

Искусственные нейронные сети

Высшая школа экономики, Нижегородский
филиал

Кафедра прикладной математики и
информатики

ст.преп. Бабкина Татьяна Сергеевна

tbabkina@hse.ru

taty-bab@yandex.ru

2016

Литература

- Хайкин С. Нейронные сети. Полный курс, 2-е изд. – М.: «И.Д. Вильямс», 2016.
- Галушкин А.И. Нейронные сети: основы теории. – М.: Горячая линия – Телеком, 2012.
- Тархов Д.А. Нейросетевые модели и алгоритмы. Справочник – Радиотехника, 2014.
- Рутковская Д., Пилиньский М., Рутковский Л. Нейронные сети, генетические алгоритмы и нечеткие системы. – М.: Горячая линия – Телеком, 2013.
- Борисов В.В., Круглов В.В., Федулов А.С. Нечеткие модели и сети. – М.: Горячая линия – Телеком, 2012.
- Коньшева Л.К., Назаров Д.М. Основы теории нечетких множеств. – Питер, 2011.
- Круглов В.В., Борисов В.В. Искусственные нейронные сети. Теория и практика. – М.: Горячая линия – Телеком, 2001– 382 с.
- Круглов В.В., Дли М.И., Голунов Р.Ю. Нечеткая логика и искусственные нейронные сети: Учеб. Пособие. – М.: Издательство Физико-математической литературы, 2001– 224 с.
- George F. Luger Artificial Intelligence. Fifth Edition. – Addison Wesley, 2005. – 903 p. (353 – 357, 453 – 506 pp.)

**Искусственные нейронные сети –
вычислительные структуры, которые
моделируют процессы человеческого мозга**

Способность к обучению

Искусственный нейрон

Немного истории

- **1904** С.Рамон-и-Кахаль

Открытие нейронов – нервных клеток

- **1943** У.Мак-Каллок, У.Питтс

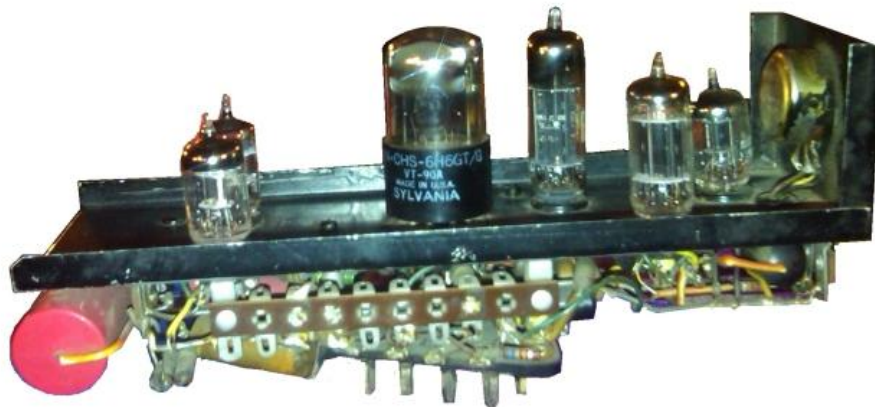
Модель искусственного нейрона; основные принципы построения ИНС

- **1949** Д.Хебб

Идеи о соединении и взаимодействии нейронов мозга; правила обучения ИНС

- **1951** М.Минский, Д.Эдмондс

SNARC: крыса в лабиринте



Немного истории

- **1957** В.И.Арнольд, А.Н.Колмогоров

Решение 13 проблемы Гильберта: любую непрерывную функцию многих переменных можно представить как суперпозицию непрерывных функций одной переменной и сложения

- **1962** Ф.Розенблатт **Энтузиазм**

Персептрон; попытки предсказания погоды, анализа электрокардиограмм, распознавания рукописных букв

- **1969** М.Минский, С.Пейперт **Пессимизм**

Критическая работа «Персептроны»; возможности персептронов ограничены

- **1970е** Т.Кохонен, С.Гроссберг и др.

Многослойные сети различной архитектуры; успешное решение задач, считавшихся неразрешимыми

- **1982** Дж.Хопфилд

Динамическая сеть

Немного истории

- **1986** Д.Румельхарт, Дж.Хинтон, Р.Вильямс

Метод обратного распространения ошибки

- **Середина 1980х** *Нейросетевой бум*

Новые модели нейронных сетей

- **1987** Р. Хехт-Нильсен

Вариант 13 проблемы Гильберта: любую непрерывную функцию многих переменных можно с любой точностью приблизить с помощью трехслойного персептрона

- **1990е**

Развитие новых нейропарадигм замедлилось; нейрочипы, нейрокомпьютеры

Задачи, решаемые с помощью ИНС

- **Классификация образов**

Определить принадлежность входного образа одному из заранее заданных классов

Распознавание речи и изображений, мед.диагностика

- **Кластеризация (классификация без учителя)**

Разместить близкие образы в один кластер (кластеры заранее не определены)

Сжатие данных, извлечение знаний

- **Аппроксимация функций**

Найти оценку неизвестной функции $F(x)$ по обучающей выборке $\{(x_1, y_1) \dots (x_N, y_N)\}$

Инженерное и научное моделирование

- **Прогноз**

Найти неизвестное $y(t_{N+1})$ по известным $y(t_1) \dots y(t_N)$

Принятие решений в науке, технике, бизнесе

Задачи, решаемые с помощью ИНС

- **Оптимизация**

Найти решение x , удовлетворяющее системе ограничений и максимизирующее (минимизирующее) целевую функцию

