

3. Применение НС для распознавания изображений

3.1. Введение

3.2. Сеть динамических связей

3.3. Проблема адаптации распознавания объектов

3.4. Заключение

3.1. Введение

Рассмотрев основные вопросы теории НС мы можем перейти к вопросам конкретного их применения в задаче распознавания изображений. Также рассмотрим основные проблемы, которые возникают на пути внедрения НС в данную область:

- ограниченность возможности обработки больших изображений;
- отсутствие инвариантности к перемещениям объекта в изображении;
- хранение больших объемов информации в памяти;
- низкая скорость обучения.

В предыдущей главе были описаны основные виды нейронных сетей, методы их контролируемого (с учителем) и неконтролируемого (без учителя) обучения и организации ассоциативной памяти. И, как было показано, при реализации данных методов НС также могут страдать от ограничений. Например, уровень обучаемости для многослойных НС обратного распространения очень низок, и данную модель в этом случае невозможно применить для распознавания больших изображений, имеющих высокую разрядность, которую в настоящий момент способны выдавать современные СТЗ (800×600 и выше). Возможность хранения больших объемов информации в ассоциативной памяти обычной сети Хопфилда также низка, и сеть выдаст ошибку при ее переполнении.

Однако мы можем решить некоторые из этих проблем, изменив метод подачи сигнала на входы сети, а также его тип. В случае многослойной НС обратного распространения входной сигнал может быть выражен через сигма-пи составляющие [Rummelhart, Hinton и Williams, 1986] или через динамические связи, которые активизируются по значению выхода корреляционной функции в обобщенной модели правила дельты [Pomerleau, 1987]. Такого рода методы могут оказать существенное воздействие на повышения уровня обучаемости сети без ее неконтролируемого увеличения. Подобным образом в 1986 году было предложено увеличить объем памяти в сетях Хопфилда, введя высокоуровневые динамические внутренние связи [26].

В этой главе мы рассмотрим, наряду с общими вопросами проблематики применения НС в задаче распознавания образов, несколько другие методы увеличения применимости многослойной НС в данном контексте.

Результаты описания парадигмов НС пока еще не дали ответа на вопрос об их преимуществе перед стандартными методами обработки изображений и распознавания объектов, на чем также будет заострен вопрос.

3.2. Сеть динамических связей

Существуют разные методы реализации алгоритмов многослойной НС обратного распространения. С одной точки зрения, нейроны первого скрытого слоя образуют некую поверхность, нейроны второго слоя образуют область значений, и другие слои определяют правила установки результатов решений AND и OR операций [22]. С другой, также имеющей силу, точки зрения последовательные слои определяют последовательность наложений до тех пор, пока мы не получим описание – что это, наложение именно в том в выбранном пространстве, где будет достигнуто их желательное взаимоположение и разделение [6]. Рассмотрим более подробно второй подход.

Вместо группирования по кругу, пока мы не найдем нужную последовательность трансформаций, мы можем увеличить оригинальное описание изображения прямо с начала там, где это возможно, в линейно независимой форме, для того, чтобы гиперплоскости нейронов могли быть обучены в более наглядной форме. Первый путь увеличить описание входного изображения, это составить описание в многомерном пространстве. Тогда встают вопросы :

- каким образом должны быть выбраны дополнительные измерения?,
- какие именно значения должны быть описаны по выбранным дополнительным осям?

Мы обоснуем данный подход в терминах динамических связей. Таким образом линейные синаптические веса создаются на линейных связях многослойной НС, а динамические связи действуют на определенные элементы входного изображения либо на все изображение в целом посредством генерации линейно независимых функций с последующей оценкой изображения через эти функции, для которых оно является входным аргументом. То есть, в сам процесс не внедряется никакой новой информации; тем не менее, описание определенно увеличивается, и ее разделение может быть достигнуто в новом многомерном пространстве.

Было замечено, что применение динамических связей не только увеличивает уровень обучаемости сети, но также имеет положительный побочный эффект, выражающийся в упрощении обучающих алгоритмов и унификации архитектуры всех трех основных типов нейронных сетей. Таким образом новая сеть является более приспособленной ко всем трем основным задачам: контролируемому обучению, организации ассоциативной памяти и неконтролируемого обучения.

Когда ведется разговор об увеличении разрядности входного сигнала сети динамических связей (СДС), совершенно ясно, что мы не заинтересованы непосредственно в увеличении количества переменных на входе сети. Однако в этом случае, непосредственной задачей СДС является размножение каждого элемента входного изображения во входной вектор, то есть получение в результате тензора, который является выходным продуктом обработки входного изображения.

Немного позже несколько неожиданно было обнаружено [6], что корреляция с целью увеличения разрядности входного сигнала может быть рассмотрена как специальный случай сети динамических связей. Но, если говорить строго, то данная задача не является основным предназначением сети.

3.2.1. Принцип трансформации по нелинейным связям

В контексте данной задачи можно найти решение от противного на основе так называемого «моментального» подхода. В этом смысле мы можем снова прибегнуть к помощи нейробиологии или применить подход, основанный на правилах и методах математики. Объяснить принцип действия СДС будет проще, если применить второй подход, однако в результате станет ясно, что полученные выкладки только подтвердят знания, приобретенные из биологического построения нейронных сетей.

Основной принцип СДС заключен именно в понятии динамики или функциональности ее связей. Таким образом математическая модель сети более подходит для распараллеливания процессов, происходящих в сети, а различные встроенные функции нейрона могут быть реализованы только в случае активизации самого нейрона. Как показано на рис.3.1, активизация нейрона k может сделать возможным реализацию процессов $f_0(o_k), f_1(o_k), \dots, f_n(o_k)$.

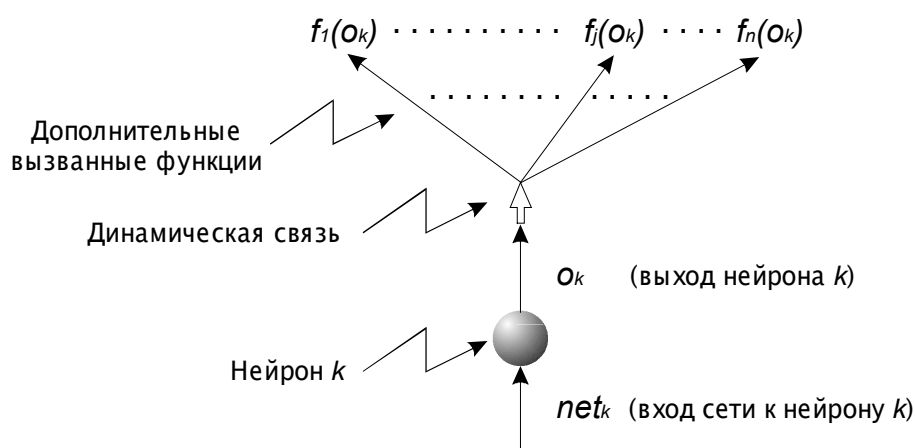


Рис. 3.1

Мы можем получить различные результаты работы СДС опираясь на различные постулаты для данной сети. Таким образом, рассмотрим две основных модели сети: модель динамического расширения и модель получения тензора. Последняя модель является частным случаем основного метода, применяемого к СДС. Данный вывод был получен на основе того, что результаты, полученные через призму второй модели, сходятся результатами других математических методов, которые берут за основу корреляционные постулаты.

В модели динамического расширения СДС активизирует каждый нейрон по отдельности. Это означает активизацию встроенных функций для каждого нейрона входного слоя. Данная концепция схематически представлена на рис. 3.2. в этой модели, каждый компонент входного вектора активизируется динамической связью соответственно параметрам $f_1(o_k), f_2(o_k), \dots, f_n(o_k)$.

При некоторых обстоятельствах, $f_1(x)$ может быть просто равна x , $f_2(x)$ равна x^2 и так далее. Либо функции могут быть заданы определенным числом ортогональных базисных функций, расположенных в n -мерном пространстве описания, например, как $\sin \pi x$, $\cos \pi x$, $\sin 2\pi x$, $\cos 2\pi x$ и так далее. Основной задачей сети при этом является конвертирование входной карты изображения в карту, описанную в многомерном пространстве, или, другими словами, расширенную карту. Мы составим новое описание на основе преобразования каждого компонента x вместе с его степенями x, x^2, x^3 и так

далее, или вместе с x , $\sin \pi x$, $\cos \pi x$ и так далее, основываясь на заданных функциях, которые мы считаем подходящими. При этом никакой новой информации в изображение не привносится, однако его описание увеличивается.

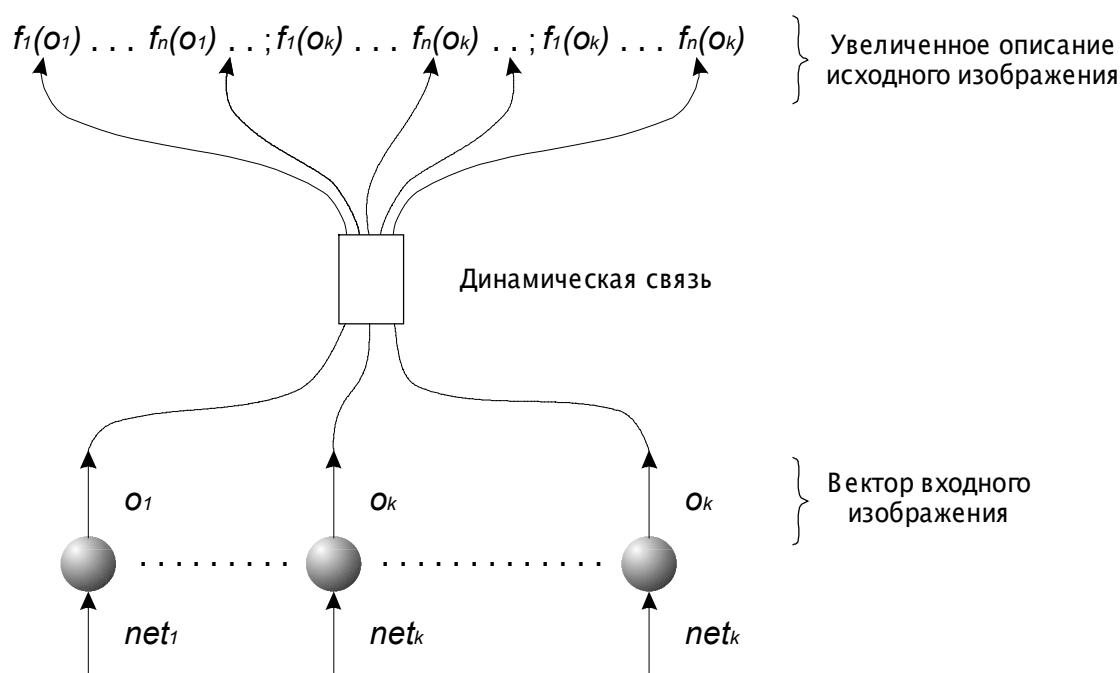


Рис. 3.2

Тензор является всего лишь частным случаем описанной выше модели. В модели тензора каждый компонент входного изображения мультиплицируется в отдельный входной вектор, при помощи динамической связи. Данный процесс может быть описан на основе информации об образовании тензора между исходным входным вектором и другим таким же вектором, полученным из оригинального увеличением его на основе значений, являющихся результатом применения расширенных функций. Увеличение вектора позволит осуществить регенерацию исходного изображения.

В обеих моделях процесс увеличения описания может использоваться периодически либо нет, когда требуется, причем оба метода могут использоваться вместе и по отдельности.

В модели тензора основной эффект нелинейной динамической трансформации выражается в изменении входного изображения, опираясь на описание компонентов $\{x_i\}$, представленных как:

$$\left\{ x_i, x_i x_j \right\}_{i \geq j},$$

или как :

$$\left\{ x_i, x_i x_j, x_i x_j x_k \right\}_{\substack{i \geq j \\ k \geq j \geq i}},$$

и так далее. В этом случае снова никакой новой информации в описание изображения не вводится, но активация соединений становится более доступна и наглядна.

