Основы искусственных нейронных сетей

Материал из курса: **Основы теории нейронных сетей**

Г.Э. Яхъяева http://www.intuit.ru/department/ds/neuronnets/1/

Основы искусственных нейронных сетей

Биологический прототип

Развитие искусственных нейронных сетей вдохновляется биологией. То есть, рассматривая сетевые конфигурации и алгоритмы, исследователи применяют термины, заимствованные из принципов организации мозговой деятельности. Но на этом аналогия заканчивается. Наши знания о работе мозга столь ограничены, что мало бы нашлось точно доказанных закономерностей для тех, кто пожелал бы руководствоваться ими. Поэтому разработчикам сетей приходится выходить за пределы современных биологических знаний в поисках структур, способных выполнять полезные функции. Во многих случаях это приводит к необходимости отказа от биологического правдоподобия, мозг становится просто метафорой, и создаются сети, невозможные в живой материи или требующие неправдоподобно больших допущений об анатомии и функционировании мозга.

Несмотря на то, что связь с биологией слаба и зачастую несущественна, искусственные нейронные сети продолжают сравнивать с мозгом. Их функционирование часто имеет внешнее сходство с человеческим познанием, поэтому трудно избежать этой аналогии. К сожалению, такие сравнения неплодотворны и создают неоправданные ожидания, неизбежно ведущие к разочарованию.

Нервная система человека, построенная из элементов, называемых нейронами, имеет ошеломляющую сложность. Около 10^{11} нейронов участвуют в примерно 10^{15} передающих связях, имеющих длину метр и более. Каждый нейрон обладает многими свойствами, общими с другими органами тела, но ему присущи абсолютно уникальные способности: принимать, обрабатывать и передавать электрохимические сигналы по нервным путям, которые образуют коммуникационную систему мозга.

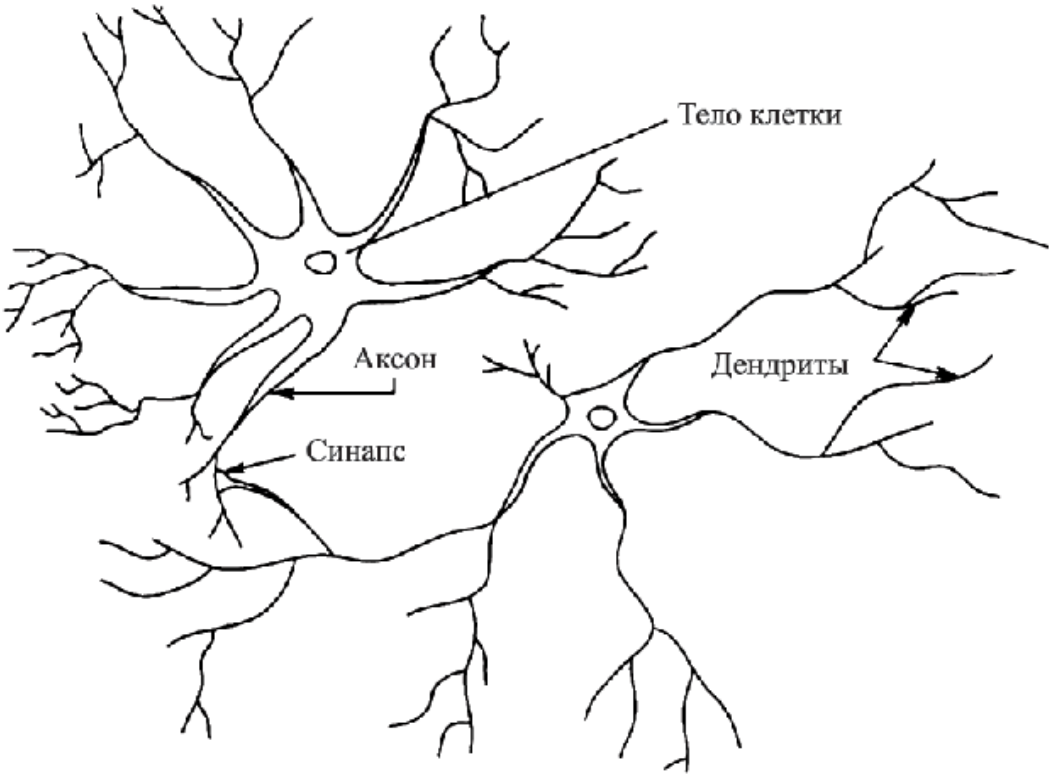


Рис. 1.1.

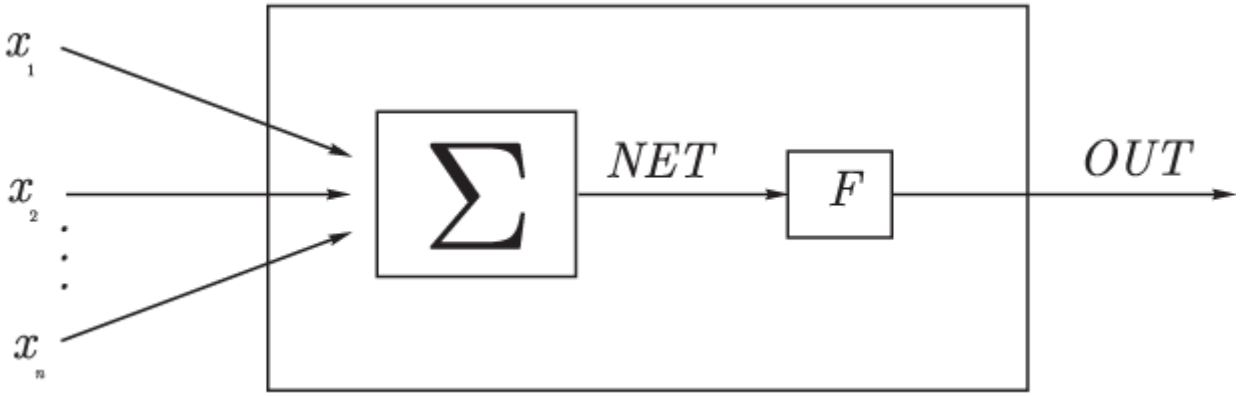
На рис. 1.1 показана структура пары типичных биологических нейронов. Дендриты идут от тела

нервной клетки к другим нейронам, где они принимают сигналы в точках соединения, называемых синапсами. Принятые синапсом входные сигналы передаются к телу нейрона. Здесь они суммируются, причем одни входы стремятся возбудить нейрон, другие — воспрепятствовать его возбуждению.

Когда суммарное возбуждение в теле нейрона превышает некоторый порог, нейрон возбуждается, посылая по аксону сигнал другим нейронам. У этой основной функциональной схемы много усложнений и исключений, тем не менее, большинство искусственных нейронных сетей моделируют лишь эти простые свойства.

Искусственный нейрон

Искусственный нейрон имитирует в первом приближении свойства биологического нейрона. На вход искусственного нейрона поступает некоторое множество сигналов, каждый из которых является выходом другого нейрона. Каждый вход умножается на соответствующий вес, аналогичный синаптической силе, и все произведения суммируются, определяя уровень активации нейрона.



На рис. 1.2 представлена модель, реализующая эту идею. Множество входных сигналов, обозначенных x1,x2,...,xn, поступает на искусственный нейрон. Эти входные сигналы, в совокупности обозначаемые вектором , соответствуют сигналам, приходящим в синапсы биологического нейрона. Каждый сигнал умножается на соответствующий вес ?1,?2,...,?n, и поступает на суммирующий блок, обозначенный ?. Каждый вес соответствует "силе" одной биологической синаптической связи. (Множество весов в совокупности обозначается вектором .) Суммирующий блок, соответствующий телу биологического элемента, складывает взвешенные входы алгебраически, создавая выход, который мы будем называть NET. В векторных обозначениях это может быть компактно записано следующим образом: NET=XW
Сигнал NET далее, как правило, преобразуется активационной функцией F и дает выходной нейронный сигнал OUT. Активационная функция может быть обычной линейной функцией OUT=F (NET)
где F — константа, пороговой функцией

$$OUT = \begin{cases} 1, & \text{если } NET > T; \\ 0, & \text{если } NET \leqslant T \end{cases}$$

где T — некоторая постоянная пороговая величина, или же функцией, более точно моделирующей нелинейную передаточную характеристику биологического нейрона и предоставляющей нейронной сети большие возможности.

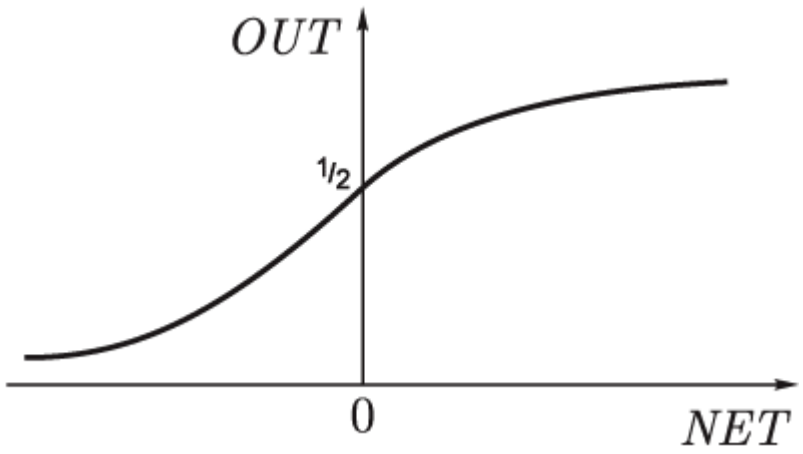


Рис. 1.3.

На рис. 1.2 блок, обозначенный F, принимает сигнал NET и выдает сигнал OUT. Если блок F сужает диапазон изменения величины NET так, что при любых значениях NET значения OUT принадлежат некоторому конечному интервалу, то F называется **"сжимающей" функцией**. В качестве "сжимающей" функции часто используется логистическая или "сигмоидальная" (S-образная) функция, показанная на рис. 1.3. Эта функция математически выражается как $F(x) = 1/(1 + e^{-x})$.

$$OUT = \frac{1}{1 + e^{-NET}}.$$

Таким образом,

По аналогии с электронными системами активационную функцию можно считать нелинейной усилительной характеристикой искусственного нейрона. Коэффициент усиления вычисляется как отношение приращения величины OUT к вызвавшему его небольшому приращению величины NET.

Он выражается наклоном кривой при определенном уровне возбуждения и изменяется от малых значений при больших отрицательных возбуждениях (кривая почти горизонтальна) до максимального значения при нулевом возбуждении и снова уменьшается, когда возбуждение становится большим положительным. С. Гроссберг (1973) обнаружил, что подобная нелинейная характеристика решает поставленную им дилемму шумового насыщения. Каким образом одна и та же сеть может обрабатывать как слабые, так и сильные сигналы? Слабые сигналы нуждаются в большом сетевом усилении, чтобы дать пригодный к использованию выходной сигнал. Однако усилительные каскады с большими коэффициентами усиления могут привести к насыщению выхода шумами усилителей (случайными флуктуациями), которые присутствуют в любой физически реализованной сети. Сильные входные сигналы, в свою очередь, также будут приводить к насыщению усилительных каскадов, исключая возможность полезного использования выхода. Центральная область логистической функции, имеющая большой коэффициент усиления, решает проблему обработки слабых сигналов, в то время как области с падающим усилением на положительном и отрицательном концах подходят для больших возбуждений. Таким образом, нейрон функционирует с большим усилением в широком диапазоне уровня входного сигнала

$$OUT = \frac{1}{1 + e^{-NET}} = F(NET).$$

Другой широко используемой активационной функцией является гиперболический тангенс. По форме она сходна с логистической функцией и часто используется биологами в качестве математической модели активации нервной клетки. В качестве активационной функции искусственной нейронной сети она записывается следующим образом:

$$OUT = \mathrm{th}(x).$$

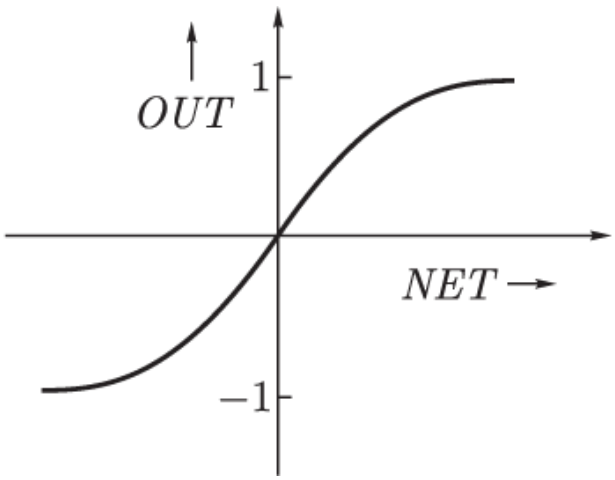


Рис. 1.4.

