Journal of Engineering and Technology Research Vol. 2(4), pp. 58-72, April 2010
Available online at http://www.academicjournals.org/JETR
ISSN 2006-9790 © 2010 Academic Journals

Full Length Research Paper

Study of load distribution in fully connected client server network using feedback neural network architecture

Manu Pratap Singh[1]* Avanish Kumar[2], Pradeep Kumar[3] and Garima Krishna[4]

[1]Department of Computer Science, Institute of Basic Sciences, Ambedkar University, Khandari, Agra (U. P.), India.
[2]Department of Mathematical Sciences and Computer Science, Bundelkhand University, Jhansi (U. P.), India.
[3]RPBD, Central Building Research Institute, Roorkee 247667 (U. A.), India.
[4]Department of Computer Sciences, College of Engineering (COER), Roorkee - 247667 (U. A.), India.

Accepted 1 March, 2010

A fully connected client server network reminiscent of a flat group (analogous to a Hopfield type artificial neural network), consisting of a few clients is considered herein. Distinct client nodes are inter-connected in fully connected architecture as in Hopfield's associative memory model. Weights of connections are symmetric and the states of the clients are bipolar. Stability of such a network can be obtained by using energy function analysis of Hopfield type network. The problem to explore the condition for stability becomes complicated when the server gets connected with the clients, as the connections of the server with the clients are bidirectional but are not symmetric. Hence it does not anymore satisfy the criteria of a Hopfield network to act as an associative memory. In this case the stability is not fixed point stability; rather it is in the form of chaotic or oscillatory states. Under these conditions it is tedious to establish the stability for the entire network in order to determine the optimal load distribution as reflected from connection strengths of the network. This paper explores the stability of clients' network and subsequently the stability of the entire network after adding the server to investigate correlation.

Key words: Artificial neural network, Hopfield energy function, optimized stability, client-server network.

INTRODUCTION

Under the client server paradigm, a network may be configured under different topologies. Considering a fully connected client server network depicting the Hopfield type artificial neural network, it is easy to visualize that by virtue of the full connectivity among the nodes, such a network forms the essential basis of all valid deductive inference for the sake of generalization, as we can achieve different topologies simply by pruning the redundant edges. A flat group is inherently fault tolerant because if one process in a group fails, some other process can take over it through replication (Guerraoui et al., 1997). Following such an approach which is in consonance with Hopfield's classical model, different clients or nodes may be connected with each other to form a fully connected network and by adding a server in the existing network, capabilities of the network may be enhanced. The server is also fully connected with all the clients. The problem is precipitated to determine the optimized stability of the entire network. Initially, the stability condition of the client network can easily be analyzed with the help of Hopfield energy function analysis (Hopfield, 1982; Hopfield, 1984). In the process, it is necessary to assume that the clients are connected with each other in bidirectional fashion and the connection strength is symmetric, which is true in suggested network. Thus the loads are equal from both sides. The output state of every client is bipolar (1 or -1) reflecting that the client under consideration is participating in the communication or not. The stability of this network can be found out by determining the minimum energy state of the network. The stability

*Corresponding author. E-mail: manu_p_singh@hotmail.com.

condition of this network will refer to the minimum energy state in the region as equilibrium for the state space of network. Hence the stability provides optimal values of the connection strengths. These values reflect the required load distribution among the clients in order to keep the system stable. The network may become unstable whenever a server is connected to the existing network. The server is connected with every client of the network but the forward and backward connection strengths are asymmetric. It implies that the loads from client to server and from server to client are not the same. Therefore the entire network now cannot be viewed as the Hopfield model which has the fixed point stabilities. In this condition the state of the network stability may be chaotic or oscillatory. Now the problem leads to the determination of optimized load sharing for the entire network which is required to protect the fault tolerant capabilities of the network. Three categories of neural networks with five different architectures were briefly discussed by Logeswaran and Rajasvaran (2006), for their error tolerance and graceful performance degradation properties in the presence of noise and network failures.

It is obvious in the client server network that if the client is overloaded by the server or it is overloading the server then the load must be redistributed, or shared by the other clients (Hopfield, 1987) to maintain the performance and efficiency of the network. The stability in this case cannot be absolute or fixed point stability. In this situation we must find some other stability state acceptable in the given circumstances. Thus to obtain the optimal stability, in this paper we are applying the neural dynamics of bidirectional associative memory (Bart, 1988; Liwanag and Becker, 1997; Ma Jinwen, 2003). Hopfield energy function was used to determine the minimum energy for the network of clients leading to the stable condition of the network Hebb (2002). Once this stability or minimum energy state is obtained, we may determine the minimum energy state of the entire network so that we have the situation where the stable network is communicating with additional node, and still remains stable. The minimum energy state of the unstable network is calculated and the covariance between the energy states of the stable network and unstable network is computed. The line of regression is drawn to represent the optimal situation for the stability. At this point the connection strengths of the network are modified to keep the network in this optimal stability situation. Thus the stable network adjusts itself against the perturbations. Modified weights represent the distribution of load among clients and also from client to server. Hence it indicates that the network has attained the optimal stability due to the optimal load sharing in the entire network. The next section presents general functioning of the Hopfield neural network and its stability with the energy function analysis. The mathematical model of the system's behavior with the simulation design and algorithm are presented. Also the description of the results and the conclusion of the study and gives some directions for future research.

FEEDBACK NEURAL NETWORKS AND STABILITY

Hopfield (1982) proposed the use of fully connected neural network as an associative memory. The intent is to store a set of bit patterns in such a way that when a new pattern is presented to it, the network produces the stored pattern that is closest to it. Hopfield made use of Hebbian postulates (Hebb, 2002; Hebb, 1949), to develop learning algorithm for training the network. The Hopfield model is one of the popular feedback neural networks. The task of pattern storage is accomplished by the feedback network comprising of processing units with non-linear output functions. Outputs of the processing units at any instant of time define the output state of the network at that instant. The activation dynamics of the network determines the state of the network at successive instants of the time, which is the trajectory of the states associated with each output state, is an energy which depends upon the network parameters, like the weights and thresholds, besides the state of the network. The energy as a function of the state of network corresponds to an energy landscape. Feedback among the units and the non-linear processing in the units may create basins of attractions in the energy landscape. When the weights satisfy certain constraints, the basin of the attraction in the energy landscape tends to be the region of stable equilibrium state (Cohen and Grossberg, 1983). If there is a fixed state in each of the basin where the energy is the minimum then the state corresponds to a fixed point of equilibrium. The basin could also be periodic (oscillatory) region of equilibrium. In this type of equilibrium, the state as the network changes continuously in an oscillatory manner. Thus the state of the network is not predictable but it is confined to the equilibrium region. An arbitrary state may not correspond to equilibrium or a stable state. As the dynamics of the network evolves, the network may eventually settle at a stable state. The dynamical behavior of the neurons' states strongly de-pends on the connection strengths between neurons.

The connection strength from neuron *j* to neuron *i* is the weight w_{ij}. The objective of the Hopfield model is to determine the suitable weights within the dynamic range of the neurons to find out the optimized solution. The Hopfield model as a feedback network may be considered with N processing units and N^2 connection strengths that is weights. This model is a fully connected feedback network with symmetric weights and asynchronous states modulation. Each neuron can be in one of the states that is ±1. In this network the output of each unit is fed to all other units with weights w_{ij}, for all *i* and *j*. The output of the i^{th} unit can be shown as:

$$S_i = f(x_i) = f\left[\sum_{j=1}^{N} W_{ij} S_j - \theta_i\right] \quad (2.1)$$

Where, θ_i is the threshold for the unit i. Let θ_i be zero. Then we have,

$$S_i = f(x_i) = f\left[\sum_{j=1}^{N} W_{ij} S_j\right] \quad \text{for all } i, \qquad (2.2)$$

Hence, due to feedback the state of a unit depends on the states of the other units. The updating of the states of a unit can be done asynchronously in the Hopfield neural network model. In this updating process, a unit is selected at random and its new state is computed. Another unit is selected at random and its state is updated using the current state of the network. The updating process using the random choice of a unit is continued until no further change in the state takes place for all the units, that is, the state at time (t + 1) is the same as at time t for all the units as shown below:

$$S_i(t+1) = f\left[\sum_{\substack{j=1 \\ i \neq j}}^{N} W_{ij} S_j(t)\right] = S_i(t) \quad \text{for all i} \qquad (2.3)$$

Thus, in this situation, the network activation dynamics has reached a stable state so that for all new input situations, the weights or connection strengths between the processing units have been updated in order to learn this behavior. The learning for new behavior in Hopfield model can be accomplished with Hebbian learning rule as:

$$W_{ij}(t+1) = W_{ij}(t) + S_i(t)S_j(t), and W_{ii} = 0, for all i \,\&\, j \qquad (2.4)$$

