An Artificial Neural Networks approach with NP hardness for efficiency approximation

R.Murugadoss[1*], Dr. M.Ramakrishnan[2]

1. Associate Professor, St Ann's College of Engineering & Technology, Chirala-523187, Andhrapradesh, Research Scholar Dept. of CSE, Sathyabama University, Chennai. murugadossphd@gmail.com
2. Professor & Head, Department of Information Technology, Velammal Engineering College, Chennai. ramkishod@gmail.com

Abstract:

An Artificial Neural Networks approach with NP hardness for efficiency approximation is presented in this paper. The aim of present research is to show that finding the solution and efficiency approximation to the solution of the global optimization problem using NP-hard problem. NP hardness is used as the training algorithm of an artificial neural networks approach for calculating the efficiency approximation. In the ANN initial condition for the input–hidden connection, the left sigmoid function is used as an activation function and for the hidden–output inter connection, the right sigmoid function is used as an activation function.

1. Introduction:

During the last decade, there has been increased use of neural networks (NNs), fuzzy logic (FL) and genetic algorithms (GAs) in several applications. However, the focus often has been on a single technology heuristically adapted to a problem. While this approach has been productive, it may have been sub optimal, in the sense that studies may have been constrained by the limitations of the technology and opportunities. For example, while NNs have the positive attributes of adaptation and learning, they have the negative attribute of a "black box" syndrome. By the same token, FL has the advantage of approximate reasoning but the disadvantage that it lacks an effective learning capability. Merging these technologies provides an opportunity to capitalize on their strengths and compensate for their shortcomings. Global optimization deals with the finding of global maximum or minimum of a given function in a certain feasible region. Global optimization problems are very diverse and come from areas such as chemistry and biology, computer science, engineering, operations research, economics, etc. The NP-hardness of the global optimization problem for the functions restricted to a unit simplex.

2. Artificial Neural Networks (ANNs):

Artificial neural networks (ANNs) are relatively new computational tools that have found extensive utilization in solving many complex real world problems. The attractiveness of ANNs comes from their remarkable information processing characteristics pertinent mainly to nonlinearity, high parallelism, fault and noise tolerance, and learning and generalization capabilities.

Artificial neural networks (ANNs) are flexible computing frameworks and universal approximators that can be applied to a wide range of forecasting problems with a high degree of accuracy and explained by several experts such as S.K. Halgamuge and M. Glesner [1] used a special multilayer perceptron architecture known as FuNe for generating fuzzy systems for a number of real world applications. Marcin Szczuka and Dominik Ślęzaka [2] presented the possible extensions of the classical multilayer artificial neural network model to the situation when the signals processed by the network are by definition compound and possibly structured. Di Wang., Xiao-Jun Zeng and John A. Keane [3] proposed a novel hierarchical hybrid fuzzy neural network to represent systems with mixed input variables. Mehdi Khashei and Mehdi Bijari [4] proposed a novel hybridization of artificial neural networks and ARIMA model to overcome mentioned limitation of ANNs and yield more general and more accurate forecasting model than traditional hybrid ARIMA-ANNs models. Marcin Szczuk and Dominik Ślęzak [5] presented the possible extensions of the classical multilayer artificial neural network model to the situation when the signals processed.

By defining separate membership functions for each fuzzy output class, it allows dynamic adjustment of the functions during training. Real-world systems usually involve both continuous and discrete input variables. However, in existing learning algorithms of both neural networks and fuzzy systems and is represented by so many researchers as John A. Drakopoulos [6] defined a new sigmoidal theory and analyzed a new family of functions called sigmoidal functions. Jie Zhang and Julian Morris [7] studied the process modeling and fault diagnosis using fuzzy neural networks. Ralf Östermark [8] proposed a multi group classification algorithm based on a hybrid fuzzy neural net framework. Camps-Valls, G et al [9] proposed the use of the fuzzy sigmoid function as non-positive semi-definite kernel in the support vector machines framework.

An incremental learning algorithm for feed-forward neural networks used as approximators of real world data. This algorithm allows neural networks of limited size to be obtained, providing better performances and are reported by different people as Leila Ait Gougam et al [10] used a model which takes advantage of wavelet-like functions in the functional form of a neural network for function approximation. Jacques M. Bahi et al [11] presented an incremental learning algorithm for feed-forward neural networks used as approximators of real world data. János D. Pintér [12] reported ANNs as a potentially multi-modal optimization problem, and then introduces a corresponding global optimization framework.

