ТЕХНОСФЕРА

Лекция №10

Введение в машинное обучение

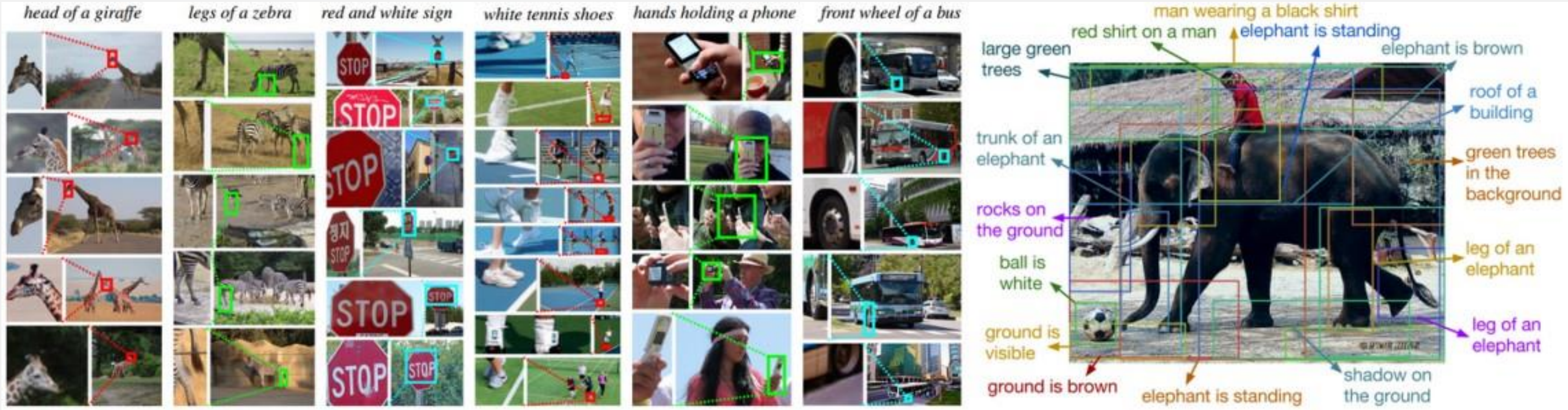
Спасёнов Алексей

Содержание лекции

1. История развития нейронных сетей
2. Методы оптимизации
3. Метод обратного распространения ошибки
4. Инструменты для создания нейронных сетей

Нейронные сети. Примеры.

Describing photos



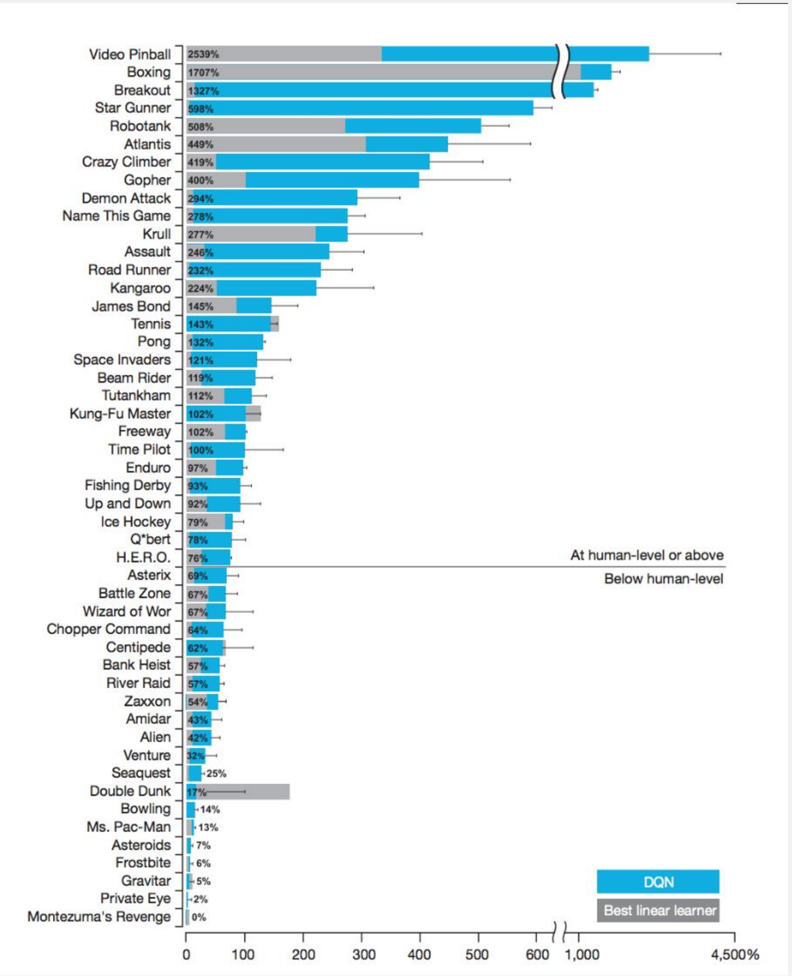
Нейронные сети. Примеры.

Restore colors in B&W photos and videos

Colorado National Park, 1941 | Textile Mill, June 1937 | Berry Field, June 1909 | Hamilton, 1936

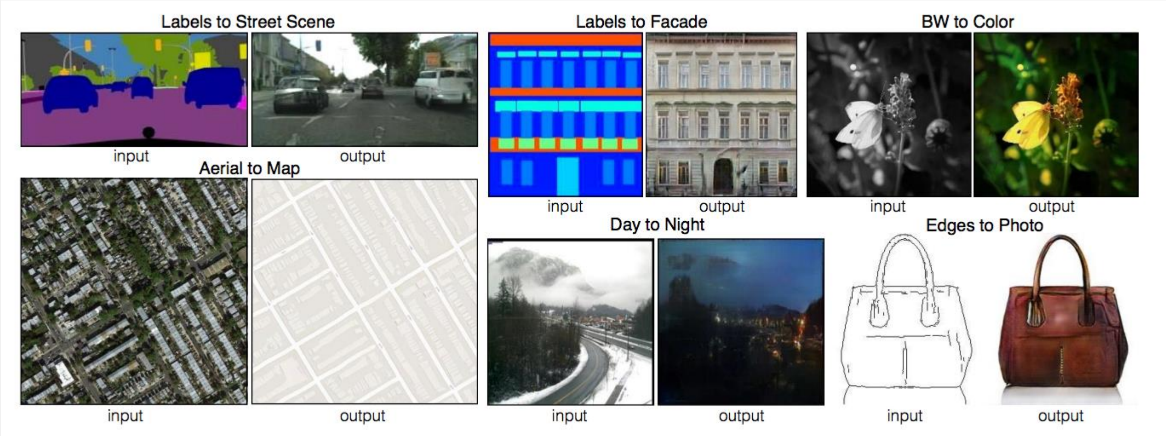
Нейронные сети. Примеры.

Beating people in dozens of computer games



Нейронные сети. Примеры.

Create new images



Нейронные сети. Примеры.

Transferring style from famous paintings

Нейронные сети. Примеры.

Transferring style from famous paintings

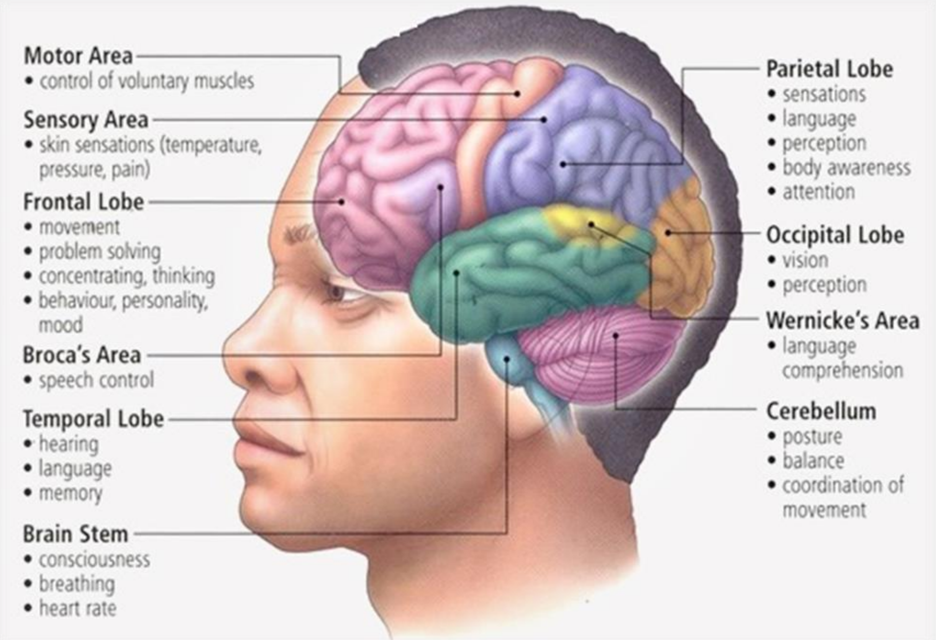
Нейронные сети




Добро пожаловать в deep learning!

