

В.А. ГОЛОВКО

**НЕЙРОИНТЕЛЛЕКТ:
ТЕОРИЯ И
ПРИМЕНЕНИЕ**

КНИГА 1

**Организация и обучение нейронных
сетей с прямыми и обратными связями**

Брест 1999

УДК 681.324

Печатается по решению научного совета "Информационные технологии и системы" Института технической кибернетики Национальной Академии наук Республики Беларусь.

Рецензенты:

В.В. Голенков, доктор технических наук, профессор

Р.Х. Салыхов, доктор технических наук, профессор

В.Н. Ярмолик, доктор технических наук, профессор

Головко В.А. Нейроинтеллект: теория и применение. Книга 1: Организация и обучение нейронных сетей с прямым и обратными связями. Брест Изд. БПИ, 1999 - 264 с.

Книга посвящена вопросам проектирования и применения систем искусственного интеллекта, которые базируются на нейросетевом подходе. В основе такого подхода лежит нейронная организация искусственных систем, что имеет биологические предпосылки. Изложены математические и алгоритмические аспекты функционирования нейронных сетей с прямыми и обратными связями. Отражены биологические основы нейросетевых систем. Большое внимание уделено применению и проектированию нейронных сетей для решения различного рода задач.

Предназначена для научных работников, специалистов в области вычислительной техники и искусственного интеллекта, а также аспирантов и студентов соответствующих специальностей.

ISBN 985-6584-01-9

© Головко В.А., 1999

ISBN 985-6584-04-3

© Брестский политехнический институт, 1999

ПРЕДИСЛОВИЕ

Слово *интеллект* произошло от латинского *intellectus*, что означает познание, понимание, рассудок и характеризует способность к мышлению [1]. Еще на заре развития вычислительной техники стояла проблема создания устройств, способных к различного рода интеллектуальной деятельности. Это связано с тем, что многие задачи не могут быть решены точными алгоритмическими методами. В результате возник термин искусственный интеллект, который характеризует способность к мышлению искусственных систем. Другим аспектом этой проблемы является то, чтобы искусственная система не просто функционировала по детерминированным алгоритмам, а обладала бы способностью к генерации знаний и алгоритмов решения задач. Развитие таких систем происходит за счет *самоорганизации*: в результате которой осуществляется адаптация к различным условиям функционирования.

Существует два направления в теории искусственного интеллекта. Первое направление является традиционным. Оно использует методы логических рассуждений и символической обработки информации. Второй подход связан с построением сетей, состоящих из нейронных элементов. Он опирается на биологические основы естественного интеллекта и позволяет проектировать системы, способные к обучению и самоорганизации. Использование неirosетевых методов в теории искусственного интеллекта называется *нейроинтеллектом*.

В настоящей работе основное внимание уделено рассмотрению теории нейронных сетей и их использованию для решения различного рода задач.

Предлагаемый труд "Нейроинтеллект: теория и применение" состоит из двух книг, которые охватывают различные аспекты неirosетевых технологий.

Предисловие

Данная книга является первой из двух книг, в которой рассматривается методология организации и обучения нейронных сетей с прямыми и обратными связями.

Глава 1 посвящена различным аспектам нейронной организации головного мозга и обучения биологических систем. Рассмотрены процессы морфогенеза и самоорганизации как физических, так и биологических систем. Показано, что самоорганизация происходит в результате динамической перестройки системы с целью адаптации к внешней среде. Приведены модели самоорганизации различных систем. Рассмотрены основные этапы развития искусственных нейронных сетей и их классификация.

В главе 2 рассматриваются нейронные сети с одним обрабатывающим слоем и прямыми связями (*feed forward networks*). Такие сети сыграли большую роль на пути развития нейроинтеллекта. В главе приводятся основные определения и различные алгоритмы обучения однослойных нейронных сетей. Рассматривается вывод выражения для адаптивного шага обучения, использование которого позволяет уменьшить временную сложность процесса обучения. Показано применение однослойных нейронных сетей в качестве ассоциативной памяти и для прогнозирования временных рядов.

В главе 3 отражены различные аспекты построения и обучения многослойных нейронных сетей с прямыми связями (*multi layer feed forward networks*). Такие сети позволяют осуществить любое отображение входных сигналов в выходные. Рассмотрены алгоритм обратного распространения ошибки (*backpropagation algorithm*) и его модификации, которые ориентированы на ускорение процесса обучения. Анализируется применение многослойных нейронных сетей для решения задач распознавания, прогнозирования и кодирования информации.

В главе 4 рассматриваются функционирование и алгоритмы обучения

рекуррентных и рециркуляционных нейронных сетей. Такие сети характеризуются как прямым, так и обратным распространением информации. Рекуррентные нейронные сети применяются в основном для прогнозирования различных временных процессов, а рециркуляционные - для сжатия и восстановления информации. Рассмотрены различные алгоритмы обучения таких нейронных сетей, как известные, так и разработанные автором.

Глава 5 посвящена теоретическому анализу и различным аспектам применения релаксационных нейронных сетей. Такие сети характеризуются обратными связями и циркуляцией информации до тех пор, пока не установится стабильное состояние. К релаксационным сетям относится сеть Хопфилда, Хэмминга и двунаправленная ассоциативная память. В главе приводится архитектура, функционирование и обучение таких нейронных сетей. Для сети Хопфилда рассмотрен механизм получения функции Ляпунова и анализ устойчивости.

Автор стремился изложить материал в доступной для широкого круга специалистов форме и восполнить пробелы в отечественной литературе по данной тематике. Поэтому в монографии подробно рассмотрены различные нейронные сети и приводится много примеров.

Данная работа выполнена в соответствии с INTAS проектами 97-0606 "Development of an intelligent sensing instrumentation structure" и 97-2028 "Intelligent neural system for autonomous control of a mobile robot". Автор благодарит INTAS за поддержку исследований в области нейроинтеллекта.

В заключение автор считает своим долгом выразить глубокую признательность доктору технических наук, профессору В.А. Мищенко за постоянное внимание и поддержку в работе, а так же рецензентам: заведующему кафедрой интеллектуальных информационных технологий БГУИР профессору В.В. Голенкову; заведующему кафедрой ЭВМ БГУИР

Предисловие

профессору Р.Х. Садыхову и заведующему кафедрой программного обеспечения и информационных технологий БГУИР профессору В.Н. Ярмолику. Автор выражает искреннюю благодарность профессорам К. Шиллингу и Н. Роту (Германия) за плодотворные дискуссии и предоставление экспериментальной базы для проведения исследований. Автор выражает также свою признательность ректору Брестского политехнического института В.Г. Федорову и заведующему кафедрой ЭВМ Брестского политехнического института Н.В. Кудинову за внимание и поддержку, а также своим коллегам по группе нейронных сетей Брестского политехнического института за помощь в проведении экспериментальных исследований.

ГЛАВА 1. ОСНОВЫ ЕСТЕСТВЕННОГО ИНТЕЛЛЕКТА

Интеллект характеризует способность в процессе мышления к генерированию и выбору способа действий, адекватно отражающих решаемую проблему. Естественный интеллект возник и развивается в процессе биологической эволюции с целью адаптации к внешней среде. Он присущ в той или иной мере биологическим формам жизни. *Искусственный интеллект* характеризует способность к мышлению искусственных систем. Он опирается на биологические основы естественного интеллекта и пытается в той или иной мере моделировать мыслительные процессы живых существ. В результате развития систем искусственного интеллекта возник термин “искусственная жизнь” (Artificial Life), который отражает различные аспекты поведения и взаимодействия искусственных систем. В главе рассматривается нейронная организация и механизмы обучения естественных систем. Обсуждаются процессы самоорганизации и морфогенеза. Приводятся основные этапы развития искусственных нейронных систем и их классификация.

1.1. На стыке наук

Человеческий мозг является самой сложной в обозримом мире структурой, которая является результатом многих миллионов лет эволюции. В отличие от компьютера он создавался без жестко определенной схемы, а в результате естественного отбора – главной движущей силы эволюции. Способность человеческого мозга к обучению, запоминанию сложной информации и логическому мышлению с давних пор привлекают к нему философов, нейробиологов, психологов и кибернетиков. В прошлом многие,

например Декарт, представляли психику как нечто нематериальное. Сейчас большинство исследователей сходятся к мысли, что психику можно объяснить материалистически, как результат взаимодействия клеток мозга. До недавнего времени представители различных наук работали изолированно. Так, **нейробиология** – наука о мозге, изучала функционирование мозга на молекулярном уровне. Здесь она использует принцип “от молекулы к разуму”. **Психология** долгое время находилась под влиянием бихевиоризма и исследовала мозг как “черный ящик”, основываясь лишь на наблюдении поведения без учета морфологии мозга. В противовес этому в середине 50-х годов возникла **когнитивная психология** (наука о разуме), которая дала возможность анализировать психические процессы. **Кибернетика**, используя теорию искусственных нейронных сетей и опираясь на достижения нейробиологии и когнитивной психологии, исследует искусственные системы, которые позволяют воспроизводить мыслительные процессы биологических существ.

В настоящее время происходит интеграция этих наук. Так когнитивная нейробиология пытается выяснить, как психические события скоррелированы с электрическими сигналами в мозге, и описать высшие психические функции в терминах точных наук.

Теория искусственных нейронных сетей, моделируя в той или иной степени мыслительные процессы, позволяет глубже понять функционирование мозга. Задача объединения этих наук состоит в том, чтобы описывать высшие психические функции как скоординированную активацию нейронов в коре головного мозга. Это создает потенциальные предпосылки для создания **нейрокомпьютеров**, которые являются новым этапом в эволюции вычислительной техники.

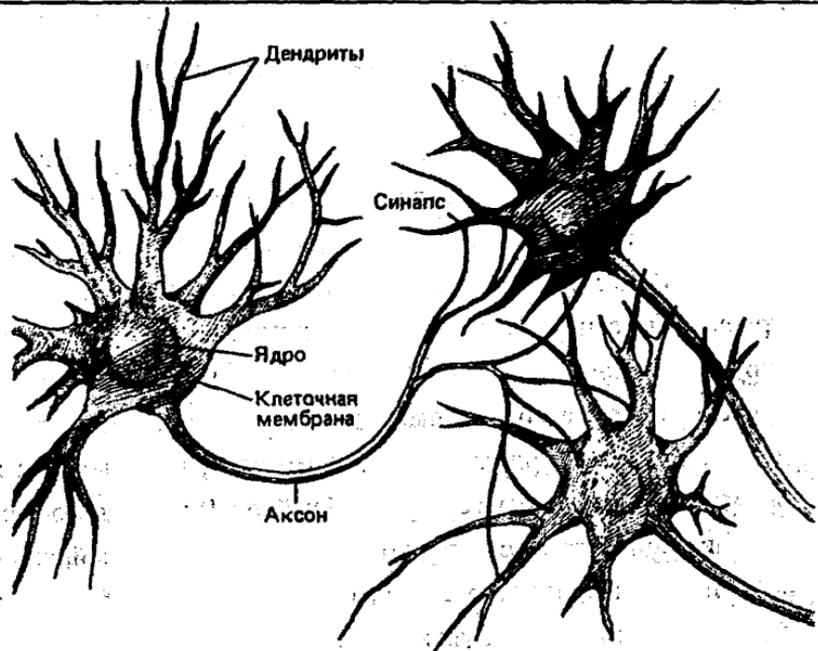


Рис 1.1. Биологический нейрон.

1.2. Биологический нейрон

Головной мозг человека весит от 1.3 до 1.8 кг и содержит триллион клеток, из которых 100 млрд. представлены соединенными в сети нейронами [2]. По порядку величины это сравнимо с числом звезд в Млечном Пути. Нейроны, соединенные разнообразными связями в сеть определяют интеллект, творческие способности и память человека. Количество нейронов в головном мозге человека больше чем у всех остальных известных форм жизни. *Нейрон* представляет собой особый вид клеток, которые обладают электрической активностью [3]. Он получает информацию (рис. 1.1) при помощи сильно разветвленных отростков, называемых *дендритами* и

передает информацию вдоль тонкого волокна, которое называется *аксоном*. Аксон имеет множество ответвлений, на конце каждого из которых находится область, называемая *синапсом*. Посредством синапсов осуществляется связь между различными нейронами. Каждый нейрон может иметь тысячи связей с соседними нейронами. Информация по аксонам передается в виде коротких электрических импульсов [2], амплитуда которых составляет около 100 мВ, а длительность 1 мс. На участках контакта между нейронами (синапсы) электрические импульсы превращаются в химические сигналы, которые стимулируют проникновение в клетку нейрона положительных зарядов. Когда достигается некоторое критическое значение потенциала, называемое пороговым, возникает электрический импульс, который как волна распространяется по аксону на следующий нейрон. Вклад одного синапса в установление соответствующего потенциала на выходе нейрона очень маленькая. Для возникновения электрического импульса, необходимо чтобы нейрон непрерывно интегрировал множество синаптических входов.

При этом такая интеграция является нелинейным преобразованием и не соответствует простой операции линейного суммирования. Использование технологии окраски нейронов солями серебра [2], позволяет выявить большое разнообразие типов нейронов в коре головного мозга. Так существуют пирамидальные нейроны, нейроны Таламуса, нейроны Пуркинье и т.д., всего около 50 типов. Из этого следует, что не все компоненты, из которых построен мозг, взаимозаменяемы.

Скорость распространения нервного импульса в аксоне составляет приблизительно 100 м/с, что в миллион раз меньше скорости распространения электрического сигнала по медной проволоке [2]. Однако параллельная обработка нейронами информации, которая одновременно распространяется по множеству связей, компенсирует этот недостаток.

Таким образом, в процессе психической деятельности, в коре головного

мозга распространяются нервные импульсы, которые активизируют соответствующие области нейронов. Совокупность нейронов и связей между ними образуют *нейронную сеть*, от функционирования которой зависят эмоциональные реакции, сознательная деятельность и память человека.

1.3. Нейронная организация мозга

Структурно головной мозг состоит из двух симметричных полушарий, соединенных между собой пучком из примерно полмиллиарда аксонов [2], называемым *мозолистым телом* (рис. 1.2). Известно, что если мозолистое тело перерезать, то одна сторона мозга не будет осознавать, что видит другая. Мозг содержит белое и серое вещество. Серое вещество представляет собой совокупность дендритов, аксонов и нейронов. Белое вещество состоит главным образом из нервных волокон, которые соединяют различные области мозга друг с другом.

Анатомически мозг разделен на ряд зон, которые выполняют различные функции. Это *префронтальная кора, гиппокамп, гипоталамус, зрительная кора, лимбическая система* и т.д. Каждая область представляет собой совокупность нейронов различных типов, соединенных между собой и другими частями мозга разнообразными связями. Как показывают исследования [2] молекулярный состав и функции отдельных нейронов приблизительно идентичны для биологических существ. Вероятно, именно объемом, организацией связей и взаимодействием нейронов *Homo sapiens* отличается от других форм жизни.

В настоящее время активно изучаются различные зоны мозга и их связь с психикой индивидуума. Так *лимбическая система* участвует в эмоциональном поведении и долговременной памяти, которая хранит длительный период цифры, факты, правила и события [2]. Для того чтобы

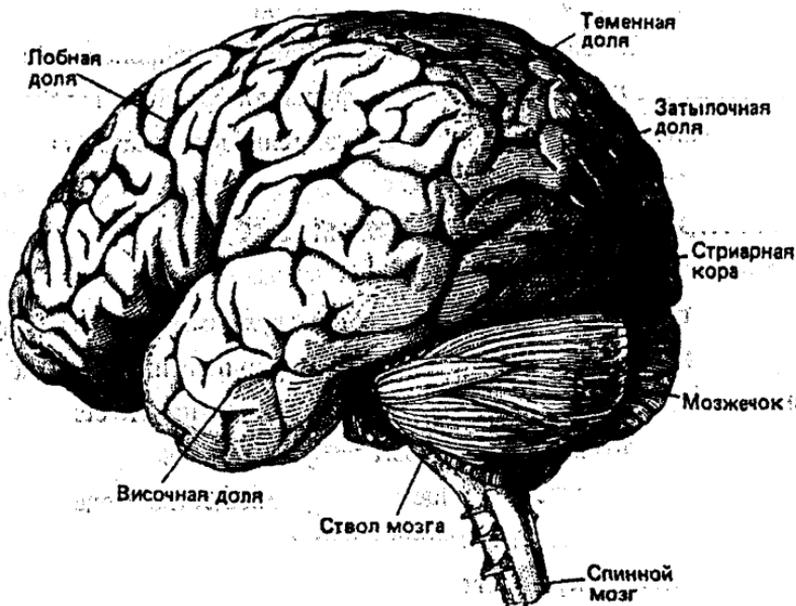


Рис. 1.2. Головной мозг.

записи в этой памяти не забывались, необходимо через определенное время активизировать соответствующую нейронную структуру этой памяти. При повреждении нейронов долговременной памяти человек утрачивает связь со своим прошлым. [4]. *Гиппокамп* выполняет функции кратковременной памяти, которая хранит информацию без реактивации соответствующей нейронной сети от нескольких минут до нескольких часов [5]. Время реактивации кратковременной памяти, чтобы информация не исчезла, значительно меньше чем долговременной. Нейронная структура гиппокампа перерабатывает и хранит новую информацию, полученную в результате обучения и при соответствующей реактивации, в течение недель или месяцев, а затем передает ее в другую зону мозга для длительного хранения. *Префронтальная кора* участвует в образовании оперативной памяти,

которая необходима для извлечения фактов, событий и правил из долговременной памяти и манипулирования ими, или промежуточными результатами в соответствии с обстоятельствами. В качестве примера оперативной памяти можно привести планирование шахматного хода или операцию переноса при вычислении в уме, когда требуется в процессе выполнения операции запоминание промежуточных результатов. Соответствие перечисленных выше зон головного мозга различным типам памяти было подтверждено экспериментально [5]. Так, у больных с повреждением нейронов височной части - гиппокампа, нарушено запоминание новой информации. При этом они хорошо помнят прошлые события, но не воспринимают текущую информацию. Повреждение префронтальной коры является причиной многих психических расстройств, включая шизофрению [4].

Из разделения мозга на различные зоны вытекает концепция функциональной организации нейронных структур головного мозга [5]. Согласно ей, различная информация обрабатывается и хранится в различных нейронных сетях мозга. Так, некоторые исследователи считают, что префронтальная кора представляет собой совокупность участков памяти, каждый из которых специализируется на информации определенного рода. Например, цвет, размеры объекта, семантические и математические знания. При этом разные характеристики одного объекта восприятия обрабатываются параллельно в различных нейронных схемах. Это подтвердилось при исследовании зрительной системы обезьяны [7]. Так ее зрительная кора состоит из различных зон. Первая зона содержит с топографической точностью карту поверхности сетчатки, от которой информация параллельно поступает к остальным нейронным зонам. При этом информация, соответствующая форме объекта поступает к одной нейронной сети, цвету объекта - к другой сети и т.д. Как показали эксперименты [7], повреждение

первой зоны приводит к полному отказу зрительного тракта (слепоте), а поражение нейронов одной из специализированных зон – к недоступности и неосознаваемости соответствующего ей атрибута. Так, например, повреждение зоны цвета ведет к ахроматописии, при которой все окружающее видится лишь в оттенках серого цвета.

Отсюда следует, что один и тот же объект представляется в мозге в виде совокупности различных атрибутов: зрительный образ, набор слов, звуков, запах, цвет и т.д. Эти представления об одном и том же объекте взаимодействуют между собой и охватывают множество нейронных структур, относящихся к разным функциональным областям мозга. Осознание и понимание увиденного происходит одновременно благодаря синхронизации активности соответствующих нейронных структур.

Например, когда человек видит какой-то объект, его нейронная структура реагирует на цвет, форму, запах и функциональное назначение объекта. При этом в результате нервной активности могут активизироваться в мозге так же зоны нейронных элементов, которые учитывают прошлый опыт человека о соответствующем объекте, то есть происходит процесс воспоминания. Таким образом, активация одной из нейронных структур может генерировать работу другой и т.д. Результатом этой деятельности является совокупное представление об объекте. Аналогичные ассоциации возникают при воздействии слова, характеризующего тот или иной образ.

Каким образом взаимодействуют различные функциональные зоны мозга для получения единой интеграционной картины объекта? Простейшим решением здесь является передача информации от различных зон некоей высшей зоне, интегрирующей поступающую информацию. Однако анатомические данные не подтверждают этого предположения. Исследования показали [2], что между различными нейронными структурами мозга существуют обратные связи. Эти связи, как предполагается, синхронизируют

Глава 1: Основы естественного интеллекта

активность различных функциональных зон, благодаря чему создаются единая интеграционная картина об объекте.

Как уже отмечалось выше, скорость распространения сигналов в нервных волокнах намного меньше, чем скорость распространения сигналов в электрических схемах. Однако параллельная обработка нейронами образной информации и организация взаимодействия между нейронами, как было показано на примере зрительного тракта, позволяет нейтрализовать этот недостаток. Поэтому на операциях распознавания образов и аналогичных им, человеческий мозг намного эффективнее современных компьютеров. Последовательные алгоритмы и последовательная обработка информации в компьютерах является конструкцией нашего разума, а значит, в нем также должна быть заложена последовательная обработка информации. Так при оперировании символьной информацией, например, логический вывод или вычисление нескольких арифметических операций, конструкция нашего мозга использует последовательный алгоритм работы с привлечением оперативной памяти для запоминания промежуточной информации. Поэтому на таких операциях, требующих детерминированной последовательности действий, производительность вычислений человеческим мозгом ниже компьютера. Другой аспект этой проблемы состоит в том, что мозг сам, в отличие от компьютера, может конструировать алгоритмы решения задачи. Он это производит путем поиска, методом проб и ошибок, используя также последовательный режим работы.

Исходя из приведенных выше рассуждений, можно сделать следующие выводы [8]:

Мозг разделен на зоны, которые состоят из различных функциональных структур нейронных сетей. Различная информация хранится и обрабатывается в разных нейронных структурах головного мозга.

Между нейронными структурами мозга существуют как прямые, так

и обратные связи. Обратные связи в частности синхронизируют активность различных функциональных нейронных зон, благодаря чему создается единая интеграционная картина об объекте.

Осознание и понимание увиденного мозгом происходит одновременно, благодаря синхронизации активности соответствующих нейронных структур, которые обрабатывают различные атрибуты информации (цвет, форма, запах и т.п.)

Головному мозгу характерна как последовательная, так и параллельная обработка информации.

Образная информация обрабатывается параллельно, а операции с символической информацией производятся в общем случае последовательно.

Система памяти головного мозга состоит из кратковременной, оперативной и долговременной.

Существует последовательное и параллельное соединение нейронных структур мозга с точки зрения надежности. При нарушении одной из параллельно соединенных нейронных структур происходит частичная потеря функций соответствующей нейронной системы (не различение цветов и т.д.). При дефекте одной из последовательно соединенных нейронных структур происходит отказ, ведущий к полной потере функций соответствующей нейронной системы.

1.4. Морфогенез мозга

Развитие мозга - это непрерывный процесс, который происходит в результате обучения посредством взаимодействия с внешней средой и с учетом внутренних факторов организма. Оно начинается еще в эмбриональный период и происходит в течение всей жизни. Как показывают исследования [7], именно в эмбриональный период закладываются основы

разума путем образования нейронных структур и связей между ними. Мозг новорожденного по массе составляет около четверти массы взрослого человека. В процессе развития индивидуума происходит увеличение размеров нейронов, числа их связей, изменяется структурная организация нейронных элементов и взаимодействие между ними. Некоторые исследователи полагают [7], что формирование мозга в эмбриональный период происходит согласно заданной схеме аналогично сборке компьютера. Однако, как показывают исследования, это не так. Установление и образование нейронных структур мозга зависит как от генетической программы развития нервной системы, так и от внешних воздействий, стимулирующих нервную активность мозга [7]. Под управлением генетической программы происходит рост аксонов в определенных направлениях для установления синаптических связей между нейронами. Однако не существует точной спецификации каждого межнейронного соединения. Для этого потребовалось бы огромное количество генов. Как только растущие кончики аксонов достигают соответствующей им области, на выбор конкретной межнейронной связи начинают влиять события внешнего мира, стимулирующие определенную нервную активность. От воздействия этой активности и зависит установление соответствующих межнейронных связей.

Так искусственное блокирование зрительной системы у новорожденных котят, приводит к ненормальному развитию зрительной коры [7].

Таким образом, формирование нейронов и связей между ними начинается в эмбриональный период. Развитие индивида происходит посредством непрерывного процесса организации и модификации связей между нейронными элементами, а также роста размеров и количества нейронов. По мере взросления человека общее число нейронов в мозге снижается, уменьшаются размеры крупных нейронов и атрофируются связи между нейронами [9]. Соответственно снижается вес мозга. Однако влияние

этих изменений на интеллект варьируется у различных людей. Для некоторых индивидуумов структурные изменения становятся заметными только после 70 лет. Так некоторые исследователи [9] наблюдали у людей от 40 до 70 лет рост дендритов в коре головного мозга, с последующей регрессией их после 80 лет. Они предположили, что рост дендритов отражает попытку жизнеспособных нейронов нейтрализовать утрату с возрастом соседних нейронов. Это показывает, что мозг способен к динамической перестройке нейронных сетей даже в поздние годы жизни.

В процессе развития мозга непрерывно происходит динамическая перестройка нейронных сетей. При этом формируется сознание, и то, что З. Фрейд называет бессознательным (инстинкты, влечения и т.д.) [10]. *Сознание* - это не состояние, а процесс, как отмечал американский психолог Уильям Джеймс [11]. К этому можно добавить, что сознание - это процесс непрерывного развития и структурной организации нейронов мозга, который происходит под воздействием внутренних и внешних факторов, во времени и в пространстве. З. Фрейд полагал, что душевные бессознательные процессы находятся вне времени [10]. Однако, это не так. Даже если содержание бессознательных процессов является статическим, то форма их проявления меняется с развитием цивилизации. Как уже отмечалось, формирование мозга происходит под воздействием генетической программы и факторов внешней среды. Так как каждый индивидуум имеет неповторимую генетическую информацию и развивается в различных условиях (испытывает воздействие различных сочетаний раздражителей), то архитектура мозга формируется у разных людей по-своему. Это составляет его *индивидуальность*. Человек и его сознание живет и развивается во времени и в пространстве. Поэтому несмотря даже на одинаковую генетическую информацию, нельзя полностью воспроизвести архитектуру мозга и соответственно интеллект индивидуума. Это делает несерьезными разговоры о *клонировании* с целью

воспроизводства интеллекта конкретного индивидуума. Как писал Гегель, в одну и ту же реку нельзя войти дважды.

Итак, в процессе морфогенеза непрерывно осуществляется динамическая перестройка нейронных сетей головного мозга. У разных индивидуумов это происходит по-разному в зависимости от генетической программы и внешних воздействий, стимулирующих нервную активность мозга. По мере взросления человека общее число нейронов в мозге снижается, однако путем перестройки соответствующих нейронных структур, мозг стремится компенсировать эту утрату.

1.5. Механизмы обучения

Обучение - это процесс непрерывного развития и формирования сознания посредством взаимодействия с внешней средой и с учетом индивидуальности организма. В результате обучения происходит динамическая перестройка нейронных сетей головного мозга. При этом, как отмечалось, увеличивается число связей между нейронами, совершенствуются нейроны и взаимодействие между ними. Способность синапсов и нейронных сетей динамически изменяться в результате воздействий называется *пластичностью*. Механизм пластичности лежит в основе обучения и создает в коре головного мозга соответствующие структуры нейронных сетей, которые определяют интеллект, память и эмоции индивидуума: *Синаптическая пластичность* происходит в результате изменения эффективности и количества связей между нейронами. В 1949 году канадский психолог D. Hebb сформировал принцип обучения [12], который был экспериментально подтвержден для некоторых типов синапсов. Согласно этому принципу, для усиления связи между пресинаптическим и постсинаптическим нейронами, необходимо совпадение их активности во

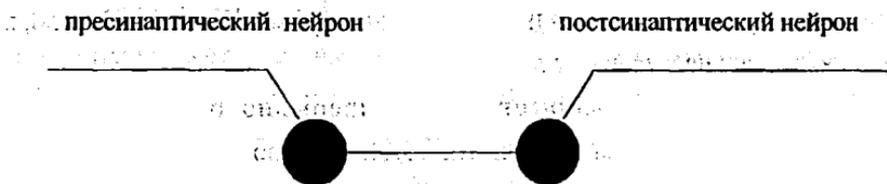


Рис. 1.3. Правило обучения Хебба.

времени (рис. 1.3). Нейроны, совпадение активности которых необходимо для усиления синаптической связи, изображены на рисунке черным цветом. Усиление связи приводит к повышению эффективности синаптической передачи между двумя нейронами. Данный механизм экспериментально подтвердился для некоторых синапсов гиппокампа [5], которые участвуют в образовании кратковременной памяти. Считается, что правило Хебба играет также большую роль в *явном обучении*, которое требует участия сознательных процессов.

В 1963 г. французские исследователи Л. Тауц и Э. Кэндел, изучая нервную систему морской улитки, сформулировали другой принцип ассоциативного обучения [5].

Они обнаружили, что для усиления синаптической связи между двумя нейронами не обязательна активность постсинаптического нейрона, если на пресинаптическую клетку действует некий третий нейрон. Этот третий нейрон они назвали модулирующим. Если одновременно с пресинаптическим нейроном активен модулирующий нейрон, то происходит усиление связи между пресинаптическим и постсинаптическим нейроном (рис. 1.4). Данный механизм обучения участвует в образовании условных рефлексов и *явном обучении* [4], которое не требует участия сознания и сводится к механическому повторению определенных действий.

Кроме механизма усиления связей, существует также механизм их

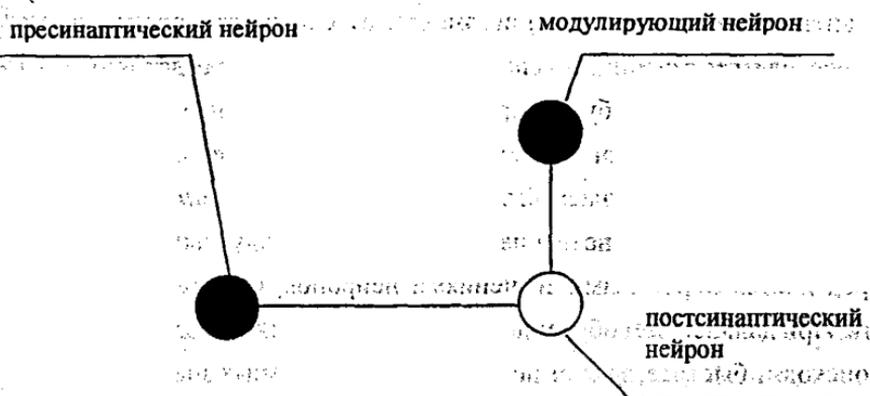


Рис. 1.4. Принцип ассоциативного обучения с модулирующим нейроном.

ослабления [5]. В этом случае сила связи между двумя нейронами уменьшается, если активность пресинаптического нейрона не сопровождается активностью постсинаптического нейрона. Исследования показали [4], что имеющиеся в префронтальной коре пирамидальные нейроны имеют симметричные и асимметричные синапсы. Симметричные синапсы соответствуют тормозным связям, а асимметричные синапсы - возбуждающим связям. Результатом действия тормозных связей может являться депрессия и стрессовые состояния [4].

Итак, в процессе обучения происходит усиление синаптических связей между соответствующими нейронами головного мозга. В результате возникает кратковременное запоминание информации, которая хранится без реактивации соответствующей нейронной сети от нескольких минут до нескольких часов. При долговременном запоминании информации, которое длится месяцами, наблюдается активация и экспрессия генов, синтез соответствующих белков и рост новых связей [5]. При более сложных видах обучения участвуют эти два механизма. Например, осмысление нового материала требует определенного времени. В процессе этого усиливаются

синаптические связи, и в зависимости от сложности проблемы может осуществляться рост новых связей. Данный механизм будет продолжаться до тех пор, пока не будет преодолено какое-то пороговое значение соответствующей нейронной сети головного мозга. В результате происходит понимание (инсайт) решаемой задачи. *Пороговое значение* нейронной сети характеризует степень незнания материала, индивидуальность организма и определяется пороговыми значениями нейронов, составляющих искомую сеть. При дальнейшем обучении в аналогичной области, процесс понимания происходит быстрее, за счет использования начальных знаний, заложенных в соответствующих синапсах.

В зависимости от вида взаимодействия обучающегося с внешней средой можно условно выделить обучение с учителем и без него. Обучение с учителем происходит при взаимодействии ученика с конкретным индивидуумом (учителем), с которым он находится в состоянии обратной связи.

В этом случае имеется конкретный желаемый выход и алгоритм его получения. В процессе взаимодействия реальная реакция ученика сравнивается с эталонной реакцией учителя. В зависимости от величины несовпадения их (целевая функция ошибки) происходит соответствующая перестройка синаптических связей с целью минимизации ошибки. При обучении без учителя нет конкретного учителя (учитель - внешняя среда) и ученик находится в состоянии обратной связи с внешней средой. Обучение здесь сводится к адаптации индивидуума к внешней среде. В обоих типах обучения используются как положительные, так и отрицательные обратные связи в соответствующих нейронных структурах головного мозга. Так, обучение с отрицательной обратной связью происходит для минимизации ошибки целевой функции. Положительная обратная связь может интенсифицировать процесс обучения при успешном взаимодействии.

индивидуума со средой.

Важной характеристикой процесса обучения является *обобщающая способность*. Она характеризует способность индивида интегрировать частные данные для определения закономерностей и пролонгации результатов. К этому относится способность после обучения на одних данных применять полученные знания для других данных или рассуждения от частного к общему.

Обобщающая способность является важной чертой нейронной организации мозга и является тем, что отличает *Homo sapiens* от остального мира.

1.6. Самоорганизация и адаптация

Адаптация характеризует процесс приспособления организма или популяции организмов к внешней среде. Она определяется как внутренней организацией организма, которая характеризует его индивидуальность, так и воздействием внешней среды. В зависимости от этого возможны различные варианты адаптационного взаимодействия (модели поведения). Например, конформистская, ритуальная, агрессивная и т.д. В дальнейшем будем называть популяцию организмов биологической системой.

Самоорганизацией называется процесс динамической перестройки организма или биологической системы с целью адаптации к внешней среде. Самоорганизация на уровне организма происходит посредством обучения, в результате которого происходит динамическая перестройка нейронных структур головного мозга. Самоорганизация на уровне биологической системы осуществляется посредством взаимодействия между элементами системы и внешней средой, в результате чего происходит динамическая перестройка связей между организмами. В настоящее время возникла

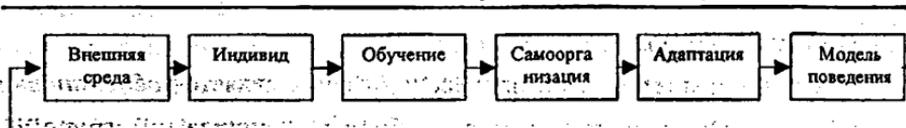


Рис. 1.5. Модель взаимодействия индивида с внешней средой.

специальная наука о самоорганизации, которая называется *синергетикой* [13]. Модель взаимодействия индивида с внешней средой в общем случае можно представить в виде следующей схемы (рис. 1.5).

Самоорганизация и адаптация здесь происходят посредством обучения в результате взаимодействия с внешней средой. В ходе этого осуществляется динамическая перестройка соответствующих нейронных структур головного мозга. Следует отметить, что на процессы самоорганизации и адаптации большое влияние оказывают связанные наложенные обществом на индивида (устройство общества, религия, мораль и т.д.). Аналогичная модель (рис. 1.5.) в общем случае характерна и для биологической системы. Однако в отличие от технической системы функционирование биологической системы, как правило, не может быть выведено, исходя из функций составляющих ее элементов. На процессы самоорганизации и адаптации биологической системы большое влияние оказывает ее прошлое (история), а молекулы ее являются результатом многих лет эволюции. Самоорганизация играет большую роль в эволюции Вселенной и биологических систем. Так лауреат Нобелевской премии И. Пригожин считает, что жизнь – это высшее проявление процессов самоорганизации [14]. Поэтому рассмотрим подробно процессы самоорганизации живой и неживой материи.

1.6.1. Самоорганизация физических систем

Самоорганизацией физических систем называется переход от

хаотической к упорядоченной структуре (от хаоса к порядку). *Состояние хаоса* соответствует состоянию равновесия физической системы. Для изолированной системы это эквивалентно максимуму *энтропии*, которая является мерой неупорядоченности или мерой вероятности пребывания системы в данном состоянии:

$$S = k \cdot \ln(P),$$

где k - постоянная Больцмана, P - вероятность.

Максимум энтропии для систем, состоящих из большого числа частиц, достигается при их равномерном распределении, что соответствует наибольшей вероятности этого состояния. Иначе говоря, для изолированной системы равновесное состояние является аттрактором (притягивающее множество). Эволюция изолированной системы всегда приводит к равновесному состоянию.

Для замкнутой системы (температура ее поддерживается постоянной) равновесие соответствует минимуму функции свободной энергии:

$$F = E - T \cdot S,$$

где E - энергия системы, T - температура по шкале Кельвина, S - энтропия.

При низких температурах это приводит к образованию кристаллов, когда каждая молекула скована взаимодействием со своими соседями.

Равновесная структура инертна и, в случае изоляции, ее состояние можно поддерживать бесконечно долго. Примером самоорганизации физических систем являются ячейки Бенара [14] и химические часы [14]. *Ячейки Бенара* представляют собой шестигранные структуры, которые возникают в подогреваемом снизу слое жидкости. Между нижним и верхним слоями жидкости образуется градиент температур.

Когда он достигает некоторого критического значения, то возникает согласованное движение множества молекул, которые образуют ячейки в

форме правильных шестиугольников. Такому распределению молекул, которое приводит к самоорганизации, соответствует очень малая вероятность и энтропия. Но, тем не менее, самоорганизация происходит. Аналогичные процессы происходят при образовании *химических часов*: В этом случае, при взаимодействии молекул двух сортов (например, красных и синих) возникает периодическая смена окраски раствора. В один момент времени он будет иметь синий цвет, затем красный и т.д.

Теоретическое обоснование процессов самоорганизации физических систем произвел бельгийский ученый русского происхождения И. Пригожин [14]. Согласно ему, процессы самоорганизации протекают в сильно неравновесных условиях, источником которых являются различного рода флуктуации. При превышении флуктуациями некоторого критического значения, они захватывают всю систему, и она начинает эволюционировать к новому состоянию. Критическое значение при этом соответствует порогу устойчивости системы и называется *точкой бифуркации*. В окрестности точки бифуркации происходят сильные флуктуации, поэтому такая область называется хаотической.

В ней невозможно предсказать, в каком направлении будет происходить дальнейшее развитие системы: перейдет ли она к той или иной форме самоорганизации или станет хаотической. Это определяется случайными факторами. При этом даже незначительные воздействия могут существенно влиять на выбор направления развития. Когда выбор направления произошел, случайность уступает место детерминированному пути развития. Таким образом, *неравновесность* может приводить к различным формам самоорганизации. При этом в отличие от равновесных структур, где молекулы ведут себя инертно, в неравновесных условиях между ними действуют дальнедействующие корреляции, как будто каждая молекула информирована о состоянии системы в целом. Как было сказано выше,

большую роль в процессах самоорганизации играет случайность. Одним из крупных оппонентов введения случайности в физику был А. Эйнштейн. Известно его изречение: «Бог не играет в кости». До недавнего времени возникновение случайности связывалось со сложностью физической системы, когда очень трудно описать ее детерминированным образом. Однако исследования показывают, что для многих нелинейных динамических систем статистический подход - единственно верное отражение реального поведения системы [15].

Таким образом, динамика открытых физических систем может происходить как по пути самоорганизации (энтропия уменьшается), так и хаотического развития (энтропия увеличивается).

1.6.2. Самоорганизация простейших биологических систем

Процессы самоорганизации биологических систем выработались в течение многих лет эволюции для адаптации к внешней среде.

Рассмотрим механизм самоорганизации у коллективных амёб [14]. При наличии достаточного пищевого ресурса амёбы растут и размножаются как одноклеточные организмы. В этот период они распределены равномерно в области обитания (равновесное состояние). Когда пищевой ресурс истощается, некоторые амёбы начинают испускать химическое вещество (циклическая АМФ), которое сигнализирует о недостатках питательной среды. Реагируя на химические сигналы, амёбы начинают мигрировать, выделяя при этом так же химическое вещество.

При достижении в среде критического значения градиента концентрации циклической АМФ, флуктуации усиливаются и захватывают всю популяцию. В результате, однородное распределение становится неустойчивым и амёбы эволюционируют к неоднородному распределению.

При этом отдельные амебы (клетки) соединяются в колонию, состоящую из нескольких тысяч клеток. Благодаря миграции образуется многоклеточный организм, способный передвигаться и осваивать новые области обитания.

Аналогичные процессы происходят при постройке гнезд термитами [14]. Процесс начинается с беспорядочного разбрасывания кусочков земли, которые пропитаны гормоном, притягивающим других термитов. В результате возрастает концентрация комочков земли, которая в свою очередь привлекает других термитов. Процесс нарастает лавинообразно и заканчивается построением термитника. Большую роль в поведении отдельных компонентов биологических систем играют не только генетические факторы, но связи и взаимодействия в популяции. Так, например, в популяции некоторых видов муравьев можно условно выделить две категории: труженики и лентяи. Если изолировать их между собой, то в каждой популяции опять произойдет расслоение на тружеников и лентяев [14]. При этом часть лентяев превращается в тружеников.

Как видно из приведенных выше примеров, простейшие биологические системы в процессе самоорганизации производят коллективные действия, источником которых является неравновесное состояние с внешней средой. Когерентное поведение таких систем характеризуется отсутствием организующего разума: отдельный компонент системы не в курсе глобальной задачи и все взаимодействия локальны. Энтропия простейших биологических систем изменяется с течением времени по колебательному закону: уменьшается при самоорганизации и увеличивается при стремлении их к равновесному состоянию.

1.6.3. Самоорганизация высших биологических систем

Рассмотрим эволюцию и самоорганизацию высших биологических

Глава 1. Основы естественного интеллекта

систем на уровне человеческого общества. Компонентом такой системы является Homo sapiens. В отличие от других организмов, человек обладает высоким уровнем интеллекта и разума, которые характеризуют способность его к познанию, осмыслению и целенаправленной сознательной деятельности. Индивидуумы здесь находятся на более высокой ступени дифференциации (более различимы): каждый из них обладает своим уровнем развития, характером, опытом и историей, т.е. тем, что составляет его индивидуальность. Эволюция высших биологических систем зависит от связей, существующих в системе (устройство общества), развития и взаимодействия ее компонентов, исторического прошлого и воздействия внешней среды. Связи системы ограничивают степень свободы ее элементов и отражают структуру общества. В зависимости от связей происходит взаимодействие и развитие, как компонентов, так и системы в целом. Описание поведения отдельных элементов недостаточно для определения поведения биологической системы как целого [14]. Американский ученый А. Эддингтон ввел различие между первичными и вторичными законами. Первичным законам подчиняется поведение отдельных элементов, в то время как вторичные применимы в целом к системе. Эти законы могут быть универсальными, характерными для всех биологических систем, так и специфичными, отражающими особенности конкретно взятой системы. К *универсальным законам* относится тенденция уменьшения энтропии биологического развития, которая характеризует эволюцию человеческого общества на более высокие ступени организации. *Специфичные законы* зависят от истории, связей и структуры конкретной биологической системы. Энтропия отдельной системы может изменяться по колебательному закону, сохраняя в глобальном масштабе тенденцию к уменьшению. Примером увеличения энтропии биологической системы в процессе ее развития может являться хаотическое состояние общества (кризис, гражданская война) или

Иерархическая организация общества с жесткими связями. Такая организация уходит корнями к идее древнегреческого философа Аристотеля [14] о естественной иерархии и характеризуется однонаправленностью связей (сверху вниз). Связи здесь практически сковывают степень свободы компонентов системы и нейтрализуют процесс коллективного поведения, который способен отобразить систему на более высокие уровни организации. В физике аналогом такой организации общества являются кристаллы, где молекулы скованы связями. Как отмечалось в 1.6.1, структура кристалла является инертной и в случае изоляции такое состояние можно поддерживать бесконечно долго. Биологические системы являются открытыми, что способствует их трансформации во времени с целью адаптации к внешней среде. В процессе функционирования биологической системы из-за воздействия внешней и внутренней среды, в ней могут появляться различного рода флуктуации, которые приводят к образованию локальных областей неравновесности. Это способствует частичной дезадаптации системы, которую она стремится нейтрализовать. При этом в системе с гибкими связями происходит их перестройка для компенсации неравновесности, а система с жесткими связями пытается вытолкнуть неравновесность как инородное тело. Назовем биологическую систему *устойчивой*, если она способна путем развития и самоорганизации нейтрализовать дезадаптацию с внешним и внутренним миром. Данное определение предполагает не только пассивную адаптацию в рамках локальной перестройки старых связей, но и активное преобразование структуры системы. Примером неустойчивости являются общественные системы, у которых стрела развития на определенном промежутке времени направлена к более низким ступеням организации по сравнению с предыдущим состоянием. Большую роль в трансформации системы играет явление резонанса. *Резонанс* – это лавинообразное возрастание флуктуаций системы, в результате чего область

неравновесности системы стремительно расширяется. Небольшая неравновесность эхом откликается по всей системе и захватывает все больше ее компонентов. Он может играть как конструктивную роль, когда способствует переходу системы на более высокий уровень организации, так и деструктивную роль в противоположном случае. Причиной резонанса является дезадаптация системы, которая отображается на компоненты системы и при совпадении с внутренней дезадаптацией индивидуумов, вызывает цепную реакцию. В процессе расширения области неравновесности системы происходит конкуренция между флуктуациями и связями системы, которые стремятся их нейтрализовать. Согласно диалектике Гегеля происходит процесс единства и борьбы противоположностей, который является высшим принципом развития. Чем больше жесткость связей систем конкуренция. В результате, если флуктуации захватывают достаточную часть системы, возникает область неопределенности (окрестность точки бифуркации), которая характеризуется неопределенностью выбора дальнейшего пути развития системы [14]. В этой области система может начинать развиваться по различным траекториям. Так она может возвратиться к прежней структуре в обновленном состоянии, перейти в результате самоорганизации к новой структуре (более высокого или низкого уровня) или впасть в неустойчивое хаотическое состояние до новой точки бифуркации. В области неопределенности система может находиться длительное время. В этом состоянии она характеризуется тем, что становится чувствительной к небольшим воздействиям. При этом даже незначительная идея или действие индивидуума может резонировать (усиливаться) до гипертрофированной величины и захватывать всю систему, порождая соответствующую структуру.

