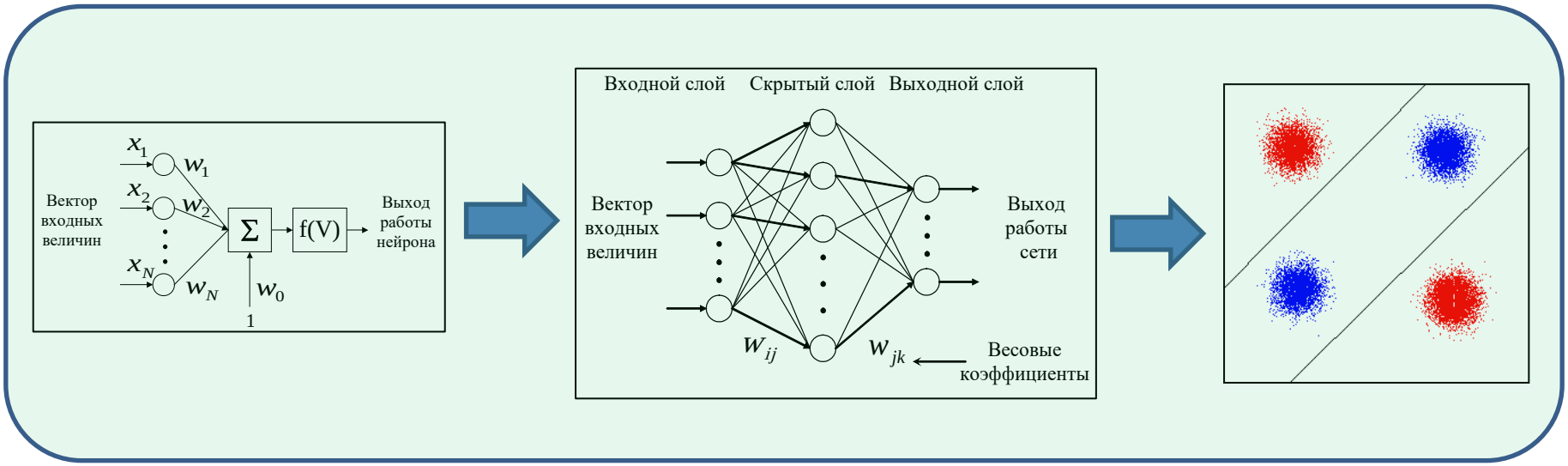
Методы машинного обучения

Лекция 7

Классификация - Нейросетевой подход

Предпосылки

- Логистическая регрессия
- Линейный дискриминантный анализ (линейный дискриминант Фишера)
- Метод градиентного спуска

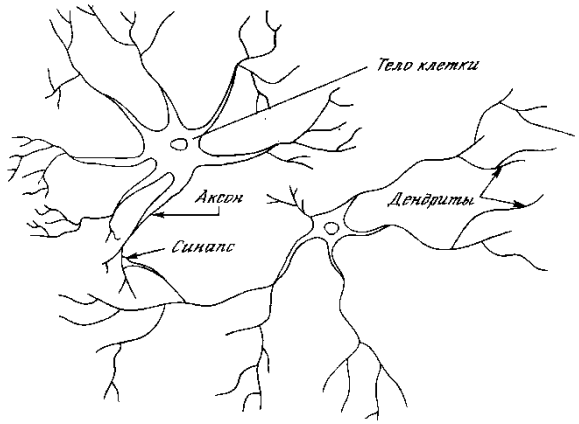


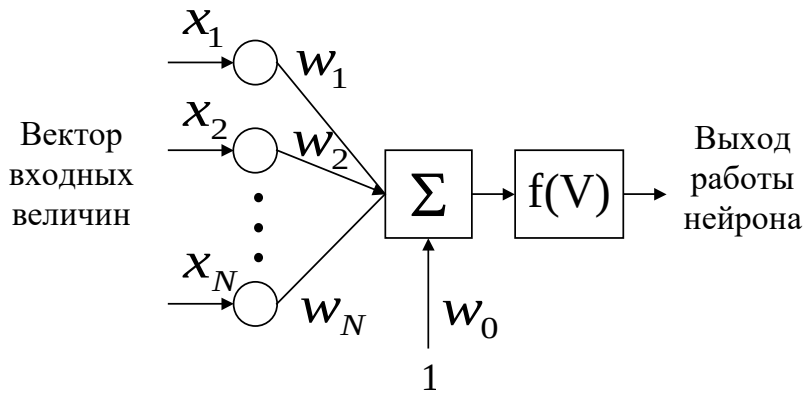
Логистическая регрессия и математическая модель нейрона

Логистическую регрессию можно представить в виде однослойной нейронной сети с сигмоидальной функцией активации, веса которой есть коэффициенты логистической регрессии.

Структурная схема нейрона:

$$V = \sum_{i=0}^{N} w_i \cdot x_i \qquad f(V) = \frac{1}{1+\exp(-bV)}$$





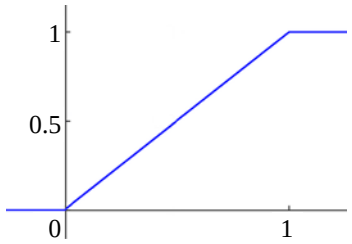
Первой попыткой математически описать процесс функционирования нейрона следует считать работу Мак-Каллока и Питтса, написанную ими еще в 1943 году. Отдельный нейрон состоит из “тела” (по аналогии с биологическим нейроном), суммирующего сигналы, поступающие с синапсов. При этом каждый синапс может передавать либо возбуждающий, либо тормозящий сигнал. В зависимости от соотношения возбуждающих и тормозящих сигналов нейрон либо возбуждается и передает дальше возбуждающий сигнал, либо тормозится и передает тормозящий сигнал.

Следующий шаг в описании процесса функционирования нейронных сетей внес Розенблатт, описавший функционирование персептрона (1957). Персептрон Розенблатта представляет полноценную математическую модель отдельного нейрона.

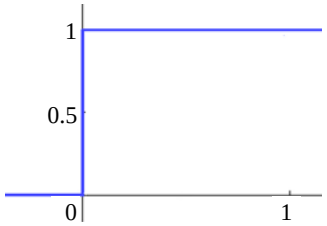
Функции активации

- Линейная: $f(V) = V$

- Линейная с насыщением:
$$f(V) = \begin{cases} 0, & V \le 0 \\ 1, & V \ge 1 \\ V, & 0 < V < 1 \end{cases}$$

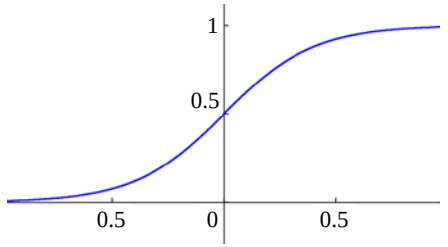


- Ступенчатая (пороговая):
$$f(V) = \begin{cases} 1, V \ge 0 \\ 0, V < 0 \end{cases}$$

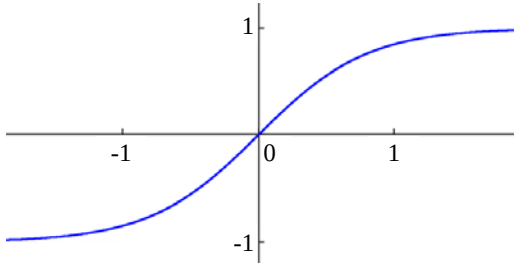


- Многопороговая

- Сигмоидная (логистическая):
$$f(V) = \frac{1}{1 + \exp(-bV)}$$



- Гиперболический тангенс :
$$f(V) = \tanh(V) \equiv \frac{\exp(bV) - 1}{\exp(bV) + 1}$$



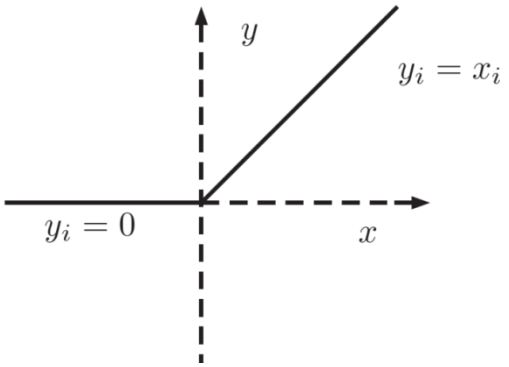
Семейство функций активации ReLU

- **ReLU (Rectified linear unit)**

  $f(x) = max(0, x)$

  ✓Отсутствие эффекта насыщения;

  ✓*«Dying ReLu Problem» (проблема «умирающего» ReLu),* до 40% ReLU нейронов никогда не активируются.

