Pergamon

Nonlinear Analysis, Theory, Methods & Applications, Vol. 30, No. 3, pp. 1343–1354, 1997
Proc. 2nd World Congress of Nonlinear Analysts

0362-546X/97 $17.00 + 0.00

PII: S0362-546X(97)00223-X

ON FUNCTION RECOVERY BY NEURAL NETWORKS BASED ON ORTHOGONAL EXPANSIONS

E. RAFAJŁOWICZ† and m. PAWLAK††

†Institute of Engineering Cybernetics, Technical University of Wrocław, Wybrzeże Wyspiańskiego 27, 50 370 Wrocław, Poland

†† Department of Electrical and Computer Engineering, The University of Manitoba, Winnipeg, Manitoba, Canada

Keywords and phrases: function recovery, noisy data, neural networks, orthogonal series, Kronecker products, least squares, stepwise regression, consistency, pruning, net structure selection

1. INTRODUCTION

Applications of neural nets to the problem of approximating unknown functions is now rapidly expanding. The following main streams of research in this area can be distinguished:

A) approximation by the sigmoidal type nets with one hidden layer (S-nets)

B) approximation by the radial basis functions (RBF-nets)

C) polynomial type nets (P-nets)

D) nets originated from orthogonal wavelets (W-nets).

The number of papers on the above topics grows exponentially, so below we provide only selected recent contributions, where references to earlier original results can be found. In [15] [26], [11], [1] [2], [17] the reader may find results on approximation properties of the nets, while in papers [7], [25], [16], [6] one can find discussions on neural nets properties as viewed from the statistical point of view. Finally, in [22] and [12] the problem of selecting the net architecture, suitable for function recovery, is discussed.

Many fundamental results concerning approximation abilities of the above types of nets have been obtained. It seems, however, that we are far from the full understanding similarities and differences between all the above mentioned types of nets. Our aim in this paper is to discuss neural networks, which are based on orthogonal expansions (OE-nets) of unknown functions.

Due to the results of Donoho and Johnstone [7] one can look at S-nets and RBF-nets in an unifying manner, using orthogonal expansions as a mathematical tool.

It is also clear that P-nets can be analyzed by orthogonal expansions, while W-nets directly fall to this class. In this context, it seems desirable to consider a net architecture, which directly reflects orthogonal expansions.

We should add, that the net architecture proposed here is not intended to mimic any biological neural net. It can be treated as a tool for analyzing other networks, which are closer to biological counterparts. On the other hand, we hope that the proposed net structure can be hardwired in the future, providing a useful tool for engineering applications. The second reason, for which we propose to construct OE-nets is in well known difficulties in learning S- and RBF-nets, which manifests in training process, which usually needs hundreds of epochs. In contrary, in learning OE-nets one can use classical results from the theory of least-squares.

We put emphasis on fast and reliable learning process, since in engineering applications the only reason for constructing a specialized net hardware for function approximation is when an unknown function changes from time to time and one needs a fast update of its current approximation.

The paper is organized as follows. In the next section we formulate a number of questions and requirements, which should influence our decision concerning a proper choice of a functional net. Then, in Section 3 the problem of constructing a net based on orthogonal expansion is stated and the net architecture is discussed. In Section 4 we consider the learning procedures, while Section 5 contains conditions for the net to be able to recover asymptotically any square integrable function observed in the presence of noise. Finally, by means of simulations, we concentrate on the fundamental questions of choosing not only the net size but also on adaptive pruning of unnecessary links, when the number of observations is finite.

2. QUALITAIVE REQUIREMENTS

In the last decade a number of fundamental results concerning approximation capabilities of feedforeward nets have been established (see Introduction). It should be realized, however, that results of these type are necessary but not sufficient for successful use of approximation nets for function recovery from imperfect data.

Below, we try to summarize some requirements (sometimes contradictive) and questions, which may be useful in deciding what kind of net is appropriate for the application at hand. Trying to give partial answers to the questions below, we assume the following simple model according to which the learning sequence $\{(x_i, y_i)\}_{i=1}^n$ is generated:

$$y_i = m(x_i) + \varepsilon_i, \quad i = 1, 2, \ldots, n, \tag{1}$$

where $m : R^s \to R$ is unknown "smooth" function, observed at random points x_i, while the result of observation y_i contains zero mean, finite variance, uncorrelated random errors ε_i. Furthermore, sequences $\{\varepsilon_i\}$ and $\{x_i\}$ are mutually independent.

R1) Robustness

The term robust estimation has about twenty years long history in the statistical literature. (see, e.g., [23], [13]). Applied here, in its wide understanding, it means that a robust net, to some extent, is insensitive to changes of assumptions under which the network was designed and trained. In particular, robustness can be required with respect assumptions imposed on: smoothness of $m(.)$, random errors $\{\varepsilon_i\}$, input sequence $\{x_i\}$. We discuss briefly each of the above.

a) Smoothness of unknown function. Continuity and/or square integrability of $m(.)$ is usually required in any discussion concerning functional networks. These are really minimal requirements, but one should remember that in C_0 and L_2 one can met fractal-like monsters, which are nowhere differentiable. Thus, a net which is able to approximate to arbitrary accuracy any function from C_0 (L_2) is robust with respect to the smoothness assumptions, but at the same time this net is able to approximate equally well a brownian random noise.

