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»

Студентка гр. 8382 _____ Звегинцева Е.Н.

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.

Задачи.

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

Требования.

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;

Ход работы.

1) Была создана и обучена модель искусственной нейронной сети в соответствии с условиями (весь код представлен в приложении А).

2) Графики ошибок и точности строились с помощью библиотеки matplotlib.

Код графика ошибок:

```
loss_values = history_dict['loss']
val_loss_values = history_dict['val_loss']
epochs = range(1, len(loss_values) +1)
plt.plot(epochs, loss_values, 'g', label='Training loss')
plt.plot(epochs, val_loss_values, 'y', label='Validation loss')
plt.title('Training and validation loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()
plt.show()
```

Ниже представлен пример графика ошибок на рис.1.

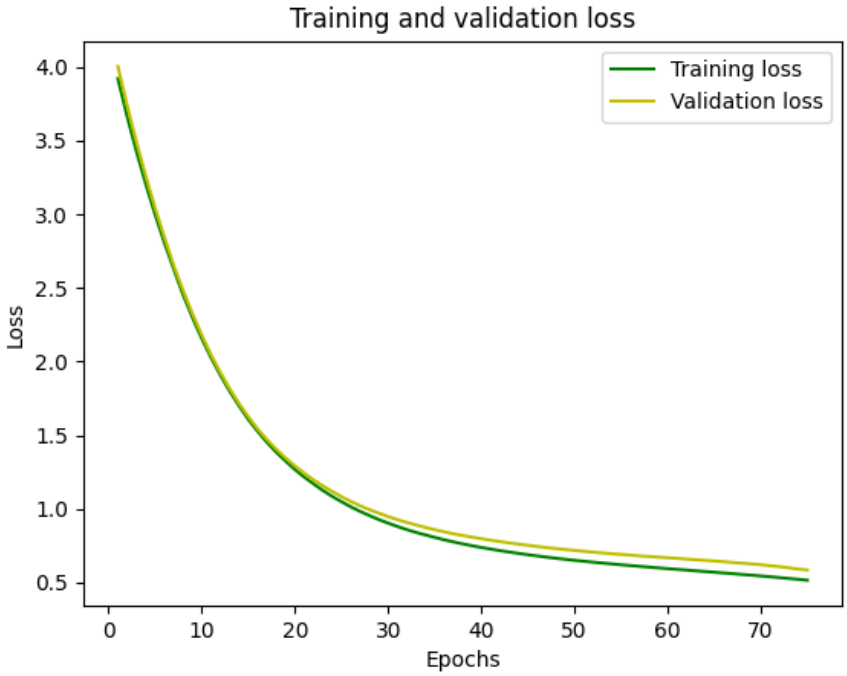


Рисунок 1 – Пример графика ошибок при обучении модели

Код графика точности:

```
plt.clf()
acc_values = history_dict['accuracy']
val_acc_values = history_dict['val_accuracy']
plt.plot(epochs, acc_values, 'r', label='Training accurate')
```

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

Ниже представлен пример графика точности на рис.2.

