A New Incremental Method for Function Approximation using Feed-forward Neural Networks

Enrique Romero and René Alquézar
Departament de Llenguatges i Sistemes Informàtics, Universitat Politècnica de Catalunya

Abstract - A sequential method for approximating vectors in Hilbert spaces, called *Sequential Approximation with Optimal Coefficients and Interacting Frequencies* (*SAOCIF*), is presented. *SAOCIF* combines two key ideas. The first one is the optimization of the coefficients (the linear part of the approximation). The second one is the flexibility to choose the frequencies (the non-linear part). The approximations defined by *SAOCIF* maintain orthogonal-like properties. The theoretical results obtained prove that, under reasonable conditions, the residue of the approximation obtained with *SAOCIF* (in the limit) is the best one that can be obtained with any subset of the given set of vectors. In the particular case of L^2, it can be applied to approximations by algebraic polynomials, Fourier series, wavelets and feed-forward neural networks, among others. Also, a particular algorithm with feed-forward neural networks is presented. The method combines the locality of sequential approximations, where only one frequency is found at every step, with the globality of non-sequential ones, where every frequency interacts with the others. Experimental results show a very satisfactory performance.

I. Introduction

Vector approximation in Hilbert spaces is present in different areas, such as Statistics, Signal Processing or Artificial Intelligence. In most cases, the Hilbert space of interest is L^2, where the vector f is a square integrable function defined on a subset of $\mathbb{R}^I$, that we want to approximate by linear combinations of simpler functions. Linear expansions in a single basis are not flexible enough. The information can be diluted across the whole basis [18], and the approximation error cannot be made smaller than $O(1/(d\sqrt[d]{n}))$, where d is the dimension of the input to the function [2]. This happens even with an orthogonal basis. An attractive way to construct an approximation is, starting from scratch, adding terms one at a time to the partial approximations, until the desired approximation accuracy is achieved. This is the aim of sequential (or incremental) methods. Most of the existing methods choose the new term so that it matches the previous residue as best as possible (see Section V). As it is well known, although this strategy leads to approximations convergent towards the target function, it may be far from being the best strategy. Trying to approximate the residue does not take into account the interactions with the previous selected terms, even with optimal coefficients.

This work was supported by Consejo Interministerial de Ciencia y Tecnología (CICYT), under project TAP1999-0747

In this paper we present a general sequential method for function approximation, named *SAOCIF*, that takes into account these problems. On the one hand, it optimizes the coefficients, so that we always achieve the best approximation with the selected vectors. On the other, the vectors can be selected at every step in a flexible manner, taking into account the interactions with the previous terms. A particular algorithm with neural networks is also presented. Experimental results show a very satisfactory performance.

The paper is organized as follows. The definition of *SAOCIF* and the main properties are explained in Section II. The particular algorithm using neural networks is presented in Section III. The experimental results are shown in Section IV. An overview of the related work is presented in Section V. Finally, some conclusions are drawn in Section VI.

II. Definition of *SAOCIF* and Main properties

A more extended discussion, and the proofs of the theoretical results listed in this section can be found in [23].

A. Definition

The problem of approximation in Hilbert spaces that we will deal with in this paper can be defined as follows: "Let H be a Hilbert space with inner product $\langle , \rangle : H \times H \rightarrow \mathbb{C}$, a space of parameters Ω, and $f \in H$ a vector to approximate with vectors $v_\omega = v(\omega)$, $v : \Omega \rightarrow H$, $\omega \in \Omega$, such that $\forall \omega \in \Omega \quad \|v_\omega\| \neq 0$. We want to find $\omega_1, \omega_2, \cdots \in \Omega$ and $\lambda_1, \lambda_2, \cdots \in \mathbb{C}$ such that $\lim_{N \rightarrow \infty} \left\| f - \sum_{k=1}^{N} \lambda_k v_{\omega_k} \right\| = 0$. As commonly used, the term *frequency* refers to every $\omega_1, \omega_2, \cdots \in \Omega$, and *coefficient* to every $\lambda_1, \lambda_2, \cdots \in \mathbb{C}$ ". This definition is, in essence, the traditional one in approximation of vectors in Hilbert spaces. In L^2, usually, $\Omega \subseteq \mathbb{C}^p$.

