

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. 8382

Кобенко В.П.

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.

Требования к выполнению работы.

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

Задачи.

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

Ход работы.

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

Модель 1:

Модель была взята из методички.

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

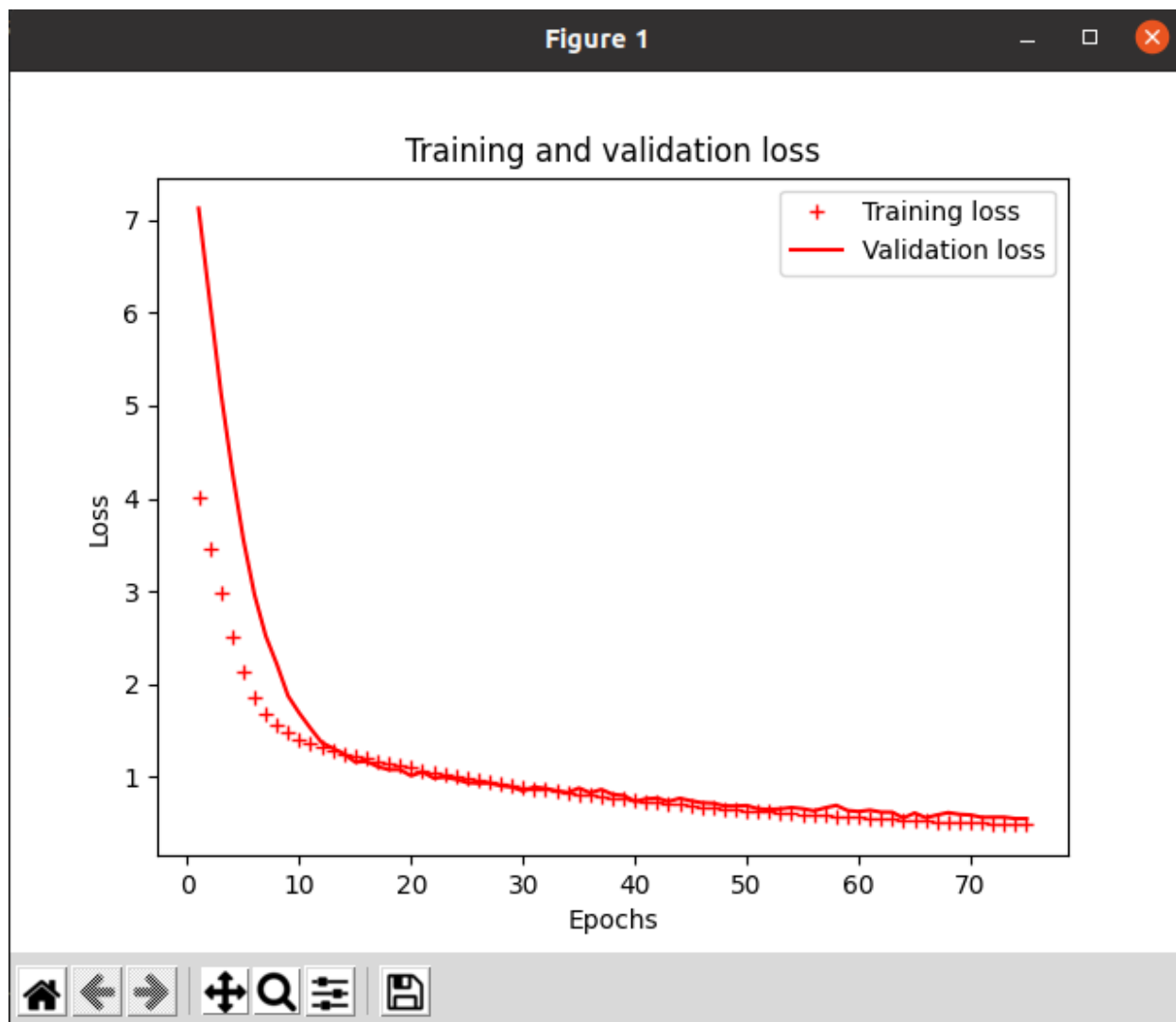


Рисунок 1 – График потерь модели 1

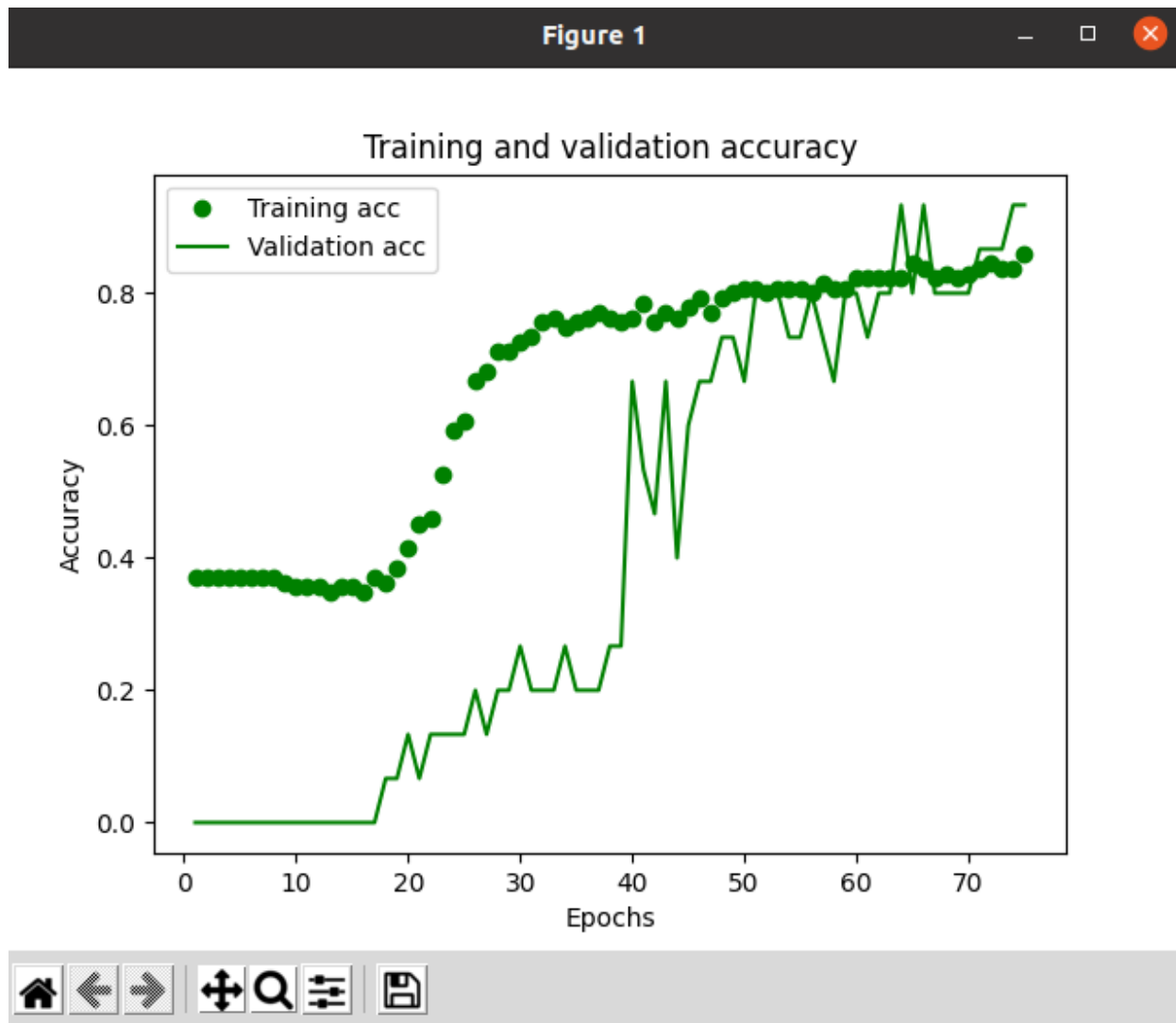


Рисунок 2 – График точности модели 1

Модель 2:

Модель отличается тем, что был добавлен новый слой.

