Алгоритм обратного распространения ошибки

Цель алгоритма - обновить вес каждой связи в сети так, чтобы получаемые на выходе значения были как можно ближе к целевым значениям, тем самым уменьшая ошибку как для каждого выходного нейрона, так и для сети в целом.

После ввода начальных данных и прохождения сигнала по сети нейроны выходного слоя формируют ответ. При обучении полученный ответ сравнивается с целевым значением, вычисляется ошибка для каждого нейрона и для сети в целом (с помощью функции потерь).

Чтобы оценить хорошо или плохо сеть решает поставленные задачи строится `целевая функция` для всей сети, а для каждого выходного нейрона - `функция оценки` (другое название `функция потерь`; на англ.: `cost function` и `loss function` соответственно).

Самая простая функция оценки - `квадратичная`, ошибка на произвольном нейроне выходного слоя равна:

$$error_j = (target_j - output_j)^2$$

Простая целевая функция для всей сети - квадратичная, полученная по методу наименьших квадратов:

$$E_{total} = \frac{1}{2}\sum_j (target_j - output_j)^2$$

Обратное распространение ошибки и обновление весов происходит от выходного слоя к входному.

Разные источники используют разную нотацию при обозначении функции активации, целевой и сумматорной функций. Для наглядности приведена таблица обозначений в источниках, использованных при написании данной пояснительной записки:

Обозначаемый элемент	**Первый вариант нотации**	**Второй вариант нотации**
Целевая функция	C или J	E_{total} или Q
Значение на выходе после активации	a	out
Сумматорная функция	z	net
Функция активации	σ	g или σ
Коэф. скорости обучения (learning rate)	η	η (возм. ϵ или α)
момент (momentum)	α (возм. μ)	α (возм. ρ)

Сумматорная функция выполняет сложение взвешенных входов в нейрон. В общем виде для нейрона j из слоя с номером l можно записать:

$$z_j^l = \sum_k w_{jk}^l a_k^{l-1} + b_j^l$$

где:

- w_{jk}^l - вес связи которая соединяет текущий нейрон j (в слое l) с нейроном k из предыдущего слоя (номер предыдущего слоя l-1)
- a_k^{l-1} - значение на выходе нейрона k из предыдущего слоя после применения функции активации.

Значение на выходе нейрона j из слоя l после активации в общем виде можно записать:

$$a_j^l = \sigma\left(\sum_k w_{jk}^l a_k^{l-1} + b_j^l\right) = \sigma(z_j^l)$$

где σ- функция активации (activation function или другое название transfer function).

В качестве функции активации (σ) примем гиперболический тангенс(см. рис.4)

