

# Введение в обучающиеся системы и нейронные сети

Интеллектуальные системы в машиностроении  
Лекция 5.1

# 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



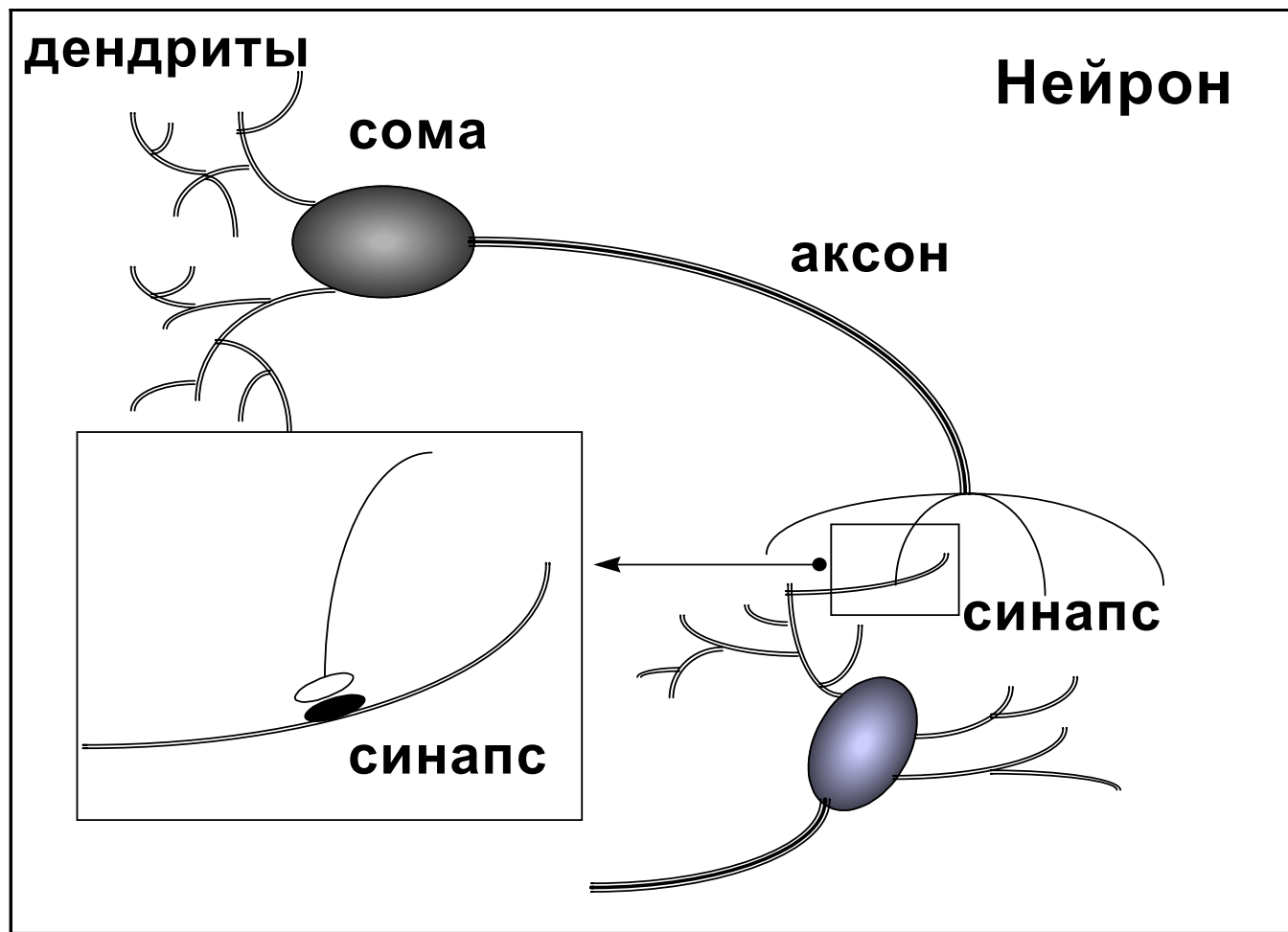
## Примеры применения нейронных сетей

- **Автопилотируемый гиперзвуковой самолет-разведчик.** Названный LoFLYTE (Low-Observable Flight Test Experiment) реактивный беспилотный самолет длиной 2,5 м был разработан для NASA и Air Force фирмой Accurate Automation Corp., Chattanooga, TN в рамках программы поддержки малого инновационного бизнеса. Это экспериментальная разработка для исследования новых принципов пилотирования, включая нейронные сети, позволяющие автопилоту обучаться, копируя приемы пилотирования летчика.
- **Системы безопасности в аэропортах.** Американская фирма SAIC (Science Application International Corporation) использовала нейронные сети в своем проекте TNA. TNA представляет собой ящик стоимостью \$750.000, который способен обнаруживать пластиковую взрывчатку в запакованном багаже. TNA бомбардирует багаж медленными нейтронами, вызывающими вторичное гамма-излучение, спектр которого анализируется нейронной сетью.

