International Journal of Scientific Engineering and Technology (ISSN : 2277-1581)
Volume 2 Issue 5, pp : 322-330 1 May 2013

Prediction of Geomagnetic Variations Using ANN

N. Gururajan[1], V. Kayalvizhi Prabhakaran[2]

[1] Retd. Director, [2] Dept. of Mathematics, Kanchi Mamunivar Centre for Post Graduate Studies, Pondicherry-8.
vgngpondy@yahoo.com ,kayaldew@yahoo.co.in

ABSTRACT

Considering the three components of the geomagnetic field as stochastic quantities, we used neural networks to study their time evolution in years. In order to find the best NN for the time predictions, we tested different kinds of NN and different ways of their training, where the inputs and targets are long hourly time series of geomagnetic field values. The found NN was used to predict the values of the hourly mean values of the geomagnetic field components.

KEY WORDS: Geomagnetic field –Geomagnetic components –Neural networks (NN) –Time series –Time prediction.

INTRODUCTION

The absolute measure of the magnetic field at any location on the Earth is the combination of the sources in the Earth's core, in the Earth's crust and in the near and far space environment of the earth such as ionosphere and magnetosphere. The main magnetic field contributes to almost 99% of the observed magnetic field, changes associated with this field is the secular variation or long-term variation. **[1]**

In addition to the main field variations, as suggested by the hypothesis of Stewart in 1882, daily oscillations are due to the currents in the ionized atmosphere of the earth.

Stewart's ideas on these variations were attributed to the currents flowing in the upper atmosphere due to periodic air motions in the presence of magnetic field. Later this proposition was referred as atmospheric dynamo theory of solar quiet (Sq) and lunar (L) variations.

Schuster confirmed the inference of Stewart and Breit and Tuve, Appleton and Barnett confirmed this proposition experimentally in 1924. Thus the measurement of the magnetic field at any point on the surface of our planet Earth, is considered as a less expensive experimental basic tool to study and infer the geophysical processes taking place inside the Earth and its near and far space environment.**[7].** Chapman and Bartels (1940) and Vestine (1960) represented the source of the regular magnetic variation in terms of a current sheet at an altitude of ~100 km above the earth's surface with two vertices, at around 35^0 latitude. Regular and systematic variations of the magnetic components at many observatories around the globe have subsequently brought out by researchers in 1950s the important finding of the equatorial electrojet, a phenomenon representing the enhanced flow of east-west current over the magnetic equator.

The geographical location of India in the global scenario is ideal for monitoring the developments in the Equatorial Electrojet activities and its associated effect in the Global Sq current system. The existing extensive network of magnetic observatories through Russia from magnetic pole to peninsular India centered along 145^0 geomagnetic meridian provides unique data set of magnetic variations from pole to dip equator. At any point on the surface of the Earth, the magnetic field is a vector, having amplitude and direction. The magnitude of the field is represented by the Total field F. The three components of the magnetic field are Horizontal Intensity 'H', Vertical Intensity 'Z' and the Declination component 'D' of the location defining the angle between the true north and the magnetic north. Observations of these recordings are done on a continuous basis at magnetic observatories with the help of magnetometers specially designed to monitor the sensitive aspects of the changes in the earth's magnetic field.

It is known that the components of geomagnetic activity to some extent reflect the physical processes occuring in the interaction of the perturbed solar wind with Earth's magnetosphere. Therefore, they are connected to each other and with the parameters of near-Earth space. When the physical problem is complex, the technology of artificial neural networks is applied. Such approach for development of the automated forecast of geomagnetic components with the establishment of creative software-computational neural network complex is used. Each neural network experiments are carried out at this complex aims to predict the geomagnetic components.

Artificial Neural Networks (ANN) are very powerful algorithms that may be used to construct empirical computer models of non-linear physical phenomena. These algorithms successfully operate with large volumes of data. ANN are used in solar-terrestrial physics very efficiently.**[2]** Over the past few years artificial intelligence (AI) methods have been increasingly recognized as powerful analysis tools in solar-terrestrial physics (McPherron, 1993; Joselyn et al., 1993). Artificial neural networks (ANNs) are a branch of AI methods which are proving particularly successful in solar-terrestrial time series prediction and pattern recognition; they appear to be especially effective in modeling the time development of irregular processes (Koons and Gorney, 1991; Lundst-edt, 1992; Gorney et al., 1993; Lundstedt and Wintoft, 1994; Williscroft and Poole, 1996; Wu and Lundstedt, 1996; Sutcli€e, 1997; Weigel et al., 1999).

