

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

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

Студент гр. 8383

Преподаватель

Бабенко Н.С.

Жангиров Т.Р.

Санкт-Петербург

2021

Цель.

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

Задание.

- Ознакомиться с задачей классификации
- Загрузить данные
- Создать модель ИНС в Keras
- Настроить параметры обучения
- Обучить и оценить модель

Требуется:

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

Выполнение работы.

Были подготовлены данные для ИНС – приведены к категориальному виду, отделены ответы и параметры. Листинг:

```
import pandas
import numpy
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)
```

Графики выводятся с помощью библиотеки `matplotlib.pyplot`. Листинг:

```
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()

#Построение графика ошибки

plt.clf()
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()
```

Для начала была выбрана сеть с двумя слоями:

- 3 нейрона, Relu (входной)
- 4 нейрона, Softmax (выходной)

Сеть была обучена на 100 эпохах, `batch_size=10`, графики приведены на рис. 1-2.

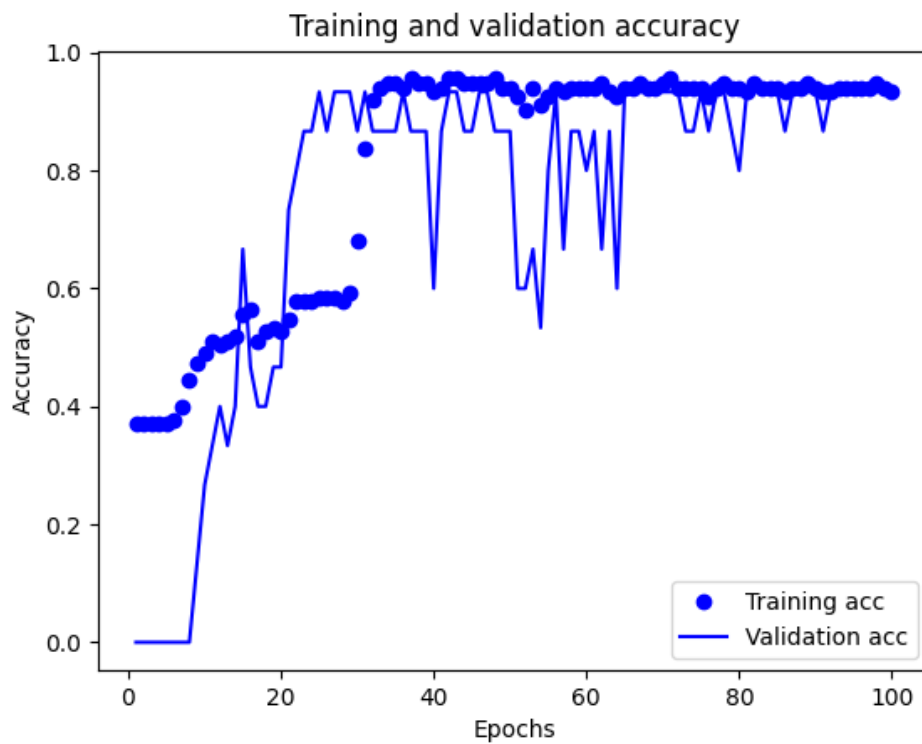


Рисунок 1 – Обучение сети из 2-х слоев, точность

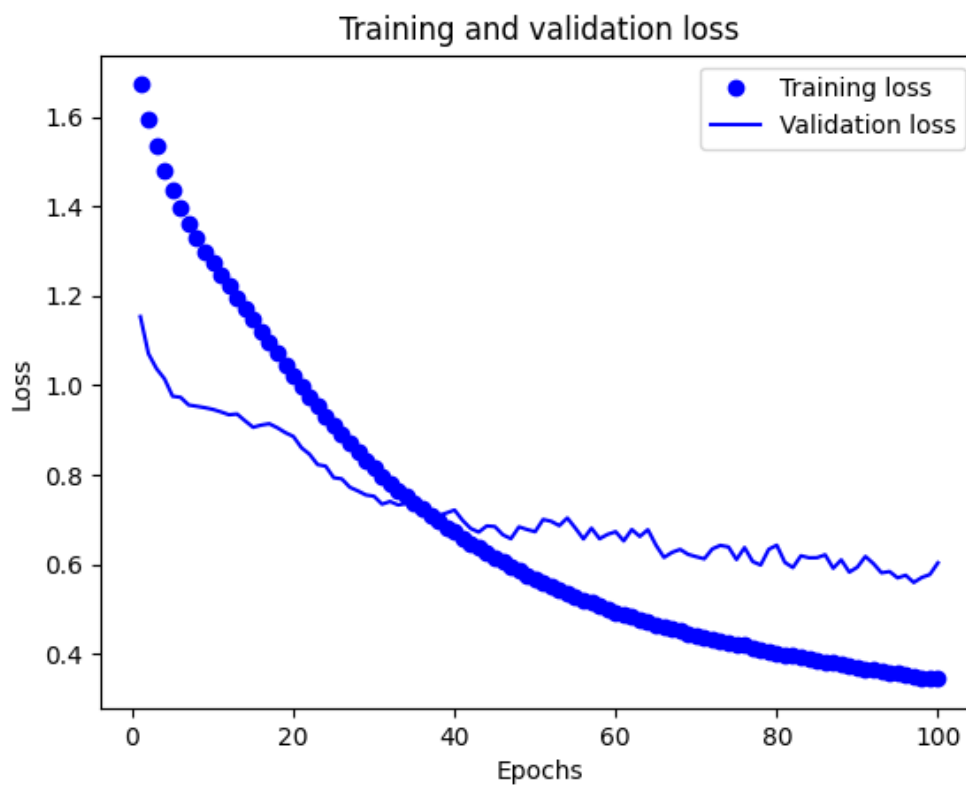


Рисунок 2 – Обучение сети из 2-х слоев, потери

Несмотря на неплохие значения точности при обучении, при проверке точность нестабильна и не достигает значения 100%. Потери: > 0.3 при обучении, > 0.6 при проверке, также высокие.

В сеть был добавлен еще один слой, теперь ее структура:

- 3 нейрона, Relu (входной)
- 15 нейронов, Relu (скрытый)
- 4 нейрона, Softmax (выходной)

Сеть была обучена на 100 эпохах, `batch_size=10`, графики приведены на рис. 3-4.

