A Neural Network Shortest Path Algorithm for Routing in Packet-Switched Communication Networks

Michael W. Dixon, Graeme R. Cole, and Matthew I. Bellgard
Computer Science Programme
School of Mathematical & Physical Sciences
Murdoch University
Murdoch, Western Australia 6150

ABSTRACT: This paper presents a Hopfield neural network that solves the routing problem in a communication network. It uses mean field annealing to eliminate the constraint terms in the energy function. Since there are no penalty parameters this approach should avoid the problems of scaling. Computer simulations of the neural network algorithm have shown that it can find optimal or near-optimal valid routes for all origin-destination pairs in a fourteen node communication network.

I. Introduction

The parallel nature of neural network models makes them suitable for simulation on parallel computer architectures. Neural networks are expected to be especially useful in the communications industry. Their promise as pattern recognizers and data correlators is naturally applicable to areas such as switching and queuing, transmission, error-correcting, data compression and routing of data communications traffic.

Neural networks have often been formulated to solve difficult (for the most part, NP-complete) optimization problems. The routing problem may be viewed as an optimization problem where the objective is to minimize total cost when selecting a route for a given origin-destination pair. As a result there has been widespread research into the application of neural networks to the routing problem with varying degrees of success [2, 6, 7, 10, 12]. The major limitations that have been encountered are: i) some systems have failed to consistently converge to a valid route, ii) valid routes are not typically optimal, iii) most of these systems do not scale to larger sized communication networks, and iv) preprocessing is often needed to simplify the problem (e.g. calculating the minimum number of hops from origin to destination).

In this paper, we present a version of the Hopfield model that employs mean field annealing [4, 9, 13, 14] in an attempt to solve the routing problem. The Hopfield model has achieved prominence because of the relative ease of building it into hardware using currently available VLSI technology. Current simulations on a fourteen node communication network have shown that the system converges to a valid route for all origin-destination pairs. These routes are optimal or near optimal. Unlike other systems which use penalty constraint terms there is no need to tune parameters. This tuning has been found to be difficult and problem specific [1, 3]. Also, there is no need to pre-determine the minimum number of hops corresponding to the optimal route.

II. Hopfield Model

In 1985, Hopfield showed how the Hopfield model could be used to solve combinatorial optimization problems of the Travelling Salesman type [5]. The Hopfield model is a fully connected network of simple processing units, V_i, with numerically weighted symmetric connections, T_{ij}, between units V_i, V_j. Processing units have states (either discrete in $\{0,1\}$, or continuous in $[0,1]$ depending on whether the discrete or the continuous version of the network is being considered. Each processing unit performs simple and identical computations which generally involve summing weighted inputs to the unit, applying an internal transfer function, and changing state if necessary. The power of the Hopfield model lies in the connections between units and the weights of these connections. An energy function was defined by Hopfield on the states of the network (values of all the units). The energy function, E, in its simplest form is:

$$E = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} T_{ij}V_jV_i + \sum_{i=1}^{N} V_iI_i \qquad (1)$$

where V_i denotes the current state (value) of the ith neuron and I_i denotes its bias. Hopfield used the fact that the E in (1) is a Liapunov function (bounded from below) to show that, from any starting state, the network would always converge to some energy function minimum upon applying a sequence of asynchronous local state updates (that locally reduce energy). To solve any particular problem, first a decision must be made on how to set the network parameters T and I, so that minimization of the

0-7803-2486-2/95 $4.00 © 1995 IEEE

	1	*2*	*3*	*4*	*5*
1	1	1	0	0	0
2	0	0	0	0	0
3	0	0	1	1	1
4	0	0	0	0	0
5	0	0	0	0	0

Table 1: A neural network representation of a route from node 1 to node 3.

Liapunov function (1) coincides with the minimization of the problem objective function and enforces satisfaction of the problem constraints; this process is termed 'mapping' the problem onto the network. Hopfield gives the motion equation of the ith neuron [5]:

$$\frac{dU_i}{dt} = \frac{U_i}{\tau} - \frac{\partial E}{\partial V_i} \qquad (2)$$

where the output is usually related to the state by a simple nondecreasing monotonic output function $g(U_i)$. Typically, a step function or a hyperbolic tangent function is used.

III. Mapping the Routing Problem Onto the Hopfield Model

NxN neurons will be used to represent a valid route from the source to destination node. The neurons are grouped into *N* groups of *N* neurons which are used to represent the path that the packet (message) will take. In a five node network which is fully interconnected (all the nodes have links to each other) twenty five neurons will be required to represent a route by an output matrix *V*.

