Back Propagation Artificial Neural Network Structure Error Reduction by Defined Factor of Capacity and Algorithm Reinforcement Method

V. Rahmati, M. Husainy Yar, J. Khalilpour, A. R. Malekijavan

Abstract—This paper investigates how to reduce error and increase speed of Back propagation ANN by certain defined Capacity factor. For the years from 1965 to 1980 the use of a variety of ANNs for problem solving was retarded significantly because of limitations in one layer networks that weren’t good enough for enhancements of a specific issue, although there were low expectancies for even simple tasks and mathematical operations. Multi-layer networks have a serious covenant to improve this privation by more effective error reduction for example by least squares error method and a better learning factor like the one that is considered in MLP which is modified, enhanced version of Perception network that has provided a better chance of using these networks for intelligent signal processing. But the purpose of this paper is not showing capabilities of these networks alone but to consider error reduction while the weighting equations both satisfy ordinary task of algorithm and at the same time reduces presumptions of errors by a predetermined Capacitance factor that is not very anomalous to other bunch of clustering pedagogy styles anent the other types of ANNs. Unlike a single layer network with many limitations in learning, approximating and estimating a mapping function, multi-layer networks are well prepared for estimation of any uniformly continues subordination with tunable accuracy. Hidden layer in many applications does the job of enhancement, but sometimes poly-layer methods are used for this error reduction separately by some factor definitions (and new hidden parts that paper adds to gets error reduced) that paper tries to measure for exact improvements which were envisaged in design process. And as a result understand how to use Capacity factor for BPANN algorithm, and error reduction in general that holds convergence, speed improvement and error smoothing at the same time.

Index Terms—BPANN enhancement, Error smoothing, MLP, Intelligent signal processing

I. INTRODUCTION

It was always interesting for authors to use some gimmicks to make certain ANN algorithms faster by defining a new factor [1] that modifies weighting matrix to resemble a faster convergence because of inline algorithm code modifications that steps down the possibility of producing error because of \( \sigma_c \) factor that keeps its gradient parameter dynamically changed respect to slope parameter [2-10], this is also proportional to better learning of certain machines. Seemingly for latent (hidden) layers the same story but this time for \( \Delta w_{ij} \) occurs using small case indices while calculating first derivative of error minimized as it is equal to

\[
-\left[ t_k - y_{ik} \right] f'(y_{in_k}) z_k
\]

so each round of sums for hidden layers that uses this factor reclusively is not rectified well and this all happens in the existence of arbitrary activation function [20-30]. If user needs to design a Back propagation ANN algorithm some considerations for weighting and hidden layers should be added to scheming process in which a harmonic single stage inline error reduction happens [31] exactly at the time of weighting vectors and because error itself is used for calculation of \( \delta_k \) that is for single output node \( y_k \) error compared to target \( t_k \) and it happens in learning process, the error reduction also involves hidden nodes joint with \( Y_k \) at timepieces. Weighting for first layer cannot [32,33,34,35] be modified while the new weights for other layers are not ready; the reason is simple and because \( \delta_k \) is not treated as a multi stage factor for layers and weighting must be done for all layers at the same time, for e.g. \( Z_i \) with hidden value \( z_j \) and weight of \( v_{ij} \) that primarily is calculated by \( \delta_i \) and because this is essential for original structure of algorithm new method wouldn’t reclaim structure but tries to enhance error cutback by some strong mathematically supported ideas. As learning process for these types of networks have 3 main steps of Feed forward for input pattern, calculation and Back propagation of error and weight calculation, paper also uses the same specimen for speed improvement but using its own definitions inside new algorithm to eke learning ratio and acquisition.
Back Propagation Artificial Neural Network Structure Error Reduction by Defined Factor of Capacity and Algorithm Reinforcement Method

I. INTRODUCTION

Capacity [36-50]. As title inculcates, improvement for error reduction has to be gathered by factor of Capacity which will be shown and this factor itself is calculated by Reliability factor; that is the result of calculation of $p_c$ and $p_w$ according to $\delta_i$ coefficient for each round of algorithm and is considered fully in algorithm implementation part [51-55].