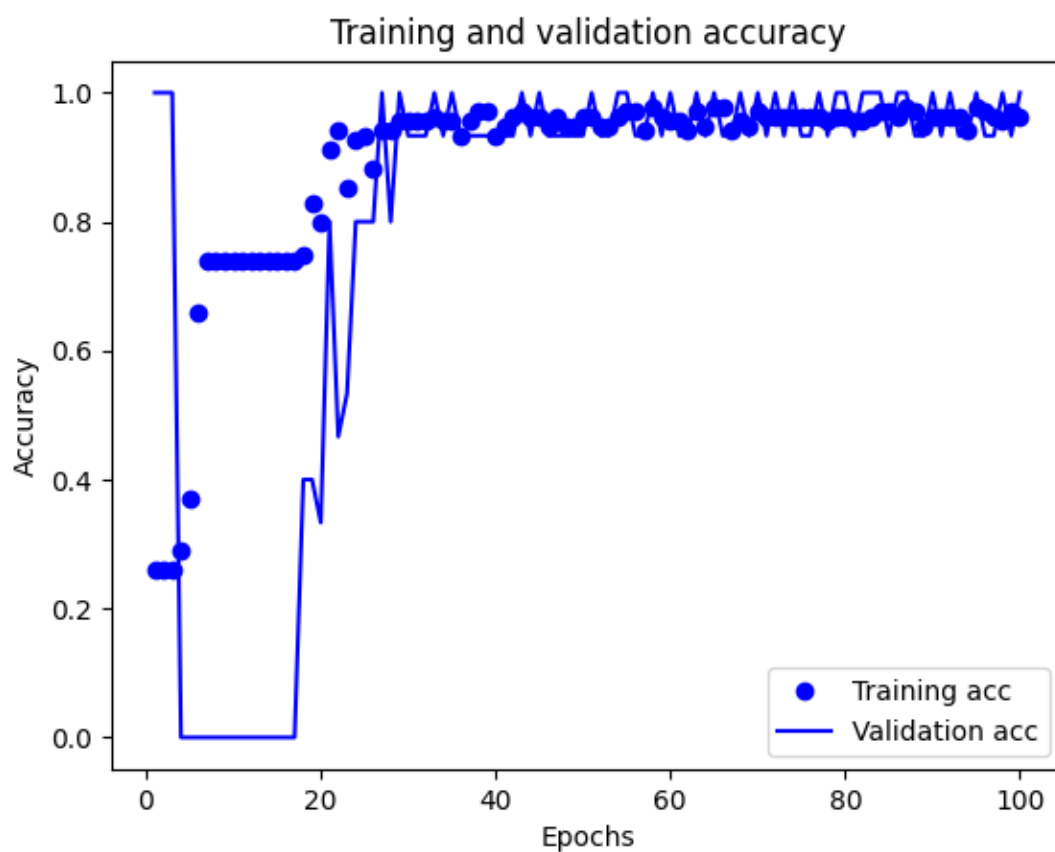


Рисунок 3 – Обучение сети из 3-х слоев, точность

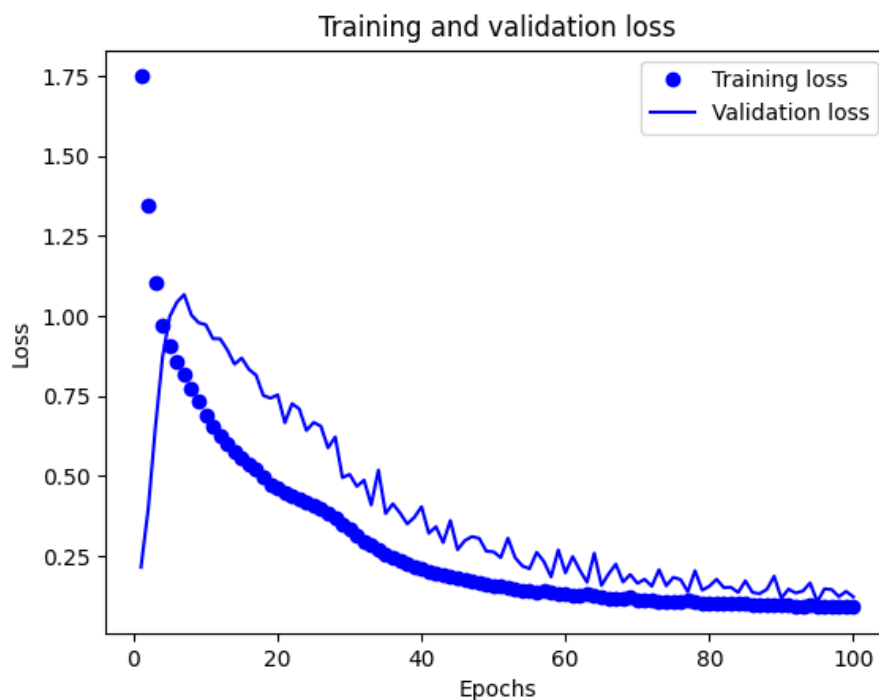


Рисунок 4 – Обучение сети из 3-х слоев, потери

Добавление скрытого слоя значительно уменьшило потери, повысило точность на данных не для обучения. За 100 эпох потери при обучении – 0.1684, точность при проверке – 100%.

В скрытом слое было увеличено число нейронов. Теперь структура сети:

- 3 нейрона, Relu (входной)
- 50 нейронов, Relu (скрытый)
- 4 нейрона, Softmax (выходной)

Сеть была обучена на 100 эпохах, `batch_size=10`, графики приведены на рис. 5-6.

