UNIVERSITY OF CALIFORNIA, DAVIS
Department of Electrical and Computer Engineering

EEC181 DIGITAL SYSTEMS DESIGN PROJECT WINTER 2016

LAB 5: HAND WRITTEN DIGIT RECOGNITION AND CLASSIFICATION ALGORITHM

In this lab, you will develop an image processing algorithm to detect and recognize handwritten digits. You should develop your algorithm using MATLAB® software. Hence, in this lab it is not required to write any Verilog code and you do not need to implement the algorithm on an FPGA. The neural network is already trained and its parameters are available on the smartsite. You should download these parameters, import them in MATLAB® and build a neural network using them. Then you should test this network with the provided test set and measure its accuracy. Finally you will implement your algorithm on HPS.

I. SETTING UP THE NEURAL NETWORK

A neural network has been trained to detect handwritten digits. As shown in the Figure 1, this network has 784 inputs nodes and each of which is a value of one pixel of input image. Dimensions of the input images are 28 × 28. Hence, each of them has 784 pixels. As it is illustrated in the Figure 1, the network is fully connected. It means that every single node is connected to all nodes in the next layer. The neural network has four layers, the first layer has 784 nodes, layer two has 400 nodes, layer three also has 400 nodes and finally, layer four has 10 nodes. All of the weights are uploaded to the smartsite in a package named **NN.mat**. Details of the content of this package are described in Table 1. Please download this file and load it in MATLAB®. For testing the neural network and measuring the accuracy, package **testSet.mat**[1] is prepared for you. Please download this package too and load it in MATLAB®. Details of the content of this package are described in Table 2.

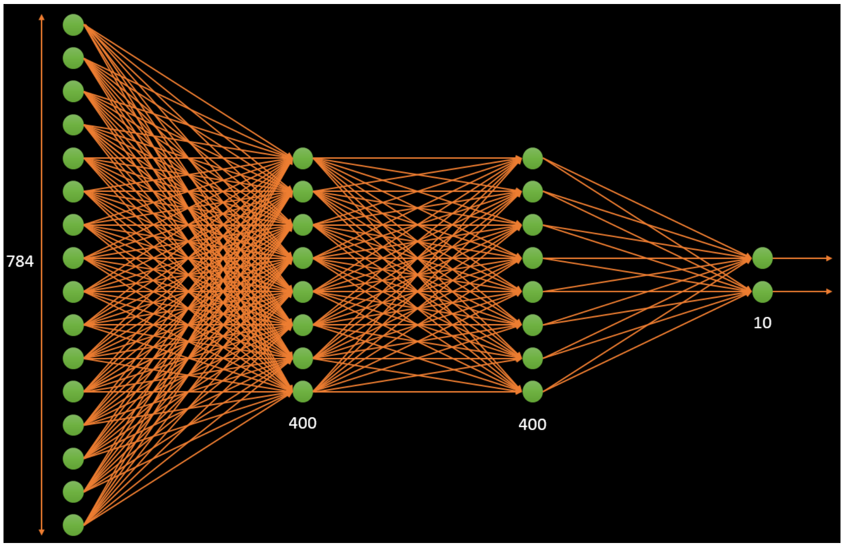

Figure 1. Structure of the neural network

[1] This package is the test set of MNIST dataset. For more information please visit this URL: http://yann.lecun.com/exdb/mnist/

Table 1. Names and descriptions of files in the **`NN.mat`** package

Matrix Name	Matrix Description
finalB1L1	Value of biases in layer one
final B1L2	Value of biases in layer two
finalW1L1	Weights of the first layer
finalW1L2	Weights of the second layer
finalSoftmaxTheta	Weights of the last layer

Table 2. Names and descriptions of files in the **`testSet.mat`** package

Matrix Name	Matrix Description
testData	This matrix contains 10000 images of hand written digits. Exactly one digit per image.
testLables	This matrix contains the corresponding labels of images in testData matrix, e.g. , if the n'th label is 7, it means that the n'th image is a 7.

A. Compute the number of all connections between different nodes in the network.

B. If we use IEEE 754 double precision standard to store weights of connections, what is the size of all weights? What about single precision?

C. Write a simple MATLAB® function that is able to select and show a digit from **testData** matrix. The function signature is shown here:

```
func(x, testData);
```

When you call this function, it should fetch sample number x from the **testData** matrix and show it. For example, func(112), should load sample 112 from **testData** and show it.

D. Use the provided weights and build the network. Then use the provided test data and measure its precision. If the precision in digit recognition is less than 97 percent, you are doing something wrong.

<u>Demonstrate your work to the TA</u>

II. Running the neural network on HPS

1. Rewrite the code that you created in the last section in C language. You have to find a way for reading .mat files. Probably the easiest way is to convert them to text file and read them.
2. Create a system that allows the HPS to access SDRAM. Add all the necessary components, compile and program it.
3. Run your C code on your system. Find a solution for transferring weights and data to SDRAM. Test your system accuracy and make sure it is accurate. Measure the execution time.

<u>Demonstrate your work to the TA</u>