

# Глава 1

## Нейронные сети

Данная глава носит обзорный характер и имеет своей целью лишь заинтересовать читателя.

В настоящее время «...искусственные *нейронные сети* применяются для решения очень многих задач обработки изображений, управления роботами и непрерывными производствами, для понимания и синтеза речи, для диагностики заболеваний людей и технических неполадок в машинах и приборах, для предсказания курсов валют и результатов скачек...» [1].

Искусственные нейронные сети – это еще одна модель дискретной динамической системы. Качественной особенностью этой модели является способность к «обучению», иными словами, нейросеть может «приспосабливаться» к окружающей среде<sup>1</sup>.

---

<sup>1</sup>Нейронные сети позволяют, к примеру, с любой точностью вычислять произвольную *непрерывную* функцию. Вообще говоря, ответ на вопрос «что могут нейронные сети?» весьма оптимистичен – «нейронные сети могут все!». Остается открытым другой вопрос – «как их этому научить?» [1].

## 1.1. Модель нейрона Хебба

В основе искусственных нейронных сетей лежит идея о том, что нейроны можно моделировать довольно простыми автоматами, а вся сложность функционирования сети определяется связями между нейронами: «структура связей - все, свойства элементов - ничто» [1].

Канадский физиолог Д. Хебб в конце 40-х годов XX века в процессе исследования нервных клеток заметил, что связь между двумя клетками усиливается, если обе клетки пробуждаются (становятся активными) в один и тот же момент времени. Если  $j$ -я клетка с выходным сигналом  $x_j$  связана с  $i$ -й клеткой, имеющей выходной сигнал  $x_i$ , связью с весом  $\alpha_{ij}$ , то на силу связи этих клеток влияют оба значения выходных сигналов  $x_i$  и  $x_j$ . Д. Хебб предложил формальное правило, в котором отразились результаты его наблюдений. В соответствии с **правилом Хебба**, вес  $\alpha_{ij}$  синапса (связи), ведущего от нейрона  $j$  к нейрону  $i$  изменяется пропорционально произведению входного и выходного сигналов нейрона  $i$ :

$$\Delta\alpha_{ij} = \eta x_i x_j. \quad (9.1)$$

Заметим, что для тормозящих синапсов можно ввести **анти-хеббовское правило**:

$$\Delta\alpha_{ij} = -\eta x_i x_j. \quad (9.2)$$

Здесь  $\eta$  – это **коэффициент обучения**, значение которого выбирается в интервале  $(0, 1)$ . Правило Хебба может применяться для *нейронных сетей* различных типов с разнообразными функциями активации отдельных нейронов.

Структурная схема **нейрона Хебба** представлена на рисунке 1.1. Связь с весом, способ подбора значения которого задается правилом Хебба, соединяет входной сигнал  $x_j$  с **сумматором**  $i$ -го нейрона, вырабатывающего выходной сигнал  $x_i$ .

**Неоднородный адаптивный сумматор** (см. рис. 1.1) вычисляет скалярное произведение вектора весов на вектор

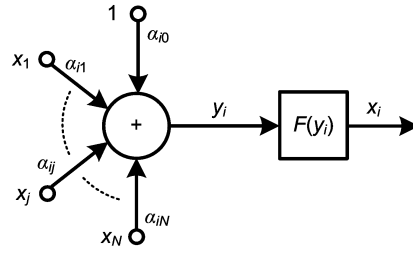


Рис. 1.1. Структурная схема нейрона Хебба

входных сигналов согласно формуле:

$$y_i = \alpha_{i0} + \sum_{j=1}^N \alpha_{ij} x_j. \quad (9.3)$$

Слагаемое  $\alpha_{i0}$  может отсутствовать, в этом случае сумматор будет однородным. Функция  $F$  (см. рис. 1.1) называется **нелинейным преобразователем** сигнала.

**Обучение нейрона** заключается в преобразовании весов согласно правилу Хебба. Это правило характеризуется тем, что в результате его применения веса могут принимать произвольно большие значения, поскольку в каждом цикле обучения происходит суммирование текущего значения веса и его приращения.

