

**Севастопольский государственный университет**  
**Институт информационных технологий**

Дополнительная профессиональная программа профессиональной  
переподготовки «Глубокие нейросети в компьютерном зрении»

**Основы нейронных сетей**

**Лекция 1**

**Модели нейронов и архитектура нейросетей**

**Бондарев Владимир Николаевич**

# Цели и задачи дисциплины

**Целью дисциплины** является формирование профессиональных компетенций в области проектирования базовых архитектур нейронных сетей и правил их обучения.

## **Задачи дисциплины:**

- обучение студентов основным понятиям нейронных сетей;
- обучение основным теоретическим принципам и алгоритмам поиска оптимума целевых функций;
- формирование компетенций в области обучения базовых архитектур нейронных сетей: однослойного и многослойного персептрона, линейного адаптивного элемента, состязательных сетей, сетей на основе радиальных базисных функций;
- формирование практических умений по использованию инструментальных средств разработки нейронных сетей.

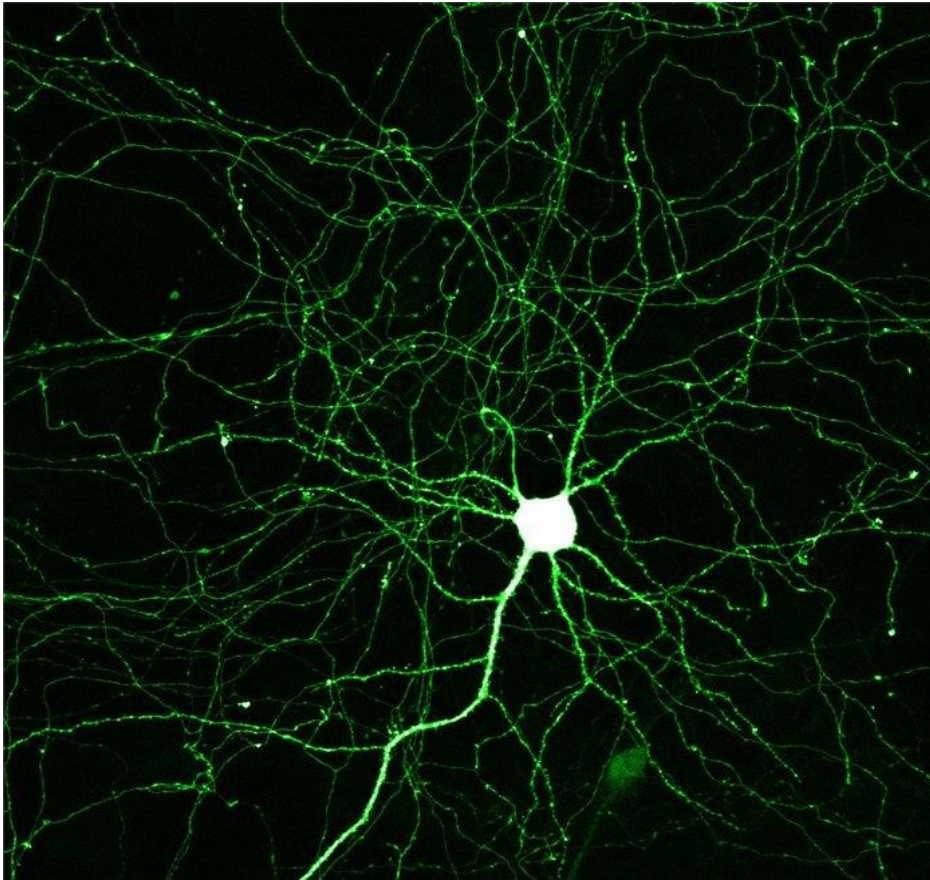
## Содержание учебной дисциплины

1. Модели нейронов и архитектура нейросетей. Правило обучения персептрона.
2. Целевые функции нейросетей и условия оптимумов. Алгоритмы оптимизации нейросетей.
3. Линейный адаптивный элемент и правило обучения Уидроу-Хоффа.
4. Алгоритм обратного распространения ошибок (BP).
5. Вариации BP. Эвристические модификации BP. Алгоритм сопряженных градиентов. Алгоритм Левенберга-Марквардта
6. Методы улучшения обобщающих способностей сетей. Регуляризация.
7. Обучение без учителя. Правило Хебба. Состязательные сети и правило Кохонена. SOM. Обучение на основе правила LVQ.
8. Радиальные базисные сети (RBF).

# Литература

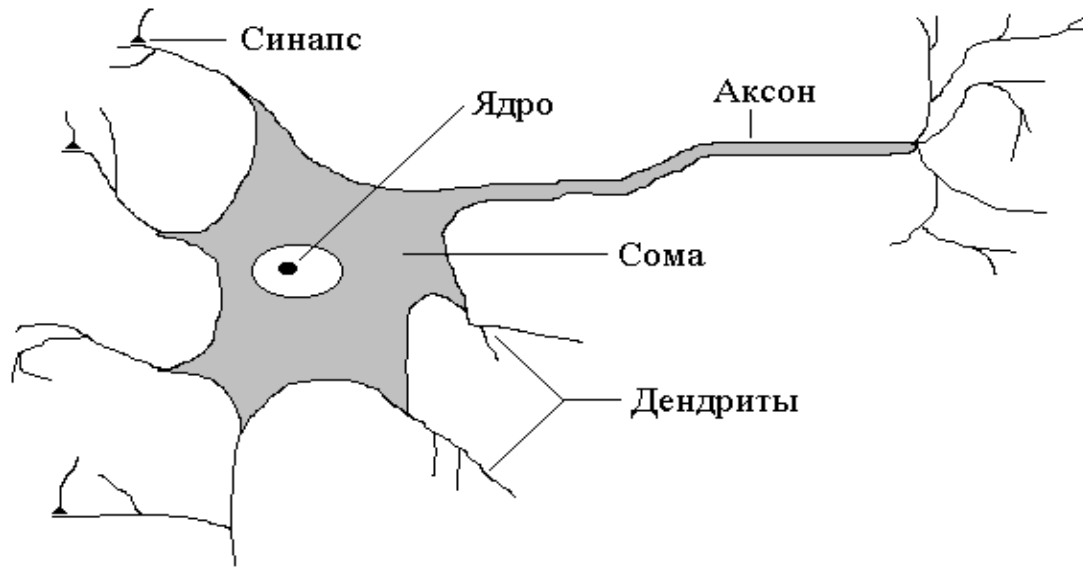
1. Бондарев В.Н. Искусственный интеллект: Учеб. пособие для студентов вузов / В. Н. Бондарев, Ф. Г. Аде. — Севастополь: Изд-во СевНТУ, 2002. — 613 с.  
[https://www.researchgate.net/profile/Vladimir\\_Bondarev/publications](https://www.researchgate.net/profile/Vladimir_Bondarev/publications)
2. Hagan M.T., Demuth H.B., Beale M.H., Jesus O. Neural Network Design. 2-nd Edition.— Frisco, Texas, 2014 – 1012 p. <http://hagan.okstate.edu/NNDesign.pdf>
3. Beale M.H., Hagan M.T., Demuth H.B. Neural Network Toolbox. User's Guide. – Natick: Math Works, Inc., 2014. – 435 p.
4. Хайкин С. Нейронные сети: Полный курс. Пер. С англ. / С. Хайкин. — М.: Изд. «Вильямс», 2006. — 1104 с.
5. Галушкин А.И. Нейронные сети: история развития теории / А.И. Галушкин, Я.З. Цыпкин. М.: Изд-во «Альянс», 2015. — 840 с.
6. Медведев, В.С. Нейронные сети. *MATLAB 6* / В.С. Медведев, В.Г. Потемкин; под общ. ред. В.Г. Потемкина. — М.: ДИАЛОГ-МИФИ, 2002. — 496 с.