При этом идея должна адекватно отражать состояние компонентов системы, которое зависит от причин дезадаптации, исторического опыта и

от переносчика идеи. Чем больше жесткость связей системы, тем больше вероятность возникновения катаклизмов при переходе к новому качеству и наоборот. Итак, в области неопределенности выбор дальнейшего пути развития во многом определяется случайными факторами, например, наличием соответствующих лидеров, которые могут воспользоваться резонансом для реализации своих идей. После того, как направление развития системы выбрано, ее поведение становится в большей степени детерминированным. Модель взаимодействия биологической системы с внешней средой показана на рисунке 1.6. Согласно [14] в основе главного механизма эволюции лежит игра бифуркаций, как механизмов развития и самоорганизации систем. Однако, вероятно, в процессе развития человечества, как перехода от более низких к более высоким формам организации, этот принцип все больше будет уступать процессу плавного (без катаклизмов) перехода к новому качеству. В любом случае возникает вопрос, каким образом человечеству удалось избежать в глобальном масштабе перманентного хаоса? По всей видимости, благодаря способности к обучению и развитию, историческому опыту, инстинкту самосохранения индивидуумов, а также универсальным ценностям: религия, мораль и т.д. Можно также предположить, что в человечестве заложена эволюционная стрела времени, как переход к более высоким формам организации и которая неразрывно связана с эволюцией Вселенной. Все это позволяет говорить о том, что в человечестве существует инстинкт самосохранения. Таким образом, хаотическое состояние неадекватно отражает человеческое общество, так как входит в противоречие с биологической эволюцией, как возрастанием степени упорядоченности систем. Как уже отмечалось, жесткость связей системы увеличивает вероятность возникновения катаклизмов при ее эволюции как открытой системы. Для гармоничного развития системы в ней должен быть заложен потенциал самоорганизации.

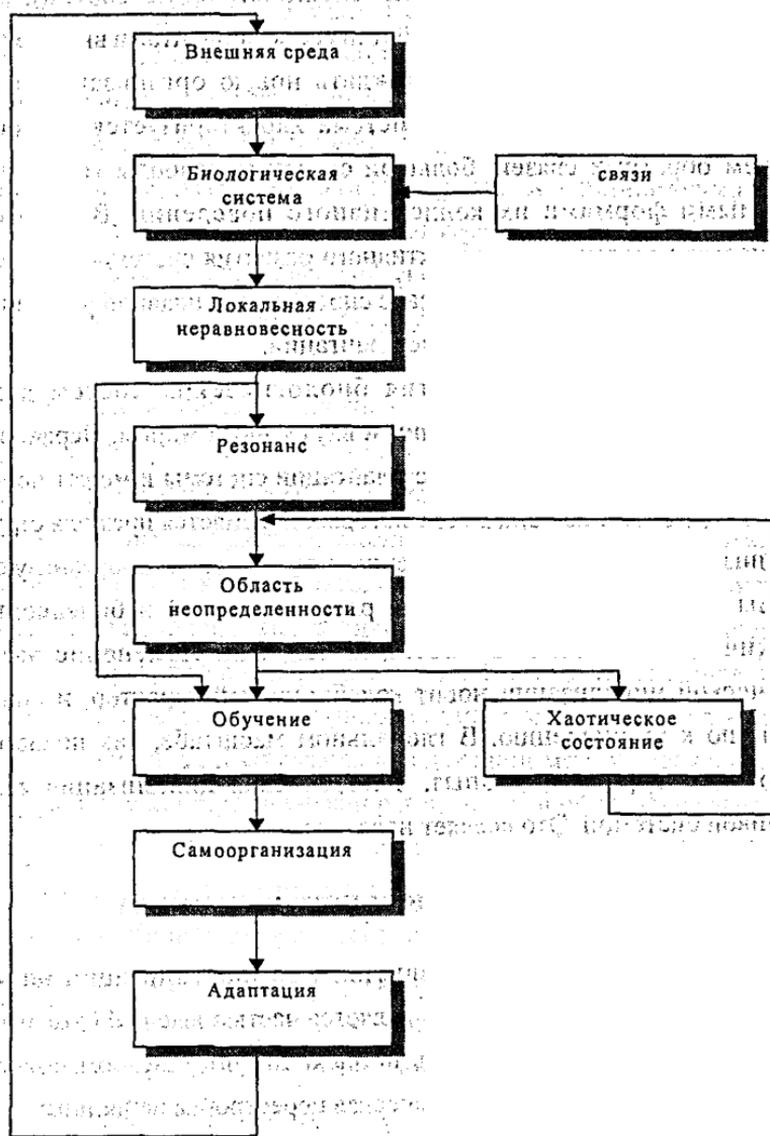


Рис. 1.6. Модель взаимодействия биологической системы с внешней средой.

Назовем *самоорганизующейся системой* такую систему, которая способна сама в процессе взаимодействия ее компонентов и внешней среды, перестраивать свои связи и порождать новую организацию с целью эффективного развития. Такая система характеризуется гибкостью и наличием обратных связей, большой степенью свободы компонентов и различными формами их коллективного поведения. В этом случае уменьшается вероятность деструктивного развития системы на отдельных временных этапах. При этом резонанс способствует плавной реорганизации системы с целью нейтрализации дезадаптации.

Итак, источником развития биологических систем является неравновесное состояние с внешним и внутренним миром. Неравновесное состояние является результатом дезадаптации системы и может порождать процессы самоорганизации, в ходе которых изменяется прежняя структура. Реорганизация происходит естественным путем для самоорганизующейся системы (в ней заложен принцип реорганизации) и болезненно для иерархической системы с жесткими связями. Изменение энтропии человеческой цивилизации носит колебательный характер, и она имеет тенденцию к уменьшению. В глобальном масштабе, как показывает в частности исторический опыт, человеческая цивилизация является устойчивой системой. Это вселяет надежды.

1.6.4. Самоорганизация отдельного организма

Рассмотрим процессы развития и самоорганизации на уровне отдельного индивидуума, который является частью высшей биологической системы. Самоорганизация здесь происходит посредством обучения, в результате чего происходит динамическая перестройка нейронных структур головного мозга и как следствие изменяются связи индивидуума с другими компонентами системы. Процесс этот происходит непрерывно и служит для

адаптации организма с внешним и внутренним миром. С точки зрения абсолютного знания человек все время находится в состоянии хаоса (неопределенности), которое характеризует степень незнания. В процессе познания уменьшается величина неопределенности, и в этом смысле энтропия уменьшается. Источником развития индивидуума является неравновесное состояние с внешним и внутренним миром, которое возникает в результате воздействий внешней среды. Это приводит к возбуждению соответствующих нейронных структур головного мозга, которые перестраиваются таким образом, чтобы поведение индивидуума нейтрализовало возникшую неравновесность. Происходит самоорганизация головного мозга, что вызывает изменение связей индивидуума в популяции. Большую роль на процессы онтогенеза и самоорганизации организма оказывает явление резонанса. *Резонанс* – это резкое возрастание возбуждения отдельных нейронных структур головного мозга. Он может происходить в результате взаимодействия с внешней средой или под воздействием химических препаратов. Сдерживающими факторами резонанса являются связи, наложенные на организм (инерционность синапсов, мораль, инстинкт самосохранения и т.д.). От исхода конкуренции между сдерживающими и возбуждающими факторами зависит развитие нервной активности организма. Резонанс является одним из способов адаптации организма и может играть как положительную, так и отрицательную роль. *Положительный резонанс* приводит к мобилизации ресурсов организма для достижения цели, а *отрицательный резонанс* приводит к стрессам, депрессиям и т.п. В результате резонанса происходит динамическая перестройка сетей головного мозга. Большую роль при этом играют положительные обратные связи, которые усиливают возбуждение нервной активности. При мыслительных операциях резонанс может привести к озарению (инсайту), когда найдено решение какой-то проблемы. Например, при идентификации образа

происходит воспоминание связанных с ним атрибутов (имя, профессия и т.д.). **Идентификация** произойдет за счет резонанса, когда входному образу будут однозначно поставлены его атрибуты. Аналогичные процессы происходят при решении какой-либо сложной проблемы и при других творческих процессах. Однако в этом случае величина возбуждения нейронов намного выше и может продолжаться длительный период, пока не будет найдено, по мнению индивидуума, решение проблемы. Ограничивающими факторами при этом процессе является инерционность нейронных структур головного мозга. В состоянии резонанса даже небольшое воздействие может резко усиливаться и приводить к *инсайту*, если оно адекватно отражает причину нервной активности. Следовательно, большую роль при высших творческих процессах индивидуума играют случайные факторы, так как для этого необходимо, чтобы флуктуация, приводящая к инсайту, была приложена в определенном времени и пространстве к объекту. Аналогичные процессы происходят при взаимодействии индивидуума с системой. При отрицательном резонансе, как уже отмечалось, возбуждение определенных нейронных структур головного мозга может приводить к неврозам, параличу, шизофрении и т.д.

Источником возбуждения здесь являются *аффекты*, которые могут быть вызваны различными ассоциациями, например, словом. В результате конкуренции между возбуждающими и сдерживающими факторами (по Фрейду цензура) за счет положительных обратных связей нейронных структур может происходить процесс усиления нервной активности. При этом, как и в предыдущем случае, организм становится очень чувствительным к некоторым воздействиям, которые способны привести к катастрофическим последствиям. Так депрессивные факторы могут привести к возбуждению нейронных сетей, связанных с этими факторами и нейтрализации цензуры. Положительная обратная связь усиливает это возбуждение и происходит: “

Глава 1. Основы естественного интеллекта

заикливание". В этом состоянии небольшое воздействие может вызвать неустойчивость организма, что способно привести к суициду. Как видно, случайные факторы здесь тоже играют большую роль. В состоянии резонанса поведение индивидуума становится трудно прогнозируемым. Аналогичные процессы происходят при других психических травмах. Так, амнезия и паралич являются одной из форм адаптации организма, в результате которой происходит нейтрализация процесса нарастания возбуждений. Нейтрализация происходит за счет блокирования соответствующих нейронных структур головного мозга, что вызывает эти болезни. Этот процесс З.Фрейд называл *вытеснением* [9]. При обратном процессе разблокирования травмы могут исчезнуть. Разблокирование нейронных структур может происходить как под гипнозом, так и при моделировании стрессовых ситуаций, которые привели к травме. Как гласит один из постулатов З.Фрейда то, что спровоцировано сознанием им же может быть устранено.

Итак, самоорганизация отдельного организма может приводить к различным последствиям. Большую роль при этом играет явление резонанса и сопутствующие ему случайные факторы.

Изменение энтропии в период активной деятельности организма носит колебательный характер, и она имеет тенденцию к уменьшению. В пассивном состоянии, когда начинают ликвидироваться нейроны головного мозга, происходит увеличение энтропии, которое достигает максимального значения при переходе организма в равновесное состояние с внешним миром.

1.7. Искусственные нейронные системы

Интерес к нейроинтеллекту возник еще на заре развития вычислительной техники. В его основе лежит нейронная организация искусственных систем, которая имеет биологические предпосылки.

Способность биологических систем к обучению, самоорганизации и адаптации имеет большое преимущество по сравнению с современными вычислительными системами. Достоинством компьютерных систем является большая скорость распространения информации и возможность учета большого объема знаний, накопленных человечеством в этой области. Разработка искусственных разумных систем, которые соединяют преимущества биологических существ и современной вычислительной техники, создает потенциальные предпосылки для перехода к качественно новому этапу эволюции в вычислительной технике.

Первые шаги в области искусственных нейронных сетей сделали в 1943 году В.Мак-Калох (W.McCulloch) и В.Питтс (W.Pitts). Они показали, что при помощи пороговых нейронных элементов можно реализовать исчисление любых логических функций [15]. В 1949 году Дональд Хебб предложил правило обучения [16], которое стало математической основой для обучения ряда нейронных сетей. В 1957-1962 годах Ф.Розенблатт предложил и исследовал модель нейронной сети, которую он назвал персептроном. Результаты исследований он обобщил в книге "Принципы нейродинамики", которая имеет большое значение для развития нейронных сетей [17]. В 1959 году В.Видроу (W.Widrow) и М.Хофф (M.Hoff) предложили процедуру обучения для линейного адаптивного элемента, который они назвали "ADALINE" [18]. Процедура обучения получила название "дельта правило". В 1969 году М.Минский (M.Minsky) и С.Пайперт (S.Papert) опубликовали монографию "Персептроны" [19], в которой осуществили математический анализ персептрона и показали ограничения, присущие ему. Выводы их были довольно пессимистичными и это сыграло негативную роль для дальнейшего развития исследований в области нейронных сетей. Работы в области персептронных сетей были практически приостановлены. В 70-е годы появился ряд работ в области ассоциативной памяти. Так Андерсон

(Anderson) предложил в 1977 году модель линейной ассоциативной памяти [20]. В этом направлении продолжил исследования Т.Кохонен (Т.Kohonen), который предложил модель оптимальной линейной ассоциативной памяти [21]. В 1976 году С.Гроссберг (S.Grossberg) разработал теорию адаптивного резонанса [22], которая может быть использована для построения ассоциативной памяти.

В 80-е годы происходит значительное усиление интенсивности исследований в области нейронных сетей. Джон Хопфилд (John Hopfield) в 1982 году произвел анализ устойчивости нейронных сетей с обратными связями и предложил использовать их для решения задач оптимизации [23]. Тео Кохонен разработал и исследовал самоорганизующиеся нейронные сети [24]. Ряд авторов (Rumelhart, Hinton, Williams) предложил и алгоритм обратного распространения ошибки, который стал мощным средством для обучения многослойных нейронных сетей [25]. В 1987 году под эгидой общества IEEE (Institute of Electrical and Electronic Engineer's) проводится первая международная конференция в области нейронных сетей.

Большой вклад в развитие теории нейронных сетей внесли российские ученые Галушкин А.И. и Горбань А.Н., а также украинские - Куссиль Э.М. и Ивахненко А.Г.

В настоящее время исследования в области искусственных нейронных сетей ориентированы в основном на создание специализированных систем систем для решения конкретных задач. Разработано большое количество нейросистем, которые применяются в различных областях: прогнозирование, управление, диагностика в медицине и технике, распознавание образов и т.д. Рынок продуктов в области нейроинтеллекта растет стремительным образом. Происходит постепенное накопление критической массы для создания универсальных нейросистем, способных к различного рода интеллектуальной деятельности. В глобальном масштабе задача состоит в

создании искусственного разума, обладающего способностью к воспроизводству и эволюции.

В последнее время наблюдается тенденция объединения теории искусственных нейронных сетей и теории хаоса. Стандартное развитие технических систем состояло в предписывании системе определенных состояний, в которых она могла находиться. Такие состояния характеризуют область устойчивости системы. Как отмечалось в предыдущих разделах, процессы самоорганизации как переход на более высокие стадии развития, происходят за границей области устойчивости. Проблема здесь состоит в том, чтобы направить неустойчивость на развитие систем.

1.8. Классификация искусственных нейронных сетей

Нейронные сети можно классифицировать в зависимости от различных характеристик:

1. По типу входной информации различают:

а) Аналоговые нейронные сети, которые используют информацию в форме действительных чисел.

б) Двоичные нейронные сети. Они оперируют с информацией, представленной в двоичном виде.

2. По характеру обучения:

а) С учителем, когда известно выходное пространство решений нейронной сети.

б) Без учителя. В этом случае нейронная сеть формирует выходное пространство решений только на основе входных воздействий. Такие сети называются самоорганизующимися.

3. По характеру настройки синапсов:

а) Сети с фиксированными связями. В этом случае весовые

коэффициенты нейронной сети выбираются сразу, исходя из условия задачи. При этом

$$\frac{dW}{dt} = 0,$$

где W – характеризует весовые коэффициенты сети

б) Сети с динамическими связями. Для них в процессе обучения происходит настройка синаптических связей, т.е.

$$\frac{dW}{dt} \neq 0.$$

4. По методу обучения:

- 4.1. Нейронные сети с алгоритмом обратного распространения ошибки.
- 4.2. Нейронные сети с конкурентным обучением.
- 4.3. Нейронные сети, использующие правило Хебба.
- 4.4. Нейронные сети с гибридным обучением, в которых используются различные алгоритмы обучения.

5. По характеру связей:

- 5.1. Нейронные сети с прямыми связями (feed-forward networks). В таких сетях происходит распространение информации только в одном направлении от слоя к слою (рис 1.7.). На рис 1.7. W – характеризует синаптические связи, а F – оператор нелинейного преобразования нейронной сети. К такого типа сетям относятся различного рода

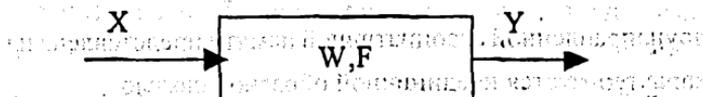


Рис. 1.7. Нейронная сеть с прямыми связями.

персептронные сети: однослойные, многослойные; гомогенные и гетерогенные.

5.2. Нейронные сети с обратным распространением информации (feedback networks). Они характеризуются как прямым, так и обратным распространением информации между слоями нейронной сети. К сетям с такими связями относятся следующие:

5.2.1. Релаксационные, в которых циркуляция информации происходит до тех пор, пока не перестанут изменяться выходные значения нейронной сети (состояние равновесия). К ним относятся нейронные сети Хопфилда, Хэмминга и двунаправленная ассоциативная память. Общая структура сети Хопфилда изображена на рис. 1.8. Она характеризуется единичной обратной связью. Структура

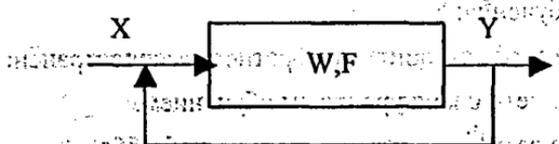


Рис. 1.8. Структура сети Хопфилда.

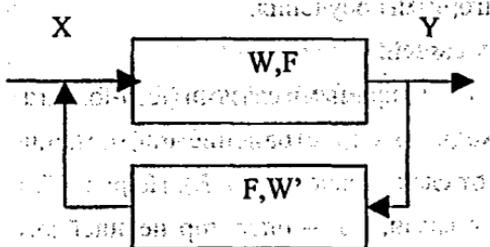


Рис. 1.9. Двунаправленная ассоциативная память.

двунаправленной ассоциативной памяти представлена на рис 1.9. Она характеризуется неединичной обратной связью.

5.2.2. Многослойные сети, в которых отсутствует процесс релаксации.

Архитектура их базируется на многослойном персептроне, а в основе обучения лежит алгоритм обратного распространения ошибки. К таким сетям относятся следующие:

5.2.2.1. Рекуррентные сети, в которых присутствует обратная связь между входом и выходом. Выходное значение, при этом определяется

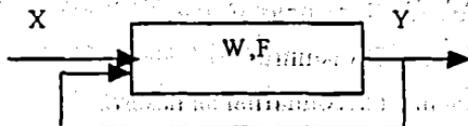


Рис. 1.10. Рекуррентная сеть.

в зависимости как от входных, так и от предшествующих выходных значений нейронной сети (рис. 1.10).

5.2.2.2. Рециркуляционные нейронные сети. Такие сети характеризуются как прямым $y = f(x)$, так и обратным $x = f(y)$ преобразованием информации. (рис. 1.11). В рециркуляционных сетях

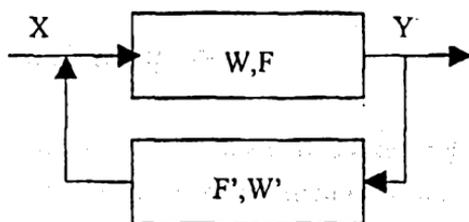


Рис. 1.11. Рециркуляционная нейронная сеть.

обучение производится без учителя, т.е. они являются самоорганизующимися в процессе работы

6. По архитектуре и обучению различают следующие известные нейронные сети:

6.1. Персептронные сети с прямыми связями.

6.2. Самоорганизующиеся нейронные сети. К ним относятся следующие

сети:

6.2.1. Нейронные сети Кохонена.

6.2.2. Нейронные сети адаптивного резонанса.

6.2.3. Рециркуляционные нейронные сети.

6.3. Нейронные сети с обратными связями:

6.3.1. Нейронные сети Хопфилда.

6.3.2. Нейронные сети Хэмминга.

6.3.3. Двухнаправленная ассоциативная память.

6.3.4. Рекуррентные нейронные сети.

6.4. Гибридные нейронные сети:

6.4.1. Нейронные сети встречного распространения.

6.4.2. Нейронные сети с радиально-базисной функцией активации.

ГЛАВА 2. ОДНОСЛОЙНЫЕ НЕЙРОННЫЕ СЕТИ

В главе рассматриваются нейронные сети с одним обрабатывающим слоем и прямыми связями (feed forward networks). Большой вклад в разработку таких сетей внесли Розенблатт (Rosenblatt, 1959), Видров и Хофф (Widrow and Hoff, 1960). *Однослойные* нейронные сети формируют линейную разделяющую поверхность, что ограничивает круг решаемых задач. Это показали в 60-х годах Минский и Пайперт (Minsky and Papert) на примере решения задачи “ИСКЛЮЧАЮЩЕЕ ИЛИ”. В главе приводятся основные определения и различные алгоритмы обучения однослойных нейронных сетей. Получено выражение для адаптивного шага обучения. Рассмотрено применение однослойных нейронных сетей в качестве ассоциативной памяти и для прогнозирования временных рядов. Результаты главы опираются на работы [12,17-20,26-29].

2.1. Формальный нейрон

Основным элементом нейронной сети является *формальный нейрон*, который осуществляет операцию нелинейного преобразования суммы произведений входных сигналов на весовые коэффициенты:

$$y = F\left(\sum_{i=1}^n \omega_i x_i\right),$$

где $X = (x_1, x_2, \dots, x_n)$ — вектор входного сигнала; $W = (\omega_1, \omega_2, \dots, \omega_n)$ — весовой вектор; F — оператор нелинейного преобразования.

Схема нейронного элемента изображена на рис. 2.1 и состоит из сумматора и блока нелинейного преобразования F . Каждому i -му входу нейрона соответствует весовой коэффициент ω_i (синапс), который характеризует силу синаптической связи по аналогии с биологическим нейроном.

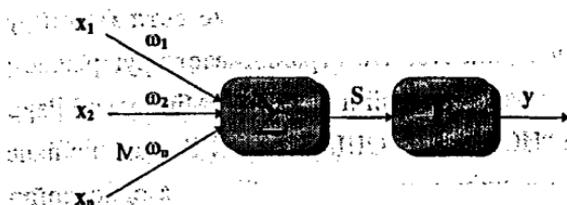


Рис. 2.1. Нейронный элемент.

Сумма произведений входных сигналов на весовые коэффициенты называется *взвешенной суммой*. Она представляет собой скалярное произведение вектора весов на входной вектор:

$$S = \sum_{i=1}^n \omega_i x_i = (W, X) = |W| \cdot |X| \cdot \cos \alpha,$$

где $|W|$, $|X|$ — соответственно длины векторов W и X , $\alpha = \widehat{W, X}$ — угол между векторами W и X .

Длины весового и входного векторов определяются через их координаты:

$$|W| = \sqrt{\omega_1^2 + \omega_2^2 + \dots + \omega_n^2},$$

$$|X| = \sqrt{x_1^2 + x_2^2 + \dots + x_n^2}.$$

Так как для нейронного элемента длина весового вектора после обучения $|W| = const$, то величина взвешенной суммы определяется проекцией входного на весовой вектор:

$$S = |W| \cdot |X| \cdot \cos \alpha = |W| \cdot X_W,$$

где X_W — проекция вектора X на вектор W .

Если входные векторы являются нормированными, т. е. $|X| = const$, то величина взвешенной суммы будет зависеть только от угла между векторами X и W . Тогда при различных входных сигналах взвешенная сумма будет изменяться по косинусоидальному закону. Максимального значения она будет достигать при коллинеарности входного и весового векторов.

Если сила связи ω_i является отрицательной, то такая связь называется *тормозящей*. В противном случае синаптическая связь является *усиливающей*.

Оператор нелинейного преобразования F называется *функцией активации* нейронного элемента. Вектор входного сигнала называется *паттерном входной активности* нейронной сети, а вектор выходного сигнала — *паттерном выходной активности*.

Рассмотренная здесь модель формального нейрона лишь отдаленно напоминает биологический нейрон. Она используется для построения

искусственных нейронных сетей.

2.2. Функции активации нейронных элементов

В качестве оператора нелинейного преобразования могут использоваться различные функции, которые определяются в соответствии с решаемой задачей и типом нейронной сети. Пусть T — порог нейронного элемента, который характеризует расположение функции активации по оси абсцисс. Представим взвешенную сумму, как

$$S = \sum_{i=1}^n \omega_i x_i - T.$$

Рассмотрим наиболее распространенные функции активации нейронных элементов.

2.2.1. Линейная функция

В этом случае выходное значение нейронного элемента равняется взвешенной сумме:

$$y = kS,$$

где k — коэффициент наклона прямой.

Изменение порога линейного элемента эквивалентно сдвигу функции активации по оси абсцисс (рис. 2.2):

2.2.2. Пороговая функция

В качестве пороговой функции активации может использоваться биполярная или бинарная функция. В случае использования пороговой

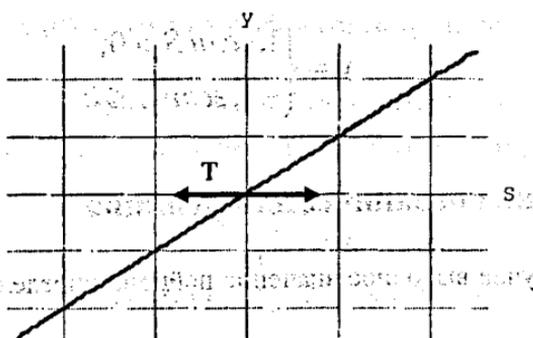


Рис. 2.2. Линейная функция активации.

пороговой функции активации (рис. 2.3):

$$y = \text{sign}(S) = \begin{cases} 1, & \text{если } S > 0; \\ 0, & \text{если } S \leq 0. \end{cases}$$

Если применяется пороговая биполярная функция активации, то выходное значение нейронного элемента:

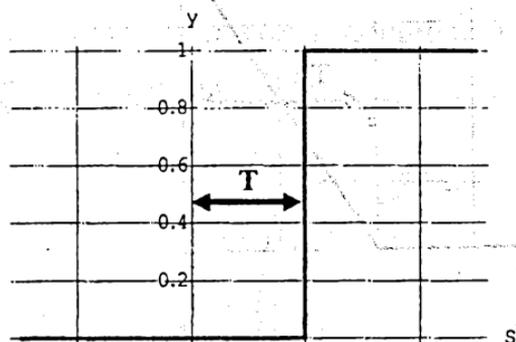


Рис. 2.3. Пороговая функция активации.

$$y = \begin{cases} 1, & \text{если } S > 0; \\ -1, & \text{если } S \leq 0. \end{cases}$$

2.2.3. Линейная ограниченная функция

В этом случае выходное значение нейрона определяется следующим образом:

$$y = \begin{cases} p, & \text{если } S > \alpha; \\ -p, & \text{если } S < -\alpha; \\ S, & \text{если } -\alpha \leq S \leq \alpha. \end{cases}$$

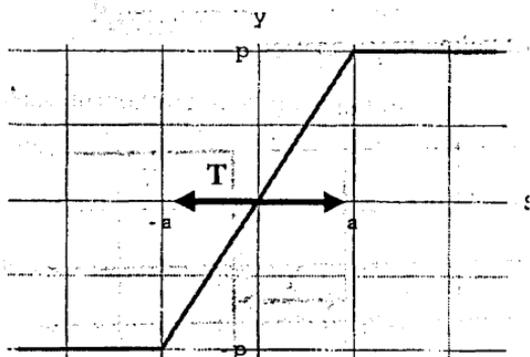


Рис. 2.4. Линейная ограниченная функция

2.2.4. Модифицированная пороговая функция

Используется в двунаправленной ассоциативной памяти. Выходное значение нейронного элемента характеризуется здесь следующей функцией:

$$y(t+1) = \begin{cases} 1, & \text{если } S > 0; \\ y(t), & \text{если } S = 0; \\ -1, & \text{если } S < 0. \end{cases}$$

Здесь t — параметр времени.

2.2.5. Сигмоидная функция

Является непрерывной, возрастающей функцией в диапазоне значений $[0, 1]$:

$$y = \frac{1}{1 + e^{-cs}},$$

где $c > 0$ — коэффициент, характеризующий ширину сигмоидной функции по оси абсцисс (рис. 2.5):

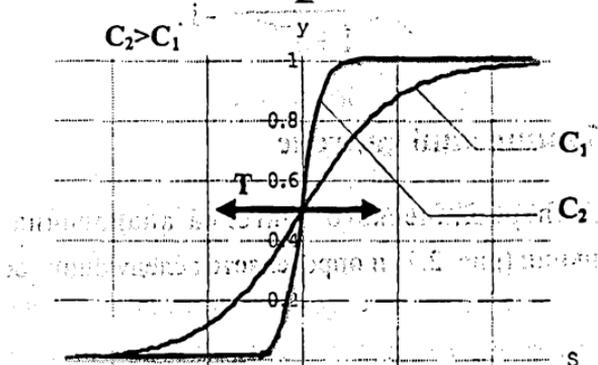


Рис. 2.5. Сигмоидная функция активации.

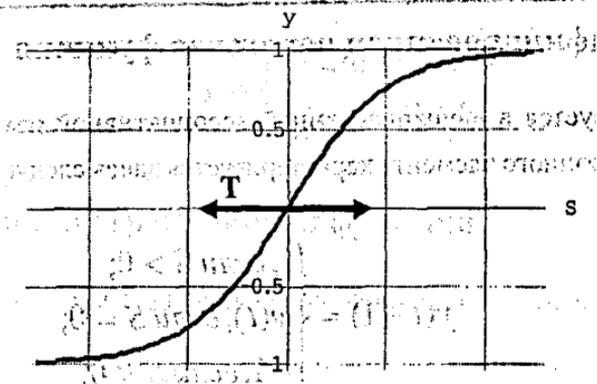


Рис. 2.6.. Биполярная сигмоидная функция активации.

Сигмоидная функция является монотонной и всюду дифференцируемой. Поэтому она получила широкое распространение в искусственных нейронных сетях.

2.2.6. Биполярная сигмоидная функция

Данная функция является расширением сигмоидной функции в диапазон значений $[-1, 1]$ (рис. 2.6):

Выходное значение нейронного элемента определяется, как

$$y = \frac{2}{1 + e^{-cs}} - 1,$$

2.2.7. Гиперболический тангенс

Функция гиперболического тангенса аналогична биполярной сигмоидной функции (рис. 2.7) и определяется следующим образом:

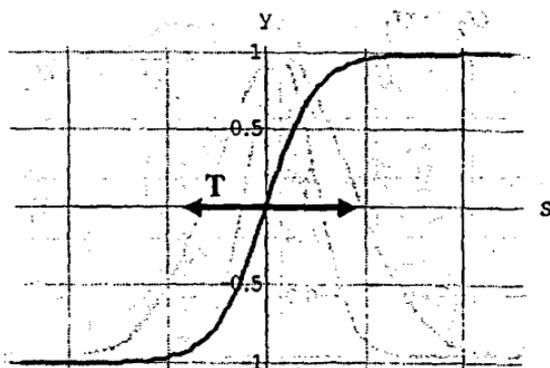


Рис. 2.7: Гиперболический тангенс.

$$y = th(cS) = \frac{e^{cS} - e^{-cS}}{e^{cS} + e^{-cS}},$$

где c , как и в случае с сигмоидной функцией, характеризует ширину функции гиперболический тангенс по оси абсцисс.

2.2.8. Радиально-базисная функция

Она характеризуется функцией Гаусса для нормального закона распределения (рис. 2.8). В соответствии с ней

$$y = \exp\left(-\frac{S^2}{2\sigma^2}\right),$$

где σ — среднеквадратичное отклонение, которое характеризует ширину радиально-базисной функции.

Величина S в данном случае будет определяться в соответствии с евклидовым расстоянием между входным и весовым векторами:

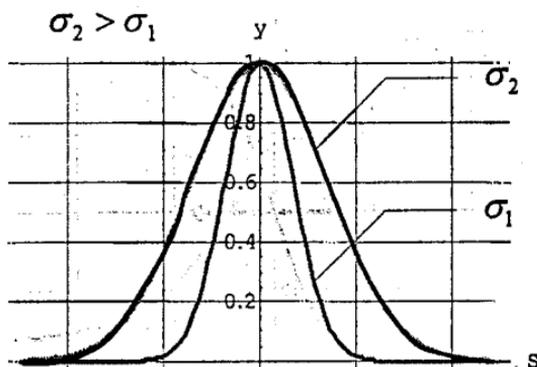


Рис. 2.8. Функция Гаусса.

$$S^2 = |X - W|^2 = \sum (x_i - \omega_i)^2.$$

В заключение отметим, что применение различных функций активации определяется классом решаемых нейронной сетью задач. Помимо перечисленных могут применяться и другие функции активации нейронных элементов, которые адекватно отражают решаемую задачу. Так, в наших исследованиях для прогнозирования временных рядов использовалась логарифмическая функция активации нейронных элементов. Результаты применения этой функции будут обсуждаться далее.

2.3. Нейронные сети с пороговой функцией активации

Совокупность нейронных элементов и связей между ними называется *нейронной сетью*. *Слоем нейронной сети* называется множество нейронных элементов, на которые в каждый такт времени параллельно поступает информация от других нейронных элементов сети. В данном разделе будут

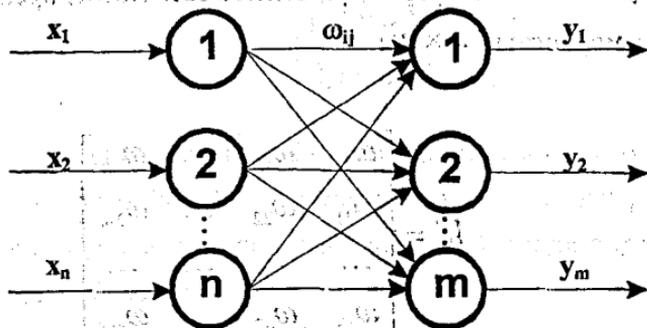


Рис. 2.9. Топология однослойной нейронной сети

рассматриваться нейронные сети, состоящие из одного слоя нейронных элементов, который осуществляет обработку входной информации. Такие сети принято изображать в виде двухслойной нейронной сети, где первый слой нейронных элементов является распределительным, а второй — обрабатывающим. Распределительный слой передает входные сигналы на обрабатывающий слой нейронных элементов, который преобразует входную информацию в соответствии с синаптическими связями и функцией активации (рис. 2.9). При этом каждый нейрон распределительного слоя имеет синаптические связи со всеми нейронами обрабатывающего слоя.

Тогда выходное значение j -го нейронного элемента второго слоя можно представить как

$$y_j = F(S_j) = F\left(\sum_{i=1}^n \omega_{ij} x_i - T_j\right), \quad (2.1)$$

где T_j — порог j -го нейронного элемента выходного слоя; ω_{ij} — сила синаптической связи между i -м нейроном распределительного слоя и j -м нейроном обрабатывающего слоя.

Совокупность весовых коэффициентов сети можно представить в виде матрицы размерностью $n \times m$:

$$W = \begin{bmatrix} \omega_{11} & \omega_{12} & \dots & \omega_{1m} \\ \omega_{21} & \omega_{22} & \dots & \omega_{2m} \\ \dots & \dots & \dots & \dots \\ \omega_{n1} & \omega_{n2} & \dots & \omega_{nm} \end{bmatrix}$$

Тогда вектор-столбец взвешенной суммы в матричном виде определяется следующим образом:

$$S = W^T X - T, \quad (2.2)$$

где T — вектор-столбец порогов нейронных элементов второго слоя.

Рассмотрим нейронную сеть с двумя нейронами входного и одним нейроном выходного слоя (рис. 2.10):

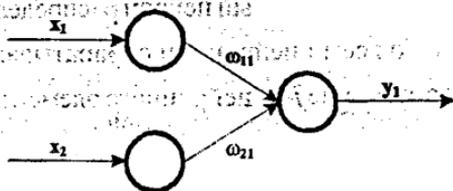


Рис. 2.10. Сеть с одним выходным нейроном.

Тогда взвешенная сумма:

$$S = \omega_{11}x_1 + \omega_{21}x_2 - T_1. \quad (2.3)$$

Соответственно выходное значение нейронной сети:

$$y = \begin{cases} 1, & \text{если } S > 0; \\ 0, & \text{если } S \leq 0. \end{cases}$$

Такая сеть осуществляет линейное разделение входного пространства сигналов на два класса [17, 26], и может использоваться для решения задач классификации образов. Уравнение разделяющей линии определяется при этом следующим образом:

$$\omega_{11}x_1 + \omega_{21}x_2 - T_1 = 0. \quad (2.4)$$

Она разделяет область решений, соответствующую одному классу, от другого класса и называется *дискриминантной линией*. Выражая из (2.4) x_2 , получим

$$x_2 = \frac{T_1}{\omega_{21}} - \frac{\omega_{11}}{\omega_{21}} x_1. \quad (2.5)$$

В системе координат (x_2, x_1) последнее уравнение изображает прямую линию (рис. 2.11), которая отделяет один класс от другого.

При этом величина $\frac{T_1}{|W|}$ представляет собой расстояние от центра координат до прямой. Тогда, если

$$X_w > \frac{T_1}{|W|},$$

то скалярное произведение

$$S = (W, X) > T_1$$

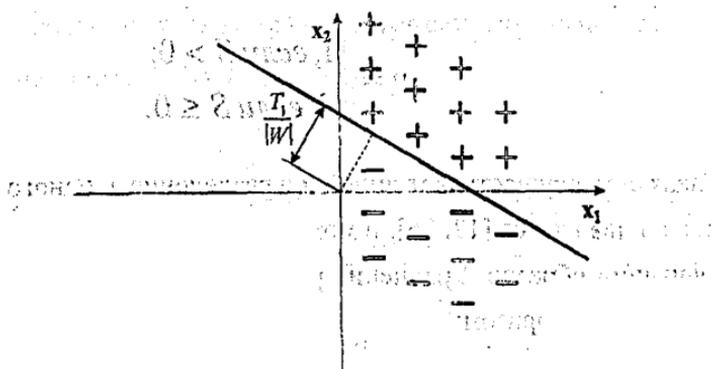


Рис. 2.11. Дискриминантная линия.

и $y = 1$. В противном случае $y = 0$. Здесь X_W — проекция входного вектора на весовой вектор.

Если размерность входного сигнала $n=3$, то разделяющей поверхностью будет являться плоскость. При $n>3$ разделяющей поверхностью является гиперплоскость.

Рассмотрим решение простейших задач логических операций на такой сети. Такие задачи решаются с использованием нейронной сети, если их пространство решений можно разбить на два класса. Это схематично представлено на рис. 2.12 соответственно для операций логического “И”, “ИЛИ” и “ИСКЛЮЧАЮЩЕЕ ИЛИ”.

Из рисунка следует, что рассмотренная нейронная сеть может решить задачу типа “И” и “ИЛИ”, но неспособна решить задачу “ИСКЛЮЧАЮЩЕЕ ИЛИ”.

Таким образом линейная разделяющая поверхность, формируемая нейронной сетью с одним слоем обрабатывающих нейронных элементов ограничивает круг решаемых задач. Это в свое время показали американские

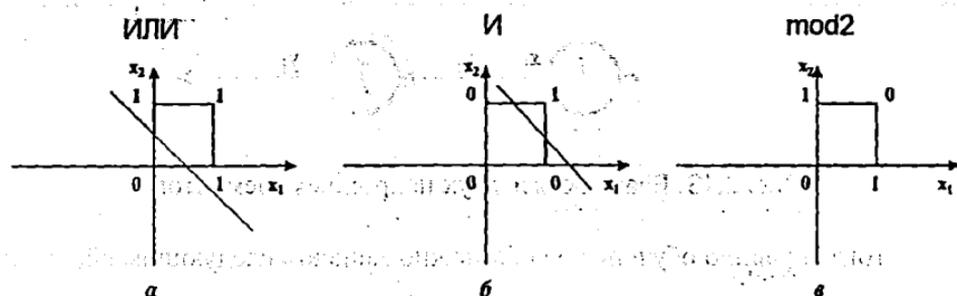


Рис. 2.12. Графическая интерпретация решения логических задач.

ученые Минский и Пайперт [19], которые сделали пессимистические выводы насчет дальнейшего развития нейронных сетей.

2.4. Правило обучения Хебба

Самоадаптация и самоорганизация нейронных сетей достигается в процессе их обучения, в ходе которого происходит определение синаптических связей между нейронными элементами. Обучающие правила определяют, как изменяются весовые коэффициенты в ответ на входное воздействие.

Правило обучения Хебба [12], как отмечалось в главе 1, имеет биологические предпосылки. Оно является основой многих методов обучения нейронных сетей. Согласно правилу Хебба, обучение происходит в результате усиления силы связи (синаптического веса) между одновременно активными нейронами. Исходя из этого, часто используемые в сети связи усиливаются, что объясняет феномен обучения путем повторения и привыкания.

Пусть имеются два нейронных элемента i и j , между которыми существует сила связи, равная ω_{ij} (рис. 2.13):

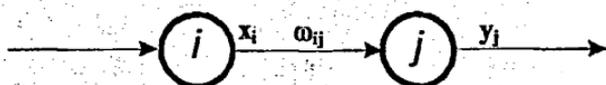


Рис. 2.13. Взаимосвязь двух нейронных элементов.

Тогда правило обучения Хебба можно записать следующим образом:

$$\omega_{ij}(t+1) = \omega_{ij}(t) + x_i y_j, \quad (2.6)$$

где t — время, x_i и y_j соответственно выходное значение i -го и j -го нейрона.

В начальный момент времени предполагается, что

$$\omega_{ij}(t=0) = 0, \quad \forall i, j.$$

Рассмотрим применение правила Хебба для простейшей нейронной сети, состоящей из двух входных и одного выходного нейрона (рис. 2.10). В такой сети порог T выходного нейронного элемента является скрытым в этом элементе. При операциях с нейронными сетями порог нейронного элемента можно вынести за его пределы и изобразить, как синаптическую связь (рис. 2.14) с весовым коэффициентом, равным значению T .

Так как входное значение, подаваемое на дополнительный нейрон, равняется -1 , то взвешенная сумма определяется, как

$$S = \omega_{11}x_1 + \omega_{21}x_2 - T. \quad (2.7)$$

Данное выражение является эквивалентным выражению (2.3). Обучение нейронной сети происходит путем настройки весовых коэффициентов и порогов нейронных элементов. Поэтому рассмотренное выше преобразование позволяет настраивать весовые коэффициенты и пороги сети, как единое целое.

Правило Хебба для нейронной сети, изображенной на рис. 2.14, можно

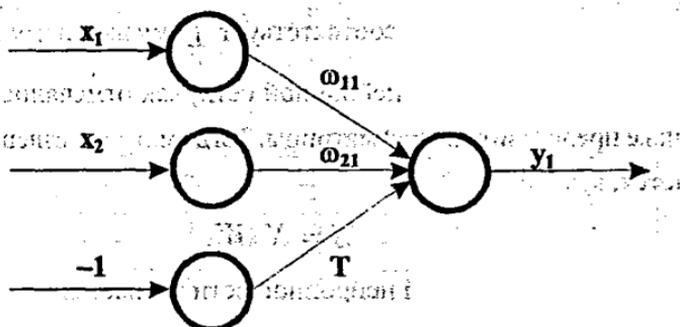


Рис. 2.14. Представление порогового значения в виде синаптической связи.

представить в виде следующих выражений:

$$\omega_{11}(t+1) = \omega_{11}(t) + x_1 y_1, \quad (2.8)$$

$$\omega_{21}(t+1) = \omega_{21}(t) + x_2 y_1, \quad (2.9)$$

$$T(t+1) = T(t) - y_1. \quad (2.10)$$

Аналогичным образом правило Хебба записывается для нейронной сети большей размерности.

Рассмотрим матричную формулировку правила Хебба. Пусть имеется L входных образов, подаваемых на нейронную сеть, и n — размерность одного образа. Тогда совокупность входных образов можно представить в виде следующей матрицы:

$$X = \begin{bmatrix} x_1^1 & x_2^1 & \dots & x_n^1 \\ x_1^2 & x_2^2 & \dots & x_n^2 \\ \dots & \dots & \dots & \dots \\ x_1^L & x_2^L & \dots & x_n^L \end{bmatrix},$$

где $X^L = (x_1^L, x_2^L, \dots, x_n^L)$ соответствует L -му входному образу.

Весовые коэффициенты нейронной сети, как отмечалось в разделе 2, можно также представить в виде матрицы. Тогда матрица взвешенной суммы определяется, как

$$S = X \cdot W, \quad (2.11)$$

а матрица выходных значений нейронной сети равняется:

$$Y = F(S). \quad (2.12)$$

Правило Хебба в матричной форме можно представить в следующем виде:

$$W = X^T \cdot Y. \quad (2.13)$$

Правило Хебба может использоваться, как при обучении с учителем, так и без него. Если в качестве выходных значений Y нейронной сети используются эталонные значения, то правило Хебба будет соответствовать обучению с учителем. При использовании в качестве Y реальных значений, которые получаются при подаче на вход сети входных образов, правило Хебба соответствует обучению без учителя.

В последнем случае весовые коэффициенты нейронной сети в начальный момент времени инициализируются случайным образом. Обучение с использованием правила Хебба заканчивается после подачи всех имеющихся входных образов на нейронную сеть. Следует также отметить, что в общем случае правило Хебба не гарантирует сходимости процедуры обучения нейронной сети.

Рассмотрим решение задачи логическое "ИЛИ" при помощи процедуры обучения Хебба. Будем использовать биполярную кодировку двоичных сигналов. Пусть размерность входного сигнала $n=2$. Тогда общее количество

Глава 2. Однослойные нейронные сети

входных образов, подаваемых на нейронную сеть, $L=4$. В таблице 2.1 изображена процедура обучения, соответствующая правилу Хебба.

