

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 животных в зоопарке

Студент гр. 8383

Мирсков А. А.

Преподаватель

Жангиров Т. Р.

Санкт-Петербург

2021

Цель работы.

Построить модель классификации животных в зоопарке, позволяющую по нескольким сведениям о животном предсказать его тип.

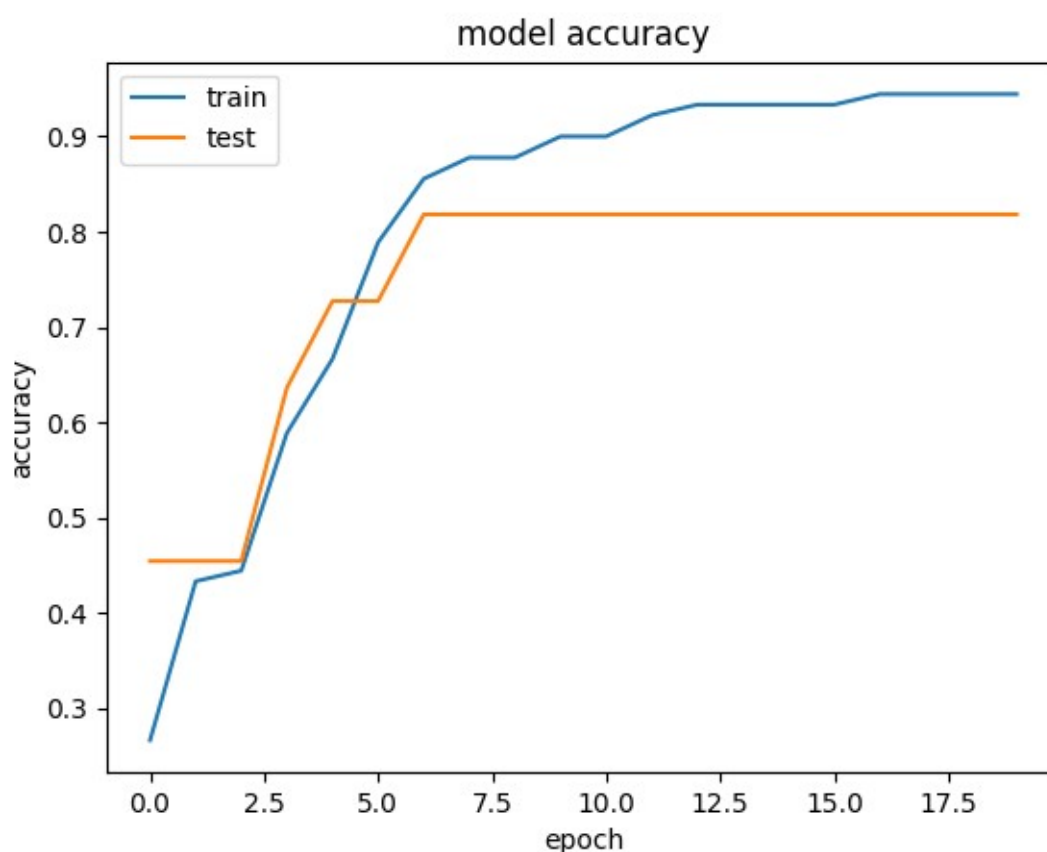
Задание

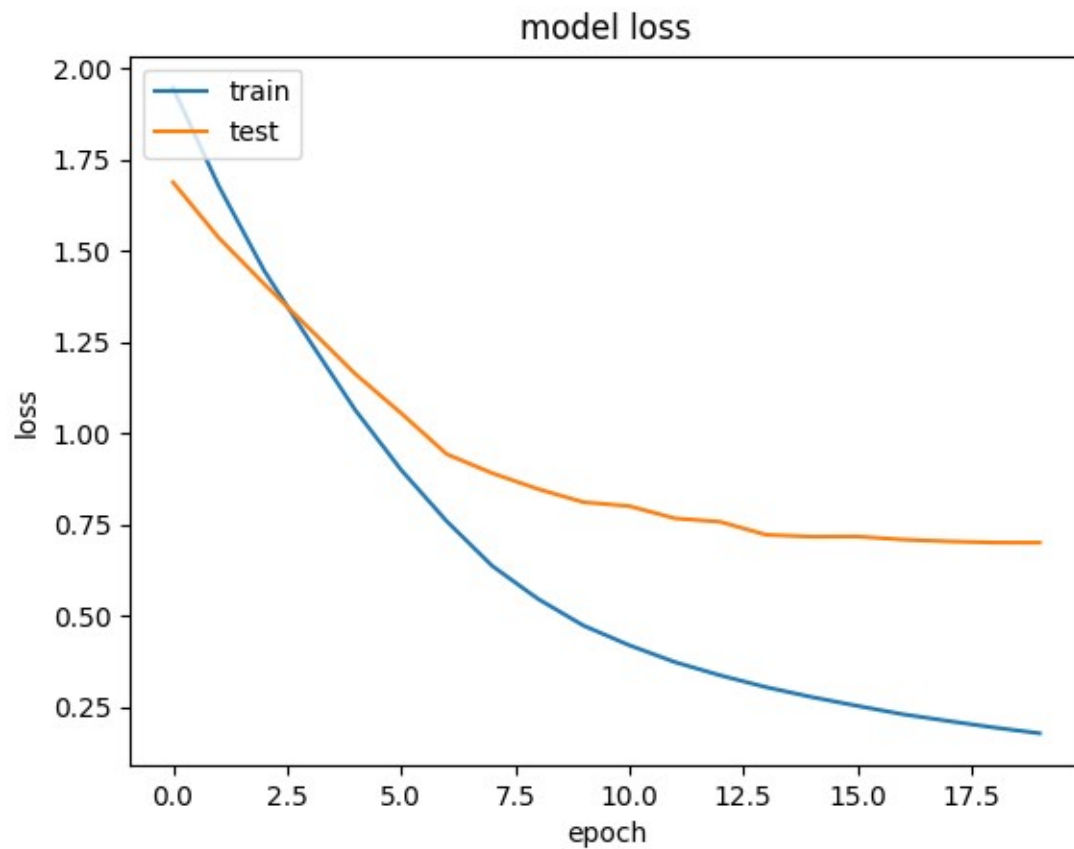
Датасет булевых значений. Включает в себя информацию о 101 животном из зоопарка, которые относятся к 7 классам.