Hopfield associated an energy function [2] with each state of the network; at the equilibrium there is a fixed stable state for which the energy is at its minimum. The energy function is considered as

$$E(S) = -\frac{1}{2}\sum_i \sum_{j \neq i} W_{ij}(t) S_i(t) S_j(t) \qquad (2.5)$$

Hopfield has shown that for symmetric weights with no self loops around the units and with bipolar output function, the dynamics of the network using the asynchronous update always leads towards energy minima at equilibrium. The states corresponding to these energy minima turns out to be stable states, which means that small perturbations around it lead to unstable states. Hence, the dynamics of the network takes the network back to a stable state again. Thus the energy function value always either reduces or remains the same as the state of the network changes that is:

$$\Delta E(S) \leq 0 \qquad (2.6)$$

The Hopfield energy function analysis has been successfully applied to model a gamut of ANN problems, which is a manifestation that it is a powerful tool for the several optimization problems (Ackley et al., 1985; Zhang et al., 2006; Thomas, 1995; Hideki et al., 2000). Given the energy function or an objective function for a problem in terms of its variables and constraints, if we can identify the coefficients associated with $s_i s_j$ and constant terms in the function then a feedback network can be built with weights corresponding to these coefficients. Then, by using the activation dynamics for the network, the equilibrium state or states can be found. These states correspond to the minima or maxima of the energy function. Therefore this energy function analysis is useful to identify the weights of the network for the optimization problem. Thus the Hopfield energy function analysis can be used to explore the problem of identifying the load or strengths on each connection between different nodes of the network. The requirement is inherited from the Hopfield model i.e. the network is fully connected and the load between the nodes is symmetric. It can be understood easily that if the network is stable or at equilibrium state then the network has minimum error. The minimum error can be interpreted as the objective function for the stability so that if the network has perturbed due to any node, the neural dynamics drives the network towards the stability or the minimum error state. Thus the connection strength or the load between the nodes will modify in order to settle the network in stable state again. The final optimal values of the weights against all the perturbations provide the stability for the network. These optimal values of the weights reflect the optimal load in the network. Hence, it is quite obvious that the load distribution in the network can be defined with the Hopfield energy function analysis, if the network is designed as the fully connected feedback neural network of Hopfield type.

MATHEMATICAL MODELING

The client-server model is one of the central ideas of network computing in which client-server describes the relationship between two computer programs wherein one program, the client, makes a service request from another program, the server, which serves the request. Although the client-server idea can be used by programs within a single computer, it is a more important idea in a network. In a network, the client-server model provides a means to interconnect programs that are distributed efficiently across different locations. Now, here we consider a client server network topology which consists of N number of clients and a server. The total number of nodes is N + 1. The nodes in the network are fully connected. The communication among the clients, with each other, can be considered as a sub-network of the entire network, that is

$$N_c \subset N \qquad (3.1)$$

Hence N_c is the fully connected network. The communication links between the nodes in N_c are bidirectional and symmetric. It means for the nodes S_i and S_j the communication link or connection is defined as

$$w_{ij}^{c} = w_{ji}^{c} \quad \text{And } W_{ii}^{c} = 0 \text{ , for all i.} \qquad (3.2)$$

Thus the N_c can be assumed as feedback recurrent network of the Hopfield type. The nodes in the N_c are assumed to be in two states whether they are participating in communication or not. These two stages of each node can represent the bipolar states that is +1 (participating) and -1 (not participating). Initially all the nodes of N_c are in steady state and the loads on the connections are arbitrary. It is understood that initially N_c is stable with the optimally distributed load on the connections.

Now due to the perturbations in any one of the client's state the change in the N_c state can be discussed with the neural dynamics of feedback neural network using asynchronous state changes for the client. The neural dynamics changes the state of N_c asynchronously until it converges to the stable state. The connection strength of the communication link is modified by using Hebbian learning to attain the stability. Thus the Hopfield energy function analysis gets employed to describe the stability as the equilibrium state or minimum energy state of the network. Therefore as the network changes from one state to another state, the energy function should either remain the same or decrease. Thus, the minimum energy state of N_c against all the perturbations of the clients will exhibit the optimal value of the connection strength, which reflects the distribution of load among the clients for the stable functioning of N_c. For mathematical modeling of this concept let us consider N_c with $C_1, C_2 \ldots C_n$ clients and $w^c_{11}, w^c_{12} \ldots w^c_{1n}$, $w^c_{21}, w^c_{22}, \ldots w^c_{nn}$ as the connection strength of the communication links. The steady state for the clients can be shown as;

$$C_i(t+1) = f[\sum_{\substack{j=1 \\ j \neq i}}^{n} C_j(t).w_{ij}^{c}] = C_i(t)$$

$$C_i(t+1) = f[\sum_{\substack{j=1 \\ j \neq i}}^{n} C_j(t).w_{ij}^{c}] = C_i(t)$$

$$\text{for all } i \in N_c \qquad (3.3)$$

Therefore, the steady state of all the nodes of N_c will represent the stable state of the network. The state of N_c will change as any one of the nodes changes its state. Hence on associating Hopfield energy function as shown in equation (2.5) with each output state of N_c the energy of the clients at any state of N_c can be shown as,

$$E^{client} = -\frac{1}{2}\sum_{i}^{n} \sum_{\substack{j=1 \\ j \neq i}}^{n} w_{ij}^{c} C_i(t) C_j(t) \qquad (3.4)$$

Due to the perturbations in the state of any node, the network N_c becomes unstable. The synaptic dynamics of the network will modify the connection strength to reflect the changes in the network shown as,

$$w_{ij}^{c}(t+1) = w_{ij}^{c}(t) + c_i(t) c_j(t) \qquad (3.5)$$

Hence the new state of any node i due to change in weights and the states of other nodes can be shown as,

$$c_i(t+2) = f[\sum_{j \neq i}^{n} w_{ij}(t+1) c_j(t+1)] \qquad (3.6)$$

The nodes in N_c will update their states asynchronously that is a randomly selected single node is updated one at a time. The updating using the random choice of a node is continued until no further change in the state takes place for all the nodes that is,

$$c_i(t+2) = c_i(t+1) \quad \text{For all } i \in N_c \qquad (3.7)$$

This steady state situation of the nodes provides the stable state for N_c. Hence to exhibit the stability for N_c the Hopfield energy function analysis can be used. Now let us consider the change of states in N_c due to update of a node say k, at some instant (all other units remain unchanged). The expression for energy before and after the change and weight modification can be written as,

$$E_{client}^{old} = -\frac{1}{2}\sum_{i}^{n}\sum_{j \neq i}^{n} w_{ij}^{c}(t) c_i(t) c_j(t) - \frac{1}{2}\sum_{i=1}^{n} w_{ik}^{c}(t) c_i(t) c_k(t) - \frac{1}{2}\sum_{j=1}^{n} w_{jk}^{c}(t) c_j(t) c_k(t)$$

And

$$E_{client}^{new} = -\frac{1}{2}\sum_{i}^{n}\sum_{j \neq i}^{n} w_{ij}^{c}(t+1) c_i(t+1) c_j(t+1) - \frac{1}{2}\sum_{i=1}^{n} w_{ik}^{c}(t+1) c_i(t+1) c_k(t+1) - \frac{1}{2}\sum_{j=1}^{n} w_{jk}^{c}(t+1) c_j(t+1) c_k(t+1) \qquad (3.8)$$

Now the change in energy can be established as

$$\Delta E = E_{client}^{new} - E_{client}^{old}$$

$$= -\frac{1}{2}\sum_{i}^{n}\sum_{j \neq i}^{n} \Delta w_{ij}^{c} c_i(t+1) c_j(t+1) - \frac{1}{2}\sum_{i}^{n} c_i(t+1)[-c_k(t+1) w_{ik}^{c}(t+1) - c_k(t) w_{ik}^{c}(t)]$$

From equation (3.5) we have,

$$\Delta w_{ij}^{c} = c_i(t) c_j(t) \quad \text{So that}$$

$$\Delta E = -\frac{1}{2}\sum_{i}^{n}\sum_{j\neq i}^{n} C_i^2(t+1)C_j^2(t+1) - \frac{1}{2}\sum_{i} C_i(t+1)[W_{ik}^{c}(t)\Delta C_k + C_k(t+1)C_k(t)C_i(t)] \quad (3.9)$$

