МИНОБРНАУКИ РОССИИ

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ

ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)

Кафедра МО ЭВМ

ОТЧЁТ

по лабораторной работе №1

по дисциплине «Искусственные нейронные сети»

Тема: Многоклассовая классификация цветов

Студент гр.8382 _____ Фильцин И.В.

Преподаватель _____ Жангиров Т.Р.

Санкт-Петербург

2021

Цель работы

Реализовать классификацию сортов растения ирис (Iris Setosa - 0, Iris Versicolour - 1, Iris Virginica - 2) по четырем признакам: размерам пестиков и тычинок его цветков.

Задание

Ознакомиться с задачей классификации

Загрузить данные

Создать модель ИНС в Keras

Настроить параметры обучения

Обучить и оценить модель

Ход работы

Для анализа исходных данных выберем структуру ИНС. Для этого проведем исследование различных архитектур, сравнивая их показатели. Будем обучать сеть на 70 эпохах.

Рассмотрим первую архитектуру. Первый слой состоит из 4 нейронов, функция активации - relu, второй слой состоит из 3 нейронов, функция активации - softmax. Результаты приведены на рис. 1 и рис. 2

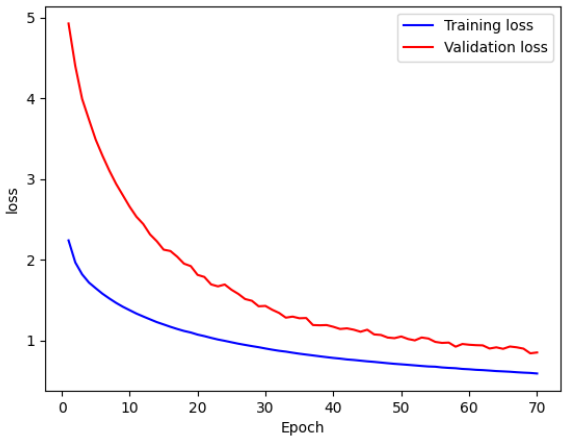

Рис. 1: "Арх.1"

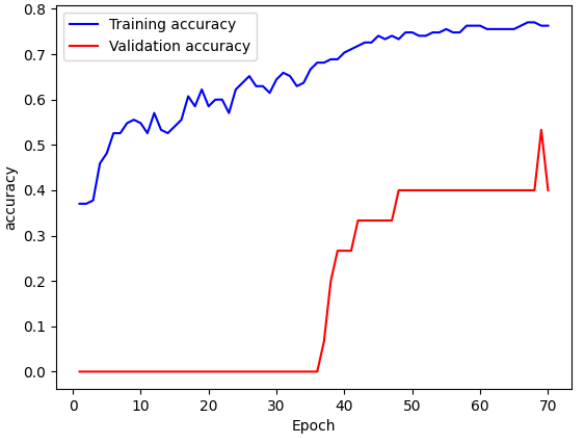

Рис. 2: "Арх.1"

На графиках видно, что сеть нормально обучается на тренировачных данных (но имеет все же малую точность), но на проверочных данных выдает очень слабые результаты.

Попробуем добавить еще 1 слой.

Рассмотрим вторую архитектуру. Первый слой состоит из 4 нейронов, функция активации - relu, второй слой состоит из 5 нейронов, функция активации - relu, третий слой состоит из 3 нейронов, функция активации - softmax.

В результате некоторых запусков сеть показала плохие результаты, не сумев сойтись за 70 эпох (см. рис. 3 и рис. 4).

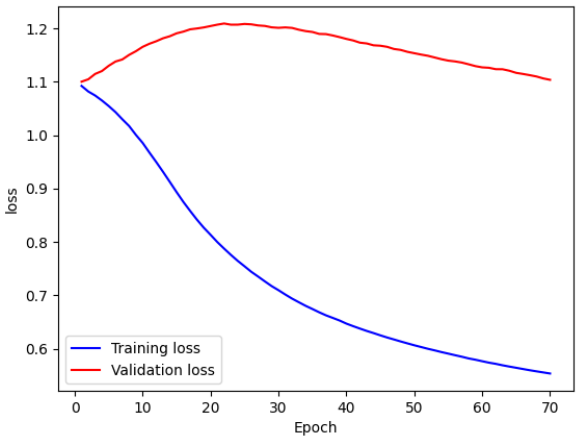

Рис. 3: "Арх.2"

