

С.Г. ТОЛМАЧЕВ

**СИСТЕМЫ
ИСКУССТВЕННОГО
ИНТЕЛЛЕКТА.
НЕЙРОСЕТЕВЫЕ МОДЕЛИ**

Министерство образования и науки Российской Федерации
Балтийский государственный технический университет «Военмех»
Кафедра «Системы обработки информации и управления»

С.Г. ТОЛМАЧЕВ

СИСТЕМЫ
ИСКУССТВЕННОГО
ИНТЕЛЛЕКТА.
НЕЙРОСЕТЕВЫЕ МОДЕЛИ

Учебное пособие

Санкт-Петербург
2011

УДК 004.8(075.8)
Т52

Толмачев, С.Г.

Т52

Системы искусственного интеллекта.
Нейросетевые модели: учебное пособие / С.Г.
Толмачев; Балт. гос. техн. ун-т. – СПб., 2011. –
132 с
ISBN 978-5 -85546-633-1

Приводятся основные сведения о структуре и принципах функционирования искусственных нейронных сетей. Рассматриваются функционирование формального нейрона, классификация нейронных сетей по их архитектуре и видам обучения, типовые постановки различных нейросетевых задач и способы их решения.

Предназначено для студентов старших курсов, обучающихся по специальностям «Информационные системы и технологии» и «Автоматизированные системы обработки информации и управления».

УДК 004.8(075.8)

Р е ц е н з е н т ы: д-р техн. наук проф., глав. науч. сотрудник ОАО
«Концерн «Гранит-Электрон»» *С.Н. Шаров*; канд. техн. наук,
проф., зав. каф. И5 БГТУ *Н.Н. Смирнова*

*Утверждено
редакционно-издательским
советом университета*

ISBN 978-5 -85546-633-1

© С.Г. Толмачев, 2011
© БГТУ, 2011

ВВЕДЕНИЕ

Одним из наиболее мощных инструментов создания интеллектуальных систем являются искусственные нейронные сети (ИНС), моделирующие базовые механизмы обработки информации, присущие человеческому мозгу. Известно, что мозг работает принципиально другим образом и зачастую более эффективно, чем любая вычислительная машина, созданная человеком. Именно этот факт в течение многих лет побуждает ученых к работе по созданию и исследованию искусственных нейронных сетей.

Мозг представляет собой чрезвычайно сложную систему обработки информации. Он обладает способностью организовывать свои структурные компоненты, называемые нейронами, так, чтобы они могли выполнять конкретные задачи (распознавание образов, обработка сигналов органов чувств, моторные функции) во много раз быстрее, чем самые быстродействующие современные компьютеры. Примером такой задачи может служить обычное зрение. В функции зрительной системы входит создание представления об окружающем мире в таком виде, который обеспечивает возможность взаимодействия с ним. Мозг последовательно выполняет задачи по распознаванию (например, распознавание знакомого лица в незнакомом окружении) и затрачивает на это 100...200 мс. Выполнение аналогичных задач меньшей сложности на компьютере может занять несколько часов.

Чтобы осознать масштабы проблемы создания машины, работающей так же совершенно, как наш мозг, достаточно задуматься над некоторыми рутинными задачами, выполняемыми нами ежедневно. Допустим, Вы сидите за рабочим столом, и в это время в комнату входит Ваш коллега, вернувшийся из отпуска. На нем новая футболка, на загорелом лице – солнцезащитные очки, и выглядит он несколько помолодевшим, потому что сбрил бороду. Узнаете ли вы его? Несомненно, поскольку маскировка не входит в его планы. В процессе разговора он спрашивает Вас: «А где книга, которую я дал Вам почитать?». Вы интерпретируете вопрос как просьбу вернуть книгу. Затем переводите взгляд на свой стол и

видите среди лежащих на нем книг и стопок бумаг ту книгу, о которой идет речь, протягиваете к ней руку, извлекаете из груды документов и отдаете коллеге. Такие повседневные задачи не требуют от нас большого интеллектуального напряжения, но решение каждой из них включает в себя множество точно рассчитанных шагов. Сложность решения таких задач можно почувствовать, пытаясь запрограммировать компьютерную систему для распознавания объектов по их внешнему виду или другим признакам, принимающую решения, в зависимости от контекста и т.п.

Более простым примером может служить сонар летучей мыши, представляющий собой систему активной эхо-локации. Кроме представления информации о расстоянии до нужного объекта, этот локатор позволяет вычислить такие параметры объекта, как относительная скорость, размер отдельных элементов и направление движения. Для выделения этой информации из получаемого сигнала крохотный мозг летучей мыши производит сложные нейронные вычисления.

Что же позволяет мозгу человека или летучей мыши добиться таких результатов? При рождении мозг уже имеет совершенную структуру, позволяющую строить собственные правила на основании того, что обычно называется опытом. Опыт накапливается с течением времени до последних дней жизни человека, при этом особенно масштабные изменения происходят в первые два года жизни.

Развитие нейронов связано с понятием пластичности мозга – способности настройки нервной системы в соответствии с окружающими условиями. Пластичность играет самую важную роль в работе нейронов в качестве элементарных единиц обработки информации в человеческом мозге. Аналогичным образом в ИНС происходит настройка искусственных нейронов. В общем случае ИНС представляет собой машину, моделирующую способ решения мозгом конкретной задачи. Эта сеть реализуется с помощью электронных компонентов (нейропроцессоров) или моделируется программой, выполняемой на цифровом компьютере. Для того чтобы добиться высокой производительности, в ИНС используют множество взаимосвязей между элементарными ячейками вычислений – нейронами. Среди множества определений нейронных сетей наиболее точным является определение ИНС как адаптивной машины: *искусственная нейронная сеть – это распределенный*

параллельный процессор, состоящий из типовых элементов обработки информации, накапливающих экспериментальные знания, и предоставляющий их для последующей обработки. Нейронная сеть сходна с мозгом в двух аспектах:

1) знания поступают в нейронную сеть из окружающей среды и используются сетью в процессе обучения;

2) для накопления знаний используются межнейронные связи, называемые также синаптическими весами.

Процедура, используемая для осуществления процесса обучения, называется алгоритмом обучения. Ее функция состоит в модификации синаптических весов ИНС определенным образом так, чтобы сеть приобрела необходимые свойства.

Модификация весов является традиционным способом обучения ИНС. Такой подход близок к теории адаптивных линейных фильтров, которые применяются в управлении. Однако для ИНС существует еще и возможность модификации собственной топологии, основывающаяся на том факте, что в живом мозге нейроны могут отмирать, а новые синаптические связи могут создаваться.

Таким образом, ИНС реализуют свою вычислительную мощь благодаря двум основным своим свойствам: параллельно-распределенной структуре и способности обучаться и обобщать полученные знания. Под свойством обобщения понимается способность ИНС генерировать правильные выходы для входных сигналов, которые не были учтены в процессе обучения (тренировки). Эти два свойства делают ИНС системой обработки информации, способной решать сложные многомерные задачи, которые на сегодняшний день являются трудноразрешимыми.

Следует отметить, что на практике автономные ИНС зачастую не могут обеспечить готовые решения. Их следует интегрировать в сложные системы. Комплексную задачу можно разбить на ряд более простых задач, часть из которых может быть решена нейронными сетями.

Области применения ИНС весьма разнообразны: распознавание и анализ текста и речи, семантический поиск, экспертные системы и системы поддержки принятия решений, предсказание курсов акций, системы безопасности. Можно привести несколько примеров использования ИНС в разных областях.

1. Системы безопасности на транспорте. Американская фирма Science Application International Corporation использовала ИНС в

своем проекте TNA. Разрабатываемое устройство предназначено для обнаружения пластиковой взрывчатки в запакованном багаже. Багаж подвергается бомбардировке частицами, вызывающими вторичное излучение, спектр которого анализируется нейронной сетью. Устройство обеспечивает вероятность обнаружения взрывчатки выше 97% и способно просматривать 10 мест багажа в минуту.

2. Нейросетевые программные пакеты на финансовых рынках. Американский Chemical Bank использует нейросетевую систему фирмы Neural Data для предварительной обработки транзакций на валютных биржах, фильтруя «подозрительные» сделки. Citibank использует нейросетевые предсказания с 1990 г. Автоматический дилинг показывает доходность, превышающую показатели большинства брокеров. Можно отметить, что труды семинара «Искусственный интеллект на Уолл-стрит» составляют несколько увесистых томов.

3. Мониторинг и автоматическая рубрикация новостей. Распознавание темы текстовых сообщений – другой пример использования ИНС. Сервер новостей Convectis (продукт фирмы Aptex Software Inc.) обеспечивает автоматическую рубрикацию сообщений по категориям. Сверяя значения слов по контексту, Convectis способен в реальном масштабе времени распознавать тематику и рубрицировать огромные потоки текстовых сообщений, передаваемых по сетям Reuters, NBC, CBS и др. После анализа сообщения генерируется аннотация, список ключевых слов и список рубрик, к которым относится данное сообщение [2]

4. Автопилотирование беспилотных летательных аппаратов. Гиперзвуковой самолет-разведчик LoFLYTE (Low-Observable Flight Test Experiment) – реактивный беспилотный самолет длиной 2,5 м – был разработан для NASA и ВВС США фирмой Accurate Automation Corp. в рамках программы поддержки малого инновационного бизнеса. Это экспериментальная разработка для исследования новых принципов пилотирования. Она включает нейронные сети, позволяющие автопилоту обучаться, копируя приемы пилотирования летчика. Со временем нейросети перенимают опыт управления, а скорость обработки информации позволяет быстро находить выход в экстремальных и аварийных ситуациях. LoFLYTE предназначен для полетов со сверхзвуковой скоростью, когда скорости реакции пилота может не хватить для адекватного реагирования на изменения режима полета.

В настоящее время ИНС являются важным расширением понятия вычисления. Они уже позволили справиться с рядом непростых проблем и обещают создание новых программ и устройств, способных решать задачи, которые пока под силу только человеку. Современные нейροкомпьютеры используются, в основном, в виде программных продуктов и поэтому редко задействуют свой потенциал «параллелизма». Эпоха настоящих параллельных нейровычислений начнется с появлением на рынке аппаратных реализаций – специализированных нейрочипов и плат расширения, предназначенных для обработки речи, видео, статических изображений и других типов образной информации.

Другой областью применения ИНС является их использование в специализированных программных агентах-роботах, предназначенных для обработки информации, а не для физической работы. Интеллектуальные помощники должны облегчать пользователям общение с компьютером. Их отличительной чертой будет стремление как можно лучше понять, что от них требуется, за счет наблюдения и анализа поведения своего «хозяина». Стараясь обнаружить в этом поведении некоторые закономерности, интеллектуальные агенты должны своевременно предложить свои услуги для выполнения определенных операций, например для фильтрации новостных сообщений, для резервного копирования документов, над которыми работает пользователь, и т.п. Именно поэтому ИНС, способные обобщать данные и находить в них закономерности, являются естественным компонентом подобных программных агентов.

1. КОМПЬЮТЕРЫ И МОЗГ

1.1. Биологический нейрон

Нервную систему человека можно упрощенно представить в виде трехступенчатой структуры. Центром этой системы является мозг, состоящий из сети нейронов (рис. 1.1). Он получает информацию, анализирует ее и выдает соответствующие решения. Рецепторы преобразуют сигналы из окружающей среды и внутренних органов в электрические импульсы, воспринимаемые нейронной сетью (мозгом). Рецепторы обеспечивают связь нашего мозга с внешним миром, реализуя поступление в него зрительной, слуховой, вкусовой, обонятельной и осязательной информации. Эф-

факторы преобразуют электрические импульсы, сгенерированные мозгом, в выходные сигналы, управляющие мышцами, внутренними органами, стенками сосудов. Таким образом мозг контролирует работу сердца, дыхание, кровяное давление, температуру, поддерживает нужное содержание кислорода в крови и т.д. Промежуточные нейроны обрабатывают информацию, получаемую от сенсорных нейронов, и передают ее эффекторным нейронам.

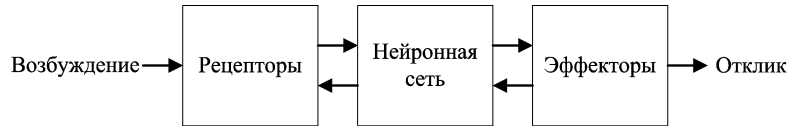


Рис. 1.1. Упрощенная схема нервной системы

Нужно отметить, что мозг построен из клеток двух типов: глиальных и нейронов. И хотя роль глиальных клеток, видимо, довольно значительна, большинство ученых полагают, что в основном понять работу мозга можно при изучении нейронов, объединенных в единую связанную сеть. Этот подход и используется при построении искусственных нейронных сетей (ИНС).

Следует отметить, что имеются и другие мнения. Некоторые исследователи считают, что главные процессы происходят не в нейронной сети, а в самих клетках, а именно в их цитоскелетоне, в так называемых микротрубочках. Согласно этой точке зрения, и память, и даже сознание определяются изменениями белков во внутриклеточных структурах и связанными с ними квантовыми эффектами.

Количество нейронов в мозге оценивается величиной $10^{10} \dots 10^{11}$. В биологическом нейроне можно выделить следующие структурные единицы (рис. 1.2):

- тело клетки (сому);
- дендриты — множество ветвящихся коротких (не более 1 мм) нервных волокон, которые собирают информацию от других нейронов;
- аксон — единственное тонкое длинное (иногда больше метра) нервное волокно. Аксон обеспечивает проведение импульса и передачу воздействия на другие нейроны или мышечные волокна. На своем окончании аксон также разветвляется и образует контакты с дендритами других нейронов;

- синапс – место контакта нервных волокон, которое передает возбуждение от клетки к клетке. Передача импульса через синапс почти всегда односторонняя. По направлению передачи импульса различают пресинаптические и постсинаптические клетки.

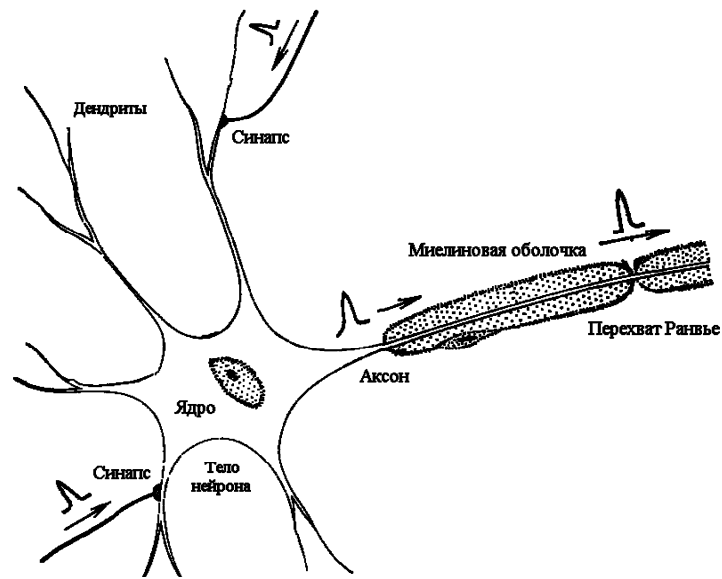


Рис. 1.2. Биологический нейрон

Клеточная мембрана. Процесс генерации нервных импульсов обусловлен переносом ионов через оболочку клетки – клеточную мембрану (рис. 1.3). Она создает и поддерживает постоянную концентрацию веществ внутри клетки, обеспечивает ее механическую прочность и транспортировку молекул и ионов в обоих направлениях, а для нервных волокон – проведение нервных импульсов по волокну.

Мембрана состоит из двух слоев молекул липидов (жиров). Молекулы липидов содержат полярную "голову" и два гидрофобных углеводородных "хвоста", ориентированных внутрь мембраны. Такая конструкция оказывается стабильной и энергетически выгодной (по структуре она аналогична пленке мыльного пузыря). По химическому составу мембраны весьма разнообразны. Толщина мембраны – около 50 нм. В ее состав входит множество механиз-

мов, необходимых для работы клетки. Рассмотрим только те, что имеют отношение к возникновению и передаче нервных сигналов. Это ионные натриевые насосы, натриевые и калиевые каналы.

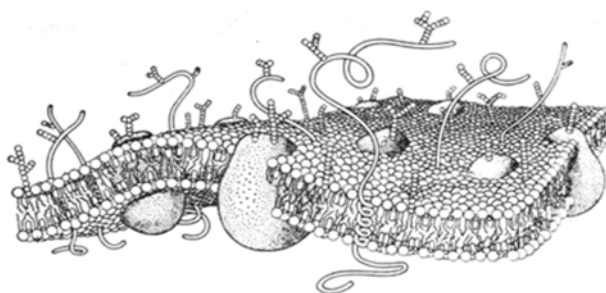


Рис. 1.3. Клеточная мембрана

Натриевые насосы. Внутри клетки, в цитоплазме, преобладают ионы K^+ , снаружи – ионы Na^+ . Активный мембранный транспорт, биологический натриевый насос, переносит K^+ внутрь, а Na^+ наружу (рис. 1.4). Такой перенос происходит в направлении роста электрохимического потенциала и требует затрат энергии. Для работы натриевого насоса используется энергия гидролиза аденозинтрифосфата (АТФ) – основного энергетического аккумулятора клетки. Гидролизом называется химическая реакция расщепления вещества с участием воды. Механизм переноса использует белок-носитель, образующий комплекс с ионами на одной стороне мембраны и отщепляющий эти ионы на противоположной стороне. Носитель с ионами Na^+ , K^+ не покидает мембрану. В итоге через мембрану проходят потоки ионов K^+ , направленные внутрь клетки и Na^+ – наружу.

Известно, что за полный цикл работы натриевого насоса происходит гидролиз одной молекулы АТФ, перенос трех ионов Na^+ из клетки в среду и двух ионов K^+ из среды в клетку (рис. 1.5). Число ионов, транспортируемых за один цикл работы насоса, не зависит от их концентраций по обе стороны мембраны. Транспортная система, осуществляющая перенос ионов, имеет строго заданное количество мест: три для ионов Na^+ и два для ионов K^+ . Таким образом, в результате работы натриевого насоса из клетки выносятся избыточный положительный заряд, генерируется электрический ток и концентрация ионов K^+ и Na^+ снаружи и внутри клетки становится неравновесной.

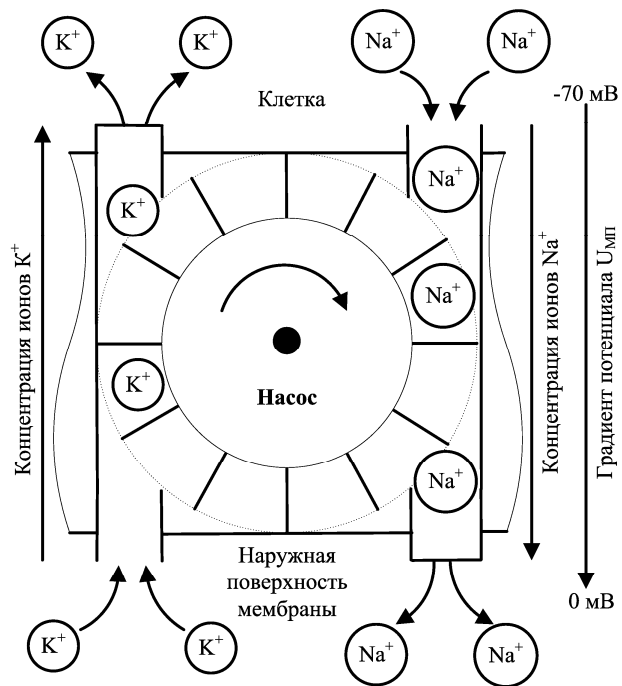


Рис. 1.4. Натриевый насос

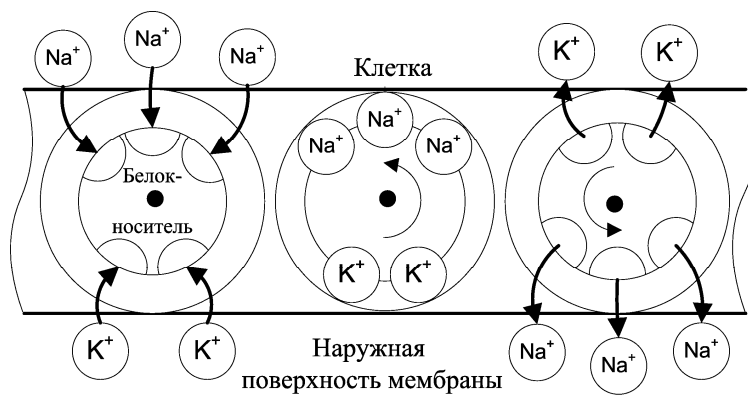


Рис. 1.5. Схема переноса ионов натрия и калия за один цикл работы натриевого насоса

Калиевые и натриевые каналы. В мембране существуют каналы, пропускающие только ионы K^+ в обоих направлениях. Каналы являются управляемыми и могут открываться или закрываться в зависимости от разности потенциалов между цитоплазмой и внешней средой, а также обладают инерционными свойствами. В состоянии покоя часть K^+ -каналов открыта. Из-за разницы концентраций K^+ и Na^+ на разных сторонах мембраны ионы K^+ начинают диффундировать через K -каналы из цитоплазмы наружу. Этот процесс приводит к оттоку положительного заряда из клетки, и цитоплазма заряжается отрицательно. Возникают кулоновские силы, препятствующие дальнейшей диффузии ионов K^+ . Как только эти процессы уравниваются друг друга, диффузия ионов K^+ через K -каналы прекращается и потенциал цитоплазмы достигает равновесного состояния. Между клеткой и средой образуется разность потенциалов, называемая мембранным потенциалом (МП). Клетка заряжается отрицательно относительно среды. В состоянии покоя МП составляет величину порядка -70 мВ.

Натриевые каналы аналогичны калиевым, но пропускают только ионы Na^+ . Отличаются они также уровнем потенциала, открывающим канал, и инерционными характеристиками. В состоянии покоя закрыты, открываются при изменении потенциала цитоплазмы в положительную сторону, закрываются сами по себе, через некоторое время после открытия.

Нервный импульс. Нервный импульс (НИ) или спайк – процесс распространения возбуждения по аксону от тела клетки до окончания аксона. Это основная единица информации, передаваемая по волокну, поэтому модель генерации и распространения НИ – одна из важнейших в теории ИНС. Импульсы по волокну передаются в виде скачков потенциала внутриклеточной среды по отношению к внешней среде, окружающей клетку. Скорость передачи – от 1 до 100 м/с.

Процесс генерации нервного импульса можно условно разделить на несколько стадий.

1. Возбуждение импульсов происходит по принципу "все или ничего". Если потенциал цитоплазмы станет положительнее, чем -50 мВ, то произойдет генерация импульса. Изменение потенциала может быть вызвано внешним воздействием (электродом, введенным в цитоплазму и подключенным к источнику напряжения) или явиться следствием воздействия импульса от соседнего

нейрона. Деполяризацией мембраны называется изменение ее разности потенциалов от состояния равновесия в положительную сторону, например, от -70 до -50 мВ.

2. Если мембрана деполяризуется до уровня -50 мВ, то открываются натриевые каналы и поток ионов Na^+ начинает поступать в клетку. Возникает ток I_{Na} , направленный внутрь клетки, что приводит к дальнейшей деполяризации мембраны. Так как Na-каналы управляются разностью потенциалов на мембране, дальнейшая деполяризация открывает все новые Na-каналы, что приводит к росту I_{Na} . Положительная обратная связь в данном процессе дает резкий скачок потенциала в сторону деполяризации.

3. Натриевые каналы закрываются самопроизвольно, через некоторое время после открытия. Закрывшись, находятся в неактивном состоянии около 1 мс, т.е. не могут быть открыты вновь, несмотря на деполяризованность мембраны. Ток I_{Na} падает до нуля, деполяризация прекращается. В этот момент потенциал достигает значения $+40$ мВ.

4. Калиевые каналы тоже управляются разностью потенциалов, но они более инертны и открываются с задержкой по сравнению с натриевыми. Так как потенциал смещен в сторону деполяризации, открытие калиевых каналов приводит к диффузионному калиевому току, направленному из клетки. Ток I_{K} нарастает медленно и направлен противоположно I_{Na} . К моменту, когда I_{K} достигает максимума, ток I_{Na} уменьшается и потенциал начинает меняться в противоположную сторону: мембрана реполяризуется.

5. За счет тока I_{K} мембрана реполяризуется до исходного потенциала -70 мВ, а из-за инерционности K-каналов гиперполяризуется до -90 мВ. Калиевый ток прекращается.

6. По окончании импульса натриевые каналы пребывают в неактивном состоянии около 1 мс, когда возникновение нового импульса невозможно. Происходит деполяризация до равновесного мембранного потенциала -70 мВ.

7. Вследствие изменения мембранного потенциала открываются натриевые каналы на соседнем участке и процесс возбуждения распространяется вдоль волокна. Зависимости токов I_{K} , I_{Na} и мембранного потенциала $U_{\text{мп}}$ от времени приведены на рис. 1.6.

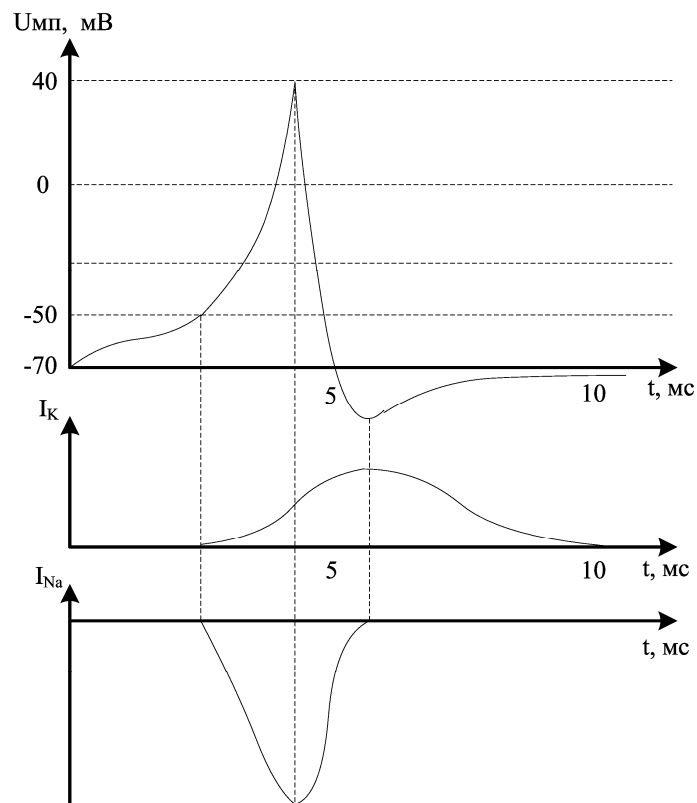


Рис. 1.6. Зависимость ионных токов и мембранного потенциала от времени

В результате прохождения импульса часть ионов калия выходит наружу, а часть ионов натрия входит в клетку, причем в итоге потенциал возвращается к исходному значению -70 мВ . Следовательно, число ионов Na^+ , вошедших в клетку, в точности равно числу ионов K^+ , вышедших из клетки. Натриевый насос восстанавливает разницу концентраций ионов.

Таким образом, при формировании нервного импульса открываются высокопроводящие каналы сначала для ионов натрия, а затем для ионов калия. Импульсная система нейрона устроена так, что, пропустив через себя сигнал, она короткое время не способна принимать сигналы и, следовательно, клетка не будет накапливать ионы натрия. Этот период называется рефракторной фазой, когда

натриевый канал деактивирован в силу особенностей своей структуры. Эта фаза используется натриевым насосом для откачки излишков натрия из клетки.

Стабильность информационного канала в огромной степени зависит от работы натриевого насоса, который обеспечивает неизменность электроэнергетического состояния нейрона. Кроме восстановления исходного мембранного потенциала, рефракторная фаза необходима также для обеспечения согласованного функционирования молекулярных систем мембраны и других частей нервной системы. Большая частота генерации нервных импульсов нейроном лишена смысла, так как передача нервного импульса в месте синаптического контакта двух клеток ограничена медленной скоростью протекания химических процессов в месте соединения нервных волокон. Поэтому частота, с которой нейрон может генерировать импульсы, ограничена величиной 100 Гц.

Электрический импульс длительностью 10 мс и амплитудой 110 мВ является основным элементом информации, передаваемой по нервам и мышцам. Он распространяется вдоль нервного волокна – аксона – от тела клетки в сторону другой нервной клетки или же по направлению того органа, который должен принять этот импульс. Длина аксона может достигать нескольких метров, а диаметр волокна колеблется от 1 до 1000 мкм. Аксоны бывают двух типов: гладкие и миелинизированные. У гладких аксонов мембрана непосредственно соприкасается с окружающей средой, а у миелинизированных толстая жировая (миелиновая) оболочка изолирует волокно от окружающей среды. Через каждые 1...2 мм эта оболочка прерывается, оголяя мембрану (перехват Ранвье). В местах, где отсутствует жировая изолирующая оболочка, аксон соприкасается с окружающей средой. В гладких аксонах, лишенных изолирующей оболочки, электрический импульс распространяется по волокну от точки к точке, как по проводнику. На каждом участке мембрана аксона деполяризуется в результате ионного обмена со средой. В нервных волокнах, упрятанных в изолирующую оболочку, импульс не может распространяться непрерывно по всему волокну, так как оно на больших промежутках изолировано от окружающей среды. Поэтому от одного перехвата Ранвье до другого нервное волокно ведет себя подобно пассивному кабелю, изолированному от окружающей среды миелиновой оболочкой. Электрическая активность мембраны проявляется только в перехватах

Раньше, где имеется непосредственная связь со средой. Электрический импульс как бы перескакивает от одного ретрансляционного участка к другому за счет процессов деполяризации мембраны в местах с отсутствующей изоляцией. Скорость распространения нервного импульса в миелинизированных аксонах выше, чем в гладких.

При распространении вдоль аксона форма импульса не меняется. Импульсы не затухают. На рис. 1.7 показаны нервные импульсы, возникающие в зрительном нервном волокне при воздействии вспышки света постоянной длительности для различных интенсивностей вспышки.

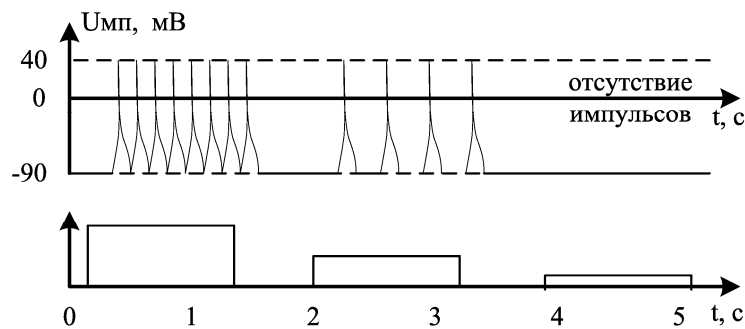


Рис. 1.7. Изменение мембранного потенциала в зрительном волокне при воздействии вспышек света различной интенсивности

Свет воздействует на фоторецепторы и вызывает реакцию в соответствующем зрительном волокне. От интенсивности света зависят не амплитуда импульсов и их форма, а частота следования и общее количество генерируемых импульсов.

Синаптическая передача нервного импульса. Распространение нервного импульса заканчивается у синаптического соединения. Синапс – соединение для передачи нервного импульса от нейрона к нейрону или от нейрона к мышечному волокну. В месте контакта мембраны клеток не сливаются, между ними всегда существует небольшой промежуток – синаптическая щель.

Синапсы бывают химические и электрические, в центральной нервной системе преобладают химические синапсы. У одного нейрона может быть от 300 до 20 000 синапсов между аксонами,

аксонами и дендритами, дендритами и дендритами, аксонами и мышечными волокнами, аксонами и телом клетки и т.п. Схема химического синапса представлена на рис. 1.8.

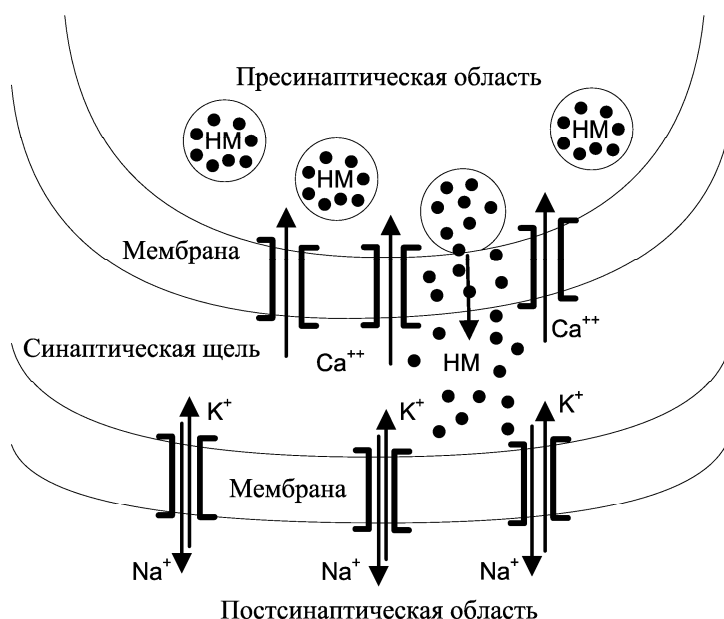


Рис. 1.8. Химический синапс

Передача нервного импульса в химическом синапсе происходит в несколько этапов.

1. Импульс подходит к пресинаптической части волокна.
2. Открываются кальциевые каналы, ионы Ca^{++} поступают в пресинаптическую цитоплазму.