Один из способов стабилизации процесса обучения по правилу Хебба состоит в учете для уточнения веса последнего значения  $\alpha_{ij}$ , уменьшенного на коэффициент забывания  $\gamma$ . При этом правило Хебба представляется в виде:

$$\alpha_{ij}(t+1) = \alpha_{ij}(t)(1 - \gamma) + \Delta\alpha_{ij}. \quad (9.4)$$

Значение коэффициента забывания  $\gamma$  выбирается, как правило, из интервала  $(0, 1)$ , и чаще всего составляет некоторый процент от коэффициента обучения  $\eta$ . Применение больших значений  $\gamma$  приводит к тому, что нейрон забывает значительную часть того, чему он обучился в прошлом. Рекомендуемые

значения коэффициента забывания  $\gamma < 0, 1$ . При таком выборе  $\gamma$  нейрон сохраняет большую часть информации, накопленной в процессе обучения, и получает возможность стабилизировать значения весов на определенном уровне.

## 1.2. Архитектура нейронных сетей

Будем рассматривать только *синхронно функционирующие* нейронные сети, в которых в дискретные моменты времени все нейроны срабатывают одновременно.

В теории нейросетей выделяют две базовых *архитектуры* - слоистые и полносвязные сети.

В *слоистых* нейронных сетях нейроны расположены в несколько слоев. Нейроны первого слоя получают входные сигналы, преобразуют их и через точки ветвления передают нейронам второго слоя. Далее срабатывает второй слой и т.д. до  $k$ -го слоя, который выдает выходные сигналы. Если не оговорено противное, то каждый выходной сигнал  $i$ -го слоя подается на вход всех нейронов  $(i + 1)$ -го. Число нейронов в каждом слое может быть любым и никак заранее не связано с количеством нейронов в других слоях. Все входные сигналы подаются всем нейронам первого слоя. Особое распространение получили трехслойные сети, в которых каждый слой имеет свое наименование: первый – входной, второй – скрытый, третий – выходной.

В *полносвязных* нейронных сетях каждый нейрон передает свой выходной сигнал остальным нейронам, включая самого себя. Выходными сигналами сети могут быть все или некоторые выходные сигналы нейронов после нескольких тактов функционирования сети. Все входные сигналы подаются всем нейронам.

Независимо от архитектуры, *процесс обучения сети* строится так: «...существует задачник – набор примеров с заданными ответами. Эти примеры предъявляются системе. Нейроны получают по входным связям сигналы – условия примера преобразуют их, несколько раз обмениваются преобра-

зованными сигналами и, наконец, выдают ответ – также набор сигналов. Отклонение от правильного ответа штрафует-ся. Обучение состоит в минимизации штрафа как (неявной) функции связей...» [1].

### 1.3. Решение задач при помощи нейронных сетей

В этом параграфе рассмотрим простейшие приложения полносвязных нейронных сетей. Представим нейронную сеть как *полносвязный граф* с весами. Как и прежде, вес связи, ведущей от  $j$ -го нейрона к  $i$ -му, равен  $\alpha_{ij}$  (еще раз обратите внимание на порядок индексов). Для полносвязных сетей определены значения  $\alpha_{ij}$  при всех  $i$  и  $j$ . Отметим, что для других архитектур связи, ведущие от  $j$ -го нейрона к  $i$ -му, для некоторых  $i$  и  $j$  не определены. В этом случае можно положить  $\alpha_{ij} = 0$ . В общем случае «сила связи» от  $i$  к  $j$  необязательно равна значению «силы связи» от  $j$  к  $i$ .

Пусть на выходах всех нейронов получены сигналы  $x_j$  ( $j$  – номер нейрона). Обозначим через  $x$  вектор этих выходных сигналов. Теперь, если пропустить вектора сигналов  $x$  через сеть связей, то результирующий вектор  $y$  входных сигналов нелинейных элементов нейронов можно получить как произведение матрицы  $[\alpha_{ij}]$  на вектор сигналов  $x$ :  $y_i = \sum_j \alpha_{ij} x_j$ .

«...Соответствие: «*прохождение сети*  $\equiv$  *умножение матрицы связей на вектор сигналов*», является основой для перевода обычных численных методов на нейросетевой язык и обратно. Практически всюду, где основной операцией является умножение матрицы на вектор, применимы нейронные сети. С другой стороны, любое устройство, позволяющее быстро осуществлять такое умножение, может использоваться для реализации нейронных сетей...» [1].

