

**Министерство образования, науки, молодежи и спорта Украины**  
**Донбасская государственная машиностроительная академия**

**В. Б. Гитис**

**ТЕОРИЯ И ПРАКТИКА ПРИМЕНЕНИЯ  
НЕЙРОННЫХ СЕТЕЙ**

**Учебное пособие**

Утверждено  
на заседании ученого совета  
Протокол №     от

**Краматорськ 2012**

**УДК 004.032.26**

**ББК 32.818**

**Г 51**

**Рецензенти:**

**Гузь Н. Г.**, д-р экон. наук, профессор, Донецкий национальный университет;

**Рамазанов С. К.**, д-р техн. наук, профессор, Восточноукраинский университет им. В. Даля

Навчальний посібник містить основні поняття теорії нейронних мереж, а також включає приклади використання нейромережних технологій для вирішення задач, що важко формалізуються та виникають при моделюванні складних технічних і економічних систем. Рекомендується студентам спеціальності «Системи і методи прийняття рішень» як додаткова довідкова література при вивченні дисциплін, присвячених нейромережним технологіями, а також в процесі дипломного проектування.

**Гітіс, В. Б.**

**Г 51** Теорія і практика застосування нейронних мереж : навчальний посібник / В. Б. Гітіс. – Краматорськ: ДДМА, 2011. – 96 с.

ISBN

Учебное пособие содержит основные понятия теории нейронных сетей, а также включает примеры использования нейросетевых технологий для решения трудно формализуемых задач, возникающих при моделировании сложных технических и экономических систем. Рекомендуется студентам специальности «Системы и методы принятия решений» в качестве дополнительной справочной литературы при изучении дисциплин, посвященных нейросетевым технологиями, а также в процессе выполнения дипломного проекта.

**УДК 004.032.26**

**ББК 32.818**

ISBN

© В. Б. Гітіс, 2011

© ДГМА, 2011

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	5
1 БИОЛОГИЧЕСКИЕ ОСНОВЫ НЕЙРОННЫХ СЕТЕЙ.....	8
1.1 Свойства биологических нейронных сетей .....	8
1.2 Биологический нейрон .....	9
1.4 Нервный импульс .....	12
2 ОСНОВЫ ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ .....	14
2.1 Искусственный нейрон .....	14
2.2 Функции активации .....	15
2.3 Свойства искусственных нейронных сетей .....	20
2.4 Применение нейронных сетей в задачах распознавания образов ....	24
2.5 Принципы обучения нейронных сетей.....	26
3 ПЕРСЕПТРОНЫ .....	27
3.1 Однослойный персептрон.....	27
3.2 Многослойный персептрон.....	29
3.3 Построение многослойных персептронов.....	32
3.3.1 Этапы построения многослойных персептронов.....	32
3.3.2 Предварительная обработка исходных данных .....	33
3.3.3 Выбор архитектуры нейронной сети.....	34
3.3.4 Аналитическое определение преобразования, осуществляемого нейронной сетью.....	40
3.3.5 Формирование функционала оптимизации .....	41
3.3.6 Выбор метода настройки нейросети.....	43
3.3.7 Выбор начальных условий при настройке нейронной сети....	43
3.3.8 Интерпретация полученных результатов.....	44
3.3.9 Тестирование сети .....	44
3.3.10 Определение показателей значимости сигналов и элементов нейронной сети.....	44
3.3.11 Алгоритмы построения многослойных прямо направленных нейронных сетей.....	46
4 МЕТОДЫ ОБУЧЕНИЯ НЕЙРОННЫХ СЕТЕЙ.....	52
4.1 Метод обучения Хэбба.....	52
4.2 Метод обучения Уидроу-Хоффа (дельта-правило).....	53
4.3 Алгоритм обратного распространения ошибки .....	54
4.4 Модификации алгоритма обратного распространения ошибки .....	56
4.4.1 Пакетное обратное распространение.....	56
4.4.2 Фрагментарное обратное распространение .....	57
4.4.3 Обратное распространение с угасанием весов .....	57
4.4.4 Быстрое обратное распространение .....	57
4.4.5 Алгоритм Дельта-Дельта с чертой.....	58
4.4.6 Расширенная Дельта-Дельта с чертой.....	59
4.4.7 Эластичное обратное распространение.....	60
4.5 Алгоритм имитации отжига.....	61

4.6 Генетические алгоритмы .....	62
5 ПРЯМОНАПРАВЛЕННЫЕ НЕЙРОННЫЕ СЕТИ .....	64
5.1 Радиально-базисная сеть.....	64
5.2 Звезды Гроссберга .....	67
5.3 Нейросетевой гауссов классификатор.....	69
5.4 Сеть поиска максимума с прямыми связями .....	71
6 СЕТИ ЕСТЕСТВЕННОЙ КЛАССИФИКАЦИИ .....	72
6.1 Сеть Кохонена.....	72
6.2 Сеть встречного распространения .....	76
6.3 Когнитрон .....	78
6.4 Неокогнитрон .....	80
7 СЕТИ ЦИКЛИЧЕСКОГО ФУНКЦИОНИРОВАНИЯ (РЕКУРРЕНТНЫЕ СЕТИ) .....	82
7.1 Сеть Хопфилда.....	82
7.2 Двухнаправленная ассоциативная память .....	84
7.3 Сеть поиска максимума .....	85
7.4 Сеть Хемминга.....	87
7.5 Сеть теории адаптивного резонанса .....	89
СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ .....	93

## ВВЕДЕНИЕ

Общими свойствами интеллектуальной деятельности человека и ЭВМ являются сбор и обработка информации. В целом биологическую нервную систему (в том числе и человеческую) можно представить в виде информационной системы, которая собирает информацию об окружающей среде, обрабатывает ее и генерирует выходной информационный поток. Этот поток представляет собой набор управляющих команд для всех систем биологического организма.

Нервная система живых существ в процессе эволюции прошла путь от совокупности примитивных рефлексов у простейших до сложной системы анализа и синтеза информации у высших приматов [1].

Нервная система требуется не всем живым существам. Она не нужна тем, кому для выживания не требуется ни быстрой реакции, ни мгновенной перестройки организма. Это характерно для тех, кто был и будет неподвижен, то есть для растений, а также для организмов, живущих в «идеальной» для них среде обитания, например, для паразитических червей.

Одним из этапов в эволюции нервной системы стало появление упреждающей адаптации, позволяющей организму подготовиться к изменению окружающей среды заранее, до непосредственного контакта с раздражителем. Для этого развились разнообразные органы чувств, в основе работы которых лежат три механизма: химическая, физическая и электромагнитная чувствительность мембраны нервной клетки.

Однако с появлением упреждающей адаптации возник ряд проблем. Во-первых, разнотипность получаемых от рецепторов сигналов и как следствие – разнородность информации. Сопоставить сигналы можно только при их однотипной кодировке. Универсальным кодом, позволяющим сравнивать сигналы из разных органов чувств, стал электрохимический импульс, генерирующийся в нервных клетках (*нейронах*) в ответ на информацию, полученную от органов чувств.

Во-вторых, необходимость анализа полученных от разных органов чувств сигналов. Сигналы должны прийти в одно и то же место, где их можно было бы сравнить и выбрать самый важный на данный момент, который и станет побуждением к действию. Для сравнения сигналов от разных органов чувств у беспозвоночных появляются скопления нервных клеток, называемые *ганглиями* или *узлами*, которые отвечают за восприятие информации различной природы. В узлах располагаются чувствительные нейроны или их отростки, что позволяет клеткам получать информацию с периферии тела.

Однако система восприятия и анализа окружающей среды бесполезна без системы управления реакциями на сигналы – сокращением или расслаблением мышц, выбросом различных физиологически активных веществ и др. Для осуществления функций сравнения сигналов и управления реакциями у хордовых возникает головной и спинной мозг.

В постоянно меняющихся условиях окружающей среды простых адаптивных реакций становится недостаточно. В то же время изменения среды подчиняются неким физическим и планетарным законам. Сделать адекватный поведенческий выбор в нестабильной среде можно, только сравнивая разнородные сигналы с аналогичными сигналами, полученными ранее. Поэтому в процессе эволюции организм вынужден был приобрести еще одно важное преимущество – возможность сравнивать информацию во времени, оценивая опыт предыдущей жизни. Это новое свойство нервной системы называется *памятью*.

Эволюция искусственных вычислительных систем происходила, в основном, в направлении совершенствования аппаратной основы: увеличивалась тактовая частота, разрядность и др. При этом вычислительный процесс в целом осуществляется последовательно. Организовать такую последовательную обработку информации оказалось проще и понятней: сознание человека также функционирует последовательно.

Однако на низшем, «аппаратном» уровне информация в мозге обрабатывается параллельно множеством относительно простых и медленных вычислительных элементов (нейронов). Это позволяет мозгу быстро решать разнообразные сложные задачи, значительно опережая самые мощные современные вычислительные системы, созданные человеком.

Очевидно, что применение принципа массового параллелизма в обработке информации, используемого природой, позволит перейти на качественно новый уровень развития вычислительных систем. Наиболее простым, на первый взгляд, путем решением такой задачи может быть копирование действующего биологического образца (или его фрагмента). Однако на практике это оказалось чрезвычайно сложно выполнить. Огромное количество нейронов, тесно взаимодействующих друг с другом, представляют собой очень сложную систему, детально понять все аспекты работы которой пока не удастся.

Тем не менее, разработка информационных компьютерных систем, использующих принципы работы мозга, стала важным шагом в развитии интеллектуальных технологий, в результате чего был создан новый класс систем искусственного интеллекта – искусственные нейронные сети.

Теория искусственных нейронных сетей вытекает из множества дисциплин, включая физиологию, математику, нейробионику, физику, философию, биологию и лингвистику, то есть нейросетевые технологии – результат работы многих наук по одному направлению – созданию интеллектуальных систем, имеющих практическое приложение в различных областях знаний.

Пособие рекомендуется для изучения дисциплины «Нейросетевые технологии».

Дисциплина формирует представление о моделировании на основе нейронных сетей, дает возможность студентам решать широкий класс таких задач как прогнозирование развития систем и процессов, оптимизация и поддержка принятия управленческих решений, построение моделей объ-

ектов и их исследование. Место дисциплины в структурно логической схеме изучения специальности – это дисциплины, которые создают навыки моделирования объектов.

Дисциплина связана с такими дисциплинами как «Алгебра и геометрия», «Дискретная математика», «Теория вероятностей и математическая статистика», «Методы оптимизации и исследования операций», «Моделирование сложных систем», «Методы искусственного интеллекта», «Интеллектуальный анализ данных Data mining», а также используется в курсовом и дипломном проектировании.

Целью изучения курса является необходимость формирования представлений о методологии моделирования, прогнозирования, оптимизации, диагностики и управления с применением аппарата нейронных сетей.

В результате изучения дисциплины студенты должны знать:

- элементную базу нейронных сетей;
- способы соединения нейронов;
- теоретические основы функционирования нейронных сетей разных типов;

уметь:

- формализовывать задачи для их последующей обработки средствами нейросетевых технологий;
- осуществлять выбор конкретной нейросетевой парадигмы, которая обуславливается типом решаемой задачи;
- выполнять обучение нейронной сети, используя различные методы настройки;
- выполнять постобработку и анализ результатов работы нейронной сети, а также оптимизацию структуры сети.

В настоящем пособии, помимо основных положений теории нейронных сетей, приводятся примеры применения нейросетей различных видов для решения прикладных задач, основанные на опыте работы в области нейронных сетей в Донбасской государственной машиностроительной академии.

# 1 БИОЛОГИЧЕСКИЕ ОСНОВЫ НЕЙРОННЫХ СЕТЕЙ

## 1.1 Свойства биологических нейронных сетей

Объем головного мозга (или его масса), как правило, определяет уровень интеллекта, поскольку, чем больше масса головного мозга, тем большее количество нейронов он содержит. В головном мозге среднего человека содержится около  $10^{11}$  нейронов. По статистике масса мозга талантливых и гениальных людей в основном составляет 1700-1800 г, в то время как у среднего человека она порядка 1300 г, хотя есть и исключения.

Головной мозг человека, занимая только около 2 % массы тела, является основным потребителем энергии – около 20 Вт или 10 % энергетических затрат всего организма (в 5 раз больше, чем любой другой орган), причем в неактивном состоянии. Учитывая энергопотребление спинного мозга и периферической нервной системы, общие затраты на содержание нервной системы составляют около 15 % энергии всего организма в состоянии покоя. В активном состоянии энергетические затраты только головного мозга возрастают более чем в 2 раза [1]. Аналогичная ситуация складывается и в отношении потребления кислорода. В состоянии покоя головной мозг человека использует до 20 % кислорода тела.

В связи с этим у многих небольших животных с относительно большим мозгом возник механизм защиты организма от перерасхода энергии – *торпидность* или впадение в спячку на несколько часов. Мелкие теплокровные вообще могут находиться только в двух основных состояниях: гиперактивности и спячки. Промежуточное состояние малоэффективно, поскольку энергетические расходы не компенсируются поступающей пищей. В физиологии крупных млекопитающих торпидность невозможна, но все же крупные теплокровные тоже различными способами защищают себя от повышенных энергозатрат (например, зимняя псевдоспячка медведей, пребывание кошачьих в полудреме до 80% своей жизни).

Чем меньше времени мозг работает в интенсивном режиме, тем дешевле обходится его содержание. Минимизация времени интенсивного режима работы нервной системы в основном достигается большим набором врожденных, инстинктивных программ поведения, которые хранятся в мозге как набор инструкций. В целях экономии энергии мозг почти не используется для принятия решений, основанных на личном опыте животного.

В то же время следует отметить, что если сравнивать удельные энергозатраты мозга и ЭВМ при решении одинаковых задач, то мозг с энергетической точки зрения намного эффективнее ЭВМ: на одну элементарную операцию в секунду мозг тратит  $10^{-16}$  Дж, а компьютер –  $10^{-6}$  Дж [2].

При решении ряда задач мозг превосходит в производительности обычную ЭВМ на 4-6 порядков. При этом реакция нейронов на 5-6 порядков медленнее реакции кремниевых логических элементов: длительность



событий в кремниевых элементах измеряется в наносекундах, а в нейронах – в миллисекундах. Разница компенсируется параллельностью обработки информации в мозге.

Основные особенности работы мозга [3]:

1 *Параллельность обработки информации.* Каждый нейрон формирует свой выход только на основе своих входов и собственного внутреннего состояния под воздействием общих механизмов регуляции нервной системы;

2 *Способность к полной обработке информации.* Все известные человеку задачи решаются нейронными сетями. Неизвестно никаких принципиальных ограничений на сложность задач, решаемых биологическими нейронными сетями;

3 *Самоорганизация.* В процессе работы биологические нейросети самостоятельно, под воздействием внешней среды, обучаются решению разнообразных задач. Нервная система сама формирует алгоритмы своей деятельности, уточняя и усложняя их в течение жизни;

4 Биологические нейросети являются *аналоговыми системами*. Информация поступает в сеть по большому количеству каналов и кодируется по пространственному принципу: вид информации определяется номером нервного волокна, по которому она передается;

5 *Надежность.* Выход из строя даже 10 % нейронов в нервной системе не прерывает ее работы. В последовательных ЭВМ, основанных на принципах фон-Неймана, сбой одной ячейки памяти или одного узла в аппаратуре приводит к краху системы.

## 1.2 Биологический нейрон

*Нейрон* – это клетка, являющаяся основной структурно-функциональной единицей нервной системы.

Схема биологического нейрона представлена на рис. 1.1.

Биологический нейрон содержит следующие основные структурные единицы:

*Сома (тело клетки)* содержит ядро, митохондрии, обеспечивающие клетку энергией, и другие органеллы, поддерживающие жизнедеятельность клетки. Тело клетки заключено в мембрану.

*Дендриты* – входные волокна, собирающие информацию от других нейронов. Длина их обычно не больше 1 мм.

*Аксон* – длинное, иногда больше метра, выходное нервное волокно клетки, обеспечивающее проведение нервного импульса и передачу воздействия на другие нейроны или мышечные волокна. Импульс генерируется в *аксонном холмике*. Ближе к концу аксон часто ветвится.

*Синапс* – место контакта нервных волокон, через которое передается возбуждение от клетки к клетке.

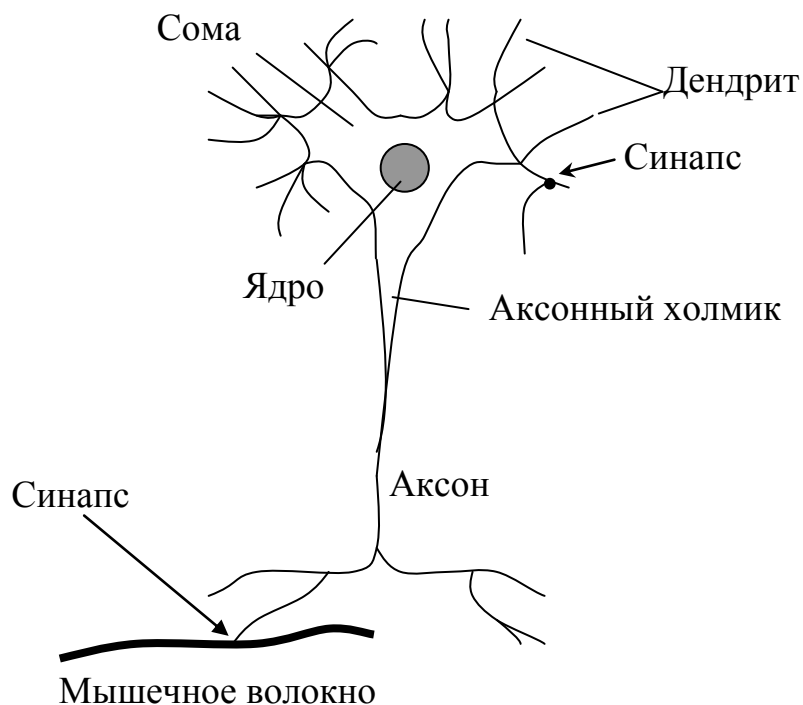


Рисунок 1.1 – Структурная схема биологического нейрона

Нейрон обладает следующими свойствами:

1 Меняет внутреннее состояние с течением времени, реагируя на входные сигналы и формируя выходные воздействия. Таким образом, нейрон является активной динамической системой.

2 Взаимодействует с другими нейронами путем обмена электрохимическими сигналами.

3 Имеет большое количество синапсов – контактов для передачи информации.

В центральной нервной системе человека насчитывается до 1000 типов нейронов. Они отличаются картиной дендритов, длиной аксона, распределением синапсов около клетки и имеют диаметр от 5 до 100 мкм.

Близкие по функциям клетки образуют скопления, шаровидные или параллельные слоистые. В мозгу выделены сотни скоплений. Например, кора головного мозга. Толщина коры – 2 мм, а площадь – около 1000 см<sup>2</sup>.

Нейроны окружены *глиальными* клетками (нейроглия – «нервный клей»). Эти клетки более многочисленны, чем нейроны, и составляют около 50 % от объема центральной нервной системы. Они способны к делению в течение всей жизни. По размеру глиальные клетки в 3-4 раза меньше нервных, их число достигает  $1,4 \cdot 10^{11}$ . С возрастом доля глиальных клеток в мозге увеличивается, а число нейронов уменьшается. Глиальные клетки формируют защитный *гематоэнцефалический барьер*. Кроме этого, они обеспечивают структурную основу мозга. Весь объем мозга, не занятый нейронами и кровеносными сосудами, заполнен глиальными клетками. Также они выполняют снабжение нейронов питательными веществами.

Нейроны связаны между собой посредством синапсов в *нейронную сеть*. Синапсы представляют собой не просто «сросшиеся» нервные волокна, а устройства преобразования информации. Упрощенная схема прохождения сигнала через синапс представлена на рис. 1.2.

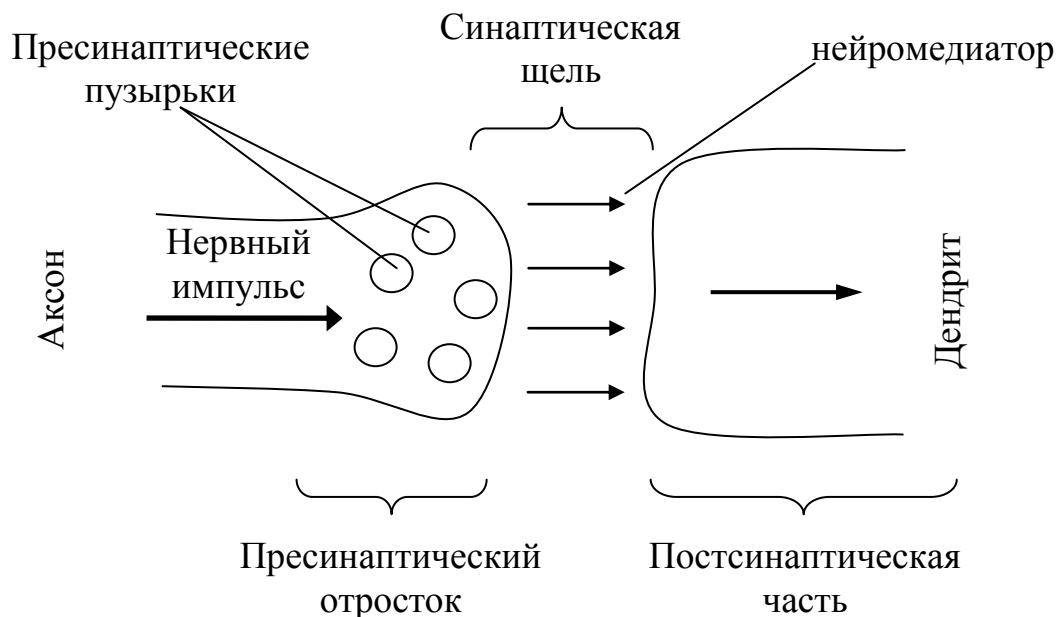


Рисунок 1.2 – Схема прохождения сигнала через синапс.

В месте контакта мембраны клеток не сливаются, между ними всегда существует небольшой промежуток – синаптическая щель шириной 40-50 нм, поэтому сигнал передается с задержкой около 0,5 с. Передача через синапс почти всегда однонаправленная.

По эффекту различают возбуждающие и тормозящие синапсы. Синапс может усиливать или ослаблять входной сигнал, причем коэффициент усиления может меняться. Таким образом, синапс можно упрощенно рассматривать как *усилительное звено* с коэффициентом усиления  $K$  (см. рис. 1.3).

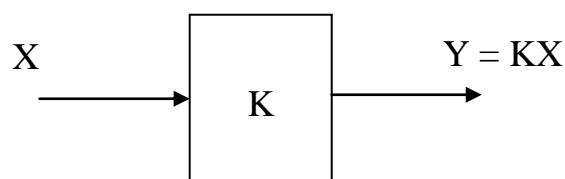


Рисунок 1.3 – Представление синапса усилительным звеном.

У одного нейрона может быть от 300 до 20 000 синапсов. Синапсы являются важнейшими элементами нервной системы. Блокировка работы синапсов вызывает почти мгновенную смерть. Таким действием обладают, например, курареподобные яды.

## 1.4 Нервный импульс

*Нервный импульс (спайк)* – процесс распространения возбуждения по аксону от тела клетки (аксонного холмика) до окончания аксона. Это основная единица информации, передаваемая по волокну. Потенциал нервного импульса составляет 80-110 мВ. Импульсы по волокну передаются в виде скачков потенциала внутриклеточной среды по отношению к внешней среде, окружающей клетку. В качестве заряженных частиц используются ионы натрия и калия. Скорость передачи нервного импульса составляет от 1 до 100 м/с.

На рис. 1.4 показан вид нервных импульсов в зрительном нервном волокне при воздействии вспышек света разной интенсивности [3].

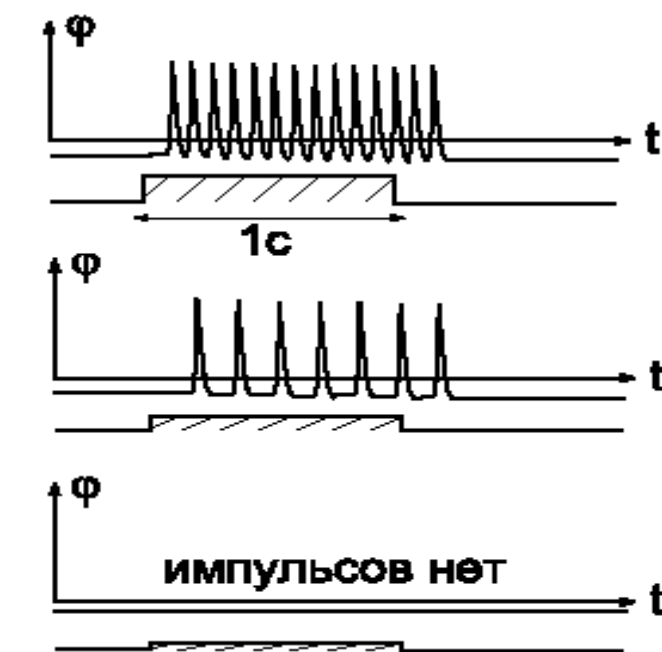


Рисунок 1.4 – Нервные импульсы, возникающие в зрительном нервном волокне, при воздействии вспышки света постоянной длительности, но разной интенсивности

Свет действует на фоторецепторы и вызывает импульсацию в соответствующем зрительном волокне. От интенсивности света зависит не амплитуда импульсов и их форма, а плотность и общее количество. Длительность импульса составляет около 0,5 мс. После импульса наступает *абсолютная рефрактерная фаза* – полная невозбудимость клетки, которая продолжается 1-2 мс и прямо зависит от длительности импульса.

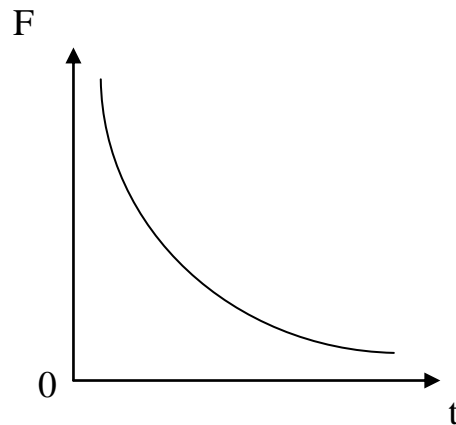
Импульс подчиняется закону «все или ничего»: он либо совсем не возникает на раздражение клетки, либо он максимальной величины.

Для того чтобы раздражитель вызвал возбуждение, должны выполняться следующие законы:

1 *Закон силы*: чтобы возникло возбуждение, раздражитель должен быть достаточно сильным – пороговым или выше порогового;

2 *Закон градиента*: для того чтобы раздражитель вызвал возбуждение, он должен нарастать достаточно быстро. Если раздражитель нарастает медленно, то происходит повышение порога раздражения. Поэтому для получения возбуждения величина стимула должна быть больше, чем при мгновенном нарастании;

3 *Закон времени*: раздражитель, вызывающий возбуждение, должен быть достаточно длительным, чтобы вызвать возбуждение. Зависимость пороговой силы раздражителя ( $F$ ) от длительности его действия ( $t$ ) носит гиперболический характер – чем меньше по времени действует на ткань раздражитель, тем выше требуется его сила для инициации возбуждения (см. рис. 1.5).



*Рисунок 1.5 – Закон времени*

*Контрольные вопросы по теме 1:*

- 1 Приведите свойства биологических нейронных сетей.
- 2 Изобразите схематично устройство биологического нейрона.
- 3 Перечислите структурные единицы биологического нейрона.
- 4 Перечислите свойства биологического нейрона.
- 5 Какова роль синапсов в нейронной сети?
- 6 Каким образом синапс участвует в обработке информации?
- 7 Что такое нервный импульс?
- 8 Изложите законы, необходимые для того, чтобы раздражитель вызвал возбуждение нейрона.
- 9 Каким образом импульс отражает интенсивность раздражителя, который его вызвал?

