CSE 250B Project 4A

For this project you must work in a team of exactly two students. The joint report for your team must be submitted in hard copy by 10am on Monday March 17, 2014.

The objective of this project is to understand the recursive auto encoder (RAE) method of learning meanings for sentences. Specifically, the goal is to reproduce experimental results from the paper *Semi-Supervised Recursive Autoencoders for Predicting Sentiment Distributions* by Richard Socher *et al.* You should aim to duplicate the accuracy of 76.8% reported for RAEs with random word initialization on the movie reviews (MR) dataset in Table 4 of the paper.

Each team should implement the semi-supervised RAE method, based on the description in the paper. There are details of the method that are not explained in the paper. In particular, the gradients that are needed for training are not stated explicitly. Give all the equations for these in your report.

You can find the authors' software and datasets at `http://www.socher.org`. Read the code there to help fill in details that are not explained in the paper, but do not copy that code, whether at a high level or at a low level. You can get good results using vectors of length 20. Longer vectors can give slightly higher accuracy. Training may require more iterations of BFGS for shorter vectors, possibly because these may cause more local optima to exist.

While developing your code, take steps to confirm that it is correct. In particular, compare the result of evaluating your function that computes partial derivatives with the result of numerical differentiation of the loss function. The relative difference should be less than 10^{-6} between each numerical derivative and the corresponding backpropagation derivative. To reduce error, use central differences for numerical derivatives.

Suppose that the weight matrix W is in $\mathbb{R}^{2d \times d}$. Then the time needed to run the neural network once is $O(d^2)$. It follows that the time needed to compute the partial derivative numerically for each entry W_{ij} of W is $O(d^4)$, which is not feasible in practice. How can you verify that numerical and backpropagation derivatives are equal in less than $O(d^4)$ time? Also explain in your report other steps that you take to verify that your implementation is correct.

Investigate what the trained model has and has not learned. In particular:

1. Tabulate the ten most positive and ten most negative words.

2. Show the ten phrases predicted to be most positive and most negative in one of the cross-validation test folds.

3. Pick some interesting words and phrases, and show the other words and phrases whose meanings are most similar according to the trained model.

4. Pick some interesting sentences and show the tree structure that the greedy algorithm finds for them.

Do experiments to investigate whether it is necessary to adjust the vector representing each word. Compare the following three alternatives: (i) the full method, (ii) the full method without using derivatives to adjust the meaning vector for each word, and (iii) a bag-of-words method that adjusts meaning vectors, but does not attempt to represent the structure or meaning of sentences. For case (iii), discuss how similar this model is to standard bag-of-words logistic regression.

Implement the method first without normalization of vectors, that is using the pointwise transfer function $\tanh(a)$. After you succeed in doing the whole project with the unnormalized transfer function, then try using the normalized transfer function $\tanh(a)/||\tanh(a)||$, which is not pointwise. The symbolic derivative of the normalized function is quite complicated; be sure that you work it out correctly. Do you need to use the full Jacobian of the normalized function?

Investigate experimentally whether normalization is beneficial. Look for trends in the meaning vectors with and without normalization, and try to give a qualitative mathematical explanation of the experimental results with and without normalization. Hint: what happens when a sigmoid function is saturated?