Лекция 4.
Системы распознавания образов (идентификации)

- Перцептроны.

- Нейронные сети. История исследований, модель с обратным распространением ошибки.

Перцептроны

Одним из методов решения задач обучения распознаванию образов основан на моделировании гипотетического механизма человеческого мозга.

Структура модели заранее постулируется. При таком подходе уровень биологических знаний или гипотез о биологических механизмах является исходной предпосылкой, на которой базируются модели этих механизмов.

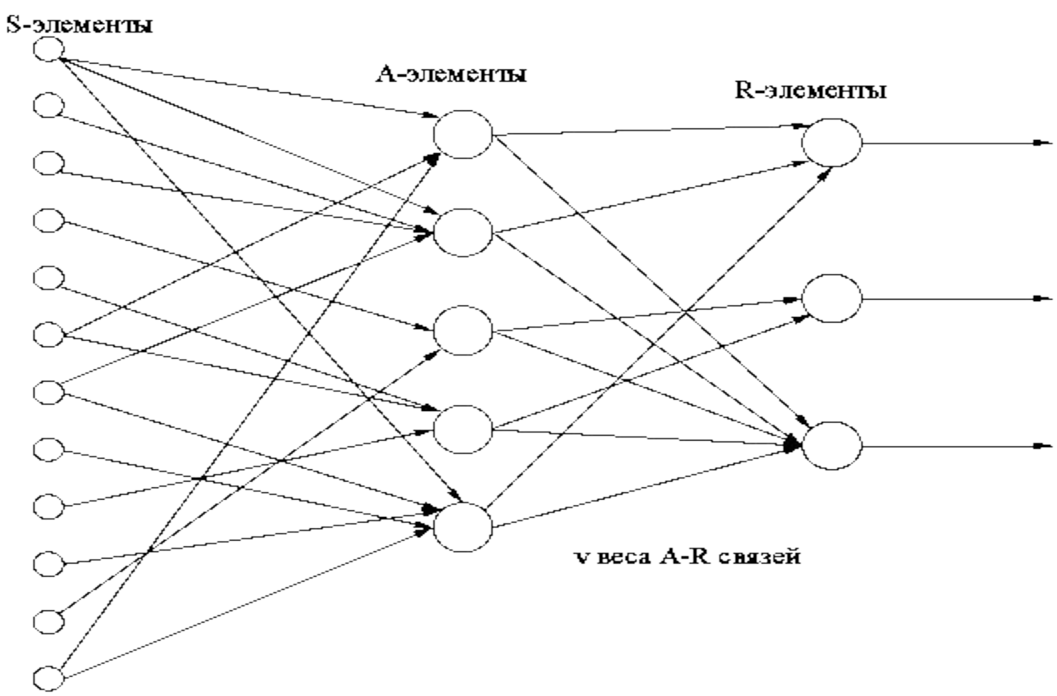
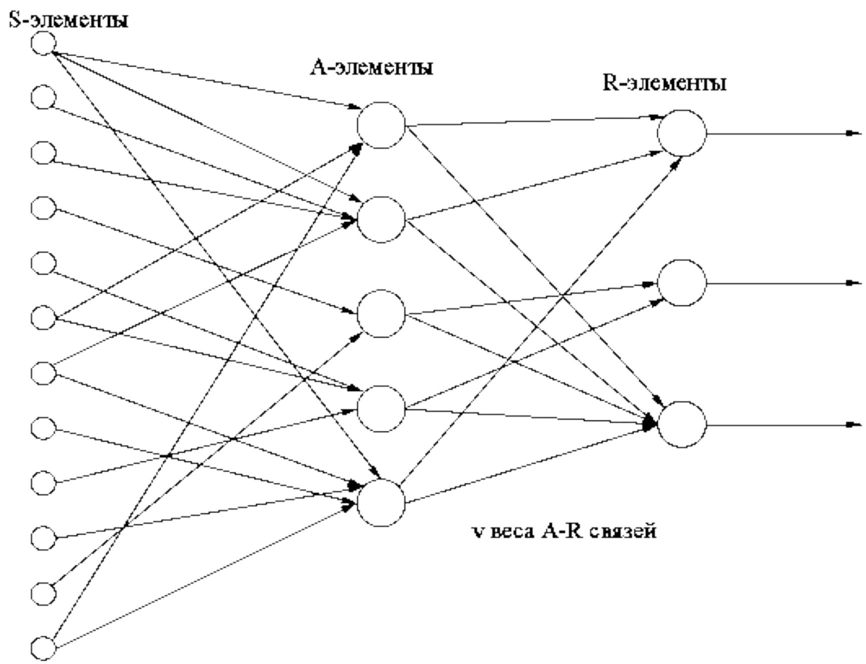


