Министерство образования и науки Российской Федерации
Южно-Уральский государственный университет
Кафедра «Прикладная математика»

004(07)
К885

Б.М. Кувшинов

НЕЙРОННЫЕ СЕТИ

Учебное пособие

Челябинск
Издательский центр ЮУрГУ
2015

УДК 004.032.26(075.8)+004.8.032.26(075.8)
К885

Одобрено учебно-методической комиссией факультета математики, механики и компьютерных наук.

Рецензенты:
А.А. Соловьев, А.Г. Комирев

Кувшинов, Б.М.
К885 Нейронные сети: учебное пособие / Б.М. Кувшинов. – Челябинск: Издательский центр ЮУрГУ, 2015. – 66 с.

Пособие содержит обзорные учебные материалы по принципам построения и машинного обучения искусственных нейронных сетей. Рассмотрены основные архитектуры сетей и алгоритмы их обучения, используемые при решении задач оценивания, прогнозирования и поддержки принятия решений.

Учебное пособие предназначено для студентов направления подготовки 01.03.04 при изучении курса «Нейронные сети», а также при подготовке выпускных квалификационных работ, в которых используются модели нейронных сетей.

ВВЕДЕНИЕ

Учебное пособие предназначено для студентов направления подготовки 01.03.04 «Прикладная математика» при изучении курса «Нейронные сети», а также при подготовке выпускных квалификационных работ, в которых используются модели нейронных сетей. Пособие содержит обзорные учебные материалы по принципам построения и машинного обучения искусственных нейронных сетей. Рассмотрены основные архитектуры сетей и алгоритмы их обучения, используемые при решении задач оценивания, прогнозирования и поддержки принятия решений.

В первой главе обсуждаются общие свойства математических моделей, относящихся к категории искусственных нейронных сетей. Рассмотрены различные свойства биологического прототипа, которые воспроизведены в известных на сегодняшний день моделях.

Вторая и третья главы посвящены двум большим категориям нейронных сетей: сетям без обратных связей и сетям с обратными связями. Для этих категорий сетей рассмотрены наиболее известные и активно применяемые на практике архитектуры: многослойный перцептрон, сеть встречного распространения, сеть радиальных базисных функций, сверточные сети, сети Хопфилда, Хэмминга, Коско, сеть адаптивного резонанса. По каждой теме изложение дано в соответствии с единой схемой: структура сети, алгоритм обучения сети, особенности использования сети при решении прикладных задач.

В четвертой главе рассмотрены вопросы, имеющие общее значение для различных видов сетей: обеспечение глобального оптимума при обучении и использовании обученной сети, совместное использование нейронных сетей, генетических алгоритмов и методов нечеткой логики. Материал этого раздела предполагает, что читатели уже имеют базовые представления о методах нечеткой логики, методах эволюционного обучения.

В конце каждой главы приведен список контрольных вопросов, которые помогают уяснить взаимосвязь рассматриваемых вопросов, понять достоинства и недостатки нейронных сетей с различной архитектурой.

Глава 1. ОБЩАЯ ХАРАКТЕРИСТИКА ИНС

1.1. ИНС как упрощенная модель биологической системы

1.1.1. Моделируемые свойства биологической системы

Искусственные нейронные сети (ИНС) возникли как попытка реализации простой идеи: для создания искусственной системы, способной выполнять разумные рассуждения, необходимо повторить структурную организацию естественной нервной системы человека. В целом эта гипотеза оказалась далека от реальности, однако сам принцип: построение сложной системы из множества простых однотипных элементов – показал свою жизнеспособность и привел к появлению большого количества родственных математических моделей, обозначаемых сегодня термином «искусственная нейронная сеть».

С точки зрения организации вычислительного процесса биологическая нейронная сеть – это большое количество однотипных простых элементов (нейронов), между которыми по огромному количеству связей (синапсов) передаются сигналы (рис. 1.1).

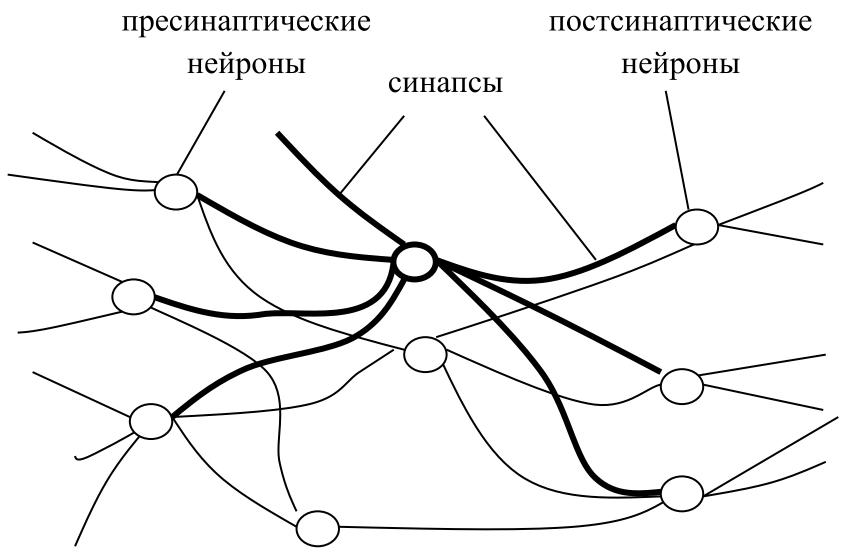


Рис. 1.1. Фрагмент биологической нейронной сети

Состояние каждого нейрона характеризуется текущим уровнем его выходного сигнала (возбуждения). Уровень возбуждения меняется во времени в зависимости от следующих факторов:

а) уровень сигналов, поступающих на данный нейрон от других (пресинаптических) нейронов;

б) сила связей данного нейрона с пресинаптическими нейронами.

Каждый нейрон, в свою очередь, передает сигнал, соответствующий текущему уровню его возбуждения, на другие (постсинаптические) нейроны. На некоторые нейроны входные сигналы поступают «извне», от органов чувств, с некоторых нейронов выходные сигналы отправляются «вовне», к исполнительным механизмам, обеспечивающим взаимодействие организма с внешней средой.

Таким образом, нейронную сеть можно рассматривать как сложное функциональное преобразование входного сигнала в выходной сигнал. В рамках существующих моделей ИНС предполагается, как правило, что зависимости вида (б) – линейные, а зависимости вида (а) – нелинейные (обычно пороговые). Поэтому реализуемое функциональное преобразование может быть сколь угодно сложным – для этого необходимо лишь, чтобы сеть включала в себя достаточное количество нейронов и связей между ними.

Сила связей между нейронами также меняется во времени – в зависимости от уровня сигналов, передаваемых по этим связям. В результате возникает задача настройки (обучения) сети: установить такую силу связей между отдельными нейронами, чтобы в ответ на различные входные сигналы сеть генерировала на выходе некоторые «желаемые» выходные сигналы, обеспечивающие наилучшую реакцию организма на текущее состояние внешней среды.

Приведенная модель нейронной сети содержит существенные допущения по отношению к своему биологическому прототипу. В частности, не принимаются во внимание задержки во времени, которые воздействуют на динамику системы, не учитывается воздействий функции частотной модуляции, синхронизирующей динамику нейронов, и т.д. Кроме того, реальная нервная система включает порядка 10^{11} нейронов, которые участвуют примерно в 10^{15} передающих связях. Искусственные нейронные сети имеют гораздо меньший объем – от единиц до тысяч нейронов.

Несмотря на эти ограничения, искусственные нейронные сети демонстрируют многие важные свойства, присущие мозгу: обучаются на основе опыта, обобщают предыдущие прецеденты на новые случаи, извлекают существенные свойства из поступающей информации, содержащей излишние данные.

1.1.2. Формальная модель нейрона

Отдельный нейрон в составе ИНС – это функциональное преобразование вида:

$$O = \mathrm{F}(<\mathbf{X};\mathbf{W}>), \qquad (1.1)$$

где $\mathbf{X} \in \mathbf{R}^N$ – входные сигналы, поступающие на данный нейрон с N пресинаптических нейронов (либо со входов всей сети); $\mathbf{W} \in \mathbf{R}^N$ – веса пресинаптических связей; F – нелинейное функциональное преобразование (называемое активационной или пороговой функцией нейрона); $O \in \mathbf{R}$ – выходной сигнал нейрона, который передается на постсинаптические нейроны (или на выход всей сети).

В некоторых моделях ИНС на входные, выходные сигналы и веса связей нейронов накладываются более строгие ограничения, например:

$$\mathbf{X}, \mathbf{W} \in \{0,1\}^N, O \in \{0,1\}.$$

Наиболее употребительные варианты активационной функции:

1) ступенчатая:

$$O = \begin{cases} C, \text{если} < \mathbf{X}; \mathbf{W} >> T; \\ 0, \text{иначе}; \end{cases} \quad (1.2)$$

где C, T – константы;

2) логистическая функция:

$$O = \frac{1}{1 + \mathrm{e}^{<\mathbf{X};\mathbf{W}>}}; \quad (1.3)$$

3) гиперболический тангенс:

$$O = \mathrm{th}(< \mathbf{X}; \mathbf{W} >) = \frac{\mathrm{e}^{<\mathbf{X};\mathbf{W}>} - \mathrm{e}^{-<\mathbf{X};\mathbf{W}>}}{\mathrm{e}^{<\mathbf{X};\mathbf{W}>} + \mathrm{e}^{-<\mathbf{X};\mathbf{W}>}}. \quad (1.4)$$

Внешний вид перечисленных активационных функций приведен на рис. 1.2. Все они обладают двумя важными свойствами.

1. Нелинейность. Если бы все нейроны в сети имели линейную активационную функцию, последовательное распространение сигнала по сети сводилось бы к его масштабированию. В таком случае увеличение мощности (количества нейронов в сети) не приводило бы к дополнительным эффектам. Нелинейные активационные функции нейронов позволяют делать функциональную зависимость, воспроизводимую нейронной сетью, все более сложной после прохождения сигнала через каждый нейрон.

2. Ограниченная область допустимых значений при неограниченной области определения. Даже при большом количестве пресинаптических нейронов, сигналы которых суммируются в данном нейроне, величина выходного сигнала остается в заданных пределах. Поэтому количество связей между нейронами может быть сколь угодно большим, вплоть до полносвязной архитектуры, когда каждый нейрон является пресинаптическим и постсинаптическим для всех остальных нейронов сети.

Стоит отметить, что существует целый ряд моделей ИНС, в которых рассмотренное определение нейрона формально не соблюдается. Речь идет о том, что выходы разных нейронов являются зависимыми друг от друга. Т.е. существует некоторое дополнительное, по отношению к (1.1), правило, общее для целой группы нейронов. Например: только один нейрон из группы может иметь выходной сигнал, отличный от 0. Подробнее этот момент рассмотрен в главе 3.

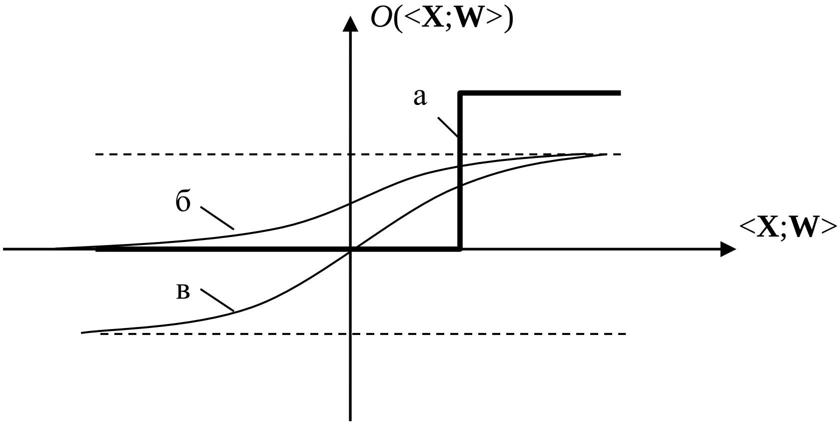


Рис. 1.2. Типовые варианты активационной функции нейрона:
а – ступенчатая; б – логистическая; в – гиперболический тангенс

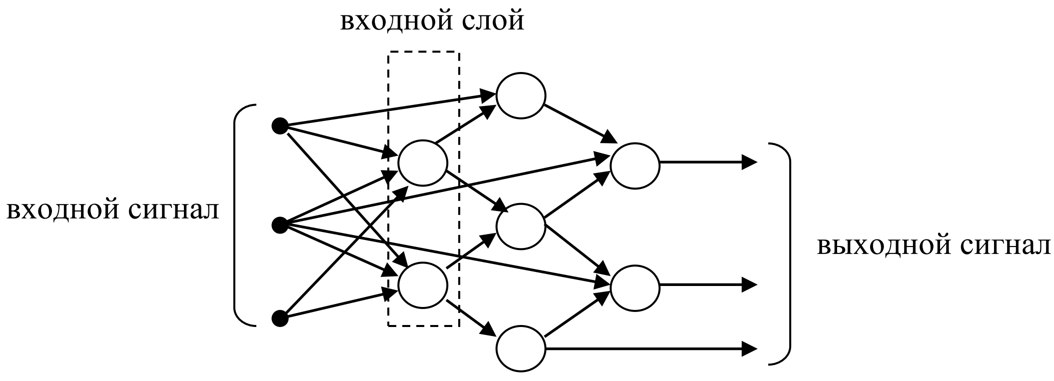
1.1.3. Топология нейронной сети

Хотя отдельный нейрон и способен выполнять простейшее преобразование сигнала, построение эффективной ИНС требует включение в сеть большого количество нейронов, как правило однотипных, либо разделенных на несколько групп разных типов. Поведение сети определяется не только видом используемых в ней нейронов, но и характером связей между ними. С точки зрения топологии связей можно выделить две большие категории сетей.

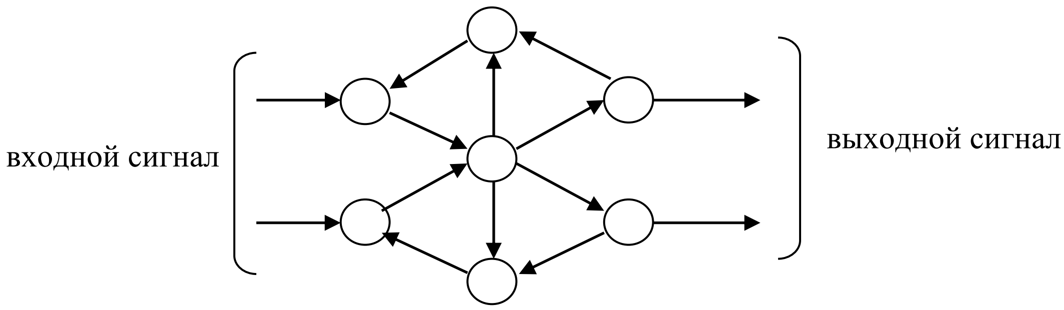
1. Слоистые сети без обратных связей (сети прямого распространения) (рис. 1.3,а). Все множество нейронов разделено на несколько последовательных слоев. Один из слоев является «входным» – т.е. на его нейроны подаются только входные сигналы сети. Выходные сигналы сети могут сниматься как с одного специально выделенного слоя, так и с разных слоев сети. Порядок связей между нейронами таков, что сигнал не может распространяться циклически: выходной сигнал любого нейрона не попадает на его вход ни напрямую, ни косвенно, после прохождения через другие нейроны. Сеть без обратных связей явным образом задает функциональное преобразование и для нее можно решать задачу аппроксимации такого функционального преобразования по эмпирическим данным. Обсуждению сетей такого вида посвящена глава 2.

2. Сети с обратными связями (рис. 1.3,б). В графе, соответствующем нейронной сети, имеется хотя бы один цикл. В результате текущее состояние нейрона, включенного в цикл, зависит не только от сигналов, полученных извне (со входа сети или с других нейронов), но и от его собственного состояния в предыдущие моменты времени. Таким образом, сеть с обратными связями не вполне соответствует понятию функционального преобразования: выходной сигнал всей сети зависит не только от входного сигнала, но и от времени. Интерес представляют,

прежде всего, устойчивые сети, в которых выходной сигнал после некоторого количества итераций сходится к постоянным значениям. В этом случае можно говорить о функциональной зависимости установившегося выходного сигнала от входного сигнала. Чаще всего этот процесс интерпретируется как «ассоциативная память» ИНС. Сети такого вида обсуждаются в главе 3.



а)



б)

Рис. 1.3. Варианты топологии нейронной сети:
а) слоистая сеть; б) сеть с обратными связями

1.2. Задачи искусственной нейронной сети

1.2.1. Обучение и использование обученной сети

Конечная цель использования ИНС – обеспечить «желаемый» выходной сигнал при подаче входного сигнала из некоторого допустимого множества. В рамках такой задачи нейронная сеть может применяться как инструмент аппроксимации сложной функциональной зависимости в самых разных задачах поддержки принятия решения, оценивания и управления. Понятия «входной сигнал» и «желаемый выходной сигнал» определяются здесь конкретной прикладной задачей. Сеть, в любом случае, должна воспроизводить «желаемый» выходной сигнал не только для известных входных сигналов, но и для других, новых входных значений из некоторого допустимого множества.

По этой причине в использовании ИНС можно выделить две фазы.

1. Обучение сети на конечном наборе прецедентов. Сети предъявляется обучающая выборка, т.е. некоторое количество образцов, которые представляют собой пары (входной сигнал; выходной сигнал), либо только входные сигналы. Задача обучения – таким образом настроить веса связей между нейронами, чтобы воспроизводимая сетью функциональная зависимость соответствовала предъявленной выборке.

2. Использование обученной сети на новых входных сигналах (распознавание). На вход уже обученной сети поступают входные сигналы, соответствующие новым прецедентам, не обязательно представленным в обучающей выборке. Сеть генерирует выходной сигнал, согласно функциональной зависимости, которая зафиксирована в ней в форме значений весов связей между нейронами.

Нередко вторая фаза дополняется дообучением сети: веса связей незначительно корректируются, чтобы учесть дополнительную информацию о свойствах восстанавливаемой функциональной зависимости, представленную в предъявленном примере.

С точки зрения количества информации, доступной для обучения, различают режимы обучения ИНС с учителем и без учителя. Обучение с учителем предполагает, что для каждого входного вектора в обучающей выборке существует целевой вектор, представляющий собой требуемый выход. Разница между «желаемым» и фактическим выходом сети для заданного входа используется как критерий оптимизации при изменении весов связей между нейронами. Обучение может быть итерационным – когда примеры предъявляются поочередно и каждый раз незначительно изменяют уже полученные значения весов, либо единовременным – когда все веса связей в сети однозначно определяются множеством прецедентов.

Несмотря на многочисленные прикладные достижения, обучение с учителем достаточно далеко от биологического прототипа. Обучение без учителя – это режим, в котором обучающая выборка состоит только из входных векторов. Обучающий алгоритм должен настроить веса связей в сети так, чтобы получались согласованные выходные векторы, т. е. чтобы предъявление достаточно близких входных векторов давало одинаковые или близкие выходы.

1.2.2. Функциональные свойства ИНС

К настоящему времени разработано большое количество различных конфигураций ИНС и правил их настройки, позволяющих решать задачи обучения с учителем или самообучения. Все эти решения направлены на то, чтобы обеспечить нейронную сеть рядом свойств, необходимых для успешного решения с их помощью прикладных задач. Можно указать следующие важные свойства, присущие различным архитектурам ИНС.