Принятие решений в науке, технике, бизнесе

- **Ассоциативная память**

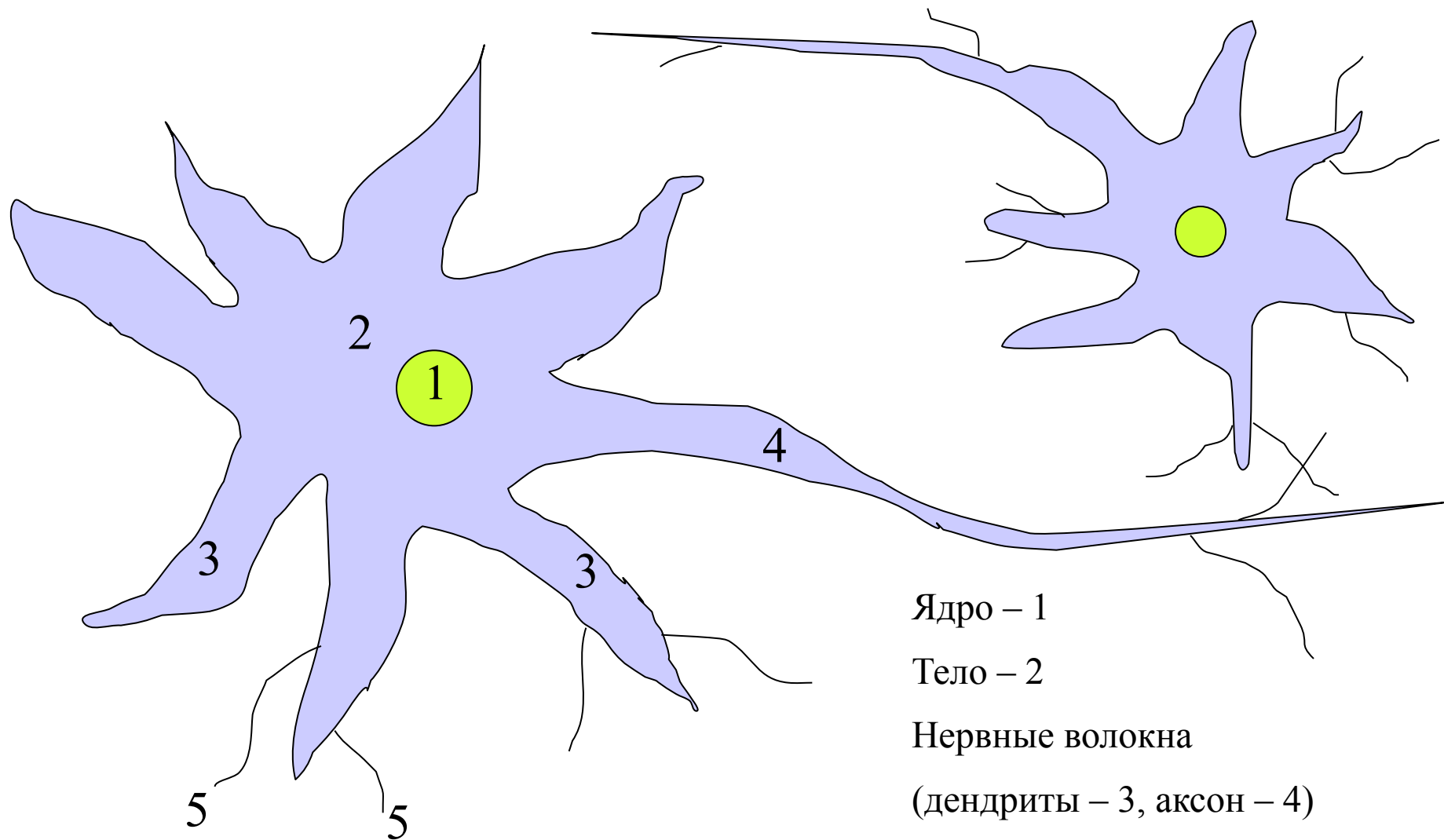
Организовать память с доступом по частичному или искаженному содержанию

Информационно-вычислительные системы

- **Управление**

Рассчитать управляющее воздействие и на динамическую систему $\{u(t), y(t)\}$, чтобы выход y совпадал с эталоном

