PR3: Sentiment Analysis

Published Date:
Feb 08, 2022, 12:00 a.m.

Deadline Date:
May 22, 2022, 11:59 p.m.

Description:

****
This is an individual assignment.
****

Overview and Assignment Goals:

The objective of this assignment are as follows:

- Deal with text data: parsing, feature extraction, etc.
- Implement one or more recurrent neural network type models (RNN, LSTM, GRU, etc.) for solving a binary classification problem.
- Evaluate parameter choices for chosen algorithms.
    - Use part of the training set to validate parameter choices.
- Choose the best model, i.e., best classifier + best hyper-parameter choices.

Detailed Description:
Develop predictive RNN models that can determine, given a review, whether it is positive or negative.

A practical application in e-commerce applications is to infer sentiment (or polarity) from free-form review text submitted for a range of products. In this assignment, you are given the review text and asked to predict the sentiment for 25000 movie reviews provided in the test file (*test.dat*). *Positive sentiment* is represented by a review rating of +1 and *negative sentiment* is represented by a review rating of -1. In *test.dat* you are only provided the reviews and no ground truth rating is provided. These data will be used for comparing your predictions.

Training data consists of 25000 reviews as well, provided in the file *train.dat*. Associated labels for each row of the matrix are stored in *train.labels*.

For evaluation purposes (Leaderboard Ranking) we will use the F1 Score metric comparing the predictions submitted by you on the test set with the ground truth. Some things to note:

- Some of your classmates may choose not to see the leaderboard status prior to the submission deadline. Please do not share leaderboard status information with others.

- The public leaderboard shows results for 50% of randomly chosen test instances only. This is a standard practice in data mining challenges to avoid gaming of the system. The private leaderboard will be released after the submission deadline, based on all the entries in the test set.
- In a given day (00:00:00 to 23:59:59), you are allowed to submit a prediction file only 5 times.
- The final ranking will always be based on the last submission unless you specify a certain submission to be used. Carefully decide what your final submission should be.

format.dat shows an example file containing 25000 rows alternating with +1 and -1. Your test.dat should be similar to format.dat with the same number of rows (25000), but containing the sentiment score generated by your developed model.

Rules:

- This is an individual assignment. Discussion of broad level strategies are allowed but any copying of prediction files and source codes will result in an honor code violation. This includes reusing code posted on the Web by others.
- You must train your own RNN from scratch. You may not use pre-trained language models or external features (other than the text input) in any way in this project.

Deliverables:

- **Valid submissions to the Leader Board website:** https://clp.engr.scu.edu (username is your SCU username and your password is your SCU password). You must submit your code as a Python file along with your predicted labels. Your submitted code should produce the submitted labels.
- **Canvas submission of your report:**
  Write a 2-page, single-spaced report describing details regarding the steps you followed for implementing and testing your models. The report should be in PDF format and the file should be called **<SCU_ID>.pdf**. Be sure to include the following in the report:
  - Name and SCU ID.
  - Rank & score for your submission (at the time of writing the report). If you chose not to see the leaderboard, state so.
  - Your approach. Describe your network architecture/structure in detail.
  - Feature extraction techniques you used.
  - Your methodology of choosing the approach and associated parameters.
  - A discussion on your bias/variance analysis for your training.
  - Any special instructions for running your code.

Grading:

Grading for the Assignment will be split on your implementation (70%) and report (30%). Extra credit (1% of final grade) will be awarded to the top-3 performing algorithms. Note that extra credit throughout the semester will be tallied outside of Canvas and will be added to the final grade at the end of the semester.

Files you will find on Canvas:

Train Data:	train.dat
Train Labels:	train.labels
Test Data:	test.dat
Format File:	format.txt