ТАБЛИЦА 2.1

x_1	x_2	x_3	y	ω_{11}	ω_{21}	T
-1	-1	-1	-1	1	1	1
1	-1	-1	1	2	0	0
-1	1	-1	1	1	1	-1
1	1	-1	1	2	2	-2

В данной таблице x_3 соответствует постоянному сигналу -1 , подаваемому на третий вход нейронной сети (рис. 2.14). В данном случае происходит обучение с учителем. Поэтому в качестве y используются эталонные значения. В начальный момент времени предполагается, что $\omega_{11}(0) = 0$, $\omega_{21}(0) = 0$, $T(0) = 0$. Обучение происходит согласно выражениям (2.8), (2.9), (2.10) и заканчивается после подачи всех входных образов на сеть. В результате обучения получается следующее уравнение разделяющей линии:

$$2x_1 + 2x_2 + 2 = 0,$$

$$x_1 + x_2 = -1.$$

Графическая интерпретация решения задачи логическое "ИЛИ" приведена на рис. 2.15.

Как следует из рисунка, сеть обучилась поставленной задаче. Рассмотрим матричную формулировку процедуры обучения. Тогда

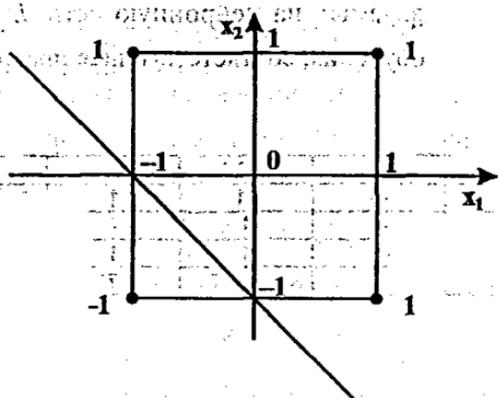


Рис. 2.15. Решение задачи логическое «ИЛИ».

$$W = X^T Y = \begin{bmatrix} -1 & 1 & -1 & 1 \\ -1 & -1 & 1 & 1 \\ -1 & -1 & -1 & -1 \end{bmatrix} \cdot \begin{bmatrix} -1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \\ -2 \end{bmatrix},$$

что аналогично результату, приведенному в таблице 2.1.

Как уже отмечалось, правило Хебба не гарантирует сходимости процедуры обучения. Так, если взять, например, размерность входного сигнала $n=5$, то процедура обучения Хебба для задачи логического «ИЛИ» не приведет к желаемому результату. Данное правило, в основном, используется в различного рода нейросетевой памяти.

2.5. Процедура обучения Розенблатта

Данную процедуру предложил американский ученый Ф. Розенблатт в 1959 году для нейронной сети, которую он назвал персептроном [17].

Глава 2. Однослойные нейронные сети

Перцептрон — это сеть, состоящая из S, A и R нейронных элементов (рис. 2.16). Нейроны слоя S называются *сенсорными* и предназначены для формирования входных сигналов в результате внешних воздействий. Нейроны слоя A называются *ассоциативными* и предназначены для непосредственной обработки входной информации. Нейроны слоя R называются *эффекторными*. Они служат для передачи сигналов возбуждения к соответствующему объекту, например, к мышцам.

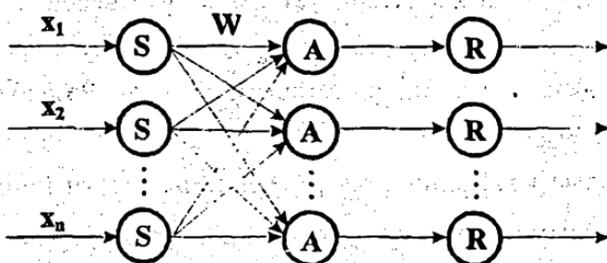


Рис. 2.16. Структура перцептрона.

Таким образом, перцептрон Розенблатта содержит один слой обрабатывающих нейронных элементов, в качестве которых используются нейронные элементы с пороговой функцией активации. Поэтому обучение перцептрона происходит путем настройки весовых коэффициентов W между слоями S и A. В дальнейшем будем использовать стандартную интерпретацию нейронной сети Розенблатта, которая состоит из набора входных нейронов, выполняющих чисто распределительные функции, и набора выходных нейронов с пороговой функцией активации.

Математическую формулировку правила обучения Розенблатта можно

представить в следующем виде:

$$\omega_{ij}(t+1) = \omega_{ij}(t) + \alpha x_i t_j \quad (2.14)$$

где t_j — эталонное значение j -го выхода нейронной сети; α — характеризует скорость обучения сети.

Величина скорости обучения характеризуется следующими значениями:

$$\alpha = \text{const}, \quad 0 < \alpha \leq 1.$$

Процедура обучения Розенблатта называется алгоритмом обучения с подкреплением. Она характеризуется тем, что весовые коэффициенты нейронной сети изменяются только в том случае, если выходная реакция сети не совпадает с эталонной. Алгоритм обучения Розенблатта состоит из следующих шагов:

1. Весовые коэффициенты W нейронной сети инициализируются случайным образом или устанавливаются в нулевое состояние.
2. На входы сети поочередно подаются входные образы X из обучающей выборки, которые трансформируются в выходные сигналы нейронных элементов Y .
3. Если реакция нейронной сети Y_j совпадает с эталонным значением t_j , т. е. $Y_j = t_j$, то весовой коэффициент ω_{ij} не изменяется.
4. Если выходная реакция не совпадает с эталонной, т. е. $Y_j \neq t_j$, то производится модификация весовых коэффициентов по следующему правилу:

$$\omega_{ij}(t+1) = \omega_{ij}(t) + \alpha x_i t_j.$$

5. Алгоритм продолжается до тех пор, пока не станет $y_j = t_j$ для всех входных образов, или не перестанут изменяться весовые коэффициенты.

Согласно теореме сходимости персептрона [19], рассмотренный выше алгоритм сходится за конечное число шагов, если существует решение задачи.

Основные отличия процедуры обучения Розенблатта от правила Хебба заключаются в следующем:

1. В правиле обучения для персептрона присутствует скорость обучения α .
2. Персептрон не изменяет весовые коэффициенты, если выходные сигналы совпадают с эталонными.
3. Входные образы из обучающей выборки в модели персептрона подаются до тех пор, пока не произойдет обучение сети.
4. Персептрон обучается за конечное число шагов, если существует решение задачи.

Линейная разделяющая поверхность, формируемая персептроном, ограничивает круг решаемых им задач. Это исследовали в 1969 г. Минский и Пайперт [19], которые показали, что персептрон не может решить задачу "ИСКЛЮЧАЮЩЕЕ ИЛИ". Выводы их по поводу перспектив персептронной модели были весьма пессимистичными. В связи с этим исследования в области нейронных сетей были почти полностью прекращены вплоть до конца 70-х годов.

2.6. Геометрическая интерпретация процедуры обучения персептрона

Представим взвешенную сумму нейронной сети в виде скалярного

произведения весового вектора на входной вектор:

$$S = (W, X) = |W| \cdot |X| \cdot \cos \gamma, \quad (2.15)$$

где $\gamma = \widehat{W, X}$.

Тогда, если $0 < \gamma < 90^\circ$, то $S > 0$ и $y = 1$. Если $90^\circ \leq \gamma < 270^\circ$, то $S \leq 0$ и $y = -1$. Рассмотрим векторы W и X . Пусть угол между ними $\gamma \geq 90^\circ$ (рис. 2.17). В этом случае $S \leq 0$ и $y = 0$. Предположим, что эталонное значение $t = 1$. Тогда вектор W необходимо преобразовать таким образом, чтобы угол γ стал меньше 90° . Отсюда правило преобразования вектора W будет следующим:

$$W' = W + \alpha X. \quad (2.16)$$

Это схематично представлено на рис. 2.17.

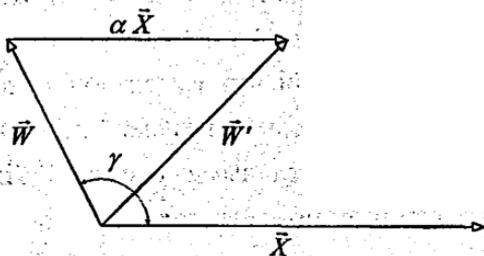


Рис. 2.17. Геометрическое представление обучения персептрона при гамма больше 90 градусов.

Рассмотрим теперь другой случай. Пусть $\gamma < 90^\circ$. Тогда $S > 0$ и

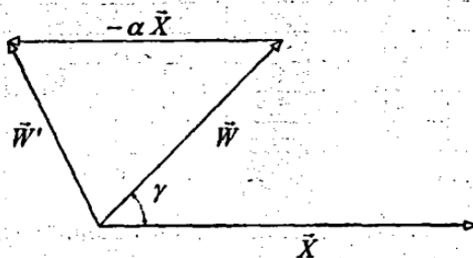


Рис. 2.18. Геометрическое представление обучения персептрона при гамма меньше 90 градусов.

$y = 1$. Предположим, что эталонное значение $t = 0$. Тогда вектор W необходимо преобразовать так, чтобы угол γ стал больше 90° . Правило преобразования можно представить, как

$$W' = W - \alpha X \quad (2.17)$$

Данный процесс представлен на рис. 2.18.

Пусть $W' = W(t+1)$, а $W = W(t)$. Тогда, объединяя приведенные выше два правила в одно, получим следующее выражение для настройки весовых коэффициентов:

$$W(t+1) = W(t) - \alpha(y-t)X = W(t) + \alpha(t-y)X. \quad (2.18)$$

Последнее выражение характеризует правило обучения Розенблатта в общей форме.

Рассмотрим пример процедуры обучения Розенблатта для решения задачи логическое "И". Пусть размерность входного сигнала $n=2$. Процедура обучения представлена в таблице 2.2:

ТАБЛИЦА 2.2

x_1	x_2	x_3	y	r	a_{11}	a_{21}	r
1	1	-1	-1	1	2	2	-2
1	-1	-1	1	-1	0	4	0
-1	1	-1	1	-1	2	2	2
-1	-1	-1	-1	-1	2	2	2

В начальный момент времени здесь предполагалось, что весовые коэффициенты равняются нулю.

В результате обучения получается следующее уравнение разделяющей линии:

$$x_1 + x_2 = 1.$$

Геометрическая интерпретация решения задачи логическое "И" приведена на рис. 2.19:

В заключение отметим, что правило обучения Розенблатта, в отличие от правила Хебба, решает задачи логическое "И" и "ИЛИ" для любой размерности входных сигналов.

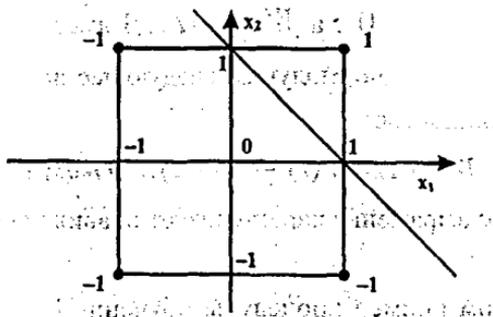


Рис. 2.19. Решение задачи логическое «И».

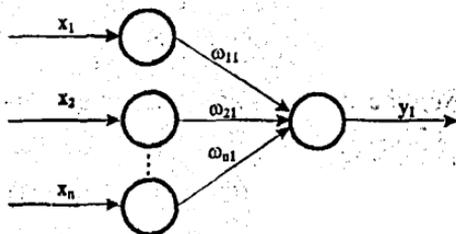


Рис. 2.20. Линейная сеть.

2.7. Правило обучения Видроу-Хоффа

Используется для обучения нейронной сети, состоящей из распределительных нейронов и одного выходного нейрона, который имеет линейную функцию активации (рис.2.20):

Такая сеть называется адаптивным нейронным элементом или "ADALINE" (Adaptive Linear Element). Его предложили в 1960 г. Видроу (Widrow) и Хофф (Hoff) [18]. Выходное значение такой сети определяется, как

$$y_1 = \sum_{j=1}^n \omega_{j1} x_j - T. \quad (2.19)$$

Правило обучения Видроу Хоффа известно под названием *дельта правила* (delta rule). Оно предполагает минимизацию среднеквадратичной ошибки нейронной сети, которая для L входных образов определяется следующим образом:

$$E = \sum_{k=1}^L E(k) = \frac{1}{2} \sum_{k=1}^L (y_1^k - t^k)^2, \quad (2.20)$$

где $E(k)$ — среднеквадратичная ошибка сети для k -го образа; y_1^k и t^k — соответственно выходное и эталонное значение нейронной сети для k -го образа.

Критерий (2.20) характеризуется тем, что при малых ошибках ущерб является также малой величиной, т. к. E меньше чем величина отклонения $(y - t)$. При больших ошибках ущерб возрастает, так как E возрастает с ростом величины ошибки.

Среднеквадратичная ошибка нейронной сети для одного входного образа определяется, как

$$E(k) = \frac{1}{2} (y_1^k - t^k)^2. \quad (2.21)$$

Правило обучения Видроу-Хорфа базируется на методе градиентного спуска в пространстве весовых коэффициентов и порогов нейронной сети. Согласно этому правилу, весовые коэффициенты и пороги нейронной сети необходимо изменять с течением времени по следующим выражениям:

$$\omega_{j1}(t+1) = \omega_{j1}(t) - \alpha \frac{\partial E(k)}{\partial \omega_{j1}(t)}, \quad (2.22)$$

$$T(t+1) = T(t) - \alpha \frac{\partial E(k)}{\partial T(t)}, \quad (2.23)$$

где $j = \overline{1, n}$; α — скорость или шаг обучения. Найдем производные

среднеквадратичной ошибки E по настраиваемым параметрам сети ω_{j1} и T . Тогда

$$\frac{\partial E}{\partial \omega_{j1}(t)} = \frac{\partial E}{\partial y_1^k} \cdot \frac{\partial y_1^k}{\partial \omega_{j1}} = (y_1^k - t^k) x_j^k,$$

$$\frac{\partial E}{\partial T(t)} = \frac{\partial E}{\partial y_1^k} \cdot \frac{\partial y_1^k}{\partial T} = -(y_1^k - t^k),$$

где x_j^k — j -ая компонента k -го образа.

Отсюда получаем следующие выражения для обучения нейронной сети по дельта правилу:

$$\omega_{j1}(t+1) = \omega_{j1}(t) - \alpha (y_1^k - t^k) x_j^k, \quad (2.24)$$

$$T(t+1) = T(t) + \alpha (y_1^k - t^k), \quad (2.25)$$

где $j = \overline{1, n}$.

Выражение (2.24) эквивалентно правилу обучения Розенблатта (2.18), если на выходе линейного адаптивного элемента поставить пороговый элемент. Тогда y_1^k в выражении (2.24) будет являться результатом нелинейного преобразования взвешенной суммы.

Видроу и Хофф доказали [18], что данный закон обучения всегда позволяет находить весовые коэффициенты нейронного элемента таким образом, чтобы минимизировать среднеквадратичную ошибку сети независимо от начальных значений весовых коэффициентов.

Алгоритм обучения, в основе которого лежит дельта правило состоит из следующих шагов:

1. Задается скорость обучения α ($0 < \alpha < 1$) и минимальная среднеквадратичная ошибка сети E_m , которой необходимо достичь в процессе обучения.
2. Случайным образом инициализируются весовые коэффициенты и порог нейронной сети.
3. Подаются входные образы на нейронную сеть и вычисляются векторы выходной активности сети.
4. Производится изменение весовых коэффициентов и порога нейронной сети согласно выражениям (2.24) и (2.25).
5. Алгоритм продолжается до тех пор, пока суммарная среднеквадратичная ошибка сети не станет меньше заданной, т. е. $E \leq E_m$.

В алгоритме Видроу-Хоффа существует проблема выбора значения шага обучения α . Если коэффициент α слишком мал, то процесс обучения является очень длительным. В случае, когда шаг обучения большой, процесс обучения может оказаться расходящимся, то есть не привести к решению задачи. Таким образом сходимость алгоритма обучения не избавляет от разумного выбора значения шага обучения. В некоторых работах [26] предполагается выбирать значение α , которое уменьшается в процессе обучения следующим образом:

$$\alpha(k) = \frac{1}{k},$$

где k — номер итерации в алгоритме обучения. Однако здесь важным

фактором является скорость, с которой $\alpha(k)$ приближается к нулю. Если скорость слишком велика, то процесс обучения может закончиться до того, как будут получены оптимальные результаты.

Таким образом, временная сложность алгоритма обучения и достижение оптимального решения зависит от выбора величины шага обучения. В следующем разделе этот вопрос будет обсуждаться подробно.

2.8. Выбор адаптивного шага обучения

Для ускорения процедуры обучения градиентного спуска, вместо постоянного шага обучения можно использовать *адаптивный шаг* обучения $\alpha(t)$. Назовем адаптивным шагом обучения такой шаг, который целенаправленно выбирается на каждом этапе алгоритма таким образом, чтобы минимизировать среднеквадратичную ошибку сети.

Рассмотрим линейную нейронную сеть, которая состоит из распределительного слоя нейронных элементов и выходного слоя (рис. 2.21.).

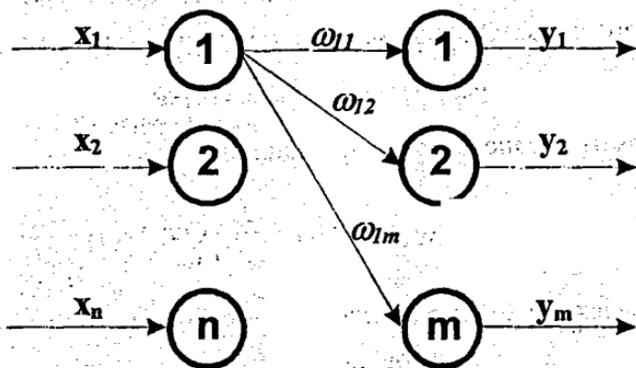


Рис. 2.21. Линейная нейронная сеть.

В качестве нейронов выходного слоя используются нейронные элементы с линейной функцией активации.

Каждый нейрон распределительного слоя имеет синаптические связи со всеми нейронами обрабатывающего слоя. Выходное значение j -го нейрона сети определяется, как

$$y_j = \sum_i \omega_{ij} x_i - T_j \quad (2.26)$$

Среднеквадратичная ошибка сети для всей обучающей выборки будет равна

$$E_s = \frac{1}{2} \sum_k \sum_j (y_j^k - t_j^k)^2 \quad (2.27)$$

Соответственно среднеквадратичную ошибку для одного образа представим в виде:

$$E = \frac{1}{2} \sum_j (y_j - t_j)^2 \quad (2.28)$$

Для нахождения адаптивного шага обучения будем использовать метод наискорейшего спуска. Он заключается в том, чтобы на каждом шаге обучения нейронной сети необходимо выбирать такую скорость обучения $\alpha(t)$, чтобы минимизировать среднеквадратичную ошибку:

$$\alpha(t) = \min E \left\{ (\omega_{ij}(t) - \alpha(t) \frac{\partial E}{\partial \omega_{ij}(t)}); (T_j - \alpha(t) \frac{\partial E}{\partial T_j(t)}) \right\} \quad (2.29)$$

Подставляя в (2.26) выражения для изменения весовых коэффициентов и порогов нейронных элементов, получим:

$$y_j' = \sum_i (\omega_{ij}(t) - \alpha(t) \frac{\partial E}{\partial \omega_{ij}(t)}) x_i - T_j(t) + \alpha(t) \frac{\partial E}{\partial T_j(t)}$$

Представим последнее выражение в следующем виде:

$$y_j' = \sum_i \omega_{ij}(t) x_i - T_j(t) - \alpha(t) \left(\sum_i \frac{\partial E}{\partial \omega_{ij}(t)} x_i - \frac{\partial E}{\partial T_j(t)} \right). \quad (2.30)$$

Обозначим:

$$a_j = \sum_i \frac{\partial E}{\partial \omega_{ij}(t)} x_i - \frac{\partial E}{\partial T_j(t)}. \quad (2.31)$$

Тогда с учетом выражения (2.31) выражение (2.30) можно записать, как

$$y_j' = y_j - \alpha(t) a_j. \quad (2.32)$$

Для определения адаптивного шага обучения необходимо найти такое значение $\alpha(t)$, чтобы среднеквадратичная ошибка была минимальной:

$$E = \frac{1}{2} \sum_j (y_j' - t_j)^2 = \frac{1}{2} \sum_j (y_j - \alpha a_j - t_j)^2 \rightarrow \min$$

Тогда

$$\frac{\partial E}{\partial \alpha} = \sum_j (\alpha a_j^2 - a_j (y_j - t_j)) = 0.$$

Отсюда

$$\alpha = \frac{\sum_j a_j (y_j - t_j)}{\sum_j a_j^2}. \quad (2.33)$$

Так как $\frac{\partial^2 E}{\partial^2 \alpha} > 0$, то при данном α обеспечивается минимум среднеквадратичной ошибки.

Найдем выражение для a_j . Так как

$$\frac{\partial E}{\partial \omega_j(t)} = (y_j - t_j)x_i,$$

$$\frac{\partial E}{\partial T_j} = -(y_j - t_j),$$

то

$$a_j = (y_j - t_j)(1 + \sum_i x_i^2). \quad (2.34)$$

Подставляя (2.34) в выражение (2.33), получим окончательное выражение для адаптивного шага обучения на каждой итерации алгоритма:

$$\alpha(t) = \frac{1}{1 + \sum_i x_i^2(t)} \quad (2.35)$$

В результате проведенных рассуждений доказана следующая теорема.

Теорема 2.1. Для линейной нейронной сети значение адаптивного шага обучения вычисляется на основе выражения 2.35.

Данная теорема справедлива, когда обучение происходит последовательно после подачи каждого образа на вход нейронной сети. Существует также второй вариант обучения, когда модификация синаптических связей производится только после подачи на вход сети нескольких образов. Назовем такой способ обучения групповым.

Теорема 2.2. Для линейной нейронной сети при групповом обучении величина адаптивного шага обучения определяется следующим образом:

$$\alpha = \frac{\sum_k \sum_j (y_j^k - t_j^k) a_j^k}{\sum_k \sum_j (a_j^k)^2} \quad (2.36)$$

$$a_j^k = \sum_p (y_j^p - t_j^p) \left(1 + \sum_i x_i^p x_i^k\right)$$

где $k = \overline{1, L}$; L — количество образов, подаваемых на вход сети при групповом обучении; x_i^k — i -ая компонента k -го образа.

Данная теорема доказывается аналогично предыдущей. При использовании группового обучения модификация синаптических связей будет происходить после подачи на вход сети L образов:

$$\omega_{ij}(L) = \omega_{ij}(0) - \alpha(t) \sum_{k=1}^L (y_j^k - t_j^k) \cdot x_i^k,$$

$$T_j(L) = T_j(0) + \alpha(t) \sum_{k=1}^L (y_j^k - t_j^k).$$

Эксперименты показали, что последовательный метод обучения является эффективнее группового.

Выражения (2.35) и (2.36) для выбора шага обучения нейронных сетей были получены в работе [27]. Адаптивный шаг обучения позволяет значительно повысить скорость обучения линейной нейронной сети и достичь оптимального решения задачи.

2.9. Использование псевдообратной матрицы для обучения линейных нейронных сетей

Для настройки весовых коэффициентов линейной нейронной сети с целью минимизации среднеквадратичной ошибки можно использовать матричное решение системы линейных уравнений. Уравнение линейной

нейронной сети в матричной форме можно представить в следующем виде:

$$Y = XW.$$

Матрицу выходных значений нейронной сети можно записать, как

$$Y = \begin{bmatrix} y_1^1 & y_1^2 & \dots & y_1^L \\ y_2^1 & y_2^2 & \dots & y_2^L \\ \dots & \dots & \dots & \dots \\ y_m^1 & y_m^2 & \dots & y_m^L \end{bmatrix},$$

где L — размерность обучающей выборки; m — количество выходных нейронов сети.

Аналогичным образом:

$$X = \begin{bmatrix} x_1^1 & x_2^1 & \dots & x_n^1 \\ x_1^2 & x_2^2 & \dots & x_n^2 \\ \dots & \dots & \dots & \dots \\ x_1^L & x_2^L & \dots & x_n^L \end{bmatrix},$$

$$W = \begin{bmatrix} \omega_{11} & \omega_{12} & \dots & \omega_{1m} \\ \omega_{21} & \omega_{22} & \dots & \omega_{2m} \\ \dots & \dots & \dots & \dots \\ \omega_{n1} & \omega_{n2} & \dots & \omega_{nm} \end{bmatrix}.$$

В случае использования порогов нейронных элементов в матрицу X добавляется строка, содержащая значения -1 , а в матрицу W — строка, в которой находятся пороговые значения нейронных элементов T .

В работе [28] показано, что наилучшим приближенным решением

системы линейных уравнений, при котором среднеквадратичная ошибка сети достигает своего наименьшего значения, является следующее выражение:

$$W = X^+ Y,$$

где X^+ — псевдообратная матрица для матрицы X .

Наилучшее приближенное решение является единственным [28] и минимизирует среднеквадратичную ошибку сети:

$$E = |Y - WX|^2.$$

Псевдообратная матрица является обобщением обратной матрицы на случай произвольной размерности и определяется следующим образом:

$$X^+ = (X^T X)^{-1} X^T.$$

Существуют различные способы нахождения псевдообратной матрицы [28], например, метод Гревилля, который не требует вычисления детерминантов.

При использовании псевдообратной матрицы возникают проблемы, когда матрица $(X^T X)$ является вырожденной. В случае большой размерности входных сигналов определение псевдообратной матрицы требует значительных вычислительных затрат. Поэтому метод градиентного спуска имеет преимущество по сравнению с рассмотренным методом, особенно при использовании адаптивного шага обучения.

2.10. Анализ линейных нейронных сетей

Рассмотрим многослойную нейронную сеть, каждый нейрон которой имеет линейную функцию активации. Такая нейронная сеть является сетью

с прямым распространением сигналов. Она характеризуется тем, что каждый нейронный элемент предыдущего слоя имеет синаптические связи со всеми нейронными элементами следующего слоя (рис. 2.22):

На рис. 2.22 изображены синаптические соединения только для одного нейрона каждого из слоев. Связи остальных нейронов будут идентичными. В работе [29] приводится следующая теорема.

Теорема 2.3. Многослойная нейронная сеть с линейной функцией активации нейронных элементов эквивалентна однослойной нейронной сети с соответствующей функцией активации нейронных элементов.

Доказательство. Выходные значения нейронной сети в матричной форме определяются следующим образом:

$$Y = X^{(2)}W^{(2)},$$

где $X^{(2)}$ — матрица выходных сигналов промежуточного слоя; $W^{(2)}$ — матрица весовых коэффициентов выходного слоя.

Аналогично, матрицу выходных значений промежуточного слоя можно представить, как

$$X^{(2)} = X^{(1)}W^{(1)}.$$

Отсюда получаем, что

$$Y = X^{(1)}W^{(1)}W^{(2)}.$$

Так как $W^{(1)}W^{(2)} = W$, то

$$Y = X^{(1)}W,$$

что и требовалось доказать.

Линейная нейронная сеть позволяет осуществить такое отображение входных сигналов в выходные, если между ними существует линейная

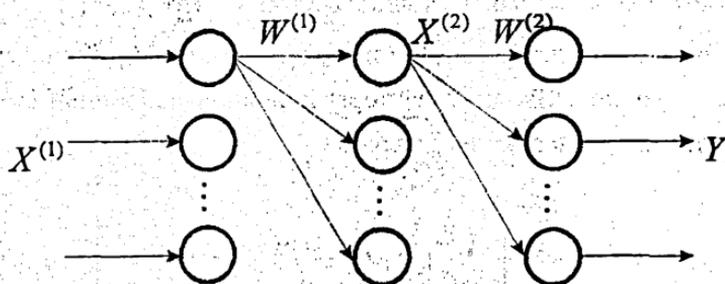


Рис. 2.22. Многослойная линейная сеть.

зависимость. Такая сеть может решать задачи сложения или вычитания десятичных чисел, задачи линейного авторегрессионного анализа или использоваться в качестве памяти. Способность нейронной сети выдавать корректные данные на примерах, которые не входили в обучающую выборку, называется *обобщающей способностью* сети. Так, например, достаточно обучить линейную нейронную сеть небольшому количеству операций сложения десятичных чисел, чтобы она корректно складывала любые десятичные числа. Применение линейных нейронных сетей будет подробно рассмотрено в следующих разделах.

2.11. Линейная ассоциативная память

Линейная ассоциативная память (linear associative memory) представляет собой двухслойную нейронную сеть, где в качестве выходных нейронов используются нейроны с линейной функцией активации (рис. 2.21.)[26]. Она была предложена в 1968 году Андерсоном (Anderson) и называется линейным ассоциатором. Линейный ассоциатор может использоваться в качестве гетероассоциативной памяти. В этом случае происходит отображение

входного вектора X в такой выходной вектор Y , что $X \neq Y$.

Линейный ассоциатор осуществляет линейное преобразование входного сигнала в выходной. Входные векторы линейной нейронной сети можно представить в виде следующей матрицы:

$$X = \begin{bmatrix} X^1 \\ X^2 \\ \vdots \\ X^L \end{bmatrix} = \begin{bmatrix} x_1^1 & x_2^1 & \dots & x_n^1 \\ x_1^2 & x_2^2 & \dots & x_n^2 \\ \dots & \dots & \dots & \dots \\ x_1^L & x_2^L & \dots & x_n^L \end{bmatrix},$$

где L — объем ассоциативной памяти; n — размерность входных образов.

Для функционирования линейной сети в качестве ассоциативной памяти необходимо, чтобы система входных векторов X^1, X^2, \dots, X^L была ортонормированной, т. е.

$$(X^i, X^j) = \sum_k x_k^i x_k^j = \begin{cases} 0, & \text{при } i \neq j; \\ 1, & \text{при } i = j, \end{cases} \quad (2.37)$$

где $i, j = 1, 2, \dots, n$.

Как следует из (2.37) в ортонормированной системе входные векторы являются ортогональными и нормированными, т. е.

$$|X^i| = 1, \forall i.$$

При этом нормированный вектор определяется следующим образом:

$$X^i = \left(\frac{x_1^i}{|X^i|}, \frac{x_2^i}{|X^i|}, \dots, \frac{x_n^i}{|X^i|} \right),$$

где $|X^i| = \sqrt{(x_1^i)^2 + (x_2^i)^2 + \dots + (x_n^i)^2}$ — длина вектора X^i . Обучение линейного ассоциатора производится по правилу Хебба.

В соответствии с ним можно записать следующие выражения для определения весовых коэффициентов нейронной сети:

$$W = \sum_{k=1}^L (X^k)^T Y^k = X^T Y,$$

$$\omega_{ij} = \sum_{k=1}^L x_i^k y_j^k,$$

где $Y^k = \{y_1^k, y_2^k, \dots, y_m^k\}$ — выходной вектор сети.

Рассмотрим функционирование линейной нейронной сети в качестве гетероассоциативной памяти. Пусть выходной слой сети состоит из одного линейного нейронного элемента. Тогда матрицу весовых коэффициентов можно представить следующим образом:

$$W = X^T Y = \sum_{k=1}^L (x^k)^T y^k.$$

При подаче на вход нейронной сети вектора X^i выходное значение сети определяется, как

$$Y^i = X^i W = \sum_{\substack{k=1 \\ k \neq i}}^L X^i (X^k)^T Y^k + X^i (X^i)^T Y^i. \quad (2.38)$$

Так как входные векторы являются ортонормированными, то

$$X^i(X^k)^T = 0 \quad \forall i \neq k,$$

$$X^i(X^i)^T = 1 \quad \forall i = k.$$

Тогда значение выражения (2.38) будет равняться

$$Y^i = Y^i,$$

т. е. линейная нейронная сеть выполняет функции гетероассоциативной памяти.

В случае использования биполярных и ортогональных, но не нормализованных векторов настройка весовых коэффициентов нейронной сети производится следующим образом:

$$w_{ij} = \frac{1}{n} \sum_{k=1}^L x_i^k x_j^k, \quad (2.39)$$

$$W = \frac{1}{n} X^T X,$$

где n -размерность входного вектора.

Тогда

$$Y^i = \frac{1}{n} X^i W = \frac{1}{n} \sum_{k=1}^L X^i (X^k)^T Y^k + \frac{1}{n} X^i (X^i)^T Y^L \quad (2.40)$$

В такой модели предполагается, что используются биполярные векторы,

т.е. $x_i \in [-1, +1]$. Тогда

$$X^i (X^i)^T = (X^i)^2 = n \quad (2.41).$$

Так как входные векторы являются ортогональными, то первый терм в выражении (2.40) равен нулю. Поэтому

$$Y^i = Y^i$$

Для произвольной системы входных векторов можно использовать псевдообращение матрицы для нахождения весовых коэффициентов сети. Тогда для гетероассоциативной памяти

$$W = X^+ Y.$$

В этом случае весовые коэффициенты нейронной сети настраиваются таким образом, чтобы минимизировать среднеквадратичное отклонение.

Для устойчивой работы линейной ассоциативной памяти необходимо, чтобы количество запоминаемых образов было меньше количества нейронов распределительного слоя. т. е.

$$L < n.$$

2.12. Нелинейная ассоциативная память

Использует нелинейную функцию активации нейронных элементов. В качестве такой функции обычно используется пороговая функция:

$$y_i = \begin{cases} 1, & \text{если } S_i > 0 \\ -1, & \text{если } S_i \leq 0 \end{cases}$$

Рассмотрим функционирование нелинейной нейронной сети в качестве автоассоциативной памяти. Обучение здесь производится по правилу Хебба в соответствии со следующим выражением :

$$W = X^T X.$$

Пусть нейронная сеть должна запомнить один образ:

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

Тогда матрица весовых коэффициентов сети определяется, как

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

При подаче на вход сети неискаженного паттерна X

$$y = F(XW) = F([4, 4, -4, 4]) = [1 \ 1 \ -1 \ 1].$$

Пусть теперь во входном векторе имеется один искаженный разряд:

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

Тогда

$$y = F(XW) = F([2 \ 2 \ -2 \ 2]) = [1 \ 1 \ -1 \ 1].$$

Таким образом нейронная сеть восстановила входной паттерн. Если сеть содержит n входных и n выходных нейронных элементов, то ее емкость

$$L \leq n - 1.$$

Для устойчивой работы такой памяти необходимо, чтобы входные векторы были ортогональными

2.13. Использование линейной нейронной сети для прогнозирования

Способность нейронных сетей после обучения к обобщению и пролонгации результатов создает потенциальные предпосылки для построения на базе их различного рода прогнозирующих систем. В данном разделе рассмотрим прогнозирование временных рядов при помощи линейных нейронных сетей. Пусть дан временной ряд $x(t)$ на промежутке $t = \overline{1, m}$. Тогда задача прогнозирования состоит в том, чтобы найти продолжение временного ряда на неизвестном промежутке, т. е. необходимо определить $x(m+1)$, $x(m+2)$ и так далее (рис. 2.23):

Совокупность известных значений временного ряда образуют обучающую выборку, размерность которой равняется m . Для прогнозирования временных рядов используется метод "скользящего окна". Он характеризуется длиной окна p , которая равняется количеству элементов ряда, одновременно подаваемых на нейронную сеть. Это определяет структуру нейронной сети, которая состоит из p распределительных нейронов и одного выходного нейрона.

Такая модель соответствует линейной авторегрессии и описывается следующим выражением:

$$\overline{x(n)} = \sum_{k=1}^p \omega_k x(n-p+k-1)$$

где ω_k , $k = \overline{1, p}$ — весовые коэффициенты нейронной сети; $\overline{x(n)}$ — оценка

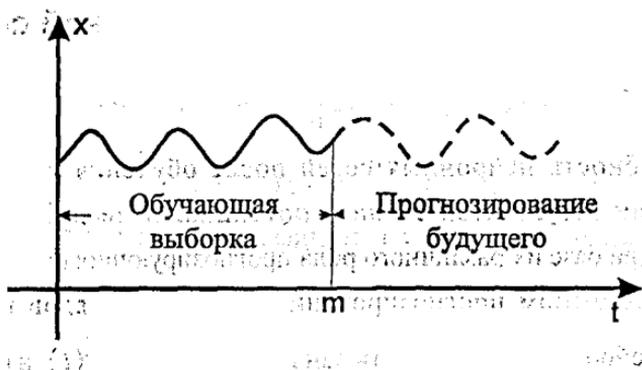


Рис. 2.23. Прогнозирование временного ряда.

значения ряда $x(n)$ в момент времени n .

Ошибка прогнозирования определяется, как

$$e(n) = x(n) - \overline{x(n)}$$

Модель линейной авторегрессии формирует значение ряда $x(n)$, как взвешенную сумму предыдущих значений ряда. Обучающую выборку нейронной сети можно представить в виде матрицы, строки которой характеризуют векторы, подаваемые на вход сети:

$$X = \begin{bmatrix} x(1) & x(2) & \dots & x(p) \\ x(2) & x(3) & \dots & x(p+1) \\ \dots & \dots & \dots & \dots \\ x(m-p) & x(m-p+1) & \dots & x(m-1) \end{bmatrix}$$

Это эквивалентно перемещению окна по ряду $x(t)$, с единичным

шагом.

Таким образом для обучения нейронной сети прогнозированию используется выборка известных членов ряда. После обучения сеть должна прогнозировать временной ряд на упреждающий промежуток времени. Рассмотрим прогнозирование следующей функции:

$$y = 0.4 \sin(0.3x) + 0.5$$

Пусть размер окна равен $p = 4$, а размерность обучающей выборки $L = 20$ для $x = 1, 20$. Тогда структура нейронной сети состоит из четырех входных и одного линейного нейрона. На рис. 2.24 приведены графики изменения суммарной среднеквадратичной ошибки сети в зависимости от количества итераций обучения. При обучении с постоянным шагом $\alpha = 0.01$ достигается суммарная среднеквадратичная ошибка $E = 9.98 \cdot 10^{-3}$ при количестве итераций равном 810. При обучении с адаптивным шагом суммарная среднеквадратичная ошибка $E = 9.98 \cdot 10^{-3}$ достигается при количестве итераций равном 303. Таким образом скорость обучения с адаптивным шагом приблизительно в 2.7 раза больше чем при использовании постоянного шага. В таблицах 2.3 и 2.4 приведены характеристики прогнозирующих свойств линейной нейронной сети, которая обучалась с постоянным и адаптивным шагом. При этом прогнозирование осуществлялось за пределами обучающей выборки на 20 шагов вперед.

Таблица 2.3

Эталонное	Полученное	Отклонение	Квадрат отклонения
5,9867E-01	5,6086E-01	3,7809E-02	0,0014295
6,4846E-01	6,0577E-01	4,2688E-02	0,0018222
6,9676E-01	6,5041E-01	4,6351E-02	0,0021485
7,4195E-01	6,9326E-01	4,8684E-02	0,0023701
7,8250E-01	7,3288E-01	4,9620E-02	0,0024622
8,1707E-01	7,6792E-01	4,9150E-02	0,0024157
8,4456E-01	7,9725E-01	4,7312E-02	0,0022384
8,6410E-01	8,1990E-01	4,4200E-02	0,0019536
8,7516E-01	8,3521E-01	3,9951E-02	0,0015961
8,7749E-01	8,4274E-01	3,4748E-02	0,0012074
8,7118E-01	8,4238E-01	2,8807E-02	0,0008298
8,5666E-01	8,3429E-01	2,2367E-02	0,0005003
8,3463E-01	8,1895E-01	1,5685E-02	0,000246
8,0609E-01	7,9706E-01	9,0249E-03	8,145E-05
7,7224E-01	7,6960E-01	2,6427E-03	6,984E-06
7,3450E-01	7,3771E-01	-3,2182E-03	1,036E-05
6,9437E-01	7,0271E-01	-8,3390E-03	6,954E-05
6,5347E-01	6,6600E-01	-1,2535E-02	0,0001571
6,1337E-01	6,2903E-01	-1,5659E-02	0,0002452
5,7561E-01	5,9322E-01	-1,7615E-02	0,0003103

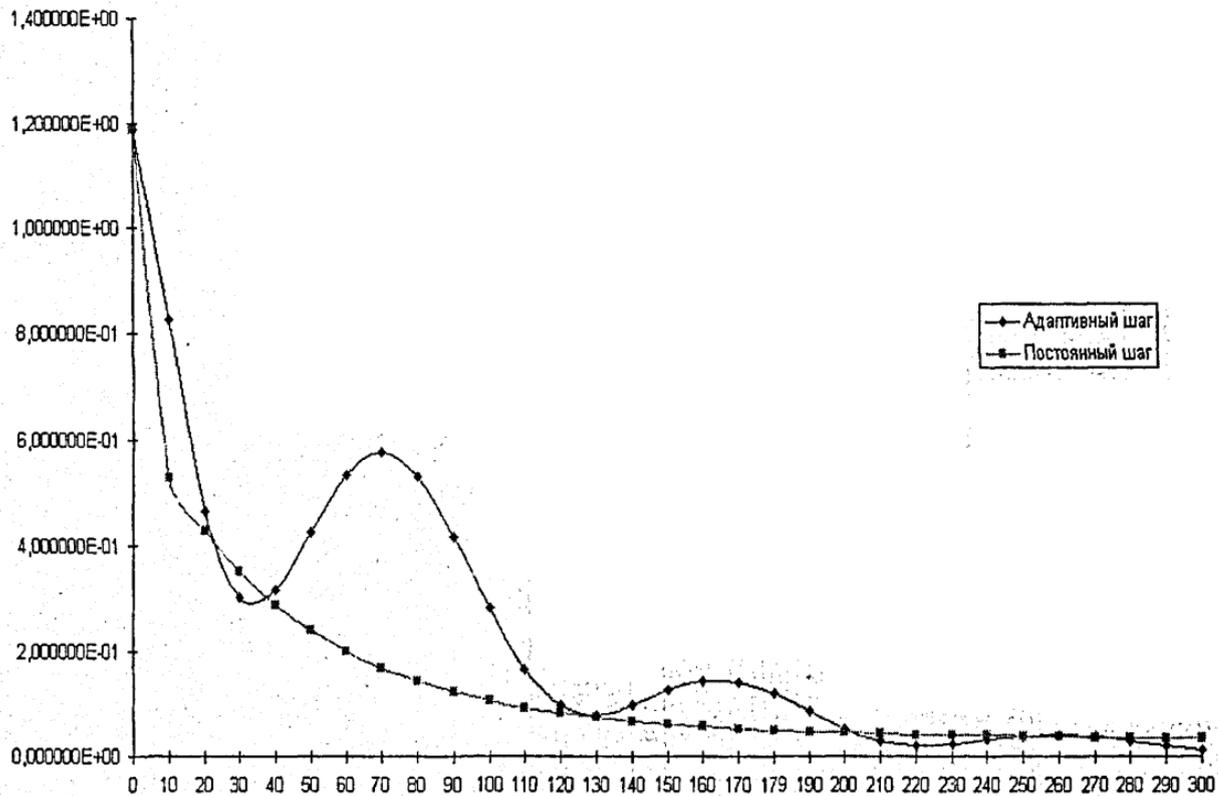


Рис. 2.24. Изменение среднеквадратичной ошибки для постоянного и адаптивного шага

Таблица 2.4

Эталонное	Полученное	Отклонение	Квадрат отклонения
5,9867E-01	5,9867E-01	1,9984E-15	3,9936E-30
6,4846E-01	6,4732E-01	1,1401E-03	1,2999E-06
6,9676E-01	6,9359E-01	3,1689E-03	1,0042E-05
7,4195E-01	7,3591E-01	6,0380E-03	3,6457E-05
7,8250E-01	7,7283E-01	9,6671E-03	9,3452E-05
8,1707E-01	8,0312E-01	1,3950E-02	1,9459E-04
8,4456E-01	8,2581E-01	1,8748E-02	3,5149E-04
8,6410E-01	8,4019E-01	2,3911E-02	5,7174E-04
8,7516E-01	8,4589E-01	2,9266E-02	8,5650E-04
8,7749E-01	8,4285E-01	3,4637E-02	1,1997E-03
8,7118E-01	8,3134E-01	3,9846E-02	1,5877E-03
8,5666E-01	8,1194E-01	4,4718E-02	1,9997E-03
8,3463E-01	7,8554E-01	4,9091E-02	2,4099E-03
8,0609E-01	7,5327E-01	5,2821E-02	2,7901E-03
7,7224E-01	7,1645E-01	5,5787E-02	3,1122E-03
7,3450E-01	6,7660E-01	5,7897E-02	3,3521E-03
6,9437E-01	6,3528E-01	5,9090E-02	3,4916E-03
6,5347E-01	5,9413E-01	5,9338E-02	3,5210E-03
6,1337E-01	5,5472E-01	5,8650E-02	3,4398E-03
5,7561E-01	5,1854E-01	5,7063E-02	3,2562E-03

ГЛАВА 3. МНОГОСЛОЙНЫЕ НЕЙРОННЫЕ СЕТИ

В данной главе рассматриваются многослойные нейронные сети с прямым распространением сигналов (multi layer feed forward networks). *Многослойная нейронная сеть* способна осуществлять любое отображение входных векторов в выходные. Это свойство многослойных нейронных сетей отмечали еще в 60-х годах Минский и Пайперт. Однако на тот момент времени не было эффективного алгоритма обучения таких сетей. Поэтому их выводы на счет перспектив развития многослойных нейронных сетей были весьма пессимистичными. В 1986 рядом авторов (Rumelhart, Hinton, Williams) независимо друг от друга был предложен *алгоритм обратного распространения ошибки* (backpropagation algorithm), который стал эффективным средством обучения многослойных нейронных сетей. В главе приводится алгоритм обратного распространения ошибки и его модификации, которые были предложены в [27, 30-32]. Рассматривается применение многослойных нейронных сетей для решения различного рода задач.

3.1. Топология и анализ многослойных нейронных сетей

Архитектура многослойной нейронной сети состоит из множества слоев нейронных элементов (рис. 3.1.).

Входной слой (input layer) нейронных элементов выполняет распределительные функции. *Выходной слой* (output layer) нейронов служит для обработки информации от предыдущих слоев и выдачи результатов. Слои нейронных элементов, расположенные между входным и выходным слоем,

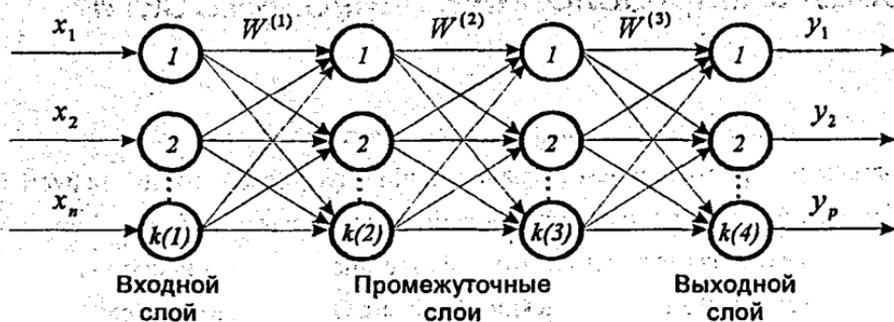


Рис 3.1. Многослойная нейронная сеть

называются *промежуточными* или *скрытыми* (hidden layers). Как и выходной слой, скрытые слои являются обрабатывающими. Выход каждого нейронного элемента предыдущего слоя нейронной сети соединен синаптическими связями со всеми входами нейронных элементов следующего слоя. Таким образом, топология многослойной нейронной сети является однородной и регулярной.

