

Введение в робототехнику

Лекция 9.
Нейронные сети

Введение в обучающиеся системы или системы приобретения знаний

На уровне знаний обучение сводится к выявлению и формализации новых знаний. Например, из фактов

журавль умеет летать,
воробей умеет летать,
синица умеет летать,
журавль есть птица,
воробей есть птица,
синица есть птица

система может сформулировать правило-продукцию

Если X есть птица
то X умеет летать.

По признаку применяемого метода обучения различают системы, в которых используются аналитические или эмпирические методы обучения. Аналитические в свою очередь делятся на использующие глубинные (knowledge-rich) или поверхностные (knowledge-driven) знания.

Эмпирические делятся на использующие знания (knowledge-learning) или данные (data-driven).

С другой стороны в инженерии знаний известны три основных подхода к приобретению знаний: индуктивный вывод, вывод по аналогии и обучение на примерах. В основе индуктивного вывода лежит процесс получения знаний из данных и/или других знаний (в продукционных системах – правил из фактов и/или других правил). Вывод по аналогии основан на задании и обнаружении аналогий между объектами (ситуациями, образами, постановками задачи, фрагментами знаний) и применением известных методов (процедур) к аналогичным объектам. В основе обучения на примерах лежит демонстрация системе и запоминание ей примеров решения задач. Резкой границы между этими методами не существует, т.к. все они базируются на обобщении, реализованной в той или иной форме, т.е. реализуют переход от более конкретного знания (фактов) к более абстрактному знанию.

Виды обучающихся систем



ДСМ-метод

В ДСМ-методе используется представление знаний об экспериментах (наблюдениях), подтверждающих причинно-следственные связи между факторами, в виде матрицы гипотез

$$M^* = \begin{array}{c|cccc} a \backslash b & b_1 \dots & b_j & b_m \\ \hline a_1 & q_{11} & q_{1j} & q_{1m} \\ \dots & & & \\ a_i & q_{i1} & q_{ij} & q_{im} \\ \dots & & & \\ a_n & q_{n1} & q_{nj} & q_{nm} \end{array}$$

где: a_i – факторы-причины,
 b_j – факторы-следствия,
 q_{ij} – оценки истинности (силы) причинно-следственной связи между соответствующими факторами.

Оценка истинности определяется в процессе обучения (экспериментов или наблюдений) как k^+/k , k -общее количество экспериментов (примеров), а k^+ – количество экспериментов, подтверждающих причинно-следственную связь (положительных примеров).

Таким образом, после обучения мы имеем матрицу со значениями q_{ij} в пределах интервала $(0,1)$. Если значение $q_{ij} \approx 1$, это означает, что между факторами a_i и b_j есть причинно-следственная связь и ее можно записать в виде правила.

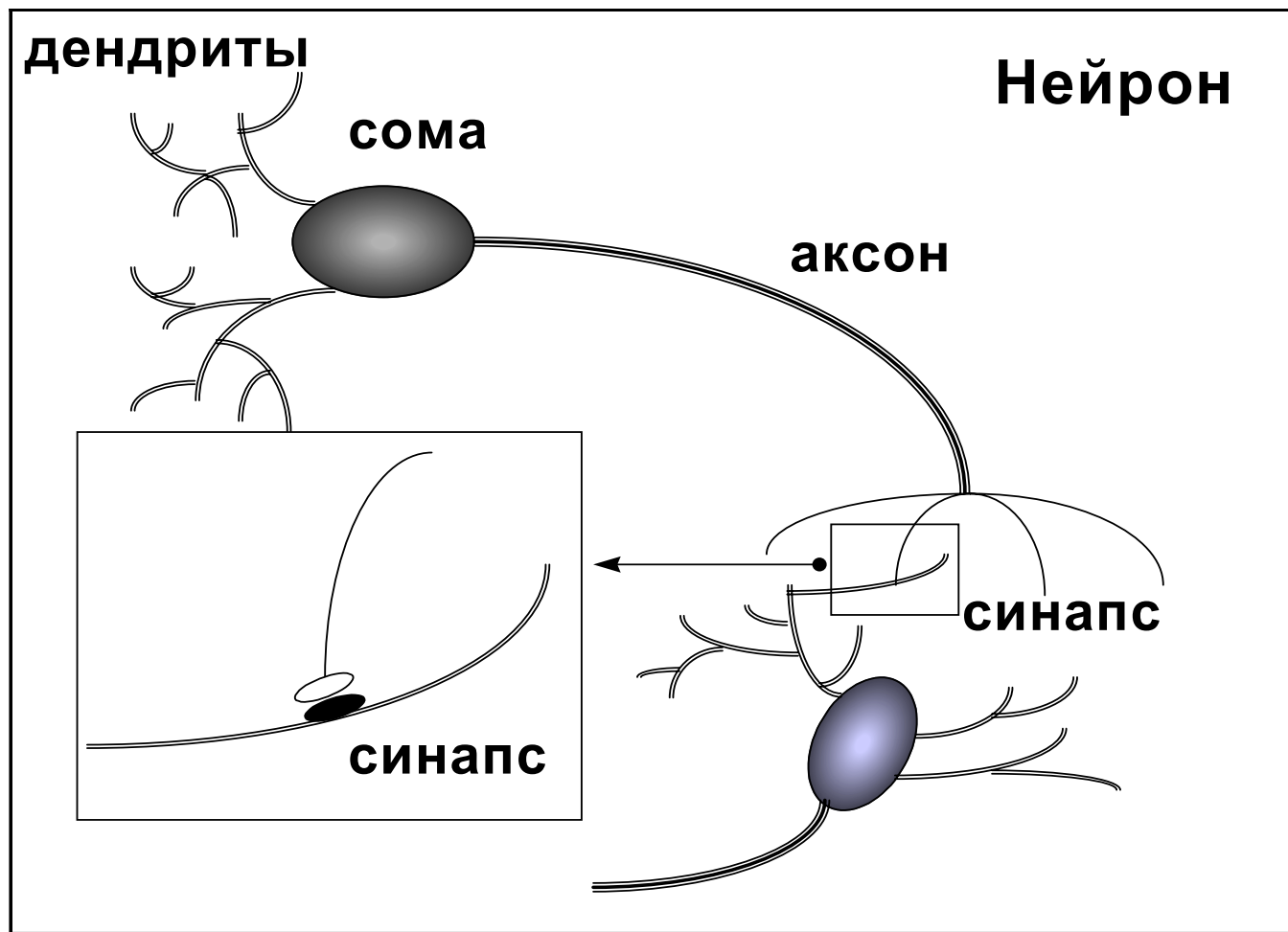
Иногда в ДСМ-методе используется и матрица отрицательных примеров M^- .

