Artificial Intelligence II
Deep Learning for Natural Language Processing
Fall Semester 2023
Homework 3
25% of the course mark
Announced: December 27, 2023
Due: January 31, 2024 before 23:59

Description

As in Homework 2, in this homework, you have to develop a **sentiment classifier**. This time you will be using **a bidirectional stacked RNNs with LSTM/GRU cells** for the **Twitter dataset about the Greek general elections**, that has been provided in the previous homework.

You should again use the machine learning framework PyTorch (https://pytorch.org/), and the inputs to your model must be **Word2Vec word embeddings**.

Experiments

Use the **Adam optimizer** and the **cross-entropy loss function** for your experiments. For the development of the models, you can experiment with:

- the number of stacked RNNs
- the number of hidden layers
- type of cells
- skip connections
- gradient clipping
- dropout probability

Before you start the homework, make sure that you have studied the relevant slides of the course (PDF files "Language Modelling and RNNs", "Vanishing Gradients and fancy RNNs", and "Machine Translation") or any other relevant literature you may find useful.

It is your responsibility to choose all the details of developing a good model (e.g., whether to choose the hyper-parameters of the algorithm, how to make sure that your model does not underfit or overfit etc.).

Auto-configuration framework

Selecting the most efficient configuration of the model is tricky, hence you should use an auto-configuration framework. Optuna is recommended. More about this framework will also be presented in the labs. Keep in mind, that using this framework will be for your benefit, as it will find the best model and at the same time it will "explain" how. If you use it, provide plots and tables created by this framework and explain what you observe. More about the visualizations here.

Bonus

Attention is all you need! You will get a bonus 20% if you also utilize profitably the mechanism of attention in your architecture.

Evaluation

You should plot learning curves that show that your models are not over-fitting or under-fitting. Also, you should use the toolkit Scikit-Learn (`https://scikit-learn.org/stable/`) and evaluate your classifier using *precision, recall* and *F-measure*.

Kaggle

You will submit your code (in the form of a Jupyter Notebook) through a Kaggle competition. Make sure to do the following:

- Your **team name** and the **notebook name** must be your academic identification number (Αριθμός Μητρώου - For example if it's sdiXXYYYYY then sdiXXYYYYY.ipynb and sdiXXYYYYY).
- Your solution must be submitted as a Notebook that outputs a result file named "submission.csv", **NOT AS A FILE UPLOAD**! The result file must follow the format specified in the provided "sample_submission.csv" file and must contain the predictions that your model makes over the test set.
- You must share your Notebook on Kaggle with the Teaching Assistant responsible for grading this assignment. **DON'T SHARE YOUR NOTEBOOK PUBLICLY**!

Data

You can view the data here. You should read your dataset from your kaggle notebook. No need to download/upload it.

Report

For this project, and the next ones, you are asked to create a detailed report. For this reason we provide you with a template in LaTeX. You may use Overleaf online editor. Find the template **here**. Open Overleaf, create an account if you don't have one already, and then upload the zip file by selecting: New project; Upload project; Select a .zip file; (it uses a pdfLaTeX compiler).

If you are having any issues in writing with LaTeX, you can write it to word/docs following the template in LaTeX. However we are strongly advice you, to create it in LaTeX, as Overleaf now provides you with many shortcuts and abilities making it easier for you.

Keep in mind that in your report we want to see your remarks and comparison with your approaches in Homework 1 and 2.

Grading

Implementation: Code, kaggle submission [**Total 70%**]

- Data processing: [**10%**]
- Model creation: [**20%**]
- Experiments: [**30%**]
- Fine-tuning & Optimization: [**10%**]

Report: Analysis and Presentation [**Total 30%**]

- Experiments: [**10%**]
- Analysis: [**15%**]
- Plots: [**5%**]

Bonus: Attention mechanism [**Total 20%**]

- Implementation: [**10%**]
- Experiments & Analysis: [**5%**]
- Reporting: [**5%**]

So, the total for this homework sums to: [**Total: 120%**]

Submission guides

We expect you to:

1. Submit your **Jupyter Notebook** (and make is available to supervisors) in **Kaggle** and **only**.*
2. Submit your report in a **.pdf** format from e-class. Name your report like: [**full-id].pdf** (e.g. ZZZZZZXXYYYYY.pdf if you are a bachelor student in this department).

**We won't accept code submissions from e-class/e-mails, etc.*

Support

Konstantinos Nikoletos (k.nikoletos[at]di.uoa.gr) will be supervising this assignment. Please submit your questions on Piazza under the corresponding directory **(hw3)**.