

**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

\_\_\_\_\_

Александров 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.

## Постановка задачи.

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

## Выполнение работы.

Попробуем обучить нейронную сеть при следующих параметрах:

- 2 слоя, где в первом слое – 4 нейрона, во втором – 3 нейрона,
- 75 эпох,
- размер батча равный 10.

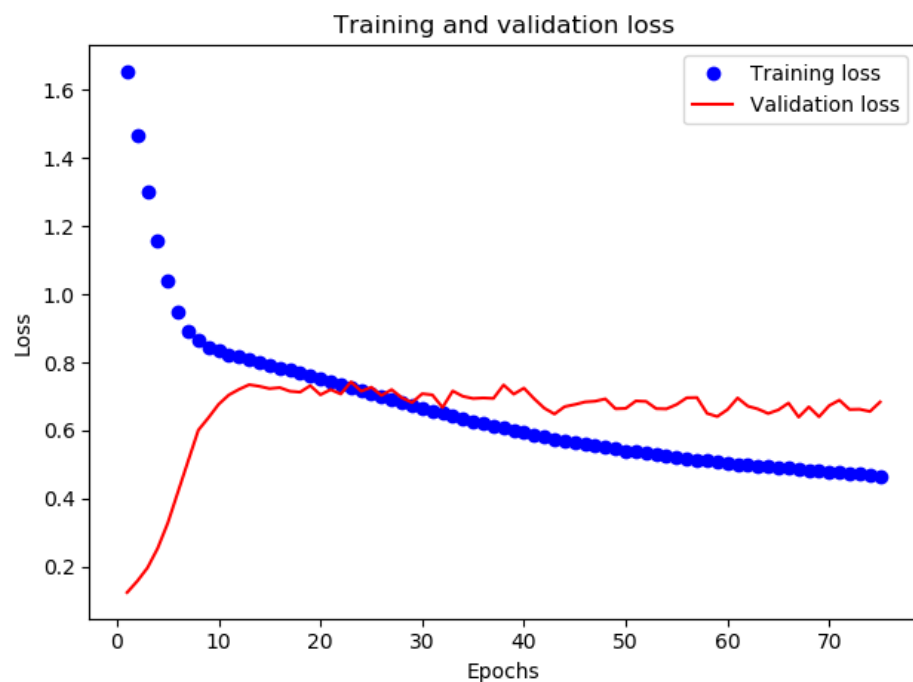


Рисунок 1 – Ошибки на 2 слоях с 75 эпохами

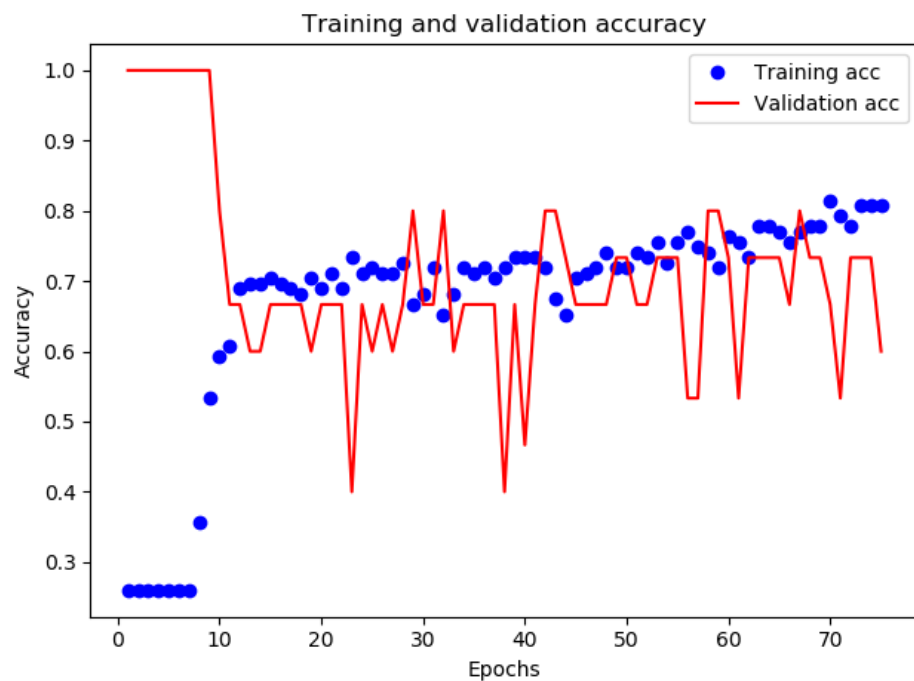


Рисунок 2 – Точность на 2 слоях с 75 эпохами

Увеличим число эпох до 450.