1. Обучение. Искусственные нейронные сети могут менять свое поведение в зависимости от внешней среды. После предъявления входных сигналов (возможно, вместе с требуемыми выходами) они самонастраиваются, чтобы обеспечивать требуемую реакцию. Было разработано множество обучающих алгоритмов, каждый со своими сильными и слабыми сторонами. При этом все еще существуют проблемы относительно того, чему сеть может обучиться и как обучение должно проводиться.

2. Обобщение. Отклик сети после обучения может быть до некоторой степени нечувствителен к небольшим изменениям входных сигналов. Эта внутренне присущая способность видеть образ сквозь шум и искажения важна для распознавания образов в реальных прикладных задачах. Важно, что ИНС делает обобщения автоматически, благодаря своей структуре, а не с помощью специально написанных алгоритмов, привязанных к конкретной прикладной задаче.

3. Абстрагирование. Некоторые из искусственных нейронных сетей обладают способностью извлекать сущность из входных сигналов. Например, сеть может быть обучена на последовательность искаженных изображений буквы «А». После соответствующего обучения предъявление такого искаженного примера приведет к тому, что сеть породит изображение буквы правильной формы. В некотором смысле она научится порождать то, что никогда не видела.

1.3. Место ИНС в задачах прикладной математики

В 50–60 г.г. 20 века были созданы первые ИНС – сначала как электронные сети, а позднее как среда компьютерного моделирования. Первые успехи вызвали взрыв активности и оптимизма. Такие исследователи как М. Минский, Ф. Розенблатт, Б. Уидроу и другие разработали нейронные сети, состоящие из одного слоя нейронов [7]. Часто называемые перцептронами, они были использованы для такого широкого класса задач, как предсказание погоды, анализ электрокардиограмм и искусственное зрение. В течение некоторого времени казалось, что найден универсальный принцип построения искусственного интеллекта и воспроизведение человеческого мозга является лишь вопросом конструирования достаточно большой сети.

Однако затем эта иллюзия рассеялась: сети не могли решать задачи, внешне весьма сходные с теми, которые они успешно решали. С этих необъяснимых неудач начался период интенсивного анализа. М.Минский, используя точные математические методы, строго доказал ряд теорем, относящихся к функционированию сетей [5]. В частности было доказано, что однослойные сети теоретически неспособны решить многие простые задачи, в том числе реализовать функцию «исключающее ИЛИ».

Тем не менее, ряд исследователей, таких как Т. Кохонен, С. Гросберг, Дж. Андерсон продолжили развивать теорию нейронных сетей [4, 10]. Постепенно появился теоретический фундамент, на основе которого сегодня конструируются наиболее мощные многослойные сети. Оценка М. Минского оказалась излишне пессимистичной, многие из поставленных в его работе задач решаются сейчас сетями с помощью стандартных процедур.

На сегодня нейронные сети занимают определенную нишу среди других инструментов обработки информации и на практике их успешно комбинируют с другими концепциями извлечения и анализа знаний [1, 2, 8, 11, 13]. Как правило, ИНС отвечает за низкоуровневую обработку информации, требующую быстрой реакции. Когда ИНС не в состоянии извлечь знания из анализируемых данных и систематизировать их, задача передается для обработки другими методами. Их применение может потребовать дополнительной информации и больших ресурсов, но качество полученных в результате решений может быть выше.

Контрольные вопросы к главе 1

1. Какие аспекты функционирования биологической нейронной сети не учитываются в существующих моделях ИНС? Чем это обусловлено?
2. С какой целью в нейронах используется нелинейная активационная функция?
3. Какие допущения используются в классической модели нейрона по отношению к его биологическому прототипу?
4. В чем состоит принципиальное отличие в поведении сетей с обратными связями и сетей, не имеющих в своей структуре таких связей?
5. Могут ли фазы обучения сети и распознавания быть совмещены во времени?
6. Какие критерии качества обучения используются при обучении сети с учителем и в режиме самообучения?
7. Каким образом использование нейронных сетей может быть комбинировано с применением других процедур извлечения знаний из данных?

Глава 2. СЕТИ ПРЯМОГО РАСПРОСТРАНЕНИЯ

2.1. Многослойный перцептрон

2.1.1. Структура многослойного перцептрона

Многослойный перцептрон – это слоистая нейронная сеть без обратных связей, на которую наложены следующие ограничения (рис. 2.1):

– все нейроны соответствуют классической модели (1.1); чаще всего они имеют одинаковую активационную функцию, поэтому поведение каждого нейрона определяется исключительно значениями весов его пресинаптических связей;

– множество нейронов разбито на конечное число последовательных слоев; выходной сигнал со всех нейронов данного слоя передается на все нейроны следующего слоя;

– связей между нейронами одного слоя и между нейронами несмежных слоев нет;

– каждая компонента входной сигнала поступает на все нейроны первого слоя, выходной сигнал снимается со всех нейронов последнего слоя.

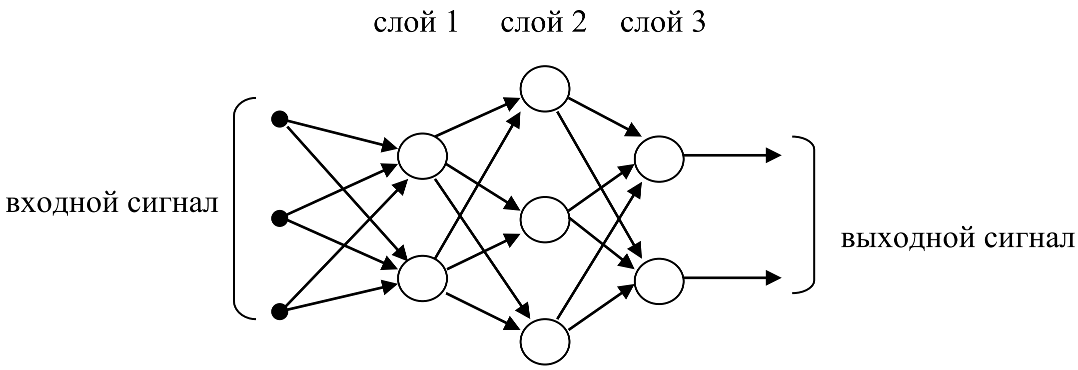


Рис. 2.1. Структура многослойного перцептрона

Правила распространения сигнала для данной структуры сети определяются естественным образом: сначала, согласно значениям входных сигналов сети, вычисляются выходы всех нейронов первого слоя, затем они используются для вычисления выходов всех нейронов второго слоя и т.д. вплоть до вычисления выходных сигналов всей сети на последнем слое.

Если все нейроны используют одинаковую функцию активации, то преобразование сигнала, выполняемое сетью, удобно представить в матричной форме:

$$\mathbf{O}_1 = \mathrm{F}(\mathbf{W}_1\mathbf{X});\, \mathbf{O}_2 = \mathrm{F}(\mathbf{W}_2\mathbf{O}_1);\, \mathbf{O}_3 = \mathrm{F}(\mathbf{W}_3\mathbf{O}_2);\, \ldots,\, \mathbf{Y} = \mathbf{O}_N = \mathrm{F}(\mathbf{W}_N\mathbf{O}_{N-1}), \quad (2.1)$$

где $\mathbf{X}$ – вектор входных сигналов сети; $\mathbf{O}_i$ – вектор выходных сигналов i-го слоя; $\mathbf{Y}$ – вектор выходных сигналов сети; $\mathbf{W}_i$ – матрица весовых коэффициентов связей между нейронами (i–1)-го и i-го слоев; F – активационная функция нейронов.

2.1.2. Обобщающая способность сети: пример

Задача сети – представить некоторую сложную функциональную зависимость. Выше уже было отмечено, что ИНС позволяет задать сколь угодно сложную зависимость – для этого необходимо включить в сеть достаточное количество нейронов и присвоить правильные значения весам связей между ними. Рассмотрим частный случай, наглядно демонстрирующий зависимость обобщающей способности многослойного перцептрона от количества слоев и нейронов в сети.

Пусть на вход сети поступают сигналы $\mathbf{X} \in \mathbf{R}^2$, а желаемые выходы $\mathbf{Y} \in \{0,1\}$. Тогда множество объектов обучающей выборки $\left\{(\mathbf{X}_i, \mathbf{Y}_i) \in \mathbf{R}^2 \times \{0,1\}, i = \overline{1, N}\right\}$ можно интерпретировать как множество точек на плоскости, каждая из которых отнесена к одному из двух классов, согласно значению $\mathbf{Y}_i$.

Пусть сеть сначала состоит из единственного нейрона (рис. 2.2, а). Он имеет два входа и ступенчатую активационную функцию вида:

$$O = \begin{cases} 1, \text{если} < \mathbf{X}; \mathbf{W} >> 0{,}5; \\ 0, \text{иначе.} \end{cases}$$

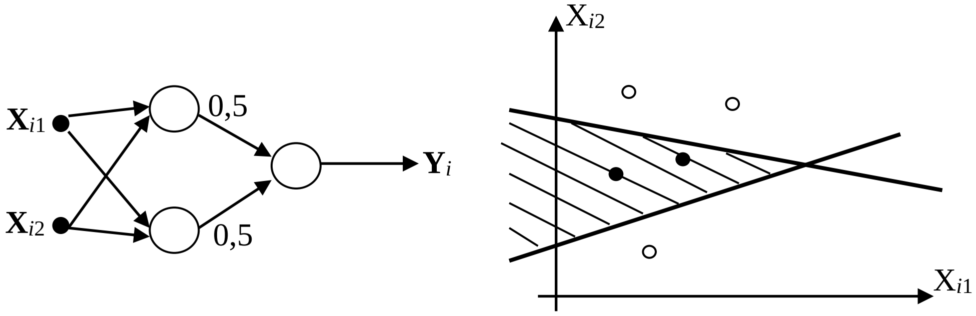
Тогда уравнение $< \mathbf{X}; \mathbf{W} >= 0{,}5$ определяет границу на плоскости $\mathbf{X} \in \mathbf{R}^2$ между областями, которые нейронная сеть относит к разным классам (рис. 2.2, б). При любых значениях $\mathbf{W} \in \mathbf{R}^2$ эта граница является линейной, а каждый класс представляет собой полуплоскость. Поэтому нейронная сеть, состоящая из одного нейрона, принципиально не способна описать, например, функцию от 2 аргументов «исключающее ИЛИ».

Рис. 2.2. Нейронная сеть с одним нейроном:
а) структура сети; б) классифицируемые образы

Пусть теперь сеть включает 2 слоя (рис. 2.3,а). Первый слой состоит из 2 нейронов, аналогичных рассмотренному выше. Для нейрона второго слоя оба веса связей равны 0,5, а активационная функция имеет вид:

$$O = \begin{cases} 1, \text{если} < \mathbf{X}; \mathbf{W} >> 0{,}75; \\ 0, \text{иначе}. \end{cases}$$

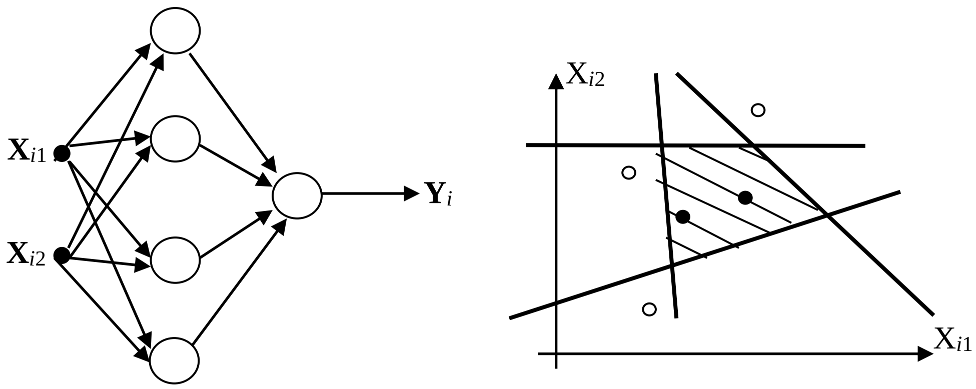
Поскольку нейроны первого слоя имеют выходные сигналы из множества $\{0,1\}$, нейрон второго слоя с указанными параметрами реализует функцию логическое И. Тогда класс, соответствующий выходному значению сети, равному 1 – это пересечение полуплоскостей, заданных каждым из нейронов первого слоя (рис. 2.3,б).



а) б)

Рис. 2.3. Нейронная сеть с тремя нейронами:
а) структура сети; б) классифицируемые образы

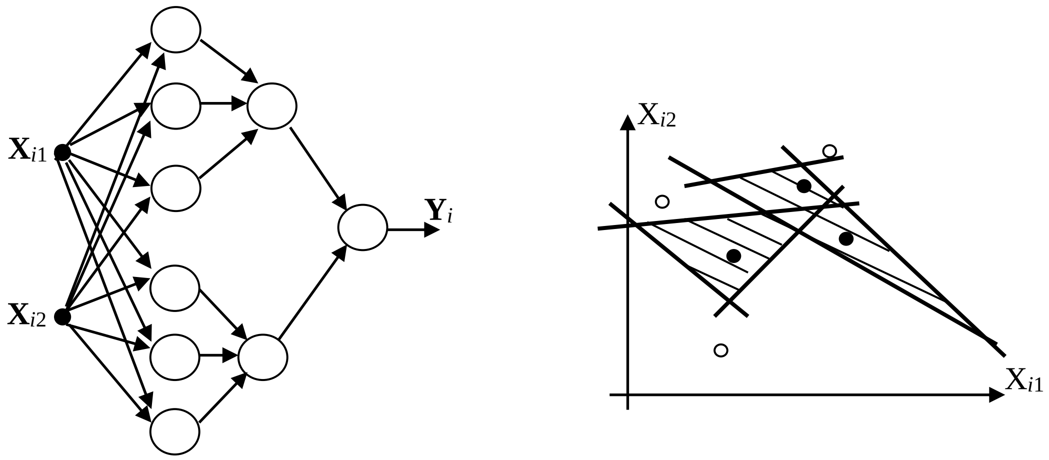
Добавив в первый слой дополнительные нейроны и модифицировав соответствующим образом активационную функцию нейрона второго слоя, получим перцептрон, описывающий на плоскости класс в форме многогранника (рис. 2.4).



а) б)

Рис. 2.4. Нейронная сеть с 2 слоями и 5 нейронами:
а) структура сети; б) классифицируемые образы

Далее объединим несколько перцептронов такого вида с помощь нейрона третьего слоя, реализующего функцию ИЛИ от выходных сигналов нейронов второго слоя. Такая нейронная сеть может описывать на плоскости класс, имеющий форму многогранника произвольного вида (рис. 2.5).



а) б)

Рис. 2.5. Нейронная сеть с 3 слоями и 9 нейронами:
а) структура сети; б) классифицируемые образы

Пусть, наконец, весовые коэффициенты связей для всех слоев будут принимать произвольные значения, а активационные функции нейронов будут иметь непрерывную область допустимых значений. Тогда функциональная зависимость, представляемая перцептроном, будет еще более сложной и потеряет геометрическую интерпретацию.

2.1.3. Алгоритм обучения перцептрона

Перцептрон обучается в режиме обучения с учителем. Для этого используется итерационная процедура, которая опирается на следующий базовый принцип. Очередной образ **X** подается на вход сети и вычисляется ее выход **Y**. Если **Y** совпадает с желаемым выходом, указанным для данного прецедента в обучающей выборке, то настройка сети не меняется. В противном случае веса всех связей в сети модифицируются, причем величина их коррекции пропорциональна:

а) величине ошибки на соответствующем выходе сети;

б) величине сигнала, переданного по соответствующей связи.

Для одиночного нейрона такая процедура реализуется следующим алгоритмом (алгоритм линейной коррекции).

1. Инициализировать вектор весов **W** случайным образом.
2. Подать очередной входной образ $\mathbf{X}_i$ и вычислить выход сети:

$$O = \mathrm{F}(\langle \mathbf{W}; \mathbf{X}_i \rangle).$$

3. Вычислить ошибку выхода:

$$D = Y_i - O.$$

4. Модифицировать веса по правилу:

$$\Delta \mathbf{W} = \eta \mathbf{X} D,$$

где η – коэффициент скорости обучения, который в процессе обучения должен постепенно снижаться от 1 к 0.

5. Выбрать следующий обучающий пример и перейти к шагу (2).

Коэффициент скорости обучения, который называют также коэффициентом обжига, предназначен для того, чтобы постепенно снижать влияние очередного обучающего примера на параметры сети. Чем больше итераций алгоритма уже совершено, тем в большей степени веса связей в сети определяются данными всей обучающей выборки. Соответственно, очередной единичный прецедент должен в меньшей степени влиять на эти веса.

Для однослойного перцептрона общего вида алгоритм обучения аналогичен, за исключением следующих моментов:

– выходной сигнал сети также является вектором;

– веса связей определяются матрицей $\mathbf{W}$ с размерностью $N \times M$, где N – размерность входного вектора $\mathbf{X}$, M – количество нейронов, совпадающее с размерностью выходного вектора $\mathbf{Y}$.

В случае многослойного перцептрона обучение происходит по аналогичному принципу, однако возникает еще две особенности.

1. Вклад в ошибку, возникшую для очередного обучающего примера на выходном слое, должен распределяться по нейронам предыдущих слоев пропорционально их вкладу в результирующий сигнал, т.е. пропорционально весам соответствующих связей между нейронами.

2. Необходимо учитывать, что в каждом слое сигнал подвергается нелинейному преобразованию с помощью активационных функций. Поэтому для определения вклада каждого нейрона предыдущего слоя в ошибку нейрона данного слоя размер ошибки дополнительно нужно умножать на производную от активационной функции. Последнее означает, что активационные функции всех нейронов должны быть дифференцируемы. В частности, для наиболее часто используемой активационной функции вида (1.3) производная равна $\mathbf{O}(1-\mathbf{O})$.

Пусть многослойный перцептрон включает N слоев с весовыми матрицами $\mathbf{W}_1,\ldots,\mathbf{W}_N$ и активационными функциями нейронов вида (1.3). Алгоритм обратного распространения ошибки, соответствующий описанным выше требованиям, имеет следующий вид.

1. Инициализировать матрицы весов $\mathbf{W}_1,\ldots,\mathbf{W}_N$ случайным образом.

2. Подать очередной входной образ $\mathbf{X}_i$ и вычислить выходы каждого слоя и выход сети согласно формулам (2.1).

3. Вычислить ошибки нейронов выходного слоя:

$$\mathbf{D}_N = (\mathbf{Y}_i - \mathbf{O}_N)\mathbf{O}_N(\mathbf{I} - \mathbf{O}_N), \tag{2.2}$$

где $\mathbf{I}$ – единичная матрица.

4. Модифицировать веса связей выходного слоя согласно правилам:

$$\Delta \mathbf{W}_N = \eta \mathbf{O}_N \mathbf{D}_N^T. \qquad (2.3)$$

5. Поочередно рассчитать ошибки и модифицировать веса связей для слоев $N-1,\ldots,1$ согласно правилам:

$$\begin{cases} \mathbf{D}_n = (\mathbf{W}_{n+1}\mathbf{D}_{n+1})\mathbf{O}(\mathbf{I}-\mathbf{O}_n); \\ \Delta \mathbf{W}_n = \eta \mathbf{O}_{n-1} \mathbf{D}_n^T; \\ n = N-1,\ldots,1, \end{cases} \qquad (2.4)$$

где $\mathbf{O}_0 = \mathbf{X}$.

