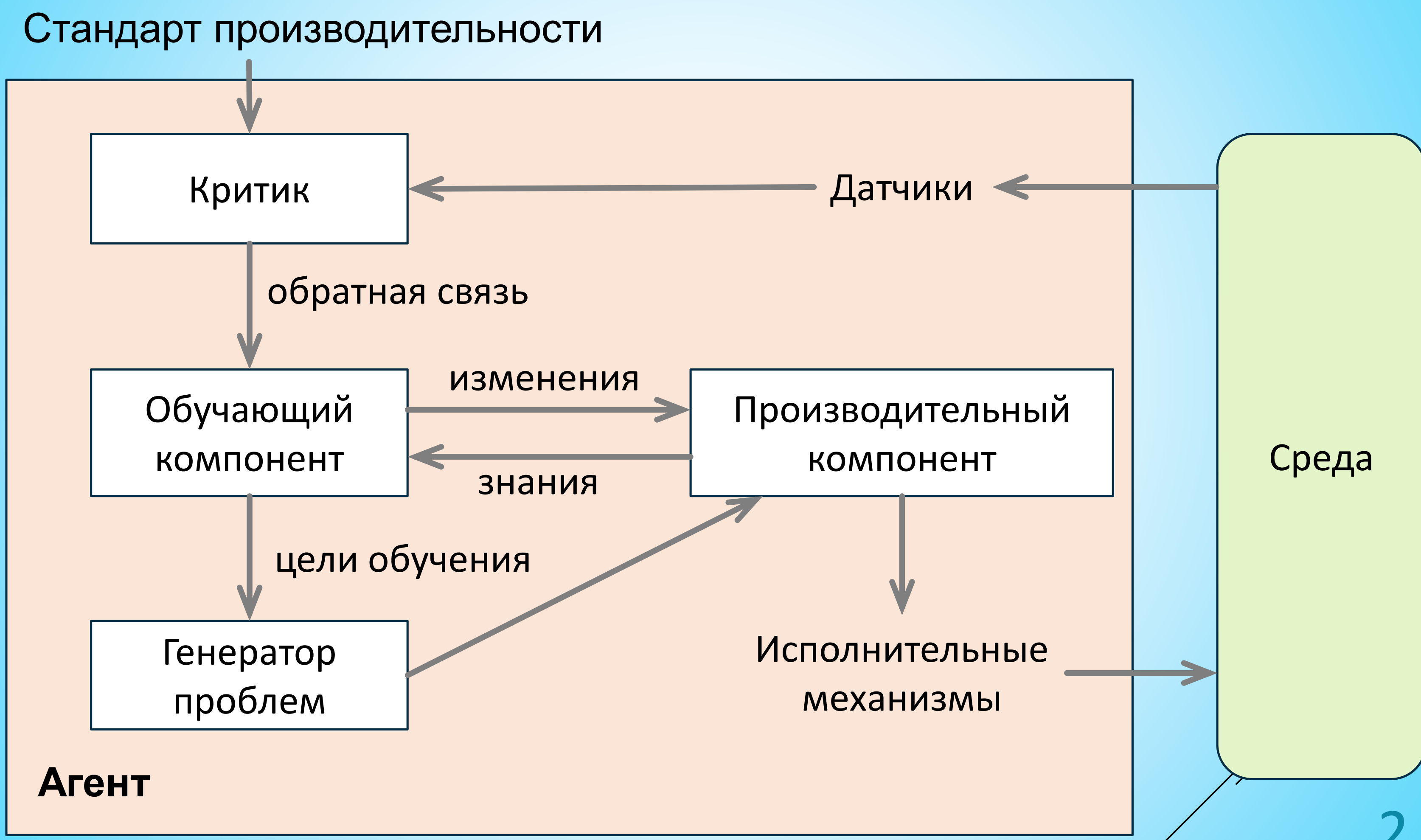
Федеральное государственное автономное образовательное учреждение высшего образования
Санкт-Петербургский национальный исследовательский университет
информационных технологий, механики и оптики

ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ

ЛЕКЦИЯ 5. ОБУЧЕНИЕ В ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМАХ. ДЕРЕВЬЯ РЕШЕНИЙ И НЕЙРОННЫЕ СЕТИ

к.т.н., Кашевник Алексей Михайлович,
alexey@iias.spb.su
к.т.н., Пономарёв Андрей Васильевич
ponomarev@iias.spb.su

СХЕМА ОБУЧАЮЩЕГОСЯ АГЕНТА



ВЫБОР ОБУЧАЮЩЕГО ЭЛЕМЕНТА

Аспекты, влияющие на проект обучающего элемента:

- Компоненты производительного элемента, подлежащие обучению
- Обратные связи, которые могут применяться для обучения этих компонентов
- Способы представления, используемые для компонентов

КОМПОНЕНТЫ ПРОИЗВОДИТЕЛЬНОГО ЭЛЕМЕНТА

1. Средства прямого отображения условий (распространяющихся на текущее состояние) в действия.

2. Средства логического вывода релевантных свойств мира из последовательности результатов восприятия.

3. Информация о том, как развивается мир и какие результаты возможных действий могут быть получены агентом.

4. Информация о полезности, которая показывает, насколько желательными являются те или иные состояния мира.

5. Информация о ценности действий, показывающая желательность действий.

6. Цели, описывающие классы состояний, достижение которых максимизирует полезность для агента.

ТИПЫ ОБРАТНОЙ СВЯЗИ

В области машинного обучения, как правило, различаются три случая:

- **контролируемое обучение** (обучение с учителем, supervised learning)
- **неконтролируемое обучение** (обучение без учителя, unsupervised learning)
- **обучение с подкреплением** (reinforcement learning)

СПОСОБЫ ПРЕДСТАВЛЕНИЯ ИНФОРМАЦИИ

- Функции полезности в виде полиномов (в программах ведения игр)
- Высказывания (в пропозициональной логике и логике первого порядка)
- Вероятностные описания (в системах вероятностного вывода)

ИНДУКТИВНОЕ ОБУЧЕНИЕ

Алгоритм контролируемого обучения получает в качестве исходной информации правильные значения неизвестной функции, соответствующие конкретным входным данным, и должен предпринять попытку восстановить эту неизвестную функцию.

Пример – пара (*x*, *f(x)*), где *x* – входное, а *f*(*x*) – выходное значение функции, применяемой к *x*.

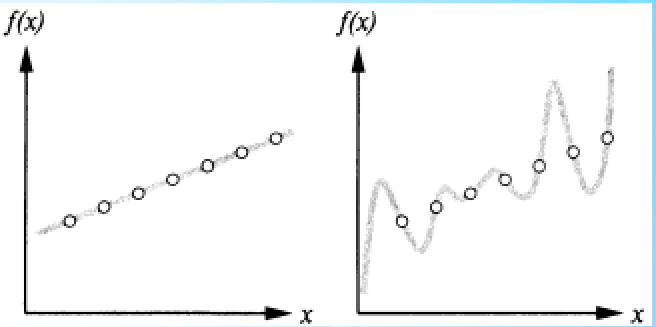
Основная задача **чисто индуктивного логического вывода** (или просто индукции):

На основании совокупности примеров входных и выходных данных функции f получить функцию h, которая аппроксимирует f.

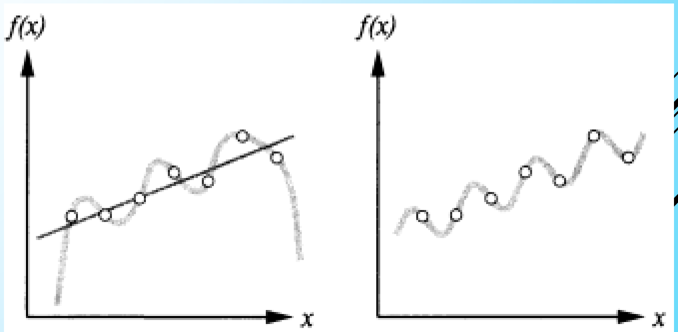
h – **гипотеза**, выбираемая обычно из **пространства гипотез** *H*.

ФАКТОРЫ, ВЛИЯЮЩИЕ НА ВЫБОР ГИПОТЕЗЫ

- Простые гипотезы, как правило, лучше сложных.



- При наличии недетерминированных функций (истинные входные данные не полностью наблюдаемы) приходится искать компромисс между сложностью гипотезы и степенью её согласования с данными.



- Необходимо найти компромисс между выразительностью пространства гипотез и сложностью поиска простой, совместимой гипотезы в этом пространстве.

