Neural Networks, Vol. 5, pp. 207–220, 1992
Printed in the USA.

0893-6080/92 $5.00 + .00

ORIGINAL CONTRIBUTION

Approximation of a Function and its Derivative with a Neural Network

PIERRE CARDALIAGUET

Alcatel-Alsthom Recherche and University of Paris-Dauphine

AND

GUILLAUME EUVRARD

Alcatel-Alsthom Recherche and E.S.P.C.I.

(*Received* 13 *October* 1990; *accepted* 23 *July* 1991)

Abstract—*This paper deals with the approximation of both a function and its derivative by feedforward neural networks. We propose an explicit formula of approximation which is noise resistant and can be easily modified with the patterns. We apply these results to approach a function defined implicitly, which is useful in control theory.*

Keywords—Feedforward neural networks, Functions approximation, Interpolation, Bell-shaped functions, Squashing functions, Robustness with respect to noise, Implicit function, Control.

1. INTRODUCTION

It is well known that feedforward neural networks are "universal approximators." It is proved in Cybenko, (1989) and in Funahashi (1989), that any continuous function can be approximated on a compact set with the uniform topology, by a layered network with one hidden layer. Hornik, Stinchcombe, and White (1989), have shown that any measurable function can be approached with such a network. Furthermore, these authors proved, in Hornik, Stinchcombe, and White (1990), that the functions of the Sobolev's spaces can be approached with all their derivatives. Yet, these results only give theorems about the existence of an approximation. The goal of this paper is to show that there exist explicit approximation formulas, from which it is possible to build the network from examples.

When one tries to approach a continuous function with a neural network, the usual method consists in taking a set of values (often called "examples") of the function, and minimizing an error function on these points, using the so called "Gradient Back Propagation" algorithm (Rumelhart, Hinton, & Williams, 1986). The main drawback of this algorithm is the large number of iterations it needs to converge. One can also say that one never knows whether it will converge, and whether it will converge to a function having the desired properties (problem of local minima).

Futhermore, the functions obtained only make an *interpolation* of the patterns, and one never knows, even if the number of patterns increases to infinity, if these functions will converge to the initial function for a given norm. It is well known, for instance (the Runge phenomenon, see Dieudonné, 1980), that there exist analytic functions for which interpolation with polynomials does not converge.

In this paper we show that it is possible, when some values of the function are known, to obtain an *approximation* of this function by a feedforward neural network whose weights are *explicitly* given instead of an interpolation. This formula is noise resistant and can be generalized to the approximation of both a function and its derivative. We see that this kind of approximation is necessary in control theory: we develop the case of a function defined implicitly (for example a command determined by a process),

Acknowledgement: This research was supported by Alcatel Alsthom Recherche. We want to thank the members of the "Robotique et Neuronal" group for their advice and remarks. A particular mention for Isabelle Salaün, Emmanuel Schalit, and Guy Tabary, whose presence has been crucial for the motivation and the advancement of this paper.

Requests for reprints should be sent to Guillaume Euvrard, Alcatel Alsthom Recherche, D.I.A., Route de Nozay, 91460 Marcoussis, France.