3. В пресинаптической цитоплазме постоянно находится большое количество синаптических пузырьков – образований, содержащих молекулы вещества – нейромедиатора (НМ). Самым распространенным нейромедиатором является ацетилхолин (АХ); кроме него существует около 20 других видов нейромедиаторов. Вследствие появления ионов Ca^{++} в пресинаптической цитоплазме посредством неизвестного пока механизма пузырьки подходят к мембране и лопаются, сливая медиатор в синаптическую щель. Пузырьки выбрасываются не по одному, а квантами по несколько

пузырьков. За один квант в щель попадает $10^3 \dots 10^4$ молекул медиатора. За один импульс проходит $100 \dots 200$ квантов медиатора. Даже в отсутствие нервных импульсов каждую секунду выбрасывается несколько квантов медиатора и синапс поддерживается в состоянии готовности к передаче.

4. Попавший в щель медиатор диффундирует на постсинаптическую сторону щели. Этот процесс занимает время около 0,5 мс и вносит существенный вклад в задержку передачи.

5. Молекулы медиатора улавливаются рецепторами на постсинаптической стороне (рис. 1.9).

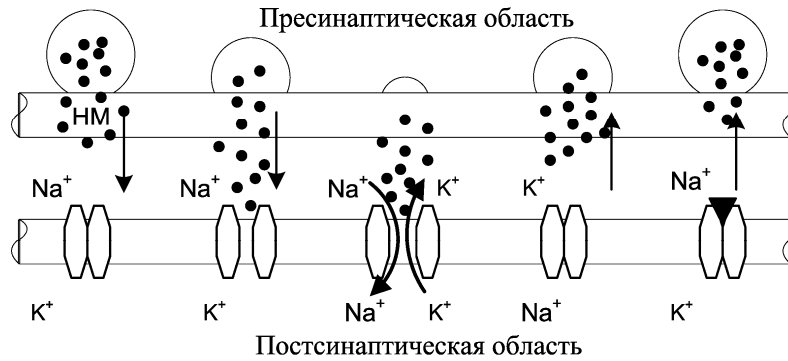


Рис. 1.9. Схема функционирования ионных каналов постсинаптической области

6. Попадание ацетилхолина на рецептор увеличивает проводимость канала в обоих направлениях для ионов Na^+ и K^+ в равной степени. Это эквивалентно небольшому кратковременному "проколу" мембраны. Нейромедиатор действует в течение очень короткого времени, после чего разрушается специфическим ферментом. Так, ацетилхолин, попавший в рецепторы, гидролизруется ацетилхолинэстеразой (АХЭ). Например, яд кураре и подобные ему вещества попадают в молекулы АХЭ, предотвращают гидролиз АХ и останавливают работу синапсов. Одновременно часть нейромедиатора может перемещаться с помощью белков-переносчиков через постсинаптическую мембрану (прямой захват) и в обратном направлении через пресинаптическую мембрану (обратный захват – см. рис. 1.9). В ряде случаев нейромедиатор также поглощается соседними глиальными клетками.

7. Открытие каналов на постсинаптической стороне вызывает поток ионов натрия внутрь, а калия – наружу. Возникший ионный ток возбуждает нервный импульс, который распространяется дальше по постсинаптическому волокну.

Существуют медиаторы, которые работают по-другому. Так как контакты бывают дендро-дендритные, дендро-аксональные, то синапсы оказываются разнообразными по структуре. Каналы являются специфическими для различных ионов (калия, натрия, хлора). В зависимости от вида ионов постсинаптическая мембрана или гиперполяризуется при прохождении соответствующих ионов через канал (тормозящий синапс), или деполяризуется (возбуждающий синапс).

Отдельный нервный импульс слабо влияет на постсинаптическую клетку и не является самостоятельным носителем информации. Для существенного возбуждения необходимо поступление пачки импульсов, плотность которой и является носителем информации в нервной системе. Пачка нервных импульсов является аналоговым сигналом, так как плотность импульсов в пачке может меняться непрерывно. Синаптические потенциалы тоже являются аналоговыми сигналами. Поэтому нервная система человека и животных оказывается аналоговой информационной системой.

Синапсы отличаются друг от друга размерами и возможностями концентрации нейромедиатора вблизи своей оболочки. По этой причине импульсы одинаковой величины, поступающие на входы нервной клетки через различные синапсы, могут возбуждать ее в разной степени. Мерой возбуждения клетки считается уровень поляризации ее мембраны, зависящий от суммарного количества нейромедиатора, выделенного на всех синапсах.

С каждым входом клетки можно сопоставить численные коэффициенты (веса), пропорциональные количеству нейромедиатора, однократно выделяемого на соответствующем синапсе. В математической модели нейрона входные сигналы должны умножаться на эти коэффициенты для того, чтобы корректно учитывать влияние каждого сигнала на состояние нервной клетки. Синаптические веса должны быть вещественными числами, принимающими как положительные, так и отрицательные значения. В первом случае синапс оказывает возбуждающее, а во втором – тормозящее действие, препятствующее возбуждению клетки другими сигналами. В итоге действие возбуждающего синапса может моделиро-

ваться положительным значением синаптического веса, а действие тормозящего синапса – отрицательным значением.

Таким образом, амплитуды сигналов, передаваемых в нервной системе, остаются постоянными и не зависят от важности передаваемой информации или от каких-либо других причин. В то же время частота следования импульсов имеет прямое отношение к внешним событиям. Для кодирования информации в нейронах используется частотный способ кодирования.

Кроме того, нервные импульсы распространяются группами (пачками). Известно, что аксон невозможно постоянно поддерживать в возбужденном состоянии, так как за группой непрерывно следующих друг за другом импульсов следует фаза глубокой гиперполяризации, когда МП аксона становится настолько отрицательным, что волокно уже не способно проявлять свою импульсную активность. В этом проявляется еще одно свойство натриевого насоса: он не только поддерживает стабильность канала информации, но и генерирует такую разность потенциалов, которая препятствует прохождению через аксон следующей группы импульсов, т.е. натриевый насос работает как ограничитель потока информации.

1.2. Особенности взаимодействия нейронов

Количество взаимодействующих друг с другом нервных клеток чрезвычайно велико. Считается, что человеческий мозг содержит около 10^{11} нейронов, каждый из которых выполняет относительно примитивное суммирование весовых коэффициентов входных сигналов и сравнение полученной суммы с пороговым значением. Каждый нейрон имеет свои веса и пороговые значения. Они определяются местонахождением нейрона и решаемой им задачей и могут интерпретироваться аналогично содержанию локальной памяти процессора.

Так как количество нейронов и межнейронных связей огромно, ошибка в срабатывании отдельного нейрона остается незаметной в общей массе взаимодействующих клеток. Нейронная сеть проявляет высокую устойчивость к помехам, отдельные сбои не существенно влияют на результаты ее функционирования. Таково главное отличие нейронных систем от обычных электронных схем, созданных человеком.

Пока ни одна современная технология не позволяет построить ИНС, близкую по масштабам к нейронной сети мозга. Однако изучение биологических нейронных систем позволяет надеяться на создание нового поколения электронных устройств с аналогичными характеристиками.

Другая важная особенность нейронных систем – высокая скорость их функционирования, несмотря на относительно длительный цикл срабатывания каждой отдельной клетки, измеряемый в миллисекундах. Она достигается благодаря параллельной обработке информации в мозге огромным количеством нейронов. Такие операции, как распознавание образов и звуков либо принятие решений, выполняются человеческим мозгом за промежутки времени, измеряемые сотнями миллисекунд. Это означает, что вычисления требуют не более 100 последовательных стадий. Другими словами, для таких сложных задач мозг "запускает" параллельные программы, содержащие около 100 шагов. Это свойство мозга известно как «правило ста шагов». Рассуждая аналогичным образом, можно обнаружить, что количество информации, посылаемое за это время от одного нейрона другому, должно быть очень небольшим (несколько бит). Отсюда следует, что основная информация не передается непосредственно, а захватывается и распределяется в связях между нейронами.

Достижение такого результата при использовании полупроводниковой технологии все еще выходит за рамки современных технических возможностей, хотя цикл срабатывания отдельных исполнительных элементов интегральных схем является достаточно коротким – порядка 10^{-8} с. Если удастся, взяв за образец биологическую нейронную систему, создать устройство с высокой степенью параллельности выполнения отдельных операций, то его быстродействие может быть существенно увеличено и приближено к уровню, наблюдаемому в процессах обработки информации биологическими системами.

1.3. Нейрокомпьютерная обработка информации

Мозг управляет организмом в целом, его движением, восприятием окружающей среды, оперативно реагируя на всевозможные ее изменения. Высшие формы живых организмов в ходе эволюции выработали способность к изменению и адаптации в течение жиз-

ни организма благодаря особому свойству нервной системы – пластичности. Масса мозга человека составляет примерно 5% от общей массы тела, при этом он потребляет около 20% энергии. Обычно мозг не задействован «на полную мощность»: одновременно в нем активны 2...3% нейронов. Благодаря такой избыточности мозг человека обладает огромным запасом прочности, позволяющим ему работать даже несмотря на серьезные повреждения. Известны случаи, когда люди продолжали сознательную жизнь после удаления большей части коры головного мозга. Современные компьютеры лишены этой способности.

Восприятие. Существующие компьютерные системы позволяют вводить в них изображения, звуковую и другие виды информации. Однако, в отличие от компьютера, снабженного сканером, телекамерой и микрофоном, мозг обладает вниманием – свойством, которое позволяет ему сосредотачиваться на важной в данный момент информации и игнорировать несущественную. Эффективные системы предобработки сенсорной информации, выработанные в течение жизни путем обучения, позволяют мозгу эффективно распознавать различные образы. Эта способность пока мало освоена современными компьютерами.

Память. Компьютеры имеют память, в которую можно записать программы, данные, изображения. Однако все они хранятся как некоторые именованные переменные. Специальные процедуры позволяют вычислить по имени переменной ее адрес в физической памяти, и именно по этому адресу будет извлекаться соответствующая запись. Никакой связи между адресом, по которому находятся данные, и содержанием самих данных не существует. Такая адресация предполагает пассивность данных в процессе поиска. Это обстоятельство чрезвычайно затрудняет поиск данных с частично известным содержанием.

Мозг использует другой способ поиска информации: не по адресу, а по содержанию, вернее, по его достаточно представительной части. Несколько нот известной мелодии позволяют восстановить недостающие фрагменты и вспомнить всю песню. Очевидно, что эти ноты являются лишь частью песни, в которой музыка вместе с текстом составляют единый информационный образ. Память, способная восстанавливать полную информацию по ее достаточной части, называется содержательно-адресованной. При этом мозг способен извлекать информацию и в случае, если исходные

данные (ключ) не являются ее частью, но связаны с ней устойчивой связью. Такая память называется в общем случае ассоциативной.

Еще одним свойством нашей памяти является ее распределенность. Это означает, что в мозге нет специализированного нейрона, отвечающего за распознавание того или иного образа. Наоборот, в запоминании некоторой информации участвует множество нейронов, так что разрушение некоторых из них обычно не удаляет соответствующий образ из памяти. Более того, мозг обладает огромной компенсаторной способностью: поражение обширных участков приводит к тому, что соответствующие функции берут на себя другие его части. Такое свойство системы называется робастностью (*robust* – крепкий, здоровый). Напротив, если испортить несколько бит в компьютерной программе, то это приведет к катастрофическим для нее последствиям.

Итак, человеческая память отличается от компьютерной тем, что она содержательно-адресованная, ассоциативная и робастная.

Мышление. Современные компьютеры способны решать задачи высокой вычислительной сложности: интегрировать сложнейшие системы дифференциальных уравнений, осуществлять логический вывод, играть в шахматы не хуже человека. Однако все эти задачи могут быть решены путем формализации – представления решения в виде алгоритма – последовательности арифметических и логических операций. Современный компьютер – это быстродействующий арифмометр, способный выполнить любую инструкцию, заключенную в программе. Структура компьютера является реализацией универсальной машины Тьюринга, в которой структура полностью отделена и независима от обрабатываемых данных. Американский математик Ричард Беллман заметил, что современные компьютеры по своей природе наиболее приспособлены для решения задач с начальными условиями (задачи Коши). Такие задачи могут быть решены последовательным получением решения от одного момента времени к другому.

Математические вычисления и логический вывод доступны и человеческому мозгу. Реализацию этих функций обычно связывают с одним, левым его полушарием. Однако этим мышление не исчерпывается. Работа левого полушария позволяет нам говорить, строить грамматически правильные фразы, писать. Оно ответственно за наше восприятие времени и обрабатывает информацию

последовательно, шаг за шагом. Между тем, математика, логика и наука в целом являются позднейшими достижениями человеческого мышления, т.е. развитие вычислительной техники началось с имитации наиболее поздних результатов эволюции человеческого мозга. Пропущенным оказался целый пласт его возможностей, реализованных в другом, правом, полушарии.

Многие исследования свидетельствуют в пользу того, что правое полушарие мозга ответственно за наше восприятие пространства, за смысл слов, интуицию, образное мышление. В отличие от левого оно обрабатывает информацию параллельным способом. Одной из основных его особенностей является то, что оно работает не с абстрактными именами объектов – символами, а с образами конкретных объектов, информацию о которых мозг получает из внешнего мира. Все виденное нами в жизни хранится в правом полушарии, все имена – в левом. Можно сказать, что понятие «автомобиль» известно левому полушарию, а вот образ конкретного «Запорожца» находится в правом.

Правое полушарие способно учиться узнавать предметы по их предъявлению, а не по описанию, в его способе функционирования данные и метод составляют единое целое. Способность обучаться на примерах объясняется в теории нейронных сетей пластичностью синаптических связей – их способностью менять свою силу, настраиваясь на решение определенной задачи. Образное мышление, воображение и интуиция, которые связывают с работой правого полушария, позволяют нам принимать решения в тех случаях, когда никакого рецепта не существует или он нам неизвестен. Эти качества являются более древними приобретениями мозга, чем логическое мышление, и напрямую связаны с творчеством.

Нейрокомпьютерный подход к обработке информации призван реализовать возможности, заложенные в правом полушарии мозга. Он должен не заменить существующие компьютерные методы, а лишь восполнить те возможности, для которых не удастся построить формальные алгоритмические схемы. Подобно тому, как в человеческом мозге левое и правое полушария работают сообща, современные информационные системы должны использовать симбиоз традиционных алгоритмических и нейрокомпьютерных методов для полноценной и продуктивной обработки информации (табл. 1.1).

Особенности обработки информации

Традиционная обработка информации в ЭВМ (условное левое полушарие)	Нейрокомпьютерная обработка информации (условное правое полушарие)
Последовательная обработка	Параллельная обработка
Заданный алгоритм обработки	Алгоритм формируется путем обучения на примерах
Иерархическая структура алгоритмов, разбиение сложной задачи на простые	Непосредственные операции с образами

2. ИСКУССТВЕННЫЙ НЕЙРОН

Искусственный нейрон (или просто нейрон) является элементарным функциональным модулем, из которых строятся ИНС. Он представляет собой модель биологического нейрона только лишь в смысле осуществляемых им преобразований, но не по способу функционирования. Известны логические, непрерывные и импульсные модели нейрона. Логические модели активно исследовались в 1960-70-х гг., 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 имитирует в первом приближении свойства биологического нейрона и включает в себя набор синапсов или связей, каждая из которых характеризуется своим весом (*weight*). В частности, сигнал x_i на входе синапса i умножается на вес w_i . Положительные значения весов w_i соответствуют возбуждающим синапсам, отрицательные – тормозным. Сумматор вычисляет взвешенную относительно соответствующих синапсов сумму входных сигналов нейрона. Функция активации ограничивает амплитуду выходного сигнала. Эта функция также называется функцией сжатия. Обычно нормализованный диапазон амплитуд выхода нейрона лежит в интервале $[0,1]$ или $[-1,1]$.

Математическую модель формального нейрона можно представить уравнением

$$S = \sum_{i=1}^n w_i x_i + w_0, \quad y = f(s) \quad (2.1)$$

где y – выходной сигнал нейрона; $f(s)$ – функция активации нейрона; w_i – весовой коэффициент синаптической связи i -го входа; x_i – i -й входной сигнал нейрона; w_0 – начальное состояние (возбуждение) нейрона; $i = 1, 2 \dots n$ – номера входов нейрона; n – число входов.

Выражению (2.1) может быть поставлена в соответствие структурная схема формального нейрона, представленная на рис. 2.1.

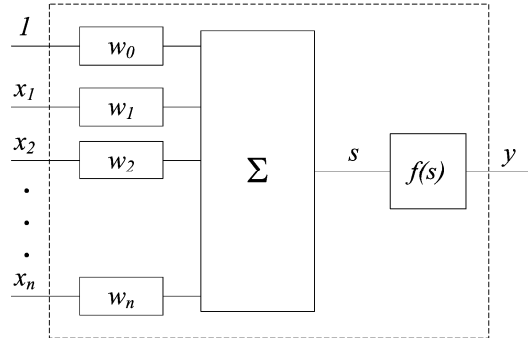


Рис. 2.1. Структурная схема формального нейрона

В общем случае работа формального нейрона заключается в следующем. Перед началом работы на блок сумматора подают сигнал начального состояния (возбуждения) нейрона – w_0 . На каждый i -й вход нейрона поступают сигналы x_i либо от других нейронов, либо с устройства ввода исходного образа. Каждый i -й входной сигнал x_i умножается на коэффициент синаптической связи w_i . В блоке сумматора взвешенные входные сигналы и начальное возбуждение w_0 алгебраически складываются. Результат суммирования (взвешенная сумма) s подается на блок функционального преобразования $f(s)$. Нейрон, схема которого приведена на рис. 2.1, является классическим. Вместе с тем, в некоторых случаях используются и другие модели формальных нейронов: нейроны с квадратичным сумматором, нейрон со счетчиком совпадений и др. Выбор модели определяется прежде всего характером решаемой задачи.

2.1. Функции активации

Активационная функция нейрона $f(s)$ определяет нелинейное преобразование, осуществляемое нейроном. Существует множество видов активационных функций, основные из них приведены в табл. 2.1.

Т а б л и ц а 2.1

Активационные функции нейронов

Название функции	Формула	Область значений
Линейная	$f(s) = ks$	$(-\infty, \infty)$
Полулинейная	$f(s) = \begin{cases} ks, s \geq 0 \\ 0, s < 0 \end{cases}$	$(0, \infty)$
Логистическая (сигмоидная)	$f(s) = \frac{1}{1 + e^{-as}}$	$(0, 1)$
Симметричная сигмоидная	$f(s) = \frac{e^{as} - e^{-as}}{e^{as} + e^{-as}}$	$(-1, 1)$
Рациональная сигмоидная	$f(s) = \frac{s}{a + s }$	$(-1, 1)$
Пороговая	$f(s) = \begin{cases} 1, s \geq \theta \\ 0, s < \theta \end{cases}$	$(0, 1)$
Сигнатурная	$f(s) = \begin{cases} 1, s \geq \theta \\ -1, s < \theta \end{cases}$	$(-1, 1)$

Наиболее распространенными функциями активации являются пороговая, линейная (в том числе с насыщением) и сигмоидные – логистическая и симметричная (гиперболический тангенс).

Пороговая и сигнатурная (симметричная пороговая) функции относятся к классу дискретных (рис. 2.2). Основным недостатком нейронов с пороговыми функциями активации – отсутствие достаточной гибкости при обучении и настройке ИНС на решаемую задачу. Если значение вычисляемого скалярного произведения s не достигает заданного порога, то выходной сигнал не формируется и нейрон “не срабатывает”. Это означает, что теряется интенсивность выходного сигнала данного нейрона и, следовательно, формируется невысокое значение уровня на

взвешенных входах в следующем слое нейронов. По этой причине пороговые функции обычно используются для представления нейросетями функций логического типа. Нейроны с пороговыми функциями активации используются также в сетях, обучаемых по методу конкуренции. Нейрон с наибольшим значением вычисляемого скалярного произведения s (победитель) получает право иметь на выходе значение $y=1$, формируемое пороговой функцией.

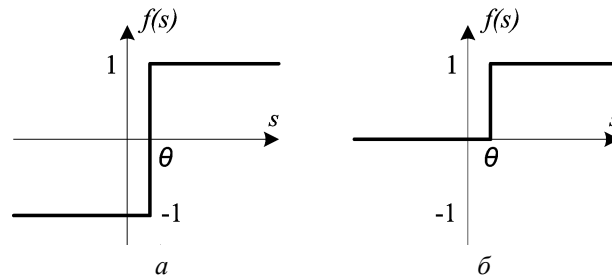


Рис. 2.2. Графики функций активации: a – сигнатурная, $б$ – пороговая

Сигмоидные функции активации относятся к классу непрерывных (рис. 2.3). Они удовлетворяют основным требованиям, предъявляемым к непрерывным активационным функциям нейронов, используемым в обучаемых ИНС: непрерывность, монотонное возрастание и дифференцируемость. Можно отметить, что с уменьшением коэффициента наклона a сигмоиды становятся более пологими, а при $a \rightarrow \infty$ превращаются соответственно в пороговую и сигнатурную функции.

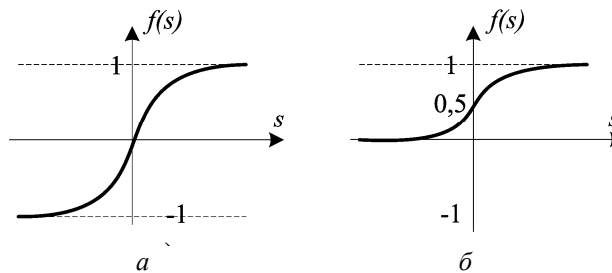


Рис. 2.3. Графики сигмоидных функций активации: a – симметричная, $б$ – логистическая

В числе их достоинств следует также упомянуть относительную простоту и непрерывность производных и свойство усиливать слабые сигналы лучше, чем большие. Рациональная или «упрощенная» сигмоида представляет интерес в силу простоты ее программной реализации. Нелинейность сигмоидных функций позволяет им выделять в поисковом пространстве области сложной формы, в том числе невыпуклые.

Линейные функции активации также относятся к классу непрерывных (рис. 2.4). Линейный участок такой функции позволяет оперировать непрерывными сигналами. Зоны нечувствительности определяются физической реализуемостью этих функций.

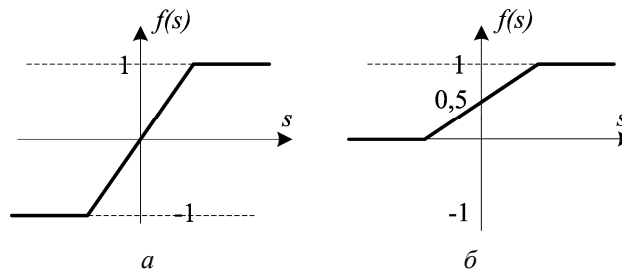


Рис. 2.4. Графики линейных функций активации с насыщением: *а* – линейная, *б* – полулинейная

Выбор функции активации определяется спецификой задачи, удобством реализации (на ЭВМ, в виде электрической схемы или другим способом) и алгоритмом обучения: некоторые алгоритмы накладывают ограничения на вид функции, которые нужно учитывать. Удачный выбор функции активации может сократить время обучения в несколько раз.

Формальный нейрон фактически представляет собой процессор с очень ограниченной специальной системой команд (нейросетевым базисом). По способу представления информации формальные нейроны делятся на аналоговые и цифровые. И те и другие выполняют единообразные вычислительные действия и не требуют внешнего управления. Большое число параллельно работающих процессоров обеспечивают высокое быстродействие.

Рассмотренная простая модель искусственного нейрона игнорирует многие свойства своего биологического двойника:

1. Вычисления выхода нейрона предполагаются мгновенными, не вносящими задержки. Моделировать динамические системы, имеющие "внутреннее состояние", с помощью таких нейронов нельзя.

2. В модели отсутствуют нервные импульсы. Нет модуляции уровня сигнала плотностью импульсов, как в нервной системе. Не проявляются эффекты синхронизации, когда скопления нейронов обрабатывают информацию синхронно, под управлением периодических волн возбуждения-торможения.

3. Нет четких алгоритмов для выбора функции активации.

4. Нет механизмов, регулирующих работу сети в целом (пример – гормональная регуляция активности в биологических нервных сетях).

5. Чрезмерно формализованы понятия: "порог", "весовые коэффициенты". В реальных нейронах нет числового порога, он динамически меняется в зависимости от активности нейрона и общего состояния сети. Весовые коэффициенты синапсов тоже непостоянны. "Живые" синапсы обладают пластичностью и стабильностью: весовые коэффициенты настраиваются в зависимости от сигналов, проходящих через синапс.

6. Существует большое разнообразие биологических синапсов. В различных частях клетки они выполняют различные функции. Тормозные и возбуждающие синапсы реализуются в данной модели в виде весовых коэффициентов противоположного знака, но разнообразие синапсов этим не ограничивается. Дендро-дендритные, аксо-аксональные синапсы не реализуются в модели формального нейрона.

Несмотря на эти ограничения, сети, построенные из формальных нейронов, обнаруживают свойства, сильно напоминающие биологическую систему.

2.2. Схема абстрактного нейрокомпьютера

Искусственной нейронной сетью, или просто нейронной сетью, называется динамическая система, состоящая из совокупности связанных между собой формальных нейронов, способная генерировать выходную информацию в ответ на входное воздействие. Связь формальных нейронов осуществляется по типу узлов направленного графа. Нейронная сеть является основной

операционной частью нейронных компьютеров (НК), реализуемых аппаратным или программным способом. Нейронная сеть реализует алгоритм решения поставленной задачи.

Структурная схема абстрактного НК, построенного из формальных нейронов, представлена на рис. 2.5. Такую схему можно назвать обобщенной, так как она поясняет принцип работы любого НК независимо от его конструктивного исполнения. Эта схема напоминает классическую схему вычислительной машины Дж. фон Неймана, представленную им еще в 1945 г. в отчете «Предварительный доклад о машине EDVAC». В нем приводится ставшее впоследствии классическим деление компьютера на АЛУ, УУ и память. Здесь же высказана идея программы, хранимой в памяти. Однако НК принципиально отличается от этой машины.

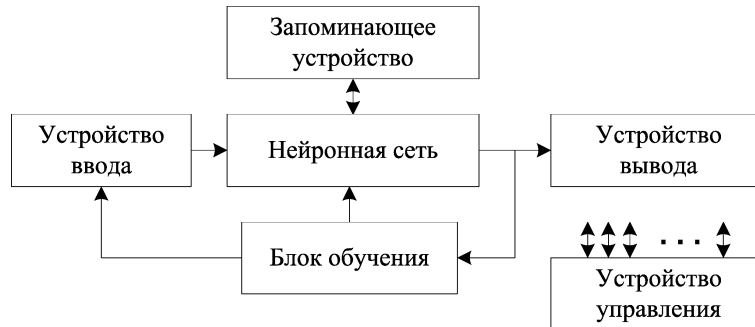


Рис. 2.5. Схема абстрактного НК

Основным операционным блоком НК, его процессором, является ИНС. Сеть представляет собой совокупность простейших модулей, называемых формальными нейронами, соединенными каналами передачи информации. Количественная характеристика каждого канала определяется решаемой задачей.

Нейронная сеть не производит вычислений, как это делает АЛУ машины фон Неймана. Она трансформирует входной сигнал (входной образ) в выходной в соответствии со своей топологией и значениями коэффициентов межнейронной связи. В запоминающем устройстве (ЗУ) НК хранится не программа решения задачи, как в ЭВМ, а программа изменения коэффициентов связи между нейронами. Устройства ввода и вывода информации в принципе выполняют те же функции. Устройство управления (УУ)

служит для синхронизации работы всех структурных блоков НК при решении конкретной задачи.

Для абстрактного НК характерно наличие двух главных режимов работы режима обучения и режима функционирования (решения задачи). Для того чтобы НК решил требуемую задачу, его ИНС должна пройти через режим обучения, суть которого заключается в настройке коэффициентов межнейронных связей на совокупность входных образов задачи. Настройка коэффициентов осуществляется на примерах, сгруппированных в обучающие множества (выборки).

При подаче на вход ИНС очередного эталонного образа выходной сигнал отличается от желаемого. Блок обучения оценивает величину ошибки и корректирует коэффициенты синаптических связей с целью ее уменьшения. При каждой последующей подаче на вход ИНС этого же эталонного входного образа ошибка уменьшается. Процесс продолжается до тех пор, пока ошибка не достигнет требуемого значения.

С математической точки зрения процесс обучения представляет собой решение задачи оптимизации, целью которой является минимизация функции ошибки на данном множестве примеров путем выбора коэффициентов синаптических связей.

В рабочем режиме блок обучения, как правило, отключен и на вход ИНС подаются сигналы, требующие распознавания. На эти сигналы (входные образы), как правило, наложен шум. Обученная ИНС фильтрует шум и относит образ к нужному классу.

Существенным недостатком ЭВМ является принципиально низкая производительность, обусловленная последовательным характером организации вычислительного процесса. С наличием одного АЛУ (процессора) связан другой недостаток – низкая эффективность использования памяти. Память ЭВМ можно представить как длинную последовательность ячеек. АЛУ выбирает содержимое одной из них, дешифрует, исполняет команду и при необходимости возвращает результат в заранее обусловленную ячейку памяти. Затем обращается к очередной ячейке для считывания следующей команды, и процесс повторяется до тех пор, пока не будет выбрана последняя команда исполняемой программы.

Нетрудно заметить, что подавляющее большинство ячеек памяти бездействует. Если ввести понятие коэффициента использования аппаратуры как отношение

$$\frac{\text{число}_{\text{одновременно}_{\text{используемых}_{\text{элементов}_{\text{ЭВМ}}}}}{\text{общее}_{\text{число}_{\text{элементов}_{\text{ЭВМ}}}}},$$

то для ЭВМ этот коэффициент будет очень низким. ЭВМ и НК различаются также по принципу взаимодействия структуры машины и решаемой задачи. Решая задачу на ЭВМ, разработчику приходится подстраивать алгоритм решения задачи под жесткую структуру машины. При использовании НК разработчик, наоборот, подстраивает структуру машины под решаемую задачу.

3. КЛАССИФИКАЦИЯ НЕЙРОСЕТЕВЫХ МОДЕЛЕЙ

Можно выделить три основных признака, положенных в основу классификации ИНС: архитектура межнейронных связей, тип обучения, класс решаемых задач. Архитектура нейронных сетей тесно связана с используемыми алгоритмами обучения и определяется постановкой задачи.

3.1. Классификация по типам связей между нейронами

ИНС может рассматриваться как направленный граф со взвешенными связями, в котором искусственные нейроны являются узлами. По архитектуре связей ИНС могут быть сгруппированы в два класса (рис. 3.1): сети прямого распространения, в которых графы не имеют петель, и рекуррентные сети, или сети с обратными связями.

Нейронные сети, называемые персептронами (от лат. perceptio – восприятие), представляют математическую модель процесса восприятия образов. Эта модель реализуется в виде слоев нейронов: рецепторного слоя и одного или нескольких слоев преобразующих нейронов.

К первому классу относятся многослойные сети прямого распространения, или ациклические сети. В этих сетях нейроны располагаются по слоям. В простейшем случае в такой сети существует входной слой узлов источника сигнала, информация от которого

передается на выходной слой нейронов (вычислительные узлы), но не наоборот.



Рис. 3.1. Классификация ИНС по видам межнейронных связей

На рис. 3.2 показана структура такой сети для случая трех узлов во входном слое и четырех в выходном. Такая нейронная сеть называется однослойной, при этом под единственным слоем подразумевается слой вычислительных элементов (нейронов). При подсчете числа слоев не принимаются во внимание узлы источника, так как они не выполняют никаких вычислений.

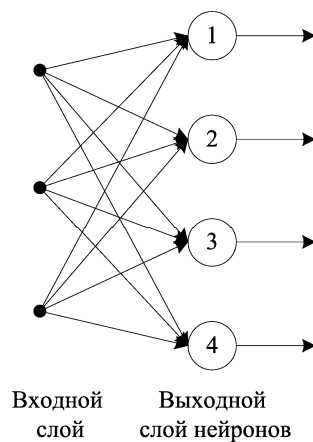


Рис. 3.2. Сеть прямого распространения с одним слоем

Другой класс нейронных сетей прямого распространения характеризуется наличием одного или нескольких скрытых слоев, узлы которых называются скрытыми нейронами. Функция последних заключается в дополнительной вычислительной обработке информации и посредничестве между внешним

входным сигналом и выходом нейронной сети. Добавляя скрытые слои, можно выделить статистические закономерности высокого порядка. Такая сеть позволяет выявлять глобальные свойства данных за счет наличия дополнительных синаптических связей и повышения уровня взаимодействия нейронов.

Узлы источника входного слоя сети формируют входной вектор образа, поступающий на нейроны второго слоя (первого скрытого слоя). Выходные сигналы второго слоя используются в качестве входных для третьего и т.д. Обычно нейроны каждого из слоев сети используют в качестве входных сигналов выходные сигналы нейронов только предыдущего слоя. Набор выходных сигналов выходного слоя определяет общий отклик сети на данный входной образ, сформированный узлами входного слоя. В общем случае сеть прямого распространения с m входами, h_1 нейронами первого скрытого слоя, h_2 нейронами второго скрытого слоя и q нейронами выходного слоя называется сетью $m-h_1-h_2-q$. На рис. 3.3 показана сеть 3-4-3 (3 входных узла, 4 скрытых и 3 выходных нейрона).

