Modeling Magnetic Materials using Artificial Neural Networks

H.H.Saliah and D.A.Lowther
CADLab, Electrical Engineering Department, McGill University
Montreal, Quebec, CANADA

B.Forghani
Infolytica Corporation, Montreal, Quebec, CANADA

Abstract-**The accurate and effective modeling of magnetic materials is critical to the prediction of the performance of electromagnetic devices. The paper discusses the use of artificial neural networks as a uniform method for modeling the behavior of magnetic materials both isotropic and anisotropic, and with and without hysteresis.**

Index terms- **Material modeling, Neural networks, Finite element methods, Computer aided engineering.**

I. INTRODUCTION

Magnetic materials, i.e. those that exhibit both permeabilities greater than air and, possibly, hysteresis, are crucial in the design of electromagnetic devices. They can be used as sources of magnetic field (permanent magnets), magnetic conductors, memory systems, etc. In general, they are used to construct the electromagnetic field distributions which are required to accomplish a desired task whether it be the translation of energy between electric, magnetic and mechanical forms or the creation of a particular field structure at specified points in space or the storage of energy.

The design process for such devices has, at its core, a requirement for an accurate prediction of the performance of a device. This requires modeling the material behavior in an accurate manner. In effect, the accuracy of the prediction is controlled by the accuracy of the material modeling.

However, the magnetic properties of materials are dependent on several parameters including temperature and stress. In addition, they all exhibit "memory" to a greater or lesser extent in the form of hysteresis. All of these features mean that constructing a computational model of a magnetic material is a complex and difficult business.

In the past, the methodology used for constructing a computer model of a material has been dependent on the final goal of the analysis package and, since these have been both limited and specialized, the material models have been likewise. For example, if the material is to be modeled as non-hysteretic, the initial magnetization curve can be handled by a polynomial, by piecewise linear segments, by a sequence of cubic splines, etc. If a range of temperatures is to be considered, then a different model for each temperature needs to be constructed. For analyses where the hysteretic properties become important, polynomials [1], [2] have been used as well as, phenomenological models based around Stoner-Wohlfarth and Preisach [3].

Thus, in general, the modeling methodology has been dependent on the characteristics of the material of interest in the analysis. While this has been a satisfactory approach because many of the analysis systems have been very specialized, it is becoming unwieldy to handle multiple representations of the same material in analysis systems which are becoming much more general purpose.

Artificial neural networks provide a means of representing complex multi-dimensional surfaces in a uniform manner. A trained network can be considered as providing a form of least squares fit to the hypersurface defined by the input and output vectors. Thus such a system offers the possibility of creating a uniform model for all the properties of a magnetic material, including hysteretic, thermal and stress effects. In addition, the data and computational requirements are extremely low resulting in an efficient system from a point of view of memory and time.

II. THE DESIGN OF NEURAL NETWORKS

Previous work involving the use of neural networks for modeling magnetic materials, [4], [5], has concentrated on the hysteretic properties and has shown that a conventional feedforward neural network based around perceptron-like neurons, [6], is capable of modeling such aspects of a material. It would seem reasonable to expect that such a model will also be effective for non-hysteretic behaviour. However, little attention was paid in the previous work to the construction of the input vector. To handle the non-linear nature of the data, the network requires at least one hidden layer. The size of the layer can be determined in two ways. The first is a simple trial and error approach - neurons are added to the layer until the network can provide a close fit to the training data. The second is to use a pruning algorithm [7] to remove neurons without affecting the accuracy of the representation. Alternate neuron structures using radial basis functions have also been considered. Such systems lead to improved training times and simplified architectures but the overall behavior is similar to more conventional feedforward systems.

The goal of the network design, then, is to create a structure which can handle the full range of material characteristics including anisotropy and hysteresis. The

Manuscript received November 3, 1997. This work was supported in part by the Natural Science and Engineering Research Council of Canada.
D. A. Lowther, email D_Lowther@compuserve.com

training system has to be constructed such that an effective model is created for the desired application.

Within a finite element analysis system, the material model is subjected to an arbitrary driving field, H. Thus the training scheme for the neural network has to teach it an appropriate response to each arbitrary H value.