Definition. A *Sequential Approximation with Optimal Coefficients and Interacting Frequencies* (*SAOCIF*) is a sequence of vectors $\{X_N\}_{N\geqslant 0} \in H$, which terms are defined as:

1. $X_0 = 0$.
2. $X_N = \sum_{k=1}^{N-1} \lambda_k^N v_{\omega_k} + \lambda_N^N v_{\omega_N}$, so that
   (a) The coefficients are optimal. That is, the vector X_N is the best approximation of f with vectors $v_{\omega_1}, \cdots, v_{\omega_{N-1}}, v_{\omega_N}$.
   (b) For every $\mu \in \mathbb{C}$ and every $\omega_0 \in \Omega$ we have $\|f - X_N\|^2 \leqslant \|f - (X_{N-1} + \mu v_{\omega_0})\|^2 + \alpha_N$, where $\alpha_N \geqslant 0$. That is, the approximation of f with X_N is better (up to α_N) than the best approximation of the residue $f - X_{N-1}$ that one could achieve with only one vector $v_{\omega_0} \in v(\Omega)$ (or, equivalently, keeping fixed the coefficients of X_{N-1}).

Remarks.

1. At step N, a new frequency (ω_N) is considered, the number of terms of the approximation is increased by one ($\lambda_N^N v_{\omega_N}$), and the coefficients $\lambda_1^N, \lambda_2^N, \cdots, \lambda_{N-1}^N$ are recalculated in order to obtain the best approximation of f with $v_{\omega_1}, \cdots, v_{\omega_{N-1}}, v_{\omega_N}$. The frequencies $\omega_1, \omega_2, \cdots, \omega_{N-1}$ are kept fixed. The vectors $v_{\omega_1}, \cdots, v_{\omega_{N-1}}, v_{\omega_N}$ are not necessarily mutually orthogonal.
2. Observe that, if $\alpha_N > 0$, at step N there exists at least one frequency ω_N satisfying the property (b) of *SAOCIF* 's definition. Moreover, there can be infinitely many frequencies satisfying this property.
3. Since X_N is the best approximation of f with $v_{\omega_1}, \cdots, v_{\omega_{N-1}}, v_{\omega_N}$, it holds that [1]

$$A_N \cdot \left(\lambda_1^N, \ldots, \lambda_N^N\right)^T = \left(\left\langle f, v_{\omega_1}\right\rangle, \ldots, \left\langle f, v_{\omega_N}\right\rangle\right))^T \quad (1)$$

   where $A_N(i,j) = \left\langle v_{\omega_i}, v_{\omega_j}\right\rangle$. Consequently, once the frequencies $\omega_1, \cdots, \omega_{N-1}, \omega_N \in \Omega$ are fixed, the optimal coefficients $\lambda_1^N, \cdots, \lambda_{N-1}^N, \lambda_N^N \in \mathbb{C}$ can be calculated by solving (1). It can be proved easily that the system has only one solution if and only if $v_{\omega_1}, \cdots, v_{\omega_{N-1}}, v_{\omega_N}$ are linearly independent. Otherwise, the system has more than one solution. Since the frequencies $\omega_1, \omega_2, \cdots, \omega_{N-1}$ are kept fixed, the proposed system at step N is equal to the system at step $N-1$, but with a new row and a new column.
4. If $H = L^2$ and we only have a dataset X, the inner products can be approximated by

$$\left\langle g, v_{\omega_j}\right\rangle \cong \frac{1}{|X|} \sum_{x \in X} g(x)\overline{v_{\omega_j}(x)}.$$

   In this case we will suppose that the integral is defined with regard to the probability measure of the problem represented by the dataset. In addition, solving (1) is equivalent to solving the Least Squares (LS) problem associated with the dataset.

B. Theoretical properties

Among others properties (see [23]), it can be proved the following results.