Нейронные сети

Структура мозга человека



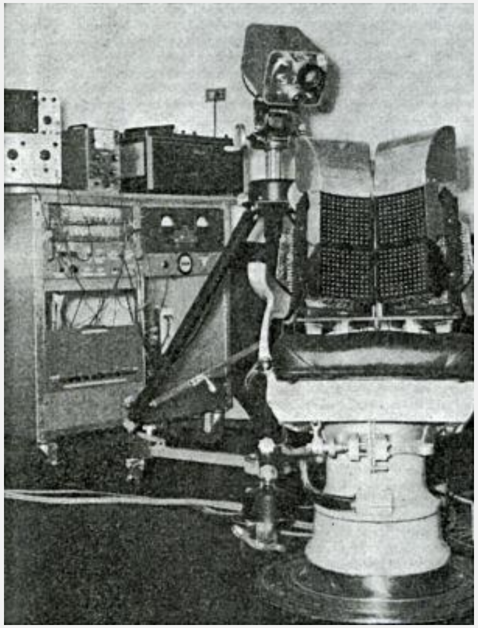
Количество нейронов: $86 \cdot 10^9$

Число транзисторов:
1) GPU Turing: $18.6 \cdot 10^9$
2) Xeon Skylake-E5: $13.2 \cdot 10^9$

Нейроны обладают нейропластичностью

Нейронные сети

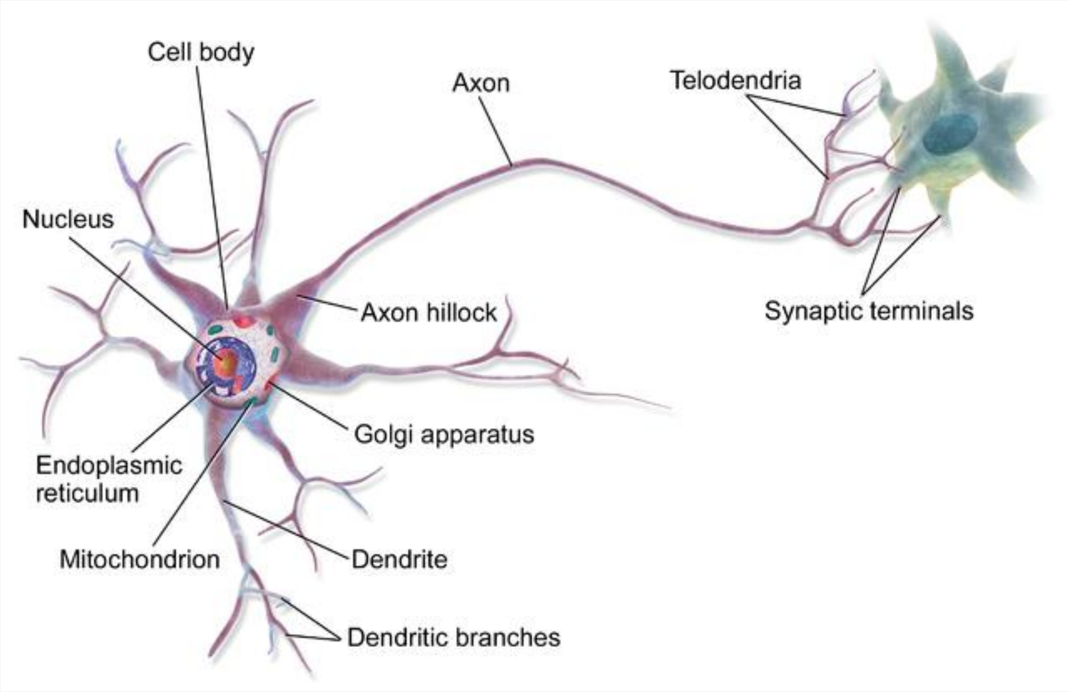
Нейропластичность



Пол Бак-и-Рита (Paul Bach-y-Rita)
Создан прибор (BrainPort), основой которого является матрица электродов (до 20 на 20 штук), присоединяемых к языку (1969)

Нейронные сети

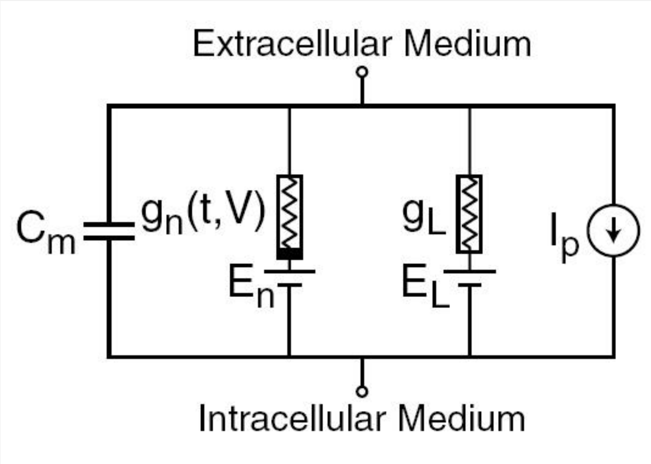
Биологический нейрон



Нейронные сети

Модель Ходжкина — Хаксли

Математическая модель, описывающая генерацию и распространение потенциалов действия в нейронах (1952 год).



C_m - электроёмкость соответствует внутреннему липидному слою клеточной мембраны

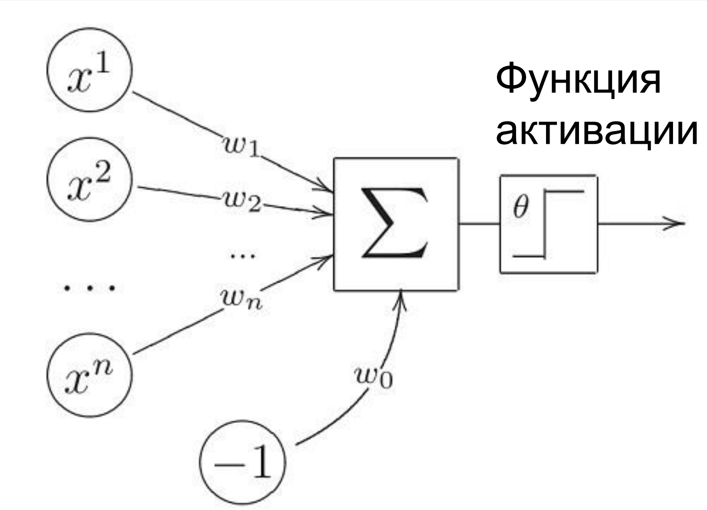
g_n - потенциал-зависимые ионные каналы отвечают за нелинейную электрическую проводимость

g_l - каналы мембранных пор отвечают за пассивную проводимость

E_n, E_L- источники напряжения побуждает ионы к движению через мембранные каналы

Нейронные сети

Нейрон МакКаллока-Питтса (1943 год)



$$y(X) = f\left(\sum_{i=1}^{n}\left(w_i x^i\right) - w_0\right)$$

$f(z)$- ступенчатая функция Хевисайда.

Модель МакКаллока-Питтса эквивалентна пороговому линейному классификатору

Правила Хебба (1949 год)

В 1949 физиологом Дональдом Олдингсом Хеббом была написана книга «Организация сознания», в которой автор описал процесс адаптирования нейронов в мозге человека в процессе обучения.

Правила Хебба:

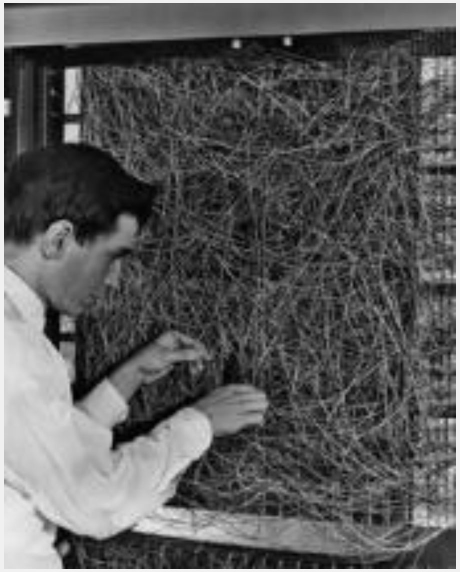
1) Если два нейрона по разные стороны от синапсов активируются синхронно, то «вес» синапса слегка возрастает.

