Timothy Truong Jr. October 8th, 2017

6.806 Assignment 2

Hidden Markov Model and Neural Network Classifier With Word Embeddings

1. (a) Number of transition probability parameters: $|\Sigma|^2$
   Number of emission probability parameters: $N|\Sigma|$
   Number of starting probability parameters: N
   Total: $N + N|\Sigma| + |\Sigma|^2$

   (b) $p(x, y|\theta) = \pi_1 \cdot p(\text{the}|1)p(2|1)p(\text{dog}|2)p(1|2)p(\text{the}|1)p(3|1) = 0$

2. (a) $|\tau|^n$

   (b) $\pi(0, v) = \begin{cases} 1 & \text{if } v = * \\ 0 & \text{if } v \neq * \end{cases}$

   (c)

$$\pi(k, v) = \max_{y_1,...,y_k \in S(k,v)} r(y_1, ..., y_k) \tag{1}$$

$$\pi(k, v) = \max_{y_1,...,y_k \in S(k,v)} \prod_{i=1}^{k} a_{y_{i-1},y_i} \prod_{i=1}^{k} b_{y_i}(x_i) \tag{2}$$

$$\pi(k, v) = \max_{y_1,...,y_{k-1} \in S(k-1,u)} \prod_{i=1}^{k-1} a_{y_{i-1},y_i} \prod_{i=1}^{k-1} b_{y_i}(x_i) a_{u,v} b_v(x_k) \tag{3}$$

$$\pi(k, v) = \max_{u \in \tau} \pi(k - 1, u) a_{u,v} b_v(x_k) \tag{4}$$

   (d) Number of subproblems: $O(n|\tau|)$
   Time per subproblem: $O(|\tau|)$
   Time to backtrack: $O(n|\tau|)$
   Total: $O(n|\tau|^2)$

3. (a) $\overline{count}(u \to v) = \sum_{i=1}^{m} \sum_{y' \in \tau^{|x^i|}} p(y'|x^i, \theta) count(x^i, y', u \to v)$ where $|x^i|$ denotes the length of output sequence x^i and τ^n represents the n-fold Cartesian product of τ.

   (b) $a_{u,v} = \frac{\overline{count}(u \to v)}{\sum_{v' \in \tau} \overline{count}(u \to v')}$

   (c) We marginalize over all $y_j \in \tau$ and apply the Markov assumption ($x_j, ..., x_n$ inde-

pendent of $x_1, ..., x_{j-1}$ given y_n):

$$p(x_1, ..., x_n|\theta) = \sum_{y_j \in \tau} p(x_1, .., x_{j-1}, .., x_j, .., x_{,n}, y_j|\theta) \tag{5}$$

$$p(x_1, ..., x_n|\theta) = \sum_{y_j \in \tau} p(x_1, ..., x_{j-1}, y_j|\theta) p(x_j, ..., x_n|y_j, \theta) \tag{6}$$

$$p(x_1, ..., x_n|\theta) = \sum_{y_j \in \tau} \alpha_j(y_j)\beta_j(y_j) \tag{7}$$

$$p(x_1, ..., x_n|\theta) = \sum_{p \in \tau} \alpha_j(p)\beta_j(p) \tag{8}$$

(d) We apply Baye's rule and the Markov assumption as above:

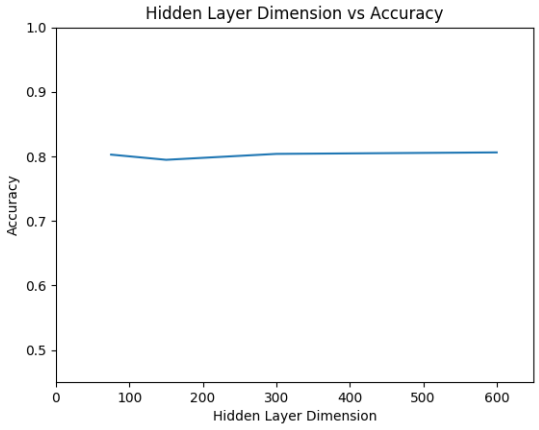
$$p(y_i = p|x_1, ..., x_n, \theta) = \frac{p(x_1, ..., x_n, y_i = p|\theta)}{p(x_1, ..., x_n|\theta)} \tag{9}$$

$$p(y_i = p|x_1, ..., x_n, \theta) = \frac{p(x_1, ..., x_i, y_i = p|\theta) p(x_i, ..., x_n, |y_i = p, \theta)}{\sum_q \alpha_q(i)\beta_q(i)} \tag{10}$$

