Cluster based pruning and survival selection using soft computing

S Shivaji [#1], R Muthaiah [*2]
[#] School of Computing, M.Tech VLSI design, SASTRA University, Thanjavur, TamilNadu, 613401, India
[1] ersshivaji@gmail.com,
2 sjamuthaiah@core.sastra.edu

***Abstract*—Self-adaptive evolutionary constructive and pruning algorithm (SAECPA) is a new structural hearing algorithm, which be planned for proposing of artificial neural networks (ANNs). SAECPA begins with set of ANN and it's a simplest formation one hidden neuron is linked towards single input node, in that network intersect and network transmutation which increases the network inhabitants, then using cluster pruning (CP) and survival selection (SS) to prune the network.As a manifestation of the method, SAECPA is concerned to the forecasting problem - the Mackey-Glass time series. Here user defined constraints intersect probability (p_c) and transmutation probability (p_m) are considered as input, but it may well be developed self-adaptive to enlarge the unknown neurons and links further proficiently.**

Keyword- Evolutionary, Pruning, Prediction, Self –adaptive, Artificial neural network.

I. INTRODUCTION

Accurately compute the developments of time series various numerical algorithms has been proposed. For forecasting and prediction purposes, feed-forward ANN with one hidden layer be enough towards reach any wished accurateness. Thus constructive-pruning algorithm having ANN structure of three-layered ANN with single hidden neuron [2]. It adjoins hidden neuron during training phase one by one. Benefit of the algorithm is we can set initial phase simply. In pruning algorithm (PA), we remove unnecessary hidden neurons during training. In constructive algorithm (CA), we estimate the number of hidden neuron via constructive method. In (ECPA) the network topology using pruning and constructive methods in advancement manner is directed[1]. Also it is not predefine the chromosome length; due to variable-chromosome demonstration is accepted. Thus it makes the memory usage efficiently. Then the positive method initiates into intersect and transmutation operations which set the initial network with one input and one hidden neuron. After that crossover and mutations extend the structural design by adding together hidden neurons. In Self adaptive method, intersect and transmutation operations create as self-adaptive. SAECPA developed a design consists of pruning and survival selection which prunes the resulting networks[3].

II. SELF-ADAPTIVE ECPA

As discussed in [2], a neural network of feed forward with single input layer, one unknown layer with single output layer is designed. Fig. 1 shows summarization and explanation of major steps of the self-adaptive ECPA.

Primary phase

(Step 1) Design a preliminary population with population size (N_p) ANNs (i.e., three-layer feed forward ANN), where the ANN starts with single input layer, one unknown layer, withsingle output layer.

(Step 2) Tutor all thenetworks by means of back propagationalgorithm for φ epochs[4].

(Step 3) Determine its fitness.

Secondary phase

(Step 4) Using tournament selection method select two parents. And by network intersect with intersect probability (P_c) produces one child from the two parents[10].

(Step 5)Do transmutation with transmutation probability (P_m), to the off spring.

(Step 6)Make the intersect probability (P_c) and the mutation probability (P_m), as self-adaptive[3].

(Step 7)Using BP Train the child ANN for φ epochs and find out its fitness.

(Step 8)Apply cluster based pruning on the off spring. Go to Step 5 until N_p offspring is generated. Devise self-adaptive.

(Step 9)Perform Age-Based Survival Selection to childwith their parents which produce the primaries of the subsequent age group. Go to pace 5 until the utmost quantity of age groups (G) is achieved[10].

(Step 10) Choose a single finest neuralnetwork among the final inhabitants.

