Cost estimation of Transformer Main Materials using Artificial Neural Networks

[1]H.D. Mehta, [2] R.M. Patel, [3] Mr. P.H. Trivedi

[1]Asso. Prof. at L.D. College of Engg. Ahmedabad, India
[2]Principal, HJD Institute of Technology, Kera, Kachchh,India
[3]Lecturer, Govt. Polytechnic, Bhuj, Kachchh, India

Email: hdm732000@yahoo.com, rajeshpatel@ieee.org, pratik5@ymail.com

Abstract—**This paper presents a method of calculating the cost of transformer main materials using Artificial Neural Networks. In initial phase of project, having knowledge of estimated weights and hence subsequently, the cost in a short time can prove to be beneficial. As the major amount of transformer cost depends on the weight of its main materials, its cost estimation is of vital importance. A Multi-Layer Perceptron (MLP) neural network has been proposed for predicting the weights of aluminium, iron and transformer oil (which are neural network outputs). Once the weights of main materials are known the total cost can be calculated. The inputs to MPNN are kVA rating, maximum flux density, maximum current density and volt per turn. A MATLAB program has been developed to train the neural network. The data required for training the neural network has been obtained from the transformers made by Jyoti Transelect Company, Bhuj, India.**

Keywords—**Artificial Neural Networks(ANN),Conventional Back Propagation (CBP),Cost estimation, MATLAB**

I. Introduction

The transformers are the most vital components of power systems and they play an important role in transmission and distribution of electrical power [1]. In an industrial environment, accurate prediction of transformer cost is an important task for manufacturer and industrial installations. Product Cost Estimation (PCE) deals with predicting the cost of a product before it is manufactured. Due to extremely competitive market there is a need to predict product cost of a transformer early and accurately. However, the available methods of cost estimation compromise accuracy in an attempt to deliver early results. Conversely, the accuracy can only be properly achieved once the design and process planning details are known, but by then the time cost estimation will be too late. The main aim of the work is to develop a methodology for early and accurate estimation of a transformer cost without relying on design and process planning details. If the transformer is designed at lower magnetic induction, and lower current density there would be an increase in transformer cost since more magnetic material and more conducting material is required [2]. Various methods have been suggested for optimal transformer design such as recursive genetic algorithm-FE method for transformer cost minimization problem [3], geometric programming [4], design of power transformers using Artificial Intelligence Techniques [5], transformer design optimization using evolutionary design [6] and some other new techniques [7-8]. were used to design a transformer. The transformer's cost mainly depends on the raw materials which play a crucial role in cost estimation process.

Recently neural networks have been used extensively in the field of transformers such as oil service life identification using neural networks [9], discrimination between inrush and fault current using neural networks [10], parameter estimation of transformer HV winding are some of the few topics on which studies have been carried out.

In this paper an attempt has been made to estimate the transformer cost using Artificial Neural Network (ANN) using the data that were obtained from one transformer manufacturer (Jyoti Transelect Co., Bhuj,). In order to accomplish this, Multilayer Perceptron (MLP) has been trained using the conventional back propagation algorithm for transformer having rating from 25 kVA to 200 kVA transformers manufactured in India. The transformer cost is predicted using suitable co-efficients for the weight of aluminium, iron and oil.

II. AN OVERVIEW OF ANNs

Biological neuronal network comprises of numerous (in the order of trillion) interconnections among the neurons. The power of neuron comes from its collective behavior in a network where all neurons are interconnected. Hypothetically, it is assumed that in a human brain approximately 10 billion neurons are acting collectively, via approximately 10^{14} interconnections. Similarly, ANN also develops from the interconnections of several unit neurons or *nodes*. The arrangement of the neurons is quite arbitrary. It depends on several factors, like, the nature of application, inspiration from real biological structure seen under the microscope and so forth. However, most commonly, as a rule, in the artificial networks, the following layers of neurons are used.

- **Input layer**: The number of neurons in this layer corresponds to the number of inputs to the neural network. This layer consists of passive nodes, i.e., which do not take part in the actual signal modification, but only transmits the signal to the following layer.

- **Hidden layer**: This layer has arbitrary number of layers with arbitrary number of neurons. The nodes in this layer take part in the signal modification, hence, they are active.

- **Output layer**: The number of neurons in the output layer corresponds to the number of the output values of the neural network. The nodes in this layer are active ones.