- **Нейросети на финансовых рынках.** Американский Citibank использует нейросетевые предсказания с 1990 года. В 1992 году, по свидетельству журнала The Economist, автоматический дилинг показывал доходность 25% годовых, что намного превышает показатели большинства брокеров. Chemical Bank использует нейро-систему фирмы Neural Data для предварительной обработки транзакций на валютных биржах 23 стран, фильтруя “подозрительные” сделки. Fidelity of Boston использует нейросети при управлении портфелями с суммарным объемом \$3 миллиарда. Полностью автоматизированные системы ведения портфелей с использованием нейросетей применяют, например, Deere & Co - на сумму \$100 млн и LBS Capital - на сумму \$400 млн. В последнем случае экспертная система объединяется с примерно 900 нейросетями.
- **Распознавание краденных кредитных карт.** Продукт Falcon (Сокол), выпущенный в 1992 г. фирмой HNC Software Corp., контролирует сейчас более 220 млн карточных счетов, выявляя и предотвращая в реальном времени подозрительные сделки по, возможно, краденным кредитным/дебетным картам. Искусственные нейросети обучаются типичному поведению клиентов, различая резкую смену характера покупок, сигнализирующую о возможной краже.

- **Активная реклама в Internet.** Нейросетевой продукт SelectCast фирмы Aptex Software Inc. (дочерней фирмы HNC Software Corp.) выявляет профили интересов пользователей Internet и предлагает им соответствующим образом отфильтрованную рекламу.
- **Мониторинг и персонализированная рассылка новостей.** Распознавание темы текстовых сообщений - другой пример успешного использования искусственных нейросетей. Сервер новостей Convectis (продукт все той же Aptex Software Inc. <http://www.aptex.com>) выбран в июне 1997г. лидером персонализированной доставки новостей в Internet - PointCast Inc. - для автоматической рубрикации сообщений по категориям. Сверяя значения слов по контексту, Convectis способен в реальном масштабе времени распознавать тематику и рубрицировать огромные потоки текстовых сообщений, передаваемых по сетям Reuters, NBC, CBS и др. Так, например, информационное агентство Scoop, специализирующееся на поставке бизнес-новостей и также лицензировавшее Convectis в июле 1997г., использует свыше 1600 источников информации. После анализа сообщения Convectis генерирует аннотацию, список ключевых слов и список рубрик, к которым относится данное сообщение.

# Биологический нейрон



В строении мозга как на фотографии запечатлен весь эволюционный путь его развития. Так, мозг человека фактически включает в себя три больших отдела, управляющие наиболее древней структурой (*нейрошасси*), доставшейся нам еще от рыб и амфибий, и ответственной за поддержание жизнедеятельности и размножение. Первый отдел - *рептильный комплекс* - возник несколько сот миллионов лет назад и помогает нам ориентироваться в пространстве. Второй отдел - *лимбическая система* - образовался около ста пятидесяти миллионов лет назад. Благодаря ей мы обладаем эмоциями. Наконец, последнее достижение - *новая кора*, особо развитая у человека, позволяет нам говорить и логически мыслить.

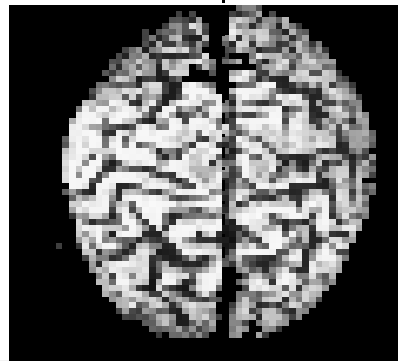
В дополнение к эволюционному разделению частей мозга, о котором мы говорили выше, в нервной системе явно различаются три типа нейронных структур: *сенсорные*, *внутренние* и *эффекторные*. Первые связывают наш мозг с внешним миром и обеспечивают поступление в него зрительной, слуховой, вкусовой, обонятельной и осязательной информации. Есть у нас и шестое чувство - чувство равновесия, за которое отвечает вестибулярный аппарат. Его особенность заключается в том, что он не вынесен наружу. Эффекторные нейроны управляют мышцами, внутренними органами, стенками сосудов и пр. Мозг таким образом контролирует работу сердца, дыхание, кровяное давление, температуру, поддерживает нужное содержание кислорода в крови, осуществляет гормональную регуляцию и пр. Промежуточные нейроны обрабатывают информацию, получаемую от сенсорных и передаваемую эффекторным нейронам.