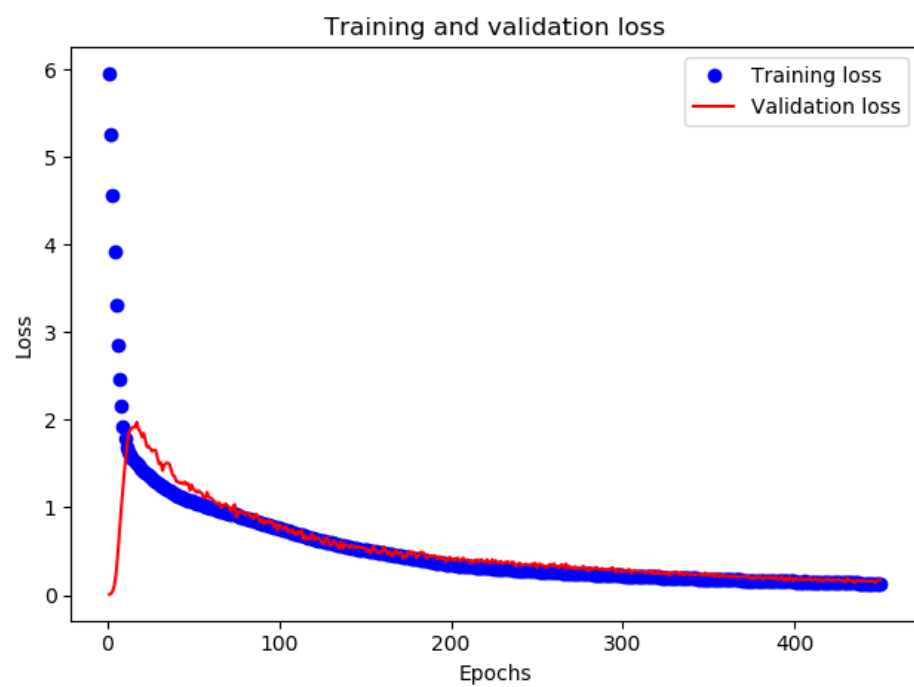


Рисунок 3 – Ошибки на 2 слоях с 450 эпохами

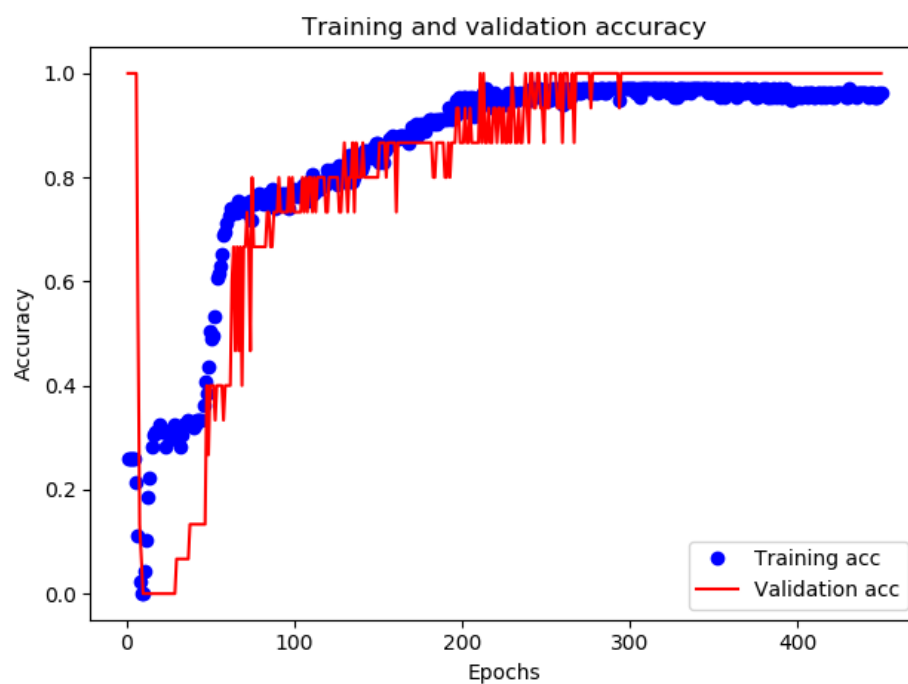


Рисунок 4 – Точность на 2 слоях с 450 эпохами

Увеличим число эпох до 900.