2) Если два нейрона по разные стороны от синапсов активируются асинхронно, то «вес» синапса слегка ослабевает или синапс удаляется.

Нейронные сети

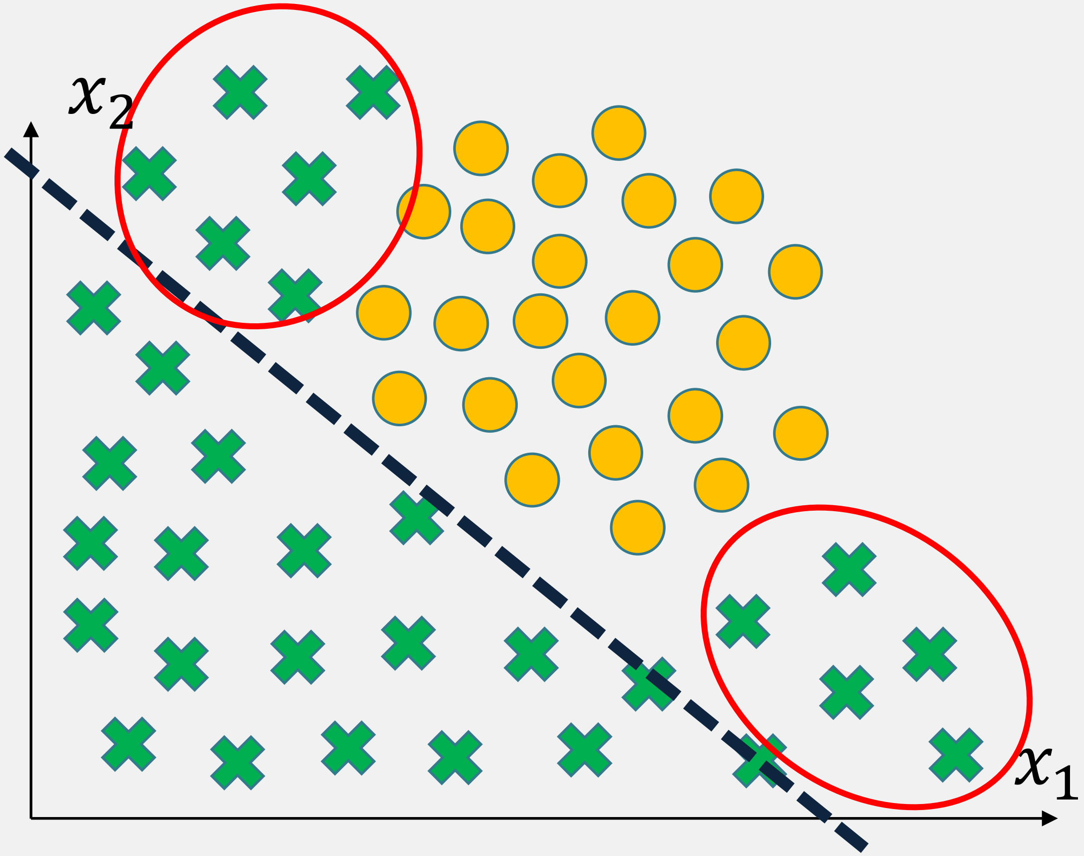
Персептрон Розенблатта (1958 год)

В 1958 нейрофизиологом Френком Розенблаттом была предложена модель восприятия информации человеком и названа «персептроном». Реализовал первый нейрокомпьютер «MARK-1»



Нейронные сети

Персептрон



Персептрон

Ма́рвин Ли Ми́нский (Marvin Lee Minsky), американский учёный в области искусственного интеллекта, провёл детальный математический анализ персептрона и опубликовал формальное доказательство ограниченности этой модели (1969 год).

Отсутствие преимуществ и ограниченность модели убавили интерес научного сообщества к нейронным сетям.

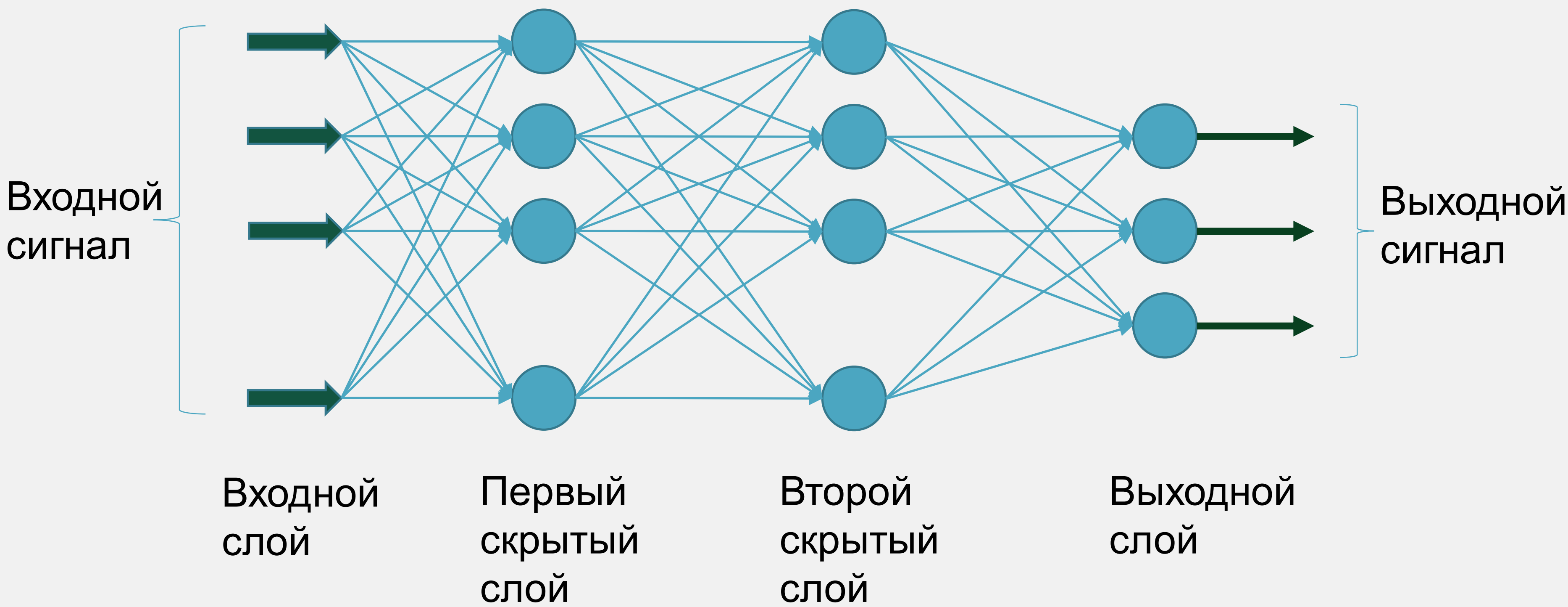
Нейронные сети

Дальнейшее развитие:

1) 1972 год – Т. Кохонен разрабатывает новый тип нейросетей для задачи manifold embedding и topology preserving mapping
2) 1975-1980 – К. Фукусима разрабатывает когнитрон и неокогнитрон, совершенно новый тип многослойной свёрточной сети, созданной по аналогии со строением зрительной системы
3) 1982 год – Дж. Хопфилд разрабатывает новый тип нейросетей с обратными связями, выполняющей функции ассоциативной памяти
4) 1986 год – Д. Румельхарт, Дж. Хинтон и Р. Вильямс разрабатывают вычислительно эффективный метод обучения многослойных нейронных сетей – метод обратного распространения ошибки

