UNITED NATIONS EDUCATIONAL, SCIENTIFIC AND CULTURAL ORGANIZATION
INTERNATIONAL CENTRE FOR THEORETICAL PHYSICS
I.C.T.P., P.O. BOX 586, 34100 TRIESTE, ITALY, CABLE: CENTRATOM TRIESTE

H4.SMR/473-11

COLLEGE ON NEUROPHYSICS

"Neural correlates of behaviour, development, plasticity and memory"

1-19 October 1990

Regularization algorithms for learning that are equivalent to multilayer networks

Tomaso Poggio and F. Girosi

M.I.T. Cambridge, USA

Reprint Series
23 February 1990, Volume 247, pp. 978–982

SCIENCE

Regularization Algorithms for Learning That Are Equivalent to Multilayer Networks

T. POGGIO AND F. GIROSI

Regularization Algorithms for Learning That Are Equivalent to Multilayer Networks

T. POGGIO AND F. GIROSI

Learning an input-output mapping from a set of examples, of the type that many neural networks have been constructed to perform, can be regarded as synthesizing an approximation of a multidimensional function (that is, solving the problem of hypersurface reconstruction). From this point of view, this form of learning is closely related to classical approximation techniques, such as generalized splines and regularization theory. A theory is reported that shows the equivalence between regularization and a class of three-layer networks called regularization networks or hyper basis functions. These networks are not only equivalent to generalized splines but are also closely related to the classical radial basis functions used for interpolation tasks and to several pattern recognition and neural network algorithms. They also have an interesting interpretation in terms of prototypes that are synthesized and optimally combined during the learning stage.

MOST NEURAL NETWORKS ATtempt to synthesize modules that transduce inputs into desired outputs from a set of correct input-output pairs, called examples. Some of the best known applications are a network that maps English spelling into its phonetic pronunciation (*1*) and a network that learns the mapping corresponding to a chaotic dynamical system, thereby predicting the future from

Artificial Intelligence Laboratory, Center for Biological Information Processing, Massachusetts Institute of Technology, Cambridge, MA 02139.

the past (*2*). In these cases, learning takes place when the weights of connections in a multilayer network of simple units are changed, according to a gradient descent scheme called backpropagation (*3*). It would be highly desirable to establish theoretical foundations for using multilayer networks of this general type to learn from examples. To show how this goal can be achieved, we first explain how to rephrase the problem of learning from examples as a problem of approximating a multivariate function.

To illustrate the connection, let us draw an analogy between learning an input-output mapping and a standard approximation problem, two-dimensional (2-D) surface reconstruction from sparse data points. Learning simply means collecting the examples, that is, the input coordinates x_i, y_i and the corresponding output values at those locations, the heights of the surface d_i. Generalization means estimating d at locations x, y where there are no examples, that is, no data. This requires interpolating or, more generally, approximating the surface (the function) between the data points (interpolation is the limit of approximation when there is no noise in the data). In this sense, learning is a problem of hypersurface reconstruction (*4*, *5*).

From this point of view, learning a smooth mapping from examples is clearly an ill-posed problem (*6*), in the sense that the information in the data is not sufficient to reconstruct uniquely the mapping in regions where data are not available. In addition, the data are usually noisy. A priori assumptions about the mapping are needed to make the problem well-posed. One of the simplest assumptions is that the mapping is smooth: small changes in the inputs cause a small change in the output (*7*).

Techniques that exploit smoothness constraints in order to transform an ill-posed problem into a well-posed one are well known under the term of regularization theory (*6*, *8*). Consider the inverse problem of finding the hypersurface $f(\mathbf{x})$, given its value d_i on a finite set of points $\{\boldsymbol{\xi}_i\}$ of its domain. This problem is clearly ill-posed because it has an infinite number of solutions, and some constraint must be imposed on the solution. A standard technique in regularization theory solves the problem by minimizing a cost functional consisting of two terms. The first term measures the distance between the data and the desired solution f; the second term measures the cost associated with the deviation from smoothness. Its form is $\|Pf\|^2$, where P is usually a differential operator, called a stabilizer, and $\|\cdot\|$ is a norm on the function space to which Pf belongs (usually the L^2 norm). The term is small for smooth f whose derivatives have small norms. Thus, the method selects the hypersurface f that solves the variational problem of minimizing the functional

$$H[f] = \sum_{i=1}^{N} (d_i - f(\boldsymbol{\xi}_i))^2 + \lambda \|Pf\|^2 \quad (1)$$

where d_i are the values of the hypersurface at the given N points $\boldsymbol{\xi}_i$, and λ, the regularization parameter, controls the compromise between the degree of smoothness of the solution and its closeness to the data (*9*). For instance, in one dimension with

$$\|Pf\|^2 = \int_R dx \left[\frac{d^2 f(x)}{dx^2}\right]^2 \quad (2)$$

the function $f(x)$ that minimizes the functional of Eq. 1 is a "cubic spline," a curve that is a cubic polynomial between the knots, with continuous second-order derivative at the knots (*10*).

The formulation of the learning problem

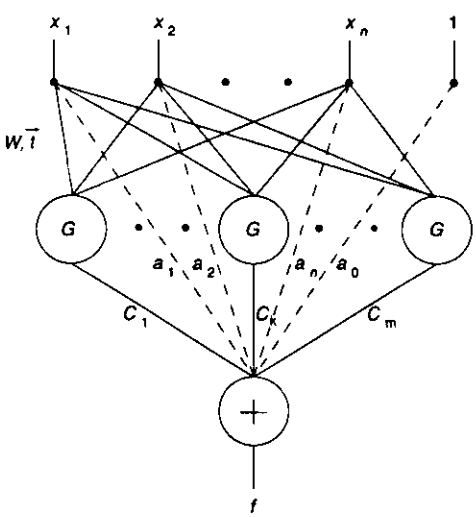

Fig. 1. The HyperBf network used to approximate a mapping between $x_1, x_2, \ldots, x_n$ and f, given a set of sparse, noisy data. The data, a set of points for which the value of the function is known, can be considered as examples to be used during learning. The hidden units evaluate the function $G(\mathbf{x};\mathbf{t}_n)$, and a fixed, nonlinear, invertible function may be present after the summation. The units are, in general, fewer than the number of examples. The parameters that may be determined during learning are the coefficients c_n, the centers $\mathbf{t}_n$, and the matrix W. In the radial case, $G = G(\|\mathbf{x} - \mathbf{t}_n\|^2_W)$ and the hidden units simply compute the radial basis functions G at the "centers" $\mathbf{t}_n$. The RBFs may be regarded as matching the input vectors against the "templates" or "prototypes" that correspond to the centers (consider, for instance, a radial Gaussian around its center, which is a point in the n-dimensional space of inputs). Updating a center $\mathbf{t}_n$ during learning is equivalent to modifying the corresponding prototype. Changing the weights W corresponds to performing dimensionality reduction on the input features. In addition to the linear combination of basis functions, the figure includes other terms that contribute to the output: constant and linear terms are shown here as direct connections from the input to the output with weights $a_0, a_1, a_2, \ldots a_n$ (*37*).

in terms of regularization is satisfying from a theoretical point of view, because it establishes connections with a large body of results in the area of Bayesian estimation and in the theory of approximation of multivariate functions (*11*). In particular, Eq. 1 can be used to define generalized splines in any dimension. At this point, it is natural to ask about the connection between this perspective on learning as an approximation problem and feedforward networks, such as backpropagation, that have become popular recently, exactly because of their capabilities to "learn from examples."

In the following, we provide an answer to the previous question by showing that the solution to the approximation problem given by regularization theory can be expressed in terms of a class of multilayer networks that we call regularization networks or hyper basis functions (HyperBFs) (see Fig. 1) and that are similar to previously suggested networks (*12*, *13*). Our main result is that the regularization approach is equivalent to an expansion of the solution in terms of a certain class of functions that depends only on the form of the stabilizing operator. We explain how this expansion can be interpreted in terms of a network with one layer of hidden units whose characteristics are dictated by the theory. We also discuss a computationally efficient scheme for synthesizing the associated network from a set of examples, which has an interesting interpretation and several promising extensions.

We outline first how an approximation in terms of a specific class of functions, often radial, can be derived directly from regularization. The regularization approach selects the function f that solves the variational problem of minimizing the functional of Eq. 1. It can be proved (*5*) that the solution has the following simple form:

$$f(\mathbf{x}) = \sum_{i=1}^{N} c_i\, G(\mathbf{x};\boldsymbol{\xi}_i) \quad (3)$$

where $G(\mathbf{x})$ is the Green's function of the self-adjoint differential operator $\hat{P}P$, $\hat{P}$ being the adjoint operator of P, and the coefficients c_i satisfy a linear system of equations that depend on the N "examples," that is, the data to be approximated (*14*). If P is an operator with radial symmetry, the Green's function G is radial and therefore the approximating function becomes:

$$f(\mathbf{x}) = \sum_{i=1}^{N} c_i\, G(\|\mathbf{x} - \boldsymbol{\xi}_i\|^2) \quad (4)$$

which is a sum of radial functions, each with its center $\boldsymbol{\xi}_i$ on a distinct data point. Thus the number of radial functions, and corresponding centers, is the same as the number of examples.

Our derivation shows that the type of

basis functions depends on the stabilizer P, that is, on the specific a priori assumption (*5*). Depending on P we obtain the Gaussian $G(r) = e^{-(r/c)^2}$, the well-known "thin-plate spline" $G(r) = r^2 \ln r$, and other specific functions, radial or not (*15*). As observed by Broomhead and Lowe (*12*) in the radial case, a superposition of functions such as that in Eq. 3 is equivalent to a network of the type shown in Fig. 1. The interpretation of Eq. 4 is simple: in the 2-D case, for instance, the surface is approximated by the superposition of, say, several 2-D Gaussian distributions, each centered on one of the data points.

