Министерство образования Республики Беларусь
Учреждение образования
«Белорусский государственный университет
информатики и радиоэлектроники»

Кафедра электронных вычислительных машин

Р.Х. Садыхов, Е. В. Пушкин, А. О. Шадура

НЕЙРОСЕТЕВЫЕ ТЕХНОЛОГИИ ПРИНЯТИЯ РЕШЕНИЙ

Методическое пособие
для студентов специальности 1-98 80 03
«Аппаратное программно-техническое
обеспечение информационной безопасности»

Минск БГУИР 2012

УДК 004.032.26 (076.6)
ББК 32.813 я73
С14

Рецензент
старший научный сотрудник
лаборатории №222 ОИПИ НАН Беларуси,
кандидат технических наук Е. А. Шестаков

Садыхов, Р. Х.

Б12 Нейросетевые технологии принятия решений: метод. пособие для студ. спец. 1-98 80 03 «Аппаратное программно-техническое обеспечение информационной безопасности» / Р.Х. Садыхов, Е. В. Пушкин, А. О. Шадура. – Минск: БГУИР, 2012. – 42 с. : ил.

ISBN 978-985-488-319-9

Методическое пособие по курсу «Нейросетевые технологии принятия решений» разработано в соответствии с действующей учебной программой. Предназначено для оказания методической помощи в изучении данного курса в рамках магистерской подготовки специальности 1-98 80 03.

УДК 004.032.26 (076.6)
ББК 32.813 я73

ISBN 978-985-488-635-0

Содержание

1. Основы теории принятия решений **4**
1.1. Общие положения 4
1.2. Основные понятия системного анализа 5
1.3. Основные понятия исследования операций 8
1.4. Постановка задачи принятия оптимального решения 9
1.5. Принятие решений в условиях неопределённости 12

2. Искусственные нейронные сети **14**
2.1. Модели нейронов 14
2.2. Функции активации 15
2.3. Представление нейронных сетей с помощью направленных графов 17
2.4. Архитектура сетей 19
2.5. Представление знаний 22

3. Обучение нейронных сетей **27**
3.1. Обучение сети топологии многослойного персептрона 27
3.2. Сеть на основе слоя ячеек РБФ 29
3.3. Анализ главных компонентов 31
3.3.1. Матричный метод расчёта собственных векторов 32
3.3.2. Нейросетевой подход к определению главных компонентов 33
3.4. Слой Кохонена 34
3.4.1. Архитектура сети встречного распространения 34
3.4.2. Обучение слоя Кохонена в сетях встречного распространения . . 36
3.4.3. Достоинства, недостатки и проблемы обучения 41

1. Основы теории принятия решений

1.1. Общие положения

Практическая потребность общества в научных основах принятия решений возникла только в XVIII в. с развитием науки и техники. Основоположником науки «теория принятия решений» обычно считают Жозефа Луи Лагранжа.

Можно выделить следующие научно-технические предпосылки становления теории принятия решений:

— *удорожание «цены ошибки»*. Чем сложнее, дороже, масштабнее планируемое мероприятие, тем менее допустимы в нем «волевые» решения и тем важнее становятся научные методы, позволяющие заранее оценить последствия каждого решения, заранее исключить недопустимые варианты и рекомендовать наиболее удачные;

— *ускорение научно-технической революции техники и технологии*. Жизненный цикл технического изделия сократился настолько, что «опыт» не успевал накапливаться и требовалось применение более развитого математического аппарата в проектировании;

— *развитие ЭВМ*. Размерность и сложность реальных инженерных задач не позволяли использовать аналитические метода.

Как это часто бывает, данная наука, с одной стороны, стала определённой ветвью других, более общих наук (теория систем, системный анализ, кибернетика и т. д.), а с другой — стала синтезом фундаментальных, частных наук (исследование операций, оптимизация и т. д.), создав при этом и собственную методологию.

Инженерное дело теснейшим образом связано с совокупностями объектов, которые принято называть *сложными системами*, которые характеризуются многочисленными и разнообразными по типу связями между отдельно существующими элементами системы, а также наличием у системы функции назначения, которой нет у составляющих ее частей. На первый взгляд каждая сложная система имеет уникальную организацию. Однако более детальное изучение способно выделить общее в системе команд ЭВМ, в процессах проектирования лесной машины, самолета и космического корабля.

В научно-технической литературе существует ряд терминов, имеющих отношение к исследованию сложных систем.

Наиболее общий термин «*теория систем*» относится ко всевозможным аспектам исследования систем. Её основными частями являются:

— *системный анализ*, который понимается как исследование проблемы принятия решения в сложной системе;

— *кибернетика*, которая рассматривается как наука об управлении и преобразовании информации.

Следует заметить, что понятие *управления* не совпадает с *принятием решения*. Условная граница между кибернетикой и системным анализом состоит в том, что первая изучает отдельные и строго формализованные процессы, а системный анализ — совокупность процессов и процедур.

Очень близкое к термину «системный анализ» понятие «*исследование операций*», которое традиционно обозначает математическую дисциплину, охватывающую исследование математических моделей для выбора величин, оптимизирующих заданную математическую конструкцию (критерий). Системный анализ может сводиться к решению ряда задач исследования операций, но обладает свойствами, не охватываемыми этой дисциплиной. Однако в зарубежной литературе термин «исследование операций» не является чисто математическим и приближается к термину «системный анализ». Широкая опора системного анализа на исследование операций приводит к таким его математизированным разделам, как

— постановка задач принятия решения;
— описание множества альтернатив;
— исследование многокритериальных задач;
— методы решения задач оптимизации;
— обработка экспертных оценок;
— работа с макромоделями системы.

1.2. Основные понятия системного анализа

Системный анализ — наука, занимающаяся проблемой принятия решения в условиях анализа большого количества информации различной природы.

Из определения следует, что целью применения системного анализа к конкретной проблеме является повышение степени обоснованности принимаемого решения, расширение множества вариантов, среди которых производится выбор, с одновременным указанием способов отбрасывания, заведомо уступающим другим.

В системном анализе выделяют:
— методологию;
— аппаратную реализацию;
— практические приложения.

Методология включает в себя **определения** используемых понятий и **принципы системного подхода**.

Ниже приводятся *определения* системного анализа.

Элемент — некоторый объект (материальный, энергетический, информационный), который обладает рядом важных свойств, но внутреннее строение (содержание) которого безотносительно к цели рассмотрения.

Связь — важный для целей рассмотрения обмен между элементами веществом, энергией, информацией.

Система — совокупность элементов, которая обладает следующими признаками:

— связями, которые позволяют посредством переходов по ним от элемента к элементу соединить два любых элемента совокупности;

— свойствами, отличными от свойств отдельных элементов совокупности.

Практически любой объект с определённой точки зрения может быть рассмотрен как система. Вопрос состоит в том, насколько целесообразна такая точка зрения.

Большая система — система, которая включает значительное число однотипных элементов и однотипных связей. В качестве примера можно привести трубопровод. Элементами последнего будут участки между швами или опорами. Для расчетов на прочность по методу конечных элементов элементами системы считаются небольшие участки трубы, а связь имеет силовой (энергетический) характер — каждый элемент действует на соседние.

Сложная система — система, которая состоит из элементов разных типов и обладает разнородными связями между ними. В качестве примера можно привести ЭВМ, лесной трактор или судно.

Автоматизированная система — сложная система с определяющей ролью элементов двух типов:

— в виде технических средств;

— в виде действия человека.

Для сложной системы автоматизированный режим считается более предпочтительным, чем автоматический. Например, посадка самолета или захват дерева харвестерной головкой выполняется при участии человека, а автопилот или бортовой компьютер используется лишь при относительно простых операциях. Типична также ситуация, когда решение, выработанное техническими средствами, утверждается к исполнению человеком.

Структура системы — расчленение системы на группы элементов с указанием связей между ними, неизменное на все время рассмотрения и дающее представление о системе в целом. Указанное расчленение может иметь материальную, функциональную, алгоритмическую или другую основу. Пример материальной структуры — структурная схема сборного моста, которая состоит из отдельных, собираемых на месте секций и указывает только эти секции и порядок их соединения. Пример функциональной структуры — деление двигателя внутреннего сгорания на системы питания, смазки, охлаждения, передачи крутящего момента. Пример алгоритмической структуры — алгоритм программного средства, указывающего последовательность действий, или инструкция, которая определяет действия при отыскании неисправности технического устройства.

Структура системы может быть охарактеризована по имеющимся в ней типам связей. Простейшими из них являются последовательное, параллельное соединение и обратная связь.

Декомпозиция — деление системы на части, удобное для каких-либо операций с этой системой. Примерами будут: разделение объекта на отдельно проектируемые части, зоны обслуживания; рассмотрение физического явления или математическое описание отдельно для данной части системы.

Иерархия — структура с наличием подчиненности, т. е. неравноправных связей между элементами, когда воздействие в одном из направлений оказывает гораздо большее влияние на элемент, чем в другом. Виды иерархических структур разнообразны, но важных для практики иерархических структур всего две — древовидная и ромбовидная.

Древовидная структура наиболее проста для анализа и реализации. Кроме того, в ней всегда удобно выделять иерархические уровни — группы элементов, находящиеся на одинаковом удалении от верхнего элемента. Пример древовидной структуры — задача проектирования технического объекта от его основных характеристик (верхний уровень) через проектирование основных частей, функциональных систем, групп агрегатов, механизмов до уровня отдельных деталей.

Принципы системного подхода — положения общего характера, являющиеся обобщением опыта работы человека со сложными системами. Их часто считают ядром методологии. Известно около двух десятков таких принципов, ряд из которых целесообразно рассмотреть:

- принцип конечной цели: абсолютный приоритет конечной цели;
- принцип единства: совместное рассмотрение системы как целого и как совокупности элементов;
- принцип связности: рассмотрение любой части совместно с её связями с окружением;
- принцип модульного построения: полезно выделение модулей в системе и рассмотрение её как совокупности модулей;
- принцип иерархии: полезно введение иерархии элементов и(или) их ранжирование;
- принцип функциональности: совместное рассмотрение структуры и функции с приоритетом функции над структурой;
- принцип развития: учёт изменяемости системы, её способности к развитию, расширению, замене частей, накапливанию информации;
- принцип децентрализации: сочетание в принимаемых решениях и управлении централизации и децентрализации;
- принцип неопределённости: учёт неопределённостей и случайностей в системе.

Аппаратная реализация включает стандартные приёмы моделирования принятия решения в сложной системе и общие способы работы с этими моделями. Модель строится в виде связных множеств отдельных процедур. Системный анализ исследует как организацию таких множеств, так и вид

отдельных процедур, которые максимально приспосабливают для принятия согласующихся и управленческих решений в сложной системе.

Модель принятия решения чаще всего изображается в виде схемы с ячейками, связями между ячейками и логическими переходами. Ячейки содержат конкретные действия — процедуры. Совместное изучение процедур и их организации вытекает из того, что без учёта содержания и особенностей ячеек создание схем оказывается невозможным. Эти схемы определяют стратегию принятия решения в сложной системе. Именно с проработки связанного множества основных процедур принято начинать решение конкретной прикладной задачи.

Отдельные же процедуры (операции) принято классифицировать на *формализуемые* и *неформализуемые*. В отличие от большинства научных дисциплин, стремящихся к формализации, системный анализ допускает, что в определённых ситуациях неформализуемые решения, принимаемые человеком, являются более предпочтительными. Следовательно, системный анализ рассматривает в совокупности формализуемые и неформализуемые процедуры, и одной из его задач является определение их оптимального соотношения.