The basic structure of the ANN is shown in Fig.1

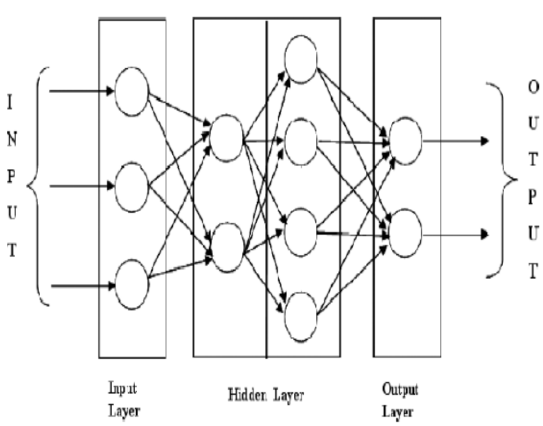

Fig 1. Basic structure of the ANN.

III. NETWORK TRAINING

The objective of training the network is to adjust the weight. So that application of a set of inputs produces the desired set of outputs. These input – output sets can be referred to as vectors. Training assumes that each input vector is paired with a target vector representing the desired output, and these are called a training pair. Usually, a network is trained over a number of training pairs. This group of training pairs is called a training set.

Fig.1 shows the block diagram of a two hidden layer multiplayer perceptron (MLP). The inputs are fed into the input layer and get multiplied by interconnection weights as they are passed from the input layer to the first hidden layer. Within the first hidden layer, they get summed then processed by a nonlinear function (usually the hyperbolic tangent). As the processed data leaves the first hidden layer, again it gets multiplied by interconnection weights, then summed and processed by the second hidden layer. Finally the data is multiplied by interconnection weights then processed one last time within the output layer to produce the neural network output.

To determine the weights of neural network, number of different data are required to train the network. Table- I shows the sample input for 100 kVA transformer which is one of the inputs used for training the neural network.

TABLE-I

Sr. No	Parameter	Input
1	kVA rating	100
2	Volt/turn	3.48
3	Current Density	1.453 A/mm^2
4	Flux Density	1.29 Wb/m^2

The size and hence the weight of a transformer depends on kVA rating. The core area and its weight depends on flux density and volt/turn. Similarly the conductor weight and area depends on current density. Therefore, the above mentioned inputs are used to estimate weights of transformer main materials using ANN

IV. BACK PROPAGATION TECHNIQUE.

The MATLAB program has been developed using back propagation technique [11] given below

1. Normalize the inputs and outputs with respect to their maximum value. It is proved that the neural networks work better if inputs and outputs lie between 0-1. For each training pair, assume there are 'l' inputs given by $\{I\}_{I\,l \times 1}$ and n outputs $\{O\}_{O\,n \times 1}$ in a normalized form.

2. Assume the number of neurons in the hidden layer to lie between l<m<2l

3. [V] Represents the weight of synapses connecting input neurons and hidden neurons and [W] represent the weights of synapses connecting hidden neurons and output neurons. Initialize the weights to small random values usually from -1 to 1. For general

problems, λ can be assumed as 1 and the threshold values can be taken as zero.

$[V]^0$= [random values]

$[W]^0$ = [random weights]

$[\Delta V]^0 = [\Delta W]^0 = [0]$

4. For the training data, present one of set of inputs and outputs present the pattern to the input layer $\{I\}_I$ as inputs to the input layer. By using linear activation function, the output of the input layer may be evaluated as

$$\underset{l \times 1}{\{o\}_I} = \underset{l \times 1}{\{I\}_I}$$

5. Compute the inputs to the hidden layers by multiplying corresponding weights of synapses as

$$\{I\}_w = [v]^r \{O\}_I$$

6. Let the hidden layer units evaluate then output using the sigmoidal function as

$$\underset{m \times 1}{\{o\}_H} = \begin{Bmatrix} \cdot \\ \cdot \\ \frac{1}{1+e^{-I m}} \\ \cdot \\ \cdot \\ \cdot \end{Bmatrix}$$

7. Compute the inputs to output layer by multiplying corresponding weights of synapses as

$$\{I\}_o = [W]^T \{O\}_H$$

8. let the output layers units evaluate the output using the sigmoidal function

$$\{o\}_o = \begin{Bmatrix} \cdot \\ \cdot \\ \frac{1}{1+e^{-I m}} \\ \cdot \\ \cdot \\ \cdot \end{Bmatrix}$$

The above is network output.