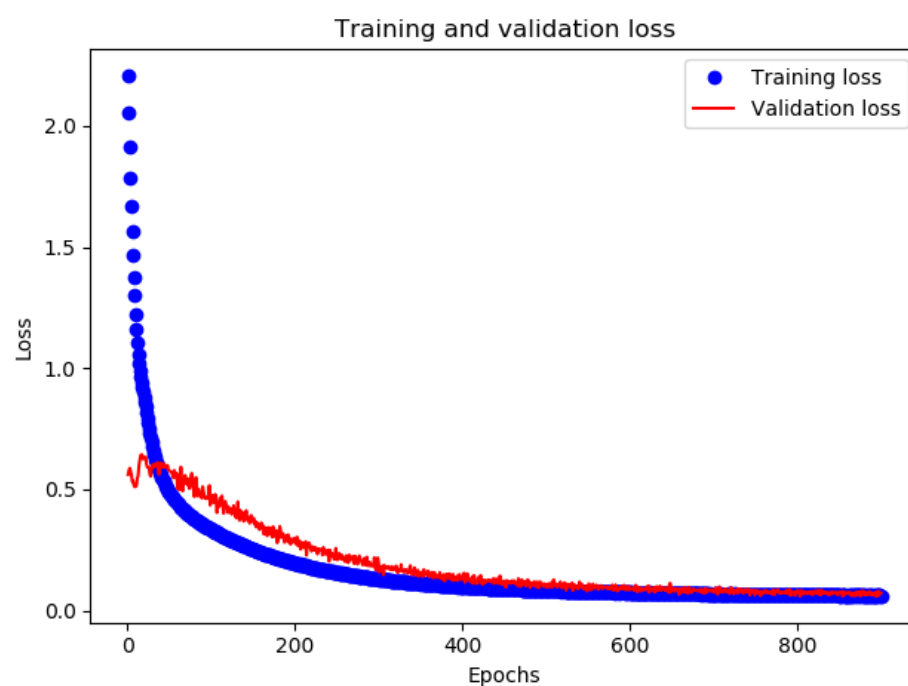


Рисунок 5 – Ошибки на 2 слоях с 900 эпохами

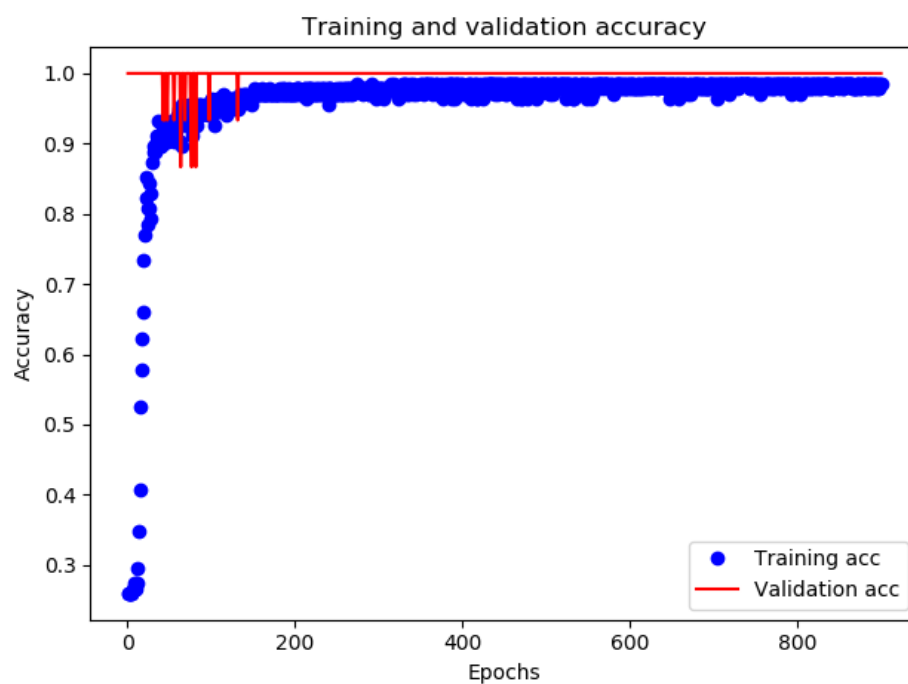


Рисунок 6 – Точность на 2 слоях с 900 эпохами

Добавим еще один слой на 4 нейрона.