В качестве функции активации нейронных элементов здесь используются обычно гиперболический тангенс или сигмоидная функция.

Пусть $W^{(i)}$ — матрица весовых коэффициентов i -го слоя многослойной сети. Тогда для нейронной сети с двумя скрытыми слоями выходные значения Y можно определить следующим образом:

$$Y = F(F(F(XW^{(1)}W^{(2)}W^{(3)})), \dots) \quad (3.1)$$

где $X = (x_1, x_2, \dots, x_n)$ — вектор-строка входных сигналов; F — оператор нелинейного преобразования.

Общее количество синаптических связей многослойной нейронной сети равняется:

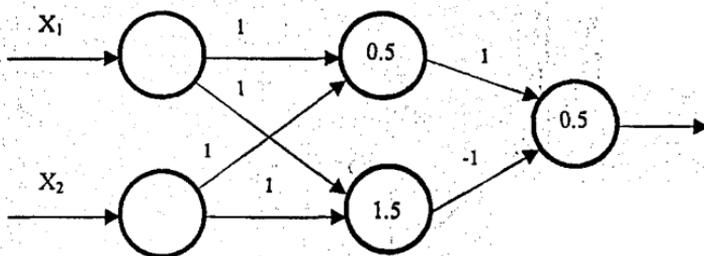


Рис. 3.2. Нейронная сеть для решения задачи «ИСКЛЮЧАЮЩЕЕ ИЛИ».

$$V = \sum_{i=1}^{p-1} k(i) \cdot k(i+1), \quad (3.2)$$

где p — общее количество слоев нейронной сети; $k(i)$ — количество нейронных элементов в i -ом слое.

Количество слоев в многослойной нейронной сети характеризует, каким образом входное пространство может быть разбито на подпространства меньшей размерности. Так, двухслойная нейронная сеть с одним слоем нелинейных нейронов разбивает входное пространство образов на классы при помощи *гиперплоскости* [33]. Трехслойная нейронная сеть, где в качестве двух последних слоев используются нейронные элементы с нелинейной функцией активации, позволяет формировать любые выпуклые области в пространстве решений [33, 34]. При помощи четырехслойной нейронной сети, которая имеет три нелинейных слоя, можно получать область решений любой формы и сложности, в том числе и невыпуклой.

Рассмотрим решение задачи «ИСКЛЮЧАЮЩЕЕ ИЛИ» при помощи трехслойной нейронной сети (рис. 3.2). На рис. 3.2 показаны весовые коэффициенты и пороговые значения нейронных элементов. Слой скрытых нейронных элементов производит разбиение входного пространства образов на классы при помощи двух прямых рис 3.3. Выходной нейронный элемент

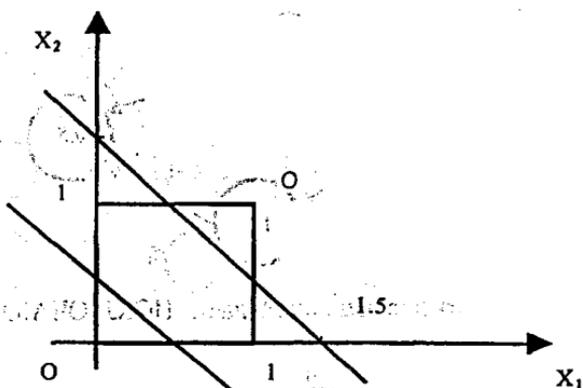


Рис. 3.3. Графическая иллюстрация решения задачи «ИСКЛЮЧАЮЩЕ ИЛИ».

осуществляет композицию классов для реализации функции «ИСКЛЮЧАЮЩЕ ИЛИ».

В 1957 Колмогоров показал [35], что любую непрерывную функцию n переменных на единичном отрезке $[0,1]$ можно представить в виде суммы конечного числа одномерных функций:

$$f(x_1, x_2, \dots, x_n) = \sum_{p=0}^{2n+1} g\left(\sum_{i=1}^n \lambda_i \varphi_p(x_i)\right), \quad (3.3)$$

где функции g и φ_p являются одномерными и непрерывными; $\lambda_i = const$ для всех i .

Данная теорема легла в основу построения многослойных нейронных сетей для аппроксимации функций. Из нее следует, что любую непрерывную функцию $f: [0,1]^n \rightarrow [0,1]$ можно аппроксимировать при помощи

трехслойной нейронной сети, которая имеет n входных, $(2n + 1)$ скрытых и один выходной нейрон. Однако проблема здесь состоит в выборе соответствующих функций g и φ . В 1988 г. ряд авторов [36, 38] обобщили приведенные выше результаты на многослойную нейронную сеть с алгоритмом обратного распространения ошибки.

Теорема 3.1. Любая непрерывная функция $f: [0,1]^n \rightarrow [0,1]$ может быть аппроксимирована трехслойной нейронной сетью с одним скрытым слоем и алгоритмом обратного распространения ошибки с любой степенью точности.

Данная теорема является основной для аппроксимации функций при помощи многослойных нейронных сетей.

3.2. Нейронные сети высокого порядка

Нейронные сети *высокого порядка* (higher order networks) позволяют так же, как и многослойные нейронные сети разбивать входное пространство на классы при помощи нелинейных разделяющих поверхностей [39]. Взвешенная сумма нейронного элемента многослойной нейронной сети содержит только термины первого порядка. В отличие от этого, взвешенная сумма нейронного элемента для сети высокого порядка содержит произведения двух или более компонент входного вектора X . Так для нейронной сети второго порядка

$$S = \sum_{i=1}^n w_i x_i + \sum_{i=1}^n \sum_{j=1}^n w_{ij} x_i x_j - T$$

Разделяющая поверхность, определяемая уравнением $S=0$ является

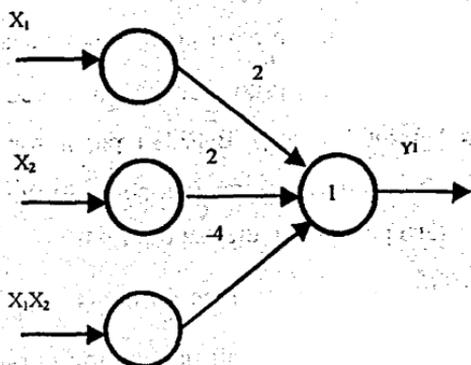


Рис. 3.4. Нейронная сеть второго порядка.

поверхностью второго порядка и называется *гиперквадрикой* [26].

Если продолжить вводить дополнительные члены в произведение компонент вектора X , например такие как $w_{ij}x_i x_j$, то получится класс полиномиальных разделяющих поверхностей. Рассмотрим решение задачи «ИСКЛЮЧАЮЩЕЕ ИЛИ» с помощью нейронной сети второго порядка. Тогда взвешенная сумма

$$S = w_1x_1 + w_2x_2 + w_{12}x_1x_2 - T$$

На рис. 3.4 изображена структура такой сети для решения задачи «ИСКЛЮЧАЮЩЕЕ ИЛИ».

В качестве функции активации выходного нейрона используется пороговая функция. С учетом определенных на рис. 3.4. весовых коэффициентов и порогов нейронной сети взвешенную сумму можно представить, как

$$S = 2x_1 + 2x_2 - 4x_1x_2 - 1$$

Для обучения нейронных сетей высокого порядка может использоваться алгоритм обратного распространения ошибки [40]. Однако более

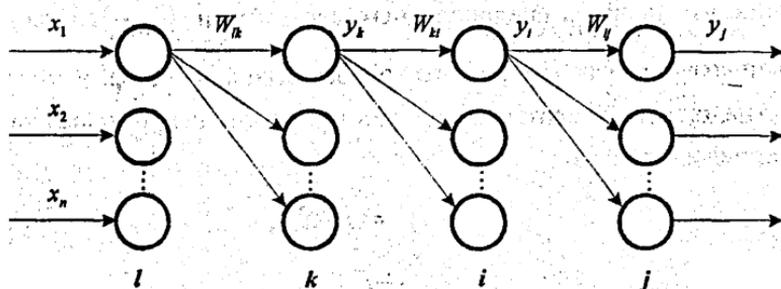


Рис. 3.5. Четырехслойная нейронная сеть.

эффективным является применение многослойных нейронных сетей.

3.3. Математические основы алгоритма обратного распространения ошибки.

Алгоритм обратного распространения ошибки был предложен в [25] и является эффективным средством для обучения многослойных нейронных сетей.

Рассмотрим нейронную сеть, состоящую из четырех слоев (рис.3.5). Обозначим слои нейронных элементов от входа к выходу соответственно через l, k, i, j . Тогда выходное значение j -го нейрона последнего слоя равняется:

$$y_j = F(S_j), \quad (3.4)$$

$$S_j = \sum_i \omega_{ij} y_i - T_j, \quad (3.5)$$

где S_j — взвешенная сумма j -го нейрона выходного слоя; y_i — выходное

значение i -го нейрона предпоследнего слоя; ω_{ij} и T_j — соответственно весовой коэффициент и порог j -го нейрона выходного слоя.

Аналогичным образом выходное значение i -го нейрона предпоследнего слоя определяется, как:

$$y_i = F(S_i), \quad (3.6)$$

$$S_i = \sum_k \omega_{ki} y_k - T_i. \quad (3.7)$$

Соответственно для k -го слоя:

$$y_k = F(S_k) \quad (3.8)$$

$$S_k = \sum_l \omega_{lk} y_l - T_k. \quad (3.9)$$

Алгоритм обратного распространения ошибки минимизирует среднеквадратичную ошибку нейронной сети. Для этого с целью настройки синаптических связей используется метод градиентного спуска в пространстве весовых коэффициентов и порогов нейронной сети. Согласно методу градиентного спуска изменение весовых коэффициентов и порогов нейронной сети происходит по следующему правилу:

$$\omega_{ij}(t+1) = \omega_{ij}(t) - \alpha \frac{\partial E}{\partial \omega_{ij}(t)}, \quad (3.10)$$

$$T_j(t+1) = T_j(t) - \alpha \frac{\partial E}{\partial T_j(t)}, \quad (3.11)$$

где E — среднеквадратичная ошибка нейронной сети для одного образа.

Она определяется, как

$$E = \frac{1}{2} \sum_i (y_i - t_i)^2, \quad (3.12)$$

где t_j — эталонное выходное значение j -го нейрона.

Ошибка j -го нейрона выходного слоя равняется:

$$\gamma_j = y_j - t_j. \quad (3.13)$$

Теорема 3.2. Для любого скрытого слоя i ошибка i -го нейронного элемента определяется рекурсивным образом через ошибки нейронов следующего слоя j :

$$\gamma_i = \sum_{j=1}^m \gamma_j F'(S_j) \omega_{ij}, \quad (3.14)$$

где m — количество нейронов следующего слоя по отношению к слою i ;

ω_{ij} — синаптическая связь между i -м и j -м нейроном различных слоев;

S_j — взвешенная сумма j -го нейрона.

Доказательство. Ошибка i -го нейронного элемента равняется

$$\gamma_i = \frac{\partial E}{\partial y_i},$$

где y_i — выходное значение i -го нейрона.

Ошибка γ_i зависит от ошибки нейронных элементов следующего слоя j и определяется, как

$$\gamma_i = \frac{\partial E}{\partial y_i} = \sum_j \frac{\partial E}{\partial y_j} \cdot \frac{\partial y_j}{\partial S_j} \cdot \frac{\partial S_j}{\partial y_i}$$

Так как

$$\frac{\partial E}{\partial y_j} = \gamma_j, \quad \frac{\partial y_j}{\partial S_j} = F'(S_j), \quad \frac{\partial S_j}{\partial y_i} = \omega_{ij},$$

то

$$\gamma_i = \sum_j \gamma_j F'(S_j) \omega_{ij},$$

что и требовалось доказать.

Используя результаты данной теоремы можно определить ошибки нейронов скрытого слоя через ошибки нейронов следующего слоя по отношению к скрытому слою.

Теорема 3.3. Производные среднеквадратичной ошибки по весовым коэффициентам и порогам нейронных элементов для любых двух слоев j и j многослойной сети определяются следующим образом:

$$\frac{\partial E}{\partial \omega_{ij}} = \gamma_j F'(S_j) y_i, \quad (3.15)$$

$$\frac{\partial E}{\partial T_j} = -\gamma_j F'(S_j). \quad (3.16)$$

Глава 3. Многослойные нейронные сети

Доказательство. Определим производные среднеквадратичной ошибки для различных слоев нейронной сети (рис. 3.5). Тогда для выходного слоя

$$\frac{\partial E}{\partial \omega_{ij}} = \frac{\partial E}{\partial y_j} \cdot \frac{\partial y_j}{\partial S_j} \cdot \frac{\partial S_j}{\partial \omega_{ij}},$$

$$\frac{\partial E}{\partial T_j} = \frac{\partial E}{\partial y_j} \cdot \frac{\partial y_j}{\partial S_j} \cdot \frac{\partial S_j}{\partial T_j}.$$

Так как

$$\frac{\partial E}{\partial y_j} = \gamma_j = y_j - t_j, \quad \frac{\partial y_j}{\partial S_j} = F'(S_j), \quad \frac{\partial S_j}{\partial \omega_{ij}} = y_i, \quad \frac{\partial S_j}{\partial T_j} = -1,$$

то

$$\frac{\partial E}{\partial \omega_{ij}} = (y_j - t_j) F'(S_j) y_i,$$

$$\frac{\partial E}{\partial T_j} = -(y_j - t_j) F'(S_j).$$

Таким образом, для выходного слоя теорема доказана. Рассмотрим теперь скрытый слой. Для него производные среднеквадратичной ошибки по весовым коэффициентам и порогам определяются следующим образом:

$$\frac{\partial E}{\partial \omega_{ki}} = \sum_j \frac{\partial E}{\partial y_j} \cdot \frac{\partial y_j}{\partial S_j} \cdot \frac{\partial S_j}{\partial y_i} \cdot \frac{\partial y_i}{\partial S_i} \cdot \frac{\partial S_i}{\partial \omega_{ki}},$$

$$\frac{\partial E}{\partial T_i} = \sum_j \frac{\partial E}{\partial y_j} \cdot \frac{\partial y_j}{\partial S_j} \cdot \frac{\partial S_j}{\partial y_i} \cdot \frac{\partial y_i}{\partial S_i} \cdot \frac{\partial S_i}{\partial T_i}$$

Так как

$$\gamma_i = \frac{\partial E}{\partial y_i} = \sum_j \frac{\partial E}{\partial y_j} \cdot \frac{\partial y_j}{\partial S_j} \cdot \frac{\partial S_j}{\partial y_i} = \sum_j \gamma_j F'(S_j) \omega_{ij},$$

то

$$\frac{\partial E}{\partial \omega_{ki}} = \gamma_i \frac{\partial y_i}{\partial S_i} \cdot \frac{\partial S_i}{\partial \omega_{ki}}, \quad (3.17)$$

$$(3.18)$$

$$\frac{\partial E}{\partial T_i} = \gamma_i \frac{\partial y_i}{\partial S_i} \cdot \frac{\partial S_i}{\partial T_i}$$

Определим оставшиеся производные:

$$\frac{\partial y_i}{\partial S_i} = F'(S_i), \quad (3.19)$$

$$\frac{\partial S_i}{\partial \omega_{ki}} = y_k, \quad (3.20)$$

$$\frac{\partial S_i}{\partial T_i} = -1. \quad (3.21)$$

Подставляя (3.19), (3.20) и (3.21) в выражения (3.17) и (3.18), получим:

$$\frac{\partial E}{\partial \omega_{ki}} = \gamma_i F'(S_i) y_k,$$

$$\frac{\partial E}{\partial T_i} = -\gamma_i F'(S_i),$$

что и требовалось доказать.

Следствие 3.1: Для минимизации среднеквадратичной ошибки сети весовые коэффициенты и пороги нейронных элементов должны изменяться с течением времени следующим образом:

$$\omega_{ij}(t+1) = \omega_{ij}(t) - \alpha \gamma_j F'(S_j) y_i, \quad (3.22)$$

$$T_j(t+1) = T_j(t) + \alpha \gamma_j F'(S_j), \quad (3.23)$$

где α — скорость обучения.

Данное следствие является очевидным. Оно определяет правило обучения многослойных нейронных сетей в общем виде, которое называется *обобщенным дельта правилом*.

3.4. Обобщенное дельта правило для различных функций активации нейронных элементов

Определим выражения (3.22) и (3.23) для различных функций активации нейронных элементов.

3.4.1. Сигмоидная функция

Выходное значение j -го нейронного элемента определяется следующим

образом:

$$y_j = \frac{1}{1 + e^{-S_j}}, \quad (3.24)$$

$$S_j = \sum_i \omega_{ij} y_i - T_j. \quad (3.25)$$

Тогда

$$F'(S_j) = \frac{\partial y_j}{\partial S_j} = y_j(1 - y_j) \quad (3.26)$$

В результате обобщенное дельта правило для сигмоидной функции активации можно представить в следующем виде:

$$\omega_{ij}(t+1) = \omega_{ij}(t) - \alpha \gamma_j y_j(1 - y_j) y_i, \quad (3.27)$$

$$T_j(t+1) = T_j + \alpha \gamma_j y_j(1 - y_j). \quad (3.28)$$

Ошибка для j -го нейрона выходного слоя определяется, как

$$\gamma_j = y_j - t_j. \quad (3.29)$$

Для j -го нейронного элемента скрытого слоя:

$$\gamma_j = \sum_{i=1}^m \gamma_i y_i(1 - y_i) \omega_{ji}, \quad (3.30)$$

где m — количество нейронных элементов следующего слоя по отношению к слою j (рис. 3.6).

3.4.2. Биполярная сигмоидная функция

Выходное значение j -го нейрона определяется, как

Подставляя (3.19), (3.20) и (3.21) в выражения (3.17) и (3.18), получим:

$$\frac{\partial E}{\partial \omega_{ki}} = \gamma_i F'(S_i) y_k,$$

$$\frac{\partial E}{\partial T_i} = -\gamma_i F'(S_i),$$

что и требовалось доказать.

Следствие 3.1: Для минимизации среднеквадратичной ошибки сети весовые коэффициенты и пороги нейронных элементов должны изменяться с течением времени следующим образом:

$$\omega_{ij}(t+1) = \omega_{ij}(t) - \alpha \gamma_j F'(S_j) y_i, \quad (3.22)$$

$$T_j(t+1) = T_j(t) + \alpha \gamma_j F'(S_j), \quad (3.23)$$

где α — скорость обучения.

Данное следствие является очевидным. Оно определяет правило обучения многослойных нейронных сетей в общем виде, которое называется *обобщенным дельта правилом*.

3.4. Обобщенное дельта правило для различных функций активации нейронных элементов

Определим выражения (3.22) и (3.23) для различных функций активации нейронных элементов.

3.4.1. Сигмоидная функция

Выходное значение j -го нейронного элемента определяется следующим

3.4.3. Гиперболический тангенс

Для данной функции активации выходное значение j -го нейрона определяется следующим образом:

$$y_j = th(S_j) = \frac{e^{S_j} - e^{-S_j}}{e^{S_j} + e^{-S_j}}. \quad (3.37)$$

Определим производную функции гиперболический тангенс:

$$F'(S_j) = \frac{\partial y_j}{\partial S_j} = \frac{1}{ch^2(S_j)} = (1 - y_j^2). \quad (3.38)$$

Тогда правило обучения можно представить в виде следующих выражений:

$$\omega_{ij}(t+1) = \omega_{ij}(t) - \alpha \gamma_j (1 - y_j^2) y_i, \quad (3.39)$$

$$T_j(t+1) = T_j(t) + \alpha \gamma_j (1 - y_j^2). \quad (3.40)$$

Ошибка для j -го нейрона выходного и скрытого слоев соответственно равняется:

$$\gamma_j = y_j - t_j, \quad (3.41)$$

$$\gamma_j = \sum_i \gamma_i (1 - y_i^2) \omega_{ji}. \quad (3.42)$$

Используя полученные в данном разделе выражения можно определить алгоритм обратного распространения ошибки для различных функций активации нейронных элементов.

3.5. Алгоритм обратного распространения ошибки

Как уже отмечалось, алгоритм обратного распространения ошибки был предложен в 1986 г. рядом авторов независимо друг от друга. Он является эффективным средством обучения нейронных сетей и представляет собой следующую последовательность шагов:

1. Задается шаг обучения α ($0 < \alpha < 1$) и желаемая среднеквадратичная ошибка нейронной сети E_m .
2. Случайным образом инициализируются весовые коэффициенты и пороговые значения нейронной сети.
3. Последовательно подаются образы из обучающей выборки на вход нейронной сети. При этом для каждого входного образа выполняются следующие действия:
 - a) производится фаза прямого распространения входного образа по нейронной сети. При этом вычисляется выходная активность всех нейронных элементов сети:

$$y_j = F\left(\sum_i \omega_{ij} y_i - T_j\right)$$

где индекс j характеризует нейроны следующего слоя по отношению к слою i .

- b) производится фаза обратного распространения сигнала, в результате которой определяется ошибка γ_j , $j = 1, 2, \dots$ нейронных элементов для всех слоев сети. При этом соответственно для выходного и скрытого слоев:

$$\gamma_j = y_j - t_j,$$

$$\gamma_j = \sum_i \gamma_i F'(S_i) \omega_{ji}.$$

В последнем выражении индекс i характеризует нейронные элементы следующего слоя по отношению к слою j .

- с) для каждого слоя нейронной сети происходит изменение весовых коэффициентов и порогов нейронных элементов:

$$\omega_{ij}(t+1) = \omega_{ij}(t) - \alpha \gamma_j F'(S_j) y_i,$$

$$T_j(t+1) = T_j(t) + \alpha \gamma_j F'(S_j).$$

4. Вычисляется суммарная среднеквадратичная ошибка нейронной сети:

$$E = \frac{1}{2} \sum_{k=1}^L \sum_i (y_j^k - t_j^k)^2,$$

где L — размерность обучающей выборки.

5. Если $E > E_m$ то происходит переход к шагу 3 алгоритма. В противном случае алгоритм обратного распространения ошибки заканчивается.

Таким образом, данный алгоритм функционирует до тех пор, пока суммарная среднеквадратичная ошибка сети не станет меньше заданной, т. е. $E \leq E_m$.

В 1990 было показано [37], что проблема обучения многослойных нейронных сетей, в которых все весовые коэффициенты неизвестны, является NP-полной. Это означает, что не существует алгоритма, который за полиномиальное время может решить задачу модификации синаптических

связей. Под полиномиальным временем понимается такое время, которое растет с размером сети не быстрее полинома конечной степени. Отсюда следует, что с возрастанием размерности нейронной сети, проблема ее обучения становится все более актуальной.

3.6. Недостатки алгоритма обратного распространения ошибки

Алгоритм обратного распространения ошибки, в основе которого лежит градиентный метод, создает ряд проблем при обучении многослойных нейронных сетей. К таким проблемам можно отнести следующие:

1. Неизвестен выбор количества слоев и количества нейронных элементов в слое для многослойных сетей.
2. Медленная сходимости градиентного метода с постоянным шагом обучения.
3. Проблема выбора подходящей скорости обучения α . Так, слишком малая скорость обучения увеличивает время обучения и приводит к скатыванию нейронной сети в локальный минимум. Большая скорость обучения может привести к пропуску глобального минимума и сделать процесс обучения расходящимся.
4. Градиентный метод не различает точек локального и глобального минимумов (рис. 3.7).
5. Влияние случайной инициализации весовых коэффициентов нейронной сети на поиск минимума функции среднеквадратичной ошибки.

Последний пункт отражает, что при разной инициализации синаптических связей, могут получаться различные решения задачи. Это характеризует неустойчивость алгоритма обучения, когда нейронная сеть в

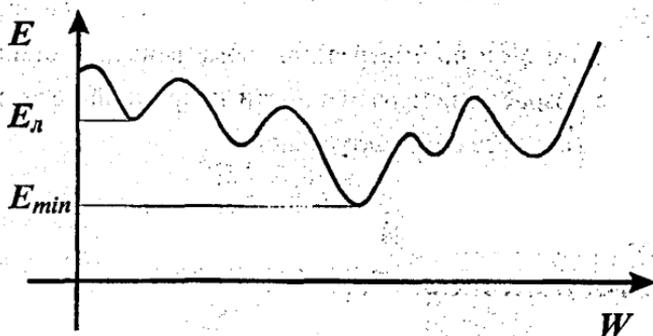


Рис. 3.7. Функция среднеквадратичной ошибки: E_n - локальный минимум, E_{min} - глобальный минимум.

одних случаях может обучаться до требуемой суммарной среднеквадратичной ошибки, а в других нет. То, что алгоритм обратного распространения ошибки не позволяет в общем случае достичь глобального минимума не умаляет его достоинств. Это связано с тем, что во многих практических задачах достаточно обучить нейронную сеть до требуемой среднеквадратичной ошибки. Является ли при этом найденный минимум локальным или глобальным не имеет большого значения.

В последующих разделах данной главы будут рассмотрены модификации алгоритма обратного распространения ошибки с целью нейтрализации приведенных выше недостатков.

3.7. Рекомендации по обучению и архитектуре многослойных нейронных сетей.

Эффективность обучения многослойных нейронных сетей зависит от количества слоев, числа элементов в скрытых слоях нейронной сети и начальной инициализации весовых коэффициентов. Рассмотрим эти вопросы

более подробно.

Как отмечалось в предыдущем разделе, разная инициализация весовых коэффициентов нейронной сети может приводить к различным решениям задачи. Важную роль здесь играет размер случайно инициализируемых синаптических связей [41, 42]. Так для сигмоидной функции активации нейронных элементов, если весовые коэффициенты будут иметь большие значения (положительные или отрицательные), то выходная активность нейронных элементов будет близка к единице или нулю. Тогда значение выражения $y_j(1-y_j)$ будет близко к нулю и согласно правилу обучения (3.27) весовые коэффициенты будут изменяться незначительно. Это приведет к тому, что процесс обучения остановится в ближайшем локальном минимуме от стартовой точки. В работе [41] рекомендуется случайно выбирать значения весовых коэффициентов, которые имеют следующий порядок :

$$\omega_{ij} \approx \frac{1}{\sqrt{n(i)}}$$

где $n(i)$ – количество нейронных элементов в слое i .

Другие авторы рекомендуют случайным образом инициализировать весовые коэффициенты в диапазоне $[-0.05, 0.05]$ или $[-0.1, 0.1]$. При этом пороговые значения нейронных элементов устанавливаются в начальный момент времени в единичные значения.

Большую роль для эффективности обучения играет архитектура нейронной сети. Размерность входного и выходного слоев нейронной сети определяется из условия решаемой задачи или обучающей выборки. Как было отмечено в разделе 3.1, при помощи трехслойной нейронной сети можно аппроксимировать любую функцию со сколь угодно заданной точностью. При этом точность аппроксимации зависит от количества нейронов в скрытом

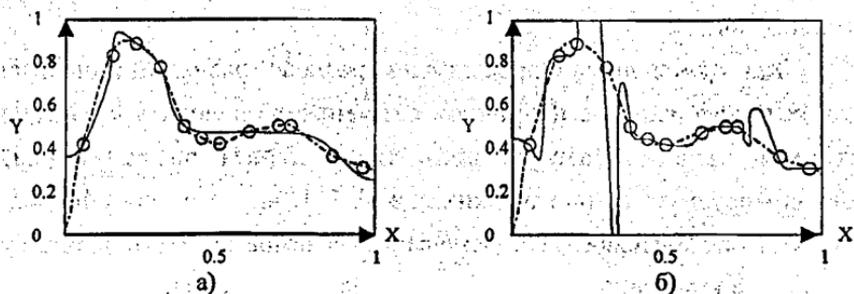


Рис. 3.8. Обучение нейронной сети: а - сеть имеет 5 скрытых нейронов, б - количество скрытых нейронов равняется 20.

слое. Чем больше количество нейронных элементов в скрытом слое, тем больше точность. Однако при слишком большой размерности скрытого слоя может наступить явление, которое называется перетренировкой сети. На рис. 3.8 изображены примеры обучения нейронной сети для аппроксимации функции. Рисунок 3.8 а характеризует нейронную сеть с 5 скрытыми, а рис. 3.8 б - с 20 скрытыми нейронными элементами. Объем обучающей выборки при этом равняется 12. Как следует из рисунков, нейронная сеть с 5 скрытыми нейронами достаточно точно аппроксимирует функцию на тех данных, которым ее не учили (сплошная линия). Нейронная сеть с 20 скрытыми нейронами хорошо аппроксимирует функцию только на тренировочных наборах (на рис. 3.8 они представлены кружками). Отсюда следует, что слишком большое количество нейронов в скрытом слое ухудшает обобщающие способности нейронных сетей. Поэтому количество нейронных элементов в скрытом слое должно быть меньше количества тренировочных образцов. С другой стороны, при слишком малой размерности скрытого слоя можно попасть в нежелательный локальный минимум или процесс обучения будет слишком длительным. Поэтому здесь необходим разумный компромисс.

Для обеспечения требуемой точности и обобщающей способности можно использовать нейронную сеть с двумя скрытыми слоями, размерность которых меньше, чем при использовании трехслойной сети. Однако нейронные сети, которые имеют несколько скрытых слоев, обучаются значительно медленнее.

Как отмечалось ранее, одним из недостатков метода градиентного спуска является застревание его в нежелательных локальных минимумах. Для нейтрализации этого недостатка можно использовать метод тяжелого шарика [41, 43]. В этом случае модификация синаптических связей нейронной сети происходит в соответствии со следующим выражением.:

$$\Delta\omega_{ij}(t+1) = -\alpha\gamma_j F'(s_j)y_i + \dot{\gamma}\Delta\omega_{ij}(t) \quad (3.43)$$

где γ – величина постоянная, которая называется моментным параметром.

Значение моментного параметра выбирается из диапазона [0..1]. На практике часто используется значение $\gamma = 0.9$ [41].

Последнее слагаемое в выражении (3.43) называется моментным термом (momentum term). Введение его позволяет выходить из небольших локальных минимумов.

Исходя из проведенных в данном разделе рассуждений можно сделать следующие выводы :

1. Нейронная сеть с одним скрытым слоем позволяет осуществить любое отображение входных сигналов в выходные;
2. Количество нейронных элементов в промежуточном слое должно быть меньше количества тренировочных образцов;
3. Мощность нейронной сети можно увеличивать как за счет количества нейронов в слое, так и за счет количества слоев. Если на нейронную сеть накладываются ограничения пункта 2 и она не может решить поставленную задачу, то необходимо увеличивать количество скрытых

слоев нейронной сети.

4. Случайная инициализация весовых коэффициентов нейронной сети должна проходить в достаточно узком диапазоне значений.

3.8. Адаптивный шаг обучения

В разделе 3.5 отмечалось, что в стандартном алгоритме обратного распространения ошибки существует проблема выбора подходящего шага обучения, чтобы увеличить быстродействие и обеспечить сходимость алгоритма. Для выбора адаптивного шага обучения $\alpha(t)$ можно использовать метод наискорейшего спуска [27, 30, 31]. В соответствии с ним, на каждой итерации обучения нейронной сети, необходимо выбирать шаг обучения для каждого слоя таким, чтобы минимизировать среднеквадратичную ошибку сети :

$$\alpha(t) = \min E(y_j(t+1)) \quad (3.44)$$

где $j = \overline{1, m}$, m – количество нейронных элементов последнего слоя.

Выходное значение j -го нейрона зависит от функции активации нейронных элементов и в общем случае определяется следующим образом :

$$y_j(t+1) = F(\omega_j(t+1), T_j(t+1)) \quad (3.45)$$

При этом весовые коэффициенты и пороги нейронной сети модифицируются, как

$$\omega_j(t+1) = \omega_j(t) - \alpha(t) \frac{\partial E}{\partial \omega_j(t)}, \quad (3.46)$$

$$T_j(t+1) = T_j(t) - \alpha(t) \frac{\partial E}{\partial T_j(t)}. \quad (3.47)$$

Среднеквадратичная ошибка нейронной сети равняется:

$$E = \frac{1}{2} \sum_j (y_j - t_j)^2. \quad (3.48)$$

Тогда для нахождения $\alpha(t)$ необходимо решить следующее уравнение:

$$\frac{\partial E}{\partial \alpha(t)} = \frac{\partial E}{\partial y_j(t+1)} \cdot \frac{\partial y_j(t+1)}{\partial \alpha(t)} = 0 \quad (3.49)$$

Данное уравнение невозможно решить относительно $\alpha(t)$ аналитическим путем. Поэтому в ряде работ для определения адаптивного шага обучения предлагается использовать методы линейного поиска [41]. Однако это связано со значительными вычислениями. Поэтому можно предложить приближенный метод нахождения скорости обучения $\alpha(t)$. Он базируется на разложении функции активации нейронных элементов в ряд Тейлора. Рассмотрим это подробно.

Пусть выходное значение j -ого нейрона последнего слоя нейронной сети равняется:

$$y_j(t) = F(S_j(t)),$$

$$S_j(t) = \sum_i y_i(t) \omega_{ij}(t) - T_j(t). \quad (3.51)$$

где $y_i(t)$ – выходное значение i -ого нейрона скрытого слоя.

Для определения взвешенной суммы j -ого нейрона в момент

времени $t+1$ подставим в (3.51) выражения (3.46) и (3.47):

$$\begin{aligned} S_j(t+1) &= \sum_i y_i (\omega_{ij} - \alpha \frac{\partial E}{\partial \omega_{ij}}) - T_j + \alpha \frac{\partial E}{\partial T_j} = \\ &= \sum_i y_i \omega_{ij} - T_j - \alpha \cdot (\sum_i y_i \cdot \frac{\partial E}{\partial \omega_{ij}} - \frac{\partial E}{\partial T_j}). \end{aligned} \quad (3.52)$$

Обозначим

$$a_j = \sum_i y_i \frac{\partial E}{\partial \omega_{ij}} - \frac{\partial E}{\partial T_j}. \quad (3.53)$$

Тогда выражение (3.52) можно представить в следующем виде:

$$S_j(t+1) = S_j(t) - \alpha \cdot a_j. \quad (3.54)$$

Выходное значение j -ого нейрона в момент времени $t+1$ равняется

$$y_j(t+1) = F(S_j(t+1)). \quad (3.55)$$

Разложим данное выражение по формуле Тейлора и ограничимся первыми двумя членами:

$$y_j(t+1) = F(0) + F'(0) \cdot S_j(t+1), \quad (3.56)$$

где

$$F'(0) = \frac{\partial F}{\partial S_j} \text{ при } S_j=0.$$

Подставим в (3.56) выражение (3.54). Тогда

$$y_j(t+1) = F(0) + F'(0)S_j(t) - \alpha F'(0)a_j \quad (3.57)$$

Так как

$$y_j(t) = F(0) + F'(0)S_j(t), \quad (3.58)$$

то выражение (3.57) можно представить в следующем виде:

$$y_j(t+1) = y_j(t) - \alpha F'(0)a_j \quad (3.59)$$

Для определения адаптивного шага обучения необходимо обеспечить :

$$E = \frac{1}{2} \sum_j (y_j(t+1) - t_j)^2 \rightarrow \min \quad (3.60)$$

Тогда

$$\frac{\partial E}{\partial \alpha} = \sum_j (y_j(t) - t_j - \alpha F'(0)a_j) \cdot (-F'(0)a_j) = 0$$

Выражая из последнего уравнения $\alpha(t)$, получим :

$$\alpha(t) = \frac{\sum_j (y_j(t) - t_j)a_j}{F'(0) \sum_j a_j^2} \quad (3.61)$$

Так как $\frac{\partial^2 E}{\partial \alpha^2} > 0$, то при данном $\alpha(t)$ обеспечивается минимум

среднеквадратичной ошибки. Найдем выражение для a_j . Для этого определим

$$\frac{\partial E}{\partial \omega_{ij}} = \frac{\partial E}{\partial y_j} \cdot \frac{\partial y_j}{\partial S_j} \cdot \frac{\partial S_j}{\partial \omega_{ij}} = (y_j - t_j) F'(S_j) \cdot y_i, \quad (3.62)$$

$$\frac{\partial E}{\partial T_j} = \frac{\partial E}{\partial y_j} \cdot \frac{\partial y_j}{\partial S_j} \cdot \frac{\partial S_j}{\partial T_j} = -(y_j - t_j) F'(S_j), \quad (3.63)$$

Подставляя (3.62) и (3.63) в выражение (3.53), получим:

$$a_j = (1 + \sum_i y_i^2) \cdot (y_j - t_j) \cdot F'(S_j) \quad (3.64)$$

Исходя из принципа независимости слоев, предполагаем, что

$$\gamma_j = y_j - t_j \quad (3.65)$$

Подставляя выражения (3.65) и (3.64) в (3.61), получим приближенное выражение для вычисления адаптивного шага обучения различных слоев нейронной сети:

$$\alpha(t) = \frac{\sum_j \gamma_j^2 F'(S_j)}{F'(0) (1 + \sum_i y_i^2) (\sum_j \gamma_j^2 (F'(S_j))^2)}, \quad (3.66)$$

где γ_j - ошибка j -ого нейронного элемента, которая для различных слоев сети вычисляется в соответствии с (3.14).

Рассмотрим определение выражения (3.66) для различных функций

активации нейронных элементов.

3.8.1. Сигмоидная функция

Так как производные сигмоидной функции :

$$y_j' = F'(S_j) = y_j(1 - y_j),$$

$$y_j'(0) = F'(0) = \frac{1}{4},$$

то выражение для адаптивного шага обучения можно представить в следующем виде:

$$\alpha(t) = \frac{4 \sum_j \gamma_j^2 y_j (1 - y_j)}{(1 + \sum_i y_i^2) \cdot \sum_j \gamma_j^2 y_j^2 (1 - y_j)^2}, \quad (3.67)$$

где ошибка j -ого нейронного элемента скрытого слоя равняется :

$$\gamma_j = \sum_i \gamma_i y_i (1 - y_i) \omega_{ji} \quad (3.68)$$

Здесь i характеризует нейроны следующего слоя по отношению к слою j .

3.8.2. Гиперболический тангенс.

Так как

$$y_j' = F'(S_j) = \frac{1}{ch^2(S_j)} = (1 - y_j^2),$$

$$y_j'(0) = F'(0) = 1,$$

то

$$\alpha(t) = \frac{\sum_j \gamma_j^2 (1 - y_j^2)}{(1 + \sum_i y_i^2) \cdot \sum_j \gamma_j^2 (1 - y_j^2)^2} \quad (3.69)$$

Ошибка j -ого нейрона скрытого слоя определяется как

$$\gamma_j = \sum_i \gamma_i (1 - y_i^2) \omega_{ji} \quad (3.70)$$

3.8.3. Биполярная сигмоидная функция

Определим производные данной функции :

$$y_j' = F'(S_j) = \frac{1}{2} (1 - y_j^2),$$

$$y_j'(0) = F'(0) = \frac{1}{2}.$$

Тогда

$$\alpha(t) = \frac{4 \sum_j \gamma_j^2 (1 - y_j^2)}{(1 + \sum_j y_j^2) \sum_j \gamma_j^2 (1 - y_j^2)^2} \quad (3.71)$$

где

$$\gamma_j = \sum_i \gamma_i \frac{1}{2} (1 - y_j^2) \omega_{ji} \quad (3.72)$$

Заметим, что в приведенных выше выражениях $\alpha(t)$ вычисляется отдельно для каждого слоя нейронной сети. Как показывают эксперименты, при использовании адаптивного шага обучения могут получаться слишком большие значения $\alpha(t)$. Это может привести к десинхронизации процесса обучения, когда весовые коэффициенты резко изменяются в определенном направлении. В результате изменение среднеквадратичной ошибки с течением времени будет иметь колебательный характер. Поэтому рекомендуется ограничивать $\alpha(t)$ по абсолютному значению не больше 10 ($\alpha(t) \leq 10$). Полученные выше выражения для адаптивного шага обучения позволяют значительно повысить скорость обучения нейронной сети и избежать выбора подходящего шага. Это является существенным преимуществом по сравнению со стандартным алгоритмом обратного распространения ошибки. Хотя при удачном выборе постоянного шага обучения данный алгоритм будет сходиться не быстрее, чем метод градиентного спуска.

Как и в случае линейной нейронной сети (раздел 2.8), здесь также возможен групповой вариант обучения. Тогда величина адаптивного шага обучения определяется в соответствии со следующим выражением :

$$\alpha(t) = \frac{\sum_k \sum_j \gamma_j^k a_j^k}{F'(0) \cdot \sum_k \sum_j (a_j^k)^2} \quad (3.73)$$

$$a_j^k = \sum_p \gamma_j^p F'(S_j^p) (1 + \sum_i y_i^p y_i^k)$$

где $k = \overline{1, L}$; L – количество образцов, после подачи которых на вход нейронной сети производится настройка весовых коэффициентов и порогов нейронов. Данное выражение получается аналогично выражению (3.66). Модификация синаптических связей при этом происходит следующим образом:

$$\omega_{ij}(L) = \omega_{ij}(0) - \alpha(t) \cdot \sum_k (y_j^k - t_j^k) F'(S_j^k) y_i^k \quad (3.74)$$

$$T_j(L) = T_j(0) + \alpha(t) \cdot \sum_k (y_j^k - t_j^k) F'(S_j^k) \quad (3.75)$$

Как показали эксперименты, групповой метод с точки зрения временной сложности и устойчивости обучения является менее эффективным по сравнению с обычным методом, где используется выражение (3.66) для выбора адаптивного шага.

3.9. Модификация адаптивного шага обучения

Приведенные в предыдущем разделе выражения для адаптивного шага обучения являются приближенными. Степень неточности их определяется разложением в ряд Тейлора и получаемым при этом выражением (3.59), а

Глава 3. Многослойные нейронные сети

также использованием при выводе (3.66) принципа независимости слоев. Этот принцип предполагает обобщение результатов, полученных для последнего слоя на остальные слои нейронной сети. С аналитической точки зрения это является не верным. Поэтому можно нейтрализовать данный недостаток. В дальнейшем будем использовать обозначения приведенные на рисунке 3.5.

Теорема 3.4. Для последнего скрытого слоя нейронной сети приближенное выражение для адаптивного шага обучения можно представить следующим образом:

$$\alpha_2(t) = \frac{\sum_i C_i \sum_j (y_j - t_j) \omega_{ij}}{(F'(0))^2 \sum (\sum C_i \omega_{ij})^2} \quad (3.76)$$

где

$$C_i = \gamma_i F'(S_i) (1 + \sum_k y_k^2) \quad (3.77)$$

Здесь j -выходной, i -последний скрытый, а k – предпоследний скрытый слой нейронной сети.

Доказательство. Используя результаты раздела (3.7) выражение для взвешенной суммы i -го нейронного элемента последнего скрытого слоя можно представить в следующем виде:

$$S_i(t+1) = S_i(t) - \alpha_2 C_i \quad (3.78)$$

где

$$C_i = \sum_k y_k \frac{\partial E}{\partial \omega_{ij}} - \frac{\partial E}{\partial T_i} = \gamma_i F'(S_i) (1 + \sum_k y_k^2) \quad (3.79)$$

Тогда используя разложение в ряд Тейлора выходное значение i -го нейрона равняется:

$$y_i(t+1) = F(0) + F'(0)S_i(t+1) \quad (3.80)$$

Преобразуя последнее выражение, получим:

$$y_i(t+1) = y_i(t) - \alpha_2 F'(0)C_i \quad (3.81)$$

Взвешенную сумму j -го нейронного элемента выходного слоя нейронной сети можно представить, как

$$S_j(t+1) = \sum_i \omega_{ij} y_i(t+1) - T_j \quad (3.82)$$

Подставляя (3.81) в (3.82) и преобразуя его, получим:

$$S_j(t+1) = S_j(t) - \alpha_2 F'(0) \sum_i \omega_{ij} C_i \quad (3.83)$$

Выходное значение j -го нейронного элемента последнего слоя определяется следующим образом:

$$y_j(t+1) = F(0) + F'(0)S_j(t+1) = y_j(t) - (F'(0))^2 \alpha_2 \sum_i \omega_{ij} C_i \quad (3.84)$$

Тогда

$$\frac{\partial E}{\partial \alpha_2} = \sum_j (y_j(t) - t_j - \alpha_2 (F'(0))^2 \sum_i \omega_{ij} C_i) (-F'(0))^2 \sum_i \omega_{ij} C_i \quad (3.85)$$

Приравнявая последнее уравнение к нулю и выражая из него $\alpha_2(t)$ получим:

3.10. Алгоритм послонного обучения

Как уже отмечалось, сходимость алгоритма обратного распространения ошибки зависит от начальной инициализации синаптических связей. Это происходит из-за наличия локальных минимумов в целевой функции. В результате не всякая попытка приводит к обучению нейронной сети, что характеризует неустойчивость алгоритма обучения. В данном разделе рассматривается альтернативный алгоритм обучения нейронной сети, который при решении некоторых задач проявил 100% устойчивость [32].

Рассмотрим теоретические основы алгоритма послонного обучения на примере нейронной сети с одним скрытым слоем, где в качестве функции активации используется гиперболический тангенс. Общая структура сети изображена на рис. 3.9.

При использовании функции гиперболический тангенс значение j -го нейрона выходного слоя определяется как

$$y_j = th(S_j) = \frac{e^{S_j} - e^{-S_j}}{e^{S_j} + e^{-S_j}} \quad (3.89)$$

где S_j характеризует взвешенную сумму j -го элемента.

