REPORT ON MNIST CLASSIFICATION USING TENSORFLOW

1. Dataset used is MNIST consisting of 60,000 training images and 10,000 testing images of handwritten digits and characters.

2. Used keras to classify MNIST images which is already associated with tensorflow 2.0.

Fig 1: MNIST dataset image

Source:https://en.wikipedia.org/wiki/MNIST_database

Steps for classifying:

1. First the dataset is loaded and splitted into train,test and validation set. The validation set consists of 1000 images and the train set consists of 5000 images.

2. To bring all the images to a single level, I have normalized the images into [-0.5, 0.5].

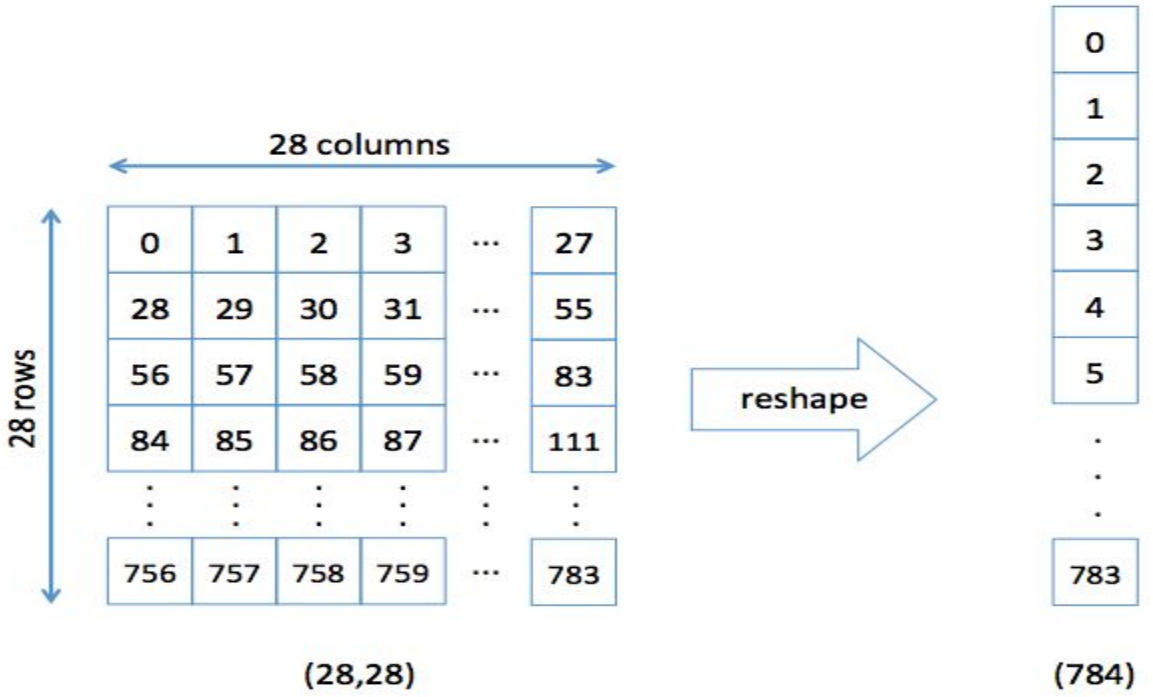

Fig 2: Reshaping of image matrix to one column vector

Source:https://raw.githubusercontent.com/minsuk-heo/deeplearning/master/img/reshape_mnist.png

3. Sequential Model is build with 1 hidden layer with 64 neurons and relu as activation function. And in other case with 32 neurons and relu as activation function.

4. Sequential Model is build with 2 hidden layer with 64 neurons with relu and sigmoid as activation functions. And in other case, 2 hidden layers with 32 neurons each with relu and sigmoid as activation functions.

