



Южно-Уральский
государственный
университет

Национальный
исследовательский
университет

Создание искусственного интеллекта на основе нейронной сети

Соколинский Леонид Борисович
доктор физ.-мат. наук, профессор

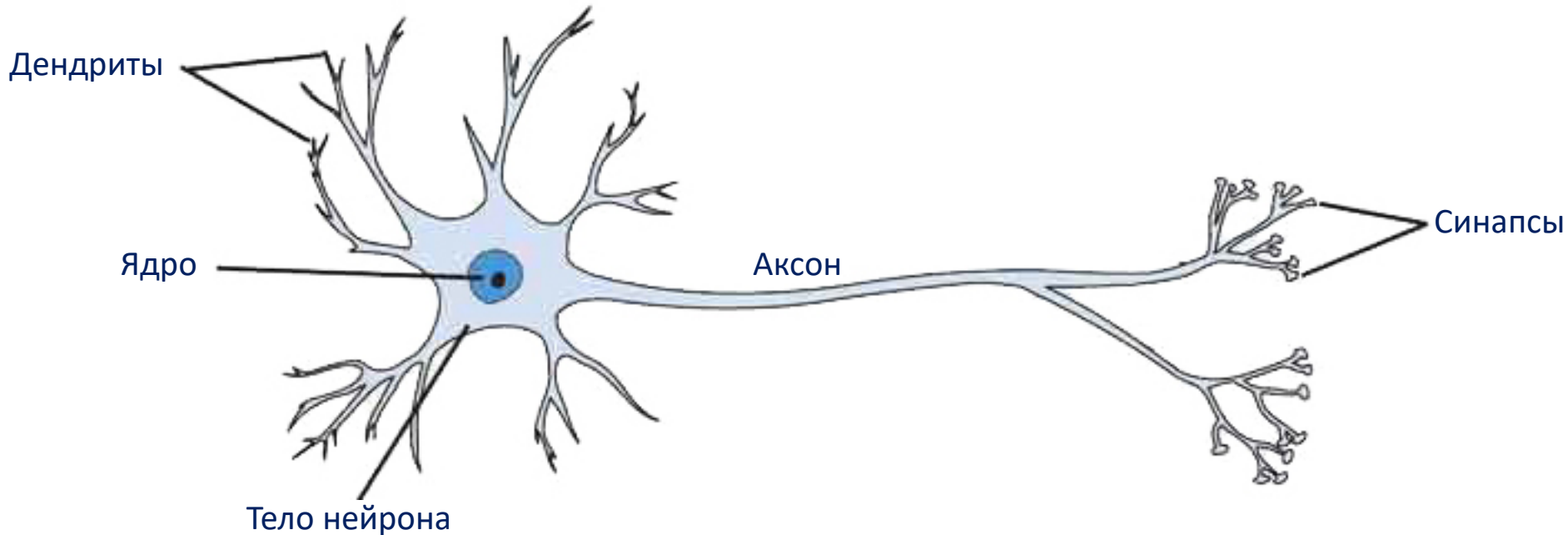
Востребованность на рынке труда

- С 2012 по 2019 годы количество вакансий специалистов по машинному обучению в России выросло в 20 раз
- Прямо сейчас специалистов по машинному обучению ищут Яндекс, Mail.ru, Сбербанк, МТС, РwС, М-Видео, Северсталь

Классификация методов машинного обучения

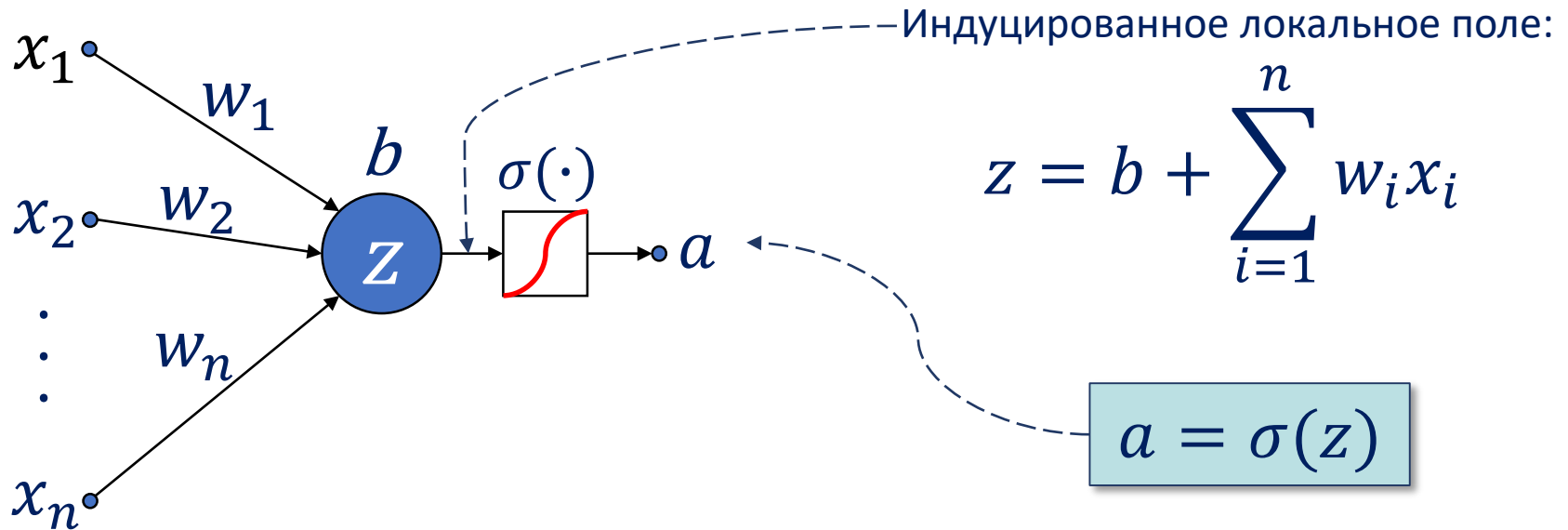


Биологический нейрон человеческого мозга



- *Дендриты* получают входные сигналы от других нейронов
- *Тело нейрона* суммирует входные сигналы, полученные от других нейронов, и в случае, когда сумма превышает определенный порог, генерирует выходной сигнал
- *Аксон* служит для передачи выходного сигнала другим нейронам
- *Синапсы* соединяют аксон с дендритами других нейронов; величина сигнала зависит от силы (*синаптического веса*) связи

Искусственный нейрон



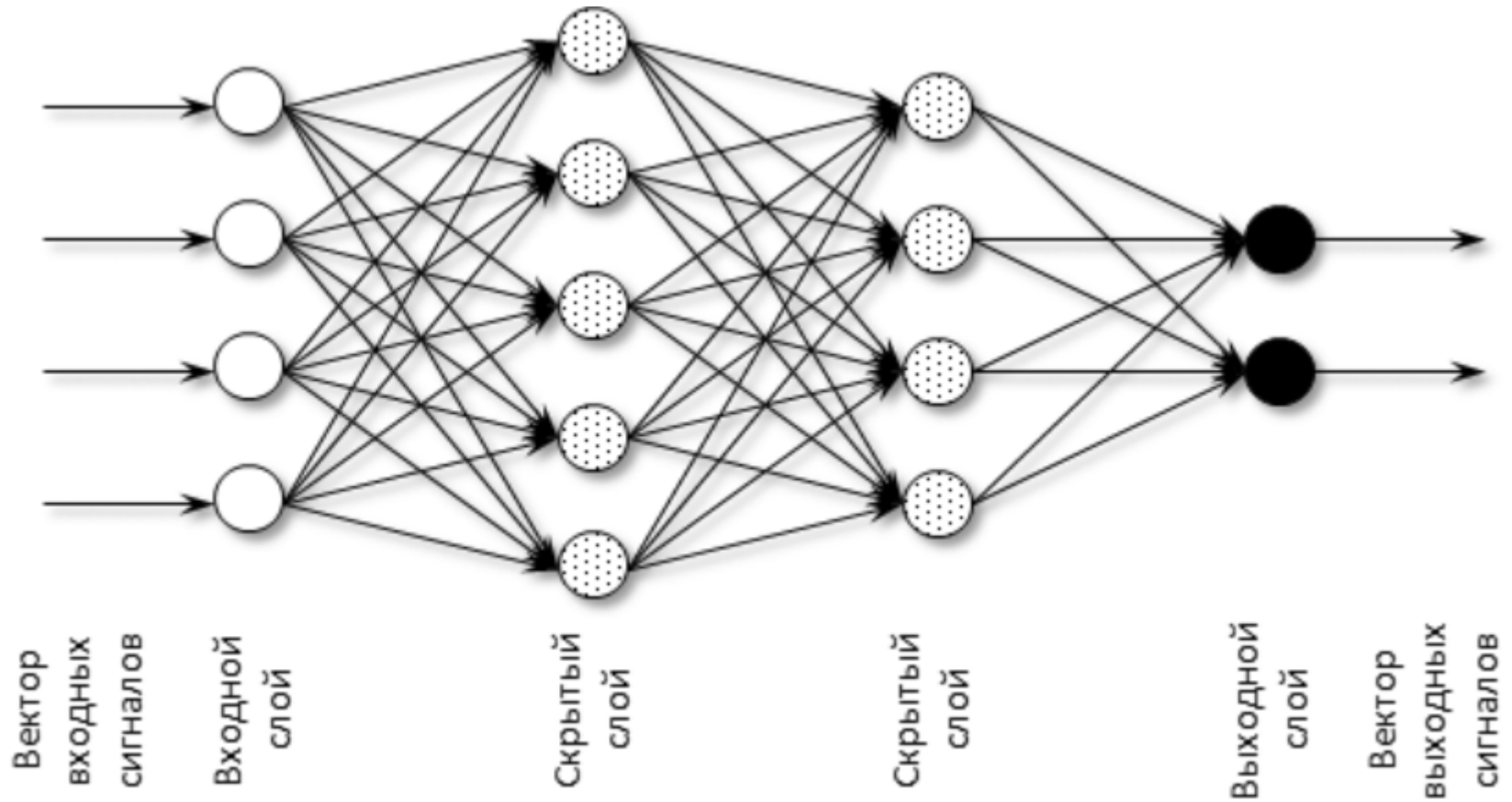
$\vec{x} = (x_1, x_2, \dots, x_n)$ – входные сигналы: $x_i \in \mathbb{R}$

