



Ахтеров А.В., [Кирильченко А.А.](#)

Основы теоретической  
робототехники.  
Искусственные нейронные  
сети. Обзор

**Рекомендуемая форма библиографической ссылки:** Ахтеров А.В., Кирильченко А.А. Основы теоретической робототехники. Искусственные нейронные сети. Обзор // Препринты ИПМ им. М.В.Келдыша. 2008. № 2. 20 с. URL: <http://library.keldysh.ru/preprint.asp?id=2008-2>



**Ордена Ленина  
ИНСТИТУТ ПРИКЛАДНОЙ МАТЕМАТИКИ  
имени М.В. Келдыша  
Российской академии наук**

А.В. Ахтёров, А.А. Кирильченко

**ОСНОВЫ ТЕОРЕТИЧЕСКОЙ  
РОБОТОТЕХНИКИ.  
ИСКУССТВЕННЫЕ НЕЙРОННЫЕ СЕТИ.  
(ОБЗОР)**

**Препринт №**

**Москва**

Российская Академия Наук  
ОРДЕНА ЛЕНИНА ИНСТИТУТ ПРИКЛАДНОЙ МАТЕМАТИКИ  
им. М.В. Келдыша

А.В. Ахтёров, А.А. Кирильченко

**Основы теоретической робототехники.  
Искусственные нейронные сети.  
(Обзор)**

Москва, 2007 г.

А.В. Ахтёров, А.А. Кирильченко

ОСНОВЫ ТЕОРЕТИЧЕСКОЙ РОБОТОТЕХНИКИ.. ИСКУССТВЕННЫЕ НЕЙРОННЫЕ СЕТИ.. ОБЗОР

A.V. Akhterov, A.A. Kiril'chenko

BASES OF THE THEORETICAL ROBOTOTECHNICS. ARTIFICIAL NEURAL NETWORKS. REVIEW

АННОТАЦИЯ

Искусственные нейронные сети являются одним из основных направлений современной теории искусственного интеллекта.

Изложены биологические предпосылки создания теории искусственных нейронных сетей. Дано определение формального нейрона. Рассмотрены однослойные и многослойные нейронные сети. Приведена основная теорема для аппроксимации функций при помощи многослойных нейронных сетей и описан базовый алгоритм обучения нейронных сетей. Даны общие характеристики релаксационных и самоорганизующихся сетей.

Вторая часть работы посвящена применению искусственных нейронных сетей в различных областях. Приведен пример движения мобильного робота на основе дальномерных данных. Рассмотрено распознавание локальных относительных признаков при движении мобильного робота, а также описано применение самоорганизующихся карт Кохонена при распределении финансирования высших учебных заведений.

ABSTRACT

Artificial neural networks one of the basic directions of the modern theory of an artificial intelligence.

In the first part of work biological preconditions of creation of the theory of artificial neural networks are stated. It is defined formal neuron. Single-layered and multilayered neural networks are considered. The basic theorem for approximation of functions by means of multilayered neural networks is resulted and the base algorithm of training of neural networks is described. General characteristics of the relaxing and self-organizing neural networks are given.

The second part of work is devoted to application of artificial neural networks in various areas. The example of movement of the mobile robot on the basis of the rangefinder's data is resulted. Recognition during movement of the mobile robot of a local relative attributes is considered, and application self-organizing maps at distribution of financings of institute is also described.

Работа выполнена при частичной поддержке грантов РФФИ  
06-08-1151, 05-01-00-885.

## СОДЕРЖАНИЕ

	стр.
Введение .....	3
1. Биологические предпосылки возникновения теории искусственных нейронных сетей .....	4
2. Основные понятия и однослойные нейронные сети.....	4
2.1. Искусственный нейрон и функция активации .....	4
2.2. Однослойные нейронные сети и правила их обучения .....	5
3. Многослойные нейронные сети .....	9
3.1. Алгоритм обратного распространения ошибки .....	9
4. Релаксационные нейронные сети.....	11
4.1. Нейронная сеть Хопфилда .....	11
5. Самоорганизующиеся нейронные сети Кохонена.....	14
5.1. Конкурентное обучение .....	14
5.2. Самоорганизующиеся карты Кохонена .....	15
Литература.....	16
Рисунки .....	17

## ВВЕДЕНИЕ

Система искусственного интеллекта способна в зависимости от состояния входов, изменять не только параметры функционирования, но и сам способ своего поведения, причём способ поведения зависит не только от текущего состояния входов, но также от предыдущих состояний системы.

В настоящее время искусственный интеллект (ИИ) сформировался уже не как одно из направлений информационных технологий, а как обширная область исследований и разработок интеллектуальных систем, предназначенных для работы в трудно формализуемых областях деятельности. Можно выделить следующие направления развития ИИ: разработка интеллектуальных систем, основанных на знаниях, распознавание образов, игры и творчество, компьютерная лингвистика, интеллектуальные роботы и т.д. Одна из основных тем исследований в области ИИ является нейросетевые и нейрокомпьютерные технологии. В основе этих исследований лежат искусственные нейронные сети (НС), которые представляют собой математическую конструкцию, основным элементом которой является *искусственный нейрон* [5-7].

Данная работа продолжает линию, начатую [8].

При составлении данного обзора активно использовались работы [1-7,9].

## 1. БИОЛОГИЧЕСКИЕ ПРЕДПОСЫЛКИ ВОЗНИКНОВЕНИЯ ТЕОРИИ ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ

В основе теории искусственных нейронных сетей лежит идея создания интеллектуального устройства по образу и подобию человеческого мозга. Дело в том, что способ обработки информации человеческим мозгом принципиально отличается от способов, используемых в обычных – неймановских компьютерах. Основное преимущество биологического мозга заключается в параллельности производимых им вычислений [5-7].

В первом приближении, мозг человека состоит из нейронов и соединяющих их волокон. Каждый нейрон состоит из тела клетки, дендритов и аксонов, и, обладающей электрической активностью. Упрощённо принцип действия нейрона выглядит следующим образом: через *дендриты* нейрон получает нервные импульсы от других нейронов; в *теле нейрона* сигнал обрабатывается; обработанный сигнал, через *аксон*, разветвляющийся на конце на множество нервных нитей - *синапсы*, передаётся дальше, к другим нейронам. На рис. 1 представлена схема биологического нейрона. Простейший нейрон может иметь порядка  $10^3 \dots 10^4$  дендритов, т.е. быть связанным с 10000 других нейронов. Всего в человеческом мозге приблизительно  $10^{11}$  нейронов.

