Ann Based Design of A Rapid Battery Charger

Patel, Himanshu K. (Sr. Lecturer)
Institute of Technology, Nirma University
Hkp_ic.it@nirmauni.ac.in

Jadeja, Rajendra B.(Assistant Professor)
C.U.Shah Engg. College, Wadhwan
rbjadeja_2001@yahoo.com

Abstract*-- *This paper presents an application of ANDEng, an Artificial Neural Network Design Automation tool for the design of a rapid Nickel Cadmium battery charger. Rapid charging needs high charge current of the order of 8C where C represents the capacity of the battery. The battery charger needs an intelligent strategy as the charging has to be controlled to avoid damage to the battery due to rise in temperature caused by high charging currents. The temperature control assumes significance as the rise in temperature is exponential after initial charge. The paper also presents the analog implementation of neurons with log sigmoid and linear activation functions using discrete components.

I. INTRODUCTION

THE present day world presents a scenario of high mobility. Due to this, the recent years have seen fast pace development in the field of mobile technologies and hike in the use of mobile devices such as laptops, cell phones etc. The development in this sector infers a growth in the electronics industry and is directly dependent upon the source of power for these devices i.e. the battery. The boom in the electronics industry has revolutionized the field of battery management, which includes regulating the charging, protection, and monitoring of the battery [1]. Managing the battery resources properly has become a necessity because the batteries limit the performance of the electronic devices. Keeping in view all the above points, the need to accentuate research in the field of battery management is clear enough.

The lifetime of a battery is decided by its charging and discharging cycles. Earlier this was the total lifetime of the battery but now with recharging as an option, it is not so. Conventionally, batteries were charged using a constant current source. The continuous monitoring of battery charge was impossible. Thus, at times the batteries were overcharged or under charged. Also, since the temperature rises with the flow of current, the batteries also got damaged because of overheating. This called for the need to devise a system, which could decide for itself, the amount of charging current, which needs to be supplied to the battery at a particular combination of inputs. For a device to be able to make decision, it has to be an intelligent one. Any device can be converted into an artificially intelligent device just by applying any technique by which it is able to make decisions for itself for e.g. Fuzzy, Artificial Neural Networks etc.

This paper presents a Nickel Cadmium battery charger, which has been rendered intelligent by the application of Artificial Neural Networks. Our battery charger continuously monitors the battery and as the input varies, pumps varying charging current into the battery accordingly. Since, any battery is most affected by the increase in temperature as the charging continues; we have taken temperature and temperature gradient as the two input factors. The battery charger would be constantly monitoring the battery for any changes in temperature and temperature gradient. As soon as any change occurs, it manipulates the output current being pumped accordingly.

This charger increases the life of battery by maintaining proper charging. It also charges the battery to 66.6% of its maximum value within the first 5 minutes of charging time and the remaining capacity of 33.33% can be charged in another 5 minutes at lower charging rate. It has been observed that during the first few minutes of providing the charge, the battery temperature falls down and as such, we can apply full blast charging current to it without the fear of getting damaged and after that as the temperature goes on increasing, the charging current also needs to be reduced to prevent over heating of the battery. Thus, at moderate atmospheric temperature the charger would charge a completely discharged battery within 15minutes or so. If the temperature is higher, then slightly more than 15minutes time would be sufficient. So, we consider temperature & temperature gradient as the two control parameters for the charger.

II. ARTIFICIAL NEURAL NETWORKS

The concept of artificial neural networks is rooted deep into the recognition that though the human brain performs the functions about a million times slower than the digital computers, yet the human brain is more efficient when it comes to performing a complex set of the tasks. ANNs have been proved to be universal function approximators. These can learn and reproduce any complex function which makes them very useful in intelligent systems [2-7].

Any neural network can be represented as a combination of input layer, hidden layer and output layers. Each layer further contains one or more neurons of the same type. Here, p1 and p2 are the two inputs, w11, w12, w21, w22 are the respective weights from the input layer to the hidden layer.