III. The Input Vector

The design of the input vector for a neural network is critical. If the inputs are not independent then the training time can increase with little benefit in the modeling. If the input set is not complete then the result will not provide an accurate representation of the system. The proposed method, if it is to eventually handle hysteretic as well as non-hysteretic materials, has to retain a history of the behavior under an arbitrary driving field, H. In a non-hysteretic material, only one input is needed and only one output. If the system is anisotropic, then three inputs are required - the three orthogonal components of the magnetic field - and three outputs need to be considered. If there is no coupling between the three principal directions, i.e. the permeability tensor has no off-diagonal terms, the system is really the equivalent of three independent networks.

However, when hysteresis is considered, the problem becomes somewhat more complex. At any point in the M-H plane, the next state depends not only on the current state but also on the previous state. The current state alone is not sufficient because, for any given (M,H) pair, there are a large number of possible M-H trajectories which pass through the point. Thus enough information has to be provided to determine which trajectory is being followed. The minimum is two states to define the curve. While this is overkill for a non-hysteretic material, the same input vector can handle both hysteretic and non-hysteretic materials, providing a uniform method for handling all forms of material. The complete input vector for a single component of H then consists of 5 variables: the previous (M,H) pair, the current (M,H) pair and the next H. The output is the next value of M. For an anisotropic system, the number of input variables is 15.

IV. The Network Architecture

The architecture is a feedforward neural network trained with a variant of a supervised learning to determine the parameters (the weights and biases). 5 inputs and 1 output are enough to capture and represent the material behavior within a finite element solver. The goal, from the point of view of the analysis system, is to determine the appropriate value for M, the magnetization, given the current value of H, the magnetic field.

The final architecture for each magnetic axis of the material consists of five inputs and six hidden units with a hyperbolic tangent activation function and one output to be predicted, Fig. 1. However, it is also easy to use a single architecture with fifteen inputs (3 x 5) and three outputs to

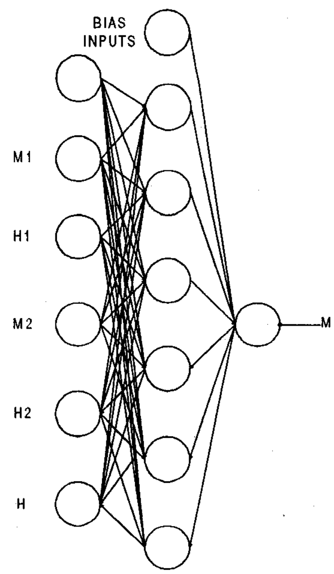

Fig. 1. The 5-6-1 Feedforward Network with Bias.

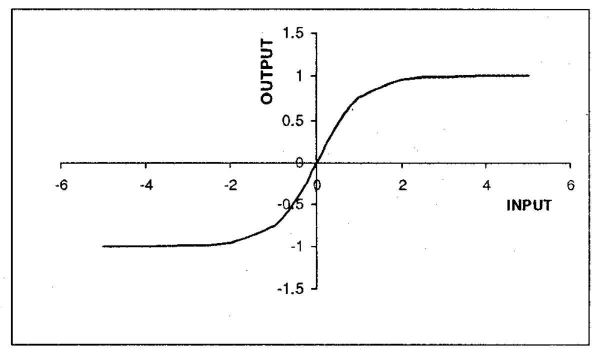

Fig. 2. The activation function for each neuron.

represent a material, [4]. This 5-6-1 model is more flexible and provides a universal framework - the same interface can be used for all types of materials isotropic or not. In the non-hysteretic case, the inputs M1, H1, M2 and H2 are all set to zero since the memory information is not required.

Each neuron consists of a summing unit followed by an "activation" function - this is used to limit the outputs of the network to a specific range and is designed to be differentiable in order to speed up the training of the network. For this problem, the tanh function was used for this purpose, Fig. 2. The training process generates values for the interconnection and bias weights at the input to each neuron and, for the network shown, there are 36 interconnections and 7 bias weights. In the isotropic, non-hysteretic model being considered in this paper, the number of weights needed reduces to 12 for the interconnections and 7 for the bias.

V. The Training Sets

The performance of a feedforward network is determined both by its architecture and the values of the weights associated with the inputs and output of each neuron. The weights are found through a training process in which the network is presented with a large number of sets of the input data and corresponding outputs. The weights are adjusted until the trained response of the network matches the desired response. The set of available data is usually broken into two parts; the first for the actual training, the second for testing the performance of the trained network.