Рис. 1

Перцептроны



- совокупности <u>сенсорных элементов</u> <u>**S-элементов**</u>
- S-элементы <u>случайным образом</u> связаны с совокупностью <u>ассоциативных элементов</u> <u>**A-элементов**</u>, выход которых <u>отличается от нуля</u> только тогда, когда <u>возбуждено</u> достаточно большое число S-элементов.

- <u>A-элементы</u> соединены с <u>реагирующими элементами</u> <u>**R-элементами**</u> связями, <u>коэффициенты усиления</u> <u>**v**</u> которых переменны и <u>изменяются в процессе обучения</u>.
- Взвешенные комбинации выходов <u>R-элементов</u> составляют <u>реакцию системы</u>, которая указывает на <u>принадлежность</u> распознаваемого <u>объекта</u> <u>**определенному образу**</u>.

Перцептроны

Если распознаются только два образа, то в перцептроне устанавливается только один R-элемент, который обладает двумя реакциями — **положительной** и **отрицательной**.

Если образов больше двух, то для каждого образа устанавливают свой R-элемент, а выход каждого такого элемента представляет линейную комбинацию выходов A-элементов:

$$R_j = \Theta_j + \sum_{i=1}^{n} \nu_{ij} x_i \qquad (1)$$

R_j - реакция j-го *R*-элемента;

x_i - реакция i-го *A*-элемента;

Θ_j - порог j-го R-элемента.

ν_{ij} - вес связи от i-го A-элемента к j-му R-элементу;

Аналогично записывается уравнение i-го A-элемента:

$$x_i = \Theta_i + \sum_{k=1}^{S} y_k \qquad (2)$$

Сигнал *yk* может быть непрерывным, но чаще всего он принимает только два значения: 0 или 1. Сигналы от S-элементов подаются на входы A-элементов с постоянными весами равными 1, но каждый A-элемент связан только с группой случайно выбранных S-элементов.

Перцептроны

Предположим, что требуется обучить перцептрон различать два образа *V1* и *V2*.

Будем считать, что в перцептроне существует два *R*-элемента, один из которых предназначен образу *V1*, а другой — образу *V2*.

Перцептрон будет обучен правильно, если выход *R1* превышает *R2*, когда распознаваемый объект принадлежит образу *V1*, и наоборот.

Разделение объектов на два образа можно провести и с помощью только одного *R*-элемента. Тогда объекту образа *V1* должна соответствовать положительная реакция *R*-элемента, а объектам образа *V2* — отрицательная.

Перцептроны

Перцептрон обучается путем предъявления обучающей последовательности изображений объектов, принадлежащих образам *V1* и *V2*.

В процессе обучения изменяются веса *vi* *A*-элементов. В частности, если применяется система подкрепления с коррекцией ошибок, прежде всего учитывается правильность решения, принимаемого перцептроном.

Если решение правильно, то веса связей всех сработавших *A*-элементов, ведущих к *R*-элементу, выдавшему правильное решение, увеличиваются, а веса несработавших *A*-элементов остаются неизменными.

Можно оставлять неизменными веса сработавших *A*-элементов, но уменьшать веса несработавших. В некоторых случаях веса сработавших связей увеличивают, а несработавших — уменьшают.

После процесса обучения перцептрон сам, без учителя, начинает классифицировать новые объекты.

Перцептроны

Если в перцептроне допускаются лишь связи, идущие от бинарных *S*-элементов к *A*-элементам и от *A*-элементов к **единственному *R*-элементу**, то такой перцептрон принято называть **элементарным альфа-перцептроном**.

Обычно классификация *C(W)* задается учителем. Перцептрон должен выработать в процессе обучения классификацию, задуманную учителем.