One of the long-standing goals of magnetospheric research is the prediction of the geomagnetic activity. There is an increasing need amongst users of geomagnetic field models for prediction and modeling with greater accuracy and, in particular, to have the ability to correct for short term temporal variations. This paper presents the overview of ANN applications for the development of expert system for the prediction of the geomagnetic activity. Considering the three components of the geomagnetic field D,H and Z as stochastic quantities, we use neural networks to study their time evolution in years. In order to find the best NN for the time predictions, we test different kinds of NN and different ways of their training, where the inputs and targets are long hourly time series of geomagnetic field values at the geomagnetic observatory Pondicherry. The found NN was used to predict the values of the hourly mean values of the geomagnetic field components.

TABLE 1: Source of Data

Station Name	:	Pondicherry
IAGA CODE	:	PND
Geodetic Latitude	:	11.917
Geodetic Longitude	:	79.917
Reported	:	D,H,Z

by WDC for Geomagnetism, Kyoto.

ARTIFICIAL NEURAL NETWORK

Artificial neural networks (ANN or shortly NN) are set of connected neuron layers, which contain one or more neurons.**[8].** Each neuron represents a known function ‘f’ which transfers the input quantity ‘p’, multiplied by a weight ‘w’ and added by a bias ‘b’, to the output ‘a’:

$$A = f(p.w + 1.b)$$

w and b are both adjustable parameters of the neurons.**[3]**

Commonly, neural networks are adjusted or trained, so that a particular input leads to a specific target output. During training the weights and biases of the network are iteratively adjusted to minimize the network performance function. The default performance function is the mean square error (mse) (Demuth and Beale, 2004).

The neural networks have been used to study time series and predictions of different quantities that have a stochastic behaviour (Frank et al., 2001). The geomagnetic field components H,D and Z have such behaviour not only in the time scale of seconds or minutes (Hengre et al., 1999), but even in the larger time scale of years (Duke, 2005). For our intentions, the network inputs are the hourly time series of the geomagnetic field component values H, D and Z and the targets are the known hourly values of the geomagnetic filed components that are shifted in some way from the input values. The predicted values of the geomagnetic field components are simulated by the trained networks when the inputs are shifted from the initial inputs.

The matlab software package, used to create the appropriated neural networks, contains different kinds of networks, but the most successful for the time prediction are those that use the << multi-layer-forward error back propagation algorithm >> (Kugblenu et al. 1999). <<Back propagation>> is a gradient descent algorithm, in which the network weights are moved along the negative of the gradient of the performance function (Demuth and Beale, 2004). There are a number of variations on the basic algorithm that are based on other standard optimization techniques and are used in different kind of training function, as :- train scg (scaled conjugate gradient), - train bfg (Quasi Newton methods), - train br (Bayesian regularization back propagation), - trainlm (Levenberg – marquardt back propagation) etc. of the matlab software (Agarwal et al., 2005).Different neural networks are newff(feed-forward back propagation network), newcf (cascade-forward back propagation network), newelm (Elman back propagation network) (Demuth and Beale, 2004).

TIME SERIES

A time series is a chronological sequence of observations on a particular variable. Usually the observations are taken at regular intervals (days, months, years), but the sampling could be irregular. Common examples of time series are the Dow Jones Industrial Average, Gross Domestic Product, unemployment rate, and airline passenger loads. A time series analysis consists of two steps: (1) building a model that represents a time series, and (2) using the model to predict (forecast) future values. If a time series has a regular pattern, then a value of the series should be a function of previous values. If Y is the target value that we are trying to model and predict, and Y(t) is the value of Y at time t.

The goal of building a time series model is the same as the goal for other types of predictive models which is to create a model such that the error between the predicted value of the target variable and the actual value is as small as possible.

DATA USED

We used the time series of hourly values of the geomagnetic components D,H and Z ranging from January 1998 to December 2006 at the geomagnetic station Pondicherry as our input data.

CHOOSING THE NEURAL NETWORK

The training and simulation procedures of neural network need statistically significant data sets. In this paper we have considered the time series of hourly values of geomagnetic components H,D and Z as stochastic quantities and used neural networks to study their time evolution in years. The time series S(t) are the hourly values of the geomagnetic components H, D and Z at the geomagnetic station Pondicherry ranging from January 1998 to December 2006. During network training, the input is a vector P(1x9*24*365), which values are taken from the series H(t), D(t) and Z(t) where t = 1998,….2006. The target is also vector T(1x9*24*365), which values are taken from the same series H(t), D(t) and Z(t) where t = 1998 + 1,…..,2005+1. During simulation process the input is B(1x9*24*365), taken from the same series H(t), D(t) and Z(t) where t = 1998+1+1, ….2004+1+1. So, the target T is shifted from the input P by d = 1 year, B will be shifted from the target T by the same d. Then the simulation supply the prediction of the value d = 1. 'd' is called the horizon of prediction.

