МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ

по лабораторной работе №1

по дисциплине «Искусственные нейронные сети»

Тема: «Многоклассовая классификация цветов»

Студент гр. 7383 _____ Сычевский Р.А.

Преподаватель _____ Жукова Н.А.

Санкт-Петербург

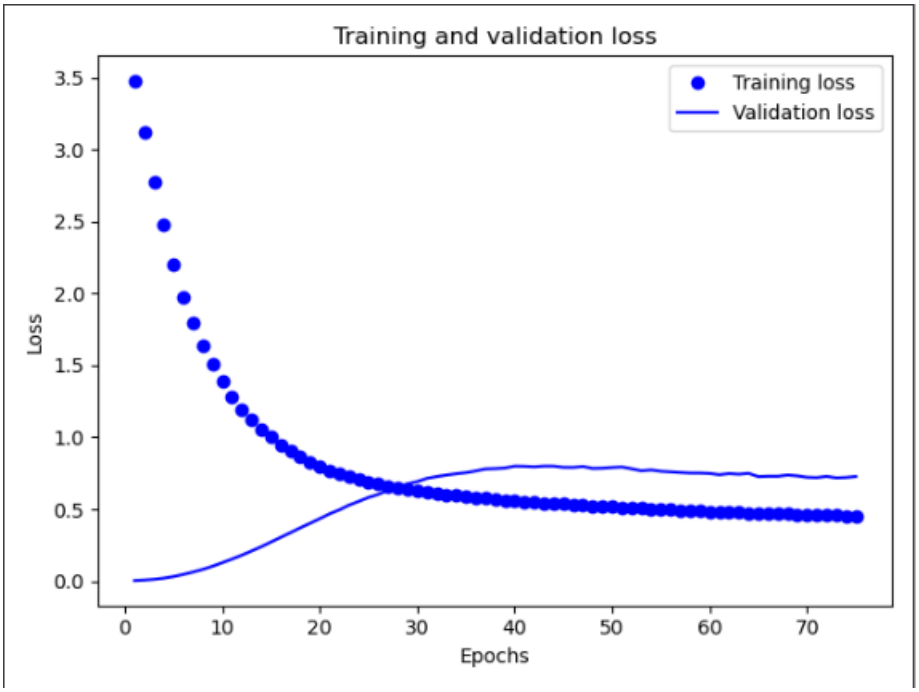
2020

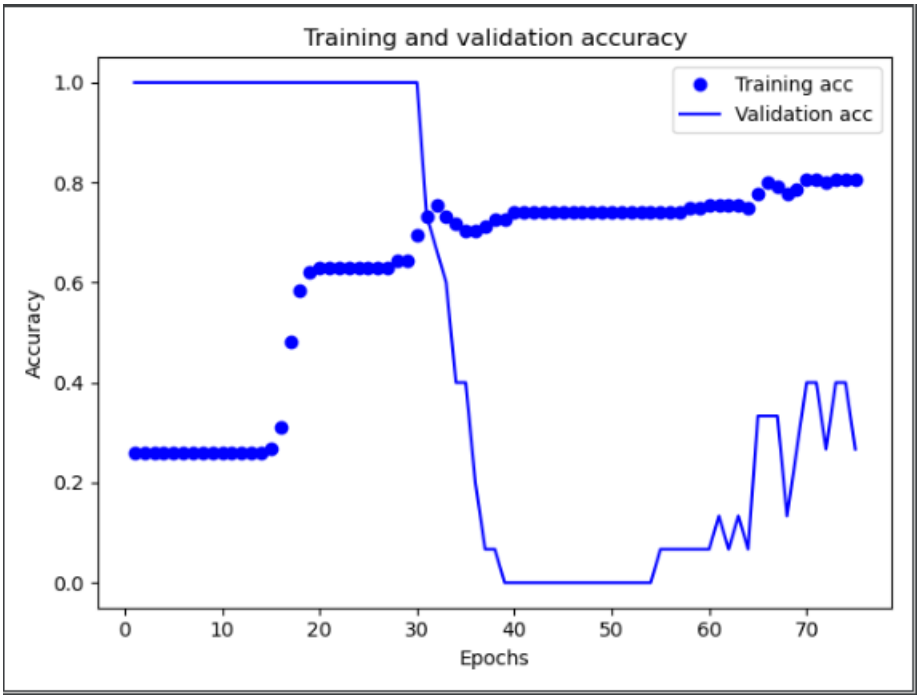
Цель работы.