## ◆ 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

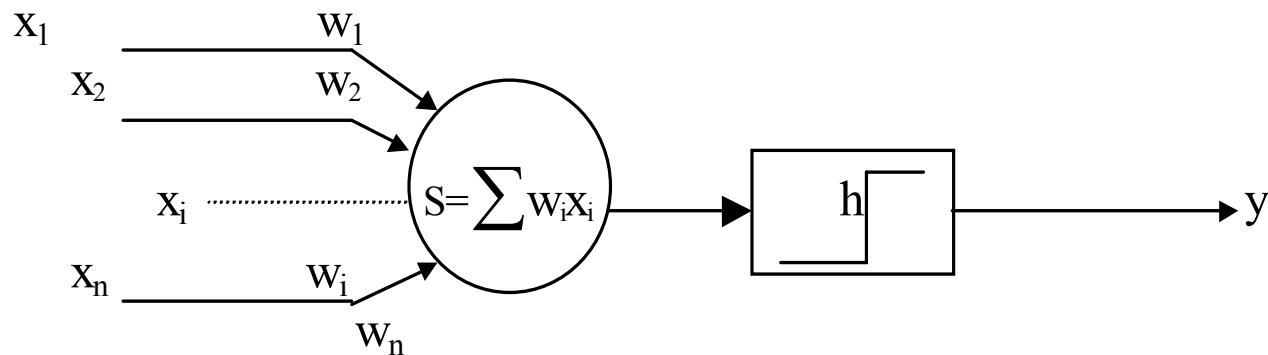


*Правое  
полушарие*

# Парадигмы нейрокомпьютинга

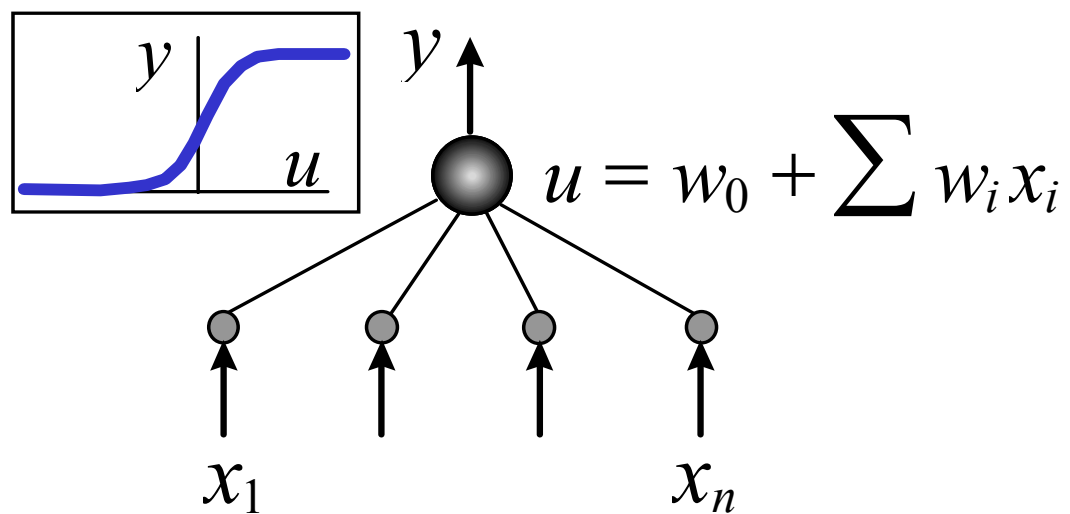
- Коннекционизм (важность связей между узлами)
- Локальность и параллелизм вычислений
- Программирование: обучение, основанное на данных
- Универсальность обучающих алгоритмов

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





# Модель нейрона с сигмоидной функцией активации



# Виды сигмоидальной активационной функции

- Рациональная сигмоида

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

- Экспоненциальная сигмоида

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

- Гиперболический тангенс

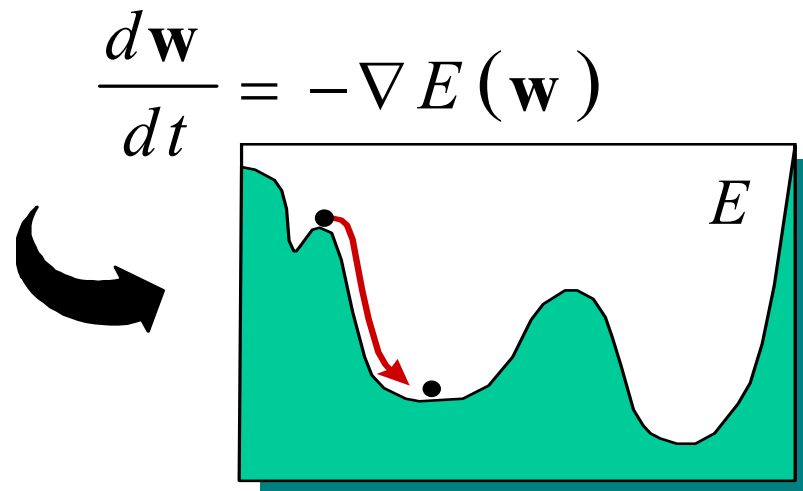
$$f(s) = \operatorname{th} \frac{s}{\alpha} = \frac{e^{-\frac{s}{\alpha}} - e^{\frac{s}{\alpha}}}{e^{-\frac{s}{\alpha}} + e^{\frac{s}{\alpha}}}$$

# 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 $y^\alpha$ - 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.