В частности, пусть нам требуется *вычислить градиент квадратичной формы*  $H = \frac{1}{2}(x, Qx)$ . Для этого можно использовать полносвязную нейросеть с *симметричной матри-*

цей связей  $Q$  ( $q_{ij} = q_{ji}$ ), так как  $\text{grad}(H) = Qx$ .

Умение вычислять градиент квадратичной формы дает нам возможность решать многие прикладные задачи. Приведем некоторые из них.

Найдем *точку минимума многочлена второго порядка*, используя *метод наискорейшего спуска*. Пусть задан такой многочлен:  $P(x) = \frac{1}{2}(x, Qx) + (b, x)$ . Его градиент равен  $\text{grad}(P(x)) = Qx + b$ .

Исходя из вышесказанного, вектор  $\text{grad}(P(x))$  может быть получен при прохождении вектора  $x$  через нейросеть с весами связей  $\alpha_{ij} = q_{ij} = q_{ji}$  при условии, что на входной сумматор каждого нейрона по дополнительной связи веса  $b$  подается стандартный единичный сигнал [1].

В соответствии с методом наискорейшего спуска, функционирование такой нейросети зададим формулой

$$x' = x - \tau \text{grad}(P) = x - \tau(Qx + b). \quad (9.5)$$

Входные веса каждого  $j$ -го нейрона определяются по следующим правилам:  $\alpha_{ij} = -hq_{ij}$  ( $i \neq j$ ) для связей с другими нейронами, вес  $-b_j$  для постоянного единичного входного сигнала, и вес  $\alpha_{jj} = 1 - hq_{jj}$  для связи нейрона с самим собой (то есть, для передачи на нейрон его же сигнала с предыдущего шага).

Выбор шага  $\tau$  ( $\tau > 0$ ) строго не оговаривается (он зависит от коэффициентов минимизируемого многочлена). Тем не менее, достаточно, чтобы шаг стремился со временем к нулю, а сумма шагов – к бесконечности (например,  $\tau_t = \frac{1}{t}$  или  $\tau_t = \frac{1}{\sqrt{t}}$ , где  $t$  – шаг по времени).

«...Итак, простая симметричная полносвязная сеть без нелинейных элементов может методом наискорейшего спуска искать точку минимума квадратичного многочлена...» [1]. Заметим, что в данном случае обучение сети не происходит – веса нейронов не изменяются.

Небольшая модификация сети позволяет вместо безусловного минимума многочлена второго порядка  $P$  искать *точку*

**условного минимума** с условиями  $x_i = c_i$ ,  $i = i_1, \dots, i_k$ . Для этого формулу (9.5) необходимо дополнить следующим образом:

$$x'_i = \begin{cases} c_i, & \text{при } i = i_1, \dots, i_k, \\ x_i - \tau \frac{\partial P}{\partial x_i} = x_i - \tau (\sum_j q_{ij} x_j + b_i), & \text{иначе.} \end{cases} \quad (9.6)$$

Устройство, способное находить точку условного минимума многочлена второго порядка, может быть преобразовано для решения важной задачи – **проблемы заполнения пробелов в данных** (и, в частности, для построения линейной регрессии). Но в этом случае уже потребуется обучение сети – пошаговое уточнение матрицы весов. Подробное описание такой нейросети можно найти в [1].

Обратим внимание на тот факт, что «...во всех задачах оптимизации существенную роль играет вопрос о *правилах остановки*: когда следует прекратить циклическое функционирование сети, остановиться и считать полученный результат ответом? Простейший выбор – остановка по малости изменений: если изменения сигналов сети за цикл меньше некоторого фиксированного малого  $\delta$  (при использовании переменного шага  $\delta$  может быть его функцией), то оптимизация заканчивается...» [1].

Мы рассмотрели задачи, которые сводятся к минимизации положительно определенных квадратичных форм. Однако, самое знаменитое приложение полносвязных нейросетей – **системы ассоциативной памяти**.

Пусть задано несколько *эталонных* векторов данных  $x^1, \dots, x^m$ . При обработке поступившего на вход системы вектора  $x$  требуется получить на выходе ближайший к нему эталонный вектор. Мерой сходства будем считать косинус угла между векторами – для векторов фиксированной длины это просто скалярное произведение.

