Analog Feed Forward Neural Network Neuron Design and Implementation

Daniel J. B. Clarke
Gildart Haase School of
Computer Sciences & Engineering
Fairleigh Dickinson University
Email: danieljbclarke@gmail.com

***Abstract*—With Machine Learning methods being employed for many different use cases from speech and handwriting recognition to autonomous vehicles in big data analytics, the demand for high performance learning continues to increase. Artificial Neural Networks are widely used in software because of their theoretical simplicity and versatility for tackling many types of problems. While GPU acceleration is helping industry meet processing demand for network training, an analog integrated circuit that performs the training required by these networks may prove extremely valuable for scaling up neural network performance and bringing larger and faster networks to smaller devices.**

A single neuron of an analog feed forward neural network is designed in this paper with two input synapses and one output. The ability of the circuit to learn logic functions, AND, OR, and XOR is tested. Given the nature of artificial neural networks, such a network would have the ability to scale upwards to provide NxM-layer feed-forward neural network capabilities. Utilizing GPIO pins on a Raspberry PI, software is written to interface with the circuit and performance is benchmarked against an equivalent software-based neural network.

I. Introduction

Neural Networks model the operation of a biologic brain. Used extensively in machine learning, a simplified model of these networks, feed forward neural networks, have proven useful for a number of problems including speech and handwriting recognition. Neural networks operate like binary classifiers driven by linearly separated features across multiple dimensions. The limitation of linear separability, though initially thought to be too great to warrant any attention to the algorithm has become less of multi-layer networks. [**?**]

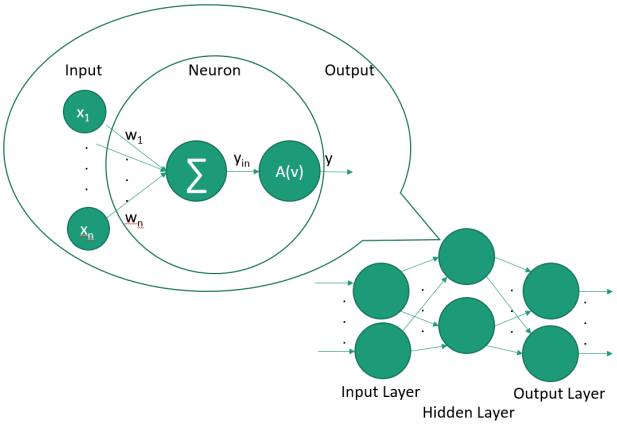

Fig. 1. Feed Forward Neural Network

The sigmoid function is often used in the field of machine learning due to its uniformity and differentiability as an activation function. The activation function receives the sum of the weights times the inputs and turns it into the output y_j. $\tanh(x)$ is a bipolar sigmoid function. It results in a training algorithm known as delta rule of the form:

$$\Delta w_{ji} = \alpha(t_j - y_j)x_i \tag{1}$$

Where Δw_{ji} is the change to be applied to a single weight in the network, α is a small learning rate, t_j is the expected output, y_j is the network output, and x_i is the input.

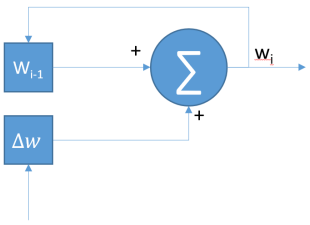

Fig. 2. Weight Circuit Block Diagram

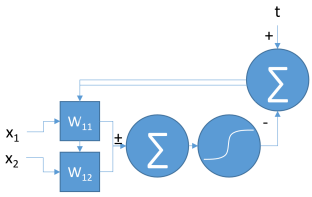

Fig. 3. Single neuron

A. Discussion

Graf et. al. suggested the construction of an analog electronic neural network circuit as far back as 1989 focusing on potential interconnected architecture. A balance needed between resolution and complexity is suggestedworrying about the limitations of electronic hardware above modeling the biological function precisely. [2] Algorithms have since been developed that are being widely used in software.

Nishitani et. al. explore back-propagation operation with analog neural networks while using FeMEM, a ferroelectric memristor, for the synaptic weights [3]. The primary issue they dealt with is the effect of hysterisis on the conductance of their memristors. A focus is placed on mitigating the effects caused by the highly non-linearity of those devices.

Rosenthal et. al. build an analog neural network utilizing a structure very similar to the one in this paper, but they also use memristors for the synaptic weights. They experienced the effects discussed in [3] but still managed to get some results. While they found ten-fold reductions in runtime benchmarked to a MATLAB based implementation, MATLAB itself is about ten times slower than a typical neural network implementation in C, these results are therefore, not too impressive.