Следует отметить, что биологически мозг обладает способностью организовывать нейроны и связи между ними так, что они выполняют такие задачи как распознавание образов, обработку сигналов органов чувств, моторные функции и т.д. во много раз быстрее чем, это делают самые современные компьютеры, так как хотя скорость распространения нервного импульса в аксоне приблизительно 100 м/с, что в миллион раз меньше, чем скорость передачи электрического сигнала, однако за счёт огромного количества соединений между нейронами, т.е. за счёт параллельной обработки информации, человеческий мозг мгновенно справляется с задачами, для решения которых «обычным» компьютерам требуются несравнимо большее время.

## 2. ОСНОВНЫЕ ПОНЯТИЯ И ОДНОСЛОЙНЫЕ НЕЙРОННЫЕ СЕТИ

### 2.1. ИСКУССТВЕННЫЙ НЕЙРОН И ФУНКЦИЯ АКТИВАЦИИ

Как говорилось ранее искусственная нейронная сеть (НС), это математическая модель, основной частью которой является искусственный нейрон, осуществляющий нелинейное преобразование  $F$  суммы произведения входных сигналов на весовые коэффициенты (рис. 2):

$$y = F\left(\sum_{i=1}^n w_i x_i\right) = F(S) = F(WX^T),$$

где  $W = (w_1, w_2, \dots, w_n)$  - вектор весовых коэффициентов,  $X = (x_1, x_2, \dots, x_n)$  - вектор входных сигналов,  $S = \sum_{i=1}^n w_i x_i$  - взвешенная сумма,  $F$  – оператор нелинейного преобразования или *функция активации*. Взвешенную сумму  $S$  иногда пред-

ставляют в виде  $S = \sum_{i=1}^n w_i x_i - T$ , где  $T$  - порог нейронного элемента, характеризующий сдвиг функции активации по оси абсцисс. Если продолжить биологические аналогии, то входы  $x_i$  являются для искусственного нейрона дендритами, операторы суммирования и нелинейного преобразования – телом нейрона, а выход  $y$  – аксоном.

В качестве оператора нелинейного преобразования могут выступать различные функции, которые выбираются в соответствии с рассматриваемой задачей и типом НС. Рассмотрим наиболее часто используемые функции активации.

- *Линейная функция:*  $y = kS$ .

- *Пороговая функция:*  $y = \text{sign}(S) = \begin{cases} 1, S > 0; \\ 0, S \leq 0. \end{cases}$  или  $y = \begin{cases} 1, S > 0; \\ -1, S \leq 0. \end{cases}$

- *Сигмоидная функция:*  $y = \frac{1}{1 + e^{-cS}}$ , где  $c > 0$  - коэффициент, определяющий ширины функции по оси абсцисс.

Следует отметить, что в силу монотонности и всюду дифференцируемости сигмоидная функция является наиболее распространённой. Её график приведён на рис.3.

## 2.2. ОДНОСЛОЙНЫЕ НЕЙРОННЫЕ СЕТИ И ПРАВИЛА ИХ ОБУЧЕНИЯ

В многослойной НС нейроны располагаются по слоям. В простейшем случае в такой сети существует *входной слой*, информация от которого передаётся на *выходной слой*. Такая сеть называется *однослойной*. Её топология представлена на рис. 4. Так как в таких сетях сигналы передаются в одном направлении, они получили названия *сети прямого распространения*.

Как видно из рисунка, каждый вход  $x_i$  соединён с каждым нейроном единственного слоя НС. Выход  $j$ -го нейронного элемента вычисляется по формуле:

$$y_j = F\left(\sum_{i=1}^n w_{ij} x_i - T_j\right),$$

где  $w_{ij}$  - весовой коэффициент между  $i$ -ым входом и  $j$ -ым нейроном. Таким образом весовые коэффициенты НС можно представить в виде матрицы размерностью  $n \times m$ , где  $n$  - количество вход НС,  $m$  - количество нейронных элементов:

$$W = \begin{pmatrix} w_{11} & w_{12} & \dots & w_{1m} \\ w_{21} & w_{22} & \dots & w_{2m} \\ \dots & \dots & \dots & \dots \\ w_{n1} & w_{n2} & \dots & w_{nm} \end{pmatrix}.$$

Одной из основных задач при использовании нейронных сетей является подбор весовых коэффициентов, для решения рассматриваемой задачи. Для этой цели существуют *правила обучения нейронных сетей*, которые определяют как изменяются значение весовых коэффициентов в зависимости от входных сигналов. Рассмотрим правила обучения однослойных нейронных сетей.

### 1. Правило обучения Хебба.

Обучение методом Хебба является самым известным среди всех правил обучения. Оно основывается на биологической предпосылке о том, что если два нейрона при подаче сигнала активизируются одновременно, «прочность» этого соединения возрастает. Математически правило обучения Хебба записывается следующим образом:

$$w_{ij}(0) = 0, w_{ij}(k+1) = w_{ij}(k) + x_i y_j,$$

где  $k$  - номер итерации.

