Elementary derivative tasks and neural net multiscale analysis of tasks

B. G. Giraud*
Service de Physique Théorique, DSM, CE Saclay, F-91191 Gif-sur-Yvette, France

A. Touzeau
École Centrale de Lyon, 36 avenue Guy de Colongues, 69130 Ecully, France

(Received 27 July 2001; published 14 December 2001)

Formal neurons implementing wavelets have been shown to build nets that are able to approximate any multidimensional task. In this paper, we use a finite number of formal neurons implementing elementary tasks such as ‘‘sombrero’’ responses or even simpler ‘‘window’’ responses, with adjustable widths. We show this to provide a reasonably efficient, practical and robust, multifrequency analysis of tasks. The translation degree of freedom of wavelets is shown to be unnecessary. A training algorithm, optimizing the output task with respect to the widths of the responses, reveals two distinct training modes. The first mode keeps the formal neurons distinct. The other mode induces some of the formal neurons to become identical, with output weights of equal strengths but opposite signs. Hence this latter mode promotes tasks that are derivatives of the elementary tasks with respect to the width parameter. Such results, obtained from optimizations with respect to a width parameter, can be generalized for any other parameters of the elementary tasks.

DOI: 10.1103/PhysRevE.65.016109 PACS number(s): 84.35.+i, 87.18.Sn

I. INTRODUCTION

The ability of neural nets to be universal approximators has been proved by Refs. [1,2] and studied by further authors in different contexts. For instance, neurons or small neuronal groups implementing ‘‘plane wave responses’’ have been considered by Refs. [3] and Ref. [4]. Also, pairs of neurons implementing ‘‘window responses’’ have been investigated by Refs. [5]. Any complete enough basis of functions, that is able to span a sufficiently large vector space of response functions, is of interest. For instance, the wavelet analysis has been the subject of a complete investigation by Refs. [6] and [7].

In this paper, we visit again the subject of a linear expansion of tasks in wavelets, but with an emphasis upon neglecting the usual ‘‘translational’’ parameters. We mainly use a scale parameter only. This is somewhat different from the usual wavelet approach, which takes advantage of both translation and scale. But we shall find that a multifrequency reconstruction of tasks occurs as well.

For the sake of robustness and biological relevance, we introduce a significant amount of randomness, corrected by training, in the initial choice of the implemented neuronal parameters. Furthermore, our basic neuronal units are not necessarily strictly related to wavelets. They can be those ‘‘window response’’ pairs advocated earlier [5], because of biological relevance too. Such deviations from the more rigorous approaches of Refs. [6] and [7] are expected to make cheaper the practical implementation of such neural nets. They turn out to give similar results, namely, the multifrequency analysis works as well with windowlike, sombrerolike or any more general elementary response.

We investigate two training operations. The first one consists of an easy optimization of the output synaptic layer connecting a layer of intermediate, ‘‘elementary task neurons’’ to an output, purely *linear* neuron. The second training consists of optimizing the scale parameters of such a layer of intermediate neurons. It will be found that one may start from random values of such parameters and, however, sometimes reach solutions where some among the intermediate neurons are driven to become identical. This ‘‘dynamical identification’’ training will be discussed.

For the generic case of multidimensional inputs, we separate a ‘‘radial’’ from an ‘‘angular’’ analysis of the task. This technical manipulation does not change our results.

In Sec. II, we describe our formalism, including a traditional universality theorem. We also reduce the realistic, multidimensional situations to a one-dimensional problem. In Sec. III we illustrate such considerations by numerical examples of network training. Section IV contains a few considerations for the generalization of the results obtained with the wavelet model and similar models. Finally Sec. V contains our discussion and conclusion.

II. FORMALISM

A. Definitions, architecture

Consider an input $X>0$ that must be processed into an output (a task) $F(X)$. This input is taken here to be a positive number, such as the intensity of a spike or the average intensity (or frequency) of a spike train. One may view X as a ‘‘radial’’ coordinate in a suitable space. There is no loss of generality in restricting X to be a positive number, because, should negative values of X be necessary for the argument, then F could always be split into even and odd parts, $[F(X)\pm F(-X)]/2$, respectively. Such even and odd parts need only be known for $X>0$, obviously. Outputs, in turn, will have both signs, in order to account for both excitation or inhibition. Finally there is no need to tell a scalar task $F(X)$ from a vector task $\{F_1(X),F_2(X),\ldots\}$, since any

