

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. 8382

Гордиенко А.М.

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.

Порядок выполнения работы.

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

Требования к выполнению задания.

1. Изучить различные архитектуры ИНС (Разное кол-во слоев, разное кол-во 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.

Для изучения различной структуры ИНС была разработана первая модель сети, состоящая из двух слоев из 4 и 3 нейронов и с функциями 'relu' и 'softmax' соответственно.

В качестве параметров обучения были выбраны: Оптимизатор – adam, функция потерь - categorical_crossentropy, метрика – точность, количество эпох – 75, размер партии – 10.

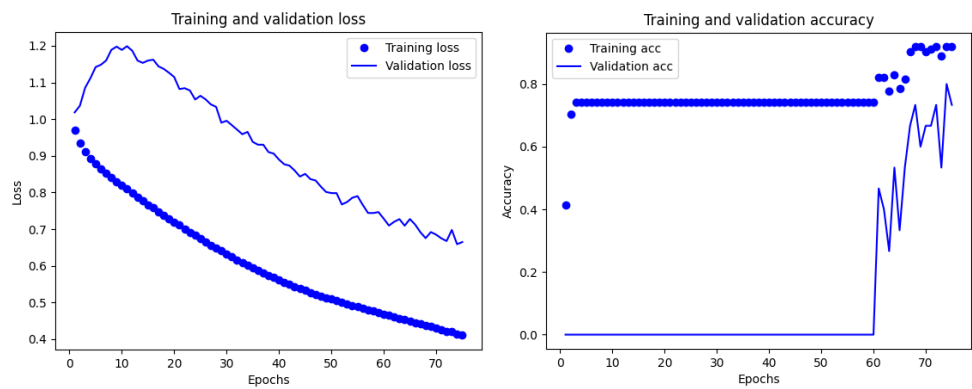


Рисунок 1, 2 – Графики потерь и точности первой модели

Наблюдаем малую точность и большие ошибки. Увеличим количество нейронов в первом слое до 30.

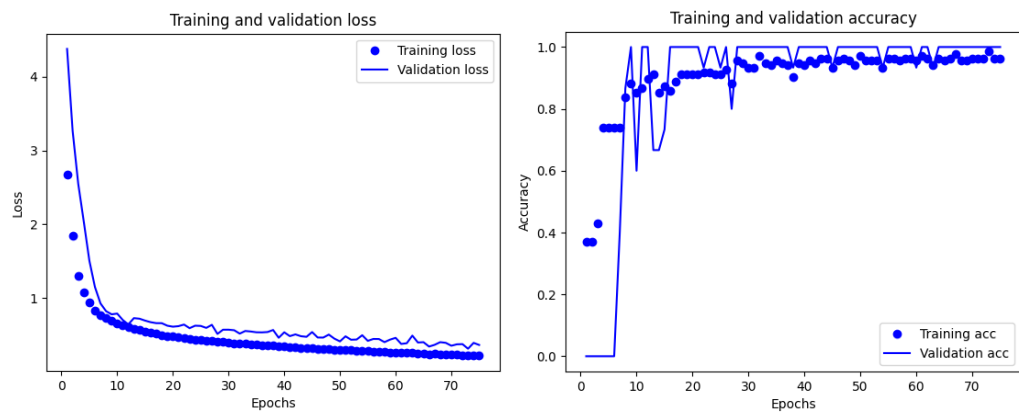


Рисунок 3, 4 – Графики потерь и точности второй модели

Наблюдаем заметное увеличение точности и меньше ошибок. Продолжим модифицировать сеть. В этот раз добавим новый промежуточный слой из 30 нейронов с функцией 'relu'.