2.1 *Layer Architecture:* The number of neurons in the input layer equals to the number of inputs to the circuit and the number of neurons in the output layer equals to the number of outputs of the circuit. In the hidden layer, the user decides the number of neurons usually but our software ANDEng [8] does the optimization for the user.

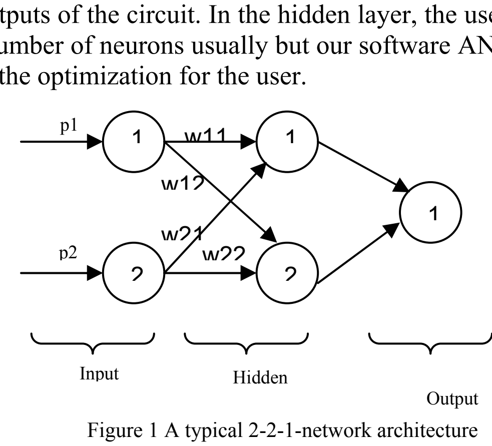


Figure 1 A typical 2-2-1-network architecture

The design of the network is such as to exploit the most beneficial property of the neural networks i.e. parallelism. The neuron in each layer takes the input from their common input line, multiply it with their corresponding weight and add the bias value. This sum is passed on to the activation function for that layer. The output of the activation function marks the end of one layer and is passed on to the next layer and the same process follows invariably.

2.2 Neuron Architecture

Various types of neurons can be classified depending upon the activation function they are employing. The user depending upon the kind of problem being implemented decides the kind of neuron in a network. The outputs of the previous layer neurons multiplied by their corresponding weights enter serially into the neuron along with the bias value for the neuron. The supporting algorithm provides the weights and the bias values.

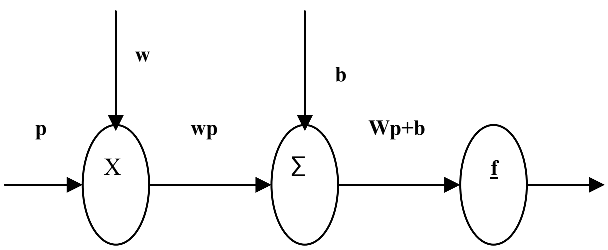


Figure 2: Structure of a Neuron

Here, w is the weights required, p is the input to the neuron, b is the bias value, f is the activation function which is actually the deciding factor for the output we get.
The design of any Artificial Neural Network involves the following considerations:-

1) The network topology
2) Number of layers in the network
3) Number of neurons in each layer
4) Type of neurons in each layer

III. Implementation

The design of any application in Artificial Neural Network involves following steps:-

1) Fix a single neuron structure,
2) Train the ANN and evaluate network performance
3) If performance is as desired then go to step 5.
4) If performance level is below the desired value then modify the network structure (network structure is first modified by adding the number of neurons one at a time up to a certain number followed by adding a hidden layer) and Go to step 2.
5) Hardware Implementation.

We have used ANDEng [8] a design automation tool to design an ANN to meet a particular performance parameter.

3.1 Network Topology and Training

For training our network, we have used the dataset as given in [9]. ANDEng first selects single hidden layer structure, then trains the network and if the required performance is not achieved, the network structure is changed by adding neurons in the hidden layer. In case, the required performance is still not met, the number of hidden layers is increased and the network is retrained using feed forward back propagation algorithm. ANDEng uses gradient descent with momentum. This software not only trains the network but also optimizes the number of hidden layers depending upon the number of iterations and the mean square error. The output of this software gives us – the number of layers in the network and number of neurons in each layer. Also, the weight and bias values for each neuron are given. The type of network and type of neurons in each layer is to be decided by the user himself depending upon the specifications of the problem. Having gathered all the above information, the system is now ready to be implemented in hardware.