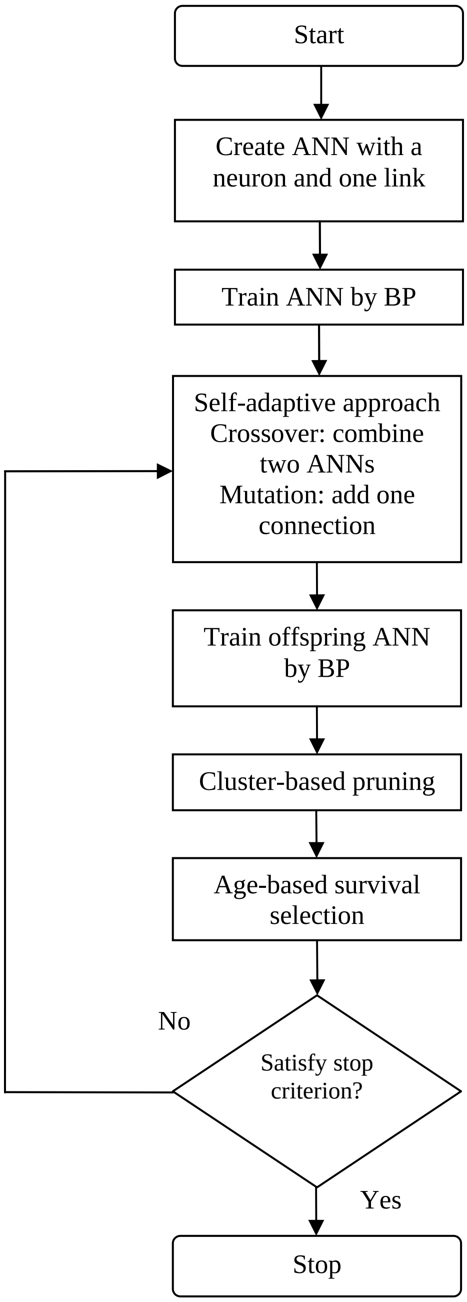

Fig. 1 Main paces evolved in Self-Adaptive ECPA

A. Primary population and encoding scheme

Initially ANN starts with single input layer, oneunknown layer, withsingle output layer, here randomly selected a single link starting one of the inputs. The first loads are produced arbitrary by an identical allocation in the series [-1.0, 1.0] as suggests in [6]. For encoding a network into a gene it be signified as a vector, and its size define the length of ANN, hence its memory be able to use efficiently[6][5].

Fig. 2 shows the chromosome representation of two ANNs. It consists of the network weights and connections where the first unknown neuron output load, bias weight with connected load among first unknown neuron and second input node indicate as w_1, w_{b1} and w_{12}.

ANN

w_1	w_{b1}	w_{11}	w_{12}	w_{13}	w_{14}
0.7	0	0	0	2.5	−0.2

Fig. 2 Encoding of ANN

The input-output connection for the above ANN is as follows: $Y = W_1 . h (W_{11} . X)$ (1)

Where W_1 is the yield weight, X is the input node, is the weight linked from X to the hidden node and h is the hidden node opening task. It is able to a hyperbolic tangent function.

B. Self-adaptive Approach

1. Network intersect

Network intersect is done by selecting two parents (ANNs) through tournament selection method and combines them together. An offspring ANN is created from combining of the two parents. But here we employed self-adaptive network crossover instead of network crossover. So the user specified parameter crossover probability (p_c) could be created as self-adaptive which increases the performance and convergence of the algorithm. Now consider two ANNs ANN_a, ANN_bwhich are having one and two hidden neurons respectively. And their input-output relationship of them as follows:

For $ANN_a Y^a = W^a{}_1 .h (W^a{}_{11} . X_1 + W^a{}_{12} . X_2)$ (eqn. for one hidden neuron) (2)

For $ANN_b Y^b = W^b{}_1 .h (W^b{}_{11} . X_1) + W^b{}_2 .h (W^b{}_{22} . X_2 + W^a{}_{24} . X_4)$ (eqn. for two hidden neurons) (3)

Where W_1 is the output load, X is input node, is the load linked from X to the hidden node and h is the hidden neuron opening task. Superscript indicates the network key, and then subscript signifies the relation between neurons.Network intersect is created from joins the bases of ANN_a and ANN_b, hence it gives a subsequently offspring ANN_c[6][7]. The relationship eqn. as follows:

$Y^c = 1/2(Y^a + Y^b)$ (4)

i.e., $Y^c = 0.5W^a{}_1 .h (W^a{}_{11} . X_1 + W^a{}_{12} . X_2) + 0.5\ Y^b = W^b{}_1 .h (W^b{}_{11} . X_1) + W^b{}_2 .h (W^b{}_{22} . X_2 + W^a{}_{24} . X_4)$ (5)

2. Network Mutation

In network crossover some offspring ANNs have additional secreted neurons, so it have much procedure ability, thus it be able to incompetent by only adding the hidden neurons for ANN performance. It is feasible to set up more inputs into every unknown neuron to enlarge the forecast accurateness. For that transmutation set ups a fresh link into network, the link be built among casually certain an input and a unknown neuron. The first loads are generated randomly by an identical allotment in series [-0.01, 0.01]. The input-output relationship as follows:

$Y^b = W^b{}_1 .h (W^b{}_{11} . X_1 + W^b{}_{12} . X_2) + W^b{}_2 .h (W^b{}_{22} . X_2 + W^a{}_{24} . X_4)$ (6)

Where $W^b{}_{12} = 0$ and thus, $ANN_{b'}$hold on to the concert of ANN_b. For more progress the act of $ANN_{b'}$, the training method performs Step 5[8].

3. Cluster Pruning

The CBP operators keep hold of major neurons with prune irrelevant neurons on an odds basis and so prevent the model development of neural network. CBP performed in two steps.