The behaviour of an ANN (Artificial Neural Network) depends on both the weights and the input-output function (transfer function) that is specified for the units. To make a neural network that performs some specific task, we must choose how the units are connected to one another and we must set the weights on the connections appropriately. The connections determine whether it is possible for one unit to influence another. The weights specify the strength of the influence. We can teach a three-layer network to perform a particular task by using the following procedure:

1.We present the network with training examples, which consist of a pattern of activities for the input units together with the desired pattern of activities for the output units.

2. We determine how closely the actual output of the network matches the desired output.

3.We change the weight of each connection so that the network produces a better approximation of the desired output.**[10]**

Using the input series of nine years long, we studied the influence of the kind of network on the relative error of prediction of known values of H, D, Z components. It emerges that the so called newff network has the smallest error of prediction than two known networks newelm and newcf for the 3 geomagnetic components when NN are trained and simulated separately for each component, which is illustrated in the figures 1,2 and 3.

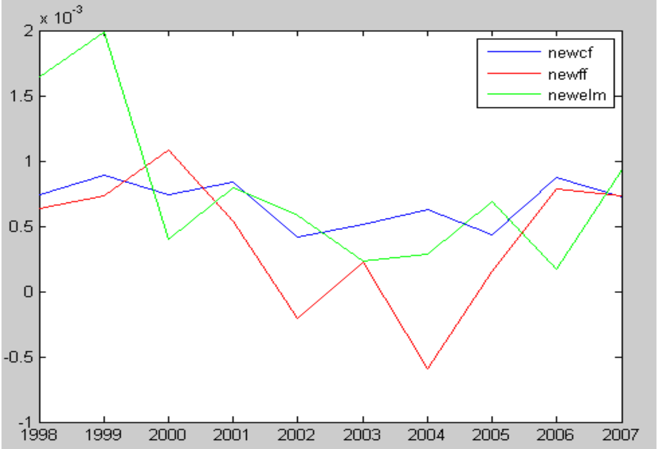

Fig. 1. Relative error of prediction of H-component for three different networks

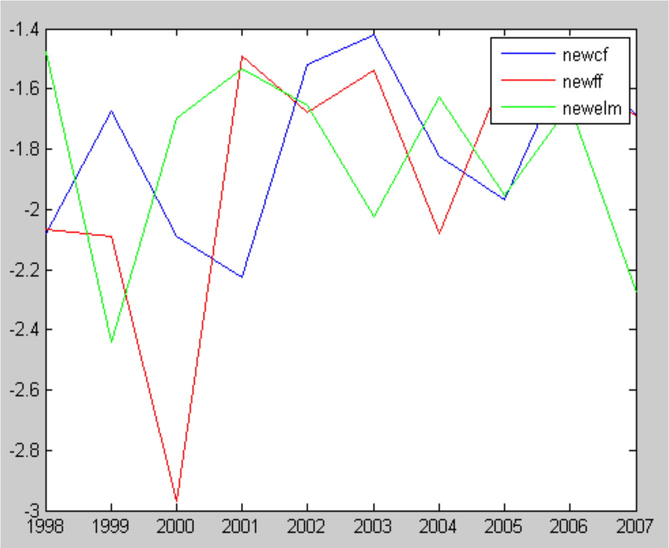

Fig. 2. Relative error of prediction of D-component for three different networks

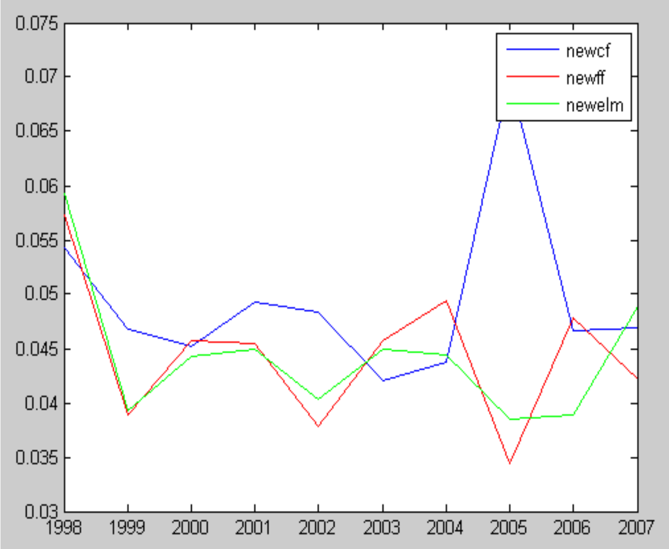

Fig. 3. Relative error of prediction of Z-component for three different networks

Constructing a Feed-Forward Network

When used without qualification, the terms "Neural Network" (NN) and "Artificial Neural Network" (ANN) usually refer to a **Multilayer Perceptron Network (MLP)**.An MLP distinguishes itself by the presence of one or more hidden layers, with computation nodes called hidden neurons, whose function is to intervene between the external input and the network output in a useful manner. By adding hidden layer(s), the network is enabled to extract higher-order statistics. In a rather loose sense, the network acquires a global perspective despite its local connectivity due to the extra set of synaptic connections and the extra dimension of NN interconnections. For more information about MLP, see Haykin (1994). The MLP can have more than one hidden layer; however, theoretical works have shown that a single hidden layer is sufficient for an ANN to approximate any complex nonlinear function (Cybenco, 1989; Hornik *et al.*, 1989). Therefore, in this study, a one-hidden-layer MLP is used.

