PERIODICA POLYTECHNICA SER. EL. ENG. VOL. 38, NO. 3, PP. 221-238 (1994)

THE HOPFIELD NEURAL NETWORK AND ITS APPLICATION FOR DIRECT SOLUTION AND INVERSE OPTIMIZATION IN FINITE ELEMENT ANALYSIS

Hideo YAMASHITA and Vlatko ČINGOSKI

Electric Machinery Laboratory, Faculty of Engineering
Hiroshima University, Kagamiyama 1-4-1
Higashi-hiroshima, Japan
Phone: +81 824 24 7665, Fax: +81 824 22 7195
e-mail: yama@eml.hiroshima-u.ac.jp

Received: Dec. 10, 1994

Abstract

The application of artificial neural network technique and particularly the Hopfield neural network in ordinary finite element analysis is presented. Due to the main property of the Hopfield neural network to minimize the stored network energy, this type of neural network can easily find application in finite element analysis. In this paper two specific applications of the Hopfield neural network will be discussed: First, for obtaining the solution of finite element analysis directly by minimizing the energy of the network – same as minimization of energy functional in ordinary finite element analysis, and second, for obtaining the solution of inverse optimization problems also in connection with finite element analysis. Some basic mathematical calculus and correlations between neural network energy and energy functional that has to be minimized in finite element analysis are discussed. Some application examples to clarify the main idea are also presented.

Keywords: finite element analysis, the Hopfield neural network, inverse optimization problems.

1. Introduction

Artificial neural networks, or short, neural networks (NNs) were first proposed in the early 1960s, but they did not receive much attention until the mid-1980s. Before that time, due to extended development in computer technology, they remained in the experimental stage. A NN is an implementation of an algorithm inspired by research into the brain. It is an artificial information processing system that simulates the process of the human brain, unfortunately, still at a low level. The real breakthrough in NN research came with the discovery of the back-propagation training method, which was widely publicized in the mid-1980s, although discovered in 1974 [1]. Since then and closely connected with the development of fast and relatively inexpensive computers, the interest in NN research has dramatically increased. Different types and construction patterns of NN were discovered

in order to deal successfully with a variety of problems. One of the pioneers in NN research was J. J. HOPFIELD, who quite possibly for the first time in 1982 gave a sophisticated and coherent theoretical picture of how a NN could work, and what it could do [2]. The NN model that he introduced in 1982 [2] and extended in 1984 [3], is still one of the most widely used NN models.

In this model, today called the Hopfield Neural Network (HNN), the interconnected neurons have the main property of decreasing the energy until it reaches a (perhaps local) minimum with the time evolution of the system. This process is very similar to the minimization process of the energy functional defined by ordinary finite element analysis (FEA). This similarity, therefore, enables the usage of HNN in ordinary FEA relatively easy. The initial work in this area was done by AHN, LEE, LEE and LEE [4] (although HNN was not used) in the area of generation of finite element meshes and was also presented in other papers where NNs were employed as expert knowledge-based systems [5]–[6]. Another area where NNs were employed in connection with FEA was in the solution of inverse optimization problems [7]–[10].

In this paper, the authors present another application of HNN for direct solution in FEA [11]. At the same time, new considerations in the area of HNN application in inverse optimization problems are discussed. First, the main properties and construction of HNN are presented, and the mathematical correlation between HNN and ordinary FEA is then determined. Next the procedure for determining the shape of a sigmoid function and its parameters together with its influence on the convergence and accuracy of the obtained results for direct solution in one and two-dimensional FEA, are discussed. Finally, an optimization problem in two-dimensional magnetostatic FEA is presented. Some problems, future research and conclusions are also presented.

2. Hopfield Neural Network in Brief

As mentioned previously, a NN is an attempt to simulate the behavior of the human brain, although the human brain is far more complex than NN models developed currently. In this context, one of the most popular NN models is the HOPFIELD NN [2], [3]. The standard approach to any NN is to propose a learning rule, usually based on already processed data with or without known solution. After the learning procedure is finished, the trained NN may express an appropriate output pattern for new input data which are similar to the data with which it was trained. Hopfield, however, starts by saying that the function of the nervous system is to develop a

number of locally stable points in state space. Other points in state space flow into these stable points. This allows a mechanism for correcting errors, since deviations disappear from the stable points. Then he proceeds with the development of the network that shows this desired behaviour. He assumes that threshold logic units are the basic elements of the network. If the sum of all inputs in one neuron is above that threshold, the neuron responds to a 1, otherwise with a 0 [2], or perhaps to a graded intermediate state between 1 and 0 [3]. In this way the developed network is recurrent, with all the neurons connected, and with the exception that a neuron is connected to itself. Therefore, the connection matrix has zeros along the main diagonal, a NN presented schematically in *Fig. 1.*