ДЕРЕВО РЕШЕНИЙ

Дерево решений – это дерево, обладающее следующими метками:

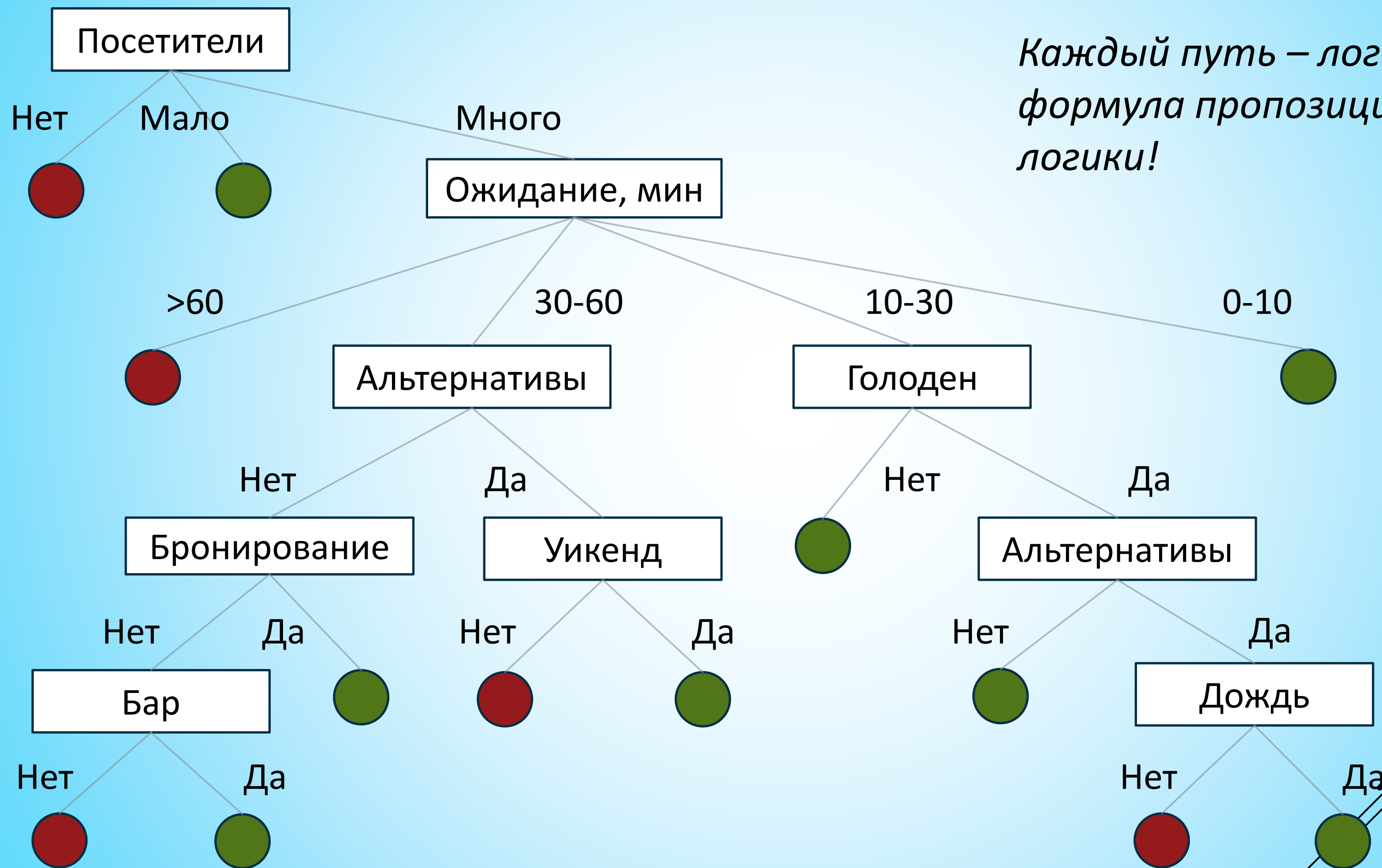
- У внутренней вершины – это атрибут.
- У листовой вершины – значение целевой функции.
- У ребра – значения атрибута.

Дерево решений принимает в качестве входных данных объект или ситуацию, описанную с помощью множества **атрибутов**, и возвращает «решение» – предсказанное выходное значение, соответствующее входным данным.

Входные атрибуты: дискретные или непрерывные.

Выходные значения: дискретные (**классификация**), непрерывные (**регрессия**).

ДЕРЕВО РЕШЕНИЙ. ПРИМЕР



Каждый путь – логическая формула пропозициональной логики!

Пример из *Рассел С., Норвиг П. Искусственный интеллект. Современный подход. 2 изд.*

ЗАДАЧА ИНДУКТИВНОГО ВЫВОДА ДЕРЕВА РЕШЕНИЙ (ДЛЯ БИНАРНОЙ КЛАССИФИКАЦИИ)

Пример для булева дерева решений состоит из вектора входных атрибутов *X* и одного булева выходного значения *y*.
Полное множество примеров – **обучающее множество**: $\{(x_1, y_1), \ldots, (x_n, y_n)\}$.
Положительные примеры – выходное значение истинно (y_i = *True*).
Отрицательные примеры – выходное значение ложно (y_i = *False*).

ВЫВОД ДЕРЕВА РЕШЕНИЙ. ПРОСТЕЙШИЙ АЛГОРИТМ

Пол	Сумма	Авто	Доход	Кредит
М	<=10	Есть	Средний	Одобрен
Ж	<=10	Есть	Средний	Одобрен
Ж	<=10	Нет	Низкий	Отказ
М	<=10	Нет	Низкий	Отказ
М	>10	Есть	Низкий	Отказ
Ж	>10	Нет	Высокий	Одобрен
М	>10	Есть	Высокий	Одобрен
Ж	>10	Нет	Средний	Отказ

ВЫВОД ДЕРЕВА РЕШЕНИЙ. ПРОСТЕЙШИЙ АЛГОРИТМ

Пол

М

Сумма	Авто	Доход	Кредит
<=10	Есть	Средний	Одобрен
<=10	Нет	Низкий	Отказ
>10	Есть	Низкий	Отказ
>10	Есть	Высокий	Одобрен

Ж

Сумма	Авто	Доход	Кредит
<=10	Есть	Средний	Одобрен
<=10	Нет	Низкий	Отказ
>10	Нет	Высокий	Одобрен
>10	Нет	Средний	Отказ

ВЫВОД ДЕРЕВА РЕШЕНИЙ. ПРОСТЕЙШИЙ АЛГОРИТМ

Пол

М

Сумма

<=10

Авто	Доход	Кредит
Есть	Средний	Одобрен
Нет	Низкий	Отказ

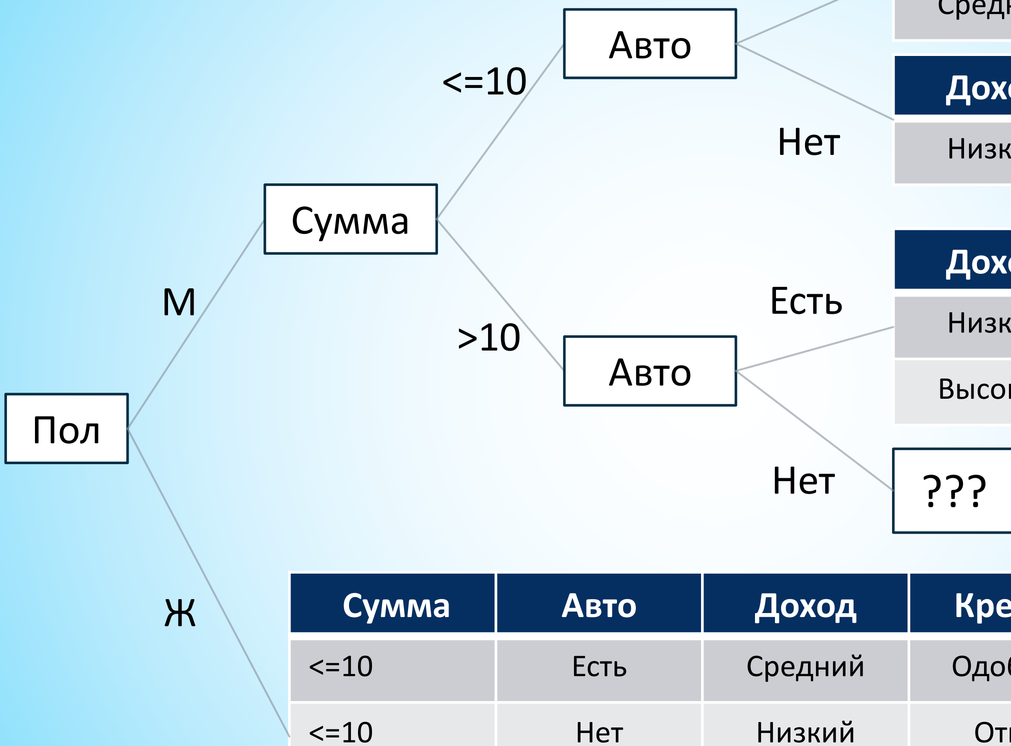
>10

Авто	Доход	Кредит
Есть	Низкий	Отказ
Есть	Высокий	Одобрен

Ж

Сумма	Авто	Доход	Кредит
<=10	Есть	Средний	Одобрен
<=10	Нет	Низкий	Отказ
>10	Нет	Высокий	Одобрен
>10	Нет	Средний	Отказ