Equation 4 has the same form as an interpolation technique, called radial basis functions (RBFs), that has been extensively studied (*16*). In 1986 Micchelli proved a powerful result that justifies the use of a large class of functions as interpolating RBFs (*17, 18*). It turns out (*5*) that the class of radial functions satisfying Micchelli's condition is closely related to the larger class of functions defined by Eq. 1.

The network associated with Eq. 4 has a complexity (number of radial functions) that is independent of the dimensionality of the input space but is on the order of the dimensionality of the training set (number of examples), which is usually high. Broomhead and Lowe (*12*) used fewer centers than data points. A heuristic scheme with movable centers and Gaussian functions has also been proposed and tested (*13*). It turns out that our previous rigorous result can be extended in a natural way to a scheme in which the number of centers is much smaller than the number of examples. In the framework of regularization the consistent extension we derive has the feature of center positions that are modified during learning (*5*). The extension is

$$f^*(\mathbf{x}) = \sum_{\alpha=1}^{n} c_\alpha G(\mathbf{x};\mathbf{t}_\alpha) \qquad (5)$$

where the parameters $\mathbf{t}_\alpha$, which we call "centers" in the radial case, and the coefficients c_α are unknown and are in general fewer than the data points ($n \leq N$) (*19*). Equation 5, which can be implemented by the network of Fig. 1, is equivalent to generalized splines with free knots, whereas Eq. 4 is equivalent to generalized splines with fixed knots. This scheme can be further extended by considering in Eq. 5 the superposition of different types of functions G, such as Gaussians at different scales (*20*). In addition, the norm $\|\mathbf{x} - \boldsymbol{\xi}\|$ may be considered as a weighted norm

$$\|\mathbf{x} - \boldsymbol{\xi}_i\|_W^2 = (\mathbf{x} - \boldsymbol{\xi}_i)^T W^T W (\mathbf{x} - \boldsymbol{\xi}_i) \qquad (6)$$

where W is a matrix and the superscript T indicates the transpose. In the simple case of diagonal W, the diagonal elements w_i assign a specific weight to each input coordinate. They play a critical role whenever different types of inputs are present. Iterative methods of the gradient descent type can be used to find the optimal values of the various sets of parameters, the c_α, the w_{ij}, and the $\mathbf{t}_\alpha$, that minimize an error functional on the set of examples. Since this functional is no longer convex, a stochastic term in the gradient descent equations may be used to avoid local minima (*21*).

The network of Fig. 1 may be interpreted as follows. The centers of the radial functions are similar to prototypes, since they are points in the multidimensional input space. Each unit computes a (weighted) distance of the inputs from its center, which is a measure of their similarity, and applies to it the radial function. In the case of the Gaussian, a unit will have maximum activity when the new input exactly matches its center. The output of the network is the linear superposition of the activities of all the radial functions in the network. One finds the corresponding weights during learning by minimizing a measure of the error between the network's prediction and each of the examples. At the same time, the centers of the radial functions and the weights in the norm are also updated during learning. Moving the centers is equivalent to modifying the corresponding prototypes and corresponds to task-dependent clustering. Finding the optimal weights for the norm is equivalent to transforming appropriately, for instance, scaling, the input coordinates and corresponds to task-dependent dimensionality reduction.

Figure 2 shows a specific application of HyperBFs. Consider the problem of recognizing a wire-frame 3-D object from any of its perspective views. A view of the object is represented as a $2N$ vector $x_1, y_1, x_2, y_2, \ldots, x_N, y_N$ of the coordinates on the image plane of N labeled and visible points on the object. Additional different types of features can also be used, such as angles between vertices. The network learns to map any view of the object into a standard view. The results with images generated with computer graphics tools (of the type indicated in Fig. 2B) are encouraging and have promising extensions to more realistic data (*22*).

Many existing schemes for networks that learn are encompassed by the HyperBF framework (*5*). Past work, in the special case of fixed centers, indicates good performance in a number of tasks (*23*). Our own preliminary work, as well as earlier experiments of Moody and Darken with a similar network (*13*), suggests that the more general form of HyperBFs has a promising performance.

The scheme is a satisfying theory of networks for learning. HyperBFs are the feedforward network versions of regularization and are therefore equivalent to generalized splines. The HyperBF network is similar to the architecture used for backpropagation, being a multilayer network with one hidden

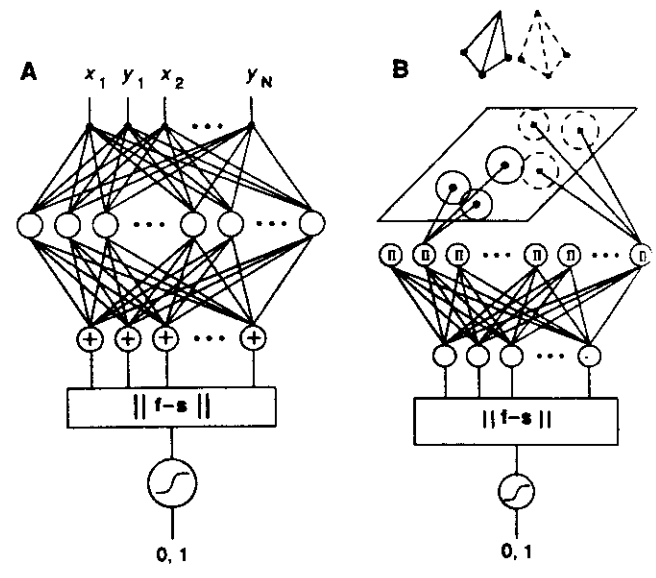

Fig. 2. (**A**) The HyperBF network proposed for the recognition of a 3-D object from any of its perspective views. The network attempts to map any view (as defined in the text) into a standard view, arbitrarily chosen. The norm of the difference between the output vector **f** and the standard view **s** is thresholded to yield a 0, 1 answer. The $2N$ inputs accommodate the input vector **v** representing an arbitrary view. Each of the K RBFs is initially centered on one of a subset of the M views used to synthesize the system ($K \leq M$). During training each of the M inputs in the training set is associated with the desired output, the standard view **s**. (**B**) A completely equivalent interpretation of (A) for the special case of Gaussian RBFs. Gaussian functions can be synthesized by multiplying the outputs of 2-D Gaussian receptive fields that "look" at the retinotopic map of the object point features. The solid circles in the image plane represent the 2-D Gaussians associated with the first RBF, which represents the first view of the object. The dotted circles represent the 2-D receptive fields that synthesize the Gaussian RBF associated with another view. The 2-D Gaussian receptive fields transduce positions of features, represented implicitly as activity in a retinotopic array, and their product "computes" the radial function without the need to calculate norms and exponentials explicitly. See (*5*) for more details.

layer and two or even three sets of adjustable parameters. Its Boolean limiting version carves the input space into hyperspheres, each corresponding to a center: a radial unit is active if the input vector is within a certain radius of its center and is otherwise silent. The Boolean limit of backpropagation carves the space with hyperplanes. With an arbitrary number of units each network can approximate the other, since each network can approximate arbitrarily well continuous functions on a limited interval (*24, 25*). Multilayer networks with sigmoid units do not have, however, the best approximation property that regularization networks have (*25*). The Boolean limit of HyperBF is almost identical to Kanerva's associative memory algorithm (*26*), which is itself closely related to vector quantization. Parzen windows, potential techniques in pattern recognition, and kernel estimation methods, in general (*27*), can be regarded as special cases of the HyperBF method. Close analogies between Kanerva's model and Marr's (*28*) and Albus's (*29*) models of the cerebellum also exist (*5, 30*). The update equation that controls the evolution of the centers $\mathbf{t}_\alpha$ [see Eq. 14 in (*21*)] is also similar to Kohonen's topology-preserving algorithm (*5, 31*) [which is also similar to the *k*-means algorithm (*32*)] and can be interpreted as a learning scheme in which the centers of the radial functions move to find centers of clusters of input vectors (*33*). Coarse coding techniques and product units (*34*) can be interpreted neatly within the HyperBF framework (for the special case of Gaussian RBFs) (*5, 35*).

Thus HyperBFs represent a general framework for learning smooth mappings that rigorously connects approximation theory and regularization with feedforward multilayer networks. In particular, it suggests that the performance of networks of this general type can be understood in the framework of classical approximation theory, providing limits on what feedforward networks may be expected to perform (*5*).

In the Gaussian case, it also suggests a scheme for learning a large class of mappings that has intriguing features from the point of view of a brain scientist, since the overall computation is a simple but powerful extension of a look-up table, that is, a memory, and can be performed by the superposition of "units," in the appropriate multidimensional input space. These units would be somewhat similar to "grandmother" filters with a graded response, rather than binary detectors, each representing a prototype. They would be synthesized as the conjunction of, for instance, 2-D Gaussian receptive fields looking at a retinotopic map of features (see Fig. 2B). During learning, the weights of the various prototypes in the network output are modified to find the optimal values that minimize the overall error. The prototypes themselves are slowly changed to find optimal prototypes for the task. The weights of the different input features are also modified to perform task-dependent dimensionality reduction.

