Assignment - 2

Devanshu Singh Gaharwar
Abhiraj Kanse

September 2018

1 Graphs

1.1 For Logistic Loss

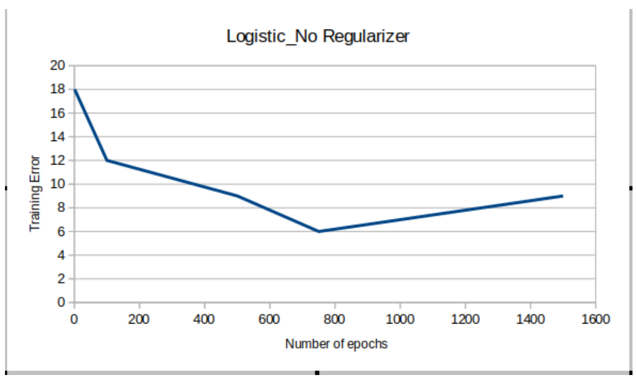

Figure 1: training error vs number of epochs without regularizer

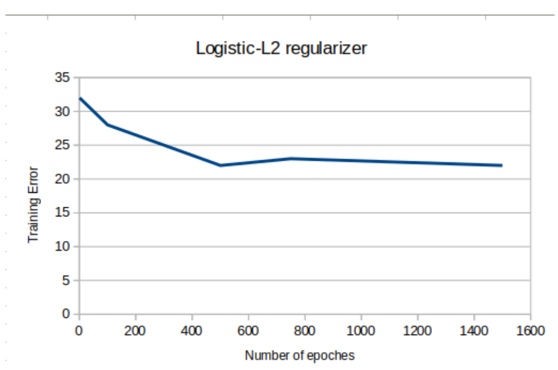

Figure 2: training error vs number of epochs with L2 regularizer

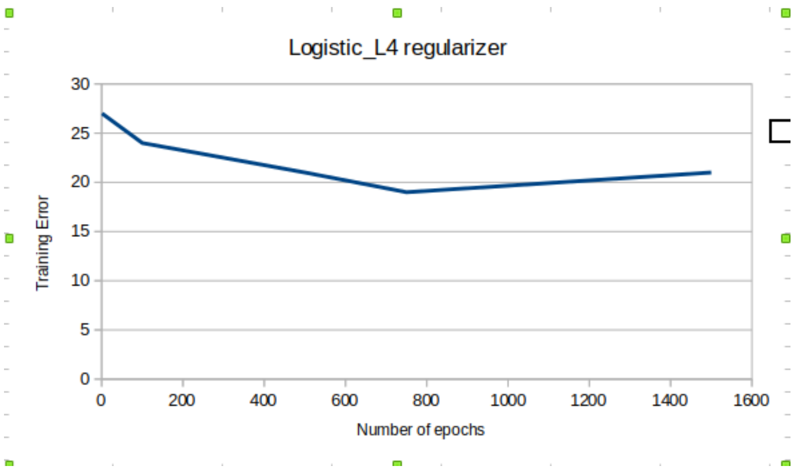

Figure 3: training error vs number of epochs with L4 regularizer

1.2 For Square Hinge Loss

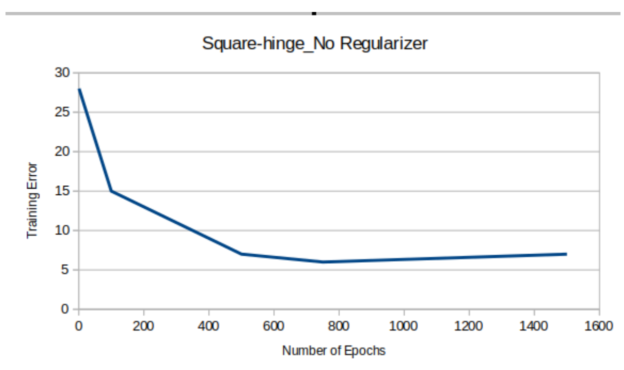

Figure 4: training error vs number of epochs without regularizer

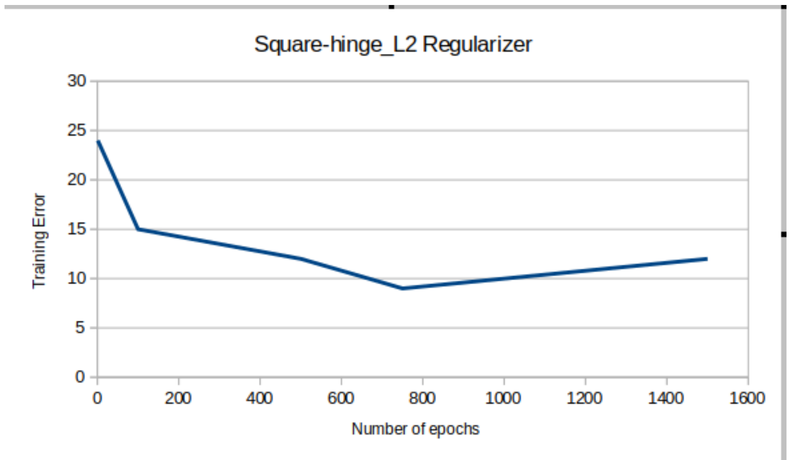

Figure 5: training error vs number of epochs with L2 regularizer

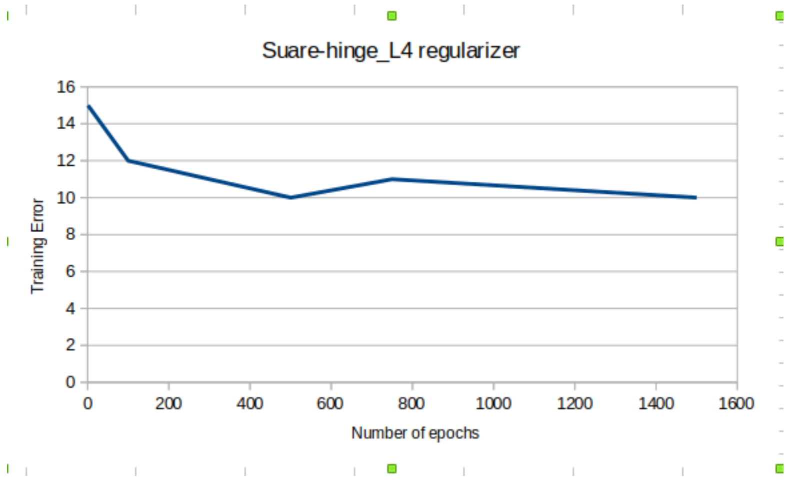

Figure 6: training error vs number of epochs with L4 regularizer

1.3 For Perceptron Loss

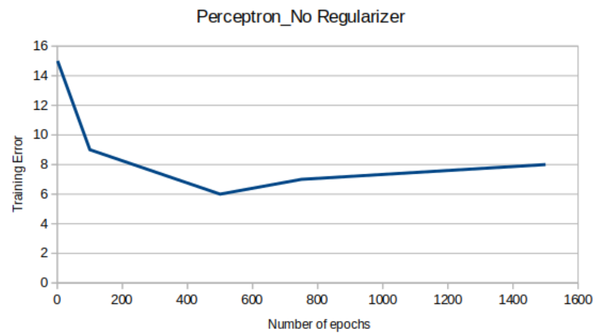

Figure 7: training error vs number of epochs without regularizer

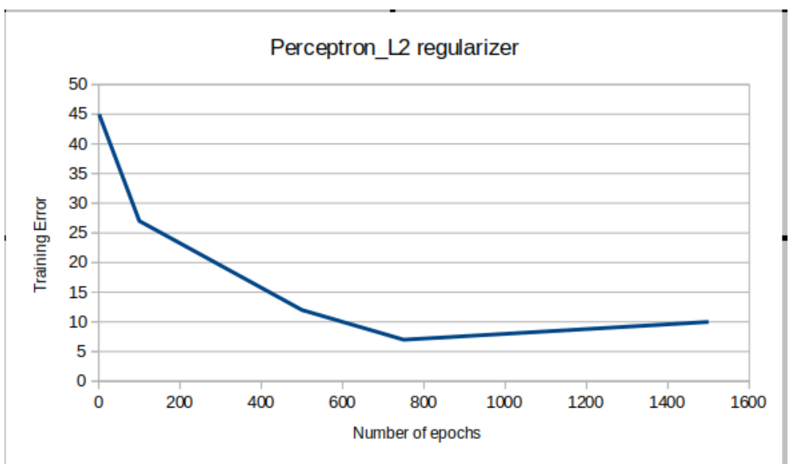

Figure 8: training error vs number of epochs with L2 regularizer

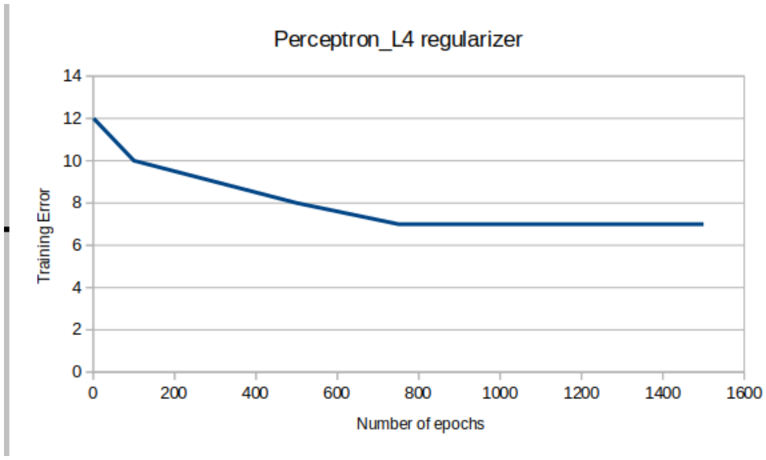

Figure 9: training error vs number of epochs with L4 regularizer