Подобно логистической функции гиперболический тангенс является S-образной функцией, но он симметричен относительно начала координат, и в точке NET=0 значение выходного сигнала OUT равно нулю (см. рис. 1.4). В отличие от логистической функции, гиперболический тангенс принимает значения различных знаков, и это его свойство применяется для целого ряда сетей.

Рассмотренная простая модель искусственного нейрона игнорирует многие свойства своего биологического двойника. Например, она не принимает во внимание задержки во времени, которые воздействуют на динамику системы. Входные сигналы сразу же порождают выходной сигнал. И, что более важно, она не учитывает воздействий функции частотной модуляции или синхронизирующей функции биологического нейрона, которые ряд исследователей считают решающими в нервной деятельности естественного мозга.

Несмотря на эти ограничения, сети, построенные из таких нейронов, обнаруживают свойства, сильно напоминающие биологическую систему. Только время и исследования смогут ответить на вопрос, являются ли подобные совпадения случайными или же они есть следствие того, что в модели верно схвачены важнейшие черты биологического нейрона.

Однослойные искусственные нейронные сети

Хотя один нейрон и способен выполнять простейшие процедуры распознавания, но для серьезных нейронных вычислений необходимо соединять нейроны в сети. Простейшая сеть состоит из группы нейронов, образующих слой, как показано в правой части рис. 1.5. Отметим, что вершины-круги слева служат лишь для распределения входных сигналов. Они не выполняют каких-либо вычислений и поэтому не будут считаться слоем. Для большей наглядности обозначим их кругами, чтобы отличать их от вычисляющих нейронов, обозначенных квадратами. Каждый элемент из множества X входов отдельным весом соединен с каждым искусственным нейроном. А каждый нейрон выдает взвешенную сумму входов в сеть. В искусственных и биологических сетях многие соединения могут отсутствовать, но здесь они показаны все для демонстрации общей картины. Могут существовать также соединения между выходами и входами элементов в слое.

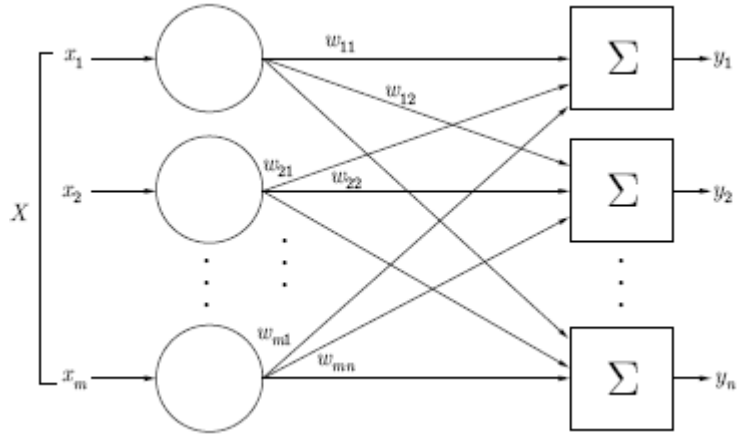


Рис. 1.5.

Удобно считать веса элементами матрицы W. Матрица имеет m строк и n столбцов, где m — число входов, а n — число нейронов. Например, $w_{2,3}$ — это вес, связывающий второй вход с третьим нейроном. Таким образом, вычисление выходного вектора N, компонентами которого являются выходы OUT нейронов, сводится к матричному умножению N=XW, где N и X — векторы-строки.

Многослойные искусственные нейронные сети

Более крупные и сложные нейронные сети обладают, как правило, и большими вычислительными возможностями. Хотя созданы сети всех конфигураций, какие только можно себе представить, послойная организация нейронов копирует слоистые структуры определенных отделов мозга. Оказалось, что такие многослойные сети обладают большими возможностями, чем однослойные, и в последние годы были разработаны алгоритмы для их обучения. Многослойные сети могут строиться из каскадов слоев. Выход одного слоя является входом для последующего слоя. Подобная сеть показана на рис. 1.6 и снова изображена со всеми соединениями. Многослойные сети не могут привести к увеличению вычислительной мощности по сравнению с однослойной сетью, если активационная функция между слоями линейна. Вычисление выхода слоя заключается в умножении входного вектора на первую весовую матрицу с последующим умножением (если отсутствует нелинейная активационная функция) результирующего вектора на вторую весовую матрицу

$$OUT = (XW_1)W_2.$$

Так как умножение матриц ассоциативно, то $(XW_1)W_2 = X(W_1W_2)$

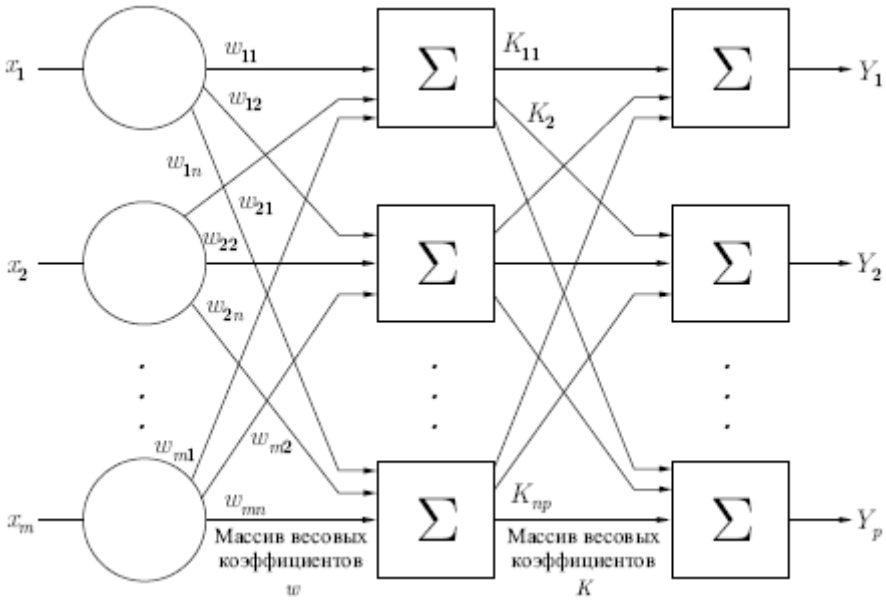


Рис. 1.6.

Это показывает, что двухслойная линейная сеть эквивалентна одному слою с весовой матрицей, равной произведению двух весовых матриц. Следовательно, любая многослойная линейная сеть может быть заменена эквивалентной однослойной сетью. Однако однослойные сети весьма ограничены по своим вычислительным возможностям. Таким образом, для расширения возможностей сетей по сравнению с однослойной сетью необходима нелинейная активационная функция.

У сетей, рассмотренных до сих пор, не было обратных связей, т. е. соединений, идущих от выходов некоторого слоя к входам этого же слоя или предшествующих слоев. Этот специальный класс сетей, называемых сетями без обратных связей или сетями прямого распространения, представляет большой интерес и широко используется. Сети более общего вида, имеющие соединения от выходов к входам, называются сетями с обратными связями. У сетей без обратных связей нет памяти, их выход полностью определяется текущими входами и значениями весов. В некоторых конфигурациях сетей с обратными связями предыдущие значения выходов возвращаются на входы; выход, следовательно, определяется как текущим входом, так и предыдущими выходами. Поэтому сети с обратными связями могут обладать свойствами, сходными с кратковременной человеческой памятью, где сетевые выходы тоже частично зависят от предыдущих входов.

К сожалению, нет общепринятого способа подсчета числа слоев в сети. Многослойная сеть состоит, как показано на рис. 1.6, из чередующихся множеств нейронов и весов. Ранее, в связи с рис. 1.5, уже говорилось, что входной слой не выполняет суммирования. Эти нейроны служат лишь в качестве разветвлений для первого множества весов и не влияют на вычислительные возможности сети. По этой причине первый слой не принимается во внимание при подсчете слоев, и сеть, подобная изображенной на рисунке 1.6, считается двуслойной, так как только два слоя выполняют вычисления. Далее, веса слоя считаются связанными со следующими за ними нейронами. Следовательно, слой состоит из множества весов со следующими за ними нейронами, суммирующими взвешенные сигналы.

Обучение искусственных нейронных сетей

Среди всех интересных свойств искусственных нейронных сетей ни одно не захватывает так воображения, как их способность к обучению. Их обучение до такой степени напоминает процесс интеллектуального развития человеческой личности, что может показаться, будто нами достигнуто глубокое понимание этого процесса. Но, проявляя осторожность, следует сдерживать эйфорию. Возможности обучения искусственных нейронных сетей ограничены, и нужно решить много сложных задач, чтобы определить, находимся ли мы на правильном пути.

Цель обучения

Сеть обучается, чтобы для некоторого множества входов давать желаемое (или, по крайней мере, сообразное с ним) множество выходов. Каждое такое входное (или выходное) множество рассматривается как вектор. Обучение осуществляется путем последовательного предъявления входных векторов с одновременной подстройкой весов в соответствии с определенной процедурой. В процессе обучения веса сети постепенно становятся такими, чтобы каждый входной вектор вырабатывал выходной вектор.

Обучение с учителем

Различают алгоритмы обучения с учителем и без учителя. Обучение с учителем предполагает, что для каждого входного вектора существует целевой вектор, представляющий собой требуемый выход. Вместе они называются обучающей парой. Обычно сеть обучается на некотором числе таких обучающих пар. Предъявляется выходной вектор, вычисляется выход сети и сравнивается с соответствующим целевым вектором, разность (ошибка) с помощью обратной связи подается в сеть, и веса изменяются в соответствии с алгоритмом, стремящимся минимизировать ошибку. Векторы обучающего множества предъявляются последовательно, ошибки вычисляются и веса подстраиваются для каждого вектора до тех пор, пока ошибка по всему обучающему массиву не достигнет приемлемо низкого уровня.

Обучение без учителя

Несмотря на многочисленные прикладные достижения, обучение с учителем критиковалось за свою биологическую неправдоподобность. Трудно вообразить обучающий механизм в мозге, который бы сравнивал желаемые и действительные значения выходов, выполняя коррекцию с помощью обратной связи. Обучение без учителя является намного более правдоподобной моделью обучения для биологической системы. Развитая Кохоненом и многими другими, она не нуждается в целевом векторе для выходов и, следовательно, не требует сравнения с предопределенными идеальными ответами. Обучающее множество состоит лишь из входных векторов. Обучающий алгоритм подстраивает веса сети так, чтобы получались согласованные выходные векторы, т. е. чтобы предъявление достаточно близких входных векторов давало одинаковые выходы. Процесс обучения, следовательно, выделяет статистические свойства обучающего множества и группирует сходные векторы в классы. Предъявление на вход вектора из данного класса даст определенный выходной вектор, но до обучения невозможно предсказать, какой выход будет производиться данным классом входных векторов. Следовательно, выходы подобной сети должны трансформироваться в некоторую понятную форму, обусловленную процессом обучения. Это не является серьезной проблемой. Обычно не сложно идентифицировать связь между входом и выходом, установленную сетью.

Алгоритмы обучения

Большинство современных алгоритмов обучения выросло из концепций Д.О. Хэбба. Он предложил модель обучения без учителя, в которой синаптическая сила (вес) возрастает, если активированы оба нейрона, источник и приемник. Таким образом, часто используемые пути в сети усиливаются и феномены привычки и обучения через повторение получают объяснение.

