On the Selection of Nodes in Linear-in-the-Weight Neural Networks

ELIAS B. KOSMATOPOULOS[1] & NIKITAS J. DIMOPOULOS[2]

Abstract— **In this paper, we propose algorithms for selecting the regressor terms in linear-in-the-weight neural networks. These algorithms are accompanied by appropriate learning algorithms for adjusting the weights of the neural network. By analyzing an appropriate error functional, we investigate the convergence properties of the proposed algorithms; moreover, we investigate the optimality of these algorithms and we construct conditions - regarding the nature of the regressor terms - under which the proposed algorithms are optimal.**

I. LINEAR-IN-THE-WEIGHTS NEURAL NETWORKS

Neural networks are known to possess powerful approximation capabilities; theoretical works by many authors [3, 12, 8, 9, 10] have shown that various neural network models are capable of approximating either functions or dynamical systems to any degree of accuracy. In this paper, we will concentrate our attention to linear-in-the-weights neural networks. In general, such neural networks are mathematically described as follows

$$y = w^T \phi(x) \tag{1.1}$$

where $x \in \Re^{n_1}$ denotes the input vector to the neural network, $y \in \Re^{n_2}$ denotes the output vector of the neural network, $w \in \Re^{n_2 \times L}$ denotes the synaptic matrix of the neural network, $\phi : \Re^{n_1} \mapsto \Re^L$ is a nonlinear vector function of *regressor terms* and the integer L denotes the number of regressor terms. It can be shown that various neural network models belong to the class (1.1); for instance high order neural networks, radial basis function networks, neural network with shifted sigmoidals, adaptive fuzzy systems, etc, (see e.g. [3, 12, 8, 9, 10] and the references therein) belong to the class of neural networks (1.1).

A very important property that many neural models of the form (1.1) satisfy is the following:

(P1) We say that a family of neural networks of the form (1.1) is a family of universal approximators, if for every continuous function $F : \Re^{n_1} \mapsto \Re^{n_2}$, for every $\varepsilon > 0$ and every compact subset $\mathcal{X} \subset \Re^{n_1}$ there is an integer L and a matrix w^* such that the neural network with L regressor terms satisfies $\int_{\mathcal{X}} |F(x) - w^{*T}\phi(x)|^2 dx < \varepsilon$.

Many families of neural models of the form (1.1) such as high order neural networks, radial basis function networks, neural network with shifted sigmoidals satisfy property (P1).

Suppose that $\phi_1(\cdot), \phi_2(\cdot), \ldots, \phi_k(\cdot), \ldots$ are all the possible regressor terms. Let us now fix the function $F(\cdot)$ and the compact set $\mathcal{X}$; let also a neural network with L regressor terms, whose indices belong to the set of integers $\mathcal{L}$ with cardinality L. Then, the *optimal synaptic matrix* w^* and the *optimal modeling error* $\nu(\cdot)$ w.r.t. $\mathcal{L}, \phi, F(\cdot)$ and $\mathcal{X}$ are defined as follows: $w^* \triangleq \mathcal{W}(\mathcal{L}, \phi, F, \mathcal{X}) \triangleq \arg\min_w \int_{\mathcal{X}} |F(x) - \sum_{i \in \mathcal{L}} w_i^T \phi_i(x)|^2 dx$ where w denotes the matrix whose columns equal w_i, $i \in \mathcal{L}$, and $\nu(x) \triangleq \mathcal{N}(\mathcal{L}, \phi, F, \mathcal{X})(x) \triangleq F(x) - \sum_{i \in \mathcal{L}} w_i^T \phi_i(x)$. It is worth noticing that from property (P1) $\int_{\mathcal{X}} |\nu(x)|^2$ can be made arbitrarily small by appropriately selecting L. In general, $\int_{\mathcal{X}} |\nu(x)|^2$ becomes smaller whenever L increases [9, 10, 12]. We will say that the neural network (1.1) approximates $F(\cdot)$ *with degree of accuracy* ε if $\int_{\mathcal{X}} |F(x) - \sum_{i \in \mathcal{L}} w_i^T \phi_i(x)|^2 dx < \varepsilon$ for some $w_i \in \Re^{n_2}$. In [12, 8, 9, 10] many *globally convergent* learning laws for adjusting the weights of linear-in-the-weights neural networks have been proposed. Among the nice properties of these learning laws is that - contrary to the backpropagation learning algorithms - they guarantee convergence to the global minimum. While in multilayer neural networks the error functional possesses many local minima, in the case of linear-in-the-weights neural networks there exists one and only one (global) minimum.

