Exercise 9: Feedforward Neural Network based Language Models

You can earn up to 10 points on this exercise.
You may work as a group of up to 3 people, but please submit your own version.
You will be using python based library keras for this assignment.
Please submit all the code required to run the neural networks on our machines.
Any submission that we cannot run on our computers without installing things, must be presented in the class

*Please **email** your solution to* `mittul.singh@lsv.uni-saarland.de` by **10:15 am, January 20, 2016**. While submitting your assignment by email, please name the file as Ex09_ <*your name*>.pdf .

Task

In this task you are required to build parts of a feedforward neural network (FFNN) based bigram language model (LM) provided at `https://bitbucket.org/mittulsingh/09-ffnn-assignment`. The code is written in python, using keras library to implement a neural network based LM. To learn more about using keras, read `http://keras.io/#getting-started-30-seconds-to-keras`. The sub tasks to building and using this neural network are as follows:

1. Install keras and h5py. Keras installation instructions are at `http://keras.io/#installation`. h5py is used to save the FFNN models, its installation instructions are available at `http://docs.h5py.org/en/latest/build.html` (0 points)

2. Process the data used in Assignment 4's Task 2. Carry out the first three steps as described in Assignment 4's Task 2. After completing these steps you should have 2385374 words in en.txt. In case you do not, you might have an encoding issue. Try setting `$LC_CTYPE` to `en_US.UTF-8` and re-doing the steps. (0 points)

3. Construct a vocabulary of words. Use the data from the previous step to construct a vocabulary file (say, en.voc) with each unique word in a separate line. This format is essential to use in the FFNN code provided. Report the number of words in the vocabulary. (Hint: use sort -u on bash shell) (0.5 points)

4. Split the resulting corpus into training/development/test sets, with the ratio 18:1:1 respectively, using the script `http://jon.dehdari.org/corpus_tools/generate_splits.pl` . Use the `--help` argument for usage info. You should have 1836227 words in the training set, 102130 words in the development set and 98345 words in the test set. (0 points)

5. Add sentence markers. Put a "<s>" (sentence begin marker) at the beginning of every sentence and a "</s>" (sentence end marker) at the end of every sentence in the training/development/test sets. This is to conform with the input data format of the provided code. (0.5 points)

6. Build the FFNN with one hidden layer. To complete this part of the task write the function `build` in the code provided. Use the arguments in the signature of the function to construct a single layer FFNN. (1 point)

7. Train the FFNN. To complete this part of the task write the function `fit` in the code provided. Use `Adagrad` optimizer and the categorical cross entropy loss function, both provided by the keras library and the arguments in the signature of the function to write the `fit` function. (1 point)

8. Run the FFNN on the training set with the constructed vocabulary (en.voc). The usage info of the tool can be printed using the -h flag. Run the FFNN for 10, 20, 50, 100, 200 and 500 hidden nodes. Use 0.2 as drop out rate. Apply the hyperbolic tangent (tanh) as the activation function. Plot the perplexity on training set, development set and test set as a function of the hidden layer size. Report the best perplexity and corresponding hidden layer size. (3 point)

9. Write a new `build` function to construct an FFNN with 2 hidden layers with the same activation function as the first. (1 point)

10. Use the best configuration from sub-part 8 for instantiating the size of the first hidden layer. Vary the second hidden layer size similar to the sub part 8 and then plot the perplexity for this neural network on training set, development set and test set. Report the best perplexity and corresponding second hidden layer size. Discuss the difference in perplexities for the two neural networks built in this assignment. (3 points)