Mathematical & Computational Applications, Vol. 8, No. 2, pp. 201-208, 2003

NEURAL NETWORK USING GENETIC ALGORITHM FOR MAGNETIC PERFORMANCE PREDICTION OF TOROIDAL WOUND CORES AT 50 Hz

Ilker Kucuk and Naim Derebasi
Uludag University, Arts and Sciences Faculty, Physics Department,
16059 Gorukle-Bursa, Turkey
ikucuk@uludag.edu.tr, naim@uludag.edu.tr

Abstract- Geometrical and building parameters have a strong influence on magnetic performance of toroidal wound cores made from grain oriented 3% SiFe electrical steel. From a sample of 40 cores with dimensions ranging from 35 to 160 mm outer diameter, 25 to 100 mm inner diameter and 10 to 70 mm strip width and a flux density range of 0.1 to 1.7 T have been obtained and used as training data to a generalised feedforward neural network.

Keywords- Artificial neural network, Genetic algorithm, toroidal wound core, magnetic performance

1. INTRODUCTION

Thousands of tons of electrical steel are used annually in toroidal wound cores for ultimate use in electromagnetic devices. The cores, although simple in geometry are subjected to various parameters which influence their magnetic permeability and the power loss [1]. Because performance of toroidal wound cores is determined by various distinct parameters of the core and knowledge on the corresponding physical processes is quite restricted, analytical descriptions are extremely difficult making the analytical prediction of power loss and permeability impractical.

Artificial neural networks are increasingly becoming useful in the prediction of magnetic performance in electromagnetic devices [2] [3]. This paper presents a neural network used genetic algorithm to predict power losses and relative permeability of grain oriented 3% SiFe toroidal wound cores induction frequency at 50 Hz. Previous measurements from a sample of 40 cores with dimensions ranging from 35 to 160 mm outer diameter, 25 to 100 mm inner diameter and 10 to 70 mm strip width, and a magnetic flux density range of 0.2 to 1.7 T, have been obtained and used as training data to a generalized feedforward neural network, which has 4 input neurons, 2 output neurons model with four hidden layers, and full connectivity between neurons was developed. The input parameters were outer and inner diameters, strip width of toroidal sample and magnetising flux density. The network has been trained using genetic algorithm with the hyperbolic tangent transfer function in hidden layer and output layer. After the network was tested with training data set, the linear correlation coefficient was found to be 99,43% for power loss and 97,88% for permeability. The network outputs are within the acceptable error limits.

2. NEURAL NETWORK

A neural network is an interconnected assembly of simple processing elements, units or neuron, whose functionality is loosely based on the human neuron [4]. The processing ability of the network is stored in the inter-unit connection strengths or weights, obtained by a process of adaptation to, or learning from, a set of training patterns. Models usually assume that computation is distributed over several processing units, which are interconnected and operate in parallel. The most popular neural network is the multi-layer perceptron, which is a feedforward network, i.e., all signals flow in a single direction from the input to the output of the network.

2.1 Multi-Layer Perception and Generalized Feedforward Networks

Multi-layer perceptions (MLPs) are layered feedforward networks typically trained with static back-propagation. These networks have found their way into countless applications requiring static pattern classification. Their main advantage is that they are easy to use, and that they can approximate any input/output map. The key disadvantages are that they train slowly, and require lots of training data (typically three times more training samples than network weights).