ВЫВОД ДЕРЕВА РЕШЕНИЙ. ПРОСТЕЙШИЙ АЛГОРИТМ



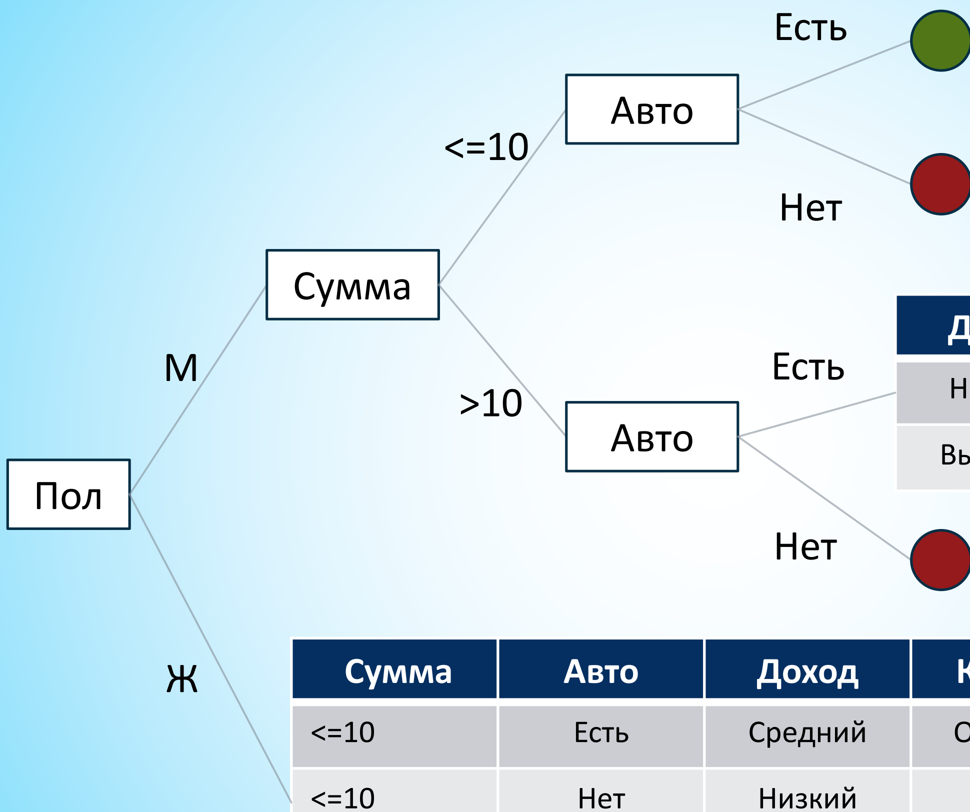
Доход	Кредит
Средний	Одобрен

Доход	Кредит
Низкий	Отказ

Доход	Кредит
Низкий	Отказ
Высокий	Одобрен

Сумма	Авто	Доход	Кредит
<=10	Есть	Средний	Одобрен
<=10	Нет	Низкий	Отказ
>10	Нет	Высокий	Одобрен
>10	Нет	Средний	Отказ

ВЫВОД ДЕРЕВА РЕШЕНИЙ. ПРОСТЕЙШИЙ АЛГОРИТМ



Доход	Кредит
Низкий	Отказ
Высокий	Одобрен

Мажоритарная классификация по родительскому узлу

Сумма	Авто	Доход	Кредит
<=10	Есть	Средний	Одобрен
<=10	Нет	Низкий	Отказ
>10	Нет	Высокий	Одобрен
>10	Нет	Средний	Отказ

ВЫВОД ДЕРЕВА РЕШЕНИЙ. ПРОСТЕЙШИЙ АЛГОРИТМ. ОБСУЖДЕНИЕ

- **Достоинства:**
  - Позволяет точно описать все примеры обучающей выборки
  - Прост в реализации
- **Недостатки:**
  - Большой размер получающегося дерева
  - Не позволяет улавливать закономерности, обобщать примеры (просто запоминание)

Поиск *минимального* дерева решений – трудноразрешимая задача, но есть полезные эвристики.

ВЫБОР АТРИБУТА

- **Цель:**
  - Минимизация глубины окончательного дерева.
- **Идея:**
  - Выбрать в первую очередь тот атрибут, который позволяет сразу выполнить максимально возможный объем работы по классификации примеров.

- Способ: оценить объём **информации**, предоставляемой атрибутом.

ЭНТРОПИЯ И ПРИРОСТ ИНФОРМАЦИИ

Пусть есть множество *A* из *n* элементов, *m* из которых обладают некоторым свойством *S*. Тогда энтропия множества *A* по отношению к свойств *S* определяется следующим образом:

$$H(A,S) = -\frac{m}{n} log_2 \frac{m}{n} - \frac{n-m}{n} log_2 \frac{n-m}{n}$$

Предположим, множество *A* классифицировано посредством атрибута *Q*, имеющего *q* возможных значений. Тогда **прирост информации** (information gain) определяется как

$$Gain(A,Q) = H(A,S) - \sum_{i=1}^{q} \frac{|A_i|}{|A|} H(A_i,S),$$

где A_i – множество элементов *A*, на которых атрибут *Q* имеет значение *i*.

Атрибут для классификации нужно выбирать так, чтобы прирост информации был максимальным.

ЭНТРОПИЯ И ПРИРОСТ ИНФОРМАЦИИ. ПРИМЕР

Для первого ветвления:

$$H(A, \text{Кредит}) = -\frac{4}{8}log_2\frac{4}{8} - \frac{4}{8}log_2\frac{4}{8} = 1$$

$$Gain(A, \text{Пол}) = 1 - \left(\frac{4}{8}H(\text{Пол} = \text{М}, \text{Кредит}) + \frac{4}{8}H(\text{Пол} = \text{Ж}, \text{Кредит})\right) =$$

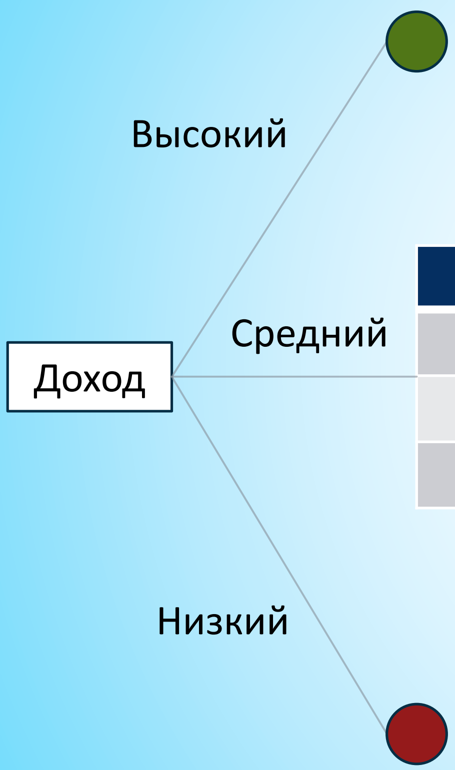
$$= 1 - \left(\frac{4}{8} * 1 + \frac{4}{8} * 1\right) = 0$$

$$Gain(A, \text{Сумма}) = 1 - \left(\frac{4}{8} * 1 + \frac{4}{8} * 1\right) = 0$$

$$Gain(A, \text{Авто}) = 1 - \left(\frac{4}{8} * 0{,}811 + \frac{4}{8} * 0{,}811\right) \approx 0.189$$

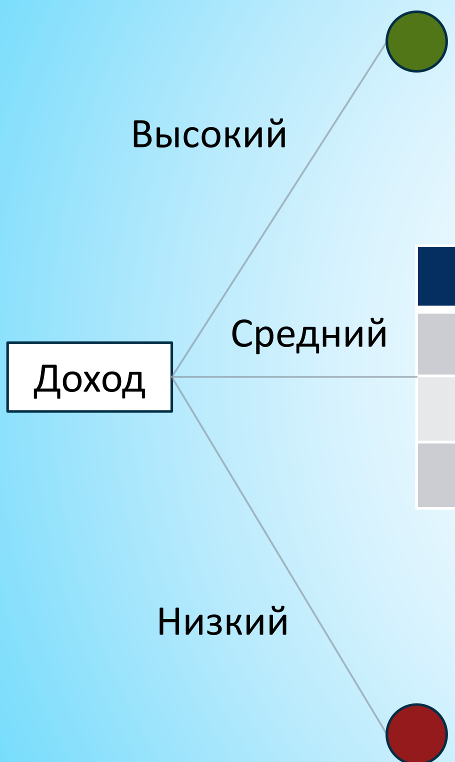
$$Gain(A, \text{Доход}) = 1 - \left(\frac{3}{8} * 0 + \frac{3}{8} * 0{,}918 + \frac{2}{8} * 0\right) \approx \mathbf{0.656}$$

