

ОДНОСЛОЙНАЯ НЕЙРОННАЯ СЕТЬ

НЕЙРОННАЯ СЕТЬ И ЗАДАЧА РЕГРЕССИИ

- › Нейронная сеть — универсальная модель, решающая широкий спектр задач
- › Рассмотрим выборку:
$$(x_i, y_i), \quad x_i \in \mathbb{R}^d, \quad y_i \in \mathbb{R}^1, \quad i = 1, \dots, \ell$$
- › x — описание объекта, вектор из d элементов — признаков x_j
- › y — зависимая переменная

НЕЙРОННАЯ СЕТЬ И ЗАДАЧА РЕГРЕССИИ

› Рассмотрим выборку:

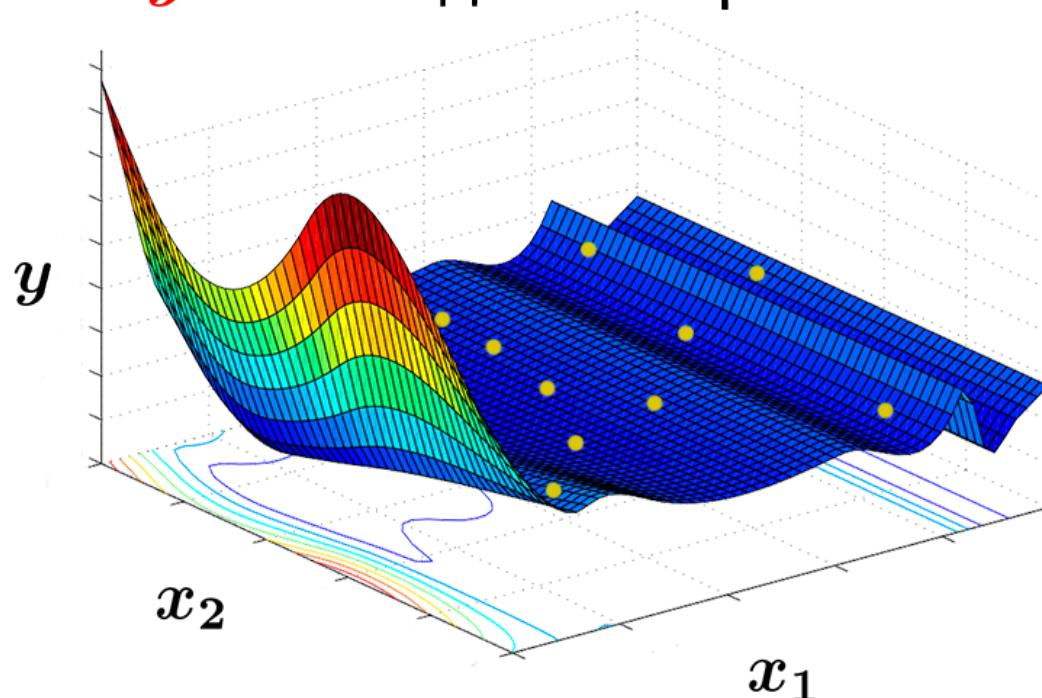
$$(x_i, y_i), \quad x_i \in \mathbb{R}^d, \quad y_i \in \mathbb{R}^1, \quad i = 1, \dots, \ell$$

- › x — описание объекта, вектор из d элементов — признаков x_j
- › y — зависимая переменная
- › Требуется построить аппроксимирующую поверхность $a(x)$

НЕЙРОННАЯ СЕТЬ И ЗАДАЧА РЕГРЕССИИ

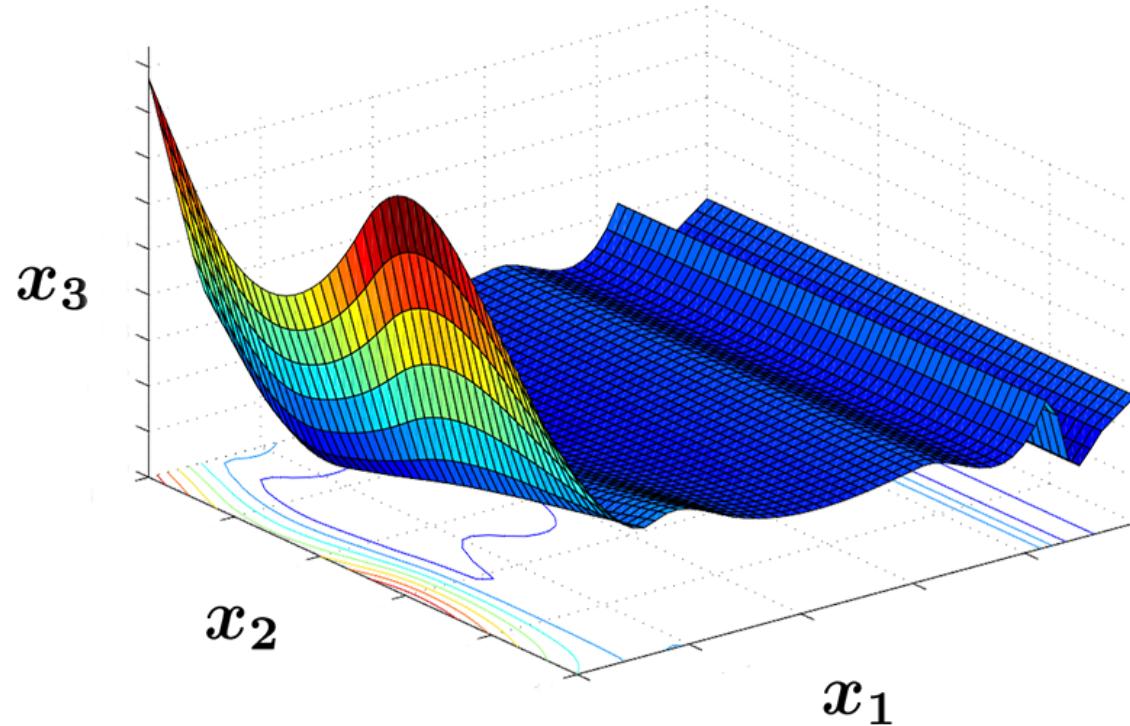
› Задача оценки риска в страховании и финансах:

- ▶ x_1 — время до страхового случая
- ▶ x_2 — цена страховки
- ▶ y — ожидаемый риск



НЕЙРОННАЯ СЕТЬ И ЗАДАЧА РЕГРЕССИИ

- › Задача оценки риска в страховании и финансах:



- › Поверхность $a(\mathbf{x})$ аппроксимирует исторический риск y_i в точках \mathbf{x}_i

НЕЙРОННАЯ СЕТЬ И ЗАДАЧА РЕГРЕССИИ

- › Примеры задач: $a(x) \mapsto y$
- › x — вектор исторических цен электроэнергии
- › y — цена электроэнергии в следующий час

НЕЙРОННАЯ СЕТЬ И ЗАДАЧА РЕГРЕССИИ

- › Примеры задач: $a(x) \mapsto y$
- › x — векторизированный снимок поверхности земли со спутника
- › y — объём зелёных насаждений

НЕЙРОННАЯ СЕТЬ И ЗАДАЧА РЕГРЕССИИ

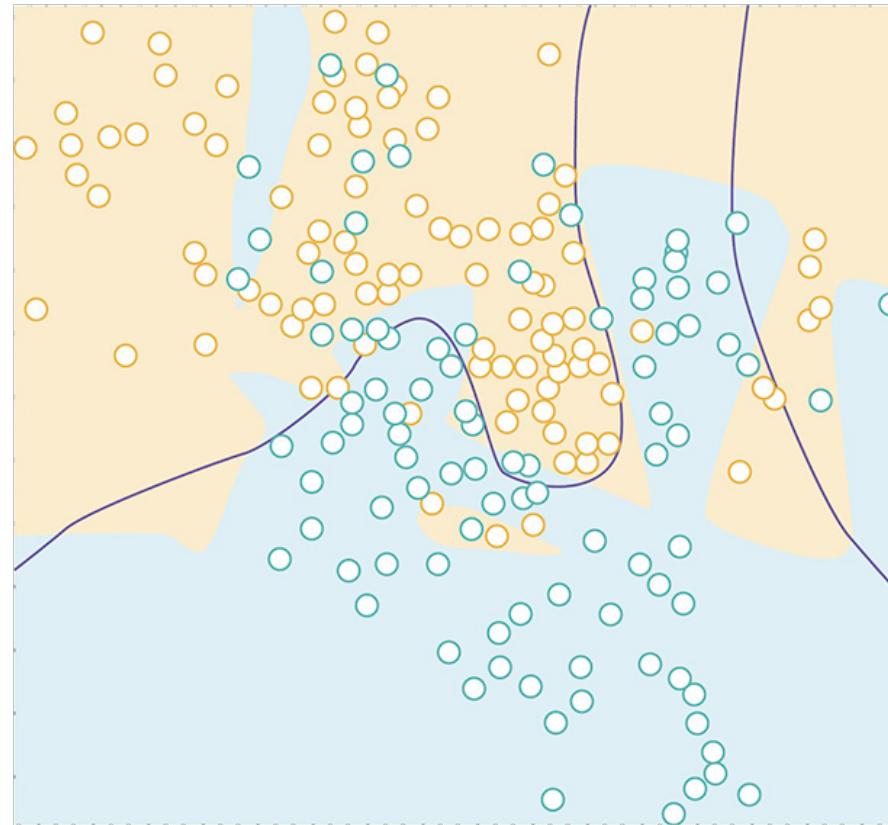
- › Примеры задач: $a(x) \mapsto y$
- › x — история продаж товара
- › y — уровень потребительского спроса

НЕЙРОННАЯ СЕТЬ И ЗАДАЧА КЛАССИФИКАЦИИ

- › Требуется построить разделяющую поверхность: $a(x) \mapsto y \in \{-1, +1\}$
- › Значения этой функции:
 - ▶ $a(x) > 0$, если $y = 1$
 - ▶ $a(x) \leq 0$, если $y = -1$

НЕЙРОННАЯ СЕТЬ И ЗАДАЧА КЛАССИФИКАЦИИ

- › Требуется построить разделяющую поверхность: $a(x) \mapsto y \in \{-1, +1\}$



НЕЙРОННАЯ СЕТЬ И ЗАДАЧА КЛАССИФИКАЦИИ

- › Примеры задач: $a(x) \mapsto y$
- › x — временной ряд акселерометра мобильного телефона
- › y — вид физической активности

НЕЙРОННАЯ СЕТЬ И ЗАДАЧА КЛАССИФИКАЦИИ

- › Примеры задач: $a(x) \mapsto y$
- › x — страница документа
- › y — нужно ли показать документ по
поисковому запросу

НЕЙРОННАЯ СЕТЬ И ЗАДАЧА КЛАССИФИКАЦИИ

- › Примеры задач: $a(x) \mapsto y$
- › x — нотная запись музыкального произведения
- › y — следующая нота

НЕЙРОННАЯ СЕТЬ КАК УНИВЕРСАЛЬНАЯ МОДЕЛЬ

- › Теорема (А.Н.Колмогоров, 1957)
- › Каждая непрерывная функция $a(x)$, заданная на единичном кубе d -мерного пространства, представима в виде:

$$a(x) = \sum_{i=1}^{2d+1} \sigma_i \left(\sum_{j=1}^d f_{ij}(x_j) \right),$$

где $x = [x_1, \dots, x_d]^T$, функции $\sigma_i(\cdot)$, $f_{ij}(\cdot)$ непрерывны, причем f_{ij} не зависят от выбора a

