

A Functional Link Network With Ordered Basis Functions

Saurabh Sureka, *Student Member, IEEE*, Michael Manry, *Sr Member, IEEE*

Abstract—A procedure is presented for selecting and ordering the polynomial basis functions in the functional link net (FLN). This procedure, based upon a modified Gram Schmidt orthonormalization, eliminates linearly dependent and less useful basis functions at an early stage, reducing the possibility of combinatorial explosion. The number of passes through the training data is minimized through the use of correlations. A one-pass method is used for validation and network sizing. Function approximation and learning examples are presented. Results for the Ordered FLN are compared with those for the FLN, Group Method of Data Handling, and Multi-Layer Perceptron.

I. INTRODUCTION

Multilayer perceptron (MLP) neural nets[1] have proved useful in many approximation and classification applications because of their universal approximation properties [2,3], their relation to optimal approximations [4] and classifiers [5], and the existence of workable training algorithms such as backpropagation[1]. They have the unique property that their basis functions develop during training rather than being arbitrarily chosen ahead of time. However, the MLP has long training time, is sensitive to initial weight choices, and its training error may not converge to a global minimum. In addition, there is a wide gap between the MLP’s theoretical capabilities and its actual performance with currently available training algorithms.

In contrast, the functional link network[6] (FLN) has a flat architecture, with pre-defined basis functions like the trigonometric functions, algebraic polynomials, Chebyshev polynomials and Hermite polynomials. The global minimum of its error function can be found by solving linear equations [6] or by techniques such as genetic algorithms [7]. As with other Polynomial Neural Networks (PNN) faster learning rates are achieved [8] for a given network-size.

However, with an increase in degree of approximation and inputs there is a combinatorial explosion in the number of basis functions required during training. This is the biggest drawback in application of the FLN. Several other PNNs have been developed to remedy this including group method of data handling (GMDH) networks [9,10], in which many trial polynomial basis functions are generated, but only those which prove useful are kept. Pi-sigma networks [11] do not have the universal approximation property, but Ridged-

Polynomial network [12] may address that problem.

Orthonormal Least Squares (OLS) has been used by Chen et al. [13] to efficiently find subsets of center vectors in radial basis function (RBF) networks. Kamniski and Strumillo [14] used modified Gram Schmidt orthonormalization (MGSO) to compute hidden weights in RBF networks.

In this work, we use MGSO to orthonormalize sets of FLN basis functions and methodically order them according to their contribution to the decrease in the mean square error (MSE). The network thus obtained is called the Ordered FLN (OFLN). In Section 2 the FLN is briefly reviewed. The notation and structure of the OFLN is introduced in section 3. Section 4 summarizes the MGSO training procedure for the OFLN. Numerical results in section 5 compare the FLN, GMDH, MLP and OFLN.

II. FUNCTIONAL LINK NETWORK

FLNs often have a fixed number of polynomial or trigonometric basis functions. A 2nd degree FLN with 2 inputs and 1 output is shown in Fig. 1.

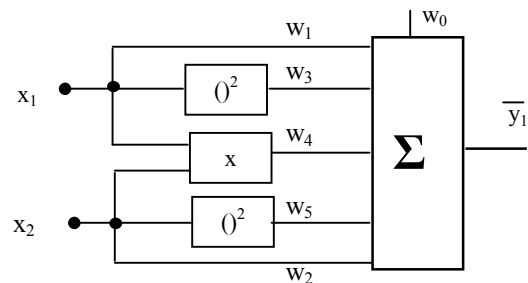


Fig. 1. A FLN with 2 inputs and 1 output.

This two-input unit can be scaled to a single-layer network mapping $R^N \rightarrow R^M$ where N and M respectively denote the numbers of inputs and outputs. Alternately, several of these two-input units can be used in a multi-layered network such as the GMDH. In either case, the resulting network approximates the Kolmogorov-Gabor or Ivakhnenko polynomial [10,15]

$$\bar{y}_k = \alpha_{k0} + \sum_{i=1}^N \alpha_{ki} \cdot x_i + \sum_{i=1}^N \sum_{j=1}^N \alpha_{kij} \cdot x_i \cdot x_j + \sum_{i=1}^N \sum_{j=1}^N \sum_{l=1}^N \alpha_{kijl} \cdot x_i \cdot x_j \cdot x_l + \dots \quad 1 \leq k \leq M \quad (1)$$

where \bar{y}_k is the k^{th} estimated output and α_k 's are the corresponding coefficients. Let \mathbf{X} be a column vector of

S Sureka is a Masters student in Electrical Engineering Department, University of Texas at Arlington, TX 76010, USA (e-mail: saurabh@uta.edu).

