций очень важным являются случаи, когда для функции $y = f(x)$ область определения $D(f)$ и область значений $E(f)$ задаются множествами чисел.

Пусть $D(f)$ и $E(f)$ – множества чисел, тогда говорят, что на множестве $D(f)$ задана *числовая функция* от числового аргумента x и с числовым значением $f(x)$.

Среди числовых функций интерес представляет *действительные функции*, у которых $D(f)$ и $E(f)$ являются подмножествами множества действительных чисел R, т.е. $D(f) \subseteq R$ и $E(f) \subseteq$ R.

Имеется большое разнообразие действительных функций, но среди них наиболее изучены *основные элементарные функции*.

К основным элементарным функциям относятся:

1. Степенная функция $y=x^a$ ($a \in R$ – постоянная величина);
2. Показательная функция $y=a^x$ ($a \in R$ –$a>0$, $a \neq 1$);
3. Логарифмическая функция $y=log_a x$ ($a \in R$ –$a>0$, $a \neq 1$);
4. Тригонометрические функции:
   $y= sin\ x$, $y= cos\ x$, $y=tg\ x$, $y=ctg\ x$, $y=sec\ x$, $y=cosec\ x$;
5. Обратные тригонометрические:
   $y=arc\ sin\ x$, $y=arc\ cos\ x$, $y=arctg\ x$, $y=arcctg\ x$,
   $y=arcsec\ x$, $y=arccosec\ x$.

Элементарными функциями называются функции, которые можно выразить через основные элементарные функции, используя арифметические операции и суперпозицию конечное число раз.

НИИ искусственный интеллект

Известно, что множество чисел можно изобразить прямой, отметив в ней точками определенные числовые значения. При этом все точки на горизонтальной прямой слева (на вертикальной прямой ниже) от точки, которая отмечает значение 0, изображают отрицательные числа, а справа (выше) - положительные числа.

Для графического изображения заданной функций на оси абсцисс отмечаем ряд значений x_1, x_2, x_3,… аргумента x и строим ординаты y_1, y_2, y_3,… , представляющие соответствующие значения функции y. В результате получаем ряд точек $M_1(x_1,y_1)$, $M_2(x_2,y_2)$, $M_3(x_3,y_3)$,… . Соединяя их кривой линией, получаем график данной функциональной зависимости, представленный на рисунке I.2.2.1.

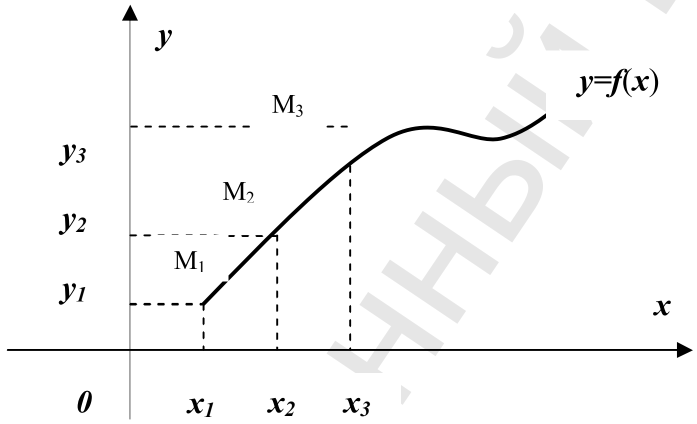


Рисунок I.2.2.1. График функциональной зависимости.

Преимуществом графического изображения по сравнению с табличным являются его наглядность и легкая обозримость; недостатком – малая степень точности. Большое практическое значение имеет удачный выбор масштабов.

I.2.2. Примеры:

1. Из основных элементарных функций ограниченными являются *sin x*, *cos x*, *arcsin x*, *arccos x*, *arctg x*, *arcctg x*, *arcsec x*, *arccosec x*, например, $|\cos x| \le 1$.

2. Все основные элементарные функции монотонны или во всей области определения, или в отдельных ее частях.

3. Возьмем на графике функции $y = f(x)$ произвольная точка M_0 с абсциссой $x_0 \in]a; b[$ и проведем через точку M_0 касательную. Ее уравнение $y = f'(x_0)(x - x_0) + f(x_0)$. Мы должны показать, что

график функции на]a; b[лежит ниже этой касательной (выпуклый), т.е. при одном и том же значении x ордината кривой $y=f(x)$ будет меньше ординаты касательной, как это показано на рисунке I.2.2.2.

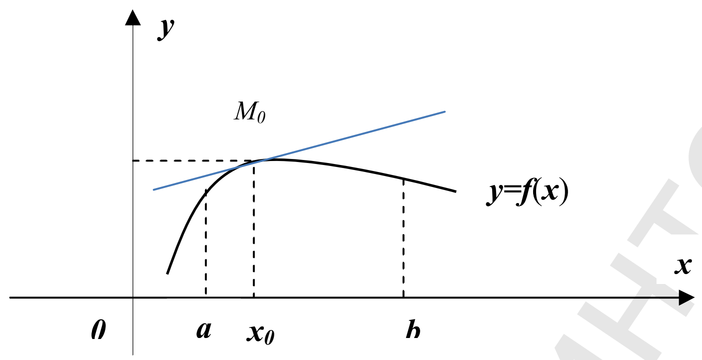


Рисунок I.2.2.2. Выпуклый график.

4. График функции $y = sin\, x$ на [0,2; π], выпуклый в интервале]0; π[и вогнутый в]π; 2π[.

5. График функции $y=x^3$ может быть выпуклым и вогнутым на интервале]-∞ ;+∞[, так на интервале]-∞; 0[– выпуклый, а на интервале]0;+∞[– вогнутый, при этом точка x=0 отделяет выпуклую часть графика функции $y=x^3$ от вогнутой (см. рисунок I.2.2.3.).

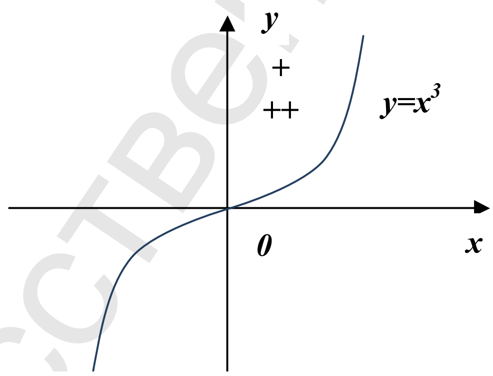


Рисунок I.2.2.3. Выпуклый и вогнутый график $\boldsymbol{y=x^3}$.

I.2.2. Задания:

1. Укажите область определения действительной функции.
2. Укажите область значения действительной функции.
3. Перечислите основные элементарные функции.

I.2.2. Вопросы:

1. Как определяются степенные и показательные функции?
2. Как определяются логарифмические функции?

НИИ Искусственный интеллект

3. Как определяются тригонометрические функции?

НИИ Искусственный интеллект

I.2.3.Действительные функции.

В этом праграфе рассматриваются виды действительных функций, а также приведены примеры, предложены задания, заданы вопросы. При формировании учебного материала данного параграфа были использованы следующие источники [14, 17, 22, 27, 28, 33].

В зависимости от свойств действительные функции подразделяются на следующие виды:

- ограниченные и неограниченные функции;
- возрастающие и убывающие функции;
- ступенчатые функции;
- четные и нечетные функции;
- периодические функции;
- выпуклые и вогнутые функции.

Ограниченные и неограниченные функции.

Функция $y = f(x)$ называется *ограниченной*, если можно указать такое положительное действительное число M, что при любом $x \in D(f)$ выполняется условие $|f(x)| \leq M$.

Функция $y = f(x)$ называется *неограниченной*, если для любого положительного числа M найдется такая точка $x \in D(f)$, что справедливо неравенство $|f(x)| > M$.

График ограниченной функции будет заключен между двумя параллельными прямыми $y = -M$ и $y = M$.

Возрастающие и убывающие функции.

Пусть некоторое числовое множество содержится X в области определения функции $D(f)$, т.е. $X \subseteq D(f)$, тогда говорят:

1) Функция $y = f(x)$ называется *возрастающей* (строго возрастающей) на числовом множестве X, если для любых чисел $x_1 < x_2$ из множества X выполняется неравенство $f(x_1) < f(x_2)$.

НИИ искусственный интеллект

2) Функция $y = f(x)$ называется *убывающей* (*строго убывающей*) на множестве X, если из условия $x_1<x_2$ следует, что $f(x_1)>f(x_2)$.

3) Функция $y = f(x)$, являющаяся во всей своей области определения $D(f)$ возрастающей или убывающей, называется *строго монотонной.*

4) Функция $y=f(x)$ называется *неубывающей* на множестве X, если для любых чисел $x_1<x_2$ из множества X выполняется условие $f(x_1)\leq f(x_2)$.

5) Функция $y=f(x)$ называется *невозрастающей*, если из условия $x_1<x_2$ следует, что $f(x_1)\geq f(x_2)$.

6) Функция, являющаяся в каждом связном подмножестве своей области определения неубывающей или невозрастающей, называется *монотонной.*

7) Сумма двух возрастающих (убывающих) функций является возрастающей (убывающей) функцией.

8) Произведение двух положительных (возрастающих (убывающих) функций является возрастающей (убывающей) функцией.

9) Если функция $y = f(x)$ возрастающая и $f(x) \neq 0$, то функция $1/f(x)$ убывающая, и наоборот.

10) Если функция $y = f(x)$ строго монотонна, то обратная функция $x = f^{-1}(y)$ строго монотонна.

11) Функция $y = f(x)$ называется *кусочно-монотонной* на множестве X, если это множество можно разбить на конечное число подмножеств так, что на каждом из них данная функция является монотонной.

Если функция $y = f(x)$ возрастающая (или убывающая) на множестве X, то она на этом множестве *обратима*, т.е. из условия $x_1\neq x_2$ следует, что справедливо $f(x_1) \neq f(x_2)$.

Ступенчатые функции.

НИИ Искусственный интеллект

Если можно разбить область определения $D(f)$ на взаимно непересекающиеся подобласти D_1, D_2 ,…, то при аналитическом задании функции при одних значений аргумента $x \in D_1$ функция может задаваться одним выражением, а при других значений аргумента $x \in D_2$ – вторым выражением и т.д. Иногда такие функции называются *ступенчатой функцией.*

Четные и нечетные функции.

Функция $f(x)$ называется *четной*, если для всех значений аргумента $x \in D(f)$ выполняется равенство $f(-x) = f(x)$.

Функция $f(x)$ называется *нечетной*, если для всех значений аргумента $x \in D(f)$ выполняется равенство $f(-x) = -f(x)$.

Четные и нечетные функции имеют следующие свойства:

1) Сумма двух четных (нечетных) функций является четной (нечетной) функцией.

2) Произведение двух четных (нечетных) функций является четной функцией, а произведение четной и нечетной функций есть функция нечетная.

3) Если первая функция $y = \psi(u)$ и вторая функция $u = \varphi(x)$ нечетные, то сложная функция $y = \psi(\varphi(x))$ также нечетная.

4) Если первая функция $y = \psi(u)$ четная, а вторая функция $u = \varphi(x)$ нечетная, то сложная функция $\psi(\varphi(x))$ четная.

5) Если функция $f(x)$ четная и выполняется условие $f(x) \neq 0$, то функция в виде выражения $1 / f(x)$ тоже четная.

График четной функции симметричен относительно оси ординат, а нечетной – относительно начала координат. При этом области определения четных и нечетных функций симметричны относительно начала координат. Это необходимое (но не достаточное) условие того, чтобы функция была четной или нечетной.

Периодические функции.

НИИ искусственный интеллект

Функция $f(x)$ называется *периодической*, если ее значение y не меняется при прибавлении к значению аргумента $x \in D(f)$ некоторого фиксированного ненулевого числа T, т.е. для любого $x \in D(f)$ и $T \neq 0$ будет выполняться равенство $f(x) = f(x+T)$. При этом число T называется *периодом* функции $f(x)$.

Наименьший из положительных периодов называют *основным периодом* периодической функции. Если периодическая функция имеет основной период T_0, то любой другой ее период будет кратным основному периоду T_0.

Выпуклые и вогнутые функции.

Функция называется выпуклой, если её график (кривая) $y = f(x)$ является выпуклым (обращенным выпуклостью вверх) на интервале]a; b[, если он расположен ниже любой своей касательной на этом интервале.

График функции (кривая) $y = f(x)$ называется *вогнутым* (обращенным выпуклостью вниз) на интервале $]a; b[$, если он расположен выше любой своей касательной на этом интервале.

Если в этой точке существует касательная, то ее называют *точкой перегиба.*

На рисунке I.2.3.1 показан график функции, на интервале $]a; b[$ выпуклый и на $]b; c[$ вогнутый.

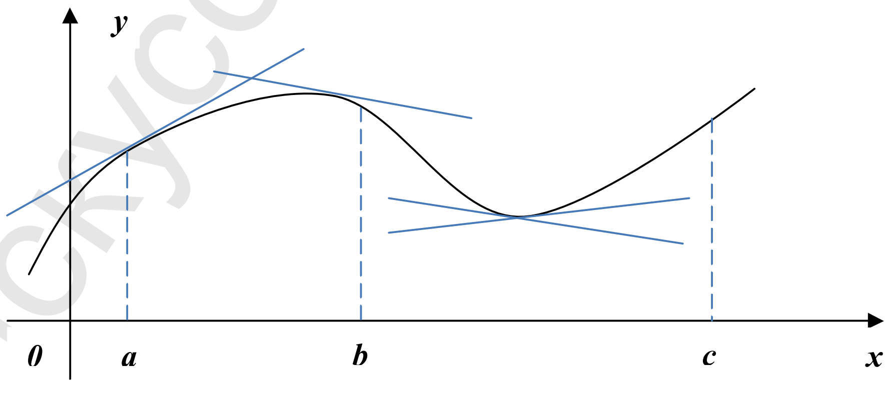


Рисунок I.2.3.1. Выпуклый и вогнутый график.

НИИ Искусственный интеллект

Область называется выпуклой, если для любых двух ее точек соединяющий их отрезок целиком лежит в области. Область называется ограниченной, если ее можно заключить в некоторый круг. Неограниченную область невозможно заключить внутрь круга (например, область между двумя параллельными линиями). Примеры выпуклых ограниченных и неограниченных областей представлены на рисунке I.2.3.2.

Выпуклые области

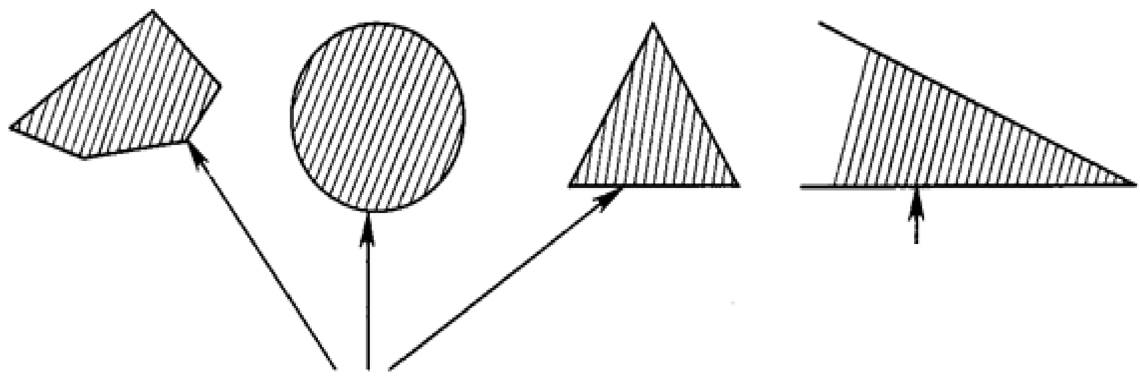


Рисунок I.2.3.2 Выпуклые ограниченные и неограниченные.

Примеры 1.2.3:

1. Не каждая функция является четной или нечетной, например, функция $y = f(x)$ = x+1 ни четная, ни нечетная, поскольку

$$f(-x) = -x+1 \neq f(x) \text{ и } f(-x) = -x+1 \neq -f(x).$$

2. Некоторые основные элементарные функции являются четными, а другие нечетными: $cos\ x$ – четная функция, $sin\ x$ – нечетная функция.

I.2.3. Задания:

1. Дайте определение ограниченной и неограниченной функций.

2. Определить четные функции среди следующих:
$y = x^2$, $y = x^5$.

3. Определите условия кусочно-монотонности функции.

4. Построить ступенчатую функцию со значениями -1, 0, 1.

5. Определите условия периодичности функции.

НИИ искусственного интеллекта

I.2.3. Вопросы:

1. Какие функции называются ступенчатыми?
2. В чем разница между убывающей и возрастающей функциями?
3. В чем разница между четной и нечетной функциями?
4. Как определяется периодичность функции?
5. Как определяется выпуклость и вогнутость графика функции?

НИИ Искусственный интеллект

II.1. ЭЛЕМЕНТЫ ТЕОРИИ НЕЙРОННЫХ СЕТЕЙ

II.1. Основные понятия

II.1.1. Естественные нейроны

В этом параграфе рассматриваются естественные нейроны, предлагаются задания и формулируются вопросы. При подготовке учебного материала данного параграфа были использованы следующие источники [14, 17, 22, 27, 28, 33].

Можно считать, что изучение человеческого мозга началось с тех времен, когда людей начало интересовать их собственное мышление. Размышление о себе самом является, возможно, отличительной чертой мозга человека. Имеется множество размышлений о природе мышления, простирающихся от духовных до анатомических. Обсуждение этого вопроса, протекавшее в горячих спорах философов и теологов с физиологами и анатомами, принесло мало пользы из-за трудности изучения предмета. Те, кто опирался на самоанализ и размышление, пришли к выводам, не отвечающим уровню строгости точных наук. Экспериментаторы же нашли, что мозг труден для наблюдения и ставит в тупик своей организацией. Говоря другими словами, мощные методы научного исследования, изменившие наш взгляд на реальный мир, оказались бессильными в понимании самого человека.

Мозг человека является самой сложной из известных на сегодняшний день систем обработки информации и состоит, примерно, из 100 миллиардов (10^{11}) связанных между собой *нейронов*. Каждый нейрон имеет в среднем 10 тысяч (10^{4}) связей, что порождает 10^{15} взаимосвязей. При этом мозг человека чрезвычайно надежен, несмотря на то, что ежедневно погибает большое количество нейронов, мозг продолжает функционировать. Обработка огромных объемов информации осуществляется мозгом очень быстро, за доли секунды, с учетом того, что собственно нейрон является медленнодействующим элементом со временем реакции не менее нескольких миллисекунд. На данный момент

НИИ искусственный интеллект

неясно, как мозгу удается получить столь впечатляющее сочетание надежности и быстродействия.

Современной наукой довольно хорошо изучены структура и функции отдельных нейронов, имеются данные об организации внутренних и внешних связей между нейронами, некоторых структурных образований мозга, и совсем мало известно об участии различных структур в процессах переработки информации.

Каждый нейрон имеет тело нейрона - *сома*, множество входных связей – *дендриты*, единственную выходящую связь - *аксон*, которая на конце также разветвляется, и контакты – *синапсы* для образования связей аксонов с дендритами других нейронов.

Структура естественного нейрона показана на рисунке II.1.1.

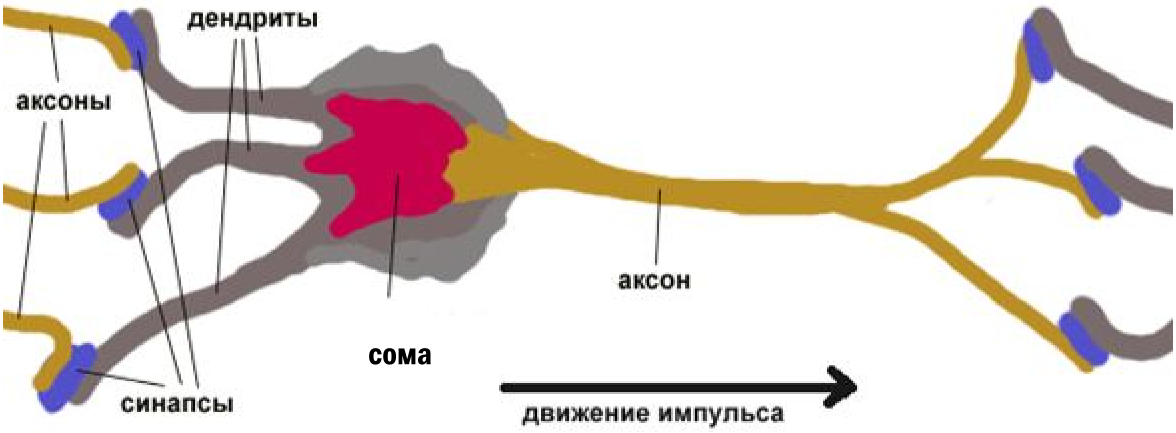


Рисунок II.1.1. Структура естественного нейрона.

Входную информацию нейрон получает через свои дендриты от аксонов других нейронов. Дендриты сильно ветвятся, пронизывая сравнительно большое пространство в окрестности нейрона. Длина дендритов может достигать 1 мм.

Полученная информация преобразуется сомой, чтобы сгенерировать выходную информацию. Сома имеет поперечный размер в несколько десятков микрон. Выходная информация передается другим нейронам через аксон, разветвляющийся на конце на многие синапсы. Длина аксона может достигать сотен миллиметров. На соме и на дендритах располагаются окончания аксонов, идущих от других нервных клеток, причем каждое такое

НИИ Искусственный интеллект

окончание имеет вид утолщения, называемого синаптической бляшкой, или синапсом. Поперечные размеры синапса, как правило, не превышают нескольких микрон, чаще всего эти размеры составляют около 1 мкм. Синапс является элементарной структурой и функциональным узлом между двумя нейронами. Передача через синапс почти всегда однонаправленная. Различают пресинаптические и постсинаптические клетки – по направлению передачи импульса.

Когда импульс достигает синаптического окончания, высвобождаются определенные химические вещества, называемые нейротрансмиттерами. Нейротрансмиттеры диффундируют через синаптическую щель, возбуждая или затормаживая в зависимости от типа синапса, способность нейрона-приемника генерировать электрические импульсы.

Отдельный нейрон не является элементарной единицей обработки информации, а выполняет функции нервного центра. Дендриты и аксоны могут вступать в связи с участками мембран других нейронов, образуя сети. Эти сети и служат системами обработки информации.

Алгоритм работы биологического (естественного) нейрона заключается в следующем. Проходя через синапс, электрический сигнал меняет свою амплитуду: увеличивает или уменьшает. Это можно интерпретировать как умножение амплитуды сигнала на весовой (синаптический) коэффициент. Взвешенные в дендритном дереве входные сигналы суммируются в теле клетки, и затем на аксонном выходе генерируется выходной импульс (спайк) или пачка импульсов. Выходной сигнал проходит по ветви аксона и достигает синапсов, которые соединяют аксон с дендритными деревьями других нейронов. Через синапсы сигнал трансформируется в новый входной сигнал для смежных нейронов. Этот сигнал может быть положительным или отрицательным (возбуждающим или тормозящим) в зависимости от вида синапса. Величина сигнала, генерируемого на выходе синапса, определяется

НИИ искусственный интеллект

синаптическим коэффициентом (весом синапса), который может меняться в процессе функционирования синапса.

Нейроны можно разбить на три большие группы: рецепторные, эффекторные и промежуточные. Рецепторные нейроны обеспечивают ввод в мозг сенсорной информации. Они трансформируют сигналы, поступающие на органы чувств (оптические сигналы в сетчатке глаза, акустические в ушной улитке или обонятельные в хеморецепторах носа), в последовательность электрических импульсов своих аксонов. Эффекторные нейроны передают приходящие на них сигналы исполнительным органам. На конце их аксонов имеются специальные синаптические соединения с исполнительными органами, например, мышцами, где возбуждение нейронов трансформируется в сокращения мышц. Промежуточные нейроны осуществляют обработку информации, получаемой от рецепторов, и формируют управляющие сигналы для эффекторов. Именно они образуют центральную нервную систему.

Входные сигналы дендритного дерева взвешиваются и суммируются в соме, причем если результат не превышает некоторого порога, то выходной сигнал не формируется вовсе – нейрон «не срабатывает». Выходной сигнал проходит по ветвям аксона и достигает синапсов, которые соединяют аксоны с дендритными деревьями других нейронов. Через синапсы сигнал трансформируется в новый входной сигнал для смежных нейронов. Этот входной сигнал может быть положительным и отрицательным, т.е. возбуждающим или тормозящим в зависимости от вида синапсов. Величина входного сигнала, генерируемого синапсом, может быть различной даже при одинаковой величине сигнала, приходящего в синапс. Эти различия определяются эффективностью или весом синапса, причем синаптический вес может изменяться в процессе функционирования синапса. Многие ученые считают такое изменение нейрофизиологическим коррелятом, т.е. следом памяти.

НИИ Искусственный интеллект

В таком случае роль механизмов молекулярной памяти заключается в долговременном закреплении этих следов.

Функционально работа естественного нейрона описывается так:

- принятые от аксонов других нейронов входные сигналы проходят через синапсы и изменяются пропорционально весам синапсов;

- поступившие к соме одновременно по нескольким дендритам измененные входные сигналы суммируются;

- если суммарный импульс превышает некоторый порог, то нейрон возбуждается и формирует выходной сигнал, который отводится аксоном из сомы и разветвляется;

- ветви выходного сигнала передаются через синапсы к дендритам других нейронов.

При этом вес каждой из синаптических связей может изменяться со временем, следовательно, может меняться и поведение нейрона.

В целом естественные (биологические) нейронные сети обладают следующими свойствами:

- распределенное представление информации и вычисления;
- параллельность обработки информации.
- изменяющиеся по весу связи между нейронами;
- способность к обучению и способность к обобщению;
- адаптивность;
- свойство контекстуальной обработки информации;
- толерантность к ошибкам.

Задания II.1.1:

1. Нарисуйте состав и структуру естественного нейрона.
2. Опишите функциональную работу естественного нейрона.
3. Опишите функцию рецепторных нейронов.
4. Опишите функцию эффекторных нейронов.
5. Опишите функцию промежуточных нейронов.

Вопросы II.1.1:

1. Сколько нейронов в мозге человека?

НИИ искусственный интеллект

2. Сколько связей имеет каждый нейрон?
3. Сколько всего связей имеется в человеческом мозге?
4. Посредством чего взаимодействуют нейроны?
5. Посредством чего принимает нейрон информацию?
6. Какими свойствами обладают естественные нейроны?

НИИ Искусственный интеллект

II.1.2. Естественные нейронные сети

В этом параграфе рассматриваются естественные нейронные сети, предлагаются задания и формулируются вопросы. При подготовке учебного материала данного параграфа были использованы следующие источники [14, 17, 22, 27, 28, 33].

Нейроны, соединяясь с друг с другом через синапсы, образуют единую сложную *нейронную сеть.* Струкутра естественной нейронной сети представлена на рисунках II.1.2.1.

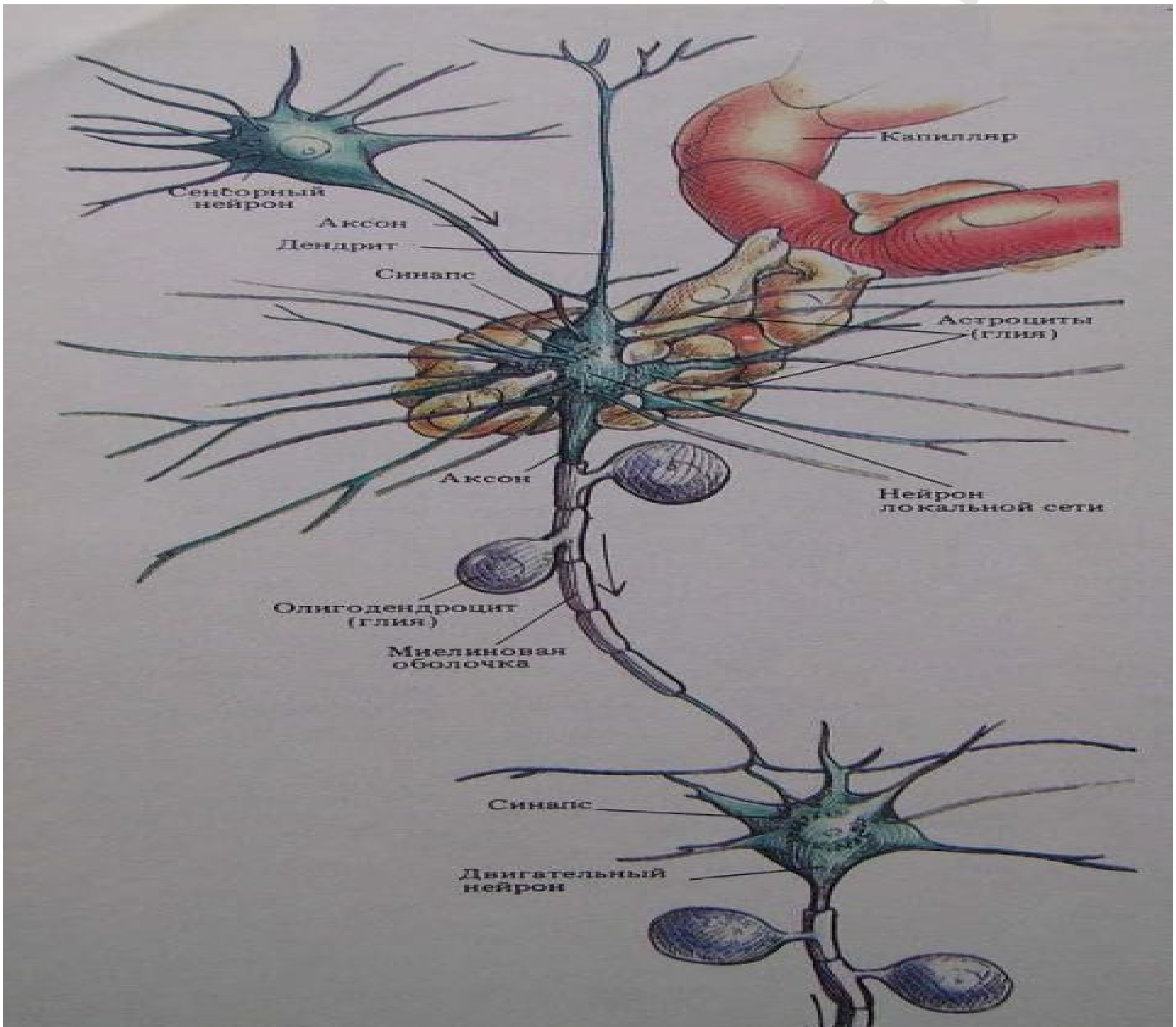


Рисунок II.1.2.1. Струкутра естественной нейронной сети.

Нейроны в живых организмах представляют собой особый вид клеток, обладающих электрической активностью, основное назначение которых заключается в оперативном управлении организмом.

Нейроны в нейронной сети взаимодействуют посредством серий электрохимических импульсов, которые появляются в результате некоторого нейрофизиологического процесса в мозге. Каждый импульс представляет собой частотный сигнал с частотой от нескольких единиц до сотен герц. Если учесть, что любой нейрофизиологический процесс активизирует сразу множество связанных между собой нейронов, то можно представить себе то количество информации или сигналов, которое возникает в мозгу человека. Картина электрохимических импульсов в нейронной сети показана на рисунке II.1.2.2.

Рисунок II.1.2.2. Электрохимические импульсы в нейронной сети.

Как уже упоминалось ранее, электрохимические импульсы длятся несколько миллисекунд, каждый импульс представляет собой частотный сигнал с частотой от нескольких единиц до сотен герц. По сравнению с современными компьютерами это невообразимо медленно, но в тоже время человеческий мозг гораздо быстрее машины способен обрабатывать аналоговую информацию, например: узнавать изображения, чувствовать вкус, узнавать звуки, читать чужой почерк, оперировать качественными параметрами. Все вышеперечисленное реализуется посредством сети нейронов, соединенных между собой синапсами. Другими словами, мозг - это система из параллельных процессоров, работающая гораздо эффективнее, чем последовательные вычисления.

Сотни миллиардов нейронов, каждый из которых соединен с сотнями или тысячами других, образуют систему, далеко превосходящую самые мощные суперкомпьютеры. Кроме того, обычно мозг человека задействован только на 2-3% от своей возможности. Благодаря такой избыточности, мозг человека обладает огромным запасом прочности, позволяющим ему работать, несмотря на серьезные повреждения и утраты. Подобной способности лишены современные компьютеры.

Следует отметить, что в современных компьютерах технология последовательных вычислений подошла к пределу своих технических возможностей, и в настоящее время остро стоит проблема развития методов параллельного программирования и создания параллельных компьютеров. В этой связи нейронные сети являются очередным шагом в этом направлении.

Однако отличие компьютеров (моделируемых нейронных сетей) от естественных нейронных сетей заключается в том, что моделируемые сети строго упорядочены и их элементы обладают фиксированным количеством связей, в то время как естественные - хаотичны, динамичны как по структуре, так и по количеству связей между нейронами. Естественные нейронные сети самоорганизующиеся, что делает их существенно эффективнее чем моделируемые, кроме того им не нужно задавать целеполагание, последнее у них определяется гомеостазом клетки и всей колонии.

Задания II.1.2:

1. Опишите способ образования нейронных сетей.
2. Объясните причину наличия огромного запаса прочности человеческого мозга.
3. Укажите проблему в развитии компьютерной технологии.

Вопросы II.1.2:

1. Посредством чего взаимодействует нейроны в естественной нейронной сети?

НИИ искусственный интеллект

2. Что из себя представляет каждый электрохимический импульс?

3. В чем состоит перспектива развития нейронных сетей?

НИИ Искусственный интеллект

II.1.3. История искусственных нейронов и нейронных сетей

В этом параграфе предложена краткая история появления и развития искусственных нейронов и нейронных сетей, приведены задания и сформулированы вопросы. При подготовке учебного материала данного параграфа были использованы следующие источники [14, 17, 22, 27, 28, 33].

В 1943 году нейрофизиолог У.Мак-Каллок и математик У.Питтс в статье «Логическое исчисление идей, относящихся к нервной активности» ввели понятие искусственной нейронной сети и предложили формальную модель искусственного нейрона. В 1948 году Н. Винер вместе с соратниками опубликовал работу о кибернетике, в которой предложили идею о представлении сложных биологических процессов математическими моделями. В 1949 году Д. Хебб в работе «Организация поведения» описал основные принципы и первый алгоритм обучения нейронов.

В 1950-е гг. появляются программные модели искусственных нейронных сетей. Первые работы проведены Н. Рочестером из исследовательской лаборатории IBM. Хотя дальнейшие реализации были успешными, эта модель потерпела неудачу, поскольку бурный рост традиционных вычислений оставил в тени нейронные исследования.

В 1956 году Дартмутский исследовательский институт искусственного интеллекта обеспечил подъем исследований в области искусственного интеллекта, в частности, нейронных сетей. Стимулирование исследований искусственного интеллекта разделилось на два направления: промышленное применение систем искусственного интеллекта (экспертные системы) и моделирование мозга.

В 1957 году Ф. Розенблатт предложил компьютерную модель мозга, называемую персептрон (от латинского perceptio - восприятие), и в том же году в Корнельской Лаборатории Аэронавтики успешно было завершено моделирование работы персептрона на компьютере IBM 704. Персептрон передавал сигналы от фотоэлементов, представляющих собой сенсорное поле, в блоки электромеханических ячеек памяти. Эти ячейки соединялись между собой случайным образом в соответствии с принципами коннективизма.

В 1958 году Ф. Розенблатт опубликовал статью «Персептрон: Вероятная модель хранения и организации информации в головном мозге», где вводит однослойный персептрон и демонстрирует его способность решать задачи классификации. Персептрон обрёл популярность и его начали использовать для распознавания образов, прогнозирования погоды и т. д. В этом же году Джон фон Нейман предложил имитацию простых функций нейронов с использованием вакуумных трубок.

В 1959-60 годы Б. Уидроу и М. Хофф разработали модели ADALINE и MADALINE (Множественные Адаптивные Линейные Элементы). ADALINE (адаптивный сумматор) сначала использовался для задач предсказания и адаптивного управления, затем он стал стандартным элементом многих систем обработки сигналов. MADALINE действовал, как адаптивный фильтр, устраняющий эхо на телефонных линиях, и до сих пор находится в коммерческом использовании.

В 1960 году в Корнельском университете был продемонстрирован первый нейрокомпьютер – «Марк-1», который мог распознавать некоторые буквы английского алфавита.

В 1962 году Ф. Розенблатт подробно описал свои теории и предположения относительно процессов восприятия и персептронов в книге «Принципы нейродинамики: Персептроны и теория механизмов мозга». Он рассматривает уже не только готовые модели персептрона с одним скрытым слоем, но и многослойные персептроны с перекрёстными и обратными связями, доказывает теорему сходимости персептрона. Тогда однослойный персептрон, реализованный аппаратно, считался классической нейронной сетью и использовался для классификации входных сигналов в один из двух классов. Впервые с помощью персептрона были решены задачи классификации и распознавания букв английского языка. Для этого был разработан специальный

НИ... КТ

итерационный метод обучения на основании проб и ошибок, напоминающий процесс обучения человека методом коррекции ошибки. Кроме того, при распознании той или иной буквы персептрон мог выделять характерные особенности буквы, статистически чаще встречающиеся, чем малозначимые отличия в индивидуальных случаях. Тем самым персептрон был способен обобщать буквы, написанные различным образом (почерком), в один обобщённый образ.

В 1963 году в Институте проблем передачи информации АН СССР А. П. Петровым проводится подробное исследованием задач, «трудных» для персептрона. Эта пионерская работа в области моделирования искусственных нейронных сетей (ИНС) в СССР послужила отправной точкой для комплекса идей М. М. Бонгарда – «как сравнительно небольшой переделкой персептрона можно исправить его недостатки».

Чрезмерное ожидание от нейронных сетей, процветающее в академическом и техническом мире, заразило общую литературу этого времени. Опасение, что эффект "мыслящей машины", отразится на человеке все время подогревалось писателями. Однако, возможности персептрона были весьма ограниченными. Машина не могла надежно распознавать частично закрытые буквы, а также буквы иного размера, расположенные со смещением или поворотом, нежели те, которые использовались на этапе ее обучения.

В 1969 году М. Минский и С. Пейперт доказали, что персептроны теоретически неспособны решить многие простые задачи, в том числе реализовать бинарную функцию «Исключающее ИЛИ – eXclusive OR (XOR)», проблему «чётности» или «один в блоке», связанные с инвариантностью представлений. Данный факт и опасения, объединенные с невыполненными обещаниями, вызвали множество разочарований специалистов, подвергших критике исследования нейронных сетей. В результате интерес к разработкам нейронных сетей пошел на спад и было прекращено финансирование.

В 1972 году Т. Кохонен и Дж. Андерсен независимо предлагают новый тип нейронных сетей, способных функционировать в качестве памяти.

В 1973 году Б. В. Хакимов предлагает нелинейную модель с синапсами на основе сплайнов и внедряет её для решения задач в медицине, геологии, экологии.

В 1974 году Пол Дж. Вербос и А. И. Галушкин одновременно изобретают алгоритм обратного распространения ошибки для обучения многослойных персептронов.

В 1975 году Фукусима представляет когнитрон, являющегося самоорганизующейся сетью, предназначенной для инвариантного распознавания образов. Тем не менее, это было достигнуто только при помощи запоминания практически всех состояний образа.

В 1982 году к возрождению интереса привело несколько событий. Дж. Хопфилд представил статью в национальную Академию Наук США. Он показал, что нейронная сеть с обратными связями может представлять собой систему, минимизирующую энергию, т.е. показал возможности моделирования нейронных сетей на принципе новой архитектуры, называемой сетью Хопфилда. В то же время в Киото (Япония) состоялась Объединенная американо-японская конференция по нейронным сетям, на которой сети были объявлены достижением пятой генерации. Американские периодические издания подхватили данную историю, акцентируя на возможности отставания США в данной области, что привело к росту финансирования исследования нейросетей. Затем Кохоненом была представлена модель сети, называемая нейронной сетью Кохонена (самоорганизующаяся карта Кохонена), которая обучается без учителя и решает задачи кластеризации, визуализации данных и другие задачи анализа данных.

С 1985 года Американский Институт Физики начал ежегодные встречи - "Нейронные сети для вычислений".

В 1986 году Д. Румельхартом, Дж. Хинтоном и Р. Вильямсом, а также независимо и одновременно С. И. Барцевым и

НИИ Интеллект

В. А. Охониным, был переоткрыт и существенно развит метод обратного распространения ошибки. Начался взрыв интереса к обучаемым нейронным сетям.

В 1989 году на встрече "Нейронные сети для обороны" Б. Уидроу сообщил аудитории о начале четвертой мировой войны, где полем боя являются мировые рынки и производства.

В 2007 году Джеффри Хинтоном в университете Торонто созданы алгоритмы глубокого обучения многослойных нейронных сетей. Успех обусловлен тем, что Хинтон при обучении нижних слоев сети использовал ограниченную машину Больцмана (RBM – Restricted Boltzmann Machine). Глубокое обучение по Хинтону – это очень медленный процесс. Необходимо использовать много примеров распознаваемых образов (например, множество лиц людей на разных фонах). После обучения получается готовое быстро работающее приложение, способное решать конкретную задачу,например, осуществлять поиск лиц на изображении. Функция поиска лиц людей на сегодняшний день стала стандартной и встроена во все современные цифровые фотоаппараты. Технология глубокого обучения активно используется интернет-поисковиками при классификации картинок по содержащимся в них образам. Искусственные нейронные сети, применяемые при распознавании, могут иметь до 9 слоев нейронов. Их обучение ведётся на миллионах изображений с отыскиваемым образом.

Сегодня обсуждение нейронных сетей происходит везде. Перспектива их использования кажется довольно яркой, в свете решения нетрадиционных проблем и является ключом к целой технологии. На данное время большинство разработок нейронных сетей является работающими, но существуют процессорные ограничения. Исследования направлены на программные и аппаратные реализации нейросетей. Компании работают над созданием трех типов нейрочипов: цифровых, аналоговых и оптических, которые обещают быть трендом близкого будущего.

НИИ искусственный интеллект

В целом понимание функционирования нейрона и картины его связей позволило исследователям создать математические модели для проверки своих теорий. Эксперименты теперь могут проводиться на цифровых компьютерах без привлечения человека или животных, что решает многие практические и морально-этические проблемы. В первых же работах выяснилось, что эти модели не только повторяют функции мозга, но и способны выполнять функции, имеющие свою собственную ценность. Поэтому возникли и остаются в настоящее время две взаимно обогащающие друг друга цели нейронного моделирования:

– понять функционирование нервной системы человека на уровне физиологии и психологии;

– создать вычислительные системы (искусственные нейронные сети), выполняющие функции, сходные с функциями мозга.

Задания II.1.3:

1. Назовите задачи, которые решались с помощью персептрона.
2. Назовите авторов, которые доказали ограниченность персептронов, которые не могли реализовать функцию XOR.
3. Назовите автора, который предложил когнитрон для решения задач инвариантного распознавания образов.
4. Назовите авторов, которые независимо предложили тип нейронных сетей, способных функционировать в качестве памяти.
5. Назовите автора, который создал алгоритмы глубокого обучения многослойных нейронных сетей.

Вопросы II.1.3:

1. Как действует модели множественных адаптивных линейных элементов?
2. По какой причине и до какого момента времени пропал интерес к искусственным нейронным сетям?
3. Что повлияло на возрождение интереса к нейронным сетям?
4. Кто изобрел алгоритм обратного распространения ошибки для обучения многослойных персептронов?
5. Когда,кем переоткрыт и существенно развит метод обратного распространения ошибки?

НИИ искусственный интеллект

6. Как работают алгоритмы глубокого обучения многослойных нейронных сетей?

НИИ Искусственный интеллект

II.1.4. Формальная модель искусственного нейрона

В этом параграфе будут рассмотрены формальная (математическая) модель, структурная схема и алгоритм работы первого искусственного нейрона, предложены задания и сформулированы вопросы. При подготовке учебного материала данного параграфа были использованы следующие источники [1-9].

Как следует из предыдущего параграфа, первую модель искусственного нейрона на электрических схемах, имитирующая работу естественного (биологического) нейрона, предложили ученые Уоррен Мак-Каллок (Warren McCulloch) и Уолтер Питтс (Walter Pitts).

Искусственный нейрон представляет собой абстрактный компьютер, состоящий из трех блоков (*умножитель*, *сумматор*, *преобразователь*), и позволят производить переработку входных сигналов на выходные сигналы с учетом значений начального состояния, соответствующих весов синаптических связей и функции активации.

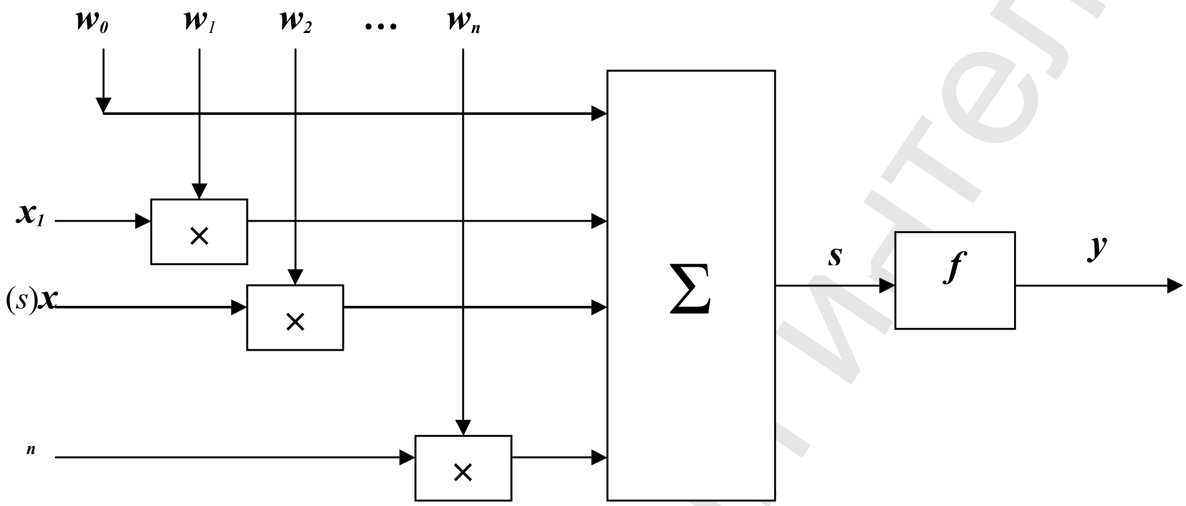
В общем случае искусственный нейрон имеет $n \geq 1$ входов и к синапсам этих входов поступает вектор входных сигналов $X = (x_1, x_2, \dots, x_n)$, после прохождения синапсов сигналы изменяются пропорционально вектору весов синапсов $W = (w_1, w_2, \dots, w_n)$, т.е. получается $W \cdot X = w_1 \cdot x_1, w_2 \cdot x_2, \dots, w_n \cdot x_n$. Измененные входные сигналы через дендриты поступают к соме, где получается суммарный импульс $s = w_1 \cdot x_1 + w_2 \cdot x_2 + \dots + w_n \cdot x_n$, называемый *уровнем активации*. Активацию вызывает определенная функция f, называемая *функцией активации.* После действия функции активации уровень активации сравнивается с некоторым пороговым значением w_0. Если значение уровня активации превышает этот порог, то нейрон возбуждается и формирует выходной сигнал y.

В общем случае значения входного сигнала, веса и смещения являются действительными числами. Значение выходного сигнала

НИИ искусственный интеллект

определяется видом функции активации и может быть, как действительным, так и целым числом.

Структурная схема искусственного нейрона Мак-Каллока и Питтса показана на рисунке II.1.4.



I.1

Искусственный нейрон Мак-Каллока и Питтса действует по следующему алгоритму:

1. Перед началом работы на блок сумматора подается значение сигнала начального состояния (смещения - bias) w_0;

2. В блоке умножителя значение каждого входного сигнала x_i умножается на соответствующее значение веса синапса w_i, $i= 1,2,\dots,n$;

3. В сумматоре все значения взвешенных входных сигналов (произведения значений входных сигналов и весов) и значение сигнала начального состояния суммируются, определяя уровень активации нейрона $\sum_{i=1}^{n} w_i x_i + w_0 = W \cdot X + w_0 = s$;

4. Взвешенная сумма (результат суммирования) s подается на блок функционального преобразователя f, где вырабатывается значения выходного сигнала с учетом заданного порогового значения $y = f(s)$, если имеется функция активации, в противном случае выходной сигнал будет равен взвешенной сумме s, полученной на предыдущем шаге.

НИИ искусственный интеллект

Тогда математическую модель искусственного нейрона Мак-Каллок и Питтса можно представить в виде

$$y = f\left(\sum_{i=1}^{n} w_i x_i + w_0\right) = f(W \cdot X + w_0) = f(s),$$

где x_i – входной сигнал, w_i – вес синапса, w_0 – сигнал начального состояния, *s* – сумма, *y* – выходной сигнал, *f* – функция активации, *n* – число входов (весов), при *i*-тая синаптическая связь является *возбуждающей* (*тормозящей*), если значение веса w_i положительное (отрицательное), *i=1*, *2*, ..., *n*.

Для решения с помощью искусственного нейрона реальных задач необходимо нормализовать значения входных данных. *Нормализация* входных данных - это процесс, при котором все входные данные проходят процесс "выравнивания", т.е. приведения к интервалу [0,1] или [-1,1].

Если не провести нормализацию, то входные данные будут оказывать дополнительное влияние на нейрон, что приведет к неверным решениям. Другими словами, как можно сравнивать величины разных порядков?

В общем виде формула нормализации выглядит так:

$$x_{norm} = \frac{(x - x_{min}) \cdot (d_2 - d_1)}{x_{max} - x_{min}} + d_1 \qquad \text{(II.1.4)}$$

где *x* - значение, подлежащее нормализации, $]x_{\min}, x_{\max}[$ - интервал значений *x*, $]d_1, d_2[$ - интервал, к которому будет приведено значение *x*.

Поясним сказанное на примере. Пусть есть *n* входных данных из интервала [0,10], тогда $x_{\max} = 10$, $x_{\min} = 0$.

Данные будем приводить к интервалу [0,1], тогда $d_1 = 0, d_2 = 1$. Подставив все значения в формулу (II.1.4), можно вычислить для любого *x* из *n* входных данных нормализованные значения x_{norm}.

НИУ Исследовательский интеллект

Сначала мы указываем крайние значения для выходного значения, после получаем минимальное и максимальное значения индикатора (копирование данных с индикатора пропущено, но к примеру, там может быть 10 последних значений). Последним шагом произведем в цикле нормализацию каждого входного элемента (значения индикатора на различных барах) и сохраним в массив для дальнейшей работы с ним.

В модели Мак-Каллок-Питтса отсутствуют временные задержки входных сигналов, поэтому значение s определяет полное внешнее возбуждение, воспринятое нейроном. Отклик нейрона далее описывается по принципу "все или ничего", т. е. переменная подвергается нелинейному пороговому преобразованию, при котором выход (состояние активации нейрона) y устанавливается равным единице, если взвешенная сумма больше порогового значения s и выход y равняется нулю в обратном случае. Значение порога γ (часто полагаемое равным нулю) также хранится в локальной памяти. Таким образом, в модели Мак-Каллок-Питтса используется пороговая функция активации:

$$y=f(s)=\begin{cases}1, & \text{если} \quad s>0 \\ 0, & \text{если} \quad s\leq 0\end{cases}$$

Свойства функции активации оказывают определяющее влияние на выбор способа обучения нейрона, заключающегося в подборе весовых коэффициентов. Выделяют два подхода: *обучение с учителем* (*supervised learning*) и *обучение без учителя* (*unsupervised learning*).

При обучении с учителем предполагается, что, помимо входных сигналов x_i, известны также и ожидаемые выходные сигналы нейрона d_i. Подбор весов организуется так, чтобы фактические выходные сигналы нейрона $\boldsymbol{y}_i$ принимали бы значения, как можно более близкие к ожидаемым значениям d_i.

Если такой подход невозможен, то применяют стратегию обучения без учителя. Подбор весовых коэффициентов в этом случае проводится на основании, либо конкуренции нейронов между собой, либо с учетом корреляции обучающих и выходных сигналов. При этом на этапе адаптации нейрона невозможно спрогнозировать его выходные сигналы.

Структура, представленная на рисунке II.1.4. является *стандартной моделью искусственного нейрона*. Большинство математических моделей созданы в соответствии с принципами функционирования биологических нейронов и основываются на стандартной модели Мак-Каллок-Питтса, например, *персептрон*, *сигмоидальный нейрон*.

Позже разработчики искусственных нейронов предложили несколько нестандартные детерминированные модели искусственного нейрона (*паде-нейрон*, *нейрон с квадратичным сумматором*, *адаптивный линейный нейрон*, *сигма-пи нейрон*, *WTA*, *Хебба*, *Гроссберга*, *нейрон со счетчиком совпадений и др*.), а также стохастическую (вероятностную) модель нейрона.

Кроме того, заметим, что если значения входного сигнала, веса синапса и смещения являются:

1) произвольными вещественными числами и к ним применяются вещественная арифметика, то говорим о *вещественных нейронах*;

2) логическими значениями 0 или 1 и к ним применяются логические операции, то речь идет о *логических нейронах*;

3) числами в интервале [0, 1] и к ним применяются операции нечеткой (fuzzy) логики, то имеем дело с *нечеткими нейронами.*

Для иллюстрации рассмотрим конкретный пример. Пусть нейрон получает на вход сигнал, уровни которого равны X. Соответствующие веса связей равны W. Вычислить выходное значение нейрона j для полулинейного нейрона $\alpha = 3$.

Дано:

$\boldsymbol{X} = (1{,}3;\ 2{,}3;\ -3)$; $\boldsymbol{W} = (0{,}2;\ 0{,}6;\ -0{,}02)$, w_0=-1,2.

Функция активации: $y = f(s) = \begin{cases} \alpha s, & \text{если} \quad s > 0 \\ 0, & \text{если} \quad s \le 0 \end{cases}$

Решение:

НИИ искусственный интеллект

Функционирование нейрона можно описать следующей парой уравнений

$$s=\sum_{i=1}^{n} w_i x_i \ , \ y=f(s+w_0) \ ,$$

где $X=(x_1, x_2, ..., x_n)$ – входные сигналы; $W=(w_1, w_2, ..., w_n)$ – веса синаптических связей; w_0 – порог; s – взвешенная сумма; $f(s)$ – функция активации; y – выходной сигнал нейрона.

Имеем:

$$x_1=-1{,}3; x_2=2{,}3; x_3=-3; w_1=0{,}2; w_2=0{,}6; w_3=-0{,}02; w_0=b=-1{,}2.$$

$$s=\sum_{i=1}^{n} w_i x_i = x_1 w_1 + x_2 w_2 + x_3 w_3 = -1{,}3\cdot 0{,}2 + 2{,}3\cdot 0{,}6 + (-3)\cdot(-0{,}02) = 1{,}18$$

$$y=f(s+b)=f(1{,}18\text{-}1{,}2)=f(\text{-}0{,}02)=0 \ .$$

Ответ: $y = 0$

II.1.4. Задания:

1. Укажите блоки, из которых состоит искусственный нейрон.
2. Опишите работу искусственного нейрона.
3. Опишите алгоритм работы нейрона Мак-Каллок и Питтса.
4. Напишите математическую модель нейрона Мак-Каллок и Питтса.
5. Напишите формулу нормализации входных данных.
6. Напишите формулу функционирования нейрона.

II.1.4. Вопросы:

1. Какие значения могут иметь входные сигналы, веса и смещения?
2. Что подается перед началом работы на блок сумматора?
3. В каком случае синаптическая связь будет возбуждающей или тормозящей?
4. Какая роль выполняет функция активации?
5. К чему приведет подача ненормализованных входных данные?
6. Какие нестандартные детерминированные модели искусственного нейрона были предложены разработчиками позже?

НИИ искусственный интеллект

II.1.5. Классификация искусственных нейронов

В этом параграфе рассматривается классификация искусственных нейронов в зависимости от различных критериев, приводятся примеры, предложены задания, сформулированы вопросы. При подготовке учебного материала данного параграфа были использованы следующие источники [1-9].

Искусственные нейроны классифицируют в зависимости от их положения в сети, от типов оперируемых данных, от способов описания модели нейрона и от способов их аппаратной реализации.

Искусственные нейроны в зависимости от их положения (роли) в сети подразделяются на *входные нейроны*, *промежуточные нейроны* и *выходные нейроны*.

Входные (рецепторные) нейроны принимают входной вектор, кодирующий исходный сигнал и не выполняют операций над ним, а просто передают закодированный исходный сигнал на выход, возможно, усилив или ослабив его.

Выходные (эффекторные) нейроны представляют выходные сигналы сети. В выходных нейронах могут производиться какие-либо операции над выходными сигналами.

Промежуточные нейроны выполняют основные операции над исходным и промежуточным сигналом.

Искусственные нейроны по типу обрабатываемых данных подразделяются на *числовые нейроны* и *логические нейроны*.

Числовые нейроны оперируют с числовыми данными, представленными в форме действительных чисел. Допускается использование целых чисел при решении некоторых задач распознавания образов.

Логические нейроны оперируют с логическими данными. Причем, если значением логических данных являются только 0 или 1, то речь идет о *двоичном логическом нейроне*, использующем операции классической (булевой) логики. В случае, когда значения логических данных находятся в интервале [0,1], то мы имеем дело с

НИИ искусственный интеллект

нечетким логическим нейроном, который использует операции нечеткой логики.

Искусственные нейроны в зависимости от их аппаратной реализации подразделяются на *цифровые* и *аналоговые нейроны*.

Цифровые нейроны реализуются микросхемами средней и высокой степени интеграции, при этом на кристалле реализуется непосредственно нейронная сеть, т. е. отдельные нейроны не выделяются как самостоятельные единицы, т.к. цифровую модель разрабатывают для решения конкретной задачи, которую невозможно решить с помощью одного нейрона.

Аналоговые нейроны позволяют воспроизводить пространственное и временное суммированием возбуждающих сигналов, свойства абсолютной и относительной рефрактерности (невосприимчивости), процессы переработки информации в отдельном нейроне и при взаимодействии нейронов. Аналоговую модель рассматривают как устройство, состояние которого может непрерывно меняться от полного покоя до некоторого максимального уровня возбуждения.

Искусственные нейроны в зависимости от способов описания их модели подразделяются на *формальные модели*, *физиологические модели* и *феноменологические модели нейрона*.

Формальные модели нейрона отличаются хорошо разработанным математическим аппаратом, но игнорируют многие свойства своего биологического прототипа.

Физиологические модели нейрона отличаются количественным описанием поведения нейрона, порождены из экспериментов над биологическими нейронами.

Феноменологические модели нейронов не имеют строгой математической и опытной базы.

Формальные модели нейронов, в свою очередь, также могут быть разделены в зависимости от изменяемости весов входов, от вида функции активации и от вероятности значения выходного сигнала.

НИИ Искусственный интеллект

По изменяемости весов нейроны подразделяются на *нейроны с неизменными весами входов*, *нейроны с управляемыми весами входов*, *нейроны с перестраивающимися весами в зависимости от входного по данному каналу сигнала*, *нейроны с изменяющимися весами под влиянием выходного сигнала*.

В зависимости от вероятности значения выходного сигнала нейроны подразделяются на *детерминированные модели нейрона* и *стохастические модели нейрона*.

В дальнейшем мы будем рассматривать только формальные модели нейрона.

II.1.5. Задания:

1. Приведите классификацию нейронных сетей в зависимости от их положения в сети.
2. Приведите классификацию нейронных сетей в по типу оперируемых (обрабатываемых) данных.
3. Приведите классификацию нейронных сетей в зависимости от их аппаратной реализации.
4. Приведите классификацию нейронных сетей в зависимости от способов описания их модели.
5. Приведите классификацию нейронных сетей в зависимости от изменяемости весов входов, от вида функции активации и от вероятности значения выходного сигнала.

II.1.5. Вопросы:

1. По какому критерию подразделены входные нейроны, промежуточные нейроны и выходные нейроны?
2. По какому критерию подразделены числовые нейроны и логические нейроны нейроны?
3. По какому критерию подразделены формальные модели, физиологические модели, феноменлогические модели?
4. По какому критерию подразделены нейроны с неизменными весами входов, нейроны с управляемыми весами входов, нейроны с перестраивающимися весами в зависимости от входного по данному каналу сигнала, нейроны с изменяющимися весами под влиянием выходного сигнала?

НИИ искусственный интеллект

5. По какому критерию подразделены детерминированные модели нейрона и стохастические модели нейрона?

НИИ Искусственный интелект

II.1.6. Классификация нейронных сетей

В этом параграфе обсуждается классификация нейронных сетей, предлагаются задания и сформулированы вопросы. При подготовке учебного материала данного параграфа были использованы следующие источники [1-9].

Искусственная нейронная сеть (ИНС, нейросеть) - это набор нейронов, соединенных между собой. Как правило, функции активации всех нейронов в сети фиксированы, а веса являются параметрами сети и могут изменяться. Некоторые входы нейронов помечены как внешние входы сети, а некоторые выходы - как внешние выходы сети. Подавая любые числа (сигналы) на входы сети, мы получаем какой-то набор чисел (сигналов) на выходах сети.

Таким образом, работа нейросети состоит в преобразовании входного вектора в выходной вектор, причем это преобразование задается весами сети.

Классификацию нейронных сетей можно произвести по следующим признакам:

1. ***Тип сигналов в нейронной сети.*** По типу значений используемых сигналов на входах и выходах нейронные сети разделяют на *цифровые* (*бинарные*) и *аналоговые* (*действительные*) *сети*. Бинарные нейронные сети оперируют только двоичными сигналами, и выход каждого нейрона может принимать значение либо логической единицы (возбужденное состояние), либо логического ноля (заторможенное состояние). У аналоговых нейронных сетей значения входных и выходных сигналов будут действительными числами.

2. ***Характер функций активации.*** Если функция активации одна для всех нейронов сети, то сеть называют *однородной* (*гомогенной*) *сетью*. Если же функция активации зависит еще от некоторых параметров, значения которых меняются от нейрона к нейрону, то сеть называют *неоднородной* (*гетерогенной*) *сетью*.

НИИ искусственный интеллект

3. ***Тип функции активации.*** По типу функции активации, используемой на различных слоях нейронной сети, различают *линейные* и *нелинейные сети*.

