

# ВВЕДЕНИЕ В НЕЙРОННЫЕ СЕТИ

ЗАНЯТИЕ #1



# ПЛАН ЗАНЯТИЯ

---



- Создание простой нейронной сети
- Активационные функции
- Функции ошибок
- Оптимайзеры
- Градиентный спуск
- Нейронная сеть для распознавания рукописных цифр

# СОЗДАНИЕ ПРОСТОЙ НЕЙРОННОЙ СЕТИ

```
model = Sequential()  
model.add(Dense(2, input_dim=2, use_bias=False))  
model.add(Dense(1, use_bias=False))
```

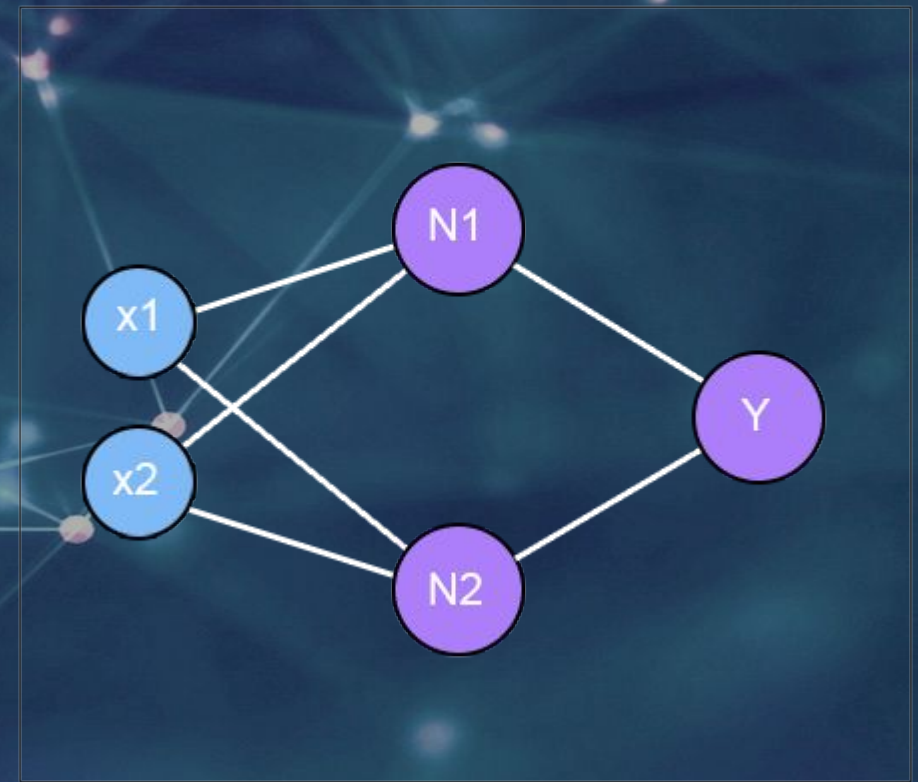
Расчет значений нейронов скрытого слоя:

$$N1 = x1 * w1 + x2 * w2$$

$$N2 = x1 * w3 + x2 * w4$$

Расчет выхода сети:

$$Y = N1 * w5 + N2 * w6$$





# ФУНКЦИИ АКТИВАЦИИ

<https://keras.io/activations/>

Применение функции к выходу нейрона:

$$N1 = x1 * w1 + x2 * w2 \Rightarrow N1 = f(x) = f(x1 * w1 + x2 * w2)$$

- sigmoid (сигмоид):