6. Выбрать следующий обучающий пример и перейти к шагу (2).

Таким образом, при прямом проходе по слоям сети $1,\ldots,N$ распространяется сигнал, а при обратном проходе по слоям $N,\ldots,1$ – ошибки. По этой причине алгоритм и получил свое название.

На рис. 2.6 приведены условные обозначения, использованные в описании алгоритма.

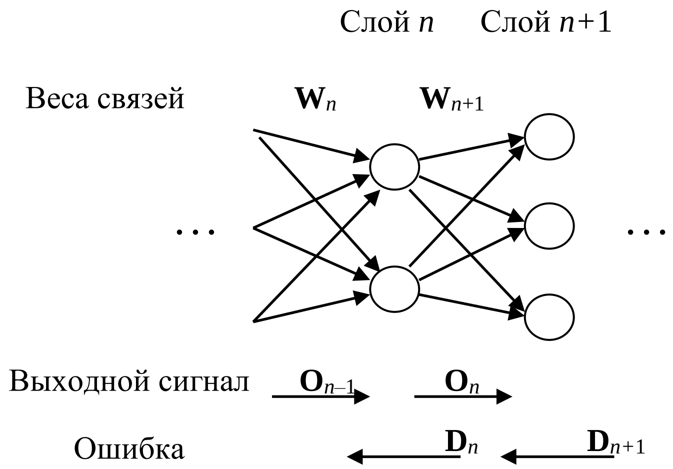


Рис. 2.6. Условные обозначения для алгоритма обратного распространения ошибки

Все объекты обучающей выборки подаются на вход сети циклически и процесс повторяется до тех пор, пока не будет достигнуто одно из следующих условий останова:

– суммарная ошибка сети на всех объектах стала меньше заданного порога;

– на протяжении некоторого количества последних итераций изменение суммарной ошибки не превышает некоторой величины.

2.1.4. Проблемы алгоритма обратного распространения и способы улучшения сходимости

Рассмотренная процедура фактически представляет собой реализацию метода градиентного спуска для оптимизируемой многомерной и многопараметрической функции, представленной нейронной сетью. Причем оптимизации подвергается не сама функция, а ошибка, возникающая на выходе сети относительно очередного обучающего примера. Поскольку на каждом шаге обучения сети предъявляется новый пример, критерий оптимизации все время меняется. Тем не менее, циклическая подача на вход всех примеров обучающей выборки дает возможность постепенно изменять веса связей сети в сторону локального минимума суммарной ошибки по всем объектам выборки.

Алгоритм обратного распространения ошибки получил широкое распространение при решении самых разных прикладных задач с помощью нейронных сетей [3, 6, 9, 12]. Однако метод градиентного спуска способен обнаруживать только локальный минимум оптимизируемой функции и плохо приспособлен для поиска минимумов в «овражных» функциях. В то же время нейронная сеть – это модель с очень большим количеством параметров, задающая многоэкстремальные овражные функции. Поэтому использование алгоритма обратного распространения часто сопряжено с несколькими типовыми проблемами.

1. Паралич сети. В процессе обучения сети значения весов связей могут, в результате коррекции, стать очень большими величинами. Это может привести к тому, что все или большинство нейронов будут функционировать при больших значениях аргумента активационной функции, где производная активационной функции очень мала. Поскольку посылаемая обратно в процессе обучения ошибка пропорциональна этой производной, процесс обучения может практически остановиться. Обычно этого избегают уменьшением коэффициента скорости обучения η, но в таком случае увеличивается время обучения.

2. Попадание в локальные минимумы. Функция ошибки сложной сети сильно изрезана и состоит из холмов, долин, складок и оврагов в пространстве высокой размерности. Сеть может попасть в локальный минимум (неглубокую долину), когда рядом имеется гораздо более глубокий минимум.

3. Временная неустойчивость. Обучение может оказаться бесполезным, если сеть находится в постоянно меняющейся внешней среде, когда второй раз один и тот же вектор может уже не повториться. В этом случае процесс обучения может оказаться бесконечно долгим и никогда не сойтись.

Рассмотрим некоторые способы борьбы с перечисленными проблемами.

1. Добавление смещения к активационной функции. Активационную функцию каждого нейрона можно наделить дополнительным параметром: настраиваемым смещением вдоль оси абсцисс. Это позволяет сдвигать начало отсчета и приводит к ускорению процесса обучения. Такая возможность может быть легко введена в алгоритм с помощью добавляемого к каждому нейрону веса, присоединенного к постоянному входному сигналу со значением 1. В процессе обучения этот вес настраивается по тем же правилам, что и все остальные веса связей, за исключением того, что подаваемый на него сигнал всегда равен 1, а не выходу нейрона предыдущего слоя.

2. Использование импульса. К формуле коррекции весов связей между нейронами добавляется слагаемое, которое пропорционально величине изменения веса на предыдущей итерации алгоритма. Как только происходит коррекция, она запоминается и служит для модификации всех последующих коррекций. Формулы (2.4) модифицируются следующим образом:

$$\begin{cases} \mathbf{D}_n = (\mathbf{W}_{n+1}\mathbf{D}_{n+1})\mathbf{O}(\mathbf{I}-\mathbf{O}_n); \\ \Delta\mathbf{W}_n(t+1) = \eta\mathbf{O}_{n-1}\mathbf{D}_n^T + \alpha\Delta\mathbf{W}_n(t); \\ n = N-1,\dots,1, \end{cases}$$

где α – коэффициент импульса (обычно используется значение порядка 0,9).

При использовании такой процедуры оптимизационный поиск стремится идти по дну узких оврагов поверхности ошибки (если таковые имеются), а не двигаться от склона к склону. Этот метод хорошо работает на некоторых задачах, однако дает слабый или даже отрицательный эффект на других.

2.2. Сеть встречного распространения

2.2.1. Структура сети встречного распространения

Сеть встречного распространения [10, 11] (рис. 2.7) состоит из двух слоев, имеющих разное назначение и разные алгоритмы обучения. Более того, обучения этих слоев выполняется в разных режимах: в режиме самообучения в случае слоя Кохонена и в режиме обучения с учителем в случае слоя Гросберга.

Нейроны слоя Кохонена имеют ступенчатую активационную функцию: для данного входного вектора **X** один и только один нейрон Кохонена, имеющий вектор весов связей, наиболее близкий ко входному вектору, объявляется «победителем» и выдает на выходе значение 1, а все остальные нейроны имеют выход, равный 0. Правило определения «победителя»:

$$i = \arg\min_i(\|\mathbf{X} - \mathbf{W}_i\|), \tag{2.5}$$

где $i \in \overline{1, K}$ – номер победившего нейрона слоя Кохонена; K – количество нейронов в слое Кохонена; $\mathbf{W}_i \in \mathbf{R}^N$ – вектор весов связей i-го нейрона слоя Кохонена.

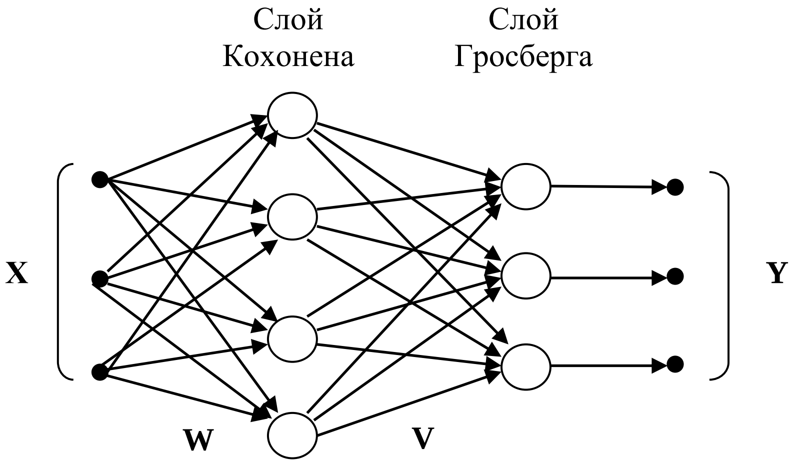


Рис. 2.7. Структура сети встречного распространения

Таким образом, выходы разных нейронов слоя Кохонена зависят друг от друга. Это обстоятельство можно формализовать, используя независимые нейроны и связи между нейронами внутри слоя. Однако такая формализация сильно усложнит логическую структуру сети, поэтому проще предполагать зависимость выходов, не зафиксированную в связях между нейронами. С учетом данного допущения сеть встречного распространения по-прежнему можно считать сетью без обратных связей.

Нейроны слоя Гросберга выполняют линейное преобразование

$$\mathbf{Y} = \mathbf{O}_2 = <\mathbf{V};\mathbf{O}_1>.$$

Поскольку для слоя Кохонена лишь у одного нейрона выход равен 1, а у остальных 0, то каждый нейрон слоя Гросберга лишь выдает величину веса, который связывает этот нейрон с единственным активным нейроном Кохонена.

Таким образом, общий принцип работы сети встречного распространения состоит в следующем.

1. Слой Кохонена классифицирует входные векторы на группы схожих. Это достигается с помощью такой настройки весов слоя Кохонена, что близкие входные векторы активируют один и тот же нейрон данного слоя. При этом не имеет значения, какой именно нейрон слоя Кохонена будет активироваться для каждой группы схожих входных векторов.

2. Задачей слоя Гросберга является получение выходов, которые наилучшим образом аппроксимируют «желаемые» выходные сигналы для каждого подмножества прецедентов, отнесенных слоем Кохонена к одному классу.

2.2.2. Обучение сети

Обучение слоя Кохонена

Обучение слоя Кохонена выполняется в режиме самообучения. Цель самообучения – обеспечить такие значения весов, чтобы близкие входные векторы активировали один и тот же нейрон Кохонена, а существенно отличающиеся входные векторы – разные нейроны Кохонена.

Процедура обучения определяется следующим алгоритмом.

1. Инициализировать матрицу весов **W**.
2. Подать очередной входной образ $\mathbf{X}_i$ и сравнить его с весовыми коэффициентами каждого нейрона для определения «победителя» согласно правилу (2.5).
3. Изменить веса для «победителя» по правилу:

$$\mathbf{W}_i(t+1) = \mathbf{W}_i(t) + \alpha(\mathbf{X} - \mathbf{W}_i(t)),$$

где $\mathbf{W}_i(t)$ – значения весов связей i-го нейрона слоя Кохонена на t-й итерации алгоритма; α– коэффициент скорости обучения.

4. Выбрать следующий обучающий пример и перейти к шагу (2).

Прецеденты из обучающей выборки подаются на вход сети циклически. Критерием останова для алгоритма служит условие: на очередной итерации все вектора обучающей выборки активируют те же нейроны, которые они активировали на предыдущей итерации.

Окончательные значения весовых векторов слоя Кохонена после обучения должны быть как можно ближе ко входными векторами. Поэтому процесс обучения можно ускорить, если предварительно нормализовать входные вектора и вектора весов всех нейронов.

При использовании нормализованных векторов процесс удобно представить геометрически – на поверхности сферы (рис. 2.8). Для очередного обучающего примера **X** для победившего нейрона i находится направление коррекции вектора $\mathbf{W}_i(t)$ – вектор $(\mathbf{X} - \mathbf{W}_i(t))$. Затем этот вектор масштабируется умножением его на скалярную величину α. Окончательно новый весовой вектор $\mathbf{W}_i(t+1)$ является отрезком, направленным из начала координат в конец вектора δ. Таким образом, эффект обучения состоит во вращении весового вектора в направлении входного вектора без существенного изменения его длины.

Переменная α является коэффициентом скорости обучения, который вначале обычно равен 0,7 и может постепенно уменьшаться в процессе обучения. Это позволяет делать большие начальные шаги для быстрого грубого обучения и меньшие шаги при подходе к окончательному состоянию сети.

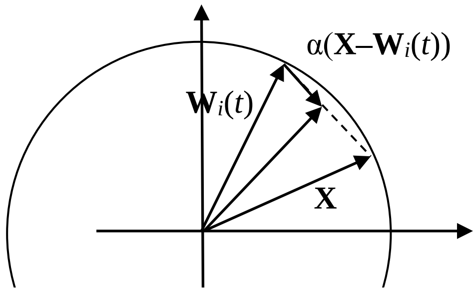


Рис. 2.8. Вращение нормализованных весовых векторов слоя Кохонена

Если бы с каждым нейроном Кохонена ассоциировался один входной вектор, то слой Кохонена мог бы быть обучен с помощью одного вычисления на вес: веса нейрона-победителя приравнивались бы к компонентам обучающего вектора при $\alpha = 1$. Как правило, обучающее множество включает много сходных между собой входных векторов, и сеть должна быть обучена активировать один и тот же нейрон Кохонена для каждого из них. В этом случае веса этого нейрона должны получаться усреднением входных векторов, которые должны его активировать. Постепенное уменьшение величины α уменьшает воздействие каждого обучающего шага, так что окончательное значение будет средней величиной от входных векторов, на которых происходит обучение. Таким образом, веса, ассоциированные с нейроном, примут значение вблизи центра подмножества входных векторов, для которых данный нейрон является «победителем».

Способы инициализации слоя Кохонена

Рандомизация весов слоя Кохонена на шаге (1) алгоритма обучения часто оказывается неэффективным вариантом инициализации. При таком способе инициализации весовые векторы распределяются равномерно по поверхности гиперсферы. В то же время входные векторы, как правило, распределены неравномерно и имеют тенденцию группироваться на относительно малой части поверхности гиперсферы. Поэтому большинство весовых векторов будут так удалены от любого входного вектора, что они никогда не будут «победителями». Поэтому:

– эти нейроны Кохонена будут всегда иметь нулевой выход и окажутся бесполезными;

– оставшихся активных нейронов может быть слишком мало, для того чтобы выделить необходимое количество классов данных.

Известные следующие варианты решения проблемы инициализации весов [10, 11].

1. Метод выпуклой комбинации. Все веса инициализируются одним и тем же значением $\mathbf{W}_{ij} = \frac{1}{\sqrt{N}}$, где N – количество входов сети. Для каждой компоненты входа $\mathbf{X}$ вычисляется скорректированное значение $\mathbf{X}_i' = \alpha \mathbf{X}_i + \frac{1-\alpha}{\sqrt{N}}$. Вначале α очень мало, вследствие чего все входные векторы имеют длину, близкую к $\frac{1}{\sqrt{N}}$, и почти совпадают с векторами весов. В процессе обучения сети α постепенно возрастает, приближаясь к единице. Т.е. обучающие вектора постепенно «разъезжаются» по поверхности сферы, увлекая за собой вектора весов. Весовые векторы отслеживают один или небольшую группу входных векторов и в конце обучения дают требуемую картину выходов.

2. Равномерное распределение весовых векторов и добавление шума к входным векторам. Тем самым входные вектора подвергаются случайным изменениям, «схватывая», в конце концов, какой-нибудь весовой вектор.

3. Веса инициализируются случайными значениями, но на начальной стадии обучения изменению подвергаются веса всех нейронов, а не только нейрона, признанного «победителем». Тем самым весовые векторы перемещаются ближе к области входных векторов. В процессе обучения коррекция весов начинает производиться лишь для ближайших к победителю нейронов Кохонена. Этот радиус коррекции постепенно уменьшается, так что в конце концов корректируются только веса, связанные с выигравшим нейроном Кохонена.

4. Наделение нейрона «чувством справедливости». Если какой-либо нейрон становится победителем чаще средней доли случаев $\frac{1}{K}$, то он временно исключается из рассмотрения при выборе «победителя», тем самым давая возможность обучаться и другим нейронам.

Обучение слоя Гросберга

Слой Гросберга обучается в режиме обучения с учителем по ранее рассмотренному методу линейной коррекции. Величина коррекции веса пропорциональна разности между текущим весом и требуемым выходом нейрона Гросберга, к которому он относится:

$$\mathbf{V}_{ij}(t+1) = \mathbf{V}_{ij}(t) + \beta(\mathbf{Y}_j - \mathbf{V}_{ij}(t))\mathbf{O}_i,$$

где $\mathbf{O}_i$ – выход *i*-го нейрона Кохонена; $\mathbf{Y}_j$ – *j*-я компонента вектора желаемых выходов. Первоначально β берется равным ~0,1 и затем постепенно уменьшается в процессе обучения.

Отсюда видно, что при обучении слоя Гросберга каждый вес корректируется лишь в том случае, если он соединен с нейроном Кохонена, имеющим ненулевой выход. Поэтому веса слоя Гросберга будут сходиться к средним величинам от желаемых выходов, как в однослойном перцептроне, но только независимо для каждого класса предъявляемых образов.

В целом, если сравнить алгоритм обучения сети встречного распространения с алгоритмом обратного распространения ошибки, то можно выделить следующие ключевые отличия:

– более высокая скорость обучения; в реальных задачах количество итераций, необходимых для обучения сети может отличаться на несколько порядков;

– более низкая точность аппроксимации восстанавливаемой функциональной зависимости, поскольку для объектов каждого класса, выделенного слоем Кохонена, выходной сигнал сети будет одинаковым.

2.2.3. Использование сети для восстановления обратной функции

На рис. 2.9 показана сеть встречного распространения, в которой интерпретация входов и выходов сети отличается от использованной ранее. На вход сети поступают векторы **X** и **Y**, и обученная сеть выдает на выходе векторы **X**’ и **Y**’, являющиеся аппроксимациями для **X** и **Y**, соответственно. Входные и желаемые выходные векторы предполагаются здесь нормализованными единичными векторами. Следовательно, порождаемые на выходе векторы также будут иметь тенденцию быть нормализованными.

В процессе обучения векторы **X** и **Y** подаются одновременно и как входные векторы сети, и как желаемые выходные сигналы. Вектор **X** используется для обучения выходов **X**’, а вектор **Y** – для обучения выходов **Y**’ слоя Гросберга. Сеть встречного распространения целиком обучается с использованием описанного выше алгоритма. Нейроны Кохонена принимают входные сигналы как от векторов **X**, так и от векторов **Y**, но с точки зрения алгоритма обучения сети это неотличимо от ситуации, когда имеется один вектор, составленный из векторов **X** и **Y**, и никак не влияет на алгоритм обучения.

Настроенная сеть аппроксимирует единичное отображение, при котором предъявление пары входных векторов порождает их копии на выходе. Если теперь предъявлять на входе только вектор **X** (с вектором **Y**, равным нулю) то сеть обеспечит выходной сигнал как на выходах **X’**, так и на выходах **Y’.** Если F – функция, отображающая **X** в **Y’**, то сеть аппроксимирует ее. Если F обратима, то предъявление только вектора **Y** (при **X**=0) порождает **X’**.