*Email address: giraud@spht.saclay.cea.fr

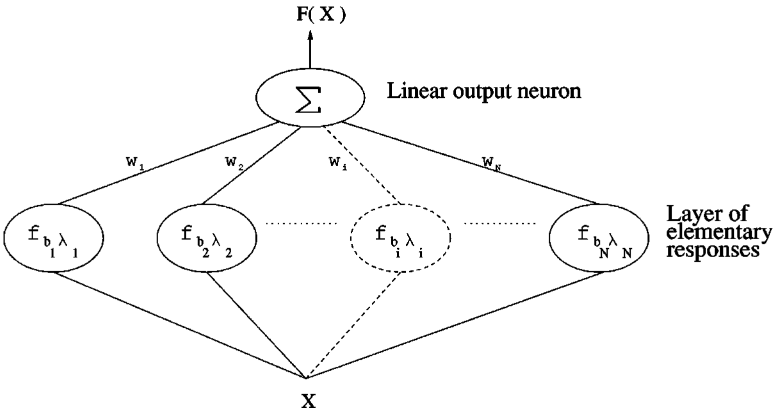

FIG. 1. All elementary units (FN's) receive the same input X. Each unit returns an output f, which depends on parameters such as a threshold b and a scale λ. Output synaptic weights $w(b,\lambda)$ linearly mix such elementary outputs $f_{b\lambda}(X) \equiv f(X;b,\lambda)$ into a global output $F(X)$.

component $F_k(X)$ boils down to a separate scalar task, and this can be processed by a parallel architecture.

Consider now neuronal units which, for instance, may be excitatory-inhibitory pairs of neurons providing a window-like elementary response. Or they may be more complicated assemblies of neurons, providing a more elaborate "mother wavelet," such as a "sombrero." We denote $f(X)$ the response function of such a unit and, for short, call this unit a "formal neuron" (FN). The traditional wavelet approach uses a set of such FN's with various thresholds b and scale sensitivities λ, hence a space of elementary responses $f_{b\lambda}(X)\equiv f[(X-b)/\lambda]$. The same approach expands F in this set,

$$F(X)=\int db\,d\lambda\,w(b,\lambda)f[(X-b)/\lambda)], \tag{1}$$

where the integral is most often reduced to a discrete sum. Also, b and λ do not need to be independent parameters. The expansion coefficients, $w(b,\lambda)$, are output synaptic weights and are the unknowns of the problem. This well-known architecture is shown in Fig. 1.

B. One-dimensional universality, radial case

The following, seemingly poorer, but simpler expansion,

$$F(X)=\int_0^\infty d\lambda\, w(\lambda)\lambda^{-1}f(X/\lambda) \tag{2}$$

does not use the translation parameter b. Here it is assumed that there exists a suitable electronic or biological tuning mechanism, able to recruit or adjust FN's with suitable gains λ^{-1}, but no threshold tuning. Such gains are positive numbers, naturally. The outputs of such FN's are then added, via synaptic output efficiencies $w(\lambda)$, which can be both positive and negative, namely excitatory and inhibitory, respectively. The coefficient λ^{-1} is introduced in Eq. (2) for convenience only. It can be absorbed in $w(\lambda)$.

This expansion, Eq. (2) allows a universality theorem. Define $Y=\ln X$ and $L=\ln\lambda$. The same expansion becomes,

$$G(Y)\equiv F(e^Y)=\int_{-\infty}^{\infty} dL\, W(L)g(Y-L), \tag{3}$$

where $W(L)\equiv w(e^L)$ and $g(Y)\equiv f(e^Y)$. This reduces the "scale expansion," Eq. (2), into a "translational expansion" where a basis is generated by arbitrary translations of a given function. The solution of this inverse convolution problem is trivially known as $\hat{W}(p)=\hat{G}(p)/\hat{g}(p)$, where the superscript $\hat{}$ refers to the Fourier transforms of W, G, and g, respectively, and p is the relevant "momentum." This result will make our claim for universality. In the following, this paper empirically assumes that the needed analytical properties of $f,\hat{f},\ldots\hat{W}$ are satisfied. Actually, for the sake of biological or industrial relevance, we are only concerned with discretizations of Eq. (2), with N units,

$$F_{app}(X)=\sum_{i=1}^{N} w(\lambda_i)f(X/\lambda_i), \tag{4}$$

where we now let w include the coefficient $\lambda_i^{-1}\,d\lambda$. Also, in an obvious short notation, we will use $w_i\equiv w(\lambda_i)$.