II. A CONVERGENT ALGORITHM

In order to simplify the analysis, we will assume that $n_2 = 1$, i.e., we will concentrate our attention to single-output neural networks. It is worth noticing that all the results of this paper can be straightforwardly extended to the multi-dimensional case. Consider now that we select a linear-in-the-weights neural network satisfying property (P1) - i.e., it is a universal approximator - and suppose that we wish to approximate an unknown function $F(\cdot)$ with degree of accuracy ε. Let us consider now a family of L regressor terms $\Phi := \{\phi_1(\cdot), \phi_2(\cdot), \ldots, \phi_L(\cdot)\}$, and suppose that we want to construct a neural network of the form (1.1) in order to approximate the function $F(\cdot)$ over $\mathcal{X}$. Moreover, let us suppose that we initially select - probably randomly - ℓ_1 regressor terms belonging to Φ, and try to approximate $F(\cdot)$ using a neural network with these ℓ_1 regressor terms. Let $\mathcal{C}$ denote a subset of $\underline{L} := \{1, 2, \ldots, L\}$ such that $i \in \mathcal{C}$ if the regressor term $\phi_i(\cdot)$ is contained in our initial choice of ℓ_1 regressor terms (obviously the cardinality of $\mathcal{C}$ is ℓ_1). Then the neural network used for approximation of $F(\cdot)$ is described as follows

$$y_1 = \sum_{i \in \mathcal{C}} w_i \phi_i(x) \tag{2.1}$$

Suppose now that we train neural network (2.1) in order to approximate $F(\cdot)$. Obviously the mean-square approximation error is given by $E := \int_{\mathcal{X}} \left| \sum_{i \in \mathcal{C}} w_i \phi_i(x) - F(x) \right|^2 dx$. Let us now pick a regressor term - not belonging to the initial set of ℓ_1 regressor terms - i.e., let us select a $j^* \in \underline{L}$ such that $j^* \notin \mathcal{C}$, and let us form a new neural network containing this additional regressor term,

$$y_1 = \sum_{i \in \mathcal{C}} w_i \phi_i(x) + w_{j^*} \phi_{j^*}(x) \tag{2.2}$$

[1]Electrical Engineering - Systems, University of Southern California, Los Angeles, CA 90089, e-mail: kosmatop@bode.usc.edu

[2]Department of Electrical and Computer Engineering, University of Victoria, Victoria, B.C., Canada V8W 3P6, e-mail: nikitas@sirius.uvic.ca

Then the mean-square error of the new neural network is given by

$$E' := \int_{\mathcal{X}} \left| \sum_{i \in \mathcal{C}} w_i \phi_i(x) + w_{j^*} \phi_{j^*}(x) - F(x) \right|^2 dx$$

or

$$\begin{aligned} E' &= \int_{\mathcal{X}} \left| \sum_{i \in \mathcal{C}} w_i \phi_i - F \right|^2 dx + \int_{\mathcal{X}} |w_{j^*} \phi_{j^*}|^2 dx \\ &\quad + 2 \int_{\mathcal{X}} \left(\sum_{i \in \mathcal{C}} w_i \phi_i - F \right) w_{j^*} \phi_{j^*} dx \\ &= E + w_{j^*}^2 \int_{\mathcal{X}} \phi_{j^*}(x)^2 dx \qquad (2.3) \\ &\quad + 2 w_{j^*} \int_{\mathcal{X}} \left(\sum_{i \in \mathcal{C}} w_i \phi_i(x) - F(x) \right) \phi_{j^*}(x) dx \end{aligned}$$

Let $\mathcal{A}_{j^*} := -\frac{\int_{\mathcal{X}} \left(\sum_{i \in \mathcal{C}} w_i \phi_i(x) - F(x) \right) \phi_{j^*}(x) dx}{\int_{\mathcal{X}} \phi_{j^*}(x)^2 dx}$. If, the - quadratic in w_{j^*} - term $\mathcal{Q}_{j^*} := w_{j^*}^2 \int_{\mathcal{X}} \phi_{j^*}(x)^2 dx + 2 w_{j^*} \int_{\mathcal{X}} \left(\sum_{i \in \mathcal{C}} w_i \phi_i(x) - F(x) \right) \phi_{j^*}(x) dx$ is negative then we have that $E' < E$. It can be easily seen that $\mathcal{Q}_{j^*}$ is negative iff either $\mathcal{A}_{j^*} < 0$ and $w_{j^*} \in (2\mathcal{A}_{j^*}, 0)$ or $\mathcal{A}_{j^*} > 0$ and $w_{j^*} \in (0, 2\mathcal{A}_{j^*})$. In other words, *an additional regressor term will improve the neural network's performance, provided that $\mathcal{A}_{j^*}$ is different than zero and that the weight w_{j^*} is appropriately chosen.*

From the above analysis, it is clear that if the quantity $\mathcal{A}_{j^*}$ can be computed, then we can construct very easily convergent algorithms for selecting the regressor terms of the neural network. Suppose now that we are provided with N training points of the form $(x^{(k)}, y^{(k)}), k = 1, \ldots, N$ where $y^{(k)}$ satisfies[1] $y^{(k)} = F(x^{(k)})$. In this case, we can approximate $\mathcal{A}_{j^*}$ by $\hat{\mathcal{A}}_{j^*}$ computed as follows

$$\hat{\mathcal{A}}_{j^*} := -\frac{\sum_{k=1}^{N} \left(\sum_{i \in \mathcal{C}} w_i \phi_i(x^{(k)}) - y^{(k)} \right) \phi_{j^*}(x^{(k)})}{\sum_{k=1}^{N} \phi_{j^*}(x^{(k)})^2} \qquad (2.4)$$

Other, more accurate - and of course more complicated - numerical integration methods can be applied as well. In this paper we will assume that

(A1) The training points $(x^{(k)}, y^{(k)})$ are such that $|\mathcal{A}_{j^*} - \hat{\mathcal{A}}_{j^*}| \to 0$ as $N \to \infty$.