Биологический нейрон



Ядро – 1

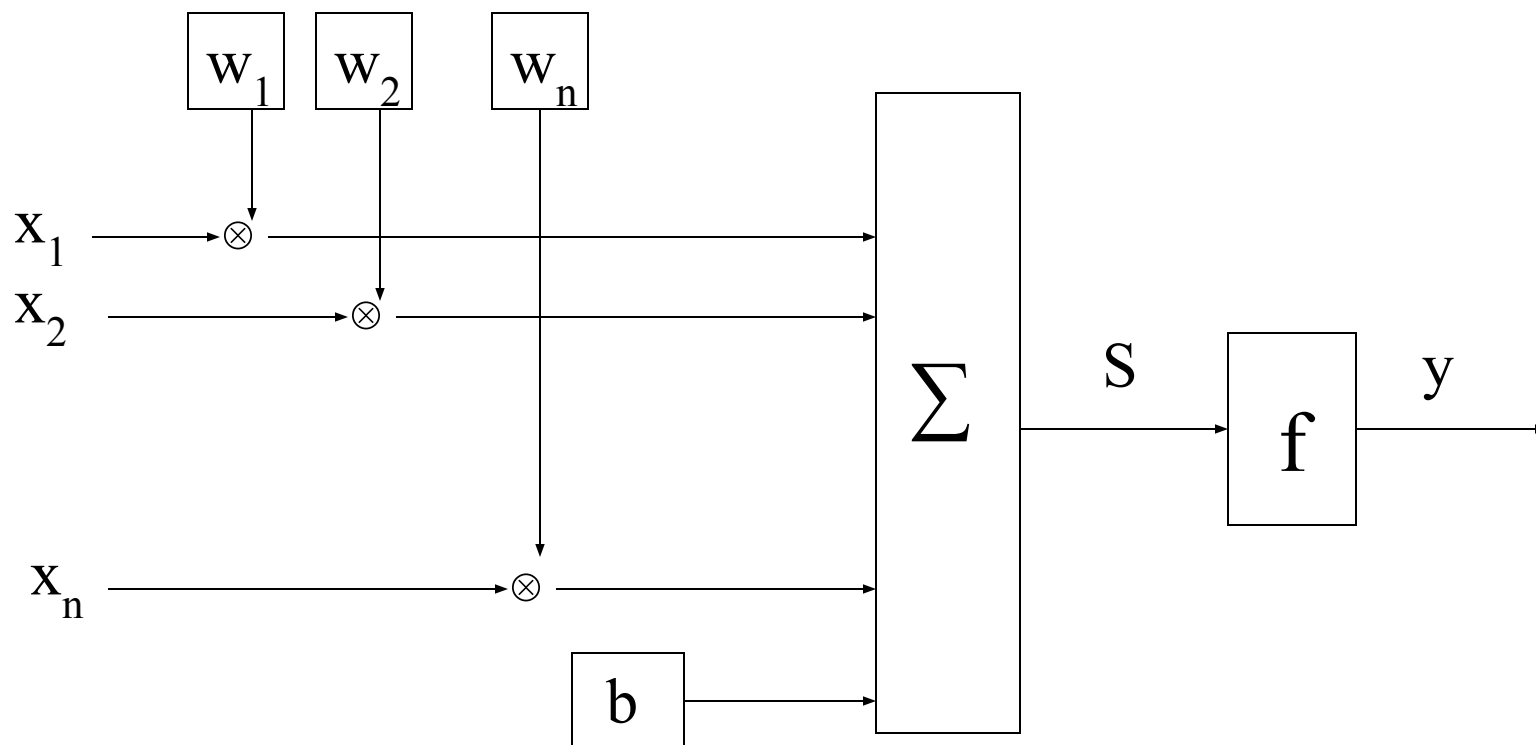
Тело – 2

Нервные волокна

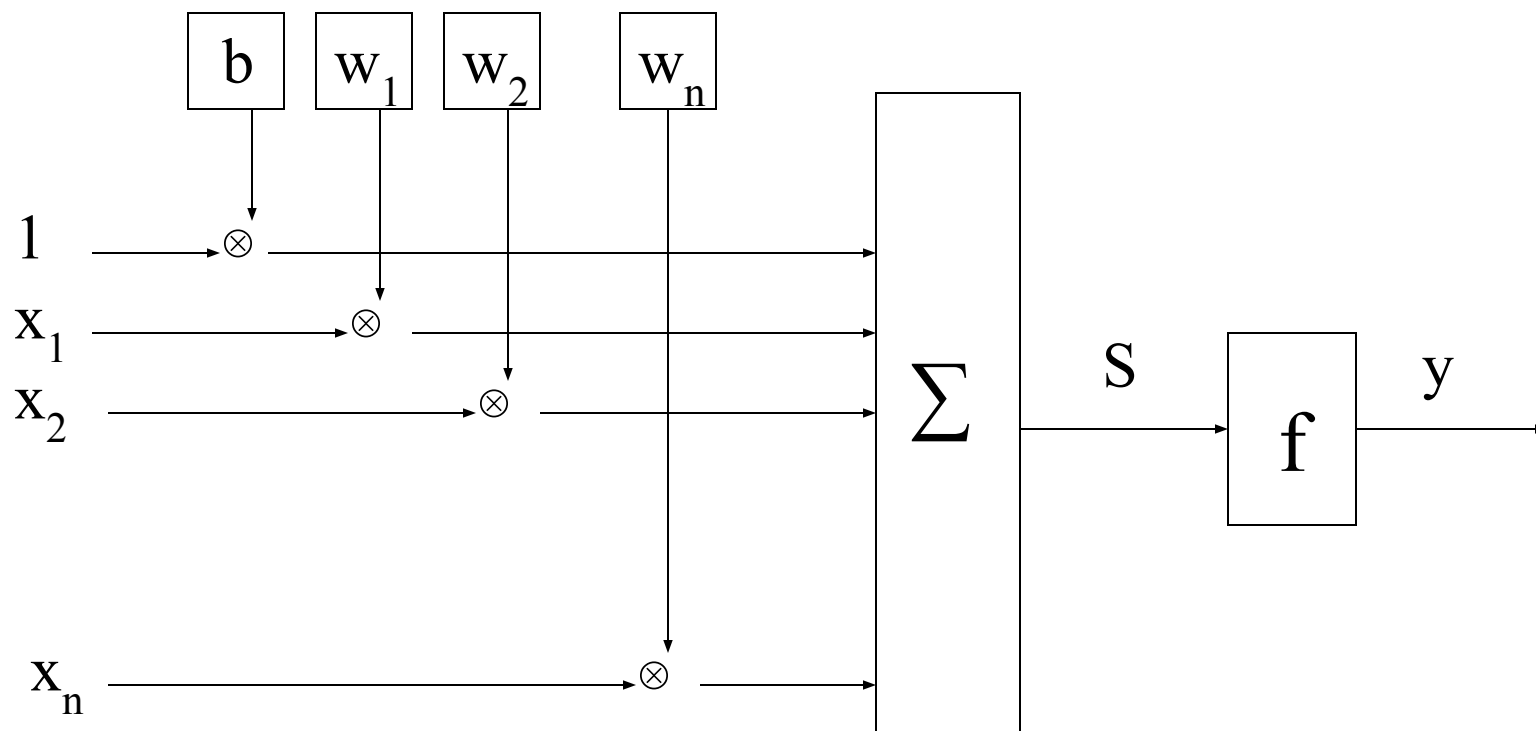
(дендриты – 3, аксон – 4)

Синаптические окончания – 5

Искусственный нейрон



Искусственный нейрон



Математическая модель искусственного нейрона

$$S = \sum w_i x_i + b$$

$$y = f(S)$$

$x_i, i=1,2,...,n$ – сигнал от другого нейрона

$w_i, i=1,2,...,n$ – сила синаптической связи

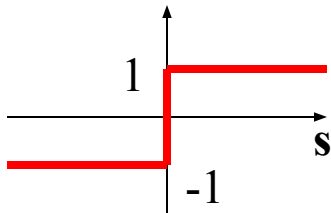
b – смещение

f – функция активации нейрона

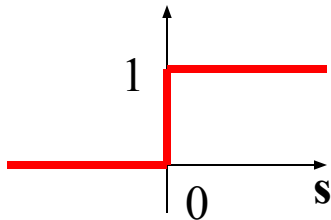
y – выходной сигнал нейрона

Функции активации

Пороговые



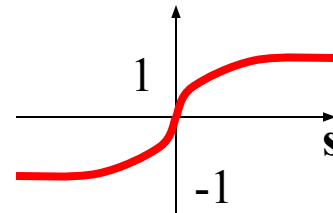
$$\text{sign}(s) = \begin{cases} 1, & s > 0 \\ -1, & s < 0 \end{cases}$$