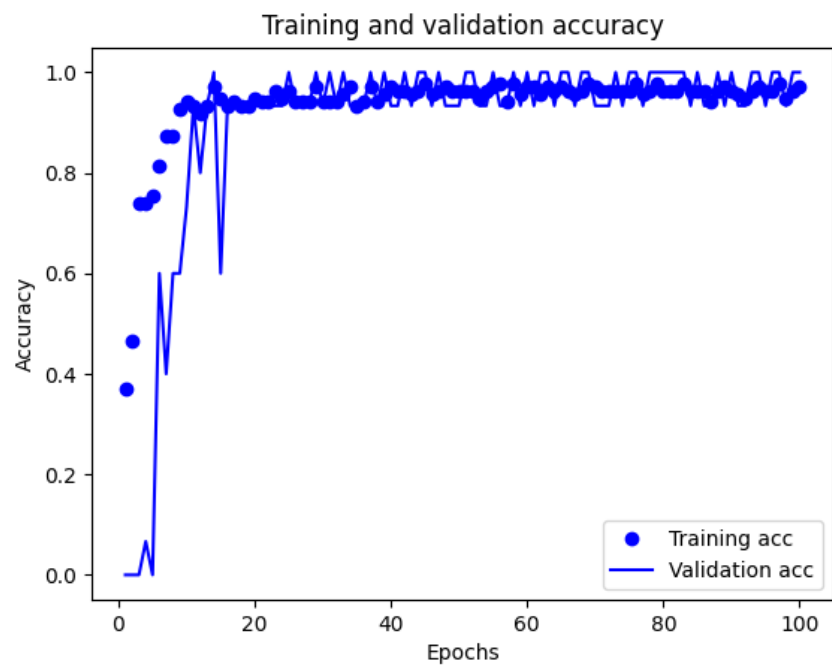


Рисунок 5 – Обучение сети из 3-х слоев, точность

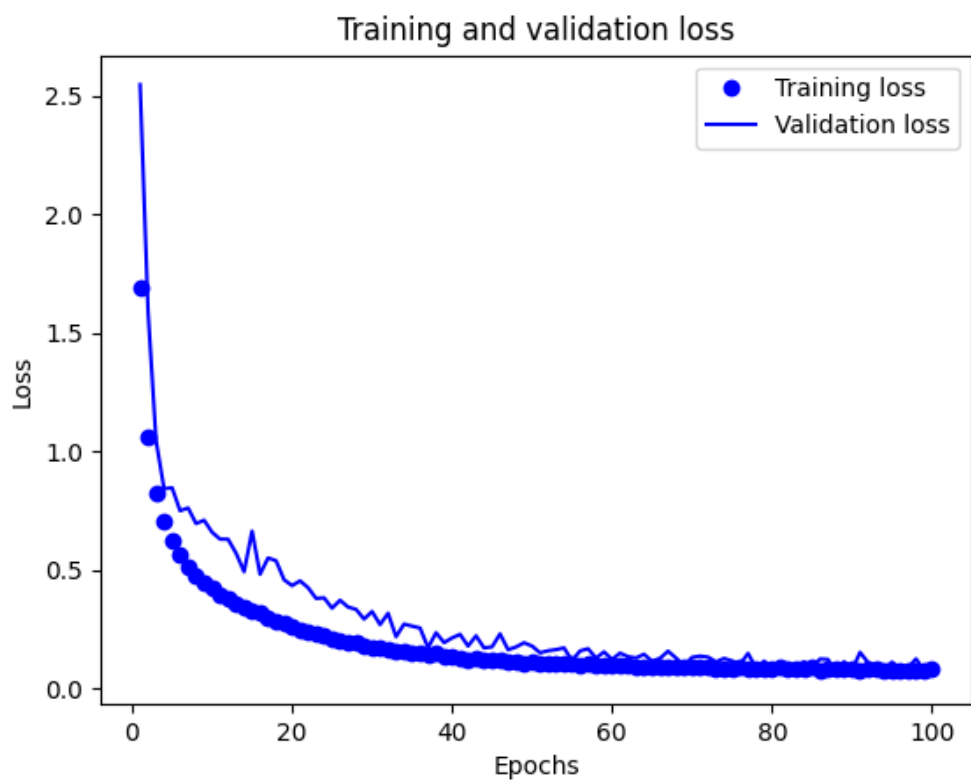


Рисунок 6 – Обучение сети из 3-х слоев, потери

При увеличении нейронов уменьшились потери, а рост точности ускорился. Модель еще быстрее обучается и также достигает 100% точности при проверке. Увеличение числа нейронов улучшило сеть.

В сеть был добавлен еще один скрытый слой. Структура сети:

- 3 нейрона, Relu (входной)
- 50 нейронов, Relu (скрытый)
- 50 нейронов, Relu (скрытый)
- 4 нейрона, Softmax (выходной)

Сеть была обучена на 100 эпохах, `batch_size=10`, графики приведены на рис. 7-8.