Tensor Processing Units by Google are ASICs (application-specific integrated circuits) designed to do tensor calculus operations, used for Google TensorFlow, their framework for building and training large neural networks. They operate at lower precision to get better efficiency. While they are dedicated chips for neural network processing that acheive a ten-fold performance increase above using GPUs, they still rely on digital approaches for neural network training. [1]

In this paper, we construct an analog neural network and fall back to a less precise but much more predictable form of synaptic weights by charging capacitors as suggested in [2], but by using bipolar weights instead of digital memory. Using analog internally and a digital interface we achieve the power of analog learning while still retaining the ability to interface with existing digital device drivers.

II. Objectives

To validate the feasibility and performance of our approach we design and construct a single analog neuron with 2 inputs and 1 output, we test it against logic gates in hardware and benchmark it against a software based implementation.

The proper functioning of the circuit can be verified by ensuring the error tends to zero as training occurs. The following signals can be provided to the circuit for the logic functions learning.

In an effort to illustrate the circuit's use as hardware acceleration, a raspberry pi will interface with the circuit over its GPIO pins. It will drive the circuit to learn AND, OR, and XOR and will perform the same analysis in software as a comparison.

III. Design

A. Binary Product

Recognizing that y_{ij} is a summation of weights multiplied by the input x_{ij}. Assuming x_{ij} are binary, we can find the binary product

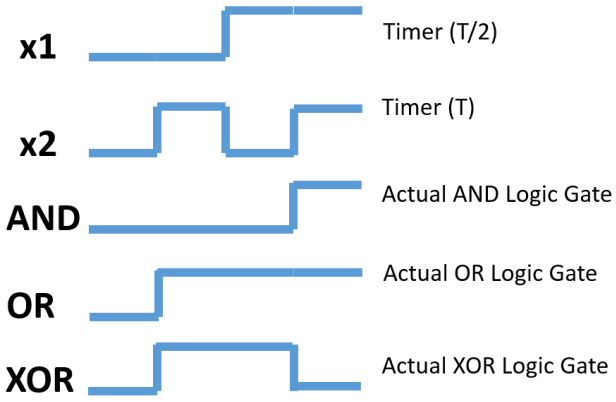

Fig. 4. Test Signal

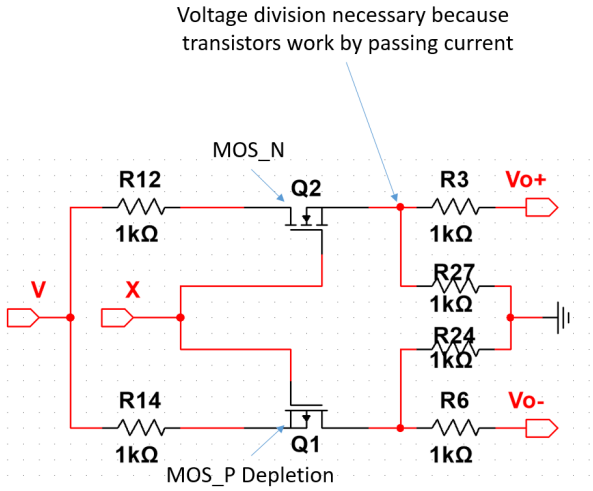

Fig. 5. Bipolar Binary Multiplication

by passing w_{ij} to the positive or negative side of an op amp based on x_{ij}.

B. Synaptic Weight Control

Storing the value of w_{ij} as V_c due to charge in a capacitor, the back-propagation equation can charge or discharge the capacitor until the error becomes zero according to the delta rule. α is dictated by the natural time constant $\tau = RC$ of the capacitor-resistor circuit.

C. Training

The network can be trained by providing inputs x_{ij} and targets t_{ij}. This can be done in hardware initially using two square waves, one

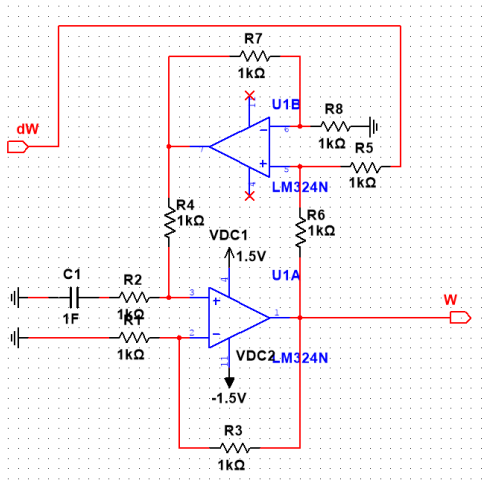

Fig. 6. Weight Control Circuit

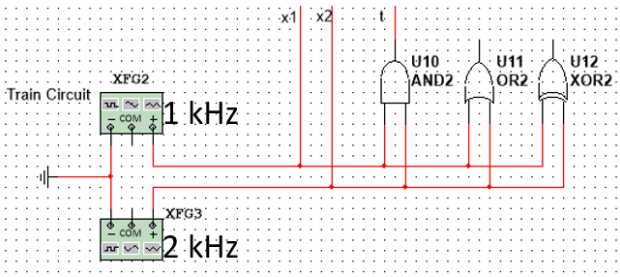

Fig. 7. Test Circuit

with twice the frequency as the other connected to the logic function itself.