M. Manry is with the Electrical Engineering Department, University of Texas at Arlington, TX 76010, USA (e-mail: manry@uta.edu).

basis functions, then the i^{th} polynomial basis function (PBF) element X_i is an element of the monomial set $\{1, x_i, x_i \cdot x_j, x_i \cdot x_j \cdot x_k, \dots\}$ for $1 \leq i \leq j \leq k \dots \leq N$. Now, (1) is rewritten as

$$\bar{y}_k = \sum_{i=0}^{L-1} w_{ki} \cdot X_i \quad 1 \leq k \leq M \quad (2)$$

where w_{ki} connects X_i to the k^{th} output \bar{y}_k . For an FLN that approximates a function with maximum degree monomial D , the total number of basis functions is given by

$$L = \binom{N+D}{D} \quad (3)$$

Assume that the N -dimensional input vector \mathbf{x} has mean vector \mathbf{m} and a vector $\boldsymbol{\sigma}$ of standard deviations. We normalize \mathbf{x} as

$$x_{np} \leftarrow (x_{np} - m_n) / \sigma_n \quad 1 \leq n \leq N, 1 \leq p \leq N_v \quad (4)$$

where N_v is the number of training patterns. The bias term x_0 as in (1) is defined to be 1. For the p^{th} pattern, let y_{kp} be the k^{th} observed or desired output and let X_{ip} denote the p^{th} value of X_i . The error function to be minimized in FLN training is

$$\mathcal{E} = \sum_{k=1}^M \mathcal{E}_k = \sum_{k=1}^M \frac{1}{N_v} \sum_{p=1}^{N_v} \left[y_{kp} - \sum_{i=0}^{L-1} w_{ki} \cdot X_{ip} \right]^2 \quad (5)$$

Minimization of (5) can be reduced to solving M sets of L linear equations in L unknowns.

Clearly the problem of combinatorial explosion with an increase in D and N is evident from (3), as an example with 5 inputs and 2^{nd} degree approximation, L is 21 and for 25 inputs and 4^{th} degree approximation, L is 23751. This makes it impractical to design and apply high degree FLNs.

III. ALGORITHM APPROACH AND NOTATION

In order to prevent combinatorial explosion in the FLN, we need to limit the value of L . One approach is to limit the FLN's degree D . However, this limits its ability to model complicated functions. A better approach is to use only the most useful basis functions. If the elements of \mathbf{X} are to be in descending order of usefulness, a method is needed for generating these elements efficiently, in any possible order.

Consider an L by $(D+1)$ position matrix \mathbf{K} , whose i^{th} row specifies how to generate the PBF X_i . For element $K(i,j)$, the ranges of i and j are $0 \leq i \leq L-1$ and $0 \leq j \leq D$. The i^{th} PBF, with $K(i,D)$ denoting its degree, is defined as

$$X_i = \prod_{j=1}^{K(i,D)} x_{K(i,j-1)} \quad 1 \leq i \leq L-1 \quad (6)$$

The first basis function as in (2) is fixed as $X_0 = 1$. Thus \mathbf{X} can be generated from \mathbf{K} and the normalized input vector \mathbf{x} . An orthonormalized representation of \mathbf{X} is \mathbf{X}^o , which is to be generated by the transformation

$$\mathbf{X}^o = \mathbf{A} \cdot \mathbf{X} \quad (7)$$

where \mathbf{A} is a lower triangular L by L transformation matrix

$$\mathbf{A} = \begin{bmatrix} a_{00} & 0 & 0 & 0 \\ a_{10} & a_{11} & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot \\ a_{(L-1)0} & a_{(L-1)1} & \cdot & a_{(L-1)(L-1)} \end{bmatrix} \quad (8)$$

Given \mathbf{X} , as generated by an initial \mathbf{K} matrix, let elements of the array \mathbf{J} index basis functions according to their usefulness. If $J_0 = 3$ and $J_3 = 8$, for example, the 1^{st} and 4^{th} most useful basis functions are X_3 and X_8 respectively.

Our general approach for training the OFLN is to iteratively generate \mathbf{K} and \mathbf{J} for higher and higher degrees, D , finding the basis function coefficients each time via MGSO. In each iteration, redundant and useless basis functions are eliminated, in order to prevent combinatorial explosion in the subsequent iterations.