$\vec{w} = (w_1, w_2, \dots, w_n)$ – синапсические веса: $w_i \in \mathbb{R}$

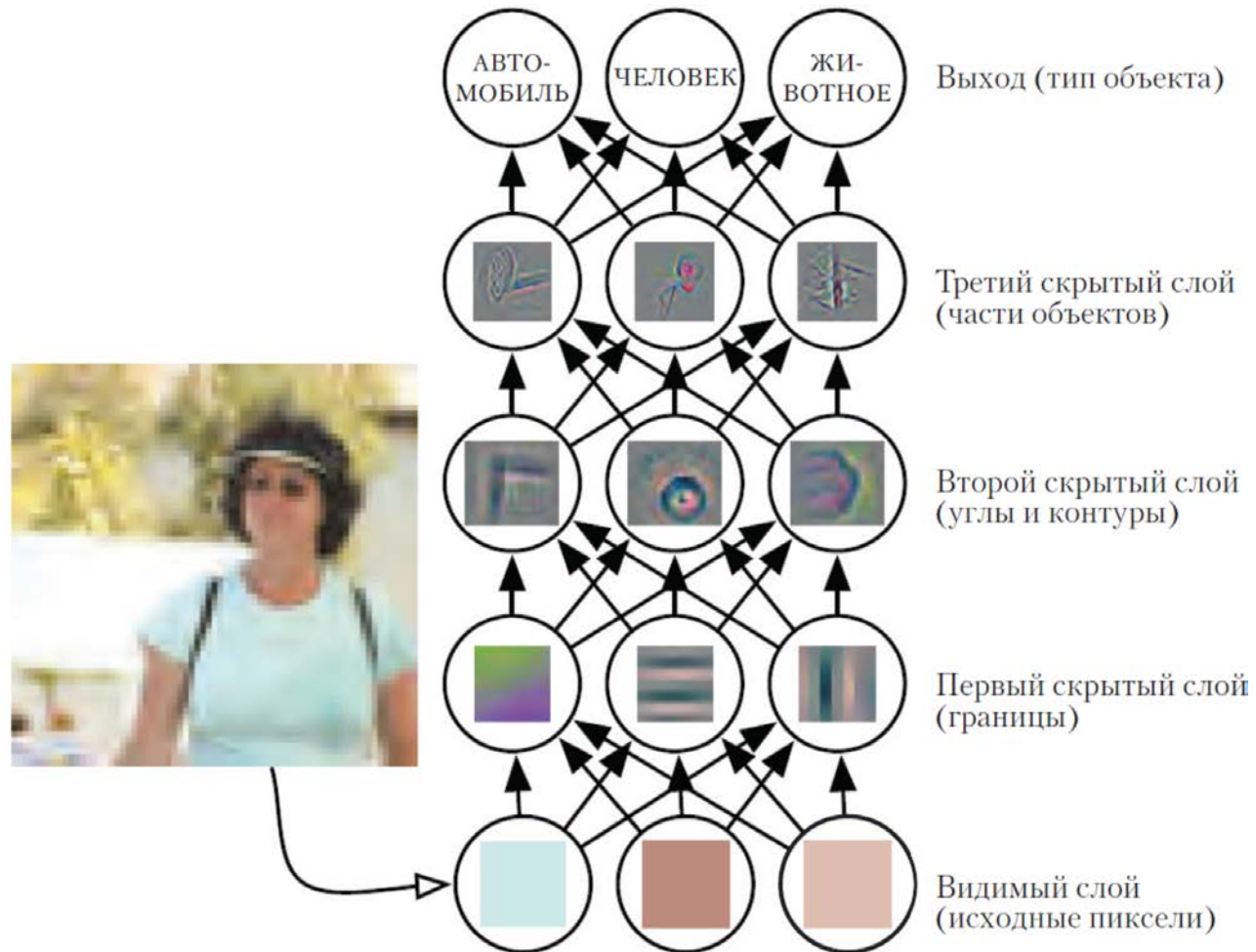
b – смещение: $b \in \mathbb{R}$

a – выходной сигнал: $0 < a < 1$

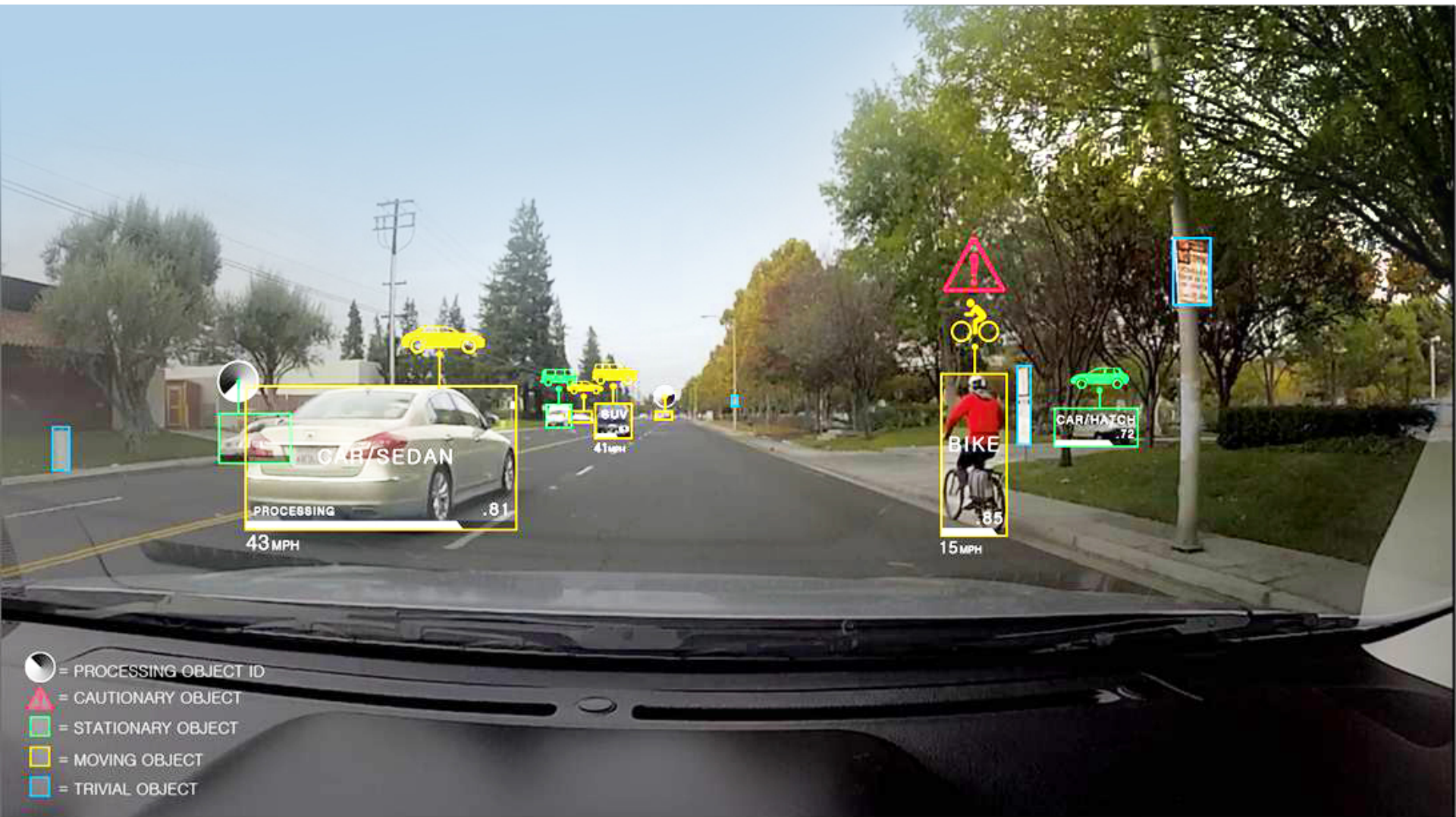
Искусственная нейронная сеть



Распознавание изображений



Беспилотный автомобиль



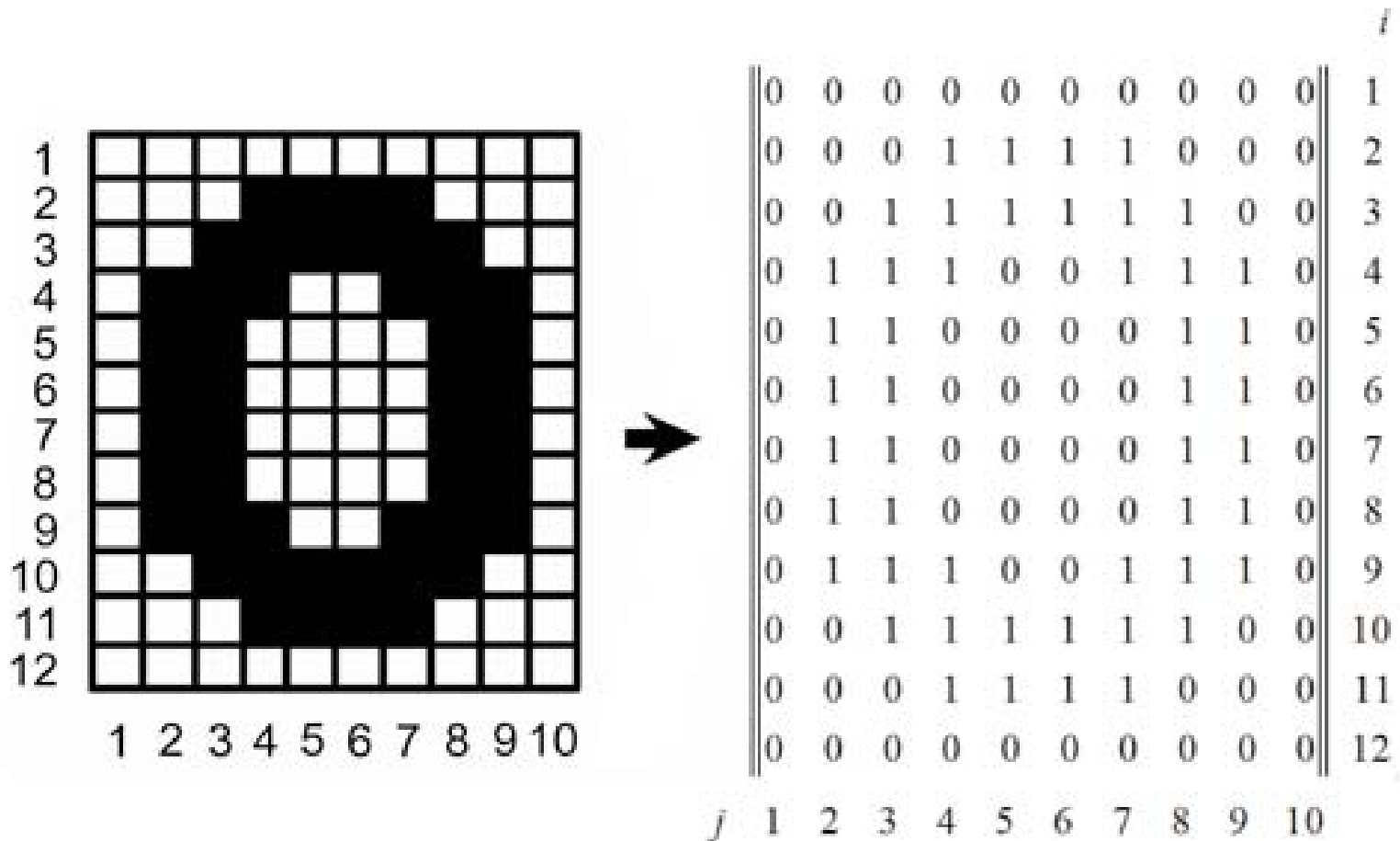
Сферы применения искусственных нейронных сетей

- **Распознавание образов и классификация** (отнесение образца к одному из predetermined классов)