C. Rotational analysis

Obviously, input patterns to be processed by a net cannot be reduced to one degree of freedom X only. Rather, they consist of a vector $\vec{X}$ with many components $X_1,X_2,\ldots,X_P$. These may be considered as, and recoded into, a radial variable $X=\sqrt{\Sigma_{j=1}^{P}X_j^2}$ and, to specify a direction on the suitable hypersphere, $(P-1)$ angles $\alpha_1,\alpha_2,\ldots,\alpha_{P-1}$. Enough special functions (Legendre polynomials, spherical harmonics, and rotation matrices, etc.) are available to generate complete functional bases in angular space and one might invoke some formal neurons as implementing such base angular functions. The design of such FN's, and as well the design of such a polar coordinate recoding, is a little far fetched, though. In this paper, we prefer to take advantage of the following argument, based upon the synaptic weights of the input layer, shown in Fig. 2.

In the left part of the figure, Fig. 2, all the FN's have the same input synaptic weights $\vec{u}\equiv\{u_1,u_2,\ldots,u_P\}$, hence receive the same input $X=\vec{u}\cdot\vec{X}$ when contributing to a global task F. For the right part of Fig. 2, it is again assumed that all FN's have equal input weights, with, however, weights $\vec{u}\,'$ deduced from $\vec{u}$ by a sheer rotation, $\vec{u}\,'=\mathcal{R}\vec{u}$. Accordingly, if the output weights of the left part are the same as those of the

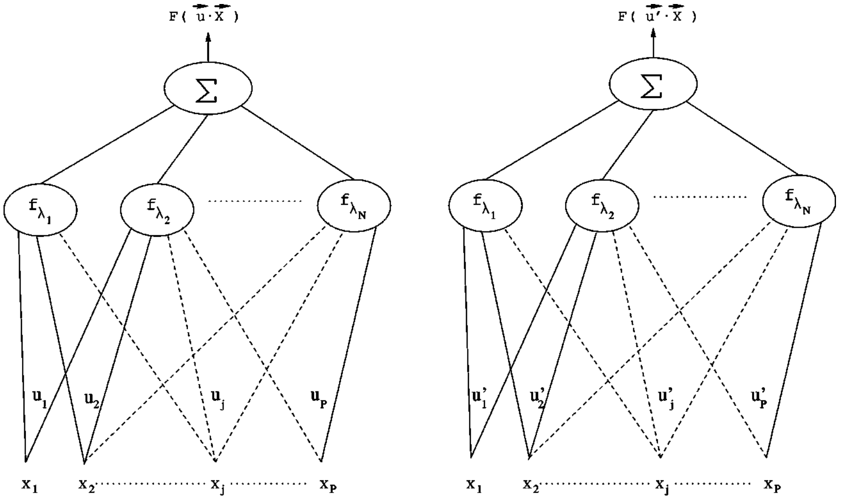

FIG. 2. Architecture showing how a task can be rotated by means of the input synaptic weights.

right one, the global task $F(\vec{u}'\cdot\vec{X})$ performed by the right part is a rotated task, $F'=\mathcal{R}F$. An expansion of any task $\mathcal{F}$ upon the $(P-1)$ rotation group is thus available,

$$\mathcal{F}=\int d\mathcal{R}\,\mathcal{W}(\mathcal{R})\,\mathcal{R}F, \tag{5}$$

where discretizations are in order, naturally, with suitable output weights $\mathcal{W}$. Here F plays the role of an elementary task, and it might be of some interest to study cases where F belongs to specific representations of the rotation group. This broad subject exceeds the scope of the present paper, however, and, in the following, except in Sec. IV, we mainly restrict our considerations to scalar tasks $F(X)$ of a scalar input X according to Fig. 1 only.

D. Training output weights

Let us return to Eq. (4), in an obvious, short notation $F_{app}=\Sigma_i w_i f_i$. Two kinds of parameters can be used to best reconstruct F: the output synaptic weights w_i and, hidden inside the elementary tasks f_i, the scales λ_i. Let $\langle|\rangle$ denote a suitable scalar product in the functional space spanned by all the f_i's of interest. We assume, naturally, that the same scalar product makes sense for the F's. Incidentally, there is no loss of generality if F is normalized, $\langle F|F\rangle=1$, since the final neuron is linear.

