

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

ОТЧЕТ
по лабораторной работе №1
по дисциплине «Искусственные нейронные сети»
Тема: Многоклассовая классификация цветов

Студент гр. 7382

Токарев А.П.

Преподаватель

Жукова Е.Е.

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.

Порядок выполнения работы.

1. Ознакомиться с задачей классификации;
2. Загрузить данные;
3. Создать модель ИНС в Keras;
4. Настроить параметры обучения;
5. Обучить и оценить модель.

Требования к выполнению задания.

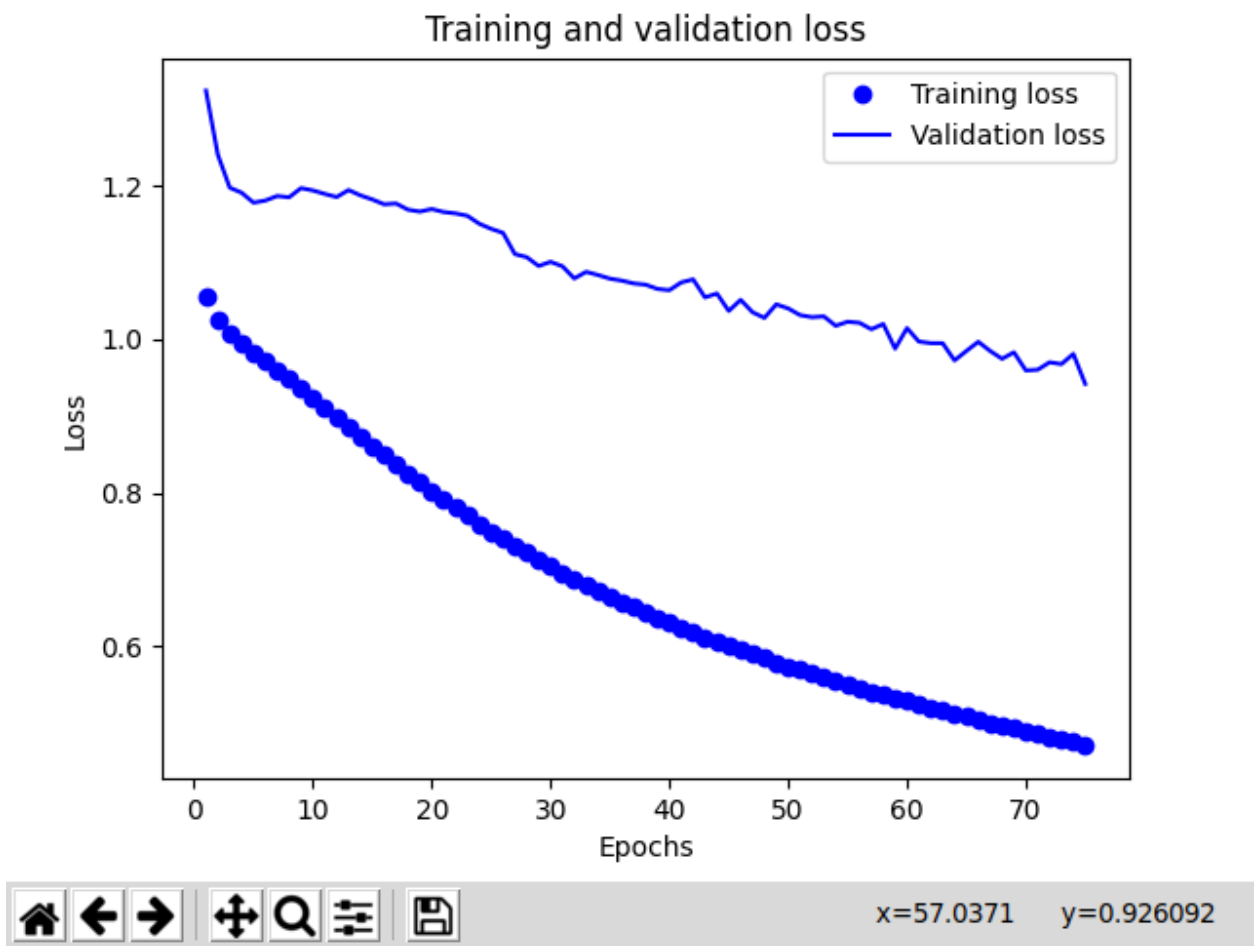
1. Изучить различные архитектуры ИНС (Разное количество слоев, разное количество нейронов на слоях);
2. Изучить обучение при различных параметрах обучения (параметры функций fit);
3. Построить графики ошибок и точности в ходе обучения;
4. Выбрать наилучшую модель.

