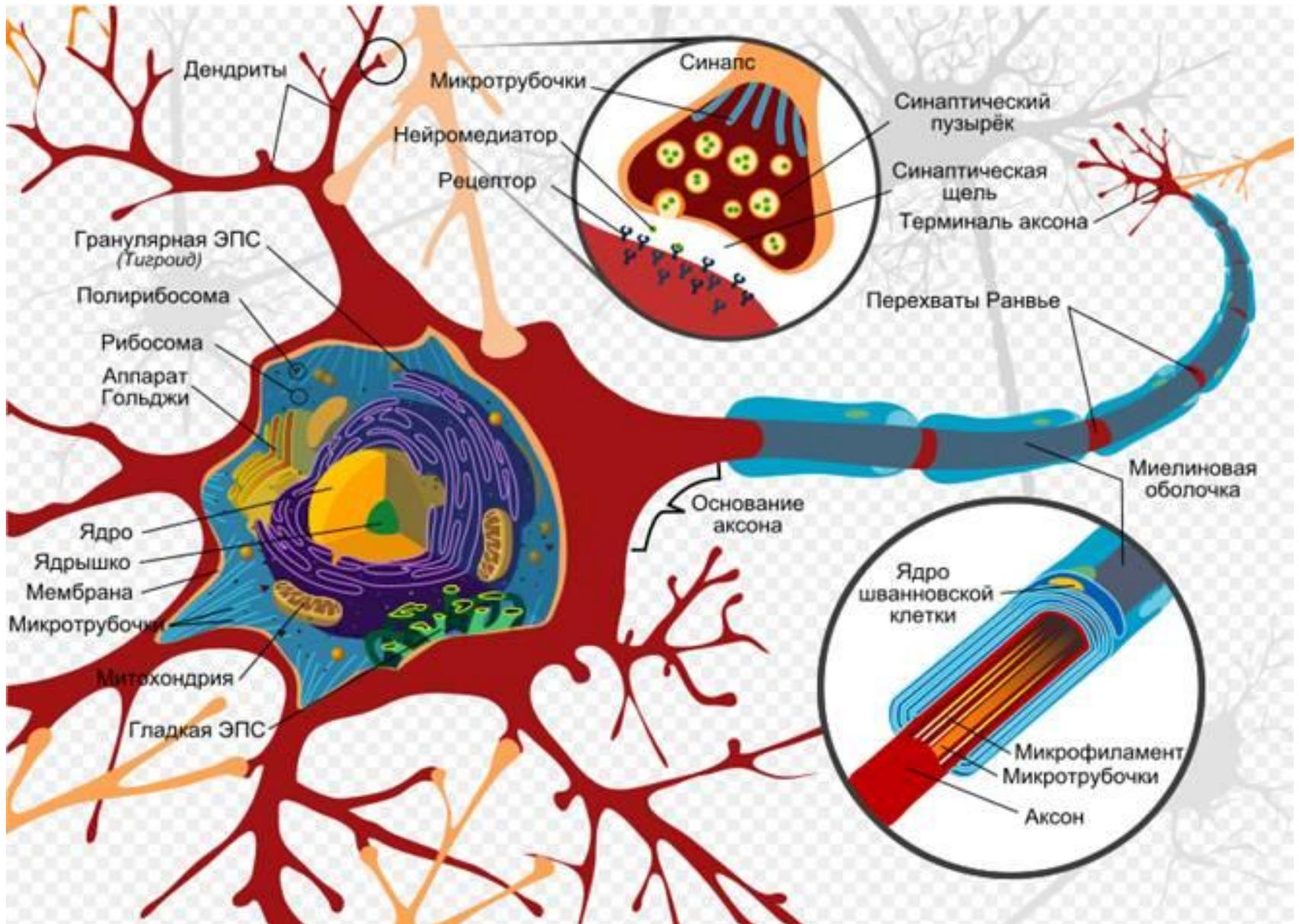


# Искусственные нейронные сети

# История

- 1943 - Warren McCulloch, Walter Pitts. A logical calculus of the ideas immanent in nervous activity. Bull. Math. Biophys. – 1943. – v.5. – pp.115–133.
- 1943 - "Поведение, Цель и Телеология" Arturo Rosenblueth, Norbert Wiener, и Julian Bigelow
- 1957 – Фрэнк Розенблатт начал работы над нейронными сетями, перцептрон (персептрон).
- 1969 – Марвин Мински и Сеймур Паперт «Перцептроны. Введение в вычислительную геометрию» – формальное доказательство ограниченности возможностей перцептрона и показывает, что он неспособен решать некоторые задачи (Проблема "четности" и "один в блоке"), связанные с инвариантностью представлений. Интерес к нейронным сетям резко спадает.
- 1974 – Пол Дж. Вербос и А. И. Галушкин одновременно изобретают алгоритм обратного распространения ошибки для обучения многослойных перцептронов. Изобретение не привлекло особого внимания.
- 1982 – после длительного упадка, интерес к нейросетям вновь возрастает. Хопфилд показал, что нейронная сеть с обратными связями может представлять собой систему, минимизирующую энергию (так называемая сеть Хопфилда). Кохоненом представлены модели сети, обучающейся без учителя (Нейронная сеть Кохонена), решающей задачи кластеризации, визуализации данных (самоорганизующаяся карта Кохонена) и другие задачи предварительного анализа данных.