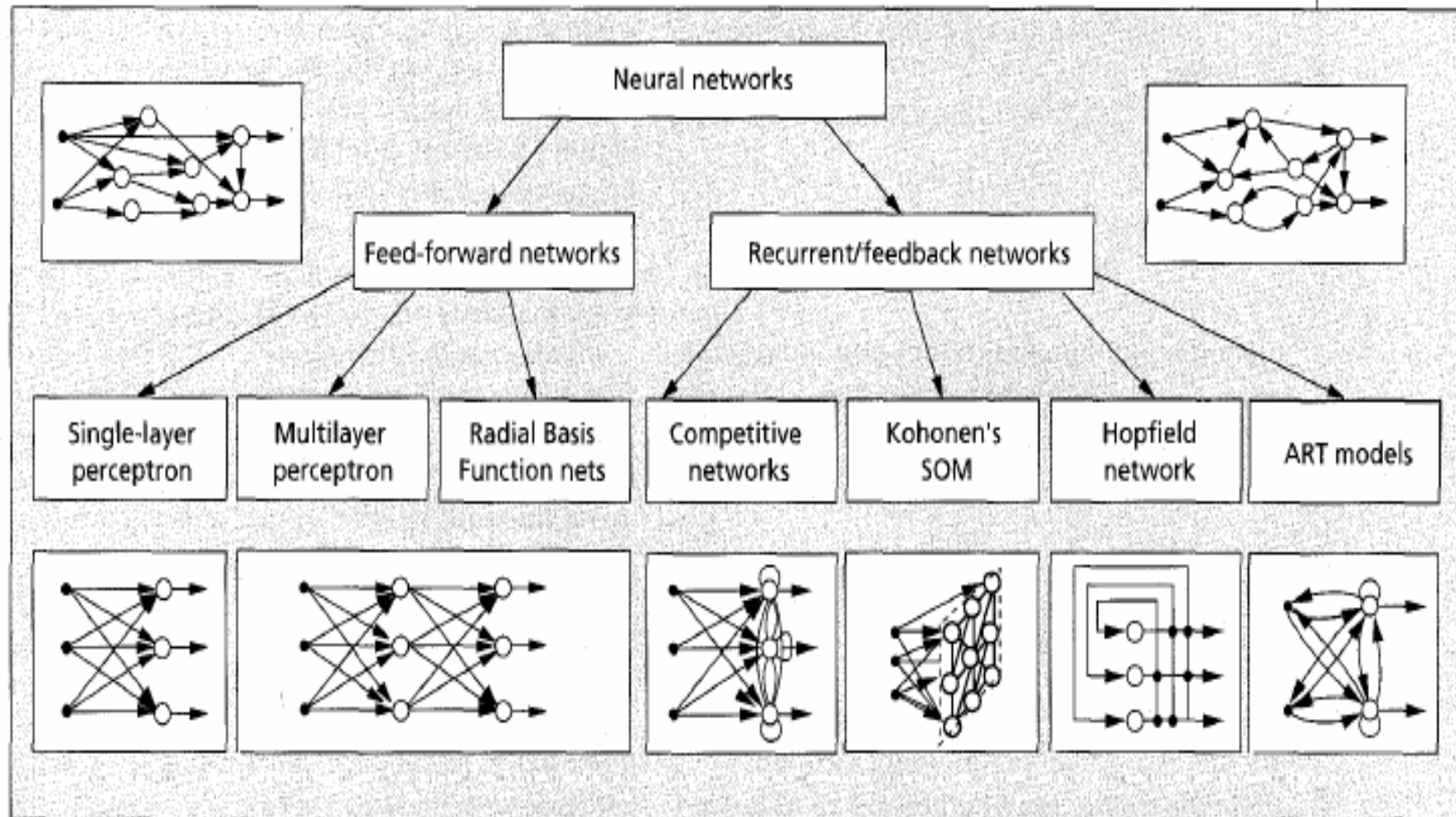
# Классификация НС

<i>Тип обучения (Coding) →</i> <i>Тип связей (Decoding) ↓</i>	<i>С "учителем"</i>	<i>Без "учителя"</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

