МИСИС
УНИВЕРСИТЕТ

Глубокое обучение. Вводная лекция

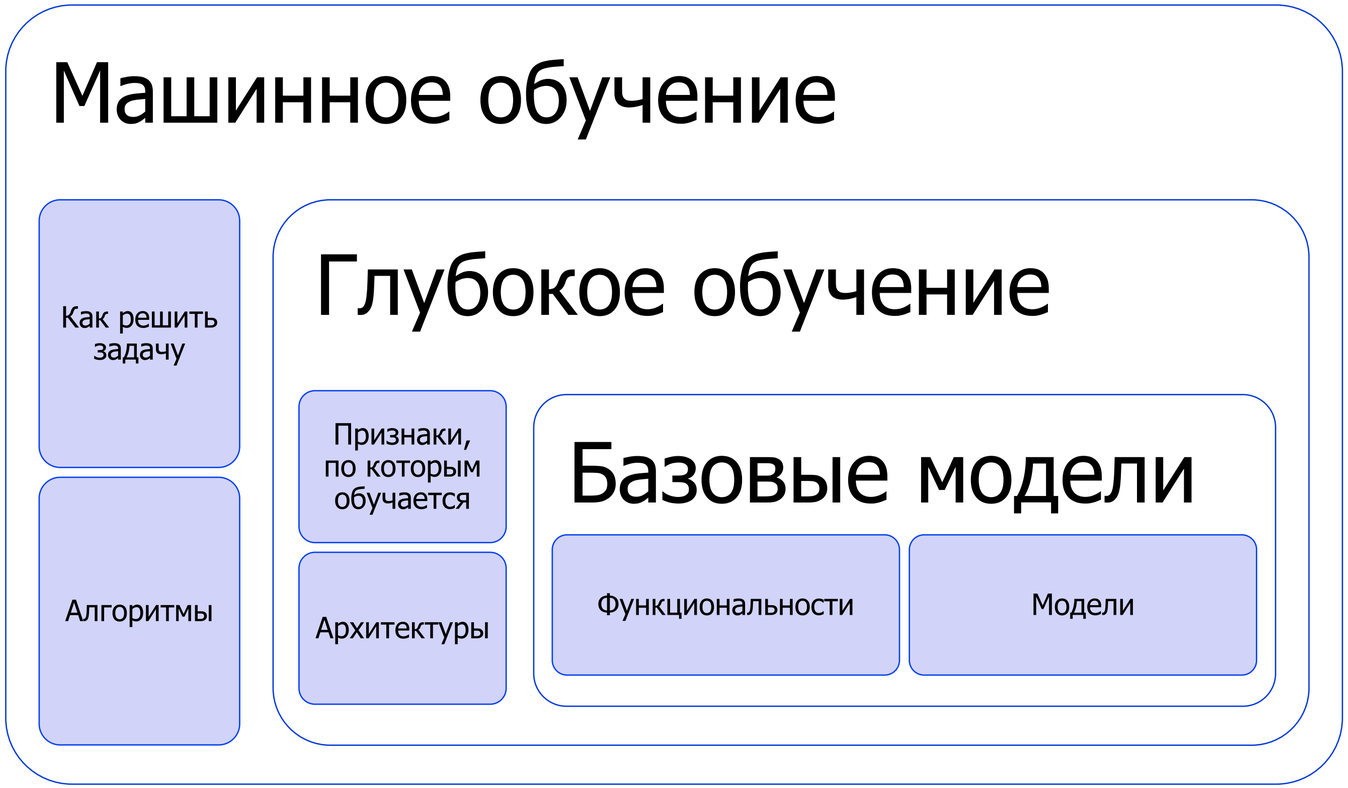
Корнет Мария Евгеньевна
старший преподаватель кафедры
Инженерная кибернетика

План

- Понятие глубокого обучения
- Что такое нейронная сеть:

- Основные компоненты нейронных сетей
- Вычислительные возможности
- Многослойная нейронная сеть

Технологии ИИ



универсализация

Глубокое обучение
Deep learning

Классическое ML Эти модели часто требуют ручной инженерии признаков и хорошо работают на табличных, не слишком больших по объёму данных.
Пример: SVM, Random Forest, градиентный бустинг (XGBoost, LightGBM), логистическая регрессия и др.

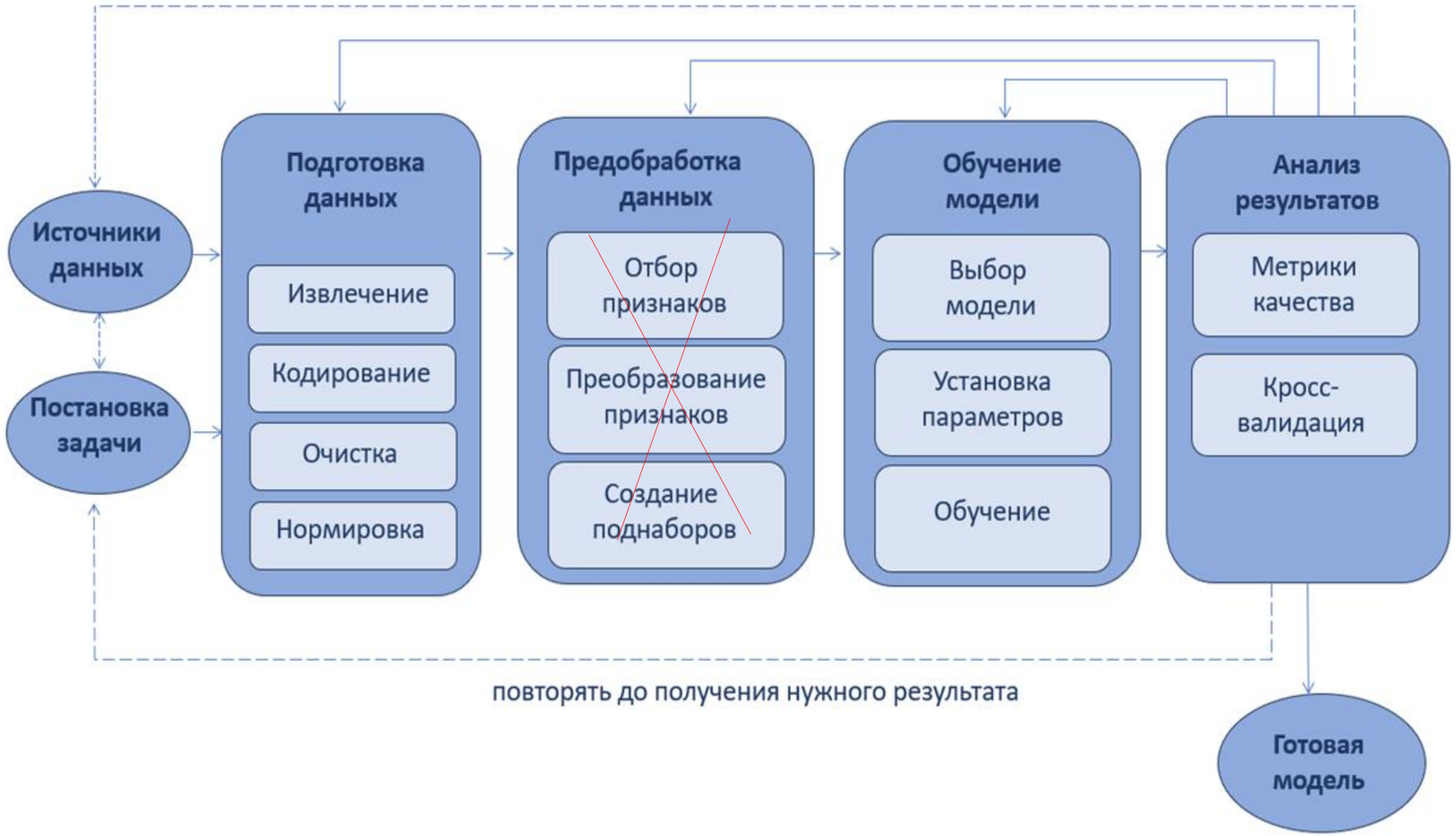
Глубокое обучение DL — область ML, использует искусственные многослойные нейронные сети для **извлечения информации из данных.**

Идея: убрать этап формального описания знаний человеком. Происходит автоматическое извлечение признаков из данных (текст, изображение, звук).

Общая схема машинного обучения

Общая схема глубокого обучения

В отличие от традиционного машинного обучения, где признаки создаются вручную, нейросети автоматически формируют их на основе наблюдений



DL vs ML

	Классическое ML	Глубокое обучение
Требование к объёму данных	Работает при малом числе примеров	Требует больших объёмов данных
Проектирование признаков	Feature engineering — обязательно	Признаки извлекаются автоматически
Скорость обучения	Быстрее на малых данных	Долго обучается, требует мощных GPU
Интерпретируемость	Высокая (например, в деревьях решений)	Низкая, модель — «чёрный ящик»
Вычислительные ресурсы	Небольшие (CPU достаточно)	Требуются GPU/TPU и большая память
Обобщающая способность	Хорошая при правильной настройке	Высокая, но склонна к переобучению

История развития

1943	Маккалок и Питтс	Первые формальные идеи нейрона как логической схемы
1958	Персептрон Розенблатта	Линейный классификатор, начало практического применения
1969	Работа Минского и Паперта	Показаны ограничения простых нейросетей, “XOR” и др.
1974	Александр Галушкин и Пол Вербос (обратное распространение ошибки)	Впервые предложен
1986	Backpropagation (Румельхарт–Хинтон–Уильямс)	Существенно развит метод обучения многослойных сетей
1989	LeNet (CNN)	Успех в распознавании цифр
2006	Deep Belief Networks (Hinton) возрождение DL	Появление pretraining, возможности обучения более глубоких моделей

История развития

2012	**AlexNet выигрывает ImageNet**	**Поворотный момент: DL показывает реальный прорыв в CV**
2014	Изобретение GAN, seq2seq	Новая ветвь генеративных моделей
2015	ResNet, очень глубокие CNN	Преодоление проблемы обучения очень глубоких сетей
2017	Transformer	Переход к вниманиям, масштабируемый подход в NLP
2018	BERT (Google)	Двусторонний контекст, лучшие представления в NLP
2020	GPT-3	Огромная языковая модель 175млрд параметров
2021-2022	CLIP, DALL·E, Stable Diffusion	Мультимодальные модели, генерация изображений по тексту
2023-2025	GPT-4-5	Модели мультисенсорные, обучение без потери памяти