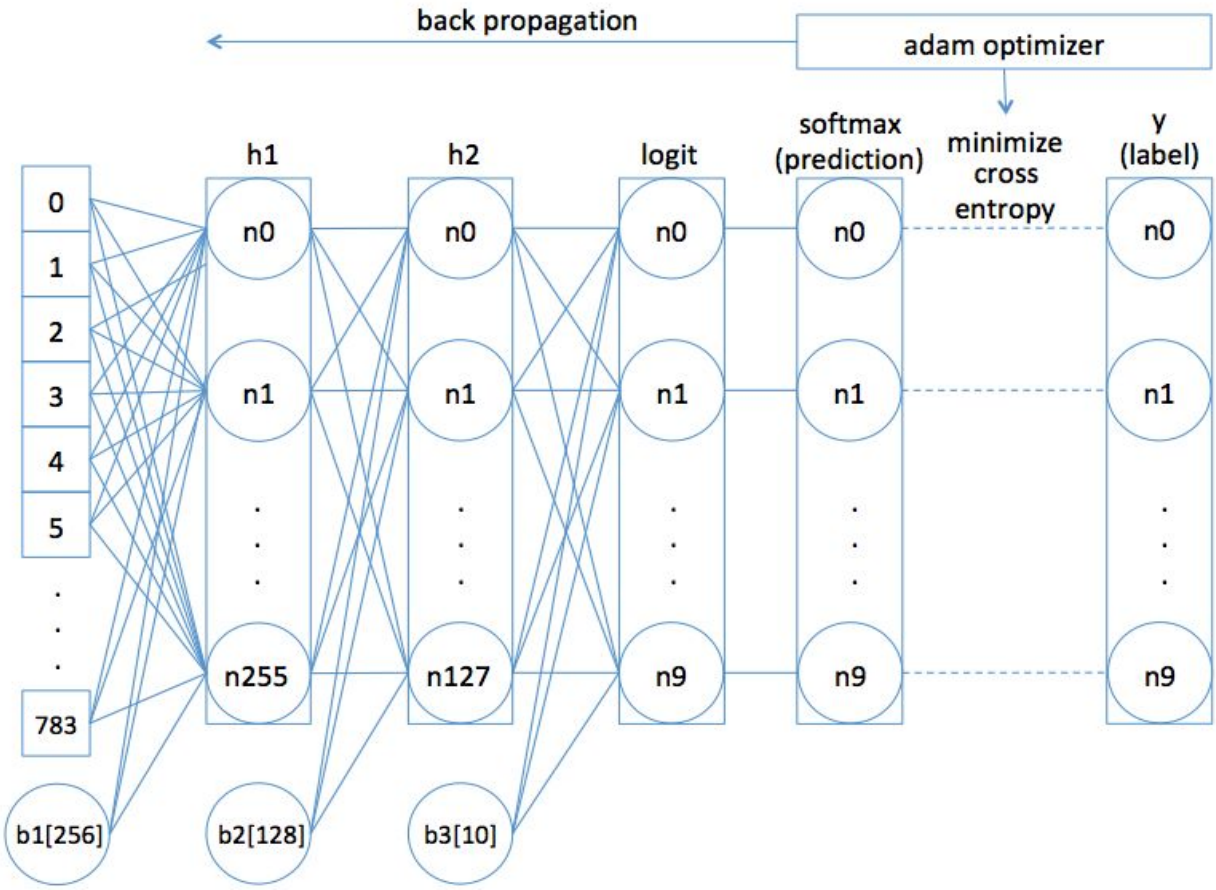

Fig 2: Overview of neural network for mnist data classification

Source: https://raw.githubusercontent.com/minsuk-heo/deeplearning/master/img/simple_mlp_mnist.png

Results:

1. **Case 1: Model with 1 hidden layer with 64 neurons and relu as activation function**

   From the results, I can say that when epochs reach **30**, the validation loss starts increasing, showing that training should stop.
   The test accuracy achieved around **97%**.

2. **Case 2: Model with 1 hidden layer with 32 neurons and relu as activation function**

   From the results, I can say that when epochs reaches to around **30**, the validation loss starts increasing, showing that training should stop.
   The test accuracy achieved around **95.45%**. Since the no of neurons decreases so model performs slightly less accurate as compared to previous one.

3. **Case 3: Model with 2 hidden layer with 64 neurons each with relu and sigmoid as activation functions.**

   From the results, I can say that when epochs reaches to around **20**, the validation loss starts increasing and fluctuating, showing that training should stop.
   The test accuracy achieved around **96.86%**. Increased slightly because of addition of 1 additional layer.

4. **Case 4: Model with 2 hidden layer with 32 neurons each with relu and sigmoid as activation functions.**

   From the results, I can say that when epochs reaches to around **33**, the validation loss starts increasing and fluctuating, showing that training should stop.
   The test accuracy achieved around **95.64%**. Decreased slightly because of removal of neurons and model is thus trained on less no of neurons.