Она определяется следующим образом :

$$S_j = \sum \omega_{ij} y_i - T_j \quad (3.90)$$

Среднеквадратичная ошибка нейронной сети для одного образа определяется, как

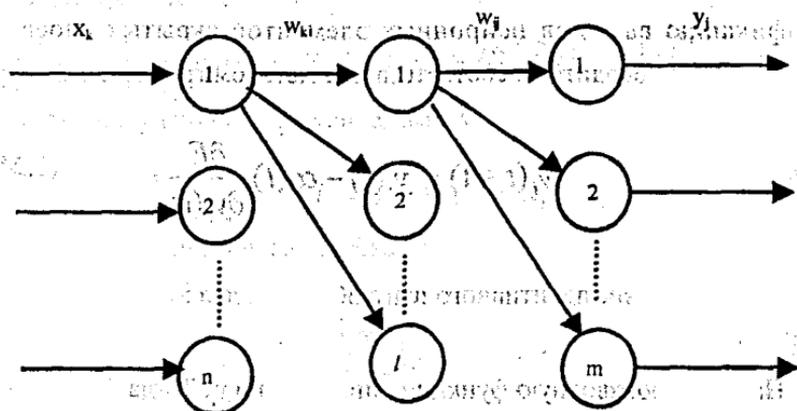


Рис. 3.9. Нейронная сеть с одним скрытым слоем.

$$E = \frac{1}{2} \sum_j (y_j - t_j)^2 \quad (3.91)$$

Тогда при использовании алгоритма обратного распространения ошибки выражения для обучения нейронной сети можно представить следующим образом:

$$\omega_{ij}(t+1) = \omega_{ij}(t) - \alpha(t) \gamma_j (1 - y_j^2) y_i, \quad (3.92)$$

$$T_j(t+1) = T_j(t) + \alpha(t) \gamma_j (1 - y_j^2), \quad (3.93)$$

где $\alpha(t)$ - определяется в соответствии с выражением (3.69), а γ_j - в соответствии с выражением (3.41) или (3.42).

Алгоритм послойного обучения предполагает наряду с настройкой весовых коэффициентов и порогов нейронных элементов, проводить так же

модификацию выходов нейронных элементов скрытых слоев. Такая модификация проводится в соответствии с методом градиентного спуска

$$y_i(t+1) = y_i(t) - \alpha_i(t) \frac{\partial E}{\partial y_i(t)}, \quad (3.94)$$

где $\alpha_i(t)$ - значение адаптивного шага обучения для i -го нейрона скрытого слоя.

Найдем производную функции ошибки E по y_i . Тогда

$$\frac{\partial E}{\partial y_i} = \frac{\partial E}{\partial y_j} \frac{\partial y_j}{\partial S_j} \frac{\partial S_j}{\partial y_i} = \sum_j (y_j - t_j)(1 - y_j^2) w_{ij} = \gamma_i$$

где γ_i характеризует ошибку i -го нейронного элемента скрытого слоя.

Для нахождения адаптивного шага обучения $\alpha_i(t)$ будем использовать метод наискорейшего спуска. В соответствии с ним

$$\alpha_i(t) = \min \left\{ E(y_i - \alpha_i(t) \frac{\partial E}{\partial y_i(t)}) \right\} \quad (3.95)$$

Определим взвешенную сумму, как

$$S_j(t+1) = \omega_{ij}(y_i(t) - \alpha_i \gamma_i) + \sum_{k \neq i} \omega_{kj} y_k(t) - T_j \quad (3.96)$$

После преобразования этого выражения, получим

$$S_j(t+1) = S_j(t) - \alpha_i \gamma_i \omega_{ij} \quad (3.97)$$

Раскладывая функцию $y_j(t+1) = th(S_j(t+1))$ в ряд Тейлора и преобразуя полученное выражение, имеем

$$y_j(t+1) = y_j(t) - \alpha_i \gamma_i \omega_{ij} \quad (3.98)$$

Отсюда среднеквадратичная ошибка сети

$$E = \frac{1}{2} \sum (y_j(t+1) - t_j)^2 \quad (3.99)$$

Находим такое значение α_i , при котором среднеквадратичная ошибка сети является минимальной

$$\frac{\partial E}{\partial \alpha_i} = \sum_j (y_j - t_j - \alpha_i \gamma_i \omega_{ij})(-\gamma_i \omega_{ij}) = 0 \quad (3.100)$$

Выражая из последнего уравнения α_i , находим значение адаптивного шага обучения для i -го нейрона скрытого слоя

$$\alpha_i = \frac{\sum_j (y_j - t_j) \omega_{ij}}{\sum_j \omega_{ij}^2 \sum_j (y_j - t_j)(1 - y_j^2) \omega_{ij}} \quad (3.101)$$

Упростим последнее выражение. Тогда

$$\alpha_i = \frac{\sum_j (y_j - t_j) \omega_{ij}}{\gamma_i \sum_j \omega_{ij}^2} \quad (3.102)$$

В результате получаем, что выходы нейронных элементов скрытого слоя должны модифицироваться на основе следующего выражения:

$$y_i(t+1) = y_i(t) - \alpha_i \gamma_i \quad (3.103)$$

Определяя при помощи выражения (3.103) эталонные выходы \bar{y}_i нейронных элементов скрытого слоя, можно отдельно рассматривать обучение скрытого слоя сети. При этом ошибка i -го нейрона скрытого слоя определяется, как

$$\bar{\gamma}_i = y_i - \bar{y}_i \quad (3.104)$$

где y_i - реальное выходное значение i -го нейрона скрытого слоя.

Так как область значений D функции гиперболический тангенс принадлежит отрезку $[-1,1]$, то после получения эталонных выходов \bar{y}_i необходимо провести их нормализацию. Для этого определяется максимальное значение среди эталонных выходов всех образов

$$y_m = \max_{k, l} |y_i^k| \quad (3.105)$$

где $i = \overline{1, l}$, $k = \overline{1, L}$.

Если значение $y_m \in D$, то нормализация не происходит. В противном случае производится пересчет всех \overline{y}_i в соответствии со следующим выражением:

$$\overline{y}_i^k = \frac{y_i^k}{y_m} \quad (3.106)$$

Затем происходит изменение весовых коэффициентов нейронных элементов выходного слоя

$$\omega_{ij} = y_m \omega_{ij} \quad (3.107)$$

где $i = \overline{1, l}$, $j = \overline{1, m}$.

Алгоритм послыйного обучения состоит из следующих шагов:

1. Случайная инициализация весовых коэффициентов нейронной сети и задание минимальной среднеквадратичной ошибки E_m .
2. На вход сети последовательно подаются L входных образов, и для последнего слоя производится модификация весовых коэффициентов и порогов нейронных элементов в соответствии с выражениями (3.92) и (3.93). Одновременно для каждого входного паттерна в соответствии с (3.103) осуществляется модификация выходных значений y_i нейронов скрытого слоя.
3. Рассматривается только последний слой нейронной сети. В качестве входной информации используются полученные на предыдущем шаге выходные значения y_i нейронов скрытого слоя, а в качестве выходной — эталонные выходы.
4. Входные образы y_i последовательно подаются на нейронные

элементы последнего слоя и производится модификация весовых коэффициентов, порогов и значений y_i в соответствии с выражениями (3.92), (3.103). Данный пункт продолжается до тех пор, пока суммарная среднеквадратичная ошибка сети не станет меньше заданной. В результате выполнения данного пункта произойдет определение эталонных значений \bar{y}_i нейронов скрытого слоя.

5. Осуществляется нормализация эталонных значений \bar{y}_i .
6. Для L входных образов производится модификация весовых коэффициентов и порогов следующего слоя нейронной сети. При этом ошибка i -го нейронного элемента равняется

$$\bar{y}_i = y_i - \bar{y}_i.$$

7. Процедура повторяется, начиная с шага 2, пока суммарная среднеквадратичная ошибка сети не станет меньше заданной.

Рассмотренный алгоритм показал 100% устойчивость при решении задачи циклического кодирования информации. Результаты применения данного алгоритма обсуждаются в следующем разделе.

3.11. Сравнительный анализ алгоритмов обучения

Рассмотрим применение различных алгоритмов обучения для решения задач прогнозирования, распознавания печатных цифр и кодирования информации.

3.11.1. Прогнозирование временных рядов

Как отмечалось в разделе 2.12, для прогнозирования временных рядов

Глава 3: Многослойные нейронные сети

используется метод скользящего окна. Он характеризуется длиной окна p , которая определяет количество нейронов распределительного слоя сети. При использовании нелинейной нейронной сети, оценка временного ряда в момент времени t определяется как нелинейное преобразование от предыдущих p членов ряда:

$$\overline{y(t)} = F(y(t-1), y(t-2), \dots, y(t-p))$$

Для обучения нейронной сети прогнозированию используется выборка известных членов ряда. После обучения сеть должна прогнозировать временной ряд на упреждающий промежуток времени. Рассмотрим прогнозирование математических функций. Пусть дана функция

$$y = 0.1 \sin(3x) + 0,5$$

Для получения дискретного ряда протабулируем ее с шагом равным 0.25. Предположим, что размер окна $p=4$, а размерность обучающей выборки - 15 членов ряда. В качестве сети будем использовать нейронную сеть с одним скрытым слоем и сигмоидной функцией активации. Структура сети состоит из 5 входных, 11 скрытых и одного выходного нейрона. На рис. 3.7 приведены графики изменения суммарной ошибки сети в зависимости от количества итераций обучения.

При обучении с постоянным шагом обучения $\alpha = 0,001$ суммарная среднеквадратичная ошибка сети достигает значения $68 \cdot 10^{-3}$ при номере итерации равном 1000 и дальше практически не изменяется. При обучении с адаптивным шагом такой ошибки нейронная сеть достигает уже при номере итерации равном 240. При $N=1000$ ошибка обучения достигает значения $2.03 \cdot 10^{-3}$, что намного меньше чем при постоянном шаге. Таким образом на данном примере алгоритм обучения с адаптивным шагом работает в 4 раза быстрее чем при использовании постоянного шага. Всплески на графике изменения суммарной среднеквадратичной ошибки происходят из-за того,

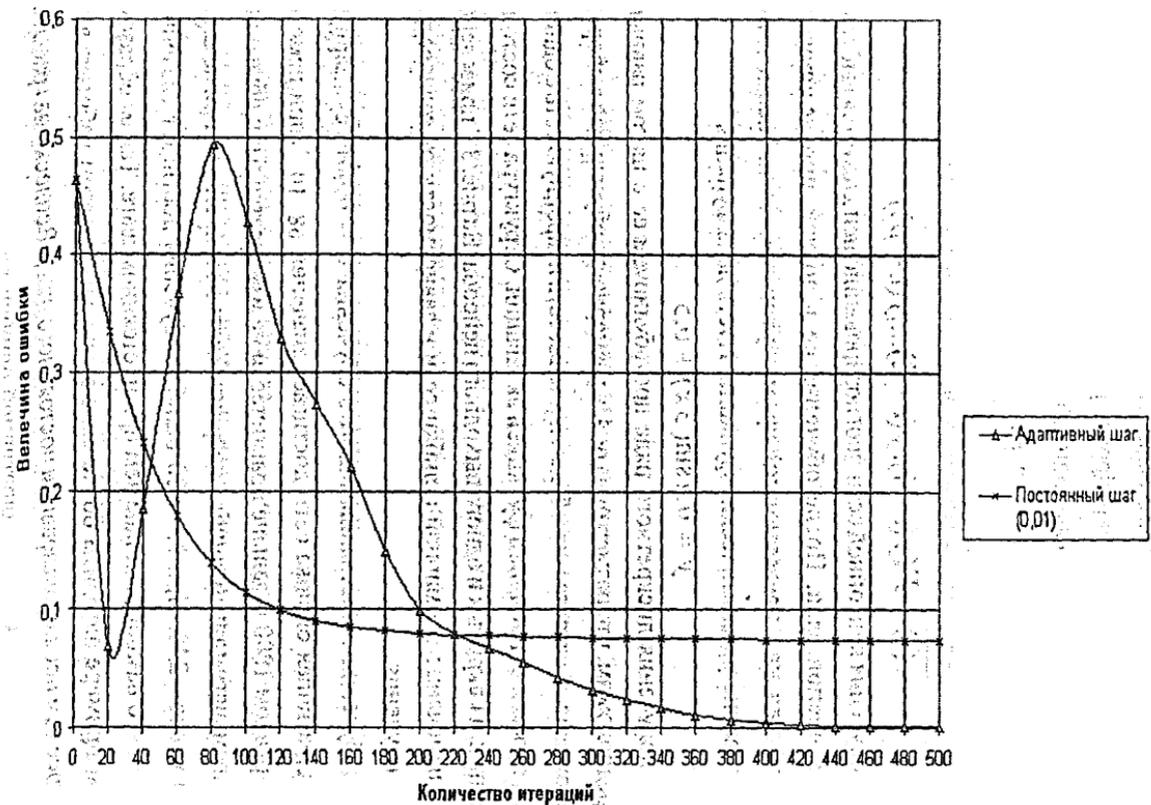


рис. 3.7. Графики изменения среднеквадратичной ошибки в процессе обучения ИНС

Таблица 3.1.

Шаг прогноза	Реальное значение	Прогнозируемое значение	Среднеквадратичная ошибка
1	5,99E-01	5,98E-01	4,47758E-07
2	6,48E-01	6,43E-01	2,86894E-05
3	6,97E-01	6,77E-01	0,000388288
4	7,42E-01	7,01E-01	0,001664458
5	7,82E-01	7,18E-01	0,004210635
6	8,17E-01	7,28E-01	0,007879152
7	8,45E-01	7,34E-01	0,01211502
8	8,64E-01	7,37E-01	0,016182886
9	8,75E-01	7,36E-01	0,019372797
10	8,77E-01	7,32E-01	0,021148307

что применяемый здесь метод наискорейшего спуска минимизирует на каждой итерации среднеквадратичную ошибку для одного паттерна. Для устранения всплесков необходимо ограничить $\alpha(t)$ по абсолютной величине. Как уже отмечалось, при удачном выборе постоянного шага обучения, алгоритм с адаптивным шагом будет сходиться не быстрее чем алгоритм с постоянной скоростью обучения. Однако здесь возникает проблема выбора подходящего шага. С этой точки зрения алгоритм с адаптивным шагом имеет преимущество. В таблице 3.1 приведены характеристики прогнозирующих свойств многослойной нейронной сети.

При этом прогноз осуществляется за пределами обучающей выборки на 10 шагов вперед. Несмотря на то, что объем обучающей выборки является не большим ($L=15$), сеть демонстрирует достаточно хорошие прогнозирующие свойства.

3.11.2. Распознавание цифр

Пусть десятичные цифры от 0 до 9 изображаются на бинарной матрице

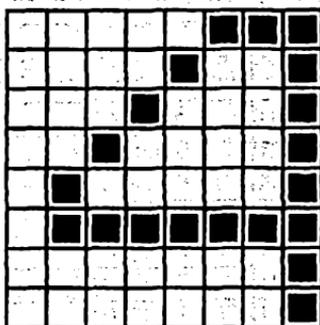


Рис. 3.8. Изображение цифры на матрице размерности 8x8.

размерностью 8x8 (рис 3.8).

Задача нейронной сети состоит в распознавании таких цифр в условиях помех. Будем использовать для этого трехслойную нейронную сеть, состоящую из 64-ех нейронов входного слоя, 10 нейронов скрытого и 10 нейронов выходного слоя. Каждому нейронному элементу выходного слоя поставим в соответствие определенную десятичную цифру по порядку нумерации нейронов. Тогда номер выходного нейрона, который имеет максимальную активность на выходе, определяет соответствующую десятичную цифру. Размерность обучающей выборки равняется десяти. В таблице 3.2 приведены эксперименты по обучению такой нейронной сети.

В последнем столбце таблицы показано количество итераций, необходимых для обучения нейронной сети, при использовании адаптивного шага обучения. Все данные получены для одной и той же суммарной среднеквадратичной ошибки. Как следует из таблицы алгоритм обучения с адаптивным шагом обучения характеризуется наименьшей временной сложностью. После обучения нейронная сеть продемонстрировала способность распознавать искаженные входные образы. В заключение отметим, что алгоритм с адаптивным шагом не решает проблемы

Таблица 3.2.

Значение альфа	0,1	0,2	0,6	0,8	var
Количество итераций	4318	2529	930	601	224

устойчивости процесса обучения.

3.11.3. Формирование циклических кодов

Процедура образования циклических кодов состоит в следующем [44]. Многочлен информационного сообщения $S(x)$ умножается на x^m , где m — степень примитивного полинома и соответственно количество контрольных разрядов в циклическом коде. Результат произведения делится на примитивный полином и определяется остаток от деления $R(x)$, который характеризует контрольные разряды циклического кода. Путем добавления контрольных разрядов к информационным образуется циклический код:

$$P(x) = S(x)x^m \oplus R(x).$$

Возьмем в качестве примитивного следующий полином:

$$g(x) = x^3 + x^2 + 1$$

Тогда количество контрольных разрядов равняется трем. Определим разрядность информационного сообщения равной четырем. Задача нейронной сети состоит в преобразовании информационного сообщения в циклический код. Она будет состоять из четырех нейронов входного, 8 нейронов скрытого и 7 нейронов выходного слоя. Размер обучающей выборки составляет 16 тренировочных наборов, которые образуются в соответствии с приведенным выше алгоритмом. В таблице 3.3 приведены сравнительные характеристики различных алгоритмов обучения. Строки соответствующие

Таблица 3.3.

Алгоритм обучения (значение шага)	Количество опытов	Количество отказов	Среднее количество итераций обучения
var	500	17	233,4
mvar	500	2	278,9
pvar	500	0	400
0.1	500	37	1264,4
0.2	500	39	634,6
0.3	500	30	434,6
0.4	500	36	428,3
0.5	500	41	356,3
0.6	500	61	258,8
0.7	500	66	207,6
0.8	500	51	184,5
0.9	500	55	168,6

$\alpha = \text{var}$ и $\alpha = m \text{var}$ характеризуют адаптивный и модифицированный адаптивный шаг обучения. Строка, соответствующая $p \text{var}$ характеризует послойный метод обучения. Количество опытов в таблице 3.3 соответствует числу попыток обучить сеть при использовании данного алгоритма. Число отказов характеризует количество опытов, когда не произошло обучения сети для решения поставленной задачи. В последнем столбце приведено количество равноценных итераций, которым соответствует одинаковая временная сложность. Как следует из таблицы 3.3 наилучшим с точки зрения временной сложности является алгоритм с постоянным шагом $\alpha = 0.9$. Однако ему соответствует большое количество отказов. Алгоритм с адаптивным шагом обладает меньшим числом отказов по сравнению с постоянным шагом и достаточно высоким быстродействием. Наименьшее количество отказов соответствует алгоритму с послойным обучением. В заключении следует

отметить, что нельзя делать вывод об универсальном преимуществе одного алгоритма обучения перед остальными. Для разных задач эффективность применения того или иного алгоритма может быть различной. Поэтому необходимо иметь набор различных алгоритмов для обучения нейронных сетей. В наших опытах по решению различных задач на многослойных нейронных сетях наиболее эффективными оказались алгоритм обучения с адаптивным шагом и алгоритм послойного обучения.

3.12. Гетерогенные нейронные сети

В предыдущих разделах рассматривались сети, нейронные элементы которых имеют одинаковую функцию активации. Такие нейронные сети называются *гомогенными*. В данном разделе рассмотрим вычисление адаптивного шага обучения для нейронных сетей, различные слои которых могут иметь разные функции активации нейронных элементов. Такие сети называются *гетерогенными*. Рассмотрим простейшую гетерогенную сеть, состоящую из одного скрытого слоя с нелинейной функцией активации нейронных элементов и выходного линейного нейрона (рисунок 3.9).

Тогда выходное значение нейронной сети равняется:

$$y = \sum_i v_i y_i - T, \quad (3.108)$$

где v_i – i -й весовой коэффициент выходного нейрона.

Выходные значения нейронных элементов скрытого слоя определяются, как

$$y_i = F(S_i) = F\left(\sum_l \omega_{li} x_l - T_i\right) \quad (3.109)$$

Для различных слоев нейронной сети необходимо использовать разные выражения для вычисления адаптивного шага обучения. Так как выходной

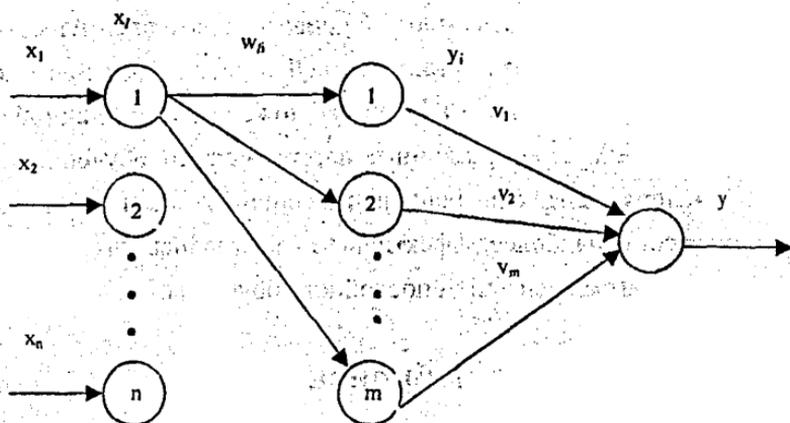


Рис. 3.9. Гетерогенная нейронная сеть.

слой имеет линейную функцию активации, то вычисление адаптивного шага обучения для него производится следующим образом:

$$\alpha_i = \frac{1}{1 + \sum_i y_i^2} \quad (3.110)$$

Для скрытого слоя определение адаптивного шага обучения можно производить используя выражения (3.61) или (3.76). При этом ошибка нейронных элементов скрытого слоя равняется:

$$\gamma_i = (y - t)v_i \quad (3.111)$$

Тогда выражение (3.61) для адаптивного шага обучения можно представить следующим образом:

$$\alpha_2 = \frac{\sum_i v_i^2 F'(S_i)}{F'(0)(1 + \sum_l x_l^2) \sum_i v_i^2 (F'(S_i))^2} \quad (3.112)$$

В случае гетерогенных нейронных сетей выражение (3.76) для модифицированного адаптивного шага обучения трансформируется следующим образом:

$$\alpha_2 = \frac{\sum_i C_i \sum_j (y_j - t_j) \omega_{ij}}{F_1'(0) F_2'(0) \sum_j (\sum_i C_i \omega_{ij})^2} \quad (3.113)$$

где $F_1'(0)$ и $F_2'(0)$ - соответственно производные функции активации для выходного и скрытого слоев.

Так как для линейной функции активации

$$F_1'(0) = 1,$$

то выражение (3.113) для нейронной сети, изображенной на рисунке 3.9, можно представить следующим образом:

$$\alpha_2 = \frac{\sum_i C_i (y - t) v_i}{F_2'(0) (\sum_i C_i v_i)^2} \quad (3.114)$$

С учетом того, что

$$C_i = \gamma_i F'(S_i) \left(1 + \sum_l x_l^2\right), \quad (3.115)$$

выражение (3.114) преобразуется к следующему виду:

$$\alpha_2 = \frac{1}{F_2'(0) \left(1 + \sum_l x_l^2\right) \sum_i v_i^2 F'(S_i)} \quad (3.116)$$

где $F'(S_i)$ – производная функции активации нейронных элементов скрытого слоя.

Если для нейронных элементов скрытого слоя используется сигмоидная функция активации, то

$$F_2'(0) = \frac{1}{4}$$

$$F'(S_i) = y_i(1 - y_i)$$

Тогда выражение (3.116) можно представить в следующем виде:

$$\alpha_2 = \frac{4}{\left(1 + \sum_l x_l^2\right) \sum_i v_i^2 y_i(1 - y_i)} \quad (3.117)$$

Обучающие правила для выходного слоя можно представить, как

$$v_i(t+1) = v_i(t) - \alpha_1(t)(y - t)y_i \quad (3.118)$$

$$T(t+1) = T(t) + \alpha_1(t)(y - t) \quad (3.119)$$

Соответственно для скрытого слоя

$$\omega_{li}(t+1) = \omega_{li}(t) - \alpha_2(t)\gamma_i F'(S_i)x_l \quad (3.120)$$

$$T_i(t+1) = T_i(t) + \alpha_2(t)\gamma_i F'(S_i) \quad (3.121)$$

где $\gamma_i = (y - t)v_i$.

Для определения адаптивного шага обучения $\alpha_1(t)$ выходного слоя используется выражение (3.110). При вычислении адаптивного шага обучения скрытого слоя сети может применяться выражение (3.117) или (3.112). Аналогичным образом можно определить правила обучения для различного типа гетерогенных сетей. Использование стандартного алгоритма обратного распространения ошибки для гетерогенных нейронных сетей характеризуется нестабильностью процесса обучения. Это происходит из-за применения различных функций активации нейронных элементов в таких сетях. Поэтому в гетерогенных нейронных сетях для обучения может использоваться алгоритм многократного распространения ошибки.

3.13. Алгоритм многократного распространения ошибки

Алгоритм обратного распространения ошибки предполагает для каждого тренировочного набора модификацию синаптических связей всех слоев нейронной сети. При этом изменение весовых коэффициентов одного слоя нейронной сети происходит без учета изменения весовых коэффициентов остальных слоев. Это может привести к нестабильности

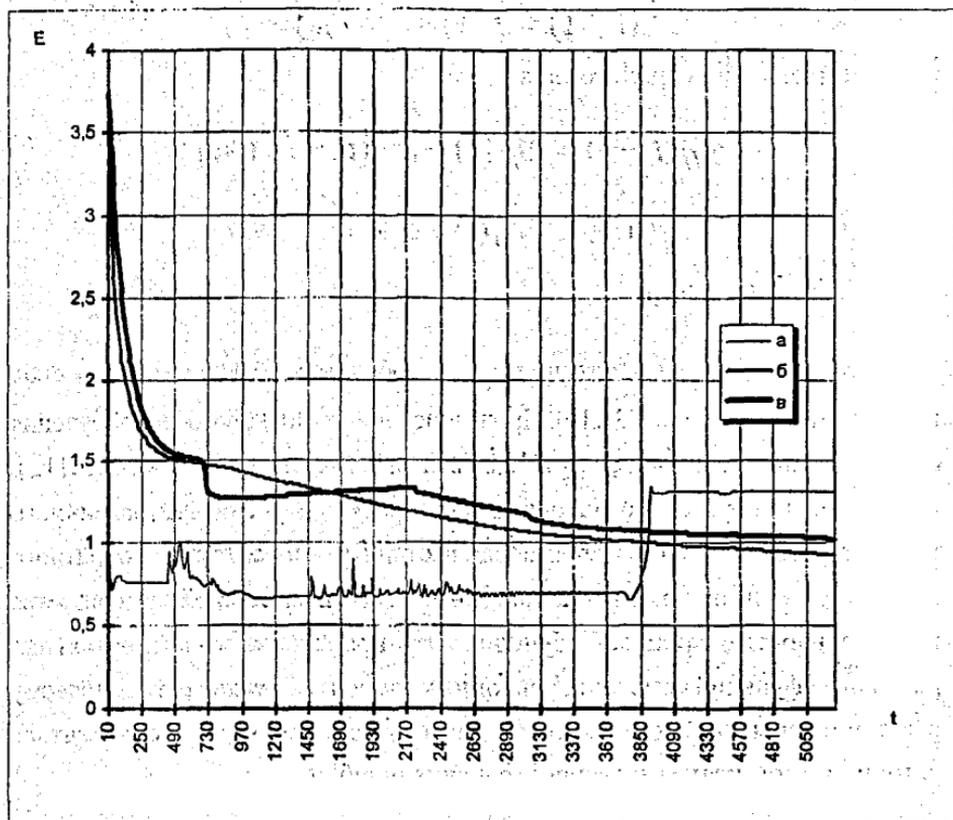


Рис. 3.10. Динамика обучения гетерогенной нейронной сети с сигмоидной функцией активации нейроэлементов скрытого слоя и линейной функцией активации нейроэлементов выходного слоя :

а - $\alpha_1 = 0.099, \alpha_2 = 9.99$; б - $\alpha_1 = 9.9 \cdot 10^{-4}, \alpha_2 = 9.9 \cdot 10^{-2}$; в -
 $\alpha_1 = 0.499, \alpha_2 = 4.99$.

процесса обучения, который характеризуется отсутствием тенденции к снижению среднеквадратичной ошибки сети. Особенно актуальной является эта проблема для гетерогенных нейронных сетей. Из-за различия функций

активации нейронных элементов может возникнуть рассинхронизация процесса обучения между разными слоями сети. В результате этого процесс обучения будет характеризоваться нестабильностью. На рис.3.10 показана динамика обучения гетерогенной нейронной сети, изображенной на рис.3.9. Здесь α_1 и α_2 соответственно шаги обучения для выходного и скрытого слоёв нейронной сети. Для устранения таких явлений можно использовать алгоритм многократного распространения ошибки. Он предполагает на каждой итерации обучения модификацию синаптических связей только одного слоя нейронной сети. В соответствии с этим каждый образ будет последовательно подаваться на нейронную сеть столько раз, сколько настраиваемых слоёв имеет сеть.

Пусть p -число настраиваемых слоёв нейронной сети. Тогда алгоритм многократного распространения ошибки состоит из следующих шагов:

1. Задается желаемая среднеквадратичная ошибка нейронной сети E_m .
2. Происходит случайная инициализация синаптических связей сети.
3. В счетчик количества настраиваемых слоёв записывается число p .
4. Подается первый тренировочный набор на вход нейронной сети.

Производится фаза прямого и обратного распространения сигнала. В результате осуществляется модификация весовых коэффициентов и порогов нейронных элементов только для p -го слоя нейронной сети.

$$v_{ip}(t+1) = \omega_{ip}(t) - \alpha(t)\gamma_p F_p'(S_p) y_i$$

$$T_p(t+1) = T_p(t) + \alpha(t)\gamma_p F_p'(S_p),$$

где $F_p(S_p)$ – функция активации нейронных элементов p -го слоя,
 $i = p-1$

5. Устанавливается: $p=p-1$

6. Если $p \neq 0$, то перейти к пункту 4. В противном случае перейти к шагу 7 алгоритма.
7. Повторить процесс обучения начиная с пункта 3 для всех тренированных наборов обучающей выборки.
8. Вычислить суммарную среднеквадратичную ошибку нейронной сети:

$$E = \frac{1}{2} \sum_{k=1}^L \sum_j (y_j^k - t_j^k)^2,$$

где L - общее количество тренированных наборов.

9. Если $E > E_m$, то перейти к пункту 3. В противном случае закончить процесс обучения.

3.14. Использование логарифмической функции активации

Недостаток рассмотренных ранее нелинейных функций активации состоит в том, что область выходных значений их ограничена отрезком $[0,1]$ или $[-1,1]$. Это приводит к необходимости масштабирования данных, если они не принадлежат указанным выше диапазонам значений. В работе [31] рассматривалась логарифмическая функция активации для решения задач прогнозирования. Она представляет собой обратную гиперболическую функцию арксинус. В результате выходное значение i -го нейронного элемента можно представить в виде следующего выражения:

$$y_i = \ln(S_i + \sqrt{S_i^2 + 1}), \quad (3.122)$$

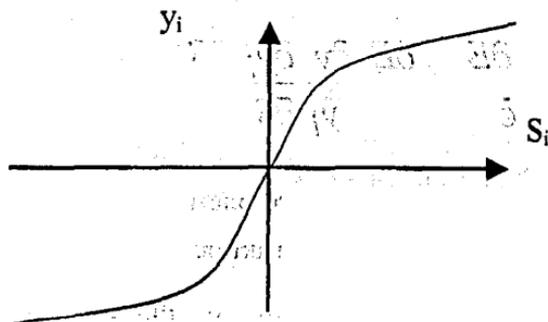


Рис. 3.11. Логарифмическая функция активации.

$$S_i = \sum_l w_{li} x_l = T_i. \quad (3.123)$$

Логарифмическая функция активации является монотонно возрастающей от $-\infty$ до $+\infty$ и имеет точку перегиба в начале координат (рис. 3.11).

Рассмотрим гетерогенную нейронную сеть (рис. 3.9), которая имеет слой скрытых нейронов с логарифмической функцией активации и один линейный выходной нейрон. Тогда выходное значение сети определяется, как

$$y = \sum_i v_i y_i - T. \quad (3.124)$$

Такая нейронная сеть, как будет показано далее, используется для решения задач прогнозирования. Определим для нее правила обучения. Тогда для скрытого слоя:

$$\frac{\partial E}{\partial w_{li}} = \frac{\partial E}{\partial y} \frac{\partial y}{\partial y_i} \frac{\partial y_i}{\partial S_i} \frac{\partial S_i}{\partial w_{li}} = \gamma_i x_l y_i', \quad (3.125)$$

$$\frac{\partial E}{\partial T_i} = \frac{\partial E}{\partial y} \frac{\partial y}{\partial y_i} \frac{\partial y_i}{\partial S_i} \frac{\partial S_i}{\partial T_i} = -\gamma_i y'_i, \quad (3.126)$$

где γ_i - ошибка i -го нейронного элемента скрытого слоя, y'_i - производная логарифмической функции активации от выходной активности i -го нейрона.

Ошибка i -го нейрона скрытого слоя равняется

$$\gamma_i = \frac{\partial E}{\partial y} \frac{\partial y}{\partial y_i} = (y - t) v_i. \quad (3.127)$$

Производная логарифмической функции

$$y'_i = \frac{\partial y_i}{\partial S_i} = \frac{1}{\sqrt{S_i^2 + 1}}. \quad (3.128)$$

Выражения для модификации настраиваемых параметров сети можно представить в следующем виде:

$$w_{li}(t+1) = w_{li}(t) - \alpha_2 (y - t) v_i x_l y'_i, \quad (3.129)$$

$$T_i(t+1) = T_i(t) + \alpha_2 (y - t) v_i y'_i. \quad (3.129)$$

Определим адаптивный шаг $\alpha_2(t)$ для обучения скрытого слоя нейронной сети. Тогда в соответствии с разделом 3.7

$$\alpha_2 = \frac{\sum_i v_i^2 y_i'}{(1 + \sum_l x_l^2) \sum_i v_i^2 (y_i')^2}$$

Параметры выходного слоя нейронной сети настраиваются в соответствии с выражениями (3.118), (3.119) и (3.110). Рассмотрим применение гетерогенных нейронных сетей для решения задач прогнозирования.

3.15. Применение гетерогенных нейронных сетей для прогнозирования

Гетерогенные нейронные сети могут использоваться для прогнозирования различных процессов. Рассмотрим прогнозирование стока вод реки Березины на территории республики Беларусь. Статистическая таблица размеров стока в зависимости от года приведена в приложении 1. Возьмем обучающую выборку, состоящую из членов ряда, взятых с 1938 по 1981 годы. Для функционирования нейронной сети будем использовать метод скользящего окна размерностью 10 и шагом 1. Тогда размерность тренировочной выборки равняется 35. Архитектура нейронной сети содержит 10 входных, 15 скрытых и 1 линейный выходной нейрон. Для обучения сети используется алгоритм многократного распространения ошибки с адаптивным шагом.

Пусть для нейронных элементов скрытого слоя используется сигмоидная функция активации. Для обучения сети возьмем члены временного ряда (приложение 1) умноженные на коэффициент 0.001. В

результате обучения общее количество итераций составило 10388 при суммарной среднеквадратичной ошибке $6.9 \cdot 10^{-3}$. Для проверки функционирования сети осуществим прогноз на упреждающий промежуток времени с 1982 по 1991 годы. Результаты прогноза показаны на рис. 3.12.

Проведем аналогичные эксперименты, где в качестве функции активации нейронных элементов скрытого слоя используется логарифмическая функция. В результате обучения такой сети была достигнута суммарная среднеквадратичная ошибка равная $3.9 \cdot 10^{-4}$ при количестве итераций 5388. Результаты осуществления прогноза приведены на рис.3.13.

Как следует из рисунка, нейронная сеть с логарифмической функцией активации производит прогноз значительно точнее, чем при использовании сигмоидной функции.

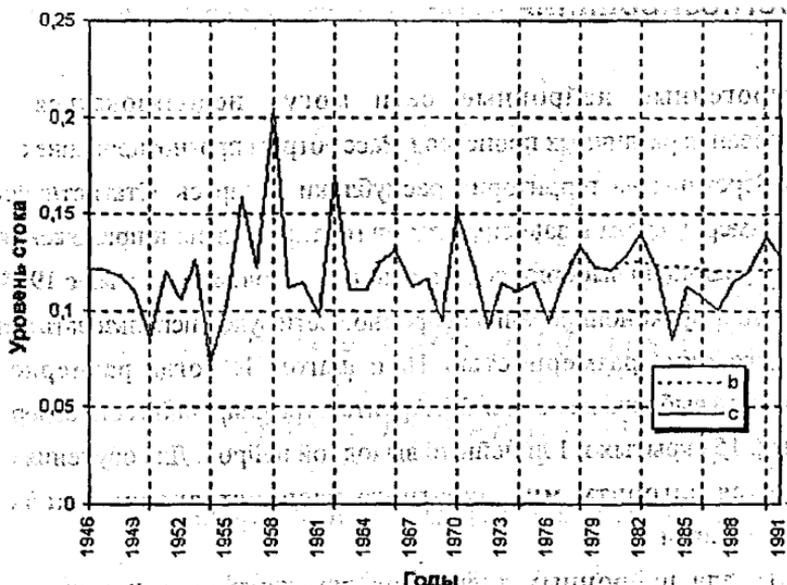


Рис. 3.12. Результат прогнозирования уровня стока вод реки Березина сигмоидно-линейной нейросетевой моделью: с - реальные значения стока; b - спрогнозированные значения.

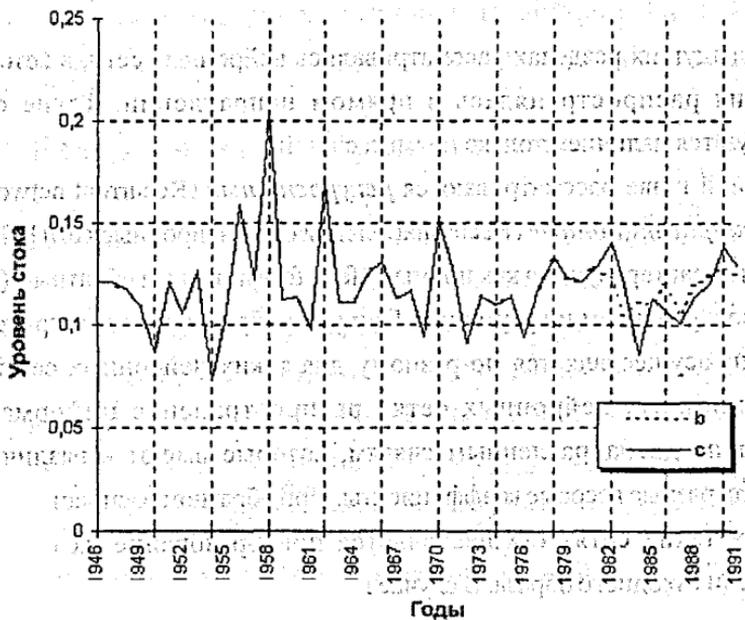


Рис. 3.13. Результат прогнозирования уровня стока вод реки Березина логарифмическо-линейной нейросетевой моделью: с - реальные значения стока; b- спрогнозированные значения.

ГЛАВА 4. МНОГОСЛОЙНЫЕ НЕЙРОННЫЕ СЕТИ С ОБРАТНЫМ РАСПРОСТРАНЕНИЕМ ИНФОРМАЦИИ

В предыдущих разделах рассматривались нейронные сети, в которых информация распространялась в прямом направлении. Такие сети характеризуются наличием только прямых связей.

В данной главе рассматриваются *рекуррентные* (Recurrent networks) [45-48] и *рециркуляционные* (Recirculation networks) нейронные сети [42, 45, 53, 54]. Они характеризуются как прямым (feed forward) так и обратным (feed back) распространением информации. При этом обратное распространение информации осуществляется по-разному для таких нейронных сетей. В рециркуляционных нейронных сетях распространение информации происходит по двунаправленным связям, которые имеют в различных направлениях разные весовые коэффициенты. При обратном распространении сигналов, в таких сетях осуществляется преобразование их с целью восстановления входного образа. В случае прямого распространения сигналов происходит сжатие входных данных. В результате осуществляется рециркуляция информации. Обучение рециркуляционных нейронных сетей производится без учителя. Рекуррентные нейронные сети в отличие от этого характеризуются обучением с учителем и обратными связями, по которым передаются результаты обработки сетью данных на предыдущем этапе. В результате этого входом рекуррентной нейронной сети в каждый фиксированный момент времени является вектор входных данных и результаты обработки информации сетью на предыдущем этапе. Обучение таких сетей базируется на алгоритме обратного распространения ошибки. Это дало повод, несмотря на различия между рекуррентными и рециркуляционными нейронными сетями отнести их к одному классу.

В данной главе описывается архитектура и обучение многослойных нейронных сетей с обратным распространением информации. Приводятся выражения для адаптивного шага обучения таких сетей. Для рециркуляционной нейронной сети предложен алгоритм послойного обучения, который отличается от стандартного алгоритма обратного распространения ошибки.

4.1. Рекуррентные нейронные сети

Рекуррентными нейронными сетями (Recurrent Networks) называются такие сети, в которых выходы нейронных элементов последующих слоев имеют синаптические соединения с нейронами предшествующих слоев. Это приводит к возможности учета результатов преобразования нейронной сетью информации на предыдущем этапе для обработки входного вектора на следующем этапе функционирования сети. Рекуррентные сети могут использоваться для решения задач прогнозирования и управления [42].

4.1.1. Архитектура рекуррентных сетей

Существуют различные варианты архитектур рекуррентных нейронных сетей [45-48]. В 1986 году Джордан (Jordan) предложил рекуррентную сеть (рис.4.1), в которой выходы нейронных элементов последнего слоя соединены посредством специальных входных нейронов с нейронами промежуточного слоя [45]. Такие входные нейронные элементы называются *контекстными* нейронами (context units). Они распределяют выходные данные нейронной сети на нейронные элементы промежуточного слоя (рис.4.1).

Количество контекстных нейронов равняется числу выходных нейронных элементов рекуррентной сети. В качестве выходного слоя таких сетей используются нейронные элементы с линейной функцией активации. Тогда

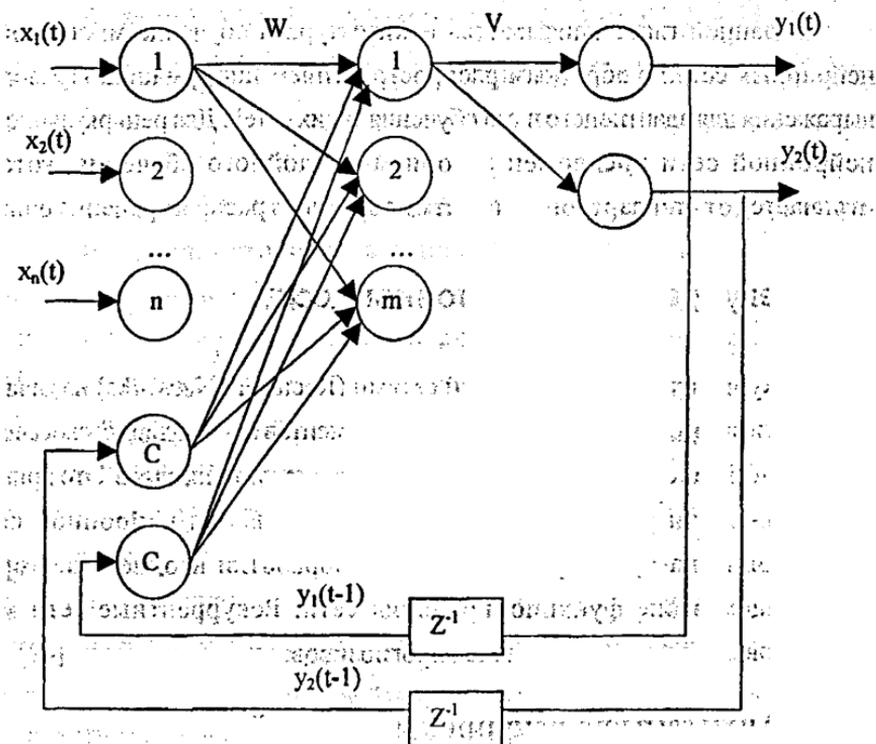


Рис. 4.1. Архитектура рекуррентной нейронной сети с обратными связями от нейронов выходного слоя; z^{-1} - элемент задержки; C - контекстный нейрон

выходное значение j -го нейронного элемента последнего слоя определяется как

$$y_j(t) = \sum_{i=1}^m v_{ij} p_i(t) - T_j, \quad (4.1)$$

где v_{ij} - весовой коэффициент между i -м нейроном промежуточного и

Глава 4. Многослойные нейронные сети с обратным распространением информации

j -м нейроном выходного слоя, $p_i(t)$ - выходное значение i -го нейрона промежуточного слоя, T_j - пороговое значение j -го нейрона выходного слоя.

Взвешенная сумма i -го нейронного элемента промежуточного слоя определяется следующим образом :

$$S_i(t) = \sum_{j=1}^n w_{ji} x_j(t) + \sum_{k=1}^p w_{ki} y_k(t-1) - T_i, \quad (4.2)$$

где w_{ji} - весовой коэффициент между j -м нейроном входного и i -м нейроном промежуточного слоя, T_i - пороговое значение i -го нейрона промежуточного слоя, n - размерность входного вектора, p - количество нейронов выходного слоя; w_{ki} - весовой коэффициент между k -м контекстным нейроном и i -м нейроном промежуточного слоя.

Тогда выходное значение i -го нейрона скрытого слоя

$$p_i(t) = F(S_i(t)) \quad (4.3)$$

В качестве функции нелинейного преобразования F обычно используется гиперболический тангенс или сигмоидная функция.

Другой вариант рекуррентной нейронной сети предложил Элман (Elman) в 1990 году [46]. В такой сети выходы нейронных элементов промежуточного слоя соединяются с контекстными нейронами входного слоя (рис.4.2).

Тогда взвешенная сумма i -го нейронного элемента промежуточного слоя равняется

$$S_i(t) = \sum_{j=1}^n w_{ji} x_j(t) + \sum_{k=1}^m w_{ki} p_k(t-1) - T_i, \quad (4.4)$$

где m - количество нейронов промежуточного слоя, $p_k(t-1)$ - выходное значение k -го нейрона промежуточного слоя.