One way to define the "best" F_{app} is to minimize the square norm of the error $(F-F_{app})$. In terms of the w_i's, this consists in solving the equations,

$$\frac{\partial}{\partial w_i}\left(\langle F|F\rangle-2\sum_{j=1}^{N} w_j\langle f_j|F\rangle+\sum_{j,k=1}^{N} w_j\langle f_j|f_k\rangle w_k\right)=0, \qquad i=1,\dots,N. \tag{6}$$

Let $\mathcal{G}$ be that matrix with elements $\mathcal{G}_{jk}=\langle f_j|f_k\rangle$. Its inverse $\mathcal{G}^{-1}$ usually exists. Even in those rare cases when $\mathcal{G}$ is very ill conditioned, or its rank is lower than N, it is easy to define a pseudoinverse such that, in all cases, the operator $\mathcal{P}=\Sigma_{i,j=1}^{N}|f_i\rangle(\mathcal{G}^{-1})_{ij}\langle f_j|$ is the projector upon the subspace spanned by the f_i's. Then an easy solution, $F_{app}=\mathcal{P}F$, is found for Eqs. (6),

$$w_i=\sum_{j=1}^{N}(\mathcal{G}^{-1})_{ij}\langle f_j|F\rangle, \qquad i=1,\dots,N. \tag{7}$$

Given F and the f_i's, this projection, which can be achieved by elementary trainings of the output layer of synaptic weights, will be understood in the following. It makes the w_j's functions of the λ_i's.

E. Training elementary tasks

Now we are concerned with the choice of the parameters λ_i of the FN's performing elementary tasks. This is of some importance, for the number N of FN's in the intermediate layer is quite limited in practice. The subspace spanned by the f_i's is thus, most undercomplete. Hence, every time one requests an approximator to a new F, an optimization with respect to the intermediate layer is in order, to patch likely weaknesses of the "projector" solution, (6).

Let us again minimize the square norm $\mathcal{E}=\langle(F-F_{app})|(F-F_{app})\rangle$ of the error. We know from Eqs. (6) that the w_i's are functions of the λ_j's, but there is no need to use chain rules $\partial w_i/\partial\lambda_j\ \partial/\partial w_i$, because the same equations, (6), cancel the corresponding contributions, the w_i's being optimal. Derivatives of the f_i's with respect to their scales λ_i are enough. The gradient of $\mathcal{E}$, to be cancelled, reads,

$$\frac{\partial\mathcal{E}}{\partial\lambda_j}=\frac{2w_j}{\lambda_j^2}\langle Xf'(X/\lambda_j)|(F-F_{app})\rangle=0, \qquad j=1,\dots,N. \tag{8}$$

Here f' is the straight derivative of the reference elementary task, before any scaling. There is no difficulty in implementing a training algorithm for a gradient descent in the λ space.

The next section, Sec. III, gives a brief sample of the results, we obtained when solving Eqs. (6) and (8) for many choices of the global task F and elementary task f.

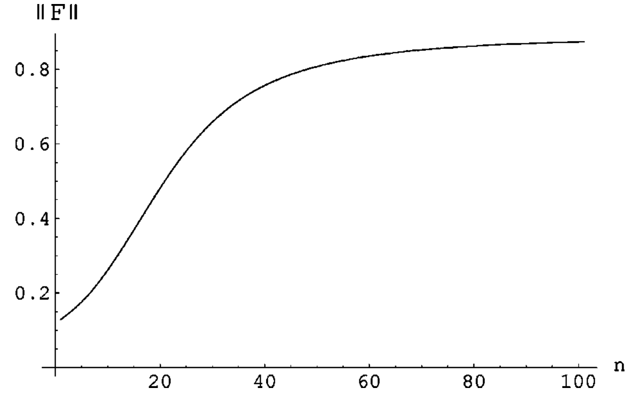

FIG. 3. Learning curve: the norm $||F||$ of F_{app} increases as a function of the number n of learning steps, then saturates.