4. ***Модельное время нейронных сетей.*** По типу модельного (непрерывного или дискретного) времени нейронные сети подразделяются *на сети с непрерывным* и *дискретным временем*.

5. ***Характер настройки синапсов.*** По характеру настройки синапсов нейронные сети делятся на *нейронные сети с фиксированными связями* и *нейронные сети с динамическими связями.*

В нейронных сетях с фиксированными связями множество весов синаптических связей выбираются сразу, исходя из условий задачи, при этом $\frac{dW}{dt}=0$, где W – множество весов синаптических связей сети. В нейронных сетях с динамическими связями множество весов синаптических связей настраиваются в процессе обучения, т.е. $\frac{dW}{dt}\neq 0$, где W – множество весов синаптических связей сети.

6. ***Время передачи сигналов.*** По времени передачи сигнала (по способу изменения состояния нейронов) нейронные сети делятся на *синхронные нейронные сети и асинхронные нейронные сети.* В синхронных нейронных сетях в каждый момент времени лишь один нейрон меняет свое состояние, в асинхронных сетях состояние меняется сразу у целой группы нейронов, как правило, у всего слоя. Алгоритмически ход времени в нейронных сетях задается итерационным выполнением однотипных действий над нейронами.

Способ синхронизации. В ряде нейронных сетей функция активации может зависеть не только от весовых коэффициентов связей w_{ij}, но и от времени передачи сигнала по каналам связи τ_{ij}. Тогда функция активации связи c_{ij} от элемента u_i к элементу u_j имеет вид $c_{ij}^{*}=f\left(w_{ij}(t),\ u_{ij}^{*}(t-\tau_{ij})\right)$. При этом, если время передачи

НИИ Искусственный интеллект

τ_{ij} каждой связи равно либо нулю, либо фиксированной постоянной τ, то такая сеть называется *синхронной нейронной сетью.* Если время передачи τ_{ij} каждой связи между элементами u_i и u_j свое и постоянное, то такая сеть называется *асинхронной нейронной сетью*. При программной реализации нейронных сетей вопрос синхронизации не актуален. Для других способов реализации он весьма важен.

7. ***Виды решаемых задач.*** По виду решаемых задач нейронные сети разделяются на *формируемые сети*, *сети с формируемой матрицей связи*, *комбинированные нейронные сети*, *обучаемые нейронные сети*.

Формируемые нейронные сети проектируются для формализуемых задач, имеющих четко сформулированный в нейросетевом базисе алгоритм решения. Нейронные сети с формируемой матрицей связи применяются для трудноформализуемых задач. Обычно такие нейроннные сети имеют одинаковую структуру и различаются лишь матрицей связи (сеть Хопфилда). Такие сети наглядно показаны в работе.

Комбинированные нейронные сети сочетают в себе признаки двух или трех основных видов сетей и являются многослойными. Каждый слой комбинированной сети представляется различной топологией и обучается по определенному алгоритму. Класс комбинированных нейронных сетей дает самые широкие возможности разработчику.

8. **Тип структуры нейронов**. По типу структуры нейронов нейронные сети делятся на *гомогенные* (однородные) и *гетерогенные* (разнородные) сети. Гомогенные нейронные сети состоят из нейронов одного типа с единой функции активации, в гетерогенную сеть входят нейроны с различными функциями активации.

9. ***Топология.*** По виду топологии нейронные сети подразделяются на *однослойные* и *многослойные.* Топология нейронной сети — это графическая иллюстрация соединения

НИИ Искусственный интеллект

нейронов между собой, представленная направленным графом со взвешенными связями, в котором нейроны являются узлами.

Выбор структуры нейронной сети осуществляется в соответствии с особенностями и сложностью задачи. Для решения отдельных классов задач уже существуют оптимальные конфигурации нейронных сетей. Если же решаемая задача не может быть сведена ни к одному известному классу задач, приходится решать сложную проблему синтеза новой конфигурации. Так проблема синтеза нейронной сети сильно зависит от решаемой задачи, предоставить общие подробные рекомендации затруднительно. В большинстве случаев оптимальный вариант получается на основе интуитивного подбора.

В настоящее время известны следующие нейронные сети: персептрон Розенблатта, многослойный персептрон, сеть Джордана, сеть Элмана, сеть Хэмминга, сеть Ворда, сеть Хопфилда, сеть Кохонена, нейронный газ, когнитрон, неокогнитрон, хаотическая нейронная сеть, осцилляторная нейронная сеть, сеть встречного распространения, сеть радиальных базисных функций (RBF-сеть), сеть обобщенной регрессии, вероятностная сеть, сиамская нейронная сеть, сети адаптивного резонанса.

II.1.6. Задания:

1. Приведите классификацию нейронных сетей по характеру функций активации, по виду топологии и типу связи.
2. Приведите классификацию нейронных сетей по характеру настройки синапсов.
3. Опишите способ распространения входных сигналов в полносвязных нейронных сетях.
4. Опишите способ передачи выходных сигналов в полносвязных нейронных сетях.

II.1.6. Вопросы:

1. По какому критерию подразделены цифровые нейронные сети и аналоговые нейронные сети?

НИИ искусственный интеллект

2. По какому критерию подразделены однородные нейронные сети и неоднородные нейронные сети?

3. По какому критерию подразделены однослойные нейронные сети и многослойные нейронные сети?

4. От чего зависит количество связей нейрона в регулярных нейронных сетях?

НИИ Искусственный интеллект

II.2. Функции активации

II.2.1. Общие сведения о функциях активации

В этом параграфе будут рассмотрены общие сведения о функциях активации, предложены задания и сформулированы вопросы. При подготовке учебного материала данного параграфа были использованы следующие источники [1-9].

Функция активации (активационная функция, передаточная функция) – это функция *f*, вычисляющая выходной сигнал искусственного нейрона *y*, которая в качестве аргумента принимает взвешенную сумму *s*, получаемую на выходе сумматора, т.е. $y=f(s)$.

Вид функции активации является усилительной характеристикой искусственного нейрона, определяет его функциональные возможности и метод его обучения.

Преимущественно применяют нелинейную функцию активации, поскольку линейные функции ограничены и их выход пропорционален входу. Применение линейных функций активации было проблемой в ранних моделях нейронных сетей, и их ограниченность и нецелесообразность была доказана в книге Мински и Пейперта "Персептроны".

В настоящее время имеются много видов функций активации. Примеры функций активации представлены в таблице II.2.1.

Таблица II.2.1. Примеры функций активации.

№	Название	Вид	Области определения и значения
1.	Линейные функции активации (Purelin)	$f(s)=ks$	$D(f)=(-\infty,+\infty)$ $E(f)=(-\infty,+\infty)$
2.	Полулинейные функции активации (Poslin)	$f(s)=\begin{cases} ks, \text{ если } s>0 \\ 0, \text{ если } s\le 0 \end{cases}$	$D(f)=(-\infty,+\infty)$, $E(f)=(-\infty,+\infty)$
3.	Смещенные насыщающие линейные - шаговые функции активации (Satlin)	$f(s)=\begin{cases} 0, \text{ если} s\le T \\ \dfrac{s-t}{\Delta}, \text{ если} T<s<T+\Delta \\ t, \text{ если} s\ge T+\Delta \end{cases}$	$D(f)=(-\infty,+\infty)$ $E(f)=(0,+t)$

4.	Симметричные насыщающие линейные - шаговые функции активации (satlins)	$f(s)=\begin{cases}-t, \text{ если } s\le -T\\ s\cdot\frac{t}{T}, \text{ если } \lvert s\rvert<T\\ t, \text{ если } s\ge T\end{cases}$	$D(f)=(-\infty,+\infty)$ $E(f) = (-t,+t)$
5.	Треугольные функции активации (Tribas)	$f(s)=\begin{cases}0, & \text{если } s<-1\\ 1-\lvert s\rvert, & \text{если } -1\le s\le 1\\ 0, & \text{если } s>1\end{cases}$	$D(f)=(-\infty,+\infty)$ $E(f) = (0,+1)$
6.	Смещенные ступенчатые - пороговые функции активации(Hardlim)	$f(s)=\begin{cases}t, \text{ если } s\ge T\\ 0, \quad \text{иначе}\end{cases}$	$D(f)=(-\infty,+\infty)$ $E(f)=\{0\}\cup\{+t\}$
7.	Симметричные ступенчатые – пороговые функции активации (Hardlims)	$f(s)=\begin{cases}1, \quad \text{если } s\ge 0\\ -1, \quad \text{если } s<0\end{cases}$	$D(f)=(-\infty,+\infty)$, $E(f)=\{-1\}\cup\{+1\}$
8.	Модульные функции активации (Modul)	$f(s)=\lvert s\rvert$	$D(f)=(-\infty,+\infty)$ $E(f) = (0,+\infty)$
9.	Смещенные сигмоидальные– логистические функции активации (Logsig)	$f(s)=\dfrac{1}{1+e^{-\alpha s}}$	$D(f)=(-\infty,+\infty)$ $E(f) = (0,+1)$
10.	Симметричная сигмоидальная функция активации - гиперболический тангенс (Tansig)	$f(s)=\dfrac{e^{2s}-1}{e^{2s}+1}$	$D(f)=(-\infty,+\infty)$ $E(f) = (-1,+1)$
11.	Рациональные сигмоидальные функции активации (Rsigmoid)	$f(s)=\dfrac{s}{k+\lvert s\rvert}$, $f(s)=\dfrac{s}{k+s}$	$D(f)=(-\infty,+\infty)$, $k>0$ $E(f) = (-1,+1)$

НИИ искусственный интеллект

12.	Смещенные пилообобразные функции активации(Sawto)	$f(s)=\begin{cases}1, & \text{если} \quad s\geq 1\\ s, & \text{если} \quad 0<s<1\\ 0, & \text{если} \quad s\leq -1\end{cases}$	$D(f)=(-\infty,+\infty)$, $E(f) = (0,+1)$
13.	Симметричные пилообобразные функции активации (Sawtos)	$f(s)=\begin{cases}1, & \text{если} \quad s\geq 1\\ s, & \text{если} \quad -1<s<1\\ -1, & \text{если} \quad s\leq -1\end{cases}$	$D(f)=(-\infty,+\infty)$, $E(f) = (-1,+1)$
14.	Синусоидальные функции активации (Sin)	$f(s) = \sin(s)$	$D(f)=(-\infty,+\infty)$, $E(f) = (-1,+1)$
15.	Степенные функции активации (Deg)	$f(s) = s^n$	$D(f)=(-\infty,+\infty)$, $E(f)= (0,+\infty)$, $E(f)= (-\infty,+\infty)$
16.	Функции активации - квадратичный корень (Square)	$f(s)=\sqrt{s}$	$D(f)=(0,+\infty)$, $E(f) =(0, +\infty)$
17.	Экспоненциальные функции активации (Exp)	$f(s)=e^{-ks}$	$D(f)=(-\infty,+\infty)$, $E(f) =(0, +\infty)$
18.	Функции активации, уменьшающие входные значения (Softmax)	$f(s_i)=\dfrac{e^{s_i}}{\sum_j e^{s_j}}$,	$D(f)=(-\infty,+\infty)$, $E(f) =(0, +1)$
19.	Радиальные базисные функции активации (Radbas)	$f(s)=\varphi(\dfrac{s^2}{\sigma^2})$	$D(f)=(-\infty,+\infty)$, $E(f) =(0, +1)$
20.	Конкурирующие функции активации (Compet)	$f(s)=\begin{cases}1, \text{ если } s=\max\limits_i \sum\limits_{i=1}^{n} x_i w_i\\ 0, \text{ в ином случае}\end{cases}$	$D(f)=(-\infty,+\infty)$, $E(f) =(0, +1)$

НИИ Искусственный интеллект

II.2.1. Задания:

1. Напишите формулу линейной функции активации.
2. Напишите формулу полулинейной функции активации.
3. Напишите формулу треугольной функции активации.
4. Напишите формулу модульной функции активации.
5. Напишите формулу смещенной ступенчатой функции.
6. Напишите формулу симметричной ступенчатой функции.
7. Напишите формулу логистической функции активации.
8. Напишите формулу гиперболической функции активации.
9. Напишите формулу экспоненциальной функции активации.
10. Напишите формулу синусоидальной функции активации.
11. Напишите формулу рациональной функции активации.
12. Напишите формулу шаговой функции активации.
13. Напишите формулу пороговой функции активации.
14. Напишите формулу смещенной пилообобразной функции.
15. Напишите формулу сигнатурной функции активации.
16. Напишите формулу квадратичной функции активации.
17. Напишите формулу радиально-базисной функции активации.

II.2.1. Вопросы:

1. Для чего нужно использовать функции активации?
2. Куда подается сигнал *s*?
3. Что определяет функция активации?
4. На какие виды подразделяются функции активации?
5. Как определяются синусоидальные функции активации?
6. Как определяются радиальные базисные функции?
7. Как определяются пилообразные функции активации?
8. Как определяются функции гиперболический тангенс?
9. Как определяются рациональные сигмоидальные функции?
10. Как определяются логистические базисные функции?
11. Как определяются модульные функции активации?
12. Как определяются сигмоидальные функции активации?
13. Как определяются конкурирующие функции активации?

НИИ искусственный интеллект

14. Как определяются радиально-базисные функции активации?

НИИ Искусственный интеллект

II.2.2. Линейные функция активации

В этом параграфе будут рассмотрены линейные функции активации и их модификации, предложены задания и сформулированы вопросы. При подготовке учебного материала данного параграфа были использованы следующие источники [1-9].

Линейная функция активации вырабатывает выходной сигнал y пропорционально значению потенциала (значению взвешенной суммы) нейрона s и представляется выражением $f(s) = ks$.

У линейной функции активации областью определения является $D(f)=(-\infty,+\infty)$, а областью значений – $E(f) = (-\infty,+\infty)$. Её график показан на рисунке II.2.2.1.

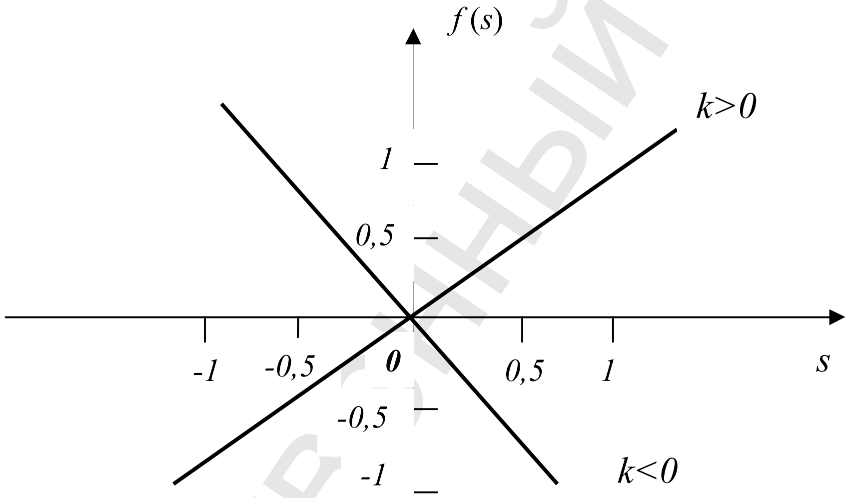


Рисунок II.2.2.1. График линейной функции активации.

Линейная функция активации используется в нейронных сетях различных типов, в том числе линейных, а также в выходных слоях сетей на радиальных базисных функциях.

Линейные функции активации имеют следующие модификации: *полулинейные*, *шаговые и треугольные.*

Полулинейная функция активации: если значение потенциала нейрона положителен, то вырабатывает выходной сигнал y пропорционально потенциалу нейрона s, иначе не вырабатывает выходной сигнал и представляется следующим выражением:

НИИСКВСС Веб- учебный интеллект

$$f(s) = \begin{cases} ks, \text{ если } & s > 0 \\ 0, \text{ если } & s \le 0 \end{cases}$$

В зависимости от положительности или отрицательности значения коэффициента, полулинейная функция может принимать положительное или отрицательное значение соответственно. Поэтому у полулинейной функции активации областью определения является $D(f) = (-\infty,+\infty)$, а областью значений – $E(f) = (-\infty,+\infty)$.

График полулинейной функции показан на рисунке II.2.2.2.

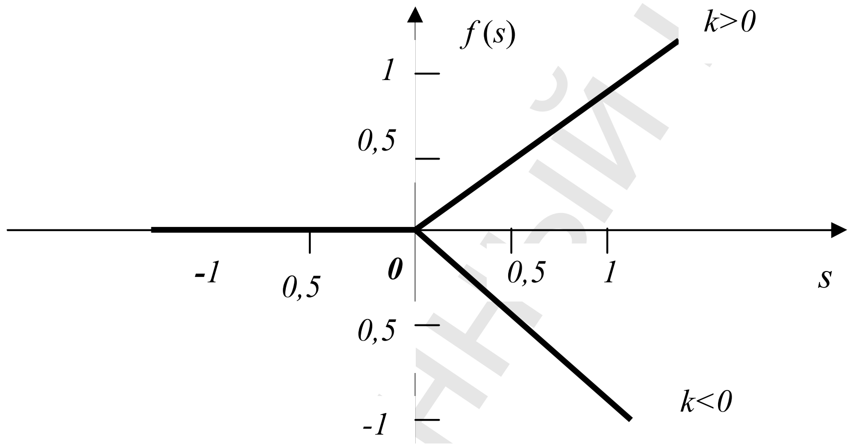


Рисунок II.2.2.2. График полулинейной функции активации.

Полулинейная функция активации используется при моделировании простых сетей, например, персептронов.

Шаговая функция активации (*Линейная функция активации с насыщением*) имеет два линейных участка, где её значение тождественно равна минимально допустимому -*t* и максимально допустимому значению *t* и есть участок, на котором функция строго монотонно возрастает. При этом возможен сдвиг функции по обеим осям: *t* – по оси абциссы, Δ – по оси ординат.

У шаговой функции активации областью определения является $D(f) = (-\infty,+\infty)$, а областью значений – $E(f) = (-t,+t)$.

Имеются два вида шаговой функции: *смещенная шаговая функция*, *симметричная шаговая функция*.

НИИ Искусственный интеллект

Смещенная шаговая функция активации представляется следующим выражением:

$$f(s) = \begin{cases} 0, \text{ если } s \le T \\ \dfrac{s-t}{\Delta}, \text{ если } T < s < T+\Delta \\ t, \text{ если } s \ge T+\Delta \end{cases}$$

График шаговой функции показан на рисунке II.2.2.3.

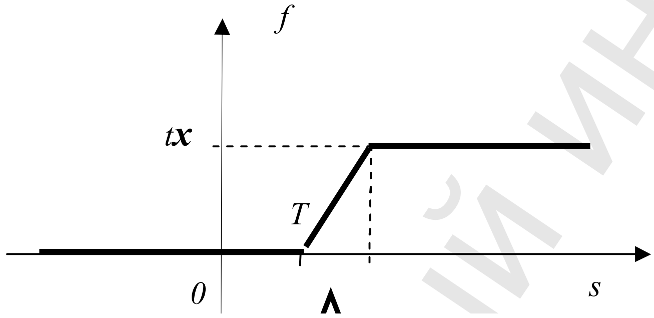


Рисунок II.2.2.3. График смещенной шаговой функции активации.

В общем случае симметричная шаговая функция активации представляется следующим выражением:

$$f(s) = \begin{cases} -t, \text{ если } s \le -T \\ s \cdot \dfrac{t}{T}, \text{ если } |s| < T \\ t, \text{ если } s \ge T \end{cases}$$

При *t=1* и *T=1* график симметричной шаговой функции будет как на рисунке II.2.2.4.

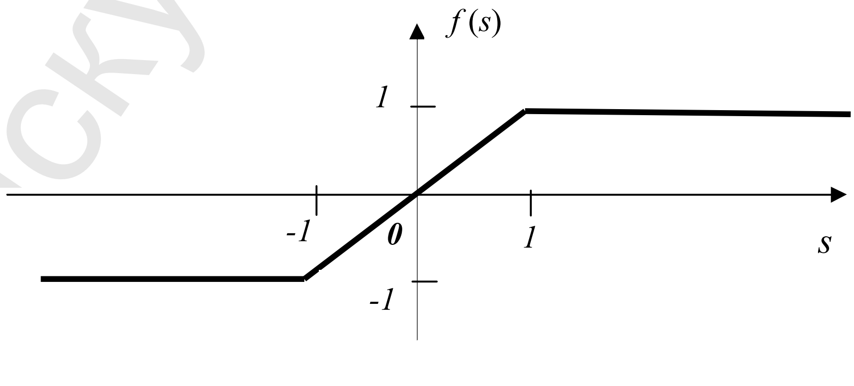


Рисунок II.2.2.4. График симметричной шаговой функции активации.

Шаговая функция активации используется при моделировании простых сетей, например, персептронов.

Недостатками полулинейной и шаговой функций активации относительно линейной можно назвать то, что они не являются дифференцируемыми на всей числовой оси, а значит не могут быть использованы при обучении по некоторым алгоритмам, требующим дифференцируемости функции активации, например, алгоритм обратного распространения ошибок.

Треугольная функция активации (Кусочно-линейная функция с ограничением) и имеет два линейных участка, где её значение тождественно равна нулю и есть два участка, на первом функция строго монотонно возрастает и на втором функция строго монотонно убывает.

$$f(s)=\begin{cases} 0, & \text{если} \quad s<-1 \\ 1-|s|, & \text{если} \quad -1\le s\le 1 \\ 0, & \text{если} \quad s>1 \end{cases}$$

График треугольной функции активации представлен на рисунке II.2.2.5.

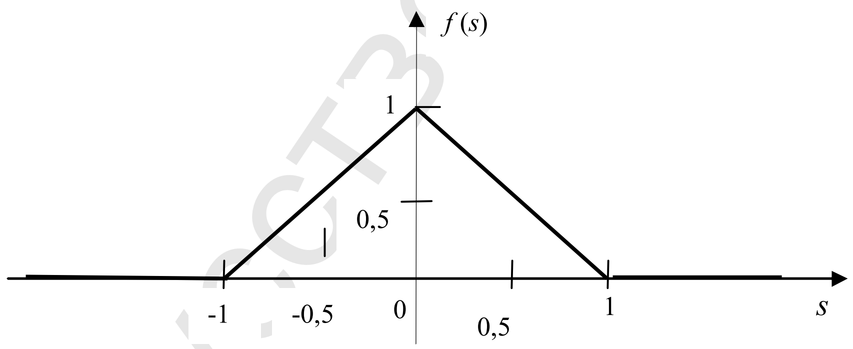


Рисунок II.2.2.5. График треугольной функции активации.

II.2.2. Задания:

1. Напишите формулу линейной функции активации.
2. Напишите формулу полулинейной функции активации.
3. Напишите формулу смещенной пошаговой функции активации.

НИИ Электронный архив

4. Напишите формулу симметричной пошаговой функции активации.

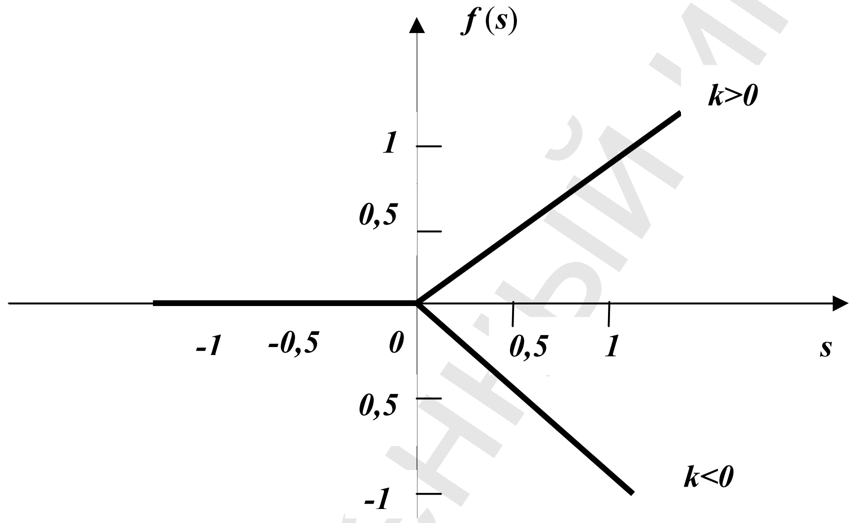
5. Напишите формулу треугольной функции активации.

II.2.2. Вопросы:

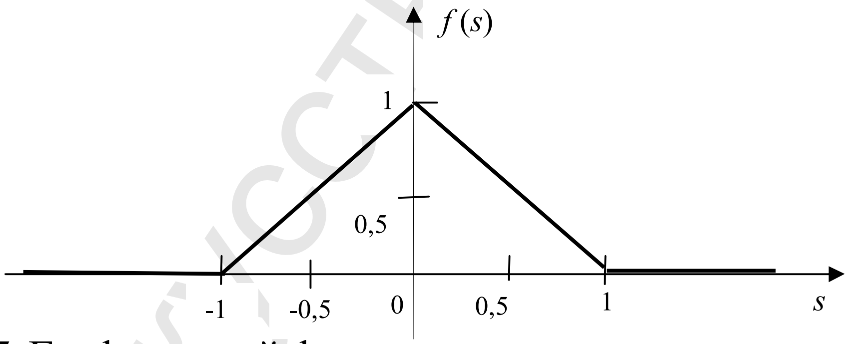
1. Чем отличается полулинейная функция активации от линейной функции?

2. Чем отличается линейная функция активации от треугольной функции?

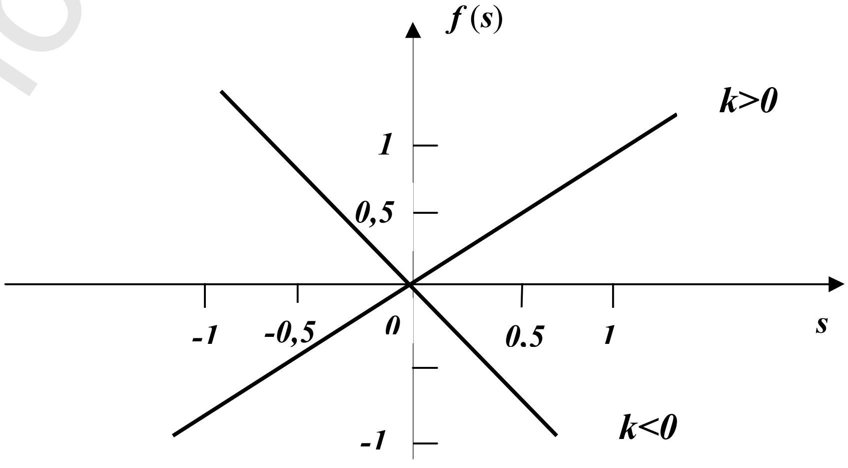
3. График какой функции показан на нижнем рисунке?



4. График какой функции показан на нижнем рисунке?



5. График какой функции показан на нижнем рисунке?



НИИ Искусственный интеллект

II.2.3. Пороговые функции активации

В этом параграфе будут рассмотрены пороговые функции активации, предложены задания и сформулированы вопросы. При подготовке учебного материала данного параграфа были использованы следующие источники [1-9].

Пороговые функции активации представляют собой перепад и вырабатывают выходной сигнал с предельными значениями a и b в зависимости от активности нейрона. При этом, если значение потенциала нейрона не достигает некоторого уровня T, то выходной сигнал равен нулю, в противном случае выходной сигнал скачкообразно изменяется на t.

Смешенная пороговая функция представляется выражением:

$$f(s) = \begin{cases} t, \text{ если } & s \geq T \\ 0, & \text{иначе} \end{cases}$$

Здесь T является сдвигом функции активации относительно горизонтальной оси и $T = -w_0 \cdot x_0$, соответственно под s следует понимать взвешенную сумму сигналов на входах нейрона без учёта этого слагаемого.

Пороговую функцию активацию иногда называют *ступенчатой функцией активации.*

Имеются *смещенная пороговая функция активации* и *симметричная пороговая функция активации.*

У смещенной пороговой функции активации областью определения является $D(f) = (-\infty, +\infty)$, а областью значений – $E(f) = \{-t\} \cup \{+t\}$. На рисунке II.2.3.1 представлен график смещенной пороговой функции активации:

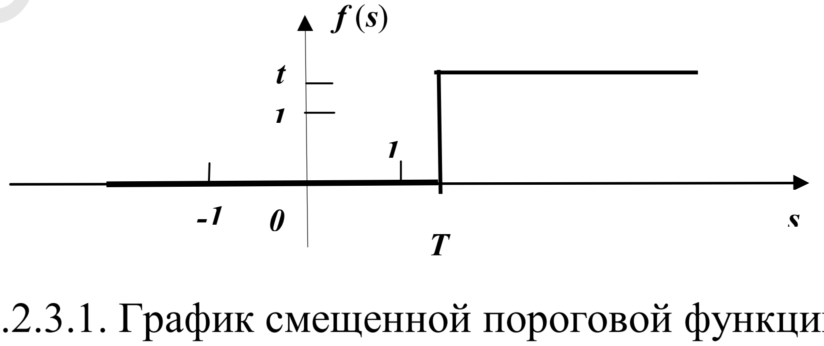


Рисунок II.2.3.1. График смещенной пороговой функции.

НИИ Искусственный интеллект

Если в смещенной пороговой функции активации *T=0* и *t=0*, то она превратится в *функцию единичного скачка* (*функцию Хэвисайда*), представленную следующим выражением и на рисунке II.2.3.2.

$$f(s) = \begin{cases} 1, & \text{если} \quad s \geq 0 \\ 0, & \text{если} \quad s < 0 \end{cases}$$

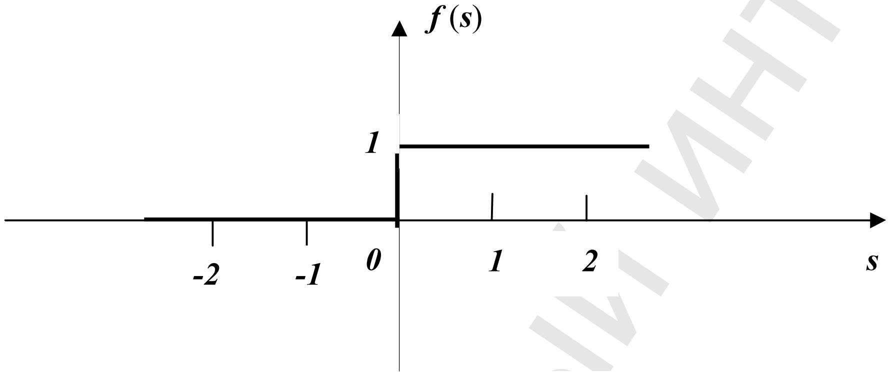


Рисунок II.2.3.2. График функции активации единичного скачка.

Симметричная пороговая (знаковоя) функция активации представляется следующим выражением:

$$f(s) = \begin{cases} 1, & \text{если} \ s \geq 0 \\ -1, & \text{если} \ s < 0 \end{cases}$$

Областью определения симметричной пороговой функции активации является интервал *D*(*f*) = (–∞,+∞), а областью значений – множество *E*(*f*) = {-1}∪{1}.

Симметричная пороговая функция иногда называется *биполярной функцией* активации или *жесткой знаковой функцией*.

График симметричной пороговой функции активации представлен на рисунке II.2.3.3.

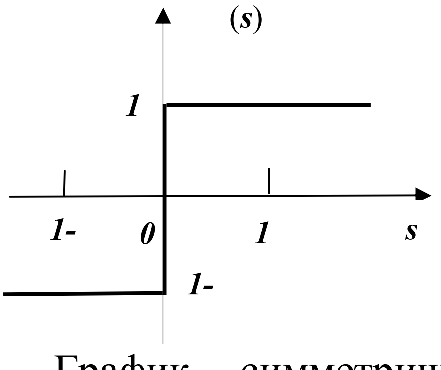


Рисунок II.2.3.3. График симметричной пороговой функции

НИИ Искусственный интеллект

Можно определить многопороговую функцию активации, если выходной сигнал нейрона будет принимать одно из *m* значений, определяемых *m-1* порогом внутри предельных значений *a* и *b*.

Ввиду того, что данная функция не дифференцируема на всей оси абсцисс, её нельзя использовать в сетях, обучающихся по алгоритму обратного распространения ошибки и другим алгоритмам, требующим дифференцируемости функции активации.

Пороговые функции могут использоваться при моделировании нейронных сетей, которые не обладают гибкостью при настройке и обучении их на решаемую задачу, т.к., если зачение суммы не достигает порога, то выходной сигнал не формируется, и сеть «не срабатывает». Теряется интенсивность выходного сигнала нейрона и формируется невысокое значение уровня на взвешенных входах в следующем слое нейронов.

II.2.3. Задания:

1. Приведите виды пороговых функций активации.
2. Напишите выражение, определяющее значение порога *T*.
3. Напишите формулу смещенной пороговой функции.
4. Нарисуйте график пороговой функции.
5. Напишите формулу симметричной пороговой функции.
6. Нарисуйте график смещенной пороговой функции.
7. Нарисуйте график симметричной пороговой функции.

II.2.3. Вопросы:

1. Как иногда называется пороговая функция?
2. Что является сдвигом функции активации относительно горизонтальной оси?
3. Что является областью значения пороговой функции?
4. Чем отличается смещенная пороговая функция от симметричной пороговой функции?
5. Как иногда называется симметричная пороговая функция?
6. Используется ли пороговая функция активации для обучения нейронной сети по алгоритму обратного распространения ошибки?

НИИ искусственный интеллект

7. Можно ли использовать для обучения линейных функций и их модификации метод обратного распространения ошибок.

НИИ Искусственный интеллект

II.2.4. Модульные функции активации

В этом параграфе будут рассмотрены модульные функции активации, предложены задания и сформулированы вопросы. При подготовке учебного материала данного параграфа были использованы следующие источники [1-9].

Модульная функция активации вырабатывает непрерывный линейный выходной сигнал в зависимости от активности нейрона и представляется следующим выражением:

$$f(s) = |s|$$

У модульной функции активации областью определения является $D(f) = (-\infty, +\infty)$, а областью значений – $E(f) = (0, +\infty)$. График модульной функции активации показан на рисунке II.2.4.1.

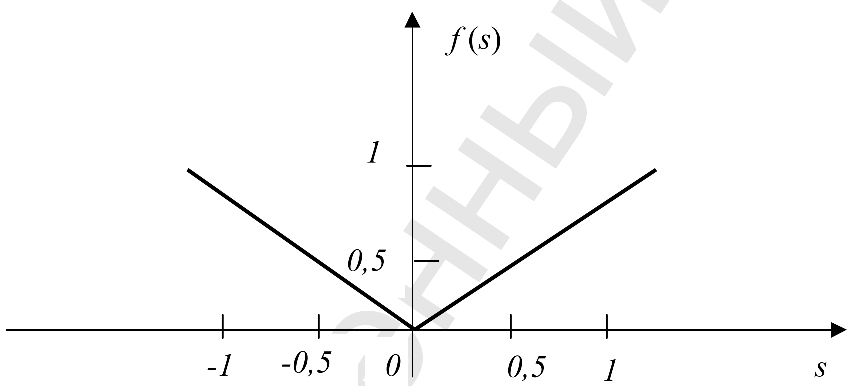


Рисунок II.2.4.1. График модульной функции активации.

Для обучения нейрона с модульной функции активации можно исользовать метод обратного распространения ошибки

Для взятия производной от этой функции нужно учесть, что в точках, где модульная функция равна 0, производная не существует, в остальных точках нужно раскрыть модуль и продифференцировать, т.е.

$$f'(s) = \begin{cases} -s, \text{ если } s < 0 \\ s, \text{ если } s > 0 \end{cases}$$

Легко заметить, что выражение для производной модульной функции совпадает с выражением пороговой (знаковой) функции активации. Поэтому график производной модульной функции активации будет иметь вид как на рисунке II.2.4.2.

НИИ Искусственный интеллект

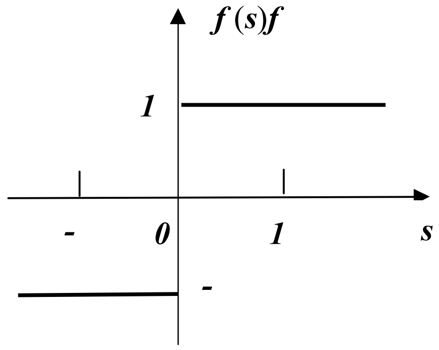


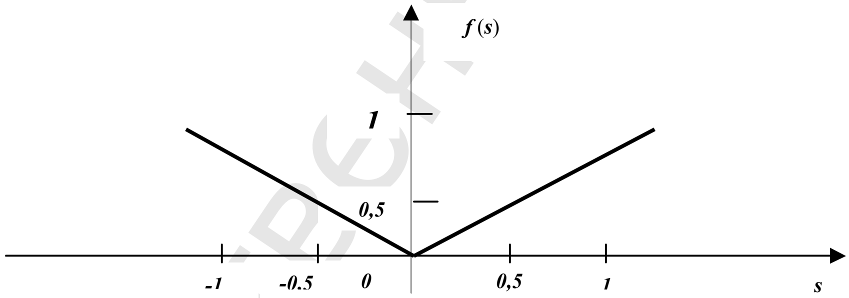
Рисунок II.2.4.2. График производной модульной функции активации.

II.2.4. Задания:

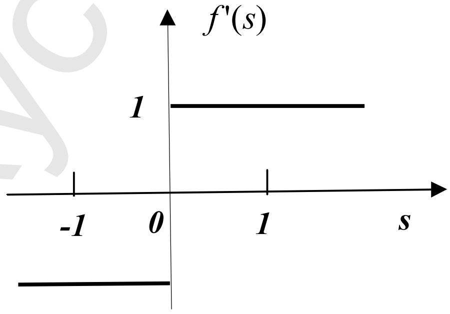
1. Напишите выражение для модульной функции активации.
2. Укажите область определения модульной функции активации.
3. Укажите область значений модульной функции активации.

II.2.4. Вопросы:

1. График какой функции показан на нижнем рисунке?



2. График какой функции показан на нижнем рисунке?



3. Что нужно учесть для взятия производной модульной функции активации?
4. По какой формуле вычисляется производная модульной функции активации?

НИИ Искусственный интеллект

НИИ Искусственный интеллект

II.2.5. Сигмоидальные функции активации.

В этом параграфе будут рассмотрены сигмоидальные функции активации, предложены задания и сформулированы вопросы. При подготовке учебного материала данного параграфа были использованы следующие источники [1-9].

В настоящее время сигмоидальные функции активации являются одними из самых часто используемых функций активации. Введение функций сигмоидального типа было обусловлено ограниченностью нейронных сетей с пороговой функцией активации нейронов.При такой функции активации любой из выходов сети равен либо нулю, либо единице, что ограничивает использование сетей в задачах, не относящихся к задачам классификации. Использование сигмоидальных функций позволило перейти от бинарных (цифровых) выходов нейрона к действительным (аналоговым). Функции активации такого типа, как правило, присущи нейронам, находящимся во внутренних слоях нейронной сети.

Сигмоидальные функции активации вырабатывают непрерывный выходной сигнал, значения которого лежат в интервале (*a*, *b*), и формируют график в форме сигмоида (S-образной кривой).

Сигмоидальные функции активации подразделяются на *смещенные сигмоидальные функции активации*, *симметричные сигмоидальные функции активации* и *рациональные сигмоидальные функции активации*.

Смещенная сигмоидальная функция активации представляется следующим выражением:

$$f(s) = \frac{1}{1+e^{-\alpha s}}$$

где α – это параметр функции, определяющий крутизну. Когда α стремится к бесконечности, функция вырождается в пороговую функцию. При α=0 сигмоидальная функция вырождается в постоянную функцию со значением 0,5.

НИИ Искусственный интеллект

Смещенная сигмоидальная функция иногда называется *логистической функцией активации*.

У логистической функции активации областью определения является $D(f)=(-\infty,+\infty)$, а областью значений – $E(f)=(0, 1)$. График логистической функции активации представлен на рисунке II.2.5.1.

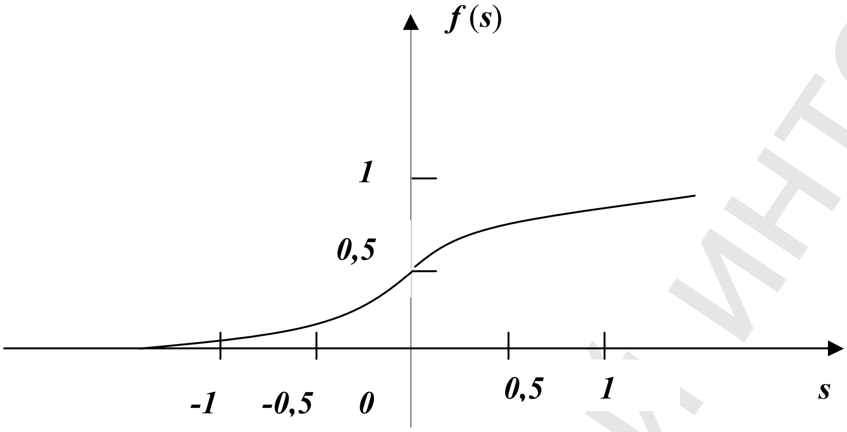


Рисунок II.2.5.1. График логистической функции активации.

Логистическая функция дифференцируема на всей оси абсцисс и имеет очень простую производную:

$$f'(s) = \alpha f(s)(1 - f(s))$$

То, что производная логистической функции может быть выражена через её значение облегчает её использование при обучении нейронной сети по алгоритму обратного распространения. Особенностью нейронов с такой передаточной характеристикой является то, что они усиливают сильные сигналы существенно меньше, чем слабые, поскольку области сильных сигналов соответствуют пологим участкам характеристики. Это позволяет предотвратить насыщение, происходящее из-за больших сигналов.

Симметричная сигмоидальная функция активации представляется выражением гиперболического тангенса:

$$f(s) = \frac{e^{s} - e^{-s}}{e^{s} - e^{-s}} = \frac{e^{2s} - 1}{e^{2s} + 1}.$$

Симметричная сигмоидальная функция иногда называется *гиперболической функцией активации* (*гиперболический тангенс*).

НИИ Искусственный интеллект

У гиперболического тангенса активации областью определения является $D(f)=(-\infty,+\infty)$, а областью значений – $E(f)=(-1, 1)$.

График гиперболической функции активации показан на рисунке II.2.5.2.

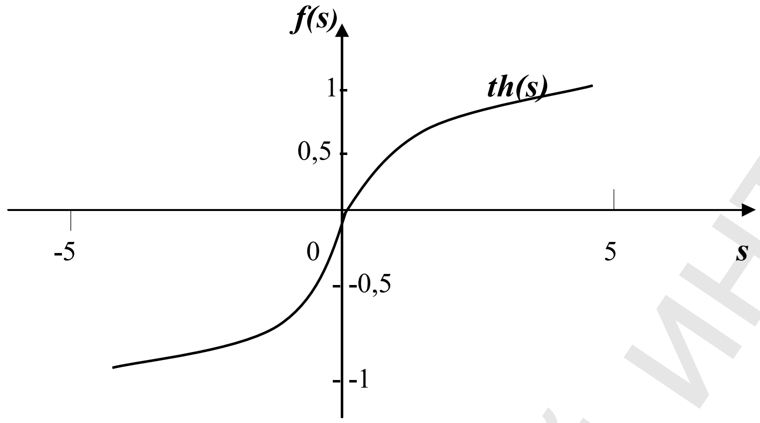


Рисунок II.2.5.2. График гиперболической функции активации.

Гиперболическая функция активации симметрична относительно начала координат, и в точке $s = 0$ значение выходного сигнала $y = f(s)$ равно нулю.

Производная гиперболического тангенса вычисляется так:

$$f'(s) = th(s) = \frac{1}{ch^2(s)}.$$

В отличие от логистической функции, гиперболическая функция принимает значения различных знаков, и это его свойство применяется для целого ряда сетей. Идеально подходит для пользовательской настройки многослойных персептронов.

Рациональная сигмоидальная функция (упрощенная сигмоида) относится к классу сигмоидальных функций активации и имеет вид:

$$f(s) = \frac{s}{k + |s|}, \quad k>0,$$

У рациональной сигмоидальной функции активации область определения есть $D(f)=(-\infty,+\infty)$, а областью значений – $E(f) = (-1, 1)$, $k>0$ – параметр наклона сигмоидальной функции активации,

НИИ Искусственный интеллект

изменяя этот параметр, можно построить функции с различной крутизной. График рациональной сигмоидальной функции активации представлен на рисунке II.2.5.3.

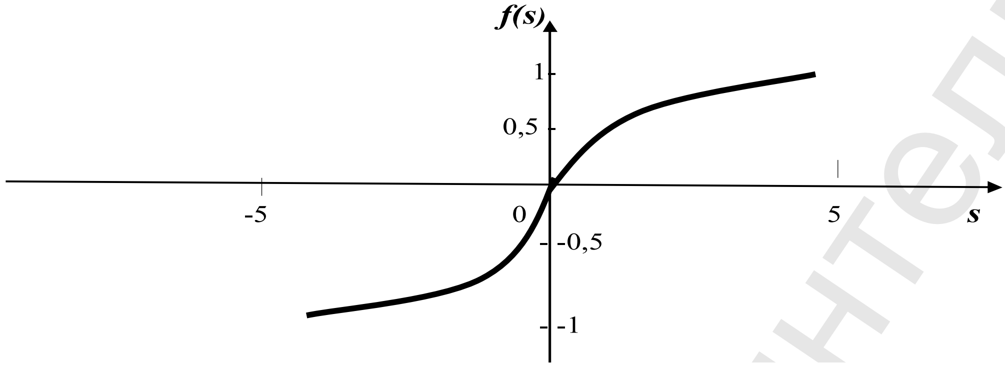


Рисунок II.2.5.3. График рациональной сигмоидальной функции

Сигмоидальные функции активации позволяют выделять в поисковом пространстве исследуемых объектов области сложной формы, в том числе невыпуклые и несвязанные.

II.2.5. Задания:

1. Укажите виды сигмоидальных функций активации.
2. Напишите формулу смещенной сигмоидальной функции.
3. Напишите формулу симметричной сигмоидальной функции.
4. Напишите формулу рациональной сигмоидальной функции.
5. Напишите формулу для производной логистической функции.

II.2.5. Задания:

1. Какой сигнал вырабатывает сигмоидальная функция активации?
2. Относительно чего симметрична гиперболическая функция активации?
3. Какие значения принимает гиперболическая функция в отличие от логистической функции?
4. Какие функции позволяют перейти от бинарных (цифровых) выходов нейрона к действительным (аналоговым)?
5. Функции активации какого типа присущи нейронам, находящимся во внутренних слоях нейронной сети?

НИИ Искусственный интеллект

6. Для решения каких задач применяется сигмоидальные функции активации?

НИИ Искусственный интеллект

II.2.6. Пилообразные функции активации

В этом параграфе будут рассмотрены пилообразные функции активации, предложены задания и сформулированы вопросы. При подготовке учебного материала данного параграфа были использованы следующие источники [1-9].

Пилообразная функция активации является кусочно-линейным вариантом сигмоидальной функции и вырабатывает значения, промежуточные между двумя симметричными предельными значениями *a* и *b*.

Пилообразные функции активации подразделяются на *смещенные пилообразные функции активации* и *на симметричные пилообразные функции активации.*

Смещенная пилообразная функция представляется выражением:

$$f(s) = \begin{cases} 1, & \text{если} \quad s \geq 1 \\ s, & \text{если} \quad 0 < s < 1 \\ 0, & \text{если} \quad s \leq -1 \end{cases}$$

У смещенной пилообразной функции активации область определения есть $D(f) = (-\infty, +\infty)$, а область значений – $E(f) = (0, 1)$.

График пилообразной функции активации представлен на рисунке II.2.6.1.

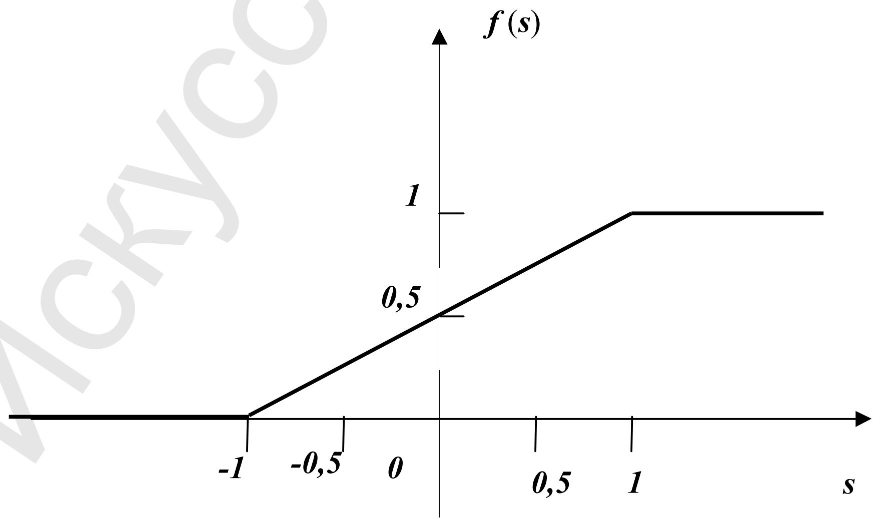


Рисунок II.2.6.1. График смещенной пилообразной функции активации.

НИИ искусственный интеллект

Симметричная пилообразная функция активации представляется следующим выражением:

$$f(s)=\begin{cases} 1, & \text{если} \quad s \ge 1 \\ s, & \text{если} \quad -1 < s < 1 \\ -1, & \text{если} \quad s \le -1 \end{cases}$$

У симметричной пилообразной функции активации область определения есть $D(f)=(-\infty,+\infty)$, а область значений – $E(f) = (-1,1)$.

График симметричной пилообразной функции представлен на рисунке II.2.6.2.

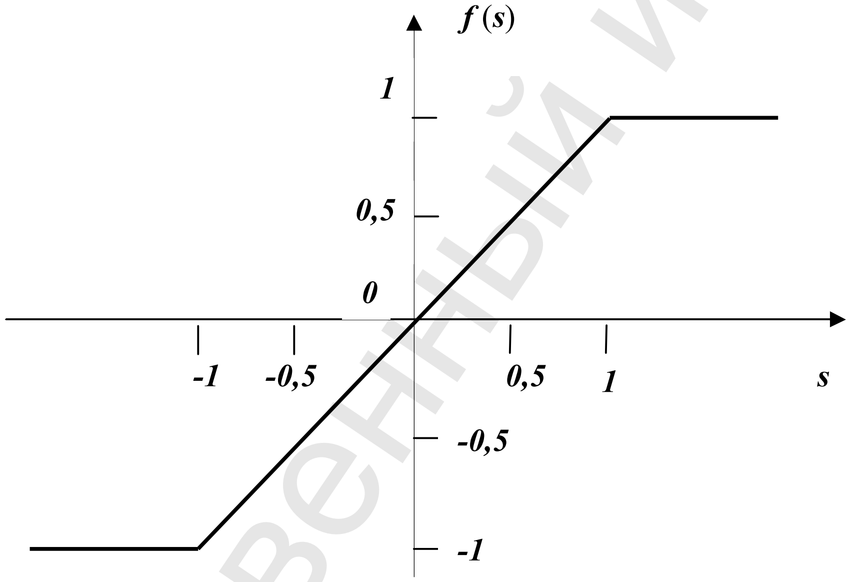


Рисунок II.2.6.2. График симметричной пилообразной функции активации.

Симметричная пилообразная функция активации является кусочно-линейным вариантом сигмоидальной функции.

II.2.6. Задания:

1. Приведите виды пилообразной функции активации.
2. Напишите формулу смещенной пилообобразной функции.
3. Напишите формулу симметричной пилообобразной функции.

II.2.6. Вопросы:

1. Какая связь имеется между сигмоидальной функцией и пилообразной функцией активации?
2. Что является областью значений у смещенной сигмоидальной функции активации?

НИИ искусственный интеллект

3. В чем разница между смещенной и симметричной сигмоидальной функциями активации?

НИИ Искусственный интеллект

II.2.7. Синусоидальные функции активации

В этом параграфе будут рассмотрены синусоидальные функции активации, предложены задания и сформулированы вопросы. При подготовке учебного материала данного параграфа были использованы следующие источники [1-9].

Синусоидальные функции активации являются периодическими функциями с периодом 2π, где π =3,14…

Синусоидальная функция активации представляется выражением:

$$f(s) = sin(s) \text{ или } f(s) = cos(s)$$

У синусоидальной функции активации областью определения является $D(f)=(-\infty,+\infty)$, а областью значений – $E(f)=(-1, +1)$.

График синусоидальной функции представлен на рисунке II.2.7.1 и II.2.7.2.

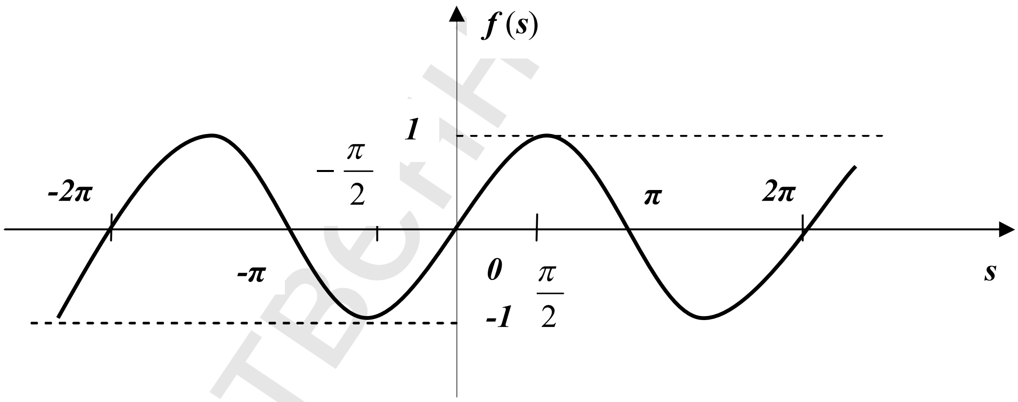


Рисунок II.2.7.1. График функции активации *sin*(*s*).

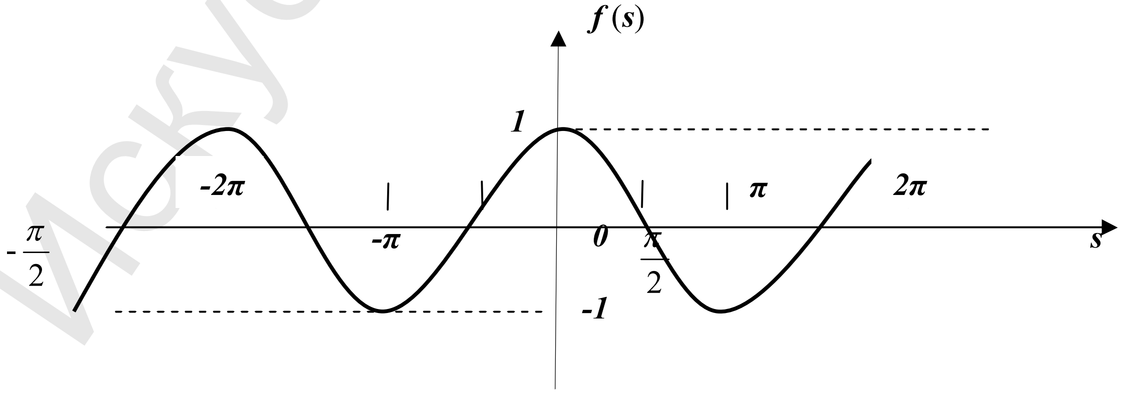


Рисунок II.2.7.2. График функции активации *cos*(*s*).

НИИ Искусственный интеллект

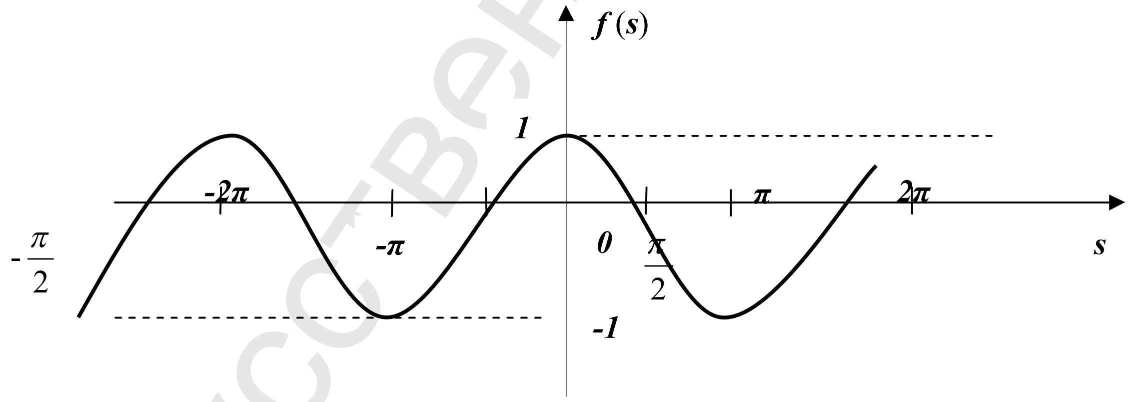
Синусоидальная функция активации может быть полезна для распознавания радиально распределенных данных, по умолчанию не используется.

II.2.7. Задания:

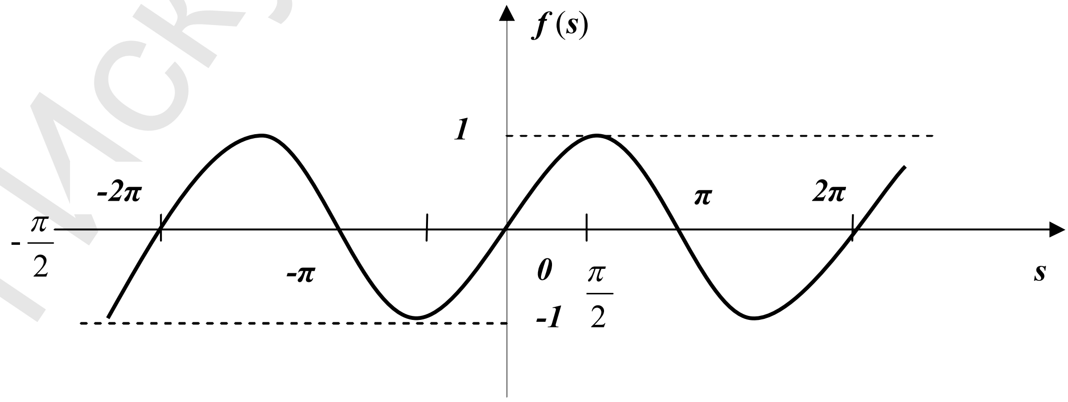
1. Назовите виды синусоидальной функции активации.
2. Напишите формулу синусоидальной функции активации.
3. Нарисуйте график функции активации *sin*(*x*).
4. Нарисуйте график функции активации *cos*(*x*).

II.2.7. Вопросы:

1. Что является областью определения и областью значения у синусоидальной функции активации?
2. Что является областью значений у синусоидальной функции активации?
3. Для каких задач может быть полезна синусоидальная функция активации?
4. График какой функции активации показан на рисунке ниже?



5. График какой функции активации показан на рисунке ниже?



НИИ Искусственный интеллект

II.2.8. Степенные функции активации

В этом параграфе будут рассмотрены степенные функции активации, предложены задания и сформулированы вопросы. При подготовке учебного материала данного параграфа были использованы следующие источники [1-9].

Степенные функции активации представляются выражением:

$$f(s) = s^n,$$

где постоянная величина n =2,3,4,… - показатель степени.

У степенных функций активации областью определения является $D(f)=(-\infty,+\infty)$, а область значений зависит от четности показателя степени: если значение показателя четное, то область значений есть $E(f)= (0,+\infty)$, иначе будет $E(f)= (-\infty,+\infty)$.

Требованиям монотонности, непрерывности и дифференцируемости, предъявляемым к функциям активации в полной мере отвечают лишь степенные функции с нечетными показателями степени.

II.2.7. Примеры:

1. В степенной функции, если n=2, то речь идет о квадратичной функции активации. График квадратичной функции активации показан на рисунке II.2.8.1.

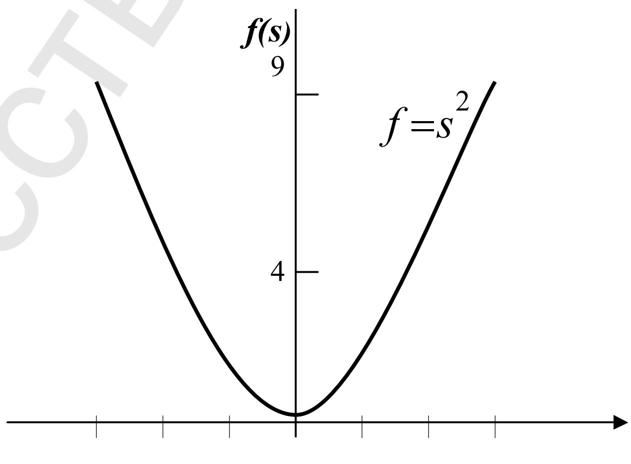


Рисунок II.2.8.1. График квадратичной функции активации.

2. В степенной функции при n=3, имеем дело с кубической квадратичной функцией активации.

НИИ Искусственный интеллект

3. График кубической функции активации представлен на рисунке II.2.8.2.

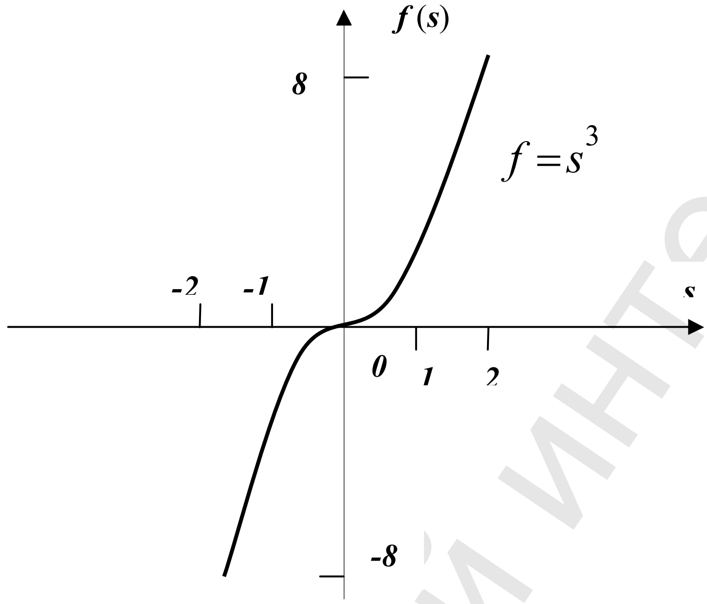


Рисунок II.2.8.2. График кубической функции активации.