Or,

$$\Delta E = -\frac{1}{2}\sum_{i}^{n}\sum_{j\neq i}^{n} C_i^2(t+1)C_j^2(t+1) - \frac{1}{2}\sum_{i}[W_{ik}^{c}(t)\Delta C_k C_i(t+1)] - \frac{1}{2}\sum_{i}[C_k(t+1)C_k(t)C_i^2(t+1)]$$

Where, $C_i(t+1) = C_i(t)$

Hence $$\Delta E = -n - \frac{1}{2}\Delta C_k \sum_{i} [W_{ik}^{c} C_i[(t+1)] - \frac{n}{2} C_k(t+1)C_k(t) \quad (3.10)$$

Now the update rule for the unit k can be given as,

(i) if $\sum_{i} [W_{ik}^{c}(t)C_i(t+1) > 0$ then $C_k(t+1) = 1$

in this case if $C_k(t) = 1$ then, $\Delta E = -n - \frac{n}{2}$

$$\Rightarrow \Delta E < 0 \quad (3.11)$$

And if $C_k(t) = -1$ then, $\Delta E = -n + \frac{n}{2} - 1 = \frac{-n-2}{2}$

$\Rightarrow \Delta E < 0$, where $n \geq 0$ (3.12)

(ii) If $\sum_{i} [W_{ik}^{c}(t)C_i(t+1) < 0$ then $C_k(t+1) = -1$

In this case if $C_k(t) = 1$ then

$$\Delta E = -n + \frac{n}{2} - 1 = \frac{-n-2}{2} \Rightarrow \Delta E < 0 \quad (3.13)$$

and if $C_k(t) = -1$ then, $\Delta E = -n + \frac{n}{2} = -\frac{n}{2} \Rightarrow \Delta E < 0$ (3.14)

(iii) if $\sum_{i} [W_{ik}^{c}(t)C_i(t+1)$ =0 then $C_k(t+1) = C_k(t)$

in this case if $C_n(t) = 1$ then,

$$\Delta E = -n + \frac{n}{2} - 1 = \frac{-n-2}{2} \Rightarrow \Delta E < 0 \quad (3.15)$$

And if $C_k(t) = -1$ then, $\Delta E = -n - \frac{n}{2} = -\frac{3n}{2}$ (3.16)

Thus for every situation, we have ∆E < = 0. Therefore the energy always decreases when a node, selected at random is updated. Hence for each perturbation the N_c is adjustable. Thus the N_c will lead towards the stable state or the minimal energy state against all the perturbations. The optimal connection strength in the N_c at the stable state reflects the optimal load on each link between the nodes to keep the N_c at stable state or the state of equilibrium.

After getting N_c stabilized the server S is allowed to be made functional. Now we have the complete network N, which includes a server and N_c. The N_c has already become stable and on adding the server S, the entire network again moves out of equilibrium. The functionality of the server can be treated as the perturbations in the network N_c and also in the entire network. In this network the connections between the client to client are symmetric and between clients to server are asymmetric. This is obvious because the load from client to server is different from the load which is assigned from server to client. The load from client to server is in the form of request and the load from server to client is in the form of service. Thus the load from server to client is more that the load from the client to server. Hence we have the feedback network which is fully connected (with the nodes of the clients and server both) are having the bipolar state and the connections between the nodes are bidirectional, asymmetric and symmetric both. It means for the nodes (clients) C_i and C_j the connections is defined as,

$W_{ij}^{c} = W_{ji}^{c}$ And $W_{ii}^{c} = 0$ for all $i \in N_c \subset N$

And for the nodes (clients to server) S and C_i the connection is defined as,

$$W_{si}^{sc} \neq W_{is}^{cs}, and, W_{ss}^{s} = 0, W_{ii}^{c} = 0, \forall, i \in N, S \in N$$

It is considered that N_c is stable and the nodes of N_c are in steady state. The optimal load on the connection is determined. The server node S is connected in the existing network of clients and due to the participation of server in the network the entire network can be considered as unstable. Again the perturbations occurring due to the server or any client activity the change in the network state can be discussed with neural dynamics of feedback neural network using asynchronous state changes for the client and for the server. Here we have three policies of state change of nodes in the network.

- Only one randomly selected client will update its state
- Only Server updates its state
- One randomly selected client and the server both update their state.

In all the three cases the neural dynamics changes the state of the network until it converges to the stable states so that to achieve the stability the connection strength of the communication link will get modified by using the Hebbian learning rule. Hence as the network changes the state the energy of the new state either remains the same or decreases. Thus the minimum energy state represents the stability of the network, which has been lying from unstable state due to perturbation. Therefore to settle the network back in stable state from the instability due to perturbations, the modified optimal connection strength has achieved. These connection strengths reflect the optimal distribution of load in the entire network for the stable functioning of the network N. Now we consider the network N with C_1,C_2….C_n clients and S as the server are the connection strength from the client to server and server to client. The steady state for the server can be defined as,

$$S(t+1) = f\left[\sum_{i=1}^{n} C_i(t).W_{is}^{cs}(t)\right] = S(t) \quad (3.17)$$

and, for the steady state of the client can be shown as,

$$C_i(t+1) = f\left[\sum_{j=1}^{n} C_i(t)\, w_{ij}^{c} + w_{si}^{cs}(t)s(t)\right] = C_i(t) \quad (3.18)$$

For all $i \in N_c$ & $S \in N$

Thus, the steady state of all the clients and the server will represent the stable state of the entire network. The state of the network can change if any one of the three above mentioned conditions occurs. Hence, on associating the energy function with each state of network the energy landscape can be described. The energy for any state of the network in the energy landscape can be shown as,

$$E^{sc} = -\frac{1}{2}\sum_{i=1}^{n}\sum_{\substack{j=1\\ i\neq j}}^{n} W_{ij}^{c} C_i(t) C_j(t) - \frac{1}{2}\sum_{i} W_{is}^{sc} C_i(t) S(t) - \frac{1}{2}\sum_{i}^{n} W_{si}^{sc} S(t) C_i(t) \quad (3.19)$$

$$E^{sc} = E^{client} - \frac{1}{2}\sum_{i} W_{is}^{sc} C_i(t) S(t) - \frac{1}{2}\sum_{i}^{n} W_{si}^{sc} S(t) C_i(t) \quad (3.20)$$

Hence the synaptic dynamics of the network will modify the connection strength to reflect the changes in the network in the order to bring the network back to stable state; from the instability due to the perturbations. The modifications in the connection strength between clients to client can be shown as,

$$W_{ij}^{c}(t+1) = W_{ij}^{c}(t) + C_i(t)C_j(t) \quad (3.21)$$

The modifications in the connection strength between clients to server can be shown as,

$W_{is}^{sc}(t+1) = W_{is}^{sc}(t) + C_i(t)S(t)$, and

$$W_{si}^{sc}(t+1) = W_{si}^{sc}(t) + S(t)C_i(t) \quad (3.22)$$

The new state of any client node i due to the change in weight and the state of other node can be shown as,

$$C_i(t+2) = f\left[\sum_{\substack{j=1\\ j\neq i}}^{n} C_j(t+1)\, w_{ij}^{c} + w_{si}^{sc}(t+1)S(t+1)\right] \quad (3.23)$$

The new state of server can be shown as

$$S(t+2) = f\left[\sum_{i=1}^{n} C_i(t+1).\, w_{is}^{sc}(t+1)\right] \quad (3.24)$$

The nodes in the entire network will update their states as per the above discussed policy. The process of update in the nodes will continue until no further change in the state of node takes place, that is

$C_i(t+2) = C_i(t+1)$ For all $i \in N_c$

And, S (t+2) = S (t+1) where $S \in N$ (3.25)

Therefore to exhibit the stability for the network N the Hopfield energy function analysis can be used. Correspondingly, in our approach the Hopfield energy function will be used for all the three cases of state change policy. Now we will discuss the stability condition for all the three cases of state change policy.