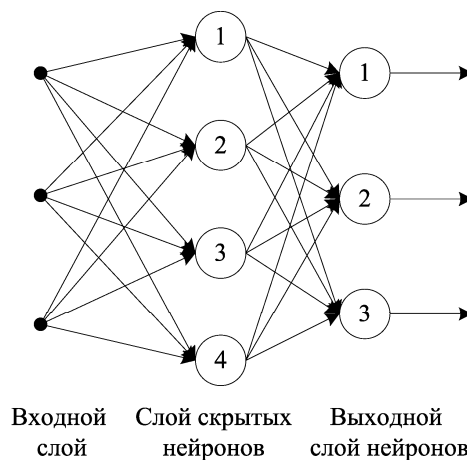


Рис. 3.3. Сеть прямого распространения с одним скрытым слоем

ИНС считается полносвязной в том смысле, что все узлы каждого слоя соединены со всеми узлами смежных слоев. Если некоторые из синаптических связей отсутствуют, такая сеть называется неполносвязной или частично связной. В таких прямонаправленных ИНС нейроны одного слоя связаны только с определенной

частью нейронов следующего слоя. Эта архитектура позволяет заложить в ИНС априорные знания о желаемом законе обработки сигналов в сети. Трехслойные прямонаправленные ИНС широко используются для решения задач классификации, распознавания образов, аппроксимации и управления.

Рекуррентная нейронная сеть отличается от сети прямого распространения наличием по крайней мере одной обратной связи. Рекуррентная сеть может состоять из единственного слоя нейронов, каждый из которых направляет свой выходной сигнал на входы всех остальных нейронов этого слоя (рис. 3.4). В приведенной структуре отсутствуют обратные связи нейронов с самими собой и скрытые нейроны. Обратная связь подразумевает использование элементов единичной задержки (обозначенные как z^{-1}), что приводит к нелинейному динамическому поведению. Наличие обратных связей в рекуррентных сетях непосредственно влияет на способность таких сетей к обучению и на их производительность.

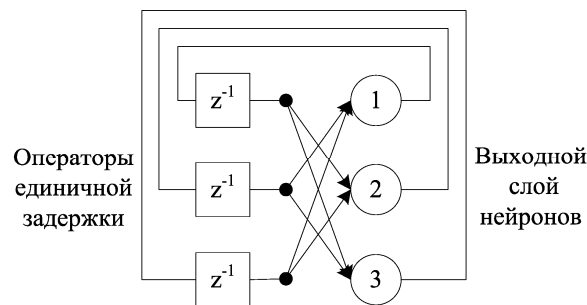


Рис. 3.4. Однослойная рекуррентная сеть без скрытых нейронов

Рекуррентные сети могут содержать слои скрытых нейронов (рис. 3.5). В этом случае каждый нейрон получает, кроме входных, еще и все выходные сигналы сети. Часть ИНС, охваченная обратными связями, может иметь и большее количество скрытых слоев.

Одним из главных применений рекуррентных ИНС являются нейроэмуляторы динамических объектов, т.е. их нейросетевые модели. Такие сети могут также использоваться для аппроксимации временных последовательностей, классификации, распознавания образов и управления.

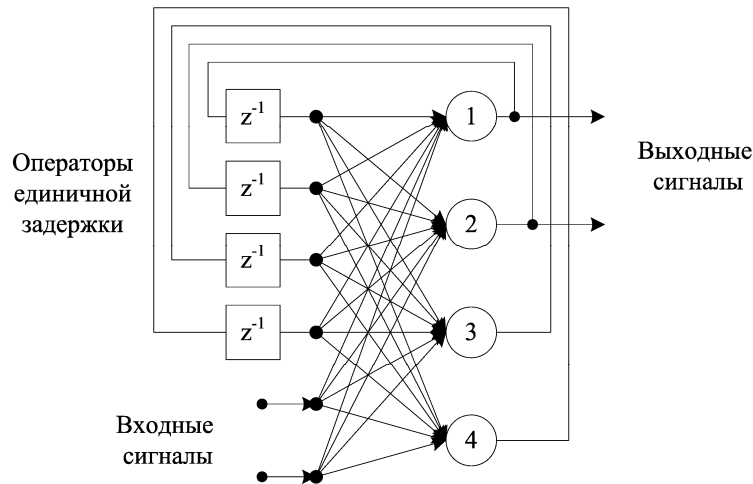


Рис. 3.5. Рекуррентная сеть со скрытыми нейронами

Сети прямого распространения являются статическими в том смысле, что на заданный вход они вырабатывают одну совокупность выходных значений, не зависящих от предыдущего состояния сети. Рекуррентные же сети являются динамическими, так как в силу обратных связей в них модифицируются входы нейронов, что приводит к изменению состояния сети.

3.2. Классификация по методам обучения

Способность к обучению является фундаментальным свойством мозга. Применительно к ИНС процесс обучения может рассматриваться как настройка архитектуры сети и весовых коэффициентов для эффективного выполнения специальной задачи. Обычно нейронная сеть должна настроить весовые коэффициенты по имеющейся обучающей выборке. Функционирование сети улучшается по мере выполнения итераций этой настройки. Свойство сетей обучаться на примерах делает их более привлекательными по сравнению с системами, которые следуют определенным правилам функционирования, сформулированным экспертами.

Для конструирования процесса обучения прежде всего необходимо иметь модель внешней среды, в которой функционирует нейронная сеть, т.е. знать доступную для сети информацию. Эта

модель определяет способ обучения. Во-вторых, необходимо понять, как модифицировать весовые параметры сети, т.е. какие правила обучения управляют процессом настройки. Алгоритм обучения означает процедуру, в которой используются правила обучения для настройки весов.

Существует множество определений термина «обучение», однако применительно к ИНС наиболее подходит следующее, данное Менделем и Маклареном [12]: *обучение — это процесс, при котором свободные параметры нейронной сети адаптируются в результате ее непрерывной стимуляции внешним окружением. Тип обучения определяется тем способом, которым производятся изменения параметров.*

Это определение предполагает следующую последовательность событий:

1. В нейронную сеть поступают стимулы из внешней среды.
2. В результате этого изменяются свободные параметры нейронной сети.
3. После изменения внутренней структуры нейронная сеть отвечает на возбуждения уже иным образом.

Кроме термина «обучение», используются равноправные понятия «тренировка сети» и «настройка параметров сети».

Можно выделить два основных вида обучения: контролируемое (supervised learning) и самообучение (self-organized learning). Первый вид подразумевает наличие «учителя», который наблюдает реакцию сети и направляет изменения ее параметров. Во втором случае сеть самоорганизуется под действием внешней среды и изучает ее самостоятельно, без помощи «учителя». Самообучение свойственно задачам категоризации (кластеризации). При решении задач классификации образов и управления обычно используется контролируемое обучение ИНС.

Существуют две разновидности контролируемого обучения: прямое и стимулированное (reinforcement learning). Так как первый вид появился раньше второго и более распространен, то обычно его называют просто «контролируемым обучением».

На рис. 3.6 представлена схема прямого контролируемого обучения. Присутствующий в ней «учитель» обладает знаниями о внешней среде, представленными в виде набора образцов вход-выход. Пара таких образцов называется тренировочным шаблоном, который включает в себя соответственно входной и выходной

шаблоны. Множество этих пар данных называется тренировочным набором шаблонов или обучающей выборкой.

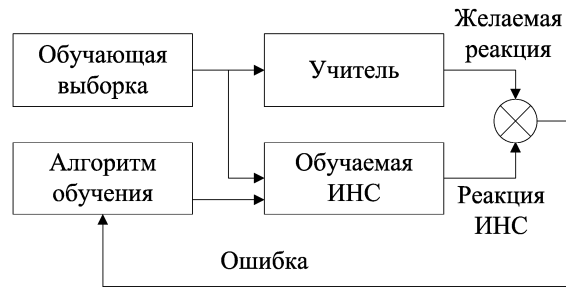


Рис. 3.6. Схема прямого контролируемого обучения

ИНС изначально никакими знаниями не обладает. В процессе обучения «учитель» и сеть подвергаются воздействию со стороны внешней среды, т.е. на их входы поступает тренировочный сигнал, совпадающий с одним из входных шаблонов. «Учитель» сообщает сети, какой должна быть правильная (желаемая) реакция на поступившее воздействие, выдавая соответствующий выходной шаблон. На основании величины ошибки между реальным и желаемым выходами сети по определенному правилу проводится настройка ее синаптических связей. Повторяя этот процесс итеративно, можно настроить ИНС так, что она будет эмулировать «учителя», т.е. его знания о внешней среде перейдут к ней.

Типичными задачами, решаемыми с помощью прямого обучения, являются аппроксимация неизвестной функции, описываемой набором данных, и классификация образов, т.е. отнесение образа к одному из заранее известных классов. В этих задачах известны входные сигналы и правильная реакция на них, т.е. тренировочный набор шаблонов. Наиболее известным методом прямого контролируемого обучения многослойных прямонаправленных ИНС (персептронов) является алгоритм обратного распространения ошибки (backpropagation algorithm), представляющий собой обобщение метода наименьших квадратов.

Стимулируемое обучение не использует знания «учителя» о желаемом выходе ИНС; вместо этого обучение проводится по результату оценки проводимого сетью преобразования вход-выход. Оценку выполняет внешнее окружение после подачи на вход сети

тренировочного воздействия. При этом настройка параметров ИНС проводится так, чтобы максимизировать скалярный индекс этой оценки, называемый стимулом (reinforcement signal).

Идея такого способа опирается на реальный процесс обучения, проходящий у живых существ. В психологии он известен как закон действия Торндайка. Применительно к стимулируемому обучению ИНС этот закон может быть перефразирован следующим образом [12]: *если действие, предпринятое системой обучения, приводит к удовлетворительному результату, то тенденция системы проводить это же действие увеличивается (система стимулируется). В противном случае тенденция производить такое действие уменьшается.*

Характерный пример стимулированного обучения – адаптивная система управления. В ней обучаемой частью является контроллер, а объект управления, внешние воздействия и сигналы задания выступают его внешней средой. В результате воздействия этой среды контроллер вырабатывает определенный управляющий сигнал, который переводит объект управления в новое состояние. При этом качество управления можно оценить лишь по выходному сигналу объекта. Так как требуемая реакция контроллера, обеспечивающая заданное состояние объекта, заранее не известна, то нельзя сформировать тренировочный набор шаблонов и, следовательно, применить прямое контролируемое обучение. В этом случае возможно лишь стимулированное обучение контроллера по качеству работы всей системы управления в целом, т.е. по оценке состояния внешней среды.

Самообучение (самоорганизация) также не использует знания «учителя» о желаемом выходе ИНС. Обучение проводится по реакции системы на тренировочное воздействие (рис. 3.7).

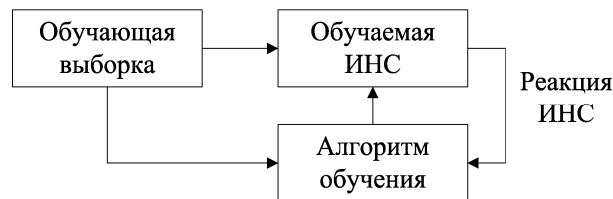


Рис. 3.7. Схема самообучения

Можно заметить, что система, обучающаяся лишь по результатам своего взаимодействия с внешней средой, является более интеллектуальной, чем использующая дополнительную информацию «учителя».

3.3. Классификация по решаемым задачам

Типовые задачи, решаемые в контексте ИНС и представляющие научный и практический интерес, можно подразделить следующим образом.

1. Классификация образов. Задача состоит в указании принадлежности входного образа (например, речевого сигнала или рукописного символа), представленного вектором признаков, одному или нескольким предварительно определенным классам. К известным приложениям относятся распознавание букв, распознавание речи, классификация сигнала электрокардиограммы, классификация клеток крови.

2. Кластеризация/категоризация. При решении задачи кластеризации, которая известна также как классификация образов «без учителя», отсутствует обучающая выборка с метками классов. Алгоритм кластеризации основан на подобию образов и размещает близкие образы в один кластер. Кластеризация применяется для извлечения знаний, сжатия данных и исследования их свойств.

3. Аппроксимация функций. Предположим, что имеется обучающая выборка $((x_1, y_1), (x_2, y_2), \dots, (x_n, y_n))$, т.е. пары данных вход-выход. Выборка генерируется неизвестной функцией $y(x)$, искаженной шумом. Задача аппроксимации состоит в нахождении оценки неизвестной функции $\hat{y}(x)$. Аппроксимация функций необходима при решении многочисленных инженерных и научных задач моделирования.

4. Предсказание/прогноз. Пусть заданы n дискретных отсчетов $\{y(t_1), y(t_2), \dots, y(t_n)\}$ в последовательные моменты времени t_1, t_2, \dots, t_n . Задача состоит в предсказании значения $y(t_n+1)$ в некоторый будущий момент времени t_n+1 . Прогнозирование играет важную роль при принятии решений в бизнесе, науке и технике. Предсказание цен на фондовой бирже и прогноз погоды являются типичными приложениями техники предсказания/прогноза.

5. Оптимизация. Многочисленные проблемы в математике, статистике, технике, науке, медицине и экономике могут рассмат-

риваться как проблемы оптимизации. Задачей алгоритма оптимизации является нахождение такого решения, которое удовлетворяет системе ограничений и максимизирует или минимизирует целевую функцию. Задача коммивояжера, относящаяся к классу NP -полных, является классическим примером задачи оптимизации.

6. Память, адресуемая по содержанию. В модели вычислений фон Неймана обращение к памяти доступно только посредством адреса, который не зависит от содержания памяти. Более того, если допущена ошибка в вычислении адреса, то может быть найдена совершенно иная информация. Ассоциативная память, или память, адресуемая по содержанию, доступна по указанию заданного содержания. Содержимое памяти может быть вызвано даже по частичному входу или искаженному содержанию. Использование ассоциативной памяти крайне желательно при создании мультимедийных информационных баз данных.

7. Управление. В системах управления с эталонной моделью целью управления является расчет такого входного воздействия $u(t)$, при котором система следует по желаемой траектории, диктуемой эталонной моделью. Примером является оптимальное управление двигателем.

3.4. Классификация по способу решения задачи

Все задачи, которые решаются с помощью средств вычислительной техники, удобно разделить на три класса: формализуемые, трудноформализуемые и неформализуемые.

Формализуемая задача имеет четко сформулированный алгоритм решения. Яркий пример – вычислительные задачи: решение алгебраических, дифференциальных, интегральных и других уравнений, сортировка данных и т.п. Для решения задач этого класса в вычислительной математике разработано много эффективных методов, ориентированных именно на применение ЭВМ.

Трудноформализуемая задача имеет алгоритм решения, качество которого, или достижимость решения, трудно оценить. Этот класс включает задачи моделирования систем большой размерности. К ним можно отнести моделирование сложных электронных устройств в системах автоматизированного проектирования, интегрированную подготовку производства и т.п.

Неформализуемая задача содержит в своей постановке неявно заданные функции и параметры. К этому классу задач относят распознавание образов, кластеризацию данных, прогнозирование и т.п.

ИНС используется в основном для решения неформализуемых задач. Однако развитие теории и элементной базы позволило применить ИНС для решения и формализуемых и трудноформализуемых задач.

По способу решения задач ИНС делят на формируемые сети, сети с формируемой матрицей связи, обучаемые сети и комбинированные.

Формируемые сети проектируют для формализуемых задач, имеющих четко сформулированный в нейросетевом базисе алгоритм решения конкретной задачи. Сети с формируемой матрицей связей применяются для трудноформализуемых задач. Как правило, они имеют одинаковую структуру и различаются лишь матрицей связи (например, сети Хопфилда). Обучаемые сети используют для решения неформализуемых задач. В процессе обучения они автоматически изменяют свои свободные параметры – коэффициенты синаптических связей, а в некоторых случаях и топологию. Ключевой задачей разработки таких сетей является выбор или разработка алгоритма обучения. Комбинированные сети сочетают в себе признаки предыдущих видов. Как правило, это многослойные сети, каждый слой которых представляется различной топологией и обучается по определенному алгоритму.

4. ПЕРСЕПТРОНЫ

Стремясь воспроизвести функции человеческого мозга, первые исследователи ИНС в 1940-е гг. создали простые аппаратные модели биологического нейрона и системы его соединений. Эти модели были достаточно грубыми аппроксимациями. Тем не менее с их помощью были получены результаты, стимулировавшие дальнейшие исследования, приведшие к созданию более сложных сетей.

4.1. Формальный нейрон Маккаллока–Питтса

Первое систематическое изучение ИНС было предпринято У.Маккаллоком и У.Питтсом в 1943 г. У.Маккаллоком, по образова-

нию психиатр и нейроанатом, более 20 лет занимался вопросами представления событий в нервной системе. У.Питтс, талантливый математик, присоединился к исследованиям У. Маккаллока в 1942 г. В своей новаторской работе они описали результаты, полученные группой специалистов из университета Чикаго, и сформулировали основные положения теории деятельности головного мозга. Ими была разработана модель нейрона как простейшего процессорного элемента, выполняющего вычисление функции от скалярного произведения вектора входных сигналов и вектора весовых коэффициентов; предложена конструкция сети таких элементов для выполнения логических и арифметических операций. Они сделали основополагающее предположение о том, что такая сеть способна обучаться, распознавать образы, обобщать полученную информацию, предложили вводить для возбуждающих связей положительные весовые коэффициенты, а для тормозных – отрицательные.

Простая нейронная модель (рис. 4.1) использовалась в большей части их работы. Значения x_i , поступающие на каждый вход нейрона, умножаются на соответствующий вес w_i и суммируются. Если эта сумма больше заданного порогового значения, выход y равен единице, в противном случае – нулю (или -1):

$$s = \sum_{i=1}^n x_i w_i, \quad y = \text{sgn}(s) = \begin{cases} +1, & s \geq 0, \\ 0, & s < 0. \end{cases}$$

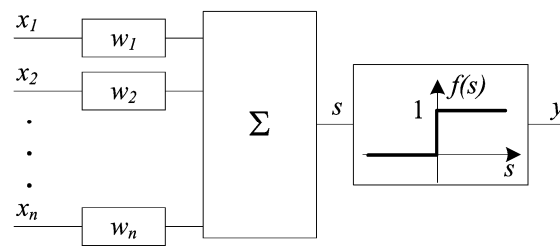


Рис. 4.1. Модель нейрона по Маккаллоку и Питтсу

Такие сети получили название однослойных персептронов. Они состоят из одного слоя искусственных нейронов, соединенных с помощью весовых коэффициентов с множеством входов. Каждый нейрон в сети определяет взвешенную сумму состояний

всех входов и сравнивает ее с порогом, чтобы определить свое собственное состояние. Пороговый вид функции не обеспечивает нейронной сети достаточную гибкость при обучении и настройке на заданную задачу.

Способность формального нейрона к обучению проявляется в возможности изменять значения вектора весов \bar{W} по аналогии с пластичностью синапсов биологических нейронов. Рассмотрим обучение формального нейрона на примере простейшей задачи детектирования границы. Пусть имеется образ, составленный из одномерной цепочки черных и белых клеток. Черные клетки соответствуют единичному сигналу, а белые клетки – нулевому. Сигнал на входах формального нейрона устанавливается равным значениям пар примыкающих клеток рассматриваемого образа. Нейрон обучается всякий раз возбуждаться и выдавать единичный выходной сигнал, если его первый вход x_1 соединен с белой клеткой, а второй x_2 – с черной (рис. 4.2). Таким образом, нейрон должен служить детектором границы перехода от светлого к темному тону образа.

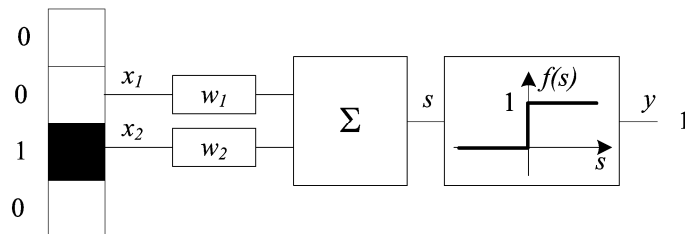


Рис. 4.2. Формальный нейрон с двумя входами, предназначенный для обработки образа в виде одномерной цепочки черных и белых клеток

Функция, выполняемая нейроном для детектирования перехода от светлого к темному ($x_1 = 0, x_2 = 1$), определяется табл. 4.1.

Т а б л и ц а 4.1

Вход 1	Вход 2	Выход
1	1	0
1	0	0
0	1	1
0	0	0

Для данной задачи значения весов и порога нейрона могут быть предъявлены и без специальной процедуры обучения. Легко убедиться, что нужным требованиям удовлетворяет набор $w_1 = -1$, $w_2 = +1$. В задаче детектирования границы перехода от темного к светлому веса нужно поменять местами.

4.2. Персептрон Розенблатта

Простейший классический персептрон Ф.Розенблатта содержит нейроподобные элементы трех типов (рис. 4.3), назначение которых в целом соответствует нейронам рефлекторной нейронной сети. *S*-элементы формируют сетчатку сенсорных клеток, принимающих двоичные сигналы от внешнего мира. Далее сигналы поступают в слой ассоциативных или *A*-элементов (для упрощения изображения часть связей от входных *S*-элементов к *A*-нейронам не показана). Только ассоциативные элементы, представляющие собой формальные нейроны, выполняют нелинейную обработку информации и имеют изменяемые веса связей. *R*-элементы с фиксированными весами формируют сигнал реакции персептрона на входной стимул (связи между *A*-нейронами и *R*-элементами также показаны частично).

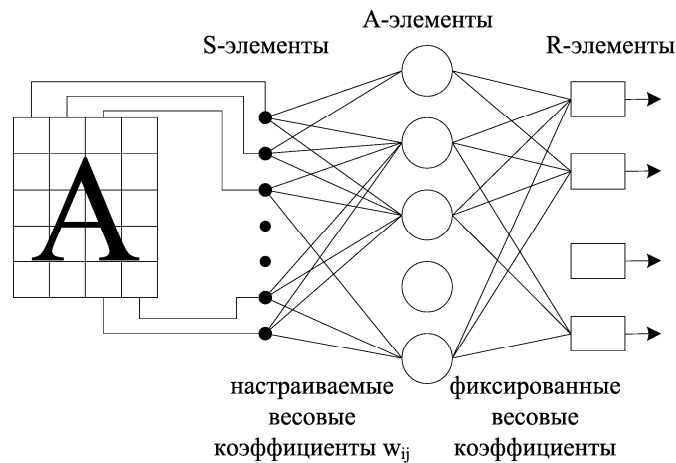


Рис. 4.3. Элементарный персептрон Ф. Розенблатта

В 1960-е гг. в Корнельской авиационной лаборатории была разработана электротехническая модель персептрона MARK-1, которая содержала 8 выходных R -элементов и 512 A -элементов, которые можно было соединять в различных комбинациях. На этом персептроне была проведена серия экспериментов по распознаванию букв алфавита и геометрических образов. Персептрон рассматривался не как конкретное техническое вычислительное устройство, а как модель работы мозга.

Розенблатт называл такую нейронную сеть трехслойной, однако в современной терминологии она обычно называется однослойной, так как имеет только один слой нейропроцессорных элементов. Однослойный персептрон характеризуется матрицей синоптических связей W от S - к A -элементам. Элемент матрицы w_{ij} отвечает связи, ведущей от i -го S -элемента к j -му A -элементу.

Ф. Розенблатт ввел в рассмотрение понятия представляемости и обучаемости персептрона.

Представляемость персептрона. Под представляемостью понимается способность персептрона или другой ИНС моделировать (представлять) определенную функцию. Обучаемость же требует наличия систематической процедуры настройки весов сети для реализации этой функции.

Проблему представляемости можно проиллюстрировать на следующем примере. Пусть у нас есть множество карт, помеченных цифрами от 0 до 9. Допустим, мы обладаем гипотетической машиной, способной отличать карты с нечетным номером от карт с четным и зажигающей индикатор на своей панели при предъявлении карты с нечетным номером (рис. 4.4).

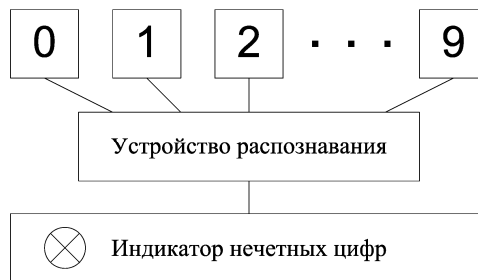


Рис. 4.4. Устройство распознавания четных и нечетных цифр

Представима ли такая машина персептроном? То есть может ли быть сконструирован персептрон и настроены его веса (неважно каким образом) так, чтобы он обладал такой же разделяющей способностью? Если это так, то говорят, что персептрон способен представлять желаемую машину.

Обучаемость персептрона. Для обучения персептрона Ф. Розенблатт предложил процедуру настройки весовых коэффициентов каждого нейрона. Пусть имеется набор пар векторов (x_q, y_q) , $q = 1 \dots p$, называемый обучающей выборкой. Будем называть нейронную сеть обученной на данной обучающей выборке, если при подаче на входы сети каждого вектора x_q на выходах всякий раз получается соответствующий вектор y_q .

Предложенный метод обучения состоит в итерационной подстройке матрицы весов, последовательно уменьшающей ошибку в выходных векторах. Алгоритм включает следующие шаги:

1. Начальные значения весов всех нейронов $W(t=0)$ полагаются случайными, где t – номер итерации обучения сети.

2. Сети предъявляется входной образ x_q . В результате формируется выходной образ $\tilde{y}_q \neq y_q$.

3. Вычисляется вектор ошибки $\delta_q = (\tilde{y}_q - y_q)$ на выходе сети. Предполагается, что изменение вектора весовых коэффициентов в области малых ошибок должно быть пропорционально ошибке на выходе и равно нулю, если ошибка равна нулю.

4. Матрица весов модифицируется по формуле

$$W(t+1) = W(t) + \eta x_q (\delta_q)^T, \quad (4.1)$$

где η – темп обучения, причем $0 < \eta < 1$.

5. Шаги 2-4 повторяются для всех обучающих векторов. Один цикл последовательного предъявления всей выборки называется эпохой обучения. Процедура настройки весов завершается по истечении нескольких эпох, когда итерации сойдутся, т.е. матрица весов перестает изменяться, когда просуммированная по всем векторам квадратичная ошибка на выходе сети станет меньше некоторого допустимого малого значения.

Использование формулы (4.1) определяет следующие особенности процесса обучения персептрона:

1) модифицируются только компоненты матрицы весов, отвечающие ненулевым значениям входов x_q ;

2) знак приращения веса соответствует знаку ошибки, т.е. положительная ошибка $\delta_q > 0$, когда значение выхода меньше требуемого значения, приводит к усилению связи;

3) обучение каждого нейрона происходит независимо от обучения остальных нейронов, что соответствует важному с биологической точки зрения принципу локальности обучения.

Данный метод обучения был назван Ф. Розенблаттом «методом коррекции с обратной передачей сигнала ошибки». Позднее более широко стало применяться название « δ -правило». Представленный алгоритм относится к широкому классу алгоритмов обучения с учителем, поскольку известны как входные векторы, так и требуемые значения выходных (имеется учитель, способный оценить правильность ответа ученика).

Теорема о сходимости обучения по « δ -правилу» говорит о том, что персептрон способен обучиться любому обучающему набору, который он может представить. В работах Ф. Розенблатта было сделано заключение о том, что нейронная сеть рассмотренной им архитектуры способна к воспроизведению любой логической функции. Однако, как было показано позднее, этот вывод оказался неточным. Были выявлены принципиальные неустраняемые ограничения однослойных персептронов, не позволяющие им представить многие простые логические функции.

С сегодняшних позиций однослойный персептрон представляет скорее исторический интерес, но на его примере могут быть изучены основные понятия и простые алгоритмы обучения нейронных сетей.

4.3. Проблема обучаемости персептронов

Функция «исключающее ИЛИ». Известно, что однослойный персептрон не может воспроизвести такую простую функцию, как «исключающее ИЛИ» (XOR). Это функция от двух аргументов, таблица истинности которой приведена в табл. 4.1. Проблему можно проиллюстрировать с помощью однослойной однеуронной сети с двумя входами (см. рис. 4.5). Обозначим один вход через x_1 , а другой через x_2 . Тогда все их возможные комбинации будут состоять из четырех точек на плоскости аргументов $x_1 - x_2$. Соответствие между точками на плоскости $x_1 - x_2$ и значениями функ-

ции XOR приведены в табл. 4.1. Сочетания значений аргументов, которые должны давать нулевой выход, помечены A_0 и A_1 , единичный выход – B_0 и B_1 .

Т а б л и ц а 4.1

Таблица истинности для функции XOR

Номер примера	Точки на плоскости x_1-x_2	Вход x_1	Вход x_2	Желаемый выход y	Условие превышения порога θ
1	A_0	0	0	0	$w_1*0+w_2*0 < \theta$
2	B_0	1	0	1	$w_1*1+w_2*0 \geq \theta$
3	A_1	0	1	1	$w_1*0+w_2*1 \geq \theta$
4	B_1	1	1	0	$w_1*1+w_2*1 < \theta$

В сети на рис. 4.5 функция $f(s)$ является обычным порогом, так что y принимает значение «ноль», когда s меньше θ , и «единица», когда s больше или равно θ . Нейрон выполняет следующее вычисление:

$$s = x_1w_1 + x_2w_2. \quad (4.2)$$

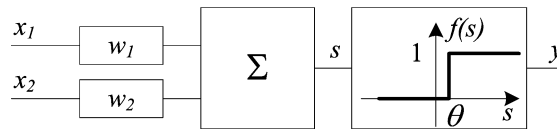


Рис. 4.5. Персептронная сеть

Никакая комбинация значений двух весов не может дать соотношения между входом и выходом, задаваемого табл. 4.1. Чтобы понять это ограничение, достаточно попарно сложить неравенства из этой таблицы. Первое и четвертое неравенства дают условие: $w_1+w_2 < 2\theta$, второе и третье – $w_1+w_2 \geq 2\theta$. Несовместимость этих двух условий не позволяет реализовать функцию XOR путем настройки коэффициентов w_1 , w_2 и выбором порога θ .

Иначе можно представить значения s как поверхность (в данном случае плоскость) над плоскостью аргументов функции x_1-x_2 . Каждая точка этой s -поверхности находится над соответствующей точкой плоскости x_1-x_2 на расстоянии, равном значению s в этой точке. Наклон s -плоскости постоянен и определяется значениями

весов w_1, w_2 . Все точки, в которых значение s равно величине порога $s = \theta$, лежат на линии констант, образованной пересечением плоскостей $s = x_1w_1 + x_2w_2$ и $s = \theta$ (рис. 4.6). Проекция линии констант на плоскость x_1 - x_2 образует разделяющую прямую (рис. 4.7).

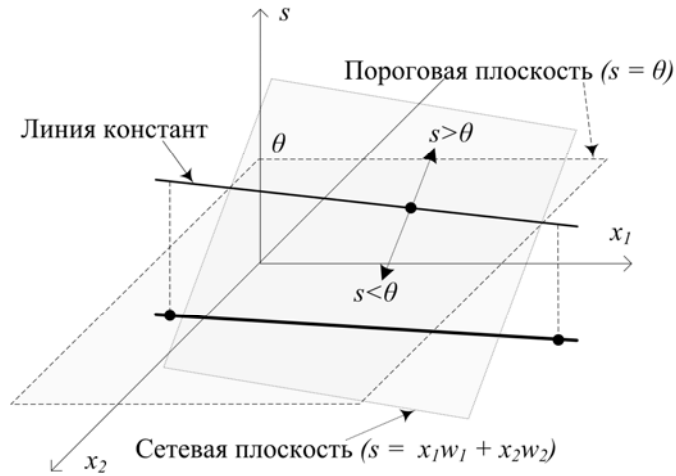


Рис. 4.6. Персептронные поверхности

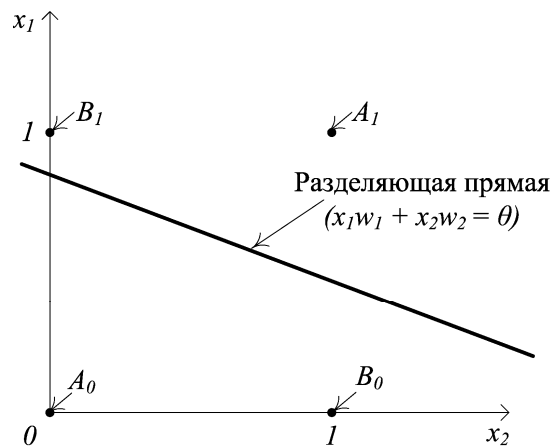


Рис. 4.7. Проблема представимости XOR

Ясно, что все точки по одну сторону линии констант спроецируются в значения s , превышающие θ , а точки по другую сторону – меньшие θ . Таким образом, разделяющая прямая разбивает плоскость x_1 - x_2 на две области. Во всех точках по одну сторону разделяющей прямой $y = 1$, по другую – $y = 0$. Входные значения x_1 и x_2 с одной стороны прямой обеспечат значения s больше порога, следовательно, $y = 1$. Входные значения по другую сторону прямой обеспечат значения меньше порогового, т.е. $y = 0$. Изменения значений w_1 , w_2 и θ будут менять наклон и положение прямой. Для того чтобы сеть реализовала функцию XOR, заданную табл. 4.1, нужно расположить прямую так, чтобы точки A_i были с одной ее стороны, а точки B_i – с другой. Попытавшись провести такую прямую на рис. 4.7, убеждаемся, что это невозможно. Следовательно, какие бы значения ни принимали веса w_i и порог θ , сеть не способна воспроизвести соотношение между входом и выходом, требуемое для представления функции XOR.

Линейная разделимость. XOR не является единственным примером функции, которую не способен представить однослойный персептрон. Имеется обширный класс функций, не реализуемых однослойной сетью. Об этих функциях говорят, что они являются линейно неразделимыми, и они накладывают определенные ограничения на возможности однослойных сетей.