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

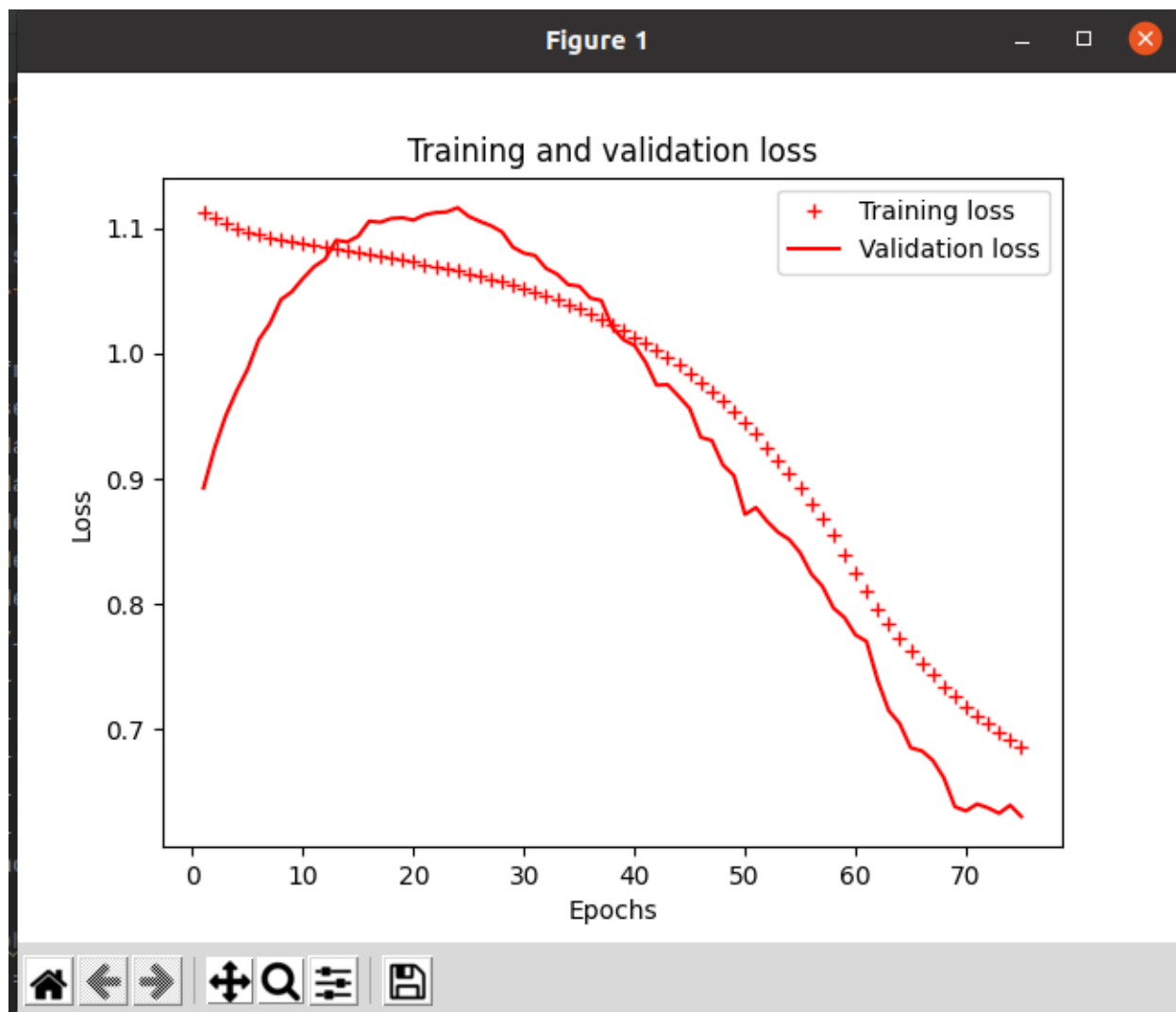


Рисунок 3 – График потерь модели 2

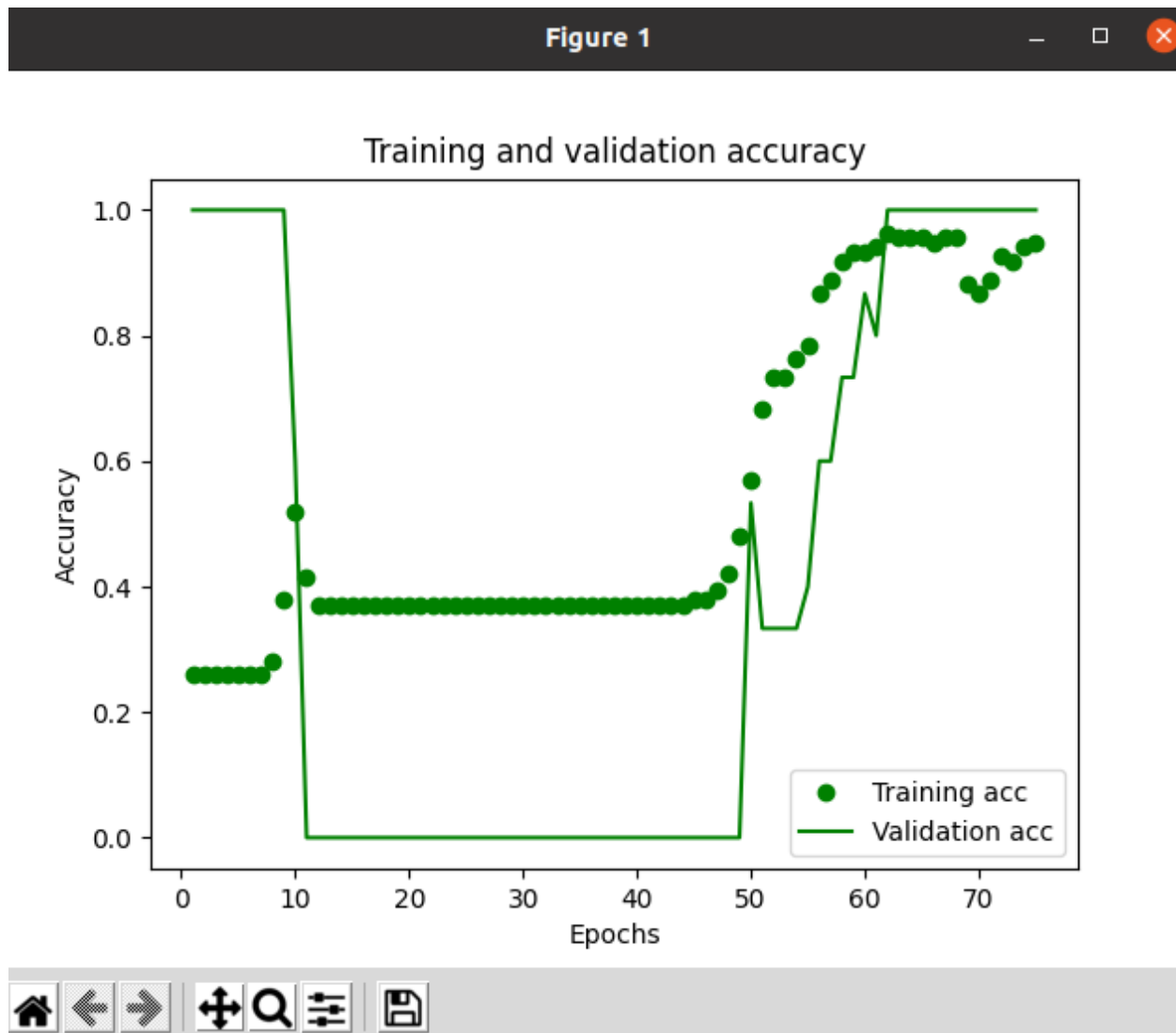


Рисунок 4 – График точности модели 2

Модель 3:

Модель отличается тем, что было поменяно количество нейронов.