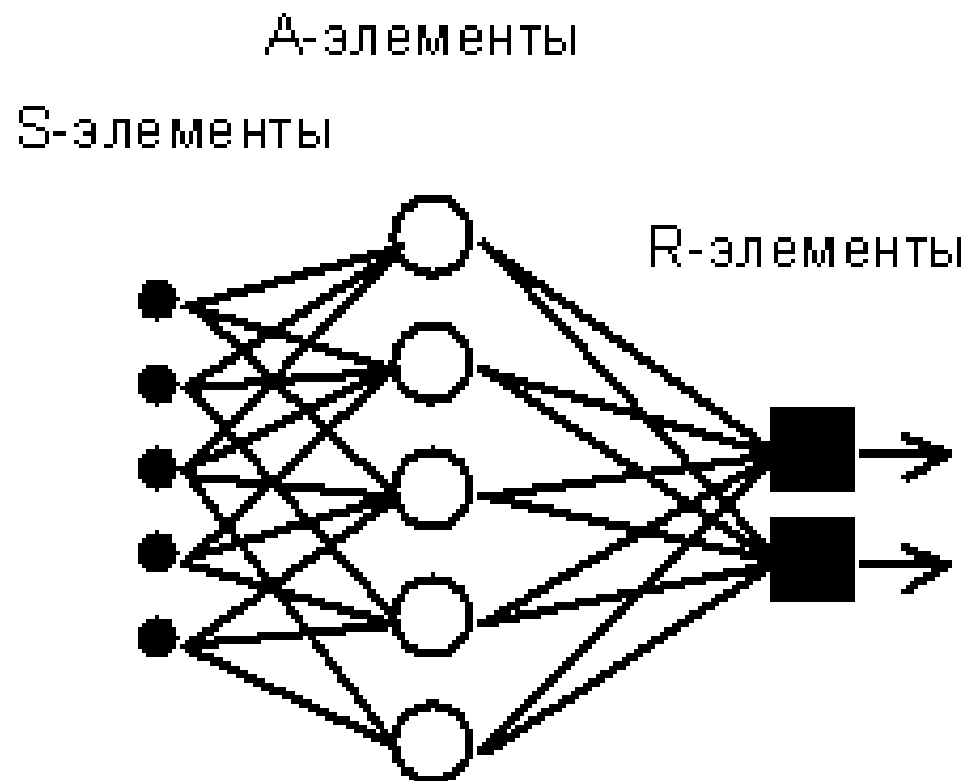
- Прямое распространения
  - Однослойный персептрон
  - Многослойный персептрон (MLP)
  - Радиально-базисные сети (RBF)
- Рекуррентные (с обратной связью)
  - Соревновательные сети (победитель получает все)
  - Самоорганизующиеся карты Кохонена (SOM)
  - Модель Хопфилда
  - Модели Адаптивного резонанса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

