TWO-DIMENSIONAL FEED-FORWARD FUNCTIONALLY EXPANDED NEURAL NETWORK

Spyros PANAGOPOULOS [1] *& John J. SORAGHAN* [2]

Signal Processing Division, Department of Electronic & Electrical Engineering,
University of Strathclyde, Royal College Building, Glasgow, G1 1XW, Scotland, UK
E-mail: [1]spyros@spd.eee.strath.ac.uk , [2]j.soraghan@eee.strath.ac.uk

ABSTRACT

This paper is concerned with the development of a two-dimensional feed-forward functionally expanded neural network (2D FFENN) surface modeler. New nonlinear surface basis functions are proposed for the network's functional expansion. A network optimization technique based on an iterative function selection strategy is also described**.** Comparative simulation results for surface mappings generated by the 2D FFENN, Multi-layered Perceptron (MLP) and Radial Basis Function (RBF) architectures are presented.

I. INTRODUCTION

One of the main properties of feed-forward neural networks is that of learning an input-output mapping from a set of examples characterizing a real system. The network is trained with some examples comprising an input signal and the desired response. The network weights are then modified, using an adaptive optimization technique to minimize the difference between the desired response and actual response.

Two well-known feed-forward artificial neural networks are the MLP and RBF. Both networks have been termed as *universal approximators* [1], [2]. That implies that if the nonlinear transformations are of sigmoidal or radial symmetric type i.e. the non-linearity is a non-constant, bounded and monotone increasing function, and the number of nodes in the hidden layer goes to infinity, then based on Universal Approximation Theorem [3], one may provide an approximate realization of any continuous function. Their performance has been demonstrated in various application areas such as, linear and nonlinear adaptive filtering [4], time series prediction [5], dynamic reconstruction [6] and black box modeling [7]. However, these networks suffer from a number of drawbacks, such as convergence characteristics and network topology selection [3].

MLP networks employ sigmoidal basis functions that cannot model local non-linearity very well. Also, their nonlinear in-the-parameters structure requires complex and computationally intense learning algorithms, such as the back-propagation algorithm. Furthermore, there is no way to say whether a single hidden layer is optimum to support the MLP network learning or a way to specify the exact number of hidden neurons required in order a system to be generalizable. On the other hand, RBF networks that employ radial symmetric functions, cover only small-localized regions and therefore they cannot model well global non-linearity. Moreover, dealing with RBF networks great difficulty is experienced in selecting the appropriate centers for the radial basis functional expansion. Additionally, a large number of basis functions is usually required in order to cover high dimensional input spaces. Nonetheless, simple learning algorithms may be used for training, as the RBF structure is linear in the parameters.

In this paper the design of a new single hidden layer, linear in the parameters, feed-forward functionally expanded neural network surface modeler (2D FFENN) is presented. Previously, a 1D FFENN has been successfully applied to time series prediction [8], [9] and co-channel interference [10]. The aim of this new design is to explore the modeling capabilities of such a feed-forward network in two dimensions. The main objective is to approximate a nonlinear continuous surface to an arbitrary degree of accuracy. As its predecessor 1D FFENN design, the design of 2D FFENN can be considered as a hybrid neural network. In essence, is an extended model that incorporates the modeling capabilities of the existent architectures of MLP, RBF and volterra neural networks [3], VNN.

The paper is organized as follows. In section II an overview of the complete 2D FFENN is given. Section III describes the characteristics of a function pruning technique, which is developed to optimize the network's functional expansion. In section IV, a number of representative simulation results are presented that illustrate the surface modeling capabilities of the network and the results are compared with the well-known architectures of MLP and RBF networks. We conclude the paper with section V.

II. THE 2D FFENN STRUCTURE

The topology of the 2D FFENN structure is depicted in Figure 1. Two layers describe it: a single hidden layer and an output layer. The hidden layer acts like a feature detection layer. As the learning process progresses, the hidden neurons begin to gradually discover the salient features that characterize the training data. This is achieved by the functional expansion unit, which performs a nonlinear transformation of the input data into a new space called the feature space. The output layer of the network comprises a set of linear combiners that join together all the weighted functionally expanded inputs to form a single output.