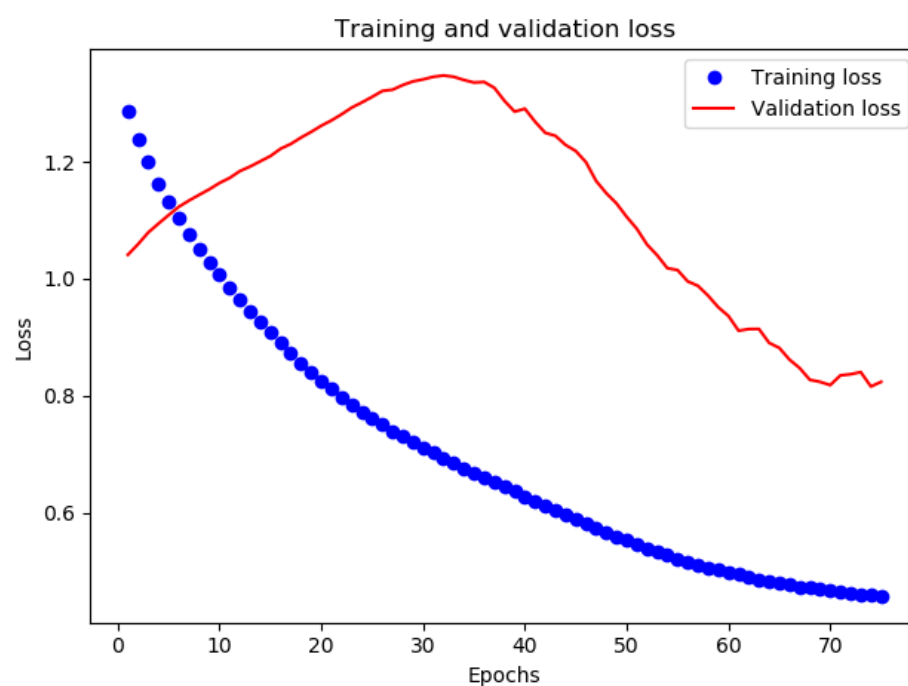


Рисунок 7 – Ошибки на 3 слоях с 75 эпохами

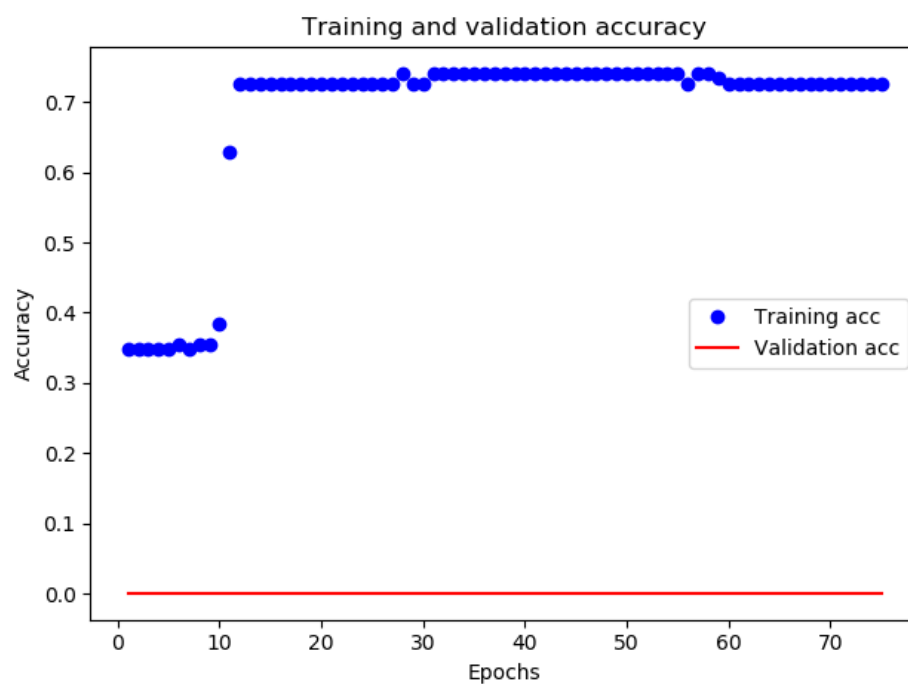


Рисунок 8 – Точность на 3 слоях с 75 эпохами

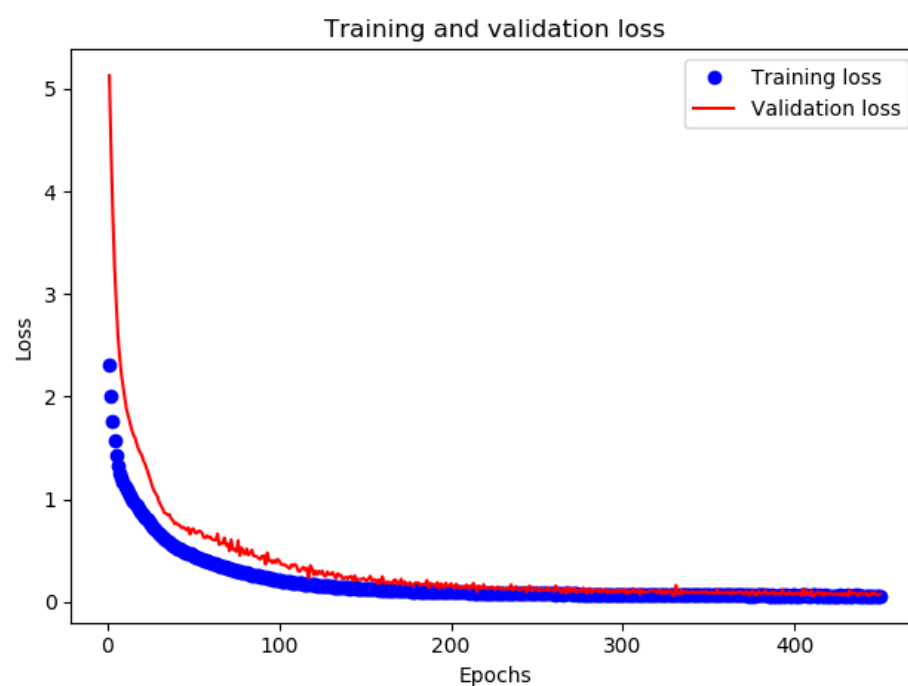


