Synthesis of a Asynchronous Multilayer Perceptron on an FPGA

Fabian Franz

October 2020

Contents

1 Introduction

This project has the claim to design a low-power neural network on an FPGA. To do so, the next sections give a brief introduction to the basic principles of how such a neural network can be modeled.

1.1 Perceptron

A perceptron describes an analog model of a biological human cell in the computer domain. This perceptron can be described with the following graphical and mathematical expressions:

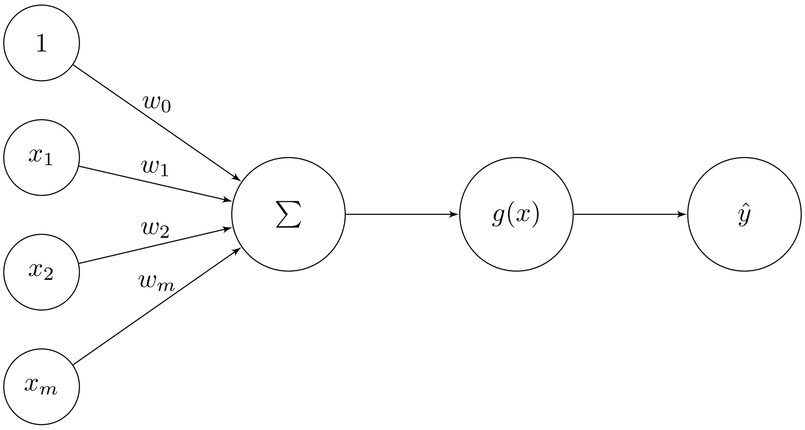

Figure 1: Model of perceptron

$$\hat{y} = g\left(w_0 + \sum_{i=1}^{m} x_i w_i\right) \tag{1.1}$$

With:

g = Activation function
w_o = Bias

In vector form:

$$\hat{y} = g(W_0 + X^T W) \tag{1.2}$$

With:

$$W = \begin{pmatrix} w_1 \\ \vdots \\ w_m \end{pmatrix}, \text{X} = \begin{pmatrix} x_1 \\ \vdots \\ x_m \end{pmatrix}$$

As seen, in a conventional model of a perceptron every input of an layer is multiplied by a weighting. It can be seen, that the implementation of such a perceptron can be handled by given hardware architecture like GPU's or matrix multiplier, because of the possibility to proceed every input matrix and their associated weight matrix independently.

1.2 Activation function

The activation function of a perceptron has the purpose to determines the behavior of the perceptron in response to external stimuli. There are different kinds of activation functions which can be used for different purpuse.

2 Design Overview

The hardware of this project consist of a 32-bit microcontroller named SAM D21/DA1 and a Cyclone 10 FPGA. Both chips are placed on a board, so called Arduino Vidor 4000. This board comes with an USB interface, which will be the communication between frontend (PC) and the executing hardware. The figure 2 show the most abstract hardware model. This model show the mentioned frontend and hardware, but also the "Top Level Resolver", which distribute the data to either the storage, which holds the data to be proceed, or the new settings for the single perceptron activation functions. In the concret model, the address line has a size of 11-Bit and the data line a size of 4-Bit. This results in a maximum count of 256 perceptrons in 16 layers, if one perceptron of the incomming layer is connected with all perceptrons of the previous outgoing layer. The load line tell the single perceptrons, that they can storage the new information after addressing. The address line needs to be a 11-Bit one, causing the fact, that every oerceptron needs 5 informations to store. More related to this will be described later. For simlifity purpose, the clock and reset signals for every single module is not shown in the figure.