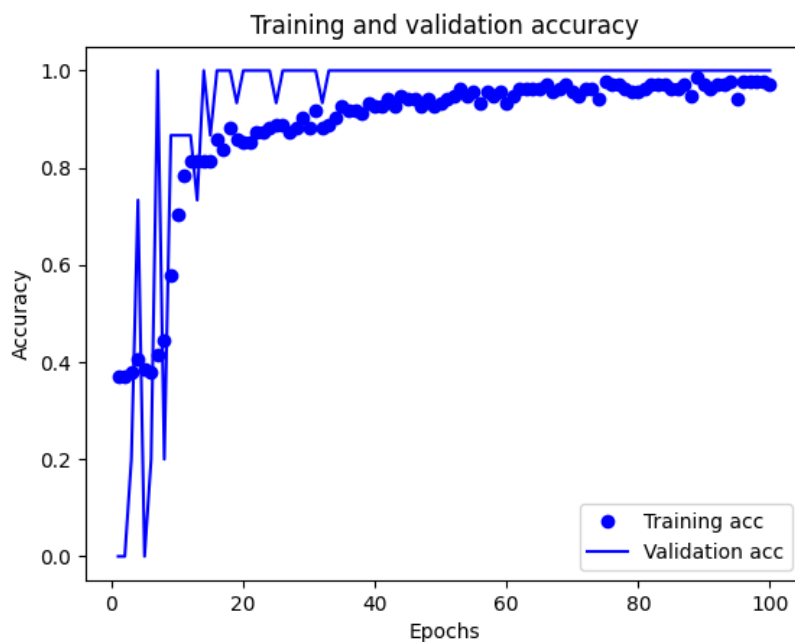


Рисунок 7 – Обучение сети из 4-х слоев, точность

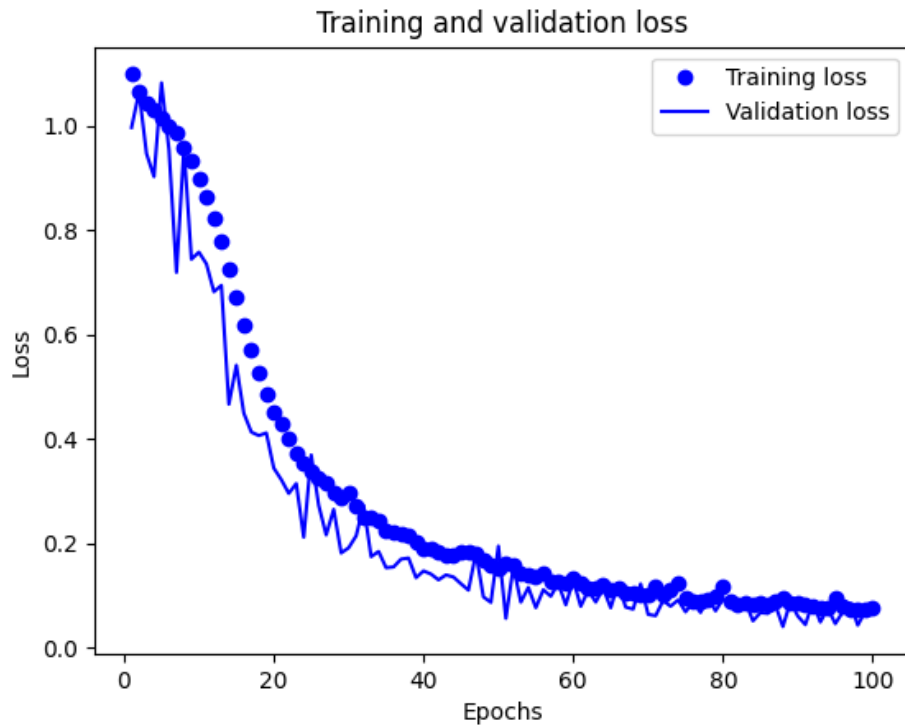


Рисунок 8 – Обучение сети из 4-х слоев, потери

Сет обладает лучшей точностью на данных для проверки, однако в то же время первые 40 эпох наблюдаются довольно высокие потери, точность при обучении ниже. Но так как именно точность при проверке является решающим показателем, данную сеть можно считать наиболее подходящей для задачи.

Были изменены параметры обучения:

- `batch_size = 50`

Графики обучения сети приведены на рис. 9-10.

