International Conference on Systems, Science, Control, Communication, Engineering and Technology 2015 [ICSSCCET 2015]

ISBN	978-81-929866-1-6	**VOL**	01
Website	icssccet.org	**eMail**	icssccet@asdf.res.in
Received	10 - July - 2015	**Accepted**	31- July - 2015
Article ID	ICSSCCET050	**eAID**	ICSSCCET.2015.050

Optimization of Training phase of Elman Neural Networks by suitable adjustments on the Network parameters

N.Mohana Sundaram[1] , P.N Ramesh[2]
[1, 2] Faculty of Computer Science and Engineering
Karpagam Institute of Technology, Coimbatore

ABSTRACT : *The Elman Neural Network (ENN) is a type of recurrent network that has a context layer as an inside self-referenced layer .The ENN is trained in a supervised manner using the popular back propagation algorithm, based on the inputs and targets given to the network. Various parameters of the network like initialization of weights, types of inputs, number of hidden neurons, learning rate and momentum factor influence the training behaviour of the network. There exists no solid formula to guarantee that the network will converge to an optimum solution or to a faster convergence or convergence even occurs at all. If the parameters of the network are wrongly selected, then it may take a long time for the network to train or some times the network may not get converged at all. In this work the performance of the network is analyzed using extensive tests by adjusting the values of the above referred parameters to find the optimum condition of learning. A digital system, Binary to ASCII Converter is considered for carrying out the experiments. The optimum conditions are presented in this paper.*

Keyword: Elman neural network, weights, hidden neurons, network parameters, learning rate, momentum factor.

I. INTRODUCTION

A recurrent neural network called Elman Neural Network (ENN) was proposed by JEFFREY L.ELMAN which has two-layers back with an additional feedback connection from the output of the hidden layer to its input layer called context layer [1]. The additional units are called "Context Units". It is a recurrent network with context layer containing context units with a fixed weight of 1 such that the contents of hidden layer are copied to context layer on a one-to-one basis [1, 2,12] as shown in FIG 1.

Cite this article as: N.Mohana Sundaram, P.N Ramesh. "Optimization of Training phase of Elman Neural Networks by suitable adjustments on the Network parameters." *International Conference on Systems, Science, Control, Communication, Engineering and Technology (2015)*: 229-235 Print.

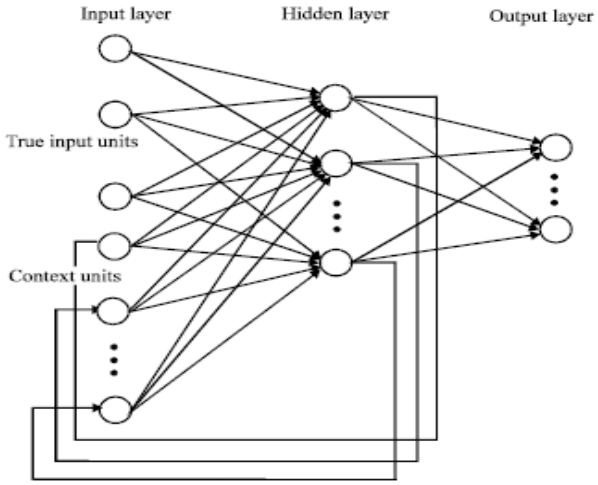

Fig 1. ELMAN network showing the additional Context layer

The context units save previous output values of hidden layer neurons and are fed back fully connected to hidden layer neurons and thus they serve as additional inputs to the network. During the operation of the net both the current input from the input layer and previous state of the hidden layer saved in the context layer are passed to the hidden layer. The hidden layer processes them and pass to the output layer.