A scheme of this type is broadly consistent with recent physiological evidence [see, for instance, (*36*)] on face recognition neurons in the monkey inferotemporal cortex. Some of the neurons described have several of the properties expected from the units of Fig. 2 with a center, that is, a prototype that corresponds to a view of a specific face. A similar scheme could be used to learn other visual tasks, such as the computation of color constancy or shape from shading from a set of examples, although the biological relevance in such cases is more questionable. In any case, it remains to be seen whether some cortical neurons indeed have the multidimensional, possibly Gaussian-like, receptive fields suggested by this approach.

REFERENCES AND NOTES

1. T. J. Sejnowski and C. R. Rosenberg, *Complex Syst.* **1**, 145 (1987).
2. A. Lapedes and R. Farber, *Tech. Rep. LA-UR-87-2662* (Los Alamos National Laboratory, Los Alamos, NM, 1982).
3. D. E. Rumelhart, G. E. Hinton, R. J. Williams, *Nature* **323**, 533 (1986).
4. T. Poggio *et al.*, in *Proceedings Image Understanding Workshop*, L. Bauman, Ed. (Cambridge, MA, April 1988) (Morgan Kaufmann, San Mateo, CA, 1988), pp. 1–12. See also S. Omohundro, *Complex Syst.* **1**, 273 (1987).
5. T. Poggio and F. Girosi, *Artif. Intell. Memo 1140* (Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, 1989).
6. A. N. Tikhonov and V. Y. Arsenin, *Solutions of Ill-Posed Problems* (Winston, Washington, DC, 1977).
7. Other stronger a priori constraints may be known, for instance, that the mapping is linear, or has a positive range or a limited domain, or is invariant under some group of transformations.
8. T. Poggio, V. Torre, C. Koch, *Nature* **317**, 314 (1985); M. Bertero, T. Poggio, V. Torre, *Proc. IEEE* **76**, 869 (1988); J. L. Marroquin, S. Mitter, T. Poggio, *J. Am. Stat. Assoc.* **82**, 76 (1987); G. Wahba, *Spines Models for Observational Data* (Series in Applied Mathematics, Society for Industrial and Applied Mathematics, Philadelphia, 1990), vol. 59, and references therein.
9. The parameter λ is directly related to the degree of generalization that is enforced and to an estimate of the noise (*8*).
10. L. L. Schumaker, *Spline Functions: Basic Theory* (Wiley, New York, 1981).
11. Equation 1 can be grounded on Bayesian estimation [see, for instance, G. Kimeldorf and G. Wahba, *Ann. Math. Stat.* **41**, 495 (1970)] and connected to estimation and coding principles such as the minimum length principle of J. Rissanen [*Automatica* **14**, 465 (1978)]. See also (*5*). The first term of Eq. 1 is associated with the conditional probability that corresponds to a model of Gaussian additive noise, whereas the second term is associated with the prior probability of the solution. Minimizing the functional corresponds to the maximum a posteriori (MAP) estimate, that is, maximizing the posterior probability of f given the data $f(\xi_i)$ (*8*). In addition, the solutions provided by standard regularization are known to be equivalent to generalized splines. This allows the use of a large body of results on fitting and approximating with splines. Furthermore, we have shown that standard regularization can be implemented by using analog networks of resistors and batteries (*8*). Thus, spline-based learning can be implemented in terms of the same analog, iterative networks used for 2-D surface reconstruction, but with higher connectivity.
12. D. S. Broomhead and D. Lowe, *Complex Syst.* **2**, 321 (1988).
13. J. Moody and C. Darken, *Neural Comput.* **1**, 281 (1989).
14. Depending on the stabilizer, a term belonging to the null space of P, usually a polynomial, may have to be added to the right side of Eq. 3. Disregarding this term for simplicity, the coefficients are given by $\mathbf{c} = (G + \lambda I)^{-1}\mathbf{d}$, where $(\mathbf{d})_i = d_i$ and $(G)_{ij} = G(\xi_i;\xi_j)$ (I is the identity operator). In the limit $\lambda = 0$, corresponding to noiseless data, we obtain a method of interpolating a multivariate function.
15. Thin-plate splines in two dimensions correspond to the functional

$$\|Pf\|^2 = \int_{R^2} dxdy \left\{ \left[\frac{\partial^2 f(x,y)}{\partial x^2} \right]^2 + 2\left[\frac{\partial^2 f(x,y)}{\partial x \partial y} \right]^2 + \left[\frac{\partial^2 f(x,y)}{\partial y^2} \right]^2 \right\} \quad (7)$$

that is, the bending energy of a thin plate of infinite extent. The d-dimensional Gaussian G of variance σ is generated by

$$\|Pf\|^2 = \sum_{m=0}^{\infty} \frac{\sigma^{2m}}{m!2^m} \int_{R^d} d\mathbf{x} [D^m f(\mathbf{x})]^2 \quad (8)$$

where $D^{2m} = \nabla^{2m}$, $D^{2m+1} = \vec{\nabla}\nabla^{2m}$, ∇^2 is the Laplacian operator, and $\vec{\nabla}$ is the gradient operator. Tensor product splines correspond to stabilizing operators that are the product of "one-dimensional" operators and are not radial. In two dimensions, for example, they correspond to stabilizers of the form $P = P_x P_y$, where $P_x(P_y)$ is a differential operator involving only derivatives with respect to $x(y)$. The Green's functions associated with $P_x P_y$ is the product of the Green's functions associated with P_x and P_y. The 2-D problem is then regarded as the "tensor product" of two 1-D problems.
16. M. J. D. Powell, in *Algorithms for Approximation*, J. C. Mason and M. G. Cox, Eds. (Clarendon, Oxford, 1987), pp. 143–167; R. Franke, *Math. Comput.* **38**, 181 (1982).
17. C. A. Micchelli, *Constr. Approx.* **2**, 11 (1986).
18. For nonzero λ our method can be considered as an extension of the original RBF method to approximate noisy data: regularization justifies the use of RBF expansions as an approximation method.
19. The extension amounts to searching a solution in a lower dimensional space. A standard technique is to expand the solution $f(\mathbf{x})$ on a finite basis, that is,

$$f(\mathbf{x}) = \sum_{\alpha=1}^{n} c_\alpha \phi_\alpha(\mathbf{x}) \quad (9)$$

where $\{\phi_\alpha\}_{\alpha=1}^n$ is a set of linearly independent functions [see, for instance, G. Wahba, in *Approximation Theory III*, E. W. Cheney, Ed. (Academic Press, New York, 1980), p. 905]. The coefficients c_α are then found according to a rule that guarantees a minimum deviation from the true solution. In our case we set $n < N$ and $\phi_\alpha = G(\|\mathbf{x} - \mathbf{t}_\alpha\|^2)$, where the set of "centers" $\{\mathbf{t}_\alpha\}_{\alpha=1}^n$ is to be determined. This is the only choice that guarantees that in the case of $n = N$ and $\{\mathbf{t}_\alpha\}_{\alpha=1}^n = \{\xi_i\}_{i=1}^N$ the correct solution (of Eq. 1) is consistently recovered. The chosen expansion has the additional desirable property of being a universal approximator (*25*).
20. In the HyperBF scheme the basis functions may be nonradial, at different resolutions, and of different types

$$f(\mathbf{x}) = \sum_{m=1}^{p} \sum_{\alpha=1}^{n} c_\alpha^m G^m(\mathbf{x};\mathbf{t}_\alpha^m) \quad (10)$$

where the set of parameters c_α^m and $\mathbf{t}_\alpha^m$ are unknown. This corresponds to the prior assumptions of f being the superposition of several components f^m, each

with its own stabilizer P^m. In an example of a priori information, the function to be approximated has components on a number p of scales $\sigma_1, \ldots \sigma_p$. Then

$$\|P^m f^m\|^2 = \sum_{k=0}^{\infty} a_k^m \int_{R^n} dx [D^k f^k(\mathbf{x})]^2 \qquad (11)$$

where $D^{2k} = \nabla^{2k}$, $D^{2k+1} = \nabla\nabla^{2k}$, and $a_k^m = \sigma_m^{2k}/k!2^k$. As a result, the solution will be a superposition of superpositions of Gaussians of different variance. In the radial case the norm is in general a weighted norm that scales differently the different types of input dimensions. Basis functions associated with different stabilizers may have differently weighted norms.

21. We consider the radial case for simplicity. For fixed ξ_α, the c_α can be found as $c = (G^TG + \lambda g)^{-1}G^Td$, where $G_{i\alpha} = G(\|\xi_i - t_\alpha\|_W^2)$, $g_{\alpha\beta} = G(\|t_\alpha - t_\beta\|_W^2)$, and G^T is the transpose of G. We consider the case of movable centers t_α and a weighted norm with matrix W. If the least-square error is minimized, the updating rules for the coefficients c_α, the norm matrix W, and the centers t_α are (in the case of $\lambda \to 0$):

$$c_\alpha^{(k+1)} = c_\alpha^{(k)} + 2\omega \sum_{i=1}^{N} \Delta_i G(\|\xi_i - t_\alpha\|_W^2) + \mu_\alpha^k, \quad \alpha = 1, \ldots n \qquad (12)$$

$$W^{(k+1)} = W^{(k)} - 4W^{(k)}\omega \sum_{\alpha=1}^{n}\sum_{i=1}^{N} \Delta_i G' \times (\|\xi_i - t_\alpha\|_W^2)\, Q_{i\alpha}{}^T + \gamma^n \qquad (13)$$

$$t_\alpha^{(k+1)} = t_\alpha^{(k)} - 4\omega c_\alpha \sum_{i=1}^{N} \Delta_i G'(\|\xi_i - t_\alpha\|_W^2) \times W^TW(\xi_i - t_\alpha) + \epsilon_\alpha^k, \; \alpha = 1, \ldots n \qquad (14)$$

where ω is a parameter related to the rate of convergence to the fixed point, d is the dimensionality of the input space; μ_α, ϵ_α, and γ_j are Gaussian noise terms; $(\xi_i - t_\alpha)_j$ is the jth component of the vector $(\xi_i - t_\alpha)$,

$$\Delta_i = y_i - \sum_{\alpha=1}^{K} c_\alpha G(\|\xi_i - t_\alpha\|_W^2) \qquad (15)$$

is the error between the desired output and the network's output for example i, and $Q_{i\alpha} = (\xi_i - t_\alpha)(\xi_i - t_\alpha)^T$. Notice that

$$\sum_{i=1}^{N} Q_{i\alpha}$$

are correlation matrices of the input vectors. Other similar, more efficient, iterative methods for minimizing a cost functional should be used in practice.

