(Article: 37)

AN ANALYTICAL NEURAL NETWORK FOR ARITHMETIC LOGIC UNIT OF MICROPROCESSORS

Mainak Basu[1], Abhishek Pandey[2], Abhijit Mustafi[3]

[1&2]Dept. of Electronics & Communication Engineering, Birla Institute of Technology, Mesra, Ranchi, India
[3]Dept. of Computer Science Engineering, Birla Institute of Technology, Mesra, Ranchi, India
[1]e-mail: mainak.basu87@gmail.com

ABSTRACT

In this article, we describe a new approach for analog computation using Deep Learning Neural Networks to perform multiple types of mathematical operations within one machine cycle. the weights and biases of the pre-trained ANNs can be used to fabricate the network in hardware either using a FPGA system or discrete components. The trained network has shown the potential to execute simple and complex mathematical operations alongside basic signal processing within an acceptable error limit, without retraining or reconfiguration.

INTRODUCTION

Digital computation offers significant advantages over analog computation, as any mathematical problem can be broken into a set of Boolean primitives that can be operation on by the Arithmetic Logic Unit (ALU) or the Floating Point Unit (FPU) of the microprocessor. Several microprocessors/microcontrollers implement also implement several modifications to operate on more complex mathematical functions in a reduced period of time. Despite these advantages, analog computation offers a greater speed of operation, using reduced number of components but requires application specific circuits for each operation.

INTRODUCTION

In this article, we propose a radical redesign of the Arithmetic Logic Unit (ALU) or the Floating Point Unit (FPU) of the microprocessor/microcontroller. Unlike conventional circuits which use discreet logic sections to perform a specific task, we propose the use of a hardware based Artificial Neural Network (ANN) circuit, which can perform several mathematical operations without retraining or changes in components. ANNs have been fabricated in hardware to perform a variety of tasks including adaptive computation [1] and increasing the interconnection density network-on-chip routers [2]. There have also been several attempts to build mathematical processing circuits using Neural Networks [3, 4]. Several articles have also been fabricated to simulate the properties of artificial neurons [5 - 7]. In order to retain the training and adaptive capability of a neuron, the networks have been developed with FPGA kits, which enables a simplified implementation of the ANN, but still uses digital logic to employ the mathematical operations. There have also been several attempts to develop standalone circuits using more conventional hardware [8 - 10].

In the present work, the Neural Network is trained in an application specific manner, using a data pre-processing that is implemented before the training takes place, hence dimensionally increasing the data that is to be input into the ANN. The construction of the ANN is made of standard artificial neurons utilizing tan-sigmoid and linear transfer functions in the hidden and output layers respectively. This also enables simpler circuitry, using a reduced number of components. Once the network is trained, the weights and the biases can be used to construct the electronic circuits for the individual neurons. Simulations have been performed to test several deep learning ANN architectures with two hidden layers employing a maximum of 20 neurons per layer. A single ANN having two hidden layers, trained with the application specific method as mentioned above to perform five mathematical operations (addition, subtraction, multiplication, division and power) alongside denoising of sine, cosine and gaussian profile signals. It has also been necessitated to develop a appropriate analog logic that can be used for practical implementation on a circuit. Efforts are currently underway to develop a practical implementation of the Neural Network circuit.