Такого рода трансформации довольно сильно увеличивают количество компонентов в терминах описания входного изображения. Мы можем упростить процесс увеличения

описания, повторяя термины и придавая им большее количество индексов, а также используя термины, в которых корреляция не согласована со входным изображением. Полученная последовательность трансформаций выражена в формуле (3.1):

$$\{x_i\} \Rightarrow \left\{ x_i, x_i x_j \right\}_{j \geq i} \Rightarrow \left\{ x_i, x_i x_j, x_i x_j x_k \right\}_{\substack{j \geq i \\ k \geq j \geq i}} \Rightarrow \dots \quad (3.1)$$

и пропускаем термины, для которых:

$$\sum_{p=1}^P (x_i x_j)_p \rightarrow 0. \quad (3.2)$$

Табл.3.1 Увеличение количества компонентов изображения

Количество компонентов в исходном изображении $\{x_i\}$	Количество компонентов в $\left\{ x_i, x_i x_j \right\}_{i \geq j}$	Количество компонентов в $\left\{ x_i, x_i x_j, x_i x_j x_k \right\}_{\substack{i \geq j \\ k \geq j \geq i}}$
2	3	3
3	6	7
4	10	14
5	15	25
10	35	155

В табл. 3.1 значения переменных x_i и x_j изменяются в диапазоне от -1 до $+1$. На практике пропускаются только диагональные термины, так как увеличение компонентов в целом может быть некорректным. Такого рода увеличение может привести к непредвиденным результатам, соответствующие примеры будут приведены немного позже в этой главе. Для контролируемого обучения, сильно увеличивается уровень обучения не только для многослойных сетей, но также часто и для однослойных сетей, для которых это совершенно не нужно. Обучение с учителем таким образом может воздействовать также и на сети, архитектура которых создана для другого типа обучения (как например, для сетей архитектуры ART).

Непосредственно сами примеры приведены в пункте 3.2.3.

Пример 1 в частности демонстрирует положительное качество получения карты изображения через увеличение описания исходного сигнала. Принимая во внимание то, что проблема четности по модулю 2 (XOR) не может быть решена без наличия скрытого слоя в НС, можно найти упрощенное решение используя представление входного изображения в трехмерном пространстве на основе терминов расширения описания. В таком случае несколько увеличится уровень обучаемости сети.

Пример 2 представляет собой другую демонстрацию упрощения архитектуры сети вследствие применения выше описанного метода. Результаты применения метода соответствуют результатам, полученным Максвеллом [26]. Если проводить расширение описания через призму модели тензора, то некоторые из его основных терминов представляют активационные функции связей. С другой стороны, модель динамического расширения просто увеличивает количество измерений пространства описания, без введения активационных функций. Опыт ведущих ученых в данной области показывает, что низкая степень расширения приводит к различным непредвиденным результатам. Прямое увеличение значения каждого компонента в отдельности первоначально приводит к увеличению разрешения классификационных или оценочных активационных функций, подключенных к изображению.

Модель тензора вычисляет с гораздо большей скоростью качественный характер результата, то есть определяет, какая именно классификация наиболее подходит к данной модели сети без скрытых слоев.

3.2.2. Основные математические принципы построения СДС

В данном пункте мы не будем приводить сложных математических выкладок и полных выводов формул, за полной информацией можно обратиться к специальной литературе. Просто сошлемся на то, что современные решения в данной области действительно доказывают что, в принципе, проблема контролируемого обучения может быть решена для однослойных сетей, при помощи подхода на основе динамических связей. Таким образом можно доказать, что для решения задач распознавания совершенно не обязательно наличие скрытых слоев.

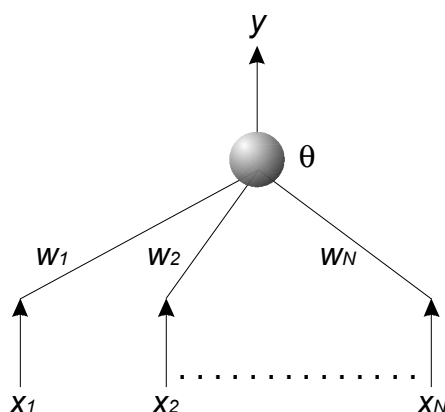


Рис.3.3

Рассмотрим возможность обучения однослойных сетей более подробно. Пусть существуют P входных изображений, каждая из которых состоит из N элементов. Применим такой подход для сети с одним выходом, показанной на рис. 3.3. Для изображения p , карта входных компонентов выражается как $x_i^{(p)}$, а выходных как $y^{(p)}$. Синаптические веса связей обозначим как w_i , а порог активизации нейрона как θ , это является верным для всех изображений $p=1,2,3,\dots,P$.

Таким образом, мы можем выразить формулу:

$$y = \frac{1}{1 + e^{-net}}, \quad (3.3)$$

или

$$net = \ln \left(\frac{y}{1-y} \right)^{1/\alpha}. \quad (3.4)$$

Следовательно:

$$w_1 x_1^{(p)} + \dots + w_N x_N^{(p)} + \theta = z^{(p)} \quad p = 1, 2, \dots, P \quad (3.5)$$

или, для всех изображений,

$$\begin{bmatrix} x_1^{(1)} & \dots & x_N^{(1)} & 1 \\ \vdots & \ddots & \vdots & \vdots \\ x_1^{(P)} & \dots & x_N^{(P)} & 1 \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_N \\ \theta \end{bmatrix} = \begin{bmatrix} z^{(1)} \\ \vdots \\ z^{(P)} \end{bmatrix}. \quad (3.6)$$

Таким образом, нахождение весовых коэффициентов для однослойной сети включает в себя решение системы линейных уравнений для весов w_i и порога θ . В последующих рассуждениях мы опустим разницу между w и θ и рассмотрим сумму номеров весов и порога как N . Такого рода допущение упростит изложение и не окажет существенного влияния на результат.

Итак, мы должны разрешить систему линейных уравнений:

$$\underline{X}\omega = \underline{z}, \quad (3.7)$$

где размерность \underline{X} равна $P \times N$.

Заметим, что если $P=N$ и детерминант X не равен нулю, то есть $Det X \neq 0$, тогда

$$\underline{w} = \underline{X}^{-1} \underline{z}. \quad (3.8)$$

Если $P < N$, то из матрицы \underline{X} мы можем выразить субматрицу \underline{X}_F размерностью $P \times P$. Зададим $w_{P+1} = w_{P+2} = \dots = w_N = 0$ и, если $Det X_F \neq 0$, тогда:

$$\underline{w} = \underline{X}_F^{-1} \underline{z}. \quad (3.9)$$

Если мы не будем проводить явного разделения, то сможем получить большее количество решений, возможно даже бесконечное их количество, причем все они будут удовлетворять тем пределам, которые накладывают изображения, описанные ниже в примерах.

Однако более интересным случаем, является случай, когда $P > N$, тогда мы имеем:

$$\begin{matrix} \underline{X} & \underline{\omega} & = & \underline{z} \\ \text{(P \times N)} & \text{(N \times 1)} & & \text{(P \times 1)} \end{matrix} \quad (3.10)$$

Так как динамическая связь может генерировать бесконечное число ортогональных функций, мы можем «увеличить» количество переменных, то есть увеличить высоту столбцов в \underline{X} , увеличив N до N_{FL} , причем N_{FL} всегда может быть сделана либо равной либо больше, чем P . Таким образом, мы имеем:

$$\begin{matrix} \underline{X} & \underline{\omega} & = & \underline{z} \\ \text{(P \times N_{FL})} & \text{(N_{FL} \times 1)} & & \text{(P \times 1)} \end{matrix} \quad (3.11)$$

Причем, если $N_{FL} = P$ и $Det X_{FL} \neq 0$, тогда:

$$\omega_{FL} = \underline{X}_{FL}^{-1} \underline{z}. \quad (3.12)$$

Формула (3.11) является точным решением проблемы однослойной сети.

Если $N_{FL} > P$, и ряд $\underline{X}_{FL} = P$, тогда мы возвращаемся к формуле (3.9). Когда $P > N$, мы можем с определенным допущением получить формулу (3.13):

$$\underline{\omega} = \left(\underline{X}^t \underline{X} \right)^{-1} \underline{X}^t \underline{z}. \quad (3.13)$$

Здесь псевдоинверсия действительно дает наиболее подходящее решение. О том, что решение иногда может быть недоступно, сигнализирует высокое значение ошибки в конце процесса обучения.

Проведенный анализ показывает, что модель динамических связей всегда уступает полученному решению для однослойных сетей, если в процессе увеличения описания используется действительно большое количество добавленных ортогональных функций, а также используются методы сжатия данных для удаления излишних изображений.

3.2.3. Контролируемое обучение СДС

В данном пункте мы используем три примера для иллюстрации характеристик СДС в случае использования контролируемого обучения, которые были разработаны в [6]. В примерах описание изображений увеличиваются на основе модели тензора. Основной целью показа данных примеров является демонстрация преимуществ СДС.

Пример 1. Проблема четности по модулю 2.

В проблеме четности по модулю два, или проблеме XOR, изображения и соответствующие им выходы показаны в таблице 3.2.

Табл.3.2

Номер изображения	x_1	x_2	Соответствующий выход
1	-1	-1	1
2	-1	1	-1
3	1	-1	-1
4	1	1	1

Изображения могут быть размещены в двухмерном пространстве, как показано на рис.3.4 (а). Заметим, что изображения уравновешенной или одинаковой четности имеют соответствующее значение выхода, равное 1, и изображения случайной четности имеют соответствующий выход -1 . Соответствующие вход-выход пары обучаются вместе на основе подхода динамических связей, а также подхода правила дельты. Архитектуры нейронных сетей для СДС и для обычной многослойной НС обратного распространения сравниваются на рис. 3.4 (b) и 3.4 (c). Уровень уменьшения ошибки системы с увеличением разрядности описания для всех четырех изображений показан на рис. 3.5. Увеличение уровня обучаемости сети не соответствует ожидаемому.

Пример 2. Проблема четности по модулю 3.

Результаты, полученные для проблемы четности по модулю три схожи с результатами предыдущего примера. Данные результаты показаны на рис. 3.6 и 3.7. Заметим однако, что в данном случае, существенно обратить внимание на последние две операции. Если мы остановимся на первом шаге, то и сеть динамических связей и обычная многослойная сеть не смогут решить поставленные задачи. Точка перехода показана на рис. 3.8.

Пример 3. Обучение однопараметрической функции.

Задача состоит в обучении функции, которая зависит только от одной переменной. В данной задаче мы представим сеть с 20-ю точками на кривой и скажем, что сеть обучается парами вход-выход, а потом генерирует оценку исходной функции. По сути мы используем сеть для представления функции.

В этом случае используем модель динамических связей и разложим один компонент входного изображения в многомерном пространстве описанном через как $\sin \pi x$, $\cos \pi x$, $\sin 2\pi x$, $\cos 2\pi x$ и так далее, исключая сохраненные нами пределы, показанные на рис. 3.8 (а). Также могут быть установлены и другие пределы. Архитектуры однослойной СДС и многослойной НС обратного распространения сравниваются на рис. 3.9. Достигнутые при этом уровни обучения сетей показаны на рис. 3.10. Сравнение ошибки обучения для обеих сетей показаны на рис. 3.11.