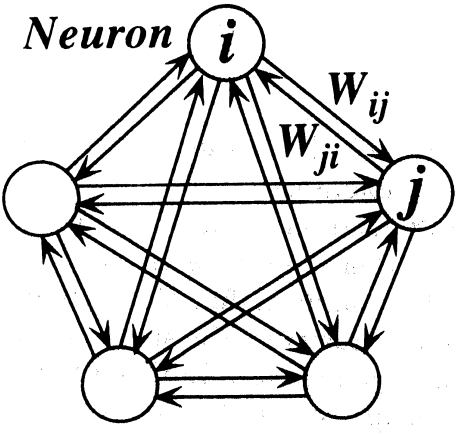


Fig. 1. Hopfield Neural Network

The connectivity is enabled by introduction of the weight function $W_{i,j}$ between any two arbitrary neurons i and j. Another important point is that Hopfield takes into account the special case of the symmetrical matrix, where $W_{i,j} = W_{j,i}$. Then he defines a quantity, called E, which is the sum of all of the terms

$$E = -\frac{1}{2}\sum_{i \neq j} \sum W_{i,j} V_i V_j \,. \tag{1}$$

The quantity E is equivalent to physical *energy.* It can be proved [3] that the energy E is bounded and that as the system evolves, due to its feedback dynamics, the energy decreases until it reaches a minimum. The updating rule of the system is, therefore, an energy minimization rule, where the modification of element activities, actually modification of the weights, continues until a stable state is reached, that is, until the lowest boundary of the energy is reached. This is a fundamental property of HNN that

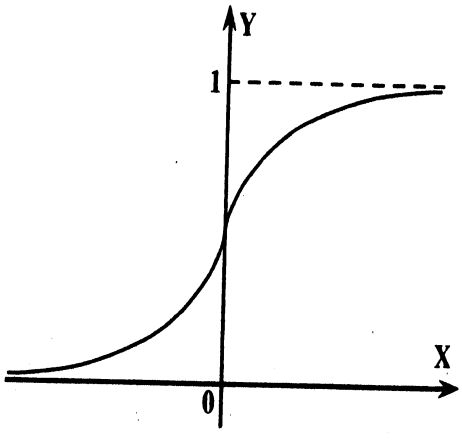


Fig. 2. Sigmoid function

makes it easily applicable in FEA where the solution is to be obtained by minimizing the energy functional.

The input-output relation of each neuron is established through the nonlinear *sigmoid* function, the general shape of which is presented in *Fig. 2.* This sigmoidal nonlinear function is monotone increasing as the sum of inputs increases. However, the slope is very low for large values of the sum of inputs, so large increases in the sum have only a small effect on output activity. The slope of the sigmoid function is low for small values of the sum as well. The slope, together with the function's boundaries, can be freely determined by scaling or shifting. This is advisable and sometimes necessary in order to obtain the desired solution. This sigmoid nonlinear function is usually expressed by the following equation

$$Y = \frac{1}{1 + e^{-\frac{X}{T}}}, \tag{2}$$

where Y is the output value, X is the input value and T is the parameter. Depending on the parameter T, the slope of the sigmoid function changes.

Let us consider more closely the similarities in the mathematical representation between ordinary FEA and HNN. The input of each neuron i came from two sources, external inputs I_i and inputs V_j from other neurons. The total input to neuron i is then

$$H_i = \sum_{j \neq i} W_{i,j} V_j + I_i \,. \tag{3}$$

The total energy in the network is expressed by the following equation

$$E = -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} W_{i,j}V_iV_j - \sum_{i=1}^{n} I_iV_i \,, \tag{4}$$

where n is the number of neurons in the network and $W_{i,j}$ the weight between neurons i and j. The simplified model of the neuron is presented in *Fig. 3.*

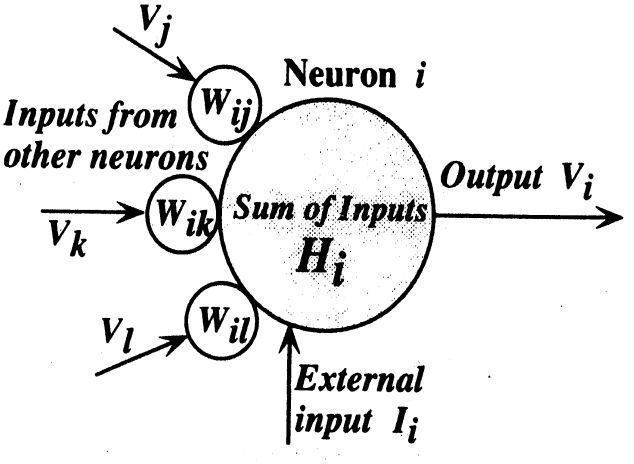


Fig. 3. Model of the neuron

On the other side, the governing equations for electrostatic and magnetostatic field problems can be expressed as

$$\textit{Electrostatics} \qquad \varepsilon\,\nabla^2 V = -\rho \,, \tag{5}$$

$$\textit{Magnetostatics} \qquad \mu^{-1}\,\nabla^2 \boldsymbol{A} = -\boldsymbol{J} \,, \tag{6}$$

where V is electric potential, ε and ρ are permittivity and electric charge density values, respectively. $\boldsymbol{A}$ is magnetic vector potential, $\boldsymbol{J}$ is current density and μ is permeability. The similarities between (5) and (6) are apparent. Using (5) or (6), one can easily develop the energy functional that has to be minimized in FEA. For example, the energy functional for two-dimensional analysis is in case of magnetostatics