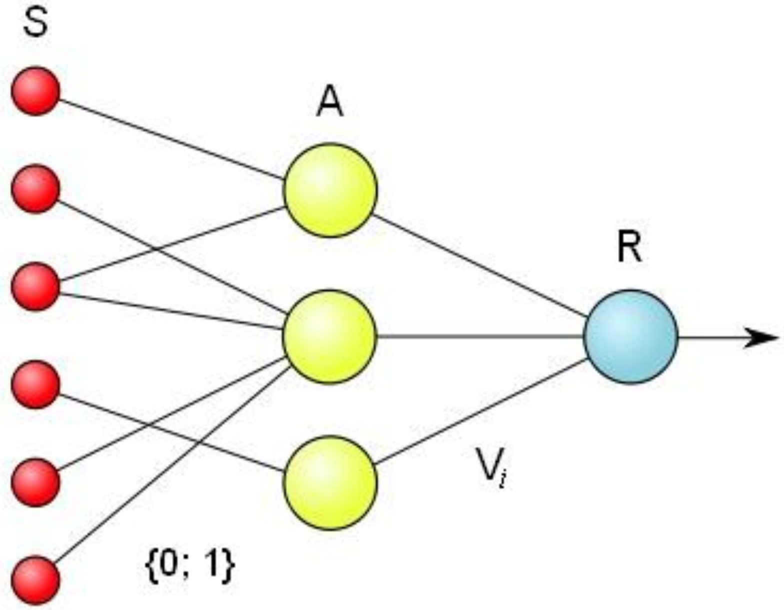


Рис. 2. Элементарный α-перцептрон

Перцептроны. Теоремы

О перцептронах было сформулировано и доказано несколько основополагающих теорем, две из которых, определяющие основные свойства перцептрона, приведены ниже.

Теорема 1. Класс элементарных альфа-перцептронов, для которых существует решение для любой задуманной классификации, не является пустым.

Эта теорема утверждает, что для любой классификации обучающей последовательности можно подобрать такой набор (из бесконечного набора) А-элементов, в котором будет осуществлено задуманное разделение обучающей последовательности при помощи линейного решающего правила).

Перцептроны. Теоремы

Теорема 2. Если для некоторой классификации C(W) решение существует, то в процессе обучения a -перцептрона с коррекцией ошибок, начинающегося с произвольного исходного состояния, это решение будет достигнуто в течение конечного промежутка времени.

Смысл этой теоремы состоит в том, что если относительно задуманной классификации можно найти набор А-элементов, в котором существует решение, то в рамках этого набора оно будет достигнуто в конечный промежуток времени.

Перцептроны. Теоремы

Обычно обсуждают свойства бесконечного перцептрона, т. е. перцептрона с бесконечным числом *A*-элементов со всевозможными связями с *S*-элементами (полный набор *A*-элементов).

В таких перцептронах решение всегда существует, а раз оно существует, то оно и достижимо в альфа-перцептронах с коррекцией ошибок.

Очень интересную область исследований представляют собой многослойные перцептроны и перцептроны с перекрестными связями, но теория этих систем практически еще не разработана.

Нейронные сети

- История исследований в области нейронных сетей
- Модель нейронной сети с обратным распространением ошибки (back propagation)

Нейронные сети. История исследований.

Способность нейронной сети к обучению впервые исследована **Дж. Маккалоком** и **У. Питтом**.

В 1943 году вышла их работа "Логическое исчисление идей, относящихся к нервной деятельности", в которой была построена модель нейрона, и сформулированы принципы построения искусственных нейронных сетей.

Крупный толчок развитию нейрокибернетики дал американский нейрофизиолог **Фрэнк Розенблатт**, предложивший в 1962 году свою модель нейронной сети — персептрон.

В 1982 году американский биофизик **Дж. Хопфилд** предложил оригинальную модель нейронной сети, названную его именем. В последующие несколько лет было найдено множество эффективных алгоритмов:

- сеть встречного потока,
- двунаправленная ассоциативная память и др.

В киевском институте кибернетики с 70-х годов ведутся работы над стохастическими нейронными сетями.

Модель нейронной сети с обратным распространением ошибки (back propagation)

В <u>1986</u> году <u>Дж. Хинтон</u> и его коллеги опубликовали статью с описанием модели нейронной сети и алгоритмом ее обучения, что дало новый толчок исследованиям в области искусственных нейронных сетей.

<u>Нейронная сеть состоит</u> из множества одинаковых элементов — <u>нейронов</u>, поэтому начнем с них рассмотрение работы <u>искусственной нейронной сети</u>.

