МАТЕМАТИЧЕСКИЕ ОСНОВЫ ТЕОРИИ ПРОГНОЗИРОВАНИЯ

Лектор

Сенько Олег Валентинович

Лекция 6

Нейросетевые методы распознавания

- В основе нейросетевых методов лежит попытка компьютерного моделирования процессов обучения, используемых в живых организмах. Когнитивные способности живых существ связаны с функционированием сетей связанных между собой биологических нейронов – клеток нервной системы. Для моделирования биологических нейросетей используются сети искусственных нейронов, Можно выделить три типа искусственных нейронов: нейроны-рецепторы, внутренние нейроны и реагирующие нейроны.

Модель искусственного нейрона

Каждый внутренний или реагирующий нейрон имеет множество входных связей, по которым поступают сигналы от рецепторов или других нейронов. Предположим, что нейрон имеет r внешних связей, по которым поступают сигналы

$(u_1, \dots, u_r)$. Поступившие сигналы суммируются с весами $(w_1, \dots, w_r)$

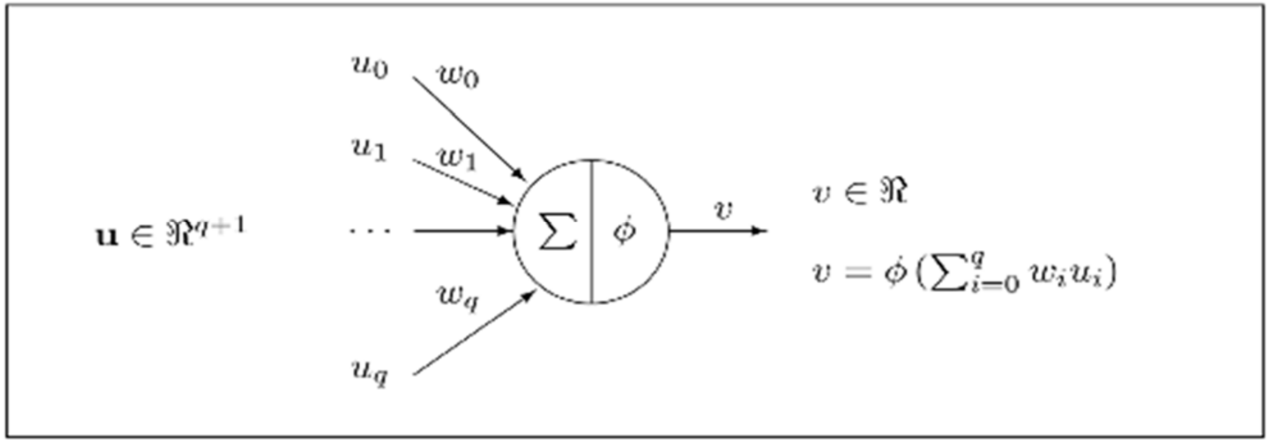
. На выходе нейрона вырабатывается сигнал $\Phi(z)$, где

$z = \sum_{i=1}^{r} w_i u_i + w_0$, w_0 - параметр сдвига. Может быть использована также форма записи $z = \sum_{i=0}^{r} w_i u_i$, где фиктивный «сигнал» u_0 тождественно равен 1.

Модель искусственного нейрона

Функцию $\Phi(\xi)$ обычно называют, активационной функцией.

Модель внутреннего или реагирующего нейрона быть схематично изображена на рисунке 1.



Модель искусственного нейрона

Могут использоваться различные виды активационных функций.

Например,

Пороговая функция

$$\Phi(\xi) = \begin{cases} a_u & npu\ \xi \geq b \\ a_l & npu\ \xi < b \end{cases}$$

Сигмоидная функция

$$\Phi(\xi) = 1/[1 + \exp(-a\xi)]$$

Гиперболический тангенс

$$\Phi(\xi) = th(\xi)$$

Тождественное преобразование

$$\Phi(\xi) = \xi$$

Перцептрон Розенблатта

- Первой нейросетевой моделью стал перцептрон Розенблатта, предложенный в 1957 году. В данной модели используется единственный реагирующий нейрон. Модель, реализующая линейную разделяющую функцию в пространстве входных сигналов, может быть использована для решении задач распознавания с двумя классами, помеченными метками 1 или -1. В качестве активационной функции используется пороговая функция $\Phi(\xi) = \begin{cases} 1 & при\ \xi \geq 0 \\ -1 & при\ \xi < 0 \end{cases}$