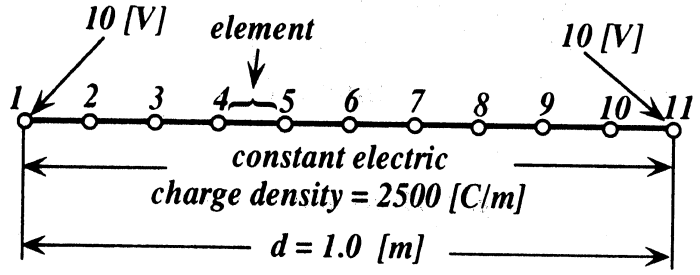
$$\mathcal{F}(\boldsymbol{A}) = \frac{1}{2}\int_{\Omega} \mu^{-1}(\nabla \boldsymbol{A})^2 d\Omega - \int_{\Omega} \boldsymbol{J}\boldsymbol{A}\, d\Omega \,. \tag{7}$$

It is quite easy to recognize the similarities between (4) and (7), or between energy of the HNN and the energy functional in FEA. In other words, the solution obtained by minimization of the energy functional in FEA is equivalent to the solution obtained from the HNN, because the solution of the HNN is derived by minimizing the network's energy.

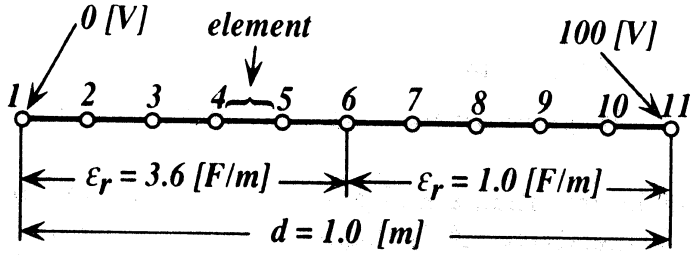
3. Direct Solution in FEA Using Hopfield NN

3.1 One-dimensional Electrostatic Problem

To use HNN for solving directly in FEA, the simple one-dimensional modells presented in *Fig. 4* where developed. The parameters of the models were: length $d = 1$ [m], number of neurons $n = 11$, and number of elements $n - 1$. The electric parameters and the boundary conditions of the models are also presented in *Fig. 4*.



a) Model 1



b) Model 2

Fig. 4. One-dimensional models

The governing equation for electrostatic problems (5) in one-dimensional space leads to the following energy functional

$$\mathcal{F} = \sum_{i=1}^{n-1} \frac{\varepsilon_i}{2} \frac{(V_{i+1} - V_i)^2}{|X_{i+1} - X_i|} - \sum_{i=1}^{n-1} \frac{\rho_i}{2} |X_{i+1} - X_i| (V_{i+1} + V_i) . \quad (8)$$

In the above equation, V_i and X_i are electric potential and x-coordinate at point i in the mesh (neuron i). By analogy between the energy functional (8) and energy of the HNN (4), the weights and external forces for each neuron are determined easily. Extending the energy functional (8), the following matrix equation is obtained

$$\mathcal{F} = [V_1 \quad \cdots \quad V_n] \times \begin{bmatrix} W_{1,1} & W_{1,2} & 0 & \cdots & 0 & 0 \\ W_{2,1} & W_{2,2} & W_{2,3} & \cdots & 0 & 0 \\ 0 & W_{3,2} & W_{3,3} & \cdots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & W_{n-1,n-2} & W_{n-1,n-1} & W_{n-1,n} \\ 0 & 0 & \cdots & 0 & W_{n,n-1} & W_{n,n} \end{bmatrix} \times \begin{bmatrix} V_1 \\ \vdots \\ V_n \end{bmatrix} - [I_1 \quad \cdots \quad I_n] \begin{bmatrix} V_1 \\ \vdots \\ V_n \end{bmatrix} . \quad (9)$$

In the above equation, each element in the matrix of the system is constructed as a weighted sum of the inputs in each neuron. Therefore, its coefficients are

$$W_{1,1} = -\frac{\varepsilon_1}{2} \frac{1}{|X_1 - X_2|} ,$$

$$W_{i,i} = -\frac{\varepsilon_{i-1}}{2} \frac{1}{|X_{i-1} - X_i|} - \frac{\varepsilon_i}{2} \frac{1}{|X_i - X_{i+1}|} ,$$

(for $i \neq 1$ and $i \neq n$)

$$W_{i,i+1} = W_{i+1,i} = -\frac{\varepsilon_i}{2} \frac{1}{|X_i - X_{i+1}|} ,$$

$$W_{n,n} = -\frac{\varepsilon_{n-1}}{2} \frac{1}{|X_{n-1} - X_n|} , \quad (10)$$

$$I_1 = \frac{\rho_1}{2} |X_2 - X_1| ,$$

$$I_i = \frac{\rho_{i-1}}{2} |X_i - X_{i-1}| + \frac{\rho_i}{2} |X_{i+1} - X_i| \quad (\text{for } i = 2, \ldots, n-1) ,$$

$$I_n = \frac{\rho_{n-1}}{2} |X_n - X_{n-1}| . \quad (11)$$

As we could see, the above defined equations describe the generated HNN with diagonal elements whose values are not zero, therefore, different from ordinary HNN. Moreover, the diagonal elements are usually larger than all other coefficients in the matrix of the system. This is result of the nature of the FEA and could not be avoided. Fortunately, the values of diagonal elements are always negative, enabling uniform convergence of the obtained system of equations.

