МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МОЭВМ

ОТЧЕТ

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

Студент гр. 8382 _____ Янкин Д.О.

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.

Постановка задачи.

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

Выполнение работы.

Данные для анализа считываются из файла iris.csv. Данные разделяются на параметры цветов X и сорта цветов Y, которые преобразуются в двоичную матрицу:

```
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)
```

Создается модель со входным слоем из четырех нейронов и выходным из трех нейронов, устанавливаются параметры обучения, происходит обучение 300 эпох.

```
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=300, batch_size=10,
validation_split=0.1)
```

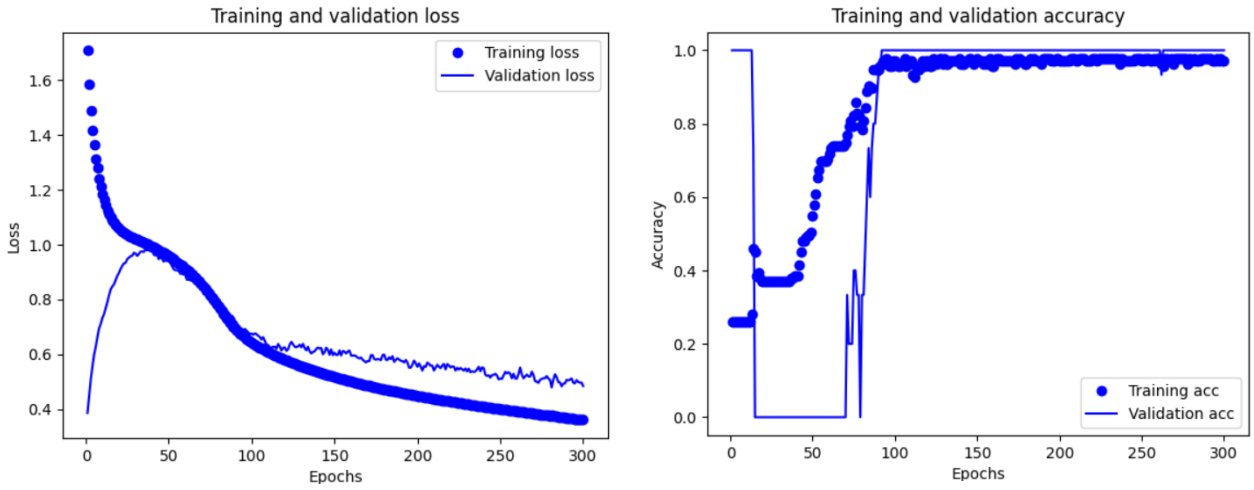

Рисунок 1. Ошибки и точность изначальной сети без скрытых слоев

Показатели обучения нейронной сети изображены на графиках на рисунке 1, сеть начинает показывать нормальный результат после 90 эпох, однако эти графики сильно меняются при каждом запуске программы.

В модель был добавлен один слой на 16 нейронов. Хорошие результаты сеть показала после 60 эпох.

