Introduction

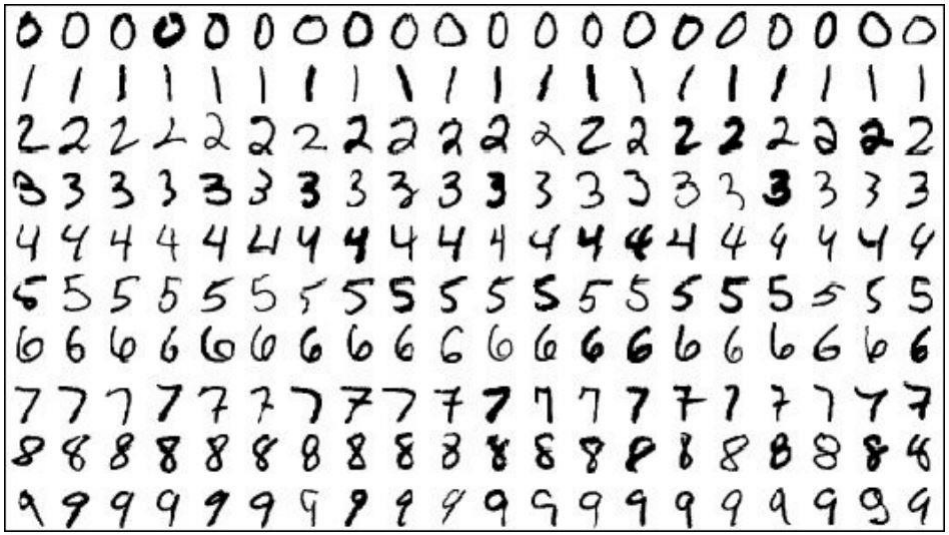
Logistic Regression in Tensor Flow

Objective:

Implementing Logistic Regression in Tensor Flow using MNIST database.

Dataset:

The MNIST database (Mixed National Institute of Standards and Technology database) is probably one of the most popular databases used for training various image processing systems It is a database of handwritten digits. Each image is 28 x 28 pixels, flatten to be a 1-d tensor of size 784. Each comes with a label. For example, images on the first row is labelled as 0, the second as 1, and so on. The images look like this:



Approach:

Approach is the construction looks similar to the linear regression model. Only difference is we have a large data here. We calculated gradient, so here it will be very slow. So, we are going to use batch processing. Tensor Flow has great support for batching data.

To do batch logistic regression, we just change the dimensions of X_placeholder and Y_placeholder to be able to accommodate batch_size data points.

When you feed in data to the placeholder, instead of feeding each data point, we can feed in the batch_size number of data points.

Parameters:

Learning rate = 0.01

Training_epochs=25

Batch_size = 100

Display step=1

Evaluation and Discussion:

We are going to set the model weights initially, construct the model using softmax layer, then minimize the error using cross entropy, find the gradient Descent using gradient descent optimizer and then train the model. We loop over all batches. The batch version of the model with batch size of 100 ran in 1 min using gpu on my pc. We compute average loss and then test the model. Finally, we calculate the accuracy for only digit “0”.

We see that using lower learning rate, we achieve faster evaluation. Iterations will be faster.

Conclusion:

Achieved the accuracy of 91.37% for digit ‘0’ after 25 epochs.