9. Calculate the error and difference between the network output and desired output as for the i^{th} training set as

$$E^P = \frac{\sqrt{\Sigma (T_j - O_{OJ})}^{\;2}}{n}$$

10. Find {d} as

$$\underset{n \times 1}{\{d\}_o} = \begin{Bmatrix} \cdot \\ \cdot \\ (T_k - O_{ok}) O_{ok} (1 - O_{ok}) \\ \cdot \\ \cdot \\ \cdot \end{Bmatrix}$$

11. Find the [Y] matrix as

$$\underset{m \times n}{[Y]} = \underset{m \times 1}{\{o\}_H} \; \underset{n \times 1}{\langle d \rangle}$$

12. Find $\underset{m \times n}{[\Delta W]^{t+1}} = \alpha \underset{m \times n}{[W]^t} + \eta \underset{m \times n}{[Y]}$

13. Find $\underset{m \times 1}{\{e\}} = \underset{m \times n}{[W]} \underset{n \times 1}{\{d\}}$

$$\{d\cdot\} = \begin{Bmatrix} \cdot \\ \cdot \\ e_i (O_{Hi})(1 - O_{Hi}) \\ \cdot \\ \cdot \\ \cdot \end{Bmatrix} \qquad m \times 1 \quad m \times 1$$

Find [X] matrix as

$$\underset{1 \times m}{[X]} = \underset{1 \times 1}{\{O\}_I} \; \underset{1 \times m}{\langle d\cdot \rangle} = \underset{1 \times 1}{\{I\}_I} \; \underset{1 \times m}{\langle d\cdot \rangle}$$

14. Find

$$\underset{1 \times m}{[\Delta V]^{t+1}} = \alpha \underset{1 \times m}{[V]^t} + \eta \underset{1 \times m}{[X]}$$

15. Find $[V]^{t+1} = [V]^t + [\Delta V]^{t+1}$

$$[W]^{t+1} = [W]^t + [\Delta W]^{t+1}$$

16. Find error rate as :

$$\text{Error rate} = \frac{\sum E_P}{\text{nset}}$$

17 Repeat steps 4-16 until the convergence in the error rate is less than the tolerance value.

V. EMPIRICAL RESULTS

In this study the data for 25 kVA, 63 kVA, 100 kVA, and 200 kVA (11/0.4 kV) transformers have been used. The inputs to the MATLAB program kVA rating, volt/turn, current density and flux density. The outputs are the estimated weights of aluminium, iron and oil. For training the neural network, 16 training sets have been used. The program has been tested using different values of learning rate, momentum coefficient and hidden neurons. Best results are obtained when learning rate η = 0.6, momentum coefficient α = 0.9 and number of hidden neurons =9. Table –II shows the ANN architecture for generating the results.

TABLE –II

ANN ARCHITECTURE

Number of layers	3
Number of input neurons	4
Number of neurons in hidden layer	9
Number of neurons in output layer	3
Momentum coefficient, α	0.9
Learning rate, η	0.6
Learning rule	Back propagation
Initial weights	Random using MATLAB 'rand' function
Activation function	Tan sigmoid
Training time	91.35 seconds on MATLAB 7.10 (R2010a) using Intel core i3-2100 CPU @3.10 GHz.
No. of epochs	32559

The mean absolute percentage error (MAPE) is used to evaluate the performance ANN architecture. It is defined as:

$$\text{MAPE} = \frac{1}{N}\sum_{i=1}^{N}\left[\frac{P_{actual\,i}-P_{predicted\,i}}{P_{actual\,i}}\right]$$

Tables-III, IV and V show the comparison between actual weights and weights of aluminium, iron and oil predicted using ANN

TABLE – III
Comparison between actual Aluminium weight and weight predicted using ANN

kVA rating	Actual Wt(kg)	Predicted Wt (kg)	Absolute Error	% Error
25	38.7	37.58805546	0.025306	2.530588
25	36.45	38.45349847	0.054966	5.496566
25	47.4	47.71833029	0.006716	0.671583
25	39.9	39.25863297	0.016074	1.607436
63	70.8	68.00569554	0.039468	3.946758
63	55.65	57.57183701	0.034534	3.453436
63	47.55	45.97287554	0.033168	3.316771
63	50.5	52.65886061	0.04275	4.274972
100	90	90.57422754	0.00638	0.638031
100	82.9	80.14818957	0.033194	3.319434
100	81.6	79.38521667	0.027142	2.714195
100	78.45	78.35843253	0.001167	0.116721
200	135.3	137.9460438	0.019557	1.955686
200	142.3	137.5918356	0.033086	3.308619
200	137.2	138.1022582	0.006576	0.657623
200	124.5	125.3538004	0.006858	0.685783