Линейная разделимость ограничивает однослойные сети задачами классификации, в которых множества точек (соответствующих входным значениям) могут быть разделены геометрически. Для случая с двумя входами разделитель является прямой линией. В случае трех входов разделение осуществляется плоскостью, пересекающей трехмерное пространство. Для четырех или более входов визуализация невозможна, и необходимо мысленно представить n -мерное пространство, разделяемое «гиперплоскостью» – геометрическим объектом, который рассекает пространство четырех или большего числа измерений.

Так как линейная разделимость ограничивает возможности персептронного представления, то важно знать, является ли данная функция разделимой. К сожалению, не существует простого способа определить это, если число переменных велико.

Нейрон с n двоичными входами может иметь 2^n различных входных образов, состоящих из нулей и единиц. Так как каждый входной образ может соответствовать двум различным бинарным

выходам (единица и ноль), то всего имеется 2^{2^n} функций от n переменных.

Как видно из табл. 4.2, вероятность того, что случайно выбранная функция окажется линейно разделимой, весьма мала даже для малого числа переменных. По этой причине применение однослойных персептронов на практике ограничено простыми задачами.

Т а б л и ц а 4.2

Линейно разделимые функции

Число двоичных входов n	Число функций от n переменных 2^{2^n}	Число линейно разделимых функций
1	4	4
2	16	14
3	256	104
4	65536	1882

Преодоление ограничения линейной разделимости. Способ преодоления этого серьезного ограничения представляемости логических функций однослойными персептронами заключается в добавлении дополнительных слоев нейронов. Например, двухслойные сети можно получить каскадным соединением двух однослойных сетей. На рис. 4.8 показана двухслойная сеть, способная представить функцию XOR.

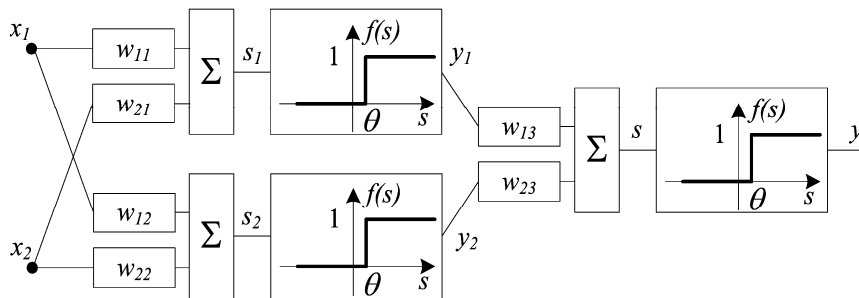


Рис. 4.8. Двухслойная сеть, реализующая функцию XOR.

Допустим, что весовые коэффициенты сети имеют значения $w_{11}=w_{22}=w_{13}=w_{23}=w$, $w_{21}=w_{12}=-w$, а значения порогов $\theta = w$. Тогда таблицу истинности для XOR можно представить в виде табл. 4.3.

Т а б л и ц а 4.3

Таблица истинности для реализации функции XOR

x_1	x_2	s_1	s_2	y_1	y_2	s	y
0	0	0	0	0	0	0	0
0	1	$-w$	w	0	1	w	1
1	0	w	$-w$	1	0	w	1
1	1	0	0	0	0	0	0

В общем случае, для того чтобы порог θ был превышен и на выходе y нейрона второго слоя появилась единица, требуется, чтобы оба нейрона первого слоя на выходе имели единицу. Таким образом, выходной нейрон реализует логическую функцию «И». На рис. 4.9 каждый нейрон первого слоя разбивает плоскость x_1 - x_2 на две полуплоскости. Один обеспечивает единичный выход для входов ниже верхней линии, другой – для входов выше нижней линии.

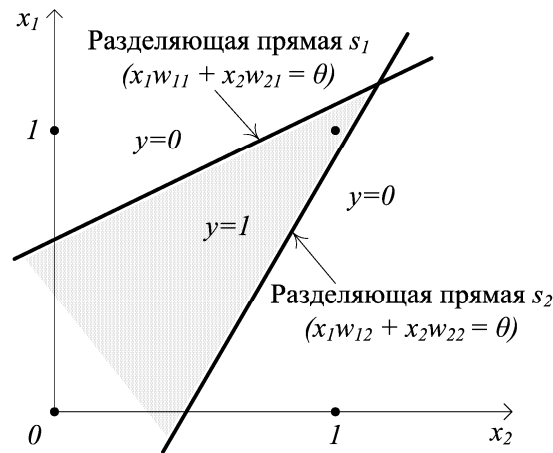


Рис. 4.9. Выпуклая область решений, задаваемая двухслойной сетью

На рис. 4.9 показан результат такого двойного разбиения, где выходной сигнал нейрона второго слоя y равен единице только внутри V-образной области. Аналогично во входном слое может быть использовано три нейрона с дальнейшим разбиением плоскости и созданием области треугольной формы. Включением достаточного числа нейронов во входной слой может быть образован выпуклый многоугольник любой желаемой формы. Так как они образованы с помощью операции «И» над областями, задаваемыми линиями, то все эти многоугольники выпуклые.

Нейрон второго слоя не ограничен функцией «И». Он может реализовывать многие другие функции при подходящем выборе весов и порога. Например, можно сделать так, чтобы единичный выход любого из нейронов первого слоя приводил к появлению единицы на выходе нейрона второго слоя, реализовав тем самым логическое «ИЛИ».

Входы персептрона не обязательно должны быть двоичными. Вектор непрерывных входов может представлять собой произвольную точку на плоскости x_1 - x_2 . В этом случае речь идет о способности сети разбивать плоскость на непрерывные области, а не о разделении дискретных множеств точек. Для всех этих функций линейная разделимость показывает, что выход нейрона второго слоя равен единице только в части плоскости x_1 - x_2 , ограниченной выпуклой многоугольной областью. Поэтому для разделения плоскостей P и Q необходимо, чтобы все точки P лежали внутри выпуклой многоугольной области, не содержащей точек Q (или наоборот). Точки, не образующие выпуклые области, не могут быть отделены от других точек плоскости двухслойной сетью.

Область называется выпуклой, если для любых двух ее точек соединяющий их отрезок целиком лежит в ней. Область называется ограниченной, если ее можно заключить в некоторый круг. Неограниченную область невозможно заключить внутри круга (например, область между двумя параллельными линиями). Примеры выпуклых ограниченных и неограниченных областей приведены на рис. 4.10.

Таким образом, двухслойные персептроны способны решать более широкий класс задач классификации, отделяя те точки, которые содержатся в выпуклых ограниченных или неограниченных областях.

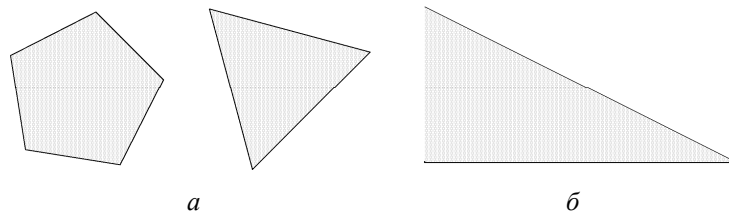


Рис. 4.10. Выпуклые ограниченные (а) и неограниченные (б) области

Трехслойные персептроны могут разделять точки, образующие области любой формы, как выпуклые, так и невыпуклые. Их возможности классификации образов ограничены лишь числом искусственных нейронов и весов. Нейрон третьего слоя принимает в качестве входа набор выпуклых многоугольников, и их логическая комбинация может быть невыпуклой. На рис. 4.11 иллюстрируется случай, когда две треугольные области A и B , скомбинированные с помощью функций « A и не B », задают невыпуклую область.



Рис. 4.11. «Вогнутая» область решений, формируемая трехслойной сетью

При добавлении нейронов и весов число сторон многоугольников может неограниченно возрастать. Это позволяет аппроксимировать область любой формы с любой точностью. Вдобавок не все выходные области второго слоя должны пересекаться. Следовательно, можно объединять различные области, выдавая на выходе единицу всякий раз, когда входной вектор принадлежит одной из них.

Несмотря на то что возможности многослойных сетей были известны, в течение многих лет отсутствовал теоретически обоснованный алгоритм для настройки их весовых коэффициентов. Лишь в середине 1980-х годов несколькими исследователями, независимо друг от друга, был предложен эффективный метод обучения многослойных персептронов, основанный на вычислении градиента функции ошибки и названный "обратным распространением ошибки".

5. ОБУЧЕНИЕ МЕТОДОМ ОБРАТНОГО РАСПРОСТРАНЕНИЯ ОШИБКИ

5.1. Процедура обучения

Среди различных структур ИНС одной из наиболее известных является многослойная сеть, состоящая из множества сенсорных элементов (входных узлов или узлов источника), одного или нескольких скрытых слоев (hidden layer) вычислительных нейронов и одного выходного слоя (output layer) нейронов. В этих сетях каждый нейрон скрытого слоя связан со всеми нейронами следующего слоя, а входной сигнал распространяется в прямом направлении, от слоя к слою. Такие сети называются многослойными персептронами (рис. 5.1), которые отличаются тремя признаками:

1) каждый нейрон имеет нелинейную функцию активации. Эта функция является гладкой (т.е. всюду дифференцируемой), в отличие от пороговой функции, используемой в персептроне Розенблатта. Чаще всего используется логистическая функция, удовлетворяющая этому требованию. Наличие нелинейности играет очень важную роль, так как в противном случае сеть можно свести к обычному однослойному персептрону;

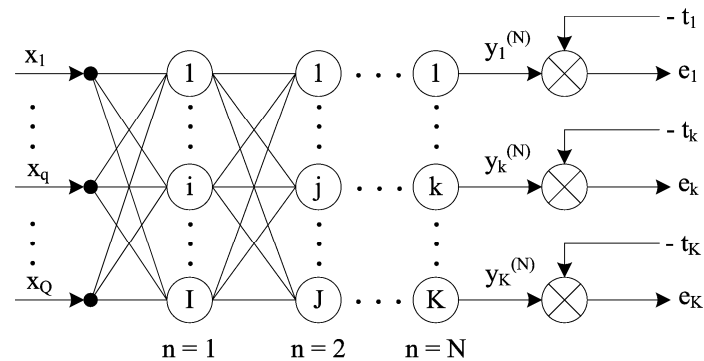


Рис. 5.1. Структура обучаемой сети

2) сеть содержит один или несколько слоев скрытых нейронов, не являющихся частью входа или выхода сети. Эти нейроны позволяют сети обучаться решению сложных задач, последовательно извлекая наиболее важные признаки из входного образа (вектора);

3) сеть обладает высокой степенью связности, реализуемой посредством синаптических соединений.

Под обучением многослойного персептрона понимается процесс адаптации сети к предъявленным эталонным образцам путем модификации весовых коэффициентов связей между нейронами. В однослойной сети алгоритм обучения с учителем очевиден, так как желаемые выходные состояния нейронов единственного выходного слоя t_k заведомо известны. Подстройка синаптических связей идет в направлении минимизации ошибки E на выходе сети. В многослойных сетях желаемое значение выходов нейронов всех внутренних слоев, кроме последнего, неизвестны. Поэтому многослойный персептрон уже невозможно обучить, руководствуясь только величинами ошибок на выходе ИНС.

Обучение многослойного персептрона методом обратного распространения ошибки предполагает два прохода по всем слоям сети: прямой и обратный. При прямом проходе образ (входной вектор) подается на сенсорные узлы сети, после чего распространяется от слоя к слою. В результате генерируется набор выходных сигналов, который и является фактической реакцией сети на данный входной образ. Во время прямого прохода все синаптические веса сети фиксированы. Во время обратного прохода они настраиваются в соответствии с правилом коррекции ошибок, а именно: фактический выход сети вычитается из желаемого отклика, в результате чего формируется сигнал ошибки. Этот сигнал впоследствии распространяется по сети в направлении, обратном направлению синаптических связей. Отсюда и название «алгоритм обратного распространения ошибки», упрощенно – «алгоритм обратного распространения».

Для многослойного персептрона выделяют два типа сигналов:

1) функциональный. Это входной сигнал (стимул), поступающий в сеть и передаваемый в прямом направлении от нейрона к нейрону. Такой сигнал достигает конца сети в виде выходного сигнала, т.е. сеть реализует некоторую функциональную зависимость выхода от стимула;

2) сигнал ошибки. Он берет свое начало на выходе сети и распространяется в обратном направлении от слоя к слою. Свое название получил благодаря тому, что вычисляется каждым нейроном сети на основе функции ошибки, представленной в той или иной форме.

Каждый нейрон многослойного персептрона может выполнять два типа вычислений:

1) вычисление функционального сигнала на выходе нейрона, реализуемого в виде непрерывной нелинейной функции от входного сигнала и синаптических весов, связанных с данным нейроном;

2) вычисление оценки вектора градиента ошибки по синаптическим весам на входе данного нейрона. Эти значения необходимы для обратного прохода через сеть.

Рассмотрим простой метод обучения, в котором веса обновляются для каждого обучающего примера в пределах одной эпохи обучения. Настройка весов выполняется в соответствии с ошибками, вычисленными для каждого образа, представленного сети.

Согласно методу наименьших квадратов минимизируемой целевой функцией ошибки ИНС является величина

$$E(w) = \sum_{p=1}^P \sum_{k=1}^K \left(y_{k,p}^{(N)} - t_{k,p} \right)^2,$$

где $y_{k,p}^{(N)}$ – выходное состояние нейрона k выходного слоя $n = N$ при подаче на вход образа номер p из обучающей выборки. Структура обучающей выборки представлена в табл. 5.1.

Т а б л и ц а 5.1

Обучающая выборка

Номер образа	Вход						Желаемый выход					
	x_1	x_2	...	x_q	...	x_Q	t_1	t_2	...	t_k	...	t_K
1												
...												
p												
...												
P												

Здесь P – количество эталонных образов; Q – размерность векторов входных образов сети; K – количество нейронов в выходном слое сети.

Минимизация целевой функции ведется методом градиентного спуска, что означает подстройку весовых коэффициентов сети согласно дельта-правилу:

$$\Delta w_{ij}^{(n)} = -\eta \frac{\partial E}{\partial w_{ij}},$$

где $\Delta w_{ij}^{(n)}$ – приращение весового коэффициента синаптической связи, соединяющей i -й нейрон слоя $n-1$ с j -м нейроном слоя n , η – скорость обучения (коэффициент), причем $0 < \eta < 1$.

Доказано, что $\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial y_j} \cdot \frac{dy_j}{ds_j} \cdot \frac{\partial s_j}{\partial w_{ij}}$, где s_j – взвешенная сум-

ма входных сигналов нейрона j , т.е. аргумент активационной функции. Так как множитель dy_j/ds_j является производной этой функции по ее аргументу, то производная активационной функции должна быть определена на всей оси абсцисс. В связи с этим функция единичного скачка и прочие активационные функции с разрывами не подходят для рассматриваемых ИНС. В них применяются такие гладкие функции, как гиперболический тангенс или классическая сигмоидная функция (рис. 5.2).

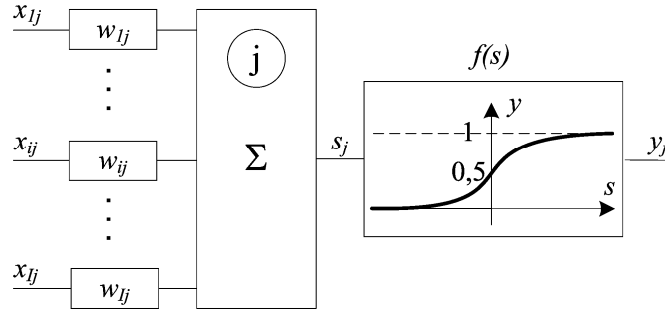


Рис. 5.2. Нейрон j промежуточного слоя

Третий множитель $\partial s_j / \partial w_{ij} = y_i^{(n-1)} = x_{ij}^{(n)}$, где $y_i^{(n-1)}$ – выход нейрона i предыдущего слоя n , т.е. вход нейрона j слоя n .

Доказано, что первый множитель легко раскладывается следующим образом:

$$\frac{\partial E}{\partial y_j} = \sum_k \frac{\partial E}{\partial y_k} \cdot \frac{dy_k}{ds_k} \cdot \frac{\partial s_k}{\partial y_j} = \sum_k \frac{\partial E}{\partial y_k} \cdot \frac{dy_k}{ds_k} \cdot w_{jk}^{(n+1)}.$$

Суммирование по k выполняется среди нейронов следующего слоя $n+1$. Введем новую переменную: $\delta_j^{(n)} = \frac{\partial E}{\partial y_j} \cdot \frac{dy_j}{ds_j}$, и получим рекурсивную формулу для расчетов величин $\delta_j^{(n)}$ слоя n из величин $\delta_k^{(n+1)}$ более старшего слоя $n+1$:

$$\delta_j^{(n)} = \left[\sum_k \delta_k^{(n+1)} \cdot w_{jk}^{(n+1)} \right] \cdot \frac{dy_j}{ds_j}. \quad (5.1)$$

Для выходного слоя

$$\delta_k^{(N)} = (y_k^{(N)} - t_k) \cdot \frac{dy_k}{ds_k}. \quad (5.2)$$

Теперь можно записать в раскрытом виде:

$$\Delta w_{ij}^{(n)} = -\eta \cdot \delta_j^{(n)} \cdot y_i^{(n-1)}. \quad (5.3)$$

Чем меньше параметр скорости обучения η , тем меньше корректировка синаптических весов, осуществляемая на каждой итерации, и тем более гладкой является траектория в пространстве весов. Однако это улучшение происходит за счет замедления процесса обучения. С другой стороны, если увеличить параметр η для повышения скорости обучения, то результирующие большие изменения синаптических весов могут привести систему в неустойчивое состояние. Простейшим способом повышения скорости обучения без потери устойчивости является изменение дельта-правила (5.3) за счет добавления к нему коэффициента инерционности:

$$\Delta w_{ij}^{(n)}(t) = -(\mu \cdot \Delta w_{ij}^{(n)}(t-1) + \eta \cdot \delta_j^{(n)}(t) \cdot y_i^{(n-1)}(t)), \quad (5.4)$$

где μ – коэффициент инерционности, t – номер текущего шага обучения. Уравнение (5.4) называют обобщенным дельта-правилом, при $\mu = 0$ оно вырождается в обычное дельта-правило.

Таким образом, полный алгоритм обучения ИНС с помощью процедуры обратного распространения ошибки включает следующие шаги:

1. Подать на входы сети один из возможных образов и в режиме обычного функционирования ИНС, когда сигналы распространяются от входов к выходам, рассчитать значения последних:

$$s_j^{(n)} = \sum_{i=0}^I y_i^{(n-1)} \cdot w_{ij}^{(n)},$$

где I – число нейронов в слое $n-1$ с учетом того, что $y_i^{(n-1)} = x_{ij}^{(n)}$ – i -й вход нейрона j слоя n ; $y_j^{(n)} = f(s_j^{(n)})$, где $f()$ – активационная функция нейрона; $y_q^{(0)} = x_q$, где x_q – q -я компонента вектора входного образа.

2. Вычислить $\delta_k^{(N)}$ для всех нейронов выходного слоя по формуле (5.2). Рассчитать по формуле (5.3) изменения весовых коэффициентов $\Delta w_{jk}^{(n)}$ слоя $n=N$.

3. Вычислить по формулам (5.1) и (5.3) величины $\delta_j^{(n)}$ и $\Delta w_{ij}^{(n)}$ для всех остальных слоев $n < N$ ($n = N-1, \dots, 1$).

4. Скорректировать все весовые коэффициенты ИНС:

$$w_{ij}^{(n)}(t) = w_{ij}^{(n)}(t-1) + \Delta w_{ij}^{(n)}(t).$$

5. Если ошибка сети, вычисленная по всем P эталонным образцам, существенна, перейти на шаг 1, в противном случае – «конец».

Нейронной сети на шаге 1 попеременно в случайном порядке предъявляются все тренировочные образы, чтобы ИНС «не забывала» одни по мере запоминания других. Алгоритм расчета значений $\delta_k^{(N)}$ в выходном слое представлен на рис. 5.3.

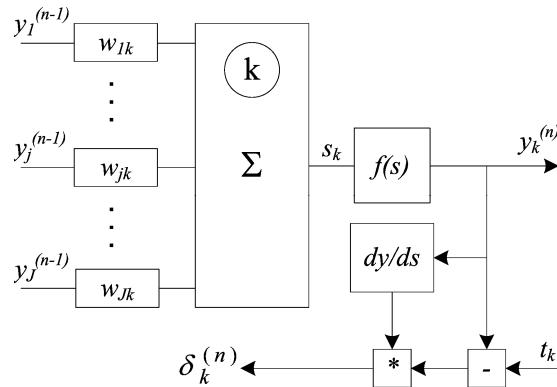


Рис. 5.3. Расчет сигнала обратного распространения в выходном слое ($n=N$)

Алгоритм расчета значений $\delta_j^{(n)}$ при обратном распространении в промежуточных слоях приведен на рис. 5.4.

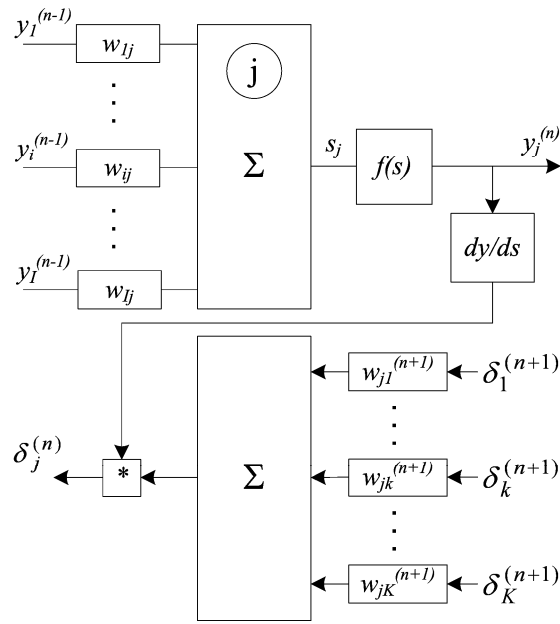


Рис. 5.4. Расчет сигнала обратного распространения в промежуточном слое ($n < N$)

5.2. Особенности обучения

Скорость обучения. При выводе алгоритма обратного распространения предполагалось, что параметр скорости обучения представлен константой η . Однако на практике он может задаваться как $\eta_{ji}^{(n)}$. Это значит, что параметр скорости обучения является локальным и определяется для каждой конкретной связи. Применяя различные параметры скорости обучения в разных областях сети, можно добиться интересных результатов. Следует отметить, что при реализации алгоритма обратного распространения можно изменять как все синаптические веса сети, так и только часть из них, оставляя остальные на время адаптации фиксированными.

В последнем случае сигнал ошибки распространяется по сети в обычном порядке, однако фиксированные синаптические веса будут оставаться неизменными. Этого можно добиться, установив для соответствующих синаптических весов $\Delta w_{ij}^{(n)}$ параметр $\eta_i^{(n)}$ равным нулю.

Режимы обучения. В практических приложениях алгоритма обратного распространения в процессе обучения многослойного персептрона ему многократно предъявляется предопределенное множество обучающих примеров. Один полный цикл предъявления полного набора примеров называют эпохой. Процесс обучения проводится от эпохи к эпохе, пока синаптические веса не стабилизируются, а среднеквадратическая ошибка на всем обучающем множестве не сойдется к некоторому минимальному значению. Целесообразно случайным образом изменять порядок представления примеров обучения для разных эпох. Такой принцип предъявления образов делает поиск в пространстве весов стохастическим, предотвращая потенциальную возможность появления замкнутых циклов в процессе эволюции синаптических весов. Для данного обучающего множества алгоритм обратного распространения можно реализовать в двух режимах.

Последовательный режим обучения по методу обратного распространения также иногда называют стохастическим или интерактивным. В этом режиме корректировка весов проводится после подачи каждого примера. Эпоха состоит из P упорядоченных примеров (табл. 5.1): $(x(1), t(1)), \dots, (x(P), t(P))$. Сети предъявляется первый пример $(x(1), t(1))$ этой эпохи, после чего выполняются описанные выше прямые и обратные вычисления. В результате проводится корректировка синаптических весов. После этого сети предъявляется вторая пара $(x(2), t(2))$ в этой эпохе, повторяются прямой и обратный проходы, приводящие к следующей коррекции синаптических весов. Этот процесс повторяется до тех пор, пока сеть не завершит обработку последнего примера (пары) данной эпохи – $(x(P), t(P))$.

В *пакетном режиме обучения* корректировка весов проводится после подачи в сеть всех примеров (эпохи). Для конкретной эпохи функция стоимости определяется как среднеквадратическая ошибка в составной форме:

$$E_{av}(w) = \frac{1}{2P} \sum_{p=1}^P \sum_{k=1}^K (y_{k,p}^{(N)} - t_{k,p})^2. \quad (5.5)$$

В выражении (5.5) внутреннее суммирование по k выполняется по всем нейронам выходного слоя сети, в то время как внешнее суммирование по p – по всем образам обучающей выборки (эпохи). Подстройка весовых коэффициентов осуществляется по дельта-правилу:

$$\Delta w_{ij}^{(n)} = -\eta \frac{\partial E_{av}}{\partial w_{ij}}. \quad (5.6)$$

Согласно (5.6) в пакетном режиме корректировка веса $\Delta w_{ij}^{(n)}$ выполняется только после прохождения по сети всего множества примеров из обучающей выборки (табл. 5.1).

Последовательный режим более предпочтителен, чем пакетный, так как позволяет реализовать стохастический поиск в пространстве весов за счет предъявления обучающих примеров в случайном порядке. Это уменьшает возможность остановки алгоритма в точке локального минимума.

Критерий останова алгоритма. Не существует какого-либо четко определенного критерия останова алгоритма обратного распространения. Известно лишь несколько обоснованных критериев, которые можно использовать для прекращения корректировки весов. Каждый из них имеет свои практические преимущества. Для того чтобы сформулировать такой критерий, можно рассуждать в терминах свойств локального и глобального минимума поверхности ошибок.

Обозначим символом w^* вектор весов, обеспечивающий минимум целевой функции ошибки сети, будь то локальный или глобальный. Необходимым условием минимума является то, что вектор градиента $g(w)$ (т.е. вектор частных производных первого порядка $\frac{\partial E}{\partial w_{ij}}$) для поверхности ошибки в этой точке равен нулевому.

Следовательно, можно сформулировать разумный критерий сходимости алгоритма обучения обратного распространения.

Считается, что алгоритм сошелся, если евклидова норма вектора градиента $\|g(w)\|$ достигает достаточно малых значений. Не-

достаток этого критерия в том, что для сходимости обучения может потребоваться довольно много времени. Кроме того, необходимо постоянно вычислять вектор градиента $g(w)$.

Другим уникальным свойством минимума является то, что целевая функция $E_{av}(w)$ в точке $w = w^*$ стабилизируется. Отсюда можно вывести еще один критерий сходимости – достаточно малую абсолютную интенсивность изменений среднеквадратической ошибки $E_{av}(w)$ от эпохи к эпохе. В соседних эпохах она обычно считается достаточно малой, если лежит в пределах 0,1...1%. К сожалению, этот критерий может привести к преждевременной остановке процесса обучения.

Существует еще один теоретически подкрепленный критерий. После каждой итерации обучения сеть тестируется на эффективность обобщения. Процесс обучения останавливается, когда эффективность обобщения становится удовлетворительной или когда оказывается, что пик эффективности уже пройден. Рассмотрим понятие обобщения сетью обучающих примеров.

Обобщение. При обучении методом обратного распространения в сеть подают примеры из обучающей выборки и вычисляют синаптические веса многослойного персептрона. При этом разработчик надеется, что обученная таким образом сеть будет способна к обобщению. Считается, что сеть обладает хорошей обобщающей способностью, если осуществляемое ею отображение входа на выход является корректным для данных, никогда ранее не «виденных» сетью в процессе обучения. При этом считается, что примеры для проверки обобщающей способности сети принадлежат той же совокупности, из которой они брались для обучения.

Процесс обучения нейронной сети можно рассматривать как задачу аппроксимации кривой. Сама сеть при этом выступает как нелинейный оператор. Тогда обобщение – это результат хорошей нелинейной интерполяции входных данных. Сеть осуществляет корректную интерполяцию в основном за счет того, что непрерывность отдельных функций активации многослойного персептрона обеспечивает непрерывность общей выходной функции. На рис. 5.5 показано, как происходит обобщение в гипотетической сети. Нелинейное функциональное преобразование, осуществляе-

мое сетью, определяется в результате обучения по дискретным точкам (обучающим данным). Точку, полученную в процессе обобщения и обозначенную кружком, можно рассматривать как результат выполняемой сетью интерполяции.

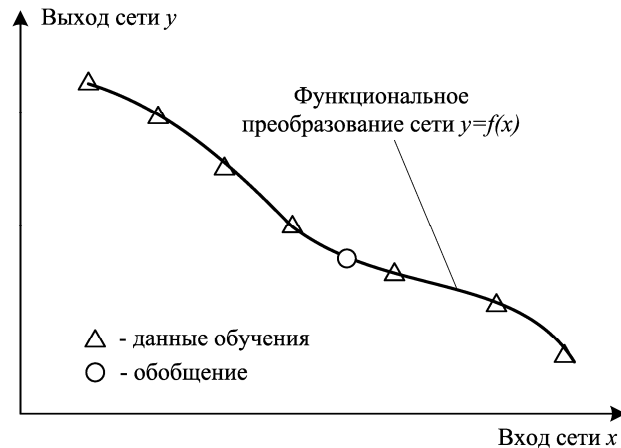


Рис. 5.5. Обобщение обучающих примеров

Нейронная сеть, спроектированная с учетом хорошего обобщения, будет корректно отображать вход на выход даже тогда, когда входной сигнал слегка отличается от примеров, использованных для обучения сети (что и показано на рис. 5.5). Однако если сеть обучается на слишком большом количестве примеров, все может закончиться только запоминанием данных обучения. Это может произойти за счет нахождения таких признаков (например, благодаря шуму), которые присутствуют в примерах обучения, но не свойственны самой моделируемой функции отображения. Такое явление называется избыточным обучением, или переобучением (overtraining). Если сеть «переобучена», она теряет свою способность к обобщению на аналогичных входных сигналах.

Использование большого количества скрытых нейронов, чем это действительно необходимо, выливается в нежелательное изменение входного пространства из-за шума, который содержится в синаптических весах сети. Пример плохого обобщения вследствие простого запоминания обучающих образов приведен на рис. 5.6

для тех же данных, что и на рис. 5.5. Результат запоминания, в сущности, представляет справочную таблицу – список пар «вход-выход», вычисленных нейронной сетью. При этом отображение теряет свою гладкость.



Рис. 5.6. Пример плохого обобщения

Способность к обобщению определяется тремя факторами: размером обучающего множества и его представительностью, архитектурой нейронной сети и физической сложностью рассматриваемой задачи.

В контексте двух первых факторов вопрос обобщения можно рассматривать с двух различных точек зрения:

- 1) архитектура сети фиксирована, и вопрос сводится к определению размера обучающего множества, необходимого для хорошего обобщения;
- 2) размер обучающего множества фиксирован, и вопрос сводится к определению наилучшей архитектуры сети, позволяющей достичь хорошего обобщения.

Обе точки зрения по-своему правильны. На практике оказывается, что для хорошего обобщения достаточно, чтобы размер обучающего множества P удовлетворял соотношению

$$P = O(N_W / \epsilon), \quad (5.7)$$

где N_W – общее количество свободных параметров сети (т.е. синаптических весов), ϵ – допустимая точность ошибки классификации;

$O(.)$ – порядок заключенной в скобки величины. Например, для ошибки в 10% количество примеров обучения должно в 10 раз превосходить количество свободных параметров сети.

Перекрестная проверка. Рассмотрим процесс обучения сети с точки зрения настройки ее параметров для заданного множества данных. Тогда проблему можно рассматривать как задачу выбора наилучшей модели из множества структур «сетей-кандидатов» с учетом определенного критерия.

Основой для решения такой задачи может стать стандартный статистический подход, получивший название перекрестной проверки (cross-validation). В рамках этого подхода имеющиеся в наличии данные сначала случайным образом разбиваются на обучающее множество (training set) и тестовое множество (test set). Обучающее множество, в свою очередь, разбивается на два следующих несвязанных подмножества: подмножество для оценивания (estimation subset), используемое для выбора модели, и проверочное подмножество (validation subset) – для тестирования модели.

Идея этого подхода состоит в проверке качества модели на данных, отличных от использованных для параметрического оценивания. Таким образом, обучающее множество можно использовать для проверки эффективности различных структур «сетей-кандидатов», из которых необходимо выбрать лучшую. Однако существует некоторая вероятность того, что отобранная модель по параметрам эффективности окажется излишне переученной на проверочном подмножестве. Чтобы предотвратить такую опасность, эффективность обобщения выбранной модели измеряется также на тестовом множестве, которое отличается от проверочного. Перекрестную проверку можно использовать для выбора количества скрытых нейронов в многослойном персептроне.

Метод обучения с ранним остановом. При нормальных условиях обучения среднеквадратическая ошибка уменьшается по мере увеличения количества эпох обучения: она начинается с довольно больших значений, стремительно уменьшается, а затем убывает все медленнее по мере продвижения сети к локальному минимуму на поверхности ошибок. Если вовремя не остановить сеанс обучения, то существенно повышается вероятность излишнего переобучения сети. Если главной целью является хорошая способность к обобщению, то по виду кривой ошибки довольно сложно определить момент, когда следует остановить процесс обучения.

Наступление стадии излишнего переобучения можно определить с помощью перекрестной проверки, в которой данные разбиты на два подмножества: оценивания и проверки. Множество оценивания используется для обычного обучения сети с небольшой модификацией: сеанс обучения периодически останавливается (через каждые несколько эпох), после чего сеть тестируется на проверочном множестве.