The ANN used for the development of our geomagnetic daily variation model is a full-connected, three layer, feed-forward, perceptron neural network trained using the back-propagation algorithm (Haykin,1994), which continues to be the most widely used supervised training method. During this iterative process the ANN is trained to map a set of input parameters to a single target value. "Fully connected" means that the output from each input and hidden neuron is distributed to all of the neurons in the following layer.. "Feed forward" means that the values only move from input to hidden to output layers; no values are fed back to earlier layers. All neural networks have an input layer and an output layer, but the number of hidden layers may vary. To solve our problem we will need a feed-forward neural network with one input layer and one output layer .Because that the problem is not linearly separable it will also need a hidden layer.**[14]**

The neural network toolbox makes it easier to use neural networks in matlab. The toolbox consists of a set of functions and structures that handle neural networks, so we do not need to write code for all activation functions, training algorithms, etc. that we want to use! Matlab provides high-level network creation functions, like newcf, newff, newlm and so on .

To create a new feed forward neural network the command newff is used. We have to enter the max and min of the input values, the number of neurons in each layer and optionally the activation functions. We create the perceptron layer with correct input range type.

The NN functioning depends on several properties as the train function, transfer function, NN structure (no of layers, no of neurons in each layer), the training epoch (the no of repetition of the training algorithm) etc., (Demuth Beale 2004).

In order to get better structure of the given kind of NN (newff) for the given series of inputs we studied the dependency of the prediction error from the training function and NN structure in figure , there are shown H, D, Z component variation by the year of prediction (the horizon of the prediction is d=1) for different kind of training functions: trainbr, trainbfg, trainlm. It seems that we have smallest error of prediction using 'trainlm' function when the NN structure are trained and simulated by inputting together the 3 component H,D,Z of geomagnetic field, which is explained in figure 4.

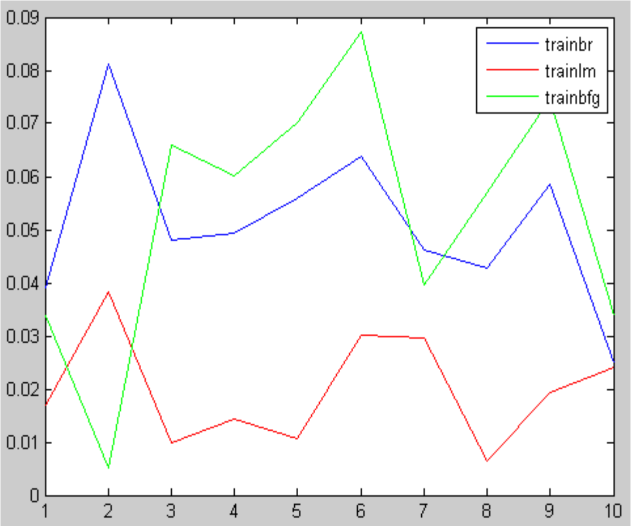

Fig. 4. . Variation of relative error of prediction by the year of prediction,for different training functions by inputting the components H,D and Z together.

