

Нейронные сети

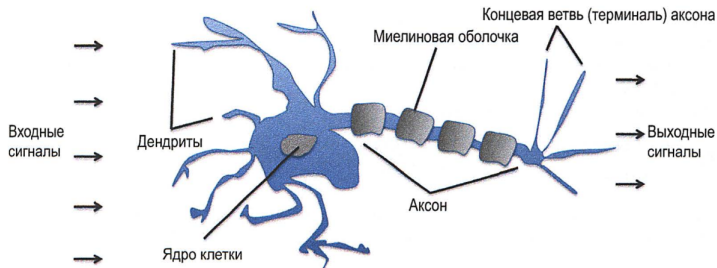
Наумов Д.А., доц. каф. КТ

Экспертные системы и искусственный интеллект, 2019

1 Математические основы нейронных сетей

Нейронные сети и нейрокомпьютеры

направление в компьютерной индустрии, в основе которого лежит идея создания искусственных интеллектуальных устройств по образу и подобию человеческого мозга.

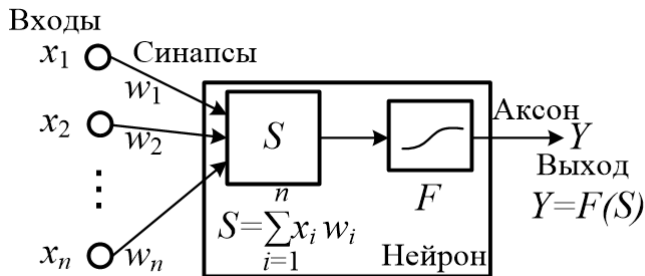


Нейрон (у человека прилизительно 10^{11}):

- получает информацию через дендриты (до 10^5);
- передает информацию через аксон, разветвляющийся на синапсы (до 10^4).
- может находиться в возбужденном и невозбужденном состоянии.

Модель нейрона

Гипотеза математического нейрона — устройства, моделирующего нейрон мозга человека.



- 1 У. Маккалок и У. Питтс, «Логическое исчисление идей, относящихся к нервной активности», Bulletin of Mathematical Biophysics (Бюллетень математической биофизики), 5 (4):115-133, 1943);

Модель нейрона

Математический нейрон:

- имеет несколько входов, N – количество входов;
- принимаются входные сигналы $X = \{x_1, x_2, \dots, x_N\} = \{x_i\}, i = 1..N$;
- суммирует входные сигналы, умножая их на весовой коэффициент $W = \{w_i\}, i = 1..N$;

$$S = \sum_{i=1}^N x_i * w_i$$

Выходной сигнал y может принимать одно из двух значений – нуль или единица и формируется по следующему правилу:

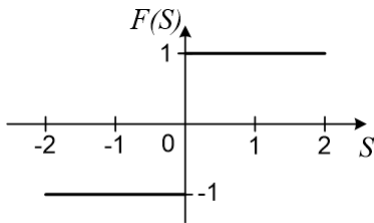
$$y = \begin{cases} 1 & \text{если } S \geq w_0 \\ 0 & \text{если } S < w_0 \end{cases}$$

Каждый нейрон:

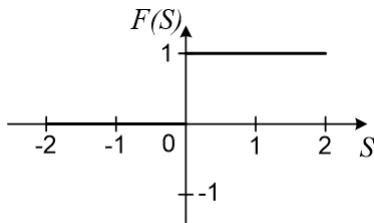
- представляет собой пороговый элемент с несколькими входами и одним выходом;
- имеет свое определенное значение порога w_0 ;
- если взвешенная сумма входных сигналов не достигает порога чувствительности, то нейрон не возбужден и его выходной сигнал $y = 0$;
- если же входные сигналы достаточно интенсивны и их сумма достигает порога чувствительности, то нейрон переходит в возбужденное состояние и на его выходе образуется сигнал $y = 1$;
- весовые коэффициенты w_i имитируют электропроводность нервных волокон — силу синаптических связей между нейронами.

Логическая функция $y(S)$ называется **активационной функцией**.

$$y = \begin{cases} 1 & \text{если } S \geq w_0 \\ 0 & \text{если } S < w_0 \end{cases}$$



а



б

- *а* - *симметричная* активационная функция;
- *б* - *смещенная* активационная функция.