Нейронные сети

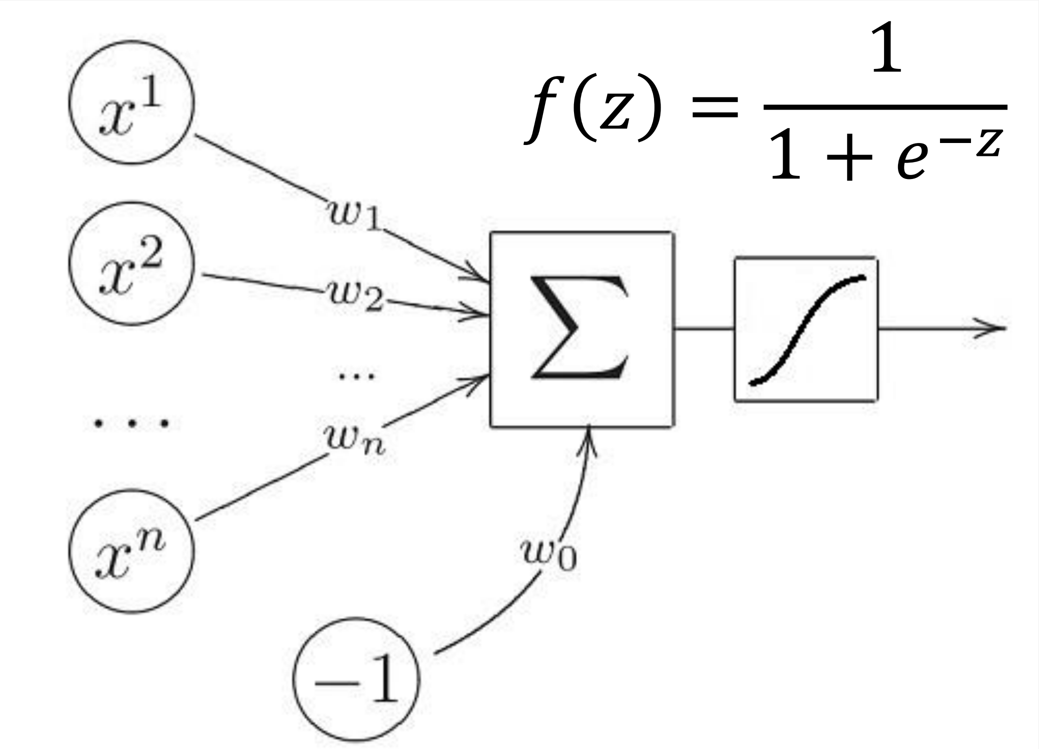
Многослойный персептрон



Нейронные сети

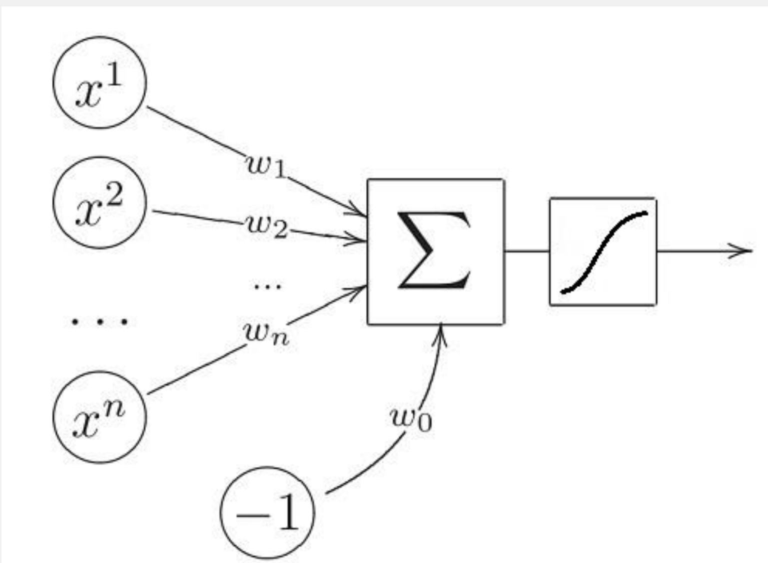
Логистическая регрессия

В качестве функции $f(z)$ возьмём функцию:

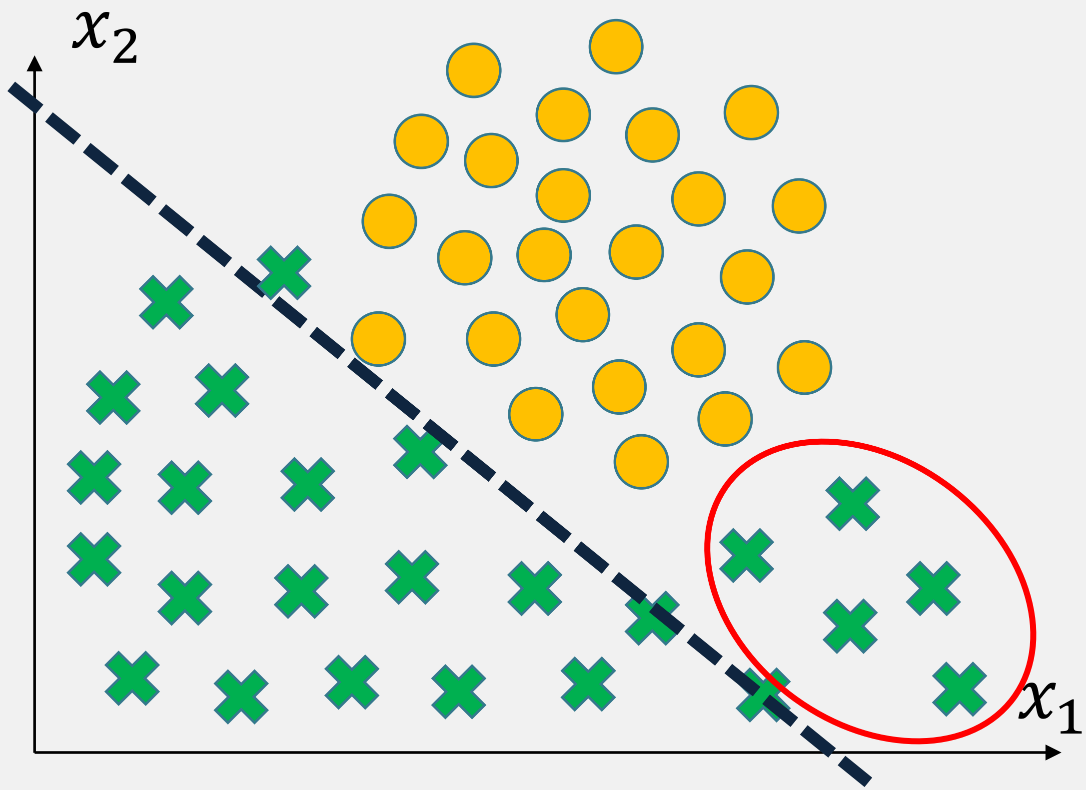


Нейронные сети

Логистическая регрессия

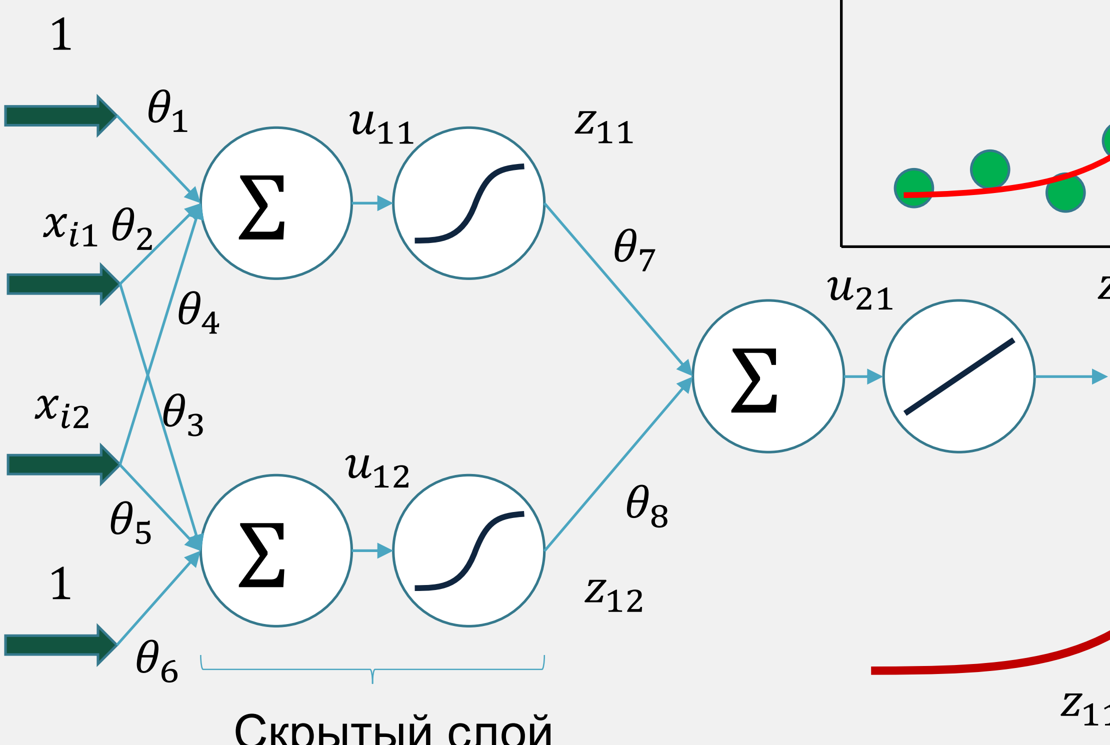


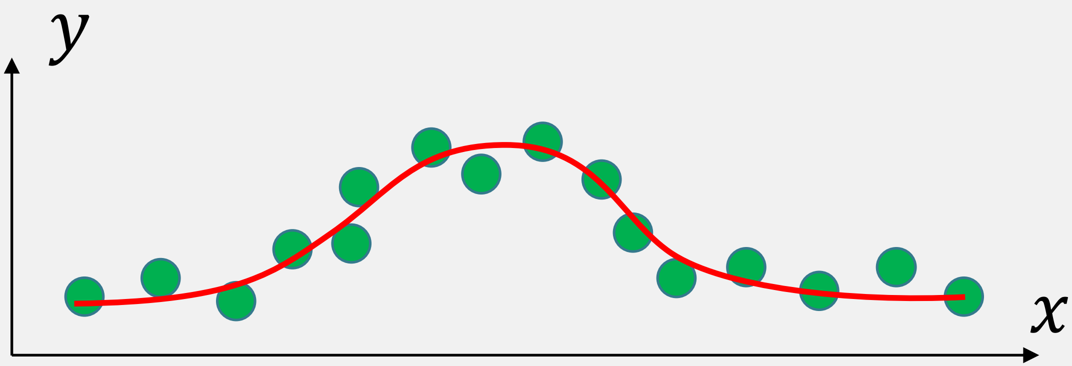
$$f(z) = \frac{1}{1+e^{-z}}$$

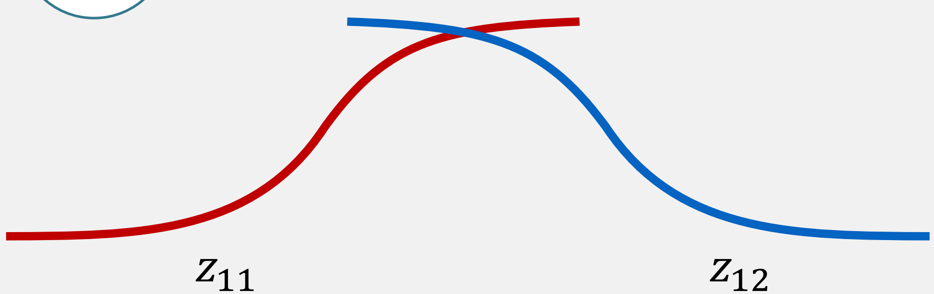


Нейронные сети

Многослойный персептрон (MLP) Решаем задачу регрессии







Как находить параметры модели?

Нейронные сети

Оптимизация

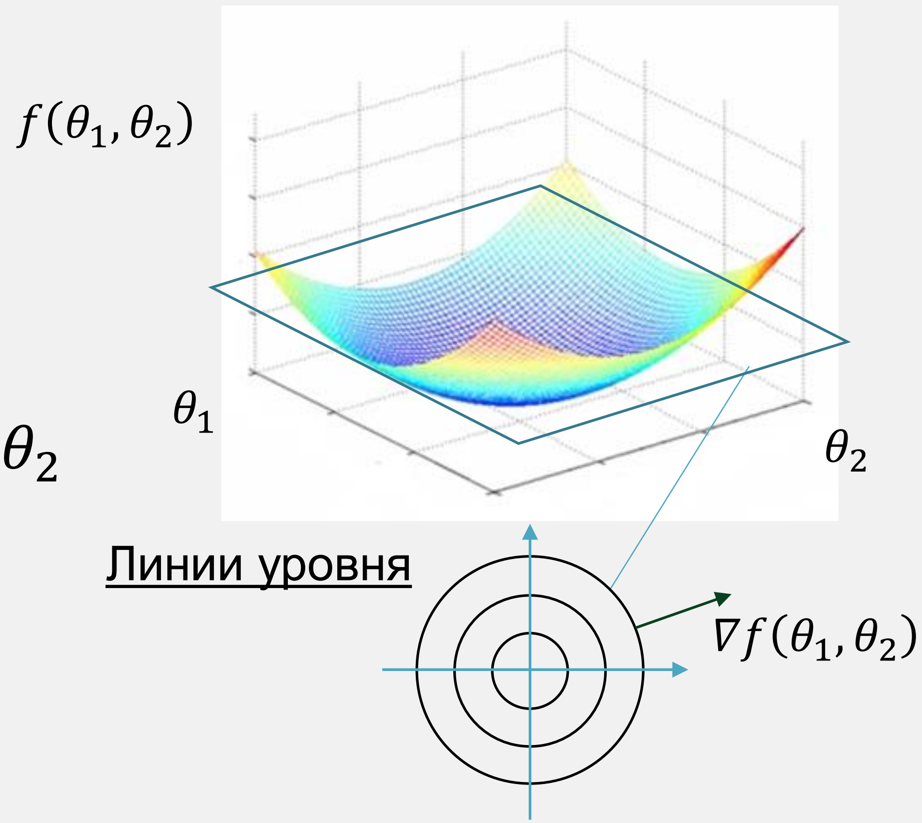
Функция:

$$f(\theta_1, \theta_2) = \ {\theta_1}^2 + {\theta_2}^2$$

Частные производные:

$$\frac{\delta f(\theta_1,\theta_2)}{\delta \theta_1} = 2\theta_1, \frac{\delta f(\theta_1,\theta_2)}{\delta \theta_2} = 2\theta_2$$

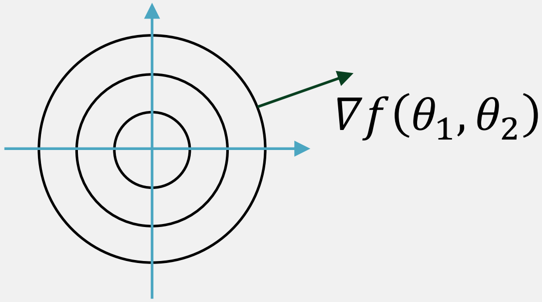
$$\nabla f(\theta_1, \theta_2) = \begin{bmatrix} 2\theta_1 \\ 2\theta_2 \end{bmatrix}$$



Нейронные сети

Метод градиентного спуска

$$\Theta_{k+1} = \Theta_k - \eta_k \boldsymbol{g_k} = \Theta_k - \eta_k \boldsymbol{\nabla f(\theta_k)}$$



Нейронные сети

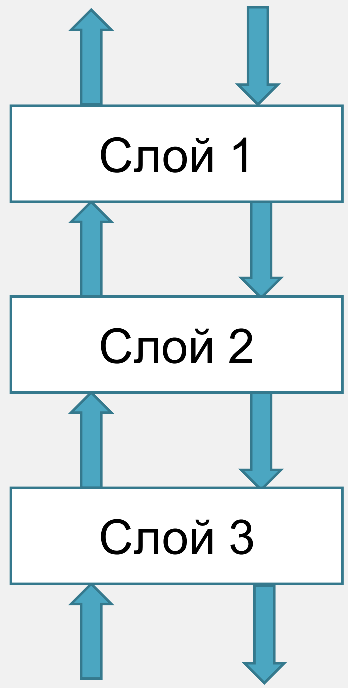
Метод обратного распространения ошибки

$$C = z^4 = f_3(z^3)$$

$$z^3 = f_2(z^2)$$

$$z^2 = f_1(z^1)$$

$$z^1 = X$$



δ^4

δ^3

δ^2

δ^1

$$\delta_i^L = \frac{\partial C}{\partial z_i^L} = \sum_j \frac{\partial C}{\partial z_j^{L+1}} \frac{\partial z_j^{L+1}}{\partial z_i^L} =$$

$$= \sum_j \delta_i^{L+1} \frac{\partial z_j^{L+1}}{\partial z_i^L}$$

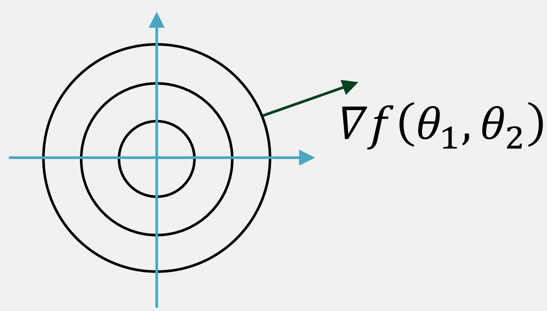
C – некоторая целевая функция
То, что мы хотим оптимизировать

