AIT 726 Homework 2

In this assignment, you will build a language model using feed forward neural networks using either Keras or pytorch. We will use the same data from assignment 1. However, we will limit the dataset to only the 1000 reviews from the training set. Attached, you will find the relevant reviews.

We will treat language modeling as a binary classification of positive and negative n-grams, for n=2. Positive n-grams belong to the language model, whereas negative ones do not. You will create the positive and negative n-grams from the provided data as follows:

Pre-processing: Read the complete data word by word. Remove any markup tags, e.g., HTML tags, from the data. Lower case capitalized words (i.e., starts with a capital letter) but not all capital words (e.g., USA). Do not remove stopwords. Tokenize at white space and also at each punctuation. Consider emoticons in this process. You can use an emoticon tokenizer, if you so choose. If yes, specify which one.

Construct your n-grams: Create positive n-gram samples by collecting all pairs of adjacent tokens. Create 2 negative samples for each positive sample by keeping the first word the same as the positive sample, but randomly sampling the rest of the corpus for the second word. The second word can be any word in the corpus except for the first word itself.

Create your training and test data: Split your generated samples into training and test sets randomly. Keep 20% for testing. Use the rest for training.

Build and train a feed forward neural network: Build your FFNN with 2 layers (1 hidden layer and 1 output layer) with hidden vector size 20. Initialize the weights with random numbers. Experiment with mean squared error and cross entropy as your loss function. Experiment with different hidden vector sizes. Use sigmoid as the activation function and a learning rate of 0.00001. You must tune any parameters using cross-validation on the training data only. Once you have finalized your system, you are ready to evaluate on test data.

Evaluate: Compute the most likely class for each n-gram in the test set. Save your results in a.txt or .log file.

Evaluation Methods: Compute and report accuracy. Save your output in a .txt or .log file.

Documentation: Use the same documentation format from Assignment 1. Start all your files with a description of your code. Write short description of each function on top of it.

Deliverables: Submit a zip file named with student1[firstname initial][lastname]_student2[firstname initial][lastname]_[hw#].zip (i.e. student 1 jamie lee, student 2 kahyun lee: jlee_klee_hw1.zip).

Zip file should include: Your code(s), and .log file or .txt file that contains your output. You can choose whatever is convenient for you. Log can be created using logging library. Txt file can be created using simpleio library.