- **Прогнозирование** (предсказание поведения сложной системы на основе анализа исторических данных)
- **Кластеризация** (разбиение объектов на классы в зависимости от схожести определенных свойств)

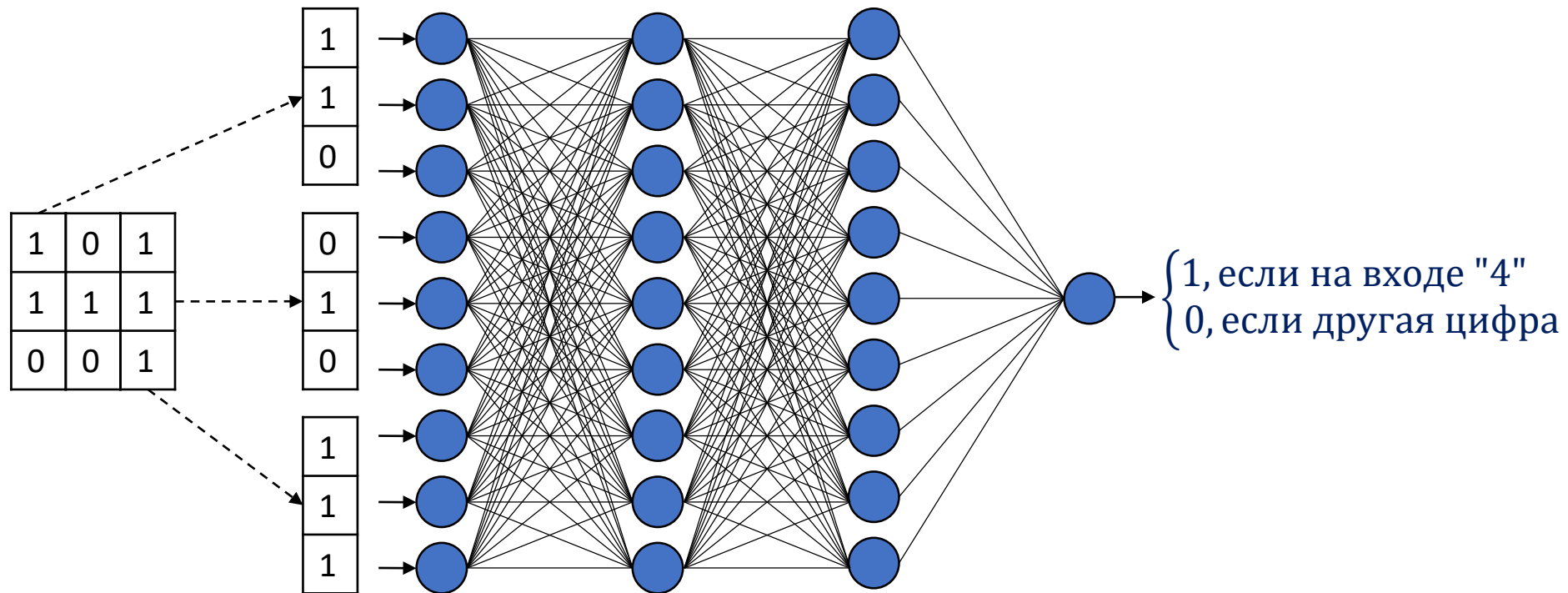
Задача распознавания рукописных цифр



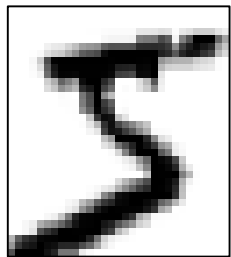
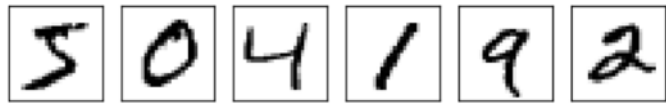
Цифровой образ изображения



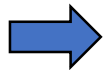
Распознавание цифры «4»