II.TRAINING THE ENN

The famous Back Propagation Algorithm or the Error Back Propagation Algorithm is used to train the Elman Neural Network which updates the hidden–output, the input–hidden and the context–hidden weights in order to reduce the difference between the output of the output layer and its desired output. Since the network training algorithm is supervised, the desired outputs are necessary and serve as a reference to calculate errors. Basically, error back propagation algorithm consists of two passes through the different layers of the network: a forward pass and a back-ward pass. In the forward pass, input vector is applied to the input node of the network, and its effect propagates through the network layer by layer. Finally, a set of outputs is produced as the actual response of the network. The synaptic weights of the networks are all fixed during the forward pass. The backward pass starts at the output layer by passing error signals leftward through the network and recursively computing the local gradient for each neuron. This permits the synaptic weights of the net-work to be all adjusted in accordance with an error-correction rule.[2,8,12] The algorithm is stopped when the error has become small and within in the set tolerance error value. Finding the best set of weights and biases for the neural network is the objective of the training. Training with back-propagation is an iterative process. At each iteration, back-propagation algorithm computes a new set of neural network weight and bias values that in theory generate output values that are closer to the target values. So Back-propagation algorithm calculates the gradient of the error and then propagates error backward through the network to modify the weights and biases.[2,4,8,12].

II.MODELING AND SIMULATION

The Elman Network is modelled and simulated using MATLAB.The Problem considered for analysis is a digital system, "Binary to ASCII Encoder". The Seven Bit ASCII equivalent Binary code is the Input to the Network. The Output is the Normalized Decimal Equivalent of the ASCII.For Example the ASCII value of 'A' is 65. The equivalent Binary code is 1000001. So for 'A' the input is: 1000001 and the output is: 0.65 (65 normalized to 0.65) . In case of bipolar inputs the input is 1 -1 -1 -1 -1 -1 1.For 'B' the input is: 1000010 (in case of bipolar inputs it is 1 -1 -1 -1 -1 1 -1 and the output is 0.66 and so on.A set of 60 data a used for training and for testing/validation 30 data are used.The inputs are Binary values and the outputs are Analog values (Decimal values) which is more effective for the analysis.So the ENN has Seven (7) input neurons and One (1) output Neuron.The number of Hidden layer Neurons varied with different values to conduct the analysis. The number of neurons in Context layer is same as the number of neurons in hidden layer.Transfer function of the hidden neurons and output neurons are the Tan sigmoid function or the hyperbolic tangent function which is shown in Fig2.

Cite this article as: N.Mohana Sundaram, P.N Ramesh. "Optimization of Training phase of Elman Neural Networks by suitable adjustments on the Network parameters." *International Conference on Systems, Science, Control, Communication, Engineering and Technology (2015)*: 229-235 Print.

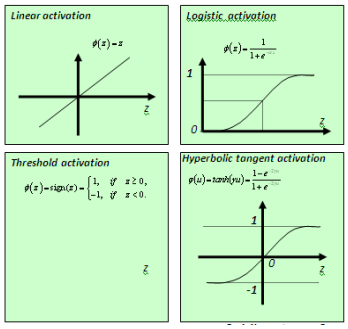

Fig 2. Various Activation Functions

The Sigmoid (logistic) function is chosen because the biological neurons are activated by similar function and it offers a fast response. Further as the outputs are positive analog values, the LOG Sigmoid is chosen rather than Tan Sigmoid. Error function is SSE (Sum Squared Error) which measures the performance of the network .The following Table shows the Architecture of the ENN and the parameters

Table 1
Table showing the ENN Architecture and parameters

Number of Input Neurons	7
Number of output Neurons	1
Number of Hidden Layers	1
Number of Hidden Neurons	Varied to find optimum
Activation Function for both layers	Tan Sigmoid
Performance function (error)	SSE(Sum Squared Error)
Training Algorithm	Back Propagation
Learning Rate	Varied to find optimum
Momentum factor	Varied to find optimum
Weight Initialization	Different methods

Also the training function for the network updates weight and bias values with Levenberg–Marquardt optimization. The self-adaptive learning function is the gradient descent with momentum weight and bias learning function.

III.DATA REPRESENTATION

For Binary inputs the data may be represented in Binary form (0, 1) or Bipolar Form (-1, 1).The output of a neuron depends on the factor 'input x weight'. If input is '0' then the output may be of a small value and the Gradient may be constant which results in long period of Convergence or No convergence at all.This suggests that learning may be improved if the input is represented in bipolar form and the bipolar sigmoid is used for the activation function In this work the Convergence take after more than 4700 epochs for binary inputs(fig 3a) and 2556 Epochs for Bipolar Inputs(Fig 3b) , the other conditions being same.