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

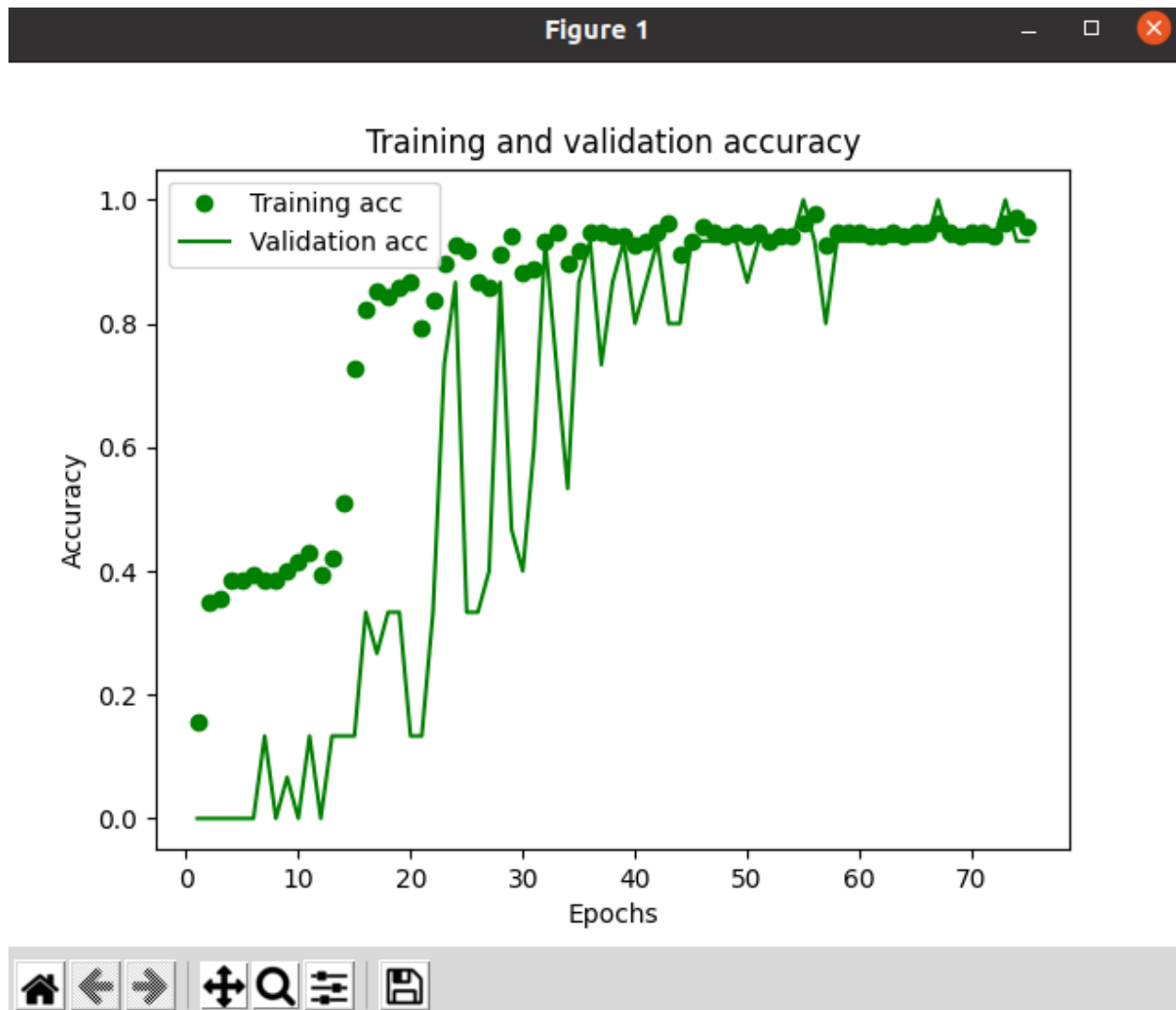


Рисунок 5 – График точности модели 3

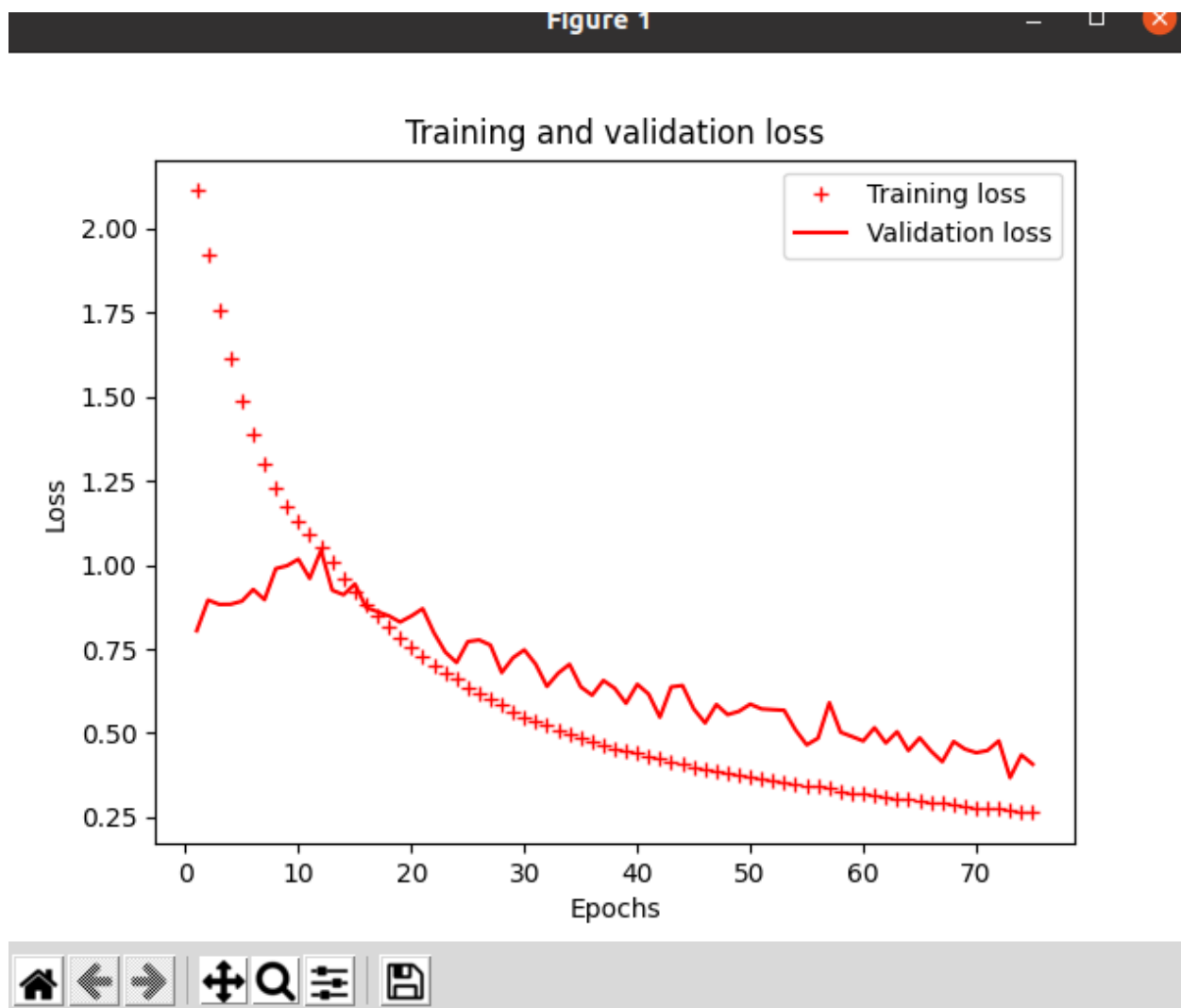


Рисунок 6 – График потерь модели 3

Модель 4:

Модель отличается тем, что было увеличено количество эпох.

```
h = model.fit(X, dummy_y, epochs=150, batch_size=10, validation_split=0.1)
```

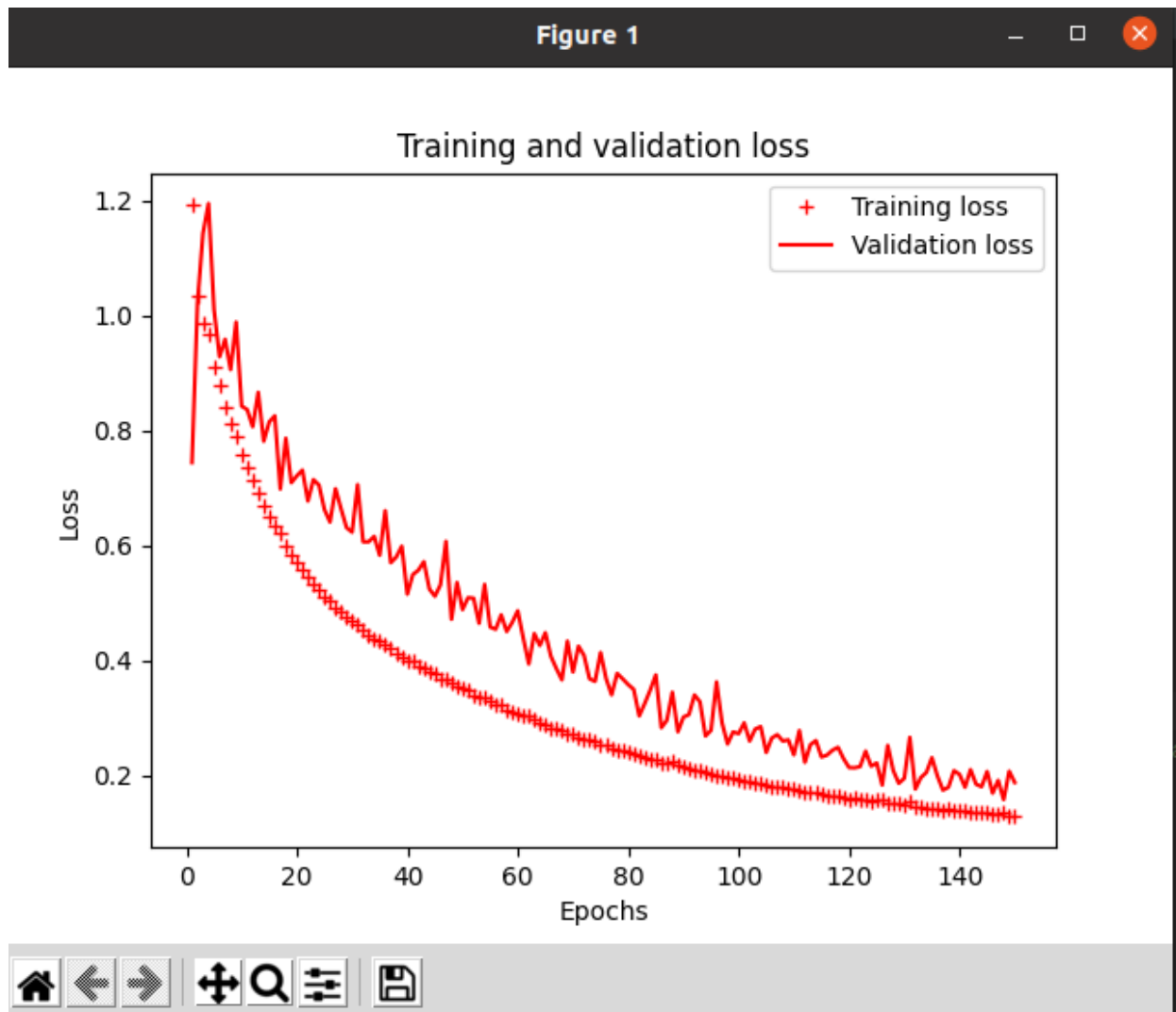



Рисунок 7 – График потерь модели 4