<u>Биологический нейрон</u> моделируется как <u>устройство</u>, имеющее несколько <u>входов</u> (дендриты), и один <u>выход</u> (аксон).

<u>Каждому входу</u> ставится в соответствие некоторый <u>весовой коэффициент</u> **w**, характеризующий <u>пропускную способность канала</u> и оценивающий <u>степень влияния</u> сигнала с этого входа на сигнал на выходе.

Модель нейронной сети с обратным распространением ошибки (back propagation)

В зависимости от конкретной реализации, обрабатываемые нейроном <u>сигналы</u> могут быть <u>аналоговыми</u> или <u>цифровыми</u> (**1** или **0**).

В <u>теле нейрона</u> происходит <u>взвешенное суммирование</u> входных возбуждений, и далее это значение является аргументом <u>активационной функции</u> нейрона, один из возможных вариантов которой представлен на Рис. 3.

Модель нейронной сети с обратным распространением ошибки (back propagation)

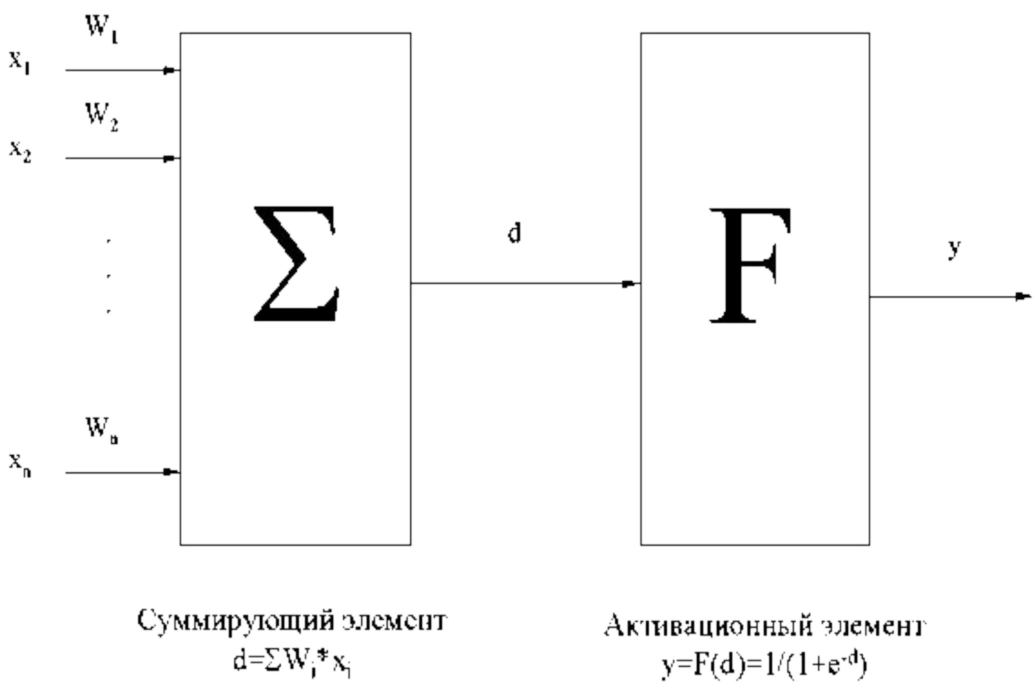


Рис. 3. Искусственный нейрон

Будучи соединенными определенным образом, нейроны образуют нейронную сеть. Работа сети разделяется на обучение и адаптацию.

Под обучением понимается процесс адаптации сети к предъявляемым эталонным образцам путем модификации (в соответствии с тем или иным алгоритмом) весовых коэффициентов связей между нейронами.

Заметим, что этот процесс является результатом алгоритма функционирования сети, а не предварительно заложенных в нее знаний человека, как это часто бывает в системах искусственного интеллекта.

Модель нейронной сети с обратным распространением ошибки (back propagation)

Среди различных структур нейронных сетей (НС) одной из наиболее известных является многослойная структура, в которой каждый нейрон произвольного слоя связан со всеми аксонами нейронов предыдущего слоя или, в случае первого слоя, со всеми входами НС.

Такие НС называются **полносвязными**.

Когда в сети только один слой, алгоритм ее обучения с учителем довольно очевиден, так как правильные выходные состояния нейронов единственного слоя заведомо известны, и подстройка синаптических связей идет в направлении, минимизирующем ошибку на выходе сети.