Case - I

Let us consider the change of state in the network due to the update of one client node say k, at some instant. All the unit nodes including the server remain unchanged. Therefore the extension for energy function before and after the change in state of k^th unit and the connection strength modification can be shown as,

$$E_N^{old} = -\frac{1}{2}\sum_{i}^{n}\sum_{j\neq i}^{n} w_{ij}^{c}(t) C_i(t) C_j(t) - \frac{1}{2}\sum_{i=1}^{n} w_{is}^{sc}(t) C_i(t) S(t) - \frac{1}{2}\sum_{i=1}^{n} w_{si}^{sc}(t) S(t) C_i(t)$$
$$-\frac{1}{2}\sum_{i}^{n} w_{ik}^{c}(t) C_i(t) C_k(t) - \frac{1}{2}\sum_{j}^{n} w_{jk}^{c}(t) C_j(t) C_k(t) - \frac{1}{2}\sum_{i=1}^{n} w_{ks}^{sc}(t) C_k(t) S(t) - \frac{1}{2}\sum_{i=1}^{n} w_{sk}^{sc}(t) S(t) C_k(t) \quad (3.26)$$

And,

$$E_N^{New} = -\frac{1}{2}\sum_{i}^{n}\sum_{j\neq i}^{n} w_{ij}^{c}(t+1) C_i(t+1) C_j(t+1) - \frac{1}{2}\sum_{i=1}^{n} w_{is}^{sc}(t+1) C_i(t+1) S(t+1) - \frac{1}{2}\sum_{i=1}^{n} w_{si}^{sc}(t+1) S(t+1) C_i(t+1)$$
$$-\frac{1}{2}\sum_{i}^{n} w_{ik}^{c}(t+1) C_i(t+1) C_k(t+1) - \frac{1}{2}\sum_{j}^{n} w_{jk}^{c}(t+1) C_j(t+1) C_k(t+1) - \frac{1}{2}\sum_{i=1}^{n} w_{ks}^{sc}(t+1) C_k(t+1) S(t+1)$$
$$-\frac{1}{2}\sum_{i=1}^{n} w_{sk}^{sc}(t+1) S(t+1) C_k(t+1) \quad (3.27)$$

Therefore, the change in energy is given by;

$$\Delta E = E_N^{New} - E_N^{old}$$

$$= -\frac{1}{2}\sum_i^n\sum_{j\neq i}^n W_{ij}^c(t+1)C_i(t+1)C_j(t+1) - \frac{1}{2}\sum_{i=1}^n W_{is}^{sc}(t+1)C_i(t+1)S(t+1) - \frac{1}{2}\sum_{i=1}^n W_{si}^{sc}(t+1)S(t+1)C_i(t+1)$$

$$-\frac{1}{2}\sum_i^n W_{ik}^c(t+1)C_i(t+1)C_k(t+1) - \frac{1}{2}\sum_j^n W_{jk}^c(t+1)C_j(t+1)C_k(t+1) - \frac{1}{2}\sum_{i=1}^n W_{ks}^{sc}(t+1)C_k(t+1)S(t+1)$$

$$-\frac{1}{2}\sum_{i=1}^n W_{sk}^{sc}(t+1)S(t+1)C_k(t+1)\{-\frac{1}{2}\sum_i^n\sum_{j\neq i}^n W_{ij}^c(t)C_i(t)C_j(t) - \frac{1}{2}\sum_{i=1}^n W_{is}^{sc}(t)C_i(t)S(t) - \frac{1}{2}\sum_{i=1}^n W_{si}^{sc}(t)S(t)C_i(t)$$

$$-\frac{1}{2}\sum_i^n W_{ik}^c(t)C_i(t)C_k(t) - \frac{1}{2}\sum_j^n W_{jk}^c(t)C_j(t)C_k(t) - \frac{1}{2}\sum_{i=1}^n W_{ks}^{sc}(t)C_k(t)S(t) - \frac{1}{2}\sum_{i=1}^n W_{sk}^{sc}(t)S(t)C_k(t)\}$$

Since in this case

S (t+1) = S (t), $C_i(t+1) = C_i(t)$ for all i except $i \neq k$ and $w_{ij}^c = w_{ji}^c$ (3.28)

Hence, from equation (3.9), we have,

$$\Delta E = \Delta E^{client} - \frac{1}{2}\sum_{i=1}^n W_{is}^{sc}(t)C_i(t)S(t) - \frac{1}{2}\sum_{i=1}^n C_i(t)[W_{is}^{sc}(t+1)S(t+1) - W_{is}^{sc}(t)S(t)$$

$$-\frac{1}{2}S(t)[W_{ks}^{sc}(t+1)C_k(t+1) - W_{ks}^{sc}(t)C_k(t) - \frac{1}{2}S(t)[W_{sk}^{sc}(t+1)C_k(t+1) - W_{sk}^{sc}(t)C_k(t)$$

Here, $\Delta E = \Delta E^{client} - F_1 - F_2 - F_3 - F_4$ (3.30)

We have already determined the solution for ΔE^{client} as shown in equation (3.10). Now we determine the solutions for F_1, F_2, F_3, and F_4 and find out the final solution of ΔE. So, we have,

$$F_1 = \frac{1}{2}\sum_{i=1}^n \Delta W_{ic}^{sc} C_i(t)S(t) \quad (3.31)$$

Now from equation (3.22) we have;

$$\Delta W_{is}^{sc} = S(t)C_i(t) \quad (3.32)$$

From equation (3.31) and (3.32) we have;

$$\Delta E = \Delta E^{client} - \frac{1}{2}\sum_{i=1}^n W_{is}^{sc}(t)C_i(t)S(t) - \frac{1}{2}\sum_{i=1}^n C_i(t)[W_{is}^{sc}(t+1)S(t+1) - W_{is}^{sc}(t)S(t)$$

$$-\frac{1}{2}S(t)[W_{ks}^{sc}(t+1)C_k(t+1) - W_{ks}^{sc}(t)C_k(t) - \frac{1}{2}S(t)[W_{sk}^{sc}(t+1)C_k(t+1) - W_{sk}^{sc}(t)C_k(t)$$

(3.33)

Since the states are bipolar. Now, for the term F_2 we have;

$$F_2 = \frac{1}{2}\sum_{i=1}^n C_i(t)[W_{si}^{sc}(t+1)S(t) - W_{si}^{sc}(t)S(t)]$$

or, $$F_2 = \frac{1}{2}\sum_{i=1}^n C_i(t)\Delta W_{si}^{sc} S(t)$$

Hence from equation (3.22) we have,

$$F_2 = \frac{1}{2}\sum_{i=1}^n C_i(t)S(t)C_i(t)S(t) = \frac{1}{2}S^2(t)\sum_{i=1}^n C_i^2(t) = \frac{n}{2} > 0$$

(3.34)

Now for the term F_3 we have

$$F_3 = \frac{1}{2}S(t)[W_{ks}^{sc}(t+1)C_k(t+1) - W_{ks}^{sc}(t)C_k(t)]$$

From equation (3.32) we have

$$W_{ks}^{sc}(t+1) = W_{ks}^{sc}(t) + C_k(t)S(t) \quad (3.29) \quad (3.35)$$

So that from equation (3.35) we have,

$$F3 = \frac{1}{2}S(t)[(W_{ks}^{sc}(t)C_k(t+1) + C_k(t)C_k(t+1)S(t) - W_{ks}^{sc}(t)C_k(t)]$$

$$= \frac{1}{2}S(t)[W_{ks}^{sc}(t)\Delta C_k(t) + C_k(t)C_k(t+1)S(t)]$$

$$= \frac{1}{2}\Delta C_k[W_{ks}^{sc}(t)S(t)] + \frac{1}{2}C_k(t)C_k(t+1)S^2(t)]$$

or, $$F3 = \frac{1}{2}S(t)[(W_{ks}^{sc}(t)C_k(t+1) + C_k(t)C_k(t+1)S(t) - W_{ks}^{sc}(t)C_k(t)]$$

since $S^2(t) = 1$ (3.36)

Hence the update rule for each k unit can be considered as;

(i) If $W_{ks}^{sc}(t)S(t) > 0$ then $C_k(t+1) = 1$

In this case if $C_k(t) = 1$ then $F_3 = \frac{1}{2} \Rightarrow F_3 > 0$

And if $C_k(t) = -1$ then $F_3 = 1 - \frac{1}{2} \Rightarrow F_3 = \frac{1}{2} \Rightarrow F_3 > 0$

(ii) If $W_{ks}^{sc}(t)S(t) < 0$ then $C_k(t+1) = -1$

In this case if $C_k(t)=1$ then

$$F_3 = 1+\frac{1}{2}(1)(-1) \Rightarrow F_3 = \frac{1}{2} \Rightarrow F_3 > 0$$

And if $C_k(t)=-1$ then $F_3 = \frac{1}{2} \Rightarrow F_3 > 0$

(iii) If $W_{ks}^{sc}(t)S(t)=0$ then $C_k(t+1)=C_k(t)$

In this case if $C_k(t)=1$ then $F_3 = \frac{1}{2} \Rightarrow F_3 > 0$

And if $C_k(t)=-1$ then $F_3 = \frac{1}{2} \Rightarrow F_3 > 0$

Therefore, for every case, we have
$F_3 > 0$

Now for the term F_4 we have,

$$F_4 = \frac{1}{2}S(t)[W_{sk}^{sc}(t+1)C_k(t+1) - W_{sk}^{sc}(t)C_k(t)]$$

Hence from equation (3.22) we have

$$W_{sk}^{sc}(t+1) = W_{sk}^{sc}(t) + S(t)C_k(t)] \qquad (3.37)$$

So that from equation (3.36) we have,

$$F_4 = \frac{1}{2}S(t)[(W_{sk}^{sc}(t)C_k(t+1) + S(t)C_k(t))C_k(t+1) - W_{sk}^{sc}(t)C_k(t)]$$
$$= \frac{1}{2}\Delta C_k[W_{sk}^{sc}(t)S(t)] + \frac{1}{2}C_k(t)C_k(t+1) \qquad (3.38)$$

Where $S^2(t)=1$

The update rules for each k unit are similar to the equation (3.36) therefore we always have $F_4>0$. From the equation (3.30) it is obvious that always $\Delta E < 0$.