Задача заключается в классификации животного по его описанию.

Выполнение работы.

Был скачан набор данных и написана программа обучения нейронной сети и построения графиков ошибки и точности. Код программы представлен в приложении А. Графики, полученные в результате работы программы представлены ниже.





Модель показала точность 0.8182 на тестовых данных.

Выводы.

В ходе выполнения лабораторной работы была реализована классификация животных в зоопарке по 16 признакам.

Приложение А

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

# Classes
classes = {'aardvark':0, 'antelope':0, 'bear':0, 'boar':0,
          'buffalo':0, 'calf':0,
          'cavy':0, 'cheetah':0, 'deer':0, 'dolphin':0, 'elephant':0,
          'fruitbat':0, 'giraffe':0, 'girl':0, 'goat':0, 'gorilla':0,
          'hamster':0,
          'hare':0, 'leopard':0, 'lion':0, 'lynx':0, 'mink':0,
          'mole':0, 'mongoose':0,
          'opossum':0, 'oryx':0, 'platypus':0, 'polecat':0, 'pony':0,
          'porpoise':0, 'puma':0, 'pussycat':0, 'raccoon':0,
          'reindeer':0,
          'seal':0, 'sealion':0, 'squirrel':0, 'vampire':0, 'vole':0,
          'wallaby':0, 'wolf':0,

          'chicken':1, 'crow':1, 'dove':1, 'duck':1, 'flamingo':1,
          'gull':1, 'hawk':1,
          'kiwi':1, 'lark':1, 'ostrich':1, 'parakeet':1, 'penguin':1,
          'pheasant':1,
          'rhea':1, 'skimmer':1, 'skua':1, 'sparrow':1, 'swan':1,
          'vulture':1, 'wren':1,

          'pitviper':2, 'seasnake':2, 'slowworm':2, 'tortoise':2,
          'tuatara':2,
```

```
'bass':3, 'carp':3, 'catfish':3, 'chub':3, 'dogfish':3,
'haddock':3,
'herring':3, 'pike':3, 'piranha':3, 'seahorse':3, 'sole':3,
'stingray':3, 'tuna':3,

'frog':4, 'frog':4, 'newt':4, 'toad':4,

'flea':5, 'gnat':5, 'honeybee':5, 'housefly':5, 'ladybird':5,
'moth':5, 'termite':5, 'wasp':5,

'clam':6, 'crab':6, 'crayfish':6, 'lobster':6, 'octopus':6,
'scorpion':6, 'seawasp':6, 'slug':6, 'starfish':6, 'worm':6}
```

```
# Loading data
```

```
dataframe = pandas.read_csv("zoo.csv", header=None)
```

```
dataset = dataframe.values
```

```
X = dataset[:,1:].astype(float)
```

```
Y = dataset[:,0]
```

```
Y = [classes[one_class] for one_class in Y]
```

```
print(Y)
```

```
# Text labels to categorical vector
```

```
encoder = LabelEncoder()
```

```
encoder.fit(Y)
```

```
encoded_Y = encoder.transform(Y)
```

```
dummy_y = to_categorical(encoded_Y)
```

```
# Creating model
```

```
model = Sequential()
```

```
model.add(Dense(17, activation='relu'))
```

```
model.add(Dense(34, activation='relu'))
model.add(Dense(7, activation='softmax'))

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

# Training net
history = model.fit(X, dummy_y, epochs=20, batch_size=5,
validation_split=0.1)

# Plotting accuracy
plt.plot(history.history['acc'])
plt.plot(history.history['val_acc'])
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()

# Plotting loss
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
```