В качестве примера рассмотрим решение двумерной логической задачи «И» при помощи процедуры обучения Хебба. Для решения используется простейшая НС состоящая из 2 нейронов входного слоя и 1 нейрона выходного слоя (рис. 5). Как видно из рисунка порог нейронного элемента можно вынести в качестве отдельного входа с весовым коэффициентом равным  $T$  и входным сигналом  $-1$ . Для данной сети взвешенная сумма равна:

$$S = w_{11}x_1 + w_{21}x_2 - T,$$

а правило Хебба запишется в виде следующей системы:

$$w_{11}(k+1) = w_{11}(k) + x_1 y_1;$$

$$w_{21}(k+1) = w_{21}(k) + x_2 y_1;$$

$$T(k+1) = T(k) - y_1;$$

Рассмотрим решение данной задачи методом обучения с учителем, т.е. в качестве выходных значений НС будем использовать эталонные значения. Остальные параметры НС выглядят следующим образом:

- биполярная кодировка входных сигналов;
- функции активации: биполярная пороговая функция 
$$y = \begin{cases} 1, S > 0; \\ -1, S \leq 0 \end{cases}.$$

Имеем следующую обучающую выборку:

$x_1$	$x_2$	$y$
-1	-1	-1
1	-1	-1
-1	1	-1
1	1	1

Итак рассмотрим процесс обучения:

$k = 0: w_{11} = 0, w_{21} = 0, T = 0;$	
$k = 1: x_1 = -1, x_2 = -1, x_3 = -1, y = -1 \Rightarrow$ $w_{11} = 0 + (-1) \cdot (-1) = 1,$ $w_{21} = 0 + (-1) \cdot (-1) = 1,$ $T = 0 - (-1) = 1;$	$k = 2: x_1 = 1, x_2 = -1, x_3 = -1, y = -1 \Rightarrow$ $w_{11} = 1 + 1 \cdot (-1) = 0,$ $w_{21} = 1 + (-1) \cdot (-1) = 2,$ $T = 1 - (-1) = 2;$

$k = 3: x_1 = -1, x_2 = 1, x_3 = -1, y = -1 \Rightarrow$ $w_{11} = 0 + (-1) \cdot (-1) = 1,$ $w_{21} = 2 + 1 \cdot (-1) = 1,$ $T = 2 - (-1) = 3;$	$k = 4: x_1 = 1, x_2 = 1, x_3 = -1, y = 1 \Rightarrow$ $w_{11} = 1 + 1 \cdot 1 = 2,$ $w_{21} = 1 + 1 \cdot 1 = 2,$ $T = 3 - 1 = 2;$
--	--

Обучение прекращается при подаче на вход НС всех входных образов. В итоге, мы обучили НС (подобрали весовые коэффициенты) решать задачу логического «И».

### 2. Персептрон Розенблатта.

Персептроном Розенблатта называется нейронная сеть, состоящая из двух слоёв. Первый слой нейронов этой НС называется сенсорным и выполняет исключительно распределительную функцию входных сигналов. Второй слой называется ассоциативным. Нейронные элементы в нём имеют пороговую функцию активации и именно они производят непосредственную обработку информации.

Процедура обучения персептрона выглядит следующим образом:

$$w_{ij}(k+1) = w_{ij}(k) + \alpha x_i t_j,$$

где  $t_j$  - эталонное значение  $j$ -го выхода НС,  $\alpha$  - скорость обучения сети ( $0 < \alpha \leq 1$ ).

Таким образом, процедура обучения Розенблатта, это во-первых, обучение с учителем (имеем эталонные значения выходов), а во-вторых, обучение с подкреплением (весовые коэффициенты нейронного элемента изменяются только в том случае, если выходное значение этого элемента отличается от эталонного).

Следует отметить, что персептрон Розенблатта способен решать задачи логического «И» и логического «ИЛИ», но не может решить задачу «ИСКЛЮЧАЮЩЕЕ ИЛИ». В связи с этим, в 1969 г. М.Минский и С.Пайперт в своей работе «Персептроны: введение в вычислительную геометрию» сделали довольно пессимистичные выводы по поводу дальнейшего развития нейронных сетей, что привело к приостановке исследований в этой области практически на 20 лет.

### 3. Правило обучения Видроу-Хоффа

Данное правило используется для обучения НС в выходном слое которых, расположен один нейрон с линейной функцией активации (рис. 6). Такая НС называется ADALINE (ADActive LINer Element). Правило основано на минимизации среднеквадратичной ошибки НС, которая определяется по следующей формуле:

$$E = \frac{1}{2} \sum_{m=1}^L (y_1^m - t^m)^2,$$

где  $L$  - размерность обучающей выборки,  $y_1^m$  - выход НС, для  $m$ -го входного образа,  $t^m$  - эталонное значение для  $m$ -го входного образа. Величина

$E(m) = \frac{1}{2} (y_1^m - t^m)^2$  является среднеквадратичной ошибкой НС для  $m$ -го входного образа.

Обучение основано на методе градиентного спуска в пространстве порогов и весовых коэффициентов:

$$w_{j1}(k+1) = w_{j1}(k) - \alpha \frac{\partial E(m)}{\partial w_{j1}(k)};$$

$$T(k+1) = T(k) - \alpha \frac{\partial E(m)}{T(k)},$$

где  $j=1\dots n$ ,  $n$  – размерность входного сигнала,  $\alpha$  ( $0 < \alpha < 1$ ) - скорость обучения.

После приведения формулы запишутся следующим образом:

$$w_{j1}(k+1) = w_{j1}(k) - \alpha x_j^m (y_1^m - t^m);$$

$$T(k+1) = T(k) - \alpha (y_1^m - t^m),$$

где  $j=1\dots n$ ,  $x_j^m$  -  $j$ -ая компонента образа  $m$ .

Схема алгоритма обучения по правилу Видроу-Хоффа выглядит следующим образом:

**Шаг 1.** Задаются скорость обучения  $\alpha$  ( $0 < \alpha < 1$ ) и минимальная средне-квадратичная ошибка сети  $E_{\min}$ ;

**Шаг 2.** Случайным образом инициализируются весовые коэффициенты -  $w_{j1}$  и порог -  $T$ ;

**Шаг 3.** На вход НС поочередно подаются все  $L$  входных образов. Для каждого образа вычисляется его среднеквадратичная ошибка.

**Шаг 4.** Вычисляется среднеквадратичная ошибка всей НС -  $E$ . Если  $E \leq E_{\min}$ , то сеть считается обученной, иначе к 5.

**Шаг 5.** По приведённым выше формулам модифицируются весовые коэффициенты и порог нейрона.

Следует отметить, что если скорость обучения  $\alpha$  выбрать слишком маленькой, то процесс сходимости будет очень долгим, если же его выбрать слишком большим, то в процессе решения можно «проскочить» минимум среднеквадратичной ошибки. Решение этой проблемы возможно с помощью адаптивного шага обучения. В литературе [1] предлагается следующее выражение:

$$\alpha(k) = \frac{1}{1 + \sum_{j=1}^n x_j^2}.$$

Правило Видроу-Хоффа носит также название дельта-правило.

