COMP 331 Programming Assignment 2

Fall 2025

Due October 26, 2025 by 11:59PM

Submission instructions

You may submit this assignment alone or as a pair (one submission per pair).

Description

In this assignment, you will implement a neural network model for text classification and make predictions with it using PyTorch. I have provided starter code, datasets, and a relevant tutorial. Please make sure to review the PyTorch tutorial part II linked in the assignment (on Canvas) before working on this assignment.

You will write:

1. a trainer for a neural network model. You will build a model with PyTorch, complete the training loop and an evaluation function. Stub code with TODOs is provided in `train.py`.
2. a classifier that takes a pre-trained model and make predictions on unlabeled data, and save them to a file. Stub code with TODOs is provided in `predict.py`.
3. a brief write-up in a `.pdf` file summarizing your implementation and model performance. Based on your model's predictions on unlabeled data (a few examples), does it behave as you would expect? If you have experimented with different model architectures, features (e.g., pre-trained embeddings, additional features – combining embeddings with handcrafted features?), parameters (e.g., training epochs, embedding size, hidden sizes, batch sizes), loss functions, optimizers, discuss those as well for extra credit.

Datasets

You are provided with IMDB movie reviews [1] for binary sentiment classification, including a training set, a test set, and an unlabeled set. It is a randomly sampled subset of the original large dataset created by Andrew Maas and collaborators [1]. This dataset contains highly polar reviews with two classes: positive and negative.

Size of the data you will work on:

- 10,000 data points in `train.txt`
- 5,000 in `test.txt`
- 5,000 in `unlabeled.txt`

They are in the `data` folder.

Data structure: ⟨text⟩TAB⟨label⟩, or ⟨text⟩`\t`⟨label⟩

- `text`: text strings, IMDB movie reviews
- `label`: 0 (negative), 1 (positive)

[1] Downloaded from (https://huggingface.co/datasets/stanfordnlp/imdb)

Submission

Submit the following files as **one zipped file** on Canvas:

1. `utils.py`: This file contains data loading and text preprocessing utilities. You are not required to modify this file. However, if you would like to implement different tokenizers, preprocessing steps, or even pre-trained embeddings, you are free to modify them. (Understanding the text processing utilities in this file would be necessary for you to complete your tasks in the next two files.)
2. `train.py`: For the model class, you must include an embedding layer, at least one linear layer and an activation function. It should output a trained model and provide evaluation of your model. Stub code is provided, and you can modify the classes and functions accordingly.
3. {`model_name`}`.pth`: Your model output stored as a PyTorch state dictionary (see tutorial).
4. `predict.py`: It takes in a trained model (output from `train.py`) and output predictions for the unlabeled dataset `unlabeled.txt`.
5. (Optional) Other scripts required by your model. Describe them in `README.md`
6. `README.md`: Briefly describe how to use your code.
7. (Optional). `requirements.txt`: Optional, but recommended if you installed additional packages than imported in the starter code. This includes all dependencies for your code. They will be installed via `pip install -r requirements.txt`. DO NOT include unused dependencies.
8. `pa2_report.pdf`: A write-up discussing your implementation, model performance, and findings (e.g., what parameters, model architecture, or optimizer worked). You could include tables or figures as appropriate (not required). Just one page is fine.
9. DO NOT submit the data

AI use

If you use AI as a thought partner or learning assistant for this assignment, refer to the AI policy on syllabus. Include a proper AI statement at the end of your report. If you have questions regarding what is or is not acceptable AI use, do ask. Make sure you understand the code you submitted in all cases. I reserve the right to ask for your explanation of any snippets of your code.

Rubric

- Completion: about 50%. Complete all the TODO tasks required.
- Correctness: about 20%. The tasks were correctly implemented. Is your model learning patterns from your data? Target an accuracy of above .60.
- Style: about 10%. Your code is well-written and well-documented (easy to understand, less hard coding, separation of concerns).
- Report: about 20%. Clear communication of what you implemented and experimented with and your conclusions drawn from these practices.
- Extra credit: up to 8%, if you tested different pre-processing ideas, pre-trained embeddings, model architectures beyond what this assignment asks for. Discuss them in your report (even better, with figures/tables!).

References

[1] Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., and Potts, C. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies* (Portland, Oregon, USA, June 2011), Association for Computational Linguistics, pp. 142–150.