II.2.8. Задания:

1. Каким выражением представляются степенные функции активации в общем виде?

2. Покажите область определения квадратичной функции активации.

3. Нарисуйте график квадратичной функции активации.

4. Покажите область определения кубической функции активации.

5. Нарисуйте график кубической функции активации.

II.2.8. Вопросы:

1. К чему относятся требования монотонности, непрерывности и дифференцируемости?

2. Как определяется область значений у степенной функции активации, если значение показателя степени является четным?

3. Как определяется область значений у степенной функции активации, если значение показателя степени является нечетным?

4. Что является областью значений у квадратичной функции активации?

5. Какая область значений у кубической функции активации?

НИИ искусственный интеллект

II.2.9. Функция активации квадратный корень

В этом параграфе будет рассмотрена функция активации квадратный корень, предложены задания и сформулированы вопросы. При формировании учебного материала данного параграфа были использованы следующие источники [1-9].

Функция активации квадратный корень представляется следующим выражением:

$$f(s) = \sqrt{s}$$

Областью определения функции активации квадратный корень является $D(f)=(0,+\infty)$, а областью значений – $E(f) = (0, +\infty)$.

Функция активации квадратный корень используется в нейронной сети Кохонена. Квадратный корень преобразует активации сети Кохонена, т.е. квадраты расстояний, в выходные значения, представляющие сами расстояния. Сеть Кохонена рассчитана на обучение без учителя и применяется для разведывательного анализа данных (распознавания кластеров в данных, устанавления близости классов, решения задач классификации, обнаружения новых явлений).

II.2.9. Задания:

1. Напишите выражение для представлени функции активации квадратный корень.
2. Покажите область определения функции активации квадратный корень.
3. Покажите область значений функции активации квадратный корень.

II.2.9. Вопросы:

1. В каких нейронных сетях используется функция активации квадратный корень?
2. Что делает функция активации квадратный корень с нейронной сетью Кохонена?
3. Почему квадраты расстояний преобразуются в выходные значения нейронных сетей, представляющие сами расстояния?

НИИ искусственный интеллект

II.2.10. Экспоненциальные функции активации

В этом параграфе будут рассмотрены экспоненциальные функции активации, предложены задания и сформулированы вопросы. При подготовке учебного материала данного параграфа были использованы следующие источники [1-9].

Экспоненциальная функция активации имеет вид:

$$f(s) = e^s,$$

где $e = 2{,}71$ – постоянная величина.

Областью определения экспоненциальной функции активации является $D(f)=(-\infty,+\infty)$, а область значений – $E(f) = (0, +\infty)$.

График экспоненциальной функции активации представлен на рисунке II.2.10.

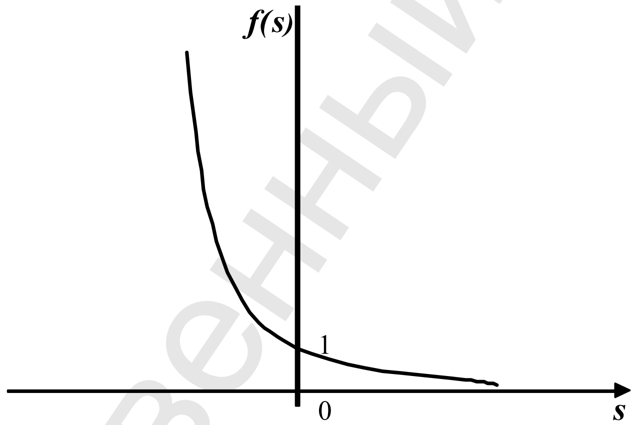


Рисунок II.2.10. График экспоненциальной функции активации.

Экспоненциальная функция активации используется в первом скрытом слое нейронов вероятностных нейронных сетей. Достоинством таких сетей является то, что их архитектура позволяет не только осуществить классификацию объектов, но и определить вероятность справедливости принимаемых решений.

Иногда используют экспоненциальные функции, выходы которых нормируются так, чтобы сумма всех активаций слоя равнялась 1. Их применяют в выходных слоях многослойных персептронов, специально сконструированных для задач классификации таким образом, чтобы выходы можно было интерпретировать как вероятности принадлежности к классу.

НИИ искусственный интеллект

Экспоненциальная функция активации идеально подходит для радиальных элементов. Комбинация радиальной постсинаптической потенциальной функции (PSP-функции) и отрицательной экспоненциальной функции активации дает элемент, моделирующий гауссову функцию (колоколообразной формы) с центром в векторе весов. Стандартное отклонение Гауссиана определяется по формуле $\sigma = \sqrt{1/d}$, где через d обозначено "отклонение" элемента, хранимое как его пороговое значение.

Напомним, что PSP-функция применяется к входным сигналам, его весам и порогам и выдает уровень активации этого элемента. Наиболее часто используются линейные (взвешенная сумма входов минус порог) и радиальные (промасштабированный квадрат расстояния от вектора весов до входного вектора) PSP-функции.

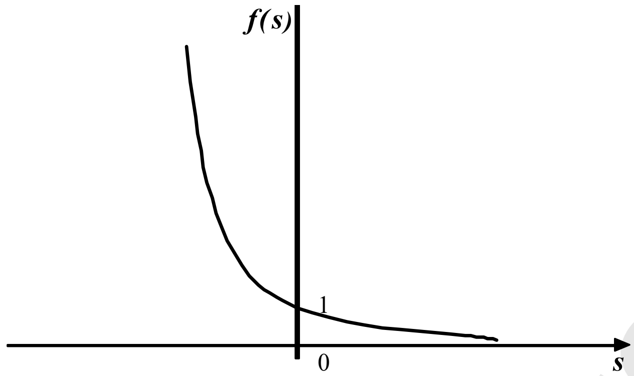
II.2.10. Задания:

1. Напишите выражение для представлени экспоненциальной функции активации.
2. Покажите область определения экспоненциальной функции активации.
3. Нарисуйте график экспоненциальной функции активации.

II.2.10. Вопросы:

1. В каких нейронных сетях используются экспоненциальные функции активации?
2. Что является областью значений экспоненциальной функции активации?
3. График какой функции активации нарисован на рисунке ниже?

НИИ искусственный интеллект



НИИ Искусственный интеллект

II.2.11. Функции активации, уменьшающие входные значения

В этом параграфе будут рассмотрены функции активации, уменьшающие входные значения, предложены задания и сформулированы вопросы. При подготовке учебного материала данного параграфа были использованы следующие источники [1-9].

Функция активации, уменьшающая входные значения (*softmax*), является экспоненциальной функцией и нормирует значение выхода так, чтобы сумма всех активаций нейронов равнялась единице.

В многослойной нейронной сети каждый *i*-тый нейрон softmax-группы будет иметь функцию активации, которая представляется следующим выражением:

$$y_i = \frac{e^{s_i}}{\sum_j e^{s_j}}, \quad (II.2.11.1)$$

где $i \in N$ – индекс нейрона, $j \in N$ – номер слоя.

Областью определения функции активации softmax является $D(f) = (-\infty, +\infty)$, а областью значений – $E(f) = (0, +1)$.

Из выражения (II.2.11.1) видно, что выход каждого нейрона зависит от сумматоров всех остальных нейронов softmax-группы, а сумма выходных значений всей группы равняется единице.

На каждом обучающем примере мы будем получать выход сети, который моделирует нужное нам вероятностное распределение, а для сравнения двух вероятностных распределений необходима корректная мера. В качестве такой меры будет использоваться перекрестная энтропия:

$$C = -\sum_j d_j \log y_j, \quad (II.2.11.2)$$

где d_j – требуемые выходы для обучающего примера *j*, y_j – реальные выходы для обучающего примера *j*.

Функции активации softmax используются для решения задач классификации, когда выход нейрона нужно интерпретировать как вероятность принадлежности к определенному классу, т.е. когда

НИИ искусственный интеллект

необходимо нейронной сетью обеспечить способ моделирования вероятностного распределения . Для этого используется многослойная нейронная сеть прямого распространения, такая что:

- сеть содержит некоторое количество скрытых слоев, все нейроны могут иметь свою собственную функцию активации;
- на последнем слое находится такое количество нейронов, которое соответствует количеству классов; все эти нейроны будут называться softmax-группой.

II.2.11. Задания:

1. Напишите формулу для функции активации softmax.
2. Покажите области определения и значений функции активации softmax.
3. Напишите формулу для перекрестной энтропии.
4. Определите название группы, когда на последнем слое нейронной сети находится такое количество нейронов, которое соответствует количеству классов.
5. Объясните условие «когда необходимо нейронной сетью обеспечить способ моделирования вероятностного распределения».
6. Опишите понятие «перекрестная энтропия».

II.2.11. Вопросы:

1. Является ли функция активации softmax экспоненциальной функцией?
2. С какой целью функция активации softmax нормирует значение выхода искусственного нейрона?
3. Что зависит от сумматоров всех остальных нейронов softmax-группы?
4. Чему равна сумма выходных значений всей группы?
5. Для какой задачи используется функция активации softmax?
6. Что означает условие «выход нейрона нужно интерпретировать как вероятность принадлежности к определенному классу»?

НИИ искусственный интеллект

7. Для чего используется многослойная нейронная сеть прямого распространения?

НИИ искусственный интеллект

II.2.12. Функции активации с переменной крутизной

В этом параграфе будут рассмотрены функции активации с переменной крутизной, предложены задания, сформулированы вопросы. При подготовке учебного материала данного параграфа были использованы следующие источники [1-9].

Степенные функции активации с переменной крутизной определятся выражением следующего вида:

$$y=k\left(\sum_{i=1}^{n} x_i w_i + w_0\right)^t,$$

В этой модели коэффициент k изменяет крутизну параболы ***t***-ой степени, что значительно расширяет функциональные возможности отдельного нейрона. Непосредственная реализация такого нейрона требует настройку двух параметров w_i и k**.** Чтобы уменьшить время настройки нейрона, а в конечном счете и время решения конкретной задачи, можно, видоизменяя выражение этой модели, отказаться от настройки k**.** Для этого нужно внести коэффициент k под знак суммы, т.е. можно написать

$$y=k\left(\sum_{i=1}^{n} x_i w_i + w_0\right)^t = \left(\sum_{i=1}^{n} x_i k^t w_i + k^t w_0\right)^t = \left(\sum_{i=1}^{n} x_i u_i + u_0\right)^t,$$

где коэффициенты $u_j = k^t w_j$, $j=0,1,2,\ldots,t$, что дает возможность варьировать крутизной при общей настройке синаптических коэффицентов нейрона.

Для сигмоидальной, экспоненциальной, и синусоидальной функций активации также полезно использовать коэфициент крутизны сигмоида k. Тогда определяющие их выражения запишутся в виде:

$f(s) = \dfrac{1}{1+e^{-ks}}$ - для смещенной сигмоидальной функции активации;

$f(s) = \dfrac{1-e^{-ks}}{1+e^{-ks}}$ - для симметричной сигмоидальной функции активации;

НИИ Искусственный интеллект

$$f(s) = \frac{e^{ks} - e^{-ks}}{e^{ks} + e^{-ks}}$$ - для гиперболической функции активации;

$$f(s) = \frac{s}{k + |s|}$$ - для рациональной сигмоидальной функции активации;

$$f(s) = e^{-ks}$$ - для экспоненциальной функции активации;

$$f(s) = \sin(ks)$$ - для синусоидальной функции активации.

На практике применяют функции активации других видов, отличных от перечисленных выше. Все определяется характером решаемой задачи. В каждом конкретном случае разработчик должен обосновать свой выбор:

1) Единичная сумма в вероятностных нейронных сетях для перемасштабирования данных так, чтобы сумма всех значений равнялась 1. Эта функция представляется выражением:

$$f(s) = \frac{s}{\sum_i s_i}$$

Область значений этой функции активации $\boldsymbol{E(f)} = (0, +1)$

2) Квадратный корень $\sqrt{s}$ преобразует квадраты расстояний в выходные значения, представляющие сами расстояния

$$\sigma = \sqrt{\frac{1}{d}}.$$

Теперь можно отметить недостатки и достоинства нейронов с функциями активации, рассмотренными выше.

Основным недостатком нейронов со ступенчатыми (пороговыми) функциями активации является отсутствие достаточной гибкости при настройке и обучении нейронной сети на решаемую задачу, так как, если зачение $\sum_{i=1}^{n} x_i w_i$ даже незначительно не достигает порога, то выходной сигнал не формируется, и нейрон «не срабатывает». Это означает, что теряется интенсивность выходного сигнала данного нейрона, и,

НИИ Искусственный интеллект

следовательно, формируется невысокое значение уровня на взвешенных входах в следующем слое нейронов.

Такого недостатка в большей степени лишена линейная (кусочно-линейная) функция активации, реализация которой обеспечивает невысокую вычислительную сложность.

Сигмоидальная функция содержит достоинства ступенчатых и линейных функций. По аналогии со ступенчатой функцией она не линейна, и это дает возможность выделять в поисковом пространстве исследуемых объектов области сложной формы, в том числе невыпуклые и несвязанные. С другой стороны, в отличие от ступенчатой функции, она позволяет переходить от одного значения входного сигнала к другому без разрывов. Однако любую из функций активации необходимо рассматривать как приближенную. Учитывая сложность архитектуры и трудность настройки ее параметров на решение определенной задачи, необходимо переходить к более гибким произвольным нелинейным функциям.

II.2.12. Задания:

1. Напишите выражение для степенной функции активации с переменной крутизной.
2. Напишите видоизменение выражения, чтобы внести коэффициент k под знак суммы.
3. Напишите определяющие выражения для сигмоидальных, экспоненциальной, и синусоидальной функций активации.
4. Напишите выражение для степенной функции активации с переменной крутизной.
5. Напишите выражение для функции единичной суммы в вероятностных нейронных сетях.

II.2.12. Вопросы:

1. Какую роль играет коэффициент k в записи функции активации с переменной крутизной?
2. Какими недостатками обладают нейроны со ступенчатыми (пороговыми) функциями активации?

НИИ Искусственный интеллект

3. Содержит ли сигмоидальная функция достоинства ступенчатых и линейных функций?

НИИ Искусственный интеллект

II.2.13. Радиально-базисная функция активации

В этом параграфе будут рассмотрена радиально-базисная функция активации, предложены задания и сформулированы вопросы. При подготовке учебного материала данного параграфа были использованы следующие источники [1-9].

Радиально-базисная функция принимает в качестве аргумента расстояние между входным вектором и некоторым наперед заданным центром функции активации. Значение этой функции тем выше, чем ближе входной вектор к центру. В моделях радиально-базисной функции могут быть использованы различные способы измерения расстояния между векторами, а также функции активации нейронов скрытого слоя.

Имеется большой класс радиальных базисных функций, среди которых наибольший интерес представляют следующие:

1. Мультиквадратичная функция:

$$f(s) = (s^2 + c^2)^{\frac{1}{2}}, \quad c > 0 \cdot$$

2. Обратная мультиквадратичная функция:

$$f(s) = \frac{1}{(s^2 + c^2)^{\frac{1}{2}}}, \quad c > 0 \cdot$$

3. Функция Гаусса:

$$f(s) = e^{-s^2/2\sigma^2}, \quad \sigma > 0.$$

Здесь s – вектор входных сигналов нейрона, σ – ширина окна функции, $f(s)$ – убывающая функция (чаще всего, равная нулю вне некоторого отрезка). Интересно то, что функции 2 и 3 являются локализованными, то есть $f(s) \to 0$ при $s \to 0$, а функция 1 наоборот неограниченно возрастает с ростом s, что не мешает ей тем не менее быть функцией пригодной для использования в сетях RBF.

Областью определения радиально-базисной функции является $D(f) = (-\infty, +\infty)$, а областью значений – $E(f) = (0, +1)$.

характеризуется тремя особенностями:

НИИ искусственный интеллект

1. Единственный скрытый слой;
2. Только нейроны скрытого слоя имеют нелинейную активационную функцию;
3. Сумма всех весов синаптических связей входного и скрытого слоев равны единице.

В качестве радиально-базисной функции чаще всего используют модифицированную функцию Гаусса:

$$y = f(s) = e^{-(S-R)^2/2\sigma^2},$$

где $S = \|X - C\|$ – расстояние между центром C и вектором входных сигналов X. Скалярный параметр σ определяет скорость спадания функции при удалении вектора от центра и называется *шириной окна*, параметр R определяет смещение функции активации по оси абсцисс.

В качестве расстояния между векторами могут быть использованы различные метрики, чаще всего используется евклидово расстояние:

$$S = \sqrt{\sum_{j=1}^{n} (x_j - c_j)^2},$$

где x_j – j-я компонента вектора, поданного на вход нейрона, а c_j – j-я компонента вектора, определяющего положение центра передаточной функции.

Радиальная базисная функция активации имеет максимальное значение 1, когда вход её равен 0. Когда расстояние между x_{ij} и c_{ij} уменьшается, выход увеличивается. Таким образом, радиальный базисный нейрон действует как датчик, который выдает 1 всякий раз, когда вход x_{ij} идентичен его вектору центра радиальной функции c_{ij}. Это позволяет разбить пространства окружностями или (в общем случае) гиперсферами. Соответственно, сети с такими нейронами называются вероятностными и регрессионными.

В реальных сетях функция активации этих нейронов может отражать распределение вероятности какой-либо случайной

НИИ Искусственный интеллект

величины, либо обозначать какие-либо эвристические зависимости между величинами.

Нейроны, использующие радиально-базисные функции, называются радиально симметричными. Они используются для решения задач классификации и являются наиболее эффективными, когда доступно большое количество обучающих векторов.

Нейронные сети радиальных базисных функций имеют множество применений, в том числе функции приближения, прогнозирования временных рядов, классификации и системы управления.

II.2.13. Задания:

1. Напишите формулу мультиквадратичной функции активации.
2. Напишите формулу обратной мультиквадратичной функции активации.
3. Напишите формулу радиально-базисной функции активации.
4. Напишите формулу евклидового расстояния.
5. Напишите формулу функции активации Гаусса.
6. Напишите формулу модифицированной функции активации Гаусса.
7. Опишите действие, выполняемое радиально-базисной функцией активации в реальных сетях.

II.2.13. Вопросы:

1. Что принимает радиально-базисная функция активации в качестве аргумента?
2. Как меняется значение радиально-базисной функции активации в зависимости от близости входного вектора к центру?
3. Какие классы имеются у радиальных базисных функций?
4. Что позволяет сделать применение евклидового расстояния?
5. Какими особенностями характеризуется радиально-базисная нейронная сеть?
6. Какие нейронные сети называются вероятностными и регрессионными?

НИИ искусственный интеллект

7. Для решения каких задач используются нейроны, использующие радиально-базисные функции?

НИИ Искусственный интеллект

II.2.14. Конкурирующая функция активации

В этом параграфе будут рассмотрены конкурирующие функции активации, предложены задания и сформулированы вопросы. При подготовке учебного материала данного параграфа были использованы следующие источники [1-9].

Вход конкурирующей функции активации является вектор n^1, являющийся результатом суммирования вычисленного расстояния с вектором смещения b. Конкурирующая функция активации анализирует значения элементов вектора n^1 и формирует выходы нейронов, равные 0 для всех нейронов, кроме одного нейрона-победителя, имеющего на входе максимальное значение. Таким образом, вектор выхода слоя a^1 имеет единственный элемент, равный 1, который соответствует нейрону-победителю, а остальные равны 0. Если все смещения нулевые, то максимальное значение n^1 не может превышать 0. Нулевое значение n^1 возможно только, когда вектор входа s оказывается равным вектору веса одного из нейронов. Если смещения отличны от 0, то возможны и положительные значения для элементов вектора n^1.

Такая функция активации (активационная характеристика) может быть описана следующим образом:

$$f(s) = \begin{cases} 1, \text{ если } i = i^*, \ i^* = \arg(\max n_i^1) \\ 0, \text{ в противном случае} \end{cases}$$

Заметим, что эта активационная характеристика устанавливается не на отдельный нейрон, а на слой. Поэтому такая активационная характеристика и получила название конкурирующей. Номер активного нейрона i^* определяет ту группу (кластер), к которой наиболее близок входной вектор.

У конкурирующей функции активации областью определения является $D(f)=(-\infty,+\infty)$, а областью значений – $E(f) = (0,+1)$.

Конкурирующие функции активации используются для формирования вероятных и самоорганизующихся нейронных сетей.

НИИ искусственный интеллект

II.2.13. Задания:

1. Напишите формулу определения конкурирующей функции активации.

2. Опишите функциональную работу конкурирующей функции активации.

3. Определите область определения конкурирующей функции активации.

4. Определите область значений конкурирующей функции активации.

5. Опишите активационную характеристику, которая устанавливается не на отдельный нейрон.

6. Напишите назначение номера нейрона $i*$?

7. Опишите ситуацию при которой n^1 получает нулевое значение.

8. Опишите область определения конкурирующей функции активации.

II.2.13. Вопросы:

1. Какой вектор является входом конкурирующей функции активации?

2. Что формирует конкурирующая функция активации после анализа значения элементов вектора n^1?

3. Для решения каких задач используются конкурирующие функции активации?

4. Что будет, если все смещения нулевые?

5. Когда возможны положительные значения для элементов вектора n^1?

6. Почему описанная активационная характеристика получила название конкурирующей?

7. Что будет, если вектор входа s оказывается равным вектору веса одного из нейронов.

8. Для решения какой задачи используются конкурирующие функции активации?

НИИ Искусственный интеллект

II.3. Модели нейронов

II.3.1. Персептроны

В этом параграфе рассмотрены персептроны, обсуждены возможности персептрона, предложены задания и сформулированы вопросы. При подготовке учебного материала данного параграфа были использованы следующие источники [1-9].

Структурная схема и алгоритм работы однонейронного персептрона будут такими же, как у нейрона Мак-Каллока и Питтса.

Персептрон состоит из трёх типов элементов: *S*-элементов, *A*-элементов и одного *R*-элемента. *S*-элементы – это слой сенсоров, или рецепторов. Каждый рецептор может находиться в одном из двух состояний – *покоя* или *возбуждения*, и только в последнем случае он передаёт единичный сигнал в следующий слой, ассоциативным элементам, затем реагирующим элементам. Формально *A*-элементы, как и *R*-элементы, представляют собой сумматоры с порогом, то есть они являются одиночными нейронами. *A*-элементы называются ассоциативными, потому что каждому такому элементу, как правило, соответствует целый набор (ассоциация) *S*-элементов. *A*-элемент активизируется, как только количество сигналов от *S*-элементов на его входе превысило некоторую величину w_0. Таким образом, если набор соответствующих S-элементов располагается на сенсорном поле в форме буквы «Д», А-элемент активизируется, если достаточное количество рецепторов сообщило о появлении «белого пятна света» в их окрестности, то есть А-элемент будет как бы ассоциирован с наличием/отсутствием буквы «Д» в некоторой области.

Сигналы от возбудившихся *A*-элементов, в свою очередь, передаются в сумматор *R*, причём сигнал от *i*-го ассоциативного элемента передаётся с коэффициентом w_i. Этот коэффициент называется весом *A*–*R* связи.

Так же, как и *A*-элементы, *R*-элемент подсчитывает сумму значений входных сигналов, помноженных на веса. *R*-элемент, а

НИИ Искусственный интеллект

вместе с ним и персептрон, выдаёт «1», если линейная форма превышает порог w_0, иначе на выходе будет «−1». Математически, функцию, реализуемую *R*-элементом, можно записать так:

$$y = f(s) = f\left(\sum_{i=1}^{n} w_i x_i - w_0\right),$$

где x_i – *i*-тый входной сигнал, w_i – вес *i*-той синаптической связи, w_0 – смещение (порог), *s* – взвешенная сумма, *y* – выходной сигнал, *f* – пороговая функция активации, *n* – число входов, *i*-тая синаптическая связь является возбуждающей (тормозящей), если значение веса синапса w_i положительное (отрицательное).

Обучение персептрона состоит в изменении весов w_i синаптических связей *A–R*. Веса связей *S–A* (которые могут принимать значения {−1; 0; +1}) и значения порогов *A*-элементов выбираются случайным образом в самом начале и затем не изменяются. После обучения персептрон готов работать в режиме распознавания или обобщения. В этом режиме персептрону предъявляются ранее неизвестные ему объекты, и персептрон должен установить, к какому классу они принадлежат. Работа персептрона состоит в следующем: при предъявлении объекта, возбудившиеся *A*-элементы передают сигнал *R*-элементу, равный сумме соответствующих коэффициентов w_i. Если эта сумма положительна, то принимается решение, что данный объект принадлежит к первому классу, а если она отрицательна – то ко второму. Таким образом функция активации персептрона имеет вид:

1) функция активации, реализуемая *S*- и *A*-элементами:

$$y = f(s) = \begin{cases} 0, & \text{если} \quad s < w_0 \\ 1, & \text{если} \quad s \geq w_0 \end{cases}$$

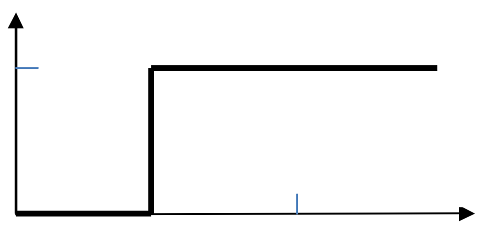


Рисунок II.3.1.2. График функции активации *S*- и *A*-элементов

НИИ Искусственный интеллект

2) функция активации, реализуемая простым *R*-элементом:

$$y = f(s) = \begin{cases} -1, & \text{если} \quad s < 0 \\ 1, & \text{если} \quad s \geq 0 \end{cases}$$

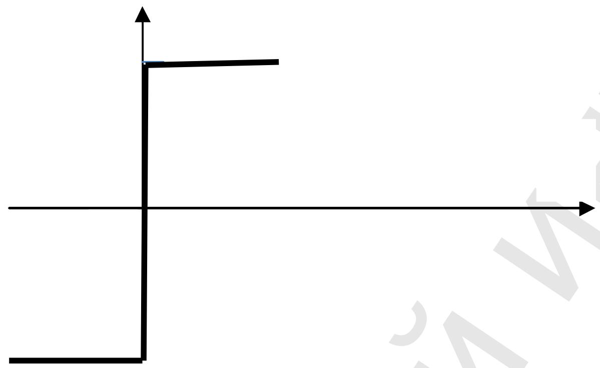


Рисунок II.3.1.2. График функции активации простого *R*-элемента.

Теперь можно привести описание на основе сигналов, приведенное автором персептрона - Ф. Розенблаттом. Для этого сначала определим составные элементы персептрона, которые являются частными случаями искусственного нейрона с пороговой функцией активации:

- простым *S*-элементом (сенсорным) является чувствительный элемент, который от воздействия какого-либо из видов энергии (например, света, звука, давления, тепла и т. п.) вырабатывает сигнал. Если входной сигнал превышает некоторый порог w_0, на выходе элемента получаем +1, в противном случае − 0.
- простым *A*-элементом (ассоциативным) называется логический решающий элемент, который даёт выходной сигнал +1, когда алгебраическая сумма его входных сигналов превышает некоторую пороговую величину w_0 (говорят, что элемент *активный*), в противном случае выход равен нулю.
- простым *R*-элементом (реагирующим, то есть действующим) называется элемент, который выдаёт сигнал +1, если сумма его входных сигналов является строго положительной, и сигнал −1, если сумма его входных сигналов является строго отрицательной.

Если сумма входных сигналов равна нулю, выход считается либо равным нулю, либо неопределённым.

Если на выходе любого элемента мы получаем 1, то говорят, что элемент *активен* или *возбуждён.*

Все рассмотренные элементы называются *простыми*, так как они реализуют *ступенчатые* (*скачкообразные*) *функции*. Розенблатт утверждал также, что для решения более сложных задач могут потребоваться другие виды функций, например, линейные функции.

В результате Розенблатт ввёл следующие определения:

• *Персептрон* представляет собой сеть, состоящую из *S*-, *A*-, *R*-элементов, с переменной матрицей взаимодействия *W* (элементы которой w_{ij} – веса синаптических связей), определяемой последовательностью прошлых состояний активности сети.

• *Персептрон с последовательными связями* есть система, в которой все связи, начиная от элементов с логическим расстоянием *d* от ближайшего *S*-элемента, оканчиваются на элементах с логическим расстоянием *d*+1 от ближайшего *S*-элемента.

• *Простым персептроном* называется любая система, удовлетворяющая следующим пяти условиям:

1) в системе имеется только один *R*-элемент (естественно, он связан со всеми *A*-элементами);

2) система представляет собой персептрон с последовательными связями, идущими только от *S*-элементов к *A*-элементам и от *A*-элементов к *R*-элементам;

3) веса всех связей от *S*-элементов к *A*-элементам (*S*-*A* связей) неизменны;

4) время передачи каждой связи равно либо нулю, либо фиксированной постоянной τ;

5) все активирующие функции *S*-, *A*-, *R*-элементов имеют вид

$u_i(t)=f(a_i(\mathrm{t}))$, где $a_i(t)$ – алгебраическая сумма всех сигналов, поступающих одновременно на вход элемента u_i.

НИКТ

• *Элементарным персептроном* называется простой персептрон, у которого *все элементы − простые*. В этом случае его активизирующая функция имеет вид $c_{ij}(t)=u_i(t\text{-}\tau)\cdot w_{ij}(t)$.

Дополнительно можно указать на следующие концепции, предложенные Розенблатом, и позднее развитые в рамках теории нейронных сетей:

1) *Персептрон с перекрёстными связями* − это система, в которой существуют связи между элементами одного типа (*S*, *A* или *R*), находящиеся на одинаковом логическом расстоянии от *S*-элементов, причем все остальные связи − последовательного типа.

2) *Персептрон с обратной связью* − это система, в которой существует хотя бы одна связь от логически более удалённого элемента к менее удалённому. Согласно современной терминологии такие сети называются рекуррентными.

3) *Персептрон с переменными S-A связями* − это система, в которой снято ограничение на фиксированные связи от *S*-элементов к *A*-элементам. Доказано, что путём оптимизации *S-A* связей можно добиться значительного улучшения характеристик персептрона.

Для анализа свойств персептрона Марвином Минским пришлось переизложить теорию персептронов на язык предикатов. Суть подхода заключалась в следующем:

- множеству сигналов от *S*-элементов была сопоставлена переменная *X*;
- каждому *A*-элементу был сопоставлен предикат φ(*X*), названный частным предикатом;
- каждому *R*-элементу был сопоставлен предикат ψ, зависящий от частных предикатов;
- персептроном было названо устройство, способное вычислять все предикаты типа ψ.

Значения входных сигналов являются двоичными числами, а значения весов синапсов и смещения являются действительными числами. Значение выходного сигнала определяется видом функции активации и может быть действительным или двоичным числом.

Если значение веса синапса w_i положительное (отрицательное), то i-тая синаптическая связь является возбуждающей (тормозящей).

Можно выделить 5 семейств персептронов, обладающих интересными свойствами:

1. *Персептроны, ограниченные по диаметру* – каждая фигура *X*, распознаваемая частными предикатами, не превосходит по диаметру некоторую фиксированную величину.

2. *Персептроны ограниченного порядка* – каждый частный предикат зависит от ограниченного количества точек из *X*.

3. *Персептроны Гамбы* – каждый частный предикат должен быть линейной пороговой функцией, то есть мини-персептроном.

4. *Случайные персептроны* – персептроны ограниченного порядка, где частные предикаты представляют собой случайно выбранные булевы функции.

5. *Ограниченные персептроны* – множество частных предикатов бесконечно, а множество возможных значений коэффициентов a_i конечно.

Классический метод обучения персептрона – это метод коррекции ошибки. Он представляет собой такой вид обучения с учителем, при котором вес связи не изменяется до тех пор, пока текущая реакция персептрона остается правильной. При появлении неправильной реакции вес изменяется на единицу, а знак (+/−) определяется противоположным от знака ошибки.

Допустим, мы хотим обучить персептрон разделять два класса объектов так, чтобы при предъявлении объектов первого класса выход персептрона был положителен (+1), а при предъявлении объектов второго класса − отрицательным (−1). Для этого выполним следующий алгоритм:

1. Случайным образом выбираем пороги для *A*-элементов и устанавливаем связи *S*−*A* (далее они изменяться не будут).

2. Начальные коэффициенты w_i полагаем равными нулю.

3. Предъявляем обучающую выборку: объекты (например, круги либо квадраты) с указанием класса, к которым они принадлежат.

- Показываем персептрону объект первого класса. При этом некоторые *A*-элементы возбудятся. Коэффициенты w_i, соответствующие этим возбуждённым элементам, *увеличиваем* на 1.
- Предъявляем объект второго класса и коэффициенты w_i тех А-элементов, которые возбудятся при этом показе, уменьшаем на 1.

4. Обе части шага 3 выполним для всей обучающей выборки. В результате обучения сформируются значения весов связей w_i.

Теорема сходимости персептрона, описанная и доказанная Ф. Розенблаттом (с участием Блока, Джозефа, Кестена и других исследователей, работавших вместе с ним), показывает, что элементарный персептрон, обучаемый по такому алгоритму, независимо от начального состояния весовых коэффициентов и последовательности появления стимулов всегда приведет к достижению решения за конечный промежуток времени. Более полный и конкретный материал по вопросам обучения персептрона рассмотрен в III.3.1.

II.3.1. Задания:

1. Перечислите составные элементы персептрона и опишите их свойства и назначения.
2. Опишите процесс прохождения входного сигнала через составные элементы персептрона.
3. Напишите формулу однонейронного персептрона.
4. Напишите формулу функции активации персептрона.
5. Дайте описание персептрона на языке предикатов.

II.3.1. Вопросы:

1. Для чего предназначен распределительный слой персептрона?
2. Когда активизируется А-элементы?

3. Какой элемент подсчитывает сумму значений входных сигналов, помноженных на веса?

4. Какое значение выдает персептрон, если линейная форма превышает порог w_0

5. Какого типа значения входных сигналов, весов синапсов и смещения?

НИИ Искусственный интеллект

II.3.2. Сигмоидальные нейроны

В этом параграфе будут рассмотрены сигмоидальные нейроны, показаны примеры, предложены задания, сформулированы вопросы. При подготовке учебного материала данного параграфа были использованы следующие источники [1-9].

Сигмоидальный нейрон имеет такую же структуру, что и модели нейрона Мак-Каллок-Питтса, но имеет дифференцируемую функцию активации, которая может быть выражена в виде сигмоидальной униполярной или биполярной функции. Структура сигмоидального нейрона показана на рисунке II.3.2.1.

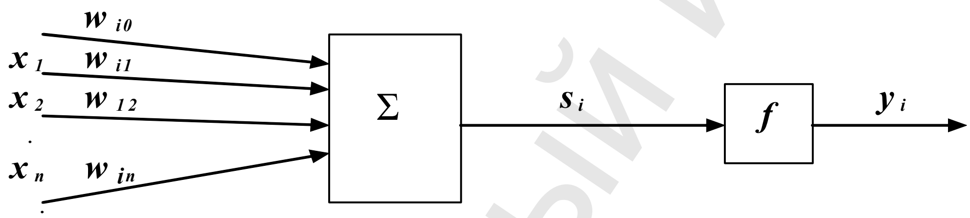


Рисунок II.3.2.1. Структура сигмоидального нейрона.

Униполярная функция активации представляется выражением:

$$y_i = f(s_i) = \frac{1}{1 + e^{-\beta s_i}} \quad (\text{II.3.2.1})$$

График униполярной функции показан на рисунке II.3.2.2.

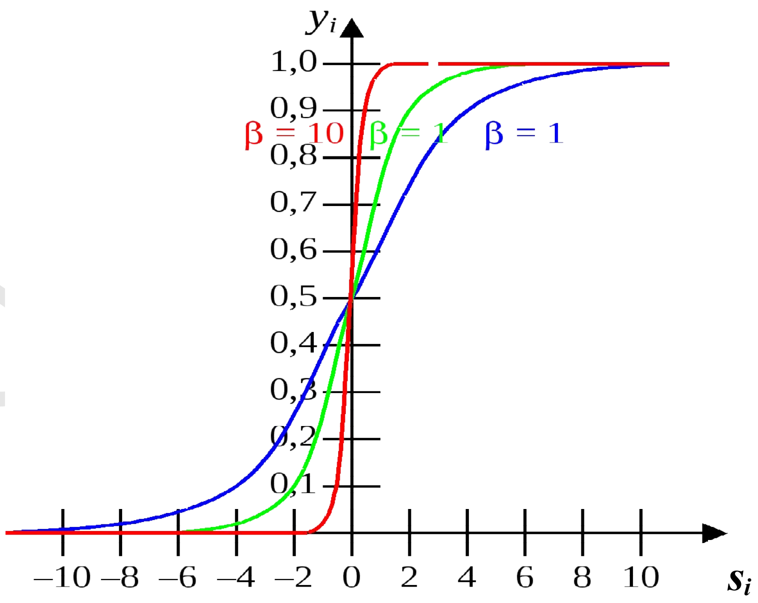


Рисунок II.3.2.2. График униполярной функции.

Биполярная функция активации записывается формулой:

$$y_i = f(s_i) = tanh\ (\beta s_i) = \frac{e^{\beta s_i} - e^{-\beta s_i}}{e^{\beta s_i} + e^{-\beta s_i}} \quad \text{(II.3.2.2)}$$

График биполярной функции показан на рисунке II.3.2.3.

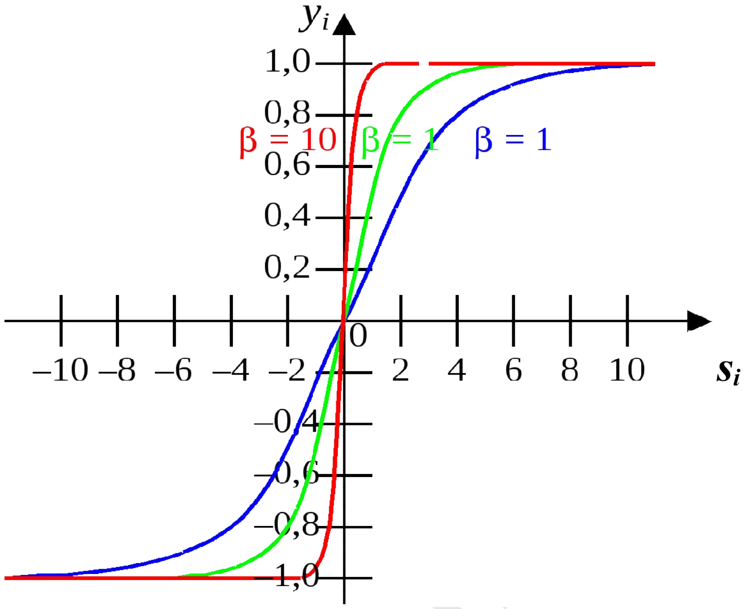


Рисунок II.3.2.3. График биполярной функции.

Значение параметра β подбирается пользователем и влияет на форму функции активации. При $\beta \to \infty$ сигмоидальная функция превращается в функцию ступенчатого типа, идентичную функции активации персептрона. На практике обычно для упрощения используется значение $\beta = 1$.

Важным свойством сигмоидальной функции является ее дифференцируемость. Для униполярной функции производная определяется по формуле:

$$\frac{dy_i}{ds_i} = \beta \cdot f(s_i) \cdot (1 - f(s_i)) = \beta \left(\frac{1}{1 + e^{-\beta s_i}} \right) \cdot \left(1 - \frac{1}{1 + e^{-\beta s_i}} \right) = \frac{\beta e^{-\beta s_i}}{(1 + e^{-\beta s_i})^2},$$

а для биполярной функции:

$$\frac{dy_i}{ds_i} = \beta \cdot (1 - f^2(s_i)) = \beta \cdot \left(1 - \left(\frac{e^{\beta s_i} - e^{-\beta s_i}}{e^{\beta s_i} + e^{-\beta s_i}} \right)^2 \right) = \frac{4\beta}{e^{2\beta s_i} + e^{-2\beta s_i} + 2}$$

Графики производных сигмоидальных функций имеют колоколообразный вид.

График производной униполярной сигмоидальной функции показан на рисунке II.3.2.4.

.

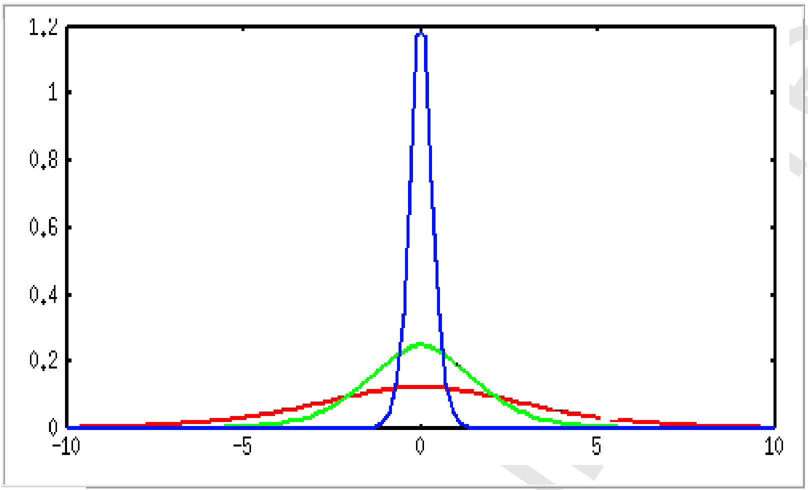


Рисунок II.3.2.4. График производной униполярной сигмоидальной функции

График производной биполярной сигмоидальной функции показан на рисунке II.3.2.5.

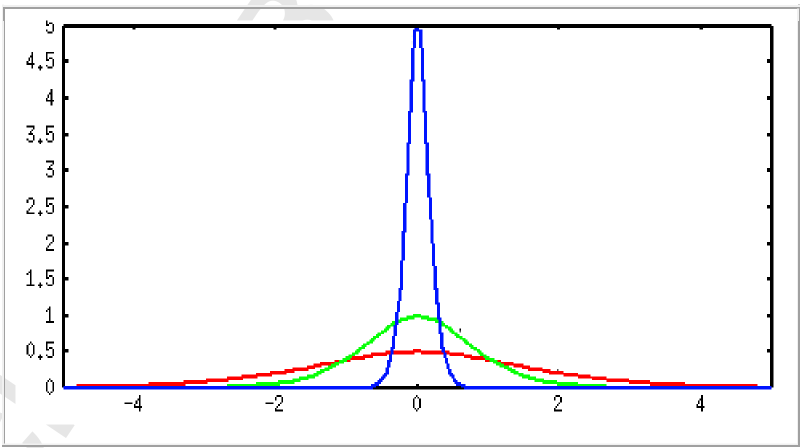


Рисунок II.3.2.5. График производной биполярной сигмоидальной функции.

Для обучения сигмоидального нейрона используется парадигма (стратегия) "с учителем" с применением градиентных методов,

НИИ Интеллект

позволяющих минимизировать целевую функцию обучения $E(W_i)$. Здесь в отличие от персептрона, для поиска минимума целевой функции используются методы поисковой оптимизации первого порядка, в которых целенаправленное изменение весовых коэффициентов w_{ij} осуществляется в направлении отрицательного градиента:

$$E(W_i)=\frac{1}{2}\sum_{k=1}^{r}(y_i^k-d_i^k)^2=\frac{1}{2}\sum_{k=1}^{r}(\sum_{i=0}^{m}w_{ij}^k x_i^k-d_i^k)^2\to\min_w \qquad (II.3.2.3)$$

где $y_i^k=f(s_i^k)=f(\sum_{i=0}^{m}w_{ij}^k x_i^k))$ является сигмоидальной функцией; x – входной вектор, $x=[x_0, x_1,\ldots, x_n]^T$ со значением $x_0=1$ при наличии поляризации и $x_0=0$ при ее отсутствии, а d_i – соответствующее ему ожидаемое значение на выходе i-го нейрона.

Применение непрерывной функции активации позволяет использовать при обучении градиентные методы. Проще всего реализовать метод наискорейшего спуска, в соответствии с которым уточнение вектора весов $w=[w_0, w_1,\ldots, w_n]^T$ проводится в направлении отрицательного градиента целевой функции. Если эта функция определена выражением (II.3.2.3), то j-я составляющая градиента будет иметь вид:

$$\nabla_j E(w_i)=\frac{dE(w_i)}{d\,w_{ij}}=\sum_{k=1}^{r}\left(y_i^k-d_i^k\right)\cdot\frac{d\,y_i^k}{d\,w_{ij}}=\sum_{k=1}^{r}\left(y_i^k-d_i^k\right)\cdot\frac{d\,f(s_i^k)}{d\,s_i^k}x_j^k \ ,$$

где уточнение весов $\boldsymbol{w}_{ij}$ по методу наискорейшего спуска проводится в направлении отрицательного градиента целевой функции.

Обозначим $\delta_i^k=\left(y_i^k-d_i^k\right)\cdot\frac{df(s_i^k)}{d\,s_i^k}$, имеем $\frac{dE(W_i)}{d\,w_{ij}}=\sum_{k=1}^{r}\delta_i^k\cdot x_j^k$.

Также возможно обучение сигмоидального нейрона и дискретным способом – сериями циклов уточнения входных весов

НИИ Интеллектуальный ...

для каждой эталонной пары $<X^k, d_i^k>$ (см. правило персептрона). При этом коррекция весов после каждого цикла выполняется по следующей формуле:

$$w_{ij}(t+1) = w_{ij}(t) - \eta \delta_i^k(t) x_j^k,$$

где η – коэффициент обучения, , значения которого выбирают либо эмпирически из интервала (0, 1).

Изменение весов можно осуществлять и решением следующего разностного уравнения:

$$\frac{dw_{ij}(t)}{dt} = -\mu \delta_i x_j,$$

где $\mu \in (0, 1)$ - коэффициент обучения.

Надо напомнить, что все методы поисковой оптимизации 1-го порядка – это методы локального поиска, не гарантирующие достижения глобального экстремума. Для преодоления этого недостатка было предложено обучение с моментом, в котором коррекция весов выполняется следующим образом:

$$w_{ij}(t+1) = w_{ij}(t) - \eta \cdot \delta_i^k(t) \cdot x_j^k + \alpha \cdot (w_{ij}(t) - w_{ij}(t)).$$

Последнее слагаемое в формуле называется моментом и характеризует фактическое изменение веса в предыдущем цикле (α выбирается в диапазоне (0, 1)). Существует надежда, что при приближении к точке локального минимума (где градиентная составляющая $\delta_i^k(t).x_j^k$ стремится к нулю) составляющая момента выведет поиск из области локального минимума в более перспективную область.

Следует отметить, что применение градиентных методов обучения нейрона гарантирует достижение только локального экстремума, который для полимодальной $E(w)$ может быть достаточно далек от глобального минимума. Выход из его окрестности при использовании простого алгоритма наискорейшего спуска невозможен, поэтому на практике применяют обучение с разбросом или моментом.

Здесь процесс уточнения весов определяется не только информацией о градиенте функции, но также и фактическим трендом изменения весов:

$$\Delta w_{ij}(t+1) = -\eta \delta_i x_j + \alpha \Delta w_{ij}(t)$$

где $\alpha \in (0, 1)$ – коэффициент момента, или использование стохастических методов оптимизации.

Влияние момента на подбор весов увеличивается с ростом значения α, поэтому следует выбирать α таким, чтобы момент не доминировал в процессе обучения, так как это приводит к нестабильности (расходимости) алгоритма обучения.

II.3.2. Задания:

1. Напишите формулу униполярной функции активации.
2. Напишите формулу биполярной функции активации.
3. Напишите формулу вычисления производной униполярной функции активации.
4. Напишите формулу вычисления производной биполярной функции активации.
5. Напишите формулу для *j*-ой составляющей градиента.
6. Напишите формулу обучение сигмоидального нейрона дискретным способом.
7. Напишите формулу обучение сигмоидального нейрона непрерывным способом.

II.3.2. Вопросы:

1. Как выглядит график униполярной функции активации?
2. Как выглядит график производной униполярной функции активации?
3. Как выглядит график биполярной функции активации?
4. Как выглядит график производной биполярной функции активации?
5. Какая парадигма используется для обучения сигмоидального нейрона?
6. Что гарантирует градиентный метод обучения нейрона?
7. В связи с чем увеличивается подбор весов?

НИУ «Искусственный интеллект»

II.3.3. Паде-нейроны

В этом параграфе рассмотрены паде-нейроны, предложены задания и сформулированы вопросы. При подготовке учебного материала данного параграфа были использованы источники [1-9].

Паде-нейрон состоит из двух сумматоров, одного элемента, вычисляющего частное от взвешенных сумм, и одного блока функционального преобразователя. Свое название паде-нейрон получил от паде-метода аппроксимации рациональной функций.

Структура паде-нейрона показана на рисунке II.3.3.

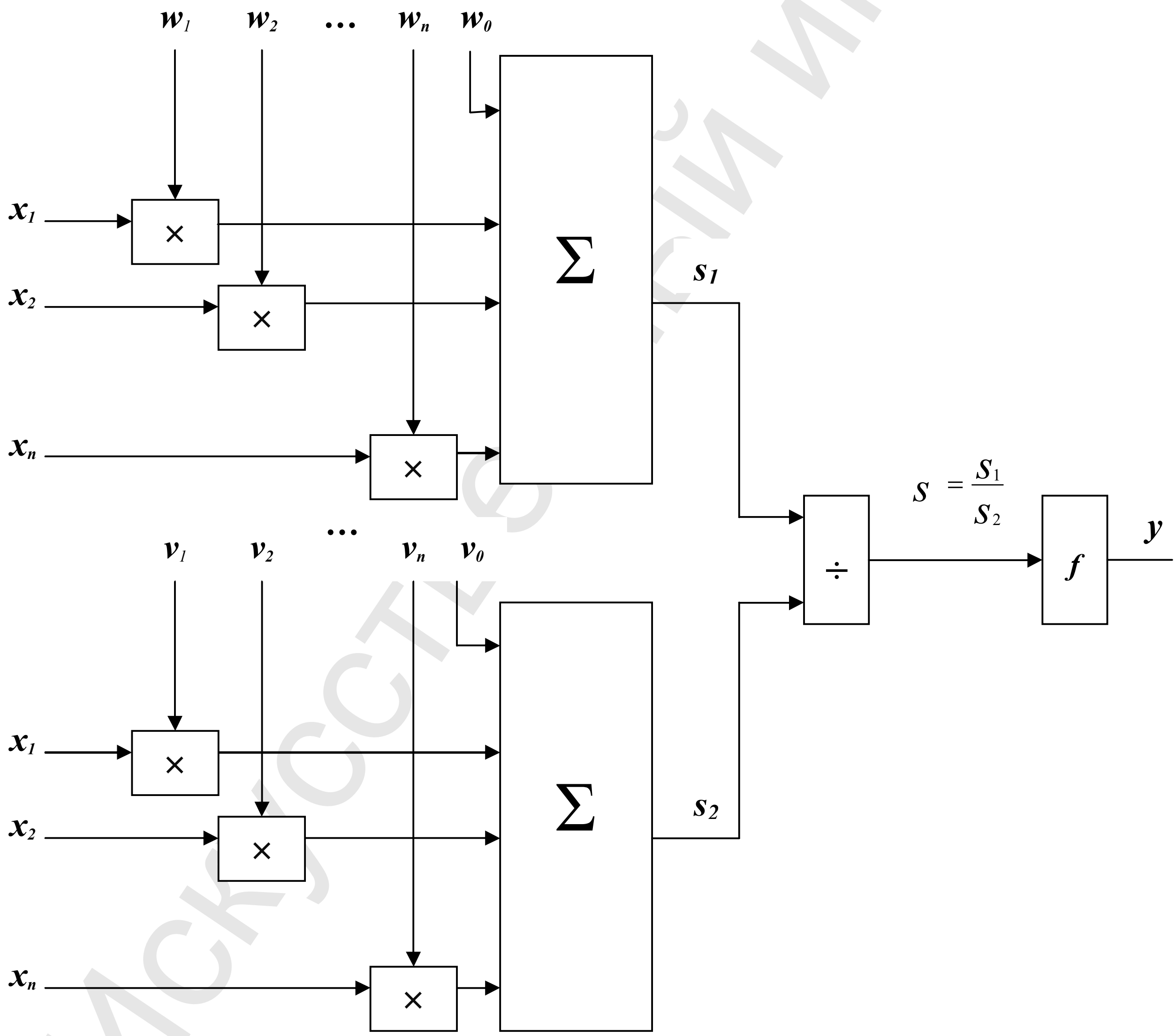


Рисунок II.3.3. Структурная схема паде-нейрона.

На практике такая модель практически не используется, т.к. она усложняет процесс построения нейросетевой модели, не привнося значительных дополнительных возможностей.

НИИ искусственный интеллект

Уровень активации в этом нейроне определяется формулой:

$$s=\frac{\sum_{i=0}^{n} w_i x_i}{\sum_{i=0}^{n} v_i x_i}.$$

Минимизация квадратичной ошибки в этом случае представлена выражением.

$$E(w)=\sum_{i=1}^{n}\left(a\left(\frac{w_i x_i}{v_i x_i}\right)-y_i\right)^2\to\min_{w}.$$

Математическая модель паде-нейрона представляется в виде:

$$y=f(s)=f\left(\frac{s_1}{s_2}\right)=f\left(\frac{w_0+\sum_{i=1}^{n} w_i x_i}{v_0+\sum_{i=1}^{n} v_i x_i}\right),$$

где w_i – веса первого сумматора, w_0 – начальное состояние первого сумматора, v_i – веса второго сумматора, v_0 – начальное состояние второго сумматора, т.е. количество настраиваемых весов равно *2n.*

В паде-нейроне количество настраиваемых весов в два раза больше, чем в классическом нейроне, т.е. равно *2n*.

Паде-нейрон может использоваться как обобщение нейрона типа "адалайн" в тех случаях, когда линейных функций становится недостаточно.

Задания II.3.3.

1. Напишите формулу вычисления уровня активации паде-нейрона.
2. Напишите формулу минимизации квадратичной ошибки паде-нейрона.
3. Напишите математическую модель паде-нейрона.

Вопросы II.3.3.

1. Из чего состоит паде-нейрон?
2. Сколько настраиваемых весов у паде-нейрона?

НИИ Искусственный интеллект

3. Для чего и в каких случаях используется паде-нейрон?

НИИ Искусственный интеллект

II.3.4. Нейроны с квадратичным сумматором

В этом параграфе рассмотрены нейроны с квадратичным сумматором, обсуждены его возможности, предложены задания и сформулированы вопросы. При подготовке учебного материала данного параграфа были использованы источники [1-9].

Нейроны с квадратичным сумматором относятся к нестандартной модели искусственного нейрона. На функциональный преобразователь нейрона подают взвешенную квадратичную форму входного сигнала. Структура нейрона с квадратичным сумматором представлена на рисунке II.3.4.

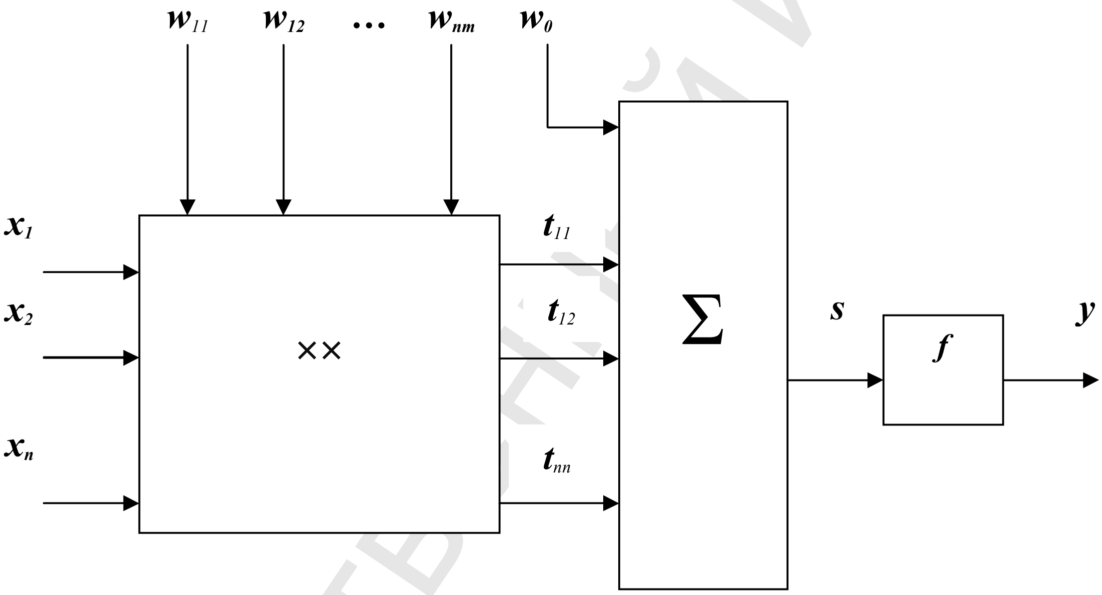


Рисунок II.3.4. Структура нейрона с квадратичным сумматором

В нейроне с квадратичным сумматором сначала квадратичные формы из комбинаций пар входных сигналов умножаются на весв синаптических связей, затем взвешенные квадратичные формы z_{ij} и значение сигнала начального состояния w_0 суммируются, что описывается следующим выражением:

$$s = w_0 + \sum_{i=1}^{n} v_i \ x_i + \sum_{i=1}^{n} \sum_{j=1}^{m} w_{ij} \ x_i x_j \ .$$

У нейрона с квадратичным сумматором на выходной блок подается взвешенная квадратичная форма входного сигнала и эта взвешенная сумма преобразуется в соответствии с функции f и вырабатывается значение выходного сигнала $y = f(s)$.

Математическую модель нейрона с квадратичным сумматором можно представить в виде:

$$y=f(s)=f(w_0+\sum_{i=1}^{n}v_i\ x_i\ +\sum_{i=1}^{n}\sum_{j=1}^{m}w_{ij}\ x_i x_j\ )$$

В нейроне с квадратичным сумматором количество настраиваемых весов равно $n(n+1)/2$.

Для многомерных нормальных распределений нейрон с квадратичным сумматором является наилучшим классификатором. Минимум вероятности ошибки дает квадратичная разделяющая поверхность:

– если $f(s)>0$, то объект принадлежит первому классу;

– если $f(s)\leq 0$, то объект принадлежит второму классу (при условии правильного выбора коэффициентов Q(x)).

Квадратичная ошибка здесь определяется как:

$$E(w_{ij},v_i,w_0)=\frac{e^2}{2}=\frac{(d-f(s))^2}{2}$$

Коэффициенты квадратичного сумматора уточняются, исходя из определения квадратичной ошибки по формулам:

$$w_{ij}(t+1)=w_{ij}(t)+2\alpha\cdot e\cdot x_i\cdot x_j,$$

$$v_i(t+1)=v_i(t)+\alpha\cdot e\cdot x_i,$$

$$w_0(t+1)=w_0(t)+\alpha\cdot e,$$

где $\alpha\in(0, 1)$ – коэффициент обучения.

Нейрон с квадратичным сумматором является наилучшим классификатором среди искусственных нейронов для многомерных

НИИ Искусственный интеллект

нормальных распределений. Недостаток такого классификатора - большое число настраиваемых параметров.

Можно разработать и другие модели искусственного нейрона. Однако при любом предположении модель сети взаимодействующих нейронов оказывается исключительно богатой и обладающей свойствами, которые можно сопоставить с реальными возможностями мозга.

При составлении сети с квадратичными сумматорами из простых элементов на пользователя ложится большой объем работы по проведению связей и организации вычисления произведений.

Задания II.3.4.

1. Укажите состав и нарисуйте структуру нейрона с квадратичным сумматором.
2. Напишите математическую модель нейрона с квадратичным сумматором.
3. Напишите формулу вычисления квадратичной ошибки для нейрона с квадратичным сумматором.
4. Опишите способ уточнения коэффициентов сумматора для нейрона с квадратичным сумматором.
5. Укажите достоинства и недостатки нейрона с квадратичным сумматором.

Вопросы II.3.4.

1. Сколько настраиваемых весов в нейроне с квадратичным сумматором?
2. Является ли наилучшим классификатором нейрон с квадратичным сумматором?
3. По каким формулам уточняются коэффициенты классификатора на основе нейрона с квадратичным сумматором?
4. Что является наилучшим классификатором для многомерных нормальных распределений?

НИИ искусственный интеллект

5. В чем заключается недостаток классификатора – нейрона с квадратичным сумматором?

НИИ Искусственный интеллект

II.3.5. Адаптивные линейные нейроны

В этом параграфе рассмотрены адаптивные линейные нейроны, обсуждены его возможности, предложены задания и сформулированы вопросы. При подготовке учебного материала данного параграфа были использованы источники [1-9].

Адаптивные линейные нейроны – ADAptive LInear NEuron (или нейрон типа «адалайн» – «adaline») были предложены Б. Уидроу (B.Widrow). Они относятся к нестандартной модели искусственного нейрона. По методу весового суммирования сигналов нейрон типа «адалайн» аналогичен модели Мак-Каллок-Питтса, но в качестве функции активации использует сигнатурную (знаковую) функцию, т.е. у адаптивного линейного нейрона функцией активацией является функция типа «сигнум» - «signum»:

$$y_i = f(s_i) = \begin{cases} 1, & \text{если} \quad s_i > 0 \\ -1, & \text{если} \quad s_i \le 0 \end{cases}$$

Структура адаптивного линейного нейрона представлена на рисунке II.3.5.