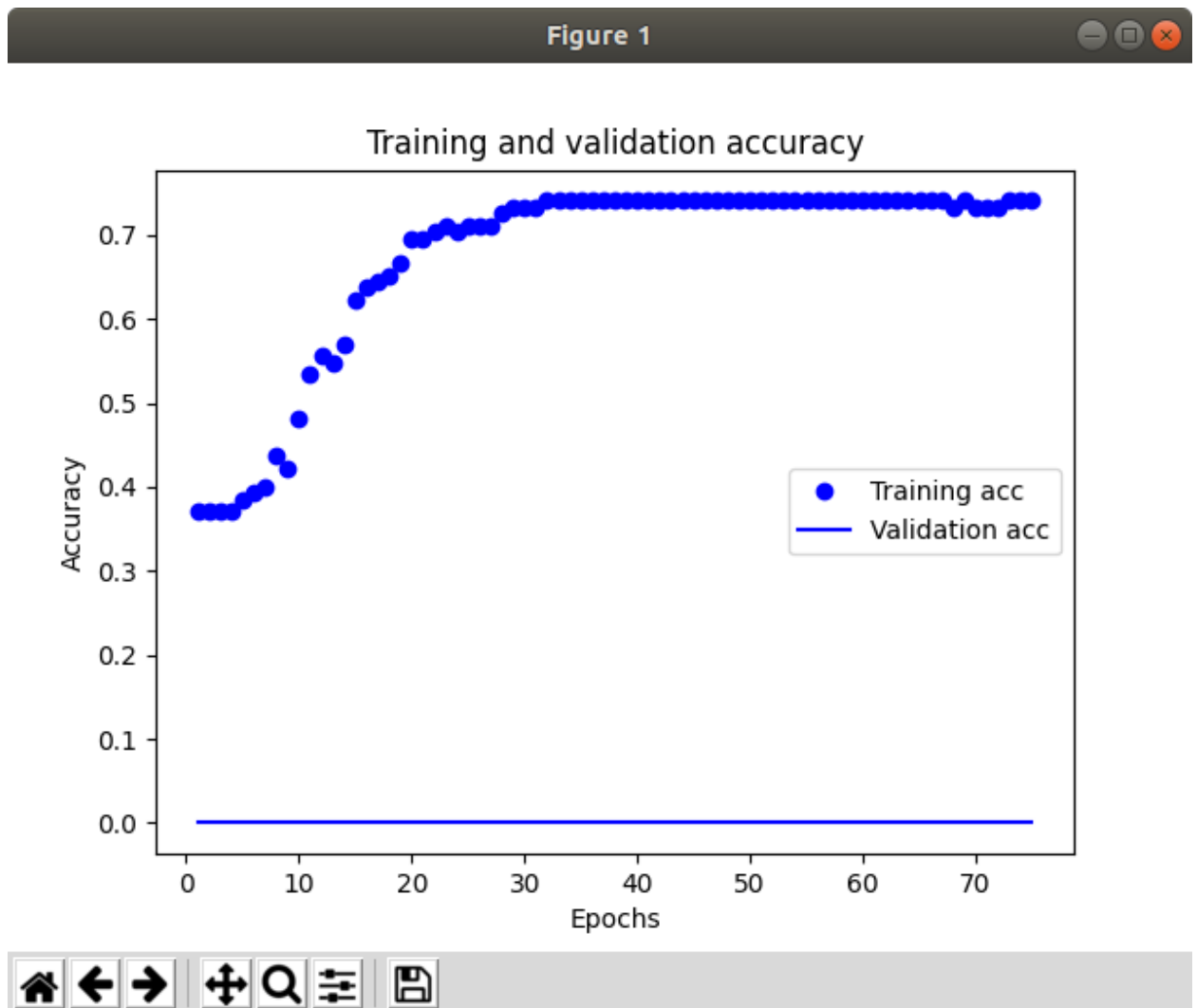
Ход работы.