Нейронная сеть для распознавания рукописных цифр



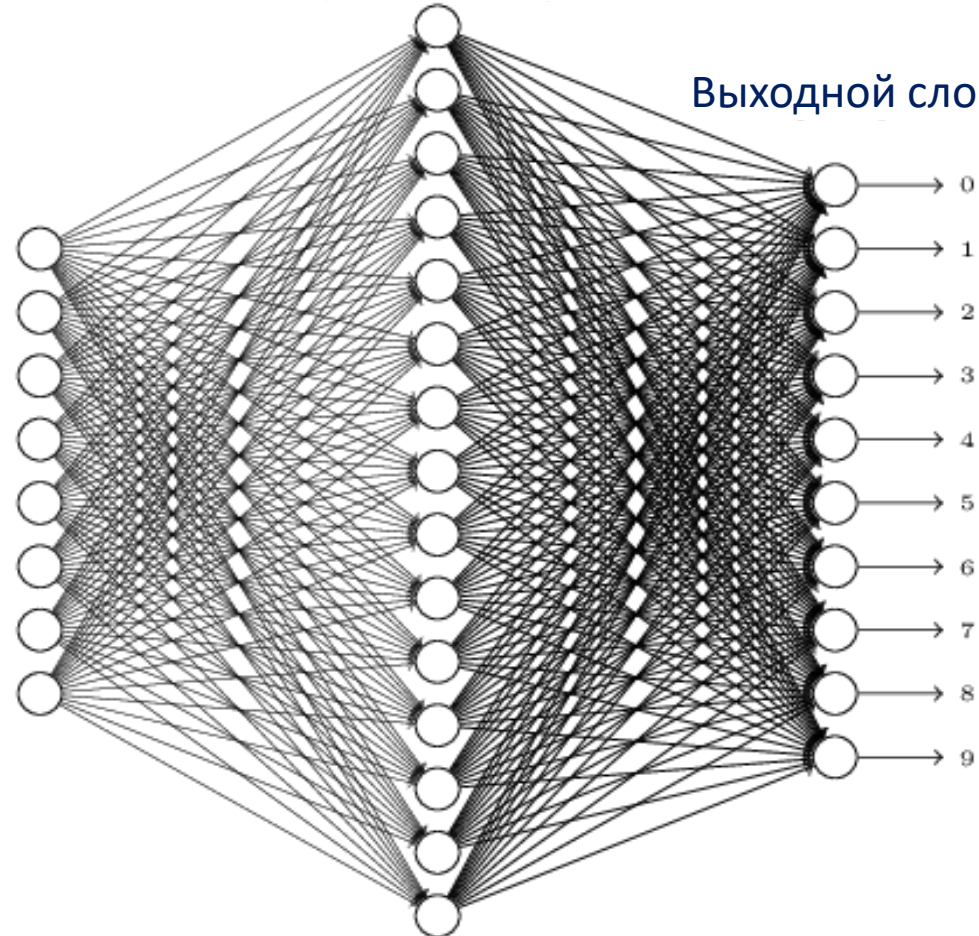
28x28



Входной слой
(784 нейрона)

Скрытый слой (15 нейронов)

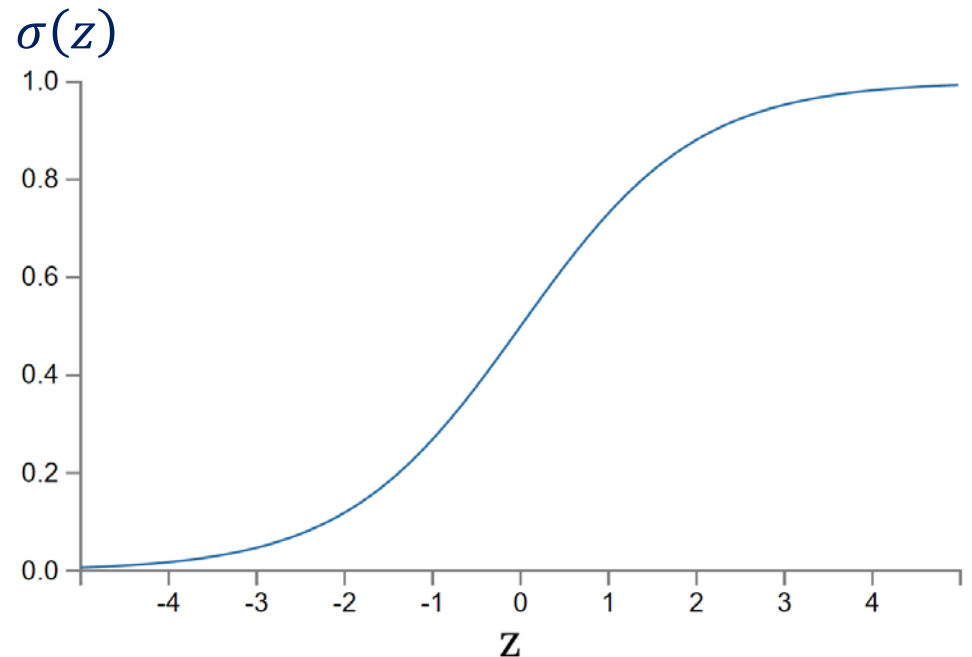
Выходной слой



Сигмоидальная (логистическая) функция

- Область определения: \mathbb{R}
- Множество значений: $(0;1)$

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



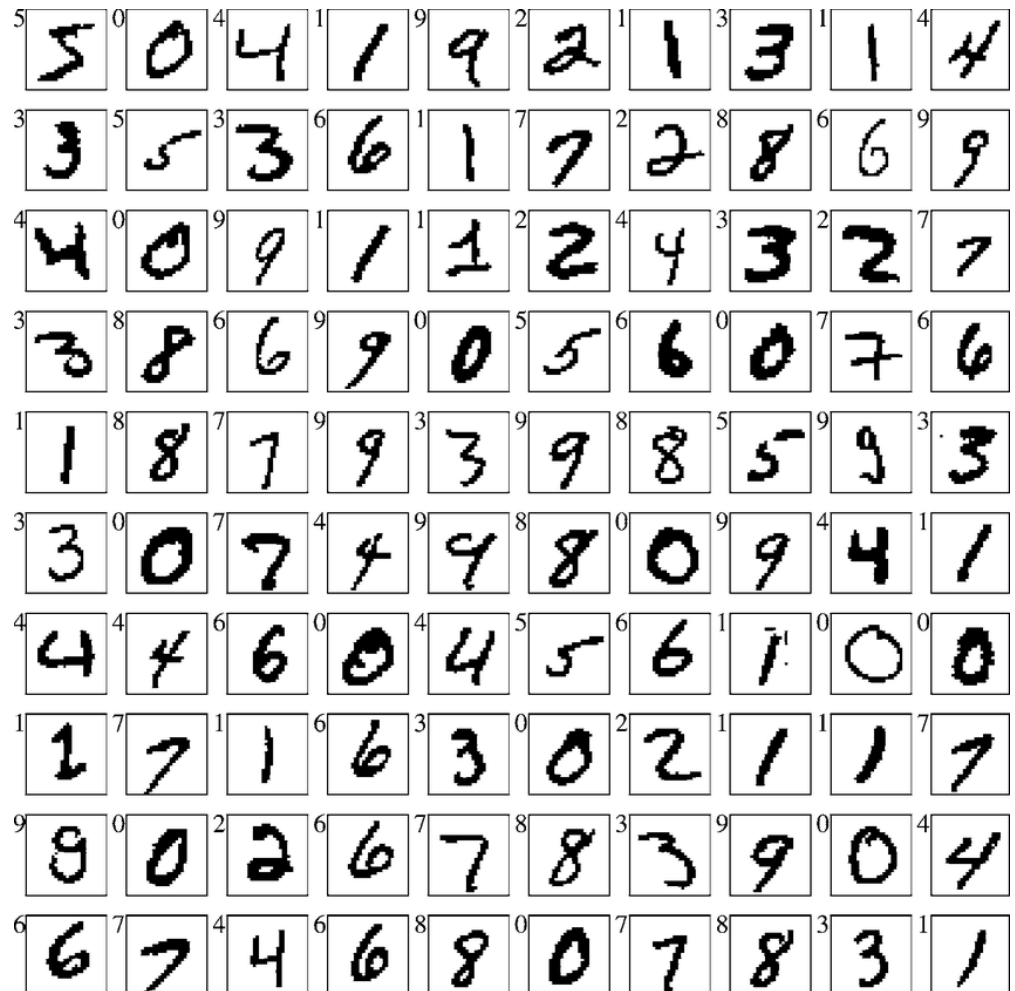
Модель обучения нейронной сети

- $\vec{x} \in \mathbb{R}^{28 \times 28}$ – изображение 28×28 , подаваемое на вход нейронной сети
- $\vec{y} \in \mathbb{R}^{10}$ – правильный ответ
- $\vec{\alpha}(\vec{x}) \in \mathbb{R}^{10}$ – ответ, выдаваемый нейронной сетью для входного изображения \vec{x}
- $V = \{(\vec{x}_k, \vec{y}_k) | 1 \leq i \leq K\}$ – обучающая выборка, (\vec{x}_k, \vec{y}_k) – прецедент
- $\vec{w} \in \mathbb{R}^Q$ – вектор, содержащий все синаптические веса сети
- $\vec{b} \in \mathbb{R}^P$ – вектор, содержащий все смещения
- $C_{(x,y)} = \frac{\|\vec{\alpha}(\vec{x}) - \vec{y}\|^2}{2}$ – функция ошибки
- Необходимо минимизировать среднюю ошибку по всем прецедентам обучающей выборки путем подбора значений \vec{w} и \vec{b}

Обучающая выборка: база данных MNIST

<http://yann.lecun.com/exdb/mnist/>

- База данных образцов рукописного написания цифр
- Все изображения имеют размер 28X28 пикселей в градациях серого
- 60000 изображений для обучения нейронной сети
- 10000 изображений для тестирования нейронной сети
- Каждое изображение в базе данных промаркировано правильной цифрой