In general, some level of pre-processing is needed to generate an appropriate input set for a neural network. In the particular case of magnetic material modeling, the training is critical and must take into account the way in which the neural network will be expected to function within the analysis system. A finite element solver will tend, at least in the early stages of the process, to oscillate and present increasing and decreasing H magnitudes and thus the network should be able to work effectively with this. The pre-processing scheme used in this case is described below:

1. *The inputs and the output are scaled in such a way that the maximum values of H and M equal one.*
2. *A lookup table is created and more data generated using a cubic spline based model.*
3. *The inputs are processed in a manner that a large spectrum of various data with diversified amplitudes can be used to capture the random nature of a finite element solver's requests.*
4. *The data is resampled at a lower rate after low pass filtering using a Chebyshev filter. This action contributes to reduce the amount of data to be used for the training gracefully and tries for a good generalization (validation tests will be done later using the whole set of data).*

The training algorithm is based on a variant of the Levenberg-Marquardt approach [8].

The learning performance and the final curve generated are given in Figs. 3 and 4

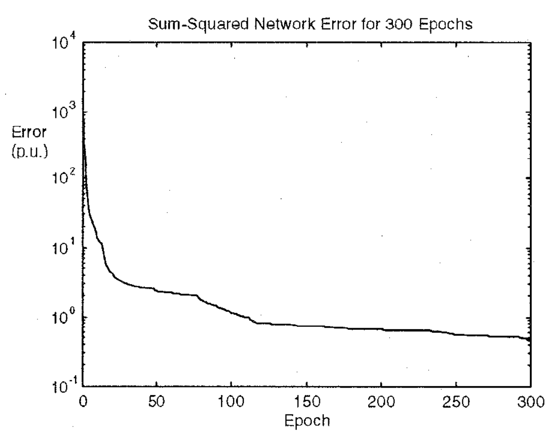

Fig. 3. Learning performance of the network.

VI. Results

The neural network was trained with data representing the M and H values for a typical electrical steel, M19, and an interface was developed to the MagNet program [9] to allow this model to be used instead of the built in magnetization curve model which is based on a set of hermite polynomials.

The two models of M19 were then compared on the same problem, shown in Fig. 5, in terms of the accuracy of the results, the amount of memory used for each model and the operation count in the code. The device chosen was intended to be a simple geometry which would result in a non-uniform flux density distribution in the core. The basic process in each case is that a value of H is provided to the curve model and a new value of M is returned. The problem was non-linear and was solved using a Newton-Raphson process. The intention was to look at testing the generalization of the neural network over a range of H values.

The polynomial model for the material has the advantage that it can return the value of M, the permeability, μ, and $\partial\mu/\partial H^2$. The neural network, at present, returns only M and the calculation of μ and $\partial\mu/\partial H^2$ are done externally. Clearly, it would be appropriate to train the network to generate μ as well as M. Since these calculations are performed using a finite difference approximation, there is likely to be some error introduced in the calculations. Fig. 5 shows the error field, i.e. the difference between B values computed using the two different curve models on the same finite element discretization. The errors have been quantized into about 8 levels of grey and show an average error in the core of less than 1.4%. The solution used a Newton convergence tolerance of 0.01%. The flux densities in the core ranged from 0.9 Tesla (below saturation) to around 2 Tesla. The differences in the solutions are being caused by the differences in the two material curves shown in Fig. 4. However, the inductance computed from the two solutions, a measure of the total energy in the system, agreed to within 0.3%. It should be remembered that the polynomial curve

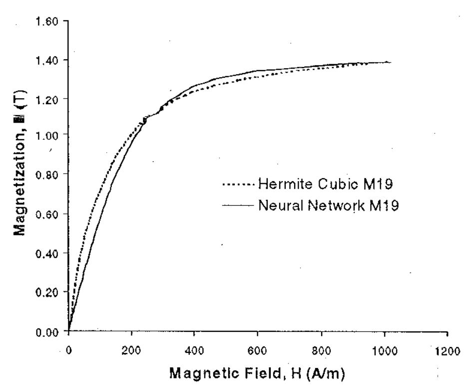

Fig. 4. M-H curve for M19 around the initial magnetization region.

used is also only an approximation to the measured data and thus the error in the core flux densities between predicted and measured could well be of the same order as the errors between the two models.

In terms of the memory required, the polynomial model currently requires 83 floating point numbers to store the data while the neural network requires 19 for the isotropic case. This reduction becomes important when hysteresis is considered. The operation counts for the two methods are given in table I for each model.