In simple words, the above assumption states that the training data are rich enough in the sense that they contain enough information about the unknown function $F(\cdot)$. Thus, in the case where the above assumption is valid, we have no pathological situations such as the case where big areas in the input-output space $\mathcal{X} \times F(\mathcal{X})$ contain no training data, the case where all the data belong to a small subset of $\mathcal{X} \times F(\mathcal{X})$, etc. Moreover, for technical reasons we will assume that $\int_{x \in \mathcal{X}} \phi_i(x)^2 dx \neq 0, \quad \forall i \in \underline{L}$. If the above assumption does not hold for a particular regressor term $\phi_i(\cdot)$, then the effect of this term to the neural network output will be negligible, and thus it does not make sense to include this regressor term in the set Φ.

[1] Our analysis can be easily extended in the case of noisy training points, i.e., in the case where $y^{(k)} = F(x^{(k)}) + \xi$, where ξ is a zero-mean random process.

We are now ready to propose our first algorithm for selecting the regressor terms in the neural network. The algorithm is as follows (here $\mathcal{C}$ denotes the set of indices of the regressor terms included in the current neural network while $\mathcal{R}$ denotes the indices of the remaining regressor terms; κ denotes the number of iterations we have run the algorithm; in the sequel we will say the a particular neural network is *associated* with the set $\mathcal{C}$ if its regressor terms' indices form the set $\mathcal{C}$):

ALGORITHM A.1

1. Set $\mathcal{R} := \{1, \ldots, L\}$ and $\mathcal{C} := \emptyset$. Set the desired degree of accuracy ε; set $\kappa = 0$.

2. Select - possibly randomly - the initial ℓ_1 regressor terms; train the neural network using any appropriate training algorithm and using the training data $(x^{(k)}, y^{(k)}), k = 1, \ldots, N$. Let $\mathcal{C}$ be the set of integers satisfying the following condition: $i \in \mathcal{C}$ if the regressor term $\phi_i(\cdot)$ is contained in our initial choice of ℓ_1 regressor terms. Set $\mathcal{R} = \underline{L} \cap \mathcal{C}$.

3. For all $j \in \mathcal{R}$ set $w_j^* = \hat{\mathcal{A}}_j$. and calculate $e_j := w_j^{*2} \int_{\mathcal{X}} \phi_j(x)^2 dx + 2 w_j^* \int_{\mathcal{X}} \left(\sum_{i \in \mathcal{C}} w_i \phi_i(x) - F(x) \right) \phi_j(x) dx$

4. Select j^* as follows $j^* = \arg\min_j e_j$

5. If $\hat{\mathcal{A}}_{j^*} = 0$, then STOP.

6. If $\hat{\mathcal{A}}_{j^*} \neq 0$, add the regressor term $\phi_{j^*}(\cdot)$ in the neural network and set $\mathcal{C} = \mathcal{C} + \{j^*\}$ and $\mathcal{R} = \mathcal{R} + \{j^*\}$. Set $w_{j^*} = \hat{\mathcal{A}}_{j^*}$.

7. Calculate - numerically - the mean-square approximation error $E := \int_{\mathcal{X}} \left| \sum_{i \in \mathcal{C}} w_i \phi_i(x) - F(x) \right|^2 dx$ If $E \leq \varepsilon$ OR $\mathcal{R} = \emptyset$, STOP; OTHERWISE GO TO STEP 3.

□

The next theorem summarizes the convergence properties of algorithm A.2.

Theorem II.1: Consider the algorithms A.2. Suppose that assumption (A1) holds and let E^κ be the mean-square approximation error of the neural network at the κ-th iteration of the algorithm. Then, as $N \to \infty$, $E^{\kappa+1} \leq E^\kappa$. Moreover, let κ^{total} denote the total number of iterations of the algorithm. Then $E^{\kappa+1} < E^\kappa$ for all $\kappa \in (0, \kappa^{total} - 1)$.

III. Algorithms based on the Optimal Selection of Weights

In this section we present two new algorithms for selecting the regressor terms. The difference between these new algorithms and the ones of the previous section is that in the new algorithms the weights are computed in such a way that they optimize the mean-square error. Thus, although these new algorithms are very similar to the Algorithm A.2, in the sense that the regressor term that is selected to be added is the one that minimizes the quadratic term $\mathcal{Q}_j$, the advantage of the new algorithms is that the weights are "optimally selected". Of course, such an optimal selection of the weights results in a more computationally complicated algorithm. Let us now consider the problem of optimal selection of weights in a neural network with fixed regressor terms (let us assume that

this neural network is associated with the set $\mathcal{C}_h$). Then, the mean-square error for this neural network is given by

$$E^h := \int_{\mathcal{X}} \left| \sum_{i \in \mathcal{C}_h} w_i \phi_i(x) - F(x) \right|^2 dx \qquad (3.5)$$