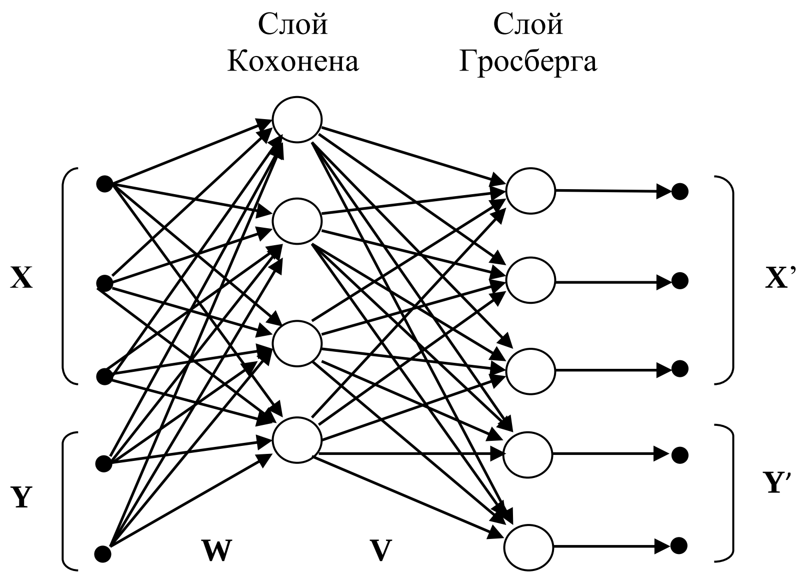


Рис. 2.9. Полная сеть встречного распространения

Способность одновременно аппроксимировать как прямую, так и обратную функцию делает сеть встречного распространения полезной в ряде приложений. Допустим, что требуется передать по каналу связи с ограниченной пропускной способностью некоторое изображение. Оно может быть разбито на подизображения S_{ij}, как показано на рис. 2.10. Каждое подизображение разбито на пиксели и описывается вектором яркостей. Если при передаче допустимы некоторые искажения, то можно передавать не все подизображения, а несколько «типовых подизображений» и коды типовых подизображений, из которых приближенно состоит изображение.

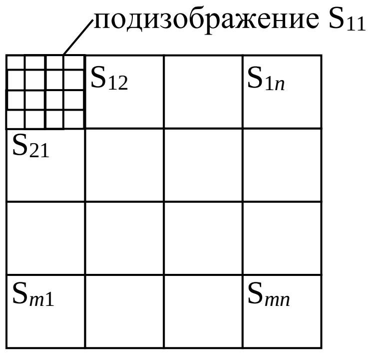


Рис. 2.10. Структура сжимаемого изображения

Множество векторов подизображений используется в качестве входа для обучения слоя Кохонена и дает выход только одного нейрона равным 1. Веса слоя Гросберга обучают выдавать бинарный код номера того нейрона Кохонена, выход которого равен 1. Это и будет являться более короткой битовой последовательностью передаваемых символов.

На приемный конец канала связи одноразово передают веса настроенной сети встречного распространения. При передаче очередного изображения оно кодируется сетью на стороне источника. По каналу связи передаются только коды всех подизображений. Идентичная сеть на стороне приемника получает двоичный код и реализует обратную функцию, аппроксимируя первоначальные подизображения.

2.3. Сеть радиальных базисных функций

Сеть радиальных базисных функций (часто используется термин «RBF-сеть») использует идею описания классифицируемых образов с помощью поля потенциалов [9]. Активационная функция нейронов – не пороговая, а подобна функции плотности вероятности нормального распределения: нейрон дает высокий отклик в некоторой окрестности своего входного вектора. Суммируя выходы большого числа нейронов такого вида можно аппроксимировать сколь угодно сложные функциональные зависимости (рис. 2.11).

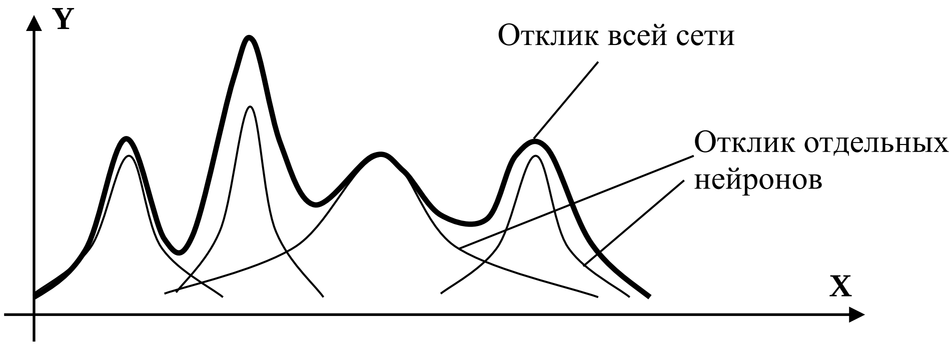


Рис. 2.11. Аппроксимация сложной функциональной зависимости суммой радиальных базисных функций

Структура сети радиальных базисных функций представлена на рис. 2.12. Она аналогична структуре двуслойного перцептрона с тем отличием, что активационные функции нейронов первого слоя имеют вид, подобный функции плотности вероятности нормального распределения. Количество нейронов впервом слое совпадает с количеством объектов в обучающей выборке, количество нейронов второго слоя определяется размерностью выходного сигнала сети. Как будет показано ниже, в ходе обучения с учителем сеть настраивается независимо по каждой компоненте выходного сигнала, поэтому на рисунке все нейроны выходного слоя, кроме одного, показаны пунктирными линиями.

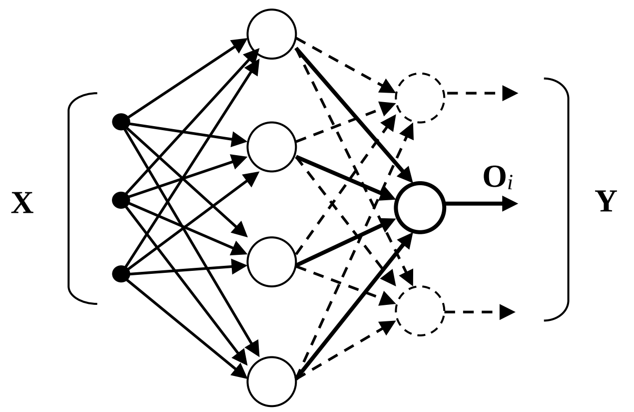


Рис. 2.12. Структура сети радиальных базисных функций

Каждый нейрон первого слоя рассчитывает значение одномерной функции:

$$\mathrm{h}_i(\mathbf{X}) = \exp\left(-\frac{(\|\mathbf{X}-\mathbf{X}_i\|)^2}{2\sigma_i^2}\right),$$

где $\mathbf{X}$ – входной вектор; $\mathbf{X}_i$ – i-й образ из обучающей выборки; σ_i – параметр рассеяния; $\|\mathbf{X}-\mathbf{X}_i\|$ – некоторая метрика (обычно используется евклидово расстояние).

Для выбора σ обычно применяется метод k ближайших соседей. Сначала в составе обучающей выборки определяется k ближайших соседей для объекта $\mathbf{X}_i$. Выбирается вектор $\mathbf{X}_i'$, занимающий k-е место по удаленности от вектора $\mathbf{X}_i$. Расстояние между $\mathbf{X}_i$ и $\mathbf{X}_i'$ служит мерой для выбора значения параметра σ_i.

Выходной слой является слоем Гросберга, т.е. реализует преобразование вида:

$$\mathrm{f}(\mathbf{X}) = \sum_{i=1}^{N} \mathbf{C}_i \,\mathrm{h}_i(\mathbf{X}).$$

Относительно каждой компоненты выходного сигнала это преобразование независимо, поэтому каждый весовой коэффициент можно найти как решение системы линейных уравнений:

$$\sum_{i=1}^{N} \mathbf{C}_i \,\mathrm{h}_i(\mathbf{X}_j) = \mathbf{Y}_j, \quad j = \overline{1, N}. \tag{2.6}$$

Введем в рассмотрение матрицу:

$$\mathbf{H} = \begin{bmatrix} \mathrm{h}_1(\mathbf{X}_1) & \cdots & \mathrm{h}_N(\mathbf{X}_1) \\ \cdots & \cdots & \cdots \\ \mathrm{h}_1(\mathbf{X}_N) & \cdots & \mathrm{h}_N(\mathbf{X}_N) \end{bmatrix}.$$

Тогда проблему интерполяции (2.6) можно представить следующим образом:

$$\mathbf{HC} = \mathbf{Y}. \tag{2.7}$$

Решение этой задачи определяется обратной матрицей для **H**:

$$\mathbf{C} = \mathbf{H}^{-1}\mathbf{Y}.$$

Если входные данные искажены (зашумлены), то точная интерполяция не имеет смысла. Следовательно, описанный выше метод следует модифицировать: проблема состоит в таком решении системы линейных уравнений (2.7), которая является робастной относительно шумов (помех) во входных данных. Поэтому матрица **H** заменяется матрицей (**H**+λ**I**), где **I** – единичная матрица, а λ – небольшое положительное число. Его величина пропорциональна силе шума во входных данных. Весовые коэффициенты второго слоя сети определяются при этом следующим образом:

$$\mathbf{C} = (\mathbf{H} + \lambda\mathbf{I})^{-1}\mathbf{Y}.$$

Отметим преимущества сети радиальных базисных функций по сравнению с другими видами сетей.

1. Активационная функция нейронов первого слоя принимает большие значения лишь в тех случаях, когда входной образ находится вблизи от опорной точки соответствующего этого нейрона. Для входных сигналов вне области, «покрытой» образами обучающей последовательности, сеть формирует лишь небольшие значения на своих выходах. Таким образом, можно различить ситуации, когда обучающей информации недостаточно для определения выхода сети по некоторому входному сигналу.
2. Веса связей в сети рассчитываются не итерационно, поэтому вычислительная сложность настройки сети существенно ниже, чем для других видов сетей прямого распространения.

Заслуживает также внимания возможность инициализации сети радиальных базисных функций на основе прямого расчета весов с последующим их дообучением на основе алгоритмов обучения с учителем, например, на основе алгоритма обратного распространения ошибки.

Можно рассматривать вариант, когда количество нейронов скрытого слоя меньше количества обучающих образов. В этом случае первый слой сети необходим предварительно обучить методом обучения без учителя, аналогично алгоритму обучения слоя Кохонена.

2.4. Сети с ограниченными областями связи

Рассмотренные до сих пор сети имеют принципиальный общий недостаток: если какой-либо нейрон получает слишком сильное подкрепление, то он может заблокировать работу всей сети, бесконечно усиливая собственную активность. Такая ситуация может возникнуть для сетей прямого распространения по ходу настройки весов, а для сетей с обратными связями – при вычислении выходного сигнала. Эту проблему можно решить с помощью различных искусственных ограничений, однако существует и другой способ: локализовать часть сети, на которую может влиять каждый нейрон. В этом случае сколь угодно высокая активность отдельного нейрона не сказывается на поведении нейронов, находящихся за пределами его области влияния. Такая идея реализуется в рамках сверточных нейронных сетей, в частности в когнитроне и неокогнитроне [10].

2.4.1. Структура связей отдельного нейрона в когнитроне

Сеть имеет слоистую структуру, но каждый нейрон связан не со всеми нейронами предыдущего слоя, а только с некоторыми соседними (в своей «области связи»). Это соответствует принципам работы биологической нейронной сети, где прямые связи между значительно удаленными друг от друга нейронами невозможны.

Области связи соседних нейронов одного слоя в пресинаптическом слое значительно перекрываются, поэтому в постсинаптическом слое у каждого нейрона есть область конкуренции – множество соседних нейронов, которые имеют общие с данным пресинаптические нейроны (рис. 2.13).

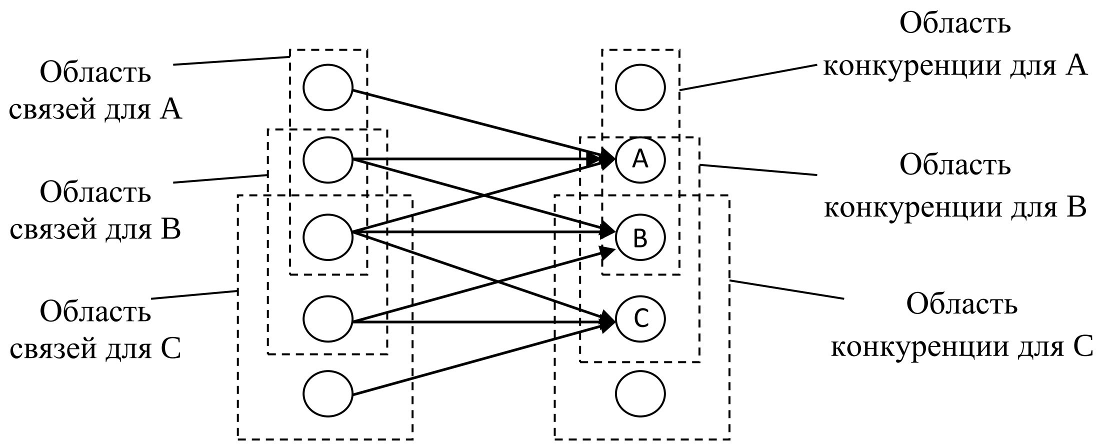


Рис. 2.13. Области связи и конкуренции для пары соседних слоев когнитрона

Область связи и область конкуренции для каждого нейрона определяют структуру его связей (рис. 2.14).

В когнитроне используются два типа нейронов:

– возбуждающие – они усиливают активность постсинаптических нейронов (на рисунке обозначены светлой закраской);

– тормозящие – они снижают активность постсинаптических нейронов (на рисунке обозначены темной закраской).

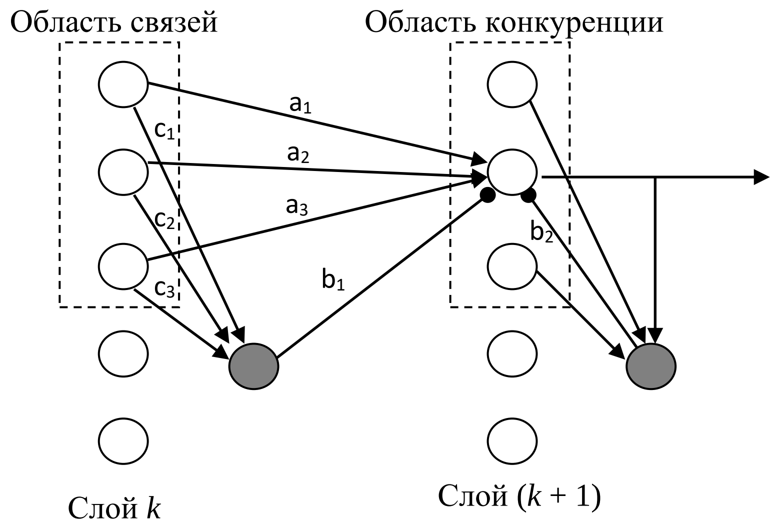


Рис. 2.14. Структура связей отдельного нейрона

Каждый возбуждающий нейрон слоя (k+1) получает на входе:

– сигналы от возбуждающих нейронов слоя k из своей области связи;

– сигнал от тормозящего нейрона слоя k, соединенного с теми же возбуждающими нейронами слоя k;

– сигнал от тормозящего нейрона слоя (k+1), соединенного с возбуждающими нейронами слоя (k+1) из своей области конкуренции.

Входы возбуждающего нейрона, соединенные с возбуждающими и тормозящими нейронами, суммируются раздельно:

$$E = \sum_{\mathrm{i}} a_{\mathrm{i}} u_{\mathrm{i}}, \quad I = \sum_{\mathrm{j}} b_{\mathrm{j}} v_{\mathrm{j}},$$

где a_i – вес i-го возбуждающего синапса, u_i – выход i-го возбуждающего нейрона, b_j – вес j-го тормозящего синапса, v_j – выход j-го тормозящего нейрона. При этом веса имеют только положительные значения.

Выход нейрона вычисляется следующим образом:

$$S = \frac{1+E}{1+I} - 1;$$

$$O = \begin{cases} S, \text{если } S \geq 0; \\ S, \text{если } S < 0. \end{cases}$$

Тормозящий нейрон не имеет пороговой функции. Его выход:

$$b_j = \sum_i c_i x_i,$$

где $\sum_i c_i = 1$.

2.4.2. Обучение когнитрона

Сеть обучается без учителя. Принцип обучения когнитрона: настраиваются веса только наиболее активного нейрона из области конкуренции, поэтому на последующих шагах он будет иметь еще более высокую вероятность победы.

Веса связей тормозящих нейронов при обучении не меняются, достаточно выбрать их такими, чтобы их сумма равнялась 1. Веса связей возбуждающего нейрона с возбуждающими пресинаптическими нейронами в ходе обучения изменяются по правилу:

$$\Delta a_{\mathrm{i}} = q c_i u_i ,$$

где c_i – тормозящий вес связи нейрона i в слое k с тормозящим нейроном, u_i – выход нейрона i в пресинаптическом слое, a_i – возбуждающий вес этого нейрона, q – нормирующий коэффициент обучения.

Веса связей возбуждающего нейрона с тормозящими пресинаптическими нейронами изменяются по правилу:

$$\Delta b_j = (q/2)\sum_i a_i u_i \Big/ \sum_i c_i u_i .$$

Процесс обучения начинается с нулевыми значениями весов. Пока победителя в области конкуренции нейронов нет, изменение весов связей происходит следующим образом:

$$\Delta a_i = q' c_i u_i ; \Delta b_j = q' \sum_i c_i u_i ; 0 < q' < q .$$

Описанные правила позволяют весам связей возрастать без ограничений. Приведенная стратегия настройки гарантирует, что узлы с большой реакцией заставляют веса возбуждающих связей увеличиваться сильнее, чем веса тормозящих связей. И наоборот, узлы, имеющие малую реакцию, вызывают малое возрастание весов возбуждающих связей, но большее возрастание весов тормозящих связей.

Можно считать, что в каждой области конкуренции сеть ведет себя как сеть Кохонена, выдавая на выходе номер распознанного образа. Но поскольку область конкуренции каждого нейрона локальна – в итоге активными могут остаться несколько нейронов слоя, расположенных в непересекающихся областях конкуренции.

2.4.3. Полная структура когнитрона

До сих пор речь шла об упрощенной модели, где каждый слой является одномерным (с точки зрения топологии области связей) и слоев всего 2. Полная структура когнитрона предполагает следующее (рис. 2.15).

1. Топология областей связи двумерная – т.е. каждый нейрон связан с нейронами предыдущего слоя из окрестности, имеющей две координаты.

2. Количество слоев может быть достаточно большим. Выход каждого слоя представляет собой классификацию входных сигналов. Результаты классификации по всему слою передаются на следующий слой как исходные данные и т.д.

Таким образом, в каждом слое происходит категоризация входных данных, полученных с предыдущего слоя. Результат категоризации – это более короткий код, в который свернуты входные данные. Далее он используется как входные данные для следующего слоя, который выполняет последующее его свертывание и т.д. – этим объясняется происхождение термина «сверточная нейронная сеть». Можно сказать, что с каждым слоем категоризация происходит на все более высоком уровне обобщения информации. Это весьма близко к принципу работы коры головного мозга.