Формализуемые стороны отдельных операций лежат в области прикладной математики и использования ЭВМ. В ряде случаев математическими методами исследуется связное множество процедур и производится само моделирование принятие решения. Все это позволяет говорить о математической основе системного анализа. Такие области прикладной математики, как исследование операций и системное программирование, наиболее близки к системной постановке вопросов.

Практическое приложение системного анализа чрезвычайно обширно по содержанию. Важнейшими являются научно-технические разработки и различные задачи экономики. Ссылки на системность исследований, анализа, подхода включают биологию, экологию, военное дело, психологию, социологию, медицину, управление государством и регионом, лесное и сельское хозяйство, обучение и многое другое.

1.3. Основные понятия исследования операций

Операцией называется всякое мероприятие (система действий), объединённое единым замыслом и направленное к достижению какой-то цели.

Цель исследования операций — предварительное количественное обоснование оптимальных решений.

Всякий определённый выбор зависящих от нас параметров называется *решением*. *Оптимальными* называются *решения*, по тем или другим признакам предпочтительные перед другими.

Параметры, совокупность которых образует решение, называются *элементами решения*.

Множеством допустимых решений называются заданные условия, которые фиксированы и не могут быть нарушены.

Показатель эффективности — количественная мера, позволяющая сравнивать разные решения по эффективности.

Все решения принимаются всегда на основе информации, которой располагает *лицо, принимающее решение* (ЛПР).

Каждая задача в своей постановке должна отражать структуру и динамику знаний ЛПР о множестве допустимых решений и о показателе эффективности.

Задача называется *статической*, если принятие решения происходит в заранее известном и не изменяющемся информационном состоянии. Если информационные состояния в ходе принятия решения сменяют друг друга, то задача называется *динамической*.

Информационные состояния ЛПР могут по-разному характеризовать его физическое состояние:

— если информационное состояние состоит из единственного физического состояния, то задача называется *определённой*;
— если информационное состояние содержит несколько физических состояний и ЛПР кроме их множества знает ещё и вероятности каждого из этих физических состояний, то задача называется *стохастической* (частично неопределённой);
— если информационное состояние содержит несколько физических состояний, но ЛПР кроме их множества ничего не знает о вероятности каждого из этих физических состояний, то задача называется *неопределённой*.

1.4. Постановка задачи принятия оптимального решения

Несмотря на то, что методы принятия решений отличаются универсальностью, их успешное применение в значительной мере зависит от профессиональной подготовки специалиста, который должен иметь чёткое представление о специфических особенностях изучаемой системы и уметь корректно поставить задачу. Искусство постановки задач постигается на примерах успешно реализованных разработок и основывается на чётком представлении о преимуществах, недостатках и специфике различных методов оптимизации. В первом приближении можно сформулировать следующую последовательность действий, которые составляют содержание процесса постановки задачи:

— *установление границы подлежащей оптимизации системы*, т. е. представление системы в виде некоторой изолированной части реального мира. Расширение границ системы повышает размерность и сложность многокомпонентной системы и тем самым затрудняет её анализ. Следовательно, в инженерной практике следует к декомпозиции сложных

систем на подсистемы, которые можно изучать по отдельности без излишнего упрощения реальной ситуации;

— *определение показателя эффективности*, на основе которого можно оценить характеристики системы или её проекта с тем, чтобы выявить «наилучший» проект или множество «наилучших» условий функционирования системы. В инженерных приложениях обычно выбираются показатели экономического (издержки, прибыль и т. д.) или технологического (производительность, энергоёмкость, материалоёмкость и т. д.) характера. «Наилучшему» варианту всегда соответствует экстремальное значение показателя эффективности функционирования системы;

— *выбор внутрисистемных независимых переменных*, которые должны адекватно описывать допустимые проекты или условия функционирования системы и способствовать тому, чтобы все важнейшие технико-экономические решения нашли отражение в формулировке задачи;

— *построение модели*, которая описывает взаимосвязи между переменными задачи и отражает влияние независимых переменных на значение показателя эффективности. В самом общем случае структура модели включает основные уравнения материальных и энергетических балансов, соотношения, связанные с проектными решениями, уравнения, описывающие физические процессы, протекающие в системе, неравенства, которые определяют область допустимых значений независимых переменных и устанавливают лимиты имеющихся ресурсов. Элементы модели содержат всю информацию, которая обычно используется при расчете проекта или прогнозировании характеристик инженерной системы. Очевидно, процесс построения модели является весьма трудоемким и требует четкого понимания специфических особенностей рассматриваемой системы.

Несмотря на это, модели принятия оптимальных решений отличаются универсальностью, их успешное применение зависит от профессиональной подготовки инженера, который должен иметь полное представление о специфике изучаемой системы. Основная цель рассмотрения приводимых ниже примеров — продемонстрировать разнообразие постановок оптимизационных задач на основе общности их формы.

Все оптимизационные задачи имеют общую структуру. Их можно классифицировать как задачи минимизации (максимизации) M-векторного векторного показателя эффективности $W_m(x)$, $m = 1, 2, \ldots, M$, N-мерного векторного аргумента $x = (x_1, x_2, \ldots, x_N)$, компоненты которого удовлетворяют системе ограничений-равенств $h_k(x) = 0$, $k = 1, 2, \ldots, K$, ограничений-неравенств $g_j(x) > 0$, $j = 1, 2, \ldots, J$, областным ограничениям $x_{li} < x_i < x_{ui}$, $i = 1, 2, \ldots, N$.

Все задачи принятия оптимальных решений можно классифицировать в соответствии с видом функций $W_m(x)$, $h_k(x)$, $g_j(x)$ и размерностью и содержанием вектора x:

— *одноцелевое принятие решений* — $W_m(x)$ — скаляр;
— *многоцелевое принятие решений* — $W_m(x)$ — вектор;
— *принятие решений в условиях определённости* — исходные данные — детерминированные;
— *принятие решений в условиях неопределённости* — исходные данные — случайные.

Наиболее разработан и широко используется на практике аппарат одноцелевого принятия решений в условиях определённости, который получил название *математического программирования.*

Математический аппарат одноцелевого принятия решений в условиях неопределённости включает в себя стохастическое программирование (известны законы распределения случайных величин), теорию игр и теорию статистических решений (закон распределения случайных величин неизвестен).

Рассмотрим процесс принятия решений с самых общих позиций. Психологами установлено, что *решение* не является начальным процессом творческой деятельности. Оказывается, непосредственно акту решения предшествует тонкий и обширный процесс работы мозга, который формирует и предопределяет направленность решения. В этот этап, который можно назвать «предрешением» входят следующие элементы:

— *мотивация*, т. е. желание или необходимость что-то сделать. Мотивация определяет цель какого-либо действия, используя весь прошлый опыт, включая результаты;
— возможность неоднозначности результатов;
— возможность неоднозначности способов достижения результатов, т. е. свобода выбора.

После этого предварительного этапа следует, собственно, этап *принятия решения.* Но на нем процесс не заканчивается, т. к. обычно после принятия решения следует оценка результатов и корректировка действий. Таким образом, принятие решений следует воспринимать не как единовременный акт, а как последовательный процесс.

Выдвинутые выше положения носят достаточно общий характер, обычно подробно исследуемый психологами. Более близкой, с точки зрения инженера, будет следующая схема процесса принятия решения. Эта схема включает в себя следующие компоненты:

— анализ исходной ситуации;
— анализ возможностей выбора;
— выбор решения;
— оценка последствий решения и его корректировка.

1.5. Принятие решений в условиях неопределённости

Постановка задачи. В изложенном выше речь шла о постановках и методах решения задач, не содержащих неопределённостей. Однако, как правило, большинство реальных инженерных задач содержит в том или ином виде неопределённость. Как правило, решение задач с учетом разного вида неопределённостей является *общим*, а принятие решений без их учета — *частным*. Однако из-за концептуальных и методических трудностей в настоящее время не существует единого методологического подхода к решению таких задач. Тем не менее, накоплено достаточно большое число методов формализации постановки и принятия решений с учетом неопределённостей. При использовании этих методов следует иметь в виду, что все они носят рекомендательный характер и выбор окончательного решения всегда остается за человеком (ЛПР).

Как уже указывалось, при решении конкретных задач с учетом неопределённостей инженер сталкивается с разными их типами. В исследовании операций принято различать три типа неопределённостей:

- неопределённость целей;
- неопределённость наших знаний об окружающей обстановке и действующих в данном явлении факторах (неопределённость природы);
- неопределённость действий активного или пассивного партнера или противника.

В приведенной выше классификации тип неопределённостей рассматривается с позиций того или иного элемента математической модели. Так, например, неопределённость целей отражается при постановке задачи на выборе либо отдельных критериев, либо всего вектора полезного эффекта.

С другой стороны, два других типа неопределённостей в основном влияют на составление целевой функции уравнений ограничений и метода принятия решения. Конечно, приведенное выше утверждение является достаточно условным, как, впрочем, и любая классификация. Оно приводится лишь с целью выделить еще некоторые особенности неопределённостей, которые надо иметь в виду в процессе принятия решений.

Кроме рассмотренной выше классификации неопределённостей надо учитывать их тип (или «род») с точки зрения отношения к *случайности*.

По этому признаку можно различать *стохастическую* (вероятностную) неопределённость, когда неизвестные факторы *статистически устойчивы* и поэтому представляют собой обычные объекты теории вероятностей — случайные величины (или случайные функции, события и т. д.). При этом должны быть известны или определены при постановке задачи все необходимые статистические характеристики (законы распределения и их параметры).

Примером таких задач могут быть, в частности, система технического обслуживания и ремонта любого вида техники и т. д.

Другим крайним случаем может быть неопределённость *нестохастического* вида (по выражению Е. С. Вентцель — «дурная неопределённость»), при которой никаких предположений о стохастической устойчивости не существует. Наконец, можно говорить о промежуточном типе неопределённости, когда решение принимается на основании каких-либо гипотез о законах распределения случайных величин. При этом ЛПР должен иметь в виду опасность несовпадения его результатов с реальными условиями. Эта опасность несовпадения формализуется с помощью *коэффициентов риска*.

Принятие решений в условиях риска. Как указывалось выше, с точки зрения знаний об исходных данных в процессе принятия решений можно представить два крайних случая: определённость и неопределённость. В некоторых случаях неопределённость знаний является как бы «неполной» и дополняется некоторыми сведениями о действующих факторах, в частности, знанием законов распределения описывающих их случайных величин. Этот промежуточный случай соответствует ситуации *риска*. Принятие решений в условиях риска может быть основано на одном из следующих критериев:

— критерий ожидаемого значения;
— комбинации ожидаемого значения и дисперсии;
— известного предельного уровня;
— наиболее вероятного события в будущем.

2. Искусственные нейронные сети

2.1. Модели нейронов

Искусственный нейрон (или просто нейрон) является элементарным функциональным модулем, из множества которых строятся ИНС. Он представляет собой модель живого нейрона, однако лишь в смысле осуществляемых им преобразований, а не способа функционирования. Существуют логические, непрерывные и импульсные модели нейрона. Логические модели нейрона (в частности, описываемый картой Вена формальный нейрон) активно исследовались в 60 — 70-х годах, но не получили дальнейшего развития. Импульсные модели более близки к физической природе процессов, происходящих в нервной клетке, однако их теория не так развита как у непрерывных, и они все еще не находят широкого применения.

