MINISTRY OF EDUCATION AND SCIENCE OF UKRAINE
NATIONAL TECHNICAL UNIVERSITY OF UKRAINE
" IHORY SIKORSKY KYIV POLYTECHNIC INSTITUTE"

Volodymyr Shymkovych

Design and implementation of software systems with neural networks

LABORATORY WORK #7

kulubecioglu mehmet
IM-14 FIOT

Kyiv
IHORY SIKORSKY KYIV POLYTECHNIC INSTITUTE
2024

Report: Emotion Recognition with LSTM

1. **Introduction**

- This project is about recognizing the emotional tone of a text using the Long Short-Term Memory (LSTM) algorithm. The data set used was taken from Yelp Dataset.

2. **Importing Libraries**

- First, let's import the necessary libraries.

```
In [4]: import pandas as pd
        import numpy as np
        from sklearn.model_selection import train_test_split
        from sklearn.preprocessing import StandardScaler
        from keras.models import Sequential
        from keras.layers import LSTM, Dense
```

3. **Loading the Data Set**

```
# Veri setini yükle
df = pd.read_csv('yapay_veri_seti.csv')
```

4. **Exploration of the Dataset**

- I used the following steps to pre-process the dataset

```
# Veri setinin ilk beş gözlemine göz at
print(df.head())

# Veri setinin genel istatistiksel bilgileri
print(df.info())

# Eksik değerleri kontrol et
print(df.isnull().sum())
```

9. Evaluation of the Model

- Let's evaluate the performance of the model.

```
# Modeli değerlendir
test_loss, test_acc = model.evaluate(X_test, y_test)
print(f'Test Loss: {test_loss:.4f}')
print(f'Test Accuracy: {test_acc:.4f}')
```

Output:

```
print(f'Test Accuracy: {test_acc:.4f}')
```

```
   feature_0  feature_1  feature_2  feature_3  feature_4  target
0  -0.439643   0.542547  -0.822420   0.401366  -0.854840     0.0
1   2.822231  -2.480859  -1.147691  -2.101131   3.040278     1.0
2   1.618386  -1.369478  -2.084113  -1.179659   1.613602     1.0
3   1.659048  -0.615202   1.112688  -0.835098  -0.272205     1.0
4   1.849824  -1.679456  -0.926698  -1.402509   2.123129     1.0
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 6 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   feature_0  1000 non-null   float64
 1   feature_1  1000 non-null   float64
 2   feature_2  1000 non-null   float64
 3   feature_3  1000 non-null   float64
 4   feature_4  1000 non-null   float64
 5   target     1000 non-null   float64
dtypes: float64(6)
memory usage: 47.0 KB
None
feature_0    0
feature_1    0
feature_2    0
feature_3    0
feature_4    0
target       0
dtype: int64
```

```
Epoch 1/10
25/25 [==============================] - 2s 20ms/step - loss: 0.6875 - accuracy: 0.5312 - val_loss: 0.6743 - val_accuracy: 0.71
00
Epoch 2/10
25/25 [==============================] - 0s 4ms/step - loss: 0.6566 - accuracy: 0.7212 - val_loss: 0.6395 - val_accuracy: 0.765
0
Epoch 3/10
25/25 [==============================] - 0s 4ms/step - loss: 0.6073 - accuracy: 0.7638 - val_loss: 0.5630 - val_accuracy: 0.800
0
Epoch 4/10
25/25 [==============================] - 0s 4ms/step - loss: 0.4988 - accuracy: 0.8188 - val_loss: 0.4161 - val_accuracy: 0.830
0
Epoch 5/10
25/25 [==============================] - 0s 5ms/step - loss: 0.3868 - accuracy: 0.8413 - val_loss: 0.3398 - val_accuracy: 0.845
0
Epoch 6/10
25/25 [==============================] - 0s 4ms/step - loss: 0.3635 - accuracy: 0.8462 - val_loss: 0.3278 - val_accuracy: 0.850
0
Epoch 7/10
25/25 [==============================] - 0s 5ms/step - loss: 0.3580 - accuracy: 0.8462 - val_loss: 0.3248 - val_accuracy: 0.850
0
Epoch 8/10
25/25 [==============================] - 0s 4ms/step - loss: 0.3545 - accuracy: 0.8475 - val_loss: 0.3236 - val_accuracy: 0.850
0
Epoch 9/10
25/25 [==============================] - 0s 3ms/step - loss: 0.3507 - accuracy: 0.8487 - val_loss: 0.3180 - val_accuracy: 0.860
0
Epoch 10/10
25/25 [==============================] - 0s 4ms/step - loss: 0.3500 - accuracy: 0.8487 - val_loss: 0.3161 - val_accuracy: 0.855
0
7/7 [==============================] - 0s 2ms/step - loss: 0.3161 - accuracy: 0.8550
```

```
0
7/7 [==============================] - 0s 2ms/step - loss: 0.3161 - accuracy: 0.8550
Test Loss: 0.3161
Test Accuracy: 0.8550
```

Output Evaluation:

1. **Training and Validation Results:**

Training Loss and Accuracy:

- In the beginning, training loss is high and accuracy is low. The model starts learning the data set.
- As epochs progress, loss decreases and accuracy increases. This indicates that the model performs better on the training set.

Verification Loss and Accuracy:

- As epochs increase, verification loss and accuracy increase. The generalization ability of the model appears to be good.

2. **Test results:**

Test Loss and Accuracy:

- The performance of the model on the test set is quite good. Test loss is low and accuracy is high, indicating that the model can generalize to new data.

3. **Evaluation:**

- The model performs similarly on the validation and test sets. This indicates that the model is not overfitting and can generalize to new data.
- The final accuracy of the model is 85.5%, which means that the model can correctly classify 85.5% of the observations in the given dataset.