# Нервная система человека



Мозг человека содержит  
 $10^{11}$  нейронов.  
Каждый нейрон имеет  $10^3$   
- $10^5$  соединений.

# Биологический нейрон

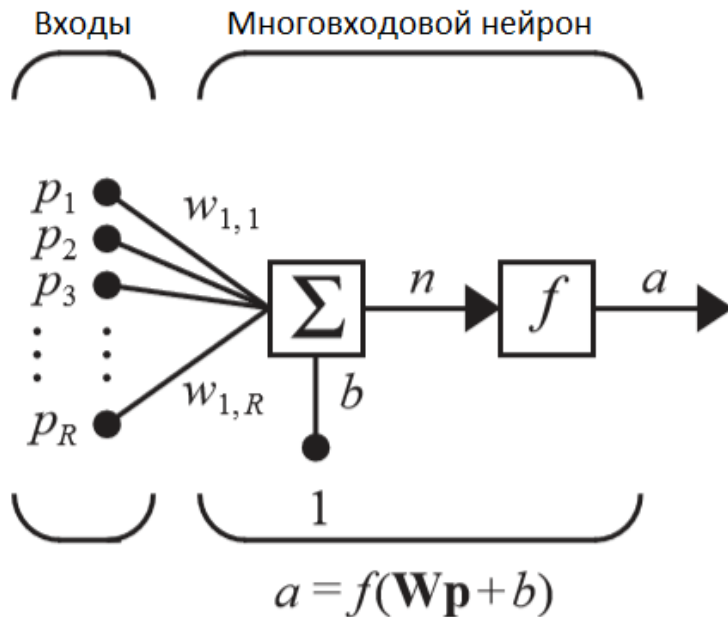


**Дендриты** – ветвеобразные отростки, которые обеспечивают сбор сигналов от других нейронов или рецепторов.

**Сомма** нейрона представляет тело клетки. В сомме происходят сложные биохимические процессы, благодаря которым осуществляются нелинейные преобразования сигналов, поступающих через дендриты.

**Аксон** является отростком клетки, по которому ее выходной сигнал поступает на дендриты других нейронов. Аксон разветвляется на большое число волокон. Место соединения волокон с дендритами называется **синапсом**.

# Формальный нейронный элемент (НЭ) Маккаллоха-Питтса



$p_1, p_2, \dots, p_R$  — ВХОДНЫЕ СИГНАЛЫ

$w_{11}, w_{12}, \dots, w_{1R}$  — веса синаптических связей

Сетевой выход  $n$  равен:

$$n = w_{1,1}p_1 + w_{1,2}p_2 + \dots + w_{1,R}p_R + b.$$

Или матричной форме

$$n = \mathbf{W}\mathbf{p} + b,$$

где  $\mathbf{W}$  — матрица весов (в данном случае одна строка, размером  $1 \times R$ ),  $\mathbf{p}$  — вектор входных сигналов,  $b$  — смещение (порог);  $R$  — количество входов НЭ.

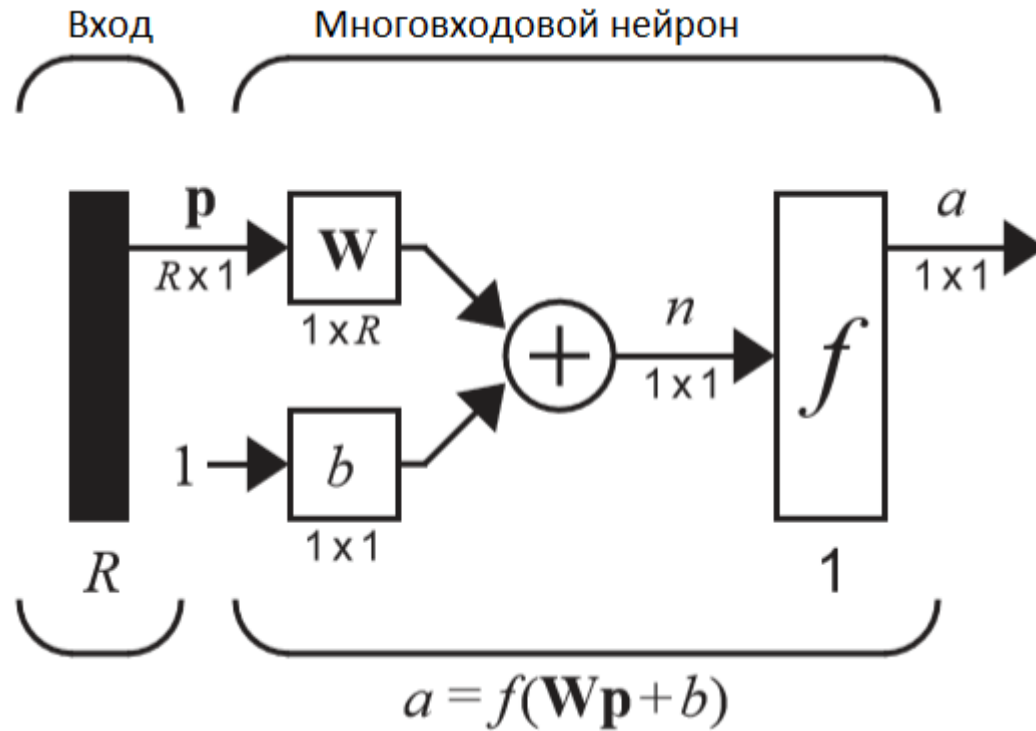
Выход нейрона  $a$  (активность на выходе)