Более точно периодический процесс оценивания и тестирования выполняется следующим образом. По завершении этапа обучения синаптические веса и уровни порогов многослойного персептрона фиксируются и сеть переключается в режим прямого прохода. Ошибка сети вычисляется для каждого примера из проверочного подмножества. По завершении тестирования наступает следующий этап процесса обучения и все повторяется.

Такая процедура называется методом обучения с ранним остановом. На рис. 5.7 показана форма двух кривых обучения. Первая относится к измерениям на подмножестве для оценивания, а вторая – к измерениям на проверочном подмножестве. Обычно на втором подмножестве ошибка тестирования ведет себя не так, как на первом, на основе которого формируется архитектура сети.

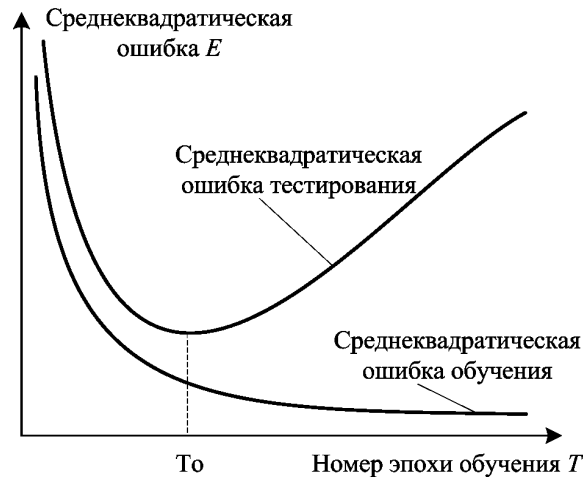


Рис. 5.7. Обучение с ранним остановом

Функция обучения на подмножестве оценивания монотонно убывает с увеличением числа эпох, в то время как кривая обучения на проверочном подмножестве монотонно убывает до некоторого минимума, после чего при продолжении обучения начинает возрастать. Если учитывать только кривую, построенную на подмножестве оценивания, то возникает желание продолжить обучение и после прохождения точки минимума на кривой, построенной на проверочном подмножестве. Однако на практике оказывается, что после прохождения этой точки сеть обучается только посторонним шумам, содержащимся в обучающей выборке. Поэтому можно выдвинуть эвристическое предположение о том, что точку минимума кривой тестирования можно использовать в качестве критерия останова сеанса обучения.

Статистическая теория явления чрезмерного переобучения дает некоторые рекомендации по выбору размера обучающего множества. Согласно полученным результатам переобучение может наступить при $P < 30N_W$, где P – размер обучающей выборки, N_W – количество свободных параметров сети. Только в этом случае существует практическая польза от использования перекрестной проверки для останова сеанса обучения. Оптимальное значение параметра r , определяющее разбиение множества примеров на подмножества для тестирования и обучения, должно выбираться из соотношения $r = 1 - (\sqrt{2N_W - 1} - 1) / 2(N_W - 1)$. Например, при $N_W = 100$ $r = 0,07$. Это значит, что 93% данных обучения составляет подмножество обучения и только 7% приходится на тестовое подмножество.

При большем размере обучающего множества ($P > 30N_W$) эффективность метода раннего останова обучения снижается [12].

Выбор диапазона входных значений. Следует отметить, что общепринятый динамический диапазон входов и выходов $[0,1]$ скрытых нейронов неоптимален. Так как величина коррекции веса Δw_{ij} пропорциональна выходному уровню нейрона, то из (5.3) следует, что, когда выходное значение $y_i^{(n-1)}$ стремится к нулю, эффективность обучения заметно снижается. При двоичных входных векторах в среднем половина весовых коэффициентов не будет корректироваться, поэтому область возможных значений выходов нейронов $[0,1]$ желательно сдвинуть в пределы $[-0.5, +0.5]$, что достигается простыми модификациями активационных функций. Например, активационная функция сигмоидного вида преобразу-

ется к виду $f(x) = -0.5 + \frac{1}{1 + e^{-\alpha \cdot x}}$. С помощью таких простых средств время сходимости сокращается в среднем от 30 до 50%. Это является одним из примеров модификации, существенно улучшающей характеристику алгоритма.

Недостатки алгоритма обратного распространения. Алгоритм обратного распространения ошибки имеет существенный недостаток, проявляющийся в сложных задачах. Больше всего неприятностей доставляет неопределенно долгий процесс обучения сети. Неудачи в обучении обычно возникают по двум причинам: в результате так называемого «паралича» сети и попадания в локальный минимум.

«Паралич» сети означает остановку процесса обучения, когда весовые коэффициенты перестают изменяться до достижения требуемой ошибки обучения. Эта ситуация возникает, когда большие положительные или отрицательные значения весовых коэффициентов смещают рабочую точку на активационных сигмоидных (логистических) функциях многих нейронов в область насыщения. Малые величины производной от логистической функции приведут, в соответствии с (5.1) и (5.2), к остановке обучения, что парализует ИНС.

Попадание в область локального минимума обусловлено тем, что обратное распространение использует разновидность градиентного спуска, т.е. осуществляет спуск вниз по поверхности ошибки, непрерывно подстраивая веса в направлении минимума. Поверхность ошибки сложной сети сильно изрезана в пространстве высокой размерности. Сеть может попасть в локальный минимум (неглубокую долину), когда рядом имеется гораздо более глубокий минимум. В точке локального минимума все направления ведут вверх, и сеть не способна из него выбраться.

Эта проблема связана еще с одной, а именно – с выбором величины скорости обучения. Доказательство сходимости обучения в процессе обратного распространения основано на производных, т.е. приращения весов и, следовательно, скорость обучения должны быть бесконечно малыми, однако в этом случае обучение будет происходить неприемлемо медленно. С другой стороны, слишком большие коррекции весов могут привести к постоянной неустойчивости процесса обучения. Поэтому обычно выбирают величину скорости обучения η меньше единицы, но не очень больше нуля,

например $\eta=0,1$. В процессе обучения она может постепенно уменьшаться. Кроме того, для исключения случайных попаданий в локальные минимумы иногда, после того как значения весовых коэффициентов стабилизируются, η кратковременно сильно увеличивают, чтобы начать градиентный спуск из новой точки. Если повторение этой процедуры несколько раз приведет алгоритм в одно и то же состояние ИНС, можно более или менее уверенно сказать, что найден глобальный минимум ошибки, а не какой-то другой. Существуют также стохастические методы исключения локальных минимумов и паралича ИНС.

Стохастические методы обучения выполняют псевдослучайные изменения величин весов, сохраняя те изменения, которые ведут к улучшениям. Для обучения сети может быть использована следующая процедура:

1. Предъявить сети множество обучающих примеров (см. табл. 5.1) и вычислить значения на выходе сети.
2. Сравнить эти выходы с желаемыми и вычислить значение целевой функции $E(w)$ по формуле (5.5). Целью обучения является минимизация ошибки обучения $E(w)$.
3. Выбрать случайным образом весовой коэффициент и подкорректировать его на небольшое случайное значение. Если коррекция помогает (уменьшает целевую функцию), то сохранить ее, в противном случае вернуться к первоначальному значению веса.
4. Повторять шаги 1 – 3 до тех пор, пока сеть не будет обучена в достаточной степени.

Чтобы исключить опасность попадания в локальный минимум целевой функции, на начальных шагах обучения случайные приращения весов делают большими, а затем постепенно уменьшают. Это позволяет сети вырываться из локальных минимумов и в то же время гарантирует окончательную стабилизацию сети. Известны также другие методы стохастического обучения: больцмановское обучение, обучение Коши, метод искусственной теплоемкости [11].

Теперь коснемся вопроса емкости ИНС – числа образов, предъявляемых на ее входы, которые она способна научиться распознавать. Известно, что для ИНС с двумя слоями, т.е. выходным и одним скрытым слоем, емкость сети C оценивается так:

$$N_w/K < C < N_w/K \log(N_w/K),$$

где N_w – число подстраиваемых весов, K – число нейронов в выходном слое.

Для разделения множества входных образов, например по двум классам, достаточно всего одного выхода. При этом каждый логический уровень – "1" и "0" – будет обозначать отдельный класс. На двух выходах можно закодировать уже четыре класса и т.д. Однако результаты работы сети, организованной таким образом, не очень надежны. Для повышения достоверности классификации желательно ввести избыточность путем выделения каждому классу одного нейрона в выходном слое или, что еще лучше, нескольких, каждый из которых обучается определять принадлежность образа к классу со своей степенью достоверности, например высокой, средней и низкой. Такие ИНС позволяют проводить классификацию входных образов, объединенных в нечеткие (размытые или пересекающиеся) множества. Это свойство приближает подобные ИНС к условиям реальной жизни.

5.3. Пример обучения ИНС методом обратного распространения ошибки

Контрольный пример обучения трехслойного персептрона.

Рассмотрим контрольный пример для иллюстрации работы алгоритма обратного распространения ошибки на примере трехслойного персептрона (рис. 5.8) с количеством нейронов по слоям 2-2-1.

Функции активации нейронов сигмоидного типа: $f(s) = (1 + \exp(-a_i s))^{-1}$. Коэффициенты наклона a_i задаются для каждого слоя i : $[a_1 \ a_2 \ a_3] = [0,6 \ 0,8 \ 2]$.

Первая производная активационной функции, используемая при вычислении сигнала обратного распространения, может быть преобразована к виду $f'(s) = a_i f(s) (1 + f(s))$.

Вектор входных значений:

$$[x_1 \ x_2 \ x_3] = [0,1 \ 0,5 \ 0,7].$$

Желаемый выход сети при этих входных значениях: $t_1 = 0,1$.

Начальные значения весовых коэффициентов по слоям:

первый слой:

$$[w_{11}^{(1)} \ w_{21}^{(1)} \ w_{31}^{(1)}] = [-2 \ 3 \ 2],$$

$$[w_{12}^{(1)} \ w_{22}^{(1)} \ w_{32}^{(1)}] = [-3 \ 3 \ 2],$$

второй:

$$[w_{11}^{(2)} \ w_{21}^{(2)}] = [-2 \ -3],$$

$$[w_{12}^{(2)} \ w_{22}^{(2)}] = [2 \ 3],$$

третий:

$$[w_1^{(3)} \ w_2^{(3)}] = [2 \ 1].$$

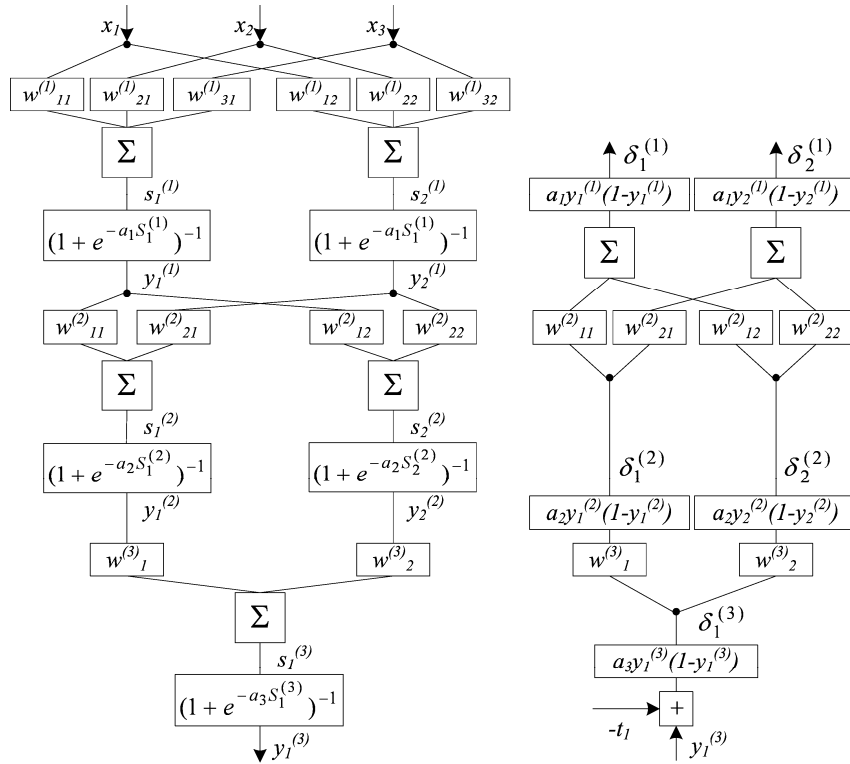


Рис. 5.8. Прямое и обратное распространение сигнала в трехслойном персептроне

Прямое распространение сигнала:

первый слой:

$$\begin{aligned}
 s_1^{(1)} &= x_1 w_{11}^{(1)} + x_2 w_{21}^{(1)} + x_3 w_{31}^{(1)} = 0,1 \cdot (-2) + 0,5 \cdot 3 + 0,7 \cdot 2 = 2,7; \\
 s_2^{(1)} &= x_1 w_{12}^{(1)} + x_2 w_{22}^{(1)} + x_3 w_{32}^{(1)} = 0,1 \cdot (-3) + 0,5 \cdot 3 + 0,7 \cdot 2 = 2,6; \\
 y_1^{(1)} &= (1 + \exp(-a_1 s_1^{(1)}))^{-1} = (1 + \exp(-0,6 \cdot 2,7))^{-1} = 0,835; \\
 y_2^{(1)} &= (1 + \exp(-a_1 s_2^{(1)}))^{-1} = (1 + \exp(-0,6 \cdot 2,6))^{-1} = 0,826;
 \end{aligned}$$

второй:

$$\begin{aligned}
 s_1^{(2)} &= y_1^{(1)} w_{11}^{(2)} + y_2^{(1)} w_{21}^{(2)} = 0,835 \cdot (-2) + 0,826 \cdot (-3) = -4,148; \\
 s_2^{(2)} &= y_1^{(1)} w_{12}^{(2)} + y_2^{(1)} w_{22}^{(2)} = 0,835 \cdot 2 + 0,826 \cdot 3 = 4,148; \\
 y_1^{(2)} &= (1 + \exp(-a_2 s_1^{(2)}))^{-1} = (1 + \exp(-0,8 \cdot (-4,148)))^{-1} = 0,035; \\
 y_2^{(2)} &= (1 + \exp(-a_2 s_2^{(2)}))^{-1} = (1 + \exp(-0,8 \cdot 4,148))^{-1} = 0,965;
 \end{aligned}$$

третий:

$$s_1^{(3)} = y_1^{(2)} w^{(3)}_{11} + y_2^{(2)} w^{(3)}_{12} = 0,035 \cdot 2 + 0,965 \cdot 1 = 1,035;$$

$$y_1^{(3)} = (1 + \exp(-a_3 s_1^{(3)}))^{-1} = (1 + \exp(-2 \cdot 1,035))^{-1} = 0,888.$$

$$\text{Ошибка сети: } e = y_1^{(3)} - t_1 = 0,888 - 0,1 = 0,788.$$

Обратное распространение ошибки (см. рис. 5.8):

третий слой:

$$\delta_1^{(3)} = (y_1^{(3)} - t_1) a_3 y_1^{(3)} (1 - y_1^{(3)}) = 0,788 \cdot 2 \cdot 0,888 \cdot (1 - 0,888) = 0,157;$$

второй:

$$\delta_1^{(2)} = \delta_1^{(3)} w^{(3)}_{11} a_2 y_1^{(2)} (1 - y_1^{(2)}) = 0,157 \cdot 2 \cdot 0,8 \cdot 0,035 \cdot (1 - 0,035) = 0,008;$$

$$\delta_2^{(2)} = \delta_1^{(3)} w^{(3)}_{12} a_2 y_2^{(2)} (1 - y_2^{(2)}) = 0,157 \cdot 1 \cdot 0,8 \cdot 0,965 \cdot (1 - 0,965) = 0,004;$$

первый:

$$\begin{aligned} \delta_1^{(1)} &= (\delta_1^{(2)} w^{(2)}_{11} + \delta_2^{(2)} w^{(2)}_{12}) a_1 y_1^{(1)} (1 - y_1^{(1)}) = \\ &= (0,008 \cdot (-2) + 0,004 \cdot (-3)) \cdot 0,6 \cdot 0,835 \cdot (1 - 0,835) = -0,002; \end{aligned}$$

$$\begin{aligned} \delta_2^{(1)} &= (\delta_1^{(2)} w^{(2)}_{21} + \delta_2^{(2)} w^{(2)}_{22}) a_1 y_2^{(1)} (1 - y_2^{(1)}) = \\ &= (0,008 \cdot (2) + 0,004 \cdot (3)) \cdot 0,6 \cdot 0,826 \cdot (1 - 0,826) = 0,002. \end{aligned}$$

Корректировка весовых коэффициентов:

первый слой:

$$w^{(1)}_{11} = w^{(1)}_{11} - \eta \delta_1^{(1)} x_1 = -2 - 0,8 \cdot (-0,002) \cdot 0,1 = -1,9998;$$

$$w^{(1)}_{21} = w^{(1)}_{21} - \eta \delta_1^{(1)} x_2 = 3 - 0,8 \cdot (-0,002) \cdot 0,5 = 3,0008;$$

$$w^{(1)}_{31} = w^{(1)}_{31} - \eta \delta_1^{(1)} x_3 = 2 - 0,8 \cdot (-0,002) \cdot 0,7 = 2,0011;$$

$$w^{(1)}_{12} = w^{(1)}_{12} - \eta \delta_2^{(1)} x_1 = -3 - 0,8 \cdot (0,002) \cdot 0,1 = -3,0002;$$

$$w^{(1)}_{22} = w^{(1)}_{22} - \eta \delta_2^{(1)} x_2 = 3 - 0,8 \cdot (0,002) \cdot 0,5 = 2,9992;$$

$$w^{(1)}_{32} = w^{(1)}_{32} - \eta \delta_2^{(1)} x_3 = 2 - 0,8 \cdot (0,002) \cdot 0,7 = 1,9989;$$

второй:

$$w^{(2)}_{11} = w^{(2)}_{11} - \eta \delta_1^{(2)} y_1^{(1)} = -2 - 0,8 \cdot 0,008 \cdot 0,835 = -2,0053;$$

$$w^{(2)}_{21} = w^{(2)}_{21} - \eta \delta_1^{(2)} y_2^{(1)} = -3 - 0,8 \cdot 0,008 \cdot 0,826 = -3,0053;$$

$$w^{(2)}_{12} = w^{(2)}_{12} - \eta \delta_2^{(2)} y_1^{(1)} = 2 - 0,8 \cdot 0,004 \cdot 0,835 = 1,9973;$$

$$w^{(2)}_{22} = w^{(2)}_{22} - \eta \delta_2^{(2)} y_2^{(1)} = 3 - 0,8 \cdot 0,004 \cdot 0,826 = 2,9974;$$

третий:

$$w^{(3)}_1 = w^{(3)}_1 - \eta \delta^{(3)}_1 y_1^{(2)} = 2 - 0,8 \cdot 0,157 \cdot 0,035 = 1,9956;$$

$$w^{(3)}_2 = w^{(3)}_2 - \eta \delta^{(3)}_1 y_2^{(2)} = 1 - 0,8 \cdot 0,157 \cdot 0,965 = 0,8788.$$

Прямое распространение сигнала после корректировки весовых коэффициентов:

первый слой:

$$s_1^{(1)} = x_1 w^{(1)}_{11} + x_2 w^{(1)}_{21} + x_3 w^{(1)}_{31} =$$

$$= 0,1 \cdot (-1,9998) + 0,5 \cdot 3,0008 + 0,7 \cdot 2,0011 = 2,7012;$$

$$s_2^{(1)} = x_1 w^{(1)}_{12} + x_2 w^{(1)}_{22} + x_3 w^{(1)}_{32} =$$

$$= 0,1 \cdot (-3,0002) + 0,5 \cdot 2,9992 + 0,7 \cdot 1,9988 = 2,5988;$$

$$y_1^{(1)} = (1 + \exp(-a_1 s_1^{(1)}))^{-1} = (1 + \exp(-0,6 \cdot 2,7012))^{-1} = 0,8349;$$

$$y_2^{(1)} = (1 + \exp(-a_1 s_2^{(1)}))^{-1} = (1 + \exp(-0,6 \cdot 2,5988))^{-1} = 0,8263;$$

второй:

$$s_1^{(2)} = y_1^{(1)} w^{(2)}_{11} + y_2^{(1)} w^{(2)}_{21} = 0,8349 \cdot (-2,0053) + 0,8263 \cdot (-3,0053) = -4,1575;$$

$$s_2^{(2)} = y_1^{(1)} w^{(2)}_{12} + y_2^{(1)} w^{(2)}_{22} = 0,8349 \cdot 1,9973 + 0,8263 \cdot 2,9974 = 4,1443;$$

$$y_1^{(2)} = (1 + \exp(-a_2 s_1^{(2)}))^{-1} = (1 + \exp(-0,8 \cdot (-4,1575)))^{-1} = 0,0347;$$

$$y_2^{(2)} = (1 + \exp(-a_2 s_2^{(2)}))^{-1} = (1 + \exp(-0,8 \cdot 4,1443))^{-1} = 0,9650;$$

третий:

$$s_1^{(3)} = y_1^{(2)} w^{(3)}_{11} + y_2^{(2)} w^{(3)}_{21} = 0,0347 \cdot 1,9956 + 0,9650 \cdot 0,8788 = 0,9172;$$

$$y_1^{(3)} = (1 + \exp(-a_3 s_1^{(3)}))^{-1} = (1 + \exp(-2 \cdot 0,9172))^{-1} = 0,8623.$$

$$\text{Ошибка сети после корректировки: } e = y_1^{(3)} - t_1 = 0,8623 - 0,1 = 0,7623.$$

В результате настройки весовых коэффициентов сети после первого цикла обучения ошибка уменьшилась на $(0,788 - 0,7623) = 0,0257$.

Прогнозирование результатов выборов. Данная задача является классическим примером демонстрации работы нейросетевого классификатора, реализованного с помощью многослойного персептрона. Значения всех компонент обучающих векторов заданы в двоичной форме и основаны на реальных данных. Содержательная постановка задачи приведена в [1].

Рассмотрим использование нейросетевого классификатора на примере предсказания итогов выборов президента США. При существовании в США двухпартийной системы соперничество за пост президента идет между кандидатами от двух партий: демократической и республиканской. Одна из них на момент выборов является правящей, другая находится в оппозиции. Ответ на во-

прос, кто победит, кандидат от правящей или оппозиционной партии, и является целью прогноза.

Предполагается, что результат выборов определяется рядом объективных и субъективных факторов, характеризующих ситуацию, сложившуюся в стране накануне голосования. Эксперты отобрали наиболее значимые, с их точки зрения, факторы, сформулировав их в виде вопросов, на которые можно дать однозначные ответы:

1. Правящая партия была у власти более одного срока?
2. Правящая партия получила больше 50% на прошлых выборах?
3. В год выборов была активна третья политическая сила?
4. Была серьезная конкуренция при выдвижении кандидата от правящей партии?
5. Кандидат от правящей партии был президентом в год выборов?
6. Год выборов был временем спада или депрессии?
7. Рост среднего национального валового продукта на душу населения в год выборов был более 2,1%?
8. Произвел ли правящий президент существенные изменения в политике?
9. Во время последнего президентского срока были существенные социальные волнения?
10. Администрация правящей партии виновна в серьезной ошибке или скандале?
11. Кандидат правящей партии – национальный герой?
12. Кандидат оппозиционной партии – национальный герой?

Обучающая выборка состоит из 31 примера, каждый из которых представляет ситуацию выборов, начиная с 1864 г. (табл. 5.2), где ответы «Да» обозначены «1», а ответы «Нет» – «0».

Классификация заключается в разделении всех примеров на два класса. Класс «1» означает, что в данной ситуации был избран кандидат от правящей партии, «0» – кандидат от оппозиционной партии. После обучения ИНС должна предсказать ответ для ситуации, которая не входила в обучающую выборку (задача решалась в 1992 г., и результаты выборов еще не были известны).

Т а б л и ц а 5.2

Обучающая выборка

№	Год	Вход												Выход
		1	2	3	4	5	6	7	8	9	10	11	12	Класс
1	1860	1	0	1	1	0	0	1	0	1	0	0	0	0
2	1864	0	0	0	0	1	0	0	1	1	0	0	0	1
3	1868	1	1	1	0	0	0	1	1	1	0	1	0	1
4	1872	1	1	0	0	1	0	1	0	0	0	1	0	1
5	1876	1	1	0	1	0	1	0	0	0	1	0	0	0
6	1880	1	0	0	1	0	0	1	1	0	0	0	0	1
7	1884	1	0	0	1	0	0	1	0	1	0	1	0	0
8	1888	0	0	0	0	1	0	0	0	0	0	0	0	1
9	1892	0	0	1	0	1	0	0	1	1	0	0	1	0
10	1896	0	0	0	1	0	1	0	1	1	0	1	0	0
11	1900	0	1	0	0	1	0	1	0	0	0	0	1	1
12	1904	1	1	0	0	1	0	0	0	0	0	1	0	1
13	1908	1	1	0	0	0	0	0	1	0	0	0	1	1
14	1912	1	1	1	1	1	0	1	0	0	0	0	0	0
15	1916	0	0	0	0	1	0	0	1	0	0	0	0	1
16	1920	1	0	0	1	0	0	0	1	1	0	0	0	0
17	1924	0	1	1	0	1	0	1	1	0	1	0	0	1
18	1928	1	1	0	0	0	0	1	0	0	0	0	0	1
19	1932	1	1	0	0	1	1	0	0	1	0	0	1	0
20	1936	0	1	0	0	1	1	1	1	0	0	1	0	1
21	1940	1	1	0	0	1	1	1	1	0	0	1	0	1
22	1944	1	1	0	0	1	0	1	1	0	0	1	0	1
23	1948	1	1	1	0	1	0	0	1	0	0	0	0	1
24	1952	1	0	0	1	0	0	1	0	0	1	0	1	0
25	1956	0	1	0	0	1	0	0	0	0	0	1	0	1
26	1960	1	1	0	0	0	1	0	0	0	0	0	1	0
27	1964	0	0	0	0	1	0	1	0	0	0	0	0	1
28	1968	1	1	1	1	0	0	1	1	1	0	0	0	0
29	1972	0	0	0	0	1	0	1	1	1	0	0	0	1
30	1976	1	1	0	1	1	0	0	0	0	1	0	0	0
31	1980	0	0	1	1	1	1	0	0	0	1	0	1	0
32	1992	0	1	1	1	1	1	0	1	1	1	0	0	?

Для решения задачи используется персептрон с одним скрытым слоем. Очевидно, что число входных нейронов – 12, выходных – 1. Число нейронов в скрытом слое можно рассчитать: $N/2 - n - m < L < N/2 - n - m$, где n – размерность входного сигнала, m – размерность выходного сигнала, N – число элементов обучающей выборки, L – число нейронов в скрытых слоях.

Создадим нейронную сеть заданной топологии и проведем ее обучение методом обратного распространения. Для тестирования сети выберем в качестве исходных данных последнюю – 32-ю строку таблицы, которая не использовалась для обучения. Полученный результат, близкий к значению «0», говорит о том, что на 1992 г. прогнозировалась победа кандидата от оппозиционной партии, т.е. прогноз оказался правильным – Б. Клинтон обошел Д. Буша-старшего.

6. СЕТИ АДАПТИВНОЙ РЕЗОНАНСНОЙ ТЕОРИИ

6.1. Проблема стабильности-пластичности

Восприятие внешнего мира человеком связано с решением определенной дилеммы. Зачастую требуется определить, является ли некоторый образ "новой" информацией, требующей сохранения в памяти, либо он является "старой", уже знакомой картиной, и в этом случае запоминание не требуется. Таким образом, восприятие человека одновременно пластично, адаптировано к новой информации и при этом стабильно, т.е. не разрушает память о старых образах. Человеческая память эффективно хранит и корректирует запоминаемые образы. Ни предъявление нового образа, ни изменение старых не приводит к уничтожению памяти или к невозможности запоминания.

Большинство нейронных сетей не приспособлены к решению этой задачи. Например, многослойный персептрон, обучающийся методом обратного распространения, запоминает весь набор обучающей информации. Образы из обучающей выборки многократно предъявляются сети в процессе обучения. Если полностью обученный персептрон должен запомнить новый обучающий образ, он может изменить веса настолько, что потребуются полное переобучение сети. Таким образом, персептрон не способен к запоми-

нению новой информации без разрушения памяти о ранее запомненных образах.

Более того, в реальной ситуации ИНС может никогда не увидеть один и тот же обучающий образ дважды. При таких обстоятельствах сеть зачастую не будет обучаться; она будет непрерывно изменять свои веса, не достигая удовлетворительных результатов.

В сетях адаптивной резонансной теории (АРТ) предпринимается попытка приблизить механизм запоминания образов в ИНС к биологическому. Результатом работы АРТ является устойчивый набор запомненных образов и возможность выборки "похожего" образа по произвольному вектору, предъявленному на вход сети. Важное качество АРТ – динамическое запоминание новых образов без полного переобучения и отсутствие потерь уже запомненных образов при предъявлении новых.

Математическое описание АРТ сложно, но основные идеи и принципы реализации достаточно просты для понимания. АРТ включает в себя несколько моделей, каждая из которых определяется формой входных данных и способом их обработки. Модель АРТ-1 предназначена для обработки двоичных входных векторов. Более поздние обобщения этой модели могут классифицировать как двоичные, так и непрерывные векторы.

6.2. Архитектура АРТ

Упрощенная схема АРТ. Сеть АРТ представляет собой векторный классификатор. Входной вектор классифицируется в зависимости от того, на какой из множества ранее запомненных образов он похож. Свое классификационное решение АРТ выражает в форме возбуждения одного из нейронов распознающего слоя. Если входной вектор не соответствует ни одному из запомненных образов, создается новая категория посредством запоминания образа, идентичного новому входному вектору. Если входной вектор по определенному критерию схож с одним из ранее запомненных векторов, то этот запомненный вектор будет изменяться (переобучаться) таким образом, чтобы стать более похожим на новый входной вектор. Запомненный образ не будет изменяться, если текущий входной вектор не окажется достаточно похожим на него. Таким способом решается дилемма стабильности-пластичности. Новый образ может создавать дополнительные классификацион-

ные категории, однако новый входной образ не может изменить существующую память.

На рис. 6.1 приведена упрощенная конфигурация сети АРТ, представленная в виде пяти функциональных модулей. Она включает два слоя нейронов: слой сравнения и слой распознавания. Приемники Пр1 и Пр2 и «Сброс» обеспечивают управляющие функции, необходимые для обучения и классификации.

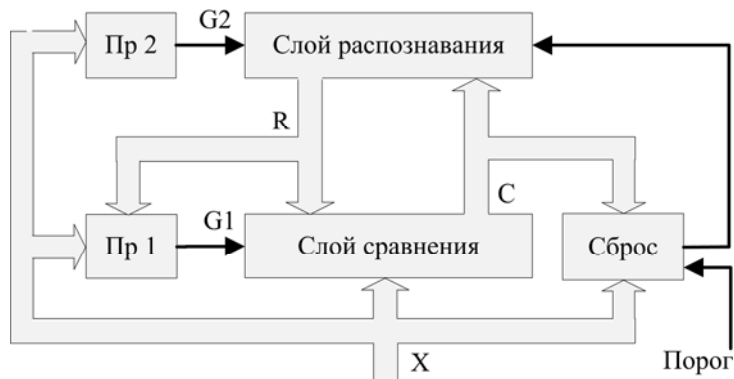


Рис. 6.1. Упрощенная сеть АРТ

Слой сравнения. Слой сравнения получает двоичный входной вектор X и первоначально пропускает его без изменения для формирования выходного вектора $C=X$. На более поздней фазе распознавания двоичный вектор R модифицирует вектор C , тогда в общем случае $C \neq X$.

Каждый нейрон в слое сравнения (рис. 6.2) получает три двоичных входа (0 или 1):

- 1) первая компонента – x_i входного вектора X ;
- 2) вторая компонента – реакция распознающего слоя $p_i = r_k t_{ki}$;
- 3) третья компонента – вход от Пр1 (один и тот же сигнал подается на все нейроны этого слоя).

Чтобы получить на выходе нейрона единичное значение, как минимум два из трех его входов должны равняться единице; в противном случае его выход будет нулевым, т.е. реализуется правило «2 из 3». Первоначально выходной сигнал Пр1 $G1=1$ обеспечивает один из необходимых для возбуждения нейронов входов, а все компоненты вектора R установлены в 0; следовательно, в этот момент вектор C идентичен двоичному входному вектору X .

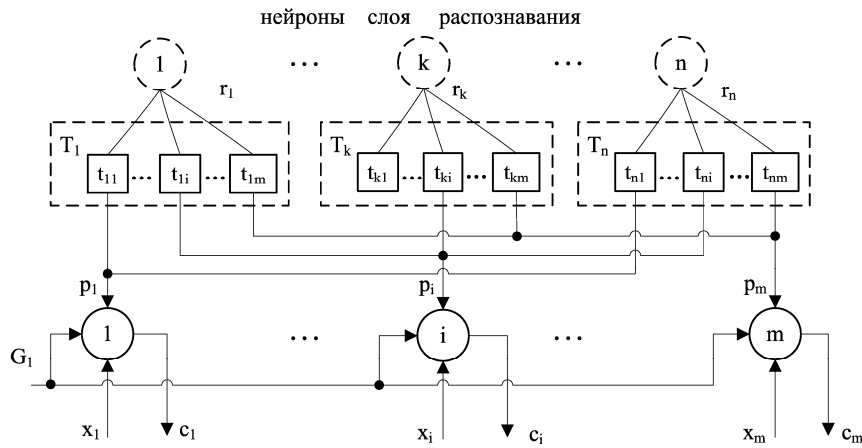


Рис. 6.2. Слой сравнения

Слой распознавания. Слой распознавания осуществляет классификацию входных векторов. Число нейронов этого слоя соответствует числу запомненных образов n (категорий классификации). Значение n увеличивается по мере запоминания новых категорий. В каждый момент времени в слое распознавания возбуждается только один нейрон с наибольшим уровнем активации. На выходе этого нейрона устанавливается единичный уровень сигнала $r_k = 1$. Все остальные нейроны заторможены и имеют нулевой уровень сигнала на выходе: $r_i = 0$, где $i \neq k$. Это правило называется «победитель забирает все».