Mean absolute percentage error is 2.41 %

TABLE-IV
Comparison between actual core weight and weight predicted using ANN

kVA rating	Actual Wt (kg)	Predicted Wt (kg)	Absolute Error	% Error
25	72.2	70.37291567	0.025306	2.530588
25	51.3	51.24969436	0.000981	0.098062
25	75.85	75.28366174	0.007467	0.746656
25	66.6	66.5054246	0.00142	0.142005
63	121.3	121.7957485	0.004087	0.408696
63	138.1	140.6184832	0.018237	1.823666
63	144.78	144.9888495	0.001443	0.144253
63	133.15	133.2757799	0.000945	0.094465
100	171.85	171.1628671	0.003998	0.399845
100	213.1	203.1768723	0.046566	4.656559
100	152.3	157.9129654	0.036855	3.685466
100	229.5	230.8925376	0.006068	0.60677
200	343.75	352.6023606	0.025752	2.575232
200	360.8	354.4222526	0.017677	1.767668
200	352.3	352.9311944	0.001792	0.179164
200	356.5	355.3527606	0.003218	0.321806

Mean absolute percentage error is 1.26 %

TABLE-V
Comparison between actual oil weight and oil weight predicted using ANN

kVA rating	Actual Wt (kg)	Predicted Wt (kg)	Absolute Error	% Error
25	125	123.9403061	0.008478	0.847755
25	118	118.2047	0.001735	0.173475
25	106	103.6479038	0.02219	2.218959
25	120	120.8168882	0.006807	0.68074
63	145	141.7045773	0.022727	2.272705
63	145	146.1128161	0.007675	0.767459
63	170	165.6334872	0.025685	2.568537
63	170	169.9317428	0.000402	0.040151
100	190	184.3231352	0.029878	2.987824
100	210	226.870033	0.080333	8.033349
100	231	217.7972947	0.057155	5.715457
100	250	240.8680249	0.036528	3.65279
200	300	290.4987556	0.031671	3.167081
200	280	288.2067841	0.02931	2.930994
200	290	289.8802272	0.000413	0.041301
200	320	313.144891	0.021422	2.142222

The mean absolute percentage error is 2.39 %

The typical experimental results obtained above have been presented in figs. 2, 3 and 4 which indicates accuracy of ANN in predicting the weights of transformer main materials