22. T. Poggio and S. Edelman, *Nature*, in press.
23. M. Casdagli, *Physica D* **35**, 335 (1989); S. Renals and R. Rohwer, in *Proceedings of the International Joint Conference on Neural Networks* (Washington, DC, June 1989) (IEEE TAB Neural Network Committee, Institute of Electrical and Electronic Engineers, New York, 1990), vol. 1, pp. 461–467; D. H. Wolpert, in *Abstract of the First Annual International Neural Network Society Meeting* (Pergamon, New York, 1988), p. 474; D. H. Wolpert, *Biocybernetics* **61**, 303 (1989).
24. G. Cybenko, *Math. Control Syst. Signals*, in press.
25. F. Girosi and T. Poggio, *Artif. Intell. Memo 1164* (1989).
26. P. Kanerva, *Sparse Distributed Memory* (MIT Press, Cambridge, MA, 1988).
27. D. J. Hand, *Kernel Discriminant Analysis* (Research Studies Press–Wiley, New York, 1982); R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis* (Wiley, New York, 1973).
28. D. Marr, *J. Physiol. (London)* **202**, 436 (1969).
29. J. S. Albus, *Math. Biosci.* **10**, 25 (1971).
30. J. D. Keeler, *Cognitive Sci.* **12**, 299 (1988).
31. T. Kohonen, *Biol. Cybern.* **43**, 59 (1982).
32. J. MacQueen, in *Proceedings: 5th Berkeley Symposium on Mathematics, Statistics, and Probability*, L. M. LeCam and J. Neyman, Eds. (Univ. of California Press, Berkeley, 1967), p. 281.
33. This observation suggests faster update schemes, in which a suboptimal position of the centers is first found and then the c are determined, similar to the algorithm developed and tested successfully by Moody and Darken (*13*). After this stage the coupled gradient descent equations are then used more effectively.
34. R. Durbin and D. E. Rumelhart, in *Neural Comput.* **1**, 133 (1989).
35. Multidimensional radial Gaussian units can be synthesized as the product of lower dimensional Gaussians, and 1-D and 2-D Gaussians can be implemented directly in terms of direct weighted connections from the input space (as real dendritic trees can implement a Gaussian receptive field) (*5*).
36. D. I. Perrett *et al.*, *Trends Neurosci.* **10**, 358 (1987).
37. Constant, linear, and even higher order polynomials may be required in a regularization network, depending on the stabilizer P (of which they should span the null space). Gaussian RBFs do not need additional terms. It is always possible, however, to add polynomial terms even in the case of the Gaussian. On the other hand, the theorem in appendix C of (*25*) shows that good approximations can always be obtained by the superposition of the Green's functions associated with regularization, even without the polynomial terms.

38. We are grateful to S. Edelman, E. Grimson, E. Hildreth, D. Hillis, B. Moore, L. Tucker, S. Ullman, and especially A. Hurlbert for useful discussions and suggestions. Support for this research was provided by a grant from the Office of Naval Research, Cognitive and Neural Sciences Division, by the Artificial Intelligence Center of Hughes Aircraft Corporation, and by the North Atlantic Treaty Organization Scientific Affairs Division (0403/87). Support for the Artificial Intelligence Laboratory's research is provided by the Advanced Research Projects Agency of the Department of Defense under Army contract DACA76-85-C-0010 and in part under Office of Naval Research contract N00014-85-K-0124. T.P. is supported by the Uncas and Ellen Whitaker Chair.

27 July 1989; accepted 29 November 1989

Representation Properties of Networks: Kolmogorov's Theorem Is Irrelevant

Federico Girosi
Tomaso Poggio
Massachusetts Institute of Technology, Artificial Intelligence Laboratory, Cambridge, MA 02142 USA
and
Center for Biological Information Processing, Whitaker College, Cambridge, MA 02142 USA

Many neural networks can be regarded as attempting to approximate a multivariate function in terms of one-input one-output units. This note considers the problem of an exact representation of nonlinear mappings in terms of simpler functions of fewer variables. We review Kolmogorov's theorem on the representation of functions of several variables in terms of functions of one variable and show that it is irrelevant in the context of networks for learning.

1 Kolmogorov's Theorem: An Exact Representation Is Hopeless

A crucial point in approximation theory is the choice of the representation of the approximant function. Since each representation can be mapped in an appropriate network choosing the representation is equivalent to choosing a particular network architecture. In recent years it has been suggested that a result of Kolmogorov (1957) could be used to justify the use of multilayer networks composed of simple one-input–one-output units. This theorem and a previous result of Arnol'd (1957) can be considered as the definitive disproof of *Hilbert's conjecture* (his thirteenth problem, Hilbert 1900): *there are continuous functions of three variables, not representable as superpositions of continuous functions of two variables.*

The original statement of Kolmogorov's theorem is the following (Lorentz 1976):

Theorem 1.1. (Kolmogorov 1957). *There exist fixed increasing continuous functions $h_{pq}(x)$, on $I = [0, 1]$ so that each continuous function f on I^n can be written in the form*

$$f(x_1, \ldots, x_n) = \sum_{q=1}^{2n+1} g_q\left(\sum_{p=1}^{n} h_{pq}(x_p)\right),$$

where g_q are properly chosen continuous functions of one variable.

Neural Computation **1, 465–469** (1989)

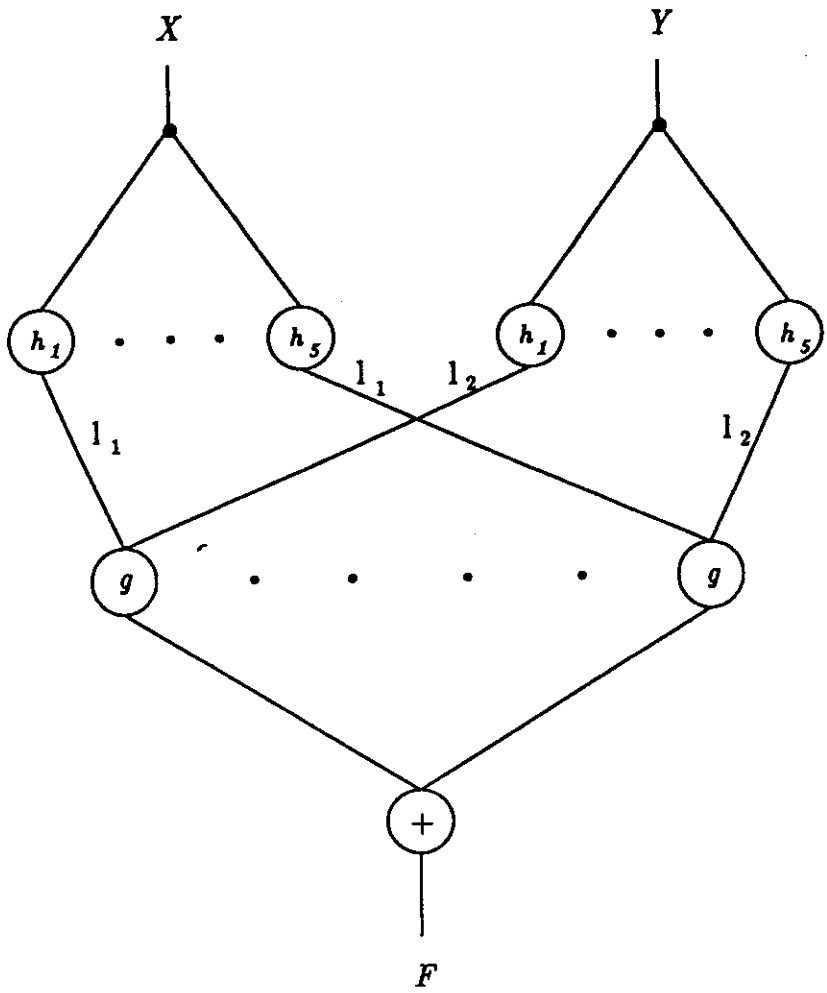

Figure 1: The network representation of an improved version of Kolmogorov's theorem, due to Kahane (1975). The figure shows the case of a bivariate function. The Kahane's representation formula is $f(x_1, \ldots, x_n) = \sum_{q=1}^{2n+1} g[\sum_{p=1}^{n} l_p h_q(x_p)]$ where h_q are strictly monotonic functions and l_p are strictly positive constants smaller than 1.

This result asserts that every multivariate continuous function can be represented by the superposition of a small number of univariate continuous functions. In terms of networks this means that every continuous function of many variables can be computed by a network with two hidden layers (see Figure 1) whose hidden units compute continuous functions (the functions g_q and h_{pq}).