For this data set, the output of the software gives a 4-1-1 network, which implies we have a network with two hidden layers and one output layer. The type of neurons for the hidden layers for this application is logsigmoidal and linear neurons for output layer. The first hidden layer consists of 4 neurons and second layer consists of a single neuron. The output layer contains a single neuron of linear type.

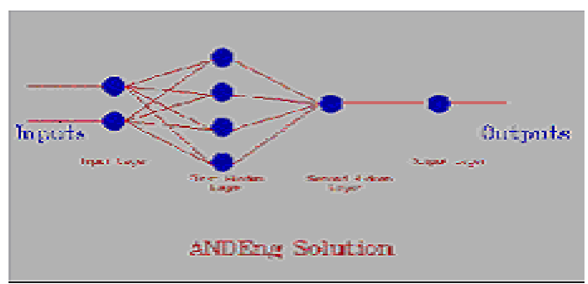


Figure 3

Hardware Implementation

The software-trained network would do nothing for the designer except providing a test bench for his concepts. With the application area of ANN ever widening, the need to realize the concepts in hardware is vivid.
The main concern here is to design the neurons involved in the network and the weight control circuit because these two factors actually decide the output of a network. This section presents the design of Logsigmoidal and linear neurons.

3.1.1 Design of Log sigmoid neurons

The sigmoid transfer function shown below takes the input, which may have any value between plus and minus infinity, and squashes the output into the range 0 to 1.

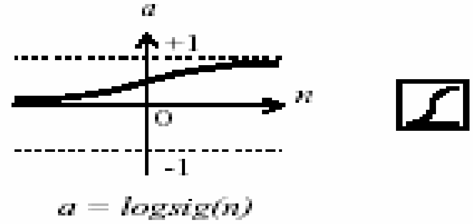
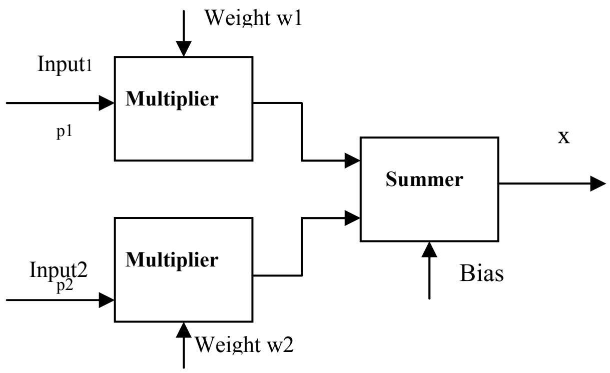


Figure 4

The transfer function

$$f = 1 / (1 + \exp(-x))$$

Where, x = wp + b, i.e. the summation of the product of all inputs to the neuron with their respective weights and the bias values. Therefore, the block diagram for the implementation of the logsigmoidal neuron is shown in Figure 5 and Figure 6.



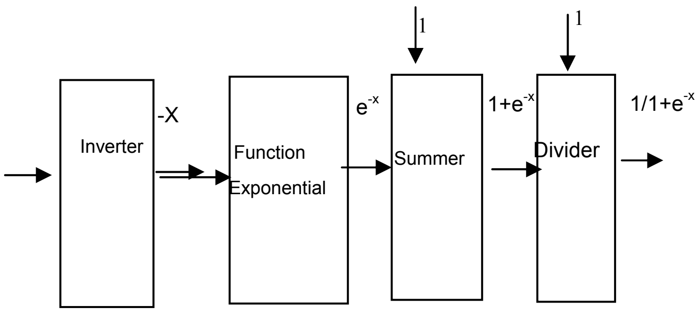


Figure5 & 6 : Block diagram for the implementation of the logsigmoidal neuron

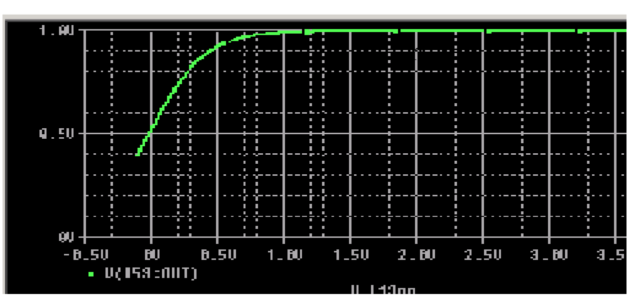
All the circuits in this paper have been simulated using ORCAD V 9.1. The simulation of the designed neuron produces the output given in fig 8, which is in close resemblance with the original transfer function curve.