Relevant correlations for MGSO are the auto and cross-correlation functions defined as

$$r_{ij} = \frac{1}{N_v} \sum_{p=1}^{N_v} X_{ip} \cdot X_{jp} \quad 0 \leq i \leq L-1, 0 \leq j \leq L-1 \quad (9)$$

$$c_{ij} = \frac{1}{N_v} \sum_{p=1}^{N_v} y_{ip} \cdot X_{jp} \quad 1 \leq i \leq M, 0 \leq j \leq L-1 \quad (10)$$

IV. MODIFIED GRAM SCHMIDT ORTHONORMALIZATION PROCEDURE

The MGSO algorithm orthonormalizes a set of linearly independent functions or vectors in inner product space. It is fast and numerically stable and has been used in Least Square Estimation problems[16], optimal MLP pruning [17], and in efficient feature selection using piecewise linear networks (PLN)[18].

In this section we use MGSO to simultaneously orthonormalize PBFs and solve linear equations for the OFLN's weights. Rewriting (2) in terms of p^{th} training pattern of the i^{th} orthonormal vector X_i^o ,

$$\bar{y}_{kp} = \sum_{i=0}^{L-1} w_{ki}^o \cdot X_{ip}^o \quad 1 \leq k \leq M \quad 1 \leq p \leq N_v \quad (11)$$

where w_{ki}^o is the transformed weight corresponding to the k^{th} output \bar{y}_k and i^{th} orthonormal PBF X_i^o .

A. Degree D Up to One

In the first OFLN training iteration, we order the inputs, discarding those that are linearly dependent or less useful for estimating outputs. Here, $L = N+1$ and \mathbf{K} and \mathbf{J} are initialized as

$$\begin{aligned} K(i,0) &= i & K(i,1) &= 1 & 0 \leq i \leq N \\ J_i &= i & & & 0 \leq i \leq N \end{aligned} \quad (12)$$

Following the basis function definition in (6), (12) indicates that the maximum degree is $D=1$, the first basis function is the constant 1, and the remaining basis functions are inputs. Our goal is to find the vector \mathbf{J} whose elements are indices of linearly independent inputs, in order of their contribution to reduce the MSE. The m^{th} orthonormal basis function X_m^o

from (7) and (8) is given by

$$X^o_m = \sum_{i=0}^m a_{mi} \cdot X_{J_i} \quad (13)$$

Then for $m=0$,

$$X^o_0 = a_{00} \cdot X_{J_0} \quad (14)$$

$$a_{00} = \frac{1}{\|X_{J_0}\|} = \frac{1}{r_{J_0 J_0}} \quad (15)$$

For $1 \leq m \leq L-1$, perform the following operations

$$\beta_i = \sum_{q=0}^i a_{iq} \cdot r_{J_q J_m} \quad 0 \leq i \leq m-1 \quad (16)$$

$$\gamma_m = 1$$

$$\gamma_k = -\sum_{i=k}^{m-1} \beta_i \cdot a_{ik} \quad 0 \leq k \leq m-1 \quad (17)$$

$$a_{mk} = \frac{\gamma_k}{\left[r_{J_m J_m} - \sum_{i=0}^{m-1} \beta_i^2 \right]^{\frac{1}{2}}} \quad 1 \leq k \leq m \quad (18)$$

$$w^o_{km} = \sum_{i=0}^m a_{mi} \cdot c_{kJ_i} \quad 1 \leq k \leq M \quad (19)$$

It can be easily proved that for values of m , $a_{mn} \rightarrow \infty$, X^o_m is linearly dependent on previously generated PBFs.

Lemma 1: If any input x_m is linearly dependent on other inputs then higher order basis functions that include x_m can be expressed using basis functions of the same degree that do not include x_m .

Proof: For any input x_m that is linearly dependent on other inputs there exists at least one non-zero λ_i such that

$$x_m = \sum_{i=1, i \neq m}^N \lambda_i \cdot x_i \quad (20)$$

Consider a degree D basis function with dependent input x_m raised to the d^{th} power. We have

$$x_m^d \prod_{n=1, k(n) \neq m}^{D-d} x_{k(n)} = \left(\sum_{i=1, i \neq m}^N \lambda_i \cdot x_i \right)^d \prod_{n=1, k(n) \neq m}^{D-d} x_{k(n)} \quad (21)$$

In (21), the right hand side has no x_m and the degree D is also unchanged. Thus each m^{th} linearly dependent function can be eliminated as

$$J_i = J_{i+1} \quad \text{for } m \leq i < L-1, L \leftarrow L-1 \quad (22)$$

The corresponding MSE for the orthonormal system is given by

$$\mathcal{E}^o_k = \frac{1}{N_v} \sum_{p=1}^{N_v} \left[y_{kp} - \sum_{i=0}^{L-1} w^o_{ki} \cdot X^o_{ip} \right]^2 \quad 1 \leq k \leq M \quad (23)$$

