ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ  
«ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра АСУ

Отчет

о лабораторной работе №3

по дисциплине «Интеллектуальный анализ данных»

на тему: «Нейронные сети для решения задач классификации и регрессии»

Выполнил:

студент группы ИСТ-19а

Деркач К. И.

Проверили:

Васяева Т. А.

Шуватова Е. А.

Донецк – 2022

**Цель работы:** познакомиться с понятием полносвязной нейронной сети, с понятием метрик качества моделей обучения, функциями потерь и оптимизаторами обучения, с понятием «переобучение» модели нейронной сети; научиться строить с нуля и обучать нейронную сеть с помощью Keras и TensorFlow

Порядок выполнения работы

1. Выбрать одну из баз данных (согласовать с преподавателем), список которых представлен на сайте http://archive.ics.uci.edu/ml/.
2. Исходные данные должны храниться в файле. Программным способом считать эти данные и сформировать обучающую, проверочную и тестовую выборки. Обучающая выборка должна включать достаточное количество примеров для обучения. Тестовая выборка может быть вполовину меньше обучающей. Примеры из тестовой выборки не должны быть включены в обучающую. Для оценки обучения модели можно использовать проверочную (валидационную) выборку (при достаточном количестве исходных данных) или же использовать кросс-валидацию (если данных мало).
3. Разработать программу на языке Python с использованием библиотек Keras и TensorFlow. Программа должна обеспечивать: построение и обучение 64 НС; тестирование НС; вычисление точности и ошибки классификации; сохранение / считывание модели; вывод на экран структуру НС.
4. Согласно варианту заданий (табл. 3.2) построить модель НС таким образом, чтобы точность на обучающей выборке была не 100%, но более чем 90% (желательно около 97-98%). В процессе экспериментов записывайте в таблицу характеристики и показатели точности различных моделей НС на обучающей и тестовой выборках в процессе подбора гиперпараметров.

**Индивидуальное задание:** Вариант 4, Классификация многоклассовая

**Описание набора данных**

С помощью камеры были получены изображения 13 611 зерен сухих бобов, из которых выведены параметры каждого зерна.

Задача классификации — определить сорт (из 7 исследуемых) зерна

Входные данные — 16 float значений, характеризующих размеры и форму (площадь, округлость и др.)

**Листинг**

import tensorflow as tf

from tensorflow import keras

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Dense, LeakyReLU

from sklearn.preprocessing import StandardScaler

from scipy.io.arff import loadarff

import matplotlib.pyplot as plt

from sklearn.model\_selection import train\_test\_split

import pandas as pd

from tensorflow.keras.utils import to\_categorical

from itertools import product

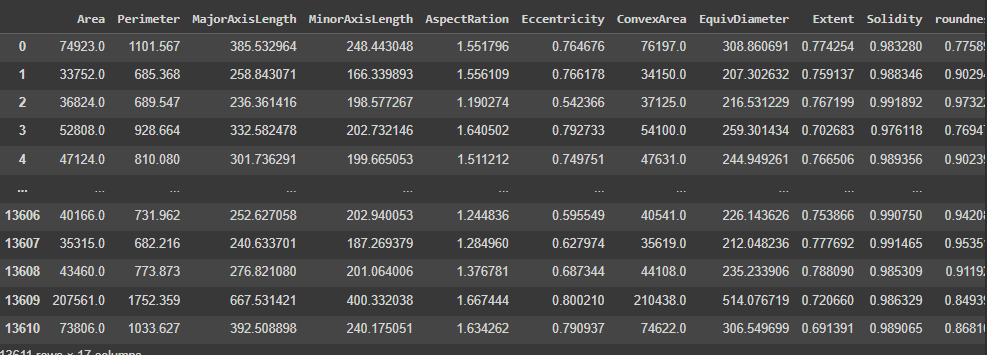
raw\_data = loadarff('Dry\_Bean\_Dataset.arff')

df = pd.DataFrame(raw\_data[0])

df.Class = pd.factorize(df.Class)[0]

df = df.sample(frac=1).reset\_index(drop=True)

df



y = df['Class']

x = df.drop('Class',axis=1)

x\_train, x\_test, y\_train, y\_test = train\_test\_split(x, y, test\_size=0.33, random\_state = 1)

scaler = StandardScaler()

x\_train = scaler.fit\_transform(x\_train)

x\_test = scaler.transform(x\_test)

y\_train = to\_categorical(y\_train, 7)

y\_test = to\_categorical(y\_test, 7)

activation = ["relu", LeakyReLU()]

loss = ["categorical\_crossentropy", "kullback\_leibler\_divergence"]

optimizer = ["SGD", "adam"]

res = pd.DataFrame(columns=[

'activation', 'loss', 'optimizer',

'Точность на обучающем','Точность на тестовом'

])

for a, l, o in product(activation, loss, optimizer):

model = Sequential()

model.add(Dense(units=28, input\_dim = x\_train.shape[1], activation = a))

model.add(Dense(7, activation = "softmax"))

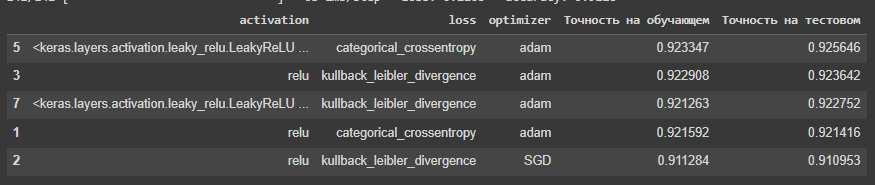
model.compile(loss=l, optimizer=o,metrics = ["accuracy"])

model.fit(x\_train, y\_train, validation\_split=0.2, epochs=10, batch\_size=50, verbose=1)

res.loc[res.shape[0]] = [a,l,o,model.evaluate(x\_train, y\_train)[1],model.evaluate(x\_test, y\_test)[1]]

res = res.sort\_values('Точность на тестовом', ascending=False)

res.head()



model = Sequential()

model.add(Dense(units=280, input\_dim = x\_train.shape[1], activation = res.iloc[0]['activation']))

model.add(Dense(7, activation = "softmax"))

model.compile(loss=res.iloc[0]['loss'], optimizer=res.iloc[0]['optimizer'],metrics = ["accuracy"])

model.fit(x\_train, y\_train, validation\_split=0.2, epochs=10, batch\_size=50, verbose=0)

model.save('/content/model')

print(f'Точность на тестовой выборке: {model.evaluate(x\_test, y\_test)[1]\*100:.3f}%')

print('\nСтруктура модели:')

model.summary()