Для определения индуктивного вывода необходимо определить следующие его составляющие:

- 1) метод и форма предъявления данных (фактов или обучающих примеров);
- 2) метод индуктивного вывода (алгоритмы обработки фактов);
- 3) что и в каком виде мы хотим получить (вывести);
- 4) условия, при которых можно считать, что вывод завершен.

Нейронные сети

Биологический нейрон

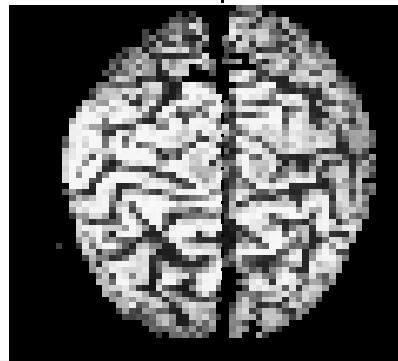


◆ Традиционные ЭВМ

- ◆ Последовательные
- ◆ Заданный алгоритм
- ◆ Иерархическая структура алгоритмов, разбиение сложной задачи на простые



*Левое
полушарие*



◆ Нейрокомпьютеры

- ◆ Параллельные
- ◆ Алгоритм формируется путем обучения на примерах
- ◆ Непосредственные операции с образами

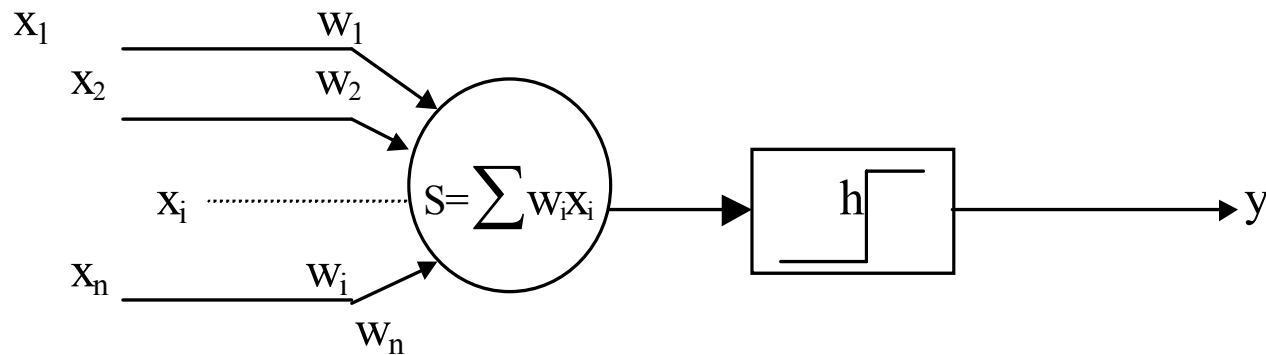
*Правое
полушарие*



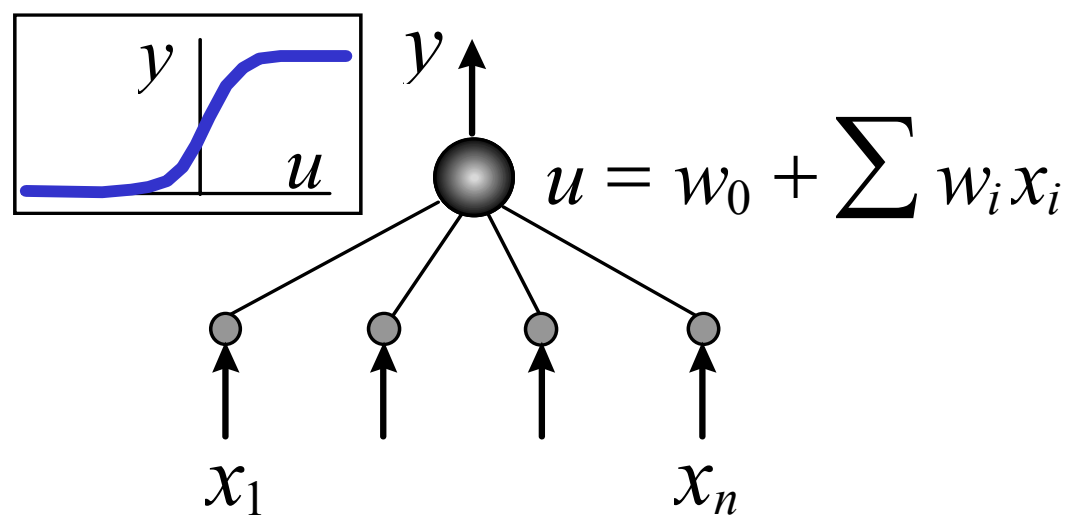
Парадигмы нейрокомпьютинга

- Коннекционизм (важность связей между узлами)
- Локальность и параллелизм вычислений
- Программирование: обучение, основанное на данных
- Универсальность обучающих алгоритмов

Модели нейронов.
Формальный нейрон Мак-Каллока-
Питтса (1943)
(пороговый нейрон)



Модель нейрона с сигмоидной функцией активации



Виды сигмоидальной активационной функции

- Рациональная
сигмоида
- Экспоненциальная
сигмоида
- Гиперболический
тангенс

$$f(S) = S / (|S| + a)$$

$$f(S) = 1 / (1 + \exp(-aS))$$