До сих пор речь шла о НС, содержащих один распределительный (входной) слой и один выходной слой, в котором происходила обработка информации. Если обрабатывающих информацию слоёв два или больше, то такая НС называется многослойной.

### 3. МНОГОСЛОЙНЫЕ НЕЙРОННЫЕ СЕТИ

Топология многослойной НС представлена на рисунке 7. В качестве функций активации во всех нейронных элементах обычно используется сигмоидная функция, так как она, как говорилось ранее, является монотонной и всюду дифференцируемой. Для обучения многослойных НС используется алгоритм обратного распространения ошибки, о котором речь пойдёт ниже. Следует отметить, что в основе использования этого типа НС лежит теорема А.Н. Колмогорова: *любую непрерывную функцию  $n$  переменных на единичном отрезке  $[0,1]$  можно представить в виде суммы конечного числа одномерных функций:*

$$f(x_1, x_2, \dots, x_n) = \sum_{p=1}^{2n+1} g\left(\sum_{i=1}^n \lambda_i \varphi_p(x_i)\right),$$

где функции  $g$  и  $\varphi_p$  являются одномерными и непрерывными,  $\lambda_i = \text{const}$  для всех  $i$ .

Обобщая эту теорему на многослойные нейронные сети, можно сформулировать следующее утверждение [1,4]:

*Любая непрерывная функция  $f: [0,1]^n \rightarrow [0,1]$  может быть аппроксимирована с любой степенью точности трёхслойной нейронной сетью с одним скрытым слоем, весовые коэффициенты которой, настроены с помощью алгоритма обратного распространения ошибки.*

В связи с появлением как минимум ещё одного слоя, введённые ранее величины изменяться:

$y_j = F(S_j)$ , где  $y_j$  - выход  $j$ -го нейронного элемента выходного слоя,  $S_j$  - взвешенная сумма  $j$ -го нейронного элемента;

$S_j = \sum_{i=1}^{n_3} w_{ij} y_i - T_j$ , где  $w_{ij}$  - весовой коэффициент связи идущей от  $i$ -го нейронного элемента скрытого слоя к  $j$ -у нейронному элементу выходного слоя,  $T_j$  - порог  $j$ -го нейронного элемента выходного слоя.

#### 3.1. АЛГОРИТМ ОБРАТНОГО РАСПРОСТРАНЕНИЯ ОШИБКИ

*Алгоритм обратного распространения ошибки* используется для обучения многослойных нейронных сетей. Как и правило Вудроу-Хоффа (дельта-правило) данный алгоритм минимизирует среднеквадратичную ошибку НС, используя для этого метод градиентного спуска в пространстве порогов и весовых коэффициентов. В процессе работы алгоритма можно выделить две фазы: *фаза прямого распространения входного сигнала* (входной образ подаётся на вход сети и вычисляется выходная активность НС) и *фаза обратного распространения сигнала* (по сети, в обратном направлении распространяется ошибка - отклонение полученного выхода от эталонного значения для данного входного образа).

Не вдаваясь в математические выкладки [1-4] сразу приведём схему алгоритма обратного распространения ошибки.

**Шаг 1.** Задаются скорость обучения  $\alpha$  ( $0 < \alpha < 1$ ) и минимальная среднеквадратичная ошибка сети  $E_{\min}$ ;

**Шаг 2.** Случайным образом инициализируются весовые коэффициенты -  $w_{ij}$  и пороги -  $T_j$ ;

**Шаг 3.** На вход НС поочерёдно подаются все  $L$  входных образов из обучающей выборки. Для каждого образа выполняются следующие операции:

- фаза прямого распространения сигнала, на которой вычисляется выходная активность всех слоёв НС:

$$y_j = F\left(\sum_i w_{ij} y_i - T_j\right),$$

где индекс  $j$  обозначает нейронные элементы, расположенные в слое, следующем за слоем  $i$ ;

- фаза обратного распространения сигнала, на которой вычисляется ошибка каждого из слоёв:

ошибка выходного слоя НС:  $\gamma_j = y_j - t_j$ ,

где  $\gamma_j$  - ошибка  $j$ -го нейронного элемента выходного слоя,

$y_j$  - выходное значение  $j$ -го нейронного элемента выходного слоя,

$t_j$  - эталонное значение  $j$ -го нейронного элемента выходного слоя;

ошибка всех остальных слоёв НС:  $\gamma_j = \sum_i \gamma_i F'(S_j) w_{ji}$ ,

где индекс  $i$  означает следующий слой по отношению к слою  $j$ .

**Шаг 4.** Модифицируются весовые коэффициенты и пороги НС:

$$w_{ij}(t+1) = w_{ij}(t) - \alpha \gamma_j F'(S_j) y_i;$$

$$T_j(t+1) = T_j(t) + \alpha \gamma_j F'(S_j).$$

**Шаг 5.** Вычисляется среднеквадратичная ошибка сети:

$$E = \frac{1}{2} \sum_{k=1}^L \sum_j (y_j^k - t_j^k)^2,$$

где  $L$  - размерность обучающей выборки.

**Шаг 6.** Если  $E > E_{\min}$ , то обратно к шагу 3, иначе НС считается обученной.

Следует отметить, что алгоритм обратного распространения ошибки обладает некоторыми недостатками. В следующей таблице даны способы их преодоления.