В искусственной нейронной сети, использующей обучение по Хэббу, наращивание весов определяется произведением уровней возбуждения передающего и принимающего нейронов. Это можно записать как

$$w_{ij}(n+1) = w(n) + \alpha OUT_i OUT_j,$$

где $w_{ij}(n)$ — значение веса от нейрона i к нейрону j до подстройки, $w_{ij}(n+1)$ — значение веса от нейрона i к нейрону j после подстройки, — коэффициент скорости обучения, — выход нейрона и вход нейрона , — выход нейрона .

Сети, использующие обучение по Хэббу, конструктивно развивались, однако за последние 20 лет появились и разрабатывались более эффективные алгоритмы обучения. В частности, были развиты алгоритмы обучения с учителем, приводящие к сетям с более широким диапазоном характеристик обучающих входных образов и большими скоростями обучения, чем использующие простое обучение по Хэббу.

Искусственный нейрон

Искусственный нейрон имитирует в первом приближении свойства биологического нейрона. На вход искусственного нейрона поступает некоторое множество сигналов, каждый из которых является выходом другого нейрона. Каждый вход умножается на соответствующий вес, аналогичный синаптической силе, и все произведения суммируются, определяя уровень активации нейрона.

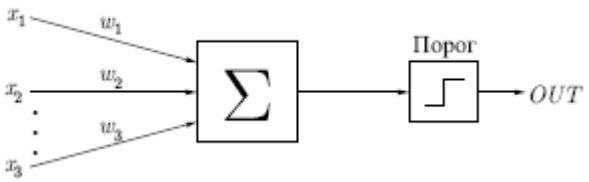


Рис. 2.1.

Первое систематическое изучение искусственных нейронных сетей было предпринято Маккалокком и Питтсом в 1943 г. Позднее они исследовали сетевые парадигмы для распознавания изображений, подвергаемых сдвигам и поворотам. Простая нейронная модель, показанная на рис. 2.1, использовалась в большей части их работ. Элемент ? умножает каждый вход x на вес w и суммирует взвешенные входы. Если полученная сумма больше заданного порогового значения, выход равен единице, в противном случае — нулю. Эти системы (и множество им подобных) получили название персептронов. Они состоят из одного слоя искусственных нейронов, соединенных с помощью весовых коэффициентов с множеством входов (см. рис. 2.2), хотя, в принципе, описываются и более сложные системы. В 60-е годы персептроны вызвали большой интерес и оптимизм. Одной из первых искусственных сетей, способных к перцепции (восприятию) и формированию реакции на воспринятый раздражитель, явился PERCEPTRON Розенблатта (F.Rosenblatt, 1957). Персептрон рассматривался его автором не как конкретное техническое (вычислительное) устройство, а как модель работы мозга. Розенблатт называл такую нейронную сеть трехслойной, однако, по современной терминологии, представленная сеть обычно называется однослойной, так как имеет только один слой нейропроцессорных элементов.

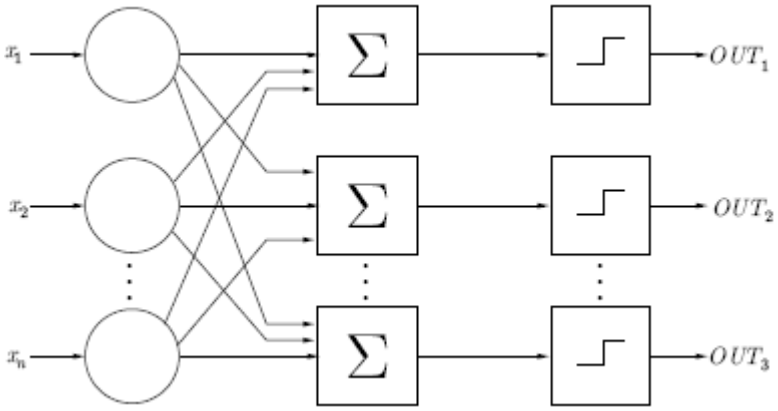


Рис. 2.2.

В Корнеллской авиационной лаборатории была разработана электротехническая модель персептрона MARK-1, которая содержала 8 выходных элементов. На этом персептроне была проведена серия экспериментов по распознаванию букв алфавита и геометрических образов.

Ф. Розенблатт доказал замечательную теорему об обучении персептронов (которую мы рассмотрим на следующей лекции). Д. Уидроу дал ряд убедительных демонстраций систем персептронного типа, и исследователи во всем мире стремились изучить возможности этих систем. Первоначальная эйфория сменилась разочарованием, когда оказалось, что персептроны не способны обучаться решению ряда простых задач. Минский строго проанализировал эту проблему и показал, что имеются жесткие ограничения того, что могут выполнять однослойные персептроны, и, следовательно, того, чему они могут обучаться. Так как в то время методы обучения многослойных сетей не были известны, исследователи занялись более многообещающими проектами, и исследования в области нейронных сетей пришли в упадок. Недавнее открытие методов обучения многослойных сетей привело к возрождению интереса и возобновлению исследований.

Работа М.Л.Минского, возможно, и охладила пыл энтузиастов персептрона, но обеспечила время для необходимой консолидации и развития лежащей в основе теории. Важно отметить, что анализ Минского не был опровергнут. Он остается актуальным исследованием и должен непременно учитываться как часть базовых знаний, чтобы ошибки 60-х годов не повторились. Несмотря на свои ограничения, персептроны широко изучались. Теория персептронов является основой для многих других типов искусственных нейронных сетей, персептроны иллюстрируют важные принципы. В силу этих причин они являются логической исходной точкой для изучения искусственных нейронных сетей.

Персептронная представляемость

Доказательство теоремы обучения персептрона показало, что персептрон способен научиться всему, что он способен представлять. Важно при этом уметь различать представляемость и обучаемость. Понятие представляемости относится к способности персептрона (или другой сети) моделировать определенную функцию. Обучаемость же требует наличия систематической процедуры настройки весов сети для реализации этой функции.

Для иллюстрации проблемы представляемости допустим, что у нас есть множество карт, помеченных цифрами от 0 до 9. Допустим также, что мы обладаем гипотетической машиной, способной отличать карты с нечетным номером от карт с четным номером и зажигающей индикатор на своей панели при предъявлении карты с нечетным номером. Представима ли такая машина персептроном? То есть возможно ли сконструировать персептрон и настроить его веса (неважно, каким образом) так, чтобы он обладал такой же разделяющей способностью? Если это достижимо, то говорят, что персептрон способен представлять желаемую машину. Мы увидим, что возможности представления однослойными персептронами весьма ограниченны. Имеется много простых машин, которые не могут быть представлены персептроном, независимо от того, как настраиваются его веса.

Проблема функции ИСКЛЮЧАЮЩЕГО ИЛИ

Один из самых пессимистических результатов М.Л. Минского гласит, что однослойный персептрон не может воспроизвести такую простую функцию, как ИСКЛЮЧАЮЩЕЕ ИЛИ. Это функция от двух аргументов, каждый из которых может быть нулем или единицей. Она принимает значение единицы, когда один из аргументов равен единице (но не оба).

Линейная разделимость

Имеется обширный класс функций, не реализуемых однослойной сетью. Об этих функциях говорят, что они являются линейно неразделимыми: они-то и накладывают определенные ограничения на возможности однослойных сетей.

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

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

Преодоление ограничения линейной разделимости

К концу 1960-х годов проблема линейной разделимости была хорошо понята. К тому же, было известно, что это серьезное ограничение представляемости однослойными сетями можно

преодолеть, добавив дополнительные слои. Например, двухслойные сети можно получить каскадным соединением двух однослойных сетей. Они способны выполнять более общие классификации, отделяя те точки, которые содержатся в выпуклых ограниченных или неограниченных областях. Область называется выпуклой, если для любых двух ее точек соединяющий их отрезок целиком лежит в области. Область называется ограниченной, если ее можно заключить в некоторый круг. Неограниченную область невозможно заключить внутрь круга (например, область между двумя параллельными линиями). Примеры выпуклых ограниченных и неограниченных областей представлены на рис. 2.6.

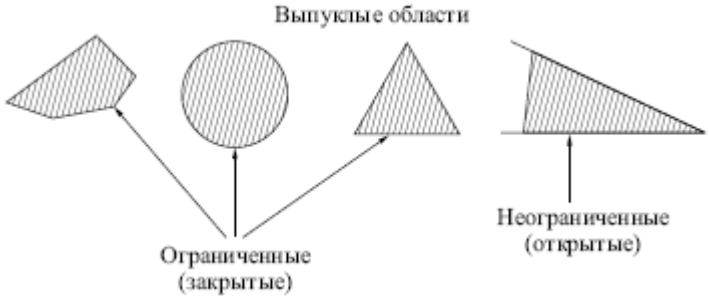


Рис. 2.6.

Чтобы уточнить требование выпуклости, рассмотрим простую двуслойную сеть с двумя входами, которые подведены к двум нейронам первого слоя, соединенными с единственным нейроном в слое 2 (см. рис. 2.7а). Пусть порог выходного нейрона равен 0,75, а оба его веса равны 0,5. В этом случае для того, чтобы порог был превышен и на выходе появилась единица, требуется, чтобы оба нейрона первого уровня на выходе имели единицу. Таким образом, выходной нейрон реализует логическую функцию И. На рис. 2.7а каждый нейрон слоя 1 разбивает плоскость XOY на две полуплоскости, один обеспечивает единичный выход для входов ниже верхней линии, другой — для входов выше нижней линии. На рис. 2.7б показан результат такого двойного разбиения, где выходной сигнал нейрона второго слоя равен единице только внутри V-образной области. Аналогично, во втором слое может быть использовано три нейрона с дальнейшим разбиением плоскости и созданием области треугольной формы. Включением достаточного числа нейронов во входной слой может быть образован выпуклый многоугольник любой желаемой формы. Все такие многогранники выпуклы, так как они образованы с помощью операции И над областями, задаваемыми линиями: следовательно, только выпуклые области и возникают. Точки, не составляющие выпуклой области, не могут быть отделены от других точек плоскости двухслойной сетью.

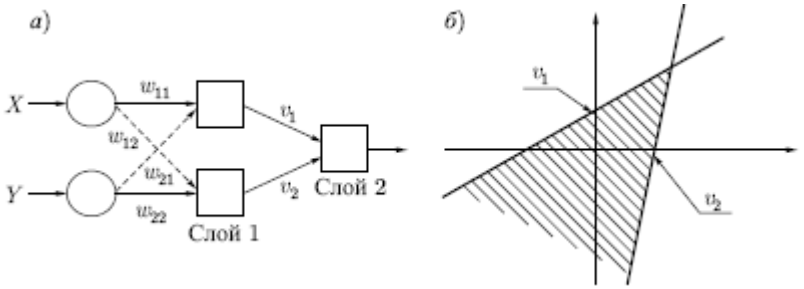


Рис. 2.7.

Нейрон второго слоя не ограничен функцией И. Он может реализовывать многие другие функции при подходящем выборе весов и порога. Например, можно сделать так, чтобы единичный выход любого из нейронов первого слоя приводил к появлению единицы на выходе нейрона второго слоя, реализовав тем самым логическое ИЛИ. Например, имеется 16 двоичных функций от двух переменных. Если выбирать подходящим образом веса и порог, то можно воспроизвести 14 из них (все, кроме ИСКЛЮЧАЮЩЕЕ ИЛИ и ИСКЛЮЧАЮЩЕЕ НЕТ).

Входы не обязательно должны быть двоичными. Вектор непрерывных входов может представлять собой произвольную точку на плоскости XOY. В этом случае мы имеем дело со способностью сети разбивать плоскость на непрерывные области, а не с разделением дискретных множеств точек. Для всех этих функций, однако, линейная разделимость показывает, что выход нейрона второго слоя равен единице только в части плоскости XOY, ограниченной многоугольной областью. Поэтому для разделения плоскостей P и Q необходимо, чтобы все P лежали внутри выпуклой многоугольной области, не содержащей точек Q (или наоборот).