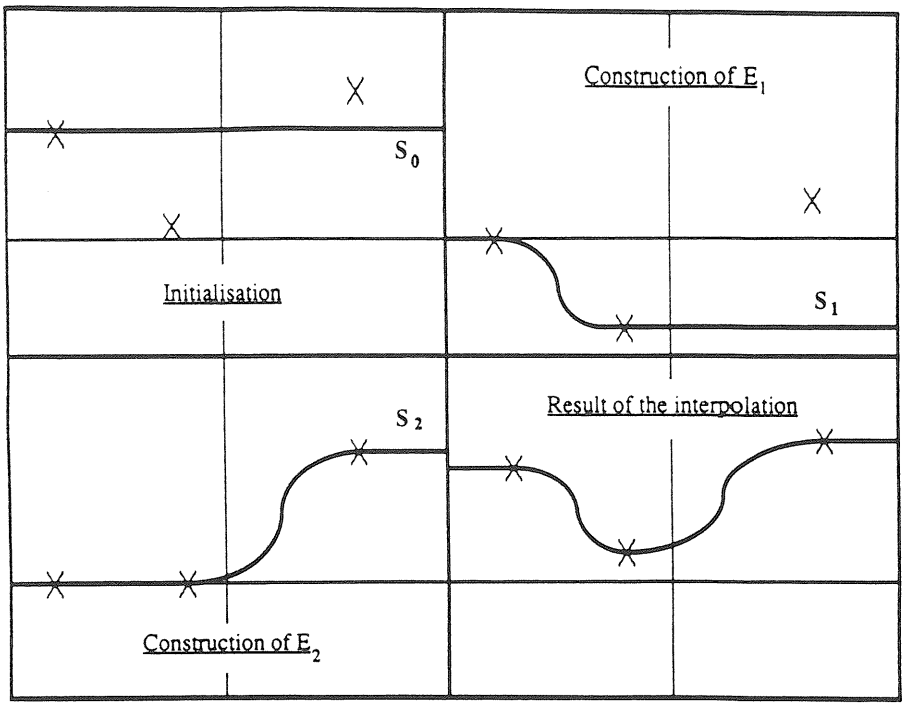

FIGURE 1. Illustration of the interpolation algorithm. For each new input, a neuron is added which does not change the outputs of the previous points, and makes the correct output for this input.

3. THE BELL-SHAPED FUNCTIONS: DEFINITION AND MAIN PROPERTIES

3.1. Definition and Examples

We will now introduce a function family (the bell-shaped functions), the linear combinations of which can make an approximation from examples of any continuous function.

DEFINITION. *A function* $b: \mathbf{R} \to \mathbf{R}$ *is said to be bell-shaped if* b *belongs to* L^1 *and its integral is nonzero, if it is nondecreasing on* $(-\infty, a)$ *and nonincreasing on* $[a, +\infty)$, *where* a *belongs to* $\mathbf{R}$. *In particular* $b(x)$ *is a non-negative number and the number* a *is a global maximum for* b; *it is the center of the bell-shaped function. A bell-shaped function is said to be centered if its center is zero.*

Examples (see Figure 2).

1. *Piecewise constant functions.* The most elementary bell-shaped function is the characteristic function of $[-1, 1]$. It will give an approximation by piecewise constant functions.

   It can be easily obtained from the Heaviside function H, where $H(x)$ is equal to 1 if x is non-negative and 0 otherwise: just consider the expression:

$$\frac{H(x + 1) + H(1 - x)}{2}$$

2. *The bell-shaped functions Gauss algebra.* Let us call $b_{a,d}$ the following class of bell-shaped functions:

$$x \to \exp \frac{(x - a)^2}{d} \qquad \text{where } x, a, \text{ and } d \text{ belong to } \mathbf{R}.$$

   The bell-shaped functions Gauss algebra is the set of functions:

$$\sum_k c_k \cdot b_{a_k, d_k}, \qquad \text{where} \qquad \sum_k |c_k| < \infty.$$

   It contains in particular the functions from the Schwartz's class (Meyer, 1990).
3. *Bell-shaped and spline functions.* If we consider the following kind of bell-shaped functions:

$$\begin{cases} 0 \text{ on } (-\infty, -1] \text{ and on } [1, +\infty) \\ (1 - x^2)^{2n} \text{ on } [-1, 1] \end{cases}$$

   we will obtain, in using Theorem 2, the approximation by spline functions. We notice that those functions are C^{2n-1}.
4. *Relation with squashing functions.* The most interesting property of the bell-shaped functions for neural networks is that the primitive of a bell-shaped function is a squashing function. For example the derivative of the hyperbolic tangent is bell-shaped. C^∞, even and centered. This property will be used when we will study the possibility of approximating both a function and its derivative.

   Given a squashing function the derivative of which is even, centered, and bell-shaped, and a

x_j^n is within a distance $\frac{1}{n}$ of $\frac{k}{n}$. The proof of this property is the same as the one given above.

