

Enhanced Elman Neural Network and Its Applications in Agriculture

Shyamsundar Narayan Patil^{#1}, Priti Subramaniam^{*2}

[#] *Research Scholar, Department of Computer Science and Engineering, S.S.G.B.C.O.E.T. Bhusawal Maharashtra India*

^{*} *Asst. Professor, Department of Computer Science and Engineering, S.S.G.B.C.O.E.T. Bhusawal Maharashtra India*

Abstract— Elman Neural Network has been an efficient system identification tool in many areas. However, one of the problems often associated with this type of network is the speed of learning which is too slow. The HF Elman neural network is presented for the modelling of unknown delay and high-order nonlinear system. Then chaos searching is imported to train it, make BP algorithm may skip the local minimum and find the global minimum easily. The simulation result shows that the proposed method may speed up the original ENN algorithm and get good results for the prediction tasks.

Keywords— Elman Neural Network, HF Neural Network, BP Algorithm, Simulation, Prediction.

INTRODUCTION

Artificial neural network contend a novel role within the field of nonlinear systems modelling, identification and management as a result of its nonlinear mapping and data processing capability, recurrent neural network have the flexibility of coping with dynamic info due to its inherent feed-back structure and represent the event manner during this space [1, 2]. So far, individuals are developed tens of continual neural network, in which, Elman network is that the most generally used kind.

Elman network is proposed by J. L. Elman in 1990 for voice processing [3], its full name is “simple recurrent neural network”, Since only partial feedback connection adopted, the accuracy has declined once it's used for prime steps system identification, Therefore, people advanced a range of modified structure [4], to enhance its dynamic mapping capabilities. Elman networks are usually adopted to spot or generate the temporal outputs of nonlinear systems. It's documented that a recurrent network is capable of approximating a finite state machine [5] and therefore will simulate any statistic. Therefore recurrent networks area unit currently wide employed in fields involved with temporal issues. In printed literature, however, all the initial weights of recurrent networks are set arbitrarily rather than exploitation any previous information and therefore the trained networks are imprecise to human and their convergence speed is slow. Additionally, the temporal generalization capability of easy recurrent networks isn't therefore smart [6]. These two major issues build the applications of recurrent networks with temporal identification and management of systems tougher.

BP algorithm is used to train Elman network usually, that the network convergence slowly and straightforward to fall local minimum inevitably. People progress several helpful

enhancements supported the standard BP algorithm, and suggest numerous training algorithm supported nonlinear optimization that improved the network's performance effectively [7].

HF ELMAN NETWORK

Fig. 1 shows the structure of a simple ENN. In Fig. 1, when the hidden units are calculated, their values are used to calculate the output of the network and all are stored as “extra inputs” (called context unit) to be used when the next time the network is operated. Thus, the recurrent contexts offer a weighted sum of the previous values of the hidden units as input to the hidden units. As shown in the Fig. 1, the activations are copied from hidden layer to context layer on a one for one basis, with fixed weight of 1.0 ($w=1.0$). The forward connection weight is trained between hidden units and context units as well as alternative weights. If self-connections are introduced to the context unit when the values of the self-connections weights (a) are fixed between 0.0 and 1.0 (usually 0.5) before the training process before the training process.

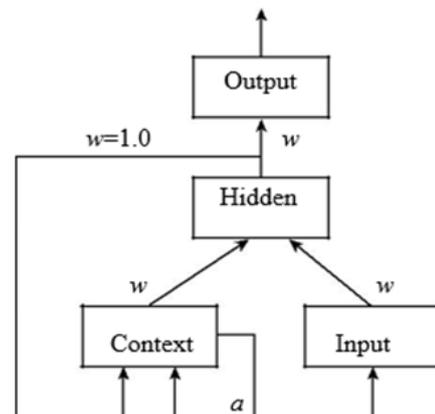


Fig. 1 The structure of the ENN

A. Internal Process Analysis of ENN

Fig. 2 is the internal learning process of ENN by the error back-propagation algorithm. From Fig. 2 we can see that training such a network isn't easy since the output of the network depends on the inputs and conjointly all previous inputs to the network. So, it ought to trace the previous values according to the recurrent connections (Fig. 3). So, the calculation of the functional derivatives isn't easy and it leads to low efficiency to handle various signal problems.