Another important point is the definition of the sigmoid function. Due to the nature of FEA, the solution of the problem is usually not restricted only to the binary values 1 or 0. On the contrary, the values of the unknown potential could be any real number. Therefore, we 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

$$Y = \tan\left\{\frac{\pi}{2}\left(\frac{2}{1+e^{-\frac{X}{T}}} - 1\right)\right\} . \tag{12}$$

Computation of the above equation is considerably slow due to the fact that this function contains several time-consuming operations such as exponential function and, especially, *tangent* function. In order to overcome this problem, in our research we simplified this equation into the following shape

$$Y = kX , \tag{13}$$

where X is the input value, Y is the output value and $k > 0$ is the parameter. Another important reason for choosing this expression is that the first derivatives of both the original sigmoid function (2) and our function (13) are always positive.

The obtained results for the electrostatic problem described in *Fig. 4*, together with the curve of minimization of the energy of the HNN vs. number of iterations, are presented in *Figs. 5* and *6*. The results we obtained agree with analytically obtained results up to four significant decimal digits.

3.2 Two-dimensional Electrostatic and Magnetostatic Problems

The procedure described above was also implemented for solving two-dimensional electrostatic and magnetostatic problems. Here, only the models and the obtained results will be presented.

A two-dimensional electrostatic model with imposed boundary conditions and generated two-dimensional mesh is presented in *Fig. 7*. The electric potential distribution obtained directly from HNN is presented in *Fig. 8*. For comparison, in *Fig. 9* we see the obtained electric potential

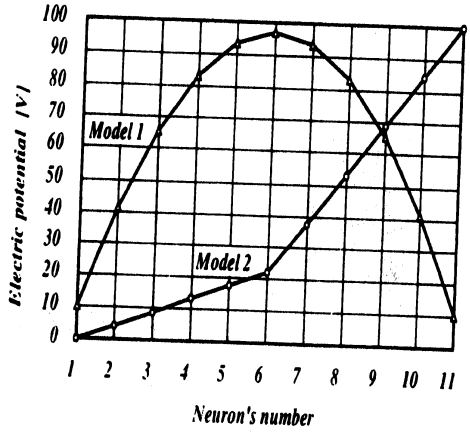


Fig. 5. Electric potential distribution

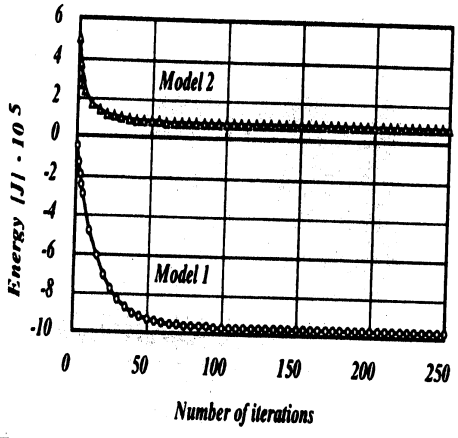


Fig. 6. Minimization of the energy vs. number of iterations

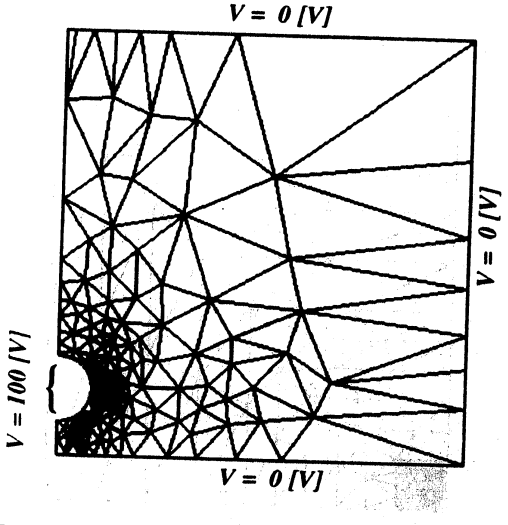


Fig. 7. Two dimensional electrostatic model

distribution for the same model by ordinary FEA. The uniqueness of both solutions is apparent.

The two-dimensional magnetostatic model presented in *Fig. 10* with imposed boundary conditions was also treated directly by HNN. 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 accuracy of the obtained results. The distribution of

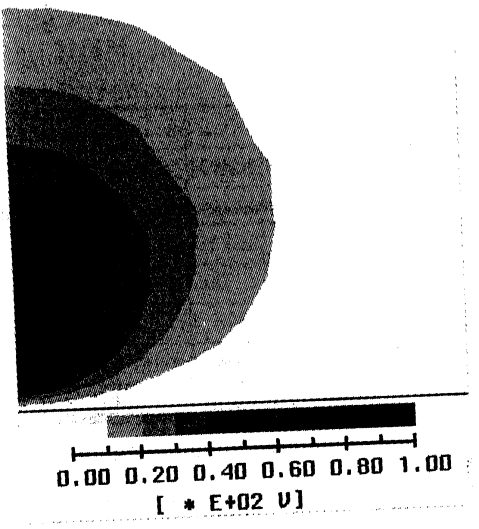


Fig. 8. Electric potential distribution obtained directly by Hopfield Neural Network