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

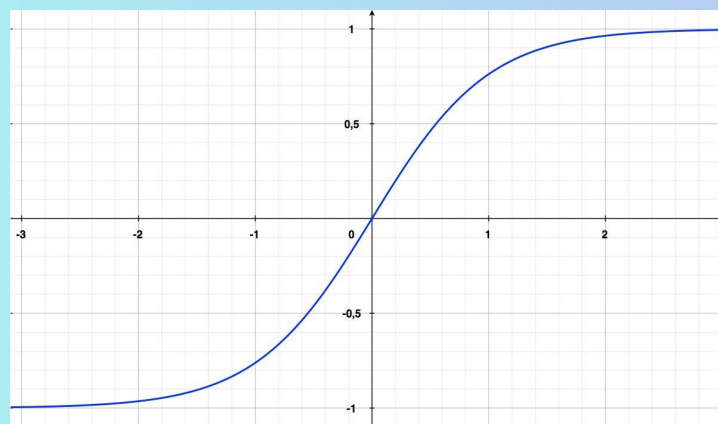
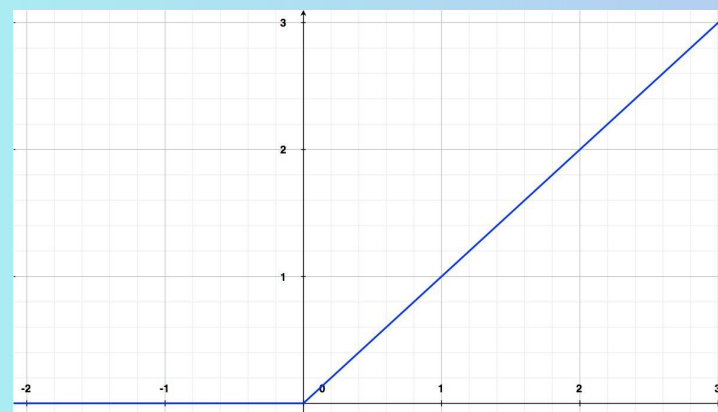
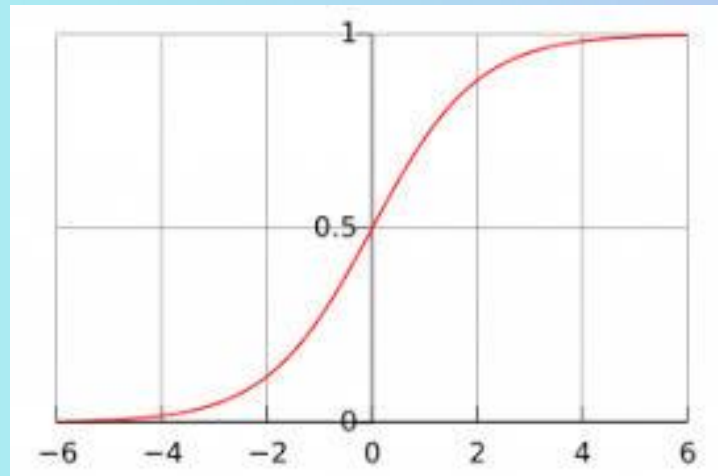
- relu (линейный выпрямитель):

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

- tanh(гиперболический тангенс):

$$f(x) = \text{th}(x) = \frac{(e^x - e^{-x})}{(e^x + e^{-x})}$$

**Варианты:** elu, softmax, selu, softplus, softsign, hard\_sigmoid, exponential, LeakyReLU, PReLU, ELU, ThresholdedReLU, Softmax, ReLU



# ФУНКЦИИ ОШИБОК

$y_{\text{true}}$  vs  $y_{\text{pred}}$

<https://keras.io/losses/>

- средняя абсолютная ошибка (mean absolute error) (MAE):

$$\text{MAE} = \frac{\sum_{i=1}^n |y_i - x_i|}{n} = \frac{\sum_{i=1}^n |e_i|}{n}.$$

- среднеквадратическая ошибка (mean squared error) (MSE):

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2.$$

- бинарная кроссэнтропия (binary crossentropy):

$$H_p(q) = -\frac{1}{N} \sum_{i=1}^N y_i \cdot \log(p(y_i)) + (1 - y_i) \cdot \log(1 - p(y_i))$$

**Варианты:** mean\_absolute\_percentage\_error, hinge, mean\_squared\_logarithmic\_error, squared\_hinge, kullback\_leibler\_divergence, poisson, categorical\_hinge, huber\_loss, logcosh, sparse\_categorical\_crossentropy, cosine\_proximity, is\_categorical\_crossentropy