# Биологический нейрон



# Нейрофизиология

- Нейрон имеет тело (с дендритами) и аксон;
- Синапсы – места контакта нейронов;
- Нейрон раздражается импульсами (1-150 м/с);

## Функционирование нейрона:

1. Поступают импульсы от других нейронов;
2. Мембрана аксона динамически изменяет проводимость – при превышении суммарного внутреннего потенциала порогового значения масштаба -50 мВ (латентный период ~ 0,25 мс);
3. Мембрана достигает *потенциала действия* около +40 мВ (~2 мс), изменяет свою полярность (деполяризуется);
4. Время прохождения импульса – синаптическая задержка (от 0,3—0,5 до нескольких мсек);
5. *Период рефрактерности* (~ 200 мс) – нейрон полностью пассивен, практически неизменно сохраняя потенциал внутри аксона на уровне около -70 мВ, на импульсы не реагирует;
6. Восстановление возбудимости до уровня несколько ниже исходного, потом повышение до исходного. Частые возбуждения увеличивают понижение порога возбудимости в конце периода рефрактерности.

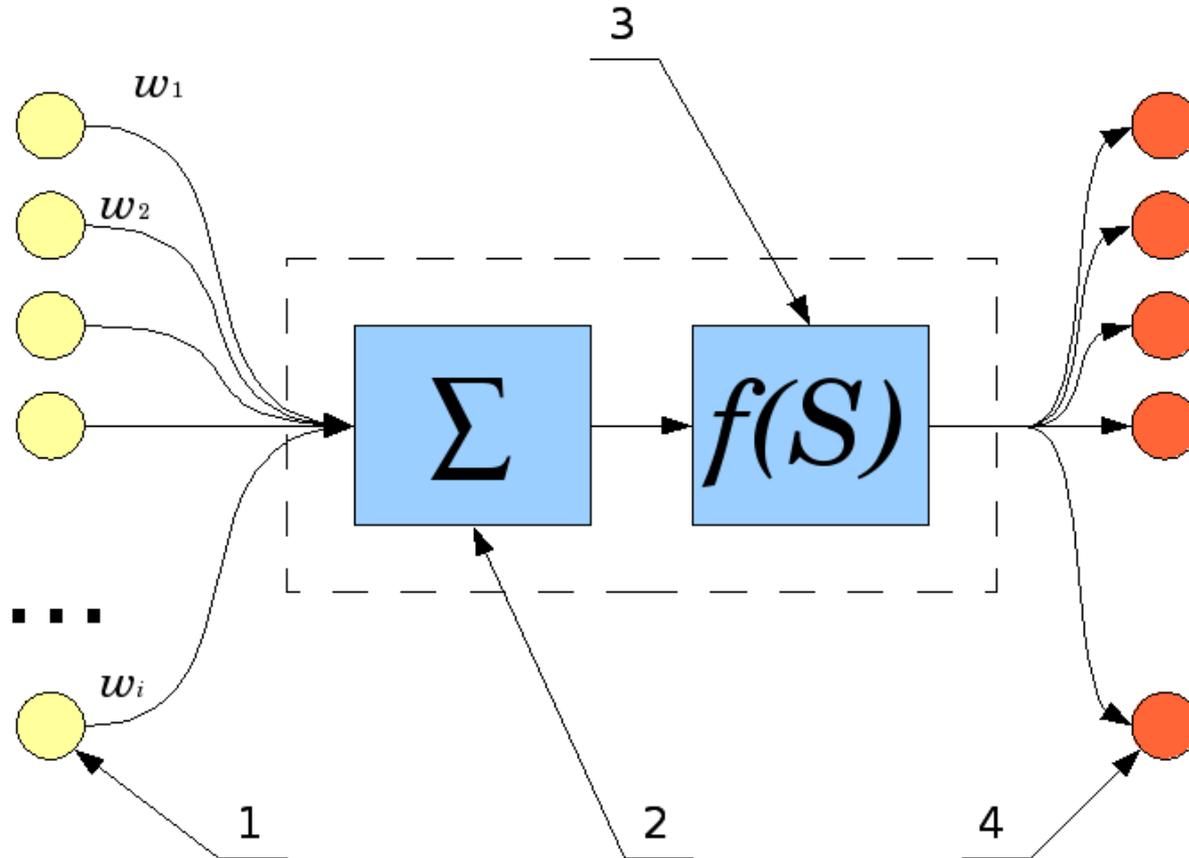
# Нейрофизиология

- Мозг – высокоциклическая сеть нейронов (10 -1000 млрд. по разным данным);
- Каждая нервная клетка связана в среднем с 1000 - 10000 других нейронов;
- Совокупность нейронов в объеме масштаба  $1 \text{ мм}^3$  формирует относительно независимую локальную сеть, несущую определенную функциональную нагрузку.