Трехслойная сеть, впрочем, есть более общий случай. Ее классифицирующие возможности ограничены лишь числом искусственных нейронов и весов. Ограничения на выпуклость отсутствуют. Теперь нейрон третьего слоя принимает в качестве входа набор выпуклых многоугольников, и их логическая комбинация может быть невыпуклой. На рис. 2.8б иллюстрируется ситуация, когда два треугольника A и B, скомбинированные с помощью функций "A и не B", задают невыпуклую область. При добавлении нейронов и весов число сторон многоугольников может неограниченно возрастать. Это позволяет аппроксимировать область любой формы с любой точностью. Вдобавок, не все выходные области второго слоя должны пересекаться. Возможно, следовательно, объединять различные области, выпуклые и невыпуклые, выдавая на выходе единицу всякий раз, когда входной вектор принадлежит одной из них.

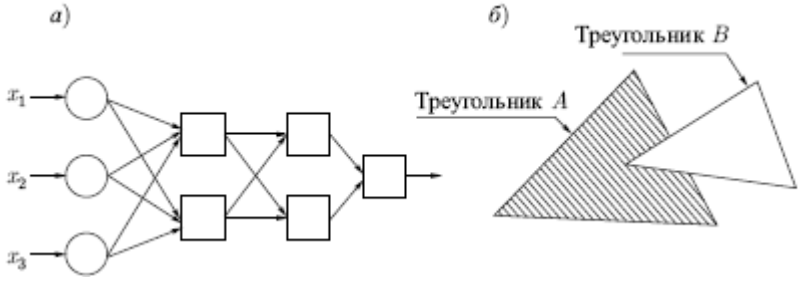


Рис. 2.8.

Обучение персептрона

Способность искусственных нейронных сетей к обучению является их наиболее интригующим свойством. Подобно биологическим системам, которые они моделируют, эти нейронные сети сами совершенствуют себя в результате попыток создать лучшую модель поведения.

Используя критерий линейной разделимости, можно решить, способна ли однослойная нейронная сеть реализовывать требуемую функцию. Даже в том случае, когда ответ положительный, это принесет мало пользы, если у нас нет способа найти нужные значения для весов и порогов. Чтобы сеть представляла практическую ценность, нужен систематический метод (алгоритм) для вычисления этих значений. Ф.Розенблатт создал такой метод в своем алгоритме обучения персептрона и доказал: персептрон может быть обучен всему, что он может реализовывать.

Обучение может быть с учителем или без него. Для обучения с учителем нужен "внешний" учитель, который оценивал бы поведение системы и управлял ее последующими модификациями. При обучении без учителя, которое будет рассмотрено на последующих лекциях, сеть путем самоорганизации делает требуемые изменения. Обучение персептрона является обучением с учителем.

Алгоритм обучения персептрона может быть реализован на цифровом компьютере или другом электронном устройстве, и сеть становится в определенном смысле самоподстраивающейся. По этой причине процедуру подстройки весов обычно называют "обучением" и говорят, что сеть "обучается". Доказательство Розенблатта стало основной вехой и дало мощный импульс исследованиям в этой области. Сегодня в той или иной форме элементы алгоритма обучения персептрона встречаются во многих сетевых парадигмах. Несмотря на то, что возможности многослойных сетей были известны давно, в течение многих лет не было теоретически обоснованного алгоритма для настройки их весов. В последующих главах мы детально изучим многослойные обучающие алгоритмы, но сейчас достаточно понимать суть проблемы и знать, что исследования привели к определенным результатам.

Эффективность запоминания

Серьезные вопросы существуют относительно эффективности запоминания информации в персептроне (или любых других нейронных сетях) по сравнению с обычной компьютерной памятью и методами поиска информации в ней. Например, в компьютерной памяти можно хранить все входные образы вместе с классифицирующими битами. Компьютер должен найти требуемый образ и дать его классификацию. Многочисленные и хорошо известные методы могли бы применяться для ускорения поиска. Если точное соответствие не найдено, то для ответа может быть использовано правило ближайшего соседа.

Число битов, необходимое для хранения этой же информации в весах персептрона, может быть значительно меньшим по сравнению с методом обычной компьютерной памяти, если образы допускают экономичную запись. Однако М.Л.Минский построил патологические примеры, в которых число битов, требуемых для представления весов, растет в зависимости от размерности

задачи быстрее, чем экспоненциально. В этих случаях требования к памяти быстро становятся невыполнимыми. Если, как он предположил, эта ситуация не является исключением, то персептроны часто могут быть ограничены только малыми задачами. Насколько общими являются такие неподатливые множества образов? Вопрос остается открытым и относится ко всем нейронным сетям. Поиски ответа чрезвычайно важны для дальнейших исследований в этой области.

Первое правило. Если на выходе персептрона получен 0, а правильный ответ равен 1, то необходимо увеличить веса связей между одновременно активными нейронами. При этом выходной персептрон считается активным. Второй тип ошибки: на выходе персептрона — 1, а правильный ответ равен нулю. Для обучения правильному решению данного примера следует уменьшить сумму в правой части (1). Следовательно, необходимо уменьшить веса связей при тех переменных, которые равны 1 (поскольку нет смысла уменьшать веса связей при равных нулю переменных). Необходимо также провести эту процедуру для всех активных нейронов предыдущих слоев. В результате получаем второе правило.

Второе правило. Если на выходе персептрона получена единица, а правильный ответ равен нулю, то необходимо уменьшить веса связей между одновременно активными нейронами.

Таким образом, процедура обучения сводится к последовательному перебору всех примеров обучающего множества с применением правил обучения для ошибочно решенных примеров. Если после очередного цикла предъявления всех примеров окажется, что все они решены правильно, то процедура обучения завершается.

Нерассмотренными остались два вопроса. Первый — о сходимости процедуры обучения. Второй — на сколько нужно увеличивать (уменьшать) веса связей при применении правил обучения.

Ответ на первый вопрос дают следующие теоремы.

Теорема о сходимости персептрона. Если существует вектор параметров , при котором персептрон правильно решает все примеры обучающей выборки, то при обучении персептрона по вышеописанному алгоритму решение будет найдено за конечное число шагов.

Теорема о "зацикливании" персептрона.Если не существует вектора параметров , при котором персептрон правильно решает все примеры обучающей выборки, то при обучении персептрона по данному правилу через конечное число шагов вектор весов начнет повторяться.

Таким образом, данные теоремы утверждают, что, запустив процедуру обучения персептрона, через конечное время мы либо получим обучившийся персептрон, либо ответ, что данный персептрон поставленной задаче обучиться не может.

Двуслойность персептрона

Алгоритм обучения персептрона возможно использовать и для многослойных персептронов. Однако теоремы о сходимости и зацикливании персептрона, приведенные выше, верны только при обучении однослойного персептрона — или многослойного персептрона при условии, что обучаются только веса персептрона, стоящего в последнем слое сети. В случае произвольного многослойного персептрона они не работают. Следующий пример демонстрирует основную проблему, возникающую при обучении многослойных персептронов.

Пусть веса всех слоев персептрона в ходе обучения сформировались так, что все примеры обучающего множества, кроме первого, решаются правильно. При этом правильным ответом первого примера является 1. Все входные сигналы персептрона последнего слоя равны нулю. В этом случае первое правило не дает результата, поскольку все нейроны предпоследнего слоя не активны. Существует множество способов решать эту проблему. Однако все эти методы не являются регулярными и не гарантируют сходимость многослойного персептрона к решению, даже при условии, что такое решение существует.

В действительности, проблема настройки (обучения) многослойного персептрона решается следующей теоремой.

Теорема о двуслойности персептрона. Любой многослойный персептрон может быть представлен в виде двуслойного персептрона с необучаемыми весами первого слоя.

Трудности с алгоритмом обучения персептрона

Иногда бывает сложно определить, выполнено ли условие разделимости для конкретного обучающего множества. Кроме того, во многих встречающихся на практике ситуациях входы часто меняются во времени и могут быть разделимы в один момент времени и неразделимы - в другой. В доказательстве алгоритма обучения персептрона ничего не говорится также о том,

сколько шагов требуется для обучения сети. Мало утешительного знать, что обучение закончится за конечное число шагов, если необходимое для этого время сравнимо с геологической эпохой. Кроме того, не доказано, что персептронный алгоритм обучения более быстр по сравнению с простым перебором всех возможных значений весов, и в некоторых случаях этот примитивный подход может оказаться лучше.

На эти вопросы никогда не находилось удовлетворительного ответа, они относятся к природе обучающего материала. В различной форме они возникнут на последующих лекциях, где рассматриваются другие сетевые парадигмы. Ответы для современных сетей, как правило, не более удовлетворительны, чем для персептрона. Эти проблемы являются важной областью современных исследований.

Переобучение и обобщение

Одна из наиболее серьезных трудностей алгоритма обратного распространения заключается в том, что таким образом мы минимизируем не ту ошибку, которую на самом деле нужно минимизировать, — ошибку, которую можно ожидать от сети, когда ей будут подаваться совершенно новые наблюдения. Иначе говоря, мы хотели бы, чтобы нейронная сеть обладала способностью обобщать результат на новые наблюдения. В действительности, сеть обучается минимизировать ошибку на обучающем множестве, и в отсутствие идеального и бесконечно большого обучающего множества это совсем не то же самое, что минимизировать "настоящую" ошибку на поверхности ошибок в заранее неизвестной модели явления.

Сильнее всего это различие проявляется в проблеме переобучения, или слишком близкой подгонки. Это явление проще будет продемонстрировать не для нейронной сети, а на примере аппроксимации посредством полиномов, — при этом суть явления абсолютно та же.

Полином (или многочлен) — это выражение, содержащее только константы и целые степени независимой переменной. Графики полиномов могут иметь различную форму, причем чем выше степень многочлена (и, тем самым, чем больше членов в него входит), тем более сложной может быть эта форма. Если у нас есть некоторые данные, мы можем попробовать подогнать к ним полиномиальную кривую (модель) и получить, таким образом, объяснение для имеющейся зависимости. Наши данные могут быть зашумлены, поэтому нельзя считать, что самая лучшая модель задается кривой, которая в точности проходит через все имеющиеся точки. Полином низкого порядка может быть недостаточно гибким средством для аппроксимации данных, в то время как полином высокого порядка может оказаться чересчур гибким и будет точно следовать данным, принимая при этом форму замысловатую и не имеющую никакого отношения к реальной зависимости.

У нейронной сети проблема точно такая же. Сети с большим числом весов моделируют более сложные функции и, следовательно, склонны к переобучению. Сеть же с небольшим числом весов может оказаться недостаточно гибкой для того, чтобы смоделировать имеющуюся зависимость. Например, сеть без промежуточных слоев моделирует обычную линейную функцию.

Как же выбрать "правильную" степень сложности для сети? Почти всегда более сложная сеть дает меньшую ошибку, но это может свидетельствовать не о хорошем качестве модели, а о переобучении. Выход состоит в том, чтобы использовать механизм контрольной кросс-проверки. Мы резервируем часть обучающих наблюдений и не используем их в обучении по алгоритму обратного распространения. Вместо этого, по мере работы алгоритма, они используются для независимого контроля результата. В самом начале работы ошибка сети на обучающем и контрольном множестве будет одинаковой (если они существенно отличаются, то, вероятно, разбиение всех наблюдений на два множества было неоднородно). По мере того как сеть обучается, ошибка обучения, естественно, убывает, и, пока обучение уменьшает действительную функцию ошибок, ошибка на контрольном множестве также будет убывать. Если же контрольная ошибка перестала убывать или даже стала расти, значит, сеть начала слишком близко аппроксимировать данные и обучение следует остановить. Это явление чересчур точной аппроксимации в процессе обучения и называется переобучением. Если такое случилось, то обычно советуют уменьшить число скрытых элементов и/или слоев, ибо сеть является слишком мощной для данной задачи. Если же сеть, наоборот, была взята недостаточно богатой для того, чтобы моделировать имеющуюся зависимость, то переобучения, скорее всего, не произойдет и обе ошибки — обучения и проверки — не достигнут достаточного уровня малости.