Успехи DL

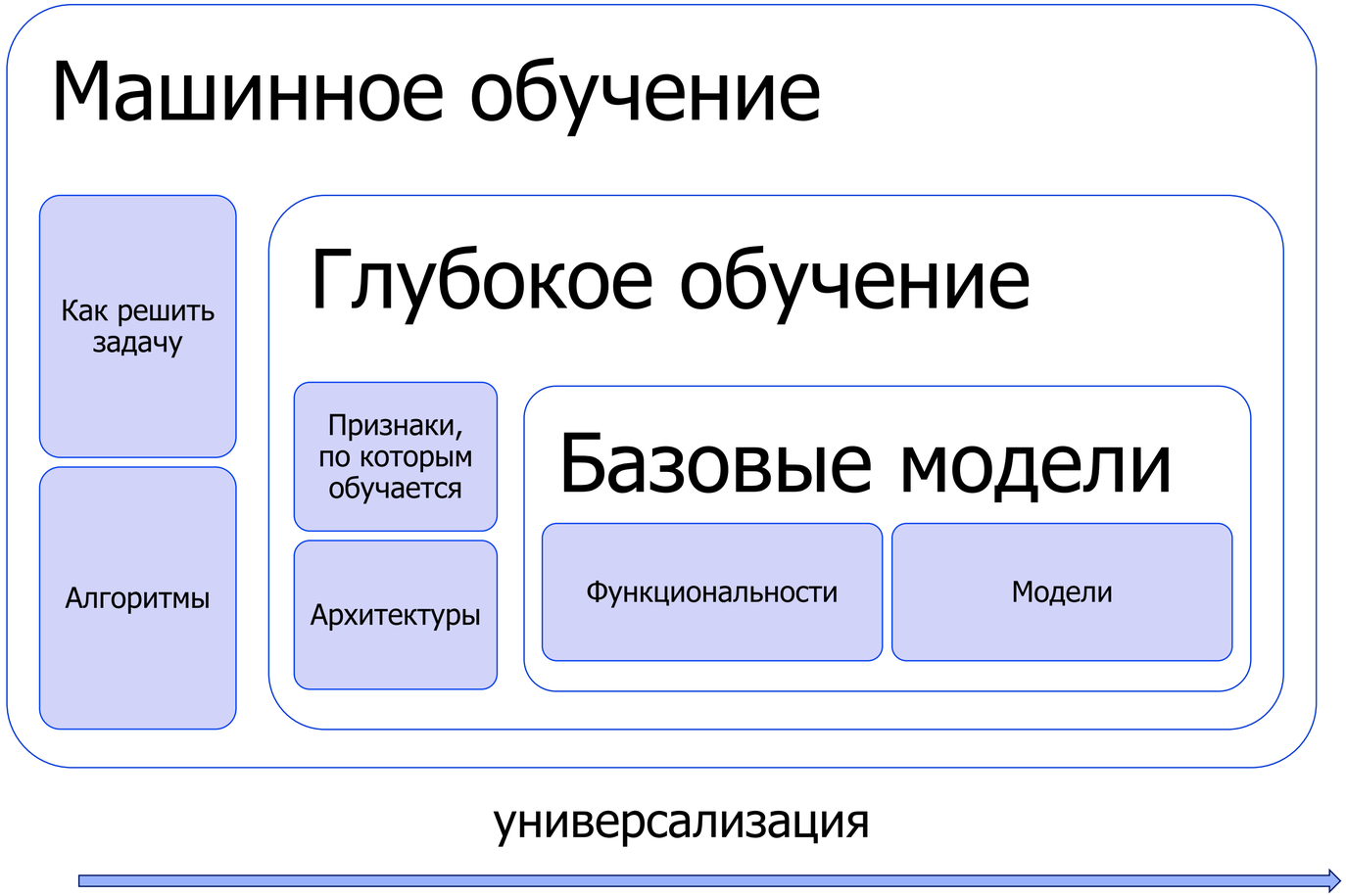
Задача	Подходы / архитектуры	Примеры успеха
Классификация изображений, детектирование объектов	CNN, сверточные сети, ResNet, Faster R-CNN, YOLO, SSD	Различение объектов на ImageNet; детекторы реального времени; распознавание лиц и объектов в видео
Обработка естественного языка: машинный перевод, языковое моделирование	RNNs, LSTM, GRU трансформеры	Переводные системы; GPT-, BERT-семейство; автодополнение текста
Речевые технологии: преобразование речи в текст, и обратно	Сверточные и рекуррентные сети, трансформеры, end-to-end модели	ASR системы; синтез речи, голосовые помощники

Успехи DL

Задача	Подходы / архитектуры	Примеры успеха
Генерация изображений и художественных данных	GANы, VAE, Flow-модели, diffusion-модели	Создание реалистичных изображений; style transfer; генерация лиц
Управление / игры	Deep RL (DQN, PPO, A3C и др.)	Агент, побеждающий человека в играх (Atari, Go, шахматы); управление роботами в симуляции

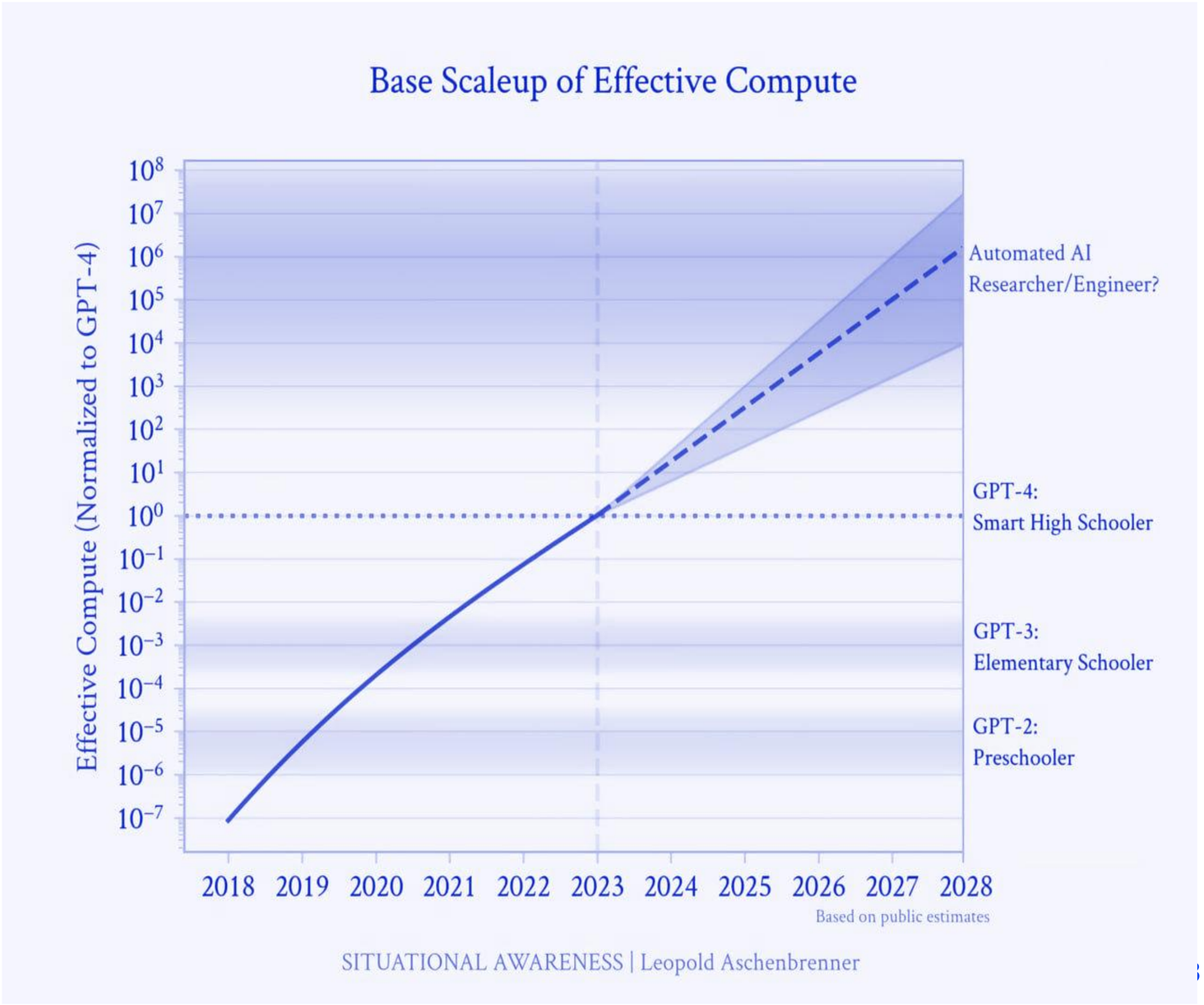
Глубокое обучение н.в.