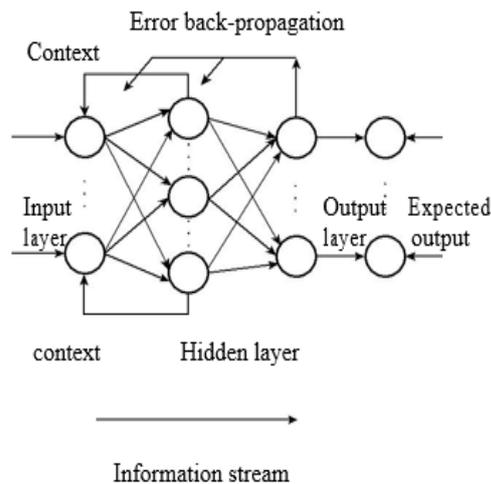


Fig. 2 Internal process analysis of ENN

HF Elman network has two context layers, context 1 is analogous to the context in Fig. 1, it's used to achieve the feedback of the hidden state; context 2 may be a special party that used to achieve the feedback data of the output, and it forms two feedback loops, one of the links from the output nodes delayed when one-step time back to the hidden nodes and other from the output nodes by one-step time delays then back to its input. Additionally, the two context layers structural the self-feedback loop, and set up the self-feedback gain factor, severally. So the network can't only dynamic back a moment before the present state, but also to reflect additional information of earlier moment. From here we tend to see that the HF Elman network combine state feedback, output feedback and time delay organically.

B. Chaotic Training Algorithm

BP rule is basically for the best resolution within the direction of gradient descent, the road of gradient descent are mounted once the initial weights selected at random, if get in local minimum, we've to selected the initial weights at random again, and then reuse gradient method, so the weights which selected later could also be in the area already searched before, and therefore the algorithm can do the work meaningless. Thanks to the argotic property of chaos, chaos variables are often used to optimize the weights and escape the local optimum.

Chaotic sequence is used to optimize all weights of the HF Elman network. The Basic steps are as follows:

- (1) Initialization weight vectors. Generating random number in the interval (-A, A) and assign it to the initial weight W , where A is the extreme of initial estimated weight, setting the initial weight E_{best} .
- (2) Let $W' = W/2A + 1/2$, map the initial weights to the definition of chaos variables, use chaotic Algorithms to optimize the weight. Generating chaotic weight W'' from $W'' = 4W'(1 - W')$; Then map W'' to the original interval from $W''' = 2AW'' - A$.
- (3) Computing the error of network from W''' and marked it to E , if $E < E_{best}$, then let $E_{best} = E$, and mark the weight W''' to W_{best} .

(4) Refresh the initial weigh for the next chaotic mapping. Let $W = W_{best} + \epsilon u$, where ϵ is a small weight, u is a random variable in the interval (1, 1).

(5) On a given network training accuracy (or cycles), stop learning if the value of E_{best} satisfies the training precision, otherwise turn to step (2).

EXPERIMENTAL SETUP

A. Training data Sets

A training set could be a set of data used to discover potentially predictive relationships. This is the data we use to estimate the parameters of the model. We additionally obtain in-sample match diagnostics. We usually use training set to estimate and pre-select promising model or filtrate bad models. For instance, check out t-statistics, significance etc. to remove hopeless models from consideration.

