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

Шевченко М. В.

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 12, Бинарная классификация

**Описание набора данных**

Данные были извлечены из изображений, снятых с подлинных и поддельных образцов, похожих на банкноты. Для оцифровки использовалась промышленная камера, обычно используемая для проверки печати. Инструмент Wavelet Transform использовался для извлечения признаков из изображений.

Информация об атрибутах:

1. дисперсия изображения, преобразованного вейвлетом (непрерывное)

2. асимметрия изображения, преобразованного вейвлетом (непрерывное)

3. эксцесс изображения, преобразованного вейвлетом (непрерывное)

4. энтропия изображения (непрерывное)

5. класс (целое число)

**Листинг программы**

from tensorflow import keras

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Dense

import tensorflow as tf

import matplotlib.pyplot as plt

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

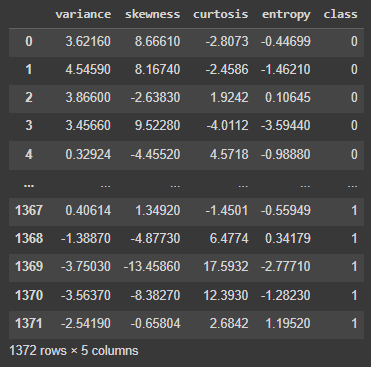
import pandas as pd

from tensorflow.keras.utils import to\_categorical

cols = ['variance ','skewness','curtosis','entropy', 'class']

df = pd.read\_csv('data\_banknote\_authentication.txt', names=cols, index\_col=False)

df



y = df['class']

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

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

activation = ['relu', 'softmax', 'tanh']

optimizer = ['adam', 'SGD']

loss = ['binary\_crossentropy','hinge', 'squared\_hinge']

batch\_size = [15, 32, 100]

t = pd.DataFrame(columns=[

'activation', 'optimizer', 'loss', 'batch\_size',

'train score','test score'

])

for a in activation:

for o in optimizer:

for l in loss:

for bs in batch\_size:

model = Sequential()

model.add(Dense(6, activation=a, input\_dim=train\_x.shape[1]))

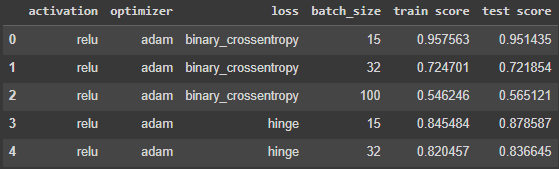
model.add(Dense(1, activation='sigmoid'))

model.compile(optimizer=o, loss=l,metrics=['binary\_accuracy'])

model.fit(train\_x, train\_y, validation\_split=0.2, epochs=10, batch\_size=bs, verbose=0)

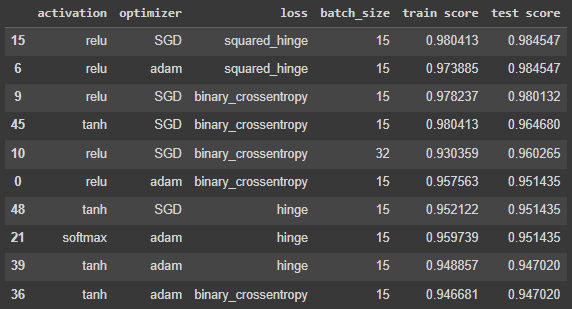
t.loc[t.shape[0]] = [a,o,l,bs,model.evaluate(train\_x, train\_y)[1],model.evaluate(test\_x, test\_y)[1]]

t.head()



t = t.sort\_values('test score', ascending=False)

t.head(10)



model = Sequential()

model.add(Dense(6, activation=t.iloc[0]['activation'], input\_dim=train\_x.shape[1]))

model.add(Dense(1, activation='sigmoid'))

model.compile(optimizer=t.iloc[0]['optimizer'], loss=t.iloc[0]['loss'],metrics=['binary\_accuracy'])

history = model.fit(train\_x, train\_y, validation\_split=0.2, epochs=10, batch\_size=t.iloc[0]['batch\_size'], verbose=0)

model.save('myModel')

model = keras.models.load\_model('myModel')

model.summary()