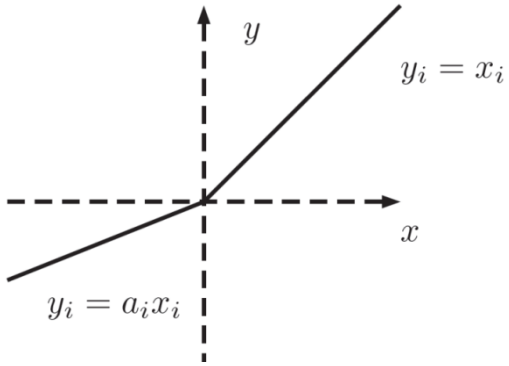


- **Leaky ReLU** - ReLU с «утечкой»

  для $x < 0$ имеет небольшое отрицательное значение, определяемое малым угловым коэффициентом:

$$f(x) = \begin{cases} x & x \geq 0 \\ \alpha x & x < 0 \end{cases}$$

  где α – малая константа порядка 0,01.

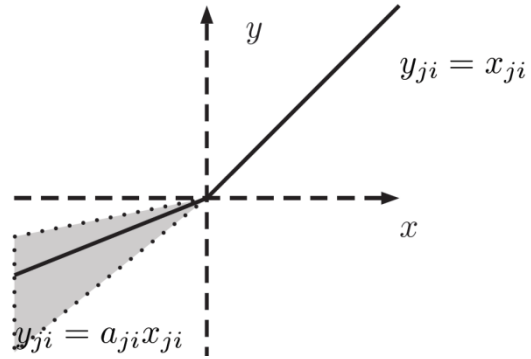


- **Parametric ReLU**

  (Parameteric rectified linear unit, PReLU)

- **Randomized ReLU**

  (Randomized leaky rectified linear unit)



Искусственная нейронная сеть (Мак-Каллок и Питтс)

Мак-Каллок и Питтс исходили из предположений о том, что нервная система биологических существ состоит из большого количества нейронов, каждый из которых имеет несколько входных и один выходной сигнал. Описывая процесс функционирования нейронной сети, они приняли следующие допущения (1943 год):

- активность нейрона удовлетворяет принципу “все или ничего”;
- возбуждению нейрона предшествует период накопления возбуждений фиксированного количества его синапсов, причем это число не зависит от предыдущей активности и расположения синапсов в нейроне;
- единственным запаздыванием в нервной системе, имеющим значение, является синаптическая задержка;
- активность какого-либо тормозящего синапса исключает возбуждение нейрона в рассматриваемый момент времени;
- с течением времени структура нейронной сети не меняется.

Поскольку нейрон имеет как входные, так и выходные сигналы, то различные нейроны могут быть соединены между собой, образуя нейронную сеть.

Внутри биологической клетки сигнал распространяется гораздо медленнее, чем в электронных схемах. Однако, биологическая нейросеть в совокупности оказывается высокоэффективной в решении множества сложных задач, таких как распознавание образов (зрение, речь и т.д.). Можно предположить, что причина способностей кроется в высокой организации связей и параллелизме.

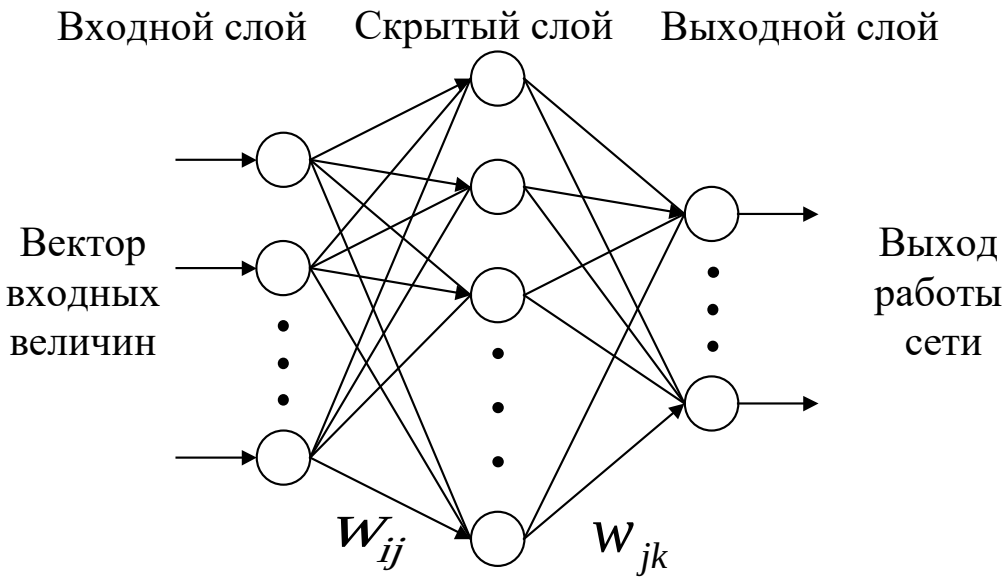
Искусственная нейронная сеть
Artificial neural network (ANN)

Математическая модель, а также ее программные или аппаратные реализации, построенная в некотором смысле по образу и подобию сетей нервных клеток живого организма.

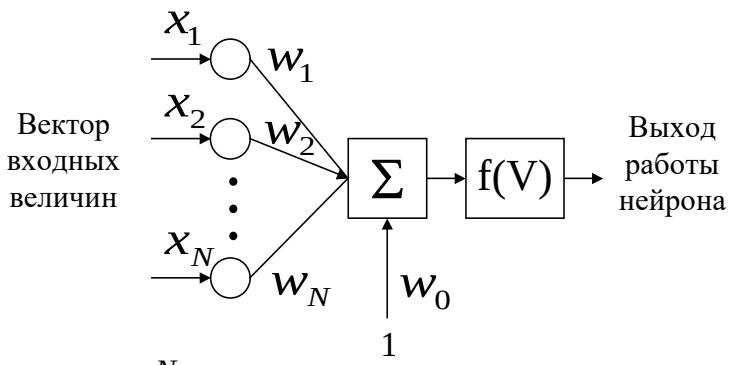
Под искусственной нейронной сетью в дальнейшем будем понимать сеть искусственных нейронов, соединенных между собой. Здесь предполагается, что нейроны могут соединяться между собой произвольным образом и образовывать таким образом разнообразные нейронные структуры.

- С точки зрения машинного обучения, нейронная сеть представляет собой частный случай методов распознавания образов, дискриминантного анализа;
- С точки зрения математики, обучение нейронных сетей — это многопараметрическая задача нелинейной оптимизации;
- С точки зрения кибернетики, нейронная сеть используется в задачах адаптивного управления и как алгоритмы для робототехники;
- С точки зрения развития вычислительной техники и программирования, нейронная сеть — способ решения проблемы эффективного параллелизма;
- С точки зрения искусственного интеллекта, ИНС является основой философского течения коннекционизма и основным направлением в структурном подходе по изучению возможности построения (моделирования) естественного интеллекта с помощью компьютерных алгоритмов.