III. NUMERICAL ILLUSTRATIVE EXAMPLES

A. Symmetry and degeneracy

Define for instance the scalar product in the functional space as, $\langle f_i|f_j\rangle\equiv\int_0^{20}dX\, f_i(X)\, f_j(X), \forall\, f_i, f_j$. Among many numerical tests, we show here the results obtained when the target task reads, $F(X)=0.10167\, e^{-X/10}\{0.60\,717\tanh[4(X-1.66\,133)]-4.33\,575\tanh[4(X-9.565\,91)]\}$. Let the elementary task of a FN read $f(X/\lambda)=(1-X^2/\lambda^2)\, e^{-X^2/(2\lambda^2)}$, a sombrero. Set $N=5$, and initial values 1/4, 1/2, 1, 2, and 4 for the λ_i's. Keeping Eqs. (6) satisfied at each step, start a gradient descent from such initial values. Our increments of the λ_i's at each step read, $\delta\lambda_i=-2\partial\mathcal{E}/\partial\lambda_i$, see Eqs. (8). After $\simeq 90$ steps, a saturation of $||F_{app}||^2=\langle F|\mathcal{P}|F\rangle$ begins, see Fig. 3.

A comparison between F and F_{app} is provided by Fig. 4.

This saturation makes it reasonable to interrupt the learning. For the sake of rigor, however, another run, with 1000 steps, was used to verify the saturation. While saturation is confirmed, the convergence of the λ_i's is slightly slow. The values of the λ_i's and w_i's at the end of this second run read $\{0.249,0.535,1.0512,1.0522,11.13\}$ and $\{-0.0008,-0.0002,38.107,-38.121,0.3764\}$, respectively. The weakness of w_1 and w_2 is explained by the lack of a fine structure in F. The large, almost opposite values of w_3 and w_4 clearly mean a renormalization of (f_3-f_4), since λ_3 and λ_4 are so close to each other. Numerical care because of difference effects may therefore be necessary for practical applications. We show in Fig. 5 how the λ_i's evolved during the first 100 steps of the gradient descent.

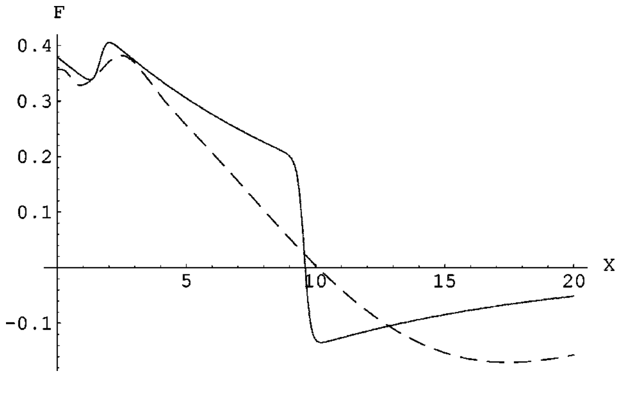

FIG. 4. A target task (solid line) and its best approximation (dashed) after learning saturation.

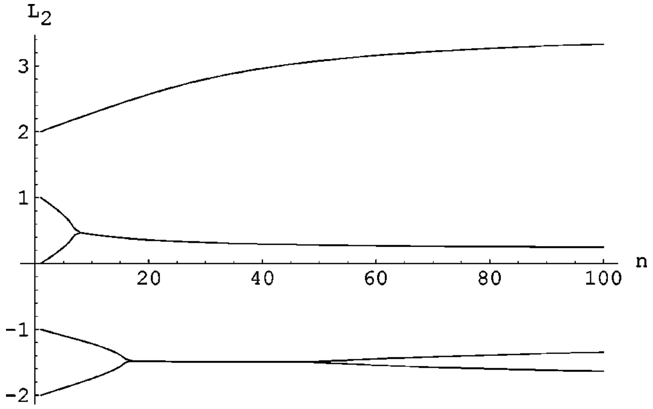

FIG. 5. Evolution of $L_2\equiv\log_2\lambda_i$, $i=1,\dots 5$, as functions of the learning step n. Case where pairs of formal neurons create subunits that perform those tasks that are derivatives of the elementary task with respect to the parameter under training. Notice furthermore, in that special example, the fusions of *two* pairs of scales, then the splitting of one of them.

A temporary merging of λ_1 with λ_2, then a final episode in which they become distinct again, are striking, as well as the merging of λ_3 with λ_4. It will be stressed at this stage that the whole process is invariant under any permutation of the λ_i's (and of their associated w_i's), hence a "triangular" rule, $\lambda_i\leqslant\lambda_{i+1}$ can be implemented without restricting learning flexibility. Furthermore, as a symmetric function under pairwise exchanges of such parameters, the error square norm $\mathcal{E}$ has a vanishing "transverse" derivative, $\partial\mathcal{E}/\partial(\lambda_i-\lambda_j)=0$, every time $\lambda_i=\lambda_j$. It is thus not surprising that, at least for part of the learning process, the learning path rides lines where such parameters merge.

