



Міністерство освіти і науки України  
Національний технічний університет України  
“Київський політехнічний інститут імені Ігоря Сікорського”  
Факультет інформатики та обчислювальної техніки  
Кафедра інформатики та програмної інженерії

**Лабораторна робота №7**  
**Проектування та реалізація програмних систем з**  
**нейронними мережами**

Київ 2023

**Завдання:**

Написати програму, що реалізує рекурентну нейронну мережу LSTM для розпізнавання емоційного забарвлення тексту, використати датасет Yelp Dataset.

## Хід роботи:

Для цієї лабораторної роботи було обрано Yelp датасет з пакету tensorflow\_datasets.

Зчитуємо дані з датасету:

```
1 data = tfds.load("yelp_polarity_reviews")
2 trainData = data['train']
3 testData = data['test']
4 train = []
5 test = []
6 for dataPoint in data['train']:
7     train.append(dataPoint)
8
9 for dataPoint in data['test']:
10    test.append(dataPoint)
11
12 trainX = []
13 trainY = []
14 for dataPoint in train:
15     trainX.append(dataPoint['text'].numpy().decode())
16     trainY.append(int(dataPoint['label'].numpy()))
17
18 testX = []
19 testY = []
20 for dataPoint in test:
21     testX.append(dataPoint['text'].numpy().decode())
22     testY.append(int(dataPoint['label'].numpy()))
23
24 print(len(trainX))
25 print(trainX[0])
```

Датасет містить ~560000 записів різних текстів.

Після зчитуванн нормалізуємо дані:

```
1 vocabularySize = 20000
2 length = 200
3 tok = Tokenizer(num_words = vocabularySize)
4 tok.fit_on_texts(trainX)
5 trainTokenized = tok.texts_to_sequences(trainX)
6 testTokenized = tok.texts_to_sequences(testX)
7 trainTokenized = pad_sequences(trainTokenized, maxlen=length)
8 testTokenized = pad_sequences(testTokenized, maxlen=length)
9 trainY = np.array(trainY)
10 testY = np.array(testY)
```

Модель виглядає наступним чином:

```

1 model = Sequential([Embedding(vocabularySize, 16),
2                       LSTM(32), Dense(64, activation='relu'),
3                       Dense(1, activation='sigmoid')])
4 model.compile(loss = 'binary_crossentropy', optimizer = 'RMSprop', metrics = ['accuracy'])
5 model.summary()

```

```

└─ lstm (LSTM)                (None, 32)                6272

    dense (Dense)              (None, 64)                2112

        dense_1 (Dense)        (None, 1)                  65

    =====
    Total params: 328,449
    Trainable params: 328,449
    Non-trainable params: 0
    =====

```

Тренуємо модель на 3-ти епохах. Пікова точність склала також приблизно 95%!

```

1 history = model.fit(trainTokenized, trainY, epochs=3, validation_data=(testTokenized, testY))

└─ 0.1355 - val_accuracy: 0.9498
   Epoch 3/3
   17500/17500 [=====] - 811s 46ms/step - loss: 0.1256 - accuracy: 0.9535 - val_loss:
   0.1251 - val_accuracy: 0.9543

```

## Prediction

```

1 model.evaluate(test_x, test_y)
2 preds = model.predict(test_x)
3 for i in range(len(preds)):
4     print("Pred:", np.argmax(preds[i]), ";", " Real: ", np.argmax(test_y[i]))

```

```

└─ Pred: 1 ; True: 1
   Pred: 0 ; True: 0
   Pred: 1 ; True: 1
   Pred: 2 ; True: 8
   Pred: 8 ; True: 3
   Pred: 0 ; True: 11
   Pred: 3 ; True: 3
   Pred: 3 ; True: 3
   Pred: 11 ; True: 11
   Pred: 0 ; True: 11
   Pred: 4 ; True: 4

```

Підготуємо тексти з різним емоційним забарвленням:

```
1 texts = ["I woke up to the golden rays of the sun gently streaming through my window, filling my room with warmth and  
hope. As I stepped outside, the crisp morning air kissed my cheeks, invigorating my senses. Birds sang a melodious  
symphony, their songs dancing in harmony with the swaying trees. With a heart full of gratitude, I embarked on a new  
day, knowing that endless possibilities awaited me. Life felt like a beautiful tapestry woven with love and joy, and I  
couldn't help but smile, ready to embrace every moment with open arms.",  
2 "The room felt suffocating, its walls closing in on me, amplifying the weight of despair in my heart. A thick  
fog of gloom settled around me, clouding my thoughts and suffusing my every breath. The echoes of shattered  
dreams reverberated through the silence, reminding me of what was lost. Each step felt like wading through  
quicksand, the burden of sorrow dragging me deeper into the abyss. The world seemed devoid of color, and even  
the simplest tasks felt like insurmountable mountains to climb. The heaviness of sadness weighed me down,  
leaving me longing for a glimmer of light in the darkness.",  
3 "I sat by the window, watching as raindrops rhythmically tapped against the glass. The gray sky mirrored the  
calmness in my mind, offering a moment of tranquility amidst the chaos of life. Thoughts drifted lazily, like  
leaves carried by a gentle breeze, as I sipped on a warm cup of tea. The world outside continued its bustling  
rhythm, people hurrying through the streets, umbrellas shielding them from the downpour. It was a day without  
extremes, where the mundane and ordinary held their own quiet beauty. Time passed by unhurriedly, and I found  
solace in the simplicity of the present moment."]
```

Намагаємося передбачити емоційне забарвлення тексту:

```
1 for text in texts:  
2     print(text)  
3     text = pad_sequences(tok.texts_to_sequences([text]))  
4     prediction = model.predict(text, verbose=0)  
5     print(prediction)  
6     print(history.history['accuracy'])  
7     plt.figure(figsize=(10, 10))  
8     plt.plot(history.history['accuracy'], label='train accuracy')  
9     plt.plot(history.history['val_accuracy'], label='validation accuracy')  
10    plt.xlabel("Epoch")  
11    plt.ylabel("Accuracy")  
12    plt.legend()  
13    plt.show()  
14    history.history
```

В кінці отримуємо такі результати:

1. Перший текст отримав значення 0.9955267, що свідчить, що він позитивно забарвлений.
2. Другий текст отримав значення 0.04258246, що свідчить, що він негативно забарвлений.
3. Третій текст отримав значення 0.8615486, що свідчить про його позитивне забарвлення, але меншою мірою ніж перший текст.

**Висновок:**

Отже, по ходу цієї лабораторної роботи я познайомився з новою рекурентною нейронною мережею LSTM та використав її для визначення емоційного забарвлення в тексті.