Theorem 1. Let H be a Hilbert space, and $f \in H$. Any *SAOCIF* $\{X_N\}_{N\geqslant 0}$ satisfies the following properties:

(Ta) $\|f - X_N\|^2 = \|f\|^2 - \|X_N\|^2$.
(Tb) $\|X_N\|^2 = \sum_{k=1}^{N} \lambda_k^N \overline{\langle f, v_{\omega_k}\rangle}$.
(Tc) Suppose that $\lambda_N^N \neq 0$. The vector v_{ω_N} is orthogonal to the space spanned by $\{v_{\omega_1}, \cdots, v_{\omega_{N-1}}\}$ if and only if the previous existing coefficients do not change between the steps $N-1$ and N.
(Td) $\{X_N\}_{N\geqslant 0}$ is convergent in H. That is,

$$\exists g \in H \quad \lim_{N\to\infty} \|g - X_N\| = 0.$$

(Te) Suppose that $\lim_{N\to\infty} \alpha_N = 0$. Then
   (Te1) There is no subset of vectors in $v(\Omega)$ that approximate f more than g.
   (Te2) If there exists $A \subseteq \Omega$ such that the set of vectors $\{v_\psi : \psi \in A\}$ spans H (that is, its linear span is dense in H), then $\{X_N\}_{N\geqslant 0}$ converges towards f.

There is a great parallelism between these properties and those satisfied by an approximation with orthogonal vectors. Observe that by (Tc), the only directions that guarantee that, without recalculating the coefficients, the approximation is optimal are the orthogonal directions. Hence, if the approximation vectors are not mutually orthogonal, the coefficients must be recalculated. Regarding convergence, observe that these results are not very restrictive. In order to assure the convergence towards f, the family of vectors used in the construction must have the capability of approximating any vector. Hence, *SAOCIF* allows us (by selecting Ω and $v(\Omega)$) to choose any (or some) of the multiple vector families satisfying this property. The hypothesis about the tolerance α_N is the same as in [13], [2], [15] or [14].

C. Practical properties in $H = L^2$

From now on we will work in the space L^2. *SAOCIF* can be applied to a number of vector families that are very usual in the literature, since the universal approximation capability of a family of functions is enough to apply *SAOCIF* with guarantee of convergence to f:

1. Algebraic Polynomials, with $v_\omega(\vec{t}) = \Pi_{i=1}^{I} t_i^{\omega_i}$.
2. Nonharmonic Fourier Series, with $v_\omega(\vec{t}) = e^{i\vec{\omega}\cdot\vec{t}}$. If the function is real, it can be approximated by sines and cosines ($v_\omega(\vec{t}) = \sin(\vec{\omega}\cdot\vec{t})$ or $v_\omega(\vec{t}) = \cos(\vec{\omega}\cdot\vec{t})$).
3. Wavelets, with $v_\omega(t) = \phi_{u,s}(t)$, where $\phi_{u,s}(t)$ is a (translated and scaled) time-frequency atom.
4. Feed-forward Neural Networks. In particular,
   (a) *Multi-layer Perceptrons* (MLPs), with activation functions leading to the universal approximation property (see, for example, [16]).

(b) *Radial Basis Function Networks* (RBFNs), with radial basis functions with the universal approximation capability (see, for example, [19]).

As has been said before, the system (1) at step N is equal to the system at step $N-1$, but with a new row and a new column. Therefore, it can be solved efficiently in a similar fashion that a bordered system [8].

In addition, if we only have a dataset, the error $\|f - X_N\|^2$ can be computed avoiding one pass through the dataset. By (Tb), we know that $\|X_N\|^2 = \sum_{k=1}^{N} \lambda_k^N \overline{\langle f, v_{\omega_k} \rangle}$. But the values of $\{\langle f, v_{\omega_k} \rangle\}_{1 \leqslant k \leqslant N}$ are the independent vector of the linear equations system (1) just solved to obtain $\{\lambda_k^N\}_{1 \leqslant k \leqslant N}$. Hence, once the coefficients have been obtained, we can compute $\|X_N\|^2$ with cost $O(N)$. By (Ta) we have $\|f - X_N\|^2 = \|f\|^2 - \|X_N\|^2$, with $\|f\|^2$ constant. Note that the cost of computing $\|X_N\|^2$ or $\|f - X_N\|^2$ directly from the dataset is $O(T{\cdot}N)$, where T is the number of points in the dataset.