A important feature of neural networks is an iterative learning process during which data cases (rows) are presented to the network one at a time, and therefore the weights associated with the input values are adjusted every time. Finally cases are presented, the process usually starts over again. Throughout this learning phase, the network learns by adjusting the weights so as to be ready to predict the correct class label of input samples. Neural network learning is additionally referred as "connectionist learning," because of connections between the units. Benefits of neural networks include their high tolerance to noisy data and also their ability to classify patterns on which they have not been trained. The foremost common neural network algorithm is back-propagation algorithm proposed in the 1980's.

There are general rules selected over time and followed by most researchers applying this architecture to their problems.

Rule One: As the complexity in the relationship between the input data and the desired output increases, the number of the processing elements in the hidden layer should also increase.

Rule Two: If the process being modeled is separable into multiple stages, then additional hidden layer(s) may be required. If the process is not separable into stages, then additional layers may simply enable memorization of the training set, and not a true general solution effective with other data.

Rule Three: The amount of training data available sets an upper bound for the number of processing elements in the hidden layer(s). To calculate this upper bound, use the number of cases in the training data set and divide that number by the sum of the number of nodes in the input and output layers in the network. Then divide that result again by a scaling factor between five and ten. Larger scaling factors are used for relatively less noisy data. If you use too many artificial neurons the training set will be memorized. If that happens, generalization of the data will not occur, making the network useless on new data sets.

B. Testing data Sets

A set will be test set of data when it is used to assess the strength and utility of a predictive relationship. This is the set which we didn't use for estimation. We predict the

outcomes of this data set, and obtain out-of-sample diagnostics like RMSFE (root mean square forecast error). We use testing set to select the final model, the best one in all. Of course, we take under consideration the in-sample diagnostics, but out-of-sample metrics would somewhat more prominent.

To summarize, the learning set is used for creating a model, validation set is used for verifying the model, and the testing set is used for testing of the usability of the model. Here we have taken 300 to 500 records at a time to test the data sets.

RESULT ANALYSIS

A. Visualisation of Parameters

Visualisation of parameters actually clears the ideas about how data is distributed and can be view systematically.

1) *Rainfall*: In our Simulation we have taken three categories low, moderate and high. On five hundred testing records we get the following results (see fig.3)

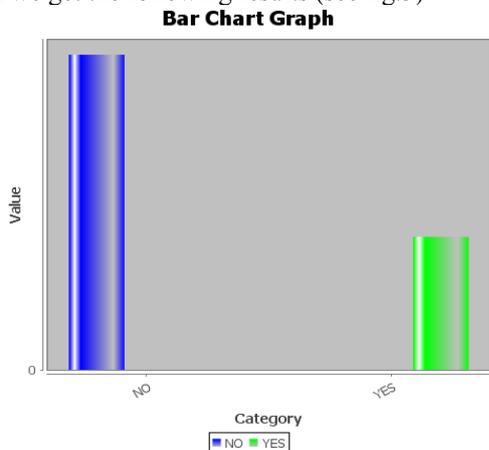


Fig. 3 Visualisation of Rainfall as parameters

2) *Temperature*: In our Simulation we have taken temperature range from 270c to 480c. (See fig. 4 below)

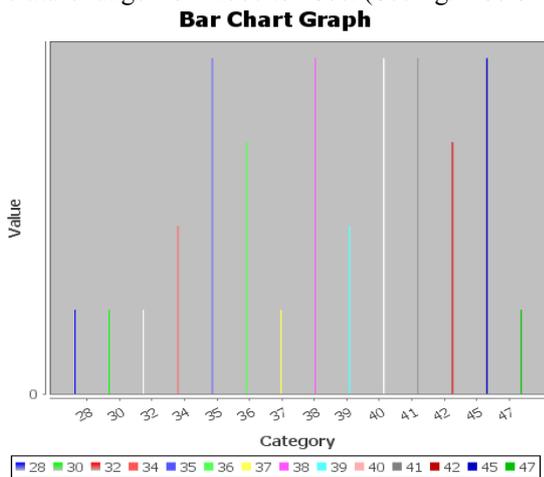


Fig. 4 Visualisation of Rainfall as parameters

B. Classification

Elman neural network is one of feedback neural network; based on BP neural network hidden layer adds an undertake layer, as delay operator, purpose of memory, so that the

network system has ability to adapt to the time-varying characteristics and has strong global stability.