$$a = f(\mathbf{W}\mathbf{p} + b).$$

Функция преобразования  $f(n)$  для НЭ Маккалоха-Питтса математически соответствует функции Хевисайда  $H(n)$

$$f(n) = H(n) = \begin{cases} 1, & n \geq 0 \\ 0, & n < 0 \end{cases}$$

# Многовходовой нейрон – упрощенное графическое обозначение





# Сетевые функции НЭ

Выражение 
$$n = \mathbf{W}\mathbf{p} + b = \sum_{j=1}^R w_{1j} p_j + b = \sum_{j=1}^{R+1} w_{1j} p_j, \quad (1)$$

где  $w_{1,R+1}=b$  и  $p_R=1$  определяет способ объединения входных признаков и называется *сетевой* функцией НЭ.

Формальные модели нейронов отличаются между собой видом *сетевой функции* и видом *функции преобразования*  $f(.)$ .

Кроме линейной сетевой функции (1), в ИНС широко используются следующие сетевые функции:

а) *квадратическая*

$$n = \sum_{j=1}^R w_{1j} p_j^2 \quad (2)$$

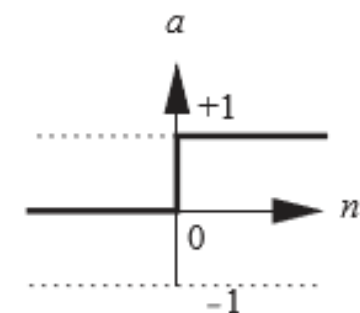
б) *радиальная (сферическая)*

$$n = \sum_{j=1}^R (p_j - w_{1j})^2 \quad (3)$$

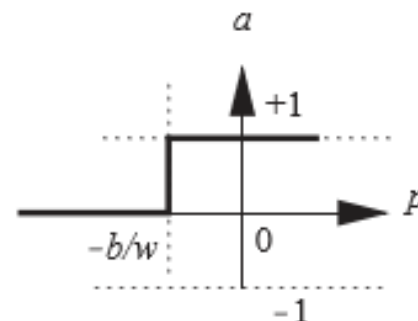
# Функции преобразования НЭ (функции активации)

Часто используемые функции:

1) пороговая функция (единичная функция Хевисайда)

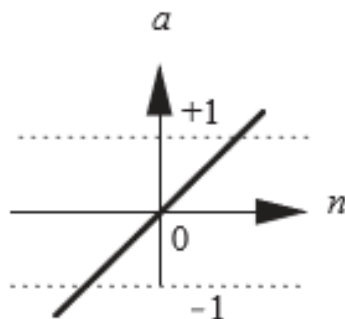


$$a = \text{hardlim}(n)$$

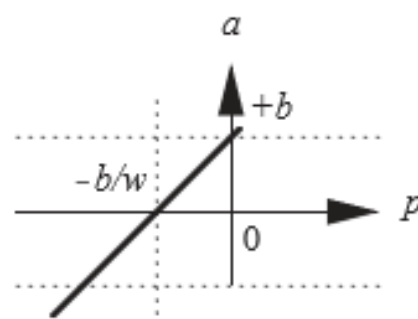


$$a = \text{hardlim}(wp + b)$$

2) линейная функция :  $a=n$



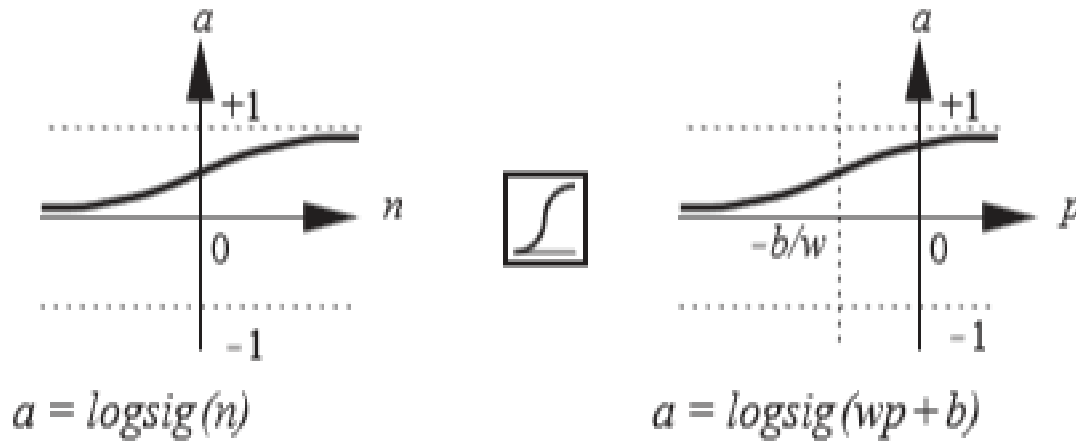
$$a = \text{purelin}(n)$$



$$a = \text{purelin}(wp + b)$$










# Функции преобразования НЭ (функции активации)

3) униполярная сигмовидная функция (лог-сигмовидная)



$$a = \frac{1}{1 + e^{-n}}$$

# Функции преобразования НЭ (функции активации)

Hard Limit	$a = 0 \quad n < 0$ $a = 1 \quad n \geq 0$		hardlim
Symmetrical Hard Limit	$a = -1 \quad n < 0$ $a = +1 \quad n \geq 0$		hardlims
Linear	$a = n$		purelin
Saturating Linear	$a = 0 \quad n < 0$ $a = n \quad 0 \leq n \leq 1$ $a = 1 \quad n > 1$		satlin
Symmetric Saturating Linear	$a = -1 \quad n < -1$ $a = n \quad -1 \leq n \leq 1$ $a = 1 \quad n > 1$		satlins
Log-Sigmoid	$a = \frac{1}{1 + e^{-n}}$		logsig
Hyperbolic Tangent Sigmoid	$a = \frac{e^n - e^{-n}}{e^n + e^{-n}}$		tansig
Positive Linear	$a = 0 \quad n < 0$ $a = n \quad 0 \leq n$		poslin
Competitive	$a = 1 \quad \text{neuron with max } n$ $a = 0 \quad \text{all other neurons}$		compet

