Optimal Packet Routing using Hopfield Network
Pankaj Kamani1 Prof. H. B Jethva2
1PG Student 2Associate Professor
1,2Department of Computer Engineering
1,2L.D College of Engineering

Abstract— In this research paper the modelling of shortest path routing problem with artificial neural network (hopfield network) is considered and simulation of that modelling is done. Finally simulation result is compared with traditional routing algorithms to find out the performance of model.

Key words: Neural network, Hopfield network, dynamic routing, congestion control

Abbreviations: NN-neural network

I. INTRODUCTION
Routing is a critical issue in packet-switched networks due to its significant impact on the network performance.

Because of Artificial Neural Network characteristics like computational speed and ability to implement on hardware, it could be a considerable tool to solve shortest path problems. One obvious application of such network might be in computer networks. Route is a list of routers in the path from source to destination node. Shortest path is route with minimum delay and higher reliability in terms of time and packet loss respectively. Our task is to implement the algorithm for finding best route from available once with the help of hopfield network.

One type of ANN that used on solving SP is Hopfield neural network (HNN). Firstly, HNN is introduced by Hopfield and Tank (1985) for solving Traveling Salesman Problem (TSP). The use of HNN to find the shortest path between two nodes in a communication network was initiated by Rauch and Winarske (1988). However, this propose requires a prior knowledge of the network topology like number of nodes in the shortest path. To outperform this limitation, Ali and Kamoun (1993) proposed a novel adaptive algorithm, where the weight matrix just carries convergence information. Eventually, Bastos-Filho et al. (2007) proposed a new method to solve energy function faster.

II. BASIC CONCEPT
The Hopfield neural computational circuit is shown in Fig.2.1.

Fig. 2.1: Hopfield neural network circuit
Each neuron is modelled as a nonlinear device with monotonic increasing sigmoid function with output of $V_i$ that is correspond to its input $W_i$. The output could be any value between 0 and 1 based on input value.

The sigmoid function is as follow

$$V_i = g_i(U_i) = \frac{1}{1+e^{-U_i}}$$  … (1)

Where

- $V_i = \text{output voltage of } i^{th} \text{ neuron}$
- $W_i = \text{weight of the } i^{th} \text{ neuron}$
- $U_i = \text{governs the slope of linearity}$

III. NEURON MODELS FOR SHORTEST PATH
A. Model-I:
In this model first every path between every source and destination is identified and then every possible path is identified and for each that path one neuron is defined.

B. Model-II:
In our second model of Hopfield neural network we are creating a separate neuron for each single link in the network between router we are also maintaining the connection matrix that describe the link status between routers.

This model is as follows:

The edge path representation method is used to represent the HNN. It uses a $n \times n$ binary matrix to represent the edges. Each neuron in the array is identified by double indices $(x, i)$, where $x$ and $I$ indicate the row and column number, respectively. The neuron at location $(x, i)$ shows the link from node $x$ to $i$ in the network. In order to characterize the neuron activities at location $(x, i)$, the neuron state $V_{xi}$ is defined as
Corresponding to each edge \((x, i)\), there is a non-negative weight \(C_{xi}\) representing the cost from node \(x\) to node \(i\).

To compute the best path using the HNN model, the objective function is defined as:

\[
E_{obj} = \frac{1}{2} \sum_{x=1}^{n} \sum_{i=1}^{n} C_{xi} V_{xi}
\]

Since we select the neuron to be activated or deactivated heuristically, there is no need to add the term that checks the availability of only one active neuron in each row and column. Also, the term that checks the generated path contains the start and destination nodes.

\[
V_{xi}(t + 1) = \begin{cases} 
1 & \text{if } net_{xi} > 0 \\
V_{xi}(t) net_{xi} = 0 & \text{if } net_{xi} < 0
\end{cases}
\]

Where

\[
net_{xi} = \sum_{j=1}^{n} W_{xixj} V_{yj} + I_{xi}
\]

\(W_{xixj}\) = weight of link between neuron \(xi\) and \(yj\)

### IV. COMPARISON BETWEEN TRADITIONAL AND NEURAL ROUTING ALGORITHMS

<table>
<thead>
<tr>
<th>Complexity</th>
<th>Computation</th>
<th>Communication</th>
<th>Adaptation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asynchronous distributed Bellman-Ford</td>
<td>(O(N^3))</td>
<td>Each node broadcasts to its neighbors of the latest route cost</td>
<td>Yes</td>
</tr>
<tr>
<td>Dijkstra</td>
<td>(O(N^2))</td>
<td>Each node broadcasts to all other nodes of its connectivity to neighbor nodes</td>
<td>Yes</td>
</tr>
<tr>
<td>Neural nets (JEB 6)</td>
<td>(O(N^3)) no of messages required in learning</td>
<td>none</td>
<td>Yes</td>
</tr>
</tbody>
</table>

### V. CONCLUSION

From this review we can conclude that routing using traditional methods such as hot-potato routing, Dijkstra, floyd-warshall algorithm and bell-men ford for optimal routing in communication network are inefficient for growing demand of network bandwidth and throughput so there required for high-speed and dynamic scheme for optimal routing and hence we are exploring the capabilities of neural network for such purpose.

Many research works shows that better result of using neural network than traditional routing algorithms, so there is a strong chance that future belongs to neural network algorithms for routing.

### REFERENCES