Перцептрон Розенблатта

Особенностью модели Розенблатта является очень простая, но вместе с тем эффективная, процедура обучения, вычисляющая значения весовых коэффициентов $(w_0, \ldots, w_n)$ Сигналы, поступающие на вход перцептрона интерпретируются как входные признаки $(X_1, \ldots, X_n)$

На первом этапе производится преобразование векторов сигналов (признаковых описаний) для объектов обучающей выборки. В набор исходных признаков добавляется тождественно равная 1 нулевая компонента. Затем вектора описаний из класса K_2 умножаются на -1. Вектора описаний из класса K_1 не изменяются.

Перцептрон Розенблатта

Процедура обучения перцептрона. . Нулевое приближение вектора весовых коэффициентов $\mathbf{w}^{(0)} = (w_0^{(0)}, \ldots, w_n^{(0)})$

выбирается случайным образом

Преобразованные описания объектов обучающей выборки последовательно подаются на вход перцептрона. В случае если описание $\mathbf{x}^{(k)}$, поданное на k-ом шаге классифицируется неправильно, то происходит коррекция по правилу $\mathbf{w}^{(k)} = \mathbf{w}^{(k-1)} + \mathbf{x}^{(k)}$

В случае правильной классификации $\mathbf{w}^{(k)} = \mathbf{w}^{(k-1)}$

Перцептрон Розенблатта

Отметим, что правильной классификации всегда соответствует выполнение равенства $(\mathbf{w}^{(k-1)}, \mathbf{x}^{(k)}) \geq 0$

, а неправильной $(\mathbf{w}^{(k-1)}, \mathbf{x}^{(k)}) < 0$. Процедура повторяется до тех пор, пока не будет выполнено одно из следующих условий:

- достигается полное разделение объектов из классов K_1 и K_2;

- повторение подряд заранее заданного числа итераций не приводит к улучшению разделения;

- - оказывается исчерпанным заранее заданный лимит итераций.

-

Перцептрон Розенблатта

Справедлива следующая

Теорема. В случае, если описания объектов обучающей выборки линейно разделимы в пространстве признаковых описаний, то Процедура обучения перцептрона построит линейную гиперплоскость разделяющую объекты двух классов за конечное число шагов.

Многослойный перцептрон

Отсутствие линейной разделимости двух классов приводит к бесконечному зацикливанию Процедуры обучения перцептрона.

Существенно более высокой аппроксимирующей способностью обладают нейросетевые методы распознавания, задаваемые комбинациями связанных между собой нейронов. Таким методом является Многослойный перцептрон.