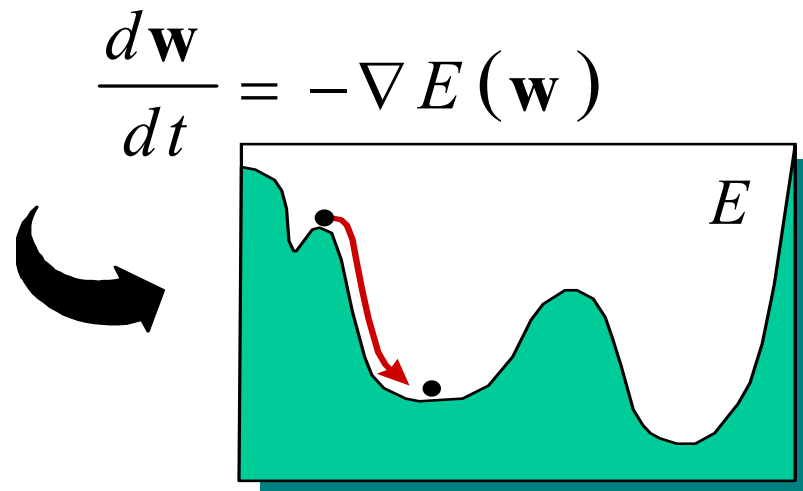
$$f(s) = \operatorname{th} \frac{s}{\alpha} = \frac{e^{-\frac{s}{\alpha}} - e^{\frac{s}{\alpha}}}{e^{-\frac{s}{\alpha}} + e^{\frac{s}{\alpha}}}$$

Обучение сети

- Обучить нейронную сеть это значит, сообщить ей, чего от нее добиваются.
- Показав ребенку изображение буквы и получив неверный ответ, ему сообщается тот, который хотят получить.
- Ребенок запоминает этот пример с верным ответом и в его памяти происходят изменения в нужном направлении.

Обучение как задача ОПТИМИЗАЦИИ

Ошибка – разница между требуемым и фактическим
ВЫХОДОМ



Типы обучения НС

<i>Вид обучения:</i>	<i>С "учителем"</i>	<i>С "подкреплением"</i>	<i>Без "учителя"</i>
Что подается в качестве обучающих примеров	Набор пар входов-выходов $\{x^\alpha, y^\alpha\}$	Оценка выходов сети $\{x^\alpha, y(x^\alpha)\}$	Только набор входных значений $\{x^\alpha\}$
Что требуется от сети	Найти функцию, обобщающую примеры, в случае дискретных y^α - классифицировать входы. В целом - научиться реагировать схожим образом в схожих ситуациях.	Научиться заданной "правильной" линии поведения.	Найти закономерности в массиве данных, отыскать порождающую данные функцию распределения, найти более компактное описание данных.

Задачи в робототехнике: автономное поведение, картографирование
обучение манипулированию