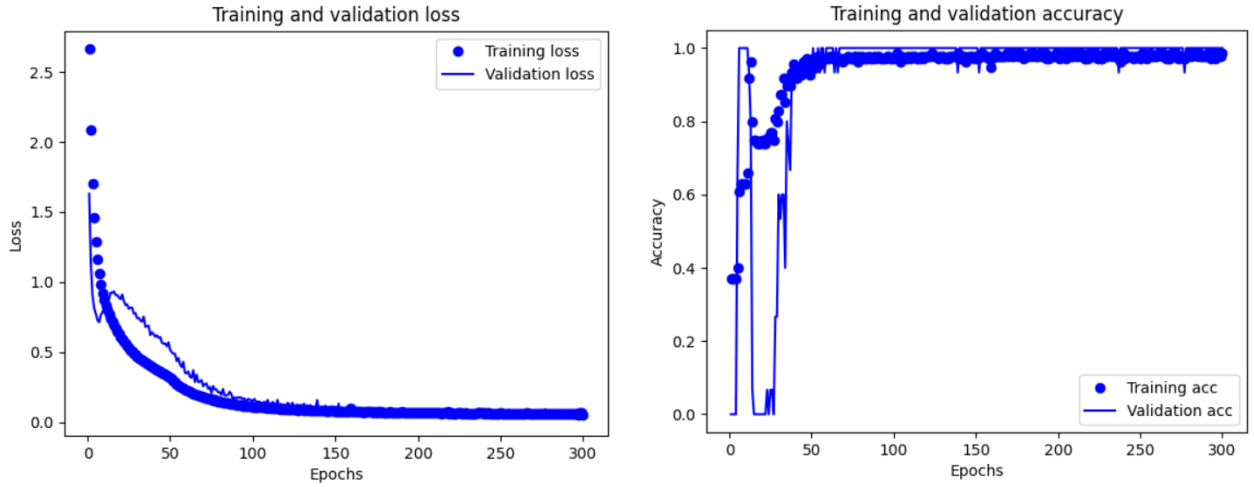

Рисунок 2. Ошибки и точность сети со скрытым слоем на 16 нейронов

В модель был добавлен еще один слой (уже второй скрытый) на 16 нейронов. Сеть стала обучаться быстрее.

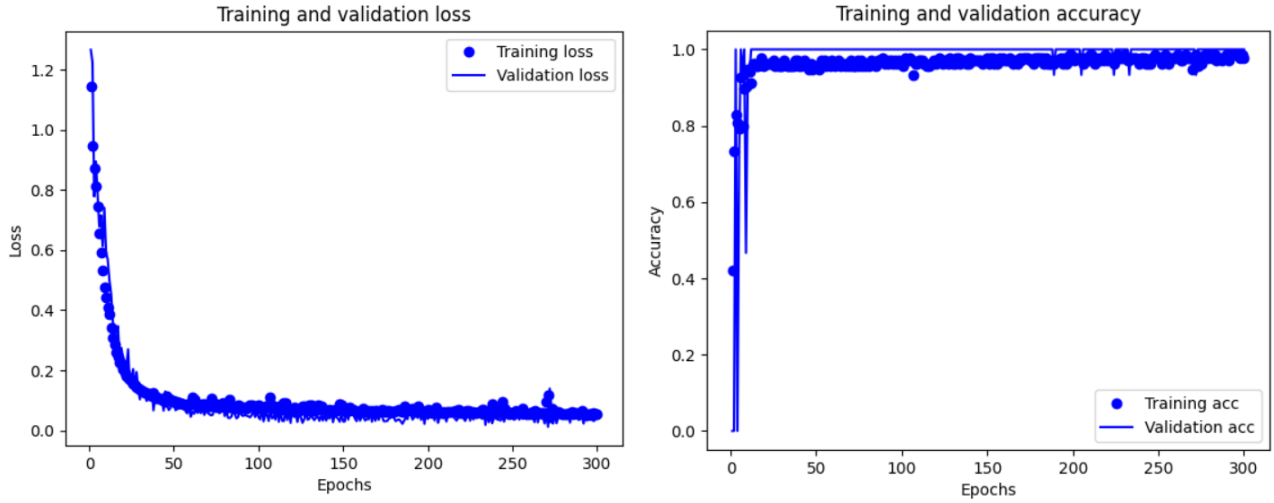

Рисунок 3. Ошибки и точность сети с двумя скрытыми слоями на 16 нейронов

Сделана сеть с тремя скрытыми сломи по 16 нейронов. Результаты незначительно отличаются от предыдущей сети, скорость обучения осталась примерно на том же уровне.