The perception and the multi-layer perception are trained with error correction learning, which means that the desired response for the system must be known. This is normally the case with pattern recognition. Error correction learning works in the following way: From the system response at PE i at iteration n, and the desired response for a given input pattern an instantaneous error is defined by

$$e_i(n)=d_i(n)-y_i(n) \qquad (1)$$

Using the theory of gradient descent learning, each weight in the network can be adapted by correcting the present value of the weight with a term that is proportional to the present input at the weight and the present error at the weight

$$w_{ij}(n+1)=w_{ij}(n)+\eta\partial_i(n)\chi_j(n) \qquad (2)$$

Momentum learning is an improvement to the straight gradient descent in the sense that a memory term (the past increment to the weight) is utilized to speed up and stabilize convergence. In momentum learning the equation to update the weights becomes:

$$w_{ij}(n+1)=w_{ij}(n)+\eta\partial_i(n)\chi_j(n)+\alpha(w_{ij}(n)-w_{ij}(n-1)) \qquad (3)$$

where α is the momentum. Normally the α should be set between 0.1 and 0.9. The training can be implemented in two ways: Either we present a pattern and update the weights (on-line learning); or we present all the patterns in the input file (an epoch), store the weight update for each pattern, and then update the weights with the average weight update (batch learning). They are equivalent theoretically, but the former sometimes has advantages in tough problems (ones with many similar input-output pairs).

To start back-propagation, an initial value needs to be loaded for each weight (normally a small random value), and proceed until some stopping criteria is met. Three most common criteria are: The number of iterations, the mean square error of the training set.

Generalized feedforward networks are a generalization of the MLP such that connections can jump over one or more layers. In theory, a MLP can solve any problem that a generalized feed-forward network can solve. In practice, however, generalized feedforward networks often solve the problem much more efficiently. A classic example of this is the two-spiral problem. Without describing the problem, it suffices to say that a standard MLP requires hundreds of times more training epochs than the generalized feedforward network containing the same number of processing elements [5].

3. GENETIC ALGORITHM

The concept of GAs was first proposed by Holland and then described by Goldberg [6]. GAs are stochastic search techniques based on the mechanism of natural selection and natural genetics. The GAs, differing from conventional search techniques, start with an initial set of solutions called a population. Each individual in the population is called a chromosome, and in our context, represent a solution to the problem at hand. A chromosome is a string of symbols; it is usually, but not necessarily, a binary bit string. The chromosomes evolve through successive iterations, called generations. During each generation, the chromosomes are evaluated, using some measures of fitness .To create the next generation, new chromosomes, called offspring, are formed by either (a) merging two chromosomes from the current generation using a *crossover* operation, or (b) modifying a chromosome using a mutation operator. A new generation is formed from this intermediate population by (a) selecting, according to the fitness values, some of the parents and offspring, and (b) rejecting others so as to keep the population size constant. Fitter chromosomes have higher probabilities of being selected. After several generations, the best solution converges, which hopefully represents the optimum or sub optimal solution to the problem. Let P (t) and C (t) be parents and offspring in current generation t ;the general structure of GAs is illustrated in Fig. 1 and described as follows:

Step 1: Set t=0
Step 2: Generate initial population, P (t)
Step 3: Evaluate P (t) to create values
Step 4: *While* (not termination condition) *do*
Step 5: Recombine P (t) to yield C (t),selecting from P (t)according to the values
Step 6: Evaluate C (t)
Step 7: Generate P (t +1) from P (t) and C (t)
Step 8: Set t=t+1
Step 9: *End*
Step 10: Stop

4. ESTABLISHMENT OF TRAINING DATA and MODEL IMPLEMENTATION

A wide range of strip wound cores varied dimensions at 50 Hz magnetizing frequencies have been magnetically characterized [7]. With 40 different combinations of core dimensions and magnetic flux density sweep from 0.2 T-1.7 T, a total of 360 input vectors were available in the training data set.

A developed generalized feed-forward network, which has 4 input neurons, 2 output neurons, 6 neurons of first hidden layer, 4 neurons second, third and fourth hidden layers, and full connectivity between neurons was as shown in Figure 2. The input parameters were outer diameter (d_1), inner diameter (d_2), strip width (h) and the peak value of flux density (B) when the core magnetized at 50 Hz. The output parameters were power loss (P) and relative permeability (μ).