II. APPROXIMATIONS V POINT TRACKING

A good Feed forward function estimator ANN that is used as Universal Approximator for any continues function like Kolmogorov that solves an important problem of multi variable function survey by single variable functions as (1-1):

$$f(x) = \sum_{i=1}^{2n+1} x_i (\sum_{i=1}^{n} \psi_j(x_i))$$

(1-1)

Is an example of successful point tracking while the rest of the mathematically algorithm don’t get even near error like it. For a network with 2 latent layers of nodes, $Z$ and $ZZ$ unites that can be used by bias without any diagonal predefined value shown by $w_{o,i}$ for any output node $Y_k$ and latent nodes bias value of $v_{0,j}$ for hidden bias node of $Z_j$.

Robot Path Planning: Now what about a robot path-planning techniques that is simply divided into 2 main categories, first by current points that robots get from sensors and is called local planning and the second type is global path-planning from a map that is saved in memory and is processable and also accessible for each error reduction variable that depends on ANN type and its algorithm. The best idea for reducing error in this case is using both strategies because a map can be misleading if the path is changed fundamentally or a local planning has issue of non-optimal solution choice, means the path is okay but complex. For applications, robot uses several sensor data maps and compares it using for example Cognitron for path detecting and for this detection excitatory and inhibitory inputs denoted by $e$ and $h$ are considered separately and by Fukushima notations:

$$u(k) = e \left[ 1 + \frac{e}{1 + h} - 1 \right]$$

(1-2)

$$e - h = \frac{e - \eta}{2\eta} \left( 1 + \tanh \left( \frac{1}{2} \log nx \right) \right)$$

(1-3)

Please care that an ordinary one to one correspondence relation is used for activation function and note how this approach can show distinction when an excitatory input must be distinguished from inhibitory one; as an example imagine movement of an object by an engine force and outer power. If you switch machine [73] on and start to move while having an ordinary soaring speed by input variable $f_c$ force of engine or when you push/pull that object by outer power, movement appears as that expected output but this time engine is off and the source of potency is different and dynamic response of object is different too! By this notation it is possible to distinguish 2 different types of variables that both have the same result.

III. ALGORITHM CONSIDERATIONS

Generalized Back propagation single layer ANN can have several latent layers in which each input node receives one input signal (rare cases the input itself can be divided separately and receives mixed signal like adders or a bias can have similar task) and sends this signal to latent layers nodes of $Z_i$ in first stage. Activators, then calculate correct value and send it back for $ZZ_i$ unites again activation process happens for second latent layer to find network response of input pattern using $y_j$ for $Y_i$ [56-70].

IV. ALGORITHM IMPLEMENTATIONS

Feed Forward

In this section a vivid process for implementation of a Back propagation ANN with two latent layers and also $M$ successful subsumed clusters is studied. In this type of ANN bound for
number of samples that was needed, is minimum of number of inputs and latent nodes multiplied by $M$ [59]. Now consider an ANN with two latent layers, and $n$ input units of $X_i$ that send signals to hidden parts called $Z_h$ so to calculate input signal consider (4):

$$Z_{ih} = u_{0h} + \sum_{i=1}^{n} x_i u_{ih} \tag{4}$$

Now the activation function for output nodes is as (5):

$$z_h = f(Z_{ih}) \tag{5}$$

So output is also injected to second (hidden) layer for preprocessing and generating output main signal, which are $ZZ'_j$ nodes. Now to calculate input signal consider (6):

$$zz_{inj} = v_{0j} + \sum_{k=1}^{q} z_h v_{hj} \tag{6}$$

Again output activation process to get updated value is defined this time by (7):

$$zz_j = f(zz_{inj}) \tag{7}$$

Each $m$ output node $Y_k$ by weighted input signal is as (8):

$$y_{inj} = w_{0k} + \sum_{j=1}^{p} zz_j w_{jk} \tag{8}$$

And once again activation function defines outputs nodes deal as (9):

$$y_k = f(y_{inj}) \tag{9}$$

### Back Propagation of Error

This part contains ordinary modified main Back propagation algorithm plus $C'$ calculation which is inline and dwindles error by some important considerations as follow:

#### A) Algorithm Enhancement

a. $C'$ is calculated inline, unlike SOM based optimizations this factor is essential to each step calculation of $\delta_k$ and not each new input vector, the one estimated successfully for SOM based nets(check proof 1).

b. $p_{c}$ and $p_{w}$ are calculated for each stage in which weights have to be updated by $\Delta u_{ih} = \alpha \delta_k x_i$.