Thus, in this state change policy for every situation we have $\Delta E \le 0$. Therefore the energy always decreases when a client node is randomly selected for the update. Hence for each perturbation due to the client the network N is adjustable. The optimal connection strength at the equilibrium state of network reflects the optimal load distribution in the network to keep the network in equilibrium state for the case-I.

Case-II

Let us consider the change of state in the network due to the update of the server node at some instant. All the client nodes remain unchanged. Thus the expression for energy before and after the change in the server state and the connection strength can be written as,

$$E_N^{old} = -\frac{1}{2}\sum_i^n\sum_{j\neq i}^n W_{ij}^c(t)W_{ij}^c C_i(t)C_j(t) - \frac{1}{2}\sum_{i=1}^n W_{is}^{sc}(t)C_i(t)S(t) - \frac{1}{2}\sum_{i=1}^n W_{si}^{sc}(t)S(t)C_i(t)$$

And

$$E_N^{new} = -\frac{1}{2}\sum_i^n\sum_{j\neq i}^n W_{ij}^c(t+1)C_i(t+1)C_j(t+1) - \frac{1}{2}\sum_{i=1}^n W_{ij}^{sc}(t+1)C_i(t+1)S(t+1) - \frac{1}{2}\sum_{i=1}^n W_{si}^{sc}(t+1)S(t+1)C_i(t+1)$$
$$-\frac{1}{2}\sum_{i=1}^n W_{si}^{sc}(t+1)S(t+1)C_k(t+1)$$

The change in the energy can be expressed as

$$\Delta E = E_N^{new} - E_N^{old}$$
$$= -\frac{1}{2}\sum_i^n\sum_{j\neq i}^n C_i(t)C_j(t)\Delta W_{ij}^c - \frac{1}{2}\sum_{i=1}^n C_i(t)[W_{is}^{sc}(t+1)S(t+1) - W_{is}^{sc}(t)S(t)]$$
$$-\frac{1}{2}\sum_{i=1}^n C_i(t)[W_{si}^{sc}(t+1)S(t+1) - W_{si}^{sc}(t)S(t)]$$

And

$$\Delta E = -n - \frac{1}{2}\Delta S\sum_{i=1}^n [W_{is}^{sc}(t)C_i(t)] - \frac{n}{2}S(t)C_j(t+1)$$
$$-\frac{1}{2}\Delta S\sum_{i=1}^n [W_{si}^{sc}(t)C_i(t)] - \frac{n}{2}S(t)S(t+1)$$

Or, $\Delta E = -n + F_1 + F_2$ (3.40)

Hence for the term F_1 and F_2 we have,

$$F_1 = -\frac{1}{2}\Delta S\sum_{i=1}^n [W_{is}^{sc}(t)C_i(t)] - \frac{n}{2}S(t)S(t+1)$$
$$F_2 = -\frac{1}{2}\Delta S\sum_{i=1}^n [W_{si}^{sc}(t)C_i(t)] - \frac{n}{2}S(t)S(t+1)$$

Now the update rule for the state of the server is as;

(i) If $\sum_{i=1}^n [W_{is}^{sc}(t)C_i(t)] > 0$ then $S(t+1)=1$

In this case if $S(t)=1$ then $F_1 = -\frac{n}{2}$

and, if $\sum_{i=1}^n [W_{si}^{sc}(t)C_i(t)] > 0$ then $S(t+1)=1$

In this case, if $S(t)=1$ then $F_2 = -\frac{n}{2}$

So from equation (3.40) we have,

$$\Delta E = -n - \frac{n}{2} - \frac{n}{2} \Rightarrow \Delta E < 0$$

And, if $S(t) = -1$ then $F_1 = \frac{n-2}{2}$

Similarly $F_2 = \frac{n-2}{2}$

So again $\Delta E = -n + \frac{n-2}{2} + \frac{n-2}{2} \Rightarrow \Delta E < 0$

(ii) if $\sum_{i=1}^{n} [W_{is}^{sc}(t) C_i(t)] < 0$ then $S(t+1) = -1$ (3.39)

In this case if $S(t) = 1$ then $F_1 = -\frac{1}{2}$

And similarly, $F_2 = -\frac{1}{2}$

And, $\Delta E = -n - \frac{1}{2} - \frac{1}{2} \Rightarrow \Delta E < 0$

And if $S(t) = -1$

Then $F_1 = -1 + \frac{n}{2}$ and $F_2 = -1 + \frac{n}{2}$

So again

$$\Delta E = -n - 1 + \frac{n}{2} - 1 + \frac{n}{2} \Rightarrow \Delta E < 0$$

(iii) If $\sum_{i=1}^{n} [W_{is}^{sc}(t) C_i(t)] = 0$ then $S(t+1) = S(t)$

In this case if $S(t) = 1$ then $F_1 = -\frac{1}{2}$ and also $F_2 = -\frac{1}{2}$

And, $\Delta E = -n - \frac{1}{2} - \frac{1}{2} \Rightarrow \Delta E < 0$

And if, $S(t) = -1$ then $F_1 = -\frac{1}{2}$

And also, $F_2 = -\frac{1}{2}$

Hence, $\Delta E < 0$

Thus in this state change policy, after every iteration we have $\Delta E < 0$. Therefore, the energy of the network always decreases when the server node alone is selected for updating. Hence for each perturbation due to the server the network N is adjustable. The connection strengths in the entire network at this situation reflect the optimal distribution of the load between client and server and vice-versa to keep the network at equilibrium state.

Case - III

In this policy we consider the change of state in the network due to the update of the server node and a randomly selected client node simultaneously. Thus for the network N_c the state is changing asynchronously, but for the entire network N the nodes will update their states, one is server and the other is randomly selected client node so that the expression for energy before and after the change in the state can be shown as;

$$E_N^{Old} = -\frac{1}{2}\sum_{i}^{n}\sum_{j \neq i}^{n} w_{ij}^{c}(t) C_i(t) C_j(t) - \frac{1}{2}\sum_{i=1}^{n} w_{ik}^{c}(t) C_i(t) C_k(t) - \frac{1}{2}\sum_{j=1}^{n} w_{jk}^{c}(t) C_j(t) C_k(t)$$
$$-\frac{1}{2}\sum_{i=1}^{n} w_{sii}^{sc}(t) S(t) C_i(t) - \frac{1}{2}\sum_{i=1}^{n} w_{is}^{sc}(t) C_i(t) S(t) - \frac{1}{2}\sum_{i=1}^{n} w_{sk}^{sc}(t) S(t) c_k(t) - \frac{1}{2}\sum_{i=1}^{n} w_{ks}^{sc}(t) C_k(t) S(t)$$

And

$$E_N^{New} = -\frac{1}{2}\sum_{i}^{n}\sum_{j \neq i}^{n} w_{ij}^{c}(t+1) C_i(t+1) C_j(t+1) - \frac{1}{2}\sum_{i=1}^{n} w_{ik}^{c}(t+1) C_i(t+1) C_k(t+1)$$
$$-\frac{1}{2}\sum_{j=1}^{n} w_{jk}^{c}(t+1) C_j(t+1) C_k(t+1) - \frac{1}{2}\sum_{i=1}^{n} w_{si}^{sc}(t+1) S(t+1) C_i(t+1) - \frac{1}{2}\sum_{i=1}^{n} w_{is}^{sc}(t+1) C_i(t+1) S(t+1)$$
$$-\frac{1}{2}\sum_{i=1}^{n} w_{sk}^{sc}(t+1) S(t+1) C_k(t+1) - \frac{1}{2}\sum_{i=1}^{n} w_{ks}^{sc}(t+1) C_k(t+1) S(t+1)$$