A difficult task with the MLP method is choosing the number of hidden nodes. There is no theory yet to tell how many hidden units are needed to approximate any given function. The network geometry is problem dependent. Here, the three-layer MLP with one hidden layer is used and the common trial-and-error method is used to select the number of hidden nodes. **[16]**

In order to illustrate the influence of the NN structure on the quality of prediction, we present in figure 5 , the variation of the relative error of prediction by the year of prediction d=1 for fixed number of neurons together in all the layers of the NN chosen. It seems that for H, D, Z components the best structure of the NN (newff) should be this of three layers : one input layer, one hidden layer with 5 neurons, one output layer with one neuron. Always, in any case in order to get best prediction we tried to find the appropriate structure of the NN after a lot of test with different structures. Figure 5 illustrates the number of neurons to be chosen for this network.

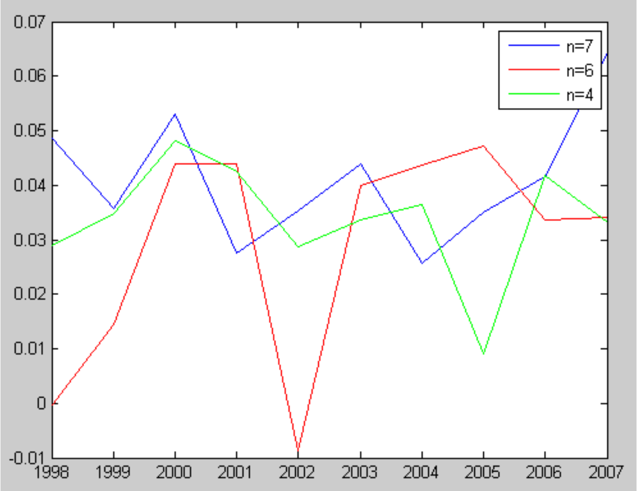

Fig. 5. Variation of relative error of prediction by the prediction year, for different number *n* of neurons per layer.

The variable net will now contain an untrained feed-forward neural network with five neurons in the hidden layer and one neuron in the output layer. The activation function used for all the layers is the tansig function. The 'tansig' tells matlab that we want to use the tansig function as activation function in all layers. There are several issues involved in designing and training a multilayer perceptron network:

- Selecting how many hidden layers to use in the network.
- Deciding how many neurons to use in each hidden layer.
- Finding a globally optimal solution that avoids local minima.
- Converging to an optimal solution in a reasonable period of time.
- Validating the neural network to test for over-fitting.

The goal of the training process is to find the set of weight values that will cause the output from the neural network to match the actual target values as closely as possible. When using 'train', only batch training will be used, regardless of the format. The advantage of train is that it provides a lot more choice in training functions (gradient descent, Levenberg-Marquardt, etc.) which are implemented very efficiently.

The default training algorithm Levenberg-Marquardt backpropagation, 'trainlm' is applied to train our network. Because the error information is propagated backward through the network, this type of training method is called *backward propagation*. The *backpropagation* training algorithm was first described by Rumelhart and McClelland in 1986; it was the first practical method for training neural networks. This MLP is trained using the Levenberg–Marquardt technique as this technique is more powerful than the conventional gradient descent techniques (Hagan & Menhaj, 1994; El-Bakyr, 2003; Cigizoglu & Kisi, 2004).

The Levenberg-Marquardt algorithm

While back-propagation with gradient descent technique is a steepest descent algorithm, the Levenberg-Marquardt algorithm is an approximation to Newton's method.**[15]**
If a function V(x) is to be minimized with respect to the parameter vector x, then Newton's method would be:

$$\Delta x = - [\nabla^2 V(x)]^{-1} \nabla V(x) \text{--------}(1)$$

where $\nabla^2 V(x)$ is the Hessian matrix and $\nabla V(x)$ is the gradient. If V(x) reads:

$$V(x) = \sum_{i=1}^{N} e_i^2(x)$$

then it can be shown that:

$$\nabla V(x) = J^T(x)\, e(x)$$

$$\nabla^2 V(x) = J^T(x)\, J(x) + S(x)$$

where J(x) is the Jacobian matrix and

$$S(x) = \sum_{i=1}^{N} e_i \nabla^2 e_i(x)$$

For the Gauss-Newton method it is assumed that $S(x) \approx 0$, and equation (1) becomes:

□

□ $\Delta x = [J^T(x)\, J(x)]^{-1} J^T(x) e(x)$

The Levenberg-Marquardt modification to the Gauss-Newton method is:

□ $\Delta x = [J^T(x)\, J(x) + \mu I\,]^{-1} J^T(x) e(x)$

The parameter μ is multiplied by some factor (β) whenever a step would result in an increased V(x). When a step reduces V(x), μ is divided by β. When the scalar μ is very large the Levenberg-Marquardt algorithm approximates the steepest descent method. However, when μ is small, it is the same as the Gauss-Newton method. Since the Gauss-Newton method converges faster and more accurately towards an error minimum, the goal is to shift towards the Gauss-Newton method as quickly as possible. The value of μ is decreased after each step unless the change in error is positive; i.e. the error increases. For the neural network-mapping problem, the terms in the Jacobian matrix can be computed by a simple modification to the backpropagation
algorithm (Hagan & Menhaj, 1994).