The functional expansion unit of 2D FFENN takes two inputs, $\bar{t}_1$ and $\bar{t}_2$, which are the grid indices that specify the two-dimensional data set to be modeled. Both inputs are normalized to within the range (+1, -1).

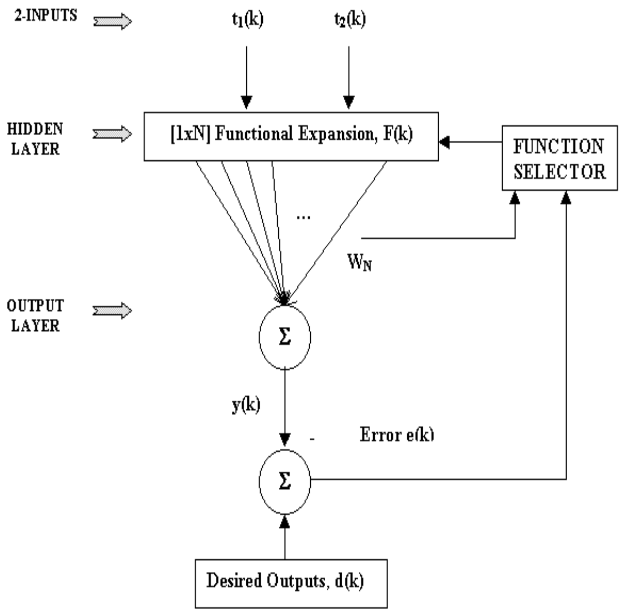

Figure 1: The 2D feed-forward functionally expanded neural network (2D FFENN) structure

The entire functional expansion is described by *F(k)*, as follows:

F(k) = sum of N (linear & nonlinear) basis functions

In a similar fashion to the functions described in [11], the linear terms of the expansion are the original input terms, whilst the nonlinear terms are a combination of *trigonometric* and *polynomial* two-dimensional functions of the input.

The modeling efficiency of the 2D FFENN is the result of this hybrid functional expansion. These functions have been chosen in such a way that combine the global approximation capability of the MLP network, the local approximation capability of the RBF network and also emulate the modeling capability of the VNN.

In general, a Multi-Input Multi-Output (MIMO) FFENN (*n*, *N*, *m*) will completely be specified for a given number of n inputs and m outputs by a similar *F(k)* expansion.

The output for the two layer, two input and N-term functionally expanded FFENN (*2*, *N*, *1*) is defined as follows:

1. Hidden layer functional expansion vector at time, k:
$$F(k) = [f_1(k), f_2(k), \ldots\ldots, \; f_N(k)]^T \qquad (1)$$
and the associated weight vector, as:
$$W(k) = [w_1(k), w_2(k), \ldots\ldots, \; w_N(k)]^T \qquad (2)$$
2. Single 2D FFENN output:
$$y(k) = F^T(k) \cdot W(k) \qquad (3)$$
3. Prediction error:
$$e(k) = d(k) - y(k) \qquad (4)$$
where *d(k)* is the reference response.

Network weight adaptation is achieved using the exponentially recursive least squares (RLS) algorithm [12]. Complex training algorithms are not required, because of the linear in-the-parameters network structure.

The function selection unit shown in Figure 1 is used in order to reduce the size of the functionally expanded network. The scope of the functional expander is to introduce to the network new functional terms in order to enhance its nonlinear approximation ability. However, this process can lead the expansion to very large and highly redundant networks. For this reason, a pruning or function selection scheme is occupied to choose only the most significant functions.

III. PRUNING OF THE FULLY EXPANDED 2D FFENN

A large functional expansion can achieve better prediction results. Nevertheless, depending on the type and level of complexity of the surface to be modeled the network may assume too many free parameters. This is because a much smaller number of functions are probably needed to characterize the specific function or surface. For this reason, a pruning scheme is utilized. Its task is to select only those functions, which have a significant contribution to the output of the network. In other words, we want to choose only the dominant weights of the functional expansion.