Непрерывная модель нейрона работает следующим образом. Входные сигналы поступают на блоки, реализующие функцию синапсов. Каждый из них характеризуется своим весовым коэффициентом (синаптичесим весом). Положительные значения весов w_{kj} соответствуют возбудительным синапсам, отрицательные — тормозным. Взвешенные входные сигналы подаются на линейный сумматор, после чего результат их сложения поступает на блок активационной функции. Обычно активационная функция ограничивает выходной сигнал нейрона в диапазоне $[0; 1]$ или $[-1; 1]$. Модель нейрона также включает в себя сдвиг b, который добавляется к входному сигналу блока активационной функции.

Математически модель нейрона описывается следующими зависимостями:

$$v_k = \sum_{j-1}^{l} w_{jk} z_j; \tag{2.1}$$

$$s_k = v_k + b_k; \tag{2.2}$$

$$y_k = \varphi(s_k). \tag{2.3}$$

Эту запись можно упростить, введя дополнительный входной сигнал $z_0 = 1$ и вес $w_{0k} = b_k$. Тогда (2.2) примет вид

$$s_k = \sum_{j=0}^{l} w_{jk} x_j. \tag{2.4}$$

В дальнейшем w_{0k} условно будет называться сдвигом. На рис. 2.1 изображена функциональная схема такой модели искусственного нейрона непрерывного типа.

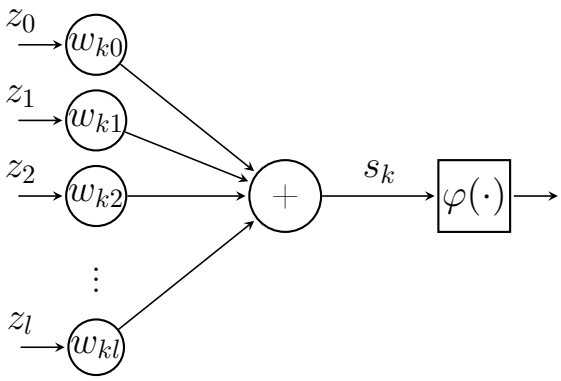

Рис. 2.1. Модель нейрона

2.2. Функции активации

Активационная функция нейрона $\varphi(s)$ определяет нелинейное преобразование, осуществляемое нейроном. Существует множество видов активационных функций, но более всего распространены следующие четыре:

1. Пороговая функция:

$$\varphi(s) = \begin{cases} 1, & \text{если } s \geqslant 0, \\ 0, & \text{если } s < 0. \end{cases} \tag{2.5}$$

Это — первая из введенных активационных функций, она была описана в работе Мак-Каллока и Питтса. В честь этих учёных модель нейрона с пороговой активационной функцией называется моделью Мак-Каллока — Питтса. Кроме того, в литературе непрерывную модель нейрона типа $\varphi(s)$, где s определяется по (2.4), обычно называют моделью Мак-Каллока — Питтса.

2. Кусочно-линейная функция. Она описывается следующей зависимостью:

$$\varphi(s) = \begin{cases} 1, & \text{если } s \geqslant a, \\ s, & \text{если } a > s > -a, \\ -1, & \text{если } s \leqslant -a. \end{cases} \tag{2.6}$$

В данном случае $a = 1$, т. е. коэффициент наклона линейного участка выбран единичным, а вся функция может интерпретироваться как аппроксимация нелинейного усилителя. При бесконечно большом коэффициенте наклона линейного участка функция вырождается в пороговую.

В большинстве типов ИНС используются нейроны с линейной активационной функцией $\varphi(s) = s$, представляющей собой частный случай (2.6) с неограниченным линейным участком.

3. Сигмоидальная функция. Наиболее широко используемый тип активационной функции. Она была введена по аналогии с пороговой функцией, но везде является строго монотонно возрастающей, непрерывной и дифферен-

цируемой. Дифференцируемость является важным свойством для анализа искусственных нейронных сетей и некоторых методов их обучения.

В общем виде сигмоидальная активационная функция описывается зависимостью

$$\varphi(s) = \frac{1}{1 + \mathrm{e}^{-as}}, \tag{2.7}$$

где a — параметр, определяющий наклон функции.

Варьированием его могут быть получены разные виды сигмоида. Наиболее часто используется $a = 1$. В случае бесконечно большого a сигмоидальная функция вырождается в пороговую.

Помимо перечисленных функций, изменяющихся в диапазоне $[0, 1]$, вводятся также их аналоги с областью значений $[-1, 1]$. Так, например, пороговая функция может быть переопределена следующим образом:

$$\varphi(s) = \begin{cases} 1, & \text{если } s > 0, \\ s, & \text{если } s = 0, \\ -1, & \text{если } s < 0. \end{cases} \tag{2.8}$$

Вместо сигмоидальной активационной функции широко применяется гиперболический тангенс, обладающий аналогичными свойствами:

$$\varphi(s) = \mathrm{th}(s) = \frac{e^{s} - e^{-s}}{e^{s} + e^{-s}}. \tag{2.9}$$

Нечетность этой функции делает ее удобной для решения задач управления.

4. Во введенных Брумхедом и Лоуе РБФ-сетях в качестве активационной применяется функция Гаусса:

$$\varphi(s) = \mathrm{e}^{-\frac{s^2}{\sigma^2}}. \tag{2.10}$$

Однако в отличие от (2.4) ее аргумент рассчитывается по формуле

$$s = \|\mathbf{z} - \mathbf{c}\|, \tag{2.11}$$

где $\mathbf{z}$ — вектор входных сигналов нейрона,
$\mathbf{c}$ — вектор координат центра окна активационной функции,
σ — ширина окна,
норма $\|\cdot\|$ — евклидово расстояние.

В теории ИНС активационные функции типа

$$\varphi(\mathbf{z}) = \varphi(\|\mathbf{z} - \mathbf{c}\|) \tag{2.12}$$

называются радиально-базисными функциями (РБФ), а основанные на них сети — РБФ-сетями (от англ. RBF — radial basis function).

2.3. Представление нейронных сетей с помощью направленных графов

Блочные диаграммы, представленные на рис. 2.1, обеспечивают функциональное описание различных элементов, из которых состоит модель искусственного нейрона. Внешний вид модели можно в значительной мере упростить, применив идею графов прохождения сигнала. Графы передачи сигнала с наборами правил были введены Мейсоном (Mason) в 1953 году для описания линейных сетей. Поэтому наличие нелинейности в модели нейрона ограничивает область применения этой парадигмы в нейронных сетях. Тем не менее графы прохождения сигнала дают хорошее представление о передаче сигнала по нейронным сетям. Именно этот вопрос и будет рассматриваться далее.

Граф передачи (или прохождения) сигнала (signal flow graph) представляет собой сеть направленных связей (links) (или ветвей (branches), соединяющих отдельные точки (узлы). С каждым узлом j связан сигнал X_j. Обычная направленная связь начинается в некотором узле j и заканчивается в другом узле k. С ней связана некоторая передаточная функция (transfer function), определяющая зависимость сигнала Y_k узла k от сигнала $X_j>$ узла j. Прохождение сигнала по различным частям графа подчиняется трем основным правилам.

Правило 1. Направление прохождения сигнала вдоль каждой связи определяется направлением стрелки. При этом можно выделить два типа связей:

1. Синаптические связи (synaptic link). Их поведение определяется линейным (linear) соотношением вход — выход. А именно, как показано на рис. 2.2, а, сигнал узла X_j умножается на синаптический вес W_{kj}, в результате чего получается сигнал узла Y_k.
2. Активационные связи (activation link). Их поведение определяется нелинейным (nonlinear) соотношением вход — выход. Этот вид связи показан на рис. 2.2, б, где $\varphi(s)$ — нелинейная функция активации.

Правило 2. Сигнал узла равен алгебраической сумме сигналов, поступающих на его вход. На рис. 2.2, в это правило проиллюстрировано для случая синаптической сходимости (synaptic convergence).

Правило 3. Сигнал данного узла передается по каждой исходящей связи без учёта передаточных функций исходящих связей. Это правило проиллюстрировано на рис. 2.2, г для синаптической дивергенции (synaptic divergence) или расходимости.

На рис. 2.3 показан пример графа передачи сигнала. Это модель нейрона, соответствующая блочной диаграмме, приведенной на рис. 2.2. По своему внешнему виду граф на рис. 2.3 значительно проще диаграммы на рис. 2.2, хотя и содержит все функциональные детали. Обратите внимание, что на

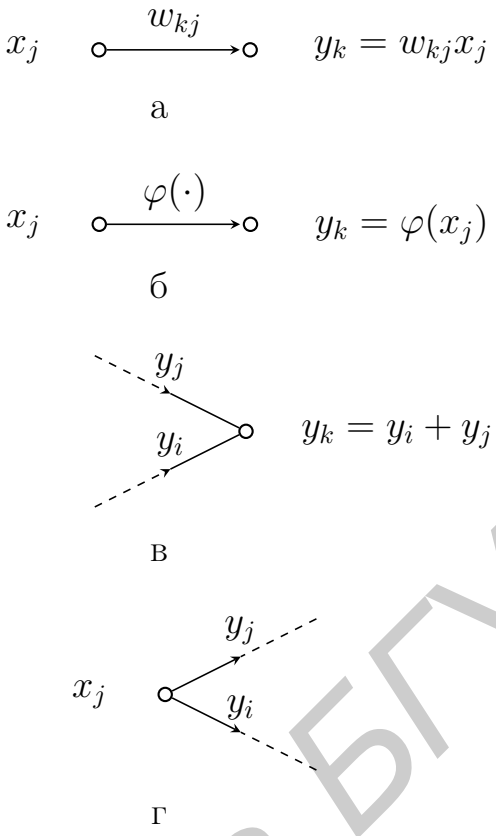

Рис. 2.2. Основные правила построения графов передачи сигнала

обоих рисунках входной сигнал принимает значение $x_0 = +1$, а соответствующий синаптический вес $w_{k0} = b_k$, где b_k — порог нейрона k.

Принимая в качестве модели нейрона граф прохождения сигнала, показанный на рис. 2.3, можно сформулировать еще одно определение нейронной сети.

Нейронная сеть — направленный граф, состоящий из узлов, соединенных синаптическими и активационными связями, который характеризуется следующими четырьмя свойствами:

1. Каждый нейрон представляется множеством линейных синаптических связей, внешним порогом и, возможно, нелинейной связью активации. Порог, представляемый входной синаптической связью, считается равным $+1$.
2. Синаптические связи нейрона используются для взвешивания соответствующих входных сигналов.
3. Взвешенная сумма входных сигналов определяет индуцированное локальное поле каждого конкретного нейрона.
4. Активационные связи модифицируют индуцированное локальное поле нейрона, создавая выходной сигнал.

Направленный граф, определённый указанным выше способом, является полным (complete). Это значит, что он описывает не только прохождение

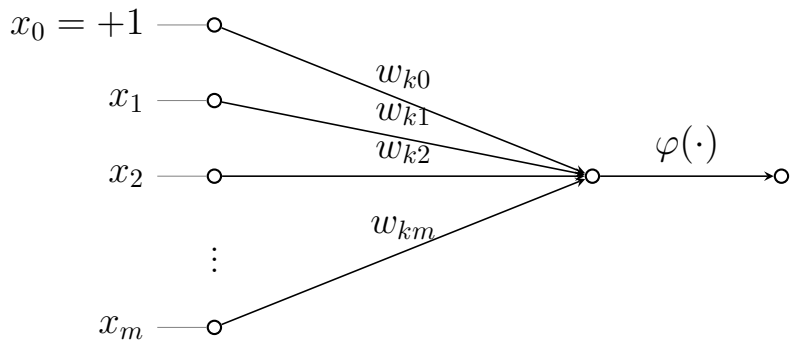

Рис. 2.3. Граф передачи сигнала для одного нейрона

сигнала между нейронами, но и передачу сигнала в самом нейроне. Если необходимо описать только прохождение сигнала между нейронами, то можно использовать сокращенную форму этого графа, опускающую детали передачи сигнала внутри нейронов. Такой направленный граф называется частично полным (partially complete) и характеризуется следующими свойствами.