СТРОИМ ПО-НОВОМУ



Пол	Сумма	Авто	Кредит
М	<=10	Есть	Одобрен
Ж	<=10	Есть	Одобрен
Ж	>10	Нет	Отказ

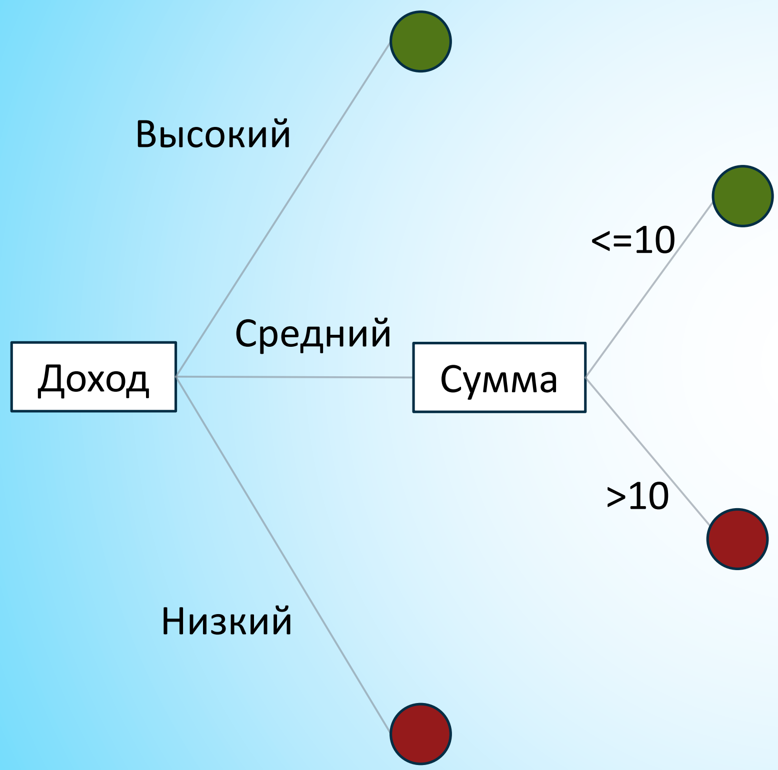
СТРОИМ ПО-НОВОМУ



0,25 **0,916** **0,916**

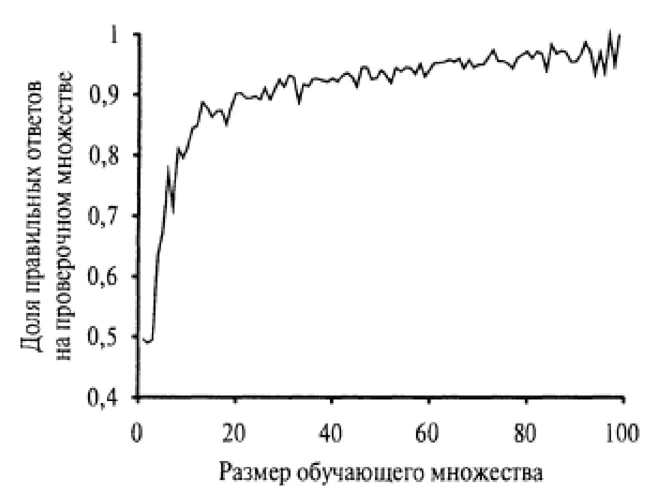
Пол	Сумма	Авто	Кредит
М	<=10	Есть	Одобрен
Ж	<=10	Есть	Одобрен
Ж	>10	Нет	Отказ

СТРОИМ ПО-НОВОМУ



ОЦЕНКА ПРОИЗВОДИТЕЛЬНОСТИ ОБУЧАЮЩЕГО АЛГОРИТМА

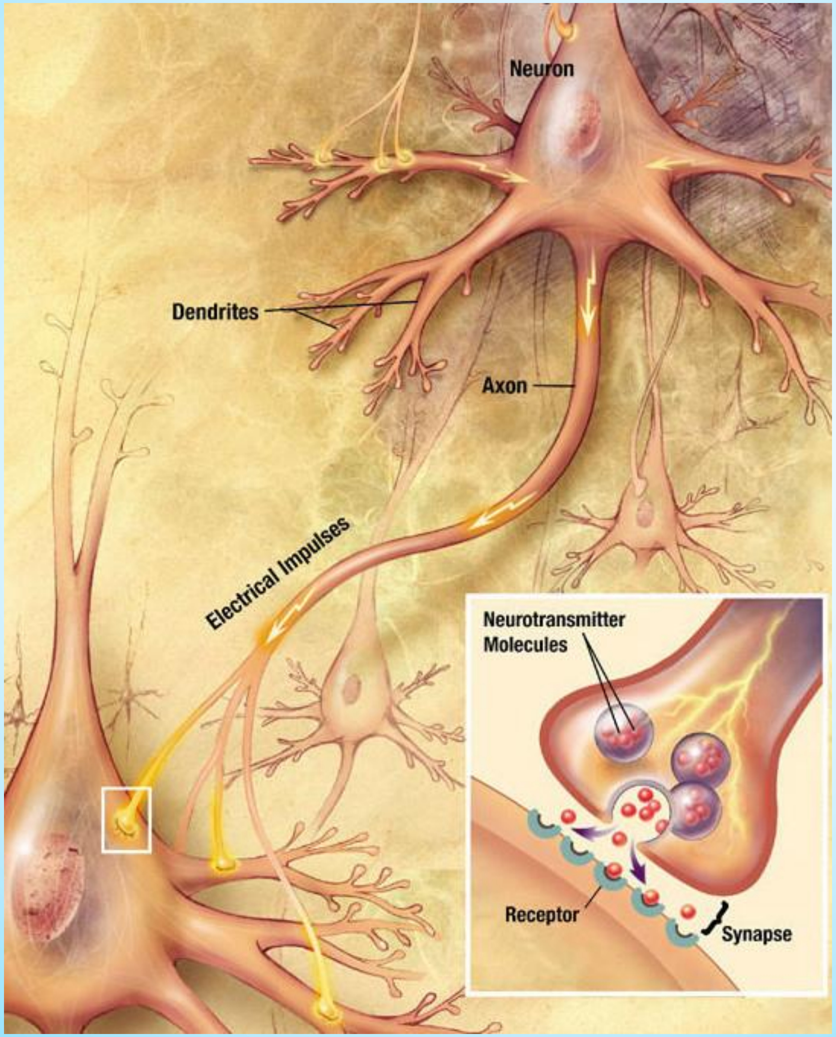
1. Собрать множество примеров большого объема.
2. Разделить его на два <u>непересекающихся</u> подмножества: **обучающее множество** и **проверочное множество**.
3. Применить обучающий алгоритм к обучающему множеству для формирования гипотезы *h*.
4. Определить, какой процент примеров в проверочном множестве правильно классифицируется с помощью гипотезы *h*.
5. Повторять этапы 2-4 для различных размеров обучающих множеств и различных случайно выбранных обучающих множеств каждого размера.



ДЕРЕВЬЯ РЕШЕНИЙ. РАЗВИТИЕ ИДЕИ

- **Чрезмерная подгонка**
    - Отсечение ветвей дерева решений (например, в алгоритме C4.5)
- **Недостающие данные**
    - Заполнение пропусков, например наиболее вероятными значениями
- **Непрерывные и целочисленные входные атрибуты**
    - Поиск точек разбиения, возможно, на тех же принципах
- **Выходные атрибуты с непрерывными значениями**
    - Дерево регрессии. В каждом листе – линейная функция от оставшихся атрибутов, которая сопоставляется с данными линейной регрессией
- **Ансамбль деревьев**
    - Random forest