Выходное значение k -го нейрона промежуточного слоя определяется

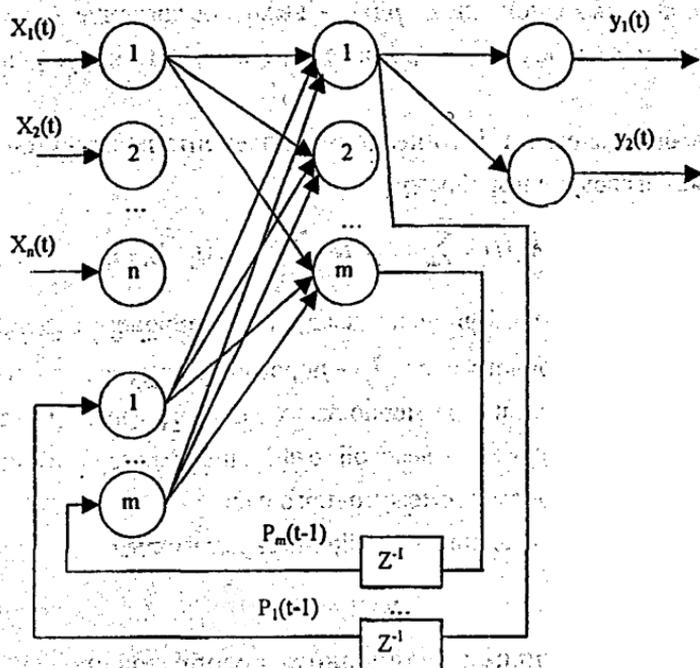


Рис.4.2. Архитектура рекуррентной сети с обратными связями от нейронов скрытого слоя

следующим образом:

$$p_k(t-1) = F(S_i(t-1)) \quad (4.5)$$

Для построения рекуррентных нейронных сетей можно использовать также два приведенных выше подхода [42]. В этом случае существуют обратные связи к контекстным нейронам как от нейронных элементов выходного, так и промежуточного слоя (рис. 4.3)

Количество контекстных нейронов входного слоя равняется общему числу нейронов промежуточного и выходного слоев. Тогда

$$S_i(t) = \sum_{j=1}^n w_{ji} x_j(t) + \sum_{k=1}^p w_{ki} y_k(t-1) + \sum_{l=1}^m w_{li} p_l(t-1) - T_i, \quad (4.6)$$

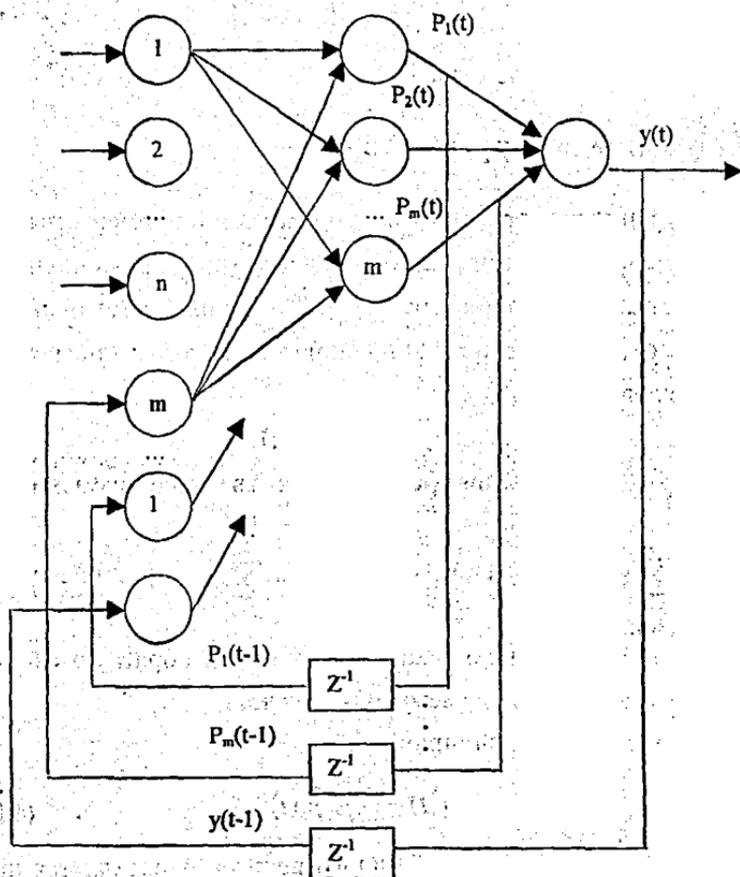


Рис.4.3. Архитектура рекуррентной сети с обратными связями от нейронов скрытого и выходного слоя

где p – количество нейронов выходного слоя.

При использовании рекуррентных нейронных сетей для прогнозирования временных рядов в выражения для взвешенной суммы (4.2), (4.4) или (4.6) необходимо подставить вместо $x_j(t)$ следующее значение

Глава 4. Многослойные нейронные сети с обратным распространением информации

$$\frac{dE}{dv_i} = \frac{dE}{dy} \frac{dy}{dv_i} = (y(t) - x(t+1)) p_i(t) \quad (4.12)$$

$$\frac{dE}{dT} = \frac{dE}{dy} \frac{dy}{dT} = -(y(t) - x(t+1)) \quad (4.13)$$

$$\frac{dE}{dw_{j\mu}} = \frac{dE}{dy} \frac{dy}{dp_i} \frac{dp_i}{dS_i} \frac{dS_i}{dw_{j\mu}} = (y(t) - x(t+1)) \cdot v_i p_i'(t) x(t-j+1) \quad (4.14)$$

$$\frac{dE}{dw_i} = \frac{dE}{dy} \frac{dy}{dp_i} \frac{dp_i}{dS_i} \frac{dS_i}{dw_i} = (y(t) - x(t+1)) \cdot v_i p_i'(t) y(t-1) \quad (4.15)$$

$$\frac{dE}{dw_{ii}} = \frac{dE}{dy} \frac{dy}{dp_i} \frac{dp_i}{dS_i} \frac{dS_i}{dw_{ii}} = (y(t) - x(t+1)) \cdot v_i p_i'(t) p_i(t-1) \quad (4.16)$$

$$\frac{dE}{dT_i} = \frac{dE}{dy} \frac{dy}{dp_i} \frac{dp_i}{dS_i} \frac{dS_i}{dT_i} = -(y(t) - x(t+1)) \cdot v_i p_i'(t) \quad (4.17)$$

Отсюда получаем выражения для настройки параметров обучения рекуррентной сети:

$$\Delta v_i(t+1) = -\alpha (y(t) - x(t+1)) \cdot p_i(t) \quad (4.18)$$

$$\Delta T(t) = \alpha (y(t) - x(t+1)) \quad (4.19)$$

$$\Delta w_{j\mu}(t+1) = -\alpha (y(t) - x(t+1)) \cdot v_i p_i'(t) x(t-j+1) \quad (4.20)$$

$$\Delta w_i(t+1) = -\alpha (y(t) - x(t+1)) \cdot v_i p_i'(t) y(t-1) \quad (4.21)$$

$$\Delta w_{ii}(t+1) = -\alpha (y(t) - x(t+1)) \cdot v_i p_i'(t) p_i(t-1) \quad (4.22)$$

$$\Delta T_i(t+1) = \alpha (y(t) - x(t+1)) \cdot v_i p_i'(t) \quad (4.23)$$

В качестве функции активации нейронных элементов может применяться

функция гиперболический тангенс или сигмоидная функция. В случае использования сигмоидной функции

$$p_i'(t) = \frac{\partial p_i(t)}{\partial S_i(t)} = p_i(t)(1 - p_i(t)) \quad (4.24)$$

При использовании функции гиперболический тангенс

$$p_i'(t) = (1 - p_i^2(t)) \quad (4.25)$$

Для ускорения процесса обучения можно использовать адаптивный шаг обучения. Тогда для выходного слоя нейронной сети

$$\alpha_1(t) = \frac{1}{1 + \sum_{i=1}^m p_i^2(t)} \quad (4.26)$$

Для обучения скрытого слоя рекуррентной нейронной сети можно использовать адаптивный или модифицированный адаптивный шаг обучения $\alpha_2(t)$ (глава 3). Рассмотрим применение модифицированного адаптивного шага обучения. Тогда в соответствии с разделом 3.8

$$C_i = \gamma_i F'(S_i) \left(1 + \sum_{j=1}^n x^2(t-j+1) + y^2(t-1) + \sum_I p_i^2(t-1) \right), \quad (4.27)$$

где ошибка i -го нейронного элемента скрытого слоя определяется следующим образом:

$$\gamma_i = (y(t) - x(t+1))v_i \quad (4.28)$$

Здесь $x(t+1)$ – значение временного ряда, на которое в данный момент времени производится обучение нейронной сети. Обозначим

$$R = \left(1 + \sum_{j=1}^n x^2(t-j+1) + y^2(t-1) + \sum_{I=1}^m p_i^2(t-1) \right) \quad (4.29)$$

Глава 4. Многослойные нейронные сети с обратным распространением информации

Тогда выражение (3.86) для модифицированного адаптивного шага обучения рекуррентной нейронной сети можно представить в следующем виде:

$$\alpha_2(t) = \frac{1}{F_2'(0)R^2 \sum_{i=1}^m v_i^2 F'(S_i)} \quad (4.30)$$

При использовании сигмоидной функции активации

$$F_2'(0) = \frac{1}{4} \quad (4.31)$$

$$F'(S_i) = p_i'(t) = p_i(t)(1 - p_i(t)) \quad (4.32)$$

Тогда

$$\alpha_2(t) = \frac{4}{R^2 \sum_{i=1}^m v_i^2 p_i(t)(1 - p_i(t))} \quad (4.33)$$

При использовании гиперболического тангенса в качестве функции активации нейронных элементов скрытого слоя

$$F_2'(0) = 1 \quad (4.34)$$

$$F'(S_i) = p_i'(t) = (1 - p_i^2(t)) \quad (4.35)$$

В этом случае выражение для адаптивного шага обучения определяется, как

$$\alpha_2(t) = \frac{1}{R^2 \sum_{i=1}^m v_i^2 (1 - p_i^2(t))} \quad (4.36)$$

Используя приведенные выше выражения можно ускорить процесс обучения

рекуррентной нейронной сети.

Алгоритм обучения рекуррентной нейронной сети в общем случае состоит из следующих шагов:

1. В начальный момент времени $t=1$ все контекстные нейроны устанавливаются в нулевое состояние, т.е. их выходные значения равняются нулю.

2. Входной образ подается на сеть и происходит прямое распространение его в нейронной сети.

3. В соответствии с алгоритмом обратного распространения ошибки производится модификация весовых коэффициентов и пороговых значений нейронных элементов.

4. Устанавливается $t = t + 1$ и осуществляется переход к пункту 2.

Обучение рекуррентной сети производится до тех пор, пока суммарная среднеквадратичная ошибка сети не станет меньше заданной.

4.2. Рециркуляционные нейронные сети

Рециркуляционные сети характеризуются как прямым $Y=f(X)$, так и обратным $X=f(Y)$ преобразованием информации. Задачей такого преобразования является достижение наилучшего автопрогноза или самовоспроизводимости вектора X . Рециркуляционные нейронные сети применяются для сжатия (прямое преобразование) и восстановления исходной (обратное преобразование) информации. Такие сети являются самоорганизующимися в процессе работы, где обучение производится без учителя. Они были предложены в 1988 году в работе [49]. Теоретической основой рециркуляционных нейронных сетей является *анализ главных компонент* (principal component analyse) [50-52].

4.2.1. Метод главных компонент

Метод главных компонент применяется в статистике для сжатия информации без существенных потерь ее информативности [50]. Он состоит в линейном ортогональном преобразовании входного вектора X размерности n в выходной вектор Y размерности p , где $p < n$. При этом компоненты вектора Y являются некоррелированными и общая дисперсия после преобразования остается неизменной. Совокупность входных паттернов представим в виде матрицы:

$$X = \begin{bmatrix} x_1^1 & x_2^1 & \dots & x_n^1 \\ x_1^2 & x_2^2 & \dots & x_n^2 \\ \dots & \dots & \dots & \dots \\ x_1^L & x_2^L & \dots & x_n^L \end{bmatrix},$$

где $X^k = (x_1^k, x_2^k, \dots, x_n^k)$ соответствует k -му входному образу, L – общее количество образов.

Будем считать, что матрица X является центрированной, то есть вектор математических ожиданий $\mu = 0$. Этого добиваются при помощи следующих преобразований:

$$x_j^i = x_j^i - \mu^j, \quad (4.37)$$

$$\mu^j = \sum_{i=1}^L x_j^i / L \quad (4.38)$$

Матрица ковариаций входных данных X определяется как

$$K = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \dots & \sigma_{1n} \\ \sigma_{21} & \sigma_{22} & \dots & \sigma_{2n} \\ \dots & \dots & \dots & \dots \\ \sigma_{n1} & \sigma_{n2} & \dots & \sigma_{nn} \end{bmatrix},$$

где σ_{ij} – ковариация между i -ой и j -ой компонентой входных образов.

Элементы матрицы ковариаций можно вычислить следующим образом:

$$\sigma_{ij} = \frac{1}{L} \sum_{k=1}^L (x_i^k - \mu^i)(x_j^k - \mu^j), \quad (4.39)$$

где $i, j = \overline{1, n}$.

Таким образом, на основе матрицы входных образов можно определить выборочную ковариационную матрицу. В дальнейшем изложении будем оперировать с входной информацией, представленной в виде вектора $X = (x_1, x_2, \dots, x_n)$.

Метод главных компонент состоит в нахождении таких линейных комбинаций исходных переменных

$$y_1 = w_{11}x_1 + w_{21}x_2 + \dots + w_{n1}x_n$$

$$y_2 = w_{12}x_1 + w_{22}x_2 + \dots + w_{n2}x_n$$

.....

$$y_p = w_{1p}x_1 + w_{2p}x_2 + \dots + w_{np}x_n,$$

что

$$\sigma(y_i, y_j) = 0; \quad i, j = \overline{1, n};$$

$$\sigma(y_1) > \sigma(y_2) > \dots > \sigma(y_p);$$

$$\sum_i^n \sigma_{ii} = \sum_i^n \sigma(y_i).$$

Из последних выражений следует, что переменные y_i некоррелированы, упорядочены по возрастанию дисперсии и сумма дисперсий входных образов остается без изменений. Тогда подмножество первых p переменных y характеризует большую часть общей дисперсии. В результате получается сжатое представление входной информации.

Переменные y_i , $i = \overline{1, p}$ называются главными компонентами. В матричной форме преобразование главных компонент можно представить как

$$Y = W^T X, \quad (4.40)$$

где строки матрицы W^T должны удовлетворять условию ортогональности, т.е.

$$\begin{aligned} W_i W_j^T &= 1, \forall i = j \\ W_i W_j^T &= 0, \forall i \neq j \end{aligned} \quad (4.41)$$

При этом вектор W_i определяется как

$$W_i = (w_{i1}, w_{i2}, \dots, w_{in})$$

Для определения главных компонент необходимо определить весовые коэффициенты W_i , $i = \overline{1, p}$.

В работе [50] показано, что каждая главная компонента получается как линейная комбинация $y_k = W_k X$, где W_k — собственный вектор ковариационной матрицы K , соответствующий k -му по величине собственному значению β_k этой матрицы. Для определения собственных значений β ковариационной матрицы, необходимо решить характеристическое уравнение:

$$\det(K - \beta I) = 0, \quad (4.42)$$

где I – единичная матрица.

Так как ковариационная матрица K является симметричной, то уравнение (4.42) имеет n вещественных корней:

$$\beta_1 > \beta_2 > \dots > \beta_n > 0. \quad (4.43)$$

Для определения первой главной компоненты, необходимо выбрать из p собственных значений матрицы K наибольшее (β_1) и решить следующую систему уравнений:

$$(K - \beta_1 I)W_1^T = 0, \quad (4.44)$$

где W_1^T – вектор столбец.

Из системы уравнений (4.44) определяется собственный вектор W_1 .

Как известно, собственные векторы действительной симметрической матрицы являются ортогональными. Для получения ортонормированного вектора W_1 необходимо пронормировать его:

$$W_1 = \left(\frac{w_{11}}{|W_1|}, \frac{w_{21}}{|W_1|}, \dots, \frac{w_{n1}}{|W_1|} \right), \quad (4.45)$$

где $|W_1| = \sqrt{w_{11}^2 + w_{21}^2 + \dots + w_{n1}^2}$.

В результате проведенных вычислений получается первая главная компонента $y_1 = W_1 X$, которая имеет максимальную дисперсию $\sigma(y_1)$. Аналогичным образом определяются остальные главные компоненты. При этом вторая компонента будет иметь следующую по величине дисперсию и так далее. Получаемая матрица весовых коэффициентов является ортогональной, т.е

$$W W^T = I. \quad (4.46)$$

Собственные числа β матрицы K характеризуют дисперсию главных компонент.

Глава 4. Многослойные нейронные сети с обратным распространением информации

При этом сумма дисперсий в пространстве исходных признаков равняется сумме дисперсий в пространстве выходных признаков:

$$\sum_{i=1}^n \sigma_{ii}(x) = \sum_{i=1}^n \beta_i \quad (4.47)$$

В работе [50] показано, что метод главных компонент имеет следующий критерий информативности:

$$J = \frac{\beta_1 + \beta_2 + \dots + \beta_p}{\beta_1 + \beta_2 + \dots + \beta_n} \quad (4.48)$$

Данный критерий позволяет ориентировочно определить число главных компонент p . Так, анализируя при помощи выражения (4.48) изменение J в зависимости от числа p , можно подобрать необходимое количество компонент без существенной потери информативности J .

Рассмотрим отображение выходного вектора Y во входной вектор X . Такое отображение называется автопрогнозом. Пусть

$$X = QY + e, \quad (4.49)$$

где e - дисперсия остатка, Q - матрица размерности $n \times p$.

В [52] приводится теорема, определяющая матрицу Q .

Теорема 4.1. Минимальное значение дисперсии остатка в выражении (4.49) достигается, когда столбцы матрицы Q равняются собственным векторам W_p , вычисленных в соответствии с методом главных компонент.

Таким образом

$$Q = W = \begin{bmatrix} w_{11} & w_{12} & \dots & w_{1p} \\ w_{21} & w_{22} & \dots & w_{2p} \\ \dots & \dots & \dots & \dots \\ w_{n1} & w_{n2} & \dots & w_{np} \end{bmatrix}$$

Тогда наилучший автопрогноз достигается, когда

$$X=WY \quad (4.50)$$

Величина абсолютной ошибки прогноза выражается через собственные числа ковариационной матрицы.

$$\Delta = \sum_{i=1}^{n-p} \beta_{p+i} \quad (4.51)$$

Относительная ошибка определяется, как

$$\delta = \frac{\Delta}{\sum_{i=1}^n \beta_i} 100\% \quad (4.52)$$

Метод главных компонент является эффективным средством для сжатия и восстановления данных [50].

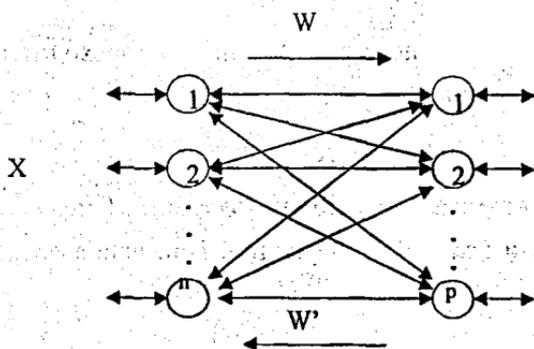
4.2.2. Архитектура рециркуляционной нейронной сети

Рециркуляционная нейронная сеть представляет собой совокупность двух слоев нейронных элементов, которые соединены между собой двунаправленными связями (рис. 4.4).

Каждый из слоев нейронных элементов может использоваться в качестве входного или выходного. Если слой нейронных элементов служит в качестве входного, то он выполняет распределительные функции.

В противном случае нейронные элементы слоя являются обрабатывающими. Весовые коэффициенты соответствующие прямым и обратным связям характеризуются матрицей весовых коэффициентов W и W' . Для наглядности, рециркуляционную сеть можно представить в развернутом виде, как показано на рис. 4.5.

Такое представление сети является эквивалентным и характеризует



ис. 4.4. Архитектура рециркуляционной нейронной сети

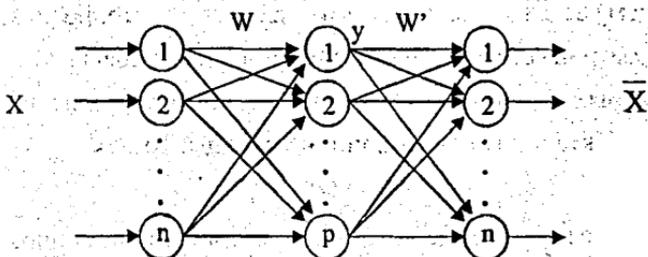


Рис. 4.5. Эквивалентное представление рециркуляционной сети

полный цикл преобразования информации. При этом промежуточный слой нейронных элементов производит кодирование (сжатие) входных данных X , а последний слой осуществляет восстановление сжатой информации Y . Назовем слой нейронной сети, соответствующий матрице связи W прямым, а соответствующий матрице связей W' – обратным.

Рециркуляционная сеть предназначена как для сжатия данных, так и для восстановления сжатой информации. Сжатие данных осуществляется при прямом преобразовании информации в соответствие с выражением:

$$Y = F(W^T X). \quad (4.53)$$

Восстановление или реконструкция данных происходит при обратном преобразовании информации:

$$\bar{X} = F(W' Y). \quad (4.54)$$

В качестве функции активации нейронных элементов F может использоваться как линейная, так и нелинейная функции. При использовании линейной функции активации:

$$Y = W^T X, \quad (4.55)$$

$$\bar{X} = W' Y. \quad (4.56)$$

В предыдущем разделе отмечалось, что наилучший автопрогноз достигается тогда, когда матрица весовых коэффициентов сформирована в соответствии с методом главных компонент. При этом столбцы матрицы W' равняются собственным векторам ковариационной матрицы. Тогда:

$$W' \equiv W,$$

Таким образом весовые коэффициенты линейной рециркуляционной нейронной сети можно определить при помощи метода главных компонент. В этом случае матрица W является ортогональной и

$$W W^T = 1.$$

Линейные рециркуляционные сети, в которых весовые коэффициенты определяются в соответствии с методом главных компонент называются PCA сетями [42]. Рассмотрим другие методы обучения рециркуляционных нейронных сетей.

4.2.3. Алгоритм обратного распространения ошибки

Рециркуляционные нейронные сети должны обеспечивать такое

Глава 4. Многослойные нейронные сети с обратным распространением информации

преобразование информации, чтобы достигалась минимальная среднеквадратичная ошибка между входным X и реконструированным образами:

$$E = \frac{1}{2} \sum_{i=1}^n (\bar{x}_i - x_i)^2, \quad (4.57)$$

где \bar{x}_i – i -ая компонента реконструированного образа. Она определяется следующим образом:

$$\bar{x}_i = \sum_{j=1}^p w_{ji} y_j. \quad (4.58)$$

В работе [50] показано, что функция (4.57) достигает минимума, если в качестве y_j ($j = \overline{1, p}$) использовать первые p главных компоненты вектора X . Это является теоретической основой для использования алгоритма обратного распространения ошибки, который минимизирует среднеквадратичную ошибку методом градиентного спуска. Существуют различные подходы в применении алгоритма обратного распространения ошибки для обучения рекуррентных сетей. Так Коттрел (Cottrell) для сжатия изображений использовал стандартный алгоритм обратного распространения [53, 54]. Хинтон (Hinton) применял кумулятивное дельта правило (Cumulative Delta Rule), которое представляет собой незначительную модификацию стандартного backpropagation [49]. Рассмотрим применение кумулятивного дельта правила для обучения рециркуляционных нейронных сетей.

В процессе обучения рециркуляционной сети для каждого входного образа производится три цикла распространения информации: прямое, обратное и прямое. После этого производится настройка весовых коэффициентов сети. Для наглядности процесса распространения

информации введем обозначения. Пусть $x_i(0)$, входной вектор поступающий на вход сети в начальный момент времени. Тогда выходной вектор сети в момент $t = 1$ определяется в результате прямого преобразования информации:

$$y_j(1) = \sum_{i=1}^n w_{ij} x_i(0), \quad (4.59)$$

где $j = \overline{1, p}$.

Вектор, получается в результате обратного преобразования вектора $Y(1)$:

$$\bar{x}_i(2) = \sum_{j=1}^p w'_{ji} y_j(1), \quad (4.60)$$

где $j = \overline{1, n}$.

На третьем этапе распространения информации определяется вектор $\bar{Y}(3)$:

$$\bar{y}_j(3) = \sum_{i=1}^n w_{ij} \bar{x}_i(2), \quad (4.61)$$

где $j = \overline{1, p}$.

Такое преобразование информации можно представить в виде цепочки, изображенной на рис. 4.6.

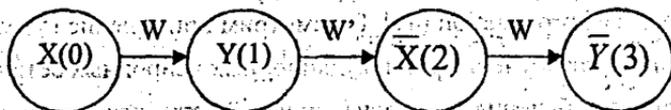


Рис. 4.6. Последовательное преобразование информации в рециркуляционной сети

Глава 4. Многослойные нейронные сети с обратным распространением информации

Тогда ошибка восстановления информации в первом слое нейронной сети определяется как:

$$E = \frac{1}{2} \sum_i (\bar{x}_i(2) - x_i(0))^2 \quad (4.62)$$

Ошибку воспроизведения информации во втором слое нейронной сети можно представить следующим образом:

$$E' = \frac{1}{2} \sum_j (\bar{y}_j(3) - y_j(1))^2 \quad (4.63)$$

Обучение рециркуляционной нейронной сети производится как с целью минимизации ошибки E , так и E' . При этом значение $y_j(1)$ в выражение (4.63) принимается за эталонное. Тогда в соответствии с методом градиентного спуска в пространстве весовых коэффициентов:

$$w_{ij}(t+1) = w_{ij}(t) - \alpha \frac{\partial E}{\partial w_{ij}(t)} \quad (4.64)$$

$$w'_{jk}(t+1) = w'_{jk}(t) - \alpha \frac{\partial E'}{\partial w'_{jk}(t)} \quad (4.65)$$

Определим производные для линейной рециркуляционной сети. Тогда:

$$\frac{\partial E}{\partial w'_{ij}(t)} = (\bar{y}_j(3) - y_j(1)) x_i(2) \quad (4.66)$$

$$\frac{\partial E}{\partial w'_{jk}(t)} = (\bar{x}_i(2) - x_i(0)) y_j(1) \quad (4.67)$$

В результате, выражения для настройки весовых коэффициентов рециркуляционной нейронной сети примут следующий вид:

$$w_{ij}(t+1) = w_{ij}(t) - \alpha \bar{x}_i(2)(\bar{y}_j(3) - y_j(1)), \quad (4.68)$$

$$w'_{ji}(t+1) = w'_{ji}(t) - \alpha y_j(1)(\bar{x}_i(2) - x_i(0)). \quad (4.69)$$

Для получения ортонормированных весовых векторов w_k для каждого нейрона необходимо ввести нормированное правило обучения. Пусть $W_k = (w_{1k}, w_{2k}, \dots, w_{nk})$ – весовой вектор k -го нейронного элемента. Тогда длина его в момент времени $t+1$ равняется:

$$|W_k(t+1)| = \sqrt{w_{1k}^2(t+1) + w_{2k}^2(t+1) + \dots + w_{nk}^2(t+1)} \quad (4.70)$$

В соответствии с этим нормированное правило обучения для весовых коэффициентов k -го нейрона можно представить следующим образом:

$$w_{ik}(t+1) = \frac{w_{ik}(t) - \alpha x_i(2)(y_k(3) - y_k(1))}{|W_k(t+1)|} \quad (4.71)$$

Аналогично производится формирование весовых коэффициентов W' . Как уже отмечалось в процессе обучения рециркуляционной сети для каждого входного образа происходит три цикла распространения информации. После этого осуществляется модификация весовых коэффициентов сети. Процедура обучения осуществляется до тех пор, пока суммарная среднеквадратичная ошибка сети не станет меньше заданной.

В начальный момент времени производится случайная инициализация весовых коэффициентов. Хинтон рекомендовал, чтобы веса инициализировались с нулевым средним [49]. При этом желательно обеспечивать симметричность весовых коэффициентов прямого и обратного слоя ($w_{ij} = w'_{ji}$). Точно такие же выражения, (4.68) и (4.69), Хинтон использовал для модификации синаптических связей нелинейных рециркуляционных сетей, хотя с формальной точки зрения это является не

Глава 4. Многослойные нейронные сети с обратным распространением информации

совсем верным. Критическим параметром кумулятивного дельта правила является выбор подходящего шага обучения α , чтобы обеспечить достижение минимальной среднеквадратичной ошибки. При этом для различных задач требуются разные значения шага [49]. Для устранения этих недостатков можно использовать адаптивный шаг обучения. Тогда при настройке весовых коэффициентов прямого слоя величина скорости обучения определяется, как

$$\alpha(w) = \frac{1}{\sum_{i=1}^n x_i^2} \quad (4.72)$$

При модификации весовых коэффициентов обратного слоя w' значение адаптивного шага равняется:

$$\alpha(w') = \frac{1}{\sum_{j=1}^p y_j^2} \quad (4.73)$$

Технология вывода приведенных выше выражений является аналогичной той, которая была рассмотрена в разделе 2.8. Приведенный алгоритм характеризуется неустойчивостью процесса обучения. Поэтому можно предложить алгоритм послойной модификации синаптических связей, который более адекватно отражает задачи решаемые рециркуляционной сетью.

4.2.4. Алгоритм послойного обучения

Характеризуется последовательной модификацией весовых коэффициентов различных слоев нейронной сети, и состоит из двух этапов.

На первом этапе происходит определение весовых коэффициентов W' и значений главных компонент \bar{Y} для осуществления наилучшего автопрогноза. Для этого необходимо обеспечить:

$$\|X - W' \bar{Y}\|^2 \rightarrow \min, \quad (4.74)$$

Это эквивалентно минимизации среднеквадратичной ошибки:

$$E_s = \frac{1}{2} \sum_{k=1}^L \sum_{i=1}^n (x_i^k - \bar{x}_i^k)^2. \quad (4.75)$$

На втором этапе осуществляется настройка весовых коэффициентов прямого слоя. При этом в качестве эталонных выходов используются значения \bar{Y} , полученные на предыдущем этапе алгоритма. Тогда обучение на втором этапе происходит с целью минимизации следующего выражения:

$$E'_s = \frac{1}{2} \sum_{k=1}^L \sum_{i=1}^n (y_i^k - \bar{y}_i^k)^2. \quad (4.76)$$

В процессе обучения здесь, в отличие от кумулятивного дельта правила, происходит два цикла распространения информации: прямое и обратное. Рассмотрим алгоритм послойного обучения для линейных и нелинейных рециркуляционных сетей:

4.2.4.1. Линейные рециркуляционные сети

Определим выражение для настройки весовых коэффициентов W' и значений \bar{Y} на первом этапе алгоритма. Тогда для каждого входного вектора X необходимо определить такой эталонный вектор \bar{Y} , чтобы обеспечить

Глава 4. Многослойные нейронные сети с обратным распространением информации

минимизацию выражения (4.75). Для этого будем использовать метод градиентного спуска. Тогда:

$$y_j'(t+1) = y_j'(t) - \alpha_j(t) \frac{\partial E}{\partial y_j(t)}, \quad (4.77)$$

где

$$E = \frac{1}{2} \sum_{i=1}^n (\bar{x}_i - x_i)^2,$$

$$\bar{x}_i = \sum_j w_{ji} y_j.$$

Найдем производную:

$$\frac{\partial E}{\partial y_j} = \frac{\partial E}{\partial \bar{x}_i} \frac{\partial \bar{x}_i}{\partial y_j} = \sum_i w_{ji} (\bar{x}_i - x_i) = \gamma_j. \quad (4.78)$$

Для нахождения адаптивного шага обучения $\alpha_j(t)$ будем использовать метод наискорейшего спуска.

Теорема 4.2. Значение адаптивного шага обучения $\alpha_j(t)$ для модификации состояния j -го выходного нейрона прямого слоя с целью минимизации среднеквадратичной ошибки определяется на основе следующего выражения:

$$\alpha_j(t) = \frac{1}{\sum (w_{ji}(t))^2}. \quad (4.79)$$

Доказательство. Для нахождения адаптивного шага обучения используем метод наискорейшего спуска. Тогда:

$$\alpha_j(t) = \min \left\{ E(y_j(t) - \alpha_j(t) \frac{\partial E}{\partial y_j(t)}) \right\}. \quad (4.80)$$

Определим выходное значение i -го нейрона обратного слоя, как:

$$\bar{x}_i(t+1) = w'_{ji}(y_j(t) - \alpha_j \gamma_j) + \sum_{k \neq j} w_{ki} y_k(t). \quad (4.81)$$

После преобразования последнего выражения получим:

$$\bar{x}_i(t+1) = \bar{x}_i(t) - \alpha_j \gamma_j w'_{ji}. \quad (4.82)$$

Определим среднеквадратичную ошибку сети, как

$$E = \frac{1}{2} \sum_{i=1}^n (\bar{x}_i(t+1) - x_i)^2. \quad (4.83)$$

Находим такое a_j , при котором среднеквадратичная ошибка сети является минимальной:

$$\frac{\partial E}{\partial \alpha_j} = \sum_i (\bar{x}_i(t+1) - x_i - \alpha_j \gamma_j w'_{ji})(-\gamma_j w'_{ji}) = 0. \quad (4.84)$$

Преобразуя последнее выражение, получим значение адаптивного шага обучения:

$$\alpha_j(t) = \frac{1}{\sum_{i=1}^n (w'_{ji}(t))^2}. \quad (4.85)$$

Так как $\frac{\partial^2 E}{\partial \alpha_j^2} > 0$, то при данном значении a_j обеспечивается минимум среднеквадратичной ошибки.

Теорема доказана.

Тогда выражение (4.77) можно представить следующим образом:

$$y_j(t+1) = y_j(t) - \frac{\gamma_j}{\sum_{j=1}^n (w'_{\mu}(t))^2} \quad (4.86)$$

где

$$\gamma_j = \sum_{i=1}^n w'_{\mu}(t)(\bar{x}_i - x_i) \quad (4.87)$$

Определим теперь выражения для настройки весовых коэффициентов W' . Тогда в соответствии с методом градиентного спуска:

$$w'_{\mu}(t+1) = w'_{\mu}(t) - \alpha(t)(\bar{x}_i - x_i)y_j \quad (4.88)$$

где

$$\alpha(t) = \frac{1}{\sum_{j=1}^p y_j^2(t)} \quad (4.89)$$

Таким образом на первом этапе алгоритма происходит определение весовых коэффициентов обратного слоя W_{μ} и p – первых главных компонент

\bar{y}_j .

На втором этапе происходит определение весовых коэффициентов прямого слоя, где в качестве эталонных выходов используются значения \bar{Y} .

Для этого необходимо минимизировать среднеквадратичную ошибку:

$$E = \frac{1}{2} \sum_{j=1}^p (y_j - \bar{y}_j)^2$$

где главные компоненты \bar{y}_j выступают в качестве эталонных значений.

Выходное значение j -го нейрона второго слоя определяется следующим образом:

$$y_j = \sum_{i=1}^n w_{ij} x_i.$$

Тогда используя метод градиентного спуска получим

$$w_{ij}(t+1) = w_{ij}(t) - \alpha(t)(y_j - \bar{y}_j)x_i, \quad (4.90)$$

где

$$\alpha(t) = \frac{1}{\sum_{i=1}^n x_i^2(t)}. \quad (4.91)$$

По аналогии с разделом 4.2.3 можно использовать нормированное правило обучения для настройки весовых коэффициентов. Алгоритм послонного обучения можно представить в виде следующей последовательности шагов:

1. Случайная инициализация весовых коэффициентов нейронной сети, и задание минимальной среднеквадратичной ошибки E_m .
2. Последовательно подаются L образов на нейронную сеть. При этом для каждого образа происходит прямое и обратное распространение информации. В результате этого, для обратного слоя сети осуществляется модификация весовых коэффициентов W' в соответствии с выражением (4.88). Одновременно для каждого образа происходит настройка входов Y обратного слоя согласно выражению (4.86).

Глава 4. Многослойные нейронные сети с обратным распространением информации

3. Рассматривается только обратный слой сети. В качестве входной информации используются значения Y , определенные на предыдущем шаге алгоритма. В качестве эталонных значений выходов обратного слоя принимается вектор исходных данных X . При этом производится следующая последовательность действий:

4. Для L входных образов производится настройка весовых коэффициентов W' и желаемых выходов U по выражениям (4.88) и (4.86).

5. Пункт 3.1 повторяется, пока суммарная среднеквадратичная ошибка обратного слоя не станет меньше заданной E_m .

6. Производится модификация весовых коэффициентов прямого слоя в соответствии с выражением (4.90). Для этого входные образы последовательно подаются на сеть и для каждого образа происходит только прямое распространение информации. В качестве эталонных данных используются значения \bar{Y} , полученные на предыдущем шаге алгоритма. Пункт 4 продолжается до тех пор, пока суммарная среднеквадратичная ошибка прямого слоя не станет меньше заданной.

4.2.4.2. Нелинейные рециркуляционные сети

В качестве функции активации нейронных элементов в таких сетях используется нелинейная функция. В этом случае происходит нелинейное отображение входного пространства X в пространство меньшей размерности. Для многих ситуаций данное отображение позволяет добиться более точного преобразования, в результате которого уменьшается суммарное искажение геометрической конфигурации данных по сравнению с линейным [50]. В качестве функции активации нейронных элементов обычно используется сигмоидная функция или гиперболический тангенс. Алгоритм послыного обучения остается при этом неизменным, меняются только выражения для

настройки параметров рециркуляционной нейронной сети. Определим их в случае использования в качестве функции активации нейронных элементов гиперболического тангенса. Тогда выходное значение нейронных элементов прямого слоя определяется, как:

$$y_j = th(S_j), \quad (4.92)$$

$$S_j = \sum_{i=1}^n w_{ji} x_i, \quad (4.93)$$

где $j = \overline{1, p}$.

Соответственно для обратного слоя рециркуляционной сети:

$$x_i = th(S_i), \quad (4.94)$$

$$S_i = \sum_{j=1}^p w'_{ji} y_j, \quad (4.95)$$

тогда:

$$\frac{\partial E}{\partial y_j} = \frac{\partial E}{\partial x_i} \frac{\partial x_i}{\partial S_i} \frac{\partial S_i}{\partial y_j} = \sum_i (\bar{x}_i - x_i)(1 - \bar{x}_i^2) w'_{ji}.$$

Обозначим:

$$\gamma_j = \sum_i (\bar{x}_i - x_i)(1 - \bar{x}_i^2) w'_{ji}. \quad (4.96)$$

Выражение для адаптивного шага обучения α_j можно получить используя метод наискорейшего спуска и разложение в ряд Тейлора. В результате:

$$\alpha_j = \frac{\sum_{i=1}^n (\bar{x}_i - x_i) w'_{ji}}{\gamma_j \sum_{i=1}^n (w'_{ji})^2} \quad (4.97)$$

Тогда выражение для настройки входов у обратного слоя можно представить в следующем виде:

$$y_j(t+1) = y_j(t) - \alpha_j \gamma_j \quad (4.98)$$

Определим выражение для настройки весовых коэффициентов W' . Тогда:

$$\frac{\partial E}{\partial w'_{ji}} = \frac{\partial E}{\partial x_i} \frac{\partial \bar{x}_i}{\partial S_i} \frac{\partial S_i}{\partial w'_{ji}} = (\bar{x}_i - x_i)(1 - \bar{x}_i^2) y_j \quad (4.99)$$

Скорость обучения получается также, как в разделе 3.7.

$$\alpha(t) = \frac{\sum_{i=1}^n (\bar{x}_i - x_i)^2 (1 - \bar{x}_i^2)}{\left(\sum_{j=1}^p y_j^2 \right) \sum_{i=1}^n (\bar{x}_i - x_i)^2 (1 - \bar{x}_i^2)^2} \quad (4.100)$$

Тогда модификация синаптических связей W' происходит следующим образом:

$$w'_{ji}(t+1) = w'_{ji}(t) - \alpha(t) (\bar{x}_i - x_i) (1 - \bar{x}_i^2) y_j \quad (4.101)$$

Аналогичным образом получается выражение для настройки весовых коэффициентов прямого слоя. Так как:

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial y_j} \frac{\partial y_j}{\partial S_j} \frac{\partial S_j}{\partial w_{ij}} = (y_j - \bar{y}_j) (1 - y_j^2) x_i \quad (4.102)$$

$$\alpha(t) = \frac{\sum_{j=1}^p (y_j - \bar{y}_j)^2 (1 - y_j^2)}{(\sum_{i=1}^n x_i^2) \sum_{j=1}^p (y_j - \bar{y}_j)^2 (1 - y_j^2)^2}, \quad (4.103)$$

то

$$w_{ij}(t+1) = w_{ij}(t) - \alpha(t)(y_j - \bar{y}_j)(1 - y_j^2)x_i, \quad (4.104)$$

Используя приведенные выше выражения и алгоритм послыонного обучения можно настраивать нелинейную рециркуляционную сеть для сжатия данных.

4.2.5. Нормализованное правило Хебба

При нормировании весовых коэффициентов в процессе обучения нейронной сети можно получить нормализованное правило Хебба.

В этом случае

$$w_{ij}(t+1) = \frac{w_{ij}(t) + \alpha x_i(t) y_j(t)}{|W_j(t+1)|}. \quad (4.105)$$

В работе [42] показано, что данное правило может быть аппроксимировано путем разложения в ряд Тейлора следующим выражением:

$$w_{ij}(t+1) = w_{ij}(t) + \alpha y_j(t) (x_i(t) - y_j(t) w_{ij}(t)). \quad (4.106)$$

Последнее выражение называется "Ожа" правило ("Oja" learning rule). Первый терм в (4.106) $y_j(t)x_i(t)$ представляет собой стандартное правило Хебба, а второй терм $y_j(t)y_j(t)w_{ij}(t)$ предназначен для нормализации весовых коэффициентов. В работе [42] показано, что при использовании нормализованного правила Хебба можно получить собственные вектора

ковариационной матрицы входных данных. В результате происходит определение главных компонент. Поэтому правило Хебба может использоваться для обучения рециркуляционных нейронных сетей.

4.2.6. Обработка изображений

Рециркуляционные нейронные сети можно применять для сжатия и восстановления изображений. Для упрощения процедуры обработки, изображение размерности $n \times n$ можно разделить на множество блоков размерности $p \times p$, где $p < n$ (рис 4.7). При этом количество таких блоков равняется

$$k = \left(\frac{n}{p}\right)^2$$

Назовем блок размерностью $p \times p$ окном и поставим окну в соответствие рециркуляционную нейронную сеть. Количество нейронов первого слоя такой сети соответствует размерности окна и равняется $p \times p$, число нейронов второго слоя обозначим через r . Сканируя изображение при помощи окна и подавая его на нейронную сеть, можно сжать входное изображение.

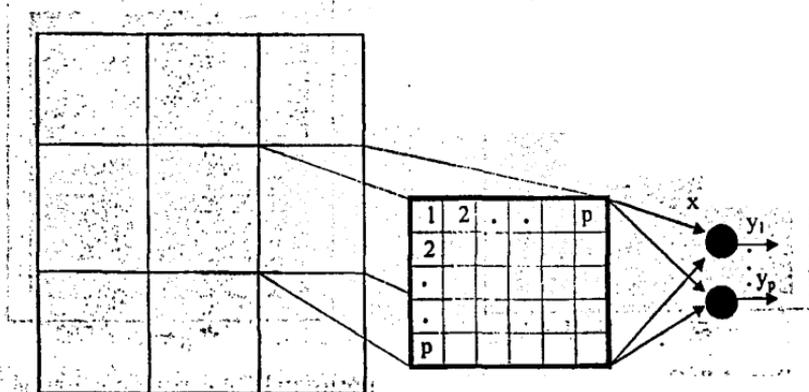


Рис. 4.7. Разбиение исходного изображения на блоки размерности $p \times p$

Такой подход использовал Сангер (Sanger) в 1989 году для сжатия изображений [42, 36]. В качестве рециркуляционной сети он применял PCA сеть, в которой весовые коэффициенты нейронов прямого слоя соответствовали собственным векторам ковариационной матрицы. В качестве теста использовалось изображение, представленное на рис.4.8. Размерность его составляет 256×256 пикселей, где каждый пиксель содержит 8 бит. Размерность окна равняется 8×8 . PCA сеть состоит из 64-х нейронов первого и восьми нейронов второго слоя. На рис. 4.8 показано тестовое изображение лица, а на рис.4.9 – весовые коэффициенты восьми нейронов второго слоя [42].

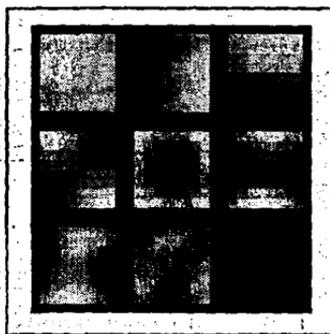
Более темным линиям на рис.4.9 соответствует большая величина синаптических связей. Сжатое изображение может восстанавливаться при помощи обратного распространения информации.

Мы рассмотрим применение линейной PCA сети для обработки изображений. Аналогичные эксперименты проводились при использовании алгоритма обратного распространения ошибки [54].

Рассмотрим сравнительный анализ различных алгоритмов обучения при



а



б

Рис. 4.8. Сжатие изображения при помощи PCA-сети: а-исходное изображение; б-весовые коэффициенты нейронов сети.

Глава 4. Многослойные нейронные сети с обратным распространением информации

помощи обратного распространения информации.

Мы рассмотрим применение линейной РСА-сети для обработки изображений. Аналогичные эксперименты проводились при использовании алгоритма обратного распространения ошибки [54].

Рассмотрим сравнительный анализ различных алгоритмов обучения при использовании нелинейной рециркуляционной сети для обработки изображений. В качестве функции активации нейронных элементов будем использовать функцию гиперболического тангенса. В качестве теста будем использовать изображение, приведенное на рис. 4.9. Размерность его составляет 256×256 пикселей, где каждый пиксель содержит 24 бита. Для обработки данного изображения будем использовать рециркуляционную сеть нейронную сеть, состоящую из 12 нейронов первого и 6 нейронов второго слоя. На каждый вход нейронной сети при этом подается 1 байт информации (вещественное нормированное число). Размерность обучающей выборки составила 16384 образца. Графики изменения суммарной среднеквадратичной ошибки в процессе обучения для различных алгоритмов



Рис. 4.9. Фрагменты тестового изображения (слева) и изображения после реконструкции (справа).

представлены на рис. 4.10. на основании графиков можно судить о том, что алгоритм послойного обучения имеет минимальную временную сложность. Эксперименты также показали, что алгоритм послойного обучения является более стабильным по сравнению с обычными. Так при любой инициализации весовых коэффициентов нейронная сеть обучалась до минимальной ошибки. Фрагменты тестового изображения и изображения после реконструкции представлены на рис. 4.9. В данном разделе рассмотрено применение рециркуляционных нейронных сетей для обработки изображений. Наиболее эффективным является алгоритм послойного обучения рециркуляционной сети.

