Assignment 1: Sarcasm Detection

CS 577: Natural Language Processing • Fall 2025

Due: September 26th @ 11:59pm EST

Overview.

In this assignment you will train a model on a *sarcasm detection* dataset that we will provide. This dataset is a collection of news headlines, collected from the Onion (a satirical news site) and the Huff Post (a serious news source). You will

1. Train a neural network on the dataset
2. Perform a feature analysis to gain insight in to what features the model focuses on
3. Answer the discussion questions below

The dataset is a list of news headlines annotated as either sarcastic, or serious. An example of a sarcastic headline would be *"internal affairs investigator disappointed conspiracy doesn't go all the way to the top"* or *"Mars probe destroyed by orbiting Spielberg-Gates space palace"*. These are headlines that are not true, and are meant to be funny. We've done most of the preprocessing for you, so you should be able to focus on the model. You are welcome to change how the data is vectorized, but note that **we will be running the grading script *as is* without looking at your code, so your code must run according to specification.**

Part A (50 points): Model Training and Analysis

You are free to create whatever kind of neural network you want to, so long as it works with the grading script we've provided. **Your model must achieve higher than 75 percent accuracy to receive credit**. You are free to adjust any hyperparameters in the neural network that you want, and use any architecture that you think will work. You can also use any form of representation in your model **except for pretrained LLM embeddings**. Note that for this assignment, **you have to use raw, untrained PyTorch layers that you train from scratch to construct your model, instead of relying on anything that's pretrained. Not doing so will result in a loss of credit**.

However, after you train your model, you will have to perform *feature analysis*, which is the process of examining and understanding what the model has learned and how it makes decisions. This will be more difficult if your model is more complex. The only requirement for the feature analysis is that whatever conclusions you arrive at must be backed by an examination of the model, whether that's by looking at the internal weights, or trying example sentences, or something else. Again, the more complicated your model is, the harder this will be.

Type up the results of your feature analysis in the same PDF as your discussion questions, but on a different set of pages. Your analysis must include:

- The way you performed the analysis
- What features are most significant to the model
- What features are the least significant to the model
- Whether or not any of your results surprises you or not, and a hypothesis for why your model learned to behave in this manner.

We need you to provide your code demonstrating your analysis and results as well. The easiest way to do this would be to do your work in a Jupyter notebook with your code and results clearly displayed. You can save the notebook as a PDF and then attach it to the PDF containing your discussion questions. There are lots of websites on the internet that allow you to combine two PDFs into one so you can use this before submitting to Gradescope. Also be sure to use the `conda env` specified in the `yml` file as well. Here is a resource on how to do this.

Part B (40 pts): Discussion Questions

Answer the following questions on a separate page for each (i.e., one page per question). After each question below, the TA who created it is listed, and they will likewise grade those questions, so reach out to them if you have any questions.

1. Based on your feature analysis, do you think you could construct a model that's deterministic (not neural) to classify the headlines? What would this model look like? How well do you think it would do? (Nathaniel)
2. OpenAI announces it has an LLM that scores 100 percent on this and other sarcasm benchmarks. Has the task of sarcasm detection been solved? Or is there something inherent to this task that makes this not true? (Nathaniel)
3. Neural networks are powerful machines; if not regularized, they will tend to overfit to the training data and not generalize well. Discuss three possible techniques (one related to data augmentation, one related to the loss function, and one related to the training procedure) to regularize the neural network. (Yunxin)
4. Can your neural network be reused/fine-tuned for other tasks? If no, state the reasons. If yes, state what kind of tasks are suitable, and what steps do you need to take? (Yunxin)

Submission Instructions

- **Report PDF:** submit your report with your results from Part A and discussion questions to Gradescope.
- **Code**: Submit your code using the `turnin` command on the CS machine (`antor` or `data`):

```
turnin -c cs577 -p hw1 hw1-dir
```

Replace `hw1-dir` with the actual name of the directory.
Verify your submission using: `turnin -c cs577 -p hw1`.

- If you are unable to see your name/account on Gradescope or cs machine then please contact a course staff and one of us will add you.