T-MAG/31/3/1/12122

# Direct Solution Method for Finite Element Analysis Using Hopfield Neural Network

Hideo Yamashita
Norio Kowata
Vlatko Čingoski
Kazufumi Kaneda

Reprinted from
IEEE TRANSACTIONS ON MAGNETICS
Vol. 31, No. 3, May 1995

# Direct Solution Method for Finite Element Analysis Using Hopfield Neural Network

Hideo Yamashita Norio Kowata Vlatko Čingoski Kazufumi Kaneda
Faculty of Engineering, Hiroshima University,
Kagamiyama 1-4-1, Higashi-hiroshima, 724 JAPAN

**ABSTRACT – One property of the Hopfield neural network is the monotonous minimization of energy as time proceeds. In this paper, this property is applied to minimize the energy functional obtained by ordinary finite element analysis. The mathematical representation and correlation between finite element and neural network calculus are presented. The selection of the sigmoid function and its influence on the iteration process is discussed. The obtained results using the proposed method show excellent agreement with theoretical solutions.**

## I. INTRODUCTION

The development of higher efficiency computational methods is necessary to solve accurately and inexpensively complex and higher-dimensional electromagnetic field problems. The implementation of artificial neural networks for the solution of ordinary electromagnetic field problems enables parallel computational process, which provides fast and accurate analysis. A neural network (NN) is an artificial information processing system that simulates the process of human brain [1]. It is a system of similar processing units – neurons with the same input-output characteristics, each of which can be computationally processed separately, enabling multi and parallel processing at the same time.

Different types of NN have already been developed to deal successfully with various numerical problems. The model introduced in 1982 by J. J. Hopfield is one of the most widely used NN models [1]. The main property of the Hopfield neural network (HNN), constructed of interconnected neurons, is to decrease the energy of the network until it reaches a (perhaps local) minimum with the time evolution of the system. This process is very similar to the minimization of the energy functional defined by ordinary finite element analysis (FEA). This similarity, therefore, makes usage of the HNN in ordinary FEA relatively easy. The initial work in this area was done by Ahn, Lee, Lee and Lee [2] (although HNN was not used) in the area of generation of finite element meshes and was also presented in other papers where NN was employed as expert knowledge-based system [3], [4]. Another area where NNs were employed in connection with FEA was in the solution of inverse optimization problems [5], [6].

In this paper, the authors present another application of HNN: direct solution in FEA. First, the mathematical correlation between ordinary FEA and HNN is established and the selection of a sigmoid function is discussed. Then, the application of HNN for directly obtaining the solution of ordinary FEA is verified with several one- and two-dimensional electrostatic and magnetostatic problems. The obtained results are compared with theoretical results and with those obtained by ordinary FEA. The agreement between the results is excellent. Some conclusions and future research problems are also pointed out.

Manuscript received July 5, 1994.

## II. MATHEMATICAL BACKGROUND

### A. Correlation Between Ordinary FEA and HNN

The mathematical correlation between ordinary FEA and HNN, here for simplification, is developed only for a one-dimensional electrostatic model. The governing equation for electrostatic field problems can be expressed as

$$\varepsilon \nabla^2 V = -\rho , \tag{1}$$

where $\varepsilon$ and $\rho$ are permittivity and electric charge density, respectively, and $V$ is electric potential. The development of the energy functional

$$\mathcal{F}(V) = \int_R \left( \frac{1}{2} \varepsilon \, [\nabla V]^2 - \rho V \right) dR , \tag{2}$$

and its discretisation in a one-dimensional domain, leads to the following system of equations

$$\mathcal{F} = \sum_{i=1}^{n-1} \frac{\varepsilon_i}{2} \frac{(V_{i+1} - V_i)^2}{|X_{i+1} - X_i|} - \sum_{i=1}^{n-1} \frac{\rho_i}{2} |X_{i+1} - X_i| \, (V_{i+1} + V_i) . \tag{3}$$

In (3) $n$ is the number of nodes (neurons), and $X_i$ and $V_i$ are the x-coordinate and potential value at node $i$. Here, the total number of finite elements is $n-1$.

On the other side, the energy stored in the HNN can be expressed by [1]

$$\mathcal{E} = -\frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} W_{i,j} V_i V_j - \sum_{i=1}^{n} \Theta_i V_i , \tag{4}$$

where $n$ is the number of neurons in the network, $W_{i,j} = W_{j,i}$ is the weighted function between two arbitrary neurons $i$ and $j$, and $\Theta_i$ is the bias of the neuron $i$.

By comparing (3) and (4), both quadratic forms of potential $V_i$, the weight $W_{i,j}$ and bias $\Theta_i$ of each neuron in the network can be determined easily.