On the other hand, we rarely know for sure that $m(.)$ is k-times differentiable ($m \in C_k$). Thus, it would be desirable to have networks, which are adaptive with respect to k, i.e., performing "equally well" independently of the true value of k. As far as we know, at present there are no nets adaptive in the above sense. Nevetherless, the method of local least squares (see, e.g., [5], [4], [14]) serves as an example of the estimation algorithm which is adaptive with respect to smoothness.

b) Random errors. Most of the existing training algorithms employ the least squares criterion (LSQ) as a tool for updating the net parameters. From the theory of robust estimation it is known that LSQ criterion is not robust against outliers in the learning sequence. Outliers can occur as the result of untidy data handling or, what is more difficult to detect, as a consequence of the fact that the tails of the errors distribution are much havier than that of the Gaussian one. As a remedy to outliers, criterions other than LSQ should be used, minimization of the sum of absolute errors being the most popular.

Here, we will employ LSQ as the most simple criterion.

c) Observations of independent variables. It is a common practice to assume that observations of independent variables, $\{x_i\}$ sequence, are exact, i.e., taken without errors, while the errors are present only in $\{y_i\}$ sequence. As far as we know, this aspect of robustness is rarely considered. The reader may consult [20], for the results indicating that orthogonal series expansions are to some extent robust against errors in $\{x_i\}$. On the other hand, there are no similar results on sigmoidal approximations. Furthermore, one can expect that sigmoids would amplify errors of this kind.

R2) Fast convergence rate versus shape preserving properties of approximation.

Considering a neural net as a device for function approximation, one can ask, which type of the net provides the largest decrease of an approximation error, when the net size increases. If we confine attention to the space of r-times differentiable functions with square integrable r-th derivative, H_r say, then the answer provided by the approximation theory is following, the minimal possible approximation error in L_2 norm is of order $O(N^{-r/d})$ in the worst case, where N is the number of terms in the approximant

$$m_N(x) = \sum_{k=1}^{N} a_k v_k(x) \tag{2}$$

of $m(x)$. In (2), $\{a_k\}$ denote parameters to be chosen when the basis $\{v_k\}$ is fixed, but $\|m - m_N\|$ can also be minimized with respect to all N dimensional basis. The answer on the best choice of $\{v_k\}$ is provided but the theory of Kolmogorov's N-widths and in the case of H_r the best (not necessarily unique) solution is provided by the orthogonal expansion w.r.t. trigonometric system. Thus, from the view point of minimizing $\|m - m_N\|$, one should use nets based on the trigonometric expansions. The price that we pay for this kind of optimality is in a wigly behavior of $m_N(.)$ in a vicinity of $m(.)$, even if the observations $\{y_i\}$ are noiseless. Analogous behavior can be expected when wavelets are used as the basis in (2). Their fast convergence rate may result in the frequent sign changes in $m(x) - m_N(x)$.

The other extreme is provided by approximations using Bernstein polynomials. It is well known that in one dimensional case ($s = 1$) $\|m - m_N\|$ is of order $O(1/\sqrt{N})$ if in (2) $\{v_k\}$ are chosen to be the Bernstein polynomials of the order N. Furthermore, for $s = 1$ the error rate $O(1/\sqrt{N})$ remains the same independently how smooth is $m(.)$. This slow convergence rate has however important advantages. Namely, $m_N(.)$ retains shape features of m, i.e., $m_N(.)$ is increasing (convex) whenever $m(.)$ is increasing (convex).

Interesting intermediate case is provided by sigmoidal approximations, when $\{v_k(x)\}$ in (2) are of the form $v_k(x) = \sigma(\alpha_k^T x)$, $k = 1, 2, \ldots, N$, where $\sigma : R \to R$ is a sigmoidal function, while $\alpha_k \in R^d$ are vectors to be chosen using the learning sequence. Recently, Barron [2] proved that for sigmoidal approximation $\|m - m_N\| = O(N^{-1})$, independently on dimension $d \geq 1$. It is not clear, at least to the authors, where "curse of dimensionality" is hidden in this case. However, the rate $O(N^{-1})$ - being intermediate between $O(N^{-r/d})$, $r > d$ and $O(N^{-1/2})$ - may be an indicator of intermediate behavior of sigmoidal approximations, between wigly trigonometric approximations and slowly convergent, but shape preserving Bernstein polynomials. This is the topic for further research.

For robustness aspects of learning the reader may consult [3].