Consider each output unit as $Y_k$ then error for current existing learning pattern is (10):

$$e_k = (t_k - y_k) \tag{10}$$

And dip of the activation function multiplied by error gives static $\delta_k$:

$$\delta_k = e_k f'(y_{inj}) \tag{11}$$

Here the $C'$ factor shows its inscription after calculation of $p_{c}$ and $p_{w}$ for each stage:

$$\delta_k' = C' e_k f'(y_{inj}) \tag{12}$$

Then weight has to be updated through $\Delta v_{hj}$ and consider (17) and (18) for bias:

$$\Delta v_{hj} = \alpha \delta_j' z_h \tag{17}$$

For first hidden layer $z_h, h = 1, 2, ..., q$, then consider (19):

$$\delta \tag{19}$$

And weight correction is (20):

$$\Delta u_{ih} = \alpha C' e_k f'(y_{inj}) x_i \tag{20}$$

This will lead to an improvement which has been done once for a chosen Self-Organizing-Map[1] for example Kohonen Map of certain inputs (multi vector improvement) by calculation of $C'$ as an inline factor of improvement and modifying static weighting process [74-78].

#### B) Theorem 1

Error reduction in presented algorithm is achieved if and only if $\rho = 0$ as follows:

$$\rho = \left| \frac{\Delta u_{ih}}{\Delta u_{ih}} \right| \tag{21}$$

#### C) Proof 1

To prove this identity, let’s make sure that these assumptions all hold:

1- Preserved error of each round is near zero, and the convergence criterion holds.
2- $C'$ Renewed for every new input vector (unlike SOM type and because $\delta_k$ has to be calculated separately and our assumptions have to be based on assumption of original algorithm).
3- Weighting process is independent of $C'$ calculation process.

Now consider (22) and (23):

$$E \to 0 \text{ then } \delta_k = 0 \text{ so } t_k = y_k \tag{22}$$

$$\rho \to 0 \text{ because } t_k = y_k \text{ and } C' \rightarrow 1 \text{ so } \rho \to 0 \tag{23}$$

And again and again $\rho$ tends to zero in this loop in a few rounds as $C'$ calculated for every $\delta_k$, in fact $\rho = 0$ is a detector for us that shows how to end our calculations, because slightest mistake
will make too much error mostly for upcoming event predictions that heavily relies on existed information (mapped data). As mentioned in literature the most interesting effect of Back propagation method applications is on nonlinear models and many have tried to improve multilayer networks because of their abilities of problem solving. Researchers have arrived at this point that even by using Back propagation training methods it is still hard to achieve best performance (Fast learning) because of intrinsic parts of algorithms themselves and modifications have to be done to both reduce error and to elevate speed proficiency. The aim of this paper will be accomplished if the new defined Capacity factor is used for algorithm enhancements as next part objective is to show this fact.

V. RESULTS

In this section the results of simulations that have led to speed enhancement and error reduction of a simple Back propagation algorithm with 2 latent layers is brought. Indeed speed improvement is because of better convergence that is the result of error reduction. This new Capacity factor makes algorithm faster and is calculated inline while the algorithm is doing its weight correction as mentioned by (20) and reliability factor is also part of C’ calculations as each round of algorithm is done. Unlike SOM this method won’t be calculated for every new single vector, but consider that in algorithm the C’ does not affect propagation between layers, because in more than 2 latent based networks time complexity is \(O(t^2)\) which is for both upper and lower layers and is complex enough not to be further modified. Error is also calculated in this algorithm which is difference of target and output node value so this is implemented while output layer Net and error is calculated. So error is distinguished for different number of inputs and change of prediction error while the prediction model is fixed, and also target values are fixed too. In other words, error reduction for such cases that user has more than enough number of data sets for best prediction is expected. As in Fig. 2, Shows error reduced smoothly in C’ based algorithm with \(\rho = 0\) because reliability is calculated inline, and main algorithm also experiments error reduction by increasing number of data sets but it is not as smooth as new algorithm. What mentioned as smooth error reduction has an important mathematical background, because if predication has less error it also estimates real function of that experiment deterministically not because the process is necessarily deterministic, but mostly when designer deal with natural events in real world which are the results of several known and some unknown parameters. While main algorithm error reduction is not very smooth it behaves obscure and is unpredictable. The interesting fact is that smoothing gives an explicit predictable functional equation for sides of algorithm while non-smoothed error reduction is not even safe, and it doesn’t mean that new method always does the task of smoothing.