НЕЙРОННАЯ СЕТЬ КАК УНИВЕРСАЛЬНАЯ МОДЕЛЬ

› Иначе:

функцию от d аргументов можно представить в виде комбинации $d(2d + 1)$ функций от одного аргумента

НЕЙРОННАЯ СЕТЬ КАК УНИВЕРСАЛЬНАЯ МОДЕЛЬ

- › Единичный куб d -мерного пространства включает элементы выборки (\mathbf{x}_i, y_i) , $i = 1, \dots, \ell$, которые аппроксимирует функция $a(\mathbf{x})$
- › Если измерения \mathbf{x}_j — элементы вектора \mathbf{x} сделаны в разных шкалах (килограммы, амперы, секунды), их следует обезразмерить
- › Например, отобразить измерения каждого из d признаков в отрезок $[0, 1]$

НЕЙРОННАЯ СЕТЬ КАК УНИВЕРСАЛЬНАЯ МОДЕЛЬ

- › Важно: Колмогоров не указал, какими именно должны быть функции σ_i, f_{ij}

ОДНОСЛОЙНАЯ НЕЙРОННАЯ СЕТЬ КАК НЕЙРОН

- › Однослойная нейронная сеть (или нейрон) — это комбинация:

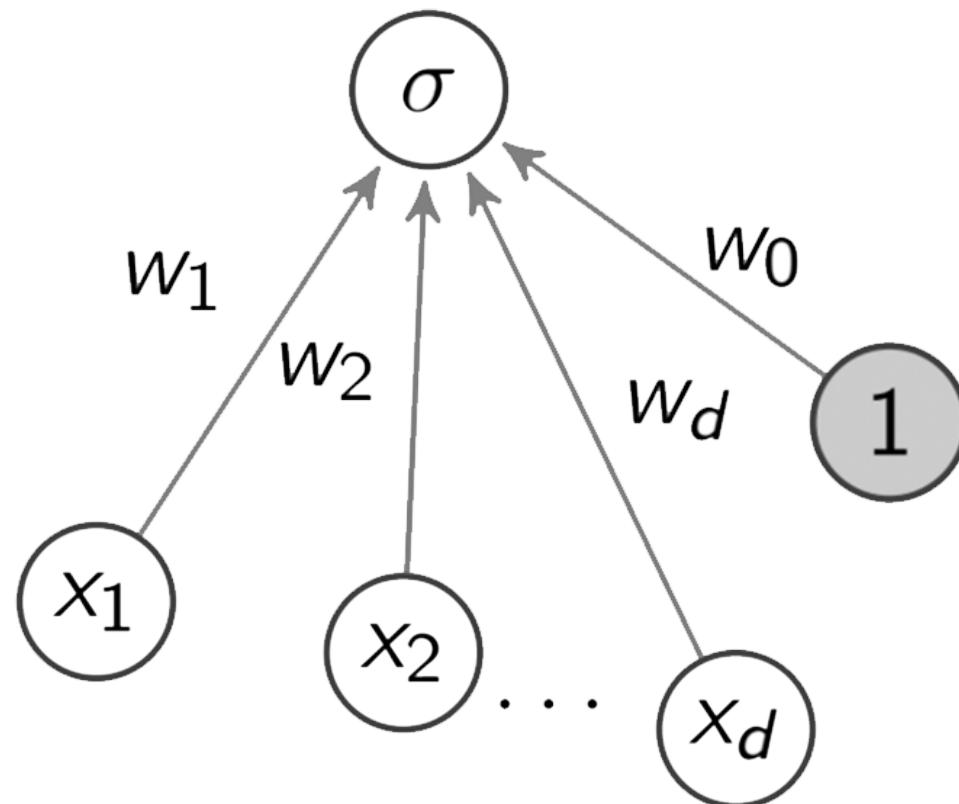
$$a(x, w) = \sigma(w^T x) = \sigma\left(\sum_{j=1}^d w_j^{(1)} x_j + w_0^{(1)}\right)$$

- › σ — функция активации, непрерывная монотонная функция, желательно, дифференцируемая

- › w — вектор параметров (весов)

- › x — объект, вектор с присоединенным элементом 1 для веса w_0

ОДНОСЛОЙНАЯ НЕЙРОННАЯ СЕТЬ КАК НЕЙРОН

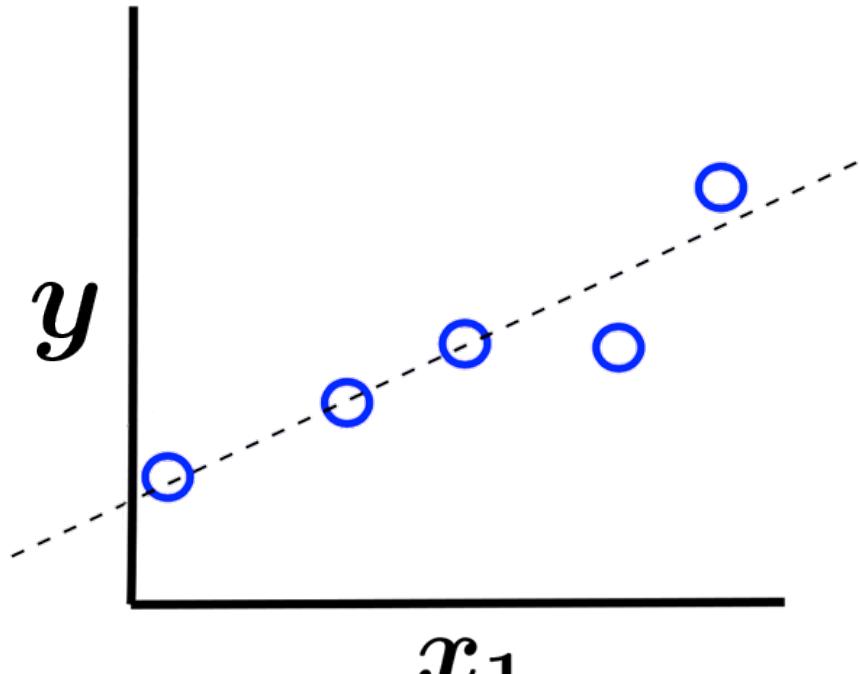


ОДНОСЛОЙНАЯ НЕЙРОННАЯ СЕТЬ — ЛИНЕЙНАЯ МОДЕЛЬ

- › Сеть с линейной функцией активации, задача восстановления регрессии x на y :

$$a(x, w) = w^T x$$

- › Функция активации: $\sigma = id$

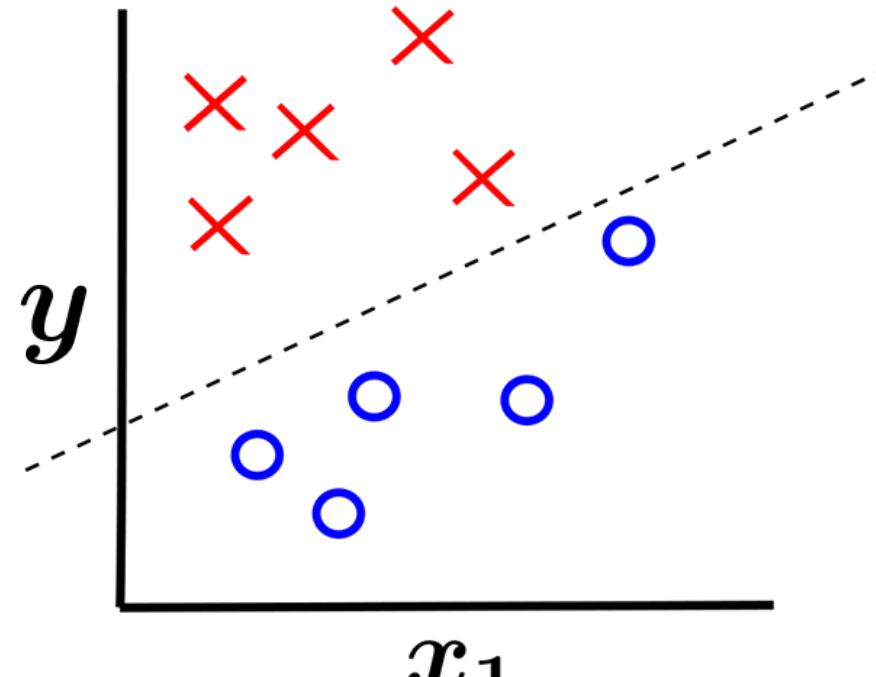


ОДНОСЛОЙНАЯ НЕЙРОННАЯ СЕТЬ — ЛИНЕЙНАЯ МОДЕЛЬ

- › Сеть с пороговой функцией активации, задача восстановления регрессии x на y :

$$a(x, w) = \text{sign}(w^T x)$$

- › Функция активации: $\sigma = \text{sign}(\cdot)$

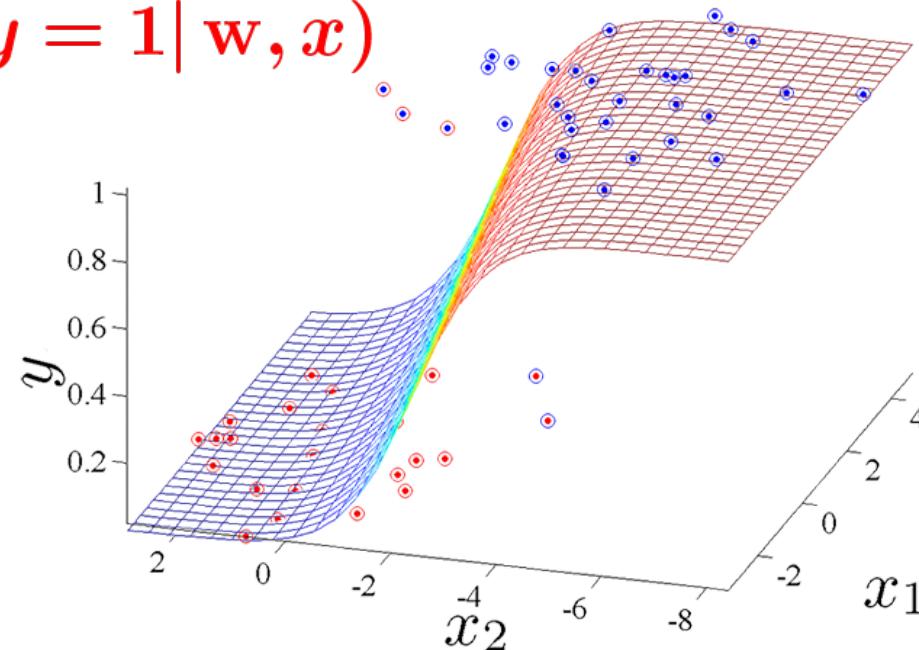


ОДНОСЛОЙНАЯ СЕТЬ – МОДЕЛЬ ЛОГИСТИЧЕСКОЙ РЕГРЕССИИ

- » Сигмоидная функция активации:

$$a(x, w) = \sigma(w^T x) = \frac{1}{1+exp(-w^T x)}$$

- » $\sigma(w^T x) = P(y = 1 | w, x)$



МНОГОКЛАССОВАЯ КЛАССИФИКАЦИЯ

- › Обобщение для многоклассового случая $y = [y^1, \dots, y^K]^T$:

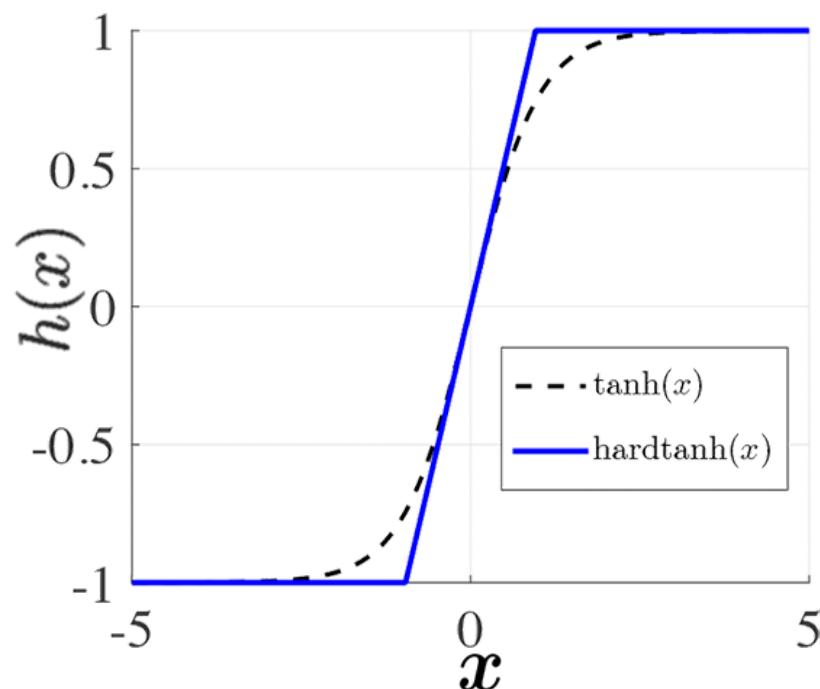
$$\sigma = \text{softmax}(\mathbf{w}_1^T \mathbf{x}, \dots, \mathbf{w}_K^T \mathbf{x}) = \frac{\exp(\mathbf{w}_k^T \mathbf{x})}{\sum_{i=1}^K \exp(\mathbf{w}_i^T \mathbf{x})}$$

- › Сеть состоит из K нейронов, и вычисляет вероятности принадлежности объекта \mathbf{x} к различным K классам одновременно

ВИДЫ ФУНКЦИЙ АКТИВАЦИИ

› Гиперболический тангенс:

$$\tanh(x) = \frac{\exp(2x)-1}{\exp(2x)+1}$$

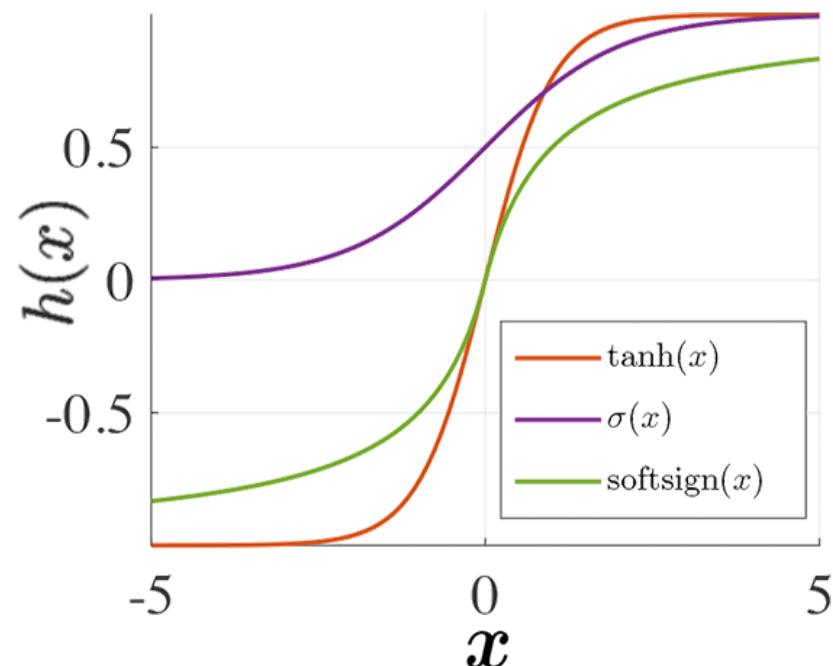


ВИДЫ ФУНКЦИЙ АКТИВАЦИИ

» Функция активации $\sigma(x)$:

$$\text{softsign}(f) = \frac{x}{1+|x|}$$

» Сходится к $+1$ или -1 медленнее, чем $\tanh(x)$

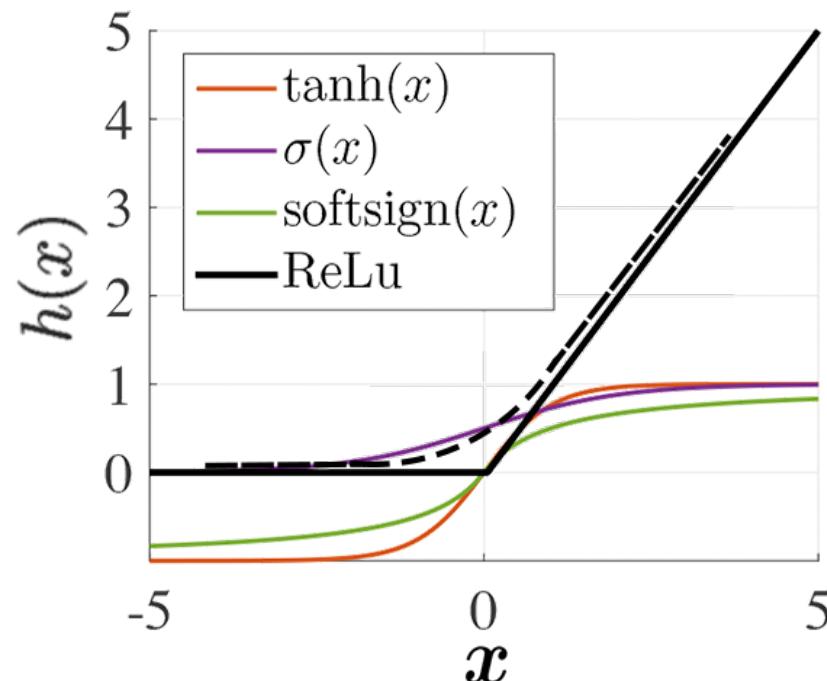


ВИДЫ ФУНКЦИЙ АКТИВАЦИИ

› Линейный выпрямитель, Rectified linear unit:

$$\text{ReLU}(f) = \max(0, f)$$

› И его приближение: $\ln(1 + \exp(f))$



РЕЗЮМЕ

- › Нейронная сеть предназначена для решения широкого класса прикладных задач регрессии и классификации (обучение с учителем)
- › Она является универсальной моделью, так как может приближать функции любой сложности

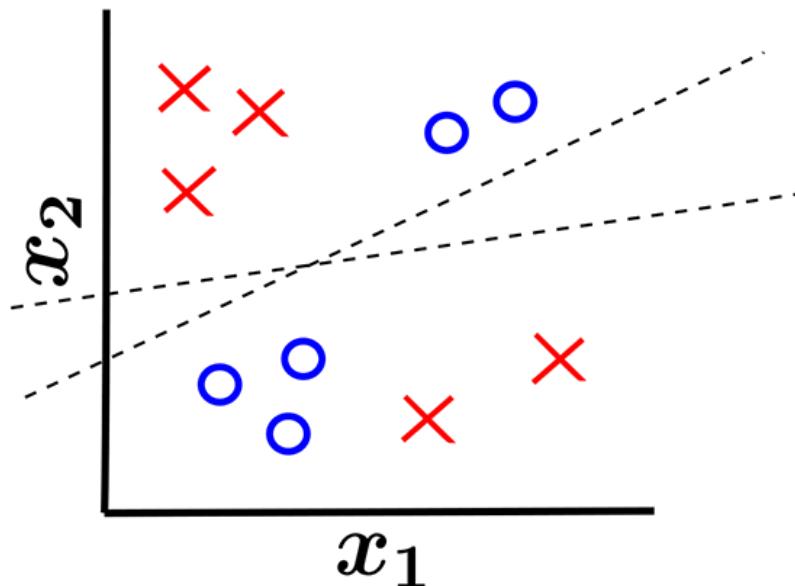
РЕЗЮМЕ

- › Нейрон, или однослойная нейронная сеть — функция активации от линейной комбинации признаков объекта
- › При построении сети используются функции активации различных видов
- › Далее: двуслойные и многослойные нейронные сети

МНОГОСЛОЙНАЯ НЕЙРОННАЯ СЕТЬ ФУНКЦИЯ ОШИБКИ

ПРЕДЕЛЫ ПРИМЕНИМОСТИ ОДНОСЛОЙНЫХ СЕТЕЙ

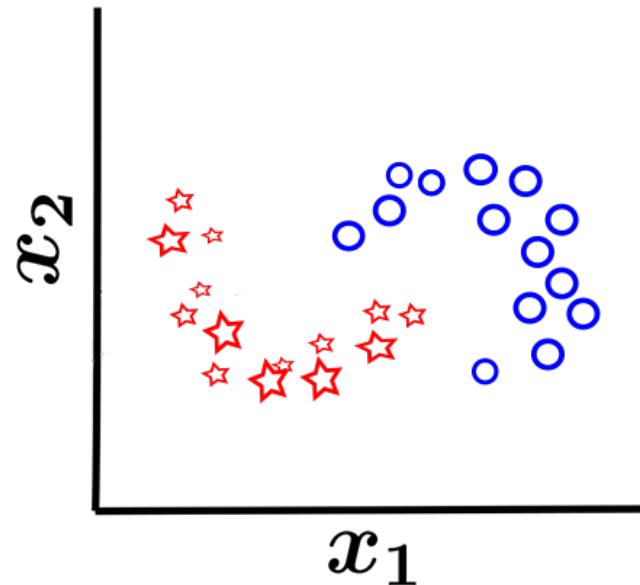
- › Однослойные сети применимы только для линейно разделимых выборок



- › Не существует плоскости, разделяющей данную выборку на два класса корректно (без ошибок)

ПРЕДЕЛЫ ПРИМЕНИМОСТИ ОДНОСЛОЙНЫХ СЕТЕЙ

- » Для корректного разделения этой выборки подходит кривая линия



КОМБИНАЦИЯ НЕЙРОННЫХ СЕТЬ

- › Эта сеть состоит из линейной комбинации нейронов (однослойных нейронных сетей):

$$a(x, \mathbf{W}) = \sigma^{(2)} \left(\sum_{i=1}^D w_i^{(2)} \cdot \sigma^{(1)} \left(\sum_{j=1}^d w_{ji}^{(1)} x_j + w_{0i}^{(1)} \right) + w_0^{(2)} \right)$$

- › В векторных обозначениях:

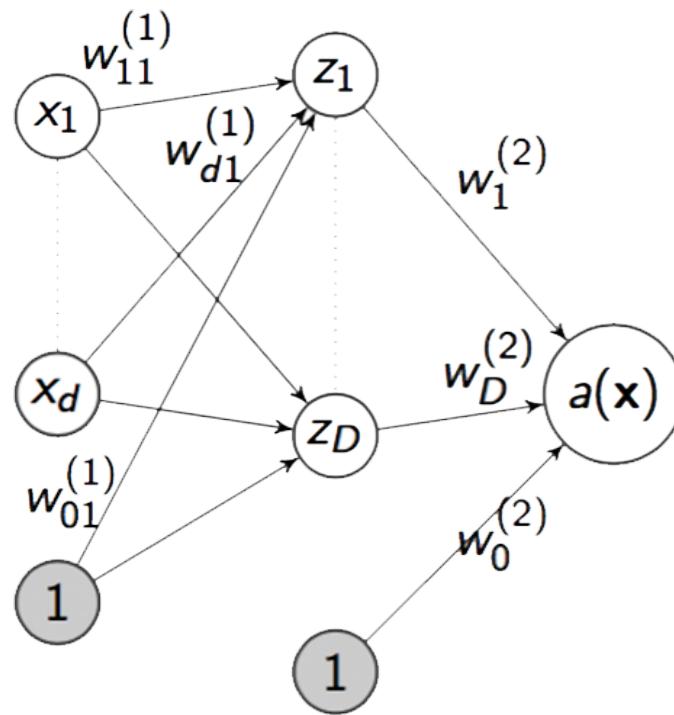
$$a(x, \mathbf{W}) = \sigma^{(2)} \left(\mathbf{w}^{T(2)} \sigma^{(1)} \left([\mathbf{w}_1^{T(1)} x, \dots, \mathbf{w}_D^{T(1)} x] \right) \right)$$

- › Соединенный вектор параметров:

$$\mathbf{w} = \{w_i^{(2)}, w_{ij}^{(1)}, w_{i0}^{(1)}, w_0^{(2)}\}$$

ДВУХСЛОЙНАЯ НЕЙРОННАЯ СЕТЬ

- › Двухслойная сеть представима в виде двудольных направленных графов, где исходящая вершина графа связана со всеми входящими вершинами



ДВУХСЛОЙНАЯ НЕЙРОННАЯ СЕТЬ

- › Граф можно продолжать вправо и дальше для получения многослойной нейронной сети

РАЗДЕЛЯЮЩАЯ СПОСОБНОСТЬ

- › Теорема Хорника (Универсальная теорема аппроксимации, 1991)

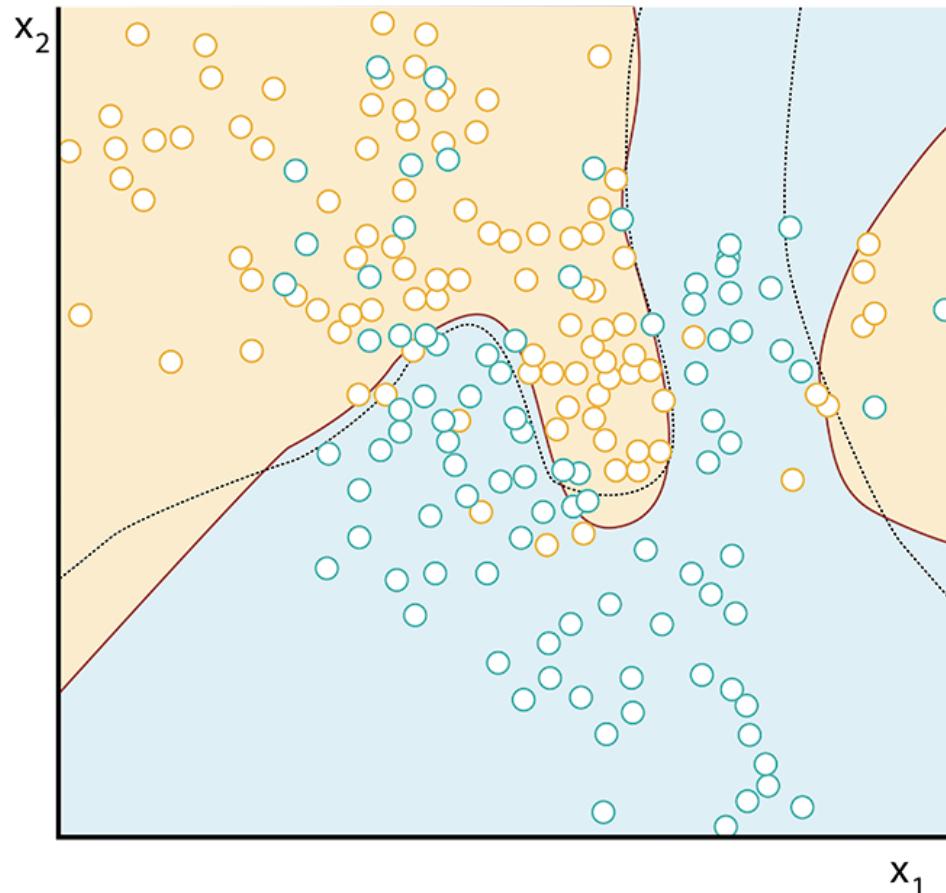
Для любой непрерывной функции найдется нейронная сеть $a(x)$ с линейным выходом, аппроксимирующая $f(x)$ в заданной точностью

РАЗДЕЛЯЮЩАЯ СПОСОБНОСТЬ

- › Теорема выполняется для $\sigma(f) = \text{sigmoid}(f)$, $\sigma(f) = \tanh(f)$ и ряда других функций активации
- › Для получения этой заданной точности необходимо определить оптимальные параметры W^*

РАЗДЕЛЯЮЩАЯ ПОВЕРХНОСТЬ

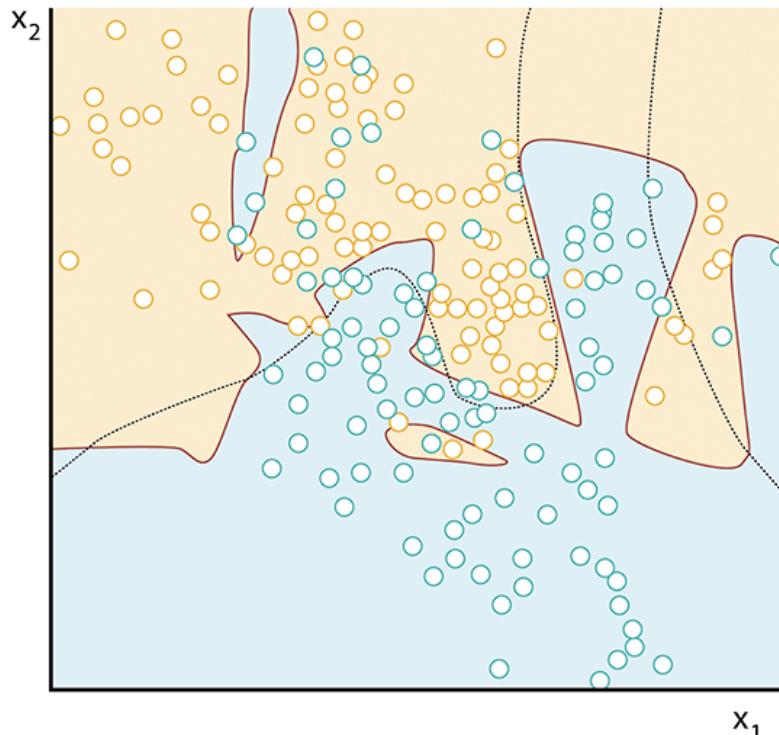
- » Качество аппроксимации y функцией $a(x, w)$ зависит от оптимизации параметров w



РАЗДЕЛЯЮЩАЯ ПОВЕРХНОСТЬ

- › Несколько объектов попали в область, принадлежащую другому классу, вследствие случайной природы выборки

РАЗДЕЛЯЮЩАЯ ПОВЕРХНОСТЬ



- › Параметры w нейросети настроены таким образом, разделение текущей выборки корректно, но при изменении состава выборки оно не будет корректным

ПЕРЕОБУЧЕНИЕ НЕЙРОННОЙ СЕТИ

- › Сеть, корректно аппроксимирующая обучающую выборку и плохо аппроксимирующая контрольную, называется переобученной

ФУНКЦИЯ ОШИБКИ НЕЙРОННОЙ СЕТИ

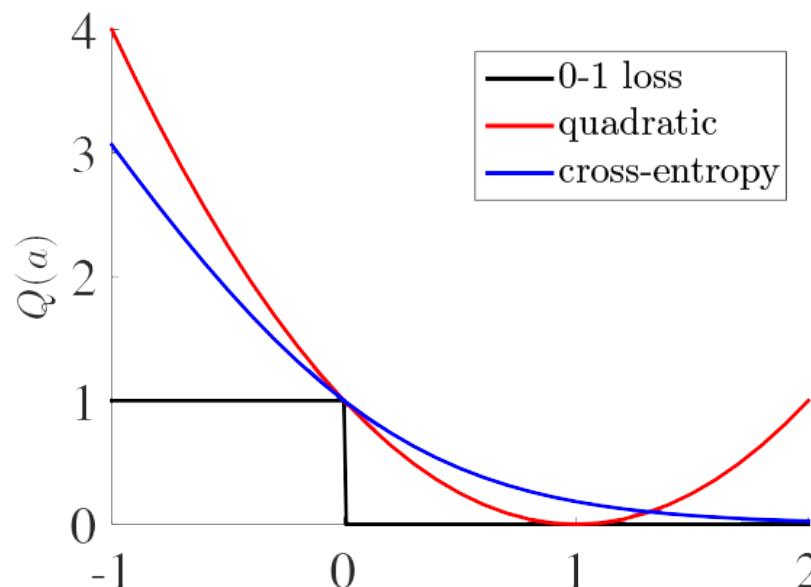
- › Оптимизируем параметры так, чтобы минимизировать значение функции ошибки:

$$w^* = \underset{w}{\operatorname{argmin}} Q(w)$$

ФУНКЦИЯ ОШИБКИ ДЛЯ ЗАДАЧИ РЕГРЕССИИ

- » Для задачи регрессии функция ошибки — “галочка”, сумма модулей разности между восстановленным значением и фактическим измерением:

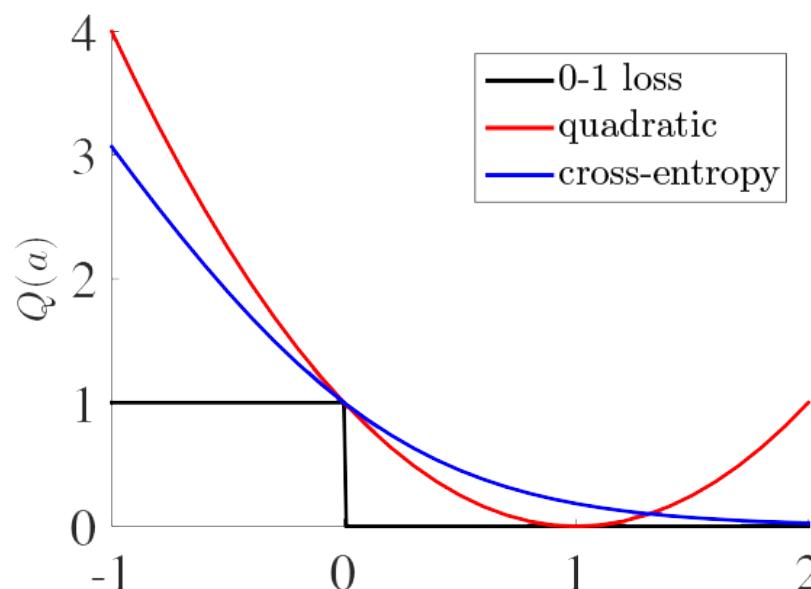
$$Q(\mathbf{w}) = \sum_{i=1}^{\ell} |a(x_i, \mathbf{w}) - y_i|$$