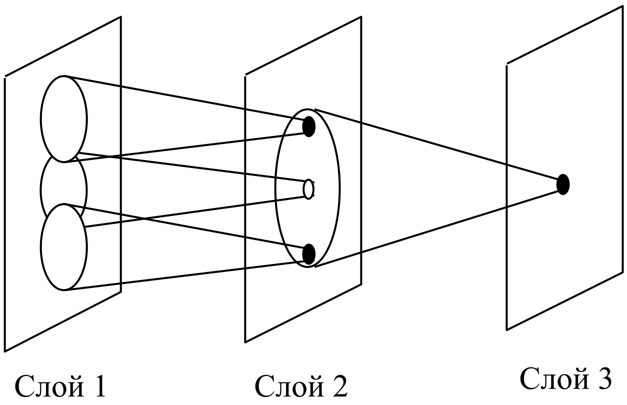


Рис. 2.15. Последовательная категоризация данных в когнитроне

Если область связи нейронов имеет постоянный размер во всех слоях, то потребуется большое количество слоев для перекрытия всего входного поля выходными нейронами. Эту проблему можно решить за счет вероятностного распределения области связей каждого нейрона: он будет связан с большинством нейронов из ближайшего окружения и с некоторым количеством нейронов из удаленный областей предыдущего слоя.

2.4.4. Неокогнитрон

Когнитрон изначально был разработан для решения задачи анализа изображений. Если считать, что вход первого слоя когнитрона – это не просто вектор, а растровое изображение, то можно решить более сложную задачу распознавания. Нейронная сеть должна распознавать не только зашумленный образ, но и образ, «растровое изображение» которого подверглось геометрическим деформациям: повороту, сдвигу, сжатию. В общем случае, когда исходные данные не являются изображением, это означает, что исходные данные могут смещаться из одних компонент входного вектора в другие некоторым систематическим образом. Эффект такого рода может возникнуть, например, как следствие корреляции между различными компонентами входного сигнала.

Решение указанной задачи обеспечивает неокогнитрон. Структура неокогнитрона такова: сеть состоит из последовательных слоев, каждый слой состоит из двух массивов плоскостей, каждая плоскость состоит из нейронов (рис. 2.16). Массив плоскостей, содержащих простые узлы, получает выходы предыдущего слоя, выделяет определенные образы и затем передает их в массив плоскостей, содержащих комплексные узлы, где они обрабатываются таким образом, чтобы сделать выделенные образы менее позиционно зависимыми.

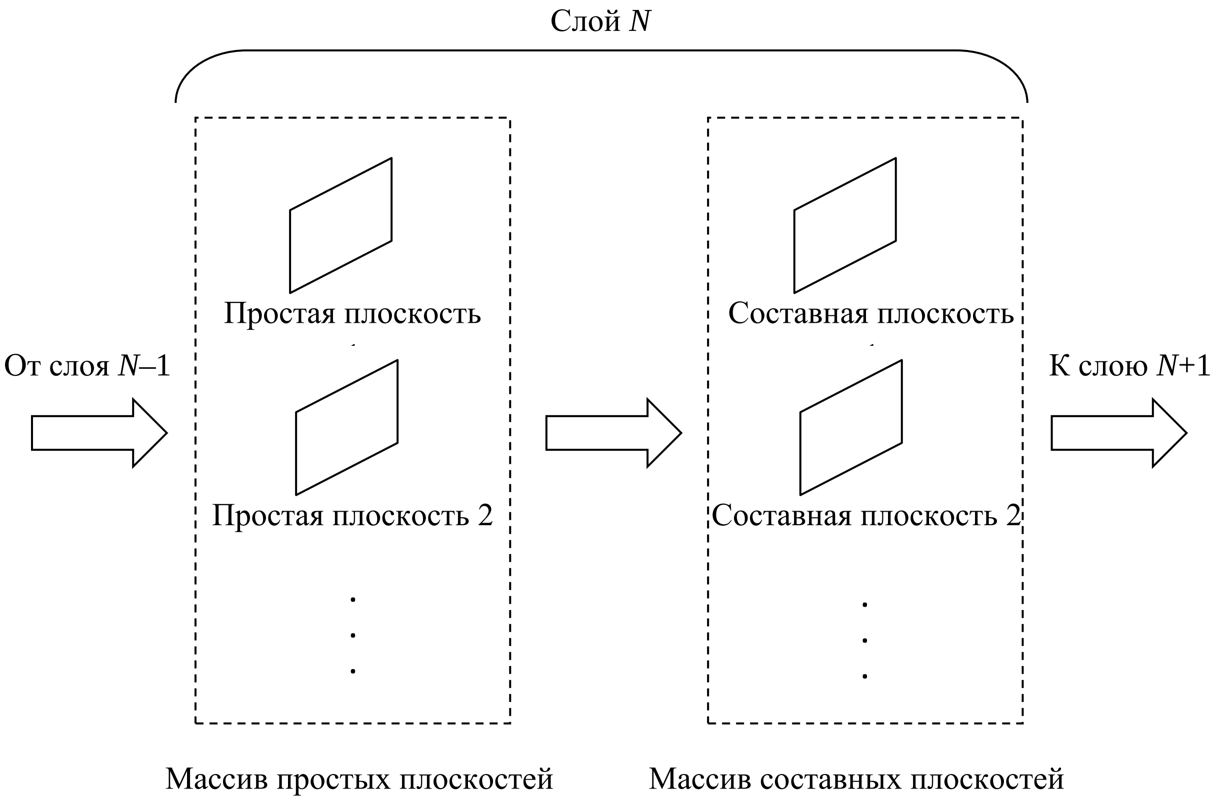


Рис. 2.16. Структура одного слоя неокогнитрона

Каждая плоскость может быть визуально представлена как двумерный массив узлов. Каждая плоскость из массива простых узлов соединена (рис. 2.17):

– пресинаптическими связями – со всеми плоскостями массива комплексных узлов предыдущего слоя;

– постсинаптическими связями – с одной плоскостью массива комплексных узлов своего слоя.

На каждый нейрон поступают сигналы из некоторой области связи, одной и той же для всех комплексных плоскостей предыдущего слоя. Таким образом, все плоскости простых узлов получают на входе один и тот же массив сигналов, но умножают его на свои веса связей.

Каждый комплексный узел получает в качестве входного образа выходы набора простых узлов из своей области связи. Возбуждение любого простого узла в области связи является достаточным для возбуждения данного комплексного узла. Таким образом, комплексный узел реагирует на тот же образ, что и простые узлы в соответствующей ему плоскости, но он менее чувствителен к позиции образа, чем любой из них по отдельности.

Нейроны каждой пары плоскостей обучаются реагировать на определенный образ, представленный в определенной ориентации. Для другого образа или для нового угла поворота образа требуется новая пара плоскостей. При больших объемах информации, неокогнитрон представляет собой огромную структуру с большим количеством плоскостей и слоев нейронов.

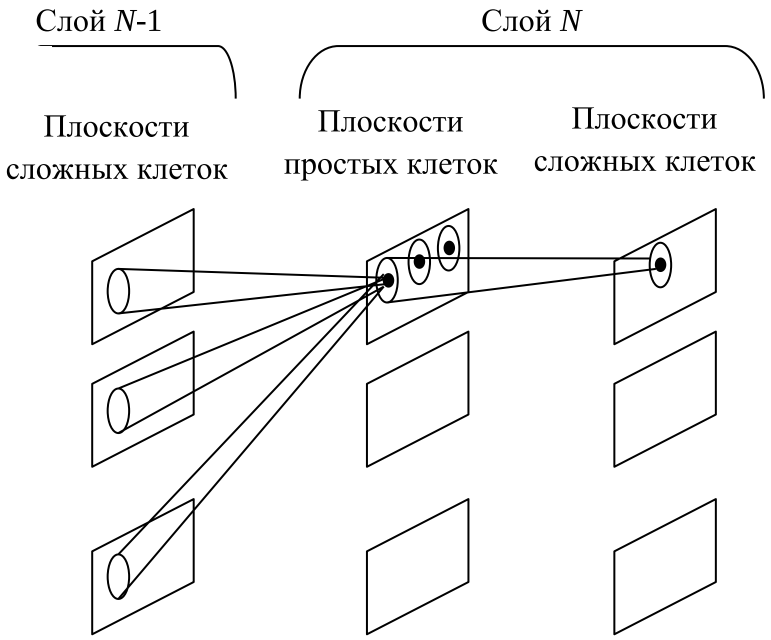


Рис. 2.17. Области связи для одного слоя неокогнитрона

По мере распространения информации от слоя слою сеть становится все менее чувствительной к ориентации и расположению образа, и, в определенных пределах, к его размеру. Нейроны выходного слоя выполняют окончательное инвариантное распознавание.

Для уменьшения объема обрабатываемой информации области связи при переходе со слоя на слой расширяются, а число нейронов уменьшается. В выходном слое на каждой плоскости остается только один нейрон, связанный со всей поверхностью плоскости предыдущего слоя.

Можно считать, что неокогнитрон – это модель слоистой нейронной сети второго уровня: вместо отдельного нейрона многослойного перцептрона здесь используется пара «плоскость простых узлов – плоскость комплексных узлов», которая работает как «метанейрон», т.е.:

– плоскость простых узлов связана с выходами всех «метанейронов» предыдущего слоя и вычисляет взвешенную сумму их сигналов;

– плоскость комплексных узлов, по аналогии с активационной функцией, обеспечивает нелинейное преобразование полученной суммы и определяет выход «метанейрона».

Контрольные вопросы к главе 2

1. Какие дополнительные требования предъявляются к структуре многослойного перцептрона по сравнению с многослойной слоистой сетью общего вида?
2. Как зависит обобщающая способность многослойного перцептрона от количества его слоев и количество нейронов в этих слоях?
3. С какой целью в алгоритме обратного распространения ошибки требуется дифференцируемость активационной функции?
4. Как ограничения процедуры градиентного спуска проявляют себя при обучении многослойного перцептрона методом обратного распространения ошибки?
5. Объясните происхождение термина «сеть встречного распространения».
6. Может ли слой Кохонена в сети встречного распространения включать количество нейронов, не совпадающее с количеством классов в решаемой этой сетью задаче классификации? Что произойдет, если количество нейронов будет меньше количества классов, больше количества классов?
7. Чем вызваны сложности при задании начальных весов нейронов Кохонена?
8. Как соотносятся характеристики многослойного перцептрона и сети встречного распространения с точки зрения скорости и качества обучения?
9. Каким образом определяется количество нейронов в сети радиальных базисных функций?
10. С какой целью в ИНС применяется процедура ограничения областей связей нейронов?
11. Объясните происхождение термина «сверточная нейронная сеть».
12. В чем состоит аналогия между структурными элементами неокогнитрона и многослойного перцептрона?

Глава 3. СЕТИ С ОБРАТНЫМИ СВЯЗЯМИ

Сеть с обратными связями – это сеть, в которой некоторые сигналы могут распространяться по циклическому пути. При этом так же, как и для слоистой сети, на некоторые нейроны могут поступать входные сигналы, а с некоторых нейронов – сниматься выходные сигналы. Наличие обратных связей означает, что при заданном входном сигнале **X** выходной сигнал **Y** меняется во времени:

$\mathbf{Y}(1) = \mathrm{F}(\mathbf{X})$;

$\mathbf{Y}(2) = \mathrm{F}(\mathbf{X},\mathbf{Y}(1))$;

$\mathbf{Y}(3) = \mathrm{F}(\mathbf{X},\mathbf{Y}(2))$;

и т.д.

Интерес представляют, прежде всего, сети, в которых выходной сигнал **Y** сходится к некоторым постоянным значениям.

Основное применение сетей с обратными связями – это реализация ассоциативной памяти. Память можно рассматривать как сложное функциональное преобразование. При этом она может быть:

а) жестко адресуемой – когда по заданному точному входу (адресу) определяется точный выход (содержимое памяти);

б) ассоциативной – когда по заданному приближенному входу определяется ассоциированный с ним выход; используется также термин «память, адресуемая по содержимому».

В зависимости от вида сети эта постановка задачи уточняется.

3.1. Сеть Хопфилда

3.1.1. Структура сети Хопфилда

С формальной точки зрения сеть Хопфилда – это просто полносвязная сеть, в которой все связи являются двунаправленными (рис. 3.1).

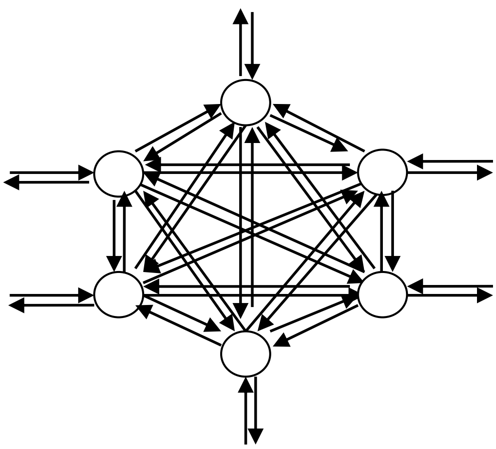


Рис. 3.1. Сеть Хопфилда как полносвязная сеть

Однако удобнее дать другую графическую интерпретацию сети (рис. 3.2). Входной сигнал сети подается на вход всех нейронов только на первой итерации. Вычисленный на очередной итерации выходной сигнал каждого нейрона на следующей итерации опять передается на входы всех нейронов.

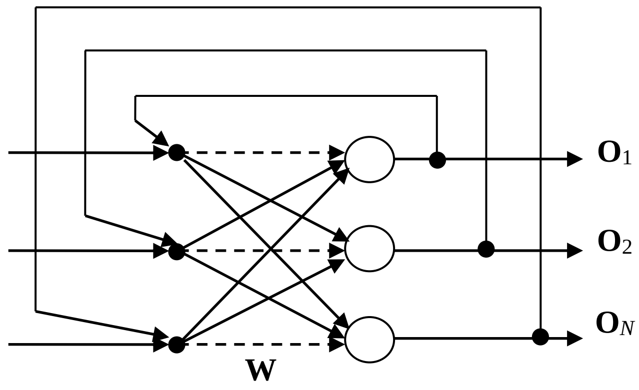


Рис. 3.2. Альтернативное представление сети Хопфилда

Активационная функция для всех нейронов предполагается ступенчатой:

$$\mathbf{O}_i = \begin{cases} 1, \text{ если } \sum_{j=1}^{N} \mathbf{W}_{ij}\mathbf{O}_j > 0; \\ -1, \text{ иначе.} \end{cases}$$

Текущее состояние сети можно интерпретировать как одну из вершин N-мерного гиперкуба. Тогда входной сигнал сети определяет стартовую вершину этого гиперкуба, а в ходе циклического распространения сигнала по сети она перемещается между вершинами гиперкуба. Если сеть устойчива, то она окончательно стабилизируется в одной из вершин. Конечная вершина определяется начальным входным сигналом и весами связей сети. Доказано, что для устойчивости сети достаточно, чтобы матрица **W** была симметрична и имела нули на главной диагонали (на рис. 3.2 соответствующие связи указаны пунктиром). Это условие достаточно, но избыточно. На практике для обеспечения устойчивости сети достаточно приближенной симметрии.

Задача сети – реализовать автоассоциативную память, т.е. для неточного входного вектора, поданного в сеть, перейти в стабильное состояние, соответствующее точному входному вектору.

3.1.2. Обучение сети

Согласно постановке задачи, для правильных выходов **O** каждого запоминаемого вектора: $\mathbf{O}(t+1) = \mathrm{F}(\mathbf{WO}(t))$, т.е состояние является устойчивым. Для M запоминаемых векторов имеем систему уравнений.

$$\mathbf{Y}_m(t+1) = \mathrm{F}(\mathbf{WY}_m(t)), \quad m = \overline{1, M}\ .$$

Отсюда значения всех весов определяются выражением:

$$\mathbf{W}_{ij} = \frac{1}{N}\sum_{m=1}^{M}\mathbf{Y}_{mi}\mathbf{Y}_{mj}\ .$$

Таким образом, обучение сети Хопфилда является неитерационным: множество запоминаемых векторов однозначным образом определяет все веса связей между нейронами.

В предельном случае, при одном запомненном образе, сеть Хопфилда решает задачу фильтрации точного образа от шумов. При нескольких запоминаемых образах сеть дополнительно определяет соответствие предъявленного зашумленного образа одному из сохраненных образов. Допустимое количество запоминаемых образов оценивается отношением $M \le 0{,}15N$, при большем количестве образов сеть перестает их корректно восстанавливать.

Технически работу любой сети с обратными связями можно реализовать:

а) в синхронном режиме: очередное «поколение» выходов вычисляется целиком, и только затем обновляется сигнал, передаваемый по обратной связи;

б) в асинхронном режиме: после вычисление нового значения каждого выхода его сигнал по обратным связям обновляется.

В случае (а) сеть Хопфилда может попасть в состояние динамического равновесия и будет бесконечно циклически переключаться между несколькими состояниями. В случае (б) сеть Хопфилда гарантированно сойдется к одному из статических состояний.

3.1.3. Использование сети для решения оптимизационной задачи: пример

Доказано, что в ходе функционирования сеть Хопфилда на каждом шаге не увеличивает значение энергии:

$$E = -\frac{1}{2}\sum_{i,j}\mathbf{W}_{ij}\mathbf{O}_i\mathbf{O}_j\ .$$

Таким образом, сеть Хопфилда также обеспечивает оптимизационный поиск, но не на этапе настройки весов, а на этапе приближения зашумленного сигнала к одному из устойчивых состояний. При этом функция энергии может иметь локальные минимумы как в точках запоминаемых образов, так и в других точках. По ходу оптимизации можно попасть в такую точку «ложного образа» – отсюда и возникает ограничение на количество запоминаемых образов.

Сеть Хопфилда можно обобщить с помощью гладкой активационной функции:

$$\mathrm{F}(z) = \frac{2}{1 + e^{-az}} - 1.$$

При большом значении a функция сходится к ступенчатой; при уменьшении a энергетические минимумы сети постепенно удаляются от вершин гиперкуба в его внутреннюю часть.

Рассмотрим пример, демонстрирующий использование сети для решения оптимизационной задачи. Будем решать стандартную задача коммивояжера: обойти все N вершин заданного графа по пути с минимальной длинной, однократно посещая каждую вершину.

Для представления решения такой задачи в сети Хопфилда воспользуемся следующей системой кодирования:

– сеть включает $N \times N$ нейронов (будем условно считать их квадратной матрицей);

– для установившегося выхода сети в каждой строке и в каждом столбце матрицы должен быть активен только один нейрон;

– активный нейрон в каждом столбце определяет очередную вершину в порядке обхода графа.

Для выполнения ограничений, заданных условиями задачи, сеть должна обладать следующими свойствами:

– каждый нейрон препятствует активации других нейронов своей строки (условие A) и своего столбца (условие B) – это требование обеспечивает корректность сигнала в сети с точки зрения выбранного способа кодирования пути в графе;

– каждый нейрон в i-м столбце тем активнее препятствует активации нейронов в (i–1) и (i+1) столбцах, чем больше расстояние до соответствующих вершин графа (условие C).

В символическом виде перечисленные требования задаются следующим выражением:

$$\mathbf{W}_{xi,yj} = -AS_{xy}(1 - S_{ij}) - BS_{ij}(1 - S_{xy}) - C\,\mathrm{r}(x, y)(S_{i,j+1} + S_{i,j-1}) + D\,,$$

где $\mathbf{W}_{xi,yj}$ – вес связи между x-м нейроном в i-м столбце и y-м нейроном в j-м столбце; A, B, C, D – константы; $Sij = \begin{cases} 1, \text{ если } i = j; \\ 0, \text{ иначе;} \end{cases}$ $\mathrm{r}(x, y)$ – расстояние между парой вершин графа.