By calculating the gradient (w.r.t. w_i) of the above quantity and by setting this gradient equal to the zero vector, we can easily see that the optimal weights $w_i^{h^*}$ (that is, the weights for which the minimum mean-square error is achieved) are given by the following equation

$$w^{h^*} = \left(\int_{\mathcal{X}} \phi^h(x) \phi^{h^T}(x) dx \right)^{-1} \left(\int_{\mathcal{X}} F(x) \phi^h(x) dx \right) \qquad (3.6)$$

where w^{h^*} is $|\mathcal{C}_h|$-dimensional vector whose entries are $w_i^{h^*}$ with $i \in \mathcal{C}_h$ and ϕ^h is $|\mathcal{C}_h|$-dimensional vector whose entries are $\phi_i(x)$ with $i \in \mathcal{C}_h$. Let now $P^h := \left(\int_{\mathcal{X}} \phi^h(x) \phi^{h^T}(x) dx \right)^{-1}$ and P_{ij}^h denote the (ij)-th entry of the matrix P^h. Then, equation (3.6) can be rewritten as

$$w_i^{h^*} = \sum_{i \in \mathcal{C}_h} P_{ij}^h \left(\int_{\mathcal{X}} F(x) \phi_j(x) dx \right) \qquad (3.7)$$

Using (3.7), equation (3.5) can be rewritten as

$$E^h = \int_{\mathcal{X}} \left| \sum_{i \in \mathcal{C}_h} \sum_{j \in \mathcal{C}_h} P_{ij}^h \phi_i(x) \left(\int_{\mathcal{X}} F(x) \phi_j(x) dx \right) - F(x) \right|^2 dx \qquad (3.8)$$

Using the same ideas as in the previous section but based on the above mathematical formulas, we can develop new algorithms for selecting the regressor terms of the neural network. As in the algorithm A.2, we start with a neural network ℓ_1 regressor terms and, at each step of the algorithm, we augment the neural network by adding the "best" regressor term. The "best" regressor term is selected to be the one which - among all the other regressor terms in $\mathcal{R}$ - contributes the best to the minimization of the mean-square approximation error. As in the previous section, the regressor term that contributes the best to the minimization of the mean-square error is the one that minimizes the quantity $\int_{\mathcal{X}} |w_j \phi_j(x)|^2 dx + 2 \int_{\mathcal{X}} \left(w_j \phi_j(x) \left[\sum_{i \in \mathcal{C}_h} w_i \phi_i(x) - F(x) \right] \right) dx$. The difference between the algorithms of the previous section and the ones proposed in this section lies on the computation of the weights w_i.

Two different algorithms are proposed: these algorithms differ in the way the weights are computed. In the first of them, the weights w_i are set equal to their optimal values for the case where all the L regressor terms are used, while in the second algorithm at each iteration we recompute the weights' optimal values. We now present the two algorithms and we will analyze and explain them further later on. The notation remains the same with that of the previous section.

ALGORITHM A.3

1. Set $\mathcal{R} := \{1, \ldots, L\}$ and $\mathcal{C} := \emptyset$. Set the desired degree of accuracy ε; set $\kappa = 0$.

2. Select - possibly randomly - the initial ℓ_1 regressor terms. Let $\mathcal{C}$ be the set of integers satisfying the following condition: $i \in \mathcal{C}$ if the regressor term $\phi_i(\cdot)$ is contained in our initial choice of ℓ_1 regressor terms. Set $\mathcal{R} = \underline{L} \cap \mathcal{C}$.

3. Let $\mathcal{C}_L$ denote the set of indices that corresponds to the neural network composed of all L regressor terms belonging in Φ. Let also P^L, ϕ^L be defined similarly to P^h, ϕ^h. Then, compute w_i^L as follows

$$w_i^L = \sum_{i \in \mathcal{C}_L} P_{ij}^L \left(\int_{\mathcal{X}} F(x) \phi_j(x) dx \right)$$

4. Select j^* as follows

$$j^* = \arg\min_{j \in \mathcal{R}} \left\{ \int_{\mathcal{X}} \left| w_j^L \phi_j(x) \right|^2 dx + 2 \int_{\mathcal{X}} \left(w_j^L \phi_j(x) \left[\sum_{i \in \mathcal{C}} w_i^L \phi_i(x) - F(x) \right] \right) dx \right\}$$

5. Add the regressor term $\phi_{j^*}(\cdot)$ in the neural network and set $\mathcal{C} = \mathcal{C} + \{j^*\}$ and $\mathcal{R} = \mathcal{R} + \{j^*\}$; set $\kappa = \kappa + 1$.

6. Calculate - numerically - the mean-square approximation error

$$E := \int_{\mathcal{X}} \left| \sum_{i \in \mathcal{C}} w_i^L \phi_i(x) - F(x) \right|^2 dx$$

If $E \leq \varepsilon$ OR $\mathcal{R} = \emptyset$, STOP; OTHERWISE GO TO STEP 4.

□

ALGORITHM A.4

1. Set $\mathcal{R} := \{1, \ldots, L\}$ and $\mathcal{C} := \emptyset$. Set the desired degree of accuracy ε; set $\kappa = 0$.