Pruning is performed by an iterative pruning-retraining approach. Initially, the fully expanded network structure is trained on the training data set and the maximum surface level error (MSLE) value on the training set is computed. In mathematical terms, the MSLE (maximum surface level error) is defined as follows:

$$MSLE = abs\left(\max\left[e(k)\right] - \min\left[e(k)\right]\right) \qquad (5)$$

The insignificant functions in the expansion model are set with the smallest weights; these are successively pruned one by one starting with the least significant one. After each insignificant function is been pruned the output of the network is computed. Moreover, the resulting MSLE is also computed at each pruning stage. The pruning process is stopped at the stage when a pruned network structure is found to be incapable of reducing the output MSLE on the training set to the desired level. The network structure can be retrained after each time pruning is applied, with the same train set, in order to determine the optimal weights for the remaining un-pruned functions.

Furthermore, it is important to note that pruning is only an optimization strategy and can be omitted when there is no advantage to be gained. It effectively achieves to reduce the size of the functional expansion, but in the downside requires off-line training (supervised learning) and much longer computation time.

IV. SIMULATION RESULTS

In this section we present comparative simulation results of the new 2D FFENN with the existent MLP and RBF networks. In order to illustrate the modeling capability of the 2D FFENN and the effectiveness of the pruning strategy, let us produce a model for the continuous surface described by equation 6. The choice of this function is based on its surface characteristics that resemble the nature of a sea wave.

$$T\left(\bar{t}_1, \bar{t}_2\right) = \frac{0.1 + 1 + \sin\left(2 \cdot \bar{t}_1 + 3 \cdot \bar{t}_2\right)}{3.5 + \sin\left(\bar{t}_1 - \bar{t}_2\right)} \tag{6}$$

Figure 2 displays the surface characteristics of the two-dimensional function. The training set is constructed from the function $T\left(\bar{t}_1, \bar{t}_2\right)$ by sampling the domain from –1.0 to +1.0 in two dimensions at equally spaced grid points, at an interval of 0.1 for both $\bar{t}_1$ and $\bar{t}_2$. The 21 spacing indices generated for each one of the dimensions, correspond to the network inputs. In Figure 3 the mapped surface is presented. Modeling was achieved by the 2D FFENN using a functional expansion of 59 functions and batch training was performed for 5 times. Pruning has not been considered, so at this stage all 59 functions contribute to the model. The training error is shown in Figure 4. Based on the results, we can conclude that the 2D FFENN is able to produce considerably good surface mappings even after trained for a small number of epochs.

In order to test the efficiency of the pruning strategy employed by the 2D FFENN let us simulate for the same surface, but this time using the MSLE to choose the most appropriate functions for modeling. The results are shown in Figure 5. From the results is evident that pruning can effectively reduce the dimensionality of the functional expansion. In this example, 13 functions have been discarded. The size of pruning achieved mainly depends on the specific characteristics of the surface to be predicted and the number of training epochs. The algorithm allows the user to retrain also after each time pruning is performed. This leads to better weight adaptation achieving a smaller mapping error. However, in the downside the number of candidate-pruned functions is expected to be reduced. In general, the minimum the network training performed the maximum the pruning that can be achieved. Obviously this trade-off is directly reflected to the modeling accuracy requirement.

Next, in Figure 6, the MLP network training error is given. From the results, it appears to be inferior to the 2D FFENN. It requires 500 epochs under supervised batch training and approximately a network expansion of 15 hidden neurons to produce good but not better results than the 2D FFENN.

On the other hand, the RBF network can compete the performance of the 2D FFENN at the expense of a large hidden neurons expansion. The associated training error for an RBF network of 42 hidden neurons is shown in Figure 7.