Недостаток	Способ решения
<i>Проблема инициализации весовых коэффициентов:</i> разная инициализация может привести к различным решениям задачи.	Инициализировать весовые коэффициенты значениями следующего порядка: <ul style="list-style-type: none"> <li>- <math>w_{ij} \approx 1/\sqrt{n(i)}</math>, <math>n(i)</math> - размерность <math>i</math>-го слоя;</li> <li>- <math>w_{ij} \in [-0.05, 0.05]</math>;</li> <li>- <math>w_{ij} \in [-0.1, 0.1]</math>.</li> </ul>

<i>Проблема топологии НС:</i> при слишком большой размерности скрытого слоя может наступить эффект перетренированности.	Размерность скрытого слоя должна быть меньше размерности обучающей выборки.
<i>«Застревание» решения в локальных минимумах пространства порогов и весовых коэффициентов</i>	Изменение весовых коэффициентов по формуле: $\Delta w_{ij}(t+1) = -\alpha \gamma_j F'(S_j) y_i + \lambda \Delta w_{ij}(t);$ $\Delta T_j(t+1) = \alpha \gamma_j F'(S_j) + \lambda \Delta T_j(t),$ где $\lambda (0 \leq \lambda \leq 1)$ - моментный параметр, обычно $\lambda = 0.9$
<i>Проблема выбора скорости обучения:</i> при слишком маленькой скорости, процесс обучения может быть неоправданно долгим, при слишком большой, есть опасность «проскочить» решение.	Адаптивный шаг обучения

#### 4. РЕЛАКСАЦИОННЫЕ НЕЙРОННЫЕ СЕТИ

*Релаксационные сети* – это сети, в которых распространение сигнала происходит как в прямом, так и в обратном направлении. На каждом такте работы НС один или несколько нейронных элементов обрабатывают информацию, полученную на предыдущем шаге, т.е. функционирование релаксационных НС носит итеративный характер. Такая обработка информации происходит до тех пор пока все нейронные элементы не перейдут состояние равновесия. При этом состояния нейронных элементов перестает изменяться. Мы будем рассматривать релаксационные НС на примере сети Хопфилда [1,4].

##### 4.1. НЕЙРОННАЯ СЕТЬ ХОПФИЛДА

Архитектура НС Хопфилда представлена на рис. 8. Первый слой называется распределительным, второй – вычислительным. Выход НС в момент времени  $t+1$  вычисляется следующим образом:

$$y_i(t+1) = F \left( \sum_{\substack{j=1 \\ j \neq i}}^n w_{ji} y_j(t) - T_i \right).$$

Так как в данной сети все нейронные элементы соединены друг с другом, то матрица весовых коэффициентов является симметричной с нулями на главной диагонали:

$$W = \begin{pmatrix} 0 & w_{12} & \dots & w_{1n} \\ w_{21} & 0 & \dots & w_{2n} \\ \dots & \dots & \dots & \dots \\ w_{n1} & w_{n2} & \dots & 0 \end{pmatrix}, w_{ij} = w_{ji}$$

НС Хопфилда представляет собой динамическую систему, состояния которой характеризуется фазовым пространством. Функционирование сети можно представить как движение в этом пространстве к точке равновесия. Проведя биологическую аналогию, можно определить данный процесс, как процесс вспоминания информации.

Сеть Хопфилда с дискретным временем и пороговой функцией активации называется *сетью с дискретным состоянием и временем*; с непрерывной функцией активации - *сетью с непрерывным состоянием и дискретным временем*; с непрерывным временем – *непрерывной сетью*. Также выделяют сети с последовательной и параллельной динамикой. В случае с последовательной динамикой в каждый момент времени функционирует только один нейронный элемент с номером  $k$ , который выбирается случайным образом, при параллельной – своё состояние изменяют сразу все нейроны сети.

Последовательная динамика	Параллельная динамика
$y_i(t+1) = y_i(t), \forall i \neq k;$ $y_i(t+1) = F\left(\sum_{\substack{j=1 \\ i \neq j}}^n w_{ji}y_j(t) - T_i\right), \text{ при } i = k.$	$y_i(t+1) = F\left(\sum_{\substack{j=1 \\ i \neq j}}^n w_{ji}y_j(t) - T_i\right), \forall i.$

Одно из основных свойств сети Хопфилда это способность распознать зашумлённые сигналы. Представим входные образы, которые необходимо запомнить в виде матрицы:

$$Y = \begin{pmatrix} y_1^1 & y_2^1 & \dots & y_n^1 \\ y_1^2 & y_2^2 & \dots & y_n^2 \\ \dots & \dots & \dots & \dots \\ y_1^L & y_2^L & \dots & y_n^L \end{pmatrix}.$$

На практике обычно используют пороговую функцию активации, бинарные входные векторы и нулевые пороги нейронных элементов. В этом случае матрица весовых коэффициентов представим в виде:

$$W = Y^T Y - I.$$

Алгоритма функционирования НС Хопфилда выглядит следующим образом:

1. Используя образы, которые необходимо запомнить, вычисляется матрица весовых коэффициентов  $W$ .
2. На вход сети подаётся зашумлённый образ.
3. До тех пор пока  $y_i(t+1) \neq y_i(t), \forall i$  вычисляются состояния нейронных элементов.
  - при последовательной динамике:

$$y_i(t+1) = y_i(t), \forall i \neq k;$$

$$y_i(t+1) = F \left( \sum_{\substack{j=1 \\ i \neq j}}^n w_{ji} y_j(t) \right), \text{ при } i = k.$$

Нейронный элемент  $k$  выбирается случайным образом;

– при параллельной динамике:

$$y_i(t+1) = F \left( \sum_{\substack{j=1 \\ i \neq j}}^n w_{ji} y_j(t) \right), \forall i.$$

Равенство  $y_i(t+1) = y_i(t), \forall i$  означает, что сеть пришла в установившееся состояние и  $y_i(t), \forall i$  и есть распознанный образ.

Приведём пример. Пусть необходимо, что бы сеть запомнила следующий образ:  $Y = (0 \ 1 \ 1 \ 0)$ . НС Хопфилда будет состоять из четырёх соединённых друг с другом нейронных элементов (рис. 9). Матрица весовых коэффициентов примет вид:

$$W = Y^T Y - I = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \end{pmatrix} (0 \ 1 \ 1 \ 0) - \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}$$

Пусть теперь на вход подаётся зашумлённый образ:  $Y = (0 \ 1 \ 1 \ 1)$ . Рассмотрим алгоритм функционирования сети при параллельной динамике. Так как мы используем пороговую функцию активации, то  $y(t+1) = \text{sign}(Y(t)W)$ .

$t = 1$ :

$$Y(1) = \text{sign}(Y(0)W) = \text{sign} \left( (0 \ 1 \ 1 \ 1) \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} \right) = \text{sign}(0 \ 1 \ 1 \ -1) = (0 \ 1 \ 1 \ 0);$$

$t = 2$ :

$$Y(2) = \text{sign}(Y(1)W) = \text{sign} \left( (0 \ 1 \ 1 \ 0) \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} \right) = \text{sign}(0 \ 1 \ 1 \ 0) = (0 \ 1 \ 1 \ 0);$$

Как можно заметить состояние НС перестало изменяться, т.е. сеть за одну итерацию пришла в устойчивое положение и распознала зашумлённый образ.