2. Select - possibly randomly - the initial ℓ_1 regressor terms. Let $\mathcal{C}$ be the set of integers satisfying the following condition: $i \in \mathcal{C}$ if the regressor term $\phi_i(\cdot)$ is contained in our initial choice of ℓ_1 regressor terms. Set $\mathcal{R} = \underline{L} \cap \mathcal{C}$.

3. Let P^c, ϕ^c be defined similarly to P^h, ϕ^h for the neural network associated with the set $\mathcal{C}$. Then, compute $w_i^{\mathcal{C}}$ as follows

$$w_i^{\mathcal{C}} = \sum_{i \in \mathcal{C}} P_{ij}^c \left(\int_{\mathcal{X}} F(x) \phi_j(x) dx \right)$$

4. Calculate - numerically - the mean-square approximation error

$$E := \int_{\mathcal{X}} \left| \sum_{i \in \mathcal{C}} w_i^{\mathcal{C}} \phi_i(x) - F(x) \right|^2 dx$$

If $E \leq \varepsilon$ OR $\mathcal{R} = \emptyset$, STOP; OTHERWISE GO TO NEXT STEP; set $\kappa = \kappa + 1$.

5. For all $j \in \mathcal{R}$ calculate $\bar{w}_j$ as follows

$$\bar{w}_j := \arg\min_{w_j} \left\{ w_j^2 \int_{\mathcal{X}} \phi_j(x)^2 dx + 2 w_j \int_{\mathcal{X}} \left(\sum_{i \in \mathcal{C}} w_i^{\mathcal{C}} \phi_i(x) - F(x) \right) \phi_j(x) dx \right\}$$

and let

$$e_j := \bar{w}_j^2 \int_{\mathcal{X}} \phi_j(x)^2 dx + 2 \bar{w}_j \int_{\mathcal{X}} \left(\sum_{i \in \mathcal{C}} w_i^{\mathcal{C}} \phi_i(x) - F(x) \right) \phi_j(x) dx$$

Select j^* as follows

$$j^* = \arg\min_{j \in \mathcal{R}} \{e_j\}$$

6. Add the regressor term $\phi_{j^*}(\cdot)$ in the neural network and set $\mathcal{C} = \mathcal{C} + \{j^*\}$ and $\mathcal{R} = \mathcal{R} + \{j^*\}$.

7. GO TO STEP 3

□

The difference between the two above algorithms is that in Algorithm A.4 we compute the optimal weights of the network associated with the set $\mathcal{C}$ at each iteration while in Algorithm A.3 we compute the optimal weights for the network composed of all L possible regressor terms and we keep the weights constant and equal to their initial value at each iteration. However, we must mention here that in the case where L is very large - and that's usually the case - Algorithm A.3 becomes impossible to be implemented in practice, since it requires calculation of all L weights w_i^L (which in turn implies that we need to calculate the inverse of the $L \times L$ matrix $(\int_{\mathcal{X}} \phi^L(x)\phi^{L^T}(x))$, etc). Therefore, Algorithm A.3 is useful only in cases where the unknown function can be approximated by a small number of regressor terms.

The next theorem summarizes the properties of the two above algorithms.

Theorem III.1: Consider the algorithms A.3 and A.4. Suppose that assumption (A1) holds and let E^κ be the mean-square approximation error of the neural network at the κ-th iteration of the algorithm. Then, as $N \to \infty$, for both algorithms the following hold:

(a) For the case of Algorithm A.4 we have that $E^{\kappa+1} \leq E^\kappa$.

(b) Fix a constant $\varepsilon > 0$. If L is sufficiently large, in the sense that there exists at least one neural network composed of some of the regressor terms in Φ that approximates $F(\cdot)$ with accuracy ε, then, for both Algorithms A.3 and A.4, there exists a finite number of algorithm iterations after which $E < \varepsilon$.

IV. Optimal Selection of the Regressor Terms

So far we have constructed convergent algorithms for the selection of regressor terms; a natural question that arises is if such algorithms are *optimal* or under which conditions these algorithms are optimal. In this section, we will examine under which conditions the proposed algorithms are optimal. However, before we proceed we must clarify what we mean by saying optimal algorithm for the selection of the regressor terms. In order to do so, consider again the problem of selection of regressor terms as stated in the first paragraph of the section III. Obviously, in the case where L is sufficiently large, there may be more than one neural networks (with regressor terms from Φ) possessing the property that the mean-square error that corresponds to these neural networks is less than the desired accuracy ε. That is, there may be more than one (say H) sets of indices $\mathcal{C}_h$, $h = 1, 2, \ldots, H$ satisfying the following property: for any $h \in \{1, \ldots, H\}$ there exists a set of[2] weights $w_i, i = 1, \ldots, |\mathcal{C}_h|$ satisfying

$$\int_{\mathcal{X}} \left| \sum_{i \in \mathcal{C}_h} w_i \phi_i(x) - F(x) \right|^2 dx < \varepsilon \qquad (4.1)$$

[2] $|\mathcal{C}|$ denotes the cardinality of the set $\mathcal{C}$.