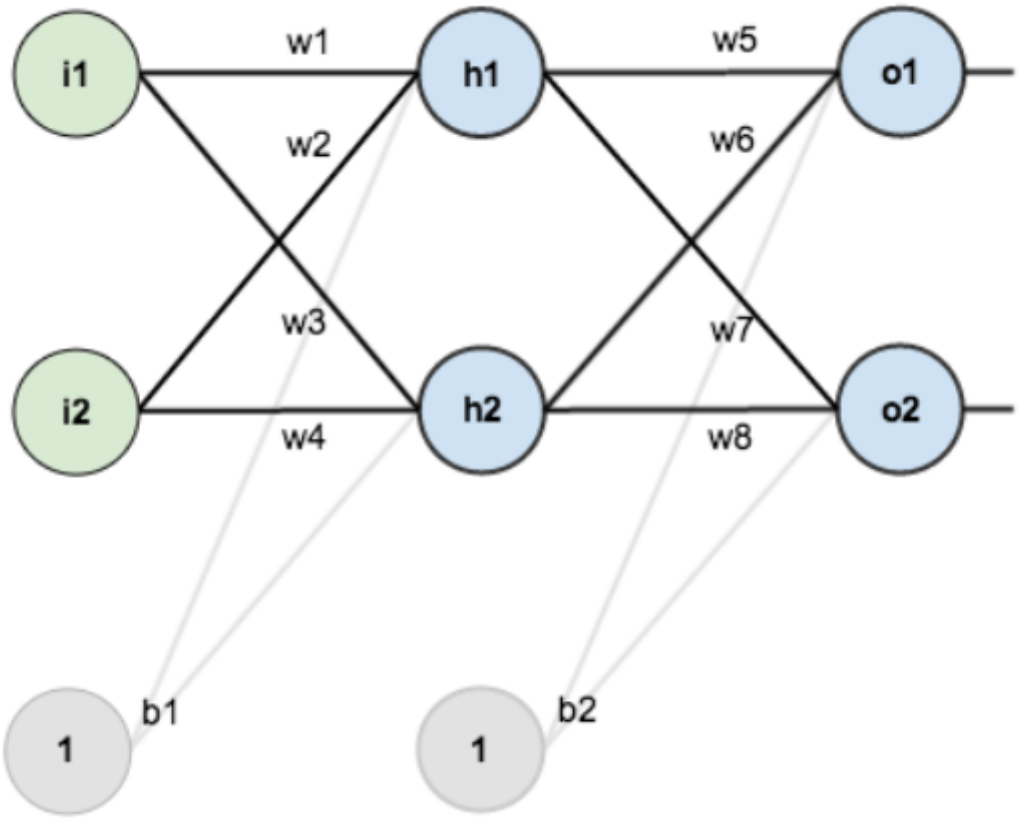

Рис.4. Гиперболический тангенс

Для изменения весов воспользуемся самым простым уравнением (позже немного усложним его):

$$(w_{jk}^l)' = w_{jk}^l - \eta\frac{\partial C}{\partial w_{jk}^l}$$

где η - коэффициент скорости обучения (learning rate)

Выходной слой

Рассмотрим алгоритм на примере простой трёхслойно сети (см.рис 5)

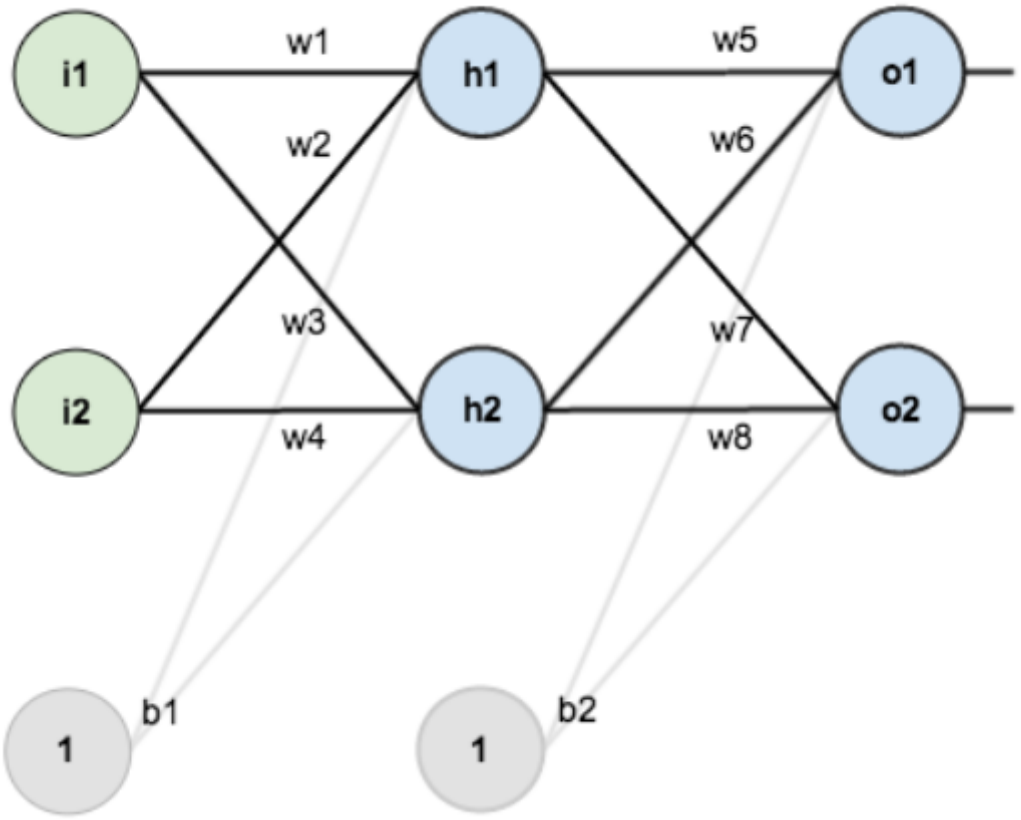

Рис. 5 Простая сеть

Нам необходимо определить какой вклад в общую ошибку вносит каждая связь нейронов выходного слоя с предыдущим слоем. Рассмотрим, какой вклад вносит связь первого нейрона выходного слоя с первым нейроном последнего скрытого слоя (вес w5). Т.е. необходимо вычислить частную производную целевой функции E_{total} по весу w5. Ошибка является функцией выходных значений, выходное значение - функцией суммы взвешенных входов, сумма входов - функцией от взвешенных входов. Тогда, применяя цепное правило можем записать:

$$\frac{\partial E_{total}}{\partial w_5} = \frac{\partial E_{total}}{\partial out_{o1}} \cdot \frac{\partial out_{o1}}{\partial net_{o1}} \cdot \frac{\partial net_{o1}}{\partial w_5}$$

Или в более наглядном виде:

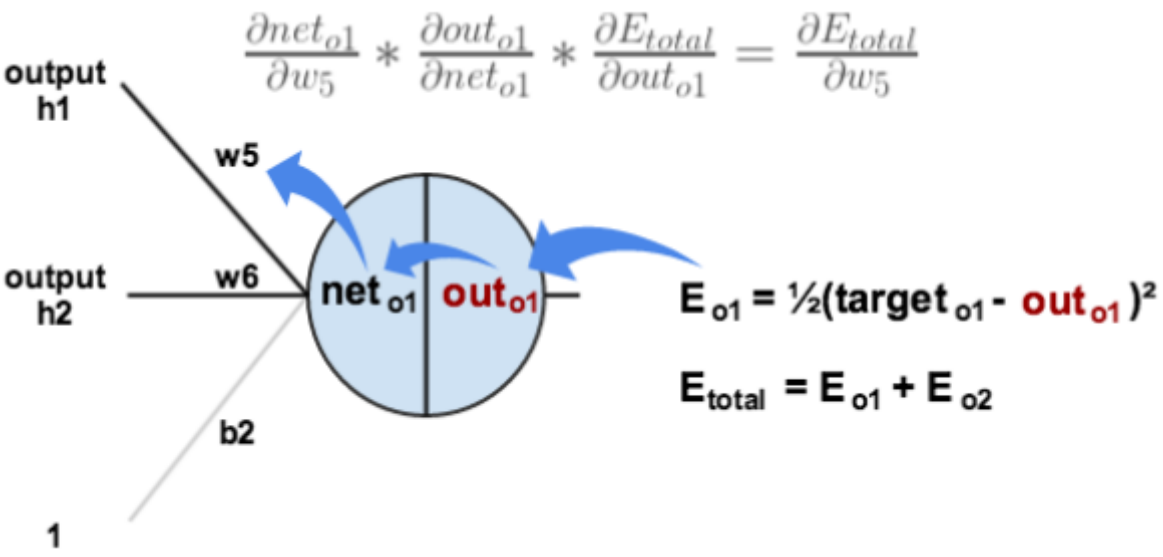

Рис.6 Вычисление вклада одной связи в общую ошибку.

Ошибка (дельта) в нейроне `j` слоя `l` по определению равна:

$$\delta_j^l \equiv \frac{\partial C}{\partial z_j^l}$$

Или в другой нотации:

$$\delta_j^l \equiv \frac{\partial E_{total}}{\partial net_j^l}$$

Ошибка (дельта) в нейроне `j` выходного слоя `L` равна:

$$\delta_j^L \equiv \frac{\partial C}{\partial a_j^L} \cdot \sigma'(z_j^L)$$

Или:

$$\delta_j^L \equiv \frac{\partial E_{total}}{\partial out_j^L} \cdot \sigma'(net_j^L)$$

В матричном виде:

$$\delta^{\mathbf{L}} = \nabla_a \mathbf{C} \odot \sigma'(\mathbf{z^L})$$

где:

- $\nabla_a \mathbf{C}$ - градиент вектора C
- $\odot$ - произведение Адамара.