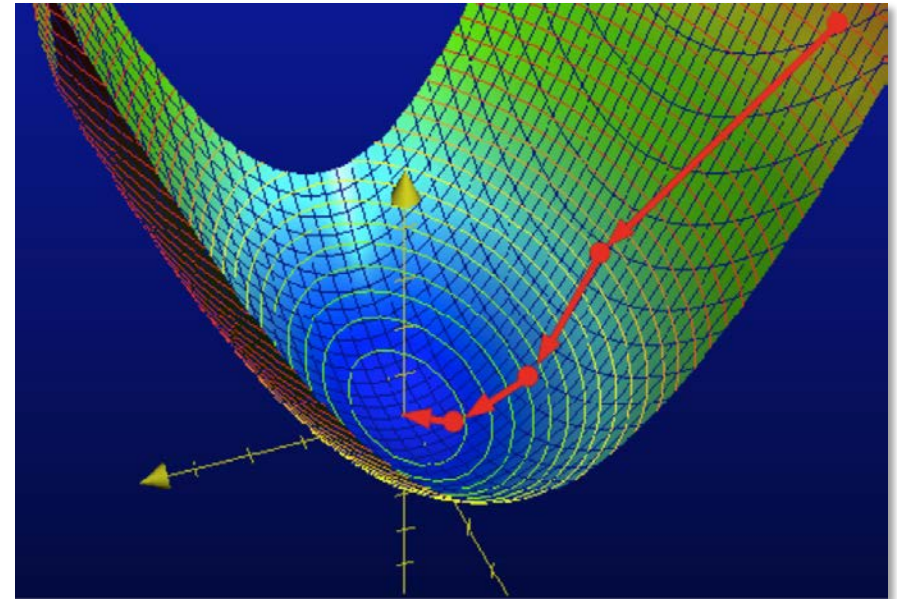
Минимизация ошибки: метод градиентного спуска

- Найти минимум функции $f(x_1, x_2): \mathbb{R}^2 \rightarrow \mathbb{R}$

- 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)



∇ – символ «набла»

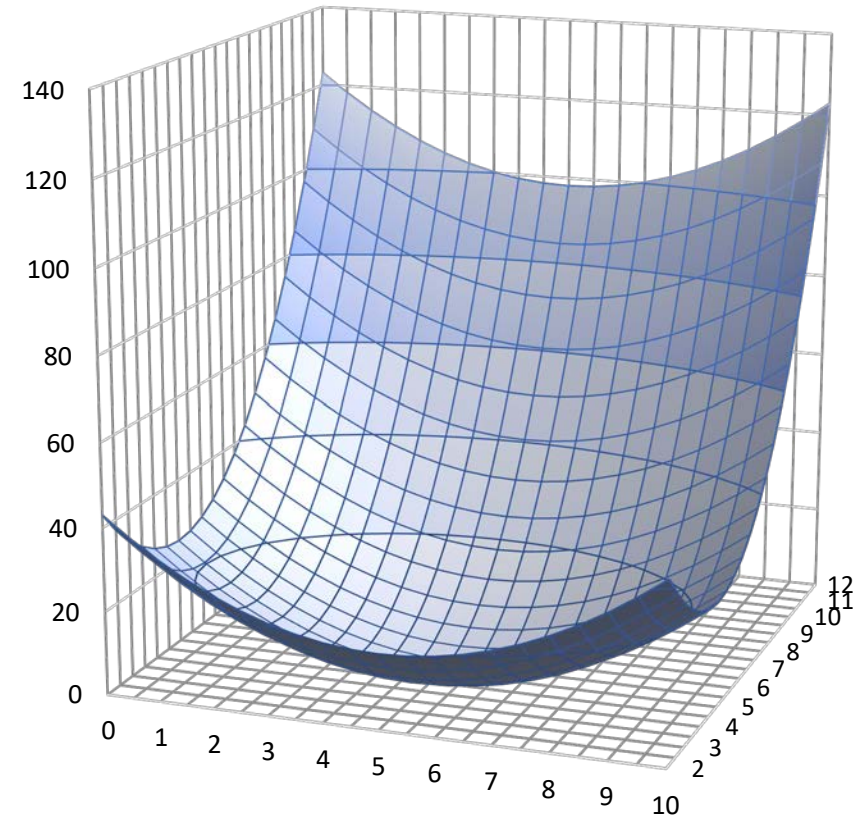
Пример вычисления градиента

$$f(x_1, x_2) = x_1^2 - 10x_1 + 2x_2^2 - 20x_2 + 75$$

$$\frac{\partial f(x_1, x_2)}{\partial x_1} = 2x_1 - 10$$

$$\frac{\partial f(x_1, x_2)}{\partial x_2} = 4x_2 - 20$$

$$\nabla f = (2x_1 - 10; 4x_2 - 20)$$



Пример градиентного спуска

$$\nabla f(x_1, x_2) = (2x_1 - 10; 4x_2 - 20)$$

$$\eta = 0.1$$

$$\Delta \vec{x} := -\eta \nabla f(x_1, x_2)$$

1. $\vec{x} := (1, 11)$

2. $\nabla f(1, 11) = (-8; 24)$

3. $\Delta \vec{x} := (0.8; -2.4)$

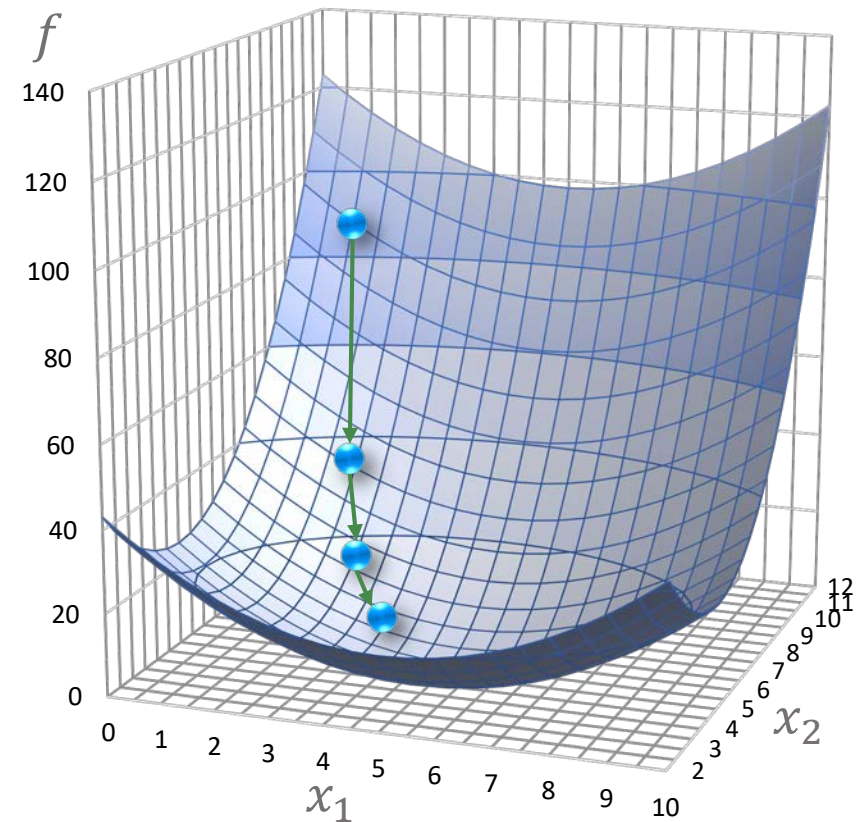
4. $\vec{x} := \vec{x} + \Delta \vec{x} = (1.8; 8.6)$

