IEEE International Conference on Advances in Engineering & Technology Research (ICAETR - 2014), August 01-02, 2014, Dr. Virendra Swarup Group of Institutions, Unnao, India

Universal Approximation Using Probabilistic Neural Networks with Sigmoid Activation Functions

R. Murugadoss[1]
Sathyabama University
Department of Computer Science and Engineering,
Chennai, India.

Dr.M. Ramakrishnan [2]
Department of Information Technology
Velammal Engineering College,
Chennai, India.

***Abstract*— In this paper we demonstrate that finite linear combinations of compositions of a fixed, univariate function and a set of affine functional can uniformly approximate any continuous function of n real variables with support in the unit hypercube; only mild conditions are imposed on the univariate function. Our results settle an open question about representability in the class of single bidden layer neural networks. In particular, we show that arbitrary decision regions can be arbitrarily well approximated by continuous feedforward neural networks with only a single internal, hidden layer and any continuous sigmoidal nonlinearity. The paper discusses approximation properties of other possible types of nonlinearities that might be implemented by artificial neural networks. The daily registration has N cases that each of the well-known stimulus-answer couples represents. The objective of this work is to develop a function that allows finding the vector of entrance variables t to the vector of exit variables P. F is any function, in this case the electric power consumption. Their modeling with Artificial Neural Network (ANN) is Multi a Perceptron Layer (PMC). Another form of modeling it is using Interpolation Algorithms (AI).**

***Keywords*— FPGA, Neural Networks, Sigmoid Activation Function, Schematic Tools.**

I. INTRODUCTION

There has been a growing interest in neural network models with massively parallel structures, which purport to resemble the human brain. Owing to the powerful capabilities of neural networks such as learning, optimization and fault-tolerance, neural networks have been applied to the various fields of complex, non-linear and large-scale power systems [l-61. The Hopfield neural network has been applied to various fields since Hopfield proposed the model in 1982[71 and 1984[81. In the problem of optimization, the Hopfield neural network has a well demonstrated capability of finding solutions to difficult optimization problems. The TSP (traveling salesman problem), typical problems of Nondeterministic polynomial)-complete class, AID conversion, linear programming and job-shopping schedule are good examplesI9-121 which the Hopfield network provides with solutions. In the field of power systems, the Hopfield network has been applied to optimal power flow and economic load dispatch problemsIl3-151. In the conventional Hopfield model, the neuron potential has arbitrary numbers during the intermediate stages but at the final stage the neuron potential converges to the limit values (0.1) or (-1,l) and thus, provides a solution. Sometimes it converges to the interior values of a hypercube instead of its vertices[l61. This is regarded as undesirable and calls for improvement. In general an optimization problem frequently needs a large numerical value as its solution. Thus far the counting method or the binary number representationIl71 of various schemes has been used to represent real numbers. These methods, however, employ a large number of neurons to represent a correspondingly large numerical value.

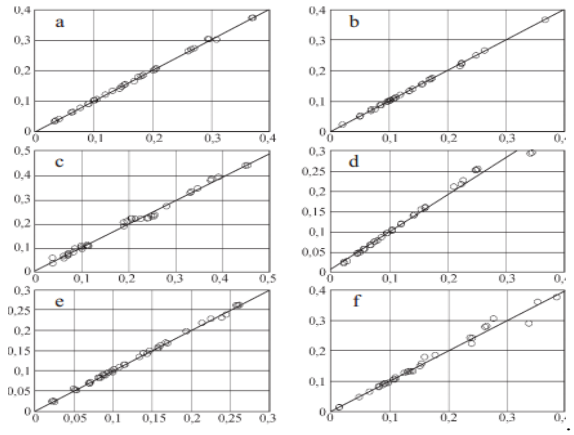

Fig 1 Represent a large value by using a single neuron.