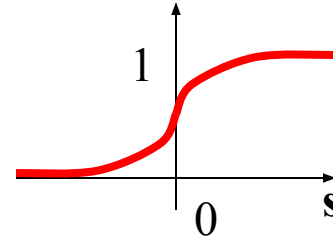
Функция Хэвисайда

$$\theta(s) = \begin{cases} 1, & s > 0 \\ 0, & s < 0 \end{cases}$$

Сигмоидные



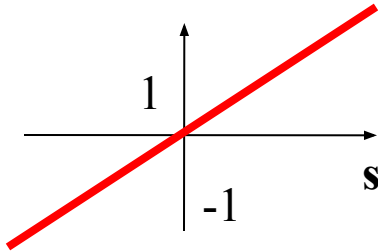
$$\text{th}(s) = \frac{e^{as} - e^{-as}}{e^{as} + e^{-as}}$$



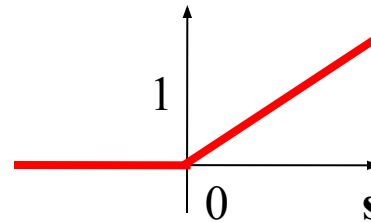
$$P(s) = \frac{1}{1 + e^{-as}}$$

Функции активации

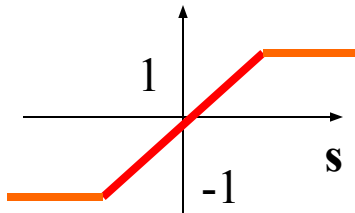
Линейная



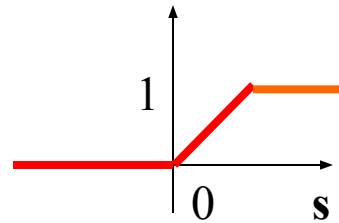
Полулинейная



Линейная с насыщением

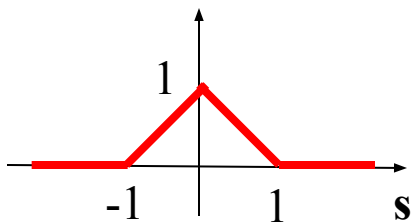


Полулинейная с насыщением

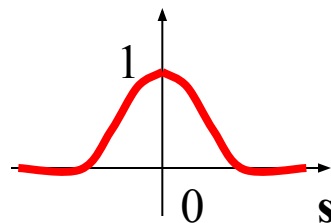


Функции активации

Треугольная



Радиальная базисная
(гауссова)



$$\varphi(s) = e^{-s^2}$$

Искусственная нейронная сеть

ИНС – набор искусственных нейронов,
соединенных между собой

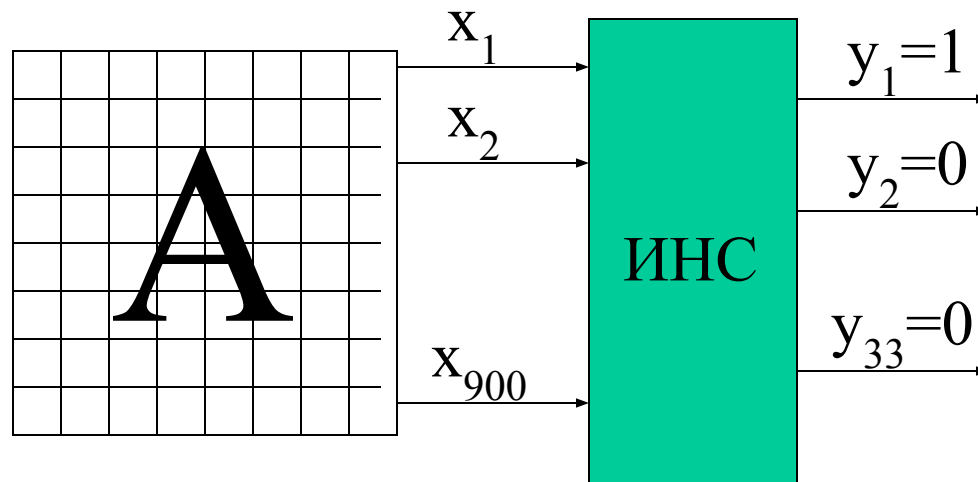
Функции активации фиксированы

Веса и смещения могут изменяться (свободные
параметры)

Входные и выходные нейроны



Пример



30×30 пикселей

Построение ИНС

- **Выбор архитектуры**

Топология

Функции активации

Входы-выходы

*Для каждой задачи – своя
архитектура*

- **Обучение**

«С учителем» и «без учителя»

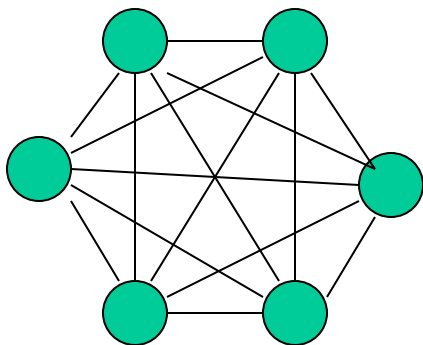
Настройка весов и смещений
(10000)

*Для каждой архитектуры – свои
алгоритмы*

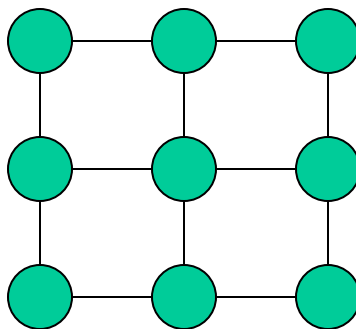
Наиболее изученные:
персептрон,
сеть Кохонена,
обобщенно-
регрессионная сеть
и др.