The total system error \mathcal{E} can be solved as

$$\mathcal{E} = \sum_{k=1}^M \mathcal{E}^o_k = \sum_{k=1}^M E[y_k^2] - \sum_{k=1}^M \sum_{i=0}^{L-1} [w^o_{ki}]^2 \quad (24)$$

where the expectation operator is given by

$$E[y_k^2] = \frac{1}{N_v} \sum_{p=1}^{N_v} [y_{kp} \cdot y_{kp}] \quad (25)$$

Equation (24) measures a basis function's contribution to reduce the MSE. Denote the second term in (24) by P_i associated with i^{th} orthonormal basis function X^o_i

$$P_i = \sum_{k=1}^M [w^o_{ki}]^2 \quad (26)$$

The desired new order of basis functions \mathbf{J} that reduces the MSE is thus obtained by maximizing P_i so that

$$P_{J_0} \geq P_{J_1} \geq \dots \geq P_{J_{L-2}} \geq P_{J_{L-1}} \quad (27)$$

The orthonormal weights are transformed back to original weights as

$$w_{ki} = \sum_{j=0}^{L-1} w^o_{kj} \cdot a_{ij} \quad 0 \leq i \leq L-1, 1 \leq k \leq M \quad (28)$$

At this point, \mathbf{J} gives us the order of the useful basis functions for a linear network. If only a first order approximation is required, then a reordered \mathbf{K} based on \mathbf{J} and weights as in (28) could be saved and they represent the OFLN of degree 1.

B. Degree D Greater Than One

As the network grows iteratively for each degree up to the desired D , the rows of \mathbf{K} are reordered based on \mathbf{J} that constitute only essential PBFs. Following lemma 1, rows of \mathbf{K} describing new candidate basis functions are generated by unique combinations of existing rows of \mathbf{K} and appended to it. Equations (14)-(19), (22)-(24), (26)-(27) are repeated with the value of L being the row count of \mathbf{K} for each degree. As a control or stopping criterion, we can stop at a given maximum number of PBF's L_{max} . Alternately, we can stop when the percentage change in error for adding a PBF is less than a user-chosen value $\Delta\mathcal{E}$. The MSE is a monotonically non-increasing function for increase in L as seen from (24).

C. One-Pass Validation

As with other nonlinear networks, there are no practical optimal approaches for determining network size. The user can pick a high but practical degree D , and find the number of basis functions that minimizes validation error.

However, using MGSO, a one-pass measurement of validation error for network size up to L is possible. The validation dataset is normalized with known values of \mathbf{m} and $\boldsymbol{\sigma}$. \mathbf{X} , \mathbf{X}^o are generated using (6) and (13) respectively. Then, for a network of size k with N_{vt} validation patterns, the total validation MSE (\mathcal{E}_{vk}) is given by

$$\mathcal{E}_{vk} = \frac{1}{N_{vt}} \sum_{p=1}^{N_{vt}} \left[\sum_{i=1}^M \left(y_{ip} - \sum_{j=0}^{k-1} w^o_{ij} \cdot X^o_{jp} \right)^2 \right] \quad 1 \leq k \leq L \quad (29)$$

Given the pattern number p , the quantity in the inner brackets is evaluated for all values of k . Hence \mathcal{E}_{vk} is

updated for all values of k in a single pass through the data.

The MGSO approach of this section reduces the OFLN's computational load in two ways. First, it allows us to greatly reduce the number of basis functions used in the FLN, speeding up the design procedure, reducing the computational load of the network. Second, it leads to a one-pass validation of OFLNs with many different values of L .

V. COMPARISON AND SIMULATION RESULTS

A. Function approximation

Two examples are given to show the function approximation capabilities of the OFLN. In the first example, 10 data values in the interval $[0:1]$ are used for learning the sine function (Fig. 2) [11]. The OFLN exhibits small error, as expected.

In the second example, a modified rastrigin function (given by $20+x_1^2+x_2^2-10(\cos(0.1\pi x_1)+\cos(0.1\pi x_2))$) is approximated by the OFLN (Fig. 3). Fig. 4 shows an important advantage of the OFLN over FLN, sigma-pi and GMDH, i.e. as percentage change in MSE is zero then minimal number of 4 basis functions of up to degree 4 are sufficient for approximation with $MSE = 0.08$. Also, from (3), for $N=2$, $D=4$, L would be 15 but 6 of them are linearly dependent and eliminated during training. Hence, unlike GMDH and sigma-pi networks there are no repeated and redundant terms for higher degree representation. Results show that this property scales extremely well for systems with large numbers of inputs and outputs.