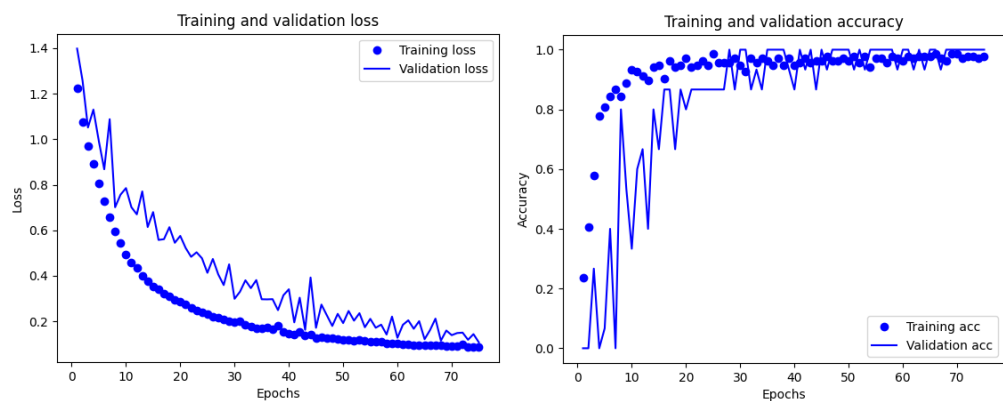


Рисунок 5, 6 – Графики потерь и точности третьей модели

Видим, что добавление нового слоя не сильно сказалось на результатах. Заменим функцию 'relu' в втором слое на 'sigmoid', а также увеличим количество нейронов на обоих слоях до 50.

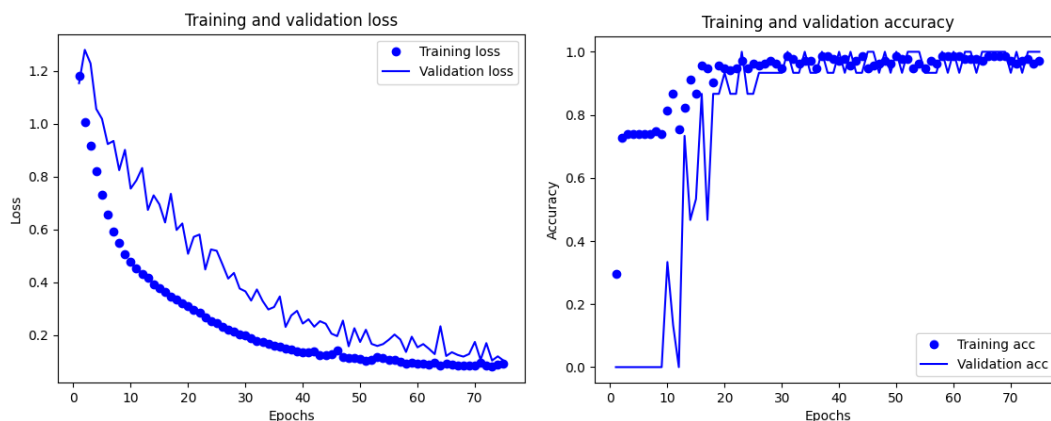


Рисунок 7, 8 – Графики потерь и точности четвертой модели

Можем видеть, что последние модификации не сильно отразились на результатах. Из чего можно сделать вывод, что есть некоторый порог, дальше которого нет смысла усложнять ИНС, поскольку улучшения точности это не повлечет.

Выводы.

В ходе выполнения лабораторной работы было изучено, какие параметры модели ИНС напрямую влияют на улучшение работы сети. Например, увеличение количество нейронов на слоях до некоторого числа будет гарантированно давать улучшение точности.

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
import matplotlib.colors as mclr
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(50, activation='relu'))
model.add(Dense(50, activation='sigmoid'))
model.add(Dense(3, activation='softmax'))
model.compile(optimizer='adam',loss='categorical_crossentropy',
metrics=['accuracy'])
H = model.fit(X, dummy_y, epochs=75, batch_size=10, validation_split=0.1)
loss = H.history['loss']
val_loss = H.history['val_loss']
acc = H.history['acc']
val_acc = H.history['val_acc']
epochs = range(1, len(loss) + 1)
#Построение графика ошибки
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()
#Построение графика точности
plt.clf()
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()
```