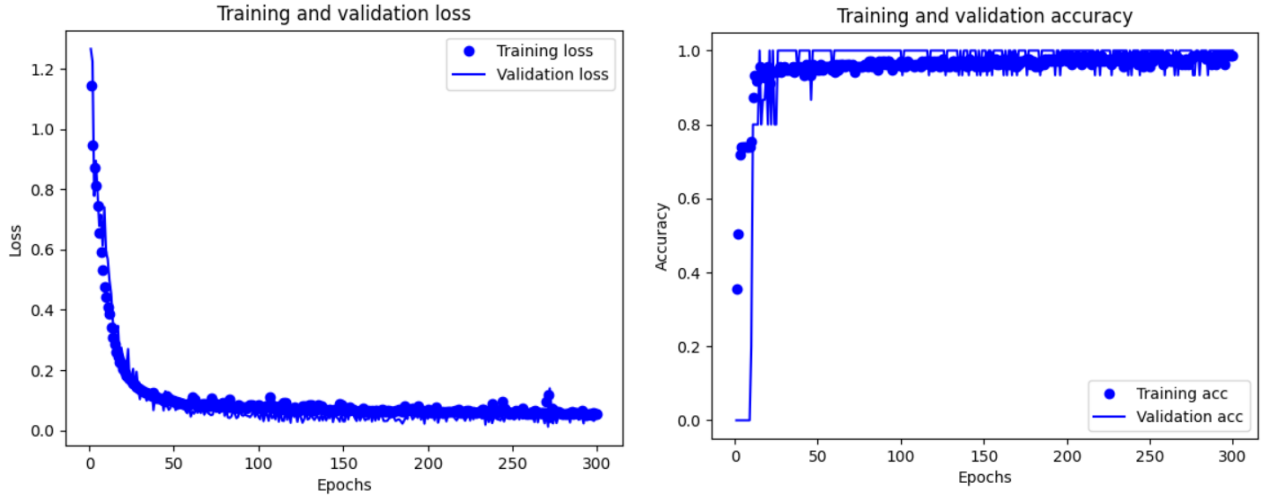

Рисунок 4. Ошибки и точность сети с тремя скрытыми слоями на 16 нейронов

Создана сеть с двумя скрытыми слоями по 8 нейронов.

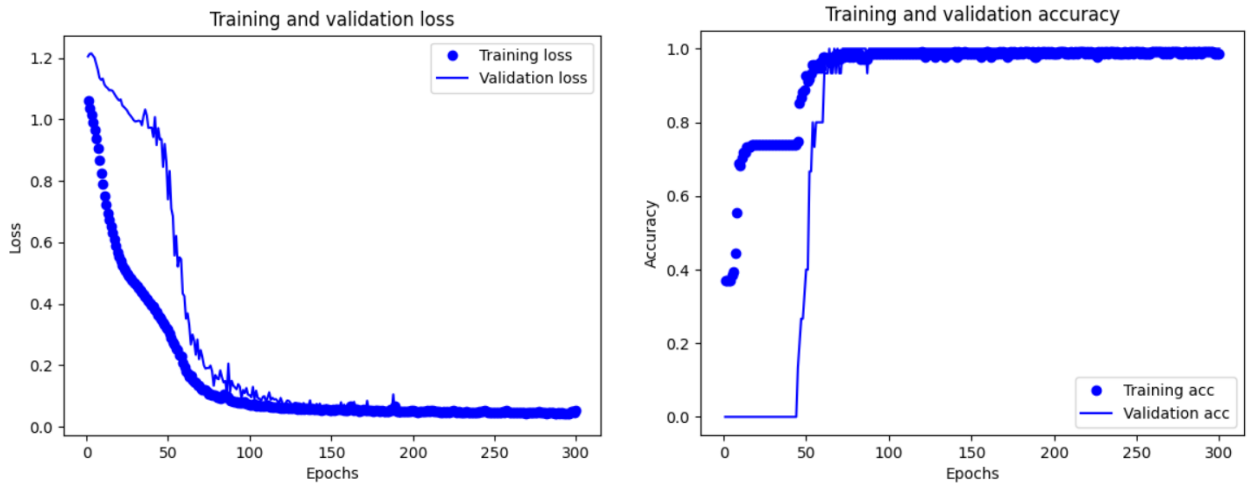

Рисунок 5. Ошибки и точность сети с двумя скрытыми слоями на 8 нейронов

Создана сеть с двумя скрытыми слоями по 32 нейрона.