## 2 ОСНОВЫ ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ

### 2.1 Искусственный нейрон

Первый искусственный (формальный) нейрон, имитирующий свойства биологического нейрона, был разработан У.С. Маккаллоком и У. Питтсом в 1943 г. [4]. Его схема представлена на рис. 2.1.

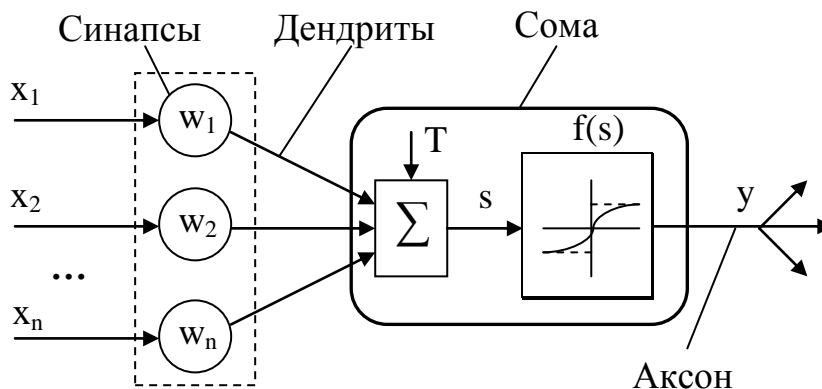


Рисунок 2.1 – Схема искусственного нейрона

Каждый нейрон характеризуется своим текущим состоянием. Он состоит из элементов трех типов:

- 1 Умножителей – аналогов синапсов биологического нейрона;
- 2 Сумматора (в биологическом нейроне функцию сумматора выполняет аксонный холмик);
- 3 Нелинейного преобразователя – порог нервного импульса биологического нейрона (сигнал «Т» на рис. 2.1).

Синапсы осуществляют связь между нейронами, умножая входной сигнал на число, характеризующее силу связи. Это число называется *вес синапса* или *весовой коэффициент*.

Сумматор выполняет сложение сигналов, поступающих по синаптическим связям от других нейронов и внешних входных сигналов.

Нелинейный преобразователь реализует нелинейную функцию одного аргумента – выхода сумматора. Эта функция называется *функцией активации нейрона* или *активационной функцией*.

Нейрон в целом реализует скалярную функцию некоторого аргумента, т. е.  $y = F(x_1, x_2, \dots, x_n)$ , где  $F$  – функция вход-выход нейрона. Определить эту функцию можно с помощью *математической модели* нейрона:

$$\begin{cases} s = \sum_{i=1}^n w_i x_i \\ y = f(s), \end{cases} \quad (2.1)$$

где  $x_i$  – компонент вектора входного сигнала (входной сигнал),  
 $i=1 \dots n$ ;

$n$  – число входов нейрона;

$w_i$  – вес синапса;

$s$  – результат суммирования;

$f$  – функция активации;

$y$  – выходной сигнал нейрона.

Как видно из формулы (2.1), сумматор выполняет скалярное произведение входного вектора  $\bar{X}$  и вектора весовых коэффициентов  $\bar{W}$ , т. е.  $s = (\bar{X}, \bar{W})$ .

В общем случае входной сигнал, весовые коэффициенты и смещение могут принимать действительные значения. Синаптические связи с положительными весами называют *возбуждающими*, а с отрицательными весами – *тормозящими*.

## 2.2 Функции активации

Наиболее часто используются следующие функции активации:

1 Пороговая (ступенчатая, функция Хэвисайда)

$$f(s) = \begin{cases} 0, s < T \\ 1, s \geq T, \end{cases} \quad (2.2)$$

где  $T$  – порог.

Или

$$f(s) = \begin{cases} -1, s < T \\ 1, s \geq T. \end{cases} \quad (2.3)$$

В первом случае (формула (2.2)) функция имеет следующий вид:

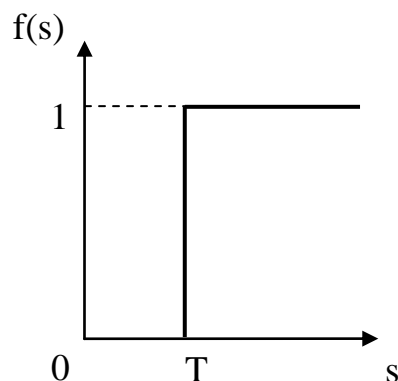


Рисунок 2.2 – Пороговая функции активации с диапазоном изменения  $\{0;1\}$

Во втором случае (формула (2.3)):

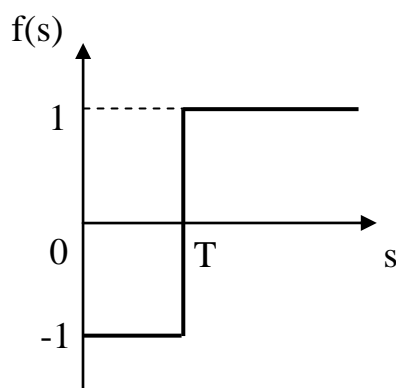


Рисунок 2.3 – Пороговая функции активации с диапазоном изменения  $\{-1;1\}$

Пороговая функция не позволяет моделировать схемы с непрерывными сигналами, что ограничивает ее применение.

2 Пологая ступенька (линейная с насыщением)

$$f(s) = \begin{cases} 0, s \leq T \\ \frac{s-T}{\Delta}, T \leq s < T + \Delta \\ 1, s \geq T + \Delta, \end{cases} \quad (2.4)$$

где  $\Delta$  – параметр, определяющий ширину линейного участка функции.

График функции представлен на рис. 2.4.

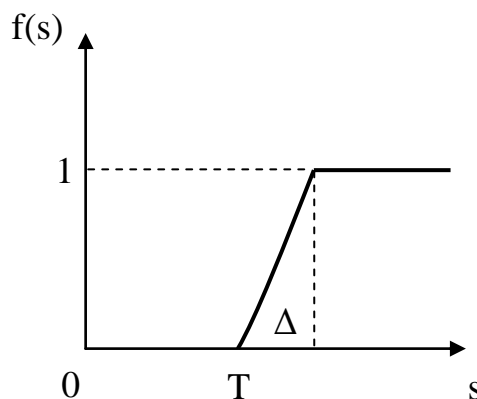


Рисунок 2.4 – Функция активации «Линейная с насыщением»

Также как и в пороговой функции может применяться диапазон изменения выходного сигнала  $[-1;1]$ .

3 Функции сигмоидального типа.

Эти функции являются строго монотонно возрастающими и имеют отличную от нуля производную на всей области определения. Сигмоидальная функция позволяет решить проблему шумового насыщения, так



как различные участки сигмоиды имеют разный коэффициент усиления: центральная область функции, имеющая большой коэффициент усиления, решает проблему обработки слабых сигналов, в то время как области с падающим усилением на положительном и отрицательном концах входного диапазона позволяют воспринимать сетью большие сигналы без насыщения. Таким образом, нейрон функционирует с большим усилением в широком диапазоне уровня входного сигнала.

К сигмоидальным функциям относятся:

### 3.1 Экспоненциальная сигмоида (логистическая функция, функция Ферми)

$$f(s) = \frac{1}{1 + e^{-c(s+T)}}, \quad (2.5)$$

где  $c$  – параметр, определяющий наклон функции активации (обычно принимают  $c = 1$ );

$T$  – значение *нейронного смещения* (эквивалент порога).

Как видно из формулы (2.5), нейронное смещение можно интерпретировать как дополнительный вход, на который постоянно подается единица, а весовой коэффициент равен  $T$ .

График экспоненциальной сигмоиды представлен на рис. 2.5.

Наклон центрального участка сигмоиды определяет ее коэффициент усиления: чем круче сигмоида, тем выше коэффициент усиления. Нейронное смещение сдвигает сигмоиду вдоль оси абсцисс, не изменяя ее формы. Перемещение сигмоиды позволяет переносить рабочий (центральный) участок сигмоиды к доминирующему диапазону входного сигнала.

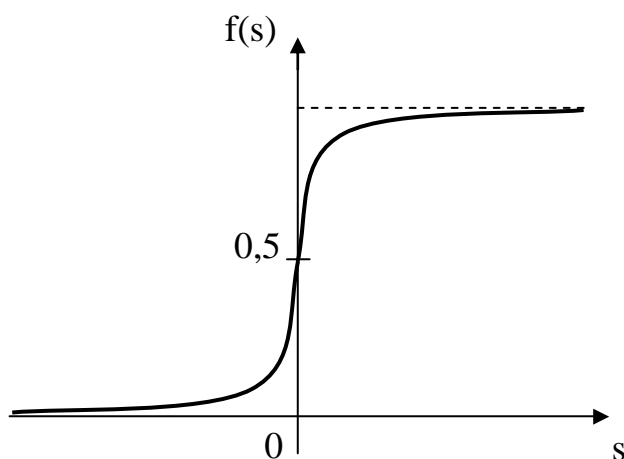


Рисунок 2.5 – Экспоненциальная сигмоида

Недостатком экспоненциальной сигмоиды является то, что она принимает значения в диапазоне  $(0; 1)$ , поскольку доказано, что при использовании симметричного диапазона, например,  $(-1; 1)$  время обучения нейросетей сокращается на 30 – 50 % [5].

### 3.2 Симметрированная экспоненциальная сигмоида

$$f(s) = \frac{2}{1 + e^{-c(s+T)}} - 1. \quad (2.6)$$

Функция является модификацией функции (2.5) и имеет диапазон изменения выходного сигнала  $(-1; 1)$ :

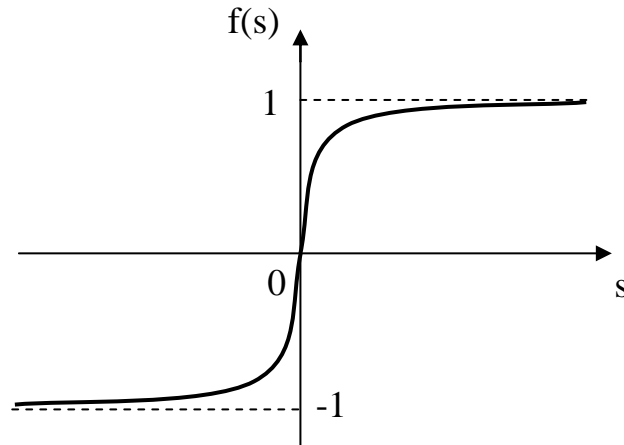


Рисунок 2.6 – Симметрированная экспоненциальная сигмоида

### 3.3 Рациональная сигмоида

$$f(s) = \frac{s + T}{c + |s + T|}, \quad (2.7)$$

где  $c$  – параметр, определяющий коэффициент усиления функции. Его необходимо либо определять в процессе настройки сети, либо задавать как константу (обычно принимают  $c = 0,1$ ).

Достоинством функции является то, что на ее вычисление требуется меньше всего времени.

### 3.4 Арктангенс

$$f(s) = \frac{2}{\pi} \arctg(s + T). \quad (2.8)$$

### 3.5 Гиперболический тангенс

$$f(s) = \text{th}(s + T) = \frac{e^{s+T} - e^{-(s+T)}}{e^{s+T} + e^{-(s+T)}}. \quad (2.9)$$

Для вычисления этой функции требуется больше всего процессорного времени.

Функции (2.7) – (2.9) также являются симметричными относительно оси абсцисс и их график аналогичен по форме графику функции (2.6) (см. рис. 2.6).

#### 4 Кривая Гаусса (радиально-базисная функция)

$$f(s) = e^{\frac{-s^2}{\sigma^2}}, \quad (2.10)$$

где  $\sigma$  – параметр, задающий ширину окна функции;

$s$  – евклидово расстояние между входным вектором и вектором весовых коэффициентов:

$$s = \|\bar{x} - \bar{w}\| = \sqrt{\sum_{i=1}^n (x_i - w_i)^2}, \quad (2.11)$$

где  $\bar{w}$  – вектор координат центра окна активационной функции, задаваемый весовыми коэффициентами нейрона;

$\bar{x}$  – входной вектор;

$n$  – размерность входного вектора.

Таким образом, сигнал  $s$  вычисляется здесь уже не как скалярное произведение, что делает отличие от нейрона Маккаллока еще значительней.

График радиально-базисной функции представлен на рис. 2.7.

Функция применяется, когда реакция нейрона должна быть максимальной для некоторого входного сигнала.

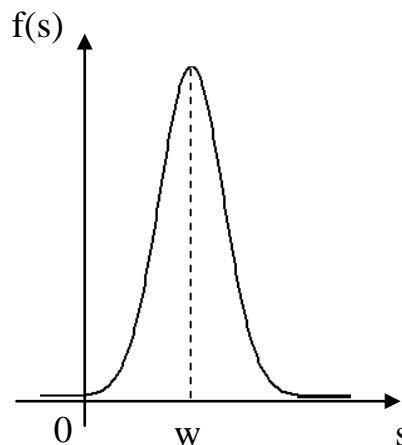


Рисунок 2.7 – Радиально-базисная функция

Выбор функции активации определяется:

- 1 Спецификой задачи;
- 2 Удобством реализации;
- 3 Алгоритмом обучения: некоторые алгоритмы накладывают ограничения на вид функции активации.

## 2.3 Свойства искусственных нейронных сетей

*Искусственные нейронные сети* – совокупность моделей биологических нейронных сетей. Искусственные нейронные сети представляют собой сеть элементов – искусственных нейронов – связанных между собой. Сеть обрабатывает входную информацию и в процессе изменения своего состояния во времени формирует совокупность выходных сигналов. Обычно нейронные сети оперируют цифровыми величинами.

Нейронную сеть обычно представляют в виде направленного графа, вершинами которого являются нейроны, а дугами – сигналы, проходящие по сети. Нейроны, расположенные в ряд, формируют *слой* нейронов (см. рис. 2.8).

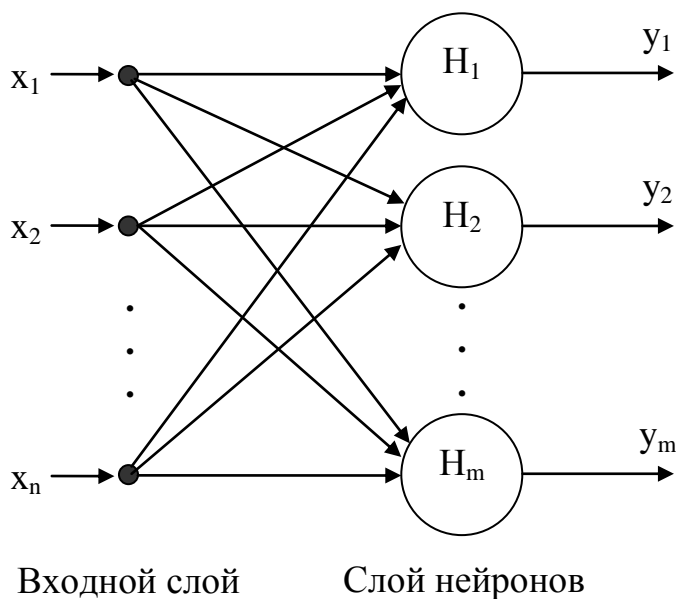


Рисунок 2.8 – Расположение нейронов в виде слоя

Если нейроны получают на вход общие сигналы, то обычно для удобства представления входные сигналы передаются через *распределительные нейроны*. Распределительные нейроны (на рис. 2.8 обозначены черными кружками) не участвуют в обработке сигналов и не имеют настраиваемых параметров. Они не производят вычислений и служат для передачи сигналов на основные нейроны, т.е. являются точками ветвления. Вид входного нейрона представлен на рис. 2.9.

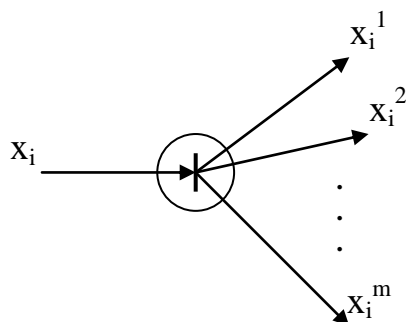


Рисунок 2.9 – Структура распределительного нейрона

На рис. 2.9  $x_i = x_i^1 = x_i^2 = \dots = x_i^m$ .

Обычно входные нейроны не учитываются в общем количестве слоев, а выделяются как *входной слой*.

В общем случае нейронная сеть может содержать произвольное количество слоев, связанных между собой выходными сигналами нейронов.

Различают несколько типов связей между слоями с номерами  $k$  и  $(k+s)$ :

- *последовательные*, если  $s = 1$ . Здесь сигнал распространяется от предыдущего слоя к следующему;
- *перекрестные*, если  $s > 1$ . Сигнал распространяется, минуя последующий слой;
- *обратные*, если  $s < 0$ . Сигнал подается на слой, находящийся ближе к входному слою, чем слой, формирующий сигнал;
- *латеральные (боковые)*, если  $s = 0$ . Такие связи соединяют нейроны одного слоя и являются разновидностью обратных связей.

На рис 2.10 изображены возможные виды связей.

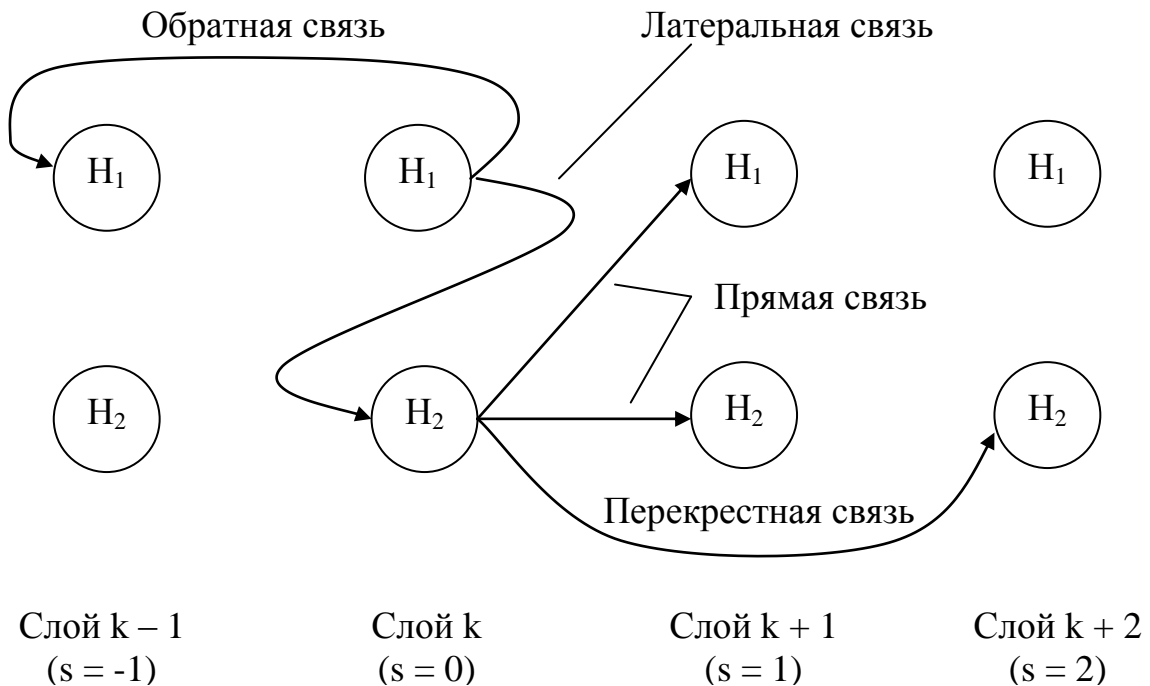
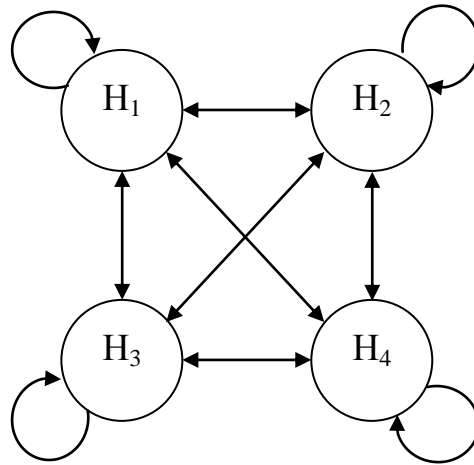


Рисунок 2.10 – Связи между слоями нейронной сети

Сети, в которых все нейроны соединены друг с другом всеми возможными способами, называются *полносвязными* (рис. 2.11).

Если в нейросети все нейроны имеют одинаковые функции активации, то такая сеть называется *однородной (гомогенной)*, в противном случае – *неоднородной (гетерогенной)*.



*Рисунок 2.11 – Полносвязная сеть*

Свойства современных нейронных сетей:

1 *Обучаемость*. Подобно биологическим системам, которые они моделируют, нейронные сети сами модифицируют себя в результате попыток достичь лучшей модели поведения;

2 *Способность к обобщению* – способность нейронных сетей извлекать из конечного набора данных знания, позволяющие распознавать незнакомые сети образы. Оно проявляется также в нечувствительности к малым изменениям входных сигналов, в том числе и шуму;

3 *Способность к абстрагированию* – способность нейронных сетей самостоятельно индуцировать из набора данных системные выводы, частично противоречащие отдельным элементам в исходных данных. Это свойство дает возможность восстанавливать правильный образ по нескольким искаженным вариантам.

Нейронные сети превосходят последовательные машины в решении тех же задач, в которых машину превосходит человек. Задачи, требующие большого объема вычислений или высокой точности, лучше выполняются обычной ЭВМ.

Нейронные сети наиболее эффективно использовать для

1 Решения не формализуемых и плохо формализуемых задач, алгоритм решения которых связан с необходимостью включения в процесс обучения реального экспериментального материала;

2 Поиска закономерностей в массивах данных;

3 Нелинейной аппроксимации многомерных функций;

4 Классификации входного пространства по многим признакам;

5 Распознавания образов;

6 Поиска по ассоциациям;

7 Построения моделей, способных адаптироваться к изменениям условий работы;

8 Решения задач, характеризующихся резким увеличением размерности пространства решений, большими объемами входной информации с зашумленными, частично противоречивыми, неполными или избыточными

ми данными.

*Преимущества* аппроксимации функций с помощью нейросетей:

1 Большая гибкость базовых функций нейросетей, связанная с адаптацией к входным данным;

2 Высокая параллельность и синхронность выполняемых операций на нейронных сетях;

3 Однородность базовых операций в нейронных сетях (умножение, сложение, нелинейное безынерционное преобразование) и отсутствие сложных иррациональных операций над операндами (деление, извлечение корня и т. д.), характерных для алгоритмов, свойственных однопроцессорным машинам;

4 Технологическая простота выполнения этих операций при различных способах физической реализации;

5 Возможность управления элементами и структурой нейронных сетей в процессе поиска наилучшей аппроксимации;

6 Возможность использования нейронных сетей для аналитического описания преобразования входного пространства в выходное. Структура нейросети позволяет также аналитически получать алгоритм ее настройки и контролировать его функционирование в процессе решения задачи;

7 Инвариантность (независимость) методов синтеза нейронных сетей от размерности пространства признаков;

8 Отображение сложности решаемой задачи в сложности нейронной сети, используемой для решения конкретной задачи. При этом параметры нейросети не имеют непосредственной физической связи с параметрами объекта, т. е. нейросети являются непараметрическими моделями;

9 Зависимость скорости сходимости алгоритма настройки нейронной сети от размера тренировочного набора данных, в отличие от гладких функций с ограниченными производными порядка  $s > 0$ , у которых скорость сходимости зависит от размерности входного пространства признаков [8 – 10].

*Сложности* реализации параллельной обработки:

1 *Тирания межсоединений*. Каждый процессор в параллельной системе связан с большим количеством других. Количество связей занимает намного больший объем, чем сами процессоры. Такая плотность связей не реализуется в обычных интегральных схемах;

2 *Трехмерность структуры связей* между процессорами. Технологически такие связи тоже пока невыполнимы.

3 *Сложность программирования*.

Из-за вышеуказанных сложностей большая часть исследований проводится двумя способами:

1 Моделирование нейронных сетей на обычных последовательных ЭВМ;

2 Создание специализированных нейроплат и нейропроцессоров для ускорения работы ЭВМ с нейронными сетями.

Первый способ дает проигрыш в быстросействии даже по сравнению с обычной ЭВМ, а второй способ не позволяет переходить от одной модели нейросети к другой, т. к. модель определяется используемой нейропла-той.

## **2.4 Применение нейронных сетей в задачах распознавания обра-зов**

*Образ* – классификационная группировка в системе классификации, объединяющая (выделяющая) определенную группу объектов по некоторому признаку. Образы обладают свойством, проявляющимся в том, что ознакомление с конечным числом явлений из одного и того же множества дает возможность узнавать сколь угодно большое число его представителей. В качестве образа можно рассматривать и некоторую совокупность состояний объекта управления, причем вся эта совокупность состояний характеризуется тем, что для достижения заданной цели требуется одинаковое воздействие на объект. Образы обладают характерными объективными свойствами в том смысле, что разные люди, обучающиеся на различном материале наблюдений, большей частью одинаково и независимо друг от друга классифицируют одни и те же объекты.

Проблема распознавания образов состоит из двух частей: *обучения* и *распознавания*. Обучение осуществляется путем показа отдельных объектов с указанием их принадлежности тому или другому образу. В результате обучения распознающая система должна приобрести способность реагировать одинаковыми реакциями на все объекты одного класса и различными – на все объекты различных классов. Процесс обучения должен завершиться только путем показов конечного числа объектов без каких-либо других подсказок. За обучением следует процесс идентификации новых объектов, который характеризует действия уже обученной системы. Автоматизация этих процедур и составляет проблему обучения распознаванию образов.

В круг задач, которые могут решаться с помощью распознающих систем, входят не только задачи распознавания зрительных и слуховых образов, но и задачи распознавания сложных процессов и явлений.

Однако каждый объект наблюдения может воздействовать по-разному, в зависимости от условий восприятия. Кроме того, объекты одного и того же образа могут достаточно сильно отличаться друг от друга и по-разному воздействовать на воспринимающие органы. Выбор исходного описания объектов является одной из центральных задач проблемы распознавания образов.

Задача распознавания сводится к следующей математической постановке: необходимо построить отображение  $X \rightarrow Y$  такое, чтобы на каждый возможный входной сигнал  $X$  формировался правильный выходной сигнал  $Y$ . Отображение задается конечным набором пар «вход-известный выход»



[6]. Число таких пар (*обучающих примеров*) существенно меньше общего числа возможных сочетаний значений входных и выходных сигналов. Совокупность всех обучающих примеров носит название *обучающей выборки*.

В результате построения отображения  $X \rightarrow Y$  необходимо добиться того, чтобы:

1 Обеспечивалось формирование правильных выходных сигналов в соответствии со всеми примерами обучающей выборки;

2 Обеспечивалось формирование правильных выходных сигналов в соответствии со всеми возможными входными сигналами, которые не вошли в обучающую выборку.

В задачах распознавания образов  $X$  – некоторое представление объекта (изображение, вектор чисел и т.д.),  $Y$  – номер класса, к которому принадлежит входной объект.

Если объект (или процесс) описывается  $n$  характеризующими его признаками, то каждый такой объект представляется в виде характеристического вектора  $\bar{X} = (x_1, x_2, \dots, x_n)$ . Сопоставив каждому признаку координатную ось и «натянув» на полученную систему координат пространство, получим  $n$ -мерное *пространство признаков*, объекты в котором представляются в виде точек. Если эти точки принадлежат к различным классам и точки одинаковых классов расположены в пространстве признаков группами, то области (подпространства), в которых они находятся и являются образами. Тогда распознавание образов геометрически заключается в задании в пространстве признаков такой поверхности, которая смогла бы разделить эти области. Реализация такого многомерного функционального преобразования является задачей аппроксимацией, к которой может быть сведена большая часть прикладных задач.