Частная производная ошибки сети по выходному значению нейрона последнего слоя:

$$\frac{\partial E_{total}}{\partial out_j^L} = \frac{\partial(\frac{1}{2}\sum_j(target_j - output_j)^2)}{\partial output_j} = (target_j - output_j) \cdot (-1) = output_j - target_j$$

Тогда, произведение первых двух частных производных из уравнения перед рис.6 можно переписать в виде:

$$\frac{\partial E_{total}}{\partial out_{o1}} \cdot \frac{\partial out_{o1}}{\partial net_{o1}} = \frac{\partial E_{total}}{\partial net_{o1}} = \delta_{o1}$$

Как уже упоминалось, эта частная производная в некоторых источниках называют `дельтой` (node delta). Само же уравнение примет вид:

$$\frac{\partial E_{total}}{\partial w_5} = \delta_{o1} \cdot \frac{\partial net_{o1}}{\partial w_5}$$

После вычисления вклада веса связи в ошибку, можем найти новое значение веса:

$$w_5^+ = w_5 - \eta \cdot \frac{\partial E_{total}}{\partial w_5}$$

где η - коэффициент скорости обучения сети (`learning rate`)

Подобным образом находим новые значения всех оставшихся весов связей между выходным слоем и последним скрытым - w6, w7, w8 (см. рис. 5)

ВАЖНО При дальнейшем выполнении алгоритма обратного распространения ошибки используются исходные веса (НЕ обновлённые). Все веса обновляются после того как будут вычислены их вклады в общую ошибку сети.

Скрытые слои

Продолжим движение к входному слою и вычислим новые значения весов w1, w2, w3, w4 (см.рис.4). Для этого нам потребуется вычислить вклад каждого из этих весов в общую ошибку:

$$\frac{\partial E_{total}}{\partial w_1} = \frac{\partial E_{total}}{\partial out_{h1}} \cdot \frac{\partial out_{h1}}{\partial net_{h1}} \cdot \frac{\partial net_{h1}}{\partial w_1}$$

Более наглядно на рис. 7:

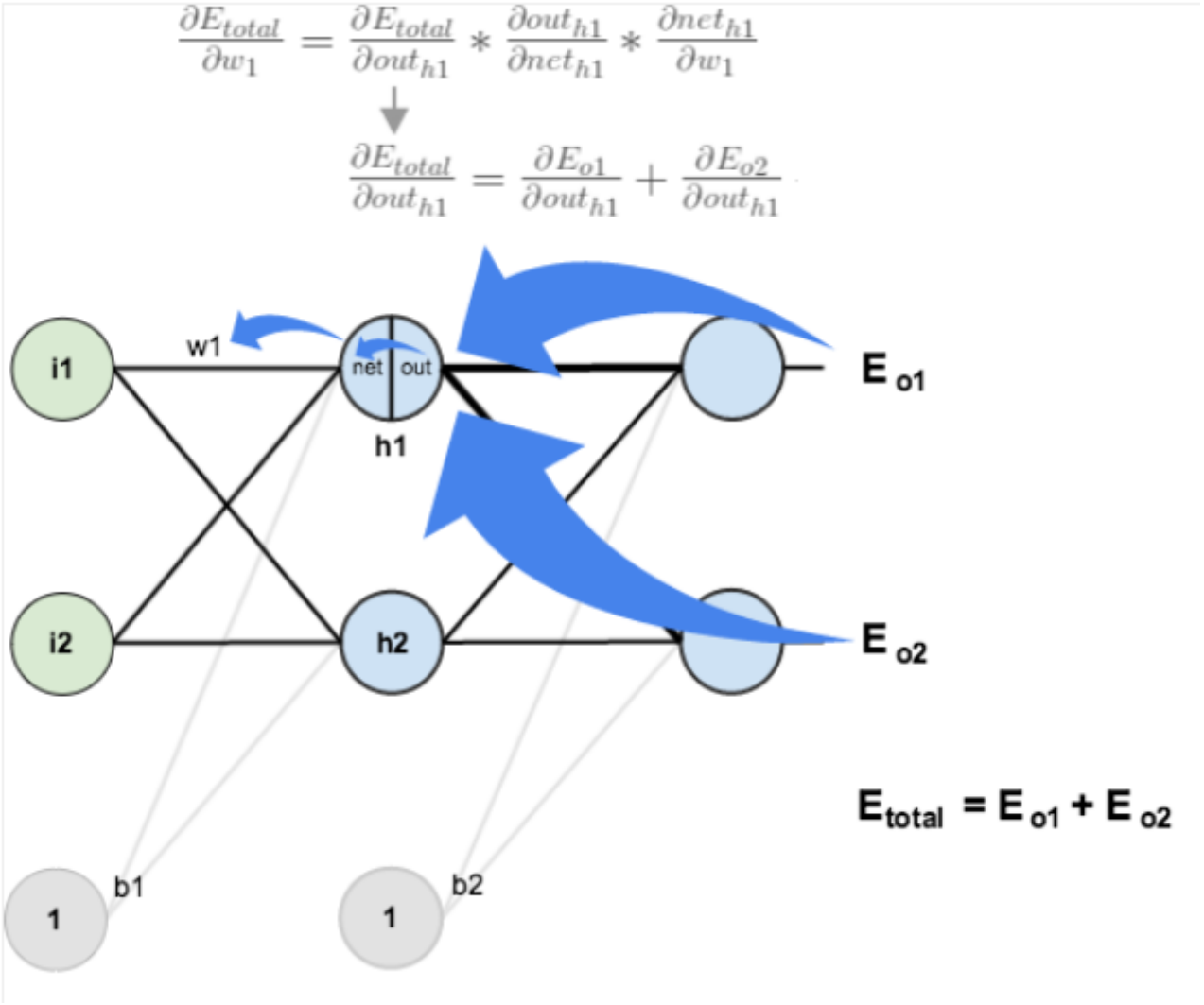

Рис.7 Вычисление вклада связи w1 в ошибку

Расчёт ошибки схож с расчётом для выходного слоя, однако, следует учесть, что выходное значение скрытого нейрона влияет на выходные значения всех нейронов последнего слоя и тем самым на ошибку.

Итак, согласно рис. 4, out_{h1} влияет на out_{o1} и на out_{o2}, поэтому можем записать:

$$\frac{\partial E_{total}}{\partial out_{h1}} = \frac{\partial E_{o1}}{\partial out_{h1}} + \frac{\partial E_{o2}}{\partial out_{h1}}$$

Рассмотрим первое слагаемое:

$$\frac{\partial E_{o1}}{\partial out_{h1}} = \frac{\partial E_{o1}}{\partial net_{o1}} \cdot \frac{\partial net_{o1}}{\partial out_{h1}}$$

Рассмотрим сомножители.

Первый сомножитель. Необходимо обратить внимание на следующее:

- по определению ошибка в узле это $\delta_j^l \equiv \frac{\partial E_{total}}{\partial net_j^l}$
- целевая функция представляет собой сумму ошибок в каждом узле выходного слоя: $E_{total} = \frac{1}{2}\sum_j (target_j - output_j)^2$

Тогда для `выходного` слоя справедливо:

$$\frac{\partial E_{total}}{\partial net_{o1}} = \frac{\partial E_{o1}}{\partial net_{o1}}$$

Таким образом:

$$\frac{\partial E_{o1}}{\partial net_{o1}} = \delta_{o1}$$

Второй сомножитель. Распишем net_{o1}(по рис.5):

$$net_{o1} = w_5 \cdot out_{h1} + w_6 \cdot out_{h2} + b_2 \cdot 1$$

Тогда:

$$\frac{\partial net_{o1}}{\partial out_{h1}} = w_5$$

Перепишем формулу для $\frac{\partial E_{o1}}{\partial out_{h1}}$:

$$\frac{\partial E_{o1}}{\partial out_{h1}} = \delta_{o1} \cdot w_5$$

По аналогии получаем, что:

$$\frac{\partial E_{o2}}{\partial out_{h1}} = \delta_{o2} \cdot w_7$$

Общий вклад в ошибку сети от выхода нейрона скрытого слоя:

$$\frac{\partial E_{total}}{\partial out_{h1}} = \frac{\partial E_{o1}}{\partial out_{h1}} + \frac{\partial E_{o2}}{\partial out_{h1}} = \delta_{o1} \cdot w_5 + \delta_{o2} \cdot w_7$$

Вернёмся к формуле

$$\frac{\partial E_{total}}{\partial w_1} = \frac{\partial E_{total}}{\partial out_{h1}} \cdot \frac{\partial out_{h1}}{\partial net_{h1}} \cdot \frac{\partial net_{h1}}{\partial w_1}$$