По этому принципу строится, например, алгоритм обучения однослойного перцептрона.

Модель нейронной сети с обратным распространением ошибки (back propagation)

В многослойных же сетях оптимальные выходные значения нейронов всех слоев, кроме последнего, как правило, не известны, и двух- или более слойный перцептрон уже невозможно обучить, руководствуясь только величинами ошибок на выходах НС.

Приемлемый вариант решения – распространение сигналов ошибки от выходов НС к ее входам, в направлении, обратном прямому распространению сигналов в обычном режиме работы.

Этот алгоритм обучения НС получил название процедуры обратного распространения. Именно он будет рассмотрен в дальнейшем.

Модель нейронной сети с обратным распространением ошибки (back propagation)

Согласно методу наименьших квадратов, минимизируемой целевой функцией ошибки НС является величина:

$$E(w) = \frac{1}{2} \sum_{j,p} (y_{j,p}^{(N)} - d_{j,p})^2 \qquad (3)$$

где $y_{j,p}^{(N)}$ – реальное выходное состояние нейрона *j* выходного слоя *N* нейронной сети при подаче на ее входы *p*-го образа; d_{jp} – идеальное (желаемое) выходное состояние этого нейрона.

Суммирование ведется <u>по всем нейронам</u> выходного слоя и <u>по всем обрабатываемым сетью образам</u>.
Минимизация ведется методом <u>градиентного спуска</u>, что означает <u>подстройку весовых коэффициентов</u> следующим образом:

$$\Delta w_{ij}^{(n)} = -\eta \cdot \frac{\partial E}{\partial w_{ij}} \qquad (4)$$

здесь w_{ij} – весовой коэффициент синаптической связи, соединяющей *i*-ый нейрон слоя *n*-1 с *j*-ым нейроном слоя *n*; η – коэффициент скорости обучения, 0 < η < 1.

Модель нейронной сети с обратным распространением ошибки (back propagation)

Как показано в [**Розенблатт Ф.** Принципы нейродинамики (перцептрон и теория механизмов мозга).]

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial y_j} \cdot \frac{dy_j}{ds_j} \cdot \frac{\partial s_j}{\partial w_{ij}} \qquad (5)$$

Здесь под y_j, как и раньше, подразумевается выход нейрона *j*, а под s_j – взвешенная сумма его входных сигналов, то есть аргумент <u>активационной функции</u>.

Так как множитель $\frac{dy_j}{ds_j}$ является производной этой функции по ее аргументу, из этого следует, что производная активационной функции должна быть определена на всей оси абсцисс.

Модель нейронной сети с обратным распространением ошибки (back propagation)

В связи с этим функция единичного скачка и прочие активационные функции с неоднородностями не подходят для рассматриваемых НС.
В них применяются такие гладкие функции, как гиперболический тангенс или классический сигмоид с экспонентой. В случае гиперболического тангенса

$$\frac{dy}{ds} = 1 - s^2 \qquad (6)$$

Третий множитель $\frac{\partial s_j}{\partial w_{ij}}$ очевидно, равен выходу нейрона предыдущего слоя $y_i^{(n-1)}$

Модель нейронной сети с обратным распространением ошибки (back propagation)

Что касается первого множителя в (5), он легко раскладывается следующим образом:

$$\frac{\partial E}{\partial y_j} = \sum_k \frac{\partial E}{\partial y_k} \cdot \frac{dy_k}{ds_k} \cdot \frac{\partial s_k}{\partial y_j} = \sum_k \frac{\partial E}{\partial y_k} \cdot \frac{dy_k}{ds_k} \cdot w_{jk}^{(n+1)} \qquad (7)$$

Здесь суммирование по *k* выполняется среди нейронов слоя *n+1*.
Введя новую переменную:

$$\delta_j^{(n)} = \frac{\partial E}{\partial y_j} \cdot \frac{dy_j}{ds_j} \qquad (8)$$

мы получим рекурсивную формулу для расчетов величин $\delta_j^{(n)}$ слоя *n* из величин $\delta_j^{(n+1)}$ более старшего слоя *n+1*.

$$\delta_{\mathrm{j}}^{(\mathrm{n})} = \left[\sum_k \delta_k^{(n+1)} \cdot w_{jk}^{(n+1)} \right] \cdot \frac{dy_j}{ds_j} \qquad (9)$$