Fig 3a. The learning curve for Binary inputs

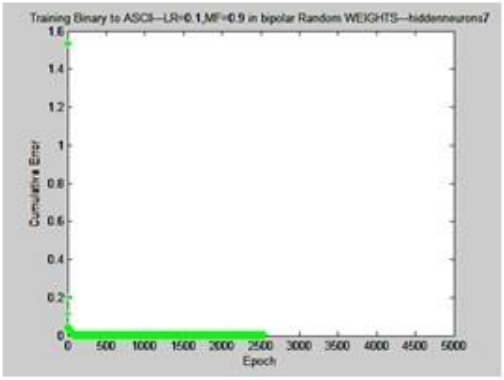

Fig 3b. The learning curve for bipolar inputs

Cite this article as: N.Mohana Sundaram, P.N Ramesh. "Optimization of Training phase of Elman Neural Networks by suitable adjustments on the Network parameters." *International Conference on Systems, Science, Control, Communication, Engineering and Technology (2015)*: 229-235 Print.

IV. WEIGHTS INITIALIZATION

Weight initialization is one of the most effective approaches in speeding up the training of all types of neural networks[2, 3, 5, 7, and 12]. In most of the cases all the initial weights of recurrent networks are set randomly instead of using any prior knowledge and thus the trained networks are vague to human and their convergence speed is slow.[2,3,5,7,12] The initial weights influence how rapidly the net converges during training phase.The initial assignment of value to weights brings a major impact towards the Learning Behavior of the Network.If the algorithm computes successfully the correct value of the weight (rather ∂w) it can converge to a faster solution.Otherwise the convergence may be slow or will not converge at all.Faster learning can be obtained by using

NGUYEN-WIDROW initialization. [7]

$$\beta=0.7(p)^{1/n}$$

where n=no of input units p=no of hidden units and β=scale factor.

The analysis is conducted with different weight initializations.In the beginning the random weights are set only with + weights and the convergence is not obtained even after 5000 epochs.The initial weights are set at random with both +ve and –ve initial weights and with 9 hidden neurons the the network converges at 2262 epochs.(fig 4a)

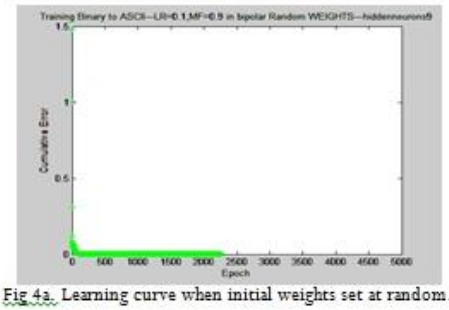

Fig 4a. Learning curve when initial weights set at random.

By setting initial weights using Nyguen-Widrow method (between -0.01 to +0.7), the network converges at 1351 epochs , other settings being the same.(fig 4b.)

Steps in NW initialization

Initialize weights with random numbers between -0.5 to +0.5

Compute the previous weights Vij(old).

Reinitialize weights as

$$Vij = \beta\ Vij(old).\ /\ I\ Vij(old)\ |$$

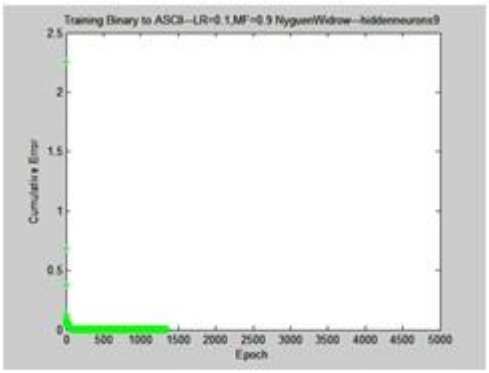

Fig 4b. Learning curve when initial weights set using Nyguen-widrow initialization.

Cite this article as: N.Mohana Sundaram, P.N Ramesh. "Optimization of Training phase of Elman Neural Networks by suitable adjustments on the Network parameters." *International Conference on Systems, Science, Control, Communication, Engineering and Technology (2015)*: 229-235 Print.

