Maching Learning assignment 3

Mei-Shin Wu

July 2016

1 Task 1 : Logistic regression

- Sklearn result
  - Unigram : 0.8435
  - Bigram : 0.8195
- Keras result
  - Unigram : 0.6285
  - Bigram : The code had error. So I can't report accuray here.

2 Task 2 : Multi-layer Perception

Every document was represented as the average of word vectors. The dimensions for unigram and bigram are 50 and 100 correspondingly.

- scikit-neuralnetwork result
  - Unigram : 0.664
  - Bigram : 0.6725

3 Task 3 : Convolutional neural network

3.1 Model description

Input length of each document was 300 words, the output was 200 units. weights were imported by word vectors.

First trail, I used sequence of words. Second trail, I recruited word vectors as the initial weights.

Input layer

- model setting
  - Dropout = 0.2

  - nb_filter=200
  - filter_length=5
  - subsempling = 2
  - nb_epoch=20
- Accuracy : 0.76

4 Task 4: Review

Among all the models, logistic regression model achieved the highest accuracy by using sklearn library. However, the model performed a very low accuracy of 0.63 when I switched to Keras.

The accuracy of second task reported the lowest accuracy among all the models. The input data transformed tokens into the average of all the word vectors. By doing so, all the useful information was discarded. Thus the second model performed a low accuracy.

The third model, convolutional neural network model reached 0.76 accuracy (first trail). It could be seen as the second best model among all the task. The data was relatively small, therefore, the accuracy highly determined on how user set the parameters. For this task, the accuracy among several trails was bouncing between 0.50 to 1.

The above models has their pros and cons. Logistic regression model works fine when the information is simple (pure word tokens) and small dataset. However, the CNN model can work better when the user provide more information as input. Although, CNN model did not performed outstandingly in this task. But it might be due to the nonspecific pretrained word vector or the training data was too small.