

# СЕТЬ ОБРАТНОГО РАСПРОСТРАНЕНИЯ

Рассмотренный в предыдущем параграфе алгоритм обучения однослойного персептрона очень прост. Однако долгие годы не удавалось обобщить этот алгоритм на случай многослойных сетей, что спровоцировало в научных кругах значительный спад интереса к нейронным сетям.

Только в 1986 году Румельхарт разработал эффективный алгоритм корректировки весов, названный алгоритмом обратного распространения ошибок (back propagation).

Нейронные сети обратного распространения – это современный инструмент поиска закономерностей, прогнозирования, качественного анализа. Такое название – сети обратного распространения – они получили из-за используемого алгоритма обучения, в котором ошибка распространяется от выходного слоя к входному, т. е. в направлении, противоположном направлению распространения сигнала при нормальном функционировании сети.

Нейронная сеть обратного распространения состоит из нескольких слоев нейронов, причем каждый нейрон предыдущего слоя связан с каждым нейроном последующего слоя. В большинстве практических приложений оказывается достаточно рассмотрения двухслойной нейронной сети, имеющей входной (скрытый) слой нейронов и выходной слой (рис 1).

Матрицу весовых коэффициентов от входов к скрытому слою обозначим  $W$ , а матрицу весов, соединяющих скрытый и выходной слой, – как  $V$ .

Для индексов примем следующие обозначения: входы будем нумеровать только индексом  $i$ , элементы скрытого слоя – индексом  $j$ , а выходы, соответственно, индексом  $k$ . Число входов сети равно  $n$ , число нейронов в скрытом слое –  $m$ , число нейронов в выходном слое –  $p$ . Пусть сеть обучается на выборке  $(X^t, D^t)$ ,  $t = 1, 2, \dots, T$

Обучение с учителем ставит перед сетью задачу обобщить  $p$  примеров, заданных парами векторов  $(X^t, D^t)$ ,  $t = \overline{1, T}$ . Вектор  $x^t = (x_1^t, x_2^t, \dots, x_i^t, \dots, x_n^t)$  в случае задачи классификации задает входной образ (вектор признаков), а вектор  $d^t = (d_1^t, d_2^t, \dots, d_p^t)$ , задающий эталонный выход, должен кодировать номер класса. При этом есть множество вариантов кодирования. Оптимальным представляется кодирование, когда номер класса определяется позицией единичной компоненты в векторе  $d^t$ , а все остальные компоненты равны 0. Каждый выходной нейрон соответствует одному классу. Такой способ позволяет при классификации определять вероятность каждого класса по величине на выходе соответствующего нейрона (чем ближе к единице, тем вероятность больше).

При обучении нейронной сети ставится задача минимизации целевой функции ошибки, которая находится по методу наименьших квадратов:

$$E = \frac{1}{2} \sum_{k=1}^p (y_k - d_k)^2$$

где  $y_k$  – полученное реальное значение  $k$ -го выхода нейросети при подаче на нее одного из входных образов обучающей выборки;  $d_k$  – требуемое (целевое) значение  $k$ -го выхода для этого образа.

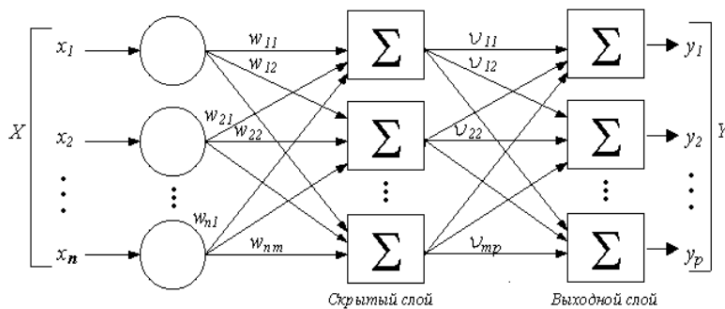


Рис. 1. Нейронная сеть обратного распространения

Обучение нейросети производится известным оптимизационным методом градиентного спуска, т. е. на каждой итерации изменение веса производится по формулам:

$$w_{ij}^{N+1} = w_{ij}^N - \alpha \frac{\partial E}{\partial w_{ij}}, \quad (2)$$

$$v_{jk}^{N+1} = v_{jk}^N - \alpha \frac{\partial E}{\partial v_{jk}}, \quad (3)$$

где  $\alpha$  – параметр, определяющий скорость обучения.