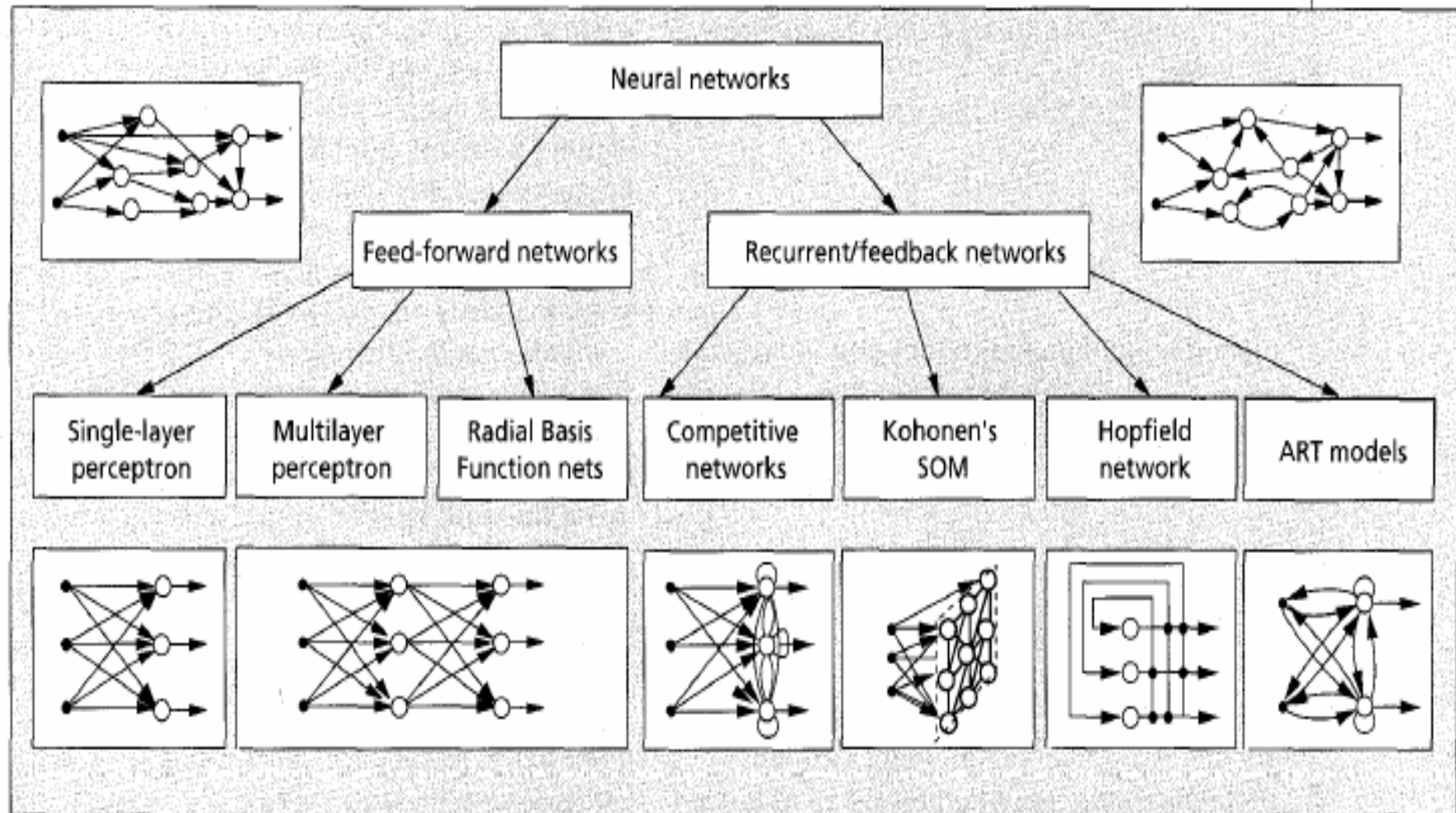
Классификация НС

<i>Тип обучения (Coding) →</i>	<i>С "учителем"</i>	<i>Без "учителя"</i>
<i>Тип связей (Decoding) ↓</i>		
<i>Без обратных связей</i>	<i>Многослойные перцептроны (аппроксимация функций, классификация)</i>	<i>Соревновательные сети, карты Кохонена (сжатие данных, выделение признаков)</i>
<i>С обратными связями</i>	<i>Рекуррентные аппроксиматоры (предсказание временных рядов, обучение в режиме on-line)</i>	<i>Сеть Хопфилда (ассоциативная память, кластеризация данных, оптимизация)</i>