An important internal factor influencing the NN behavior is the initialization of weights and bias. Usually the weights and bias are initialized randomly in the segment [-1, 1]. We noticed that there are fluctuations of the NN prediction during the retraining and simulation of the same NN and the same inputs. In cases of receiving large fluctuations, we reduced the segment [-1, 1] to a very narrow one.

The target output values were the observed hourly values of a selected component of the geomagnetic field corresponding to the input parameter. The central goal in ANN training is not to memorize the training data, but rather to model the underlying generator of the data (Bishop, 1996). An important factor in achieving this goal is that the amount of training be sufficient, but that the network not be over trained. A recommended method to ensure that an ANN is not over trained is to stop training when the error measured using an independent validation data set starts to increase. Consequently, an independent data set was used to determine when the ANN was sufficiently trained; training ceased when the RMS error between the computed and target output values reached a stable minimum. The number of nodes in the hidden layer was determined by making use of the minimum RMS error between the computed and target output values for the independent data set. Commencing with an insufficient number of hidden nodes, we found that as the number of hidden nodes was increased, this minimum RMS error generally tended to first decrease and then to increase. The number of hidden nodes leading to the minimum value was regarded as the optimum number. In order to assess the ability of the ANN to model the geomagnetic daily variation, the trained ANN was used to predict the hourly values of the geomagnetic components for the chosen data. The results were then compared, either graphically or statistically, with the real time data. **[9]**

NEURAL NETWORK PROGRAM

```
clc;
clear all;
close all;

P=(A(:,1));
T=(B(1:length(A),1));
P=P';
T=T';
x=input('Enter the value for X Component:');
% y=input('Enter the value for Y Component:');
% z=input('Enter the value for Z Component:');
Input=[x];
% Input=[40149;138.7;6557];

net=newff(P,T,5,{'tansig'},'trainlm');
activeInputRange = tansig([40000 40100]);
net = initnw(net,2);
net.trainParam.show =50;
net.trainParam.lr = 0.08;
net.trainParam.mc=0.5;
net.trainParam.epochs = 5000;
net.trainParam.goal = 1e-2;
[net,tr]=train(net,P,T);
a = sim(net,Input)


P=(A(:,2));
T=(B(1:length(A),2));
P=P';
T=T';
y=input('Enter the value for Y Component:');
% y=input('Enter the value for Y Component:');
% z=input('Enter the value for Z Component:');
Input=[y];
% Input=[40149;138.7;6557];
Input=abs(Input);
net=newff(P,T,5,{'tansig'},'trainlm');
activeInputRange = tansig([135 140]);
net = initnw(net,2);
net.trainParam.show =50;
net.trainParam.lr = 0.08;
net.trainParam.mc=0.5;
net.trainParam.epochs = 5000;
net.trainParam.goal = 1e-2;
[net,tr]=train(net,P,T);
b = sim(net,Input)
% b=-1*(b)

P=(A(:,3));
T=(B(1:length(A),3));
P=P';
T=T';
z=input('Enter the value for Z Component:');
% y=input('Enter the value for Y Component:');
% z=input('Enter the value for Z Component:');
Input=[z];
% Input=[40149;138.7;6557];

net=newff(P,T,5,{'tansig'},'trainlm');
activeInputRange = tansig([40000 40100]);
net = initnw(net,2);
net.trainParam.show =50;
net.trainParam.lr = 0.08;
net.trainParam.mc=0.5;
net.trainParam.epochs = 5000;
net.trainParam.goal = 1e-2;
[net,tr]=train(net,P,T);
c = sim(net,Input)
```

RESULT AND ANALYSIS

After training was over, the weights were saved and used to test (validate) the network performance on data for each geomagnetic component. The ANN results were transformed back to the original domain, and we evaluated the error of the predictions of this method, by calculating Rms (Root mean square of deviations) for the whole prediction horizon (d), for each component, according to the formula:

$$\mathrm{Rms} = \sqrt{\frac{\sum_{i}^{d} \left(\left| S_i^{\,p} \right| - \left| S_i^{\,m} \right| \right)^2}{d}}$$

where $S_i^{\,p}$ is the absolute value, predicted by NN that could be D,H or Z component in nT and S_i^m is the absolute value, given by the real time data.