# Формальная модель нейрона (МакКаллок, Питс)

- Активность нейрона удовлетворяет принципу «Все или ничего»;
- Возбуждению нейрона должен предшествовать латентный период накопления возбуждений определенного фиксированного числа синапсов (не зависит от предыдущей активности, синапсы равнозначны);
- Единственным запаздыванием, имеющим значение, является синаптическая задержка;
- Активность тормозящего синапса исключает возбуждение нейрона;
- С течением времени структура сети не меняется.

# Формальная модель нейрона



1. Нейроны, выходные сигналы которых поступают на вход данному
2. Сумматор входных сигналов
3. Вычислитель передаточной функции
4. Нейроны, на входы которых подаётся выходной сигнал данного
5.  $w_i$  — веса входных сигналов

# Передающая (активационная) функция

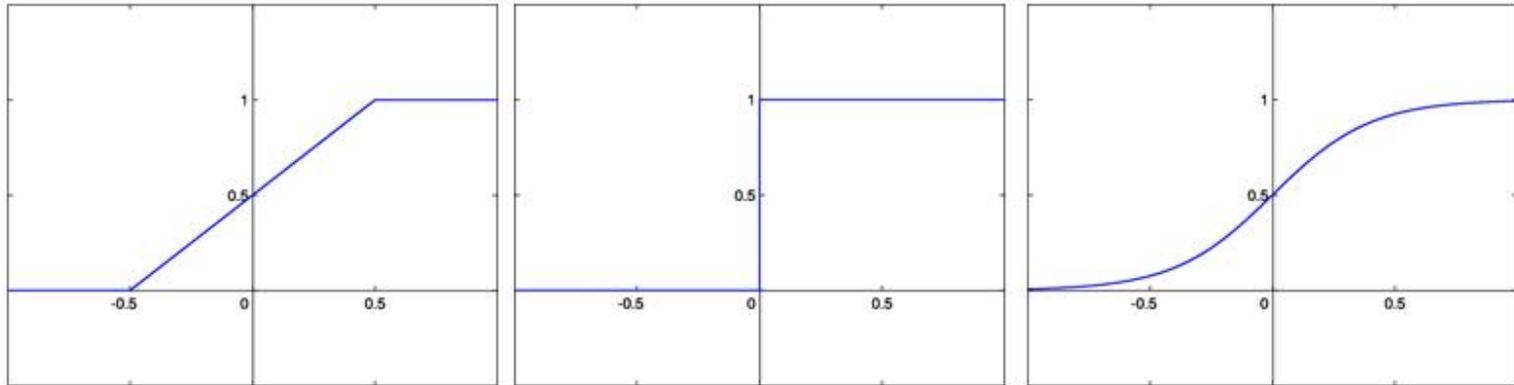
Передающая функция определяет зависимость сигнала на выходе нейрона от взвешенной суммы сигналов на его входах. В большинстве монотонно возрастающая и имеет область значений  $[-1, 1]$  или  $[0, 1]$ , однако существуют исключения.

Для некоторых алгоритмов обучения сети необходимо, чтобы она была непрерывно дифференцируемой на всей числовой оси.

Искусственный нейрон полностью характеризуется своей передающей функцией.

Использование различных передающих функций позволяет вносить нелинейность в работу нейрона и в целом нейронной сети.