В отношении расположения образов в пространстве признаков была выдвинута *гипотеза компактности*. Если предположить, что в процессе обучения пространство признаков формируется исходя из задуманной классификации, то тогда можно надеяться, что заданное пространство признаков само по себе задает свойство, под действием которого образы в этом пространстве легко разделяются.

*Гипотеза компактности*: образам соответствуют компактные множества в пространстве признаков.

Однако эту гипотезу не всегда удавалось подтвердить экспериментально. Поэтому гипотеза компактности превратилась в признак возможности удовлетворительного решения задач распознавания.

Таким образом, залогом возможности решения задачи распознавания образом является правильно сформированное пространство признаков, т. е. правильный подбор признаков, характеризующих объект моделирования.

Во время своей работы, нейронная сеть преобразовывает входной вектор  $\bar{X}$  в выходной –  $\bar{Y}$ . Или, решая задачу распознавания образов, выполняет преобразование пространства признаков в пространство решений. Это

преобразование осуществляется функцией вход-выход сети, являющейся в данном случае аппроксимирующим выражением.

## 2.5 Принципы обучения нейронных сетей

*Обучением* называют процесс выработки в некоторой системе той или иной реакции на группы внешних идентичных сигналов путем многократного воздействия на систему внешней корректировки. Механизм генерации этой корректировки практически полностью определяет алгоритм обучения. *Самообучение* отличается от обучения тем, что здесь дополнительная информация о верности реакции системе не сообщается.

*Адаптация* – это процесс изменения параметров и структуры системы на основе текущей информации с целью достижения определенного состояния системы при начальной неопределенности и изменяющихся условиях работы.

Обучающие алгоритмы могут быть классифицированы как алгоритмы *обучения с учителем* и *без учителя*. В первом случае существует учитель, который предъявляет входные образы сети, сравнивает результирующие выходы с требуемыми, а затем настраивает веса сети таким образом, чтобы уменьшить различия. Такой обучающий механизм биологически неправдоподобен. Поэтому, хотя данный подход привел к большим успехам при решении прикладных задач, он часто отвергается исследователями, полагающими, что искусственные нейронные сети обязательно должны использовать те же механизмы, что и человеческий мозг.

Во втором случае обучение проводится без учителя: при предъявлении входных образов сеть самоорганизуется посредством настройки своих весов согласно определенному алгоритму. Вследствие отсутствия указания требуемого выхода в процессе обучения результаты непредсказуемы с точки зрения определения возбуждающих образов для конкретных нейронов. При этом, однако, сеть организуется в форме, отражающей существенные характеристики обучающего набора. *Например*, входные образы могут быть классифицированы согласно степени их сходства так, что образы одного класса активизируют один и тот же выходной нейрон [7].

*Контрольные вопросы по теме 2:*

- 1 Изобразите структуру формального нейрона.
- 2 Приведите математическую модель нейрона.
- 3 Приведите функции активации нейронов.
- 4 Приведите типы связей между слоями нейронной сети.
- 5 Опишите свойства современных нейронных сетей.
- 6 Для каких задач наиболее эффективно использовать нейросети?
- 7 Перечислите преимущества аппроксимации функций с помощью нейронных сетей.
- 8 Что такое обучения с учителем и без учителя?

## 3 ПЕРСЕПТРОНЫ

### 3.1 Однослойный персептрон

Персептрон (perceptron) разработан Ф. Розенблаттом в 1959 г.

Персептрон состоит из ряда нейронов, формирующих слой (нейроны  $H_1 - H_n$ ). Нейроны имеют вид, представленный на рис. 2.1 и функционируют согласно правилу (2.1). Функция активации нейронов пороговая (формула (2.3)).

Схематически однослойный персептрон представлен на рис. 3.1.

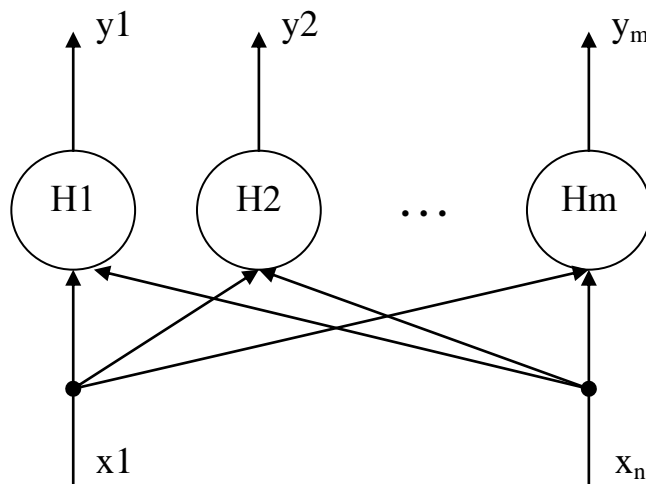


Рисунок 3.1 – Однослойный персептрон

Однослойный персептрон способен распознавать простейшие образы. Отдельный нейрон персептрона выдает значения 1 или минус 1. В зависимости от значения выходного сигнала принимается решение: 1 – входной сигнал принадлежит классу А, минус 1 – входной сигнал принадлежит классу В. Персептрон, состоящий из одного нейрона, формирует разделяющую прямую, уравнение которой зависит от значений синаптических весов и порога. На рис. 3.3 представлен пример разделения персептроном объектов на 2 класса.

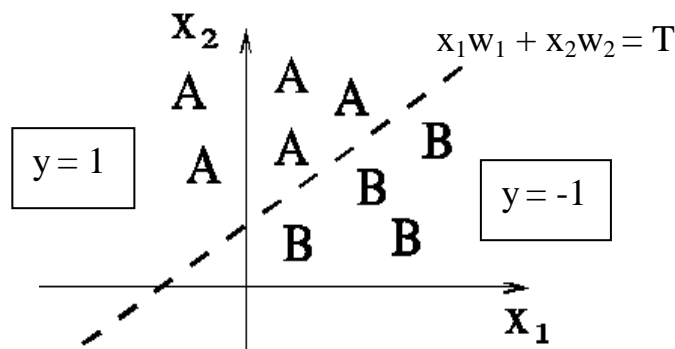


Рисунок 3.3 – Пример разделения персептроном объектов на 2 класса

В случае  $n$  входных сигналов разделяющая поверхность будет представлять собой гиперплоскость в  $n$ -мерном пространстве признаков.

Для разделения пространства признаков на большее число классов необходимо взять соответствующее число нейронов. Минимально необходимое количество нейронов для двоичного персептрона  $m = \log_2 K$ , где  $K$  – число классов. Однако на практике лучше выделять для каждого класса отдельный нейрон.

В 1962 г. Ф. Розенблатт доказал теорему об обучении персептрона: персептрон способен научиться всему, что он способен представлять.

Обучение персептрона является обучением с учителем. Персептрон обучают, подавая множество образов по одному на его вход и подстраивая веса до тех пор, пока для всех образов не будет достигнут требуемый выход.

*Алгоритм обучения дискретного персептрона:*

- 1 Подать входной вектор и вычислить выход;
- 2 а) если выход правильный, то перейти на шаг 1;  
б) если выход неправильный и равен минус 1, то добавить все входы к соответствующим им весам;  
в) если выход неправильный и равен 1, то вычесть каждый вход из соответствующего ему веса;
- 3 Перейти на шаг 1.

*Области применения однослойных персептронов:* распознавание образов, классификация.

*Преимущества:* программные или аппаратные реализации модели очень просты. Простой алгоритм обучения.

*Недостатки:*

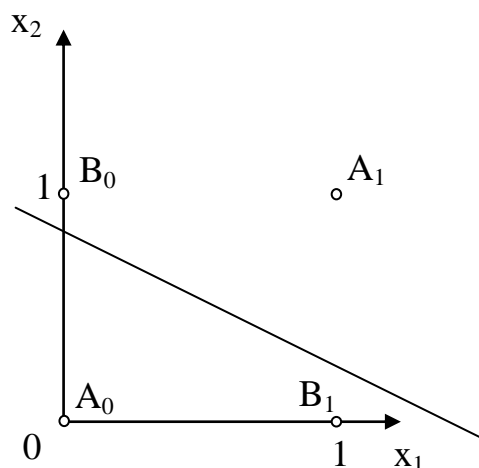
- 1 Не доказано, что персептронный алгоритм обучения быстрее прямого перебора весов;
- 2 Примитивные линейные разделяющие поверхности дают возможность решать лишь самые простые задачи распознавания.

В 1969 г. М.Л. Минский доказал, что имеются жесткие ограничения на то, что могут выполнять однослойные персептроны. В частности однослойный персептрон не может воспроизвести такую простую функцию, как ИСКЛЮЧАЮЩЕЕ ИЛИ (XOR). Это булева функция двух переменных, принимающая значение «истина», когда значения переменных различны, и «ложь» – когда одинаковы.

*Таблица 3.1 – Таблица истинности для функции ИСКЛЮЧАЮЩЕЕ ИЛИ*

Точки	$x_1$	$x_2$	Выход
$A_0$	0	0	0
$B_0$	0	1	1
$B_1$	1	0	1
$A_1$	1	1	0

На рис. 3.4 графически продемонстрирована невозможность реализации однослойным перцептроном функции ИСКЛЮЧАЮЩЕЕ ИЛИ.



*Рисунок 3.4 – Невозможность реализации однослойным перцептроном функции ИСКЛЮЧАЮЩЕЕ ИЛИ*

Как видно из рисунка перцептронная линия не в состоянии разделить классы А и В. Это справедливо и для произвольного числа признаков и выходных классов.

Такие функции как ИСКЛЮЧАЮЩЕЕ ИЛИ называются *линейно неразделимыми*. Причем таких функций значительное большинство и не существует способа предварительной проверки функции на линейную разделимость.

### 3.2 Многослойный перцептрон

*Многослойным перцептроном* (MLP – Multi layer perceptron) называется перцептрон с количеством слоев больше одного.

Преодоление ограничения линейной разделимости возможно при применении двухслойного перцептрона, получаемого каскадным соединением двух однослойных перцептронов.

На рис. 3.5 представлен двухслойный перцептрон, реализующий функцию ИСКЛЮЧАЮЩЕЕ ИЛИ.

Таблица 3.2 иллюстрирует процесс решения задачи ИСКЛЮЧАЮЩЕГО ИЛИ. В таблице приведена исходная задача ИСКЛЮЧАЮЩЕГО ИЛИ, а также последовательный расчет всех сигналов, проходящих по нейросети (для симметричности нули заменены на минус 1).

Как видно из таблицы, выходной сигнал нейросети  $y$  совпадает с требуемым значением «Выход».

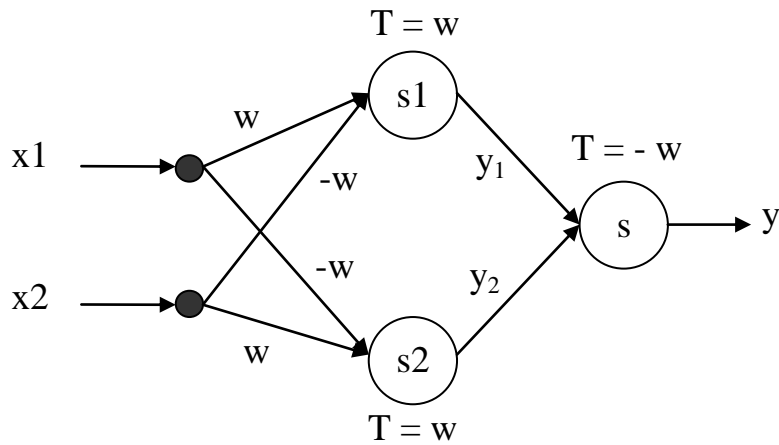


Рисунок 3.5 – Двухслойный персептрон, реализующий функцию ИСКЛЮЧАЮЩЕЕ ИЛИ

Таблица 3.2 – Таблица истинности сети

x1	x2	Выход	s1	s2	y1	y2	s	y
-1	-1	-1	0	0	-1	-1	-2	-1
-1	1	1	-2w	2w	-1	1	0	1
1	-1	1	2w	-2w	1	-1	0	1
1	1	-1	0	0	-1	-1	-2	-1

Решающая поверхность двухслойного персептрона является комбинацией от областей, создаваемых нейронами первого слоя. Вид комбинации определяется параметрами нейрона второго слоя. Двухслойный персептрон может формировать произвольные выпуклые многоугольные решающие области. Число сторон в многоугольнике совпадает с количеством нейронов в первом слое. В зависимости от положения гиперплоскостей область может быть открытой или закрытой.

На рис. 3.6 приведен пример открытой области, сформированной двухслойным персептроном с четырьмя нейронами в первом слое (выходы обозначены  $y_1 \dots y_4$ ). Внутри области выход сети равен единице.

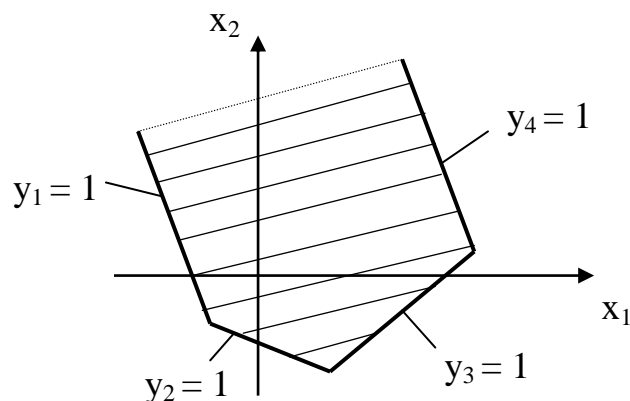


Рисунок 3.6 – Область, сформированная двухслойным персептроном

Трехслойный персептрон (см. рис. 3.7) является более общим. Нейрон третьего слоя принимает в качестве входа набор выпуклых многоугольников, и их логическая комбинация может быть невыпуклой.

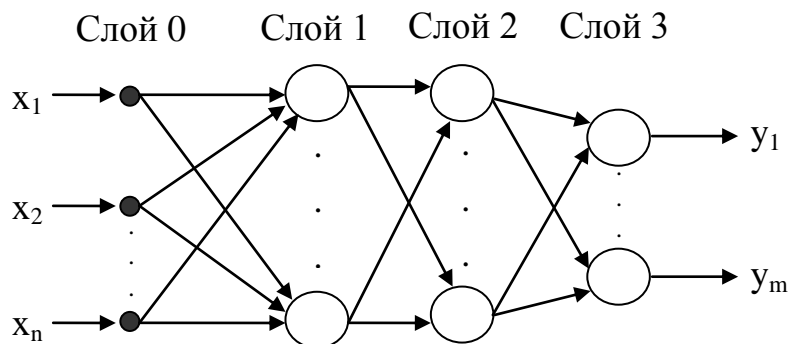


Рисунок 3.7 – Трехслойный персептрон

Первый слой персептрона называется *входным* (на рис. 3.7 – Слой 0), последний – *выходным* (на рис. 3.7 – Слой 3), остальные – *скрытыми*. Входной слой часто не учитывается в общем количестве слоев, поскольку, как уже было отмечено в п. 3.1, нейроны входного слоя не производят вычислений.

На рис. 3.8 приведен пример невыпуклой области, сформированной трехслойным персептроном.

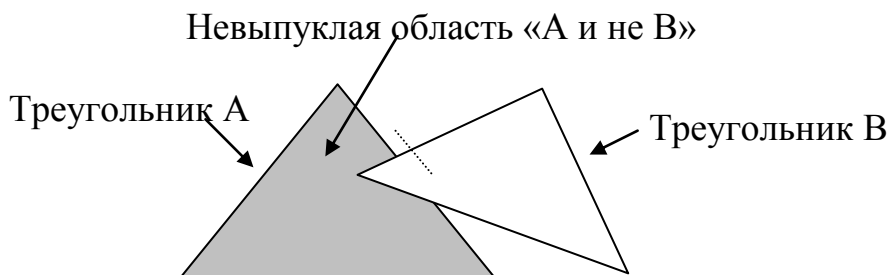


Рисунок 3.8 – Пример области, сформированной трехслойным персептроном

В общем случае многослойный персептрон может иметь любое количество слоев с любым количеством нейронов в каждом слое.

Персептрон с не пороговыми функциями активации нейронов называют *многослойной прямонаправленной нейронной сетью*. По статистике применения нейросетей такие сети занимают до 75 %.

Доказан ряд теорем (А.Н. Колмогоров, В.И. Арнольд (1957 г.); Р. Хехт-Нильсен (1987 г.); Г. Цибенко (1988 г.); К. Хорник, К. Фунахаши (1989 г.)) о том, что любую непрерывную функцию нескольких переменных можно с любой точностью реализовать с помощью трехслойной нейронной сети (с одним скрытым слоем) с нелинейными функциями активации с достаточным количеством нейронов в скрытом слое. Или для функ-

ций, заданных таблично: для любого множества пар  $(X_k, Y_k)$ , где  $Y_k$  – скаляр, существует двухслойная однородная (с одинаковыми функциями активации) нейронная сеть с последовательными связями и с конечным числом нейронов, которая выполняет отображение  $X \rightarrow Y$ .

Согласно теореме М.Н. Стоуна, обобщающей теорему Вейерштрасса о возможности равномерного приближения непрерывных функций многочленами, нелинейное преобразование, применяемое в нейронных сетях, может быть произвольной формы.

Преобразование входного пространства в выходное осуществляется за счет построения аппроксимирующей кусочно-линейной поверхности с помощью поочередного расчета линейных комбинаций и нелинейных преобразований. Таким образом, нейронные сети являются универсальными аппроксиматорами, способными с любой точностью имитировать любой непрерывный автомат.

Применение многослойных персептронов было затруднено из-за отсутствия алгоритма обучения. Однако в 1986 г. Д. Румельхарт, Г. Хинтон и Р. Вильямс разработали *алгоритм обратного распространения ошибки* (back propagation). *Обратное распространение* – это систематический метод для обучения многослойных искусственных нейронных сетей. Процедура обратного распространения применима к сетям с любым числом слоев. Для алгоритма обратного распространения требуется, чтобы активационная функция была всюду дифференцируема. Описание алгоритма обратного распространения ошибки приведено в разделе 4.

*Области применения многослойных персептронов:* распознавание образов, классификация, прогнозирование.

*Преимущества:* возможность аппроксимации функций любого вида.

*Недостатки:* неполнота теории проектирования и настройки нейросетей.

### 3.3 Построение многослойных персептронов

#### 3.3.1 Этапы построения многослойных персептронов

- 1 Предварительная обработка исходных данных;
- 2 Выбор архитектуры нейронной сети;
- 3 Аналитическое определение преобразования, осуществляемого нейронной сетью;
- 4 Формирование функционала оптимизации нейронной сети;
- 5 Выбор метода обучения нейронной сети;
- 6 Выбор начальных условий и обучение нейронной сети;
- 7 Интерпретация полученных результатов;
- 8 Тестирование сети.



### 3.3.2 Предварительная обработка исходных данных

Поскольку компоненты входных векторов имеют различные единицы измерения и диапазоны вариации, необходимо выполнить операцию *масштабирования*, для того чтобы представить все элементы входного сигнала числами одного типа из одного диапазона. Подача на вход нейронной сети немасштабированных данных затрудняет обучение сети и приводит к ошибкам в ее работе так как

1 Работа весовых коэффициентов сети в различных масштабах усложняет начальную инициализацию весов;

2 Весовые коэффициенты нейронов в зависимости от величины дисперсии примут очень большие или очень малые значения, что увеличит время обучения и снизит точность прогнозов;

3 Нейроны входного слоя или окажутся в постоянном насыщении из-за большого усредненного значения совокупности входных данных и малой дисперсии, или будут заторможены из-за малого среднего значения выборки.

Также необходимо масштабирование эталонных выходных сигналов, потому что обычно диапазон выходных значений нейронов лежит в ограниченном интервале и нейронная сеть в принципе не сможет выдать ответ в реальном масштабе данных.

Поскольку нейронные сети анализируют не абсолютные значения входных сигналов, а их изменения, то для повышения различимости сигналов кроме масштабирования при предварительной обработке данных следует выполнять *сдвиг входных данных*. Сдвиг обеспечивается при идентификации границ диапазона изменения признака и рассмотрении их в качестве границ входного диапазона. Операции сдвига и масштабирования вместе представляют собой *нормализацию* входных данных.

Для нормализации данных в заданный диапазон применяют минимаксное преобразование в гиперкуб:

$$x_i^H = \frac{(x_i - x_{\min})(b - a)}{x_{\max} - x_{\min}} + a, \quad (3.1)$$

где  $i$  – индекс примера в массиве исходных данных;

$x_i^H$  – нормализованное  $i$ -е значение сигнала;

$x_i$  – исходное  $i$ -е значение сигнала;

$x_{\min}$ ,  $x_{\max}$  – минимальное и максимальное значение сигнала в массиве исходных данных;

$a$  и  $b$  – соответственно нижняя и верхняя границы нормализованного диапазона.

При отсутствии жестких ограничений на диапазон значений входного признака может быть выполнена нормализация, дающая нулевое сред-

нее и единичную дисперсию нормализованной величине, что ускоряет обучение:

$$x_i^H = \frac{x_i - M(x)}{\sigma(x)}, \quad (3.2)$$

где  $M(x)$  – выборочное среднее компонента вектора исходных данных;

$\sigma(x)$  – среднее квадратическое отклонение данных компонента.

После нормализации данных формируется *обучающая выборка*, в которую выделяется часть примеров из общего множества исходных данных. Как правило, в обучающую выборку включается 70-80 % примеров исходного множества данных, но не менее нескольких десятков (лучше нескольких сотен). В общем случае примеры отбираются случайным образом. Однако поскольку обычно интерполяция осуществляется точнее, чем экстраполяция, то для повышения точности прогнозов сетей обучающую выборку следует формировать из *реперных (опорных)* примеров, характеризующих особенности функции, которую должна будет реализовывать обученная нейронная сеть. К таким примерам относятся все примеры, у которых значение хотя бы одного компонента лежит на границе диапазона изменения этого компонента.

Также иногда из общего множества исходных данных выделяется контрольная выборка, которая используется во время обучения нейронной сети для предотвращения эффекта переобучения сети.

Помимо нормализации и отбора примеров для обучающего множества в предварительную обработку данных входят *рандомизация данных* и *анализ обучающего множества на противоречивость*.

Рандомизация предполагает расположение примеров в случайном порядке внутри обучающего множества для ликвидации шума, появляющегося при упорядочивании примеров, когда их порядок не соответствует закономерностям, характерным для описываемого процесса.

Непротиворечивость обучающей выборки означает, что для двух одинаковых входных векторов должен существовать только один выходной вектор. Противоречивые примеры могут появляться вследствие погрешностей в исходных данных. Поэтому такие примеры следует исключать из обработки.

### **3.3.3 Выбор архитектуры нейронной сети**

Поскольку вычислительная мощность нейросетей зависит от числа связей между нейронами, то при использовании сетей с обратными и перекрестными связями для аппроксимации одной и той же функции требуется

меньшее число нейронов в скрытом слое, чем при использовании сетей с прямыми связями.

Однако скорость сходимости при обучении сетей с обратными и перекрестными связями намного ниже, чем при обучении сетей с прямыми связями. Для сетей с обратными связями также необходимо решать вопросы динамической устойчивости при их обучении и функционировании. Поэтому нейронные сети с обратными и перекрестными связями целесообразно применять при их аппаратной реализации, а сети с прямыми связями при использовании программной эмуляции.

Экспериментально доказано, что качество работы нейронной сети с прямыми связями монотонно возрастает при увеличении числа слоев и числа нейронов в каждом слое.

Определение оптимального числа нейронов в скрытом слое по заданной обучающей выборке в настоящее время не имеет математического решения (под оптимальным здесь понимается минимально необходимое для решения задачи число нейронов).

При числе нейронов ниже оптимального

- 1 Сеть не обучится и ошибка при работе сети останется большой;
- 2 Сеть не сможет аппроксимировать резкие колебания исходной функции.

Чем больше количество нейронов, тем выше аппроксимирующие возможности сети. Однако избыточное число нейронов приводит

- 1 К усложнению и замедлению обучения нейронной сети;
- 2 К переобучению, при котором нейросеть будет отображать несущественные детали в изучаемой зависимости, например, шум или ошибочные данные;
- 3 К снижению обобщающих свойств сети.

Количество нейронов скрытого слоя связано

- 1 Со сложностью задачи;
- 2 С количеством данных для обучения;
- 3 С требуемым количеством входов и выходов сети.

Оценить число нейронов в скрытых слоях можно с помощью неравенства для оценки числа весовых коэффициентов в сети необходимого для освоения заданного числа примеров в обучающей выборке:

$$\frac{N_y N_p}{1 + \log_2 N_p} \leq N_w \leq N_y \left( 1 + \frac{N_p}{N_x} \right) (N_x + N_y + 1) + N_y, \quad (3.3)$$

где  $N_w$  – число весов в сети;

$N_p$  – число элементов обучающей выборки;

$N_x$  и  $N_y$  – соответственно размерность входного и выходного сигнала.

Тогда число нейронов ( $N_H$ ) в двухслойной сети можно определить по формуле

$$N_h = \frac{N_w}{N_x + N_y}. \quad (3.4)$$

Подставляя в формулу (3.4) граничные значения  $N_w$ , рассчитанные по формуле (3.3), получаем минимальное ( $N_{h_{\min}}^w$ ) и максимальное ( $N_{h_{\max}}^w$ ) число нейронов в скрытом слое сети.

Помимо объема обучающей выборки на размер сети также влияет сложность решаемой задачи. Сложность задачи соответствует сложности аппроксимации исходной функции нейронной сетью и определяется структурой и значениями исходных данных. Оценить сложность аппроксимации таблично заданной функции (обучающей выборки) можно с помощью выборочной оценки константы Липшица, вычисляемой по следующей формуле:

$$L_{\{\bar{x}, \bar{y}\}} = \max_{i \neq j} \frac{d_E(\bar{y}^i, \bar{y}^j)}{d_E(\bar{x}^i, \bar{x}^j)}, \quad (3.5)$$

где  $i, j = 1 \dots n$  – индексы примеров в массиве исходных данных;  
 $n$  – количество примеров в исходных данных;  
 $\bar{x}^i$  – векторы входных сигналов нейросети;  
 $\bar{y}^i$  – векторы требуемых выходных сигналов нейросети;  
 $d_E(\bar{y}^i, \bar{y}^j)$  – евклидово расстояние между векторами выходных сигналов;  
 $d_E(\bar{x}^i, \bar{x}^j)$  – евклидово расстояние между векторами входных сигналов.

В качестве евклидова расстояния между векторами входных (выходных) сигналов используется норма, вычисляемая по формуле

$$d_E(\bar{x}^i, \bar{x}^j) = \|\bar{x}^i - \bar{x}^j\| = \sqrt{\sum_{k=1}^K (x_k^i - x_k^j)^2}, \quad (3.6)$$

где  $k = 1 \dots K$  – индекс компонента входного вектора;  
 $K$  – число компонент входного вектора.  
Тогда, исходя из формул (3.5) и (3.6), выборочная оценка константы Липшица будет определяться по формуле

$$L_{\{x,y\}} = \max_{i \neq j} \sqrt{\frac{\sum_{k=1}^K (y_k^i - y_k^j)^2}{\sum_{k=1}^K (x_k^i - x_k^j)^2}}. \quad (3.7)$$

Выборочная оценка константы Липшица позволяет оценить скачки аппроксимируемой функции. Чем более резкие скачки делает функция, тем сложнее ее аппроксимировать с заданной точностью и, соответственно, тем более мощная нейросеть требуется.

Определив выборочную оценку константы Липшица, можно проверить возможность нейронной сети заданной архитектуры аппроксимировать предложенную обучающую выборку. Для этого необходимо оценить константу Липшица нейросети и сравнить ее с выборочной оценкой константы Липшица.

