



НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
УНИВЕРСИТЕТ

# Нейронные сети с нуля

## Программный проект

### Факультет компьютерных наук

Гимранов Артур, БПМИ204

Научный руководитель: к.ф.-м.н., доцент Трушин Дмитрий Витальевич

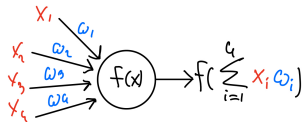
Национальный исследовательский университет  
«Высшая школа экономики» (Москва)

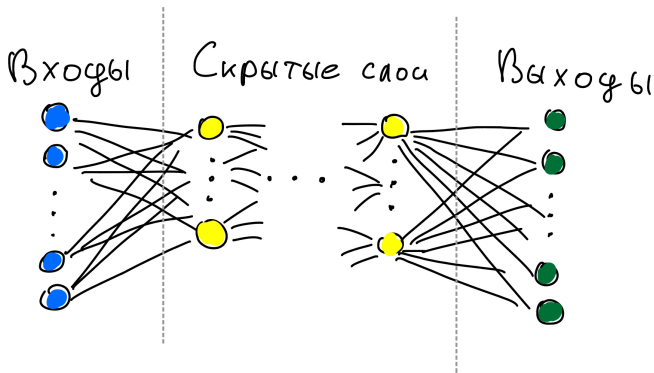
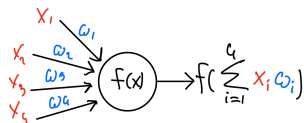
Июнь 2022

- Изучить теорию нейросетей и градиентного спуска

- Изучить теорию нейросетей и градиентного спуска
- Имплементировать необходимые классы и структуры

- Изучить теорию нейросетей и градиентного спуска
- Имплементировать необходимые классы и структуры
- Протестировать на реальных данных





## 1. Веса связей лежат в матрице $A$

1. Веса связей лежат в матрице  $A$
2. Смещение лежит в векторе  $b$



1. Веса связей лежат в матрице  $A$
2. Смещение лежит в векторе  $b$
3.  $f$  – функция активации

1. Веса связей лежат в матрице  $A$
2. Смещение лежит в векторе  $b$
3.  $f$  – функция активации
  - 3.1 softmax
  - 3.2 relu
  - 3.3 sigmoid

1. Веса связей лежат в матрице  $A$
2. Смещение лежит в векторе  $b$
3.  $f$  – функция активации
  - 3.1 softmax
  - 3.2 relu
  - 3.3 sigmoid
4. Весь выход слоя  $f(Ax + b)$

1. Веса связей лежат в матрице  $A$
2. Смещение лежит в векторе  $b$
3.  $f$  – функция активации
  - 3.1 softmax
  - 3.2 relu
  - 3.3 sigmoid
4. Весь выход слоя  $f(Ax + b)$

Зададим слой блоком, который хранит  $\theta = (A, b)$  и  $f$ .  
 $g(x, \theta) = f(Ax + b)$

$$\mathbb{R}^n \xrightarrow{x} \boxed{\theta} \xrightarrow{y} \mathbb{R}^m$$

$g(x, \theta)$

- В  $x$  лежат  $k$  признаков,  $x = [x^{(1)}, \dots, x^{(k)}]^t$

- В  $x$  лежат  $k$  признаков,  $x = [x^{(1)}, \dots, x^{(k)}]^t$
- В  $y$  лежат  $l$  признаков,  $y = [y^{(1)}, \dots, y^{(l)}]^t$

- В  $x$  лежат  $k$  признаков,  $x = [x^{(1)}, \dots, x^{(k)}]^t$
- В  $y$  лежат  $l$  признаков,  $y = [y^{(1)}, \dots, y^{(l)}]^t$
- Начальная выборка  $\forall i \leq n : x_i \rightarrow y_i$

- В  $x$  лежат  $k$  признаков,  $x = [x^{(1)}, \dots, x^{(k)}]^t$
- В  $y$  лежат  $l$  признаков,  $y = [y^{(1)}, \dots, y^{(l)}]^t$
- Начальная выборка  $\forall i \leq n : x_i \rightarrow y_i$

Хотим найти  $F(x) = y$ . Будем искать  $F$  в виде нейросети.



- В  $x$  лежат  $k$  признаков,  $x = [x^{(1)}, \dots, x^{(k)}]^t$
- В  $y$  лежат  $l$  признаков,  $y = [y^{(1)}, \dots, y^{(l)}]^t$
- Начальная выборка  $\forall i \leq n : x_i \rightarrow y_i$

Хотим найти  $F(x) = y$ . Будем искать  $F$  в виде нейросети.

$$\mathbb{R}^k \xrightarrow{x_i} \boxed{\theta_1} \xrightarrow{w_i} \boxed{\theta_2} \xrightarrow{z_i} \mathbb{R}^l$$

$g_1(x, \theta_1) \qquad g_2(x, \theta_2)$

Функция ошибки

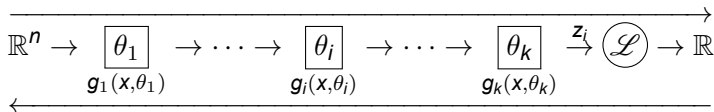
$$\phi(\theta_1, \theta_2) = \frac{1}{n} \sum_{i=1}^n \|g_2(g_1(x_i, \theta_1), \theta_2) - y_i\|^2 \rightarrow \min_{\theta_1, \theta_2}$$

Минимум по  $\theta_1, \theta_2$  найдем градиентным спуском

Функция ошибки для всей сети

$$\phi(\Theta) = \frac{1}{n} \sum_{i=1}^n \mathcal{L}(\mathbf{z}_i, y_i)$$

Прямым обходом мы считаем все выходы и ошибку



Обратным проходом считаем градиенты  $\frac{\partial \phi}{\partial \theta_i}$

1. Net. Инцилизирует ресурсы, обучает и предсказывает.

1. `Net`. Инициализирует ресурсы, обучает и предсказывает.
2. `ComputeBlock`. Наши слои (блоки). Содержит функцию активации и параметры. Вычисляет функцию и необходимые градиенты.