5. $\nabla f(1.8, 8.6) = (-6.4; 14.4)$

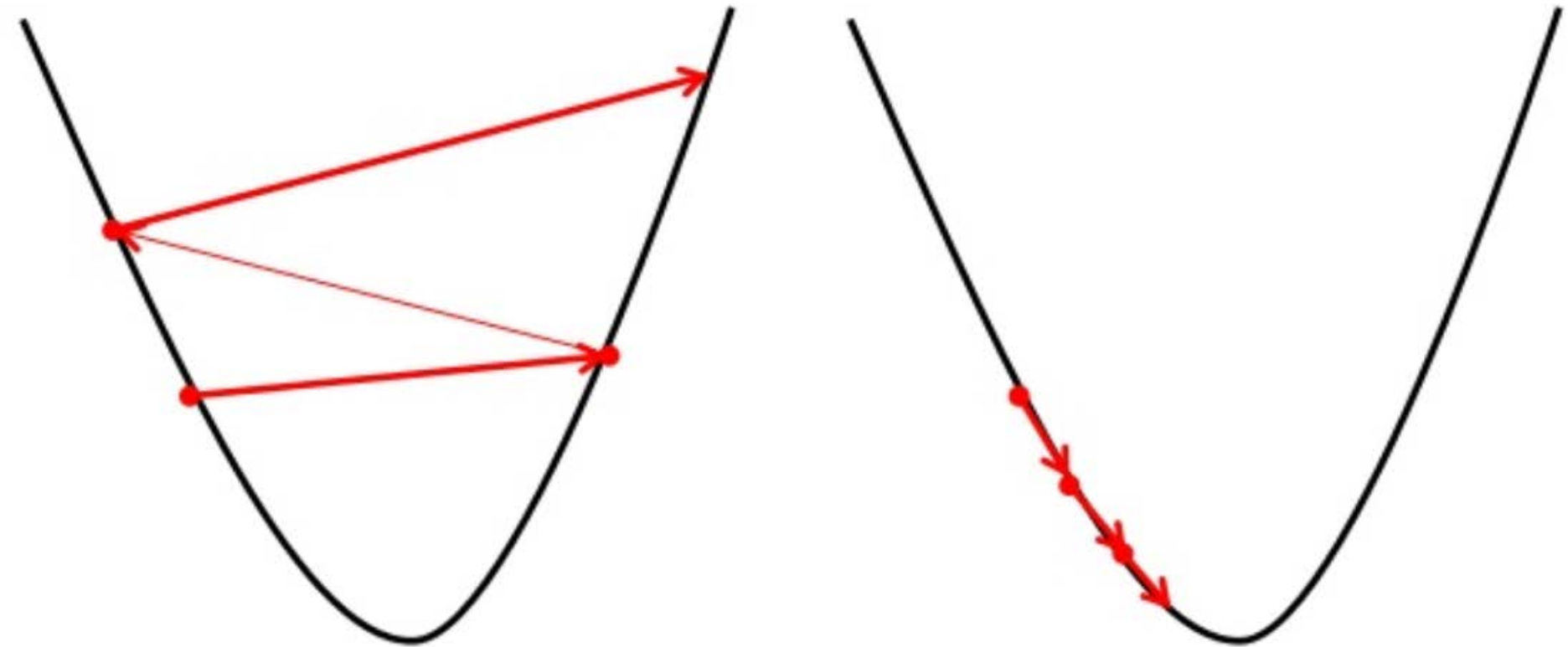
6. $\Delta \vec{x} := (0.64; -1.44)$

7. $\vec{x} := \vec{x} + \Delta \vec{x} = (2.44; 7.16)$

... ..



Выбор скорости обучения η



Большая скорость обучения

Малая скорость обучения

Стохастический градиентный спуск

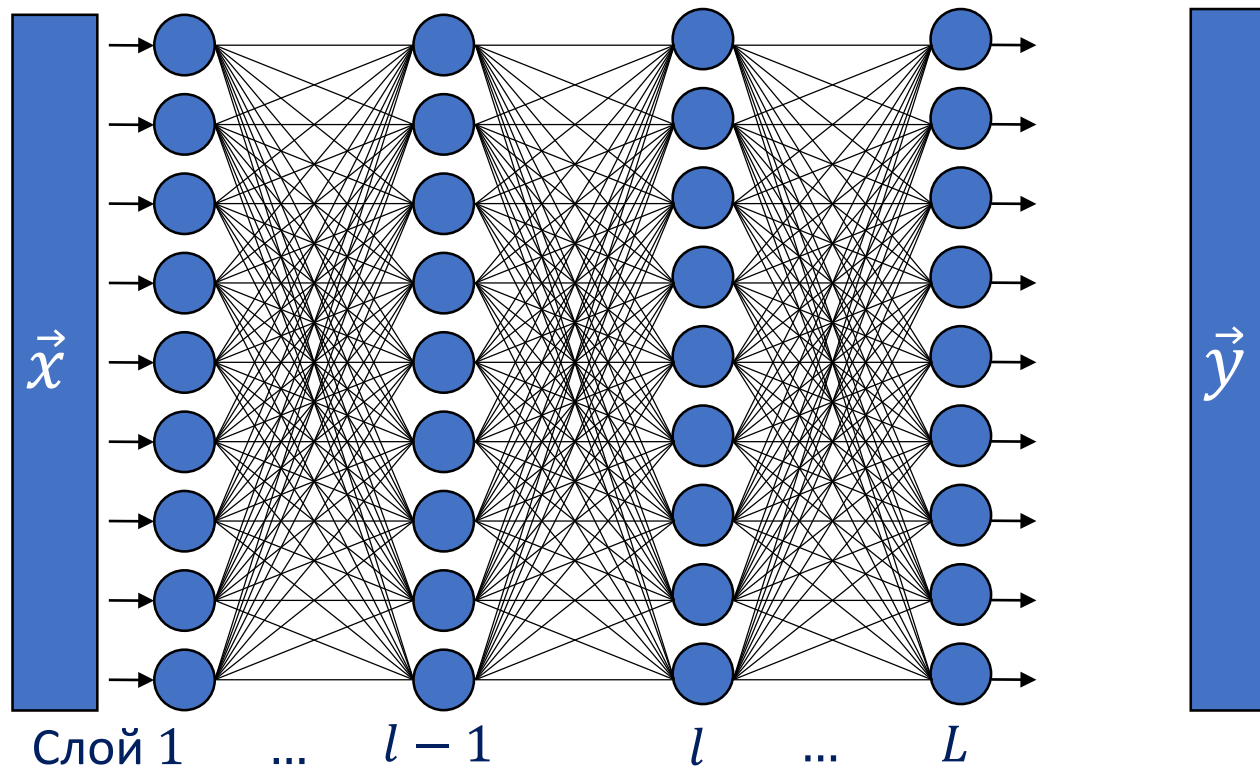
1. $\vec{w} := \overline{rnd}; \vec{b} := \overline{rnd}; epoch := 1;$
2. Последовательное разбиение V на непересекающиеся подвыборки V_1, \dots, V_M
3. $i := 0;$
4. $i := i + 1;$
5. $\nabla C_{V_i}(\vec{w}) := \frac{1}{|V_i|} \sum_{(x,y) \in V_i} \nabla C_{(x,y)}(\vec{w});$
6. $\nabla C_{V_i}(\vec{b}) := \frac{1}{|V_i|} \sum_{(x,y) \in V_i} \nabla C_{(x,y)}(\vec{b});$
7. $\vec{w} := \vec{w} - \eta \nabla C_{V_i}(\vec{w});$
8. $\vec{b} := \vec{b} - \eta \nabla C_{V_i}(\vec{b});$
9. **if** $i < M$ **goto** 4;
10. $shuffle(V); epoch := epoch + 1;$
11. **if** $epoch \leq 10$ **goto** 2;

Цикл по подвыборкам

Цикл по эпохам обучения

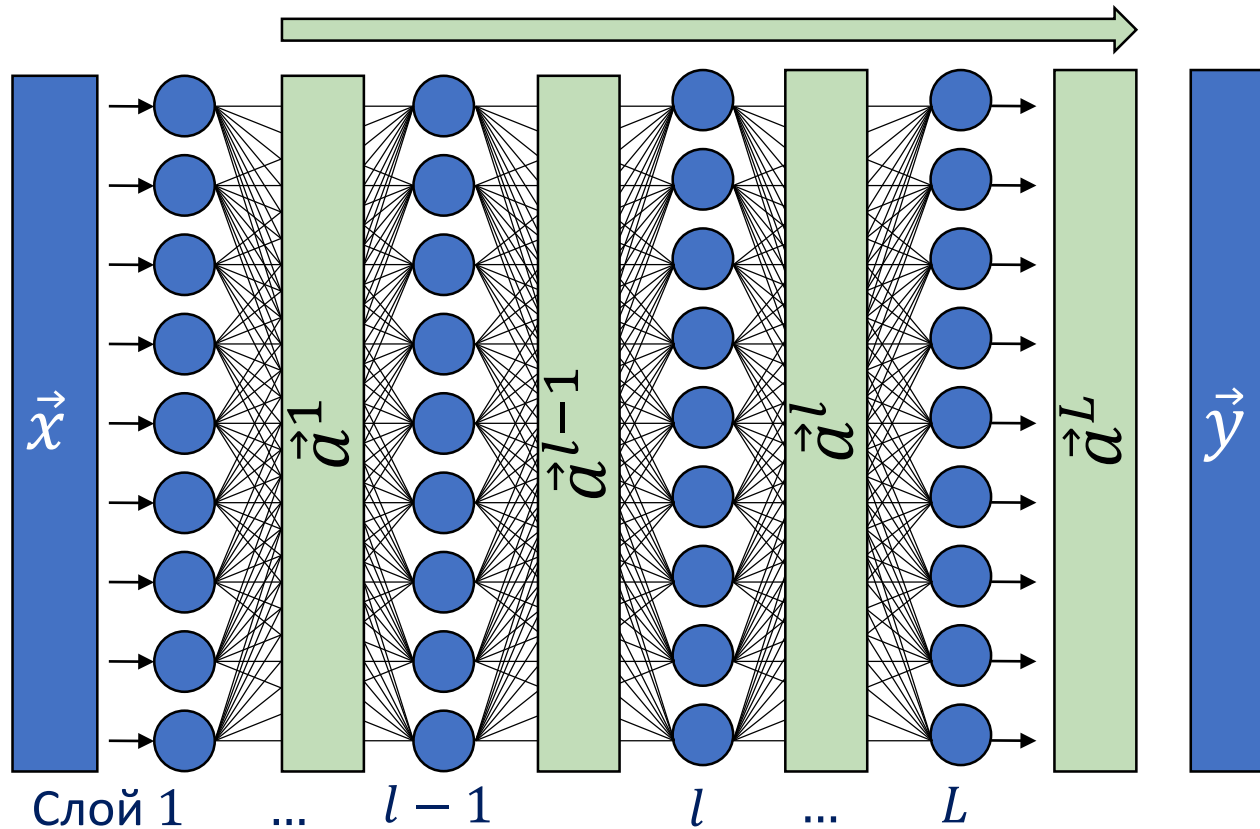
Метод обратного распространения ошибки

Шаг 1. Взять образец (\vec{x}, \vec{y}) и подать сигнал \vec{x} на вход нейронной сети