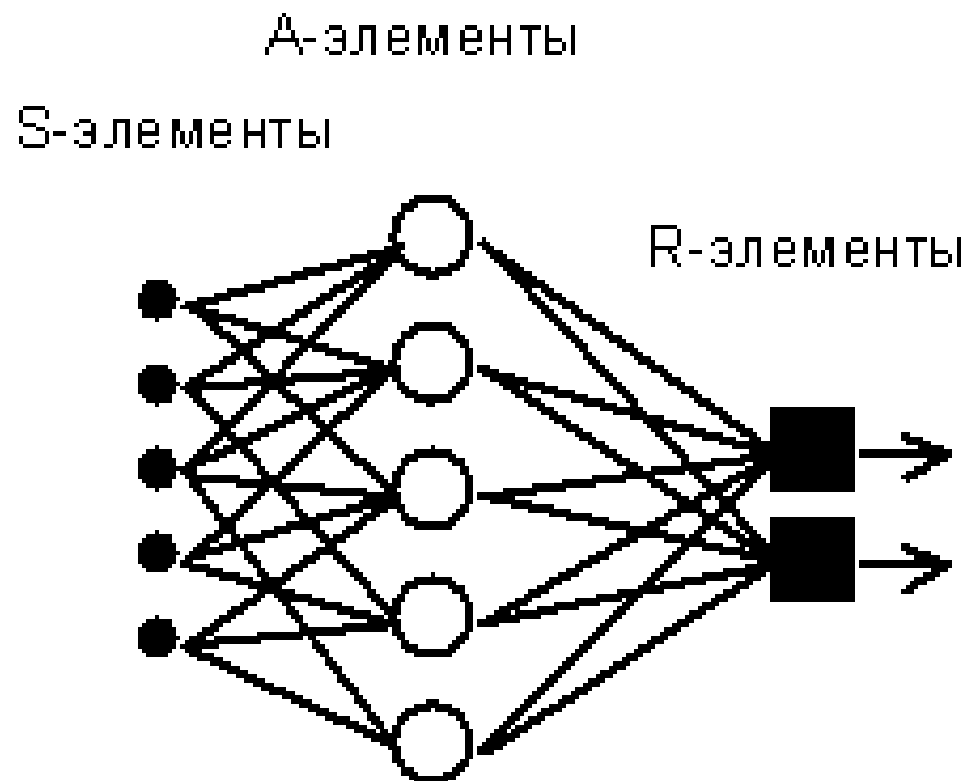
Классификация нейронных сетей

- Прямое распространение
 - Однослойный персептрон
 - Многослойный персептрон (MLP)
 - Радиально-базисные сети (RBF)
- Рекуррентные (с обратной связью)
 - Соревновательные сети (победитель получает все)
 - Самоорганизующиеся карты Кохонена (SOM)
 - Модель Хопфилда
 - Модели Адаптивного резонанса (ART)

Классификация нейронных сетей (2)

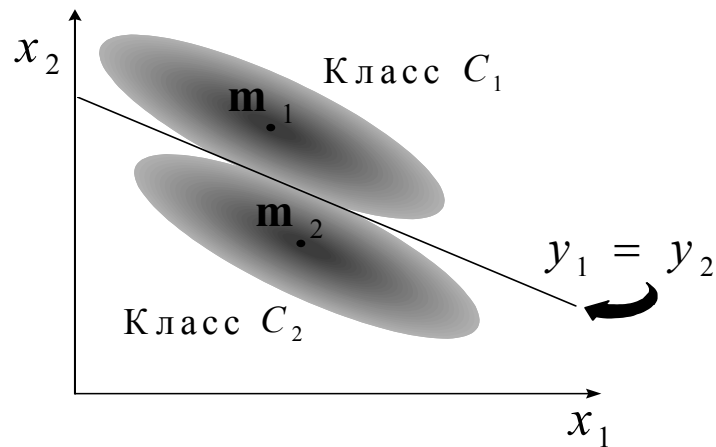


Персептрон Розенблатта

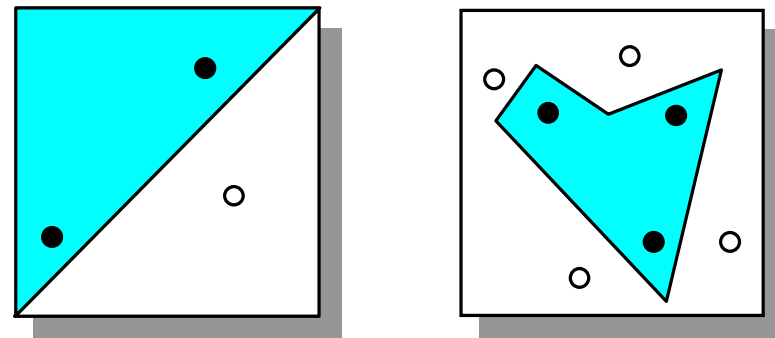


Задачи, решаемая НС - 1) классификация образов

$$y = f\left(\sum_{j=1}^d w_j x_j + w_0\right) \equiv f\left(\sum_{j=0}^d w_j x_j\right)$$