1. Входные сигналы графа формируются узлами источника (source node) или входными элементами.
2. Каждый нейрон представляется одним узлом, который называется вычислительным (computation node).
3. Линии передачи сигнала (communication links), соединяющие узлы-источники и вычислительные узлы графа, не имеют веса. Они просто определяют направление прохождения сигнала на графе.

Частично полный направленный граф, определённый указанным выше способом, называется архитектурным (architectural graph). Он описывает структуру нейронной сети.

Подводя итог сказанному, можно выделить три графических представления нейронных сетей:

— блочная диаграмма, описывающая функции нейронной сети;
— граф прохождения сигнала, обеспечивающий полное описание передачи сигнала по нейронной сети;
— архитектурный граф, описывающий структуру нейронной сети.

2.4. Архитектура сетей

Структура нейронных сетей тесно связана с используемыми алгоритмами обучения. В общем случае можно выделить три фундаментальных класса нейросетевых архитектур.

Однослойные сети прямого распространения. В многослойной нейронной сети нейроны располагаются по слоям. В простейшем случае в такой сети существует входной слой (input layer) узлов источника, информация от которого передается на выходной слой (output layer) нейронов (вычисли-

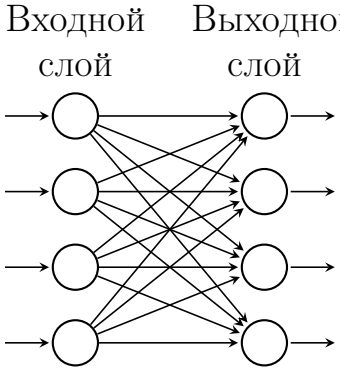

Рис. 2.4. Сеть прямого распространения с одним слоем нейронов

тельные узлы), но не наоборот. Такая сеть называется сетью прямого распространения (feed forward), или ацикличной сетью (acyclic). На рис. 2.4 показана структура такой сети для случая четырех узлов в каждом из слоёв (входном и выходном). Такая нейронная сеть называется однослойной (single layer network), при этом под единственным слоем подразумевается слой вычислительных элементов (нейронов). При подсчете числа слоёв не принимаются во внимание узлы источника, так как они не выполняют никаких вычислений.

Многослойные сети прямого распространения. Другой класс нейронных сетей прямого распространения характеризуется наличием одного или нескольких скрытых слоев (hidden layer), узлы которых называются скрытыми нейронами (hidden neuron), или скрытыми элементами (hidden unit). Функция последних заключается в посредничестве между внешним входным сигналом и выходом нейронной сети. Добавляя один или несколько скрытых слоев, можно выделить статистики высокого порядка. Такая сеть позволяет выделять глобальные свойства данных с помощью локальных соединений за счет наличия дополнительных синаптических связей и повышения уровня взаимодействия нейронов. Способность скрытых нейронов выделять статистические зависимости высокого порядка особенно существенна, когда размер входного слоя достаточно велик.

Узлы источника входного слоя сети формируют соответствующие элементы шаблона активации (входной вектор), которые составляют входной сигнал, поступающий на нейроны (вычислительные элементы) второго слоя (т. е. первого скрытого слоя). Выходные сигналы второго слоя используются в качестве входных для третьего слоя и т. д. Обычно нейроны каждого из слоев сети используют в качестве входных сигналов выходные сигналы нейронов только предыдущего слоя. Набор выходных сигналов нейронов выходного (последнего) слоя сети определяет общий отклик сети на данный входной образ, сформированный узлами источника входного (первого) слоя.

Нейронная сеть, показанная на рис. 2.5, считается полносвязной (fully connected) в том смысле, что все узлы каждого конкретного слоя соединены

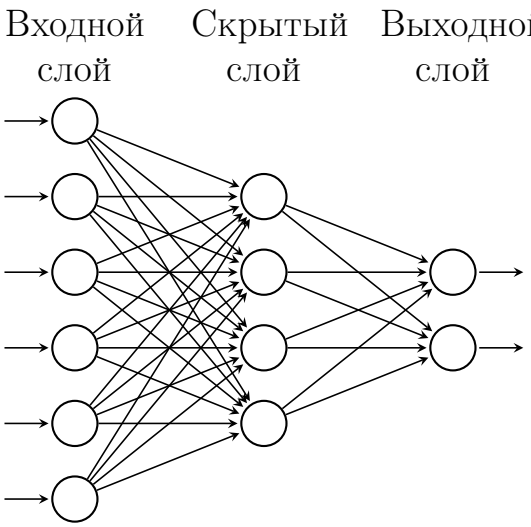

Рис. 2.5. Полносвязная сеть прямого распространения с одним скрытым и одним выходным слоем нейронов

со всеми узлами смежных слоев. Если некоторые из синаптических связей отсутствуют, такая сеть называется неполносвязной (partially connected).

Рекуррентные сети. Рекуррентная нейронная сеть (recurrent network) отличается от сети прямого распространения наличием по крайней мере одной обратной связи (feedback loop). Например, рекуррентная сеть может состоять из единственного слоя нейронов, каждый из которых направляет свой выходной сигнал на входы всех остальных нейронов слоя.

Архитектура такой нейронной сети показана на рис. 2.6. Обратите внимание, что в приведенной структуре отсутствуют обратные связи нейронов с самими собой. Рекуррентная сеть, показанная на 2.6, не имеет скрытых нейронов. На рис. 2.7 показан другой класс рекуррентных сетей со скрытыми нейронами. Здесь обратные связи исходят как из скрытых, так и из выходных нейронов.

Наличие обратных связей в сетях, показанных на рис. 2.6 и 2.7, оказывает непосредственное влияние на способность таких сетей к обучению и на их производительность. Более того, обратная связь подразумевает использование элементов единичной задержки (unit delay element) (они обозначены как z^{-1}), что приводит к нелинейному динамическому поведению, если, конечно, в сети содержатся нелинейные нейроны.

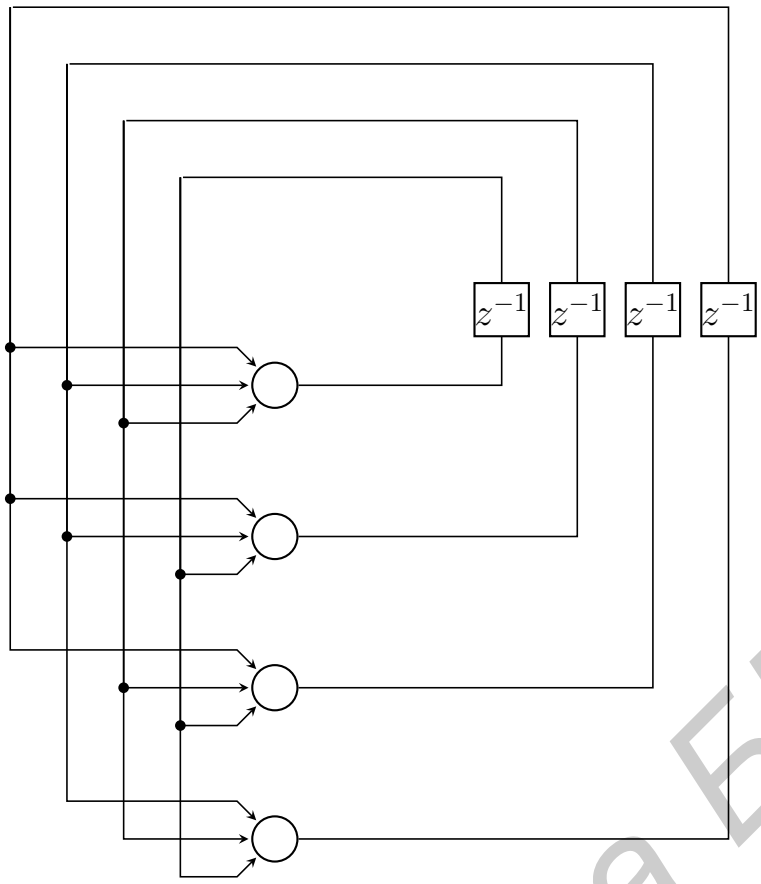

Рис. 2.6. Рекуррентная сеть без скрытых нейронов и обратных связей нейронов с самими собой

2.5. Представление знаний

Ранее при определении нейронных сетей часто использовался термин «знания» без явного описания его значения. Этот пробел можно устранить, предложив следующее общее определение.

Под знаниями понимается хранимая информация или модели, используемые человеком или машиной для интерпретации, предсказания и реакции на внешние события.

К вопросам представления знаний (knowledge representation) относятся следующие: какую информацию необходимо хранить и как эту информацию представить физически для ее последующего использования. Таким образом, исходя из самой природы знаний, способ их представления определяется поставленной целью. Относительно реальных приложений «интеллектуальных» систем можно утверждать, что успех решения зависит от хорошего представления знаний. Это касается и нейронных сетей, представляющих собой отдельный класс интеллектуальных систем. Форма представления входных сигналов может быть самой разной. Это приводит к тому, что разработка приемлемых нейросетевых решений становится творческим процессом.