### B. Definition of Input-Output Sigmoid Function

After defining the weight and bias of each neuron in the network, another essential factor is the definition of the input-output sigmoid function of each neuron. Usually

0018-9464/95$04.00 

this input-output function has a sigmoidal shape in the interval [0, 1] and is defined by the following equation

$$Y = \frac{1}{1+e^{-\frac{X}{T}}} , \tag{5}$$

where $Y$ is output value, $X$ is input value and $T$ is the parameter which defines the slope of the function. Due to the nature of FEA, the solution of the problem is usually not restricted to the binary values 1 or 0. On the contrary, the values of the unknown potential could be any real number. We therefore have to generate a sigmoid function which permits output values within the interval $[-\infty, +\infty]$. These output values can be generated by the following function

$$Y = \tan\left\{\frac{\pi}{2}\left(\frac{2}{1+e^{-\frac{X}{T}}} - 1\right)\right\} . \tag{6}$$

Computation of (6) is considerably slow due to several time-consuming operations such as exponential and tangent functions. To overcome this problem, in our research we simplified this equation into the following

$$Y = k \cdot X , \tag{7}$$

where $k > 0$ is the parameter. Another important reason for choosing (7) as the input-output function is that, the first derivations of both the original sigmoid function (5) and our function (7) are always positive.

### C. Processing of the Neural Network

Processing of the constructed neural network can be executed in two different modes: *synchronous* and *asynchronous*. The main characteristic of the synchronous mode is the simultaneous access of the output value of one neuron to other neurons of the network — in other words, the processing moves synchronously through the network. On the other hand, the asynchronous processing mode allows rather random accessing of the input-output values to each neuron in the network. In the asynchronous mode, only after each neuron receives the input value and responds to it by adequate output value using the input-output sigmoid function (7) may we consider that one iteration in the iteration process has been performed. Because the asynchronous mode's random accessing character, each run required a different number of iterations to minimize the energy of the network. We investigated separately the number of required iterations to minimize the energy of the neural network generated by the direct solution of FEA of the model presented in Fig. 7, using both the synchronous and asynchronous processing modes. The obtained results are presented in Table I. As for the asynchronous processing mode, we performed 20 test runs. Among these, only the largest, smallest and average number of iterations are presented. From Table I it is obvious that the asynchronous mode required fewer iterations for reaching the same energy minimum of $10^{-12}$. The number of iterations was decreased by up to 22 percent depending of the parameter $k$ in the input-output sigmoid function (7). The optimal value of parameter $k$, for which the minimization of HNN's energy converges fastest, strongly depends on the analysis model. The minimization of HNN's energy could diverge – for larger value of parameter $k$ than the optimal one, or it could slowly converge towards its minimum – for lower value than the optimal one. Therefore, the optimal value of the parameter $k$, should be determined for each analysis model separately. The existence of the neuron's bias $\Theta$ is determined by the existence of the source area inside analysis region. Model with source area has appropriate value of the bias $\Theta$ and always results in slower iteration process than that of the model without source area.

## III. Application of the Proposed Method

### A. One-dimensional Electrostatic Problem

Two simple one-dimensional electrostatic models, with theoretically known solutions presented in Fig. 1, were directly treated by HNN. In model 1, a constant electric charge density was applied, so the bias $\Theta_i$ in (4) was considered. For model 2 with no electric source, the bias $\Theta_i$ was zero. Other parameters of the models were: length $d = 1$ $[m]$, number of neurons $n = 11$ and number of elements $nel = 10$. The electric parameters together with the imposed boundary conditions are also presented in Fig. 1.

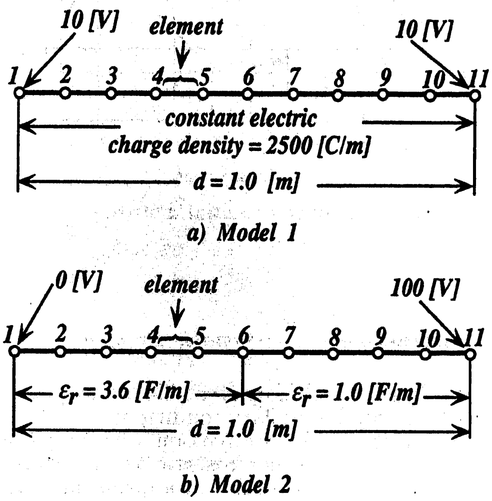


Fig. 1. One-dimensional models.

For both models, the results directly obtained from the solution of the HNN by its energy minimization and the use of the parameter $k = 0.1$ for the sigmoid function (7) are presented in Fig. 2. The minimization process for both models using synchronous and asynchronous modes is presented graphically in Fig. 3. The computed results obtained from the HNN agree with theoretically obtained results up to five significant decimal digits.