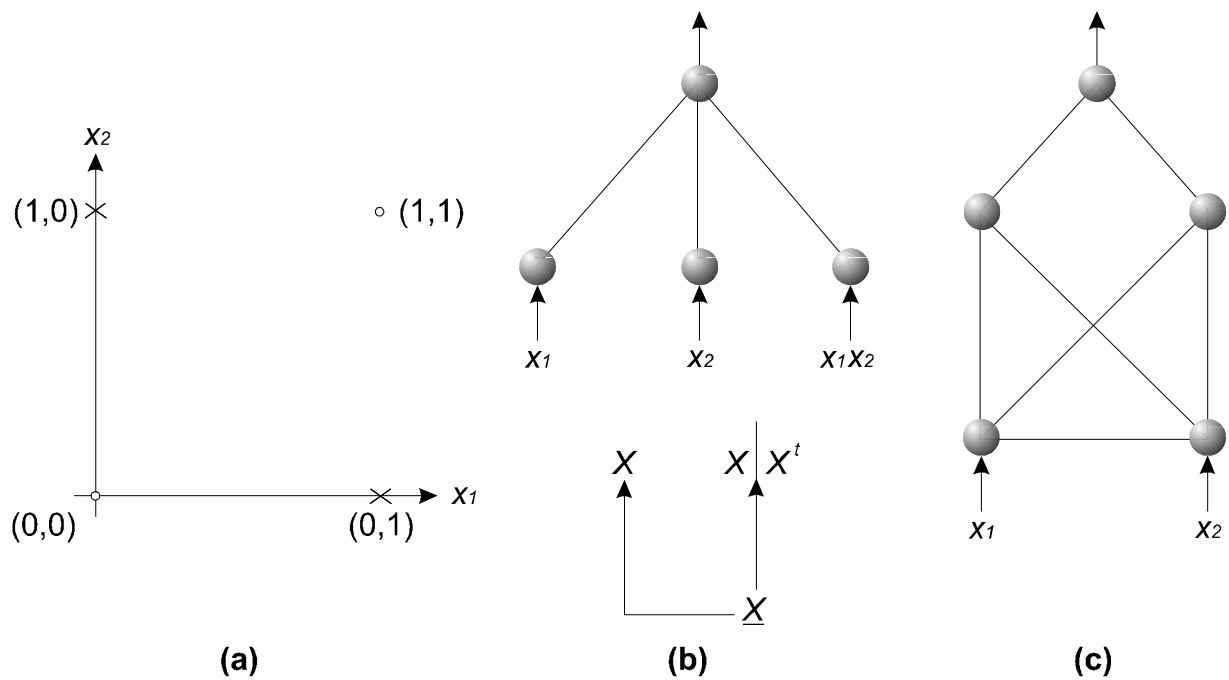


Рис. 3.4

Проблема четности по модулю 2. (а) Четыре классифицируемых изображения. (б) СДС без скрытых слоев. (с) Многослойная НС обратного распространения.

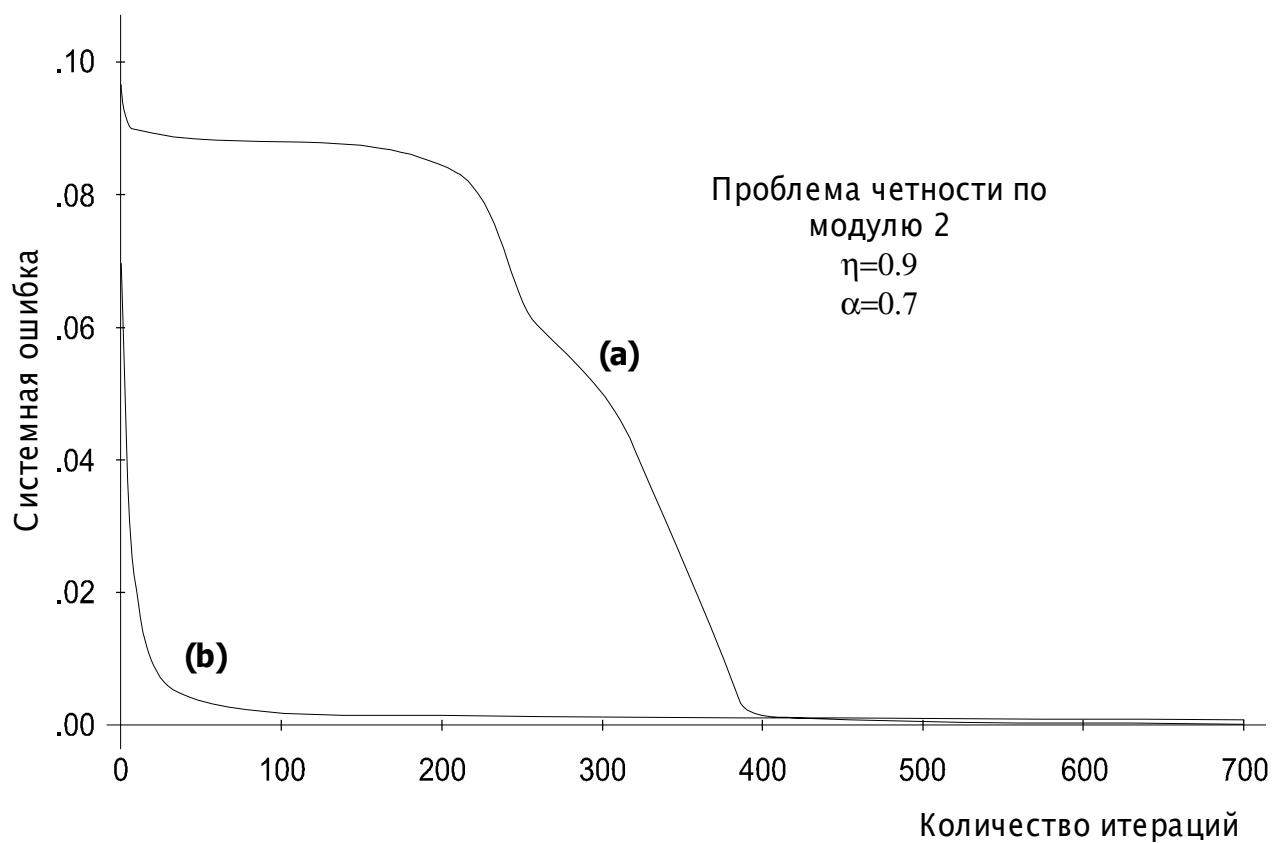


Рис. 3.5

Сравнение уровней обучения. (а) Многослойная НС с одним скрытым слоем. (б) СДС без скрытых слоев.

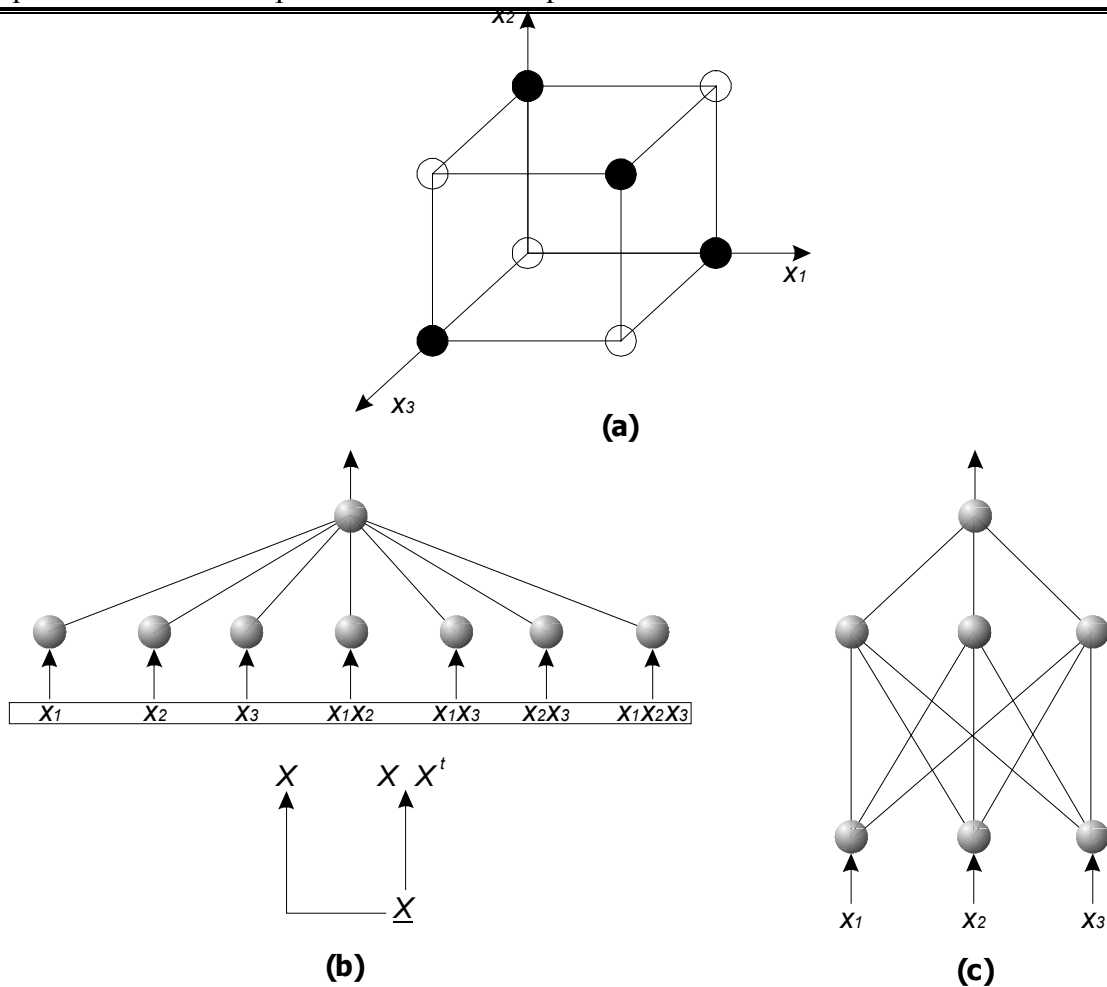


Рис. 3.6

Проблема четности по модулю 3. (a) Входные изображения. (b) СДС. (c) Многослойная НС.