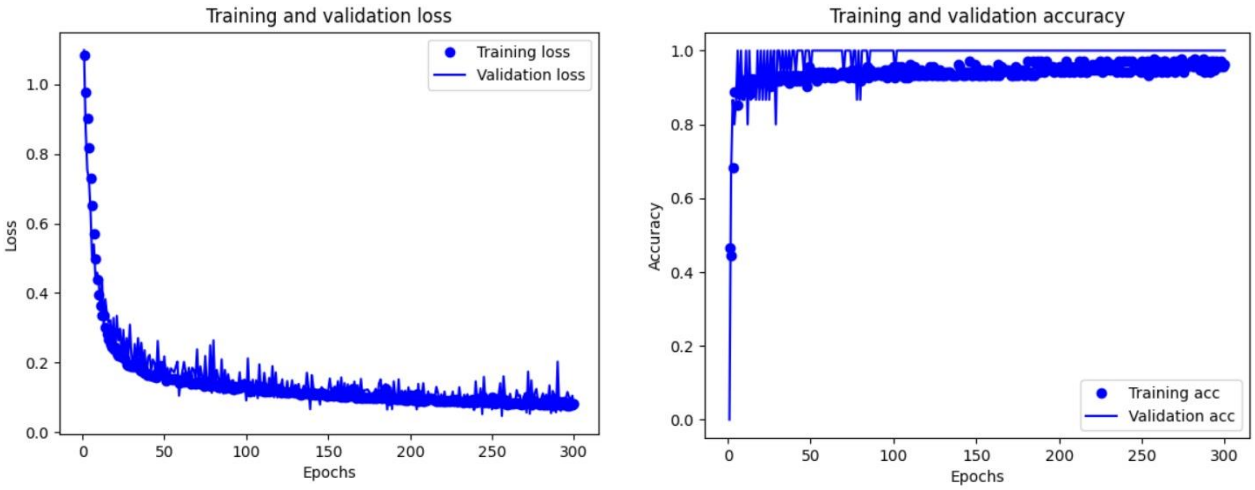

Рисунок 6. Ошибки и точность сети с двумя скрытыми слоями на 32 нейронов

В конечном варианте оставлен последний вариант – два скрытых слоя по 32 нейрона.

Параметр batch_size определяет, через какое количество обработанных элементов обновляются весы. Значение параметра уменьшено с 10 до 4