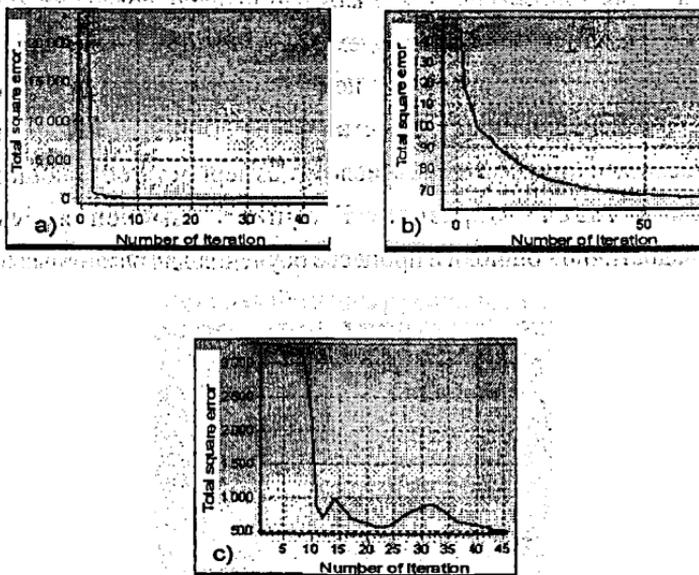


Рис. 4.10. Графики изменения суммарной среднеквадратичной ошибки обучения: а) послойное обучение, б) алгоритм обратного распространения, в) кумулятивное дельта-правило.

ГЛАВА 5. РЕЛАКСАЦИОННЫЕ НЕЙРОННЫЕ СЕТИ

Релаксационные нейронные сети характеризуются прямым и обратным распространением информации между слоями нейронной сети. В основе функционирования таких сетей лежит итеративный принцип работы. Он заключается в том, что на каждой итерации процесса происходит обработка данных, полученных на предыдущем шаге. Такая циркуляция информации продолжается до тех пор, пока не установится состояние равновесия. При этом состоянии нейронных элементов перестают изменяться и характеризуются стационарными значениями. Для анализа устойчивости релаксационных нейронных сетей используются *функции Ляпунова*. Такие сети применяются в качестве ассоциативной памяти и для решения комбинаторных задач оптимизации. К релаксационным сетям относится нейронная сеть Хопфилда [23, 55-57], Хэмминга [33], двунаправленная ассоциативная память [58-59] и машина Больцмана [60]. В главе рассмотрены архитектура и функционирование таких нейронных сетей. Для сети Хопфилда приводится механизм получения функции Ляпунова и анализ устойчивости.

5.1. Устойчивость динамических систем

Для исследования устойчивости нелинейных динамических систем используется второй метод Ляпунова [61-64]. Рассмотрим динамическую систему, которая описывается нелинейными дифференциальными уравнениями:

$$\dot{X} = f(X), \quad X(0) = X_0. \quad (5.1)$$

Здесь \dot{X} и $f(X)$ являются n -мерными векторами:

$$\dot{X}(t) = \begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \vdots \\ \dot{x}_n(t) \end{bmatrix}, \quad f(X) = \begin{bmatrix} f_1(x_1, x_2, \dots, x_n) \\ f_2(x_1, x_2, \dots, x_n) \\ \vdots \\ f_n(x_1, x_2, \dots, x_n) \end{bmatrix}$$

Совокупность величин x_1, x_2, \dots, x_n называется *фазовыми координатами* точки в n -мерном пространстве. С течением времени фазовые координаты будут меняться, что приведет к соответствующему перемещению точки. Такая точка называется *изображающей*, ее траектория — *фазовой траекторией*, а n -мерное пространство — *фазовым пространством*. Фазовая траектория является наглядной геометрической иллюстрацией динамического поведения системы в пространстве.

Второй метод Ляпунова [61] базируется на использовании совместно с уравнениями движения системы специальной функции, которая называется функцией Ляпунова. В качестве функции Ляпунова используется функция фазовых координат $E(x_1, x_2, \dots, x_n)$, обладающая определенными свойствами [62].

Назовем функцию $E(x)$ *знакопостоянной*, если в некоторой области фазового пространства она может принимать значения только одного определенного знака или обращаться в ноль. Знакопостоянная функция называется *знакоопределенной*, если она обращается в ноль только при $x_1 = x_2 = \dots = x_n = 0$. Если функция $E(x)$ является знакоопределенной,

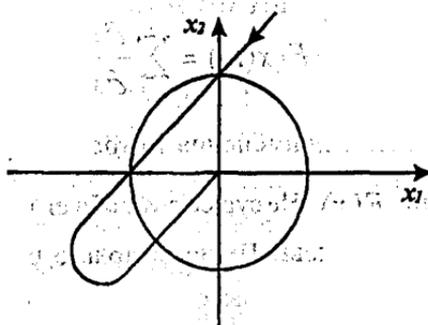


Рис. 5.1. Фазовая траектория устойчивой системы

то уравнение $E(x) = C = const$ характеризует замкнутую поверхность, охватывающую начало координат. При убывании функции $E(x)$, т. е.

производная $\frac{dE}{dt} < 0$, фазовая траектория пересекает поверхность

$E(x) = C$ в направлении начала координат (рис. 5.1).

Это соответствует устойчивой системе. Существуют различные вариации второго метода Ляпунова [62-64]. Приведем здесь только один вариант, который рассмотрен в [63].

Теорема 5.1. Если существует такая знакоопределенная функция $E(x)$, полная производная по времени которой в силу дифференциальных уравнений движения является знакопостоянной или тождественно равняется нулю, то динамическая система является устойчивой.

Полная производная по времени функции $E(x)$ определяется следующим образом:

$$\frac{d}{dt} E(x(t)) = \sum_{i=1}^n \frac{\partial E}{\partial x_i} \cdot \frac{dx_i}{dt} \quad (5.2)$$

Основная проблема применения метода Ляпунова заключается в определении функции $E(x)$. Не существует общего аналитического метода построения функций Ляпунова. Имеются только рекомендации. Так, для линейных систем функция Ляпунова строится в виде квадратичной формы координат [64]:

$$E(X, X) = \sum_i \sum_k a_{ik} x_i x_k = X^T A X, \quad (5.3)$$

где $a_{ik} = a_{ki}$ и соответственно A — симметрическая матрица.

Если система содержит множество нелинейностей $f_i(\sigma)$, то функцию Ляпунова рекомендуется [64] строить в виде суммы квадратичной формы и интеграла от нелинейности:

$$E = X^T A X + \sum_i b \int_0^z f_i(\sigma) d\sigma, \quad (5.4)$$

где b — число.

Функции Ляпунова служат хорошим инструментом для исследования различных динамических систем. Они применяются для определения устойчивости релаксационных нейронных сетей, которые будут рассмотрены в следующих разделах.

5.2. Нейронная сеть Хопфилда

Представляет собой нейронную сеть с обратными связями. Функционирование таких сетей характеризуется релаксационным процессом обработки информации, который происходит до тех пор, пока не установится состояние равновесия. Многие исследователи описывали аналогичные нейронные сети, например, Андерсон [20] (Anderson, 1977) и Литтл (Little, 1978) [36]. В 1982 г. американский биофизик Хопфилд (Hopfield) представил математический анализ релаксационных сетей с обратными связями [23]. Он базировался на теории изинговых спинов, которая используется для изучения ферромагнетиков при низких температурах. Поэтому такие нейронные сети получили название сетей Хопфилда.

5.2.1 Архитектура нейронной сети Хопфилда

Нейронная сеть Хопфилда характеризуется обратными связями. В ней каждый нейрон имеет синаптические связи со всеми остальными нейронами сети. Такую модель можно представить в виде графа (рис. 5.2):

Архитектуру такой сети будем представлять в виде двух слоев

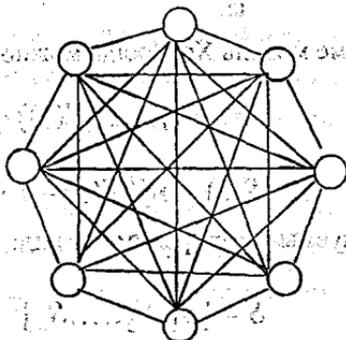


Рис. 5.2. Граф-модель сети Хопфилда

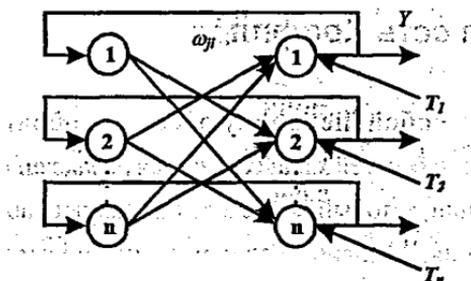


Рис 5.3. Архитектура сети Хопфилда

нейронных элементов (рис.5.3).

При этом первый слой является распределительным, а второй слой нейронных элементов осуществляет нелинейное преобразование взвешенной суммы:

$$y_i(t+1) = F\left(\sum_{j=1}^n \omega_{ji} y_j(t) - T_i\right), \quad (5.5)$$

где $y_i(t+1)$ — выходное значение i -го, нейронного элемента в момент времени $t+1$; F — оператор нелинейного преобразования; T_i — пороговое значение i -го нейрона.

В матричной форме модель Хопфилда можно представить, как

$$Y(t+1) = F(S(t)), \quad (5.6)$$

$$S(t) = W^T Y(t) - T.$$

При этом используемые векторы имеют вид:

$$S = [S_1, S_2, \dots, S_n]^t,$$

$$Y = [y_1, y_2, \dots, y_n]^T,$$

$$T = [T_1, T_2, \dots, T_n]^T,$$

$$W = \begin{bmatrix} \omega_{11} & \omega_{12} & \dots & \omega_{1n} \\ \omega_{21} & \omega_{22} & \dots & \omega_{2n} \\ \dots & \dots & \dots & \dots \\ \omega_{n1} & \omega_{n2} & \dots & \omega_{nn} \end{bmatrix}. \quad (5.7)$$

В качестве матрицы весовых коэффициентов Хопфилд использовал симметричную матрицу ($\omega_{ij} = \omega_{ji}$) с нулевой главной диагональю ($\omega_{ii} = 0$). Последнее условие соответствует отсутствию обратной связи нейронного элемента на себя. В качестве функции активации нейронных элементов F может использоваться как пороговая, так и непрерывная функция, например, сигмоидная или гиперболический тангенс.

Будем рассматривать нейронную сеть Хопфилда с дискретным временем. Тогда при использовании пороговой функции активации она называется нейронной сетью с *дискретным состоянием и временем*. Нейронная сеть с непрерывной функцией активации называется нейронной сетью с *непрерывным состоянием и дискретным временем*. При использовании непрерывного времени модель Хопфилда называется *непрерывной*.

Для описания функционирования таких сетей Хопфилд использовал аппарат статистической физики. При этом каждый нейрон имеет два состояния активности (1, -1), которые аналогичны значениям спина

некоторой частицы. Весовой коэффициент ω_{ji} можно интерпретировать как вклад поля j -ой частицы в величину потенциала i -ой частицы. Хопфилд показал, что поведение такой сети аналогично поведению изингового спинового стекла. При этом он ввел понятие вычислительной энергии [57], которую можно интерпретировать в виде ландшафта с долинами и впадинами. Структура соединений сети определяет очертания ландшафта. Нейронная сеть выполняет вычисления, следуя по пути, уменьшающему вычислительную энергию сети. Это происходит до тех пор, пока путь не приведет на дно впадины. Данный процесс аналогичен скатыванию капли жидкости по склону, когда она минимизирует свою потенциальную энергию в поле тяготения. Впадины и долины в сети Хопфилда соответствуют наборам информации, которую хранит сеть. Если процесс начинается с приближенной или неполной информации, то он следует по пути, который ведет к ближайшей впадине. Это соответствует операции ассоциативного распознавания.

5.2.2. Нейронная сеть Хопфилда, как динамическая система

Нейронная сеть Хопфилда является динамической системой с непрерывным или дискретным временем. Состояния нейронов такой сети характеризуют фазовое пространство системы, когда каждому состоянию соответствует точка в фазовом пространстве. Тогда релаксационный процесс изменения состояния нейронов можно интерпретировать, как движение точки в фазовом пространстве в устойчивое положение. Траектория движения такой точки может характеризовать процесс воспоминания информации.

Существует параллельная и последовательная динамика работы сети Хопфилда [36].

Параллельная динамика характеризуется синхронным

функционированием нейронных элементов сети. При этом за один такт работы сети все нейроны одновременно изменяют свое состояние:

$$y_i(t+1) = F\left(\sum_{j=1}^n \omega_{ji} y_j(t) - T_i\right), \quad (5.8)$$

где $i = \overline{1, n}$.

Синхронную модель функционирования предложил Литтл [36].

Последовательная динамика характеризуется асинхронным процессом работы нейронной сети. В этом случае за один такт работы нейронной сети изменяется состояние только одного нейронного элемента:

$$y_i(t+1) = y_i(t), \quad \forall i \neq k, \\ y_i(t+1) = F\left(\sum_{j=1}^n \omega_{ji} y_j(t) - T_i\right), \quad \text{при } i = k. \quad (5.9)$$

Выбор нейрона, который на данном такте должен изменить свое состояние, производится случайным образом. Асинхронную модель функционирования исследовал Хопфилд [23].

Назовем динамическую систему *диссипативной* [64], если производная энергии ее по времени всегда отрицательна или равна нулю (в равновесном состоянии). В диссипативных системах на каждом шаге происходит необратимое уменьшение энергии. В установившемся режиме ($t = \infty$) все состояния таких систем сосредотачиваются на некотором подмножестве A фазового пространства:

$$\lim_{t \rightarrow \infty} Y(t) = A. \quad (5.10)$$

Такое подмножество A называется *аттрактором* и характеризует область притяжения системы. Оно является предельным множеством для фазовых траекторий, близких к A .

Зададим динамическую систему в стандартном виде:

$$\begin{aligned} \Delta y_i(t+1) &= y_i(t+1) - y_i(t), \\ y_i(0) &= y_i^0, \end{aligned} \quad (5.11)$$

где $i = \overline{1, n}$.

В работе [65] доказана следующая теорема.

Теорема 5.2. Динамическая система (5.11), матрица весовых коэффициентов W которой имеет нулевую главную диагональ, является диссипативной.

Аттракторами диссипативных динамических систем могут быть устойчивые стационарные точки или устойчивые предельные циклы [65].

Устойчивыми стационарными точками называются такие точки, для которых в установившемся режиме:

$$y_i(t+1) = y_i(t), \quad \forall i = \overline{1, n}. \quad (5.12)$$

Устойчивыми предельными циклами длины k называются такие циклы, для которых в установившемся режиме:

$$y_i(t+k) = y_i(t), \quad \forall i = \overline{1, n}. \quad (5.13)$$

При использовании симметричной матрицы весовых коэффициентов

($\omega_{ij} = \omega_{ji}$), аттракторами диссипативной динамической системы (5.11) являются устойчивые стационарные точки и предельные циклы длины два [65].

5.2.3. Энергия сети Хопфилда

Рассмотрим нейронную сеть Хопфилда с дискретным временем и непрерывным состоянием. В этом случае, как уже отмечалось, в качестве функции активации нейронных элементов можно использовать гиперболический тангенс. Тогда

$$y_i(t+1) = \text{th}(S_i(t)), \quad (5.14)$$

$$S_i(t) = \sum_{j=1}^n \omega_{ij} y_j(t) - T_i, \quad (5.15)$$

где $i \neq j$.

Состояние нейронного элемента $y_i(t+1)$ зависит от взвешенной суммы $S_i(t)$, которая характеризует входную активность i -го нейрона. Энергия одного нейронного элемента должна определяться таким образом, чтобы изменение состояния и входной активности нейрона приводило бы к уменьшению энергии. Тогда выходное значение i -го нейрона должно быть пропорционально градиенту энергии:

$$y_i = - \frac{dE(y_i, t)}{dS_i}, \quad (5.16)$$

где $E(y_i, t)$ – энергия i -го нейронного элемента в момент времени t .

Глава 5. Релаксационные нейронные сети

Такие системы называются *градиентными* [64]. Понятие энергии в градиентных системах соответствует функции Ляпунова. Из выражения (5.16) определим энергию i -го нейронного элемента:

$$E(y_i, t) = - \int_0^{y_i(t)} y_i(t) dS_i = -y_i(t)S_i(t) + \int_0^{y_i(t)} S_i(t) dy_i \quad (5.17)$$

Входную активность S_i можно определить при помощи обратной функции F :

$$S_i = F^{-1}(y_i) \quad (5.18)$$

Тогда

$$E(y_i, t) = -y_i(t)S_i(t) + \int_0^{y_i(t)} F^{-1}(y_i) dy_i \quad (5.19)$$

С учетом (5.15) энергия i -го нейронного элемента в момент времени t равняется

$$E(y_i, t) = - \sum_j \omega_{ji} y_j(t) y_i(t) + T_i y_i(t) + \int_0^{y_i(t)} F^{-1}(y_i) dy_i \quad (5.20)$$

Изменение энергии для одного нейрона можно определить следующим образом:

$$\Delta E(y_i, t+1) = E(y_i, t+1) - E(y_i, t) \quad (5.21)$$

Определим изменение энергии при асинхронном функционировании сети. Тогда

$$E(y_i, t+1) = -\sum_j \omega_{ji} y_i(t+1) y_j(t) + T_i y_i(t+1) + \int_0^{y_i(t+1)} F^{-1}(y_i) dy_i = -y_i(t+1) S_i(t) + \int_0^{y_i(t+1)} F^{-1}(y_i) dy_i. \quad (5.22)$$

Отсюда получаем

$$\Delta E(y_i, t+1) = -\Delta y_i(t+1) S_i(t) + \int_0^{y_i(t+1)} F^{-1}(y_i) dy_i. \quad (5.23)$$

Согласно теореме о среднем определенном интеграла [66]

$$\int_0^{y_i(t+1)} F^{-1}(y_i) dy_i = F^{-1}(\varepsilon) \cdot (y_i(t+1) - y_i(t)), \quad (5.24)$$

где $y_i(t) \leq \varepsilon \leq y_i(t+1)$.

Преобразуя последнее выражение, имеем:

$$\int_0^{y_i(t+1)} F^{-1}(y_i) dy_i = \Delta y_i(t+1) S_i(\varepsilon), \quad (5.25)$$

Тогда (5.23) можно представить, как

$$\Delta E(y_i, t+1) = \Delta y_i(t+1) \cdot (S_i(\varepsilon) - S_i(t)). \quad (5.26)$$

Рассмотрим, как изменяется энергия нейрона при изменении его входной и выходной активности.

Пусть $S_i(t) > S_i(t-1)$. Тогда $y_i(t+1) > y_i(t)$, $\Delta y_i(t+1) > 0$ и $S_i(\varepsilon) < S_i(t)$. В результате получим, что $\Delta E(y_i, t+1) < 0$.

Предположим, что $S_i(t) < S_i(t-1)$. Тогда $y_i(t+1) < y_i(t)$ и $\Delta y_i(t+1) < 0$. Так как в этом случае $S_i(\varepsilon) > S_i(t)$, то $\Delta E(y_i, t+1) < 0$.

Отсюда следует, что изменение входной активности одного нейрона приводит к необратимому уменьшению энергии этого нейрона. В результате, согласно теореме Ляпунова, обеспечивается сходимость нейронного элемента.

Определим полную энергию нейронной сети для системы с дискретным временем и непрерывным состоянием. Энергия сети представляет собой сумму энергий всех нейронных элементов, образующих нейронную сеть. Тогда

$$E(t) = -\sum_i E(y_i, t) = -\sum_i S_i(t)y_i(t) + \sum_i \int_0^{y_i(t)} F^{-1}(y_i) dy_i \quad (5.27)$$

Преобразуя первое слагаемое последнего выражения, получим

$$E(t) = -\sum_i \sum_j \omega_{ji} y_i(t) y_j(t) + \sum_i y_i(t) T_i + \sum_i \int_0^{y_i(t)} F^{-1}(y_i) dy_i \quad (5.28)$$

Так как для нейронной сети Хопфилда матрица весовых коэффициентов является симметричной ($\omega_{ij} = \omega_{ji}$), то первое слагаемое выражения (5.28)

обычно умножается на $\frac{1}{2}$. Тогда

$$E(t) = -\frac{1}{2} \sum_i \sum_j \omega_{ji} y_i(t) y_j(t) + \sum_i y_i(t) T_i + \sum_i \int_0^{y_i(t)} F^{-1}(y_i) dy_i \quad (5.29)$$

В матричной форме энергии системы можно представить в следующем виде:

$$E(t) = -\frac{1}{2} Y^T W Y + Y^T T + \sum_i \int_0^{y_i(t)} F^{-1}(y_i) dy_i \quad (5.30)$$

Таким образом энергия сети определяется композицией квадратичной формы и интеграла от нелинейности. Это соответствует функции Ляпунова (5.4), приведенной в разделе 5.1.

Определим теперь энергию сети Хопфилда для системы с дискретными временем и состоянием. В этом случае в качестве функции нелинейного преобразования F используется пороговая функция. Тогда

$$y_i(t+1) = \text{sign}(S_i(t)), \quad (5.31)$$

$$S_i(t) = \sum_{j=1}^n \omega_{ji} y_j(t) - T_i, \quad (5.32)$$

где $i \neq j$.

Энергия нейронного элемента в момент времени t равняется:

$$E(y_i, t) = -y_i(t)S_i(t) = -\sum_j \omega_{ji}y_i(t)y_j(t) + T_i y_i(t). \quad (5.33)$$

Рассмотрим изменение энергии нейронного элемента при асинхронном режиме функционирования сети. Тогда

$$E(y_i, t+1) = -y_i(t+1)S_i(t). \quad (5.34)$$

Отсюда

$$\Delta E(y_i, t+1) = -\Delta y_i(t+1)S_i(t). \quad (5.35)$$

Пусть $S_i(t) \geq 0$. Тогда $y_i(t+1) = 1$ и $\Delta y_i(t+1) \geq 0$. Получаем $\Delta E(y_i, t+1) \leq 0$. Если $S_i(t) \leq 0$, то $y_i(t+1) = -1$ и $\Delta y_i(t+1) \leq 0$. В результате $\Delta E(y_i, t+1) \leq 0$.

Таким образом, в результате изменения входной активности нейронного элемента, энергия его остается отрицательной. Согласно теореме 5.1 это гарантирует устойчивость нейронного элемента. Выходное значение нейронного элемента при асинхронном режиме функционирования можно представить в зависимости от изменения энергии нейрона следующим образом:

$$y_i(t+1) = F(S_i(t)) = F\left(-\frac{\Delta E(y_i, t+1)}{\Delta y_i(t+1)}\right). \quad (5.36)$$

Энергия системы равняется:

$$E(t) = -\sum_i S_i(t)y_i(t). \quad (5.37)$$

Преобразуя последнее выражение, получим:

$$E(t) = -\frac{1}{2} \sum_i \sum_j \omega_{ji} y_i(t) y_j(t) + T_i y_i(t). \quad (5.38)$$

Представим (5.38) в матричной форме. Тогда

$$E(t) = -\frac{1}{2} Y^T W Y + Y^T T. \quad (5.39)$$

Отсюда следует, что энергия сети с дискретными временем и состоянием определяется квадратичной формой. Это соответствует функции Ляпунова (5.3). Для синхронного режима функционирования энергия сети обычно задается на двух последовательных состояниях сети:

$$E(t) = -\sum_i S_i(t-1) y_i(t). \quad (5.40)$$

Преобразуя выражение (5.40), получим:

$$E(t) = -\frac{1}{2} \sum_i \sum_j \omega_{ji} y_i(t) y_j(t-1) + T_i y_i(t). \quad (5.41)$$

Существенного различия между выражениями (5.41) и (5.38) нет.

Таким образом, состояние нейронной сети Хопфилда описывается функцией энергии, которая эквивалентна функции Ляпунова. В процессе функционирования сети состояния нейронных элементов изменяются таким образом, что функция энергии их уменьшается. Это гарантирует сходимость нейронных элементов и сети в целом.

5.2.4 Анализ аттракторов

В разделе 5.2.2 отмечалось, что аттракторами нейронной сети Хопфилда при использовании симметричной матрицы весовых коэффициентов с нулевой главной диагональю могут быть, как устойчивые стационарные точки, так и предельные циклы длины два. Исследуем аттракторы сети Хопфилда.

Утверждение 5.1. Нейронная сеть Хопфилда сходится к устойчивым стационарным точкам, если в установившемся режиме, когда при $y_i(t+1) = y_i(t)$ для всех i , изменение энергии $\Delta E(t)$ равняется нулю.

Утверждение 5.2. Сеть Хопфилда сходится к предельному циклу длины два, если в установившемся режиме, когда при $y_i(t+2) = y_i(t)$ для всех i , изменение энергии $\Delta E(t)$ равняется нулю.

Утверждение 5.3. Для сети Хопфилда с дискретным временем и состоянием, если $y_i(t+2) = y_i(t)$ для всех i , то $y_i(t) = -y_i(t+1)$ для всех i .

Утверждение 5.4. Устойчивый предельный цикл длины два будет существовать тогда, когда при $y_i(t) = -y_i(t+1)$ для всех i , изменение энергии нейронной сети $\Delta E(t)$ равняется нулю.

Данные утверждения являются очевидными. Рассмотрим аттракторы нейронной сети с дискретным временем и состоянием.

Теорема 5.3. Нейронная сеть Хопфилда с асинхронным режимом функционирования сходится только к устойчивым стационарным точкам.

Доказательство. Энергия сети в два последовательных момента

времени при асинхронном функционировании определяется следующим образом:

$$E(t) = -\sum S_i(t)y_i(t), \quad (5.42)$$

$$E(t+1) = -\sum_{i \neq k} S_i(t)y_i(t) - S_k(t)y_k(t+1), \quad (5.43)$$

где k — номер нейронного элемента, для которого производится изменение состояния.

Тогда изменение энергии равняется:

$$\Delta E(t+1) = -\Delta y_k(t+1) \sum_{i \neq k} S_i(t)y_i(t), \quad (5.44)$$

где $\Delta y_k(t+1) = y_k(t+1) - y_k(t)$.

Рассмотрим наличие устойчивых стационарных точек в такой сети. Так как при $y_k(t+1) = y_k(t)$ изменение энергии $\Delta E(t) = 0$, то согласно утверждению (5.1), сеть сходится к устойчивым стационарным точкам.

Иследуем существование предельных циклов длины два. Тогда, если $y_k(t) = -y_k(t+1)$, то $\Delta y_k(t+1) \neq 0$ и, следовательно, $\Delta E(t+1) \neq 0$.

Теорема доказана.

Теорема 5.4. Аттракторами нейронной сети Хопфилда с синхронной динамикой являются устойчивые стационарные точки и предельные циклы длины два.

Доказательство. При синхронном режиме функционирования энергия сети в два последовательных момента времени определяется, как

$$E(t) = -\sum_i S_i(t)y_i(t), \quad (5.45)$$

$$E(t+1) = -\sum_i S_i(t+1)y_i(t+1). \quad (5.46)$$

Тогда изменение энергии равняется:

$$\Delta E(t+1) = -(\sum_i S_i(t+1)y_i(t+1) - S_i(t)y_i(t)). \quad (5.47)$$

Рассмотрим наличие устойчивых стационарных точек. Если $y_i(t+1) = y_i(t)$, то $S_i(t+1) = S_i(t)$. Отсюда получим $\Delta E(t+1) = 0$, что доказывает наличие устойчивых точек.

Определим существование предельных циклов длины два. Тогда, если $y_i(t+1) = -y_i(t)$, то $S_i(t+1) = -S_i(t)$. Из выражения (5.47) получаем, что $\Delta E(t+1) = 0$.

Теорема доказана.

Пример 5.1. Рассмотрим нейронную сеть Хопфилда с двумя нейронными элементами и пороговыми значениями, равными нулю (Рис. 5.4)

В качестве функции активации нейронных элементов второго слоя будем использовать пороговую функцию. Выходные значения сети являются биполярными, т. е. $y_i \in \{1, -1\}$.

Пусть $\omega_{12} = \omega_{21} = -1$. Матрица весовых коэффициентов сети равняется:

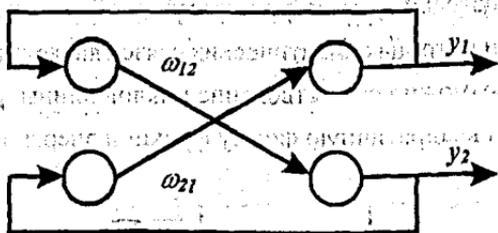


Рис. 5.4 Сеть Хопфилда с двумя нейронными элементами

$$W = \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix}$$

Тогда вектор выходных значений определяется, как

$$Y(t+1) = \text{sign}(W^T Y(t))$$

Пусть в момент времени $t = 0$: $y_1(0) = 1$, $y_2(0) = -1$. Тогда

$y_1(1) = 1$, $y_2(1) = -1$ и так далее. Отсюда следует, что точки 1 и -1 являются устойчивыми стационарными точками.

Предположим теперь, что $y_1(0) = 1$ и $y_2(0) = 1$. Тогда $y_1(1) = -1$, $y_2(1) = -1$; $y_1(2) = 1$, $y_2(2) = 1$ и так далее, т. е. $y_i(t+2) = y_i(t)$. Отсюда следует, что в такой сети присутствуют осцилляции в виде циклов длины два.

Аналогичная картина наблюдается для нейронных сетей с дискретным временем и непрерывным состоянием. В [65] доказана теорема, что если матрица весовых коэффициентов нейронной сети Хопфилда с синхронной динамикой является положительно полуопределенной (все ее собственные

значения неотрицательны), то аттракторами такой системы являются только точки покоя. Если матрица синаптических связей является несимметричной, то в такой сети возможно существование циклов длины n [65].

Рассмотрим квадратичную форму функции энергии:

$$E(t) = -\frac{1}{2} Y^T W Y = -\frac{1}{2} \sum_i \sum_j \omega_{ji} y_i(t) y_j(t) \quad (5.48)$$

При помощи ортогонального преобразования ее можно представить в канонической форме [66]:

$$E(t) = -\frac{1}{2} \sum_i \lambda_i y_i^2, \quad (5.49)$$

где λ_i , $i = \overline{1, n}$ — характеристические числа матрицы синаптических связей

W :

Можно показать, что если $y_i \in \{-1, 1\}$, то минимумы функции энергии (5.49) достигаются в узлах n -мерного куба (гиперкуба). Нейронная сеть с n

нейронами имеет 2^n состояний. При установке сети в начальное состояние происходит релаксационный процесс достижения минимума энергии, который определяется ближайшей вершиной гиперкуба.

5.2.5. Ассоциативная память

Сеть Хопфилда может использоваться в качестве ассоциативной памяти. В этом случае она способна распознавать зашумленные или искаженные образы. Для обучения нейронной сети Хопфилда используется правило

Хейбба. Представим входные образы, которые необходимо запомнить, в виде следующей матрицы:

$$Y = \begin{bmatrix} y^1 \\ y^2 \\ \dots \\ y^L \end{bmatrix} = \begin{bmatrix} y_1^1 & y_2^1 & \dots & y_n^1 \\ y_1^2 & y_2^2 & \dots & y_n^2 \\ \dots & \dots & \dots & \dots \\ y_1^L & y_2^L & \dots & y_n^L \end{bmatrix},$$

где L, n – соответственно общее количество и размерность входных образов.

Пусть входные векторы являются биполярными, т.е. $y_i \in \{-1, 1\}$. Тогда синаптические коэффициенты нейронной сети определяются, как

$$W = \sum_{k=1}^L (Y^k)^T Y^k = Y^T Y, \quad (5.50)$$

или в обычной форме:

$$\omega_{ji} = \sum_{k=1}^L y_i^k y_j^k, \quad (5.51)$$

где $i = \overline{1, n}, j = \overline{1, n}$, причем $i \neq j$.

Так как главная диагональ матрицы весовых коэффициентов сети Хопфилда должна быть нулевой, то

$$W = \sum_{k=1}^L \left((Y^k)^T Y^k - I \right) = Y^T Y - I, \quad (5.52)$$

где I – единичная матрица.

При использовании бинарных векторов, когда $y_i \in \{0,1\}$, правило обучения можно представить в следующем виде:

$$w_{ji} = \sum_{k=1}^L (2y_i^k - 1)(2y_j^k - 1), \quad (5.53)$$

$$W = (2Y - 1)^T (2Y - 1) - I, \quad (5.54)$$

где $i, j = \overline{1, n}$; $i \neq j$.

Приведенные выше выражения являются модификацией правила обучения Хебба. Они допускают усиление силы связи не только между одновременно активными, но и одновременно неактивными нейронами. На практике обычно используется биполярное представление входных сигналов.

Для настройки порогов нейронных элементов можно использовать следующее выражение:

$$T_i = - \sum_{k=1}^L y_i^k, \quad (5.55)$$

где y_i^k – i -ая компонента k -го входного образа.

В ассоциативной памяти пороговые значения нейронных элементов

обычно принимаются равными нулю. Матрица весовых коэффициентов, получаемая в соответствии с правилом Хебба является симметричной и характеризуется нулевой главной диагональю. Она представляет собой ковариационную матрицу входных образов, собственные векторы которой являются ортогональными. Тогда взвешенную сумму i -го нейрона можно представить в следующем виде:

$$S_i(t) = \lambda_i y_i(t), \quad (5.56)$$

где λ_i – i -ое собственное значение матрицы W .

В результате энергия сети определяется, как

$$E(t) = \frac{1}{2} \sum S_i(t) y_i(t) = -\frac{1}{2} \sum \lambda_i y_i^2(t). \quad (5.57)$$

Из этого выражения следует, что если $\lambda_i < 0$, то при минимизации энергии значение $y_i(t)$ должно стремиться к нулю. В противном случае, если $\lambda_i > 0$, то значение $y_i(t)$ для минимизации энергии будет увеличиваться. Таким образом, в процессе релаксации сети она будет ослаблять одни компоненты вектора Y и усиливать другие. На этом основана коррекция нейронной сетью Хопфилда ошибок во входных образах.

После обучения нейронной сети входные образы образуют впадины, соответствующие определенным наборам информации (рис. 5.5).

На рис. 5.5 ось ординат соответствует энергии сети Хопфилда, а Y_k характеризует k -ый образ, хранимый в сети. Таким образом информация хранится в локальных минимумах функции энергии, которые соответствуют

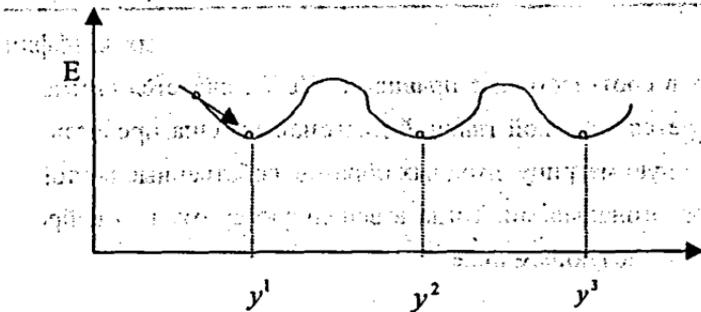


Рис. 5.5. Хранение образов в нейронной сети Хопфилда

стабильным состоянием сети. Совокупность точек, которые при равновесном состоянии сети отображаются в точку локального минимума, называется *бассейном притяжения*. Паттерны, соответствующие бассейну притяжения, характеризуют зашумленные образы памяти. При подаче на вход сети зашумленного образа она за конечное число шагов переходит в состояние, соответствующее ближайшему локальному минимуму (рис. 5.5). Количество шагов определяется расстоянием Хэмминга между искаженным и эталонным образом. Чем дальше расположен входной образ от локального минимума, тем больше итераций необходимо сети для перехода в стабильное состояние. Если наборы входных данных оказываются слишком похожими или слишком многочисленными, то в этом случае локальные минимумы будут располагаться слишком близко друг от друга и начинают интерферировать между собой (рис. 5.6). При этом появляются локальные минимумы, соответствующие комбинациям из достаточно больших фрагментов эталонных образов. Такое явление Хопфилд назвал химерами [55]. Если количество образов, хранящихся в сети, больше $0.15n$, то локальные минимумы начинают исчезать и поведение сети становится хаотическим (состояние спинового стекла). Размер бассейна притяжения можно регулировать при помощи синаптических связей и порогов нейронных

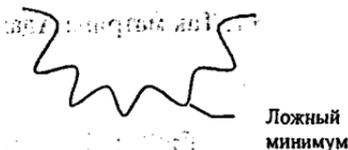


Рис. 5.6. Иллюстрация ложных минимумов

элементов. Так, если увеличивать силу тормозящих соединений, то бассейн притяжения станет больше, а впадины – глубже. Если уменьшить порог нейрона, то впадина станет также глубже. При достаточном уменьшении порога (в отрицательную сторону), бассейн притяжения может охватить все пространство конфигураций, ликвидировав впадины, соответствующие другим стабильным состояниям.

Хопфилд установил экспериментально, что объем памяти для устойчивой работы сети определяется, как

$$L = 0.15n, \quad (5.58)$$

где n – количество нейронов сети.

Позже были получены различные аналитические оценки допустимого объема памяти сети Хопфилда [36], например:

$$L = \frac{n}{4 \ln(n)}. \quad (5.59)$$

Данные выражения являются ориентировочными. Так при увеличении корреляции между входными образами и соответственно уменьшении расстояния Хэмминга емкость памяти уменьшается. Поэтому рекомендуется в качестве запоминаемых паттернов использовать линейно независимые, а лучше ортогональные векторы [67]. Для получения ортогональных векторов можно воспользоваться матрицей Адамара порядка n , элементами которой

являются действительные числа ± 1 . Так матрица Адамара второго порядка определяется, как:

$$H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix},$$

Матрица Адамара более высокого порядка определяется на основе матриц меньшего порядка [68].

$$H_{2n} = \begin{bmatrix} H_n & H_n \\ H_n & -H_n \end{bmatrix}.$$

Так, например, матрица четвертого порядка имеет следующий вид:

$$H_4 = \begin{bmatrix} H_2 & H_2 \\ H_2 & -H_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}.$$

Для ортогональных векторов расстояние Хэмминга определяется, как

$$\delta = n/2. \quad (5.60)$$

Назовем *радиусом притяжения* число битов, на которое может отличаться входной вектор от эталонного, чтобы произошел процесс идентификации. Для ортогональных векторов радиус притяжения определяется следующим образом [67]:

$$r = \frac{n}{2L}, \quad (5.61)$$

где L – количество образцов, хранимых сетью.

При увеличении емкости памяти уменьшается радиус притяжения и соответственно корректирующие свойства сети Хопфилда. В некоторых работах для увеличения емкости памяти рекомендуется использовать сети Хопфилда *высокого порядка* [69]. Так, например, для сети второго порядка, выходное значение i -го нейронного элемента определяется, как

$$y_i(t+1) = F(S_i(t)), \quad (5.62)$$

$$S_i(t) = \sum_{h=1}^n \sum_{j=1}^n w_{hji} y_h(t) y_j(t), \quad (5.63)$$

где w_{hji} – синаптическая связь от h -го и j -го к i -му нейрону.

Обобщая правило обучения Хебба можно получить выражение для настройки весовых коэффициентов сети второго порядка

$$w_{hji} = \sum_{k=1}^L X_h^k X_j^k X_i^k - I, \quad (5.64)$$

где I – трехмерная единичная матрица.

Весовые коэффициенты для такой нейронной сети являются также симметричными, т.е.

$$w_{hji} = w_{jhi} = w_{ijh}$$

Использование нейронной сети Хопфилда второго порядка позволяет повысить емкость памяти приблизительно до $0.3n$ [69].

5.2.6. Функционирование сети Хопфилда

Функционирование сети Хопфилда представляет собой релаксационный процесс в ходе которого сеть достигает устойчивого состояния. Алгоритм функционирования можно представить в виде следующей последовательности шагов.

1. На вход сети подается неизвестный образ.
2. В зависимости от синхронного или асинхронного режима работы сети производятся следующие вычисления:
 - а) При синхронном режиме функционирования сети выходы нейронных элементов изменяются одновременно

$$y_i(t+1) = F\left(\sum_j \omega_{ji} y_j(t)\right), \quad \forall i,$$

где в качестве оператора нелинейного преобразования F обычно используется пороговая функция.

- б) В случае асинхронного режима работы в каждый такт времени изменяется состояние только одного нейронного элемента k :

$$y_i(t+1) = F\left(\sum_j \bar{\omega}_{ji} y_j(t)\right), \quad \text{если } i = k,$$

$$y_i(t+1) = y_i(t), \quad \text{для } i \neq k.$$

При этом на каждом шаге работы сети такой нейронный элемент выбирается случайно из всей совокупности нейронных элементов или из тех нейронов, которые на предыдущем шаге не изменяли свое состояние.

3. Пункт 2 повторяется до тех пор, пока сеть не перейдет в стабильное состояние. При этом состояния всех нейронных элементов перестают

меняться, т.е.

$$y_i(t+1) = y_i(t), \quad \forall i.$$

Пример 5.2. Пусть необходимо, чтобы сеть Хопфилда хранила один инарный образ:

$$Y = (1 \ 1 \ 1 \ 0).$$

Тогда

$$\begin{aligned}
 W &= (2Y - 1)^T \cdot (2Y - 1) - I = \begin{bmatrix} 1 \\ 1 \\ -1 \\ -1 \end{bmatrix} \cdot [1 \ 1 \ 1 \ -1] - \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \\
 &= \begin{bmatrix} 1 & 1 & 1 & -1 \\ 1 & 1 & 1 & -1 \\ 1 & 1 & 1 & -1 \\ -1 & -1 & -1 & 1 \end{bmatrix} - \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 1 & -1 \\ 1 & 0 & 0 & -1 \\ 1 & 1 & 0 & -1 \\ -1 & -1 & -1 & 0 \end{bmatrix}.
 \end{aligned}$$

В весовой матрице j -ый столбец соответствует весовым связям j -ого нейронного элемента. Пусть на вход сети поступает зашумленный образ

$Y = (0 \ 0 \ 1 \ 0)$. Рассмотрим различные режимы функционирования:

1. Асинхронный режим

Допустим, что в начальный момент времени $t=1$ выбран первый нейрон, состояние которого будет изменяться. Тогда

$$y_1 = \text{sign}(YW_1),$$

$$S_1 = YW_1 = [0 \ 0 \ 1 \ 0] \cdot \begin{bmatrix} 0 \\ 1 \\ 1 \\ -1 \end{bmatrix} = 0 + 0 + 1 + 0 = 1,$$

где W_1 - первый столбец весовой матрицы.

Отсюда получаем, что

$$y_1 = \text{sign}(1) = 1.$$

В результате этого на выходе сети получается следующий образ:

$$Y(1) = (1 \ 0 \ 1 \ 0).$$

На втором шаге $t=2$ выбираем четвертый нейрон, состояние которого изменяется. Тогда

$$y_4 = \text{sign}(Y(1)W_4) = \text{sign}(-2) = 0,$$

$$Y(2) = (1 \ 0 \ 1 \ 0).$$

При изменении в следующий момент времени состояния третьего нейрона, получим

$$y_3 = F(Y(2)W_3) = F(1) = 1,$$

$$Y(3) = (1 \ 0 \ 1 \ 0).$$

Аналогичным образом для второго нейронного элемента:

$$y_2 = F(Y(3)W_2) = F(2) = 1,$$

$$Y(4) = (1 \ 1 \ 1 \ 0).$$

Если продолжить процесс дальше, то можно показать, что состояния нейронных элементов перестанут изменяться. Это свидетельствует о переходе сети в стабильное состояние. Таким образом сеть распознала зашумленный образ.

2. Синхронный режим

В матричной форме функционирование сети в синхронном режиме можно представить, как

$$y(t+1) = \text{sign}(S(t)),$$

$$S(t) = Y(t)W.$$

Пусть на вход сети подается зашумленный образ $Y = (0 \ 0 \ 1 \ 0)$. Тогда в начальный момент времени:

$$S(0) = Y(0) \cdot W = \begin{bmatrix} 0 & 1 & 1 & -1 \\ 1 & 0 & 1 & -1 \\ 1 & 1 & 0 & -1 \\ -1 & -1 & -1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & -1 \end{bmatrix}$$

$$Y(1) = \text{sign}(S(0)) = \begin{bmatrix} 1 & 1 & 0 & 0 \end{bmatrix}$$

Для следующего такта работы

$$S(1) = Y(1) \cdot W = \begin{bmatrix} 1 & 1 & 2 & -2 \end{bmatrix},$$

$$Y(2) = \text{sign}(S(1)) = \begin{bmatrix} 1 & 1 & 1 & 0 \end{bmatrix}.$$

При дальнейшем функционировании сеть перестает изменять свое состояние. В результате произошло распознавание зашумленного образа.

5.2.7 Решение задач оптимизации

Нейронные сети Хопфилда можно использовать для решения различного рода задач оптимизации [56, 55]. Это основано том, что стабильные состояние нейронной сети соответствуют локальным минимумам функции энергии. Пусть необходимо найти минимум целевой функции:

$$\min F(x) \quad (5.65)$$

Для решения такой задачи с использованием нейронной сети Хопфилда, необходимо поставить в соответствие целевую функцию и функцию энергии:

$$F(x)=E \quad (5.66)$$

Решая полученное уравнение относительно W , можно определить весовые коэффициенты сети Хопфилда для минимизации целевой функции. Запрограммированная подобным образом нейронная сеть будет находить решение задачи оптимизации в виде минимума функции энергии. В результате получается решение, как правило, в виде локального, а не глобального оптимума. Это является недостатком сети Хопфилда. Достоинство ее состоит в высокой скорости вычислений. Особенно это важно для NP-полных задач оптимизации. Другой аспект этой проблемы состоит в том, что во многих областях требуется найти решение задачи в реальном масштабе времени. При этом главным фактором является быстроедействие алгоритма, а не определение глобального оптимума. Для таких задач эффективно применение сети Хопфилда, которая позволяет достаточно быстро найти локальное решение задачи оптимизации.

5.2.7.1. Задача Коммивояжера

Задача *коммивояжера* (Travelling salesman problem) формулируется следующим образом.

Пусть даны города и расстояния между ними. Требуется найти замкнутый маршрут коммивояжера, который начинается и заканчивается в одном и том же городе и проходит через все города. При этом маршрут должен иметь минимальную длину, а также входить в каждый город и выходить из него по одному разу. Задача коммивояжера является NP-полной.