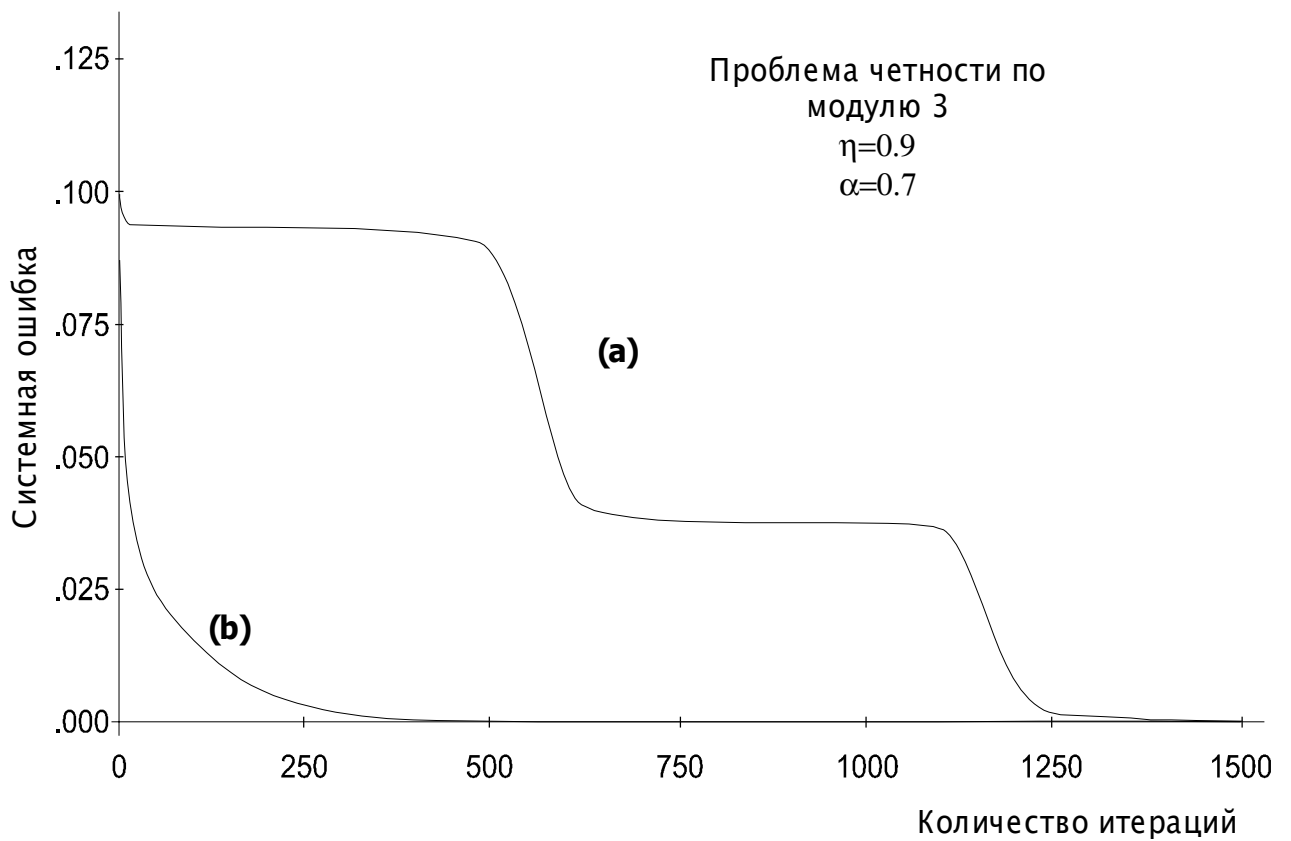


Рис. 3.7 Сравнение уровней обучения. (a) Многослойная НС. (b) СДС

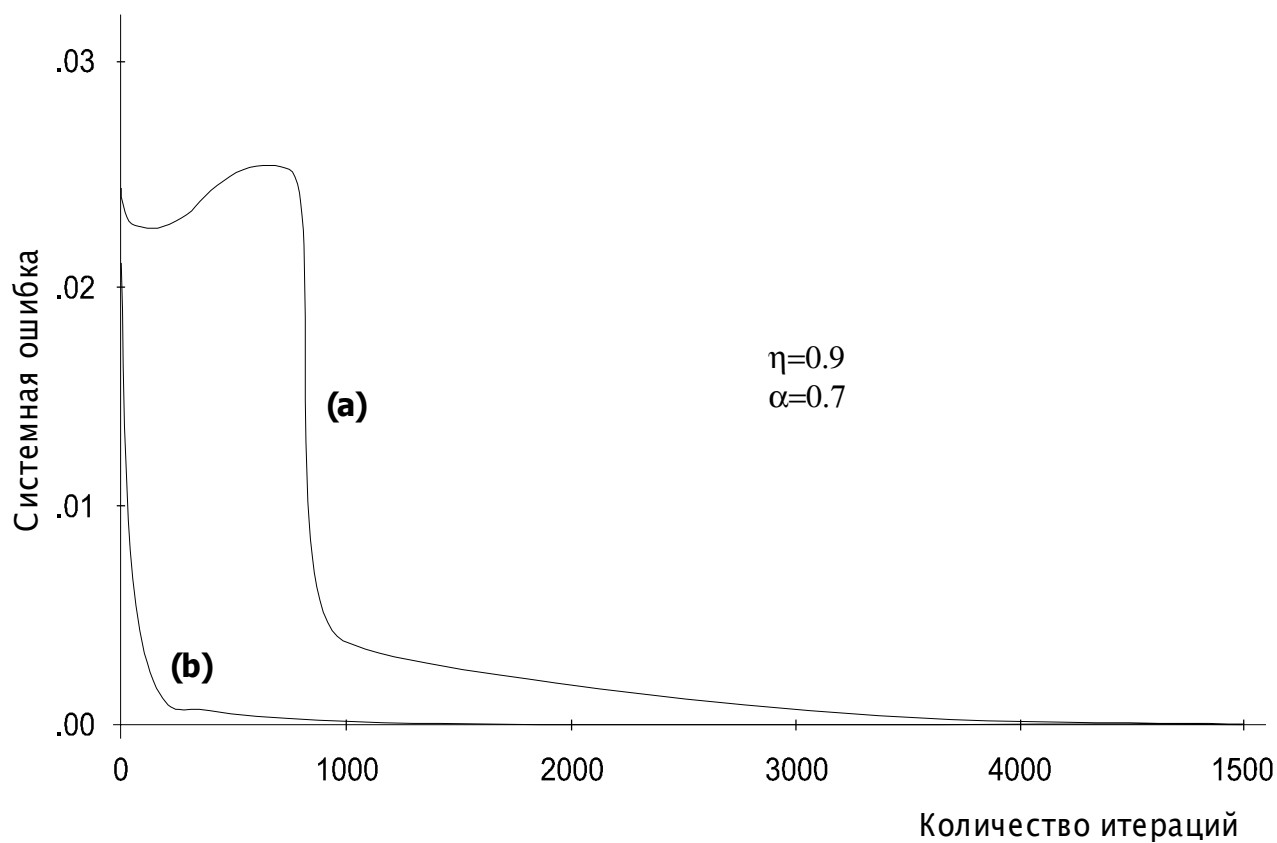


Рис. 3.8

Сравнение уровней обучения в сети, полученной через призму модели тензора.

(a) На основе одноступенчатой трансформации.

(b) На основе двухступенчатой трансформации.

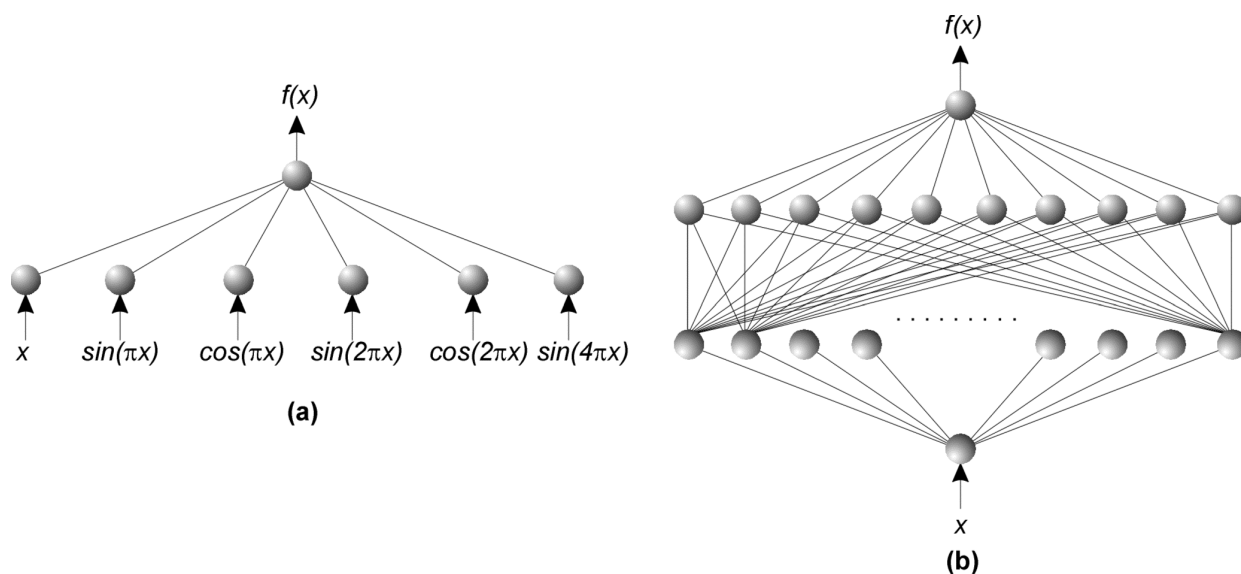
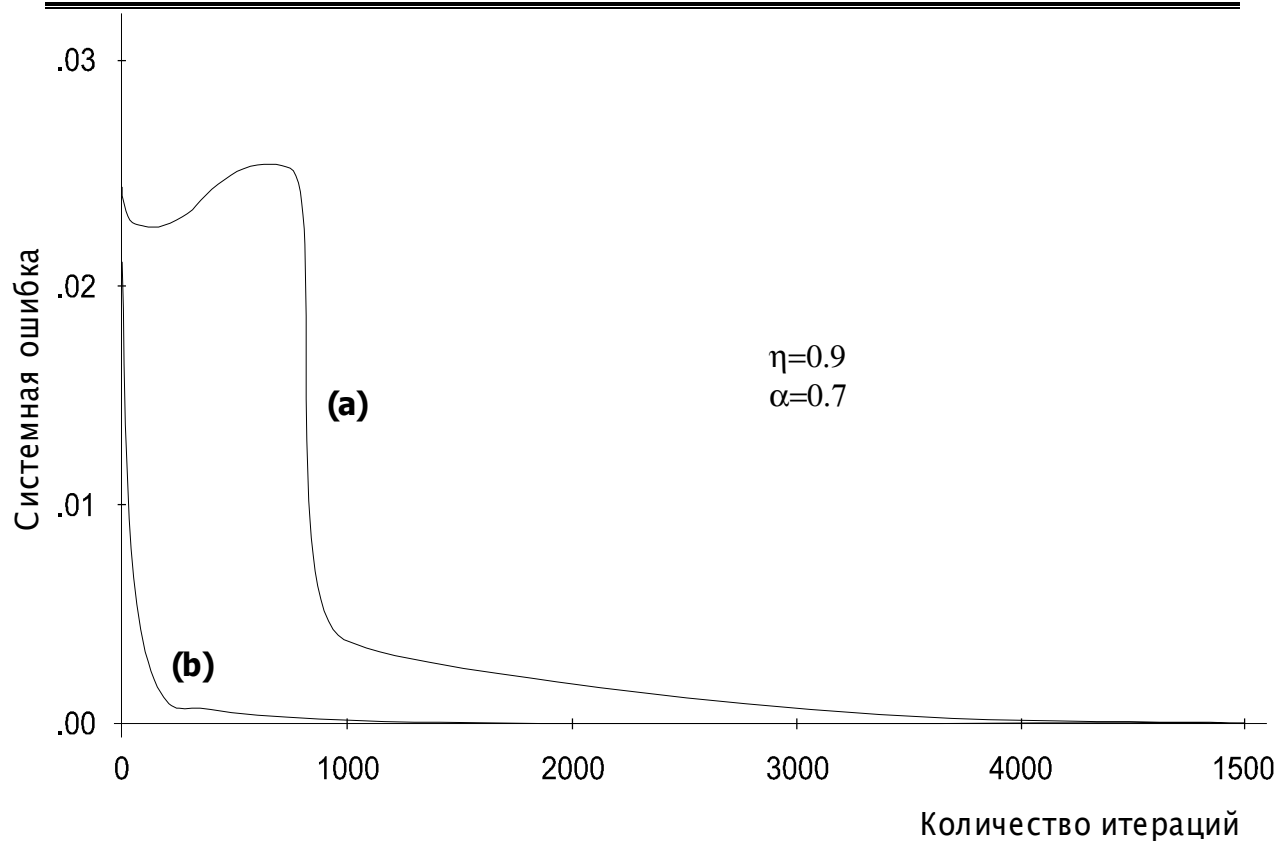