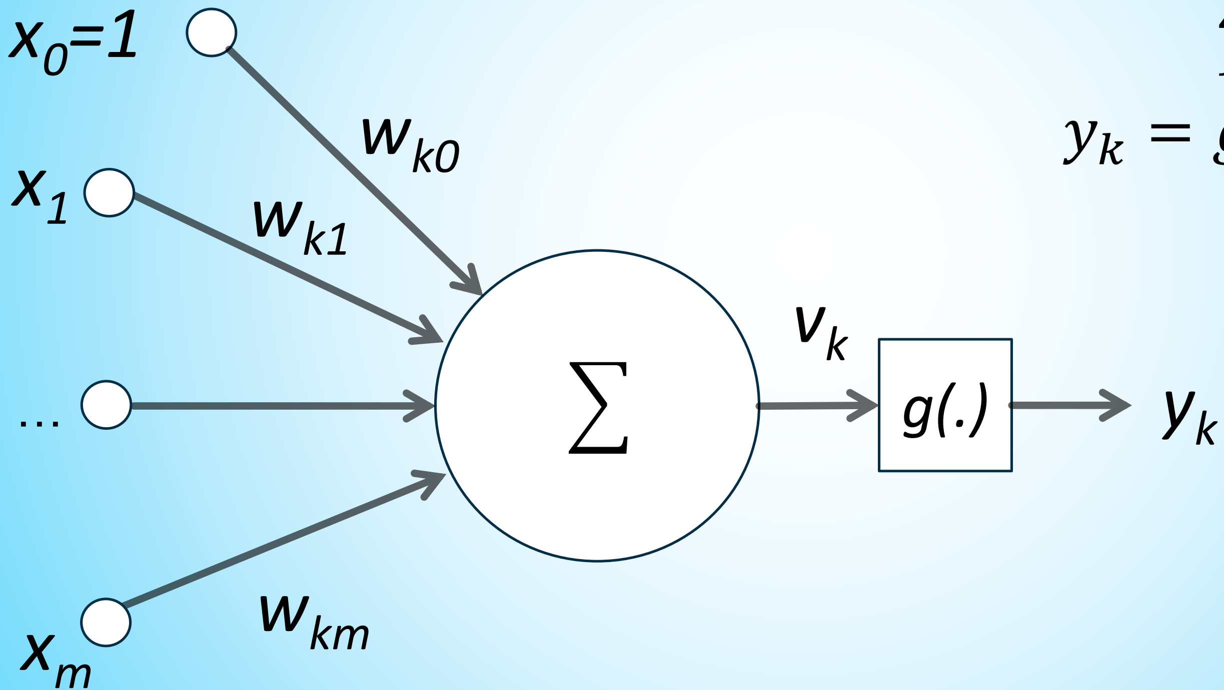
ИСКУССТВЕННЫЕ НЕЙРОННЫЕ СЕТИ



МОДЕЛЬ НЕЙРОНА

$$v_k = \sum_{j=0}^{m} w_{kj} x_j$$

$$y_k = g(v_k)$$



ФУНКЦИИ АКТИВАЦИИ

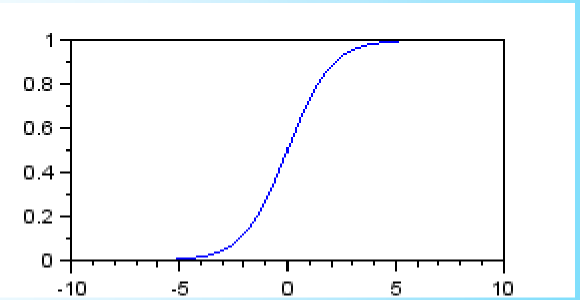
Задача функции активации – осуществление сжимающего отображения взвешенного суммарного сигнала.

1) Пороговая функция (функция Хэвисайда):

$$g(v) = \begin{cases} 1, v \geq 0; \\ 0, v < 0. \end{cases}$$

2) Сигмоидальная функция:

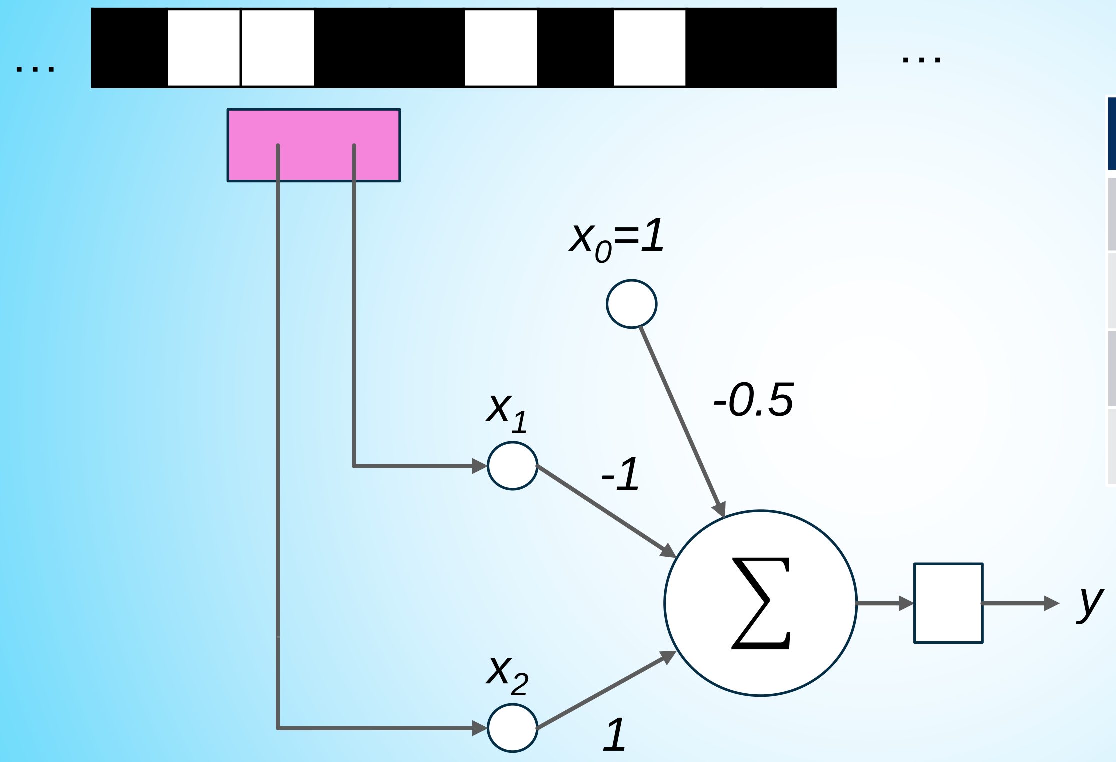
$$g(v) = \frac{1}{1 + e^{-av}}$$



3) Кусочно-линейная функция:

$$g(v) = \begin{cases} 0, v \leq -\frac{1}{2}; \\ \frac{1}{2} + v, -\frac{1}{2} < v < \frac{1}{2}; \\ 1, v > \frac{1}{2}. \end{cases}$$