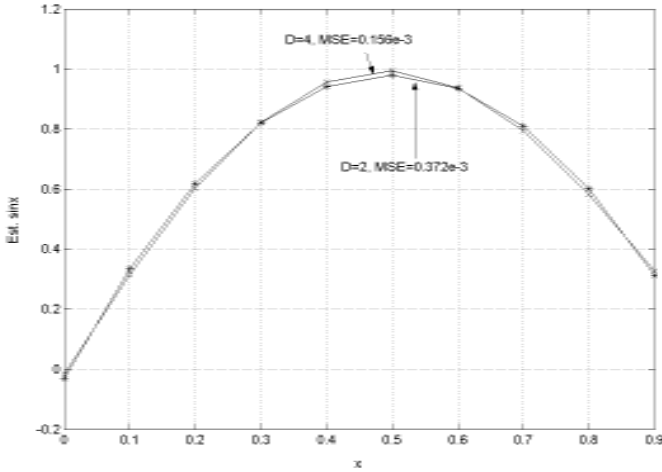


Fig. 2. $\sin(x)$ function approximation by OFLN for $D = 2$ and $D = 4$.

B. Supervised learning examples

Here, some examples for supervised learning are demonstrated for the OFLN. For comparison, a GMDH network [19,20] was designed using the Forward Prediction Error (FPE) criterion. In examples 1 and 2, an MLP was trained with back-propagation and the Levenberg-Marquardt algorithm. In examples 3 and 4 the MLP is trained using Output-Weight-Optimization Hidden-Weight-Optimization (OWO-HWO)[21].

Errors were averaged for 3 sets of random data with ratio of 7:3 for training and validation.

The 1st example is a benchmark function approximation problem "California Housing" from Statlib[22]. It has 20,640 observations for predicting the price of houses in California, with 8 continuous inputs (median income, housing median age, total rooms, total bedrooms, population, households, latitude, and longitude) and one continuous output (median house value).

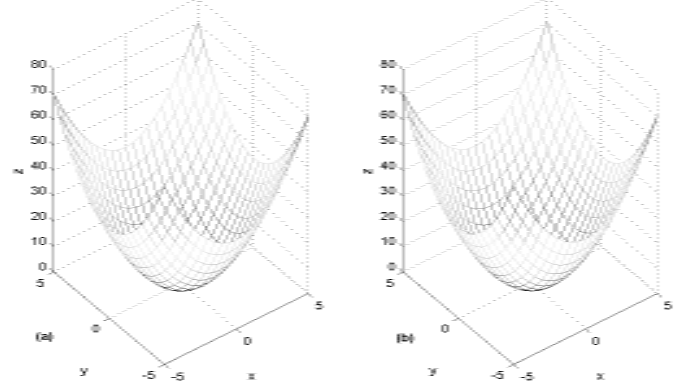


Fig. 3. (a) Modified rastrigin function (b) Approximated function by OFLN ($D = 4$).

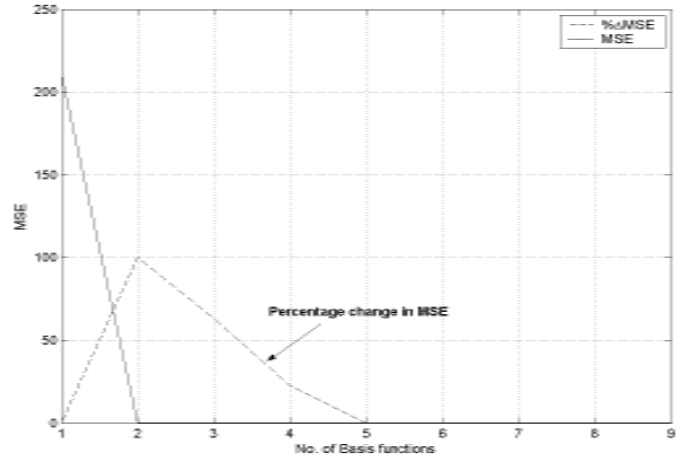


Fig. 4. MSE and Percentage change in MSE vs. no. of basis functions for approximating the Rastrigin function.

The output was normalized by subtracting the mean and dividing by the standard deviation. The MSE obtained was 0.30, 0.38 and 0.35 for the OFLN, GMDH and MLP respectively. OFLN used 27 basis functions with $D=4$. The 4th degree GMDH had 27 PBFs. An 8-18-1 MLP used a validation set for early stopping and converged at 31 epochs. Fig. 5 shows the training and validation MSEs, MSEt and MSEv vs. the number of basis function for OFLN with $D=4$. As the validation error increases for more than 27 basis functions, it is selected as a stopping criterion [10].