The results are presented in the table II which also lists the respective averaged relative error of prediction. For graphical assessments, the daily variations were generally plotted for the given time period. For statistical assessments, the RMS differences between the hourly values of the respective geomagnetic components and their relative error were computed using data for a specified interval of time. It is seen that the differences between the observed and the actual geomagnetic hourly values are very small.

Table 2: Rms and relative errors of prediction.

Time prediction horizon in years	*d* = 1	d=5	d=8
Rms for *H* in nT	3.7000	4.2000	4.6000
Relative error for *H* in %	0.0030	0.0012	0.0370
Rms for *D* in nT	1.3000	1.4100	3.4000
Relative error for *D* in %	0.0190	0.3100	0.9320
Rms for *Z* in nT	7.8000	16.4000	2.3900
Relative error for Z in %	0.0080	0.0400	0.010

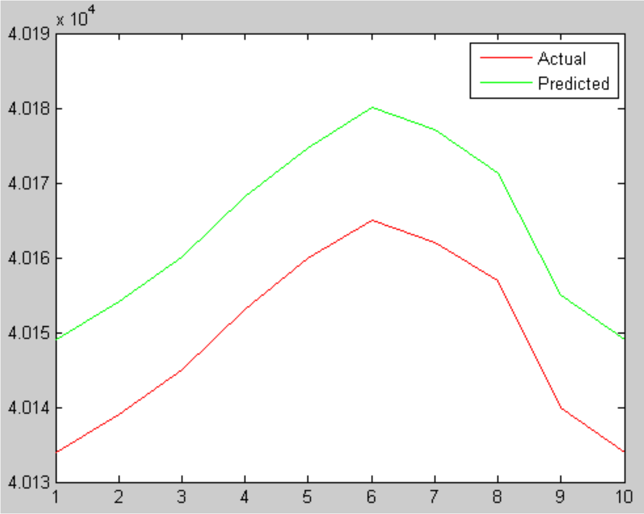

Fig. 6. Predicted values of H component by NN for 1998 – 2007 period and the respective real time data values(actual values).

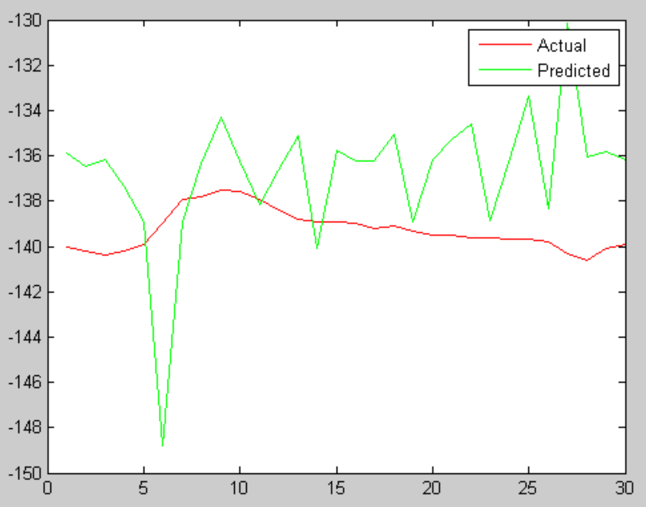

Fig. 7. Predicted values of D component by NN for 1998 – 2007 period and the respective real time data values(actual values).

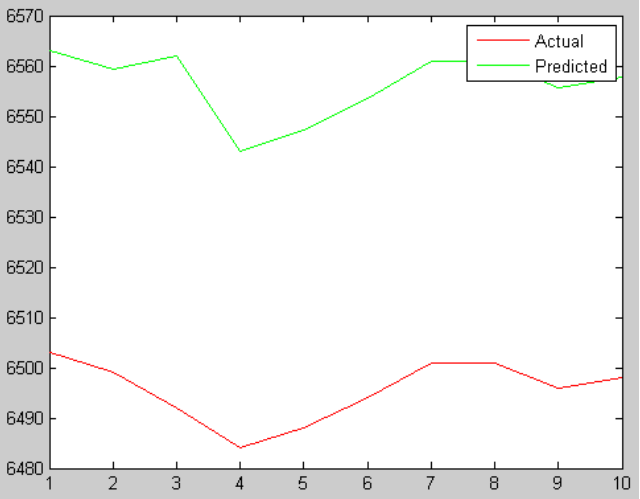

Fig. 8. Predicted values of Z component by NN for 1998 – 2007 period and the respective real time data values(actual values).