A number of diverse application areas are concerned with the representation of general functions of an n-dimensional real variable, x 9 R", by finite linear combinations of the form. Such functions arise naturally in neural network theory as the activation function of a neural node (or unit as is becoming the preferred term) ILl], IRHM]. The main result of this paper is a demonstration of the fact that sums of the form (1) are dense in the space of continuous functions on the unit cube iftr is any continuous sigmoidal function. This case is discussed in the most detail, but we state general conditions on other possible a's that guarantee similar results.

II. METHODOLOGY

Sigmoidal function is the most used as an activating function. A number of other current works are devoted to the same kinds of questions addressed in this paper. In [HSW] Hornik et al. show that monotonic sigmoidal functions in networks with single layers are complete in the space of continuous functions. Carroll and Dickinson [CD] show that the completeness property can be demonstrated constructively

978-1-4799-6393-5/14/$31.00 ©2014 IEEE

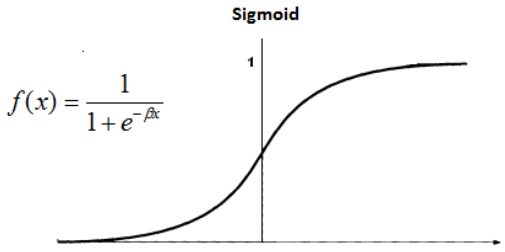

Fig 2 The shape of sigmoidal function.

by using Radon transform ideas. Jones [J] outlines a simple constructive demonstration of completeness for arbitrary bounded sigmoidal functions. Funahashi [F] has given a demonstration involving Fourier analysis and Paley-Wiener theory. In earlier work [C], we gave a constructive mathematical proof of the fact that continuous neural networks with two hidden layers can approximate arbitrary continuous functions. The main goal of this paper is to investigate conditions under which sums of the form

Let a be any continuous discriminatory function. Then finite sums of the form

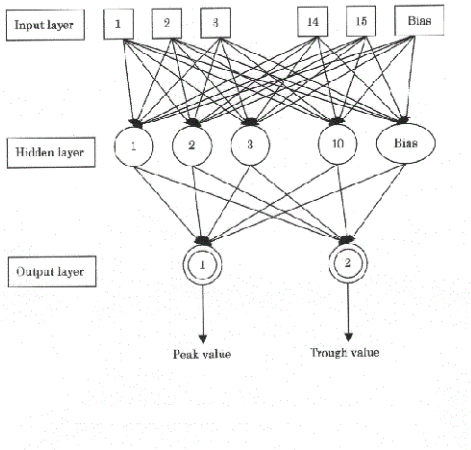

Fig 3 Structure of Three Layer Artificial Neural Network

Are dense in C (l_n) with respect to the supremum norm. In fuzzy logic, fuzzy sets define the linguistic notions and membership functions define the truth-value of such linguistic expressions. Table1 shows the difference between classic sets and fuzzy sets. Membership function defines the membership degree to a fuzzy set. The domain is the universe of discourse (set of values the object can take) and the interval [0,1] is the wrong. Membership functions are often selected triangular or trapezoidal.

III. LITERATURE SURVEY

The artificial neural network (ANN) are suitable for this kind of problems. They have been researched and applied in real systems [1] [3]. Genetic algorithms were presented in several references [3] [5]. In the short period of their developments, AG showed their superior capabilities and have been successfully applied in many fields. An MLP network consists of an input layer of neurons source, one or more hidden layers and an Release layer. The number of neurons in layer input and output based on the number of input variables and desired number of classes respectively.