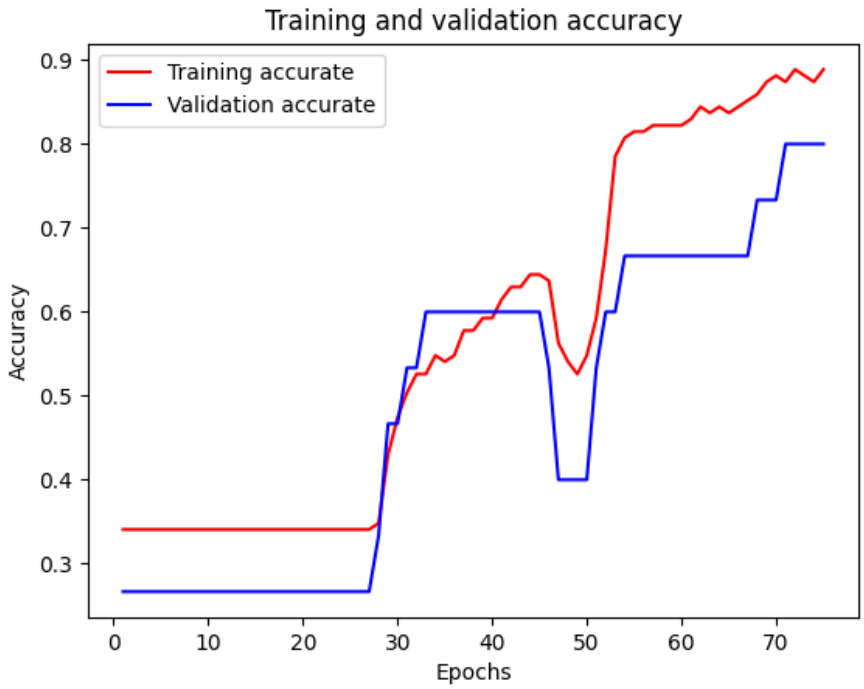
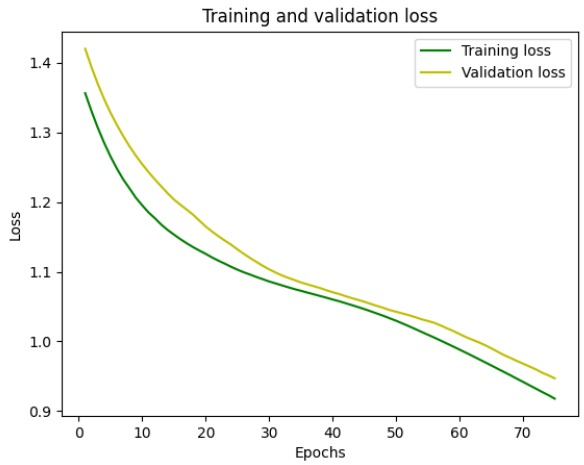
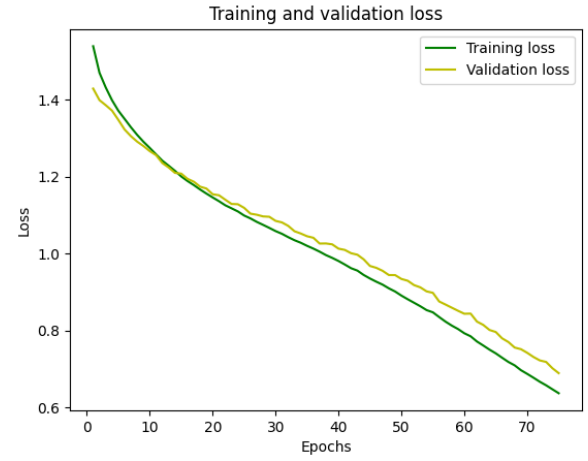
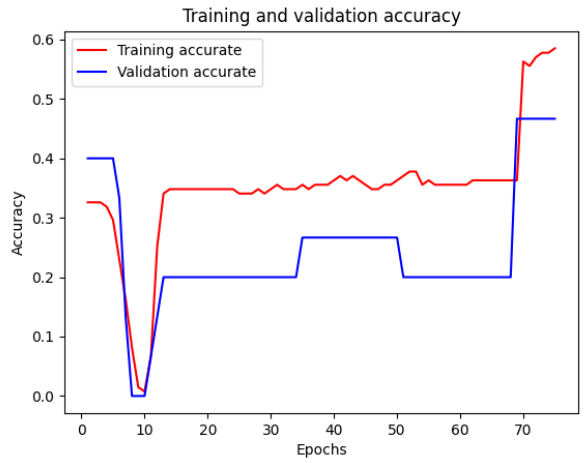
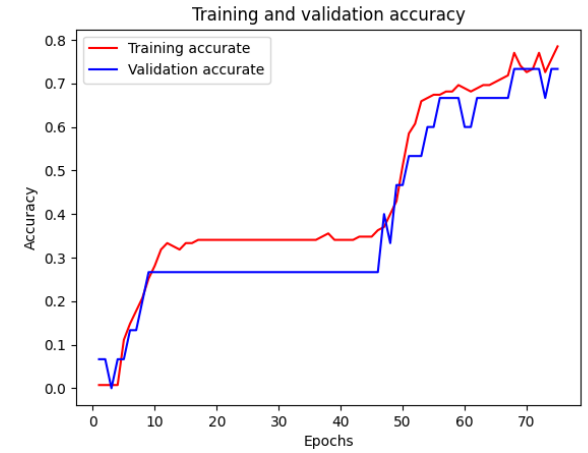


