Министерство цифрового развития, связи и массовых коммуникаций Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Сибирский государственный университет телекоммуникаций и информатики» (СибГУТИ)

Отчёт по лабораторной работе №4

Разработка нейронной сети

Выполнил:

студент гр. ИП-111

Кузьменок Д.В.

Проверил:

Старший преподаватель кафедры ПМиК

Дементьева К.И.

Новосибирск, 2024 г.

**Задание**

Целью данной лабораторной работы является разработка нейронной сети для решения задачи классификации или регрессии в зависимости от набора данных в рамках варианта. Лабораторная работа предполагает разработку на языке программирования Python с использованием библиотеки Keras.

Вариант задания:

3) Определение эмоционального окраса рецензии фильма (IMDB movie review sentiment classification dataset)

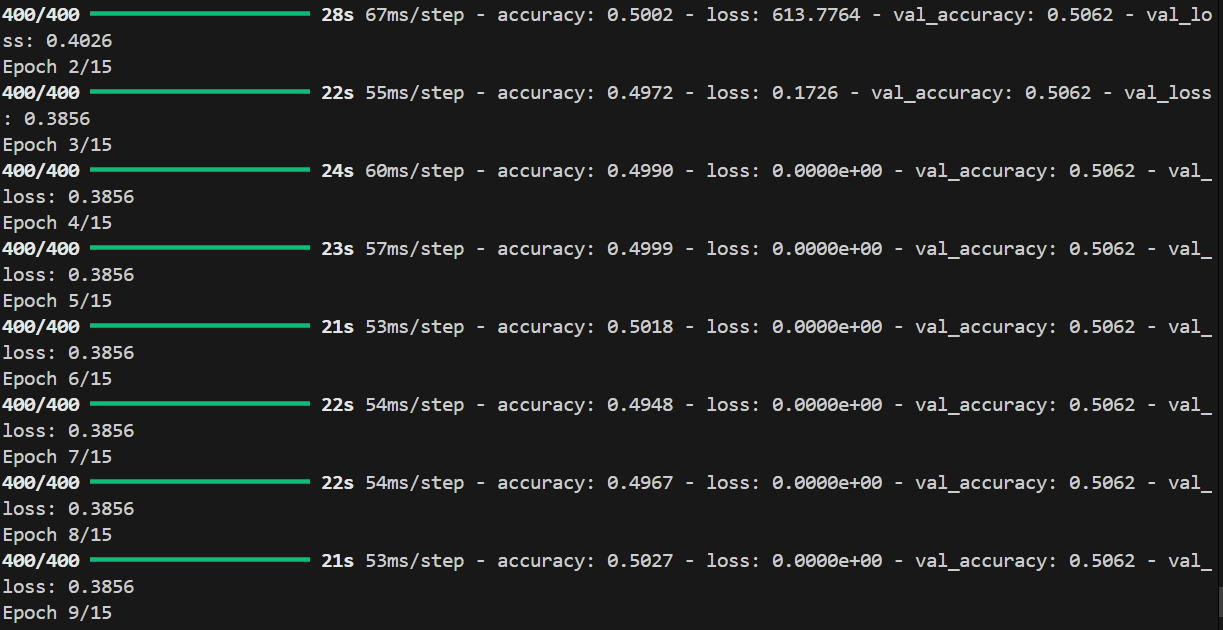
Все наборы данных доступны по ссылке: <https://keras.io/api/datasets/>

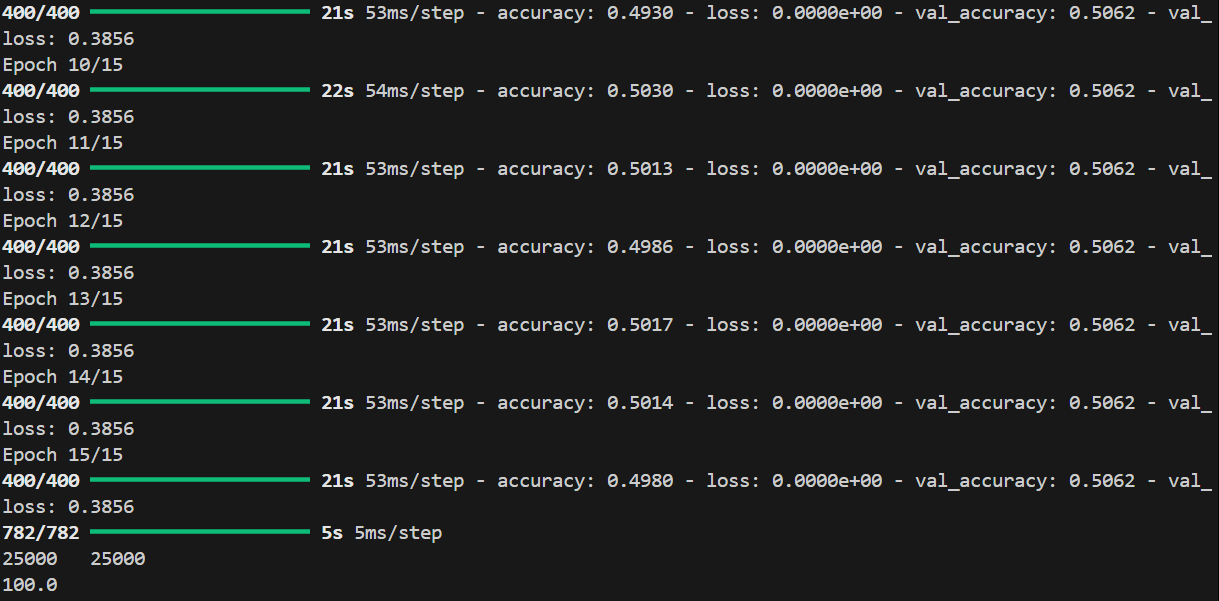
При разработке нейронной сети следует соблюсти наличие необходимых составляющих исходя из следующих вариантов:

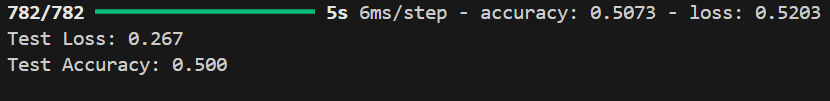
3) Нейросеть должна состоять из пяти полносвязных слоёв, обязательное использование ActivityRegularization, в качестве оптимизатора использовать RMSprop.

**Результаты**

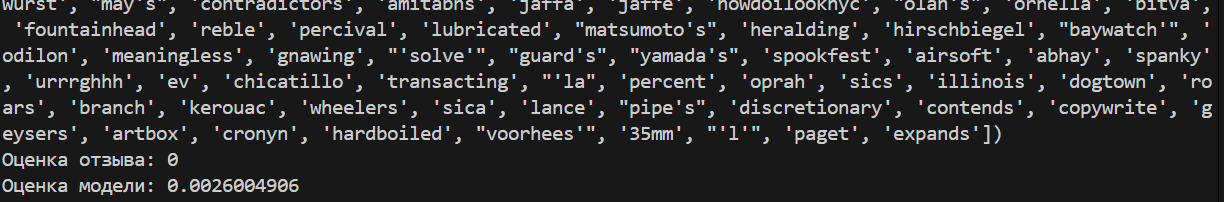
Программа у меня проходит по максимальному количеству слов, которые будут использоваться в модели (500000), с максимальной длиной одного отзыва в 50000 знаков. Такие результаты были получены:







Точность получилась ≈50%. После, чтобы визуализировать работу программы, я вывожу случайный отзыв с оценкой, и той оценкой, которая рассчитала моя модель:



Таким образом, модель максимально близко приблизилась к той оценке, которая была на самом деле, несмотря на общую точность предсказания.

**Код программы**

**import** **tensorflow** **as** **tf**

**from** **tensorflow** **import** keras

**from** **tensorflow.keras** **import** layers

**from** **tensorflow.keras.datasets** **import** imdb

**from** **tensorflow.keras.preprocessing** **import** sequence

**import** **numpy** **as** **np**

max\_words = 500000

max\_len = 50000

(x\_train, y\_train), (x\_test, y\_test) = imdb.load\_data(num\_words=max\_words)

x\_train = sequence.pad\_sequences(x\_train, maxlen=max\_len)

x\_test = sequence.pad\_sequences(x\_test, maxlen=max\_len)

model = keras.Sequential([

layers.Dense(128, activation='relu', input\_shape=(max\_len,)),

layers.ActivityRegularization(l1=0.01),

layers.Dense(64, activation='relu'),

layers.ActivityRegularization(l1=0.01),

layers.Dense(32, activation='relu'),

layers.ActivityRegularization(l1=0.01),

layers.Dense(16, activation='relu'),

layers.ActivityRegularization(l1=0.01),

layers.Dense(1, activation='sigmoid')

])

model.compile(optimizer=keras.optimizers.RMSprop(), loss='categorical\_crossentropy', metrics=['accuracy'])

history = model.fit(x\_train, y\_train, epochs=15, batch\_size=50, validation\_split=0.2)

L = len(y\_test)

correct = 0

YP = model.predict(x\_test)

**for** i **in** range(L):

y1 = np.argmax(y\_test[i])

ypred = np.argmax(YP[i])

**if** ypred == y1:

correct += 1

print(correct, ' ', L)

print(correct/L\*100)

test\_loss, test\_acc = model.evaluate(x\_test, y\_test)

print(f'Test Loss: {test\_loss:.3f}')

print(f'Test Accuracy: {test\_acc:.3f}')

random\_index = np.random.randint(0, len(x\_test))

random\_review = x\_test[random\_index]

predicted\_rating = model.predict(np.array([random\_review]))

print("Случайный отзыв:", imdb.get\_word\_index().keys())

print("Оценка отзыва:", y\_test[random\_index])

print("Оценка модели:", predicted\_rating[0][0])