

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»

ЗВІТ

з лабораторної роботи №7

з навчальної дисципліни «Проектування та реалізація програмних систем з  
нейронними мережами»

Виконав:  
студент групи ІІІ-15  
Мешков Андрій Ігорович

Перевірив:  
Шимкович В.М.

Київ 2024

## ЛАБОРАТОРНА РОБОТА №7

**Тема:** Рекурентні нейронні мережі LSTM

**Завдання** – Написати програму, що реалізує рекурентну нейронну мережу LSTM для розпізнавання емоційного забарвлення тексту, використати датасет [Yelp Dataset](#).

## Завантаження необхідних бібліотек

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import tensorflow as tf
import tensorflow_datasets as tfds
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Embedding, LSTM, Dense, Dropout,
Bidirectional
from sklearn.model_selection import train_test_split
```

## Завантаження датасету Yelp

```
data = tfds.load('yelp_polarity_reviews', split='train',
shuffle_files=True)
```

## Розділення даних на відгуки та емоції

```
reviews = []
emotions = []
```

```
for i in data.take(20000):
    reviews.append((i['text'].numpy().decode("utf-8")))
    emotions.append(int(i['label']))
```

```
2024-05-11 19:06:59.519285: W
tensorflow/core/framework/local_rendezvous.cc:404] Local rendezvous is
aborting with status: OUT_OF_RANGE: End of sequence
```

## Демонстрація тексту

```
reviews[0]
```

```
'This place is a great lunch spot for vegans, vegetarians and healthy
folk alike. The variety of prepared teas is incredible and they offer
most of their teas loose to take home and brew as well. As a bad
vegetarian that eats fish on occasion, this is a favorite place of
```

```
mine to enjoy a purely vegetarian meal packed with fresh flavors. \\
n\\n My absolute favorite is a wrap and cup of soup. Usually I opt for
the Black Bean Sunflower Wrap, which has black bean sunflower hummus,
with mixed greens, jack cheese, red onions, carrot, celery, cucumber,
bell pepper,\\nheirloom tomato, and spicy chipotle salsa. I have
discovered that no matter what their soup of the day is, it is always
perfect! Last time the watermelon gazpacho was perfect for the warm
weather and today the french onion and mushroom soup hit the spot. The
staff is always helpful and seem to know the menu very well. Thank you
Chakra 4 for filling up my belly with healthy goodness!'
```

## Відредагуємо текст від символів, які заважають тренуванню.

```
new_reviews = []
for review in reviews:
    review = review.replace("\\n", " ")
    review = review.replace("\\\"", "'")
    review = review.replace("\\r", " ")
    review = review.replace("\\\"\"", " ")
    new_reviews.append(review)
reviews = new_reviews
```

```
reviews[0]
```

```
'This place is a great lunch spot for vegans, vegetarians and healthy
folk alike. The variety of prepared teas is incredible and they offer
most of their teas loose to take home and brew as well. As a bad
vegetarian that eats fish on occasion, this is a favorite place of
mine to enjoy a purely vegetarian meal packed with fresh flavors.
My absolute favorite is a wrap and cup of soup. Usually I opt for the
Black Bean Sunflower Wrap, which has black bean sunflower hummus, with
mixed greens, jack cheese, red onions, carrot, celery, cucumber, bell
pepper, heirloom tomato, and spicy chipotle salsa. I have discovered
that no matter what their soup of the day is, it is always perfect!
Last time the watermelon gazpacho was perfect for the warm weather and
today the french onion and mushroom soup hit the spot. The staff is
always helpful and seem to know the menu very well. Thank you Chakra 4
for filling up my belly with healthy goodness!'
```

## Токенізація тексту

```
tokenizer = Tokenizer(num_words=10000, oov_token='<OOV>')
tokenizer.fit_on_texts(reviews)
sequences = tokenizer.texts_to_sequences(reviews)
```

## Подання даних у вигляді послідовностей фіксованої довжини