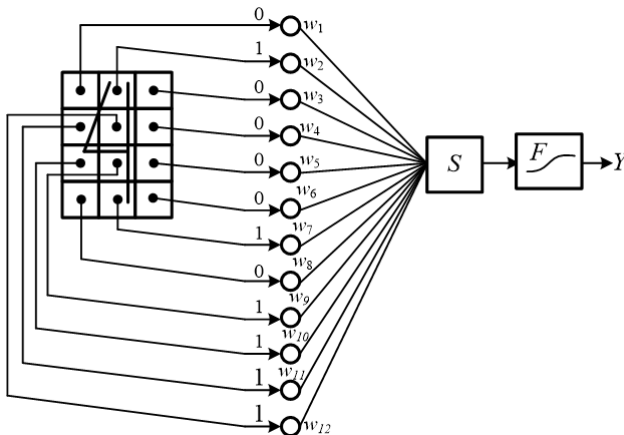
Персептрон

- У. Мак-Каллок и В. Питтс высказали идею о том, что сеть из математических нейронов в состоянии обучаться, распознавать образы, обобщать, т.е. она обладает свойствами человеческого интеллекта.
- Эта идея была материализована в 1958 году Фрэнком Розенблаттом сначала в виде компьютерной программы, а затем в виде электронного устройства, моделирующего человеческий глаз.
- Это устройство представляло собой совокупность искусственных нейронов Мак-Каллока – Питтса и было названо персептроном.
- Устройство удалось обучить решению сложнейшей интеллектуальной задачи – распознаванию букв латинского алфавита.

Ф. Розенблатт, «Персептрон, воспринимающий и распознающий автомат» Cornell Aeronautical Laboratory (Лаборатория аэронавтики Корнелльского университета), 1957).

Задача классификации цифр

- **Задача:** классификация цифр на четные и нечетные.
- **Цель обучения персептрона:** чтобы $y = 1$, если на карточке четная цифра, и $y = 0$, если цифра нечетная.



Задача бинарной классификации

- задаются **два класса**: 1 (положительный класс), -1 (отрицательный класс);
- рассчитывается **чистый вход** - линейная комбинация весов и входных значений $z = w_1 \cdot x_1 + \dots + w_m \cdot x_m$;

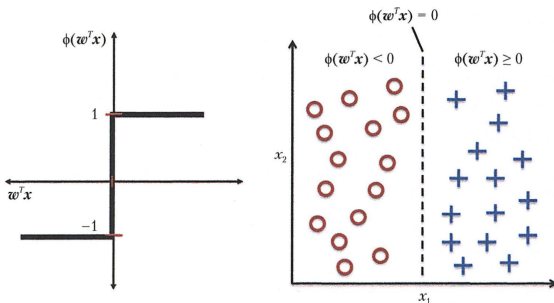
$$w = \begin{bmatrix} w_1 \\ \vdots \\ w_m \end{bmatrix}, x = \begin{bmatrix} x_1 \\ \vdots \\ x_m \end{bmatrix}.$$

- определяется передаточная функцию (функция активации) $\phi(z)$;
- в алгоритме персептрона функция активации - это простая ступенчатая функция Хевисайда;

$$\phi(z) = \begin{cases} 1, & \text{если } z \geq 0 \\ -1, & \text{иначе} \end{cases}$$

- активация отдельно взятого образца $x^{(i)}$, т. е. выход из $\phi(z)$, превышает заданный порог, то мы распознаем класс 1, в противном случае - класс -1;
- порог учитывается так: $w_0 = -\theta$, $x_0 = -1$;

$$z = w_0 x_0 + \dots + w_m x_m = \sum_{j=0}^m x_j w_j = w^T x.$$

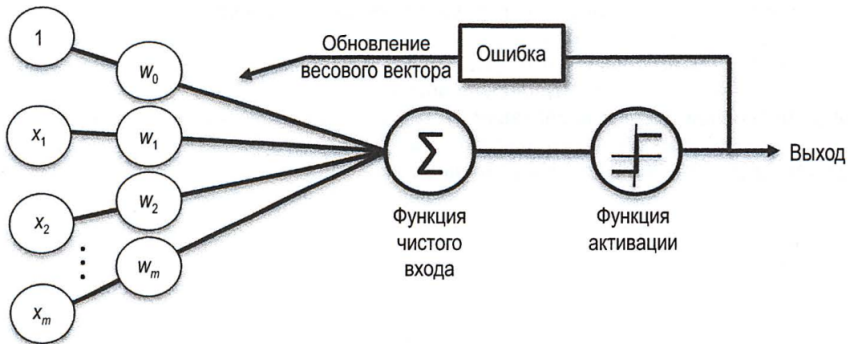


Обучение персептрона

Идея, лежащая в основе обучения персептронной модели Розенблатта:

- 1 В процессе обучения корректируются весовые коэффициенты таким образом, чтобы ошибка классификации для всех цифр была равна нулю или меньше заданной погрешности.
- 2 Обучение происходит с использованием обучающих примеров, подобно обучению маленьких детей словам или буквам.
- 3 Каждый пример предъявляется распознающему устройству неоднократно и при правильном распознавании типа цифры веса не меняются, а при наличии ошибки корректируются, пока не будет достигнута безошибочная классификация.

Схема обучения



Обучение персептрона

Алгоритм обучения персептронной модели Розенблатта:

- ❶ инициализировать веса нулями либо малыми случайными числами;
- ❷ для каждого тренировочного образца $x^{(i)}$ выполнить следующие шаги:
 - вычислить выходное значение \hat{y} ;
 - обновить веса.

Выходное значение - метка класса, идентифицированная единичной ступенчатой функцией.

$$\omega_j := \omega_j + \Delta\omega_j$$

Значение $\Delta\omega_j$ вычисляется правилом обучения персептрона:

$$\Delta\omega_j := \eta(y^{(i)} - \hat{y}^{(i)})x_j^{(i)}$$

- η - темп обучения (константа 0.0..1.0);
- $y^{(i)}$ - истинная метка класса i -го тренировочного образца;
- $\hat{y}^{(i)}$ - идентифицированная метка класса.

Персептрон правильно распознает метку класса, веса остаются неизменными:

$$\Delta w_j = \eta(-1 - -1)x_j^{(i)} = 0;$$

$$\Delta w_j = \eta(1 - 1)x_j^{(i)} = 0.$$

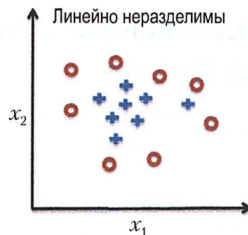
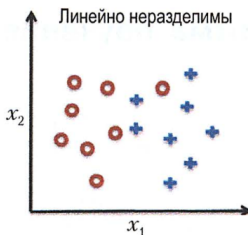
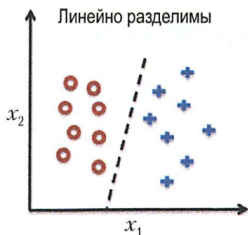
В случае неправильного распознавания веса продвигаются в направлении соответственно положительного или отрицательного целевого класса:

$$\Delta w_j = \eta(1 - -1)x_j^{(i)} = \eta(2)x_j^{(i)};$$

$$\Delta w_j = \eta(-1 - 1)x_j^{(i)} = \eta(-2)x_j^{(i)}.$$

Замечания о сходимости:

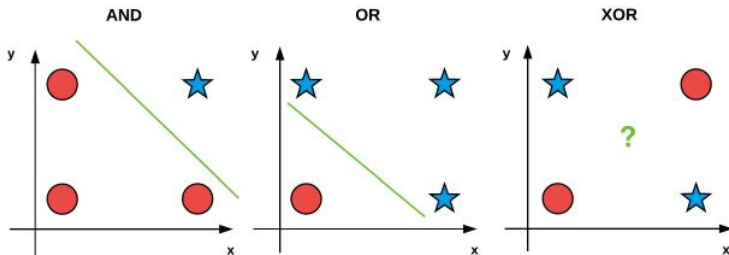
- 1 сходимость персептрона гарантируется, только если эти два класса линейно разделимы и темп обучения достаточно небольшой.
- 2 если эти два класса не могут быть разделены линейной границей решения, необходимо установить максимальное число проходов по тренировочному набору данных (эпох) и/или порог на допустимое число случаев ошибочной классификации, иначе обновление весов будет продолжаться бесконечно.



Всегда ли алгоритм обучения персептрона приводит к желаемому результату?