### B. Two-dimensional Electrostatic Problem

Following verification of the results obtained directly from HNN for one-dimensional problems, the proposed method was extended and applied for the direct solution of two-dimensional electrostatic problem, which, with its imposed boundary conditions and generated mesh, is presented in Fig. 4. Because no electric source exists in the model, the value of zero was once again considered for the bias $\Theta_i$ in (4). The electric potential distribution obtained directly from the HNN using parameter $k = 0.01$ is presented in Fig. 5. For comparison, in Fig. 6 we see the obtained electric distribution for the same model by ordinary FEA. The uniqueness of both solutions is readily apparent.

Table I: Number of iterations for synchronous and asynchronous processing mode

| $k$ | Number of Iterations | | | | $\frac{(1)-(2)}{(2)} \cdot 100[\%]$ |
|---|---|---|---|---|---|
| | (1) Synchronous Mode | Asynchronous Mode | | | |
| | | Largest | Smallest | (2) Average | |
| 0.20 | divergent | 2705 | 2640 | 2671 | / |
| 0.15 | divergent | 3989 | 3905 | 3955 | / |
| 0.10 | 8419 | 6639 | 6511 | 6567 | 22 |
| 0.05 | 16252 | 14468 | 14190 | 14389 | 12 |

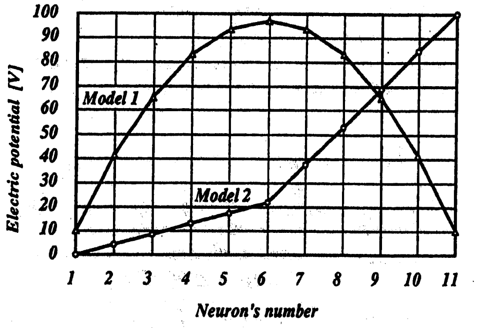


Fig. 2. Electric potential distribution.

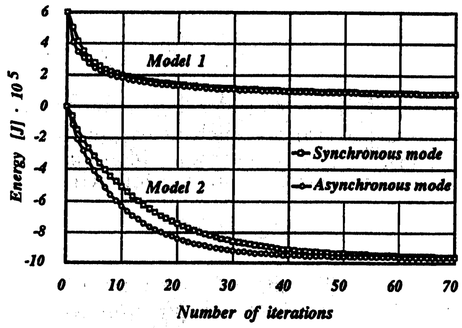


Fig. 3. Minimization of the network energy vs. number of iterations for synchronous and asynchronous modes, respectively.

### C. Two-dimensional Magnetostatic Problem

The two-dimensional magnetostatic model presented in Fig. 7, with imposed boundary conditions was also treated directly by the HNN. In comparison with the aforementioned two-dimensional electrostatic model, this model has a source coil, which results in appropriate values for the bias $\Theta_i$ in (4). Different division maps resulting in different numbers of neurons in the network were considered. An increase in the number of neurons always results in an increase in the accuracy of the obtained results. The distribution of magnetic vector potential $\mathbf{A}$, using the value of the parameter $k = 0.01$ and obtained directly from the solution of the HNN, is presented in Fig. 8. For comparison, the distribution of magnetic vector potential $\mathbf{A}$ for the same model obtained from ordinary FEA is presented in Fig. 9. Both results agree very well.

## IV. Conclusions

In this paper, the authors presented a new application of the HNN as a direct solution of electrostatic and magnetostatic field problems in one- and two-dimensional space, problems usually treated by ordinary FEA. We proved that the HNN handles these problems well, due to its fundamental property of minimizing the network energy while the network evolves over time. With a suitable selection of the sigmoid function and by employing the asynchronous processing mode, the iteration process can be further improved. The fact that the HNN can be used directly for obtaining the solution in FEA is extremely important. This is mainly because in the near future, the development of hardware equipment based on neural networks will open up a wide area for multi and parallel processing in FEA, which clearly will lead to improvements in the computational process overall.

## References


[1] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proc. Nat. Acad. Sci. USA, vol. 79*, pp. 2554-2558, 1982.

[2] C. H. Ahn, S. S. Lee, H. J. Lee and S. Y. Lee, "A self-organizing neural network approach for automatic mesh generation," *IEEE Trans. Magn., vol. 27, No. 5*, September 1991, pp. 4201-4204.

[3] D. N. Dyck, D. A. Lowther, S. McFee, "Determining an approximate finite element mesh density using neural network techniques," *IEEE. Trans. Magn., vol. 28, No. 2*, March 1992, pp. 1767-1770.

[4] D. A. Lowther and D. N. Dyck, "A density driven mesh generator guided by a neural network," *IEEE Trans. Magn., vol. 29, No. 2*, March 1993, pp. 1927-1930.