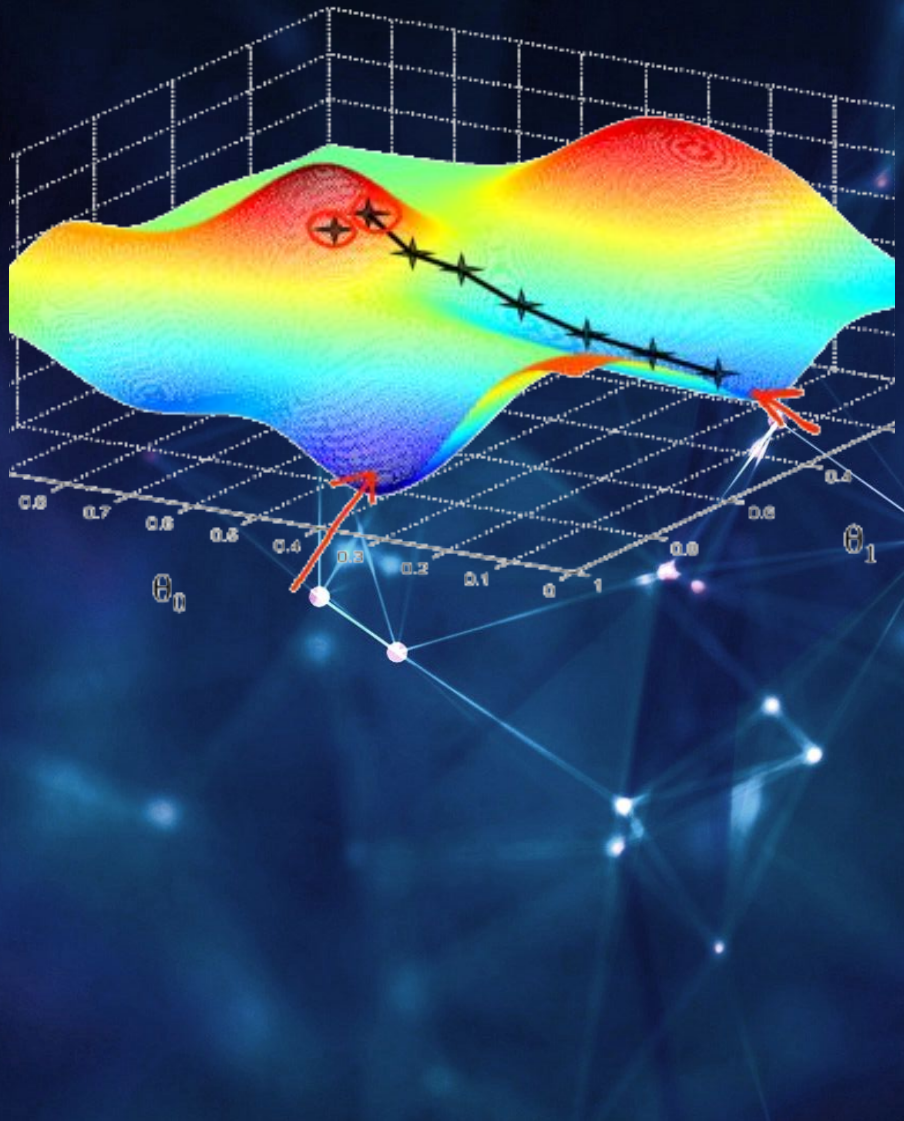
# ОПТИМАЙЗЕРЫ

<https://keras.io/optimizers/>

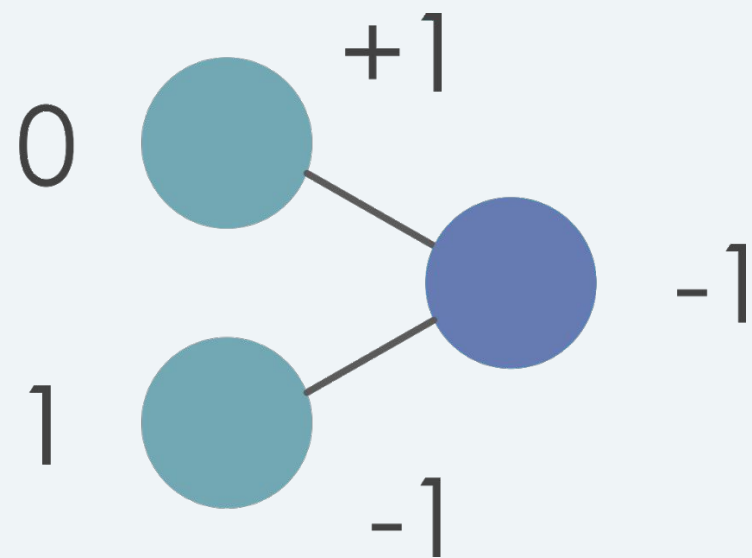
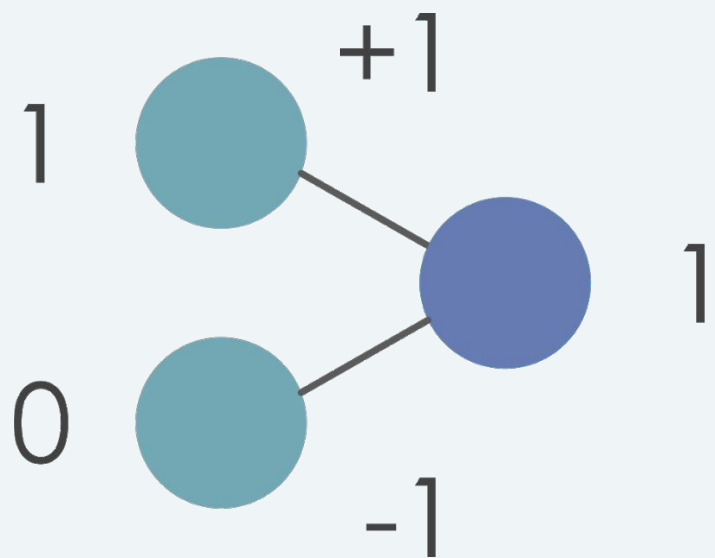
- Adam
- RMSprop
- Adadelta