ФУНКЦИЯ ОШИБКИ ДЛЯ ЗАДАЧИ РЕГРЕССИИ

- › Дифференцируемая функция ошибки — сумма квадратов разности между восстановленным значением и фактическим измерением:

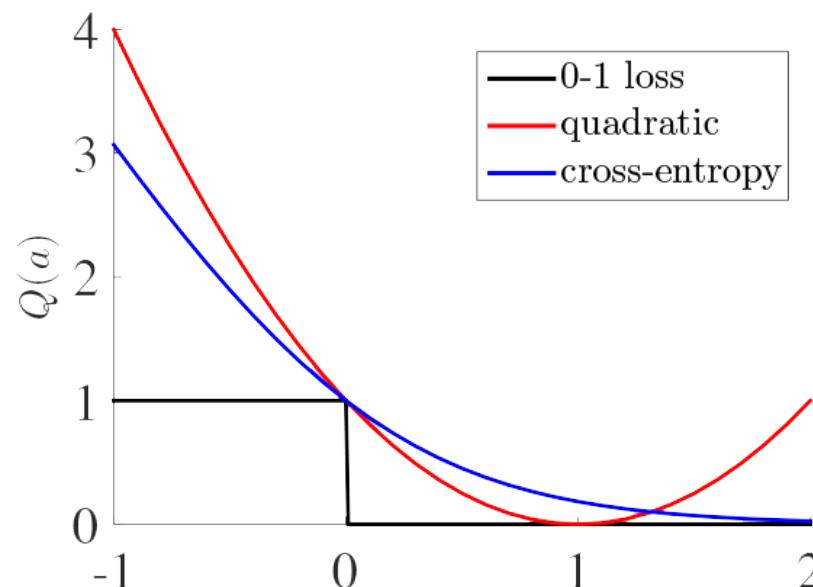
$$Q(\mathbf{w}) = \sum_{i=1}^{\ell} (a(x_i, \mathbf{w}) - y_i)^2$$



ФУНКЦИЯ ОШИБКИ ДЛЯ ЗАДАЧИ КЛАССИФИКАЦИИ

- » Функция ошибки “0–1 loss” — число несовпадений между восстановленными метками классов и фактическими:

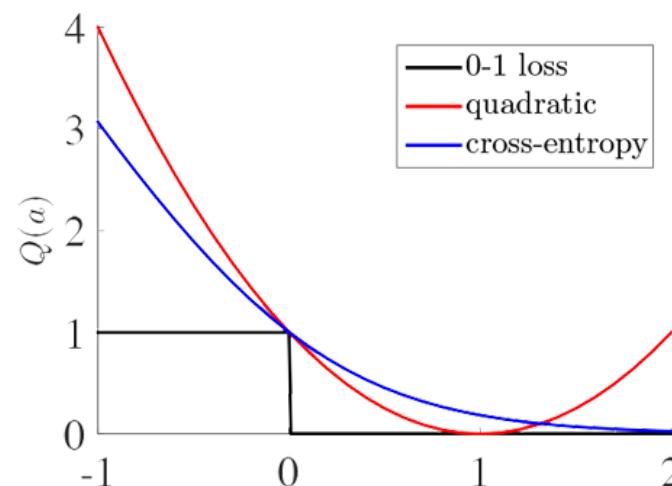
$$Q(\mathbf{w}) = \sum_{i=1}^{\ell} [\text{sign } a(\mathbf{x}_i, \mathbf{w}) \neq y_i]$$



ФУНКЦИЯ ОШИБКИ ДЛЯ ЗАДАЧИ КЛАССИФИКАЦИИ

- › Дифференцируемая функция ошибки — кросс-энтропия, функция наибольшего правдоподобия в задаче логистической регрессии:

$$Q(\mathbf{w}) = - \sum_{i=1}^{\ell} (y_i \ln a(x_i, \mathbf{w}) + (1 - y_i) \ln (1 - f(x_i, \mathbf{w})))$$



РЕЗЮМЕ

- › С помощью многослойной нейронной сети можно получить аппроксимацию высокой точности
- › Для этого надо задать оптимальную структуру нейросети и оптимизировать её параметры
- › Параметры оптимизируются с помощью функции ошибки

РЕЗЮМЕ

- › Для этого надо задать оптимальную структуру нейросети и оптимизировать её параметры
- › Параметры оптимизируются с помощью функции ошибки
- › Существуют дифференцируемые функции ошибки для оптимизации параметров

РЕЗЮМЕ

- › Параметры оптимизируются с помощью функции ошибки
- › Существуют дифференцируемые функции ошибки для оптимизации параметров
- › Далее: алгоритмы оптимизации параметров нейронной сети

ОПТИМИЗАЦИЯ ПАРАМЕТРОВ НЕЙРОННОЙ СЕТИ

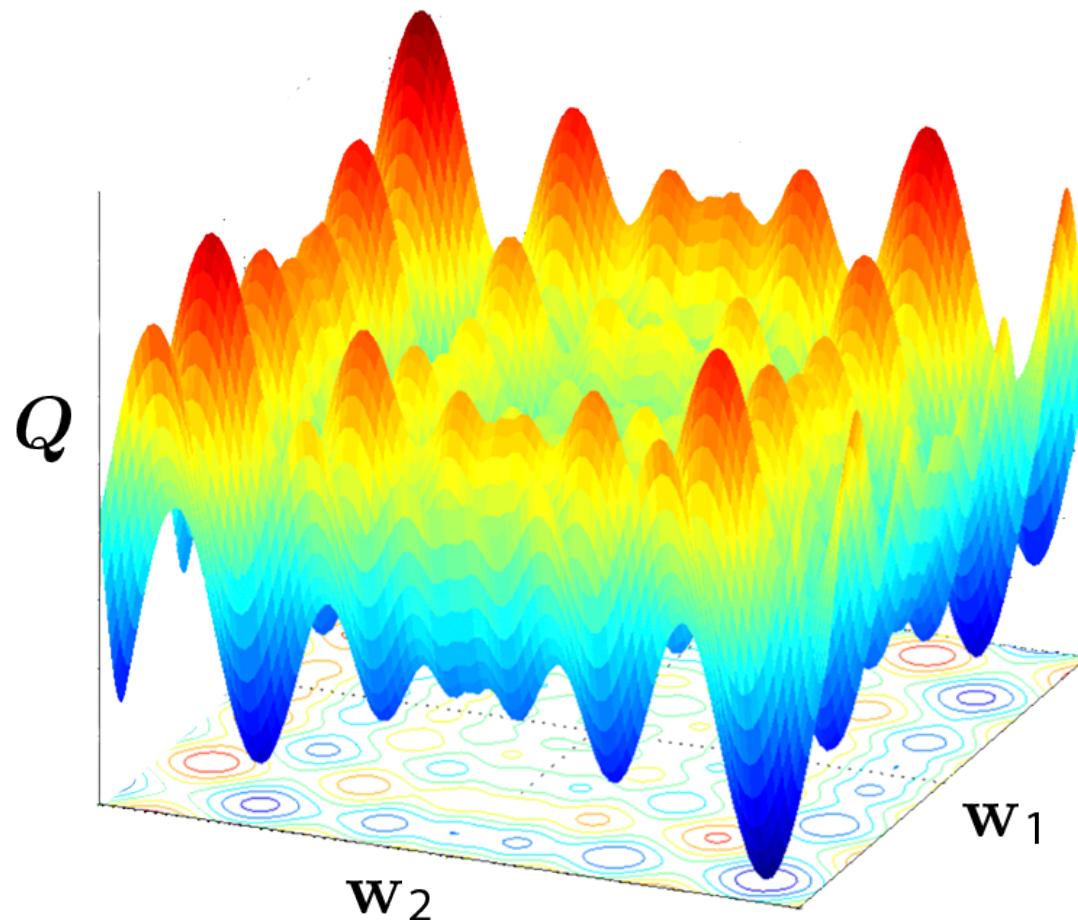
ЗАДАЧА ОПТИМИЗАЦИИ

$$\mathbf{w}^* = \underset{\mathbf{w}}{\operatorname{argmin}} Q(\mathbf{w})$$

- » Функция ошибки Q зависит от:
 - ▶ выборки $(\mathbf{x}_i, y_i), i = 1, \dots, \ell$
 - ▶ структуры нейросети (числа слоев, нейронов, видов функций активации)
 - ▶ значения вектора параметров \mathbf{w}

ФУНКЦИЯ ОШИБКИ

- › Функция ошибки может иметь значительное число локальных минимумов



СТОХАСТИЧЕСКАЯ ОПТИМИЗАЦИЯ

- › Перебираем решения w там, где они могут доставить минимум функции ошибки Q :
 - ▶ случайный перебор w_1, w_2, \dots ,
 - ▶ генетический алгоритм оптимизации
$$w_1 \rightarrow w_2 \rightarrow \dots$$
 - ▶ моделируемый отжиг, значения задаются по расписанию w

ЛОКАЛЬНАЯ ОПТИМИЗАЦИЯ ФУНКЦИИ ОШИБКИ

- » Минимизируем не функцию ошибки Q , а функцию, которая её аппроксимирует в окрестности точки w^* . Например, квадратичную функцию:

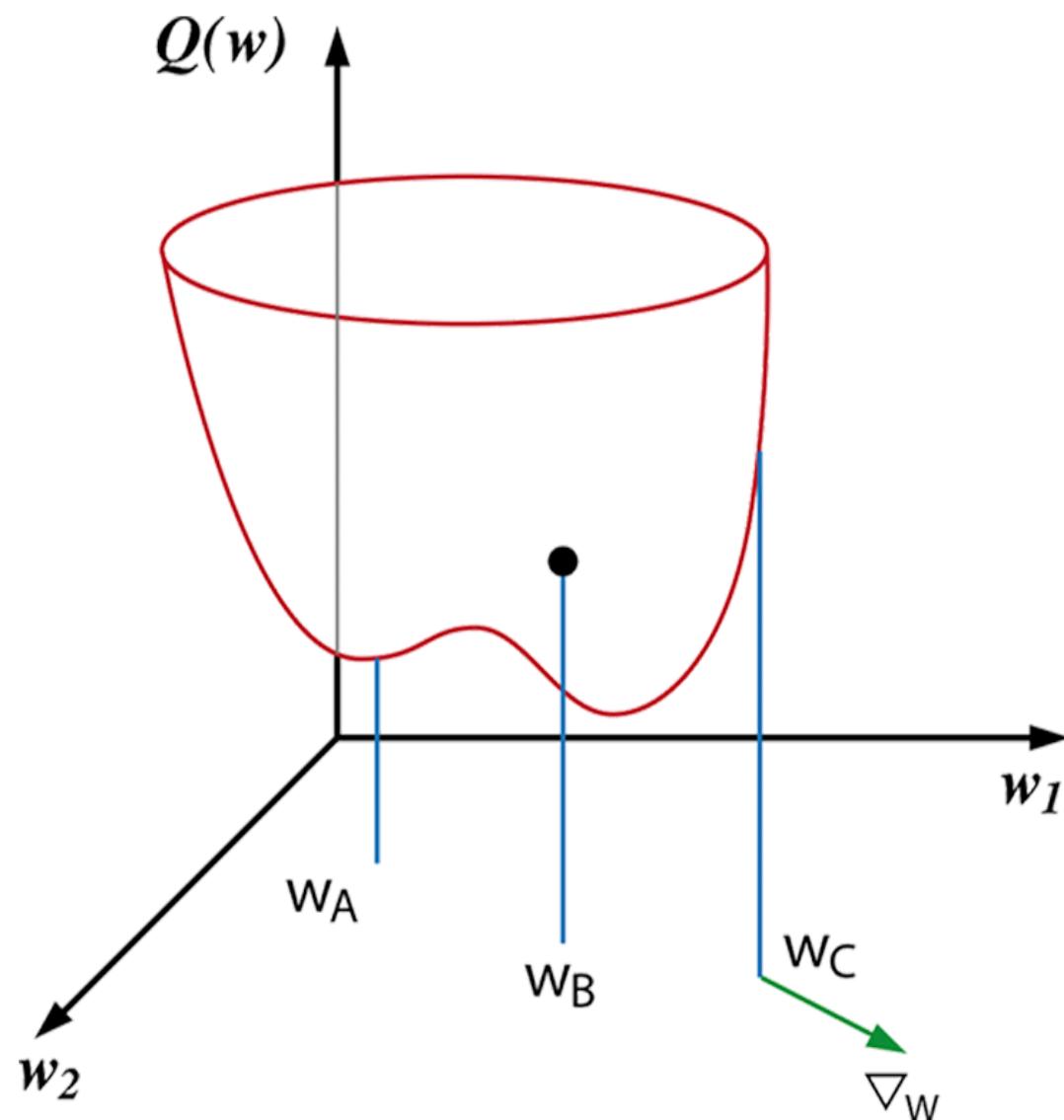
$$Q(w) \approx Q(w^*) + \frac{1}{2}(w - w^*)^T H(w - w^*)$$