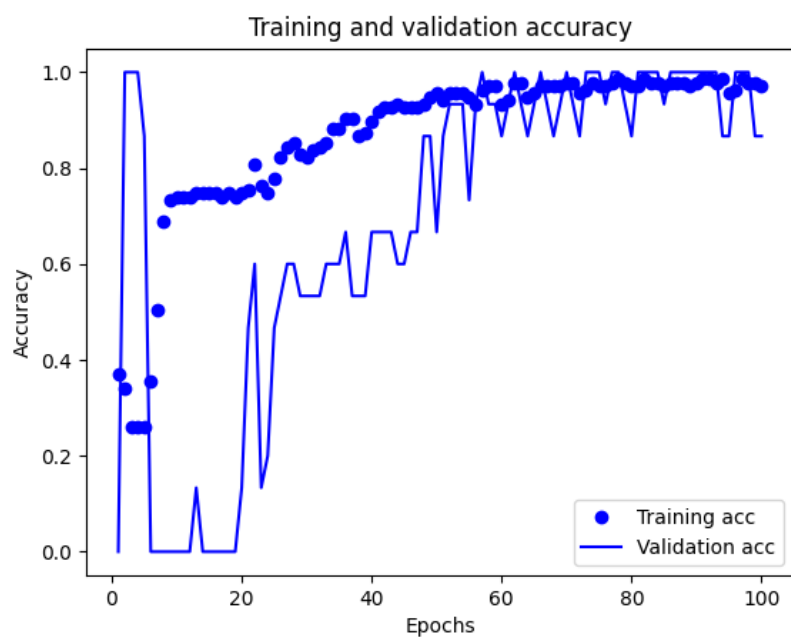


Рисунок 9 – Обучение при $\text{batch_size} = 50$, точность

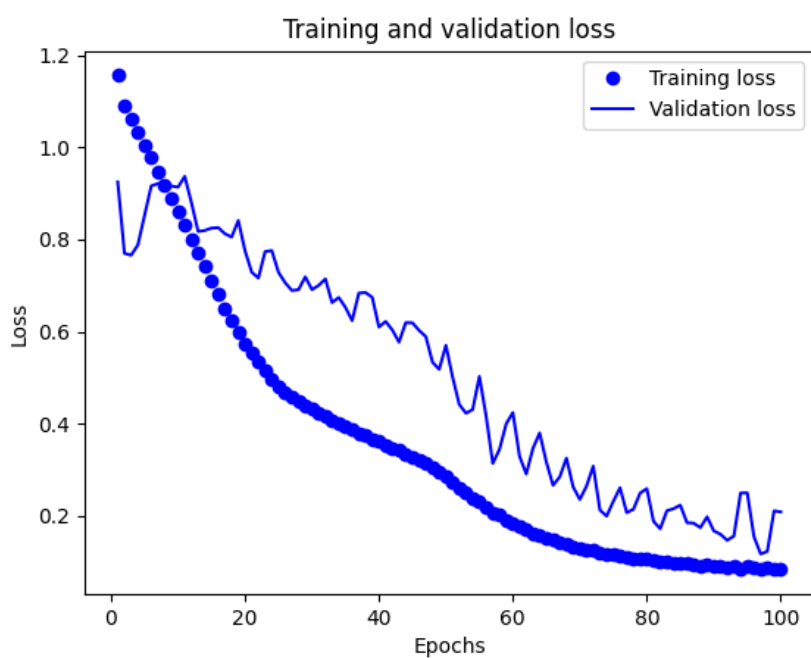


Рисунок 10 – Обучение при $\text{batch_size} = 50$, потери

Из графиков видно, что сеть хуже обучается при увеличении batch_size , так как слишком редко происходят итерации изменения весов.

Параметр batch_size был оставлен равным 10, при этом:

- $\text{validation_split} = 0.25$

Графики обучения сети приведены на рис. 11-12.