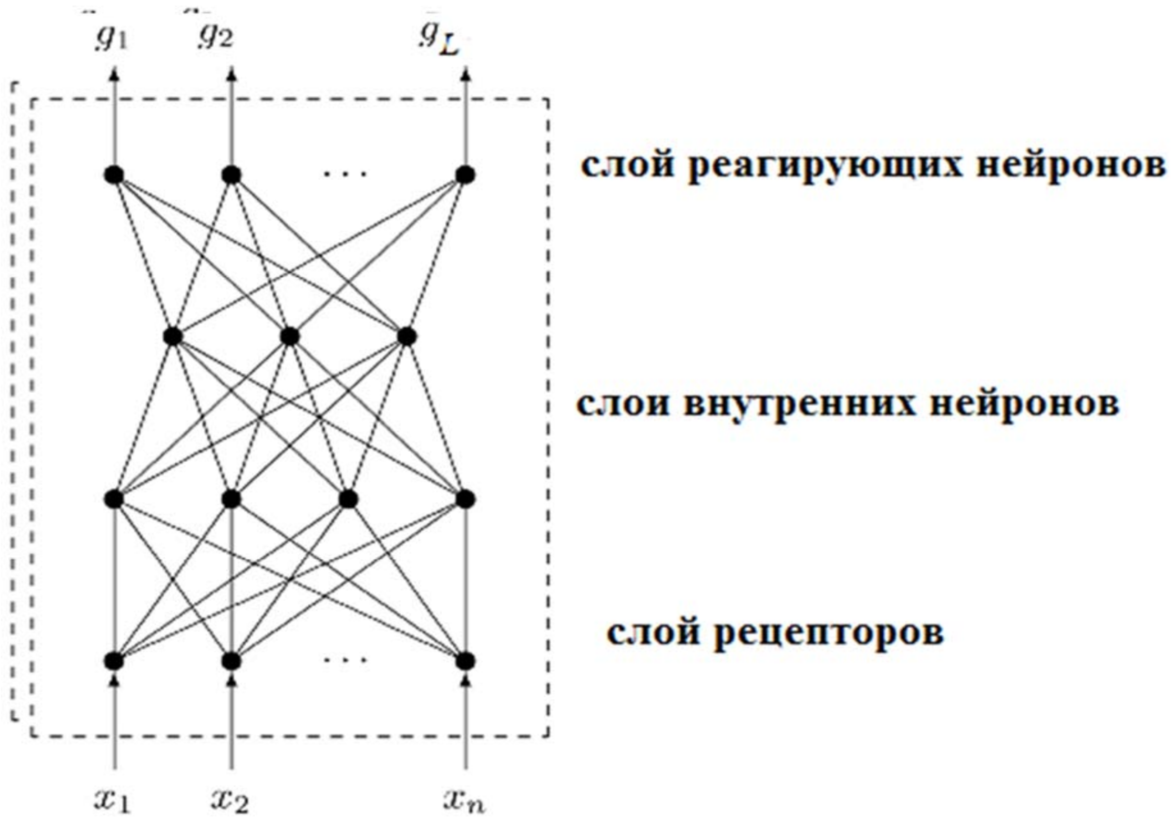
Многослойный перцептрон

В методе Многослойный перцептрон сеть формируется из нескольких слоёв нейронов.

В их число входит слой входных рецепторов, подающих сигналы на нейроны из внутренних слоёв. Слои внутренних нейронов. Слой реагирующих нейронов производит окончательную классификацию объектов на основании сигналов, поступающих от нейронов, принадлежащих внутренним слоям.

Многослойный перцептрон

Структура Многослойного перцептрона



Многослойный перцептрон

Обычно соблюдаются следующие правила формирования структуры сети.

Допускаются связи между только между нейронами, находящимися в соседних слоях.

Связи между нейронами внутри одного слоя отсутствуют.

Активационные функции для всех внутренних нейронов идентичны.

Распознавание с помощью многослойного перцептрона

Для решения задач распознавания с L классами $K_1, \dots, K_L$ используется конфигурация с L реагирующими нейронами. При этом сигналы $g_1, \dots, g_L$, вычисляемые на выходе реагирующих нейронов, интерпретируются как оценки за классы. Предполагается, что конфигурация нейронной сети задана. То есть заданы:

число слоев, *равное* $H+2$ *(входной, реагирующий и* H *внутренних);*

Распознавание с помощью многослойного перцептрона

Заданы также количества нейронов в каждом слое:

n – число входных нейронов-рецепторов;

r_h - число нейронов в *h*-ом внутреннем слое.

На первом этапе вектор рецепторы трансформируют вектор входных переменных (признаков) $X_1,\ldots,X_n$ в сигналы $u_1^0,\ldots,u_n^0$

.

Распознавание с помощью многослойного перцептрона

Предположим, что i-го нейрона 1-го внутреннего слоя связь с рецепторами осуществляется с помощью весовых коэффициентов $w_1^{i0},\ldots,w_n^{i0}$

Сумматор i-го нейрона первого внутреннего слоя вычисляет взвешенную сумму $\xi^{i1}=\sum_{t=0}^{n} w_t^{i0}u_t^0$. Сигнал на выходе i-го нейрона первого внутреннего слоя вычисляется по формуле

.
$$u_i^1=\Phi(\xi^{i1})$$

Аналогичным образом вычисляются сигналы на выходе нейронов второго внутреннего слоя.