Термин *foundation models (базовые модели)* был впервые предложен в 2021 году в исследовательском центре **Stanford Institute for Human-Centered Artificial Intelligence (HAI)** в докладе *On the Opportunities and Risks of Foundation Models* (Bommasani, 2021)



Перспективы развития сильного ИИ

Leopold Aschenbrenner (Open AI)
“Situational Awareness”



https://situational-awareness.ai/wp-content/uploads/2024/06/situationalawareness.pdf

Глубокое обучение вызовы

1. Необходимость больших размеченных данных
2. Обработка длинных контекстов и зависимостей
3. Обобщение и устойчивость
4. Интерпретируемость и объяснимость
5. Сохранение в модели полученных знаний (*Continual learning*)
6. Эффективность: вычислительные ресурсы, энергопотребление, время обучения (подход модели *Chinchilla* 2022, 70 млрд параметров)
7. Безопасность, этика

Распознавание речи

LibriSpeech — ~1000 часов английского (16kHz), стандарт для ASR

Mozilla Common Voice — краудсорс, многоязычный, 30k+ часов в актуальных версиях

Компьютерное зрение

ImageNet (ILSVRC) — 1.2M изображений/1000 классов

MS COCO — 328k изображений, 2.5M аннотаций (детекция, сегментация)

LAION-5B (мультимодальный) — 5.85 млрд пар «изображение-текст», открыт

Синтетические:
– **SynthText** (текст в сценах),
– **FlyingChairs**/Things3D (оптический поток)

Обработка ЕЯ

C4 (Colossal Clean Crawled) — очищенный, основа модели T5

The Pile — 825 GiB SQuAD — 100k+ QA (содержит академические тексты: PubMed, arXiv.
Книги: Bibliotik, Project Gutenberg.
Код: GitHub.
Юридические документы)

MS MARCO — крупные QA

WMT (ежегодные наборы и оценки для MT; с 2024 — General MT): человеческое ранжирование систем

Код
HumanEval — синтаксические Python-задачи на функциональную корректность

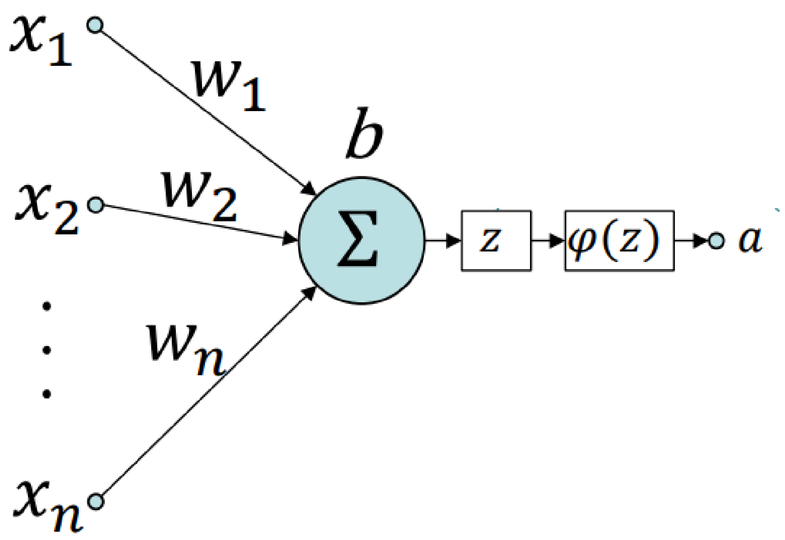
Фреймворки

PyTorch —компиляция (torch.compile, Dynamo/Inductor), ускорения без изменения модели

TensorFlow/Keras — гибкие high-level API, продакшен-инструменты (TFX, Serving), поддержка

JAX/Flax — vmap-веторизация/pmap-параллелизаия, функциональный стиль; активно используется для масштабного обучения и исследовательских библиотек

Математическая модель нейрона



1943г МакКаллок – Питтс, расширение модели 1958г.

$\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ – входные сигналы: $x_i \in \{0,1\}$

$\boldsymbol{w} = (w_1, w_2, \ldots, w_n)$ – синаптические веса: $w_i \in \mathbb{R}$

b – смещение: $b \in \mathbb{R}$

$\varphi: \mathbb{R} \to \mathbb{R}$ – функция активации

a – выходной сигнал: $a \in \{0,1\}$

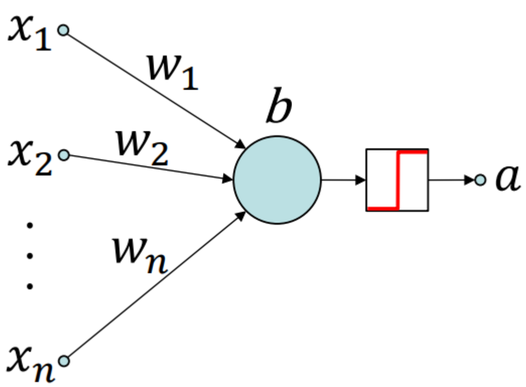
$$a = \begin{cases} 0, \text{если } \varphi(z) \leq 0 \\ 1, \text{если} \varphi(z) > 0 \end{cases}$$

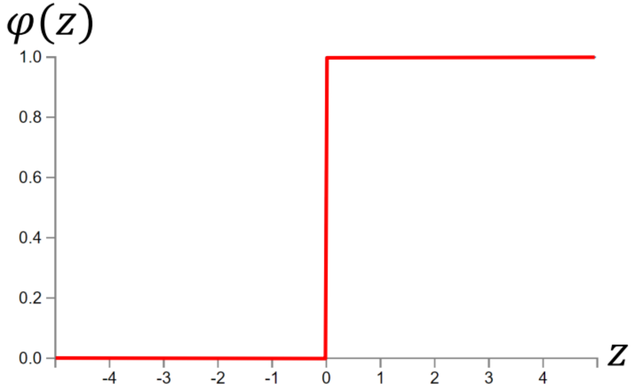
z - активационный потенциал:

$$z = b + \sum_{i=1}^{n} w_i x_i$$

Персептрон Розенблатта

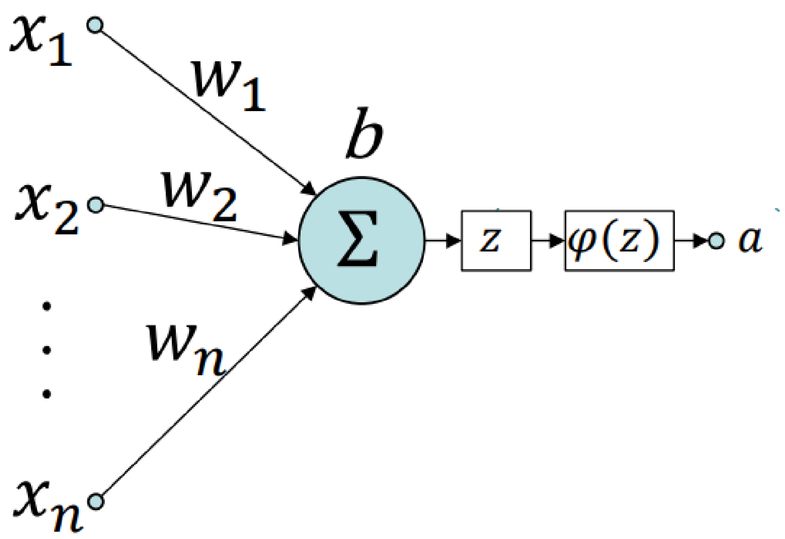
1957г Розенблатт





$$\varphi(z) = \begin{cases} 0, \text{если } z \leq 0 \\ 1, \text{если } z > 0 \end{cases}$$

Нейрон – линейная модель



Линейная регрессия:

$$a(x) = b + \sum_{i=1}^{n} w_i x_i$$

Линейная классификация:

$$a(x) = th(b + \sum_{i=1}^{n} w_i x_i)$$

Логистическая регрессия:

$$a(x) = \sigma(b + \sum_{i=1}^{n} w_i x_i)$$

Функция активации нейрона

Сигмоида

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

Гиперболический тангенс

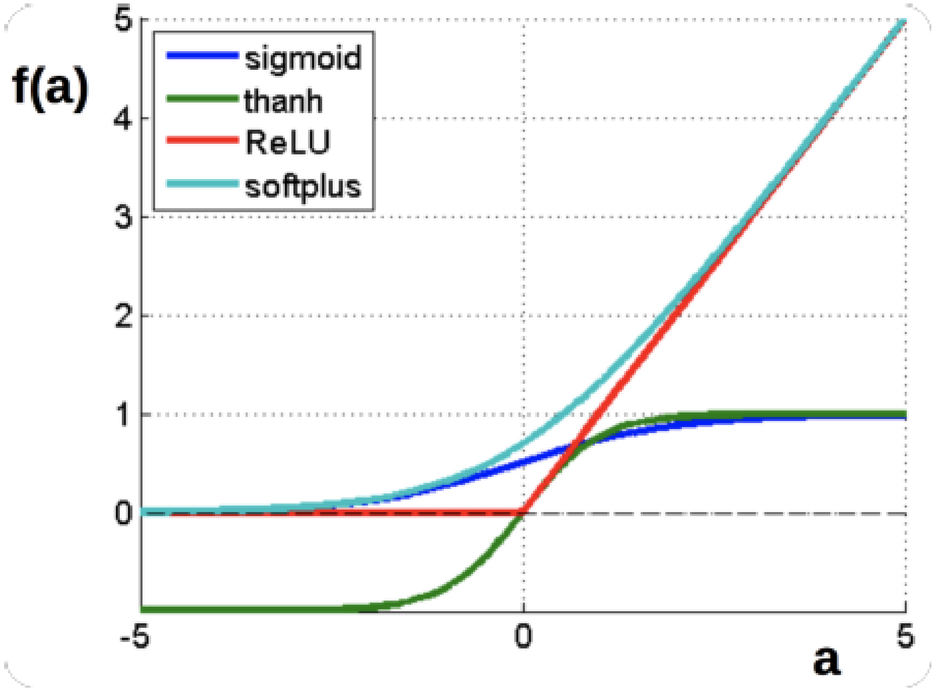
$$f(a) = \tanh(a)$$

ReLu (Rectified Linear Unit)

$$f(a) = \max(0, a)$$

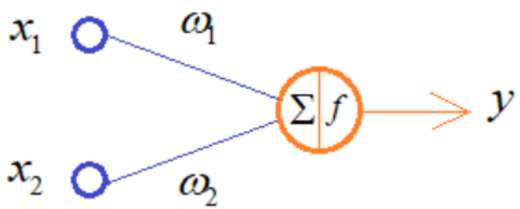
Softplus

$$f(a) = \log(1 + e^{a})$$



Функция активации определяет, будет ли нейрон активирован и какой будет его выходной сигнал

Вычислительная мощность персептрона



$$f(x) = \begin{cases} 1, \text{если } x \geq 0 & \to C_1 \\ -1, \text{если } x < 0 & \to C_2 \end{cases}$$