Для выходного же слоя:

$$\delta_l^{(N)} = (y_l^{(N)} - d_l) \cdot \frac{dy_l}{ds_l} \qquad (10)$$

Модель нейронной сети с обратным распространением ошибки (back propagation)

Теперь мы можем записать (4) в раскрытом виде:

$$\Delta w_{ij}^{(n)} = -\eta \cdot \delta_j^{(n)} \cdot y_i^{(n-1)} \qquad (11)$$

Иногда для придания процессу коррекции весов некоторой инерционности, сглаживающей резкие скачки при перемещении по поверхности целевой функции, (11) дополняется значением изменения веса на предыдущей итерации

$$\Delta w_{ij}^{(n)}(t) = -\eta \cdot (\mu \cdot \Delta w_{ij}^{(n)}(t-1) + (1-\mu) \cdot \delta_j^{(n)} \cdot y_i^{(n-1)}) \qquad (12)$$

где μ – коэффициент инерционности, t – номер текущей итерации.

Модель нейронной сети с обратным распространением ошибки (back propagation)

Таким образом, полный алгоритм обучения НС с помощью процедуры обратного распространения ошибки строится так:

1. Подать на входы сети один из возможных образов и в режиме обычного функционирования НС, когда сигналы распространяются от входов к выходам, рассчитать значения последних. Напомним, что

$$s_j^{(n)} = \sum_{i=0}^{M} y_i^{(n-1)} \cdot w_{ij}^{(n)} \qquad (13)$$

где *M* – число нейронов в слое *n-1* с учетом нейрона с постоянным выходным состоянием +1, задающего смещение;

$y_i^{(n-1)} = x_{ij}^{(n)}$ – *i* - ый вход нейрона *j* слоя *n*.

$$y_j^{(n)} = f(s_j^{(n)}) \qquad (14)$$

где *f()* – сигмоид

$$y_q^{(0)} = \mathrm{I}_q \qquad (15)$$

где I_q – *q*-ая компонента вектора входного образа.

Модель нейронной сети с обратным распространением ошибки (back propagation)

2. Рассчитать $\delta^{(N)}$ для выходного слоя по формуле (10).

Рассчитать по формуле (11) или (12) изменения весов $\Delta w^{(N)}$ слоя *N*.

3. Рассчитать по формулам (9) и (11) (или (9) и (12)) соответственно $\delta^{(n)}$ и $\Delta w^{(n)}$ для всех остальных слоев, *n = N-1, ..., 1* .

4. Скорректировать все веса в НС

$$w_{ij}^{(n)}(t) = w_{ij}^{(n)}(t-1) + \Delta w_{ij}^{(n)}(t) \tag{16}$$

5. Если ошибка сети существенна, перейти на шаг 1. В противном случае – конец.

Модель нейронной сети с обратным распространением ошибки (back propagation)