.

Распознавание с помощью многослойного перцептрона

Сигналы $g_1, \ldots, g_L$ рассчитываются с помощью той же самой процедуры, которая используется при вычислении сигналов на выходе нейронов из внутренних слоёв. То есть при вычислении

g_i на первом шаге соответствующий сумматор вычисляет

взвешенную сумму $\xi^{ir} = \sum_{t=0}^{r_H} w_t^{iH} u_t^H$, где $w_1^{iH}, \ldots, w_{r_H}^{iH}$ - весовые коэффициенты, характеризующие связь *i* –го реагирующего нейрона с нейронами последнего *H*-го внутреннего слоя.

$u_1^H, \ldots, u_{r_H}^H$ - сигналы на выходе *H*-го внутреннего слоя.

Распознавание с помощью многослойного перцептрона

Сигнал на выходе i-го реагирующего нейрона вычисляется по формуле

$$g_i = \Phi(\xi^{ir})$$

Очевидно, что вектор выходных сигналов $(g_1, \dots, g_L)$ является функцией вектора входных признаков $\mathbf{x}$ и матрицы весовых коэффициентов, связывающих нейроны из разных слоёв - $\tilde{w}$.

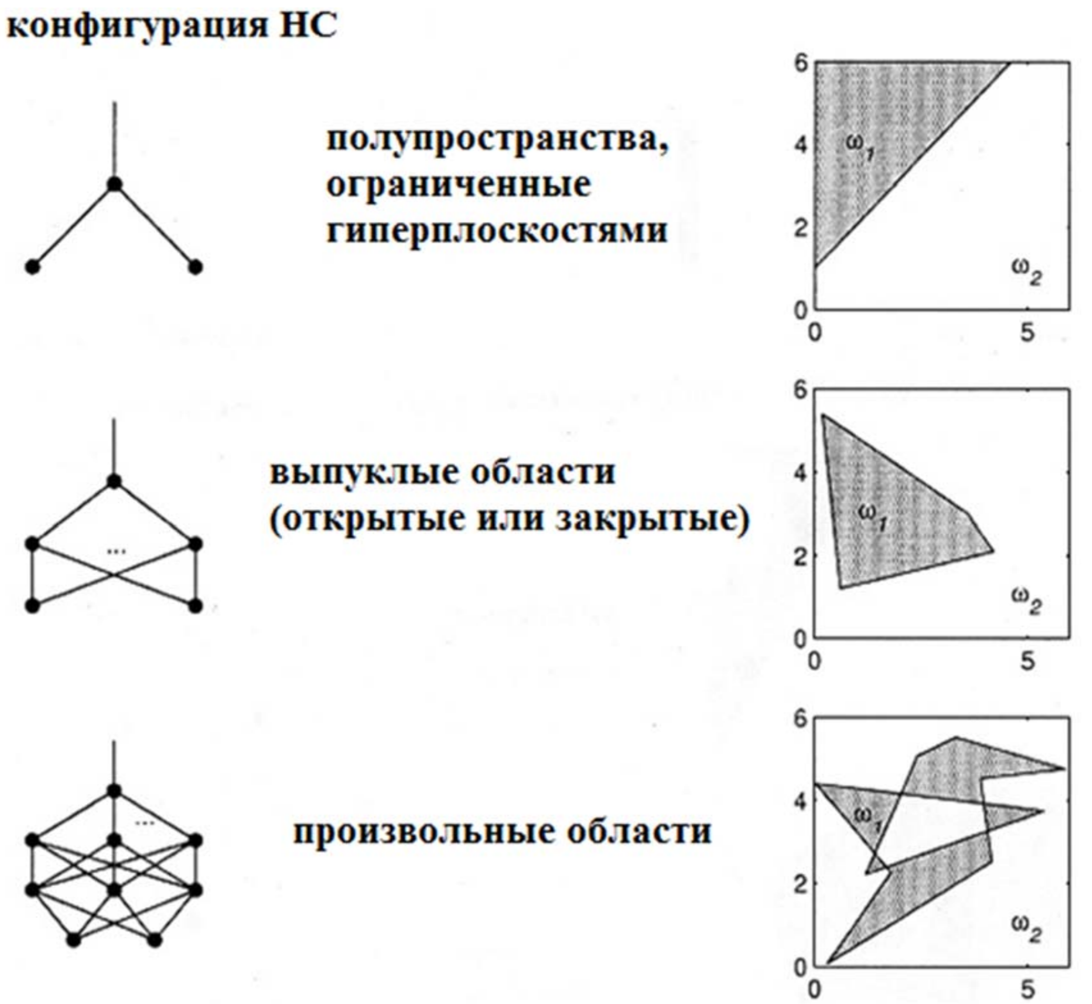
Аппроксимирующие способности Многослойных перцептронов