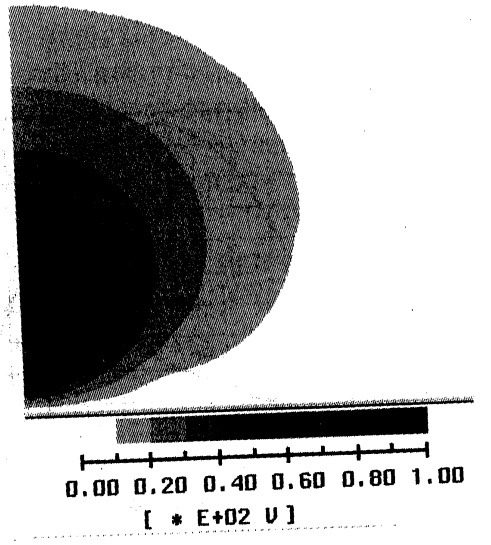


Fig. 9. Electric potential distribution obtained by ordinary finite element analysis

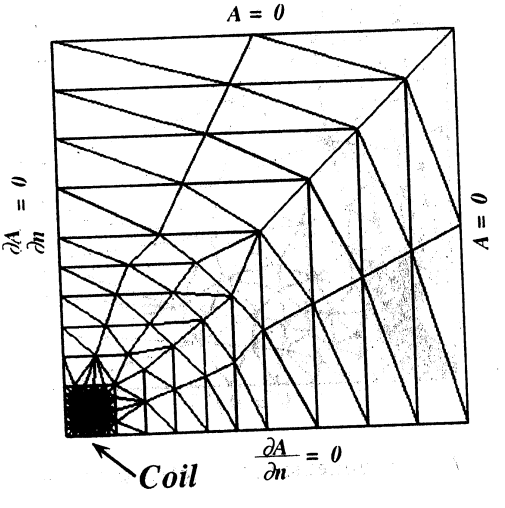


Fig. 10. Two dimensional magnetostatic model

magnetic vector potential $\boldsymbol{A}$ obtained directly from the solution of the HNN is presented in *Fig. 11.* For comparison, the distribution of magnetic vector potential for the same model obtained from ordinary FEA is presented in *Fig. 12.* Both results agree very well.

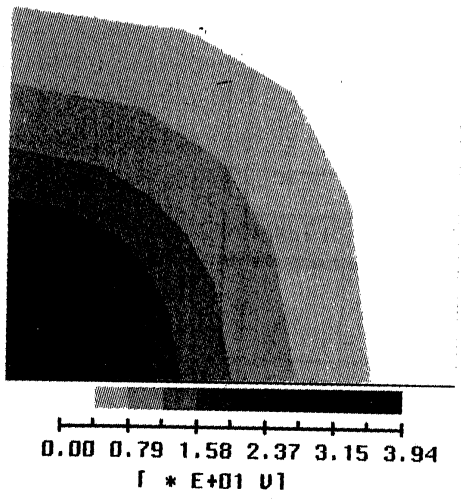


Fig. 11. Magnetic vector potential distribution obtained directly by Hopfield Neural Network

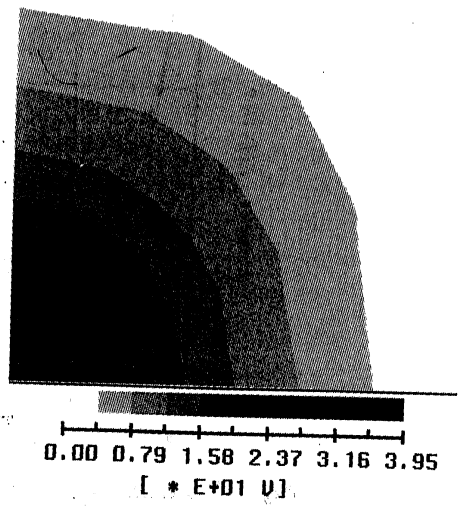


Fig. 12. Magnetic vector potential distribution obtained by ordinary finite element analysis

3.3 Definition of Parameter k in the Sigmoid Function

Let us now consider the effect that parameter k in the sigmoid function (13) has on the iteration process, its convergence, number of iterations and computation time. It was found that minor changes in the value of this parameter change the number of iterations required to minimize the energy of the model, and sometimes make the energy diverge from a stable point, even after a good convergence start. Typical curves of energy minimization for two relatively close values of the parameter k are presented in *Fig. 13.* A different definition of this parameter and changes in the sigmoid function were also considered. However, the procedure for determining this parameter and its automatic adjustment for various problems treated by the network must be considered as one of the problems where future research in this area should be concentrated.

4. Solution of Inverse Optimization Problem by Hopfield NN

In this chapter, we will discuss how HNN may be applied in the solution of inverse optimization problems. Here, a very simple two-dimensional magnetostatic problem is treated, to develop easily the main idea and the method of solution.

