Homework 3 - Neural Machine Translation

Release Date: Friday, Feb 17th, 2023
Due Date: Tuesday, March 6th, 2023 at 11:59PM

Note: For homeworks you can either work independently or in your course project groups.

In this homework, you will build a sequence-to-sequence neural machine translation model and improve it.

Part 0:
The first task for this homework is to run and replicate this paper (The code is already provided), which describes neural machine translation with attention.

The starter code for this assignment is written in PyTorch, a framework for neural networks. The starter code is based on this tutorial.

The primary file for this assignment is seq2seq.py Once you have installed PyTorch, you can view the arguments by running.

```
> python seq2seq.py -h
```

The arguments have reasonable default values for training the initial system (e.g. the file paths to the data should not need to be changed). You can inspect the defaults in the code.

One argument you should note is the load_checkpoint argument. This allows you to load in a model that was generated in a previous training run (which may be useful if you kill your training script part way through).

Part 1 (50%):
The second task for this homework is to implement the batching as in this paper. For this improvement, note the impact on training speed (you do not need to do a run training run - just run long enough to get a speed in sentences per second) and also experiment with a range of batch sizes.

Part 2 (50%, additional credit if more than one method is explored):
The final task for this homework is to try at least one method to improve your machine translation system. If you are working in a team, we expect the overall effort to scale up with team size.

Some ideas you could try include (but are not limited to):

- Implementing beam search (described here: https://arxiv.org/pdf/1211.3711.pdf http://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.pdf)
- Character Aware encoder (there are several ways of doing this, here is one: http://anthology.aclweb.org/P16-2058 but feel free to try something else creative!)
- Implementing Different types of Attention (http://aclweb.org/anthology/D15-1166)
- Copy Mechanism (https://arxiv.org/abs/1603.06393)

Other improvements:

- A good way to think about potential improvements is to look at your output and see what problems there are.
- You can also take a look at recent papers for inspiration about what problems to tackle (https://aclanthology.info).
- If you are not sure if something is a valid extension, please post on piazza.

Please be sure to describe your extensions in your write up. Please describe what extensions you tried and what experiments you ran. We are interested in your analysis in addition to your implementation.

NOTE: This is a very small data set that can be trained quickly on cpu, so some extensions may not improve performance on this data set. That's ok, please still analyze your results.

Gradescope Submission:

1. **Homework 3 - Part-1 Code (40%)**: Upload your updated version of "seq2seq.py" with batching implemented.
2. **Homework 3 - Part-2 Code (40%)**: Upload the final version of "seq2seq.py" with one or more improvements as listed in Part 2 above.
3. **Homework 3 - Part-2 Write up (20%)**: Provide a short written description (approx 2-3 pages) of what you implemented in Part 2 describing the overall algorithm and the important functions and data structures in your implementation. Cite any papers or source code you referenced in this work.

Credits: This homework assignment is adapted from prior works from Philipp Koehn, John DeNero, Chris Dyer and Adam Lopez.