Does Kolmogorov's theorem, in its present form, prove that a network with two hidden layers is a good and usable representation? The answer is definitely no. There are at least two reasons for this:

1. In a network implementation that has to be used for learning and generalization, some degree of smoothness is required for the func-

tions corresponding to the units in the network. Smoothness of the h_{pq} and of the g_q is important because the representation must be smooth in order to generalize and be stable against noise. A number of results of Vituskin (1954, 1977) and Henkin (1964) show, however, that the inner functions h_{pq} of the Kolmogorov's theorem are highly not smooth (they can be regarded as "hashing" functions). Due to this "wild" behavior of the inner functions h_{pq}, the functions g_q do not need to be smooth, even for differentiable functions f (de Boor 1987).

2. Useful representations for approximation and learning are *parametrized* representations that correspond to networks with fixed units and modifiable parameters. Kolmogorov's network is not of this type: the form of g_q (corresponding to units in the second "hidden" layer) depends on the specific function f to be represented (the h_{pq} are independent of it). g_q is at least as complex, for instance in terms of bits needed to represent it, as f.

A stable and usable *exact* representation of a function in terms of two or more layers network seems hopeless. In fact the result obtained by Kolmogorov can be considered as a "pathology" of the continuous functions: it fails to be true if the inner functions h_{pq} are required to be smooth, as it has been shown by Vitushkin (1954). The theorem, though mathematically surprising and beautiful, cannot be used by itself in any constructive way in the context of networks for learning. This conclusion seems to echo what Lorentz (1962) wrote, more than 20 years ago, asking "Will it [Kolmogorov's theorem] have useful applications?... One wonders whether Kolmogorov's theorem can be used to obtain positive results of greater [than trivial] depth." Notice that this leaves open the possibility of finding good and well founded approximate representations. This argument is discussed in some length in Poggio and Girosi (1989), and a number of results have been recently obtained by some authors (Hornik *et al.* 1989; Stinchcombe and White 1989; Carroll and Dickinson 1989; Cybenko 1989; Funahashi 1989; Hecht-Nielsen 1989).

The next section reviews Vitushkin's main results.

2 The Theorems of Vitushkin

The interpretation of Kolmogorov's theorem in term of networks is very appealing: the representation of a function requires a fixed number of nodes, polynomially increasing with the dimension of the input space. Unfortunately, these results are somewhat pathological and their practical implications very limited. The problem lies in the inner functions of Kolmogorov's formula: although they are continuous, theorems of Vitushkin and Henkin (Vitushkin 1964, 1977; Henkin 1964; Vitushkin and Henkin 1967) prove that they must be highly nonsmooth. One could ask if it is

possible to find a superposition scheme in which the functions involved are smooth. The answer is negative, even for two variable functions, and was given by Vitushkin with the following theorem (1954):

Theorem 2.1. (Vitushkin 1954). *There are $r (r = 1, 2, \ldots)$ times continuously differentiable functions of $n \geq 2$ variables, not representable by superposition of r times continuously differentiable functions of less than n variables; there are r times continuously differentiable functions of two variables that are not representable by sums and continuously differentiable functions of one variable.*

We notice that the intuition underlying Hilbert's conjecture and theorem 2.1 is the same: not all the functions with a given degree of complexity can be represented in simple way by means of functions with a lower degree of complexity. The reason for the failing of Hilbert's conjecture is a "wrong" definition of complexity: Kolmogorov's theorem shows that the number of variables is not sufficient to characterize the complexity of a function. Vitushkin showed that such a characterization is possible and gave an explicit formula. Let f be an r times continuously differentiable function defined on I^n with all its partial derivatives of order r belonging to the class $Lip[0,1]^\alpha$. Vitushkin puts $\chi = (r + \alpha)/n$ and shows that it can be used to measure the inverse of the complexity of a class of functions. In fact he succeded in proving the following:

Theorem 2.2. (Vitushkin 1954). *Not all functions of a given characteristic $\chi_0 = q_0/k_0 > 0$ can be represented by superpositions of functions of characteristic $\chi = q/k > \chi_0, q \geq 1$.*

Theorem 2.1 is easily derived from this result.

Acknowledgments

We acknowledge support from the Defense Advanced Research Projects Agency under contract number N00014-89-J-3139. Tomaso Poggio is supported by the Uncas & Helen Whitaker Chair at MIT.

References

Arnol'd, V. I. 1957. On functions of three variables. *Dokl. Akad. Nauk SSSR* **114**, 679–681.

Carroll, S. M., and Dickinson, B. W. 1989. Construction of neural nets using the Radon transform. In *Proceedings of the International Joint Conference on Neural Networks*, pp. I-607–I-611, Washington, D.C., June 1989. IEEE TAB Neural Network Committee.

Cybenko, G. 1989. Approximation by superposition of a sigmoidal function. *Math. Control Systems Signals*, in press.

de Boor, C. 1987. Multivariate approximation. In *The State of the Art in Numerical Analysis*, A. Iserles and M. J. D. Powell, eds., pp. 87–109. Clarendon Press, Oxford.

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

Hecht-Nielsen, R. 1989. Theory of backpropagation neural network. In *Proceedings of the International Joint Conference on Neural Networks*, pp. I-593–I-605, Washington D.C., June 1989. IEEE TAB Neural Network Committee.

Henkin, G. M. 1964. Linear superpositions of continuously differentiable functions. *Dokl. Akad. Nauk SSSR* **157**, 288–290.

Hilbert, D. 1900. Mathematische probleme. *Nachr. Akad. Wiss. Göttingen*, 290–329.

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

Kahane, J. P. 1975. Sur le theoreme de superposition de Kolmogorov. *J. Approx. Theory* **13**, 229–234.

Kolmogorov, A. N. 1957. On the representation of continuous functions of several variables by superposition of continuous functions of one variable and addition. *Dokl. Akad. Nauk SSSR* **114**, 953–956.

Lorentz, G. G. 1976. On the 13-th problem of Hilbert. In *Proceedings of Symposia in Pure Mathematics*, pp. 419–429, Providence, RI, 1976. American Mathematical Society.

Lorentz, G. G. 1962. Metric entropy, widths, and superposition of functions. *Am. Math. Monthly* **69**, 469–485.

Poggio, T., and Girosi, F. 1989. A theory of networks for approximation and learning. A.I. Memo No. 1140, Artificial Intelligence Laboratory, Massachusetts Institute of Technology.

Stinchcombe, M., and White, H. 1989. Universal approximation using feedforward networks with non-sigmoid hidden layer activation functions. In *Proceedings of the International Joint Conference on Neural Networks*, pp. I-607–I-611, Washington, D.C., June 1989. IEEE TAB Neural Network Committee.

Vitushkin, A. G. 1954. On Hilbert's thirteenth problem. *Dokl. Akad. Nauk SSSR* **95**, 701–704.

Vitushkin, A. G. 1964. Some properties of linear superposition of smooth functions. *Dokl. Akad. Nauk SSSR* **156**: 1003–1006.

Vitushkin, A. G. 1977. *On Representation of Functions by Means of Superpositions and Related Topics*. L'Enseignement Mathematique.

Vitushkin, A. G., and Henkin, G. M. 1967. Linear superposition of functions. *Russian Math. Surveys* **22**, 77–125.

Received 17 July 1989; accepted 30 August 1989.

Reprinted from Nature, Vol. 343, No. 6255, pp. 263-266, 18th January, 1990

A network that learns to recognize three-dimensional objects

T. Poggio & S. Edelman

Artificial Intelligence Laboratory, Center for Biological Information Processing, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, USA

THE visual recognition of three-dimensional (3-D) objects on the basis of their shape poses at least two difficult problems. First, there is the problem of variable illumination, which can be addressed by working with relatively stable features such as intensity edges rather than the raw intensity images[1,2]. Second, there is the problem of the initially unknown pose of the object relative to the viewer. In one approach to this problem, a hypothesis is first made about the viewpoint, then the appearance of a model object from such a viewpoint is computed and compared with the actual image[3-7]. Such recognition schemes generally employ 3-D models of objects, but the automatic learning of 3-D models is itself a difficult problem[8,9]. To address this problem in computational vision, we have developed a scheme, based on the theory of approximation of multivariate functions, that learns from a small set of perspective views a function mapping any viewpoint to a standard view. A network equivalent to this scheme will thus 'recognize' the object on which it was trained from any viewpoint.

Is the need for 3-D range-based or manually specified models real? Structure from motion theorems[10,11], pioneered by Ullman[12], indicate that full information about the 3-D structure of an object represented as a set of feature points (at least five to eight) is present in just two of their perspective views, provided that corresponding points are identified in each view. A view is represented as a $2N$ vector $x_1, y_1, x_2, y_2, \ldots, x_N, y_N$ of the coordinates on the image plane of N labelled and visible feature points on the object. Here, and in most of the following, we assume that all features are visible, as they are in wire-frame objects. The generalization to opaque objects follows by partitioning the viewpoint space for each object into a set of 'aspects'[13], corresponding to stable clusters of visible features. In principle, therefore, having enough 2-D views of an object is equivalent to having its 3-D structure specified.

This line of reasoning, together with properties of perspective projection, indicate (1) that for each object there exists a smooth function mapping any perspective view into a 'standard' view of the object, and (2) that this multivariate function can be synthesized, or at least approximated, from a small number of views of the object. Such a function would be object-specific, with different functions corresponding to different 3-D objects. Furthermore, the application of the function that is specific for one object to the views of a different object is expected to result in a 'wrong' standard view that can be easily detected as such.

