Blake Johnson
Perm: 9663980
ECE156a
12/8/17

Homework 7

Intro:
This lab used neurons to model sensors and produce corresponding signals. The neuron itself is relatively simple, and consists of synapse inputs, their corresponding weights, and an axon output when the active synapses' weights reach a specified threshold. When combining these components, we can approximate a function. Before we can approximate the function, the neural network needs to learn. This learning is done by modifying the weights for each synapse. This can be done in any ways, but the most basic way is the way we did it. This involves manually modifying the weights until the new network can produce desired results. This can be difficult because there can be errors in different layers and an error in one of the earlier layers can corrupt an entire network.

Procedure:
The first thing we did was create the neuron, the building block for the neural network. After this we tested the neuron to make sure that it was correct. By manually covering the generic and edge cases, I became more familiar with the behavior of the neuron. After that, we connected the neurons by connecting the axons from the previous layer to the synapses of the next level. My neural network has three main layers. The first layer decodes the input. For example, if the user wanted to get the functional output of 250, they would input 25 (25 instead of 250 because of the size of the increments I used. The scheme follows the pattern 30=3, 170 = 17, etc.). The machine would give weight to each synapse(bit), and the neurons would essentially act as states. Since the first layer can only really determine logic based on a threshold, the second layer determines the specific range. The ranges are the x distance of each block I chose to make on Figure 5. Layer 3 decodes the range into bits that are concatenated separately into a number representing the y-value of the graph.

Testing:
To test the neural network, I looked at internal wires to make sure that they were correct based on a few known inputs. After this I used a for loop to run through all possible input combinations and compared them to the graph to make sure that they were correct.

Conclusion:
This lab demonstrated the power of neural networks once they have learned how to do their intended function. They quickly calculated an obscure function by interacting with other neurons and the inputs. An issue I ran into in this lab was indexing the memory bank. Because I created my neuron in SystemVerilog, I was able to take advantage of packed and unpacked arrays and use these clean implementations to minimize weight and threshold instantiation. However, in doing so, I messed up the formatting of the intended memory bank to the point where I would have to possibly break the logic of my neuron to fix it. Despite not having a bank, my code is clearly labelled and shows where my instantiation takes place.


Wave
File Edit View Add Format Tools Bookmarks Window Help
Wave - Default
/NeuralNetworkSV_...
/NeuralNetworkSV_...
The top wave form is the x-axis input
Example: The input is 25 when the x-axis value is 250
because i used increments of 10.
The bottom waveform is the y-axis value of the function.
Now
Cursor 1