The change of energy can be defined as;

$$\Delta E = E_N^{new} - E_N^{old}$$
$$= [n - \frac{1}{2}\sum_{i=1}^{n} \Delta C_k [W_{ik}^{c}(t) C_i(t)] - \frac{n}{2} C_k(t) C_k(t+1)] + [-\frac{1}{2}\Delta S\{\sum_{i=1}^{n} W_{ji}^{sc} C_i(t)\} - \frac{n}{2} S(t) S(t+1)]$$
$$+ [\frac{1}{2}\Delta S\{\sum_{i=1}^{n} W_{ij}^{sc}(t)\} - \frac{n}{2} S(t) S(t+1)] - \frac{1}{2}[C_k(t+1)(W_{sk}^{sc}(t+1) S(t+1)) - C_k(t)(W_{sk}^{sc}(t) S(t+1))]$$
$$-\frac{1}{2}[S_k(t+1)(W_{ks}^{sc}(t+1) C(t+1)) - S_k(t)(W_{ks}^{sc}(t) S(t))]$$

Or, $\Delta E = -n + F_1 + F_2 + F_3 + F_4 + F_5$ (3.42)

Now for the F_1 we have,

$$F_1 = -n - \frac{1}{2}\sum_{i}^{n} \Delta C_k [W_{ik}^{c}(t) C_i(t) - \frac{n}{2} C_k(t) C_k(t+1)]$$

For the update rules for the state as k[th] unit can be considered as;

(i) if $\sum_{i}^{n} [\, W_{ik}^{c}(t) C_i(t) \,] > 0$ then, $C_k(t+1) = 1$

In this case if $C_k(t) = 1$ then,

$$F_1 = -n - \frac{n}{2} \Rightarrow F_1 = -\frac{3n}{2} \Rightarrow F_1 < 0$$

And, if $C_k(t) = -1$ then,

$$F_1 = -n - 1 + \frac{n}{2} \Rightarrow F_1 = \frac{-2n - 2 + n}{2}$$
$$\Rightarrow F_1 = \frac{-n-2}{2} \Rightarrow F_1 < 0$$

(ii) if $\sum_{i=1}^{n} [W_{ik}^{c}(t) C_i(t)] < 0$ then $C_k(t+1) = -1$

In this case if $C_k(t) = 1$ then

$$F_1 = -n - 1 + \frac{n}{2} \Rightarrow F_1 = \frac{-n-1}{2} \Rightarrow F_1 < 0$$

And if $C_k(t) = -1$ then,

$$F_1 = -n - \frac{n}{2} \Rightarrow F_1 = -\frac{3n}{2} \Rightarrow F_1 < 0$$

(iii) if $\sum_{i=1}^{n} [W_{ik}^{c}(t)C_i(t)] = 0$ then $C_k(t) = C_k(t+1)$

In this case if $C_k(t) = 1$ then,

$$F_1 = -n - \frac{n}{2} \Rightarrow F_1 < 0$$

And if $C_k(t) = -1$ then,

$$F_1 - n - \frac{n}{2} \Rightarrow F_1 = -\frac{3n}{2} \Rightarrow F_1 < 0$$

Therefore for all the cases of state change in k[th] unit we have $F_1 < 0$

Now for F_2 we have

$$F_2 = -\frac{1}{2}\Delta S[\sum_{i}^{n} W_{si}^{sc}C_i(t)] - \frac{n}{2}S(t)S(t+1)$$

The update rule for the state of server S can be considered as;

(i) if $\sum_{i}^{n} [W_{si}^{sc}C_i(t)] > 0$ then $S(t+1) = 1$

In this case if $s(t) = 1$ then,

$$F_2 = -\frac{n}{2} \Rightarrow F_2 < 0$$

And if $S(t) = -1$ then,

$$F_2 = -1 + \frac{n}{2} \Rightarrow F_2 = \frac{-2+n}{2} \Rightarrow F_2 > 0$$

(ii) if $\sum_{i=1}^{n} [W_{si}^{sc}C_i(t)] < 0$ then $S(t+1) = -1$

In this case if $S(t) = 1$ then,

$$F_2 = -1 + \frac{n}{2} \Rightarrow F_2 > 0$$

And if $S(t) = -1$ then,

$$F_2 = -\frac{n}{2} \Rightarrow F_2 < 0$$

(iii) if $\sum_{i=1}^{n} [W_{si}^{sc}C_i(t)] = 0$ then $S(t+1) = S(t)$

In this case if $S(t) = 1$ then

$$F_2 = -\frac{n}{2} \Rightarrow F_2 < 0$$

And if $S(t) = -1$ then also,

$$F_2 = -\frac{n}{2} \Rightarrow F_2 < 0$$

It can now be seen that F_2 is not always less than zero. In fact it is positive in two typical cases.

For F_3, we have

$$F_3 = -\frac{1}{2}\Delta S[\sum_{i}^{n} W_{is}^{sc}C_i(t)] - \frac{n}{2}S(t)S(t+1)$$

The conditions for F_3 will remain the same as for F_2. Hence F_3 is also not less than zero for two typical cases.

For the F_4 we have,

$$F_4 = -\frac{1}{2}[C_k(t+1)(W_{sk}^{sc}(t+1)S(t+1)) - C_k(t)(W_{sk}^{sc}(t)S(t))]$$

Using the equation (3.22) we have,

$$F_4 = -\frac{1}{2}[W_{sk}^{sc}(t)C_k(t+1)S(t+1) + C_k(t)C_k(t+1)S(t)S(t+1) - C_k(t)W_{sk}^{sc}S(t)]$$
$$= -\frac{1}{2}[W_{sk}^{sc}(t)\{C_k(t+1)S(t+1) - C_k(t)S(t)\} + C_k(t)C_k(t+1)S(t)S(t+1)]$$
$$= -\frac{1}{2}[W_{sk}^{sc}(t)C_k(t+1)S(t+1) - (W_{sk}^{sc}(t)C_k(t))S(t)] - \frac{1}{2}C_k(t)C_k(t+1)S(t)S(t+1) \quad (3.43)$$

The update rules for the state of server S and any client k can be considered as;

(i) if $W_{sk}^{sc}(t)C_k(t+1) > 0$ then $S(t+1) = 1$

And if $W_{sk}^{sc}(t)C_k(t) > 0$ then $S(t) = 1$

Now $F_4 = -\frac{1}{2}C_k(t)C_k(t+1)$

If $C_k(t) = C_k(t+1)$ then $F_4 < 0$ else $F_4 > 0$

(i) if $W_{sk}^{sc}(t)C_k(t+1) > 0$ then $S(t+1) = 1$

And if $W_{sk}^{sc}(t)C_k(t) < 0$ then $S(t) = -1$

Now $F_4 = \frac{1}{2}C_k(t)C_k(t+1)$

If $C_k(t) = C_k(t+1)$ then $F_4 > 0$ else $F_4 < 0$

(ii) if $W_{sk}^{sc}(t)C_k(t+1) < 0$ then $s(t+1) = -1$

And if $W_{sk}^{sc}(t)C_k(t) > 0$ then $S(t) = 1$

Now $F_4 = \frac{1}{2}C_k(t)C_k(t+1)$

Again if, $C_k(t) = C_k(t+1)$ then $F_4 > 0$ else $F_4 < 0$

(iii) if $W_{sk}^{sc}(t)C_k(t+1) < 0$ then $S(t+1) = -1$

And if $F_4 = \frac{1}{2} - \frac{1}{2} C_k(t) C_k(t+1)$ then $S(t) = -1$

Now $F_4 = -\frac{1}{2} C_k(t) C_k(t+1)$

If $F_4 = \frac{1}{2} - \frac{1}{2} C_k(t) C_k(t+1)$ then $F_4 < 0$ else $F_4 > 0$

(iv) if $W_{sk}^{sc}(t) C_k(t+1) = 0$ then $S(t+1) = S(t)$

And if $W_{sk}^{sc}(t) C_k(t) = 0$ then $S(t) = S(t-1)$

Now $F_4 = -\frac{1}{2} C_k(t) C_k(t+1) S^2(t) = -\frac{1}{2} C_k(t) C_k(t+1)$

If $C_k(t) = C_k(t+1)$ then $F_4 > 0$ else $F_4 < 0$

(v) if $W_{sk}^{sc}(t) C_k(t+1) = 0$ then $S(t+1) = S(t)$

And if $W_{sk}^{sc}(t) C_k(t) > 0$ then $S(t) = 1$

Now, $F_4 = +\frac{1}{2} - \frac{1}{2} C_k(t) C_k(t+1) = \frac{1}{2}(1 - C_k(t) C_k(t+1))$

If $C_k(t) = C_k(t+1)$ then $F_4 = 0$ else $F_4 = 1$

(vi) if $W_{sk}^{sc}(t) C_k(t+1) = 0$ then $S(t+1) = S(t)$

And if $W_{sk}^{sc}(t) C_k(t) < 0$ then $S(t) = -1$

Now,

Hence if, $C_k(t) = C_k(t+1)$ then $F_4 = 0$ else $F_4 = 1$

(vii) if , $W_{sk}^{sc}(t) C_k(t+1) > 0$ then $S(t+1) = 1$

And if, $W_{sk}^{sc}(t) C_k(t) = 0$, then, $S(t) = S(t-1)$

Now always, $F_4 = -\frac{1}{2} - \frac{1}{2} C_k(t) C_k(t+1) S(t) \leq 0$.