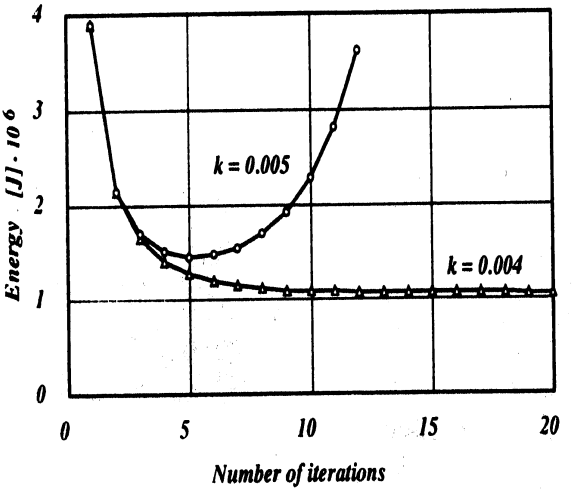


Fig. 13. Influence of parameter k on energy minimization

4.1 Definition of the Problem

Let us consider a two-dimensional magnetostatic model, constructed of a core surrounded by coil in which current with density J flows uniformly in a normal direction shown on the cross-section (*Fig. 14*). Usually in FEA we presume that we know the amount of current density J, and for that amount of current, we compute magnetic vector potential A and magnetic flux density B distributions. This is an ordinary problem. We considered, however, the inverse problem, where we have to determine the amount of source current and the shape and position of the coil to achieve the desired value of magnetic flux density B_k at a certain point k. Therefore, we formulated the following problem:

Determine the amount of current and its distribution (shape and position of the coil) that will result with desired intensity and distribution of magnetic flux density value B_k in a particular point k inside the analyzed region.

We want to solve this problem directly using HNN, i.e., results obtained by minimizing the energy of HNN until the desired solution is reached.

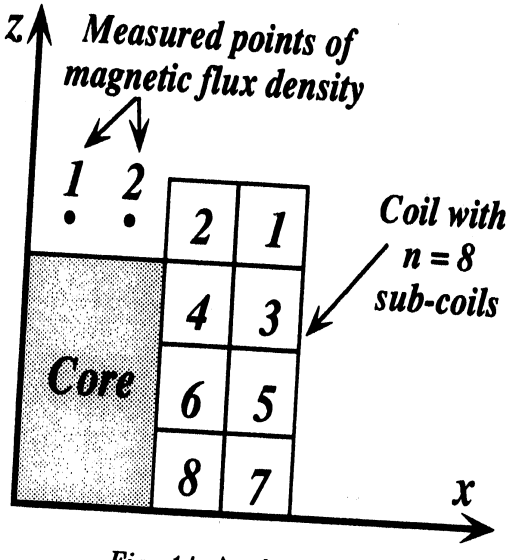


Fig. 14. Analyzed model

4.2 Mathematical Representation of the Problem

By first dividing the coil into an arbitrary number of sub-coils n (in the case of *Fig. 14*, $n = 8$), we could write the energy functional for magnetostatics in two-dimensional space as follows

$$\mathcal{F}(A) = \frac{\nu}{2} \int\int_s \left\{ \left(\frac{\partial A}{\partial x}\right)^2 + \left(\frac{\partial A}{\partial y}\right)^2 \right\} dxdy - \int\int_s p\, JA\, dxdy\,, \quad (14)$$

where ν is reluctivity coefficient, J is current density and A is magnetic vector potential. Parameter p defines which sub-coil is carrying the source current: if $p = 1$ current flow exists, if $p = 0$ current flow does not exist. Using ordinary FEA and expressing the value p for each sub-coil in one vector $\mathbf{p} = [p_1, p_2, \cdots, p_n]$, magnetic flux density B_k at arbitrary point k is calculated by the following equation

$$B_k(\mathbf{p}) = \left(\sqrt{\mathbf{p}^T\, \mathbf{M}\, \mathbf{p}}\right) \cdot I, \quad (15)$$

where $\mathbf{M}$ is the matrix of the system obtained from ordinary FEA, I is the current value, while T stands for transposition. If the desired value of magnetic flux density at point k is B_{0k}, then the minimization of the following functional will be the solution of our problem

$$F = \sum_{k=1}^{m} \left[B_{0k}^2 - B_k^2(\mathbf{p})\right]^2\,, \quad (16)$$

where m is number of points with prescribed value of magnetic flux density B. Consequently, our problem is now:

Determine a vector $\mathbf{p}$ *which minimizes the functional (16).*

Expanding (15) into the second degree Taylor series and inputting into (16), leads to the following equation

$$F = \sum_{k=1}^{m} \left[B_{0k}^2 - \left(B_k^2(\mathbf{p}_e) + B_k^2(\Delta\mathbf{p}) + \mathbf{J} \cdot \Delta\mathbf{p} \right) \right]^2 , \tag{17}$$

where $\mathbf{p}_e$ is a stationary vector and $\mathbf{J}$ is the Jacobian matrix

$$\mathbf{J} = \frac{\partial B_k^2(\mathbf{p}_e)}{\partial \mathbf{p}_e} . \tag{18}$$