3.2.2 Design of Linear neurons

In these neurons, the output gives the linear function of input n as given. y= f(n)

$$f(n) = \begin{cases} n & \text{for } n >= 0 \\ 0 & \text{for } n < 0 \end{cases}$$

The simulation of the designed linear neuron is shown in Figure 7 & 8.



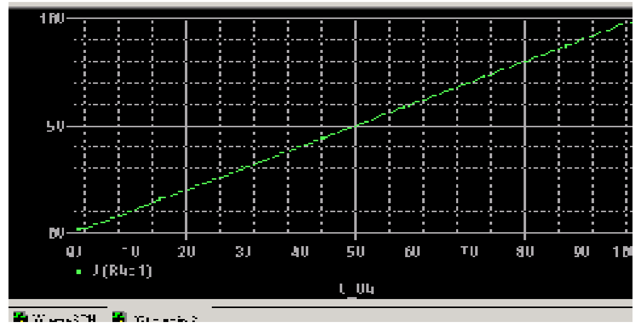


Figure 7 & 8

The circuit shown in Figure 9 is the design for the linear neuron.

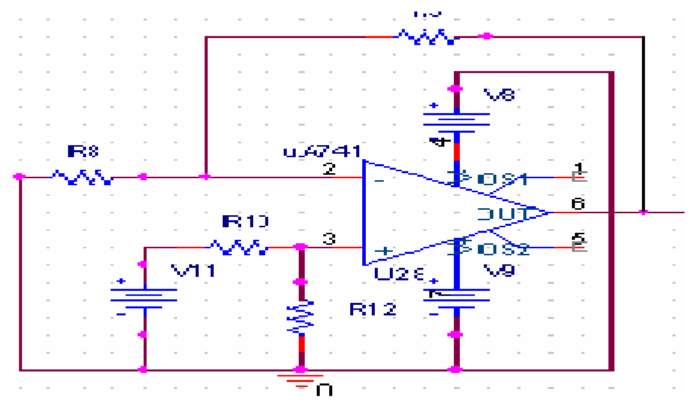


Figure 9: Design for the linear neuron

3.2.3 Weight Control Circuit:

Having designed the neurons involved in the circuit implementation, the next aspect is the design of a weight control circuit. This circuit has to produce an output, which is the product of the input parameter's value and the weight obtained for that particular neuron by training. This means that this circuit needs to provide a gain factor which when multiplied with the input produces the required output.

For example:- if the value of input is 0.3 and the weight by training comes out to be 5 then the output of the weight control circuit should be 0.3*5 = 1.5 units.

The simulation for the same example is being shown in the following figure10.

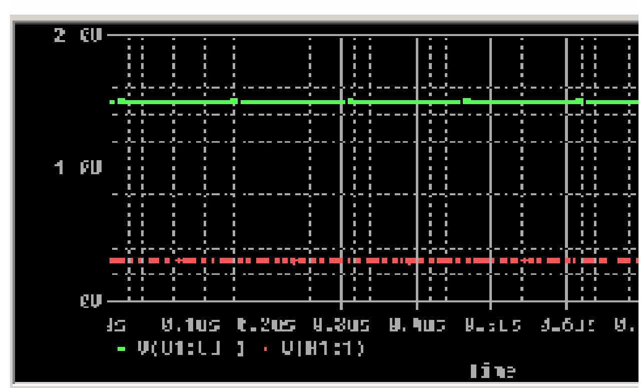


Figure 10: Simulation result from Example

The solid line shows the output 1.5V and the dotted line represents the input 0.3 V.