Например, константа Липшица гомогенной прямонаправленной нейронной сети с функциями активации вида «экспоненциальная сигмоида» определяется по формуле

$$L_s \leq c^k \sqrt{N_x N_y} \prod_{i=1}^{k-1} N_{H_i}, \quad (3.8)$$

где  $c$  – параметр активационной функции (см. формулу (2.5));

$k$  – число слоев сети (без учета входного слоя);

$N_x$  – размерность входного сигнала;

$N_y$  – размерность выходного сигнала;

$N_{H_i}$  – число нейронов в  $i$ -м слое.

Чтобы нейронная сеть смогла аппроксимировать исходную функцию, необходимо выполнения условия  $L_{\{x,y\}} \leq L_s$ . Тогда для двухслойной сети, исходя из формул (3.7) и (3.8), число скрытых нейронов не должно быть меньше

$$N_{H_{\min}}^L = \frac{L_{\{x,y\}}}{c^k \sqrt{N_x N_y}}. \quad (3.9)$$

Таким образом, с учетом формул (3.8) и (3.9) минимальное число нейронов скрытого слоя двухслойной гомогенной прямонаправленной нейронной сети с функциями активации вида «экспоненциальная сигмоида» составит

$$N_{H_{\min}} = \max\{N_{H_{\min}}^L, N_{H_{\min}}^w\}. \quad (3.10)$$

Уточнить число нейронов в скрытом слое можно в процессе настройки нейронных сетей. Способы настройки размеров нейросетей можно разделить на две группы: *конструктивные алгоритмы* и *алгоритмы сокращения*.

#### *Алгоритмы сокращения.*

В основе алгоритмов сокращения лежит принцип постепенного удаления из нейронной сети синапсов и нейронов. В начале работы алгоритма обучения с сокращением число нейронов в скрытых слоях сети заведомо избыточно.

Существуют два подхода к реализации алгоритмов сокращения:

1 *Метод штрафных функций*, при котором в целевую функцию алгоритма обучения вводится штраф за то, что значения синаптических весов отличны от нуля. Пример штрафа:

$$C = \sum_{i=1}^n \sum_{j=1}^m w_{ij}^2,$$

где  $w_{ij}$  –  $j$ -й синаптический вес  $i$ -го нейрона;

$n$  – число нейронов;

$m$  – размерность входного сигнала нейронов.

2 *Метод проекций*, при котором синаптический вес обнуляется, если его значение попало в заданный диапазон:

$$w_{ij} = \begin{cases} 0, & -\varepsilon \leq w_{ij} \leq \varepsilon \\ w_{ij}, & \text{иначе} \end{cases},$$

где  $\varepsilon$  – некоторая константа.

Алгоритмы сокращения имеют, по крайней мере, два недостатка:

1 Нет методики определения числа нейронов скрытых слоев, которое является избыточным, поэтому перед началом работы алгоритма нужно угадать это число. В качестве максимального можно принять число  $N_{\max}^w$  (формулы (3.3) и (3.4)). Однако поскольку определение числа  $N_{\max}^w$  базируется только на числе примеров в обучающей выборке, то его значение может оказаться недостаточно большим;

2 В процессе работы алгоритма сеть содержит избыточное число нейронов, поэтому обучение идет медленно.

#### *Конструктивные алгоритмы.*

Наиболее простым вариантом конструктивного подхода является методика, в соответствии с которой первоначально число нейронов принимается равным минимальному количеству (например, рассчитанному по формуле (3.10)). В случае неудачного обучения в скрытый слой добавляется один нейрон, весовым коэффициентам которого присваиваются случайные значения. При этом сеть в основном сохраняет навыки, полученные на

предыдущих этапах обучения, хотя и наблюдается небольшое увеличение значения функции ошибки. После проведения повторного обучения значение функции ошибки снижается до величины меньшей, чем была достигнута сетью с меньшим числом нейронов. Добавление нейронов продолжается до тех пор, пока качество работы нейросети не достигнет требуемого значения.

Более сложные конструктивные алгоритмы сохраняют навыки, приобретенные сетью до увеличения числа нейронов. Такие алгоритмы основаны на том, что значения синаптических весов нового нейрона определяются путем *расщепления* (*splitting*) одного из старых нейронов.

Рис. 3.9 иллюстрирует суть алгоритма расщепления.

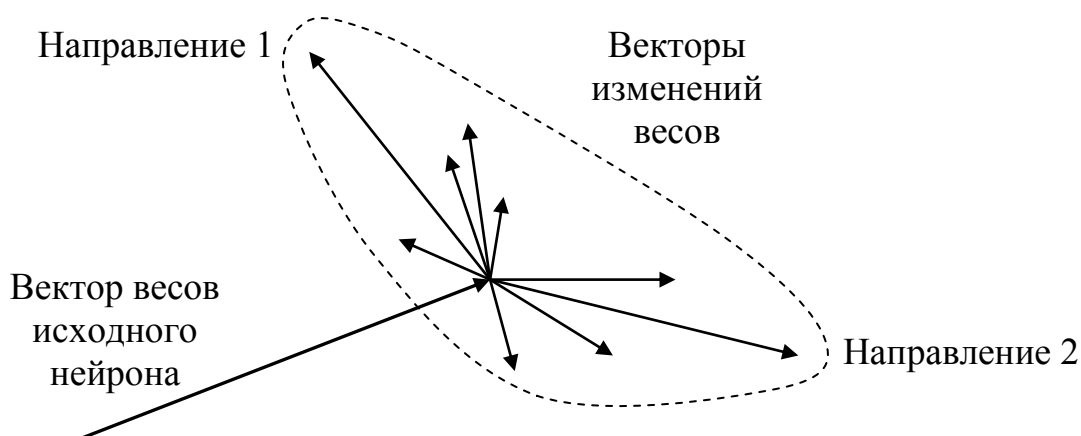


Рисунок 3.9 – Иллюстрация алгоритма расщепления нейрона

На рис. 3.9 показан вектор весов исходного нейрона скрытого слоя на некотором шаге обучения и векторы изменения весов, соответствующие отдельным обучающим примерам (т.е. векторы, показывающие необходимые изменения весов для решения отдельного примера обучающей выборки).

Векторы изменений имеют два преимущественных направления (Направление 1 и Направление 2) и образуют в пространстве область, существенно отличающуюся от сферической. Суть алгоритма заключается в выявлении и расщеплении таких нейронов. В результате расщепления вместо одного исходного нейрона в сети оказывается два новых нейрона, каждый из которых образуется путем сложения вектора весов исходного нейрона и одного из векторов изменений весов преимущественного направления.

Наличие в сети нейронов, векторы изменений которых имеют два преимущественных направления, приводит к осцилляциям при обучении классическим методом обратного распространения ошибки. При обучении методом с интегральной функцией ошибки наличие таких нейронов приводит к попаданию алгоритма обучения в локальный минимум с большим значением ошибки. Поэтому расщепление таких нейронов (удаление их из сети) облегчает процесс обучения нейросети.

Недостатком алгоритма расщепления является экспоненциальный рост объема вычислений при увеличении размерности сети.

### **3.3.4 Аналитическое определение преобразования, осуществляемого нейронной сетью**

Аппроксимируемая функция представляется в нейросетевом логическом базисе базовыми функциями  $\varphi$  (активационными функциями) с множеством коэффициентов  $w$  (весовые коэффициенты и нейронные смещения), которые настраиваются в процессе поиска наилучшей аппроксимации. На этом этапе записывается функция, определяющая зависимость выходов сети от ее входов. Т. е. определяется формула аналитического преобразования входного пространства признаков в выходное пространство классов.

Для вывода аналитического преобразования, осуществляемого нейронной сетью, необходимо определить выходные сигналы каждого нейрона сети. Поскольку нейроны функционируют согласно формуле (2.1) и имеют активационные функции (2.5) – (2.9), то их выходные сигналы рассчитываются по следующим формулам:

$$Y_i^k = \begin{cases} \varphi \left( \sum_{j=1}^{N_{k-1}} w_{ij}^k x_j + T_i^k \right), k = 1 \\ \varphi \left( \sum_{j=1}^{N_{k-1}} w_{ij}^k Y_j^{k-1} + T_i^k \right), k > 1, \end{cases} \quad (3.11)$$

где  $k$  – номер слоя;  
 $i = 1 \dots N_k$  – номер нейрона в слое;  
 $j = 1 \dots N_{k-1}$  – номер входа нейрона;  
 $N_k$  – число нейронов в  $k$ -м слое (при  $k = 1$ ,  $N_0$  – число входных сигналов);

$Y_i^k$  – выходной сигнал  $i$ -го нейрона  $k$ -го слоя;

$x_j$  –  $j$ -й входной сигнал;

$w_{ij}^k$  –  $j$ -й весовой коэффициент  $i$ -го нейрона  $k$ -го слоя;

$\varphi$  – активационная функция;

$T_i^k$  – нейронное смещение  $i$ -го нейрона  $k$ -го слоя.

Или, преобразуя (3.11), общее выражение аппроксимирующей функции нейронной сети с последовательными связями можно представить следующим образом:



$$y(x) = \dots \varphi \left( \sum_{k_2=1}^{N_2} w_{k_3 k_2}^3 \varphi \left( \underbrace{\sum_{k_1=1}^{N_1} w_{k_2 k_1}^2 \varphi \left( \underbrace{\sum_{k_0=1}^{N_0} w_{k_1 k_0}^1 x_{k_0} + T_{k_1}^1}_{\text{1-й слой}} \right) + T_{k_2}^2}_{\text{2-й слой}} \right) + T_{k_3}^3 \right), \quad (3.12)$$

3-й слой

где  $k_0$  – номер входного сигнала;

$k_1, k_2, k_3, \dots$  – номера нейронов в слоях номер 1, 2, 3, ...

Приведенные выражения описывают нейронную сеть с одним выходом. Если сеть имеет больше выходов, то остальные ее выходные сигналы рассчитываются аналогично. Отличие будет заключаться лишь в параметрах выходного нейрона.

### 3.3.5 Формирование функционала оптимизации

Значения весов и смещений сети должны соответствовать минимальной разнице между значением в обучающей выборке и значением, рассчитанным нейронной сетью для соответствующего примера. Таким образом, задача обучения нейронной сети является задачей минимизации множества функций многих переменных, где под переменными понимаются обучаемые параметры сети, а под функциями – оценки решения сетью отдельных примеров. Т. е. многокритериальная задача оптимизации сети для всего обучающего множества рассматривается как набор однокритериальных для отдельно взятых примеров.

Это может приводить к, так называемой, *временной неустойчивости*, которая возникает из-за того, что при оптимизации одной из функции значения оставшихся функций не контролируются. Вследствие этого величины ошибок уже оптимизированных функций могут возрасти. Все это снижает скорость обучения и в ряде случаев алгоритм может не сходиться.

Для преодоления временной неустойчивости необходимо, чтобы сети предъявлялись все векторы обучающего множества, прежде чем выполнялась коррекция весов. Т. е. необходима выработка единой оценки всего обучающего множества. Такая оценка называется *функционалом оптимизации* или *функцией ошибки* нейронной сети.

В качестве функций ошибки используются:

- 1 Суммарная квадратичная ошибка

$$Z = \sum_{i=1}^n \sum_{j=1}^m (Y_{ij}^{об} - Y_{ij}^c)^2 \rightarrow \min, \quad (3.13)$$

где  $i = 1 \dots n$  – номер выхода сети;

$j = 1 \dots m$  – номер примера в обучающей выборке;

$Y_{ij}^{об}$ ,  $Y_{ij}^c$  – соответственно значения выхода в обучающей выборке и выхода рассчитанного сетью (по формуле (3.12)).

2 Средняя квадратичная ошибка (здесь и далее для упрощения формулы рассматривается сеть с одним выходом)

$$Z = \sqrt{\frac{\sum_{j=1}^m (Y_j^{об} - Y_j^c)^2}{m}} \rightarrow \min. \quad (3.14)$$

3 Сумма модулей

$$Z = \sum_{j=1}^m |Y_j^{об} - Y_j^c| \rightarrow \min. \quad (3.15)$$

4 Максимум модуля

$$Z = \max_j |Y_j^{об} - Y_j^c| \rightarrow \min. \quad (3.16)$$

5 Суммарная квадратичная ошибка с люфтом

$$Z = \sum_{j=1}^m P(\Delta) \rightarrow \min, \quad (3.17)$$

где

$$\Delta = \frac{Y_j^{об} - Y_j^c}{\varepsilon};$$

$$P(\Delta) = \begin{cases} (|\Delta| - 1)^2, & |\Delta| \geq 1, \\ 0, & |\Delta| < 1. \end{cases}$$

Оценки с люфтом позволяют прекращать оптимизацию, когда достигнута заданная величина абсолютной ошибки  $\varepsilon$ .

Функционал оптимизации принимает неотрицательные значения. Обученные состояния нейросети являются точками минимума функционала оптимизации.

При эмуляции нейросети на ПЭВМ учет всех примеров обучающего множества при его большой размерности может быть невозможен из-за ограничения по объему памяти. Эта проблема может быть преодолена при использовании *пакетной коррекции* (или *многостраничного обучения*), когда обучающее множество разбивается на ряд подмножеств с меньшим числом примеров, и затем нейросеть последовательно обучается на этих подмножествах.

### **3.3.6 Выбор метода настройки нейросети**

Под настройкой или обучением нейронной сети понимается нахождение таких значений весов и смещений нейронов сети, при которых сеть способна решить задачу с заданной точностью. Другими словами необходимо минимизировать функционал оптимизации, т.е. решить классическую задачу оптимизации.

Для настройки нейросетей разработаны специализированные методы (см. главу 6). Также хорошо работают многие методы теории оптимизации, часто лучше, чем специализированные. Выбор метода определяется особенностями решаемой задачи: типом нейросети, числом переменных и др.

Для определения момента остановки процесса настройки нейронной сети можно использовать следующие критерии:

- 1 Достижение заданной величины сходимости. Величина сходимости задается априорно и обычно принадлежит диапазону  $10^{-3} \dots 10^{-5}$ ;
- 2 Решение всех примеров обучающей выборки с заданной точностью;
- 3 Достижение значением функционала оптимизации заданной величины.

Как правило при обучении нейронных сетей следует опираться на основные положения теории оптимизации.

### **3.3.7 Выбор начальных условий при настройке нейронной сети**

Наличие большого числа экстремумов функционала оптимизации приводит к необходимости внесения элементов случайности в процедуру настройки нейронной сети. Для нахождения глобального минимума функционала оптимизации или наименьшего среди всех найденных локальных минимумов необходимо проведение ряда оптимизационных расчетов при различных начальных условиях (значениях весов и смещений).

Практика показывает, что начальные значения весов и смещений должны быть небольшими по величине (обычно рекомендуется диапазон  $[-0,1; 0,1]$ ). Это гарантирует, что в сети не произойдет насыщения нейронов большими значениями весов. Закон распределения случайной величины обычно принимается равномерным.

### **3.3.8 Интерпретация полученных результатов**

Если исходные данные были подвержены процедуре нормализации, то после окончания процесса настройки нейросети выходные сигналы должны преобразовываться в диапазон реальных значений. Например, при использовании для нормализации минимаксного преобразования (формула (3.1)), интерпретация выполняется по формуле

$$x_i = \frac{(x_i^H - a)(x_{\max} - x_{\min})}{b - a} + x_{\min}. \quad (3.18)$$

### **3.3.9 Тестирование сети**

Для проверки навыков, приобретенных сетями в процессе обучения, необходимо провести их тестирование на тестовой выборке. Тестовое множество должно иметь формат данных и структуру аналогичную обучающему множеству и также состоять из пар вход-желаемый выход. Обучающая и тестовая выборки не должны пересекаться. Число примеров в тестовой выборке должно составлять 20 – 30 % от всей совокупности данных, но не менее 10 – 20.

В общем случае примеры отбираются случайным образом с равномерным распределением внутри области проектирования. В частных случаях в тестовую выборку могут дополнительно вводиться примеры, представляющие больший интерес для решаемой задачи.

### **3.3.10 Определение показателей значимости сигналов и элементов нейронной сети**

В результате настройки нейросети может оказаться, что один или несколько входных сигналов не будут оказывать существенного влияния на точность расчета выходных сигналов. Такая ситуация может возникнуть, например, из-за переоценки степени влияния факторов на результирующие показатели при постановке задачи. В этом случае такие входы можно будет исключить из рассмотрения, что позволит сократить объемы привлекаемой для работы нейросети информации и повысит скорость расчетов. Сокращение состава входных сигналов следует рассматривать как оптимизацию структуры уже построенной и обученной нейросети. Т. е. этот этап работы с нейросетью выполняется после завершения ее построения и может отсутствовать.

Для оптимизации структуры входов нейросети можно использовать оценки показателей значимости входных сигналов.

*Показателем значимости*  $i$ -го входного сигнала при решении  $k$ -го примера является величина, которая показывает, насколько изменится значение функции оценки решения нейронной сетью  $k$ -го примера, если текущее значение входного сигнала  $x_i^k$  заменить выделенным значением  $x_i^{k*}$ . В качестве выделенного значения  $x_i^{k*}$  используются средние значения каждого входного сигнала.

Для каждого примера  $x^k$  значимость сигнала  $x_i^k$  оценивается по следующей формуле:

$$\chi_i^k = \left| \frac{\partial Z(x^k)}{\partial x_i} (x_i^k - x_i^{k*}) \right|, \quad (3.19)$$

где  $Z(x^k)$  – оценка решения нейронной сетью  $k$ -го примера.

Такая оценка предполагает использование в качестве функции ошибки функцию вида (3.13). Если используются специализированные функции оценки, то на этапе вычисления значимости необходимо возвращаться к оценке (3.13).

Если для нормализации данных использовалась формула (3.2), предполагающая получение нулевых средних, то для всех входных сигналов  $x_i^{k*} = 0$ , что эквивалентно исключению сигнала из сети. Тогда формула (3.19) примет вид

$$\chi_i^k = \left| \frac{\partial Z(x^k)}{\partial x_i} x_i^k \right|. \quad (3.20)$$

Как отмечалось в п. 3.3.2, нейронные сети анализируют изменения входных сигналов. Поэтому если в множестве исходных данных окажется, что какой-либо сигнал имеет во всех примерах имеет одинаковое значение, то такой сигнал является незначимым.

Формулы (3.19), (3.20) оценивают значимость сигнала для решения отдельных примеров. Оценка значимости сигнала для всего обучающего множества производится с использованием нормы в виде суммы оценок решения нейросетью примеров обучающего множества или максимума среди полученных оценок. В первом случае общая значимость сигнала будет определяться по формуле

$$\chi_i = \sum_{k=1}^{N_p} \chi_i^k, \quad (3.21)$$

где  $N_p$  – число примеров обучающего множества.  
Во втором случае

$$\chi_i = \max_k \chi_i^k. \quad (3.22)$$

Оценка (3.21) является более объективной в случае присутствия шума в исходных данных, поскольку в этом случае расчет значимости производится с учетом всех примеров обучающей выборки, что приводит к усреднению погрешности определения значимости отдельных примеров, снижая тем самым ее максимальный уровень.

Значимость синапсов нейросети можно оценивать, удаляя их из сети и вычисляя изменение значения функции ошибки. Если удаление синапса не увеличивает погрешность работы сети или увеличивает незначительно, то такой синапс можно удалить из сети окончательно. Также можно удалять нулевые и близкие к нулю синапсы, поскольку нулевой синапс эквивалентен отсутствию связи между нейронами.

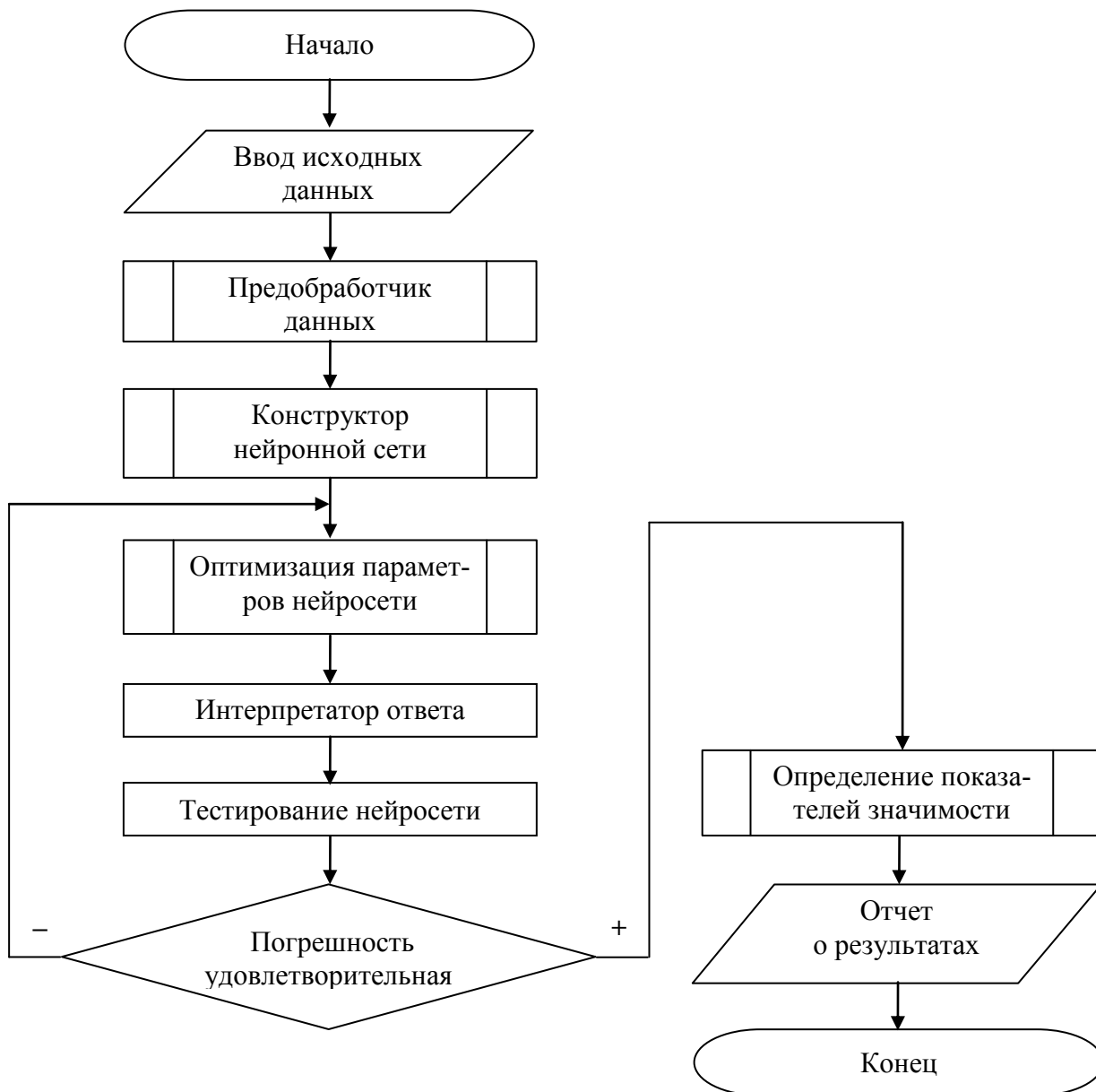
Удаление у сети малозначимых входов и синапсов может привести к удалению из сети нейрона. Если у нейрона нет ни одного входного сигнала, то такой нейрон можно исключить из нейросети, поскольку он не влияет на ее выходной сигнал. Удалив нейрон, можно удалить и синапсы других нейронов, принимающих сигнал от удаленного нейрона. И наоборот, если удалены все синапсы, по которым нейрон рассылал свой выходной сигнал, то можно удалить этот нейрон со всеми его синапсами.

Следует отметить, что сокращение элементов нейронной сети важно при ее аппаратной реализации и не столь существенно при программных эмуляциях.

### ***3.3.11 Алгоритмы построения многослойных прямонаправленных нейронных сетей***

Обобщая теоретические сведения по проектированию нейросетей, изложенные в пунктах 3.3.1 – 3.3.10, представим процесс построения многослойных прямонаправленных нейронных сетей в виде блок-схем алгоритмов.

Проектирование многослойной прямонаправленной нейросети проводится в соответствии с алгоритмом, блок-схема которого приведена на рис. 3.10.



*Рисунок 3.10 – Блок-схема алгоритма проектирования многослойной прямонаправленной нейронной сети*

Предобработчик выполняет подготовку исходных данных для работы нейронной сети. В режиме обучения работа предобработчика осуществляется по алгоритму, блок-схема которого приведена на рис. 3.11.



Рисунок 3.11 – Блок-схема алгоритма работы предобработчика в режиме обучения

В приведенной блок-схеме  $N_p$  – число примеров в исходном множестве данных. Нормализация входных и выходных сигналов осуществляется по формулам (3.1) – (3.2). Также согласно этим формулам функционирует предобработчик в режиме эксплуатации нейронной сети.



Конструктор нейронной сети работает в соответствии с алгоритмом, блок-схема которого приведена на рис. 3.12.



*Рисунок 3.12 – Блок-схема алгоритма работы конструктора нейронной сети*

При оценке числа скрытых нейронов используются формулы (3.4) и (3.10).

На рис. 3.13 представлена блок-схема алгоритма оптимизации параметров нейронной сети, соответствующего конструктивному подходу формирования ее структуры.

Удовлетворительным признается решение, при котором обеспечивается решение всех примеров обучающей выборки с заданной точностью. В случае неудовлетворительного решения выполняется повторный спуск из новой стартовой точки для нахождения очередного локального минимума функционала оптимизации.

Поиск экстремумов прекращается, когда в результате  $s_{\max}$  спусков ( $s_{\max} = 10 \dots 15$ ) не удастся найти ни одного нового экстремума. В этом случае число скрытых нейронов признается недостаточным для аппроксимации исходных данных с заданной точностью и в нейросеть вводится дополнительный скрытый нейрон.

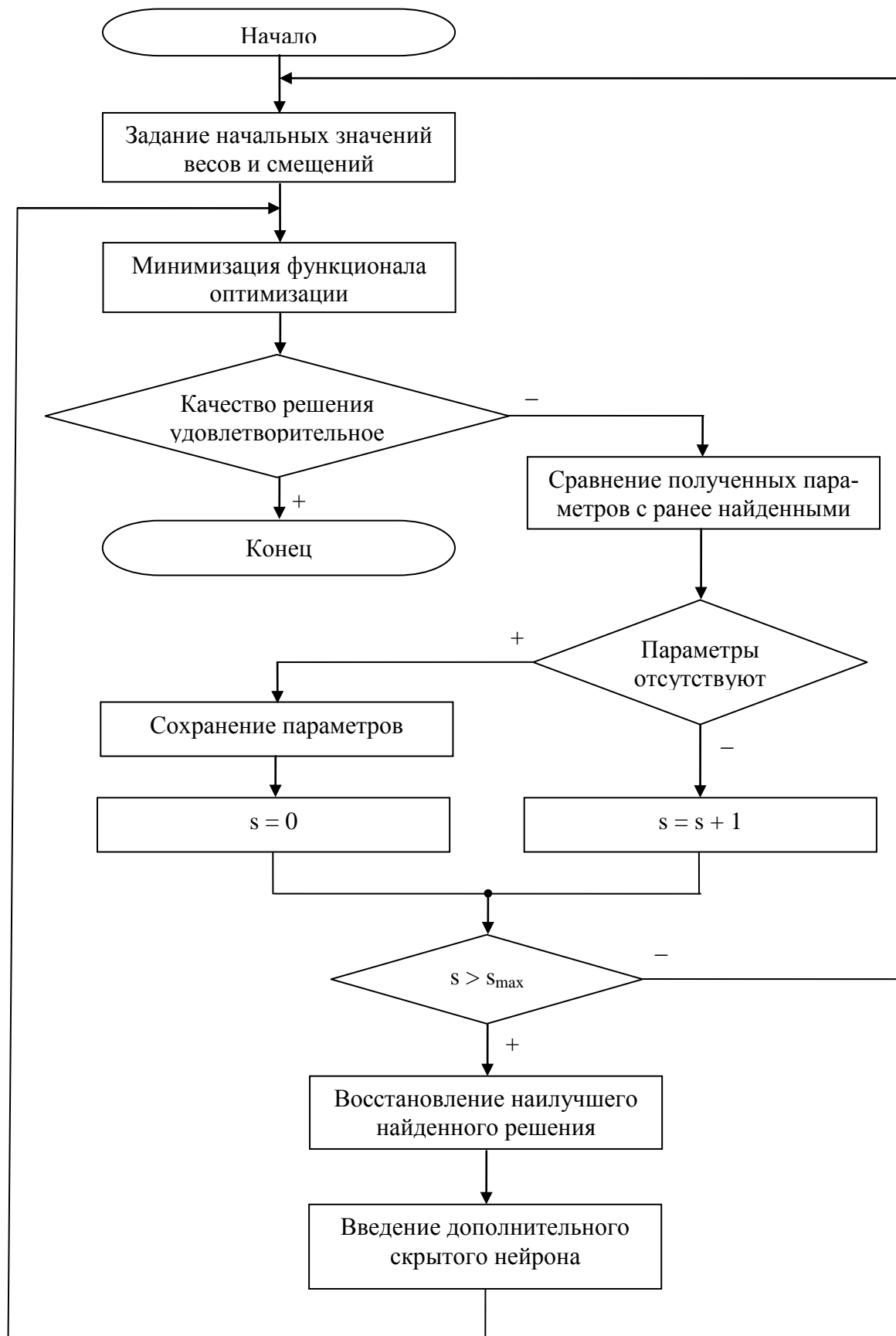


Рисунок 3.13 – Блок-схема алгоритма оптимизации параметров нейронной сети

Определение показателей значимости входных сигналов осуществляется в соответствии с алгоритмом, блок-схема которого приведена на рис. 3.14.

