Artificial neural networking

Phase 1 Proposal

Submission Date: 9/21/2016 Wednesday 11:30am Lab (Lab Section 2)
TA: Mingxuan He, Annan Ma

Prepared by:

Ryan McBee
Taylor Lipson
Cheyenne Martinez
Jack Cottom

1. Executive Summary

Modern computers are strong computational devices capable of solving a mathematical and logical challenges previously not capable by humans. Computers are capable of being given a task, a set of instructions for how to do the task, and can do these instructions a few million times a second, far faster than humans can. This advantage might make it appear that computers could easily replace humans, but what computers gain in speed, they lack in their ability to learn and recognize patterns. This is where neural networking comes into play.

Neural networking is a concept by which a computer or device models it's computation based on how the human brain works. This means having a large system of intermediate nodes in between your inputs and outputs, which carry varying weights, and these weights are capable of changing based on learning when an output is incorrect and adjusting for it. This learning step is typically very computationally heavy, requiring millions of iterations in some instances to learn the weights for a single input. This is why we want to use an fpga to create neural networks. The fpga can be optimized to do the calculations faster and more in parallel than a typical computer. To test our implementation of a neural network, we will apply it towards reading handwritten letters and having it figure out which character it is.

Successful design of the proposed accelerator will require the following resources:
- A list of digitized hand written characters represented as 32 x 32 matrix
- Reference Standard Cell Simulation Library for Mapped Design Verification
- Reference Standard Cell Technology Library for Final Design Layout Verification
- Verilog HDL Simulation and Design Synthesis Tool Chain

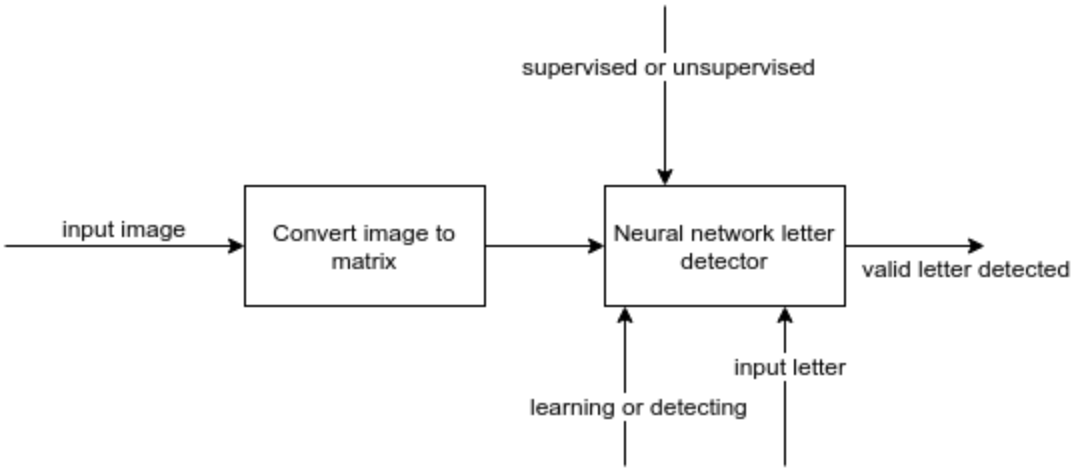
The following document contents will describe:
- Intended usage expectations and constraints for the design, in brief. (Section 1)
- Intended main implementation architecture. (Section 0)

1. Design Specifications

1.1 System Usage

1.1.1 System Usage Diagram

Figure 1: System Usage Diagram



The system described above is an artificial neural network that takes in an image to be analyzed. The system will take in whether it is learning or not, and whether it is supervised as not as it's different modes. It will then take in a preprocessed image and process the image to detect whether or not the image is found.

1.1.2 Implemented standards and algorithms

- Artificial Neural Network weight adjustment
    - Back propagation error detection
        - Creates a random neural network
        - A given known output is provided
        - The system then input an image to be processed, the error of the output is calculated, compared to the known output, and an error is formed.
        - This error is then propagated back into the networks to recalculate the weights.