For brevity

$$S_k = B_{0k}^2 - B_k^2(\mathbf{p}_e) - B_k^2(\Delta\mathbf{p}) , \tag{19}$$

inputting into (17) and after developing, leads to the following equation

$$F = \sum_{k=1}^{m} \left\{ S_k^2 - 2S_k \sum_{j=1}^{n} J_{kj}\Delta p_j + \left(\sum_{j=1}^{n} J_{kj}\Delta p_j \right)^2 \right\} . \tag{20}$$

Considering $\Delta\mathbf{p}$ in the above equation as an output value from each neuron, the similarities between energy functional (20) and the HNN energy (4) are apparent. For the weights and the external sources, the following relations are valid

$$W_{i,j} = -2 \sum_{k=1}^{m} J_{ki} J_{kj} , \tag{21}$$

$$I_i = 2 \sum_{k=1}^{m} S_k J_{ki} . \tag{22}$$

Therefore, the sum of all inputs in each neuron is a function of time, and its change for a short time interval Δt is expressed (see *Eq.* (3))

$$\begin{aligned} \Delta H_i &= \left(\sum_j W_{i,j}\Delta p_j + I_i \right) \Delta t \\ &= \left(-2 \sum_{j=1}^{m} \sum_{k=1}^{n} J_{ki} J_{kj} \Delta p_j + 2 \sum_{k=1}^{n} S_k J_{ki} \right) \Delta t . \end{aligned}$$

The output value for each neuron is expressed by the following sigmoid function

$$\Delta p = \tanh\left(\frac{H}{u_0}\right), \tag{23}$$

where H is the sum of all inputs in each neuron and u_0 is a parameter. To increase the processing speed of the neural network, the following assumptions were considered:

$$\begin{aligned} p_i > 0.65 &\implies p_i = 1.0\,, \\ p_i < 0.35 &\implies p_i = 0.0\,. \end{aligned}$$

The simplified flow chart of the program is presented in *Fig. 15.*

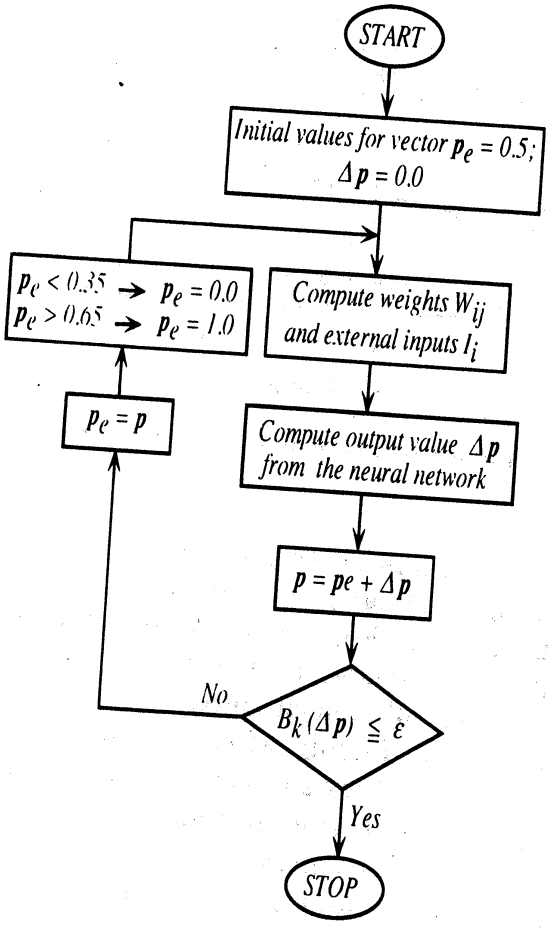


Fig. 15. Flow chart

4.3 Results

The above described procedure was applied on a very simple model presented in *Fig. 14*. The number of sub-coils was 8, while the number of points with prescribed values of magnetic flux density was 2 (see *Fig. 14*). First, ordinary FEA was performed with only three sub-coils excited — coils 1, 2 and 3. The value of magnetic flux density B was obtained at both trial points (see *Fig. 14*). Afterwards, each value of magnetic flux density B was treated as prescribed value B_{0k} in the functional (16), the HNN was constructed and its energy minimized. The obtained results are presented graphically in *Fig. 16*.

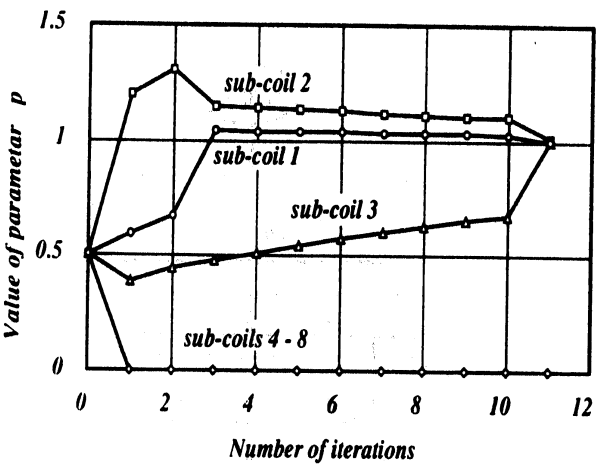


Fig. 16. Results of inverse optimization problem

The number of iterations is on the horizontal axis and the value of the parameter p is on the vertical axis, i. e., indirectly the value of the source current in each sub-coil. From *Fig. 16*, we could see that initially all sub-coils were excited with constant current value ($\mathbf{p}_e = 0.5$). As time passed, the value of p for each sub-coil changed independently of each other, either increasing or decreasing. Finally, after 11 iterations, the value of the current, stabilized to value $p = 1$ for sub-coils 1, 2 and 3, and $p = 0$ for all other sub-coils, which is the same as expected solution.