# Примеры передаточных функций



Линейная, пороговая и сигмоидальная передаточные функции

$$Y(NE\text{T}) = \frac{1}{1 + \exp(-(NE\text{T} - \Theta))}$$

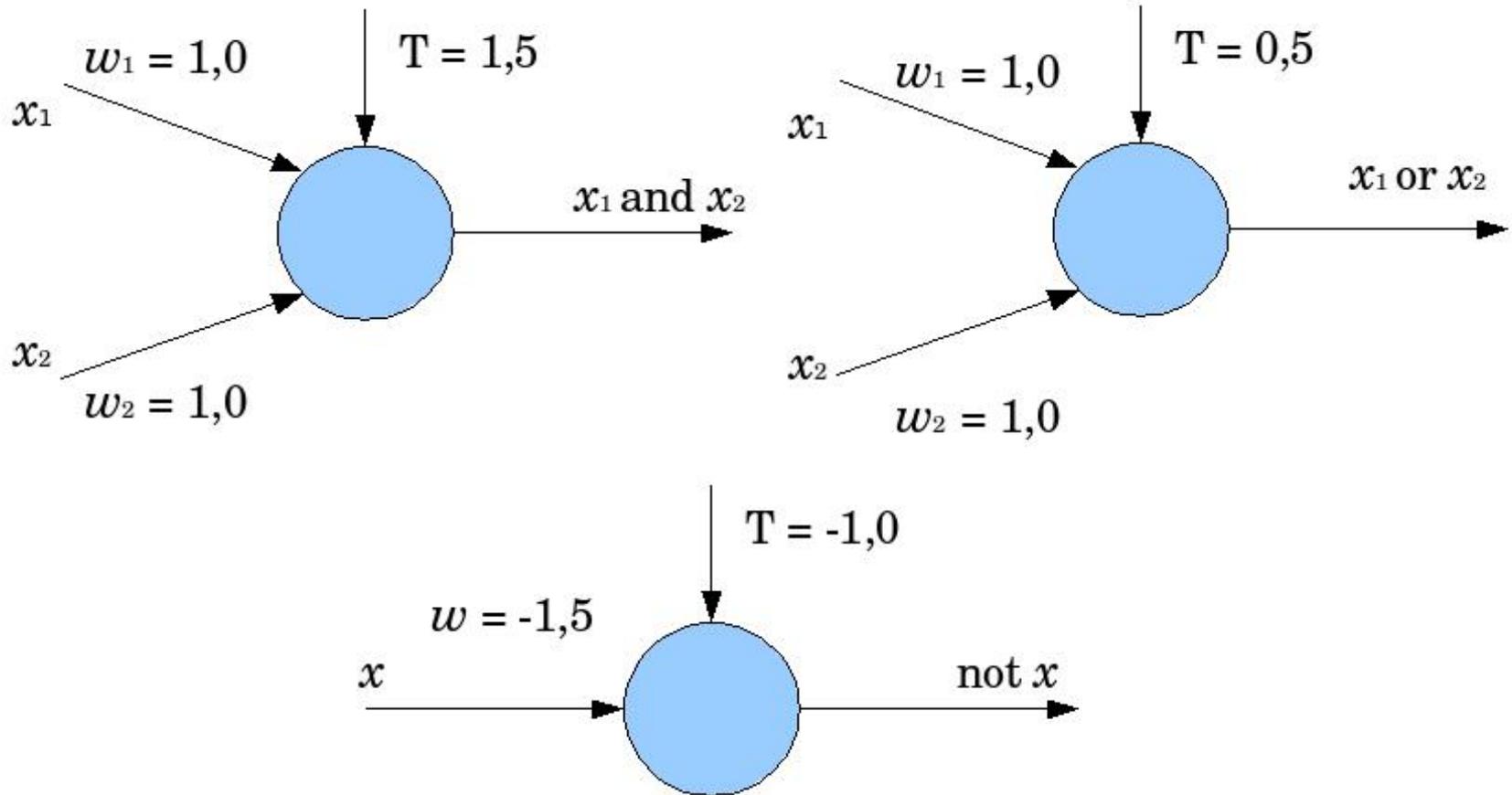
# Принцип работы искусственного нейрона

- На вход подается вектор  $A_j$  ( $j=1..N$  – количество входов);
- Вычисляем скалярное произведение  $NET$ ;
- Вычисляем значение активационной функции  $g$ , которое потом анализируется либо передается дальше:

$$NET_i = \sum_{j=0}^N w_{j,i} * a_j$$

$$a_i = g(NET_i) = g\left(\sum_{j=0}^N w_{j,i} * a_j\right)$$

# Моделирование логических функций



# Нейронные сети - применение

Имеется множество *объектов* (ситуаций), разделённых некоторым образом на *классы*. Задано конечное множество объектов, для которых известно, к каким классам они относятся. Это множество называется *выборкой*. Классовая принадлежность остальных объектов не известна. Требуется построить алгоритм, способный классифицировать произвольный объект из исходного множества.

Отделимость классов, линейная отделимость:

$$\exists \{g_i(x)\} : x^* \in X_i \Leftrightarrow g_i(x^*) = \max_k g_k(x^*)$$

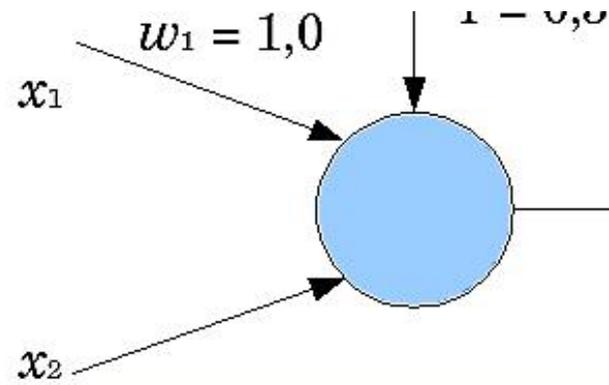
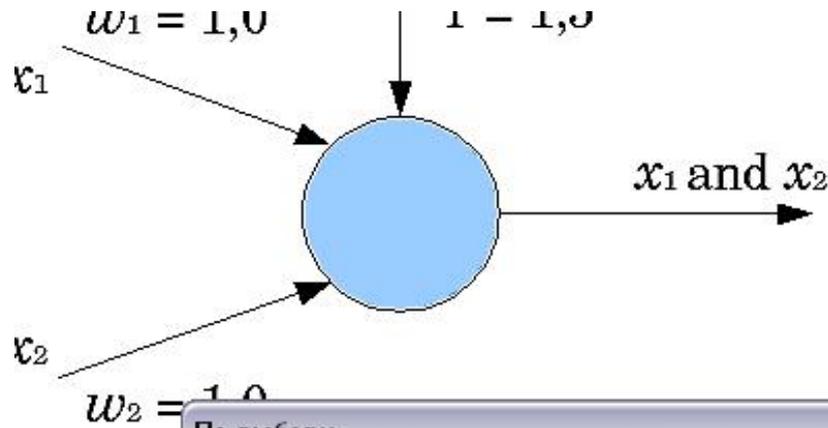
# Хранение данных

Есть черно-белое изображение, размер  $256 \times 256$  точек. На сенсорный слой подаются координаты точки, выход – цвет.

# Обработка сигналов

Есть некоторый сигнал, необходимо получить выходной. Зависимость от конфигурации сенсорного слоя. Как в общем случае сигнал свести с заданной точностью к вектору значений?

# Обработка изображений



По выбору..

$x_1$  and  $x_2$

$w_2$

$T = -1,0$

		-1		
	-1	5	-1	
		-1		

Масштаб: 1    Смещение:

ОК

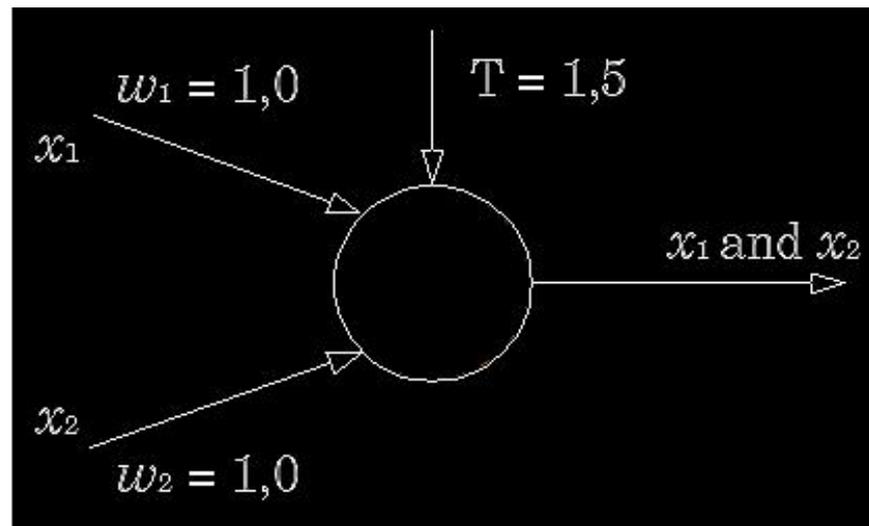
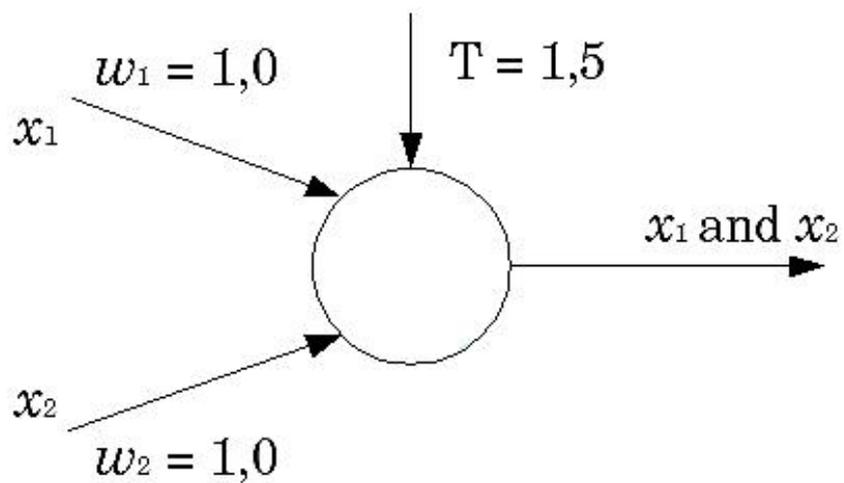
Отмена

Загрузить...