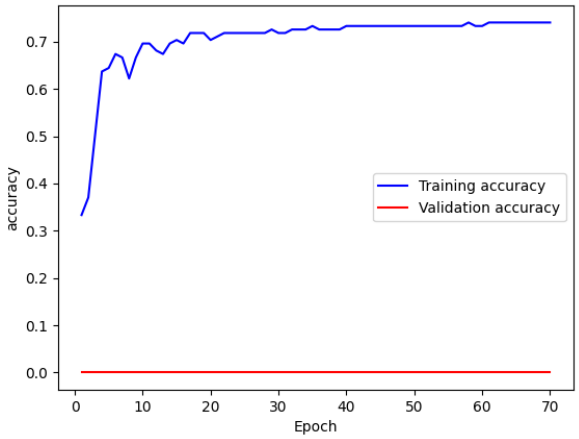

Рис. 4: "Арх.2"

В среднем точность на тестовой выборке 0.9, на проверочной - 0.7. Результаты на рис. 5 и рис. 6.

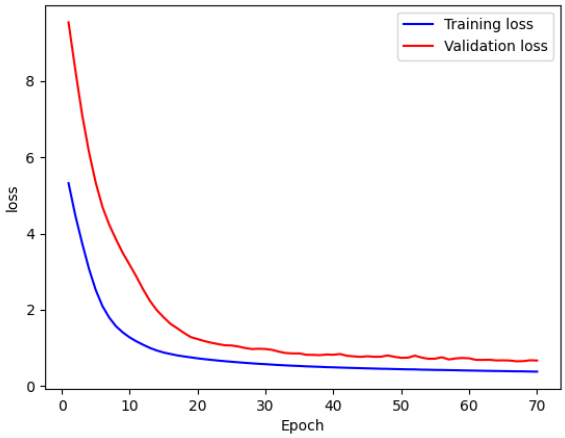

Рис. 5: "Арх.2"

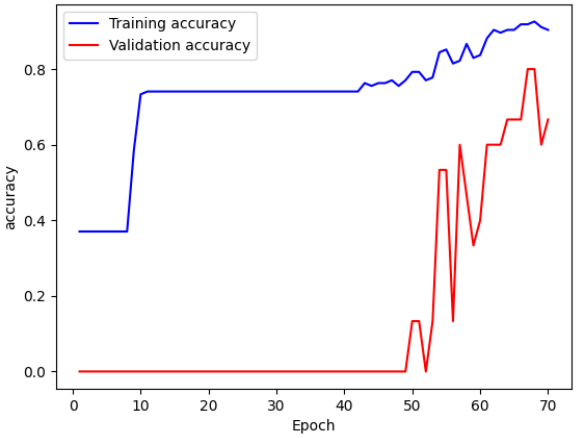

Рис. 6: "Арх.2"

Рассмотрим третью архитектуру. Первый слой состоит из 30 нейроной, функция активации - relu, третий слой состоит из 3 нейронов, функция активации - softmax. Результаты приведены на рис. 7 и рис. 8. Средняя точность на валидационных данных - 0.93, точность на тренируемом - 0.98.

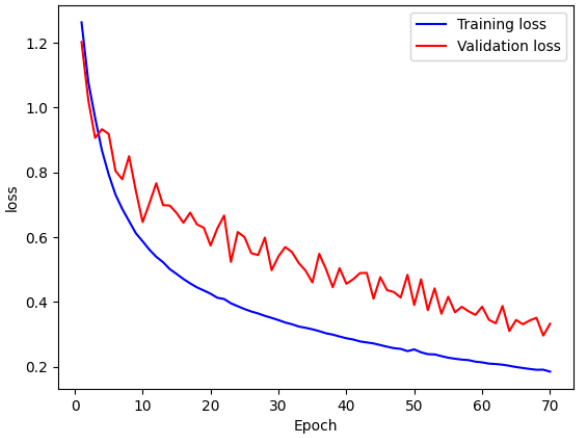

Рис. 7: "Арх.3"