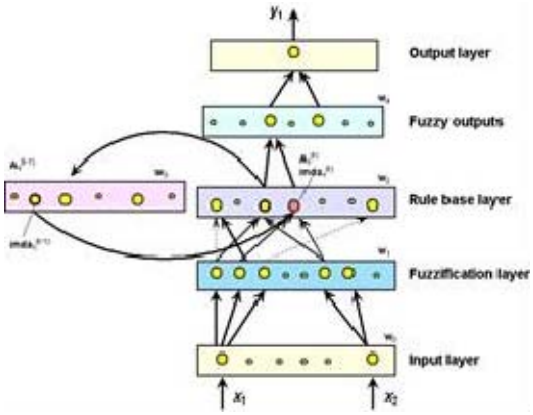

Fig 4 A General RNN Model

The number of hidden layers and the number of neurons in each layer hidden affect the generalizability of the network. In this paper, following the idea of using first-order time derivatives [21], a general RNN model with implicit dynamics is developed and analyzed for solving the problem of time-varying matrix inversion. Neural dynamics are elegantly introduced by defining the matrix-valued error-monitoring function rather than the usual scalar-valued cost function such that the computation error can be made decreasing to zero globally and asymptotically. As noted, nonlinearity and errors always exist. Even if a linear activation function is used, the nonlinear phenomenon may appear in its hardware implementation. For superior convergence and better robustness, different kinds of activation functions (linear, sigmoid, power functions, and/or their variants, e.g., power-sigmoid function) are investigated. Theoretical and simulation results both demonstrate the efficacy of the proposed neural approach. To the best of our knowledge, there is little work dealing with such a problem in the literature at present stage, except some preliminary results presented in [21].

IV. PROPOSED METHOD

Instead of the standard sigmoid function, ABFNN uses a variable sigmoid function defined as:

$$O_f = \frac{a + \tanh(x)}{1 + a} \quad (1)$$

bipolar sigmoid function in the hidden-layer and one unipolar sigmoid function in the output-layer, with the following relations :

The two related metrics allows us to go between and . In particular, any computation relating to a metric on can be equivalently formulated using its corresponding -invariant metric on Bipolar sigmoid functions Unipolar sigmoid

functions

$$f_1(x) = \frac{k_1(1-e^{-x})}{(1+e^{-x})} \quad (2)$$

$$f_2(x) = \frac{k_2}{(1+e^{-x})} \quad (3)$$

The heterogeneity of some image sets; we partition the graph into sub graphs in order to compute multiple atlases to represent the image sets. Spectral clustering and related methods provide readily available algorithms for partitioning the graph. Popular and well-known algorithms include Ratio Cut [28] and Normalized Cut [29]. The algorithm partitions the graph using the graph Laplacian matrix that is computed from the similarity matrix that encodes all the metrical information provided by the nodes where and are nearest neighborhoods. The parameter is empirically estimated by where is the th nearest neighbor of. $\boldsymbol{\sigma}[\bullet]$is composed of sigmoid functions, such as $\sigma(n) = (e^n - 1)/(e^n + 1)$ evaluated at all input-to-node variables, ni, with i = 1, …, s,

$$\boldsymbol{\sigma}[\mathbf{n}] \equiv [\sigma(n_1) \square \sigma(n_s)]^T \quad (4)$$

and:

$$\mathbf{n} = \mathbf{W}\mathbf{p} + \mathbf{d} = [n_1 \square n_s]^T \quad (5)$$

The order of differentiability of eq. 2 is the same as that of the activation function, . Given that the chosen sigmoid functions are infinitely differentiable, the derivative of the network output with respect to its inputs is:

$$\frac{\partial z}{\partial p_j} = \sum_s \frac{\partial z}{\partial n_i} \frac{\partial n_i}{\partial p_j} = \sum_s v \sigma'(n) w_{ij} \quad (6)$$

$$i=1 \qquad i=1 \qquad i$$

This is equivalent to stating that the neural adjustable parameters must satisfy the following nonlinear equations,

$$u^k = \mathbf{v}^T \boldsymbol{\sigma}[\mathbf{W}\mathbf{y}^k + \mathbf{d}] + b \quad (7)$$

Hence, the adjustable parameters must satisfy the following gradient weight equations,

$$\mathbf{c}^k = \mathbf{W}_e^T \{\mathbf{v} \otimes \boldsymbol{\sigma}'[\mathbf{n}^k]\} \quad (8)$$