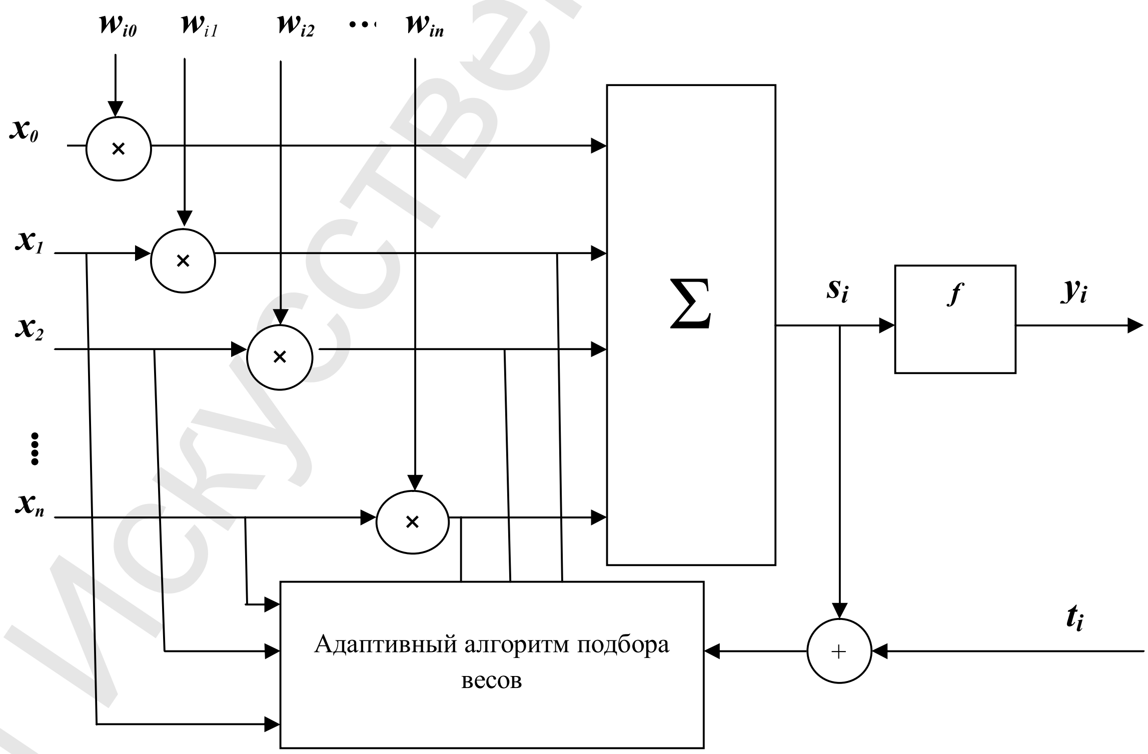


Рисунок II.3.5. Структура адаптивного линейного нейрона.

В данном нейроне адаптивный подбор весов осуществляется в процессе минимизации квадратичной ошибки:

$$E(w)=\frac{1}{2}(d_i-\sum_{j=1}^{n} w_{ij}x_j)^2 \to \min_{w} ,$$

где d_i – ожидаемые выходные сигналы нейрона.

В адаптивном алгоритме подбора весов Уидроу целевая функция является непрерывной, поэтому для ее минимизации применяется метод наискорейшего спуска. При этом веса уточняются по формуле:

$$w_{ij}(t+1)=w_{ij}(t)+\eta e_i x_i$$

где η - темп обучения, $e_i=d_i-\sum_{j=1}^{n} w_{ij}x_j$

Задания II.3.5.

1. Напишите формулу функции активации адаптивного линейного нейрона.
2. Нарисуйте структуру адаптивного линейного нейрона.
3. Напишите формулу процесса минимизации квадратичной ошибки для подбора весов адаптивного линейного нейрона.
4. Напишите формулу целевой функции для адаптивного линейного нейрона.
5. Напишите формулу уточнения весов для адаптивного линейного нейрона.

Вопросы II.3.5.

1. Кем был предложен адаптивный линейный нейрон?
2. Какой модели аналогичен нейрон типа «адалайн» по методу весового суммирования сигналов?
3. Какую функцию активации использует адаптивный линейный нейрон?
4. Дифференцируема ли функция активации в адаптивном линейном нейроне?
5. Какую роль выполняет структурный элемент «Адаптивный алгоритм подбора весов»?

НИИ искусственный интеллект

II.3.6. Сигма-пи нейроны

В этом параграфе рассмотрены сигма-пи нейроны, обсуждены их возможности, предложены задания и сформулированы вопросы. При подготовке учебного материала данного параграфа были использованы источники [1-9].

Сигма-Пи нейроны относятся к нестандартной модели искусственного нейрона и являются обобщением нейронов с линейной и квадратичной функцией активации на случай представления функции активации *s* полиномом степени *n* (*n* – число входов нейрона) и представляется следующим выражением:

$$s = \sum_{k=1}^{m} w_k \prod_{i \in I_k} x_i + \theta,$$

где I_k – множество индексов, содержащее одну из возможных 2^n комбинаций первых *n* целых чисел, $m = 2^n$.

Сигма-Пи нейроны используют в качестве функции активации пороговую - знаковую функцию:

$$y = \mathrm{sgn}(s) = \begin{cases} 1, & \text{если} \quad s \geq 0 \\ 0, & \text{если} \quad s < 0 \end{cases}$$

где $y \in \{0,1\}$, $X \in \{0,1\}^n$, $\theta \in Z$, $W \in Z$ (Z – скалярное произведение), $W \cdot X$ – скалярное произведение векторов *W* и *X*.

Сигма-Пи нейроны используются для решения задач диагностики, прогнозирования и распознавания образов.

Задания II.3.6.

1. Напишите формулу для Сигма-Пи нейронов.
2. Напишите функцию активации Сигма-Пи нейронов.

Вопросы II.3.6.

1. Расширением чего являются Сигма-Пи нейроны?
2. Сколько комбинаций содержит множество индексов?
3. Для решения каких задач применяются Сигма-Пи нейроны?

НИИ искусственный интеллект

II.3.7. Нейроны типа WTA

В этом параграфе рассмотрены нейроны типа WTA, обсуждены их возможности, предложены задания и сформулированы вопросы. При подготовке учебного материала данного параграфа были использованы источники [1-9].

Нейроны типа WTA (Winner Takes All - Победитель получает все) относятся к нестандартной модели искусственного нейрона. Нейроны данного типа используются группами, в которых конкурируют между собой и имеют входной модуль стандартного сумматора, вычисляющего сумму входных сигналов с соответствующими весами $\boldsymbol{w_{ij}}$.

Структура группы нейронов типа «Победитель получает все» представлена на рисунке II.3.7.

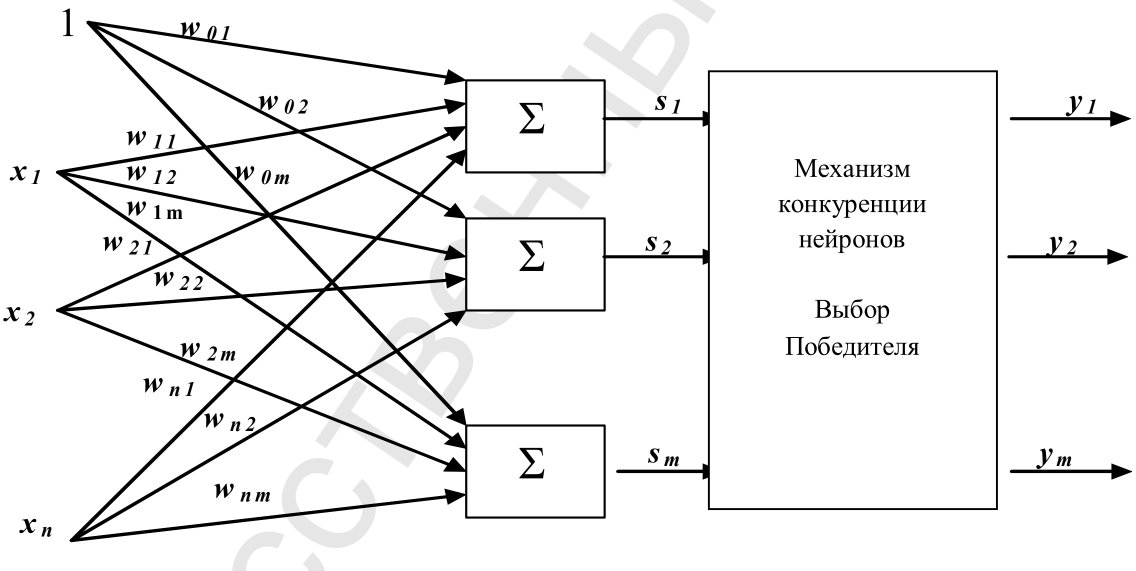


Рисунок II.3.7. Схема соединения нейронов типа WTA

Каждый конкурирующий нейрон в группе получает одни и те же входные сигналы. Каждый нейрон рассчитывает выходной сигнал своего сумматора обычным образом. Выходной сигнал j-го сумматора определяется согласно формуле, $j=1,2,\ldots,m$:

$$s_j = f(\sum_{i=0}^{n} w_{ij} x_i)$$

По результатам сравнения всех $\boldsymbol{s_j}$ выбирается нейрон-победитель, обладающий наибольшим значением $\boldsymbol{s_j}$. Выходной

НИИ Искусственный интеллект

сигнал y_j нейрона-победителя получает значение 1, выходные сигналы всех остальных (проигравших) нейронов получат значение 0 и у них блокируется процесс уточнения весов.

Для обучения нейронов типа WTA не требуется учитель. Это обучение аналогично обучению Instar Гроссберга. Начальные значения весов всех нейронов выбираются случайным образом с последующей нормализацией относительно 1. При предъявлении каждого обучающего вектора X^k определяется нейрон-победитель, что дает ему право уточнить свои веса по упрощенному правилу:

$$w_{ij}(t+1)=w_{ij}(t)+\eta(x_j^k-w_{ij}(t))$$

Все проигравшие нейроны оставляют свои веса синаптических связей неизменными. В каждом цикле обучения побеждает тот нейрон, чей текущий вектор входных весов W_i наиболее близок входному вектору X^k. При этом вектор W_i корректируется в сторону вектора X^k. Поэтому в ходе обучения каждая группа близких друг другу входных векторов (кластер) обслуживается отдельным нейроном.

Задания II.3.7.

1. Нарисуйте схему соединения нейронов типа «Победитель получает все».
2. Напишите формулу вычисления выходного сигнала j-го сумматора.
3. Напишите формулу изменение весов для нейрона типа «Победитель получает все»?

Вопросы II.3.7.

1. По какой причине нейроны типа «Победитель получает все» используются группами?
2. Какая парадигма обучения используется для нейрона типа «Победитель получает все»?
3. Что делают с синаптическими весами все проигравшие нейроны?

НИИ искусственный интеллект

II.3.8. Нейроны Хебба

В этом параграфе рассмотрены нейроны Хебба, обсуждены их возможности, предложены задания и сформулированы вопросы. При подготовке учебного материала данного параграфа были использованы источники [1-9].

В своих исследованиях Д. Хебб рассматривал стандартную модель нейрона, структура которой показана на рисунке II.3.8.

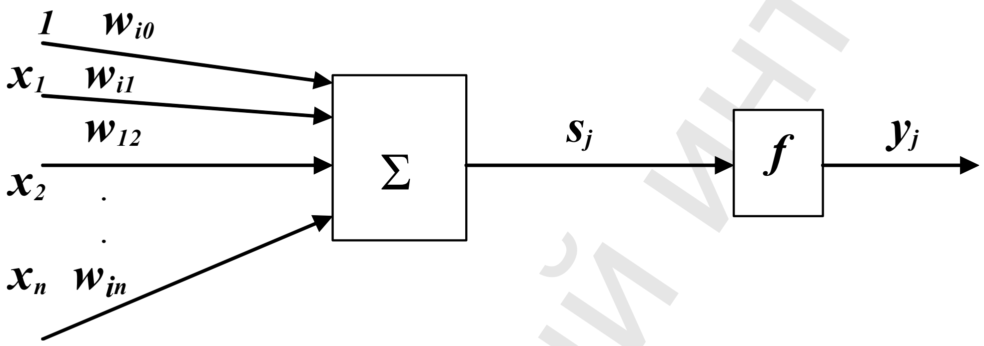


Рисунок II.3.8. Структура стандартной модели нейрона.

Он заметил, что связь между двумя нейронами усиливается, если оба нейрона становятся активными в один и тот же момент времени. Если j-й нейрон с выходным сигналом y_j связан с i-м нейроном, имеющим выходной сигнал y_i, связью с весом w_{ij}, то на силу связи этих клеток влияют значения выходных сигналов y_i и y_j. Для моделирования этого процесса Д. Хебб предложил формальное правило, в котором вес нейрона w_{ij} изменяется пропорционально произведению его входного и выходного сигналов:

$$\Delta w_{ij} = \eta y_j y_i ,$$

где η – коэффициент обучения, значение которого лежит в интервале (0, 1).

Обучение нейрона по правилу Хебба может проводиться как с учителем, так и без него. Во втором случае в правиле Хебба используется фактическое значение y_i выходного сигнала нейрона. При обучении с учителем вместо значения выходного сигнала $\boldsymbol{y_i}$ используется ожидаемая от этого нейрона реакция d_i. В этом случае правило Хебба записывается в виде:

$$\Delta w_{ij}=\eta y_j d_i$$

В результате применения правила Хебба веса w_{ij} могут принимать произвольно большие значения, поскольку в каждом цикле обучения происходит суммирование текущего значения веса и его приращения $\boldsymbol{\Delta w_{ij}}$:

$$w_{ij}(t+1)=w_{ij}(t)+\Delta w_{ij}$$

Один из способов стабилизации процесса обучения по правилу Хебба состоит в учете последнего значения w_{ij}, уменьшенного на коэффициент забывания γ. При этом правило Хебба имеет вид:

$$w_{ij}(t+1)=w_{ij}(t)(1-\gamma)+\Delta w_{ij}$$

Значение коэффициента забывания γ выбирается из интервала (0, 1) и составляет некоторый процент от коэффициента обучения η. Применение больших значений γ приводит к тому, что нейрон забывает значительную часть того, чему он обучился в прошлом. Рекомендуемые значения коэффициента забывания – $\gamma < 0{,}1\eta$, при которых нейрон сохраняет большую часть информации, накопленной в процессе обучения, и получает возможность стабилизировать значения весов на определенном уровне.

Правило Хебба может применяться для нейронных сетей различных типов с разнообразными функциями активации моделей отдельных нейронов.

Задания II.3.8.

1. Нарисуйте структуру стандартной модели нейрона.
2. Напишите формулу изменение весов, предложенную для двух активных нейронов.
3. Напишите формулу правило Хебба при обучении с учителем.

Вопросы II.3.8.

1. Что происходит между двумя активными нейронами?
2. Что происходит в каждом цикле обучения?

3. Для каких нейронов применяется правило Хебба?

II.3.9. Звезды Гроссберга

В этом параграфе рассмотрены звезды Гроссберга, обсуждены их возможности, предложены задания и сформулированы вопросы. При подготовке учебного материала данного параграфа были использованы источники [1-9].

Звезды Гроссберга являются моделью отдельных участков биологического мозга и относятся к нестандартной модели искусственного нейрона. Они используются как компоненты нейронных сетей для решения задач распознавания образов.

Нейрон в форме **входная звезда** – Input Star (Instar) имеет ***n*** входов, которым соответствуют веса синаптических связей, и один выход *y*. Структура входной звезды приведена на рисунке II.3.9.1.

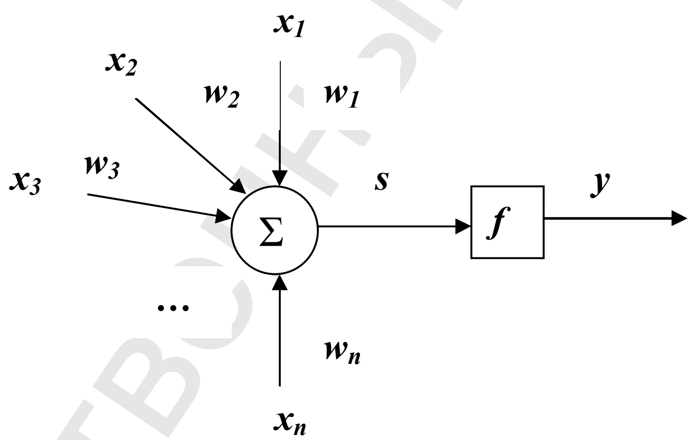


Рисунок II.3.9.1. Входная звезда

Входная звезда обучается выдавать сигнал на выходе всякий раз, когда на входы поступает определенный вектор, т.е. она является своего рода детектором состояния входов и реагирует только на свой входной вектор. На выходе нейрона формируется взвешенная сумма ее входов, представляющая свертку входного вектора с весовым вектором, и в соответствии с функцией активации вырабатывается выходной сигнал :

$$y = f(s) = f(W \cdot X) = f(\sum_{i=1}^{n} w_i x_i) \quad ,$$

НИИ искусственного интеллекта

где $X=(x_1, x_2,..., x_n)$ – входной вектор; $W=(w_1, w_2,..., w_n)$ – весовой вектор, $\boldsymbol{y}$ – выходной сигнал.

Часто в Instar применяется линейная форма функции активации, и тогда $y=s$. Обучение Instar – подбор весов w_{ij} производится по правилу Гроссберга, в соответствии с которым:

$$w_{ij}(t+1)=w_{ij}(t)+\eta y_i(x_j-w_{ij}(t))$$

где $\eta \in (0,1)$ – коэффициент обучения.

Входные данные, представляемые в виде вектора x, выражены чаще всего в нормализованной форме, в которой $|\boldsymbol{x}|=1$. Нормализация компонентов вектора x выполняется по формуле:

$$x_j^*=\frac{x_j}{\sqrt{x_1^2+x_2^2+...+x_n^2}}$$

Результаты обучения по методу Гроссберга в значительной степени зависят от коэффициента обучения η. При выборе $\eta = 1$ веса w_{ij} становятся равными значениям x_j уже после первой итерации. Ввод очередного входного вектора x вызовет адаптацию весов к новому вектору и абсолютное «забывание» предыдущих значений. Выбор $\eta < 1$ приводит к тому, что в результате обучения весов синаптических связей w_{ij} принимают усредненные значения обучающих векторов x.

Входная звезда обладает возможностью реагировать на незначительные изменения единичного вектора. Это достигается настройкой весов в процессе обучения таким образом, чтобы усреднить величины обучающих векторов, с целью реакции на любой вектор этого класса.

Нейрон в форме **выходной звезды** – Output Star (Outstar) имеет один вход и m выходов с весами.

Выходная звезда выполняет противоположную функцию по отношению входной звезды – при поступлении сигнала на вход выдается определенный выходной вектор с весами:

$$\sum_{i=1}^{n} y_i w_i = Y \cdot W = x ,$$

где x – входной сигнал; $Y=(y_1, y_2,..., y_m)$ – выходной вектор; $W=(w_1, w_2,..., w_m)$ – весовой вектор, которому в начале присваивается вектор случайных значений из интервала (0,1).

Структура выходной звезды приведена на рисунке II.3.9.2.

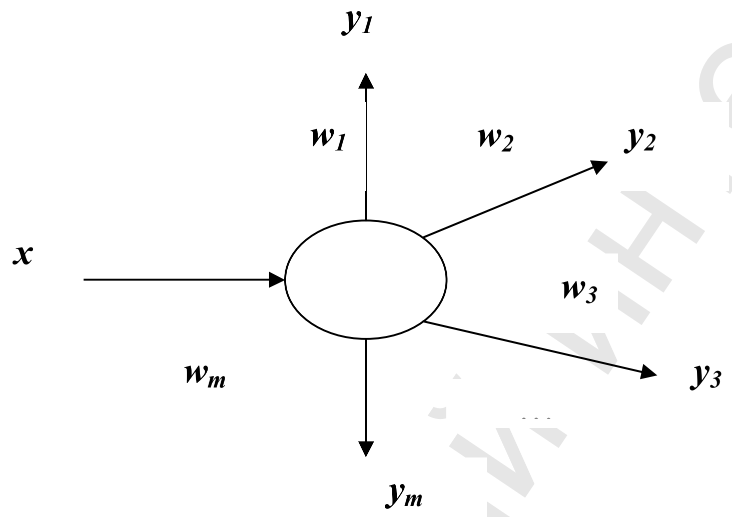


Рисунок II.3.9.2. Выходная звезда

Обучение Outstar по правилам Гроссберга представлено следующим выражением:

$$w_{ij}(t+1)=w_{ij}(t)+\eta(y_i-w_{ij}(t)) \,,$$

где $w_i(t)$ и $w_i(t+1)$ – веса i-й звезды на t-м и $(t+1)$-м шаге обучения, соответственно; η – нормирующий коэффициент обучения, который в начале приблизительно равен 1 и постепенно уменьшается до нуля в процессе обучения.

Особенностью нейронов в форме звезды Гроссберга является избирательность памяти. Каждый нейрон в форме входной звезды помнит «свой» относящийся к нему образ и игнорирует остальные. Каждой выходной звезде соответствует некоторая конкретная командная функция. Образ памяти связывается с определенным нейроном, а не возникает вследствие взаимодействия между нейронами в сети.

Входные и выходные звезды могут соединяться между собой в НС любой сложности и использоваться в отдельности во многих нейронных сетях. В частности, конфигурации входных и выходных звезд Гроссберга используются для создания иерархических нейронных сетей.

Задания II.3.9.

1. Нарисуйте структуру входной звезды, предварительно определив число входов и выходов.

2. Напишите формулу вырабатывания входной звездой выходного сигнала в соответствии с функцией активации.

3. Напишите формулу подбора весов синаптических связей по правилу Гроссберга для входной звезды.

4. Напишите формулу нормализации компонентов входного вектора для входной звезды.

5. Нарисуйте структуру выходной звезды, предварительно определив число входов и выходов.

6. Напишите формулу вырабатывания выходной звездой выходного сигнала в соответствии с функцией активации.

7. Напишите формулу обучения выходной звезды по правилам Гроссберга.

8. Какую функцию выполняет выходная звезда?

Вопросы II.3.9.

1. Чем является звезды, предложенные Гроссбергом для биологического мозга?

2. Как используются звезды Гросберга для решения задач распознавания образов?

3. Когда входная звезда является детектором состояния входов и реагирует только на свой входной вектор?

4. От какого параметра зависит результаты обучения по методу Гроссберга входной звезды?

5. Когда происходит адаптация весов к новому вектору и абсолютное «забывание» предыдущих значений.

6. Какой возможностью обладает входная звезда?

7. Могут ли соединяться между совбой входные и выходные звезды и использоваться в отдельности?

8. Что соответствует каждой выходной звезде?

НИИ Искусственный интеллект

II.3.10. Нейроны со счетчиком совпадений

В этом параграфе рассмотрены нейроны со счетчиком совпадений, обсуждены возможности таких нейронов, предложены задания и сформулированы вопросы. При подготовке учебного материала данного параграфа были использованы источники [1-9].

Нейроны со счетчиком совпадений относятся к нестандартной модели искусственного нейрона. Такой нейрон, получая на вход *n*-мерный вектор *X*, на выходе выдает целое число, равное числу совпадений x_i с a_i, т.е. число тех *i*, для которых $x_i = a_i$.

Общая схема нейрона со счетчиком совпадений показана на рисунке II.3.10.

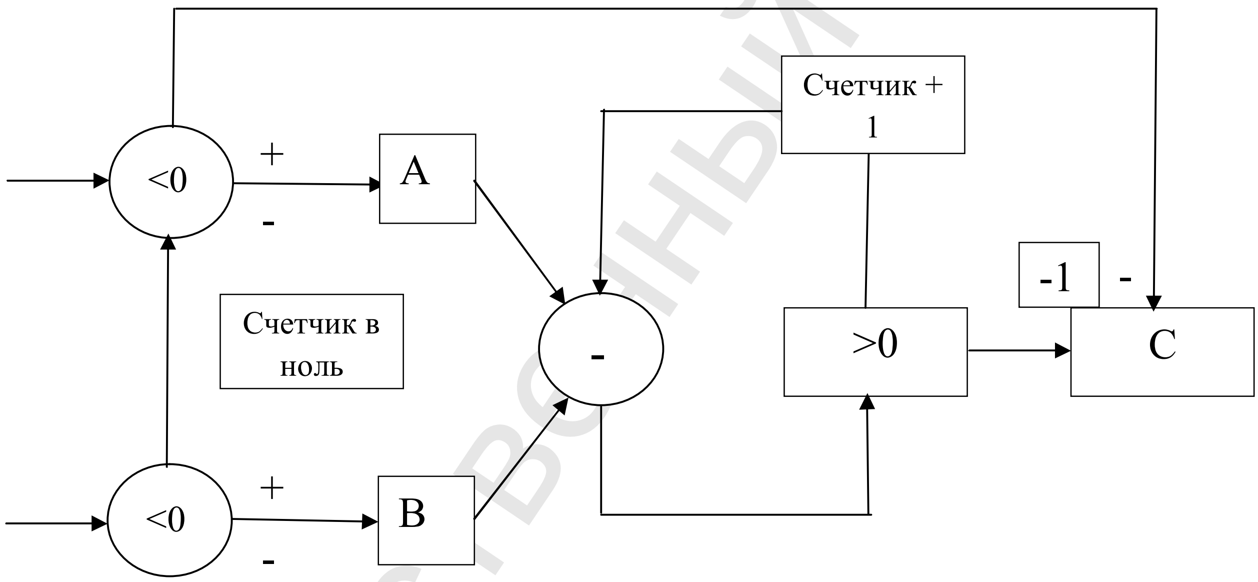


Рисунок II.3.10. Общая схема нейрона со счетчиком совпадений.

Алгоритм работы нейрона со счетчиком совпадений разбивается на три этапа:

На первом этапе векторы *X* и *A* подаются на входы. Выполняется операция вычитания вектора весовых коэффициентов (*A*) из входного вектора (*X*) с разбиением на *n* элементов, в результате которой получается вектор (*X-A*).

На втором этапе для расчета числа совпадений необходимо в векторе (*X-A*) преобразовать в 0 для совпадений и 1 для несовпадений. Для этого используется арифметическая функция активации с предварительным маскированием для избавления от отрицательных значений. В результате активации вектора (X-A) получается вектор M, содержащий единицы в позициях, где $x_i \neq a_i$ нули в позициях, где $x_i = a_i$.

На третьем этапе вектор M попадает на вход X и происходит суммирование всех его элементов. В результате получается количество несовпадений элементов векторов X и A, которое на последнем этапе вычитается из общего числа элементов, чтобы получить количество совпадений x_i с a_i.

Задания II.3.10.

1. Опишите работу нейрона со счетчиком совпадений при получении на вход n-мерного вектора X.
2. Укажите состав и нарисуйте общую схему нейрона со счетчиком совпадений.
3. Опишите действия на первом этапе алгоритма работы нейрона со счетчиком совпадений.
4. Опишите действия на втором этапе алгоритма работы нейрона со счетчиком совпадений.
5. Опишите действия на третьем этапе алгоритма работы нейрона со счетчиком совпадений.

Вопросы II.3.10.

1. К какой модели искусственного нейрона относятся нейроны со счетчиком совпадений?
2. В каких случаях нейрон со счетчиком совпадений на выходе выдает целое число?
3. Из скольких этапов состоит алгоритм работы нейрона со счетчиком совпадений?
4. Что поучается в нейроне со счетчиком совпадений в результате активации вектора (X-A)?
5. Что делается для получения количества совпадений x_i с a_i в нейроне со счетчиком совпадений?

НИИ искусственный интеллект

II.3.11. Кубические нейроны

В этом параграфе рассмотрены кубические нейроны, обсуждены возможности данных нейронов, предложены задания, сформулированы вопросы. При подготовке учебного материала данного параграфа были использованы источники [1-9].

Кубические нейроны относятся к нестандартной модели искусственного нейрона. В кубической модели нейронов вектор входных двоичных сигналов X рассматривается как адрес ячейки памяти, содержимое которой равно 0 или 1. Для размерности n вектора X существует 2^{n} возможных адресов.

Можно рассматривать ячейки памяти, как вершины $\boldsymbol{n}$ - мерного *гиперкуба*. Ячейки памяти получают значения независимо друг от друга. Полезно рассматривать ячейки памяти как содержащие поляризованные двоичные значения ± 1. Тогда работа кубического модуля описывается следующим образом.

Двоичный вход x используется как адрес памяти, поляризованная двоичная величина считывается и конвертируется в неполяризованную форму функцией h. Обозначим значения по адресу x через v_x, так что $y = h(v_x)$. Такие модули мы будем называть кубическими, чтобы подчеркнуть геометрическое представление множества адресов значений активации как множество вершин гиперкуба.

Рассмотрим двухвходовый кубический модуль. Существует 4 значения активации $\{v_{00}, v_{01}, v_{10}, v_{11}\}$. Выражение для активации будет иметь следующий вид:

$$s = v_{00}(1-x_1)(1-x_2) + v_{01}(1-x_1)x_2 + v_{10}x_1(1-x_2) + v_{11}(1+x_1)(1+x_2)$$

$x = (x_1, x_2)$ - входной вектор. Такая запись вызвана тем, что только одно из произведений в сумме должно быть ненулевым. Для поляризованных входов x_1 и x_2 активация будет

$$s = [v_{00}(1-x_1)(1-x_2) + v_{01}(1-x_1)x_2 + v_{10}x_1(1-x_2) + v_{11}(1+x_1)(1+x_2)]/4$$

В случае модуля с n входом получим

$$s=[\sum_{i=0}^{m} v_{ij}\prod_{j=1}^{n}(1-x_j)]/2^n$$

Кубические нейроны обучаются путем изменения содержимого ячейки их памяти. Обозначим через '+' операцию инкремента – установки содержимого ячейки в +1, а через '-' операцию декремента – установки в -1.

Пусть в начальном состоянии все ячейки кубического нейрона установлены в ноль. Обозначим ячейки, адресуемые обучающей выборкой, как центральные ячейки или центры. Ячейки, близкие к центрам в смысле расстояния Хемминга, будем настраивать на те же или близкие к ним значения, что и сами центры, т.е. должна происходить кластеризация значений ячейки вокруг центра. Это условие должно выполняться для сети из кубических нейронов.

Алгоритм обучения строит так называемое разбиение Вороного, при котором значение в ячейке определяется значением в ближайшем центре, а ячейки, равноудаленные от центров, остаются установленными в ноль. Кубические нейроны допускают большую функциональность, чем полулинейные, и поэтому, возможно, позволяют решать те же задачи при меньшем количестве модулей.

Задания II.3.11.

1. Напишите выражение для активизации двухвходового кубического модуля.
2. Напишите выражение активизации для поляризованных *n* входов.
3. Сравните кубические и полулинейные нейроны.

Вопросы II.3.11.

1. Как рассматривается вектор входных двоичных сигналов в кубической модели нейронов?
2. Что происходит в каждом цикле обучения?
3. Что строит алгоритм обучения?

НИИ искусств ... кт

II.3.12. Радиальные нейроны

В этом параграфе рассмотрены радиальные нейроны, обсуждены возможности данных нейронов, предложены задания и сформулированы вопросы. При подготовке учебного материала данного параграфа были использованы источники [1-9].

Радиальные нейроны относятся к нестандартной модели искусственного нейрона и представляет собой гиперсферу, которая осуществляет шаровое разделение пространства вокруг центральной точки. Именно с этой точки зрения он является естественным дополнением сигмоидального нейрона, поскольку в случае круговой симметрии данных позволяет заметно уменьшить количество нейронов, необходимых для разделения различных классов. На рисунке II.3.12.1 показаны способы разделения пространства данных сигмоидальным и радиальным нейронами.

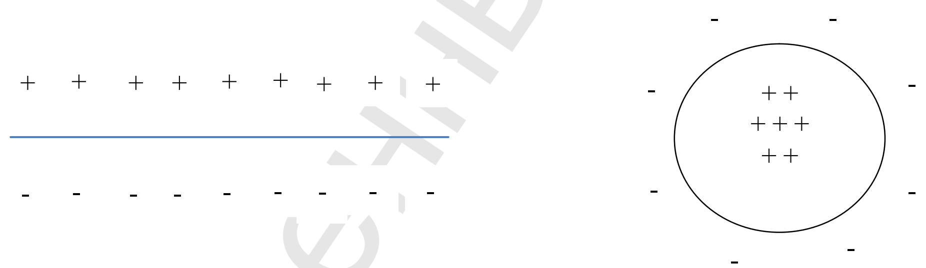


Рисунок II.3.12.1.Способы разделения пространства данных.
А) синмоидальным нейроном; В) радиальным нейроном.

Радиальные нейроны реализуют функцию, радиально изменяющуюся вокруг выбранного центра и принимающую ненулевые значения только в окрестности этого центра. Подобные функции, определяемые в виде $f(X,C)=\|X\text{-}C\|$, называются *радиальными базисными функциями*, где $X \in R^n$, $\boldsymbol{C}$ – центр.

Примерами радиальных базисных функций могут служить:

1. Мультиквадратичная функция: $f(X,C)=\left(\|X\text{-}C\|^2+a^2\right)^{\frac{1}{2}}$, где a>0.

2. Сплайн тонких пластин: $f(X,C)=\left(\|X\text{-}C\|^2 \ln\left(\|X\text{-}C\|\right)\right)^{\frac{1}{2}}$.

НИИ искусственный интеллект

3. Обратная мультиквадратичная функция: $f(X,C)=\dfrac{1}{\left(\|X-C\|^2+a^2\right)^{\frac{1}{2}}}, a>0$

4. Функция Гаусса: $f(X,C)=\exp\left(-\dfrac{\|X-C\|^2}{2\sigma^2}\right)$, где σ>0 – параметр, определяющий "ширину" функции.

Структура радиального нейрона показана на рисунке II.3.12.2.

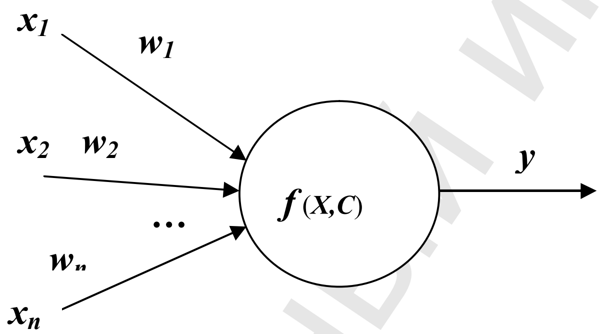


Рисунок II.3.12.2. Структура радиального нейрона.

На рисунке II.3.12.3 приведен график одномерной радиальной функции в скалярном варианте для различных значений s_i.

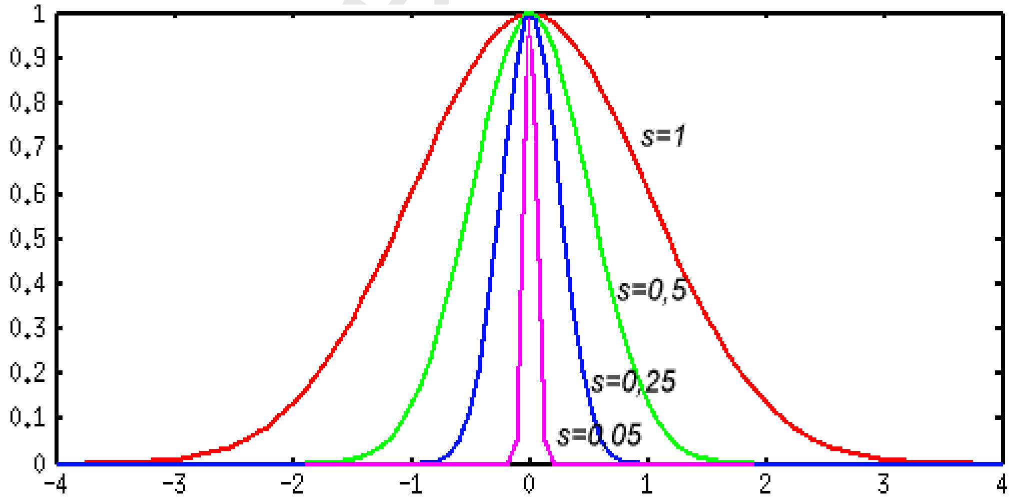


Рисунок II.3.12.3. График одномерной радиальной функции.

На рисунке II.3.12.4 приведен график двумерной радиальной функции в скалярном варианте для различных значений s_i.

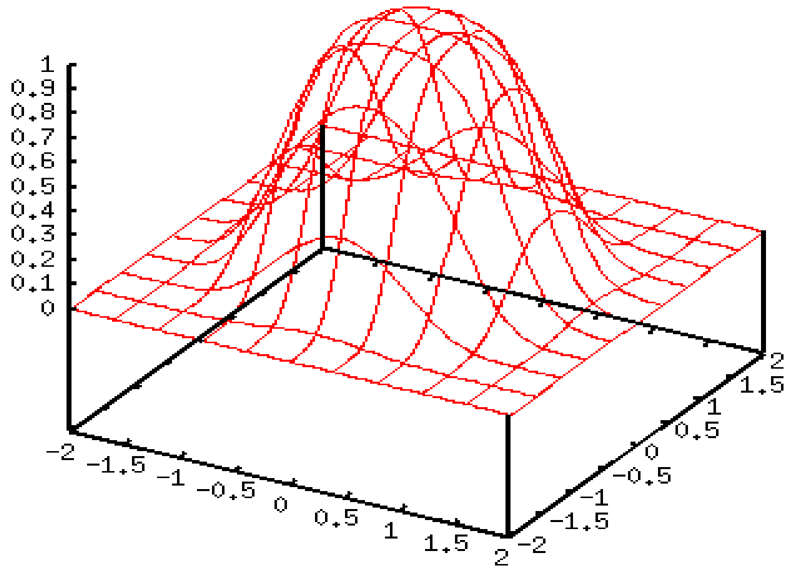


Рисунок II.3.12.4. График двумерной радиальной функции.

Принципиальное отличие радиального нейрона от сигмоидального нейрона и персептрона в том, что сигмоидальный нейрон разбивает многомерное пространство входных сигналов гиперплоскостью, а радиальный – гиперсферой.

Задания II.3.12.

1. Нарисуйте способы разделения пространства данных сигмоидальным и радиальным нейронами.
2. Напишите выражение для радиальной базисной функции.
3. Напишите выражение для мультиквадратичной функции.
4. Напишите выражение для функции сплайн тонких пластин.
5. Напишите выражение для обратной мультиквадратичной функции.

Вопросы II.3.12.

1. Какую функцию реализуют радиальные нейроны?
2. Как изменяются радиальные функции?
3. Какие значения принимают радиальные функции в окрестности центра?
4. На какую фигуру похож график двумерной радиальной функции?
5. В чем принципиальное отличие радиального нейрона от сигмоидального нейрона и персептрона?

II.3.13. Стохастические нейроны

В этом параграфе рассмотрены стохастические нейроны, обсуждены возможности данных нейроов, предложены задания и сформулированы вопросы. При подготовке учебного материала данного параграфа были использованы источники [1-9].

Выше были описаны модели детерминированных искусственных нейронов, в которых состояния на выходе однозначно определяются результатами работ сумматора входных сигналов. Имеются также стохастические нейроны, относящийся к нестандартной модели искусственного нейрона, где переключение нейрона происходит с вероятностью, зависящей от индуцированного локального поля, т.е. функция активации определена как

$$f(s) = \begin{cases} 1, & \text{если} \quad P(s) \\ 0, & \text{если} \quad 1 - P(s) \end{cases},$$

где распределение вероятности *P* обычно имеет вид сигмоида:

$$\sigma(s) = \frac{A(T)}{1 + e^{(-s/T)}} ,$$

а нормировочная константа *A* вводится для условия нормализации распределения вероятности:

$$\int_0^1 \sigma(s) ds = 1 \cdot$$

Таким образом, нейрон активируется с вероятностью *P*. Параметр *T* - аналог температуры и определяет беспорядок в нейронной сети. Если *T* устремить к 0, стохастический нейрон перейдет в обычный нейрон с передаточной функцией Хэвисайда.

В стохастической модели, в отличие от детерминированных моделей, выходное состояние нейрона зависит не только от взвешенной суммы входных сигналов, но и от некоторой случайной переменной величины, значения которой выбираются при каждой реализации из интервала (0,1).

НИИ ...КТ

В стохастической модели нейрона выходной сигнал y принимает значения ± 1 с вероятностью

$$P(y_i=\pm 1)=1/(1+e^{\pm 2\beta s_i}),$$

где s_i – взвешенная сумма входных сигналов i-го нейрона, β – положительная константа, которая чаще всего равна 1.

Процесс обучения нейрона в стохастической модели состоит из следующих этапов:

1)для каждого нейрона сети вычисляется взвешенная сумма:

$$s_j = \sum_{i=0}^{n} w_{ij} x_i ;$$

2)вычисление вероятности P того, что выходная переменная y_i принимает значение ± 1:

$$P(y_i=\pm 1)=\frac{1}{1+e^{\pm 2\beta s_i}} ;$$

3)генерация значениий случайной переменной $R\in(0,1)$;

4)формирование выходного сигнала y, если $R<P(y)$, или $-y$ в противном случае;

5)адаптация весов w_{ij} (при фиксированных y_i) по используемым правилам.

При обучении с учителем по правилу Уидроу-Хоффа адаптация весов проводится по формуле:

$$\Delta w_i = \eta\, x_i (d_i - y_i)$$

где η – коэффициент обучения нейрона, значение которого лежит в интервале (0, 1).

Совокупность нейронов, определенным образом соединенных друг с другом и внешней средой, образует нейронную сеть. Для того, чтобы узнать какие существуют нейронные сети, правильно оценить их возможности, узнать как улучшить существующие модели нейронов и нейронных сетей, предположить новые разработки в этой области, необходимо произвести их классификацию.

НИИ ИОСУ-Пэкт

Существует множество способов классификации нейронов и нейронных сетей. В следующем разделе будут рассмотрены некоторые из возможных классификаций нейронных сетей.

Задания II.3.13.

1. Напишите выражение функции активации для стохастической модели нейрона.

2. Напишите выражение распределения вероятности для стохастической модели нейрона.

3. Напишите выражение активизации для поляризованных n входов.

4. Напишите выражение вычисления взвешенной суммы для каждого нейрона.

5. Напишите выражение вероятности получения выходного сигнала стохастической модели нейрона.

6. Напишите выражение для адаптации весов при обучении с учителем по правилу Уидроу-Хоффа стохастической модели нейрона.

7. Перечислите требования для классификации нейронных сетей.

Вопросы II.3.13.

1. Чем отличаются стохастические искусственные нейроны от детерминированных нейронов?

2. Для чего вводится нормировочная константа A в выражении распределения вероятности для стохастической модели нейрона?

3. Что означает параметр T в выражении распределения вероятности для стохастической модели нейрона?

4. При каждой реализации из какого интервала выбираются значения случайной переменной величины , от которой зависит выходное состояние стохастической модели нейрона?

5. Из каких этапов состоит процесс обучения нейрона в стохастической модели?

6. Что образует совокупность нейронов, определенным образом соединенных друг с другом и внешней средой?

НИИ искусственный интеллект

7. Для чего нужна классификация нейронных сетей?

НИИ Искусственный интеллект

II.4. Модели нейронных сетей

II.4.1. Однослойные нейронные сети

В этом параграфе рассмотрены однослойные нейронные сети, предложены задания и сформулированы вопросы. При подготовке учебного материала данного параграфа были использованы источники [1-9].

Несмотря на то, что один нейрон способен выполнять простейшие процедуры распознавания, для серьезных нейронных вычислений необходимо соединять нейроны в сети. Структура простейшей нейронной сети, которая состоит из группы нейронов, образующих один слой, представлена на рисунке II.4.1.1.

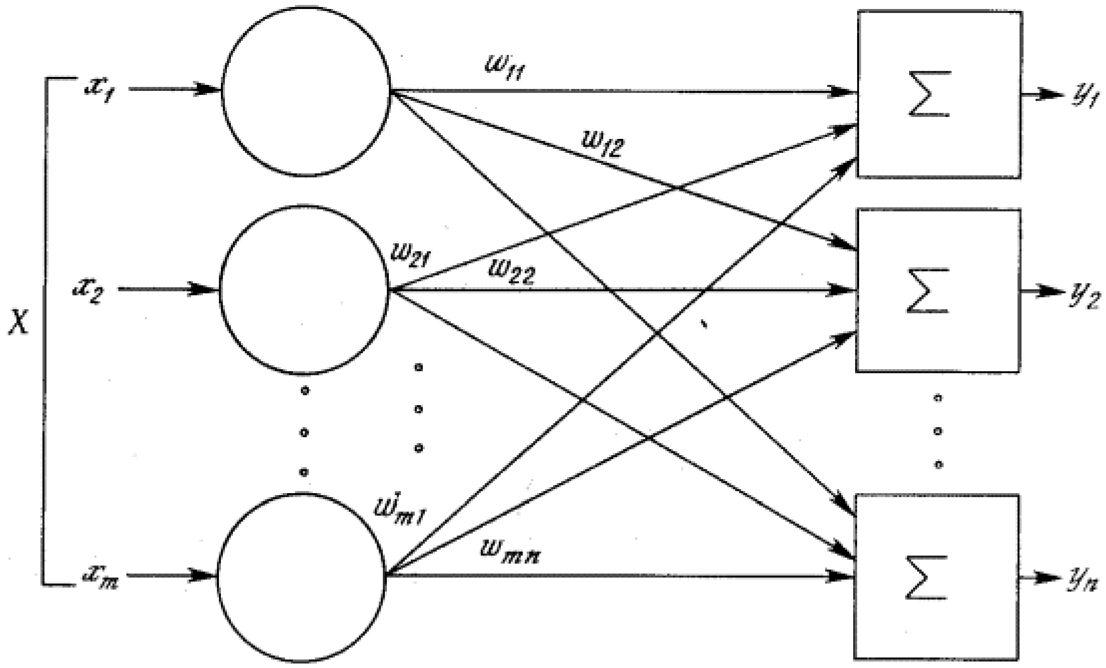


Рисунок II.4.1.1. Структура однослойной нейронной сети.

Здесь вершины - круги слева - служат лишь для распределения входных сигналов. Они не выполняют каких-либо вычислений и поэтому не будут считаться слоем. Каждый элемент x_i (i=1,2,…, n) из вектора входов X отдельным весом соединен с каждым j-тым нейроном (j=1,2,…, m). Каждый нейрон выдает взвешенную сумму входов в сеть. Веса являются элементами матрицы W, которая имеет n строк и m столбцов, где n – число входов, а m – число нейронов. Например, w_{21} – это вес, связывающий второй вход с первым нейроном.

НИИ искусственного интеллекта

Вычисление выходного вектора Y, компонентами которого являются выходы y_1, y_2, …, y_m нейронов, сводится к применению функции активации f к матричному умножению $S=XW$, где S и X – векторы-строки.

Однослойные нейронные сети по типу связи подразделяются на *полносвязные* и *слабосвязные* (*регулярные*).

В полносвязных нейронных сетях все входные сигналы подаются всем нейронам и каждый нейрон передает свой выходной сигнал остальным нейронам, в том числе и самому себе. Выходными сигналами сети могут быть все или некоторые выходные сигналы нейронов после нескольких тактов функционирования сети.

На рисунке II.4.1.2. представлен пример структуры полносвязной нейронной сети.

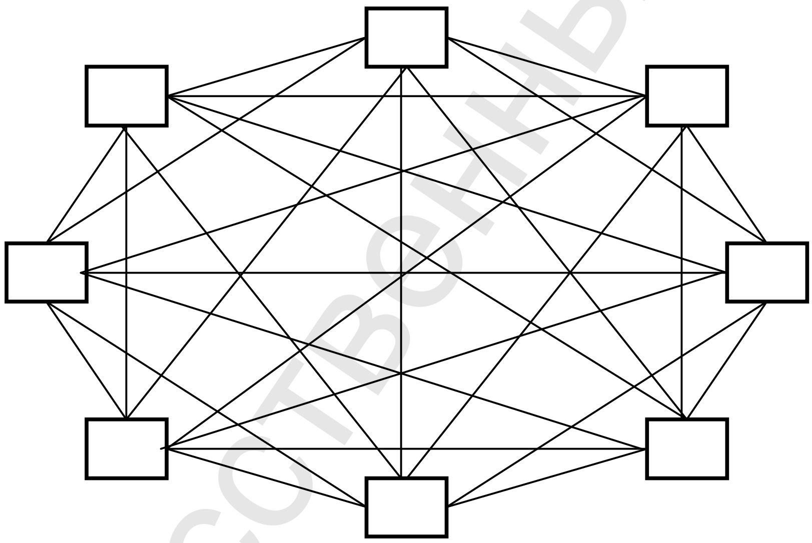


Рисунок II.4.1.2. Структура полносвязной нейронной сети.

В регулярных (слабосвязанных) нейронных сетях любой из нейронов может быть входным и выходным, и каждый нейрон имеет только определенное количество связей. Причем число связей нейрона зависит от места его расположения в сети. Если каждый нейрон связан с 4, 6 и 8 соседями, то это будет окрестностью *фон Неймана*, *Голея* и *Мура* соответственно.

На рисунке II.4.1.3 представлен пример регулярной нейронной сети, в которой угловые нейроны (расположенные в углах сети)

НИИ искусственный интеллект

связаны лишь с двумя ближайшими нейронами, расположенными на горизонтальной и вертикальной границе, а пограничные нейроны (крайние слева и справа) имеют связи с тремя ближайшими нейронами: двумя нейронами, расположенными вдоль границы, и одним внутренним нейроном.

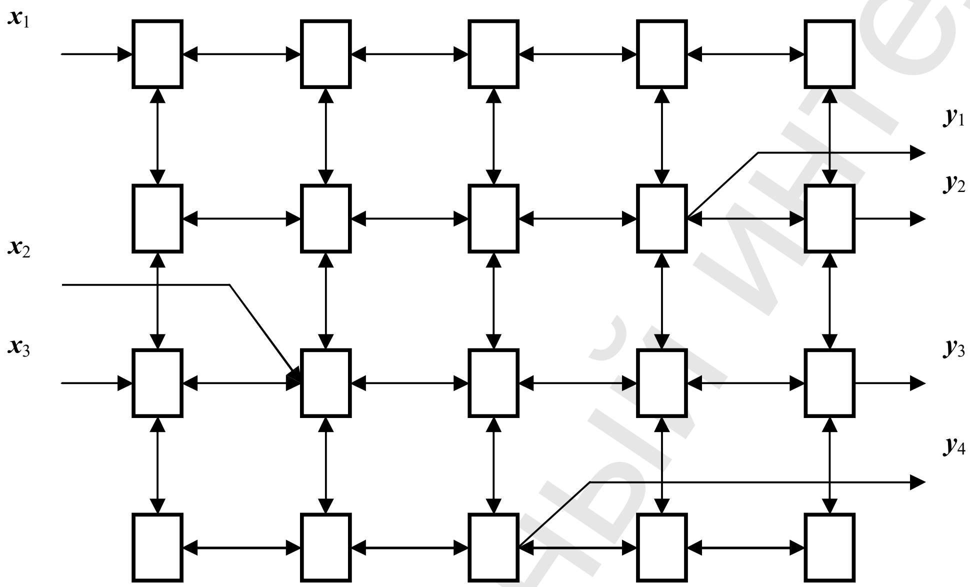


Рисунок II.4.1.3. Регулярная нейронная сеть.

На рисунке II.4.1.4 представлена пример регулярной гексагональной нейронной сети, в которой внутренние нейроны имеют по шесть связей.

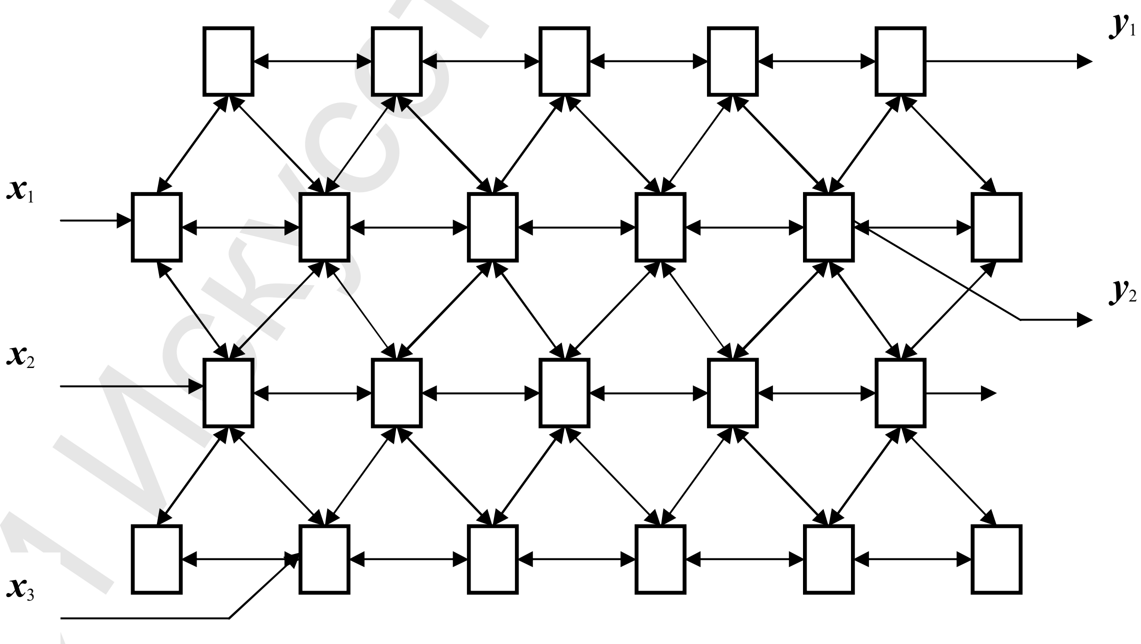


Рисунок II.4.1.4. Регулярная гексагональная нейронная сеть

НИЛ Искусственный интеллект

II.4.1. Задания:

1. Дайте определение однослойной нейронной сети.
2. Нарисуйте структуру однослойной нейронной сети.
3. Напишите формулу однослойной нейронной сети.
4. Определите вид нейронной сети, если все входные сигналы подаются всем нейронам и каждый нейрон передает свой выходной сигнал остальным нейронам, в том числе и самому себе.
5. Опишите структуру полносвязной нейронной сети.
6. Определите вид нейронной сети, если любой из нейронов может быть входным и выходным, и каждый нейрон имеет только определенное количество связей.
7. Назовите окрестность, если каждый нейрон связан с 4 соседями.
8. Назовите окрестность, если каждый нейрон связан с 6 соседями.
9. Назовите окрестность, если каждый нейрон связан с 8 соседями.

II.4.1. Вопросы:

1. Почему нужно соединять отдельные искусственные нейроны в определенную сеть?
2. Из каких элементов состоит однослойная нейронная сеть?
3. С чем соединен каждый элемент x_i из вектора входов X?
4. Какую функцию выполняет распределительный слой персептрона?
5. На какие виды подразделяются однослойные нейронные сети?
6. Какие элементы могут быть выходными сигналами полносвязной нейронной сети после нескольких тактов функционирования сети?
7. В какой нейронной сети число связей нейрона зависит от места его расположения в сети?
8. Что такое регулярная искусственная нейронная сеть?

9. Что такое регулярная гексагональная искусственная нейронная сеть?

НИИ Искусственный интеллект

II.4.2. Однослойные персептроны

В этом параграфе рассмотрены однослойные персептроны, предложены задания и сформулированы вопросы. При подготовке учебного материала данного параграфа были использованы источники [1-9].

Хотя однонейронный персептрон способен выполнять простейшие вычисления, у персептронов, полученных от соединений нескольких нейронов, вычислительная мощность намного сильнее. В этих персептронах нейроны могут образовывать один или несколько слоев, в которых каждый слой состоит из множества весов синаптических связей со следующими за ними нейронами, суммирующими взвешенные сигналы. Среди таких персептронов самыми простыми являются однослойные персептроны.

Обычно однослойный персептрон изображается двумя слоями: первый слой нейронов является *распределительным*, а второй слой нейронов – *обрабатывающим.*

Структура однослойного персептрона, имеющего n входных сигналов и состоящего из m нейронов показана на рисунке II.4.2.

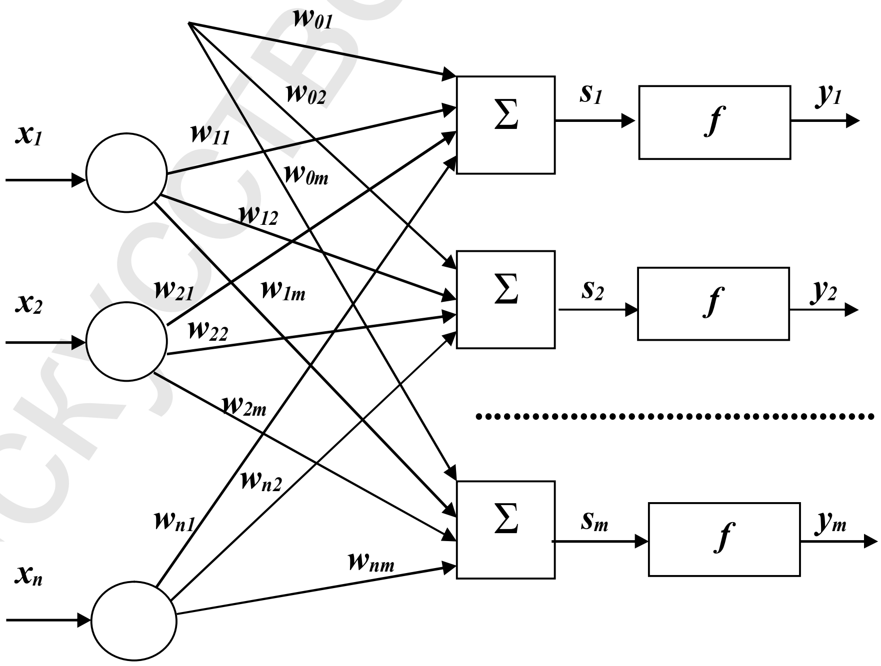


Рисунок II.4.2. Структура однослойного персептрона.

НИИ Искусственный интеллект

Здесь вершины-круги слева образуют *распределительный слой*. Они не выполняют каких-либо вычислений и отличаются от вершин-квадратов, которые выполняют определенные вычисления.

Распределительный слой передает входные сигналы на обрабатывающий слой, который преобразует входную информацию в соответствии с синаптическими связями и функцией активации. При этом каждый элемент из входного вектора X отдельным весом в распределительном слое соединен со всеми нейронами обрабатывающего слоя, и это соединение задается матрицей весов W. Каждый нейрон выдает взвешенную сумму входов s_j, представленную с помощью скалярной функции (скалярного произведения). Совокупность скалярных функций s_j объединяется в m-элементный вектор входа m функции активации слоя. Выходы слоя нейронов формируют вектор-столбец y_j.

Однослойный персептрон вычисляет взвешенную сумму элементов входного сигнала, отнимает значение порога (сдвига) и пропускает результат через жесткую пороговую функцию, выход которой равняется +1 или -1:

$$y_j = f(s_j) = f\left(\sum_{i=1}^{n} w_{ij} x_i - w_{0j}\right),$$

где x_i – i-тые входные сигналы с типом значений; бинарным (цифровые) или действительным (аналоговые) j-того нейрона; w_{ij}– вес синаптической связи между i-тым нейроном распределительного слоя и j-тым нейроном обрабатывающего слоя; w_{0j}– порог (сдвиг) j-того нейрона выходного слоя; s_j – взвешенная сумма j-того нейрона; y_j – выходной сигнал j-того нейрона, f – нелинейная функция активации; n – число входов (синаптических весов); i=1, 2, …, n; m – число нейронов, j=1, 2,…, m.

Если значение веса w_{ij} положительное (отрицательное), то i-тая синаптическая связь является возбуждающей (тормозящей).

НИИ Искусственный интеллект

Здесь нейрон, имеющий *n* двоичных входов, может принимать 2^n различных значений входа (образов), состоящих из 0 и 1. Так как каждый образ может соответствовать двум различным бинарным выходам (0 и 1), то всего имеются 2^{2^n} функций от *n* переменных.

Число линейно-разделимых функций от *n* переменных представлено в таблице II.4.2.

Таблица II.4.2. Число линейно-разделимых функций.

№	2^{2^n}	***Число линейно разделимых функций***
1	4	4
2	16	14
3	256	104
4	65536	1 882
5	$4{,}3\times10^{9}$	94 572
6	$1{,}8\times10^{19}$	15 028 134

Как видно из таблицы II.4.2, вероятность того, что случайно выбранная функция окажется линейно-разделимой, весьма мала даже для умеренного числа независимых переменных. По этой причине однослойные персептроны на практике ограничены простыми задачами.

Линейное разделение классов состоит в построении решающего правила:

$$f(\mathrm{W}\cdot X) > 0, \Rightarrow X \in \text{первому классу.}$$

$$f(\mathrm{W}\cdot X) \leq 0, \Rightarrow X \in \text{второму классу;}$$

т.е. нахождение такого вектора $W = (w_1, w_2, \ldots, w_n)$,

где w_0 – порог при котором возможно точное выполнение неравенства для всех возможных подаваемых сигналов.

Разделение центров масс – простейший способ построения решающего правила. Суть этого способа заключается в вычислении вектора весов персептрона по следующей формуле

НИИ Искусственный интеллект

$$W=\left(\sum_{i=1}^{n} X_1^i - \sum_{j=1}^{m} X_2^j\right)/(n+m),$$

где X_1^i – вектора из множества первого класса, а X_2^j – вектора из множества второго класса.

Линейные решающие правила, построенные на основании разделения центров масс, могут ошибаться на примерах из обучающей выборки. Однако, этот метод полезен как средство определения начального значения (инициализации) вектора весов для алгоритма обучения персептрона.

Однослойный персептрон позволяет решать задачи классификации на большее число классов.

II.4.2. Задания:

1. Дайте определение однослойного персептрона.
2. Нарисуйте структуру однослойного персептрона.
3. Напишите формулу однослойного персептрона.
4. Напишите формулу решающего правила для линейного разделения классов.
5. Напишите формулу вычисления вектора весов однослойного персептрона.
6. Напишите формулу функции активации персептрона.
7. Постройте график функции активации персептрона в виде сигмоиды.

II.4.2. Вопросы:

1. Сколько различных образов будет иметь нейрон с *n* двоичными входами?
2. Что такой однослойный персептрон?
3. Какую функцию выполняет распределительный слой персептрона?
4. Как вычисляется число линейно-разделимых функций?
5. Какое правило будет использоваться для обучения персептрона?

НИИ искусственный интеллект

II.4.3. Многослойные нейронные сети

В этом параграфе рассмотрены многослойные нейронные сети, предложены задания и сформулированы вопросы. При подготовке учебного материала данного параграфа были использованы источники [1-9].

В многослойных нейронных сетях нейроны объединяются в слои. Слой содержит совокупность нейронов с единым вектором входных сигналов. Число нейронов в слое может быть любым и не зависит от количества нейронов в других слоях. В общем случае сеть состоит из $q>1$ слоев, пронумерованных слева направо. Внешние входные сигналы подаются на входы нейронов входного слоя, а выходами сети являются выходные сигналы последнего слоя. Кроме входного и выходного слоев в многослойной нейронной сети есть один или несколько скрытых слоев. Связи от выходов нейронов некоторого слоя q к входам нейронов следующего слоя $(q+1)$ называются последовательными. Внутри одного слоя используется одна и та же функция активации. *Входной слой* нумеруется нулем и принимает вектор входных сигналов сети $x_1, x_2, \dots, x_n$, *выходной слой* нумеруется q и вырабатывает вектор выходных сигналов сети $y_1, y_2, \dots, y_m$, а *скрытые слои* нумеруются k $(k=1,2,\dots,q-1)$. Функции активации нейронов одного слоя одинаковые.