ПРИМЕР КЛАССИФИКАЦИИ С ИСПОЛЬЗОВАНИЕМ ОДНОГО НЕЙРОНА

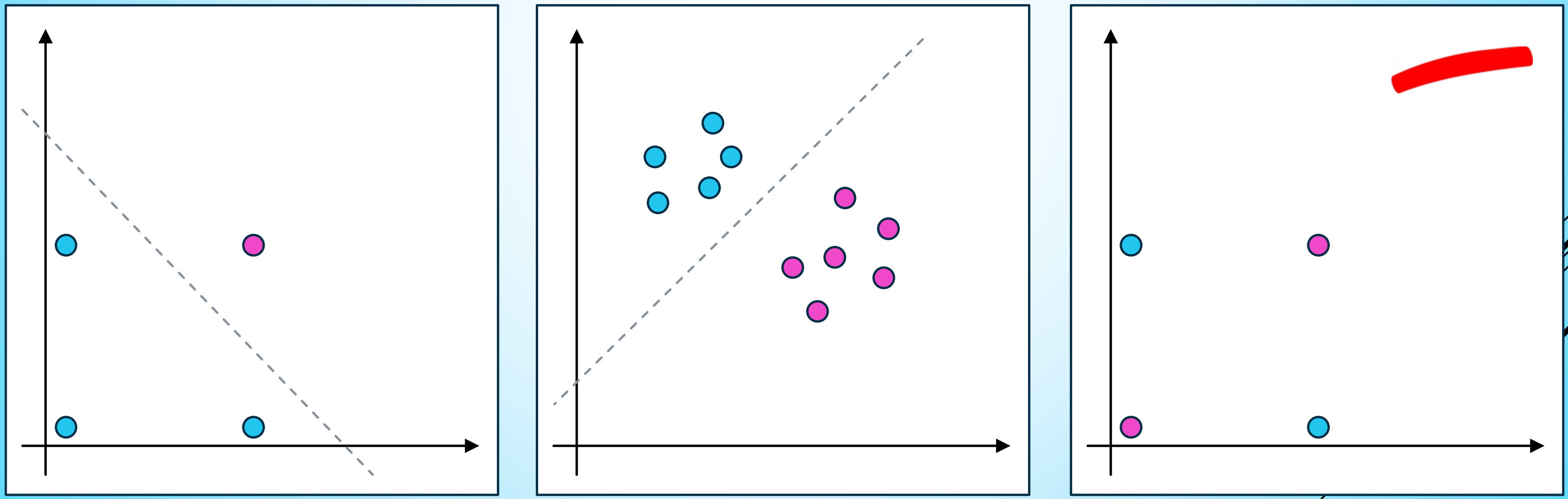


x_1	x_2	y
0	0	0
0	1	1
1	0	0
1	1	0

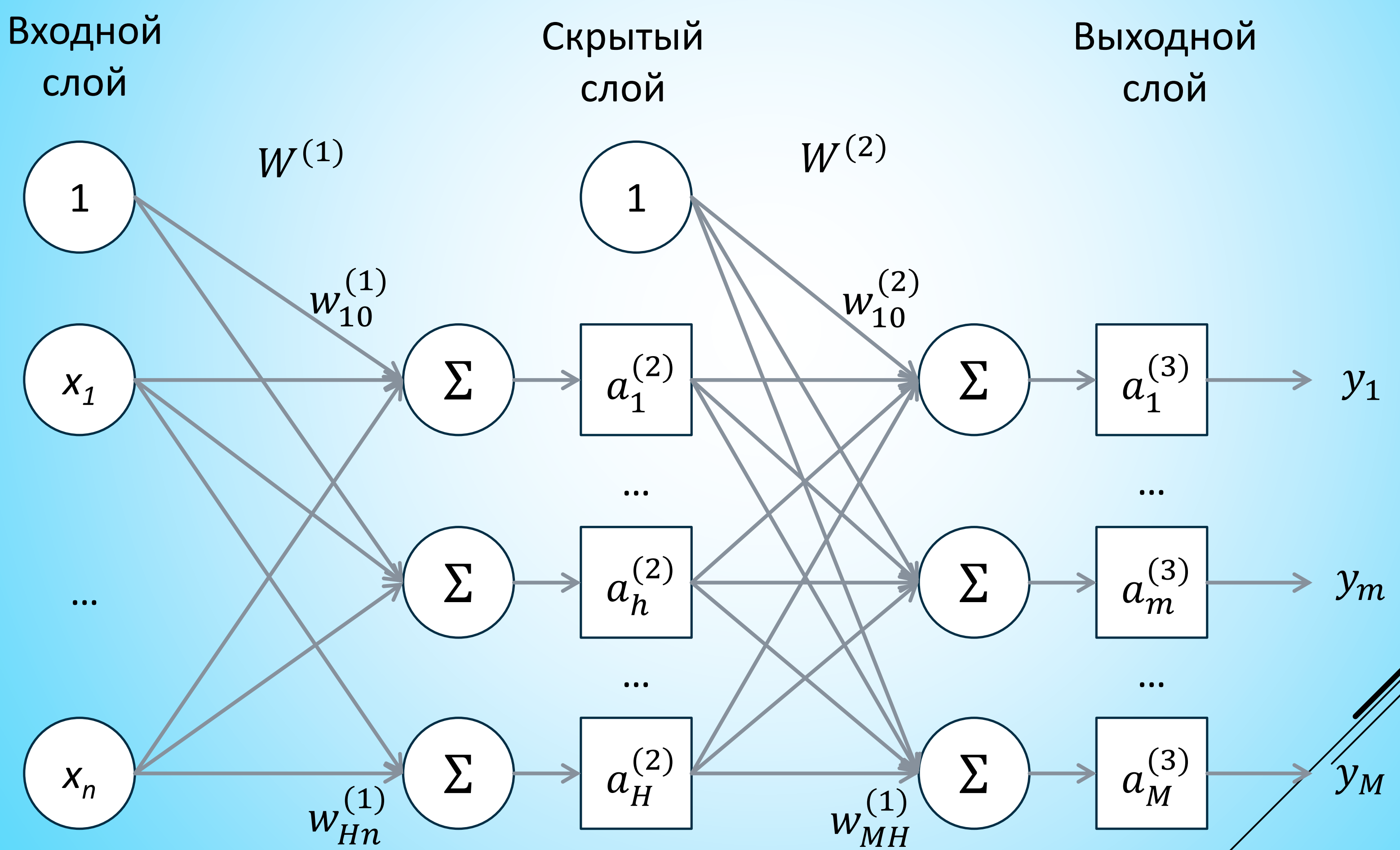
ВЫЧИСЛИТЕЛЬНЫЕ ВОЗМОЖНОСТИ НЕЙРОНОВ

$$y = \begin{cases} 1, \sum_i w_{ki}\, x_i > \theta; \\ 0, \sum_i w_{ki}\, x_i \le \theta. \end{cases}$$

Линейная разделимость



МНОГОСЛОЙНАЯ НЕЙРОННАЯ СЕТЬ



АЛГОРИТМ ПРЯМОГО РАСПРОСТРАНЕНИЯ

Исходные данные:

1) нейронная сеть с *L* слоями; синаптические веса нейронной сети заданы матрицами $W^{(1)}, W^{(2)}, \dots, W^{(L-1)}$; функция активации *g(x)*.

2) входной вектор $x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$.

Алгоритм:

1) $a^{(1)} = x$.

2) Для всех $i \epsilon \{1, .., L-1\}$ выполнить шаг 3.

3) $a^{(i+1)} = g\left(W^{(i)} \begin{pmatrix} 1 \\ a^{(i)} \end{pmatrix}\right)$.

4) $y = a^{(L)}$.

ВЫЧИСЛИТЕЛЬНЫЕ ВОЗМОЖНОСТИ МНОГОСЛОЙНОЙ НЕЙРОННОЙ СЕТИ

Любую непрерывную функцию нескольких переменных **можно с любой точностью** реализовать с помощью **двухслойной нейронной сети** с достаточным количеством нейронов в скрытом слое.

Рекомендации по выбору архитектуры сети:

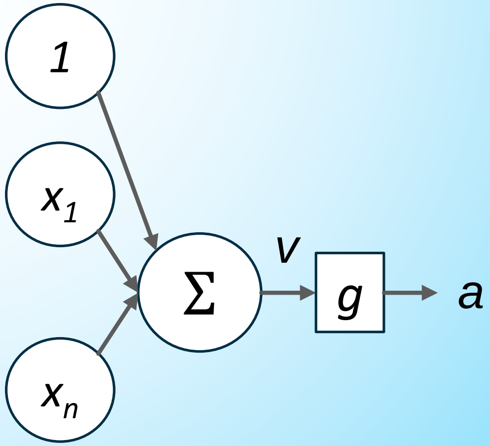
- Очень часто бывает достаточно всего одного скрытого слоя.
- Количество нейронов в скрытом слое обычно имеет порядок количества входов (n, $2n$, $3n$, $4n$).
- Если слоёв несколько, то, как правило, они содержат одинаковое количество нейронов.

ЗАДАЧА ОБУЧЕНИЯ ОДНОГО НЕЙРОНА (1)

Дано:
1) Обучающее множество $T = \{(X^{(1)}, y^{(1)}), \dots, (X^{(m)}, y^{(m)})\}$.
2) $X^{(i)} \in \mathbb{R}^n, y^{(i)} \in \{0,1\}$ (бинарная классификация).

Требуется:
Найти синаптические веса нейрона $w \in \mathbb{R}^{n+1}$, осуществляющего классификацию обучающего множества *наилучшим образом*.

Допущение:
Функция активации – логистическая кривая.



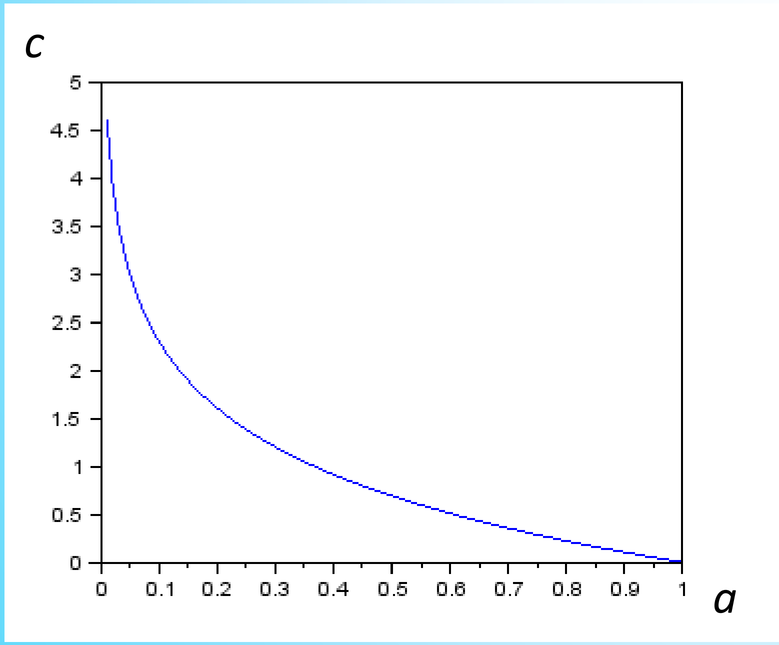
Задача *параметрического обучения*.