Классификация ИНС

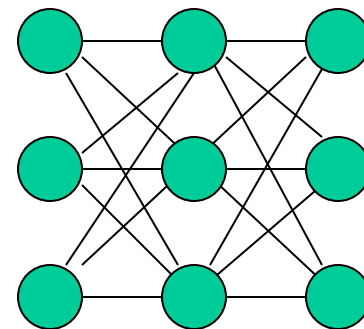
По топологии



Полносвязные



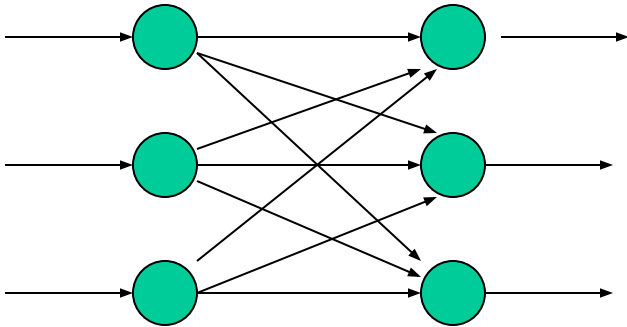
Слабосвязные



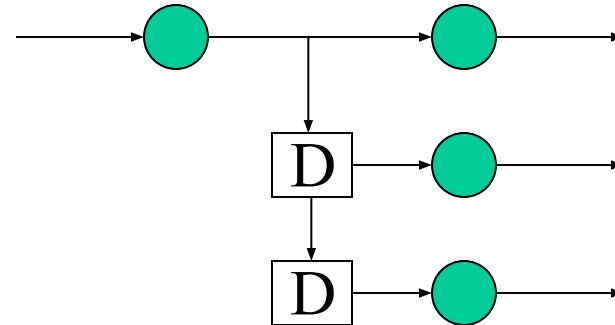
Многослойные

Классификация ИНС

По наличию задержек



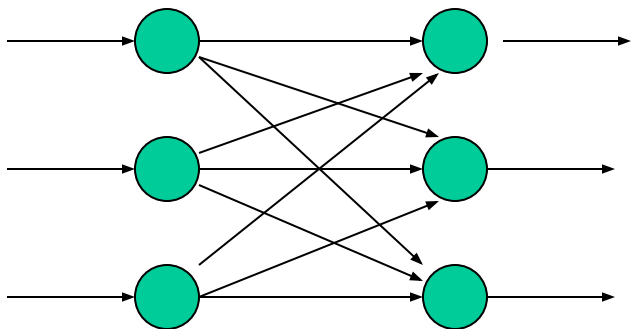
Статическая сеть



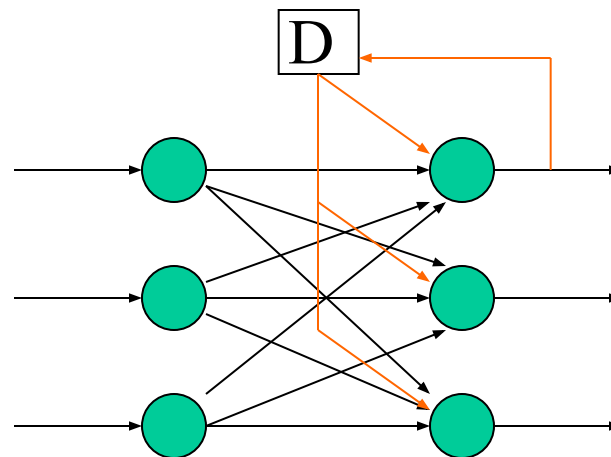
Динамическая сеть

Классификация ИНС

По наличию обратной связи



Сеть с прямым
распространением сигнала



Рекуррентная сеть

Классификация ИНС

По функциям активации

- Однородные
- Неоднородные

По типу входного сигнала

- Бинарный
- Целый
- Действительный

- Увеличение числа слоев и числа нейронов повышает возможности сети, но замедляет обучение
- Введение обратных связей повышает возможности сети, но возникает вопрос об устойчивости