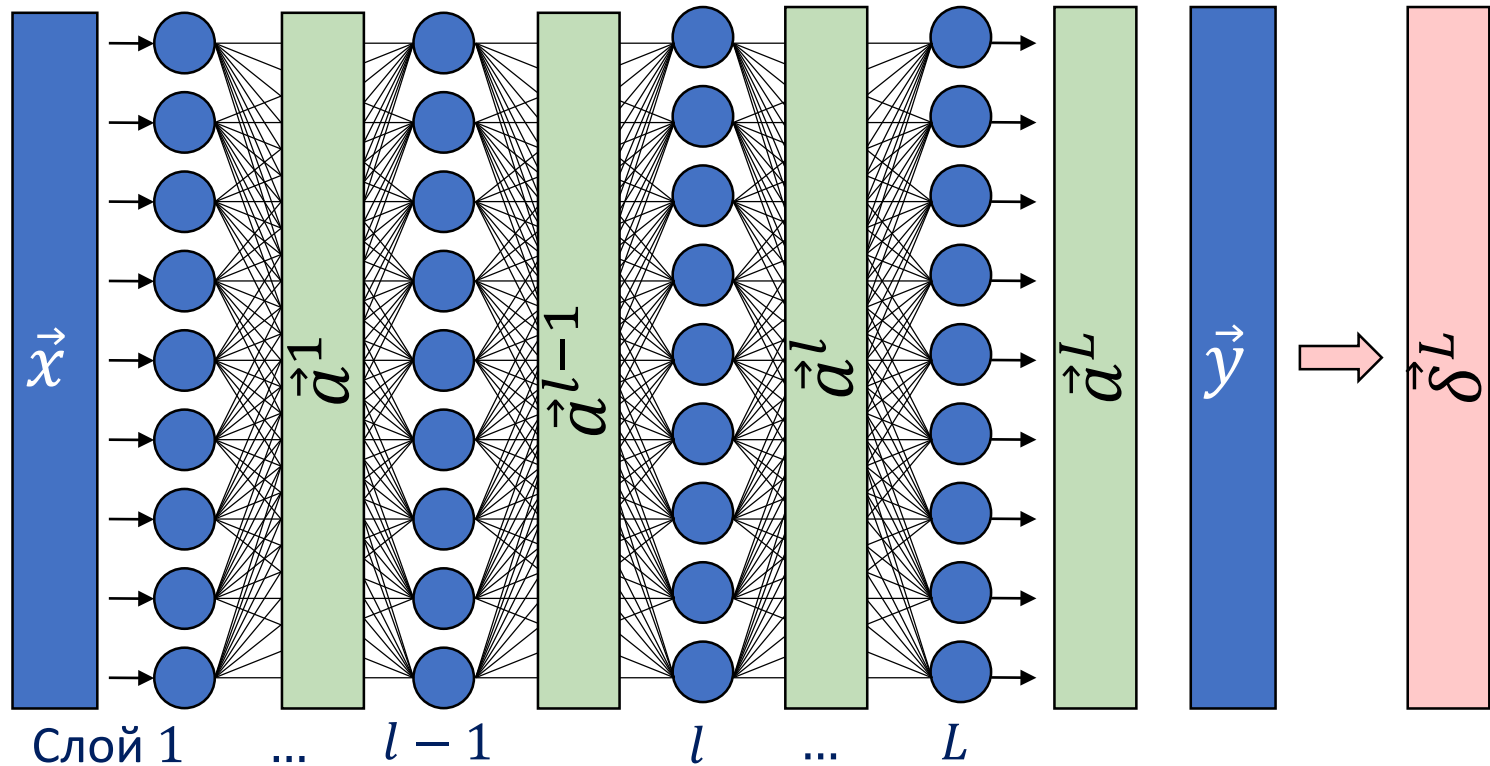
Метод обратного распространения ошибки

Шаг 2. Последовательно вычислить выходные сигналы \vec{a}^l для каждого слоя



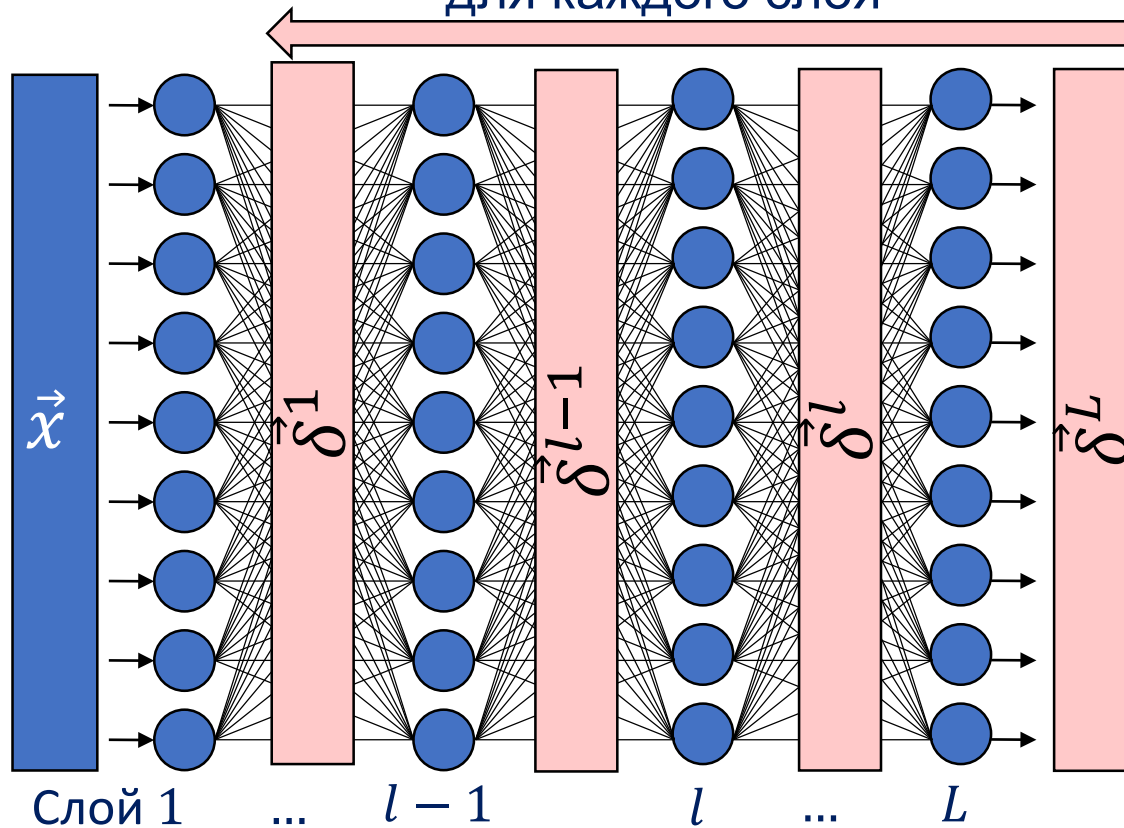
Метод обратного распространения ошибки

Шаг 3. Вычислить меру влияния $\vec{\delta}^L$ нейронов выходного слоя L на ошибку



Метод обратного распространения ошибки

Шаг 4. Вычислить в обратном порядке меру влияния $\vec{\delta}^l$ на ошибку для каждого слоя



Метод обратного распространения ошибки

Шаг 5. Используя $\vec{\delta}^l$, вычислить для каждого слоя l градиенты:

$$\nabla C_{(x,y)}(\vec{w})$$

$$\nabla C_{(x,y)}(\vec{b})$$

Формулы метода обратного распространения ошибки

$$\delta_i^L = (a_i^L - y_i) \cdot \sigma'(z_i^L) \quad (\text{BP1})$$

$$\delta_j^l = \sigma'(z_j^l) \sum_i w_{ij}^{l+1} \delta_i^{l+1} \quad (\text{BP2})$$

$$\frac{\partial \mathbb{C}(x,y)}{\partial b_i^l} = \delta_i^l \quad (\text{BP3})$$

$$\frac{\partial \mathbb{C}(x,y)}{\partial w_{ij}^l} = \delta_i^l \cdot a_j^{l-1} \quad (\text{BP4})$$

Создание на языке Python нейронной сети для распознавания рукописных цифр

Методические указания к лабораторной работе:

<https://www.susu.ru/ru/webform/sozdanie-iskusstvennogo-intellekta-na-osnove-neyronnoy-seti>