# Классификация нейронных сетей (2)



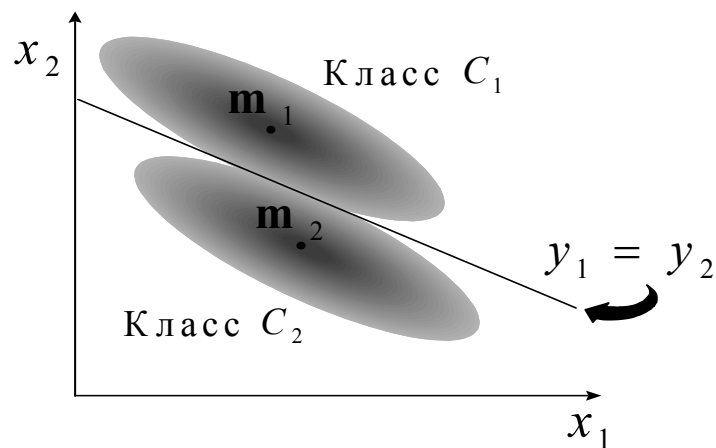


# Персептрон Розенблатта

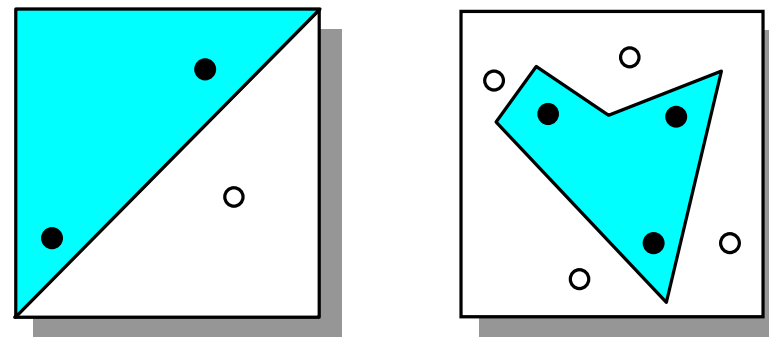


# Задачи, решаемая НС - 1) классификация образов

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

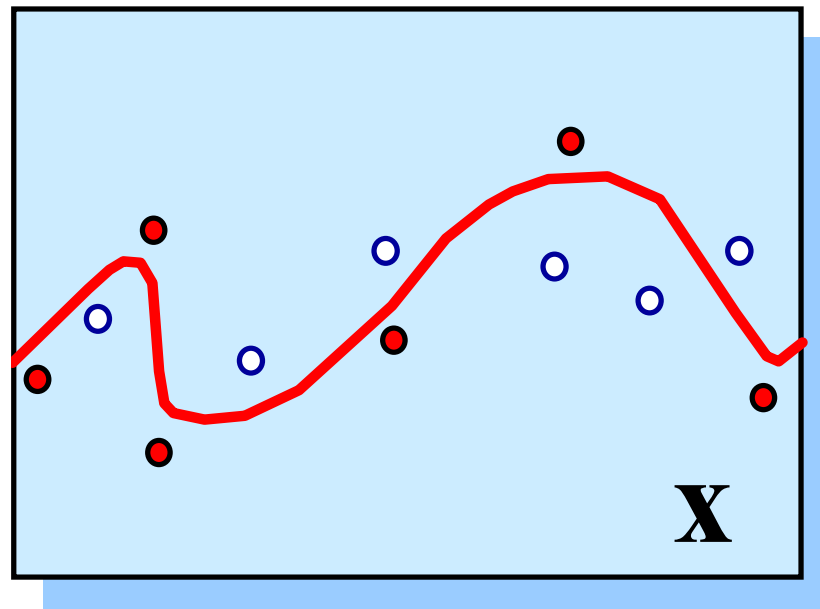


Двухслойный персептрон

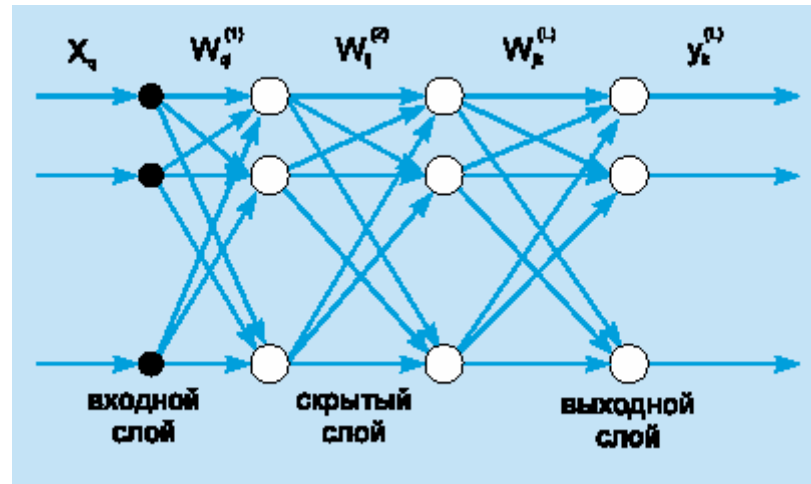


# Задачи, решаемые НС -

## 2) аппроксимация функции



# Перцептроны



- **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  $x^\alpha$ , в результате формируется выходной образ.