1.2 Design Pinout

Table 1: Miscellaneous Pinout Table

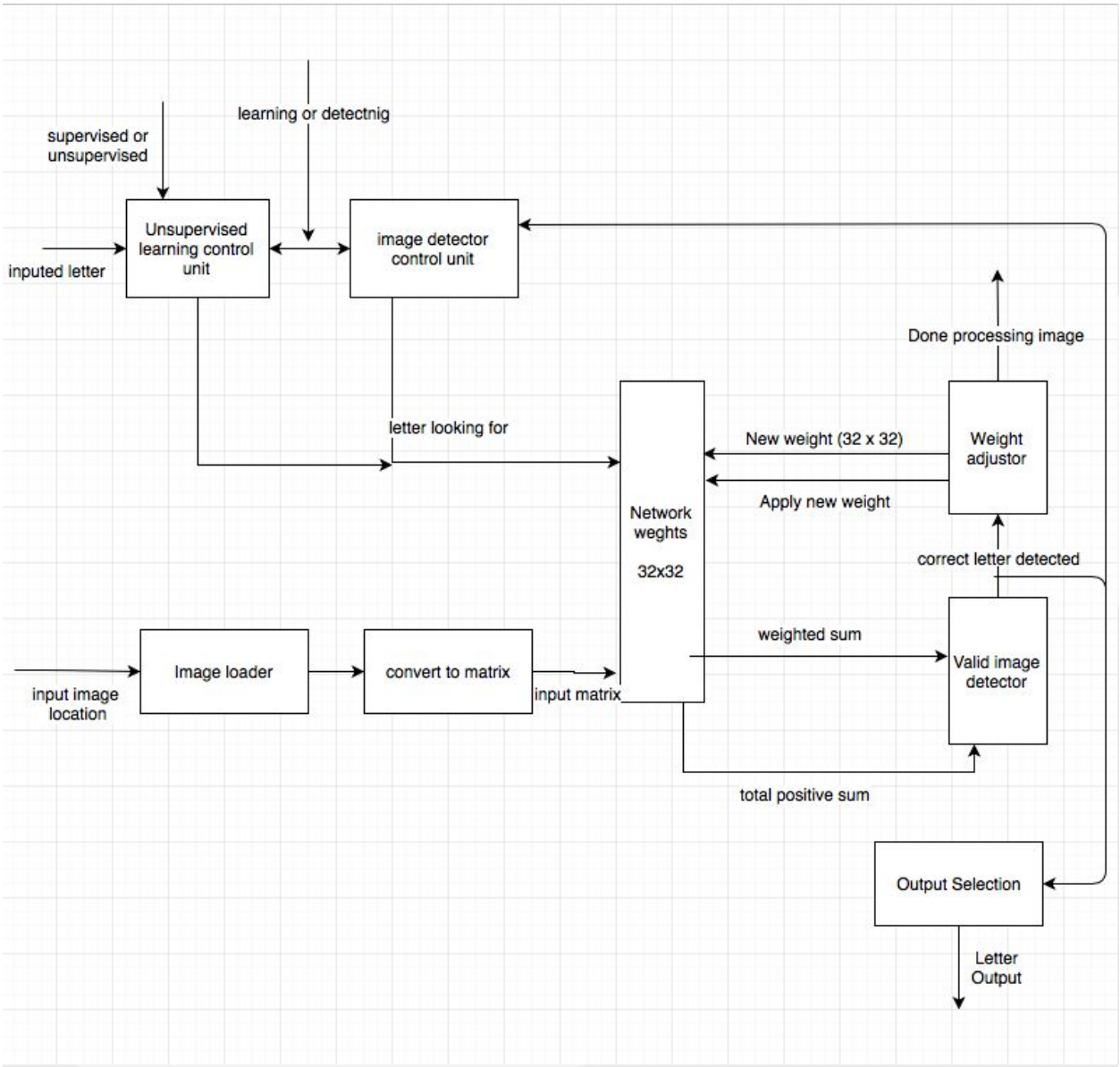
Signal Name	Direction (IN/OUT/Bidir)	Number of Bits	Description
vcc	PWR		Power Pin
gnd	GND		Ground Pin
clk	IN	1	System Clock (speed?)
n_rst	IN	1	Asynchronous Reset. (Active Low)

Table 2: Interface Pinout Table

Signal Name	Direction (IN/OUT/Bidir)	Number of Bits	Description
supervised	IN	1	Asserted when input is guaranteed to be a letter. (Active High)
learning/!detecting	IN	1	Asserted when the weights of the aNN are to be adjusted based on the input data. (Active High)
input_letter	IN	8	Data in ascii specifying the letter that the input image matches.
Input_image	IN	8	Data that is checked by the ANN to determine if it is a valid letter or for the ANN to learn.
done_processing_letter	OUT	1	Asserted high when the system is done processing the image.
letter_output	OUT	8	The letter detected by the system represented in ascii.

3. Design Architecture

Figure 2: Artificial Neural Network, Letter Recognizer Diagram



The implementation for this architecture is depicted above. The input image is loaded and then converted into a 32x32 matrix. The matrix is 1's and 0's where 1 is the ink and 0 is the whitespace. With this image now formatted for calculations, the ANN can learn and match from the images.

There are two forms of learning, unsupervised and supervised. In supervised mode the input image is told to be a correct pattern that matches the inputted letter and will have the weights of the ANN adjust accordingly. In unsupervised mode it is unknown what letter to expect from the input image, so the control unit will go through states of a FSM that sets the letter to be looked for to every possible letter until a match is found or it has tried to match every letter. If a letter is

matched, the weights will adjust accordingly. This is unsupervised because we did not tell the ANN what letter to look for or if it was even a correct pattern for a letter in the first place. Learning can also be set to false so that the weights of the ANN are not adjusted at all.

The network weights are a series of 32x32 matrices where each cell contains a weight correlating to how often that cell is a 1 when matching a specific letter. Using these weights, a weighted sum is calculated to determine how well an image matches. This can be compared to the total positive sum which is a perfect match.

Valid Image block takes the weighted sum and the total positive sum and calculates a Recognition Quotient. If the quotient is above a threshold, the image is a match to the letter and the correct letter detected is sent to the Weight Adjuster and Output Selection.

When the ANN is learning, the Weight Adjuster block takes the data if the letter is correct and recalculates the weights of that letter based on the new pattern that was recognized. It outputs a signal telling the system when it is done processing the new weights.

The Output Selection block is a simple series of registers and that outputs the previous letter detected until the current image being processed has been determined to be a recognized pattern of a letter or not a pattern at all.