В качестве активационной функции в сети обратного распространения обычно используется логистическая функция  $f(s) = \frac{1}{(1 + e^{-s})}$ , где  $s$  – взвешенная сумма входов нейрона. Эта функция удобна для вычислений в градиентном методе, так как имеет простую производную:

$$f'(S) = \left( \frac{1}{(1 + e^{-S})} \right)' = (-1)(-1) \frac{e^{-S}}{(1 + e^{-S})^2} = f(S)(1 - f(S)) \quad (4)$$

Функция ошибки в явном виде не содержит зависимости от весовых коэффициентов  $v_{jk}$  и  $w_{ij}$ , поэтому для вычисления производных  $\frac{\partial E}{\partial w_{ij}}$ ,  $\frac{\partial E}{\partial v_{jk}}$  воспользуемся формулами дифференцирования сложной функции:

$$\frac{\partial E}{\partial v_{jk}} = \frac{\partial E}{\partial y_k} \frac{\partial y_k}{\partial s_k} \frac{\partial s_k}{\partial v_{jk}},$$

где  $s_k$  – взвешенная сумма входных сигналов  $k$ -го нейрона выходного слоя.

Обозначим  $y_j^c$  – значение выхода  $j$ -го нейрона скрытого слоя. Тогда

$$s_k = \sum_{j=1}^m v_{jk} y_j^c \quad \text{и} \quad \frac{\partial s_k}{\partial v_{jk}} = y_j^c.$$

Так как  $y_k = f(s_k)$  то  $\frac{\partial y_k}{\partial s_k} = f(s_k)(1 - f(s_k)) = y_k(1 - y_k)$

Наконец,  $\frac{\partial E}{\partial y_k} = y_k - d_k$ , таким образом получили выражение для производной

$$\frac{\partial E}{\partial v_{jk}} = (y_k - d_k) y_k (1 - y_k) y_j^c.$$

Выведем теперь формулу для производной  $\frac{\partial E}{\partial w_{ij}}$ . Аналогично запишем:

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial y_j^c} \frac{\partial y_j^c}{\partial s_j} \frac{\partial s_j}{\partial w_{ij}}.$$

Здесь  $s_j = \sum_{i=1}^n w_{ij} x_i$ , поэтому  $\frac{\partial s_j}{\partial w_{ij}} = x_i$  ( $x_i$  –  $i$ -я компонента, поданного на вход образа обучающей выборки);

$$\frac{\partial y_j^c}{\partial s_j} = y_j^c (1 - y_j^c)$$

Так как функция ошибки не зависит в явном виде от выходов скрытого слоя  $y_j^c$ , то

производная  $\frac{\partial E}{\partial y_j^c}$  усложняется:  $\frac{\partial E}{\partial y_j^c} = \sum_{k=1}^p \frac{\partial E}{\partial y_k} \frac{\partial y_k}{\partial s_k} \frac{\partial s_k}{\partial y_j^c}$ .

Воспользовавшись имеющимися выражениями для  $\frac{\partial E}{\partial y_k}$ ,  $\frac{\partial y_k}{\partial s_k}$  и  $s_k$ , запишем:

$$\frac{\partial E}{\partial y_j^c} = \sum_{k=1}^p (y_k - d_k) y_k (1 - y_k) v_{jk}.$$

Если ввести обозначение  $\delta_k = \frac{\partial E}{\partial y_k} \frac{\partial y_k}{\partial s_k} = (y_k - d_k)y_k(1 - y_k)$ ,

получим следующие выражения для производных:

$$\frac{\partial E}{\partial v_{jk}} = \delta_k y_j^c, \quad \frac{\partial E}{\partial w_{ij}} = \left( \sum_{k=1}^p \delta_k v_{jk} \right) y_j^c (1 - y_j^c) x_i$$

### Алгоритм обучения сети обратного распространения

Рассмотрим теперь полный алгоритм обучения нейросети.