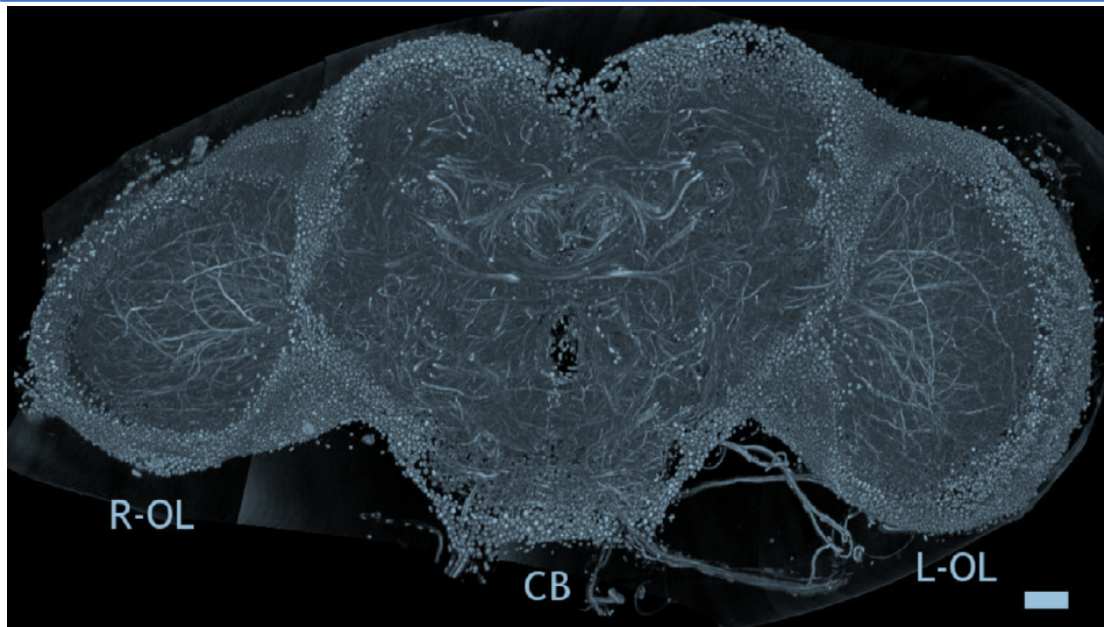


Может ли искусственная нейронная сеть моделировать интеллект человека?

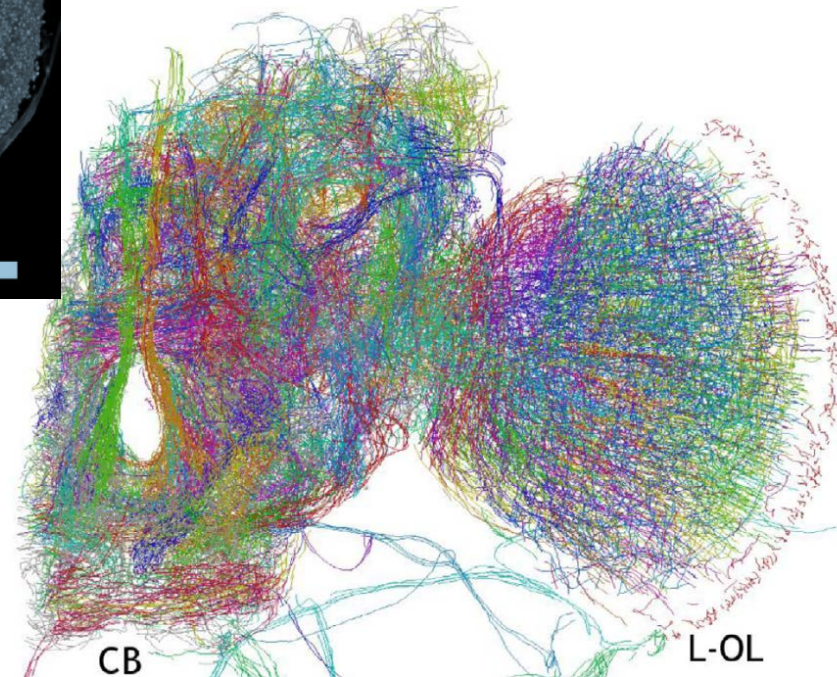
Искусственная нейронная сеть пока не может моделировать даже поведение плодовой мушки дрозофилы!



Мозг дрозофилы



100 000 нейронов



Количество нейронов в головном мозге живых существ



Попугай: $2 \cdot 10^9$

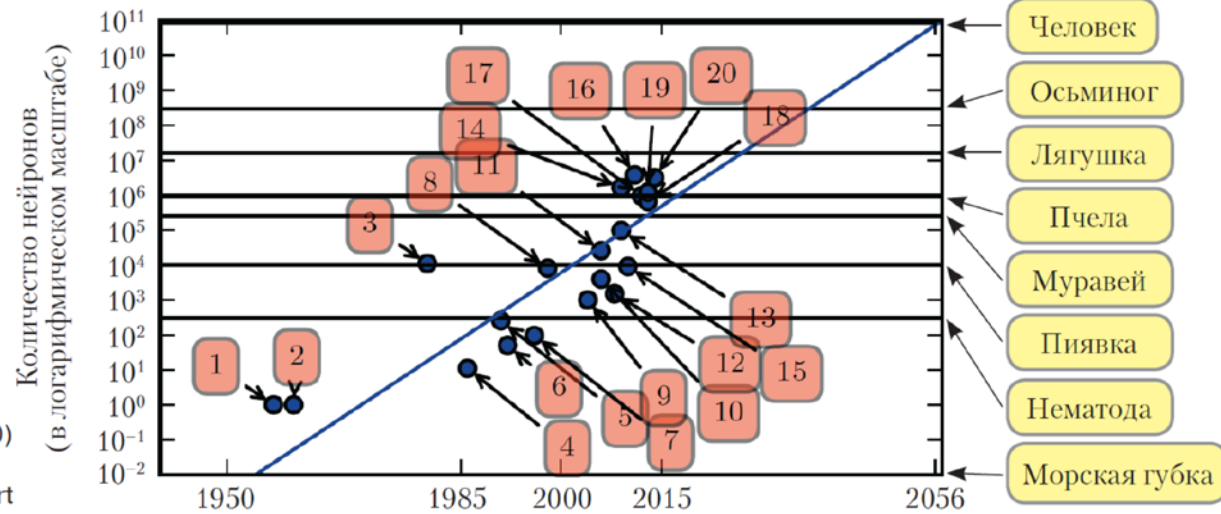


Горилла: $33 \cdot 10^9$



Человек: $86 \cdot 10^9$

Искусственная нейронная сеть сравнивается с человеческим мозгом по числу нейронов не раньше 2050-х годов



1. Перцептрон (Rosenblatt, 1958, 1962)
2. Адаптивный линейный элемент (Widrow and Hoff, 1960)
3. Неокогнитрон (Fukushima, 1980)
4. Ранняя сеть с обратным распространением (Rumelhart)
5. Рекуррентная нейронная сеть для распознавания речи (Robinson and Fallside, 1991)
6. Многослойный перцептрон для распознавания речи (Bengio et al., 1991)
7. Сигмоидальная сеть доверия со средним полем (Saul et al., 1996)
8. LeNet-5 (LeCun et al., 1998b)
9. Нейронная эхо-сеть (Jaeger and Haas, 2004)
10. Глубокая сеть доверия (Hinton et al., 2006)
11. GPU-ускоренная сверточная сеть (Chellapilla et al., 2006)
12. Глубокая машина Больцмана (Salakhutdinov and Hinton, 2009a)
13. GPU-ускоренная сеть глубокого доверия (Raina et al., 2009)
14. Сверточная сеть без учителя (Jarrett et al., 2009)
15. GPU-ускоренный многослойный перцептрон (Ciresan et al., 2010)
16. Сеть OMP-1 (Coates and Ng, 2011)
17. Распределенный автокодировщик (Le et al., 2012)
18. Сверточная сеть с несколькими GPU (Krizhevsky et al., 2012)
19. Сверточная сеть без учителя на компьютерах типа COTS HPC (Coates et al., 2013)
20. GoogLeNet (Szegedy et al., 2014a)

Магистерская программа «Машинное обучение и анализ больших данных»

Направление подготовки

- ▶ 02.04.02 «Фундаментальная информатика и информационные технологии»

Аннотация

- ▶ Машинное обучение и анализ больших данных является одним из главных трендов в IT-индустрии в настоящее время, именно поэтому профессионалы в этой области очень востребованы на современном рынке труда. Магистерская программа «Машинное обучение и анализ больших данных» готовит специалистов высшей квалификации в области искусственного интеллекта и анализа больших данных. В процессе обучения студенты осваивают современные методы машинного обучения, разработки нейронных сетей, интеллектуального анализа данных. Теоретическое обучение сочетается с большим количеством практических занятий. Выпускник магистратуры подготовлен к деятельности, требующей использования самых современных методов машинного обучения и анализа данных.

Ключевые учебные курсы

- ▶ Математические основы машинного обучения
- ▶ Машинное обучение
- ▶ Применение искусственных нейронных сетей
- ▶ Хранение и обработка больших данных
- ▶ Современные методы анализа больших данных
- ▶ Теоретические основы разработки систем управления большими данными
- ▶ Python для решения задач интеллектуального анализа данных
- ▶ Современные NoSQL-системы
- ▶ Высокопроизводительные вычисления



Руководитель программы:
проф., д.ф.-м.н.
Л.Б. Соколинский
leonid.sokolinsky@susu.ru



Южно-Уральский
государственный
университет

Национальный
исследовательский
университет

Спасибо
за внимание!

www.susu.ru