Из произвольного начального состояния выходной сигнал сети должен сойтись к локальному минимуму функции энергии, тем самым минимизируя суммарную длину пути, заданного матрицей $\mathbf{W}_{xi,yj}$.

3.2. Сеть Хэмминга

Во многих прикладных задачах нет необходимости, чтобы сеть в явном виде выдавала образец: достаточно получить на выходе сети номер распознанного образца. При такой постановке задачи автоассоциативную память успешно реализует сеть Хэмминга. По сравнению с сетью Хопфилда, она требует меньших затрат памяти и объема вычислений.

Сеть состоит из двух слоев (рис. 3.3). Первый и второй слои имеют по M нейронов, где M – число хранимых образов. N-мерный входной сигнал передается на все нейроны первого слоя по связям, веса которых заданны матрицей **W**. Все нейроны второго слоя связаны между собой ингибиторными (отрицательными обратными) связями с весами, равными $-\varepsilon$. Обратная связь нейрона с самим собой имеет положительный вес, равный +1.

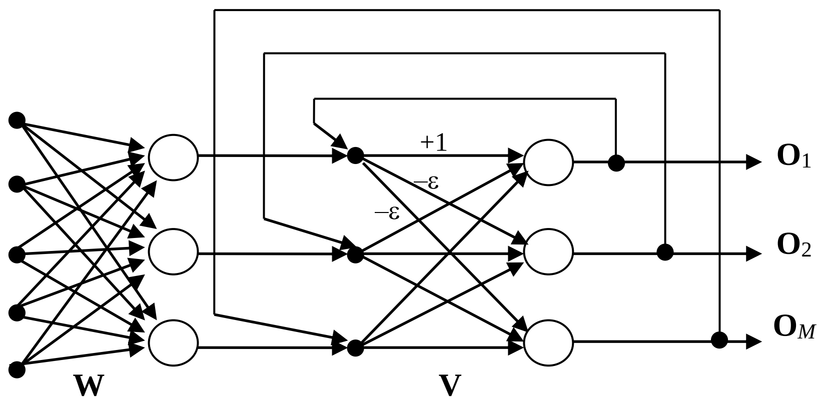


Рис. 3.3. Структурная схема сети Хэмминга

Принцип работы сети состоит в следующем: входной вектор преобразуется 1-м слоем в бинарный код, а затем 2-й слой активирует только один нейрон – тот, который будет соответствовать этому коду. При предъявлении 2-му слою искаженного кода входного вектора изначально активны будут несколько нейронов, а затем за счет обратных связей 2-й слой сойдется к одному из устойчивых состояний – в котором активен только один нейрон. Он и будет определять номер запомненного образа, которому соответствует предъявленный сети зашумленный вход.

Активационная функция F имеет вид порога. На стадии обучения весовым коэффициентам первого слоя и порогу активационной функции присваиваются следующие значения:

$$\mathbf{W}_{ik} = \frac{\mathbf{X}_i^k}{2}, \quad i = \overline{1, N}, \quad k = \overline{1, M},$$

$$T_k = \frac{N}{2}, \quad k = \overline{1, M},$$

где $\mathbf{X}_i^k$ – i-й элемент k-го образца.

Алгоритм функционирования обученной сети Хэмминга имеет следующий вид.

1. Подать на вход сети распознаваемый вектор $\mathbf{X}$ и рассчитать состояния нейронов первого слоя (верхний индекс в скобках указывает номер слоя):

$$\mathbf{O}_j^{(1)} = \sum_{i=1}^{N} \mathbf{W}_{ij} \mathbf{X}_i + T_j, \quad j = \overline{1, M}.$$

2. Инициализировать выходы второго слоя:

$$\mathbf{O}_j^{(2)} = \mathbf{O}_j^{(1)}, \quad j = \overline{1, M}.$$

3. Вычислить новые состояния нейронов второго слоя:

$$\mathbf{O}_j^{(2)}(t+1) = \mathbf{V}\mathbf{O}_j^{(2)}(t), \quad j = \overline{1, M}.$$

4. Если $\mathbf{O}^{(2)}(t+1) = \mathbf{O}^{(1)}(t)$, то распознавание закончено. Активный нейрон 2-го слоя определяет номер распознанного образа. Иначе перейти к шагу (3).

3.3. Двунаправленная ассоциативная память (сеть Коско)

3.3.1. Структура сети Коско

С формальной точки зрения, двунаправленная ассоциативная память (есть Коско) – это сеть с двумя компонентами: внутри компонент связей нет, а между компонентами связи полные, причем связь между парой нейронов в обе стороны имеет одинаковый вес (рис. 3.4).

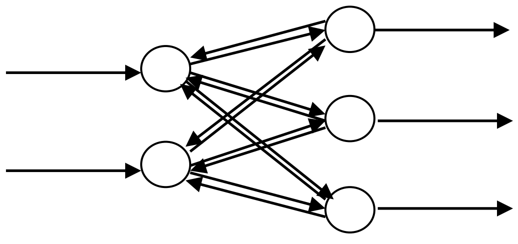


Рис. 3.4. Двунаправленная ассоциативная память

Такая структура сети позволяет перейти от реализации автоассоциативной памяти к реализации гетероассоциативной памяти: на одну из компонент подаются входные сигналы, а с другой, после получения стационарного сигнала, снимаются выходные сигналы.

Можно дать и другую графическую интерпретацию такой сети (рис. 3.5). В представленном виде двунаправленная ассоциативная память является прямым развитием сети Хопфилда, где слой нейронов по-прежнему возвращает выходные сигналы сам на себя. Два слоя сети Коско можно рассматривать как последовательные состояния сети Хопфилда на двух итерациях преобразования сигнала. При этом для обоих слоев используется одна и та же весовая матрица, но выходной сигнал на каждом такте переключается между размерностью входного вектора и размерностью выходного вектора.

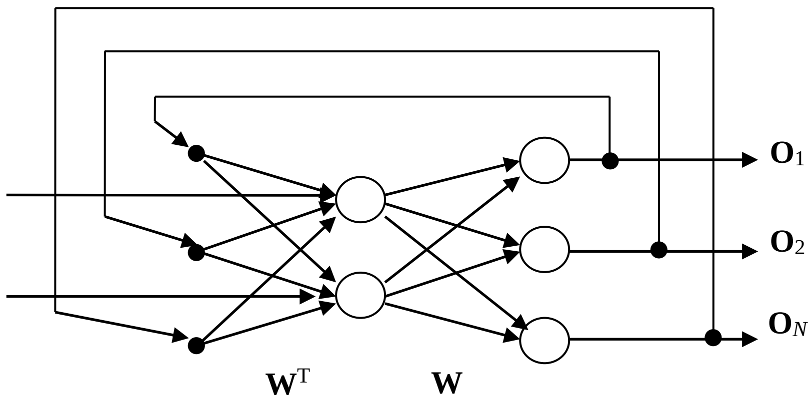


Рис. 3.5. Альтернативное представление двунаправленной ассоциативной памяти

Пусть сеть действует в синхронном режиме – т.е. все выходы слоя вычисляются до того, как они будут переданы на следующий слой. Если матрица весов квадратная и симметричная (т.е. $\mathbf{W} = \mathbf{W}^T$), то входы и выходы – это одно и тоже, и двунаправленная ассоциативная память вырождается в сеть Хопфилда, обеспечивая функцию автоассоциативной памяти. В общем случае сеть Коско дает возможность организовать не автоассоциативную, а гетероассоциативную память – когда по заданному входу восстанавливается ассоциированный с ним выход.

По ходу функционирования обученной сети она сходится к устойчивому состоянию аналогично сети Хопфилда:

$\mathbf{Y}(1) = F(\mathbf{W}, \mathbf{X}(1))$;

$\mathbf{X}(2) = F(\mathbf{W}, \mathbf{Y}(1))$;

$\mathbf{Y}(2) = F(\mathbf{W}, \mathbf{X}(2))$;

…

3.3.2. Обучение сети

Если функционирование сети Хопфилда – это оптимизационный поиск в пространстве входных сигналов **X**, то функционирование сети Коско – это оптимизационный поиск в пространстве сигналов (**X**,**Y**). При этом движение по координатам, входящим в **X**, и по координатам, входящим **Y**, происходит поочередно.

Правило вычисления весов построено по тому же принципу, что и для сети Хопфилда. Для множества обучающих пар ($\mathbf{X}_k$,$\mathbf{Y}_k$) значения весов матрицы:

$$\mathbf{W} = \sum_{i=1}^{K} \mathbf{X}_i \mathbf{Y}_i^{\mathrm{T}},$$

где K – количество запоминаемых образов.

Максимальное допустимое количество запоминаемых образов оценивается величиной $\dfrac{N}{4\log_2 N}$.

Двунаправленная ассоциативная память свободна от ограничений, которые использовались в сети Хопфилда для обеспечения ее сходимости к стационарному состоянию:

– матрица весов может содержать любые значения;
– сеть может работать и в асинхронном и в синхронном режиме;
– активационная функция нейронов может быть непрерывной.

3.4. Многослойные сети с обратными связями

Если двунаправленная ассоциативная память – это обобщение сети Хопфилда на случай с 2 слоями, то можно рассматривать и случаи с большим количеством слоев, сохраняя требование к наличию обратных связей между какими-либо слоями.

Сеть Джордана – это нейронная сеть, которую можно получить из многослойного перцептрона, если на его вход подать, помимо входного вектора, выходной с задержкой на один или несколько тактов (рис. 3.6). С одной стороны, такая сеть, подобно многослойному перцептрону, может реализовать более сложную функциональную зависимость выходных сигналов от входных, чем сеть Коско. При этом появляется возможность ее дообучения с помощью метода обратного распространения ошибки. С другой стороны, подобно другим сетям с обратными связями, веса сети Джордана можно рассчитать по обучающей выборке без применения ресурсоемкой итерационной процедуры обучения.

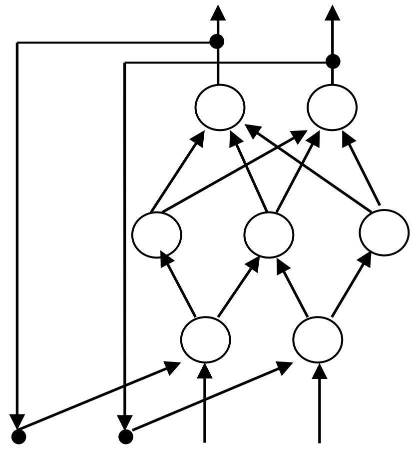


Рис. 3.6. Структура сети Джордана

Сеть Элмана отличается тем, что обратные связи идут не от выхода сети, а от выходов внутренних нейронов (рис. 3.7). Такая структура сети хорошо подходит для обработки информации, представленной как последовательность сигналов.

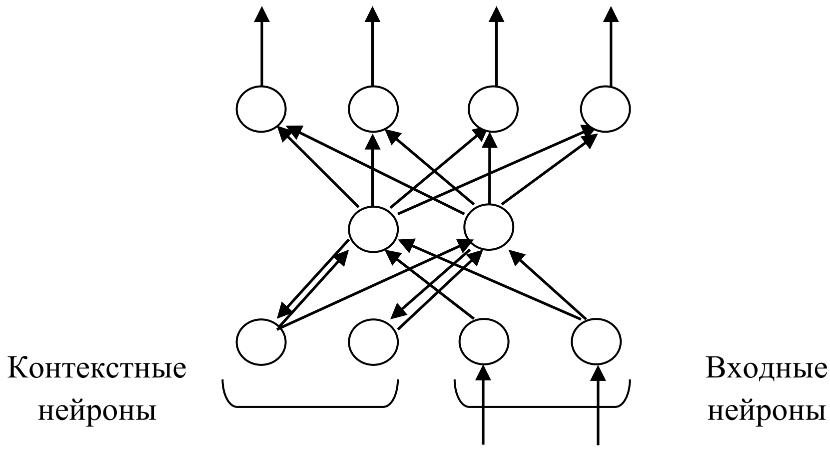


Рис. 3.7. Структура сети Элмана

Рассмотрим частный случай, когда сеть включает 3 слоя с количеством нейронов $2N$, N и $2N$, соответственно. Входной сигнал подается на первые N нейронов первого слоя (будем называть их «входными»), а на остальные нейроны первого слоя подается сигнал со второго слоя сети (будем называть эти нейроны «контекстными»).

На первой итерации на вход и выход сети подается 1-й сигнал последовательности и она обучается методом обратного распространения ошибки.

Затем на входные нейроны 1-го слоя подается 2-й сигнал, а на контекстные нейроны 1-го слоя – сигнал со скрытого слоя. Аналогичный сигнал подается в качестве целевого на выход сети, и она дообучается согласно методу обратного распространения ошибки.

Далее процесс повторяется для последующих входных сигналов. В процессе такого рекурсивного обучения сеть приобретает свойства фильтра: выход сети зависит не только от текущего входного сигнала, но и от входных сигналов, поступивших в сеть ранее. Соответственно, выходной сигнал представляет из себя сглаженную версию входного, в котором шумы подавляются за счет сравнения со входными сигналами, полученными сетью ранее.

3.5. Сеть адаптивного резонанса

Если при обучении нейронной сети предъявляется только фиксированный набор обучающих векторов, они могут подаваться циклически. Если, однако, полностью обученная сеть должна запомнить новый обучающий вектор, он может изменить веса связей настолько, что потребуется полное переобучение сети. Поэтому к нейронной сети, действующей в изменяющихся условиях, предъявляется два противоречивых требования:

– сеть должна выявлять образы новых классов, ранее не представленных в обучающей выборке (пластичность);

– изученные классы образов должны сохраняться (стабильность).

Проблема стабильности-пластичности решается в рамках теории адаптивного резонанса. Общий порядок работы сети адаптивного резонанса следующий.

1. Изначально в слое распознавания все нейроны «свободны» и не хранят информацию о каком-либо образе.
2. Очередной входной вектор классифицируется согласно множеству уже запомненных образов. В результате активируется один из нейронов слоя распознавания.
3. Входной вектор сравнивается с запомненным образом. Если они достаточно похожи – процесс распознавания заканчивается, а информация о распознанном образе корректируется.
4. Иначе данный запомненный образ пропускается и выбирается следующий по схожести со входным вектором образ и т.д.
5. Если входной вектор слишком сильно отличается от всех уже запомненных образов – в памяти сети обеспечивается создание нового образа, а существующие образы не меняются.

3.5.1. Структура сети АРТ-1

Функционирование сети основано на простых принципах, которые можно показать на двоичных образах (это соответствует модели АРТ-1). В модели АРТ-2 эти принципы обобщаются на случай вещественных векторов, но математическая модель получается более громоздкой. Ограничимся рассмотрением модели АРТ-1 (рис. 3.8).

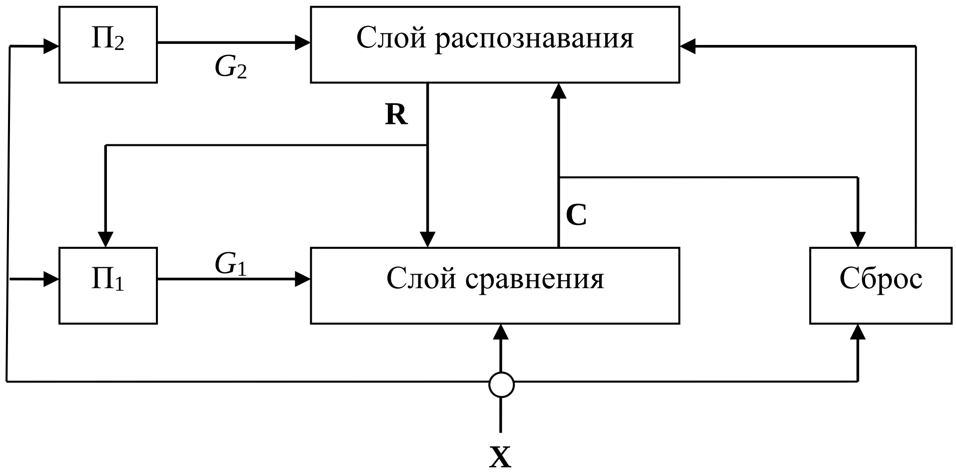


Рис. 3.8. Структурная схема сети АРТ-1

Структурно сеть АРТ-1 включает следующие блоки.

1. Слой сравнения (рис. 3.9).

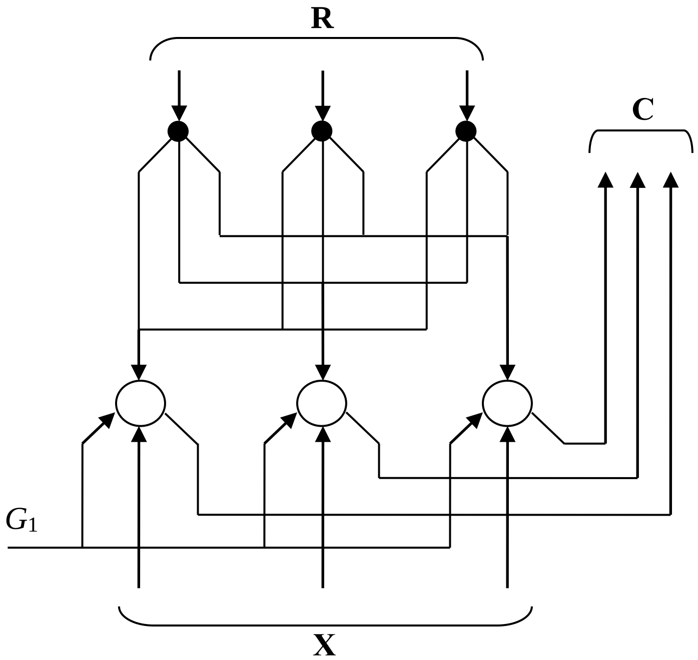


Рис. 3.9. Слой сравнения сети АРТ-1

В необученной сети слой сравнения просто передает входной вектор **X** на выход. В обученной сети каждый нейрон получает на входе:

– компоненту вектора **X**;

– общий скалярный сигнал от блока $П_1$;

– сигналы обратной связи со всех нейронов слоя распознавания (только один из них может быть равен 1), взвешенные бинарными весами.

Выход нейрона сравнения равен 1, если хотя бы два из его входов равны 1.

2. Слой распознавания. Слой распознавания классифицирует полученные на входе вектора, активируя только один из своих нейронов, имеющий наиболее похожий на входной вектор веса. Технически такое поведение обеспечивается тормозящими связями внутри слоя нейронов.

3. Блок $П_2$ реализует логическое ИЛИ от компонент входного вектора **X**.

4. Блок $П_1$ реализует логическое ИЛИ от компонент входного вектора **X**, которое перемножается на отрицание логического ИЛИ от компонент вектора **R**. Таким образом, таблица истинности для логической функции, реализуемой этим блоком, имеет следующий вид (табл. 1).

Таблица 1

ИЛИ от компонент вектора **X**	ИЛИ от компонент вектора **R**	Сигнал G_1
0	0	0
1	0	1
1	1	0
0	1	0

5. Блок сброса вырабатывает сигнал сброса, посылаемый на слой распознавания, если расстояние Хемминга между векторами **X** и **C**, т.е. количество отличающихся бит в этих векторах, больше некоторого порога.

3.5.2. Алгоритм работы сети

Процесс классификации в сети АРТ-1 состоит из 4 основных фаз: распознавание, сравнение, поиск и обучение.