*Шаг 1. Инициализация сети.*

Весовым коэффициентам присваиваются малые случайные значения, например, из диапазона  $(-0.3, 0.3)$ ; задаются  $\varepsilon$  – параметр точности обучения,  $\alpha$  – параметр скорости обучения (как правило  $\alpha \approx 0,1$  и может еще уменьшаться в процессе обучения),  $N_{\max}$  – максимально допустимое число итераций.

*Шаг 2. Вычисление текущего выходного сигнала.*

На вход сети подается один из образов обучающей выборки и определяются значения выходов всех нейронов нейросети.

*Шаг 3. Настройка синаптических весов.*

Рассчитать изменение весов для выходного слоя нейронной сети по формулам

$$v_{jk}^{N+1} = v_{jk}^N - \alpha \frac{\partial E}{\partial v_{jk}}, \quad \text{где } \frac{\partial E}{\partial v_{jk}} = \delta_k y_j^c, \quad \delta_k = (y_k - d_k)y_k(1 - y_k).$$

Рассчитать изменение весов для скрытого слоя по формулам

$$w_{ij}^{N+1} = w_{ij}^N - \alpha \frac{\partial E}{\partial w_{ij}}, \quad \text{где } \frac{\partial E}{\partial w_{ij}} = \left( \sum_{k=1}^p \delta_k v_{jk}^{N+1} \right) y_j^c (1 - y_j^c) x_i.$$

*Шаг 4.*

Шаги 2–3 повторяются для всех обучающих векторов. Обучение завершается по достижении для каждого из обучающих образов значения функции ошибки, не превосходящего  $\varepsilon$  или после максимально допустимого числа итераций.

### Оптимальное число скрытых нейронов

Способность персептрона-классификатора разделять образы в пространстве признаков прежде всего зависит от его скрытого слоя. Именно на этот слой возлагается задача сделать множество классов линейно разделимым для успешной работы выходного слоя. Очевидно, что чем больше нейронов в скрытом слое, тем большее количество примеров этот слой может разделять. Кроме этого, увеличение числа признаков входных образов также способствует успешному их разделению в пространстве признаков. Однако увеличение этих параметров приводит к росту ошибок сети и времени обучения. Увеличение размерности входов  $n$  приводит к росту ошибки аппроксимации сети, возникающей из-за обобщения данных. Увеличение числа нейронов  $h$  скрытого слоя приводит к росту ошибки, связанной со сложностью модели. Персептрону легче провести функцию через эталонные точки, однако при этом обобщающая способность сети ухудшается. Он хуже предсказывает поведение аппроксимируемой функции на образах, не входящих в обучающую выборку. Такое состояние сети называется переобучением.

Оптимальное соотношение между этими параметрами оценивают как

$$h \sim \sqrt{\frac{p}{n}}.$$

Эксперименты показывают, что обучение максимально успешно проходит на множестве классов, хорошо (желательно линейно) разделенных в пространстве признаков. Это достигается

удачным подбором информативных признаков. Если классы в пространстве признаков хорошо кластеризуются, т.е. образы каждого класса составляют компактную группу, достаточно удаленную от других групп, то есть возможность уменьшить размер обучающей выборки  $p$  (используются только центры кластеров) и затем уменьшить число нейронов  $h$ . Это приводит к ускорению обучения и улучшает работу классификатора.

**Замечание 1.** На шаге 2 векторы из обучающей последовательности лучше предъявлять на вход в случайном порядке.

**Замечание 2.** Во многих случаях желательно наделять каждый нейрон обучаемым смещением. Это позволяет сдвигать начало отсчета логистической функции, давая эффект, аналогичный подстройке порога персептронного нейрона, и приводит к ускорению процесса обучения. Эта возможность может быть легко введена в обучающий алгоритм с помощью добавления к каждому нейрону дополнительного входа. Его вес обучается так же, как и все остальные веса, за исключением того, что подаваемый на него сигнал всегда равен  $+1$ , а не выходу нейрона предыдущего слоя.

**Замечание 3.** Количество входов и выходов сети, как правило, диктуется условиями задачи, а размер скрытого слоя находят экспериментально. Обычно число нейронов в нем составляет 30–50 % от числа входов. Слишком большое количество нейронов скрытого слоя приводит к тому, что сеть теряет способность к обобщению (она просто досконально запоминает элементы обучающей выборки и не реагирует на схожие образцы, что неприемлемо для задач распознавания). Если число нейронов в скрытом слое слишком мало, сеть оказывается просто не в состоянии обучиться.