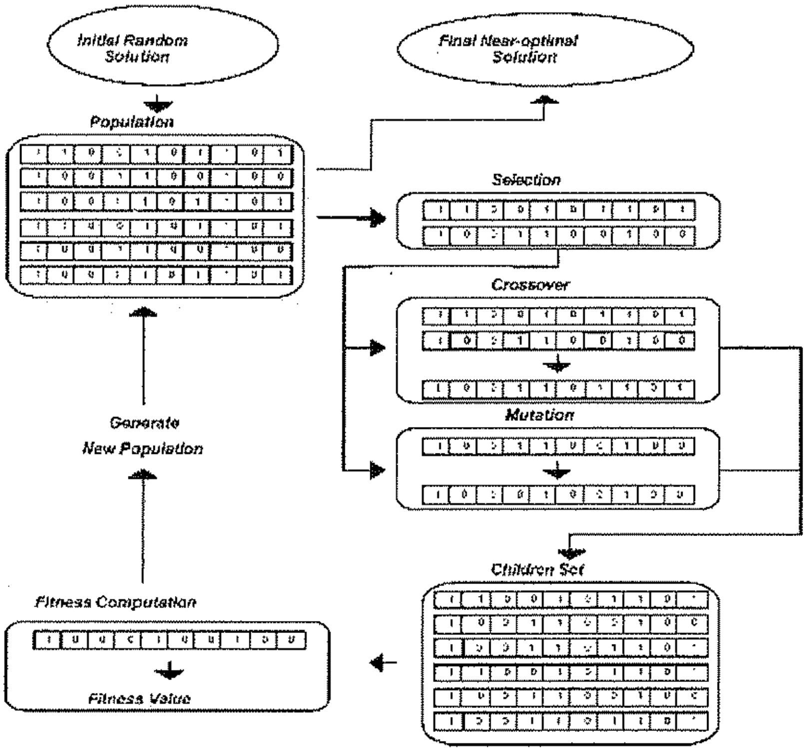


Figure 1. The general structure of genetic algorithm

An evaluation mode of NeuroSolutions 4.01, a commercial neural network package has been used for training the networks, giving the advantage of rapid network development through flexible choices of algorithms, network type, transfer functions and other training parameters, thereby enhancing accuracy.

The network in the Fig. 2 has been trained using genetic algorithm with the hyperbolic tangent transfer function in hidden layer and output layer, 1000 epochs, 50 population size and 100 maximum generations. After the network was trained, best fitness (mean squared error) was found to be 0.00248 as shown Fig. 3. Furthermore, the network was also trained with 1, 2, 3 and 5 hidden layers. But, correlation coefficient in these networks was smaller than the correlation coefficient of the network with 4 hidden layers.

When the network has been tested with training data set, the linear correlation coefficient was found to be 99.43% for the power loss and 97.88% for the permeability. Fig 4 shows the measurement power loss versus the network outputs, for all 360 training data set. The diagonal line in this graph shows perfect match between measurement and network output. Fig. 5 shows the measurement permeability versus the network output for all 360 training data set.