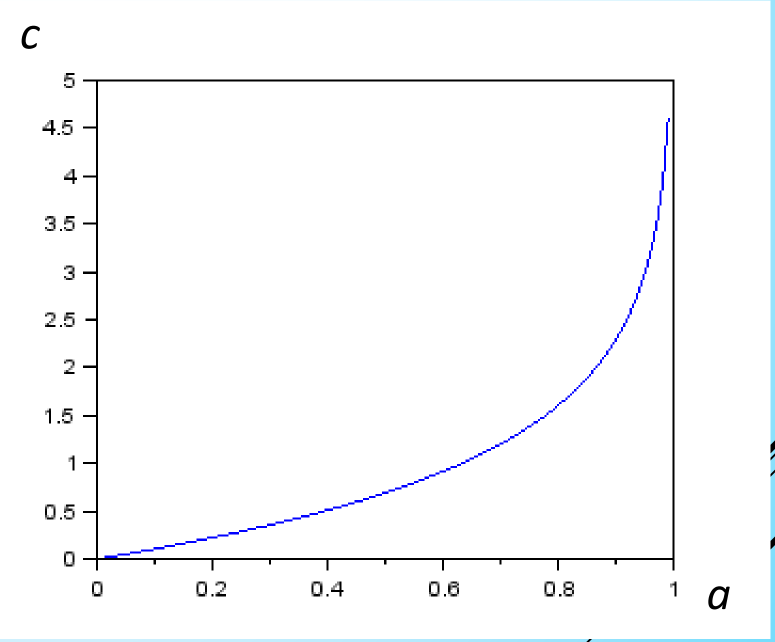
(Та же сама логистическая регрессия)

ЗАДАЧА ОБУЧЕНИЯ ОДНОГО НЕЙРОНА (2)

Пусть *c(a,y)* – функция стоимости, «штраф», накладываемый на выходное значение *a* при обработке примера, имеющего «эталонный» ответ *y*.

$$c(a,y) = \begin{cases} -\log(a)\,,\text{если } y = 1 \\ -\log(1-a)\,,\text{если } y = 0 \end{cases}$$





ЗАДАЧА ОБУЧЕНИЯ ОДНОГО НЕЙРОНА (3)

$$c(a,y) = \begin{cases} -\log(a)\text{, если } y = 1 \\ -\log(1-a)\text{, если } y = 0 \end{cases}$$

$$c(a,y) = -y\log(a) - (1-y)\log(1-a)$$

$$J(w) = \frac{1}{m}\sum_{i=1}^{m}\left(c\left(a^{(i)}, y^{(i)}\right)\right) =$$

$$= -\frac{1}{m}\left[\sum_{i=1}^{m}\left(y^{(i)}\log\left(a^{(i)}\right) + \left(1 - y^{(i)}\right)\log(1 - a^{(i)})\right)\right]$$

ЗАДАЧА ОБУЧЕНИЯ ОДНОГО НЕЙРОНА (4)

$$\min J(w)$$

$$J(w) = -\frac{1}{m}\left[\sum_{i=1}^{m}\left(y^{(i)}\log\left(g_w\left(x^{(i)}\right)\right) + \left(1 - y^{(i)}\right)\log(1 - g_w\left(x^{(i)}\right))\right)\right]$$

$$\frac{\partial J(w)}{\partial w_j} = \frac{1}{m}\sum_{i=1}^{m}\left(g_w\left(x^{(i)}\right) - y^{(i)}\right)x_j^{(i)}$$

Градиентный спуск (например):

$w = w_0$

while (not <условие останова>) **do**

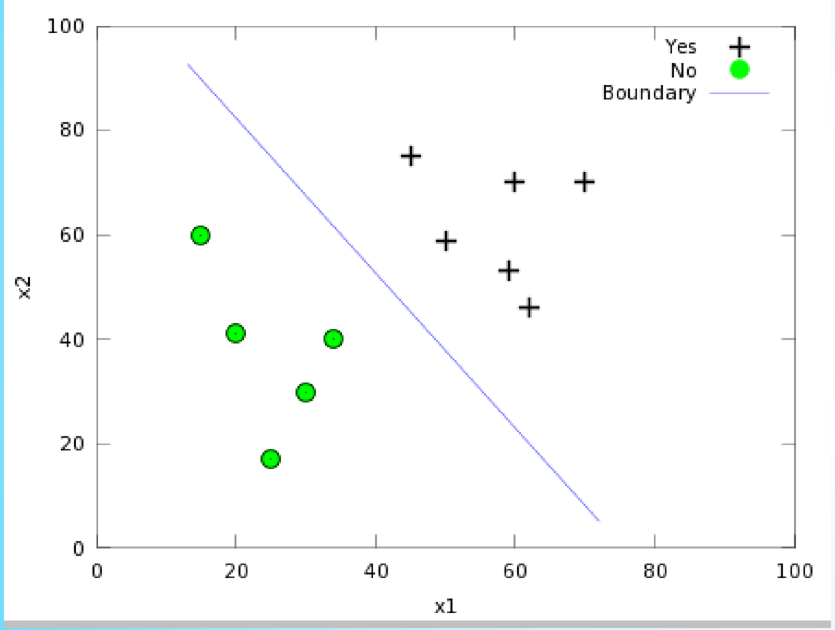
  $nw = w$

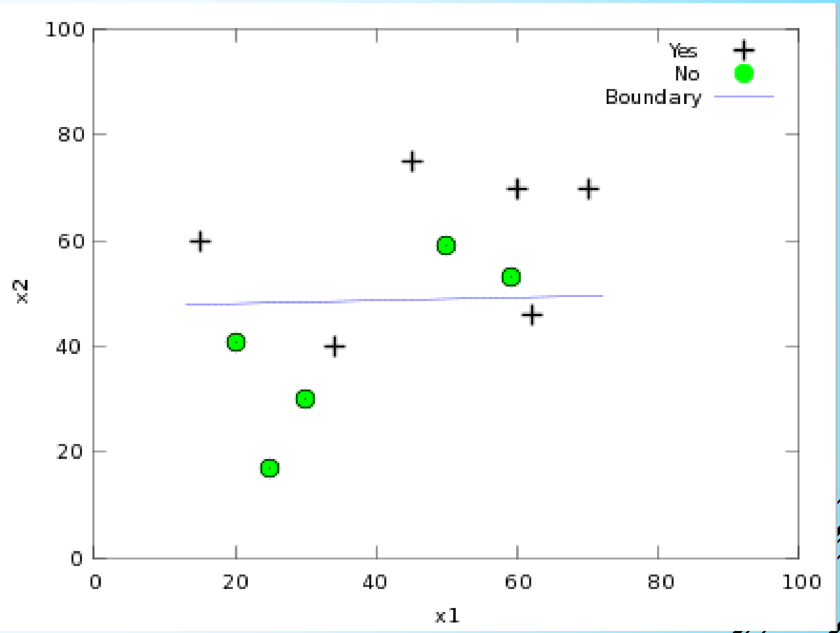
  for j = 0 .. n **do**

    $nw_j = nw_j - \alpha\frac{1}{m}\sum_{i=1}^{m}\left(g_w\left(x^{(i)}\right) - y^{(i)}\right)x_j^{(i)}$