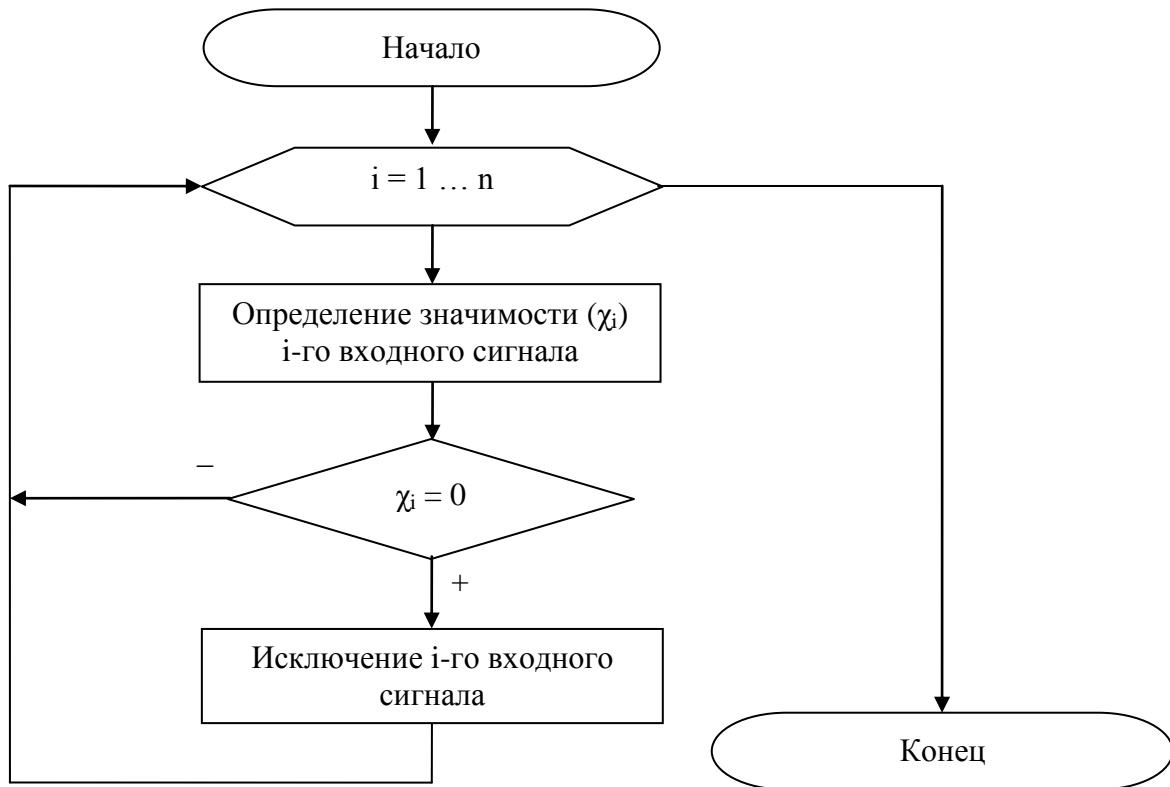


Рисунок 3.14 – Блок-схема алгоритма оценки показателей значимости входных сигналов

В случае нулевой значимости фактора он может быть исключен из состава входных сигналов сети (на соответствующий вход подается 0). Аналогично можно исключать другие элементы нейросети.

Интерпретатор ответа преобразует выходные сигналы нейросети в формат исходных данных. Работа интерпретатора ответа в режиме настройки сети не отличается от работы в режиме эксплуатации.

*Контрольные вопросы по теме 3:*

- 1 Какова структура однослойного и многослойного персептронов?
- 2 Приведите алгоритм обучения дискретного персептрона.
- 3 Что такое линейная неразделимость?
- 4 Приведите этапы построения многослойных персептронов.
- 5 Приведите формулы нормализации входных данных.
- 6 Приведите формулы оценки числа нейронов в скрытых слоях.
- 7 Что такое алгоритмы сокращения и конструктивные алгоритмы?
- 8 Приведите используемые функции ошибки.
- 9 Приведите формулы оценки показателей значимости

## 4 МЕТОДЫ ОБУЧЕНИЯ НЕЙРОННЫХ СЕТЕЙ

### 4.1 Метод обучения Хэбба

В 1949 г., основываясь на физиологических и психологических исследованиях, Д. Хэбб выдвинул гипотезу о том, каким образом может обучаться набор биологических нейронов. Его теория предполагает только локальное взаимодействие между нейронами при отсутствии глобального учителя. Следовательно, обучение является неуправляемым.

Хэбб предположил, что синаптическое соединение двух нейронов усиливается, если оба эти нейрона возбуждены. Это можно представить как усиление синапса в соответствии с корреляцией уровней возбужденных нейронов, соединяемых данным синапсом. По этой причине алгоритм обучения Хэбба иногда называется *корреляционным алгоритмом*.

Идея алгоритма выражается следующим равенством:

$$w_{ij}(t+1) = w_{ij}(t) + Y_i(t)Y_j(t), \quad (4.1)$$

где  $w_{ij}(t)$  – вес синапса от нейрона  $i$  к нейрону  $j$  в момент времени  $t$ ;

$Y_i(t)$  – уровень возбуждения пресинаптического нейрона в момент времени  $t$ ;

$Y_j(t)$  – уровень возбуждения постсинаптического нейрона в момент времени  $t$ .

Различают *метод сигнального обучения Хэбба* и *метод дифференциального обучения Хэбба*.

Метод сигнального обучения Хэбба реализуется с помощью правила (6.1). Метод дифференциального обучения Хэбба, использует следующее равенство:

$$w_{ij}(t+1) = w_{ij}(t) + [Y_i(t) - Y_i(t-1)][Y_j(t) - Y_j(t-1)].$$

Концепция Хэбба отвечает на вопрос, каким образом обучение может проводиться без учителя. В методе Хэбба обучение является исключительно локальным явлением, охватывающим только два нейрона и соединяющий их синапс. Т. е. не требуется глобальной системы обратной связи для развития нейронных образований.

Однако последующее использование метода Хэбба для обучения нейросетей показало ограниченность метода, т.к. некоторые задачи не могут быть решены с его помощью.

## 4.2 Метод обучения Уидроу-Хоффа (дельта-правило)

В 1959 г. Б. Уидроу и М. Е. Хофф расширили алгоритм обучения дискретного персептрона на случай непрерывных выходов, используя сигмоидальную функцию активации. Их первая модель – Адалин – имеет один выходной нейрон, более поздняя модель – Мадалин – расширяет ее на случай со многими выходными нейронами.

Шаг 2 алгоритма обучения персептрона может быть сформулирован в обобщенной форме с помощью введения величины  $\delta$ , которая равна разности между требуемым или целевым выходом  $T$  и реальным выходом  $Y$

$$\delta = (T - Y).$$

Случай, когда  $\delta = 0$ , соответствует шагу 2а, когда выход правилен и в сети ничего не изменяется. Шаг 2б соответствует случаю  $\delta > 0$ , а шаг 2в случаю  $\delta < 0$ .

В любом из этих случаев персептронный алгоритм обучения сохраняется, если  $\delta$  умножается на величину каждого входа  $x_i$  и это произведение добавляется к соответствующему весу. С целью обобщения вводится коэффициент «скорости обучения»  $\eta$ , который умножается на  $\delta x_i$ , что позволяет управлять средней величиной изменения весов.

В алгебраической форме записи

$$\begin{aligned}\Delta_i &= \eta \delta x_i, \\ w_i(n+1) &= w_i(n) + \Delta_i,\end{aligned}$$

где  $\Delta_i$  – коррекция, связанная с  $i$ -м входом  $x_i$ ;  $w_i(n+1)$  – значение веса  $i$  после коррекции;

$w_i(n)$  – значение веса  $i$  до коррекции.

Дельта-правило модифицирует веса в соответствии с требуемым и действительными значениями выхода каждой полярности как для непрерывных, так и для бинарных входов и выходов.

*Трудности, возникающие при обучении персептрона:*

1 Трудность проверки условия разделимости для конкретного обучающего множества.

2 Встречаются ситуации, когда входы меняются во времени и могут быть разделимы в один момент времени и неразделимы в другой.

3 Неизвестно, сколько шагов потребуется для обучения сети.

### 4.3 Алгоритм обратного распространения ошибки

Целью обучения сети является такая подстройка ее весов, чтобы приложение некоторого множества входов приводило к требуемому множеству выходов.

Перед началом обучения всем весам должны быть присвоены небольшие начальные значения, выбранные случайным образом. Это гарантирует, что в сети не произойдет насыщения большими значениями весов.

Обучение сети обратного распространения требует выполнения следующих операций:

- 1 Выбрать очередную обучающую пару из обучающего множества; подать входной вектор на вход сети.

- 2 Вычислить выход сети.

- 3 Вычислить разность между выходом сети и требуемым выходом (целевым вектором обучающей пары).

- 4 Подкорректировать веса сети так, чтобы минимизировать ошибку.

- 5 Повторять шаги с 1 по 4 для каждого вектора обучающего множества до тех пор, пока ошибка на всем множестве не достигнет приемлемого уровня.

Операции, выполняемые шагами 1 и 2, сходны с теми, которые выполняются при функционировании уже обученной сети.

На шаге 3 каждый из выходов сети вычитается из соответствующей компоненты целевого вектора, чтобы получить ошибку. Эта ошибка используется на шаге 4 для коррекции весов сети, причем знак и величина изменений весов определяются алгоритмом обучения.

Шаги 1 и 2 можно рассматривать как «проход вперед», так как сигнал распространяется по сети от входа к выходу. Шаги 3, 4 составляют «обратный проход», здесь вычисляемый сигнал ошибки распространяется обратно по сети и используется для подстройки весов.

*Обратный проход.*

*Подстройка весов выходного слоя.* Т. к. для каждого нейрона выходного слоя задано целевое значение, то подстройка весов осуществляется с использованием модифицированного дельта-правила. Для выходов скрытых слоев не имеется целевых значений для сравнения. Поэтому обучение усложняется.

Пусть необходимо обучить вес от нейрона  $p$  в скрытом слое  $j$  к нейрону  $q$  в выходном слое  $k$ . Выход нейрона слоя  $k$ , вычитаясь из целевого значения ( $T$ ), дает сигнал ошибки. Он умножается на производную активационной функции, вычисленную для этого нейрона слоя  $k$ , давая, таким образом, величину  $\delta$ .

Производные некоторых активационных функций можно рассчитать с помощью выходных сигналов нейронов (OUT):

- для экспоненциальной сигмоиды  $F' = \text{OUT}(1 - \text{OUT})$ ;
- для гиперболического тангенса  $F' = 1 - \text{OUT}^2$ .

Тогда, например, для экспоненциальной сигмоиды

$$\delta = \text{OUT}(1 - \text{OUT})(T - \text{OUT}).$$

Затем  $\delta$  умножается на величину  $\text{OUT}$  нейрона  $j$ , из которого выходит рассматриваемый вес. Это произведение в свою очередь умножается на коэффициент скорости обучения  $\eta$  (обычно от 0,01 до 1,0), и результат прибавляется к весу:

$$\begin{aligned}\Delta w_{pqk} &= \eta \delta_{qk} \text{OUT}_{pj}; \\ w_{pqk}(n+1) &= w_{pqk}(n) + \Delta w_{pqk},\end{aligned}$$

где  $w_{pqk}(n)$  – величина веса от нейрона  $p$  в скрытом слое к нейрону  $q$  в выходном слое на шаге  $n$  (до коррекции);

$w_{pqk}(n+1)$  – величина веса на шаге  $n + 1$  (после коррекции);

$\delta_{qk}$  – величина  $\delta$  для нейрона  $q$ , в выходном слое  $k$ ;

$\text{OUT}_{pj}$  – значение выходного сигнала нейрона  $p$  в скрытом слое  $j$ .

Такая же процедура выполняется для каждого веса от нейрона скрытого слоя к нейрону в выходном слое.

*Подстройка весов скрытого слоя.* Рассмотрим один нейрон в скрытом слое, предшествующем выходному слою. При проходе вперед этот нейрон передает свой выходной сигнал нейронам в выходном слое через соединяющие их веса. Во время обучения эти веса функционируют в обратном порядке, пропуская величину  $\delta$  от выходного слоя назад к скрытому слою. Каждый из этих весов умножается на величину  $\delta$  нейрона, к которому он присоединен в выходном слое. Величина  $\delta$ , необходимая для нейрона скрытого слоя, получается суммированием всех таких произведений и умножением на производную активационной функции:

$$\delta_{pj} = \text{OUT}_{pj}(1 - \text{OUT}_{pj}) \left[ \sum_q \delta_{qk} w_{pqk} \right].$$

Когда значение  $\delta$  получено, веса, питающие первый скрытый уровень, могут быть подкорректированы.

Вычисление  $\delta$  и подстройка весов должны быть выполнены для каждого нейрона в данном скрытом слое. Этот процесс повторяется слой за слоем по направлению к входу, пока все веса не будут подкорректированы.

*Достоинства метода:*

- 1 Возможность обучения многослойных нейросетей;
- 2 Простота;
- 3 Невысокие требования к объему памяти.

*Недостатки:*

1 *Неопределенно долгий процесс обучения.* В сложных задачах сеть может вообще не обучиться. Длительное время обучения может быть результатом неоптимального выбора длины шага.

2 *Паралич сети*. В процессе обучения сети значения весов могут в результате коррекции стать очень большими. Это может привести к тому, что все или большинство нейронов будут функционировать при очень больших значениях ОУТ в области, где производная активационной функции очень мала. Так как посылаемая обратно в процессе обучения ошибка пропорциональна этой производной, то процесс обучения может практически замереть. Обычно этого избегают уменьшением размера шага  $\eta$ , но это увеличивает время обучения.

3 *Локальные минимумы*. Обратное распространение использует разновидность градиентного спуска, т. е. осуществляет спуск вниз по поверхности ошибки, непрерывно подстраивая веса в направлении минимума. Сеть может попасть в локальный минимум, когда рядом имеется гораздо более глубокий минимум. В точке локального минимума все направления ведут вверх, и сеть неспособна из него выбраться.

4 *Временная неустойчивость*. Многокритериальная задача оптимизации рассматривается как набор однокритериальных – на каждой итерации происходят изменения значений параметров сети, улучшающие работу лишь с одним примером обучающей выборки. Такой подход существенно уменьшает скорость обучения. Поэтому процесс обучения должен быть таким, чтобы сеть обучалась на всем обучающем множестве без пропусков того, что уже выучено. Для этого необходимые изменения весов должны вычисляться на всем множестве, а не на одиночных примерах.

5 *Трудности в выборе размера шага*. Коррекции весов в алгоритме предполагаются бесконечно малыми, однако это неосуществимо на практике, так как ведет к бесконечному времени обучения. Размер шага должен браться конечным. Если размер шага очень мал, то сходимость будет слишком медленная, если же очень велик, то может возникнуть паралич или постоянная неустойчивость.

## **4.4 Модификации алгоритма обратного распространения ошибки**

### ***4.4.1 Пакетное обратное распространение***

Пакетное обратное распространение модифицирует алгоритм обратного распространения ошибки, в части времени обновления весов. В отличие от обратного распространения, где обновление весов происходит после каждого наблюдения, в пакетном обратном распространении все изменения суммируются, и обновление происходит только после того, как все наблюдения обучающего набора будут использованы, то есть после эпохи обучения.

*Преимущество:* противодействие временной неустойчивости.



#### **4.4.2 Фрагментарное обратное распространение**

Обновления весов происходят через каждые  $n$  наблюдений, где  $0 < n \leq N$  – размер фрагмента,  $N$  – количество примеров в обучающей выборке.

Данный алгоритм является обобщением подхода к проблеме обновления. При  $n = N$  получаем пакетное обратное распространение, а при  $n = 1$  – обычное обратное распространение.

*Преимущество:* возможность управления периодами обновления весов.

#### **4.4.3 Обратное распространение с угасанием весов**

Предложено Вербосом в 1988 г. Оно состоит в уменьшении веса в процессе обучения обратным распространением. Вдобавок к обычному изменению весов значение веса уменьшается на некоторую часть  $d$  старого значения веса. Таким образом, обучение веса от  $i$ -го к  $j$ -му нейрону осуществляется по формуле

$$\Delta w_{ij}(t + 1) = \eta \delta_j \text{OUT}_i - d w_{ij}(t),$$

где  $0 < d < 1$ .

*Преимущество:* в процессе обучения незначимые веса уменьшаются и в конце обучения становятся равными нулю, т.е. происходит упрощение сети.

#### **4.4.4 Быстрое обратное распространение**

Быстрое обратное распространение (Quickpropagation – Qprop) предложено Фелманом в 1988 г. Быстрое обратное распространение предполагает, что поверхность функции ошибки локально является квадратичной и пытается переместиться за один шаг прямо в точку минимума. После вычисления градиента обычным способом обратного распространения коррекция весов производится по следующей формуле:

$$\Delta w_{ij}(t + 1) = \frac{S_{ij}(t + 1)}{S_{ij}(t) - S_{ij}(t + 1)} \Delta w_{ij}(t),$$

где  $\Delta w_{ij}(t + 1)$  – текущее изменение веса между нейронами  $i$  и  $j$ ;

$S_{ij}(t + 1)$  – частная производная функции ошибки по  $w_{ij}$ ;

$S_{ij}(t)$  – предыдущая частная производная функции ошибки по  $w_{ij}$ ;

$\Delta w_{ij}(t)$  – предыдущее изменение веса между узлами  $i$  и  $j$ .

*Преимущество:* ускорение обучения.

#### 4.4.5 Алгоритм Дельта-Дельта с чертой

Алгоритм Дельта-Дельта с чертой (DBD – Delta-Bar-Delta) разработан Якобсом в 1988 г.

Размерности пространства весов могут значительно различаться с точки зрения общей поверхности ошибки. Размер шага обучения для одного веса не всегда подходит в качестве единого шага обучения для всех весов. Более того, этот размер может со временем изменяться.

Якобс предложил ряд эвристик, суть которых в том, что каждый вес должен изменяться в соответствии со своей индивидуальной скоростью обучения. При этом используются предыдущие значения градиента функции.

Изменение веса на последующем шаге:

$$w[i+1] = w[i] + (a[i] + da[i])g[i],$$

где  $w[i]$  – значение веса на шаге  $i$ ;

$a[i]$  – коэффициент скорости обучения на шаге  $i$ ;

$g[i]$  – градиент на шаге  $i$ ;

$da[i]$  – величина изменения скорости обучения на шаге  $i$ .

$$da[i] = \begin{cases} k_1, g_c[i-1]g[i] > 0 \\ -k_2a[i], g_c[i-1]g[i] < 0 \\ 0, g_c[i-1]g[i] = 0, \end{cases}$$

где  $k_1$  – константа увеличения скорости обучения;

$k_2$  – константа уменьшения скорости обучения,  $k_2 \in (0;1)$ ;

$g_c[i]$  – среднее взвешенное изменение градиента на шаге  $i$ .

$$g_c[i] = (1 - c)g[i] + cg[i-1],$$

где  $c$  – фактор выпуклости весов,  $c \in [0;1]$ .

Линейное увеличение изменения скорости позволяет избежать слишком быстрого роста скорости. Геометрическое уменьшение позволяет оставлять скорость обучения всегда положительной. При этом скорость может уменьшаться более быстро на сильно нелинейных участках.

Были предложены и другие эвристики.

*Кестен:* если последовательные изменения веса имеют противоположные знаки, то данный вес осциллирует, и скорость обучения должна быть уменьшена;

*Садирис:* если серия последовательных изменений веса имеет одинаковые знаки, то скорость обучения должна быть увеличена.

*Преимущества:* ускоряется процесс сходимости алгоритма обратного распространения за счет использования дополнительной информации об изменении параметров и весов во время обучения.

*Недостатки:*

1 Даже небольшое линейное увеличение коэффициента обучения может привести к значительному росту скорости обучения, что вызовет скачки в пространстве весов.

2 Геометрическое уменьшение коэффициента иногда оказывается не достаточно быстрым.

#### 4.4.6 Расширенная Дельта-Дельта с чертой

Минай и Вильямс в 1990 г. усовершенствовали алгоритм DBD, предложив расширенную Дельта-Дельта с чертой (EDBD – ExtendedDelta-Bar-Delta).

Алгоритм вводит параметр *момент связи* (momentum – импульс), представляющий собой некоторое число, пропорциональное предыдущему изменению веса.

Изменение веса на последующем шаге:

$$w[i+1] = w[i] + (a[i] + da[i])g[i] + (m[i] + dm[i])dw[i],$$

где  $dw[i]$  – изменение веса на шаге  $i$ ;

$m[i]$  – значения момента на шаге  $i$ ;

$dm[i]$  – изменение значения момента на шаге  $i$ .

$$da[i] = \begin{cases} k_{a1}e^{-y_a|g[i]|}, g_c[i-1]g[i] > 0 \\ -k_{a2}a[i], g_c[i-1]g[i] < 0 \\ 0, g_c[i-1]g[i] = 0, \end{cases}$$

где  $k_{a1}, k_{a2}$  – факторы масштабирования скорости обучения;

$y_a$  – экспоненциальный фактор скорости обучения.

$$dm[i] = \begin{cases} k_{m1}e^{-y_m|g[i]|}, g_c[i-1]g[i] > 0 \\ -k_{m2}m[i], g_c[i-1]g[i] < 0 \\ 0, g_c[i-1]g[i] = 0, \end{cases}$$

где  $k_{m1}, k_{m2}$  – факторы масштабирования момента;

$y_m$  – экспоненциальный фактор момента.

Для скорости обучения и момента задаются верхние границы:  
 $a[i] < a_{\max}, m[i] < m_{\max}$ .

Кроме этого в алгоритм встраивается память со свойством восстановления. После каждой эпохи обучения, оценивается накопленная погрешность. Если текущая ошибка превысила минимальную предыдущую ошибку с учетом максимального отклонения ( $f$ ), то восстанавливаются все веса для наилучшего варианта, а коэффициенты скорости обучения и момента уменьшаются.

*Преимущества:*

1 Обучение происходит быстрее, чем в алгоритме DBD, за счет прохождения пологих мест на поверхности функции ошибки большими шагами.

2 Снижается вероятность осцилляции весов.

*Недостатки:* эффективность метода зависит от конкретной задачи. На некоторых задачах он может давать даже отрицательный эффект.

#### 4.4.7 Эластичное обратное распространение

Эластичное обратное распространение (Resilientpropagation – Rprop) предложено Ридмиллером и Брауном в 1993 г.

Метод учитывает только знак частной производной для того, чтобы указать направление шага. Размер шага определяется при помощи специфического для каждого веса «значения обновления»  $\Delta_{ij}(t + 1)$ :

$$\Delta w_{ij}(t + 1) = \begin{cases} -\Delta_{ij}(t + 1), & \text{если } \frac{\partial E(t+1)}{\partial w_{ij}} > 0, \\ +\Delta_{ij}(t + 1), & \text{если } \frac{\partial E(t+1)}{\partial w_{ij}} < 0 \\ 0 & \text{иначе,} \end{cases}$$

где  $E$  – функция ошибки по всем наблюдениям в обучающей выборке.

Для определения величины коррекции используется следующее правило:

$$\Delta_{ij}(t + 1) = \begin{cases} \eta^+ \Delta_{ij}(t), & \text{если } \frac{\partial E(t)}{\partial w_{ij}} \frac{\partial E(t+1)}{\partial w_{ij}} > 0, \\ \eta^- \Delta_{ij}(t), & \text{если } \frac{\partial E(t)}{\partial w_{ij}} \frac{\partial E(t+1)}{\partial w_{ij}} < 0, \\ \Delta_{ij}(t) & \text{иначе,} \end{cases}$$

где  $0 < \eta^- < 1 < \eta^+$  (рекомендуется  $\eta^- = 0,5$  и  $\eta^+ = 1,2$ ).

Случай, когда частная производная по весу  $w_{ij}$  изменяет свой знак, свидетельствует о том, что последнее обновление весов было слишком большим, и локальный минимум был пройден. Тогда значение обновления  $\Delta_{ij}(t + 1)$  уменьшается в  $\eta^-$  раз. Если производная не меняет знак, то шаг увеличивается в  $\eta^+$  раз, чтобы быстрее подойти к минимуму.

Чтобы не допустить слишком больших или малых значений весов, величину коррекции ограничивают:  $\Delta_{\min} \leq \Delta_{ij} \leq \Delta_{\max}$  (рекомендуется  $\Delta_{\min} = 10^{-6}$  и  $\Delta_{\max} = 50$ ).

Начальные значения для всех  $\Delta_{ij}$  устанавливаются равными 0,1.

*Преимущество:* алгоритм сходится в 4-5 раз быстрее, чем стандартный алгоритм обратного распространения ошибки.

#### 4.5 Алгоритм имитации отжига

*Шаги алгоритма:*

1 Определить переменную  $T$ , представляющую искусственную температуру. Придать  $T$  большое начальное значение.

2 Предъявить сети пример и вычислить целевую функцию (энергию) –  $E(w)$ .

3 Дать случайное изменение весу ( $w$ ) и пересчитать целевую функцию. Величина изменения определяется различными способами, например, подобно тепловой системе в соответствии с гауссовым распределением:

$$P(w) = e^{-\frac{w^2}{T^2}}.$$

4 а) Если целевая функция уменьшилась, то сохранить изменение веса.

б) Если изменение веса приводит к увеличению целевой функции, то вероятность сохранения этого изменения вычисляется с помощью распределения Больцмана:

$$P(\Delta E) = e^{-\frac{\Delta E}{kT}},$$

где  $\Delta E$  – изменение в целевой функции;

$k$  – константа, аналогичная константе Больцмана.

Выбирается случайное число  $r$  из равномерного распределения от 0 до 1. Если  $P(\Delta E) > r$ , то изменение сохраняется, иначе – нет. Это позволяет системе делать случайный шаг в направлении, портящем целевую функцию, позволяя ей тем самым вырываться из локальных минимумов.

5 Повторить шаги 3 и 4 для каждого из весов сети, постепенно уменьшая температуру  $T$ , до достижения целевой функцией допустимо низкого значения.

6 Предъявить следующий пример и повторить обучение.

*Преимущества:*

1 Преодоление проблемы локальных минимумов.

- 2 Возможность использования для обучения полносвязных сетей.
- 3 Метод не требует непрерывной дифференцируемости активационных функций.

*Недостаток:* низкая скорость обучения.