Описанные проблемы с локальными минимумами и выбором размера сети приводят к тому, что при практической работе с нейронными сетями, как правило, приходится экспериментировать с большим числом различных сетей, порой обучая каждую из них несколько раз (чтобы не быть введенным в заблуждение локальными минимумами) и сравнивая полученные результаты. Главным показателем качества результата является здесь контрольная ошибка. В соответствии с общенаучным принципом, согласно которому при прочих равных следует предпочесть более

простую модель, имеет смысл из двух сетей с приблизительно равными ошибками контроля выбрать ту, которая меньше.

Необходимость многократных экспериментов ведет к тому, что контрольное множество начинает играть ключевую роль в выборе модели, то есть становится частью процесса обучения. Тем самым ослабляется его роль как независимого критерия качества модели — при большом числе экспериментов есть риск выбрать "удачную" сеть, дающую хороший результат на контрольном множестве. Для того чтобы придать окончательной модели должную надежность, часто (по крайней мере, когда объем обучающих данных это позволяет) поступают так: резервируют еще одно, тестовое множество наблюдений. Итоговая модель тестируется на данных из этого множества, чтобы убедиться, что результаты, достигнутые на обучающем и контрольном множествах, реальны, а не являются артефактами процесса обучения. Разумеется, для того чтобы соответствовать своей роли, тестовое множество должно быть использовано только один раз: если его использовать повторно для корректировки процесса обучения, то оно фактически превратится в контрольное множество.

Отбор данных

На всех предыдущих этапах мы постоянно опирались на одно предположение, а именно: обучающее, контрольное и тестовое множества должны быть репрезентативными (представительными) с точки зрения существа задачи (более того, эти множества должны быть репрезентативны каждое в отдельности). Известное изречение программистов "garbage in, garbage out" ("мусор на входе — мусор на выходе") нигде не справедливо в такой степени, как при нейросетевом моделировании. Если обучающие данные не репрезентативны, то модель, как минимум, будет не очень хорошей, а в худшем случае — бесполезной. Имеет смысл перечислить ряд причин, которые ухудшают качество обучающего множества.

Будущее непохоже на прошлое. Обычно в качестве обучающих берутся исторические данные. Если обстоятельства изменились, то закономерности, имевшие место в прошлом, могут больше не действовать.

Следует учесть все возможности. Нейронная сеть может обучаться только на тех данных, которыми она располагает. Предположим, что лица с годовым доходом более $100000 имеют высокий кредитный риск, а обучающее множество не содержало лиц с доходом более $40000 в год. Тогда едва ли можно ожидать от сети правильного решения в совершенно новой для нее ситуации.

Сеть обучается тому, чему проще всего обучиться. Классическим (возможно, вымышленным) примером является система машинного зрения, предназначенная для автоматического распознавания танков. Сеть обучалась на ста картинках, содержащих изображения танков, и на ста других картинках, где танков не было. Был достигнут стопроцентно "правильный" результат. Но когда на вход сети были поданы новые данные, она безнадежно провалилась. В чем же была причина? Выяснилось, что фотографии с танками были сделаны в пасмурную, дождливую погоду, а фотографии без танков — в солнечный день. Сеть научилась улавливать (очевидную) разницу в общей освещенности. Чтобы сеть могла результативно работать, ее следовало обучать на данных, где присутствовали бы все погодные условия и типы освещения, при которых сеть будут реально использовать, — и это не говоря еще о рельефе местности, угле и дистанции съемки и т.д.

Несбалансированный набор данных. Коль скоро сеть минимизирует общую погрешность, важное значение приобретают пропорции, в которых представлены данные различных типов. Сеть, обученная на 900 хороших и 100 плохих примерах, будет искажать результат в пользу хороших наблюдений, поскольку это позволит алгоритму уменьшить общую погрешность (которая определяется в основном хорошими случаями). Если в реальной популяции хорошие и плохие объекты представлены в другой пропорции, то результаты, выдаваемые сетью, могут оказаться неверными. Хорошим примером служит задача выявления заболеваний. Пусть, например, при обычных обследованиях в среднем 90% человек оказываются здоровыми. Сеть обучается на имеющихся данных, в которых пропорция здоровые/больные равна 90/10. Затем она применяется для диагностики пациентов с определенным жалобами, среди которых это соотношение уже 50/50. В этом случае сеть будет ставить диагноз чересчур осторожно и не распознает заболевание у некоторых больных. Если же, наоборот, сеть обучить на данных "с жалобами", а затем протестировать на "обычных" данных, то она будет выдавать повышенное число неправильных диагнозов о наличии заболевания. В таких ситуациях обучающие данные нужно скорректировать так, чтобы были учтены различия в распределении (например, можно повторять редкие наблюдения или удалить часто встречающиеся), или же видоизменить решения, выдаваемые сетью, посредством матрицы потерь. Как правило, лучше всего постараться равномерно представить наблюдения различных типов и соответственно этому интерпретировать результаты, которые выдает сеть.

Как обучается многослойный персептрон

Мы сможем лучше понять, как устроен и как обучается многослойный персептрон, если выясним, какие функции он способен моделировать. Вспомним, что уровнем активации элемента называется взвешенная сумма его входов с добавленным к ней пороговым значением. Таким образом, уровень активации представляет собой простую линейную функцию входов. Эта активация затем преобразуется с помощью сигмоидной (имеющей S- образную форму) кривой.

Комбинация линейной функции нескольких переменных и скалярной сигмовидной функции приводит к характерному профилю "сигмовидного склона", который выдает элемент первого промежуточного слоя. На рис. 5.1 соответствующая поверхность изображена в виде функции двух входных переменных. Элемент с большим числом входов выдает многомерный аналог такой поверхности. При изменении весов и порогов меняется и поверхность отклика; может меняться как ориентация всей поверхности, так и крутизна склона — большим значениям весов соответствует более крутой склон. Так, например, если увеличить все веса в два раза, то ориентация не изменится, а наклон будет более крутым. В многослойной сети подобные функции отклика комбинируются друг с другом с помощью последовательного взятия их линейных комбинаций и применения нелинейных функций активации. На рис. 5.2 изображена типичная поверхность отклика для сети с одним промежуточным слоем, состоящим из двух элементов, и одним выходным элементом, для классической задачи "исключающего или". Две разных сигмоидных поверхности объединены в одну поверхность, имеющую форму буквы "U".

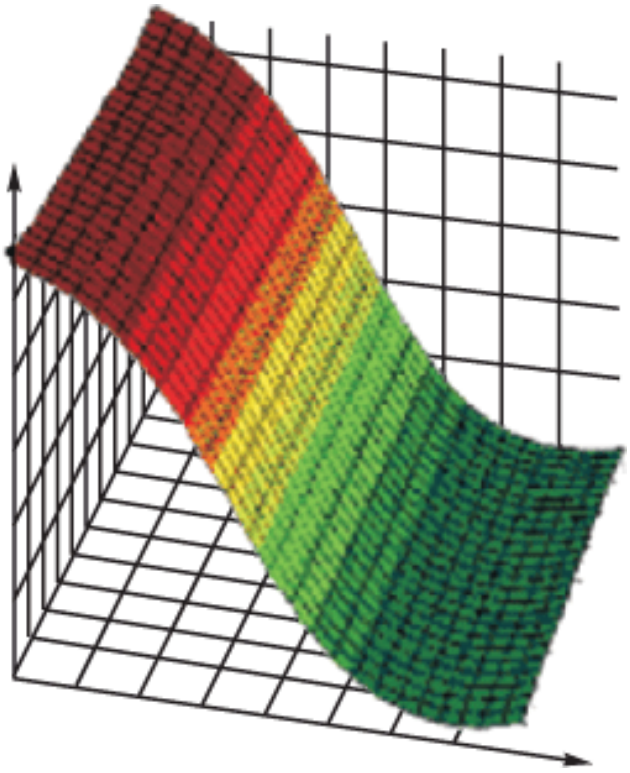


Рис. 5.1.

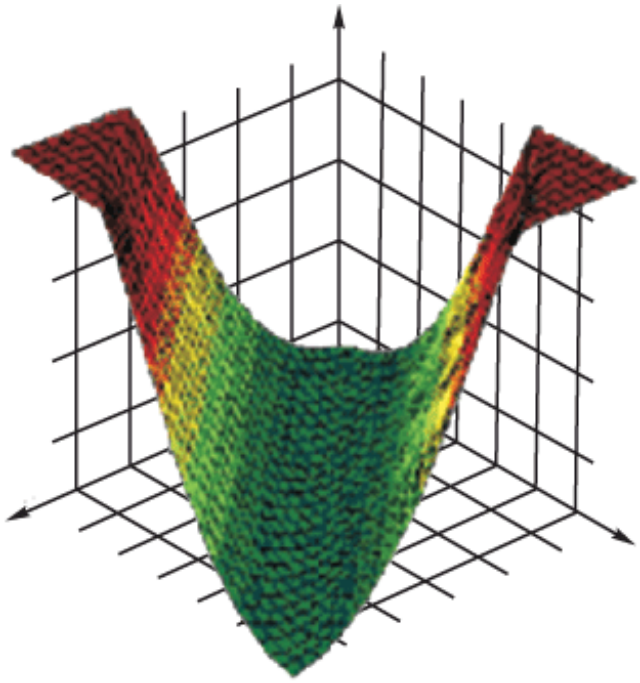


Рис. 5.2.

Перед началом обучения сети весам и порогам случайным образом присваиваются небольшие по величине начальные значения. Тем самым, отклики отдельных элементов сети имеют малый наклон и ориентированы хаотично — фактически они не связаны друг с другом. По мере того, как происходит обучение, поверхности отклика элементов сети вращаются и сдвигаются в нужное положение, а значения весов увеличиваются, поскольку они должны моделировать отдельные участки целевой поверхности отклика.

В задачах классификации выходной элемент должен выдавать сильный сигнал в случае, если данное наблюдение принадлежит к интересующему нас классу, и слабый — в противоположном случае. Иначе говоря, этот элемент должен стремиться смоделировать функцию, равную единице в области пространства объектов, где располагаются объекты из нужного класса, и равную нулю вне этой области. Такая конструкция известна как дискриминантная функция в задачах распознавания. "Идеальная" дискриминантная функция должна иметь плоскую структуру: точки соответствующей поверхности будут располагаться либо на нулевом уровне, либо на высоте "единица".

Если сеть не содержит скрытых элементов, то на выходе она может моделировать только одинарный "сигмовидный склон": точки, находящиеся по одну его сторону, располагаются низко, по другую — высоко. При этом всегда будет существовать область между ними (на склоне), где высота принимает промежуточные значения, но по мере увеличения весов эта область будет сужаться.

Такой сигмовидный склон фактически работает как линейная дискриминантная функция. Точки, лежащие по одну сторону склона, классифицируются как принадлежащие нужному классу, а лежащие по другую сторону — как не принадлежащие. Следовательно, сеть без скрытых слоев может служить классификатором только в линейно-отделимых задачах: когда можно провести линию (или, в случае более высоких размерностей, гиперплоскость), разделяющую точки в пространстве признаков.

Сеть, содержащая один промежуточный слой, строит несколько сигмоидных склонов, — по одному для каждого скрытого элемента, — и затем выходной элемент комбинирует из них "возвышенность". Эта возвышенность получается выпуклой, т.е. не содержащей впадин. При этом в некоторых направлениях она может уходить на бесконечность (как длинный полуостров). Подобная сеть может моделировать большинство реальных задач классификации.

Сеть с двумя промежуточными слоями строит комбинацию из нескольких таких возвышенностей. Их будет столько, сколько элементов во втором слое, и у каждой из них будет столько сторон, сколько элементов было в первом скрытом слое. После несложного размышления делаем вывод, что, используя достаточное число таких возвышенностей, можно воспроизвести поверхность любой формы — в том числе с впадинами и вогнутостями.

Как следствие наших рассмотрений мы получаем, что, теоретически, для моделирования любой задачи достаточно многослойного персептрона с двумя промежуточными слоями (в точной формулировке этот результат известен как теорема Колмогорова). При этом может оказаться, что для решения некоторой конкретной задачи будет более простой и удобной сеть с еще большим числом слоев. Однако для решения большинства практических задач достаточно всего одного промежуточного слоя, два слоя применяются как резерв в особых случаях, а сети с тремя слоями практически не применяются.