1. Распознавание:

– изначально $G_2 = 0$, поэтому все нейроны слоя распознавания пассивны;

– при подаче входного вектора **X** сигналы G_1 и G_2 получают значение 1;

– в результате слой сравнения начинает выдавать на выходе вектор **X** (рис. 3.10, а);

– нейрон слоя распознавания с весами, наиболее близкими к **X**, выигрывает и на выходе слоя распознавания возникает вектор **R**, одна из компонент которого равна 1 (рис. 3.10, б).

2. Сравнение:

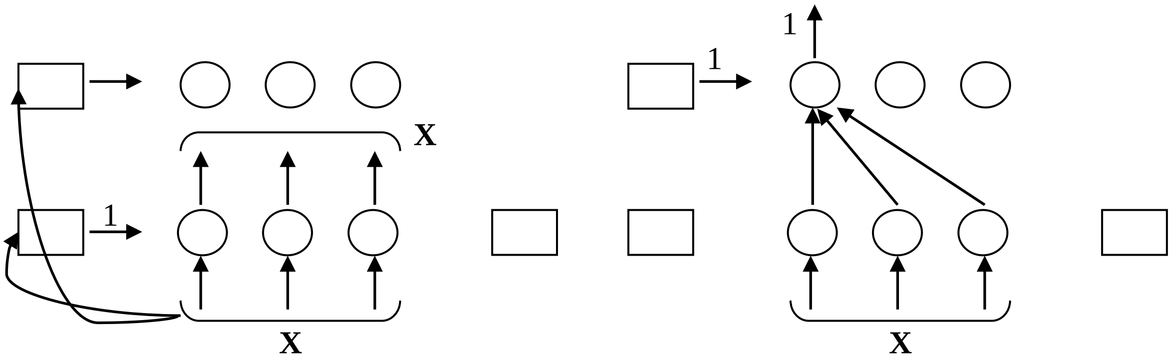
– единичный сигнал с единственного нейрона слоя распознавания рассылается на все нейроны слоя сравнения через бинарные веса (те же самые, которые являются входными для нейрона распознавания и определяют запомненный им образ);

– сигнал G_1 обнуляется – в результате в слое сравнения активируются только нейроны, одновременно получающие единицу и со входа, и от слоя распознавания (рис. 3.11, а);

– если расстояние Хемминга между **X** и **P** велико – несколько нейронов на фазе сравнения будут возбуждаться и **C** будет содержать много нулей, в то время как **X** содержит единицы;

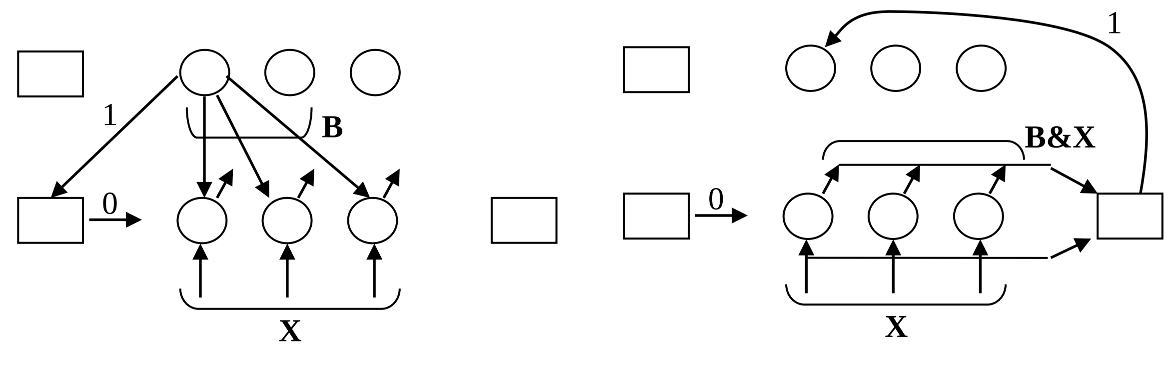
– это означает, что возвращенный вектор **P** не является искомым и возбужденные нейроны в слое распознавания должны быть заторможены;

– это торможение производится блоком сброса, который сравнивает входной вектор **X** и вектор **C** и вырабатывает сигнал сброса, если степень сходства этих векторов меньше некоторого уровня (рис. 3.11, б).



а) б)

Рис. 3.10. Распространение сигналов в сети АРТ-1 на фазе распознавания: а) для слоя сравнения; б) для слоя распознавания



а) б)

Рис. 3.11. Распространение сигналов в сети АРТ-1 на фазе сравнения: а) для слоя сравнения; б) для блока сброса

3. Поиск:

– если не выработан сигнал сброса, то сходство признается достаточным и процесс классификации завершается;

– в противном случае другие запомненные образы должны быть исследованы с целью поиска лучшего соответствия; при этом торможение возбужденного нейрона в распознающем слое приводит к установке всех компонент вектора **R** в 0, G_1 устанавливается в 1 и входной вектор **X** опять без изменений передается в **C** (рис. 3.12);

– в слое распознавания снова ищется нейрон с весами, наиболее похожими на входной вектор; однако предыдущий ранее отвергнутые блоком сброса нейроны в поиске уже не участвуют (это обеспечивается весами внутренних связей слоя распознавания);

– новый нейрон выигрывает соревнование в слое распознавания и другой запомненный образ **P** возвращается в слой сравнения;

– если **P** не соответствует **X**, то возбужденный нейрон в слое распознавания снова тормозится.

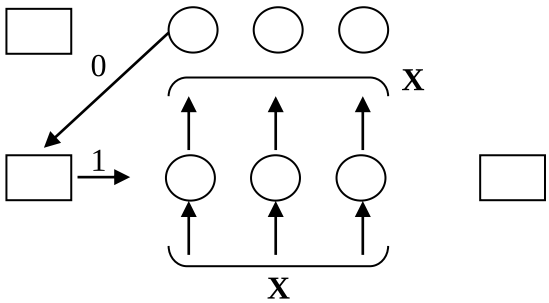


Рис. 3.12. Распространение сигналов в сети АРТ-1 на фазе поиска

Этот процесс повторяется до тех пор, пока не встретится одно из двух событий:

– найден запомненный образ, сходство которого с вектором **X** выше уровня параметра сходства;

– все запомненные образы проверены, определено, что они не соответствуют входному вектору, и все нейроны слоя распознавания заторможены.

В последнем случае предварительно не распределенный нейрон в слое распознавания выделяется этому образу и его весовой вектор устанавливаются соответствующим новому входному образу. Неудачный поиск будет автоматически завершаться на несвязанном нейроне, так как его веса все равны единице – своему начальному значению.

4. Обучение.

После того, как окончательно выбран победивший нейрон, проводится дообучение сети, в процессе которого модифицируются веса связей возбужденного нейрона слоя распознавания:

– если этот нейрон уже хранит какой-то образ, то информация о новом векторе скорректирует этот образ;

– если это новый необученный нейрон, то его веса будут идентичны входному вектору.

3.5.3. Поведение сети

Доказан ряд полезных свойств, которым обладают сети адаптивного резонанса.

1. Повторное предъявление вектора, который уже использовался для обучения сети, сразу приведет к активации нужного нейрона слоя распознавания, т.е. время на фазу поиска не потребуется.
2. Фаза обучения является устойчивой, т.е. она гарантированно не приведет к смене активного нейрона.
3. Процесс запоминания образов конечен, т.е. сеть полностью обучится за конечное число шагов не только для конечной последовательности входных векторов, но и для бесконечной последовательности циклически повторяющихся векторов (в отличие от других типов сетей, которые в этом случае могут осциллировать).

Сделаем также несколько замечаний относительно особенностей реализации сети АРТ-1.

1. При использовании двоичных сигналов каждый запомненный образ, по сути, представляет собой логическое произведение обучающих векторов, которые были отнесены сетью к данному образу. Если какая-то компонента запомненного образа по ходу обучения получила значение 0, то она уже никогда не вернется к значению 1. Именно поэтому изначально необученные нейроны имеют весовые вектора, целиком состоящие из значений 1.
2. Необходимое количество нейронов в слое распознавания, по сути, определяется по ходу запоминания предъявленных образов: как только сеть обнаруживает входной вектор, слишком сильно отличающийся от всех ранее запомненных – она задействует новый нейрон в слое распознавания.
3. Сеть должна производить последовательный поиск среди всех запомненных образов. В аналоговых реализациях это может происходить очень быстро, однако при моделировании на цифровых компьютерах этот процесс может оказаться очень длительным. Если же сеть адаптивного резонанса реализуется на параллельных процессорах, то все свертки на распознающем уровне могут вычисляться одновременно. В этом случае поиск также не потребует большого времени.

Контрольные вопросы к главе 3

1. В чем состоит разница между автоассоциативной и гетероассоциативной памятью?
2. Чем обусловлено ограничение на количество образов, которые могут быть сохранены в ИНС, реализующей ассоциативную память?
3. Чем отличаются синхронный и асинхронный режимы работы сети с точки зрения сходимости сигнала?
4. В каких случаях сеть Хэмминга может заменить собою сеть Хопфилда?
5. Можно ли считать сеть Коско обобщением сети Хэмминга на случай с двумя слоями нейронов?
6. Каким образом реализуется режим дообучения в многослойных ИНС с обратными связями?
7. Что такое проблема пластичности-стабильности?
8. Почему сеть АРТ можно считать сетью с переменным количеством нейронов?
9. Используются ли в сети АРТ связи внутри одного слоя нейронов?
10. Почему новые нейроны в слое распознавания сети АРТ инициализируются весами, равными 1?

Глава 4. ПРИКЛАДНЫЕ АСПЕКТЫ ИСПОЛЬЗОВАНИЯ НЕЙРОННЫХ СЕТЕЙ

4.1. Поиск глобального оптимума в нейронных сетях

В большинстве нейронных сетей используется механизм итеративного поиска:

– для сетей прямого распространения он применяется при обучении нейронной сети;

– для сетей с обратными связями он применяется при расчете выходного сигнала обученной сети.

Этот механизм подвержен следующему принципиальному недостатку: он обеспечивает сходимость процедуры к ближайшему локальному оптимуму, который:

а) может быть существенно хуже глобального оптимума;

б) может представлять из себя «ложный образ», не соответствующий обучающим данным.

Для борьбы с этим недостатком можно использовать следующие подходы:

1) введение штрафных функций, стремящихся вытолкнуть поиск из локального минимума (тем самым приблизив многоэкстремальную функцию к одноэкстремальной);

2) введение случайно составляющей в итерационную процедуру: если поиск попал в точку локального минимума, то благодаря случайной составляющей он может оттуда выбраться;

3) параллельный поиск решения на системе различных точек оптимизируемой функции: они охватывают область поиска целиком, тем самым избегая локализации поиска в окрестности одного оптимума.

Рассмотрим подробнее методы группы (2) (вероятностные методы) и (3) (эволюционные методы).

4.1.1. Стохастическое обучение нейронной сети

Общая схема стохастического обучения предполагает следующую последовательность действий.

1. Выбрать изменяемый весовой коэффициент (или группу коэффициентов).

2. Подвергнуть их случайному изменению (согласно некоторому вероятностному распределению, которое зависит от текущего состояния процесса обучения).

3. Оценить возникшее изменение целевой функции (в случае обучения с учителем – как изменение качества работы нейронной сети на

обучающей выборке, в случае самообучения – как изменение значения «энергии» сети).

4. Если коррекция улучшает значение целевой функции – сохранить ее, иначе – сохранить ее с некоторой вероятностью (вероятность зависит от текущего состояния процесса обучения).

5. Если не достигнут критерий останова, то повторить процесс с шага (1).

Размер выполняемых вероятностных изменений должен постепенно снижаться с определенной скоростью:

– если изменения станут слишком малы, когда сеть еще находится в окрестности «плохого» локального минимума, то она не сможет оттуда выбраться;

– если изменения будут слишком велики, то сеть будет постоянно перескакивать между локальными окрестностями «плохих» и «хороших» локальных оптимумов.

На шаге (4) должны с некоторой вероятностью сохраняться изменения, ухудшающие целевую функцию. Можно использовать следующее правило:

$$P(c) = e^{-\frac{c}{kT(t)}}.$$

где $P(c)$ – вероятность изменения c в целевой функции; k – константа, t – номер итерации обучения, $T(t) = \dfrac{T_0}{1+t}$ – искусственная температура, которая снижается по ходу обучения.

Выбирается случайное число r из равномерного распределения от нуля до единицы. Если $P(c)$ больше, чем r, то изменение сохраняется, в противном случае величина веса возвращается к предыдущему значению.

На шаге (5) в качестве критерия останова может использоваться:

– пороговое значение целевой функции;

– ограничение на общее кол-во итераций;

– ограничение на кол-во последних итераций, не улучшивших значение целевой функции.

4.1.2. Использование генетических алгоритмов в нейронных сетях

Возможны различные варианты совместного использования нейронной сети и генетических алгоритмов [8]:

1) предобработка данных для нейронной сети генетическим алгоритмом;

2) обучение нейронной сети генетическим алгоритмом;

3) определение структуры нейронной сети с помощью генетического алгоритма;

4) управление скоростью обучения нейронной сети с помощью генетического алгоритма;

5) генерация начальной популяции экземпляров генетического алгоритма с помощью нейронной сети.

Рассмотрим некоторые из этих возможностей.

Предобработка исходных данных для нейронной сети

Входной вектор, описывающий состояние объекта, пропускается через блок генетического алгоритма, который сжимает его. Затем сжатый вектор подается на вход нейронной сети, которая уже решает задачу классификации с помощью стандартных методов обучения или самообучения. Поскольку размерность входа нейронной сети сокращается, то уменьшается и вычислительная сложность алгоритмов ее обучения.

Блок генетического алгоритма может выполнять сжатие в виде:

а) нелинейного преобразования пространства параметров объектов, с использованием метода опорных векторов;

б) селекции части признаков из исходного вектора.

В случае (б) метод кодирования экземпляров и операторы для генетического алгоритма очевидны, но критерий качества требует обучить нейронную сеть для каждого предложенного генетическим алгоритмом набора отобранных признаков; возможен вариант, когда по каждому новому экземпляру, порожденному генетическим алгоритмом, уже обученная сеть родителя только пытается дообучиться за небольшое число шагов.

Генетические алгоритмы для определения структуры нейронной сети

Генетический код может определять такие структурные характеристики ИНС как количество нейронов, наличие связей между ними, вид активационных функций.

Возможны следующие подходы к кодированию.

1. Непосредственное кодирование: код определяет наличие или отсутствие связи, тип активационной функции для каждой пары нейронов.

Рассмотрим пример: код сети из N нейронов – это квадратная матрица размера $N \times N$ со значениями из множества $\{0,1\}$, которая определяет наличие связей для каждой пары нейронов. Недостаток такого варианта кодирования – быстрый рост длины кода с увеличением объема сети. Его можно нивелировать дополнительными ограничениями, в частности:

– между парой нейронов сигнал распространяется только в одну сторону – в этом случае будет достаточно половины матрицы;

– сеть слоистая – в этом случае в матрице будут задействованы только отдельные блоки элементов (рис. 4.1).

2. Косвенное кодирование: код определяет только интегральные характеристики сети, такие как общее количество нейронов, количество слоев, коэффициент полноты связей между слоями.

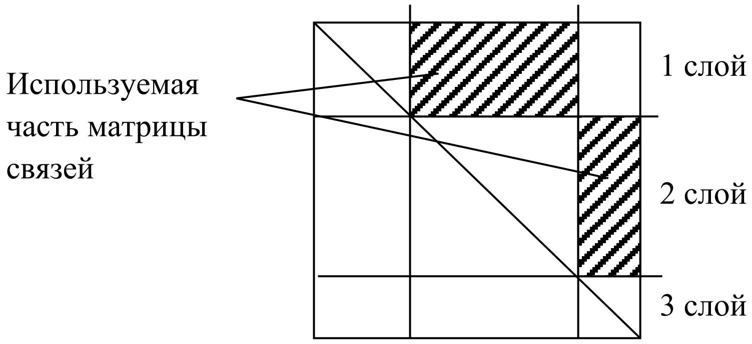


Рис. 4.1.Прямое кодирование структуры связей для слоистой сети (на примере 3-слойной сети)

Генетические алгоритмы для обучения нейронной сети

Рассмотрим теперь случай, когда топология нейронной сети фиксирована, а каждый экземпляр популяции в генетическом алгоритме определяет значения весов всех связей в сети. Оптимизируемая целевая функция – это качество работы экземпляра сети на объектах обучающей выборки. Таким образом, алгоритм обучения, обеспечивающий настройку весов одной ИНС, заменяется на генетическую оптимизацию по множеству ИНС с разными значениями весов.

Порядок работы алгоритма обучения следующий:

1) создать начальное поколение сетей, описанных случайными кодами;

2) выполнить рекомбинацию и мутацию кодов;

3) рассчитать уровень ошибок каждой сети-экземпляра на обучающей выборке, в соответствии с ее кодом;

4) если требуемое качество распознавания еще не достигнуто – выполнить отбор кодов для следующего поколения и перейти к шагу (2).

Преимущества такого алгоритма обучения перед стандартным:

– возможность поиска глобального, а не локального минимума;

– в отличие от алгоритма обратного распространения ошибки нет необходимости вычислять градиент, т.е. не нужно брать аналитические производные от функций активации нейронов; поэтому функции активации нейронов могут быть любыми, в т.ч. недифференцируемыми.

4.2. Гибридные нечетко-нейронные системы

Использование аппарата нечеткой логики позволяет представлять знания в форме, которая легко интерпретируется экспертом. Однако определение функции принадлежности посылок и заключений, используемых в нечетких правилах, происходит экспертно, без согласования с данными измерений, что ограничивает эффективность таких систем. Нейронные сети, в свою очередь, обеспечивают возможность извлечения знаний о структуре анализируемых процессов из формальных данных измерений, однако зависимость между входными данными и результатами не наглядна и не интерпретируема экспертом.

Возможны следующие варианты синтеза, совмещающие достоинства этих подходов [8, 13].

1. Фаззификация отдельных элементов нейронной сети:

– замена арифметических операций в нейроне на нечеткие логические;

– использование нечеткого вывода в процессе обучения нейронной сети.

2. Использование нейронной сети для описания отдельных элементов знаний в рамках парадигмы нечеткой логики:

– моделирование функций принадлежности нейронной сетью;

– моделирование нечетких логических операций нейронной сетью;

– моделирование отношения «посылка – заключение» нейронной сетью;

– управление процессом вывода в нечеткой продукционной системе с помощью нейронной сети.

4.2.1. Нечеткие числа в нейронной сети

Возможны следующие варианты замены арифметических операций в нейроне на нечеткие логические:

– нейрон имеет четкие входные сигналы, но нечеткие веса;

– нейрон имеет нечеткие входные сигналы и нечеткие веса;

– нейрон имеет нечеткие входные сигналы, нечеткие веса и нечеткую функцию активации.

Нейрон с четкими входными сигналами и нечеткими весами

Каждый весовой коэффициент представляет собой нечеткое множество, которое задано характеристической функцией. Вместо взвешенной суммы входных сигналов в активационную функцию нейрона в качестве аргумента подается некоторый «уровень правдоподобия», вычисленный на основе n входных сигналов и n функций принадлежности для них:

$$m(\mathbf{X}_1,...,\mathbf{X}_n) = m_{\mathbf{W}_1}(\mathbf{X}_1) \otimes \ldots \otimes m_{\mathbf{W}_n}(\mathbf{X}_n),$$

где $\otimes$ – нечеткая логическая операция, позволяющая агрегировать нечеткие логические сигналы, поступающие с разных входов.