**Замечание 4.** В результате слишком слишком близкой подгонки может возникнуть проблема переобучения. Если данные в обучающей выборке содержали шумы, то в результате переобучения сеть запоминает эти шумы и утрачивает способность к обобщению. Для контроля за переобучением сети вся доступная исходная выборка должна быть разделена на два множества – обучающее ( $\approx 75\%$  данных) и тестовое ( $\approx 25\%$  данных).

**Замечание 5.** Выходы каждого нейрона сети лежат в диапазоне  $(0,1)$  – области значений логистической функции – это надо учитывать при формировании обучающей выборки. Желательно, чтобы и входы сети принадлежали этому диапазону. Если надо получить от сети бинарный выход, то, как правило, вместо 0 используют 0,1, а вместо «1» 0,9, так как границы интервала слишком недостижимы.

### Модификации алгоритма

Модификации алгоритма обратного распространения связаны с использованием различных функций ошибки, других активационных функций, различных процедур определения направления и величины шага.

Если значения выходов большинства нейронов задолго до окончания обучения становятся близки к асимптотическим значениям функции активации (для логистической функции это 0 и 1), то возникает так называемый **паралич сети**. Производная функции активации в этом случае близка к 0 и весовые коэффициенты практически перестают изменяться. В итоге обучение становится неприемлемо медленным. Простейшим усовершенствованием метода градиентного спуска является введение т. н. момента, когда влияние градиента на изменение весов накапливается со временем. Дополнительное преимущество от введения момента – появляющаяся у алгоритма способность преодолевать мелкие локальные минимумы. Правда, недостатком данного метода является введение еще одного глобального настроенного параметра.

Еще одной удачной модификацией обучения является алгоритм RPROP, в котором каждый вес имеет свой адаптивно настраиваемый темп обучения. RPROP использует лишь знаки частных производных по каждому весу, а не их значения

Обратное распространение было использовано в широкой сфере прикладных исследований. В частности фирма NEC в Японии использовала обратное распространение для визуального распознавания букв (в том числе рукописных), причем точность превысила 99 %. Достигнут впечатляющий успех с Net-Talk, системой, которая превращает печатный английский текст в высококачественную речь. Магнитофонная запись процесса обучения сильно напоминает звуки ребенка на разных этапах обучения речи.

Но несмотря на многочисленные успешные применения обратного распространения, оно не является панацеей. Больше всего неприятностей приносит неопределенно долгий процесс обучения. В сложных задачах для обучения сети могут потребоваться часы или даже дни, она может и вообще не обучиться. Неудачи в обучении часто возникают по причине попадания сети в локальный минимум, что, к сожалению, является характерной особенностью методов градиентного спуска. Исправить ситуацию в таком случае иногда помогают небольшие случайные изменения весовых значений сети.

## ПРЕДОСТЕРЕЖЕНИЕ

Несмотря на многочисленные успешные применения обратного распространения, оно не является панацеей. Больше всего неприятностей приносит 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

Если один из весов при обучении получает слишком большое значение, то при обычных значениях этого входа выход нейрона окажется в насыщении, т.е. будет близок к предельному значению функции активации. Выход нейрона будет мало зависеть от  $w$ , и поэтому производная  $\frac{\partial E}{\partial w} \approx 0$ . Обучение по этому весу будет очень медленным, ведь изменение веса пропорционально производной. Выходной сигнал нейрона будет мало зависеть не только от веса, но и от входного сигнала  $x$  данного нейрона, а производная по  $x$  участвует в обратном распространении ошибки. Следовательно, предшествующие нейроны тоже будут обучаться медленно. Такое замедление обучения называется *параличом сети*.

В теоретическом отношении эта проблема плохо изучена. Обычно этого избегают уменьшением размера шага  $\alpha$ , 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

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.

Другой проблемой является то, что алгоритм градиентного спуска не гарантирует нахождение глобального минимума среднеквадратичной ошибки сети, а гарантируется определение только локального минимума. Проблемы, возникающие в процессе градиентного спуска, можно проанализировать на примере функции ошибки, схематически изображенной на рис. 2.2.