Нейронные сети

Метод градиентного спуска

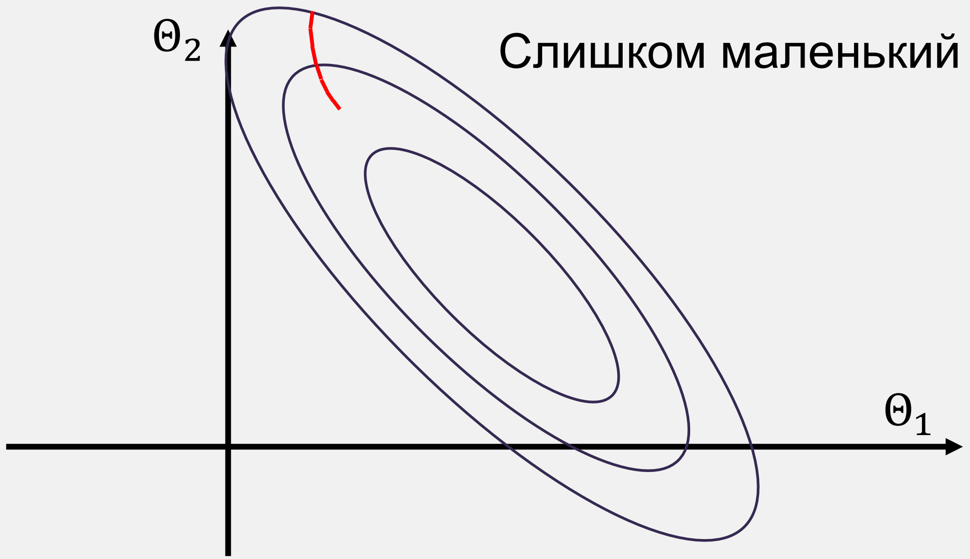
$$\Theta_{k+1} = \Theta_k - \eta_k \boldsymbol{g_k} = \Theta_k - \eta_k \boldsymbol{\nabla f(\theta_k)}$$

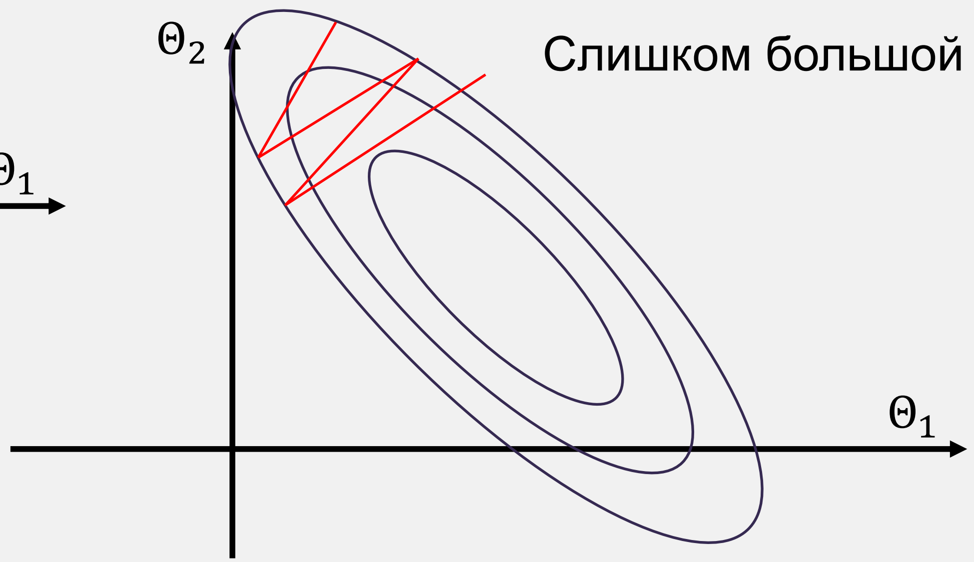
Коэффициент обучения
(learning rate)



Нейронные сети

Метод градиентного спуска





Метод Ньютона

Используем матрицу Гессе

$\Theta_{k+1} = \Theta_k - H_k^{-1} g_k$

$$f_{quad}(\Theta) = f(\Theta_k) + g_k^T(\Theta - \Theta_k) + \frac{1}{2}(\Theta - \Theta_k)^T H_k (\Theta - \Theta_k)$$

$$\frac{\partial}{\partial \Theta} f_{quad}(\Theta) = 0 + g_k^T + H_k(\Theta - \Theta_k) = 0$$

Нейронные сети

Типы обучения

1)Offline обучения

$\Theta_{k+1} = \Theta_k - \eta_k \sum_{i=1}^{n} X_i^T (y_i - X_i \Theta_k)$, где n – размер выборки

2) Online обучения

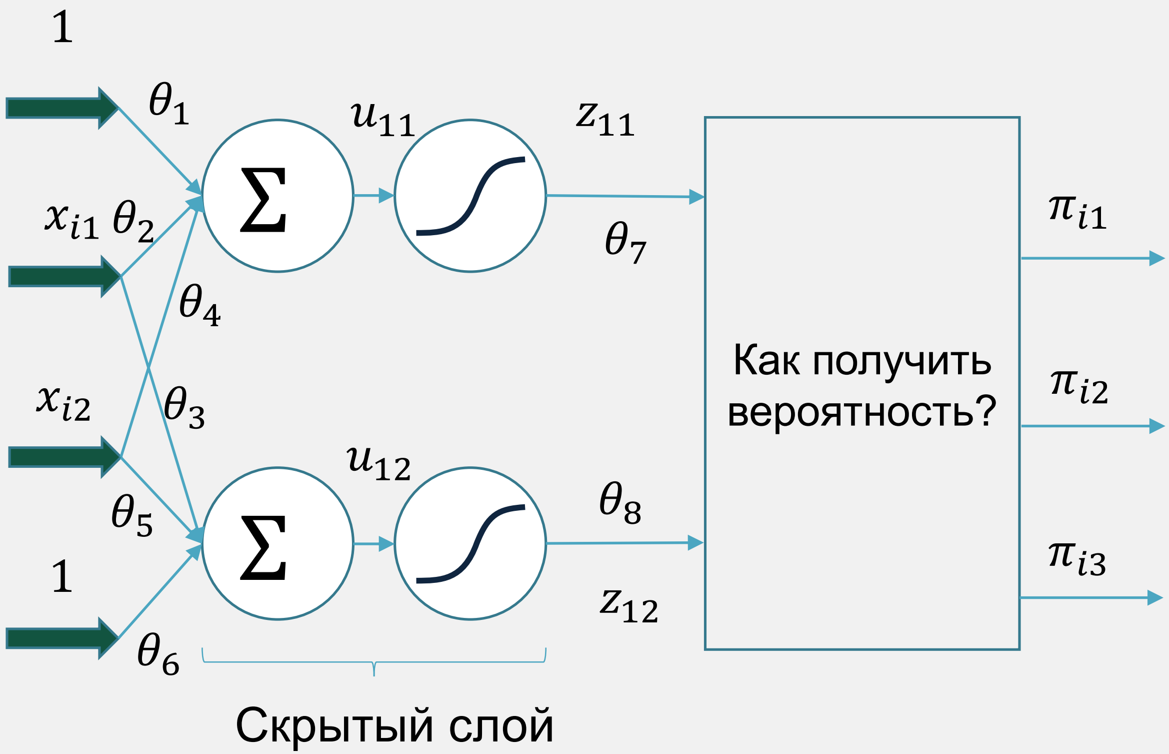
$\Theta_{k+1} = \Theta_k - \eta_k X_k^T (y_k - X_k \Theta_k)$