Рисунок 2 – Пример графика точности при обучении модели

3) При исследовании разных архитектур и обучение при различных параметрах обучения ИНС менялись следующие параметры:

- Batch_size

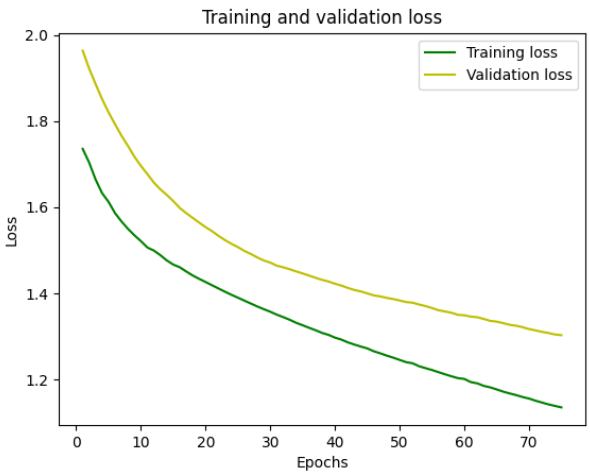
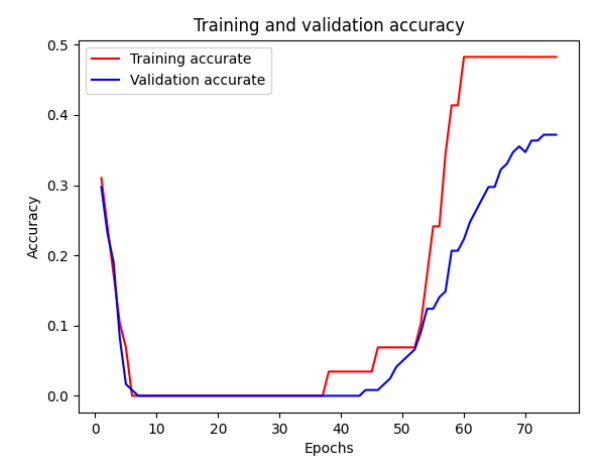
Это параметр, отвечающий за обновление весов нейросети. В нашем случае достаточно 10, так как малое количество данных и с большим шагом нейросеть будет работать быстрее, но ошибок будет больше.

batch_size = 30	batch_size = 10
	
	