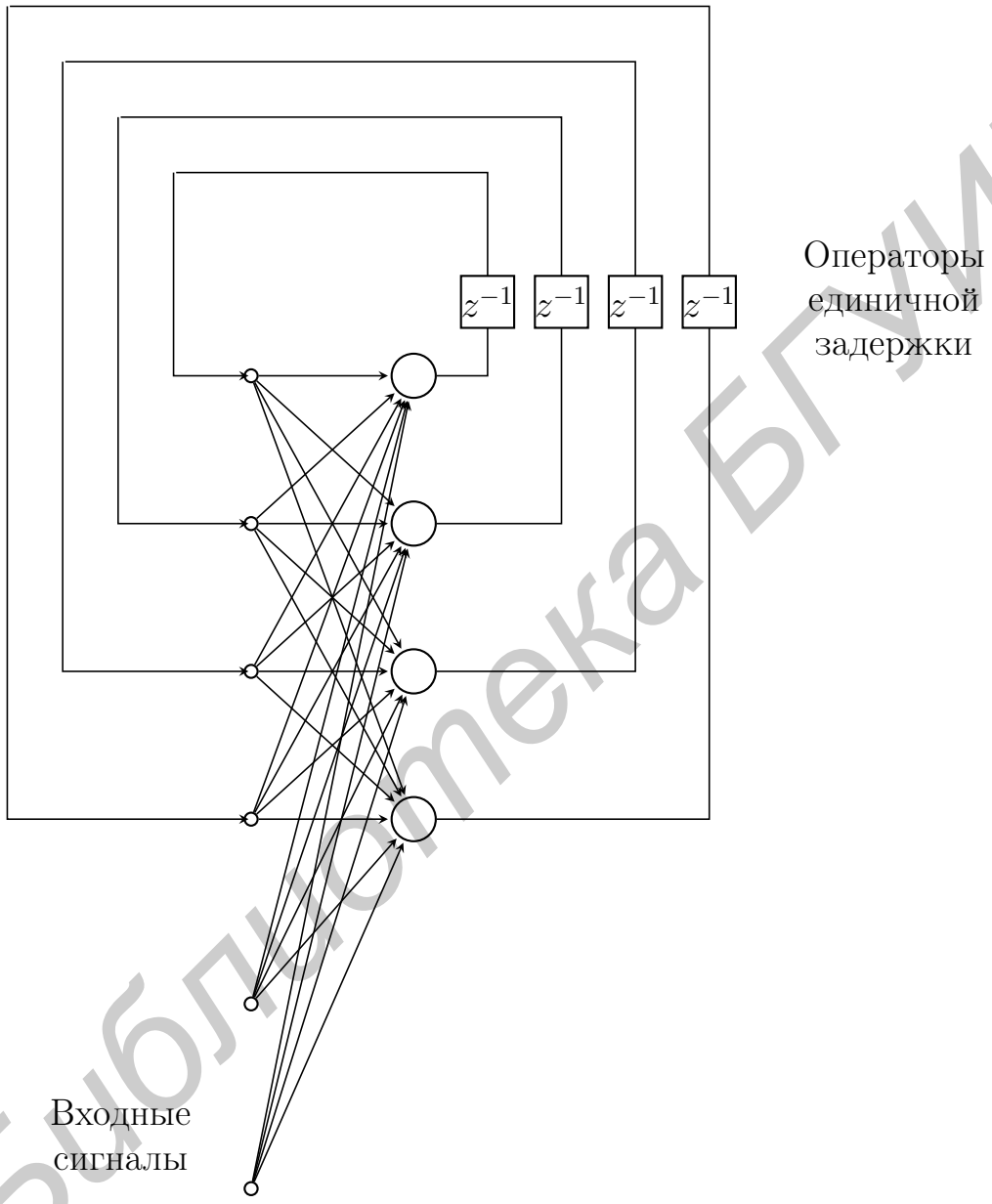

Рис. 2.7. Рекуррентная сеть со скрытыми нейронами

Основной задачей нейронной сети является наилучшее обучение модели окружающего мира для решения поставленной задачи. Знания о мире включают два типа информации:

1. Известное состояние окружающего мира, представленное имеющимися в наличии достоверными фактами. Такая информация называется априорной (prior).
2. Наблюдения за окружающим миром (измерения), полученные с помощью сенсоров, адаптированных для конкретных условий, в которых должна функционировать данная нейронная сеть. Обычно такие измерения в значительной мере зашумлены, что потенциально может стать источником ошибок. В любом случае измерения, полученные таким способом, формируют множество информации, примеры из которого используются для обучения нейронной сети.

Примеры могут быть маркированными (labeled) и немаркированными (unlabeled). В маркированных примерах входному сигналу (input signal) соответствует желаемый отклик (desired response). Немаркированные примеры состоят из нескольких различных реализаций одного входного сигнала. В любом случае набор примеров, будь то маркированных или нет, представляет собой знания об интересующей предметной области, на основании которых и проводится обучение нейронной сети.

Множество пар сигналов вход – выход, каждая из которых состоит из входного сигнала и соответствующего ему желаемого выхода, называют обучающими данными (training data), или обучающей выборкой (training sample). Для примера рассмотрим задачу распознавания цифр (digit recognition problem). В этой задаче входной сигнал (изображение) представляет собой матрицу, состоящую из черных и белых точек. Каждое изображение представляет одну из десяти рукописных цифр на белом фоне. Желаемым откликом сети является конкретная цифра, изображение которой подается в качестве входного сигнала. Обычно обучающая выборка состоит из большого числа рукописных цифр, что отражает ситуацию, которая может возникнуть в реальном мире. При наличии такого набора примеров нейронная сеть создается следующим образом.

Во-первых, выбирается соответствующая нейросетевая архитектура, в которой размер входного слоя соответствует количеству пикселов на рисунке, а в выходном слое содержится десять нейронов, соответствующих цифрам. После этого выполняется настройка весовых коэффициентов сети на основе обучающего множества. Этот режим работы сети называется обучением.

Во-вторых, эффективность обучения сети проверяется (тестируется) на множестве примеров, отличных от использованных при обучении. При этом на вход сети подается изображение, для которого известен целевой выход сети. Эффективность обучения сети проверяется путем сравнения результа-

тов распознавания с реальными цифрами. Этот этап работы нейронной сети называют обобщением (generalization) (данный термин взят из психологии).

Здесь и кроется фундаментальное различие между созданием нейронной сети и разработкой классических методов обработки информации для задач классификации. В последнем случае мы прежде всего формулируем математическую модель исследуемой среды, верифицируем ее на реальных данных, а затем разрабатываем классификатор на основе этой модели. Создание нейронной сети основывается непосредственно на реальных данных, которые говорят сами за себя. Таким образом, нейронные сети не только реализуют полноценную модель среды, но и обеспечивают обработку данных.

Набор данных, используемый для обучения сети, должен содержать как положительные, так и отрицательные примеры. Например, в задаче пассивной эхолокации положительные примеры включают сигналы, отраженные от интересующего объекта (например подводной лодки). Однако в реальной среде на отклик радара влияют и морские объекты, случайно попавшие в зону сигнала. Чтобы понизить вероятность неверной трактовки сигнала, в множество примеров добавляют сигналы, полученные при отсутствии искомого объекта.

В нейронной сети заданной архитектуры знания об окружающей среде представляются множеством свободных параметров (т. е. синаптических весов и порогов) сети. Такая форма представления знаний соответствует самой природе нейронных сетей. Именно в ней кроется ключ к эффективности нейросетевых моделей.

Вопрос представления знаний в нейронной сети является очень сложным. Тем не менее можно выделить четыре общих правила.

Правило 1. Сходные входные сигналы от схожих классов должны формировать единое представление в нейронной сети. Исходя из этого, они должны быть классифицированы как принадлежащие к одной категории.

Правило 2. Элементы, отнесенные к различным классам, должны иметь в сети как можно более отличающиеся друг от друга представления. Это правило прямо противоположно первому.

Правило 3. Если некоторое свойство имеет важное значение, то для его представления в сети необходимо использовать большое количество нейронов.

Правило 4. В структуру нейронной сети должны быть встроены априорная информация и инварианты, что упрощает архитектуру сети и процесс ее обучения.

Это правило играет особую роль, поскольку правильная конфигурация сети обеспечивает ее специализацию (specialized structure), что очень важно по следующим причинам.

1. Биологические сети, обеспечивающие обработку зрительной и слуховой информации, сильно специализированы.

2. Нейронная сеть со специализированной структурой обычно включает значительно меньшее количество свободных параметров, которые нужно настраивать, чем полносвязная сеть. Из этого следует, что для обучения специализированной сети требуется меньше данных. При этом на обучение затрачивается меньше времени, и такая сеть обладает лучшей обобщающей способностью.
3. Специализированные сети обладают большей пропускной способностью.
4. Стоимость создания специализированных нейронных сетей сокращается, поскольку их размер существенно меньше размера полносвязных сетей.

3. Обучение нейронных сетей

3.1. Обучение сети топологии многослойного персептрона

Долгое время не существовало теоретически обоснованного алгоритма для обучения многослойных искусственных нейронных сетей. А так как возможности представления с помощью однослойных нейронных сетей оказались весьма ограниченными, то и вся область в целом пришла в упадок.

Разработка алгоритма обратного распространения сыграла важную роль в возрождении интереса к искусственным нейронным сетям. Обратное распространение — это систематический метод для обучения многослойных искусственных нейронных сетей. Он имеет солидное математическое обоснование. Несмотря на некоторые ограничения, процедура обратного распространения сильно расширила область проблем, в которых могут быть использованы искусственные нейронные сети, и убедительно продемонстрировала свою мощь.

Целью обучения сети является такая подстройка ее весов, чтобы приложение некоторого множества входов приводило к требуемому множеству выходов. Для краткости эти множества входов и выходов будут называться векторами. При обучении предполагается, что для каждого входного вектора существует парный ему целевой вектор, задающий требуемый выход. Вместе они называются обучающей парой. Как правило, сеть обучается на многих парах. Например, входная часть обучающей пары может состоять из набора нулей и единиц, представляющего двоичный образ некоторой буквы алфавита. При необходимости распознавать с помощью сети все буквы алфавита, потребовалось бы 26 обучающих пар. Такая группа обучающих пар называется обучающим множеством (обучающей выборкой).

Перед началом обучения всем весам должны быть присвоены небольшие начальные значения, выбранные случайным образом. Это гарантирует, что в сети не произойдет насыщения большими значениями весов, и предотвращает ряд других патологических случаев. Например, если всем весам придать одинаковые начальные значения, а для требуемого функционирования нужны неравные значения, то сеть не сможет обучиться.

В своей работе Нгуен и Видроу описали алгоритм инициализации весов многослойного персептрона, способствующий ускорению процесса обучения. Этот метод заключается в генерации начальных весов и смещений для слоёв так, чтобы активные области нейронов были распределены равномерно относительно области входов, что обеспечивает минимизацию числа нейронов сети и времени обучения.

Обучение сети обратного распространения требует выполнения следующих операций:

1. Выбрать очередную обучающую пару из обучающего множества; подать входной вектор на вход сети.
2. Вычислить выход сети.
3. Вычислить разность между выходом сети и требуемым выходом (целевым вектором обучающей пары).
4. Подкорректировать веса сети так, чтобы минимизировать ошибку.
5. Повторять шаги с 1 по 4 для каждого вектора обучающего множества до тех пор, пока ошибка на всем множестве не достигнет приемлемого уровня.

Операции, выполняемые шагами 1 и 2, сходны с теми, которые выполняются при функционировании уже обученной сети, т. е. подается входной вектор и вычисляется получающийся выход. Вычисления выполняются послойно.

После достаточного числа повторений этих четырех шагов разность между действительными выходами и целевыми выходами должна уменьшиться до приемлемой величины, при этом говорят, что сеть обучилась. Теперь сеть используется для распознавания, и веса не изменяются.

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

Так как для каждого нейрона выходного слоя задано целевое значение, то подстройка весов легко осуществляется с использованием (3.1). Внутренние слои называют «скрытыми слоями», для их выходов не имеется целевых значений для сравнения. Поэтому коррекция весов скрытых слоёв осуществляется с использованием (3.2).

$$w_{jk} = w_{jk} + \alpha\varphi_k(1-\varphi_k)d_k g_j, \tag{3.1}$$

$$v_{ij} = v_{ij} + \beta g_j(1-g_j)e_j x_i. \tag{3.2}$$

Веса от нейрона p в скрытом слое j к нейрону j в выходном слое k обучаются следующим образом. Выход нейрона слоя k, вычитаясь из целевого значения (φ), дает сигнал ошибки. Он умножается на производную сжимающей функции $\varphi(1-\varphi)$, вычисленную для этого нейрона слоя k, давая, таким образом, величину δ.

$$\delta = \varphi_k(1-\varphi_k)d_k. \tag{3.3}$$

Затем δ умножается на величину φ нейрона j, из которого выходит рассматриваемый вес. Это произведение в свою очередь умножается на коэффициент скорости обучения α (обычно от 0,01 до 1,0), и результат прибавляется к весу. Такая же процедура выполняется для каждого веса от нейрона скрытого слоя к нейрону в выходном слое.

Следующие уравнения иллюстрируют это вычисление:

$$\Delta w_{pq,k} = \alpha \delta_{q,k} \varphi_{p,j}, \tag{3.4}$$

$$w_{pq,k}(n+1) = w_{pq,k}(n) + \Delta w_{pq,k}, \tag{3.5}$$

где $w_{pq,k}(n)$ — величина веса от нейрона p в скрытом слое к нейрону q в выходном слое на шаге n (до коррекции); индекс k относится к слою, в котором заканчивается данный вес, т. е., согласно принятому соглашению, с которым он объединен; $w_{pq,k}(n+1)$ — величина веса на шаге $n+1$ (после коррекции); $\delta_{q,k}$ — величина δ для нейрона q, в выходном слое k; $\varphi_{p,j}$ — величина φ для нейрона p в скрытом слое j.