Next in Fig. 3, The same algorithm for new set of data plus Feed forward modified version that leaves accuracy up to 2 digits after decimal point with \(\rho = 0.01\) is tested. As it is clear in some cases the new algorithm has more error because number of rounds is also reduced and C’ based algorithm with \(\rho = 0\) is in fact slower than C’ based algorithm with \(\rho = 0.01\) but \(\rho = 0\) is more accurate because (21) is calculated up to desired level.

Almost any process can be modeled using (24) that employs different types of \(A(j)\) with some considerations that are out of the topic of this manuscript.
This paper investigated a method that was first done before for a chosen Self Organizing Mapand improvement of convergence and also reduction of error for a faster response of ANN for a certain problem that were all achieved too. But this time instead of SOM, paper did the same task on a Back propagation Network that can also be MLP, so pay attention that learning rules and weightings for biases can be calculated but separately (without ignoring convergence criterion of Capacity factor, in this paper using $\rho$) and readers cannot do this unless they consider convergence criterion for weighting matrix that must be considered for each network alone. Capacity factor calculated inline for several data sets and reduced error was calculated, but concept of error smoothing was not considered in this paper but instead an error reduction strategy was introduced. Paper claimed if error function is smoothed well, regardless of prediction model or shape and even deterministic or ideterministic input signals, the error will be zero for infinity many samples (similar result is not possible for ordinary linear predictors). In other words and with better insight may say, “error smoothing is more valuable than error reduction” because authors can show that unlike ideterministic rough functions, smoothed functions can be predicated better. Finally paper demonstrated that error was decreased better while factor $\rho$ was almost equal zero, and the maximum time of calculation occurs when this factor is exactly equal to zero.

VI. CONCLUSION

REFERENCES

5. Zhi-Hua Zhou, “Rule Extraction Using Neural Networks or For Neural Networks” National Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, China 2007
8. Yanbo Huang. “Advances in Artificial Neural Networks – Methodological Development and Application” United States Mississippi 38776, USA;
9. Ajith Abraham.” Artificial Neural Networks” Oklahoma State University, Stillwater OK, USA 2006
61.
62.
63.
64.
65.
66.
67.
68.

44. Faussett, L. “Fundamentals of Neural Networks” Prentice Hall, USA. 1996.
46. Fuyad, Usama, Ramakrishna “ Evolving Data mining into solutions for Insights”, communications of the ACM 45, no. 8
49. Tamara Knutsen et al, “Designing an Underwater Eel-Like Robot and Developing Anguilliform Locomotion Control” Harvard University 2004
53. Hebah H. O. Nasereeddin “Stream Data Mining” Department of computer Information system Faculty of IT Amman Arab University for Graduate Studies Amman Jordan 2009.
57. Gobinda G. Chaudhary “Natural Language Processing” Dept. of Computer and Information Sciences University of Strathclyde, Glasgow G1 1XH, UK 2003
60. G Vijay Kumar and Dr GV Raju, “Biological Early Brain Cancer Detection using Artificial Neural Network”, International Journal on Computer Science and Engineering, Vol. 02, No. 08, 2010;
70. Mark Fliesser “Foundations of Swarm Intelligence From Principles to Practice” Institute for Systems Research University of Maryland College Park, Maryland 2005
73. Khajanchi, Anit, Artificial Neural Networks: The next intelligence
77. K.R. Chaudhary “Goals, Roots and Sub-fields of Artificial Intelligence. MBM Engineering College, Jodhpur, India 2012