## 5. САМООРГАНИЗУЮЩИЕСЯ НЕЙРОННЫЕ СЕТИ КОХОНЕНА

В предыдущих разделах речь шла о нейронных сетях, которые перед началом работы необходимо было обучить, т.е. найти «правильные» пороги и весовые коэффициенты. Однако, существует тип сетей, которые не нуждаются в предварительном обучении. Такие сети, называемые *самоорганизующимися*, подстраивают свои весовые коэффициенты непосредственно о время работы. На практике наибольшее распространения получили сети Кохонена. В общем случае они представляют собой двухслойную нейронную сеть, разбивающие входной  $n$ -мерное пространство на  $m$  областей (*кластеров*), каждой из которой соответствует один нейронный элемент, т.е. НС Кохонена состоит из  $n$  нейронов входного слоя и  $m$  нейронов выходного слоя. В качестве процедуры обучения используется метод, получивший название *конкурентное обучение*.

### 5.1. КОНКУРЕНТНОЕ ОБУЧЕНИЕ

Нейронные сети, обучение которых происходит по методу конкурентного обучения, называются *конкурентными*. Как было сказано выше, конкурентная НС представляет собой двухслойную сеть прямого распространения сигнала. Процедура обучения выглядит следующим образом:

**Шаг 1.** Случайным образом инициализируются весовые коэффициенты НС -  $W$ .

**Шаг 2.** Задаётся начальное значение номера итерации -  $t = 1$ .

**Шаг 3.** На вход НС поочерёдно подаются все  $l = 1 \dots L$  входных образов. Для каждого образа выполняются следующие операции:

- определяется номер  $k$  *нейрона-победителя*, для которого  $S_k = \max_j S_j, j = 1 \dots m$ ,
- рассчитывается активность выходных нейронов:  $y_j = F(S_j) = \begin{cases} 1, & j = k; \\ 0, & j \neq k. \end{cases}$
- изменяются весовые коэффициенты нейрона-победителя по следующему правилу:  $W_k(t+1) = W_k(t) + \gamma(X^l - W_k(t))$ , где параметр  $0 < \gamma < 1$  называется скоростью обучения.

**Шаг 4.** Увеличивается номер итерации -  $t = t + 1$ ;

**Шаг 5.** Процедура обучения повторяется с шага 3 до тех пор пока не будет получена необходимая степень кластеризации или не перестанут изменяться весовые коэффициенты.

В некоторых источниках советуют уменьшать скорость обучения с течением времени, например, по закону  $\gamma = 1/t$  [1]. Также следует отметить, что для правильного определения нейрона-победителя рекомендуют нормировать как входные образы, так и векторы весовых коэффициентов.

На практике иногда изменяют весовые коэффициенты не только у нейрона-победителя но и у соседних нейронов. В случае использования этой схемы, формула модификации весовых коэффициентов примет следующий вид:  $W_k(t+1) = W_k(t) + \gamma(X - W_k(t)), k \in V$ , где  $V$  множество нейронов, весовые коэффициенты которых необходимо изменять, т.е. множество соседей нейрона-победителя.

Если изменить топология конкурентной НС таким образом, что нейроны выходного слоя будут расположены на плоскости, то мы получим *самоорганизующуюся карту Кохонена*.

## 5.2. САМООРГАНИЗУЮЩИЕСЯ КАРТЫ КОХОНЕНА

Топология самоорганизующейся карты Кохонена представлена на рис. 10. Данная НС состоит из нейронов входного слоя, выполняющих распределительную функцию, и вычислительного слоя, нейроны которого располагаются на плоскости – слой Кохонена. Цель функционирования сети выполнить топологическое упорядочивание входных сигналов, т.е. «соседним» входным образом соответствуют соседние нейронные элементы вычислительного слоя. Для решения этой задачи вводят *область притяжения* нейрона-победителя, таким образом, что все нейроны, попадающие в эту область изменяют свои весовые коэффициенты и как бы «притягиваются» к нейрону-победителю. Область притяжения можно задать в с помощью *функции притяжения*  $h(t, k, p)$ , где  $t$  - время,  $k$  - номер нейрона победителя и  $p$  - номера текущего нейрона. На практике, в качестве функции  $h$  обычно используют функцию Гаусса:

$$h(t, k, p) = e^{-\frac{|u_k - u_p|^2}{2\sigma^2(t)}} = e^{-\frac{-(i_k - i_p)^2 + (j_k - j_p)^2}{2\sigma^2(t)}},$$

где  $u_k$  - нейрон-победитель с номером  $k$  и координатами  $(i_k, j_k)$  в матрице выходного слоя сети (нейроны выходного слоя лежат в плоскости, образуя матрицу);  $u_p$  - текущий нейрон с номером  $p$  и координатами  $(i_p, j_p)$  в матрице выходного слоя сети;  $u_k - u_p$  - расстояние между нейроном-победителем  $k$  и текущим нейроном  $p$ ;  $\sigma(t)$  - характеризует радиус области притяжения и является убывающей функцией времени, т.е. радиус области притяжения со временем уменьшается. Таким образом, в какой-то момент времени, область притяжения «сожмётся» до одного нейрона-победителя и весовые коэффициенты будут изменять только у него.

Далее представлен алгоритм обучения самоорганизующихся сетей. Пусть  $n$  - размерность входных образов и мы хотим разделить все  $L$  входных образов на  $M$  кластеров. Тогда самоорганизующаяся карта Кохонена имеет  $n$  распределительных нейронов и  $M$  нейронов в вычислительном слое. Представим эти  $M$  нейронов в виде матрицы размерностью  $m_1 \times m_2$ , тогда каждый нейрон вычислительного слоя имеет координаты  $(i, j)$ , где  $i = 1..m_1, j = 1..m_2$ , а вектор весовых коэффициентов нейрона с координатами  $(i, j)$  запишется в виде:  $W_{ij} = (w_{1ij} \ w_{2ij} \ \dots \ w_{nij})$ . Для такой НС процедура обучения будет выглядеть следующим образом:

**Шаг 1.** Случайным образом инициализируются весовые коэффициенты –  $W$ .

**Шаг 2.** Номер итерации (время) устанавливается  $t=1$ , задают функцию  $\sigma(t)$  и определяют начальное значение радиуса притяжения -  $\sigma(1)$ .

