**Project 2 Report**

Yahan Liu

**Problem 1**

For Problem 1, we first imported the RCV1 dataset and labeled the CCAT topic (column 33 inrcv1 [‘target’]). Then, we made a new 1 column array from the former ‘0’and ‘1’ sparse matrix. Each article has a ‘1’ if it has been classified as CCAT and others are -1.

We change the label vector y into array. The data point X remain sparse matrix. This information will be used in the following code, since array and matrix have different calculate way in ‘numpy’.

After that, we split the data into two parts: training sets and test sets, which contain 100,000 and 700,000 articles, respectively.

**Problem 2**

For Problem 2, we used PEGASOS to train an SVM on the training articles and made a plot of training error vs. number of iterations. The Input, Process and Output of the approach is:

Regularization parameter λ

Number of iterations T

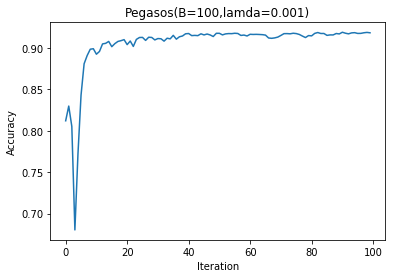
Initialize: choose w1 s.t.

For t = 1,2,3,…,T:

Choose

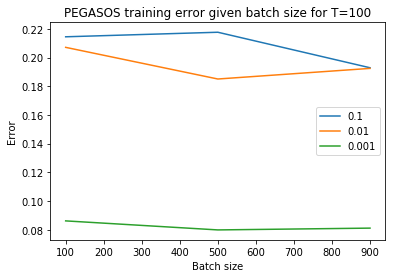
Output:

The following is the image of Accuracy VS Iteration. From this picture we can see that when it is about 18 to 20 iteration, the accuracy increase very fast. And then the accuracy increases in fluctuation. And at 100 iterations, the accuracy is largest



Plot1. PEGASOS Accuracy VS Iteration for Batch Size=100, lamda=0.001

We can change the value of Regularization Parameter λ and Number of Batch Size B to see the change of PEGASOS training error. For example, when Batch Size=500, we compare the changes of λ when T is fixed by 100. It turns out that when λ=10-3, the training error goes the lowest, which is about 8%. Therefore, through the experiment, we should choose T=100 and λ=10-3 for this problem.

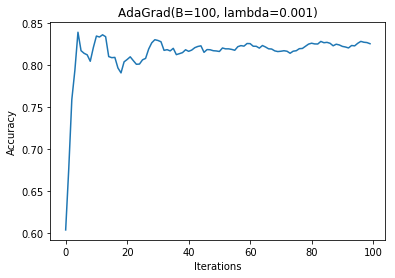


Plot2. PEGASOS Training Error VS Batch Size for Different lamda and Given Iteration T = 100

**Problem 3**

For Problem 3, we used AdaGrad and trained a classifier on the training articles. We referred a blog for the solution of this problem.[[1]](#footnote-1) For different parameters, we should get different training errors. The difference of PEGASOS with AdaGrad algorithm is that the learning rate or step size of the gradient is always the same for every iteration. For AdaGrad, however, the learning rate is always change with the gradient and the number of iteration.

And from the image of Accuracy VS Iteration, we can see that for AdaGrad, the accuracy increase faster than PEGASOS. But the accuracy is less stable and smaller.



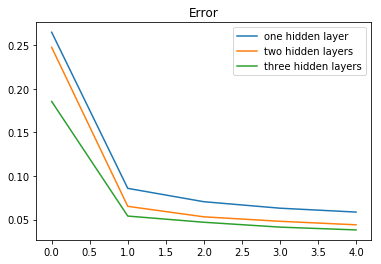
Plot3 AdaGrad Accuracy VS Iteration for Batch Size=100, lamda=0.001

**Problem 4**

In Problem 4, we use the multi-layer perception neutral network to train. For the structure of the multi-layer perception, there is an input layer, some hidden layers and an output layer. And we use ‘relu’ as the activation function. For the first part, we use one, two and three hidden layer network. We can see the errors decrease when the epoch number increase. And also the errors decrease when the number of layers increase.

‘Binary\_crossentropy’ need to use log function to calculate loss. So we must turn the ‘-1’ and ‘1’ label into ‘0’ and ‘1’label at first.

And then we use different unit numbers for the same 6 hidden layers and ‘relu’ activation function.



Plot4 NN Training Error VS Epochs for Different Number of Layers

At last we use different number of units for 6 hidden layers. And the errors are shown in the following list. The first number of the tuple is the number of units and the second number of the tuple is the errors for 6 hidden layers and 5 epochs.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Number of Units | 80 | 90 | 100 | 110 | 120 |
| Errors | 0.036219994 | 0.035139994 | 0.036029994 | 0.037199994 | 0.036879994 |

Plot5 NN Training Error of Different Number of Units

**Problem 5**

At last, we choose the parameter as following. For Pegasos and AdaGrad, iteration T = 100, Batch Size = 500, lamda = 0.001.

And for NN, we use 3 hidden layers to reduce the running time and 90 units per layer.

We use the three algorithm for training the test data. We can see that , for Pegasos and Adagrad, the best accuracy are 0.94 and 0.86 respectively. The NN can get the best accuracy about 0.97, however, the time it takes is longest. For about 30 minutes on my computer.

1. AgaGrad: <https://nipunbatra.github.io/blog/2017/linear-regression-adagrad-vs-gd.html> [↑](#footnote-ref-1)