The University of Texas at Dallas
Design and Analysis of Reconfigurable Systems
Group Project (groups of 2) – Handwritten Number Detection through ANNs

1. Objectives

Learn how to build an ANN in HW in this case to recognize handwritten numbers.
Learn how to build complete HW/SW co-design systems on a configurable SoC FPGA

Due Date: December 10 11:59pm (Sunday)

Project Goals

Artificial Neural Networks have shown to lead to very good results in many domains. One of them is image recognition.

The goal of this project is to build a complete HW/SW system where the HPS selects a handwritten number from the MNIST data set, sends the data to the ANN mapped onto the FPGA and then receives back the predicted value as shown below in figure 1.

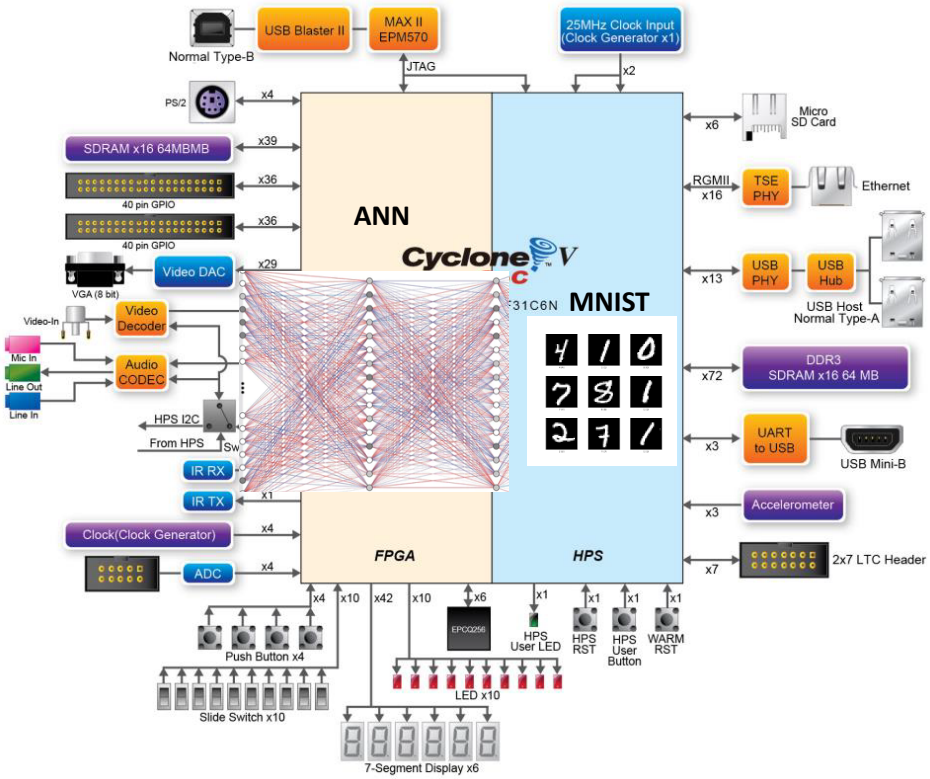

Figure 1. Overview of complete HW/SW system

Figure 2 shows a typical ANN structure composed of :

Input layer: The image is decomposed into individual pixels that are then sent to the input layer of the ANN. One pixel to each neuron in the input layer. For the images used in this case this equates to 784 neurons as each MNIST picture has 24x24 pixels.

Hidden Layer: The ANN can have different number of hidden layers. In this case it should have either 1 or 2 layers with variable number of neurons. Neurons basically perform multiple accumulate operations with different weights obtained during the training phase. Typically, the more hidden layers the more accurate the ANN is.

Output Layer: Finally the output layer contains as many neurons as different responses are needed. In this case 10 as we want to predict the number in the image from 0 to 9.