SciLab NeuralNetworks 2.0  
ann\_hardlim\_activ

ann\_purelin\_activ

ann\_logsig\_activ

ann\_tansig\_activ

ann\_compet\_activ

# Функции преобразования НЭ (функции активации)

```
function y = ann_tansig_activ(x)
```

```
// биполярная тангенциальная сигмовидная функция активации
```

```
// вызов: y = ann_logsig_activ(x)
```

```
//
```

```
// параметры
```

```
// x : ВХОДНОЙ вектор
```

```
// y : ВЫХОДНОЙ вектор
```

$$\begin{aligned} y &= (\exp(n) - \exp(-n)) / (\exp(n) + \exp(-n)) = \\ &= (\exp(n) - \exp(-n)) / (\exp(n) + \exp(-n)) * (\exp(-n) / \exp(-n)) = \\ &= \dots = 2 / (1 + \exp(-2n)) - 1 \end{aligned}$$

```
y = 2 ./ (1 + exp(-2.*x)) - 1;
```

```
endfunction
```

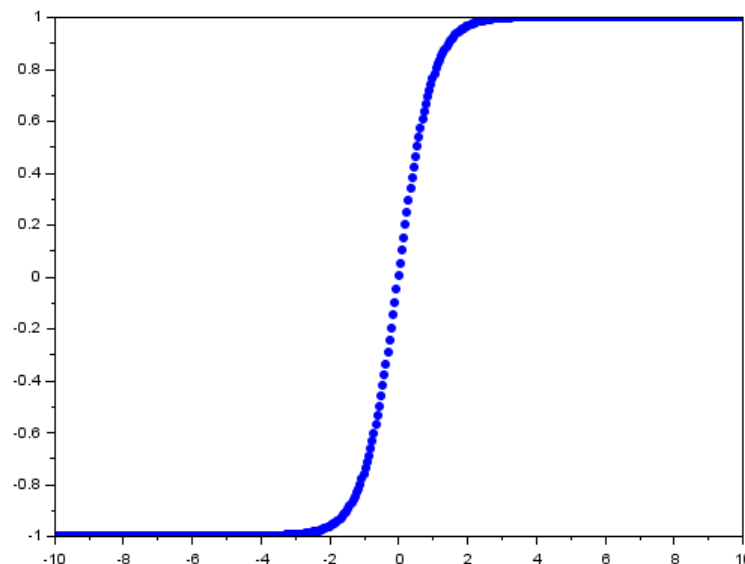
SciLab

```
// пример вызова
```

```
x = [-10:0.05:10];
```

```
y = ann_tansig_activ(x);
```

```
plot(x,y,'.');
```

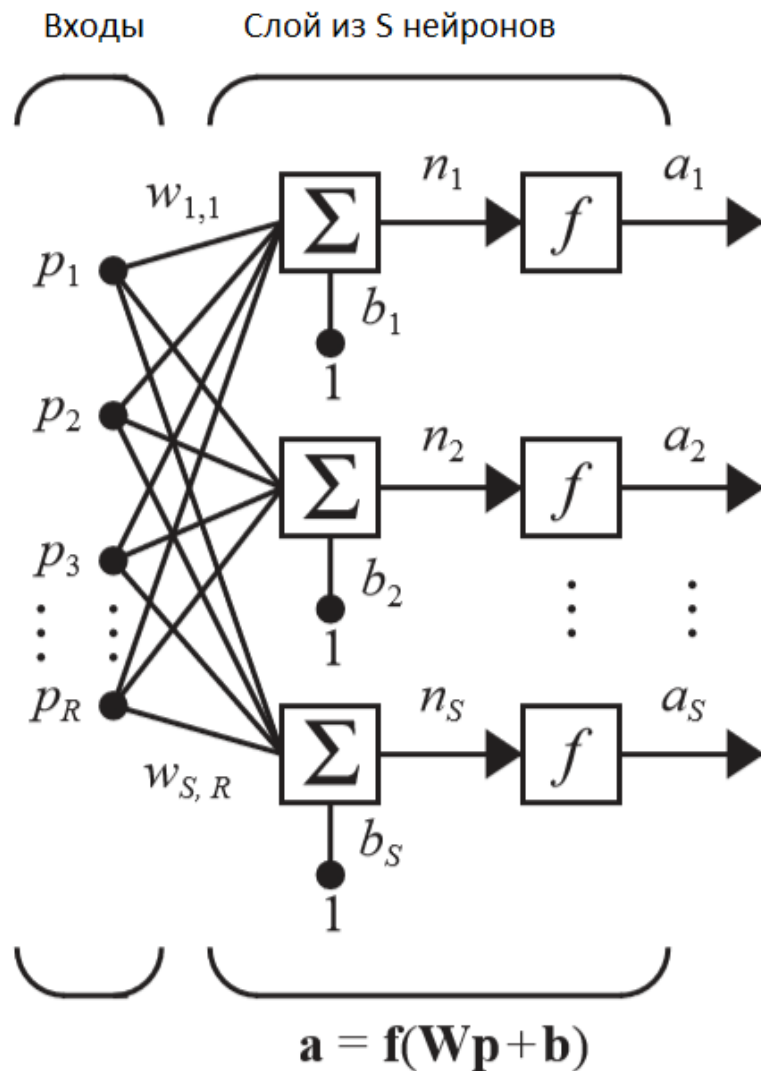


# Архитектура нейронных сетей

ИНС состоит из большого числа взаимосвязанных НЭ.