Один реагирующий нейрон позволяет аппроксимировать области, являющиеся полупространствами, ограниченными гиперплоскостями.

Нейронная сеть с одним внутренним слоем позволяет аппрокси-мировать произвольную выпуклую область в многомерном признаковом пространстве (открытую или закрытую).

Было доказано также, что МП с двумя внутренними слоями позволяет аппроксимировать произвольные области многомерного признакового пространства

Аппроксимирующие способности Многослойных перцептронов



Метод обратного распространения ошибки

Наиболее распространённым способом обучения нейросетевых алгоритмов является метод обратного распространения ошибки. Пусть на выходе i-го реагирующего нейрона g_i вычисляется оценка за класс K_i, принадлежащая отрезку [0,1] Обозначим через

$$\boldsymbol{\alpha}_* = (\alpha_{*1}, \ldots, \alpha_{*L})$$

вектор индикаторных функций классов $K_1, \ldots, K_L$ на объекте S_*. То есть $\alpha_{*i} = 1, \textbf{если } s_* \in K_i$

$$\alpha_{*i} = 0, \textbf{если } s_* \notin K_i$$

Метод обратного распространения ошибки

Качество аппроксимации на обучающей выборке

$\tilde{S}_t = \{(\boldsymbol{\alpha}_1, \mathbf{x}_1), \ldots, (\boldsymbol{\alpha}_m, \mathbf{x}_m)\}$ оценивается с помощью функционала

$$\tilde{E}(\tilde{S}_t, \tilde{w}) = \sum_{j=1}^{m} \sum_{t=1}^{L} [\alpha_{jt} - g_t(\mathbf{x}_j, \tilde{w})]^2$$

Где $\tilde{w}$ - множество весовых коэффициентов связей меду нейронами .

Обучение заключается в поиске значений коэффициентов из $\tilde{w}$. ,

при которых достигает минимума функционал $E(\tilde{S}_t, \tilde{w})$

Метод обратного распространения ошибки

В основе обучения лежит метод градиентного спуска. Метод градиентного спуска является итерационным методом оптимизации произвольного функционала F, зависящего от параметров $\theta_1, \ldots, \theta_n$ и дифференцируемого по каждому из параметров в произвольной точке $\mathbf{R}^n$. Новые значения вектора параметров на k-ой итерации $\boldsymbol{\theta}^{(k)}$ вычисляется через вектор $\boldsymbol{\theta}^{(k-1)}$, полученный на предыдущей итерации по формуле, $\boldsymbol{\theta}^{(k)} = \boldsymbol{\theta}^{(k-1)} + \eta * \mathbf{grad}(F)$ где $\eta > 0$ вещественный параметр, задающий размер каждого шага.

Метод обратного распространения ошибки

- На предварительном этапе обучения весовым коэффициентам из $\tilde{w}$ случайным образом присваиваются исходные значения. На обучение подаётся некоторый объект обучающей выборки $s_j = (\boldsymbol{\alpha}_j, \mathbf{x}_j)$, по описанию которого вычисляются входные и выходные сигналы внутренних нейронов сети, а также выходные сигналы реагирующих нейронов $g_1(\mathbf{x}_j, \tilde{w}), \dots, g_L(\mathbf{x}_j, \tilde{w})$

  Проведём коррекцию весовых коэффициентов связей i-го реагирующего нейрона с нейронами предшествующего внутреннего слоя: $(w_0^{iH}, \dots, w_{r(H)}^{iH})$.