Finally, in order to validate the performance of the 2D FFENN, MLP and RBF networks, a test data set was constructed using grid spacing of 0.03195, a value chosen to avoid replication of training set points, forcing the network to interpolate. All results are shown in Table 1.

Network	Training MSLE	Validation MSLE
2D FFENN, No Pruning	0.00259	0.00221
2D FFENN, MSLE Pruning	0.00045	0.00042
MLP	0.00785	0.00722
RBF	0.00104	0.00093

Table 1: Network Training and Validation Errors

V. CONCLUSION

In this paper a two-dimensional functionally expanded neural network was presented. The network's backbone architecture was described and comparative simulation results were given. An efficient function pruning strategy was also devised. The comparative results obtained by the proposed system demonstrate the effectiveness of such a network structure to produce surface mappings under short training times.

We are currently extending the proposed network design to a multi-scaled functionally expanded structure in order to enhance its nonlinear modeling ability for surfaces were discontinuities are present. Our future development plan includes the extension of 2D FFENN to a three-dimensional volume modeler, 3D FFENN, which lies on straightforward modifications to the existent system. The main application area of interest for the proposed system is target detection from two-dimensional data sets. For example, target detection in sea by sea-clutter suppression.

ACKNOWLEDGMENT

This work was supported in part by QinetiQ, UK. The authors wish to thank especially Dr. A. Brown for his valuable comments.

REFERENCES

[1] D.R.Hush and B.G. Horne, "Progress in Supervised Neural Networks: What's new since Lippman," IEEE Signal Processing Magazine, pp.25-49, March 1993.

[2] J. Park, and I.W.Sandberg. "Universal approximation using radial-basis-function networks", Neural Computation, vol.3, pp.246-257, 1991.

[3] S. Haykin, "Neural Networks – A Comprehensive Foundation, 2nd edition, Chap. 4,5,15, pp. 208, 293 & 762.

[4] D. Low and A.R Webb, "Time series prediction by adaptive networks: A dynamical system prespective", IEE Proc.-F, Vol.138 , pp. 17-25, 1991.

[5] S. Haykin and J. Principe, "Making Sense of a Complex World", IEEE Signal Processing Magazine, Vol.15, No.3, pp 66-81.

[6] S. Haykin, S. Puthusserypady and P. Yee, "Dynamic Reconstruction of Sea Clutter using Regularized RBF Networks", in Proceedings of the 32nd Asilomar Conference on Signals, Systems & Computers, Pacific Grove, CA, Nov.1-4, 1998.

[7] S. Haykin and H. Leung, “Chaotic Model of Sea Clutter Using a Neural Network”, Proceedings of the SPIE Conference, San Diego, CA.1989.

[8] A. Hussain, J.J. Soraghan and T.S. Durrani, “A New Hybrid Neural Network Structure for Nonlinear Time Series Modeling”, Journal of Computational Intelligence in Finance, Special Issue on *Hybrid Neural Networks for Financial Time Series Forecasting*, Vol.5, No.1, Finance & Technology Publishing Inc., USA (ISSN: 1074-5637), pp.16-26, 1997

[9] A. Hussain, J.J. Soraghan, T.S. Durrani and D.R. Campbell, “A New Neural Network Structure For Modeling Nonlinear Dynamical Systems”, In: Proceedings IEEE International Workshop On Image And Signal Processing (Iwisp’96), *Advances In Computational Intelligence*, (Eds.) B.G.Mertzios And P.Liatsis, Elsevier Science Publ., Manchester, U.K, Pp.119-122, 4-7 November 1996.

[10] A. Hussain, J.J. Soraghan and T.S. Durrani, “A New Adaptive Functional-Link Neural Network Based DFE for Overcoming Co-Channel Interference”, IEEE Transactions on Communications, Vol.45, No.11, (ISSN: 0090-6778), pp.1358-1362, 1997.

[11] A. Hussain and J.J. Soraghan, FENN Methodology: “Improved Neural Network”, International Patent, UK Patent Number 9624298.7, November 1996.