The topology is mainly divided into four layers: input layer, hidden layer, undertake layer, and output layer. Undertake layer is used to remember the output of hidden layer, which can be seen as a step delay operator. Based on BP network, the output of hidden associates with its input through the delay and storage of undertake layer. This way of association is sensitive to previous data, and internal feedback network can increase the ability of handing dynamic information. Remembering the internal state makes it have dynamic mapping function, which makes the system have the ability to adapt to time-varying characteristics.

Here classification for the given parameters is represented by the following (see fig. 5 below)

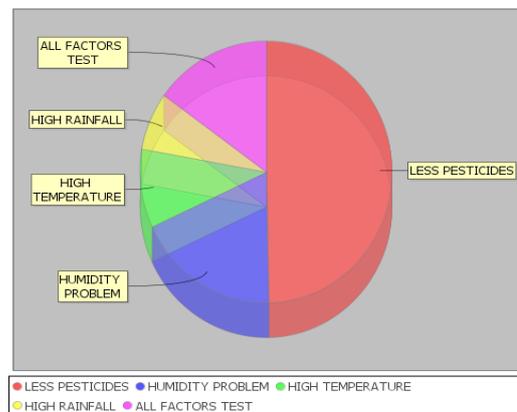


Fig. 5 Classification of crops based on Different conditions

If we observe that classification carefully we will get it in tabular form (see table 1 below)

TABLE I Classification of Crops Based on Agriculture Conditions

Classification of crops (500 records are taken)	No. of Crops and Their percentage
Less Pesticides	(250, 50%)
Humidity Problem	(90, 18%)
High Temperature	(50, 10%)
High Rainfall	(35, 7%)
All Factors Test	(75, 15%)

And pictorial representation of above data is shown in figure 6 below.

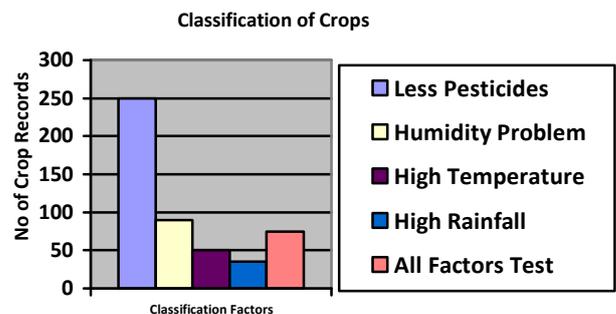


Fig. 6 Classification of crops based on Agriculture conditions

C. Accuracy

Elman neural network is an optimizing BP neural network, so it inherits the characteristics of BP, but BP has some defects; for instance, it is easy to fall into local minimum, the fixed learning rate, the uncertain number of hidden layer neurons, and if facing small sample this is more difficult. In order to enhance operating efficiency and recognition accuracy of neural networks, we introduce PLS and CA into Elman neural networks algorithm. Using PLS longitudinal dimension reduction, it can fully take into consideration the level of correlation between feature variables and the dependent variables and can solve the small sample problems caused by clustering of that variables. The accuracy can be calculate by following formula.

$$Accuracy\ of\ network = \frac{Total\ no\ of\ records\ retrived}{Total\ no\ of\ records} * 100$$

$$Accuracy\ of\ network = \frac{409}{500} * 100 = 97.80$$

The comparison table is shown table 2 below.