Рисунок 9 – Ошибки на 3 слоях с 450 эпохами

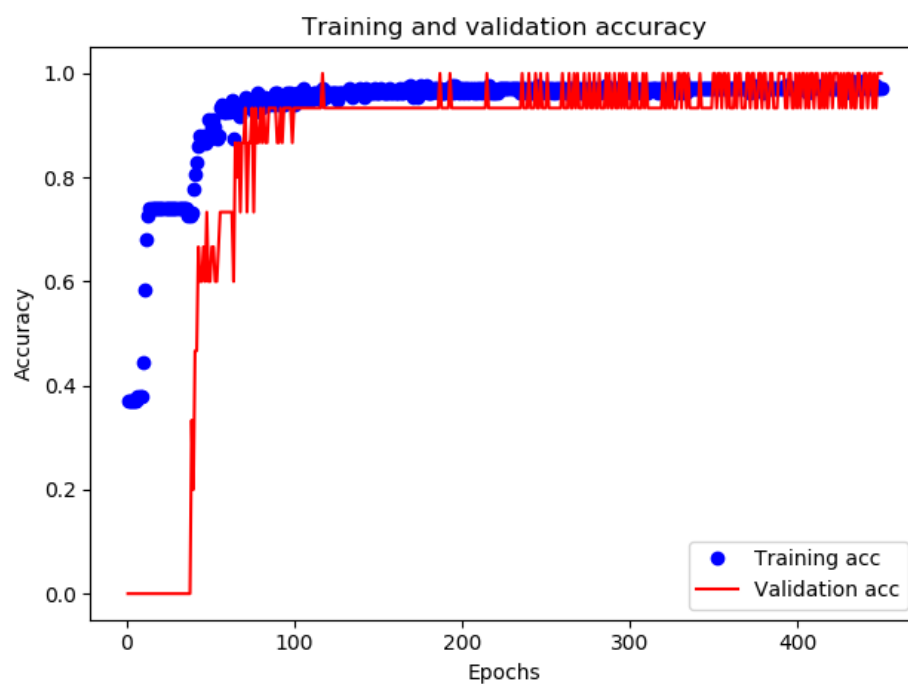


Рисунок 10 – Точность на 3 слоях с 450 эпохами

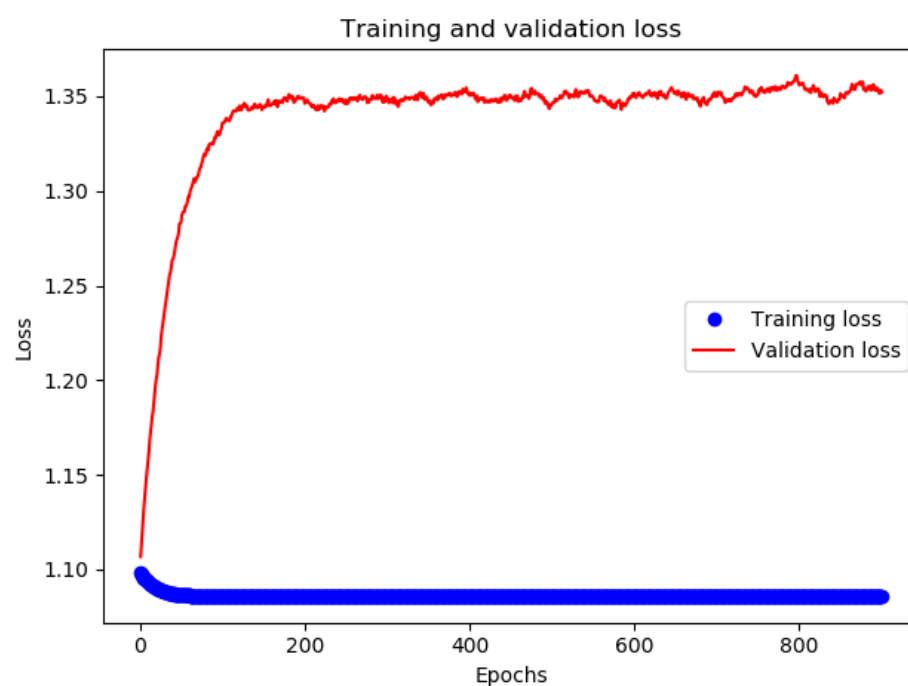