[12] A. Hussain, “Novel Artificial Neural Network based Architectures and Algorithms for Nonlinear Dynamical Systen Modeling and Digital Communications Applications”, Ph.D Thesis, p.p 77-79.

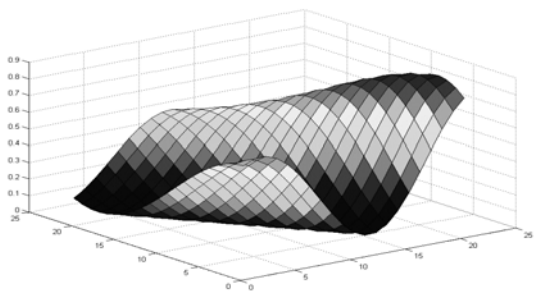

Figure 2: Original Surface, $T(\bar{t}_1,\bar{t}_2)$

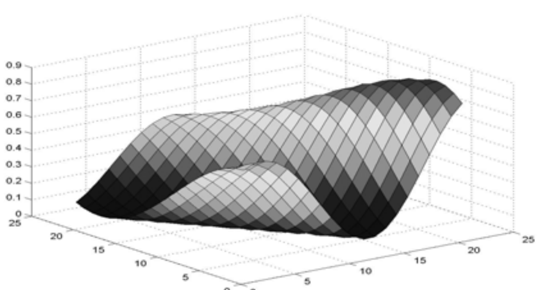

Figure 3: Modeled Surface by 2D FFENN

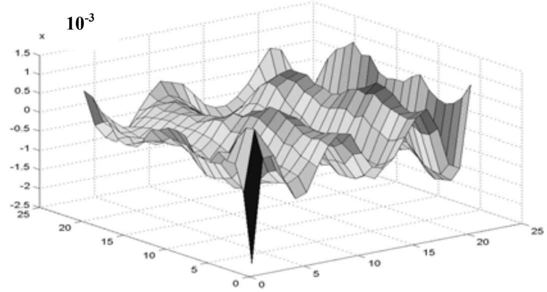

Figure 4: Error Surface – 2D FFENN, No Pruning, 59 Functions, 5-epochs

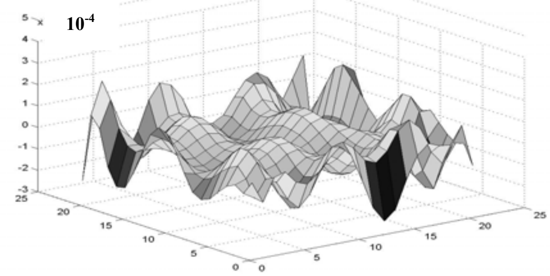

Figure 5: Error Surface – 2D FFENN, MSLE Pruning, 46 Functions, 5-epochs

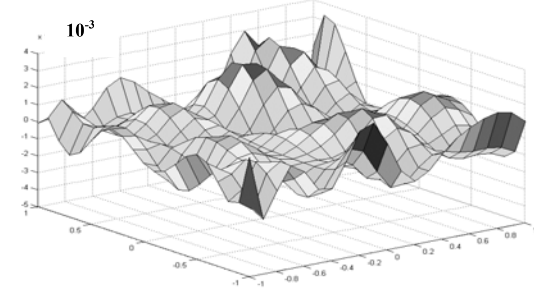

Figure 6: Error Surface – MLP, 500 epochs, 15 H. N

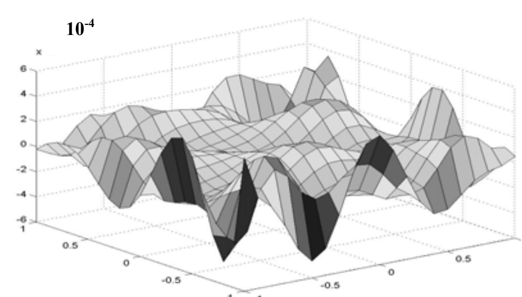

Figure 7: Error Surface – RBF, 42 Hidden Neurons