**Лабораторная работа №7. Распознавание рукописных цифр.**

MNIST – база рукописных цифр. Содержит в себе 60000 изображений цифр для обучающей выборки и 10000 для проверочной.

Используя базу рукописных цифр MNIST, создайте нейронную сеть для распознавания цифр.

При загрузке базы (рекомендации 1 и 2) опишите, за что отвечает каждый из массивов, какие данные он содержит.

Отобразите графики точности и ошибки в процессе обучения.

Проведите серию экспериментов:

1. Используйте в первом скрытом слое 10/100/5000 нейронов;
2. Используйте размер пакета данных обучения – 1/10/100/60000 (вся база).

В текстовом документе зафиксируйте все наблюдения, с примерами графиков.

Случайным образом из всей выборки выберите 10 изображений, выведите их на экран и возле них выведите рассчитанное значение из нейронной сети.

Модифицируйте ноутбук таким образом, чтобы он мог считывать изображения из файлов. Создайте в графическом редакторе изображение с цифрой, и попробуйте её распознать нейронной сетью.

Попробуйте загрузить рукописную цифру.

Подпишите, для чего был импортирован каждый из модулей (если какому-то из них не нашлось применения, удалите).

**Рекомендации к выполнению лабораторной работы:**

Импортируйте модули

from tensorflow.keras.datasets import mnist

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Dense

from tensorflow.keras.optimizers import Adam

from tensorflow.keras import utils

from tensorflow.keras.preprocessing import image

import numpy as np import matplotlib.pyplot as plt

from PIL import Image

from numpy import asarray

1. Загрузите данные в следующем виде:

(x\_train\_org, y\_train\_org), (x\_test\_org, y\_test\_org) = mnist.load\_data()

1. y\_train\_org преобразуйте в массив значений категорий:

y\_train = utils.to\_categorical(y\_train\_org, 10); изучите проведённое преобразование (т.е. по какому принципу сформировался новый массив?)

1. Входной слой лучше всего делать одномерным. База MNIST содержит изображения 28х28 пикселей – соответственно, входной слой размерностью по числу пикселей.
2. Начните с трёх полносвязных слоёв (800-400-10 нейронов);
3. Используйте relu (для первых двух) и softmax (для последнего)-функции активации;
4. В качестве оптимизатора выберите Adam, в качестве функции потерь – caterogical\_crossentropy (разбиение по категориям);
5. Распознаватель нейронной сети принимает массив примеров; в случае одного примера его массив нужно расширить до двухмерного, например:

print(x.shape)

(784,)

x = np.expand\_dims(x, axis=0)

print(x.shape)

(1,784)

1. При загрузке картинок воспользуйтесь их переводом в серые тона, переведите изображение в массив, проведите нормализацию (чтобы значения цветов были в пределах от 0 до 1) и сделайте его одномерным.

Преобразование в массив изображений из файла можно сделать с помощью load\_img\_array = asarray(im) или любым другим подходящим методом.

1. Результат распознавания должен выдавать массив из 10 значений. Элемент массива с наибольшим индексом будет соответствовать конечной цифре.
2. В этой и следующих лабораторных работах – в случае долгого обучения сети можете воспользоваться сохранением полученных весов, чтобы потом быстро их загрузить, а не обучать сеть по новой.