```
max_length = 200
padded_sequences = pad_sequences(sequences, maxlen=max_length,
padding='post', truncating='post')
```

## Поділ даних на тренувальний та тестувальний набори

```
X_train, X_test, y_train, y_test = train_test_split(padded_sequences,
emotions, test_size=0.2, random_state=42)
```

## Побудова моделі LSTM

```
model = Sequential([
    Embedding(input_dim=10000, output_dim=64,
input_length=max_length),
    Bidirectional(LSTM(20, return_sequences=True)),
    Bidirectional(LSTM(20)),
    Dense(1, activation='sigmoid'),
])

/Users/andrey/Documents/Document_study/Year 3.2/ПЗПраPHC/Lab
7/src/.venv/lib/python3.9/site-packages/keras/src/layers/core/embeddin
g.py:90: UserWarning: Argument `input_length` is deprecated. Just
remove it.
  warnings.warn(
```

## Компіляція моделі

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

```
model.summary()
```

```
Model: "sequential_1"
```

Layer (type)	Output Shape
Param #	

embedding_1 (Embedding)	?	0
(unbuilt)		
bidirectional_2 (Bidirectional)	?	0
(unbuilt)		
bidirectional_3 (Bidirectional)	?	0
(unbuilt)		
dense_1 (Dense)	?	0
(unbuilt)		
Total params: 0 (0.00 B)		
Trainable params: 0 (0.00 B)		
Non-trainable params: 0 (0.00 B)		

## Підгонка моделі

```
history = model.fit(np.array(X_train), np.array(y_train), epochs=10,
validation_data=(np.array(X_test), np.array(y_test)), verbose=2)
```

Epoch 1/10

500/500 - 56s - 112ms/step - accuracy: 0.8126 - loss: 0.4065 -  
val\_accuracy: 0.8860 - val\_loss: 0.2881

Epoch 2/10

500/500 - 60s - 120ms/step - accuracy: 0.9252 - loss: 0.2137 -  
val\_accuracy: 0.8790 - val\_loss: 0.2916

Epoch 3/10

500/500 - 55s - 110ms/step - accuracy: 0.9491 - loss: 0.1487 -  
val\_accuracy: 0.8820 - val\_loss: 0.3305

Epoch 4/10

500/500 - 53s - 105ms/step - accuracy: 0.9660 - loss: 0.1027 -  
val\_accuracy: 0.8888 - val\_loss: 0.3424

Epoch 5/10

500/500 - 52s - 104ms/step - accuracy: 0.9706 - loss: 0.0894 -  
val\_accuracy: 0.8788 - val\_loss: 0.4051

Epoch 6/10

500/500 - 53s - 105ms/step - accuracy: 0.9807 - loss: 0.0648 -  
val\_accuracy: 0.8900 - val\_loss: 0.3802

Epoch 7/10