Then we say that the neural network whose set of indices is the set $\mathcal{C}_{h^*}$ is *optimal*, if this neural network satisfies the above property (4.1) and moreover it has the least number of high order connections among all the other neural networks satisfying (4.1), that is, the optimal neural network satisfies

$$|\mathcal{C}_{h^*}| = \min_{h \in \{1, \ldots, H\}} |\mathcal{C}_h|$$

An algorithm that converges to a neural network associated with the set $\mathcal{C}_{h^*}$ will be called *ε-optimal* algorithm, while the neural network associated with $\mathcal{C}_{h^*}$ will be called *ε-optimal* neural network.

A. Algorithm A.3 is 0-Optimal in the Case of Perfect Matching

Let us consider the case where the unknown function $F(\cdot)$ can be *exactly reconstructed* by a neural network, that is, there is a set of indices $\mathcal{C}^*$ and a set of weights w_i^*, $i \in \mathcal{C}^*$ such that

$$F(x) = \sum_{i \in \mathcal{C}^*} w_i^* \phi_i(x) \qquad (4.2)$$

Obviously, the neural network (4.2) is an 0-optimal neural network In order to examine the conditions under which the proposed algorithms are optimal, we will make the following two assumptions:

(A2) The set $\mathcal{C}^*$ is a subset of $\{1, \ldots, L\}$.

(A3) The set $\mathcal{C}^0$ of ℓ_1 regressor terms (see Step 2 in all Algorithms) is a subset of $\mathcal{C}^*$.

The first assumption states that the set Φ of all possible regressor terms that the Algorithm examines contains all the regressor terms of the 0-optimal neural network (4.2). The second assumption states that the initial neural network that is used in order to start the Algorithm is such that its regressor terms are regressor terms of the the 0-optimal neural network (4.2). Then, we have the followign result.

Proposition IV.1: Consider Algorithm A.3 and assume that the unknown function $F(\cdot)$ satisfies (4.2). Also assume that (A1), (A2), (A3) hold. Then, as $N \to \infty$, the Algorithm is 0-optimal, and, moreover, it converges to the neural network (4.2), i.e., the Algorithm converges to a neural network which does not have more regressor terms than the neural network (4.2).

B. The case of Orthogonal Regressor Terms

In this subsection, we will consider the extreme case where the regressor terms $\phi_i(\cdot)$ are mutually orthogonal. It is worth noticing that the problem of optimally selecting the regressor terms in orthogonal networks was solved decades ago (the most known case is the case of Fourier series; the reader is referred to any textbook about Fourier series); in this paper we consider the case of orthogonal regressor terms in order to examine the optimality of the proposed algorithms, on the one hand, and to examine the case where pseudo-orthogonal regressor terms (for the definition of pseudo-orthogonal regressor terms see next subsection) are used since pseudo-orthogonal regressor terms possess many advantages over orthogonal ones. Consider now that the regressor terms $\phi_i(\cdot)$ are mutually orthogonal, that is, they satisfy the following[3] relation:

$$\int_{\mathcal{X}} \phi_i(x)\phi_j(x)dx = \begin{cases} 0 & \text{if } i \neq j \\ 1 & \text{if } i = j \end{cases} \qquad (4.4)$$

[3] For simplicity, we will consider the case where the regressor terms are orthonormal, i.e., they satisfy relation (4.4); however, all the results can be

Many known networks (see e.g. [12, 6]), satisfy the above relation.

The next proposition summarizes the properties of the proposed algorithms in the case where the regressor terms are ortogonal.

Proposition IV.2: In the case of orthogonal regressor terms, Algorithms A.2, A.3 and A.4 are ε-optimal for all non-negative ε.

C. The case of Pseudo-Orthogonal Regressor Terms

Let us now consider the case where instead of using orthogonal regressor terms we use *pseudo-orthogonal* ones; here the term pseudo-orthogonal is used for regressor terms that satisfy

$$\int_{\mathcal{X}} \phi_i(x)\phi_j(x)dx \overset{\delta}{\approx} \begin{cases} 0 & \text{if } i \neq j \\ 1 & \text{if } i = j \end{cases} \tag{4.5}$$

where δ is a small positive number and the symbol $\overset{\delta}{\approx}$ must be interpreted as follows

$$x \overset{\delta}{\approx} y \Longleftrightarrow |x - y| < \delta$$

Pseudo-orthogonal terms possess certain advantages over orthogonal ones; the most important of them is that pseudo-orthogonal terms possess better approximation capabilities in the sense that if there are two networks approximating the same function, one with orthogonal regressor terms and the other with pseudo-orthogonal ones, and moreover for every regressor term $\phi_i(\cdot)$ in the orthogonal network there is a regressor term $\hat{\phi}_j(\cdot)$ in the pseudo-orthogonal network such that $\phi_i(\cdot) \overset{\delta}{\approx} \hat{\phi}_j(\cdot)$, then the pseudo-orthogonal network produces better (smaller) mean-square error than the orthogonal network does. In other words, if we "deform" the regressor terms of an orthogonal network by making them pseudo-orthogonal then we increase the approximation capabilities of the network.