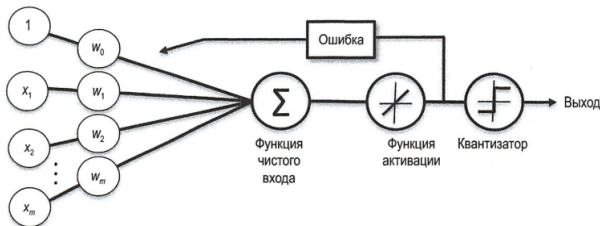
Теорема сходимости персептрона

если существует множество значений весов, которые обеспечивают конкретное различие образов, то в конечном итоге алгоритм обучения персептрона приводит либо к этому множеству, либо к эквивалентному множеству, такому, что данное различие образов будет достигнуто.



ADALINE – ADaptive Llinear NEuron, адаптивный линейный нейрон.

- для обновления весов используется линейная функция активации $\phi(\omega^T x) = \omega^T x$, а не единичная ступенчатая, как в персептроне;
- с целью распознавания меток классов используется квантизатор, аналогичный встречавшейся ранее единичной ступенчатой функции.



Б. Видроу и др., Адаптивный нейрон «Adaline» с использованием химических «мемисторов» - Number Technical Report Stanford Electron Labs, Стэнфорд, Калифорния, 1960

Ключевая составляющая алгоритмов машинного обучения с учителем: задание целевой функции, которая подлежит оптимизации во время процесса обучения. Пример среднеквадратической ошибки распознавания цифр:

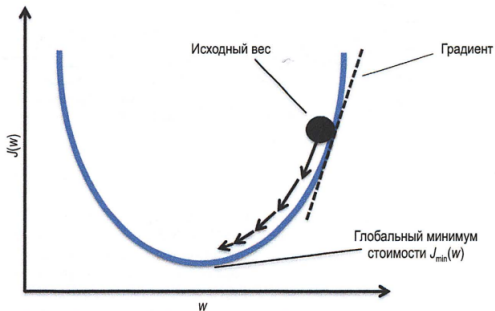
$$J(W) = 1/2 \sum_{k=0}^9 (y_k(X_k, W) - d_k)^2$$

- $X_k = x_{k,i}, i = 1..12$ – вектор входных сигналов, для каждой цифры – своя битовая строка;
- $W_k = w_i, i = 1..12$ – весовые коэффициенты, значения которых должны быть найдены в процессе обучения;
- $y_k(X_k, W)$ – сигнал на выходе нейрона при подаче на его вход k -й цифры и значениях весовых коэффициентов;
- d_k – заданное значение выходного сигнала для k -й цифры (если цифра четная, то $d_k = 1$, если цифра нечетная, то $d_k = 0$);
- $e_k = |y_k(X_k, W) - d_k|$ – ошибка распознавания k -й цифры;

Градиент

вектор, состоящий из частных производных анализируемой функции по ее аргументам.

- задает направление наискорейшего возрастания функции в некоторой точке;
- при поиске минимума функции необходимо двигаться в направлении, противоположном вектору градиента, или вдоль вектора, составляющего тупой угол с вектором градиента.



Обновление веса на основе градиентного спуска путем выполнения шага в противоположную сторону от градиента целевой функции:

$$w_i(t+1) = w_i(t) - \eta \cdot G_i$$

где

- $w_i(t+1)$ – новое значение весовых коэффициентов;
- $w_i(t)$ – старое значение весовых коэффициентов;
- η – скорость обучения;
- $G_i = \partial J / \partial w_i$ – вектор градиента.

Для нейрона, распознающего цифры, вектор градиента будет иметь вид:

$$G_i = \frac{\partial J}{\partial y_k} \frac{\partial y_k}{\partial w_i} = \frac{1}{2} \cdot 2 \cdot (y_k - d_k) \frac{\partial \sum_{i=1}^{12} x_{k_i} w_i}{\partial w_i} = e_k \cdot x_{k,i}$$

где

- i – номер входного сигнала;
- e_k – ошибка при предъявлении k -го примера (k -ой цифры);