Рассмотрим нейропостановку такой задачи [56, 70]. Пусть количество городов равняется n . Тогда структура нейронной сети Хопфилда представляет собой матрицу нейронных элементов размерностью $n \times n$. В такой матрице номер строки соответствует городу, а номер столбца характеризует позицию города в путешествии. Пусть $n = 4$ и обозначим города как A, B, C, D. Матрица

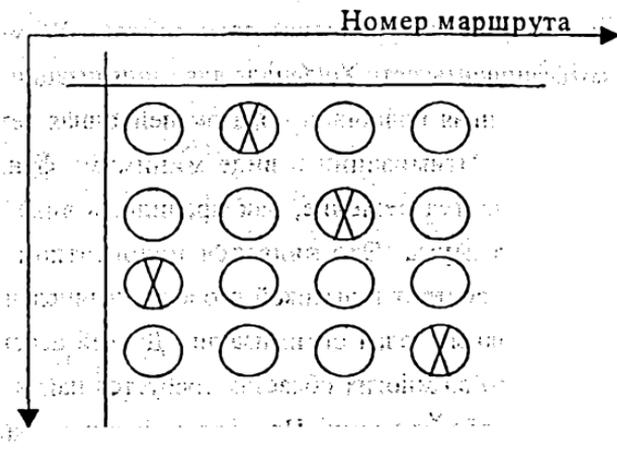


Рис. 5.7. Матрица нейронных элементов для четырех городов

нейронных элементов для этого случая изображены на рис. 5.7.

Определим выходные значения нейронных элементов следующим образом:

$$Y_{Ai} = \begin{cases} 1, & \text{если город } A \text{ стоит в маршруте на } i\text{-ом месте} \\ 0, & \text{иначе} \end{cases}$$

Тогда, если активными будут нейроны, показанные на рис.5.7, то маршрут коммивояжера будет следующим:

$$C \rightarrow A \rightarrow B \rightarrow D \rightarrow C.$$

Целевая функция задачи коммивояжера будет состоять из следующих частей:

1. Части, отвечающей за допустимость решения, которая определяет синтаксис задачи.

2. Части, представляющей качество решения. Она непосредственно отвечает за минимизацию целевой функции.

Синтаксис задачи должен удовлетворять следующим условиям:

1. Каждый город должен стоять в маршруте не более, чем на одном месте.
2. На каждом месте маршрута должен стоять не более чем один город.
3. Общее число единиц в матрице нейронных элементов должно быть равно n .

Тогда целевую функцию, ответственную за синтаксис задачи можно представить, как

$$F_c = \frac{a}{2} \sum_A \sum_I \sum_{j \neq A} Y_{AI} Y_{Aj} + \frac{b}{2} \sum_I \sum_A \sum_{j \neq A} Y_{AI} Y_{Aj} + \frac{c}{2} (\sum_A \sum_I Y_{AI} - n)^2 \quad (5.67)$$

где a, b, c - постоянные, которые больше нуля. Минимум данного выражения равен нулю, когда выполняются три перечисленных выше условия синтаксиса задачи.

Целевая функция, отвечающая за качество решения должна обеспечивать минимальное расстояние выбранного маршрута. Она определяется следующим образом:

$$F_v = \frac{d}{2} \sum_A \sum_{B \neq A} \sum_I \text{dist}(A, B) Y_{AI} (Y_{BI+1} + Y_{BI-1}) \quad (5.68)$$

где $\text{dist}(A, B)$ – расстояние между городами A и B , d - постоянная, которая больше нуля.

Энергия дискретной сети Холфилда для матрицы нейронных элементов определяется, как

$$E(t) = \frac{1}{2} \sum_A \sum_{B \neq A} \sum_I \sum_{j \neq A} \omega_{AI, Bj} Y_{AI}(t) Y_{Bj}(t) + \sum_A \sum_I T_{AI} Y_{AI}(t) \quad (5.69)$$

Составим уравнение:

$$E(t) = Fc + Fe \quad (5.70)$$

Тогда решение уравнения (5.70) можно представить в следующем виде:

$$W_{AiBj} = W_{AiBj}(1) + W_{AiBj}(2) + W_{AiBj}(3) + W_{AiBj}(4) \quad (5.71)$$

Слагаемые последнего выражения определяются, как

$$W_{Ai,Bj}(1) = -a\delta_{AB}(1 - \delta_{ij}), \quad (5.72)$$

$$W_{Ai,Bj}(2) = -b\delta_{ij}(1 - \delta_{AB}), \quad (5.73)$$

$$W_{Ai,Bj}(3) = -c, \quad (5.74)$$

$$W_{Ai,Bj}(4) = -d(1 - \delta_{AB})(\delta_{ij+1} + \delta_{ij-1}) \text{dist}(A, B) \quad (5.75)$$

Пороги нейронных элементов при этом одинаковые и равняются:

$$T_{Ai} = -cn \quad (5.76)$$

Тогда уравнение динамики сети Хопфилда можно представить как

$$S_{Ai}(t) = -a \sum_{j \neq i} Y_{Aj} - b \sum_{B \neq A} Y_{Bi} - c \sum_{B \neq A} \sum_{j \neq i} Y_{Bj} - d \sum_{B \neq A} \text{dist}(A, B) (Y_{Bi+1} + Y_{Bi-1}) + cn \quad (5.77)$$

Можно также задать динамику функционирования сети Хопфилда через изменение функции энергии.

$$Y_{Ai}(t+1) = \text{sign}\left(\frac{-\Delta E(t)}{\Delta Y_{Ai}(t)}\right) \quad (5.78)$$

Хопфилд использовал для моделирования задачи коммивояжера следующие константы:

$$a=b=500, c=200, d=500.$$

Недостатком такой сети является сложность выбора подходящих постоянных a, b, c, d . В работе [70] приводится методика выбора данных постоянных. Как уже отмечалось, сеть Хопфилда дает решение задач

оптимизации в основном в виде локального минимума. По оценке Хопфилда [42] только 50% всех решений получается в области, близкой к глобальному оптимуму. Другие авторы [71] снижают эту цифру до 15%. В приложении 2 приведены эксперименты моделирования бинарной сети Хопфилда для решения задачи коммивояжера. Как следует из экспериментов, только две попытки из десяти привели к решению с 25%-й относительной погрешностью относительно глобального оптимума.

5.2.7.2. Машина Больцмана

В предыдущем разделе было отмечено, что нейронная сеть Хопфилда находит локальный оптимум задачи оптимизации. Для устранения этого недостатка можно использовать машину Больцмана, которая является расширением сети Хопфилда [60]. В основе сети Больцмана лежит метод *имитационного отжига*, который является разновидностью процедуры случайного поиска [72]. Такой метод позволяет выбраться из локального минимума целевой функции. Имитационный отжиг моделирует процесс релаксации системы, когда она при некоторой конечной температуре стремится к состоянию равновесия. Флуктуации, возникающие при высоких температурах имеют вероятностный характер и в общем случае при понижении температуры система приходит в состояние равновесия, которое соответствует минимальному значению энергии. В машине Больцмана имитационный отжиг имитируется путем вероятностного механизма перехода нейронного элемента в новое состояние:

$$P_i(S) = \frac{1}{1 + \exp\left(-\frac{\Delta E_i}{T}\right)} \quad (5.79)$$

где ΔE_i – изменение энергии i -го нейронного элемента, T – параметр

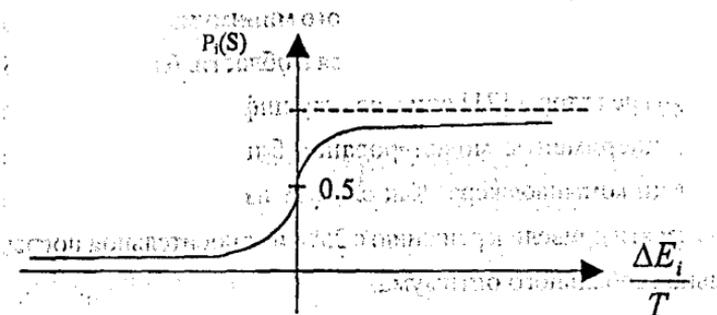


Рис.5.8. График функции вероятности изменения состояния нейронного элемента

температуры, S – состояние нейронного элемента в момент времени $t+1$.

График изменения функции (5.79) приведен на рис.5.8.

При $T \rightarrow 0$ вероятность перехода P_i нейронного элемента в новое состояние стремится к 0.5 и сеть находится в *хаотическом состоянии*.

Если $T=0$ и $\Delta E_i > 0$, то $P_i = 1$ и система становится детерминированной с единичной вероятностью перехода в новое состояние. При этом она переходит в состояние равновесия, скатываясь в ближайший локальный минимум. В случае, когда $T=0$ и $\Delta E_i < 0$, то $P_i = 0$ и система стабилизируется в этом состоянии. Таким образом, если начать процесс релаксации с высокой температуры и постепенно понижать ее, то система придет в *состояние равновесия*.

Рассмотрим некоторые аспекты использования машины Больцмана. Существуют различные вариации ее работы. Так, при использовании пороговой функции активации, состояния нейронных элементов выбираются случайным образом:

$$Y_i = \text{random}(0, 1) \quad (5.80)$$

Если применяется сигмоидная функция активации, то состояние

нейрона можно выбирать в соответствии со следующими выражениями.

$$y_i = \begin{cases} 1, & \text{если } \text{random}(0,1) \leq \bar{y}_i \\ 0, & \text{если } \text{random}(0,1) > \bar{y}_i \end{cases}$$

где

$$\bar{y}_i = \frac{1}{1 + \exp\left(-\frac{S_i}{T}\right)} \quad (5.81)$$

Здесь S_i - взвешенная сумма i -го нейрона.

Температура в процессе релаксации должна уменьшаться. Для этого могут использоваться различные механизмы [36]:

$$T(t+1) = aT(t), \quad (5.82)$$

$$T(t) = \frac{T_0}{1 + \log(t)}, \quad (5.83)$$

где $0 < a < 1$; T_0 - начальная температура сети.

Алгоритм работы машины Больцмана в общем случае можно представить в виде следующей последовательности шагов:

1. Устанавливаются в начальное состояние $Y(t)(t=0)$ нейронные элементы и температура T сети.
2. Случайным образом выбирается нейронный элемент с номером i и для него в соответствии с выражением (5.80) или (5.81) генерируется новое состояние $y_i(t+1)$.
3. Вычисляется изменение энергии ΔE_i для i -го нейронного элемента. В зависимости от величины ΔE_i производятся следующие действия:

Глава 5. Релаксационные нейронные сети

3.1. Если $\Delta E_i < 0$, то фиксируется новое состояние i -го нейронного элемента, т.е. $y_i(t+1) = y_i(t)$.

3.2 В случае, когда $\Delta E_i \geq 0$, то с вероятностью P_i фиксируется новое состояние нейронного элемента $y_i(t+1) = y_i(t)$, а с вероятностью $(1-P_i)$ состояние i -го нейрона не изменяется, т.е. $y_i(t+1) = y_i(t)$.

3.3 Алгоритм повторяется, начиная с пункта 2 для всех нейронных элементов сети.

4. Проверяется выполнение условия равновесия. Если это условие не выполняется, то $T = \alpha T$ и алгоритм повторяется начиная с пункта 2. В противном случае происходит останов алгоритма и выдача результатов.

5.3. Нейронная сеть Хэмминга

Нейронная сеть Хэмминга (Hamming Network) была предложена в 1987 году Липпманом (Lippman) [33]. Она представляет собой релаксационную многослойную сеть, в которой используются обратные связи между отдельными слоями. Сеть Хэмминга используется в качестве ассоциативной памяти. При распознавании образов она использует в качестве меры близости расстояние Хэмминга. Весовые коэффициенты и пороги сети Хэмминга определяются из условия задачи. Поэтому такая сеть является нейронной сетью с фиксированными связями.

5.3.1. Архитектура сети

Сеть Хэмминга является многослойной, состоящей из различных классов нейронных сетей. Пусть имеется m образов, каждый из которых имеет размерность n :

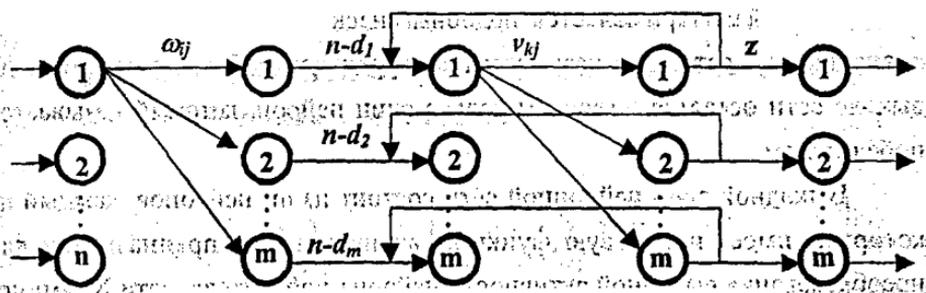


Рис. 5.9. Архитектура нейронной сети Хэмминга

$$X^1 = [x_1^1, x_2^1, \dots, x_n^1]$$

$$X^2 = [x_1^2, x_2^2, \dots, x_n^2]$$

$$\dots$$

$$X^m = [x_1^m, x_2^m, \dots, x_n^m]$$

Тогда нейронная сеть Хэмминга будет состоять из сети с прямыми связями, сети Хопфилда и слоя выходных нейронов (рис. 5.9).

Сеть с прямыми связями состоит из n входных распределительных и m выходных нейронных элементов. Она вычисляет меру подобия между входным и эталонными образами хранящимися в сети. В качестве меры подобия используется количество одинаковых разрядов между входным и эталонным образом. Тогда выходное значение i -го нейрона второго слоя представляет собой меру подобия P_i между входным и i -м эталонным образом:

$$P_i = n - d_i, \quad (5.84)$$

где d_i – расстояние Хэмминга между входным и i -м эталонным паттерном.

Сеть Хопфилда используется для разрешения возникающих конфликтов, когда входной паттерн является подобным нескольким эталонным образам, хранящимся в сети. Это называется конкурентной работой сети, когда на выходе сети остается активным только один нейрон, который называется победителем.

Выходной слой нейронной сети состоит из m нейронов, каждый из которых имеет пороговую функцию активации. Он предназначен для преобразования выходной активности нейрона-победителя сети Хэмминга в единичное значение. При этом значения всех остальных нейронов выходного слоя устанавливаются в нулевое состояние. Таким образом происходит идентификация входного паттерна, который кодируется номером нейрона выходного слоя имеющем единичное значение. Если входной образ не совпадает с эталонным, то на выходе сети Хэмминга будет формироваться такой эталонный паттерн, который имеет минимальное расстояние Хэмминга по отношению к выходному образу.

5.3.2 Обучение

Рассмотрим правила определения весовых коэффициентов для различных слоев нейронной сети Хэмминга.

Весовые коэффициенты и пороги сети с прямыми связями настраиваются таким образом, чтобы выходное значение i -го нейронного элемента соответствовало i -ой мере подобия P_i между входным и i -м эталонным образом. Для этого необходимо, чтобы

$$w_{ij} = \frac{x_i^j}{2}, T_j = \frac{n}{2}, \quad (5.85)$$

где $i = \overline{1, n}$; $j = \overline{1, m}$.

Тогда выходная активность j -го нейронного элемента сети с прямыми связями определяется следующим образом:

$$y_j = \sum_{i=1}^n w_{ij} x_i + T_j = \sum_{i=1}^n \frac{x_i'}{2} x_i + \frac{n}{2} = \frac{1}{2} (\sum_{i=1}^n x_i' x_i + n). \quad (5.86)$$

Легко показать, что выражение (5.86) эквивалентно мере подобия между входным и j -м эталонным образом.

$$y_j = P_j = n - d_j. \quad (5.87)$$

Сеть Хопфилда, как уже отмечалось, предназначена для устранения возможных конфликтов, когда входной паттерн является похожим на несколько эталонных образов, хранящихся в сети. Для этого весовые коэффициенты сети Хопфилда определяются следующим образом:

$$v_{kj} = \begin{cases} 1, & \text{если } k = j \\ -e, & \text{если } k \neq j \end{cases} \quad (5.88)$$

где $e = \text{const}$. Параметр e обычно лежит в диапазоне

$$0 < e < \frac{1}{m}. \quad (5.89)$$

Таким образом каждый нейрон связан с остальными нейронными элементами сети Хопфилда только тормозящими связями. В качестве функции активации нейронных элементов сети Хопфилда используется следующая функция:

$$z_j = F(S_j) = \begin{cases} S_j, & \text{если } S_j > 0 \\ 0, & \text{если } S_j \leq 0 \end{cases}$$

Начальная инициализация в момент времени $t = 0$ нейронных элементов сети Хопфилда соответствует преобразованной выходной активности нейронов второго слоя сети Хемминга. Тогда:

$$S_j(0) = y_j = \sum_{i=1}^n w_{ij} x_i + T_j = n - d_j, \quad (5.90)$$

$$z_j(0) = F(y_j) = \begin{cases} y_j, & \text{если } y_j \geq 0 \\ 0, & \text{иначе} \end{cases} \quad (5.91)$$

В момент времени t выходная активность сети Хопфилда вычисляется в соответствии со следующим выражением:

$$z_j(t) = F(z_j(t-1) - e \sum_{\substack{k \\ k \neq j}} z_k(t-1)). \quad (5.92)$$

Релаксационный процесс происходит до тех пор, пока только один нейронный элемент сети Хопфилда не останется с положительной активностью. Такой нейронный элемент является победителем в конкурентной борьбе. Выходной слой сети Хэмминга преобразует выходную активность нейрона-победителя в единичное значение, а остальных нейронов в нулевое значение. Для этого нейроны выходного слоя используют пороговую функцию активации. Номер нейрона победителя идентифицирует распознанный образ. Количество образов хранимых в сети равняется

количеству нейронных элементов выходного слоя.

5.3.3. Алгоритм функционирования

Рассмотрим алгоритм функционирования нейронной сети Хэмминга.

Он состоит из следующих шагов:

1. Определяются весовые коэффициенты и пороговые значения для соответствующих слоев нейронной сети.

2. На вход сети подается неизвестный образ и производится инициализация нейронных элементов сети Хопфилда в соответствии с выражениями (5.90) и (5.91).

3. Производится итерационная процедура расчета выходных значений нейронной сети Хопфилда до тех пор, пока она не стабилизируется. В этом случае на выходе сети Хэмминга один нейронный элемент будет иметь единичное состояние, а остальные – нулевое состояние.

4. Если в выходном слое существует несколько нейронных элементов победителей, то выбор одного из них производится случайным образом.

5.4 Двухнаправленная ассоциативная память

Двухнаправленную ассоциативную память (Bidirectional Associative Memory) предложил Коско (Kosko) в 1988 году [38, 59]. Она является дальнейшим развитием сети Хопфилда и представляет собой релаксационную сеть с циркуляцией информации.

5.4.1 Архитектура

Двухнаправленная ассоциативная память представляет собой нейронную сеть, состоящую из двух слоев нейронных элементов (рис. 5.10). Нейронные

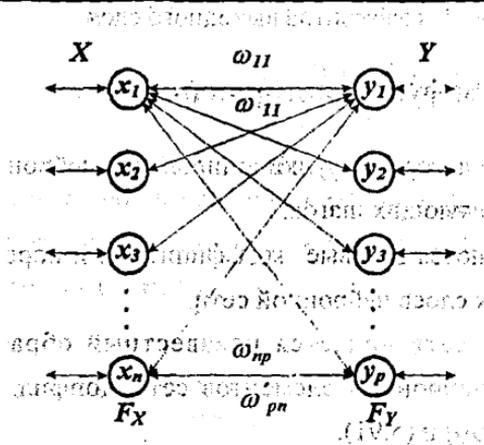


Рис. 5.10. Архитектура двунаправленной ассоциативной памяти

элементы каждого из слоев могут быть как входными, так и выходными.

При прямом распространении информации нейронные элементы слоя F_x являются входными, а слоя F_y – выходными. При обратном распространении информации входными являются нейроны слоя F_y , а выходными – слоя F_x . Каждый нейронный элемент слоя F_x имеет синаптические связи w_{ij} со всеми нейронами слоя F_y . Соответственно синаптические связи в обратном направлении, от слоя F_y к слою F_x , обозначим как w'_{ji} . Архитектура двунаправленной памяти аналогична рециркуляционной нейронной сети. Однако принципы функционирования таких сетей разные. Двунаправленная ассоциативная память в отличие от рециркуляционной сети является релаксационной сетью.

5.4.2 Обучение и функционирование

Для обучения двунаправленной памяти используется правило Хебба. Пусть задано L пар векторов X и Y :

$$X_k = [x_1^k, x_2^k, \dots, x_n^k]$$

$$Y_k = [y_1^k, y_2^k, \dots, y_p^k],$$

где $k = \overline{1, L}$; L – количество образов одного типа, хранимых в нейронной сети.

Тогда в соответствии с правилом Хебба весовые коэффициенты такой сети определяются следующим образом:

$$w_{ij} = \sum_{k=1}^L x_i^k y_j^k, \quad (5.93)$$

или в матричной форме:

$$W = \sum_{k=1}^L X_k^T Y_k. \quad (5.94)$$

Весовые коэффициенты W' обратного слоя определяются, как

$$W' = W^T. \quad (5.95)$$

Взвешенная сумма S_j j -го нейронного элемента слоя F_y равняется:

$$S_j = \sum_{i=1}^n x_i w_{ij}. \quad (5.96)$$

Выходные значения нейронов слоя F_y формируются следующим образом:

$$y_j(t+1) = F(S_j) = \begin{cases} 1, \text{если } S_j > 0 \\ y_j(t), \text{если } S_j = 0 \\ -1, \text{если } S_j < 0 \end{cases}$$

где $j = \overline{1, p}$; F – оператор нелинейного преобразования. Взвешенную сумму i -го нейронного элемента слоя F_x можно представить, как

$$S_i = \sum_{j=1}^p y_j w_{ji}. \quad (5.97)$$

Выходные значения нейронов слоя F_x формируются следующим образом:

$$x_i(t+1) = F(S_i) = \begin{cases} 1, \text{если } S_i > 0 \\ x_i(t), \text{если } S_i = 0 \\ -1, \text{если } S_i < 0 \end{cases}$$

В матричной форме выходные значения слоев F_y и F_x определяются следующим образом:

$$Y = F(XW), \quad (5.98)$$

$$X = F(YW^T). \quad (5.99)$$

Емкость памяти можно вычислить при помощи следующего выражения [73]:

$$L = \frac{m}{4 \ln(m)}, \quad (5.100)$$

где $m = \min(n, p)$.

Устойчивость двунаправленной ассоциативной памяти доказал Коско [59]. Для этого он использовал функцию Ляпунова, которая определяется также, как для нейронной сети Хопфилда.

5.4.3 Алгоритм функционирования

Алгоритм функционирования двунаправленной ассоциативной памяти можно представить в виде следующей последовательности шагов:

1. На входные слои F_X и F_Y подаются входные образы X и Y или только один из них.
2. Активация нейронных элементов в слое F_X передается через весовую матрицу W на слой F_Y .
3. Вычисляются выходные значения слоя F_Y .
4. Выходные значения слоя F_Y через весовую матрицу W' поступают на слой F_X .
5. Вычисляются выходные значения нейронных элементов слоя F_X .
6. Процедура повторяется, начиная с пункта 2, пока все выходные значения нейронных элементов слоев F_X и F_Y не перестанут изменяться. Это положение равновесия называется резонансом.

Пример 5.3. Пусть имеется две пары входных образов, которые должны быть запомнены нейронной сетью:

$$X_1 = [-1 \ 1 \ -1], \quad Y_1 = [-1 \ 1 \ -1 \ -1];$$

$$X_2 = [1 \ -1 \ 1], \quad Y_2 = [1 \ -1 \ 1 \ 1].$$

Тогда в соответствии с правилом (5.94) матрица синаптических связей определяется следующим образом:

$$W = \sum_{k=1}^2 X_k^T Y_k = \begin{bmatrix} -1 \\ 1 \\ -1 \end{bmatrix} \cdot [-1 \ 1 \ -1 \ -1] + \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix} \cdot [1 \ -1 \ 1 \ 1] =$$

$$= \begin{bmatrix} 1 & -1 & 1 & 1 \\ -1 & 1 & -1 & -1 \\ 1 & -1 & 1 & 1 \end{bmatrix} + \begin{bmatrix} 1 & -1 & 1 & 1 \\ -2 & 2 & -2 & -2 \\ 2 & -2 & 2 & 2 \end{bmatrix}.$$

Проверим работоспособность сети при распознавании эталонных образов. Подадим на вход F_X сети образ X_1 . Тогда в соответствии с выражениями (5.98), (5.99) на выходах сети получим:

$$Y = F(X_1 W) = F([-6 \ 6 \ -6 \ -6]) = [-1 \ 1 \ -1 \ -1] = Y_1$$

$$X = F(Y_1 W^T) = F([-8 \ 8 \ -8]) = [-1 \ 1 \ -1] = X_1.$$

Отсюда следует, что сеть корректно функционирует в качестве ассоциативной памяти.

Рассмотрим функционирование сети при распознавании зашумленных

ПРИЛОЖЕНИЕ 1

Статистическая таблица размеров стока вод реки Березина:

Таблица 1

Год наблюдения	Размер стока	Год наблюдения	Размер стока
1877	139	1937	100
1878	119	1938	100
1879	124	1939	103
1880	111	1940	123
1881	93,9	1941	142
1882	82,4	1942	125
1883	198	1943	118
1884	110	1944	112
1885	110	1945	120
1886	106	1946	121
1887	111	1947	121
1888	111	1948	118
1889	116	1949	110
1890	93,9	1950	86
1891	95,6	1951	120
1892	87,6	1952	106
1893	116	1953	126
1894	98,4	1954	73,5
1895	121	1955	102
1896	141	1956	158
1897	128	1957	122
1898	94,3	1958	204
1899	119	1959	112
1900	114	1960	114
1901	102	1961	97,6
1902	138	1962	169
1903	141	1963	111
1904	101	1964	111
1905	130	1965	125
1906	129	1966	131

Продолжение таблицы 1

Год наблюдения	Размер стока	Год наблюдения	Размер стока
1907	130	1967	113
1908	136	1968	116
1909	109	1969	94,4
1910	124	1970	151
1911	98,9	1971	125
1912	140	1972	91,4
1913	126	1973	114
1914	108	1974	110
1915	97,1	1975	114
1916	147	1976	94
1917	152	1977	118
1918	122	1978	133
1919	115	1979	123
1920	96,4	1980	121
1921	66,9	1981	128
1922	117	1982	140
1923	114	1983	120
1924	133	1984	85,1
1925	88,2	1985	113
1926	113	1986	107
1927	131	1987	101
1928	144	1988	114
1929	105	1989	120
1930	90,9	1990	138
1931	165	1991	128
1932	155		
1933	165		
1934	123		
1935	118		
1936	119		

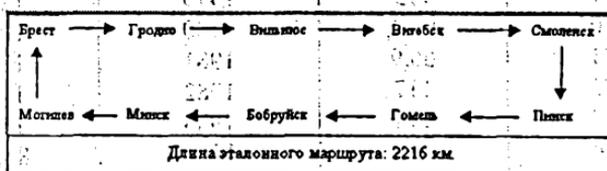
ПРИЛОЖЕНИЕ 2

Результаты работы сети Хопфилда в задаче коммивояжера

Для моделирования сети Хопфилда в задаче коммивояжера использовалась таблица расстояний между десятью городами. Вручную был проложен эталонный маршрут, который представлен в таблице 2. В таблице 3 представлены результаты работы сети Хопфилда и относительная погрешность вычислений.

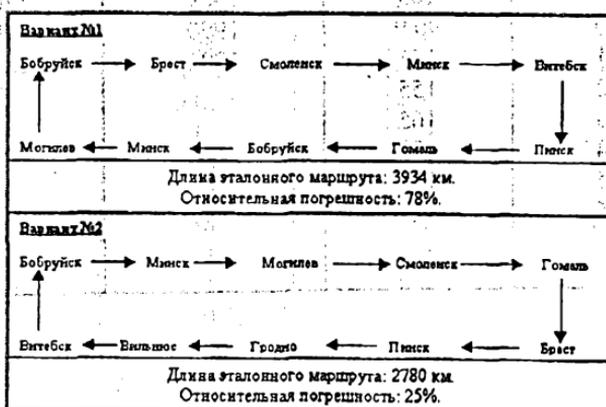
Эталонный маршрут :

Таблица 2

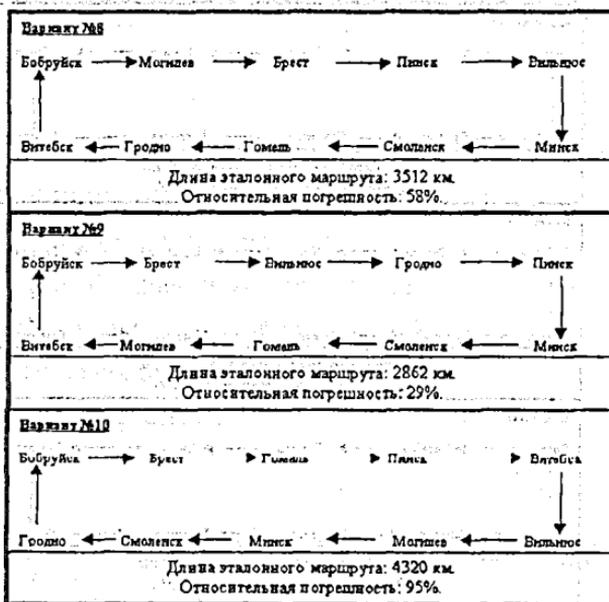


Результаты работы дискретной сети Хопфилда:

Таблица 3



<p>Вариант №3</p> <p>Бобруйск → Вильнюс → Ерст → Минск → Понск</p> <p>↑</p> <p>Витебск ← Гродно ← Смоленик ← Могилев ← Гомель</p> <p>↓</p> <p>Длина эталонного маршрута: 3586 км. Относительная погрешность: 62%.</p>
<p>Вариант №4</p> <p>Бобруйск → Вильнюс → Понск → Витебск → Гомель</p> <p>↑</p> <p>Минск ← Гродно ← Ерст ← Могилев ← Смоленик</p> <p>↓</p> <p>Длина эталонного маршрута: 3463 км. Относительная погрешность: 56%.</p>
<p>Вариант №5</p> <p>Бобруйск → Витебск → Гомель → Смоленик → Гродно</p> <p>↑</p> <p>Понск ← Минск ← Вильнюс ← Могилев ← Ерст</p> <p>↓</p> <p>Длина эталонного маршрута: 3652 км. Относительная погрешность: 65%.</p>
<p>Вариант №6</p> <p>Бобруйск → Смоленик → Могилев → Гродно → Гомель</p> <p>↑</p> <p>Вильнюс ← Понск ← Ерст ← Минск ← Витебск</p> <p>↓</p> <p>Длина эталонного маршрута: 3433 км. Относительная погрешность: 55%.</p>
<p>Вариант №7</p> <p>Бобруйск → Мина → Гомель → Смоленик → Могилев</p> <p>↑</p> <p>Понск ← Гродно ← Витебск ← Ерст ← Вильнюс</p> <p>↓</p> <p>Длина эталонного маршрута: 3605 км. Относительная погрешность: 63%.</p>



ЛИТЕРАТУРА

1. Советский энциклопедический словарь./гл. ред: А. М. Прохоров.—Москва: Советская энциклопедия.—1983.—1600с.
2. Фишбах Д. Психика и мозг // В мире науки. — 1992.— N11-12.—С.10—20.
3. Хьюбелл Д. Глаз, мозг, зрение: Пер. с англ. — М.: Мир.—1990.—239с.
4. Гольдман-Ракич П. Оперативная память и разум// В мире науки. — 1992.—N11-12. — С. 63—70.
5. Кендел Э. Хокинг Р. Биологические основы обучения и индивидуальности// В мире науки.—1992.— N11-12.— С. 43—51.
6. Зеки С. Зрительный образ в сознании и мозге// В мире науки. — 1992.— N11-12.— С.33—41.
7. Шатц К. Развивающийся мозг// В мире науки.—1992.— N11-12.— С. 23—30.
8. Головки В. А. Нейроинтеллект: проблемы и исследования// Труды X научно-технической конференции.—Брест: БПИ.—1988. — 63.—7.
9. Сепко Д. Стареющий мозг// В мире науки. —1992.— N11-12. С.93—100.
10. Фрейд Зигмунд. Психология бессознательного: Сб. произведений/ Научный ред. М. Г. Ярошевский. — М.: Просвещение. — 1989. — 448с.
11. Крик Ф., Кох К. Проблема сознания// В мире науки. —1992.— N11-12. — С.113— 120.
12. Невв Д. The Organization of Behaviour. New York: Wiley.—1949.
13. Хакен. Г. Синергетика: Пер. с англ. — М.: Мир.—1980.
14. Пригожин И. Порядок из хаоса: Пер. с англ. —М.: Прогресс.—1986.— 432с.

15. Рабинович М. К., Трубецков Д. И. Введение в теорию колебаний и волн: Учебное пособие. – М.: Наука, 1984.–432с.

16. McCulloch W., Pitts W. A logical calculus of the ideas immanent in nervous activity// Bulletin of Mathematical Biophysics. –1943.–N5.– P.115–133.

17. Розенблат Ф. Принципы нейродинамики: Персептрон и теория механизмов мозга: Пер. с англ. – М.: Мир. – 1965.–480с.

18. Widrow B., Hoff M. Adaptive switching circuits// In 1960 IRE WESCON Convention Record.–DUNNO. –1960. –P.96 – 104.

19. Minsky M., Papert S. Perceptrons: An introduction to Computational Geometry. The MIT Press.– 1969.

20. Anderson J. Neural models with cognitive implications// Basic Processes in Reading Perception and Comprehension Models.– Hillsdale, NJ: Erlbaum.– 1977.– P.27–90.

21. Kohonen T. Associative Memory: A System – Theoretical Approach. Springer Verlag.–1977.

22. Grossberg S. Adaptive pattern classification and universal recoding // Biological Cybernetics – 1976. – N. 23.– P.121–134.

23. Hopfield J. Neural Networks and physical systems with emergent collective computational abilities// Proceedings of the National Academy of Sciences USA.–1982.–79.–P.2554–2558.

24. Kohonen T. Self-organised formation of topologically correct feature maps// Biological Cybernetics. – 1982. – N43.–P.59–69.

25. Rumelhart D., Hinton G., Williams R: Learning representation by backpropagation errors// Nature.–1986.–N323.–P.533–536.

26. Дуда Р. Харт П. Распознавание образов и анализ сцен: Пер. с англ. – М.: Мир. – 1976.–511с.

27. Golovko V., Savitsky Ju., Glaischuk. Predicting Neural Net// Proceedings Int. Conf. CM NDT–D5.–Berlin:DGZfP.–1995.–P.348–353.

28. Гантмахер Ф. Теория матриц.–М.: Наука.–1988.–548с.
29. Ruediger Brause. Neuronale Netze: eine Einfuehrung in die Neuroinformatik.–Stuttgart: Teubner.–1995.–462p.
30. Golovko V., Savitsky Ju., Glaischuk V. Neural Nets Training Algorithms for Pattern Recognition and Prediction // Proceeding-Int. Conf. on Pattern Recognition.–Szczecin: Technical University, Poland.–1995.–P.167–171.
31. Golovko V., Savitsky Ju., Glaischuk V. A neural net for prediction problems // Proceedings of Int. Conf. on technical informatics. –Timisoara: University of Timisoara, Romania. – 1996.–P.49–52.
32. Головки В. А., Дунец А. П., Левонюк Д. Н. Метод обучения многослойной нейронной сети // Труды X научно-технической конференции.–Брест: БПИ.–1988.–С.8–15.
33. Lipmann R. An introduction to computing with neural nets // IEEE Acoustic, Speech and Signal Processing Magazine.–1987.–N2– P.4–22.
34. Скурихин А. Н. Нейронные сети: определения, концепции, применение.–М.: ЦНИИ управления экономики и информатики.–1991.–52с.
35. Колмогоров А. Н. Представление непрерывных функций многих переменных суперпозицией функций одной переменной и сложением // ДАН.–1958.–N.5.–С.953–956.
36. Ruediger Brause. Neuronale Netze: eine Einfuehrung in die Neuroinformatik.–Stuttgart: Teubner.–1995.–462p.
37. Rojas Raul. Theorie der neuronalen Netze. Eine systematische Einfuehrung.– Berlin: Springer-Verlag.– 1993.– 446p.
38. Hornik K., Stinchcombe M. and White H. Multilayer feedforward networks are universal approximators // Neural Networks.– 1989.–N.2 – P.359–366.
39. Maxwell T., Giles C., Lee Y., Chen H. Nonlinear Dynamics of Artificial Neural Systems // Proceedings of the Conf. On Neural Networks for Computing.–

American Institute of Physics.—1986.

40. Rumelhart D., Hinton G. and Williams R. Learning Internal Representations by errors propagation. // Parallel Distributed Processing.—Cambridge: MIT Press.—1986.

41. Hertz J., Krogh A., Palmer R. Introduction to the theory of neural computation.—Addison Wesley Publishing Company.—1991.—327p.

42. Kroese B. An introduction to Neural Networks.—Amsterdam: University of Amsterdam.—1996.—120p.

43. Поляк Б. Т. Введение в оптимизацию.—М.: Наука.—1983.—384с.

44. Темников Ф. Е., Афонин В. А., Дмитриев В. И. Теоретические основы информационной техники: Учебное пособие для вузов.—М.: Энергия.—1979.—512с.

45. Jordan M. Attractor dynamics and parallelism in a connectionist sequential machine // Proceedings of the Eighth Annual Conference of the Cognitive Science Society.—Hillsdale: Erlbaum.—1986.—P.531—546.

46. Elman J. Finding structure in time // Cognitive Science.—1990.—N.14.—P.179—211.

47. Haykin S. Neural Networks: A comprehensive Foundation.—New York: Macmillan College Publishing Company.—1994.

48. Pedersen M. Training Recurrent Networks // Proceeding of the IEEE Workshop on Neural Networks for Signal Processing VII.—New Jersey: IEEE.—1997.

49. Hinton G., McClelland J. Learning Representation by Recirculation // Proceedings of IEEE Conference on Neural Information Processing Systems.—1989.

50. Прикладная статистика: Классификация и снижение размерности: Справ. изд. / Айвазян С. А., Бухштабер В. М., Енюков И. С., Мешалкин Л. Д.; Под ред. Айвазяна С. А.—М.: Финансы и статистика.—1989.—607с.

51. Афифи А., Эйзен С. Статистический анализ: Подход с использованием ЭВМ: Пер. с англ.—М.: Мир.—1982—488с.
52. Малиновский Л. Г. Классификация объектов средствами дискриминантного анализа.—М.: Наука.—1978.—260с.
53. Cottrell G., Munro P., Zipser D. Learning Internal Representation from Gray-Scale Images: An Example of Extensional Programming // Proceedings 9th Annual Conference of the Cognitive Science Society.—1987.—P.461-473.
54. Cottrell G., Munro P., Zipser D. Image compression by back-propagation: a demonstration of extensional programming // Tech. Rep. N.TR8702.—USCD: Institute of Cognitive Sciences.—1987.
55. Hopfield J. Neurons with graded response have collective computational properties like those of two-state neurons // Proceeding of the National Academy of Science USA.—1984.—V.81.—P.3088-3092.
56. Hopfield J., Tank D. Neural computation of decisions in optimization problems // Biological Cybernetics. —1985.—V.52.—P.141-152.
57. Тэнк Д., Хопфилд Д. Коллективные вычисления в нейроподобных электронных схемах // В мире науки.—1988.—N2.—С.45-53.
58. Kosko B. Competitive Adaptive Bidirectional Associative Memories // Proceedings of the IEEE First International Conference on Neural Networks.—IEEE Press.—1987.—V.2.
59. Kosko B. Feedback Stability and Unsupervised Learning // Proceedings of the IEEE Second International Conference on Neural Networks.—IEEE Press.—1988.
60. Ackley D., Hinton G., Sejnowski T. A learning algorithm for Boltzmann machines // Cognitive Sciences. —1985.—9(1)—P.147-169.
61. Ляпунов А. М. Общая задача об устойчивости движения.—М.: Гостехиздат.—1952.
62. Воронов А. А. Титов В. К. Новогранов Б. Н. Основы теории

- автоматического регулирования и управления: Учебное пособие для вузов.— М.: Высшая школа.—1977.—232с.
63. Теория автоматического управления / Под. ред. Шаталова А. С. Учебное пособие для вузов.—М.: Высшая школа.—1977.—448с.
64. Справочник по теории автоматического управления / Под ред. Красковского А. А.—М.: Наука.—1987.—712с.
65. Меламед И. И. Нейронные сети и комбинаторная оптимизация // Автоматика и телемеханика.—1994.—N11.—С.3—40.
66. Гусак А. А. Гусак Г.М. Справочник по высшей математике: Справ.—Мн.: Навука і тэхніка.—1991.—480с.
67. Nijhuis J., Spaanenburg L. Fault tolerance of neural associative memories // Comput. and Digital Techn.—1989.—N5.—P.389—394.
68. Лосев В. В. Микропроцессорные устройства обработки информации. Алгоритмы цифровой обработки: Учебное пособие для вузов.— Мн.: Выш. шк.—1990.—132с.
69. Lam D., Carroli J. Pattern recognition using a double layer assotiative memory // Proceedings Int. Conf. on Real-Time Signal Process.—San Diego.—1989.—P.177—188.
70. Aiyer V., Niranjan M., Fallside F. A Theoretical Investigation into the Performance of the Hopfield Model // IEEE Transactions on Neural Networks.—1990.—V.2.—N2.—P.204—215.
71. Wilson G., Pawley G. On the Stability of the travelling salesman problem algorithm of Hopfield and Tank // Biological Cybernetics.—1988.—N.58.—P.63—70.
72. Васильев Ф. П. Численные методы решения экстремальных задач: Учебное пособие для вузов.— М.: Наука.—1988—552с.

СОДЕРЖАНИЕ

Предисловие	3
Глава 1. Основы естественного интеллекта	7
1.1. На стыке наук	7
1.2. Биологический нейрон	9
1.3. Нейронная организация мозга	11
1.4. Морфогенез мозга	16
1.5. Механизмы обучения	19
1.6. Самоорганизация и адаптация	23
1.6.1. Самоорганизация физических систем	24
1.6.2. Самоорганизация простейших биологических систем	27
1.6.3. Самоорганизация высших биологических систем	28
1.6.4. Самоорганизация отдельного организма	34
1.7. Искусственные нейронные системы	37
1.8. Классификация искусственных нейронных сетей	40
Глава 2. Однослойные нейронные сети	45
2.1. Формальный нейрон	45
2.2. Функции активации нейронных элементов	48
2.2.1. Линейная функция	48
2.2.2. Пороговая функция	48
2.2.3. Линейная ограниченная функция	50
2.2.4. Модифицированная пороговая функция	51
2.2.5. Сигмоидная функция	51
2.2.6. Биполярная сигмоидная функция	52
2.2.7. Гиперболический тангенс	52
2.2.8. Радиально-базисная функция	53
2.3. Нейронные сети с пороговой функцией активации	54
2.4. Правило обучения Хебба	59
2.5. Процедура обучения Розенблатта	64

2.6. Геометрическая интерпретация процедуры обучения персептрона	67
2.7. Правило обучения Видроу-Хоффа	71
2.8. Выбор адаптивного шага обучения	75
2.9. Использование псевдообратной матрицы для обучения линейных нейронных сетей	79
2.10. Анализ линейных нейронных сетей	81
2.11. Линейная ассоциативная память	83
2.12. Нелинейная ассоциативная память	87
2.13. Использование линейной нейронной сети для прогнозирования	89
Глава 3. Многослойные нейронные сети	95
3.1. Топология и анализ многослойных нейронных сетей	95
3.2. Нейронные сети высокого порядка	99
3.3. Математические основы алгоритма обратного распространения ошибки	101
3.4. Обобщенное дельта правило для различных функций активации нейронных элементов	107
3.4.1. Сигмоидная функция	107
3.4.2. Биполярная сигмоидная функция	108
3.4.3. Гиперболический тангенс	110
3.5. Алгоритм обратного распространения ошибки	111
3.6. Недостатки алгоритма обратного распространения ошибки	113
3.7. Рекомендации по обучению и архитектуре многослойных нейронных сетей.	114
3.8. Адаптивный шаг обучения	118
3.8.1. Сигмоидная функция	123
3.8.2. Гиперболический тангенс.	123
3.8.3. Биполярная сигмоидная функция	124
3.9. Модификация адаптивного шага обучения	126
3.10. Алгоритм послонного обучения	130
3.11. Сравнительный анализ алгоритмов обучения	136
3.11.1. Прогнозирование временных рядов	136
3.11.2. Распознавание цифр	139
3.11.3. Формирование циклических кодов	141
3.12. Гетерогенные нейронные сети	143

3.13. Алгоритм многократного распространения ошибки	147
3.14. Использование логарифмической функции активации	150
3.15. Применение гетерогенных нейронных сетей для прогнозирования 153	
Глава 4. Многослойные нейронные сети с обратным распространением информации	156
4.1. Рекуррентные нейронные сети	157
4.1.1. Архитектура рекуррентных сетей	157
4.1.2. Обучение рекуррентной сети	162
4.2. Рециркуляционные нейронные сети	166
4.2.1. Метод главных компонент	167
4.2.2. Архитектура рециркуляционной нейронной сети	172
4.2.3. Алгоритм обратного распространения ошибки	174
4.2.4. Алгоритм послойного обучения	179
4.2.4.1. Линейные рециркуляционные сети	180
4.2.4.2. Нелинейные рециркуляционные сети	185
4.2.5. Нормализованное правило Хебба	188
4.2.6. Обработка изображений	189
Глава 5. Релаксационные нейронные сети	193
5.1. Устойчивость динамических систем	193
5.2. Нейронная сеть Хопфилда	197
5.2.1 Архитектура нейронной сети Хопфилда	197
5.2.2. Нейронная сеть Хопфилда, как динамическая система	200
5.2.3. Энергия сети Хопфилда	203
5.2.4 Анализ аттракторов	210
5.2.5. Ассоциативная память	214
5.2.6. Функционирование сети Хопфилда	222
5.2.7 Решение задач оптимизации	226
5.2.7.1. Задача Коммивояжера	227
5.2.7.2. Машина Больцмана	231
5.3. Нейронная сеть Хэмминга	234
5.3.1. Архитектура сети	234
5.3.2 Обучение	236

Научное издание

Владимир Адамович Головко

НЕЙРОИНТЕЛЛЕКТ: ТЕОРИЯ И ПРИМЕНЕНИЕ

Редактор

Т.В. Строкач

Художественный редактор

А.П. Дунец

Компьютерный дизайн

и верстка

А.П. Дунец

Лицензия № 382 от 30.04.1999г. Подписано в печать 22.02.99. Формат 60x84/16.
Бумага CopyRex. Гарнитура Times New Roman. Усл. печ. л. 15,1 Уч.-изд. л. 16,25
Заказ №471 Тираж 500 экз. Отпечатано на ризографе Брестского
политехнического института. 224017, г. Брест, Московская, 267.

ISBN 985-6584-01-9



9 789856 584018