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

$$\delta_{q,k} = \varphi_{p,j}(1 - \varphi_{p,j}) \left[\sum_{q} \delta_{q,k} w_{pq,k} \right]. \tag{3.6}$$

Когда значение δ получено, веса, питающие первый скрытый уровень, могут быть подкорректированы с помощью уравнений (3.4) и (3.5), где индексы модифицируются в соответствии со слоем.

Для каждого нейрона в данном скрытом слое должно быть вычислено δ и подстроены все веса, ассоциированные с этим слоем. Этот процесс повторяется слой за слоем по направлению к входу, пока все веса не будут подкорректированы.

3.2. Сеть на основе слоя ячеек РБФ

Процесс обучения сети РБФ с учетом выбранного типа радиальной базисной функции сводится к следующему:

— подбору центров c_i и параметров σ_i формы базисных функций;

— подбору весов нейронов выходного слоя.

При этом проблема уточнения весов нейронов выходного слоя значительно упрощается.

Подбор синаптических весов нейронов выходного слоя проще всего осуществить с помощью *метода псевдообращения*:

$$\mathbf{w} = \mathbf{G}^{+}\mathbf{d} = (\mathbf{G}^{T}\mathbf{G})^{-1}\mathbf{G}^{T}, \tag{3.7}$$

где $\mathbf{d}$ — вектор желаемого отклика для множества примеров, а матрица $\mathbf{G}^+$ — псевдообратная матрице $\mathbf{G}$.

В соответствии с этим выражением, вектор весов $\mathbf{w}$ может быть определён за один шаг псевдоинверсией матрицы $\mathbf{G}$, $\mathbf{w} = \mathbf{G}^+\mathbf{d}$. Матрица $\mathbf{G}$, имеющая p строк и K столбцов, представляет реакции нейронов скрытого слоя на очередные возбуждения векторами $\mathbf{x}_i$ $(i = 1, 2, \ldots, p)$. Практически псевдоинверсия матрицы $\mathbf{G}$ рассчитывается с использованием разложения по собственным значениям, в соответствии с которым

$$\mathbf{G} = \mathbf{U}\mathbf{S}\mathbf{V}^T. \tag{3.8}$$

Матрицы $\mathbf{U}$ и $\mathbf{V}$ ортогональны и имеют размерности $p \times p$ и $K \times K$ соответственно, тогда как S — псевдодиагональная матрица размерностью $p \times K$. При этом $K < p$, а диагональные элементы $s_1 \geqslant s_2 \geqslant \cdots \geqslant s_k \geqslant 0$. Допустим, что только r первых элементов s_i имеют значимую величину, а остальными можно пренебречь. Тогда количество столбцов ортогональных матриц $\mathbf{U}$ и $\mathbf{V}$ может быть уменьшено до r.

Полученные таким образом редуцированные матрицы $\mathbf{U}_r$ и $\mathbf{V}_r$ имеют вид

$$\mathbf{U}_r = (\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_r), \tag{3.9}$$

$$\mathbf{V}_r = (\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_r), \tag{3.10}$$

а матрица $\mathbf{S}_r = \mathrm{diag}(s_1, s_2, \ldots, s_r)$ становится полностью диагональной (квадратной). Эту матрицу описывает зависимость (3.8) в форме

$$\mathbf{G} \simeq \mathbf{U}_r\mathbf{S}_r\mathbf{T}_r^T. \tag{3.11}$$

Псевдообратная к $\mathbf{G}$ матрица определяется в этом случае выражением

$$\mathbf{G}^+ = \mathbf{V}_r\mathbf{S}_r^{-1}\mathbf{U}_r^T, \tag{3.12}$$

в котором

$$\mathbf{S}_r^{-1} = \left(\frac{1}{s_1}, \frac{1}{s_2}, \ldots, \frac{1}{s_r}\right), \tag{3.13}$$

а вектор весов сети, подвергающейся обучению, задаётся формулой

$$\mathbf{w} = \mathbf{V}_r\mathbf{S}^{-1}\mathbf{U}_r^T\mathbf{d}. \tag{3.14}$$

Достоинство формулы (3.14) — её простота. Выходные веса сети подбираются за один шаг простым перемножением соответствующих матриц, при этом некоторые из них ($\mathbf{U}_r$, $\mathbf{V}_r$) ортогональны и по своей природе хорошо упорядочены (коэффициент порядка равен 1).

Принимая во внимание решение (3.14), определяющее значения весов выходного слоя, главной проблемой обучения радиальных сетей остается подбор параметров нелинейных радиальных функций, особенно центров c_i.

Одним из простейших, хотя и не самым эффективным, способом определения параметров базисных функций, считается случайный выбор. В этом решении центры c_i базисных функций выбираются случайным образом на основе равномерного распределения. Такой подход допустим применительно к классическим радиальным сетям при условии, что равномерное распределение обучающих данных хорошо соответствует специфике задачи. При выборе гауссовской формы радиальной функции задается значение стандартного отклонения s_i, зависящее от разброса выбранных случайным образом центров c_i:

$$\varphi(\|\mathbf{x}-\mathbf{c}_i\|^2 = \exp\left(\frac{K\cdot\|\mathbf{x}-\mathbf{c}_i\|^2}{d^2}\right) \tag{3.15}$$

для $i = 1, 2, \ldots, K$, где d обозначает максимальное расстояние между центрами c_i. Из выражения (3.15) следует, что стандартное отклонение гауссовской функции, характеризующее ширину кривой, устанавливается при случайном выборе, равным $\sigma = d/\sqrt{2K}$ и постоянно для всех базисных функций. Ширина функции пропорциональна максимальному разбросу центров и уменьшается с ростом их количества.

3.3. Анализ главных компонентов

Главной задачей в статистическом распознавании является выделение признаков (feature selection) и извлечение признаков (feature extraction). Под выделением признаков понимают процесс, в котором пространство данных (data space) преобразуется в пространство признаков (feature space), теоретически имеющее ту же размерность, что и исходное пространство. Однако обычно преобразование выполняется таким образом, чтобы пространство данных могло быть представлено сокращенным количеством «эффективных признаков». Таким образом, остаётся только существенная часть информации, содержащейся в данных. Другими словами, множество данных подвергается сокращению размерности (dimensionality reduction).

Анализ главных компонентов (PCA) — это статистический метод, определяющий линейное преобразование $\mathbf{y} = \mathbf{W}\mathbf{x}$. Оно трансформирует описание стационарного стохастического процесса, представленного вектором $\mathbf{x}$ размерностью N в вектор $\mathbf{y}$ размерностью K посредством матрицы $\mathbf{W}$ размерностью $\mathbf{R}^{K\times N}$ при $K < N$ таким образом, что выходное пространство редуцированного размера сохраняет наиболее важную информацию об исходном процессе. Другими словами, преобразование по методу PCA позволяет заменить большое количество информации, основанной на взаимно коррелирую-

щих входных данных, множеством статистически независимых компонентов с учётом их важности. Поэтому оно считается одной из форм компрессии с потерей информации, известной в теории связи как преобразование Карунена — Лоева (Karhunen — Loeve transformation).

Преобразование PCA позволяет определить корреляцию, возникающую между различными переменными, составляющими множество данных. Если эти переменные коррелируют между собой, то знание только части из них будет достаточно для определения остальных. Поэтому такое множество данных может представляться меньшим количеством переменных. В случае, когда переменные не коррелируются, восстановление части из них на основании других становится невозможным.

3.3.1. Матричный метод расчёта собственных векторов

Матричный метод поиска главных компонентов основан на поиске главных векторов и главных значений матрицы ковариации, рассчитанной для исходного набора данных.

Данный метод поиска предусматривает выполнение следующих стадий:

1. Сформировать множество данных для поиска главных компонентов. Примем размерность данных равной N, а количество фрагментов за K. Тогда исходным множеством данных будет являться матрица $\mathbf{D}$ имеющая N столбцов и K строк.
2. Усреднить значения матрицы по каждой размерности, т. е. выполнить следующее преобразование:

$$D_{ij}^{av} = D_{ij} - \frac{\sum_{n=1}^{k} D_{ni}}{K}, \tag{3.16}$$

где D_{ij}^{av} — усреднённый компонент исходной матрицы.

3. Рассчитать матрицу ковариации. Для пояснения понятия матрицы ковариации введём понятие ковариации для двух векторов. Данная мера общности для двух случайных последовательностей рассчитывается следующим образом:

$$\operatorname{cov}(\mathbf{X}, \mathbf{Y}) = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{n-1}, \tag{3.17}$$

где $\mathbf{X}$ и $\mathbf{Y}$ — исходные векторы, а n — их размерность.

Для исходной матрицы $\mathbf{D}$ матрица ковариации строится следующим образом:

$$\mathbf{C} = \begin{pmatrix} \operatorname{cov}(\mathbf{D}_1, \mathbf{D}_1) & \dots & \operatorname{cov}(\mathbf{D}_1, \mathbf{D}_N) \\ \vdots & \operatorname{cov}(\mathbf{D}_i, \mathbf{D}_i) & \vdots \\ \operatorname{cov}(\mathbf{D}_N, \mathbf{D}_1) & \dots & \operatorname{cov}(\mathbf{D}_N, \mathbf{D}_N) \end{pmatrix}, \tag{3.18}$$

где $\mathbf{D}_i$ — i-й столбец матрицы $\mathbf{D}$.

4. Расчёт главных значений и векторов для матрицы ковариации. Существует несколько методов разложения матрицы по собственным значениям. В систему были интегрированы методы декомпозиции Якоби и Хаусхолдера.
5. Результатом выполнения предыдущей стадии является вектор собственных значений матрицы и набор соответствующих им главных векторов. На данном этапе производится сортировка главных векторов в порядке убывания соответствующих им главных значений. Далее, в зависимости от целей эксперимента, за счёт отбрасывания некоторого количества главных векторов строится матрица преобразования исходного множества данных в пространство меньшей размерности.

3.3.2. Нейросетевой подход к определению главных компонентов

В развитии методов PCA важную роль играют хеббовские нейронные сети. Существует тесная взаимосвязь между поведением самоорганизующихся нейронных сетей и статистическим методом анализа главных компонентов. Один линейный нейрон с хеббовским правилом адаптации синаптических весов может быть преобразован в фильтр для выделения первого главного компонента входного распределения.

Для определения первого главного компонента y_1 и связанного с ним вектора $\mathbf{w}_1$, соответствующего матрице $\mathbf{D}$, было предложено использовать систему, состоящую из одного линейного нейрона. Веса его синаптических связей (т. е. значения вектора $\mathbf{w}_1$) подбираются согласно нормализованному правилу Хебба, называемому правилом Ойя:

$$w_i(n+1) = w_i(n) + \eta y(n)(x_i(n) - y(n)w_i(n)), \tag{3.19}$$

где $i = 1, 2, \ldots, m$, а η — коэффициент обучения. Подбор значения η оказывает существенное влияние на сходимость алгоритма. Хорошие результаты достигаются, когда значение η уменьшается с течением времени обучения.