Искусственная нейронная сеть
Многослойный персептрон



Структурная схема нейрона:



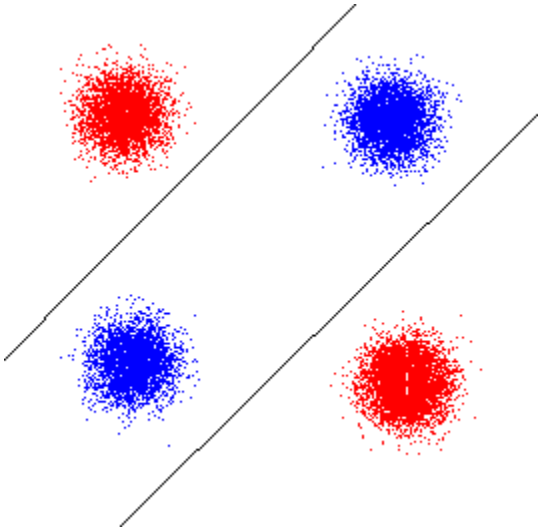
$$V = \sum_{i=0}^{N} w_i \cdot x_i \qquad f(V) = \frac{1}{1+\exp(-bV)}$$

Функционирование: $$Y_k(x) = Y_k(x_1,\ldots x_l) = f\left(\sum_{j=0}^{m} w_{jk}\, f\left(\sum_{i=0}^{n} w_{ij} x_i\right)\right)$$

Сигналы проходят от входного слоя нейронов через скрытые слои к выходным элементам. В процессе функционирования каждый узел многослойной сети проектирует свой входной вектор на вектор весов посредством скалярного произведения (т.е. вычисляет взвешенную сумму входного вектора). После этого результаты проекции подвергаются нелинейным преобразованиям. Их цель - усилить те характеристики, за которые отвечает соответствующий узел.

Многослойный персептрон – число слоев

Марвин Минский и Сеймур Паперт в своей работе "Персептроны" доказали, что простейшие однослойные нейронные сети, состоящие из входного и выходного слоев (известные, как линейный персептрон), способны решать только линейно разделимые задачи и обеспечивают универсальную линейную аппроксимацию. Это ограничение преодолимо путем добавления скрытых слоев и использования многослойных нейронных сетей.



В общем виде можно сказать, что при решении задач классификации в сети с одним скрытым слоем, входной вектор преобразуется в некоторое новое пространство, возможно с другой размерностью, а затем поверхности, соответствующие нейронам выходного слоя, разделяют его на классы. Таким образом, сеть распознает не только характеристики исходных данных, но и "характеристики характеристик", сформированные скрытым слоем.

Нейронные сети с числом скрытых слоев большим единицы называются глубокими (deep neural network). Они могут содержать меньшее число нейронов в каждом слое, чем сети с одним скрытым слоем, реализующие то же самое отображение. Однако строгой методики сопоставления таких сетей пока не существует.

Многослойный персептрон
число нейронов в каждом слое

Количество нейронов входного слоя напрямую зависит от размерности исходного пространства входных данных (пространства признаков) и от методов их кодирования.

Количество нейронов скрытого слоя

Проблема выбора количества скрытых элементов многослойного персептрона заключается в том, что с одной стороны, число скрытых элементов должно быть достаточным для решения поставленной задачи, а с другой не должно быть слишком большим, чтобы обеспечить необходимую обобщающую способность сети и избежать переобучения. То есть, нельзя просто выбрать теоретический максимум числа весовых коэффициентов, так как в этом случае сеть научится иметь дело только с теми данными, которые предъявлялись в процессе тренировки, и, поэтому обобщающая способность сети будет слабой.

Количество нейронов выходного слоя зависит от решаемой задачи и, также как для входного слоя, от способов кодирования

Многослойный персептрон число нейронов скрытого слоя

Из теоремы Колмагорова-Арнольда (1957) о представлении непрерывных функций нескольких переменных в виде суперпозиций непрерывных функций одного переменного и сложения следует теорема Хехт-Нильсена доказывающая представимость функции многих переменных достаточно общего вида с помощью двухслойной нейронной сети с прямыми полными связями с *i* нейронами входного слоя и *(2i+1)* нейронами скрытого слоя с заранее известными ограниченными функциями активации (например, сигмоидными). Теорема, таким образом, в неконструктивной форме доказывает решаемость задачи представления функции произвольного вида на нейронной сети и указывает минимальные числа нейронов, необходимых для решения.

Следствие 1. Неизвестными остаются следующие характеристики функций активации нейронов:

- Ограничение области значений (координаты асимптот) сигмоидных функций активации нейронов скрытого слоя;
- Наклон сигмоид;
- Вид функций активации нейронов выходного слоя.

Следствие 2.

Для любого множества пар входных-выходных векторов произвольной размерности $\{(X_k, Y_k), k = 1...N\}$ существует однородная двухслойная нейронная сеть с последовательными связями, с сигмоидными передаточными функциями и с конечным числом нейронов, которая для каждого входного вектора X_k формирует соответствующий ему выходной вектор Y_k.

Многослойный персептрон число нейронов скрытого слоя - оценки

- Сеть с одним скрытым слоем, содержащим *h* нейронов со ступенчатой функцией активации, способна осуществить произвольную классификацию *h*d* точек *d*-мерного пространства (т.е. классифицировать *h*d* примеров). Более того, одного скрытого слоя нейронов с сигмоидной функцией активации достаточно для аппроксимации любой границы между классами или некоторой функции со сколь угодно высокой точностью

- Хайкин, используя результаты из работ Баума и Хесслера, приводит рекомендации относительно размеров набора учебных данных относительно количества весовых коэффициентов с учетом доли ошибок, допустимых в ходе тестирования, которые можно выразить следующим неравенством: $p \geq \frac{w}{\varepsilon}$

  где ε - доля ошибок, допустимых в ходе тестирования. Так при допустимости 10% ошибок число учебных образцов должно быть в 10 раз больше числа имеющихся в сети весовых коэффициентов.

- Одной из возможных оценок может служить принцип совместной оптимизации эмпирической ошибки и сложности модели, согласно которому ошибка предсказаний сети на новых данных определяется общей длиной описания данных с помощью модели вместе с описанием самой модели. Данный принцип может быть записан в виде:

  min{ описание ошибки + описание модели }.

  Минимум ошибки (знак равенства) достигается при оптимальном числе весов в сети: $w \sim \sqrt{p \cdot i}$

  Что соответствует числу нейронов в скрытом слое равному по порядку величине: $h = \frac{w}{i} \sim \sqrt{\frac{p}{i}}$

  где *i, h* – число нейронов входного и скрытого слоев, *w* – количество весов, *p* – количество примеров обучающей выборки.

Многослойный персептрон
число нейронов скрытого слоя - оценки

Для оценки числа нейронов в скрытом слое однородной нейронной сети можно воспользоваться формулой для оценки необходимого числа синаптических весов N_w в многослойной сети с сигмоидальными функциями активации:

$$\frac{N_y N_p}{1+\log_2(N_p)} \le N_w \le N_y\left(\frac{N_p}{N_x}+1\right)\left(N_x+N_y+1\right)+N_y$$

где N_y - размерность выходного сигнала, N_p - число элементов обучающей выборки, N_x - размерность входного сигнала.

Оценив необходимое число весов, можно рассчитать число нейронов в скрытых слоях. Число нейронов в сети с одним скрытым слоем составит:

$$N = \frac{N_w}{N_x + N_y}$$

Пример: По условию задачи размерность входного вектора равна N_x =21; число нейронов в выходном слое соответствует числу классов N_y =5), на которые предполагается разбить выборку данных, состоящую из 6000 объектов. Неизвестным является число нейронов в промежуточном слое N .

$$\frac{5 \cdot 6000}{1+12{,}55} \le N_w \le 5 \cdot \left(\frac{6000}{21}+1\right) \cdot (21+5+1)+5$$

Округляя до целых, получим $2222 \le N_w \le 7746$. Тогда число нейронов в скрытом слоем составит: $N_{min} = \frac{2222}{26} \approx 85$ и $N_{max} = \frac{7746}{26} \approx 297$ т.е. $85 \le N \le 297$.