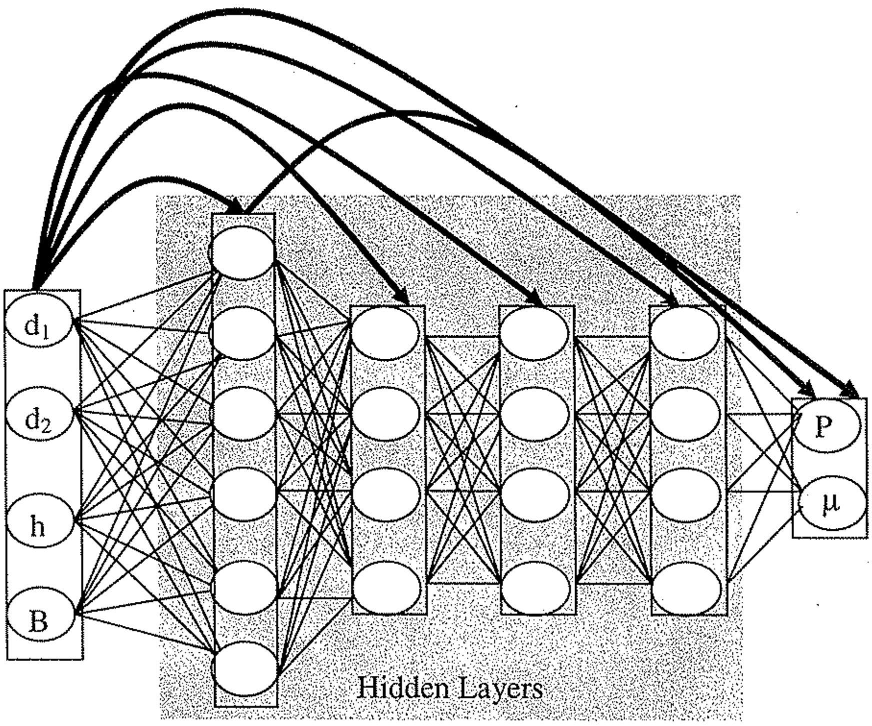


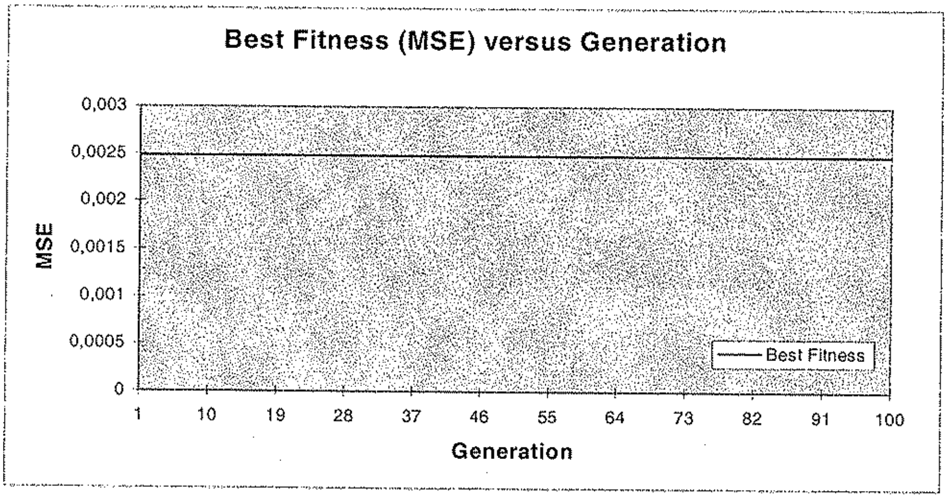
Figure 2. Developed generalized feedforward network with four input, four hidden layers and two output neurons

After the network was trained and the network outputs are within the acceptable error limits, a new data set was tested to clarify the network learned well enough or not. The data set was included original data obtained experimentally from 2 different size toroidal cores. The properties of these cores were as the following

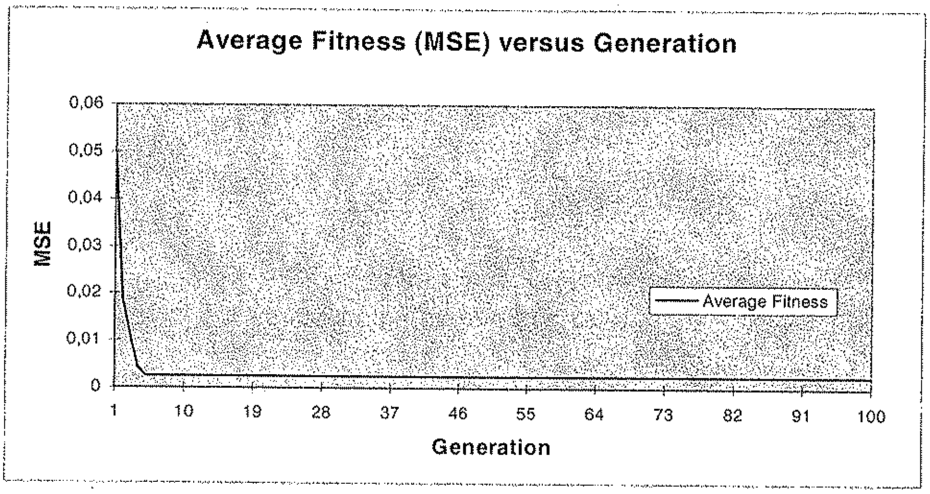
1) The core dimensions are within the limit of trained core data and it was not included the training data set (70x50x10).