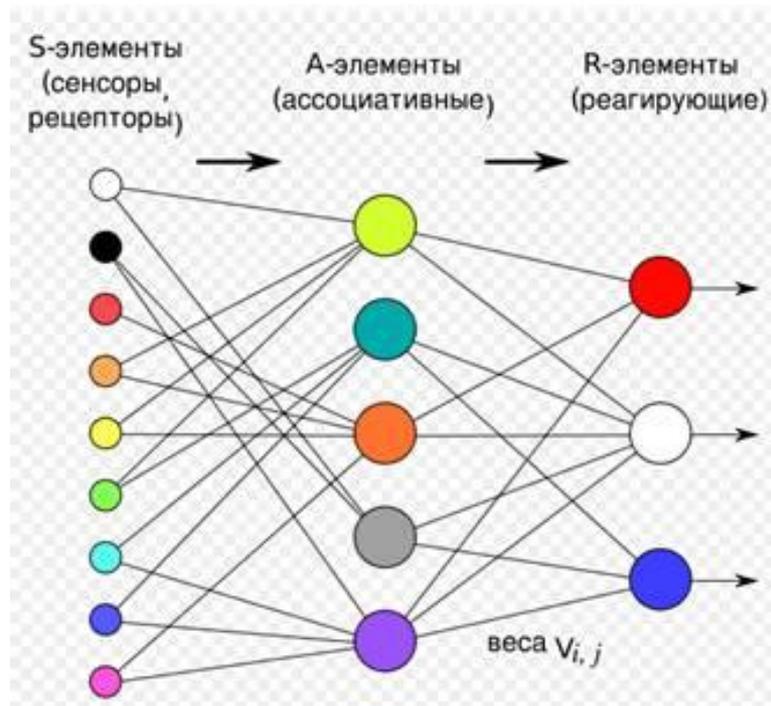
Сохранить...

Предварительный пр...

# Примеры



# Персептрон Розенблатта



- На входе и на выходе – векторы, по вектору R решаем, что получили;
- Предназначался для распознавания образов (два класса в простейшем случае, один R-элемент);
- Функции – пороговые, значение R принадлежит  $[-1,1]$ ;
- «Память» сети – это веса (в общем случае, матрица весов), и конфигурация сети.

# Классификация нейронов

- *Входные нейроны* — принимают исходный вектор, кодирующий входной сигнал. Как правило, эти нейроны не выполняют вычислительных операций, а просто передают полученный входной сигнал на выход, возможно, усилив или ослабив его;
- *Выходные нейроны* — представляют из себя выходы сети. В выходных нейронах могут производиться какие-либо вычислительные операции;
- *Промежуточные нейроны* — выполняют основные вычислительные операции.

# Простые элементы

- S-элемент: пороговый (обычно сенсор), входной сигнал из внешней среды, 1 – если больше порога, иначе 0;
- A-элемент – определения аналогичны;
- R-элемент – аналогично функции sign;
- Передаточные функции связаны с весами, учитывается время передачи сигнала между нейронами:

$$c_{ij}^*(t) = f[v_{ij}(t), u_i^*(t - \tau_{ij})]$$

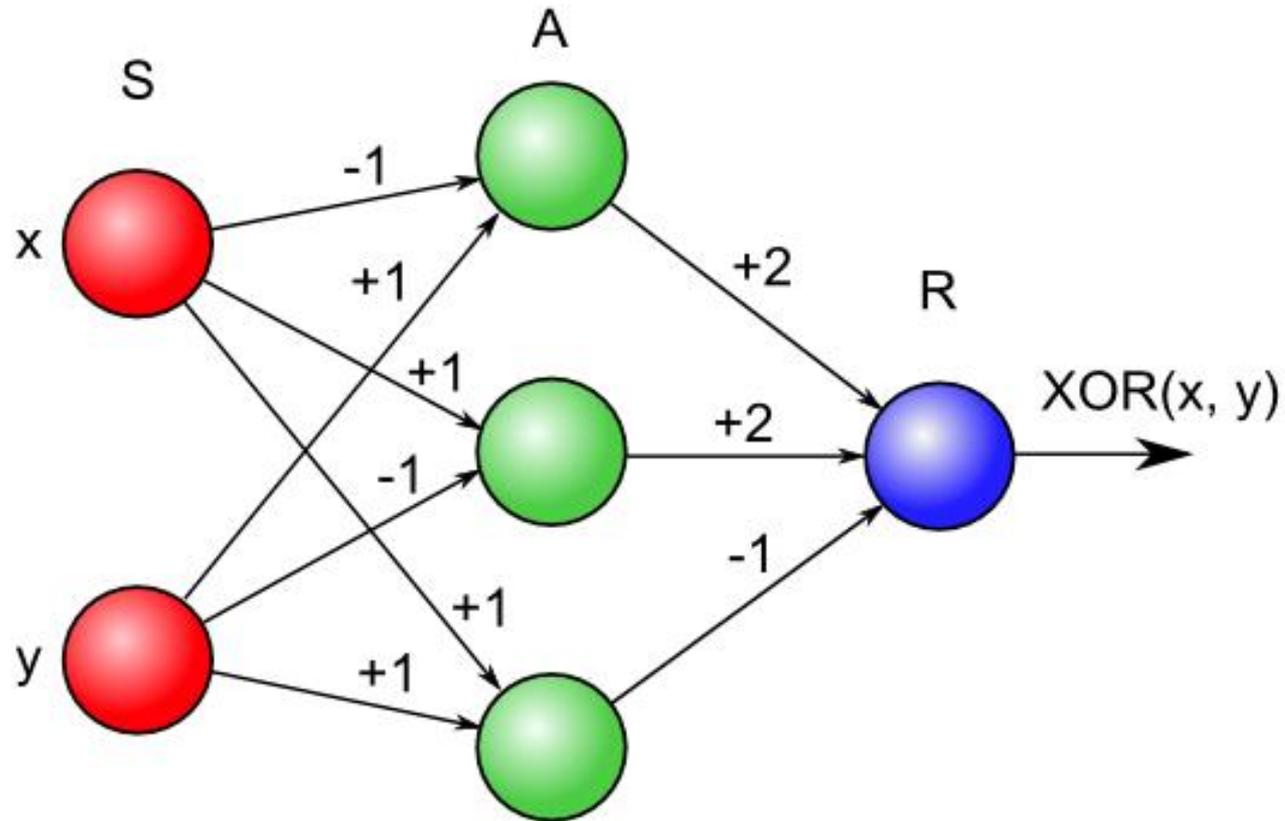
# Простой персептрон

1. Имеется только один R-элемент, связанный со всеми A-элементами;
2. Связи последовательные, от S к A, от A к R;
3. Веса всех связей S-A являются фиксированными;
4. Время передачи либо 0, либо константа;
5. Все передающие функции зависят от алгебраической суммы входных сигналов, поступивших на вход нейрона.

Элементарным называется простой персептрон с простыми R- и A-элементами, передающая функция которого имеет вид

$$c_{ij}^*(t) = v_{ij}(t)u_i^*(t - \tau)$$

# Решение задачи XOR



# Алгоритмы обучения

**Алгоритм коррекции ошибки**, предложенный Розенблаттом для простейшего перспетрона. Конфигурация: сенсоры регистрируют только сигналы  $\{0, 1\}$  (соответствуют выходам слоя S), дуги между S и A слоями не имеют весов (только передают сигналы), изменяется только матрица весов между слоями S и R, слой R – один элемент, функции пороговые, на выходе 0 – относим образ к первому классу, 1 – относим образ ко второму классу.

Правила корректировки весов (Хебб, 1949) для простейшего персептрона:

- 1. Если сигнал персептрона неверен и равен нулю, то необходимо увеличить веса тех входов, на которые была подана единица.*
- 2. Если сигнал персептрона неверен и равен единице, то необходимо уменьшить веса тех входов, на которые была подана единица.*

# Дельта-правило

*$j$ -й вес для  $i$ -го нейрона меняется по правилу:*

$$w_j(t+1) = \eta e_i x_j$$

Обобщенное правило:

$$\Delta w_{ij} = \eta (d_i - y_i) f'(S_i) x_j$$

# Теорема о сходимости персептрона

Даны элементарный перцептрон, пространство стимулов  $W$  и некоторая классификация  $C(W)$ , для которой известно, что решение существует. Предположим, что все стимулы из  $W$  появляются в любой последовательности, но при условии, что каждый стимул появляется повторно через некоторый конечный интервал времени. Тогда процесс обучения с коррекцией ошибок (с квантованием или без квантования подкрепления), начинающийся с произвольного исходного состояния, всегда приведет к достижению решения для  $C(W)$  в течении конечного промежутка времени. При этом все входные сигналы к  $R$  - элементам достигнут значения, по крайней мере равного некоторой произвольной величине  $d \geq 0$ .

# Алгоритм обратного распространения ошибки

- Описан в 1974 г. А. И. Галушкиным, а также независимо и одновременно Полем Дж. Вербосом. Далее существенно развит в 1986 г. Дэвидом И. Румельхартом, Дж. Е. Хинтоном и Рональдом Дж. Вильямсом и независимо и одновременно С. И. Барцевым и В. А. Охониным.
- Итеративный градиентный алгоритм, который используется с целью минимизации ошибки работы многослойного перцептрона и получения желаемого выхода.