Рис. 3.9

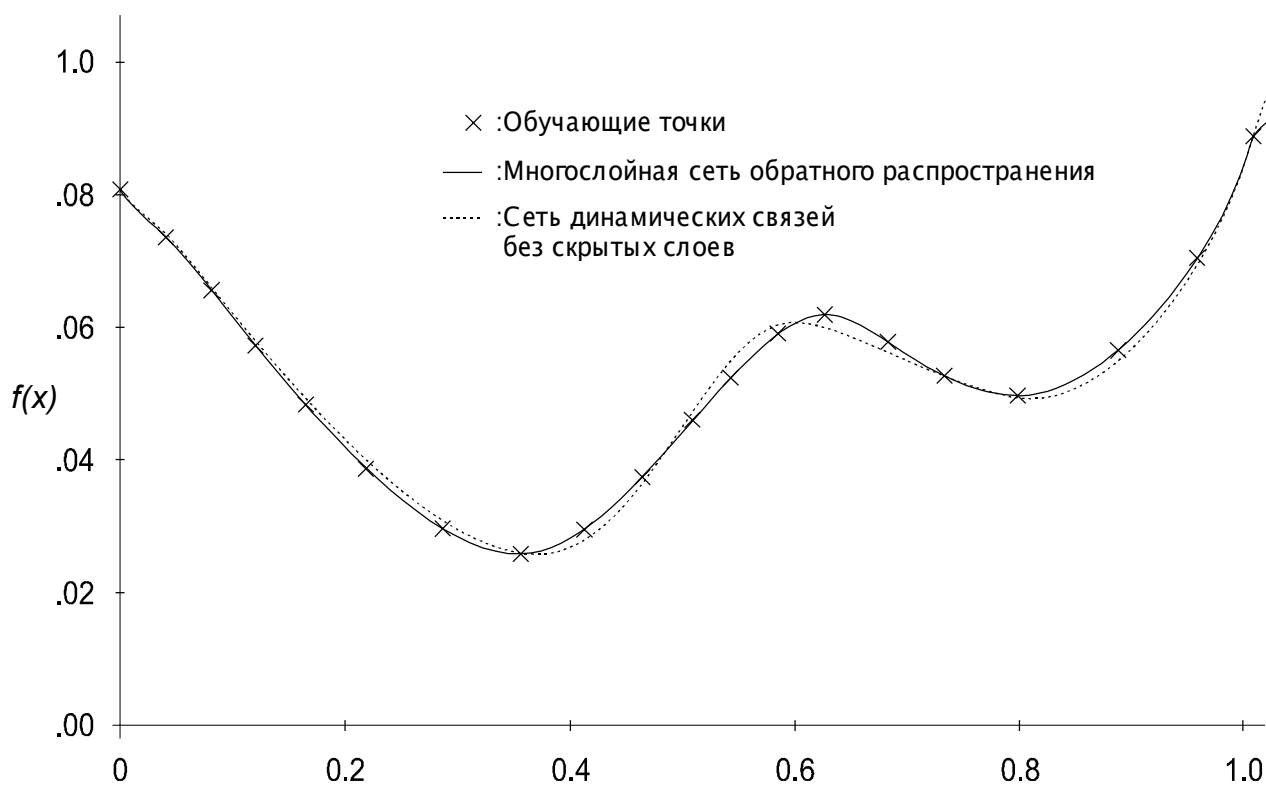
Сравнение архитектур сетей.

(a) СДС

(b) Многослойная НС обратного распространения.

**Рис. 3.10**

Сравнение уровней обучения. (а) Многослойная НС. (б) СДС

**Рис. 3.11**

Достигнутый уровень обучения для сетей прямого распространения

3.2.4. Комбинированное обучение СДС

Целью подхода динамических связей является создание существенно расширенного описания входных изображений, которое на основе связанных пар вход-выход могло бы быть обучено вместе с однослойной сетью, то есть с сетью, в которой отсутствуют скрытые слои.

Основным преимуществом использования однослойной сети является возможность, при ее определенной структуре, применить комбинацию контролируемого и неконтролируемого обучения, без перекодировки и изменения данных при переходе от одной сети к другой. Данная возможность является действительно существенной в реальной задаче распознавания изображений, а также других задачах классификации и категоризации, в которых приходится иметь дело с большими объемами данных.

Алгоритм обучения, применявшийся для сетей ART2, может быть использован для сортировки изображений в кластеры по их взаимному подобию. Рассмотрев каждый кластер в отдельности, мы можем опознать метки, уникальные для каждого кластера, и рассмотреть основные черты изображений, по которым они были классифицированы. После этого мы можем закончить расширение описания изображений на основе полученной информации об их особенностях, и рассчитать значение весовых коэффициентов. «Ассоциируемая» к соответствующим нейронам выходная модель (или модели), может использоваться в дальнейшем для организации обучения без учителя. Причем во время контролируемого обучения, могут быть рассмотрены только несколько характерных изображений из множества входных, так как все они однотипны. Остальные изображения могут быть на данный момент пропущены. Алгоритм комбинированного обучения представлен ниже.

Этап неконтролируемого обучения

1. Инициализация всех выходных нейронов $j, j=1, 2, \dots, J$
2. Инициализация весов $b_{ji} = \varepsilon_{ji}$, где ε_{ji} случайные числа в диапазоне $(-1 < \varepsilon_{ji} < 1)$
3. Подача входного изображения $\{y_i\}, i=1, \dots, N$
4. Расчет квадрата Евклидова расстояния: $ED_{ji}^2 = \sum_i (b_{ji} - y_i)^2$
5. Установка значения j , для которого $ED_{ji}^2 < ED_{ki}^2$ для всех $k = 1, 2, \dots, J, k \neq j$
6. Определение изображения $\{y_i\}$, принадлежащего нейрону j , для которого ED_{ji}^2 не больше, чем некоторый предел значения ED_{ji}^2 , который выбирается таким, что изображения, для которых этот предел превышен, в данный кластер не входят
7. Обновление значения $b_{ij}(n+1) = \frac{n}{n+1} b_{ij}(n) + \frac{1}{n+1} y_i$, (на стадии инициализации $n=0$), то есть после инициализации $b_{ij}(1) = y$
8. Подача следующего изображения, установка его принадлежности соответствующему нейрону, и обновление значения $\{b_{ij}\}$ для этого нейрона.

Этап контролируемого обучения

1. *Классификация*: все изображения, принадлежащие одному нейрону j в кластере, должны определить значения соответствующих выходов $\{t_k\}$. Это верно для однослойной сети, в которой определены выходы, «принадлежащие» кластерному нейрону j .

2. *Оценка значений признаков*: все изображения, принадлежащие одному нейрону j в кластере, на самом деле имеют некоторые различия в признаках $\{t_k\}$. Это также верно для однослойной сети, в которой определены выходы, «принадлежащие» кластерному нейрону j .

Еще одна проблема может возникнуть в случае, когда изображения не имеют меток, по которым их можно классифицировать. Это не должно удивлять, так как входные изображения совсем не обязательно должны иметь настолько характерные особенности, на базе которых можно установить метки и провести классификацию.

Данный случай, сосуществования вместе контролируемого и неконтролируемого типов обучения, может организовать сеть таким образом, чтобы она смогла решить задачу ассоциации большого количества различных изображений с соответствующим выходом. Часто, если две группы, по каким-либо причинам, не попадают в соответствующий кластер в результате классификации, результатом является сообщение о том, что описание изображения не является адекватным. В этом случае мы должны признать стохастический характер возникновения такого рода случаев.

3.2.5. *Хранение ассоциативной информации*

В данном пункте мы опишем методы использования СДС для хранения ассоциативной информации. В основном для подобной задачи применяется сеть Хопфилда, но, однако бывают случаи, когда ее применение не является эффективным.

Задача запоминания решается через контролируемое обучение. Изображение запоминается, если для нее задан соответствующий выход сети. Таким образом, пусть для примера $t_k=1$. При этом мы совершенно не обязательно должны устанавливать значение желаемого выхода $=1$, мы можем установить его равным и 0.8.

Итак, если в сети слишком искажена информация, то выход не станет равным 0.8. Он может быть равен 0.9 или 0.65, и тогда значения ошибок $(t_k - o_k)$ будут находиться в промежутке от -0.10 до $+0.15$. При этом не важно, где именно значение ошибки положительно или отрицательно, входные биты нерасширенного изображения будут изменяться асинхронно пока значение ошибки не станет равным нулю или близким к нулю и не будет в дальнейшем изменяться. В этом случае одно из исходных изображений должно быть восстановлено.

Заметим, что изменение одного или нескольких компонентов входного изображения будет влиять на значения некоторых компонентов расширенного его описания. Решая, сделать или нет соответствующее изменение, мы должны рассчитать частные суммы всех взвешенных компонентов, чтобы уменьшения ошибки. Если решение будет принято положительно, то данное изменение производится и запоминается. Данная процедура проиллюстрирована на рис. 3.12, 3.13, 3.14.

В данном примере рассматривается ситуация запоминания букв T и C в памяти разрядностью 3×3 . Как показано на рис. 3.12, сеть обучается на основе расширенного изображения, состоящего из 9 пикселей и активационных связей между пикселем и его четырьмя ближайшими соседями. Стадия обучения пройдена. На рис. 3.13 представлена однослойная НС, обученная методами контролируемого обучения, с сохраненными в памяти буквами T и C . В каждом случае, искажение изображения достигается изменением не более одного бита, на основе которого и будет возникать ошибка.

Ошибки, так же как и изменения в их значениях, показаны для различных методов синтеза помех, достигаемых с помощью асинхронной перестановки битов. Таким образом, для данного идеализированного случая, минимальное значение ошибки будет достигнуто, когда одно из исходно верных изображений будет восстановлено. В случае

T , разностная ошибка будет минимальной для коррекции, сделанной в 5-м бите. Подобный результат достигнут для C , только в этом случае достигнутое значение ошибки было равным нулю.

Подобные же результаты продемонстрированы на рис. 3.14, но в этом случае помехи в изображениях достигались за счет изменения 2-х битов.

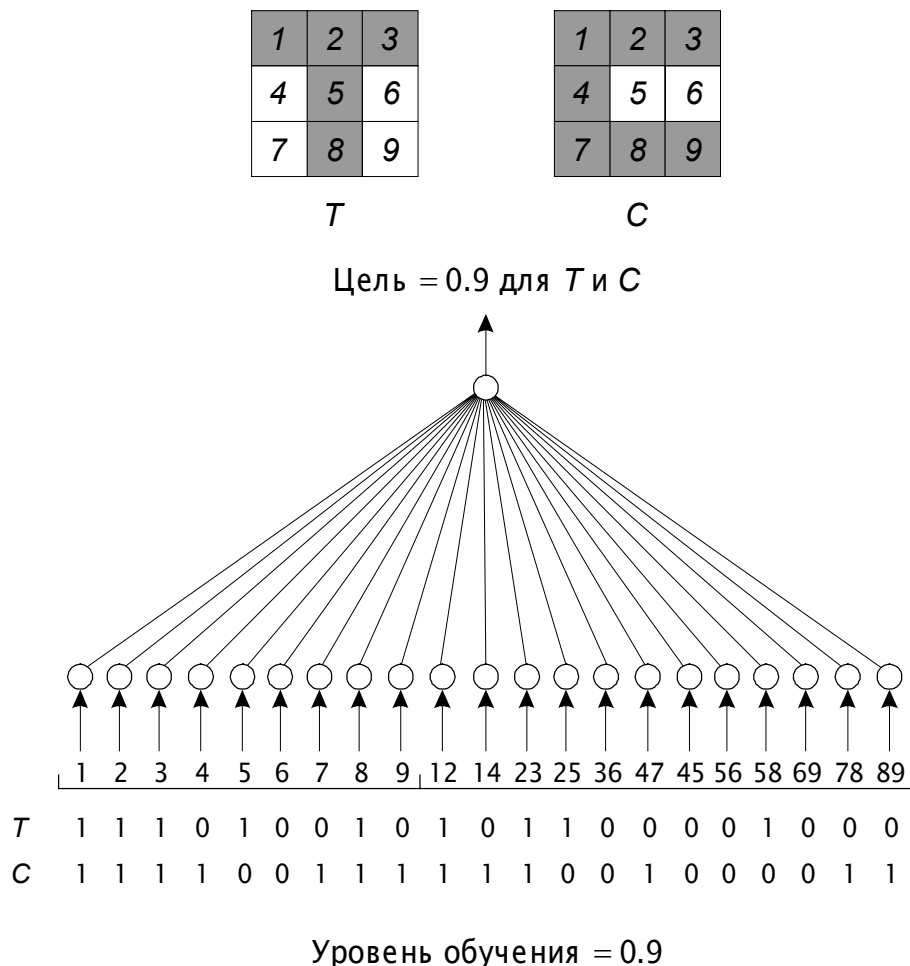


Рис. 3.12

Полученные результаты являются в некотором смысле обособленными, и пока не известно, каким образом можно было бы на их основе сделать обобщенные выводы. То есть выводы о том, сколько действительно нейронов было использовано для решения данной задачи, что помогло бы нам провести дополнительную оптимизацию сети.