- Validation_split

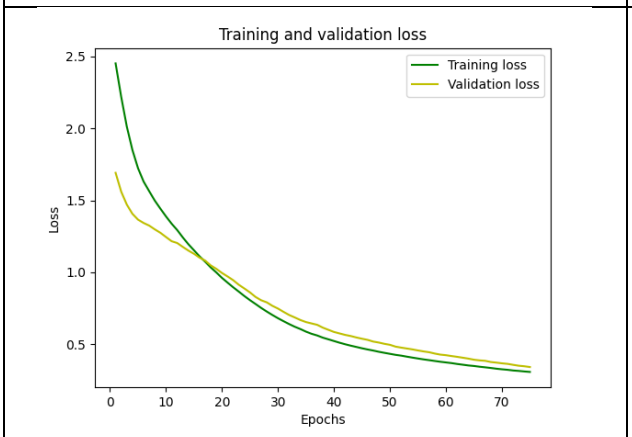
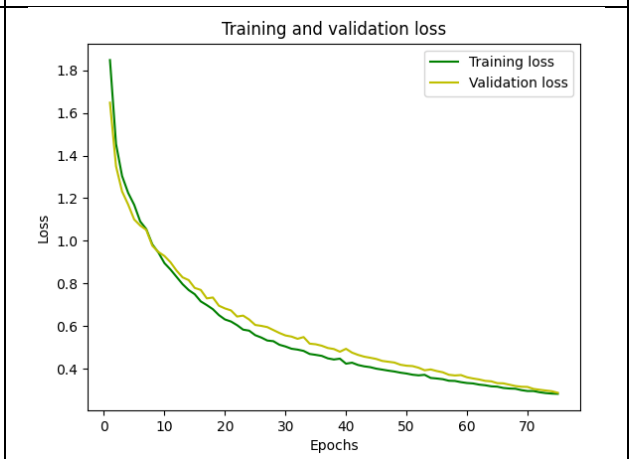
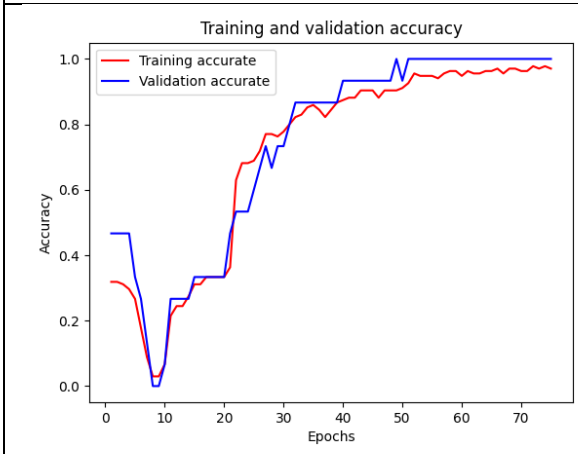
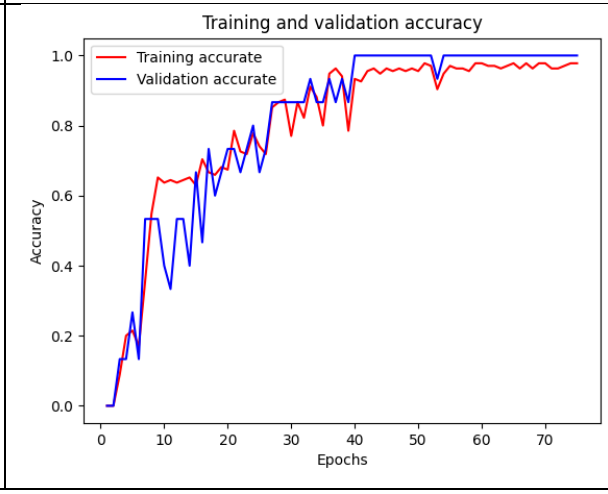
Параметр определяет процент данных, которые будут отведены под валидацию.

Приведена сеть при параметре 0,8.

Графики не сходятся, также у нас мало тренировочных данных, следовательно нужно взять меньший процент. Возьмем 0,1.

- Количество нейронов

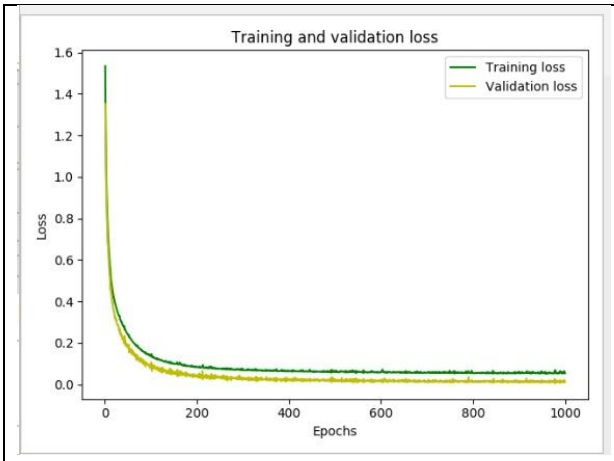
8	32
	
	

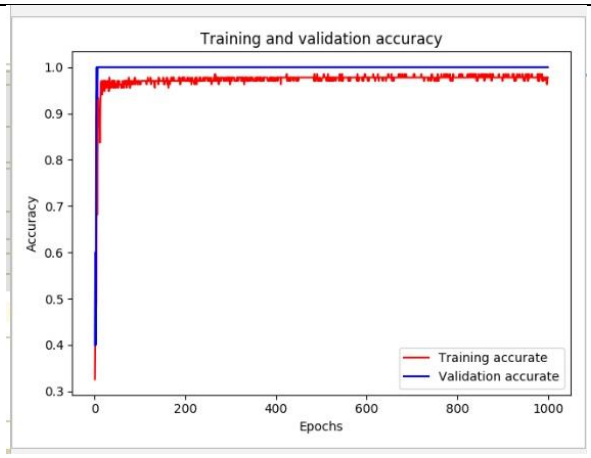
Как мы видим, с увеличением количества нейронов в слое повышается точность(количество нейронов увеличивалось до момента когда перестали происходить видимые изменения).