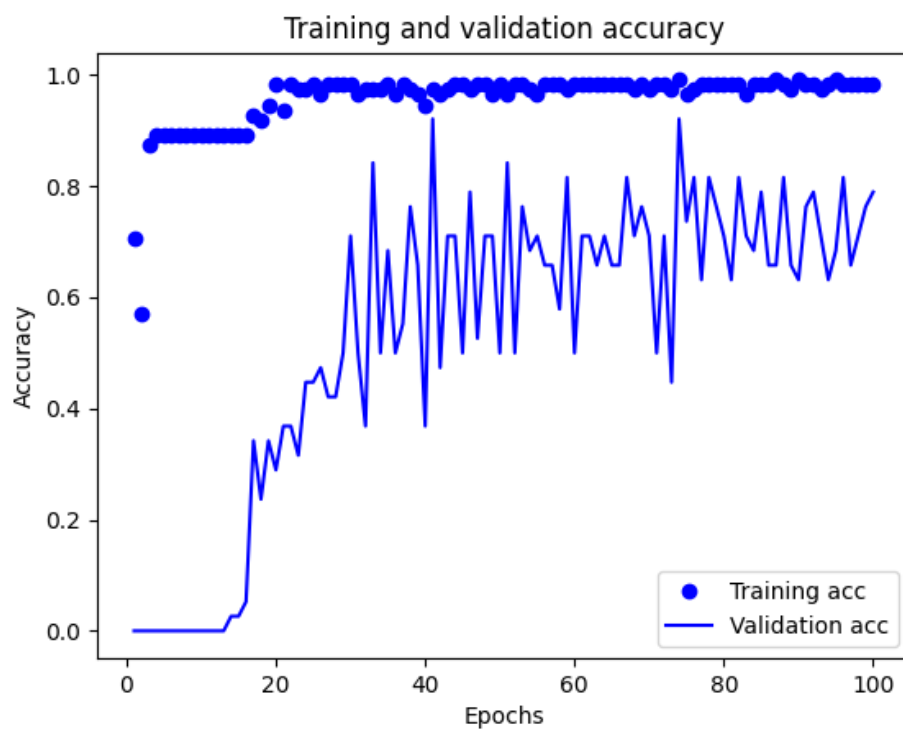


Рисунок 11 – Обучение при меньших данных для обучения, точность

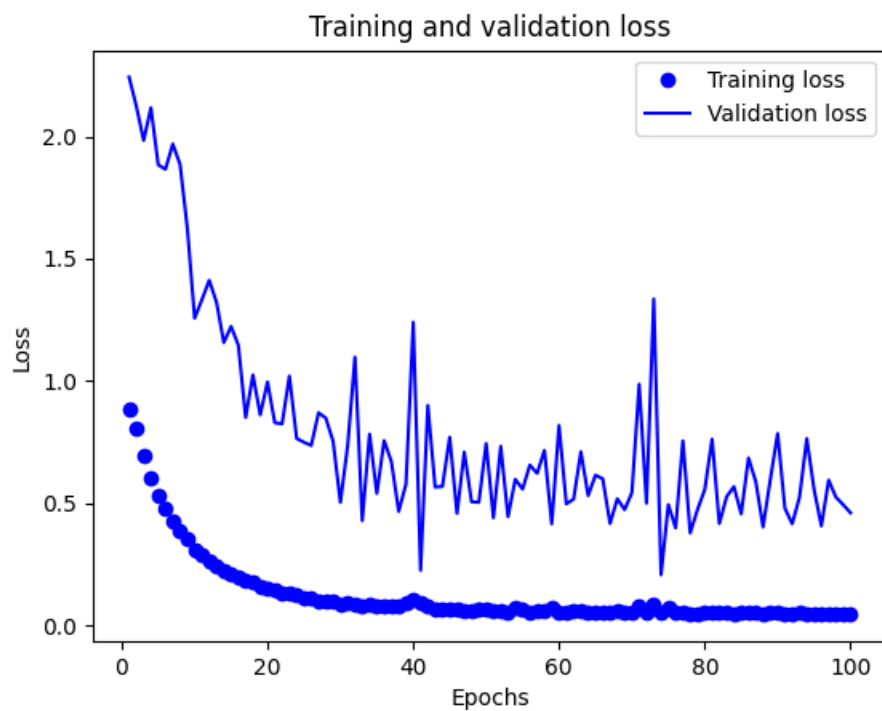


Рисунок 12 – Обучение при меньших данных для обучения, потери

Сеть достигает высокой точности при обучении, но при проверке точность низкая, а потери очень высокие. Сети не хватает 75% данных для обучения данной задаче.

Параметр `validation_split` был оставлен равным 0.1, при этом:

- `epochs = 40`

Графики обучения сети приведены на рис. 13-14.