В задачах классификации очень важно понять, как следует интерпретировать те точки, которые попали на склон или лежат близко от него. Стандартный подход заключается в том, чтобы для пороговых значений установить некоторые доверительные пределы (принятия или отвержения), которые должны быть достигнуты, чтобы данный элемент считался "принявшим решение". Например, если установлены пороги принятия/отвержения 0,95/0.05, то при уровне выходного сигнала выше 0,95 элемент считается активным, при уровне ниже 0,05 — неактивным, а в промежутке — "неопределенным". Имеется и более тонкий (и, вероятно, более полезный) способ интерпретировать уровни выходного сигнала: считать их вероятностями. В этом случае сеть выдает несколько большую информацию, чем просто "да/нет": она сообщает нам, насколько (в некотором формальном смысле) мы можем доверять ее решению. При этом, однако, вероятностная интерпретация обоснована только в том случае, если выполняются определенные предположения о распределении исходных данных (конкретно, что данные являются выборкой из некоторого распределения, принадлежащего к семейству экспоненциальных распределений). Здесь, как и ранее, может быть принято решение по классификации, но, кроме того, вероятностная интерпретация позволяет ввести концепцию "решения с минимальными затратами".

Предостережения

Несмотря на многочисленные успешные применения обратного распространения, оно не является панацеей. Больше всего неприятностей доставляет неопределенно долгий процесс обучения. В сложных задачах для обучения сети могут потребоваться дни или даже недели, она может и вообще не обучиться. Длительное время обучения может быть результатом неоптимального выбора длины шага. Неудачи в обучении обычно возникают по двум причинам: паралича сети и попадания в локальный минимум.

Паралич сети

В процессе обучения сети значения весов могут в результате коррекции стать очень большими величинами. Это может привести к тому, что все или большинство нейронов будут функционировать при очень больших значениях OUT, в области, где производная сжимающей функции очень мала. Так как посылаемая обратно в процессе обучения ошибка пропорциональна этой производной, то процесс обучения может практически замереть. Теоретически эта проблема изучена плохо. Обычно пытаются уменьшать размера шага , но это увеличивает время обучения. Различные эвристики использовались для предохранения от паралича или для восстановления после него, но пока что они могут рассматриваться лишь как экспериментальные.

Локальные минимумы

Короче говоря, происходит следующее: в данной точке поверхности находится направление скорейшего спуска, затем делается прыжок вниз на расстояние, пропорциональное коэффициенту скорости обучения и крутизне склона, при этом учитывается инерция, то есть стремление сохранить прежнее направление движения. Можно сказать, что метод ведет себя как слепой кенгуру — каждый раз прыгает в направлении, которое кажется ему наилучшим. На самом деле, шаг спуска вычисляется отдельно для всех обучающих наблюдений, взятых в случайном порядке, но в результате получается достаточно хорошая аппроксимация спуска по совокупной поверхности ошибок. Существуют и другие алгоритмы обучения, однако все они используют ту или иную стратегию скорейшего продвижения к точке минимума.

Обратное распространение использует разновидность градиентного спуска, т. е. осуществляет спуск вниз по поверхности ошибки, непрерывно подстраивая веса в направлении к минимуму. Поверхность ошибки сложной сети сильно изрезана и состоит из холмов, долин, складок и оврагов в пространстве высокой размерности. Сеть может попасть в локальный минимум (неглубокую долину), когда рядом имеется гораздо более глубокий минимум. В точке локального минимума все направления ведут вверх и сеть неспособна из него выбраться. Статистические методы обучения могут помочь избежать этой ловушки, но они медленны. П.Д.Вассерман предложил метод, объединяющий статистические методы машины Коши с градиентным спуском обратного распространения и приводящий к системе, которая находит глобальный минимум, сохраняя высокую скорость обратного распространения. Это будет обсуждаться в следующих лекциях.

Размер шага

Внимательный разбор доказательства сходимости показывает, что коррекции весов предполагаются бесконечно малыми. Ясно, что это неосуществимо на практике, так как ведет к бесконечному времени обучения. Размер шага должен браться конечным, и при определении его приходится полагаться только на опыт. Если размер шага очень мал, то сходимость слишком медленная, если же очень велик, то может возникнуть паралич или постоянная неустойчивость. П.Д.Вассерман описал адаптивный алгоритм выбора шага, автоматически корректирующий размер шага в процессе обучения.

Временная неустойчивость

Если сеть учится распознавать буквы, то нет смысла учить "Б", если при этом забывается "А". Процесс обучения должен быть таким, чтобы сеть обучалась на всем обучающем множестве без пропусков того, что уже выучено. В доказательстве сходимости это условие выполнено, но требуется также, чтобы сети предъявлялись все векторы обучающего множества, прежде чем выполняется коррекция весов. Необходимые изменения весов должны вычисляться на всем множестве, что требует дополнительной памяти; после ряда таких обучающих циклов веса сойдутся к минимальной ошибке. Этот метод может оказаться бесполезным, если сеть находится в постоянно меняющейся внешней среде, так что второй раз один и тот же вектор может уже не повториться. В этом случае процесс обучения может никогда не сойтись, бесцельно блуждая или сильно осциллируя. В этом смысле обратное распространение не похоже на биологические системы. Это несоответствие (среди прочих) привело к системе ART, принадлежащей Гроссбергу.

Введение в сети встречного распространения

По своим возможностям сети встречного распространения превосходят возможности однослойных сетей. Время же их обучения, по сравнению с обратным распространением, может уменьшаться в сто раз. Встречное распространение не настолько общее, как обратное распространение, но оно может давать решение в тех приложениях, где долгая обучающая процедура невозможна. Будет показано, что, помимо преодоления ограничений других сетей, встречное распространение обладает собственными интересными и полезными свойствами.

Во встречном распространении объединены два хорошо известных алгоритма: самоорганизующаяся карта Кохонена и звезда Гроссберга. При этом появляются свойства, которых нет ни у одного из них в отдельности.

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

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

Структура сети

На рис. 6.1 показана упрощенная версия прямого действия сети встречного распространения. Здесь иллюстрируются функциональные свойства этой парадигмы. Полная двунаправленная сеть основана на тех же принципах, она обсуждается в этой лекции позднее.

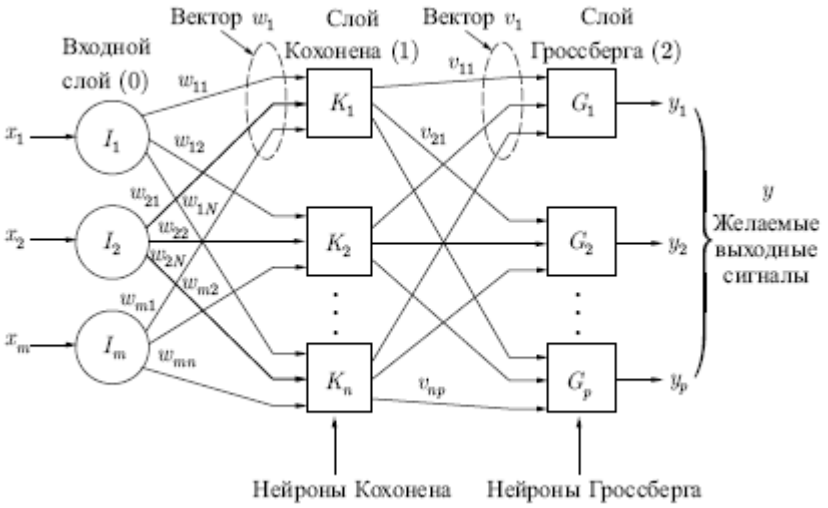


Рис. 6.1.

Нейроны слоя 0 (показанные кружками) служат лишь точками разветвления и не выполняют вычислений. Каждый нейрон слоя 0 соединен с каждым нейроном слоя 1 (называемого слоем Кохонена) отдельным весом W_{mn}. Эти веса в целом рассматриваются как матрица весов W. Аналогично, каждый нейрон в слое Кохонена (слое 1) соединен с каждым нейроном в слое Гроссберга (слое 2) весом . Эти веса образуют матрицу весов . Все это весьма напоминает другие сети, встречавшиеся в предыдущих лекциях; различие, однако, в операциях, выполняемых нейронами Кохонена и Гроссберга.

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

Нормальное функционирование

Слои Кохонена

В своей простейшей форме слой Кохонена функционирует в духе "победитель забирает все", т.е. для данного входного вектора один и только один нейрон Кохонена выдает на выходе логическую единицу, а все остальные выдают ноль. Нейроны Кохонена можно воспринимать как набор электрических лампочек, и для любого входного вектора "загорается" одна из них.

Слой Гроссберга

Слой Гроссберга функционирует в сходной манере. Фактически каждый нейрон слоя Гроссберга лишь выдает величину веса, который связывает этот нейрон с единственным ненулевым нейроном Кохонена.

Обучение слоя Кохонена

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

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

Выбор начальных значений весовых векторов

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

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

Допустим, что имеется несколько множеств входных векторов, все эти множества сходные, но необходимо разделить их на различные классы. Сеть должна быть обучена активировать отдельный нейрон Кохонена для каждого класса. Если начальная плотность весовых векторов в окрестности обучающих векторов слишком мала, то, возможно, не удастся разделить сходные классы из-за того, что весовых векторов в интересующей нас окрестности не хватит, чтобы приписать по одному из них каждому классу входных векторов.

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

Наиболее желательное решение будет таким: распределить весовые векторы в соответствии с плотностью входных векторов, подлежащих разделению, и для этого поместить больше весовых векторов в окрестности большого числа входных векторов. Конечно, на практике это невыполнимо, но существует несколько методов приближенного достижения тех же целей.

Примеры обучения

Рассмотрим примеры обучения сети Кохонена обычным методом и методом выпуклой комбинации. В первом методе будем выбирать равномерно распределенные случайные векторы весов (ядер классов). На рисунке 6.5 представлен пример обучения. Точками обозначены векторы обучающего множества, кружками — векторы весовых коэффициентов.

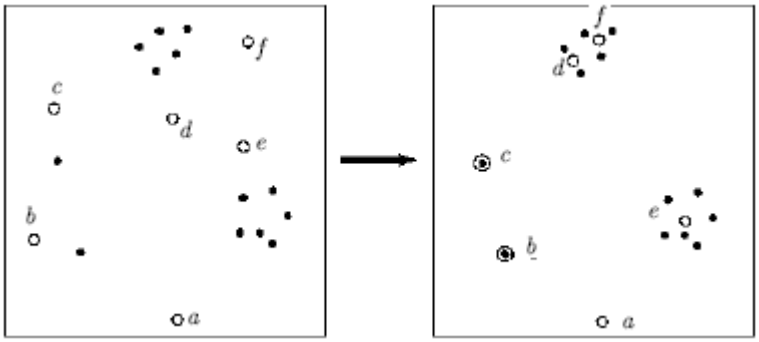


Рис. 6.5.

Вектор весов нейрона не обучается, т.к. ни для одного из векторов обучающего множества этот нейрон не получает максимального выхода. Кроме того, в области из шести обучающих векторов (справа внизу) оказывается всего один вектор весов нейрона , что не соответствует высокой плотности обучающих векторов в этой области. Эти недостатки присущи обычному методу обучения сети Кохонена.

Разберем работу метода выпуклой комбинации. Последовательное изменение картины векторов и весов показано на рис. 6.6.

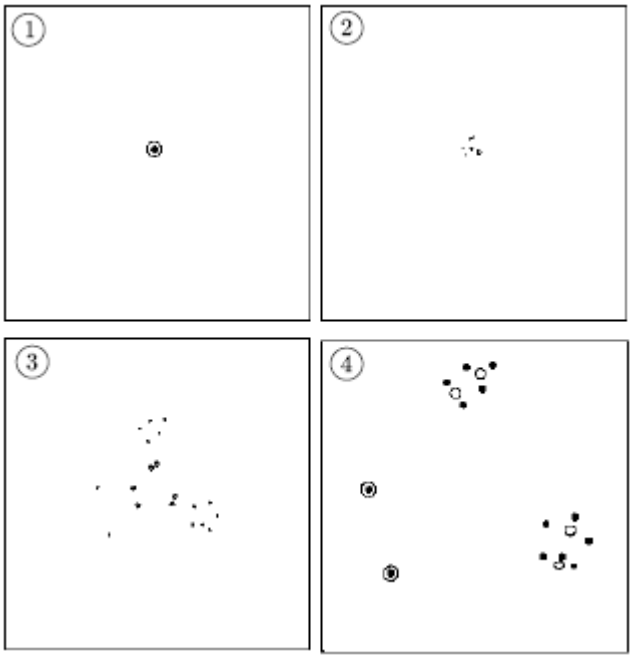


Рис. 6.6.

