Deep learning (097200) spring 2022

Exercise 2 - Emotion detection

Due date: 2.6.22

Exercise:

The field of detecting which emotion is represented in a text is developing and being studied due to its usefulness. For example, detecting if a review is positive or negative and more.

In this exercise you will detect the emotion of a sentence. The goal of this exercise is to get familiar with recurrent neural networks. You should try different model architectures, using optimization and regularization methods and changing the hyper parameters to see how it affects the performance.

data

The data is a csv file containing tweets and their labels according to the emotion – {happiness, sadness, neutral}. Every row in the file (except for the header) is an example.

Examples: (Notepad++ view)

happiness,Welcome @doeko ! Really glad to know you here. Your products rox man

sadness,Disappointment really sucks! I'm getting used to it.

neutral,I just want to Sleep.

You have a train file – "trainEmotions.csv" and a test file – "testEmotions.csv".

Submission instructions

You need to submit:

1. hw2_id1_id2.pdf - A pdf report (details below)
2. hw2_id1_id2 - A folder containing the following files:
    a. **predict.py:** This script gets as a commend line argument a path to a file (which will be in the same format as the train and the test). The script needs to create in the current directory a file named "prediction.csv" which will contain the tweets and their predicted emotion. This file should be in the same format as that train and test files (make sure that the labels are the name of the emotion). The script should predict a label to all the rows in the given file and write them in the same order as they were originally.
    b. **model.pkl:** The trained network with trained weights as a .pkl file.

c. Code (python file) able to reproduce your results. You should also submit all the code files you wrote and used for this exercise.

The report:

The report should contain:

- Data loading and pre processing of the data.
- Architecture (describe fully and/or refer to original paper and code)
- Loss function
- Optimization
- Regularization
- Hyper parameters tuning
- Train + Test loss graph
- Train + Test confusion matrix (a 3x3 matrix real x predicted)
- Train + Test accuracy (of the submitted model)
- Conclusions

Your conclusions and explanations should be based on the actual results you received during your attempts.

The report needs to be typed and in English. It should be comprehensive but does not need to be very long.

Evaluation of the wet part:

The evaluation will be based on the accuracy of your predictions. The accuracy will be calculated on the prediction.csv file from the predict.py script.

Competitive part:

In addition to the accuracy on the test for the evaluation part, the predict.py script will be applied to another dataset (in the same format) and the accuracy will be calculated for the predictions.

Grades policy:

- Report – 70:
- Wet part – 20%
- Competitive part – 20%

(max 110 points)

Notes:

You are **not** allowed to use ready models from torch.

If the size of the code or the model is too big add a Google-Drive link to the report.

The exercise will be checked on a machine like the GPU machine you have in Azure. We will compare the result of the predict.py script to the test.csv in the wet part and another file in the competitive part. You should check the this works before submitting. A failure in running the files will result in losing points in the wet part and competitive part.

Questions send to Naama – nberman@campus.technion.ac.il

Good – Luck!