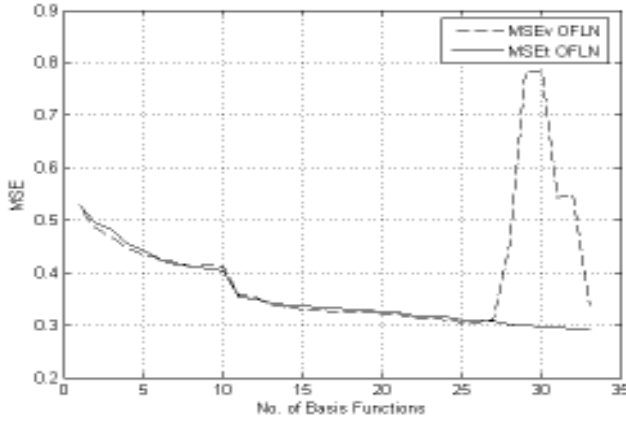


Fig. 5. Training and Validation MSE vs. no. of basis function for California Housing approximation.

The 2nd example comprises an empirical MIMO geophysical system for surface analysis from polarimetric radar measurements[23]. There are 20 inputs corresponding to VV and HH polarization at L 30, 40 deg, C 10, 30, 40, 50, 60 deg, and X 30, 40, 50 deg and 3 outputs corresponding to rms surface height, surface correlation length, and volumetric soil moisture content in g/cubic cm. Fig. 6 shows the training and validation MSE vs. no. of basis function graph for ordered PBF's 1 to 60 for D=4. Results show good generalization capabilities for the OFLN

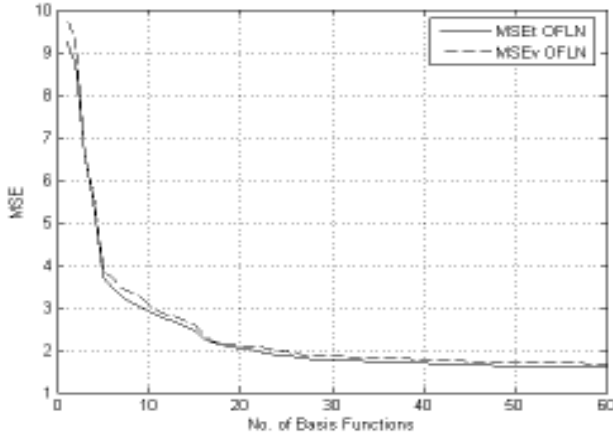


Fig. 6. Training and Validation MSE vs. no. of basis function for Radar scatter dataset.

Table 1 compares L values used for the FLN and OFLN in the California housing and radar applications. Although L_{OFLN} for California housing example for D=4 is 377 for effective generalization only 27 basis functions are sufficient, as seen in Fig. 5. Table 2 gives the corresponding training and validation MSE at each degree for the two applications.

TABLE I. NO. OF PBF'S COMPARISON FOR OFLN VS. FLN

| Degree | L_{FLN} Housing | L_{OFLN} Housing | L_{FLN} Radar | L_{OFLN} Radar |
|--------|----------------------|-----------------------|--------------------|---------------------|
| D=1 | 9 | 9 | 21 | 21 |
| D=2 | 55 | 45 | 231 | 231 |
| D=3 | 220 | 144 | 1771 | 284 |
| D=4 | 715 | 337 | 10626 | 294 |

TABLE 2. TRAINING AND VALIDATION MSE COMPARISON FOR OFLN VS. MLP

| MSE | D=1 | D=2 | D=3 | D=4 | MLP |
|----------------------|------|------|------|------|------|
| Training (Housing) | 0.36 | 0.33 | 0.31 | 0.30 | 0.35 |
| Validation (Housing) | 0.37 | 0.33 | 0.32 | 0.31 | 0.35 |
| Training (Radar) | 3.69 | 1.52 | 1.38 | 1.36 | 1.43 |
| Validation (Radar) | 3.94 | 1.81 | 1.65 | 1.6 | 1.55 |

Because of the equation (24), the OFLN's training MSE is a monotonically non-increasing function of D and L. Also for Dth degree learning, the training data need be read only (D+1) times. These attributes make OFLN training more computationally efficient than that of the GMDH.

The 3rd example involves the demodulation of a noisy Frequency Modulation (FM) samples to recover the band-limited signal. If $x[n]$ is the modulating signal, $z[n]$ is the output of FM modulator with additive noise $e[n]$, then for modulation index k_f , carrier amplitude A_c , carrier frequency f_c , modulating signal frequency f_m

$$z[n] = A_c \cdot \cos\left(2\pi f_c \cdot n + k_f \sum_{i=0}^n x[i]\right) + e[n]$$

$$k_f = \frac{2\pi f_c}{|x[n]|_{\max}} \quad (30)$$

1024 patterns are generated with $z[i]$, $0 \leq i \leq 4$ as inputs and desired $x[n]$ as output with values of A_c , f_c , f_m as 0.5, 0.1 and 0.1 respectively. Comparison of OFLN, MLP and GMDH based on training MSE (MSEt) and validation MSE (MSEv) vs. the number of basis functions is shown in Fig. 7. OFLN gives a lower MSE for training and validation compared to MLP and GMDH. The number of basis functions for MLP under consideration is given by (Number of hidden units + N + 1). The GMDH network uses 5th degree approximation for 50 iterations. Performance results for OFLN are comparatively better. Also, a system modeler can select a smaller size OFLN network with a trade-off in MSE, e.g. OFLN of size 40 compared to OFLN of size 60 has 2% additional training MSE at cost of 20 more PBF's.