**learning\_rate** - шаг оптимизатора

**Варианты:** SGD, Adagrad, Adamax, Nadam



# ДЕМОНСТРАЦИЯ ГРАДИЕНТНОГО СПУСКА





ДЕМОНСТРАЦИЯ  
**НЕЙРОННОЙ**  
**СЕТИ**

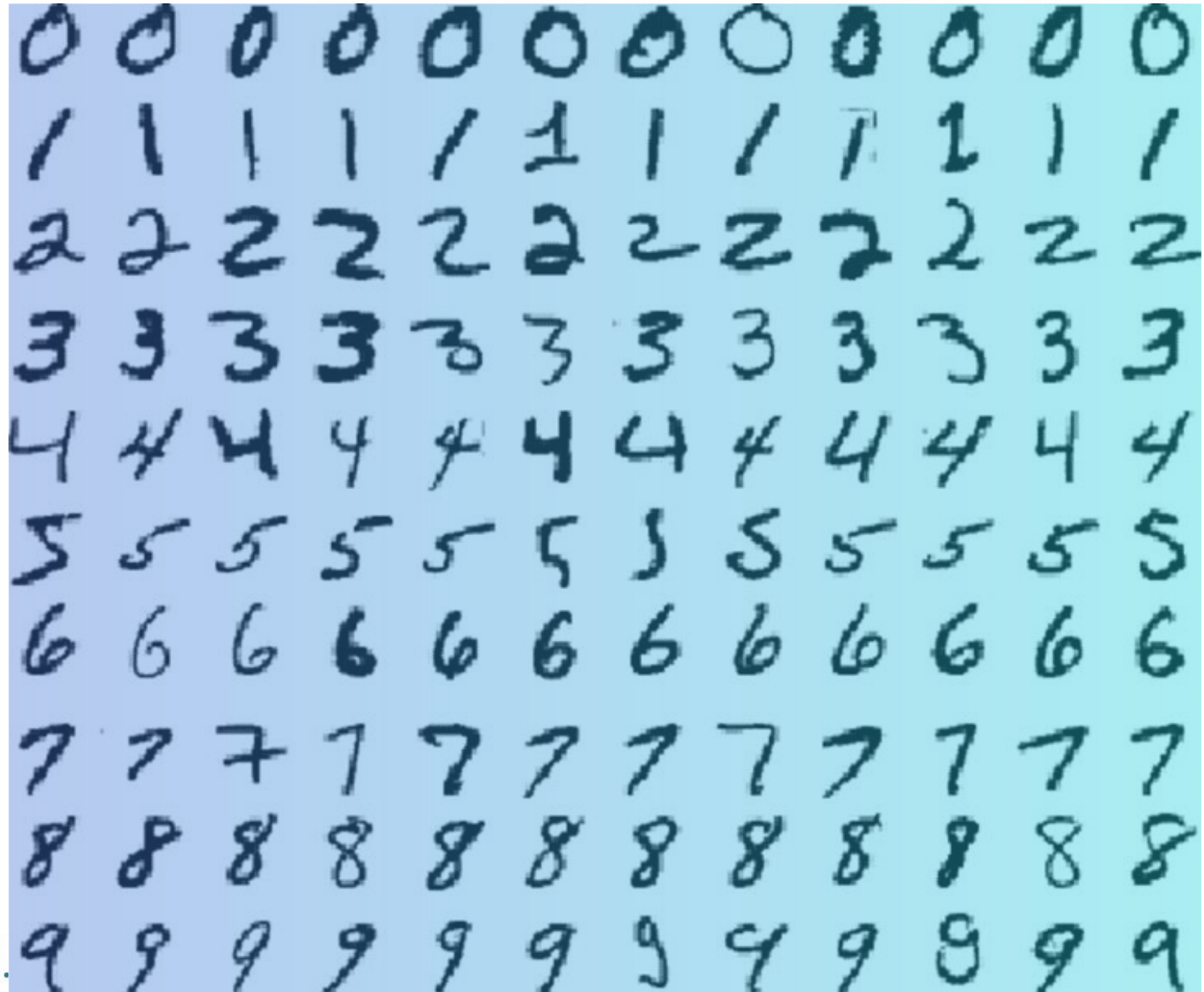


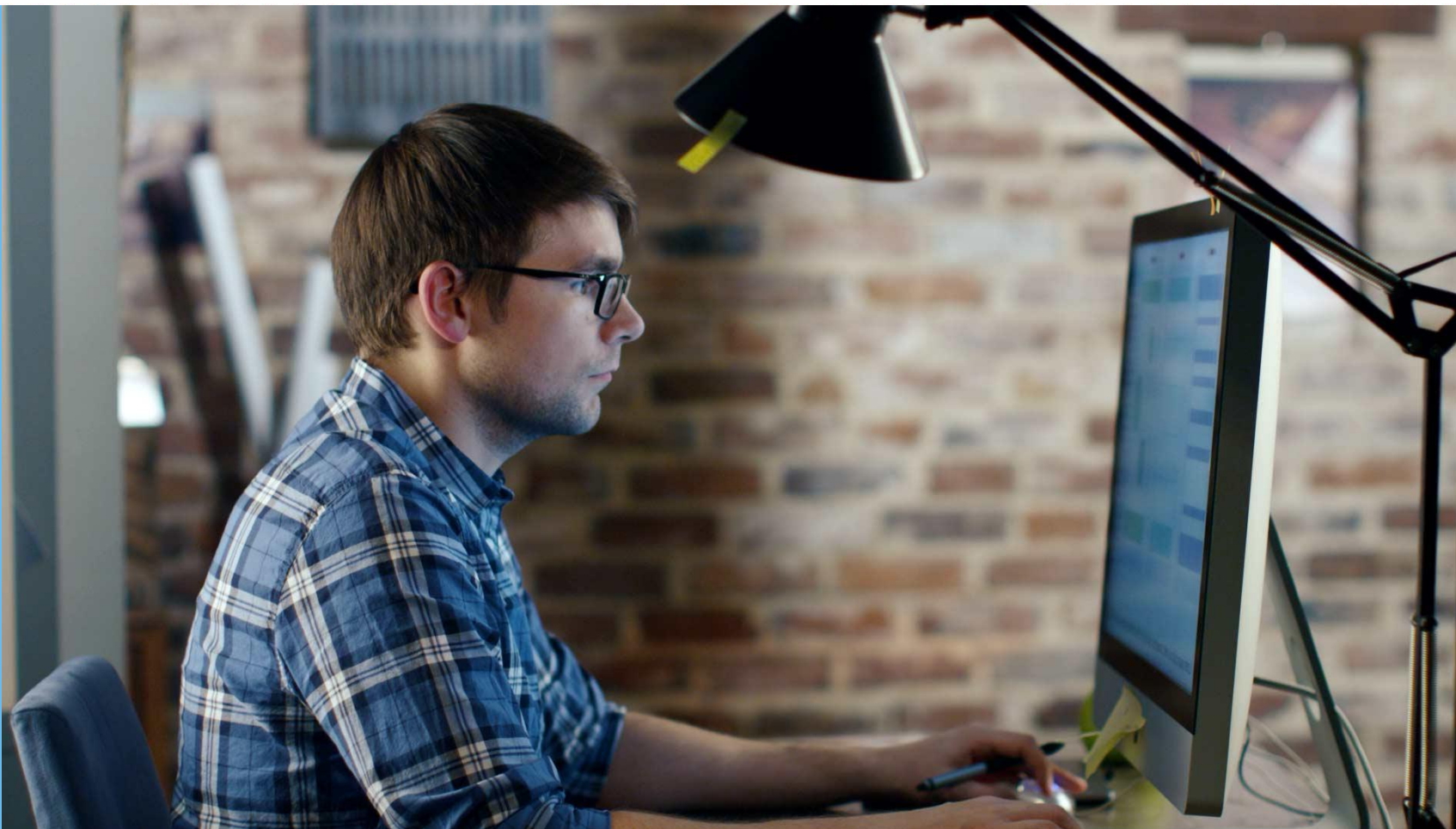


# Набор данных MNIST

Mixed National Institute  
of Standards and  
Technology database

Back-Propagation Applied to  
Handwritten Zip Code  
Recognition/ Y. LeCun, B. Boser,  
J. S. Denker et al. 1989





# ДОМАШНЕЕ ЗАДАНИЕ



СПАСИБО

ЗА ВНИМАНИЕ