CS224d: Deep Learning for Natural Language Processing

Due Date: 5/21 (Thu) 11:59 PM PST.

In this assignment we will treat you as real professional datascientist. This assignment should give you the experience to run, train, optimize, debug, augment your neural nets, then rinse wash and repeat. To get state-of-the-art, you must learn to look at your errors to gain insights, then augment your model and retrain it.

Setup

All of you by now should have a functioning Python 2.7.x environment so all setup instructions are in the handout.

Get the code: Download the starter code here and the complementary written problems here.

Submitting your work

Once you are done working, zip your code base up and call it `<your-sunet-id>.zip`, for instance if your stanford email is `jdoe@stanford.edu`, your file name should be

jdoe.zip

Upload this file to the Box for this assignment.

For the written component which includes derivations and plots, please upload a PDF file of your solutions to Scoryst. When asked to map question parts to your PDF, please map the parts accordingly as courtesy to your TAs. The last part of each problem is a placeholder for the programming component (b), you could just map it to the page of the last part in your written assignment.

Tasks

There will be two parts to this assignment, the first will be a very easy Recursive Neural Network implementation to train and test multiple times. The next part is an augmented version of the plain RNN and includes one additional layer. Your job is to see how much the model improves. There is a lot of extra credit in this PSet to give you all the opportunity to really show us how capable of datascientists you are! But we also want you focusing on your projects so this PSet is intentionally short.

Q1: Recursive Neural Network (30 points)

- (a): 5 points
- (b): 5 points
- (c): 15 points
- (d): 5 points

Q2: 2-Layer Deep RNN (70 points and 30 Extra Credit points)

- (a): 15 points
- (b): 15 points
- (c): 30 points
- (d): 10 points
- (e): 15 extra points
- (f): 15 extra points

Q3: RNTN (20 extra points)

- (a): 5 Correct Derivations
- (b): 5 Correct Implementation
- (c): 10 Found Optimal Hyperparameters

Please try to finish the written component before writing code. We designed the written component to help you think through the details in your code implementation.