

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. 7382

Бахеров Д.В.

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.

Порядок выполнения работы.

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

Требования к выполнению задания.

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

Ход работы.

При изначальных данных графики ошибки и точности модели предоставлены на рис. 1 и 2 соответственно.

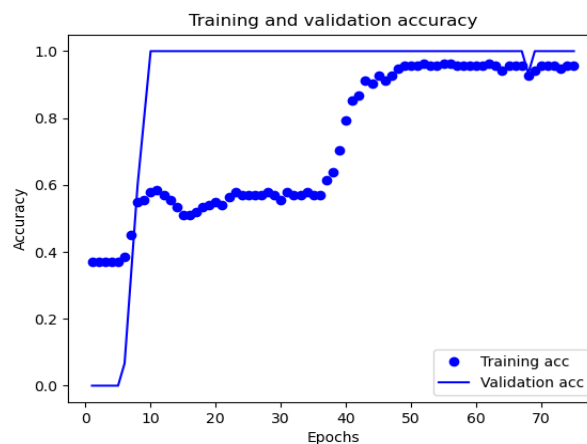


Рисунок 1 – Точность модели при изначальных данных

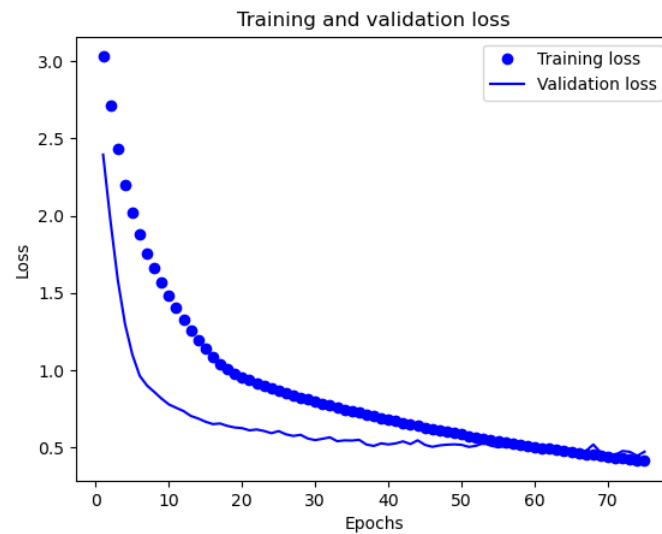


Рисунок 2 – Ошибки модели при изначальных данных

Изменим количество эпох с 75 до 350, ошибки и точность модели изменились. Результат предоставлен на рис. 3 и 4 соответственно.

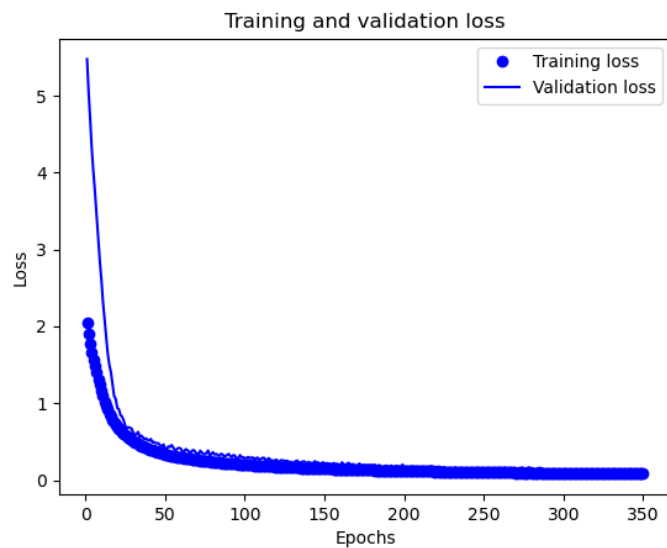


Рисунок 3 – Ошибки модели при 350 эпох

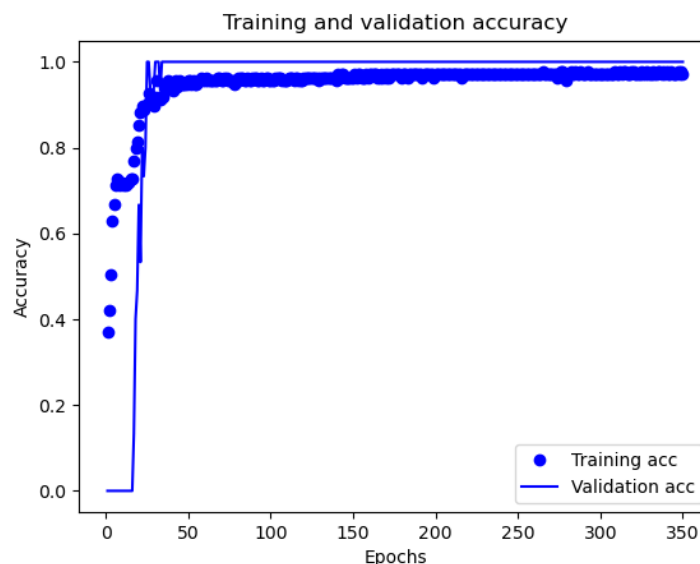


Рисунок 4 – Точность модели при 350 эпох

Увеличим количество нейронов в 1 слое до 16. Как видно из рис. 5 и 6, модель смогла достичь максимальной точности примерно за 40 поколений, а потери были минимизированы за 150 поколений. При увеличении количество слоев результат не становился лучше, поэтому дальнейшее обучение смысла не имеет. Данный результат является наилучшим (см. рис. 5 и 6).