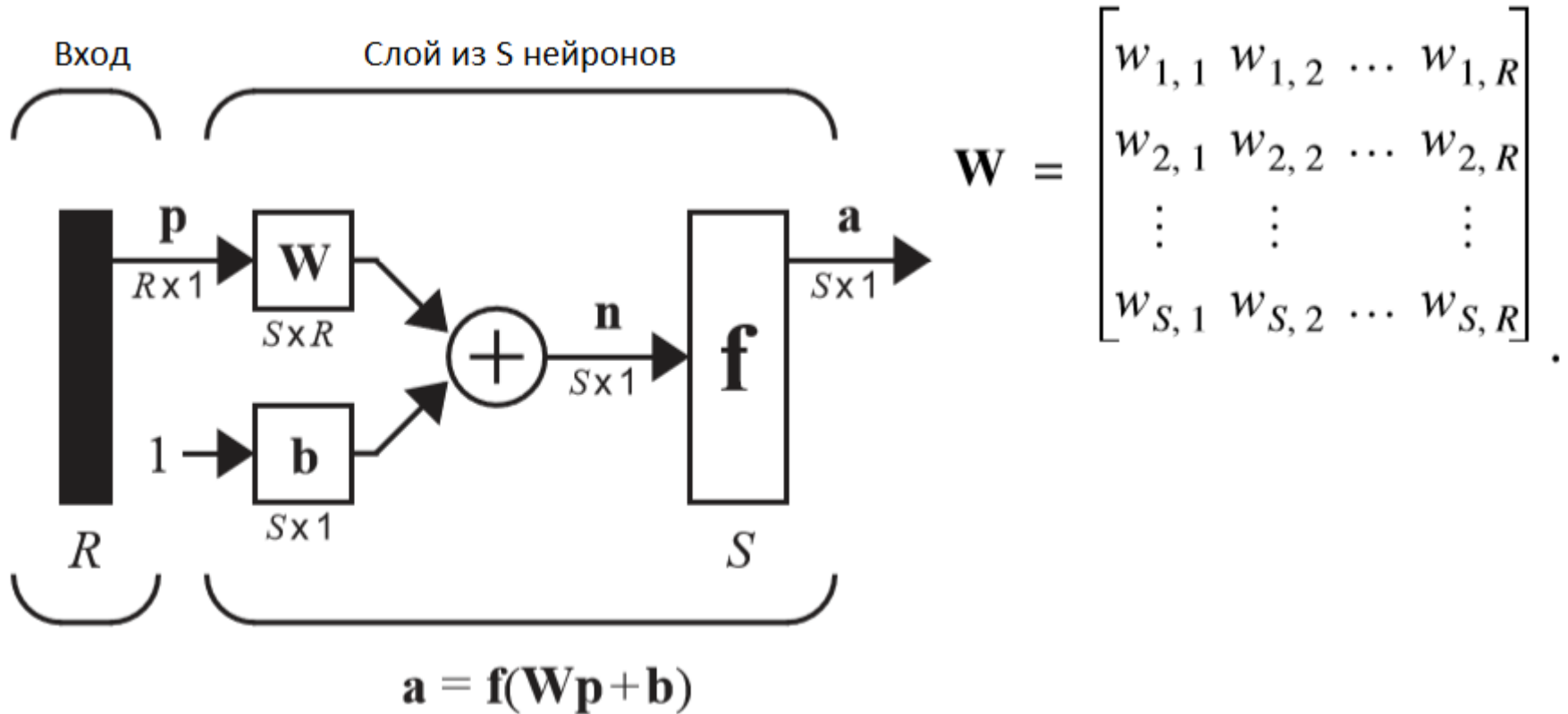
Выделяют следующие основные разновидности архитектур ИНС:

- **однослойные ИНС с прямыми связями;**



$$\mathbf{W} = \begin{bmatrix} w_{1,1} & w_{1,2} & \dots & w_{1,R} \\ w_{2,1} & w_{2,2} & \dots & w_{2,R} \\ \vdots & \vdots & & \vdots \\ w_{S,1} & w_{S,2} & \dots & w_{S,R} \end{bmatrix}.$$

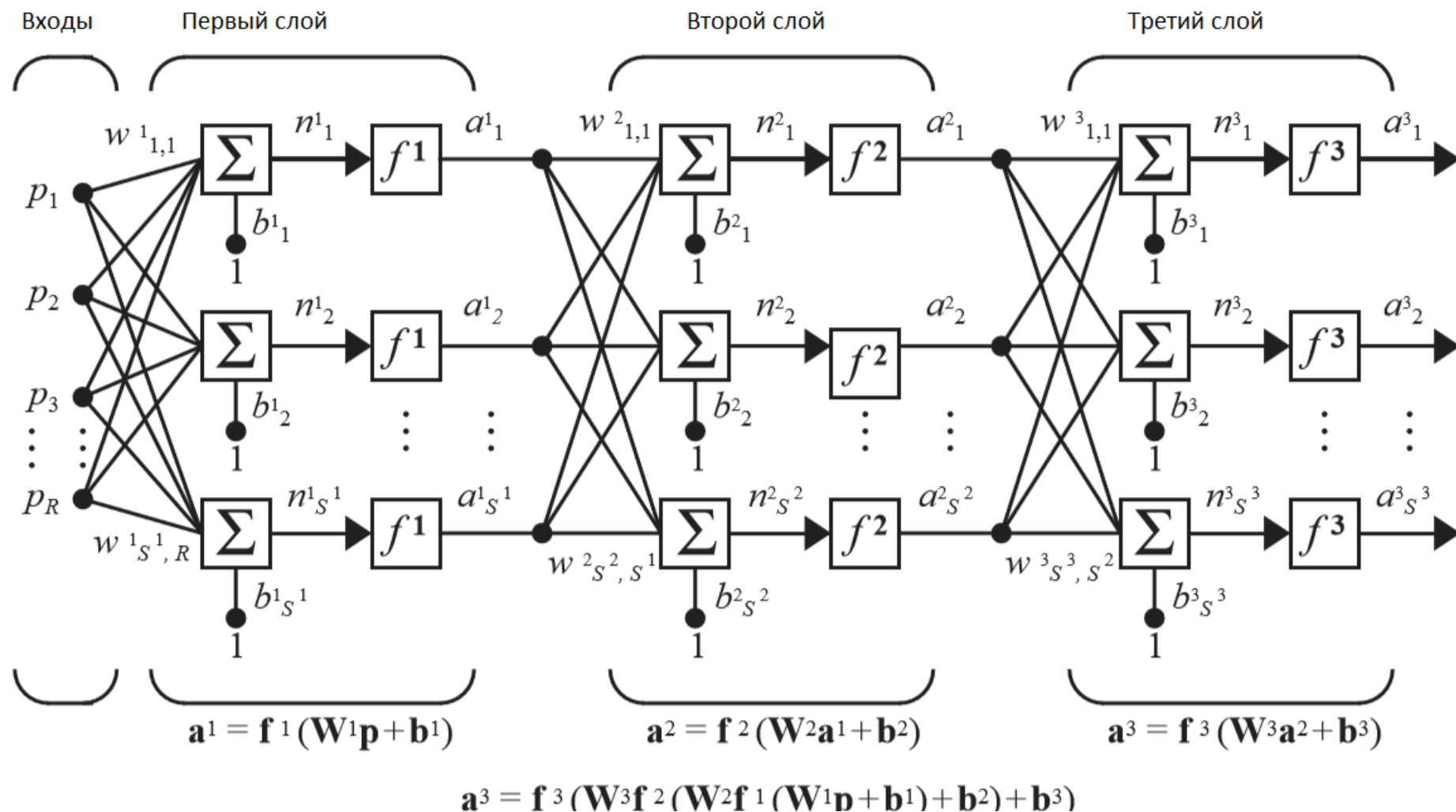
# Слой из $S$ нейронов – упрощенное обозначение