Input-to-node weight equations are obtained from the arguments of the nonlinear sigmoidal functions in above eqn

$$\mathbf{n}^k = \mathbf{W}\mathbf{y}^k + \mathbf{d} \quad (9)$$

Exact matching of the function's derivatives is achieved when the neural network's gradient evaluated at the input yk equals ck,

$$\left.\frac{\partial z}{\partial p_j}\right|_{\mathbf{y}^k} = c_j^k, \quad (10)$$

Training a neural network model essentially means selecting one model from the set of allowed models (or, in a Bayesian framework, determining a distribution over the set of allowed models) that minimizes the cost criterion. There are numerous algorithms available for training neural network models; most of them can be viewed as a straightforward application of optimization theory and statistical estimation. Most of the algorithms used in training artificial neural networks employ some form of gradient descent.

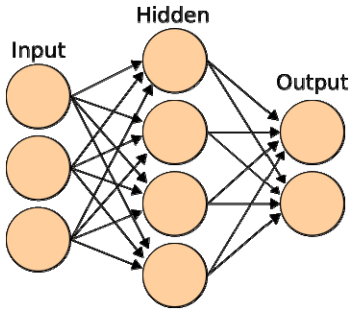

Fig 5 Artificial Neural network

This is done by simply taking the derivative of the cost function with respect to the network parameters and then changing those parameters in a gradient-related direction. Evolutionary methods,[37] gene expression programming,[38] simulated annealing,[39] expectation-maximization, non-parametric methods and particle swarm optimization[40] are some commonly used methods for training neural networks.

V. EXPERIMENTAL RESULT

Experiments have shown effectiveness proposed in completely automatic method in formation documents and categorization that is based in the applied clustering algorithm for full texts . Software implementation of the method designed for digital libraries as an element of their search engines. Such an element capable of be as independent search mechanism and serve as a means to improve the quality of other search engines , for example, search by keyword.

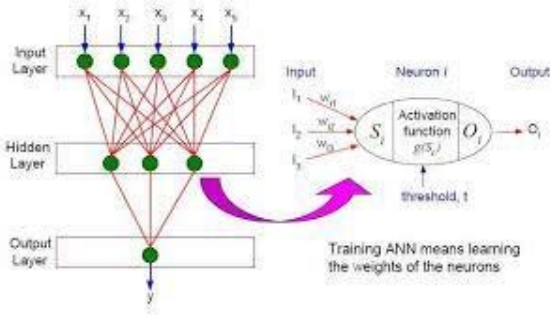

Fig 6 Sigmoidal Activating process

In this method, the obtained experimental results are evaluated through evaluation metrics namely, sensitivity, specificity and accuracy. In order to find these metrics, we first compute some of the terms like, True positive, True negative, False negative and False positive based.

It also proposed method In turn, the sigmoidal function is one category of functions used as the activation function (aka "squashing function") in FF neural networks solved using back prop. During training or prediction, the weighted sum of the inputs (for a given layer, one layer at a time) is passed in as an argument to the activation function which returns the output for

that layer. Another group of functions apparently used as the activation function is piecewise linear function.

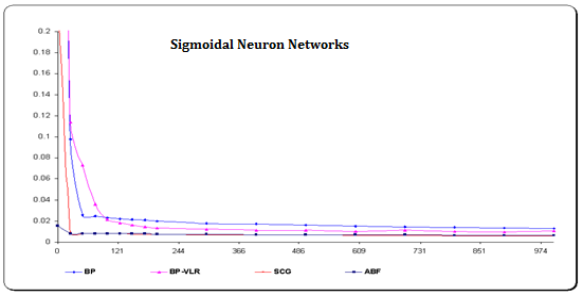

Fig 7 Sigmoidal Neuron Networks

VI. DISCUSSION AND CONCLUSION

This paper has presented the design and implementation of a neuron that will be used in any neural network, the activation function that designed inside the neuron is a sigmoid function. This article has covered the main points you need to know when designing EAs using neural networks. It has shown us the structure of a neuron and neural network architecture, outlined the activation functions and the methods for changing the activation function shape, as well as the process of optimization and input data normalization. Furthermore, we have compared an EA based on the standard logic with a neural network driven EA This design was performed using schematic editor tools in a reconfigurable device (FPGA) program. The design of the neuron will be as a micro neuron. Therefore, the user can use any numbers of this neuron by just make drag to the neuron component from the library of a schematic editor.