# Схема метода

$$\delta_j = y_j(1 - y_j)(d_j - y_j)$$

$$\delta_j = y_j(1 - y_j) \sum_{k \in OUT_j} \delta_k w_{jk}$$

$$\Delta w_{ij} = \eta \delta_j x_i$$

# Сравнение алгоритмов обучения

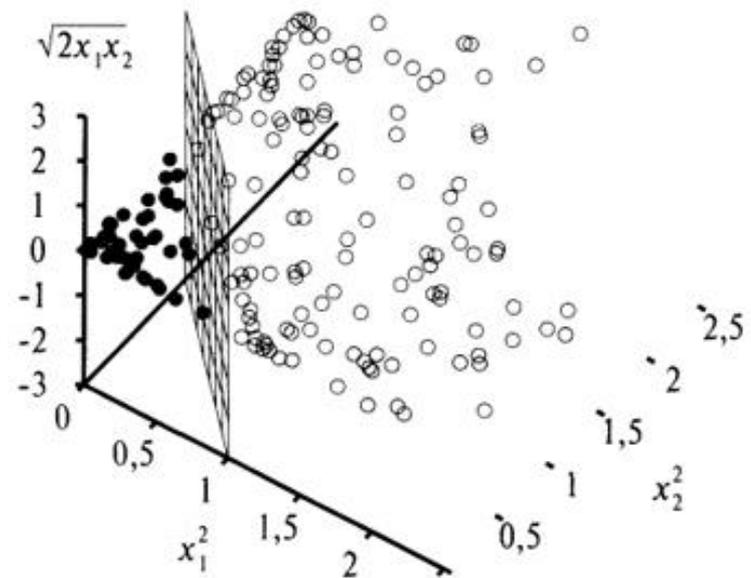
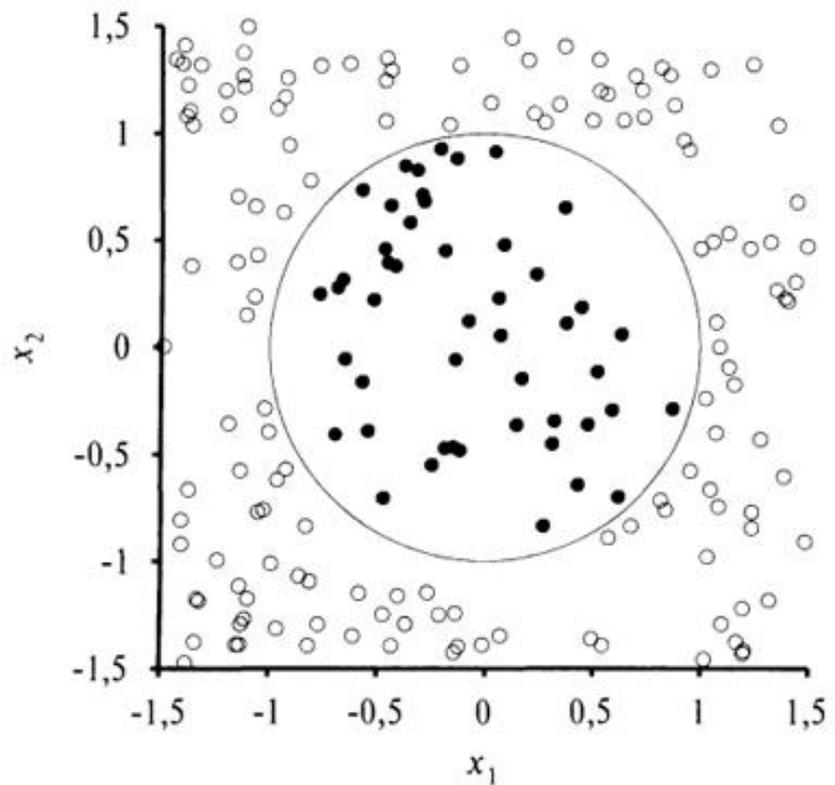
Алгоритм	Время, с	Количество циклов	Количество операций, $\times 10^6$
Наискорейшего спуска с адаптируемым шагом	57,71	980	2,5
Сопряженных градиентов	19,16	89	0,75
Переменной метрики BFGS	10,86	44	1,02
Левенберга-Марквардта	1,87	6	0,46
RPROP	12,96	185	0,56

Mathlab, пакет “Neural Networks”, ПК, персептрон 1-10-1, 41 пара обучающих данных. По данным из [1].

# Алгоритмы синтеза сетей

- **Алгоритм оптимального повреждения мозга** – удаляет из полной сети несущественные связи;
- **Алгоритм заполнения мозаики** – последовательно выполняется обучение с добавлением новых пар стимул – реакция. Если структура сети недостаточна, то усложняем.

# Предобработка и SVM



# Литература

1. Осовский С. Нейронные сети для обработки информации. М.: Финансы и статистика, 2002. – 344 с;
2. [Philipp Wagner. Machine Learning with OpenCV2;](#)
3. М. Мински, С. Пайперт. Персептроны. М.: Мир, 1971. – 263 с;
4. Ф. Розенблатт. Принципы нейродинамики. М.: Мир, 1965.