Рисунок 11 – Ошибки на 3 слоях с 900 эпохами

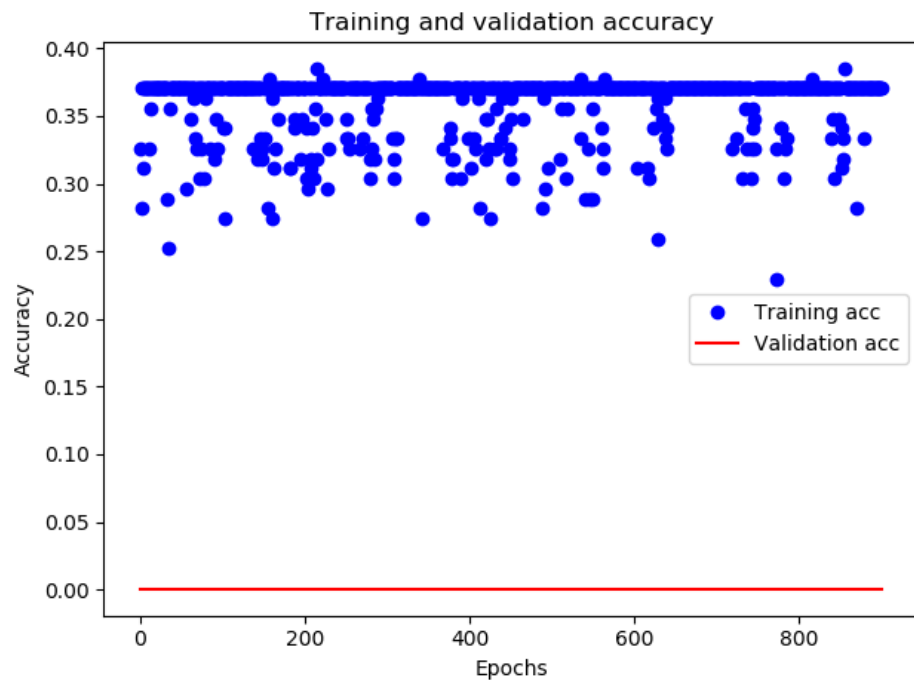


Рисунок 12 – Точность на 3 слоях с 900 эпохами

Видим, что для 450 эпох результат хороший. Попробуем увеличить число нейронов до 8 во 2 слое.