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

# Обучение персептрона

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

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

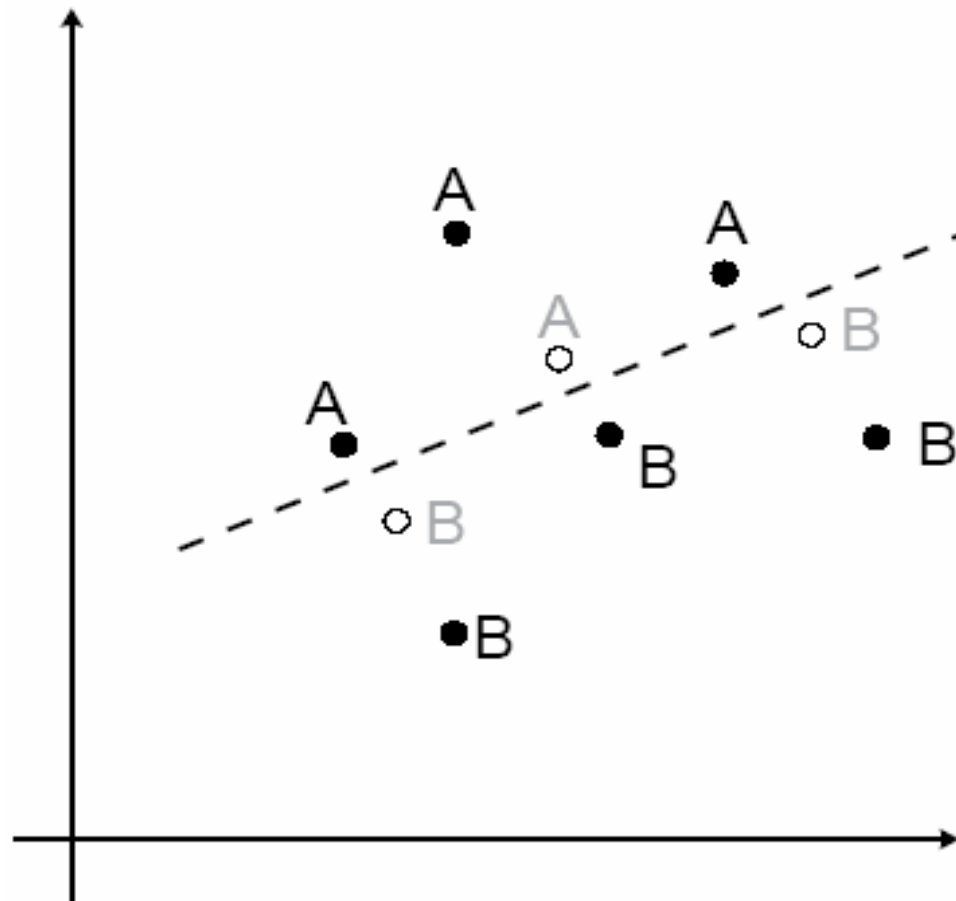
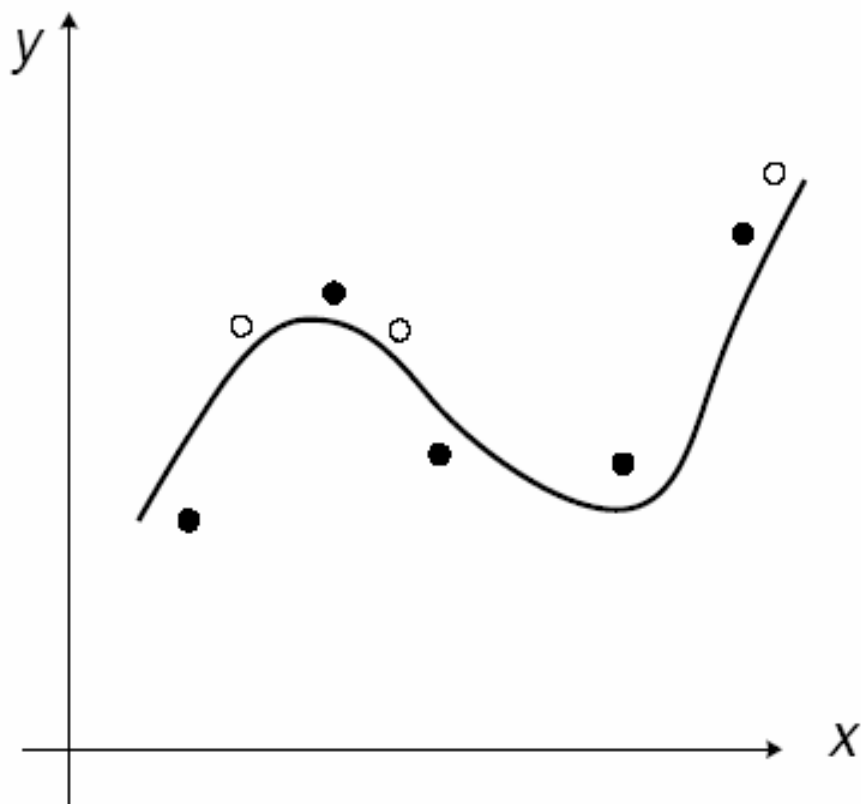
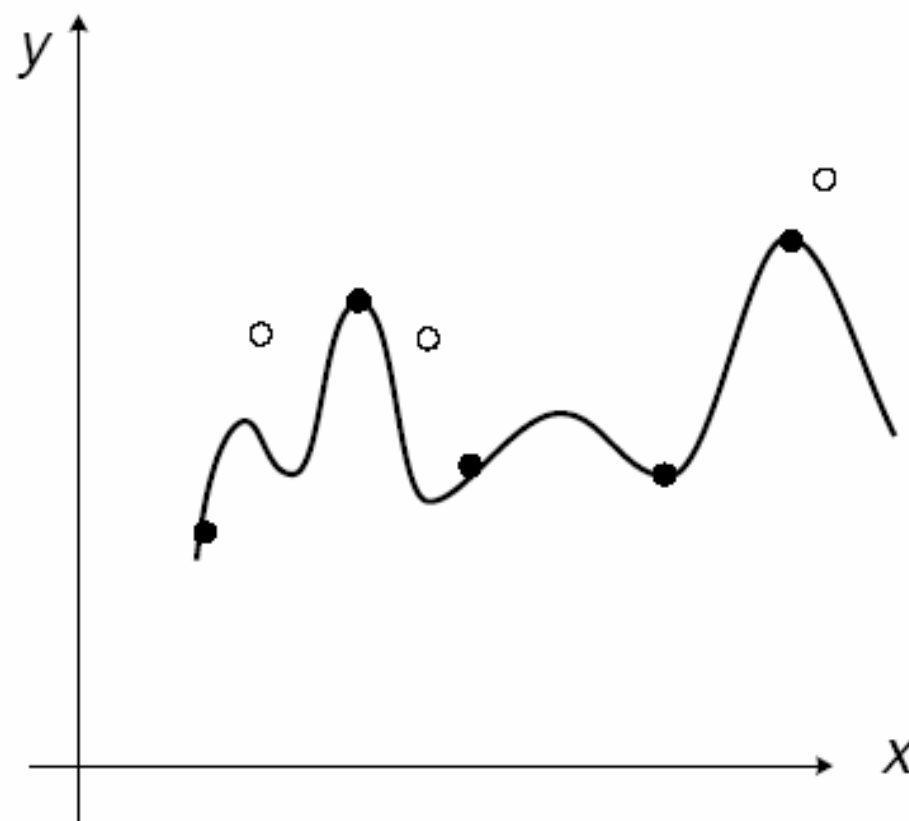