3. *Assumption on the bounds of the sum in formula(*).* In Theorem 2, we made the sum between the integers $-n^2$ and n^2. It is obvious that if we sum between $-a_n$ and b_n, where a_n and b_n are nondecreasing sequences of integers, the convergence of $f_n(x)$ to $f(x)$ is uniform on the compact sets of the interval $(-\liminf a_n/n, \liminf b_n/n)$. In particular, if the support of the functon f is in this interval, the above sequence converges to f uniformly on this interval.
4. *Robustness with respect to noise.* The formula (*) is still valid if the patterns $f(\frac{k}{n})$ are altered with noise. This will be proved in Section 4.
5. *Case of higher dimension.* If we have the "product" neuron in the network, we can have, for free, the following generalization in higher dimensions:

Let $f: \mathbf{R}^p \to \mathbf{R}$, be a continuous and bounded function. The functions

$$f_n(x) = \sum_{k_1=-n^2}^{n^2} \cdots \sum_{k_p=-n^2}^{n^2} \times \frac{f\left(\frac{k_1}{n}, \ldots, \frac{k_p}{n}\right)}{I.n^{p\alpha}} \times b\left(n^{1-\alpha}\left(x_1 - \frac{k_1}{n}\right)\right) \cdots b\left(n^{1-\alpha}\left(x_p - \frac{k_p}{n}\right)\right)$$

converge uniformly on compacta to $f(x)$. The general case will be treated in Section 5.

4. ROBUSTNESS WITH RESPECT TO NOISE

The *redundancy* is one of the most important advantages of neural networks. Each information is shared out by many connections. This particularity gives a good immunity to noise, that we describe and explain in this section.

4.1. The General Case

We will now consider the case where the values $f(\frac{k}{n})$ are not known exactly, that is the measures provide in fact $(1 + a_{k,n}) \cdot f(\frac{k}{n}) + b_{k,n}$, where $a_{k,n}$ et $b_{k,n}$ are noises. We will always make the following assumptions: the noises are independent and have zero mean.

THEOREM 3. *Let (Ω, A, P) be a probability space. If $f: \mathbf{R} \to \mathbf{R}$ is a continuous bounded function, and if $a_{k,n}$ and $b_{k,n}$ are random variables which are independent, in $\mathbf{L}^2(\Omega, \mathbf{A}, \mathbf{P})$, with zero mean and bounded variances in the aggregate, then the sequence*

$$f_n(x) = \sum_{k=-n^2}^{n^2} \frac{(1 + a_{k,n}) \cdot f\left(\frac{k}{n}\right) + b_{k,n}}{I.n^\alpha} \times b\left(n^{1-\alpha}\left(x - \frac{k}{n}\right)\right) \quad \text{where} \quad I = \int_{-c}^{c} b(t)\,dt$$

and $0 < \alpha < 1$, converges to $f(x)$ in $\mathbf{L}^2(\Omega, \mathbf{A}, \mathbf{P})$, uniformly on compacta with respect to x.

In other words, for any compact set K, there exists a sequence c_n tending to zero positively such that, for all x in K, the expected error is bounded by c_n:

$$E([f(x) - f_n(x)]^2) \leq c_n.$$

Notes.

(a) In fact, we have also the following property: $f_{n^2}(x)$ converges almost surely to $f(x)$ because the sum of the variances is finite (Feller, 1968).
(b) We can notice that the nearest to 1 is the parameter α, the most robust is the network with respect to noise. In that case, because the bell-shaped functions are flattened, we lose in precision what we get in robustness.

4.2. The Cases of Gaussian Noises

We will now consider the case of gaussian noises with zero means. This particular case is in fact a standard assumption, and we can say a little more than in the general part above. With the same notation as in Theorem 3:

THEOREM 4. *If for all n, the noises $(e_{k,n})_k$ are independent gaussian random variables, with zero means and bounded variances in the aggregate, then the sequence*

$$f_n(x) = \sum_{k=-n^2}^{n^2} \frac{f\left(\frac{k}{n}\right) + e_{k,n}}{I.n^\alpha} b\left(n^{1-\alpha}\left(x - \frac{k}{n}\right)\right)$$

converges almost surely to $f(x)$, uniformly on compacta with respect to x.

In other words, for a compact set K, the probability for the sequence f_n to converge to the function f uniformly on the variable $x \in K$, is equal to one.

5. CASE OF HIGHER DIMENSION

We have already given a theorem in any dimension, in the case where the network contains "product" neurons. Unfortunately, classical neural networks do not have any "product" neuron. This is the reason

THEOREM 6. *Let f be in $\mathbf{C}^1$, with a bounded derivative, and S be a squashing function verifying $S(0) = 0$ and having a bell-shaped derivative. Then we can approach f, uniformly on compacta, with the functions:*

$$F_n(x) = f(0) + \sum_{k=-n^2}^{n^2} \frac{f'\left(\frac{k}{n}\right)}{I.n} S\left(n^{1-\alpha}\left(x - \frac{k}{n}\right)\right).$$

Furthermore, the functions $f'_n(x)$ converge uniformly on compacta to $f'(x)$.