Обучение ИНС

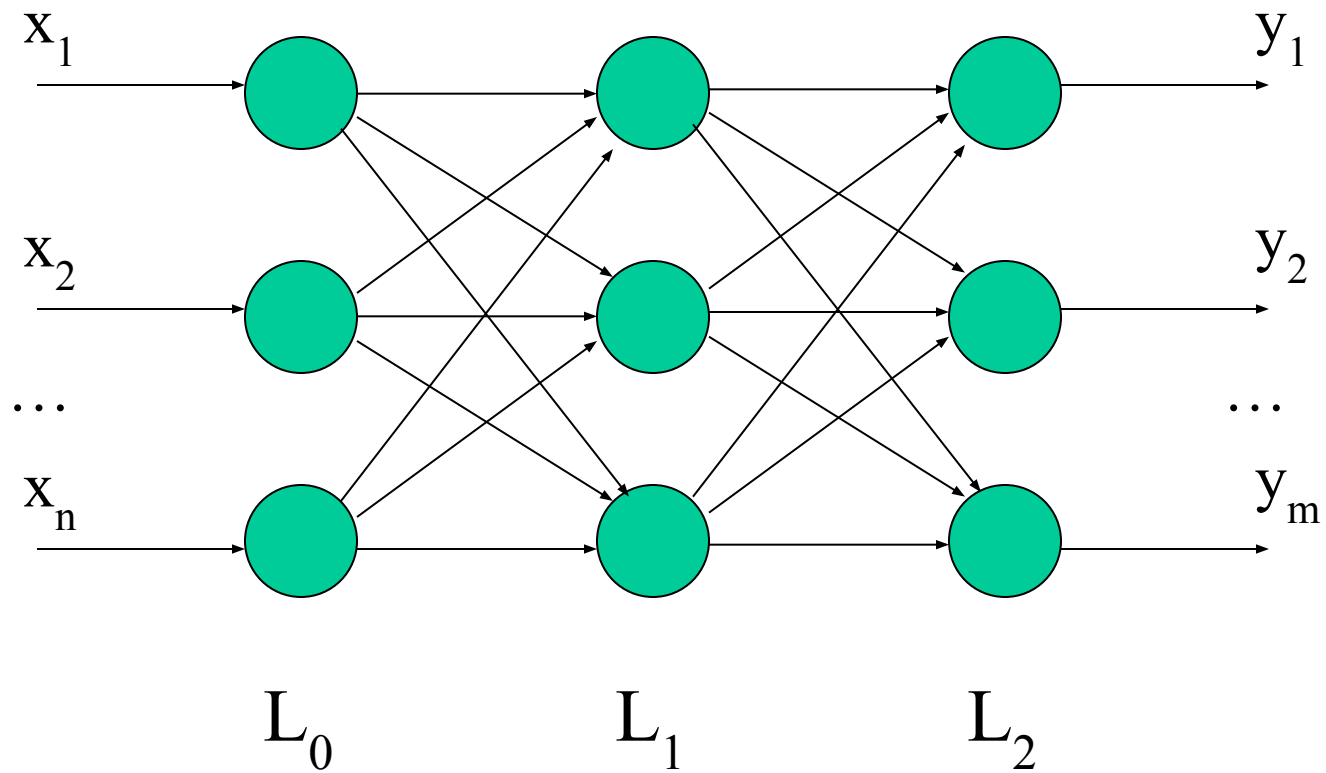
- Нейронная сеть – адаптивная система

Способность сети обрабатывать информацию развивается в результате обучения на примерах

- Нейронная сеть – «черный ящик»

Знания накапливаются в виде весовых коэффициентов и не поддаются содержательной интерпретации

Многослойная ИНС



Теоретические результаты

- **Теорема Хехт-Нильсена**

m -мерная функция от n переменных с заданной точностью может быть приближена функцией, вычисляемой двуслойной ИНС с прямыми связями

- **Теорема о полноте (следствие)**

Любая непрерывная функция на замкнутом ограниченном множестве может быть равномерно приближена функциями, вычисляемыми ИНС, функции активации которых непр. и дважды непр. дифференцируемы

Число нейронов в скрытом слое

- **Многослойная сеть с сигмоидными функциями активации**

$$mN/(1+\log_2 N) \leq L_w \leq m(N/m+1)(n+m+1)+m$$

- **Двуслойная сеть**

$$L=L_w/(n+m)$$

- **Другая формула для двуслойной сети**

$$N/10-n-m \leq L \leq N/2-n-m$$

Отсутствует строгая теория, касающаяся

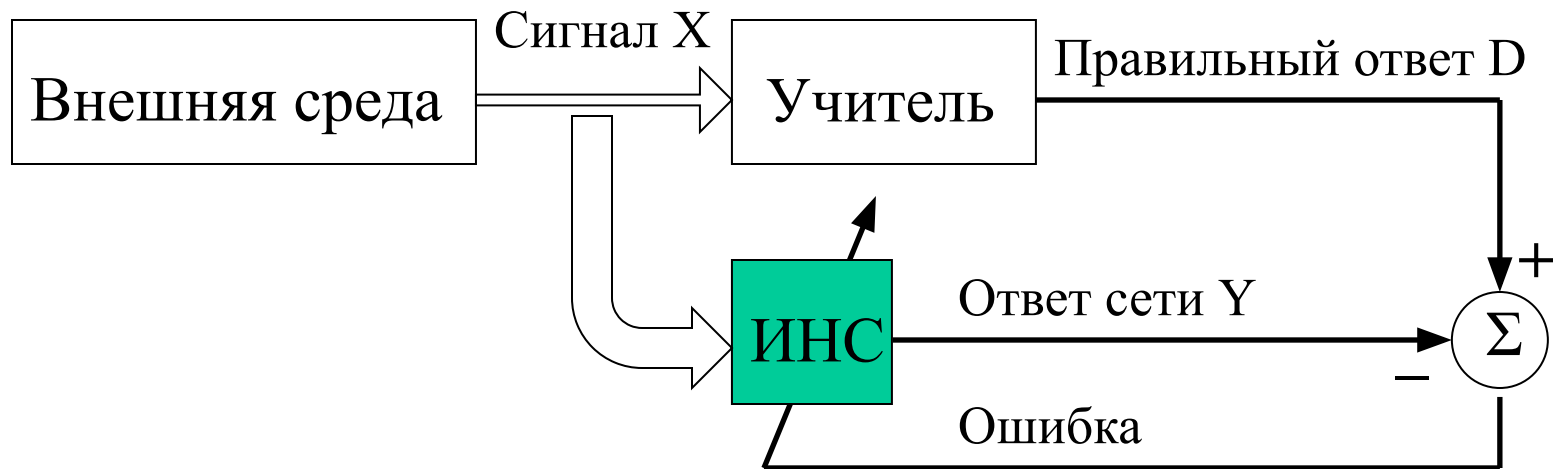
- Многослойных сетей (число слоев больше 2)
- Большинство рекуррентных сетей
- Сетей со связями от слоя l к слою $l+k$

Технологии обучения

- «С учителем»

Обучающее множество – множество пар (сигнал, правильный ответ)