  Для упрощения формул далее $g_1(\mathbf{x}_j, \tilde{w})$ будем обозначать $g_1(\mathbf{x}_j)$

Метод обратного распространения ошибки

От весовых коэффициентов $(w_0^{tH},\ldots,w_{r_H}^{tH})$ зависит только

компонента $[\alpha_{ji}-g_i(\mathbf{x}_j)]^2$ ошибки прогнозирования для

объекта s_j -

$$E(s_j,\tilde{w})=\sum_{i=1}^{L}[\alpha_{ji}-g_i(\mathbf{x}_j)]^2$$

Поэтому

$$\frac{\partial E(s_j,\tilde{w})}{\partial w_i^{tH}}=\frac{\partial E(s_j,\tilde{w})}{\partial g_t(\mathbf{x}_j)}\frac{\partial g_t(\mathbf{x}_j)}{\partial w_i^{tH}}=-2[\alpha_{jt}-g_t(\mathbf{x}_j)]\frac{\partial g_t(\mathbf{x}_j)}{\partial w_i^{th}}$$

Однако

$$\frac{\partial g_i(\mathbf{x}_j)}{\partial w_t^{iH}}=\frac{\partial g_i(\mathbf{x}_j)}{\partial \xi^{iH}}\frac{\partial \xi^{iH}}{\partial w_t^{iH}},$$

где

$$\xi^{iH}=\sum_{t=0}^{r_H}w_t^{iH}u_t^H$$

u_t^H - сигнал на выходе *t*-го нейрона слоя *H*.

Метод обратного распространения ошибки

Предположим, что g_i является сигмоидной функцией от ξ^{iH}. Тогда

$$\frac{\partial g_i(\mathbf{x}_j)}{\partial w_t^{iH}} = [1 - g_i(\mathbf{x}_j)]g_i(\mathbf{x}_j)u_t^H$$

. Таким образом $\dfrac{\partial E(s_j, \tilde{w})}{\partial w_t^{iH}} = \delta^{iH} u_t^H$

где $\delta^{iH} = \dfrac{\partial E(s_j, \tilde{w})}{\partial \xi^{iH}} = -2[\alpha_{ij} - g_i(\mathbf{x}_j)][1 - g_i(\mathbf{x}_j)]g_i(\mathbf{x}_j)$

Воспользовавшись методом градиентного спуска, запишем новые

значения весовых коэффициентов $w_t^{iH}(k)$, вычисляемые на

k-ой итерации в виде $w_t^{iH}(k) = \mathbf{w}_t^{iH}(k-1) + \eta\delta^{iH}u_t^H$

Метод обратного распространения ошибки

Рассмотрим теперь коррекцию весовых коэффициентов связей t-го нейрона из слоя *H* с нейронами предшествующего внутреннего слоя (*H-1*) – $(w_0^{t(H-1)}, \ldots, w_{r_{(H-1)}}^{t(H-1)})$

Вклад этих коэффициентов в величину ошибки осуществляется только через сигнал u_t^H на выходе . Поэтому

$$\frac{\partial E(s_j, \tilde{w})}{\partial w_i^{t(H-1)}} = \frac{\partial E(s_j, \tilde{w})}{\partial u_t^H} \frac{\partial u_t^H}{\partial w_i^{t(H-1)}}.$$

Однако .

$$\frac{\partial E(s_j, \tilde{w})}{\partial u_t^H} = \sum_{l=1}^{L} \frac{\partial E(s_j, \tilde{w})}{\partial \xi^{lH}} \frac{\partial \xi^{lH}}{\partial u_t^H}$$

Метод обратного распространения ошибки

Принимая во внимание, что $\dfrac{\partial E(s_j,\tilde{w})}{\partial \xi^{lH}} = \delta^{lH}$ $\dfrac{\partial \xi^{lH}}{\partial u_t^H} = w_t^{lH}$

получаем $\dfrac{\partial E(s_j,\tilde{w})}{\partial u_t^H} = (\sum_{l=1}^{L} \delta^{lH} w_t^{lH})$