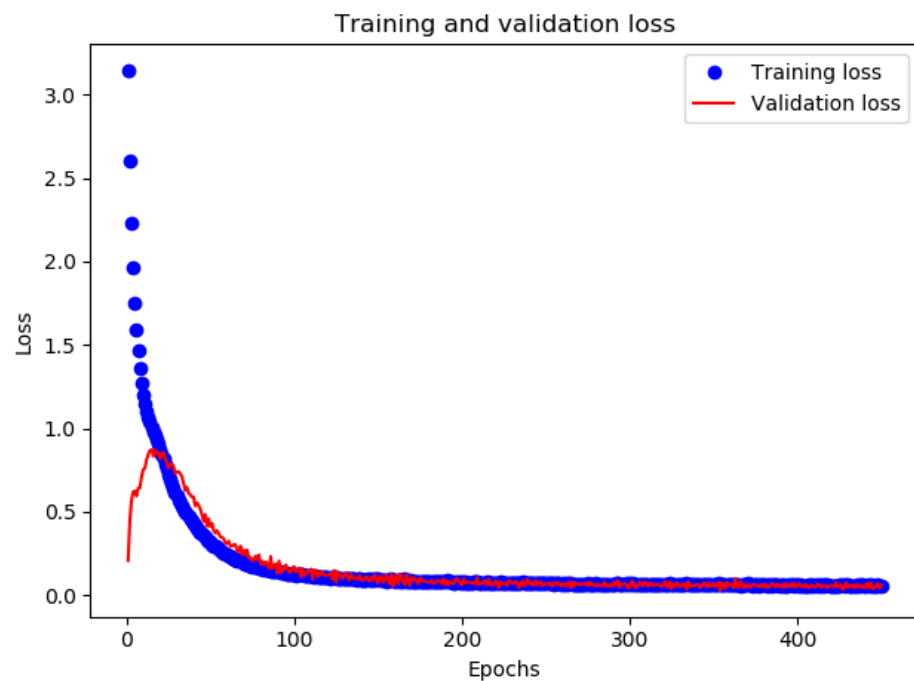


Рисунок 13 – Ошибки на 3 слоях с 450 эпохами



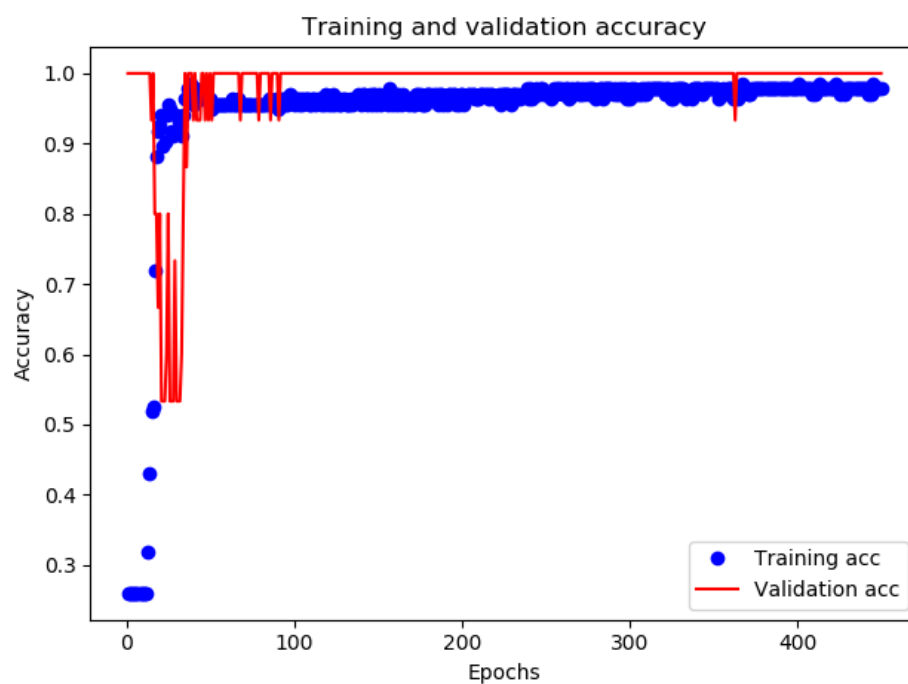


Рисунок 14 – Точность на 3 слоях с 450 эпохами

По рис. 13 и 14 видим, что параметры - 3 слоя на 3, 4, 8 нейронов, 450 эпох и размер батча равный 10 – нам подходят. Проверим еще на 900 эпохах.