```
500/500 - 51s - 103ms/step - accuracy: 0.9811 - loss: 0.0618 -  
val_accuracy: 0.8820 - val_loss: 0.4695  
Epoch 8/10  
500/500 - 54s - 108ms/step - accuracy: 0.9800 - loss: 0.0619 -  
val_accuracy: 0.8873 - val_loss: 0.4262  
Epoch 9/10  
500/500 - 52s - 105ms/step - accuracy: 0.9883 - loss: 0.0384 -  
val_accuracy: 0.8813 - val_loss: 0.4483  
Epoch 10/10  
500/500 - 51s - 103ms/step - accuracy: 0.9901 - loss: 0.0341 -  
val_accuracy: 0.8805 - val_loss: 0.5081
```

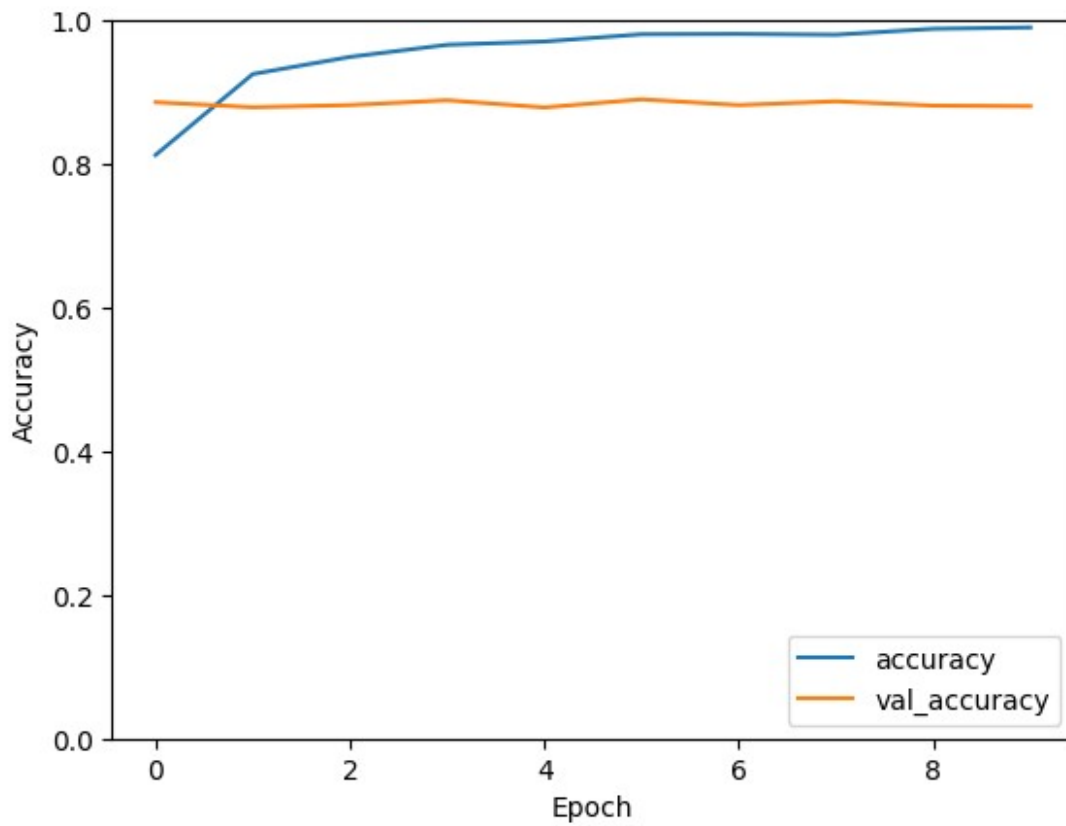
## Оцінка моделі на тестовому наборі даних