Каждый нейрон k в слое распознавания имеет соответствующий вектор весов B_k (рис. 6.3). Эти веса являются действительными числами, а не двоичными величинами. Нейрон k имеет максимальную реакцию, если вектор C , являющийся выходом слоя сравнения, наилучшим образом соответствует набору его весов. Следовательно, веса B_k представляют запомненный образ или экземпляр для категории входных векторов. Двоичная версия этого образа также запоминается в соответствующем наборе весов – T_k (см. рис. 6.2).

В процессе функционирования каждый нейрон слоя распознавания вычисляет свертку вектора собственных весов B_k и входного вектора C . Нейрон, имеющий веса, наиболее близкие вектору C ,

будет иметь самый большой выход, тем самым выигрывая соревнование у всех остальных нейронов в слое. Конкуренция между нейронами, как правило, реализуется введением связей с отрицательными весами l_{ik} с выхода каждого нейрона r_k на входы остальных нейронов. Таким образом, если нейрон имеет большой выходной сигнал, то он тормозит все остальные нейроны в слое. Кроме того, каждый нейрон имеет связь с положительным весом со своего выхода на свой собственный вход. Если нейрон имеет на выходе большой уровень сигнала, то эта обратная связь стремится усилить и поддержать его. Для простоты обратные связи между нейронами не показаны на схеме слоя распознавания (рис. 6.3).

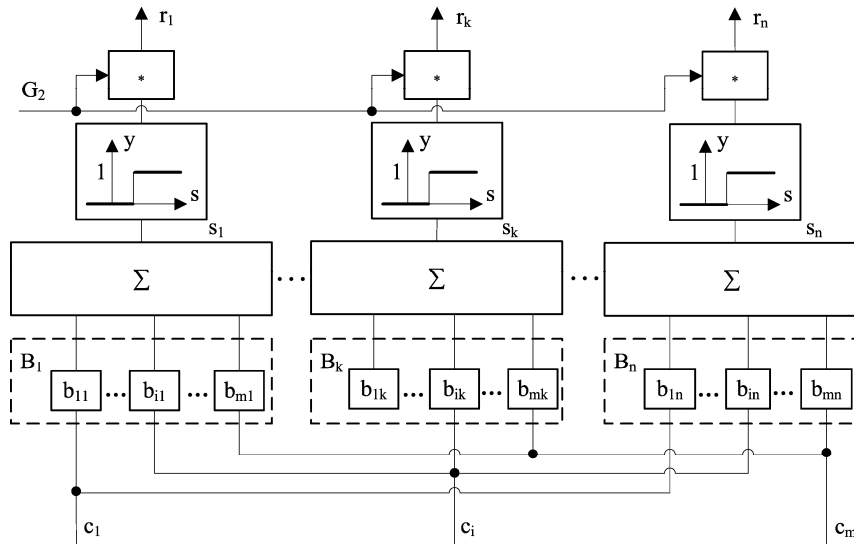


Рис. 6.3. Слой распознавания

Модули приемников и сброса. Выходной сигнал Пр1 $G1=1$, если хотя бы одна компонента x_i двоичного входного вектора X равна единице. Однако если хотя бы одна компонента r_k вектора R равна единице, то $G1=0$. Правила формирования сигнала $G1$ приведены в табл. 6.1.

Выходной сигнал Пр2 $G2=1$, если входной вектор X имеет хотя бы одну единичную компоненту $x_i=1$. Точнее, сигнал $G2$ является логическим «ИЛИ» от компонент вектора X (табл.6.1).

Т а б л и ц а 6.1

Правила формирования сигналов приемников

Логическое «ИЛИ»		G1	G2
от компонент X $x_1 \text{ or } \dots x_i \text{ or } \dots x_m$	от компонент R $r_1 \text{ or } \dots r_k \text{ or } \dots r_n$		
0	0	0	0
1	0	1	1
1	1	0	1
0	1	0	0

Модуль сброса измеряет сходство между векторами X и C . Мера сходства ρ вычисляется как отношение количества совпавших компонент в векторах C и X к количеству компонент m во входном векторе X . Если значение ρ меньше заданного порогового уровня, то вырабатывается сигнал сброса возбужденного нейрона в слое распознавания, т.е. выходной сигнал победившего нейрона обнуляется: $r_k = 0$.

6.3. Функционирование сети АРТ

В процессе классификации образов можно выделить пять основных фаз: инициализацию, распознавание, сравнение, поиск и обучение.

Инициализация. Перед началом процесса обучения сети АРТ все весовые векторы B_k и T_k , несвязанных нейронов, а также параметр сходства ρ должны быть установлены в начальные значения. Всем весам векторов B_k присваиваются одинаковые малые значения. Эти значения должны удовлетворять условию:

$$b_{ik} < L / (L - 1 + m) \quad \text{для всех } i, k, \quad (6.1)$$

где m – количество компонент входного вектора, L – константа, (обычно $L \geq 2$). Такая установка весов гарантирует, что несвязанные нейроны не будут возбуждены более, нежели уже обученные в слое распознавания. Всем весам векторов T_k присваиваются единичные значения, так что

$$t_{ki} = 1 \quad \text{для всех } k, i. \quad (6.2)$$

Значение параметра сходства ρ устанавливается в диапазоне от нуля до единицы в зависимости от требуемой степени сходства

между запомненным образом и входным вектором. При высоких значениях ρ сеть относит к одному классу только очень слабо различающиеся образы. Малое значение ρ заставляет сеть группировать образы, которые имеют слабое сходство между собой. В некоторых случаях можно изменять параметр сходства в процессе обучения. Вначале обеспечивается грубая классификация, а затем значение ρ постепенно увеличивается для выработки точной классификации к концу обучения.

Распознавание. В начальный момент времени вектор X отсутствует на входе сети. Следовательно, все компоненты входного вектора X можно рассматривать как нулевые: $x_i = 0$. Тем самым сигнал Пр2 устанавливается в $G2=0$ и, следовательно, выходы всех нейронов слоя распознавания $r_k = 0$ (см. рис. 6.3). Поскольку все нейроны слоя распознавания начинают работу в одинаковом состоянии, они имеют равные шансы выиграть в последующей конкуренции.

Затем на вход слоя сравнения подается вектор X , который должен быть классифицирован. Этот вектор должен иметь одну или более компонент, отличных от нуля $x_i \neq 0$, в результате чего сигнал Пр2 принимает значение $G2=1$. На этом этапе выходной вектор слоя распознавания отсутствует ($R = 0$), поэтому сигнал Пр1 принимает значения $G1=1$ (табл. 6.1), обеспечивая все нейроны слоя сравнения с одним из двух входов с единичным значением, необходимым для их возбуждения в соответствии с правилом «2 из 3». Тогда любая компонента вектора X , равная единице $x_i=1$, обеспечивает второй единичный вход. В результате соответствующий нейрон слоя сравнения возбуждается и его выход принимает единичное значение: $r_k=1$. Таким образом, в течение этой фазы вектор C в точности дублирует вектор X .

Каждый нейрон k в слое распознавания вычисляет свертку вектора его весов B_k и вектора C – выхода нейронов слоя сравнения (в этот момент $C=X$):

$$s_k = (B_k \cdot C), \quad (6.3)$$

где s_k – возбуждение нейрона k в слое распознавания.

Активационная функция нейронов слоя распознавания является пороговой, определяемой следующим образом:

$$y_k = 1, \text{ если } s_k \geq \theta, \quad y_k = 0, \text{ если } s_k < \theta, \quad (6.4)$$

где θ представляет собой порог.

Процесс конкуренции между нейронами существует, но игнорируется здесь для сохранения простоты выражения. В итоге только нейрон с максимальным значением s_k будет иметь выход, равный единице: $y_k=1$; все остальные нейроны будут иметь нулевой выход. Сеть АРТ запоминает образы в весах нейронов слоя распознавания, один нейрон для каждой категории классификации. Нейрон слоя распознавания, веса которого наилучшим образом соответствуют входному вектору C , возбуждается. Его выход устанавливается в единичное значение, а выходы остальных нейронов этого слоя устанавливаются в ноль.

Сравнение. Единственный возбужденный в слое распознавания нейрон возвращает единицу обратно в слой сравнения $r_k=1$. Эта единственная единица может быть визуально представлена в виде «вверного» выхода, подающегося через связь с весом t_{ki} на вход каждого нейрона в слое сравнения: $p_i = r_k t_{ki}$. Алгоритмы инициализации и обучения построены таким образом, что каждый весовой вектор T_k имеет двоичные значения весов. Это означает, что все компоненты p_i также являются двоичными величинами ($p_i=0$ или $p_i=1$).

Сигнал обратной связи r_k от слоя распознавания устанавливает значение $G=0$. Правило «2 из 3» позволяет возбуждаться только тем нейронам, которые имеют равные единице соответствующие компоненты векторов p_i и x_i . Другими словами, обратная связь от распознающего слоя действует таким образом, чтобы установить компоненты $c_i = 0$ в том случае, если входной вектор X не соответствует запомненному нейроном слоя распознавания образу P , т.е. если $p_i \neq x_i$. Поэтому в общем случае на этой фазе $C \neq X$.

Блок сброса сравнивает вектор C и входной вектор X , вырабатывая сигнал сброса, когда их сходство ρ ниже заданного порога. Вычисление параметра сходства может осуществляться следующим образом:

$$\rho = l/m, \quad (6.5)$$

где l – количество совпавших компонент в векторах X и C , m – размерность входного вектора X .

Значение ρ может изменяться от единицы (наилучшее соответствие) до нуля (наихудшее соответствие). Если степень сходства векторов X и C меньше заданного порогового уровня, вырабатывается сигнал сброса. Он устанавливает выход возбужденного нейрона в ноль ($r_k=0$), отключая его на время текущей классификации.

Поиск. Если значение параметра сходства ρ выигравшего нейрона превышает пороговый уровень, поиск не требуется, процесс классификации на этом 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 \mathbf{R} в ноль, тогда $GI=1$ и вектор \mathbf{C} опять становится идентичным входному вектору \mathbf{X} . В результате другой нейрон выигрывает соревнование в слое распознавания и другой запомненный образ \mathbf{P} возвращается в слой сравнения. Если \mathbf{P} не соответствует \mathbf{X} , возбужденный нейрон в слое распознавания снова тормозится. Этот процесс повторяется до тех пор, пока не произойдет одно из двух событий:

1) будет найден запомненный образ \mathbf{P} , сходство которого с вектором \mathbf{X} будет выше порогового уровня. Тогда проводится переобучение возбужденного нейрона, в процессе которого модифицируются вектора его весов T_k и B_k ;

2) будут проверены все запомненные образы и определено, что они не соответствуют входному вектору \mathbf{X} . Тогда вводится новый несвязанный нейрон в слое распознавания. Он выделяется для запоминания входного образа \mathbf{X} , и его весовые векторы T_k и B_k устанавливаются соответствующим образом.

Обучение. Обучающий алгоритм используется в случае как успешного, так и неуспешного поиска. Вектору весов B_k возбужденного нейрона k в распознающем слое присваиваются новые значения:

$$b_i = \frac{Lc_i}{L-1 + \sum c_i}, \quad (6.6)$$

где c_i – i -я компонента выходного вектора слоя сравнения; k – номер выигравшего нейрона в слое распознавания; b_{ik} – вес связи, соединяющей нейрон i в слое сравнения с нейроном k в слое распознавания; $L = \text{const}$ (обычно $L = 2$).

Компонентам вектора весов T_k , присваиваются соответствующие двоичные значения компонент вектора \mathbf{C} :

$$t_{ki} = c_i, \quad \text{для всех } i, \quad (6.7)$$

где t_{ki} является весом связи между выигравшим нейроном k в слое распознавания и нейроном i в слое сравнения.

6.4. Пример обучения

Рассмотрим типичный сеанс обучения сети АРТ. Для простоты будем считать, что фаза поиска при обучении отсутствует. Если параметр сходства возбужденного нейрона меньше заданного порога или в конкурентной борьбе выигрывает нераспределенный нейрон, то для запоминания входного образа в слой распознавания добавляется новый нейрон. Зададим пороговый уровень сходства на уровне $\rho_n = 0,8$.

Обучающие образы являются произвольными символами, состоящими из квадратов (рис. 6.4). Размерность каждого символа 3×3 . Закрашенный квадрат представляет компоненту вектора X с единичным значением ($x_i=1$), незакрашенный квадрат – с нулевым значением ($x_j=0$). Тогда сеть обучается на примере четырех векторов (табл. 6.2), которые последовательно предъявляются сети на вход слоя сравнения.

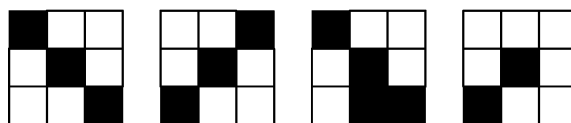


Рис. 6.4. Обучающие примеры

Т а б л и ц а 6.2

Обучающие примеры

i	1	2	3	4	5	6	7	8	9
X_1	1	0	0	0	1	0	0	0	1
X_2	0	0	1	0	1	0	1	0	0
X_3	1	0	0	0	1	0	0	1	1
X_4	0	0	0	0	1	0	1	0	0

Векторы весов B_n и T_n несвязанного нейрона вычисляются в соответствии с (6.1) и (6.2). Если константа $L=2$, тогда $b_{nk} \leq 1/5$, например, $b_{nk}=1/6$. В итоге $T_n=(1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1)$, $B_n=(1/6 \ 1/6 \ 1/6 \ 1/6 \ 1/6 \ 1/6 \ 1/6 \ 1/6)$.

Первый цикл обучения. Сети предъявляется обучающий вектор X_1 . В слое распознавания присутствует только один нераспределенный нейрон, который активизируется входным вектором. Для

запоминания первой категории в слое распознавания выделяем новый (первый) нейрон и обучаем его в соответствии с правилами (6.6) и (6.7). Определим сумму единичных компонент в векторе $C=X_1$: $\sum c_i = 3$, тогда веса первого обученного нейрона будут иметь следующие значения:

$$T_1=(1\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 1), B_1=(1/4\ 0\ 0\ 0\ 1/4\ 0\ 0\ 0\ 1/4).$$

Второй цикл обучения. Сети предъявляется обучающий вектор X_2 . Свертка (6.3), осуществляемая нераспределенным нейроном (s_n) и первым обученным нейроном (s_1) в слое распознавания, дает значения, приведенные в табл. 6.3.

Т а б л и ц а 6.3

Вычисление свертки

i	1	2	3	4	5	6	7	8	9
C_2	0	0	1	0	1	0	1	0	0
B_1	1/4	0	0	0	1/4	0	0	0	1/4
$C_2 \cdot B_1$	0	0	0	0	1/4	0	0	0	0
B_n	1/6	1/6	1/6	1/6	1/6	1/6	1/6	1/6	1/6
$C_2 \cdot B_n$	0	0	1/6	0	1/6	0	1/6	0	0

В результате $s_1 = (C_2 \cdot B_1) = 1/4$, $s_n = (C_2 \cdot B_n) = 1/2$. Победителем оказывается нераспределенный нейрон, что автоматически приводит к добавлению нового (второго) нейрона в слой распознавания. Его веса настраиваются для запоминания второй категории по правилам (6.6) и (6.7). Сумма единичных компонент в векторе $C=X_2$: $\sum c_i = 3$, тогда веса второго обученного нейрона будут иметь следующие значения:

$$T_2=(0\ 0\ 1\ 0\ 1\ 0\ 1\ 0\ 0), B_2=(0\ 0\ 1/4\ 0\ 1/4\ 0\ 1/4\ 0\ 0).$$

Третий цикл обучения. Сети предъявляется обучающий вектор X_3 . Свертка (6.3), осуществляемая нераспределенным нейроном (s_n), первым (s_1) и вторым (s_2) обученными нейронами, дает следующие значения:

$$s_1 = (C_3 \cdot B_1) = 3 \cdot 1/4 = 3/4, s_2 = (C_3 \cdot B_2) = 1/4, s_n = (C_3 \cdot B_n) = 4 \cdot 1/6 = 2/3.$$

Победителем становится первый нейрон. Вычисляется значение ρ для определения его сходства с входным вектором X_3 (табл. 6.4).

Т а б л и ц а 6.4

Вычисление сходства с X_3

i	1	2	3	4	5	6	7	8	9
$P=T_1$	1	0	0	0	1	0	0	0	1
X_3	1	0	0	0	1	0	0	1	1
C_3	1	0	0	0	1	0	0	0	1

Компоненты вектора C_3 определяются по правилу двух третей. Количество совпавших компонент в векторах X_3 и C_3 $l=8$, тогда мера сходства, определяемая по (6.5), $\rho=8/9>\rho_n$. Пороговый уровень превышен, происходит переобучение весов первого нейрона в соответствии со значениями компонент вектора C_3 . Новые значения весов в этом случае повторяют старые. Единичная компонента $x_8=1$ в векторе X_3 не запоминается, так как она отсутствовала в ранее запомненной первой категории:

$$T_1=(1\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 1), B_1=(1/4\ 0\ 0\ 0\ 1/4\ 0\ 0\ 0\ 1/4).$$

Четвертый цикл обучения. Сети предъявляется обучающий вектор X_4 . Свертка (6.3), осуществляемая нераспределенным (s_n), первым (s_1) и вторым (s_2) обученными нейронами, дает следующие значения:

$$s_1=(C_4 \cdot B_1)=1/4, s_2=(C_4 \cdot B_2)=2 \cdot 1/4=1/2, s_n=(C_4 \cdot B_n)=2 \cdot 1/6=1/3.$$

Победителем становится второй нейрон. Вычисляется значение ρ для определения его сходства с входным вектором X_4 (табл. 6.5).

Т а б л и ц а 6.5

Вычисление сходства с X_4

i	1	2	3	4	5	6	7	8	9
$P=T_2$	0	0	1	0	1	0	1	0	0
X_4	0	0	0	0	1	0	1	0	0
C_4	0	0	0	0	1	0	1	0	0

Количество совпавших компонент в векторах X_3 и C_3 $l=9$, тогда мера сходства, определяемая по (6.5), $\rho=1>\rho_n$. Для переобучения весов первого нейрона используются компоненты вектора C_4 . Новые значения весов:

$$T_2=(0\ 0\ 0\ 0\ 1\ 0\ 1\ 0\ 0), B_2=(0\ 0\ 0\ 0\ 1/3\ 0\ 1/3\ 0\ 0).$$

Единиичная компонента $x_3=1$, ранее присутствовавшая во второй категории, теперь не фиксируется, так как входной вектор X_4 ее не содержит.

Итогом обучения сети является запоминание двух категорий образов:

$$T_1 = (1 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 1), B_1 = (1/4 \ 0 \ 0 \ 0 \ 1/4 \ 0 \ 0 \ 0 \ 1/4);$$

$$T_2 = (0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0 \ 0), B_2 = (0 \ 0 \ 0 \ 0 \ 1/3 \ 0 \ 1/3 \ 0 \ 0),$$

что соответствует символам (рис. 6.5).

Рассматриваемый способ обучения приводит к «сокращению» компонент запомненных векторов, которые не соответствуют входным векторам. Если вес однажды установлен в ноль, обучающий алгоритм никогда не восстановит его единичное значение. Таким образом, запомненный вектор представляет собой логическое пересечение всех обучающих векторов и может включать существенные характеристики данной категории. Новый вектор, включающий только существенные характеристики, будет соответствовать этой категории.

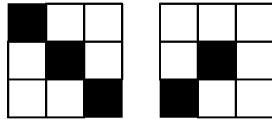


Рис. 6.5. Результат обучения

6.5. Свойства сетей АРТ

Сети АРТ обучаются посредством изменения весов таким образом, что предъявление сети входного вектора заставляет ее активировать нейроны в слое распознавания, связанные со сходным запомненным вектором. Кроме этого, обучение проводится в форме, не разрушающей запомненные ранее образы, предотвращая тем самым временную нестабильность. Эта задача решается путем выбора критерия сходства. Новый входной образ (который сети раньше не предъявлялся) не будет соответствовать ранее запомненным образам по параметру сходства, тем самым формируя новую запоминаемую категорию образов. С другой стороны, входной образ, схожий с одним из запомненных, не будет формировать новую категорию, он просто будет модифицировать веса того нейрона, на который он похож. Таким образом, при соответствующем выборе критерия сходства предотвращается временная нестабильность и искажение ранее изученных образов.

К достоинствам сети АРТ можно отнести следующие:

1) быстрый доступ к предварительно запомненным образам, обусловленный тем, что активизация нужного нейрона в слое распознавания не требует запуска процедуры поиска;

2) устойчивость процесса обучения, затрагивающего только один возбужденный нейрон;

3) конечный характер процесса обучения. Любая последовательность произвольных входных векторов будет приводить к стабильному набору весов после конечного числа обучающих циклов. Повторяющиеся последовательности обучающих векторов не будут приводить к циклическому изменению весов сети.

Но у сети АРТ есть и несколько существенных недостатков:

1) чувствительность к порядку предъявления векторов. Запоминаемые категории, сформированные сетью, принципиально меняются при различном порядке предъявления входных образов;

2) невозможность классификации зашумленных векторов. Если сетью сформирована правильная категория, к которой относится вектор X , то, как только его компонента x_i примет нулевое значение за счет шума (если векторы предъявляются неоднократно), соответствующая компонента категории также будет обнулена. Так как случайное нулевое значение может принять любая компонента X , то с течением времени все компоненты категории будут обнулены, а запомненная информация о ней утрачена. Если после этого предъявить незашумленный вариант вектора X , то для него будет выделен новый нейрон, т.е. сформирована новая категория. Это явление называется размножением категорий;

3) организация памяти по принципу «бабушкиных узелков». Каждый «узелок» – нейрон слоя распознавания – отвечает за хранение одной категории образов, что лишает память сети АРТ основного преимущества человеческой памяти – свойства распределенности. Мозг человека может пережить значительные повреждения без существенной потери памяти, за счет ее распределения по нейронам.

7. АССОЦИАТИВНАЯ ПАМЯТЬ

Компьютерная память является локально адресуемой, данные извлекаются по известному адресу ячейки памяти. Никакой связи между адресом и содержанием самих данных не существует. Такая адресация предполагает пассивность данных в процессе поиска.

Это обстоятельство затрудняет поиск данных с частично известным содержанием.

Человеческая память ассоциативна, т.е. некоторое воспоминание может порождать множество образов, связанных с ним устойчивой связью. Например, несколько музыкальных тактов могут вызвать целую гамму воспоминаний, включая образы знакомых людей, пейзажи, звуки и т.п.

Память, способная восстанавливать полную информацию по ее достаточно представительной части (ключу), называется содержательно-адресованной или ассоциативной. Когда образы, создающие ассоциацию, являются одинаковыми, то память называется автоассоциативной, а если ассоциируемые образы разные, то память называется гетероассоциативной. Если использовать нейросетевую терминологию, то можно сказать, что ассоциативная память определяет взаимную зависимость векторов запоминаемых образов. Когда на взаимозависимость исследуются компоненты одного и того же входного вектора, говорят об автоассоциативной памяти. Если же взаимозависимыми оказываются два различных вектора, X и Y , можно говорить о памяти гетероассоциативного типа.

Для реализации ассоциативной памяти используются рекуррентные ИНС.

7.1. Рекуррентные сети

Отдельную группу нейронных сетей составляют сети с обратной связью между различными слоями нейронов. Это так называемые рекуррентные сети. Их общая черта состоит в передаче сигналов с выходного или скрытого слоя во входной слой. Таким образом, данные пропускаются через одну и ту же сеть многократно. Новое качество, присущее рекуррентным сетям, – динамическая обработка информации. Изменение состояния одного нейрона отражается на всей сети вследствие обратной связи типа «один ко многим». В сети возникает переходный процесс, который завершается формированием нового устойчивого состояния. Для обеспечения сходимости процесса к устойчивому состоянию сеть должна отвечать критериям устойчивости.

Если функцию активации i -го нейрона обозначить $f(s_i)$, где s_i – взвешенная сумма его входов x_j , то состояние нейрона можно определить выходным сигналом $y_i = f(s_i)$. При наличии обратных свя-

зей роль входов x_j для нейрона играют выходные сигналы других нейронов. Тогда изменение состояния нейрона может быть описано системой дифференциальных уравнений:

$$\tau_i \frac{ds_i}{dt} = \sum_{j=1}^N w_{ij} f(s_j) - s_i - \theta_i, \quad i = 1, 2, \dots, N,$$

где θ_i – пороговое значение, коэффициент τ_i – численная константа, аналог постоянной времени. Состояние нейрона определяется в результате решения дифференциального уравнения $y_i = f(s_i)$.

При определенном уровне возбуждения нейронов, описываемом значениями их выходных сигналов y_i , с рекуррентной сетью можно сопоставить энергетическую функцию $E(y_i, y_j, w_{ij})$, по аналогии с энергетической функцией Ляпунова. Ее значение определяется состояниями всех нейронов сети и имеет тенденцию к убыванию с течением времени. Изменение состояния какого-либо нейрона вызывает изменение энергетического состояния всей сети в направлении минимума ее энергии. Обычно существует множество локальных энергетических минимумов, каждый из которых представляет одно из состояний сети, сформированных в результате обучения. В пространстве состояний локальные энергетические минимумы E представлены точками стабильности, называемыми аттракторами из-за тяготения к ним ближайшего окружения.

7.2. Сети автоассоциативной памяти

Сети Хопфилда. Сеть Хопфилда (рис. 7.1) состоит из единственного слоя нейронов, число которых является одновременно числом входов и выходов сети. Каждый нейрон связан со всеми остальными нейронами, а также имеет один вход, через который осуществляется ввод сигнала.

Сети Хопфилда имеют ряд отличительных признаков:

- обратные связи в сети идут с выходов нейронов на их входы по принципу «со всех на все»;

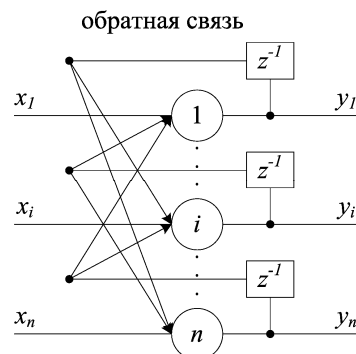


Рис. 7.1. Сеть Хопфилда

- каждый нейрон связывается со всеми другими нейронами, но не связывается с самим собой;
- обучение сети сводится к расчету весовых коэффициентов нейронов на основе обучающих примеров перед началом функционирования сети, без итеративного процесса подстройки;
- выходные состояния нейронов ограничены двумя значениями (например, -1 и 1);
- при предъявлении входного вектора состояние сети «сходится» к одному из запомненных в сети эталонов, представляющих множество устойчивых состояний, характеризующихся локальными минимумами энергетической функции.

Задача, решаемая сетью Хопфилда в качестве ассоциативной памяти, как правило, формулируется следующим образом. Известен некоторый набор эталонных двоичных сигналов (изображений, звуковых оцифровок и т.п.), которые являются для сети обучающей выборкой. Сеть должна уметь из произвольного неидеального сигнала, поданного на ее вход, выделить ("вспомнить" по частичной информации) соответствующий эталонный образец (если такой есть) или "дать заключение" о том, что входные данные не соответствуют ни одному из известных ей эталонов. В общем случае любой входной сигнал может быть описан вектором $X = \{x_1, \dots, x_i, \dots, x_n\}$, где n – число нейронов в сети и размерность входных и выходных векторов. Каждый элемент x_i равен либо +1, либо -1. Обозначим вектор, описывающий k -й эталон, через X_k , а его компоненты, соответственно, – x_{ik} , $k=1\dots P$, где P – число эталонов. Когда сеть распознаёт (т.е. "вспомнит") какой-либо эталонный образец на основе предъявленного произвольного вектора X , то на ее выходах будут сформированы компоненты именно этого эталона, т.е. $Y = X_k$, где Y – вектор выходных значений сети: $Y = \{y_1 \dots y_i \dots y_n\}$. В противном случае выходной вектор не совпадет ни с одним эталоном.

Например, если сигналы представляют собой некие изображения, то, отобразив в графическом виде данные с выхода сети, можно будет увидеть картинку, полностью совпадающую с одним из эталонов (в случае успеха) или же "вольную импровизацию" сети (в случае неудачи).

Динамическое изменение состояний сети может осуществляться синхронно и асинхронно. В первом случае состояния всех нейронов модифицируются одновременно на каждом временном

шаге, во втором случае в каждый момент времени выбирается один нейрон и изменяется его состояние. Выбор нейрона может осуществляться случайным образом. Будем рассматривать синхронный режим работы сети Хопфилда.

В режиме обучения сети весовые коэффициенты обратных связей устанавливаются следующим образом:

$$w_{ij} = \begin{cases} \sum_{k=1}^P x_{ik} x_{jk}, & i \neq j, \\ 0, & i = j, \end{cases} \quad (7.1)$$

где i и j – индексы соответственно предсинаптического и постсинаптического нейронов; x_{ik} , x_{jk} – i -й и j -й элементы вектора k -го образца.

В режиме распознавания алгоритм функционирования сети следующий (t – номер итерации).

1. На входы сети подается неизвестный сигнал. Фактически его ввод осуществляется непосредственной установкой значений выходов $y_i(t=0)$: $y_i(0) = x_i$, где $i = 1 \dots n$, поэтому обозначение на схеме сети входных сигналов в явном виде носит чисто условный характер.

2. Рассчитываются новые состояния нейронов:

$s_j(t+1) = \sum_{i=1}^n w_{ij} y_i(t)$, где $j=1 \dots n$, и новые значения их выходов

$y_j(t+1) = f[s_j(t+1)]$, где f – активационная сигнатурная функция, приведенная на рис. 2.2,а. Иначе, без учета единичных временных задержек z^{-1} сети, представляющих способ синхронизации процес-

са передачи сигналов, $y_j(t+1) = \text{sgn}(\sum_{i=1}^n w_{ij} y_i(t))$.

3. Проверяется, изменились ли выходные значения нейронов y_i за последнюю итерацию. Если да, переходят к п. 2. Иначе, если выходы застabilizировались $y_i(t+1)=y_i(t)$ для всех $i = 1 \dots n$, процесс распознавания завершен. При этом выходной вектор Y представляет собой образец, наилучшим образом ассоциированный с входными данными.

Иногда сеть не может провести распознавание и выдает на выходе образ, который не содержится среди эталонных образцов. Это связано с ограниченной емкостью памяти сети. Для сети Хоп-

филда число запоминаемых образов $P \leq 0,15 \cdot n$. Кроме того, если два образа A и B сильно похожи, они, возможно, будут вызывать у сети перекрестные ассоциации, т.е. предъявление на входы сети вектора A приведет к появлению на ее выходах вектора B и наоборот.

Требование достаточного условия слабой коррелируемости образов можно представить следующим неравенством:

$$\sum_{k \neq j}^P |(X_k, X_j)| < n.$$

Состояние сети – это множество текущих значений сигналов s_i всех нейронов. В первоначальной сети Хопфилда состояние каждого нейрона менялось в дискретные случайные моменты времени, в последующей работе состояния нейронов могли меняться одновременно. Так как выходом бинарного нейрона может быть только $y_i = 1$ или $y_i = -1$ (промежуточных уровней нет), то текущее состояние сети является двоичным числом, каждый бит которого – это сигнал y_i некоторого нейрона.

Функционированию сети можно дать геометрическую интерпретацию. На рис. 7.2,а показана интерпретация для сети из двух нейронов. Каждой вершине квадрата соответствует одно из четырех возможных состояний сети (-1-1, -11, 1-1, 11). На рис. 7.2,б показана интерпретация для трехнейронной сети, представленная кубом, имеющим восемь вершин. Каждой вершине соответствует одно из возможных состояний сети. В общем случае сеть с n нейронами имеет 2^n различных состояний и может быть представлена n -мерным гиперкубом.

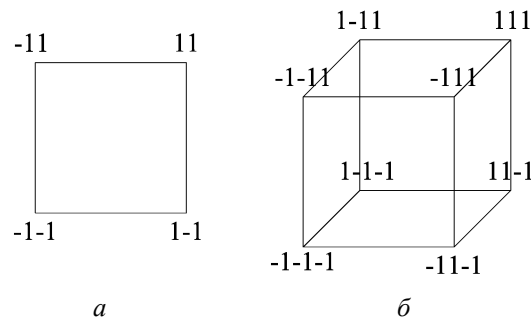


Рис. 7.2. Геометрическая интерпретация состояний сети Хопфилда

Когда подается новый входной вектор, сеть переходит из вершины в вершину, пока не стабилизируется. Устойчивая вершина определяется сетевыми весами, текущими входами и величиной порога. Если входной вектор частично неправилен или неполон, то сеть стабилизируется в вершине, ближайшей к желаемой.

Для сети Хопфилда доказана теорема, определяющая достаточное условие устойчивости сетей с обратными связями, а именно, сеть с обратными связями устойчива, если матрица ее весов симметрична ($w_{ij}=w_{ji}$) и имеет нули на главной диагонали ($w_{ii}=0$).