$$Q = \{(X^1, D^1) \dots (X^N, D^N)\}$$



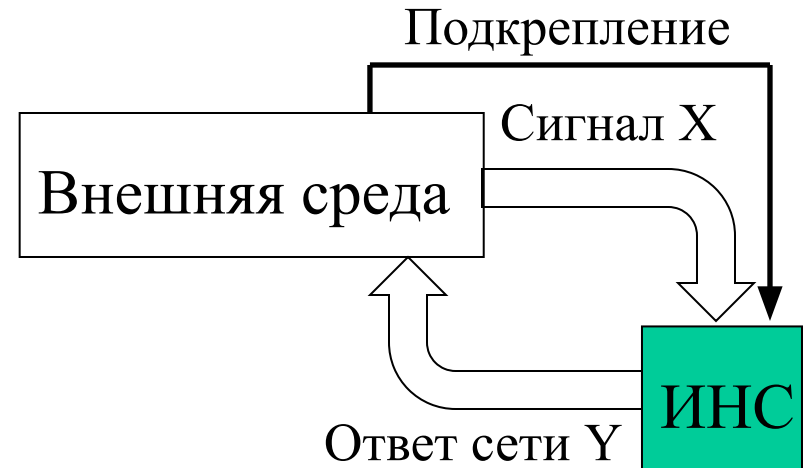
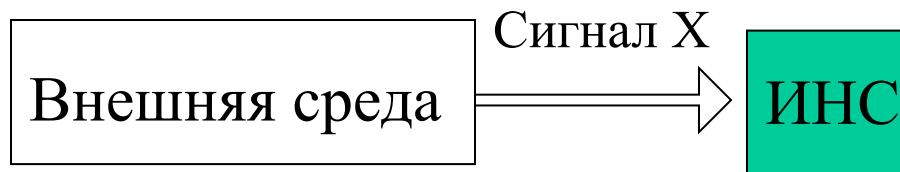
Технологии обучения

- «Без учителя»

Обучающее множество – множество примеров;
правильные ответы неизвестны

$$Q = \{X^1 \dots X^N\}$$

1. Самоорганизующееся обучение
2. Обучение с подкреплением



Обучение «с учителем»

- **Сеть обучена**, если она дает правильные ответы на примеры из обучающего множества
- **Сеть способна к обобщению**, если она дает правильные ответы на примеры из всей области определения

Обучение «с учителем»

- Имеется неизвестная функция $Y=F(X)$, $X \in G \subseteq \mathbb{R}^n$, $Y \in \mathbb{R}^m$
- Задано множество $Q=\{(X^1,D^1) \dots (X^N,D^N)\}$, где $D^k=F(X^k)$, $X^k \in G$, $k=1,2\dots N$
- Найти функцию $A(X)$, вычисляемую ИНС и близкую к $F(X)$ в смысле некоторой функции ошибки E

$$E = 1/N \sum \|D^k - A(X^k)\|^2$$

Сеть обучена: $A(X)$ близка к $F(X)$ на множестве Q

Сеть способна к обобщению: $A(X)$ близка к $F(X)$ на всем множестве G

Качество обучения зависит:

- от количества обучающих примеров
- от полноты описания области определения G

Проблемы обучения

- **Недообучение**

Сеть недостаточно точно аппроксимирует данные обучающего множества (не способна обучиться)

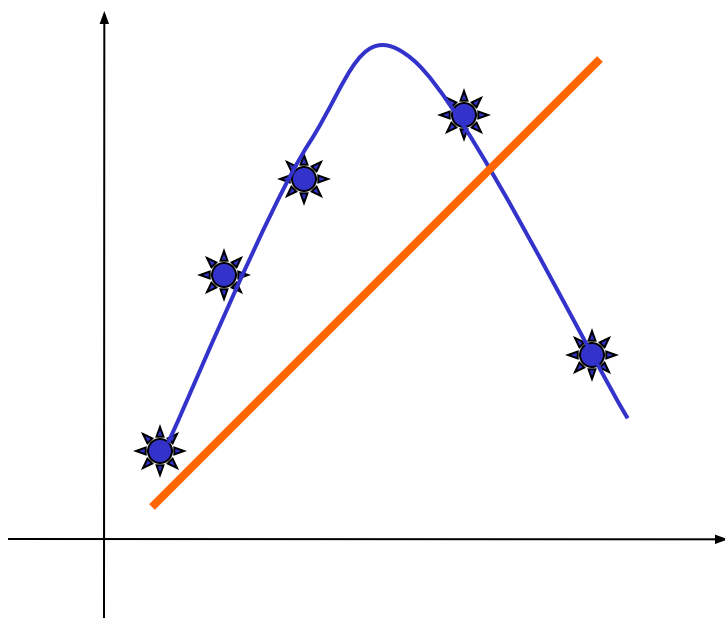
Усложнить сеть

- **Переобучение**

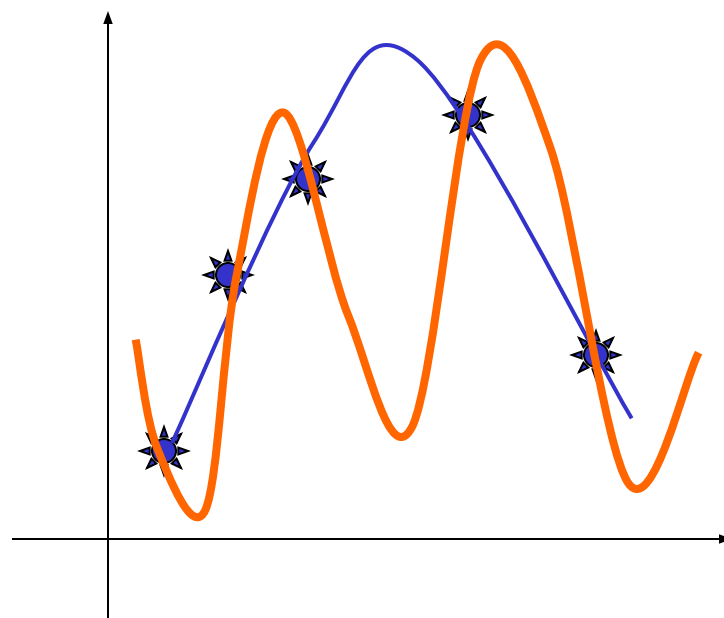
Сеть чрезмерно точно аппроксимирует данные обучающего множества (обучена, но не способна к обобщению)

Упростить сеть

Аналогия с полиномом аппроксимации



Степень полинома
недостаточна



Степень полинома
избыточна

Контрольное и тестовое множества

Как проверить качество обучения?