The next proposition summarizes the properties of the proposed algorithms in the case where the regressor terms are pseudo-ortogonal.

Proposition IV.3: Consider one of the Algorithms A.2, A.3 or A.4. Then for each of these algorithms the following is true: in the case of pseudo-orthogonal terms, there exist δ^*, ε^* such that the algorithm is ε-optimal for all $\delta \in (0, \delta^*)$ and all $\varepsilon > \varepsilon^*$.

V. Simulations

In order to examine the applicability of the proposed algorithms, we performed two different sets of simulations.

Simulation 1. In our first set of simulations we selected a High Order Neural Network (HONN) which was used for the approximation of three different nonlinear functions. In all three examples we used two-input single-output functions. The number L of all possible regressor terms was set equal to 15 and the set $\underline{L}$ of all possible regressor terms was as follows: $\underline{L} = \{1, S_1, S_2, S_1^2, S_2^2, S_1S_2, S_1^3, S_1^2S_2, S_1S_2^2, S_2^3, S_1^4, S_1^3S_2^2, S_1^2S_2^2, S_1S_2^3, S_2^4\}$

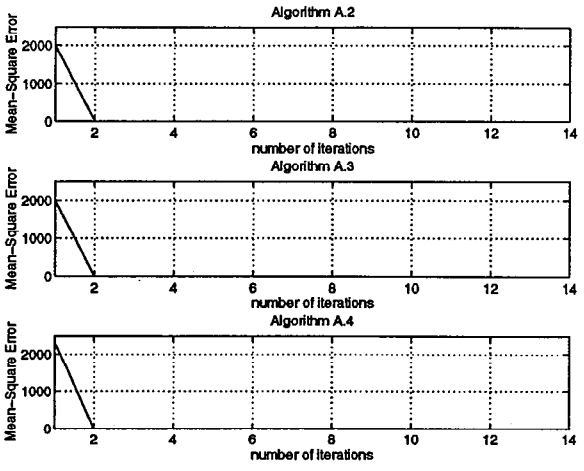

Figure 1: Performance of the Algorithms A.2, A.3 and A.4 for the case of function $F_1(\cdot)$.

where $S_i = S(x_i)$; x_i $i = 1, 2$ denote the two input variables and $S(\cdot)$ is a sigmoidal function chosen as follows

$$S(x) = \frac{6}{1 + e^{-x}} - 3$$

The three different functions used in simulations were the following:

$$F_1(x_1, x_2) = 1 - 0.9S_2^2 + 0.6S_2^2S_1$$

$$F_2(x_1, x_2) = 1 - 0.9sin(S_2^2) + 0.6S_2^2S_1$$

$$F_3(x_1, x_2) = cos(1 - 0.9sin(S_2^2) + 0.6S_2^2S_1) + sin(S_1^4)$$

As it can be easily observed, the first of the above functions is a HONN. Figures 1-3 plot the mean-square error versus number of iterations) for the Algorithm A.2 (upper subfigure), Algorithm A.3 (middle subfigure) and Algorithm A.4 (lower subfigure). As we can see, Algorithm A.3 and A.4 are superior to Algorithm A.2. Although Algorithm A.2 is a convergent algorithm there are cases (e.g. the cases of functions $F_2(\cdot)$ and $F_3(\cdot)$) where the Algorithm A.2 converges to a neural network which is very far from the optimal one. However, all three algorithms converge to the optimal network in the case where the unknown function can be exactly described by a neural network (the case of $F_1(\cdot)$). Here, we must mention that, in the case where the unknown function can be exactly described by a neural network, we can apriori guarantee that Algorithm A.2 and A.4 will converge to the optimal neural network. As we have shown in subsection V.A, only Algorithm A.3 can be proven that converges to the optimal network.

On the other hand we can see that in all three examples the algorithms are convergent with the exeption of the last example for the case of Algorithms A.3 and A.4 where we have cases where the Mean-Square Error at a particular iteration is larger than that of the previous iteration. This happens because (see also Remark IV.1) the number N is not sufficiently large and thus the calculated weights are close but not exactly equal to their optimal values.

Simulation 2. In our second set of simulations we selected a neural network with pseudo-orthogonal regressor terms. For simplicity, we considered the case where the input is one-dimensional; the number L of all possible regressor terms was set equal to 15 and the set $\underline{L}$ of all possible regressor terms was as follows: $\underline{L} = \{R(x), R(x+0.5), R(x-0.5), R(x+1), R(x-1), R(x+1.5), R(x-1.5), R(x+2), R(x-2), R(x+$

easily extended to the case where relation (4.4) is replaced by

$$\int_{\mathcal{X}} \phi_i(x)\phi_j(x)dx = \begin{cases} 0 & \text{if } i \neq j \\ p_j & \text{if } i = j \end{cases} \tag{4.3}$$

where p_j is a constant.