Многослойная нейронная сеть в зависимости от значения связи может быть монотонной. В *монотонной нейронной сети* каждый слой кроме выходного разбит на два блока: *возбуждающий* и *тормозящий*. Связи между блоками тоже разделяются на возбуждающие и тормозящие. Если от нейронов блока *A* к нейронам блока *B* ведут только возбуждающие связи, то это означает, что любой выходной сигнал блока *B* является монотонной неубывающей функцией любого выходного сигнала блока *A*. Если же эти связи только тормозящие, то любой выходной сигнал блока *B* является невозрастающей функцией любого выходного сигнала блока *A*. Для нейронов монотонных сетей необходима монотонная

НИИ искусственного интеллекта

зависимость выходного сигнала нейрона от параметров входных сигналов.

В общем случае в многослойных нейронных сетях произвольной структуры возможны различные конфигурации связей. По типам связи многослойные нейронные сети подразделяются на: *нейронные сети с латеральными связями*, *нейронные сети с прямыми связями*, *нейронные сети с перекрестными связями*, *нейронные сети с обратными связями.*

Многослойные нейронные сети с латеральными (боковыми) связями в скрытых слоях позволяет контрастировать изображение на случайном фоне и повышает эффективность распознавания графических образов. Латеральные связи могут быть возбуждающими и тормозящими. На рисунке II.4.3.1 показан один слой с латеральными связями в многослойной нейронной сети.

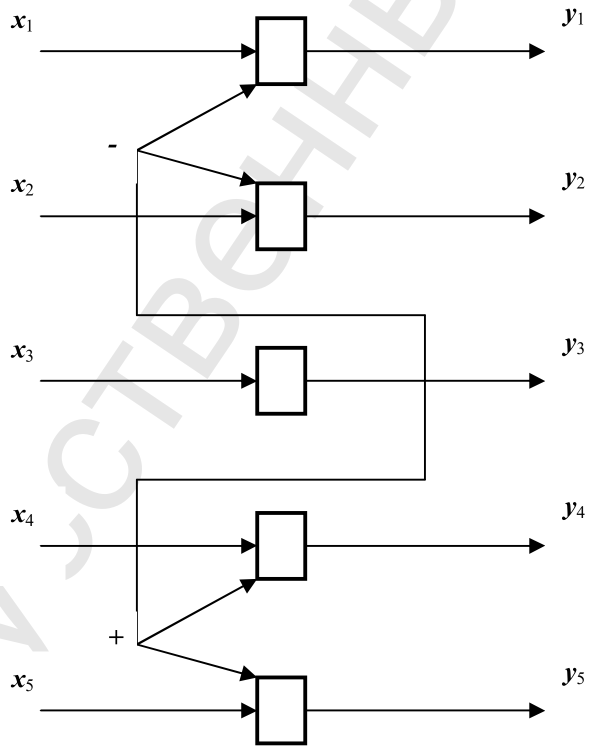


Рисунок II.4.3.1. Слой нейронной сети с латеральными связями.

Здесь, подавая сигналы через отрицательные коэффициенты синаптической связи с центрального нейрона на два верхних, добиваются снижения активности этих нейронов. Подача сигналов через положительные коэффициенты синаптической связи с

НИИ искусственный интеллект

центрального нейрона на два нижних увеличивает их активность. Поэтому центральный нейрон и два нижних нейрона будут значительно активнее двух верхних, что эквивалентно повышению контрастности входного сигнала, например, изображения.

В *многослойной сети с прямыми связями* (прямого распространения сигналов) нейроны каждого слоя не связаны между собой, и все связи направлены строго от входных нейронов к выходным, т.е. их графы не имеют петель и циклов. Входной сигнал, подаваемый на сеть, поступает на нейроны входного слоя, проходит по очереди через все слои и снимается с выходов нейронов выходного слоя в тот же момент, когда на вход подаются входные сигналы. По мере распространения сигнала по сети он претерпевает ряд преобразований, которые зависят от его начального значения, от преобразующих функций и величин весов связей. Схема *k*-слойной нейронной сети с прямыми связями изображена на рисунке II.4.3.2.

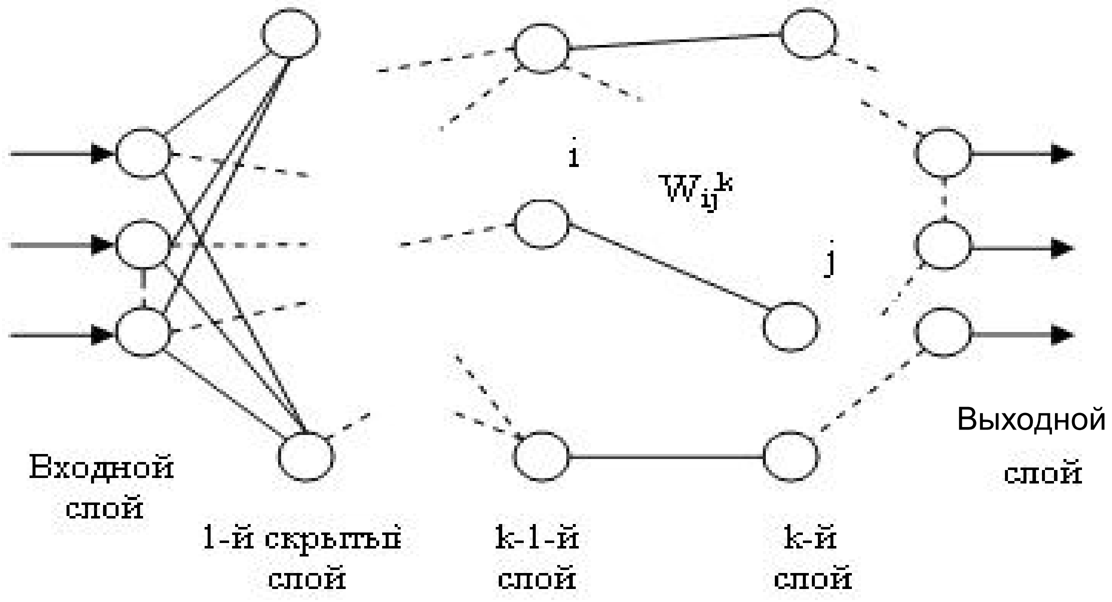


Рисунок II.4.3.2. Схема *k*-слойной нейронной сети с прямыми связями.

В нейронных сетях с прямыми связями каждый нейрон выполняет взвешенное суммирование элементов выходных сигналов и над полученным результатом затем выполняется нелинейное преобразование с помощью функции активации, значение которой считается выходом нейронной сети.

Среди многослойных нейронных сетей с прямыми связями различают полносвязные, в которых выход каждого нейрона *i*-го слоя связан с входом каждого нейрона (*i+1*)-го слоя. Такие нейронные сети с прямой связью являются нейронными сетями с временной задержкой.

Нейронные сети с перекрестными связями содержат нейроны, начиная со второго уровня, имеющие настраиваемые синаптические связи с выходов нейрона предыдущих слоев, так и непосредственно с внешнего входа сети. На рисунке II.4.3.3. показан пример двухслойной нейронной сети с перекрестными связями.

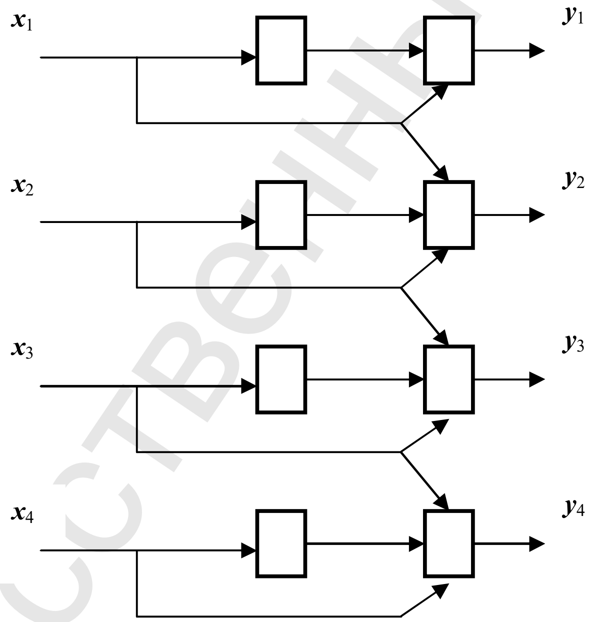


Рисунок II.4.3.3. Двухслойная нейронная сеть с перекрестными связями.

Нейронные сети с обратными связами организованы так, что каждый нейрон получает входную информацию из внешней среды и от других нейронов, возможно, и от себя самого (обратная связь). Такой обратной связью можно охватить, как отдельный слой, так и несколько слоев и даже всю сеть, т.е. граф нейронной сети с обратными связами может иметь циклы или петли.

НИИ искусственный интеллект

Нейронные сети с обратными связами состоят из динамических нейронов, чье поведение описывается дифференциальными уравнениями первого порядка или рекуррентными разностными отношениями. Поэтому их иногда называют динамическими или рекуррентными нейронными сетями. На рисунке II.4.3.4. представлен четырехслойная рекуррентная нейронная сеть.

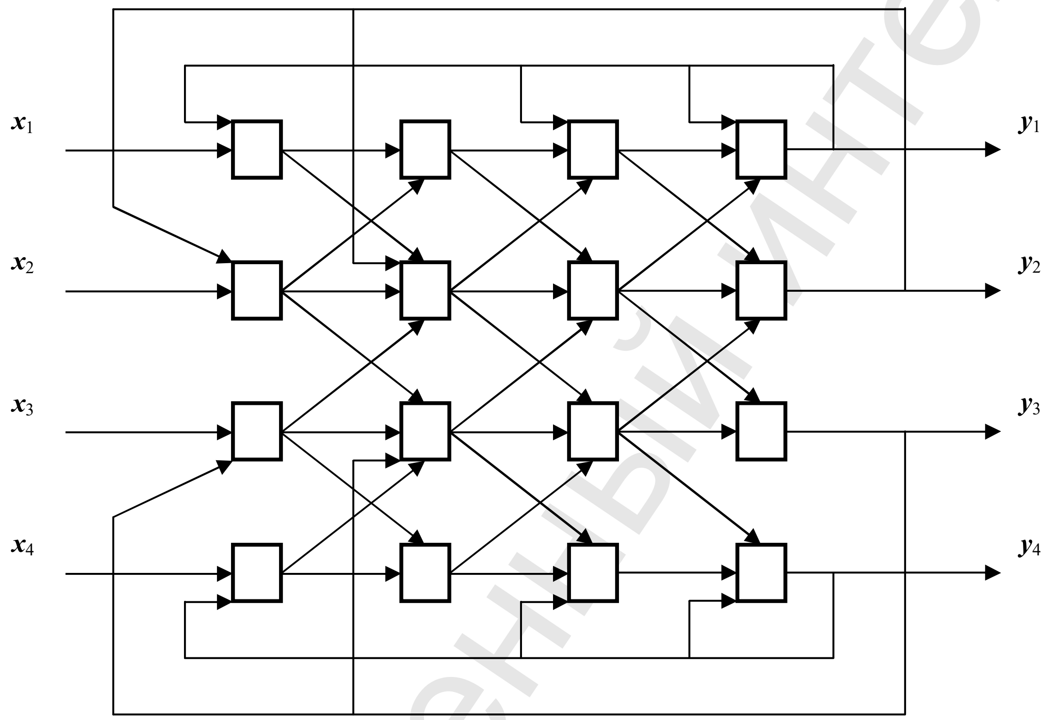


Рисунок II.4.3.4. Четырехслойная рекуррентная нейронная сеть.

Среди нейронных сетей с обратными связями часто используются частично-рекуррентные нейронные сети Элмана и Жордана. На рисунке II.4.3.5. представлены частично-рекуррентные нейронные сети Элмана.

НИИ Искусственный интеллект

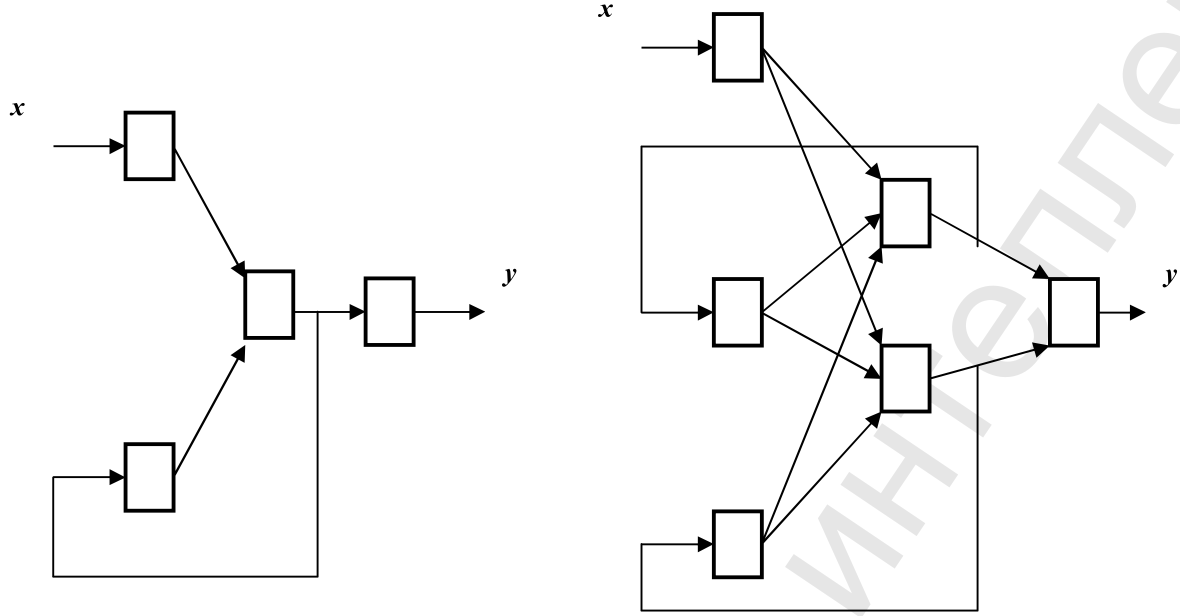


Рисунок II.4.3.5. Частично-рекуррентные нейронные сети Элмана

На рисунке II.4.3.6. представлены частично-рекуррентные нейронные сети Журдена

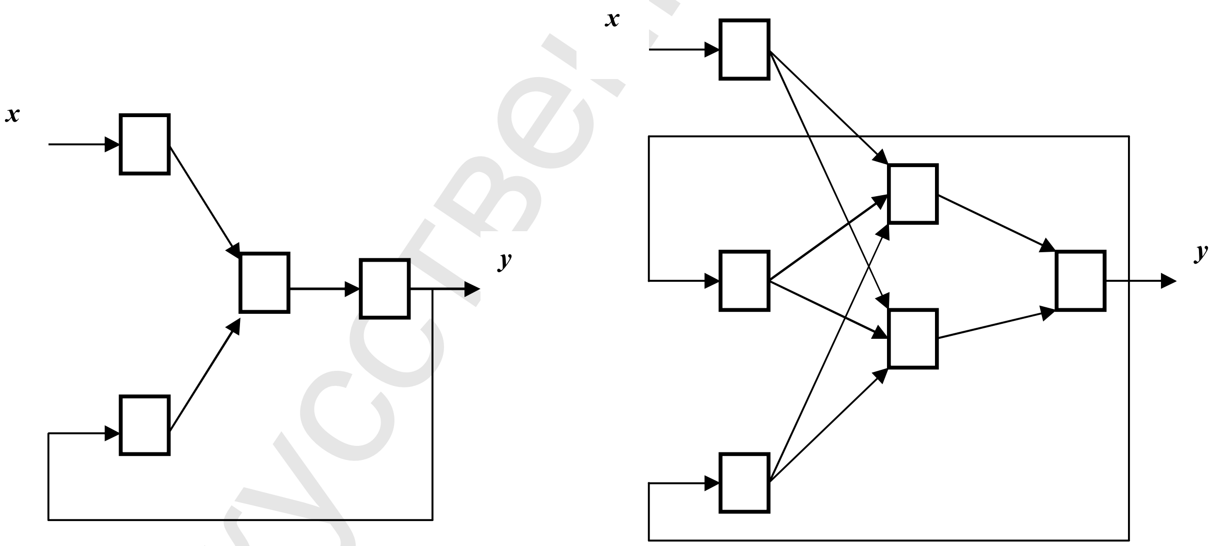


Рисунок II.4.3.6. Частично-рекуррентные нейронные сети Жордана

Частным случаем рекуррентных сетей является двунаправленные сети. В таких сетях между слоями существуют связи как в направлении от входного слоя к выходному, так и в обратном. Нейронные сети с обратными связями подразделяются на *слоисто-циклические*, *слоисто-полносвязные*, *полносвязно-*

НИИ искусственный интеллект

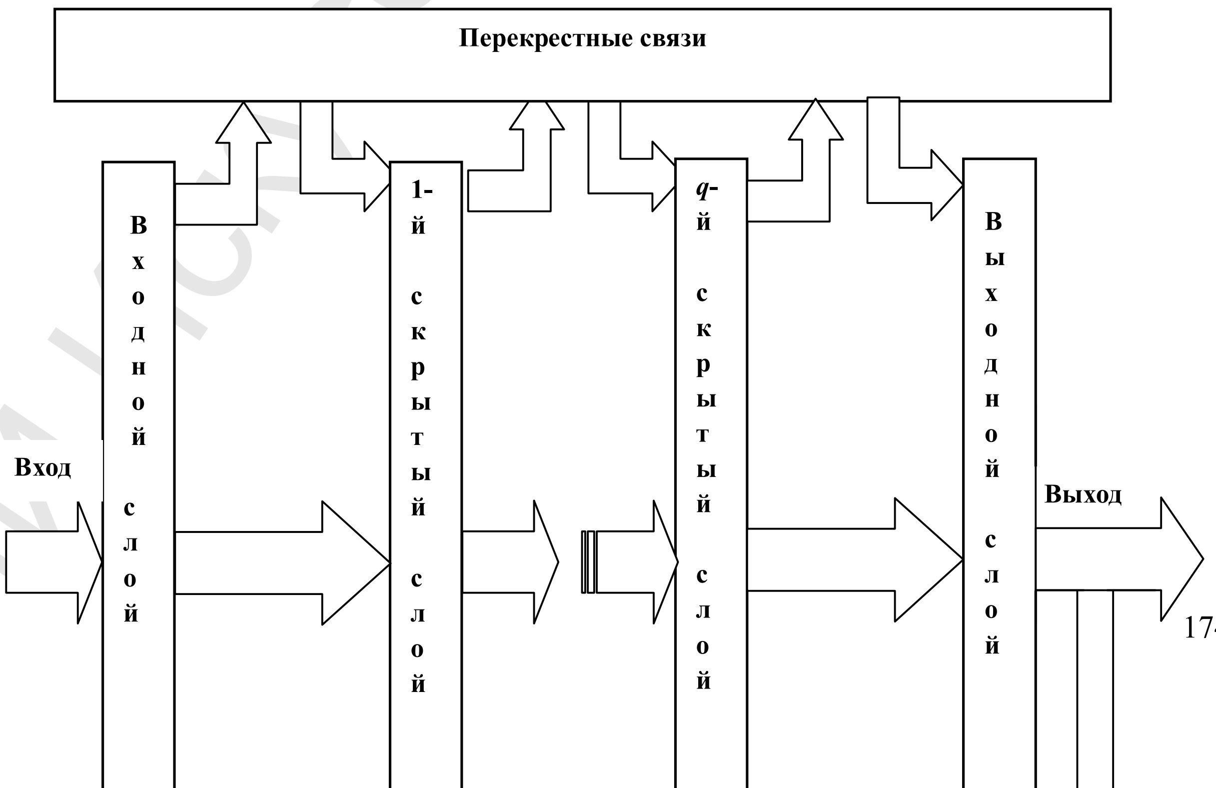
слоистые. *Слоисто-циклические* отличаются тем, что слои замкнуты в кольцо. Последний слой передает свои выходные сигналы первому. Все слои равноправны и могут как получать входные сигналы, так и выдавать выходные сигналы.

Слоисто-полносвязные сети состоят из слоев, каждый из которых представляет собой полносвязную сеть, а сигналы передаются как от слоя к слою, так и внутри слоя; в каждом слое цикл работы распадается на три части: прием сигналов с предыдущего слоя, обмен сигналами внутри слоя, выработка выходного сигнала и передача к последующему слою.

Полносвязно-слоистые сети по своей структуре аналогичны слоисто-полносвязным, но функционируют по-другому. В данных сетях не разделяются фазы обмена внутри слоя и передачи следующему, на каждом такте нейроны всех слоев принимают сигналы от нейронов как своего слоя, так и последующих.

Возможности сети возрастает с увеличением числа нейронов в сети, плотности связей между ними и числа слоев. Введение нескольких типов синапсов способствует усилению мощности нейронной сети. Включение обратных связей наряду с увеличением возможностей сети поднимает вопрос о динамической устойчивости нейронной сети.

Обобщенная структура многослойной нейронной сети, без учета связей внутри слоев, приведена на рисунке II.4.3.7.



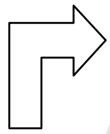


Рисунок II.4.3.7. Обобщенная структура многослойной нейронной сети

В действительности с помощью одного, достаточно большого скрытого слоя, возможно представить любую непрерывную функцию от входных данных с произвольной точностью. При наличии двух слоев можно представить даже функции с разрывами. Многослойная нейронная сеть может моделировать функцию практически любой степени сложности, причем число слоев и число элементов в каждом слое определяют сложность функции.

Теоретическое обоснование нейросетевого моделирования базируется на теореме А.Н.Колмогорова, доказавшего, что любую непрерывную многомерную функцию на единичном отрезке [0;1] можно представить в виде конечного числа одномерных:

$$f(x_1, x_2, \ldots, x_n) = \sum_{j=1}^{2n+1} g\left(\sum_{i=1}^{n} \lambda_i \varphi_j(x_i)\right),$$

где g и φ_j непрерывные и одномерные функции, $\lambda_i = \mathrm{const}$. Отсюда следует, что с помощью любой многослойной нейронной сети всего с двумя перерабатывающими слоями можно с любой точностью аппроксимировать любую многомерную функцию на единичном отрезке.

Задания II.4.3.

1. Опишите способ образования многослойных нейронных сетей.

2. Опишите способ нумерования слоев в многослойных нейронных сетях.

3. Приведите способ нумерации входного слоя, скрытых слоев и выходного слоя в многослойной нейронной сети.

4. Назовите критерии разделения на виды многослойных нейронных сетей.

5. Приведите обоснование нейросетевого моделирования.

Вопросы II.4.3.

1. Сколько нейронов могут быть в каждом слое многослойной нейронной сети?

2. Зависит ли число нейронов в конкретном слое от количества нейронов в других слоях?

3. Сколько функций активации используются внутри одного слоя в многослойной нейронной сети?

4. Какие функции можно представить с помощью многослойной нейронной сети?

НИИ Искусственный интеллект

II.4.4. Многослойные персептроны

В этом параграфе будут рассмотрены многослойные персептроны, предложены примеры, даны задания, сформулированы вопросы. При подготовке учебного материала данного параграфа были использованы источники [1-9].

Многослойные персептроны являются многослойными нейронными сетями с прямыми связями (с прямым распространением сигнала) и состоят из нескольких слоев нейронов: *входного слоя, нескольких скрытых слоев и выходного слоя*.

На входной слой поступает вектор $X=(x_1,x_2,...,x_n)$, который не выполняет никаких вычислений, а просто распределяет вектор X на следующий слой и служит приемником информации. Каждый скрытый элемент можно рассматривать как персептрон, который представляет мягкую пороговую функцию в пространстве входов. Каждый *S*-элемент связан с совокупностью ассоциативных элементов (*A*-элементов) первого промежуточного слоя, а *A*-элементы последнего слоя соединены с реагирующими элементами (*R*-элементами).

Структура многослойного персептрона с прямыми связями представлена на рисунке II.4.4.

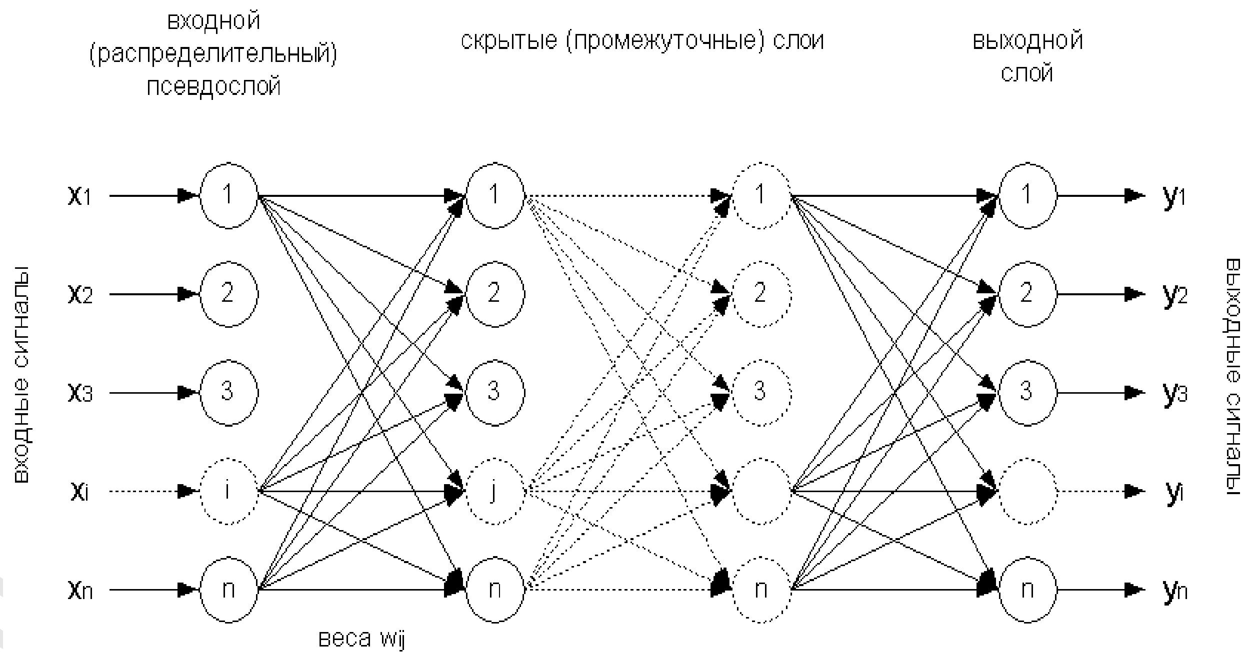


Рисунок II.4.4. Структура многослойного персептрона.

Если скрытых слоев больше, чем два, то во всех скрытых слоях одинаковое количество нейронов. При наличии большего числа скрытых элементов появляется возможность создавать еще больше выступов различных размеров, расположенных в разных местах.

Входные сигналы принимаются нейронами входного слоя,передаются нейронам первого скрытого слоя, и так далее вплоть до выходного, который выдает выходные сигналы всей нейронной сети. Каждый выходной сигнал *k*-го слоя подается на вход всех нейронов (*k+1*)-го слоя, образуя последовательные связи. Однако возможен скачок, когда *k*-ый слой соединяется с произвольным слоем ($k+q$), где $k>1$ & $q \in N$.

Введем обозначения: W^k – матрица весов синаптических связей, соединяющих нейроны *k-1*-го слоя с нейронами *k*-го слоя; w_{ij}^k – вектор-столбец связи, соединяющий *i*-й нейрон *k*-1-го слоя с *j-м* нейроном *k*-го слоя $(y_j^{k-1} \in y^{k-1}, y_j^k \in y^k)$.

Аналогично x_j^1 – входной вектор-столбец слоя 1. Тогда прямое функционирование многослойного персептрона описывается соотношениями:

$$s_j^k = f\left(\sum_{i=1}^{n(k-1)} w_{ij}^k y_i^{k-1}\right),\ \ y_j^k = f(s_j^k + w_{0j}^k),\ \ x_{ij}^{k+1} = y_j^k$$

где i – номер входа, j – номер нейрона в слое, k – номер слоя;

x_{ij}^k – i-й входной сигнал j -го нейрона в слое k ;

w_{ij}^k – вес синапса i-го входа j-го нейрона в слое k;

s_j^k – взвешенная сумма j-го нейрона в слое k;

w_{0j}^k – порог j-го нейрона в слое k.

Каждый слой рассчитывает нелинейное преобразование от линейной комбинации сигналов предыдущего слоя. Многослойная

НИИ искусственный интеллект

нейронная сеть может формировать на выходе произвольную многомерную функцию при соответствующем выборе количества слоев, диапазона изменения сигналов и параметров нейронов. Многослойные нейронные сети являются универсальным аппроксиматором функций. Общая формула, описывающая многослойный персептрон, определяется как:

$$f(s) = f\left(\underbrace{\sum_{i_K} w^{K}_{i_K j_K} \ldots f\left(\underbrace{\sum_{i_2} w^{2}_{i_2 j_2} f\left(\underbrace{\sum_{i_1} w_{i_1 j_1}}_{\text{слой 1}} + w_{j_1 2} \right)}_{\text{слой 2}} + w_{j_2 2} \right)}_{\text{слой K}} \ldots + w_{j_K K} \right)$$

За счет поочередного расчета линейных комбинаций и нелинейных преобразований достигается аппроксимация произвольной многомерной функции при соответствующем выборе параметров сети. Персептроны дают возможность строить более сложные разделяющие поверхности, вследствие чего, и более распространены.

Многослойные персептроны имеют три отличительных признака. Каждый нейрон имеет нелинейную функцию активации. Данная нелинейная функция является гладкой (всюду дифференцируемой), в отличие от жесткой пороговой функции, используемой в персептроне Розенблатта. Самой популярной формой функции, удовлетворяющей этому требованию, является сигмоидальная, определяемая логистической функцией:

$$f(s) = \frac{1}{1+e^{-s}}.$$

Наличие нелинейности играет очень важную роль, так как в противном случае отображение «вход-выход» сети можно свести к обычному однослойному персептрону. Более того, использование логистической функции мотивировано биологически, так как в ней учитывается восстановительная фаза реального нейрона.

НИИ искусственный интеллект

Несмотря на то, что возможности многослойных сетей были известны давно, в течение многих лет не было теоретически обоснованного алгоритма для настройки их весов. Далее мы детально изучим многослойные обучающие алгоритмы, на данном этапе достаточно понимать проблему и знать, что исследования привели к определенным результатам.

Многослойные нейронные сети с прямыми связями обучаются с помощью дельта-правила, которое используется при обучении с учителем и реализуется следующим образом:

$$w_{ij}(t+1)=w_{ij}(t)+\eta\, x_i(d_j-y_j),$$

где η – параметр (шаг обучения); d – эталонное (требуемое) значение выхода элемента..

Таким образом, изменение силы связей происходит в соответствии с ошибкой выходного сигнала $(d-y)$ и уровнем активности входного элемента x. Обобщение дельта-правила, называемое обратным распространением ошибки (Back – propagation), применимо к сетям с любым числом слоев. Обучение сети в этом случае состоит из следующих шагов:

1. Выбрать очередную обучающую пару (x, d). Подать входной вектор на вход сети.
2. Вычислить выход сети у.
3. Вычислить разность между выходом сети и требуемым выходом (ошибку).
4. Подкорректировать веса сети так, чтобы минимизировать ошибку.
5. Повторять шаги с 1-го по 4-й для каждой обучающей пары, пока ошибка не достигнет приемлемого уровня.
6. Ошибка функционирования сети обычно определяется как

$$E=\frac{1}{2}\cdot\sum_{j=1}^{n(k)}(d_j-y_j)^2,$$

где $y_j=y_j^k$ – выход сети. Для уменьшения этой ошибки следует изменить веса сети по правилу

НИИ Искусственный интеллект

$$w^k(t+1)=w^k(t)-\eta\cdot\frac{dE}{d\,w^k},$$

Для выходного слоя определение ошибки δ_j нейрона j тривиально:

$$\delta_j=(d_j-y_j)\cdot f'(s_j)=(d_j-y_j)\cdot C\cdot f(s_j)\cdot(1-f(s_j))$$

и напоминает систему поощрений-наказаний, используемую при обучении однослойных сетей. Для каждого предыдущего слоя ошибка определяется рекурсивно через ошибку следующего слоя:

$$\delta_j=(\sum_{j=1}^{m}\delta_i w_{ij})\cdot f'(s_j)\ ,$$

т.е. для каждого j-го нейрона ошибки следующего слоя распространяются к нему обратно сквозь соответствующие веса. Этот механизм обратного распространения дополнен традиционными для многих градиентных методов оптимизации процедурами оценки вектора кратчайшего спуска, изменения величины шага пропорционально крутизне склона и др.

Если при прямом функционировании входной сигнал распространяется по сети от входного слоя к выходному, то при подстройке весов ошибка сети распространяется от выходного слоя к входному.

Задания II.4.4.

1. Опишите функционирование многослойного персептрона.
2. Напишите общую формулу многослойного персептрона.
3. Напишите формулу функции активации многослойного персептрона.
4. Напишите формулу вычисления ошибки обучения многослойного персептрона.

Вопросы II.4.4.

1. Что будет с количеством нейронов, если количество слоев в персептроне будет больше, чем два?
2. Куда подается каждый выходной сигнал k-го слоя в многослойном персептроне?

НИИ Искусственный интеллект

3. Как уменьшается ошибка в многослойном персептроне?
4. Как распространяется ошибка в многослойном персептроне?

НИИ искусственный интеллект

II.4.5. Нейронные сети Кохонена и самоорганизующиеся карты

В этом параграфе будут рассмотрены нейронные сети Кохонена и самоорганизующиеся карты, предложены примеры, даны задания, сформулированы вопросы. При подготовке учебного материала данного параграфа были использованы источники [1-9].

Нейронные сети Кохонена представляют собой отдельный класс нейронных сетей и относятся к самоорганизующимся нейронным сетям. Нейронная сеть Кохонена является однослойной сетью, каждый нейрон которой соединен со всеми компонентами n-мерного входного вектора. Входной вектор – это описание одного из объектов, подлежащих кластеризации. Количество нейронов m совпадает с количеством кластеров, которое должна выделить сеть. В качестве нейронов сети Кохонена применяются линейные взвешенные сумматоры. Входной вектор сети имеет вид $X=(x_1,x_2,\dots,x_n)$. Каждый j-ый нейрон описывается вектором весов синаптических связей в n-мерном пространстве $W_j=(w_{1j},w_{2j},\dots,w_{nj})$.

Структура сети Кохонена представлена на рисунке II.4.5.

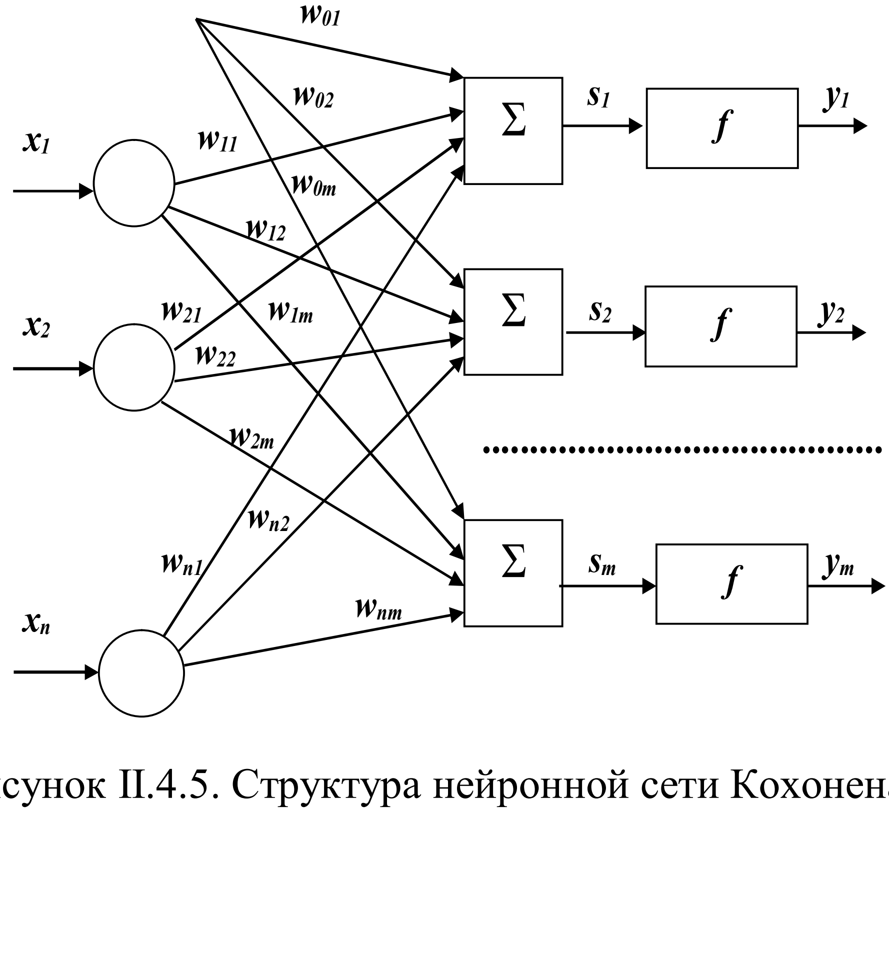


Рисунок II.4.5. Структура нейронной сети Кохонена.

Сеть Кохонена состоит из m параллельно действующих линейных элементов. Все они имеют одинаковое число входов n и получают на свои входы один и тот же вектор входных сигналов $X=(x_1,x_2,...,x_n)$. Выходной сигнал j-го линейного элемента будет:

$$y_j=w_{0j}+\sum_{i=1}^{n} w_{ij}x_i,$$

где w_{0j} – пороговый коэффициент; w_{ij} – весовой коэффициент i-го входа j-го нейрона; $i=1, \ldots, n$ – номер входного сигнала; $j=1, m$ – номер линейного элемента.

В сетях Кохонена используется обучение без учителя. Для обучения сети применяются *механизмы конкуренции*. При подаче на вход значения вектора X побеждает тот нейрон, вектор весов которого в наименьший степени отличаются от входного вектора. Для нейрона-победителя выполняется соотношение

$$d(x,W_j)=\min_{1\le j\le m} d(x,W_j),$$

m – Количество нейронов, j – номер нейрона-победителя, $d(x,W_j)$ – расстояние (в смысле выбранной метрики) между векторами x и W_j.

В качестве меры расстояния используется *евклидова мера* $\rho_j(x)$ до точек W_j.

$$\rho_j(x)=d(x,W_j)=\left\|x-W_j\right\|^2=\left\|x\right\|^2-2\sum_{j=1}^{m} x_j\cdot w_{ij}+\left\|W_j\right\|^2$$

Здесь $\|y\|$ - Евклидова длина вектора: $\|y\|^2=\sum_{j=1}^{m} y_j^{\,2}$. Первое слагаемое одинаково для всех нейронов, поэтому для нахождения ближайшей точки оно не нужно.

После прохождения слоя линейных элементов сигналы посылаются на обработку по правилу «победитель забирает всё». Среди выходных сигналов выполняется поиск максимального, определяется его номер. Окончательно, на выходе сигнал с найденным номером равен единице, остальные – нулю. Если максимум одновременно достигается для нескольких, то:

- либо принимают все соответствующие сигналы равными единице;
- либо равным единице принимают только первый сигнал в списке (по соглашению).

Задача сводится к поиску номера наибольшего из значений линейных функций, т.е.:

$$j_{\max} = \arg \max_{j} \{y_j\} = \arg \max_{j} \left\{ \sum_{j=1}^{m} x_j \cdot w_{ij} - \frac{1}{2} \|W_j\|^2 \right\} .$$

Таким образом, координаты точки $W_j = (w_{1j}, w_{2j}, \ldots, w_{nj})$ совпадают с весами линейного слоя Кохонена, при этом значение порогового коэффициента $w_{0j} = \frac{1}{2}\|W_j\|^2$.

Вокруг нейрона-победителя образуется *окружение* (*neighborhood*), или радиус обучения (*radius of learning*). Радиус обучения определяет сколько нейронов кроме нейрона-победителя участвуют в обучении (корректируют свои веса) на данной итерации. Под радиусом в данном случае подразумевается расстояние в пространстве векторов весов нейронов. То есть, любой нейрон, расстояние от вектора весов которого до вектора весов нейрона-победителя меньше, чем радиус обучения, участвует в коррекции весов на данной итерации. Радиус обучения максимален на данной итерации и уменьшается с увеличением числа итерации таким образом, что в конце обучения корректирует свои веса только нейрон-победитель.

Веса нейрона-победителя и всех нейронов, лежащих в пределах его радиуса обучения, подвергаются обучению по правилу Кохонена. При этом выделяют сети с неупорядоченными нейронами, называемые слоями Кохонена, и сети с

НИКТ

упорядоченными нейронами, называемые самоорганизующимися картами (Self-Organizing Maps – SOM) Кохонена.

Слой Кохонена состоит из адаптивных линейных сумматоров («линейных формальных нейронов»). Как правило, выходные сигналы слоя Кохонена обрабатываются по правилу «Победитель получает всё», наибольший сигнал превращается в единичный, остальные обращаются в ноль.

«Нейроны Кохонена можно воспринимать как набор электрических лампочек, так что для любого входного вектора загорается одна из них».

Сети векторного квантования. Задача векторного квантования с k кодовыми векторами W_j для заданной совокупности входных векторов X ставится как задача минимизации искажения при кодировании, то есть при замещении каждого вектора из X соответствующим кодовым вектором. В базовом варианте сетей Кохонена используется метод наименьших квадратов и искажение D вычисляется по формуле

$$D=\sum_{i=1}^{n}\sum_{x\in K_j}^{n}\left\|x-W_j\right\|^2,$$

где K_j состоит из тех точек $x\in X$, которые ближе к W_j, чем к другим W_i $(i\neq j)$. Другими словами, K_j состоит из тех точек $x\in X$, которые кодируются кодовым вектором W_j.

Если совокупность X задана и хранится в памяти, то стандартным выбором в обучении соответствующей сети Кохонена является метод K-средних. Это метод расщепления:

- при данном выборе кодовых векторов (они же весовые векторы сети) W_j минимизацией D находим множества K_j, которые состоят из тех точек $x\in X$, которые ближе к W_j, чем к другим W_i;

• при данном разбиении X на множества K_j минимизацией D находим оптимальные позиции кодовых векторов W_j, для оценки по методу наименьших квадратов это просто средние арифметические:

$$W_j = \frac{1}{|K_j|}\sum_{x \in K_j}^{n} x,$$

где $|K_j|$ – число элементов в K_j.

Далее итерируем. Этот метод расщепления сходится за конечное число шагов и даёт локальный минимум искажения.

Если же, например, совокупность X заранее не задана, или по каким-либо причинам не хранится в памяти, то широко используется онлайн метод. Векторы входных сигналов x обрабатываются по одному, для каждого из них находится ближайший кодовый вектор («победитель», который «забирает всё») $W_j(x)$. После этого данный кодовый вектор пересчитывается по формуле:

$$W_{j(x)}^{new} = W_{j(x)}^{old}(1-\theta) + \theta x,$$

где $\theta \in (0,1)$ – шаг обучения. Остальные кодовые векторы на этом шаге не изменяются.

Для обеспечения стабильности используется онлайн метод с затухающей скоростью обучения. Если T – количество шагов обучения, то полагают $\theta = \theta(T)$. Функцию $\theta(T) > 0$ выбирают таким образом, чтобы монотонно при $T \to \infty$ ряд $D = \sum_{T=1}^{n} \theta(T)$ расходился, например, $\theta(T) = \frac{\theta_0}{T}$.

Векторное квантование является намного более общей операцией, чем кластеризация, поскольку кластеры должны быть разделены между собой, тогда как совокупности K_j для разных кодовых векторов W_j не обязательно представляют собой раздельные кластеры. С другой стороны, при наличии разделяющихся кластеров векторное квантование может находить их и по-разному кодировать.

По способам настройки входных весов сумматоров и по решаемым задачам различают много разновидностей сетей Кохонена. Наиболее известные из них:

- сети векторного квантования сигналов, тесно связанные с простейшим базовым алгоритмом кластерного анализа (метод динамических ядер или K-средних);
- сети векторного квантования, обучаемые с учителем (Learning Vector Quantization).

Самоорганизующиеся карты Кохонена (Self Organizing Maps – SOM) представляют собой соревновательную нейронную сеть с обучением без учителя, выполняющую задачу визуализации и кластеризации многомерных векторов.

Примером таких алгоритмов может служить алгоритм *k*-ближайших средних (*k*-means). Важным отличием алгоритма SOM является то, что в нем все нейроны (узлы, центры классов…) упорядочены в некоторую структуру (обычно двумерную сетку). При этом в ходе обучения модифицируется не только нейрон-победитель, но и его соседи, но в меньшей степени. За счет этого SOM можно считать одним из методов проецирования многомерного пространства в пространство с более низкой размерностью (чаще всего, двумерное), применяется также для решения задач моделирования, прогнозирования и др. Является одной из версий нейронных сетей Кохонена.

При использовании этого алгоритма вектора, схожие в исходном пространстве, оказываются рядом и на полученной карте.

Сигнал в карте Кохонена поступает сразу на все нейроны, веса соответствующих синапсов интерпретируются как координаты положения узла, и выходной сигнал формируется по принципу «победитель забирает всё» – то есть ненулевой выходной сигнал имеет нейрон, ближайший (в смысле весов синапсов) к подаваемому на вход объекту. В процессе обучения веса синаптических связей настраиваются таким образом, чтобы узлы решетки «располагались» в местах локальных сгущений данных, то есть описывали кластерную структуру облака данных, с другой стороны, связи между нейронами соответствуют отношениям соседства между соответствующими кластерами в пространстве признаков.

Удобно рассматривать такие карты как двумерные сетки узлов, размещенных в многомерном пространстве. Изначально самоорганизующаяся карта представляет из себя сетку из узлов, соединенный между собой связями. Кохонен рассматривал два варианта соединения узлов – в прямоугольную и гексагональную сетку – отличие состоит в том, что в прямоугольной сетке каждый узел соединен с 4-мя соседними, а в гексагональной – с 6-ю ближайшими узлами. Для двух таких сеток процесс построения сети Кохонена отличается лишь в том месте, где перебираются ближайшие к данному узлу соседи. Начальное вложение сетки в пространство данных выбирается произвольным образом. В авторском пакете SOM_PAK предлагаются варианты случайного начального расположения узлов в пространстве и вариант расположения узлов в плоскости. После этого узлы начинают перемещаться в пространстве согласно следующему алгоритму:

1. Случайным образом выбирается точка данных x.

2. Определяется ближайший к x узел карты (BMU – Best Matching Unit).

3. Этот узел перемещается на заданный шаг по направлению к x. Однако, он перемещается не один, а увлекает за собой определенное количество ближайших узлов из некоторой окрестности на карте. Из всех двигающихся узлов наиболее сильно смещается центральный,ближайший к точке данных,узел, а остальные испытывают тем меньшие смещения, чем дальше они от BMU. В настройке карты различают два этапа – этап грубой

НИИ Искусственный Интеллект

(ordering) и этап тонкой (fine-tuning) настройки. На первом этапе выбираются большие значения окрестностей и движение узлов носит коллективный характер,в результате карта «расправляется» и грубым образом отражает структуру данных. На этапе тонкой настройки радиус окрестности равен 1-2 узла и настраиваются уже индивидуальные положения узлов. Кроме этого, величина смещения равномерно затухает со временем, то есть она велика в начале каждого из этапов обучения и близка к нулю в конце.

4. Алгоритм повторяется определенное число эпох (понятно, что число шагов может сильно изменяться в зависимости от задачи).

Самоорганизующиеся карты Кохонена служат для визуализации и первоначального («разведывательного») анализа данных.

Задания II.4.5.

1. Нарисуйте структуру нейронной сети Кохонена.
2. Напишите формулу вычисления выходного сигнала j-го линейного элемента нейронной сети Кохонена.
3. Перечислите разновидности нейронных сетей Кохонена.
4. Опишите состав слоя Кохонена.
5. Напишите формулу для выходного сигнала линейного элемента в слое Кохонена.
6. Назовите число параллельно действующих линейных элементов в слое Кохонена.
7. Напишите формулу поиска номера наибольшего значений линейных функций.

Вопросы II.4.5.

1. К какому типу относятся нейронные сети Кохонена?
2. Чему равно количество нейронов в нейронной сети Кохонена?
3. По какому правилу наибольший сигнал превращается в единичный, остальные обращаются в ноль?
4. Какой парадигмой обучаются нейронные сети Кохонена?
5. Какой механизм используется для обучения сети Кохонена?

НИИ искусственный интеллект

6. В чем состоит разница между сетью векторного квантования и самоорганизующиеся картой Кохонена?
7. По какому правилу обрабатываются выходные сигналы слоя Кохонена?

НИИ Искусственный интеллект

II.4.6. Комбинированные нейронные сети

В этом параграфе будут рассмотрены комбинированные нейронные сети, предложены примеры, даны задания, сформулированы вопросы. При подготовке учебного материала данного параграфа были использованы источники [1-9].

Комбинированные нейронные сети объединяют возможности разных типов нейронных сетей.

Одной из наиболее популярных комбинированных сетей, использующихся на практике, является сеть встречного распространения (Counter Propagation Network), первым слоем которой является сеть Кохонена, вторым – сеть Гроссберга.

Структура сети встречного распространения представлена на рисунке II.4.6.1.

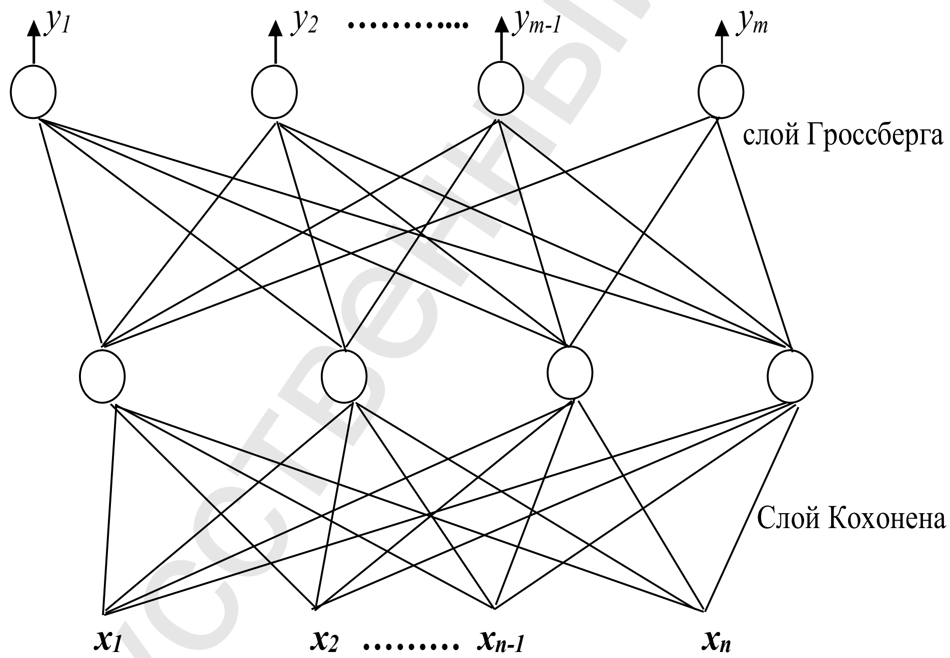


Рисунок II.4.6.1. Структура сети встречного распространения.

Слой Гроссберга представляет собой конфигурацию входных и выходных звёзд Гроссберга.

Нейрон в форме входной звезды имеет n входов, которым соответствуют весовые коэффициенты $W=(w_1,w_2,\ldots,w_n)$ и один выход Y, являющийся взвешенной суммой входов. Входная звезда

НИИ искусственный интеллект

обучается выдавать сигнал на выходе всякий раз, когда на входы подается определенный входной вектор. Структура входной звезды Гроссберга представлена на рисунке II.4.6.2.

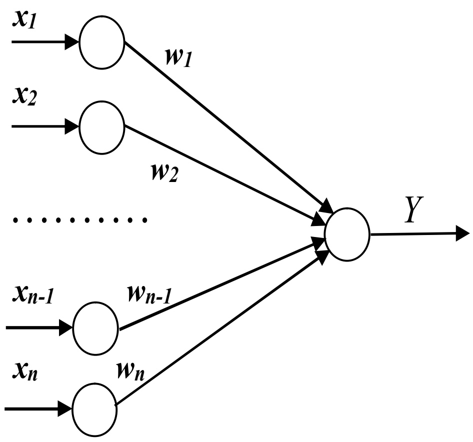


Рисунок II.4.6.2. Структура входной звезды.

Расчет весовых коэффициентов производится по формуле:

$$W_i(t+1)=W_i(t)+\mu(X_i-W_i(t)),$$

где $W_i(t)$ – весовой вектор *i*-той входной звезды на *t*-м такте обучения; X_i – входной вектор; μ – скорость обучения, которая имеет начальное значение в пределах 0,1…0,2 и постепенно уменьшается в процессе обучения.

Выходная звезда Гроссберга выполняет противоположную функцию – при поступлении сигнала на вход выдается определённый вектор. Нейрон этого типа имеет один вход и *m* выходов с весами $W=(w_1,w_2,\ldots,w_m)$, которые подстраиваются в соответствии с формулой:

$$W_i(t+1)=W_i(t)+\eta(Y_i-W_i(t)),$$

где $W_i(t)$ – весовой вектор *i*-той выходной звезды на *t*-м такте обучения; Y_i – входной вектор; η – скорость обучения.

Рекомендуется начать обучение со значения η в пределах единицы, постепенно уменьшая до значений, близкий к нулю.

НИИ искусственный интеллект

Обучение сети встречного распространения состоит из двух фаз:

1. Обучение слоя Кохонена.

В слое Кохонена нейроны используют конкурентное обучение с выявлением нейрона победителя на каждом шаге обучения. В качестве метрики используется евклидово расстояние:

$$D=\left|X-W_i\right|=\sqrt{\left(x_1-w_{1i}\right)^2+\left(x_2-w_{2i}\right)^2+\ldots+\left(x_n-w_{ni}\right)^2}\ ,$$

$D_j=\min\limits_i \left|X-W_i\right|$ – опрделяется нейрон-победитель с номером j.

Здесь i – номер входного вектора, j – номер нейрона победителя.

Выходное значение нейронов в слое Кохонена вычисляется как:

$$y_j=\begin{cases}1, \ D_j=\min\limits_i \left|X-W_i\right| \\ 0, \ \text{иначе}\end{cases}$$

Производится модификация весовых коэффициентов нейрона-победителя в соответствии со следующими:

– при корректной классификации весовые коэффициенты нейрона-победителя усиливаются $W_{ij}(t+1)=W_{ij}(t)+\gamma(X_i-W_{ij}(t))$;

– при некорректной классификации весовые коэффициенты нейрона-победителя ослабляются $W_{ij}(t+1)=W_{ij}(t)-\gamma(X_i-W_{ij}(t))$.

Выходной слой обучается с учителем путем нахождения оптимальных весовых коэффициентов связи выходного нейрона слоя Кохонена и i-того входа нейрона слоя Гроссберга V_i так, чтобы сеть наилучшим образом аппроксимировала воспроизводимую функцию $f(X_i)$. Для этого используется алгоритм обратного распространения ошибки и минимизируется ошибка обучения методом наименьших квадратов:

$$E_0=\frac{1}{2}\cdot\left[V_i-\alpha\left(V_i(t)-f(X_i)\right)\right],$$

НИИ искусственный интеллект

где α – скорость обучения: $\alpha \in [0,1]$, $f(X_i)$ – эталонное значение функции для данного входного вектора.

Ошибка обучения вычисляется не для одного вектора, а для ряда векторов, которые входят в кластер, соответствующего определенному нейрону.

2. Обучение слоя Гроссберга.

В слое Гроссберга нейроны используют обучение с учителем и линейную функцию активации, а выходное значение нейронов вычисляется по формуле:

$$z_i=\sum_{i=1}^{m} V_i y_i \ ,$$

где m – число выходных нейронов, V_i– весовые коэффициенты выходного нейрона слоя Кохонена и i-того входа нейрона слоя Гроссберга.

Таким образом, нейронная сеть встречного распространения каждому i-тому кластеру промежуточного слоя ставит в соответствии весовой коэффициент V_i. Это может быть использовано для выполнения аппроксимации функции.

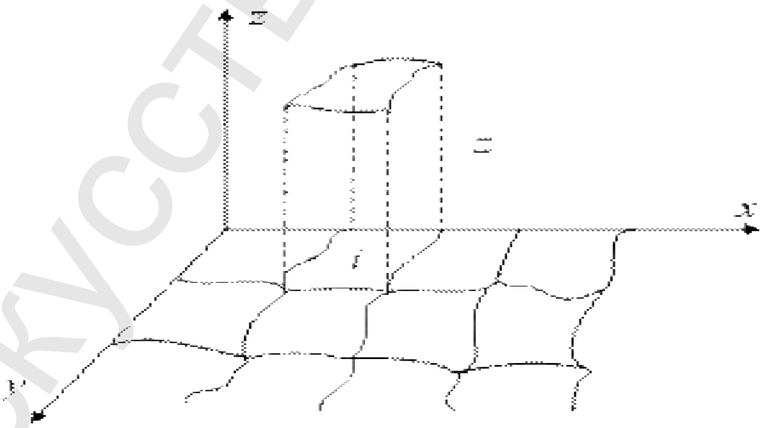


Рисунок II.4.6.3 - Разбиение входного пространства на кластеры

Например, карта Кохонена (плоскость xy) соответствует разбиению входного пространства на кластеры (см. рисунок II.4.6.3). Тогда аппроксимации функции будет соответствовать высота z: $z=f(x,y)$. Для реализации нелинейной аппроксимации

НИИ Искусственный интеллект

функции в качестве функции активации нейронов выходного слоя необходимо брать нелинейную функцию (например, сигмоидальную).

Замечание. Процедуры обучения и функционирования сети могут производится параллельно. Для этого вводится коэффициент бдительности ρ, который характеризует допустимую степень отклонения входного образа от образов, хранящихся в сети. Для каждого входного образа вычисляется расстояние D_j, которое характеризует отличие входного образа от образа на сети. Если $D_j > \rho$, то создаётся новый класс, иначе, производится обучение нейронной сети.

Задания II.4.6.

1. Назовите тип элемента первого слоя и второго слоя в нейронной сети встречного распространения.
2. Нарисуйте структуру нейронной сети встречного распространения.
3. Нарисуйте структуру входной звезды.
4. Напишите формулу для постройки весовых коэффициентов для входной звезды.
5. Напишите формулу для постройки весовых коэффициентов для выходной звезды.

Вопросы II.4.6.

1. Как образуются комбинированные нейронные сети и какими возможностями они обладают?
2. Что из себя представляет слой Гроссберга в нейронной сети встречного распространения?
3. Какую функцию выполняет выходная звезда Гроссберга?
4. Как обучается слой Кохонена в нейронной сети встречного распространения?

НИИ искусственный интеллект

5. Какая метрика используется для обучения слоя Кохонена в нейронной сети встречного распространения?

НИИ Искусственный интеллект

III. ОБУЧЕНИЕ НЕЙРОННЫХ СЕТЕЙ

III.1. Парадигмы обучения нейронных сетей

III.1.1. Основные понятия нейронных сетей

В этом параграфе нами будут рассмотрены понятия парадигмы и методов обучения искусственных нейронных сетей, предложены задания и сформулированы вопросы. При подготовке учебного материла были использованы литературные источники [14, 17, 22, 27, 28, 33].

По своей организации и функциональному назначению искусственные нейронные сети (ИНС) с несколькими входами и выходами выполняет некоторое преобразование входных сигналов (сенсорной информации о внешнем мире) в выходные управляющие сигналы. При этом число преобразуемых сигналов равно числу входов ИНС n, а число выходных сигналов соответствует числу выходов m. Совокупность всевозможных *входных* векторов размерности n образует векторное пространство X, которое будет называться *входным признаковым пространством.* Аналогично, *выходные* вектора также формируют *выходное признаковое пространство* Y. Теперь ИНС можно считать, как некоторую многомерную *функцию* F: $X \rightarrow Y$, аргумент которой принадлежит входному признаковому пространству, а значение – выходному признаковому пространству. Вид этой функции определяется ее архитектурой, в которой задаются виды и количество нейронов, синаптические связи (синапсы), значения коэффициентов синаптических связей (весов), смещений и функций активации.

Для решения какой-либо задачи с помощью ИНС необходимо пройти *этап обучения* ИНС, в котором для каждого входного вектора подбирается парный ему целевой вектор, задающий желаемый выход. Вместе они называются *обучающим примером.* Как правило, ИНС обучается на многих обучающих примерах, в которых содержится вся информация о задаче. Поэтому качество обучения ИНС зависит от количества обучающих примеров и от

НИИ искусственный интеллект

полноты описания задачи этими примерами. Считается, что для полноценной тренировки требуется несколько сотен примеров.

Процесс обучения ИНС представлен на рисунке III.1.1.