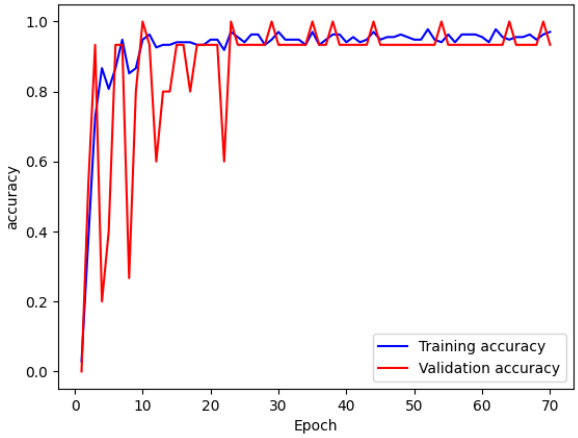

Рис. 8: "Арх.3"

Рассмотрим четвертую архитектуру. Первый слой состоит из 30 нейроной, функция активации - relu, второй слой состоит из 30 нейронов, функция активации - relu, третий слой состоит из 3 нейронов, функция активации - softmax. Результаты приведены на рис. 9 и рис. 10. Средняя точность на валидационных данных - 0.97, точность на тренируемом - 1.0.

Данная модель предоставляет достаточный уровень точности, поэтому остановимся на ней.

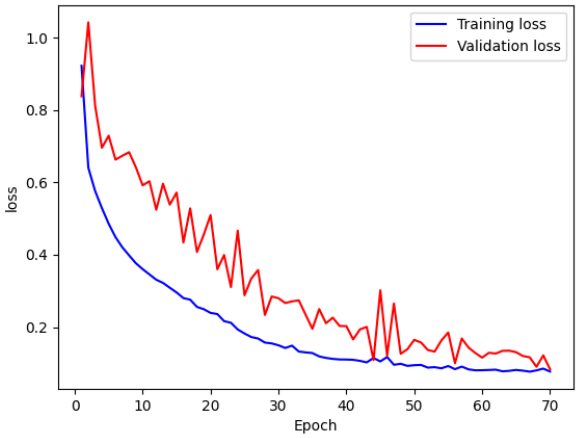

Рис. 9: "Арх.4"

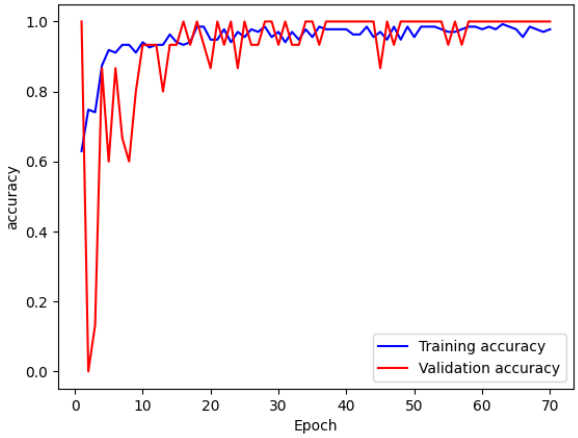

Рис. 10: "Арх.4"

Проведем тестирование обучения модели с различными параметрами. Увеличим размер батча с 10 до 20. Результаты на рис. 11 и рис. 12.

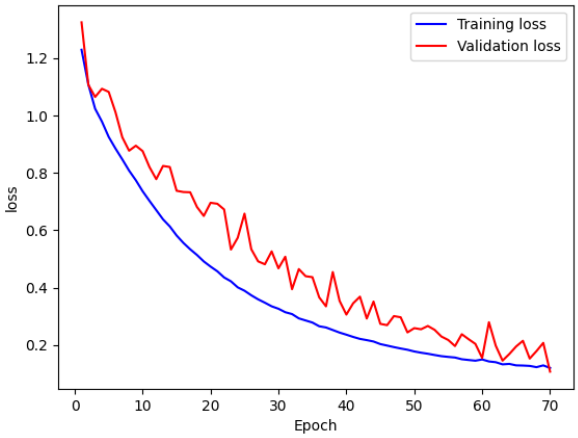

Рис. 11: "Увеличили размер батча"

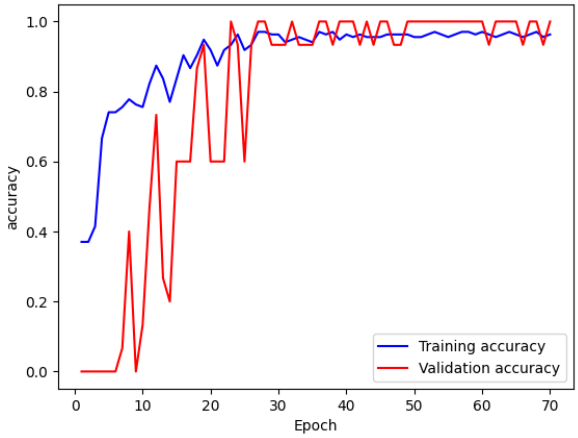

Рис. 12: "Увеличили размер батча"

Увеличим значение validation_split до 0.2. Можно заметить, что точность на валидационных данных упала до 0.7. Это, скорее всего, связано уже с недостаточностью данных для обучения сети. Результаты на рис. 13 и рис. 14.