# Архитектура нейронных сетей

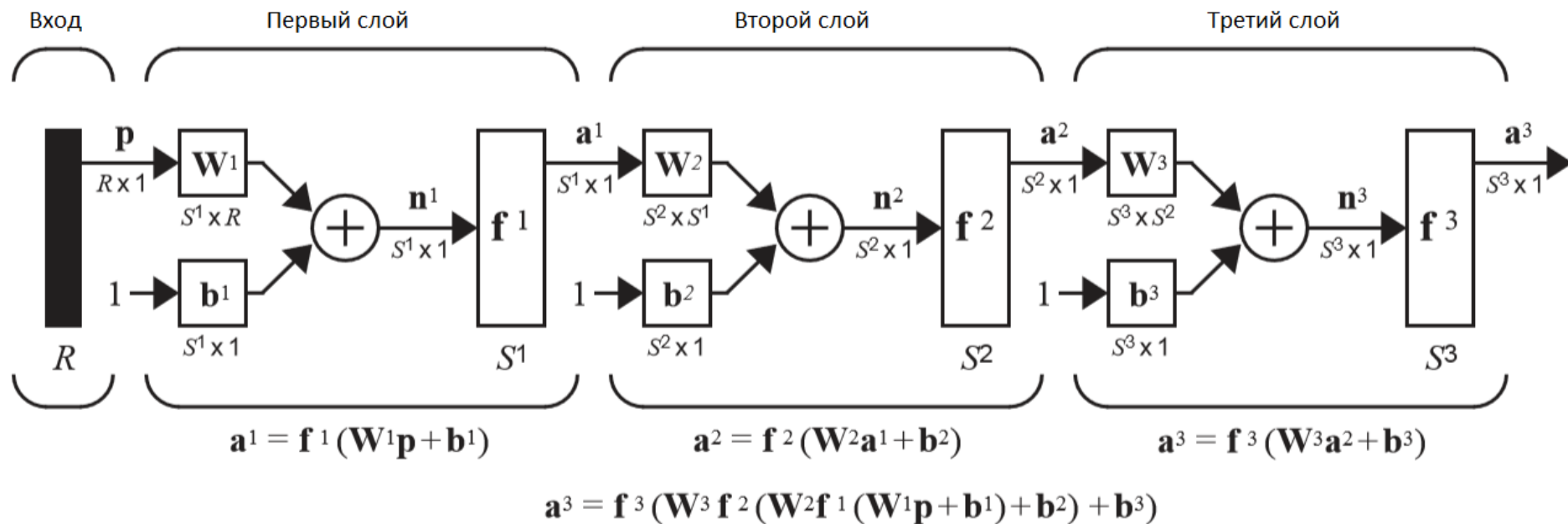
- многослойные ИНС с прямыми связями

## Трехслойная сеть





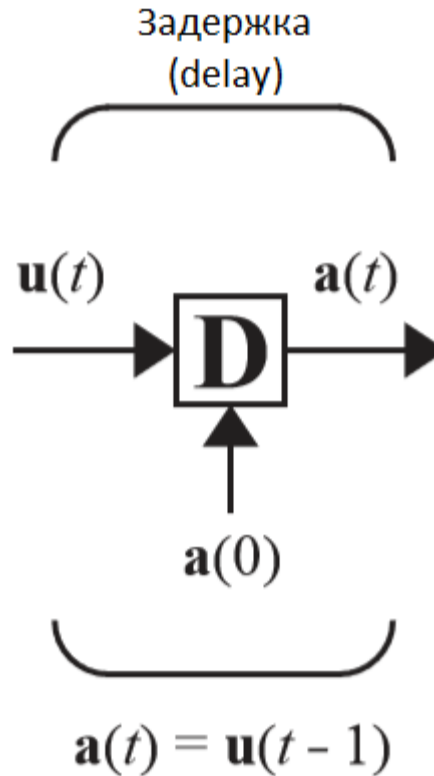
# Трехслойная сеть – упрощенное обозначение