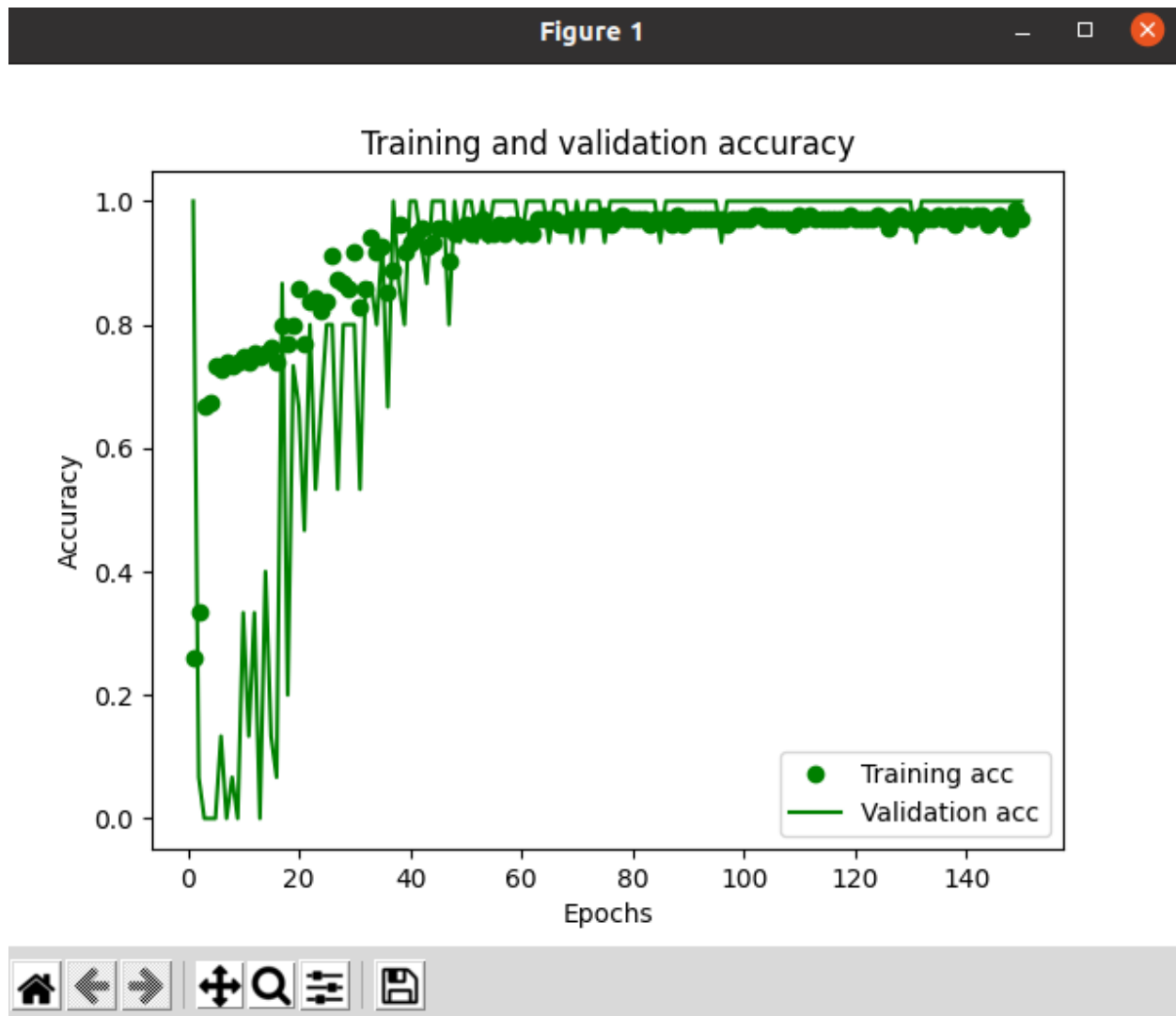


Рисунок 8 – График точности модели 4

Модель 5:

Модель отличается тем, что было уменьшено количество батчей.

```
h = model.fit(X, dummy_y, epochs=75, batch_size=3, validation_split=0.1)
```