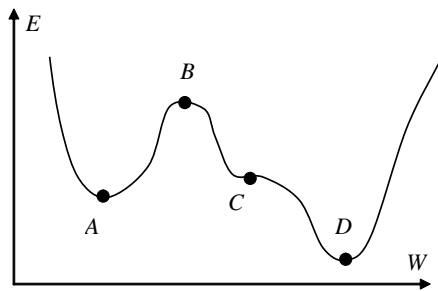


Рис. 2.2. Схематическая функция ошибки сети

На рис. 2.2 показаны четыре критические точки, производная функции ошибки в которых близка к нулю. Точка A соответствует локальному минимуму. Признаком достижения локального минимума в процессе обучения является полное прекращение уменьшения ошибки (2.17).

$$\max |d_k| < D, k = \overline{l, m, r} = \overline{l, p}, \quad (2.17)$$

В этом случае может помочь повторное обучение с другим начальным распределением знаний (тут может помочь случайная инициализация). Точка В – локальный максимум. В случае попадания в окрестность такой точки скорость резко падает, затем снова быстро растет. Не существует способа предсказать, в какую сторону лучше двигаться из точки с производной, близкой к нулю. Точка С – точка перегиба, характеризуется длительным уменьшением скорости. Точка D – глобальный минимум – цель алгоритма.

Существуют многочисленные способы оптимизации метода градиентного спуска, призванные улучшить поведение алгоритма в подобных критических точках. Эффективной модификацией является введение момента, накапливающего влияние градиента на веса со временем. Тогда величина  $\Delta w$  (2.3) в момент времени  $t$  будет вычисляться как

$$\Delta w(t) = \alpha \frac{\partial E}{\partial w} + \mu \Delta w(t-1), \quad (2.19)$$

где  $\mu$  – параметр, определяющий величину влияния момента. С использованием (2.19) скорость изменения весов возрастает на участках с постоянным знаком производной. В окрестностях минимума скорость резко падает за счет колебания знака.

Достоинства алгоритма – большая скорость в точках перегиба, возможность по инерции преодолевать небольшие локальные минимумы. Недостатки – еще один параметр, величину которого следует подбирать и настраивать.

Этот и другие алгоритмы оптимизации обучения персептрона позволяют улучшить работу сети в условиях плохой сходимости. Однако они усложняют процесс обучения, не гарантируя в то же время полного успеха во всех случаях. Успех обучения классификатора зависит от самого алгоритма обучения и качества обучающей выборки.

### ***Размер шага***

Внимательный разбор доказательства сходимости в [7] показывает, что коррекции весов предполагаются бесконечно малыми. Ясно, что это неосуществимо на практике, так как ведет к бесконечному времени обучения. Размер шага должен брать конечным, и в этом вопросе приходится опираться только на опыт. Если размер шага очень мал, то сходимость слишком медленная, если же очень велик, то может возникнуть паралич или постоянная неустойчивость. В [11] описан адаптивный алгоритм выбора шага, автоматически корректирующий размер шага в процессе обучения. (Уоссерман)

### ***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 [7] 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, а это 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. Как будет указано в гл. 8, это несоответствие (среди прочих) привело к системе ART, принадлежащей Гроссбергу.

## **ДАЛЬНЕЙШИЕ АЛГОРИТМИЧЕСКИЕ РАЗРАБОТКИ**

Многими исследователями были предложены улучшения и обобщения описанного выше основного алгоритма обратного распространения. Литература в этой области слишком обширна, чтобы ее можно было здесь охватить. Кроме того, сейчас еще слишком рано давать окончательные оценки. Некоторые из этих подходов могут оказаться действительно фундаментальными, другие же со временем исчезнут. Некоторые из наиболее многообещающих разработок обсуждаются в этом разделе.

В [5] описан метод ускорения сходимости алгоритма обратного распространения. Названный обратным распространением второго порядка, он использует вторые производные для

более точной оценки требуемой коррекции весов. В [5] показано, что этот алгоритм оптимален в том смысле, что невозможно улучшить оценку, используя производные более высокого порядка. Метод требует дополнительных вычислений по сравнению с обратным распространением первого порядка, и необходимы дальнейшие эксперименты для доказательства оправданности этих затрат.