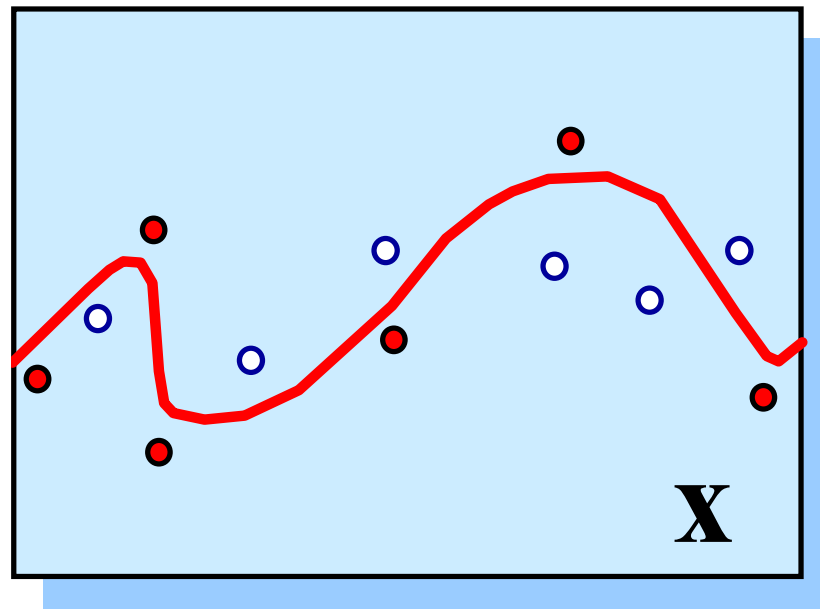


Двухслойный персептрон

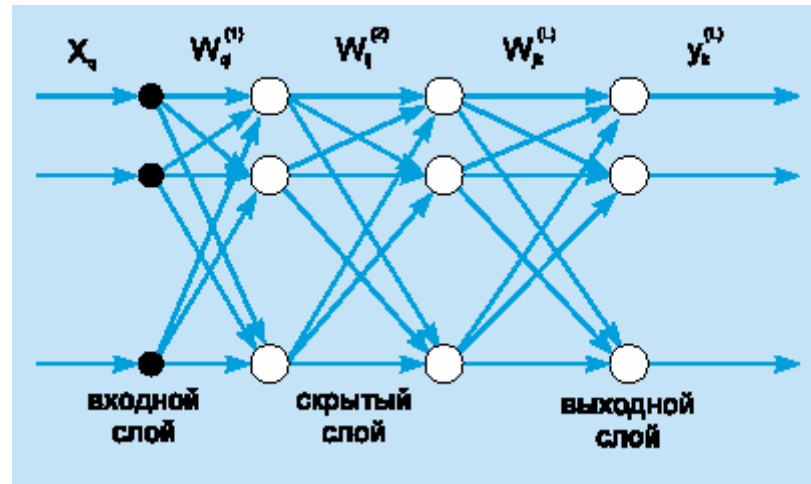


Задачи, решаемые НС -

2) аппроксимация функции



Персептроны



- **Классификация** (дискретный набор выходных значений)
- **Регрессия** (непрерывные выходные значения)

Доказанные возможности персептрона со скрытым слоем

- Сеть с одним скрытым слоем, содержащим N нейронов со *ступенчатой* функцией активации, способна осуществить произвольную классификацию Nd точек d -мерного пространства (т.е. классифицировать Nd примеров).
- Одного скрытого слоя нейронов с *сигмоидной* функцией активации достаточно для аппроксимации любой границы между классами со сколь угодно высокой точностью.
- Одного скрытого слоя нейронов с *сигмоидной* функцией активации достаточно для аппроксимации любой функции со сколь угодно высокой точностью. (Более того, такая сеть может одновременно аппроксимировать и саму функцию и ее производные.)

Обучение персептрона

Начальные значения весов всех нейронов полагаются случайными.

$$W(t = 0)$$

Сети предъявляется входной образ x^α , в результате формируется выходной образ.

$$y^\alpha \neq y^\alpha$$

Обучение персептрона

- Вычисляется вектор ошибки, делаемой сетью на выходе. $\delta^x = (y^x - \hat{y}^x)$

Идея: изменение вектора весовых коэффициентов в области малых ошибок должно быть пропорционально ошибке на выходе.

Алгоритм обучения обратным распространением ошибки (Error Back Propagation)

$$w_{i,j}(t+1) = w_{i,j}(t) + rg_j x_i,$$

где w_{ij} – вес от нейрона i или от элемента входного сигнала i к нейрону j в момент времени t ;

x_i – выход нейрона i или i -ый элемент входного сигнала;

r – шаг обучения;

g_j – значение ошибки для нейрона j .

Если нейрон с номером j принадлежит последнему слою, то

$$g_j = y_j(1 - y_j)(d_j - y_j),$$

где d_j – желаемый выход нейрона j ;

y_j – текущий выход нейрона j .

Если нейрон с номером j принадлежит одному из слоев с первого по предпоследний, то

$$g_j = x_j'(1 - x_j') \sum_k g_k w_{jk},$$

где k пробегает все нейроны слоя с номером на единицу больше, чем у того, которому принадлежит нейрон j .

Параметры

- Обучение проводится для всех обучающих векторов.
- Один цикл предъявления всей выборки называется **эпохой**.
- Обучение завершается по истечении нескольких эпох, когда вектор весов перестанет значительно меняться.