**Шаг 3.** Для всех  $l=1..L$  входных образов выполняются следующие операции:

- вычисляется норма вектор  $D_{ij} = (X^l - W_{ij})$ , где  $i = 1..m_1, j = 1..m_2$ ;
- определяется номер и координаты нейрона-победителя:  $D_{k_1 k_2} = \min_{i,j} D_{ij}$ , где нейрон победитель имеет номер  $k$  и координаты  $(k_1, k_2)$ ;

- для каждого нейрона вычисляется функция притяжения:

$$h(t, k, p) = e^{-\frac{(k_1 - i_p)^2 + (k_2 - j_p)^2}{2\sigma^2(t)}};$$

- выполняется модификация весовых коэффициентов по следующему правилу:  $W_{ij}(t+1) = W_{ij}(t) + \gamma h(t, k, p)(X^l - W_{ij}(t))$ , где  $\gamma$  - характеризует скорость обучения и может быть либо постоянной, либо уменьшаться с течением времени, например по закону  $\gamma = 1/t$ .

**Шаг 4.** Увеличивается номер итерации -  $t = t + 1$  и вычисляется новое значение радиуса притяжения – функции  $\sigma(t)$ ;

**Шаг 5.** Процедура обучения повторяется с шага 3 до тех пор, пока не будет получена необходимая степень кластеризации, или не перестанут изменяться весовые коэффициенты.

Методология применения самоорганизующихся карт Кохонена выглядит следующим образом:

1. Тренируют сеть на обучающей выборке. Так как размерность входных образов  $n$ , то довольно часто в один кластер попадают образы, которые «с первого взгляда» мало похожи.
2. На «обученную» сеть подают новый образ, определяют нейрон-победитель и выявляют кластер, соответствующий этому нейрону. Таким образом, делают вывод, что новый образ обладает свойствами, присущими всем образам, попавшим в данный кластер.

## ЛИТЕРАТУРА

1. Головкин В.А. Нейронные сети: обучение, организация и применение. Кн.4: Учеб. пособие для вузов /Общая ред. А.И.Галушкина.-М.:ИПРЖР,2001.- 256с.:ил.
2. Хайкин С. Нейронные сети: полный курс, 2-е издание.: Пер. с англ. – М.: Издательский дом «Вильямс», 2006. – 1104 с.: ил.
3. Каллан Р. Основные концепции нейронных сетей.: Пер. с англ. –М.: Издательский дом «Вильямс», 2001.287-с.:ил.
4. Круглов В.В., Борисов В.В. Искусственные нейронные сети. Теория и практика. – М.: Горячая линия – Телеком, 2001. – 382 с.: ил.
5. Смолин Д.В. Введение в искусственный интеллект. – М.: ФИЗМАТЛИ, 2004. – 208 с.
6. Ясницкий Л.Н. Введение в искусственный интеллект. – М.: Издательский центр «Академия», 2005. – 176 с.
7. Э. Хант. Искусственный интеллект. М.: Мир, 1978 – 560 с.
8. Петрин А.А. Основы теоретической робототехники. Нечёткие методы. (Обзор).//М.:Препринт ИПМ им. М.В.Келдыш РАН, 2005, №34, 23 с.
9. Голубев Ю.Ф. Нейросетевые методы в мехатронике. //М.:МГУ, 2007, 157 с.

## РИСУНКИ

Рис. 1. Схема биологического нейрона.

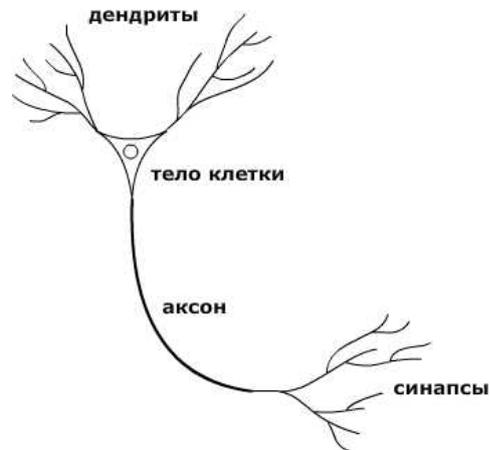


Рис. 2. Схема искусственного нейрона.

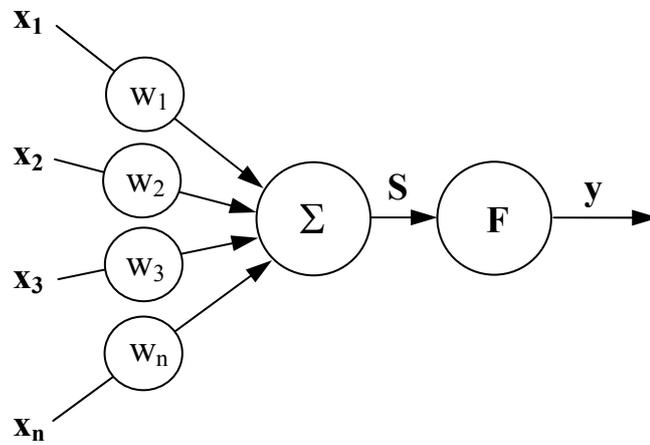
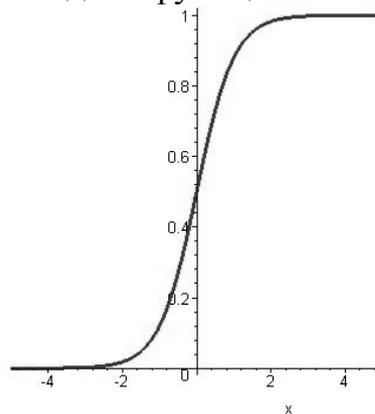
Рис. 3. Сигмоидная функция активации ( $c=2$ ).

Рис. 4. Однослойная нейронная сеть прямого распространения.

