Лабораторна робота 4. Кисляк Марини

Датасет: IMDB

Для виконання даної роботи використовувся Google Colab.

1) Завантажте датасет, підготуйте його для входу в нейронну мережу.

Набір даних IMDB складається з 50 000 відгуків до фільмів від користувачів IMDB, які позначені як позитивні (1) або негативні (0). Відгуки попередньо обробляються, і кожен з них кодується як послідовність покажчиків слів у вигляді цілих чисел. Слова індексуються за загальною частотою у наборі даних. Наприклад, ціле число “4” кодує четверте за частотою слово в даних. 50 000 оглядів поділяються на 25 000 для навчання та 25 000 для тестування.

Оскільки ми хочемо уникнути розподілу даних для тренування та тесту 50/50, ми об’єднаємо дані, щоб потім зробити розділення 80/20.

Весь набір даних містить 88584 унікальних слів, а середня довжина відгуку становить 234 слова зі стандартним відхиленням у 173 слова.

Подивимось на один з відгуків.

```
Label: 1
[1, 48, 874, 126, 110, 911, 1093, 14, 9, 4, 172, 243, 7, 2542, 5057, 2689,
405, 7, 22, 13, 423, 4, 206, 836, 4, 550, 1430, 7, 12, 32, 13, 43, 440,
36, 92, 81, 8, 14, 31, 51, 36, 122, 8, 911, 1093, 19, 15, 755, 21, 3847,
4, 31, 155, 13, 161, 40, 16, 4, 616, 1155, 799, 13, 16, 6081, 18, 4, 8087,
4, 226, 58, 22, 517, 46, 184, 76, 40, 12, 144, 362, 2, 56, 18, 6, 1189,
36, 3485, 6, 3619, 37, 6970, 317, 457, 7400, 159, 36, 188, 50, 38, 13,
482, 27, 3217, 304, 4, 107, 2481, 5, 4, 1432, 46, 23, 4, 1093, 150, 330,
25, 14, 9, 4, 333, 251, 7, 68, 1154, 7338, 195, 4, 86, 251, 36, 1084, 33,
482, 7660, 6, 4660, 3886, 38, 36, 26, 46, 50, 5, 32, 9, 52, 18, 6, 117,
137, 95, 4040, 4660, 58, 55, 1596, 25, 124, 142, 11, 14, 20, 25, 92, 67,
4, 4660, 6, 226, 2304, 7, 6, 176, 21, 54, 25, 81, 7534, 12, 9, 55, 629,
13, 119, 4, 2, 2, 5, 523, 5, 4, 2, 7, 5382, 15, 186, 8, 4, 8087, 175, 58,
29, 5245, 27, 419, 752, 4, 1093, 55, 55, 948, 21, 52, 444, 14, 9, 6, 87,
22, 48, 25, 70, 79, 501, 4, 616, 117, 799]
```

```
# if you've ever seen open water this is the same kind of gritty edgy
indie style of film i liked the action suspense the slow building of it
all i just hope they don't do to this one what they did to open water with
that sequel but anyways the one thing i didn't like was the annoying
younger sister i was rooting for the croc the whole time film starts out
pretty much like it should kids # up for a trip they hire a guide who
mysteriously left 5 mins before they got there so i guess his assistant
takes the two sisters and the boyfriend out on the water now mind you this
is the second day of their adventure curiously enough the first day they
spent at guess hmm a crocodile farm so they are out there and all is good
for a little while then bang crocodile time very intense you know
```

```
something in this movie you don't see the crocodile a whole heck of a lot
but when you do gosh it is very scary i love the # # and eyes and the # of
fog that seems to the croc every time he raises his head above the water
very very creepy but good overall this is a great film if you can get past
the annoying little sister
```

2) Будуємо рекурентну нейронну мережу та перевіряємо якість її роботи.

Ми також обираємо binary-crossentropy як втрату (оскільки ми маємо справу з бінарною класифікацією) і точність як нашу метрику оцінки.

```
Epoch 1/2
80/80 [==============================] - 1s 17ms/step - loss: 0.4109 -
accuracy: 0.8150 - val_loss: 0.2705 - val_accuracy: 0.8921
Epoch 2/2
80/80 [==============================] - 1s 14ms/step - loss: 0.2172 -
accuracy: 0.9172 - val_loss: 0.2598 - val_accuracy: 0.8941
```

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

0.8930999934673309

Бачимо, що в даному випадку немає перенавчання. Спробуємо збільшити кількість Epoch для покращення результату.

```
Epoch 1/10
40/40 [==============================] - 1s 29ms/step - loss: 0.0655 -
accuracy: 0.9784 - val_loss: 0.4130 - val_accuracy: 0.8841
Epoch 2/10
40/40 [==============================] - 1s 26ms/step - loss: 0.0451 -
accuracy: 0.9861 - val_loss: 0.4648 - val_accuracy: 0.8845
Epoch 3/10
40/40 [==============================] - 1s 26ms/step - loss: 0.0339 -
accuracy: 0.9892 - val_loss: 0.5183 - val_accuracy: 0.8823
Epoch 4/10
40/40 [==============================] - 1s 25ms/step - loss: 0.0281 -
accuracy: 0.9910 - val_loss: 0.5837 - val_accuracy: 0.8825
Epoch 5/10
40/40 [==============================] - 1s 26ms/step - loss: 0.0244 -
accuracy: 0.9918 - val_loss: 0.5715 - val_accuracy: 0.8828
Epoch 6/10
40/40 [==============================] - 1s 26ms/step - loss: 0.0213 -
accuracy: 0.9932 - val_loss: 0.6294 - val_accuracy: 0.8813
Epoch 7/10
40/40 [==============================] - 1s 26ms/step - loss: 0.0229 -
accuracy: 0.9916 - val_loss: 0.6562 - val_accuracy: 0.8791
Epoch 8/10
40/40 [==============================] - 1s 26ms/step - loss: 0.0235 -
accuracy: 0.9915 - val_loss: 0.6405 - val_accuracy: 0.8762
Epoch 9/10
40/40 [==============================] - 1s 25ms/step - loss: 0.0193 -
accuracy: 0.9931 - val_loss: 0.6853 - val_accuracy: 0.8763
Epoch 10/10
40/40 [==============================] - 1s 26ms/step - loss: 0.0165 -
accuracy: 0.9937 - val_loss: 0.6861 - val_accuracy: 0.8775
```

accuracy: 0.8806600034236908

Як бачимо, після зміни параметрів epochs та batch_size якість роботи моделі особливо не змінилася.

Спробуємо покращити результати додавши LSTM.

```
Train...
Epoch 1/2
782/782 [==============================] - 269s 344ms/step - loss: 0.4271
- accuracy: 0.8047 - val_loss: 0.3663 - val_accuracy: 0.88523
Epoch 2/2
782/782 [==============================] - 260s 332ms/step - loss: 0.1634
- accuracy: 0.8935 - val_loss: 0.3776 - val_accuracy: 0.9013
Test accuracy: 0.8982800221443176
```

Отже, максимальна точність, яку вдалося досягти на даному датасеті = 0.898. Загалом можна сказати, що перенавчання не було - точність на тестовій та тренувальній вибірці не відрізнялась при різних запусках. При зміні параметрів batch_size та epochs не вдалося значно покращити роботу моделі. Хоча збільшення batch_size, як правило, призводить до швидшого навчання. Менший batch_size повільніший у навчанні, але він може покращувати точність швидше. Це безумовно залежить від задачі, і в нашій задачі оптимальний batch_size був вибраний = 500. Також до моделі можна додавати LSTM, але в нашому випадку значного покращення не відбулося.