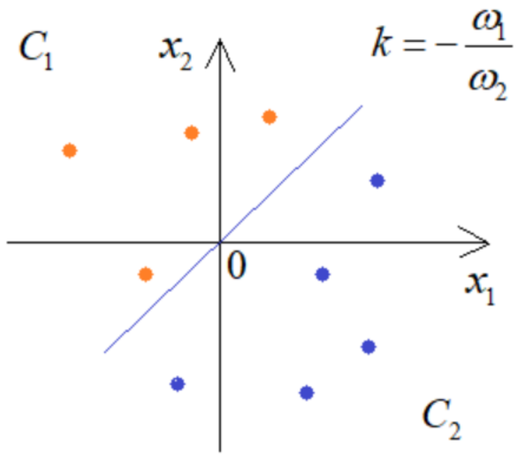
или

$$\begin{cases} w_1 x_1 + w_2 x_2 \geq 0 & \to C_1 \\ w_1 x_1 + w_2 x_2 < 0 & \to C_2 \end{cases}$$

$$w_1 x_1 + w_2 x_2 = 0$$

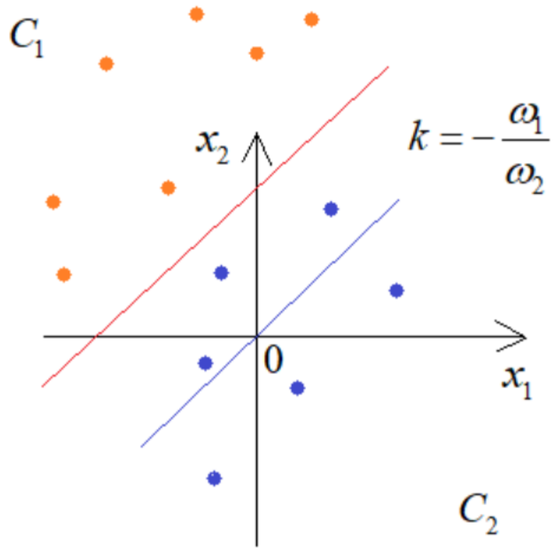
$$x_2 = -\frac{w_1}{w_2} x_1$$

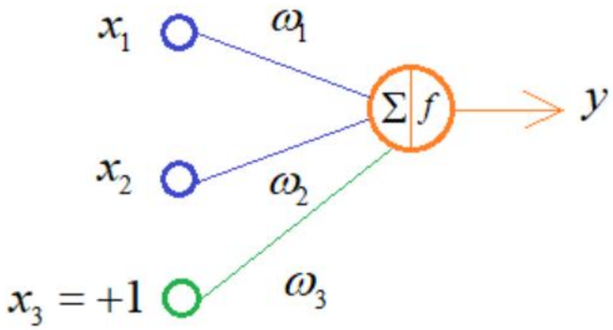
$$k = -\frac{w_1}{w_2}$$



https://proproprogs.ru/neural_network/perseptron-vozmozhnosti-klassifikacii-obrazov-zadacha-xor

Вычислительная мощность персептрона





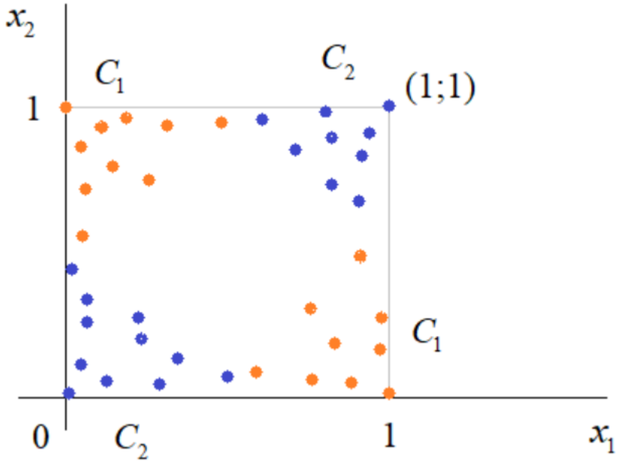
b (bias) - смещение

$$x_2 = -\frac{w_1}{w_2}x_1 - \frac{w_3}{w_2}x_3$$

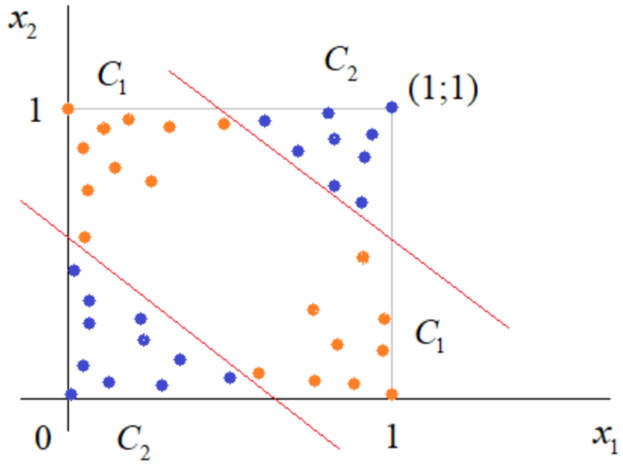
$$-\frac{w_3}{w_2} = b, \qquad \text{то } w_3 = -bw_2$$

! смещение необходимо для настройки нейрона

Вычислительная мощность персептрона (задача XOR)



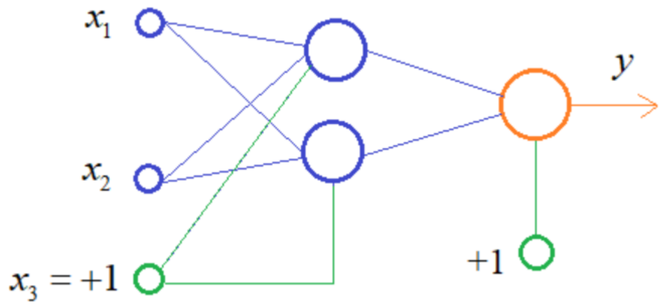
0	0	C_2
0	1	C_1
1	0	C_1
1	1	C_2



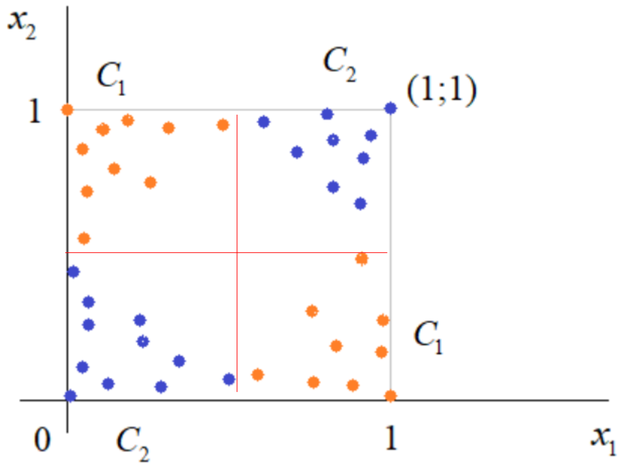
1 СПОСОБ:
каждая разделяющая линия может быть смоделирована отдельным нейроном

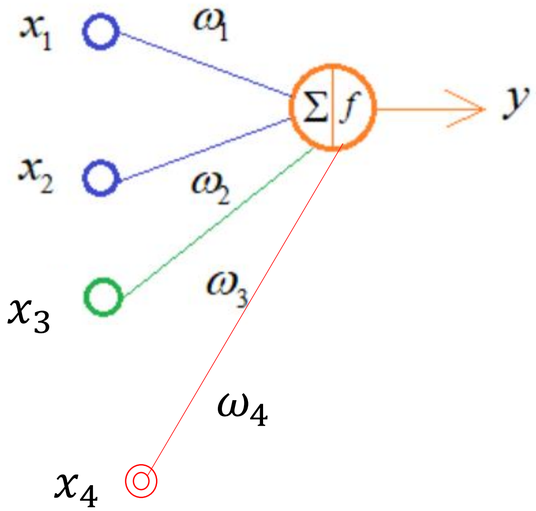
$[x_1, x_2] \in [0, 1]$

$$f(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases}$$



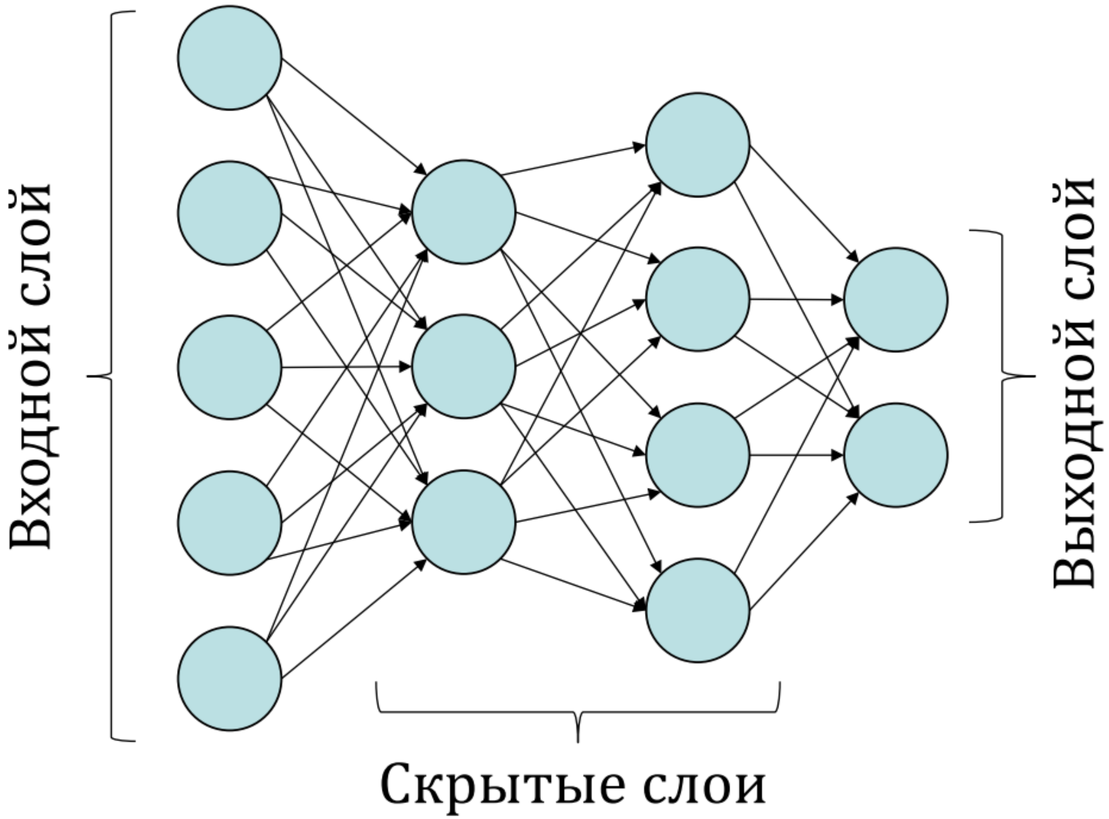
Вычислительная мощность персептрона (задача XOR)