Finally, it can be easily verified that the goodness of the new frequency ω_0 with regard to its approximation capability does not depend on the norm of the vector v_{ω_0}. In particular, we could normalize the vectors defining $v'_{\omega_N} = v_{\omega_N} / \|v_{\omega_N}\|$.

III. *SAOCIF* and Feed-forward Neural Networks

From now on, we will focus on FNNs. We want to approximate a function $f : \mathbb{R}^I \to \mathbb{R}^O$ in $H = L^2(\mathbb{R}^I)$ by MLPs or RBFNs with $\Omega = \mathbb{R}^{I+1}$. We only have the value of the function in a dataset, and the main objective is to achieve a successful generalization. The dimension of the input space may be very large (of the order of hundreds) depending on the problem at hand.

In practice, *SAOCIF* presents a problem. To find a valid frequency, we must verify that the property (b) of *SAOCIF* 's definition is satisfied, which involves a global minimization problem. Global optimization techniques are very expensive computationally. In a high-dimensional space without any kind of convexity, it becames an almost intractable problem [9]. The strategy of matching the residue also presents this problem [14]. But if we are dealing with a dataset and our main aim is the generalization, finding a good local minumum is many times enough to achieve a good performance.

FNNs are a suitable approach to deal with function approximation problems when only a dataset is available, and *SAOCIF* can serve as an inspiration to construct an FNN: adding hidden units one at a time, choosing the initial weights in a flexible and (in some sense) optimal manner, so as to adjust the network until we have a satisfactory model. The resulting incremental method combines the locality of sequential approximations, where only one frequency is found at every step, with the globality of non-sequential methods, such as Backpropagation (BP), where every frequency interacts with the others. This idea, in addition, offers a number of advantages for building the network. First, it allows to construct parsimonious networks. Second, different activation functions can be chosen at every step, so that the network adapts its architecture to the specific target function. Recent results show that the use of non-sigmoidal activation functions for MLPs may lead to very promising results [24]. Finally, any strategy can be used to select the new frequencies.

Algorithm
  while the network is not valid **do**
    Increase by 1 the number of hidden units N
    for $t := 1$ **upto** *Nfrequencies* **do**
      Assign a candidate frequency $\omega_{(t)}$ (weights in the first layer) to the new hidden unit
      Pick an activation function for the new hidden unit
      Compute the coefficients $\{\mu_k(t)\}_{1 \leqslant k \leqslant N}$ (weights in the second layer), by solving (1)
      Compute the *Output* $\|X_N(t)\|^2$ with (Tb)
      Set $\omega_N := \omega_{(t)}$ if $\|X_N(t)\|^2$ is maximized
    end for
    Optionally train the network, to tune ω_N
    Fix the frequency ω_N in the network
    Validate the network
  end while
end Algorithm

Fig. 1. An algorithm to construct an FNN inspired in *SAOCIF*.

Concerning the architecture needed to construct the approximation, it must have the following characteristics:

1. It must be a feed-forward architecture with a hidden layer of units (including both MLPs with one hidden layer and RBFNs).
2. There are no restrictions about the dimension of the input and the output. With several outputs, the total inner products must be calculated as the summation of the individual inner products of every output.
3. There is no restriction about the biases in the hidden units. The biases can be treated as part of the frequencies. The output units cannot have biases.
4. There is no restriction about the activation functions in the hidden units. In particular, they can be sines, cosines, sigmoidal functions, gaussian functions, wavelets, etc. Obviously, different units may have different activation functions. The output units must have a linear activation function.

As we can see, the restrictions only refer to the output units. The biases are not a real problem, since they can be considered as frequencies with a simple transformation. Hence, the only real restriction in the output units is the linear activation function. An algorithm to construct an approximation based on *SAOCIF* using FNNs is given in figure 1. Since the frequency goodness does not depend on the norm of its associated vector, the range of weights to look for candidate frequencies may be as large

as desired. The strategy to select the candidate frequency is probably the most important part of the algorithm. In Section IV three strategies are introduced in order to test the algorithm. In the first one, the frequencies are selected at random. In the second one (Input strategy), the frequencies are selected from the points in the dataset (as often in RBFNs, but not exclusively) in a deterministic manner: for every hidden unit to be added, every point in the training set is tested as a candidate frequency. The third one is a more sophisticated strategy from the field of Evolutionary Algorithms, where a population of frequencies evolves driven by a Breeder Genetic Algorithm (BGA) [5] with the squared error as the fitness function.