Synthesizing an approximation to a function from a small number of sparse data—the views—can be considered as learning an input–output mapping from a set of examples[14,15]. A powerful scheme for the approximation of smooth functions has been recently proposed under the name of Generalized Radial Basis Functions (GRBFs), and shown[14,15] to be equivalent to standard regularization[16,17] and generalized splines (ref. 14; see closely related work by Powell[18], and Broomhead and Lowe[19]). The approximation of $f: R^n \rightarrow R$ is given by

$$f(\mathbf{x}) = \sum_{\alpha=1}^{K} c_\alpha G(\|\mathbf{x} - \mathbf{t}_\alpha\|) \qquad (1)$$

where the K coefficients c_α and the centres $\mathbf{t}_\alpha$ are found during the learning stage and G is an appropriate basis function (see refs 14 and 15), such as the gaussian function. A polynomial term of the form $\sum_i d_i p_i(\mathbf{x})$ can be added to the right-hand side of equation (1). In this paper we omit the polynomial term (see ref. 14). If the function f is vector-valued, each component f_i is computed using equation (1) with the appropriate $c_{i\alpha}$, in which case the equation is equivalent to the network of Fig. 1.

The weights c are found during learning by minimizing a measure of the error between the network's prediction and the desired output for each of the M examples. Computationally, this amounts to inverting a matrix (when $M \neq K$, the generalized inverse is computed instead). When the number of basis functions is less than the number of views in the training set, the centres of the basis functions are also updated during learning. Updating the centres is equivalent to modifying the corresponding 'prototypical views'. For a detailed description of this approximation technique, of its theoretical motivation and its relation to other techniques such as backpropagation[20], see refs 14 and 15.

Figure 2 shows an application of GRBFs to the recognition problem. We consider here the special case of recognizing a wire-frame 3-D object from any of its perspective views with N feature points (we mainly used $N = 6$). A GRBF module, trained on several tens of random views, maps any new view of the same object into a standard view (for example, into one of the initially chosen training views).

We have also explored the use of fewer basis functions than training views and used gradient descent to look for the optimal locations of the centres $\mathbf{t}_\alpha$ in addition to the optimal value of c_α. We found satisfactory performance with just two basis units (for 10–40 training views and with the attitude of the object limited to one octant of the viewing sphere). This indicates that a very small number of units are needed for each aspect[13] of an opaque object (compare with ref. 21). It is of interest that

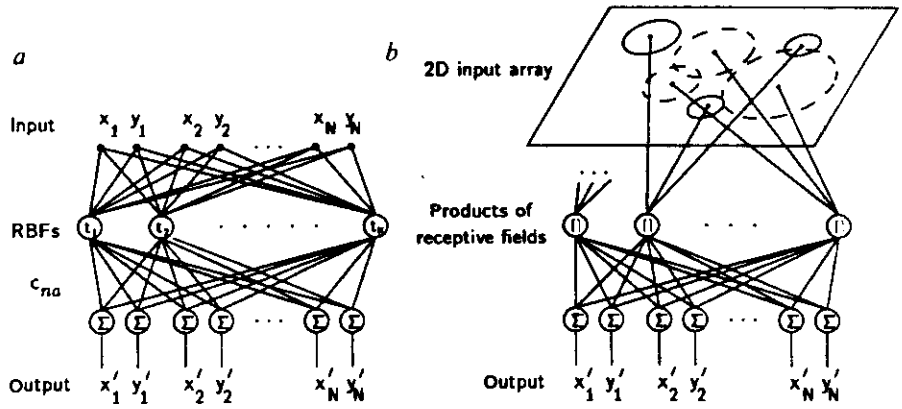

FIG. 1 *a*, Network representation of approximation by GRBFs. In a special simple case, there are as many basis functions (K) as views in the training set (M; in general, $K \leq M$). The centres of the radial functions are then fixed and are identical with the training views. Each basis unit in the 'hidden' layer computes the distance of the new view from its centre and applies to it the radial function. The resulting value $G(\|\mathbf{x} - \mathbf{t}_\alpha\|)$ can be regarded as the 'activity' of the unit. If the function G is gaussian, a basis unit will attain maximum activity when the input exactly matches its centre. The output of the network is the linear superposition of the activities of all the basis units in the network. *b*, An equivalent interpretation of *a* for the case of gaussian radial basis functions. A multidimensional gaussian function can be synthesized as the product of 2-D gaussian receptive fields operating on retinotopic maps of features. The solid circles in the image plane represent the 2-D gaussian functions associated with the first radial basis function, which corresponds to the first view of the object. The dotted circles represent the 2-D receptive fields that synthesize the gaussian radial function associated with another view. The gaussian receptive fields transduce positions of features represented implicitly as activity in a retinotopic array, and their product 'computes' the radial function without the need of calculating norms and exponentials explicitly.

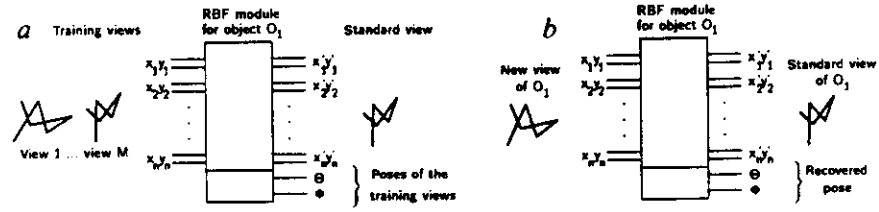

FIG. 2 Application of a general module for multivariate function approximation to the problem of recognizing a 3-D object from any of its perspective views. *a*, Module is trained to produce the vector representing the standard view of the object, given a set of examples of random perspective views of the same object. The module is also capable of recovering the viewpoint coordinates θ, ϕ (the latitude and the longitude of the camera on an imaginary sphere centred at the object) that correspond to the training views. When given a new random view of the same object (*b*), the module recognizes it by producing the standard view. Other objects are rejected by thresholding the euclidean distance between the actual output of the model and the standard view (this step corresponds to the action of a single radial function with a sharp cut-off centred on the standard view).

after training, the centres of the radial basis units correspond to views that are different from any of the training views.

It should be clear that the scheme proposed here addresses only one part of the problem of shape-based object recognition, the variability of object appearance due to changing viewpoint. The key issue of how to detect and identify image features that are stable for different illuminations and viewpoints is outside the scope of this paper. Notice that the GRBF approach to recognition does not require the x, y coordinates of image features as inputs: other parameters of appropriate features could also be used, such as a corner angles (see Fig. 4*a*) or segment lengths (compare ref. 4 and M. Villalba, thesis in preparation), or the colour and the texture of the object. Recognition of noisy and partially occluded objects, using realistic feature identification schemes, requires an extension of the scheme, even if the problems of object segmentation and selection[22] are addressed separately. A natural extension of the scheme could be based, for example, on the use of multiple lower-dimensional centres, corresponding to different subsets of detected features, instead of one $2N$-dimensional centre for each view in the example set. Our initial experiments[23] support the notion that a scheme based on low-dimensional centres is useful for recognition while being robust against occlusions and noise. Another possible extension of the scheme involves a hierarchical composition of GRBF modules, in which the outputs of lower-level modules assigned to detect objects parts and their relative disposition in space are combined to allow recognition of complex-structured objects.

In a sense, the application of the GRBF method to recognition can be considered as a generalization of the exact approach of Basri and Ullman[24]. They have recently shown that under orthographic projection, any view of a 3-D object undergoing a linear group of transformations that includes rigid transformation in 3-D space (that is, translations and rotations) can be obtained from three fixed views. They used this result to synthesize a linear operator that, for orthographic projection, maps exactly each view of a given object into the zero vector and performs fairly well also for most cases of perspective projection[24]. By comparison, the GRBF approach is based on an approximation, even in the orthographic case, and typically needs more than three views. But it can (1) use as inputs feature parameters other than the x, y coordinates (Fig. 4*a*) and (2) recover parameters, such as the attitude angles of the input object (Fig. 4*d*), that do not depend linearly on the views of the object.

In some respects, the performance of the GRBF-based recognition scheme resembles human performance in a related task. For example, the number of training views necessary to achieve an acceptable recognition rate on novel views, 80–100 for the full viewing sphere, is broadly compatible with the finding[25] that people have trouble recognizing a novel wire-frame object previously seen from one viewpoint if it is rotated away from that viewpoint by about 30° (it takes 72 $30° \times 30°$-patches to cover the viewing sphere). Furthermore, a network model recently shown to capture some of the time-course and learning characteristics of the recognition process[26], seems to be computationally related to GRBFs[27]. Experiments designed to test specific predictions of GRBF and several other recognition schemes[24,27] are now under way in our laboratory.

One feature of the GRBF scheme that could guide its interpretation in biological terms is the possibility of decomposing a multidimensional gaussian radial basis function into a product of gaussian functions of lower dimensions (Fig. 1*b*). In our case, the centre of a basis unit is similar to a prototype and the unit itself is synthesized as the product of feature detectors with 2-D gaussian receptive fields (that is, the activity of a detector depends on the distance r between the stimulus and the centre of the receptive field as e^{-r^2/σ^2}). The network's output (see equation 1) is the sum of products and therefore represents the logical disjunction of conjunctions '$\vee_\alpha \wedge_i$ (feature F_i at (x_i, y_i))', where the disjunction ranges over all the protypes of the given object.