Определение следующих компонентов PCA предполагает использование в выходном слое большего количества нейронов. В таком случае сеть должна содержать столько нейронов, сколько главных компонентов должно быть учтено. Для корректировки весов такой сети хорошие результаты даёт правило Сенгера. Уточнение весов сети производится по формуле

$$w_{ji}(n+1) = w_{ji}(n) + \eta y_j(n)x_i(n) - y_i(n)\sum_{k=1}^{j} w_{ki}(n)y_k(n), \tag{3.20}$$

где $i = 1, 2, \ldots, m$, $j = 1, 2, \ldots, l$.

Преобразование PCA чаще всего применяется для сжатия данных, при котором большое количество входной информации заменяется уменьшенной дозой, содержащейся в векторах $\mathbf{y}$ и $\mathbf{w}_i$. В зависимости от степени сжатия (количества главных компонентов) можно получить разное качество восстановления данных.

3.4. Слой Кохонена

Возможности сети встречного распространения превосходят возможности однослойных сетей. Время же обучения по сравнению с обратным распространением может уменьшаться на порядки. Встречное распространение не столь общее, как обратное распространение, но оно может давать решение в тех приложениях, где длительная обучающая процедура невозможна. Будет показано, что помимо преодоления ограничений других сетей встречное распространение обладает собственными интересными и полезными свойствами.

Во встречном распространении объединены два хорошо известных алгоритма: самоорганизующаяся карта Кохонена и звезда Гроссберга. Их объединение ведет к появлению свойств, которых нет ни у одного из них в отдельности.

Методы, которые подобно встречному распространению, объединяют различные сетевые парадигмы как строительные блоки, могут привести к сетям, более близким к мозгу по архитектуре, чем любые другие однородные структуры. Похоже, что в мозгу именно каскадные соединения модулей различной специализации позволяют выполнять требуемые вычисления.

Сеть встречного распространения функционирует подобно столу справок, способному к обобщению. В процессе обучения входные векторы ассоциируются с соответствующими выходными векторами. Эти векторы могут быть двоичными, состоящими из нулей и единиц, или непрерывными. Когда сеть обучена, приложение входного вектора приводит к требуемому выходному вектору. Обобщающая способность сети позволяет получать правильный выход даже при приложении входного вектора, который является неполным или слегка неверным. Это позволяет использовать данную сеть для распознавания образов, восстановления образов и усиления сигналов.

3.4.1. Архитектура сети встречного распространения

На рис. 3.1 показана упрощенная версия прямого действия сети встречного распространения. На нём иллюстрируются функциональные свойства этой парадигмы.

Нейроны слоя 0 служат лишь точками разветвления и не выполняют вычислений. Каждый нейрон слоя 0 соединен с каждым нейроном слоя 1 (называемого слоем Кохонена) отдельным весом w_{mn}. Эти веса в целом рассмат-

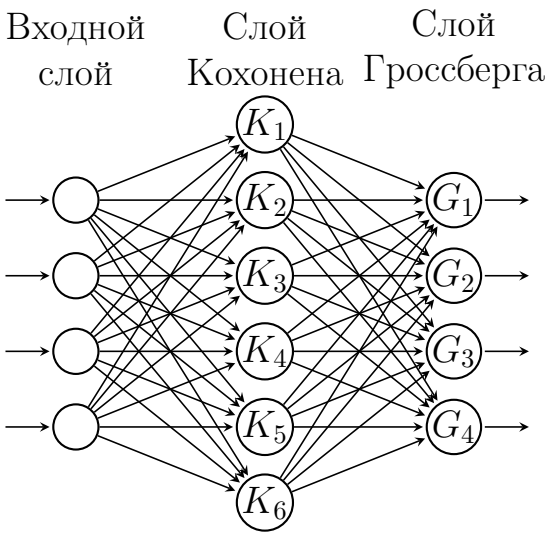

Рис. 3.1. Сеть с встречным распознаванием без обратных связей

риваются как матрица весов W. Каждый нейрон в слое Кохонена (слой 1) соединен с каждым нейроном в слое Гроссберга (слой 2) весом v_{np}. Эти веса образуют матрицу весов V.

Как и многие другие сети, встречное распространение функционирует в двух режимах: в нормальном режиме, при котором принимается входной вектор X и выдается выходной вектор Y, и в режиме обучения, при котором подается входной вектор и веса корректируются, чтобы дать требуемый выходной вектор.

В своей простейшей форме слой Кохонена функционирует по принципу «победитель забирает все», т. е. для данного входного вектора один и только один нейрон Кохонена выдает на выходе логическую единицу, все остальные выдают нуль. Нейроны Кохонена можно воспринимать как набор электрических лампочек, так что для любого входного вектора загорается одна из них.

Ассоциированное с каждым нейроном Кохонена множество весов соединяет его с каждым входом. Например, на рис. 3.1 нейрон Кохонена K_1 имеет веса $w_{11}, w_{21}, \ldots, w_{m1}$, составляющие весовой вектор W_1. Они соединяются через входной слой с входными сигналами $x_1, x_2, \ldots, x_m$, составляющими входной вектор X. Подобно нейронам большинства сетей выход NET каждого нейрона Кохонена является просто суммой взвешенных входов. Это может быть выражено следующим образом:

$$NET_j = \sum_i x_i w_{ij}. \tag{3.21}$$

Нейрон Кохонена с максимальным значением NET является «победителем». Его выход равен единице, у остальных он равен нулю.

Слой Гроссберга функционирует в сходной манере. Его выход NET является взвешенной суммой выходов $k_1, k_2, \ldots, k_n$ слоя Кохонена, образующих вектор K. Вектор соединяющих весов, обозначенный через V_1, состоит из весов $v_{11}, v_{21}, \ldots, v_{n1}$. Тогда выход NET каждого нейрона Гроссберга есть

$$NET_j = \sum_i k_i w_{ij}, \tag{3.22}$$

где NET_j — выход j-го нейрона Гроссберга

Если слой Кохонена функционирует таким образом, что если у одного нейрона величина NET равна единице, а у остальных равна нулю, то лишь один элемент вектора K отличен от нуля, и вычисления очень просты. Фактически каждый нейрон слоя Гроссберга лишь выдает величину веса, который связывает этот нейрон с единственным ненулевым нейроном Кохонена.

3.4.2. Обучение слоя Кохонена в сетях встречного распространения

Слой Кохонена классифицирует входные векторы в группы схожих. Это достигается с помощью такой подстройки весов слоя Кохонена, что близкие входные векторы активируют один и тот же нейрон данного слоя. Затем задачей слоя Гроссберга является получение требуемых выходов.

Обучение слоя Кохонена протекает без учителя. Поэтому трудно (и не нужно) предсказывать, какой именно нейрон Кохонена будет активироваться для заданного входного вектора. Необходимо лишь гарантировать, чтобы в результате обучения разделялись несхожие входные векторы.

Желательно (хотя и не обязательно) нормализовать входные векторы перед тем, как предъявлять их сети. Это выполняется с помощью деления каждого компонента входного вектора на длину вектора. Данная длина находится извлечением квадратного корня из суммы квадратов компонентов вектора. В алгебраической записи она выглядит следующим образом:

$$x_i^{'} = \frac{x_i}{\sqrt{x_1^2 + x_2^2 + \ldots + x_n^2}}. \tag{3.23}$$

Это превращает входной вектор в единичный вектор с тем же самым направлением, т. е. в вектор единичной длины в n-мерном пространстве.

Уравнение (3.23) обобщает хорошо известный случай двух измерений, когда длина вектора равна гипотенузе прямоугольного треугольника, образованного его x и y компонентами, как это следует из известной теоремы Пифагора. На рис. 3.2 такой двумерный вектор $\mathbf{V}$ представлен в координатах x — y, причем координата x равна четырем, а координата y — трем. Квадратный корень из суммы квадратов этих компонентов равен пяти. Деление каждо-

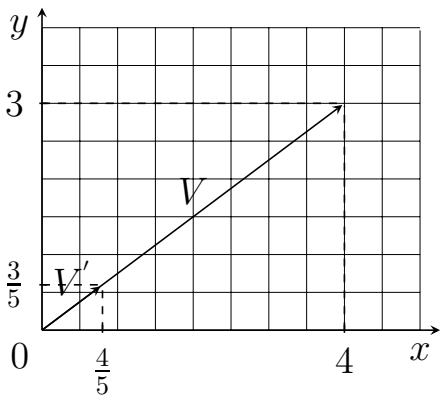

Рис. 3.2. Единичный входной вектор

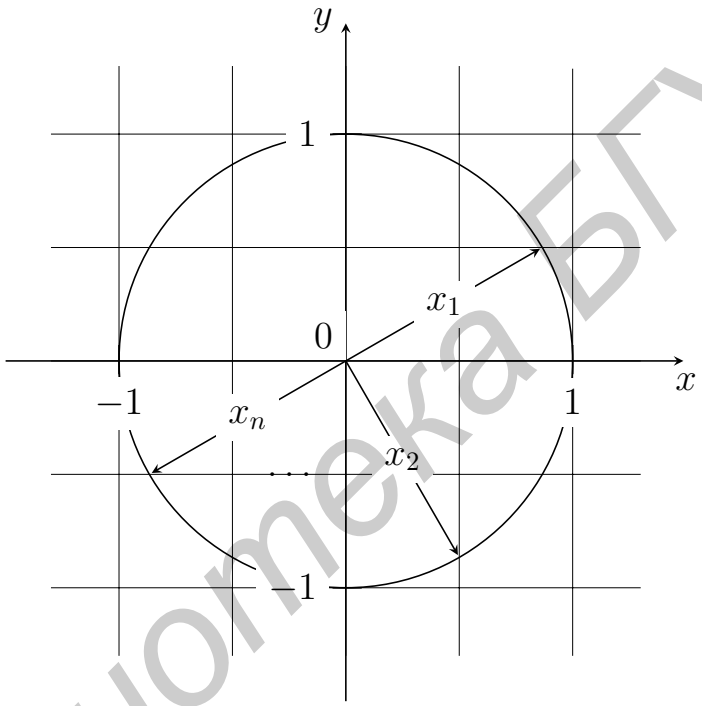

Рис. 3.3. Двумерные единичные векторы на единичной окружности

го компонента $\mathbf{V}$ на пять дает вектор $\mathbf{V}'$ с компонентами 4/5 и 3/5, где $\mathbf{V}'$ указывает в том же направлении, что и $\mathbf{V}$, но имеет единичную длину.

На рис. 3.3 показано несколько единичных векторов. Они оканчиваются в точках единичной окружности (окружности единичного радиуса), что имеет место, когда у сети лишь два входа. В случае трех входов векторы представлялись бы стрелками, оканчивающимися на поверхности единичной сферы. Эти представления могут быть перенесены на сети, имеющие произвольное число входов, где каждый входной вектор является стрелкой, оканчивающейся на поверхности единичной гиперсферы (полезной абстракцией, хотя и не допускающей непосредственной визуализации).

При обучении слоя Кохонена на вход подается входной вектор и вычисляются его скалярные произведения с векторами весов, связанными со всеми

нейронами Кохонена. Нейрон с максимальным значением скалярного произведения объявляется «победителем» и его веса подстраиваются. Так как скалярное произведение, используемое для вычисления величин NET, является мерой сходства между входным вектором и вектором весов, то процесс обучения состоит в выборе нейрона Кохонена с весовым вектором, наиболее близким к входному вектору, и дальнейшем приближении весового вектора к входному. Следует отметить, что процесс является самообучением, выполняемым без учителя. Сеть самоорганизуется таким образом, что данный нейрон Кохонена имеет максимальный выход для данного входного вектора. Уравнение, описывающее процесс обучения, имеет следующий вид:

$$w_n = w_c + \alpha(x - w_c), \tag{3.24}$$

где w_n — новое значение веса, соединяющего входной компонент x с выигравшим нейроном; w_c — предыдущее значение этого веса; α — коэффициент скорости обучения, который может варьироваться в процессе обучения.