1. `Net`. Инициализирует ресурсы, обучает и предсказывает.
2. `ComputeBlock`. Наши слои (блоки). Содержит функцию активации и параметры. Вычисляет функцию и необходимые градиенты.
3. `LossFunction`. Функция ошибок, нужна для расчета отклонения и вычисления градиента

1. `Net`. Инициализирует ресурсы, обучает и предсказывает.
2. `ComputeBlock`. Наши слои (блоки). Содержит функцию активации и параметры. Вычисляет функцию и необходимые градиенты.
3. `LossFunction`. Функция ошибок, нужна для расчета отклонения и вычисления градиента
4. `ActivationFunction`. Класс родитель для функций активации, имеет виртуальные методы для вычисления функции и ее производной в точке.
  - 4.1 `Sigmoid`
  - 4.2 `Relu`
  - 4.3 `Softmax`

Нейросеть тестировалась на популярном датасете mnist.  
Точность составила более 90% при 12 минутах обучения на  
CPU

## Условия тестирования:

- Выборка из 8500 изображений рукописных цифр



## Условия тестирования:

- Выборка из 8500 изображений рукописных цифр
- Входной слой состоял из 784 нейронов, каждый из которых отвечал за пиксель в картинке  $28 \times 28$ , с функцией активации relu

## Условия тестирования:

- Выборка из 8500 изображений рукописных цифр
- Входной слой состоял из 784 нейронов, каждый из которых отвечал за пиксель в картинке  $28 \times 28$ , с функцией активации relu
- Первый скрытый слой состоял из 16 нейронов, с функцией активации relu

## Условия тестирования:

- Выборка из 8500 изображений рукописных цифр
- Входной слой состоял из 784 нейронов, каждый из которых отвечал за пиксель в картинке  $28 \times 28$ , с функцией активации relu
- Первый скрытый слой состоял из 16 нейронов, с функцией активации relu
- Второй скрытый слой состоял из 16 нейронов, с функцией активации softmax

## Условия тестирования:

- Выборка из 8500 изображений рукописных цифр
- Входной слой состоял из 784 нейронов, каждый из которых отвечал за пиксель в картинке  $28 \times 28$ , с функцией активации relu
- Первый скрытый слой состоял из 16 нейронов, с функцией активации relu
- Второй скрытый слой состоял из 16 нейронов, с функцией активации softmax
- Выходной слой состоял из 10 нейронов

## Условия тестирования:

- Выборка из 8500 изображений рукописных цифр
- Входной слой состоял из 784 нейронов, каждый из которых отвечал за пиксель в картинке  $28 \times 28$ , с функцией активации relu
- Первый скрытый слой состоял из 16 нейронов, с функцией активации relu
- Второй скрытый слой состоял из 16 нейронов, с функцией активации softmax
- Выходной слой состоял из 10 нейронов
- Количество эпох (итераций обучения) = 3000

## Условия тестирования:

- Выборка из 8500 изображений рукописных цифр
- Входной слой состоял из 784 нейронов, каждый из которых отвечал за пиксель в картинке  $28 \times 28$ , с функцией активации relu
- Первый скрытый слой состоял из 16 нейронов, с функцией активации relu
- Второй скрытый слой состоял из 16 нейронов, с функцией активации softmax
- Выходной слой состоял из 10 нейронов
- Количество эпох (итераций обучения) = 3000
- Learning rate (шаг градиентного спуска) = 0.6

## Условия тестирования:

- Выборка из 8500 изображений рукописных цифр
- Входной слой состоял из 784 нейронов, каждый из которых отвечал за пиксель в картинке  $28 \times 28$ , с функцией активации relu
- Первый скрытый слой состоял из 16 нейронов, с функцией активации relu
- Второй скрытый слой состоял из 16 нейронов, с функцией активации softmax
- Выходной слой состоял из 10 нейронов
- Количество эпох (итераций обучения) = 3000
- Learning rate (шаг градиентного спуска) = 0.6
- Размер батча 128

## Условия тестирования:

- Выборка из 8500 изображений рукописных цифр
- Входной слой состоял из 784 нейронов, каждый из которых отвечал за пиксель в картинке  $28 \times 28$ , с функцией активации relu
- Первый скрытый слой состоял из 16 нейронов, с функцией активации relu
- Второй скрытый слой состоял из 16 нейронов, с функцией активации softmax
- Выходной слой состоял из 10 нейронов
- Количество эпох (итераций обучения) = 3000
- Learning rate (шаг градиентного спуска) = 0.6
- Размер батча 128
- Функция потерь MSE



Спасибо за внимание!

