IITP-03 Assignment 2: Sentiment Classification

Soyeon Kim
soyeonk2@andrew.cmu.edu

April 29, 2020

1 Task 1: Naïve Bayes

1. How did you use the dev data (e.g., how to split them for training and testing)?

   I use all the dev data to train Naïve Bayes model.

2. How did you preprocess the data?

   - Tokenize the text and remove the stop words (see stopword.list)
   - Turn all the tokens into lower case.
   - Remove all the punctuation in the tokens (e.g., "it's" will become "its" , "?" or "!" will become ""), if a token becomes empty after the transformation, then it should be removed.
   - Remove all the tokens that contain numbers.

3. For each feature you tried,

   (a) What is the feature (e.g., unigrams, bigrams, etc. briefly in no more than 2 sentences)?

   I use 'Collection Term Frequency(CTF)' as feature. For example, if my dictionary is [hello, bag, movie, happy, dog], and a training instance contains tokens [bag:2, movie:2, dog:1], my feature vector will be [0, 2, 2 ,0 ,1]. I used the top 1000 words to dictionary.

   (b) Why do you think it makes sense to use this feature?

   There are some pattern exists of negative review and positive reviews. For examples, some words will be only used in positive reviews and some words will be only used in negative reviews.

   (c) Did the feature improve accuracy, either on the heldout or on your own test set?

	dev	heldout
accuracy	0.851	0.75

   Yes, the accuracy of heldout is not good as dev but better than 0.5, so it improve accuracy.

   (d) Why do you think it did or did not improve accuracy?

   It is because the 'bag of word' feature can represent the sentiment of text. (similar answer with (b))

4. What is the final accuracy on your own test set?

   see 3 (c).

2 Task 2: Neural Networks

1. How did you use the dev data, if different from Task 1?

   I split dev data into train data and validation data, the ratio of validation data is 0.1 of total dev data.

2. How did you preprocess the data, if different from Task 1?

   same as Task1.

3. What is the architecture of your model? (recommended to include a figure that describes the entire architecture of your model)

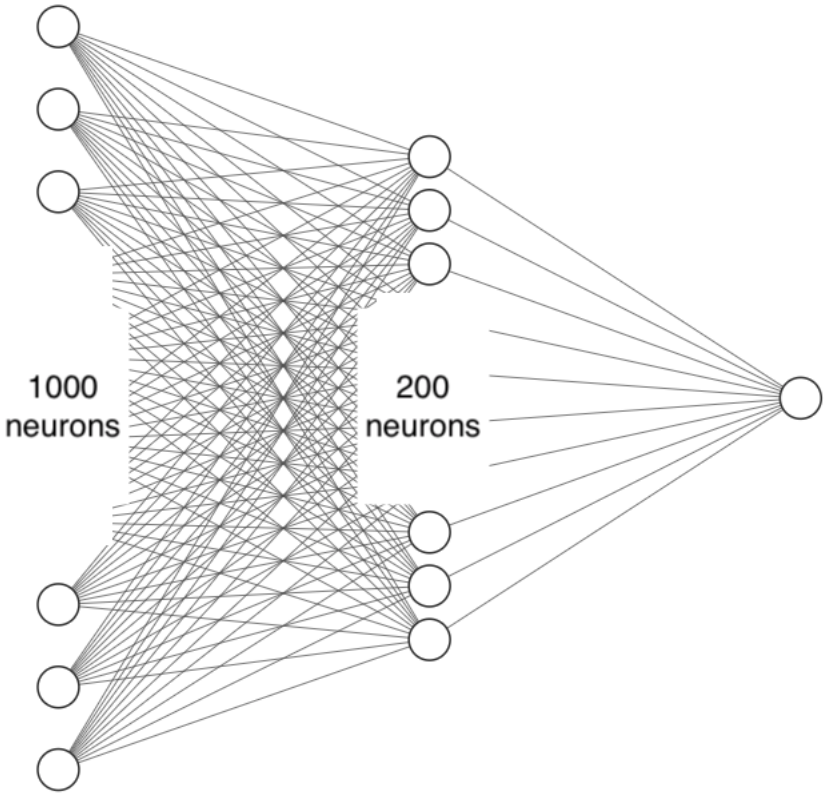

Figure 1: architecture of model

   Fully connected network with one hidden layer with 200 neurons.

4. What does the model take as input?

   Collection Term Frequency output would be an input.

5. What features did you try other than word unigrams, if any? Did they improve accuracy?

   I try to use word2vec, I don't know why but the accuracy in not improved. So I just use CTF.

6. Any non-trivial extensions you made to the model

   I set a stopping criterion. If the loss of validation grows 2 times then I stopped. It is because, if iteration goes over, the loss of train data becomes smaller but when the loss of validation grows, we can say that it implies the over-fitting. That is the reason for my stopping criteria.

	dev	heldout
naïve bayes	0.851	0.75
neural network	0.868	0.8

7. What is the final accuracy on your own test set and how is it compared to your naïve Bayes classifier?

   There is some improvement both dev and heldout. And one interesting observation is if I did not stop during train and let the neural network to train sufficiently a lot, the error for train set become 0. However, it implies the over-fitting, so the result for heldout would not be good.

8. Qualitative analysis on the model outputs and compare it with ones from your naïve Bayes

   Following example that naïve Bayes classifed as neg and neural network classifed pos:
   "I really enjoyed this movie. During the movie, I felt that I wanted Pelagia and Captain Corelli to get together. I heard myself screaming: Come on, kiss her! The movie has a happy ending. Good movie to watch in the evening when you want to chill."
   It is clearly positive review.