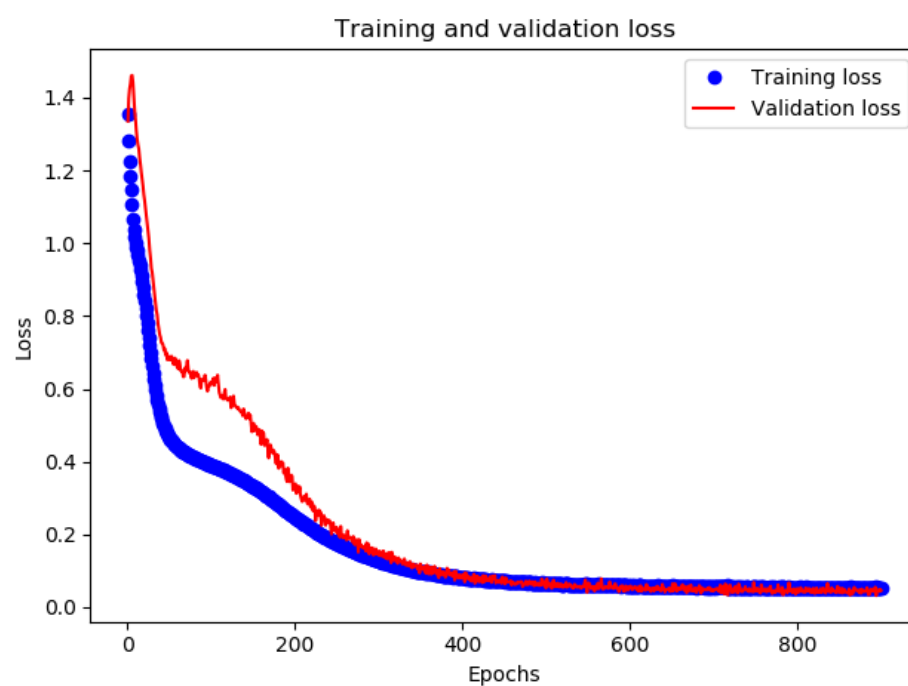


Рисунок 15 – Ошибки на 3 слоях с 900 эпохами

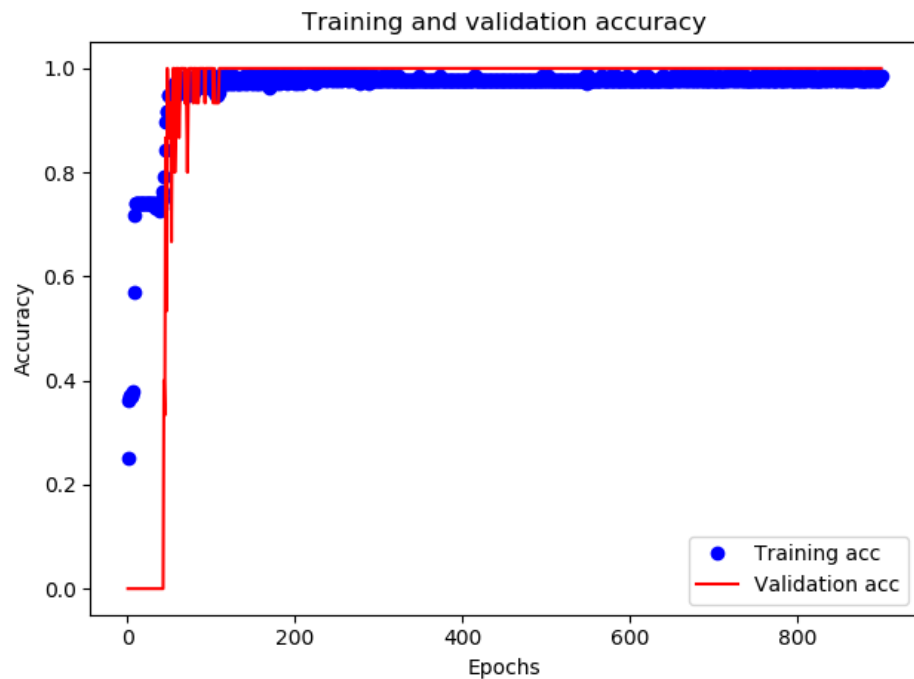


Рисунок 16 – Точность на 3 слоях с 900 эпохами

На рис. 15 явно видно переобучение модели. Следовательно, лучшим будет случай на рис. 13 и 14.

### **Выводы.**

В ходе работы были изучены основы работы с искусственными нейронными сетями с использованием python и keras, исследовано поведение сети в зависимости от ее модели и параметров обучения и выбрана наилучшая модель.

## ПРИЛОЖЕНИЕ А

### ИСХОДНЫЙ КОД ПРОГРАММЫ

```
import pandas
from sklearn.preprocessing import LabelEncoder
from tensorflow.keras.utils import to_categorical
from tensorflow.keras.layers import Dense
from tensorflow.keras.models import Sequential
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'])

history = model.fit(X, dummy_y, epochs=450, batch_size=10,
validation_split=0.1)

history_dict = history.history

loss_values = history_dict['loss']
val_loss_values = history_dict['val_loss']
acc = history_dict['accuracy']
val_acc = history_dict['val_accuracy']
epochs = range(1, len(loss_values) + 1)

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

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