Deep Learning laboratory Report

Assignment 1

Mohamed AbouHussein

Department of Machine Learning
University of Freiburg
November 2016

Summary

This report is related to the work of implementing a deep neural network. The report is divided into 3 sections as follows: Section 1 represent a simple explanation of the implementation, section 2 represents the designed neural network and section 3, represents the results

Report

1 Neural Network Implementation

The algorithm implementation starts with calling of the neural network function train function `nn.train()`. The output layer encoding is then set to `one_hot_encoding` in case of classification problems and normal in the regression problem cases. Afterwards, the Stochastic Gradient Decent function is been called `sgd_epoch()`. First, The `sgd_epoch()` function computes the prediction `self.predict()` based on the batch input size. Second, the backpropagation algorithm is called `self.backpropagate()`. The `self.backpropagate()` function computes the `output_grad` and passes it to the previous layers using the `layer.bprop()`. The `layer.bprop()` function uses the `output_grad` passed on to it as well as the `input` given to it to compute the `self.dW` and `self.db`. This represent the error gradient of the network weights with respect to the output error. Finally,the `self.W` and `self.b` network weights are update with the computed `self.dW` and `self.db` along with the learning rate.

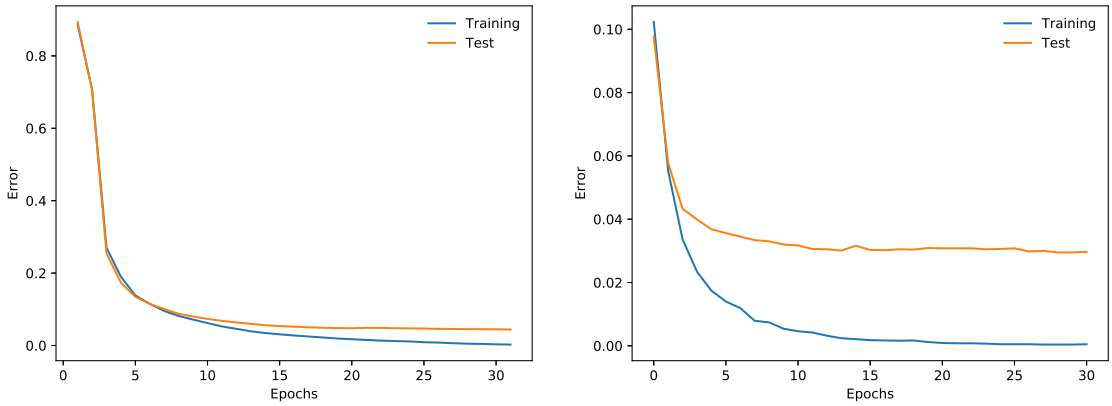
At the end of each loop of the above code flow, the training loss and the training error is computed. Likewise, the validation loss and validation error are computed. The validation set is used to asses the quality of the neural network in predicting an unseen data before.

2 Designed Neural Network

The neural network one designed consisted of 5 `FullyConnectedLayer` layers including the input and the output Layer. First hidden layer consisted of 200 `num_units` and used the `relu` activation function. The second hidden layer consisted of 150 `num_units` and also used the `relu` as an activation. The last hidden layer used a `linear` output activation function for a 10 `num_unites`. Finally, the `LinearOutput` classification function was used as the output layer.

The neural network passed parameters are `learning_rate` of 0.2, `max_epochs` of 30 and `batch_size` of 16.

3 Results and Performance Evaluation



(a) Originally provided neural network design

(b) New neural network designed

Figure 1: Training and validation error for different neural networks

The already provided neural network in the assignment had a similar design to that illustrated in section 2. However, it utilized a `SoftmaxOutput` final layer as well as having a lower `learning_rate` of 0.1 and `batch_size` of 64. In addition, the second and the third hidden layer had 100 `num_units`. The results of the originally supported neural network design is depicted in 1a. The training error and the validation error are very high at the first training iteration starting from 89.3% and drops to 2.6%. The attitude of the validation error almost mimcs that of the training error and decrease rapidly to reach 4.4%. As for the the neural network illutrated in 2 results are depicted in figure 1b. The training error and the validation error starts approximately at 10% and decreases steadily till 4%. Afterwards, the validation test error starts to converge up at 2.83% while the training set error converges to zero. In conclusion, the use of the `LinearOutput` function in the output layer as well as having a smaller `batch_size` and a higher `learning_rate` resulted in a better overall test error as well as a much quicker convergence.