НИИ Искусственный интеллект

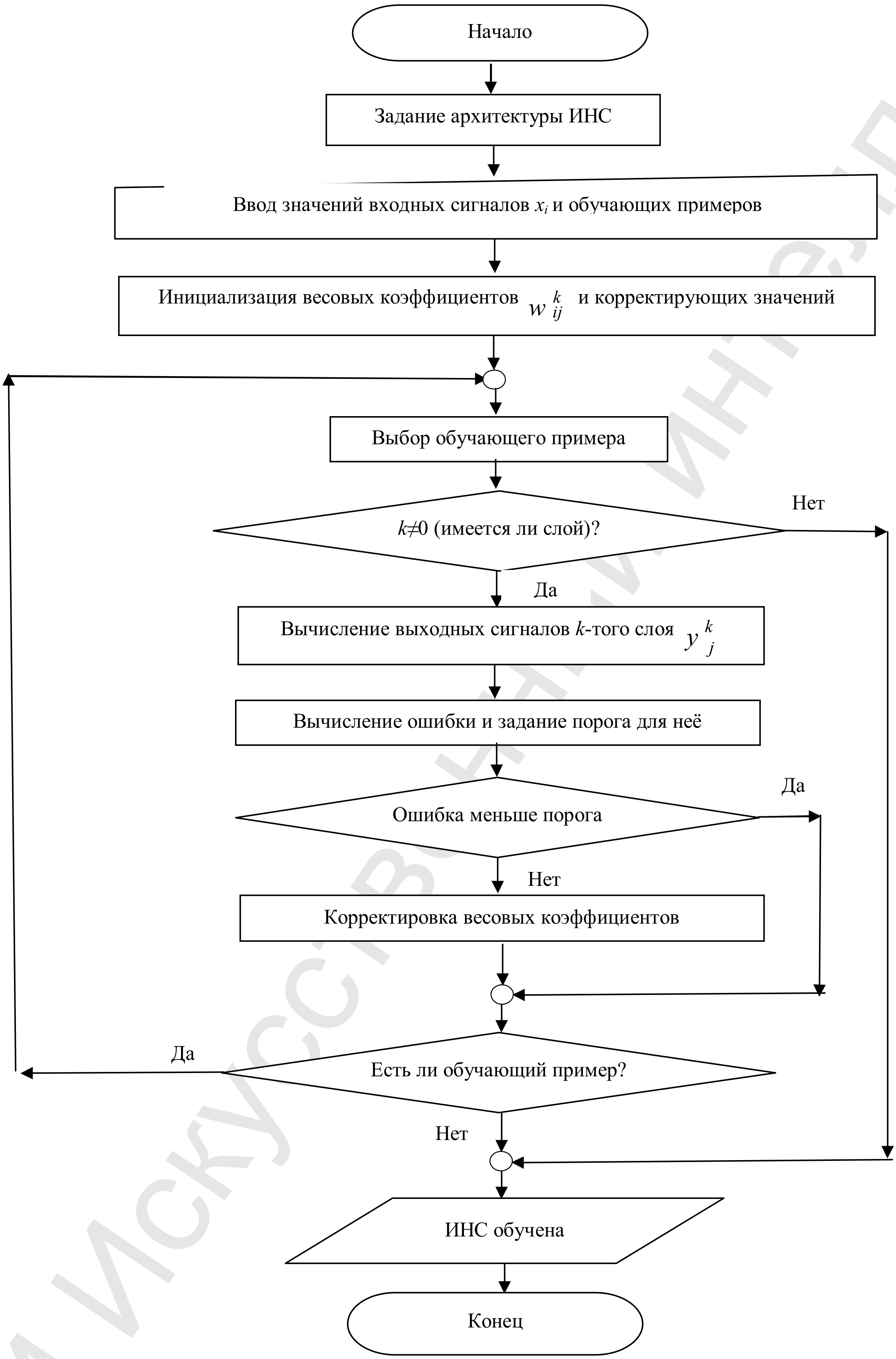


Рисунок III.1.1. Процесс обучения ИНС.

НИИ искусственный интеллект

Прежде чем начать процесс обучения необходимо построить обучаемую ИНС, предварительно определив задачи, которые будут решаться с ее помощью, следующим образом:

- на первом этапе определяется архитектура ИНС, в которой указываются её структура, число слоев и число нейронов;
- на втором этапе осуществляется отбор входных данных, влияющий на ожидаемый результат с наименьшей ошибкой;
- на третьем этапе выбирается способ представления информации, осуществляется преобразование исходных данных с учетом характера и типа проблемы и определяются выходные переменные ИНС;
- на четвертом этапе генерируется ИНС и производится её обучение на заданных примерах, которое может проводиться на основе конструктивного или деструктивного подхода.

В процессе обучения ИНС задается архитектура, вводятся значение входного вектора, выбирается обучающий пример и находятся оптимальные значения всех весов этой ИНС, чтобы на выходе генерировались желаемые результаты. В этом случае говорят, что ИНС обладает *способностью обучаться*, т.е. обладает *свойством обучаемости*, которое делает ИНС незаменимой при решении задачи, для которой алгоритмизация является невозможной, проблематичной или слишком трудоемкой.

Теперь можно описать технологию обучения ИНС, в которой обучение проводится на множестве *обучающих данных*, а проверка правильности работы ИНС – на множестве *тестовых данных*, которые не используется при ее обучении. Технология обучения ИНС состоит из двух этапов:

1) этап обучения ИНС, в котором ИНС предъявляется множество обучающих пар (примеров) до тех пор, пока не будет выполнено условие остановки обучения:

– вычисляемая функция ошибки становится меньше заданной (т.е. функционирование ИНС улучшается по мере итеративной настройки весов синаптических связей) или перестает изменяться в

НИИ интеллект

течение определенного числа итерации (обучение ИНС заканчивается);

– по истечению заданного числа итерации (обычно это число находится в интервале от 10^3 до 10^8);

2) этап проверки правильности работы ИНС, в котором ИНС предъявляется множество тестов и при этом, если ошибка обобщения больше заданной ошибки обобщения, то проводится:

– либо уменьшение шага (увеличение числа) итерации;

– либо увеличение числа обучающих примеров;

– либо модификация архитектуры (изменение состава и структуры) ИНС.

Перед началом обучения всем весам должны быть присвоены небольшие начальные значения, выбранные случайным образом. Это гарантирует, что в ИНС не произойдет насыщения большими значениями весов, а также предотвращает ряд других случаев. Например, если всем весам придать одинаковые начальные значения, а для требуемого функционирования нужны неравные значения, то ИНС не сможет обучиться.

В программных реализациях можно видеть, что в процессе обучения величина ошибки (сумма квадратов ошибок по всем выходам) постепенно уменьшается. Когда величина ошибки достигает нуля или приемлемо малого уровня, обучение останавливают, и ИНС готова к распознаванию.

От того, насколько качественным будет процесс обучения, зависит способность ИНС корректно решать поставленные перед ней задачи во время функционирования.

Веса и пороговые уровни инициализируются случайными значениями. Созданная таким образом ИНС абсолютно неадекватна решаемой задаче и может генерировать на выходе только шум. Если использовать небольшой набор обучающих примеров, то при обучении ИНС будет слишком близко следовать обучающим примерам (переобучаться) и воспринимать не столько структуру данных образов, сколько содержащиеся в ней шумы.

НИИ Искусственный интеллект

Поэтому ошибка в начале обучения очень велика, и есть смысл вводить большие коррекции параметров. Ближе к концу обучения ошибка значительно снижается, и коррекции должны быть малыми. Чтобы менять длину шагов по параметрам, используют *расписание обучения* (learning schedule). Выберем скорость обучения $\eta(t)$, зависящей от времени обучения t. Обычно скорость монотонно убывает с ростом времени. Для сходимости алгоритма необходимо:

$$\eta(t) \underset{t\to\infty}{\to} 0, \quad \int_0^\infty \eta(t)dt = +\infty .$$

Алгоритмы с расписанием обучения сходятся быстрее, чем алгоритмы без них, т.к. в начале используются большие коррекции, и дают более точные результаты за счет точной настройки параметров в конце обучения.

Способность ИНС не только учиться на множестве обучающих примеров, но и показывать хорошие результаты на новых данных называется *экстраполяцией* (обобщением). Качество обобщения данных можно определить, наблюдая за величиной ошибки, вычисленной на множестве тестов. Эта ошибка должна быть меньше заданной. Если после нескольких итераций обучения ошибка падает почти до нуля, в то время как ошибка обобщения сначала убывает, а потом снова начинает расти, то это признак переобучения, и при росте ошибки обобщения обучение следует прекратить. По окончании обучения следует проверить работу ИНС еще на третьем множестве – множестве *подтверждающих данных*.

Для определения процесса обучения, во-первых, необходимо иметь модель внешней среды, в которой функционирует ИНС – знать доступную для ИНС информацию, которая определяет парадигму обучения. Во-вторых, необходимо понять, как модифицировать веса ИНС,какими правилами обучения управлять процессом настройки, которые задаются методами обучения.

Емкость ИНС есть число образов, предъявляемых на ее входы, которые она способна научиться распознавать (запомнить), и какие функции и границы принятия решений могут быть на ней

НИУ «Искусственный интеллект»

сформированы, а также, сколько дополнительных переменных потребуются для организации процесса её функционирования. Емкость ИНС тесно связана с требуемой мощностью выходного слоя ИНС, выполняющего окончательную классификацию образов.

Сложность образов определяет число обучающих примеров, необходимых для достижения способности ИНС к обобщению. Слишком малое число примеров может вызвать "переобученность" ИНС, когда она хорошо функционирует на примерах обучающей выборки, но плохо – на новых (тестовых) примерах, подчиненных тому же статистическому распределению, и утрачивает способность обучаться. Вычислительная сложность определяет, за сколько число шагов можно обучить ИНС, чтобы получить приемлемое решение задачи. Вычислительная сложность обучения ИНС зависит от многих факторов: модели нейронов и количества нейронов, модели и количества слоев ИНС, функции активации, метода обучения.

Обучение ИНС может выполняться разными парадигмами и методами в соответствии с различными алгоритмами обучения.

Парадигма обучения определяется доступностью необходимой информации. Существуют три парадигмы обучения: *обучение с учителем* (*контролируемое обучение*); *обучение без учителя* (*неконтролируемое обучение – самообучение*); *смешанное обучение.*

Так как создание искусственного интеллекта движется по пути копирования природных прообразов, еще не прекращается спор на тему, можно ли считать парадигму обучения с учителем натуральными или же они полностью искусственны. Однако, нельзя отрицать и того, что в жизни человека не мало учителей, которые координируют внешние воздействия. Вместе в тем, чем бы не закончился спор приверженцев этих двух концепций обучения, они обе имеют право на существование.

Методы обучения ИНС предусматривают разработку алгоритма, в котором определяются правила обучения для настройки весов. Имеются следующие методы обучения: *метод обратного распространения ошибки, метод обучения Больцмана,*

НИИ искусственного интеллект

метод обучения Хебба, метод обучения Кохонена, метод обучения соревнованием, метод коррекции ошибки.

Примеры III.1.1:

1. Для ИНС с двумя слоями, емкость C_d оценивается как:

$$\frac{N_w}{N_y} < C_d < \frac{N_w}{N_y} \log\left(\frac{N_y}{N_y}\right),$$

где N_w – число весов, N_y – число нейронов в выходном слое.

2. Для разделения множества входных образов по двум классам достаточно одного выходного нейрона. При этом каждый логический уровень 0 или 1 будут обозначать отдельный класс. На двух выходных нейронах с пороговой функцией активации можно закодировать уже четыре класса. Для повышения достоверности классификации желательно ввести избыточность путем выделения в выходном слое каждому классу одного нейрона или нескольких, каждый из которых обучается определять принадлежность образа к классу со своей степенью достоверности.

3. При распознавании с помощью ИНС 26 латинских букв, нам нужно иметь 26 обучающих пар, входная часть каждой из которых будет состоять из двоичного образа некоторой буквы. Например, значение входного вектора для распознавания буквы «А» – первой буквы латинского алфавита будет иметь вид: X^1 = (010001111).

Задания III.1.1:

1. Опишите процесс обучения ИНС;
2. Опишите этапы построения обучаемой ИНС;
3. Определите интервал значений емкости для двух слойной ИНС с числом весов 16 и числом нейронов в выходном слое 4.

Вопросы III.1.1:

1. От чего зависит способность ИНС корректно решать поставленные задачи?
2. На каком множестве данных проверяется работа ИНС после завершения её обучения?
3. Что такое емкость ИНС, сложность образов, вычислительная сложность при обучении ИНС?

НИИ Искусственный интеллект

III.1.2. Парадигма обучения нейронных сетей с учителем

В этом параграфе нами будут рассмотрены вопросы обучения искусственных нейронных сетей (ИНС) с учителем, предложены задания и сформулированы вопросы. При подготовке учебного материла были использованы литературные источники [14, 17, 22, 27, 28, 33].

В общем случае, обучение с учителем предполагает предъявление обучающих примеров для оценивания и модификации состояния обучаемой системы. В случае ИНС, все обучающие примеры содержат значения входного вектора (выборки) и желаемые выходы (правильные ответы), а веса синаптических связей настраиваются так, чтобы она порождала ответы, наиболее близкие к правильным, то есть каждый обучающий пример состоит из пары: *входной вектор* и *целевой* (*желаемый выходной*) *вектор*. При предъявлении обучающего примера вычисляется реальный выходной вектор и сравнивается с соответствующим целевым вектором. Разность целевого вектора и вычисленного выходного вектора (ошибка) с помощью обратной связи подается на входы ИНС и веса изменяются так, чтобы по этим входам получать выходы максимально близкие к целевым выходам, т.е. минимизировать ошибку. Обычно ИНС обучается на некотором множестве обучающих примеров. Обучающие примеры предъявляются последовательно, вычисляются ошибки и веса настраиваются для каждого входного вектора до тех пор, пока ошибка по всему обучающему множеству не достигнет приемлемо низкого уровня.

Процесс обучения должен быть таким, чтобы ИНС обучалась на всем множестве обучающих примеров (образов) без пропусков того, что уже выучено, и ИНС предъявлялись все обучающие примеры прежде, чем выполняется коррекция весов синаптических связей. Необходимые изменения весов должны вычисляться на всем множестве обучающих примеров, а это требует дополнительной памяти. После ряда таких обучающих циклов веса сойдутся к минимальной ошибке. Этот метод может оказаться

НИИ искусственный интеллект

бесполезным, если ИНС находится в постоянно меняющейся внешней среде, так что второй раз один и тот же вектор может уже не повториться. В этом случае процесс обучения может никогда не сойтись, бесцельно блуждая или сильно осциллируя.

Пусть имеется ИНС, которая выполняет преобразование F: $X \rightarrow Y$ входных векторов из входного признакового пространства входов X в выходные вектора из выходного признакового пространства Y, т.е. F есть множество всех возможных функций, соответствующих заданной архитектуре ИНС.Тогда решить поставленную задачу с помощью этой ИНС означает построить конкретную функцию $f \in F$ и обучить ИНС – построить итерационную процедуру для подбора значений весов синаптических связей и смещений таких, чтобы функция ошибки E была минимальной для всех пар обучающих примеров (x_i, y_i),

$f_i(x_i) = y_i, \; x_i \in X, \; y_i \in Y, \; i = 1, 2, ..., K$.

Функция ошибки E, будет показывать для каждой из функций f_i степень близости к f.

ИНС находится в состоянии w из пространства состояний W. Рассмотрим полную ошибку ИНС в состоянии w:

$$E(w) = \sum_{i=1}^{K} [f_i(x_i w) - y_i]^2.$$

Отметим два свойства полной ошибки. Во-первых, ошибка $E(w)$ является функцией состояния w, определенной на пространстве состояний W. По определению, она принимает неотрицательные значения. Во-вторых, в некотором обученном состоянии w^*, в котором ИНС не делает ошибок на обучающей выборке, данная функция принимает нулевое значение. Следовательно, обученные состояния являются *точками минимума* введенной функции $E(w)$.

Функция ошибки E может иметь произвольный вид. Если выбраны множество обучающих примеров и способов вычисления функции ошибки, обучение ИНС превращается в задачу

НИИ Искусственный интеллект

многомерной оптимизации, для решения которой могут быть использованы следующие методы:

–локальной оптимизации с вычислением частных производных первого порядка (градиентный метод, методы с одномерной и двумерной оптимизацией целевой функции в направлении антиградиента, метод сопряженных градиентов, методы с учетом направление антиградиента на нескольких шагах алгоритма);

–локальной оптимизации с вычислением частных производных первого и второго порядка (метод Ньютона, методы оптимизации с разряженными матрицами Гессе, квазиньютоновские методы, метод Гаусса-Ньютона, метод Левенберга-Маркардта);

–стохастической оптимизации (метод поиска в случайном направлении, имитация отжига, метод Монте-Карло);

–глобальной оптимизации (метод перебора значений переменных, от которых зависит целевая функция).

В многослойных ИНС оптимальные выходные значения всех слоев, кроме последнего слоя, как правило, неизвестны. Если в ИНС число слоев больше трех, то невозможно ее обучить, руководствуясь только величинами ошибок на выходном слое. Одним из вариантов решения этой проблемы является разработка наборов выходных сигналов, соответствующих входным, для каждого слоя ИНС, что является очень трудоемкой операцией и не всегда осуществимо. Другой вариант – динамическая настройка весов, в ходе которой выбираются наиболее слабые синаптические связи и изменяются на малую величину в ту или иную сторону, и сохраняются только те изменения, которые повлекли уменьшение ошибки на выходе всей ИНС, что требует громоздких рутинных вычислений. Третий вариант – распространение сигналов ошибки от выходов ИНС к ее входам, в направлении, обратном прямому распространению сигналов в обычном режиме работы.

Таким образом, задача обучения ИНС является задачей поиска минимума функции ошибки в пространстве состояний, и, следовательно, для ее решения могут применяться стандартные методы теории оптимизации. Эта задача относится к классу

НИИ искусственный интеллект

многофакторных задач, так, например, для однослойного персептрона с N входами и M выходами речь идет о поиске минимума в $N \times M$-мерном пространстве.

На практике могут использоваться ИНС в состояниях с некоторым малым значением ошибки, не являющихся в точности минимумами функции ошибки. Другими словами, в качестве решения принимается некоторое состояние из окрестности обученного состояния W^*. При этом допустимый уровень ошибки определяется особенностями конкретной прикладной задачи, а также приемлемым для пользователя объемом затрат на обучение.

Самой простой функцией ошибкой является средняя квадратичная ошибка, определяемая как усредненная на K обучающих примерах сумма квадратов разностей между значениями желаемого выхода d_i и реально полученного выхода y_i для каждого обучающего примера i :

$$E(w) = \frac{1}{K} \cdot \sum_{i=1}^{K} (d_i - y_i)^2 .$$

Усиленный вариант обучения с учителем называется обучение с поощрением, в котором не указывается точное значение желаемого выхода, а выставляется оценка хорошо или плохо поработала система.

Задания III.1.2:

1. Укажите составляющие обучающих примеров.
2. Опишите способ предъявления обучающих примеров.
3. Укажите варианты решения задачи оптимизации выходных значений в многослойных ИНС.
4. Напишите формулу полной ошибки обучения ИНС.
5. Назовите самую простую функцию ошибки.

Вопросы III.1.2:

1. Что предполагает обучение с учителем?
2. Что происходят при предъявлении обучающего примера?
3. Что такое функция ошибок?

НИИ Искусственный интеллект

4. Какой задачей является задача обучения нейронной сети?
5. Что выставляется при усиленном обучении с учителем?

НИИ Искусственный интеллект

III.1.3. Парадигма обучения нейронных сетей без учителя

В этом параграфе нами будут рассмотрены вопросы обучения без учителя искусственных нейронных сетей (ИНС), предложены задания и сформулированы вопросы. При подготовке учебного материла были использованы литературные источники [14, 17, 22, 27, 28, 33].

Главной чертой обучения без учителя является его "самостоятельность". Процесс обучения, как и в случае обучения с учителем, заключается в подстраивании весов синаптических связей. Обучение без учителя используют тогда, когда не для всех примеров обучающей выборки известны правильные ответы. В таком случае предпринимаются попытки определения внутренней структуры поступающих в сеть данных с целью распределить образцы по классам (категориям), т.е. не требуются целевые выходы и сравнения с предопределенными правильными ответами.

ИНС предъявляются некоторые значения входного вектора и в ходе их обработки в ней изменяются количество нейронов и их взаимосвязи, т.е. в ней происходят процессы, приводящие к тому, что ИНС становиться способной решать какую-то задачу. Здесь веса ИНС подстраиваются так, чтобы получались согласованные выходы, т. е. чтобы предъявление достаточно близких входных векторов давало одинаковые выходы. В этом случае раскрывается внутренняя структура входных векторов или корреляции между образцами, что позволяет распределить образы по классам, т.е. процесс обучения выделяет статистические свойства множества значений входного вектора и группирует сходные входные векторы в классы. Предъявление на вход ИНС вектора из данного класса даст определенный выход, но до обучения невозможно предсказать, какой выход будет производиться данным классом входных векторов. Поэтому выходы ИНС должны трансформироваться в некоторую понятную форму, обусловленную процессом обучения. Это не является проблемой. Обычно не сложно идентифицировать связь между входом и выходом, установленную ИНС.

НИИ Искусственный интеллект

Подстройка весов синаптических связей может проводиться только на основании информации, доступной в нейроне, то есть его состояния и уже имеющихся весов синаптических связей.

Следует отметить, что вид откликов на каждый класс входных образов не известен заранее и будет представлять собой произвольное сочетание состояний нейронов выходного слоя, обусловленное случайным распределением весов на стадии инициализации. Вместе с тем, ИНС способна обобщать схожие образы, относя их к одному классу. Тестирование обученной ИНС позволяет определить топологию классов в выходном слое. Для приведения откликов обученной ИНС к удобному представлению можно дополнить сеть одним слоем, который, например, по алгоритму обучения однослойного персептрона необходимо заставить отображать выходные реакции ИНС в требуемые образы.

Некоторые алгоритмы обучения без учителя изменяют и структуру ИНС, то есть количество нейронов и их взаимосвязи. Такие преобразования правильнее назвать более широким термином – *самоорганизацией* (*self-organizing*), но этот перевод с английского, не очень удачен, так как, речь идет не об изменении структуры ИНС, а только о подстройке весов, которая может проводиться только на основании информации, доступной в нейроне, то есть об его состоянии и уже имеющихся весах синаптических связей.

После выбора из слоя q нейрона j с минимальным расстоянием

$$D_j = \sqrt{\sum_{i=0}^{p-1} (y_i^{(q-1)} - w_{ij})^2}, \qquad \text{(III.1.3.1)}$$

обучение (подстройка весов) происходит по одному из методов:

1. Метод Хебба: $w_{ij}(t) = w_{ij}(t-1) + \eta \cdot y_i^{(q-1)} \cdot y_j^{(q)}$, (III.1.3.2)
2. Метод Кохонена: $w_{ij}(t) = w_{ij}(t-1) + \eta \cdot \left[y_i^{(q-1)} - w_{ij}(t-1)\right]$, (III.1.3.3)

НИИ искусственный интеллект

Здесь j – индекс нейрона в слое q, i – индекс суммирования по нейронам слоя (q-1); p – размер векторов в p-мерном пространстве;

$y_i^{(q-1)}$ – выходное значение нейрона i слоя (q-1);

$y_i^{(q)}$ – выходное значение нейрона j слоя q;

$w_{ij}(t)$ и $w_{ij}(t-1)$ – вес синапса, соединяющего эти нейроны, на итерациях t и t-1 соответственно;

η – коэффициент скорости обучения;

i – индекс суммирования по нейронам слоя (q-1);

Выходы нейронов слоя (q-1) являются входными значениями для слоя q. Корень в формуле (III.1.3.1) брать не обязательно, т.к. важна лишь относительная оценка различных D_j.

Из вышеприведенной формулы (III.1.3.3) видно, что обучение сводится к минимизации разницы между входными сигналами нейрона, поступающими с выходов нейронов предыдущего слоя $y_i^{(q-1)}$, и его синаптическими весами.

Имеется также и дифференциальный метод обучения Хебба.

$$w_{ij}(t)=w_{ij}(t-1)+\eta\cdot\left[y_i^{(q-1)}(t)-y_i^{(q-1)}(t-1)\right]\cdot\left[y_j^{(q)}(t)-y_j^{(q)}(t-1)\right] \quad \text{(III.1.3.4)},$$

где $y_i^{(q-1)}(t)$ и $y_i^{(q-1)}(t-1)$ – выходное значение нейрона i слоя q-1 соответственно на итерациях t и t-1, $y_j^{(q)}(t)$ и $y_j^{(q)}(t-1)$ – то же самое для нейрона j слоя q.

Как видно из формулы (III.1.3.4), сильнее всего обучаются синапсы, соединяющие те нейроны, выходы которых наиболее динамично изменились в сторону увеличения.

Полный алгоритм обучения с применением вышеприведенных формул будет выглядеть так:

1. На стадии инициализации всем весовым коэффициентам присваиваются небольшие случайные значения.

2. На входы ИНС подается входной образ, и сигналы возбуждения распространяются по всем слоям согласно принципам классических прямопоточных (feedforward) сетей, то есть для

каждого нейрона рассчитывается взвешенная сумма его входов, к которой затем применяется функция активации нейрона, в результате получается его выходное значение $y_i^{(q)}$, i=0...M_i-1, где M_i – число нейронов в слое i; n=0...N-1, а N – число слоев в ИНС.

3. На основании полученных выходных значений нейронов по формуле (III.1.3.2) или (III.1.3.3) производится изменение весов синаптических связей.

4. Цикл с шага 2 повторяется, пока выходные значения ИНС не стабилизируются с заданной точностью. Применение этого нового способа определения завершения обучения, отличного от использовавшегося для ИНС обратного распространения, обусловлено тем, что подстраиваемые значения весов синаптических связей фактически не ограничены.

На втором шаге цикла попеременно предъявляются все образы из входного набора.

Полный алгоритм обучения имеет примерно такую же структуру, как в методах Хебба, но на шаге 3 из всего слоя выбирается нейрон, значения весов синаптических связей которого максимально походят на входной образ, и подстройка весов по формуле (III.1.3.4) проводится только для него. Эта, так называемая, аккредитация может сопровождаться затормаживанием всех остальных нейронов слоя и введением выбранного нейрона в насыщение. Выбор такого нейрона может осуществляться, например, расчетом скалярного произведения вектора весов синаптических связей с вектором входных значений. Максимальное произведение дает выигравший нейрон.

В этом случае "побеждает" нейрон с наименьшим расстоянием. Иногда слишком часто получающие аккредитацию нейроны принудительно исключаются из рассмотрения, чтобы "уравнять права" всех нейронов слоя. Простейший вариант такого алгоритма заключается в торможении только что выигравшего нейрона.

НИУ... КТ

При использовании алгоритма Кохонена существует практика нормализации входных образов, а также на стадии инициализации и нормализации начальных значений весов синаптических связей:

$$x_i = \frac{x_i}{\sqrt{\sum_{j=0}^{n-1} x_j^2}}, \quad (III.1.3.5)$$

где x_i – i-ая компонента вектора входного образа или вектора весов синаптических связей, а n – его размерность. Это позволяет сократить длительность процесса обучения.

Инициализация весов синаптических связей случайными значениями может привести к тому, что различные классы, которым соответствуют плотно распределенные входные образы, сольются или, наоборот, раздробятся на дополнительные подклассы в случае близких образов одного и того же класса. Во избежание такой ситуации используется метод выпуклой комбинации. Суть его сводится к тому, что входные нормализованные образы подвергаются преобразованию:

$$x_i = \alpha(t) \cdot x_i + (1 - \eta(t)) \cdot \frac{x_i}{\sqrt{n}}, \quad (III.1.3.6)$$

где x_i – i-ая компонента входного образа, n – общее число его компонент, $\eta(t)$ – коэффициент обучения, изменяющийся в процессе обучения от нуля до единицы, в результате чего вначале на входы ИНС подаются одинаковые образы, а с течением времени они все больше сходятся к исходным. Значения весов синаптических связей устанавливаются на шаге инициализации равными величине

$$w_0 = \frac{1}{\sqrt{n}}, \quad (III.1.3.7)$$

где n – размерность вектора весов инициализируемого слоя.

На основе рассмотренного выше метода строятся нейронные сети особого типа, называемые самоорганизующиеся структуры

НИУ ... КТ

(self-organizing feature maps). Для них после выбора из слоя q нейрона j с минимальным расстоянием D_j(III.1.3.1) обучается по формуле (III.1.3.2), (III.1.3.3) не только этот нейрон, но и его соседи, расположенные в окрестности R. Величина R на первых итерациях очень большая, так что обучаются все нейроны, но с течением времени она уменьшается до нуля. Это означает, что чем ближе конец обучения, тем точнее определяется группа нейронов, отвечающих каждому классу образов.

Задания III.1.3:

1. Напишите формулу минимального расстояния, используемого в методе обучения без учителя.
2. Напишите формулу подстройки весов нейронной сети методом Хебба.
3. Напишите формулу подстройки весов методом Кохонена.
4. Напишите формулу дифференциального метода обучения нейронных сетей Хебба.
5. Напишите формулу нормализации входных образов.
6. Напишите формулу нормализации начальных значений весов синаптических связей.
7. Напишите формулу для установления значения весов синаптических связей.

Вопросы III.1.3:

1. Когда используется обучение искусственных нейронных сетей без учителя?
2. Какая попытка предпринимается при обучении нейронных сетей без учителя?
3. Какие ИНС особого типа строятся на основе метода обучения без учителя?
4. Когда устанавливаются веса синаптических связей искусственных нейронных сетей?
5. К чему приведет инициализация весов синаптических связей случайными значениями?

НИИ искусственный интеллект

6. На основе чего строятся самоорганизующиеся структуры нейронных сетей?

7. На основе чего строятся нейронные сети особого типа, называемые самоорганизующимися структурами?

НИИ Искусственный интеллект

III.1.4. Парадигма смешанного обучения нейронной сети

В этом параграфе нами будут рассмотрены вопросы парадигмы смешанного обучения ИНС, предложены задания и сформулированы вопросы. При подготовке учебного материла были использованы литературные источники [14, 17, 22, 27, 28, 33].

Смешанное обучение, состоит из комбинации обучения с учителем и обучения без учителя, т.е. часть весов определяется посредством обучения с учителем, а другая часть получается с помощью алгоритмов самообучения.

Для демонстрации смешанного обучения рассмотрим полносвязную нейронную сеть Хопфилда (Рисунок III.1.4.1.).

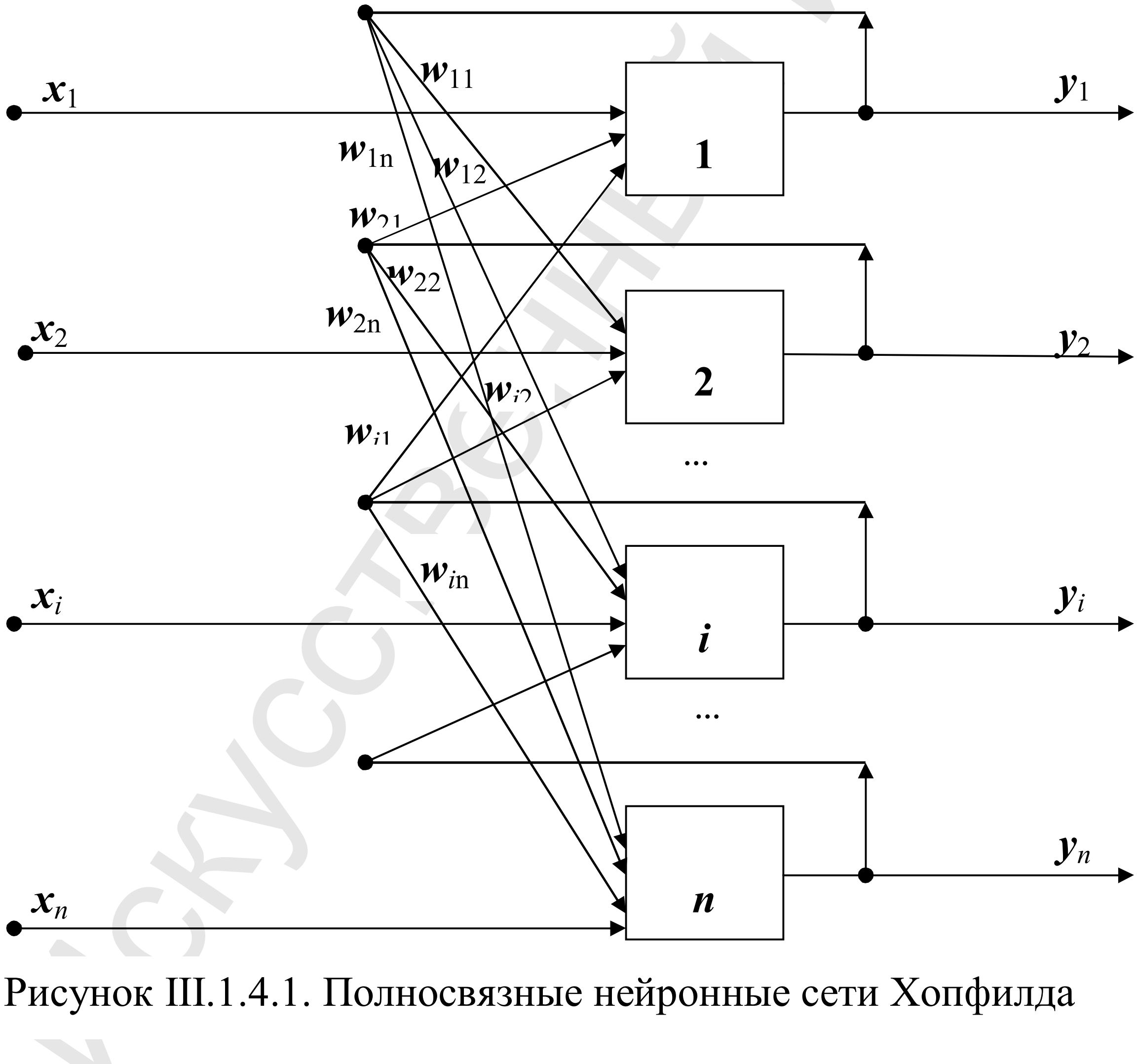


Рисунок III.1.4.1. Полносвязные нейронные сети Хопфилда

В соответсвии с рисунком входной вектор $X=(x_1,x_2,\ldots,x_n)$, n – число нейронов в ИНС и размерность входных и выходных векторов. Каждый элемент x_i равен +1, или –1. Вектор k-го

НИИ «Искусственный интеллект»

примера – X^k, а его компоненты – x_i^k, $k=1,...,m$, m – число примеров. Если образ распознан, то выход ИНС равен $Y = X^k$, где Y – вектор выходных значений ИНС: $Y=\{y_i: \ i=1,...,n\}$.

Инициализация нейронной сети Хопфилда.

Веса синаптических связей ИНС определяются как:

$$W_{ij} = \begin{cases} \sum_{k=1}^{m} x_i^k x_j^k, & i \neq j \\ 0, & i = j \end{cases},$$

где i и j – индексы соответственно предсинаптического и постсинаптического нейронов; x_i^k – i-ый и x_j^k – j-ый элементы вектора k-го примера.

Алгоритм функционирования нейронной сети Хопфилда.

Шаг 1. На входы ИНС подается неизвестный сигнал. Фактически его введение осуществляется непосредственно установкой значений аксонов: $y_i(0)=x_i$, $i=1,..., n$.

Шаг 2. Вычисляются новые состояния нейронов и значения аксонов:

$$s_j(p+1)=\sum_{i=1}^{n} w_{ij} y_i(p); \; y_j(p+1)=f(s_j(p+1))$$

где p – номер итерации, ***f*** – функция активации единичного прыжка

$$Y = \begin{cases} 0, & X < T \\ 1, & X \geq T \end{cases}.$$

Шаг 3. Производится проверка, изменились ли выходные значения аксонов за последнюю итерацию. Если да, то переход на шаг 2, в противоположном случае – конец.

Теперь рассмотрим нейронную сеть Хемминга, которая показана на рисунке III.1.4.2.

При обучении такая нейронная сеть должна выбрать пример с минимальным расстоянием Хэмминга к неизвестному входному сигналу. Напомним, что расстоянием Хэмминга называется число

НИИ искусственный интеллект

позиций, в которых соответствующие значения в двух бинарных векторах одинаковой размерности различны. В более общем случае расстояние Хэмминга применяется для векторов одинаковой размерности любых q-ичных алфавитов и служит метрикой различия (функцией, определяющей расстояние в метрическом пространстве) объектов одинаковой размерности.

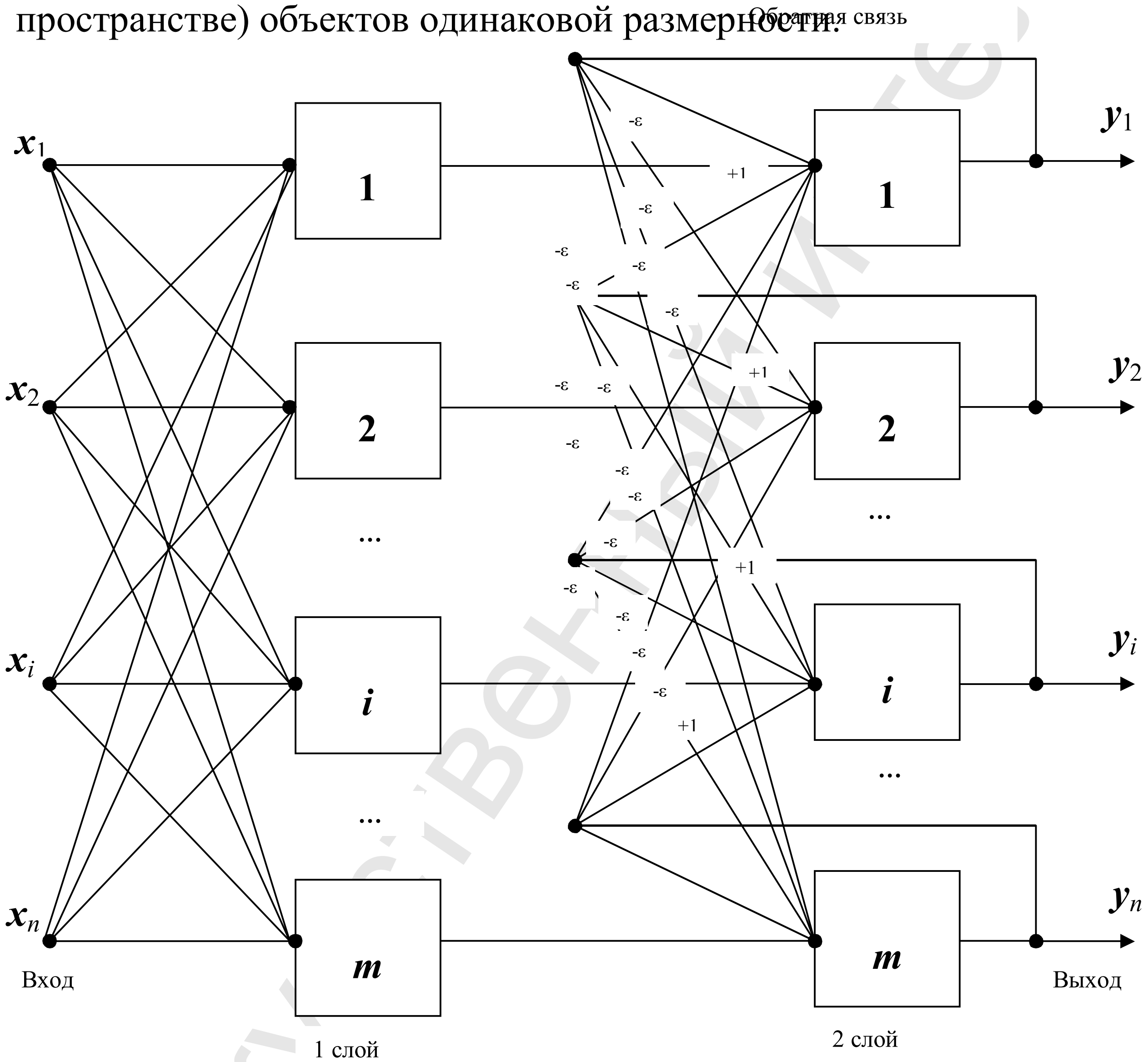


Рисунок III.1.4.2. Нейронная сеть Хемминга.

Инициализация нейронной сети Хэмминга.

Веса 1-го слоя ИНС и порог функции активации равны

$$W_{ik} = \frac{x_i^k}{2}, \qquad i = 1, ..., n-1, \qquad k = 1, ..., m-1$$

НИИ искусственный интеллект

$$T_k = \frac{n}{2}, \qquad k = 1,...,\ m-1,$$

где x_i^k – i-ый элемент k-го примера.

Тормозящие веса синаптических связей во 2-ом слое ε удовлетворяют условию 0<ε<1/m. Синапс нейрона, связанный с его аксоном, имеет вес +1.

Алгоритм функционирования нейронной сети Хэмминга.

1. На входы ИНС подается неизвестный вектор $X = (x_1, x_2, ..., x_{n-1})$, исходя из которого рассчитываются состояния нейронов первого слоя: $y_j^{(1)} = s_j^{(1)} = \sum_{i=0}^{n-1} w_{ij} x_i + T_j, \quad j = 0,1,...,m-1,$

где верхний индекс в скобках указывает номер слоя.

После этого полученными значениями инициализируются значения аксонов второго слоя: $y_j^{(2)} = y_j^{(1)}, \quad j = 0,1,...,m-1$.

2. Вычисление новых состояний нейронов второго слоя и значений их аксонов:

$$s_j^{(2)}(p+1) = y_j^{(1)}(p) - \varepsilon \sum_{k=0}^{m-1} y_k^{(2)}(p), \quad k \neq j, \qquad j = 0,1,...,m-1;$$

$$y_j^{(2)}(p+1) = f(s_j^{(2)}(p+1)), \quad j = 0,1,...,m-1.$$

Функция активации f имеет вид порога, причем её значение должно быть достаточно большим, чтобы любые возможные значения аргумента не приводили к насыщению.

3. Проверить, изменились ли выходы нейронов второго слоя за последнюю итерацию. Если да, топерейти к шагу 2,иначе – конец.

Задания III.1.4:

1. Напишите суть смешанного обучения ИНС.
2. Опишите алгоритм функционирования сети Хопфилда.
3. Опишите алгоритм функционирования сети Хэмминга.

Вопросы III.1.4:

1. Как организуется смешанное обучение нейронных сетей?

НИИ искусственный интеллект

2. Как инициализируется сеть Хопфилда?
3. Как инициализируется сеть Хэмминга?

НИИ Искусственный интеллект

III.2. Методы обучения нейронных сетей

III.2.1. Обучение нейронной сети методом обратного распространения ошибки

В этом параграфе нами будут рассмотрены вопросы обучения ИНС методом обратного распространения ошибки, предложены задания и сформулированы вопросы. При подготовке учебного материла были использованы литературные источники [14, 17, 22, 27, 28,3 3].

Методы обучения ИНС определяют три фундаментальных свойства, связанных с обучением: *емкость, сложность образов* и *вычислительная сложность*.

Метод обратного распространения ошибки (*Error back propagation*) разработан Румельхартом в 1986 году и предназначен для обучения с учителем многослойной ИНС с прямыми связями.

Идея метода состоит в:

1) определении множества обучающих пар, состоящих из векторов значений входа и желаемого выхода;

2) вычислении ошибки, равной величине разности между значением желаемого выхода и значением реального выхода;

3) распространении этой ошибки в обратном направлении – навстречу потоку данных и перераспределяется путем изменения весов синаптических связей между нейронами так, чтобы минимизировать суммарную ошибку (см. рисунок III.2.1.1).

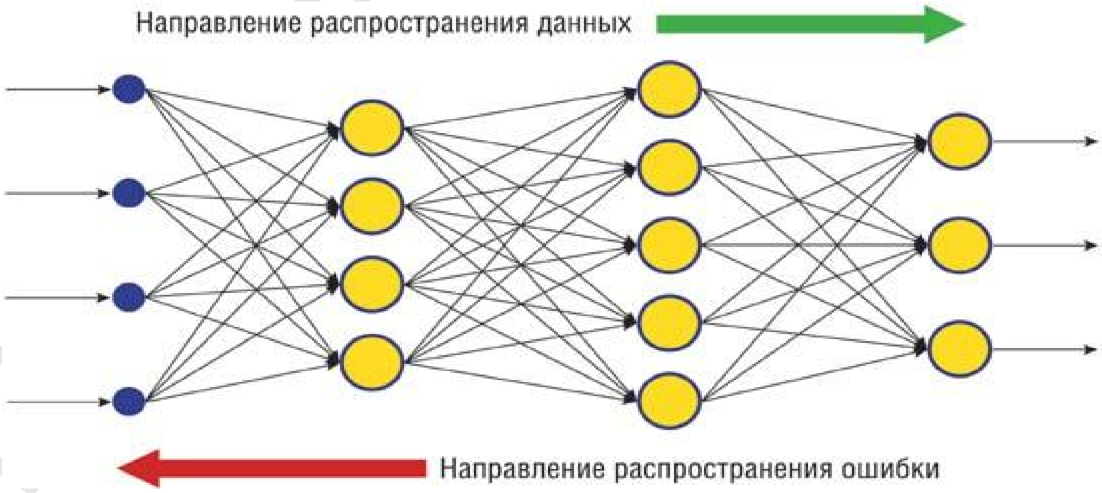


Рисунок III.2.1.1. Направление распространения ошибки.

Метод обратного распространения ошибки также известен как метод обобщения дельта-правил.

Таким образом, распространение ошибки представляет собой итеративный алгоритм обучения:

1. Выбрать случайным образом очередную обучающую пару (x_i, y_j), x_i – компоненты входного вектора, y_j – компоненты выходного вектора, $i=1,2,\ldots,K$; $j=1,2,\ldots,N$; η – коэффициент скорости обучения, $0<\eta<1$.

2. Вычислить выход ИНС в режиме обычного функционирования ИНС;

3. Вычислить разность между реальным выходом $y_{ij}^{(Q)}$ и требуемым выходом (целевым вектором) $d_{ij}^{(Q)}$ ИНС и расчитать дельту ошибки для выходного слоя по формуле:

$$\delta_{ij}^{(Q)}=(y_{ij}^{(Q)}-d_{ij}^{(Q)})\cdot\frac{dy_{ij}}{ds_{ij}},$$

здесь $\frac{dy}{ds}=1-s$ в случае гиперболического тангенса и $\frac{dy}{ds}=s(1-s)$ в случае сигмоиды.

4. Вычислить изменение весов синаптических связей для выходного слоя Q ИНС, чтобы минимизировать ошибку:

$$\Delta w_{ij}^{(Q)}=-\eta\cdot\delta_{ij}^{(Q)}\cdot y_{ij}^{(Q-1)}.$$

5. Рассчитать ошибки для всех остальных слоев по формуле:

$$\delta_j^{(q)}=\left[\sum_k\delta_k^{(q+1)}\cdot w_{jk}^{(q+1)}\right]\cdot\frac{dy_j^{(q)}}{ds_j^{(q)}}.$$

6. Скорректировать все веса синаптических связей ИНС:

$$w_{ij}^{(q)}(t)=w_{ij}^{(q)}(t-1)+\Delta w_{ij}^{(q)}(t),$$

$$w_{ij}^{(q)}(t)=-\eta\cdot(\mu\cdot\Delta w_{ij}^{(q)}(t-1)+(1-\mu)\cdot\delta_j^{(q)}\cdot\delta_i^{(q-1)}),$$

где μ – коэффициент инерционности. Эта формула «гладко» изменяет веса, что приводит к понижению перепадов ошибки.

НИИ Искусственный интеллект

Последнюю формулу целесообразно применять в конце обучения, когда ошибка близка к заданной.

7. Повторять шаги с 1 по 6 для каждого вектора обучающего множества до тех пор, пока ошибка на всем множестве не достигнет приемлемого уровня.

8. Операции, выполняемые шагами 1, 2 и 3, сходны с теми, которые выполняются при функционировании уже обученной ИНС, т. е. подается входной вектор и вычисляется результирующий выход. Вычисления выполняются послойно. В качестве критерия ошибки (целевой функции) используется средняя квадратичная ошибка, в которой суммирование ведется по всем нейронам выходного слоя и по всем образам, обрабатываемым ИНС:

$$E(w) = \frac{1}{2} \cdot \sum_{j=1}^{N} \sum_{i=1}^{K} \left(y_{ij}^{(q)} - d_{ij}^{(q)} \right)^2, \text{ (III.2.1.1)}$$

где $y_{ij}^{(q)}$ – реальный выход q-го выходного слоя ИНС для i-го нейрона на j-м обучающем примере (образа), $d_{ij}^{(q)}$ – желаемый выход этого нейрона, M – количество обучающих примеров, K – число входов, N – число выходов, q=1,2, ..., Q – число слоев ИНС.

9. Минимизировав такой функционал, мы получим решение по методу наименьших квадратов.

В итоге каждый нейрон способен определить вклад каждого своего веса в суммарную ошибку сети.

Простейшее правило обучения соответствует методу наискорейшего спуска, в котором изменение весов синаптических связей пропорционально их вкладу в общую ошибку, то есть вычисляется чувствительность ошибки ИНС к изменению весов синаптических связей. Для этого нужно вычислить частные производные от ошибки по весам и для нахождения минимума использовать метод градиентного спуска, который обеспечивает на каждом шаге обучения изменение этих весов по формуле:

$$\Delta w_{ij}^{(q)} = -\eta \cdot \frac{\partial E}{\partial w_{ij}^{(q)}}, \text{ (III.2.1.2)}$$

НИУ Искусственный интеллект

где $\Delta w_{ij}^{(q)}$ – коррекция весов синаптических связей, $w_{ij}^{(q)}$ – вес синапса *j*-го нейрона *q*-го слоя с *i*-м нейроном (*q-1*)-го слоя, η – коэффициент скорости обучения, $0<\eta<1$; $q=Q-1,Q-2,\ldots,2$.

Таким образом, требуется определить частные производные целевой функции *E* по всем синаптическим весам ИНС. Согласно правилам дифференцирования сложной функции представляется:

$$\frac{\partial E}{\partial w_{ij}^{(q)}} = \frac{\partial E}{\partial y_j^{(q)}} \cdot \frac{dy_j^{(q)}}{ds_j^{(q)}} \cdot \frac{\partial s_j^{(q)}}{\partial w_{ij}^{(q)}}, \qquad (\text{III.2.1.3})$$

где $\boldsymbol{y_j^{(q)}}$ – выход, $\boldsymbol{s_j^{(q)}}$ – взвешенная сума входов *j*-го нейрона *q*-го слоя ИНС.

Поскольку производная функции активации должна быть определена на всей оси абсцисс, то функция единичного скачка и прочие функции активации с неоднородностями не подходят для рассматриваемых ИНС. В них применяются такие гладкие функции, как гиперболический тангенс или классический сигмоид.

Зная функцию активации можно вычислить ее производную $\frac{dy_j^{(q)}}{ds_{ij}^{(q)}}$, являющуюся вторым множителем выражения (III.2.1.3). Например, если функцией активации будет:

1) классический сигмоид – логистическая функция $f(s)=\frac{1}{1+e^{-as}}$, то ее производная равна $f'(s)=a\cdot f(s)\cdot(1-f(s))$;

2) гиперболический тангенс $f(s)=\frac{e^{ks}-e^{-ks}}{e^{ks}+e^{-ks}}$, то ее производная будет равна $f'(s)=1-s^2$.

Значение второго множителя выражения (III.2.1.3) для логистической функции вычисляется по формуле:

$$\frac{dy_j^{(q)}}{ds_j^{(q)}} = a \cdot y_j^{(q)} \cdot (1 - y_j^{(q)}) \qquad (\text{III.2.1.4})$$

Третий множитель выражения (III.2.1.3) есть ни что иное, как выход i-го нейрона (q-1)-го слоя, т.е.

$$\frac{\partial s_j^{(q)}}{\partial w_{ij}^{(q)}} = y_i^{(q-1)} \quad \text{(III.2.1.5)}$$

Теперь можно легко вычислить частные производные целевой функции по весам синаптических связей нейронов выходного слоя. Для первого множителя выражения (III.2.1.3) производя дифференцирование функции ошибки по $y_j^{(Q)}$ получим

$$\frac{\partial E}{\partial y_j^{(Q)}} = (y_j^{(Q)} - d_j^{(Q)}) \quad \text{(III.2.1.6)}$$

Далее учитывая (III.2.1.5) и (III.2.1.6) на основание выражения (III.2.1.3) можно получить:

$$\frac{\partial E}{\partial w_{ij}^{(Q)}} = (d_j^{(Q)} - y_j^{(Q)}) \cdot \frac{dy_j^{(Q)}}{ds_j^{(Q)}} \cdot y_i^{(Q-1)} \quad \text{(III.2.1.7)}$$

Для весов синаптических связей нейронов внутренних слоев мы не можем сразу записать, чему равен первый множитель из выражения (III.2.1.3), однако его можно раскладывать следующим образом:

$$\frac{\partial E}{\partial y_j^{(q)}} = \sum_k \frac{\partial E}{\partial y_k^{(q+1)}} \cdot \frac{dy_k^{(q+1)}}{ds_k^{(q+1)}} \cdot \frac{\partial s_k^{(q+1)}}{\partial y_j^{(q)}} = \sum_k \frac{\partial E}{\partial y_k^{(q+1)}} \cdot \frac{dy_k^{(q+1)}}{ds_k^{(q+1)}} \cdot w_{jk}^{(q+1)} \quad \text{(III.2.1.8)}$$

Здесь суммирование по k выполняется среди нейронов слоя (q+1).
Введем новую переменную

$$\delta_j^{(q)} = \frac{\partial E}{\partial y_j^{(q)}} \cdot \frac{dy_j^{(q)}}{ds_j^{(q)}} \quad \text{(III.2.1.9)}$$

Тогда для нейронов выходного слоя учитывая (III.2.1.6) и (III.2.1.9) получим

$$\delta_j^{(Q)} = (y_j^{(Q)} - d_j^{(Q)}) \cdot \frac{dy_j^{(q)}}{ds_j^{(q)}} \quad \text{(III.2.1.10)}$$

НИИ искусственный интеллект

Заметим, что в (III.2.1.8) первые два множителя есть не что иное, как $\delta_k^{(q+1)}$. Таким образом, с помощью (III.2.1.8) можно выразить величины $\delta_k^{(q)}$ для нейронов q-го слоя через $\delta_k^{(q+1)}$ для нейронов (q+1)-го слоя. Поскольку для последнего слоя $\delta_j^{(Q)}$ легко вычисляется по (III.2.1.10), то значение $\delta_j^{(q)}$ для всех нейронов всех слоев можно получить с помощью рекурсивной формулы:

$$\delta_j^{(q)} = \left[\sum_k \delta_k^{(q+1)} \cdot w_{jk}^{(q+1)}\right] \cdot \frac{dy_j^{(q)}}{ds_j^{(q)}} \quad \text{(III.2.1.11)}$$

Теперь формулу (III.2.1.2) для модификации весов синаптических связей учитывая формулы (III.2.1.5) и (III.2.1.9) можно окончательно записать в виде:

$$\Delta w_{ij}^{(q)} = -\eta \cdot \delta_j^{(q)} \cdot y_i^{(q-1)} \quad \text{(III.2.1.12)}$$

Формула (III.2.1.12) задает неэффективное вычисление в случае, когда производные по различным весам сильно отличаются. Это соответствует ситуации, когда значение функций *s* для некоторых нейронов близки по модулю к 1 или, когда модуль некоторых весов много больше 1. В этом случае для придания процессу коррекции весов некоторой инерционности, сглаживающей резкие скачки при перемещении по поверхности целевой функции, в формулу (III.2.1.12) вводится некоторый коэффициент инерционности λ, позволяющий корректировать прираращение веса на предыдущей итерации и преодолевать локальные минимумы. Имеется два варианта модификации формулы (III.2.1.12):

1) $\Delta w_{ij}^{(q)}(t) = -\eta \cdot \delta_j^{(q)} \cdot y_i^{(q-1)} + \lambda \cdot \Delta w_{ij}^{(q)}(t-1)$, (III.2.1.13)

2) $\Delta w_{ij}^{(q)}(t) = -\eta \cdot [(1-\lambda) \cdot \delta_j^{(q)} \cdot y_i^{(q-1)} + \lambda \cdot \Delta w_{ij}^{(q)}(t-1)]$, (III.2.1.14)

где λ – коэффициент инерционности, t – номер текущей итерации, η – коэффициент скорости обучения.

НИИ Искусственный интеллект

Таким образом, полный алгоритм обучения ИНС с помощью алгоритма обратного распространения ошибки строиться следующими шагами:

Шаг 1. Присваиваем всем весам синаптических связей ИНС случайные начальные значения. При этом ИНС осуществляет какое-то случайное преобразование входных сигналов и значения функции ошибки (III.2.1.1) могут быть велики.

Шаг 2. Подаем на вход ИНС один из входных векторов из обучающего множества и вычисляем выходные значения ИНС, запоминая при этом выходные значения каждого из нейронов.

$$s_j^{(q)} = \sum_{i=1}^{I_j^{(q)}} w_{ij}^{(q)} \cdot y_i^{(q-1)}; \quad y_j^{(q)} = f(s_j^{(q)}); \; y_j^{(0)} = x_r \, ,$$

где $I_j^{(q)}$ – число входов нейрона j слоя q; j =1, 2, …, J_q; J_q – число нейронов в слое (*q-1*) с учетом нейрона с постоянным выходным состоянием +1, задающего смещения; q = *Q, Q*-1,…,2,1; Q – число слоев в ИНС; $w_{ij}^{(q)} \cdot y_i^{(q-1)}$ – i-й вход нейрона j, слоя q; f – сигмоидальная функция активации; x_r – r-я компонента входного вектора.

Шаг 3. Вычисляем сначала для выходного слоя Q значение $\delta_j^{(Q)}$ по формуле (III.2.1.10) и изменение весов синаптических связей $\Delta w^{(Q)}$ с помощью (III.2.1.12), (III.2.1.13) или (III.2.1.14), затем для всех остальных слоев q значения $\delta_j^{(q)}$ с помощью рекуррентной формулы (III.2.1.11) и изменения весов $\Delta w^{(q)}$ с помощью (III.2.1.12), (III.2.1.13) или (III.2.1.14).

Шаг 4. Скорректируем все веса синаптических связей ИНС для текущей итерации t по следующей формуле $\Delta w_{ij}^{(q)}(t) = w_{ij}^{(q)}(t-1) + \Delta w_{ij}^{(q)}(t)$.

Шаг 5. Вычисляем значение ошибки – целевой функции (III.2.1.1): если она существенна, то возвращаемся на шаг 2, иначе считаем ИНС успешно обучившейся и завершаем работу.

НИ...лект

ИНС на шаге 1 попеременно в случайном порядке предъявляются все обучающие примеры, чтобы ИНС не забывала одни по мере запоминания других.

Из выражения $\Delta w_{ij}^{(q)} = -\eta \cdot \delta_j^{(q)} \cdot y_i^{(q-1)}$ следует, что когда выходное значение $y_i^{(q-1)}$ стремится к нулю, эффективность обучения заметно снижается. При двоичных входных векторах в среднем половина весов синаптических связей не будет корректироваться, поэтому область возможных значений выходов нейронов (0,1) нужно сдвинуть в пределы (0.5, 0.5), что достигается простыми модификациями логистических функций.

Например, классический сигмоид $f(s) = \dfrac{1}{1+e^{-as}}$ преобразуется к виду: $f(s) = -0.5 + \dfrac{1}{1+e^{-as}}$.

Схема обратного распространения ошибки представлена на рисунке III.2.1.

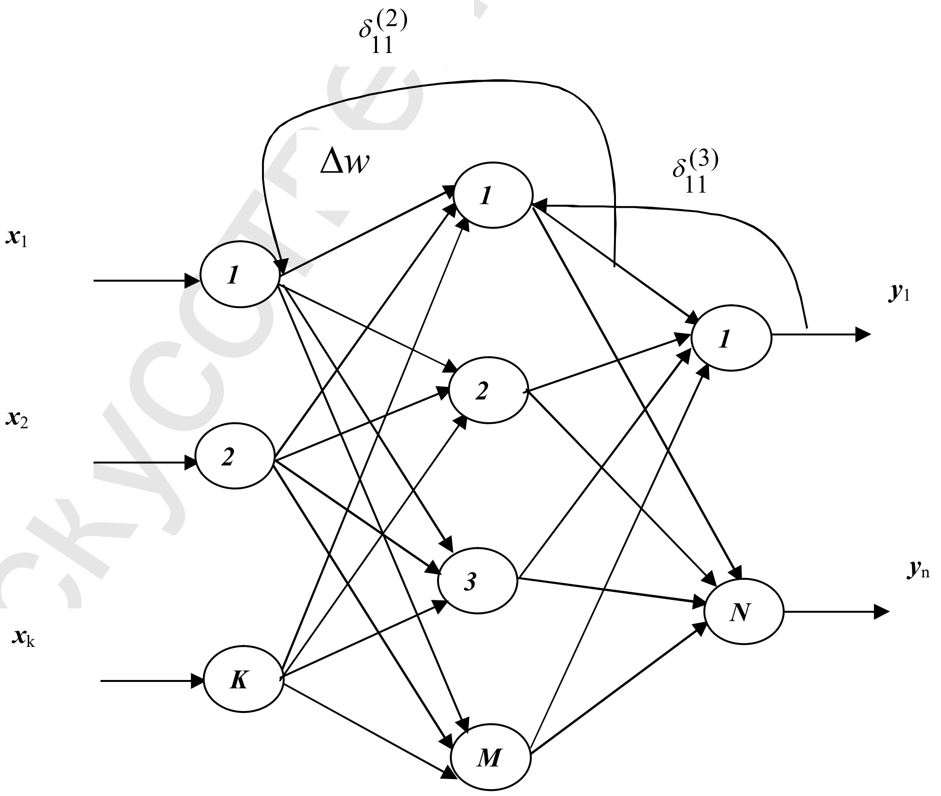


Рисунок III.2.1. Схема обратного распространения ошибки

НИИ Искусственный интеллект

Несмотря на многочисленные успешные применения метода обратного распространения ошибки, он очень долго обучает ИНС. Иногда ИНС может не обучиться. Длительное время обучения может быть результатом неоптимального выбора значения шага. Неудачи в обучении обычно возникают по следующим причинам: *паралича ИНС, попадания в локальный минимум и переобучение*.

Задания III.2.1:

1. Опишите алгоритм обучения нейронной сети методом обратного распространения.
2. Напишите формулу вычисления дельта ошибки для выходного слоя.
3. Напишите формулу изменение весов синаптических связей для выходного слоя.
4. Напишите формулу вычисления ошибки для остальных слоев.
5. Напишите формулу корректировки всех весов.
6. Напишите формулу вычисления средне-квадратичной ошибки.
7. Опишите полный алгоритм обучения ИНС с помощью алгоритма обратного распространения ошибки.

Вопросы III.2.1:

1. В чем заключается идея метода обратного распространения ошибки?
2. Что должно быть задано перед началом работы алгоритма обратного распространения ошибки?
3. Что используется в качестве критерия ошибки (целевой функции)?
4. Каким методом минимизируется функционал средней квадратичной ошибки?
5. Зачем нужен метод наискорейшего спуска для обучения нейронной сети методом обратного распространения ошибок?
6. Что обеспечивает метод градиентного спуска на каждом шаге обучения?

НИИ Искусственный интеллект

7. В каких случаях возникает неудача в обучении ИНС методом обратного распространения ошибки?

НИИ Искусственный интеллект

III.2.2. Обучение нейронной сети методом Больцмана

В этом параграфе нами будут рассмотрены вопросы обучения ИНС методом Больцмана, предложены задания и сформулированы вопросы. При подготовке учебного материла были использованы литературные источники [14, 17, 22, 27, 28, 33].