В [9] описан привлекательный метод улучшения характеристик обучения сетей обратного распространения. В работе указывается, что общепринятый от 0 до 1 динамический диапазон входов и выходов скрытых нейронов неоптимален. Так как величина коррекции веса  $\Delta w_{pq,k}$  пропорциональна выходному уровню нейрона, порождающего  $OUT_{p,j}$ , то нулевой уровень ведет к тому, что вес не меняется. При двоичных входных векторах половина входов в среднем будет равна нулю, и веса, с которыми они связаны, не будут обучаться! Решение состоит в приведении входов к значениям  $\pm 1/2$  и добавлении смещения к сжимающей функции, чтобы она также принимала значения  $\pm 1/2$ . Новая сжимающая функция выглядит следующим образом:

$$OUT = -\frac{1}{2} + \frac{1}{1 + e^{-NET}} \quad (3.13)$$

С помощью таких простых средств время сходимости сокращается в среднем от 30 до 50%. Это является одним из примеров практической модификации, существенно улучшающей характеристику алгоритма.

В [6] и [1] описана методика применения обратного распространения к сетям с обратными связями, т. е. к таким сетям, у которых выходы подаются через обратную связь на входы. Как показано в этих работах, обучение в подобных системах может быть очень быстрым и критерии устойчивости легко удовлетворяются.

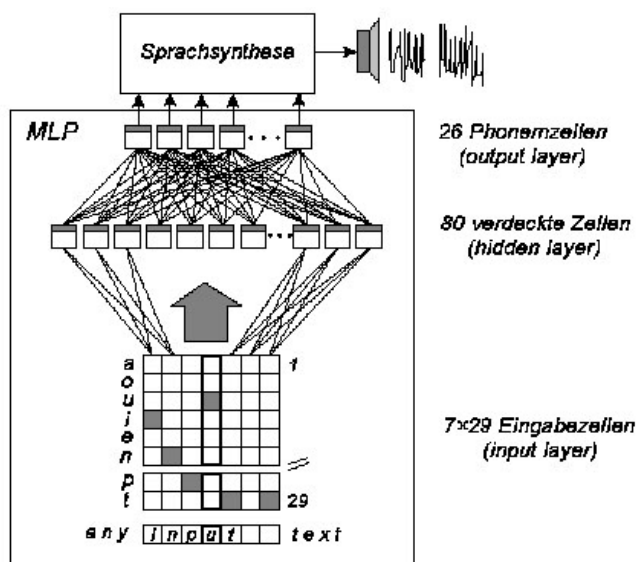
## ПРИМЕНЕНИЯ

Для успешного решения реальных задач с помощью ИНС необходимо определить ряд их характеристик, включая модель сети, ее размер, функцию активации или выхода, параметры алгоритма обучения и множество образов, классифицированных учителем, разделенное на обучающую и контролирующую последовательности. Многие задачи прикладной нейроматематики не решаются на известных типах ЭВМ. Среди них задачи, сводимые к обработке многомерных векторов действительных переменных с помощью ИНС, например: контроль кредитных карточек. В настоящее время 60% кредитных карточек в США обрабатываются с помощью нейросетевых технологий. Одним из первых примеров применения Backpropagation-сетей был проект NETtalk (Sejnowski und Rosenberg, 1986). Цель разработки состояла в преобразовании ASCII-текста в речевой сигнал.

### NETtalk – проект

Talk [to:k]

NETtalk – система предназначена для преобразования печатного текста на английском языке из 29 возможных символов в последовательность фонем их 26 символов. Она состоит из двух модулей:



многослойный персептрон, преобразующий английский текст из 29 букв в последовательность фонем (26 символов); синтезатор речи, преобразующий последовательность фонем в соответствующие акустические сигналы (рис. 2.7).

В работе [8] достигнут впечатляющий успех с Net-Talk, системой, которая превращает печатный английский текст в высококачественную речь. Магнитофонная запись процесса обучения сильно напоминает звуки ребенка на разных этапах обучения речи.

В [2] обратное распространение использовалось в машинном распознавании рукописных английских слов. Буквы, нормализованные по размеру, наносились на сетку, и брались проекции линий, пересекающих квадраты сетки. Эти проекции служили затем входами для сети обратного распространения. Сообщалось о точности 99,7% при использовании словарного фильтра.