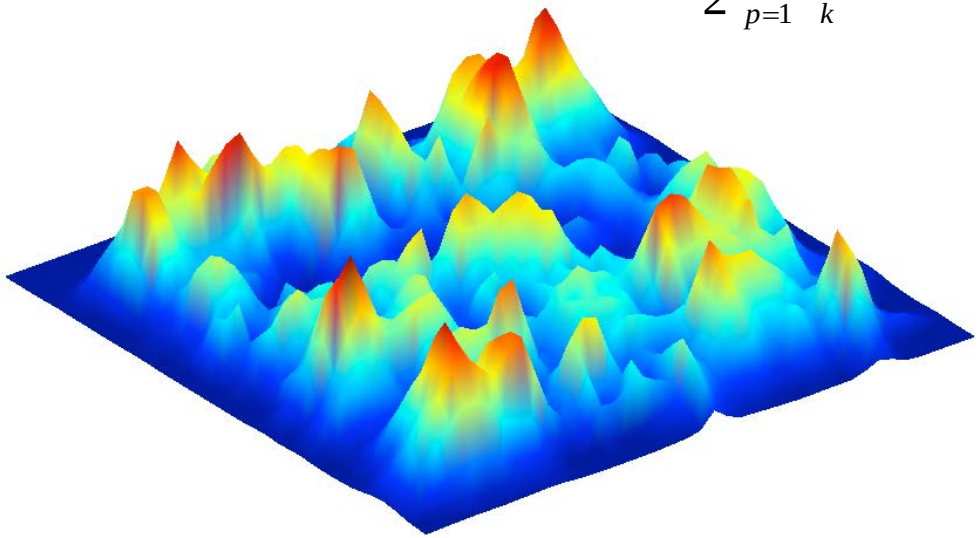
Многослойный персептрон (методы обучения)

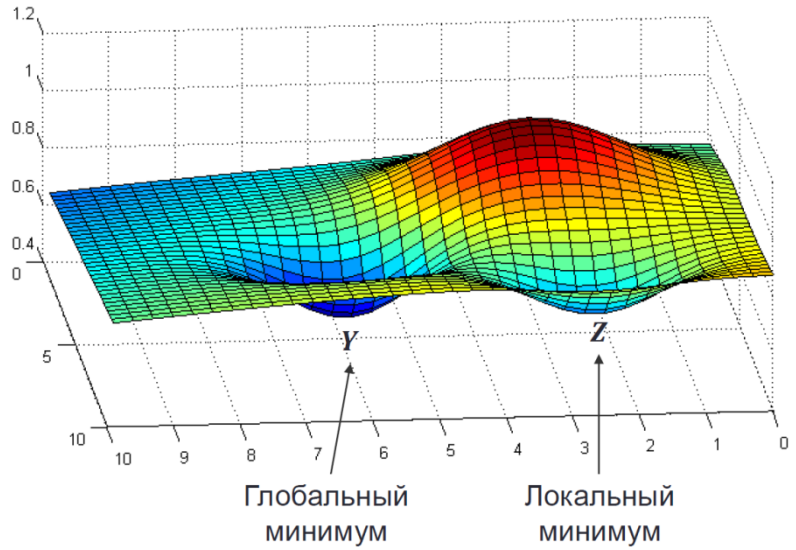
Обучение нейронной сети - интерактивный процесс корректировки синаптических весов и порогов. В процессе обучения нейронная сеть получает и обобщает знания об окружающей среде и тех данных, с которыми ей придется оперировать. Существуют два концептуальных подхода к обучению нейронных сетей: обучение с учителем и обучение без учителя.

Обучение персептрона:

- Обучение Хебба (без учителя);
- Стохастические методы обучения (“имитация отжига”);
- Обратное распространение ошибки (Backpropagation).

Ошибка сети при обучении: $E = \frac{1}{2}\sum_{p=1}^{P}\sum_{k}\left(u_p^k - y_p^k\right)^2$





Обучение Хебба

Канадский нейропсихолог Дональд Хебб сформулировал свой «нейрофизиологический постулат» в 1949 в самом значимом своем труде – *«The Organization of Behavior: A NEUROPSYCHOLOGICAL THEORY».*

Звучит следующим образом:

«Если аксон клетки А находится достаточно близко, чтобы активировать клетку В, и неоднократно или постоянно принимает участие в ее активации, то наблюдается некоторый процесс роста связи или метаболических изменений в одной или обеих клетках, ведущий к увеличению эффективности А, как одной из клеток активирующих В».

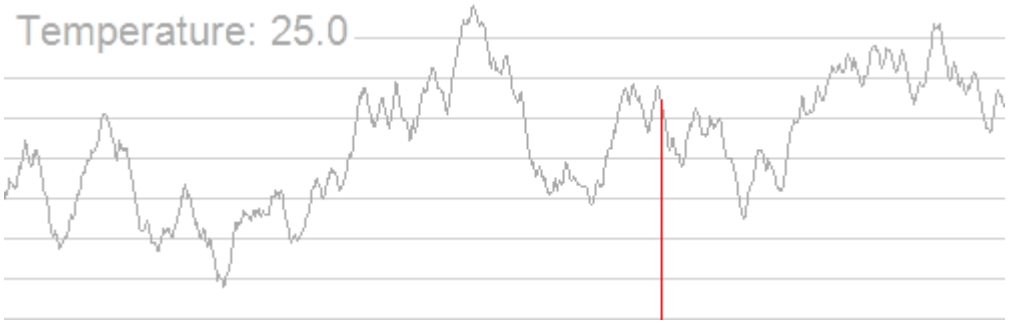
Предложена модель обучения без учителя, в которой синаптическая сила (вес) возрастает, если активированы оба нейрона, источник и приемник. Таким образом, часто используемые пути в сети усиливаются и феномен привычки и обучения через повторение получает объяснение.

$$w_{ij}(n+1) = w_{ij}(n) + \eta * OUT_i * OUT_j$$

$w_{ij}(n)$ – значение веса от нейрона *i* к нейрону *j* до подстройки, $w_{ij}(n+1)$ – значение веса от нейрона *i* к нейрону *j* после подстройки, η – коэффициент скорости обучения, OUT_i – выход нейрона *i* и вход нейрона *j*, OUT_j – выход нейрона *j*.

Стохастические методы обучения «имитация отжига»

Выполняются псевдослучайные изменения величин весов, сохраняя те изменения, которые ведут к улучшениям.



Для обучения сети может быть использована следующая процедура:

1. Выбрать значения весов случайным образом. Предъявить множество входов и вычислить получающиеся выходы.
2. Сравнить полученные значения с желаемыми выходами и вычислить величину разности между ними (сумма квадратов разностей). Целью обучения является минимизация этой разности (часто называют ***целевой функцией***).
3. Выбрать случайный весовой коэффициент и подкорректировать его на небольшое случайное значение. Если коррекция помогает (уменьшает целевую функцию), то сохранить ее, в противном случае вернуться к первоначальному значению веса. Случайное значение корректировки постепенно уменьшается со временем.
4. Повторять шаг 3 до тех пор, пока сеть не будет обучена в достаточной степени.

Скорость уменьшения температуры должна быть обратно пропорциональна логарифму времени, чтобы была достигнута сходимость к глобальному минимуму.

$$T(t) = \frac{T_0}{\log(1+t)}$$

T(t) – искусственная температура как функция времени;
T0 – начальная искусственная температура;
t – искусственное время.

Обратное распространение ошибки (Backpropagation)

Алгоритм обучения с учителем. Фактически является алгоритмом градиентного спуска, минимизирующим суммарную квадратичную ошибку:

$$E = \frac{1}{2}\sum_{k=1}^{P}\sum_{i}\left(d_k^i - y_k^j\right)^2$$

где d_k^i - желаемы выход нейрона *i*,

y_k^j - текущий выход нейрона *i* последнего слоя.