Каждый вес, связанный с выигравшим нейроном Кохонена, изменяется пропорционально разности между его величиной и величиной входа, к которому он присоединен. Направление изменения минимизирует разность между весом и его входом.

На рис. 3.4 этот процесс показан геометрически в двумерном виде. Сначала находится вектор $\mathbf{X} - \mathbf{W}_c$, для этого проводится отрезок из конца $\mathbf{W}$ в конец $\mathbf{X}$. Затем этот вектор укорачивается умножением его на скалярную величину α, меньшую единицы, в результате чего получается вектор изменения δ. Окончательно новый весовой вектор $\mathbf{W}_n$ является отрезком, направленным из начала координат в конец вектора δ. Отсюда можно видеть, что 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 примерно $0,7$ и может постепенно уменьшаться в процессе обучения. Это позволяет делать большие начальные шаги для быстрого грубого обучения и меньшие шаги при подходе к окончательной величине.

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 ($\alpha = 1$). 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

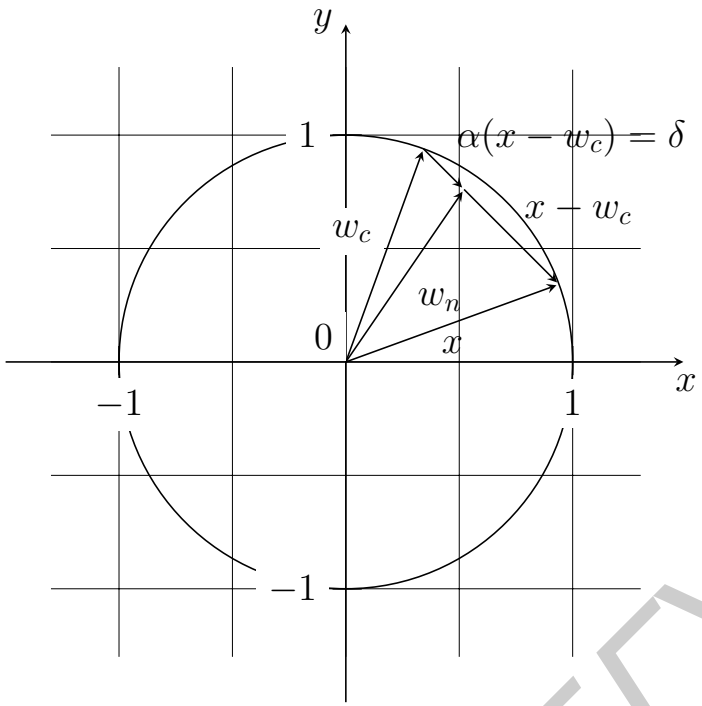

Рис. 3.4. Вращение весового вектора в процессе обучения ($\mathbf{W}_n$ — вектор новых весовых коэффициентов, $\mathbf{W}_c$ — вектор старых весовых коэффициентов)

вблизи «центра» входных векторов, для которых данный нейрон является «победителем».

Всем весам сети перед началом обучения следует придать начальные значения. Общепринятой практикой при работе с нейронными сетями является присваивание весам небольших случайных значений. При обучении слоя Кохонена случайно выбранные весовые векторы следует нормализовать. Окончательные значения весовых векторов после обучения совпадают с нормализованными входными векторами. Поэтому нормализация перед началом обучения приближает весовые векторы к их окончательным значениям, сокращая, таким образом, обучающий процесс.

Рандомизация весов слоя Кохонена может породить серьезные проблемы при обучении, т. к. в результате неё весовые векторы распределяются равномерно по поверхности гиперсферы. Из-за того, что входные векторы, как правило, распределены неравномерно и имеют тенденцию группироваться на относительно малой части поверхности гиперсферы, большинство весовых векторов будет так удалено от любого входного вектора, что они никогда не будут давать наилучшего соответствия. Эти нейроны Кохонена будут всегда иметь нулевой выход и окажутся бесполезными. Более того, оставшихся весов, дающих наилучшие соответствия, может оказаться слишком мало, чтобы разделить входные векторы на классы, которые расположены близко друг к другу на поверхности гиперсферы.

Допустим, что имеется несколько множеств входных векторов. Все множества сходные, но должны быть разделены на различные классы. Сеть должна быть обучена активировать отдельный нейрон Кохонена для каждого класса. Если начальная плотность весовых векторов в окрестности обучающих векторов слишком мала, то может оказаться невозможным разделить сходные классы из-за того, что не будет достаточного количества весовых векторов в интересующей нас окрестности, чтобы приписать по одному из них каждому классу входных векторов.

Наоборот, если несколько входных векторов получены незначительными изменениями из одного и того же образца и должны быть объединены в один класс, то они должны включать один и тот же нейрон Кохонена. Если же плотность весовых векторов очень высока вблизи группы слегка различных входных векторов, то каждый входной вектор может активировать отдельный нейрон Кохонена. Это не является катастрофой, т. к. слой Гроссберга может отобразить различные нейроны Кохонена в один и тот же выход, но это будет расточительной тратой нейронов Кохонена.

Наиболее желательное решение состоит в том, чтобы распределять весовые векторы в соответствии с плотностью входных векторов, которые должны быть разделены, помещая тем самым больше весовых векторов в окрестности большого числа входных векторов. На практике это невыполнимо, однако существует несколько методов приближенного достижения тех же целей.

Одно из решений, известное под названием метода выпуклой комбинации, состоит в том, что все веса приравниваются к одной и той же величине

$$w_i = \frac{1}{\sqrt{n}}, \tag{3.25}$$

где n — число входов и, следовательно, число компонентов каждого весового вектора. Благодаря этому все весовые векторы совпадают и имеют единичную длину. Каждому же компоненту входа $\mathbf{X}$ придается значение

$$x_i = \alpha x_i + \frac{1-\alpha}{\sqrt{n}}, \tag{3.26}$$

где n — число входов. 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. Это позволяет разделять входные векторы и окончательно приписывает им их истинные значения. Весовые векторы отслеживают один или небольшую группу входных векторов и в конце обучения дают требуемую картину выходов. Метод выпуклой комбинации хорошо работает, но замедляет процесс обучения, т. к. весовые векторы подстраиваются к изменяющейся цели. Другой подход состоит в добавлении шума к входным векторам. Тем самым они

подвергаются случайным изменениям, схватывая в конце концов весовой вектор. Этот метод также работоспособен, но еще более медленный, чем метод выпуклой комбинации.

Третий метод начинает со случайных весов, но на начальной стадии обучающего процесса подстраивает все веса, а не только связанные с выигравшим нейроном Кохонена. 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.

Еще один метод наделяет каждый нейрон Кохонена «чувством справедливости». Если он становится победителем чаще своей законной доли времени (примерно $\frac{1}{k}$, где k — число нейронов Кохонена), он временно увеличивает свой порог, что уменьшает его шансы на выигрыш, давая тем самым возможность обучаться и другим нейронам.

Во многих приложениях точность результата существенно зависит от распределения весов. К сожалению, эффективность различных решений исчерпывающим образом не оценена и остается проблемой.

Метод обучения Кохонена обладает полезной и интересной способностью извлекать статистические свойства из множества входных данных. Как показано Кохоненом, для полностью обученной сети вероятность того, что случайно выбранный входной вектор (в соответствии с функцией плотности вероятности входного множества) будет ближайшим к любому заданному весовому вектору, равна $\frac{1}{k}$, где k — число нейронов Кохонена. Это является оптимальным распределением весов на гиперсфере. (Предполагается, что используются все весовые векторы, что имеет место лишь в том случае, если используется один из обсуждавшихся методов распределения весов.)

3.4.3. Достоинства, недостатки и проблемы обучения

Роберт Хехт-Нильсон, создатель сети встречного распространения (СВР), осознавал ее ограничения: «СВР, конечно, уступает обратному распространению в большинстве приложений, связанных с сетевыми отображениями. Ее преимущества в том, что она проста и дает хорошую статистическую модель для своей среды входных векторов».

К этому можно добавить, что сеть встречного распространения быстро обучается, и при правильном использовании она может сэкономить значительное количество машинного времени. Она полезна также для быстрого моделирования систем, где большая точность обратного распространения вынуждает отдать ему предпочтение в окончательном варианте, но важна быстрая начальная аппроксимация. Возможность порождать функцию и обратную к ней также нашла применение в ряде систем.

Литература

1. Haykin, S. Neural Networks, A Comprehensive Foundation, 2nd ed., Prentice Hall, 1999.

2. Осовский, С. Нейронные сети для обработки информации / С. Осовский — М. : Финансы и статистика, 2004. — 344 с.

3. Rosenblatt, F. Principle of neurodynamics. Spartan, 1992.

4. Falhman, S. E. Faster learning variations of backpropagation: an empirical study, Proc. 1998 Connectionist Models Summer School. — Los Altos, USA: Morgan Kaufmann, p. 38–51, 1988.

5. Nguyen, D., Widrow B. Improving the Learning Speed of 2–Layer Neural Networks by Choosing Initial Values of the Adaptive Weights. Information Systems Laboratory, Stanford University, CA 94305, p. 21–26.

6. Kohonen, T. Self–organization and associative memory. 2d ed. New York, Springer–Verlag, 1988.

7. Cheng, Y. H., Lin, C. S. Learning algorithm for radial basis function network with the capability of adding and pruning neurons: Proc. 1994 Conf. ICNN. Orlando, p. 797–801, 1994.

8. Trassenko, L., Roberts, S. Supervised and unsupervised learning in radial basis function classifiers // IEEE Proc. Vis. Image Signal Process., Vol. 141, p. 210–216, 1994.

9. Hopfield, J. Neural networks and physical systems with emergent collective computational abilities // Proc. National Academy of Science USA, Vol. 79, p. 2554–2558, 1982.

10. Diamantaras, K., Kung, S. Principal component neural networks, theory and applications. — New York: Willey, 1996.

11. Oja, E. A simplified neuron model as a principal component analyzer, Journal of Mathematical Biology, Vol. 15, p. 267–273, 1982.

12. Oja, E. Principal components, minor components and linear neural networks, Neural Networks, Vol. 5, p. 927–935, 1992.

13. Sanger, T. D. Optimal unsupervised learning in a single-layer linear feedforward neural network, Neural Networks, Vol. 12, p. 459–473, 1989.

Св. план 2010, поз. 58

Учебное издание

Садыхов Рауф Хосровович
Пушкин Егор Владимирович
Шадура Андрей Олегович

НЕЙРОСЕТЕВЫЕ ТЕХНОЛОГИИ ПРИНЯТИЯ РЕШЕНИЙ

Методическое пособие
для студентов специальности 1-98 80 03
«Аппаратное программно-техническое
обеспечение информационной безопасности»

Редактор Е. Н. Батурчик
Корректор А. В. Бас

Подписано в печать	Формат 60×84 1/16.	Бумага офсетная.
Гарнитура «Компьютер модерн».	Отпечатано на ризографе.	Усл. печ. л.
Уч.-изд. л. 3,5.	Тираж 50 экз.	Заказ 544.

Издатель и полиграфическое исполнение: учреждение образования
«Белорусский государственный университет информатики и радиоэлектроники»
ЛИ №02330/0494371 от 16.03.2009. ЛП №02330/0494175 от 03.04.2009.
220013, Минск, П. Бровки, 6