The adjustment of weights c_α in the GRBF network in Fig. 1 through some pseudo-hebbian mechanism is not biologically implausible. Alternatively, a plausible biophysical implementation of the gradient-descent update of the centres (or, as in Fig. 1*b*, the location of the receptive fields) is problematic. But notice that reasonable initial performance can be obtained merely by setting the centres to a subset of the examples. A subsequent possibly slow process, much simpler and more plausible than gradient descent, may then search for optimal positions. Another

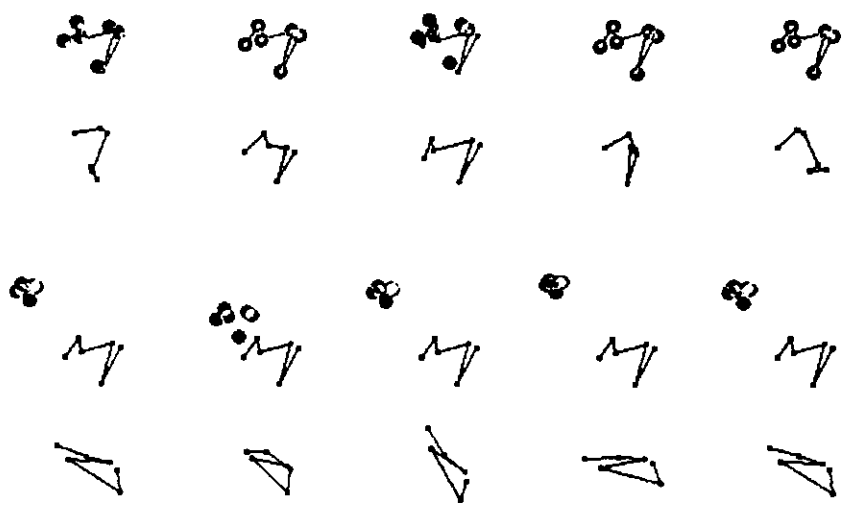

FIG. 3 Some examples of the module's operation. Standard view of a wire-frame object (top row) superimposed on its estimate by the GRBF network (large dots) when its input is a random view of the same object (second row from top). The fit is much closer than in the bottom two rows, where the input view belongs to a different object. The number of training views M is 40, the number of RBFs K is 20, and the range of attitudes θ, ϕ is 0°–90°. Gradient descent was used to obtain the optimal positions of the GRBF centres. Within a smaller range of $\theta, \phi \in [0°, 45°]$, the performance was acceptable with only two radial basis units ($M = 40$, $K = 2$).

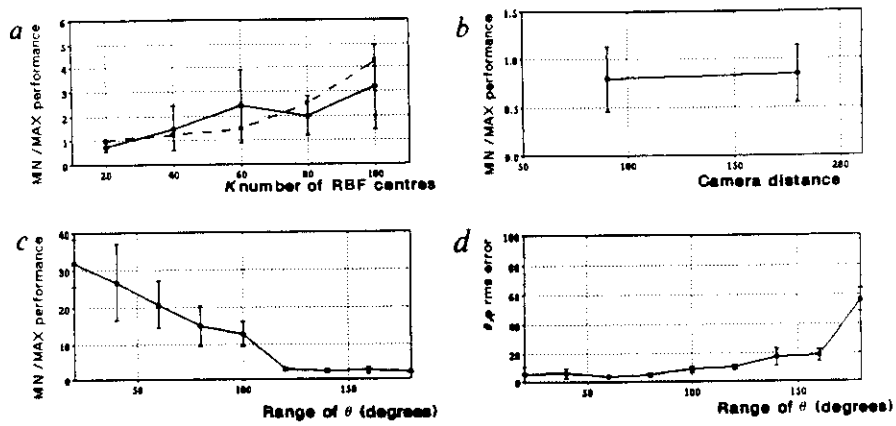

FIG. 4 *a*, Performance of a GRBF module trained to recognize a specific object over the full range of θ, ϕ (the entire viewing sphere). Views were encoded as vectors of $2N$ vertex coordinates (solid curve: error bars show the s.d. of the performance indices, computed over a set of 10 objects, each of which served in turn as the target) or as vectors of $N-2$ angles formed by pairs of segments (dashed curve). In these examples, the number of training views M is chosen to equal the number of radial basis functions K. The performance index MIN/MAX is defined as the ratio of the smallest euclidean distance E obtained for views of different objects to the largest E obtained over a set of novel random views of the object on which the module has been trained. MIN/MAX >1 is required for a perfect separation between the target and other objects using a simple threshold decision. For nearly perfect recognition, 80–100 views suffice. *b*, Performance for two conditions—near and far—corresponding to relatively high and low perspective distortion, respectively (full range of θ, ϕ in both cases). *c*, GRBF shows a slow degradation in performance with increasing range of the viewpoint coordinates θ, ϕ (the objects are a cube and an octahedron, $M=K=40$, and the error bars are s.d. over 10 sets of random training and testing views). *d*, GRBF can also provide a good estimate of the attitude of the object. The inset shows the errors in the viewpoint coordinates θ, ϕ recovered by the module versus the range of the viewpoint coordinates. In *c* and *d*, $\phi_{max}=2\theta_{max}$, so that $\theta_{max}=180°$ corresponds to the full viewing sphere.

possible solution is to select for each object a set of optimally located receptive fields out of a large available population[27,14]. Sensory-input driven selection of representation units has been demonstrated *in vivo* (for example, see refs 28 and 29).

The GRBF recognition scheme seems reasonable in terms of the biophysical mechanisms required, is attractive because an effective computation is simply performed by the combination of receptive fields, and is surprising because it bases a scheme involving units somewhat similar to 'grandmother' cells (compare refs 30 and 31) on the rigorous approximation methods of regularization and splines. □

Received 9 August; accepted 20 November 1989

1. Marr, D. *Vision* (Freeman, San Francisco, 1982).
2. Poggio, T., Gamble, E. B. & Little, J. J. *Science* **242**, 436–440 (1988).
3. Fischler, M. A. & Bolles, R. C. *Commun. ACM* **24**, 381–395 (1981).
4. Thompson, D. W. & Munday, J. L. in *Proc. IEEE Conf. Robotics and Automation* 208–220 (Raleigh, North Carolina, 1987).
5. Huttenlocher, D. P. & Ullman, S. in *Proc. 1st Int. COnf. Computer Vision* 102–111 (IEEE, Washington DC, 1987).
6. Lowe, D. G. *Perceptual Organization and Visual Recognition* (Kluwer Academic Publishers, Boston, Massuchusetts, 1986).
7. Ullman, S. *Cognition* **32**, 193–254 (1989).
8. Grimson, W. E. L. & Lozano-Perez, T. *IEEE Trans. Pattern Analysis Machine Intell.* **9**, 469–482 (1987).
9. Fan, T. J., Medioni, G. & Nevatia, R. in *Proc. 2nd Int. Conf. Computer Vision* 474–481 (Florida, IEEE, Washington DC, 1988).
10. Tsai, R. Y. & Huang, T. S. *IEEE Trans. Pattern Analysis Machine Intell.* **6**, 13–27 (1984).
11. Longuet-Higgins, H. C. *Nature* **293**, 133–135 (1981).
12. Ullman, S. *The Interpretation of Visual Motion* (MIT Press, Cambridge, Massachusetts, 1979).
13. Koenderink, J. J. & van Doorn, A. *J. Biol. Cybern.* **32**, 211–217 (1979).
14. Poggio, T. & Girosi, F. *Artif. Intell. Lab. Memo No. 1,140* (Artificial Intelligence Laboratory, MIT, Cambridge, 1989).
15. Poggio, T. & Girosi, F. *Science* (in the press).
16. Tikhonov, A. N. & Arsenin, V. Y. *Solutions of Ill-posed Problems* (Winston, Washington DC, 1977).
17. Poggio, T., Torre, V. & Koch, C. *Nature* **317**, 314–319 (1985).
18. Powell, M. J. D. in *Algorithms for Approximation* (eds Mason, J. C. & Cox, M. G.) (Clarendon, Oxford, 1987).
19. Broomhead, D. S. & Lowe, D. *Complex Syst.* **2**, 321–355 (1988).
20. Rumelhart, D. E., Hinton, G. E. & Williams, R. J. *Nature* **323**, 533–536 (1986).
21. Perrett, D. I., Mistlin, A. J. & Chitty, A. J. *Trends Neurosci.* **10**, 358–364 (1989).
22. Edelman, S. & Poggio, T. *Optic News* **15**, 8–15, May 1989.
23. Poggio, T. & Edelman, S. *Artif. Intell. Lab. Memo No. 1,181* (Artificial Intelligence Laboratory, MIT, Cambridge, 1989).
24. Basri, R. & Ullman, S. *Artif. Intell. Lab. Memo No. 1,152* (Artificial Intelligence Laboratory, MIT, Cambridge, 1989).
25. Rock, I. & DiVita, J. *Cognitive Psychol.* **19**, 280–293 (1987).
26. Edelman, S., Bülthoff, H. & Weinshall, D. *Artif. Intell. Lab. Memo No. 1,138* (Artificial Intelligence Laboratory, MIT, Cambridge, 1989).
27. Edelman, S. & Weinshall, D. *Artif. Intell. Lab. Memo No. 1,146* (Artificial Intelligence Laboratory, MIT, Cambridge, 1989).
28. Jenkins, W. M., Merzenich, M. M. & Ochs, M. T. *Soc. Neurosci. Abstr.* **10**, 665 (1984).
29. Edelman, G. M. & Finkel, L. in *Dynamical Aspects of Neocortical Function* (eds Edelman, G. M., Gall, W. E. & Cowan, W. M.) 653–695 (Wiley, New York, 1984).
30. Gross, C. G., Rocha-Miranda, C. E. & Bender, D. B. *J. Neurophys.* **35**, 96–111 (1972).
31. Perrett, D. I., Rolls, E. T. & Caan, W. *Expl Brain Res.* **47**, 329–342 (1982).