Division of the coil into more sub-coils enables a more accurate determination of the shape of the coil and actually leads to optimization of its shape and parameters. The computation time in this case increases due to the increase of the number of neurons in the network.

The influence of parameter u_0 in the sigmoid function (24) on the number of iterations was also investigated. The obtained results are pre-

Table 1
Influence of u_0 on iteration process

u_0	Iterations	Error [%]	
		Point 1	Point 2
0.4	no solution	–	–
0.45	14	0.0009	0.0008
0.5	6	0.0009	0.0008
0.6	8	0.0009	0.0008
0.8	9	0.0009	0.0008
1.0	11	0.0009	0.0008

sented in *Table 1*. From *Table 1* we see that the value of parameter u_0 is crucial in obtaining fast and accurate analysis.

5. Conclusions

In this paper, we discussed an application of HNN for the direct solution of electrostatic and magnetostatic problems in one and two-dimensional spaces, usually treated by ordinary FEA. We proved that HNN could be dealt with very well in this area, due to its fundamental property of minimizing the energy of the network while the network evolves with time. Although the computational time using this procedure is of the same rate as some other conventional methods for solution in FEA, such as the ICCG method, the fact that HNN can be used for directly obtaining the solution of FEA is very important. This is mainly because in the near future, the development of hardware equipment based on neural networks, will open a wide area for parallel processing in FEA, which will obviously lead to improvements in the computational process overall. On the other hand, in this paper we also presented the successful application of HNN in the area of inverse optimization problems in FEA. This should find a useful application especially in the design and optimization of different electromagnetic devices in two and three-dimensional spaces. Here, placing the already developed neural network software under neural-network-based hardware would bring a significant improvement in the CAD/CAM systems which are developing as a fast and accurate optimization tool.

As we pointed out in the text, there are still many problems that must be investigated in this research area. Perhaps the most important will be the definition of the sigmoid function, and the development of a procedure for its self-determination depending on the problem. The HNN could then be easily and efficiently applied to various problems in the near future.

References

1. WERBOS, P.: Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences, Ph. D. dissertation, Committee on Appl. Math., Harvard Univ., Cambridge, MA, Nov. 1974.
2. HOPFIELD, J. J.: Neural Networks and Physical Systems with Emergent Collective Computational Abilities, *Proc. Nat. Acad. Sci. USA*, Vol. 79, pp. 2554-2558, 1982.
3. HOPFIELD, J. J.: Neurons with Graded Response Have Collective Computational Properties Like Those of Two-state Neurons, *Proc. Nat. Acad. Sci. USA*, Vol. 81, pp. 3088-3092, 1984.
4. AHN, C. H. – LEE, S. S. – LEE, J. H. – LEE, S. Y.: A Self-organizing Neural Network Approach for Automatic Mesh Generation, *IEEE Trans. Magn.*, Vol. 27, No. 5, September 1991, pp. 4201-4204.
5. DYCK, D. N. – LOWTHER, D. A. – MCFEE, S.: Determining an Approximate Finite Element Mesh Density Using Neural Network Techniques, *IEEE. Trans. Magn.*, Vol. 28, No. 2, March 1992, pp. 1767-1770.
6. LOWTHER, D. A. – DYCK, D. N.: A Density Driven Mesh Generator Guided by a Neural Network, *IEEE Trans. Magn.*, Vol. 29, No. 2, March 1993, pp. 1927-1930.
7. MOHAMMED, O. M. – PARK, D. C. – ULER, F. G. – ZIQIANG, C.: Design Optimization of Electromagnetic Devices Using Artificial Neural Networks, *IEEE Trans. Magn.*, Vol. 28, No. 5, September 1992, pp. 2805-2807.
8. MOHAMMED, O. M. – MERCHANT, R. S. – ULER, F. G.: 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.
9. LOW, T. S. – CHAO, B.: The Use of Finite Elements and Neural Networks for the Solution of Inverse Electromagnetic Problems, *IEEE Trans. Magn.*, Vol. 28, No. 5, September 1992, pp. 2811-2813.
10. HOOLE, S. R. H.: Artificial Neural Network in the Solution of Inverse Electromagnetic Field Problems, *IEEE Trans. Magn.*, Vol. 29, No. 2, March 1993, pp. 1931-1934.
11. YAMASHITA, H. – KOWATA, K. – ČINGOSKI, V. – KANEDA, K.: Direct Solution Method for Finite Element Analysis Using Hopfield Neural Network, *submitted to the Sixth Biennial IEEE CEFC '94*, Aix-lex-Bains, France, 5-7 July, 1994.