На первой схеме все векторы весов и обучающего множества имеют одно и то же значение. По мере обучения обучающие векторы расходятся к своим истинным значениям, а векторы весов следуют за ними. В итоге в сети не остается необученных нейронов и плотность векторов весов соответствует плотности векторов обучающего множества. Однако метод выпуклой комбинации хорошо работает, но замедляет процесс обучения, так как весовые векторы подстраиваются к изменяющейся цели. Другой подход состоит в добавлении шума к входным векторам. Тем самым они подвергаются случайным изменениям, схватывая в конце концов весовой вектор. Этот метод также работоспособен, но еще более медленнен, чем метод выпуклой комбинации.

Третий метод начинает работу со случайных весов, но на начальной стадии обучающего процесса подстраивает все веса, а не только связанные с выигравшим нейроном Кохонена. Тем самым весовые векторы перемещаются ближе к области входных векторов. В процессе обучения коррекция весов начинает производиться лишь для ближайших к победителю нейронов Кохонена. Этот радиус коррекции постепенно уменьшается, так что в конце корректируются только веса, связанные с выигравшим нейроном Кохонена.

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

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

Модификации алгоритма обучения

Чувство справедливости: чтобы не допустить отсутствие обучения по любому из нейронов, вводится "чувство справедливости". Если нейрон чаще других выигрывает "состязание", т.е. получает максимальный выход чаще, чем в 1 из случаев, то его значение выхода искусственно уменьшается, чтобы дать возможность выиграть другим нейронам. Это включает все нейроны сети в процесс обучения.

Коррекция весов пропорционально выходу: в этой модификации корректируются веса не только выигравшего нейрона, но и всех остальных, пропорционально их нормированному выходу. Нормировка выполняется по максимальному значению выхода слоя или по его среднему значению. Этот метод также исключает "мертвые" нейроны и улучшает распределение плотности весов.

Режим интерполяции

До сих пор мы обсуждали алгоритм обучения, в котором для каждого входного вектора активировался только один нейрон Кохонена. Это называется методом аккредитации. Его

точность ограничена, так как выход полностью является функцией лишь одного нейрона Кохонена.

В методе интерполяции целая группа нейронов Кохонена, имеющих максимальные выходы, может передавать свои выходные сигналы в слой Гроссберга. Число нейронов в такой группе должно выбираться в зависимости от задачи, и убедительных данных относительно оптимального размера группы не имеется. Как только группа определена, ее множество выходов рассматривается как вектор, длина которого нормализуется на единицу делением каждого значения на корень квадратный из суммы квадратов значений в группе. Все нейроны вне группы имеют нулевые выходы.

Метод интерполяции способен устанавливать более сложные соответствия и может давать более точные результаты. По-прежнему, однако, нет убедительных данных, позволяющих сравнить достоинства и недостатки режимов интерполяции и аккредитации.

Статистические свойства обученной сети

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

Обучение слоя Гроссберга

Слой Гроссберга обучается относительно просто. Входной вектор, являющийся выходом слоя Кохонена, подается на слой нейронов Гроссберга, и выходы слоя Гроссберга вычисляются как при нормальном функционировании. Далее, каждый вес корректируется только в том случае, если он соединен с нейроном Кохонена, имеющим ненулевой выход. Величина коррекции веса пропорциональна разности между весом и требуемым выходом нейрона Гроссберга, с которым этот вес соединен.

Отсюда видно, что веса слоя Гроссберга будут сходиться к средним величинам от желаемых выходов, тогда как веса слоя Кохонена обучаются на средних значениях входов. Обучение слоя Гроссберга — это обучение с учителем, алгоритм располагает желаемым выходом, по которому он обучается. Обучающийся без учителя, самоорганизующийся слой Кохонена дает выходы в недетерминированных позициях. Они отображаются в желаемые выходы слоем Гроссберга.

Сеть встречного распространения полностью

На рис. 6.7 показана сеть встречного распространения целиком. В режиме нормального функционирования предъявляются входные векторы X и Y, и обученная сеть дает на выходе векторы X? и Y?, являющиеся аппроксимациями соответственно для X и Y. Векторы и предполагаются здесь нормализованными единичными векторами, следовательно, порождаемые на выходе векторы также будут иметь тенденцию быть нормализованными.

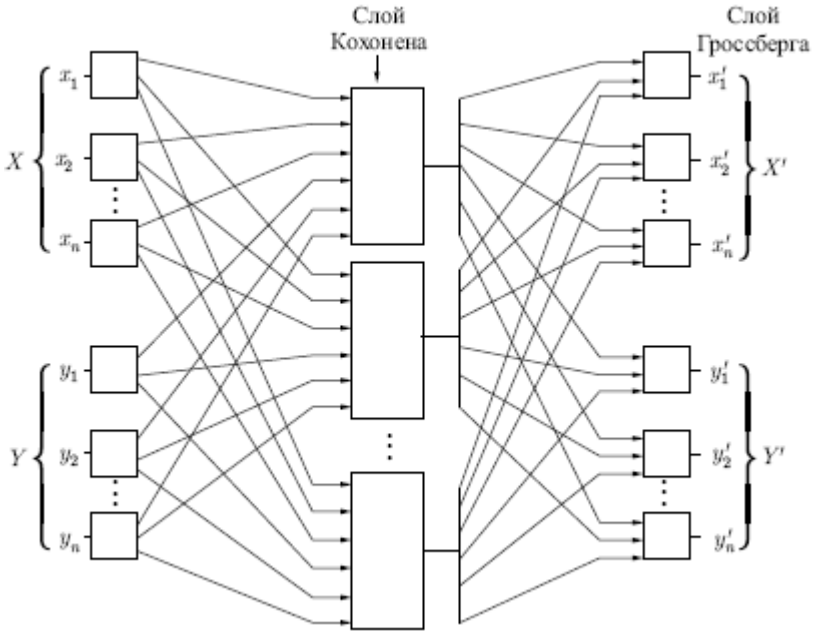


Рис. 6.7.

Рис. 6.7, в отличие от первоначальной конфигурации, не демонстрирует противоток в сети, по которому она получила свое название. Такая форма выбрана потому, что она также иллюстрирует сеть без обратных связей.

Приложение: сжатие данных

В дополнение к обычным функциям отображения векторов, встречное распространение оказывается полезным и в некоторых менее очевидных прикладных областях. Одним из наиболее интересных примеров является сжатие данных.

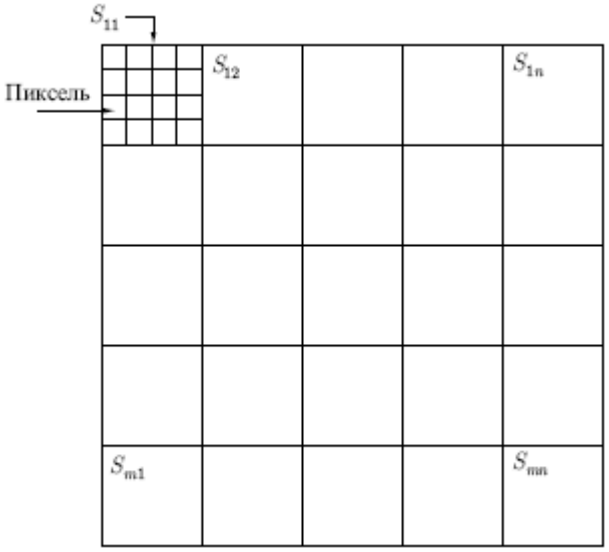


Рис. 6.8.

Сеть встречного распространения может быть использована для сжатия данных перед их передачей, уменьшая тем самым число битов, которые должны быть переданы. Допустим, что требуется передать некоторое изображение. Оно может быть разбито на подизображения , как показано на рис. 6.8. Каждое подизображение разбито на пиксели (мельчайшие элементы изображения). Тогда каждое подизображение является вектором, элементами которого являются пиксели, из которых состоит подизображение. Допустим для простоты, что каждый пиксель - это единица (свет) или нуль (чернота). Если в подизображении имеется пикселей, то для его передачи потребуется n бит. Если допустимы некоторые искажения, то для передачи типичного изображения требуется существенно меньшее число битов, что позволяет передавать изображение быстрее. Это возможно из-за статистического распределения векторов

подизображений. Некоторые из них встречаются часто, тогда как другие встречаются так редко, что могут быть грубо аппроксимированы. Метод, называемый векторным квантованием, находит более короткие последовательности битов, наилучшим образом представляющие эти подизображения.

Сеть встречного распространения может быть использована для выполнения векторного квантования. Множество векторов подизображений используется в качестве входа для обучения слоя Кохонена по методу аккредитации, когда выход единственного нейрона равен 1. Веса слоя Гроссберга обучаются выдавать бинарный код номера того нейрона Кохонена, выход которого равен 1. Например, если выходной сигнал нейрона 7 равен 1 (а все остальные равны 0), то слой Гроссберга будет обучаться выдавать 00...000111 (двоичный код числа 7). Это и будет являться более короткой битовой последовательностью передаваемых символов.

На приемном конце идентичным образом обученная сеть встречного распространения принимает двоичный код и реализует обратную функцию, аппроксимирующую первоначальное подизображение.

Этот метод применялся на практике как к речи, так и к изображениям, с коэффициентом сжатия данных от 10:1 до 100:1. Качество было приемлемым, хотя некоторые искажения данных на приемном конце признаются неизбежными.

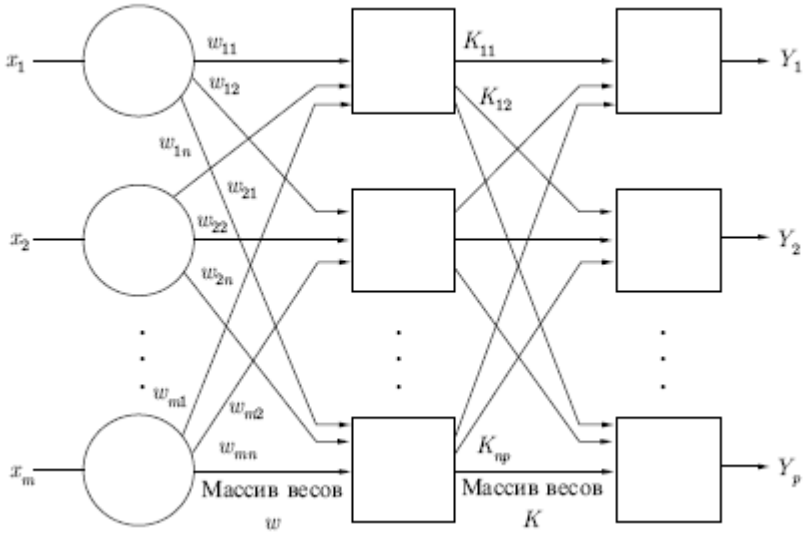
Стохастические методы обучения нейронных сетей

Использование обучения

Искусственная нейронная сеть обучается с помощью некоторого процесса, модифицирующего ее веса. Если обучение успешно, то предъявление сети множества входных сигналов приводит к появлению желаемого множества выходных сигналов. Имеется два класса обучающих методов: детерминистский и стохастический.

Детерминистский метод обучения шаг за шагом осуществляет процедуру коррекции весов сети, основанную на использовании их текущих значений, а также величин входов, фактических выходов и желаемых выходов. Обучение персептрона является примером подобного детерминистского метода.