Finally, results for non-linear 2 by 2 matrix inversion problem are shown in Fig. 8. The training file has 2000 patterns, each with N=M=4. The inputs are uniformly distributed between 0 and 1, and represent a matrix. The four desired outputs are elements of the corresponding inverse matrix. The determinants of the input matrices are constrained to be between .3 and 2. FLN, OFLN and MLP networks are compared in Fig. 8. Note that the FLN points are widely separated, giving the user few options as to network size.

REFERENCES

- [1] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, *Learning representations by back-propagating errors*, Nature, pp. 533-536, 1986.
- [2] K. Hornik, M. Stinchcombe, and H. White, "Multilayer Feedforward Networks Are Universal Approximators." *Neural Networks*, Vol. 2, No. 5, 1989, pp. 359-366.
- [3] K. Hornik, M. Stinchcombe, and H. White, "Universal Approximation of an Unknown Mapping and its Derivatives Using Multilayer Feedforward Networks," *Neural Networks*, vol. 3, 1990, pp. 551-560.
- [4] Michael T. Manry, Steven J. Apollo, and Qiang Yu, "Minimum Mean Square Estimation and Neural Networks," *Neurocomputing*, vol. 13, September 1996, pp. 59-74.
- [5] Dennis W. Ruck, Steven K. Rogers, Matthew Kabrisky, Mark E. Oxley and Bruce W. Suter, "The Multilayer Perceptron as an approximation to a Bayes optimal discriminant function," *IEEE Trans Neural Networks*, TNN-1(4):296-298, 1990.
- [6] Y. H. Pao and Y. Takefuji, "Functional-Link Net Computing: Theory, System Architecture, and Functionalities," *IEEE Computer*, Vol. 25, No. 5, 1992, pp. 76 - 79.
- [7] J. Macias.A., A. Sierra., F. Corbacho, "Evolving and assembling functional link networks", *IEEE Trans. on Evolutionary Computation*, Vol. 5, No. 1, 2001, pp. 54-65.
- [8] M. Klassen, Y. H. Pao., V. Chen, "Characteristics of the functional link net: a higher order delta rule net", *IEEE Con. Neural Networks*, Vol. 1, 1988, pp. 507-513.
- [9] A. G. Ivakhnenko, "Heuristic self-organization in problems of engineering cybernetics", *Automutica*, Vol.6, No.2, 1970, pp.207-219.
- [10] S. J. Farlow ed., "Self-organizing Methods in Modeling, GMDH-type Algorithms", Marcel Dekker, Inc., New York, 1984.
- [11] Y. Shin, J. Ghosh, "The pi-sigma network: an efficient higher-order neural network for pattern classification and function approximation", *Proc. of IJCNN*, Vol 1, 1991, pp. 13-18.
- [12] Y. Shin, J. Ghosh, "Ridge polynomial networks", *IEEE Trans. Neural Networks*, Vol. 6, Iss. 3, 1995, pp. 610 - 622.
- [13] S. Chen, C. F. N. Cowan, and P. M. Grant, "Orthogonal least squares learning algorithm for radial basis function networks," *IEEE Trans. Neural Networks*, vol. 2, 1991, pp. 302-309.
- [14] W. Kaminski and P. Strumillo, "Kernel Orthonormalization in Radial Basis Function Neural Networks," *IEEE Trans. Neural Networks*, Vol. 8, No. 5, 1997, pp. 1177-1183.
- [15] W. Kolmogorov. Interpolation and extrapolation of stationary series. *Bulletin de l'Academie de, U.S.S.R.*, 2:3, 1942.
- [16] F. Ling, D. Manokalis, J. G. Proakis, "A Recursive Modified Gram-Schmidt Algorithm for Least-Squares Estimation", *IEEE Trans. Acoustics, Speech, Signal Processing*, Vol. ASSP-34, No. 4, 1986, pp. 829-836.
- [17] F. J. Maldonado, M. T. Manry, T. Kim, "Finding optimal neural network basis function subsets using the Schmidt procedure", *Proc. of IJCNN*, Vol. 1, 2003, pp. 444 - 449.
- [18] J. Li, M. T. Manry, P. Narasimha, C. Yu, "Feature Selection Using a Piecewise Linear Network", *IEEE Trans. Neural Networks*, Vol 17, No. 5, 2006, pp.1101-1115.
- [19] I.V. Tetko, J. Gasteiger, R. Todeschini, A. Mauri, D. Livingstone, P. Ertl, V.A. Palyulin, E.V. Radchenko, N.S. Zefirov, A. S. Makarenko, V.Y. Tanchuk, V. V. Prokopenko, "Virtual computational chemistry laboratory - design and description," *J. Comput. Aid. Mol. Des.* 2005, 19, pp. 453-63.
- [20] VCCLAB, Virtual Computational Chemistry Laboratory, 2005, Available: <http://www.vcclab.org>.
- [21] Changhua Yu, Manry, M.T., "A modified hidden weight optimization algorithm for feedforward neural networks", *IEEE Asilomar Con. on Signals, Systems and Computers*, 2002. Vol. 2, 2002, pp.1034-1038.
- [22] R.K. Pace, R. Barry, "Sparse spatial autoregressions", *Statistics & Probability Letters*, Vol. 33, No. 3, May 1997, pp. 291-297.
- [23] Y. Oh, K. Sarabandi, and F.T. Ulaby, "An Empirical Model and an Inversion Technique for Radar Scattering from Bare Soil Surfaces", *IEEE Trans. Geoscience and Remote Sensing*, 1992, pp. 370-381.