Обобщение

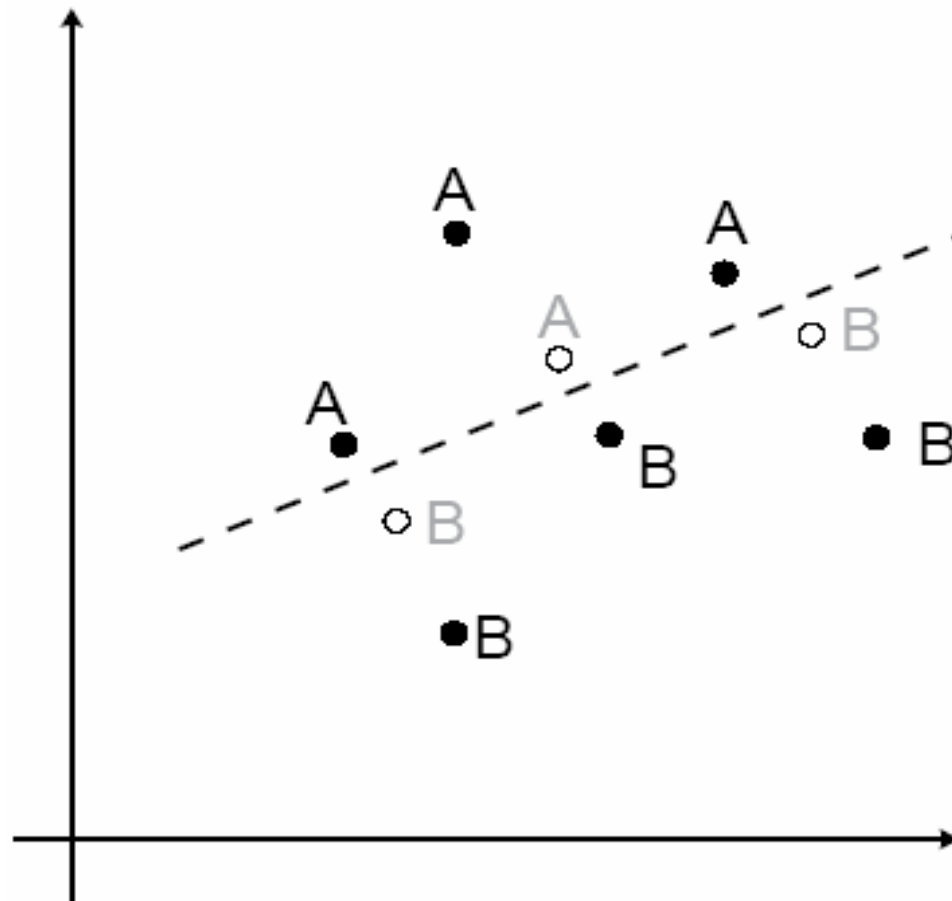
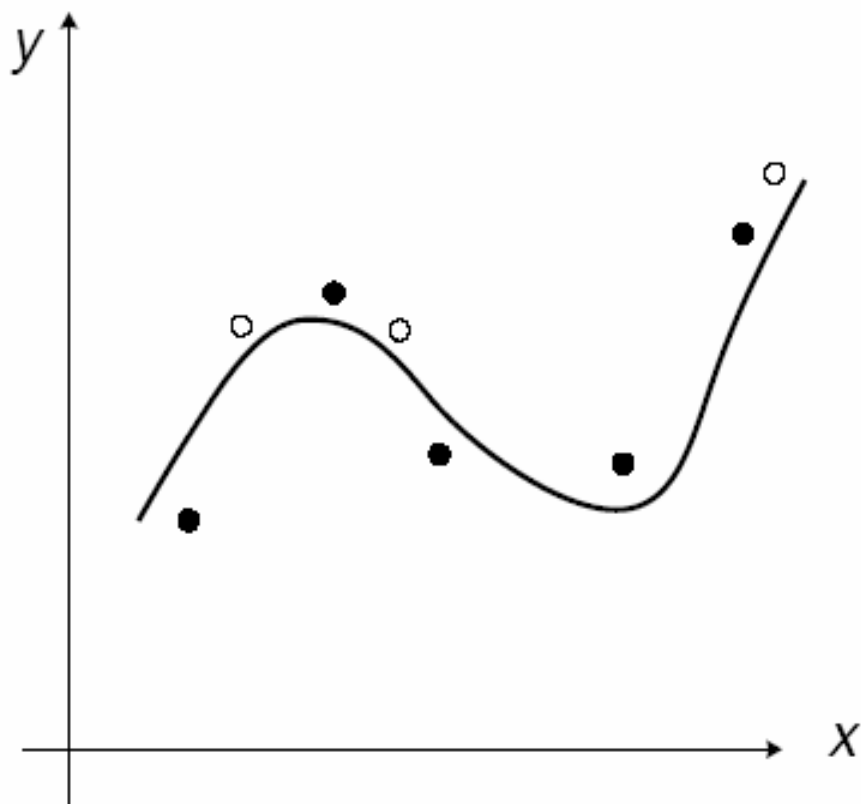
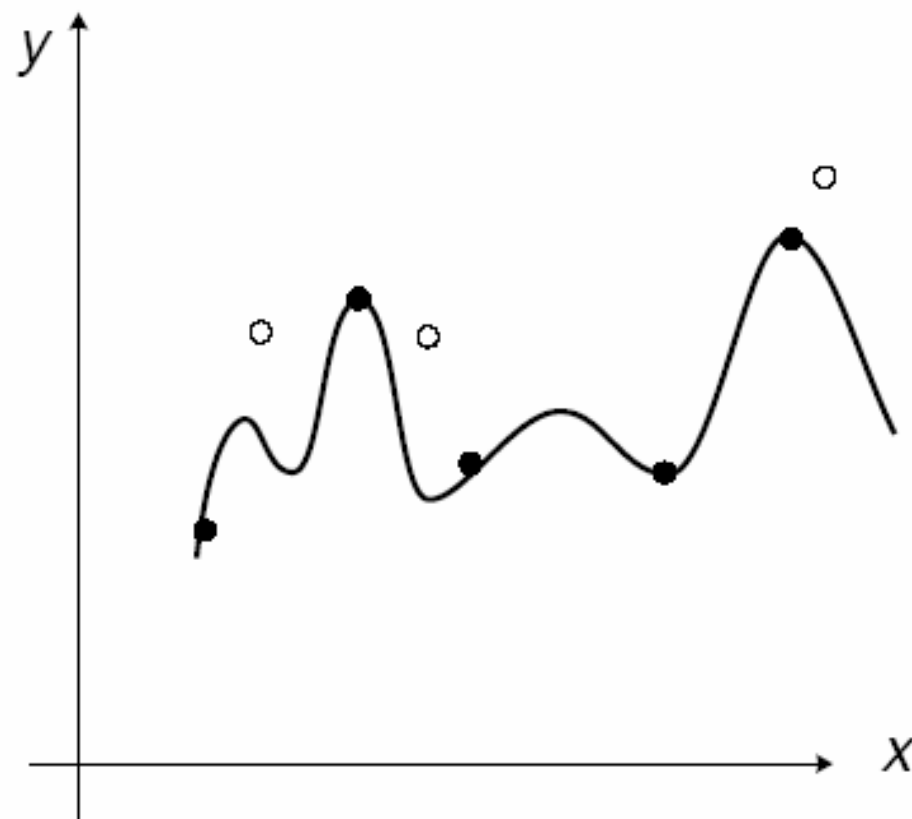