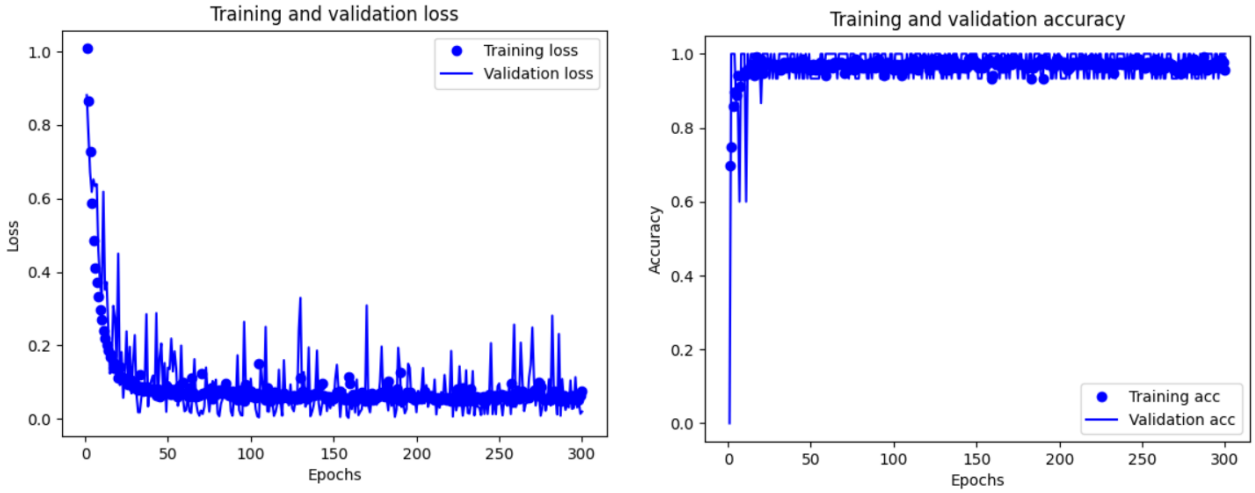

Рисунок 7. Параметр batch_size уменьшен с 10 до 4

Параметр batch_size установлен на 25

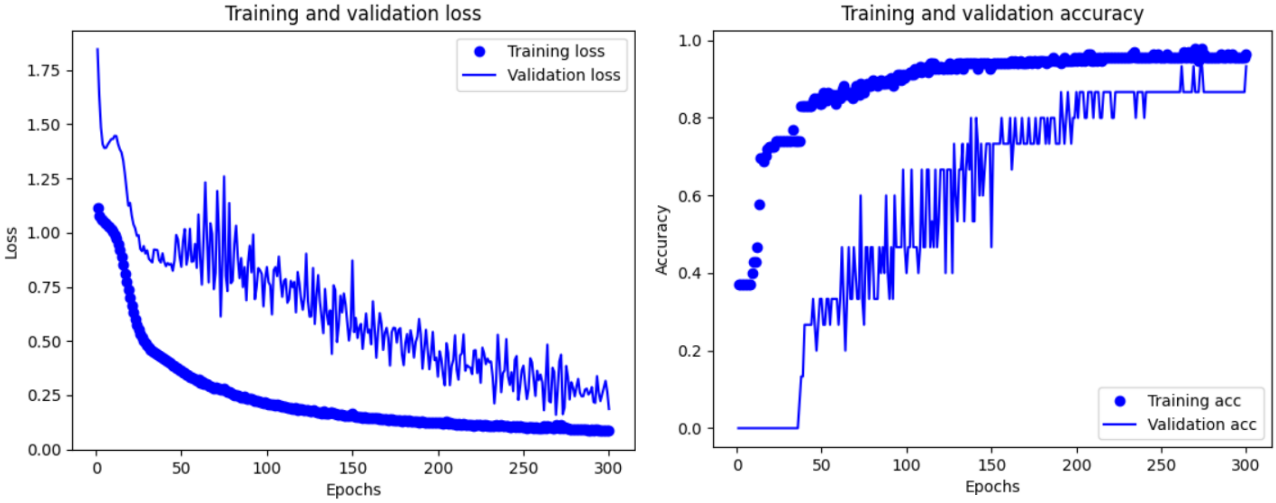

Рисунок 8. Параметр batch_size увеличен с 10 до 25

При больших значениях batch_size повышаются ошибки, снижается точность. При меньших значения точность увеличивается, но повышается время обучения из-за большего объема вычислений.

Параметр validation_split определяет, какая доля датасета будет использоваться для валидации. Он был установлен в значение 0.3 (был 0.1). Сеть обучилась только до точности 50%. Это объясняется тем, что после увеличения количества валидационных данных уменьшилось количество данных для обучения.

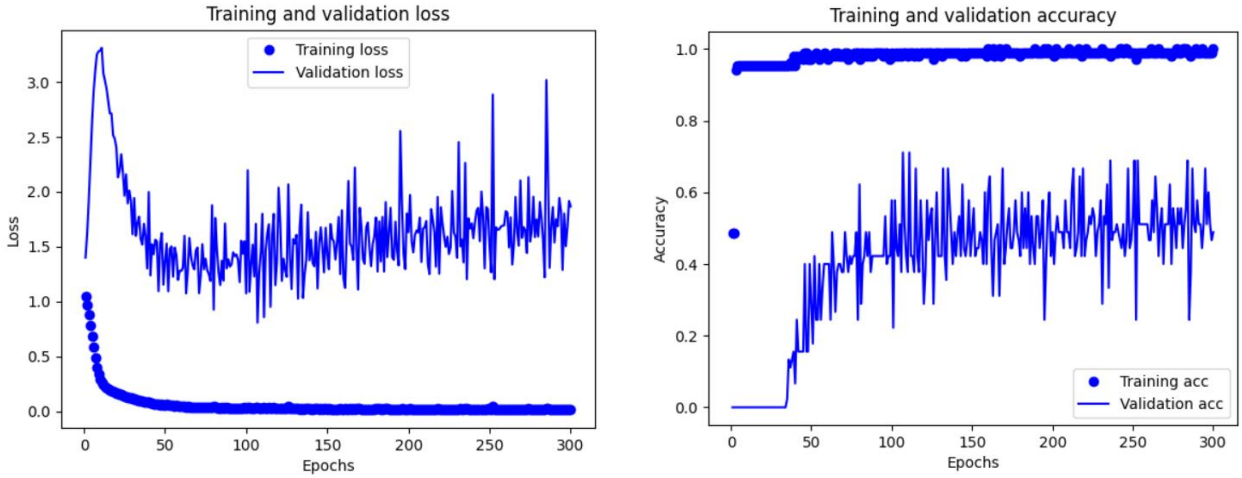

Рисунок 8. Параметр validation_split увеличен с 0.1 до 0.3

Уменьшение количества валидационных данных увеличивает количество обучающих, за счет чего увеличивается качество обучения. Но точность полученного результата при этом может пострадать.

В итоговом варианте выбрана сеть с двумя скрытыми слоями по 32 нейрона, batch_size = 10, validation_split = 0.1. В большинстве случаев точность >85% достигается за 10-30 эпох, но бывают и неудачные случаи, при которых обучение до хороших результатов сильно затягивается.

Выводы.

В ходе лабораторной работы было изучено влияние количества слоев и нейронов на скорость и качество обучения, также были исследованы параметры обучения. Была выбрана конфигурация сети, обеспечивающая быстрое обучение и классификацию для трех сортов цветов.

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(32, activation='relu'))
model.add(Dense(32, activation='relu'))
model.add(Dense(3, activation='softmax'))

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

history = model.fit(X, dummy_y, epochs=300, batch_size=10,
validation_split=0.1)
history_dict = history.history


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


plt.clf()
acc_values = history_dict['acc']
val_acc_values = history_dict['val_acc']
```

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