- » На каждом шаге находим минимум такой функции и сдвигаем окрестность

ГРАДИЕНТНЫЙ СПУСК

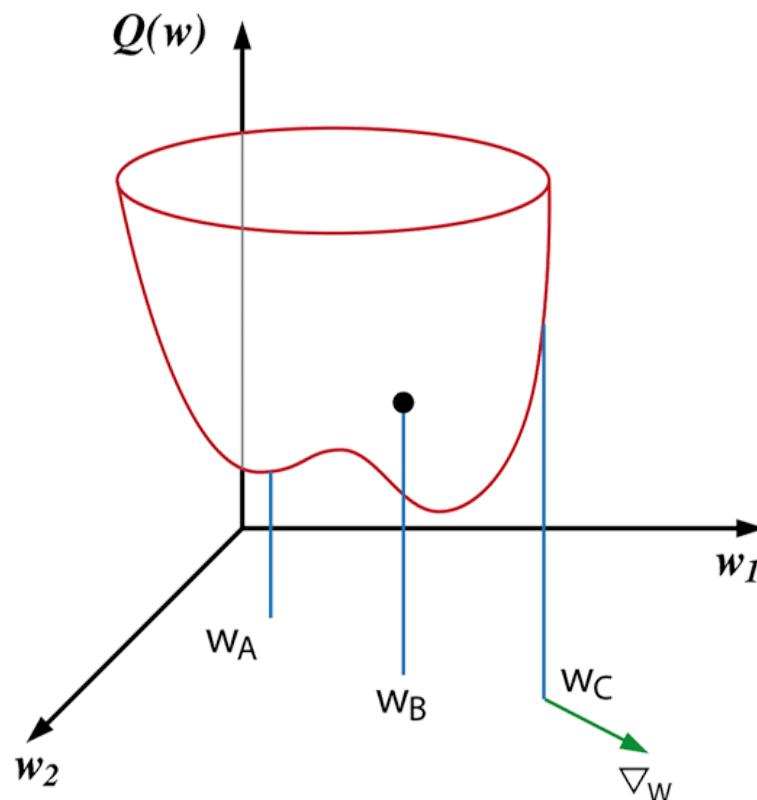
- › Предполагается, что функция ошибки $Q(\mathbf{w})$ дифференцируема (квадратичная, кросс-энтропийная)
- › Задавая значение параметров \mathbf{w} , можно вычислить значение функции ошибки $\nabla_{\mathbf{w}} Q$ и её градиент

ГРАДИЕНТНЫЙ СПУСК



ГРАДИЕНТНЫЙ СПУСК

- › Требуется, отправляясь из исходной точки w_C , спуститься к точке минимума шагая в направлении ∇_w



ГРАДИЕНТНЫЙ СПУСК

- » Пошаговая процедура нахождения \mathbf{w}^*

$$\mathbf{w}_{k+1} = \mathbf{w}_k + \alpha \sum_{i=1}^{\ell} \nabla_{\mathbf{w}} \varphi(\mathbf{w}_k, x_i)$$

- » φ — ошибка на одном объекте, её частная производная по параметрам:

$$\nabla_{\mathbf{w}} \varphi = \frac{\partial \varphi}{\partial \mathbf{w}}$$

- » α — величина шага, k — номер итерации

СТОХАСТИЧЕСКИЙ ГРАДИЕНТНЫЙ СПУСК

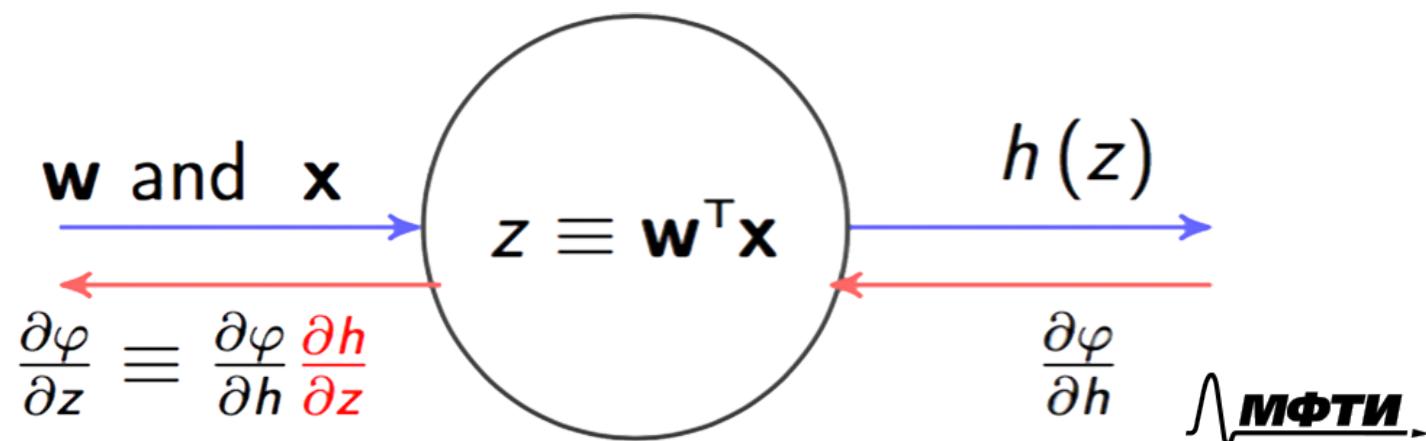
- › Направление градиента $\nabla_w \varphi$, вычисляется для случайно выбранного объекта x_i

$$w_{k+1} = w_k + \alpha \nabla_w \varphi(w_k, x_k)$$

- › Процедура использует случайно переупорядоченные объекты x_1, \dots, x_ℓ

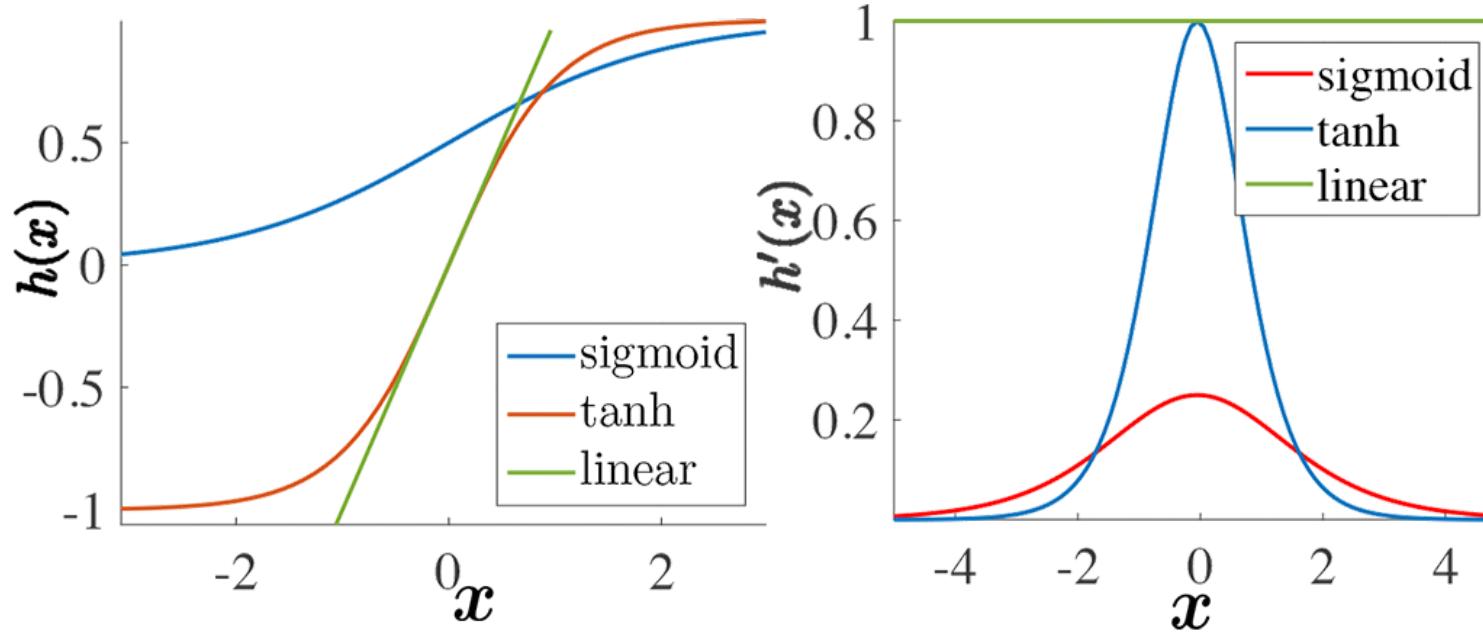
BACKPROP

- » Для каждого нейрона сети $h(\mathbf{w}^T \mathbf{x})$ и одного объекта \mathbf{x} вычисляется:
 - ▶ значение его выхода h
 - ▶ производная ошибки $\frac{\partial \varphi}{\partial z}$ по значению нейрона и его параметрам



ПРОБЛЕМА ЗАТУХАНИЯ ГРАДИЕНТА

- › При больших значениях $|x|$ значения $h'(x) \rightarrow 0$ для **sigmoid**(x) и **tanh** (x)



- › Градиент $\frac{\partial \varphi}{\partial z^s}$ не распространяется

ПРЕИМУЩЕСТВА BACKPROP

- › Градиент вычисляется за время, сравнимое с вычислением сети
- › Подходит для многих дифференцируемых функций активации
- › Необязательно использовать всю выборку

НЕДОСТАТКИ BACKPROP

- › Возможна медленная сходимость к решению
- › Решение может оказаться в локальном минимуме
- › Возможно переобучение сети

РЕЗЮМЕ

- › Для оптимизации используются стохастические и градиентные методы
- › Стохастические требуют многократного угадывания вектора параметров
- › Градиентные требуют дифференцирования функции ошибки
- › Выбор метода оптимизации остается прикладным искусством

