



## Switching Optimization By Neural Networks

Mansour M.Abaid<sup>1</sup> and Musbah A.Sharf<sup>2</sup>

<sup>1</sup>Electrical Engineering Department, Higher Institute of Technical Engineering,  
Zliten, Libya

mnsrobid@gmail.com

<sup>2</sup>Computer Engineering Department, Higher Institute of Technical  
Engineering, Zliten, Libya

mmaass79@yahoo.com

### Abstract

The priority switching problem is a combinatorial optimization problem searching for the best solution for frame forwarding in a switch. To solve this problem, the modified Hopfield neural network, what is able to solve the priority switching problem is introduced.

**Keywords:** Hopfield; neural; priority; switching.

### I. Introduction

Artificial neural networks with their parallel architecture represent a very perspective way for future data processing. In comparison with classical, sequential data processing neural networks offer a more efficient parallel processing. They are based on mathematical models of the neural system of living organisms what, in highly simplified point of view, is made of a large number of computation elements.

Hopfield and Tank [1] presented a new architecture of neural networks with feed-back connections. This network, using iteration processes, is able to solve combinatorial optimization problems well known from mathematics. Much work was spent on modifying the original architecture and making it more efficient. Very significant results were achieved by Aiyer [2]

during their works, focused on the algebraic transformation of the original mathematic model of the Hopfield network. They presented several new algorithms that give better solutions to standard optimization problems.

## **II. The Priority Switching Problem**

In classical Ethernet networks (using classical switches) there is no native support for frame prioritization. This situation is not very suitable for modern multimedia applications, where the transmission delay is a very important parameter. If the frames are of the same priority (what is the case, when priority support is not implemented), file transmission can block the switch and can increase the transmission delay for other more important communications, e.g. voice or video communication. The only solution for this problem is to add a priority field to the transmitted data expressing the importance of the frame. In the input buffer, there can be several frames waiting for the transmission. From the point of view of the switch instead of the destination address contained in the header of the frame, the corresponding output port of the switch is more important. So the destination address in the frame is used only to identify the output port. The connection process is then based only on the port numbers. The frames waiting in the input buffer of one input port can be expressed mathematically as a vector, where the position of the element in the vector specifies the required output and the value of the element expresses the priority.

A vector (4, 60, 1, 255) means that there are frames with:

Priority 4 and destination port 1,

Priority 60 and destination port 2,

Priority 1 and destination port 3,

Priority 255 and destination port 4.

If value 255 is assigned to the least important frame, it could be used in the situation when there is no request for the given port.

For each input port, such a vector can be constructed. Connecting these vectors a matrix can be constructed. This matrix can be called as priority matrix.

### III. The Hopfield Neural Network

The biggest difference between the Hopfield network and the common artificial neural networks like back prop networks, in its feed-back architecture, as shown in Figure (1).

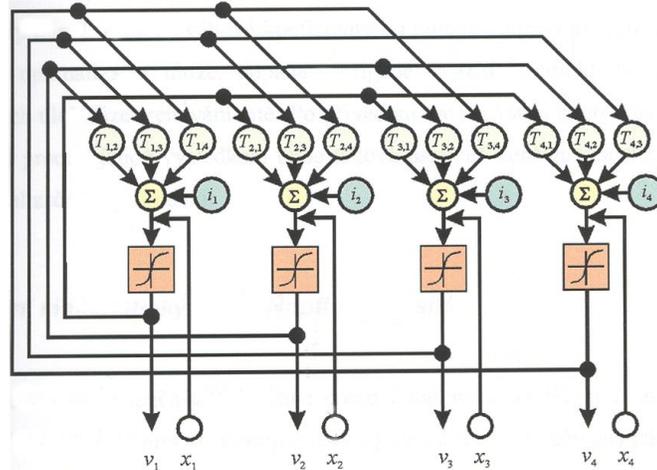


Figure 1. An example of the Hopfield neural network.

This special architecture adds a new essential feature to the network, the temporal behaviour. Generally, such a system is called dynamic, and its behaviour is expressed by a set of differential equations.

The behaviour of the  $j$ -th neuron in the network can be expressed by differential equation (1) describing the change in the state value of the neuron about time.

A mathematical description of the Hopfield network can be expressed by equations (1) and (2), where  $u$  is the state value of the neuron,  $x$  is the input and  $g(u)$  is the output value. Since the network is recurrent, value  $x$  can be expressed by variable  $u$ .

$$\frac{du_j(t)}{dt} = -u_j(t) + \sum_{n=1}^N T_{j,n} g(u_n) + i_j, \quad j=1, 2, \dots, N \quad (1)$$

$$u = \sum_{n=1}^n t1x1(t) + i \quad (2)$$

The relationship between the state value  $u$  and the output value  $v$  is expressed by the activation function of the neuron. In most cases the activation function is nonlinear. Examples of the activation function can be expressed by equations (3) or (4).

$$v_j = g(u_j) = \frac{1}{2} \left( 1 + \tanh \left( \frac{u_j}{u_{0j}} \right) \right) \quad (3)$$

$$v_j = g(u_j) = \begin{cases} 0 & \text{for } u_j < 0, \\ u_j & \text{for } 0 \leq u_j \leq 1 \\ 1 & \text{for } u_j > 1. \end{cases} \quad (4)$$

The state value of the Hopfield network change with time. If the network is configured properly, the movement of the state values in the state space starts from the initial values and ends in a stable state. During this process, the neural network moves from higher energy state to a lower one. This means that the network minimizes its energy function. In general, this energy function can be expressed by equation (5). There is a close relation between the energy function and the system of differential equations describing the neural networks.

$$E = -\frac{1}{2} \sum_i \sum_j T_{i,j} v_i v_j - \sum_i i_i v_i \quad (5)$$

With properly selected weight values  $T_{i,j}$  (the weight of the  $j$ -th input of the  $i$ -th neuron) the energy function can have special forms, such as the form of the energy function of an optimisation problem.