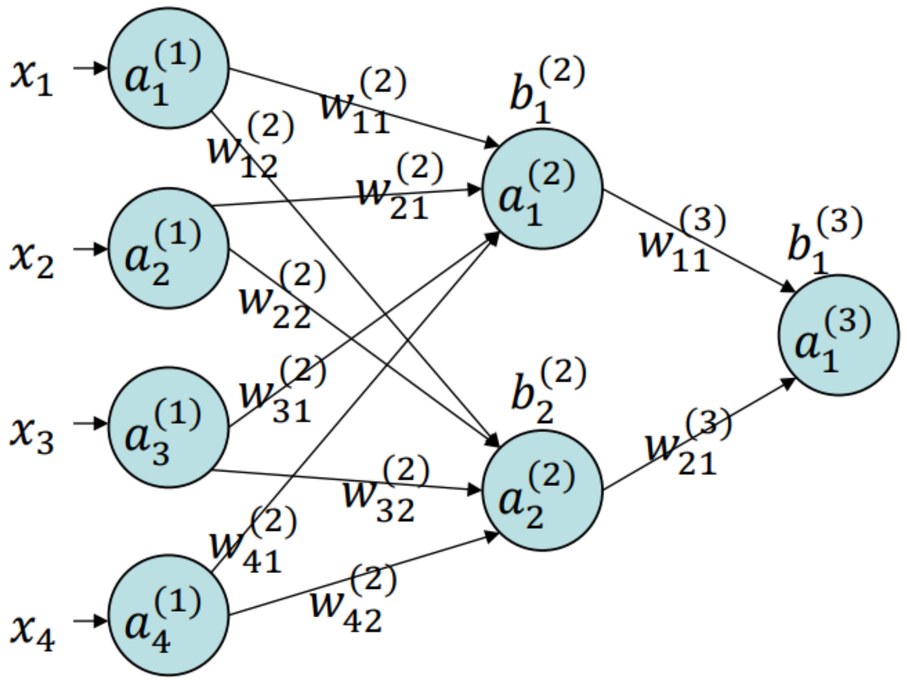
2 СПОСОБ:
добавить еще один вход – нелинейный признак

Полносвязная нейронная сеть



Full Connected Neural Network

Вычисление сигнала



$$a_j^{(1)} = x_j \ (j = 1, \dots, 4)$$

$$z_1^{(2)} = w_{11}^{(2)} a_1^{(1)} + w_{21}^{(2)} a_2^{(1)} + w_{31}^{(2)} a_3^{(1)} + w_{41}^{(2)} a_4^{(1)} + b_1^{(2)}$$

$$a_1^{(2)} = \begin{cases} 0, \text{если } z_1^{(2)} \le 0 \\ 1, \text{если } z_1^{(2)} > 0 \end{cases}$$

$$z_2^{(2)} = w_{21}^{(2)} a_1^{(1)} + w_{22}^{(2)} a_2^{(1)} + w_{23}^{(2)} a_3^{(1)} + w_{24}^{(2)} a_4^{(1)} + b_2^{(2)}$$

$$a_2^{(2)} = \begin{cases} 0, \text{если } z_2^{(2)} \le 0 \\ 1, \text{если } z_2^{(2)} > 0 \end{cases}$$

$$z_1^{(3)} = w_{11}^{(3)} a_1^{(2)} + w_{12}^{(3)} a_2^{(2)} + b_1^{(3)}$$

$$a_1^{(3)} = \begin{cases} 0, \text{если } z_1^{(3)} \le 0 \\ 1, \text{если } z_1^{(3)} > 0 \end{cases}$$

Универсальная аппроксимирующая теорема

Теорема Цыбенко (1989г.) : *Любую непрерывную функцию можно с любой точностью приблизить двухслойной нейросетью с сигмоидной функцией активации во внутреннем слое и линейной в выходном*

- Двухслойная сеть в $\{0,1\}^n$ позволяет реализовать произвольную булеву функцию
- Двухслойная сеть в $\mathbb{R}^n$ позволяет отделить произвольный выпуклый многогранник.
- С помощью линейных операций и одной нелинейной *функции активации* σ можно приблизить любую непрерывную функцию с любой желаемой точностью.

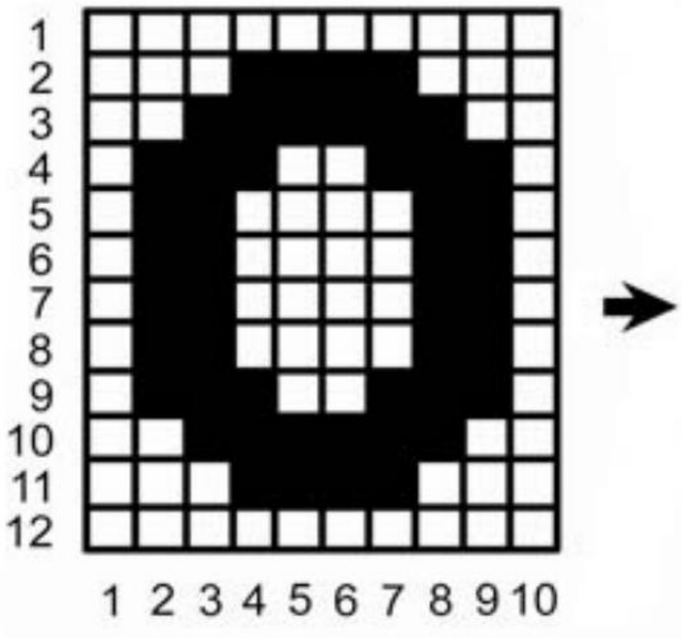
Распознавание рукописных изображений

MNIST – Mixed National Institute of Standards and Technology

База данных MNIST включает 60 000 тренировочных экземпляров изображений цифр и 10 000 проверочных изображений

На 2023г. средняя ошибка распознавания нейронной сети составляет 1,75%

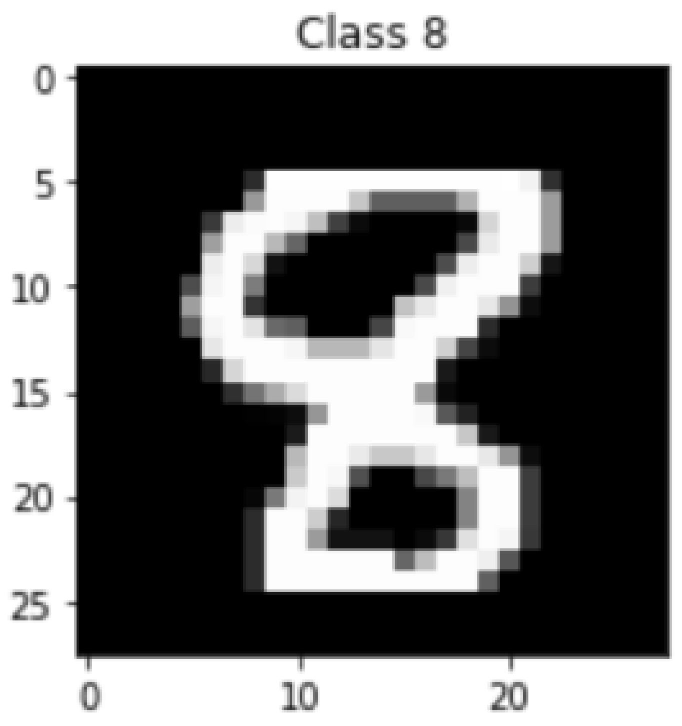
Распознавание рукописных изображений



										i
0	0	0	0	0	0	0	0	0	0	1
0	0	0	1	1	1	1	0	0	0	2
0	0	1	1	1	1	1	1	0	0	3
0	1	1	1	0	0	1	1	1	0	4
0	1	1	0	0	0	0	1	1	0	5
0	1	1	0	0	0	0	1	1	0	6
0	1	1	0	0	0	0	1	1	0	7
0	1	1	0	0	0	0	1	1	0	8
0	1	1	1	0	0	1	1	1	0	9
0	0	1	1	1	1	1	1	0	0	10
0	0	0	1	1	1	1	0	0	0	11
0	0	0	0	0	0	0	0	0	0	12
j	1	2	3	4	5	6	7	8	9	10