Печатный текст перемещается перед камерой. Одновременно считываются и анализируются 7 символов. Используется окно камеры на  $3 + 1 + 3$  символа: один текущий символ, а также три предшествующих и три последующих (для учета статистических связей между буквами английского текста).

Входной слой ИНС содержит  $7 \times 29$  бинарных ячеек. В каждом столбце активизируется ячейка, маркирующая соответствующий символ. Скрытый слой содержит 80 нейронов, а выходной слой – 26 нейронов по числу фонем и 18629 связей.

Достигнута достоверность преобразования 0,98.

Обратное распространение было использовано в широкой сфере прикладных исследований. Некоторые из них описываются здесь, чтобы продемонстрировать мощь этого метода.

Фирма NEC в Японии объявила недавно, что обратное распространение было ею использовано для визуального распознавания букв, причем точность превысила 99%. Это улучшение было достигнуто с помощью комбинации обычных алгоритмов с сетью обратного распространения, обеспечивающей дополнительную проверку.

В [3] сообщалось об успешном применении обратного распространения к сжатию изображений, когда образы представлялись одним битом на пиксель, что было восьмикратным улучшением по сравнению с входными данными.

система скрытого обнаружения веществ системой на основе тепловых нейронов и нейрокомпьютера на заказных цифровых нейрочипах. Подобная система фирмы SAIC используется уже во многих аэропортах США при досмотре багажа для обнаружения наркотиков, взрывчатых веществ, ядерных и других материалов;

система автоматизированного контроля безопасного хранения ядерных изделий;

адаптация диалога компьютерных систем к индивидуальным характеристикам пользователя. Применительно к компьютерным обучающим системам соответствующий подход на основе параметрических и непараметрических методов их обучения был предложен и разработан Свиридовым А.П. (1976). Этот подход был затем реализован UNIX-консультанте (Rich E. (1979), Wilensky, R., Chin, D.N., Luria, M., Martin, J., Mayfield, J., Wu, D.: Arens, Y. (1984, 1988)).

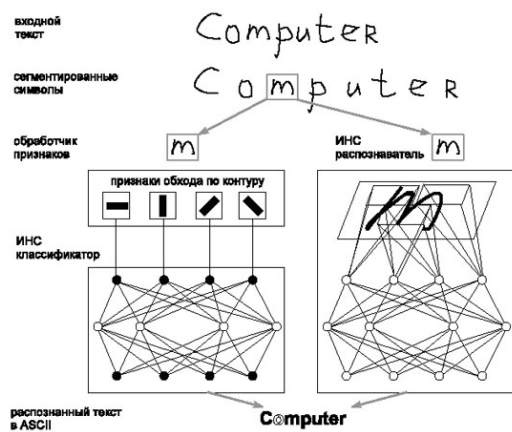
Наиболее перспективные задачи обработки изображений нейросистемами и нейрокомпьютерами - обработка аэрокосмических изображений (сжатие с восстановлением, сегментация, контрастирование и обработка текстур), выделение на изображении движущихся целей, поиск и распознавание на нем объектов заданной формы, анализ потоков изображений, обработка информации в высокопроизводительных сканерах.

Рассмотрим несколько подробнее применения сетей прямого распространения информации для распознавания изображений и компьютерного контроля знаний. Применение сетей прямого распространения информации для распознавания изображений рассмотрим на простейшем примере системы распознавания символов (задача OCR=Optical Character Recognition), предназначенной для обработки отсканированного изображения текста и его преобразовании затем в машинную форму.

***Использование нейронных сетей для распознавания изображений (система OCR).***

Система OCR обычно состоит из блоков препроцессирования, сегментации, выделения характеристик, классификации и контекстуальной обработки информации. Бумажный документ сканируется, и создается, например, бинарное (черно-белое) изображение. На стадии препроцессирования применяется фильтрация для удаления шума, область текста локализуется и преобразуется в бинарное изображение посредством адаптивных пороговых преобразователей. На шаге сегментации изображение текста разделяется на отдельные символы. Эта задача особенно трудна для рукописного текста, который содержит связи между соседними символами.





Один из эффективных приемов состоит в расчленении составного образа на малые образы (промежуточная сегментация) и нахождении точек правильной сегментации с использованием выхода классификатора образов. Вследствие различного наклона, искажений, помех и стилей письма распознавание сегментированных символов является достаточно сложной задачей.

