BME646/ECE695DL: Homework 8
Spring 2022
Due Date: Monday, April 25,2022 (11:59pm ET)
Extension with 5 points/day penalty: Saturday, April 30, 2022 (11:59pm ET)

Turn in your solutions via BrightSpace.

1 Introduction

This homework has the following goals:

1. To gain insights into the workings of Recurrent Neural Networks. These are neural networks with feedback. You need such networks for language modeling, sequence-to-sequence learning (as in automatic translation systems), time-series data prediction, etc.
2. To understand the performance quality variations with various levels of gating.
3. To use RNNs for the modeling of variable-length product reviews provided by Amazon for automatic classification of such reviews.

2 Getting Ready for This Homework

Before embarking on this homework, do the following:

1. Carefully review the Week 12 slides on "Recurrent Neural Networks for Text Classification." Make sure you understand what is meant by the gating mechanism to address the problem of vanishing gradients that are caused by the long chains created by the feedback in a neural network.

2. Review the Week 13 slides on "Word Embeddings and Sequence-to-Sequence Learning", paying particular attention to slides 14 through 41 on word2vec and fastText. Please make yourself familiar with their use and advantages over one-hot vector encoding.

3. Download and install the newly released Version 2.2.2 of DLStudio. Note that if you '*pip install*' it directly from the 'pypi.org' website, you will not get the changes in the Examples directory of the distribution.

4. Download the text datasets from the main documentation page for DLStudio into the Examples directory of the distribution and execute the following command on the downloaded gzipped archive:

   ```
   tar xvf text_datasets_for_DLStudio.tar.gz
   ```

   This will create a subdirectory 'data' in the Examples directory and deposit all the datasets in it.

   If you followed the previous instructions, you will find the following datasets in the 'Examples/data/' directory:

   ```
   sentiment_dataset_train_400.tar.gz        (vocab_size = 64,350)
   sentiment_dataset_test_400.tar.gz

   sentiment_dataset_train_200.tar.gz        (vocab_size = 43,285)
   sentiment_dataset_test_200.tar.gz

   sentiment_dataset_train_40.tar.gz         (vocab_size = 17,001)
   sentiment_dataset_test_40.tar.gz

   sentiment_dataset_train_3.tar.gz          (vocab_size = 3,402)
   sentiment_dataset_test_3.tar.gz
   ```

   Regarding the naming convention used for the archives, a number such as 200 in the name of a dataset means that the dataset is a collection of the first 200 reviews from each of the positive-reviews and the negative-reviews files in the subdirectories for each of the 25 product categories. In other words, the dataset with the number 200 in its name contains a total of 400 reviews for each product category. All the reviews pooled together are randomized and divided in 80 : 20 ratio between the training and the testing datasets. The last dataset shown above is just for your convenience as you are debugging your code.

5. Before embarking on the implementation of text classification, we want to drive your attention back to

   ```
   torch.nn.GRU(*args, **kwargs)
   ```

   Please go through slides 68 to 87 of the Week 12 slide deck to know more about PyTorch's GRU. Slide 73 provides critical insights into batching. You will find this useful for the upcoming tasks.

6. Execute the following script in the `Examples` directory of the DLStudio distribution:

   ```
   text_classification_with_GRU_word2vec.py
   ```

   You will notice that, as supplied, the script will load in the following dataset:

   ```
   sentiment_dataset_train_40.tar.gz          (vocab_size = 17,001)
   sentiment_dataset_test_40.tar.gz
   ```

   You can try the same with dataset-200 or dataset-400 and their much larger vocabulary sets.

7. Finally, you should go through Slides 94 to 116 of the Week 12 slide deck. This introduces you to Professor Kak's implementation of the minimally gated GRU or pmGRU. You will find this in both DLStudio and its co-class DataPrediction. You can run this using

   ```
   power_load_prediction_with_pmGRU.py
   ```

   situated in the ExamplesDataPrediction directory of the DLStudio module. The data used is sourced from

   ```
   dataset_for_DataPrediction.tar.gz
   ```

   Download the dataset from the main documentation page for DLStudio.

With that, you are ready to start working on the homework described in what follows.

3 Tasks

Task 1: For this task, we would want you to use **word2vec** or **fastText** to implement a GRU based RNN. You may refer to Professor Kak's code in

`text_classification_with_GRU_word2vec.py`

for your implementation of sentiment analysis using the following 200-dataset. Please be careful about how you implement batching over here.

`sentiment_dataset_train_200.tar.gz`
`sentiment_dataset_test_200.tar.gz`

Task 2: After completing Task 1, you should make a copy of your Task 1's implementation and replace PyTorch's GRU with Professor Kak's implementation of the minimally gated pmGRU. Using the same dataset obtain your results.

Task 3: Perform a comparative study on your implementation of pmGRU vs. PyTorch's GRU. Please provide us with all comparative plots and statistics generated by you.

4 Submission Instructions

You can assume that the files exist locally.

- Make sure to submit your code in Python 3.x and not Python 2.x.
- Create a .zip archive with the following required files:

  `hw08_training.py`

  `hw08_testing.py`

  pdf report(see the submission template)

  and optionally any additional helper python modules such as `model.py`, `dataloader.py`, etc. and upload it onto the assignment link on BrightSpace.

- Please include all plots and generated statistics in your report.
- **Please do NOT include your trained model in your submission:**
  `net.pth`

  We will be executing your submitted code to generate these files for verification during validation.
- **Your code must be your own work.** We will use your source code for plagiarism detection and verification of performance. Submission of both your source code and the report (in pdf) is mandatory to receive a grade.
- You can resubmit a homework assignment as many times as you want up to the deadline. Each submission will overwrite any previous submission.