6.2. Case of Dimension Greater than 1

To approach a function of several variables and its first derivatives, we must face new difficulties: (a) If we try to approximate only $\partial/\partial x^1\ F(x)$ we stand no chance, even in 2-dimension, to approach the other derivatives. Therefore, we must try to have all derivatives at the same time. The only way to do this seems to approach $\partial^p/\partial x^1 \ldots \partial x^p\ F(x)$, and to integrate after. With this solution, we have to face another problem: (b) How can we build, with squashing functions, some function G such that $\partial^p/\partial x^1 \ldots \partial x^p\ G(x)$ is a multidimensionnal bell-shaped function? If the network has a $\Sigma\Pi$-units, the functions: $(x^1, \ldots x^p) \rightarrow S(x^1) \ldots S(x^p)$ where S is a primitive of a bell-shaped function, are solutions to this problem. The following Theorem 6 bis will then give an explicit $\Sigma\Pi$-network. We have no answer in the general case, for Σ-networks. Let us just recall that in the section above, we have done a similar construction in 2-dimension:

$$G: (x, y) \rightarrow th(th(x) + th(y) - 0.5)$$

which is also a solution to this problem (see example 3, Section 5). The Theorem 6 bis gives an explicit Σ-network in 2-dimension. Yet, this construction is difficult to generalize in further dimensions: (c) Anyway, we have a result concerning the *existence* of an approximation with a one layer feedforward Σ-network in any dimension. Indeed let f be a function in C^1. We can approach f with a C^∞ function with a compact support g, for the topology of uniform convergence on compacta for the function and its derivative. Trigonometric polynomials are dense in the space of continuous functions for the norm of uniform convergence on compacta, and then can approach the function $\partial^p/\partial x^1 \ldots \partial x^p\ g(x)$. After integration, we obtain a sequence of trigonometric polynomials (which can be written as a sum of cosines of linear combinations in $x^1, \ldots, x^p$) which converges to g and whose gradient converges to the gradient of g. Therefore, we can approach the function f by linear combination of cosines for the same topology. According to Theorem 6, we can approximate the cosine function and its derivative with a one layer feedforward neural networks. The approximation of the function f and its gradient then follows.

Anyway, when the problem of finding an appropriate function G has been solved (as in 2-dimension), we have an explicit formula of approximation. We give it in 2-dimension, the case of higher dimension being generalized easily.

THEOREM 6 bis. *Let f and $G: \mathbf{R}^2 \rightarrow \mathbf{R}$ be in $\mathbf{C}^2$, such that the derivative $\partial^2/\partial x \partial y\ G(x, y)$ is a multidimensionnal bell-shaped function, and let $\mathbf{S}: \mathbf{R} \rightarrow \mathbf{R}$, be such that its derivative is bell-shaped. Then the functions*

$$\begin{aligned} f_n(x, y) = f(0, 0) &+ \sum_{k^1, k^2 = -n^2}^{n^2} \frac{1}{In^2} \frac{\partial^2}{\partial x \partial y} f\left(\frac{k^1}{n}, \frac{k^2}{n}\right) \\ &\times G\left(n^{1-\alpha}\left(x - \frac{k^1}{n}\right), n^{1-\alpha}\left(y - \frac{k^2}{n}\right)\right) \\ &+ \sum_{k=-n^2}^{n^2} \frac{1}{I'n} \left[\frac{\partial}{\partial x} f\left(0, \frac{k}{n}\right) S\left(n^{1-\alpha} \cdot \left(y - \frac{k}{n}\right)\right)\right. \\ &\left. + \frac{\partial}{\partial y} f\left(\frac{k}{n}, 0\right) S\left(n^{1-\alpha}\left(x - \frac{k}{n}\right)\right)\right] \end{aligned}$$

where I and I' are, respectively the integrals of $\partial^2/\partial x \partial y\ G(x, y)$ and S', converge uniformly on compacta to f, just as their differential to the differential of f.

