

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

ОТЧЕТ
по индивидуальному домашнему заданию
по дисциплине «Искусственные нейронные сети»
Тема: Классификация животного по его описанию

Студент гр. 8382	_____	Гордиенко А.М.
Студент гр. 8382	_____	Ершов М.И.
Преподаватель	_____	Жангиров Т.Р.

Санкт-Петербург
2021

Цель работы.

Построить модель классификации животных по его сведениям.

Задание.

Дан датасет булевых значений. Включает в себя информацию о животном из зоопарка, которые относятся к 7 классам.

Задача заключается в классификации животного по его описанию.

Ход работы.

Был импортирован набор данных и написана программа обучения нейронной сети и построения графиков точности и ошибки.

Датасет представляет из себя набор параметров:

1. Название животного
2. Наличие волосяного покрова (0, 1)
3. Наличие оперения (0, 1)
4. Животное яйцекладущее (0, 1)
5. Животное млекопитающее (0, 1)
6. Животное летающее (0, 1)
7. Животное водное (0, 1)
8. Животное хищник (0, 1)
9. Животное имеет зубы (0, 1)
10. Животное позвоночное (0, 1)
11. Животное дышит (0, 1)
12. Животное ядовитое (0, 1)
13. Животное имеет плавники (0, 1)
14. Животное имеет ноги (0, 2, 4, 6, 8)
15. Животное имеет хвост (0, 1)
16. Животное одомашненное (0, 1)
17. Класс размера животного (0, 1)
18. Принадлежность к классу (1, 2, 3, 4, 5, 6, 7)

Таким образом классификация будет проводится по 16 параметрам.

Данные считывались как тип *boolean*, не смотря на параметр [14], поскольку на точности это не отразится.

Начальная модель ИНС имела следующую архитектуру:

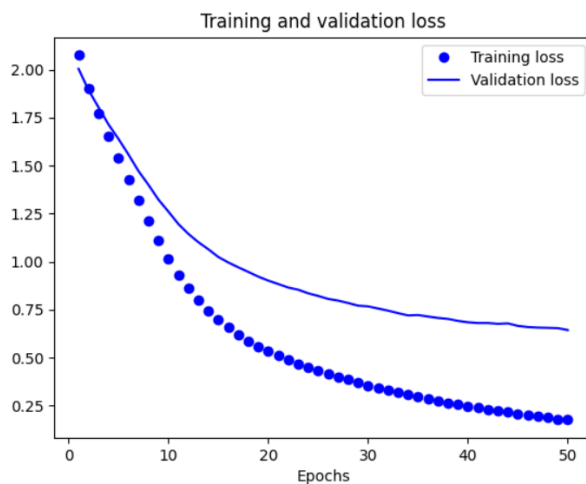
Входной слой из 16 нейронной с функцией активации *ReLU*.

Выходной слой из 7 нейронов с функцией активации *Sigmoid*.

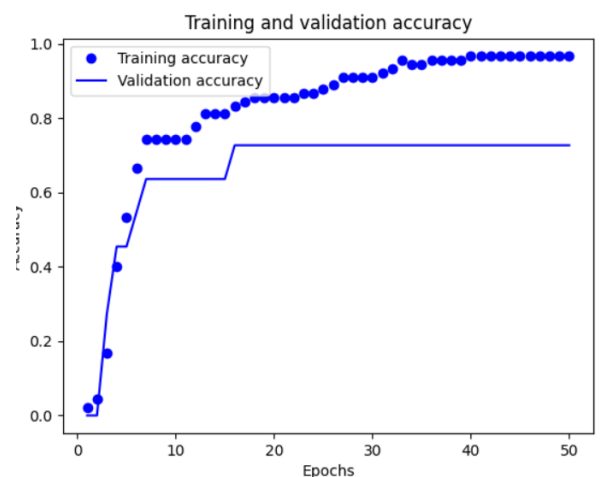
Был выбран оптимизатор *Adam*, функция минимизация потерь *categorical_crossentropy*, и точность выступила *Accuracy*.

