Exercise 3 - Emotion Recognition

Convoluted Feelings

Deadlines

The deadline for Exercise 3 is **14.11.2022, 23:59 (Zurich Time).**
The deadline for the peer review is **21.11.2022, 23:59 (Zurich Time).** You will find instructions for the peer review process at the end of this document.
The deadline for feedback to your peer reviewers is **26.11.2022, 23:59 (Zurich Time).**

Learning goals

This exercise builds on the text classification task you solved in Exercise 1. By completing this exercise you should …

- … understand CNNs.
- … be able to implement CNNs in PyTorch or PyTorch Lightning.
- … deepen your understanding of the role of hyper-parameters, regularisation, and handling class imbalance.
- … perform an error analysis of machine learning models.

Please keep in mind that you can always consult and use the exercise forum if you get stuck (note that we have a separate forum for the exercises).

Deliverables

We encourage you to hand in your solutions as a Colab-Notebook. **Download your notebook as a .ipynb file**. That way your reviewers can view and execute your code. Or can view your already executed code.

Please hand in your code and your lab report. Hand in the following files and name them exactly in the following fashion:

- ex03_cnn.ipynb
- ex03_labreport.pdf

zip it and name the zip-folder *ex03_ml4nlp1.zip*. The .ipynb files should contain your well documented AND EXECUTABLE code. We recommend you use Google's Colaboratory, where you have access to GPU time.

Please submit the lab report in PDF format. The lab report should contain a detailed description of the approaches you have used to solve this exercise. Please also include results. In this exercise description, we highlight places in green where we expect a statement about an issue in your lab report.

Please note:

- Your peers need to be able to run your code. If it does not work, you will not be able to obtain the maximum number of points.
- DO NOT submit the data files!

Data

For this exercise, you will be working on the Twitter Emotion Recognition task. The goal of this task is to infer the affectual state of a person from their tweet. You will be using the Tweeteval dataset to train your model. You can access the related GitHub repositories from the links below.

- Tweeteval Repository: https://github.com/cardiffnlp/tweeteval
- Emotion Detection (our task): https://github.com/cardiffnlp/tweeteval/tree/main/datasets/emotion

We believe that being able to process and load custom datasets for your projects/research are important skills. Most tasks you will complete as an NLP practitioner/researcher will require you to handle different data sources, formats and files. Therefore in Exercise 3, we want you to figure out how to load the emotion recognition dataset from its respective repository using all the necessary files (e.g. train_text.txt, train_labels.txt, mapping.txt) and train your model.

Emotion Recognition with a CNN

Implement an emotion recognition classifier in PyTorch or PyTorch Lightning. We suggest reusing and adjusting the class structure from exercise 2 (which may be inspired by Rao and McMahan). However, you are free to create your own, new class structure. Keep in mind that for emotion prediction we work on the word level instead of character level in Exercise 1. Thus, your Vocabulary class (that is, if you have one) will not hold a vocabulary of characters. Exercise 3 will cover binary text classification with CNNs, so your model will predict between two classes at most.

Remember to document your code with docstrings and/or comments and/or text cells.

1. Pick two different emotion classes for your model to predict (e.g. anger and joy). Load/filter your dataset to include only the related class data. Create another dataset and change only one of the classes (e.g. anger and sadness) this time.
2. Your goal is to find the optimal model architecture and training regime for your CNN classifier. Pick one of the datasets you created and start experimenting. Try out at least three different combinations of the following hyperparameters, which you consider well-performing, and report the combinations and corresponding results (accuracy and F_1-macro) on the **development set** in a table.:
    a. optimizer
    b. learning rate
    c. dropout
    d. # of filters
    e. different strides
    f. different kernel sizes
    g. different pooling strategies
    h. batch sizes
    i. any other thing you want to test
3. Use your best performing model settings to train another model on the second dataset this time. Report the model performance (accuracy and F1-macro) on the test-set of both datasets.
4. Reason about the observed effects of your 3 best hyperparameter settings and dataset variations on model performance. You do not need to be sure that the reasons you provide are correct – the goal is to provide educated (or well-reasoned) guesses.

Keep track of the loss to interrupt early if a model does not converge.

Important:

a. In order to protect your anonymity for the peer review: When posting in the forum you now have the option to choose a pseudonym. Post under a pseudonym that has no connection to you and does not allow for any inferences about who you are.
b. If you work in a group, it is enough if only one group member posts the test set result.
c. If you filtered any languages in your preprocessing, add a "*" after your accuracy and F1-macro score. This way you don't disclose many details about your preprocessing but you still let other people know that the scores are not comparable to the scores that resulted by predicting on all languages.
d. Only evaluate on the test set once and do not use the devset for retraining with the same hyperparameters, please, use only the official training set for training. Do not optimise hyperparameters based on test set results.

Important

Please make sure you run your code on Google Colab with GPU selected. Select the GPU via "Edit" → "Notebook Settings" and then choose the GPU hardware accelerator.

Peer Review Instructions

If you are not already registered on Eduflow follow this link https://app.eduflow.com/join/GXHN93 and register with the E-mail address you use for OLAT. Then you should be added to the course page automatically.
As soon as the deadline for handing in the exercise expires you will have time to review the submissions of your peers. You need to do **2 reviews** to get the maximum number of points for this exercise.
Here some more rules:

- If you do not submit 2 reviews, the maximum number of points you can achieve is 0.75 (from a total of 1).
- Please use full sentences when giving feedback.
- Be critical, helpful, and fair!
- **All reviews are anonymous: Do not put your name into the python scripts, the lab report or the file names.**
- You must also give your reviewers feedback. The same criteria as above apply.
- If you consistently provide very helpful feedback, you can be awarded with a bonus of 0.5 in total in case you didn't achieve the full 6 points from all exercises. A maximum of 6 points from the exercises can go into the final grade.

Groups:

- You can create groups of two to solve the exercise together.
- Both students should submit the solutions separately.
- If you did not already work together for the previous exercise, write a small post in the "Groups"-thread in the exercise forum on OLAT to notify the instructors about the group.
- As a group member, you still have to review two submissions with your own eduflow account. However, you may work together in the group to write all 4 reviews.