CONCLUSION

These comparisons indicate that the ANN method provides acceptable estimates of the prediction of the hourly values of the geomagnetic components D, H and Z at Pondicherry during the given time. Furthermore, the ANN can be used to make estimates of the daily variation for any data in the past or future, provided the required input parameter is specified. The results presented demonstrate that an ANN can be successfully trained to model the geomagnetic daily variation regionally. The ANN method has computational advantages. The process of computation is simplified, particularly if one wishes to take account of more variables. It is thus more amenable to computation of a realistic model. However, the ability to adequately represent the day-to-day variability still remains a problem to be resolved. Applying the ANN method to other regions or even globally should not be a problem, provided the data for training the neural network are available.

The application of ANN concepts for time series prediction have been used for numerous applications in astronomy, meteorology, geology, cartography and geodesy, satellite data analysis, etc. This technique is used here to study the time series prediction of geomagnetic variations at the Indian Geomagnetic observatory Pondicherry. Generally for a huge volume of data in a complicated analysis this technique yields proper results. As a result of this study, it is expected that future usage of this technique may be appropriate for exploring some new results in geomagnetism.

REFERENCES

I. Barton, C. E., International geomagnetic reference field: the seventh generation, J. Geomagn. Geoelectr., 49, 123-148, 1997.

II. Bishop, C. M., Neural networks for pattern recognition, Oxford University Press, New York, 1996.

III. Gorney, D. J., H. C. Koons, and R. L. Waltersheid, Some prospects for artificial intelligence techniques in solar-terrestrial predictions, Solar-Terrestrial Predictions: Workshop IV, Eds. Hruska et al., Boulder USA, 2, 555-564, Sept. 1993.

IV. Greener, J. G. and D. M. Schlapp, A study of day-to-day variability of Sq over Europe, J. Atmos.

V. Terr. Phys., 41, 217-223, 1979.Haykin, S., Neural networks. A comprehensive foundation, Mac- millan College Publishing Company, New York, 1994.

VI. Hibberd, F. H., Day-to-day variability of the Sq geomagnetic field variation, Aust. J. Phys., 34, 81-90, 1981.

VII. Joselyn, J., H. Lundstedt, and J. Trolinger, Eds., Proce. International Workshop on Arti®cial Intelligence Applications in Solar-Terrestrial Physics, Boulder USA, Dec 1993.

VIII. KotzeÂ , P. B., Regional magnetic field modelling, Proce. Workshop on the Hermanus Magnetic Observatory's Geophysical Services, 49-58, 1992.

IX. Langel, R. A., T. J. Sabaka, R. T. Baldwin, and J. A. Conrad, The near-Earth magnetic field from magnetospheric and quiet-day ionospheric sources and how it is modeled, Phys. Earth Planet.Int., 98, 235-267, 1996.

X. Lundstedt, H., Neural networks and predictions of solar-terrestrial effects, Planet. Space Sci., 40, 457-464, 1992.

XI. Lundstedt, H., and P. Wintoft, Prediction of geomagnetic storms from solar wind data with the use of a neural network, Ann. Geophysicae, 12, 19-24, 1994.

XII. McPherron, R. L., Applications of artificial intelligence to space weather forecasting, Solar- Terrestrial Predictions: Workshop IV, Eds. Hruska et al., Boulder USA, 2, 611-626, Sept 1993.

XIII. Sutcliffe, P. R., The geomagnetic Sq variation at Marion Island, S. Afr. J. Sci., 73, 173-178, 1977.

XIV. Sutcliffe, P. R., Substorm onset identification using neural Networks.

XV. Weigel, R. S., W. Horton, T. Tajima, and T. Detman, Forecasting auroral electrojet activity from solar wind input with neural networks, Geophys. Res. Lett., 26, 1353-1356, 1999.

XVI. Williscroft, L. A., and A. W. V. Poole, Neural networks, foF2, sunspot number and magnetic activity, Geophys. Res. Lett., 23, 3659-3662, 1996.

XVII. Marquardt, D. (1963) An algorithm for least squares estimation of non-linear parameters. J. Soc. Ind. Appl. Math. 431–441.

XVIII. Tsoukalas, L. H. & Uhrig, R. E. (1997) Fuzzy and Neural Approaches in Engineering. John Wiley Inc., New York, USA. Wasserman, P. D. (1993) Advanced Methods in Neural Computing, 155–161. Van Nostrand Reinhold, New York, USA.