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

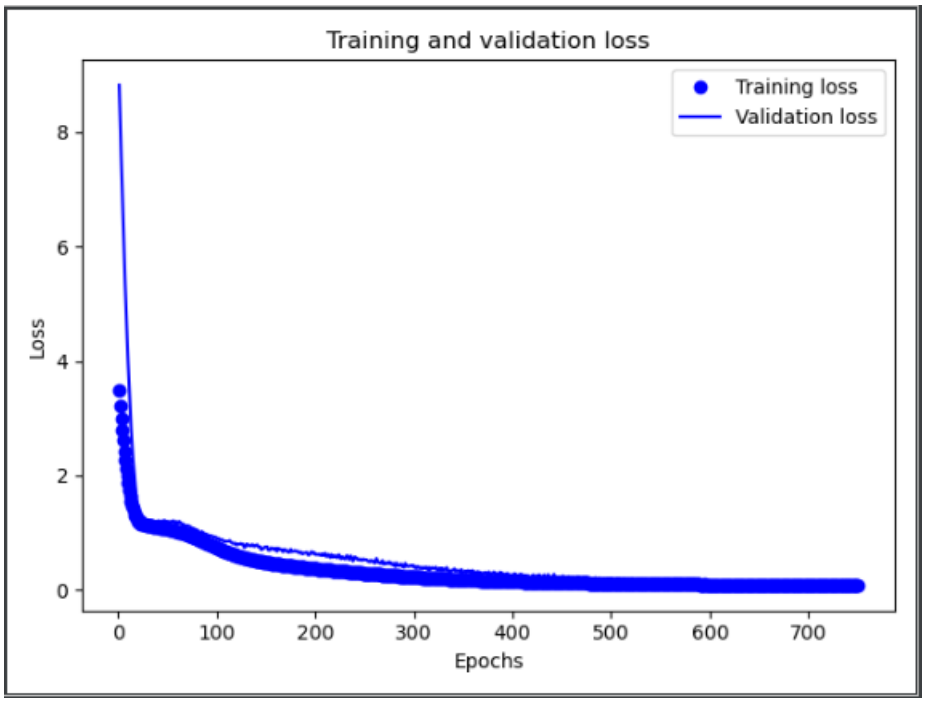
Ход работы.

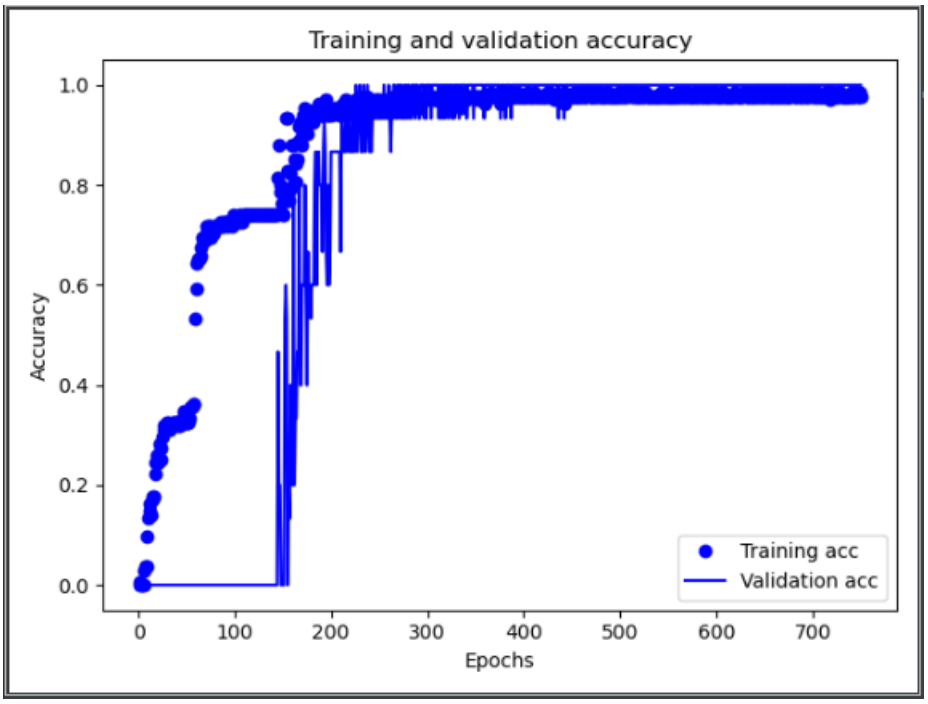
Изначальный вариант: количество эпох — 75, количество слоев — 2, количество нейронов в слое — 4. Наблюдаем:



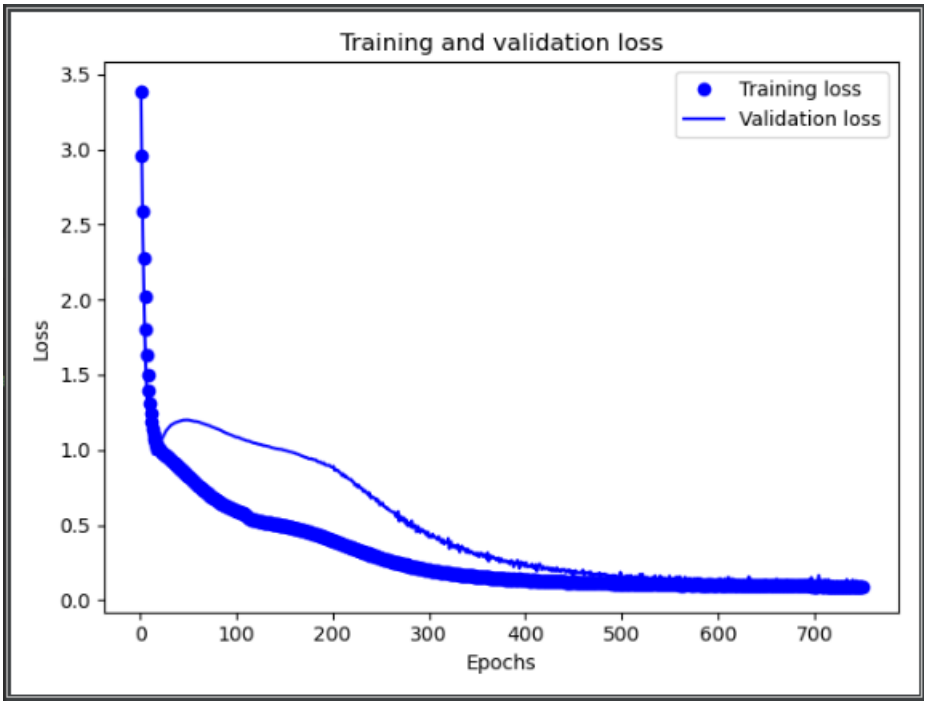


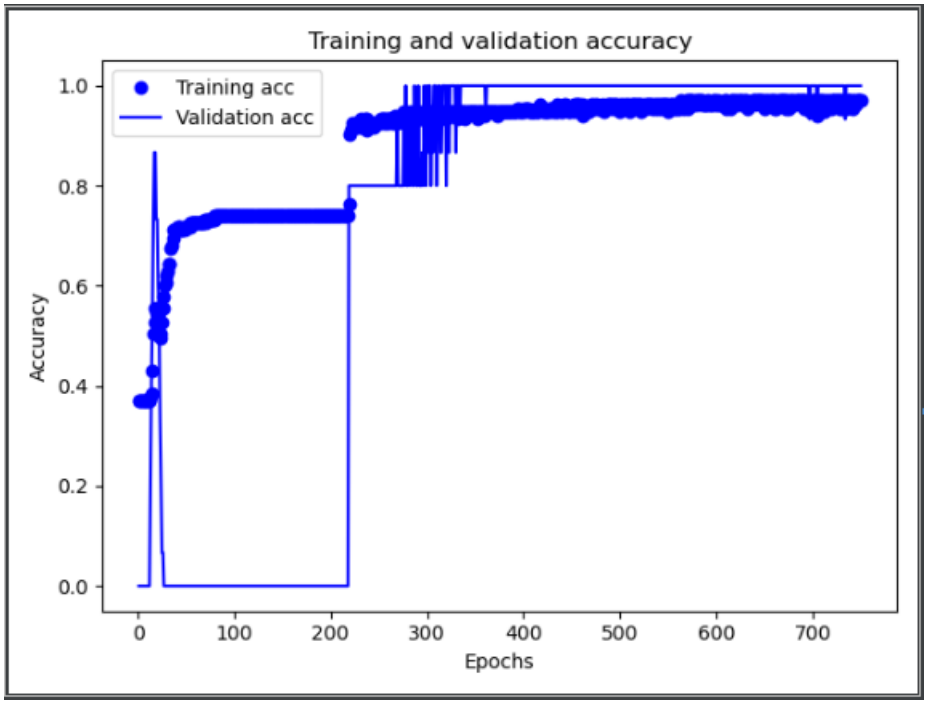
Вариант 2: количество эпох — 750, количество слоев — 2, количество нейронов в слое — 4. Наблюдаем:



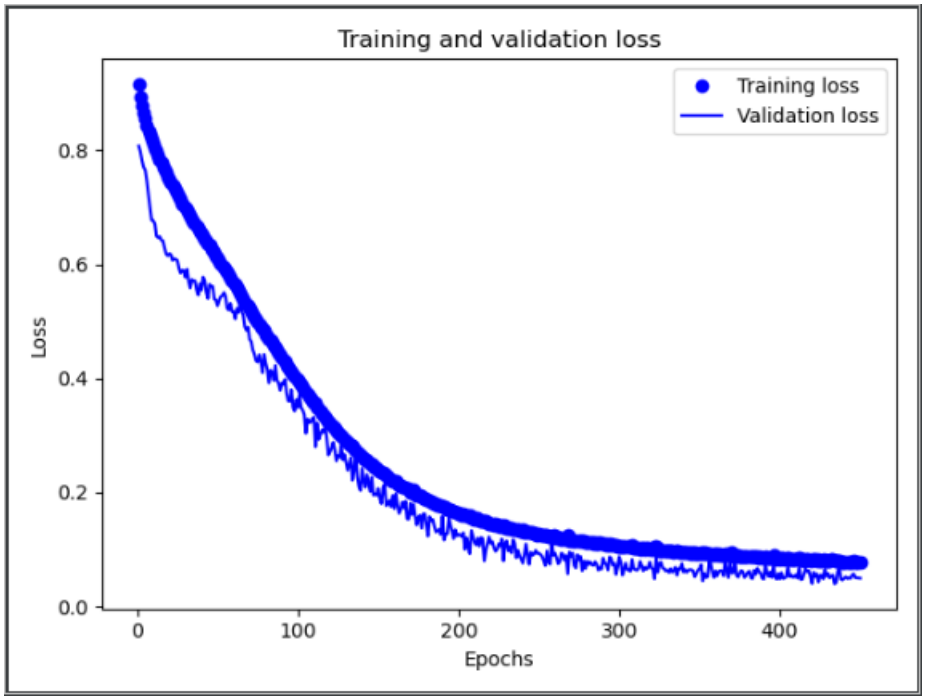


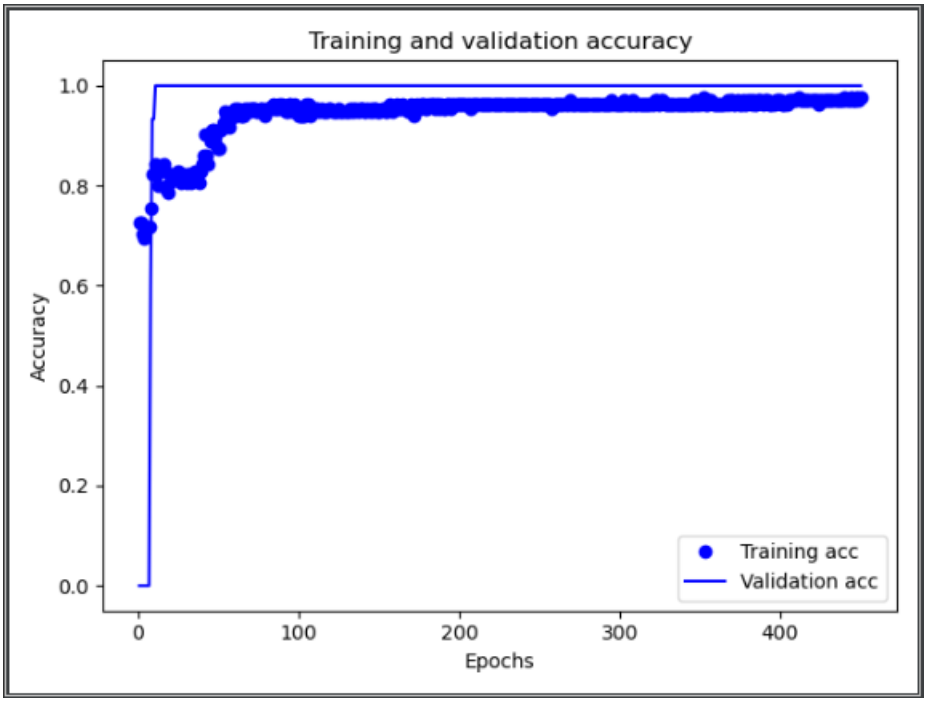
Вариант 3: количество эпох — 750, количество слоев — 3, количество нейронов в слое — 4. Наблюдаем:



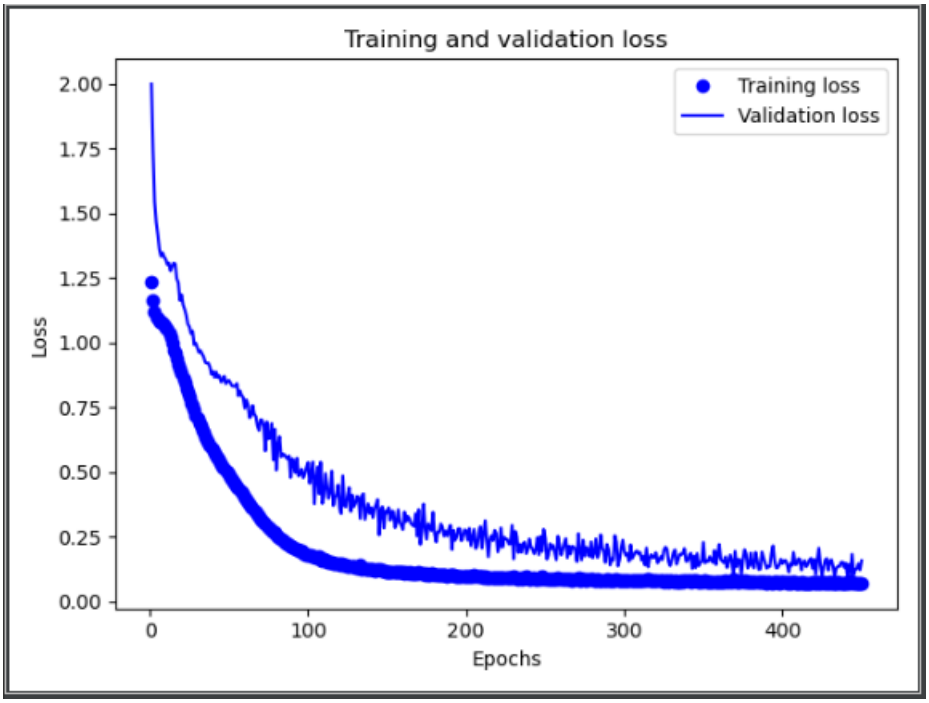


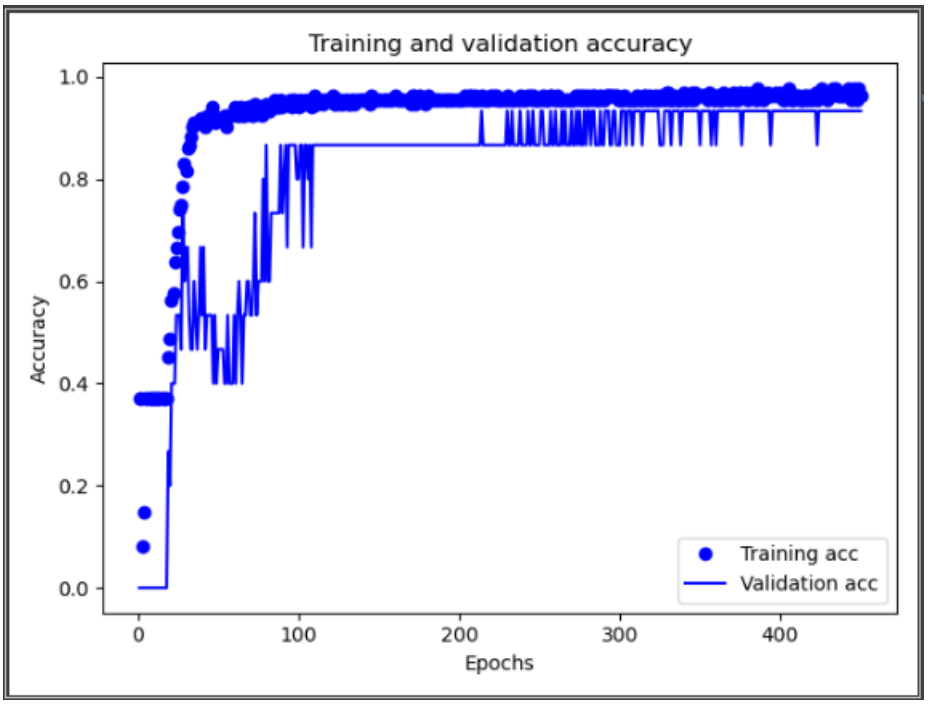
Вариант 4: количество эпох — 450, количество слоев — 3, количество нейронов в слое — 4. Наблюдаем:





Вариант 5: количество эпох — 450, количество слоев — 3, количество нейронов в слое — 8. Наблюдаем:





При дальнейшем увеличении количества слоев или количество нейронов в слоях результат не улучшался. Было решено, что наилучший — Вариант 4.

Вывод.

Были изучены основы работы с искусственными нейронными сетями. Проведены наблюдения за обучением нейронной сети при различных параметрах и выбрана наилучшая модель.

ПРИЛОЖЕНИЕ А
ИСХОДНЫЙ КОД

```
import pandas
from keras.layers import Dense
from keras.models import Sequential
from keras.utils import to_categorical
from sklearn.preprocessing import LabelEncoder
import matplotlib.pyplot as plot

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)

# Создание модели
model = Sequential()
model.add(Dense(4, activation="relu"))
model.add(Dense(4, activation="relu"))
model.add(Dense(3, activation="softmax"))

model.compile(optimizer='adam', loss='categorical_crossentropy',
metrics=['accuracy'])
H = model.fit(X, dummy_y, epochs=450, batch_size=10,
validation_split=0.1)

loss = H.history['loss']
val_loss = H.history['val_loss']
acc = H.history['accuracy']
val_acc = H.history['val_accuracy']
```

```
epochs = range(1, len(loss) + 1)

# Построение графика ошибки
plot.figure(1)
plot.plot(epochs, loss, 'bo', label='Training loss')
plot.plot(epochs, val_loss, 'b', label='Validation loss')
plot.title('Training and validation loss')
plot.xlabel('Epochs')
plot.ylabel('Loss')
plot.legend()

# Построение графика точности
plot.figure(2)
plot.clf()
plot.plot(epochs, acc, 'bo', label='Training acc')
plot.plot(epochs, val_acc, 'b', label='Validation acc')
plot.title('Training and validation accuracy')
plot.xlabel('Epochs')
plot.ylabel('Accuracy')
plot.legend()

plot.show()
```