#### **IV. Modified Hopfield Neural Network**

The biggest problem of the classical Hopfield network is that practical implementations of it are very ineffective. They quite often generate not optimal solutions. The problems with affectivity are caused by the very complicated energy function containing several terms to express both the confinement criteria and the object function of the optimization problem. Due to the complicated energy function, the classical Hopfield network can stuck in the local minimum and as a result it will not reach the global minimum corresponding to the optimal result.

A solution to this problem was offered by Aiyer [2]. In this work, the global energy function of the optimization problem is divided into two parts: the confinement function and object function. As it was described earlier, the confinement function is responsible for ensuring the limitation criteria of the optimization problem while the object function expresses the pure optimization problem. The object function expresses the total length of the route connecting the cities in a selected order. Aiyer also showed in his work how to separate the confinement function from the energy function to prevent them working against each other.

The iteration process of the Hopfield model with improved energy function presented in equation (3) can be seen in Figure (2). The process in this figure was presented by A.Gee Figure (2) shows that the iteration runs in two loops. The upper loop represents the projection of the state vector  $v$  into the subspace of valid solutions. The lower loop represents the optimization of the object function.

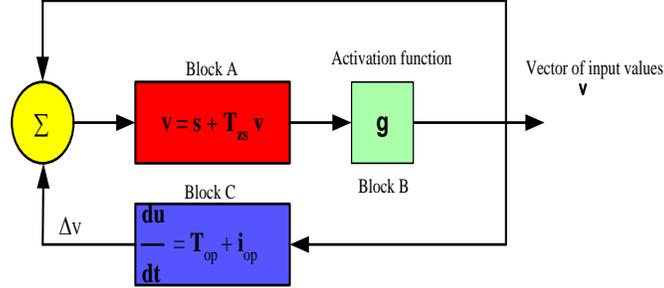


Figure 2. The iteration process of the Hopfield network with modified energy function.

If matrix  $A$  and vector  $b$  define the confinement term and if vector  $v$  satisfies equation (6) then the coefficients of the projection function can be defined by equations (7) and (8) respectively.

$$A.v = b \quad (6)$$

$$T_{zs} = I - A^T(AA^T)^{-1}A \quad (7)$$

$$S = A^T(AA^T)^{-1}b \quad (8)$$

The object function can be expressed by equation (9), where vector  $c$  contains the weights assigned to the corresponding outputs. If the output is active, this weight is added to the total sum. Matrix  $Q$  contains the weights assigned to each pair of outputs. What this means the weight of the and neuron being active at the same time.

$$f(v) = c^T v + v^T Q v \quad (9)$$

By using expressions (7), (8) and the coefficients of object function (9), the parameters of the lower loop can also be derived; the matrix is defined by equation (10) and the vector of bias values  $i$  is defined by equation (11).

$$T_{op} = -2Q + \gamma(T_{zs} - I) \quad (10)$$

$$i = \gamma s - c \quad (11)$$

Matrix I is an identity matrix, and the parameter defines the weight between the confinement function and the object.

### V. Fabric Controlled by Neural Network Switch

Switches receive data units on their input ports and forward those to the correct output port. If a priority value is assigned to each data unit, the algorithm forwarding these units from the input port to the output must first process the data units of higher priority first. The improved Hopfield neural network described in the previous chapter is able to solve the optimization problem of priority switching. A possible architecture of a switch fabric with neural control is shown in Figure (3).

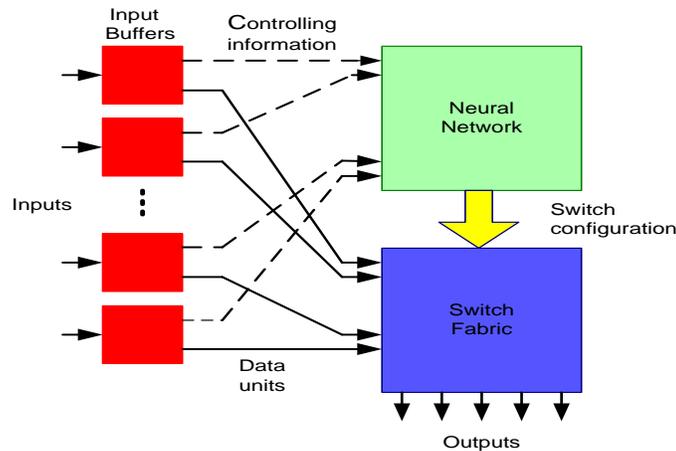


Figure. 3 Switch fabric architecture with neural network control

### VI. Conclusion

To overcome the speed limitations of the optimization problems solving by the artificial neural network is to use parallel architectures with simple function elements. By using a Hopfield network with improved performance, the very effective system architecture can be built.

**Switching Optimization By Neural Networks**  
*Mansour M.Abaid & Musbah A.Sharf*

---

**References**

- [1] K. Molnar, Switching Optimization by Neural Networks.
- [2] AIYER, Solving combinatorial optimization problems using neural networks with application in speech recognition, University of Cambridge, United Kingdom, 1991.
- [3] HOPFEILD, "Neural" Computation of decisions in optimization problems, Biological Cybernetics, Springer-Verlag 1985.
- [4] SMITH, A.K, solving combinatorial optimization problems using neural networks-University of Melbourne, Australia, 1996.
- [5] Trout, T.P Walters, S.M, Neural Network Architecture for Crossbar Switch Control, IEEE Transaction on Circuits and Systems, vol, 38, No. 1. 1991.