РЕЗЮМЕ

- › Далее: регуляризация и прореживание нейронной сети

РЕГУЛЯРИЗАЦИЯ И ПРОРЕЖИВАНИЕ НЕЙРОННОЙ СЕТИ

РЕГУЛЯРИЗАЦИЯ

- › Чтобы избежать переобучения, модифицируем задачу оптимизации — добавим к функции ошибки штраф за большие значения параметров:

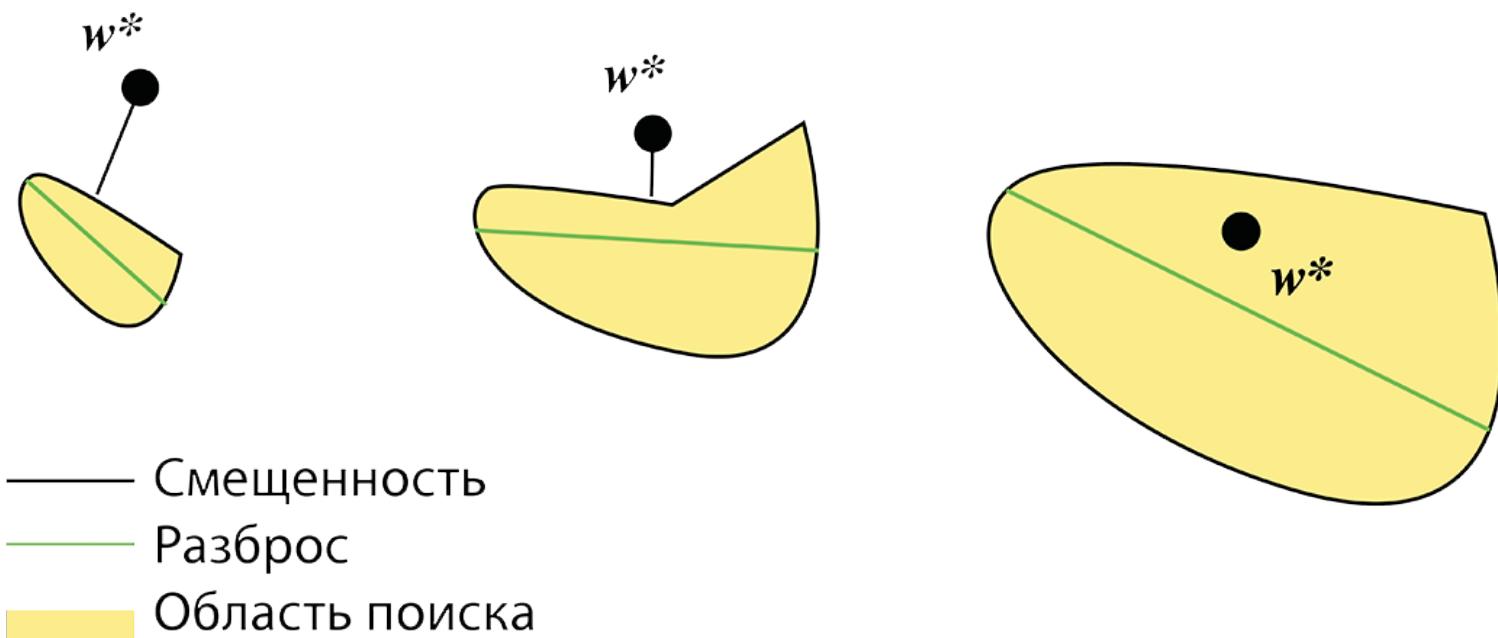
$$\mathbf{w}^* = \underset{\mathbf{w}}{\operatorname{argmin}} \left[Q(\mathbf{w}) + \tau \sum_{i,j,k} (\mathbf{w}_{ij}^{(k)})^2 \right]$$

РЕГУЛЯРИЗАЦИЯ

- › Чем меньше коэффициент регуляризации τ , тем точнее функция описывает выборку

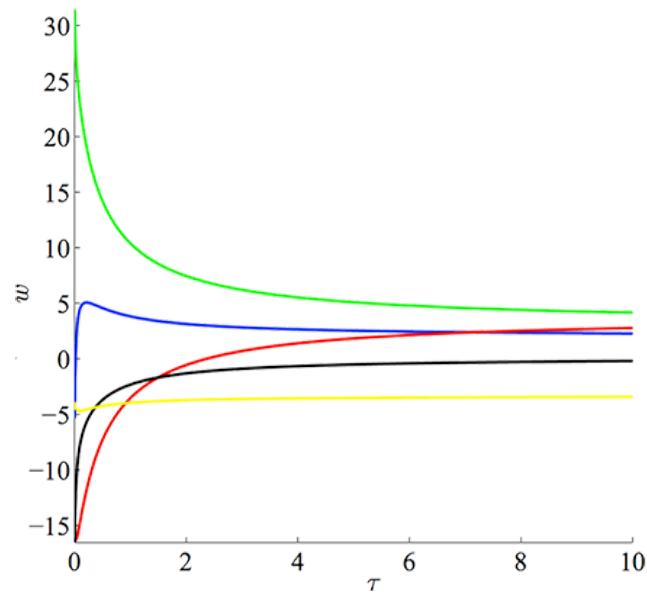
РЕГУЛЯРИЗАЦИЯ

- › Коэффициент регуляризации τ контролирует жесткость ограничений параметров w



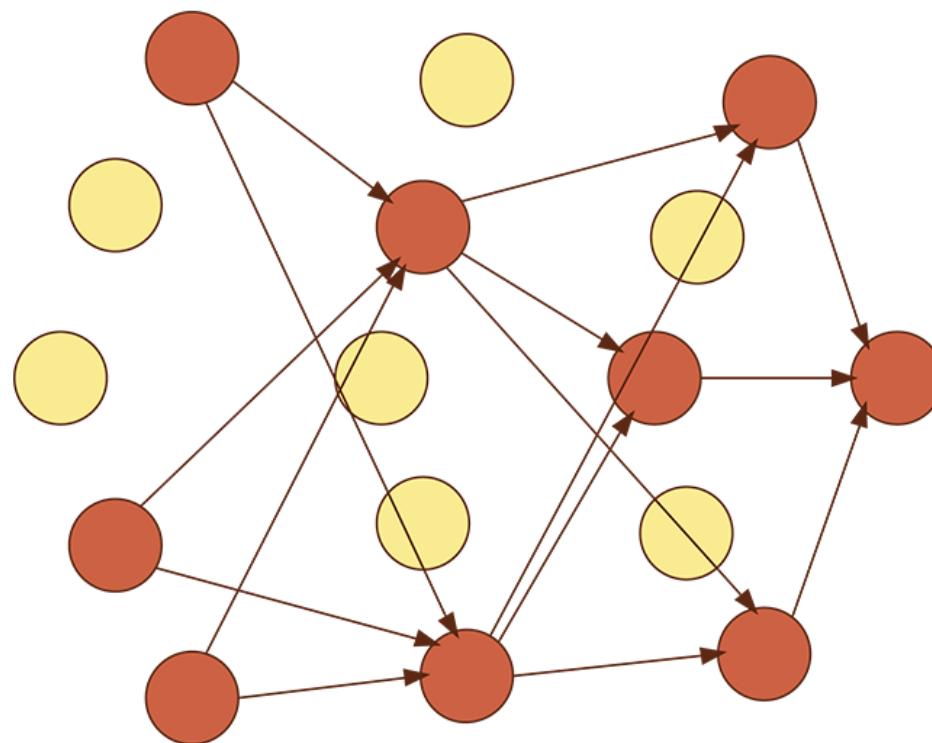
НЕДОСТАТОК РЕГУЛЯРИЗАЦИИ

- › Регуляризация не снижает число параметров и не упрощает структуру сети
- › При увеличении коэффициента τ параметры перестают изменяться



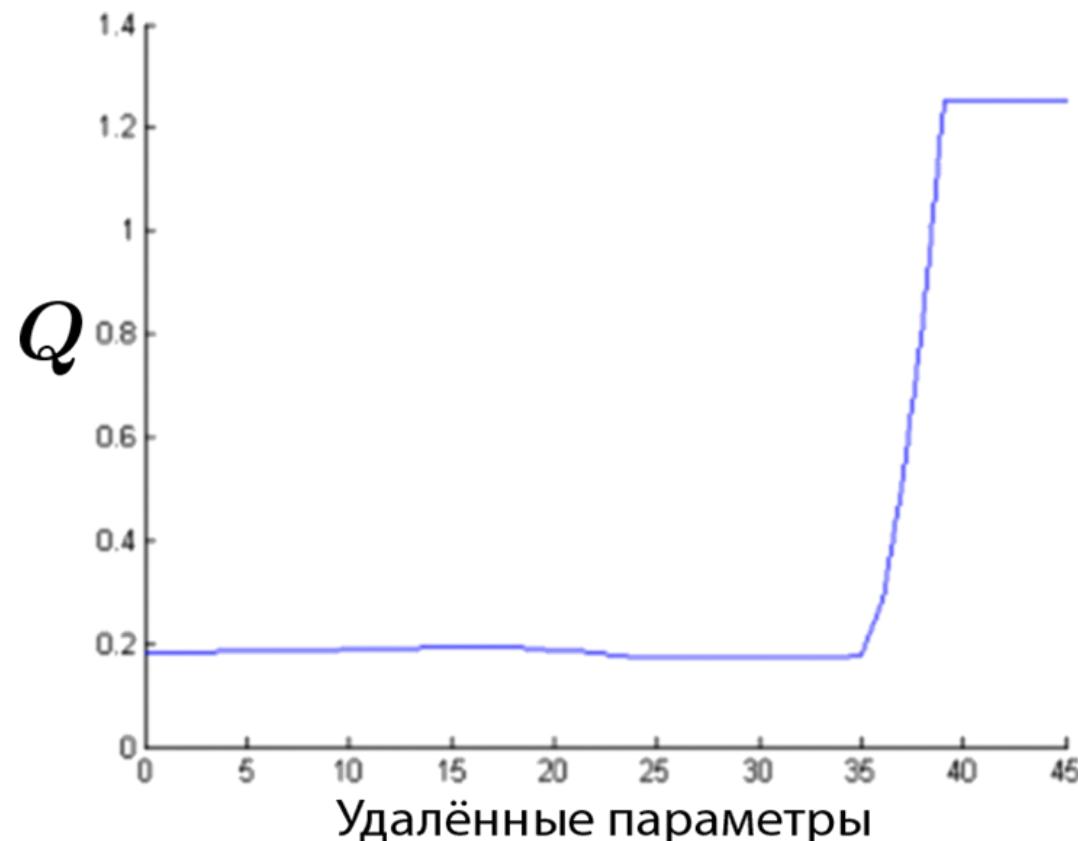
ПРОРЕЖИВАНИЕ СЕТИ

- › Для снижения числа параметров предлагается исключить некоторые нейроны или связи



ПРОРЕЖИВАНИЕ СЕТИ

- › Если функция ошибки не изменяется, сеть можно упрощать и дальше



СТРАТЕГИИ ПРОРЕЖИВАНИЯ ПАРАМЕТРОВ СЕТИ

- Параметр можно удалить, если:
 - ▶ Он имеет значение, близкое к нулю
 - ▶ Его значение сильно изменяется при изменении выборки (большая дисперсия)
 - ▶ Его удаление меньше всего влияет на изменение значения функции ошибки