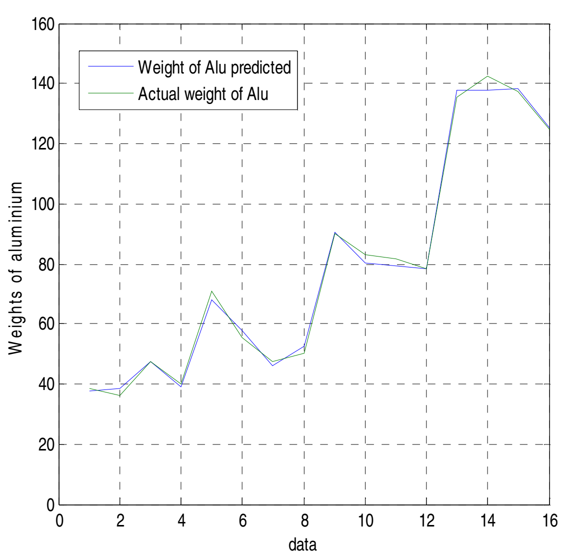

Fig.2 Comparison of actual Aluminium weight and weight predicted using ANN

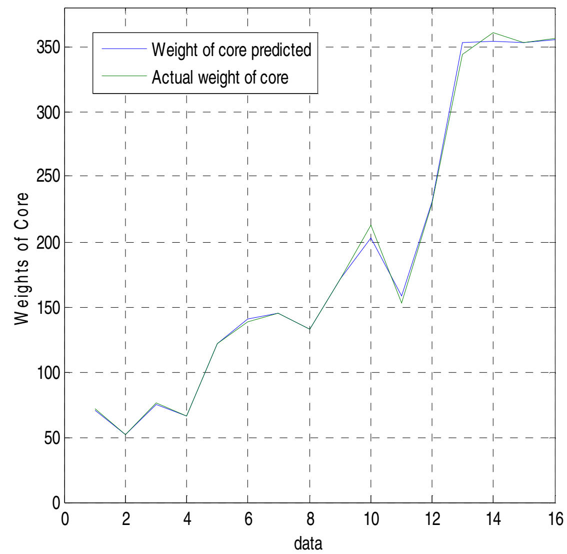

Fig.3 Comparison of actual core weight and weight predicted using ANN

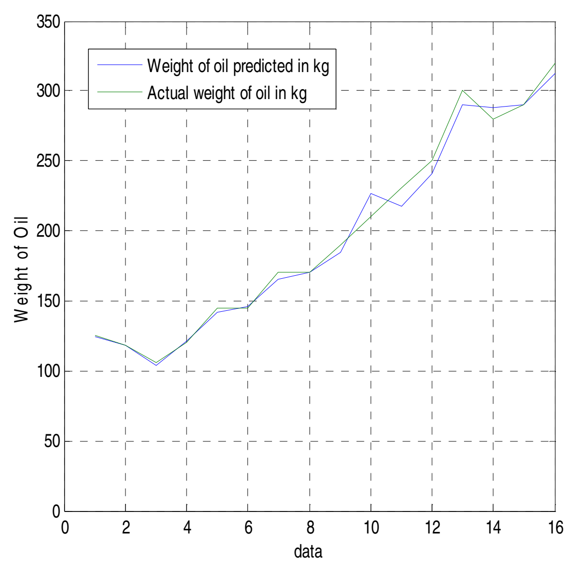

Fig.4 Comparison of actual oil weight and weight predicted using ANN

The data on the abscissa in Figs. 2,3 & 4 correspond to kVA ratings mentioned in Tables III, IV & V.

Once the weight of transformer main materials is predicted, its cost can be easily estimated by multiplying the cost coefficients with the respective weights of raw materials. The final cost can be obtained by adding the manpower cost as well as cost of other components of transformer.

VI. CONCLUSION

The result of MLP network model used for estimating the cost of transformer raw materials shows that MLP network has a good performance and reasonable prediction accuracy was achieved for this model.

The results suggest that ANN model with the developed structure can perform good prediction with least error and finally this neural network could be an important tool for estimating the transformer cost in initial phase of transformer manufacturing project.

ACKNOWLEDGEMENT

The authors would like to thank Mr. Vasant C. Vyas (Proprietor, Jyoti Transelect, Co. Bhuj, India) for support and helpful discussions

REFERENCES

[1] S.S. Karimi Madahi, & P. Salah "A Neural Network Based Method for Cost Estimation 63/20kV and 132/20kV Transformers", Journal of Science and Applied Scientific Research 2(1), pp 939-945 (2012)

[2] P.S. Georgilakis, N.D. Hatziargyriou, N.D. Doulamis & S.D Kollias "Prediction of iron losses of wound core distribution transformers based on artificial neural networks" Elsevier Neurocomputing 23 (1998) pp 15-29

[3] P.S. Georgilakis " Recursive genetic algorithm-finite element method technique for the solution transformer manufacturing cost minimization problem", IET Electr. Power. App., 2009, Vol-3, Iss. 6, pp 514-519

[4] Rabih A.Jabr "Application of Geometric Programming to Transformer Design", IEEE Transactions On Magnetics, Vol. 41 No 11, Nov 2005, pp 4261-4269

[5] Geromel Luiz H & Souza Carlos R "Designing the power transformer via the application of intelligent systems", Proceedings of the 10th Mediterranean Conference on Control and Automation-MED (2002)

[6] E.I. Amoiralis, P.S. Georgilakis, M.A. Tsili & A.G. Kladas "Global Transformer Optimization Method Using Evolutionary Design and Numerical Field Computation", IEEE Transactions On Magnetics, Vol 45.No.3, Mar 2009, pp 1720-1723

[7] C. Hernandez, M.A. Arjona & Shi Hai Dong "Object Oriented Knowledge-Based System for Distribution Transformer Design" IEEE Transactions On Magnetics, Vol. 44 No 10, Oct 2008 pp 2332-2337

[8] P .S. Georgilakis, J .C. Olivares, M. S. Esparza - Gonzalez "An Evolutionary Computation Solution to Transformer Design Optimization Problem", 7th International Conference to Electrical and Electronics Engineering Research (CIIIEE 2010) pp 226-230

[9] L. Ekonomou, P.D. Sakfidas, D.S. Oikonomou " Transformer oil's service life identification using neural networks" , Proceedings of 8th WSEAS International Conference on Electric Power Systems, High Voltages, Electric Machines (2008)pp 222-226

[10] S.R. Paraskar, M.A. Beg & G.M.Dhole " Discrimination between Inrush and Fault in a Transformer: ANN Approach"International Journal of Advancements in Technology, Vol-2, No-2, (2011) pp 306-318

[11] S.Rajasekaran and G.A.Vijayalakshmi Pai " Neural Networks Fuzzy Logic and Genetic Algorithms, Synthesis and Application"

[12] Stephen Chapman " MATLAB Programming for Engineers", Thomson Learning

[13] Rudra Pratap "Getting Started with MATLAB", Oxford University Press