## 4.6 Генетические алгоритмы

Основоположник – Холланд, 1975 г.

Алгоритмы основаны на моделировании развития биологической популяции.

Пусть  $Z(w)$  – функция ошибки нейросети,  $w$  – вектор весов и смещений. *Популяцией* называется набор векторов  $W = \{w_i\} = w_1 \dots w_n$ .  $n$  – размер популяции. Элементы  $w_i$  – *особи* (или *хромосомы*). Каждая из позиций вектора хромосомы называется *геном*.

Элементы множества  $W$  способны эволюционировать по следующим правилам:

1 Если  $Z(w_i)$  – мала, то особь  $w_i$  считается удачной и получает приоритет при размножении. Вероятность гибели этой особи снижается. Если  $Z(w_i)$  – велика, то вероятность гибели увеличивается.

2 *Мутации*. Любая точка имеет равную вероятность мутации, т. е. смещения на небольшую величину:  $w_i \rightarrow w_i + \Delta w_i$ , где  $\Delta w_i$  – небольшой по модулю вектор, характеризующий величину мутации. Обычно изменяется один или несколько ген.

3 *Размножение*: в соответствии с вероятностями, определенными на шаге 1, каждая точка имеет вероятность размножения, тем большую, чем точка удачнее. Примеры законов размножения:

а) *деление* – разделение на две (или несколько) близлежащих точки:  $w_0 + m_1$  и  $w_0 + m_2$ , где векторы  $m_1$  и  $m_2$  определяются алгоритмом;

б) *кроссинговер* – наследования участков хромосомных наборов родителей (размножение со скрещиванием): две близких точки  $w_1$  и  $w_2$  таких, что  $\|w_1 - w_2\|$  мала, скрещиваются, давая одного или нескольких потомков, состоящих из наборов ген родителей.

Предложены следующие виды кроссинговера:

– *одноточечный кроссинговер*. Случайным образом выбирается один из генов, номер которого принимается за точку разрыва. Обе родительские хромосомы в этой точке разрываются на два сегмента. Затем соответствующие сегменты разных родителей склеиваются, образуя два генотипа потомков.

– *двухточечный кроссинговер*. Выбираются две точки разрыва, и родительские хромосомы обмениваются сегментом, находящимся между этими точками.

– *равномерный кроссинговер*. Каждый бит первого родителя наследуется первым потомком с вероятностью, определенной на шаге 1, в противном случае этот бит передается второму потомку. И наоборот.

Считается, что мутации являются причиной появления новых биологических видов, а кроссинговер определяет изменчивость внутри вида.

4 *Гибель*: в соответствии с вероятностью, определенной на шаге 1, точка может погибнуть, т.е. быть бесследно удалена из множества  $W$ . В этот шаг может быть введен принцип *элитности*, который гарантирует, что при отборе обязательно будут выживать лучший или лучшие члены популяции.

Текущий набор  $W$  составляет *генофонд* популяции. На каждом шаге эволюции получают все более совершенные особи (с меньшей  $Z(w)$ ).

В генетических алгоритмах присутствуют наследственность, изменчивость и естественный отбор – движущие силы биологической эволюции. Точная количественная теория эволюции пока не построена. Поэтому оптимальность выбранных параметров алгоритма пока может быть оценена только экспериментально.

*Достоинства:*

- 1 Борьба с локальными минимумами;
- 2 Массовый параллелизм при обработке: особи в популяции функционируют независимо;
- 3 Биоподобность

*Недостатки:*

- 1 Для каждой особи требуется много памяти.
- 2 Склонность к параличу при обучении.
- 3 Низкая эффективность на фон-неймановских ЭВМ из-за отсутствия параллелизма.

*Контрольные вопросы по теме 4:*

- 1 Опишите метод обучения Хэбба.
- 2 Опишите метод обучения Уидроу-Хоффа.
- 3 Опишите алгоритм обратного распространения ошибки.
- 4 Опишите модификации алгоритма обратного распространения ошибки.
- 5 Опишите алгоритм имитации отжига.
- 6 Опишите генетический алгоритм обучения.

## 5 ПРЯМОНАПРАВЛЕННЫЕ НЕЙРОННЫЕ СЕТИ

### 5.1 Радиально-базисная сеть

*Другие названия:* RBF-сеть (RBF – Radial Basis Function), сеть радиального основания. *Авторы:* D. S. Broomhead, D. Lowe – 1988 г.; J. Moody, C.J. Darkin – 1989 г.

*RBF-сеть* – это нейронная сеть, которая содержит слой радиально-симметричных скрытых нейронов – *шаблонный слой* (см. рис. 5.1).

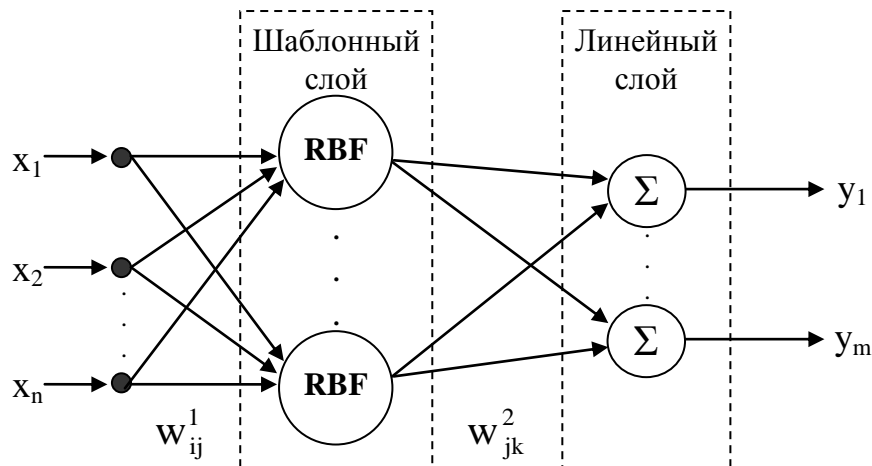


Рисунок 5.1 – Радиально-базисная сеть

Для того чтобы шаблонный слой был радиально-симметричным необходимо выполнение следующих условий:

1 Наличие центра, представленного в виде вектора во входном пространстве. Обычно этот вектор сохраняется в пространстве весов от входного слоя к слою шаблонов;

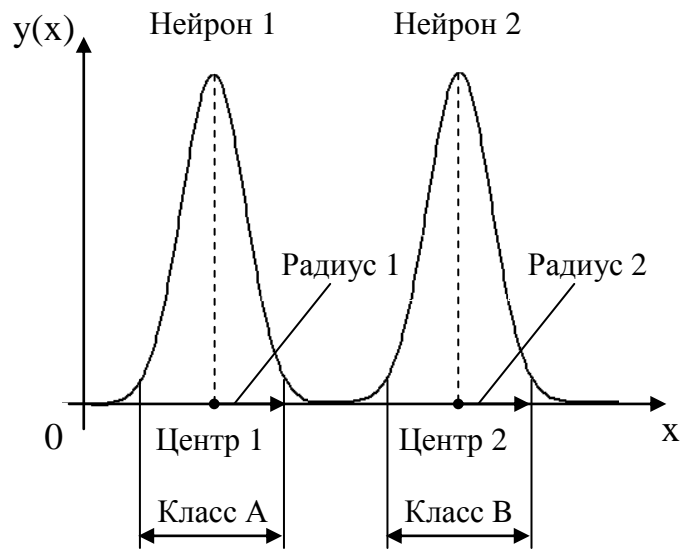
2 Наличие способа измерения расстояния входного вектора от центра. Обычно это стандартное евклидово расстояние;

3 Наличие специальной функции, которая определяет выходной сигнал нейрона как функцию расстояния. Обычно используется функция Гаусса (см. формулы (2.10), (2.11)).

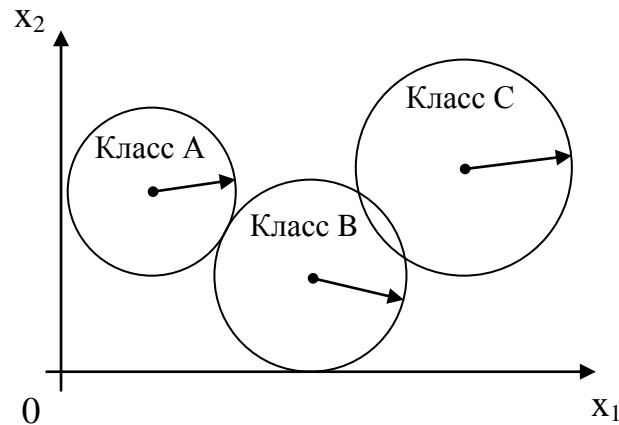
Т. е. выходной сигнал шаблонного нейрона это функция от расстояния между входным вектором и сохраненным центром. Для моделирования любой функции необходимо взять линейную комбинацию (т.е. взвешенную сумму гауссовых функций) достаточного числа радиальных элементов.

В отличие от многослойного персептрона, разбивающего пространство признаков гиперплоскостями, RBF-сеть разбивает исходное пространство гиперсферами, которые задаются своими центрами и радиусами. На рис. 5.2 показаны примеры разбиения пространства признаков шаблонными нейронами.





а) один признак



б) два признака

Рисунок 5.2 – Примеры работы RBF-сети

На рис. 5.2, а) объекты классификации, представляемые признаком  $x$ , разбиваются двумя шаблонными нейронами на классы А и В. Центры 1 и 2 соответствуют наиболее характерным представителям соответствующих классов. Как видно из рисунка – чем ближе классифицируемый объект к центру, тем он более похож на эталонный объект класса и, соответственно, тем сильнее будет отклик нейрона. Радиусы шаблонных нейронов определяют границы класса. В приведенном примере радиусы выбраны такими, что между двумя классами остается «мертвая зона». Это означает, что объекты, лежащие между классами не будут «узнаны» ни одним нейроном.

На рис. 5.2, б) показаны классы, задаваемые двумя признаками. Здесь граница класса В соприкасается с границей класса А и пересекает границу класса С. Т. е. существуют «пограничные» объекты, которые можно отнести как к классу В, так и к классу С.

Таким образом, правильность выбора центра и радиуса определяет успешность классификации: центры должны соответствовать истинным центрам класса, а радиусы – его размеру. Если радиус нейрона будет выбран слишком большим, то вместе с правильным ответом возможны ложные срабатывания соседних нейронов. Если слишком малым, то часть объектов не будет классифицирована. Определение центров и радиусов составляет задачу обучения RBF-сети.

Одним из методов задания центров является *выборка из выборки*. В качестве центров берутся несколько случайно выбранных точек обучающего множества. В силу случайности выбора они представляют статистическое распределение обучающих данных. Недостатком этого метода является то, что если число радиальных элементов небольшое, то такое представление может быть неудовлетворительным.

После определения расположения центров, необходимо найти величины радиусов. Если радиусы выбраны слишком маленькими, то сеть не будет интерполировать данные между известными точками и потеряет способность к обобщению. Если радиусы выбраны слишком большими, то области классификации, формируемые сетью будут иметь большие пересекающиеся зоны, что может привести к ошибкам при классификации. Методы задания радиусов:

- 1 *Явный* – задаются пользователем.
- 2 *Изотропный*. Радиусы задаются одинаковыми для всех элементов и определяются эвристически с учетом количества радиальных элементов и объема покрываемого пространства.
- 3 *К ближайших соседей*. Отклонение каждого элемента устанавливается индивидуально равным среднему расстоянию до  $K$  его ближайших соседей. При этом радиусы будут меньше в тех частях пространства, где точки расположены рядом, а там, где точек мало, радиусы будут большими.

Также для нахождения центров и радиусов можно использовать итеративные методы оптимизации.

После того, как выбраны центры и радиусы, параметры выходного слоя оптимизируются с помощью стандартного метода линейной алгебры – метода псевдообратных матриц, минимизирующий средний квадрат ошибки.

Вектор весовых коэффициентов  $i$ -го выходного нейрона рассчитываются по формуле

$$\bar{W}_i = (H^T H)^{-1} H^T \bar{Y}_i,$$

где  $\bar{Y}_i$ ,  $i = 1 \dots M$  – вектор эталонных выходных значений для  $i$ -го выходного нейрона (по всей обучающей выборке);

$M$  – количество выходных нейронов;

$H$  – интерполяционная матрица.

$$H = \begin{pmatrix} h_1(\bar{X}_1) & h_2(\bar{X}_1) & \dots & h_K(\bar{X}_1) \\ h_1(\bar{X}_2) & h_2(\bar{X}_2) & \dots & h_K(\bar{X}_2) \\ \vdots & \vdots & \ddots & \vdots \\ h_1(\bar{X}_p) & h_2(\bar{X}_p) & \dots & h_K(\bar{X}_p) \end{pmatrix},$$

где  $h_k(\bar{X})$  – выходной сигнал  $k$ -го радиального нейрона,  $k = 1 \dots K$ ;

$K$  – число радиальных нейронов;

$\bar{X}$  – вектор входного сигнала;

$p$  – размер обучающей выборки.

*Преимущества RBF-сетей:*

- 1 Точно известно количество слоев сети;
- 2 Параметры линейной комбинации в выходном слое можно оптимизировать с помощью скоростных методов линейного моделирования. Поэтому сеть RBF обучается очень быстро (на порядок быстрее эквивалентного многослойного персептрона).

*Недостатки RBF-сетей:*

- 1 Необходимость определения числа радиальных элементов, положения их центров и радиусов;
- 2 Для достижения одной и той же точности сеть RBF требует большего числа элементов, чем многослойный персептрон. Поэтому сеть RBF работает медленнее и требует больше памяти;
- 3 При удалении от обучающего множества значение функции отклика быстро падает до нуля. Поэтому RBF-сеть плохо экстраполирует (хотя в конкретной задаче это может оказаться достоинством: лучше не получить ответ, чем получить ложный);
- 4 Сети RBF более чувствительны к «проклятию размерности» и трудно применимы, когда число входов велико (в данном случае «проклятие размерности» – очень быстрый рост размеров сети при увеличении числа входов).

## 5.2 Звезды Гроссберга

### *Входная звезда*

Входная звезда (Instar) разработана С. Гроссбергом в 1974 г.

Звезды рассматриваются как модели отдельных участков мозга.

Входная звезда состоит из нейрона, на который подается группа входов через синаптические веса (см. рис. 5.3). Входная звезда обучается реагировать на определенный входной вектор, и ни на какой другой. Хорошо обученная входная звезда будет реагировать не только на определенный вектор, но и на незначительные изменения этого вектора. Таким образом, звезда будет проявлять способность к обобщению.

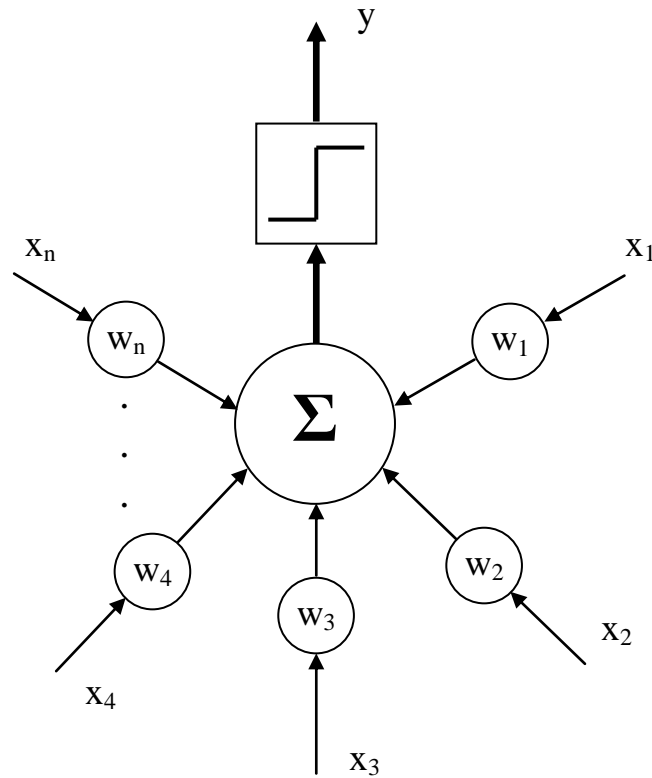


Рисунок 5.3 – Входная звезда

Процесс обучения выражается следующей формулой:

$$w_i(t+1) = w_i(t) + \alpha[x_i - w_i(t)],$$

где  $w_i$  – вес  $i$ -го входа ( $i = 1 \dots n$ );

$t$  – номер итерации;

$x_i$  –  $i$ -й вход;

$\alpha$  – нормирующий коэффициент обучения, который имеет, начальное значение 0,1 и постепенно уменьшается в процессе обучения.

#### Выходная звезда

Выходная звезда (Outstar) разработана совместно с входной звездой и состоит из нейрона, управляющего группой (см. рис. 5.4).

Выходная звезда вырабатывает требуемый возбуждающий сигнал для других нейронов всякий раз, когда возбуждается.

В процессе обучения нейрона выходной звезды, его веса настраиваются в соответствии с требуемым целевым вектором. Веса выходной звезды постепенно настраиваются на множество векторов, представляющих собой вариации идеального (исходного) вектора. В этом случае выходной сигнал нейронов представляет собой статистическую характеристику обучающего набора.

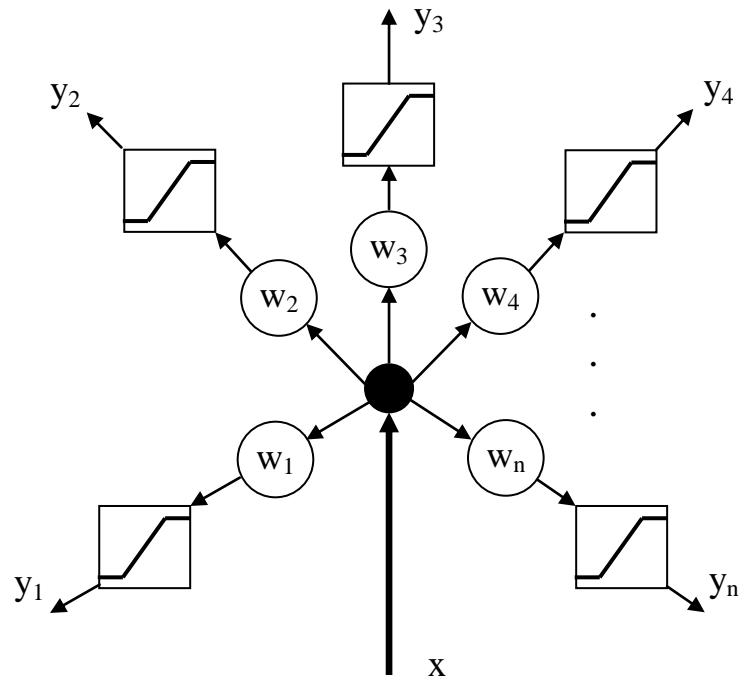


Рисунок 5.4 – Выходная звезда

Процесс обучения выражается следующей формулой:

$$w_i(t+1) = w_i(t) + b[y_i - w_i(t)],$$

где  $b$  – нормирующий коэффициент обучения, который в начале приближенно равен единице и постепенно уменьшается до нуля в процессе обучения.

*Области применения звезд:* применяются в составе нейросетевых систем.

*Преимущества звезд:* входная звезда хорошо моделирует некоторые функции компонент биологических нейронных сетей. Сеть, включающая в себя входные звезды, может быть достаточно хорошей моделью отдельных участков мозга. При решении практических задач входные звезды могут быть использованы для построения простых быстро обучаемых сетей.

*Недостатки звезд:* 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.

### 5.3 Нейросетевой гауссов классификатор

Модель предложена Р.П. Липпманом в 1987 г.

В основе построения гауссова классификатора лежат предположения о распределениях входных сигналов. Считается, что эти распределения из-

вестны и соответствуют закону Гаусса. Тогда однослойный персептрон может быть использован для реализации гауссова классификатора по максимуму вероятности.

Пусть  $M_{Ai}$  и  $S_{Ai}^2$  – математическое ожидание и дисперсия входного сигнала  $x_i$  ( $i = 1 \dots N$ ), когда входной сигнал принадлежит классу А,  $M_{Bi}$  и  $S_{Bi}^2$  – классу В, а  $S_{Ai}^2 = S_{Bi}^2 = S_i^2$ . Тогда значения вероятности, используемые классификатором по максимуму вероятности пропорциональны следующим величинам:

$$L_A = -\sum_{i=1}^N \frac{(x_i - M_{Ai})^2}{S_i^2} = -\sum_{i=1}^N \frac{x_i^2}{S_i^2} + 2\sum_{i=1}^N \frac{M_{Ai}x_i}{S_i^2} - \sum_{i=1}^N \frac{M_{Ai}^2}{S_i^2};$$

$$L_B = -\sum_{i=1}^N \frac{(x_i - M_{Bi})^2}{S_i^2} = -\sum_{i=1}^N \frac{x_i^2}{S_i^2} + 2\sum_{i=1}^N \frac{M_{Bi}x_i}{S_i^2} - \sum_{i=1}^N \frac{M_{Bi}^2}{S_i^2}.$$

Классификатор по максимуму вероятности вычисляет  $L_A$  и  $L_B$  и выбирает класс с наибольшей вероятностью. Первые слагаемые в формулах идентичны и их можно опустить. Вторые слагаемые могут быть вычислены путем умножения входных сигналов на синаптические веса. Третьи слагаемые – константы, значения которых можно присвоить смещению нейрона. Тогда параметры нейросетевого гауссова классификатора можно вычислить по следующим формулам:

1 Значения синаптических весов

$$w_i = \frac{2(M_{Ai} - M_{Bi})}{S_i^2}.$$

2 Значение смещения

$$b = \sum_{i=1}^N \frac{M_{Ai}^2 - M_{Bi}^2}{S_i^2}.$$

*Области применения сети:* распознавание образов, классификация.

*Преимущества:* простая и быстрая нейросеть. Простой способ формирования весов и смещений.

*Недостатки:* необходимо, чтобы распределения входных сигналов, соответствующие разным классам, были априорно известны.

## 5.4 Сеть поиска максимума с прямыми связями

Сеть поиска максимума с прямыми связями (Feed-Forward MAXNET) предложена Р.П. Липпманом в 1987 г.

В основе сети лежит двухвходовый нейросетевой компаратор (элемент сравнения), который выполняет сравнение двух аналоговых сигналов –  $x_1$  и  $x_2$  (см. рис. 3.19). На выходе  $z$  – среднее значение входных сигналов. Выходы  $y_1$  и  $y_2$  показывают, какой именно входной сигнал имеет максимальное значение: «1» соответствует большему входу, «-1» – меньшему. Весовые коэффициенты фиксированы и определены заранее (см. рис. 5.5). Порог нейронов, формирующих сигналы  $y_1$  и  $y_2$  равен нулю.

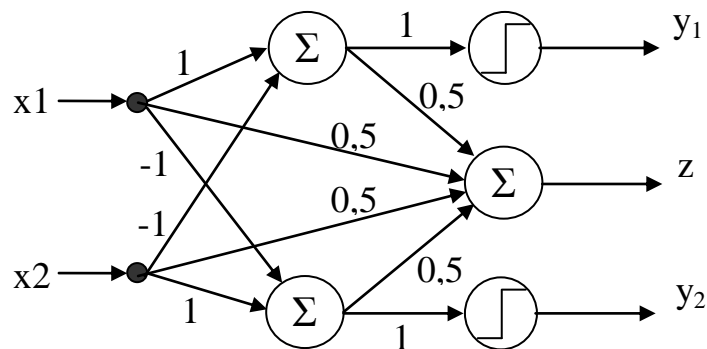


Рисунок 5.5 – Двухвходовый нейросетевой компаратор

Для сравнения  $n$  сигналов двухвходовые нейросетевые компараторы объединяются в более сложные многослойные сети. Число слоев в таких сетях равно  $\log_2 n$ , округленное до ближайшего большего целого. Полученная сеть выполняет попарное сравнение входных сигналов друг с другом. Число выходов такой сети равно  $n + 1$  ( $y_1 \dots y_n, z$ ).

*Области применения сети:* применяется в составе нейросетевых систем распознавания образов.

*Преимущество:* заранее известны значения весовых коэффициентов нейронов.

*Недостаток:* число слоев сети растет с увеличением размерности входного сигнала.

*Контрольные вопросы по теме 5:*

- 1 Опишите устройство и работу радиально-базисной сети.
- 2 Опишите устройство и работу входная звезды.
- 3 Опишите устройство и работу выходная звезды.
- 4 Опишите устройство и работу нейросетевого гауссова классификатора.
- 5 Опишите устройство и работу сети поиска максимума с прямыми связями.

## 6 СЕТИ ЕСТЕСТВЕННОЙ КЛАССИФИКАЦИИ

### 6.1 Сеть Кохонена

Сеть Кохонена (*карта Кохонена*) предложена Т. Кохоненом в 1984 г.

В мозге нейроны располагаются в определенном порядке так, что некоторые внешние физические воздействия вызывают ответную реакцию нейронов из определенной области мозга. Например, в той части мозга, которая отвечает за восприятие звуковых сигналов, нейроны группируются в соответствии с частотами входного сигнала, на которых они резонируют. Хотя строение мозга в значительной степени предопределяется генетически, отдельные структуры мозга формируются в процессе самоорганизации. Алгоритм Кохонена напоминает процессы, происходящие в мозге.

Алгоритм Кохонена дает возможность строить нейронную сеть для разделения векторов входных сигналов на подгруппы.

Сеть состоит из  $M$  нейронов, образующих прямоугольную решетку на плоскости (см. рис. 6.1). Элементы входных сигналов подаются на входы всех нейронов сети. Нейроны запоминают в своих весовых коэффициентах одно ядро класса и отвечают за определение объектов в своих классах.

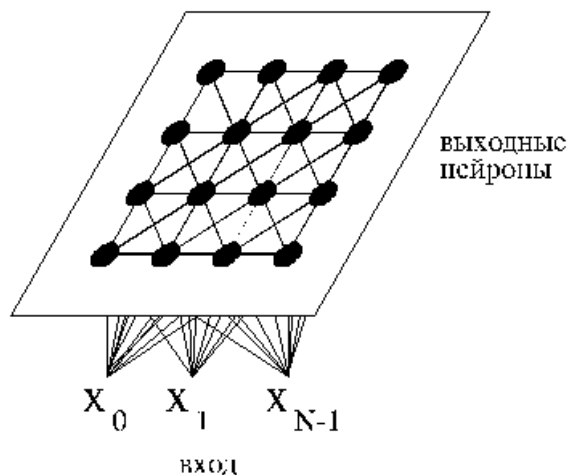


Рисунок 6.1 – Карта Кохонена

Каждый нейрон возбуждается тем больше, чем ближе значение входного вектора  $\bar{X}$  к вектору его весовых коэффициентов  $\bar{W}$ . Сравнение векторов осуществляется с помощью вычисления евклидова расстояния.

Показано, что для сравнения расстояний достаточно выполнить линейное преобразование входного сигнала по формуле:



$$d(\bar{X}, \bar{W}) = \sum_{i=1}^n x_i w_i - \frac{\sum_{i=1}^n w_i^2}{2}.$$

Эту формулу будет реализовывать линейный нейрон с порогом равным второму слагаемому.

Сеть Кохонена может функционировать в двух режимах: *режиме аккредитации* и *режиме интерполяции*.

В режиме аккредитации выходные нейроны функционируют по правилу «победитель получает всё»: нейрон, возбуждение которого максимально выдаёт единицу, остальные – нули.

В режиме интерполяции проигравшие нейроны не затормаживаются и тогда по уровням их возбуждения можно оценить степень схожести входного вектора с векторами, которые хранятся в нейронах.

Главной особенностью карты Кохонена является то, что пространственно близкие входные примеры возбуждают топологически близкие нейроны, т.е. соседние узлы карты оказываются соседними в пространстве данных.

При этом невозможно заранее предсказать, какой именно нейрон будет активироваться для заданного входного вектора. Необходимо лишь гарантировать, чтобы в результате обучения разделялись несхожие входные векторы. Идентификация классов проводится путем тестирования сети после обучения.