V.LEARNING RATE (Γ) AND MOMEMTUM FACTOR(A)

When using backpropagation in a network with n different weights w1, w2, …, wn, the i-th correction for weight wk is given by $\Delta wk(i) = -\gamma\, \partial E\, \partial wk + \alpha \Delta wk(i-1)$, where γ and α are the learning and momentum rate respectively.Normally, we are interested in accelerating convergence to a minimum of the error function which can be done by increasing the learning rate up to an optimal value. Introduction of the momentum rate allows the attenuation of oscillations in the iteration process.The adjustment of both learning parameters to obtain the best possible convergence is normally done by trial and error or by some kind of random search. Since the optimal parameters are highly dependent on the learning task, no general strategy has been developed to deal with this problem.When the learning rate is less, the learning is slow but the accuracy is high.But if it is higher more oscillations are produced and accuracy is very low even though the convergence is faster.In this work the analysis is conducted for learning rates of 0.1,0.2 and 0.3.Even for learning factor 0.2 there is no convergence and the error is high. The learning curves are shown in figures 5a,5b and 5c. When learning rate is high it can be observed more oscillations and more errors even though the learning is faster.

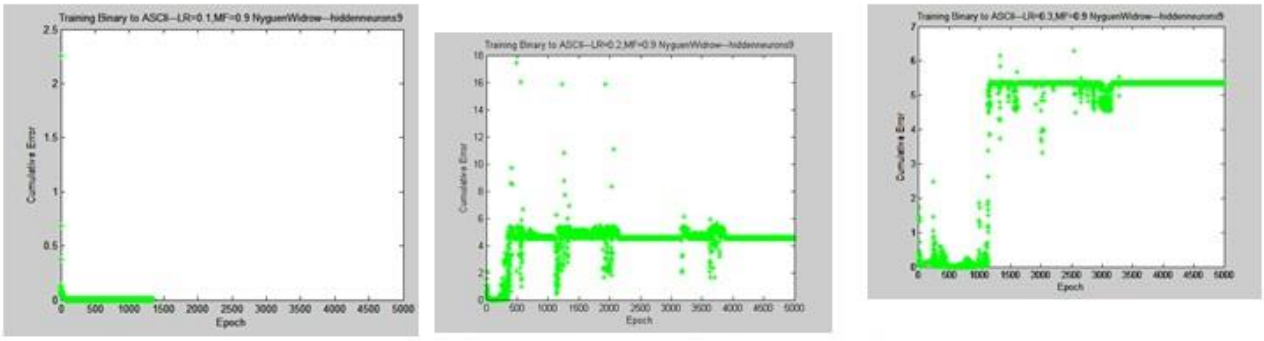

Fig 5a, 5b and 5c. Learning curves for learning rates 0.1, 0.2 and 0.3.

For higher learning rates more oscillations can be observed and more errors occur even though the convergence is faster.The Momentum factor is introduced to make the learning(convergence) faster by arresting the oscillations. In this work it is set at 0.9 and at 0.6 and lesser than 0.6 results in non-convergence. It can be observed from figures 6a and 6b that for mf=0.9 the learning curve converges at 1351 epochs while for mf=0.7 it converges at 3100 epochs , other conditions being the same.