МЕТОД ОПТИМАЛЬНОГО ПРОРЕЖИВАНИЯ

» Разложим функцию ошибки в ряд Тейлора:

$$Q(\mathbf{w} + \Delta \mathbf{w}) = \underbrace{Q(\mathbf{w})}_{\text{const}} + \underbrace{\mathbf{g}^T(\mathbf{w}) \Delta \mathbf{w}}_0 + \\ + \frac{1}{2} \Delta \mathbf{w}^T \mathbf{H} \Delta \mathbf{w} + \underbrace{o(\|\mathbf{w}\|^3)}_{\text{small}}$$

» Минимизируем, исключив параметр, для которого $\Delta \mathbf{w}_j = -\mathbf{w}_j$

МЕТОД ОПТИМАЛЬНОГО ПРОРЕЖИВАНИЯ

» Разложим функцию ошибки в ряд Тейлора:

$$Q(\mathbf{w} + \Delta \mathbf{w}) = \underbrace{Q(\mathbf{w})}_{\text{const}} + \underbrace{\mathbf{g}^T(\mathbf{w}) \Delta \mathbf{w}}_0 + \\ + \frac{1}{2} \Delta \mathbf{w}^T \mathbf{H} \Delta \mathbf{w} + \underbrace{o(\|\mathbf{w}\|^3)}_{\text{small}}$$

» Этот параметр имеет минимальное значение функции выпуклости:

$$L_j = \frac{\mathbf{w}_j^2}{2H_{jj}^{-1}}$$

ПОСТРОЕНИЕ НЕЙРОННОЙ СЕТИ

- Сеть используется в двух режимах:
 - ▶ Обучение — оптимизация параметров
 - ▶ Эксплуатация — вычисление значений $a(x, w^*)$ при фиксированных значениях параметров

ПОСТРОЕНИЕ НЕЙРОННОЙ СЕТИ

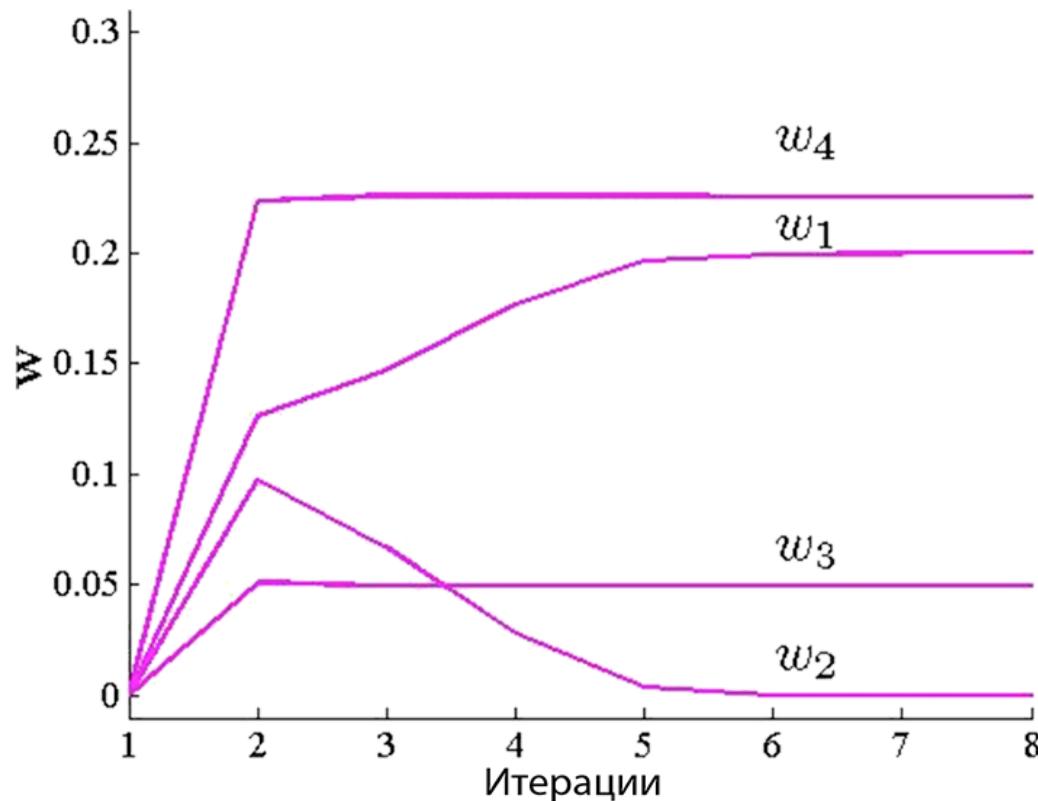
- › Для обучения требуется задать:
 - ▶ Число слоев
 - ▶ Число нейронов в каждом слое
 - ▶ Тип функции активации в каждом слое
 - ▶ Тип функции ошибки

ПОСТРОЕНИЕ НЕЙРОННОЙ СЕТИ

- › Желательно, чтобы подготовленная выборка не содержала пропусков, признаки были отнормированы

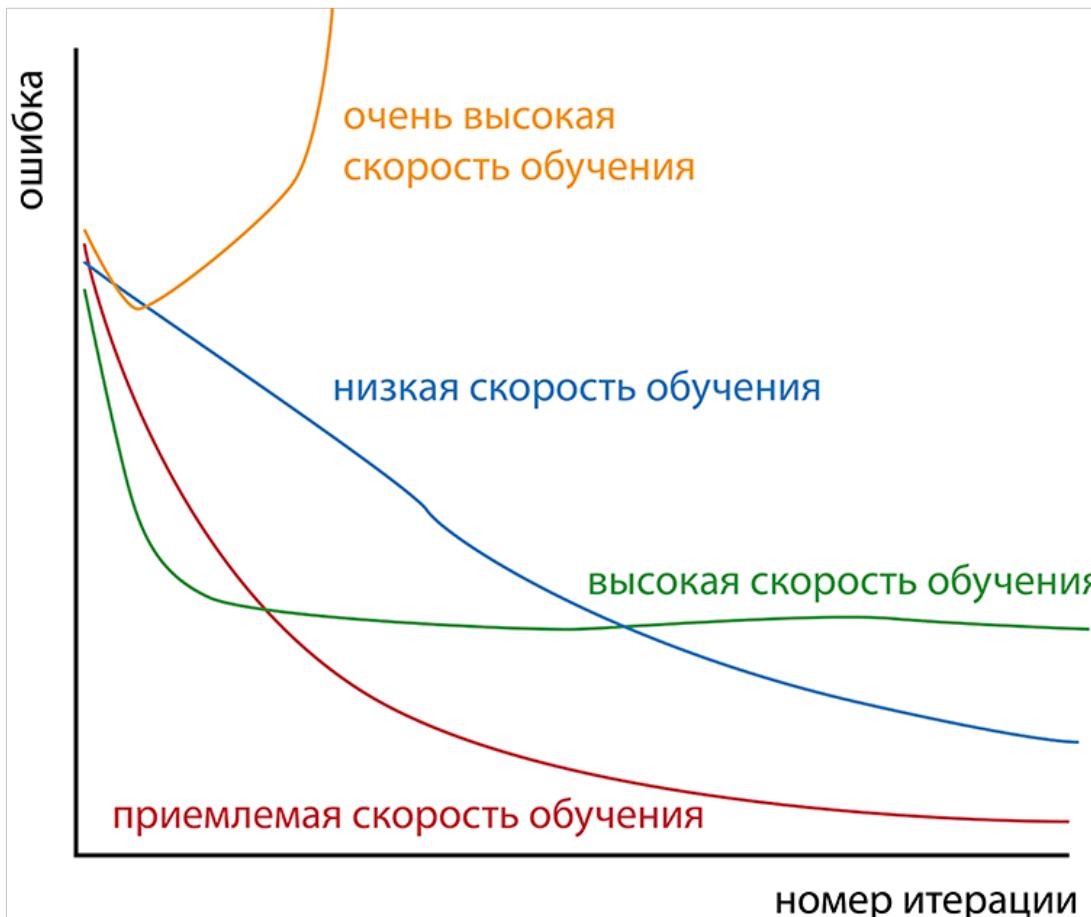
СТАБИЛИЗАЦИЯ ПАРАМЕТРОВ СЕТИ

- › При нахождении минимума (он может оказаться локальным) параметры стабилизируются



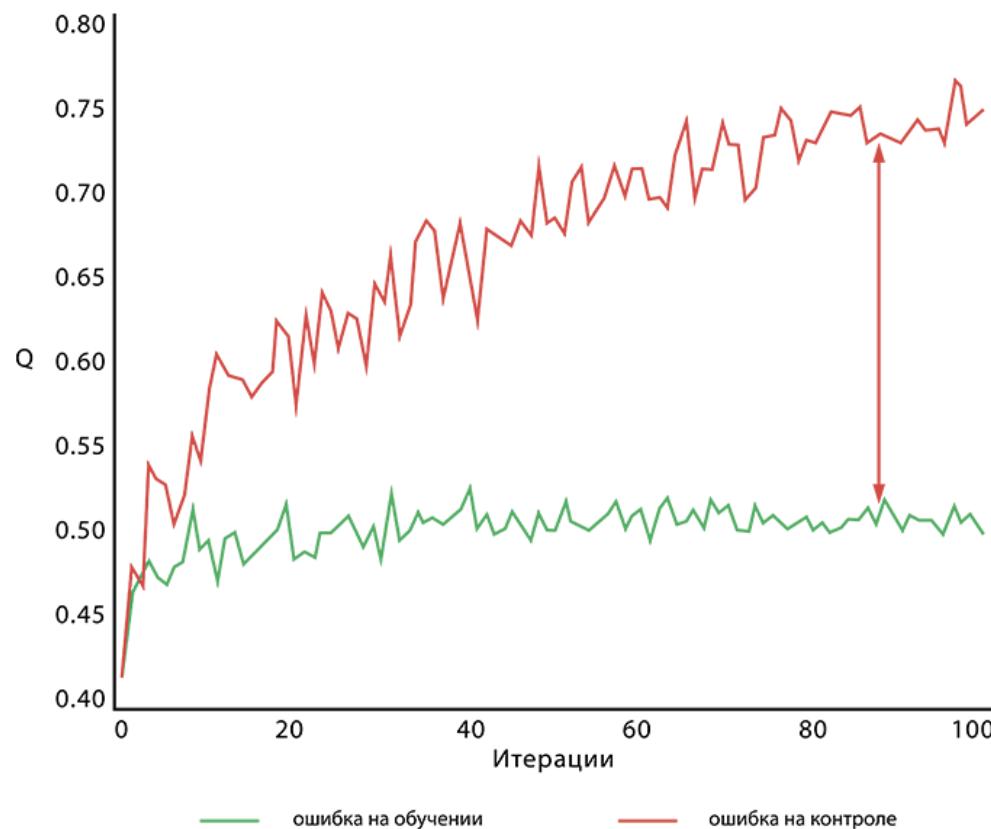
СКОРОСТЬ ОБУЧЕНИЯ СЕТИ

- › По скорости обучения можно судить о соответствии выборки и нейронной сети



ОБУЧЕНИЕ И КОНТРОЛЬ

- Разница между значениями функции ошибки на обучении и контроле не должна быть существенной



РЕЗЮМЕ

- › Регуляризация используется для снижения переобученности путем загрубления параметров
- › Оптимальное прореживание упрощает структуру сети, удаляя параметры
- › Структура нейронной сети существенно зависит от решаемой прикладной задачи

РЕЗЮМЕ

- › Оптимальное прореживание упрощает структуру сети, удаляя параметры
- › Структура нейронной сети существенно зависит от решаемой прикладной задачи
- › Её построение и оптимизация выполняются, как правило, экспериментальным путём