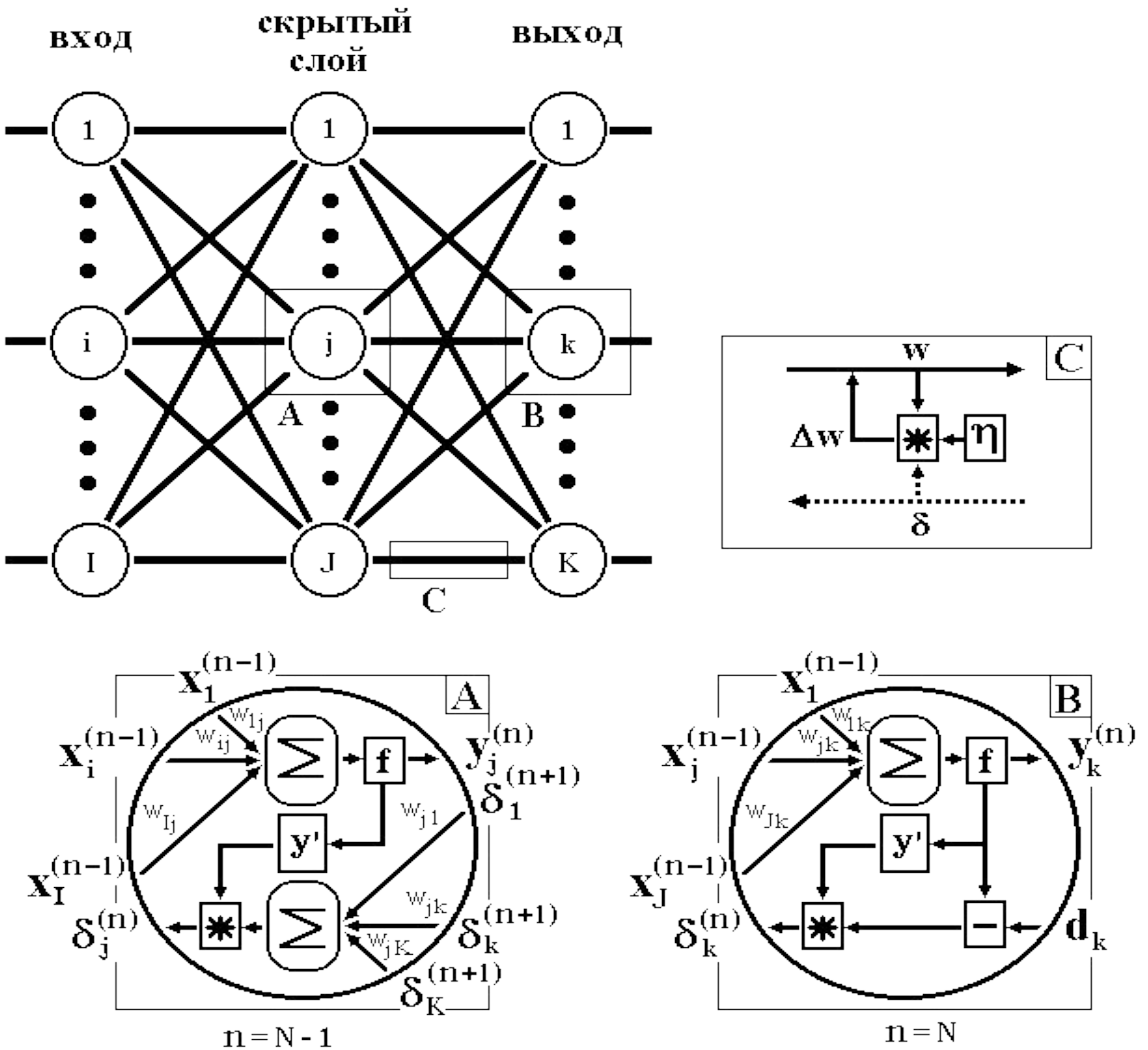


Рис. 4

Модель нейронной сети с обратным распространением ошибки (back propagation)

Сети на шаге 1 попеременно в случайном порядке предъявляются все тренировочные образы, чтобы сеть, образно говоря, не забывала одни по мере запоминания других. Алгоритм иллюстрируется Рис. 4

Из выражения (11) следует, что когда выходное значение yi(n-1) стремится к нулю, эффективность обучения заметно снижается. При двоичных входных векторах в среднем половина весовых коэффициентов не будет корректироваться[3], поэтому область возможных значений выходов нейронов [0,1] желательно сдвинуть в пределы [-0.5,+0.5], что достигается простыми модификациями логистических функций. Например, сигмоид с экспонентой преобразуется к виду:

$$f(x) = -0.5 + \frac{1}{1 + e^{-\alpha \cdot x}} \qquad (17)$$

Модель нейронной сети с обратным распространением ошибки (back propagation)

Теперь коснемся вопроса ёмкости НС, то есть числа образов, предъявляемых на ее входы, которые она способна научиться распознавать. Для сетей с числом слоев больше двух, он остается открытым. Как показано в [4], для НС с двумя слоями, то есть выходным и одним скрытым слоем, детерминистская емкость сети Cd оценивается так:

$$\frac{N_w}{N_y} < C_d < \frac{N_w}{N_y} \log\left(\frac{N_w}{N_y}\right) \qquad (18)$$

где *Nw* – число подстраиваемых весов, *Ny* – число нейронов в выходном слое.

Следует отметить, что данное выражение получено с учетом некоторых ограничений. Во-первых, число входов Nx и нейронов в скрытом слое *Nh* должно удовлетворять неравенству *Nx+Nh>Ny*.

Во-вторых, *Nw/Ny > 1000*. Однако вышеприведенная оценка выполнялась для сетей с активационными функциями нейронов в виде порога, а емкость сетей с гладкими активационными функциями, например – (17), обычно больше.