3) Обучение минибатчами

$\Theta_{k+1} = \Theta_k - \eta_k \sum_{j=1}^{20} X_j^T \left(y_j - X_j \Theta_k\right)$, где 20 – размер минибатча

Нейронные сети

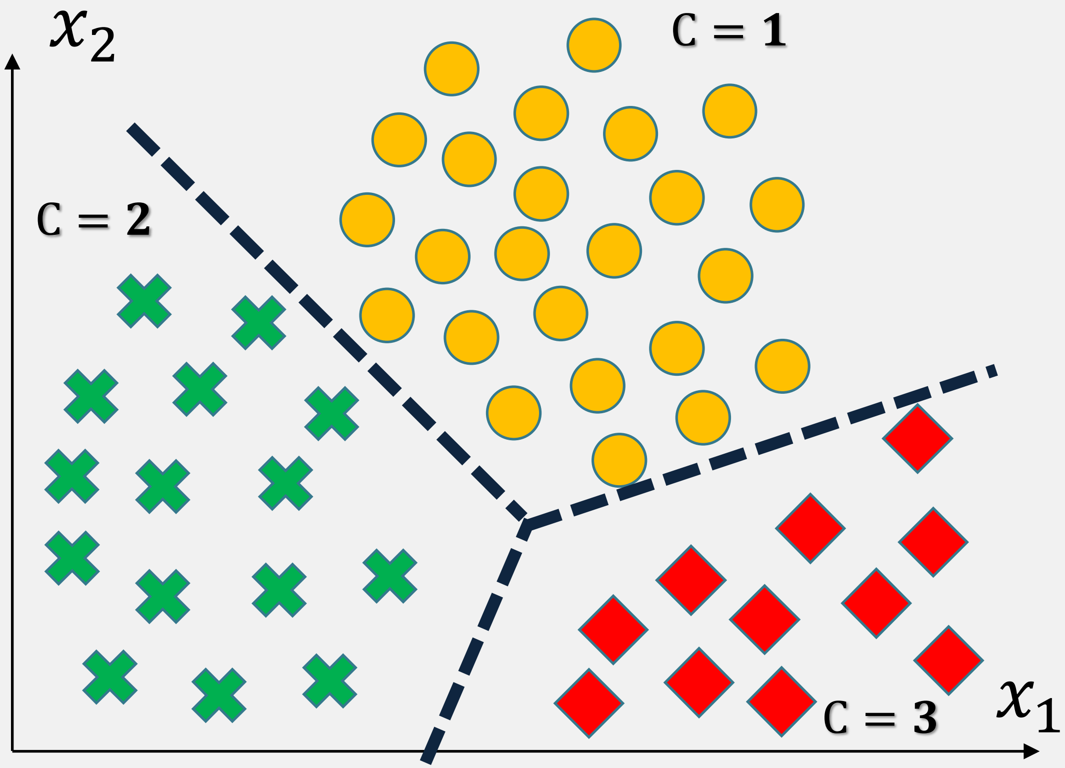
Многослойный персептрон (MLP), классификация