R3) An interplay between net structure and ease of learning

Popularity of sigmoidal nets comes from the simplicity of their structure - a net with only one hidden layer suffices to approximate any continuous function. The price paid for this simplicity is in long learning process, which is necessary, since part of the tuning parameters enter nonlinearly into the net input-output relationship. Analogously, in RBF-nets the dependence between the output and the centers of basis functions is nonlinear, what results in learning, which can take hundreds of epochs. On the other extreme we have OE-nets, where the net structure is more complicated, as

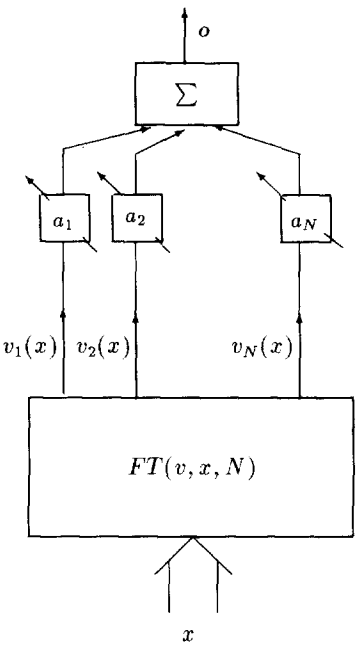

Figure 1: OE-net structure. FT - functional transformation unit, e.g., by Legendre polynomials. $\otimes$ - Kronecker products units.

we shall see later, but the dependence of the output on tuning parameters is linear, or quasilinear, if one takes the choice of the net structure into account. The input-output relationship is clearly nonlinear in OE-nets, but inputs are transformed nonlinearly by given functions, e.g., polynomials or trigonometric functions. Linearity in parameters allows for relatively easy and reliable learning procedures, which are based on the direct minimization of quadratic criterions. Furthermore, the learning process can be recurrent, in time of data acquisition and simultaneously it is one pass learning. The last feature comes from the fact that least squares estimation procedures are of the projection type.

For the above reasons we choose OE-nets as the main topic of our paper, but we shall also mention similarities and differences to other types of nets.

3. NET STRUCTURE

In the multivariate case the net structure proposed here is shown in Fig. 1.

The net performs the following transformation

$$y = a^T \cdot \bigotimes_{j=1}^{s} \bar{\phi}(x^{(j)}), \tag{3}$$

where $\bigotimes$ denotes the Kronecker product of vectors $\bar{\phi}(t) = [\phi^{(1)}(t), \phi^{(2)}(t), \ldots \phi^{(l)}(t)]^T$, $t \in [-1, 1]$, which consists of orthogonal functions, e.g., Legendre polynomials or the trigonometric system. Note that $N = dim(\bigotimes_{j=1}^{s} \bar{\phi}(x^{(j)})) = l^s$. Thus, the column vector of weights also has l^s elements.

The same group of transformation units $\bar{\phi}(.)$ is repeated s times, what makes a hardware implementation easier. Also $\otimes$ unit can be decomposed into a tree of Kronecker products, as one can deduce from the representation

$$\bigotimes_{j=1}^{q} \bar{\phi}(x^{(j)}) = \bigotimes_{j=1}^{q-1} \bar{\phi}(x^{(j)}) \otimes \bar{\phi}(x^{(q)}), \quad q = 2, 3, \ldots s. \tag{4}$$

$\otimes$ operation can be hardwired.

The main disadvantage, in comparison to S-nets, is that the number of weights gros exponentially. This drawback is compensated by the linear dependece of y on a. Furthermore, below we propose an agressive algorithm of pruning, which drasticly reduces the number of estimated parameters to those, which are necessary to represent adequately the underlying function.

4. LEARNING ALGORITHMS - ASYMPTOTIC RESULTS

Up to now particular properties of the orthogonal sequence $\{\phi^{(j)}\}$ and input observations $\{x_i\}$ were not specified. Here some detalisation is necessary. For simplicity we assume that $\{x_i\}$ are independent, identically distributed (i.i.d.) random variables (r.v's), which are uniformly distributed in the cube $[-1,1]^s$. The system $\{\phi^{(j)}\}_{j=1}^{\infty}$ is orthonormal and complete in $L_2(-1,1)$ with respect to the uniform distribution on $[-1,1]$. (If one wish to use orthonormality w.r.t. the Lebesgue measure, one should put the multiplier 2 at every occurence of $\{\phi^{(j)}\}_{j=1}^{\infty}$. In particular, this system is used to form $v_k(x)$ below and then, one should use the multiplier 2^s.)

Remark. *If the input sequence $\{x_i\}$ is not uniformly distributed, then one can choose $\{\phi^{(j)}\}_{j=1}^{\infty}$, which is orthonormal with the weight corresponding to the probabiblity density function (pdf) of $\{x_i\}$. For example, if $\{x_i\}$ are normally distributed, then a natural choice for $\{\phi^{(j)}\}_{j=1}^{\infty}$ is the system of Hermite polynomials.*

When pdf of $\{x_i\}$ is not known, one can apply the Forsythe algorithm [10] to orthogonalize polynomials in the summation sense over $\{x_i\}$.

A natural choice of the learning quality index (criterion) is the mean square error (MSE). At least two different approaches can be considered, which led to different learning algorithms. We outline them briefly below. They differ in that the operation of a criterion minimization and the substitution of discrete data are interchanged.