- Количество эпох обучения

Графики не расходятся даже при большом количестве эпох и переобучение не происходит. Большое количество эпох на данной сети бесполезно, в среднем хватит около 75.

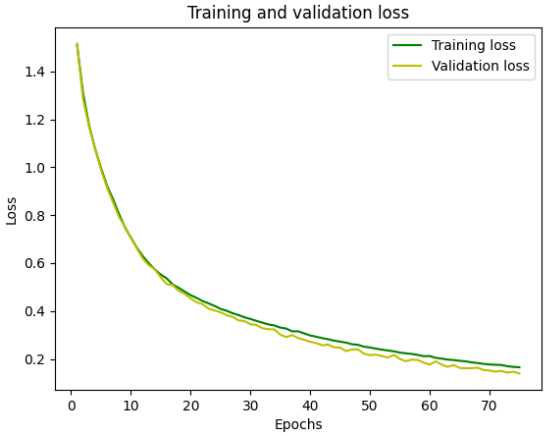
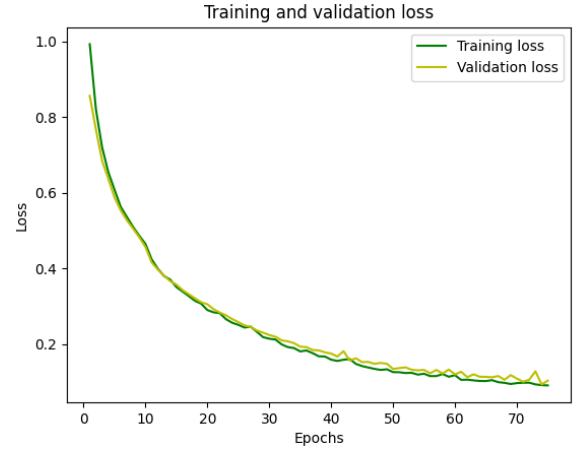
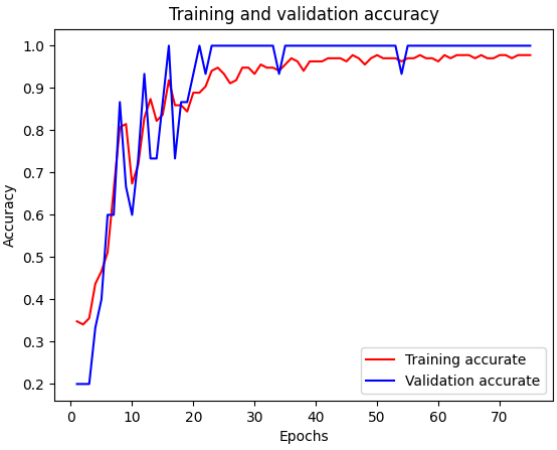
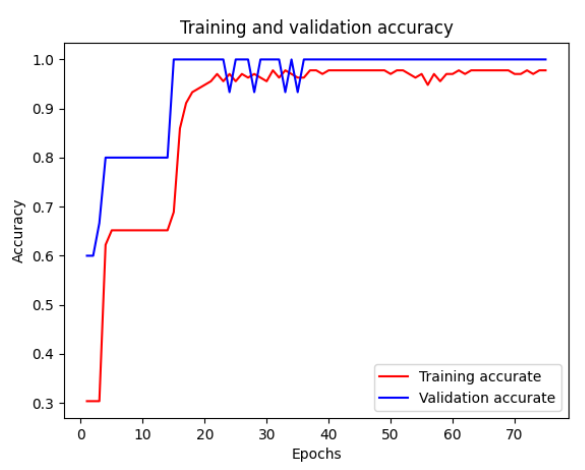
Сеть при `epochs` =1000





- Количество слоев

Возьмем 2 слоя при 32 нейронах, а 3 слоя 32 и 8 нейронов.