6.3. Formulas for Back Propagation of the Error on the Derivatives

For a function with more than 2 variables, we proved the existence of a network with no $\Sigma\Pi$ units that approximates the function with its gradient. But we do not have explicit formulas to build the network. We suggest to use the back propagation algorithm that we extend here to take into account the error on the derivatives. Yet, the drawback of back propagation then remains: there is no topology such that the network is proved to converge to the initial function as the number of examples increases.

We consider a feedforward neural network with n layers, which takes as input a vector (x_i^1). Let us denote V_j^k and x_j^k the potential and the state of neuron j from layer k, having activation function $S()$. We also denote y_{jq}^k the partial derivative of the state x_j^k with respect to the q'th input x_q^1:

$$y_{jq}^k = \frac{\partial x_j^k}{\partial x_q^1}.$$

The direct propagation is done according to the rule:

$$V_i^{k+1} = \sum_j w_{ij}^k x_j^k$$

$$x_i^{k+1} = S(V_i^{k+1})$$

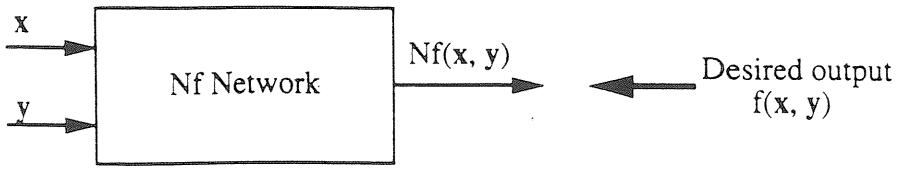

FIGURE 3. The first step to approach a function ϕ such that $f(x, \phi(x)) = 0$ with a neural network is to build N_f, an approximation of f.

distributed, and if $N_f(x_p, N_\phi(x_p))$ is close to zero for all p, then the network N_ϕ is a good approximation of ϕ (Theorem 10). The boundness of N_ϕ can be obtained if, in the learning phase, the weights of this network are kept within a given limit. This can be done with the cost function used for back propagation for instance.

7.2. The Case of a Single Approximation of the Function f

We will show here why it is necessary to force the network N_f to approach both f and its derivative.

In this section, K_x will be a cartesian product of n segments from $\mathbf{R}$, K_y a product of p segments from $\mathbf{R}$, K_y^* the interior of K_y, $K = K_x \times K_y$ and $f(x, y)$ a C^1 function defined on the compact set K, to $\mathbf{R}^p$, such that, for all (x, y) in K, the differential of f with respect to y, $\partial f/\partial y\ (x, y)$, is regular. It is also supposed that, for all x in K_x, there exists one and only one y in K_y^* such that $f(x, y) = 0$. One then has the function

$$\phi: K_x \to K_y^*$$
$$x \to y \text{ such that } f(x, y) = 0.$$

According to the implicit functions theorem, ϕ is C^1 in K_x, and its differential with respect to x is

$$\frac{\partial \phi}{\partial x}(x) = -\left[\frac{\partial f}{\partial y}(x, \phi(x))\right]^{-1} \frac{\partial f}{\partial x}(x, \phi(x)).$$

But the condition

$$|N_f(x, y) - f(x, y)| < \varepsilon$$

does not ensure the network N_f to fullfill the regularity conditions needed by the implicit functions theorem. If the derivative of N_f with respect to y is not regular, it does not hold, and a C^1 function $\Psi(x)$ verifying

$$N_f(x, \Psi(x)) = 0$$

may not exist. Typically, a "catastrophic" phenomenon can happen, as shown in Figure 5.

7.3. Theorems

We will now see that the problem can be solved if both f and its derivatives are approached. The proofs of the theorems are given in the appendix. Let us first introduce some notations.

Notations. We denote by $C^1(K)$ the space of the C^1 functions from K to $\mathbf{R}^p$, with the norm

$$\|f\|_1 = \text{Sup}\left\{\|f\|_\infty, \left\|\frac{\partial f}{\partial x}\right\|_\infty, \left\|\frac{\partial f}{\partial y}\right\|_\infty\right\}.$$

Let Ω be the subset of $C^1(K)$ containing the functions f such that, first, for all (x, y) in K, $\partial f/\partial y\ (x, y)$ is regular, and second, for all x in K_x, the equation $f(x, y) = 0$ has one and only one solution in K_y^*. At last, $C^1(K_x)$ is the space of the C^1 functions from K_x to $\mathbf{R}^p$. Its norm is the infinite norm.