A) Discretization at the end. For simplicity of formulas, we identify $v(x)$ (with consequtively numbered components $v_k(x)$, $k = 1, 2, \ldots N$) and $\bigotimes_{j=1}^{s} \bar{\phi}(x^{(j)})$. MSE between $m(x)$ and the net output is given by

$$Q_N(a) = \int_{\mathcal{X}} (m(x) - a^T v(x))^2 dx, \tag{5}$$

where $\mathcal{X} = [-1,1]^s$. Due to the orthonormality of $v_k(x)$, $k = 1, 2, \ldots N$) the minimization of (5) leads to

$$a_k^* = \int_{\mathcal{X}} m(x) v_k(x) dx, \quad k = 1, 2, \ldots N. \tag{6}$$

As the final stage, we perform discretization of (6). Noticing that $a_k^* = E(m(X_1)v_k(X_1))$, we approximate a_k^* by the following algorithm, further called Algorithm A),

$$\hat{a}_k = n^{-1} \sum_{i=1}^{n} y_i v_k(x_i), \quad k = 1, 2, \ldots N. \tag{7}$$

Note that all the weights are calculated separately and the calculations can be made reqursively w.r.t. n.

B) Discretization at the beginning. The starting point is again (5), which can be rewritten as

$$Q_N(a) = E(m(x_1) - a^T v(x_1))^2 = \left(E(y - a^T v(x))\right)^2 + var(\epsilon) \tag{8}$$

due to uniform distributions of the input patterns. A natural way of estimating $Q_N(a)$ from the data is to replace the expectation by the empirical mean, what leads to the well known least squares criterion

$$q_N(a) = \sum_{i=1}^{n} (y_i - a^T v(x_i))^2, \tag{9}$$

which is minimized by $\tilde{a} \in R^N$, being a solution of the set of the normal equations

$$G_N a = n^{-1} \sum_{i=1}^{n} y_i v(x_i), \tag{10}$$

where $N \times N$ matrix G_N is defined as $G_N = n^{-1} \sum_{i=1}^{n} v(x_i) v^T(x_i)$. The reconstruction algorithm is therefore $\tilde{m}(x) = \tilde{a}^T v(x)$. We comment on solving (10) later.

Remark It should be stressed that our problem differs from the classical theory of least squares in two aspects. Firstly, observations $\{x_i\}$ are random, what results in randomness of G_N. Secondly, N is not fixed and our aim is to choose it as well as the net structure, i.e., to decide which elements in $a \in R^N$ are zero and to the estimate the rest of them, what makes our problem nonlinear.

5. CONSISTENCY OF ALGORITHMS A) AND B)

Below, we summarize briefly, referring to [18] for details, necessary and sufficient conditions for consistency of Algorithm A). We relate the number of terms N to the length n of the learning sequence. In fact, a sequence $N = N(n)$ can not be arbitrary, as it follows from the following result.

Theorem 1 *Let for a certain constants $0 < v \leq V < \infty$, $v \leq E\epsilon_i^2 \leq V$. Then, the following conditions*

$$N(n) \to \infty, \qquad N(n)/n \to \infty \qquad as \quad n \to \infty \tag{11}$$

are both necessary and sufficient for $\int_{\mathcal{X}} E(m(x) - \hat{a}^T \cdot v(x))^2 dx \to \infty$ as $n \to \infty$ for every $m \in L_2(\mathcal{X})$.

As we shall see below, expanding the net size is also fundamental for consistency of Algorithm B).

Theorem 2 *Let $N(n) \to \infty$ in such a way that*

$$(N^2(n)/n) \log \log n \to 0, \qquad as \qquad n \to \infty, \tag{12}$$

then $\int_{\mathcal{X}} E(m(x) - \tilde{a}^T \cdot v(x))^2 dx \to 0$ as $n \to \infty$ for every $m \in L_2(\mathcal{X})$, provided that the system $\{\phi^{(j)}\}$ is complete in $L_2(0,1)$ and its elements are commonly bounded, i.e., $|\phi^{(j)}(x)| \leq B$ for a crtain $B > 0$.

Proof. To prove the theorem we can use essential parts of the proof of Thm. 1 from [19]. The main difference between the case considered here and in [19] is in that regressors are random here. One can, however, replace all the expectations in [19] by their conditional, w.r.t. $\{x_i\}$, counterparts, what gives upper bounds for $\int_{\mathcal{X}} E(m(x) - \tilde{a}^T \cdot v(x))^2 dx$ in the case considered here. The only more subtle point is in checking whether the condition (Condition 3, Thm. 1 in [19])

$$N(n) \cdot \beta_n \to 0 \qquad \text{as} \qquad n \to \infty \tag{13}$$

holds with the probability 1 (w. P. 1). In (13) β_n is defined as

$$\beta_n = \max_{k,k',j} \left| n^{-1} \sum_{i=1}^{n} \gamma_i^{(kk'j)} \right| \tag{14}$$

where, for $\delta_{kk'}$ being the Kronecker delta,

$$\gamma_i^{(kk'j)} \stackrel{def}{=} \delta_{kk'} - \phi^{(k)}(x_i^{(j)})\phi^{(k')}(x_i^{(j)}). \tag{15}$$

Note that $E\gamma_i^{(kk'j)} = 0$ and for fixed k, k' and j, $\gamma_i^{(kk'j)}$ are i.i.d r.v's with the variances

$$\mathrm{var}(\gamma_i^{(kk'j)}) \leq 2(1 + B^2). \tag{16}$$

Thus, according to the law of iterated logarithm (see, e.g., [24]) we conclude that the set of cluster points of the sequences

$$(n \log\log n)^{-1} \sum_{i=1}^{n} \gamma_i^{(kk'j)} \tag{17}$$

are commonly bounded w. P. 1. Hence, if we assure that $N(n)\sqrt{(\log\log n)/n} \to 0$ or, equivalently, (12) hold, then condition (13) also hold, what finishes the proof.