From equation (3.42) it is clear that for the values of F_1, F_2, F_3, F_4 and F_5 the change in energy is less than or equal to zero. Hence, $\Delta E \leq 0$. Now, here we may examine the ΔE from the equation (3.42) for the cases, where the F_1 to F_5 are greater than zero as one worst case. Thus for the best case and average case it is obvious that $\Delta E < 0$. The ΔE for the worst case can be shown as,

$$\Delta E = -n - \frac{3n}{2} + \frac{n-2}{2} + \frac{n-2}{2} + \frac{1}{2} + \frac{1}{2}$$

$$\Rightarrow \Delta E = \frac{-3n-2}{2}$$

$$\Rightarrow \Delta E \leq 0$$

Hence in this state change policy also for energy situations, we have $\Delta E \leq 0$. Therefore the energy decreases always when either a client or server is selected for the modification. So for each perturbation due to client or server the neural dynamics lead the network towards the minimum energy state or the stable state or the state of equilibrium. The final optimal connection strength at the equilibrium state of the network reflects the optimal distribution of the load between the clients and also from clients to server and server to clients. Thus the network equalizes the load between the nodes for the stability. It implies that the optimal load distribution in the network provides the tolerance against the noise and the distortion in the different nodes.

SIMULATED DESIGN AND IMPLEMENTATION

The experiment described in this segment was designed to determine the optimal load distribution in the client server network and to establish the stability criterion of the network. The experiment also determined the tolerance limit against the perturbations in the nodes. So the stability analysis of the entire network has been conducted in two phases. In first phase the stability of clients' network was established and in the second phase the stability of entire network after introducing the server was investigated. Four different networks of various sizes were considered. The neural network architecture varies with the number of the client nodes. The clients and server nodes are represented as the processing units of the neural network architecture with bipolar state outputs. Initially the connection strength in the neural network was assigned random values and to keep the threshold value of each unit at zero. The nodes are initialized with arbitrary states that are -1 or 1. Hebbian learning was used for connection strength modification. Each network had undergone five trials with the randomly generated states. The energy minimum for each trial was examined and results were used to determine the final state of the network. The conditions used in each experiment are summarized in the following Table 1.

The task of the neural network is to determine the stable state of the network. This stability was determined with the connection strength modification. Final optimal values of the connection strength reflect the distribution of the load between the nodes in the entire network. This optimal distribution of load provides the conditions for stable operation of the network. Thus the network has the tolerance against the perturbations.

It means that the network can adjust itself in a manner that it can retain the stability. Here we are presenting the example to demonstrate the method followed. The algorithm and the final set of equations used for the implementation will be represented here to cover all the cases and state change conditions.

EXAMPLE

Let us consider a client server network comprising of a server S and '4' clients, namely, C_1, C_2, C_3, C_4, All the clients are connected to the system through communication links. If a four neuron input field gets

Table 1. Conditions used in each experiment.

Initial state of the node	Arbitrary value: either 1 or -1
Weight initialization	zero or near to zero
Weight modification	Hebbian learning rule
Connections between clients' nodes	Symmetric
Connections between clients and server	Asymmetric
State update scheme	Asynchronous

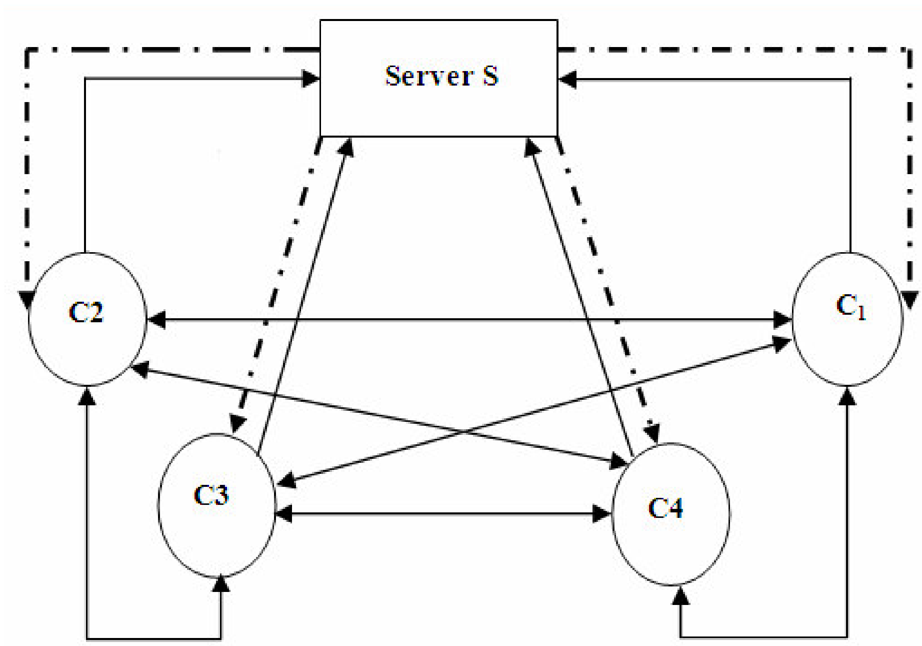

Figure 1. Fully connected client server network.

connected to a fifth neuron, vector inputs from the 4 dimensional space are transformed or mapped to signal vectors for the fifth neuron in a five dimensional space. Thus, the system gets defined as shown in Figure 1.

$$F_3 = \frac{1}{2} \Rightarrow F_3 > 0$$

Figure 1 shows the communication system of client server technology. The server is connected with each client whereas all clients are interconnected in fully connectionist mode. The connection strengths between the clients are symmetric whereas the connections between clients and server are asymmetric. The network of clients was interpreted as the fully connected feedback neural network of Hopfield type. The stability of this network was determined. Thus the final connection strength for network of clients for the minimum energy state was determined. These connection strengths reflect the load distribution in the clients' network to keep the network at equilibrium state. At this stage the server was activated and due to this the entire network became unstable. The connection strength of the network has being modified by using Hebbian rule.

This modification led the network towards the stable state. Test on each network (4 -1, 5 -1, 8 -1 and 12 - 1 client-server configuration) was conducted in order to obtain their energies corresponding to different input probes. Thus the neural dynamics exhibits the stability condition for the network. The algorithm for the entire procedure is shown below:

ALGORITHM

```
Step - 1 input si ( ),
              n← size (si())
Step - 2      for i← 1 until n do
                     wt_mat (,) ← si ()* s (i) T
                     If i: = j   then wt_mat (,) ←0
Step - 3 for j← 1 until n do
         //use randomized index to update weight matrix//
```

$$W_{cicj}(t+1) = W_{cicj}(t) + C_i * C_j$$

Step - 4 for i← 1 until 2^n do

//generate all probe combinations //

// size (probe) = size (si())//

probe (2^n) ←binary (2^n)

Step - 5 for j← 1 until n do

For j← 1 until n do

//determine output states of client nodes //

$$C_i = \sum_{j \neq i}^{n} W_{cicj} * C_j$$

Step-5.1 for i← 1 until n do

for j← 1 until n do

//calculate energy of the client network as//

$$E^{client} = -\frac{1}{2}\sum_{i=1}^{n} \sum_{j=1}^{n} W_{cicj} * C_i * C_j$$

Step - 6 for i← 1 until n do

//compute S and Cj //

$$S = \sum_{i=1}^{n} C_i * W_{ciS}$$

Step-6.1 for i← 1 until n do

for j← 1 until n do

$$C_i = S * W_{ciS} + \sum_{j \neq i}^{n} C_j * W_{ji}$$

Step - 7 for i← 1 until n do

//calculate the weights from sever to clients and clients to server //

$$W_{sci}(t+1) = W_{sci}(t) + \sum_{i=1}^{n} S * C_i$$

Step - 7.1 for i← 1 until n do

$$W_{cis}(t+1) = W_{cis}(t) + \sum_{i=1}^{n} C_i * S$$

Step - 8 for i← 1 until n do //calculate energy of the client server network//

$$E^{Sclient} = -\frac{1}{2}\sum_{i=1}^{n} W_{sci} * C_i * S - \frac{1}{2}\sum_{i=1}^{n} W_{cis} * C_i * S$$