Одноразрядная помеха в изображении *T* Одноразрядная помеха в изображении *C*

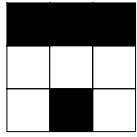



















	$E=0.032401$			$E=0.001213$
Опытные изображения		Инвертированные биты	Опытные изображения	
	$E=0.024884$	1		$E=0.002474$
	$E=0.008001$	2		$E=0.001223$
	$E=0.058000$	3		$E=0.000415$
	$E=0.032704$	4		$E=0.002264$
	$E=0.001590$	5		$E=0.000000$
	$E=0.015772$	6		$E=0.000619$
	$E=0.002764$	7		$E=0.005791$
	$E=0.061001$	8		$E=0.020387$
	$E=0.005700$	9		$E=0.000010$

Рис. 3.13

Использование СДС для организации ассоциативной памяти

Двухбитовая помеха в изображении T

Двухбитовая помеха в изображении С

Опытные изображения	Инвертированные биты	Опытные изображения		
	$E=0.00070787$	1		$E=0.00398986$
	$E=0.00037809$	2		$E=0.00636622$
⋮			⋮	
	$E=0.00187470$	9		$E=0.04481664$
	$E=0.00018003$	1,2		$E=0.00204523$
	$E=0.00175566$	1,3		$E=0.00044302$
⋮			⋮	
	$E=0.00000000$	3,7		$E=0.00000000$
⋮			⋮	
	$E=0.00574156$	8,9		$E=0.08319908$

Рис. 3.14

Получение из памяти корректной формы изображения по его зашумленной репрезентации (на основе одно- и двухбитовой помех)

3.3. Проблема адаптации распознавания объектов

В данном параграфе мы рассмотрим методы адаптивного или, другими словами, устойчивого распознавания объектов, а также некоторые примеры их применения на практике. Следует уделить повышенное внимание данной теме, так как этот вопрос является одним из наиболее трудоемких вопросов приближения искусственного интеллекта к реальному за счет увеличения инвариантности системы распознавания к положению объектов в сцене, а также их ориентации. Слово «адаптивный», применительно к теории распознавания объектов, применяется для подчеркивания важности получения возможности оперировать с изображениями, испорченными шумовыми помехами и проводить классификацию объектов в условиях отличных от идеальных. С одной стороны, не существует прямой связи между адаптивным распознаванием изображений и параллельной обработкой информации. Например, все нейронные алгоритмы – как, например ассоциативная память Хопфилда, многослойная НС обратного распространения, сети ART и СДС – могут быть применены на основе обычной цифровой логики, как обычно они и применяются. С другой стороны, концепции нейронных сетей являются наиболее релевантными в данной задаче, так как изображения могут быть большими и иметь детализацию очень высокой степени, и параллельная обработка на основе большого числа присоединенных друг к другу процессоров может существенно повлиять на общую скорость обработки изображения, а также сделать ее более адаптивной. Таким образом, можно сделать вывод, что процессы и алгоритмы, используемые в теории НС, можно эффективно применять в вопросе адаптации.

3.3.1. Распознавание структуры внутри изображения

В области геометрического распознавания объектов, входные изображения представляются как вектора, то есть кодируются в последовательность реальных чисел. Последовательности имеют фиксированную длину (несмотря на некоторую разницу во входных изображениях, которые обычно пропускаются через некоторую маску), а также имена, неявно заданные на смысловой основе компонентов изображения. В данной ситуации структурные соотношения между изображением и векторами уже не являются столь важными и обычно утрачиваются.

Например, задача состоит в том, чтобы распознать присутствие прямоугольника в поле зрения независимо от его позиции, ориентации и размеров. Или, например, следует распознать рукописные буквы, также независимо от их местонахождения и размеров в изображении.

В прошлом задачи подобного рода рассматривались в методах распознавания изображений, называемых *синтаксическим распознаванием*. Данный подход базируется на математической теории распознавания букв, идея которой состоит в том, что мы должны думать об изображении в терминах предложений или буквенных строк. Строки могут быть различной длины, и задача классификации состоит в ответе на вопрос: будет ли являться решением или нет, что предложение принадлежит заданному языку, то есть, подчинено ли оно грамматическим правилам принятого языка. Таким образом, процедура распознавания перетрансформируется в задачу определения, где именно должна находиться та, или иная, буква в предложении в соответствии с грамматикой языка. Следовательно, синтаксический подход имеет ограниченный успех в вопросах распознавания структур в изображении. Только постоянные или печатные буквы могут быть распознаны на основе этого подхода.

Подход к данной задаче, основанный на НС, был собран на основе многих других подходов к распознаванию структур в изображениях, инвариантных к позиции и ориентации этих структур.

3.3.1.1. Инвариантность к смещению

Рассмотрим одномерное бинарное изображение размером 16 пикселей, показанное в виде двухмерного массива 4×4 на рис. 3.15. Задача состоит в распознании двух темных связанных пикселей (имеющих значение, равное 1), независимо от их расположения в массиве. Остальные пиксели имеют значение, равное 0. То есть, следует определить наличие горизонтального двухпиксельного массива, независимо от его местоположения в общем массиве.

При этом следует рассмотреть важную задачу нахождения граничных условий, на которую мы обратим внимание чуть позже в этом параграфе. Для удобства, мы возьмем периодический характер граничных условий. Это означает, что мы можем сосредоточить наше внимание на ячейке в 16 пикселей, которое и будет нашим *наблюдаемым пространством*. Все рассматриваемые ячейки являются идентичными, и, рассмотрев только одну из них, мы можем повторить все те же операции и для остальных случаев. В данной модели, что было утеряно с дисплея или рассматриваемого пространства справа, то и появится в левой его части. Этот момент и предусматривает термин периодических граничных условий.

Шестнадцать одномерных массивов, представленных на рис. 3.15 (b), показывают все возможные состояния, в которых два темных пикселя могут находиться в наблюдаемом пространстве. Например, если наблюдаемые пиксели выходят за границу изображения с одной стороны, то они будут продолжаться с другой.

Итак, задача упрощается до минимальной – распознать двухпиксельный массив, на основе заданной специфической особенности, независимо от их перемещения, которую можно рассмотреть как задачу синтеза сети, которая сможет распознать все 16 изображений, изображенных на рис. 3.15 (b), и выдать на выходе 1, например.

Рассмотрим для этой задачи СДС в терминах однослойной сети, в которой отсутствуют скрытые слои. Настройка весовых коэффициентов, для транслирования изображений, в таком случае, должна соответствовать равенствам, приведенным на рис. 3.16. Сумма равенств также приведена на данном рисунке.

При обучении сети установка значений весов происходит по обычной процедуре, то есть сети последовательно представляются все 16 изображений, и желаемый выходной сигнал, в данном случае, равный 1. Однако в этом простом случае мы видим, что процесс строится не совсем стандартно.

После суммирования 16 формул, в результате мы имеем только одну формулу для всех 16 весов:

$$\left[\sum_g (x_1)_g \right] W_1 + \left[\sum_g (x_2)_g \right] W_2 + \dots + \left[\sum_g (x_{16})_g \right] W_{16} = 16. \quad (3.14)$$

Любое исходное одномерное 16-типиксельное изображение подвергнуто всем возможным неповторяющимся горизонтальным перемещениям, то есть все возможные позиции двух связанных пикселей в нем учтены. Значение каждого пикселя и сумма этих значений во всех конфигурациях вычисляются. И, таким образом, данная сумма подается на вход нейрона СДС, состоящей из 16 нейронов.

Формула (3.14) является условием для определения состояний синаптических весов для всех 16-ти входов сети. Заметим, что все они одинаковы по значению. Таким образом, можно вывести, что $W_i = W$ для всех i , следовательно:

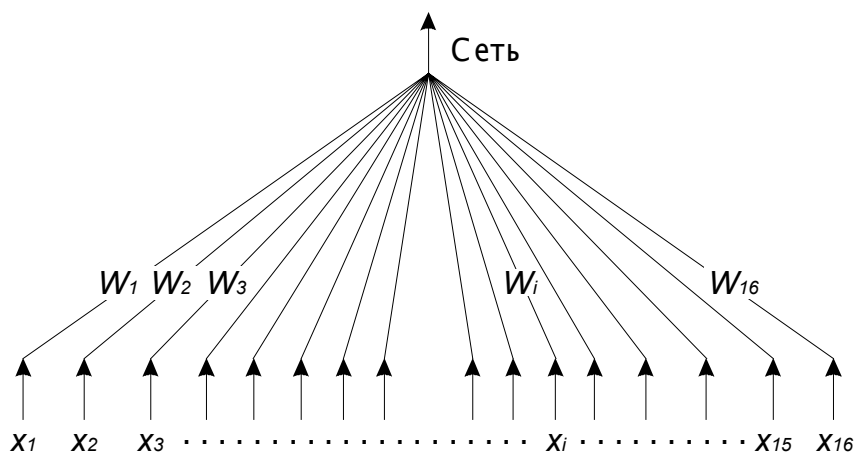
0	0	0	0
0	1	1	0
0	0	0	0
0	0	0	0

(a)

Номер изображения	Изображение															Выходное значение
1	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	1
2	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1
3	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1
⋮																⋮
15	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	1
16	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	1

(b)

Рис. 3.15



Номер изображения	Формулы для расчета весов
1	$(0)W_1+(0)W_2+ \dots +(0)W_5+1W_6+1W_7+(0)W_8+(0)W_9+ \dots +(0)W_{16}=1$
2	$(0)W_1+(0)W_2+ \dots +(0)W_5+(0)W_6+1W_7+1W_8+(0)W_9+ \dots +(0)W_{16}=1$
15	$(0)W_1+(0)W_2+ \dots +1W_4+1W_5+(0)W_6+(0)W_7+ \dots + (0)W_{16}=1$
16	$(0)W_1+(0)W_2+ \dots +(0)W_4+1W_5+1W_6+(0)W_7+ \dots + (0)W_{16}=1$
Сумма	$2 W_1+ 2 W_2+ \dots + 2 W_4+ 2 W_5+ 2 W_6+ 2 W_7+ \dots + 2 W_{16}=16$

Рис. 3.16

$$W \sum_i \left[\sum_g (x_i)_g \right] = 16. \quad (3.15)$$

То есть мы видим, что на самом деле в наличии имеется сеть только с одним нейроном во входном слое и одним весом.

Для проблемы, проиллюстрированной на рис. 3.15, данный весовой коэффициент получается равным:

$$W = 0.5. \quad (3.16)$$

Результат суммирования формулы (3.15) можно изменять. Таким образом, во время обучения сети мы оцениваем каждый пиксель во всех его возможных состояниях и, впоследствии, рассчитываем сумму всех пикселей. То есть, во время обработки сложных и специфических изображений в пространстве наблюдения мы должны просуммировать по всем пикселям, и, при этом, принятым критерием будет:

$$W \sum_i (x_i) = 1, \quad (3.17)$$

как было предусмотрено в начале и проиллюстрировано на рис. 3.16. Основным моментом здесь является то, что сумма всех пикселей инвариантна к перемещению, то есть не нужно повторять вычисления для всех возможных перемещений.

Казалось бы, что распознавание последовательности связанного массива двух темных пикселей было слишком простой задачей. Все, что нужно было сделать, это рассчитать сумму значений пикселей в наблюдаемом пространстве. Однако, мы поторопились утверждать, что данная процедура даст необходимый результат, так как последовательность темных пикселей может быть и не связанной. В конце концов, последовательность любых двух темных пикселей может дать подобный результат, но с определенным допущением. Поэтому примем за ограничение то, что распознаваемые пиксели должны быть связаны.

Соседности пикселей, продемонстрированные на рис. 3.15 и 3.16 являются необходимым, но недостаточным условием для распознавания более расширенного класса геометрических фигур.