Первые 2 сомножителя равны ошибке узла по определению:

$$\delta_{h1} \equiv \frac{\partial E_{total}}{\partial net_{h1}} = \frac{\partial E_{total}}{\partial out_{h1}} \cdot \frac{\partial out_{h1}}{\partial net_{h1}} = (\delta_{o1} \cdot w_5 + \delta_{o2} \cdot w_7) \cdot \sigma'(net_{h1})$$

Приведём без доказательства формулу для вычисления ошибки в узле `j` слоя `l` в общем случае:

$$\delta_j^l = \sum_k w_{kj}^{l+1} \delta_k^{l+1} \sigma'(z_j^l)$$

Или в другой нотации:

$$\delta_j^l = \sum_k w_{kj}^{l+1} \delta_k^{l+1} \sigma'(net_j^l)$$

Итак, для рис.5:

$$\frac{\partial E_{total}}{\partial w_1} = \frac{\partial E_{total}}{\partial out_{h1}} \cdot \frac{\partial out_{h1}}{\partial net_{h1}} \cdot \frac{\partial net_{h1}}{\partial w_1} = \delta_{h1} \cdot \frac{\partial net_{h1}}{\partial w_1} = \delta_{h1} \cdot i_1$$

где i_1 - значение (выходное) первого нейрона входного слоя

Новый вес w1:

$$w_1^+ = w_1 - \eta \cdot \frac{\partial E_{total}}{\partial w_1} = w_1 - \eta \cdot \delta_{h1} \cdot i_1$$

Аналогичным образом получим новые значения для весов w2, w3, w4.

Краткий итог

Выпишем формулы, которые потребуются для реализации алгоритма.

Целевая функция:

$$C = E_{total} = \frac{1}{2} \sum_j (target_j - output_j)^2$$

Вектор ошибок в нейронах выходного слоя:

$$\delta^{\mathbf{L}} = \nabla_a \mathbf{C} \odot \sigma'(\mathbf{z^L})$$

Компоненты градиента для нейронов выходного слоя:

$$\frac{\partial C}{\partial a_j^L} = \frac{\partial E_{total}}{\partial out_j^L} = \frac{\partial (\frac{1}{2} \sum_j (target_j - output_j)^2)}{\partial output_j} = (target_j - output_j) \cdot (-1) = output_j - target_j$$

Производная гиперболического тангенса:

$$\tanh' x = \frac{d}{dx} \tanh x = 1 - \tanh^2 x$$

Исходя из предыдущего уравнения для нашей сети справедливо:

$$\sigma'(z_j^L) = \tanh'(z_j^L) = 1 - \tanh(z_j^L) \cdot \tanh(z_j^L) = 1 - \sigma(z_j^L) \cdot \sigma(z_j^L)$$

Ошибка в узлах скрытого слоя:

$$\delta_j^l = \sum_k w_{kj}^{l+1} \delta_k^{l+1} \sigma'(z_j^l)$$

Перепишем формулу в более удобном виде и каждый множитель впоследствии реализуем отдельной функцией:

$$\delta_j^l = \sigma'(z_j^l) \cdot \sum_k w_{kj}^{l+1} \delta_k^{l+1}$$

Простая формула для обновления весов связей:

$$(w_{jk}^l)' = w_{jk}^l + \Delta w_{jk}^l = w_{jk}^l - \eta \frac{\partial C}{\partial w_{jk}^l}$$

где

$$\frac{\partial C}{\partial w_{jk}^l} = a_k^{l-1} \delta_j^l$$

В реализации воспользуемся чуть более сложной версией формулы. Изменение веса на итерации t:

$$\Delta w_{jk}^l(t) = -\eta \frac{\partial C}{\partial w_{jk}^l(t)} + \alpha \Delta w_{jk}^l(t-1)$$

После подстановки:

$$\Delta w_{jk}^l(t) = -\eta \left(a_k^{l-1}(t) \delta_j^l(t) \right) + \alpha \Delta w_{jk}^l(t-1)$$

Фактически используемы вариант формулы:

$$(w_{jk}^l)' = w_{jk}^l + \Delta w_{jk}^l = w_{jk}^l - \eta \left(a_k^{l-1} \delta_j^l \right) + \alpha (\Delta w_{jk}^l)''$$

где :

- $(\Delta w_{jk}^l)''$ - изменение веса на предыдущей итерации. Фактически мы добавили к начальной формуле только это слагаемое с коэффициентом.
- j - текущий нейрон в слое l
- k - нейрон из предыдущего слоя