Remark. If we take $c_1/n^\beta \leq N(n) \leq c_2/n^\beta$, $0 < c_1 < c_2 < \infty$, then the condition in (13) holds for $0 < \beta \leq 1/2 - \epsilon$, where $\epsilon > 0$ can be arbitrarily small.

Remark. Mimicking the proof of Corollary 2 in [19], one can prove that if $\{\phi^{(k)}\}$ is the trigonometric system, m is periodic and has all the derivatives of the order $p \geq 2$ Lipschitz continuous with the exponent $\alpha > 0$, then $E||m - \tilde{m}||^2$ attains the best possible convergence rate $O(n^{-\nu})$, $\nu \stackrel{def}{=} 2\mu/(2\mu + s)$, $\mu \stackrel{def}{=} p + \alpha$ (recall that $s = \dim(x)$). Note however that, unlike the fixed design case [19], we can assure the optimal convergence rate for smooth functions only.

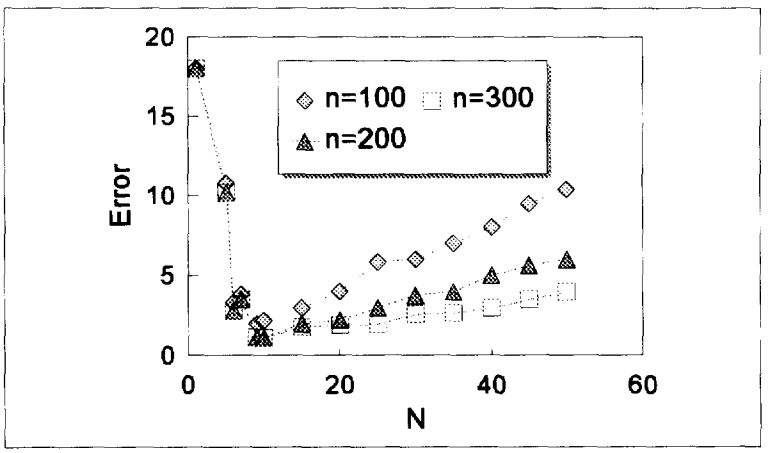

Figure 2: Dependence of MISE on N obtained from simulated data (details in the text).

6. SIMULATIONS AND THE CHOICE OF THE NET STRUCTURE

In this section we summarize results of simulations and provide indications concerning the choice of N for training sequence of a finite length. Finally, some indications for the choice of the net structure will be given.

Optimal net size. As a vehicle for presening dependence of MISE on N we choose Algorithm A), since the pattern is qualitatively the same for Algorithm B). As one can expect, too small values of

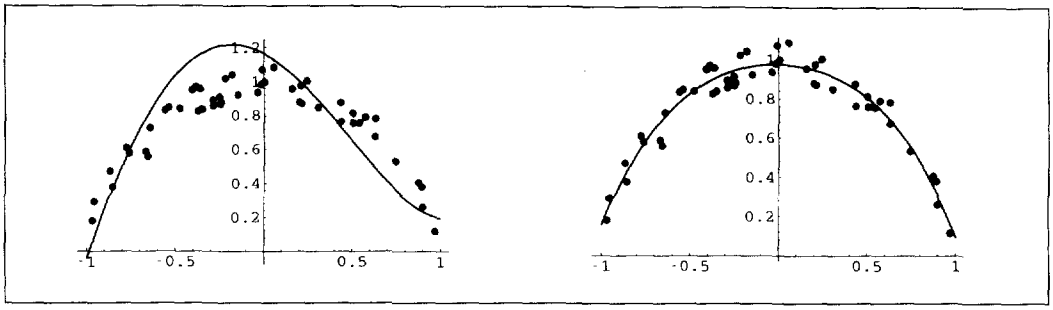

Figure 3: Comparison of learning Algorithms A) (left plot) and B) (right plot) for $N = 5$ (see details in the text).