*Алгоритм классификации:*

- 1 На вход нейронной сети подается вектор  $\bar{X}$ .
- 2 Номер нейрона, выдавшего минимальный ответ, является номером класса, к которому принадлежит вектор  $\bar{X}$ .

Для реализации алгоритма необходимо определить меру соседства нейронов (меру близости). На рис. 6.2 зоны топологического соседства на карте признаков в различные моменты времени

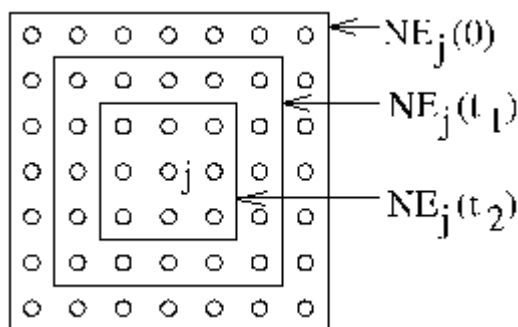


Рисунок 6.2 – Зоны топологического соседства на карте признаков в различные моменты времени

$NE_j(t)$  - множество нейронов, которые считаются соседями нейрона  $j$  в момент времени  $t$ . Зоны соседства уменьшаются с течением времени.

### Обучение карты Кохонена

Сеть использует принцип «обучение без учителя».

Этапы алгоритма:

- 1 Инициализация сети.
- 2 Предъявление сети нового входного сигнала.
- 3 Вычисление расстояния до всех нейронов сети.
- 4 Выбор нейрона-победителя.
- 5 Настройка весовых коэффициентов победившего нейрона и его соседей.

Веса корректируются так, чтобы они стали более похожи на текущий входной пример по правилу

$$w_{ij}(t+1) = w_{ij}(t) + r(t)h(t)(x_i(t) - w_{ij}(t)),$$

где  $t$  – номер итерации обучения;

$w_{ij}(t+1)$  – значение  $i$ -го весового коэффициента  $j$ -го нейрона после шага обучения;

$w_{ij}(t)$  – текущее значение  $i$ -го весового коэффициента  $j$ -го нейрона (до обучения);

$x_i(t)$  – текущее значение  $i$ -го компонента входного вектора;

$r(t)$  – коэффициент скорости обучения, начальное значение которого устанавливается близким к единице и уменьшается до нуля в процессе обучения;

$h(t)$  – функция соседства, определяющая меру соседства нейрона-победителя и окружающих его узлов. Обычно используется гауссова функция

$$h(t) = \exp\left(-\frac{\|\overline{\mathbf{W}}^* - \overline{\mathbf{W}}_j\|^2}{\sigma^2(t)}\right),$$

где  $\overline{\mathbf{W}}^*$  – координаты нейрона-победителя на карте Кохонена;

$\overline{\mathbf{W}}_j$  – координаты  $j$ -го нейрона на карте Кохонена ( $j = 1 \dots m$ );

$\sigma(t)$  – радиус захвата соседей, начальное значение которого устанавливается таким, чтобы в движении участвовало более половины узлов и уменьшается до нуля в процессе обучения.

Значение функции соседства равно единице для выигравшего нейрона и уменьшается с удалением от него.

- 6 Вычисление ошибки работы карты Кохонена.

Ошибка оценивается по среднеквадратическому расстоянию от каждой точки данных до ближайшего нейрона карты.

- 7 Проверка условия выхода из процедуры обучения.

Критерии остановки:

- весовые коэффициенты нейронов перестали меняться;
- величина ошибки нейросети стабилизировалась и перестала снижаться;
- достигнут минимум ошибки работы сети;
- все примеры обучающей выборки классифицированы с заданной точностью.

Альтернативой методу пообъектного обучения Кохонена является *метод динамических ядер*.

Рассмотрим множество из объектов  $\bar{X}$ , каждый из которых является  $n$ -мерным вектором. Зададим пространство ядер классов, и меру близости  $\text{dist}(\bar{a}, \bar{X})$ , где  $\bar{a}$  – точка из пространства ядер, а  $\bar{X}$  – точка из пространства объектов. Тогда для заданного числа классов  $K$  необходимо подобрать  $K$  ядер таким образом, чтобы суммарная мера близости была минимальной. Суммарная мера близости записывается в следующем виде:

$$H = \sum_{k=1}^K \sum_{i=1}^I \text{dist}(\bar{a}_k, \bar{X}_{ki}),$$

где  $I$  – количество объектов  $k$ -го класса.

Метод напрямую минимизирует суммарную меру близости  $H$ .

Предварительно задаются начальные значения ядер. Если ядра являются точками того же пространства, что и объекты, то в качестве начального приближения значений ядер выбирают значения объектов. Когда пространство ядер и пространство объектов не совпадают, то задается начальное разбиение объектов на классы. При этом в начальном разбиении могут участвовать не все объекты.

Метод является итерационной процедурой, каждая итерация которой состоит из двух шагов:

- 1 Разбиение на классы при фиксированных значениях ядер:

$$A_k = \left\{ \bar{X} : \text{dist}(\bar{a}_i, \bar{X}) \leq \text{dist}(\bar{a}_j, \bar{X}) \right\};$$

- 2 Оптимизация значений ядер при фиксированном разбиении на классы:

$$Z_k(\bar{a}_k) = \sum_{i=1}^I \text{dist}(\bar{a}_k, \bar{X}_{ki}) \rightarrow \min.$$

В случае равенства в формуле первого шага объект относят к классу с меньшим номером. Процедура останавливается, если после очередного выполнения разбиения на классы не изменился состав ни одного класса.

Процедура сходится за конечное число шагов, причем ни на одном шаге не происходит возрастания суммарной меры близости.

*Области применения:* кластерный анализ, распознавание образов, классификация.

*Преимущества:* сеть Кохонена способна функционировать в условиях помех, так как число классов фиксировано.

*Недостатки:* сеть может быть использована для кластерного анализа только в том случае, если заранее известно число кластеров.

## 6.2 Сеть встречного распространения

Разработана Робертом Хехт-Нильсеном в 1987 г.

### Принцип действия

В сети встречного распространения объединены карта Кохонена и выходные звезды Гроссберга. Считается, что в мозге вычисления осуществляются каскадными соединениями модулей различной специализации.

Сеть встречного распространения имеет два слоя с последовательными связями. Первый слой – слой Кохонена, второй – слой Гроссберга. Каждый элемент входного сигнала подается на все нейроны слоя Кохонена. Каждый нейрон слоя Кохонена соединен со всеми нейронами слоя Гроссберга (см. рис. 6.3).

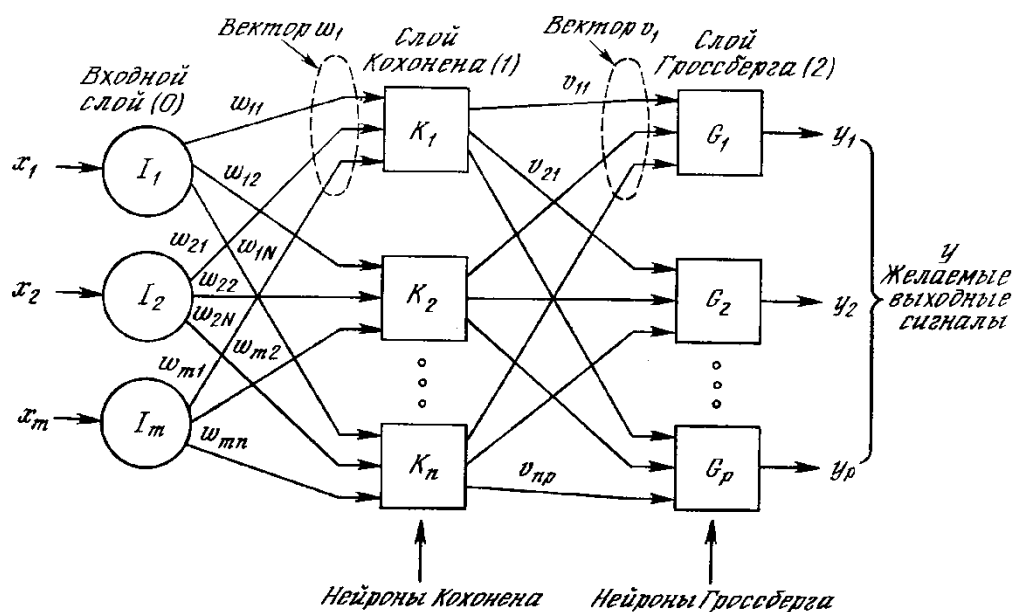


Рисунок 6.3 – Принцип действия сети встречного распространения

В процессе обучения сети встречного распространения входные векторы ассоциируются с соответствующими выходными векторами. Эти векторы могут быть двоичными или непрерывными. После обучения сеть формирует выходные сигналы, соответствующие входным сигналам.

Обобщающая способность сети дает возможность получать правильный выход, когда входной вектор неполон или искажен.

Слой Кохонена функционирует по правилу «победитель получает все». Выход каждого нейрона Кохонена является суммой взвешенных элементов входных сигналов:

$$K_j = \sum_{i=1}^m x_i w_{ij} .$$

Нейрон Кохонена с максимальным значением  $K$  является «победителем».

Выход слоя Гроссберга ( $y$ ) является взвешенной суммой выходов слоя Кохонена:

$$y_p = \sum_{j=1}^n K_j v_{jp} .$$

Каждый нейрон слоя Гроссберга выдает величину веса, который связывает этот нейрон с единственным нейроном Кохонена, чей выход отличен от нуля. Т. е. слой Гроссберга преобразует выход слоя Кохонена в линейный код с кодированием по номеру канала.

Нормализация входных векторов выполняется путем деления каждой компоненты входного вектора на длину вектора. Это превращает входной вектор в единичный вектор с тем же направлением, т.е. в вектор единичной длины в  $n$ -мерном пространстве.

#### *Полная сеть встречного распространения*

В процессе обучения векторы  $\bar{X}$  и  $\bar{Y}$  подаются одновременно и как входные векторы сети, и как желаемые выходные сигналы. Вектор  $\bar{X}$  используется для обучения выходов  $\bar{X}'$ , а вектор  $\bar{Y}$  – для обучения выходов  $\bar{Y}'$  слоя Гроссберга. Полная сеть встречного распространения показана на рис 6.4.

В качестве результирующего получается отображение, при котором предъявление пары входных векторов порождает их копии на выходе. Кроме того, предъявление одного из векторов (при обнулении второго), также порождает на выходе оба вектора. Следовательно, сеть изучает одновременно 2 отображения: прямое  $X \rightarrow Y$  и обратное  $Y \rightarrow X$ .

*Области применения:* распознавание образов, восстановление образов (ассоциативная память), сжатие данных.

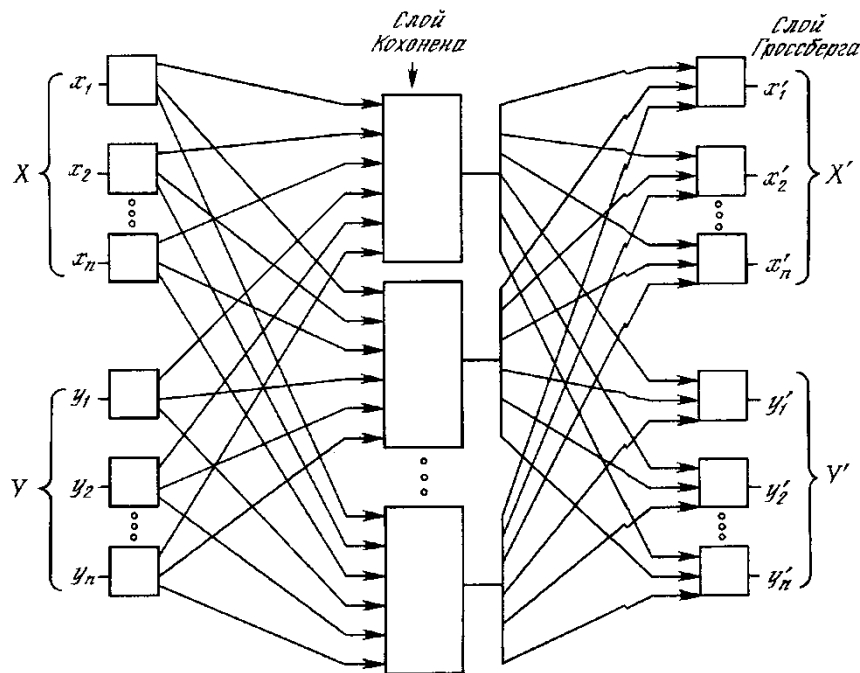


Рисунок 6.4 – Полная сеть встречного распространения

#### Преимущества:

- быстрое обучение (длительность обучения может быть в 100 раз меньше по сравнению с обратным распространением ошибки). Поэтому сеть полезна для приложений, в которых требуется быстрая начальная аппроксимация;

- уникальная способность порождать функцию и обратную к ней.

**Недостатки:** более низкая точность, чем в сети с обратным распространением.

### 6.3 Когнитрон

Был разработан Фукушимой в 1975 г. как модель системы восприятия человека. Когнитрон конструируется в виде слоев нейронов, соединенных синапсами. Имеются два типа нейронов: *возбуждающие узлы*, которые стремятся вызвать возбуждение постсинаптического узла, и *тормозящие узлы*, которые тормозят это возбуждение. Каждый нейрон связан только с нейронами в соседней области, называемой *областью связи*. Аналогично в зрительной коре редко соединяются между собой нейроны, располагающиеся друг от друга на расстоянии более 1 мм. Области связи соседних узлов значительно перекрываются, что вводит взаимную конкуренцию между ближайшими узлами. При обучении в заданной области слоя обучается только наиболее сильно возбужденный нейрон. Это достигается с помощью латерального торможения.

Суммарный возбуждающий вход в нейрон является взвешенной суммой входов от возбуждающих нейронов в предшествующем слое

$$E = \sum_i a_i u_i ,$$

где  $a_i$  – вес  $i$ -го возбуждающего синапса,  $u_i$  – выход  $i$ -го возбуждающего нейрона.

Аналогично для тормозящих нейронов:

$$I = \sum_j b_j v_j ,$$

где  $b_j$  – вес  $j$ -го тормозящего синапса,  $v_j$  – выход  $j$ -го тормозящего нейрона.

Веса имеют только положительные значения.

На рис. 6.5 показан вид когнитрон.

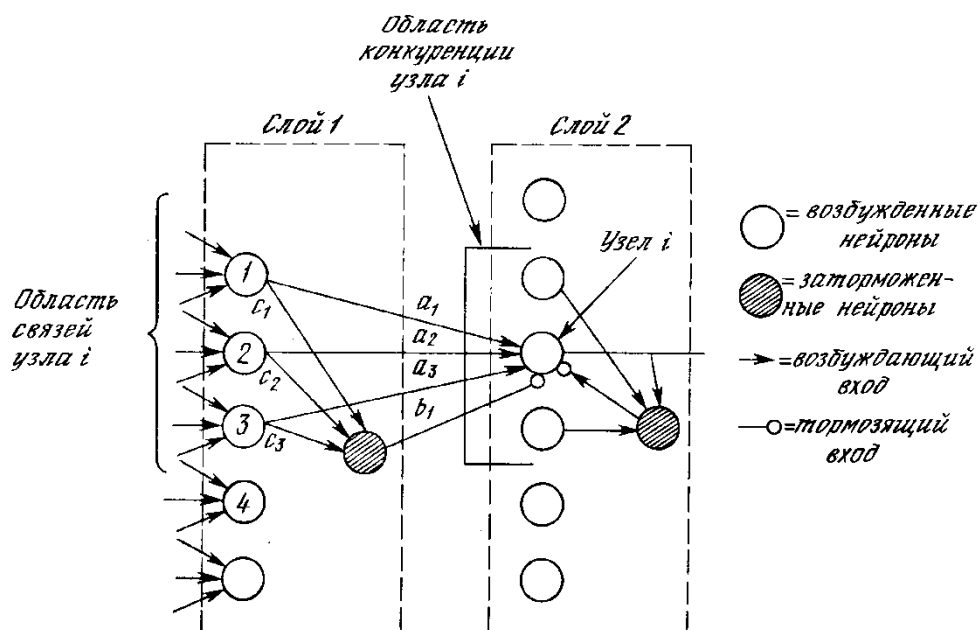


Рисунок 6.5 - Когнитрон

Выход нейрона вычисляется следующим образом:

$$\begin{aligned} \text{NET} &= \frac{1+E}{1+I} - 1; \\ \text{OUT} &= \text{NET}, \text{ при } \text{NET} \geq 0; \\ \text{OUT} &= 0, \text{ при } \text{NET} < 0. \end{aligned}$$

Синаптические веса тормозящих узлов не изменяются в процессе обучения. Их веса заранее установлены таким образом, чтобы сумма весов в любом из тормозящих нейронов была равна единице. Т.е. выход тормозящего узла является взвешенной суммой его входов, которые представляют собой среднее арифметическое выходов возбуждающих нейронов, к которым он подсоединен.

*Преимущества:* биологическое подобие.

*Недостатки:* неспособность распознавать смещенные или повернутые относительно исходного положения образы (*инвариантное* распознавание).

## 6.4 Неокогнитрон

Предложен Фукушимой в 1980 г. в качестве усовершенствования когнитрона. На рис. 6.6 показан вид неокогнитрона.

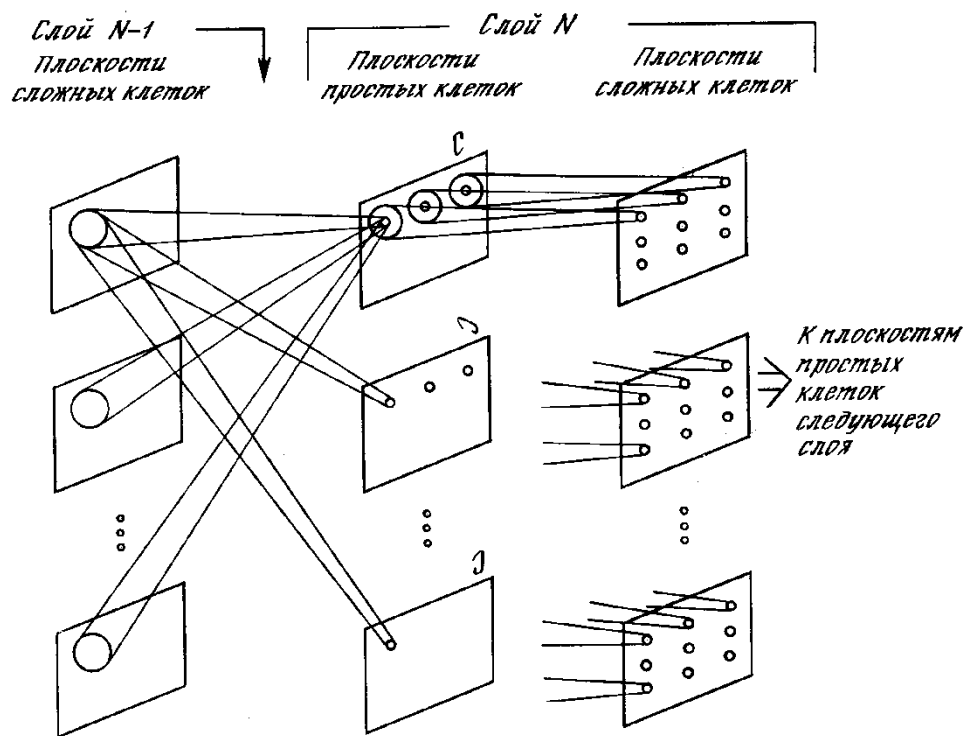


Рисунок 6.6 - Неокогнитрон

Неокогнитрон состоит из иерархии нейронных слоев, каждый из которых состоит из массива плоскостей. Каждый элемент массива состоит из пары плоскостей нейронов. Первая плоскость состоит из *простых* нейронов, которые получают сигналы от предыдущего слоя и выделяют определенные образы. Эти образы далее обрабатываются *сложными* нейронами второй плоскости.



Простые нейроны чувствительны к небольшой области входного образа, называемой *рецептивной* областью. Простой нейрон приходит в возбужденное состояние, если в его рецептивной области возникает определенный образ. Рецептивные области простых клеток перекрываются и покрывают все изображение. Сложные нейроны получают сигналы от простых клеток. При этом для возбуждения сложного нейрона достаточно одного сигнала от любого простого нейрона. Таким образом, сложная клетка регистрирует определенный образ независимо от того, какой из простых нейронов выполнил детектирование, и тем самым снижает чувствительность к расположению образа.

Рецептивные поля нейронов при переходе со слоя на слой расширяются, а число нейронов уменьшается. В выходном слое на каждой плоскости остается только один нейрон, рецептивное поле которого покрывает все поле образа предыдущего слоя.

По мере распространения информации от слоя слою, картина нейронной активности становится все менее чувствительной к ориентации, расположению и размеру образа. Нейроны выходного слоя выполняют окончательное инвариантное распознавание.

*Преимущества:* инвариантное распознавание образов.

*Недостатки:* очень сложная структура сети и большой объем вычислений.

*Контрольные вопросы по теме 6:*

- 1 Опишите устройство и работу сети Кохонена.
- 2 Приведите алгоритм обучения карты Кохонена.
- 3 Опишите устройство и принцип действия сети встречного распространения.
- 4 Опишите принцип действия полной сети встречного распространения.
- 5 Опишите устройство и работу когнитрона.
- 6 Опишите устройство и работу неокогнитрона.

## 7 СЕТИ ЦИКЛИЧЕСКОГО ФУНКЦИОНИРОВАНИЯ (РЕКУРРЕНТНЫЕ СЕТИ)

### 7.1 Сеть Хопфилда

Разработана Хопфилдом в 1984 г.

Сеть функционирует циклически. Выход каждого из нейронов подается на входы всех остальных нейронов. Нейроны имеют жесткие пороговые функции. Выход нейронов вычисляется следующим образом:

$$\begin{aligned} \text{NET}_j &= \sum_{i \neq j} w_{ij} \text{OUT}_i + \text{IN}_j; \\ \text{OUT}_j &= 1, \text{ если } \text{NET}_j > T_j; \\ \text{OUT}_j &= 0, \text{ если } \text{NET}_j < T_j; \\ \text{OUT}_j &\text{ не изменяется, если } \text{NET}_j = T_j, \end{aligned}$$

где  $T_j$  – порог  $j$ -го нейрона.

Размерности входных и выходных сигналов совпадают.

На рис. 7.1 показан вид сети Хопфилда.

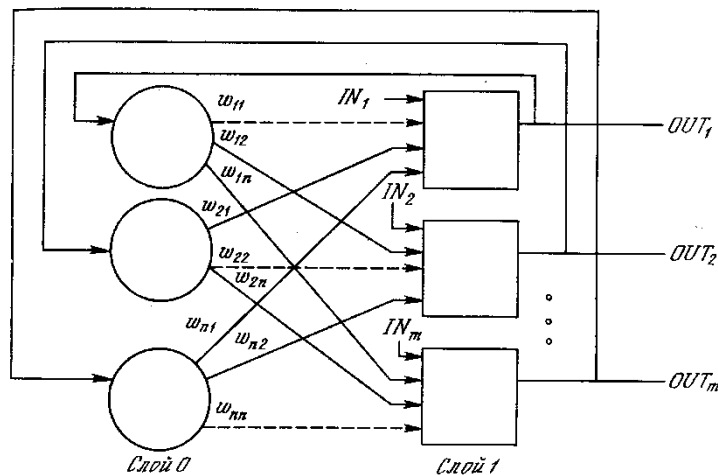


Рисунок 7.1 – Сеть Хопфилда

Текущее состояние сети является двоичным числом. Для 2-х нейронов в выходном слое оно представляется квадратом (см. рис. 7.2).

В общем случае система с  $n$  нейронами имеет  $2^n$  различных состояний и представляется  $n$ -мерным гиперкубом.

Когда подается новый входной вектор, сеть переходит из вершины в вершину, пока не стабилизируется. Если входной вектор частично неправилен или неполон, то сеть стабилизируется в вершине, ближайшей к желаемой. Стабилизация в вершине соответствует локальному энергетическому минимуму сети, называемому *аттрактором*.

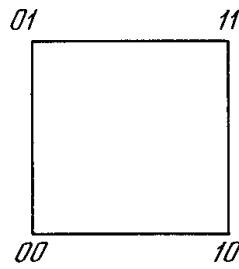


Рисунок 7.2 – Состояния сети Хопфилда

Устойчивая вершина определяется сетевыми весами, текущими входами и величиной порога. Кохен и Гроссберг доказали, что сеть с обратными связями является устойчивой, если ее матрица симметрична и имеет нули в главной диагонали, т. е. если  $w_{ij} = w_{ji}$  и  $w_{ii} = 0$  для всех  $i$ . Симметрия сети является достаточным, но не необходимым условием для устойчивости системы.

Сеть Хопфилда может работать в качестве *ассоциативной памяти*. Это означает, что по вектору, поданному на вход, сеть сформирует на выходе один из запомненных ранее векторов, наиболее «похожий» на данный входной вектор. Такой способ извлечения данных называется *адресацией по содержимому*, в отличие от адресации по номеру ячейки памяти, принятому в ЭВМ фон-неймановского типа. В соответствии с этим способом адресации организована память человека, т. е. некоторое воспоминание может порождать большую связанную с ним область.

Для реализации ассоциативной памяти весовые коэффициенты определяются по правилу Хебба:

$$w_{ij} = \begin{cases} \sum_k x_i^k x_j^k, & i \neq j \\ 0, & i = j, \end{cases}$$

где  $w_{ij}$  –  $i$ -й синаптический вес  $j$ -го нейрона;

$x_i$  –  $i$ -й элемент входного сигнала сети;

$k$  – номер запоминаемого вектора.

Предложена модификация сети Хопфилда, использующая нейроны с сигмоидальной функцией активации.

*Области применения:* ассоциативная память, распознавание образов, задачи оптимизации.

*Преимущества:* простота.

*Недостатки:*

1 Сеть обладает небольшой емкостью. Гроссбергом в 1987 г. было доказано, что в общем случае предельное значение емкости  $0,15N$ , где  $N$  – число нейронов;

2 Наряду с запомненными образами в сети хранятся и их негативы;

3 Размерность и тип входных сигналов в точности совпадают с размерностью и типом выходных сигналов, что ограничивает область применения;

4 При использовании коррелированных векторов-образцов возможно заикливание сети в процессе функционирования;

5 При увеличении размерности входного сигнала квадратично растёт число синапсов.

## 7.2 Двухнаправленная ассоциативная память

Разработана в конце 80-х годов. Наибольший вклад внес Коско.

Структура двухнаправленной ассоциативной памяти (ДАП) показан на рис. 7.3.

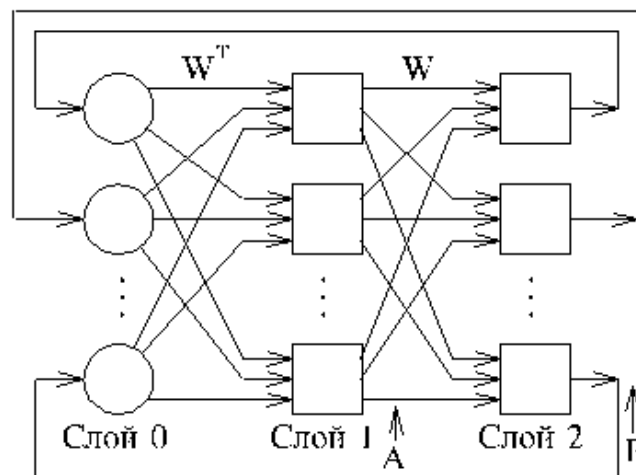


Рисунок 7.3 – Двухнаправленная ассоциативная память

ДАП относится к *гетероассоциативной* памяти: входной вектор поступает на один набор нейронов, а соответствующий выходной вектор вырабатывается на другом наборе нейронов. Входные образы ассоциируются с выходными. Сеть Хопфилда является *автоассоциативной*: входной образ может быть восстановлен или исправлен сетью, но не может быть ассоциирован с другим образом. В сети Хопфилда используется одноуровневая структура ассоциативной памяти, в которой выходной вектор появляется на выходе тех же нейронов, на которые поступает входной вектор.