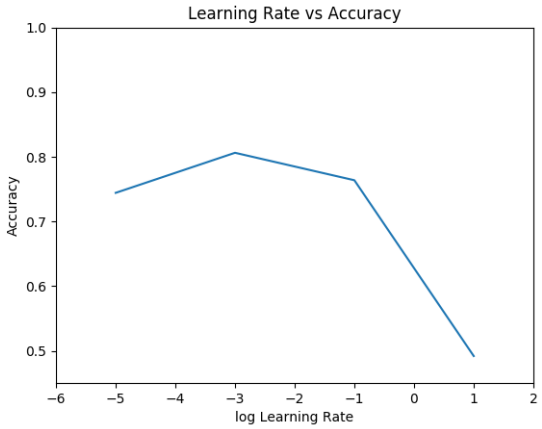
$$p(y_i = p|x_1, ..., x_n, \theta) = \frac{\alpha_p(i)\beta_p(i)}{\sum_q \alpha_q(i)\beta_q(i)} \tag{11}$$

4. (a) The parameters of the best model I found were:
Hidden Layer Dimension = 600
Learning Rate=0.001
Weight Decay=$1e-05$
The accuracy of the model on the development set was: 0.8061926605504587
The accuracy of the model on the test set was: 0.814936847885777

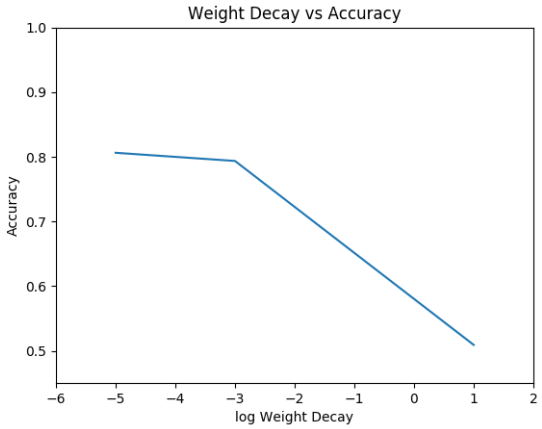
(b) Performance trend for hidden layer dimension: Approximately constant.



Performance trend for learning rate: Increasing and then decreasing.



Performance trend for weight decay: Decreasing, but more data may tell a slightly different story.



(c) Description of Model
The model was a neural network that predicted the sentiment of reviews as either positive or negative. The input of the neural network was the average embedding vector (given by GLOVE) of the words in a review. Thus the architecture of the neural network was as follows:

i. 300 dimensional input layer (GLOVE embeddings are 300 dimensional)
ii. Linear layer from input layer to hidden layer, followed by tanh activation. The size of the hidden layer varied as described in the problem.
iii. Linear layer from from the hidden layer to the output layer, which has dimension 2, for positive and negative sentiment.
iv. LogSoftmax layer on top of the output layer.

5. (a) Yes, because all the sentiment problems are quite similar. Specifically, if you look at the most important words of the logistic model from assignment 1, many of those words would probably be the most important words for analyzing hotel or restaurant reviews as well.

(b) The model would not be good for multiple aspects because even if the overall sentiment of a review is positive, the reviewer may think some aspects of the thing

being reviewed are bad. For example, its possible to that a reviewer thought a given comedy was good because it made the reviewer laugh a lot, but the reviewer may not have thought the plot of comedy was good. One way to modify the models to deal with this is to augment the dataset with the sentiment of each specific aspect we want to predict. Then the models can be trained as multi-way classifiers for all of those aspects.