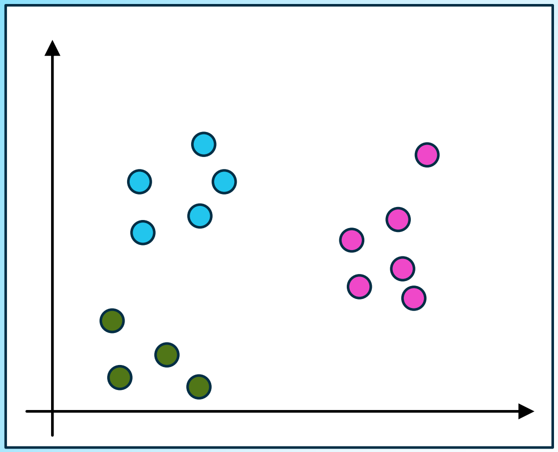
  $w = nw$

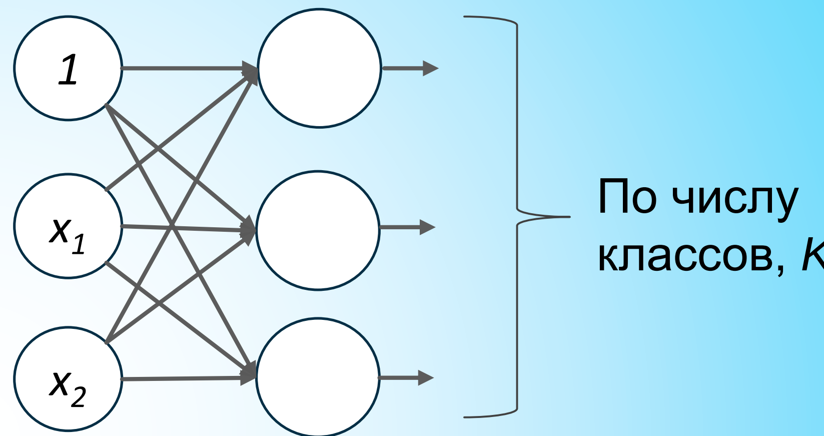
ЗАДАЧА ОБУЧЕНИЯ ОДНОГО НЕЙРОНА. ПРИМЕР

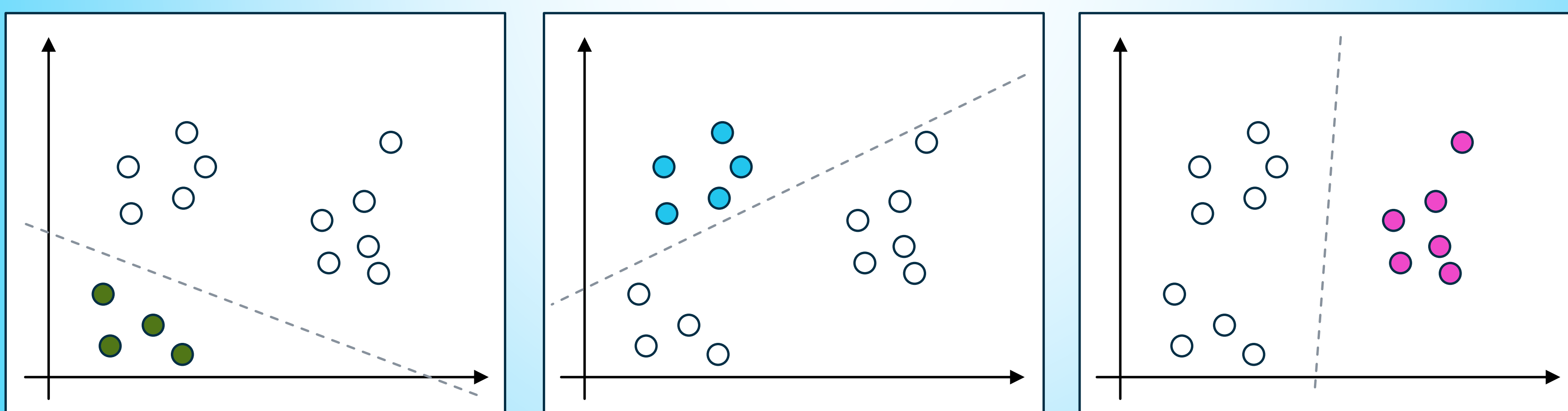




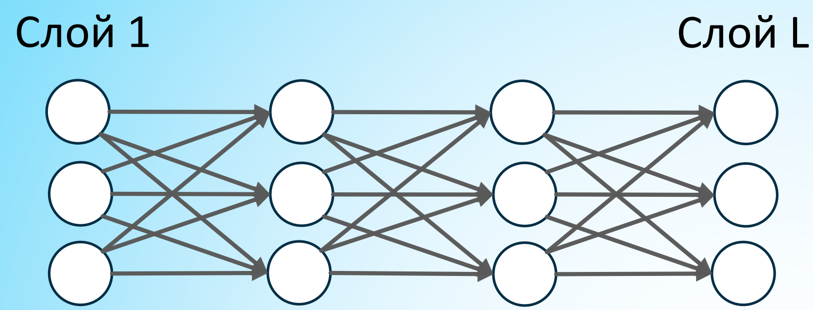
K-АРНЫЙ КЛАССИФИКАТОР







ОБУЧЕНИЕ НЕЙРОННОЙ СЕТИ



Сеть с L слоями.
s_l – количество нейронов в l-том слое (без фиктивных).

Дано:

1) Обучающее множество $T = \{(x^{(1)}, y^{(1)}), \dots, (x^{(m)}, y^{(m)})\}$.
2) $x^{(i)} \in \mathbb{R}^n, y^{(i)} \in \mathbb{R}^K$.

Требуется:

Найти синаптические веса нейронов $W^{(1)}, W^{(2)}, \dots, W^{(L-1)}$ сети аппроксимирующей *наилучшим образом* зависимость, представленную обучающим множеством.

Допущение:

Функция активации – логистическая кривая.

ОБУЧЕНИЕ НЕЙРОННОЙ СЕТИ. АЛГОРИТМ ОБРАТНОГО РАСПРОСТРАНЕНИЯ ОШИБКИ (1)

Для одного нейрона:

$$J(w) = -\frac{1}{m}\left[\sum_{i=1}^{m}\left(y^{(i)}\log\left(g_w\left(x^{(i)}\right)\right) + \left(1 - y^{(i)}\right)\log\left(1 - g_w\left(x^{(i)}\right)\right)\right)\right]$$

Для сети:

$$J(w) = -\frac{1}{m}\left[\sum_{i=1}^{m}\sum_{k=1}^{K}\left(y_k^{(i)}\log\left(g_{W,k}\left(x^{(i)}\right)\right) + \left(1 - y^{(i)}\right)\log\left(1 - g_{W,k}\left(x^{(i)}\right)\right)\right)\right]$$

ОБУЧЕНИЕ НЕЙРОННОЙ СЕТИ. АЛГОРИТМ ОБРАТНОГО РАСПРОСТРАНЕНИЯ ОШИБКИ (2)

(Вычисление градиента функции стоимости)

Рассмотрим *один элемент обучающего множества*. Выполним для него прямое распространение.

Пусть $\delta_j^{(l)}$- ошибка элемента *j* в слое *l*.

Для элементов выходного слоя нейронной сети (*L*=4):

$\delta_j^{(4)} = a_j^{(4)} - y_j$ (или в векторной форме $\delta^{(4)} = a^{(4)} - y$)

$\delta^{(3)} = \left(W^{(3)}\right)^T \delta^{(4)} \circ g'(v^{(3)})$

$\delta^{(2)} = \left(W^{(2)}\right)^T \delta^{(3)} \circ g'(v^{(2)})$

$\delta^{(1)}$ по понятным причинам нет

$$\boxed{\frac{\partial J(W)}{\partial W_{ij}^{(l)}} = a_j^{(l)} \delta_i^{(l+1)}}$$

ОБУЧЕНИЕ НЕЙРОННОЙ СЕТИ. АЛГОРИТМ ОБРАТНОГО РАСПРОСТРАНЕНИЯ ОШИБКИ (3)

Схема алгоритма:

repeat

$\Delta_{ij}^{(l)} = 0$ (для всех *i*,*j*,*l*). *(накопители для вычисления градиента)*

for *i* = 1 **to** *m* *(для каждого примера)*

$a^{(1)} = x^{(i)}$

Выполнить прямое распространение по сети, вычислив $a^{(2)}, \ldots, a^{(L)}$

$\delta^{(L)} = a^{(L)} - y^{(m)}$

Вычислить $\delta^{(L-1)}, \ldots, \delta^{(2)}$

$\Delta_{ij}^{(l)} = \Delta_{ij}^{(l)} + a_j^{(l)} \delta_i^{(l+1)}$

$D_{ij}^{(l)} = \frac{1}{m} \Delta_{ij}^{(l)}$

$W_{ij}^{(l)} = W_{ij}^{(l)} - \alpha D_{ij}^{(l)}$

until достигнут критерий останова

НЕКОТОРЫЕ ОБЩИЕ ПРОБЛЕМЫ И ВОЗМОЖНЫЕ РЕШЕНИЯ

Поиск баланса между точностью описания имеющихся данных и приемлемым качеством предсказания (т.е., обобщением имеющихся данных).

Общие методы:

- выделение в имеющемся множестве примеров обучающего и проверочного множеств, перекрёстная проверка (n-fold);
- умышленное «огрубление» результатов (в деревьях решений – отсечения, в нейронных сетях – алгоритм *прореживания сети (optimal brain damage)*, регуляризация).

ЛИТЕРАТУРА

Общие вопросы создания обучаемых интеллектуальных систем:

1) Рассел С., Норвиг П. Искусственный интеллект. Современный подход. 2-е издание (Часть 6).

Деревья решений:

1) Рассел С., Норвиг П. Искусственный интеллект. Современный подход. 2-е издание (Гл. 18).
2) Quinlan J. Ross C4.5: Programs for Machine learning. Morgan Kaufmann Publishers 1993.

Нейронные сети:

1) Хайкин С. Нейронные сети: полный курс.

ОРГАНИЗАЦИОННОЕ

На **следующей лекции (28 ноября)** состоится **тест**, по материалам **первых 4 лекций**.