

# **ИНТЕЛЛЕКТУАЛЬНЫЕ МЕТОДЫ В ЗАДАЧАХ ЗАЩИТЫ ОТ КИБЕРУГРОЗ**

## **ЛАБОРАТОРНАЯ РАБОТА № 4**

**Распознавание рукописных цифр с помощью Keras.**

**Выполнил:**  
**Мосолков Е.Н.**  
**Преподаватель:**  
**Петров А.А.**

Москва 2021 г.

## **ЦЕЛЬ РАБОТЫ**

Обучить НС выполнять классификацию цифр на изображениях.

## ЗАДАЧА ПРАКТИЧЕСКОЙ РАБОТЫ

Собрать приведенный исходный код в единое целое, поработать с параметром НС.  
Изменить можно:

- Распознаваемое изображение и проверить качество (варьируя параметр  $n$  при указании номера из тестовой выборки).
- Варьировать число нейронов на разных слоях и посмотреть как НС будет реагировать результатами распознавания.
- Изменить количество слоёв.

## КОД И ОПИСАНИЕ ЕГО КЛЮЧЕВЫХ ВЕЩЕЙ

```
from tensorflow.keras.layers import Flatten, Dense
from tensorflow.keras.datasets import mnist
from matplotlib import pyplot as plt
import numpy as np
from tensorflow import keras

(x_train, y_train), (x_test, y_test) = mnist.load_data()

plt.figure(figsize=(10,5))
for i in range(25):
    plt.subplot(5,5,i+1)
    plt.xticks([])
    plt.yticks([])
    plt.imshow(x_train[i], cmap=plt.cm.binary)
plt.show()

model = keras.Sequential({
    Flatten(input_shape=(28, 28, 1)),
    Dense(128, activation='relu'),
    Dense(10, activation='softmax')
})
print(model.summary()) # вывод структуры НС в консоль

x_train = x_train / 255
x_test = x_test / 255

y_train_cat = keras.utils.to_categorical(y_train, 10)
y_test_cat = keras.utils.to_categorical(y_test, 10)

model.compile(optimizer='adam',
loss='categorical_crossentropy',
metrics=['accuracy'])

model.fit(x_train, y_train_cat, batch_size=32, epochs=10,
validation_split=0.2)

model.evaluate(x_test, y_test_cat)

n = 1010
x = np.expand_dims(x_test[n], axis=0)
res = model.predict(x)
print( res )

print( np.argmax(res) )

plt.imshow(x_test[n], cmap=plt.cm.binary)
plt.show()

pred = model.predict(x_test)
pred = np.argmax(pred, axis=1)
print(pred.shape)
```

```
print(pred[:20])
print(y_test[:20])

mask = pred == y_test
print(mask[:10])
x_false = x_test[~mask]
y_false = x_test[~mask]
print(x_false.shape)
for i in range(5):
    print("Значение сети: "+str(y_test[i]))
    plt.imshow(x_false[i], cmap=plt.cm.binary)
    plt.show()
```

## ТЕСТИРОВАНИЕ НЕЙРОСЕТИ

Без модификаций НС показывает хорошие результаты:

На 10 эпохе значение точности близко к 1 (0.9959), а потеря близка к 0 (0.0148).

2.1. Добавил слой с активатором ReLu на 64 нейрона между двумя, которые уже были, показатели НС существенно улучшились: Точность: 0.9939, потеря: 0.0184.

2.2. Увеличил в два раза количество нейронов первого ReLu слоя в исходной НС, значения тоже улучшились: точность: 0.9967, потеря: 0.0107.

2.3. Уменьшил в два раза количество нейронов первого ReLu слоя в исходной НС, несущественно ухудшилось: точность: 0.9905, потеря: 0.0306.

Переберем случайные изображения, для этого переменную  $n$  будем инициализировать каждый раз случайными значениями:

При  $n = 344$ , НС верно предсказала цифру: 8.

При  $n = 1010$ , НС верно предсказала цифру: 4.

При  $n = 239$ , НС верно предсказала цифру: 1.

С точностью  $> 0.99$  сложно найти ошибку.

## **ВЫВОД**

В процессе выполнения лабораторной работы я поработал с НС из библиотеки keras для распознавания рукописных цифр из датасета MNIST.