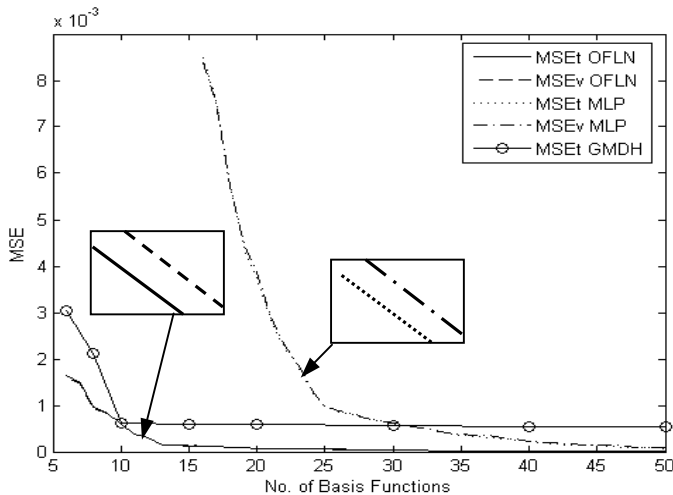


Fig. 7. Training and Validation MSE vs. no. of basis function comparison of OFLN, MLP and GMDH for FM Demodulator.

We see that all three networks perform similarly when the number of basis functions is 23 or less. However, for this dataset, the MLP has an advantage for 24 or more basis functions.

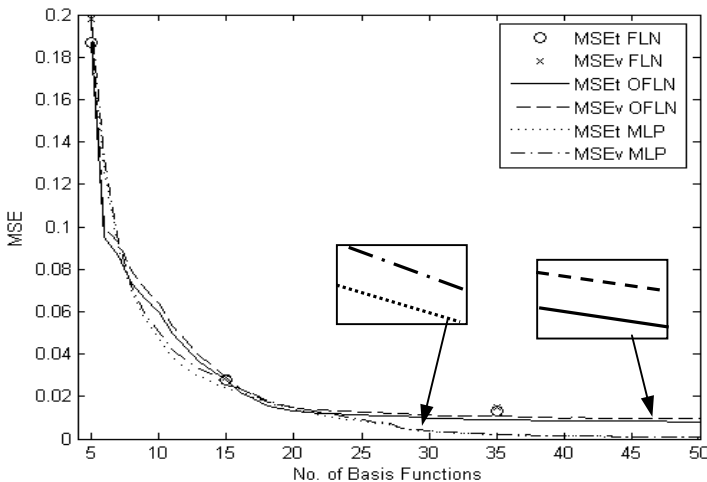


Fig. 8. Training and Validation MSE vs. no. of basis functions comparison of FLN, OFLN and MLP for 2 by 2 Matrix inversion.

VI. CONCLUSIONS

The OFLN gives a concise, methodically ordered and computationally efficient representation for supervised, non-parametric MIMO systems that is a better than that of some other PNN networks. It reduces the number of passes through the dataset. It can certainly be applied to many nonlinear function approximation, structure identification and optimization problems. The OFLN can be extended to classification problems as well. Future work will include the development of computationally more efficient methods for the generation and index storage of higher order PBFs so as to enhance the scalability of the network.