МИНОБРНАУКИ РОССИИ

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ

ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)

Кафедра МО ЭВМ

ОТЧЁТ

по лабораторной работе №6

по дисциплине «Искусственные нейронные сети»

Тема: Прогноз успеха фильмов по обзорам

Студент гр.8382	_______________	Фильцин И.В.
Преподаватель	_______________	Жангиров Т.Р.

Санкт-Петербург

2021

Цель работы

Прогноз успеха фильмов по обзорам (Predict Sentiment From Movie Reviews)

Задание

Ознакомиться с задачей классификации

Изучить способы представления текста для передачи в ИНС

Достигнуть точность прогноза не менее 95

Ход работы

Рассмотрим следующую архитектуру сети.

Входной слой - 50 нейронов с функций активации relu.

Скрытые слои - слой Dropout с rate=0.3, слой из 50 нейронов с функцией активации relu, слой Dropout с rate=0.2, слой из 50 нейронов с функцией активации relu.

Выходной слой - 1 нейрон с функцией активации sigmoid.

Проведем обучение сети с оптимизатором Adam, в качестве функции потерь будем использовать бинарную кросс-энтропию. Исследуем работу полученной сети в зависимости от максимальной длины текста.

1. Максимальный размер - 10000.

Имеем $loss : 0.2117 - accuracy : 0.9214 - val\_loss : 0.2623 - val\_accuracy : 0.8949$.

2. Максимальный размер - 15000.

Имеем $loss : 0.2073 - accuracy : 0.9219 - val\_loss : 0.2651 - val\_accuracy : 0.8955$

3. Максимальный размер - 20000.

Имеем $loss : 0.2078 - accuracy : 0.9210 - val\_loss : 0.2649 - val\_accuracy : 0.8944$

Можно заметить, что увеличение максимальной длины текста не сильно изменяет результат работы сети.

Также была написана функция, позволяющая загружать собственный текст для проверки сети. Всего было создано 4 пробных текста:

Текст 1 - 10 звезд, результат работы сети - 0.97

Wow, what a masterpiece!! Really enjoyed **this** movie, full of action, great dialogues **and** perfect acting **not** forgetting the wonderful choice of soundtracks. A very unique movie which has similar taste to the John Wick franchise but more exciting with some family action **and** moments. A little bit **short in length** but worth every scene. Bob did a wonderful job **in this** movie **and** suited him perfectly for the role, wish to see him **in** more action like this. I highly recommend **this** movie to every John Wick fan, action **or** intense **and** realistic fight sequences... With a purchase to the collection

Текст 2 - 1 звезда, результат работы сети - 0.005

The movie was a disappointment for me. I saw the trailer for the movie before I saw the actual movie itself. I thought **this** movie would be more like a vigilante movie **and** I love these sort of movies.

But **in** effect **this** was more of a dumb backstory kind of generic Hollywood movies dished out. **This** follows the same case as John Wick. The first John Wick was good, but they killed it with the sequels. They should have stopped after the first one. "I Am Wrath" is also another good movie **and** better than **this** one.

I have already seen good vigilante movies **in** the past. Death Wish **and** The Exterminator are prime examples of good movies that have survived **into** the present era. Vigilante 1983 starring the late Robert Forster **and** Fighting Back 1982 starring Tom Skerrit are also terrific.

In Nobody(2021), the story follows a family man played by Bob Odenkirk(of Breaking Bad fame) who fails to stop some inept burglars from stealing his possessions. After which he has a change of heart **and** lashes **out** vigilante—style to vent **out** his anger. The rest of the movie deals with repercussions of his decisions.

The acting **in** the movie is so—so. The direction **and** background music **and** sound—effects is typical of a generic Hollywood action thriller. Overall **this** movie was a real waste of my time **and** I would never recommend you to watch it. See all of the above movies I have listed above.

Текст 3 - 5 звезд, результат работы сети - 0.08

It’s a good time while watching but like most movies these days (ok boomer) you dislike more things about it as you think about it. Which is fine I guess.

Do yourself a favor and rent "A History of Violence" it’s the same story but with much deeper exploration of the characters **and** what something like **this** does to them

Текст 4 - 9 звезд, результат работы сети - 0.882

Finally got to see some amazing movie **in** the cinema. To be honest **this** film is filled with lots of terrific moments **and** surprises (**not** twists) with its badass characters, especially the protagonist "Bob Odenkirk" he completely nailed it.

From the start to the **end** of the film was a BANG! ... for all those who love movies like John Wick, Equalizer shouldn’t miss this movie at all ... It worth the money

Таким образом, можно сказать, что сеть выдает верные результаты.

Вывод

В ходе лабораторной работы была реализована сеть, осуществляющая распознавание объектов на фотографиях. Было исследовано использование слоя Dropout и влияние размера ядра свертки на результат.

Приложение А.

Исходный код

```
import matplotlib
import matplotlib.pyplot as plt
import numpy as np
from tensorflow.keras.utils import to_categorical
from tensorflow.keras import models, layers, Sequential
from tensorflow.keras.datasets import imdb

(training_data, training_targets), (testing_data, testing_targets) = imdb.load_data(num_words=10000)
data = np.concatenate((training_data, testing_data), axis=0)
targets = np.concatenate((training_targets, testing_targets), axis=0)

size = 10000

def vectorize(sequences, dimension=10000):
    results = np.zeros((len(sequences), dimension))
    for i, sequence in enumerate(sequences):
        results[i, sequence] = 1
    return results



def load_text(file):
    str = ""
    with open(file, ’r’) as fd:
        str = fd.read()
    result = []
    index = imdb.get_word_index()

    for w in str.split():
        i = index.get(w.lower())
        if i is not None and i < size:
            result.append(i)

    return vectorize([result], size)



data = vectorize(data, size)
targets = np.array(targets).astype("float32")

test_x = data[:10000]
test_y = targets[:10000]
train_x = data[10000:]
```

```
train_y  = targets[10000:]



model = Sequential()
# Input  — Layer
model.add(layers.Dense(50,  activation="relu",    input_shape=(size,   )))
# Hidden — Layers

model.add(layers.Dropout(0.3,   noise_shape=None, seed=None))
model.add(layers.Dense(50,  activation="relu"))
model.add(layers.Dropout(0.2,   noise_shape=None, seed=None))
model.add(layers.Dense(50,  activation="relu"))

# Output— Layer
model.add(layers.Dense(1,  activation="sigmoid"))

model.compile(
 optimizer="adam",
 loss="binary_crossentropy",
 metrics=["accuracy"]
)

results  = model.fit(
 train_x,  train_y,
 epochs=4,
 batch_size=500,
 validation_data=(test_x,    test_y)
)

print(np.mean(results.history["val_accuracy"]))

print(model.predict(load_text("1")))      #  10
print(model.predict(load_text("2")))      #  1
print(model.predict(load_text("3")))      #  5
print(model.predict(load_text("4")))      #  9
```