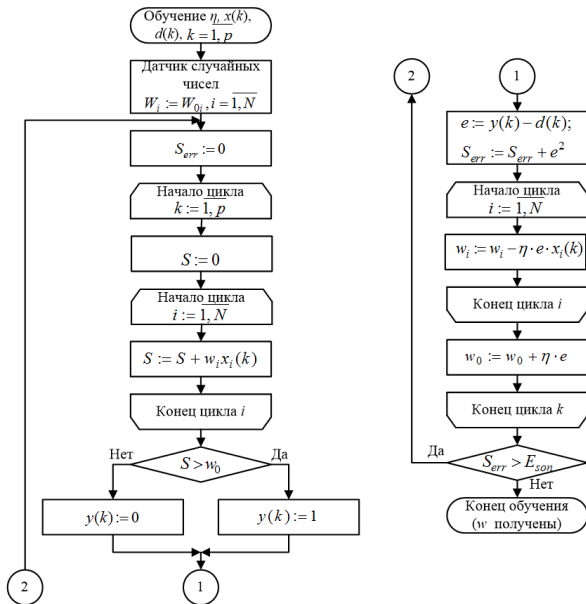
Формула подстройки весовых коэффициентов при предъявлении k -го примера:

$$w_i(t+1) = w_i(t) - \eta \cdot e \cdot x_i$$

Подстройка порогового значения нейрона:

$$w_0(t+1) = w_0(t) - \eta \cdot e$$

Первоначально весовым коэффициентам даются произвольные значения с помощью датчика случайных чисел.



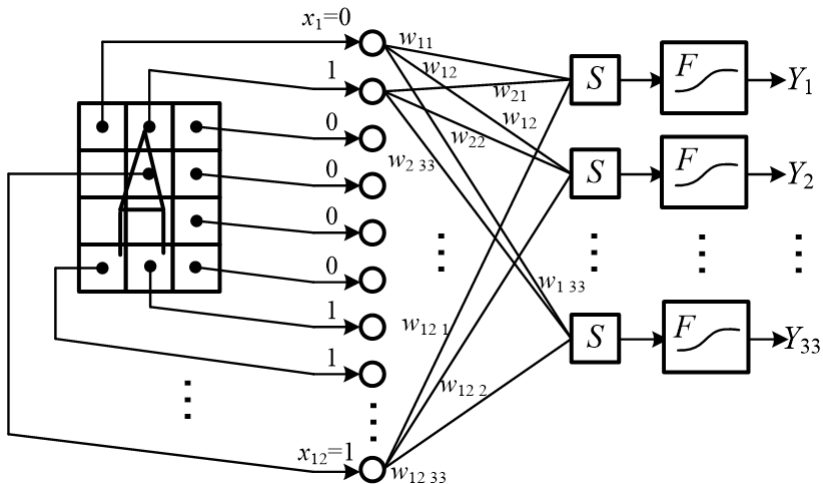
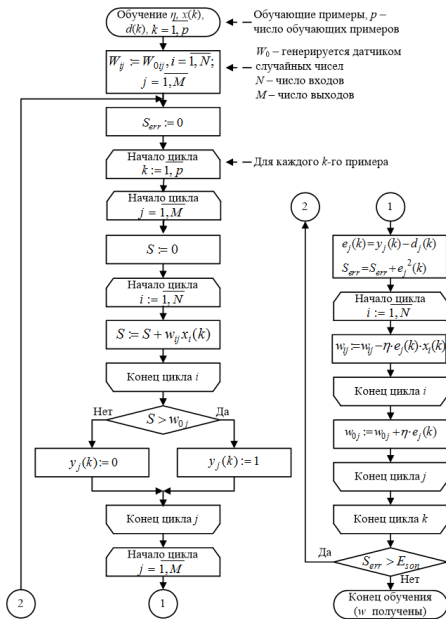


Рис.: Схема нейрона для распознавания букв

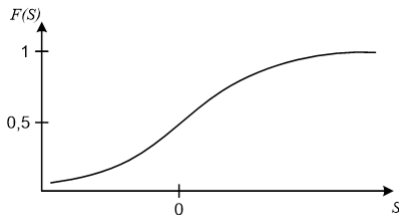


Сигмоида – нелинейная функция активации (Уидроу,Хофф):

$$y = f(S) = \frac{1}{(1 + e^{-\alpha S})}$$

где

- α – положительная константа, от значения которой зависит крутизна сигмоиды;
- $f(S)$ – обозначение произвольной активационной функции.



- выходное значение нейрона лежит в диапазоне $[0, 1]$.
- дифференцируема на всей оси абсцисс;
- усиливает малые сигналы лучше, чем большие – предотвращает насыщение от больших сигналов.