3. ANN Development:

Artificial Neural Network (ANN) is an information processing system that has certain performance characteristics in common with biological nervous system, such as the brain, on a computer. Nowadays, neural networks can be used to analyze the phenomena with no or very complicated algorithmic solutions. The facility to learn by experimental data makes ANN extremely flexible and powerful than other approaches. The network consists of layers of parallel processing elements, called neurons, with each layer being fully connected to the proceeding layer by interconnection fully connected to the proceeding layer by interconnection strengths or weights. Neurons are connected to each other by weighted links over which signals can pass. A neuron is the functioning unit of the nervous system; specialized to receive, integrate, and transmit information. The flow of information moves in the following direction: dendrite to soma to axon to terminal buttons to synapse. Each neuron receives multiple inputs from other neurons in proportion to their connection weights and generates a single output which may be propagated to several other neurons. The schematic drawing of an Artificial Neuron is shown in figure 1.

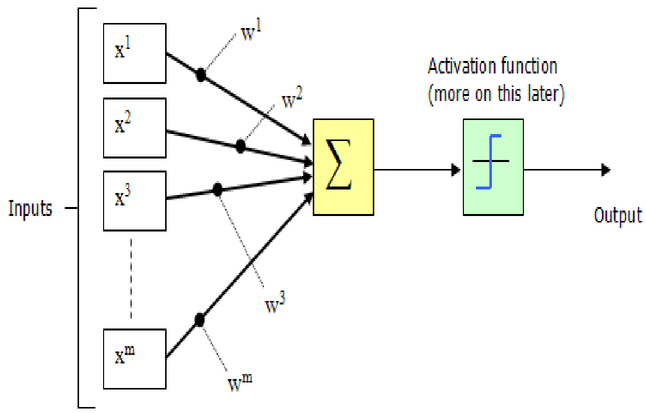

Fig.1: Schematic Drawing of an Artificial Neuron.

The weighted sum of the input components is calculated with equation (1).

$$f(x) = x_1 w_1 + x_2 w_2 + \ldots + x_n w_n \qquad (1)$$

The output is calculated with a sigmoid function as follows

$$OUT = \frac{1}{1+e^{-f(x)}} \qquad (2)$$

The training of the network is accomplished by adjusting the weights and is carried out through a large number of training sets and training cycles (epochs). The goal of the learning procedure is to find the optimal set of weights. The output of the network is compared with a desired response to produce an error. The performance is measured in terms of a desired signal and the criterion for convergence. For one sample, the mean square error (MSE) and the absolute fraction of variance (R^2) are determined as, the Mean Square Error (MSE) was calculated by:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(x_i - x_i^{!})^2 \qquad (3)$$

Where x_i is the observed data, X_i' is the calculated data and N is the number of observations. The MSE can give a quantitative indication of the model error in terms of a dimensioned quantity. An MSE equal to zero indicates a perfect match between the observed and predicted values. The R^2 was calculated by:

$$R^2 = 1 - \frac{\sum (x_i - x_i')^2}{\sum x_i^2 - \frac{\sum (x_i')^2}{N}} \qquad (4)$$

In R^2 efficiency criterion, the best fit between observed and calculated values would have $R^2=1$.

4. Activation Function:

The most common functions are the simple threshold and the sigmoid. The neuron can also work as a logic gate if the function is an And, or, Not. If, And weights = 1, thr = 2; Or: thr = 1. Not: weight = -1, thr =0.

$$\text{Threshold}: g(x) = \begin{cases} 0 & if\ x \le Threshold \\ 1 & Otherwise \end{cases}$$

$$\text{Sigmoid}: g(x) = \frac{1}{1+e^{-x}}$$

4.1 Left sigmoidal Signals:

The function $\sigma : R \to R$ is said to be left sigmoidal.

$$if\ \lim_{x \to -\infty} \sigma(x) = 0$$

The left generalized sigmoidal signals

$$\sigma(x) = \begin{cases} e^{ax}, & for\ x \le 0, \quad \alpha \ge 0 \\ \beta, & for\ x \le 0, \quad \beta \ge 0 \end{cases}$$