hhttps://sok.susu.ru/courses/MachineLearnig/lectures/02%20Model.pdf

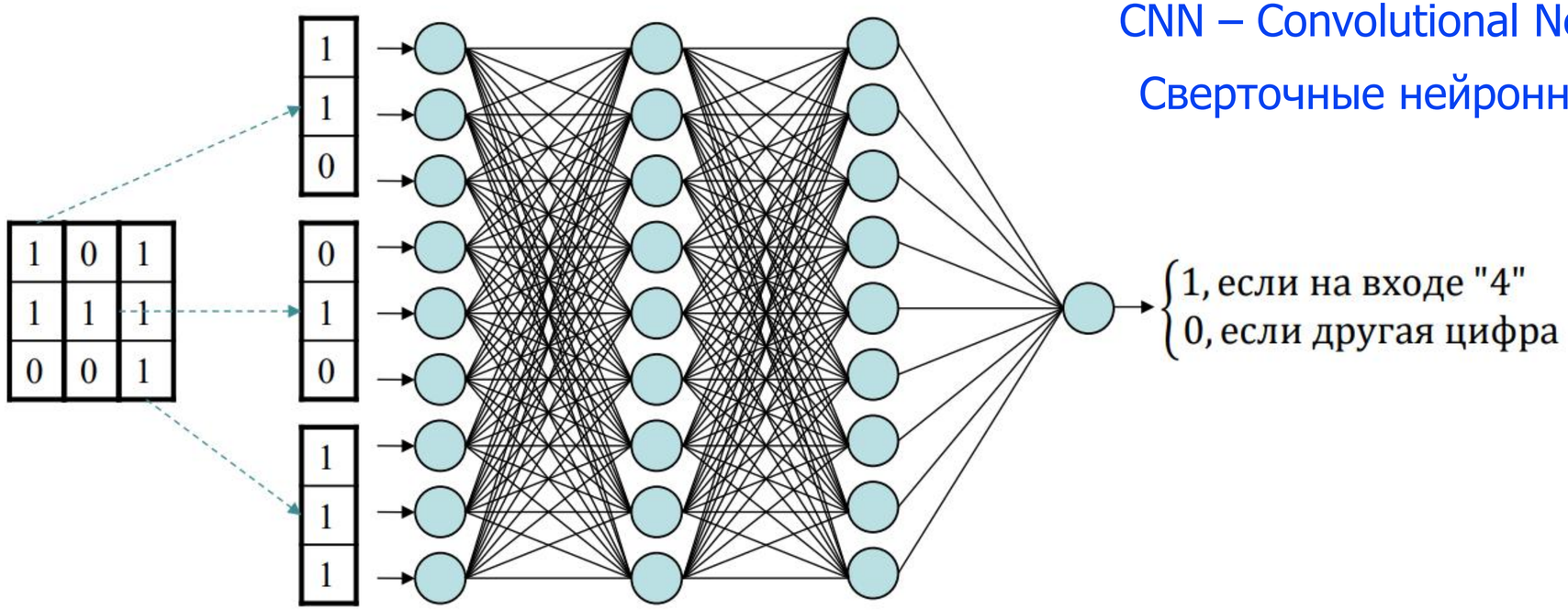
Распознавание рукописных изображений



0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	45	254	254	254	254	254	254	254	254	254	254	254	255	242	49	0	0	0	0	0
0	0	0	0	0	0	0	0	153	253	253	253	253	198	95	95	95	95	177	253	253	253	156	0	0	0	0	0
0	0	0	0	0	0	58	237	251	253	246	191	64	11	0	0	0	0	9	213	253	253	156	0	0	0	0	0
0	0	0	0	0	0	159	253	253	184	101	0	0	0	0	0	0	0	74	234	253	253	156	0	0	0	0	0
0	0	0	0	0	0	237	253	215	18	0	0	0	0	0	0	0	77	237	253	253	207	21	0	0	0	0	0
0	0	0	0	0	77	245	253	124	0	0	0	0	0	0	0	73	240	253	253	253	60	0	0	0	0	0	0
0	0	0	0	0	158	253	253	52	0	0	0	0	0	0	196	232	253	253	231	147	13	0	0	0	0	0	0
0	0	0	0	0	93	246	253	227	106	95	0	0	0	70	249	253	253	253	43	0	0	0	0	0	0	0	0
0	0	0	0	0	0	231	253	253	253	246	184	184	184	229	253	253	208	69	12	0	0	0	0	0	0	0	0
0	0	0	0	0	0	41	214	253	253	253	253	253	253	253	253	253	25	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	48	116	172	219	253	253	253	253	253	154	9	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	2	4	13	150	253	253	253	253	249	88	34	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	27	253	253	253	253	253	253	253	199	25	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	185	253	253	234	200	200	231	253	253	232	149	13	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	202	253	248	80	0	0	72	122	190	253	253	60	0	0	0	0	0	0
0	0	0	0	0	0	0	0	7	121	244	253	220	0	0	0	0	0	131	253	253	60	0	0	0	0	0	0
0	0	0	0	0	0	0	0	44	253	253	205	38	0	0	0	0	0	131	253	253	60	0	0	0	0	0	0
0	0	0	0	0	0	0	0	44	253	253	155	18	18	18	1	11	55	224	253	245	56	0	0	0	0	0	0
0	0	0	0	0	0	0	0	44	253	253	253	253	253	253	102	189	253	253	239	86	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	44	253	253	253	253	253	253	253	253	253	253	96	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

hhttps://sok.susu.ru/courses/MachineLearnig/lectures/02%20Model.pdf

Распознавание рукописных изображений

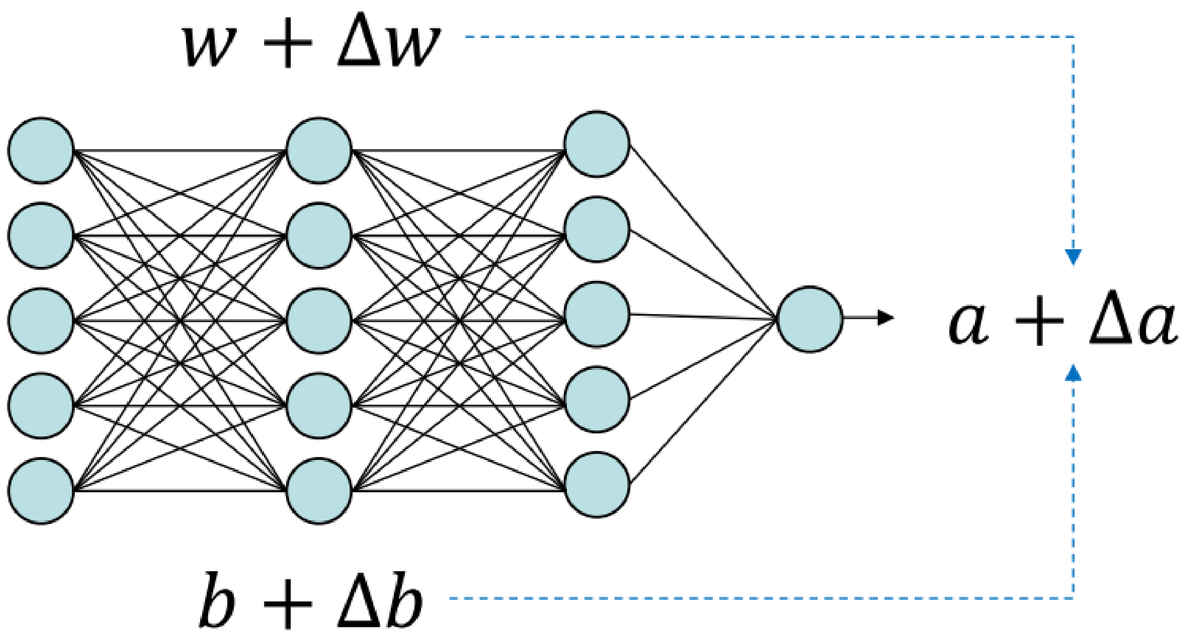


CNN – Convolutional Neural Network

Сверточные нейронные сети

hhttps://sok.susu.ru/courses/MachineLearnig/lectures/02%20Model.pdf

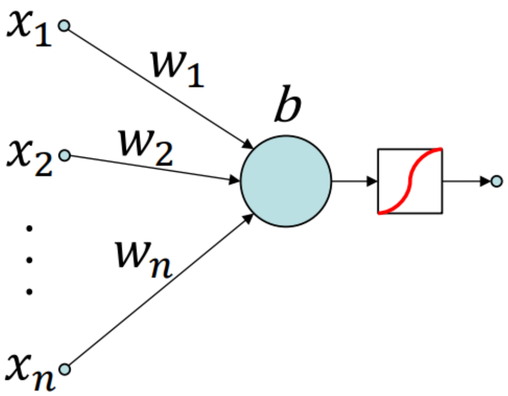
Обучение нейронной сети



Обучение –

подбор синаптических весов и смещений

Сигмоид

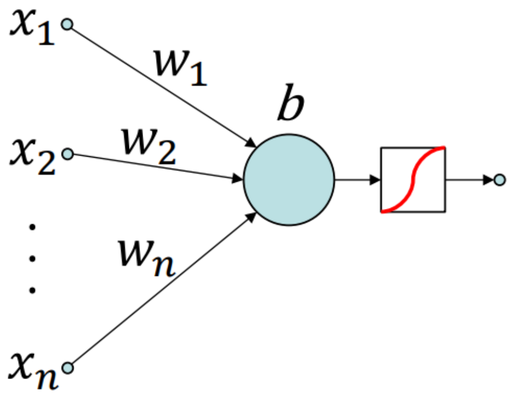


$$a = \frac{1}{1 + e^{-b-\sum_{i=1}^{n} w_i x_i}}$$

Функция активации - сигмоид

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

Возможность обучения



$$a = \frac{1}{1 + e^{-b-\sum_{i=1}^{n} w_i x_i}}$$

Частная производная a по b | Малое приращение b | Частная производная a по w_i | Малое приращение w_i

$$\Delta a \approx \frac{\partial a}{\partial b}\Delta b + \sum_{i=1}^{n} \frac{\partial a}{\partial w_i}\Delta w_i$$

Сформулировать критерий качества

Свести задачу к оптимизации критерия методом градиентного спуска (возможно если малое изменение веса (или смещения) приводит к малому изменению выходного сигнала)

Функции потерь

Для регрессии, например MSE:

$$L(\mathbf{y}, \hat{\mathbf{y}}) = (\mathbf{y} - \hat{\mathbf{y}})^2.mean()$$

Для классификации, кросс энтропия
или LogLoss:

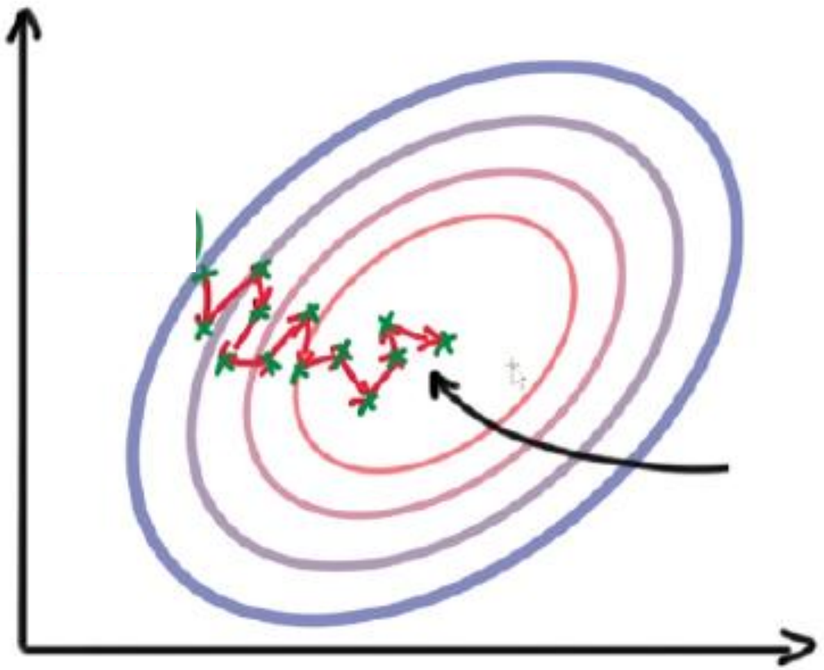
$$LogLoss = -(y_i log(\hat{y}_i) + (1 - y_i) log(1 - \hat{y}_i))$$

$$CE = -\sum_i y_i \log \hat{y}_i$$

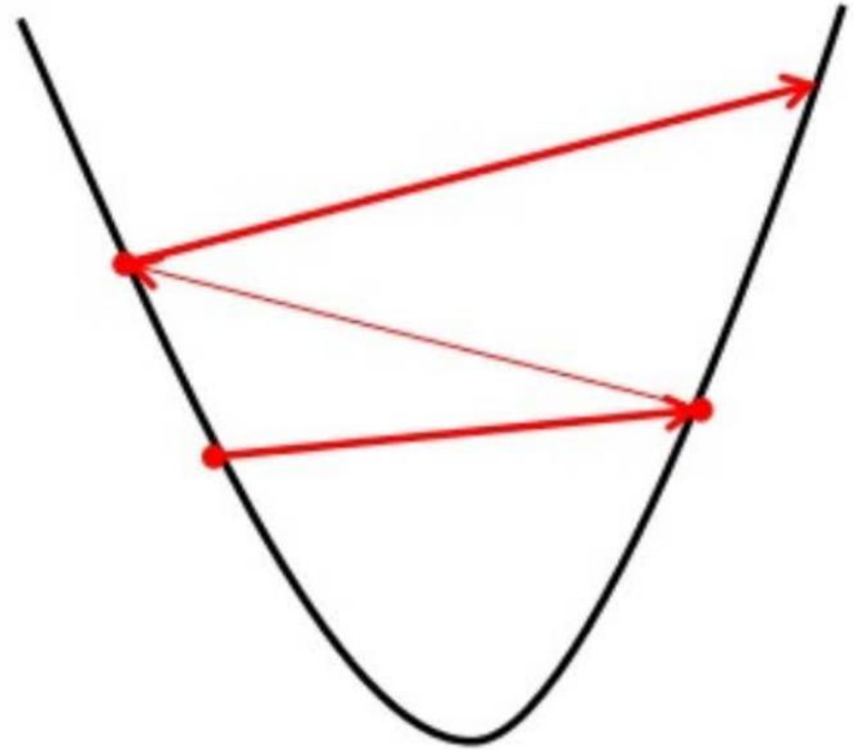
Любая дифференцируемая функция

Обучение

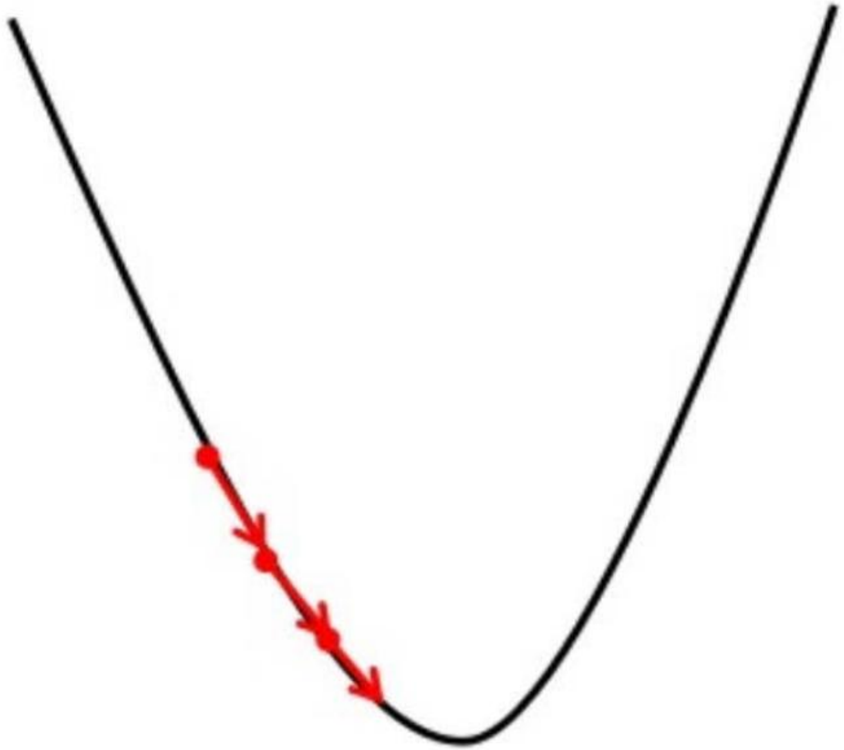
- Найти минимум функции $f(x_1, x_2)$: $\mathbb{R}^2 \to \mathbb{R}$
- Градиент:
  $\nabla f = \left(\frac{\partial f(x_1,x_2)}{\partial x_1}; \frac{\partial f(x_1,x_2)}{\partial x_2}\right)$
- $\Delta\vec{x} := -\eta\nabla f$,
  где $\eta > 0$ – малый параметр (*скорость обучения*)



Скорость обучения



Большая скорость обучения



Малая скорость обучения

Обратное распространение ошибки

1. Нейронная сеть получает входные данные и делает прогноз

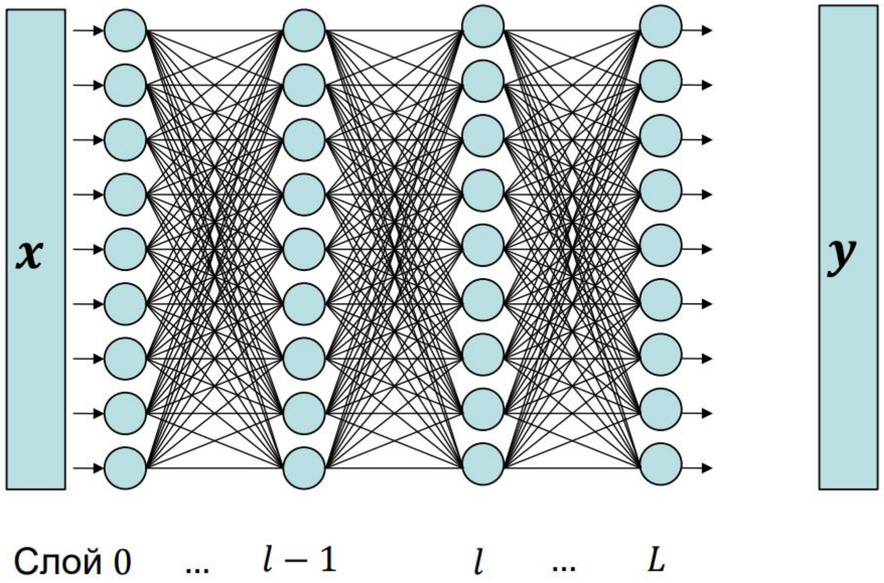
2. Вычисляет ошибку между прогнозом и реальным ответом

3. Передача это ошибки обратно через сеть – **обратное распространение**

4. Выясняем насколько каждый вес влияет на общую ошибку

5. Корректировка весов

Обратное распространение ошибки



Шаг 1.

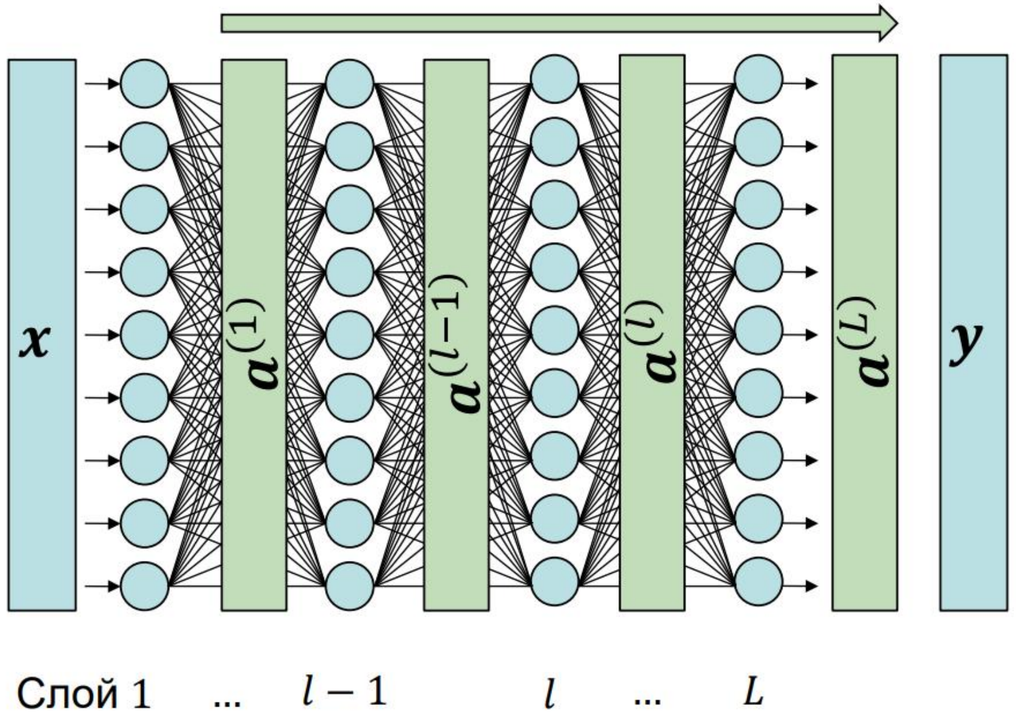
Подаем входной сигнал x

L – количество слоев нейронной сети

$\boldsymbol{a}^{(L)} = \boldsymbol{\alpha}(\boldsymbol{x})$ – выходной сигнал нейронной сети