diagram sparse training - two classes

# Переобучение (Overfitting)

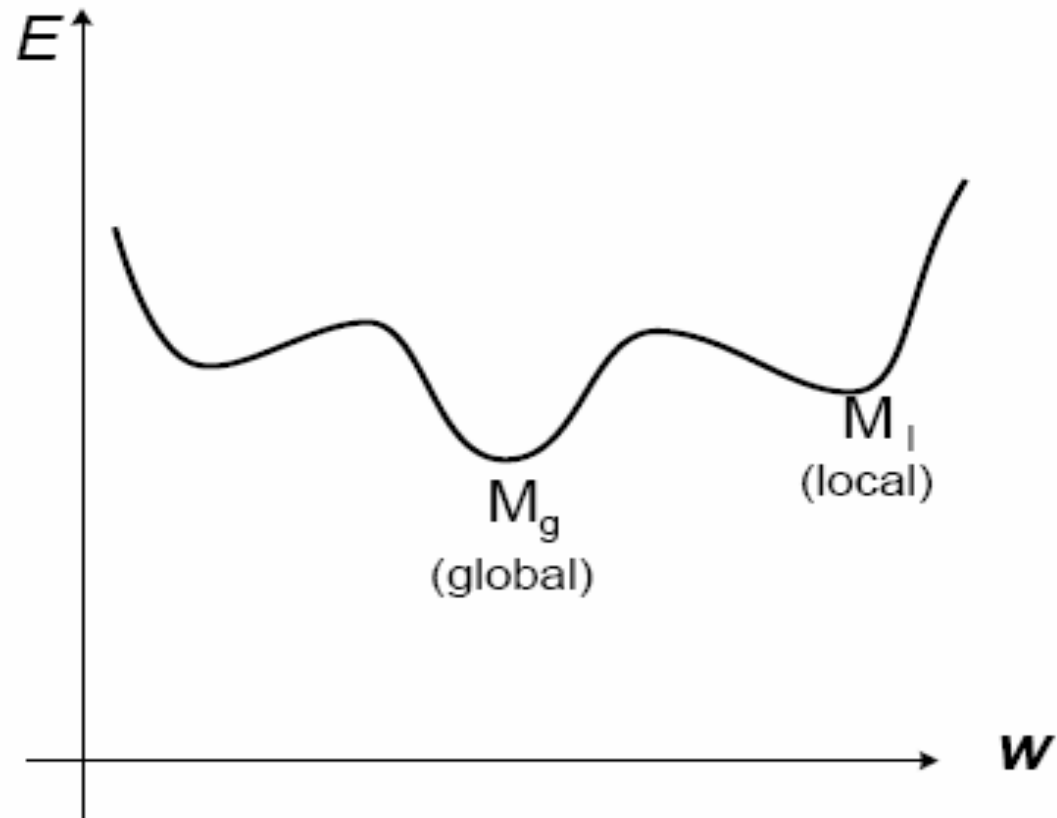


y against x for a binary classifier



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 переобучения
  - Невозможность переобучения
  - 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 так, чтобы состояния видимых нейронов соответствовали желаемому распределению вероятностей
- Правило Хебба – усиливаются связи между нейронами с похожими (или одинаковыми) состояниями и наоборот

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

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

# Подходы к решению задач с использованием нейронных сетей

- Использование консультационную фирму-исполнителя для решения задачи
- Использование оболочки, ориентированной на решение определенного типа задач
  - Metastock, Predictor, Deductor
- Использование универсальной оболочки
  - Toolbox of MATLAB, Statistica Neural Networks, Brain Maker, Neural Planner, Neural Bench,
- Разработка программной модели нейронной сети как специализированной программы (с использованием или без каких-либо компонентов для программирования нейронных сетей – SNNS, NeuralBase)
- Разработка программной модели как компоненты какого-либо прикладного ПО
  - G2
- Программирование аппаратной реализации нейронной сети