# Архитектура нейронных сетей

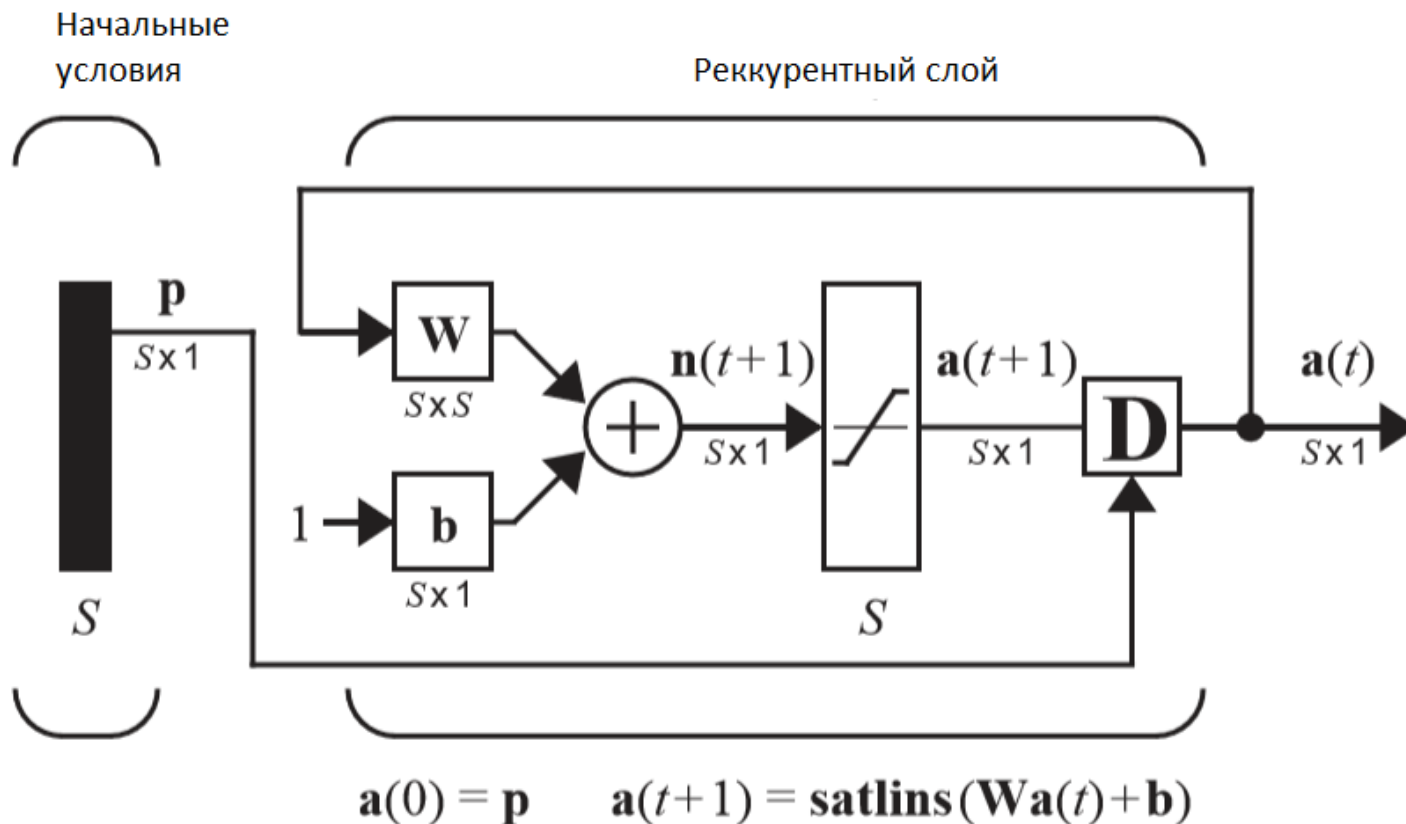
## - рекуррентные (сети с обратными связями)

Для организации обратной связи сети используют блок задержки



# Реккурентные сети

Это сети с обратными связями, где некоторые из выходов соединяются со входами



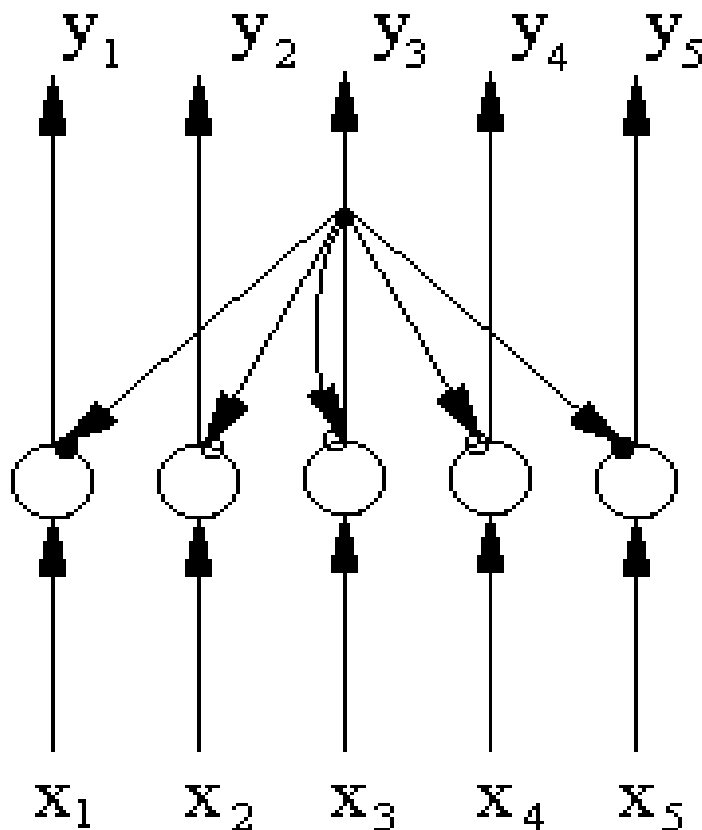
Последующие значения выходов вычисляются через их предыдущие значения

$$\mathbf{a}(1) = \text{satlins}(\mathbf{W}\mathbf{a}(0) + \mathbf{b}), \mathbf{a}(2) = \text{satlins}(\mathbf{W}\mathbf{a}(1) + \mathbf{b}), \dots$$

# Архитектура нейронных сетей

Обратные связи, действующие в пределах одного слоя НЭ, называют *латеральными* (от английского lateral – горизонтальный).