Синаптические веса настраиваются в соответствии с формулой:

$$w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij}(t)$$

изменение веса: $\Delta w_{ij} = -\varepsilon\frac{\partial E_k}{\partial w_{ij}} = -\varepsilon\delta_k^j x_k^i$

где ε - длина шага в направлении, обратном к градиенту;

δ_k^i - вычисляется через аналогичные множители из последующего слоя, и ошибка передается в обратном направлении.

Для выходных элементов: $\delta_k^j = -\left(d_k^j - y_k^j\right)f'(V_k^j) = -(d_k^j - y_k^j)\cdot y_k^j\cdot(1-y_k^j)$

Для скрытых элементов: $\delta_k^j = -f'(V_k^j)\sum_h w_{jh}\delta_k^h = out_k^j\cdot(1-out_k^j)\cdot\sum_h w_{jh}\delta_k^h$

где индекс *h* пробегает номера всех нейронов, на которые воздействует *j*-й нейрон, out_k^j - выход нейрона *j*, *k* - номер обучающего примера.

Этапы Backpropagation

Весь алгоритм разбивается на следующие этапы:

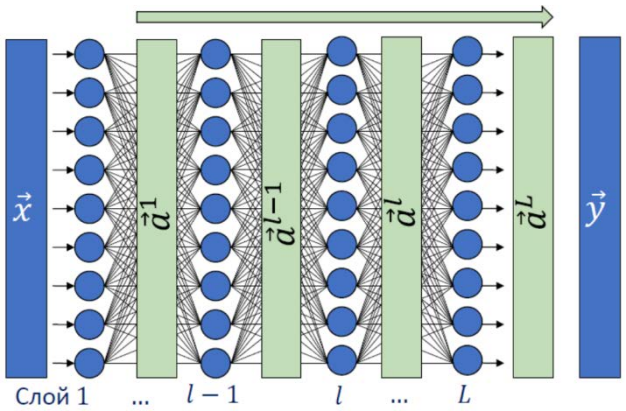
1. подать на вход сети один из требуемых образов и определить значения выходов нейронов сети .
2. рассчитать δ_k^j для выходного слоя сети по формуле

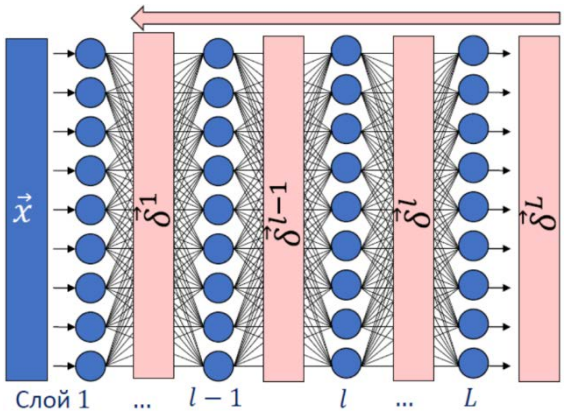
$$\delta_k^j = -\left(d_k^j - y_k^j\right) f'(V_k^j) = -(d_k^j - y_k^j) \cdot y_k^j \cdot (1 - y_k^j)$$

   и рассчитать изменения весов Δw_{ij} выходного слоя.
3. Рассчитать по формуле

$$\delta_k^j = -f'(V_k^j) \sum_h w_{jh} \delta_k^h = out_k^j \cdot (1 - out_k^j) \cdot \sum_h w_{jh} \delta_k^h$$

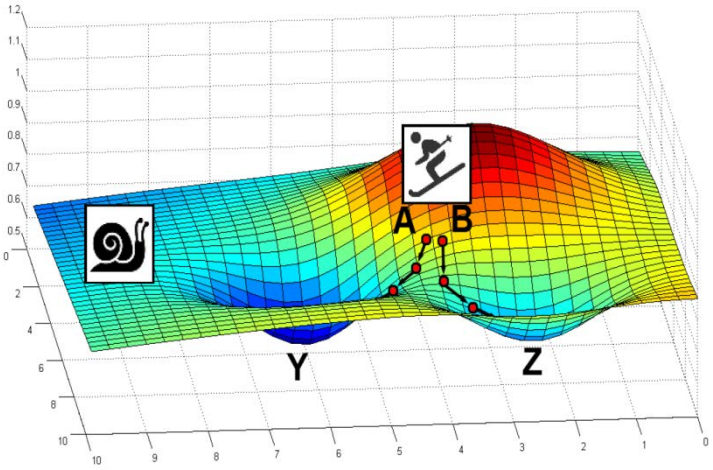
   значения δ_k^j и Δw_{ij} для остальных слоев нейросети, двигаясь от выходного слоя к входному.
4. Скорректировать все веса сети. Если ошибка существенна, то перейти на шаг 1.





Модификации алгоритма Backpropagation

Существуют различные модификации алгоритма Backpropagation призванные ускорить процесс обучения, особенно в случаях, когда рельеф функции ошибки напоминает не яму, а длинный овраг, обеспечить способность преодолевать мелкие локальные минимумы ошибки, застревая лишь в относительно глубоких, значимых минимумах, а так же избежать и некоторые другие его недостатки.



К таким модификациям можно отнести:

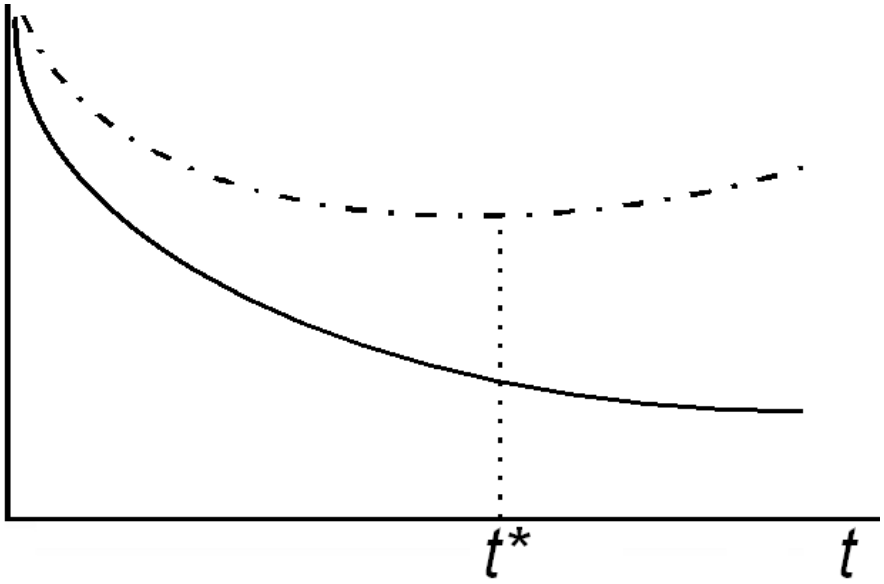
- использование момента μ, учитывающего изменение весов на предыдущем шаге обучения:

$$\Delta w_{ij}(t) = -\varepsilon \delta_j x_i + \mu \Delta w_{ij}(t-1)$$

- Алгоритм Resilient Propagation (RPROP) (resilient - эластичный), предложенный М. Ридмиллером (M.Riedmiller) и Г. Брауном (H.Braun), стремится избежать замедления темпа обучения на плоских “равнинах” ландшафта функции ошибки, характерного для схем, где изменения весов пропорциональны величине градиента.

Существуют и другие алгоритмы, ускоряющие процесс обучения, и важно иметь возможность выбрать тот алгоритм, который бы показывал наилучшую сходимость для имеющейся выборки.

