Exercise 3 - Emotion Recognition

Convoluted Feelings

Deadlines

The deadline for Exercise 3 is **12.11.2023, 23:59 (Zurich Time).**
The deadline for the peer review is **17.11.2023, 23:59 (Zurich Time).** Please read the instructions for the peer review process at the end of this document.

Learning goals

This exercise builds on a 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.
- … have a deeper understanding of the role of hyper-parameters and regularization.

Please remember that you can always consult and use the exercise forum if you get stuck!

Deliverables

We request that you hand in your solutions as self-contained Jupyter Notebooks (ipynb files). That way, your reviewers can view and execute your code without potential dependency problems and/or installing new packages or versions of packages. The notebooks contain your well-documented, EXECUTED, and EXECUTABLE code.

You will also have to write a 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 the results. We highlight sections in this sheet **in green** where we expect a short statement (general thoughts, issues, etc.) in the lab report.

Please hand in all files named exactly as follows:

- ex03_cnn.ipynb
- ex03_labreport.pdf

zip them and name the zip-folder *ex03_ml4nlp1.zip*.

Please note:

- Organize your code such that it is executable when assuming that the data is in the same folder as the scripts. However, DO NOT submit the data files!
- The cells must have already been run and the output must be visible to anyone checking your notebook without having to run the code again.
- Your assessors must be able to run your code. If it doesn't work, you can't get the maximum score.

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

The ability to process and load custom datasets for your projects/research is an important skill. Most tasks you will complete as an NLP practitioner/researcher will require you to handle different data sources, formats, and files. Therefore, 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.

Task: Emotion Recognition with a CNN

Implement an emotion recognition classifier in PyTorch or PyTorch Lightning. You can reuse the class structure from exercise 2, which is an adaption from Rao and McMahan. However, you are free to create your own, new class structure. Keep in mind that for emotion prediction, unlike in Exercise 1, we work on the *word level* instead of the character level. Thus, your Vocabulary class (if you have one) will not hold a vocabulary of characters.

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. Experiment with at least three different combinations (sets) of hyperparameters, with at least two different values each, e.g., optimizer, learning rate, dropout, # of filters, stride, kernel size, pooling, and batch size. **Report the combinations and corresponding results (accuracy and F1-macro) on the development set in a table:**
3. **Use your best-performing model settings to train another model on the second dataset. Report the model performance (accuracy and F1-macro) on the test set of both datasets.**
4. What could be the reason that the specific combination/values of hyperparameters resulted in the best model performance? Don't worry about the exact reasoning, the goal is just to provide educated (or well-reasoned) guesses.

Submission & Peer Review Guidelines:

Peer Reviews will be carried out on OLAT.
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 **3 reviews** to get the maximum points for this exercise.
Here are some additional rules:

- **All file submissions are anonymous (for peer review purposes): Do not write your name into the Python scripts, the lab report, or the file names.**
- **ONLY ONE** member of each team submits on OLAT.
- Please submit a zip folder containing all the deliverables.

Groups & Peer Reviews:

- You can create groups of up to three people to solve the exercise together. Each member should contribute equally!
- If you did not already work together for the previous exercise (or already submitted a post with the same team members), write a small post in the "Assignment Team Submission Thread" in the exercise forum on OLAT to notify the instructors about the group.
- Only **one team member submits the solutions.**
- Only the submitting team member will have access to the peer review, however, you should distribute the workload evenly!
- If you do not submit 3 reviews, the maximum number of points you can achieve is 0.75 (out of 1).
- Please use full sentences when giving feedback.
- Be critical, helpful, and fair!
- Please answer all the review questions of the peer review.