The current methodology is based on Sigmoid neural network has been proposed, in which the continuous input space is localized into some regions by using RBF (Gaussian) units where the location on the input space is represented by only one parameter. It can be said that this network has both advantages of RBF-based network and sigmoid-based multi-layer neural network

VII. REFERENCES

[1] Hecht-Nielsen, R. (1989, June). Theory of the backpropagation ne

[2] ural network. In Neural Networks,1989. IJCNN., International Joint Conference on (pp. 593-605). IEEE.

[3] Zhang, Q., & Benveniste, A. (1992). Wavelet networks. Neural Networks, IEEE Transactions on, 3(6), 889-898.

[4] Barron, A. R. (1993). Universal approximation bounds for superpositions of a sigmoidal function. Information Theory, IEEE Transactions on, 39(3), 930-945.

[5] Park, J. H., Kim, Y. S., Eom, I. K., & Lee, K. Y. (1993). Economic load dispatch for piecewise quadratic cost function using Hopfield neural network. Power Systems, IEEE Transactions on, 8(3), 1030-1038.

[6] Chen, T., & Chen, H. (1995). Approximation capability to functions of several variables, nonlinear functionals, and operators by radial basis function neural networks. Neural Networks, IEEE Transactions on, 6(4), 904-910.

[7] Kosmatopoulos, E. B., Polycarpou, M. M., Christodoulou, M. A., & Ioannou, P. A. (1995). High-order neural network structures for identification of dynamical systems. Neural Networks, IEEE Transactions on, 6(2), 422-431.

[8] Wang, L. X., & Mendel, J. M. (1992). Fuzzy basis functions, universal approximation, and orthogonal least-squares learning. Neural Networks, IEEE Transactions on, 3(5), 807-814.

[9] Chen, T., & Chen, H. (1995). Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems. Neural Networks, IEEE Transactions on, 6(4), 911-917.

[10] Cohen, M. A., & Grossberg, S. (1983). Absolute stability of global pattern formation and parallel memory storage by competitive neural networks. Systems, Man and Cybernetics, IEEE Transactions on, (5), 815-826.

[11] Lee, K. Y., Sode-Yome, A., & Park, J. H. (1998). Adaptive Hopfield neural networks for economic load dispatch. Power Systems, IEEE Transactions on, 13(2), 519-526.

[12] Park, D. C., El-Sharkawi, M. A., Marks, R. J., Atlas, L. E., & Damborg, M. J. (1991). Electric load forecasting using an artificial neural network. Power Systems, IEEE Transactions on, 6(2), 442-449.

[13] Hunt, K. J., & Sbarbaro, D. (1991, September). Neural networks for nonlinear internal model control. In Control Theory and Applications, IEE Proceedings D (Vol. 138, No. 5, pp. 431-438). IET.

[14] Chen, S., Cowan, C. F. N., & Grant, P. M. (1991). Orthogonal least squares learning algorithm for radial basis function networks. Neural Networks, IEEE Transactions on, 2(2), 302-309.

[15] Jones, L. K. (1990). Constructive approximations for neural networks by sigmoidal functions. Proceedings of the IEEE, 78(10), 1586-1589.

[16] Tamura, S. I., & Tateishi, M. (1997). Capabilities of a four-layered feedforward neural network: four layers versus three. Neural Networks, IEEE Transactions on, 8(2), 251-255.