D. Activation function $tanh$

While hyperbolic tangent isn't a simple function to implement, a diode clipper circuit is a decent estimator of the function with very little overhead.

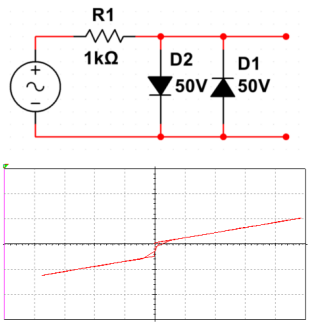

Fig. 8. Activation Circuit (Diode Clipper)

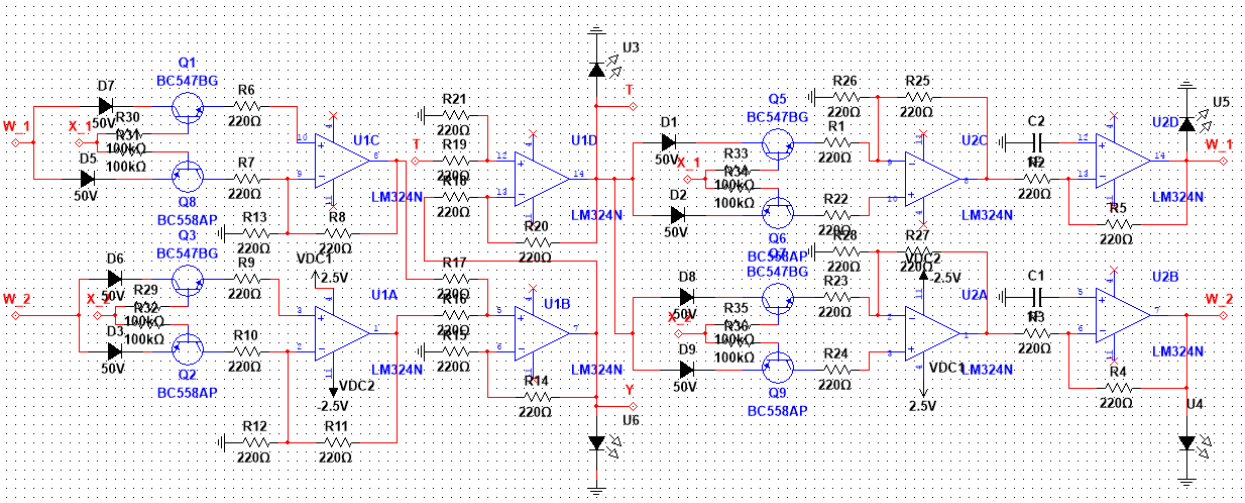

Fig. 9. Full Circuit

IV. Conclusion

A novel approach for analog neural network construction is designed and partially implemented. With individual modules working, full circuit results are expected upon completion of the full circuit. Falling back to a capacitor based analog memory approach as in [2], our approach depends only on components that can be easily aquired as opposed to costly memristor-style memory components so prevalent in current research [3], [4]. The approach taken is inherently scalable–this scalability will be demonstrated in the future with the construction of a 3 neuron network that can properly learn XOR.

References

[1] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng, "Tensorflow: A system for large-scale machine learning," in *Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation*, OSDI'16, (Berkeley, CA, USA), pp. 265–283, USENIX Association, 2016.

[2] H. P. Graf and L. D. Jackel, "Analog electronic neural network circuits," *IEEE Circuits and Devices Magazine*, vol. 5, pp. 44–49, July 1989.

[3] M. Ueda, Y. Nishitani, Y. Kaneko, and A. Omote, "Back-propagation operation for analog neural network hardware with synapse components having hysteresis characteristics," *PLOS ONE*, vol. 9, pp. 1–10, 11 2014.

[4] E. Rosenthal, S. Greshnikov, D. Soudry, and S. Kvatinsky, "A fully analog memristor-based neural network with online gradient training," in *2016 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1394–1397, May 2016.