When merging occurs, the functional basis seems to degenerate since f_i and f_j are not distinct. It will be recalled, however, that our output neuron is linear, and nothing prevents the process from using the strictly equivalent representations, $w_i f_i+w_j f_j\equiv(w_i+w_j)/2\times(f_i+f_j)+(w_i-w_j)/2(f_i-f_j)$. A trivial renormalization of the (f_i-f_j) term makes it that the functional basis still contains two independent vectors, namely, a new elementary response $\partial f/\partial\lambda$ besides $f_i=f_j$. Naturally, the renormalization has a numerical cost, since both w_i and w_j must diverge. In practice, a minute modification of the "triangular rule," which becomes, in our runs, $\lambda_{i+1}-\lambda_i\geqslant 10^{-3}$, is enough to smooth our calculations. The conclusion of this merging phenomenon, for those F's where it occurs, is of some interest: new specialized neuronal units (new FN's) may spontaneously emerge. These we call "derivative task units," because they represent a new elementary task $\partial F/\partial\lambda$, or, if $(p+1)$ parameters merge, any further derivative $\partial^p f/\partial\lambda^p$.

B. Full Symmetry Breaking

Most choices of F yield distinct values for the λ_i's. We show in Fig. 6 a trivial case. Here $f=1/[1+(X^2/\lambda^2)]$, a

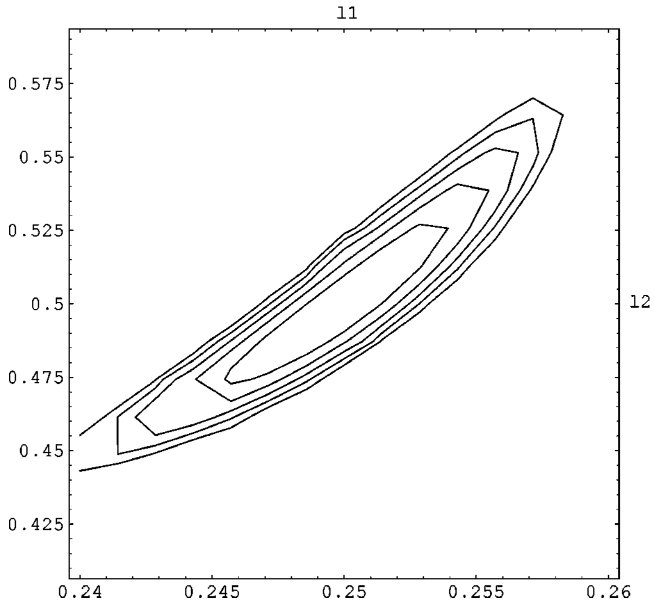

FIG. 6. Symmetry breaking case. Contours of the error in the vicinity of a symmetry breaking set of parameters. The last three out of five adjustable parameters are frozen and distinct. The minimum of the error is reached with unequal values of the first two parameters.

windowlike elementary response, and the target task reads $F=9/(1+16X^2)+5/(1+4X^2)+2/(1+X^2)-1/[1+(X^2/4)]-1/[1+(X^2/16)]$, a sum of such windows. We freeze $\lambda_3=1, \lambda_4=2$, and $\lambda_5=4$, a symmetry breaking situation, and clearly a part of the obvious solution for the minimum of $\mathcal{E}$. Then the contour map of $\mathcal{E}$ in the $\{\lambda_1,\lambda_2\}$-space does show the expected minimum for $\lambda_1=1/4$ and $\lambda_2=1/2$. The minimum turns out to be very flat, hence some robustness is likely for that special case. The learning process does reach this ''fully symmetry breaking'' configuration, together with the corresponding set of w_i's, namely, $\{9,5,2,-1,-1\}$. Many other, less academic cases generate a full symmetry breaking, namely, distinct λ_i's.