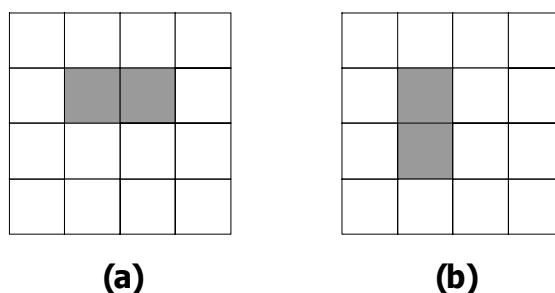


Рис. 3.17

Рассмотрим ситуацию, когда необходимо найти различие между горизонтальным массивом на рис. 3.17(a) и вертикальным массивом на рис. 3.17(b). В этом случае попробуем принять, что в описании изображения используется пространственный массив. Например, понятие x_i , $i=1,2,3...16$ не несет никакой информации относительно связности двух пикселей, которая может быть в одном случае горизонтальной, а в другом вертикальной.

Для получения данной информации используем корреляционные элементы $x_i x_{i+j}$, $i=1,2,...,16$, $j=1,2,3,4$ и так далее. Заметим, что толчком к использованию расширенных понятий в этом случае является требование получения описания геометрических фигур, а не просто расширение описания, как в случае использования СДС. Кроме того, характер понятия $x_i x_{i+j}$ для определенного значения i должно быть усреднено для всех возможных

трансформаций, если мы хотим иметь инвариантную ко всем горизонтальным трансформациям оценку. Приняв, что переносы удовлетворяют принятым граничным условиям, обнаруживается, что мы можем переопределить все 16 операций суммирования в одну операцию. Расширенный элемент теперь находится для каждого значения j . Набор таких элементов определяет новый тип изображения, которой теперь есть возможность определения разницы между двумя геометрическими структурами.

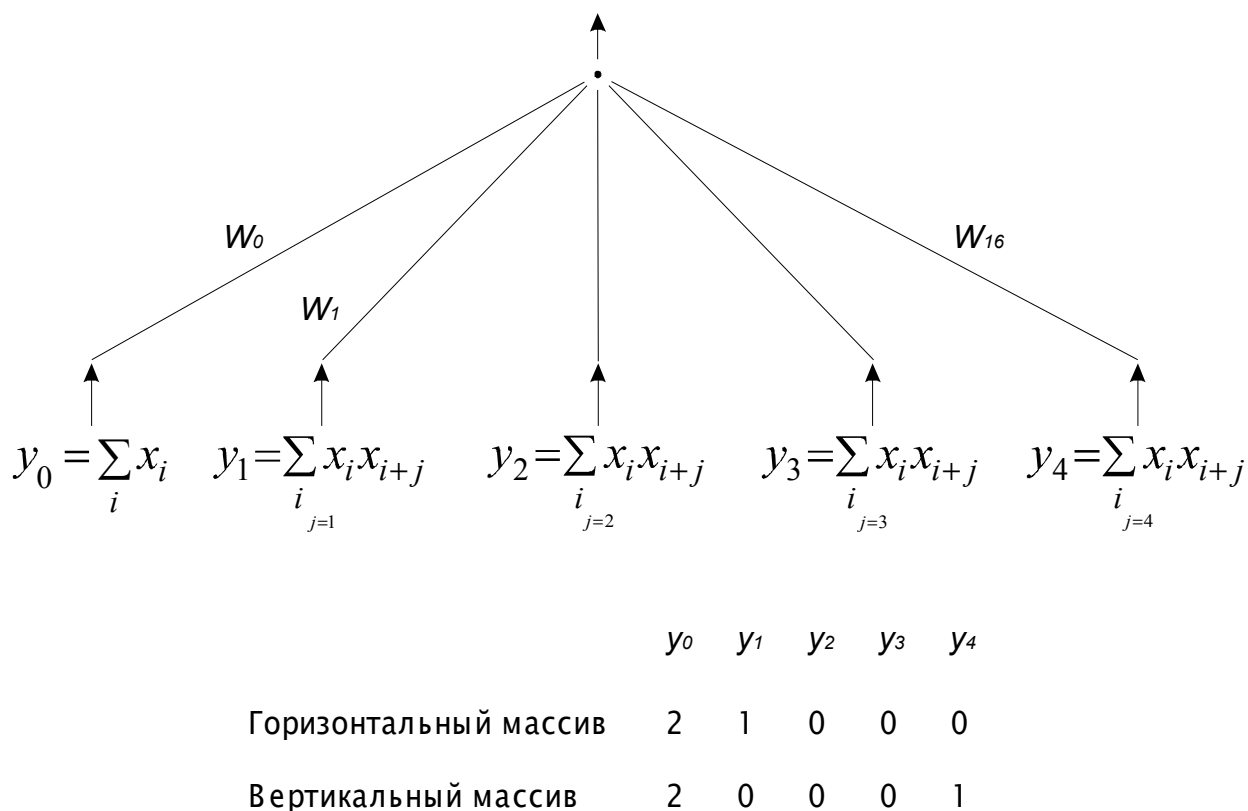


Рис. 3.18

Таким же образом новая сеть может быть синтезирована для решения задачи нахождения различия между двумя новыми изображениями. В случае надобности расширенное описание изображения на основе динамических связей также может быть использовано для данной сети. Эти вопросы рассмотрены на рис. 3.18, где мы можем видеть, что пять компонентов новых изображений, полученных после предварительной обработки, представляют собой $\sum_i x_i$, $\sum_i x_i x_{i+1}$, $\sum_i x_i x_{i+2}$, $\sum_i x_i x_{i+3}$ и $\sum_i x_i x_{i+4}$. В этом простом примере не обязательно использовать расширение описания.

Как показано на рис. 3.18, оба новых изображения включают в себя все пять компонентов, и задачей является определение, могут ли эти изображения найти разницу между двумя структурами. Идея экономичности представления информации сильно контрастирует с полученной ситуацией, когда мы имеем два изображения вместо одного, каждое по $16 \times 5 = 80$ компонентов. Но это и есть та самая цена, которую мы платим за увеличение адаптивности в распознавании.

Выводы, полученные из данной главы, говорят о том, что распознавание структур в изображениях проводится с применением методов, которые конкретно отличаются от ранее рассмотренных стандартных подходов.

3.3.1.2. Граничные условия

В задаче установки инвариантности к перемещениям принятие периодических граничных условий ведет к значительному упрощению проблемы. Нам более-менее известны свойства перемещаемой группы на периодической сетке изображения. Фактически принятая сетка, это бесконечный массив копий оригинальных ячеек изображения, это означает, что группа бесконечных перемещений описывается конечной группой отдельных операторов, и, кроме того, нашей задачей станет только сосредоточение своего наблюдения над одной элементарной ячейкой. Так как то, что произойдет с одной ячейкой, немедленно отзовется на состоянии остальных.

Рассмотрим вопрос инвариантности к перемещениям более формально и углубленно для случая периодических граничных условий и наблюдения за одной ячейкой. Это будет некоторой подготовкой к рассмотрению случая непериодических границ, так мы сможем провести границу между последующими процедурами.

Пусть $G \equiv \{g\}$ - конечная группа трансформаций, таким образом:

1. Для всех g в G и для всех изображений \underline{x} в рассматриваемой области $g\underline{x} = \underline{y}$ также принадлежит данной области.
2. G замкнута согласно 3.18:

$$g_1, g_2 \in G \Rightarrow g_1 g_2 \in G. \quad (3.18)$$

3. Там, где инверсная трансформация g^{-1} для:

$$g \in G, \quad (3.19)$$

верно:

$$g^{-1} \in G, \quad (3.20)$$

и

$$g g^{-1} \underline{x} = \underline{x}, \quad (3.21)$$

или равенство:

$$I \underline{x} = \underline{x}. \quad (3.22)$$

Набор изображений $\{\underline{x}^s\}$ эквивалентен классу G если

$$\{\underline{x}^s\} = \{g \underline{x}^s \mid g \in G\}. \quad (3.23)$$

Наблюдение Y является G -инвариантным, если последняя будет давать один и тот же выходной сигнал для всех членов равенства \underline{x}^s . Таким образом, $Y(\underline{x})$ G -инвариантен $\Leftrightarrow Y(\underline{x}^s) = Y(g \underline{x}^s)$ для всех $g \in G$.

Зададим оператор инвариантности:

$$I_G = \sum_G g. \quad (3.24)$$

Следовательно

$$g I_G \equiv I_G. \quad (3.25)$$

Для любого оператора наблюдения O и любого изображения \underline{x} можно записать следующее равенство:

$$O(I_G \underline{x}) = O(g I_G \underline{x}). \quad (3.26)$$

Откуда видно, что инвариантность оператора наблюдения может быть получена, если операции, проведенные над изображением $I_G \underline{x}$, будут более предпочтительными, чем операции непосредственно над \underline{x} .

В обработке с помощью НС мы достигаем инвариантность к трансформациям в распознавании объектов с помощью усреднения значения входа каждого нейрона сети через соответствующую группу перемещений, для получения возможности обнаружения особенностей, несовместимых с существующими вариантами. В случае, когда периодические граничные условия не достижимы, методы обработки должны быть изменены, однако это можно сделать, на основе старых методов, которые все еще могут быть полезны.

Для более реального представления, рассмотрим задачу распознавания наличия некоторой локальной структуры в изображении (то есть в секторе наблюдения), независимо от его местоположения в этом изображении. Один подход, это разработать некоторое количество операций распознавания локальной структуры, результат применения которых будет передан на входы сети. При этом неконтролируемое обучение НС может быть использовано для сортировки различных видов, в которых можно распознать данную структуру. В более реальных режимах, контролируемое обучение может увеличить точность и достоверность распознавания.

Независимо от выбранного способа распознавания все операторы инвариантности к перемещениям имеют схожие характеристики. Обычно они строятся на операции установки одного пиксела в центр дисплея и повторения этой операции для всех пикселов (или почти всех) в области наблюдения. Полученные локальные результаты суммируются, или усредняются, для подачи на один вход нейронной сети. Оператор в целом может быть рассмотрен как оператор действия над полными описаниями изображений, создающий концепцию того, что результаты выполнения операций распознавания дадут нужный результат независимо от расположения локальной структуры в изображении.

Как показано в примере, проиллюстрированном на рис. 3.15, адаптивность может быть достигнута через создание процедуры перечисления различных конфигураций, в которых локальная структура может находиться, не забывая об оценке центральности пикселов для каждой конфигурации и усреднения результатов по приведенным оценкам для каждого пиксела и каждой конфигурации. Финальным шагом процедуры получения особенностей может измениться по характеру в соответствии с усредненными результатами. Задачей при этом является замена суммы всех конфигураций суммой над всеми пикселями, или почти всеми пикселями. В этом случае граничные условия покажут, что наборы пикселов должны быть включены в специальный глобальный оператор.

В рассмотренных методах нахождения особенностей геометрических структур, инвариантных к перемещениям, подчеркивается, что не существует взаимозависимостей на периодических граничных условиях, и мы имеем дело с наборами возможных позиций предпочтительнее, чем с группами трансформаций.

3.3.2. Распознавание букв в изображении

Очень часто методы ИИ применяются для распознаваний лингвистических символов. Сильной стороной первых языков программирования ИИ, таких как LISP или PROLOG, было то, что в них символы брались из слов натуральных языков, обычно английского. На основе этого подхода программист может работать в пространстве, известного ему, человеческого языка, который он использует каждый день в обычной речи. Принятые

обозначения и цели утверждений, правила и структуры данных, таким образом, понимаются просто, на интуитивной основе. Цена, которую мы платим за эту выгоду такова, что лингвистические символы должны быть корректно переадресованы в формат, который может понять ЭВМ.

Например, в LISP имеется возможность перекодировки обычных букв из собранного листа в корректно адресуемый вид. Эта перекодировка получается через перебор букв, но, к сожалению, она не может быть произведена через стандартные методы распознавания изображений на основе параллельной обработки. Расположение одной записи в листе может помочь нам найти все остальные экземпляры.

Таим же образом в распознавании букв, во время запуска оператора IF-THEN, IF-часть должна быть проверена через соответствие изображений до того как THEN-часть будет активизирована. В PROLOG, для унификации, мы должны проделать большую работу по нахождению соответствий строк лингвистических символов, для связывания значений соответствующих переменных. Ясно, что настоящая высокоскоростная ассоциативная память, основанная на соответствиях изображений, может эффективно и быстро решать описанные задачи.