$C = C_{(\boldsymbol{x},\boldsymbol{y})} = \frac{\left\|\boldsymbol{a}^{(L)} - \boldsymbol{y}\right\|^2}{2}$ – ошибка на прецеденте $(\boldsymbol{x}, \boldsymbol{y})$

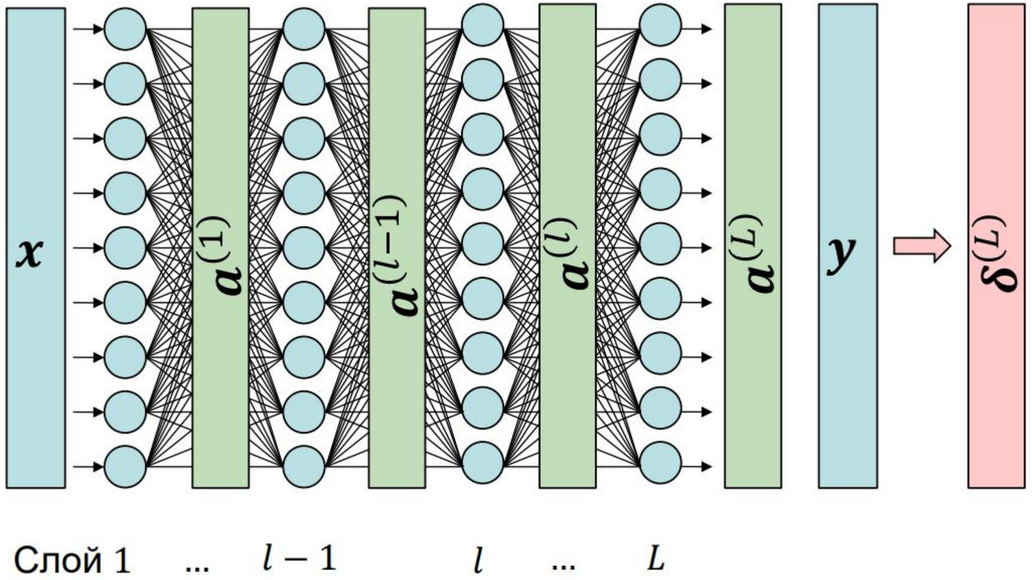
Обратное распространение ошибки



Шаг 2.

Считаем выходные сигналы для каждого слоя

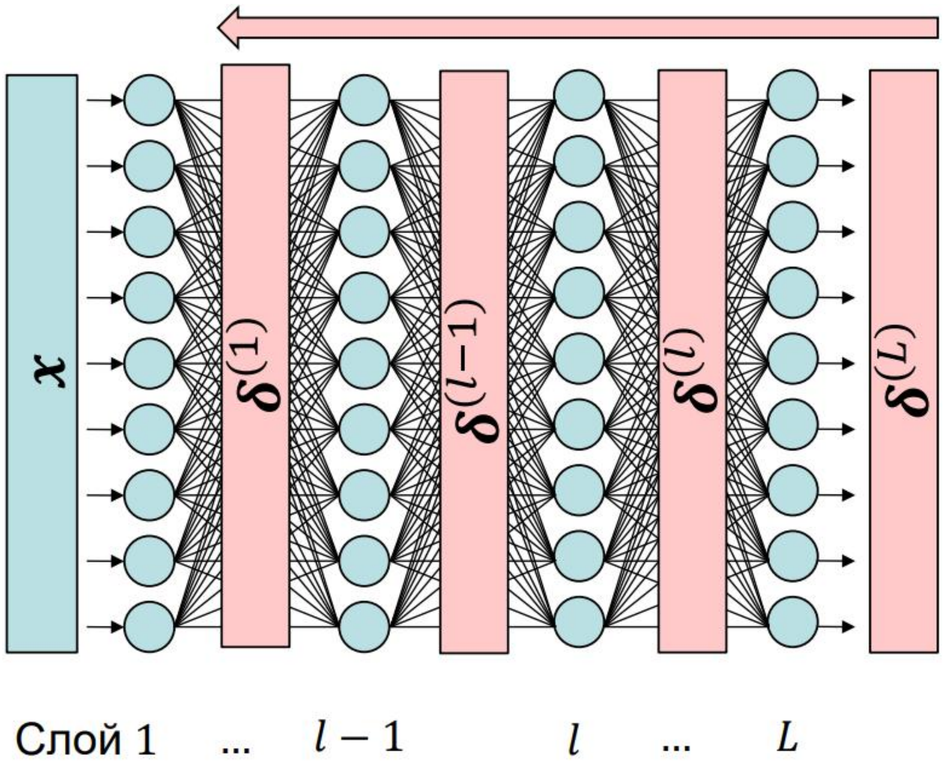
Обратное распространение ошибки



Шаг 3.

Считаем меру влияния $\delta^{(L)}$нейронов выходного слоя L на ошибку C

Обратное распространение ошибки



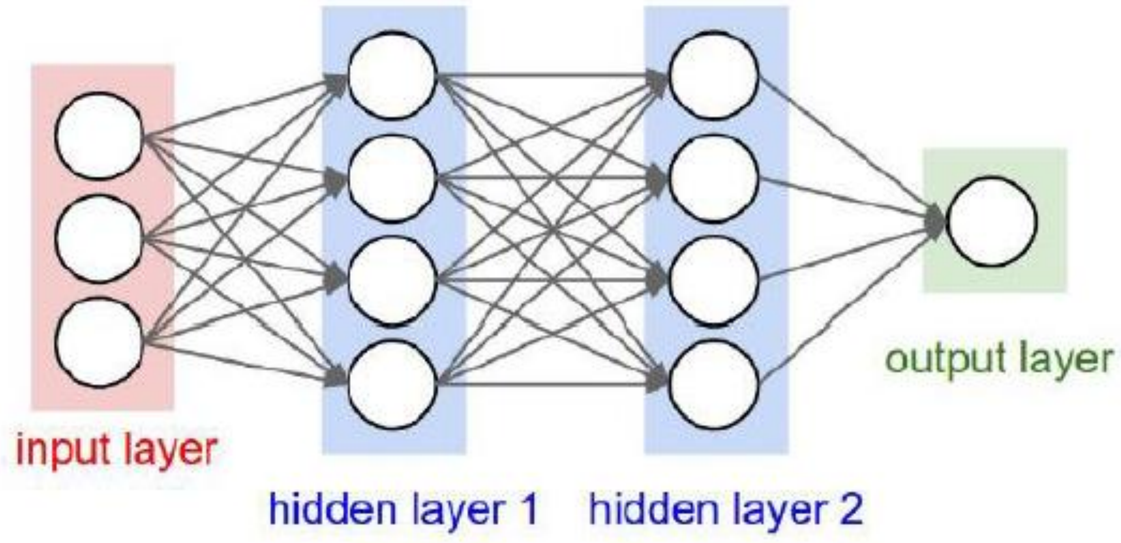
Шаг 4.

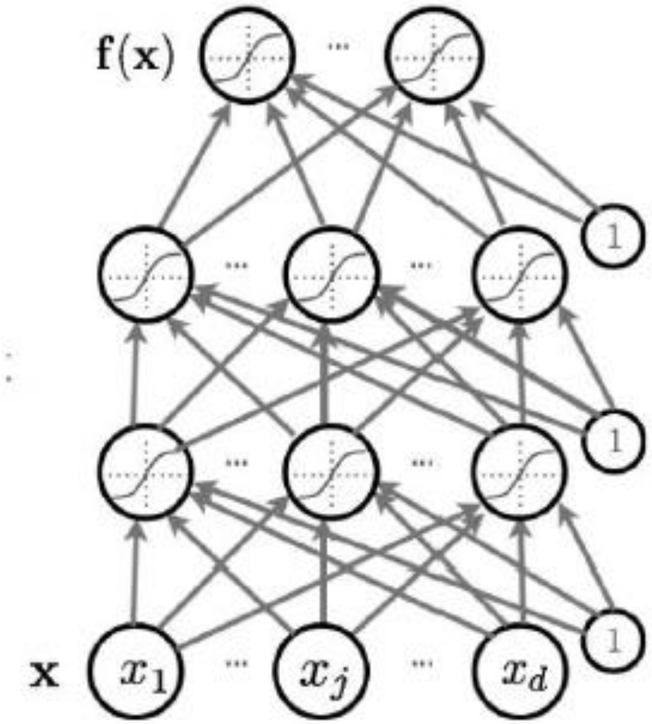
Вычислить в обратном порядке меру влияния $\delta^{(L)}$ на ошибку для каждого слоя

Шаг 5.

Градиентным методом оптимизации, зная $\delta^{(L)}$ найти
значения весов соответствующим минимуму общей ошибки

Вычислительный граф





Итог

- Нейрон – линейный классификатор или линейная регрессия
- Нейронная сеть – суперпозиция нейронов с нелинейной функцией активации
- Глубокая нейронная сеть состоит из последовательных линейных и нелинейных слоев. Чем сложнее задача, тем больше слоев, так как добавляются иерархические нелинейности
- Глубокое обучение стало возможно из-за накопленных больших данных для обучения, доступности вычислительных ресурсов, развитых фреймворков для разработки

Ресурсы

1. Я. Гудфеллоу, И. Бенжио, А. Курвиль «Глубокое обучение» http://www.deeplearningbook.org/

2. Курс от MIT https://introtodeeplearning.com/

3. Курс от Stanford http://deeplearning.stanford.edu/

4. Курс от Berkeley https://sites.google.com/view/berkeley-cs294-158-sp20/home

5. Машинное обучение (курс лекций, К.В. Воронцов) http://www.machinelearning.ru/

МИСИС
УНИВЕРСИТЕТ

Спасибо
за внимание!