```
loss, accuracy = model.evaluate(np.array(X_train), np.array(y_train))  
print(f'Test Accuracy: {accuracy}')  
print(f'Test Loss: {loss}')
```

500/500 ————— 17s 34ms/step - accuracy: 0.9927 - loss:  
0.0253  
Test Accuracy: 0.9921249747276306  
Test Loss: 0.02660629339516163

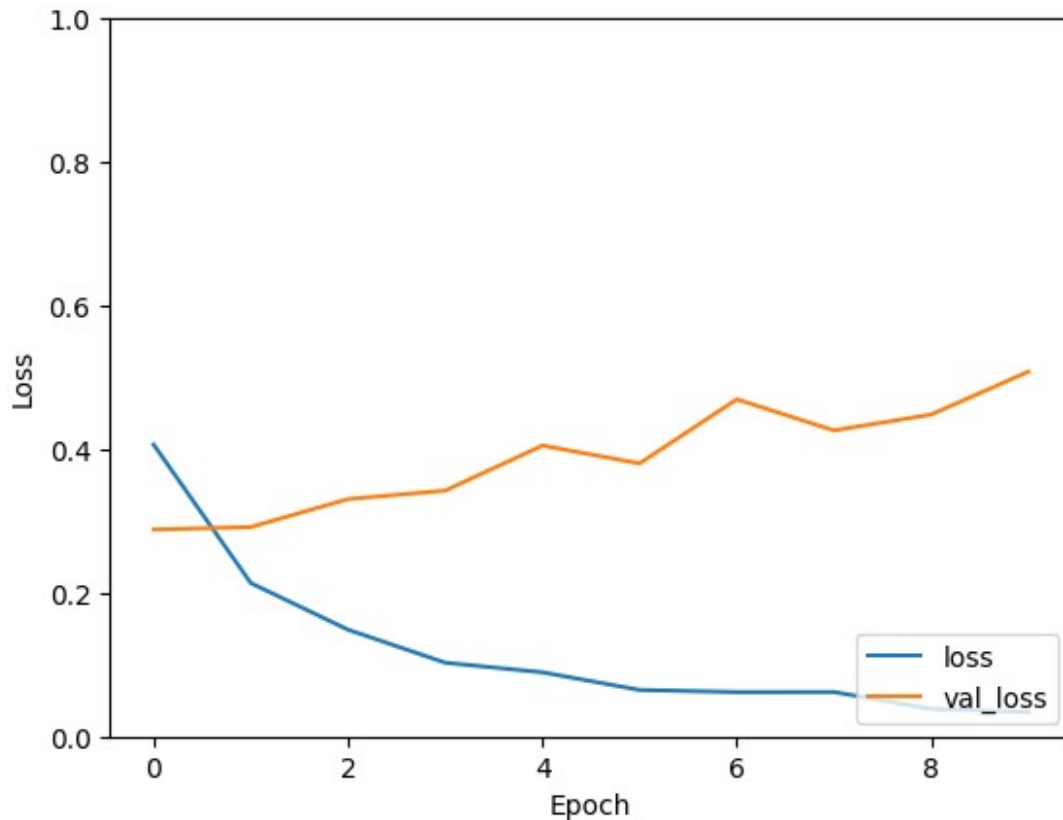
## Візуалізація точності та втрат під час тренування

```
plt.plot(history.history['accuracy'], label='accuracy')  
plt.plot(history.history['val_accuracy'], label='val_accuracy')  
plt.xlabel('Epoch')  
plt.ylabel('Accuracy')  
plt.ylim([0, 1])  
plt.legend(loc='lower right')  
plt.show()
```



```
plt.plot(history.history['loss'], label='loss')
plt.plot(history.history['val_loss'], label='val_loss')
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.ylim([0, 1])
plt.legend(loc='lower right')
plt.show()
```





```
# Прогнозування на тестовому наборі
predictions = model.predict(X_test)

# Визначення класу емоції (позитивна чи негативна) на основі ймовірності
threshold = 0.5
predicted_classes = [1 if pred > threshold else 0 for pred in predictions]

# Виведення кількості правильних та неправильних прогнозів
correct_predictions = 0

for i in range(len(predicted_classes)):
    if y_test[0] == predicted_classes[0]:
        correct_predictions += 1

total_predictions = len(y_test)
print(f'Correct predictions: {correct_predictions}/{total_predictions}')

# Приклад виведення перших 10 прогнозів та їх справжніх значень
for i in range(10):
    print(f'Predicted Emotion: {predicted_classes[i]} | True Emotion: {y_test[i]}')
```

```
{y_test[i]}')  
    print()
```

125/125 ————— 3s 25ms/step

Correct predictions: 4000/4000

Predicted Emotion: 0 | True Emotion: 0

Predicted Emotion: 0 | True Emotion: 0

Predicted Emotion: 0 | True Emotion: 0

Predicted Emotion: 0 | True Emotion: 0

Predicted Emotion: 0 | True Emotion: 0

Predicted Emotion: 0 | True Emotion: 0

Predicted Emotion: 1 | True Emotion: 1

Predicted Emotion: 0 | True Emotion: 0

Predicted Emotion: 1 | True Emotion: 1

Predicted Emotion: 0 | True Emotion: 0

## **Висновок:**

Під час виконання лабораторної роботи було реалізовано рекурентну нейронну мережу LSTM для розпізнавання емоційного забарвлення тексту.