The properties of the set Ω imply that there exists an application

$$\Psi: \Omega \to C^1(K_x)$$
$$f \to \Psi(f) = \phi \text{ such that } f(x, \phi(x)) = 0 \text{ on } K_x.$$

We can then give the four following theorems:

THEOREM 7. *The set Ω of the C^1 functions f on K such that*

for all (x, y) in K, $\frac{\partial f}{\partial y}(x, y)$ is regular

for all x in K_x, the equation $f(x, y) = 0$ has one and only one solution in K_y^*

is an opened subset of $C^1(K)$

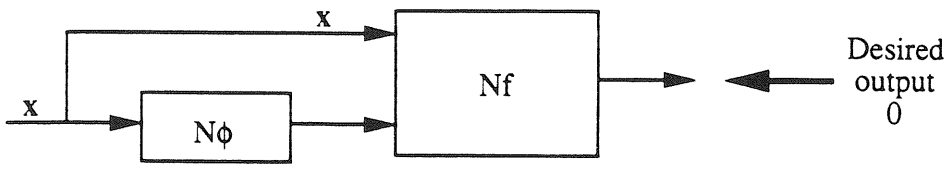

FIGURE 4. The second step to approach a function ϕ such that $f(x, \phi(x)) = 0$ with a neural network. Back propagation is used, with zero desired output, but the weights of N_f are those found in the first step and are not modified. Only the N_ϕ's one are updated.

In practice, N_ϕ is learnt by back propagation (see Figure 4). At the end of the algorithm, the real numbers $N_f(x_p, N_\phi(x_p))$ are small for all p. If the points x_p are sufficiently close to each other and sufficiently distributed, we can conclude that $\|N_\phi - \Psi(f)\|_\infty$ is small.

For Theorem 10, we could not provide a relation giving the real number η from ε, which would be interesting in practice, it can be seen in the proof of this theorem that η depends on ε, and on the real numbers A and μ from Lemmas 1 and 3. Whether these numbers A and μ can be evaluated from the network N_f is an opened question.

8. CONCLUSION

We tried, in this paper, to give results as constructive as possible. Thus, we raised and explicited the properties that are expected from neural networks: possibility of interpolation and of approximation, noise robustness derived from the information redundancy and, of course, parallelism.

We have continued the works of Cybenko, (1989); of Funahashi, (1989); and of Hornik et al., (1989, 1990) about function approximations in giving explicit formulas and in proving that neural networks can approach both a function and its differential. Particular functions, the bell-shaped functions, have a crucial importance in this paper. We believe that many of the neural network's properties are due to those of the bell-shaped functions.

In giving approximation's formulas, we also tried to highlight the network's architecture problem: the number of hidden units which, we have seen, determines how well the function will be approximated, and the organization in the network (for example, use of 3 neurons cellulas to build bell-shaped functions, or more complex cellulas with $\Sigma\Pi$ units, to approximate a function and its derivatives).

We have developed an important application, showing how to approach an implicit function. In that purpose, we mixed classical algorithms (gradient back propagation for instance), and new methods described in this paper.

Yet, some problems are still opened: we have seen how difficult it is to approximate both a function and its derivatives when there are more than one variable. This is mainly because it is necessary to approach the p-th derivative with respect to the p variables; it would be useful to avoid this. We have no explicit approximation for a function and its derivatives by Σ networks with more than two variables. That is why we gave an extension of the back propagation algorithm, to take into account the derivatives of the network's approximation. At least, examples of the function's derivatives may be impossible to have. In this case, the ideas presented in this paper may still be used to impose regularity conditions on the approximation.

To conclude, we want to emphasize the similarities between the theories described here and the signal processing. On one hand it is quite easy to build bell-shaped functions $b(x)$ such that the value of the sum, for k in Z, of the translated functions $b(x - k)$, is 1. It should then be possible to find formulas to decompose a given function as a sum of such bell-shaped functions. This would be similar to the Y. Meyer's multiresolution analysis (see, for instance, Mallat, 1989).

On the other hand, the approximation of a function with its derivatives should be helpful for recognition and classification of dynamical systems. For instance, a system's stabilizer could be searched (that is, a function which stays constant if and only if the received signal is a solution of the system's dynamical equation), and could be approximated.