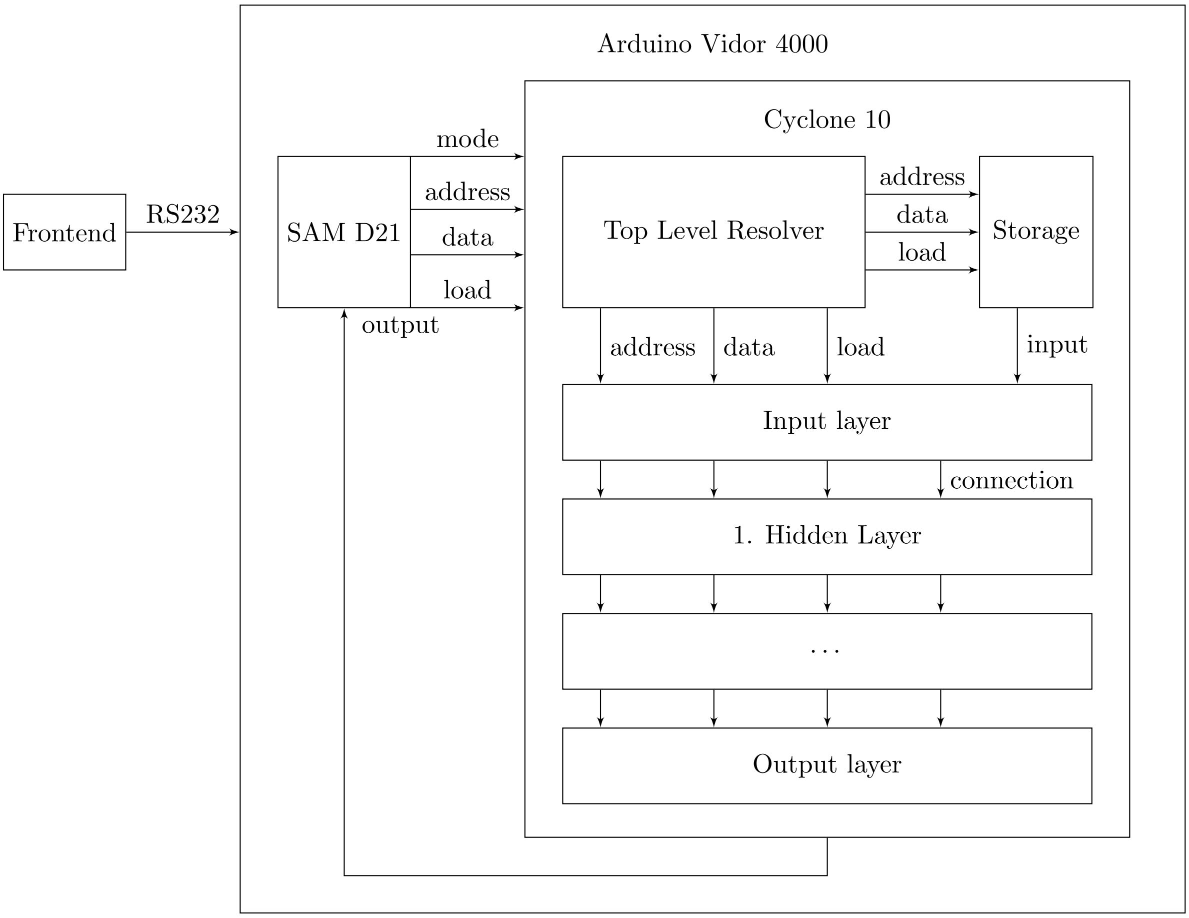

Figure 2: Top Level Hardware Model

Every layer of the hardware model consists of a ”Layer Resolver”, which takes care about the signal distribution to every single perceptron. This has the advantage, that the data lines inside the single layer model can be recduced. In the concrete hardware model, the 11-Bit address line go through all the layer resolvers, which split them in a 7-Bit address line, which is connected to every perceptron in the particular layer. The data line stay a 4-Bit and the load line a 1-Bit connection and are simply y-splitted in the resolver.

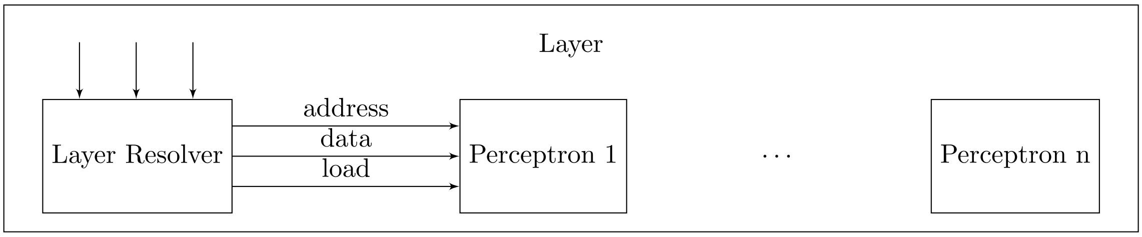

Figure 3: Layer Level Hardware Model

The perceptron itselfe consist of a two state input control instead of float point calculations. That means, that the connection from one perceptron output to an other perceptron input can only be active or non active. Furthermore, the second parameter inside the perceptron is the activation function, which is basically the summation over a count of input ”high” signals. If the summation is greater than a given value, the output of the perceptron is pulled ”high”.

3 Implementation

4 Validation