Настройка весов

Целевая функция:

$$J = 1/2 \sum_{k=1}^{33} (\vec{y}_k(X_k, W) - \vec{d}_k)^2$$

Подстройка весовых коэффициентов для k -го примера:

$$w_{ij}(t+1) = w_{ij}(t) - \eta \cdot e_j \cdot f'(S_j) \cdot x_i, i = 1..m, j = 1..n$$

Если $f(S)$ – сигмоида, то ее производная

$$f'(S) = ((1 + e^{\alpha S_j})^{-1})' = f(S)(1 - f(S))$$

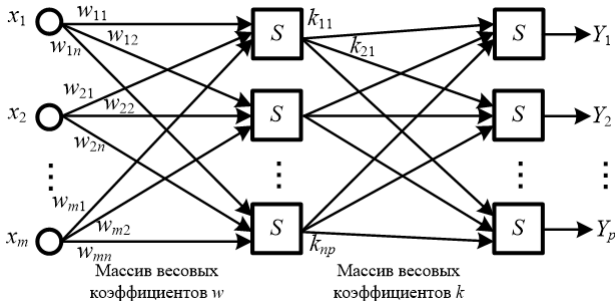
Тогда формула подстройки весов примет вид:

$$w_{ij}(t+1) = w_{ij}(t) - \eta \cdot e_j \cdot y_i \cdot (1 - y_i) \cdot x_i, i = 1..m, j = 1..n$$

где $e_j = y_j - d_j$.

Ограниченность однослойного персептрона

- персептроны в принципе не способны решать многие простые задачи («Персептроны» М. Минский и С. Пайперт);
- предложено усложнить структуру персептронов, использовать два слоя нейронов:



- Многослойные нейронные сети расширяют класс задач, решаемых персептронами.

Дискретный динамический процесс как модель нейронной сети

Рассмотрим постановку задачи подстройки весовых коэффициентов многослойной нейронной сети как задачу оптимального управления дискретным динамическим процессом. Пусть

- $x(n) = x_0(n), x_1(n), \dots, x_{m_n}(n)$ – сигнал на входе n -ого слоя сети;
- m_n – число нейронов в слое n ;
- $x(n)$ – вектор-строка;
- $x^T(n)$ – вектор-столбец;
- $x_0(n) = 1$
- $u(n) = W(n)$ – матрица весовых коэффициентов n -го слоя сети размером $m_{n-1} : m_n$;
- $x(0)$ – сигналы, подаваемые на вход нейронной сети;
- $(x_k(0) = a^k; x^k(N) = d^k), k = 1, 2, \dots, P$ – обучающие примеры: при k -м входном сигнале $x_k(0) = a^k$ на выходе сети необходимо обеспечить сигнал $x^k(N) = d^k$ путем выбора весовых коэффициентов W .

Задача настройки нейронной сети

Найти такие значения оптимального управления $u(n) = W(n)$, $n = 1, 2, \dots, N$, при которых будут выполнены ограничения:

$$x(0) = a; x(n) = f^n(S(n)) = f^n((x(n-1) \cdot W(n)))$$

где

- $f^n(..)$ - активационная функция;
- $S(n) = x(n-1) \cdot W(n)$ - вектор сумм взвешенных сигналов для n -го слоя нейронной сети;
- $n = 1, 2, \dots, N$.

Критерий качества процесса:

$$\min \Phi(x(N)) = \min \sum_{k=1}^P (x^k(N) - d^k)^2.$$

Система называется **прямой системой уравнений**, используется для получения сигнала на выходе нейронной сети при заданных значениях весовых коэффициентов W и известном входном сигнале.

Сопряженная (двойственная система)

описывает процесс, который называют «обратным распространением ошибки».

$$p(N) = \frac{\partial \Phi(x(N))}{\partial x(N)};$$

$$p(n-1) = \left[\frac{\partial f(x(n-1), u(n))}{\partial x(n-1)} \right]^T \cdot p(n), n = N, \dots, 1$$

где

- $p(n) = p_1(n), \dots, p_{mn}(n)$;
- $\frac{\partial f(x, u)}{\partial x} = \left[\frac{\partial f_i(x, u)}{\partial x_j} \right], i = 1..m_n, j = 1..m_{n-1}$
- m_{n-1}, m_n – число нейронов в слое $n-1$ и в слое n ;

Функция Гамильтона:

$$H(p(n), x(n-1), W(n)) = (p(n), f^n(x(n-1) \cdot W(n)))$$

При фиксированных p и x функция Гамильтона становится функцией только управляющего воздействия $W(n)$.