IV. Experiments

We now comment some generalities about the experiments performed in orden to test the *SAOCIF* algorithm presented in Section III. There was no further training after a new hidden unit was added. In the tables of results, method '*MFT*' means "Maximum Fourier Transform" and is a version "matching the residue" of *SAOCIF*: the previuos coefficients are not recalculated, and the coefficient of the new frequency is the maximum normalized Fourier transform of the residue at every step (that is, the coefficient that minimizes the residue for the selected frequency). The column 'WR' indicates the range of weights to look for candidate frequencies. When the value 'Input' is present, it means that the candidate frequencies are selected from the points in the dataset, as explained before. Several activation functions (AF in the tables) have been tested, such as linear (lin), logistic (lgt), sine or cosine (cos) in the MLP model and gaussian (gau) in the RBFN model. The column 'Test' indicates the generalization performance obtained as the mean of the individual performances, and the column 'Com.' indicates the generalization performance obtained by an average-output committee of the resulting networks. In the column 'NH', the average number of hidden units in the resulting networks is shown. An 'NP' value means "Not Possible", indicating that the learning of the training set was unsatisfactory.

A. The *Two Spirals* Problem

The well-known *Two spirals* problem consists in identifying the points of two interlocking spirals. It is an extremely hard problem for architectures with sigmoidal activation functions because of its intrinsic high nonlinearity. Other activation functions lead to better results: Hermite polynomials [12] or periodic functions [24].

We tested this problem with different methods and activation functions. The frequencies were selected randomly (100 attempts) within a certain range of weights. The maximum number of hidden units enabled was 500, and no more hidden units were added when the whole training set was learnt. Results are shown in Table I as the average of 10 runs. The column 'Train' indicates the percentage of the training set which has been learnt. As already known, this is a very hard problem for the logistic function, but it could be learnt with *SAOCIF* and an adequate (and very large) range of weights. The number of hidden units of the obtained solutions was somewhat greater than 100. *MFT* could not solve the problem with the same ranges, at least with 500 hidden units. With sine activation functions, and an appropriate choice of the range of weights, both *SAOCIF* and *MFT* obtained very good solutions with a few number of hidden units. Note that the number of hidden units of the solutions obtained with *SAOCIF* is always smaller than the number obtained with *MFT*, as expected. Figure 2 compares the generalization obtained by the Cascade-Correlation algorithm (see Section V) and *SAOCIF* with logistic and sine activation functions. It is worth knowing that all points in both the training and test sets are radially equidistant inside a disk of radius 6.5.

Method	AF	WR	NH	Train	Test	Com.
SAOCIF	lgt	[-16,+16]	105.6	100%	93.5%	100%
SAOCIF	lgt	[-12,+12]	108.9	100%	92.5%	99.0%
SAOCIF	lgt	[-8,+8]	NP	-	-	-
MFT	lgt	[-16,+16]	500	85.4%	84.5%	84.9%
MFT	lgt	[-12,+12]	NP	-	-	-
SAOCIF	sine	[-3,+3]	13.3	100%	98.8%	100%
SAOCIF	sine	[-2,+2]	NP	-	-	-
MFT	sine	[-3,+3]	16.5	100%	98.5%	99.5%
MFT	sine	[-2,+2]	NP	-	-	-

TABLE I
RESULTS FOR THE *Two spirals* WITH THE RANDOM STRATEGY.

Fig. 2. Generalization obtained by Cascade-Correlation (left) and *SAOCIF* with logistic [-16,+16] (middle) and sine [-3,+3] (right) functions respectively for the *Two spirals* problem. The results for *SAOCIF* (with corners $(-6.5,-6.5)$ and $(6.5,6.5)$) are the committee results.

B. *Diabetes*

We performed another comparison with a problem of medical diagnosis, using the "Pima Indians Diabetes Database" produced in the Applied Physics Laboratory, Johns Hopkins University. We used exactly the same dataset as in [20]. Every input value belongs to $[0,1]$.