Исходя из предположения

Нетрудно показать также, что $\dfrac{\partial u_i^H}{\partial w_i^{t(H-1)}} = u_t^H(1-u_t^H)u_i^{H-1}$

В итоге $\dfrac{\partial E(s_j,\tilde{w})}{\partial w_i^{t(H-1)}} = (\sum_{l=1}^{L} \delta^{lH} w_t^{lH})u_t^H(1-u_t^H)u_i^{H-1} = \delta^{t(H-1)}u_i^{H-1}$

Где $\delta^{t(H-1)} = \dfrac{\partial E(s_j,\tilde{w})}{\partial \xi^{t(H-1)}} = (\sum_{l=1}^{L} \delta^{lH} w_t^{lH})u_t^H(1-u_t^H)$

Метод обратного распространения ошибки

Воспользовавшись методом градиентного спуска, запишем новые значения весовых коэффициентов $w_t^{i(H-1)}(k)$, вычисляемые на k-ой итерации в форме

$$\mathbf{w}^{i(H-1)}(k) = \mathbf{w}^{i(H-1)}(k-1) + \eta \delta^{i(H-1)} u_t^{H-1}$$

В общем виде для весовых коэффициентов $(w_0^{t(H-h)}, \ldots, w_{r_{(H-1)}}^{t(H-h)}),\ h \geq 2$

$$\frac{\partial E(s_j, \tilde{w})}{\partial w_i^{t(H-h)}} = \frac{\partial E(s_j, \tilde{w})}{\partial u_t^{H-h+1}} \frac{\partial u_t^{H-h+1}}{\partial w_i^{t(H-h)}}$$

Метод обратного распространения ошибки

Однако

$$\frac{\partial E(s_j,\tilde{w})}{\partial u_t^{H-h+1}} = \sum_{l=1}^{r_{H-h+1}} \frac{\partial E(s_j,\tilde{w})}{\partial \xi^{l(H-h+1)}} \frac{\partial \xi^{l(H-h+1)}}{\partial u_t^{(H-h+1)}} = \sum_{l=1}^{r_{H-h+1}} \delta^{l(H-h+1)} w_t^{l(H-h+1)}$$

$$\frac{\partial u_i^{H-h+1}}{\partial w_i^{t(H-1)}} = u_t^{H-h+1}(1-u_t^{H-h+1})u_i^{H-h}$$

В итоге получаем $\frac{\partial E(s_j,\tilde{w})}{\partial w_i^{t(H-h)}} = \delta^{t(H-h)} u_i^{H-h}$, где

$$\delta^{t(H-h)} = \left(\sum_{l=1}^{r_{H-h+1}} \delta^{l(H-h+1)} w_t^{l(H-h+1)}\right) u_t^{H-h+1}(1-u_t^{H-h+1})$$

Метод обратного распространения ошибки

и коррекция согласно процедуре градиентного спуска производится по формуле:

$$\mathbf{w}^{i(H-h)}(k) = \mathbf{w}^{i(H-h)}(k-1) + \eta\delta^{i(H-h+1)}\mathbf{u}^{H-h} \qquad (1)$$

Таким образом может быть представлен общая схема метода обратного распространения ошибки для многослойного перцептрона.

На предварительном этапе выбирается архитектура сети:

задаётся число внутренних слоёв и количества нейронов в каждом слое.

Метод обратного распространения ошибки

Случайным образом задаются исходные весовые коэффициенты На вход многослойного перцептрона поочерёдно подаются векторные описания объектов обучающей выборки. С использованием описанного выше способа производится коррекция весовых коэффициентов. С использованием новых скорректированных весовых коэффициентов $\tilde{w}$ вычисляется значение функционала

$$\tilde{E}(\tilde{S}_t, \tilde{w})$$

Обучение заканчивается при выполнении одного из заранее заданных условий.

Метод обратного распространения ошибки

а) Величина функционала ошибки оказывается меньше порогового значения $\tilde{E}(\tilde{S}_t, \tilde{w}) < \varepsilon$;

б) общее число шагов (коррекций) превышает N_s;

г) Изменения функционала ошибки на протяжений нескольких последних итераций оказывается меньшим некоторого порогового значения.