Метод Больцмана представляет собой стохастическое правило обучения, которое следует из информационных, теоретических и термодинамических принципов, и может рассматриваться как специальный случай метода обратного распространения ошибки, в котором под ошибкой понимается расхождение корреляций состояний в двух режимах.

Цель метода Больцмана является такая настройка весов, при которой состояния видимых нейронов удовлетворяют желаемому распределению вероятностей. Метод Больцмана реализуется следующим алгоритмом:

1. Определить переменную T, представляющую искусственную температуру и придать T большое начальное значение.
2. Предъявить ИНС множество входов с синаптическими весами и вычислить выходы и целевую функцию.
3. Дать случайное изменение значениям весов синаптических связей и пересчитать выход ИНС и изменение целевой функции в соответствии со сделанным изменением значений этих весов.
4. Если целевая функция уменьшилась (улучшилась), то сохранить изменение значений весов, иначе, если целевая функция увеличилась, то вероятность сохранения этого изменения вычисляется с помощью распределения Больцмана:

$$P(s) = e^{-\frac{s}{kT}},$$

где $P(s)$ – вероятность изменения s в целевой функции, k – константа, аналогичная константе Больцмана, выбираемая в зависимости от задачи, T – искусственная температура.

Выбирается случайное число r из равномерного распределения от 0 до 1. Если $P(s)$ больше, чем r, то изменение сохраняется, в противном случае величина веса возвращается к предыдущему значению. Это позволяет системе делать случайный шаг в направлении, ухудшающем целевую функцию, и дает ей тем самым возможность вырываться из локальных минимумов, где любой малый шаг увеличивает целевую функцию.

Для завершения больцмановского обучения повторяют шаги 3 и 4 для каждого из весов ИНС, постепенно уменьшая температуру T, пока не будет достигнуто допустимо низкое значение целевой функции. В этот момент предъявляется другой входной вектор, и процесс обучения повторяется. ИНС обучается на всех векторах обучающего множества, с возможным повторением, пока целевая функция не станет допустимой для всех них.

Величина случайного изменения веса на шаге 3 может определяться различными способами. Например, подобно тепловой системе, изменение веса w может вычисляться в соответствии с гауссовским распределением:

$$P(w) = e^{-\frac{w^2}{kT^2}},$$

где $P(w)$ – вероятность изменения веса на величину w, k – константа, T – искусственная температура.

Так как нужна величина изменения веса Δw, а не вероятность изменения веса, имеющего величину w, то метод Монте-Карло может быть использован следующим образом:

1. Найти кумулятивную вероятность, соответствующую $P(w)$. Это есть интеграл от $P(w)$ *в* пределах от 0 до w. Так как в данном случае $P(w)$ не может быть проинтегрирована аналитически, она должна интегрироваться численно, а результат необходимо затабулировать.

2. Выбрать случайное число из равномерного распределения на интервале (0,1). Используя эту величину в качестве значения $P(w)$, найти в таблице соответствующее значение для величины изменения веса синаптических связей.

НИИ Искусственного Интеллекта / watermark

Свойства машины Больцмана изучались во многих работах, где было показано, что скорость уменьшения температуры должна быть обратно пропорциональна логарифму времени, чтобы была достигнута сходимость к глобальному минимуму. Скорость охлаждения в такой системе выражается следующим образом:

$$T(t) = \frac{T_0}{\log(1+t)},$$

где $T(t)$ – искусственная температура как функция времени, T_0 – начальная искусственная температура, t – искусственное время.

Этот результат предсказывает очень медленную скорость охлаждения (вычисления). Поэтому машины Больцмана требуют для обучения очень большого ресурса времени.

Задания III.2.2:

1. Опишите алгоритм реализации метода Больцмана.
2. Напишите формулу для распределения Больцмана.
3. Напишите формулу для распределения Гаусса.
4. Объясните причину востребования машиной Больцмана больших временных ресурсов.
5. Укажите причину завершения больцмановского обучения.
6. Укажите условия обучения ИНС на всех векторах обучающего множества.
7. Укажите условия выхода из локального минимума.

Вопросы III.2.2:

1. Что представляет собой метод Больцмана?
2. Как может рассматриваться метод Больцмана?
3. В чем заключается цель метода Больцмана?
4. Что позволяет вычислить распределение Больцмана?
5. Для чего можно использовать распределение Гаусса?
6. Почему и как можно использовать метод Монте-Карло?
7. Чем достигается возможность вырваться из локальных минимумов?

НИИ искусственный интеллект

III.2.3. Обучение нейронной сети методом Коши

В этом параграфе нами будут рассмотрены вопросы обучения ИНС методом Коши, предложены задания и сформулированы вопросы. При подготовке учебного материла были использованы литературные источники [14, 17, 22, 27, 28, 33].

Если в методе обучения нейронной сети, приведенном в III.2.2, при вычислении величины шага вместо распределения Больцмана использовать распределение Коши, то можно увеличить шаг и ускорить обучение. В действительности распределение Коши имеет бесконечную (неопределенную) дисперсию. С помощью такого простого изменения максимальная скорость уменьшения температуры становится обратно пропорциональной линейной величине, а не логарифму, как это было в алгоритме обучения нейронной сети по методу Больцмана, т.е. эту зависимость можно выразить так:

$$T(t) = \frac{T_0}{1+t} \qquad \text{(III.2.3.1)}$$

Распределение Коши имеет вид

$$P(x) = \frac{T(t)}{T(t)^2 + x^2} \qquad \text{(III.2.3.2)}$$

где $P(x)$ есть вероятность шага величины x.

Такая замена позволит резко уменьшить время обучения и по сравнению с распределением Больцмана распределение Коши, как это показано на рисунке III.2.3.1, имеет более длинные «хвосты», увеличивая тем самым вероятность больших шагов.

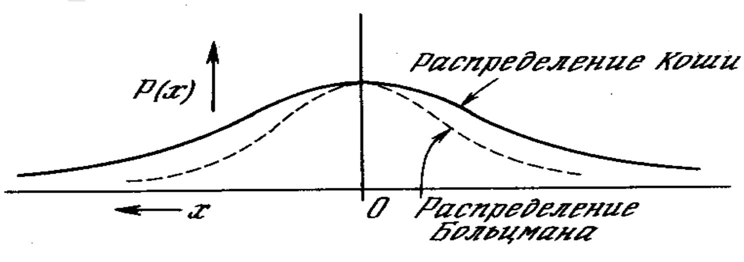


Рисунок III.2.3.1 Распределение Коши и распределение Больцмана

В уравнении (III.2.3.2) функция распределение $P(x)$ может быть проинтегрирована, затем решая относительно x, получаем

$$x_c = \rho T(t) tg((P(x)) \quad , \qquad \text{(III.2.3.3)}$$

где ρ – коэффициент скорости обучения; x_c – изменение веса.

Теперь применение метода Монте Карло становится очень простым. Для нахождения *x* в этом случае выбирается случайное число из равномерного распределения на открытом интервале (–π/2, π/2) – ограничение для функции тангенса. Оно подставляется в формулу (III.2.3.3) в качестве *P*(*x*), и с помощью текущей температуры вычисляется величина шага.

Несмотря на улучшение, достигаемое с помощью метода Коши, время обучения может оказаться все еще слишком большим. Для ускорения этого процесса можно использовать *метод искусственной теплоемкости* из термодинамики, в котором скорость уменьшения температуры изменяется в соответствии с искусственной «теплоемкостью», вычисляемой в процессе обучения.

Во время отжига металла происходят фазовые переходы, связанные с дискретными изменениями уровней энергии. При каждом фазовом переходе может иметь место резкое изменение величины, называемой теплоемкостью. *Теплоемкость* определяется как скорость изменения температуры с энергией. Изменения теплоемкости происходят из-за попадания системы в локальные энергетические минимумы.

ИНС проходят аналогичные фазы в процессе обучения. На границе фазового перехода искусственная теплоемкость может скачкообразно измениться. Эта псевдо теплоёмкость определяется как средняя скорость изменения температуры с целевой функцией. В примере шарика в коробке сильная начальная встряска делает среднюю величину целевой функции, фактически не зависящей от малых изменений температуры, т. е. теплоемкость близка к константе. Аналогично при очень низких температурах система замерзает в точке минимума, так что теплоемкость снова близка к константе. Ясно, что в каждой из этих областей допустимы сильные

НИИ искусственный интеллект

изменения температуры, так как не происходит улучшения целевой функции.

При критических температурах небольшое уменьшение температуры приводит к большому изменению средней величины целевой функции. Возвращаясь к аналогии с шариком, при «температуре», когда шарик обладает достаточной средней энергией, чтобы перейти из A в B, но недостаточной для перехода из B в A, средняя величина целевой функции испытывает скачкообразное изменение. В этих критических точках алгоритм должен изменять температуру очень медленно, чтобы гарантировать, что система не замерзнет случайно в точке A, оказавшись пойманной в локальный минимум. Критическая температура может быть обнаружена по резкому уменьшению искусственной теплоемкости, т. е. средней скорости изменения температуры с целевой функцией. При достижении критической температуры скорость изменения температуры должна замедляться, чтобы гарантировать сходимость к глобальному минимуму. При всех остальных температурах может без риска использоваться более высокая скорость снижения температуры, что приводит к значительному снижению времени обучения.

Задания III.2.3:

1. Напишите формулу для скорости уменьшения температуры в методе Коши.
2. Напишите формулу для распределения Коши, которое позволяет резко уменьшить время обучения.
3. Укажите место, где искусственная теплоемкость может скачкообразно измениться.

Вопросы III.2.3:

1. В чем заключается основная идея обучения нейронной сети методом Коши?
2. С помощью каких средств можно улучшить обучение нейронных сетей методом Коши?

НИИ искусственный интеллект

3. Когда должна замедляться скорость изменения температуры, чтобы гарантировать сходимость к глобальному минимуму?

НИИ Искусственный интеллект

III.2.4. Обучение нейронной сети методом Хебба

В этом параграфе нами будут рассмотрены вопросы обучения ИНС методом самоорганизации Хебба, предложены задания и сформулированы вопросы. При подготовке учебного материла были использованы литературные источники [14, 17, 22, 27, 28, 33].

В методе обучения ИНС без учителя процесс подстраивания весов синаптических связей происходит самостоятельно. При этом ИНС может самоорганизовываться, изменяя свою структуру. Подстройка весов синаптических связей может проводиться только на основании информации, доступной в нейроне, то есть его состояния и уже имеющихся весов. Исходя из этого соображения по аналогии с принципами самоорганизации нервных клеток, построен метод Хебба.

Метод Хебба является самым старым правилом обучения ИНС без учителя, которое опирается на следующие нейрофизиологические наблюдения. Если нейроны с обеих сторон синапса активизируются одновременно и регулярно, то вес синапса возрастает и зависит только от активных нейронов, которые связаны данным синапсом. При этом изменение веса каждой межнейронной связи зависит только от активности нейронов, образующих этот синапс, что сильно упрощает реализацию алгоритмов обучения.

При обучении ИНС методом Хебба наращивание весов определяется произведением уровней возбуждения, передающего и принимающего нейронов, записанного следующим образом:

$$w_{ij}^{(q)}(t) = w_{ij}^{(q)}(t-1) + \eta \cdot y_i^{(q-1)} \cdot y_j^{(q)}, \qquad \text{(III.2.4.1)}$$

где $y_i^{(q-1)}$ – выход нейрона i слоя (q-*1*) и вход нейрона j, $y_j^{(q)}$ – выход нейрона j слоя q, $w_{ij}^{(q)}(t)$ и $w_{ij}^{(q)}(t-1)$ – вес синапса от нейрона i к нейрону j слоя q на итерациях t и t-1 соответственно, η– коэффициент скорости обучения, $q = Q, Q\text{-}1, \ldots, 2, 1$.

НИИ Искусственный интеллект

При обучении по методу Хебба усиливаются связи между возбужденными нейронами в ИНС, что видно из формулы (III.2.4.1).

Существует также и дифференциальный метод обучения Хебба, определяемый соотношением:

$$w_{ij}^{(q)}(t) = w_{ij}^{(q)}(t-1) + \eta \cdot \left[y_i^{(q-1)}(t) - y_i^{(q-1)}(t-1)\right] \cdot \left[(y_j^{(q)}(t) - y_j^{(q)}(t)\right], \qquad \text{(III.2.4.2)}$$

где $y_i^{(q-1)}(t)$ и $y_i^{(q-1)}(t-1)$ – выход нейрона i слоя *q-1* на итерациях t и t-1 соответственно, $y_j^{(q)}(t)$ и $y_j^{(q)}(t-1)$ – выход нейрона j слоя q на итерациях t и t-1, η – коэффициент скорости обучения.

В дифференциальном методе обучения Хебба, как видно из формулы (III.2.4.2), сильнее всего обучаются синапсы, соединяющие те нейроны, выходы которых наиболее динамично изменились в сторону увеличения.

Полный алгоритм обучения ИНС с помощью правил Хебба с применением формул (III.2.4.1) и (III.2.4.2) будет иметь вид:

Шаг 1. Инициализация: Всем синаптическим весам ИНС присваиваются небольшие случайные начальные значения.

Шаг 2. Вычисление взвешенной суммы: На вход ИНС подается входной образ и веса распространяются по всем слоям согласно принципа прямого распространения и для каждого нейрона вычисляются значения выхода, к которой применяется функция:

$$\sum_{i=1}^{I_j^{(q)}} w_{ij}^{(q)} \cdot x_i^{(q)} = s_j^{(q)}; \quad f(s_j^{(q)}) = y_j^{(q)},$$

где $\boldsymbol{s}_j^{(q)}$ – взвешенная сумма нейрона j слоя q; $I_j^{(q)}$ – число входов нейрона j слоя q; j =1, 2, …, J_q; J_q – число нейронов в слое q; q =1, 2, …, Q; Q– число слоев в ИНС; $w_{ij}^{(q)} \cdot x_i^{(q)}$ – i-й вход нейрона j слоя q; f – функция активации; $y_j^{(q)}$ – выход нейрона j слоя q;

Шаг 3. Подстройка весов. Производится изменение весов синаптических связей $w_{ij}^{(q)}(n)$ на основании полученных выходных значений ИНС с помощью (III.2.4.1) или (III.2.4.2).

Шаг 4. Проверка обученности. Если значения выходов ИНС стабилизированы с заданной точностью, то считаем ИНС успешно обучившейся и завершаем работу. Иначе повторяем выполнение алгоритма начиная с шага 2, попеременно предъявляя все образы из входного набора, пока значения выходов ИНС не будут стабилизированы.

Следует отметить, что вид откликов на каждый класс входных образов заранее неизвестен и представляет собой произвольное сочетание состояний нейронов выходного слоя, обусловленное случайным распределением весов на стадии инициализации. Вместе с тем ИНС способна обобщать схожие образы, относя их к одному классу. Тестирование обученной ИНС позволяет определить топологию классов в выходном слое. Для приведения откликов обученной ИНС к удобному представлению можно дополнить ИНС одним слоем, который, например, по алгоритму обучения однослойного персептрона необходимо заставить отображать выходные реакции ИНС в требуемые образы.

Задания III.2.4:

1. Перечислите виды наблюдения, на которые опирается обучение ИНС методом Хебба.
2. Напишите формулу наращивание весов в обучении ИНС методом Хебба.
3. Напишите формулу метода обучения Хебба.
4. Опишите полный алгоритм обучения ИНС с помощью правил Хебба.
5. Напишите вычисление взвешенной суммы в методе Хебба.

Вопросы III.2.4:

1. На основе какого принципа построен метод Хебба?

НИИ искусственного интеллекта

2. Что усиливается при обучении искусственных нейронных сетей по методу Хебба?

3. Какие синапсы сильнее обучаются в дифференциальном методе обучения Хебба?

4. Как происходит проверка обученности в методе Хебба?

5. Что можно добавить для приведения откликов обученной ИНС к удобному представлению?

НИИ Искусственный интеллект

III.2.5. *Обучение нейронной сети методом Кохонена*

В этом параграфе нами будут рассмотрены вопросы обучения ИНС методом Кохонена, предложены задания и сформулированы вопросы. При подготовке учебного материла были использованы литературные источники [14, 17, 22, 27, 28, 33].

Кохонен предложил другой метод обучения без учителя, который называется *методом коррекции ошибки* и предусматривает подстройку весов синаптических связей на основании их значений на предыдущей итерации:

$$w_{ij}^{(q)}(t) = w_{ij}^{(q)}(t-1) + \eta \cdot \left[y_i^{(q-1)}(t) - w_{ij}^{(q)}(t-1)\right], \text{(III.2.5.1)}$$

где $w_{ij}^{(q)}(t)$ и $w_{ij}^{(q)}(t-1)$ – вес синапса от нейрона ***i*** к нейрону ***j*** слоя q на итерациях t и t-1 соответственно; η– коэффициент скорости обучения; $y_i^{(q-1)}(t)$ – выход нейрона i слоя (q-1) на итерации t; q; q =1, 2, …, Q; Q– число слоев в ИНС.

Из вышеприведенной формулы (III.2.5.1) видно, что обучение сводится к минимизации разницы между входными сигналами нейрона, поступающими с выходов нейронов предыдущего слоя $\boldsymbol{y}_i^{(q-1)}(\boldsymbol{t})$ и синаптическими весами. Процесс обучения ИНС состоит в коррекции исходных значений весов межнейронных синаптических связей.

Полный алгоритм обучения имеет примерно такую же структуру, как в методах Хебба, но на шаге 3 из всего слоя выбирается нейрон, значения весов синаптических связей которого максимально походят на входной образ, и подстройка весов по формуле (III.2.5.1) проводится только для него. Эта, так называемая, аккредитация может сопровождаться затормаживанием всех остальных нейронов слоя и введением выбранного нейрона в насыщение. Выбор такого нейрона может осуществляться, например, расчетом скалярного произведения вектора весов синаптических связей с вектором входных значений. Максимальное произведение дает выигравший нейрон.

Другой вариант обучения на основе минимизации разницы между входными сигналами нейрона, поступающими с выходов нейронов предыдущего слоя $\boldsymbol{y}_i^{(q-1)}(\boldsymbol{t})$ и синаптическими весами сводится к расчету расстояния:

$$D_j = \sqrt{\sum_{i=0}^{n-1} (y_i^{(q-1)} - w_{ij})^2}\ , \qquad \text{(III.2.5.2)}$$

где n – размер векторов, j – индекс нейрона в слое q, i – индекс суммирования по нейронам слоя (q-1), w_{ij} – вес синапса, соединяющего нейроны; выходы нейронов слоя (q-1) являются входными значениями для слоя q.

Корень в формуле (III.2.5.2) можно не брать, так как важна лишь относительная оценка различных D_j.

В данном случае, "побеждает" нейрон с наименьшим расстоянием. Иногда слишком часто получающие аккредитацию нейроны принудительно исключаются из рассмотрения, чтобы "уравнять права" всех нейронов слоя. Простейший вариант такого алгоритма заключается в торможении только что выигравшего нейрона. При использовании обучения по алгоритму Кохонена существует практика нормализации входных образов, а также – на стадии инициализации – и нормализации начальных значений весов синаптических связей.

$$x_i = \frac{x_i}{\sqrt{\sum_{j=0}^{n-1} x_j^2}}\ , \qquad \text{(III.2.5.3)}$$

где x_i – i-ая компонента вектора входного образа или вектора весов синаптических связей, а n – его размерность.

Это позволяет сократить длительность процесса обучения.

Инициализация весов синаптических связей случайными значениями может привести к тому, что различные классы, которым соответствуют плотно распределенные входные образы, сольются или, наоборот, раздробятся на дополнительные подклассы в случае близких образов одного и того же класса. Для избежания такой ситуации используется метод выпуклой комбинации. Суть его сводится к тому, что входные нормализованные образы подвергаются преобразованию:

$$x_i = \eta(t) \cdot x_i + (1 - \eta(t)) \cdot \frac{1}{\sqrt{n}}, \text{ (III.2.5.4)}$$

где x_i – i-ая компонента входного образа, n – общее число его компонент, $\eta(t)$ – коэффициент, изменяющийся в процессе обучения от нуля до единицы, в результате чего вначале на входы ИНС подаются практически одинаковые образы, а с течением времени они все больше сходятся к исходным.

Веса синаптических связей устанавливаются на шаге инициализации равными величине:

$$w_0 = 1/\sqrt{n},$$

где n – размерность вектора весов для нейронов инициализируемого слоя.

На основе рассмотренного выше метода строятся ИНС с подстраиваемыми синаптическими весами нейронов.

Задания III.2.5:

1. Напишите формулу другого варианта обучения нейронной сети по методу Кохонена.
2. Напишите формулу нормализации начальных значений весов синаптических связей.
3. Напишите формулу установки значений весов синаптических связей на шаге инициализации.

Вопросы III.2.5:

1. В чем заключается идея метода Кохонена?
2. В чем состоит обучение нейронной сети по методу Кохонена?
3. Из каких шагов состоит полный алгоритм обучения по методу Кохонена?

НИИ искусственный интеллект

4. Какой нейрон считается победителем во втором варианте обучения нейронной сети по методу Кохонена?

НИИ Искусственный интеллект

III.2.6. Обучение нейронной сети методом соревнования

В этом параграфе нами будут рассмотрены вопросы обучения ИНС методом соревнования, предложены задания и сформулированы вопросы. При подготовке учебного материла были использованы литературные источники [14, 17, 22, 27, 28, 33].

В отличие от метода Хебба, где множество выходных нейронов может активизироваться одновременно, при соревновательном обучении выходные нейроны соревнуются между собой за активизацию. Это явление известно, как правило "победитель получает все". Подобное обучение имеет место в биологических нейронных сетях. Обучение посредством соревнования позволяет проводить кластеризацию входных данных. Подобные примеры группируются нейронной сетью в соответствии с корреляциями и представляются одним элементом.

В процессе этого соревновательного обучения осуществляется модификация весов синаптических связей выигравшего нейрона и нейронов, расположенных в его окрестности. При обучении модифицируются только веса "победившего" нейрона. Эффект этого правила достигается за счет изменения сохраненного в ИНС образца (вектора весов победившего нейрона), при котором он становится чуть ближе к входному примеру. Нейрон с наибольшим выходным сигналом объявляется победителем и имеет возможность тормозить своих конкурентов и возбуждать соседей. Используется выходной сигнал нейрона-победителя и только ему и его соседям разрешается корректировать свои веса соединений по формуле:

$$\Delta w_{ij}(t+1) = w_{ij}(t) + \gamma[x_i - w_{ij}(t)]$$

На рисунке III.2.6.1 дана геометрическая иллюстрация обучения методом соревнования. Входные векторы нормализованы и представлены точками на поверхности сферы. Векторы весов для трех нейронов инициализированы случайными значениями. Их начальные значения (перед обучением) и конечные значения (после обучения) отмечены *X* на рисунке III.2.6.1(а) и III.2.6.1(б)

НИИ Искусственный интеллект

соответственно. Каждая из трех групп примеров обнаружена одним из выходных нейронов, чей вектор весов настроился на центр тяжести обнаруженной группы.

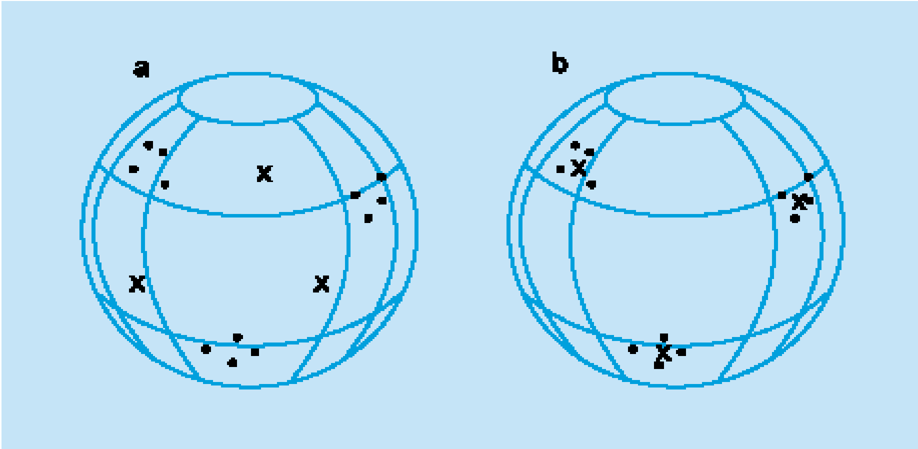


Рисунок III.2.6.1. Пример обучения методом соревнования:
(а) перед обучением; (б) после обучения

Можно заметить, что ИНС никогда не перестанет обучаться, если параметр скорости обучения не равен 0. Некоторый входной образец может активизировать другой выходной нейрон на последующих итерациях в процессе обучения. Это ставит вопрос об устойчивости обучающей системы. Система считается устойчивой, если ни один из примеров обучающей выборки не изменяет своей принадлежности к категории после конечного числа итераций обучающего процесса. Один из способов достижения стабильности состоит в постепенном уменьшении до 0 параметра скорости обучения. Однако это искусственное торможение обучения вызывает другую проблему, называемую пластичностью и связанную со способностью к адаптации к новым данным. Эти особенности обучения методом соревнования известны под названием дилеммы стабильности–пластичности Гроссберга, которая ставит вопрос:

"Как обучать новым явлениям (пластичность) и в то же время сохранить стабильность, чтобы существующие знания не были стерты или разрушены?"

НИИ искусственный интеллект

ИНС теории адаптивного резонанса, разработанные Карпентером и Гроссбергом (ART1, ART2 и ARTMAP), имеют способность решить эту дилемму. Такая сеть имеет достаточное число выходных элементов, но они не используются до тех пор, пока не возникнет в этом необходимость. Будем говорить, что элемент распределен (не распределен), если он используется (не используется). Обучающий алгоритм корректирует имеющийся прототип категории, только если входной вектор в достаточной степени ему подобен. В этом случае они резонируют. Степень подобия контролируется параметром сходства k, $0<k<1$, который связан также с числом категорий. Когда входной вектор недостаточно подобен ни одному существующему прототипу ИНС, создается новая категория, и с ней связывается нераспределенный элемент с входным вектором в качестве начального значения прототипа. Если не находится нераспределенного элемента, то новый вектор не вызывает реакции ИНС.

Чтобы проиллюстрировать модель, рассмотрим сеть ART1, которая рассчитана на бинарный вход: 0, 1. Упрощенная схема архитектуры ART1, которая содержит два слоя элементов с полными связями представлена на рисунке III.2.6.2.

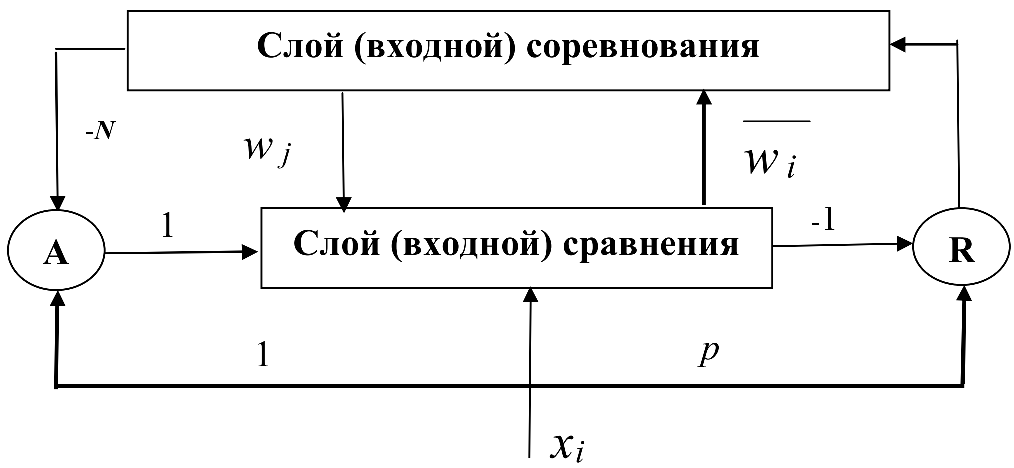


Рисунок III.2.6.2. Сеть ART1.

Направленный сверху вниз весовой вектор w_j соответствует элементу j входного слоя, а направленный снизу вверх весовой вектор $\overline{w_i}$ связан с выходным элементом i; $\overline{w_i}$ является нормализованной версией вектора w_i. Векторы w_j сохраняют прототипы кластеров.

Роль нормализации состоит в том, чтобы предотвратить доминирование векторов с большой длиной над векторами с малой длиной. Сигнал сброса R генерируется только тогда, когда подобие ниже заданного уровня.

Модель ART1 может создать новые категории и отбросить входные примеры, когда сеть исчерпала свою емкость. Однако число обнаруженных сетью категорий чувствительно к параметру сходства.

Специальным случаем ИНС, обучающейся методом соревнования, является самоорганизующаяся карта Кохонена, которая обладает благоприятным свойством сохранения топологии и определяет пространственную окрестность для каждого выходного элемента. В ней близкие входные примеры активизируют близкие выходные элементы. На рисунке III.2.6.3. показана основная архитектура карты Кохонена.

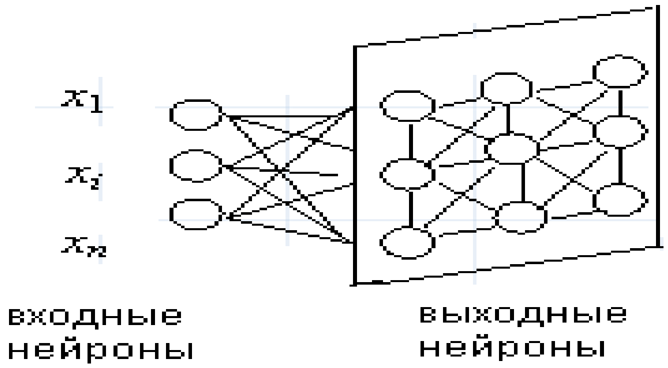


Рисунок III.2.6.3. Самоорганизующаяся карта Кохонена.

По существу, карта Кохонена представляет собой двумерный массив элементов, причем каждый элемент связан со всеми n входными узлами. Локальная окрестность может быть квадратом, прямоугольником или окружностью. Начальный размер окрестности часто устанавливается в пределах от 1/2 до 2/3 размера ИНС и сокращается согласно определенному закону (например, по экспоненциально убывающей зависимости). Во время обучения модифицируются все веса, связанные с победителем и его соседними элементами.

Задания III.2.6:

1. Постройте геометрическую иллюстрацию обучения методом соревнования.
2. Укажите условия продолжения обучения нейронной сети.
3. Сформулируйте условие устойчивости и пластичности обучающей системы.
4. Укажите один из способов достижения стабильности обучающей системы.
5. Сформулируйте дилемму стабильности–пластичности обучающей системы по Гроссбергу.
6. Назовите теорию, которая имеет способность решать дилемму стабильности–пластичности обучающей системы по Гроссбергу.
7. Опишите специальный случай ИНС, обучающейся методом соревнования, в виде самоорганизующейся карты Кохонена.

Вопросы III.2.6:

1. В чем отличие обучение нейронной сети методом соревнования от метода Хебба?
2. Какой принцип действует в методе соревнования?
3. Что позволяет обучать нейронную сеть методом соревнования?
4. Что осуществляется при обучении нейронной сети методом соревнования?
5. За счет чего достигается эффект правила «победителя» в обучении методом соревнования?
6. Чем представлены входные векторы в геометрической иллюстрации обучения методом соревнования?

НИИ искусственный интеллект

7. Может ли некоторый входной образец активизировать другой выходной нейрон на последующих итерациях в процессе обучения?

III.2.7. Соединение прямого метода и обратного метода.

В этом параграфе нами будут рассмотрены вопросы соединения прямого метода и обратного метода, предложены задания и сформулированы вопросы. При подготовке учебного материла были использованы литературные источники [14, 17, 22, 27, 28, 33].

При обучении ИНС метод обратного распространения ошибок обладает преимуществом над прямым методом, использующим распределение Коши, когда из-за случайности выбора многие шаги выполняются в неверном направлении и отнимают много времени.

Для ускорения обучения (поиска глобального минимума функции ошибки) можно соединить обратный и прямой методы обучения.

Соединение этих методов может дать хорошие результаты. Коррекция весов, равная взвешенной сумме, вычисленной алгоритмом обратного распространения, и случайный шаг, задаваемый алгоритмом Коши, приводят к системе, которая сходится и находит глобальный минимум быстрее, чем система, обучаемая каждым из методов в отдельности. Для того, чтобы избежать от паралича ИНС, который может иметь место, как при применении метода обратного распространении ошибки, так и при применении метода Коши, будет использована простая эвристика.

Рассмотрим трудности, возникающие в обоих этих методах:

1. Трудности метода обратного распространения ошибки.

Несмотря на мощь метода обратного распространения, при его применении возникает ряд трудностей, часть из которых облегчается благодаря использованию нового алгоритма.

***Сходимость*.**

Известно, что доказательство сходимости дается на языке дифференциальных уравнений в частных производных, которое

НИИ искусственный интеллект

справедливо лишь в том случае, когда коррекция весов выполняется с помощью бесконечно малых шагов в пространстве весов. Но бесконечно малые шаги ведут к бесконечному времени сходимости. Метод теряет силу в практических применениях, так как нет доказательства, что обратное распространение будет сходиться при конечном размере шага. Эксперименты показывают, что ИНС обычно обучаются, но время обучения велико и непредсказуемо.

Паралич.

При некоторых условиях ИНС может при обучении попасть в такое состояние, когда модификация весов не ведет к действительным изменениям ИНС. Такой «паралич» является серьезной проблемой. Один раз возникнув, он может увеличить время обучения на несколько порядков.

Паралич возникает в процессе обучения ИНС, когда у значительной части нейронов значения весов становятся очень большими, чтобы дать большие значения взвешенной сумме. Это может привести к тому, что все или большинство нейронов будут функционировать при очень больших аргументах сжимающей функции активации в области, где производная функции приближается к нулю (как, например, для сигмоидальной функции активации).

Как известно, что алгоритм обратного распространения при вычислении значении весов использует эту производную в формуле в качестве коэффициента (значение коррекции весов пропорционально значении производной), то процесс обучения может практически замереть и возникнуть паралич ИНС. Для пораженных параличом нейронов близость производной к нулю приводит к тому, что изменение веса становится близким к нулю.

Нет теории, способной предсказывать, будет ли сеть парализована во время обучения или нет. Экспериментально установлено, что малые размеры шага реже приводят к параличу, но шаг, малый для одной задачи, может оказаться большим для другой. Цена же паралича может быть высокой.

НИИ Искусственный интеллект

При моделировании многие часы машинного времени могут уйти на то, чтобы выйти из паралича. Обычно этого избегают уменьшением размера шага η, но это увеличивает время обучения и замедляется сходимость алгоритма.

Различные эвристики использовались для предохранения от паралича или для восстановления после него, но пока что они могут рассматриваться лишь как экспериментальные. Для преодоления такой ситуации существуют модификации алгоритма, учитывающие лишь направление антиградиента и не учитывающие его величину. Также имеется адаптивный алгоритм выбора шага, автоматически корректирующий размер шага в процессе обучения.

Локальные минимумы.

В методе обратного распространения ошибки для коррекции значений весов синаптических связей ИНС используется градиентный спуск по поверхности ошибки в пространстве весов синаптических связей в направлении к локальному минимуму. Но локальный минимум в одних случаях является приемлемым решением, в других случаях – неприемлемым.

Обычно на практике поверхность ошибки сложной ИНС сильно изрезана и состоит из холмов, долин, складок и оврагов в пространстве высокой размерности. ИНС может попасть в локальный минимум и не обнаружить глобальный минимум. В точке локального минимума все направления ведут вверх, и ИНС неспособна из него выбраться.

Для преодоления этой трудности обучения ИНС расширяют размерность пространства весов за счет увеличения скрытых слоев и числа нейронов скрытого слоя или несколько раз проводят обучение и затем выбирают тот вариант обученной ИНС, который дает наилучшие результаты. Также из локального минимума можно вывести на короткое время, увеличив скорость обучения η. Иногда, к изменению значений весов синаптических связей добавляют шум. Это также позволяет ИНС «выпрыгивать» из локальных минимумов.

НИИ искусственный интеллект

Даже после того как сеть обучена, невозможно сказать, найден ли с помощью обратного распространения глобальный минимум. Если решение неудовлетворительно, приходится давать весам новые начальные случайные значения и повторно обучать сеть без гарантии, что обучение закончится на этой попытке или что глобальный минимум вообще когда-либо будет найден.

Переобучение.

Высокая точность на обучающей выборке, может привести к неустойчивости результатов на тестовой выборке. Чем лучше ИНС адаптирована к конкретным условиям (обучающей выборке), тем меньше она способна к обобщению и экстраполяции. В этом случае ИНС моделирует не функцию, а шум, присутствующий в обучающей выборке. Это явление называется *переобучением*. Кардинальное средство борьбы с этим недостатком – использование подтверждающей выборки примеров, которая используются для выявления переобучения ИНС. Ухудшение характеристики ИНС при работе с подтверждающей выборкой указывает на возможное переобучение. Напротив, если ошибка последовательно уменьшается при подаче примеров из множества подтверждающих данных, то ИНС продолжает обучаться.

Недостатком этого приема является уменьшение числа примеров, которое можно использовать в обучающем множестве, так как уменьшение количества обучающей выборки снижает качество работы ИНС. Кроме того, возникает проблема оптимального разбиения исходных данных на обучающую, тестовую и подтверждающую выборку. Даже при случайной выборке разные разбиения базы данных дают разные оценки.

2. Трудности с алгоритмом обучения Коши.

Несмотря на улучшение скорости обучения, предоставляемое машиной Коши по сравнению с машиной Больцмана, время сходимости все еще может в 100 раз превышать время для алгоритма обратного распространения. Отметим, что сетевой

НИИ Искусственный интеллект

паралич особенно опасен для алгоритма обучения Коши, в особенности для ИНС с нелинейностью типа логистической функции. Бесконечная дисперсия распределения Коши приводит к изменениям весов неограниченной величины. Далее, большие изменения весов будут иногда приниматься даже в тех случаях, когда они неблагоприятны, часто приводя к сильному насыщению сетевых нейронов с вытекающим отсюда риском паралича.

Задания III.2.7:

1. Укажите причину, по которой прямой метод обладает недостатком по сравнению с методом обратного распространения ошибок.
2. Перечислите все положительные результаты, которые можно получить при соединении прямого метода и обратного метода обучения нейронной сети.
3. Перечислите все трудности обучения нейронной сети методом обратного распространения ошибки.
4. Перечислите все трудности обучения нейронной сети прямым методом Коши.
5. Объясните причину возникновения переобученности искусственной нейронной сети.

Вопросы III.2.7:

1. Почему возникла необходимость соединения прямого метода и обратного метода обучения нейронной сети?
2. Какие результаты можно получить при соединении прямого метода и обратного метода обучения нейронной сети?
3. Что такое сходимость обучения искусственной нейронной сети?
4. Когда возникает паралич обучения искусственной нейронной сети?
5. Имеется ли теория для предугадывания паралича обучения нейронной сети?

НИИ искусственный интеллект

6. Какой метод используется для нахождения локального минимума при обучении нейронной сети методом обратного распространения ошибки?

7. Найдется ли глобальный минимум после того как искусственная нейронная сеть будет обучена с помощью обратного распространения?

НИИ Искусственный интеллект

III.2.8. Комбинация метода обратного распространения ошибки и метода Коши

В этом параграфе нами будут рассмотрены вопросы комбинации метода обратного распространения ошибки и метода Коши, предложены задания и сформулированы вопросы. При подготовке учебного материла были использованы литературные источники [14, 17, 22, 27, 28, 33].

Коррекция весов синаптических связей в комбинированном алгоритме обучения нейронной сети, использующего метод обратного распространения ошибки и метод Коши, состоит из двух компонент: (1) направленной компоненты, вычисляемой с использованием алгоритма обратного распространения ошибки, и (2) случайной компоненты, определяемой распределением Коши.

Эти компоненты вычисляются для каждого веса, и их сумма является величиной, на которую изменяется вес. Как и в алгоритме Коши, после вычисления изменения веса вычисляется целевая функция. Если имеет место улучшение, изменение сохраняется. В противном случае оно сохраняется с вероятностью, определяемой распределением Больцмана.

Коррекция веса вычисляется с использованием представленных ранее уравнений для каждого из алгоритмов:

$$w_{ij}^{(q)}(t+1) = w_{ij}^{(q)}(t) + \eta[\alpha \Delta w_{ij}^{(q)}(t) + (1-\alpha)\delta_j^{(q)} y_{ij}] + (1-\eta)x_c$$

где $w_{ij}^{(q)}(t+1)$ – вес синапса *j*-го нейрона *q*-го слоя с *i*-м нейроном (*q-1*)-го слоя на (*t*+1) шаге итерации, $\Delta w_{ij}^{(q)}$ – коррекция весов синаптических связей, η – коэффициент скорости обучения, управляющий относительными величинами Коши и обратного распространения в компонентах весового шага, q=Q-1,Q-2,…, 2; $0<\eta<1$; $y_{j,k}^{(Q)}$ – реальный выход *Q*-го выходного слоя ИНС для *j*-го нейрона на *k*-м обучающем примере (образа).

Если η приравнивается нулю, система становится полностью машиной Коши. Если η приравнивается единице, система становится машиной обратного распространения.

НИИ Искусственный интеллект

Изменение лишь одного веса между вычислениями весовой функции неэффективно. Оказалось, что лучше сразу изменять все веса целого слоя, хотя для некоторых задач может оказаться выгоднее иная стратегия.

Преодоление сетевого паралича комбинированным методом обучения. Как и в машине Коши, если изменение веса ухудшает целевую функцию, с помощью распределения Больцмана решается, сохранить ли новое значение веса или восстановить предыдущее значение. Таким образом, имеется конечная вероятность того, что ухудшающее множество приращений весов будет сохранено. Так как распределение Коши имеет бесконечную дисперсию (диапазон изменения тангенса находится от $-\infty$ до $+\infty$), то весьма вероятно возникновение больших приращений весов, часто приводящих к сетевому параличу.

Очевидное решение, состоящее в ограничении диапазона изменения весовых шагов, ставит вопрос о математической корректности полученного таким образом алгоритма. В действительности экспериментально выявлены случаи, когда для реализации некоторой функции требуются большие веса, и два больших веса, вычитаясь, дают малую разность.

Другое решение состоит в рандомизации весов тех нейронов, которые оказались в состоянии насыщения. Недостатком такого решения является то, что оно может серьезно нарушить обучающий процесс, иногда затягивая его до бесконечности.

Для решения проблемы паралича был найден метод, не нарушающий достигнутого обучения. Насыщенные нейроны выявляются с помощью измерения их выходных сигналов. Когда величина этих сигналов приближается к своему предельному значению, положительному или отрицательному, на веса синаптических связей, питающие этот нейрон, действует сжимающая функция. Она подобна используемой для получения нейронного выходного сигнала, за исключением того, что диапазоном ее изменения является интервал (+5,–5) или другое

НИИ искусственный интеллект

подходящее множество. Тогда модифицированные значения весовых коэффициентов равны

$$w_{\mathrm{mn}} = \frac{-5+10}{1+\exp(-w_{\mathrm{mn}}/5)} .$$

Эта функция сильно уменьшает величину очень больших весов. Воздействие на малые веса значительно более слабое. Далее она поддерживает симметрию, сохраняя небольшие различия между большими весами. Экспериментально было показано, что эта функция выводит нейроны из состояния насыщения без нарушения достигнутого в ИНС обучения. При оптимизации используемой функции не было затрачено серьезных усилий , также другие значения констант могут оказаться более эффективными.

Экспериментальное результаты. Комбинированный алгоритм, использующий обратное распространение и обучение Коши, применялся для обучения нескольких больших сетей. Например, этим методом была успешно обучена система, распознающая рукописные китайские иероглифы. Тем не менее время обучения может оказаться длительным (приблизительно 36 часов машинного времени ушло на обучение).

В другом эксперименте эта сеть обучалась на задаче *исключающее или*, которая была использована в качестве теста для сравнения с другими алгоритмами. Для сходимости ИНС в среднем требовалось около 76 предъявлении обучающего множества. В качестве сравнения можно указать, что при использовании обратного распространения в среднем требовалось около 245 предъявлении для решения этой же задачи и 4 986 итераций при использовании обратного распространения второго порядка.

Ни одно из обучений не привело к локальному минимуму. Более того, ни одно из 160 обучений не обнаружило неожиданных патологий, сеть всегда обучалась корректно.

Эксперименты же с чистой машиной Коши привели к значительно большим периодам времени обучения. Например, при

НИИ искусственный интеллект

ρ = 0,002 для обучения ИНС в среднем требовалось около 2 284 предъявлении обучающего множества.

Обсуждение. Комбинированная сеть, использующая обратное распространение и обучение Коши, обучается значительно быстрее, чем каждый из алгоритмов в отдельности, и относительно нечувствительна к величинам коэффициентов. Сходимость к глобальному минимуму гарантируется алгоритмом Коши, в сотнях экспериментов по обучению сеть ни разу не попадала в ловушки локальных минимумов. Проблема сетевого паралича была решена с помощью алгоритма селективного сжатия весов, который обеспечил сходимость во всех предъявленных тестовых задачах без существенного увеличения обучающего времени.

Несмотря на такие обнадеживающие результаты, метод еще не исследован до конца, особенно на больших задачах. Значительно большая работа потребуется для определения его достоинств и недостатков.

Задания III.2.8:

1. Опишите работу каждого компонента комбинированного алгоритма обучения нейронной сети.
2. Напишите формулу вычисления коррекции весов синаптических связей для комбинированного метода обучения нейронных сетей.
3. Напишите формулу модифицированных значений весовых коэффициентов.

Вопросы III.2.8:

1. Как производится коррекция весов в комбинированном методе обучения нейронной сети?
2. Комбинированная система обучения нейронных сетей в каком случае становится машиной Коши и в каком случае - машиной обратного распространения?
3. Как будет преодолеваться сетевой паралич в комбинированном методе обучения нейронных сетей?

НИИ Искусственный интеллект

III.2.9. Настройка числа нейронов в процессе обучения ИНС

В этом параграфе нами будут рассмотрены вопросы настройки числа нейронов в процессе обучения ИНС, предложены задания и сформулированы вопросы. При подготовке учебного материла были использованы литературные источники [14, 17, 22, 27, 28, 33].

Для того, чтобы многослойная ИНС реализовывала заданное обучающей выборкой отображение, она должна иметь достаточное число нейронов в скрытых слоях.

В настоящее время нет формул для точного определения необходимого числа нейронов в ИНС по заданной обучающей выборке. Однако предложены способы настройки числа нейронов в процессе обучения, которые обеспечивают построение ИНС для решения задачи и дают возможность избежать избыточности. Эти способы настройки можно разделить на две группы: *алгоритмы сокращения* (*pruning algorithms*), *конструктивные алгоритмы* (*constructive algorithms*).

1. Алгоритмы сокращения.

В начале работы алгоритма обучения с сокращением число нейронов в скрытых слоях ИНС заведомо избыточно. Затем из ИНС постепенно удаляются синапсы и нейроны. Существуют два подхода к реализации алгоритмов сокращения:

1) *Метод штрафных функций.*

В целевую функцию алгоритма обучения вводится штрафы за то, что значения весов синаптических связей отличны от нуля, например:

$$C = \sum_{i=1}^{I} \sum_{j=1}^{J} w_{ij}^2$$

где w_{ij} – вес, i – номер нейрона, j – номер входа, I – размерность входного сигнала, J – число нейронов скрытого слоя.

2) *Метод проекций.*

Вес синапса обнуляется, если его значение попало в заданный диапазон:

$$w_{ij}=\begin{cases} 0, & |w_{ij}|\leq\varepsilon \\ w_{ij}, & |w_{ij}|>\varepsilon \end{cases}$$

где ε – некоторая константа.

Алгоритмы сокращения имеют, по крайней мере, два недостатка:

– нет методики определения числа нейронов скрытых слоев, которое является избыточным, поэтому перед началом работы алгоритма нужно угадать это число;

– в процессе работы алгоритма сеть содержит избыточное число нейронов, поэтому обучение идет медленно.

2. Конструктивные алгоритмы.

Предшественником конструктивных алгоритмов можно считать методику обучения многослойных нейронных сетей, включающую в себя следующие шаги:

Шаг 1. Выбор начального числа нейронов в скрытых слоях.

Шаг 2. Инициализация ИНС, заключающаяся в присваивании весам синапса и смещениям ИНС случайных значений из заданного диапазона.

Шаг 3. Обучение ИНС по заданной выборке.

Шаг 4. Завершение в случае успешного обучения. Если не удалось обучить ИНС, то число нейронов увеличивается и повторяются шаги со второго шага по четвертый шаг.

В конструктивных алгоритмах число нейронов в скрытых слоях также изначально мало и постепенно увеличивается. В отличие от этой методики, в конструктивных алгоритмах сохраняются навыки, приобретенные сетью до увеличения числа нейронов.

Конструктивные алгоритмы различаются правилами задания значений параметров в новых, добавленных в сеть, нейронах:

– значения параметров являются случайными числами из заданного диапазона;

– значения весов синаптических связей нового нейрона определяются путем *расщепления* (*splitting*) одного из старых нейронов.

Первое правило не требует значительных вычислений, однако его использование приводит к увеличению значения функции ошибки после каждого добавления нового нейрона. В результате случайного задания значений параметров новых нейронов может появиться избыточность в числе нейронов скрытого слоя. Расщепление нейронов лишено двух указанных недостатков. Суть алгоритма заключается в выявлении и расщеплении таких нейронов. В результате расщепления вместо одного исходного в ИНС оказывается два нейрона. Первый из этих нейронов имеет вектор весов, представляющий из себя сумму вектора весов исходного нейрона и векторов изменений весов одного из преимущественных направлений. В результате суммирования векторов изменений весов другого преимущественного направления и вектора весов исходного нейрона получают веса второго нового нейрона.

На рисунке III.2.9 показан вектор весов нейрона скрытого слоя на некотором шаге обучения и векторы изменения весов, соответствующие отдельным обучающим примерам. Векторы изменений имеют два преимущественных направления и образуют в пространстве область, существенно отличающуюся от сферической.

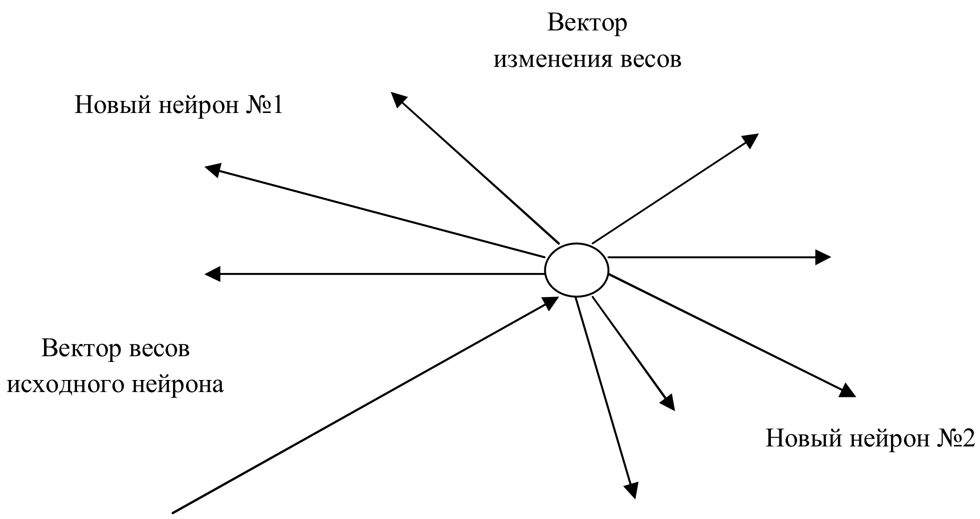


Рисунок III.2.9. Вектор весов нейрона скрытого слоя и зменения, соответствующие отдельным обучающим примерам.

Расщеплять нейроны, векторы изменений которых имеют два преимущественных направления, необходима потому, что наличие таких нейронов приводит к осцилляциям при обучении методом обратного распространения. При обучении методом с интегральной функцией ошибки, наличие таких нейронов приводит к попаданию в локальный минимум с большим значением ошибки.

Алгоритм расщепления включает в себя:

– построение ковариационной матрицы векторов изменений весов синаптических связей;

– вычисление собственных векторов и собственных значений полученной матрицы с помощью итерационного алгоритма Ойа (Oja), в соответствии с которым выполняется стохастический градиентный подъем и ортогонализация Грамма–Шмидта.

Недостатком алгоритма является экспоненциальный рост времени вычислений при увеличении размерности ИНС.

Для преодоления указанного недостатка предложен упрощенный алгоритм расщепления, который не требует значительных вычислений. Общее число нейронов в ИНС, построенной с помощью этого алгоритма по заданной обучающей выборке, может быть несколько больше, чем у ИНС, построенной с помощью исходного алгоритма.

В упрощенном алгоритме для расщепления выбирается нейрон с наибольшим значением функционала:

$$F_j = \frac{\sum_{k=1}^{N} \left|\delta w_j^k\right|}{\left|\sum_{k=1}^{N} \delta w_j^k\right|},$$

где δw_j^k – вектор изменений весов синаптических связей нейрона, k – номер обучающего примера, j – номер нейрона, N – число примеров в обучающей выборке, j = 1, 2, ..., M; M – число нейронов.

Таким образом, в качестве критерия выбора нейрона для расщепления используется отношение суммы длин векторов изменений весов синаптических связей нейрона, соответствующих различным обучающим примерам, к длине суммы этих векторов.

В результате расщепления вместо исходного нейрона в сеть вводятся два новых нейрона. Значение каждого синаптического веса нового нейрона есть значение соответствующего веса старого нейрона плюс некоторый небольшой шум. Величины весов связей выходов новых нейронов и нейронов следующего слоя равны половине величин весов связей исходного нейрона с соответствующими весами следующего слоя. Упрощенный алгоритм, как и исходный, гарантирует, что функция ошибки после расщепления увеличиваться не будет.

Кроме описанных способов выбора нейронов для расщепления, может быть использован анализ чувствительности, в процессе которого строятся матрицы Гессе для вторых производных функции ошибки по параметрам ИНС. По величине модуля второй производной судят о важности значения данного параметра для решения задачи. Параметры с малыми значениями вторых производных обнуляют. Анализ чувствительности требует больших вычислительных ресурсов.

Задания III.2.9:

1. Опишите метод штрафных функций.
2. Напишитте формулу для штрафной функции?
3. Опишите метод проекций.
4. Опишите конструктивные алгоритмы.
5. Укажите признаки различия конструктивных алгоритмов.

Вопросы III.2.9:

1. Какие имеются подходы к реализации алгоритмов сокращения.
2. Как выглядит формула функционала для расщепления?
3. Что включает алгоритм расщепления?
4. Каким недостатком обладает алгоритм расщепления?

НИИ искусство

5. Какой критерий выбора нейрона используется для расщепления?

НИИ Искусственный интеллект

III.2.10. Оценка качества работы нейронной сети

В этом параграфе нами будут рассмотрены вопросы оценки качества работы искусственных нейронных сетей, предложены задания и сформулированы вопросы. При подготовке учебного материла были использованы литературные источники [14, 17, 22, 27, 28,3 3].

В предыдущих параграфах было показано, что обучение ИНС основывается на минимизации значения некоторой функции, показывающей отклонение результатов, которые выдает ИНС на данном множестве обучающих примеров от требуемых. В свою очередь целью минимизации является отыскание глобального минимума. Достижение этого называется *сходимостью обучения*. Поиск глобального минимума осуществляется посредством итерационного процесса – *обучающего алгоритма*, который исследует поверхность невязки и стремится обнаружить на ней точку глобального минимума.

В настоящее время существуют сотни разных обучающих алгоритмов, отличающихся друг от друга стратегией оптимизации и критерием (мерой) ошибки. Удачный выбор критерия ошибки приводит к более гладкой поверхности невязки и упрощает задачу обучения. Самой простой оценкой является средняя квадратичная ошибка (*MSE – Mean Squared Error*), определяемая как усредненная на K обучающих примерах сумма квадратов разностей между значениями желаемого выхода d_k и реально полученного выхода y_k для каждого обучающего примера k:

$$E(w)=\frac{1}{K}\cdot\sum_{k=1}^{K}(d_k-y_k)^2 \quad \text{(III.2.10.1)}$$

Оценка *MSE* используется в тех случаях, когда выходные сигналы ИНС должны с заданной и одинаковой для всех сигналов точностью ε совпадать с известными векторами, где ε – *уровень надежности обучения*. Для учета уровня надежности обучения используется модифицированная оценка *MSE:*

НИИ искусственный интеллект

$$E(w) = \frac{1}{K} \cdot \sum_{k=1}^{K} \frac{(d_k - y_k)^2}{\varepsilon}, \text{ (III.2.10.2)}$$

где K – размерность входного вектора, $\boldsymbol{\varepsilon}$ **–** имеет различный диапазон изменения в зависимости от способов интерпретации:

$0 < \boldsymbol{\varepsilon} < 1$ – для знаковой интерпретации;

$0 < \boldsymbol{\varepsilon} \le 2$ – для правила «победитель получает все»;

$0 < \boldsymbol{\varepsilon} \le 2/(K\text{-}1)$ – для порядковой интерпретации.

Уровень надежности обучения вводится для обеспечения устойчивости ИНС. Критерий устойчивости ИНС формулируется следующим образом: работа ИНС считается устойчивой, если при изменении выходных сигналов ИНС на величину, меньшую $\boldsymbol{\varepsilon}$**,** интерпретация ответов не меняется. Этот критерий можно использовать для обеспечения ускоренного обучения ИНС. Целесообразно при вычислении оценки по формуле (III.2.10.2) использовать только такое множество правильных ответов, интерпретация которых не меняется при изменении их значений на величину, меньшую $\boldsymbol{\varepsilon}$**.**

Оценку *MSE* можно обобщить, если использовать суммирование квадратов разностей $(d_k - y_k)^2$ с весами:

$$E(w) = \frac{1}{K} \cdot \sum_{k=1}^{K} v_k (d_k - y_k)^2, \text{ (III.2.10.3)}$$

где v_k – вес k–го примера в обучающей выборке.

Использование оценки (III.2.10.3) позволяет выделить наиболее важные примеры из обучающей выборки, устанавливая для этого соответствующий вес v_k.

Кроме оценки *MSE* используют также и оценку Кульбака-Лейблера, связанную с критерием максимума правдоподобия:

$$E(w) = \sum_{k=1}^{M} d_k \log \frac{d_k}{y_k} + (1 - d_k) \log \frac{1 - d_k}{1 - y_k},$$

где M – число выходов ИНС.

НИИ Искусственный интеллект

Более простыми являются оценки качества работы ИНС, часто используемые при аппаратной реализации ИНС (например, ZISC Accelerator cards для IBM PC) и нейроимитаторах:

$$E(w)=\sum_{k=1}^{M}\left|d_k-y_k\right| \;,\; E(w)=\max_{M}\left|d_k-y_k\right|.$$

Для решения задач анализа временного ряда целесообразно использовать оценку по средней относительной вариации:

$$arv(s)=\frac{\sum\limits_{t\in S}(d_t-x_t)^2}{\sum\limits_{t\in S}(d_t-\langle x_t\rangle)^2}=\frac{\sum\limits_{t\in S}e_t^2}{N\sigma^2},$$

где S – временной ряд, e_t – разность между истинным значением d_t и x_t в момент времени t, $<x_t>$ – оценка для среднего значения временного ряда, N – число данных в временном ряду.

Задания III.2.10:

1. Объясните смысл формулы $E(w)=\frac{1}{K}\cdot\sum\limits_{k=1}^{K}(d_k-y_k)^2$.
2. Объясните смысл формулы $E(w)=\frac{1}{K}\cdot\sum\limits_{k=1}^{K}\frac{(d_k-y_k)^2}{\varepsilon}$.
3. Объясните смысл формулы $E(w)=\frac{1}{K}\cdot\sum\limits_{k=1}^{K}v_k(d_k-y_k)^2$.
4. Объясните смысл формулы $E(w)=\sum\limits_{k=1}^{M}d_k\log\frac{d_k}{y_k}+(1-d_k)\log\frac{1-d_k}{1-y_k}$.
5. Объясните смысл формулы $E(w)=\sum\limits_{k=1}^{M}\left|d_k-y_k\right|$.

Вопросы III.2.10:

1. На чем основывается обучение ИНС?
2. Что такое сходимость обучения?
3. Как формулируется критерий устойчивости ИНС?
4. Как осуществляется поиск глобального минимума?
5. Для чего вводится уровень надежности обучения?

НИИ искусственный интеллект

НИИ Искусственный интеллект

III.2.11. Сравнение методов обучения

В этом параграфе приведены сравнения искусственных нейронных сетей (ИНС), предложены задания и сформулированы вопросы. При этом использованы источники [14, 17, 22, 27, 28, 33].

Многими исследователями были предложены улучшения и обобщения описанных выше методов обучения. Литература в этой области слишком обширна, чтобы ее можно было здесь охватить. Более того, сейчас еще слишком рано давать окончательные оценки. Некоторые из этих подходов могут оказаться действительно фундаментальными, другие же со временем исчезнут.

В Приложении 3 представлены алгоритмы обучения и связанные с ними парадигмы обучения, обучающие правила, архитектуры сетей, решаемые задачи (список не является исчерпывающим). Каждый алгоритм обучения имеет свою парадигму и правило обучения, ориентирован на сеть определенной архитектуры и предназначен для ограниченного класса задач.

В Таблице Приложения 4 приводятся характеристики нейронных сетей: области применения, недостатки и преимущества различных нейронных сетей (список не является исчерпывающим).

Задания III.2.11:

1. Указать критерии по которым сравнивались алгоритмы обучения искусственных нейронных сетей.
2. Указать характеристики по которым сравнивались искусственные нейронные сети.
3. Указать предметную область, где искусственные нейронные сети наиболее применяются.

Вопросы III.2.11:

1. Для решения каких задач применяются искусственные нейронные сети?
2. Какая парадигма обучения приемлема для решения задач сжатия данных?

НИИ искусственный интеллект

3. У каких искусственных нейронных сетей больше преимуществ, чем недостатков?

III.3. Обучение конкретных нейронных сетей

III.3.1. Обучение персептрона

В этом параграфе будут обсуждаться вопросы обучения персептрона, предложены задания и сформулированы вопросы. При подготовке учебного материла были использованы литературные источники [14, 17, 22, 27, 28, 33].