diagram sparse training - two classes

Переобученность (Overfitting)

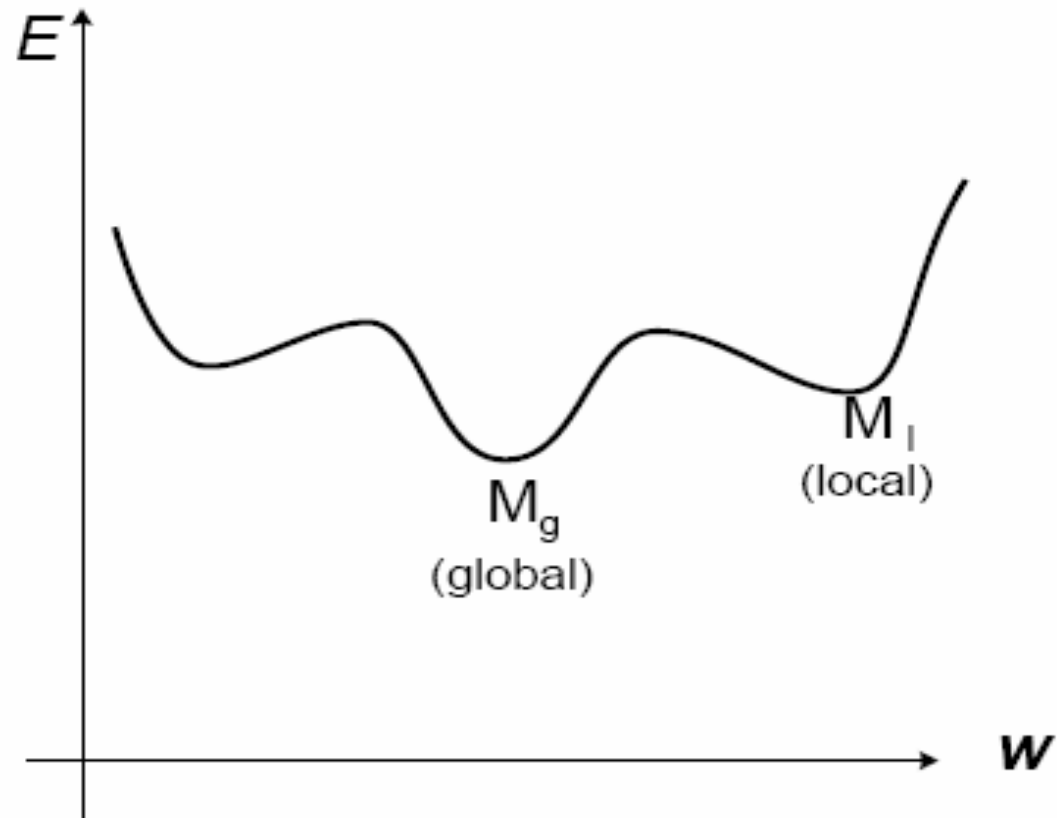


y against x for a binary classifier



overfitting in x-y space

Локальный минимум



local minimum

Достоинства и недостатки многослойного персептрона с обучением обратным распространением ошибки

- Достоинства:
 - Гарантирована возможность решения задачи
- Недостатки:
 - Низкая скорость обучения
 - Возможность переобученности
 - Невозможность переобучения (incremental learning)
 - Необходимо выбирать структуру сети под конкретную задачу (обычно это проблема)

Этапы построения сети

- Выбор модели сети
- Выбор архитектуры сети
 - Число входов
 - Функции активации
 - Как соединить нейроны
 - Что взять за вход, что за выход
- Подбор весов (обучение сети)
 - Программировать сеть или
 - Воспользоваться готовым пакетом нейросетевого моделирования
- Использование обученной сети для решения задачи

Различные подходы к обучению

- Обучение основанное на коррекции ошибки
 - Используется ошибка (d-y) между желаемым и реальным выходом для изменения весов связей с целью уменьшения ошибки
- Вероятностное обучение (машина Больцмана)
 - Адаптация весов «видимых» нейронов (входных и выходных) так, чтобы состояния видимых нейронов соответствовали желаемому распределению вероятностей
- Правило Хебба – усиливаются связи между нейронами с похожими (или одинаковыми) состояниями и наоборот

$$w_{ij}(t + 1) = w_{ij}(t) + \eta y_j(t) x_i(t)$$

при состояниях нейронов 1 и -1

- Соревновательное обучение
 - Только один нейрон (победитель) в выходном слое активизируется (*winner-take-all*, «победитель получает все»)