(Step 1) The consequence of each unknown neuron isfound out. For theith unknown neuron, the consequence is defined as$\sigma_i = \sqrt{S_i}$ (7)

Where$S_i$is obtained as follows [7], [8]: $S_i = \sqrt{\frac{\sum_{p=1}^{p}(S_i)^2}{P}}$ (8)

Thus, S_i is root-mean square of S_i^P, indicates the sensation of the group output $o^P$to the output $h_i^p$of the ith unknown neuron for the pth model, articulated as

$$S_i^P = \frac{\partial o^P}{\partial h_i^P} = w_i \quad (9)$$

Here, $w_i$is the load of the link from the ith unknown neuron to the yield neuron. The load is stable for the reason that it is unrelated to the models. So the above eqn. be able to rewrite as

$$\sigma_i = \sqrt{|w_i|} \quad (10)$$

Thus low significance hidden neuron can be removed.

(Step 2) Categorize the hidden neurons as good and poorer, the consequence of one unknown neuron is near to fine example then it categorized in good class otherwise it categorized in poorer class[1][9].

4. ABSS

When the network intersect, network transmutation and cluster based pruning finished, those in the next age group selected during the survival selection. ABSS performed in two steps.

(Step 1) Occupies usual event choice to choose $N_p$candidates for the subsequent step.

(Step 2) To remove the elderly networks from the $N_p$candidates by the fitness guide[1][10]. It is given as follows:

$H_j \ = (1 - 1/Agej)^2$ (11)

III. MACKEY-GLASS TIME SERIES RESULTS

Mackey-Glass time series estimate be familiar standard crisis in the region of artificial neural network[11][6]. Here we have to predict the performance and regression of the time series. By Euler's MG differential equation:

ds (t)/dt = 0.2s (t-tau)/ (1+s (t-tau) ^10) - 0.1s (t) (12)

This is a function for generation Mackey-Glass Time Series. Initial values are generated randomly using rand function. Let us consider nm is 3000, where MG Series is periodic for tau<17 otherwise non-periodic.

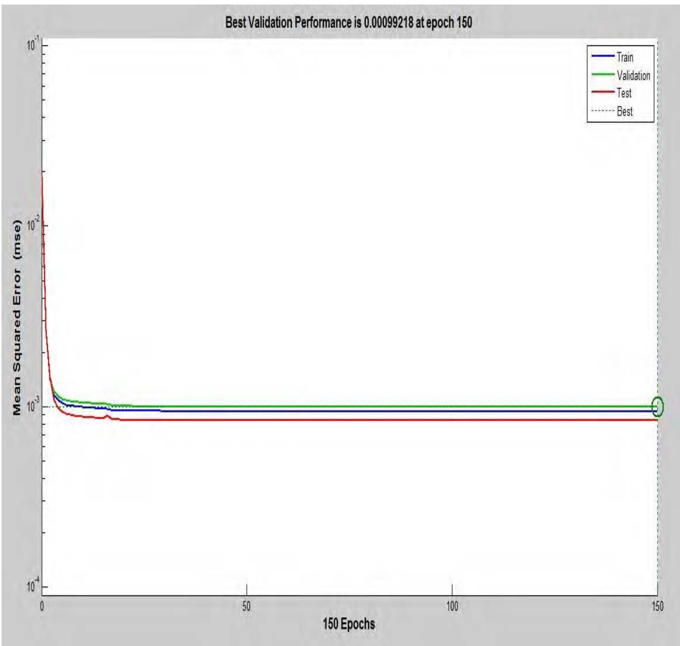

Fig 3 Performance of Mackey-Glass time series

Fig 3 illustrates the performance curve of time series, the graph is plotted between Mean Squared Error (MSE) and Epochs. And we get the best validation performance of 0.00099218 achieved at 150 epochs.