2	3
	
	

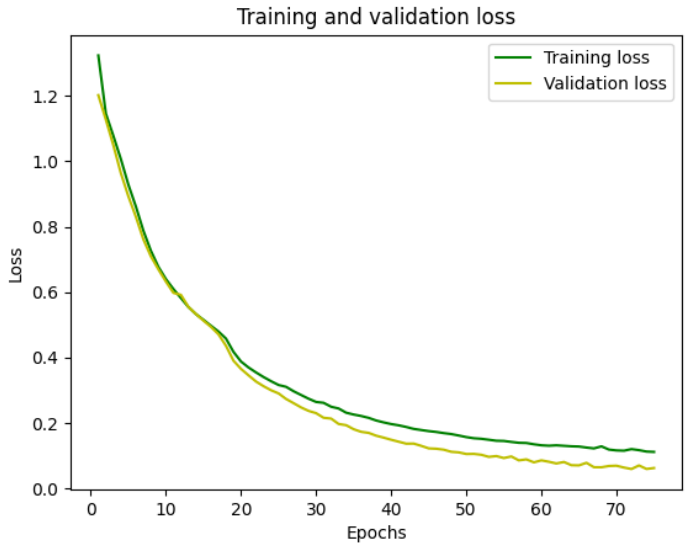
Видимые улучшения не заметны, следовательно нужно менять количество нейронов в слоях.

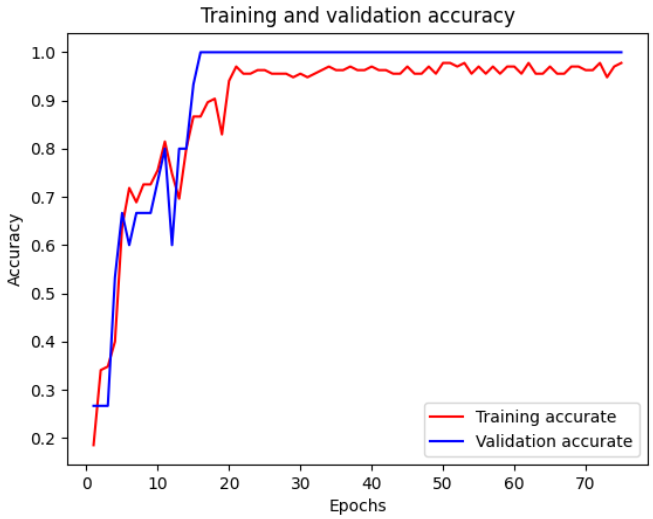
4) Путем исследований, была определена оптимальная нейронная сеть

```
model = Sequential()
model.add(Dense(16, activation='relu', input_shape=(4,)))
```

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

history = model.fit(X, dummy_y, epochs=75, batch_size=10, validation_split=0.1)
```





Выводы.

В ходе выполнения лабораторной работы были получены базовые теоретические и практические навыки по работе с ИНС и библиотекой keras.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД

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

#Загрузка данных
dataframe = pandas.read_csv("iris.csv", header=None)
dataset = dataframe.values

np.random.shuffle(dataset)

X = dataset[:,0:4].astype(float)
Y = dataset[:,4]

#Переход от текстовых меток к категориальному вектору
encoder = LabelEncoder()
encoder.fit(Y)
encoded_Y = encoder.transform(Y)
dummy_y = to_categorical(encoded_Y)

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

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

#Обучение сети
history = model.fit(X, dummy_y, epochs=75, batch_size=10,
validation_split=0.1)

history_dict = history.history
#print(history_dict.keys())

#График ошибки
loss_values = history_dict['loss']
```

```
val_loss_values = history_dict['val_loss']
epochs = range(1, len(loss_values) +1)
plt.plot(epochs, loss_values, 'g', label='Training loss')
plt.plot(epochs, val_loss_values, 'y', label='Validation loss')
plt.title('Training and validation loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()
plt.show()

#График точности
plt.clf()
acc_values = history_dict['accuracy']
val_acc_values = history_dict['val_accuracy']
plt.plot(epochs, acc_values, 'r', label='Training accurate')
plt.plot(epochs, val_acc_values, 'b', label='Validation accurate')
plt.title('Training and validation accuracy')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.legend()
plt.show()
```