[5] O. A. Mohammed, R. S. Merchant and F. G. Üler, "Utilizing Hopfield neural networks and an improved simulated annealing procedure for design optimization of electromagnetic devices," *IEEE Trans. Magn., vol. 29, No. 6*, November 1993, pp. 2404-2406.

[6] S. R. H. Hoole, "Artificial neural network in the solution of inverse electromagnetic field problems," *IEEE Trans. Magn., vol. 29, No. 2*, March 1993, pp. 1931-1934.

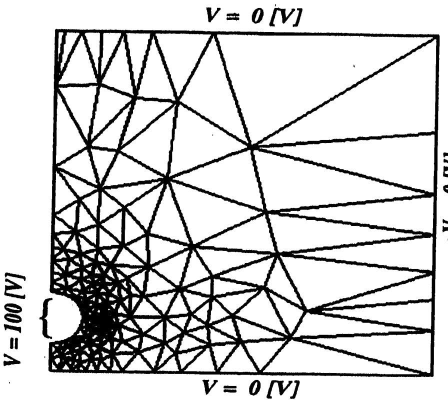


Fig. 4. Two-dimensional electrostatic model.

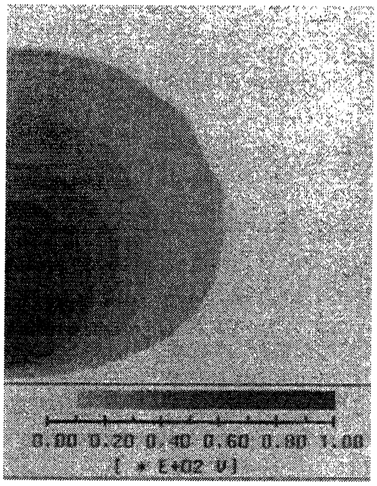


Fig. 5. Electric potential distribution obtained directly by Hopfield neural network.

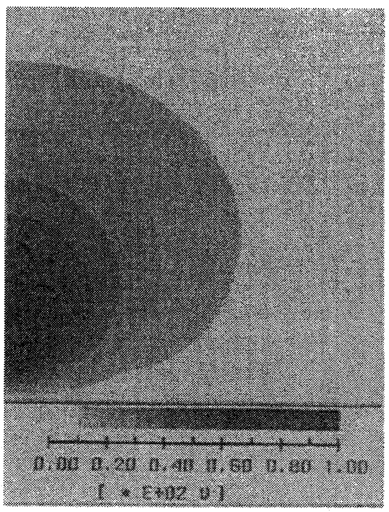


Fig. 6. Electric potential distribution obtained by ordinary finite element analysis.

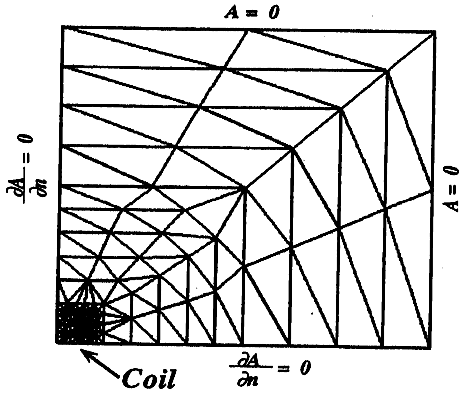


Fig. 7. Two-dimensional magnetostatic model.

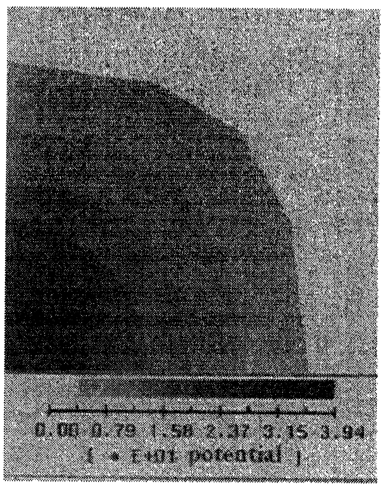


Fig. 8. Magnetic vector potential **A** distribution obtained directly by Hopfield neural network.

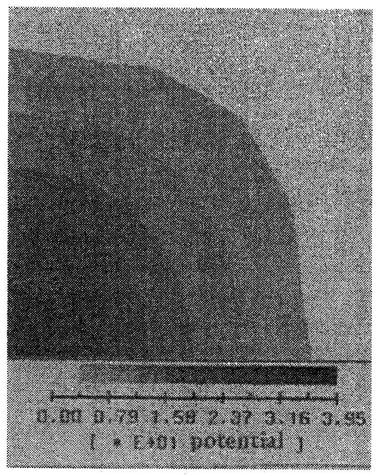


Fig. 9. Magnetic vector potential **A** distribution obtained by ordinary finite element analysis.