C. More numerical results

Besides ''windows'' and ''sombreros,'' we also used oscillatory shapes such as $(\sin X)/X$ for f. A cutoff by an exponential decay was also sometimes introduced. The range of the scalar product integration was independently varied within one order of magnitude. Sometimes the dimension N of the elementary task basis was also taken as a random number, a test of little interest, however, which just verified that F_{app} improves when N increases. For F, a few among our tests involved a small amount of random noise added to a smooth main part $F_{background}$. Furthermore, we investigated a fair amount of piecewise continuous F's, this case being of interest for image processing [8]. Alternately, we smoothed such discontinuities with a suitable definition of F, such as, $F=\Sigma_{\ell} c_{\ell} \tanh[\sigma(X-\beta_{\ell})]$, with randomized choices of the number of terms, the coefficients c_{ℓ}, the large ''slope coefficient σ'', and the positions β_{ℓ} of the steep areas. The set of initial values for the λ_i's before gradient descent was also sometimes taken at random. It was often found that a traditional sequence $\lambda_i \simeq 2^{i-(N+1)/2}$ is not a bad choice for a start.

All our runs converge reasonably smoothly to a saturation of the norm $||F_{app}||$, provided those cases where $\mathcal{G}$ becomes ill conditioned are numerically processed. There is a significant proportion of runs where the optimum seems to be quite flat, hence some robustness of the results. Local minima where the learning gets trapped do not seem to occur very often, but this problem deserves the usual caution, with the usual annealing if necessary. We did not find clear criteria for predicting whether a given F leads to a merging of some λ_i's, however. Despite this failure, all these results advocate a reasonably positive case for the learning process described by Eqs. (6) and (8) and the emergence of derivative task subunits, performing tasks that are derivatives of f with respect to its parameters.

IV. GENERALIZATIONS

Most among the considerations of this paper clearly hold if one replaces wavelets by other responses and scaling parameters by other continuous parameters.

The most important and well-known issue is that of the universality offered by nets whose architecture is described by Figs. 1 and 2, namely, four layers: (i) input weights **u**, (ii) FN's for elementary tasks **f** with adjustable parameters **M**, (iii) output weights **w**, and (iv) linear output neuron(s). The linearity of the final layer can be summarized in any dimensions by the linear transform $\mathbf{F}(\mathbf{X})=\int d\mathbf{M}\,\mathbf{w}(\mathbf{M})\mathbf{f}(\mathbf{X};\mathbf{M})$. (We use here boldface symbols to stress that the linearity generalizes to any suitable vector and tensor situations for multiparameter inputs, intermediate tasks, and outputs.) This linearity reduces the theory of such an architecture to a special case of the ''generator coordinate'' theory, well known in physics [9]. From a mathematical point of view, this also boils down to the question of the invertibility of the kernel $\mathbf{f}(\mathbf{X};\mathbf{M})$. Actually, the invertibility problem consists in identifying those classes of global tasks **F**, which belong to the functional (sub)space spanned by the **f**'s. The experience obtained in many domains of physics with the generator coordinate approach provides a qualitative general rule: one must be very clumsy with the choice of elementary tasks, or very unlucky with a very singular global task, to miss enough universality and fail a reasonable reconstruction of that global task.

Another, and harder problem, however, is to find a general theory for a minimal cost of the reconstruction. Although this paper was essentially confined to scalar tasks of scalar arguments, we briefly sketch an approach, taken from a special case of the generator coordinate theory: the angular momentum projections that are so familiar in the theory of molecular and nuclear rotational spectra [10]. The parameters **M** can be defined as including the input synaptic weight vectors $\vec{u}$, whose dimension is necessarily the same as that of the inputs $\vec{X}$ in order to generate the actual inputs $\vec{u}\cdot\vec{X}$ received by the intermediate FN's. When **M** also explicitly includes scale parameters λ, there is no loss of generality in restricting the

$\vec{u}$'s to be unitary vectors. Hence the linear kernel **f** can imply, in a natural way, an integration upon the group of rotations transforming all the $\vec{u}$'s into one another. This is of interest if the global task **F** turns out to belong to a finite representation of a rotation group and if elementary tasks, related to representations of the same group, can be implemented in a cheap way. An optimization of the expansion of **F** is then obviously available.

In any case, because of the permutation symmetry of the problem with respect to the intermediate tasks **f**, any gradient descent in any space of continuous parameters of the **f**'s may ride a line where two (or more) of such parameters become equal. The occurrence of tasks that are derivatives of the initial elementary tasks can thus be expected to be nonexceptional.

V. DISCUSSION AND CONCLUSION

In this paper we proved a universality theorem for neural nets in the special case of neurons whose response can undergo ''scaling without translating,'' a case inspired by wavelets. The elementary response under scaling, incidentally, does not need to be just a wavelet. We showed how window-like responses can make suitable building blocks of the nets.