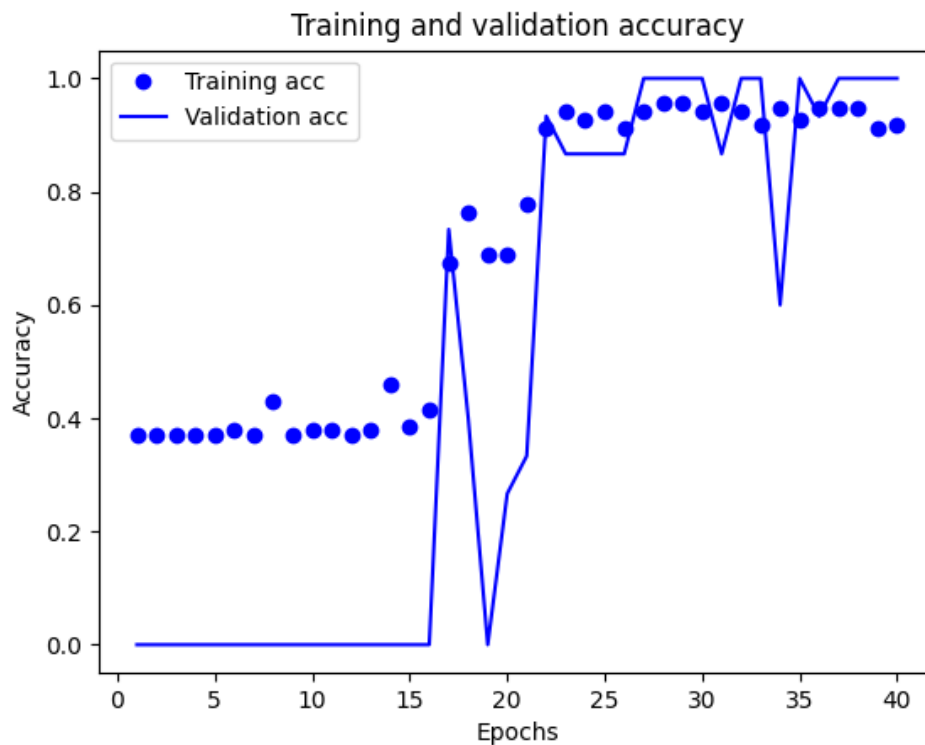


Рисунок 13 – Обучение за 40 эпох, точность

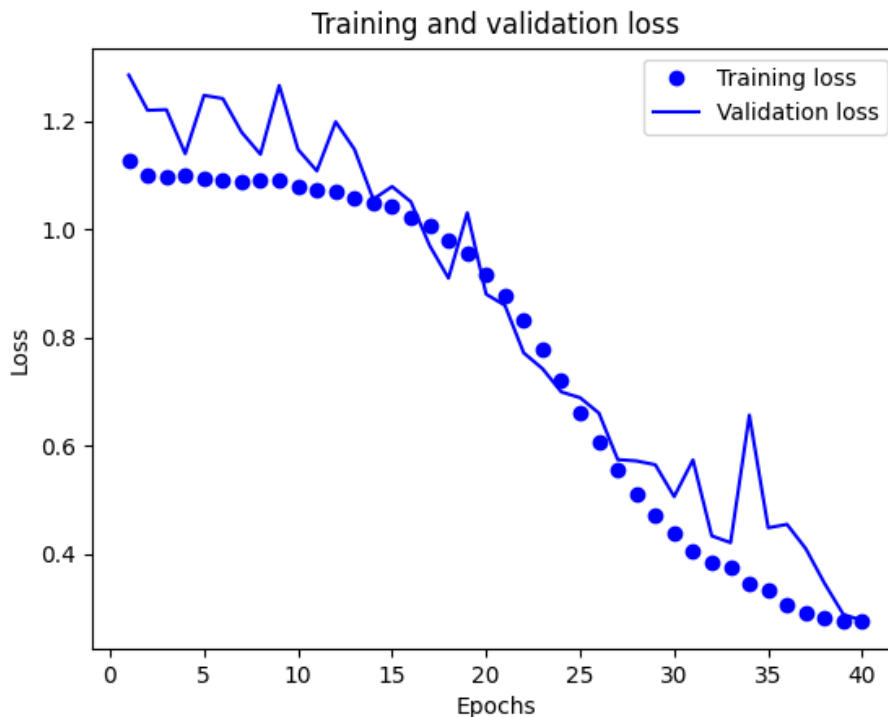


Рисунок 14 – Обучение за 40 эпох, потери

Сеть достигает точность на данных для проверки в 100% за 40 эпох. Сеть с данными параметрами и структурой можно использовать для решения поставленной задачи.

Выводы.

В результате выполнения лабораторной работы была изучена структура искусственной нейронной сети, влияния числа слоев и нейронов на ее производительность, а также параметров обучения. Была выбрана оптимальная сеть для выполнения поставленной задачи классификации.