UNIVERSITY OF ATHENS
DEPARTMENT OF INFORMATICS AND TELECOMMUNICATIONS

Deep Learning for NLP

Student name: *<Thodoris Minaidis>*
sdi: <sdi1900113>

Course: *Artificial Intelligence II (M138, M226, M262, M325)*
Semester: *Fall Semester 2023*

Contents

1. Abstract

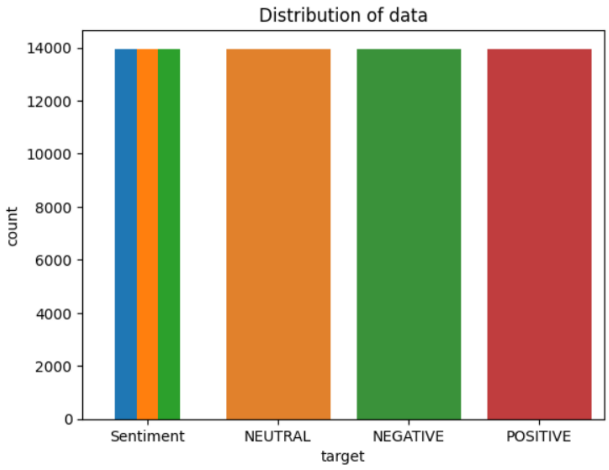
Our exercise is about sentiment analysis on tweeter comments. This is implemented using `Deep Neural Networks`. We used `Pytorch` Framework and `Word2Vec` word embeddings. There are three possible sentiments:

- POSITIVE
- NEGATIVE
- NEUTRAL

We have to process some data and create a program that make a prediction about the sentiment of a tweeter text. The prediction must be as better as it cans.

2. Data processing and analysis

2.1. Pre-processing



2.2. Pre-processing

When reading the CSV file and storing it in the DataFrame (`df`), we start the process of cleaning tweets. This cleaning includes:

- Cleaning all the '#' characters and mentions like `@minaidis` (`cleaning_txt`)
- Cleaning punctuation marks (`cleaning_punctuations`)
- Cleaning repeating characters that are often useless (`cleaning_repeating_char`)
- Cleaning URLs (`cleaning_URLs`)
- Cleaning all numbers from texts (`cleaning_numbers`)
- Cleaning all single characters (`cleaning_single_letter_words`)

After this cleaning I continued with:

- Text tokenization (`word_tokenize`)
- Cleaning stopwords (`using nltk stopwords`)
- Text lemmatization (`WordNetLemmatizer from nltk`) (exepted because Kayagle don't accept it with my implementation)

2.3. Analysis

After all the staff from the first exercise , the next step is the Word2Vec model creation, TensorDataset and DataLoader use. After, the training and validation are ready to be used by neural network.

2.4. Data partitioning for train, test and validation

The partitioning is ready. I made test and train one df to make the cleaning and after i slit them again.

3. Algorithms and Experiments

3.1. Experiments

I understood the meaning of Pytorch dataset and Word2Vec. After , i created the word2vec model, the dataset and the neural network with the name SentimentClassifier. After, i train the model and and i use it on validation

```
Training F1-Score: 0.4081
Training Classification Report:
              precision    recall  f1-score   support

           0       0.53      0.39      0.45     16628
           1       0.30      0.41      0.35      8929
           2       0.37      0.41      0.39     10951

    accuracy                           0.40     36508
   macro avg       0.40      0.41      0.40     36508
weighted avg       0.43      0.40      0.41     36508

Training Confusion Matrix:
[[6477 5098 5053]
 [2650 3695 2584]
 [3050 3369 4532]]
Validation F1-Score: 0.4109
Validation Classification Report:
              precision    recall  f1-score   support

           0       0.53      0.39      0.45      2357
           1       0.27      0.43      0.33      1057
           2       0.41      0.40      0.41      1776

    accuracy                           0.40      5190
...
Validation Confusion Matrix:
[[918 725 714]
 [298 458 301]
 [514 545 717]]
```