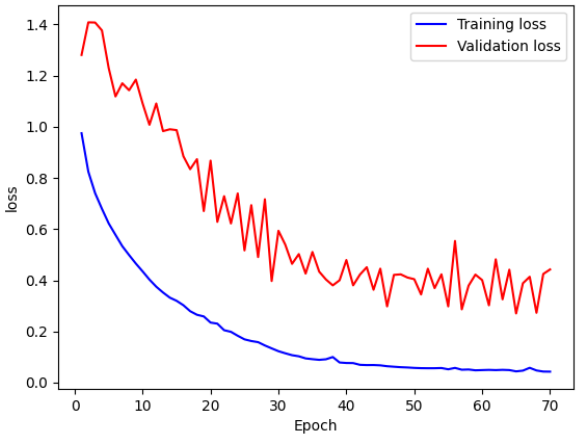

Рис. 13: "Увеличили размер split"

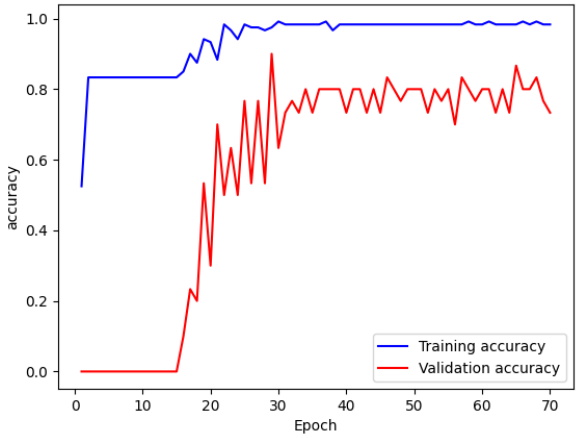

Рис. 14: "Увеличили размер split"

Вывод

В ходе лабораторной работы были получены практические навыки по созданию исскуственной нейронной сети для решения задачи многоклассовой классификации. Были исследованы различные архитектуры, проведено их взаимное сравнение. Также рассмотрено влияние параметров для обучения на результат.

Приложение А.

Исходный код

```
import pandas
from tensorflow.keras.layers import Dense, Input
from tensorflow.keras.models import Sequential
from tensorflow.keras.utils import to_categorical
from sklearn.preprocessing import LabelEncoder
import matplotlib.pyplot as plt
import matplotlib.colors as mclr

def draw_plot(xrange, training_data, validation_data, label):
    plt.plot(xrange, training_data, 'b', label='Training {}'.format(label))
    plt.plot(xrange, validation_data, 'r', label='Validation {}'.format(label))
    plt.xlabel('Epoch')
    plt.ylabel(label)
    plt.legend()
    plt.show()
    plt.clf()

def draw_result(history, nepochs):
    loss = history['loss']
    val_loss = history['val_loss']
    accuracy = history['accuracy']
    val_accuracy = history['val_accuracy']
    epochs = range(1, nepochs + 1)
    draw_plot(epochs, history['loss'], history['val_loss'], 'loss')
    draw_plot(epochs, history['accuracy'], history['val_accuracy'], 'accuracy')


dataframe = pandas.read_csv("iris.csv", header=None)
dataset = dataframe.values
X = dataset[:, 0:4].astype(float)
Y = dataset[:, 4]

encoder = LabelEncoder()
encoder.fit(Y)
encoded_Y = encoder.transform(Y)
dummy_y = to_categorical(encoded_Y)

nepochs = 70

model = Sequential()
model.add(Input(shape=(4, )))
```

```
model.add(Dense(30, activation='relu'))
model.add(Dense(30, activation='relu'))
model.add(Dense(3, activation='softmax'))
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
h = model.fit(X, dummy_y, epochs=nepochs, batch_size=10, validation_split=0.1)

model.summary()

draw_result(h.history, nepochs)
```