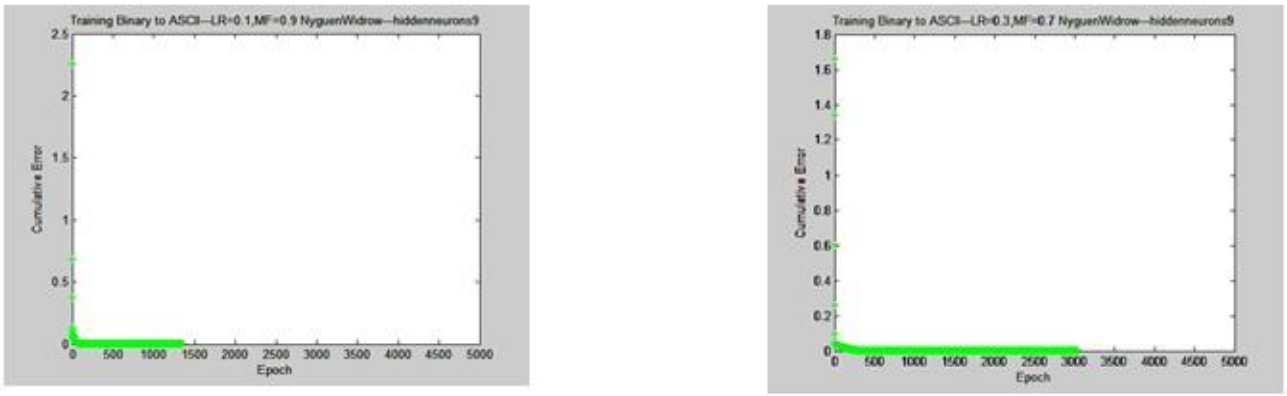

Fig 6a and 6b. Learning curves for mf=0.9 and mf=0.7

VI.NUMBER OF HIDDEN NEURONS:

Number of Hidden layer neurons play vital role in the performance. If more number of neurons is present then error calculations may consume more time and the convergence may be slower.If less number of neurons is present then the accuracy is lost.There is no Hard and fast rule or formula to fix the number of Hidden layer Neurons. In this work Number of Hidden Neurons are set as 5, ,9,12 and 14 and the performance is analyzed. The setting of different values to the network parameters and the effects are presented in Table 2.

Cite this article as: N.Mohana Sundaram, P.N Ramesh. "Optimization of Training phase of Elman Neural Networks by suitable adjustments on the Network parameters." *International Conference on Systems, Science, Control, Communication, Engineering and Technology (2015)*: 229-235 Print.

Table 2

Simulation Results --- Optimum condition is highlighted

NI=7, NO=7, NC=NH, Momentum factor = 0.9,
Input type= Bipolar inputs (-1,1) , SSE Tolerance= 0.0001

S.No.	LR	Weight initialization	NH	No of Epochs	% error * (calculated based on the number of wrong outputs out of 30 test data)
1	0.1	Random weights With both –ve and –ve values	5	2474	3.33
2			7	2350	3.33
3			9	1601	0
4			12	1693	0
5			15	2891	0
1	0.1	Nyguen- Widrow method Values between -0.01 to 0.70	5	1637	3.33
2			7	1712	0
3			9	1300	0
4			12	951	0
5			15	1255	0
1	0.2	Nyguen- Widrow method Values between -0.01 to 0.70	5	1432	66.66
2			7	2615	66.66
3			9	NO CONVERGANCE	
4			12		
5			14		

NI=7, NO=7, NC=NH, Momentum factor = 0.9,
Input type= Binary Inputs (0,1) , SSE Tolerance= 0.0001

S.No.	LR	Weight initialization	NH	No of Epochs	% error
1	0.1	Nyguen- Widrow method Values between -0.01 to 0.70	5	5085	6.66
2			7	4823	0
3			9	3745	0
4			12	3367	0
5			15	4756	0
1	0.1	Random weights With both –ve and –ve values	5	5800	0
2			7	4700	0
3			9	4050	0
4			12	NO Convergence	No output
5			15		

NI=7, NO=7, NC=NH, Momentum factor = 0.7,
Input type= Bipolar inputs , SSE Tolerance= 0.0001

S.No.	LR	Weight initialization	NH	No of Epochs	% error
1	0.1	Nyguen- Widrow method	5	5007	3.33
2			7	3702	0
3			9	3479	0
4			12	3063	0
5			14	4634	0

*The % error is calculated based on the number of incorrect outputs.Out of 30 input data if one output goes wrong then the % error is 1/30 (3.33%)

VII.DISCUSSIONS

A set of 60 data of containing 7 binary values (ASCII equivalent) and corresponding normalized decimal values are used for training.For testing 20 data are used.A set of 10 simulations are performed for each condition and the average values are tabulated. From the above shown results the weight initialization plays an important role in optimizing the learning with out affecting the accuracy. Using the Nyguen-Widrow weight initialization method the optimum condition is arrived with 951 epochs when the learning rate is 0.1, the momentum factor is 0.9 and the number of hidden neurons are 12.If the learning rate is increased even though quicker learning is achieved the errors are more implying that the network has not learned properly. If momentum factor is decreased the learning becomes slow.