Каждый эталон, запоминаемый сетью, является точкой из конечного множества равновесных точек, характеризующих минимум функции Ляпунова:

$$E = -\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n w_{ij} y_i y_j - \sum_{j=1}^n x_j y_j + \sum_{j=1}^n \theta_j y_j, \quad (7.2)$$

где E – искусственная энергия сети; w_{ij} – вес от выхода i -го нейрона ко входу j -го; x_j, y_j – вход и выход j -го нейрона; θ_j – порог j -го нейрона.

Главное свойство энергетической функции состоит в том, что в процессе изменения состояний нейронной сети значение E уменьшается и достигает локального минимума (аттрактора), в котором сохраняет постоянную энергию. Это позволяет решать задачи комбинаторной оптимизации, если они могут быть сформулированы как задачи минимизации энергии.

В режиме распознавания в результате одной итерации происходит изменение состояний всех нейронов сети. Пусть Δy_j – изменение выхода j -го нейрона после итерации. Общее приращение энергии сети ΔE , вызванное изменением состояния всех нейронов, составит:

$$\Delta E = -\sum_{j=1}^n \left(\sum_{i=1}^n w_{ij} y_i + x_j - \theta_j \right) \Delta y_j = -\sum_{j=1}^n (s_j - \theta_j) \Delta y_j = -\sum_{j=1}^n \Delta E_j.$$

Допустим, что взвешенная сумма s_j , вычисляемая j -м нейроном, $s_j \geq \theta_j$. Тогда выражение в скобках будет положительным, а выходное значение $y_j = 1$ (см. рис. 2.2), т.е. y_j может либо измениться в положительную сторону $\Delta y_j = 2$, либо остаться без изменения $\Delta y_j = 0$. Это означает, что приращение энергии сети ΔE_j , обусловленное изменением состояния любого j -го нейрона, должно

быть отрицательным или нулевым. Следовательно, энергия сети должна либо уменьшиться, либо остаться без изменения. Далее допустим, что $s_j < \theta_j$, и тогда $\Delta E_j \leq 0$. Следовательно, и в этом случае энергия сети должна уменьшиться или остаться без изменения.

Таким образом, любое изменение состояния нейрона либо уменьшит энергию, либо оставит ее без изменения. Благодаря такому непрерывному стремлению к уменьшению энергия сети в конце концов должна достигнуть минимума и прекратить изменение. По определению такая сеть является устойчивой.

На рис. 7.3 демонстрируется функционирование сети Хопфилда на примере эталонного набора из пяти цифр, представленных в пиксельной форме размерностью 7×7 . Поэтому количество нейронов сети Хопфилда составляет $n = 49$, а количество обучающих примеров $P = 5$.

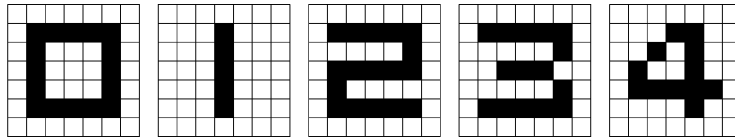


Рис. 7.3. Эталонные образы, используемые для обучения сети Хопфилда

На этапе обучения использовались идеальные (незашумленные) образы, дающие безошибочное восстановление. Обученная сеть подвергалась тестированию на 10 зашумленных образах, показанных на рис. 7.4.

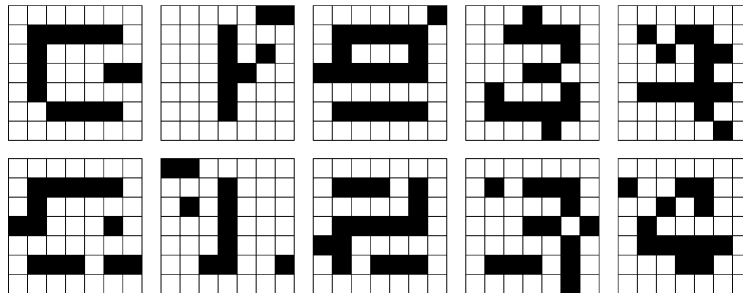


Рис. 7.4. Образы, используемые для тестирования сети Хопфилда

Сеть не всегда безошибочно распознает эталонный образ. В обучающем наборе некоторые образы сильно коррелированы, вследствие чего процесс распознавания завершается в точках локальных минимумов, далеких от эталона.

Ассоциативная память может быть реализована и другими способами, без использования нейронных сетей. Для этого достаточно последовательно сравнить входной образ X со всеми предварительно запомненными эталонами X_k . Вычислив хэммингово расстояние для каждой пары (X, X_k) , нужно выбрать тот эталон, до которого это расстояние минимально. Однако сеть Хопфилда позволяет исключить перебор эталонов и осуществить эту процедуру параллельным способом, при котором время выборки из памяти не увеличивается с ростом числа запомненных образов.

Сеть Хэмминга. Сеть Хэмминга используется в том случае, когда можно ограничиться определением номера эталона, ближайшего к предъявленному входному образу. Поскольку сеть Хэмминга в явном виде не формирует распознаваемый эталонный образ, то ее характерной чертой является меньшее количество используемых нейронов и меньший объем вычислений по сравнению с сетью Хопфилда.

Сеть состоит из двух слоев (рис. 7.5). Первый и второй слои имеют по P нейронов, где P – число запоминаемых эталонных образов. Нейроны первого слоя содержат по n связей, соединенных со всеми входами сети, где n – размерность векторов входных образов.

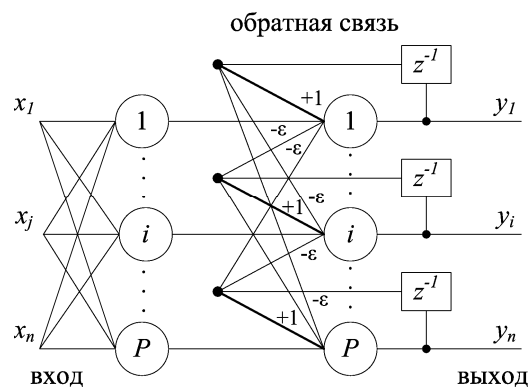


Рис. 7.5. Сеть Хэмминга

Нейроны второго слоя связаны между собой отрицательными обратными связями. Единственная положительная обратная связь каждого нейрона соединяет его вход с его же выходом. Эта связь имеет единичный весовой коэффициент.

Идея работы сети состоит в нахождении расстояния Хэмминга от тестируемого образа до всех эталонов. Расстоянием Хэмминга называется число отличающихся битов в двух бинарных векторах. Сеть должна выбрать эталон с минимальным расстоянием Хэмминга до неизвестного входного сигнала. В результате должен быть активирован только один выход сети, соответствующий номеру этого эталона.

В режиме обучения сети весовым коэффициентам первого слоя и порогу активационной функции присваиваются следующие значения: $w_{ik}^{(1)} = x_{ik}$, $i=1...n$, $k=1...P$, где x_{ik} – i -й элемент k -го эталона.

Весовые коэффициенты отрицательных обратных связей во втором слое устанавливают равными некоторой величине: $0 < \varepsilon < 1/(P-1)$.

В режиме распознавания сеть Хэмминга функционирует следующим образом.

1. На первом этапе на входы сети подается неизвестный вектор $X = \{x_1, \dots, x_i, \dots, x_n\}$, исходя из которого рассчитываются состояния нейронов первого слоя (верхний индекс в скобках указывает номер слоя):

$$s_j^{(1)} = \sum_{i=1}^n w_{ij}^{(1)} x_i, \quad j=0...P.$$

Нейроны первого слоя вычисляют расстояние Хэмминга между фактически предъявленным входным вектором X и каждым из P закодированных эталонов X_k , образующих веса нейронов первого слоя. Значения выходных сигналов этих нейронов определяются соотношением $y_i = 1 - d_n(X, X_k)/n$, где $d_n(X, X_k)$ – расстояние Хэмминга между входным вектором X и эталоном X_k , т.е. количество битов, на которое различаются эти два вектора. Значение $y_i = 1$, если $X = X_k$, и $y_i = 0$, если $X \neq X_k$. В остальных случаях y_i принимают значения в интервале $[0, 1]$.

После этого полученными значениями инициализируются значения выходов второго слоя сети: $y_j^{(2)} = y_j^{(1)}$, где $j = 1...P$. Роль первого слоя весьма условна: воспользовавшись один раз на пер-

вом этапе значениями его весовых коэффициентов, сеть больше не обращается к нему.

2. На втором этапе иницирующие сигналы $y_i^{(1)}$ удаляются и из сформированного ими начального состояния запускается итерационный процесс внутри второго слоя. Для обеспечения сходимости алгоритма веса $w_{ij}^{(2)}$ второго слоя должны отличаться друг от друга: $w_{ij}^{(2)} = \xi - 1/(P-1)$, где ξ – малая случайная величина, $i \neq j$, t – номер итерации.

Вычисляются новые состояния нейронов второго слоя:

$$s_j^{(2)}(t+1) = y_j(t) - \sum_{k=1}^P w_{kj}^{(2)} y_k^{(2)}(t), \quad k \neq j, \quad j=1 \dots P,$$

и значения их выходов: $y_j^{(2)}(t+1) = f[s_j^{(2)}(t+1)]$, $j=1 \dots P$.

Активационная функция f имеет полулинейный вид и задается выражением

$$f(s_j) = \begin{cases} s_j, & \text{для } s_j \geq 0, \\ 0, & \text{для } s_j < 0. \end{cases}$$

Итерационный процесс завершается в момент, когда все нейроны, кроме одного (победителя с выходным значением $y_j^{(2)}=1$), перейдут в нулевое состояние.

Нейроны во втором слое выбирают вектор с наименьшим расстоянием Хэмминга, определяя таким образом номер класса, к которому принадлежит предъявленный входной вектор X .

7.3. Двухнаправленная ассоциативная память

Оригинальная ДАП. Ассоциативная память, реализуемая сетью Хопфилда, автоассоциативна: исходный образ может быть завершен или исправлен, но не может быть ассоциирован с другим образом. Данный факт является результатом одноуровневой структуры ассоциативной памяти, в которой вектор появляется на выходе тех же нейронов, на которые поступает входной вектор.

Двухнаправленная ассоциативная память (ДАП) гетероассоциативна: входной вектор поступает на один набор нейронов, а соответствующий выходной вырабатывается на другом наборе. Как и сеть Хопфилда, ДАП способна к обобщению, вырабатывая правильные реакции, несмотря на искаженные входы. Кроме того, могут быть реализованы адаптивные версии ДАП, выделяющие

эталонный образ из зашумленных экземпляров. Эти возможности сильно напоминают процесс мышления человека.

Сигналы в сети (рис. 7.6) распространяются в двух направлениях: от входа к выходу и по обратной связи обратно к входу. Сеть функционирует в синхронном режиме. Это означает, что если в первом цикле сигналы вначале проходят в одну сторону для определения состояния нейронов второго слоя, то в следующем цикле они сами становятся источником, посылающим сигналы на нейроны первого слоя. Этот процесс повторяется до достижения равновесного состояния.

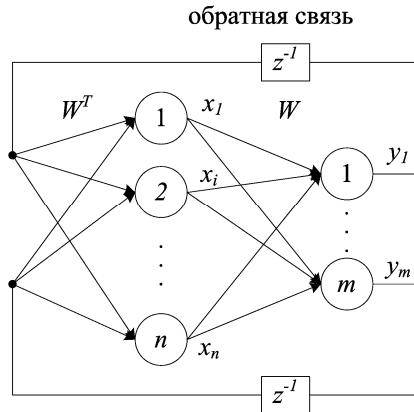


Рис. 7.6. Структура сети ДАН

Функция активации нейронов имеет вид симметричной сигмоиды:

$$y_i = f(s_i) = \frac{\exp(as_i) - \exp(-as_i)}{\exp(as_i) + \exp(-as_i)},$$

где y_i – выход i -го нейрона, s_i – взвешенная сумма входных сигналов i -го нейрона, a – константа, определяющая степень наклона кривой.

В простейших версиях ДАН значение константы a выбирается большим, в результате чего функция активации приближается к простой пороговой функции. Далее будем предполагать, что используется пороговая функция активации (см. рис. 2.2,а), причем при нулевом сигнале возбуждения нейрона его текущее состояние остается равным предыдущему. Таким образом, поведение i -го нейрона может быть описано следующими правилами:

$$\begin{aligned}
y_i(t+1) &= 1, \text{ если } s_i(t) > 0, \\
y_i(t+1) &= -1, \text{ если } s_i(t) < 0, \\
y_i(t+1) &= y_i(t), \text{ если } s_i(t) = 0,
\end{aligned}$$

где t – номер такта синхронизации.

Режим обучения сети ДАП заключается в установке расчетных значений весовых коэффициентов в обоих слоях сети. На этом этапе запоминаемые сетью ассоциированные эталонные образы кодируются биполярными значениями. Векторная компонента эталона, большая нуля, становится $+1$, а компонента, меньшая или равная нулю, становится -1 .

Матрица весов W , связывающая оба слоя сети, является действительной. С учетом симметрии связей входного и выходного слоев сети, при прямом направлении распространения сигналов веса описываются матрицей W , а при противоположном направлении – матрицей W^T .

Предположим, что набор эталонных данных определен в виде множества из P ассоциированных биполярных пар векторов $\{(X_k, Y_k)\}$, $X_k = [x_{k1}, x_{k2}, \dots, x_{kn}]$, $Y_k = [y_{k1}, y_{k2}, \dots, y_{km}]$, где n и m – размерность векторов X_k и Y_k соответственно. По правилу, предложенному Б. Коско, матрица весов W формируется как матрица корреляции:

$$W = \sum_{k=1}^P X_k^T Y_k. \quad (7.3)$$

В режиме распознавания на вход одного из слоев сети подается вектор X_0 или Y_0 . Если допустить, что начальное состояние сети было задано вектором X_0 , то процесс двунаправленной обработки сигналов состоит из последовательных циклов:

$$f(X_0 W) = Y_1 \rightarrow f(Y_1 W^T) = X_1 \rightarrow f(X_1 W) = Y_2 \rightarrow \dots \rightarrow f(Y_f W^T) = X_f \rightarrow f(X_f W) = Y_f \quad (7.4)$$

В результате формируются две стабильные величины: X_f и Y_f . При выполнении определенных условий парой (X_f, Y_f) становится одна из обучающих пар, участвующих в формировании матрицы W , которая наиболее близка по мере Хэмминга паре (X_0, Y_0) , с которой начался процесс восстановления ассоциации. При этом с каждой промежуточной точкой процесса (X_k, Y_k) можно сопоставить энергетическую функцию по Ляпунову – E_k , определяемую в виде $E_k = -X_k W Y_k^T$.

Каждое очередное изменение состояния переходного процесса ведет к уменьшению значения E_k вплоть до достижения локально-

го минимума. Этот минимум достигается за конечное количество итераций и составляет: $E_{\min} = -X_f W Y_f^T$.

В качестве примера рассмотрим обучение сети ДАП, имеющей четыре входа ($n=4$) и пять выходов ($m=5$). Задача сети состоит в запоминании множества из пяти ($P=5$) ассоциированных векторов $\{(X_k, Y_k)\}$, заданных в биполярной форме. Обучающие векторы сгруппированы в приведенные ниже матрицы X и Y . Каждая строка матрицы X представляет собой один обучающий эталонный вектор, сопряженный с соответствующей строкой матрицы Y :

$$X = \begin{bmatrix} 1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 \\ -1 & -1 & 1 & 1 \\ -1 & -1 & -1 & 1 \\ -1 & -1 & -1 & -1 \end{bmatrix}, \quad Y = \begin{bmatrix} -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & 1 \\ -1 & -1 & -1 & 1 & 1 \\ -1 & -1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 & 1 \end{bmatrix}.$$

Матрица весовых коэффициентов сети, сформированная согласно (7.3): $W = X_1^T Y_1 + X_5^T Y_5 + X_5^T Y_5 + X_5^T Y_5 + X_5^T Y_5$, имеет вид

$$W = \begin{bmatrix} 3 & 1 & -1 & -3 & -5 \\ 1 & -1 & -3 & -5 & -3 \\ -1 & -3 & -5 & -3 & -1 \\ -3 & -5 & -3 & -1 & 1 \end{bmatrix}.$$

В режиме распознавания, при начальных значениях векторов, совпадающих с эталонами, сеть распознает их безошибочно. Допустим, на вход сети подан вектор, совпадающий с первой строкой матрицы X , т.е. $X_0 = [1 \ 1 \ 1 \ 1]$, а состояния нейронов второго слоя соответствуют сопряженному вектору $Y_0 = [-1 \ -1 \ -1 \ -1 \ -1]$. Тогда процесс обработки сигналов в сети завершится за один такт:

$$Y_1 = f(X_0 W) = f([0 \ -8 \ -12 \ -12 \ -8]) = [-1 \ -1 \ -1 \ -1 \ -1],$$

$$X_1 = f(Y_1 W^T) = f([5 \ 11 \ 13 \ 11]) = [1 \ 1 \ 1 \ 1],$$

$$Y_2 = f(X_1 W) = f([0 \ -8 \ -12 \ -12 \ -8]) = [-1 \ -1 \ -1 \ -1 \ -1].$$

Сеть приходит к устойчивому состоянию: $X_f = [1 \ 1 \ 1 \ 1]$, $Y_f = [-1 \ -1 \ -1 \ -1 \ -1]$, соответствующему первой эталонной ассоциации.

Значения энергетической функции, соответствующие конечному состоянию сети, для соответствующих ассоциированных векторов равны: $E_1 = -X_1 W Y_1^T = -40$, $E_2 = -X_2 W Y_2^T = -34$, $E_3 = -X_3 W Y_3^T = -32$ и т.д.

Сеть ДАП обладает способностью к обобщению. Например, если незавершенный или частично искаженный вектор подается в качестве X_0 , сеть имеет тенденцию к выработке запомненного вектора Y_k , который, в свою очередь, стремится исправить ошибки в X_0 . Иначе говоря, при искажении эталонных значений векторов X_k и Y_k сеть может откорректировать эти векторы и восстановить эталонную ассоциацию. Допустим, на вход сети подан вектор, отличающийся одной компонентой от первой строки матрицы X , т.е. $X_0 = [1 \ -1 \ 1 \ 1]$, тогда

$$Y_1 = f(X_0 W) = f([-2 \ -6 \ -6 \ -2 \ -2]) = [-1 \ -1 \ -1 \ -1 \ -1],$$

$$X_1 = f(Y_1 W^T) = f([5 \ 11 \ 13 \ 11]) = [1 \ 1 \ 1 \ 1],$$

$$Y_2 = f(X_1 W) = f([0 \ -8 \ -12 \ -12 \ -8]) = [-1 \ -1 \ -1 \ -1 \ -1].$$

Сеть приходит к устойчивому состоянию, восстанавливая первую эталонную ассоциацию.

При искажении эталонных значений векторов X_k и Y_k , сеть, обученная по алгоритму Б. Коско, не всегда способна откорректировать эти векторы и распознает их с некоторыми погрешностями. Удовлетворительное качество распознавания можно получить при выполнении условия, накладывающего ограничение на количество запоминаемых сетью эталонов:

$$P < \sqrt{\min(n, m)}. \quad (7.5)$$

Системы с обратной связью могут иметь тенденцию к колебаниям; это означает, что они могут переходить от состояния к состоянию, никогда не достигая стабильности. Доказано, что сеть ДАП стабильна при любых значениях весов сети. Это важное свойство возникает из отношения транспонирования между двумя весовыми матрицами – W и W^T , т.е. любой набор ассоциаций может быть изучен без риска возникновения неустойчивости в сети.

Модифицированный алгоритм обучения сети ДАП. Алгоритм обучения сети ДАП по правилу Б. Коско прост и эффективен при выполнении ограничительных условий, но обладает рядом недостатков. Если сопоставленная с k -й эталонной парой векторов

(X_k, Y_k) энергия E_k не составляет локальный минимум, то эта пара не может быть распознана сетью даже тогда, когда начальные значения $(X_0, Y_0) = (X_k, Y_k)$. Кроме того, распознавание будет давать плохие результаты, если в процессе обучения используются не похожие друг на друга векторы. Например, с подобными векторами X_k сопоставляются не подобные друг другу векторы Y_k , при этом степень подобия векторов измеряется расстоянием Хэмминга $d_H(X_i, X_j)$, т.е. исследуется степень выполнения условия $d_H(X_i, X_j)/n = d_H(Y_i, Y_j)/m$ для всех значений i и j .

Существует алгоритм обучения, обеспечивающий распознавание эталонной пары векторов (X_k, Y_k) , независимо от того, соответствует ли ей локальный минимум энергетической функции E_k или нет. Вместо выражения (7.3) предлагается использовать формулу

$$W = \sum_{k=1}^P X_k^T Y_k + (q-1) X_k^T Y_k. \quad (7.6)$$

Поправочный компонент $(q-1)X_k^T Y_k$ равнозначен увеличению участия пары (X_k, Y_k) в процессе обучения. Поправки можно добавлять для каждой пары, не соответствующей условиям критерия минимума функции E_k : $W(t+1) = W(t) + (q-1)X_k^T Y_k$, где $W(t)$ – матрица весовых коэффициентов, полученная на цикле обучения t . В энергетическом смысле введенная поправка уменьшает значение функции E_k для пары векторов (X_k, Y_k) . Значение коэффициента q подбирается экспериментально.

Модификация структуры сети ДАП. Введение поправочных компонентов в алгоритм обучения сети ДАП не обеспечивает абсолютную безошибочность распознавания эталонных пар векторов. Полную достоверность распознавания эталонов из искаженного шумом вектора гарантирует только модификация структуры сети.

Матрица весовых коэффициентов W при передаче сигнала от X к Y заменяется расширенной матрицей $W_f = [W | W_y]$, а матрица W^T при передаче сигнала от Y к X – матрицей $W_b = [W^T | W_x]$. Вводимые поправки разрушают симметрию передачи сигналов в противоположных направлениях. Дополнительные матрицы W_x и W_y формируются так, что при нормальной работе алгоритма Б. Коско их влияние нивелируется; они включаются в работу при возникновении ошибок распознавания.

Обозначим через n' и m' количество обучающих пар, для которых в процессе распознавания получены неправильные ответы для

векторов Y и X соответственно. Если (X_k, Y_k) является очередной k -й эталонной парой ($k=1, 2, \dots, P$), для которой $f(X_k W) \neq Y_k$, то принимается $y'_{ki} = 1$, $y'_{kj} = 0$ для $j \neq i$ ($i=1, 2, \dots, n'$). Если для (X_k, Y_k) выполняется условие $f(X_k W) = Y_k$, то $y'_{ki} = 0$ для $i = 1, 2, \dots, n'$. Таким образом, компоненты y'_{ki} образуют вектор Y'_k длиной n' . Корректирующие матрицы W_x и W_y формируются по следующим правилам:

$$W_y = \sum_{k=1}^P X_k^T Y'_k, \quad W_x = \sum_{k=1}^P Y_k^T X'_k.$$

Затем создаются матрицы дополнительных узлов сети T_x и T_y :

$$T_y = \sum_{k=1}^P q_y Y_k'^T Y_k, \quad T_x = \sum_{k=1}^P q_x X_k'^T X_k.$$

Параметры q_x и q_y подбираются таким образом, чтобы они соответствовали условиям

$$q_x > n(P-2) - 2 \min_i \left\{ \sum_{j \neq i} d_H(X_i, X_j) \right\}, \quad q_y > m(P-2) - 2 \min_i \left\{ \sum_{j \neq i} d_H(Y_i, Y_j) \right\}.$$

Модифицированная структура сети ДАП, в которой учитываются связи через матрицы T_x и T_y , представлена на рис. 7.7. Дополнительные нейроны увеличивают размерность сети и корректируют неточности функционирования связей, задаваемых матрицей W . После предъявления на вход сети пары векторов (X_0, Y_0) начинается рекуррентный процесс, аналогичный (7.4) и приводящий к стабильным значениям (X_f, Y_f) .

Описание процесса должно содержать дополнительно введенные связи (см. рис. 7.7). При использовании введенных обозначений получим

$$\begin{aligned} Y_1 &= f(X_0 W + g_y(X_0 W_y) T_y), \\ X_1 &= f(Y_1 W^T + g_x(Y_1 W_x) T_x), \\ &\dots \\ Y_f &= f(X_{f-1} W + g_y(X_{f-1} W_y) T_y), \end{aligned}$$

где $g_x()$ и $g_y()$ – векторы функций активации дополнительных корректирующих нейронов.

Функции активации имеют пороговый вид (см. рис. 2.2,б), причем значение порога θ выбирается для каждого слоя индивидуально:

$$\begin{aligned} g_x(s) &= [g_x(s_1), g_x(s_2), \dots, g_x(s_{n'})], \\ g_y(s) &= [g_y(s_1), g_y(s_2), \dots, g_y(s_m)], \end{aligned}$$

где элементы векторов $g_x(s)$ и $g_y(s)$ определяются выражениями

$$g_x(s_i) = \begin{cases} 1, & \text{если } s_i > n - \varepsilon_1, \\ 0, & \text{если } s_i \leq n - \varepsilon_1, \end{cases} \quad g_y(s_i) = \begin{cases} 1, & \text{если } s_i > m - \varepsilon_2, \\ 0, & \text{если } s_i \leq m - \varepsilon_2. \end{cases}$$

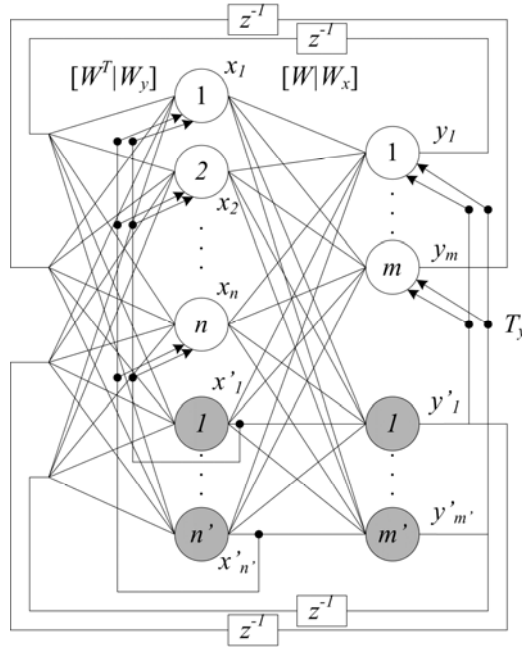


Рис. 7.7. Структура модифицированной ДАП

Величины ε_1 и ε_2 должны удовлетворять следующим условиям:

$$0 < \varepsilon_1 < 2 \min_{i \neq j} \{ d_H(X_i, X_j) \}, \quad 0 < \varepsilon_2 < 2 \min_{i \neq j} \{ d_H(Y_i, Y_j) \}.$$

При такой модификации сеть ДАП всегда обеспечивает хорошее распознавание запомненных эталонных пар векторов (X_k, Y_k) , независимо от того, соответствуют ли им локальные минимумы энергетической функции E_k или нет. Большое значение для правильного восстановления эталонов, искаженных шумом, имеет правильный выбор параметров ε_1 и ε_2 . Недостатком метода является зависимость ε_1 и ε_2 от уровня шума. Значения, подобранные оптимальным образом для одного уровня шума, необязательно будут эффективны при его изменении.

На рис. 7.8 показаны последовательные циклы функционирования модифицированной сети ДАП на примере распознавания

зашумленного эталонного образа типа «кораблик» и ассоциированного с ним эталонного образа – надписи “ship”. Обучающие данные, составляющие векторы X и Y , формировались на базе пиксельных шаблонов. Размерность вектора X –288, вектора Y –280.



Рис. 7.8. Последовательность распознавания искаженных эталонных образов модифицированной сетью ДАП

В отличие от оригинальной сети ДАП, имеющей ограничения на емкость памяти (7.5), модифицированная ДАП не имеет ограничений на распознаваемость входных векторов. Однако это достигается за счет увеличения размерности сети и количества межнейронных связей.

Существует много разновидностей сетей ДАП, основными из которых являются непрерывная ДАП (с сигмоидными функциями

активации нейронов), адаптивная (с изменяющимися весами в процессе функционирования, конкурирующая (с конкуренцией нейронов внутри каждого слоя).

8. САМООРГАНИЗУЮЩИЕСЯ СЕТИ

Основу самоорганизации нейронных сетей составляет подмеченная закономерность, заключающаяся в том, что общее упорядочение весовых коэффициентов сети становится возможным в результате операций самообучения, проводящихся независимо друг от друга в различных локальных сегментах сети. Векторы входных образов активизируют определенные нейроны сети, которые адаптируются к обучающим примерам путем изменения своих синаптических весов. В процессе обучения наблюдается тенденция к росту значений весовых коэффициентов. Создается положительная обратная связь: большие значения входных воздействий вызывают увеличение весов активизированных нейронов, что в свою очередь вызывает большую активность нейронов. При этом происходит естественное расслоение нейронов на различные группы. Отдельные нейроны или их группы сотрудничают между собой и активизируются в ответ на возбуждение, создаваемое конкретными обучающими выборками, подавляя своей активностью другие нейроны. При этом можно говорить как о сотрудничестве между нейронами внутри группы, так и о конкуренции между нейронами внутри группы и между различными группами. Важное значение для обучения имеет избыточность исходных данных, без которых обучение просто невозможно.

8.1. Самообучение на основе конкуренции

В самом простом варианте самоорганизующаяся ИНС может быть однослойной сетью. Ассоциированное с каждым нейроном множество весов соединяет его с каждым входом (рис. 8.1). Подобно нейронам большинства сетей каждый i -й нейрон вычисляет сумму взвешенных входов (s_i):

$$s_i = \sum_k x_k w_{ki} , \quad (8.1)$$

где x_k – k -я составляющая входного вектора X , w_{ki} – вес связи, соединяющей k -й вход с i -м нейроном.

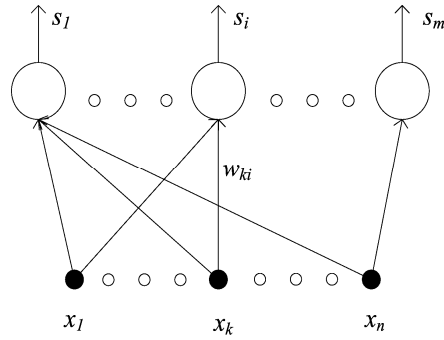


Рис. 8.1. Самообучающийся слой

Нейрон с максимальным значением s_i является «победителем». Его выход равен единице, выходные значения остальных нейронов равны нулю.

Такая сеть классифицирует входные векторы в группы схожих. Это достигается с помощью такой подстройки весов, что близкие входные векторы активируют один и тот же нейрон. В данном случае обучение протекает без учителя. Поэтому трудно предсказать, какой именно нейрон сети будет активироваться заданным входным вектором. Необходимо лишь гарантировать, чтобы в результате обучения разделялись несхожие входные векторы.

Предварительная обработка входных векторов. Перед тем, как предъявлять сети входные векторы, их следует нормализовать. Для этого выполняется деление каждой компоненты входного вектора на его длину. Длина вектора вычисляется извлечением квадратного корня из суммы квадратов его компонент:

$$x'_i = x_i / \sqrt{\sum_{k=1}^n x_k^2}.$$

Нормализованный вектор x'_i единичной длины ориентирован в n -мерном пространстве так же, как и исходный входной вектор. На рис. 8.2 приведен пример нормализации двумерного вектора ($n=2$).

На рис. 8.3 показано несколько единичных двумерных векторов, что справедливо для сети с двумя входами. Они оканчиваются на окружности единичного радиуса. В случае сети с тремя входами векторы оканчивались бы на поверхности единичной сферы. Эти представления могут быть перенесены на сети, имеющие произ-

вольное число входов (n), где каждый входной вектор оканчивается на поверхности единичной гиперсферы. Эта абстракция не допускает непосредственной визуализации.

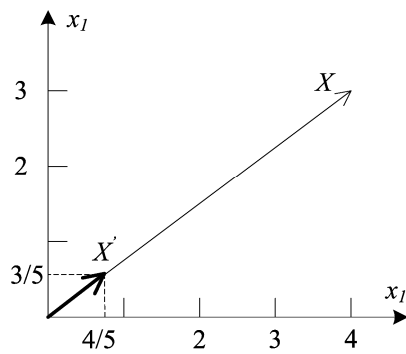


Рис. 8.2. Нормализация двумерного вектора

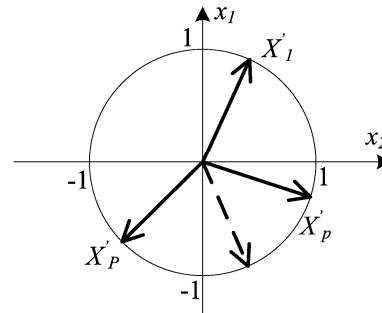


Рис. 8.3. Двумерные нормализованные входные векторы

Правило самообучения. Всем весам сети перед началом обучения следует придать случайные малые значения. Эти случайно выбранные весовые векторы W следует нормализовать так же, как и входные векторы X .

При обучении самоорганизующейся сети на вход подается входной вектор X и вычисляются его скалярные произведения с векторами весов, связанными со всеми нейронами. Нейрон с максимальным значением скалярного произведения объявляется «победителем», и его веса подстраиваются. При нормализованных векторах X и W скалярное произведение, используемое для вычисления взвешенных сумм нейронов (8.1) s_i , является мерой сходства между векторами X и W . Процесс обучения состоит в выборе нейрона с весовым вектором, наиболее близким к входному, и в дальнейшем приближении весового вектора W к входному X . Сеть самоорганизуется таким образом, чтобы нейрон-победитель увеличивал значение взвешенной суммы s_i для данного входного вектора. Уравнение, описывающее процесс обучения, имеет следующий вид:

$$W(t+1) = W(t) + \mu(X - W(t)), \quad (8.2)$$

где $W(t+1)$ – новое значение вектора весов победившего нейрона, X – вектор входных весов; $W(t)$ – предыдущее значение вектора

весов; μ – коэффициент скорости обучения, который может варьироваться в процессе обучения.