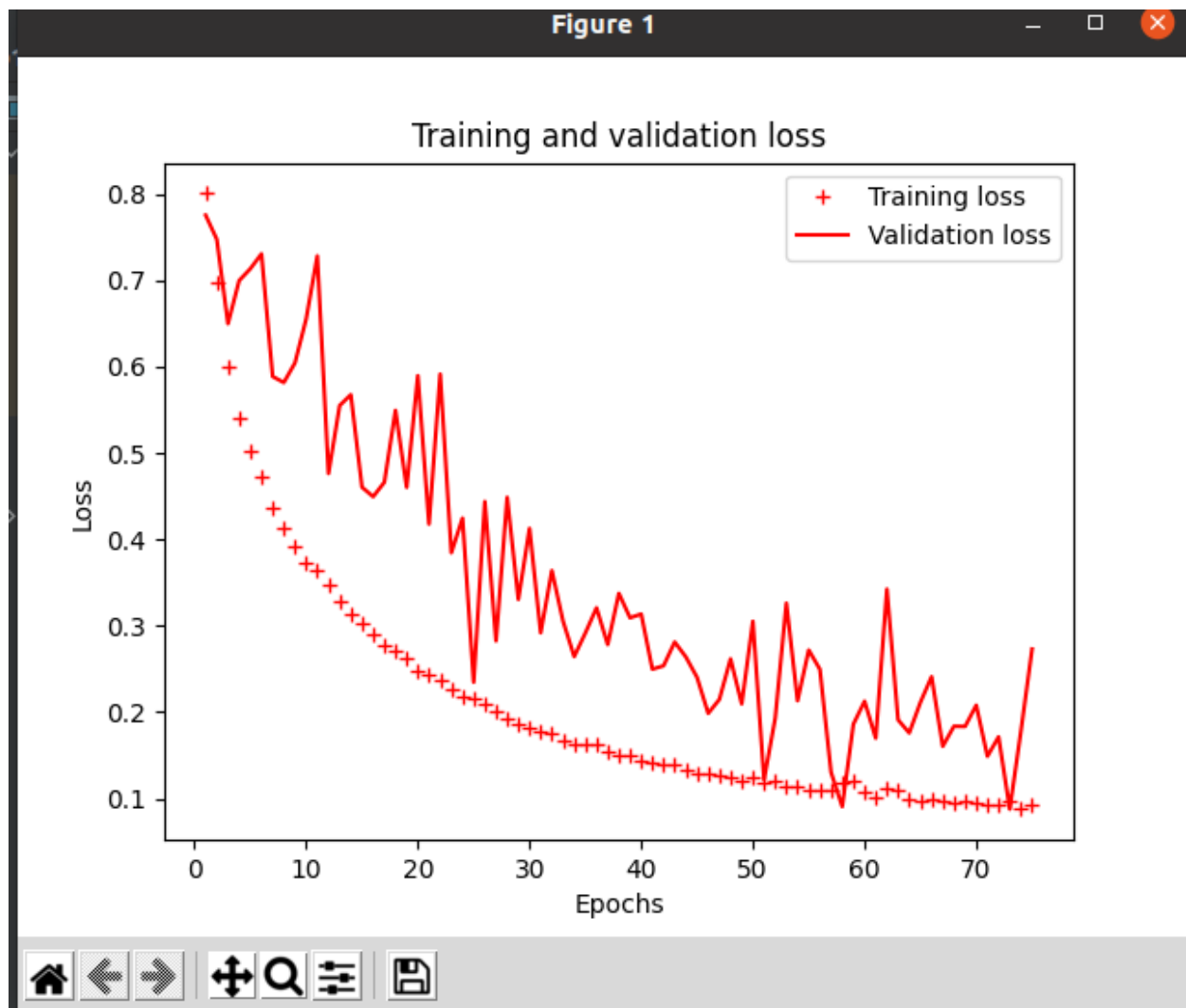


Рисунок 9 – График потерь модели 5

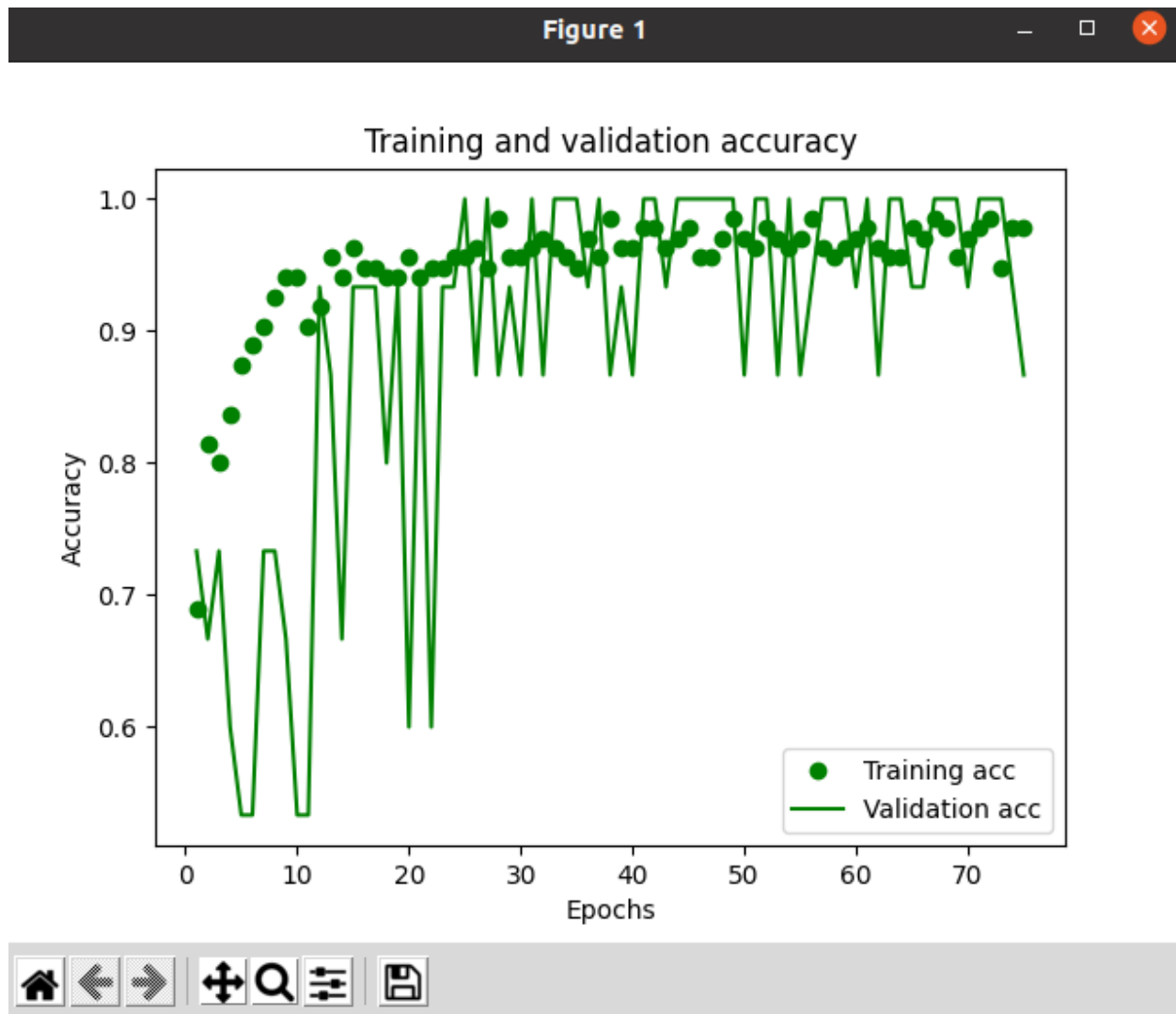


Рисунок 10 – График точности модели 5

Модель 6:

Модель отличается тем, что было увеличена доля данных обучения, используемая в качестве данных проверки.

```
h = model.fit(X, dummy_y, epochs=75, batch_size=10, validation_split=0.5)
```

