Assignment4 Report

Junhan Liu(20848916)

Running instruction

1, `cd #the assignment folder`, i.e. `cd /user/junhanliu/614_a4`

2, In your command line type `python3 main.py` or `python3 main.py /xxx(directory)/.../#assignment_folder/data` to initialize script.

3, The driver script(main.py) takes only one parameter or no parameters. If passing in a parameter, feed the data folder path to the script. If none is given, the script is going to use a default path of the current running directory using `os.getcwd()+'/data'`

4, Data folder contains:

`data splits`,
`w2v.model`,
`sentences.txt`,
`tokenizer.pickle`,
`nn_relu.model`,
`nn_sigmoid.model`,
`nn_tanh.model`

Use everything provided by `data` folder, some files were regenerared to fit minor alterations in the new script. If using other files, the script might give value errors.

5, Every file needed are in `/data` folder.

FYI: some words contained inside sentences.txt might be offensive; to draw a better line between positive and negative sentiment. They are added solely for testing purposes.

Discussion

Table 1: Accuracy of Model(with activation function)

sigmoid	relu	tanh
70.266253%	76.531249%	70.753747%

The dimension of output is 2, the label is categorized using one-hot encoding which [0, 1] denoting label 1. The activation function `relu` gives the highest accuracy. Since `ReLU` substantially reduces the computational cost for training of the net. This allows the training of larger nets with more parameters at the same computational cost, thus leading to higher capacity. If the output is binary, which means by using one value 0 or 1 we can denote output label, using sigmoid as the final layer will gives higher accuracy. By altering output dimension to 1, sigmoid gives 80% accuracy. `tanh` is also like logistic sigmoid but better. The performance revealed by accuracy shows `tanh` gives slightly higher accuracy. It is mainly used classification between two classes. Briefly saying, overfitting may result in significant generalization error and bad performance

on unseen data (or test data, in the context of model development). Regularization is the counter-measure. With regularization, sometimes the results are even worse. Dropout is a simple way to prevent neural networks from overfitting. With dropouts, the models performs better. The dropout rate is fixed, for my data, dropout=0.4 performs better.