REFERENCES

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematical Control Signals Systems*, **2**, 303–314.

Dieudonné, J. (1980). *Calcul infinitésimal*. Paris: Hermann.

Feller, W. (1968). *An introduction to probability theory and its applications*, (Vol. I). New York: Wiley & Sons.

Funahashi, K. (1989). On the approximate realization of continuous mappings by neural networks. *Neural Networks*, **2**, 183–192.

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, **2**, 359–366.

Hornik, K., Stinchcombe, M., & White, H. (1990). Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks. *Neural Networks*, **3**, 551–560.

Mallat, S. (1989). A theory for multiresolution signal decomposition: The wavalet representation. *I.E.E.E. Transactions on Pattern Analysis and Machine Intelligence*, **11**(7), 674–693.

Meyer, Y. (1990). *Ondelettes et opérateurs*, **I**. Paris: Hermann.

Poggio, T. & Girosi, F. (1989). *A Theory of networks for approximations and learning*. (A.I. Memo No. 1140, C.B.I.P. Paper No. 31). Technical Report, Cambridge: MIT.

Nguyen, D., & Widrow, B. (1989). The truck backer-upper: an example of self-learning in neural networks. *Proceedings of the Joint International Conference on Neural Networks*, **II**, 357–363.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning internal representation by error propagation. In D. E. Rumelhart & J. L. McClelland (Eds.), *Parallel distributed processing*, **I** (pp. 318–362). Cambridge, MA: MIT Press.

APPENDIX

Proof of Theorem 1. According to Heine theorem, the function f is uniformly continuous on $[a, b]$. Let $\varepsilon > 0$ be a real number and η be the coefficient of uniform continuity on $[a, b]$. We prove that if

$$\text{Sup}_i |x_{i+1}^n - x_i^n| < \eta$$

To prove the Theorems 7 to 10, we need the following three lemmas:

LEMMA 1. *If f is a function in Ω, there exists a real number $A > 0$ such that, for all (x, y) in K and for all h in $\mathbf{R}^p$,*

$$\left\| \frac{\partial f}{\partial y}(x, y)h \right\| > A \|h\|$$

Proof. We denote by Σ the unit sphere in $\mathbf{R}^p$, and we consider the function

$$K_1 \times K_2 \times \Sigma \longrightarrow \mathbf{R}$$
$$(x, y, h) \longrightarrow \left\| \frac{\partial f}{\partial y}(x, y)h \right\|$$

which is continuous on the compact set $K_1 \times K_2 \times \Sigma$. Since it is strictly positive on its domain, it is inferiorly bounded by a strictly positive real number A. ∎

LEMMA 2. *Let f be in Ω and A be a real number given by Lemma 1. There exists a real number $\alpha > 0$ such that, for all (x, y) in K and for all h in $\mathbf{R}^p$ verifying*

$$\|h\| \leq \alpha \text{ and } y + h \text{ is in } K_2,$$

one has the inequality

$$\|f(x, y + h) - f(x, y)\| \geq \frac{A}{2} \|h\|$$

Proof. This inequality follows easily from the triangular inequality, from (1):

$$\left\| f(x, y + h) - f(x, y) - \frac{\partial f}{\partial y}(x, y)h \right\| \leq \frac{A}{2} \|h\| \qquad (1)$$

and from the fact that

$$\left\| \frac{\partial f}{\partial y}(x, y)h \right\| \geq A \|h\|.$$

To prove (1), let us consider x, y, h and the function

$$[0, 1] \longrightarrow \mathbf{R}^p$$
$$t \longrightarrow f(x, y + th) - f(x, y) - \frac{\partial f}{\partial y}(x, y) \cdot (th).$$

Applying the fundamental theorem of calculus to this function, we get

$$\left\| f(x, y + h) - f(x, y) - \frac{\partial f}{\partial y}(x, y)h \right\| \leq \|h\| \operatorname{Sup}\left\{ \left\| \frac{\partial f}{\partial y}(x, y + th) - \frac{\partial f}{\partial y}(x, y) \right\|, t \in [0, 1] \right\}.$$

Since $\partial f / \partial y$ is uniformly continuous on the compact set K, one can get a real number $\alpha > 0$ such that, for $\|h\| \leq \alpha$, one has

$$\operatorname{Sup}\left\{ \left\| \frac{\partial f}{\partial y}(x, y + th) - \frac{\partial f}{\partial y}(x, y) \right\|, t \in [0, 1] \right\} \leq \frac{A}{2}.$$

Inequality (1) just follows.