Модель была обучена на 50 эпохах с размером партии 5 и размер данных для валидации 0.1.

Результаты ошибки и точности при обучении модели представлены на графике 1.



а



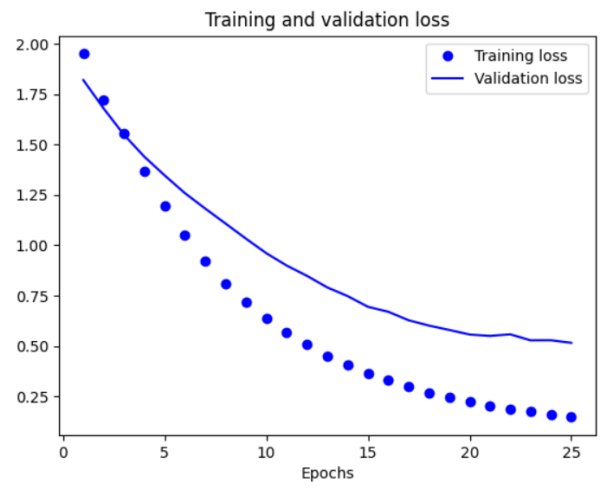
б

График 1 – Значения ошибки (а) и точности (б) первой модели.

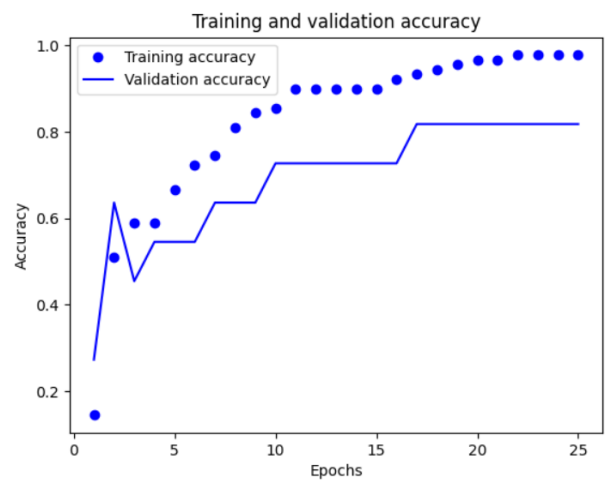
Заметно, что у модели нет проблем с точностью на обучающих данных, но точность валидации недостаточно высокая. Итоговое значение точности составило 0.727.

Добавим в модель скрытый слой из 32 нейронов с функцией активации *ReLU*. И уменьшим количество эпох до 25.

Результаты ошибки и точности при обучении модели представлены на графике 2.



а



б

График 2 – Значения ошибки (а) и точности (б) второй модели.

Итоговая точность составила 0.818.

Выводы.

В ходе выполнения ИДЗ была реализована модель нейронной сети, способная проводить классификацию животных по 16 признакам с точностью 0.818.

Приложение А

Код программы

```
import matplotlib.pyplot as plt
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

# reading data
csv_file = pandas.read_csv('zoo.csv', header=None)
dataset = csv_file.values
X = dataset[:, 1:17].astype(bool)
Y = dataset[:, 17]

# setting categories
encoder = LabelEncoder()
encoder.fit(Y)
encoded_Y = encoder.transform(Y)
dummy_y = to_categorical(encoded_Y)

# building model
model = Sequential()
model.add(Dense(16, input_dim=16, activation='relu'))
model.add(Dense(48, activation='relu'))
model.add(Dense(7, activation='softmax'))

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

# training model
h = model.fit(X, dummy_y, epochs=25, batch_size=5,
validation_split=0.1)
```

```

# plotting training and validation accuracies and losses
h_dict = h.history

loss = h_dict['loss']
accuracy = h_dict['accuracy']
val_loss = h_dict['val_loss']
val_accuracy = h_dict['val_accuracy']

epochs = range(1, len(loss)+1)

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

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.legend()
plt.show()

```