The example in Table 1 illustrates a route the packet can take from node 1 to node 3 (e.g. where 1-1-3-3-3 is interpreted as 1-3). The same path can be represented in different ways, such as: 1-3-3-3-3, 1-1-1-3-3 and 1-1-1-1-3. The total length of the route for 1-1-1-3-3 would be

$$l_{route} = d_{11} + d_{11} + d_{11} + d_{13} + d_{33}.$$

Self looping is allowed and there is no cost associated with it. The actual cost would be the d_{13} term in the route.

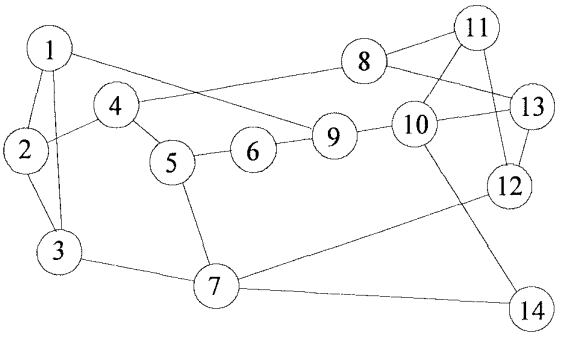

Figure 1: NFSNET-Backbone: 14 nodes and 21 bidirectional links.

For the purpose of demonstration the NFSNET-Backbone topology has been chosen (See Figure 1).

In many combinatorial optimization problems the neurons are grouped into vectors or clusters (each column will be considered a group). For all of these vectors a 1-out-of-N constraint is used, where only one from a group of *N* neurons can be equal to 1 and all other neurons in the group are to be equal to 0. Sometimes the optimization process ends up with a violation of these constraints, i.e. there are some vectors which have all neurons equal to 0 or more than one neuron is equal to 1. This follows from the fact that a neural network may converge to a local minima of the energy function [1, 3, 5]. A way to prevent this from happening is to use mean field annealing (MFA) to replace the classical sigmoid activation function of the form

$$V_i = g(U_i) = \left[1 + \tanh(U_i / T)\right] / 2 \qquad (3)$$

with a multi-dimensional activation function of the form shown in (4) below [4, 8, 13, 14].

$$V_{ia} = \frac{e^{U_{ia}}}{\sum_{b=1}^{N} e^{uib}} \qquad (4)$$

(a =1,2,...,N)(i=1,2,...,N)

According to Cichocki and Unbehauen [4] the activation function generalizes the standard sigmoid function. By employing the multi-dimensional (generalized) activation function (4) the output variables V_{ia} of the neurons grouped in the vector V_i are not independent but always satisfy the constraint that the sum of the column must be equal to 1.

Our proposed energy equation is straightforward because it will only deal with the following distance term and an auxiliary term which adds local positive feedback around neurons which stabilize the neural network and eliminate chaotic behavior (parasitic oscillations) during the optimization process [9]:

$$E = \frac{1}{2}\sum_{x=1}^{N}\sum_{y=1}^{N}\sum_{i=2}^{N-1} GD_{xy}V_{xi}(V_{y,i+1} + V_{y,i-1}) + \frac{P_1}{2}\sum_{x=1}^{N}\sum_{i=1}^{N} V_{xi}^2 \qquad (5)$$

where the subscripts x and y refer to the nodes while the subscript i refers to the position in the route. The GD_{xy} term is a function of both capacity and distance across a link. Decreasing distance and increasing capacity cause the generalized distance to decrease. For the energy function in (5) the generalized MFA equation will take the following form:

$$V_{xi} = \frac{e^{U_{xi}}}{\sum_{j=1}^{N} e^{U_{xj}}}$$

where

$$U_{xi} = -\frac{1}{T}\frac{\partial E(V)}{\partial V_{xi}} = -\sum_{y=1}^{N} GD_{xy}(V_{y,i+1} + V_{y,i-1}) + P_1 V_{xi} \qquad (6)$$

For each given Origin-Destination pair, the first and the last column in the neuron array are fixed (NxM matrix). The states of the neurons not in the first or last column are updated according to equation (6).