We tested this problem with different methods and activation functions. The maximum number of hidden units enabled was 20. The Input and the BGA strategies were tested to select the candidate frequencies. Standard parameters were used for the BGA with an initial random population of 100 individuals and a truncation rate of 25%, as in [3]. In some of the experiments, linear acti-

Method	AF	WR	NH	Train	Test
SAOCIF	lgt	Input	4.65	147.3	76.80%
MFT	lgt	Input	NP	-	-
SAOCIF	lin-lgt	Input	4.00	144.7	76.76%
MFT	lin-lgt	Input	12.90	145.0	76.83%
SAOCIF	fixlin-lgt	Input	5.65	144.2	77.19%
SAOCIF	cos	Input	5.90	143.1	77.06%
MFT	cos	Input	18.75	154.5	76.21%
SAOCIF	lin-cos	Input	7.65	140.6	76.14%
MFT	lin-cos	Input	8.25	144.9	76.90%
SAOCIF	fixlin-cos	Input	11.65	140.1	77.29%
SAOCIF	gau	Input	6.20	143.9	77.09%
MFT	gau	Input	NP	-	-
SAOCIF	lin-gau	Input	8.70	138.5	77.32%
MFT	lin-gau	Input	12.40	145.0	76.93%
SAOCIF	fixlin-gau	Input	8.40	141.3	76.90%

TABLE II

RESULTS FOR THE *Diabetes* PROBLEM WITH THE INPUT STRATEGY.

Method	AF	WR	NH	Train	Test
SAOCIF	lgt	[-0.5,0.5]	4.98	144.6	76.88%
MFT	lgt	[-0.5,0.5]	NP	-	-
SAOCIF	lin-lgt	[-0.5,0.5]	3.51	144.3	76.95%
MFT	lin-lgt	[-0.5,0.5]	12.77	145.0	76.86%
SAOCIF	fixlin-lgt	[-0.5,0.5]	5.51	143.3	76.71%
SAOCIF	cos	[-0.5,0.5]	7.51	133.5	76.93%
MFT	cos	[-0.5,0.5]	NP	-	-
SAOCIF	lin-cos	[-0.5,0.5]	6.64	136.2	77.01%
MFT	lin-cos	[-0.5,0.5]	9.72	144.9	76.78%
SAOCIF	fixlin-cos	[-0.5,0.5]	8.28	137.8	76.90%
SAOCIF	gau	[0,1]	7.68	137.1	76.59%
MFT	gau	[0,1]	NP	-	-
SAOCIF	lin-gau	[0,1]	7.80	137.2	76.24%
MFT	lin-gau	[0,1]	8.11	144.9	76.67%
SAOCIF	fixlin-gau	[0,1]	6.94	140.0	76.64%

TABLE III

RESULTS FOR THE *Diabetes* PROBLEM WITH THE BGA STRATEGY.

vation functions were combined with non-linear activation functions in the hidden layer ('lin-fun' in the tables). When linear activation functions were present, their optimal frequencies can be calculated analytically, solving a linear equations system similar to (1), setting the coefficients to 1. Moreover, when new non-linear units are added, the coefficients of the hidden units with linear activation functions may be either recalculated or keep fixed ('fixlin-fun' in the tables), in order to approximate only the non-linear component of the function. This idea only makes sense for *SAOCIF* method, since *MFT* always keeps the coefficients fixed. Different from the *Two Spirals* problem, previous experience with this problem suggested that it would be necessary to control the complexity of the model in order to obtain a good performance. We performed a 5-fold Cross-Validation, but every original training set was again divided into training set and validation set four times, as in a 4-fold Cross-Validation. For the BGA, this procedure was repeated 5 times. Results shown in Table II and Table III are the average of the results obtained after the addition of the hidden unit where the total squared error in the validation set was minimum. The column 'Train' indicates the total squared error in the training set at this point.