The operation count suggests that the neural network model is likely to cost about 20% more on each access in terms of CPU time than the polynomial equivalent. However, this is only a small component of the total solution process and the neural network system was measured as being 10% slower than the polynomial approach. In terms of Newton convergence, the polynomial model required 25 steps while the Neural network took 20. The difference here is probably due to the computation of μ described earlier. The average number of conjugate gradient steps for each Newton step differed by 5 in 100 (the network being larger).

Overall, it appears that the neural network model performs as well as more conventional models and is somewhat more efficient in terms of memory requirements although the amount of memory required for isotropic, non-hysteretic materials is fairly small in any case.

V. The Advantage of the Neural Network

In this approach, each magnetic material to be used in a particular problem would have its own weight set, which characterizes its response, but would use a common network architecture. The network for a non-hysteretic material would be the same as that for a material exhibiting hysteresis thus unifying the representation being used.

In addition, in a magnetic device a material is likely to be operating not only under a distributed set of H values but also temperatures. With the models currently in use, modeling temperature variations within a material requires that a range of curves is constructed. The solution system then interpolates between them to determine an appropriate value for the permeability and/or M. The neural network architecture given above can be modified to include an extra input which represents the temperature of the material. It is then trained with information which includes both the magnetic field and temperature. The final network will be able to generate the appropriate response without the need

TABLE I
Comparison of Operation Counts

	Polynomial	Neural Network
Additions	12	14
Multiplications	18	17
Divisions	3	6
Exponentials	0	2
Comparisons	4	0

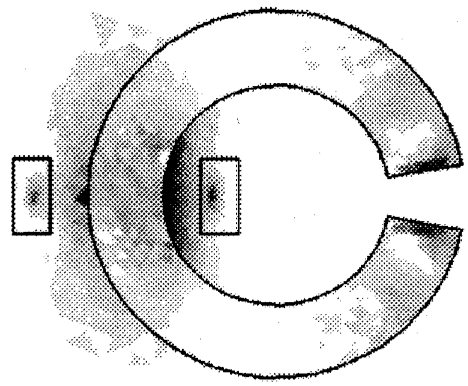

Fig. 5. Error plot between solutions for M19 using hermite cubics and the neual network.

for explicit interpolation. This is likely to both simplify the system, reduce the amount of memory needed to represent a material and reduce the solution times.

VI. Conclusions

The paper has described a possible approach for modeling the magnetic properties of materials, both isotropic and anisotropic, by using a neural network. The approach provides a uniform method of handling all materials, rather than using different representations for different properties. The performance of the system is heavily dependent on the architecture of the network chosen and as well as the input vector. Currently, this is done mostly by experiment since no good theoretical approach exists.

The results given in the paper have demonstrated that the neural network model can provide a computational model which has a cost in terms of time comparable to that of more conventional polynomial based systems.

References

[1] M.L.Hodgdon, "Mathematical theory and calculations of magnetic hysteresis curves", *IEEE Trans. on Magnetics*, vol 24, 1988, pp. 3120-3122.

[2] F.Cortial, F.Ossart, J.B.Albertini, M.Aid, "An Improved Analytical Hysteresis Model and its Implementation in Magnetic Recording Modeling by the Finite Element Method", *IEEE Trans. on Magnetics*, vol 33, 1997, pp. 1592-1595.

[3] I.D.Mayergoyz, "Mathematical models of hysteresis", Springer-Verlag, New York, 1991.

[4] H.H.Saliah and D.A.Lowther, "Magnetic material property identification using neural networks", *Applied Computational Electromagnetics Society Journal*, vol 12, 2, 1997, pp.44 - 49..

[5] H.H. Saliah, D.A.Lowther, B.Forghani, "A neural network model of magnetic hysteresis for computational magnetics", *IEEE Trans. on Magnetics*, vol 33, 1997, pp. 4146-4148.

[6] F.Rosenblatt, "Principles of Neurodynamics: Perceptrons and the Theory of Barin Mechanisms", Spartan Books, New York, 1962.

[7] B.Hassibi, D.G.Stork, "Second order derivatives for networks pruning: optimal brain surgeon", NISP 5, eds S.J.Hanson et al., 164, San Mateo, Morgan Kaufman, 1993.

[8] P.E.Gill, W.Murray, M.H.Wright, "Practical Optimization", Academic Press, London, 1981.

[9] MagNet 5.2 Reference Manual, Infolytica Corporation, Montreal, Canada, 1996.