TABLE II Simulation Results Comparison of Enhanced Elman Neural Network with Other Networks (Based on 500 records)

The forecasting accuracy of Enhanced Elman	The forecasting accuracy of IOIF-Elman	The forecasting accuracy of Elman	The forecasting accuracy of BP
0.9781	0.91342	0.89575	0.62885

And the accuracy is observed by the visualized result of the same records (see fig. 7 below)

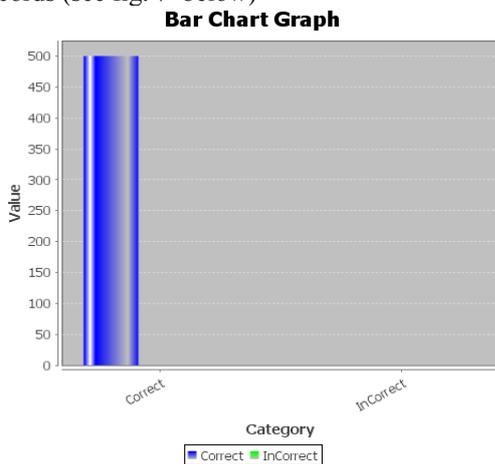


Fig. 7 Accuracy by Bar Chart Graph (500 records are taken)

CONCLUSIONS

In this paper, we implemented a new Enhanced Elman network which combine the state feed-back, output feedback and time delay organically. And introducing chaos mechanism to train the improved network, thus the shortcomings of local extreme caused by the traditional gradient algorithm is eliminated effectively, and either the network’s learning efficiency or the fore-cast accuracy has been enhanced greatly which nearly closed to perfect. Results show that, the Enhanced Elman network has super performance to the modelling of high-order, delay, nonlinear dynamic system.

ACKNOWLEDGMENT

We would like to express the deepest appreciation to all those who are involved directly or indirectly in this work. First and foremost, I am sincerely thankful to Principal Dr. R. P. Singh and Prof. D. D. Patil, Head of Department, Computer Science and Engineering, S.S.G.B.C.O.E.T., Bhusawal (Maharashtra). We would like to thanks all our friends for their support without which this work would not see the daylight.

REFERENCES

- [1] Mohamed Ibnkahla. Nonlinear system identification using neural networks trained with natural gradient descent, EURASIP Journal on Applied Signal Processing, 2, 2003, 1229-1237.
- [2] Zhi-hang Tang, Rong-jun Li. A novel data mining classification method based on neural network. Journal of Computational Information Systems, 6(4), 2010, 1069-1076.
- [3] J. L. Elman, Finding structure in time. Cognitive Science, 14(2), 1990, 179-211.
- [4] X. H. Shi, Y. C. Ling. Improved Elman networks and applications for controlling ultrasonic motors, Applied Artificial Intelligence, 7(18), 2004, 603-629.
- [5] S. C. Kremer, on the computational power of Elman-style recurrent networks, IEEE Trans. Neural Networks, vol. 6, no. 4, 1995, 1000-1004.
- [6] D. L. Wang, X. M. Liu, S. C. Ahalt, on temporal generalization of simple recurrent networks, Neural Networks, 9, 1996, 1099-1118.
- [7] Shu-xia Yang, Xiang Li, Ning Li, Shang-dong Yang, Optimizing neural network forecast by immune algorithm, Journal of Central South University of Technology, 13(5), 2006, 573-576.
- [8] D. T. Pham, X. Liu, Training of Elman networks and dynamic system modelling, International of System Science, 27(2), 1996, 221-226.
- [9] X. Z. Gao, X. M. Gao, S. J. Ovask, A modified Elman neural network model with application to dynamical system identification, IEEE International Conference on Systems, Man, and Cybernetics. Beijing, 1996, 1376-1371.
- [10] WANG Ling-ling, XUAN Da-min, XU Yong-sheng, SU Gang. Forecasting of Daily Gas Load Based on OIF-Elman Neural Network[J]. GAS & HEAT, 2008, 28(7): B07-B10.
- [11] Wang Hao-jun, Jiang Jian-qun, Li Fu-qiang. Application of Elman neural network and MATLAB to dam displacement prediction [J]. WATER POWER. 2005, 31(1): 31-33.