Сложение выходов сигмоид не гарантирует $\sum f = 1$

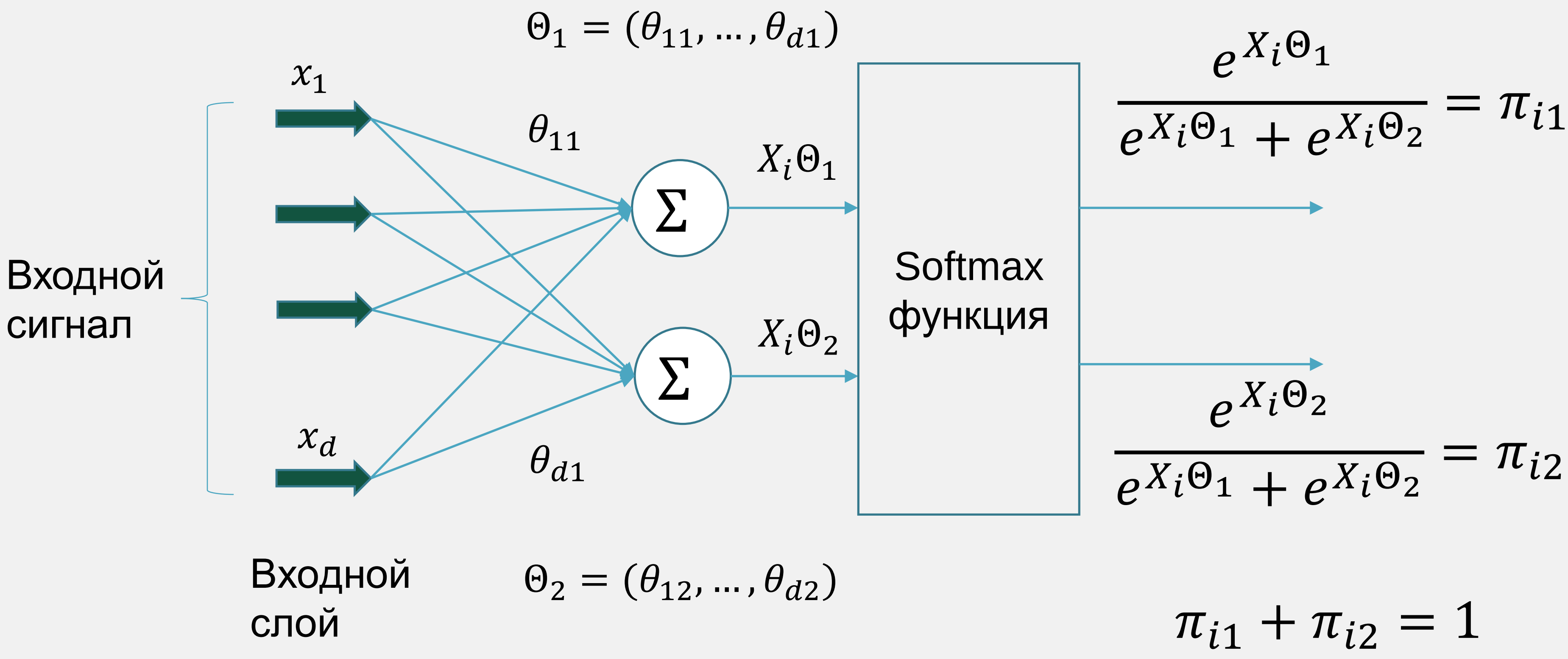
Нейронные сети

Многослойный персептрон (MLP) , классификация



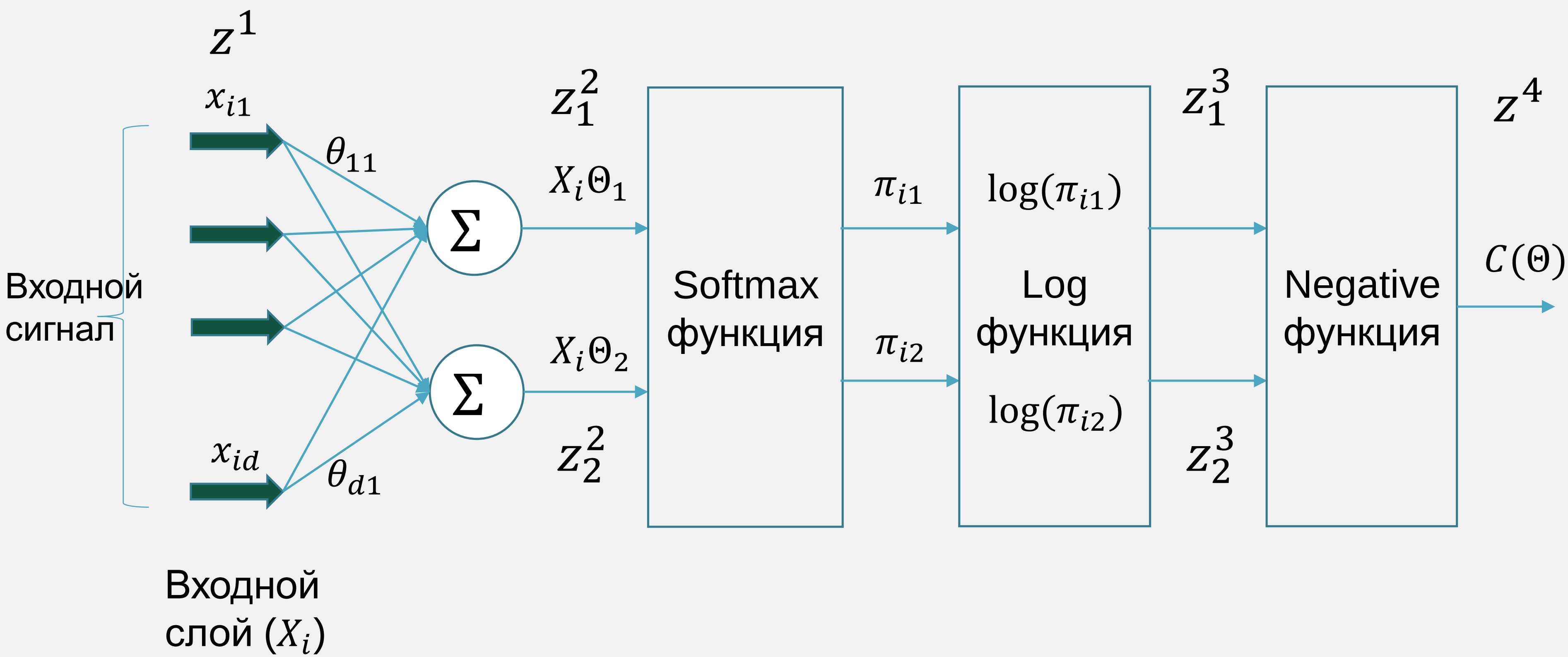
Нейронные сети

Softmax функция



Нейронные сети

Softmax функция



Нейронные сети

Softmax функция

$$C(\Theta) = z^4\{z_1^3\left[z_1^2(X_i\Theta), z_2^2(X_i\Theta)\right], z_2^3[z_1^2(X_i\Theta), z_2^2(X_i\Theta)]\}$$

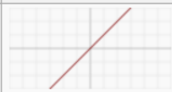
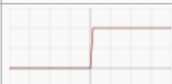
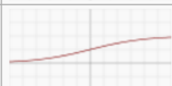
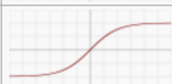
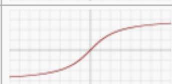
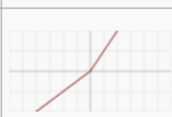
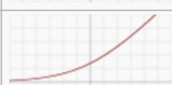
$$\frac{\partial C(\Theta)}{\partial \theta_{11}} = \frac{\partial z^4}{\partial z_1^3}\frac{\partial z_1^3}{\partial z_1^2}\frac{\partial z_1^2}{\partial \theta_{11}} + \frac{\partial z^4}{\partial z_1^3}\frac{\partial z_1^3}{\partial z_2^2}\frac{\partial z_2^2}{\partial \theta_{11}} + \frac{\partial z^4}{\partial z_2^3}\frac{\partial z_2^3}{\partial z_1^2}\frac{\partial z_1^2}{\partial \theta_{11}} + \frac{\partial z^4}{\partial z_2^3}\frac{\partial z_2^3}{\partial z_2^2}\frac{\partial z_2^2}{\partial \theta_{11}}$$

$$z = f(x(u,v), y(u,v))$$

$$\frac{\partial z}{\partial u} = \frac{\partial z}{\partial x}\frac{\partial x}{\partial u} + \frac{\partial z}{\partial y}\frac{\partial y}{\partial u}$$

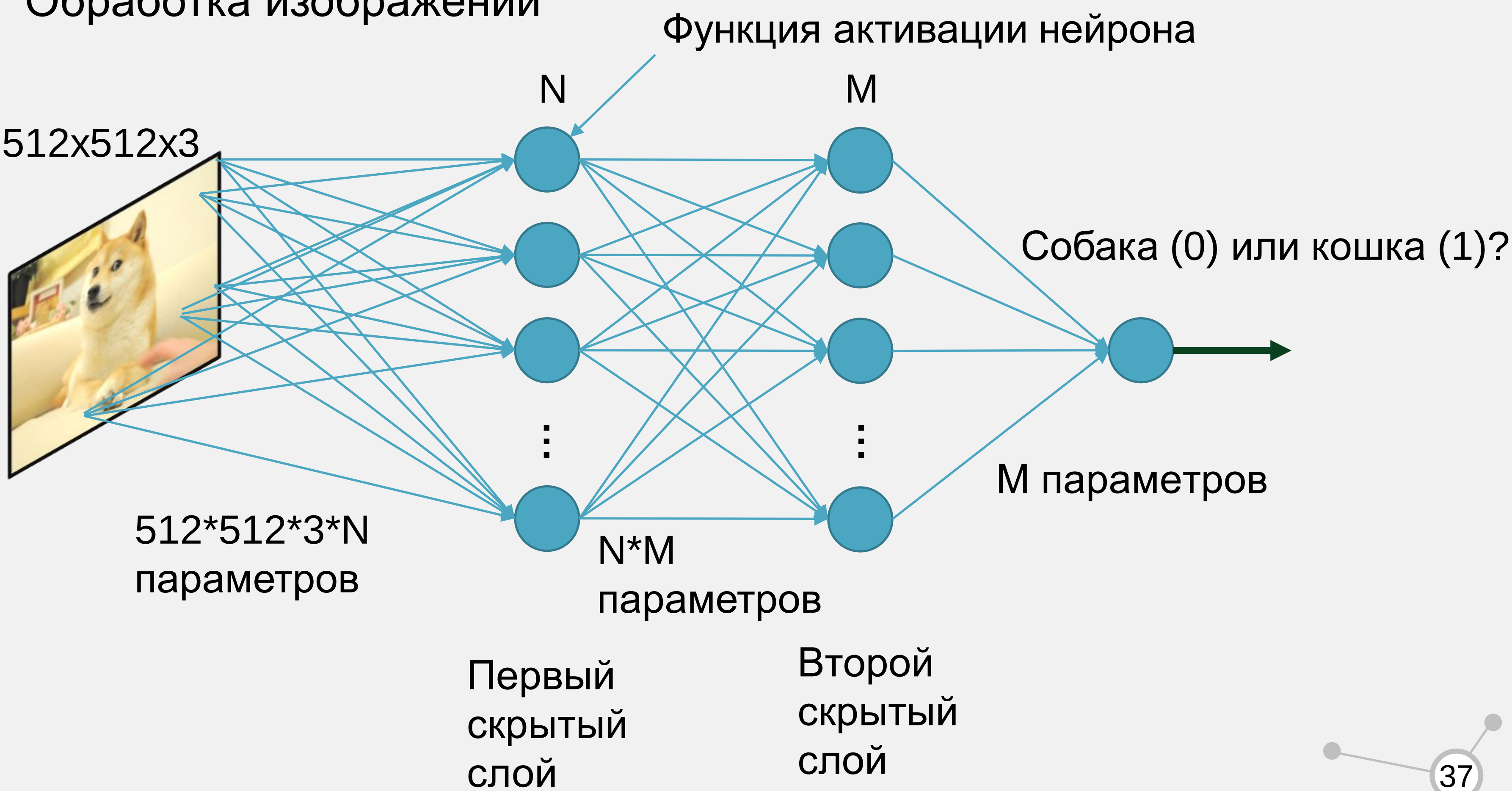
Нейронные сети

Функции активации

Name	Plot	Equation	Derivative
Identity		$f(x) = x$	$f'(x) = 1$
Binary step		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x \neq 0 \\ ? & \text{for } x = 0 \end{cases}$
Logistic (a.k.a Soft step)		$f(x) = \frac{1}{1+e^{-x}}$	$f'(x) = f(x)(1 - f(x))$
TanH		$f(x) = \tanh(x) = \frac{2}{1+e^{-2x}} - 1$	$f'(x) = 1 - f(x)^2$
ArcTan		$f(x) = \tan^{-1}(x)$	$f'(x) = \frac{1}{x^2+1}$
Rectified Linear Unit (ReLU)		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Parameteric Rectified Linear Unit (PReLU) [2]		$f(x) = \begin{cases} \alpha x & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Exponential Linear Unit (ELU) [3]		$f(x) = \begin{cases} \alpha(e^x - 1) & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} f(x) + \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
SoftPlus		$f(x) = \log_e(1 + e^x)$	$f'(x) = \frac{1}{1+e^{-x}}$

Многослойный персептрон (MLP)

Обработка изображений



Нейронные сети

Инструменты для создания нейронных сетей

PYTORCH

mxnet

theano

TensorFlow

Keras

Caffe2

Спасибо за внимание!

Спасёнов Алексей

a.spasenov@corp.mail.ru