2) The core dimensions are out of the limit of trained data and it was not included the training data set (160x100x25).

Table 1 and Table 2 show the predicted power loss and relative permeability for these cores (70x50x10 and 160x100x25) outside training data. In Table 1 and Table 2 the results show excellent predicting capability with maximum error being 17% for power loss and permeability.



(a)



(b)

Figure 3. Best fitness and average fitness versus generation

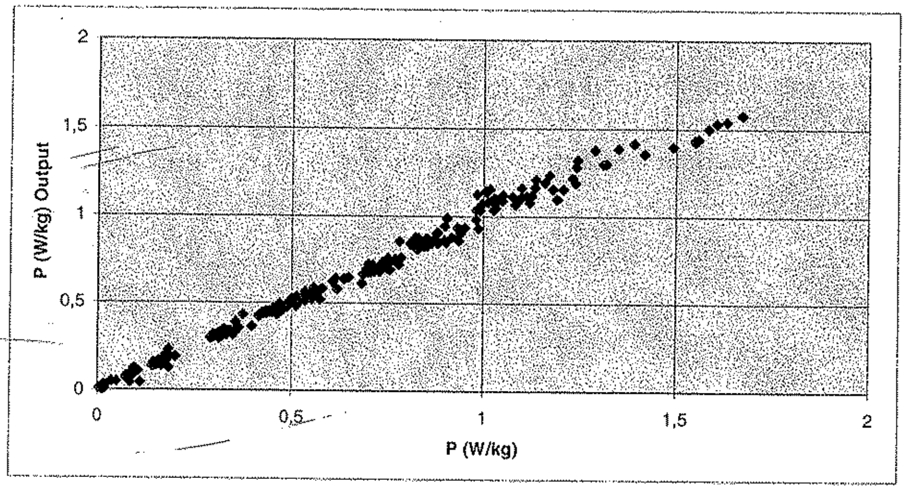


Figure 4. Plot of the measurement power loss versus the network output

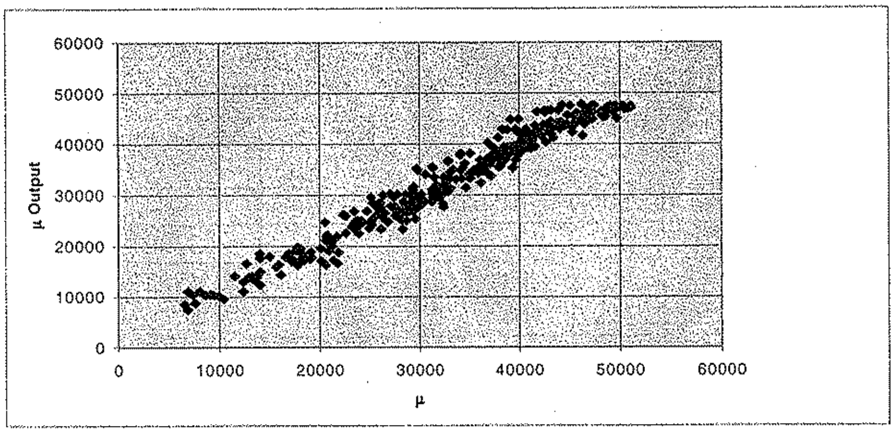


Figure 5. Plot of the measurement permeability versus the network output

Table 1. Comparison between network output and measured power loss

d_1 (mm)	d_2 (mm)	h (mm)	B (T)	Power Loss (W/Kg)	
				Network Output	Measurement
70	50	10	0.2	0.017	0.014
70	50	10	0.5	0.082	0.083
70	50	10	0.7	0.145	0.158
70	50	10	1.2	0.414	0.460
70	50	10	1.5	0.691	0.711
70	50	10	1.6	0.882	0.903
70	50	10	1.7	1.077	1.104
160	100	25	0.2	0.010	0.011
160	100	25	0.5	0.053	0.060
160	100	25	0.7	0.117	0.134
160	100	25	1.2	0.358	0.392
160	100	25	1.5	0.567	0.600
160	100	25	1.6	0.729	0.800
160	100	25	1.7	1.005	1.052

5. CONCLUSIONS

A new neural network model with 4 input neurons, 2 output neurons and four level of hidden layer with 18-neuron was developed for magnetic performance prediction of toroidal wound cores at 50 Hz. After training the network with 360 input vectors the linear correlation coefficient was found to be 99,43% for power loss and 97,88 for permeability. The results shows the artificial neural network and genetic algorithm can be successfully used for magnetic performance prediction of toroidal wound cores. The obtained results have also indicated this to be a promising tool with potential industrial applications.