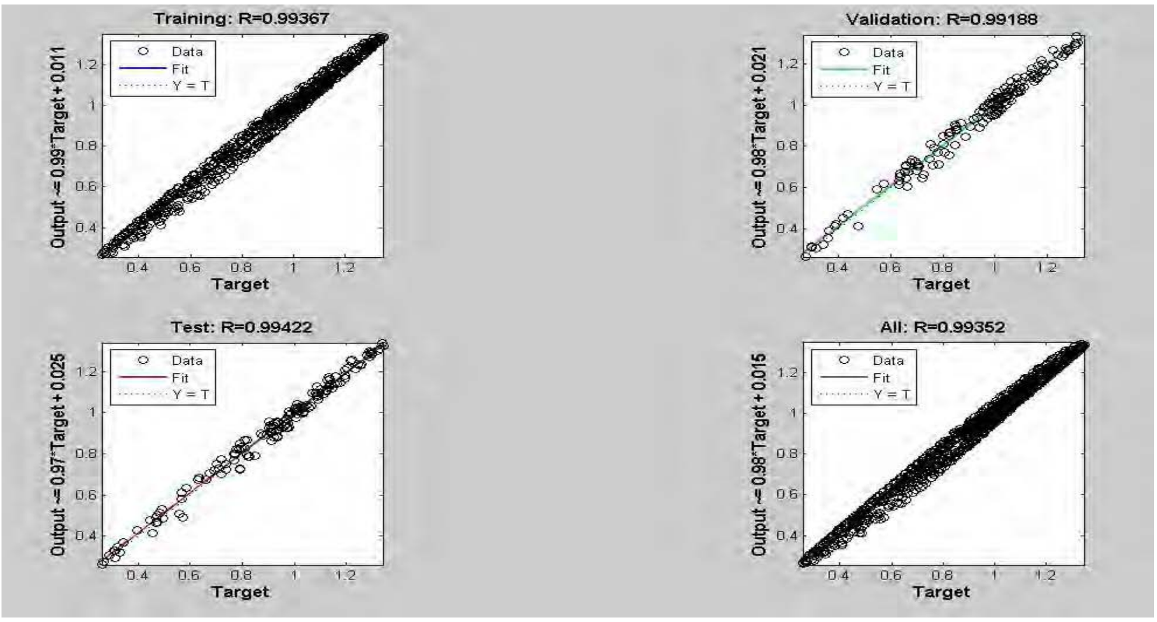

Fig 4 Regression of Mackey-Glass time series

Fig 4 shows the liaison among the network outputs with their targets. If the training were perfect, the relationshipwould be accurately equal; however the relationship is infrequently perfect in practice. The four axes indicate the data of training, testing,validation and all of them. The sunken stripe in all axis signifies the faultless result condensed from the output provide the target. The hard lines signify the finest fit linear regression line among the relations. The R data is a hint of the outputs in addition to targetsrelationship. If R = 1, which specify that around is an exact linear relationship among outputs with targets. But R is near to zero, and after that there be no linear relation between outputs with targets. In this casing the training data specifies a fine fit. The validation and test results also show R values that greater than 0.9.

IV. CONCLUSION

A novel self-adaptive ANN structure have been proposed and implemented, it was tested using Mackey-Glass time series for prediction, which indicates a fine performanceof 0.00099218 and best fitness of R value greater than 0.9 (i.e., 0.99352).

REFERENCES

[1] S.-H. Yang, Y.-P. Chen, An evolutionary constructive and pruning algorithm for artificial neuralnetworksand its prediction applications,ELSEVIER Neurocomputing 86 (2012) 140–149
[2] T.Y. Kwok, D.Y. Yeung, Constructive algorithms for structure learning in feed forward neural networks for regression problems, IEEE Trans. Neural Networks 8 (1997) 630–645.
[3] Oliver Kramer, Chuan-Kang Ting, Self-adaptive EvolutionaryAlgorithms, Bartlomiej GlogerMatr.-Nr:6054499, 28. Januar 2004.
[4] Y. Hirose, K. Yamashita, S. Hijiya, Back-propagation algorithm which variesthe number of hidden units, Neural Networks 4 (1991) 61–66.
[5] F.H.F. Leung, H.K. Lam, S.H. Ling, P.K.S. Tam, Tuning of the structure andparameters of a neuralnetworkusing an improved genetic algorithm, IEEETrans. Neural Networks 14 (2003) 79–88.
[6] H. Du, N. Zhang, Time series prediction using evolving radial basis functionnetworks with new encodingscheme, Neurocomputing 71 (2008)1388–1400.
[7] C. Emmanouilidis, A. Hunter, and J. MacIntyre, (2000) "A MultiobjectiveEvolutionary Setting forFeatureSelection and a Commonality-Based Crossover Operator" CEC '2000. The 2000 Congress on Evolutionary Computation, San Diego, California, USA.
[8] P.P. Palmes, T. Hayasaka, S. Usui, Mutation-based genetic neural network,IEEE Trans. Neural Networks 16(2005) 587–600.
[9] R. Reed, Pruning algorithms—A survey, IEEE Trans. Neural Networks 4 (1993) 740–747.
[10] R Sivaraj, T Ravichandran, A Review of Selection Methods in Genetic algorithm, IJSET (2011) 3792-97.
[11] G.P. Zhang, Time series forecasting using a hybrid ARIMA and neural networkmodel, Neurocomputing 50(2003) 159–175.