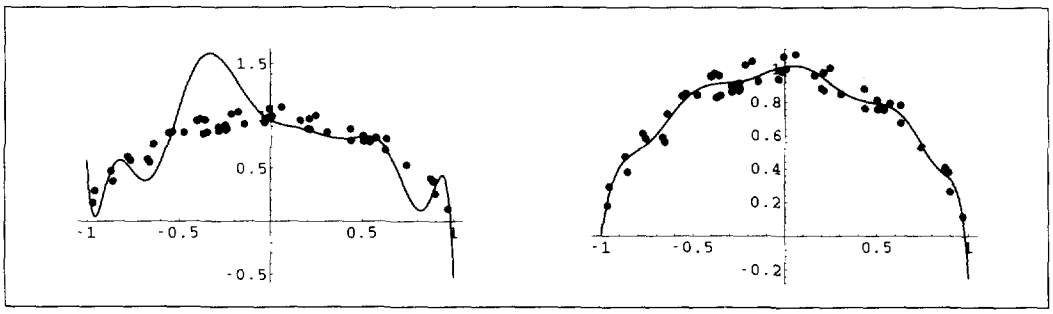

Figure 4: Comparison of learning Algorithms A) (left plot) and B) (right plot) for $N = 12$ (see details in the text).

N introduce the bias ($\int_{\mathcal{X}}(m(x) - E\hat{m}(x))^2 dx$) is too large. On the other hand, too large values of N, although reduce bias, are also not recommended, since the variance of $\hat{m}$ grows with N. Thus, existence of optimal N^*, which depends on the noise variance, n and smoothness of m is to be expected. Typical behaviour of MISE on N is shown in Fig. 2, which was obtained by simulating the training sequence from the function $m(x) = 4\sin^3(2\pi x^3)$, with the signal to noise ratio eqal 5. Then, the data generation process were and Algorithm A) run 30 times for each N in order to evaluate MISE. Existence of the optimal value is clearly visible. One can notice that MISE is not too sensitive to small under- or overestimation of N^*, while larger errors in the choice of N lead to essential increase of MISE.

Comparison of Algorithms A) and B). Here, our aim is to compare performance of Algorithms A) and B) when the same learning sequence is fed to the each algorithm only once (see pairs of plots in Fig. 3, 4).

In Fig. 3, 4 data were generated from the function $m(x) == \sqrt{\sin(\pi x)/(\pi x)}$ at $n = 50$ points, while in Fig. 5, 6 the following bivariate test function was used

$$m(x) = \sin(1.5\pi x^{(1)} \cdot x^{(2)}) \tag{18}$$

and the length of the learning sequence was chosen to be 400. Random noises with the uniform distributions were added to the observations. The range of errors being chosen at the level of about

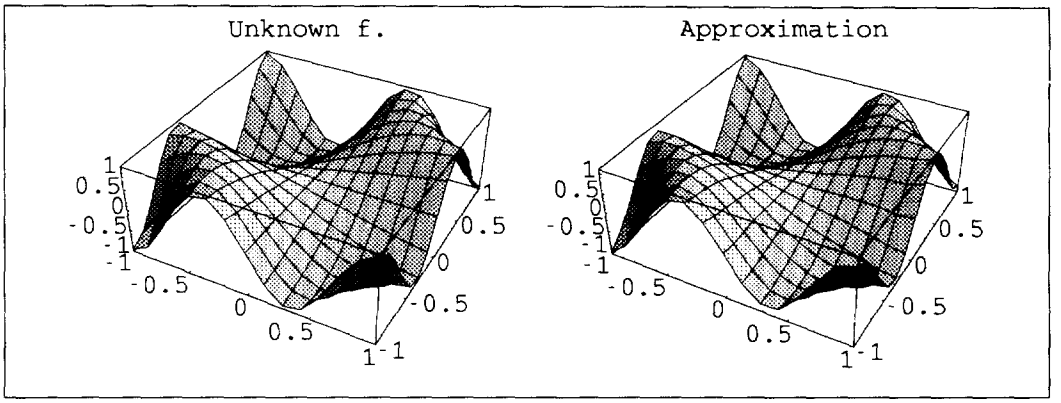

Figure 6: Result of learning by Algorithm B) for the net size $N = 8 \times 8$.

10% of the maximum of $|m(x)|$. In all the considered cases the Legendre polynomials were used as the $\phi(t)$ vector.

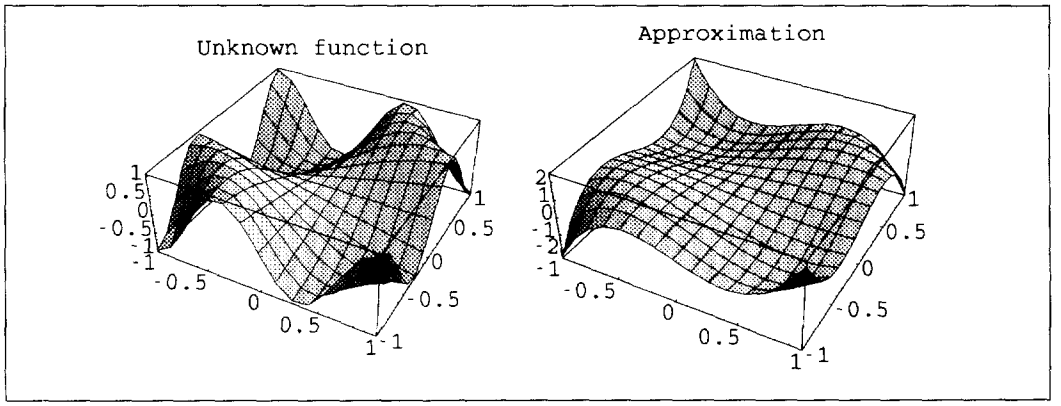

Figure 5: Result of learning by Algorithm B) for the net size $N = 4 \times 4$.