Каждый вес, связанный с выигравшим нейроном, изменяется пропорционально разности между его величиной и величиной компоненты входа, с которой он связан. Направление изменения минимизирует разность между весом и его входом.

На рис. 8.4 этот процесс показан геометрически для двумерного случая – сети с двумя входами. Сначала находится разность векторов $X-W$, для этого строится вектор из конца W в конец X . Затем этот вектор укорачивается умножением его на скалярную величину $\mu < 1$, в результате чего получается вектор изменения $\delta = \mu(X-W)$. Тогда новый весовой вектор X будет

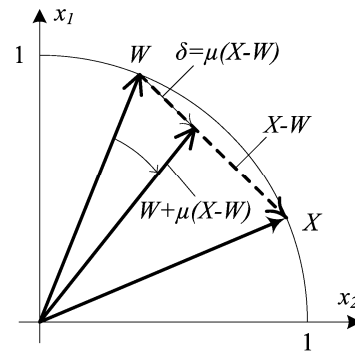


Рис. 8.4. Подстройка весов нейрона

направлен из начала координат в конец вектора δ . Очевидно, что эффект обучения состоит во вращении весового вектора в направлении входного без существенного изменения его длины.

Если бы с каждым нейроном ассоциировался один входной вектор, то сеть могла бы быть обучена с помощью одного вычисления на вес. Веса нейрона-победителя приравнивались бы к компонентам обучающего вектора ($\mu = 1$). Как правило, обучающее множество включает много сходных между собой входных векторов, и сеть должна быть обучена активировать один и тот же нейрон для каждого из них. В этом случае веса этого нейрона должны получаться усреднением тех входных векторов, которые должны его активировать. Таким образом, веса, ассоциированные с нейроном, примут значение вблизи «центра» входных векторов, для которых данный нейрон является «победителем».

Проблема «мертвых» нейронов. При инициализации сети ее весам присваиваются случайные значения, распределенные по равномерному закону. Это может породить серьезные проблемы при обучении, так как весовые векторы сети распределяются равномерно по поверхности гипербферы. Входные векторы, как правило, распределены неравномерно и имеют тенденцию группиро-

ваться на относительно малой части поверхности гиперсферы. В итоге большинство весовых векторов W_m будут так удалены от любого входного вектора X , что связанные с ними нейроны никогда не будут иметь шансов на победу. Их весовые векторы не будут корректироваться, поэтому такие нейроны остаются «мертвыми». На рис. 8.5 приведен пример возможной ситуации для сети с двумя входами. Нейроны с весовыми векторами W_i и W_j активируются каждый своей группой векторов. Нейрон с весовым вектором W_m будет «мертвым».

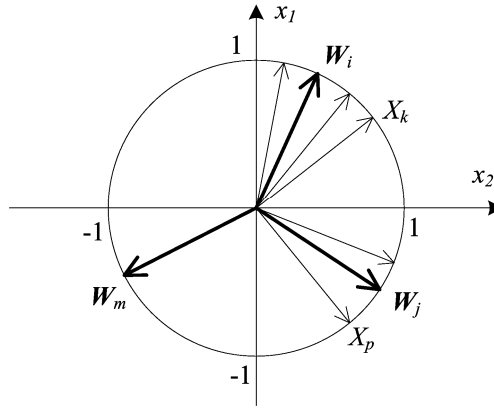


Рис. 8.5. Проблема мертвых нейронов

Более того, оставшихся нейронов, имеющих наилучшие соответствия своих весовых векторов W с некоторыми входными векторами X , может оказаться слишком мало, чтобы разделить эти входные векторы на классы, которые расположены близко друг к другу на поверхности гиперсферы. Такая сеть будет иметь большую погрешность в интерпретации данных, иначе называемую погрешностью квантования.

8.2. Самоорганизующиеся карты Кохонена

Идея сети принадлежит финскому ученому Тойво Кохонену. Отличительная особенность работы самоорганизующихся карт (Self-Organizing Maps, SOM) – введение в правило обучения нейронов информации о его расположении относительно других

нейронов сети. Здесь имеет место аналогия со свойствами коры головного мозга человека. Она обладает определенными топологическими свойствами. Например, участок, ответственный за кисть руки, примыкает к участку, ответственному за движения всей руки, и таким образом вся картина человеческого тела «отображается» на поверхность коры.

Карты Кохонена подразумевают использование упорядоченной структуры нейронов. Сеть состоит из одного слоя нейронов, располагающихся в узлах двумерной сетки с прямоугольными или шестиугольными ячейками. Каждый нейрон соединен со всеми компонентами n -мерного входного вектора X (рис. 8.6). При этом каждому i -му нейрону соответствует n -мерный вектор весовых коэффициентов $W_i = [w_{1i}, w_{2i}, \dots, w_{ni}]$.

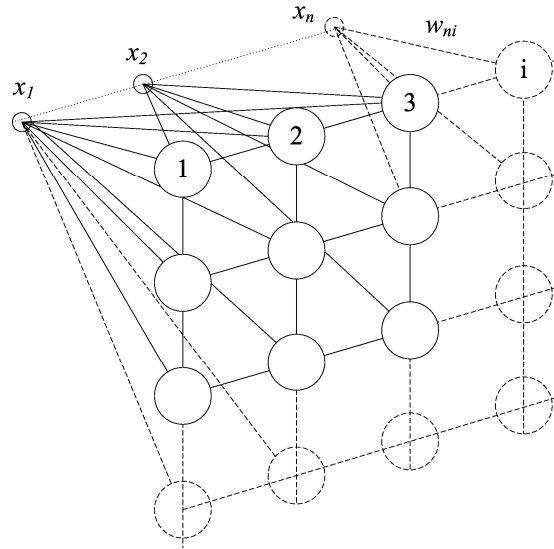


Рис. 8.6. Самоорганизующаяся карта Кохонена

Свойство топологической упорядоченности достигается в сети с помощью дополнительного использования понятия окрестности. Окрестность – это несколько ближайших нейронов, окружающих выигравший нейрон. Радиус окрестности определяет степень взаимодействия нейронов друг с другом. На рис. 8.7 приведены примеры окрестностей с разным радиусом R для четырехугольной и

гексагональной сеток. Можно заметить, что для гексагональной сетки окрестность нейрона больше совпадает с понятием евклидова расстояния между нейронами, чем для четырехугольной.

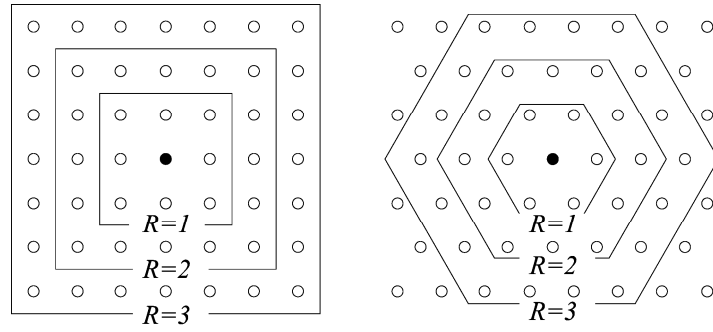


Рис. 8.7. Окрестности нейрона для четырехугольной и гексагональной сеток

На этапе инициализации сети SOM определяется конфигурация сетки (прямоугольная или шестиугольная), задается количество нейронов в сети, выбирается начальный радиус их обучения. Затем производится инициализация случайными значениями весовых коэффициентов нейронов и последующая нормализация их весовых векторов.

Векторы входных образов также нормализуются. После активации сети вектором X в конкурентной борьбе побеждает тот нейрон, веса которого W_j в наименьшей степени отличаются от соответствующих компонент этого вектора. Для j -го нейрона победителя выполняется соотношение $d(X, W_j) = \min_{i \in (1, N)} d(X, W_i)$,

где $d(X, W_j)$ – расстояние между векторами X и W_j , а N – количество нейронов в сети. В качестве меры расстояния между векторами X и W_j чаще всего используются евклидова мера:

$$d(X, W_j) = \sqrt{\sum_{i=1}^n (x_i - w_{ji})^2}, \quad (8.3)$$

скалярное произведение:

$$d(X, W_j) = 1 - |X| |W_j| \cos(X, W_j). \quad (8.4)$$

Вокруг нейрона-победителя образуется топологическая окрестность. Все нейроны, лежащие в пределах его окрестности радиусом R , подвергаются обучению, в ходе которого векторы их весов изменяются в направлении вектора X по правилу (8.2). Значение коэффициента μ уменьшается с увеличением расстояния между i -м нейроном и победителем. Доказано, что при таком обучении функция плотности распределения весовых векторов W_j нейронов сводится к дискретизированной плотности распределения векторов обучающих примеров.

8.3. Обучение

Целью обучения сети SOM считается такое упорядочение нейронов (подбор значений их весов), которое минимизирует погрешность аппроксимации входного вектора X значениями весов нейрона-победителя. Если используется евклидова метрика (8.3) для определения расстояния между векторами X и W_j , а P – количество входных векторов в обучающей выборке, то эта погрешность, называемая также погрешностью квантования E_q , может быть выражена как

$$E_q = \frac{1}{P} \sum_{p=1}^P |X_p - W_{jp}|^2,$$

где W_{jp} – весовой вектор j -го нейрона-победителя при предъявлении сети вектора X_p из обучающей выборки.

Обучение сети состоит из последовательности коррекций весовых векторов нейронов. На каждом шаге обучения из обучающей выборки случайно выбирается один из векторов X , а затем производится поиск ближайшего к нему весового вектора W_j . Для определения расстояния между X и W_j используется метрика (8.3) или (8.4). Таким образом выбирается j -й нейрон-победитель и производится корректировка весов нейронов, т.е. обучение сети.

Процедура корректировки весов нейронов может быть определена обобщенной зависимостью

$$W_i(t+1) = W_i(t) + \mu_i G(i, X)(X - W_i(t)) \quad (8.5)$$

для всех i нейронов, расположенных в окрестности нейрона-победителя. В формуле (8.5) скорость обучения μ_i нейрона i отделена от его расстояния до предъявленного вектора X функцией

$G(i, X)$, называемой функцией соседства. Существует несколько вариантов алгоритмов обучения, различающихся видом функции соседства. Рассмотрим два из них: классический алгоритм Кохонена и алгоритм нейронного газа.

Алгоритм Кохонена. В классическом алгоритме Кохонена сеть инициализируется путем приписывания нейронам определенных позиций в пространстве и связывания их с соседями на постоянной основе. После выбора нейрона-победителя уточняются не только его веса, но также и веса его соседей, находящихся в ближайшей окрестности. Функция соседства определяется функцией гауссовского типа:

$$G(i, X) = \exp\left(-\frac{d^2(i, j)}{2\lambda^2}\right), \quad (8.6)$$

где $d(i, j)$ – расстояние между i -м нейроном и j -нейроном-победителем, λ – уровень соседства. При соседстве гауссовского типа степень адаптации i -го нейрона уменьшается по мере его удаления от нейрона-победителя. Такое обучение еще называется алгоритмом типа WTM (Winner Takes Most – победитель забирает больше).

В классическом алгоритме Кохонена функция соседства определяется в виде

$$G(i, X) = \begin{cases} 1, & d(i, j) \leq R_j, \\ 0, & d(i, j) > R_j, \end{cases}$$

где $d(i, j)$ – расстояние между i -м нейроном и j -м нейроном-победителем, измеряемое количеством нейронов, т.е. величиной радиуса R окрестности нейрона j (см. рис. 8.7); R_j – радиус окрестности нейрона-победителя, внутри которой веса нейронов подвергаются обучению. Соседство такого рода называется прямоугольным.

Частным случаем классического алгоритма Кохонена является обучение, в котором функция соседства

$$G(i, X) = \begin{cases} 1, & i = j, \\ 0, & i \neq j. \end{cases}$$

В этом случае обучается только нейрон-победитель по алгоритму типа WTA (Winner Takes All – победитель забирает все).

Как правило, функция соседства гауссовского типа дает лучшие результаты обучения и обеспечивает лучшую организацию

сети, чем прямоугольное соседство. Значение параметра λ – уровня соседства (радиуса обучения) выбирают достаточно большим на начальном этапе обучения и постепенно уменьшают так, что в конечном итоге обучение происходит по типу WTA. Скорость обучения μ_i также представляет собой, убывающую от времени функцию. Наиболее часто используется функция, обратно пропорциональная времени: $\mu_i(t) = A/(t+B)$, где A и B – константы.

Обучение состоит из двух основных фаз: на первоначальном этапе выбирается достаточно большое значение скорости обучения $\mu_i(t)$ и радиуса обучения λ , что позволяет расположить векторы нейронов в соответствии с распределением примеров в выборке, а затем производится точная подстройка весов, когда значения параметров скорости и радиуса обучения много меньше начальных.

Алгоритм нейронного газа. В этом алгоритме на каждой итерации все нейроны сортируются в зависимости от их расстояния до вектора X . После сортировки нейронам присваивается номер позиции m в сортированном списке:

$$d_1 < d_2 < \dots d_m < \dots d_N, \quad (8.7)$$

где $d_m = |X - W_{im}|$ – расстояние между вектором X и вектором весов W_i i -го нейрона, занимающего в результате сортировки m -ю позицию в последовательности, возглавляемой нейроном-победителем, с которым сопоставлено расстояние d_1 . Значение функции соседства для i -го нейрона определяется по формуле

$$G(i, X) = \exp\left(-\frac{m(i)}{\lambda}\right), \quad (8.8)$$

где $m(i)$ – очередность i -го нейрона, полученная в результате сортировки ($m(i) = 1, 2, \dots, N$), а λ – параметр, аналогичный уровню соседства в алгоритме Кохонена. Алгоритм нейронного газа напоминает стратегию нечетких множеств, в соответствии с которой каждому i -му нейрону присваивается значение функции принадлежности $G(i, X)$ к окрестности нейрона-победителя.

Параметр λ уменьшается в процессе адаптации весов нейронов, сужая окрестность обучения. При $\lambda=0$ адаптации подвергается только нейрон-победитель и алгоритм превращается в обычный алгоритм WTA.

Изменение $\lambda(t)$ в зависимости от номера итерации обучения t можно описать выражением

$$\lambda(t) = \lambda_{\max} (\lambda_{\min} / \lambda_{\max})^{t/t_{\max}},$$

где λ_{\max} и λ_{\min} – принятые максимальное и минимальное значения параметра λ . Коэффициент t_{\max} определяет максимальное количество итераций обучения сети. Коэффициент скорости обучения i -го нейрона μ_i в выражении (8.5) тоже может уменьшаться линейно или показательно по мере увеличения номера итерации обучения t .

Для сокращения объема вычислений, необходимых для реализации алгоритма нейронного газа, можно включать в сортированный список (8.7) только нейроны с наиболее значимой величиной функции $G(i, X)$, вычисляемой по формуле (8.8). Если очередность нейрона $m(i) > 1$, то $G(i, X) \approx 0$. Например, если принять $K=3\lambda$, то при сортировке нейронов и при их обучении можно ограничиться только первыми K элементами.

8.4. Решение проблемы мертвых нейронов

Для решения проблемы мертвых нейронов (см. рис. 8.5) необходимо провести активацию всех нейронов сети, включив их в процесс адаптации весовых коэффициентов. Для этого применяются следующие способы.

Механизм утомления. В алгоритме обучения можно предусмотреть учет количества побед каждого нейрона, а процесс обучения организовать так, чтобы дать шанс победить и менее активным нейронам. Идея такого подхода к обучению возникла при наблюдении за поведением биологических нейронов. Отмечен факт, что нейрон-победитель сразу после возбуждения на некоторое время теряет активность, «отдыхая» перед следующим этапом конкурентной борьбы. Такой способ учета активности нейронов называется механизмом утомления.

Для учета активности нейронов используется метод подсчета потенциала p_i каждого нейрона, значение которого модифицируется всякий раз после предъявления очередной реализации входного вектора X . Потенциал нейрона определяется по следующему правилу:

$$p_i(t+1) = \begin{cases} p(t) + 1/n, & i \neq j, \\ p(t) - p_{\min}, & i = j, \end{cases}$$

где j – нейрон-победитель, p_{\min} – минимальный потенциал, разрешающий участие в конкурентной борьбе. Если фактическое значение

ние потенциала $p_i < p_{\min}$, i -й нейрон «отдыхает», а победитель ищется среди оставшихся нейронов, для которых выполняется условие $p_i > p_{\min}$. Максимальное значение потенциала ограничивается на уровне $p_i = 1$. Выбор значения p_{\min} позволяет установить порог готовности нейрона к конкурентной борьбе. При $p_{\min} = 0$ утомляемость нейронов не возникает и каждый из них сразу же после победы будет готов к продолжению соперничества (классический алгоритм Кохонена). При $p_{\min} = 1$ возникает другая крайность, вследствие которой нейроны побеждают по очереди, так как на каждой итерации обучения только один из них оказывается готов к соперничеству. На практике применяется величина $p_{\min} \approx 0,75$.

Механизм штрафа победителей. Количество побед нейрона учитывается при подсчете эффективного расстояния $d_s(X, W_i)$ между вектором весов W_i и обучающим вектором X . Это расстояние модифицируется пропорционально количеству побед данного нейрона в прошлом. Если обозначить количество побед i -го нейрона P_i , то такую модификацию можно представить в виде

$$d_s(X, W_i) = P_i d(X, W_i),$$

где $d(X, W_i)$ – фактическое расстояние между векторами X и W_i . Активные нейроны с большим значением P_i штрафуются искусственным завышением этого расстояния. Следует отметить, что величина d_s используется только для выявления победителя. При коррекции весового вектора нейрона (8.5) учитывается фактическое расстояние. Модификация расстояния имеет целью активировать все нейроны путем введения их в область с большим количеством данных. После решения этой задачи (обычно после двух или трех циклов обучения) модификация расстояния прекращается, что позволяет продолжить «честную» конкуренцию нейронов.

8.5. Применение сетей Кохонена

Сжатие данных. Главным свойством сети Кохонена считается компрессия данных, состоящая в том, что образующие кластер большие группы данных представляются единственным вектором весов нейрона-победителя. При разделении P исходных обучающих примеров на Q кластеров и представлении каждого кластера одним из N нейронов достигается значительное сокращение объема информации, которое и называется компрессией. Это компрес-

сия с потерей качества, которая определяется погрешностью квантования E_q .

Примером использования компрессионных свойств сети Кохонена может считаться сжатие изображений. Предположим, что изображение размером $K_x \times K_y$ пикселей разделяется на одинаковые кадры размером $k_x \times k_y$ (рис. 8.8).

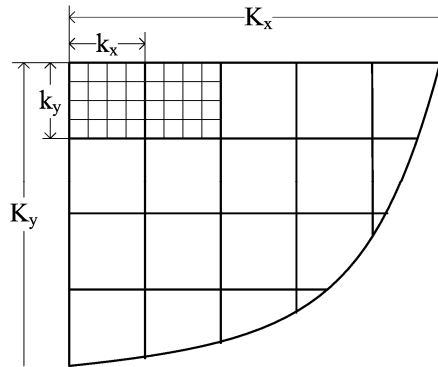


Рис. 8.8. Разделение изображения на кадры

Образующие кадр пиксели представляют собой компоненты входных векторов X . Каждый вектор X состоит из $K = k_x \times k_y$ компонент, каждая компонента x_i определяет интенсивность конкретного пикселя в кадре. Сеть с самоорганизацией содержит N нейронов, каждый из которых связан синаптическими весами со всеми компонентами входного вектора X . В результате обучения формируется структура сети, при которой вектору X каждого кадра соответствует вектор весов нейрона-победителя. При похожих составляющих вектора X для разных кадров побеждать будет один и тот же нейрон либо группа нейронов с близкими векторами W . В процессе предъявления сети очередного кадра выбирается номер нейрона-победителя, например 7, 4, 7, 7, 31 и т.д. Эти номера образуют кодовую последовательность, а весовые векторы W_i нейронов представляют средние значения, соответствующие уровням интенсивности пикселей, составляющих кадр. При обучении сеть сама формирует ядра классов, т.е. набор фрагментов, из которых строится восстановленное изображение. Пусть, например, при использовании двухуровневой (черно-белой) кодировки интенсивности

пиксела в двух нейронах Кохонена запомнены два фрагмента изображения – ядра классов (рис. 8.9, *а, б*) в виде весовых коэффициентов. Предъявим в виде входного вектора фрагмент изображения (рис. 8.9, *в*). Входной вектор больше похож на вектор весов, запомненный вторым нейроном $i = 2$, поэтому он будет активирован. При восстановлении изображения по номеру класса $i=2$ будет восстановлен фрагмент, запомненный вторым нейроном, который не полностью совпадает с исходным фрагментом.

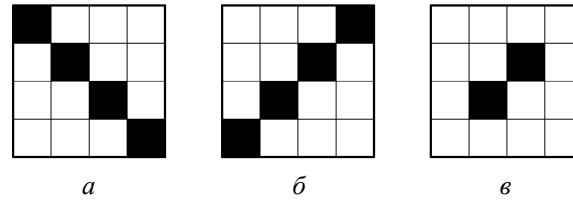


Рис. 8.9. Пример запоминания фрагментов: *а* – нейрон 1, *б* – нейрон 2, *в* – фрагмент на входе сети

Если количество нейронов N в сети намного меньше количества кадров, то можно получить существенную степень сжатия исходного изображения. В итоге коэффициент компрессии изображения определяется как $K_r = N_r k_x k_y T / (N_r \lg_2 N + N k_x k_y t)$, где k_x и k_y – размеры кадра по горизонтали x и вертикали y , N_r – число кадров в исходном изображении, N – число нейронов в сети, T и t число битов, используемых для представления соответственно градаций интенсивности пиксела и значений весов. Чем больше фрагмент и чем меньше нейронов в сети Кохонена, тем выше коэффициент компрессии и тем больше потери при восстановлении.

Этот же метод может быть применен для сжатия других типов данных, например речевых сигналов. Типичные значения коэффициента сжатия для сети Кохонена K_r – от 10 до 100.

Раскраска карты Кохонена. Так как карты Кохонена сочетают в себе два основных свойства – векторное квантование и проецирование, то их используют для поиска и анализа закономерностей в исходных данных. При этом, после того как нейроны размещены на карте, полученная карта может быть отображена и раскрашена.

Если имеется множество из сотен или тысяч объектов, каждый из которых описывается несколькими параметрами, то проанали-

зировать это множество на наличие закономерностей и аномалий весьма сложно. Самоорганизующиеся карты позволяют представить множество объектов, заданных в многомерном пространстве признаков, в виде двумерных карт. Близко расположенным в многомерном пространстве объектам соответствуют близко расположенные нейроны на плоской карте. Обученная сеть Кохонена позволяет получить на выходе достаточно наглядные двумерные карты, которые несложно проанализировать визуально.

После обучения сети будет получен набор узлов, который можно отобразить в виде двумерного изображения. При этом каждой ячейке карты можно поставить в соответствие участок на сетке, координаты которого определяются координатами соответствующего нейрона в решетке. Для визуализации определяется цвет ячеек карты. Для этого используются значения выбранной компоненты w_k весового вектора W . Самый простой вариант раскраски – использование градаций серого. В этом случае ячейки, соответствующие узлам карты, с минимальными значениями компоненты w_k , будут изображены черным цветом, а ячейки с максимальными значениями w_k – белым. Можно использовать любую градиентную палитру для раскраски. Полученные раскраски в совокупности образуют атлас, отображающий расположение компонент, связи между ними, а также относительное расположение различных значений компонент.

Рассмотрим применение сети Кохонена на небольшой практической задаче из области информационной безопасности. Для анализа сетевой активности в корпоративной сети используются специальные программно-аппаратные устройства – анализаторы трафика. Эти устройства позволяют выявить вредоносное и несанкционированное программное обеспечение, например троянские программы, клиенты пиринговых сетей и др.

Как правило, анализ фиксируемых устройством данных производится в «ручном режиме», с применением простейших средств обработки протоколов. Обработать таким способом большие массивы исходных данных не представляется возможным. Поэтому целесообразно использовать сеть Кохонена для анализа исходных данных и визуального отображения полученных результатов.

Входной массив данных формируется анализатором трафика пользователей корпоративной локальной вычислительной сети (ЛВС). Для упрощения задачи будем считать, что фиксируется

число пакетов по различным протоколам в единицу времени, в частности: по порту 25/TCP (отправка почты) – параметр “SMTP”, по порту 110/TCP (получение почты) – параметр “POP3”, по порту 21/TCP (FTP-протокол) – параметр “FTP”, по порту 80/TCP (http-протокол) – параметр “HTTP”.

В примере все пользовательские компьютеры в ЛВС работают с Интернетом через прокси-сервер. Построим карту Кохонена таким образом, чтобы число нейронов соответствовало числу компьютеров. Обучение сети производится на исходных данных, после чего можно приступить к анализу. Для анализа используем раскраску карт с помощью градаций серого. Шкала раскраски приводится для каждой карты, раскрашенной по одному параметру. Каждой градации серого соответствует диапазон значений, выраженный в количестве пакетов, зафиксированных по данному параметру за единицу времени (за час, за рабочий день и т.п.).

На рис. 8.10 представлена раскраска обученной карты по параметру POP3 (получение почты). На ней видно, что все пользователи активно работают с почтой, но явных аномалий активности по этому параметру не прослеживается. Можно отметить, что на карте выделены области, соответствующие уровню активности пользователей в части получения ими почтовых сообщений. Распределение имеет равномерный характер.

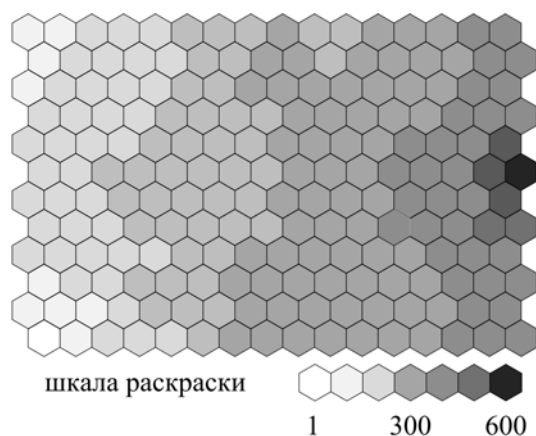


Рис. 8.10. Раскраска карты Кохонена по параметру POP3

На карте, раскрашенной по параметру SMTP (отправка почты – рис. 8.11), хорошо видно, что практически все пользователи работают с электронной почтой, но их SMTP-трафик невелик. При этом выявлена группа из нескольких компьютеров, с которых отправлено очень большое количество SMTP-пакетов.

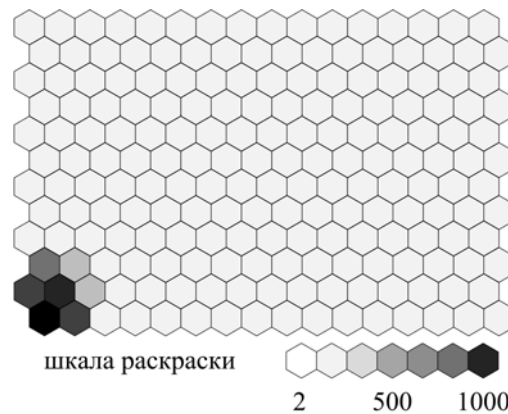


Рис. 8.11. Раскраска карты Кохонена по параметру SMTP

Анализ карты по параметру HTTP (здесь не представлен) также выявляет аномально высокую активность http-запросов, посылаемых с этих же компьютеров. Выявленные аномальные закономерности являются поводом для дальнейшего расследования и анализа. Это могут быть компьютеры, зараженные троянскими программами, осуществляющими автоматизированную рассылку рекламных сообщений.

Таким образом, как показывают рассмотренные примеры, карты Кохонена являются удобным инструментом для анализа данных с целью поиска аномалий и закономерностей – следов деятельности злоумышленников, в борьбе с вредоносными программами. Карты могут выступать аргументом в служебном расследовании, так как записи в протоколе не отличаются наглядностью, а раскрашенная карта Кохонена очень проста и наглядна.

Следует отметить, что описанные технологии в основном применяются в экономике и медицине при прогнозировании развития рынка, курсов валют, надежности компаний, при постановке диагнозов и т.п.

Вычислительный принцип, основанный на распределенной обработке информации независимыми простейшими вычислителями, который нейронные сети позаимствовали у своего биологического прототипа, является чрезвычайно эффективным. Объединение простых вычислителей лежит в основе теории нейронных сетей, что делает их способными создавать мощные вычислительные устройства с высокой степенью надежности.

Нейронные сети успешно применяются для анализа и прогнозирования на финансовом рынке, для построения систем медицинской диагностики, для обнаружения сложных зависимостей в отсутствие априорных знаний об исследуемой системе или процессе. Их можно использовать в тех областях, где невозможно построить четкий алгоритм решения задачи, где оценивание обычно производилось при помощи статистических методов анализа.

В настоящее время нейронные сети широко применяются в робототехнике и в системах управления.

В заключение можно сказать, что история исследований в области нейронных сетей довольно коротка и они еще не успели проявить себя в полную силу. Нет сомнений в том, что проникновение нейросетевых технологий в другие области является только вопросом времени.

Библиографический список

1. Горбань, А.Н. Нейронные сети на персональном компьютере / А.Н. Горбань, Д.А. Россиев. Новосибирск, Наука, 1996.
2. Ежов, А.А. Нейрокомпьютинг и его применение в экономике и бизнесе / А.А. Ежов, С.А. Шумский. М., 1998.
3. Заенцев, И.В. Нейронные сети. Основные модели / И.В. Заенцев. Воронеж: Изд-во Воронежского гос. ун-та, 1999.
4. Калан, Р. Основные концепции нейронных сетей / Р. Калан. М.: Изд. дом «Вильямс», 2003.
5. Комарцова, Л.Г. Нейрокомпьютеры / Л.Г. Комарцова, А.В. Максимов. М.: Изд-во МГТУ им. Н.Э.Баумана, 2002.
6. Короткий, С. Нейронные сети / С. Короткий. <http://www.orc.ru/~stasson/neurox.html>.
7. Круглов, В.В. Искусственные нейронные сети. Теория и практика / В.В. Круглов, В.В. Борисов. М.: Горячая линия – Телеком, 2002.
8. Мартиросов, С.М. Бионасосы – роботы клетки / С.М. Мартиросов. М.: Радио и связь, 1981.
9. Медведев, В.С. Нейронные сети. MATLAB 6. / В.С. Медведев, В.Г. Потемкин. М.: Диалог-МИФИ, 2002.
10. Осовский, С. Нейронные сети для обработки информации / С. Осовский. М.: Финансы и статистика, 2002.
11. Уоссермен, Ф. Нейрокомпьютерная техника / Ф. Уоссермен. М.: Мир, 1992.
12. Хайкин, С. Нейронные сети. Полный курс / С. Хайкин. М.: Изд. дом «Вильямс», 2006.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	3
1. КОМПЬЮТЕРЫ И МОЗГ	7
1.1. Биологический нейрон	7
1.2. Особенности взаимодействия нейронов	20
1.3. Нейрокомпьютерная обработка информации	21
2. ИСКУССТВЕННЫЙ НЕЙРОН	25
2.1. Функции активации	27
2.2. Схема абстрактного нейрокомпьютера	30
3. КЛАССИФИКАЦИЯ НЕЙРОСЕТЕВЫХ МОДЕЛЕЙ	33
3.1. Классификация по типам связей между нейронами	33
3.2. Классификация по методам обучения	37
3.3. Классификация по решаемым задачам	41
3.4. Классификация по способу решения задачи	42
4. ПЕРСЕПТРОНЫ	43
4.1. Формальный нейрон Маккаллока–Питтса	43
4.2. Персептрон Розенблатта	46
4.3. Проблема обучаемости персептронов	49
5. ОБУЧЕНИЕ МЕТОДОМ ОБРАТНОГО РАСПРОСТРАНЕНИЯ ОШИБКИ	57
5.1. Процедура обучения	57
5.2. Особенности обучения	63
5.3. Пример обучения ИНС методом обратного распространения ошибки	74
6. СЕТИ АДАПТИВНОЙ РЕЗОНАНСНОЙ ТЕОРИИ	80
6.1. Проблема стабильности-пластичности	80
6.2. Архитектура АРТ	81
6.3. Функционирование сети АРТ	85
6.4. Пример обучения	89
6.5. Свойства сетей АРТ	92
7. АССОЦИАТИВНАЯ ПАМЯТЬ	93
7.1. Рекуррентные сети	94
7.2. Сети автоассоциативной памяти	95
7.3. Двухнаправленная ассоциативная память	103
8. САМООРГАНИЗУЮЩИЕСЯ СЕТИ	112
8.1. Самообучение на основе конкуренции	112
8.2. Самоорганизующиеся карты Кохонена	116
8.3. Обучение	119
8.4. Решение проблемы мертвых нейронов	122
8.5. Применение сетей Кохонена	123
<i>Библиографический список</i>	<i>130</i>

Толмачев Сергей Геннадьевич

Системы искусственного интеллекта. Нейросетевые модели

Редактор *Г.М. Звягина*
Корректор *Л.А. Петрова*

Подписано в печать 14.09.2011 . Формат бумаги 60х84/16. Бумага документная.

Печать трафаретная. Усл. печ. л. 7,675. Тираж 200 экз. Заказ № 187.

Балтийский государственный технический университет

Типография БГТУ

190005, С.-Петербург, 1-я Красноармейская ул., д.1