IV. Generalized Distance Matrix

We tested this model with the 14 node NFSNET-Backbone shown in Figure 1. The links, nodes and distances of the network can be represented by a 14x14 distance matrix D with entries D_{ij} where each entry D_{ij} represents the distance between node i and node j. For nonexistent links a large value L was used to prevent the neural network from using the links. Also, a 14x14 capacity matrix C with entries C_{ij} was used to represent the capacity between the nodes. All the links were assumed to be bidirectional and to have identical capacity of 100. The lost function used for our simulations is a function of the capacity C_{ij} and the distance D_{ij}. Using

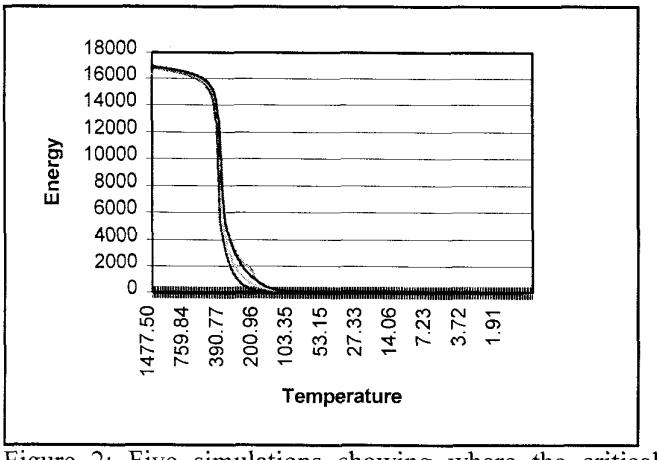

Figure 2: Five simulations showing where the critical temperature is located.

the distance and capacity matrices a (combined) generalized matrix (GD) was calculated as

$$GD_{ij} = \begin{cases} (D_{ij} / C_{ij}) & C_{ij} \neq 0 \\ \infty & C_{ij} = 0 \end{cases} \qquad (7)$$

which is represented in equation (6).

V. Parameter Selection and Sensitivity

The mapping of the routing problem to a neural network was straightforward. The only additional effort needed to complete an application of the MFA method to the routing problem is the selection the MFA equation parameters (temperature T and P_1).

The initial starting temparture was identified by finding a critical point where the energy decreases significantly. This was done by running a number of simulations and plotting the energy with respect to temperature. See Figure 2 for an example of five simulations which show that the energy drops significantly around the 390 mark. Thus, we can set the starting temparature at 500 and avoid unnecessary computations.

The annealing process was stoped when all the neuron values were within the range [0.0, 0.1] or within the range [0.9, 1.0] or when the temperature reached 0.5. The stopping temperature of 0.5 was chosen because the neural network never changed a route after cooling to this point.

The renforcement term P_1 was chosen by running the simulation hundred of times to find the value that gave the best results. We found that tying it to the temperature produced the best results.

VI. Simulation Results

For our simulations the following parameters were set: Starting Temperature of 500, Ending Temperature of 0.5, the P_1 parameter was set equal to Temperature*1.75 and Decay term to 0.985. Since the neural network should have no prior favor for a particular path, all the U_{xi} were set to a random number between -0.005 and 0.005. This random noise helped to break up the symmetry caused by the symmetric network topology. All neuron output voltages V_{xi} were set to 0.5. Each simulation was stopped when the system reached the final temperature of 0.5 or if all the columns met the stopping condition explained in section V.

In order to be valid a route need not be optimum but must not contain any nonexistent links. Between 50 and 400 iterations were required to reach the final solutions. Simulations were run for all origin-destination pairs to find valid routes and the summary of results is shown in Tables 2 & 3.

Hops	*Number of optimal routes*	*Number of near optimal routes*	*Total number of routes*
1	42	0	42
2	68	2	70
3	58	12	70

Table 2: Summary of simulation results for the 14 node NFSNET-Backbone network.

Source	*Dest.*	*Simulations Route*	*Dist.*	*Optimal Route*	*Dist.*
3	5	3-7-5	26	3-2-4-5	23
3	6	3-7-5-6	33	3-2-4-5-6	30
3	10	3-1-9-10	40	3-7-13-10	32
5	3	5-7-3	26	5-4-2-3	23
5	11	5-4-8-11	30	5-6-9-10-11	26
5	14	5-4-8-14	31	5-6-9-10-14	26
6	3	6-9-1-3	40	6-5-4-2-3	30
6	12	6-5-7-12	32	6-9-10-14-12	23
6	8	6-5-4-8	30	6-9-10-14-8	27
8	6	8-4-5-6	30	8-14-10-9-6	27
10	3	10-9-1-3	40	10-13-7-3	32
11	5	11-8-4-5	30	11-10-9-6-5	26
12	6	12-7-5-6	32	12-14-10-9-6	23
14	5	14-8-4-5	31	14-10-9-6-5	26

Table 3: Comparision of all non-optimal simulation routes with actual optimal route.