Однако уже совсем скоро появятся первые эффективные станции параллельной обработки информации. Ученые высказывают реальный интерес в вопросах представления лингвистических символов в структурах данных. Однако при этом существуют некоторые проблемы, как, например то, что «локальное» представление может не подходить для параллельной обработки, и некоторые ученые думают что, более распределенная обработка в целом может быть более грубой по сравнению с разбиением изображения на несколько малых поднаборов элементов или связей между ними, используемых в описании буквенных изображений, и параллельной обработки результатов разложения.

Одна из подобных схем обработки, называемая «грубой кодировкой», была предложена Хинтоном [27], рассмотрим эту схему более подробно.

Основой метода является «размазывание» описания символа на большем количестве ячеек памяти, чем требуется для его хранения первоначально. Обычно рабочая область памяти может содержать 2000 бинарных ячеек. Каждая ячейка содержит таблицу в формате, показанном в табл. 3.3. Каждое поле этой таблицы задано шестью символами, каждый из которых имеет три столбца, итого 6^3 или 216 троек на поле. Ячейка, описанная в табл.3.3, имеет тройки (С К R) и (F А В) в своем поле наряду с 214 остальными.

Мы можем запоминать эти тройки с рабочей области памяти путем изменения значений всех рецепторов, описывающих данную тройку. На 2000 элементарных ячейках каждая тройка занимает в среднем $(6/25)^3 \times 2000$ рецепторов, или округленно 28. Конечно, это значение приведено неточно, и реальное значение количества рецепторов может варьироваться от одной тройки к другой в полях таблицы. Таким же образом, таблицы используются только для задачи связей между ячейками используемой памяти и другими ячейками системы, и, когда эти связи будут стабилизированы, необходимость в таблицах отпадает.

Табл. 3.3

С	А	В
F	Е	D
M	Н	J
Q	К	M
S	T	P
W	Y	R

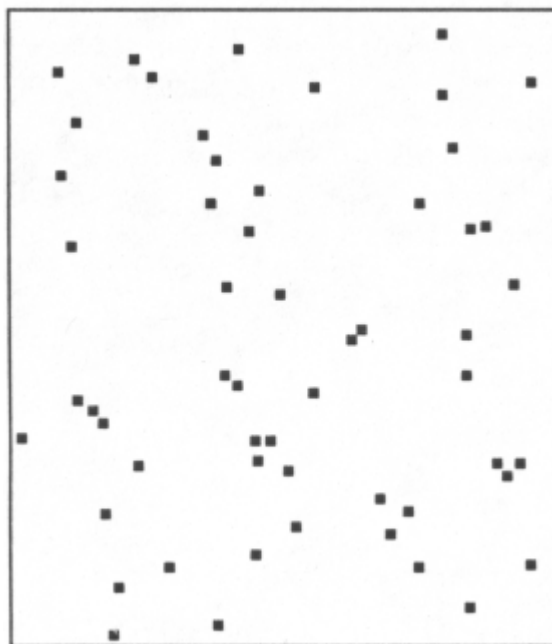


Рис. 3.19

Внешний наблюдатель может протестировать, где находится очередная тройка в памяти, с помощью проверки процентного соотношения активных, для данной тройки, рецепторов. Если это значение равно 100, то эта тройка может быть признана как текущая.

Например, если тройка (F A B) была записана в память, то соответствующая ячейка табл. 3.3 будет активна, наряду с остальными 27-ю. Однако, хотя тройка (C K R), например, также находится в активном поле, то для нее количество активных рецепторов для двух несвязанных троек будет небольшим. Поэтому, когда (F A B) записывается в память с помощью всех 28 рецепторов, содержащих тройку (C K R), только один рецептор будет активным, а именно тот, который содержит обе тройки.

В принципе, если в наличии имеется лист с общим количеством рецепторов для каждой тройки, то мы должны только оценить процент активных рецепторов и определить существует или нет данная тройка. Например, если тройки (F A B) и (F C D) занесены в рабочую область памяти, и если эта область состоит из массива 40×50 , тогда 55 ячеек будут активными, как показано на рис. 3.19. Текущие позиции ячеек не являются информативными, так как поля ячеек были сгенерированы случайным образом.

Рабочая область памяти на рис. 3.19 может быть также интерпретирована в вероятностных понятиях. Например, первая доза троек с наивысшим значением активных рецепторов показана в табл. 3.4.

Табл. 3.4

Тройка	Процент активности	Число активных рецепторов	Всего рецепторов
(F A B)	100	28	28
(F C D)	100	28	28
(F A D)	40	11	27
(F B D)	38	10	26
(F A X)	37	11	29
(S A B)	37	10	27

(F Q D)	37	10	27
(F C N)	37	10	27
(F C B)	37	10	27
(F C M)	35	10	28
(F T D)	35	10	28
(N C D)	34	10	29

В некотором смысле, не столь важно каким образом можно расшифровать значения рабочей области памяти. Это возможно осуществить через алгоритмы Хопфилда или через машину Больцмана, или через соединение всех ячеек по их соответствующим тройкам. Однако не стоит забывать о том, что лингвистические символы, описанные через количество ячеек, как бы «размазаны» по памяти. То есть достигнута некоторая голографическая картина. Если одна или две ячейки повреждены, то символ все еще может быть распознан. Большое количество различных троек может быть помещено в память, однако, только относительно небольшое их количество может быть помещено за один раз. Таким же образом, стирание одного символа из памяти может уменьшить описание остальных символов настолько, что через некоторое время символ придется восстановить обратно.

3.3.3. Адаптация с различными степенями доверия

Уровни доверия могут использоваться в адаптивном распознавании изображения двумя способами.

В одном случае, значения отдельных входов и являются уровнями доверия, или уровнями членства изображения, как показано на рис. 3.20. Этот тип распознавания изображений особенно подходит для управления на базе нечеткой логики.

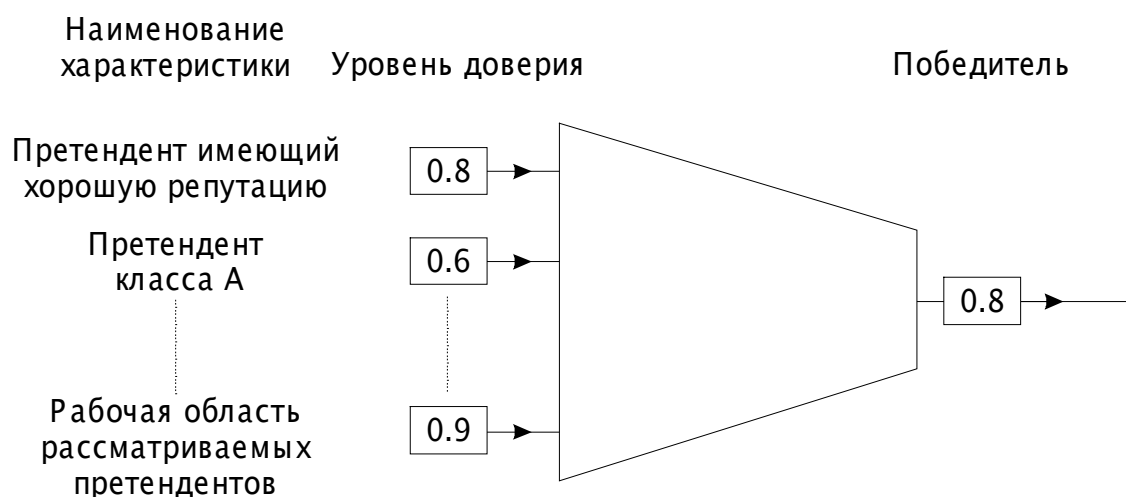


Рис. 3.20

В другом случае, уровни доверия могут быть предписаны самим изображением. Во время обучения сети классифицировать изображения или оценивать соответствия атрибутов заданным, мы можем оценить системную ошибку несколько по другому, чем это производится обычно.

Обычно, например, в обучении с учителем, системная ошибка вычисляется как среднее из всех ошибок изображения. Таким образом, мы получаем желательное

значение k -го выхода равно t_k и, если номер текущего выхода o_{pk} для изображения p , то системная ошибка минимальна если:

$$\langle E^2 \rangle = \frac{1}{2} \frac{1}{p} \sum_k \sum_p (t_k - o_{pk})^2. \quad (3.27)$$

Однако, если изменить формулу (3.27) для учета нашего уровня доверия в правильность каждого из этих изображений, тогда среднеквадратичная ошибка может быть записана как:

$$\langle E^2 \rangle = \frac{1}{2} \frac{1}{p} \sum_k \sum_p B_p (t_k - o_{pk})^2, \quad (3.28)$$

где B_p – уровень доверия для изображения \underline{x}_p .

Переменные B_p обеспечивают удобные и мощные средства для модификации дискриминантов операций обработки изображения. Например, установка $B_p = 0$ облегчит избавление от влияния \underline{x}_p , не используя при этом операции детального удаления информации из базы данных изображений.

3.3.4. Временная корреляция между входами и выходами сети

Существует несколько различных путей рассмотрения времени, как параметра в расчетах алгоритмов нейронных сетей. Если набор операторов работает асинхронно, то есть независимо от времени, то время никак не влияет на их порядок и в рассмотрение не берется. Последовательность или набор операторов, которые ведут последовательную обработку событий, независимо от времени их поступления, также не рассматриваются как функции времени. Однако, если последовательность операторов изменяется с течением времени, и их моменты синхронизации на разных стадиях процесса применяются для корректирования их взаимодействия или передачи информации, в таком случае рассмотрение времени, как параметра, невозможно обойти. Наконец, в подобных обстоятельствах, если текущее время используется совместно с одним или несколькими типами операций, тогда эти операции рассматриваются в «реальном» режиме времени.

Например, можно рассмотреть автоассоциативную сеть Хопфилда как гетероассоциативную, путем автоматического развития нормального набора переменных $\{t_{ij}\}$ в другой набор $\{T_{ij}\}$ после подачи на вход изображения \underline{x}'_p . Гетероассоциативным процессом называется процесс корректировки изображения \underline{x}'_p в \underline{x}_p через автоассоциацию, после которой веса $\{t_{ij}\}$ вместе со временем преобразуются в $\{T_{ij}\}$, и \underline{x}_p трансформируется в \underline{y}_p . Однако, приведенный процесс не отвечает понятию реального времени обработки, если уровень развития зависит от временных обстоятельств.

В еще одном методе учета времени, в котором временная константа и значение выхода сети определяется значением входов в различные временные интервалы, предшествующие времени снятия сигнала с выхода. Подобная сеть, таким образом, может использоваться для задач предсказания [28].

3.4. Заключение

В данном разделе были рассмотрены вопросы, касающиеся наиболее часто возникающих проблем в теории распознавания изображений посредством НС. В качестве вывода из приведенных выкладок можно было бы заметить, что наиболее сильной стороной данных методов является применение алгоритмов обработки информации через НС вместе с разработанными ранее методами обработки изображений. Подача на вход сети изображения, в виде простого перечисления состояния каждого пиксела в нем, привело бы к излишней загрузке сети ненужной информацией, что привело бы к понижению ее быстродействия. Подобный метод также привел бы к понижению, а точнее полной потере, инвариантности сети к изменению положения источников освещения, положения и ориентации объектов в изображении. Таким образом, основным выводом таков, что для реализации процесса распознавания объектов с помощью НС, перед непосредственной подачей сигнала на ее входы, следует предварительно обработать изображение, для уменьшения общего объема входного сигнала и увеличения его информативности. Например, в пункте 3.3.1.1. был приведен конкретный пример такого рода обработки, с помощью которой была достигнута инвариантность к перемещению объекта на экране, а также существенное уменьшение разрядности входного сигнала.

Итак, основываясь на методах, описанных в параграфах 2 и 3, можно перейти к конкретной реализации примера работы сети, целью которого будет создание реально действующей программы эмуляции процесса создания архитектуры сети, ее обучения и применения для реализации задачи распознавания объектов.