Оценка полученной архитектуры



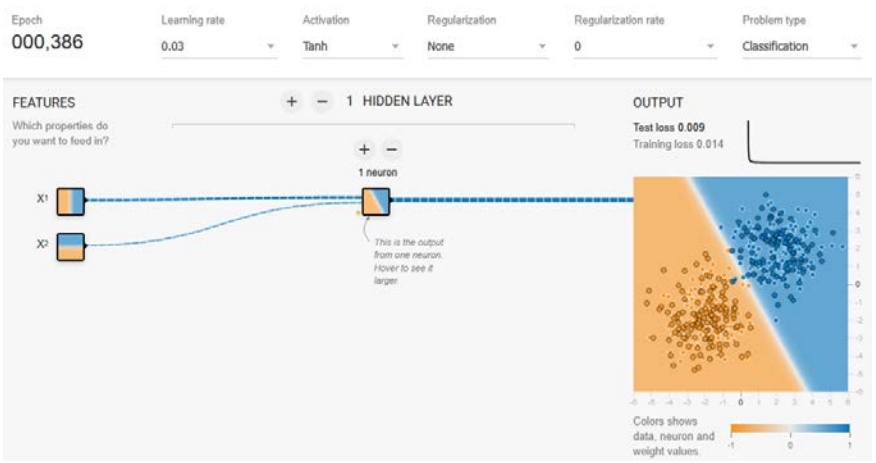
Показан момент остановки, соответствующий минимуму ошибки валидации (штрих-пунктирная кривая), при этом ошибка обучения (сплошная кривая) продолжает понижаться.

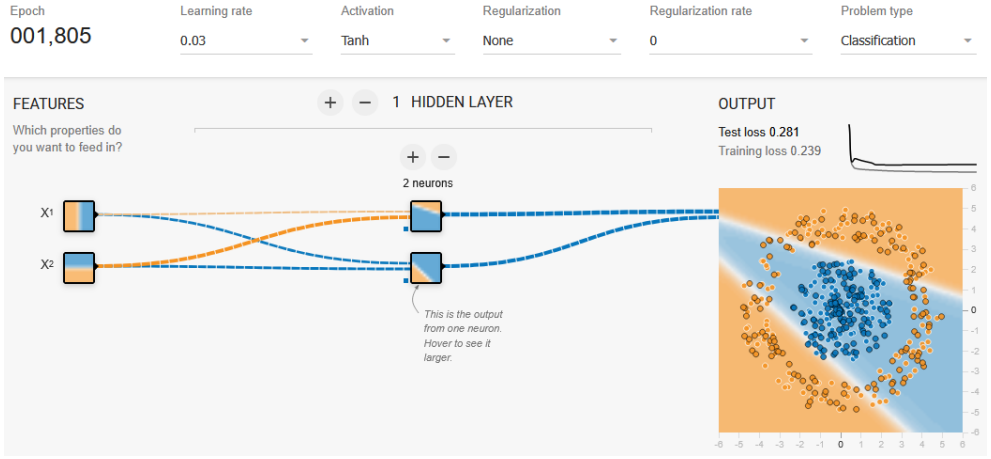
Критерий остановки обучения в момент времени *t**.

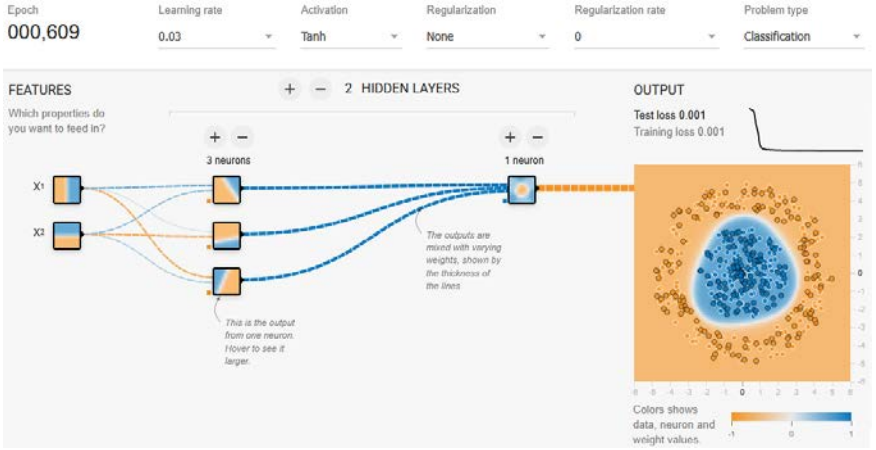
В упрощенном виде для оценки уже построенной нейросетевой модели могут быть сформулированы следующие критерии:

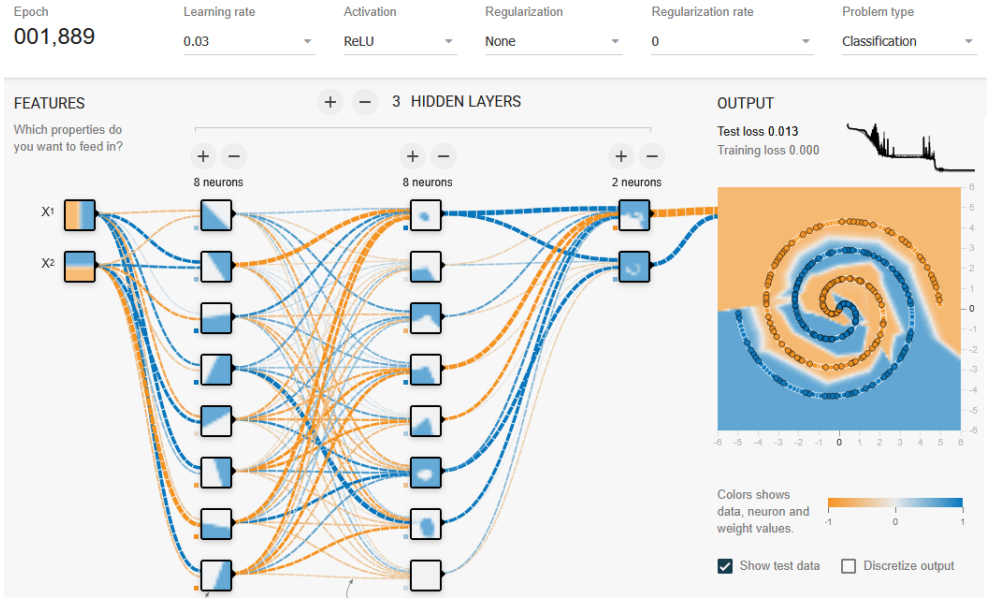
- если ошибка тренировки мала, а ошибка тестирования велика, значит, сеть содержит слишком много весовых коэффициентов;
- если и ошибка тренировки, и ошибка тестирования велики, значит весовых коэффициентов слишком мало;
- если все весовые коэффициенты очень большие, значит весовых коэффициентов слишком мало.