3.1.1. Table of trials.

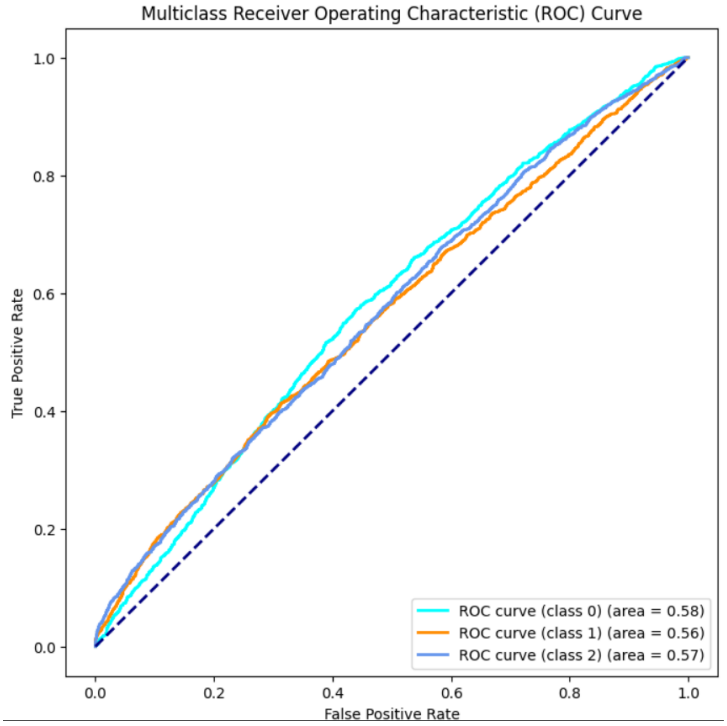
3.2. Optimization techniques

For the neural network, i used torch.optim.Adam. On general, the cleaning help the training making it faster.

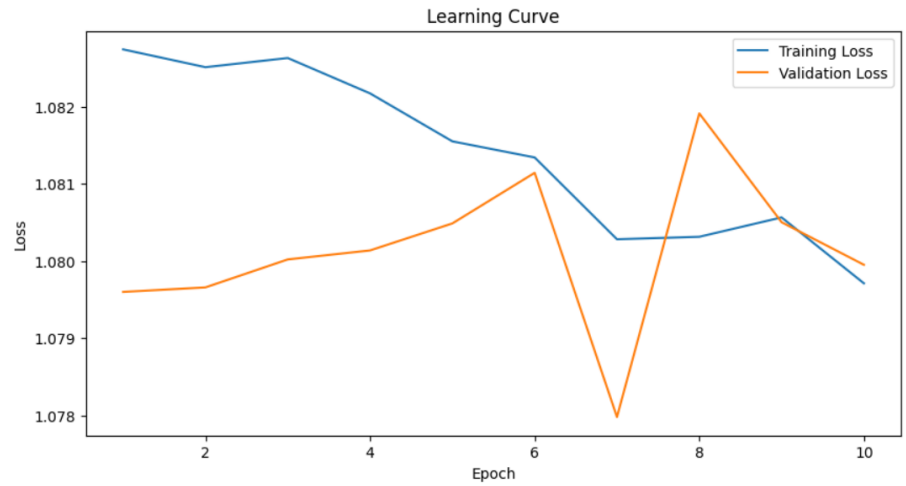
3.3. Evaluation

F1 scores found it using from sklearn.metrics the import f1 score. With these i saw these results and they were close to 1st homework scores.

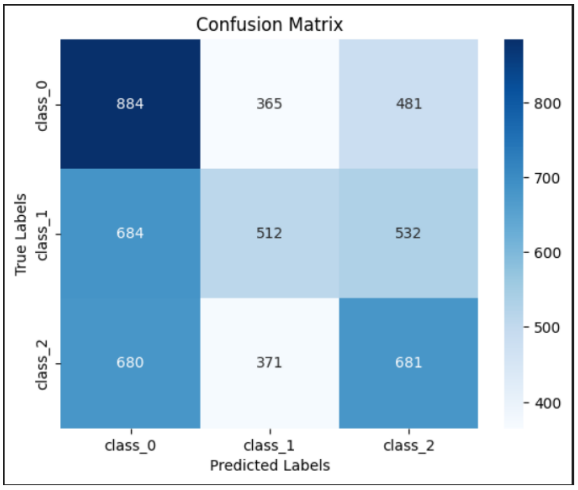
3.3.1. ROC curve



3.3.2. Learning Curve



3.3.3. Confusion matrix



.

4. Results and Overall Analysis

4.1. Results Analysis

I think predictions are on a same level as o first exercise.
<Provide and comment diagrams and curves>

4.1.1. Best trial. <Showcase best trial>

4.2. Comparison with the first project

The first exercise was a lot easier . The only difficulty was the adjustment to kaggle, LaTex again and on this type of AI. Now the neural network is a huge part, difficult to undestand. Also , because of the complexity, debugging was also difficult and sometimes you feel that you don't really understand what you are doing.

5. Bibliography

References

[1] Scikit-learn.org. URL: https://scikit-learn.org

[2] Problem solver: Stackoverflow . URL: https://stackoverflow.com

[3] ChatGPT . URL: https://www.chat.openai.com