ДАП способна к обобщению, вырабатывая правильные выходные сигналы, несмотря на искаженные входы.

ДАП функционирует следующим образом:

$$\begin{aligned} B &= F(AW); \\ A &= F(BW^T), \end{aligned}$$

где  $B$  – вектор выходных сигналов;

$A$  – вектор входных сигналов;

$W$  – матрица весов связей между слоями 1 и 2;

$F$  – сигмоидальная функция активации.

Слой 0 не производит вычислений и не имеет памяти. Он является средством распределения выходных сигналов слоя 2 к элементам матрицы  $W^T$ .

Синаптические веса вычисляются по формуле

$$W = \sum_i A_i^T B_i ,$$

где  $i$  – номер примера в обучающей выборке.

Доказано, что ДАП безусловно стабильна при любых значениях весов сети.

*Области применения:* ассоциативная память, распознавание образов.

*Преимущества:* дает возможность строить ассоциации между векторами  $A$  и  $B$ , которые в общем случае имеют разные размерности. Сеть быстро сходится в процессе функционирования.

*Недостатки:* ёмкость ДАП жестко ограничена. Число векторов, которые могут быть запомнены в сети не превышает

$$L = \frac{n}{2 \log_2 n} ,$$

где  $n$  – количество нейронов в меньшем слое.

Если этот лимит превышен, сеть может выработать неверный выходной сигнал, воспроизводя ассоциации, которым не обучена.

*Модификации:*

1 Негомогенная ДАП. Пороговое значение выбирается для каждого нейрона. Такая ДАП может иметь до  $2^n$  стабильных состояний;

2 Непрерывная ДАП. Нейроны могут изменять свое состояние в любое время, когда их вход предписывает это сделать. Т.е. такая ДАП работает в асинхронном режиме;

3 Адаптивная ДАП. Реализовано обучение без учителя;

4 Конкурирующая ДАП. Введены латеральные связи внутри слоев, что приводит к увеличению выхода наиболее высокоактивных нейронов за счет соседних. Такие системы увеличивают контрастность восприятия образа.

### 7.3 Сеть поиска максимума

При работе сети поиска максимума (MAXNET) на каждой итерации большие сигналы на выходах нейронов подавляют слабые сигналы на вы-

ходах других нейронов. Если в начале функционирования сети сигнал на выходе одного из нейронов имел максимальное значение, то в конце функционирования все выходы нейронов, кроме максимального, имеют значения близкие к нулю. Таким образом, нейрон с наибольшим выходным сигналом побеждает. Это достигается за счет использования латеральных связей.

Вид сети поиска максимума показан на рис. 7.4.

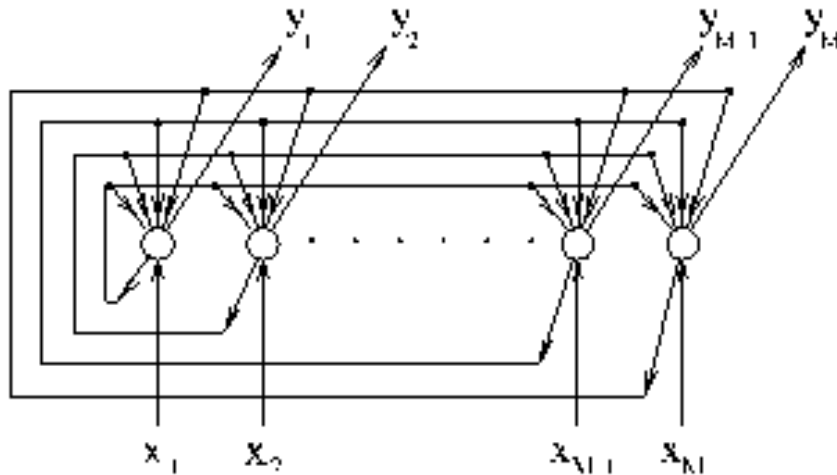


Рисунок 7.4 – Вид сети поиска максимума

Формирование синаптических весов сети

$$w_{ij} = \begin{cases} 1, & i = j \\ -\varepsilon, & i \neq j, \varepsilon \leq \frac{1}{M} \\ 1 \leq i, j \leq M, \end{cases}$$

Функционирование сети

$$\begin{aligned} y_j(0) &= x_j, \quad 1 \leq j \leq M, \\ y_i(t+1) &= f(y_i(t) - \varepsilon \sum_{j=1}^M y_j(t)), \quad 1 \leq i, j \leq M, \end{aligned}$$

где  $w_{ij}$  –  $i$ -й синаптический вес  $j$ -го нейрона,

$x_j$  –  $j$ -й элемент входного сигнала сети,

$y_j$  – выход  $j$ -го нейрона,

$M$  – количество элементов (размерность) входного сигнала, количество нейронов в сети.

$f$  – активационная функция (линейная с насыщением).

Число синапсов в сети –  $M(M - 1)$ . Размерности входных и выходных сигналов совпадают.

Итерации сети завершаются после того, как выходные сигналы нейронов перестают меняться.

*Области применения:* используется в составе нейросетевых систем распознавания образов.

*Преимущества:* простота.

*Недостатки:*

- 1 Число итераций функционирования сети заранее не определено;
- 2 Сеть определяет, какой из входных сигналов имеет максимальное значение, но в процессе функционирования теряет само значение максимального сигнала;
- 3 При увеличении размерности входного сигнала квадратично растёт число синапсов.

## 7.4 Сеть Хемминга

Разработана Липпманом в 1987 г. Модификации: Мюллер, Рейнхардт – 1990 г.

Вид сети Хемминга показан на рис. 7.5.

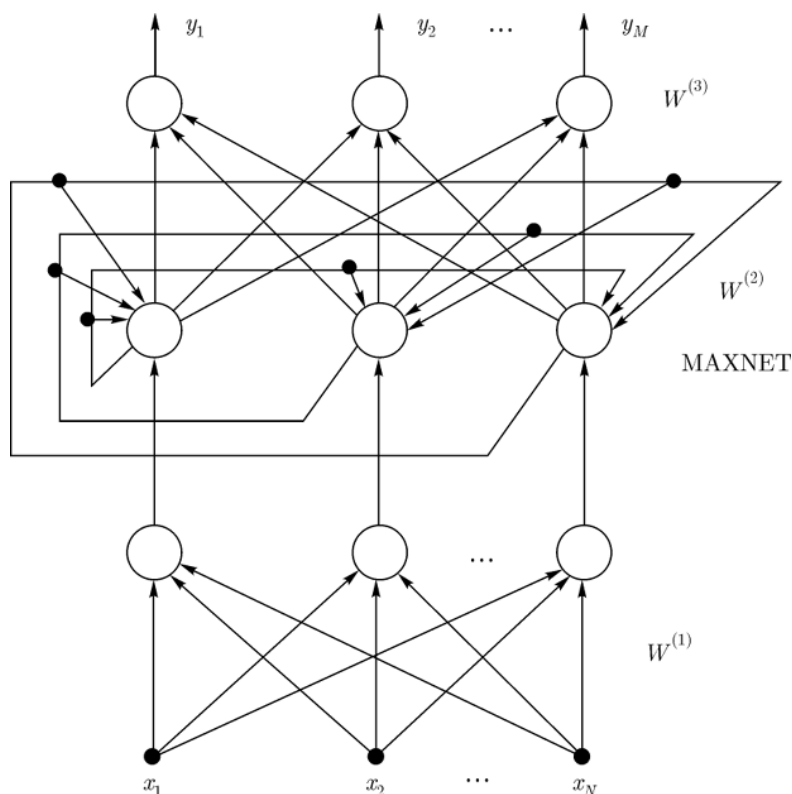


Рисунок 7.5 – Вид сети Хемминга

В наиболее полном варианте состоит из 3-х слоев. Первый слой реализует параллельное вычисление расстояний Хемминга от входного вектора до нескольких векторов. Расстояние Хемминга ( $d_H(\bar{x}_1, \bar{x}_2)$ ) между двумя бинарными векторами одинаковой длины – это число несовпадающих бит в этих векторах.

Формирование весов и смещений 1-го слоя ( $W^{(1)}$ ):

$$w_{ik}^{(1)} = \frac{x_i^k}{2}; T_k = N/2,$$

где  $i = 1 \dots N$  – номер компонента входного вектора;

$k = 1 \dots M$  – номер нейрона;

$N$  – количество элементов входного сигнала;

$M$  – количество нейронов в сети;

$w_{ik}^{(1)}$  –  $i$ -й вес  $k$ -го нейрона 1-го слоя;

$T_k$  – смещение  $k$ -го нейрона;

$x_i^k$  –  $i$ -й элемент  $k$ -го вектора-образца.

Количество нейронов ( $M$ ) равно количеству запоминаемых двоичных векторов. При подаче на вход двоичного вектора  $\bar{x}$  на выходах нейронов формируются сигналы

$$y_k^{(1)} = \sum_{i=1}^N w_{ik} x_i + T_k = N - d_H(\bar{x}^k, \bar{x}),$$

т.е. число совпадающих бит с запомненными векторами.

Второй слой представляет собой сеть MAXNET, которая определяет нейрон 1-го слоя с максимальным выходом. При этом для того, чтобы не происходило насыщение сети, величина порога активационной функции должна быть достаточно большой (обычно равна  $M$ ).

Нейрон-победитель 2-го слоя через веса линейных нейронов выходного слоя ( $W^{(3)}$ ) представляет вектор  $\bar{Y}^k$ , который соответствует вектору  $\bar{x}^k$ , признанному ближайшим к входному вектору  $\bar{x}$ . Для этого  $w_{ik}^{(3)} = y_i^k$ .

Таким образом, сеть Хемминга реализует гетероассоциативную память.

*Области применения:* распознавание образов, классификация, ассоциативная память, помехоустойчивая передача сигналов.

*Преимущества:*

1 Ёмкость сети Хемминга не зависит от размерности входного сигнала: она в точности равна количеству нейронов первого слоя;

2 Небольшое количество взвешенных связей между нейронами. Экспериментально доказано, что при одинаковом числе весов сеть Хемминга дает лучшие результаты, чем сеть Хопфилда.

*Недостатки:*

1 Сеть способна правильно распознавать только слабо зашумленные входные сигналы;

2 Если зашумленные образы находятся на одинаковом расстоянии Хемминга от двух или более эталонов, то выбор сетью Хемминга одного



из эталонов становится случайным;

3 Использование только бинарных входных сигналов.

### 7.5 Сеть теории адаптивного резонанса

Разработана Карпендером и Гроссбергом в 1986 г. Первая модель – АРТ-1 разработана для обработки двоичных входных векторов. Более позднее обобщение АРТ-2 может также работать с непрерывными векторами.

При создании ассоциативной памяти существует проблема стабильности-пластичности: память остается способной к восприятию новых образов и в то же время сохраняет стабильность, гарантирующую, что образы не разрушатся в процессе функционирования. Особенно эта проблема актуальна при работе в режиме реального времени, когда обучающее множество все время меняется.

Структурв сети теории адаптивного резонанса показан на рис. 7.6.

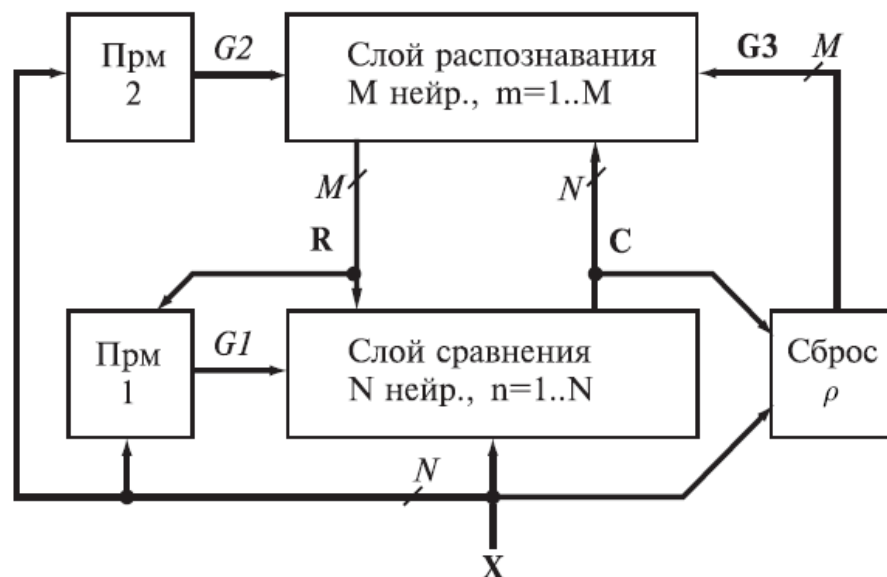


Рисунок 7.6 – Сеть теории адаптивного резонанса

**Слой сравнения.** Каждый нейрон в слое сравнения получает три двоичных входа:

- 1 Компонент  $x_i$  входного вектора  $\bar{X}$ ;
- 2 Сигнал обратной связи  $R_i$  – взвешенная сумма выходов распознающего слоя;
- 3 Вход от Приемника 1 (один и тот же сигнал подается на все нейроны этого слоя).

Порог нейронов равен 2. Поэтому, чтобы получить на выходе нейрона 1, минимум два из трех его входов должны равняться 1 (правило «двух третей»).

Структура слоя сравнения показана на рис. 7.7.

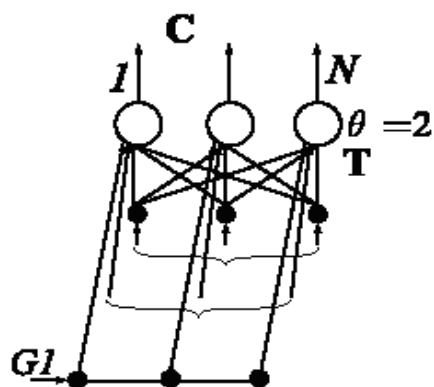


Рисунок 7.7 – Структура слоя сравнения

Работа слоя определяется формулами

$$P_n = \bar{T}^n \bar{R} = \sum_m T_m^n R_m;$$

$$NET_n = P_n + X_n + G1;$$

$$C_n = \begin{cases} 0, & NET_n < 2 \\ 1, & NET_n \geq 2. \end{cases}$$

*Слой распознавания.* В процессе функционирования каждый нейрон слоя распознавания вычисляет свертку вектора собственных весов  $\bar{V}_j$  и входного вектора  $\bar{C}$ :  $OUT_j = F(\bar{V}_j \bar{C})$ , где  $F$  – пороговая функция.

Структура слоя сравнения показана на рис. 7.8.

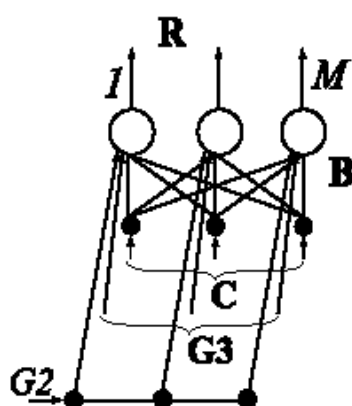


Рисунок 7.8 - Структура слоя сравнения

В отличие от слоя сравнения веса слоя распознавания непрерывные. Нейроны взаимно соединены в латерально-тормозящую сеть. Нейрон,

имеющий веса, наиболее близкие вектору  $\bar{C}$ , будет иметь самый большой выход, тем самым выигрывая соревнование и одновременно затормаживая все остальные нейроны в слое.

*Приемник 1* генерирует сигнал  $G1$ , который равен 1, если хотя бы одна компонента  $\bar{X}$  равна 1, а вектор  $\bar{R}$  – нулевой.

*Приемник 2* генерирует сигнал  $G2$ , который равен 1, если входной вектор  $\bar{X}$  имеет хотя бы одну единичную компоненту (т.е. идентифицирует подачу сигнала на вход).

*Модуль сброса.* Модуль сброса измеряет сходство между векторами  $\bar{X}$  и  $\bar{C}$ . Измерение состоит в сравнении количества единиц в векторах по формуле

$$G3_{m_0} = \begin{cases} \frac{\sum C_n}{\sum_n X_n} < \rho \\ 1, \frac{\sum C_n}{\sum_n X_n} < \rho \\ 0, \text{ иначе.} \end{cases}$$

Параметр  $\rho$  определяет уровень схожести векторов и задается в интервале от 0 до 1. Чем меньше  $\rho$ , тем менее похожие векторы будут отнесены сетью к одному классу. Если они отличаются сильнее, чем требует параметр сходства, вырабатывается сигнал сброса возбужденного нейрона в слое распознавания.

#### *Алгоритм функционирования*

##### *1 Фаза распознавания.*

- а) Подается на вход  $\bar{X}$ . При этом  $G2 = 0$  и  $\bar{R} = 0$ ;
- б) Т. к.  $\bar{X} \neq 0$ , то  $G1 = 1$ . Сигнал  $G1$  «подпитывает» нейроны слоя сравнения и  $\bar{C} = \bar{X}$ ;
- в) В слое распознавания активизируется несколько нейронов, но благодаря латеральному торможению только один нейрон под номером  $m_0$  имеет ненулевой выход. Т. е. формируется вектор  $\bar{R}$ , у которого  $R_{m_0} = 1$ , а все остальные компоненты нулевые.

##### *2 Фаза сравнения.*

- а) Т. к.  $\bar{R} \neq 0$ , то  $G1 = 0$ , что снимает «подкачку» нейронов в слое сравнения. На вход слоя сравнения подается  $R_{m_0} = 1$ . Эта единица умножается на весовые коэффициенты  $T^n$ , которые являются двоичными ядрами классов. Тогда  $NET_n = X_n + T_{m_0}^n$ . Поскольку порог равен 2, то  $C_n = X_n \wedge T_{m_0}^n$ ;

- б) Рассчитывается критерий сходства в модуле сброса.

##### *3 Фаза поиска.*

- а) Если критерий сходства выполняется, то поиск не требуется;
- б) В противном случае  $G3_{m_0} = 1$ , нейрон  $m_0$  затормаживается и шаги 1б – 2б повторяются;

в) Если ни один из нейронов не удовлетворяет критерию сходства, то считается, что входной сигнал отличается от всех образцов, и он рассматривается как новый образец. Для него в сеть вводится новый нейрон.

Фаза поиска необходима, поскольку сеть АРТ использует два критерия сходства векторов:

1 Максимум скалярного произведения векторов  $\bar{V}$  и  $\bar{X}$  при выборе победителя в слое распознавания;

2 Критерий сходства в модуле сброса.

Эти два критерия не являются эквивалентными, и может оказаться, что другой нейрон в слое распознавания будет обеспечивать более хорошее соответствие, несмотря на то, что свертка между его весовым вектором и входным вектором может иметь меньшее значение.

*Области применения:* распознавание образов, кластерный анализ.

*Преимущества:*

1 Обучение без учителя;

2 Решение дилеммы стабильности-пластичности. Новый образ может создавать дополнительные классификационные категории, однако не может изменить существующую память.

*Недостатки:*

1 В присутствии шума возникает *размножение классов*, связанное с неконтролируемым увеличением числа классов в процессе функционирования сети;

2 Чувствительность к порядку предъявления векторов. Распределение ядер классов зависит от вида упорядочения входных векторов.

*Контрольные вопросы по теме 7:*

1 Опишите устройство и работу сети Хопфилда.

2 Каковы условия устойчивости нейронных систем с обратными связями?

3 Каков принцип работы ассоциативной памяти?

4 Опишите устройство и работу двунаправленной ассоциативной памяти.

5 Каков принцип работы гетероассоциативной памяти?

6 Приведите модификации двунаправленной ассоциативной памяти.

7 Опишите устройство и работу сети поиска максимума.

8 Опишите устройство и работу сети Хемминга.

9 Опишите устройство и работу сети теории адаптивного резонанса.

## СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ

- 1 Савельев С.В. Происхождение мозга. – М.: ВЕДИ, 2005. – 368 с: ил. – ISBN 5-94624-025-0
- 2 Хайкин С. Нейронные сети: полный курс, 2-е издание.: Пер. с англ. – М.: Издательский дом "Вильямс", 2006. – 1104 с.: ил. – ISBN 5-8459-0890-6.
- 3 Заенцев И.В. Нейронные сети: основные модели. – Воронеж: ВГУ, 1999. – 157 с.
- 4 McCulloch W. S., Pitts W. H. A logical calculus of ideas immanent in nervous activity // Bull. Math. Biophysics, 1943. – Vol. 5. – P. 115-119.
- 5 Stornetta W. S., Huberman B. A. An improved three-layer, backpropagation algorithm // Proceedings of the IEEE First International Conference on Neural Networks. – San Diego: SOS Printing. – 1987. – P. 12-23.
- 6 Круглов В. В. Искусственные нейронные сети. Теория и практика / В. В. Круглов, В. В. Борисов. – [2-е изд., стереотип.]. – М.: Горячая линия-Телеком, 2002. – 382 с. – ISBN 5-93517-031-0
- 7 Уоссермен Ф. Нейрокомпьютерная техника: Теория и практика / Ф. Уоссермен; [пер. с английского Ю. А. Зуев, В. А. Точенов]. – М.: Мир, 1992. – 240 с. – ISBN 5-03-002115-9
- 8 Ежов А. А. Нейрокомпьютинг и его применения в экономике и бизнесе / А. А. Ежов, С. А. Шумский; под ред. В. В. Харитонов. – М.: МИФИ, 1998. – 224 с.
- 9 Галушкин А. И. Нейроматематика / А. И. Галушкин // Нейрокомпьютеры: разработка, применение. – 2003. – № 1. – С. 3-12.
- 10 Галушкин А. И. Теория нейронных сетей: [учебное пособие для вузов] / А. И. Галушкин. – М.: ИПРЖР, 2000. – Кн. 1. – 416 с. – ISBN 5-93108-05-8
- 11 Ахапкин Ю.К. Биотехника – новое направление компьютеризации/ Ю.К.Ахапкин, С.И.Барцев, Н.Н.Всеволодов и др. – М.: Наука, 1990. – 144 с.
- 12 Бэстенс Д.Э. Нейронные сети и финансовые рынки/ Д.Э.Бэстенс, В.М.Ван Ден Берг, Д.Вуд. – М.: ТВП «Научное издательство», 1997. – 236 с.
- 13 Галушкин А.И. Нейрокомпьютеры. – Кн.3: Учебное пособие для вузов / Под общ. ред. А.И. Галушкина. – М.: ИПРЖР, 2000. – 528 с.
- 14 Генетические алгоритмы, искусственные нейронные сети и проблемы виртуальной реальности/ Г.К. Вороновский, К.В. Махотило, С.Н. Петрашев, С.А. Сергеев. – Х.: ОСНОВА, 1997. – 112 с.
- 15 Гилл Ф. Практическая оптимизация/ Ф.Гилл, У.Мюррей, М.Райт. – М.: Мир, 1985. – 509 с.
- 16 Гитис В.Б. Применение нейросетевых технологий для нормирования времени механообработки в машиностроении/ В.Б.Гитис,

С.В.Ковалевский // VIII Всероссийская конференция «Нейрокомпьютеры и их применение»: Сборник докладов. – М.: ИПРЖР, 2002. – С. 569-572.

17 Гитис В.Б. Методика построения нейросетевых моделей типовых деталей // Искусственный интеллект: Научно-теоретический журнал. – 2004. – № 4. – С.539-544.

18 Гитис В.Б. Оценка значимости классификационных признаков детали для нейросетевого установления технических норм времени механической обработки деталей // Сборник трудов Международной научной конференции «Нейросетевые технологии и их применение 2002 – 2003». – Краматорск: ДГМА, 2003. – С.278-284.

19 Гитис В.Б. Расчет времени механической обработки ступенчатых валов с применением нейронных сетей // Материалы Международной научной конференции «Искусственный интеллект. Интеллектуальные и многопроцессорные системы-2004». – Том 1. – Таганрог: Изд-во ТРТУ, 2004. – С.478-480.

20 Гитис В.Б. Техническое нормирование на основе нейросетевых технологий // Надёжность инструмента и оптимизация технологических систем: Сб. научных трудов. – Краматорск: ДГМА, 2002. – № 12. – С.113-118.

21 Гитис В.Б. Экспресс-нормирование времени механообработки с использованием нейросетей // Сборник трудов Международной научной конференции «Нейросетевые технологии и их применение 2002 – 2003». – Краматорск: ДГМА, 2003. – С.33-39.

22 Горбань А.Н. Обучение нейронных сетей. – М.: Изд. СССР-США СП «ParaGraph», 1990. – 160 с.

23 Горбань А.Н. Нейронные сети на персональном компьютере / А.Н.Горбань, Д.А.Россиев. – Новосибирск: Наука, 1996. – 276 с.

24 Дубровін В.І. Методи оптимізації та їх застосування в задачах навчання нейронних мереж: Навчальний посібник/ В.І.Дубровін, С.О.Субботін. – Запоріжжя: ЗНТУ, 2003. – 136 с.

25 Каллан Р. Основные концепции нейронных сетей. – М: Изд. дом «Вильямс», 2001. – 287 с.

26 Колмогоров А.Н. О представлении непрерывных функций нескольких переменных в виде суперпозиций непрерывных функций одного переменного и сложения // Доклады АН СССР. – Том 114. – 1957. – № 4. – С.953-956.

27 Колмогоров А.Н. О представлении непрерывных функций нескольких переменных суперпозициями непрерывных функций меньшего числа переменных // Доклады АН СССР. – Том 108. – 1956. – № 2. – С.179-182.

28 Комашинский В.И. Нейронные сети и их применение в системах управления и связи/ В.И.Комашинский, Д.А.Смирнов. – М.: Горячая линия–Телеком, 2003. – 94 с.

29 Лысенко Н.Г. Математическое и программное обеспечение нейрокомпьютеров. – Харьков: Основа, 1991. – 91 с.

- 30 Минский М. Персептроны/ М.Минский, С.Пайперт. – М.: Мир, 1971. – 261 с.
- 31 Миркес Е.М. Нейрокомпьютер: проект стандарта. – Новосибирск: Наука, 1999. – 337 с.
- 32 Нейроинформатика / А.Н.Горбань, В.Л.Дунин-Барковский, А.Н. Кирдин и др. – Новосибирск: Наука. Сибирская издательская фирма РАН, 1998. – 296 с.
- 33 Нейронні мережі в системах автоматизації / В.І.Архангельський, І.М.Богаєнко, Г.Г. Грабовський, М.О. Рюмшин. – К.: Техніка, 1999. – 354 с.
- 34 Осовский С. Нейронные сети для обработки информации/ Пер. с польского И.Д. Рудинского. – М.: Финансы и статистика, 2002. – 344 с.
- 35 Сигеру Омату. Нейроуправление и его приложения. – Кн. 2 / Сигеру Омату, Марзуки Халид, Рубия Юсоф; Пер. с англ. Н.В. Батина; Под ред. А.И.Галушкина, В.А.Птичкина. – М.: ИПРЖР, 2000. – 272 с.

*Навчальне видання*

**ГІТІС Веніамін Борисович**

## **Теорія і практика застосування нейронних мереж**

### **Навчальний посібник**

*(Російською мовою)*

Редактор

Комп'ютерна верстка

О. П. Ордіна

107/2010. Підп. до друку \_\_\_\_\_. Формат 60 x 84/8.  
Папір офсетний. Ум. друк. арк. \_\_\_\_\_. Обл.-вид. арк. \_\_\_\_\_.  
Тираж 100 прим. Зам. № \_\_\_\_\_.

Донбаська державна машинобудівна академія  
84313, м. Краматорськ, вул. Шкадінова, 72.  
Свідоцтво про внесення суб'єкта видавничої справи  
до Державного реєстру  
серія ДК №1633 від 24.12.2003