On changing the values of c similar trends were obtained for Training Error vs Number of Epochs when Regularizer was used although as the value of c changed the training error did change.More on that in the third section of the report.

2 Inference from the above plots

From the above plots it is clearly evident that for a fixed optimization function and any given value of loss weight, as the number of epochs increase the Training Error decreases initially and later since the function to be optimized has already converged the training error shows oscillatory nature.

3 Miscellaneous

- **Pre-processing the data**: We calculated the training error by dividing the train data-set into 2/3 for training and 1/3 as the validation set. And to quantify we summed up the absolute error between the predicted class label and the actual output of the validation set.Using this we found out the required parameters which minimize the error i.e. the C value, the number of epochs needed and the regularizer to be used by grid search over different obtained values.
- **Influence of different C Values :** If Regularizer was not used there is no effect on the Training Error as the Value of C changed although the loss function increases proportionally which is obvious. When regularizer was used, for a particular number of epoch with the increase in C the validation error decreased initially and then decreases. This is also intuitive since when C is very small the only emphasis is on regularizer

which makes all weights 0 so as to minimize and thus error is large then error starts decreasing and if we increase the c value to a large extent the complete emphasis is on Loss function which leads to overfitting and hence increased error on the validation set.
Training error vs c plot is shown below for reference.

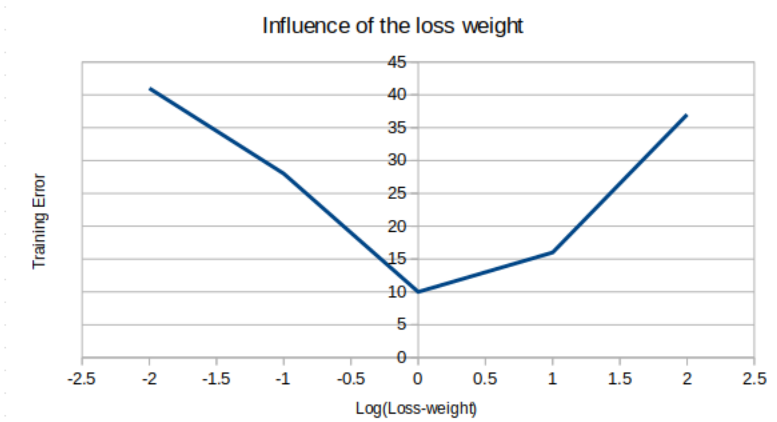

Figure 10: Influence of the loss weight

Note- The x axis is plotted in a logarithmic scale and thus c = -2 is actually 0.01 and similarly other values.

- **Value of C for Kaggle Datasets :** for Kaggle.dataset1 C = 50 , num.epoch = 50 ;
  for Kaggle.dataset2 C = 10 , num.epoch = 400
- **Regularizer used :**for Kaggle.dataset1 regularizer = L2 ;
  for Kaggle.dataset2 regularizer = L2
- **Loss Function Used : -** We used the logistic loss function in both the kaggle datasets to get the optimum value