Neural Network Hardware Implementation for Handwritten Digit Classification

Slayter Teal
A20131271
ECEN 4303
Fall 2021

Introduction

A neural network is a computer network, whether hardware and/or software, that mimics neurons within a brain to solve complex problems. In simple terms a "neuron" is something that "contains a number". This number is the representation of some specific data point, or a characteristic of the information being fed into the neural network. For our purposes the input neurons correspond to a greyscale of the 784 pixels that make up our handwritten image.

A neural network is broken into three distinct sections, the input layer, the hidden layer, and the output layer. Usually, every neuron has a connection to every following neuron The input layer is the section that takes our data (an image in our case) and breaks it down into the corresponding values assigned to portions of the image. For number classification the input layer contains the greyscale values in the image we want the system to interpret. The hidden layer is the portion of the neural network that takes the values from the input layer and "parses" the given input data–it is important to note that there can be multiple layers within the hidden layer. The hidden layer makes different decisions based on the values provided by the previous layer and outputs its information to the next. The calculations of the hidden layer are then presented to the output layer where a decision is made. For handwritten number classification the output layer has 10 "neurons" each representing a number between 0-9.

The information gathered by the input layer, in our case a greyscale value, is nothing but matrix of seemingly random values such that with this information alone our neural network cannot decide. This is where the hidden layer comes into play. As mentioned previously, the hidden layer is the portion of the network that takes our input data and parses it into an output. Each neuron in the hidden layer is connected to all of the previous layers' neurons. The receiving neuron takes in the parameters and assigns a weight to the parameter in hope that receiving neuron is expressive enough to find the pattern or shape that the neuron is looking for. The receiving neuron then establishes a weighted sum of all the parameters:

$$w_1 a_1 + w_2 a_2 + \cdots + w_n a_n$$

Where w corresponds to the weight and a corresponds to some input parameter. One problem is that, with a weighted sum, it is possible to get a result that varies wildly. To counter act this the weighted sum is fed into a function that will “squish” the weighted sum between 1 and 0. This function is referred to as a Sigmoid function σ. Additionally, a bias

is introduced to determine when the resulting sum is meaningfully active to the result of our network. In total the equation that determines a neurons value can be written as such:

$$a^{(1)} = \sigma(\mathrm{W}a^{(0)} + \mathrm{b})$$

It is vital that the weights and bias of a neuron are train correctly in order for the network to produce the expected output.

Project Timeline

Due Date	Task	Description
9/18/2021	Initial Report	Outline the underlining theory of the project as well as the sprints for the project.
10/2/2021	Implementation of Software based Neural Network	Design the software portion of the project and begin the train to obtain the weights and bias.
10/9/2021	Complete Training of the software implementation	Successfully complete the training of the network. Work towards getting the weights and bias recorded.
10/23/2021	Begin researching hardware implementation.	Given that the Verilog simulation will likely take the bulk of my time, research will be needed to approach the hardware implementation.
10/30/2021	Midterm Report	Provide a generalized progress update.
11/6-30/2021	Verilog Implementation	The month of November is dedicated to designing and implementing the Verilog simulation.
12/7/2021	Final Report	The final detailed report of the project contains the results of the Verilog simulation as well as the overall neural network classification hardware implementation.