Since we have by now designed the basic building blocks of our network, the next step is to assemble all these blocks as per our configuration.

The circuit in figure 11 shows the assembly of one logsigmoidal neuron. Similarly, the figure 12 shows the assembly of complete network. For a 4-1-1 neural net structure.

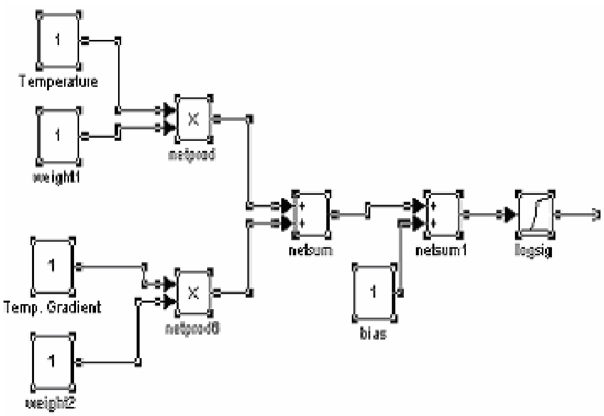


Figure 11: Assembly of one logsigmoidal neuron

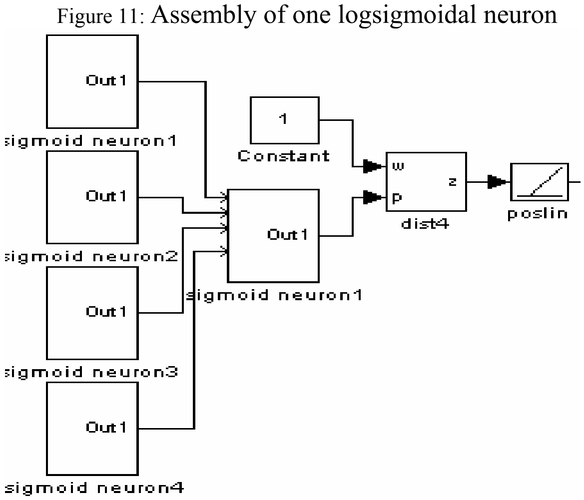


Figure 12: Assembly of complete network.

IV. Conclusions

This paper presented an analog implementation of a rapid nickel cadmium battery charger based on Artificial Neural Networks. The key concept behind a rapid battery charger was the observation that a maximum of charging current up to 8C can be applied to a 2AA Ni-Cd battery up to about the first 5 minutes of charging. Further as the temperature rises, the charging current is controlled based upon the temperature and temperature gradient of the cell. The charging has to be controlled very carefully as the temperature rises because the battery gets damaged at around 50^{o} C. The paper also presented the design of analog neurons with log sigmoid and linear activation functions. These neurons were used in implementing the 4:1:1 topology as suggested by ANDEng.

V. References

[1] Sharon Ahlers, and Mukherjee Anirban, "Advances in Battery Management using Neural Networks and Fuzzy Logic Technical report, School of Electrical and Computer Engineering, Cornell University, Ithaca, NY 14850, 16th May 2003.

[2] 4. Demuth Hagan B., Beale Mark, Thomson, "Neural Network Design", 2002.

[3] Widrow Bernard, Lehr A. Michael," 30 Years of Adaptive Neural Networks: Perceptron, Madaline and Back Propagation", Proceeding of the IEEE, Vol. 78, NO. 9, November 1990.

[4] Haykin Simon," Neural Networks: Comprehensive Foundation", Second Edition

[5] Anderson J.A.," An Introduction to Neural Networks", PHI

[6] Zurada J.M," Introduction to Artificial Neural Systems", Jaico Publishing House

[7] Minsky M, Papert S.,"Perceptrons" (Chapter 1,4), Cambridge, MIT Press, 1969