Although the results seem very similar for the different parameter configurations, there are some regularities which can be observed for this problem:

1. Non-linear activation functions different from the classical sigmoidal and gaussian (such as cosines) may be satisfactorily used. Linear (fixed or not) hidden units have mostly a positive influence on the results, both in *SAOCIF* and *MFT* method.
2. Selecting the frequencies from the points in the dataset seems well suited for MLPs, although it is not a common practice. As it was expected, the BGA strategy shows a greater rate of approximation in the training set than the Input strategy.
3. The number of hidden units of the solutions obtained with *SAOCIF* is always less than for the solutions obtained with *MFT*. In contrast with the *Two spirals* problem, this property may not be necessarily true, since the results are evaluated in the minimum of the validation set error. In addition, *SAOCIF* consistently obtains better results than *MFT* .
4. The optimal number of hidden units for this problem is probably small.

We compared these results with those that used similar experimental techniques. In [25] a generalization performance of 76.30% was obtained with 10-fold Cross-Validation without validation set. In [7] this problem was also tested with a collection of bagging and boosting methods (10-fold Cross-Validation), achieving a maximum performance of 75.60%. Support Vector Machines, RBFNs and AdaBoost were tested in [21] obtaining a maximum generalization performance of 76.50%. In [3], with an experimental technique very similar to ours, 75.27% generalization was obtained. In the same conditions aforementioned, we tested the Cascade-Correlation algorithm (see Section V). For more than 70 different sets of parameters, the best average generalization result obtained by Cascade-Correlation was 76.58%. Our best average result was 77.32%.

V. Related work

Finding the frequencies which best match the residue is the underlying idea for most of the previously proposed sequential approaches. It has appeared in different areas with different names (Projection Pursuit in the Statistics literature [10], Matching Pursuit in Signal Processing [18]). In the Neural Netwoks field this idea has also been applied with several variations (see, for example, [11], [15] or [28]). One of the most used constructive method is Cascade-Correlation (CC) [6]. CC combines two key ideas. The former is the cascade architecture, in which hidden units are added one at a time. The newly added hidden neuron receives inputs from the input layer as well as from the previously added hidden neurons. The latter

is the learning algorithm. For each new hidden unit, the algorithm tries to maximize the magnitude of the correlation (or, more precisely, the covariance) between the new unit's output and the residual error signal of the network.

VI. Conclusions and Future work

A sequential method for approximating vectors in Hilbert spaces, called *SAOCIF*, has been presented. The new term is not chosen in order to match the previous residue as best as possible. On the one hand, it optimizes the coefficients, so that we always achieve the best approximation with the selected vectors. On the other, the vectors can be selected at every step in a flexible manner, taking into account the interactions with the previous terms. The approximations defined by *SAOCIF* maintain orthogonal-like properties. Theoretical results prove that, under reasonable conditions, the residue of the approximation obtained with *SAOCIF* (in the limit) is the best one that can be obtained with any subset of the given set of vectors. In the particular case of L^2, *SAOCIF* can be applied to any family of vectors with universal approximation capability (algebraic polynomials, Fourier series, families of time-frequency wavelets, feed-forward neural networks, radial basis function networks, etc). A particular algorithm with neural networks has also been presented. *SAOCIF* can be used as a guide to construct an FNN: adding hidden units one at a time, choosing the initial weights in a flexible and (in some sense) optimal manner, so as to adjust the network until we have a satisfactory model. The resulting method combines the locality of sequential approximations, where only one frequency is found at every step, with the globality of non-sequential methods, such as BP, where every frequency interacts with the others. Experimental results show a very satisfactory performance of this method, and several suggesting ideas for future experiments, such as the selection of the non-linear weights from the dataset, or the combination of linear and non-linear activation functions in the hidden units. In the presented particular algorithm with neural networks, there are also a lot of matters to study or improve. The candidate frequencies can be selected with heuristics different from current selections. Likewise, the selection of the activation function for the new hidden unit admits any number of heuristics.

References

[1] Achieser, N.I. (1956). *Theory of Approximation*. Frederick Ungar, NY.

[2] Barron, A.R. (1993). Universal Approximation Bounds for Superposition of a Sigmoidal Function. *IEEE Trans. Information Theory* 39 (3), 930-945.

[3] Belanche, L. (2000). Heterogeneus Neural Networks: Theory and Applications. *Ph.D. dissertation*, Tech. Univ. Catalonia.

[4] Bishop, C.M. (1995). *Neural Networks for Pattern Recognition*. Oxford University Press Inc., NY.