Prunning of the net. The above visible advantages of Algorithm B) are overshadowed by its large computational complexity. Although the computational burden can be drastically reduced by exploiting the Kronecker product structure of $v(x)$ (see [21]) it still remains relatively large in comparison to algorithm A).

The computing time factor become critical when we try to select not only net size N but also its structure, i.e., to decide which weights should be set to zero. As it is known (see, e.g., [9]), data based algorithms of rejecting unnecessary terms have been deviced in the theory of linear regression. Their use for the net prunning is rather problematic, since they require recalculating all the weights (regression coefficients) after rejecting each single term. On the other hand, rejection of unnecassary terms is highly desirable from the view point of estimation accuracy, since their presence increases essentially the estimation variance.

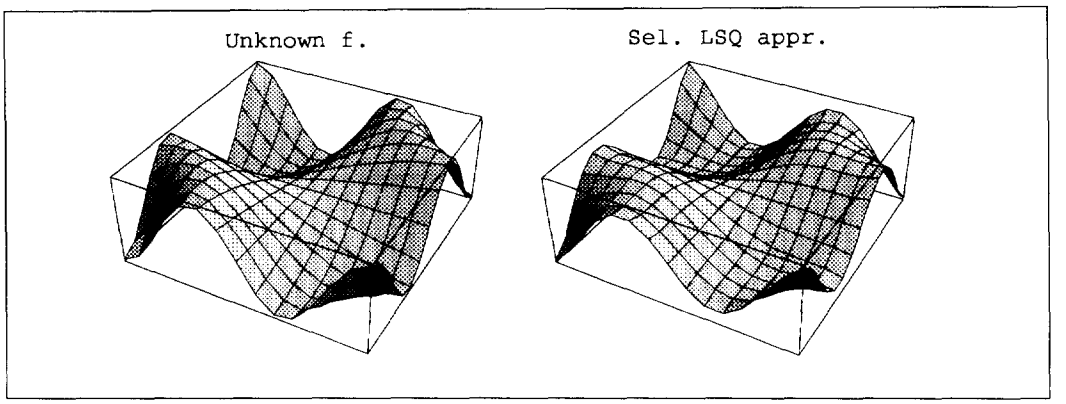

Figure 7: Performance of learning Algorithm C).

For the above reasons we propose the following, two pass algorithm for the net prunning, which can reject several unnecessary terms at once and combines adavntages of Algorithms A) and B).

Algorithm C)

Step 1) Calculate the net weights $\hat{a}$ according to Algorithm A).

Step 2) Check the condition

$$\frac{n}{\hat{\sigma}^2}\left(\hat{a}^{(k)}\right)^2 > 2\nu, \qquad \text{for} \qquad k = 1, 2, \ldots N, \tag{19}$$

where ν is defined as follows $\nu = (n - N - 2)/(n - N)$, while $\hat{\sigma}^2$ is the residual sum of squares divided by $(n - N)$.

Denote by $k_1, k_2, \ldots k_L$ those indexes, for which (19) holds.

Step 3) Cut branches in the net, for which (19) is not fulfilled. In other words, form a vector $\tilde{v}(x)$ by selecting from $v(x)$ elements with the indexes $k_1, k_2, \ldots k_L$. Apply Algorithm B) to train the network spanned by $\tilde{v}(x)$.

Some comments are in order concerning Algorithm C).

1) I/O relationship, which results from Algorithm C) is the following: $\sum_{k \in P} \tilde{a}^{(k)} v_k(x)$, where P is the set of such k for which the inequality in (19) holds. Condition (19) for selecting a regression function structure has been proposed in [8] in a different context, namely, when $\{x_i\}$ are preasigned, instead of being random as in our case.

2) Condition (19) leads to more severe pruning than the familiar Akaike's criterion, in which ν is replaced by 1.

3) Algorithm C) is in fact nonlinear w.r.t. $\{y_i\}$, but nonlinear thresholding elements appear before the output node, instead after it, as in S-networks.

Performance of Algorithm C) is illustrated in Fig. 7 (data were generated as above). The starting point was the net spanned by $N = 9 \times 9$ Legendre polynomials. After pruning terms with the following numbers $11, 13, 15, 17, 29, 31, 35, 47, 51, 53, 65, 67, 69, 71$ were retained (with successively numbered terms, according to the Kronecker product ordering). For comparison, the same data were fed into Algorithm A) and the resulting appriximation is shown in Fig. 8.