Используя критерий линейной разделимости, можно решить, способен ли однослойный персептрон реализовывать требуемую функцию. Даже в том случае, когда ответ положительный, это принесет мало пользы, если у нас нет способа найти нужные значения для весов и порогов. Если будет процедура для вычисления этих значений, которая может быть реализована на компьютере или другом электронном устройстве, то персептрон становится в определенном смысле самонастраивающимся. По этой причине процедуру настройки весов обычно называют алгоритмом обучения и говорят, что сеть «обучается».

Впервые Розенблатт разработал алгоритм обучения персептрона с учителем и доказал, что персептрон может быть обучен всему, что он может реализовывать.

В алгоритме обучения персептрона, предложенного Розенблаттом, обучающая выборка состоит из множества входных векторов, для каждого из которых указан свой требуемый выход, и предусмотрены следующие шаги:

В общем случае перептрон можно обучить с помощью следующего алгоритма:

1. Присвоить синаптическим весам $w_1, w_2, \dots, w_n$ некоторые начальные значения. Например, нулю.
2. Подать входной образ $X = (x_1, x_2, \dots, x_n)$ и вычислить y. Если y правильный, то переходят к шагу 4. Иначе к шагу 3.
3. Применяя дельта-правило (см. ниже) вычислить новые значения весов синаптических связей.

НИИ Искусственный интеллект

4. Повторить шаги 2-4 данного алгоритма обучения персептрона пока сеть не станет выдавать ожидаемый выход на векторах из обучающей выборки или пока отклонение не станет ниже некоторого порога.

Таким образом, если сигнал персептрона при некотором образе верен, то ничего корректировать не надо, если нет – производится корректировка весов.

Имеются следующие правила корректировки весов:

1. Если выход y неверен и равен нулю, то необходимо увеличить веса тех входов, на которые была подана единица.

2. Если выход y неверен и равен единице, то необходимо уменьшить веса тех входов, на которые была подана единица.

Поясним эти правила. Допустим, что на вход был подан некоторый обучающий двоичный вектор X. Этому вектору соответствует выход y=1. И этот выход неправильный. Тогда веса, присоединенные к единичным входам, должны быть уменьшены, так как они стремятся дать неверный результат. Аналогично, если некоторому другому обучающему вектору X соответствует неправильный выход y=0, то веса, присоединенные к единичным входам, должны быть уже уменьшены.

Теперь приведем *дельта-правило*, которое является математической моделью правил корректировки весов.

Введем величину δ, которая равна разности между требуемым d и реальным y выходом:

$$\delta = d - y$$

Тогда, веса персептрона после коррекции будут равны:

$$w_i(t+1) = w_i(t) - \eta\delta(t) x_i,$$

где x_i – величина входа соответствующая w_i синаптическому весу, t – номер текущей итерации обучения персептрона, η – коэффициент скорости обучения, позволяет управлять средней величиной изменения весов.

Добавление величины x_i в произведение позволяет избежать изменение тех весов, которым на входе соответствовал ноль.

НИИ Искусственный интеллект

Шаг 1. Инициализация смещения w_0 и весов синаптических связей $w_i(0)$, $(i=1,2,...,N)$ некоторыми малыми случайными числами.

Шаг 2. Предъявление персептрону нового образа: входного вектора $(x_1,..., x_N)$ и требуемого выходного сигнала d.

Шаг 3. Вычисление выходного сигнала персептрона:

$$y(t) = f(\sum_{i=1}^{N} w_i(t) \cdot x_i(t) + w_0).$$

Шаг 3. Проверить правильность реального выхода:

a) если реальный выход правильный, то перейти на шаг 4, иначе,

b) если реальный выход неправильный и равен нулю, то добавить все входы к соответствующим им весам, иначе,

c) если реальный выход неправильный и равен единице, то вычесть каждый вход из соответствующего ему веса.

Перейти на шаг 2.

Шаг 4. Если еще имеется не предъявленный образ, то перейти на шаг 2, иначе считается, что персептрон обучен, и завершить работу.

Таким образом персептрон обучают, подавая множество образов по одному на его вход и настраивая веса до тех пор, пока для всех образов не будет достигнут требуемый выход.

Важное обобщение алгоритма обучения персептрона, называемое *дельта-правилом*, переносит этот метод на непрерывные входы и выходы. Чтобы понять, как оно было получено, шаг 3 алгоритма обучения персептрона может быть сформулирован в обобщенной форме с помощью введения величины δ, которая равна разности между выходом d и реальным выходом y, т.е. $\delta = (d - y)$. При этом:

случай $\delta = 0$ соответствует шагу 3 a), когда выход правильный и в персептроне ничего не изменяется;

НИИ Искусственный интеллект

случай $\delta > 0$ соответствует шагу 3 b), когда выход неправильный и равен нулю в персептроне активация увеличивается;
случай $\delta < 0$ соответствует шагу 3 c), когда выход неправильный и равен единице в персептроне активация уменьшается.

В любом из этих случаев алгоритм обучения персептрона сохраняется, если δ умножается на значение каждого входа x_i и это произведение добавляется к соответствующему весу синаптической связи. С целью обобщения вводится коэффициент «скорости обучения» η ($0<\eta<1$), который умножается на δx_i, что позволяет управлять средней величиной изменения весов. В алгебраической форме записи это выглядит так:

$$w_i(t+1) = w_i(t) + \eta \cdot (d - y(t)) \cdot x_i(t)$$

где $w_i(t)$ – значение i-того веса до коррекции, $w_i(t+1)$ – значение i-того веса после коррекции.

Дельта-правило модифицирует веса синаптических связи персептрона в соответствии с требуемым и действительным значениями выхода каждой полярности как для непрерывных, так и для бинарных входов и выходов. Эти свойства открыли множество новых приложений.

Розенблатт доказал, что если образы (векторы), используемые для обучения персептрона, выбраны из двух линейно-разделимых классов, то алгоритм персептрона сходится и формирует поверхность решений в форме гиперплоскости, разделяющей эти два класса. Увеличивая размерность выходного (вычислительного) слоя персептрона и включая в него несколько нейронов, можно решать задачи классификации на большее число классов.

Обучение персептрона требует учителя, т.е. требует множества пар <вектор входных сигналов X^k, ожидаемое значение выходного сигнала d^k_i>. Обучение (отыскание весов w_{ij}) сводится к задаче минимизации целевой функции

$$E(W_i) = \frac{1}{2} \cdot \sum_{k=1}^{p} (y_i^k - d_i^k)^2 .$$

НИИ Искусственный интеллект

К сожалению, для персептрона в силу разрывности функции $f(s_i)$ для отыскания минимума $E(W_i)$ применимы методы оптимизации только нулевого порядка. На практике для обучения персептрона чаще всего используется *правило персептрона*, представляющее собой следующий простой алгоритм.

1. Выбираются (как правило, случайно) начальные значения весов w_{ij} (j=0, 1, 2, ..., N) нейрона.

2. Для каждой обучающей пары $<X^k, d^k_i>$ выполняется ряд циклов (их номера обозначим через t) уточнения значений входных весов по формуле:

$$w_{ij}(t+1) = w_{ij}(t) + \delta \cdot w_{ij}(t),$$

здесь:

$\delta w_{ij}(t)=0$, если $y_i(t)=\delta^k_i$;

$\delta w_{ij}(t)=x^k_j$, если $y_i(t)=0$, $\alpha\, \delta^k_i=1$;

$\delta w_{ij}(t)=-x^k_j$, если $y_i(t)=1$, $\alpha\, \delta^k_i=0$;

где α – скорость или шаг обучения.

Процесс обработки текущей обучающей пары завершается либо:

а) на цикле, в котором все $\delta w_{ij}(t) = 0$;

б) после достижения предельного количества циклов.

Следует отметить, что правило персептрона представляет собой частный случай предложенного много позже универсального правила, называемое *дельта-правилом* (*delta rule*), которое может использоваться как для обучения линейного персептрона, так и для персептрона с пороговой функцией активации. Дельта-правило было предложено в 60-х гг. Ф. Розенблаттом для персептрона с пороговой функцией активации, а для линейной сети его предложили Б.Уидроу (B.Widrow) и Т.Хофф (T.Hoff).

$$\delta w_{ij}(t) = x^k_j(d^k_i - y_i(t)).$$

Дельта-правило обучения однослойного персептрона предполагает минимизацию суммарной среднеквадратичной

НИИ Искусственный интеллект

ошибки нейронной сети, которая для $p \in \boldsymbol{N}$ входных образов определяется соотношением:

$$E_s = \sum_{k=1}^{p} E(k) = \frac{1}{2} \sum_{k=1}^{p} \sum_{j=1}^{m} (y_j^k - d_j^k)^2$$

где $E(k)$– среднеквадратичная ошибка нейронной сети для k-го образа;

y_j^k и d_j^k – выходное и эталонное значение сети для k-го образа, соответственно.

Для минимизации суммарной квадратичной ошибки используется *метод градиентного спуска*. Существует два основных подхода к обучению однослойного персептрона: *последовательное обучение* (*online learning*) и *групповое обучение* (*batch learning*). При последовательном обучении модификация синаптических связей происходит после подачи каждого образа из обучающей выборки на нейронную сеть. В этом случае в методе градиентного спуска используется квадратичная ошибка нейронной сети для одного входного образа:

$$E = \frac{1}{2} \sum_{j=1}^{m} (y_j - d_j)^2$$

Тогда в соответствии с методом градиентного спуска, веса синаптических связей и пороги нейронной сети необходимо изменять с течением времени по следующим выражениям:

$$w_{ij}(t+1) = w_{ij}(t) - \alpha \frac{dE}{d\, w_{ij}(t)} ,$$

$$w_{0j}(t+1) = w_{0j}(t) - \alpha \frac{dE}{d\, w_{0j}(t)} ,$$

где α – скорость или шаг обучения.

Отсюда можно получить дельта-правило для последовательного обучения:

$$w_{ij}(t+1) = w_{ij}(t) - \alpha (y_j - e_j) x_i ,$$

НИИ Искусственный интеллект

$$w_{0j}(t+1)=w_{0j}(t)-\alpha(y_j\text{-}e_j)x_i \ .$$

Величина адаптивного шага при последовательном обучении равна:

$$\alpha(t)=\frac{1}{1+\sum_{i=1}^{n}x^2},$$

где i=1,2,…,n; n – число входов нейронной сети.

При использовании группового обучения модификация синаптических связей будет происходить после подачи на вход сети p образов. В этом случае дельта-правило можно представить следующим образом:

$$w_{ij}(p)=w_{ij}(0)\text{-}\alpha(t)\sum_{k=1}^{p}(y_j^k\text{-}e_j^k)x_i^k,\ w_{0j}(p)=w_{0j}(0)\text{-}\alpha(t)\sum_{k=1}^{p}(y_j^k\text{-}e_j^k).$$

Величина адаптивного шага при групповом обучении:

$$\alpha_j^k(t)=\sum_l(y_j^l-e_j^l)(1+\sum_i x_i^l x_i^k),\ \alpha(t)=\sum_k\sum_l(y_j^k-e_j^k)a_j^k/\sum_k\sum_l(a_j^k)^2$$

Теория персептронов стала классической основой для изучения многих типов искусственных нейронных сетей прямого распространения.

Однако персептроны не способны обучиться решению ряда простых задач. Используя точные математические методы можно строго доказать, что могут выполнять и чему могут обучаться персептроны. Важно при этом уметь различать *выразимость* (*представимость*) и *обучаемость*. Понятие выразимости относится к способности персептрона моделировать определенную функцию. Обучаемость же требует наличия систематической процедуры настройки весов синаптических связи для реализации этой функции.

III.3.1. Задания:

1. Перечислите шаги обучения персептрона.
2. Назовите два основных подхода к обучению однослойного персептрона.

НИИ Искусственный интеллект

3. Приведите формулу для решения задачи минимизации целевой функции.

III.3.1. Вопросы:

1. Объяните важность обощения алгоритма обучения персептрона, называемое дельта-правилом?
2. Что достигается увеличением размерности выходного (вычислительного) слоя персептрона и включением в него нескольких нейронов?
3. Из каких шагов состоит алгоритм правила персептрона?

III.3.2. Обучение сигмоидального нейрона

В этом параграфе будут рассмотрены вопросы обучения сигмоидальных нейронов, показаны примеры, предложены задания, и сформулированы вопросы. При подготовке учебного материала данного параграфа были использованы следующие источники [1-9].

Для обучения сигмоидального нейрона используется парадигма "обучения с учителем" с применением градиентных методов, позволяющих минимизировать целевую функцию обучения $E(W_i)$. Здесь в отличие от персептрона, для поиска минимума целевой функции используются методы поисковой оптимизации 1 порядка, в которых целенаправленное изменение весовых коэффициентов w_{ij} осуществляется в направлении отрицательного градиента:

$$E(W_i)=\frac{1}{2}\sum_{k=1}^{r}(y_i^k-d_i^k)^2=\frac{1}{2}\sum_{k=1}^{r}(\sum_{i=0}^{m}w_{ij}^k x_i^k-d_i^k)^2\to\min_w, \qquad \text{(III.3.2)}$$

где $y_i^k=f(s_i^k)=f(\sum_{i=0}^{m}w_{ij}^k x_i^k))$ является сигмоидальной функцией, x – входной вектор, $x=[x_0, x_1, ..., x_n]^T$ со значением $x_0=1$ при наличии поляризации и $x_0=0$ при ее отсутствии, x_i – соответствующее ему значение на выходе i-того нейрона.

Применение непрерывной функции активации позволяет использовать при обучении градиентные методы. Проще всего

НИИ искусственный интеллект

реализовать метод наискорейшего спуска, в соответствии с которым уточнение вектора весов $w=[w_0,w_1,\ldots,w_n]^T$ проводится в направлении отрицательного градиента целевой функции. Если эта функция определена выражением (III.3.2), *j*-я составляющая градиента:

$$\nabla_j E(w_i)=\frac{dE(w_i)}{d\,w_{ij}}=\sum_{k=1}^{r}\left(y_i^k-d_i^k\right)\cdot\frac{d\,y_i^k}{d\,w_{ij}}=\sum_{k=1}^{r}\left(y_i^k-d_i^k\right)\cdot\frac{d\,f(s_i^k)}{d\,s_i^k}x_j^k\ ,$$

где уточнение весов w_{ij} по методу наискорейшего спуска проводится в направлении отрицательного градиента целевой функции.

Обозначим $\delta_i^k=\left(y_i^k-d_i^k\right)\cdot\frac{df(s_i^k)}{d\,s_i^k}$, имеем $\frac{dE(W_i)}{d\,w_{ij}}=\sum_{k=1}^{r}\delta_i^k\cdot x_j^k.$

Также возможно обучение сигмоидального нейрона и дискретным способом – сериями циклов уточнения входных весов для каждой эталонной пары $<X^k,d_i^k>$ (см. правило персептрона). При этом коррекция весов после каждого цикла выполняется по следующей формуле:

$$w_{ij}(t+1)=w_{ij}(t)-\eta\,\delta_i^k(t)\,x_j^k,$$

где η – коэффициент обучения, значения которого выбирают эмпирически из интервала (0, 1).

Изменение весов можно осуществлять и решением следующего разностного уравнения:

$$\frac{dw_{ij}(t)}{dt}=-\mu\delta_i x_j,$$

где $\mu\in(0, 1)$ - коэффициент обучения.

Надо напомнить, что все методы поисковой оптимизации 1-го порядка – это методы локального поиска, не гарантирующие достижения глобального экстремума. Для преодоления этого недостатка было предложено обучение с моментом, в котором коррекция весов выполняется следующим образом:

НИИ искусственный интеллект

$$w_{ij}(t+1)=w_{ij}(t)-\eta\cdot\delta_i^k(t)\cdot x_j^k+\alpha\cdot(w_{ij}(t)-w_{ij}(t)).$$

Последнее слагаемое в формуле называется моментом и характеризует фактическое изменение веса в предыдущем цикле (α выбирается в диапазоне (0, 1)). Существует надежда, что при приближении к точке локального минимума (где градиентная составляющая $\delta_i^k(t)\cdot x_j^k$ стремится к нулю) составляющая момента выведет поиск из области локального минимума в более перспективную область.

Следует отметить, что применение градиентных методов обучения нейрона гарантирует достижение только локального экстремума, который для полимодальной $E(w)$ может быть достаточно далек от глобального минимума. Выход из его окрестности при использовании простого алгоритма наискорейшего спуска невозможен, поэтому на практике применяют обучение с разбросом или моментом.

Здесь процесс уточнения весов определяется не только информацией о градиенте функции, но также и фактическим трендом изменения весов:

$$\Delta w_{ij}(t+1)=-\eta\delta_i x_j+\alpha\Delta w_{ij}(t)$$

где $\alpha \in (0, 1)$ – коэффициент момента, или использование стохастических методов оптимизации.

Влияние момента на подбор весов увеличивается с ростом значения α, поэтому следует выбирать α таким, чтобы момент не доминировал в процессе обучения, так как это приводит к нестабильности (расходимости) алгоритма обучения.

II.3.2. Задания:

1. Напишите формулу униполярной функции активации.
2. Напишите формулу биполярной функции активации.
3. Напишите формулу вычисления производной униполярной функции активации.

НИУ «БелГУ» Электронный архив открытого доступа

4. Напишите формулу вычисления производной биполярной функции активации.

III.3.2. Вопросы:

1. Как выглядит график униполярной функции активации?
2. Как выглядит график производной униполярной функции активации?
3. Как выглядит график биполярной функции активации?
4. Как выглядит график производной биполярной функции активации?

НИИ Искусственный интеллект

ПРОЛОГ

На сегодняшний день создано большое количество моделей отдельных нейронов и моделей искусственных нейронных сетей. В данной работе была предпринята попытка обобщить накопленные знания, касающиеся отдельных искусственных нейронов, и, исходя из этой цели, для существующих моделей нейронов были приведены различные классификации: по положению нейронов в топологии сети, в зависимости от аппаратной реализации нейронов, по степени разработки математического описания модели нейрона и качественного описания его поведения. По последнему критерию нейроны разделены на формальные (для которых в свою очередь также приводится классификация по нескольким критериям), физиологические и феноменологические.

В результате обзора существующих моделей становится очвидно, что существует достаточное количество нейронов, описывающих естественный нейрон очень упрощенно. Они нашли свое применение в области распознавания образов, для решения задач классификации и т.д. Также существует множество моделей, которые при описании нейрона ставят своей целью количественное описание поведения нейрона. Однако до сих пор остается непонятным, приводит ли усложнение нейрона посредстовм попыток отображения им все новых свойств естественного нейрона, к существенному прогрессу и улучшению результатов решаемых нейронами задач. Данный процесс будет продолжаться постоянно.

НИИ Искусственный интеллект

ЛИТЕРАТУРНЫЕ ИСТОЧНИКИ И ИНТЕРНЕТ РЕСУРСЫ

1. Arbib M., ed. The Handbook of Brain Theory and Neural Networks. MIT Press, 1995.

2. Bishop C. M. Neural Networks for Pattern Recognition. Oxford University Press Inc., 2003.

3. Grossberg S. 1969. Some networks that can learn, remember and reproduce any number of complicated space-time patterns. Journal of Mathematics and Mechanics, 19:53-91.

4. Haykin S. Neural Networks: A Comprehensive Foundation. NY: Macmillan, 1994.

5. Hecht-Nielsen R. 1988. Applications of Counterpropagation networks. Newral Networks 1: 131-39.

6. Khanna T. Foundations of neural networks, Addison-Wesley Publishing Co., 1990.

7. Kohonen T. 1988. Self-organization and associative memory. 2d ed. New-York, Springer-Verlag.

8. KohonenT., "Self-Organizing Maps"(2-nd edition), Springer, 1997.

9. Kröse B., Smagt P. An introduction to Neural Networks. University of Amsterdam, 1996.

10. Lipman R. An introdaction to computing with neural nets // IEEE Acoustic, Speech and Signal Processing Magazine,1987, № 2, L.4-22.

11. Smith L. An Introduction to Neural Networks. Unpublished draft, University of Stirling, 2001. http://www.cs.stir.ac.uk/~lss/NNIntro/InvSlides.html.

12. Zurada J.M. Introduction To Artificial Neural Systems. Boston: PWS Publishing Company, 1992.

13. Головко В. А. Нейронные сети: обучение, организация и применение. Учеб. пособие для вузов / Общая ред. А.И. Галушкина. – М.: ИПРЖР, 2001. – 256 с.

14. Голубев Ю. Ф. Нейросетевые методы в мехатронике. – М.: Изд-во Моск. унта, 2007. – 157 с. – ISBN 978-5-211-05434-9.

НИИ искусственный интеллект

15. Горбань А. Н., Дунин-Барковский В. Л. и др. Нейроинформатика. – Новосибирск: Наука, 1998.

16. Комарцова Л.Г., Максимов А.В. Нейрокомпьютеры: учебное пособие для вузов. – 2-е изд., перераб. И доп. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2004. – 400 с.

17. Короткий С. Нейронные сети: обучение без учителя. http://www.orc.ru/~stasson/n3.zip

18. Короткий С. Нейронные сети: основные положения. http://www.orc.ru/~stasson/nl .zip

19. Кошимский В.И., Смирнов Д.А, Нейронные сети и их применение в системах управления и связи. – М.: Горячая линия – Телеком, 2002.

20. Круглов В.В., Борисов В.В. Искусственные нейронные сети. Теория и практика. – 2-е изд., стереотип. – М.: Горячая линия-Телеком, 2002.

21. Минский М. , Пейперт С. Персептроны. Москва: Мир, 1971.

22. Мкртчян С.О.. Нейроны и нейронные сети. Москва: Энергия, 1971.

23. Нейронные сети. Алгоритм обратного распространения [электронный ресурс]: http://ai.obrazec.ru/neur-2.html

24. Розенблатт Ф. Принципы нейродинамики: Персептрон и теория механизмов мозга: Пер. с англ. – М.: Мир. – 1965. – 480 с.

25. Стариков А. Введение в RBF сети, http://www.basegroup.ru /neural/rbf.htm

26. Суровцев И.С., Клюкин В.И., Пивоварова Р.П. Нейронные сети.- Воронеж: ВГУ, 1994. 225 с.

27. Тарков М.С. Нейрокомпьютерные системы - Интернет-университет информационных технологий - ИНТУИТ.ру, БИНОМ. Лаборатория знаний, 2006. -144с.

28. Терехов С.А. Лекции по теории и приложениям искусственных нейронных сетей. Лаборатотория Искусственных Нейронных Сетей НТО-2, ВНИИТФ. Снежинск, 1998.

НИИ искусственный интеллект

29. Уоссермен Ф. Нейрокомпьютерная техника: Теория и практика. М.: Мир, 1992. 240 с.

30. Хайкин С. Нейронные сети: полный курс, 2-е издание, исправленное.: Пер. с англ. – М.: ООО «И.Д. Вильямс», 2006. – 1104.

31. How many kinds of Kohonen networks exist? Internet FAQ Archives. Online Education

32. Hecht-Nielsen, R. (1990), Neurocomputing, Reading, MA: Addison-Wesley, ISBN 0-201-09355-3.

33. Kohonen, T. (1989/1997/2001), Self-Organizing Maps, Berlin – New York: Springer-Verlag. First edition 1989, second edition 1997, third extended edition 2001, ISBN 0-387-51387-6, ISBN 3-540-67921-9

34. Kohonen, T. (1988), Learning Vector Quantization, Neural Networks, 1 (suppl 1), 303.

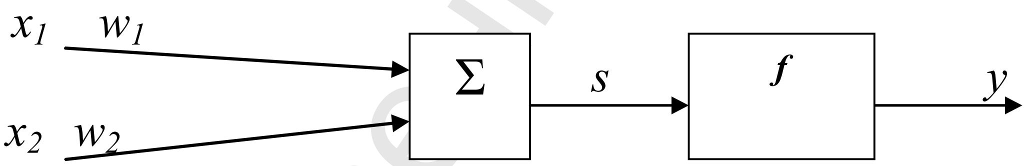
НИИ Искусственный интеллект

ПРИЛОЖЕНИЕ 1. ВЫРАЗИТЕЛЬННОСТЬ ПЕРСЕПТРОНА

В этом приложении будет обсуждаться решение проблемы реализуемости функции «Исключающее ИЛИ (XOR)» с помощью однослойного персептрона двумя способами: в первом способе доказывается её невозможность, а во втором наоборот – доказывается реализумость.

Интерес к однослойным персептронам исчез, когда М. Минский и С. Пейперт доказали, что персептроны теоретически не способны решить многие простые задачи, в том числе реализовать бинарную функцию «Исключающее ИЛИ – eXclusive OR (XOR)», которая имеет два аргумента, значением каждого из которых может быть 0 или 1 и принимает значение 1, когда только один из аргументов равен 1, и значение 0, в противном случае.

Невозможность реализации функии «Исключающее ИЛИ» можно показать с помощью однонейронного персептрона с двумя входными сигналами x_1 и x_2, представленного на рисунке П.1.1.



Здесь функция активации *f* является обычным порогом, так что выход *y* принимает значение единица, когда ***s*** больше или равно 0,5 и, ноль, когда ***s*** меньше 0,5, т.е.:

$$y = f(s) = \begin{cases} 1, & \text{если} \quad s \geq 0{,}5 \\ 0, & \text{если} \quad s < 0{,}5 \end{cases}$$

На рисунке П.1.2 показано, что все возможные комбинации входных сигналов x_1 и x_2 состоят из четырех точек $\boldsymbol{A}_0(0,0)$, $\boldsymbol{B}_0(1,0)$, $\boldsymbol{B}_1(0,1)$, $\boldsymbol{A}_1(1,1)$ на плоскости (x_1, x_2), причем входные комбинации, которые должны давать нулевой выход, помечены $\boldsymbol{A_0}$ и $\boldsymbol{A_1}$, единичный выход – $\boldsymbol{B_0}$ и $\boldsymbol{B_1}$.

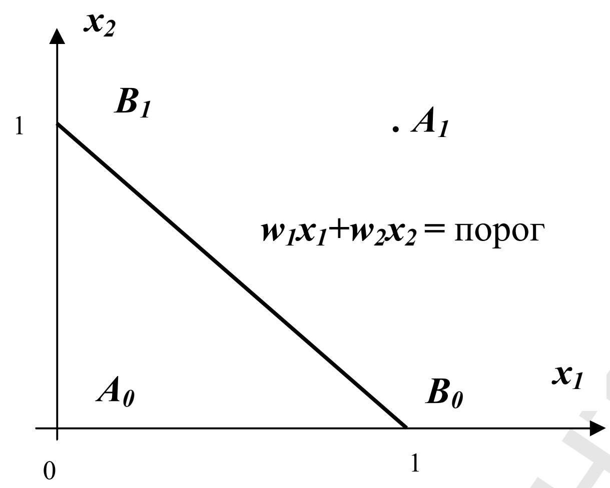


Таблица П.1 показывает семантическую модель (требуемую связь между входами и выходом) для функции XOR.

Таблица П.1. Модель функции XOR.

Точки	x_1	x_2	**Требуемый выход**
A_0	0	0	0
B_0	1	0	1
B_1	0	1	1
A_1	1	1	0

Нейрон выполняет следующее вычисление $w_1x_1+w_2x_2=s$. Здесь никакая комбинация значений двух весов w_1 и w_2 не может дать соотношения между входом и выходом, задаваемого в таблице П.1. Чтобы понять это ограничение, зафиксируем s на величине порога 0,5. В этом случае мы имеем уравнение $w_1x_1+w_2x_2=0{,}5$. Это уравнение линейно по x_1 и x_2, т. е. все значения по x_1 и x_2, удовлетворяющие этому уравнению, будут лежать на некоторой прямой в плоскости (x_1, x_2). Любые значения для x_1 и x_2 на этой линии будут давать пороговое значение 0,5 для s. Входные значения с одной стороны прямой обеспечат значения s больше порога, следовательно, y = 1. Входные значения по другую сторону

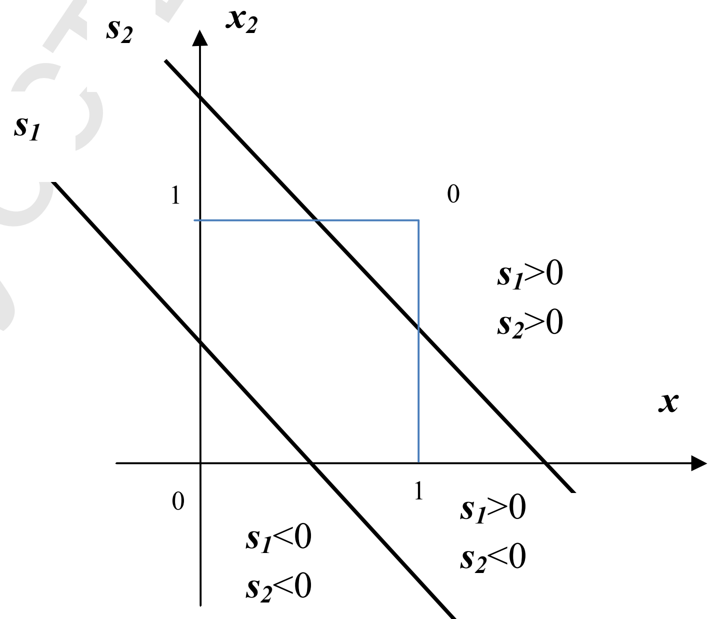
НИИ Искусственный интеллект

прямой обеспечат значения s меньше порогового значения, делая $y = 0$. Изменения значений w_1, w_2 и порога будут менять наклон и положение прямой.

Для того, чтобы персептрон реализовал функцию «Исключающее ИЛИ», заданную таблице П.1, нужно расположить прямую так, чтобы точки A_0 и A_1 были с одной стороны прямой, а точки B_0 и B_1 – с другой. Легко заметить, что это невозможно. Это означает, что какие бы значения ни приписывались весам и порогу, персептрон неспособен воспроизвести соотношение между входом и выходом, требуемое для представления логической функции «Исключающее ИЛИ».

Однако, это касается только персептрона с пороговой или монотонно возрастающей непрерывной сигмоидальной функцией активации. При использовании сигнальной или унимодальной функции активации (Гаусса) с одним максимумом однослойный персептрон способен решить проблему «Исключающее ИЛИ».

Возможность реализации функции «Исключающее ИЛИ» можно показать с помощью персептрона с сигнальной функцией активации. В этом случае, необходимо выделить область единиц или нулей при помощи двух прямых, как на рисунке П.1.3.



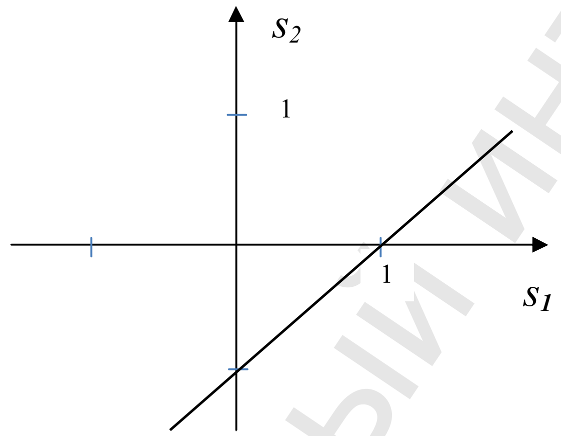
НИИ Искусственный интеллект

Область ***A***, которая определяет класс единиц, характеризуется условиями $s_1 > 0$ и $s_2 < 0$. Уравнения прямых имеют вид:

$$s_1 = x_2 + x_1 - 0.5,$$

$$s_2 = x_2 + x_1 - 1.5.$$

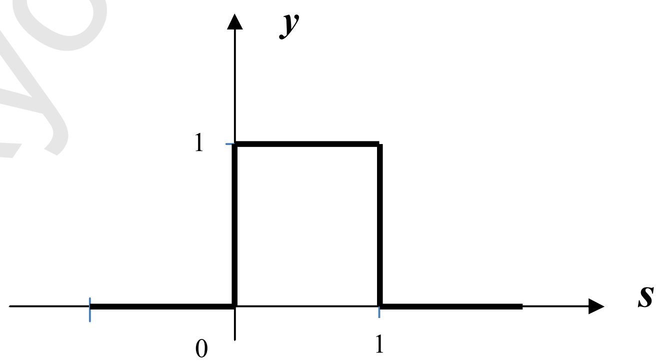
Отсюда можно получить зависимость $s_2 = s_1 - 1,$ которую можно представить на рисунке П.1.4.



Тогда, если $s_1 > 0$ и $s_2 < 0$ $\Rightarrow$ $0 < s_1 > 1$ и можно получить следующую функцию активации

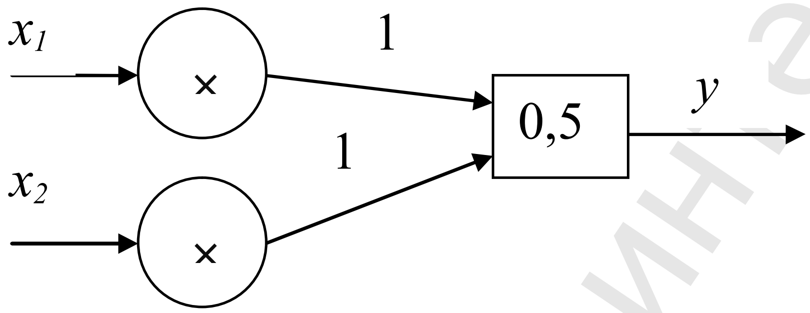
$$y = f(s) = f(s_1) = \begin{cases} 1 \ , & \text{если} \quad 0 < s < 1, \\ 0 \ , & \text{в противном} \quad \text{случае} \end{cases}$$

График этой сигнальной функции представлен на рисунке 5.



НИИ Искусственный интеллект

В результате однослойный персептрон для решения задачи «исключающее ИЛИ» будет иметь веса и порог, равные 1 и 0,5 соответственно. Структурная схема однослойного персептрона для реализации функции XOR, представлена на рисунке П.1.6.



Таким образом, при использовании сигнальной функции активации однослойный персептрон может решить задачу «исключающее ИЛИ», так как она разбивает входное пространство образов на классы при помощи двух прямых.

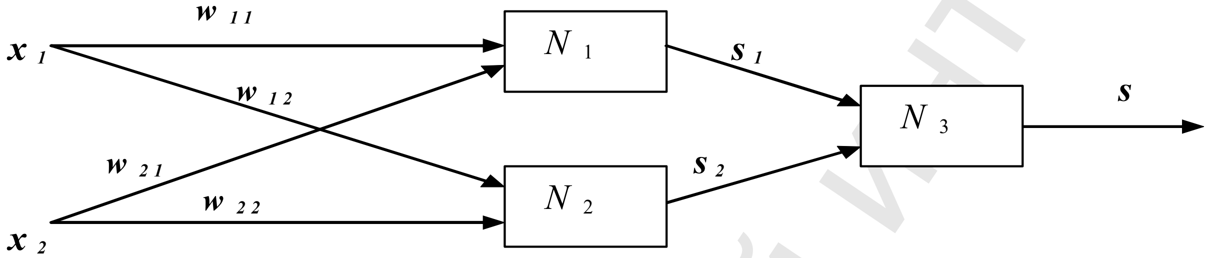
При использовании однослойного персептрона с радиально базисной функцией активации для решения данной задачи необходимо принимать выходное значение персептрона равным 1, если оно больше определённого значения.

НИИ искусственный интеллект

ПРИЛОЖЕНИЕ 2. ПРИМЕРЫ ПЕРСЕПТРОНА

В данном Приложении рассмотрены примеры двухслойного и трехслойного персептрона.

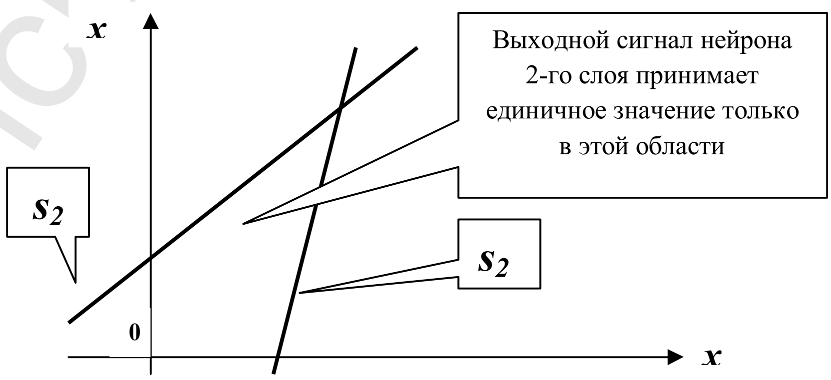
Двухслойный персептрон с двумя входами, подведенными к двум нейронам первого слоя, соединенными с единственным нейроном в втором слое. Структурная схема этого персептрона представлена на рисунке П.2.1.



Пусть для выходного нейрона порог равен 0,75, а оба его веса равны 0,5. Тогда для того, чтобы порог был превышен и на выходе появилась единица, требуется, чтобы оба нейрона первого уровня на выходе имели единицу, т.е.

$$s = 0{,}5*s_1 + 0{,}5*s_2 = 0{,}5*1 + 0{,}5*1 = 1$$

Таким образом, выходной нейрон реализует логическую функцию AND. На рисунке П.2.2 каждый нейрон слоя 1 разбивает плоскость (x_1, x_2) на две полуплоскости, один обеспечивает единичный выход для входов ниже верхней линии, другой – для входов выше нижней линии. Выходной сигнал нейрона второго слоя равен единице только внутри V-образной области.



НИИ Искусственный интеллект

Аналогично в первом слое может быть использовано три нейрона с дальнейшим разбиением плоскости и созданием области треугольной формы. Включением достаточного числа нейронов во входной слой может быть образован выпуклый многоугольник любой желаемой формы. Так как они образованы с помощью операции *AND* над областями, задаваемыми линиями, то все такие многогранники выпуклы, следовательно, только выпуклые области и возникают. Точки, не составляющие выпуклой области, не могут быть отделены от других точек плоскости двухслойной сетью.

Нейрон второго слоя не ограничен функцией *AND*. Он может реализовывать многие другие функции при подходящем выборе весов и порога. Например, можно сделать так, чтобы единичный выход любого из нейронов первого слоя приводил к появлению единицы на выходе нейрона второго слоя, реализовав тем самым логическое *OR*. Имеется 16 двоичных функций от двух переменных. Если выбирать подходящим образом веса и порог, то можно воспроизвести 14 из них (все, кроме, *XOR* и *NOTXOR*)

Математическая модель двухслойного персептрона представляется следующим выражением:

$$y_j = f_2\left(w_{20j} + \sum_{j=1}^{J} w_{2j} f_1\left(w_{10i} + \sum_{i=1}^{I} w_{1i} x_i\right)\right),$$

где f_1, f_2 – функции активации первого и второго слоев, I – число нейронов первого слоя, J – число нейронов второго слоя, w_{0i}^1 – начальное состояние i-го нейрона первого слоя, w_{0i}^2 – начальное состояние j-го нейрона второго слоя, w_{1i} и w_{2j} – весовые коэффициенты i-го нейрона первого слоя и j-го нейрона второго слоя, x_i – i-ая координата входного вектора, y_j – j-ая координата выходного вектора.

НИИ искусственный интеллект

Трехслойный персептрон является более общей нейронной сетью. Ее классифицирующие возможности ограничены лишь числом искусственных нейронов и весов. Ограничения на выпуклость отсутствуют. Теперь нейрон третьего слоя принимает в качестве входа набор выпуклых многоугольников, и их логическая комбинация может быть невыпуклой. На рисунке П.2.3 иллюстрируется случай, когда два треугольника *A* и *B*, скомбинированные с помощью функций «*A* и не *B*», задают невыпуклую область. При добавлении нейронов и весов число сторон многоугольников может неограниченно возрастать. Это позволяет аппроксимировать область любой формы с любой точностью. Вдобавок, необязательно, чтобы все выходные области второго слоя должны пересекаться. Возможно, следовательно, объединять различные области, выпуклые и невыпуклые, выдавая на выходе единицу всякий раз, когда входной вектор принадлежит одной из них.

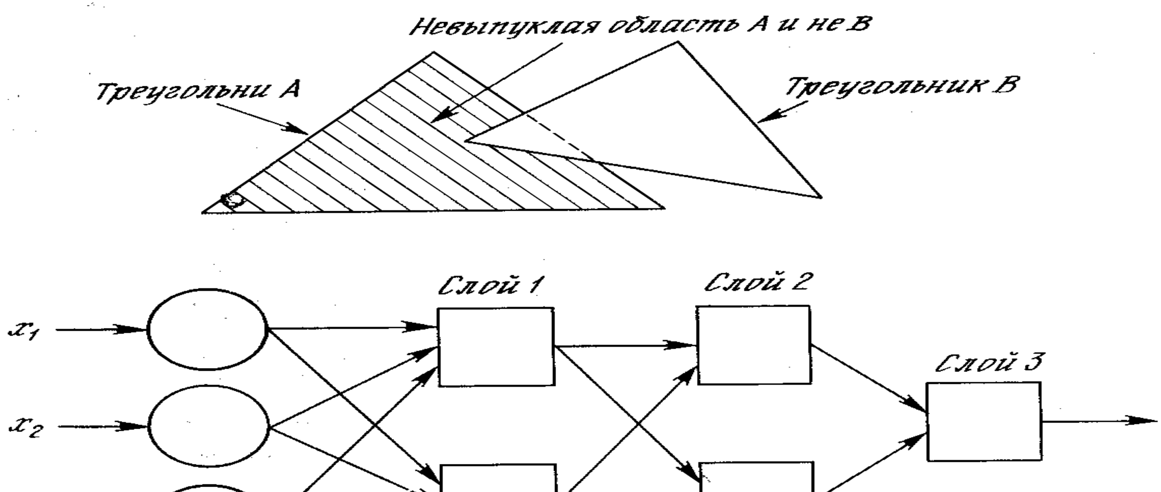


Рисунок П.2.3. Вогнутая область решений, задаваемая 3-х слойной сетью

В зависимости от значения выходного сигнала принимается решение: +1, если входной сигнал принадлежит классу *A*; -1, если входной сигнал принадлежит классу *B*.

Математическая модель трехслойного персептрона представляется следующим выражением:

$$y_j = f_3(w_{30k} + \sum_{k=1}^{K} w_{3k} f_2(w_{20j} + \sum_{j=1}^{J} w_{2j} f_1(w_{10i} + \sum_{i=1}^{I} w_{1i} x_i)),$$

где $w_{10\,i}$, $w_{1\,i}$, $w_{20\,j}$, $w_{2\,j}$, w_{30k} , w_{3k} – начальные состояния и весовые коэффициенты i-го нейрона первого слоя, j-го нейрона второго слоя, k-го нейрона третьего слоев соответственно; I – число нейронов первого слоя, J – число нейронов второго слоя, K – число нейронов третьего слоя.

НИИ Искусственный интеллект

ПРИЛОЖЕНИЕ 3. ПРИМЕР ОБУЧЕНИЯ ПЕРСЕПТРОНА

Задан набор точек s_1, принадлежащий первому классу (выходной сигнал – 1) и набор точек s_2, принадлежащий второму классу (выходной сигнал – 0). Составить уравнение гиперплоскости, линейно разделяющее пространство признаков на две части, т.е. нужно найти вектор W и смещение w_0.

Задано:

s_1: (-2, 4); (0, 3); (0, 0);

s_2: (-2, 0); (-1, -5); (3, -6); (2,- 5).

Найти. Уравнение гиперплоскости, разделяющее пространство признаков на две части.

Решение:

2. Построим линейное разделяющее правило при помощи разделения центров масс.

Определим начальное значение вектора весов.

$$W=\left(\sum_{i=1}^{n} X_1^i-\sum_{j=1}^{m} X_2^j\right)/(n+m),$$

где X_1^i – вектора из множества первого класса, а X_2^j – вектора из множества второго класса.

$$W=\left(\sum_{i=1}^{3} X_1^i-\sum_{j=1}^{4} X_2^j\right)/(3+4)=$$

$$=\frac{(2,4)+(0,3)+(0,0)-(\text{-}2,0)-(\text{-}1,\text{-}5)-(3,\text{-}6)-(2,\text{-}5)}{7}=$$

$$=\frac{(2,7)-(2,\text{-}16)}{7}=\frac{(0,23)}{7}=(0,\quad 3{,}285)$$

Начальное значение вектора весов $w_0=0;\ w_1=0;\ w_2=3{,}285.$

3. Произведем обучение персептрона по отдельным примерам, для этого возьмем обучающую выборку и поочередно рассчитаем значение выходного сигнала

$$X_1^1=(2,4),\ s=1\cdot w_0+x_1w_1+x_2w_2=0+0+4\cdot 3{,}285=13{,}14\ ,$$

НИИ Искусственный интеллект

$$y = f(s) = \begin{cases} 1, & \text{если} \quad s > 0 \\ 0, & \text{если} \quad s \le 0 \end{cases}, \Rightarrow\ y = 1;$$

$X_1^2 = (0,3)$, , $s = 1 \cdot w_0 + x_1 w_1 + x_2 w_2 = 0 + 0 + 3 \cdot 3{,}285 = 9{,}857$

$y = f(s) = 1;$

$X_1^3 = (0,0)$, $s = 1 \cdot w_0 + x_1 w_1 + x_2 w_2 = 0 + 0 + 0 \cdot 3{,}285 = 0$,

$y = f(s) = 0;$

Производим корректировку весов по формуле

$w_i(t+1) = w_i(t) + \gamma x_i$, $\gamma = 0.8$ и получим:

$w_0(t+1) = w_0(t) + \gamma x_0 = 0 + 0{,}8 \cdot 1 = 0.8$;

$w_1(t+1) = w_1(t) + \gamma x_1 = 0 + 0{,}8 \cdot 0 = 0$;

$w_2(t+1) = w_2(t) + \gamma x_2 = 3{,}285 + 0{,}8 \cdot 0 = 3{,}285$.

Теперь $W = (0{,}8;\ 0;\ 3{,}285)$

$X_1^1 = (2,4)$, $s = 1 \cdot w_0 + x_1 w_1 + x_2 w_2 = 1 \cdot 0{,}8 + 2 \cdot 0 + 4 \cdot 3{,}285 = 13{,}942$

$y = f(s) = 1;$

$X_1^2 = (0,3)$, $s = 1 \cdot w_0 + x_1 w_1 + x_2 w_2 = 1 \cdot 0{,}8 + 0 \cdot 0 + 3 \cdot 3{,}285 = 10{,}65$,

$y = f(s) = 1;$

$X_1^3 = (0,0)$, $s = 1 \cdot w_0 + x_1 w_1 + x_2 w_2 = 1 \cdot 0{,}8 + 0 + 0 \cdot 3{,}285 = 0{,}08$

$y = f(s) = 1;$

$X_2^1 = (-2,0)$, $s = 1 \cdot w_0 + x_1 w_1 + x_2 w_2 = 1 \cdot 0{,}8 + (-2) \cdot 0 + 0 \cdot 3{,}285 = 0{,}8$,

$y = f(s) = 1;$

Для второй группы данных ответ должен соответствовать 0. Коррекция весов осуществляется по формуле

$w_i(t+1) = w_i(t) - \gamma x_i$, $\gamma = 0.8$ и получим:

$w_0(t+1) = w_0(t) - \gamma x_0 = 0{,}8 - 0{,}8 \cdot 1 = 0$;

$w_1(t+1) = w_1(t) - \gamma x_1 = 0 - 0{,}8 \cdot (-2) = 1{,}6$;

НИИ искусственный интеллект

$w_2(t+1)=w_2(t)-\gamma x_2=3{,}285-0{,}8\cdot 0=3{,}285$.

Теперь W = (0; 1,6; 3,285)

$X_1^1=(2,4)$, $s=1\cdot w_0+x_1w_1+x_2w_2=1\cdot 0+2\cdot 1{,}6+4\cdot 3{,}285=16{,}34$,

$y=f(s)=1;$

$X_1^2=(0,3)$, $s=1\cdot w_0+x_1w_1+x_2w_2=1\cdot 0+0\cdot 1{,}6+3\cdot 3{,}285=9{,}855$,

$y=f(s)=1;$

$X_1^3=(0,0)$, $s=1\cdot w_0+x_1w_1+x_2w_2=1\cdot 0+0\cdot 0+0\cdot 3{,}285=0$,

$y=f(s)=0;$

Производим корректировку весов по формуле

$w_i(t+1)=w_i(t)+\gamma x_i$, $\gamma=0.8$ и получим:

$w_0(t+1)=w_0(t)+\gamma x_0=0+0{,}8\cdot 1=0.8$;

$w_1(t+1)=w_1(t)+\gamma x_1=1{,}6+0{,}8\cdot 0=1{,}6$;

$w_2(t+1)=w_2(t)+\gamma x_2=3{,}285+0{,}8\cdot 0=3{,}285$.

Теперь W = (0,8; 1,6; 3,285)

$X_1^1=(2,4)$, $s=1\cdot w_0+x_1w_1+x_2w_2=1\cdot 0{,}8+2\cdot 1{,}6+4\cdot 3{,}285=17{,}14$,

$y=f(s)=1;$

$X_1^2=(0,3)$, $s=1\cdot w_0+x_1w_1+x_2w_2=1\cdot 0{,}8+0\cdot 1{,}16+3\cdot 3{,}285=10{,}655$,

$y=f(s)=1;$

$X_1^3=(0,0)$, $s=1\cdot w_0+x_1w_1+x_2w_2=1\cdot 0{,}8+0\cdot 1{,}6+0\cdot 3{,}285=0{,}8$

$y=f(s)=1;$

$X_2^1=(-2,0)$, $s=1\cdot w_0+x_1w_1+x_2w_2=1\cdot 0{,}8+(-2)\cdot 0+0\cdot 3{,}285=-2{,}4$,

$y=f(s)=0;$

$X_2^2=(-1,-5)$, $s=1\cdot w_0+x_1w_1+x_2w_2=1\cdot 0{,}8+(-1)\cdot 1{,}6+(-5)\cdot 3{,}285=-17{,}225$,

$y=f(s)=0;$

НИИ искусственный интеллект

$X_2^3=(3,\text{-}6),\ s=1\cdot w_0+x_1w_1+x_2w_2=1\cdot 0{,}8+3\cdot 1{,}6+(-6)\cdot 3{,}285=-14{,}11$,

$X_2^4=(3,\text{-}6),\ s=1\cdot w_0+x_1w_1+x_2w_2=1\cdot 0{,}8+2\cdot 1{,}6+(-5)\cdot 3{,}285=-12{,}425$,

$f(s)=0$.

Однослойный персептрон обучен. Ответ: *W*= (0,8; 1,6; 3,285)
Уравнение разделяющей гиперплоскости:

$$0{,}8+1{,}6\cdot x_1+3{,}285\cdot x_2=0 .$$

Разделяющая гиперплоскость изображена на рисунке П.3.

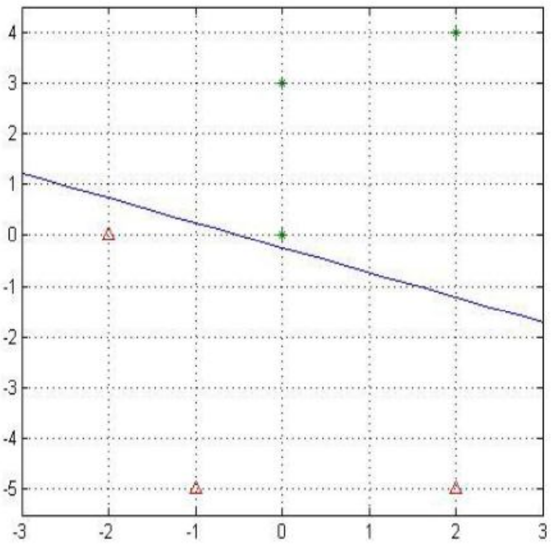


Рисунок П.3 - Разделяющая гиперплоскость

ПРИЛОЖЕНИЕ 4. ХАРАКТЕРИСТИКИ ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ

Алгоритмы обучения	Парадигмы обучения	Обучающие правила	Архитектуры сетей	Решаемые задачи
Алгоритмы обучения персептрона. Обратное распространение	С учителем	Коррекция ошибки	Однослойный и многослойный персептрон	Классификация образов, Аппроксимация функций, Предсказание, управление
Алгоритм обучения Больцмана		Больцман	Рекуррентная	Классификация образов
Линейный дискриминантный анализ		Хебб	Многослойная, прямого распространения	Анализ данных, Классификация образов
Векторное квантование		Соревнование	Соревнование	Категоризация внутри класса. Сжатие данных
ARTMap			Сеть ART	Классификация образов
Проекция Саммона	Без учителя	Коррекция	Многослойная	Категоризация внутри

НИИ Искусственный интеллект

		ошибки	прямого распространения
Анализ главных компонентов		Хебб	Прямого распространения и соревнование
Обучение ассоциативной памяти			Сеть Хопфилда
Векторное квантование		Соревнование	Соревнование
SOM Кохонена			SOM Кохонена
ART1, ART2			Сети ART
Алгоритм обучения RBF	Смешанная	Коррекция ошибки и соревнование	Сеть RBF

НИИ Искусственный интеллект

Таблица П.4.2 - Виды искусственных нейронных сетей.

Вид искусственной нейронной сети	**Решаемые задачи**	**Недостатки**	
Искусственный резонанс - 1, ART-1 Network (Adaptive Resonance Theory Network - 1)	Распознавание образов, кластерный анализ	Неограниченное увеличение числа нейронов в процессе функционирования сети. При наличии шума возникает значительный проблемы, связанные с неконтролируемым ростом числа эталонов.	Обуч
Двунаправленная ассоциативная память (ДАП). Bi-Directional Associative Memory (BAM)	Ассоциативная память, распознавание образов.	Емкость ДАП жестко ограниченна, также ДАП обладает некоторой непредсказуемостью в процессе функционирования. Возможны ложные ответы.	По с авто сеть двун памя стро вект общ разм возм гете испо шир чем Про сина прос Сеть проц
Машина Больцмана	Распознавание образов, классификация.	Медленный алгоритм обучения.	Алго выбі мин

НИИ Искусственный интеллект

			рель
Обратное распространение (Neural Network with Back Propagation Training Algorithm)	Распознавание образов, классификация, прогнозирование, синтез речи, контроль, адаптивное управление, построение экспертных систем.	Многокритериальная задача оптимизации в методе обратного распространения рассматривается как набор однокритериальных – на каждой итерации происходит изменения значений параметров сети, улучшающих работу лишь с одним примером обучающей выборки. Такой подход существенно уменьшает скорость обучения. Классический метод обратного распространения относиться к алгоритмам с линейной сходимостью. Для увеличения скорости сходимости необходимо использовать матрицы вторых производных функций ошибки.	Обра перв обуч нейр попу обуч реше мно зада
Сеть встречного распространения (Counter Propagation Network)	Распознавание и восстановление образов (ассоциативная память), сжатие данных (с потерями), статистический анализ.	Сеть не дает возможность строить точные аппроксимации. В этом она значительно уступает сетям с обратным распространением ошибки. Слабая теоретическая проработка модификаций этой сети.	Сеть расп дает стат из м Сеть обуч обра мож Сеть кото нача

НИИ Искусственный интеллект

			дает фун нахо реше
Delta Bar Delta сеть	Распознавание образов, классификация.	Стандартный алгоритм DBD не использует эвристики, основанные на моменте. Даже небольшое линейное увеличение коэффициентов может привести к значительному росту скорости обучения, что вызовет скачки в пространстве весов. Геометрическое уменьшение коэффициентов иногда оказывается недостаточно быстрым.	Пара попь конв обра счет допо изме врем
Сеть поиска максимума с прямыми связями (Feed-Forward MAXNET)	Совместно с сетью Хэмминга, в составе нейросетевых систем распознавания образов.	Число слоев сети растет с увеличением размерности входного сигнала.	В от цикл фун расс изве кото полу
Гауссов классификатор (Neural Gaussian Classifier)	Распознавание образов, классификация.	Примитивные разделяющие поверхности (гиперплоскости) дают возможность решать лишь самые простые задачи распознавания. Считаются априорно известными распределениями входных сигналов, соответствующих разным классам.	Про реал прос алго сина смеш

НИИ Искусственный интеллект

Генетический алгоритм (Genetic Algorithm)	Распознавание образов, классификация, прогнозирование.	Сложные для понимания и программной реализации.	ГА с поис адап ими обла пара Дост обуч скор град дают диск пара что у реал прив врем
Сеть Хэмминга (Hamming Net)	Распознавание образов, классификация, ассоциативная память, надежная передача сигналов в условиях помех	Сеть способна правильно распознавать (классифицировать) только слабо зашумленные входные сигналы. Возможность использования только бинарных входных сигналов существенно ограничивает область применения	Сеть и бы (реш резу всего нейр мно прох в сет функ мно нейр итер полу зара испо

НИИ Искусственный интеллект

			прос форм сина смец сети Хэм разм она нейр вход 100 обра 1000 Хэм буде
Сеть Хопфилда (Hopfield Network)	Ассоциативная память, адресуемая по содержанию; распознавание образов; задачи оптимизации (в том числе, комбинаторной оптимизации).	Сеть обладает небольшой емкостью. Кроме того, наряду с запомненными образами в сети хранятся и их 'негативы'. Размерность и тип входных сигналов совпадают с размерностью и типом выходных сигналов. Это существенно ограничивает применение сети в задачах распознавания образов. При использовании сильно коррелированных векторов–образцов возможно зацикливание сети в процессе функционирования. Квадратичный рост числа синапсов при увеличении	Позв иска

НИИ Искусственный интеллект

		размерности входного сигнала.	
Входная звезда (Instar)	Может быть использована в сетях распознавания образов.	Каждая звезда в отдельности реализует слишком простую функцию. Из таких звезд невозможно построить нейронную сеть, которая реализовала бы любое заданное отображение. Это ограничивает практическое применение входных звезд.	Вход моде ком нейр дост отде реше вход испо прос сете
Сеть Кохонена (Kohonen's Neural Network)	Кластерный анализ, распознавание образов, классификации.	Сеть может быть использована для кластерного анализа только в случае, если заранее известно число кластеров.	В от Кохо фун поме фико моди наст посл наст непр
Сеть поиска максимума (MAXNET)	Совместно с сетью Хэмминга, в составе нейросетевых систем распознавания образов.	Заранее неизвестно число итераций функционирования нейронной сети MAXNET. Она определяет, какой из входных сигналов имеет максимальное значение. Однако, в процессе функционирования сеть не сохраняет само значение максимального сигнала. Квадратичный рост числа синапсов при увеличении	Про

НИИ Искусственный интеллект

		размерности входного сигнала.	
Выходная звезда (Outstar)	Может быть использован как компонент нейронных сетей для распознавания образов.	Каждая звезда в отдельности реализует слишком простую функцию. Вычислительные возможности нейронных сетей, составленных из таких звезд, ограничены.	При зада быть пост обуч
Сеть радиального основания (Radial Basis Function Network)	Распознавание образов, классификация.	Заранее должно быть известно число эталонов, эвристики для построения функций активации нейронов скрытого слоя. Сети этого типа довольно компактны и быстро обучаются. Радиально базисная сеть обладает такими особенностями: один скрытый слой, только нейроны скрытого слоя имеют нелинейную активационную функцию и веса синаптических связей входного и скрытого слоев равны единице.	Отсу прин
Нейронные сети, имитирующие отжиг (Neural Networks with Simulated Annealing Training Algorithm)	С помощью алгоритма имитации отжига можно строить отображения векторов различной размерности. К ним сводятся многие задачи распознавания образов, адаптивного	Низкая скорость сходимости при обучении нейронных сетей большой размерности.	"Теп зало возм мин алго мож поис адап сети

НИИ Искусственный интеллект

	управления, многопараметрической идентификации, прогнозирования и диагностики.		
Однослойный персептрон (Single Layer Perceptron)	Распознавание образов, классификация.	Простые разделяющие поверхности (гиперплоскости) дают возможность решать лишь несложные задачи распознавания.	Про реал прос алго

НИИ Искусственный интеллект

ШАРИПБАЙ А.А.

НЕЙРОННЫЕ СЕТИ

Учебное пособие

Бумага офсетная Формат 60x100 1/16
Плотность 80гр/м^2. Белизна 95%. Печать РИЗО.
Усл.печ.стр. 17.3. Объем 278 стр.

ЭВЕРО

Подготовлено к изданию и отпечатано
в издательстве «Эверо»
РК, Алматы, ул. Байтурсынова, 22
тел.: 8 (727) 233 83 89, 233 83 43,
233 80 45, 233 80 42
e-mail: evero08@mail.ru

ШАРИПБАЙ Алтынбек Амирулы, доктор технических наук, профессор, академик Международной академии информатизации, академик Академии педагогических наук РК, лауреат государственной премии РК, директор НИИ «Искусственный интеллект» ЕНУ им.Л.Н.Гумилева (www.e-zerde.kz). Его научными интересами являются теоретические и прикладные проблемы информатики и информационных технологий: теория и технология программирования, автоматизированные системы, искусственный интеллект, компьютерная лингвистика и электронное обучение. В этой области им были разработаны методы формализации семантики продукционных и логических языков программирования, методы автоматизированной верификации программных и аппаратных средств. По результатам этих исследований он защитил докторскую диссертацию на тему «Верификация программных и аппаратных средств вычислительных машин и систем» по специальности «05.13.13 (05.13.11) - Вычислительные машины, системы и сети (Математическое, программное и техническое обеспечение)». Отдельные его научные результаты были внедрены в крупные научные центры: 1976-1979 годы «Транслятор с языка имитации космических систем» в Летно-испытательском институте, г. Жуковск; 1988-1989 годы «Система верификации цифровых схем» в Научно-производственном центре, г. Зеленоград; 1990-1991 годы «Система параллельного программирования для многопроцессорного вычислительного комплекса» в Научно-исследовательском центре электронной вычислительной техники, г. Москва.

Он опубликовал более 300 научных трудов, издал 5 учебников и более 10 учебных пособий, 2 монографии и 5 терминологических и толковых словарей по информатике и вычислительной технике, получил более 30 свидетельств о государственной регистрации интеллектуальной собственности, участвовал в разработке многих государственных стандартов: 10 - в области информационных технологий, 9 - в области образования.

Под научным руководством А.Шарипбай подготовлены 5 докторов и 8 кандидатов наук, 4 доктора PhD по группе специальностей «Информатика, вычислительная техника и управления». Его научной школой создана математическая теория казахского языка, разработаны методы автоматизированного анализа и синтеза устных и письменных слов и предложений казахского языка, предложена технология создания электронных учебных изданий и др. Эти научные результаты в 1994-2014 годы были применены для создания многих электронных учебных изданий, системы дистанционного обучения казахскому языку, системы распознавания и синтеза казахской речи и других автоматизированных систем, в том числе учетных и экспертных систем, внедренных в различных государственных и негосударственных структурах и организациях РК.