Figure 1

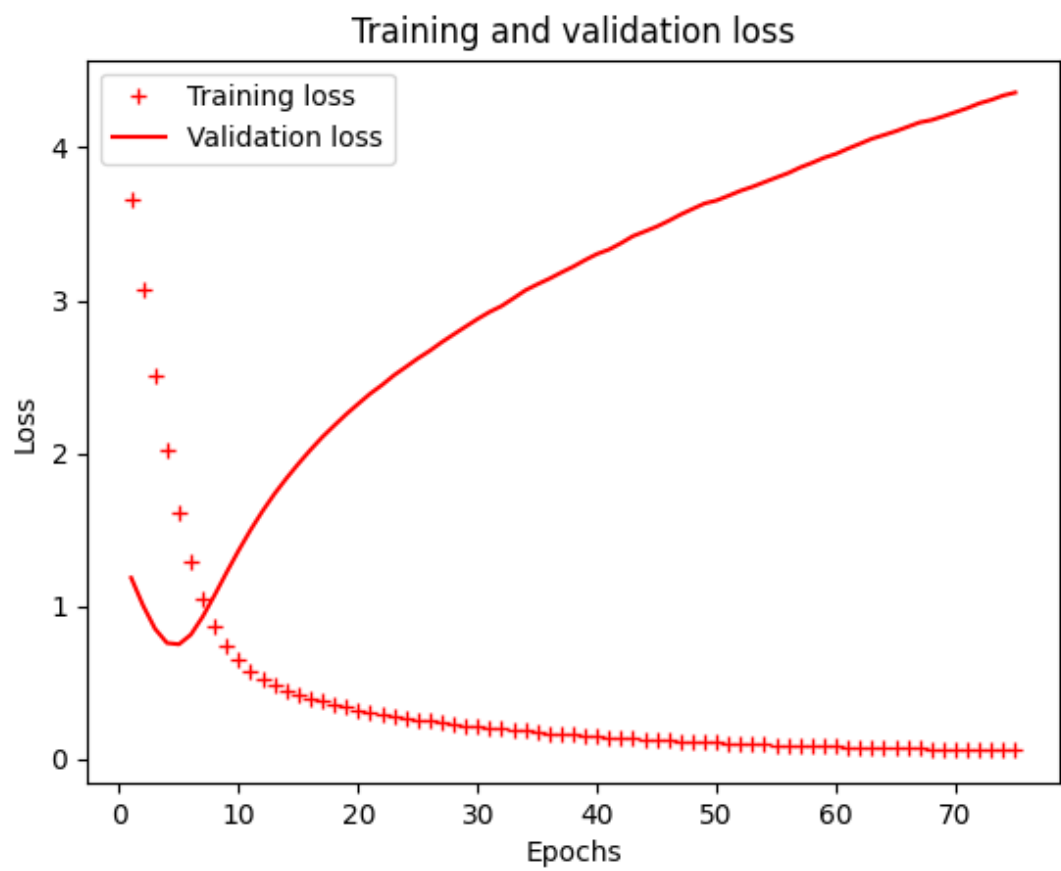


Рисунок 11 – График потерь модели 6

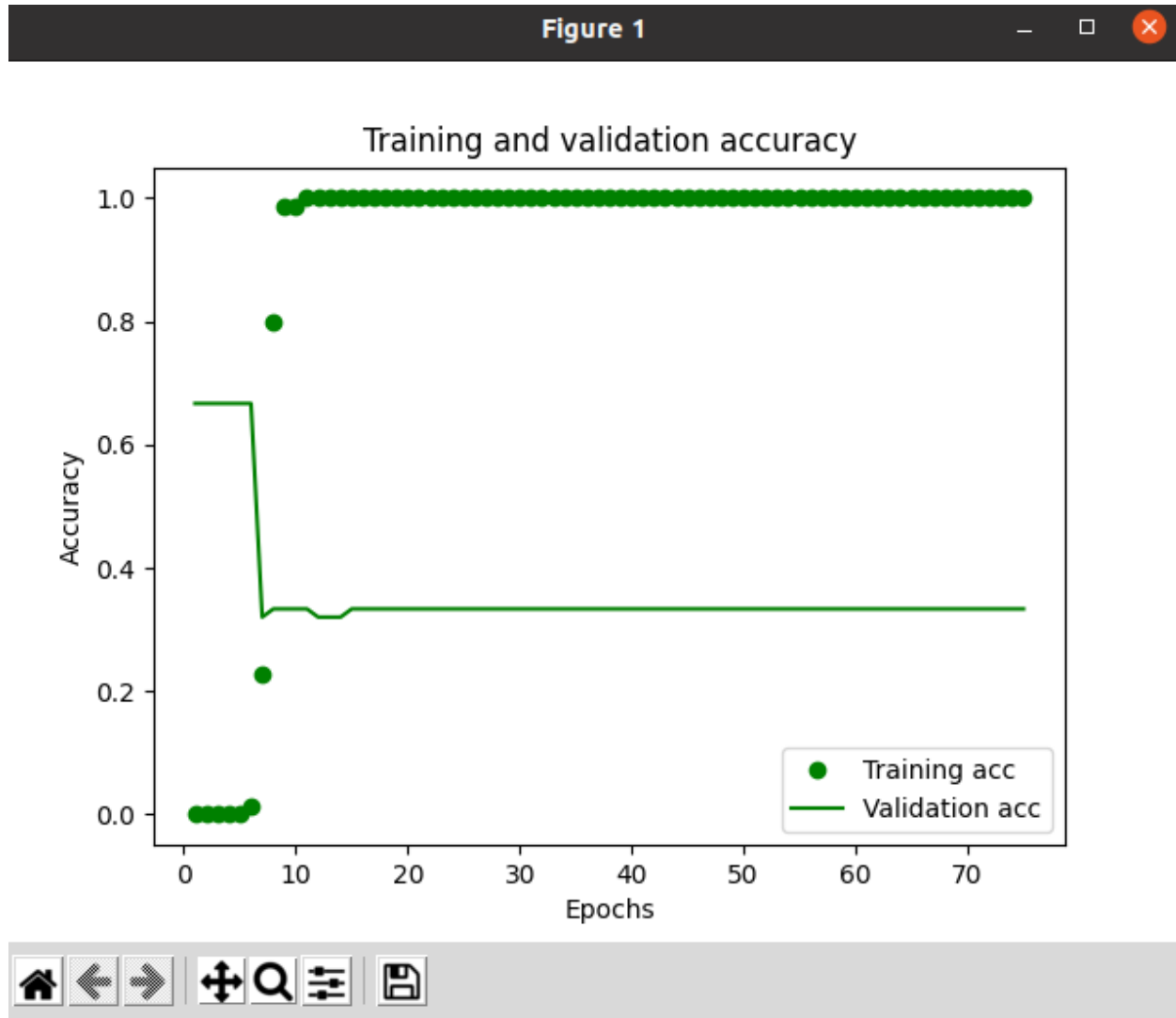


Рисунок 12 – График точности модели 6

Выводы.

При выполнении лабораторной работы ознакомились с задачей классификации, создали модель ИНС и настроили оптимальные параметры обучения, при которых ИНС выдавала приемлемые результаты.

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(3, activation='softmax'))
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
h = model.fit(X, dummy_y, epochs=75, batch_size=10, validation_split=0.1)

#graphics
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)
#1
plt.plot(epochs, loss, 'r+', label='Training loss')
plt.plot(epochs, val_loss, 'r', label='Validation loss')
plt.title("Training and validation loss")
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()
plt.show()
plt.clf()
#2
plt.plot(epochs, acc, 'go', label='Training acc')
plt.plot(epochs, val_acc, 'g', label='Validation acc')
plt.title("Training and validation accuracy")
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.legend()
plt.show()
```