Для изучения различной структуры ИНС была разработана и использована программа из приложения А.

При изначальных данных получили:

Figure 1

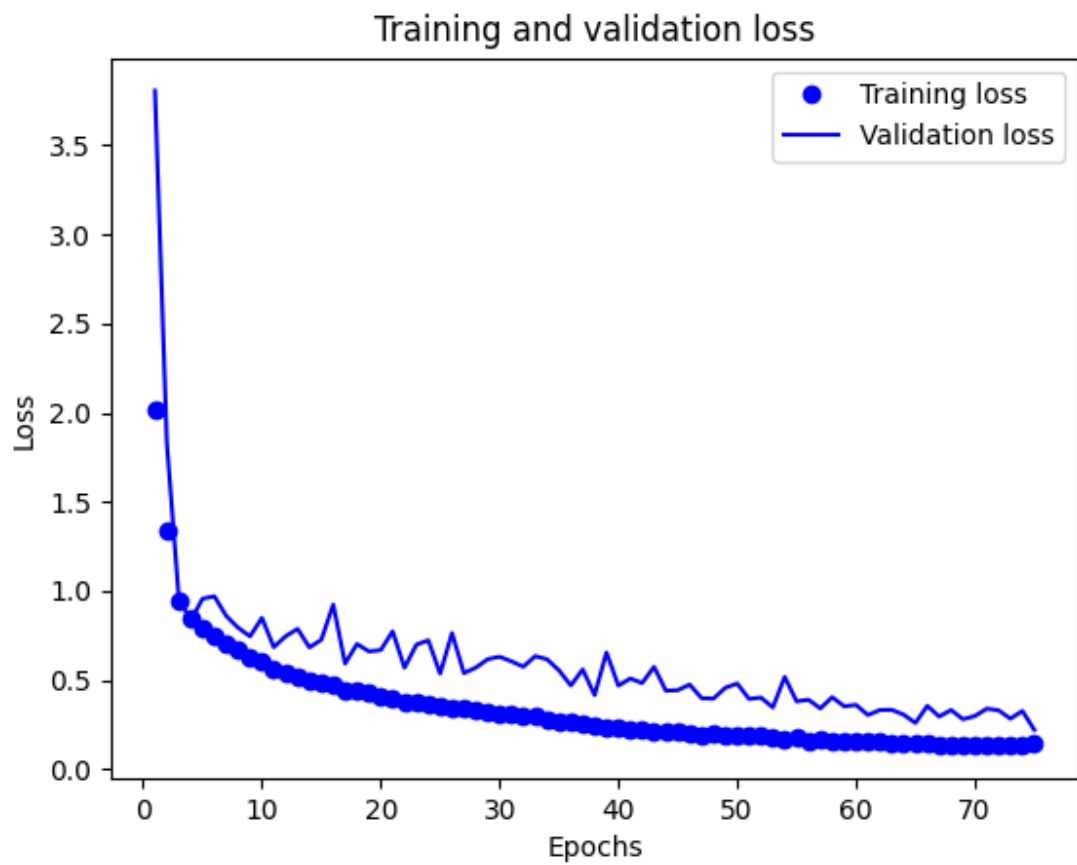


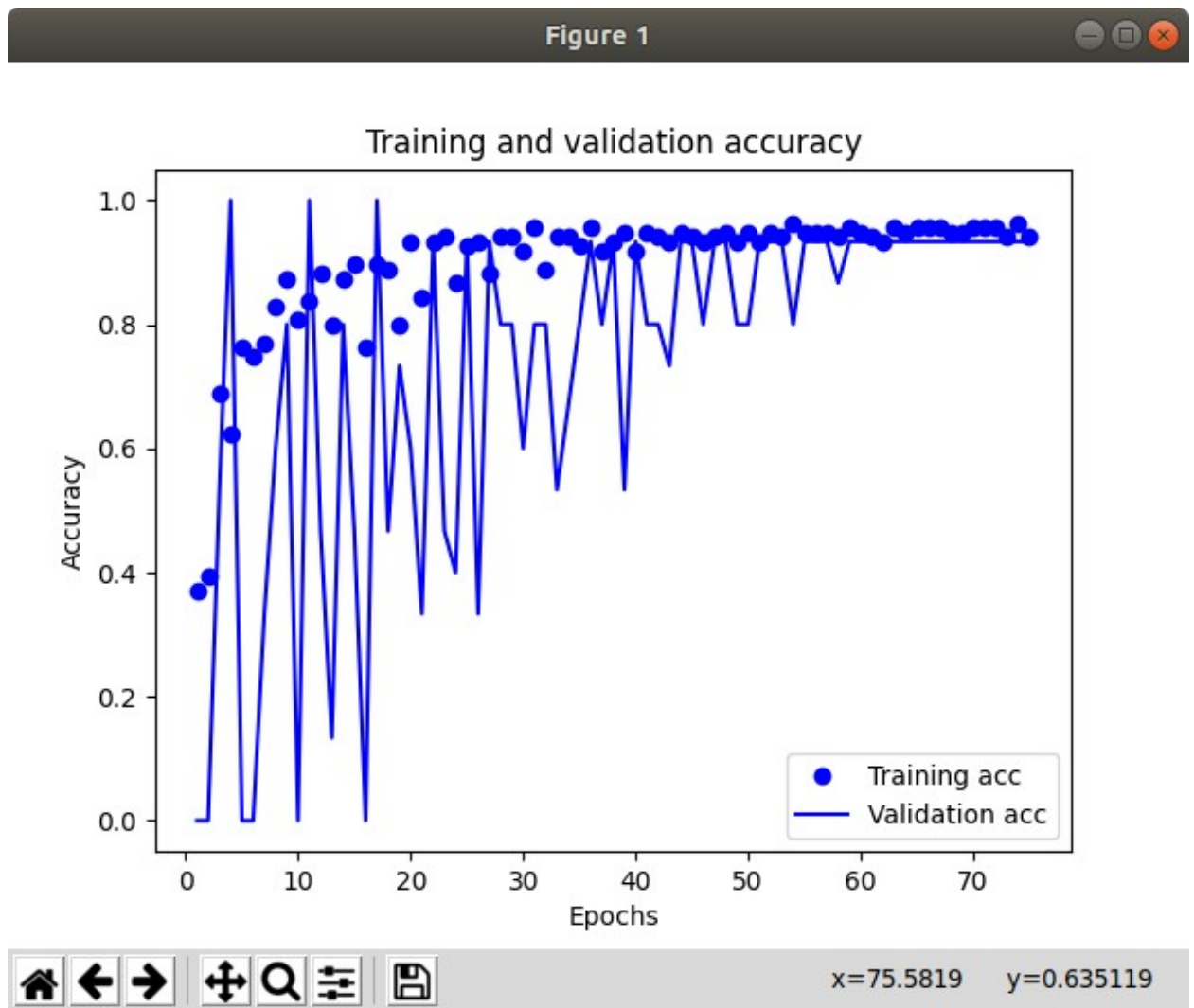


Было решено добавить ещё два слоя по 16 нейронов, исходя из мысли, что признаков всего 4, соответственно, на первом слое определяем некоторую комбинацию двух признаков, на втором всех 4

Наблюдаем:

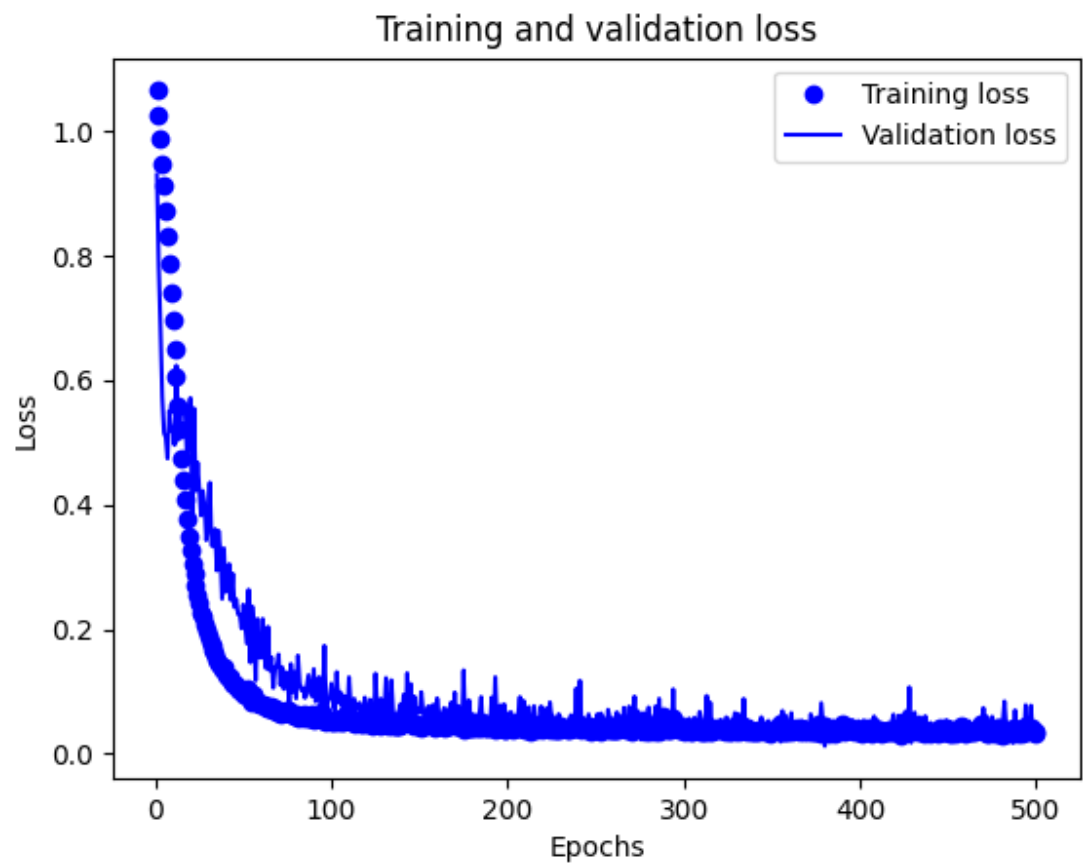
Figure 1

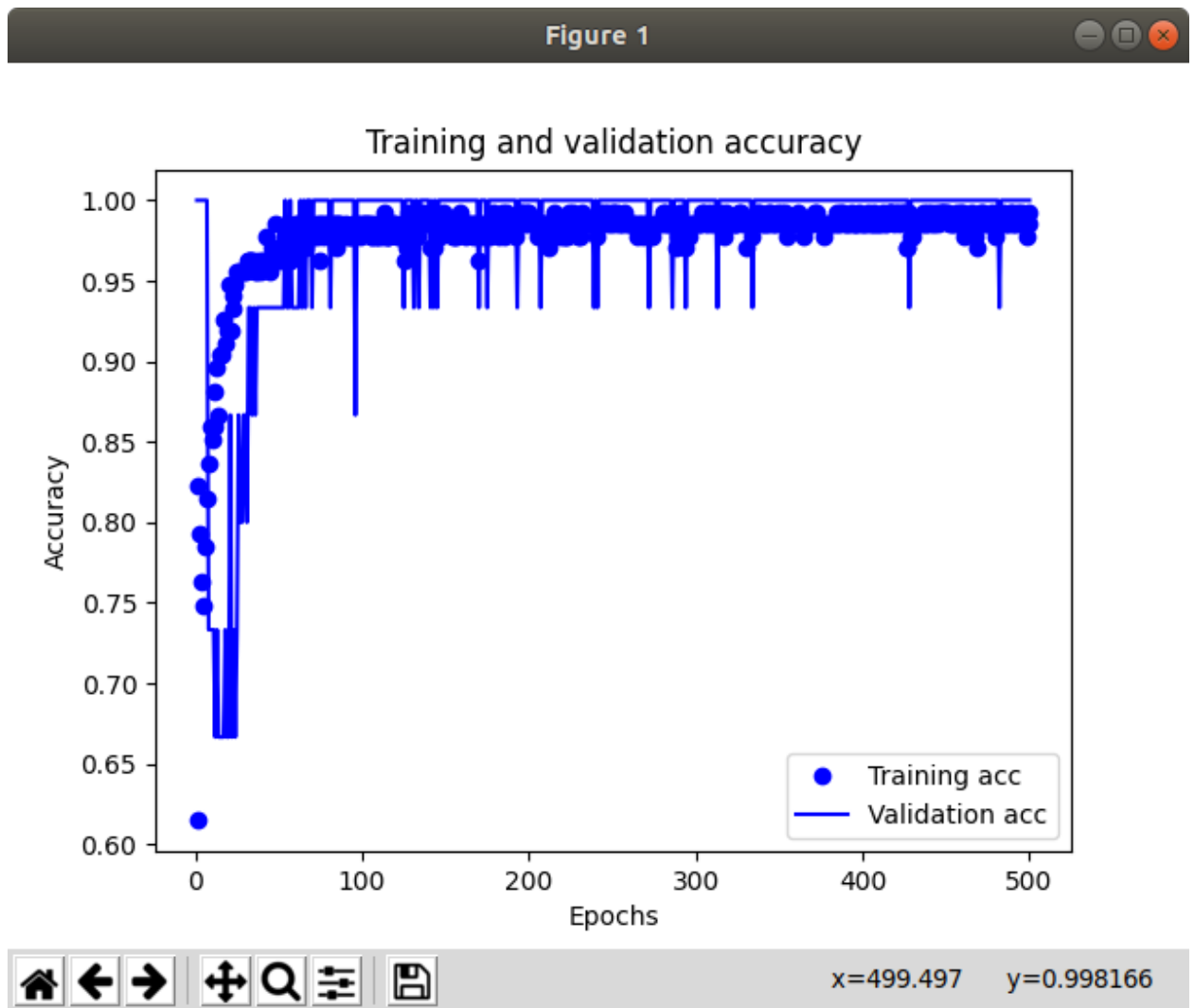




Уже неплохо, попробовал поднять количество эпох до 500:

Figure 1

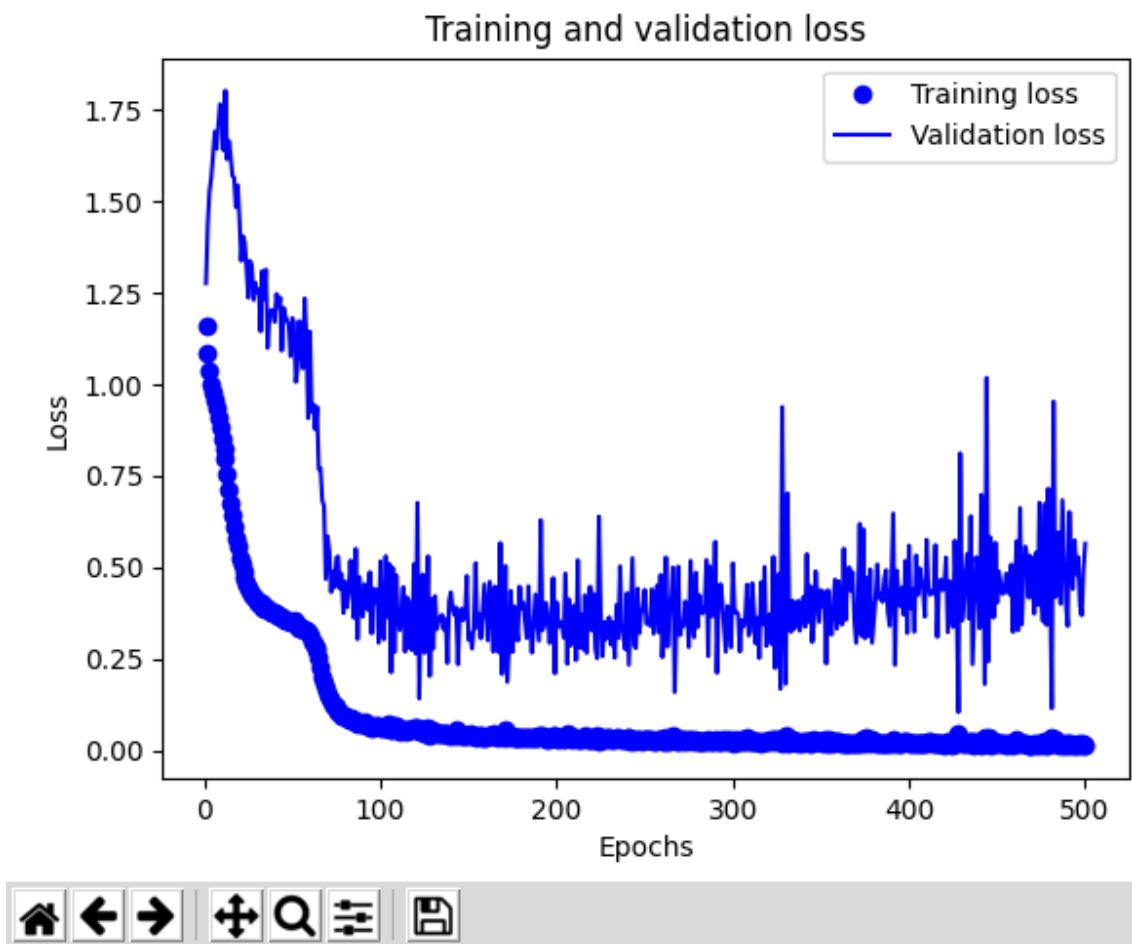


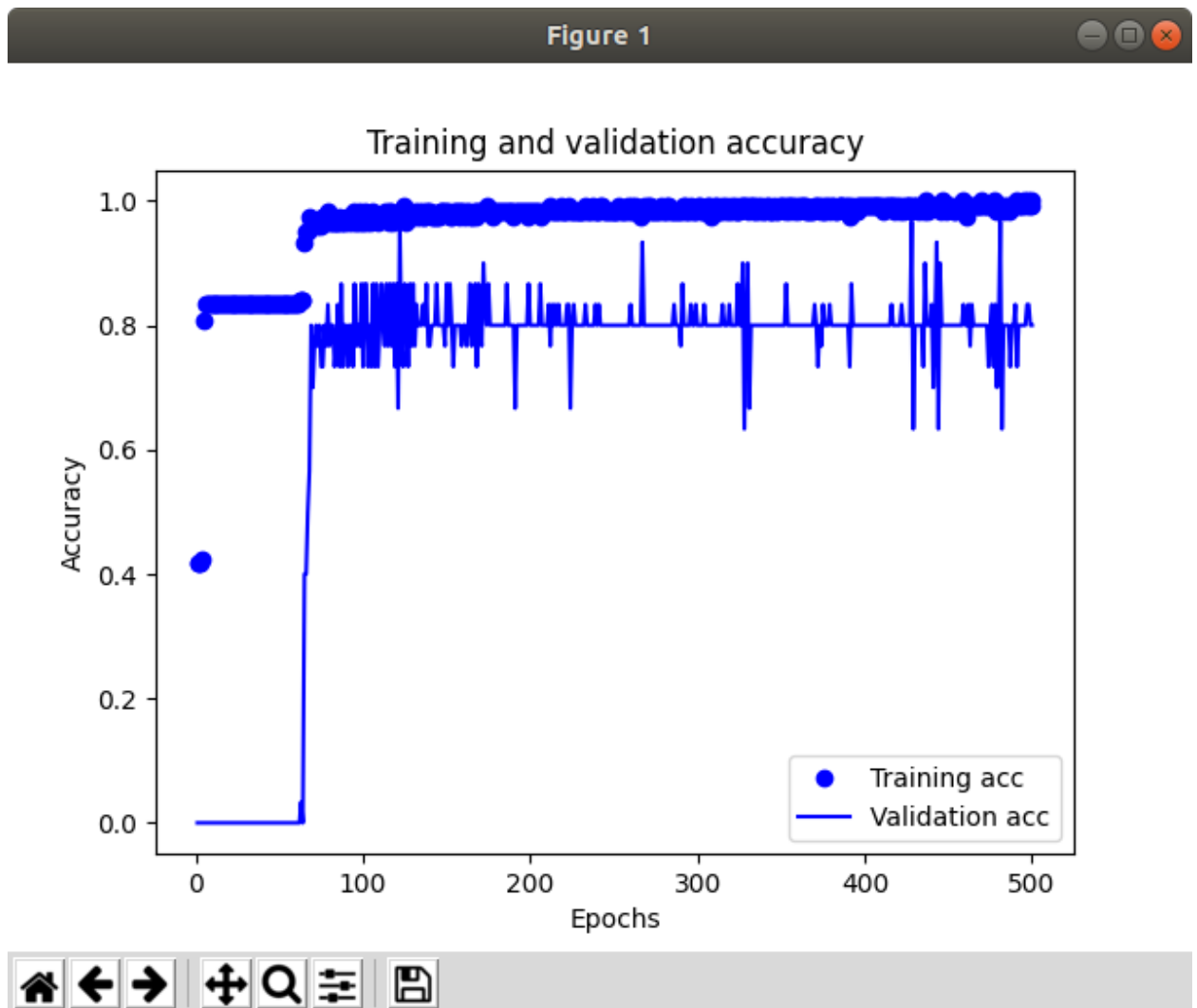


Было решено, что данный вариант — наилучший

Но, ради интереса, было испробовано увеличить размер контрольной выборки до 0.2:

Figure 1





На графике ошибки наблюдаем переобучение

Выводы.

Были изучены основы работы с искусственными нейронными сетями. Наблюдали за тем, как меняется результат в ходе изменений различных параметров, как в ходе построения модели, так и в ходе ее обучения, наблюдали переобучение.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

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

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)

model = Sequential()
model.add(Dense(4, activation='relu'))
model.add(Dense(16, activation='relu'))
model.add(Dense(16, activation='relu'))
model.add(Dense(3, activation='softmax'))

model.compile(optimizer='adam', loss='categorical_crossentropy',
metrics=['accuracy'])

H = model.fit(X, dummy_y, epochs=500, batch_size=10,
validation_split=0.1)

# Получение ошибки и точности в процессе обучения
print(H.history.keys())
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

plt.plot(epochs, loss, 'bo', label='Training loss')
plt.plot(epochs, val_loss, 'b', label='Validation loss')
plt.title('Training and validation loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()
plt.show()
```

```
# Построение графика точности
```

```
plt.clf()
plt.plot(epochs, acc, 'bo', label='Training acc')
plt.plot(epochs, val_acc, 'b', label='Validation acc')
plt.title('Training and validation accuracy')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.legend()
plt.show()
```