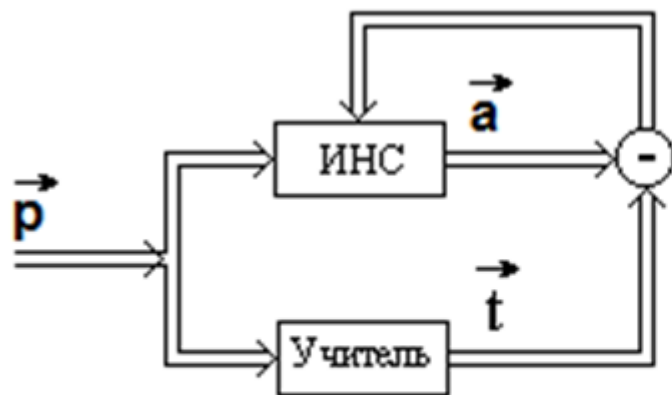
Часто используют ИНС с *латеральным торможением*.



# Виды обучения ИНС

Различают следующие виды обучения ИНС :

а) - **обучение с учителем**  
(supervised learning)



а)

Обучающие данные содержат обучающие примеры  $\mathbf{p}(q)$  и метки  $\mathbf{t}(q)$ :

$$D = \{\mathbf{p}(q), \mathbf{t}(q)\}; q = 1..Q$$

Цель обучения:

минимизировать «ошибку»

$$\mathbf{e}(q) = L(\mathbf{t}(q) - \mathbf{a}(q))$$

б) - **обучение без учителя**  
(unsupervised learning)



б)

Обучающие данные содержат только обучающие примеры  $\mathbf{p}(q)$ :

$$D = \{\mathbf{p}(q)\}; q = 1..Q$$

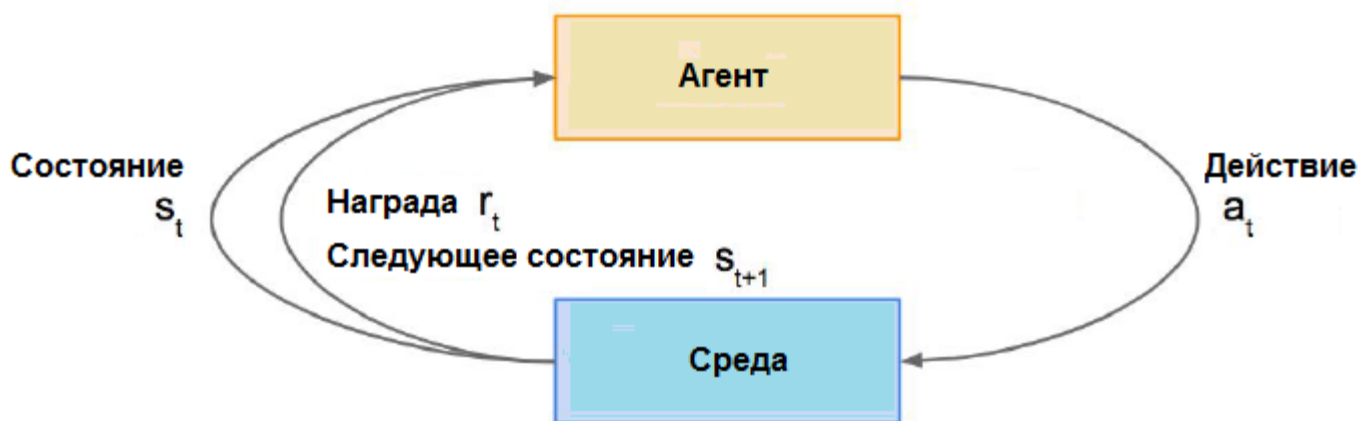
Цель обучения: в общем случае восстановить многомерное распределение вероятностей входных данных

# Виды обучения ИНС

## в) - обучение с подкреплением (RL, reinforcement learning)

Имеется агент, взаимодействующий с окружающей средой, которая в ответ на действия агента, обеспечивает его вознаграждением.

*Цель обучения:* научиться действовать, так чтобы максимизировать вознаграждение



В среде RL вы не учите агента, что и как он должен делать, вместо этого **вы даете агенту награду за каждое выполненное действие**. Таким образом, обучение превращается в процесс проб и ошибок.

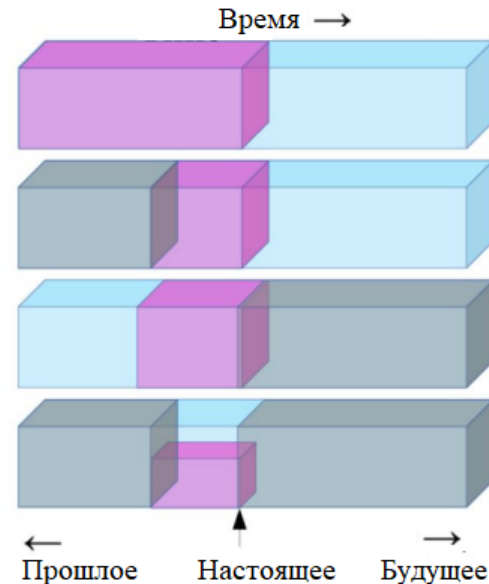
# Самоконтролируемое обучение (SSL)

**Общая идея:** полагаем, что часть данных неизвестна, обучаем нейронную сеть, таким образом, чтобы она предсказывала «неизвестную» часть данных по известной части. После SSL обучения доучиваем сеть на требуемую задачу на небольшом объеме данных.

## Self-Supervised Learning

Y. LeCun

- Предсказание любой части входа по любой другой части
- Предсказание **будущего** по **прошлому**
- Предсказание **будущего** по **недавнему прошлому**
- Предсказание **прошлого** по **настоящему**
- Предсказание **верхней** части по **нижней**
- Предсказание закрытой части по видимой
- **Представьте, что часть входных данных неизвестна, и предсказывайте её**



# “Пирог” обучения

Сколько информации используется во время обучения нейросетей?

Последующие задачи

Извлечение признаков

Обучение хорошим признакам с помощью SSL

- **“Чистое” обучение с подкреплением (вишенка)**

- Сеть предсказывает скалярное вознаграждение
- **Несколько бит на выборку**

- **Обучение с учителем (глазурь)**

- Сеть предсказывает категорию или несколько чисел для каждого входа
- Предсказание данных предоставленных человеком
- **10 → 10000 бит на выборку**

- **Самоконтролируемое обучение (бисквит)**

- Сеть предсказывает любую часть входных данных по любой наблюдаемой части
- Предсказание будущих кадров на видео
- **Миллионы бит на выборку**



Y. LeCun  
2019 Keynote at ISSCC