Федеральное агентство по образованию

Пермский национальный исследовательский политехнический университет

Кафедра вычислительной математики и механики

ЛАБОРАТОРНАЯ РАБОТА № 5

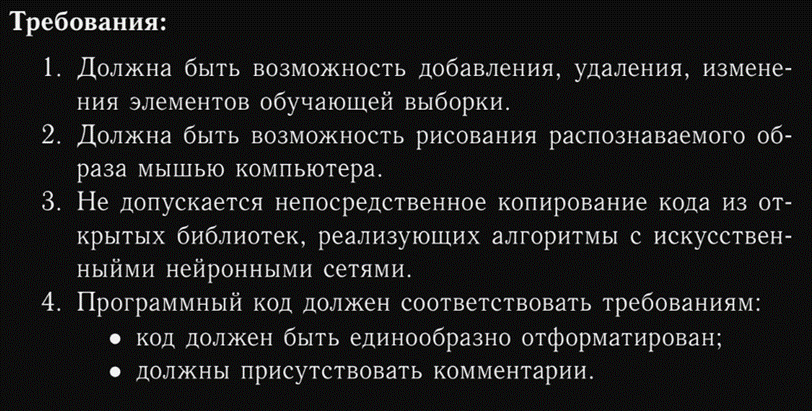
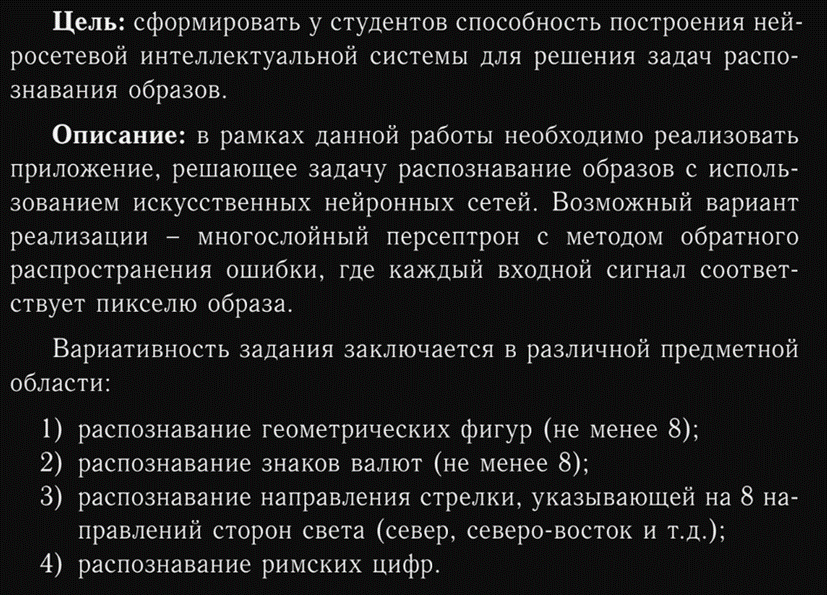
Реализация нейронных сетей для решения задачи

распознавания образов

Выполнил:

студент гр. ИСТ-19-2б А.А. Михайлишин

Пермь 2022



1. **Используемые технологии**

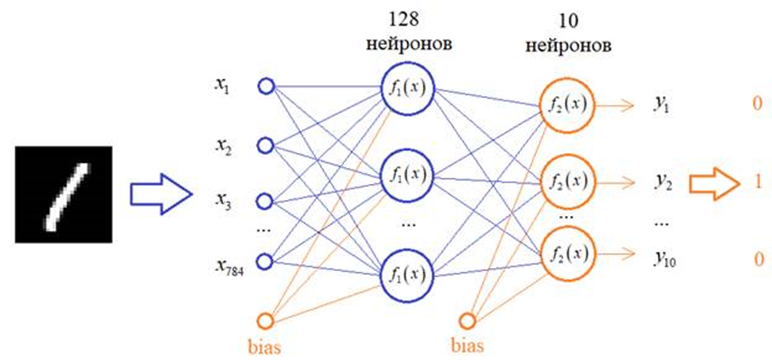
БД MNIST:

MNIST – (сокращение от «Modified National Institute of Standards and Technology») – база данных образцов рукописного написания цифр.

Tensorflow: библиотека для построения нейронной сети

ImageIO: библиотека для работы с изображениями

Numpy: библиотека для работы с массивами



На рисунке изображена схема сети.

1. **Модель Нейронной Сети**

Моя модель, состоит из 7 слоёв:

1. Слой двумерной свёртки, с размером фильтра 5х5.
2. Слой объединения данных, с размером 2х2
3. Слой двумерной свёртки, с размером фильтра 5х5.
4. Слой объединения данных, с размером 2х2
5. Слой выравнивания ввода
6. Слой dropout (разрушения связей)
7. Обычный плотно связанный слой из 1024 нейронов
8. Выходной слой из 10 нейронов
9. **Обучение Нейронной Сети**

Обучения проводится на базе данных MNIST, в 5 эпох (этапов) и размером партии около 2000. По завершению обучения и прогонки тестовых данных, мы получаем точность примерно 89% для тестовых данных.

**4. Код нейронной сети на языке Python**

import keras  
from keras.datasets import mnist  
from tensorflow.python.ops import nn  
  
(x\_train, y\_train), (x\_test, y\_test) = mnist.load\_data()  
  
num\_classes = 10  
x\_train = x\_train.reshape(x\_train.shape[0], 28, 28, 1)  
x\_test = x\_test.reshape(x\_test.shape[0], 28, 28, 1)  
  
y\_train = keras.utils.to\_categorical(y\_train, num\_classes)  
y\_test = keras.utils.to\_categorical(y\_test, num\_classes)  
  
x\_train = x\_train.astype('float32')  
x\_test = x\_test.astype('float32')  
x\_train /= 255  
x\_test /= 255  
  
epochs = 5  
  
model = keras.models.Sequential([  
 keras.layers.Conv2D(  
 input\_shape=(28, 28, 1),  
 filters=32,  
 kernel\_size=(5, 5),  
 padding='same',  
 activation='relu'  
 ),  
 keras.layers.MaxPool2D(pool\_size=(2, 2)),  
 keras.layers.Conv2D(  
 filters=64,  
 kernel\_size=(5, 5),  
 padding='same',  
 activation='relu'  
 ),  
 keras.layers.MaxPool2D(pool\_size=(2, 2)),  
 keras.layers.Flatten(),  
 keras.layers.Dense(1024, activation=nn.relu),  
 keras.layers.Dropout(0.2),  
 keras.layers.Dense(10, activation=nn.softmax)  
])  
  
model.compile(  
 optimizer=keras.optimizers.Adadelta(),  
 loss=keras.losses.categorical\_crossentropy,  
 metrics=['accuracy']  
)  
  
model.fit(  
 x\_train,  
 y\_train,  
 epochs=epochs,  
 validation\_data=(x\_test, y\_test))  
  
model.save('fit\_neural\_network.h5')  
print("Модель сохранена")

**5. Код предсказания нейронной сети на языке Python**

import imageio  
import numpy as np  
from keras.models import load\_model  
  
model = load\_model('fit\_neural\_network.h5')  
  
  
def predict(filename):  
 image = imageio.v3.imread(filename)  
 image = np.mean(image, 2, dtype=float)  
 image = image / 255  
  
 image = np.expand\_dims(image, 0)  
 image = np.expand\_dims(image, -1)  
  
 res = model.predict([image])[0]  
 return np.argmax(res), max(res)  
  
  
print("Модель загружена")