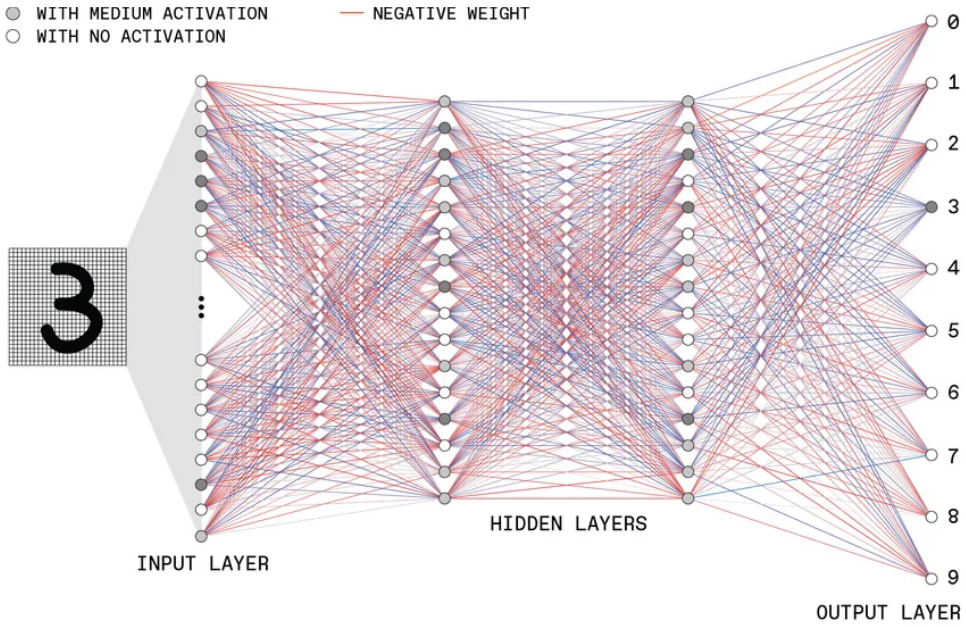

Figure 2. ANN structure composed of input layer, hidden layers and output layer

You can watch the following video to understand how ANNs allow computer to accurately identify in this case the handwritten number.

https://www.youtube.com/watch?v=aircAruvnKk&t=59s

The training input is the MNIST dataset which contains 24x24 pixel images all labeled with the correct value. Each pixel encodes the grayscale of the pixel from 0 (white) to 1 (black). This leads to an ANN of 784 neuron input layer (24x24)

You will use this to build and train an ANN that can detect handwritten numbers from the MNIST database and then synthesize and prototype the ANN on the Terasic DE1-SoC board.

ANN Design and Training
For this you will use TensorFlow is an open-sourced end-to-end platform, a library for multiple machine learning tasks, while Keras is a high-level neural network library that runs on top of TensorFlow.

Training the ANN
1. Importing Dataset
2. Split the Dataset into Test and Train
3. Model Building
4. Train the Model
5. Predicting the Accuracy

Tensorflow implementation:
https://www.youtube.com/watch?v=iqQgED9vV7k
https://www.youtube.com/watch?v=eU0FFjYumCI

Software to design and train ANN:
https://jupyter.org/

Deliverables:

1. Design and train the ANN that can detect handwritten numbers from MNIST in Tensorflow) The minimum accuracy of the ANN should be 90%.
2. Create an ANN in either C for High-Level Synthesis or RTL (Verilog or VHDL) based on the result obtained from Tensorflow. The ANN should have the same structure, weights and biases as the one obtained from Tensorflow. Synthesize it and map it to the FPGA. Show the result
3. Write an embedded C program that runs on the HPS that allows users to either randomly select an MNIST image or asks the user the number to be sent and the HPS then opens randomly an image that contains that number and sends it to the FPGA. The ANN receives the image data, computes its prediction and outputs the prediction to the HPS which will then display the predicted result. The HPS will compare the actual results with the predicted.
4. Create a set of ppts slides summarizing the project from Tensorflow to C/Verilog to FPGA implementation showing that it works (not more than 10 slides excluding cover and conclusion). Create also a YouTube video.
5. **Submit all of the project files in a zipped folder.**

Quality of Report Marking Scheme 20%

Feature	**%**		**Comments**
Power point report	50		Overall professional appearance. Usage of English, captions, etc...
YouTube video	50		Clarity and professional appearance.
TOTAL	**100**		

Working Designs Marking Scheme 80%

Feature	**%**		**Comments**
Tensorflow ANN design and training reaching 90% accuracy	20		
C/Verilog/VHDL ANN design	20		
Embedded C program to read MNIST file and send to FPGA	10		
Fully working system	50		
TOTAL	**100**		