Step - 9 for i← 1 until n do

for j← 1 until n do

//calculate total energy of the system//

$$E^{total} = -\frac{1}{2}\sum_{i=1}^{n} \sum_{j=1}^{n} W_{ci} * C_i * C_j - \frac{1}{2}\sum_{i=1}^{n} W_{sci} * C_i * S$$

Step - 10 n← n+1

For i← 1 until n do

// the final weights are obtained //

$$W^{total} = append \ (W_{sc}, W_{cs}, W_{cc})$$

Step -11 Stop

RESULTS

In this research paper we are analyzing the load distribution in the client server network to maintain the stability of the network. This stability and the tolerance of the network against the perturbations were tested. This task was successfully accomplished by obtaining the minimum energy state of the network by using the Hopfield energy function. Four different networks used in the analysis had 4 -1, 5 -1, 8 -1 and 12 -1 client server constituents. We observe that the average energies for the clients' network in every test network are zero. On the other hand, the average energy of the entire network (N) is also approximately same for every test network. Thus the size of the network does not affect the stability and the minimum energy state of the network because on average the same minimum energy state was obtained for different size of network. The standard deviations for both the energies calculated are as shown in Table 2, and are approximately the same. Thus the minimum energy state of the system is approximately near to the minimum energy state of the clients' network. It means, once the values are obtained for the stability of clients' network the perturbations due to the functionality of server are adjustable and network obtains the stable state, so that the point of stability changes with small derivation, but the region of equilibrium remains the same. Hence, the neural dynamics of the system maintains the equilibrium of entire network. The variance, co-variance and the correlation between client and client-server network have been computed and shown in Table 2. Now from the Table 2 we can also observe that the variance is also approximately same for both situations that is clients' and client - server network.

The regression line was also obtained for various network sizes to represent the nature of stability as the size of network increases. The coding of the algorithm has been done in MATLAB. Several sets of network examples have been tested to check the efficiency of the program. The average and standard deviation of these energies was computed as shown in Table 2.

Conclusion

In this paper we have carried out the study of load distribution in the fully connected client-server network. The client-server network was considered as recurrent network of Hopfield type with symmetric and asymmetric connection strength. It has been discussed in the previous section of this paper that network of client is considered Hopfield type neural network with symmetric connection strength. The entire network involved the single server node and various numbers of clients. The connection strength between the server and clients is asymmetric.

Table 2 . Combined analysis of energy data of client and system network.

Client-server network	Network	Average energy	Standard deviation	Variance	Co - variance	Correlation
4-1	Clients	0.0000	2.5298	6.4000		
	System	-1.4375	2.6323	6.9291	6.1250	0.9810
5-1	Clients	0.0000	3.2128	10.3225		
	System	-1.4687	3.2724	10.7086	10.0625	0.9879
8-1	Clients	0.0000	5.3018	28.1098		
	System	-1.4960	5.3284	28.3921	28.01563	0.9955
12-1	Clients	0.0000	8.1250	66.0161		
	System	-1.4997	8.1405	66.2691	66.00146	0.9981

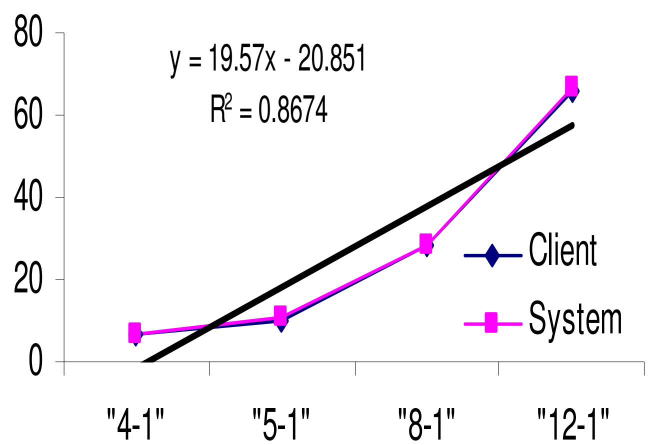

Figure 2. Variance between client and client server networks.

The Hopfield energy function analysis with Hebbian learning rule has been employed to determine the stability of the network. First, the network of clients is stabilized. In this process the connection strength of the client's network is modified and the network leads towards the minimum energy state. After that, the server node is connected with the existing stable network. The participation of server is considered which causes perturbation in the network, so that due to this perturbation the network becomes unstable. Modification in the connection strength of the entire network again leads to the stable state. The final optimal values of the connection strength reflect the optimal distribution of loads in the entire network to keep it in stable state. The results were verified by considering various networks having different size and topological layout. The statistical analysis of the results also reveals that the energies of client's network and the clients-server's network are highly correlated as is evident from Figure 2. Statistical analysis of data set indicates that the energies of client to client network and energy of entire system as a whole, including server, are strongly correlated with each other. Hence it is possible to expand the network from n nodes to n + 1 node by adding another network through a tradeoff between accuracy and tractability. The following observations have been drawn from this work:

- The energy minima for both the networks [N_c and N] are found to be distinct. It has been observed that the equilibrium state of the client's network is different from the equilibrium state of the entire network. Thus the network of clients is allowed to obtain the stable state with minimum energy state on the energy landscape. After that the server node is activated and the entire network leads towards the new minimum energy state. The minimum energy state is found to be different from the minimum energy state of the client's network. Hence, due to the perturbation in network after the activation of server node, the network moves from one stable point to another stable point.

- The difference between energy minima (that is between server-client stability values and client-client stability values are found to be small enough to reflect that both values correspond to the energy of entire system stabilities. Thus the average energy of the stable state for client-client network is found to be zero. The average energy of the entire network is found to be less than the energy of the stable state for client-client network. Hence, the stable state of the server - client network can be considered as the global minima and the stable state of the client-client network can be considered as the local minima. The optimal connection strength in the network is responsible to drive the network in to global minima.

- The final optimal values of the connection strength also reflect the corresponding load distribution among the entire network to maintain the network in the stable state landscape. Thus, on this load distribution, the network is tolerant for the small perturbations. This research work can be extended for the multiple server and multiple clients. The analysis of stabilities for the multiple server case with asymmetric connection strength can lead to interesting problems for future work.

REFERENCES

Guerraoui R, Schiper A (1997). Software-Based Replication for Fault Tolerance, 0018-9162/97, Computer, IEEE 30(4):68-74.

Hopfield JJ (1982). Neural networks and physical systems with emergent collective computational abilities, Proc. Natl. Acad. Sci. pp. 2554-2558.

Hopfield JJ (1984). Neurons with graded response have collective computational properties like those of two-state neurons, Proc. Natl. Acad. Sci. USA 81: 3088-3092.

Logeswaran R (2006). Fault Tolerant Neural Predictors for Compression of Sensor Telemetry Data, J. Universal Comp. Sci. 12(10): 1439-1454.

Hopfield JJ (1987). Learning algorithms and probability distributions in feed-forward and feed-back networks, Proc Natl. Acad. Sci. USA, December 84(23): 8429-8433.

Bart K (1988). Bidirectional Associative Memories, IEEE Transactions on Systems, Man and Cybernetics 18(1): 49-60.

Hebb D (2002). Hebb in perspective, Biol. Cybern. 87: 317–318, DOI 10.1007/s00422-002-0348-8

Hebb DO (1949). The first stage of perception: growth of the assembly, The Organisation of Behaviour - a neurophysiological theory, Wiley, New York pp. 60-78.

Cohen L, Grossberg S (1983). Absolute stability of competitive neural networks, IEEE Transactions on Systems, Man and Cybernetics, SMC 13(5): 815-826.

Ackley D, David F (1985). A Learning Algorithm for Boltzmann Machines, Cognitive Science 9: 147-169.

Zhang Fei-Yu, Hai-Feng H (2006). Global stability of Hopfield neural network under dynamical thresholds with distributed delays. Discrete Dynamics in Nature and Society, Vol. 2006: 1-11, Article ID 27941, Hindawi Publishing Corporation, DOI 10.1155/DDNS/2006/27941.

Zhang FY, Hai-Feng H (1995). Stereo correspondence using Gabor Logons and Neural Networks. 08186-7310-9/95 IEEE pp. 386-389.

Hideki K, Yoichi O (2000). Fast Combinatorial Optimization with Parallel Digital Computers. IEEE Transactions on Neural Networks 11(6): 1323-1331.

Liwanag A, Becker S (1997). Improving Associative Memory Capacity: One-Shot Learning in Multilayer Hopfield Networks. Proc. 19th Annual Conf. Cognitive Sci. Society pp. 442-447.

Ma J (2003). A Hybrid Neural Network of Addressable and Content Addressable Memory. Int. J. Neural Sys. 13(3): 205-213.