Стохастические методы обучения выполняют псевдослучайные изменения величин весов, сохраняя те изменения, которые ведут к улучшениям. Чтобы показать это наглядно, рассмотрим рис. 7.1, на котором изображена типичная сеть, где нейроны соединены с помощью весов. Выход нейрона является здесь взвешенной суммой его входов, которая преобразована с помощью нелинейной функции. Для обучения сети могут быть использованы следующие процедуры:
Выбрать вес случайным образом и подкорректировать его на небольшое случайное число.
Предъявить множество входов и вычислить получающиеся выходы.
Сравнить эти выходы с желаемыми выходами и вычислить величину разности между ними. Общепринятый метод состоит в нахождении разности между фактическим и желаемым выходами для каждого элемента обучаемой пары, возведение разностей в квадрат и нахождение суммы этих квадратов. Целью обучения является минимизация этой разности, часто называемой целевой функцией.
Выбрать вес случайным образом и подкорректировать его на небольшое случайное значение. Если коррекция помогает (уменьшает целевую функцию), то сохранить ее, в противном случае вернуться к первоначальному значению веса.
Повторять шаги с 1 по 3 до тех пор, пока сеть не будет обучена в достаточной степени.



Этот процесс стремится минимизировать целевую функцию, но может попасть, как в ловушку, в неудачное решение. На рис. 7.2 показано, как это может происходить в системе с единственным весом. Допустим, что первоначально вес взят равным значению в точке А. Если случайные шаги по весу малы, то любые отклонения от точки А увеличивают целевую функцию и будут отвергнуты. Лучшее значение веса, принимаемое в точке В, никогда не будет найдено, и система будет поймана в ловушку локальным минимумом вместо глобального минимума в точке В. Если же случайные коррекции веса очень велики, то как точка А, так и точка В будут часто посещаться, но то же самое будет верно и для каждой другой точки. Вес будет меняться так резко, что он никогда не установится в желаемом минимуме.

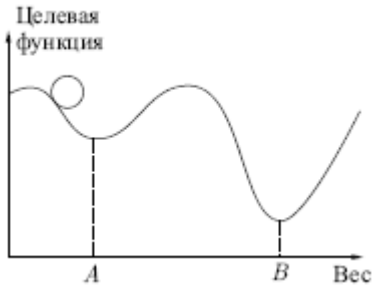


Рис. 7.2.

Полезная стратегия для избежания подобных проблем состоит в больших начальных шагах и постепенном уменьшении размера среднего случайного шага. Это позволяет сети вырываться из локальных минимумов и в то же время гарантирует окончательную стабилизацию сети.

Ловушки локальных минимумов досаждают всем алгоритмам обучения, основанным на поиске минимума (включая персептрон и сети обратного распространения), и представляют серьезную и широко распространенную трудность, которую почему-то часто игнорируют. Стохастические методы позволяют решить эту проблему. Стратегия коррекции весов, вынуждающая веса принимать значение глобального оптимума в точке В, вполне возможна.

Нейронные сети Хопфилда и Хэмминга

Сети, рассмотренные ранее, не имели обратных связей, т. е. связей, идущих от выходов сетей к их входам. Отсутствие обратной связи гарантирует безусловную устойчивость сетей. (Они не могут войти в режим, когда выход беспрерывно блуждает от состояния к состоянию и не пригоден для использования.) Но это весьма желательное качество достигается не бесплатно: сети без обратных связей обладают более ограниченными возможностями по сравнению с сетями с обратными связями. Так как сети с обратными связями имеют пути, передающие сигналы от выходов к входам, то отклик таких сетей является динамическим, т. е. после приложения нового входа вычисляется выход и, передаваясь по сети обратной связи, модифицирует вход. Затем выход повторно вычисляется, и процесс повторяется снова и снова. Для устойчивой сети последовательные итерации приводят к все меньшим изменениям выхода, пока в конце концов выход не становится постоянным. Для многих сетей процесс никогда не заканчивается, такие сети называют неустойчивыми. Неустойчивые сети обладают интересными свойствами и изучались в качестве примера хаотических систем. Однако такой большой предмет, как хаос,

находится за пределами этого курса. Вместо этого мы сконцентрируем свое внимание на устойчивых сетях, т. е. на тех, которые в завершении процесса дают постоянный выход. Проблема устойчивости ставила в тупик первых исследователей. Никто не мог предсказать, какие из сетей будут устойчивыми, а какие будут находиться в постоянном изменении. Более того, проблема представлялась столь трудной, что многие исследователи были настроены пессимистически относительно возможности ее решения. К счастью, была получена теорема, описавшая подмножество сетей с обратными связями, выходы которых в конце концов достигают устойчивого состояния. Это замечательное достижение открыло дорогу дальнейшим исследованиям, и сегодня многие ученые занимаются исследованием сложного поведения и возможностей этих систем. Дж. Хопфилд сделал важный вклад как в теорию, так и в применение систем с обратными связями. Поэтому некоторые из конфигураций известны как сети Хопфилда. Конфигурации сетей с обратными связями

Рассмотренный нами ранее персептрон относится к классу сетей с направленным потоком распространения информации и не содержит обратных связей. На этапе функционирования каждый нейрон выполняет свою функцию — передачу возбуждения другим нейронам — ровно один раз. Динамика состояний нейронов является неитерационной.

Несколько более сложной является динамика в сети Кохонена. Конкурентное соревнование нейронов достигается путем итераций, в процессе которых информация многократно передается между нейронами.

В общем случае может быть рассмотрена нейронная сеть, содержащая произвольные обратные связи, по которым переданное возбуждение возвращается к данному нейрону, и он повторно выполняет свою функцию. Наблюдения за биологическими локальными нейросетями указывают на наличие множественных обратных связей. Нейродинамика в таких системах становится итерационной. Это свойство существенно расширяет множество типов нейросетевых архитектур, но одновременно приводит к появлению новых проблем.

Неитерационная динамика состояний нейронов является, очевидно, всегда устойчивой. Обратные связи могут приводить к возникновению неустойчивостей, подобных тем, которые возникают в усилительных радиотехнических системах при положительной обратной связи. В нейронных сетях неустойчивость проявляется в блуждающей смене состояний нейронов, не приводящей к возникновению стационарных состояний. В общем случае, ответ на вопрос об устойчивости динамики произвольной системы с обратными связями крайне сложен и до настоящего времени является открытым.

Остановимся на важном частном случае нейросетевой архитектуры, для которой свойства устойчивости подробно исследованы. На рис. 8.1 показана сеть с обратными связями, состоящая из двух слоев. Способ представления несколько отличается от использованного в работе Хопфилда и других сходных, но эквивалентен им с функциональной точки зрения, а также хорошо связан с сетями, рассмотренными на предыдущих лекциях. Нулевой слой, как и на предыдущих рисунках, не выполняет вычислительной функции, а лишь распределяет выходы сети обратно на входы. Каждый нейрон первого слоя вычисляет взвешенную сумму своих входов, давая сигнал NET, который затем с помощью нелинейной функции F преобразуется в сигнал OUT. Эти операции сходны с нейронами других сетей.

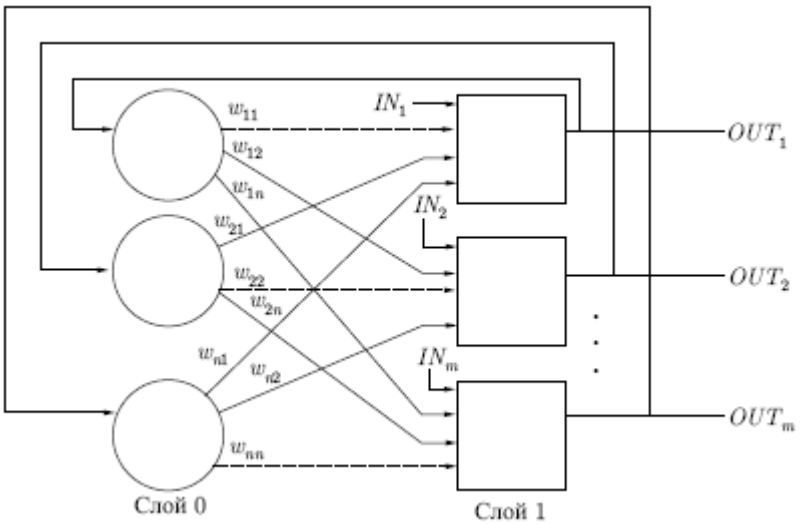


Рис. 8.1.

Как говорилось выше, иногда сеть не может провести распознавание и выдает на выходе несуществующий образ. Это связано с проблемой ограниченности возможностей сети. Для сети Хопфилда число запоминаемых образов m не должно превышать величины, примерно равной 0,15n. Кроме того, если два образа А и Б имеют значительное сходство, они, возможно, будут вызывать у сети перекрестные ассоциации, то есть предъявление на входы сети вектора А приведет к появлению на ее выходах вектора Б и наоборот.

Когда нет необходимости, чтобы сеть выдавала образец в явном виде и достаточно, скажем, получать номер образца, ассоциативную память успешно реализует сеть Хэмминга. Данная сеть характеризуется, по сравнению с сетью Хопфилда, более экономным использованием памяти и меньшим объемом вычислений, что становится очевидным из ее структуры (см. рис. 8.2).

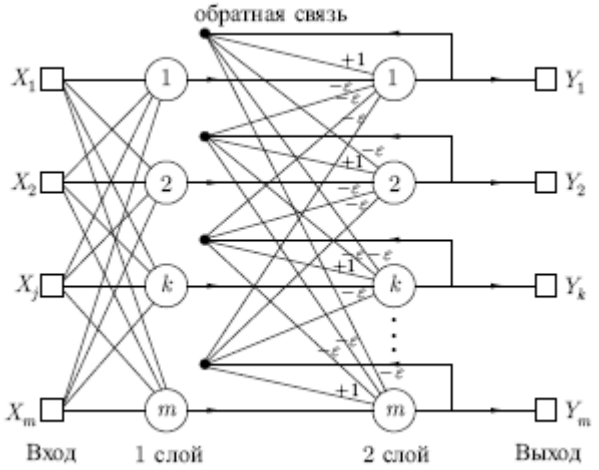


Рис. 8.2.

Сеть состоит из двух слоев. Первый и второй слои имеют по m нейронов, где m — число образцов. Нейроны первого слоя имеют по n синапсов, соединенных с входами сети (которые образуют фиктивный нулевой слой). Нейроны второго слоя связаны между собой ингибиторными (отрицательными обратными) синаптическими связями. Единственный синапс с положительной обратной связью для каждого нейрона соединен с его же аксоном.

Идея работы сети состоит в нахождении расстояния Хэмминга от тестируемого образа до всех образцов. Расстоянием Хэмминга называется число отличающихся битов в двух бинарных векторах. Сеть должна выбрать образец с минимальным расстоянием Хэмминга до неизвестного входного сигнала, в результате чего будет активизирован только один выход сети, соответствующий именно этому образцу.

Двунаправленная ассоциативная память

Память человека часто является ассоциативной; один предмет напоминает нам о другом, а другой — о третьем. Если выпустить наши мысли из-под контроля, они будут перемещаться от предмета к предмету по цепочке умственных ассоциаций. Кроме того, возможно использование ассоциативного мышления для восстановления забытых образов. Если мы забыли, где оставили свои очки, то пытаемся вспомнить, где видели их в последний раз, с кем в это время разговаривали и что делали. Так устанавливается конец цепочки ассоциаций, и это позволяет нашей памяти соединять ассоциации для получения требуемого образа.

Ассоциативная память, рассмотренная в предыдущих лекциях, является, строго говоря, автоассоциативной: это означает, что образ может быть завершен или исправлен, но не может быть ассоциирован с другим образом. Данный факт является результатом одноуровневой структуры ассоциативной памяти, в ней вектор появляется на выходе тех же нейронов, на которые поступает входной вектор.

Двунаправленная ассоциативная память (ДАП) является гетероассоциативной; входной вектор поступает на один набор нейронов, а соответствующий выходной вектор появляется на другом наборе нейронов. Как и сеть Хопфилда, ДАП способна к обобщению, вырабатывая правильные реакции, несмотря на искаженные входы. Кроме того, могут быть реализованы адаптивные версии ДАП, выделяющие эталонный образ из зашумленных экземпляров. Эти возможности сильно напоминают процесс мышления человека и позволяют искусственным нейронным сетям приблизиться к моделированию естественного мозга.