https://playground.tensorflow.org

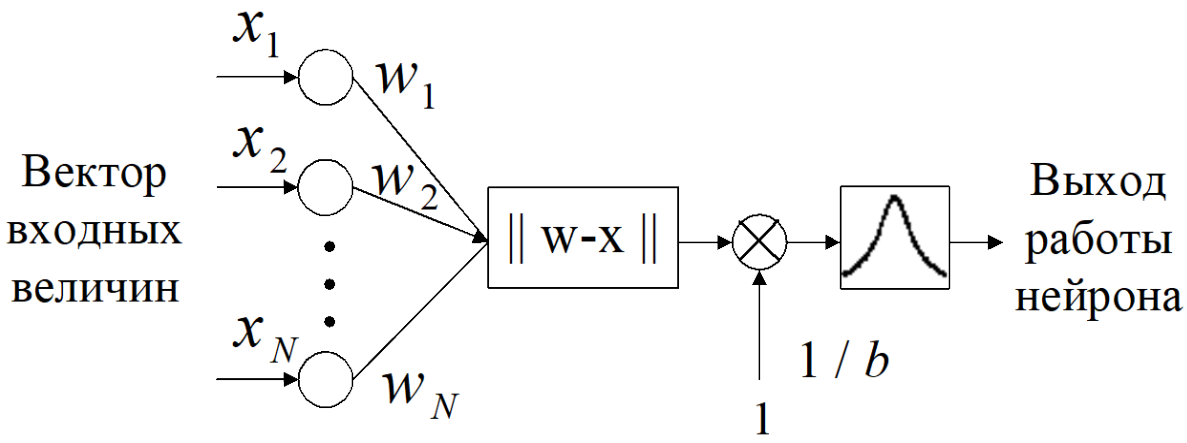








RBF-нейрон

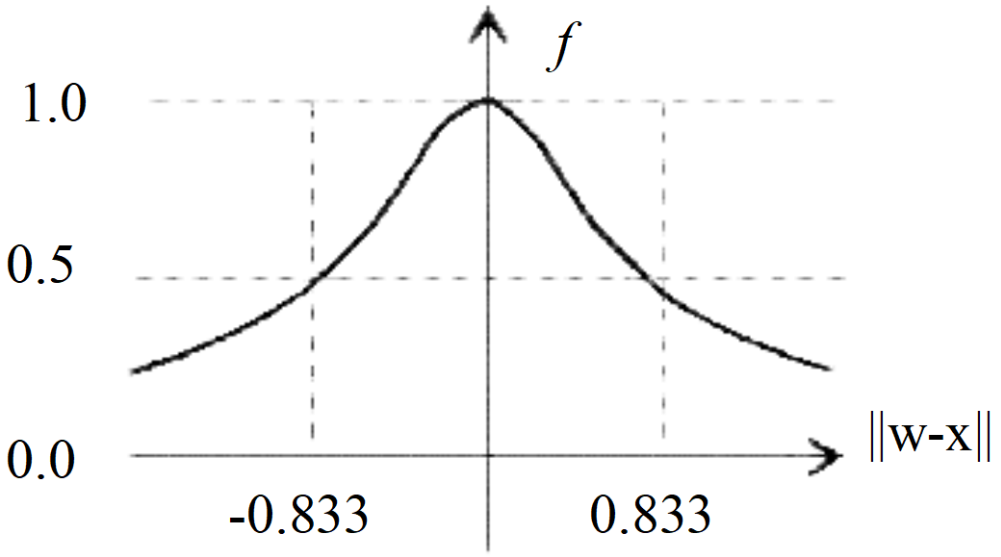


Потенциал нейрона - расстояние между векторами весовых коэффициентов и входных величин (евклидово расстояние)

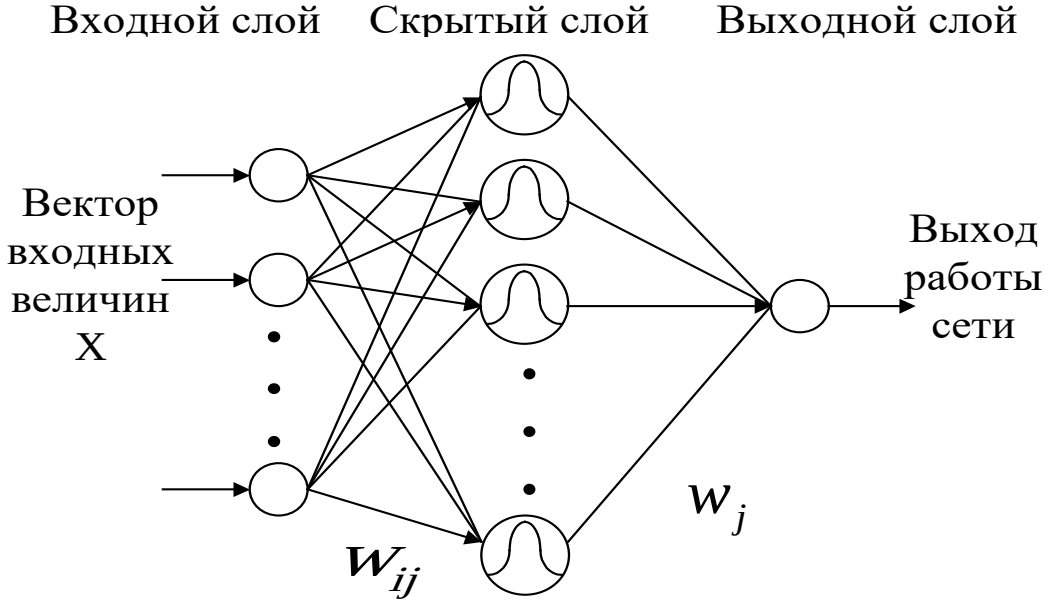
$$\|w - x\| = \sqrt{\sum_{i=1}^{N} (w_i - x_i)^2}$$

Функция активации:

$$f = e^{-\left(\frac{\|w-x\|}{b}\right)^2}$$



RBF-сеть

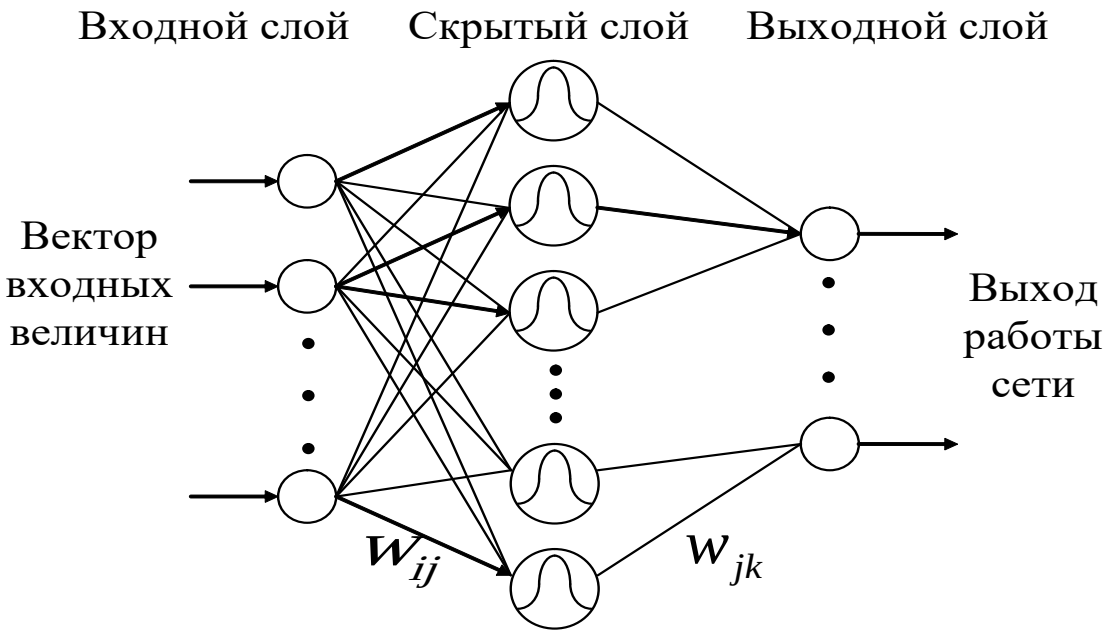


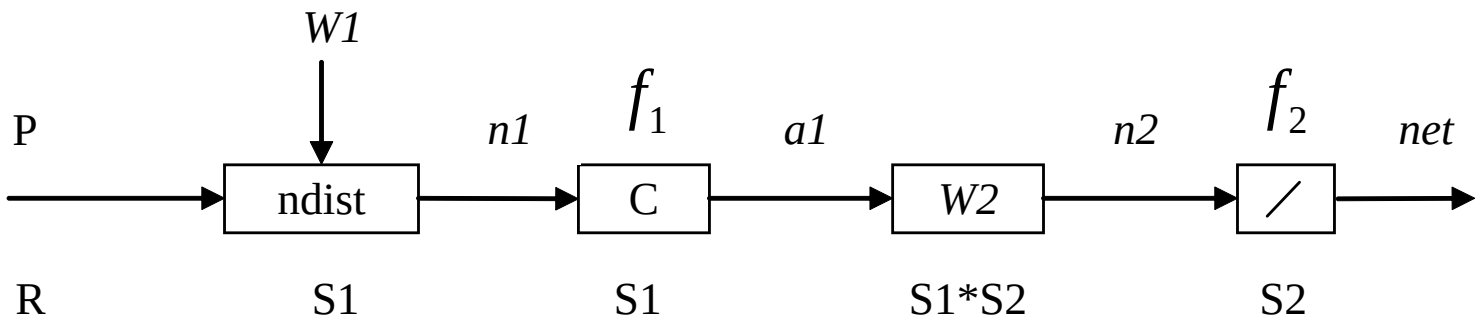
Функционирование: $u_{net} = \sum_{j=1}^{n} w_j \cdot a_j$