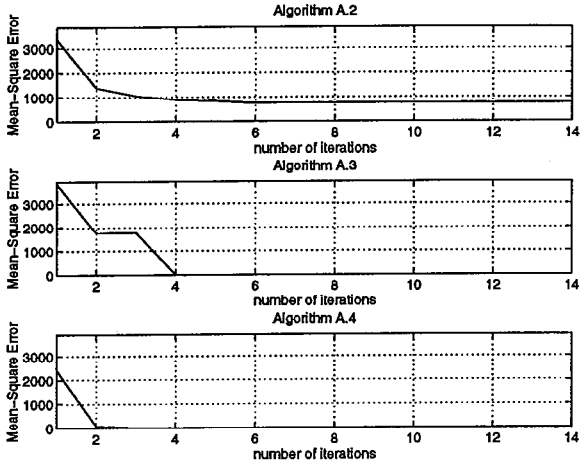

Figure 2: Performance of the Algorithms A.2, A.3 and A.4 for the case of function $F_2(\cdot)$.

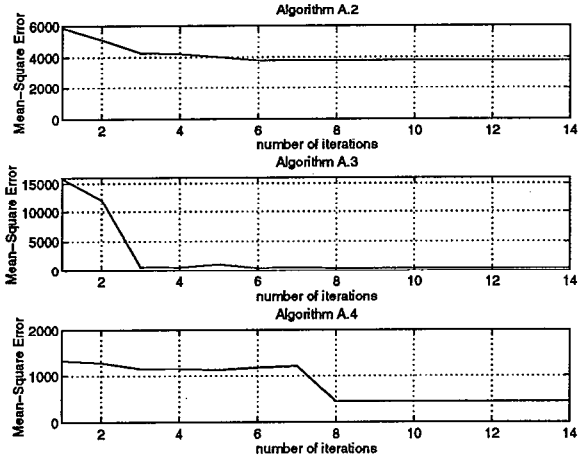

Figure 3: Performance of the Algorithms A.2, A.3 and A.4 for the case of function $F_3(\cdot)$.

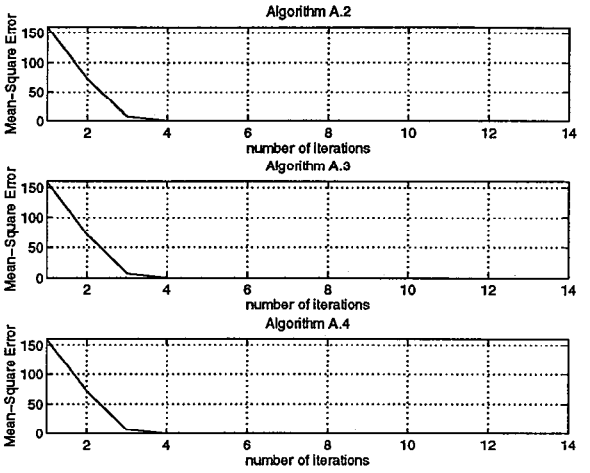

Figure 4: Performance of the Algorithms A.2, A.3 and A.4 for the case of function $F_4(\cdot)$.

$2.5), R(x-2.5), R(x+3), R(x-3), R(x+3.5), R(x-3.5),\}$ where $R(\cdot)$ is as follows

$$S(x) = -(\frac{2}{1+e^{-15x}} - 1)(\frac{2}{1+e^{-15x}} - 1) + 1$$

It can be easily seen that the regressor terms belonging to $\underline{L}$ are mutually pseudo-orthogonal. The function used in simulations was the following:

$$F_4(x) = R(x) - 0.9R(x+0.5) + 0.6R(x-0.5) + 0.7R(x+1) + 0.2R(x+2.5)$$

In figure 4 the reader can see the results of the simulations. All three Algorithms converge to the optimal network and, moreover, all three Algorithms behave the same, in the sense that at each iteration they chose the same regressor term, and they calculate the same values for the weight and the Mean-Square Error at this iteration.

References

[1] N.E. Cotter, "The Stone-Weierstrass theorem and its application to neural networks," *IEEE Trans. on Neural Networks*, vol. 1, no. 4, pp. 290-295, 1990.

[2] N. J. Dimopoulos, D. Radvan, and W.A. Keddy, "Learning in asymptotically behaving neural networks," *Proc. of the 1990 Int. Conf. Neural Networks*, San Diego, CA, vol. III, pp. 233-238, June 1990.

[3] B. Igelnik and Y.H. Pao, "Stochastic choice of basis functions in adaptive function approximation and the functional link net," *IEEE Transactions on Neural Networks*, vol. 6, no. 6, pp. 1320-1329, 1995.

[4] E. B. Kosmatopoulos, M. M. Polycarpou, M. A. Christodoulou, and P. A. Ioannou, "High-order neural network structures for identification of dynamical systems," *IEEE Transactions on Neural Networks*, vol. 6, no. 2, pp. 422-431, March 1995.

[5] E. B. Kosmatopoulos and M. A. Christodoulou, "Structural properties of gradient recurrent high-order neural networks," *IEEE Transactions on Circuits And Systems—II: Analog and Digital Signal Processing*, vol. 42, no. 9, September 1995.

[6] M.M. Polycarpou and P.A. Ioannou, "Identification and control of nonlinear systems using neural network models: design and stability analysis," *Tech. Rep. 91-09-01*, Univ. of Southern Cal., Los Angeles, September, 1991.