This paper also takes advantage of the well-known issue of the discretization of a continuous expansion, which converts kernels into finite matrices, naturally. The paper studied what happens if one trains output weights for a temporary optimum of the approximate task F_{app}, while the intermediate elementary tasks are not yet optimized. This implies a prejudice on training speeds: weights fast learners and parameters of elementary tasks slower. Other choices, such as weights slower learners and parameters faster, for instance, are as legitimate, and should be investigated too. The question is of importance for biological systems, because of obvious likely differences in the time behaviors and biochemical and metabolic factors of synapses and cell bodies. The training speed hierarchy, we chose pointed to one technical problem only, namely whether the Gram-Schmidt matrix $\mathcal{G}$ of scalar products $\langle f_i | f_j \rangle$ was easily invertible or not. We did not use a Gram-Schmidt orthogonalization of the finite basis of such f_i's, but the (pseudo) inversion of $\mathcal{G}$ amounts to the same. Once $\mathcal{G}^{-1}$ is obtained, temporarily optimal weights are easily derived.

Our further optimization of F_{app} with respect to the parameters of the intermediate FN's takes advantage of the linearity of the output(s) and the symmetry of the problem under any permutation of the FN's. Let i label such FN's, $i = 1, \ldots, N$ and denote λ_i the scaling parameter of the ith FN. We found cases where the gradient descent used to optimize F_{app} induces a few λ_i's to become large, quite close to one another, with opposite signs. Such functional clusters, because of the output linearity, may yield almost elementary tasks corresponding to derivatives of f with respect to λ. This derivative process may look similar to a Gram-Schmidt orthogonalization, but it is actually distinct, because no rank is lost in the basis. For those F's that induce such mergings of FN's, industrial applications should benefit from a preliminary simulation of training as a useful precaution. Indeed, besides straight FN's implementing f, additional, more specific FN's implementing $df/d\lambda$ will be necessary. For biological systems, diversifications of neurons, or groups of such, between tasks and such derivative tasks might also be concepts of interest.

In some cases, it may be noticed that the word ''derivative'' may hold with respect to inputs as well as parameters. Indeed, as found at the stage of Eq. (3), scale parameters reduce, in a suitable representation, to translational parameters in a task $g(Y-L)$. The sign difference between $\partial g/\partial Y$ and $\partial g/\partial L$ is obviously inconsequential.

This emergence of derivative elementary tasks prompts us into a problem yet unsolved by our numerical studies with many different F's and many different f's: given the shape of f, it would be useful to predict whether a given F leads to a full symmetry breaking or to a partial merging of the FN's. This question is under study.

ACKNOWLEDGMENT

A.T. thanks Service de Physique Théorique, Saclay, for its hospitality during this work.

[1] G. Cybenko, Mathematics Control, Signals Systems **2**, 303 (1989).

[2] K. Hornik, M. Stinchcombe, and H. White, Neural Networks **2**, 359 (1989); K. Hornik, M. Stinchcombe, H. White, and P. Auer, Neural Comput. **6 (6)**, 1262 (1994).

[3] B. Irie and S. Miyake, in *Capabilities of Three-Layered Perceptrons*, IEEE First International Conference on Neural Networks (IEEE, 1988), Vol. 1, pp. 641–648.

[4] B.G. Giraud, L.C. Liu, C. Bernard, and H. Axelrad, Neural Networks **4**, 803 (1991).

[5] B.G. Giraud, A. Lapedes, L.C. Liu, and J.C. Lemm, Neural Networks **8 (5)**, 757 (1995).

[6] A. Benveniste and Q. Zhang, Neural Networks **3 (6)**, 889 (1992).

[7] Y. Oussar and G. Dreyfus, Neurocomputing **34**, 131 (2000).

[8] J. Hertz, A. Krogh, and R.G. Palmer, *Introduction to the Theory of Neural Computation* (Addison-Wesley, Massachussetts, 1991).

[9] D.L. Hill and J.A. Wheeler, Phys. Rev. **89**, 1102 (1953); J.J. Griffin and J.A. Wheeler, *ibid.* **108**, 311 (1957).

[10] A. Bohr and B. Mottelson, *Nuclear Structure; vol. II, Nuclear Deformations* (Benjamin, Reading, New York, 1975), p. 90.