4.2 Right sigmoidal Signals:

The function $\sigma : R \to R$ is said to be right sigmoidal signal.

$$if\ \lim_{x \to +\infty} \sigma(x) = 1$$

The right generalized sigmoidal signals

$$\sigma(x) = \begin{cases} 1 + e^{-ax}, & for\ x \le 0, \quad \alpha \ge 0 \\ \beta, & for\ x \le 0, \quad \beta \ge 0 \end{cases}$$

5. Application of ANN:

MAT LAB is used to develop the back propagation learning algorithm has been used in feed forward with one hidden layer. As shown in Figure 2, this network architecture has four neurons in the input layer, five neurons in hidden layer and one neuron in the output layer. The configuration of the ANNs is set by selecting the number of hidden layers and the number of nodes in hidden layer, since the number of nodes in the input and output layers are simply determined from physical variables. This model was trained using Back Propagation Learning Algorithm. In this ANN model, a left and right sigmoid function was used as an activation function.

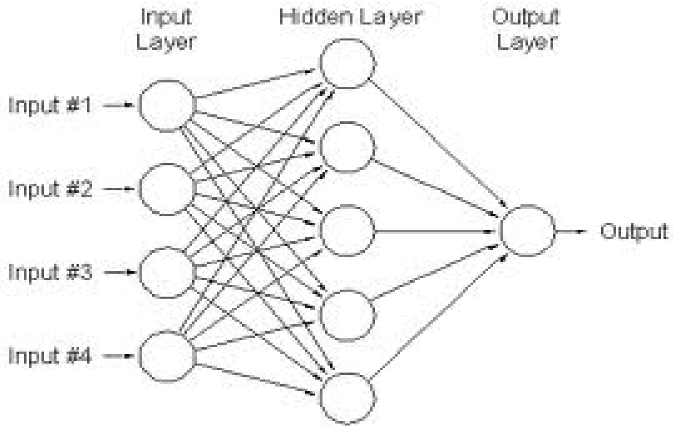

Fig. 2: ANN Architecture

6. Training:

The main aim of the training a neural network is to update the weights; with these weights a set of inputs produces the desired outputs. It calculates the appropriate weight updates, Mean-Squared-Error (MSE) and learning rate to accelerate error convergence. To carry out a fair comparison, the initial state of the networks was kept identical for each experiment. This is fulfilled through training of ANN with NP hardness using the same initial conditions for the input–hidden connection. In this, left sigmoid function is used as an activation function. For the hidden–output inter connection, right sigmoid function is used as an activation function for each experiment. A uniformly distributed initial weight set is produced within the range of [-1, 1]. The network parameters such as learning rate and momentum term are set in all attempts as 0.1 and 0.3, respectively. NP hardness is used as a training algorithm of an artificial neural networks approach for calculating the efficiency approximation.

7. NP Hardness:

NP-hard is a class of problems that are at least as hard as the NP-complete problems. NP-hard problems can be both of decision and optimization types. An optimization version of an NP-complete decision problem is NP-hard. The class NP is defined as a class of decision problems which can be solved by nondeterministic Turing machine in polynomial time. Hence, the term NP comes from nondeterministic polynomial time. Several other experts utilized the NP hard problem in several forms such as David Coeurjolly et al [13] reported that finding a minimum medial axis is an NP-hard problem for the Euclidean distance and compare two algorithms which compute an approximation of the minimum medial axis. Mathias Hauptmann et al [14] studied the approximation complexity of the Metric Dimension problem in bounded degree, dense as well as in general graphs. A greedy constant factor approximation algorithm for this kind of instances and construct an approximation preserving reduction from the bounded degree Dominating Set problem is presented. Xiandong Zhang and Steef van de Velde [15] presented the NP-hard problem of scheduling n jobs in m two-stage parallel flow shops so as to minimize the makespan. Yusuke Matsumoto et al [16] presented a approximation algorithm based on an LP relaxation.