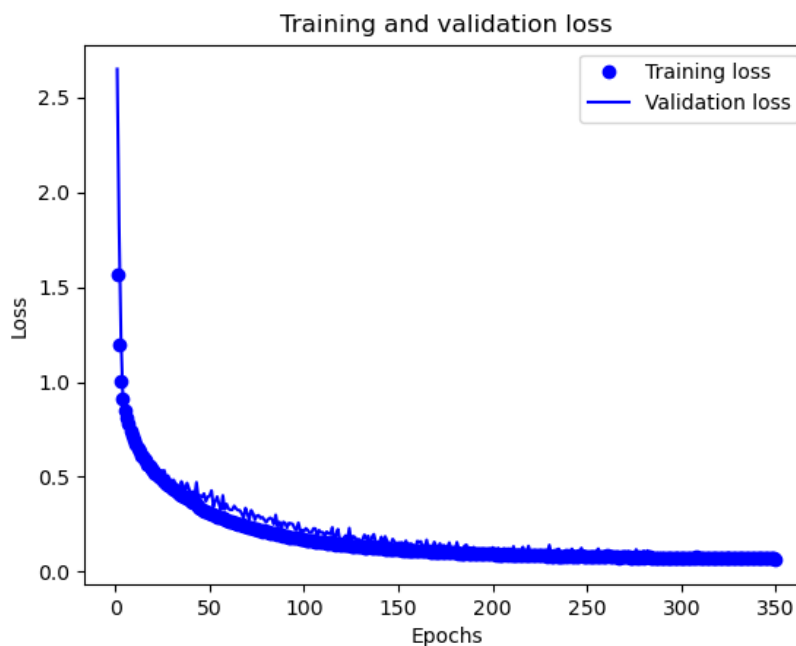


Рисунок 5 – Ошибки лучшей модели

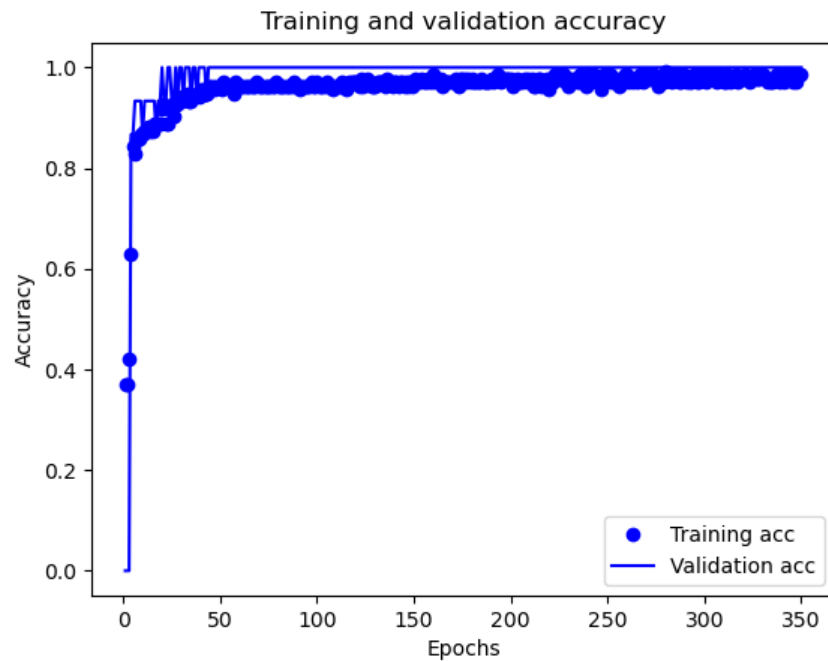


Рисунок 6 – Ошибки лучшей модели

Выводы.

Были изучены основы работы с искусственными нейронными сетями. Наблюдали за тем, как меняется результат в ходе изменений различных параметров, как в ходе построения модели, так и в ходе ее обучения.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

```
import pandas
from keras.layers import Dense
from keras.models import Sequential
from keras.utils import to_categorical
from sklearn.preprocessing import LabelEncoder
import matplotlib.pyplot as p

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

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

history = model.fit(X, dummy_y, epochs=350, batch_size=10, validation_split=0.1)

history_dict = history.history
loss_values = history_dict['loss']
val_loss_values = history_dict['val_loss']
acc_values = history_dict['accuracy']
val_acc_values = history_dict['val_accuracy']

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

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