
Data Intensive Systems for Machine Learning

Assignment-1

Upinderjit Singh
ASU ID: 1217042463
Master's in Computer Science
Arizona State University

Abstract

Implementation of Logistic Regression and Deep Neural Network to recognise handwritten digits. Using python libraries TensorFlow and MNIST to implement the Machine Learning algorithms and to get handwritten digits data respectively.

1 Part I: Logistic Regression

1.1 Defining the model

This part of the assignment implemented Logistic Regression. Logistic Regression is a variation of Linear Regression, useful when the observed dependent variable, y , is categorical. It produces a formula that predicts the probability of the class label as a function of the independent variables. If we use perceptron unit with a sigmoid function as a thresholding function and cross entropy as the loss function, then perceptron training gets us logistic regression.

1.2 Outcomes of the Implementation

The average accuracy after 10 runs of Logistic Regression = 92.02. Parameters used:
Learning Rate = 0.01
Batch Size = 128
Epochs = 30

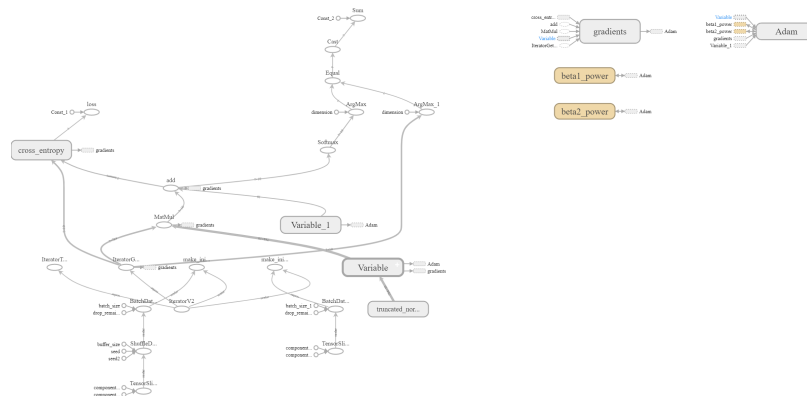


Figure 1: TensorBoard Graph

1.3 Time taken for completion

Understanding the task: 30 mins
Installation of the TensorFlow: 15 mins
Understanding the data and code: 30 mins
Implementation of required section of the code: 45 mins
Understanding and using Tensorboard: 30 mins
Testing and Compiling results: 30 mins

Total Time taken for Task I: 3 hours

2 Part II: Improving Accuracy using Deep Neural Network

Deep Neural Network is an Artificial Neural Network with multiple hidden layers between input and output. The last layer is generally a logistic regression layer which classifies the result. The layers before are just manipulating the input/features to come up with better features to feed the last layer to classify.

2.1 Outcomes of the Implementation

The average accuracy after 10 runs of Logistic Regression = 97.39. Parameters used:

Learning Rate = 0.005

Batch Size = 128

Epochs = 30

Number of hidden layers including output layer = 3

Number of nodes per layer = 500

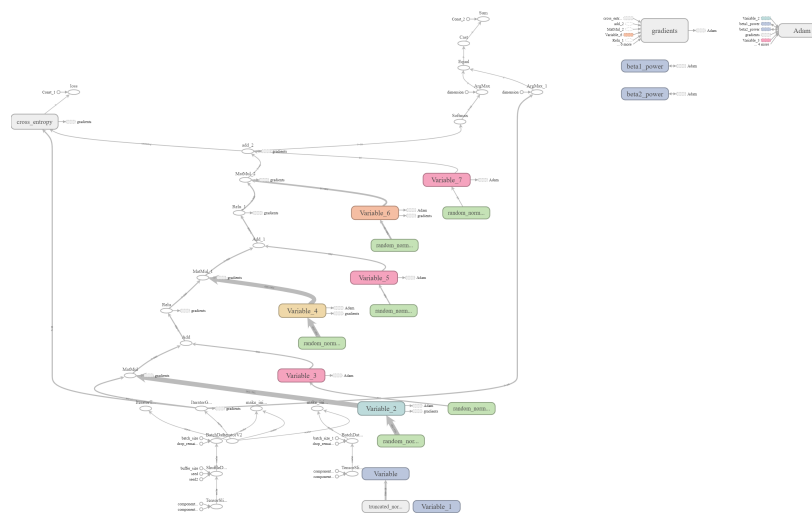


Figure 2: TensorBoard Graph

2.2 Time taken for completion

Understanding the task: 10 mins
Implementation of required code: 4 hours
Adjusting number of layers, nodes and learning rate: 2 hours
Testing and Compiling results: 1 hour

Total Time taken for the assignment: 10 hours and 10 minutes

2.3 Interesting Problems Encountered

As most of the code was already implemented and everything was pretty straight forward. Few problem I faced were:

- Adjusting the number of layers, nodes and learning rate to efficiently get the better accuracy.
- Achieving the same accuracy with same number of epochs.

3 Acknowledgments

I, as the author of this report, would like to thank Professor Jia Zou for this informative and interesting assignment. Looking forward to the next assignments.