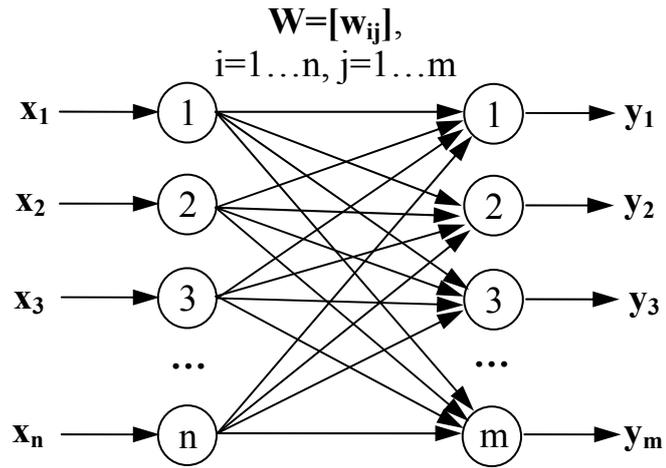


Рис. 5. Нейронная сеть, состоящая из 2 нейронов входного слоя и 1 нейрона выходного слоя (с порогом).

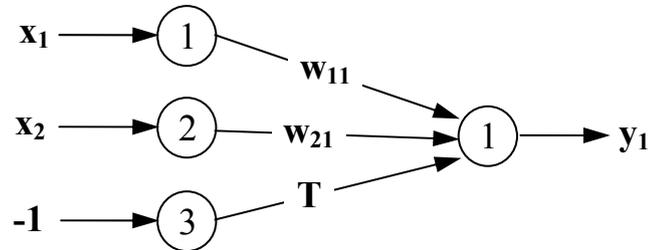


Рис. 6. ADALINE (ADAPtive LINear Element).

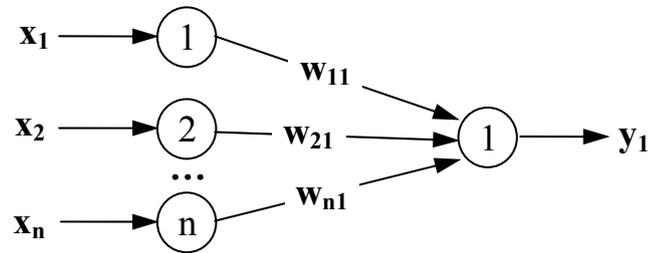


Рис. 7. Топология многослойной нейронной сети.

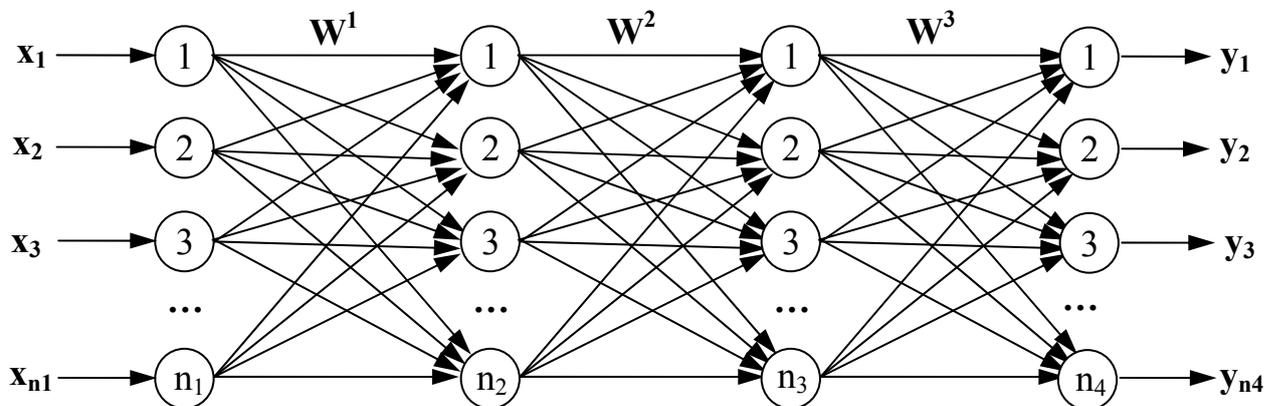


Рис. 8. Архитектура нейронной сети Хопфилда.

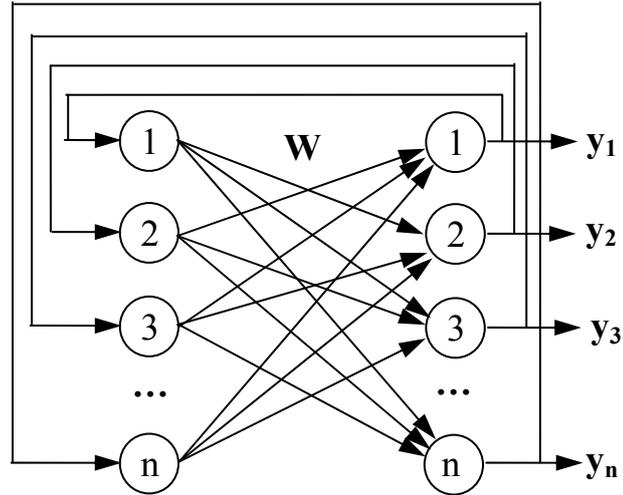
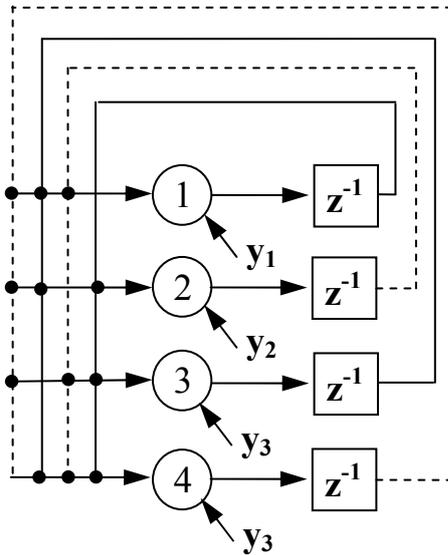
Рис. 9. Сеть Хопфилда, состоящая из четырёх нейронных элементов ( $z^{-1}$  - оператор единичной задержки).

Рис. 10. Топология самоорганизующейся карты Кохонена.