Table 2. Comparison between network output and measured relative permeability

d_1 (mm)	d_2 (mm)	h (mm)	B (T)	Relative Permeability	
				Network Output	Measurement
70	50	10	0.2	29979	24845
70	50	10	0.5	37010	33301
70	50	10	0.7	40398	37071
70	50	10	1.2	41715	40593
70	50	10	1.5	31810	31120
70	50	10	1.6	25372	24268
70	50	10	1.7	20015	18685
160	100	25	0.2	28401	28142
160	100	25	0.5	35324	37031
160	100	25	0.7	39349	40344
160	100	25	1.2	41762	43159
160	100	25	1.5	28623	30056
160	100	25	1.6	21941	22439
160	100	25	1.7	14758	14971

Acknowledgement- This work was supported by Uludag University, Scientific Research Project Unit grant No 2001/61. The authors would like to thank for this support.

REFERENCES

[1] W. Grimmond, A. J. Moses and P. C. Y. Ling, Geometrical Factors Affecting Magnetic Properties of Wound Cores, *IEEE Trans. on Magnetics*, **25**, 2686-2693, 1989.

[2] O. A. Mohammed, D. C. Park, F. G. Uler and C. Ziqiang, Design optimisation of electromagnetic devices using artifical neural networks. *IEEE Trans. on Magnetics,* **28,** 2805, 1992.

[3] C. Nussbaum, T. Booth, A. Ilo and H. Pfutzner, A neural network for the prediction of performance parameters of transformer cores, *J Magn. and Magn. Mater.,* **160,** 81, 1996.

[4] D.T. Pham, X. Liu, *Principles of Artificial Neural Network,* World Scientific Publishing Co. Pte. Ltd. Singapore 1997.

[5] S. Thrun and F. Smieja, *A general feedforward algorithm for gradient descent learning in connectionist networks.* Int. Rep. German National Res. Center Comp. Sci., 1991

[6] J. H. Jaramillo, J. Buhadury, R. Batta, On the use of genetic algorithms to solve location problems, *Computers and Operations Research,* **29,** 761-779, 2002.

[7] N.Derebasi, R Rygal, A.J. Moses and D. Fox, A Novel System For Rapid Measurement of High Frequency Magnetic Cores of Different Sizes, *J. Magn. and Magn. Mater.*, 215-216, 2000.