Несложно показать, что изменение вектора  $x$  по закону

$$x' = x + \tau \sum_k x^k (x^k, x), \quad (9.7)$$

где  $\tau$  – малый шаг, приведет к увеличению проекции  $x$  на те эталоны, скалярное произведение на которые  $(x^k, x)$  больше.

Для простоты будем рассматривать эталоны и ожидаемые результаты с координатами  $\pm 1$ . Так как сумма в формуле (9.7) задает направление антиградиента, то очевидным образом приходим к дифференциальному уравнению

$$\frac{dx}{dt} = -\text{grad}(H), \quad H = H_0 + \gamma H_1, \quad (9.8)$$

$$H_0(x) = -\frac{1}{2} \sum_k (x^k, x)^2, \quad H_1(x) = \frac{1}{2} \sum_i (x_i^2 - 1)^2. \quad (9.9)$$

$$\text{grad}(H_0) = -\sum_k x^k (x^k, x), \quad (\text{grad}(H_1))_j = (x_j^2 - 1)x_j. \quad (9.10)$$

Здесь верхними индексами обозначаются номера векторов-эталонов, нижними – координаты векторов.

«...Функция  $H$  называется «энергией» сети, она минимизируется в ходе функционирования. Слагаемое  $H_0$  вводится для того, чтобы со временем возрастала проекция вектора  $x$  на те эталоны, которые к нему ближе, слагаемое  $H_1$  обеспечивает стремление координат вектора  $x$  к  $\pm 1$ . Параметр  $\gamma$  определяет соотношение между интенсивностями этих двух процессов. Целесообразно постепенно менять  $\gamma$  со временем, начиная с малых  $\gamma < 1$ , и приходя в конце концов к  $\gamma > 1$ ...» [1].

Обсудим получающиеся веса связей. Матрица связей построенной нейросети определяется градиентом функции  $H_0$ , так как  $(\text{grad}(H_1))_j$  вычисляется «без участия сети». Вес связи между  $i$ -м и  $j$ -м нейронами не зависит от направления связи и равен

$$\alpha_{ij} = \sum_k x_i^k x_j^k. \quad (9.11)$$

Заметим, что вклад  $k$ -го эталона в связь между  $i$ -м и  $j$ -м нейронами  $(x_i^k x_j^k)$  равен  $+1$ , если  $i$ -я и  $j$ -я координаты этого эталона имеют одинаковый знак, и равен  $-1$ , если знаки координат различны. В результате, возбуждение  $i$ -го нейрона передается  $j$ -му (и симметрично, от  $j$ -го к  $i$ -му), если у большинства



эталонных знаков  $i$ -й и  $j$ -й координат совпадают. В противном случае эти нейроны тормозят друг друга: возбуждение  $i$ -го ведет к торможению  $j$ -го, торможение  $i$ -го – к возбуждению  $j$ -го (воздействие  $j$ -го на  $i$ -й симметрично). Формула (9.11) задает **правило образования ассоциативных связей**, которое является обобщением правила Хебба.

Мы рассмотрели несколько приложений полносвязных сетей. Вместе с тем слоистые сети также используются при решении многих задач. В частности – это **проблема сжатия информации**. Для иллюстрации можно использовать линейную сеть с одним скрытым слоем, изображенную на рисунке 1.2. Количество нейронов выходного слоя равно числу сиг-

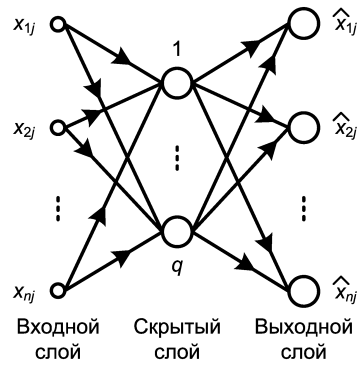


Рис. 1.2. Структура нейронной сети для сжатия данных

налов входного слоя. Скрытый слой содержит  $q$  нейронов, причем  $q \ll n$ . Входной и скрытый слои выполняют собственно компрессию данных, тогда как скрытый и выходной слои осуществляют декомпрессию. Подробное описание предложенной нейросети можно найти в [?].

# Литература

- [1] Горбань А.Н., Дунин-Барковский В.Л., Кирдин А.Н. и др. *Нейроинформатика*. Новосибирск: Наука. Сибирское предприятие РАН, 1998. 296 с. (Электронная версия.)