VIII.CONCLUSION

In this work the effects the variations of the parameters like initialization of weights, number of hidden neurons, input data, learning rate and momentum factor on the training the Elman Neural Network are analyzed and the optimum condition of the parameters is arrived. Normally the bipolar data types are to be used rather than binary type , because for binary data the network will take a long time to get trained and sometimes no convergence is arrived. Like wise Weights are initialised using Nyguen-widrow method for faster and accurate learning. Further a small learning rate and a large momentum factor are to be set for faster and accurate learning.

REFERENCES

[1]. ELMAN, J. L. "Finding Structure in Time". Cognitive Science 14, 2 (apr 1990), 179–211.

[2]. ZhiQiang Zhang, Zheng Tang and Catherine Vairappan,"A Novel Learning Method for Elman Neural Network Using Local + Search", Neural Information Processing – Letters and Reviews, Vol. 11, No. 8, August 2007.

[3]. D.L. Wang, X.M. Liu and S.C. Ahalt, "On temporal generalization of simple recurrent networks,"Neural Networks 9 (1996) 1099-1118

[4]. Yu-Tzu Chang, Jinn Lin, Jiann-Shing Shieh and MaysamF. Abbod ," Optimization the Initial Weights of Artificial Neural Networks

Cite this article as: N.Mohana Sundaram, P.N Ramesh. "Optimization of Training phase of Elman Neural Networks by suitable adjustments on the Network parameters." *International Conference on Systems, Science, Control, Communication, Engineering and Technology (2015)*: 229-235 Print.

via Genetic Algorithm Applied to Hip Bone Fracture Prediction", Hindawi Publishing Corporation,Advances in Fuzzy Systems.Volume 2012, Article ID 951247, 9 pages doi:10.1155/2012/951247.

[5]. Jim Y.F. Yam, Tommy W.S. Chow, "A weight initialization method for improving training speed in feedforward neural network", Elsevier Neurocomputing 30 (2000), 219-232.

[6]. Herbert Jaeger, Fraunhofer Institute for Autonomous Intelligent Systems (AIS), "A tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the"echo state network" approach", Fifth revision: Dec 2013

[7]. Derrick Ngyuyen and Bernard Widrow, "Improving the Learning Speed of 2 Layer Neural Nwtworks by choosing Initial Values of the Adaptive Weights", Information Systems Laboratory, Stanford University, CA 94305

[8]. Frauke Günther and Stefan Fritsch, "Neuralnet: Training of Neural Networks", The R Journal Vol. 2/1, June 2010, ISSN 2073-4859 [9]. A.Moghaddamnia , R.Remesan , M.HassanpourKashani , M.Mohammadi , D.Han e, J.Piri , "Comparison of LLR, MLP, Elman, NNARX and ANFIS Models—with a case study in solar radiation estimation" , Elsevier Journal of Atmospheric and Solar-Terrestrial Physics- 71 (2009) 975–982

[10]. Anqi Cui, Hua Xu , Peifa Jia, "An Elman neural network-based model for predicting anti-germ performances and ingredient levels with limited experimental data", Elsevier Expert Systems with Applications 38 (2011) 8186–8192.

[11]. Cheng-Yuan Liou , Jau-Chi Huang, Wen-Chie Yang, "Modeling word perception using the Elman network", Elsevier Neurocomputing 71 (2008) 3150– 3157

[12]. Zhihang Tang, Rongjun Li , "An Improved Neural Network Model and Its Applications" , Journal of Information & Computational Science 8: 10 (2011) 1881–1888.

Cite this article as: N.Mohana Sundaram, P.N Ramesh. "Optimization of Training phase of Elman Neural Networks by suitable adjustments on the Network parameters." *International Conference on Systems, Science, Control, Communication, Engineering and Technology (2015)*: 229-235 Print.