Considerable increase of accuracy is visible from comparison of Fig. 8 and 7. Although more extensive comparisons are desirable, Algorithm C) can be recommended as having moderate compu-

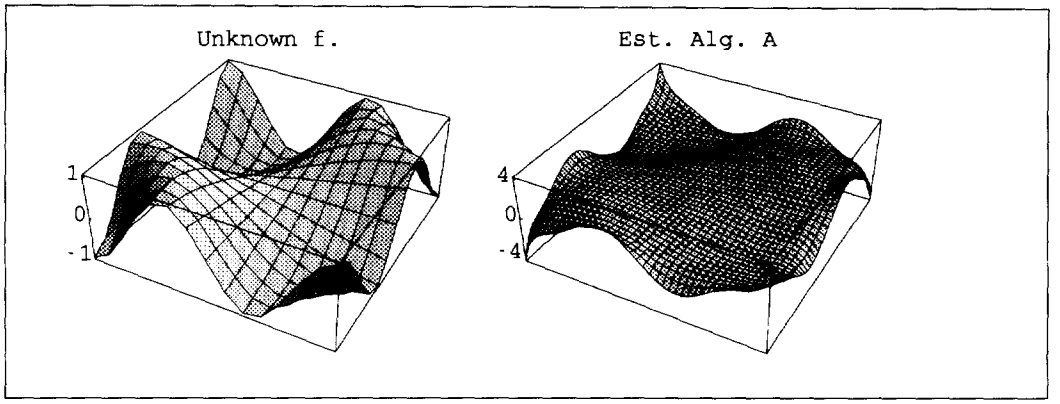

Figure 8: Learning without pruning, i.e., using Algorithm A) and the same data as in Fig. 7.

tational complexity and high estimation accuracy.

References

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

[2] Barron A.R. Approximation and estimation bounds for artificial neural networks. *Machine Learning*, 14:115-133, 1994.

[3] Chen D.S. and Jain R.C., A robust back propagation learning algorithm for function approximation. *IEEE Trans. on Neural Networks, 6:467-479, 1994.*

[4] Cleveland W. S. Robust locally weigthed regression and smoothing scatterplots. *J. Amer. Staist. Assoc.*, pages 998-1004, 1979.

[5] Cleveland W. S. and Devlin S.J. Locally weighted regression: An approach to regression analysis by local fitting. *J. Amer. Statist. Assoc.*, 83:596-610, 1988.

[6] Sánchez Pastor M.S., Buydens L.M.C. ,Derks, E.P.P.A. Robustness analysis of radial base function and multi-layered feed-forward neural network models. *Chemometrics and Intelligent Laboratory Systems*, 28:49-60, 1995.

[7] Donoho D.L. and Johnstone I.M. Projection-based approximation and a duality with kernel methods. *Annals of Statistics*, 17(1):58-106, 1989.

[8] Droge B. and Georg T. On selecting the smoothing parameter of least squares regression estimates using the minimax regret approach. *Statistics & Decisions*, 13:1-20, 1995.

[9] Seber G. A. F. *Linear regression Analysis.* Wiley, New York, 1977.

[10] FORSYTHE G. E. Generation and use of orthogonal polynomials for data-fitting with a digital computer. *J. Soc. Industr. and Appl. Math.*, 5:74–88, 1956.

[11] GIROSI F. AND POGGIO T. Networks and the best approximation property. *Biological Cybernetics*, 63:169–176, 1990.

[12] JONES M., POGGIO T., GIROSI, F. Regularization theory and neural networks architectures. *Neural Computation*, 7:219–269, 1995.

[13] HAMPEL ET ALL. *Robust statistics.* Wiley, New York, 1986.

[14] HASTIE T. AND LOADER C. Local regression: Automatic kernel carpentry. *Statistical Science*, 8:120–143, 1993.

[15] STINCHCOMBE M., HORNIK K., WHITE H. Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks. *Neural Networks*, 3:551–560, 1990.

[16] JONES L. K. A simple lemma on greedy approximation in hilbert space and convergence rates for projection pursuit regression and neural network training. *Annals of Statistics*, 20(1):608–613, 1992.

[17] LIN V.YA., PINKUS A., SCHOCKEN S., LESHNO M. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural Networks*, 6:861–867, 1993.

[18] PAWLAK M. AND RAFAJŁOWICZ E. Slowly expanding orthogonal neural net for uniformly distributed input patterns. part i, ii.

[19] RAFAJŁOWICZ E. Nonparametric least squares estimation of a regression function. *Statistics*, 19:349 – 358, 1988.

[20] RAFAJŁOWICZ E. Nonparametric identification with errors in independent variables. *Int. J. Syst. Sci.*, 51:279–292, 1994.

[21] RAFAJŁOWICZ E. AND MYSZKA W. Efficient algorithm for a class of least squares estimation problems. *IEEE Trans. Aut. Control*, AC-39, No 4:June, 1994.

[22] RIPLEY B. Statistical ideas for selecting network architectures. Preprinty.

[23] ROUSSEEUW P. J. AND LEROY A. M. *Robust regression and Outlier detection.* Wiley, New York, 1987.

[24] SERFLING R. *Approximation Theorems of Mathematical Statistics.* John Wiley, New York, 1980.

[25] SPECHT D. A general regression neural network. *IEEE Transactions on Neural Networks,*, 2(6):568–576, 1991.

[26] WHITE H. Connectionist nonparametric regression: Multilayer feedforward networks can learn arbitrary mappings. *Neural Networks*, 3:535–549, 1990.