Abstract:

The MNIST database of handwritten digits, available from this page, has a training set of 60,000 examples, and a test set of 10,000 examples. It is a subset of a larger set available from NIST. The digits have been size-normalized and centered in a fixed-size image.[1]

It is a good database for people who want to try learning techniques and pattern recognition methods on real-world data while spending minimal efforts on preprocessing and formatting.[1] We have used these datasets to train a model that will evaluate the test data set image and identify the exact number.

Introduction:

Artificial neural networks[2], usually simply called neural networks, are computing systems inspired by the biological neural networks that constitute animal brains. An ANN is based on a collection of connected units or nodes called artificial neurons, which loosely model the neurons in a biological brain.[2] In this data classification we used the same concept to classify the handwritten digits. Before creating the model we have done some data pre processing to minimize the calculation at the time of training. Train image shape is (60000, 28, 28). Images are in 2-dimension format. We can plot some images with the imshow() method. We can reshape the image from 2-dimension to 1-dimension. Though we lose some data, it reduces the calculation significantly.
Google tensorflow has a built-in feature to create models. We have added 01 input layer, 02 hidden layers and 01 output layer. Neurons in hidden layers are consecutively 256, 128 and activation function is ReLu. ReLu activation function gets the max(0, x) and activates the points. In the output layer sigmoid function is used. We have trained the model 03 times by changing some parameters in layers and optimizer.

Results:

Using Adam Optimizer: Model has achieved 98.85% accuracy in epoch 05. Model evaluation with test data is 97%. This table visualizes more pictures about other results.

Steps	Optimizer	Hidden Layers Count	Neurons	Activation (Output)	Accuracy (%)	Test Data Evaluation (%)
01	ADAM	02	256,128	softmax	98.85	97.61
02	SGD	02	256,128	sigmoid	99.76	98.17
03	RMSprop	02	256,128	softmax	98.78	98.17

Discussion:

From the results we can see the difference between the accuracy results and test validation results. When we use the ‘SGD’ as optimizer and ‘sigmoid’ as activation function in the last layer we get more accuracy than others. Here we are limited to 02 hidden layers. If we extend more layers it will do more calculation and consume more memory.

References:

1. http://yann.lecun.com/exdb/mnist/
2. https://en.wikipedia.org/wiki/Artificial_neural_network