8. NP-hardness of efficiency approximation:

Since we can be almost certain that NP hard problems cannot be solved efficiently, we have to limit ourselves to find approximate solutions. Approximation algorithms can have different measure of efficiency, or so called performance guarantee. An algorithm is said to be an "ε"-absolute approximation algorithm for a problem P, if for for some constant "ε" and for any instance I of this problem it outputs an estimate m^1 such that

$$\left|m^1 - m\right| = \varepsilon \qquad (5)$$

Where m is the exact solution of P more commonly though, performance guarantee is relative, that is the approximate value found by the algorithm is within some fixed percentage of the optimal value. NP hardness is used in the training process of an artificial neural networks approach for calculating the efficiency approximation.

9. Conclusion:

This paper presents that the training of ANN with NP hardness for efficiency approximation used left and right sigmoidal functions as an activation function for the input-hidden and the hidden–output inter connection respectively for each experiment. A uniformly distributed initial weight set is produced within the range of [-1, 1]. The network parameters such as learning rate and momentum term are set in all attempts as 0.1 and 0.3, respectively.

10. References:

[1] S.K. Halgamuge and M. Glesner "Neural networks in designing fuzzy systems for real world applications" Fuzzy Sets and Systems, Volume 65, Issue 1, July 1994, pp 1–12

[2] Marcin Szczuka and Dominik Ślęzaka "Feedforward neural networks for compound signals" Theoretical Computer Science, Vol. 412, Issue 42, September 2011, pp 5960–5973

[3] Di Wang., Xiao-Jun Zeng and John A. Keane "Hierarchical hybrid fuzzy-neural networks for approximation with mixed input variables" Neurocomputing, Volume 70, Issues 16–18, October 2007, pp 3019–3033.

[4] Mehdi Khashei and Mehdi Bijari "A novel hybridization of artificial neural networks and ARIMA models for time series forecasting" Applied Soft Computing, Vol. 11, Issue 2, March 2011, pp 2664–2675.

[5] Marcin Szczuk and Dominik Ślęzak "Feedforward neural networks for compound signals" Theoretical Computer Science, Vol. 412, Issue 42, September 2011, pp 5960–5973.

[6] John A. Drakopoulos "Sigmoidal theory" Fuzzy Sets and Systems, Vol. 76, Issue 3, December 1995, pp 349–363

[7] Jie Zhang and Julian Morris "Process modelling and fault diagnosis using fuzzy neural networks" Fuzzy Sets and Systems, Vol. 79, Issue 1, 8 April 1996, pp 127–140

[8] Ralf Östermark "A fuzzy neural network algorithm for multigroup classification" Fuzzy Sets and Systems, Vol. 105, Issue 1, July 1999, pp 113–122

[9] Camps-Valls, G., Martín-Guerrero, J, D., Rojo-Álvarez, J, L., Soria-Olivas, E "Fuzzy sigmoid kernel for support vector classifiers" Neurocomputing, Vol. 62, December 2004, pp 501–506

[10] Leila Ait Gougam., Mouloud Tribeche and Fawzia Mekideche-Chafa "A systematic investigation of a neural network for function approximation" Neural Networks, Vol. 21, Issue 9, November 2008, pp 1311–1317.

[11] Jacques M. Bahi., Sylvain Contassot-Vivier and Marc Sauget "An incremental learning algorithm for function approximation" Advances in Engineering Software, Vol. 40, Issue 8, August 2009, pp 725–730

[12] János D. Pintér "Calibrating artificial neural networks by global optimization" Expert Systems with Applications, Vol. 39, Issue 1, January 2012, pp 25–32

[13] David Coeurjolly., Jérôme Hulin and Isabelle Sivignon "Finding a minimum medial axis of a discrete shape is NP-hard" Theoretical Computer Science, Vol. 406, Issues 1–2, October 2008, pp 72–79

[14] Mathias Hauptmann., Richard Schmied and Claus Viehmann "Approximation complexity of Metric Dimension problem" Journal of Discrete Algorithms, Vol. 14, July 2012, pp 214–222

[15] Xiandong Zhang and Steef van de Velde "Approximation algorithms for the parallel flow shop problem" European Journal of Operational Research, Vol. 216, Issue 3, February 2012, pp 544–552

[16] Yusuke Matsumoto., Naoyuki Kamiyam and Keiko Imai "An approximation algorithm dependent on edge-coloring number for minimum maximal matching problem" Information Processing Letters, Vol. 111, Issue 10, April 2011, pp 465–468