[5] Mühlenbein, H. and Schlierkamp-Voosen, D. (1993). Predictive Models for the Breeder Genetic Algorithm I. Continuous parameter Optimization. *Evolutionary Computation* 1 (1), 25-49.

[6] Fahlman, S.E. and Lebiere, C. (1990). The Cascade-Correlation Learning Architecture. In *Advances in Neural Information Processing Systems* 2, 524-532. Morgan Kaufmann.

[7] Freund, Y. and Shapire, R.E. (1996). Experiments with a New Boosting Algorithm. *Int. Conf. Machine Learning*, 148-156.

[8] Govaerts, W. (1991). Stable Solvers and Block Elimination for Bordered Systems. *SIAM Journal Matrix Analysis and Applications* 12 (3), 469-483.

[9] Horst, R. and Tuy, H. (1993). *Global Optimization: Deterministic Approaches*. Springer-Verlag, Berlin.

[10] Huber, P.J. (1985). Projection Pursuit. *The Annals of Statistics* 13 (2), 435-475.

[11] Hwang, J.N., Ray, S.R., Maechler, M., Martin, D. and Schimert, J. (1994). Regression Modeling in Back-Propagation and Projection Pursuit Learning. *IEEE Trans. Neural Networks* 5 (3), 342-353.

[12] Hwang, J.N., You, S.S, Lay, S.R. and Jou, I.C. (1996). The Cascade-Correlation Learning: A Projection Pursuit Learning Perspective. *IEEE Trans. Neural Networks* 7 (2), 278-289.

[13] Jones, L.K. (1992). A simple lemma on greedy approximation in Hilbert space and convergence rates for projection pursuit regression and neural network training. *The Annals of Statistics* 20 (1), 608-613.

[14] Kůrková, V. (1997). Incremental Approximation by Neural networks. In *Dealing With Complexity: A Neural Network Approach*, 177-188. Springer-Verlag, London.

[15] Kůrková, V. and Beliczynski, B. (1995). Incremental approximation by one-hidden-layer neural networks. *Int. Conf. Artificial Neural Networks*, 505-510.

[16] Leshno, M., Lin, V.Y., Pinkus, A and Schocken, S. (1993). Multilayer Feedforward Networks With a Nonpolynomial Activation Function Can Approximate Any Function. *Neural Networks* 6, 861-867.

[17] Lorentz, G.G. (1966). *Approximation of Functions*. Chelsea, NY.

[18] Mallat, S.G. and Zhang, Z. (1993). Matching Pursuits with Time-Frequency Dictionaries. *IEEE Trans. Signal Processing* 41 (12), 3397-3415.

[19] Park, J. and Sandberg, I.W. (1993). Approximation and Radial-Basis-Function Networks. *Neural Computation* 5 (2), 305-316.

[20] Prechelt, L. (1997). Investigation of the CasCor Family of Learning Algorithms. *Neural Networks* 10 (5), 885-896.

[21] Rätsch, G., Onoda, T. and Müller, R (1998). An improvement of AdaBoost to avoid overfitting. *Proc. ICONIP*, 506-509.

[22] Reddy, B.D. (1998). *Introductory Functional Analysis with Applications to Boundary Value Problems and Finite Elements*. Springer-Verlag, NY.

[23] Romero, E. (2001). Function Approximation with *SAOCIF*: A General Sequential Method and a Particular Implementation with Neural Networks. *Technical Report LSI-01-41-R*. Tech. Univ. Catalonia.

[24] Sopena, J.M., Romero, E. and Alquézar, R. (1999). Neural Networks with Periodic and Monotonic Activation Functions: A Comparative Study in Classification Problems. *Int. Conf. Artificial Neural Networks*, 323-328.

[25] Wilson, D.R and Martinez, T.R. (1996). Heterogeneous Radial Basis Function Networks. *Int. Conf. Neural Networks*, 1263-1267.

[26] Yosida, K. (1965). *Functional Analysis*. Springer-Verlag, NY.

[27] Young, R.M. (1980). An Introduction to Nonharmonic Fourier Series. Academic Press, NY.

[28] Zhang, J., Morris, A.J. (1998). A Sequential Learning Approach for Single Hidden Layer Neural Networks. *Neural Networks* 11 (1), 65-80.