Рис. 2.8. Две схемы применения ИНС в системах OCR.

На рис. 2.8 представлены два основных подхода к использованию ИНС в OCR-системах. Первая выполняет явное извлечение характерных признаков (не обязательно на нейронной сети). Например, это могут быть признаки обхода по контуру. Выделенные признаки подаются на вход многослойной сети прямого распространения. Эта схема отличается гибкостью в отношении использования большого разнообразия признаков. Другая схема не предусматривает явного выделения признаков из исходных данных. Извлечение признаков происходит неявно в скрытых слоях ИНС. Удобство этой схемы состоит в том, что выделение признаков и классификация объединены. Однако эта схема требует большего размера сети, чем в первом случае.

На первой конференции по OCR-системам в 1992 г. более 40 систем распознавания рукописного текста были сопоставлены для одних и тех же данных. В 10 лучших вариантах распознавание производилось на основе многослойной сети прямого распространения (многослойного перцептрона) или классификатора по минимуму расстояния до эталонных точек (прототипов, стереотипов, типичных представителей отдельных символов). Последнее правило иногда называется правилом «ближайшего соседа». Экспериментально установлена более высокая эффективность систем распознавания на основе ИНС. Точность распознавания лучших OCR-систем на базе данных предварительно сегментированных символов составила около 98% для цифр, 96% для заглавных букв и 87% – для строчных. Низкая точность для строчных букв вызвана в значительной степени тем, что тестовые данные существенно отличались от данных обучающей последовательности.

По данным теста можно сделать вывод, что на изолированных символах OCR-система близка по точности распознавания к человеку. Однако человек превосходит системы OCR для овукументах, свободных от ограничений и рукописных.

### ***Задачи обработки видеоизображений движущихся объектов***

Одной из наиболее сложных и актуальных проблем обработки видеоизображений, представленных последовательностью оцифрованных кадров, является проблема выделения и распознавания движущихся объектов в условиях действия различного рода помех и искажений. Для ее решения разработана специализированная многослойная локально-связная нейронная сеть, осуществляющая выделение изображений движущихся объектов на сложном зашумленном фоне при фильтрации помех, а также оценку скорости каждого объекта, его идентификацию и сопровождение. Система обрабатывает реальные данные телевизионной системы (25 кадров/с, 320 × 200 пикселей). Размерность сети для изображения 320 × 200 пикселей составляет несколько миллионов нейронов и примерно вчетверо больше синапсов. Вероятность правильного распознавания составляет около 90%.

Одной из наиболее сложных и актуальных проблем обработки видеоизображений, представленных последовательностью оцифрованных кадров, является проблема выделения и распознавания движущихся объектов в условиях действия различного рода помех и искажений. Для ее решения разработана специализированная многослойная локально-связная нейронная сеть, осуществляющая выделение изображений движущихся объектов на сложном зашумленном

фоне при фильтрации помех, а также оценку скорости каждого объекта, его идентификацию и сопровождение. Система обрабатывает реальные данные телевизионной системы (25 кадров/с,  $320 \times 200$  пикселей). Размерность сети для изображения  $320 \times 200$  пикселей составляет несколько миллионов нейронов и примерно вчетверо больше синапсов. Вероятность правильного распознавания составляет около 90%.

#### ***Задачи обработки статических изображений***

Не менее сложными являются задачи выделения и распознавания статических изображений, которые возникают, в частности, при автоматической обработке спутниковых изображений земной поверхности. Для их решения создана автоматизированная система анализа изображений земной поверхности, полученных в оптическом диапазоне с искусственного спутника Земли. Система распознает и выделяет на обрабатываемых изображениях объекты заданных классов: дорожной сети, кварталов с характерной застройкой, аэродромов и стоящих на них самолетов. Система, способная к обучению и переобучению, инвариантна к яркостным характеристикам объектов.

#### ***Задачи обнаружения и классификации летательных аппаратов по звуку***

Для выделения сигнала с заданного направления используется фазированная антенная решетка с широкополосными сигналами на выходе, которые подвергаются предобработке, а затем в оцифрованном виде подаются на вход предварительно обученной ИНС для распознавания.