VII. Discussion

All of the source-destination paths selected were valid and 92.3% contained the optimum route. This rate of optimum route selection is higher than any currently published rate. The neural network was able to find optimal routes where the path was fairly short. The 7.7% of nonoptimal routes were mainly longer routes which contained many hops. As the data shows in Tables 2 & 3, the neural network had greater difficulty converging to a global minimum when the optimum route took more hops to get from the source to destination than did another valid route. Usually, the neural network will favor a route which requires a smaller number of hops, even though the optimum route takes more hops. This can be attributed to how the updates are performed on the neural network. The updates are done left to right (from source to destination) and have introduced an unwanted bias toward selecting paths that have a smaller number of hops versus the shortest total distance. This should be easily corrected by updating the neural network randomly.

Currently, we are investigating ways of ensuring that all routes are optimal routes by using a hill climbing term such as that presented in [11] to prevent the neural network from getting trapped in local minima. Also, more tests on larger networks are being done to ensure that the neural network will always converge to valid routes no matter how large the communication network.

VIII. Conclusion

In this paper, we described a neural network routing methodology which does not use constraint terms. The two advantages of using mean field annealing are the ability to find optimal or near-optimal valid routes in a communication network, and the elimination of the constraint terms in the energy function which should help overcome the scaling problem.

References

[1] V.B. Aiyer, M. Niranjan & F. Fallside, "A Theoretical Investigation into the Performance of the Hopfield Model," *IEEE Transactions on Neural Networks*, Vol. 1, No. 2, pp. 204-215, 1990.

[2] J. Alspector, R. Goodman & T.X. Brown, *Proceedings of the International Workshop on Applications of Neural Networks to Telecommunications*, Lawrance Erlbaum Associates, Inc., Hillsdale, New Jersey, 1994.

[3] A.R. Bizzarrie, "Convergence Properties of a Modified Hopfield-Tank Model," *Biological Cybernetics*, Vol. 64, pp. 293-300, 1991.

[4] A. Cichocki & R. Unbehauen, *Neural Networks for Optimization and Signal Processing*, John Wiley & Sons, 1993.

[5] J.J. Hopfield & D.W. Tank, "Neural Computations of Decision in Optimization Problems," *Biological Cybernetics*, Vol. 52, pp. 141-152, 1986.

[6] F. Kamoun & M.K. Ali, "A Neural Network Shortest Path Algorithm for Optimum Routing in Packet-Switched Communications Networks," *in Proceedings of the Conference Globecomm*, Phoenix, Arizona, pp. 0120-0124, 1991.

[7] Y.C. Ouyang, *Development of a Neural Network Based Routing Algorithm for Communication Networks*, PhD Thesis, Memphis State University, USA, 1992.

[8] Peterson & B. Soderber, "A New Method for Mapping Optimization Problems onto Neural Networks," *International Journal of Neural Systems*, Vol. 1, pp. 3-22, 1989.

[9] C. Peterson, "Parallel Distributed Approached to Combinatorial Optimization: Benchmark on Travelling Salesman Problems," *Neural Computation*, Vol. 2, pp. 261-269, 1990.

[10] H.E. Rauch & T. Winarske, "Neural Networks for Routing Communication Traffic", *IEEE Control Systems Magazine*, Vol. 8, No. 2, pp. 26-31, April 1988.

[11] Y. Takefuji & K. Lee, "An Artificial Hysteresis Binary Neuron: A Model Suppressing the Oscillatory Behavior of Neural Dynamics," *Biological Cybernetics*, Vol. 64, pp. 353-356, 1991.

[12] S.C.A. Thomopoulos, L. Zhang & C.D. Wann, "Neural Network Implementation of the Shortest Path Algorithm for Traffic Routing in Communication Networks," *in Proceedings of the IEEE International Joint Conference on Neural Networks*, Singapore, pp.2693-2702, 1991.

[13] D.E. Van den Bout & T.K. Miller, "Improving the Performance of the Hopfield-Tank Neural Network Through Normalization and Annealing", *Biological Cybernetics*, Vol. 62, pp. 129-139, 1989.

[14] D.E. Van den Bout & T.K. Miller, "Graph Partitionin Using Annealed Neural Networks", *Biological Cybernetics*, Vol. 62, pp. 129-139, 1990.