Кросс-проверка

$Q = Q_1 \cup Q_2$ (обучающее+контрольное)

- $E(Q_1)$ и $E(Q_2)$ убывают – обучение идет правильно
- $E(Q_1)$ убывает, $E(Q_2)$ возрастает – переобучение
- $E(Q_1)$ и $E(Q_2)$ не убывают – недообучение

Многократная проверка нескольких сетей

$Q = Q_1 \cup Q_2 \cup Q_3$ (обучающее+контрольное+тестовое)

Обучение ИНС – задача многомерной оптимизации

Задача оптимизации: $\min_W E(W)$

*Произвольный вид функции E , большая размерность
Многомерная невыпуклая задача оптимизации*

Алгоритмы

- Локальная оптимизация с использованием производных первого порядка

Градиентный метод, метод сопряженных градиентов

- Локальная оптимизация с использованием производных первого и второго порядков

Ньютона, Гаусса-Ньютона, разреженных матриц Гессе

- Стохастическая оптимизация

Монте-Карло, имитация отжига, генетические алгоритмы

Практика

Решение задачи классификации

```
import numpy as np
import pylab as pl
import neurolab as nl
```

Рисование линии, разделяющей классы

```
pl.plot([0., 5.],[5., 0.], 'b')
```

Создание и рисование обучающих примеров

```
n = 10
```

```
xtrain = np.random.uniform(0, 5, (n, 2))
```

```
for i in range(n):
```

```
    pl.plot(xtrain[i, 0], xtrain[i, 1], 'b+')
```

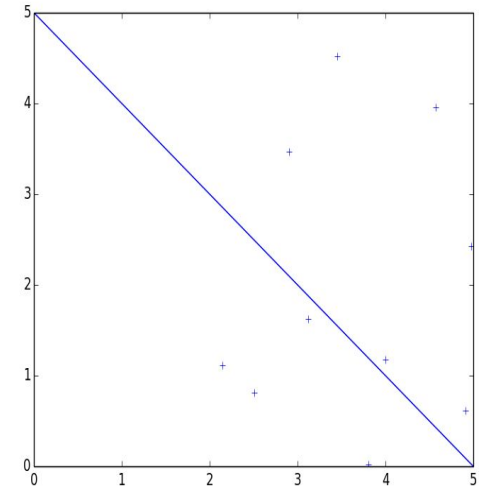
Вычисление правильных ответов

```
d = np.zeros(n)
```

```
for i in range(n):
```

```
    d[i] = np.sign(xtrain[i, 0] + xtrain[i, 1] - 5)
```

```
d = d.reshape(n, 1)
```



Практика

Решение задачи классификации

Создание сети прямого распространения с сигмоидными функциями активации

```
MyFirstNet = nl.net.newff([[0, 5], [0, 5]], [5, 1])
```

Опрос необученной сети

```
y = MyFirstNet.sim(xtrain)
```

```
for i in range(n):
```

```
    if y[i] > 0:
```

```
        pl.plot(xtrain[i, 0], xtrain[i, 1], 'ro')
```

```
    if y[i] < 0:
```

```
        pl.plot(xtrain[i, 0], xtrain[i, 1], 'go')
```

Практика

Решение задачи классификации

Обучение сети

```
err = MyFirstNet.train(xtrain, d, show=15)
```

Опрос обученной сети

```
y = MyFirstNet.sim(xtrain)
```

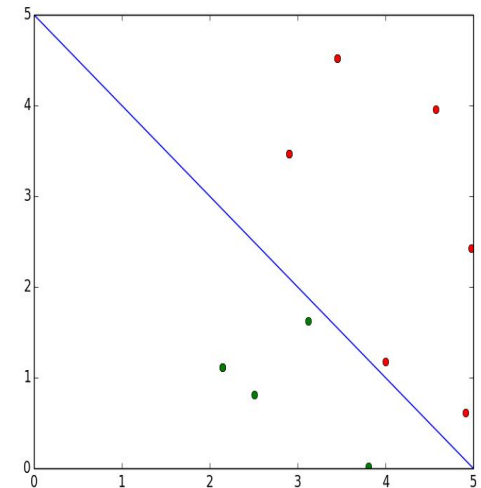
```
for i in range(n):
```

```
    if y[i] > 0:
```

```
        pl.plot(xtrain[i, 0], xtrain[i, 1], 'ro')
```

```
    if y[i] < 0:
```

```
        pl.plot(xtrain[i, 0], xtrain[i, 1], 'go')
```



Практика

Решение задачи классификации

Создание тестового множества

```
ntest = 25 * 25
```

```
xtest = np.zeros(ntest * 2)
```

```
xtest = xtest.reshape(ntest, 2)
```

```
i = 0
```

```
k0 = 0.
```

```
while k0 < 5:
```

```
    k1 = 0.
```

```
    while k1 < 5:
```

```
        xtest[i, 0] = k0
```

```
        xtest[i, 1] = k1
```

```
        i += 1
```

```
        k1 += 0.2
```

```
    k0 += 0.2
```

Практика

Решение задачи классификации

Проверка способности сети к обобщению

```
y = MyFirstNet.sim(xtest)
```

```
for i in range(ntest):
```

```
    if y[i] > 0:
```

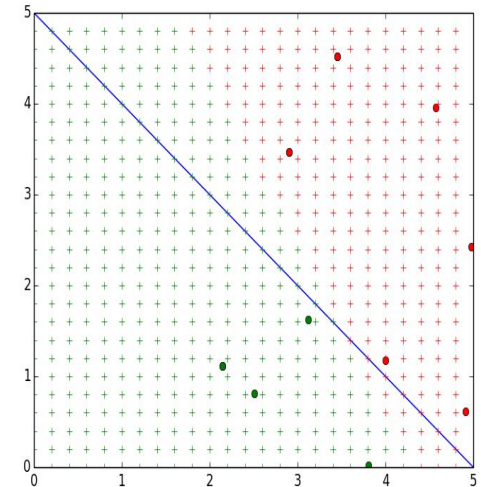
```
        pl.plot(xtest[i, 0], xtest[i, 1], 'r+')
```

```
    if y[i] < 0:
```

```
        pl.plot(xtest[i, 0], xtest[i, 1], 'g+')
```

```
pl.axis([0, 5, 0, 5])
```

```
pl.show()
```



Обучена ли сеть?

Способна ли сеть к обобщению? Хорошо ли описана область определения? Правильно ли подобрано число нейронов в скрытом слое?