Выходной сигнал нейрона при этом остается числовой величиной и подается на входы других нейронов, имеющих нечеткие весовые коэффициенты.

Нейрон с нечеткими входными сигналами и нечеткими весами

Каждый вход нейрона представляет собой нечеткий факт, описанный характеристической функцией. Этот нечеткий факт сопоставляется с нечеткой посылкой (весом). В результате определяется характеристическая функция пересечения нечеткого факта и нечеткой посылки. Дальше они агрегируются по всем входам с помощью какой-либо операции над нечеткими множествами:

$$m(\mathbf{X}_1,...,\mathbf{X}_n) = (m_{\mathbf{W}_1} * m_{X_1}) \otimes \ldots \otimes (m_{\mathbf{W}_n} * m_{\mathbf{X}_n}),$$

где $*$ – операция взвешивания нечеткого множества входного сигнала $\mathbf{X}_i$ нечетким множеством веса $\mathbf{W}_i$, $\otimes$ – нечеткая операция над множествами, позволяющая агрегировать нечеткие множества сигналов, поступающих с разных входов.

Таким образом, $m(\mathbf{X}_1,...,\mathbf{X}_n)$ задает нечеткое множество, которое является выходом нейрона и может подаваться как входной сигнал на последующие нейроны.

Нейрон с нечеткой функцией активации

В отличие от предыдущего случая, к полученному набору нечетких множеств с характеристическими функциями $(m_{\mathbf{W}_i} * m_{X_i})$ применяется операция импликации – т.е. каждая связь «i-й вход – выход» рассматривается как нечеткое продукционное правило, а весь нейрон – как база продукционных правил.

Их применение означает, что по каждому входу выполняется редукция выхода, а потом полученные нечеткие множества объединяются. Полученное в результате нечеткое множество и есть выходной сигнал нейрона.

После получения сигналов на выходе всей нейронной сети надо выполнить их дефаззификацию. Чаще всего используется формула центра масс. Для дискретной характеристической функции имеем:

$$\mathbf{Y}_{\text{четк}} = \frac{\sum m_{\mathbf{Y}}(\mathbf{Y}_j)\mathbf{Y}_j}{\sum m_{\mathbf{Y}}(\mathbf{Y}_j)},$$

где $\mathbf{Y}_j$ – всевозможные значения аргумента характеристической функции m.

В непрерывном случае сумма заменяется на интеграл.

4.2.2. Нечеткое обучение нейронной сети

Вариант использования нечеткой логики при обучении нейронной сети сильно зависит от ее архитектуры и используемой процедуры обучения. Рассмотрим случай обучения одного нейрона со ступенчатой функцией активации методом обратного распространения ошибки.

Когда множества объектов разных классов, представленные в обучающей выборке, не могут быть разделены перцептроном, возникает некоторая «зона перекрытия» разделяемых классов. Объекты из этой зоны дают алгоритму обучения противоречивую информацию и делают поведение алгоритма обучения неустойчивым. Идея использования нечеткости состоит в том, чтобы уменьшить влияние на алгоритм обучения обучающих векторов, лежащих в зоне перекрытия.

Пусть принадлежность *N*-мерного вектора входов нейрона к каждому из двух классов описывается характеристической функцией:

$$\begin{cases} m_1(\mathbf{X}) : \mathbf{R}^N \to [0;1]; \\ m_2(\mathbf{X}) : \mathbf{R}^N \to [0;1]; \\ m_1(\mathbf{X}_k) + m_1(\mathbf{X}_k) = 1, \forall \mathbf{X}_k \in \mathbf{X}^K, \end{cases}$$

где $\mathbf{X}^K$ – обучающая выборка.

Степень влияния каждого объекта обучающей выборки на алгоритм определяется величиной

$$\mathrm{S}(\mathbf{X}_k, a) = \left| m_1(\mathbf{X}_k) - m_2(\mathbf{X}_k) \right|^a, \quad a = const, \quad a > 0 .$$

Таким образом, чем ближе оценки принадлежности к ситуации неопределенности $m_1(\mathbf{X}_k) = m_2(\mathbf{X}_k) = 0{,}5,$ – тем меньше объект $\mathbf{X}_k$ должен влиять на алгоритм обучения сети.

Для алгоритма обратного распространения ошибки имеем:

$$\mathbf{W}_j(t+1) = \mathbf{W}_j(t) + h\mathrm{S}(\mathbf{X}_k, a)(\mathbf{P}_k - \mathbf{Y}(t))\mathbf{X}_k , \tag{4.1}$$

где $\mathbf{P}_k$ – целевое значение выхода нейрона.

Чем ближе значение параметра *a* к 0, тем ближе поведение алгоритма обучения к четкому случаю (т.к. $S(\mathbf{X}_k, a) \to 1, \forall \mathbf{X}_k \in \mathbf{X}^K$).

Характеристические функции для объекта $\mathbf{X}_k$, принадлежащего классу 1, задаются по формулам:

$$\begin{cases} m_1(\mathbf{X}) = \dfrac{1}{2} + \dfrac{1}{2} \dfrac{e^{d_2(\mathbf{X}) - d_1(\mathbf{X})} - e^{-f}}{e^f - e^{-f}}; \\ m_2(\mathbf{X}) = 1 - m_1(\mathbf{X}), \end{cases}$$

где $d_1(\mathbf{X})$, $d_2(\mathbf{X})$ – расстояние от вектора $\mathbf{X}$ до центра каждого из классов; $f > 2$ – параметр обучения.

Характеристические функции для объектов класса 2 – противоположные.

Критерий останова обучения определяется условием:

$$\left[\begin{array}{l} m_1(\mathbf{X}_k) > \dfrac{1}{2} + \delta; \\ m_2(\mathbf{X}_k) > \dfrac{1}{2} + \delta; \end{array}\right. \quad \forall \mathbf{X}_k \in \mathbf{X}^K.$$

Целиком алгоритм обучения имеет следующий вид.

1. Инициализировать веса $\mathbf{W}$ случайным образом.
2. Вычислить $m_1(\mathbf{X}_k), m_2(\mathbf{X}_k), \mathbf{X}_k \in \mathbf{X}^K$.
3. Установить флаг = 1.
4. Выбрать очередной объект $\mathbf{X}_k \in \mathbf{X}^K$. Если $\mathbf{Y}_k(\mathbf{X}_k) = \mathbf{P}_k$, то перейти к шагу (7)
5. Скорректировать веса $\mathbf{W}$ согласно правилу (4.1).
6. Если не выполнено условие останова, то установить флаг = 0.
7. Если не все примеры обучающей выборки рассмотрены, то перейти к шагу (4).
8. Если флаг = 0, то перейти к шагу (3).

4.2.3. Моделирование элементов продукционных правил нейронной сетью

Общая цель использования нейронных сетей для моделирования различных элементов нечетких систем – обеспечить настройку параметров характеристических функций по фактическим данным измерений, вместо экспертного выбора этих параметров. Дальнейшие пункты демонстрируют применение этого принципа на различных элементах нечеткой системы, которые описываются характеристическими функциями.

Моделирование функции принадлежности нейронной сетью

Пусть в нечеткой продукционной системе в качестве исходных данных используется вещественный параметр x, для которого в продукционных правилах должны использоваться нечеткие понятия «мало», «средне», «много».

Функцию принадлежности понятия «мало» можно задать логистической функцией:

$$m_S(x) = \frac{1}{1 + e^{-w_g(x + w_c)}},$$

где w_g, w_c – параметры, определяющие положение и форму кривой.

Тогда набор характеристических функций $m_S(x)$, $m_M(x)$, $m_L(x)$ для всех трех нечетких понятий можно представить следующей нейронной сетью (рис. 4.2, 4.3). Обучая нейронную сеть (например, методом обратного распространения ошибки) на примерах, получим оптимальные функции принадлежности.

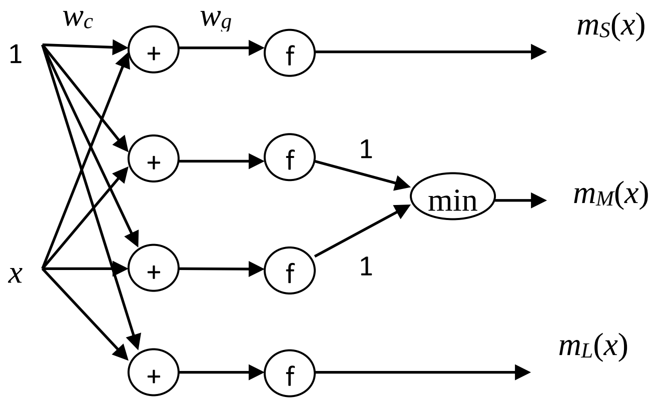


Рис. 4.2. Нейронная сеть, представляющая характеристические функции нечетких множеств для понятий «мало», «средне», «много»

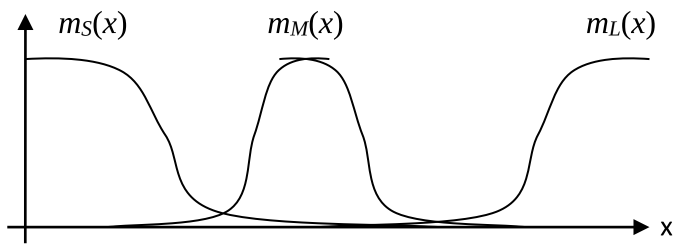


Рис. 4.3. Характеристические функции, представляемые нейронной сетью

Моделирование логической функции нейронной сетью

Аналогичным образом можно построить нейронную сеть, моделирующую логически операторы: на вход поступают значения двух нечетких логических аргументов, а выход интерпретируется как нечеткое значение логической функции от них.

Тогда функции будут априорно заданы в параметрическом виде и веса сети будут определять оптимальные характеристики этих логических функций. Например в качестве нечеткого логического И может использоваться следующая функция:

$$a \text{ И } b = \frac{ae^{-ka} + be^{-kb}}{e^{-a} + e^{-b}}.$$

Здесь параметр k определяет степень «жесткости» операции: при $k \to \infty$ эта функция сходится к $\min(a,b)$ (т.е. к стандартному нечеткому И).

Моделирование логического отношения нейронной сетью

В стандартном случае отношение «посылка – заключение» в нечеткой продукционной системе задается правилом редукции. Оно является максиминной функцией, поэтому для правила $A \to B$ и нечеткого факта A' имеем:

$$m_{B'}(y) = \max\left(\min_{x \in X}\left(m_B(y), m_{A'}(x)\right)\right).$$

Такая функция задает правило определения характеристической функции $m_{B'}(y)$ в каждой точке y, в зависимости от значений характеристической функции $m_{A'}(x)$ на всем множестве X и от значения $m_B(y)$ в точке y.

В общем случае функция отношения может быть любой. Если использовать конечные множества $X = \{x_1, ..., x_n\}$ и $Y = \{y_1, ..., y_k\}$, то эту операцию можно записать в матричном виде:

$$\left[m_{B'}(y_1)\; m_{B'}(y_2) \ldots m_{B'}(y_k)\right] =$$

$$= \left[m_{A'}(x_1)\; m_{A'}(x_2) \ldots m_{A'}(x_n)\right] \bullet \begin{bmatrix} m_R(x_1, y_1) & \ldots & m_R(x_1, y_k) \\ \ldots & \ldots & \ldots \\ m_R(x_n, y_1) & \ldots & m_R(x_n, y_k) \end{bmatrix},$$

где $m_R(x, y)$– характеристическая функция операции отношения, $\bullet$ – некоторая операция свертки вектора.

В случае редукции роль операции свертки выполняет max, роль характеристической функции $m_R(x, y)$ выполняет min. Можно обобщить эти операции, задав их нейронной сетью следующего вида (рис. 4.4).

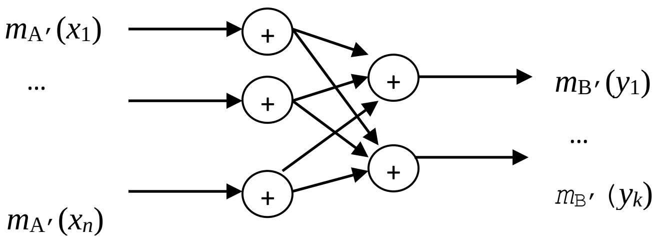


Рис. 4.4. Нейронная сеть, моделирующая нечеткое логическое отношение

В такой сети входные сигналы в совокупности составляют $m_{A'}(x)$, выходные сигналы в совокупности составляют $m_{B'}(y)$, веса связей между слоями задают характеристическую функцию $m_R(x, y)$, активационная функция нейронов определяет операцию свертки вектора.

Контрольные вопросы к главе 4

1. Каким образом процедуры оптимизационного поиска задействованы в работе нейронных сетей без обратных связей и с обратными связями?
2. Какие возможности поиска глобального минимума функции ошибки могут быть использованы в нейронных сетях?
3. Какие критерии останова могут применяться при использовании стохастического подхода к обучению нейронной сети?
4. Чем отличается прямое и косвенное кодирование структуры нейронной сети при использовании генетических алгоритмов?
5. Какие структурные элементы нейронной сети могут быть определены в результате оптимизации с помощью генетического алгоритма?
6. Каковы преимущества и недостатки генетического алгоритма обучения по сравнению с алгоритмом обратного распространения ошибки?
7. В определении каких элементов ИНС могут использоваться нечеткие множества и нечеткие числа?
8. С какой целью стандартные характеристические функции, используемые в нечеткой логике, заменяются на характеристические функции, моделируемые нейронной сетью?
9. Какие элементы системы нечетких продукционных правил могут быть описаны с помощью ИНС?

ЗАКЛЮЧЕНИЕ

В основу архитектуры всех рассмотренных видов искусственных нейронных сетей положен ряд общих принципов.

1. Феноменологический подход (принцип «черного ящика»): сеть должна воспроизвести «правильное» поведение некоторой управляющей системы, не моделируя ее структуру, а только симулируя ее внешние проявления, т.е. зависимость выходного сигнала от входного.

2. Гибкость функциональной зависимости, представляемой нейронной сетью, которая достигается за счет использования большого количества простых однотипных элементов.

3. Высокий параллелизм вычислений. Реальный эффект он него может возникнуть только при физическом распараллеливании вычислений, а не при эмуляции на последовательной вычислительной машине.

Использование этих принципов позволяет обеспечить нейронную сеть рядом важных свойств, необходимых при решении практических задач оценивания, прогнозирования, поддержки принятия решений:

– обучение – чем больше обучающих примеров предъявлено сети, тем более сложные зависимости она может описывать;

– обобщение – отклик сети после обучения становится до некоторой степени нечувствителен к небольшим изменениям входных сигналов, т.е. сеть позволяет обнаруживать образ сквозь шум и искажения;

– абстрагирование – обучение на наборе искаженных примеров позволяет сформировать неискаженный образ, который в «чистом» виде никогда не предъявлялся сети.

Подобно другим классам систем обнаружения знаний, ИНС может обучаться в режиме обучения с учителем или самообучения. Самообучение – более биологически правдоподобный режим, когда речь идет о низком уровне абстрагирования, обучение с учителем – когда речь идет о высоком уровне абстрагирования.

Чаще всего на ИНС возлагается задача классификации входных сигналов. Однако она может использоваться и для решения других задач [11, 12]: оптимизационный поиск, сжатие информации, управление с обратной связью. Ряд архитектур нейронных сетей, рассмотренных в учебном пособии, обеспечивает механизмы адаптации сети к «проблемным» свойствам исходных данных, возникающим в реальных задачах такого рода. В частности, были рассмотрены следующие механизмы адаптации.

В сети адаптивного резонанса используется гибкий подход к созданию кластеров: новые предъявленные образы тестируются на степень подобия уже существующим классам. В результате в память сети можно добавлять данные постепенно, по мере получения информации в реальном мире. Это механизм адаптации к постепенному появлению новых обучающих данных и их неустойчивости во времени.

В когнитроне за счет ограниченных областей связи обеспечивает последовательное обобщение данных по слоям. Тем самым задача аппроксимации слишком сложной функциональной зависимости одной функцией заменяется на задачу аппроксимации суперпозицией нескольких функций. Это механизм адаптации к высокой сложности моделируемых зависимостей.

В сети радиальных базисных функций задача аппроксимации слишком сложной функциональной зависимости одной функцией заменяется на задачу аппроксимации суммой большого количества простых функций. Это также механизм адаптации к высокой сложности моделируемых зависимостей.

Введение нечетких элементов в нейронную сеть позволяет ей работать с нечеткими и частично истинными исходными данными – это расширяет класс допустимых исходных данных.

Введение стохастической составляющей в процедуру поиска направлено на борьбу с проблемой локальных минимумов, применение генетического поиска преследует ту же цель. Это механизм адаптации к ситуации, когда классифицируемые образы плохо разделимы.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Агеев Д.А., Балухто А.Н. и др. Нейроматематика. Кн. 6: учеб. пособие для вузов. – М.: ИПРЖР, 2002. – 448 с.
2. Барский, А.Б. Логические нейронные сети: учебное пособие / А.Б. Барский. – М.: Интернет-Университет Информационных Технологий, 2007. – 351 с.
3. Бодянский, Е.В. Искусственные Нейронные сети: архитектуры, обучение, применения / Е.В. Бодянский, О.Г. Руденко. – Харьков: Телетех, 2004. – 369 с.
4. Галушкин, А.И. Нейронные сети: основы теории / А.И. Галушкин – М.: Горячая линия-Телеком, 2010. – 496 с.
5. Минский, М. Персептроны / М. Минский, С. Пайперт; пер. с англ. – М.: Мир, 1971. – 261 с.
6. Осовский, С. Нейронные сети для обработки информации / С. Осовский; пер. с польского. – М.: Финансы и статистика, 2002. – 344 с.
7. Розенблатт, Ф. Принципы нейродинамики: перцептрон и теория механизмов мозга / Ф. Розенблатт; пер. с англ. – М.: Мир, 1965. – 480 с.
8. Рутковская, Д. Нейронные сети, генетические алгоритмы и нечеткие системы / Д. Рутковская, М. Пилиньский, Л. Рутковский; пер. с польского. – М.: Горячая линия-Телеком, 2004. – 452 с.
9. Тархов, Д.А. Нейронные сети: модели и алгоритмы / Д.А. Тархов. – М.: «Радиотехника», 2005. – 256 с.
10. Уоссермен, Ф. Нейрокомпьютерная техника: теория и практика / Ф. Уоссермен; пер. с англ. – М.: Мир, 1992. – 184 с.
11. Хайкин, С. Нейронные сети: полный курс / С. Хайкин; пер. с англ. – М.: «Вильямс», 2006. – 1104 с.
12. Ширяев, В.И. Финансовые рынки: нейронные сети, хаос и нелинейная динамика / В.И. Ширяев. – М.: КРАСАНД, 2011. – 232 с.
13. Яхъяева, Г.Э. Нечеткие множества и нейронные сети: учеб. пособие / Г.Э. Яхъяева. – М.: Бином, 2006. – 315 с.

ОГЛАВЛЕНИЕ