LEMMA 3. *For f in Ω, and α given by Lemma 2, there exists a $\mu > 0$ such that any (x, y) verifying*

$$\|y - \Psi(f)(x)\| \geq \alpha/2$$

verifies also

$$\|f(x, y)\| \geq \mu.$$

Proof. This lemma results from the fact that the set K' of the points (x, y) in K such that

$$\|y - \Psi(f)(x)\| \geq \alpha/2$$

is compact and that $\|f(x, y)\|$ is nonzero on K'. ∎

Proof of Theorem 7. Let f be in Ω, A, α, and μ be the real numbers given by the 3 lemmas, and K' be the set of the points (x, y) in K verifying

$$\|y - \Psi(f)(x)\| \geq \alpha/2.$$

Pick $\varepsilon > 0$ such that

$\varepsilon < \alpha$ and

$\varepsilon < \operatorname{Inf}\{d(\Psi(f)(x), \partial K_2), x \in K_1\}$ (∂K_2 is the boundary of K_2)

and pick g in $C^1(K)$ such that

$$\|f - g\|_1 < \operatorname{Inf}\left\{ \frac{A}{4}, \frac{\varepsilon A}{4}, \frac{\mu}{2} \right\}.$$

We will prove that g belongs to Ω:

i. Regularity of $\frac{\partial g}{\partial y}(x, y)$:

For all couple (x, y) in K and for all vector u in $\mathbf{R}^p$, one has

$$\left\| \frac{\partial g}{\partial y}(x, y)\, u \right\| \geq \left\| \frac{\partial f}{\partial y}(x, y)u \right\| - \left\| \left[\frac{\partial g}{\partial y}(x, y) - \frac{\partial f}{\partial y}(x, y) \right] u \right\| \geq \frac{3A}{4} \|u\| > 0 \text{ if } u \neq 0$$

ii. Existence of a solution to the equation $g(x_0, y) = 0$:

Let x_0 be in K_1, y_0 be $\Psi(f)(x_0)$ and F be the ε radius closed ball centered at y_0. We proceed by contradiction and suppose that 0 is not in $g(x_0, F)$. Since F is compact, the set $g(x_0, F)$ is compact, and therefore closed. Let ζ be the function

$$\zeta: [0, 1] \longrightarrow \mathbf{R}^p$$
$$t \longrightarrow tg(x_0, y_0)$$

The set $\zeta^{-1}[g(x_0, F)]$ is closed; it contains 1 but not 0. We can then define

$$T = \operatorname{Inf} \zeta^{-1}[g(x_0, F)]$$

which is in $\zeta^{-1}[g(x_0, F)]$, and Y an element in F such that

$$g(x_0, Y) = \zeta(T) = Tg(x_0, y_0).$$

We can say that Y is on the boundary of F. Indeed, if Y was in the interior of F, since $\partial g / \partial y\,(x_0, Y)$ is regular, the implicit functions theorem would hold and $g(x_0, y)$ could be inverted from a neighbourhood of Y (included in F if Y is in the interior of F) to a neighbourhood of $g(x_0, Y)$; this contradicts the fact that $g(x_0, y_0)$ has no antecedent in F as soon as $t < T$. It follows that Y is on the boundary of F and satisfies

$$\|Y - y_0\| = \varepsilon < \alpha$$

Applying Lemma 2,

$$\|f(x_0, Y)\| = \|f(x_0, Y) - f(x_0, y_0)\| \geq \frac{A}{2} \|Y - y_0\| \geq \frac{\varepsilon A}{2}$$

one also has

$$\|g(x_0, Y)\| = T \|g(x_0, y_0)\| = T \|g(x_0, y_0) - f(x_0, y_0)\| \leq \|g - f\|_1 < \frac{\varepsilon A}{4}$$

and, at last

$$\|f(x_0, Y) - g(x_0, Y)\| \leq \|f - g\|_1 < \frac{\varepsilon A}{4}.$$

Those three inequalities are in contradiction with the triangular inequality:

$$\|f(x_0, Y)\| \leq \|f(x_0, Y) - g(x_0, Y)\| + \|g(x_0, Y)\|.$$