Алгоритм подстройки весов многослойной нейросети

- 1 задание произвольных значений весовых коэффициентов для всех слоев;
- 2 расчет по формулам сигналов $x(n)$, $n = 1, 2, \dots, N$ на выходе каждого слоя нейронов;
- 3 расчет по формулам сопряженного вектора $p(n)$, $n = N, N-1, \dots, 1$, по значению которого можно определить слой, наиболее влияющий на критерий оптимальности;
- 4 вычисление для каждого слоя функции Гамильтона;
- 5 для каждого слоя определение направления изменения значений весовых коэффициентов в сторону уменьшения функции Гамильтона одним из известных методов (градиентным, методом Ньютона и т.д.);
- 6 определение длины шага и нового управления;
- 7 оценка близости полученного решения к оптимальному решению.

Всегда ли можно построить нейросеть, выполняющую преобразование, заданное любой обучающей выборкой?

Теорема

- Для любого множества пар отличных между собой входных и выходных векторов произвольной размерности (X_q, D_q) , $q = 1, \dots, P$ существует двухслойный персептрон с сигмоидальными активационными функциями и с конечным числом нейронов, который для каждого входного вектора X_q формирует соответствующий ему выходной вектор D_q .

Количество нейронов в скрытых слоях персептрона:

$$\frac{N_y \cdot P}{1 + \log_2 P} \leq N_w \leq N_y(P/N_x + 1) \cdot (N_x + N_y + 1) + N_y$$

где

- N_y – размерность выходного сигнала;
- P – число элементов обучающей выборки;
- N_w – необходимое число синаптических весов;
- N_x – размерность входного сигнала.

Предобработка обучающих примеров

Исключение незначимых параметров:

- путем анализа значений весовых коэффициентов входных нейронов;
- путем возмущения значений входных параметров и анализа реакции сети на эти возмущения.

Предобработка обучающих примеров:

- кодирование в числовом виде;
- выравнивание диапазонов изменения величин в интервалы $[0, 1]$ или $[-1, 1]$.

$$\tilde{x}_n = \frac{x_n - x_{n\min}}{x_{n\max} - x_{n\min}} \cdot (b - a) + a$$

где

- \tilde{x}_n и x_n – значения исходного и масштабированного n -го параметра предметной области, подаваемого на n -й входной нейрон сети;
- $x_{n\max}, x_{n\min}$ – реальный диапазон изменения n -го параметра;
- $[a, b]$ – приемлемый диапазон изменения входных сигналов.

При проектировании персептронов существует проблема выбора необходимого числа нейронов:

- число нейронов входного слоя персептрона должно совпадать с размерностью вектора входных параметров X , который определен условиями решаемой задачи;
- число нейронов выходного слоя должно совпадать с размерностью выходного вектора Y , что также определено условиями задачи.
- число нейронов в скрытых слоях может быть приближенно оценено по следующим формулам, однако его желательно оптимизировать для каждой конкретной задачи.

Число нейронов скрытого слоя двухслойного персептрона:

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

- Строгой теории выбора оптимального числа скрытых слоев персептронов пока нет.
- На практике же чаще всего используются персептроны, имеющие один или два скрытых слоя, причем число нейронов в скрытых

При проектировании персептрона необходимо понимать, что персептрон должен:

- правильно реагировать на обучающие примеры;
- уметь обобщать приобретенные знания, т.е. правильно реагировать на случаи, которых в обучающей выборке не было.

Чтобы оценить способность сети к обобщению, помимо обучающей выборки примеров $(X; D)$ в рассмотрение вводят некоторое количество тестовых примеров $(X_T; D_T)$, которые относятся к той же предметной области, но в процессе обучения не участвуют.

Погрешность обучения

среднеквадратичная погрешность персептрона, вычисленная на обучающей выборке $(X; D)$.

Погрешность обобщения

среднеквадратичная погрешность персептрона, вычисленная на тестовой выборке $(X_t; D_t)$.

Переобучение

При увеличении числа нейронов внутренних слоев персептрона N :

- погрешность обучения обычно падает;
- погрешность обобщения сначала падает, а затем, начиная с некоторого оптимального значения $N = N_0$, возрастает.

Переобучение

свойство нейросети терять способность к обобщению при чрезмерном увеличении числа ее синаптических связей.

Проблема обучения персептронов: поверхность функции ошибок обычно имеет очень сложную форму с множеством локальных минимумов.

- актуальным является развитие методов глобальной оптимизации;
- наиболее успешным признается идея генетических алгоритмов для подстройки весов.