Обучение RBF-сети:

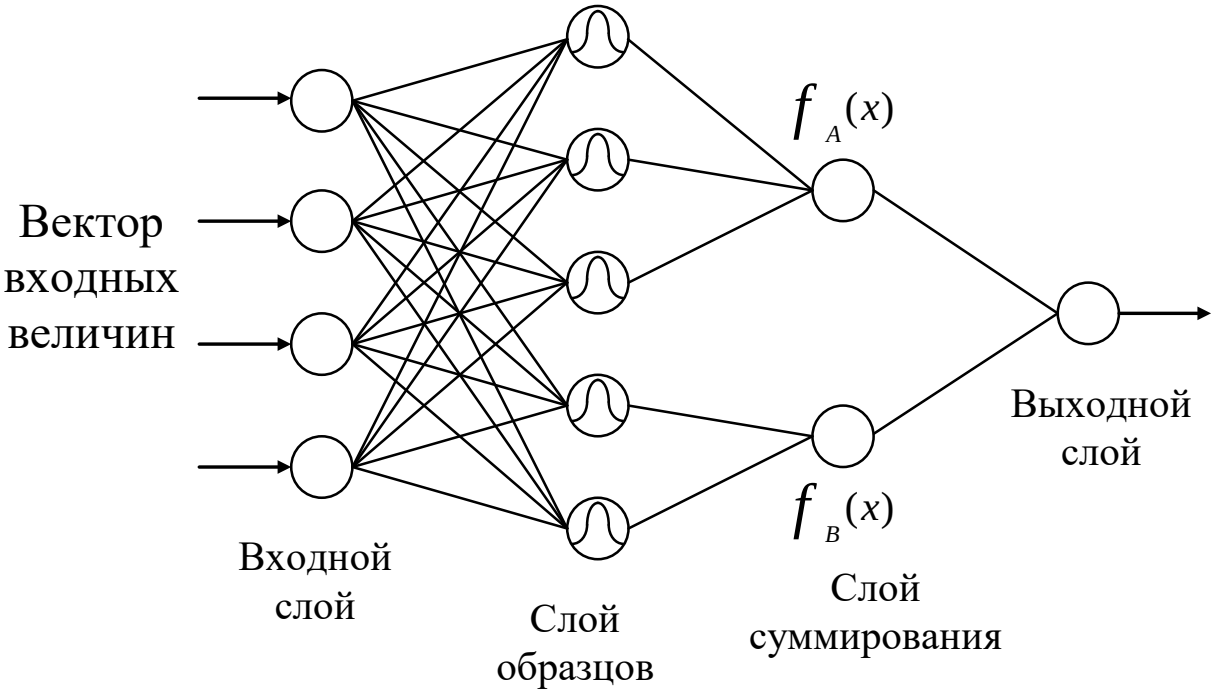
1. Формирование скрытого слоя
   - Образцами обучающей выборки;
   - С учетом имеющихся кластеров в данных;
2. Обучение выходного линейного слоя
   - Методом псевдообратных матриц: $\overline{w} = (A^T A)^{-1} A^T \overline{u}$
   - NLMS – Normalized Least Mean Squares: $w_j(t) = w_j(t-1) + \Delta(t) \cdot e_{net}(t) \cdot \frac{a_j(t)}{a^T(t)a(t)}$

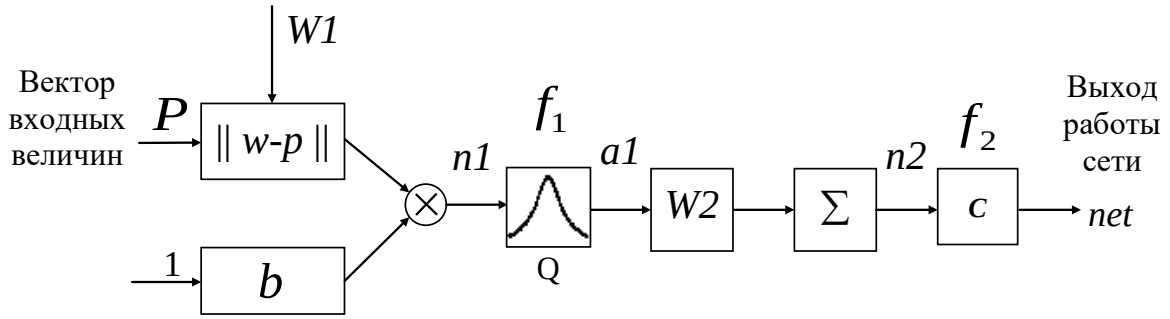
Сеть LVQ (Learning Vector Quantilization)





Вероятностная сеть (Probabilistic Neural Network - PNN)





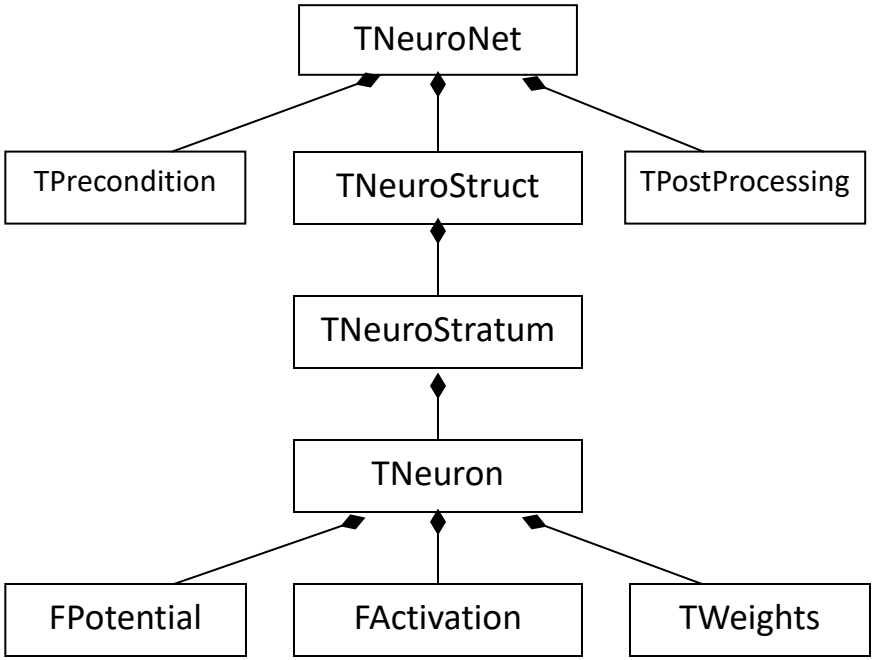
Для двух классов *A* и *B* в соответствии с данными правилами выбирается класс A, если

$$h_A c_A f_A(x) > h_B c_B f_B(x)$$

где *h* обозначает априорную вероятность, *c* – цену ошибки классификации, а *f(x)* – функцию плотности вероятности.

Иерархия классов описания нейронной сети

С точки зрения объектно-ориентированного подхода нейросеть представляет собой иерархию классов: сеть, слой, нейрон и т.д.



TNeuroNet - общее описание сети
TNeuroStruct - задает архитектуру сети
TPrecondition - алгоритм предобработки
TPostProcessing - алгоритм доп. обработки результата
TNeuroStratum - описание слоев нейронной сети
TNeuron - основные характеристики нейрона
FPotential - функция вычисления потенциала нейрона
Factivation - функция активации
TWeights - весовые коэффициенты

В рамках данной модели формирование нейросети заключается в поэтапном движении сверху вниз по иерархии классов. Таким образом, сначала создаются объекты классов определяющих наиболее общие характеристики сети. Затем определяются все более мелкие, характерные фрагменты строящейся модели, т.е. построение происходит от общего к частному.

Спасибо за внимание