ACKNOWLEDGEMENTS. We thank F. Crick, F. Girosi, E. Grimson, E. Hildreth, D. Hillis, A. Hurlbert, L. Tucker, S. Ullman and D. Weinshall for discussion. The work was done in the Artificial Intelligence Laboratory and the Center for Biological Information Processing in the Department of Brain and Cognitive Sciences. This research was supported by the ONR, Cognitive and Neural Sciences Division and the Artificial Intelligence Center of Hughes Aircraft Corporation. Support for the A.I. Laboratory's artificial intelligence research is provided by the Advanced Research Projects Agency of the Department of Defense. T.P. is supported by the Uncas and Helen Whitaker chair. S.E. is supported by a Chaim Weizmann Postdoctoral Fellowship from the Weizmann Institute of Science.

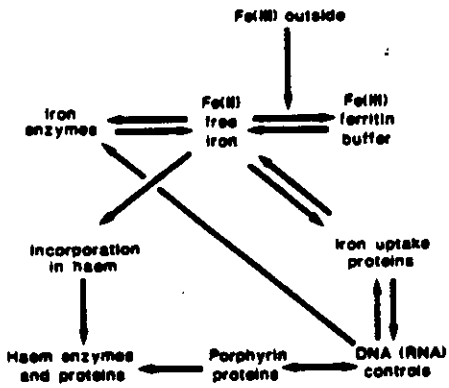

FIG. 2 The complicated interaction between iron and the controls and catalysts of a bacterial cell. Many iron enzymes are Fe_n / S_n proteins. The question posed by the findings described in this article is whether in early anaerobic life ferritin, the present-day iron store, was replaced by a mineral iron sulphide.

crystallization, and we would ourselves dearly like to have this control over crystallization.

The discovery of the extensive deposits of iron in a variety of bacteria leads to questions about the usefulness of such deposits, because some of the iron sulphides are not magnetic and presumably the earliest iron oxides deposited were not magnetites but were closely related to ferrihydrite. The need to store iron is related of course to the essential role of the element in a wide range of catalytic enzymes (which, in a sense, parallels the storage of calcium in bones). Iron homoeostatic levels also control a whole range of enzymes essential in bioenergetics (see Fig. 2). Buffering the iron levels in aerobic cells is the mineral ferritin; but curiously hidden under this name is a variety of crystalline and amorphous iron hydroxy oxides, wrapped in somewhat similar proteins, and combined with very different amounts of phosphate depending on the particular organism concerned[2]. These ferritins are likely to be precursors of the Fe_2O_3 crystals, now known to be common in soils[3].

The occurrence of several forms of iron sulphide in sulphur bacteria[2] leads one to speculate as to whether sulphur bacteria have a homoeostatic device based on Fe/S rather than on Fe/O solids, and whether this was the earlier form of iron store. In these organisms there is a very basic link between sulphide and iron metabolism and homoeostasis. Both are also necessary to maintain the core grouping of the essential primitive electron-transfer proteins, that is Fe_nS_n clusters where $n = 2$, (3), or 4. The earliest energy-capture devices leading to ATP formation were based on these proteins long before the advent of dioxygen chemistry[4]. An Fe/S homoeostatic economy for primitive life could have many other ramifications. After all pyrites (FeS_2) is half-way from sulphide to elemental sulphur, which was one of the early metabolic end-products. The general idea of early life based on iron/sulphur chemistry is not new and has been explored and reviewed by Hartman[6], who suggested that we should look further to see if very primitive redox systems could have used organic as well as inorganic sulphur chemistry in an alternative programme for chemical-bond energy capture before the use of ATP.

So could the iron sulphides have been a direct source of energy for early life? Electron-rich iron sulphides are not stable in water but are present in ocean vents and in some geological formations. The overall reaction of interest would be

$$H_2O + 2(FeS) + CO_2 \rightarrow 2FeO + >CHOH + 2S$$

which could be driven in part by light[6]. Did primitive life systems capture colloidal particles of iron as catalysts, in line with views of biological catalysis earlier in this century? The surfaces of iron sulphide minerals could also have been catalysts for such reactions as those now seen in Fe/S cluster proteins (for example, hydrogenases and dehydratases such as aconitase), which could conceivably be the reason for the appearance of different iron sulphides in the organisms described on pages 256 and 258. A magnetic sensor could not have been of much use in the anaerobic early world.

As always in biology, when a new set of compounds is uncovered — be it a pigment such as bacteriorhodopsin, a new coenzyme such as PQQ, or minerals such as Fe_3O_4, FeS_2 and Fe_3S_4 — we must search for the functional advantage of the material in the biological niche in which it is synthesized. The discovery of magnetic iron oxides in soil bacteria[1] is in itself peculiar, because it is hard to see the value of a navigational aid there, while the finding of iron sulphides deposited in cells gives us a new line to follow back towards the origin of life. Energy capture based on Fe/S compounds, now and perhaps before there was life, is as important as DNA in life's history. There could well be a huge variety of life in the sulphide-rich zones on Earth, perhaps holding fresh clues to help answer the question of how life began. □

R.J.P. Williams is in the Inorganic Chemistry Laboratory, University of Oxford, South Parks Road, Oxford OX1 3PN, UK.

1. Fassbinder, J.W.E., Stanjek, H. & Vali, H. *Nature* **343** 161–163 (1990).
2. Farina, M., Esquivel, D.M.S. & de Barros, H.G.P.L. *Nature* **343**, 256–258 (1990).
3. Mann, S., Sparks, N.H.C., Frankel, R.B., Bazylinski, D.A. & Jannasch, H.W. *Nature* **343**, 258–261 (1990).
4. Mann, S., Webb, J. & Williams, R.J.P. (eds) *Biomineralisation* (VCH, Weinheim, 1989).
5. San Pietro, A.G. (ed.) *Non-heme Iron Proteins* (Antioch Yellow Springs, Ohio, 1965).
6. Hartman, H. *J. molec. Evol.* **4**, 359–370 (1975).

ARTIFICIAL INTELLIGENCE

Recognizing three dimensions

H. Christopher Longuet-Higgins

ONE of the human abilities that robot designers would most like to emulate is that of recognizing objects from their appearances. It seems fair to say that we simply do not know how the shapes of three-dimensional objects are represented in the long-term memory, how these representations are established in the first place or how they are deployed in the task of visual identification. Our ignorance is due not so much to a shortage of neurophysiological evidence as to a dearth of ideas worth testing. How *could* a network such as the human visual system learn to recognize particular objects after seeing them only a limited number of times, from different viewpoints and under different conditions of illumination? On page 263 of this issue[1] Poggio and Edelman propose an answer to precisely this question.

As Poggio and Edelman point out, a 3-D object gives rise to a limitless variety of 2-D images, both because of its infinite number of possible positions relative to the viewer, and because of the multiplicity of possible lighting conditions. The problem of lighting may be overcome by working with relatively invariant image features such as discontinuities in intensity, and Poggio and Edelman's main concern is with the problem of variable viewpoint. One approach, widely used by robot-vision engineers, is to store in the robot's memory an explicitly 3-D representation of the object, indifferent to viewpoint. When an image appears on the robot's 'retina', its features and those of the candidate object lead to a hypothesis about the viewpoint; the system then computes the appearance of the object from the viewpoint and compares the result with the image. An unsatisfactory comparison prompts a change of hypothesis — about the viewpoint, the identity of the object or both. Not only is this process horribly expensive in computing time, it leaves unsolved the problem of how the robot is to get to know about the object in the first place. In most existing schemes this information is either supplied in advance by the user or obtained with range finders or other active sensing equipment.

Poggio and Edelman start from the hypothesis that the visual recognition of objects does *not* require that their 3-D shapes have to be explicitly represented in memory, only that the system has been exposed to enough views of a given object

to be able to recognize another such view when it sees it. What they actually do is to train their network to emit a unique 'standard' view of the object in response to the input of any one of a representative set of views. In this context the word 'view' is something of a euphemism: it signifies not an optical image but a vector whose components are the image coordinates of a few identifiable features of the object. The hypothetical 3-D objects with which Poggio and Edelman mainly deal are 'bent wire' models with two ends and four corners, so that each 2-D 'view' is a vector with 12 components altogether. Only if this vector can be approximated as a linear combination of the representative views will the network, after training, emit the standard view in response to it — a welcome demonstration of its visual competence.

Poggio and Edelman admit that the extraction of such view vectors from a set of retinal images would be a non-trivial task, involving (among other things) the solution of a multiple-correspondence problem between different images. It seems likely, moreover, that formidable book-keeping problems will arise when the appearances of a large number of objects have to be committed to visual memory. For a number of reasons, however, their work should be of interest to both theoretical and experimental psychologists.

Mathematicians working on neural nets will be reassured to know that the recently proposed approximation scheme that underlies Poggio and Edelman's system — the so-called generalized radial basis function (GRBF) scheme — works so well in such a key application. Theoreticians who are not on the circulation list for unpublished MIT research reports will be happy to see how relatively straightforward it is to implement the GRBF scheme in neural networks of a now conventional kind. They will, furthermore, be interested to learn of the recent proof by Basri and Ullman[2] that any orthogonal projection of a transparent polyhedron (a 3-D object all of whose vertices are visible from any direction) can be represented as a linear combination of three orthogonal projections along arbitrary directions in space — a fact that underlies a simple algorithm for determining whether a given image could or could not be an orthogonal projection of that object. Poggio and Edelman recognize their own scheme as a natural extension of Basri and Ullman's work.

On the experimental side, Poggio and Edelman make some pertinent observations about the numbers that emerge from their computational experiments. They find, for example, that the number of training views needed to achieve an acceptable recognition rate for novel views is 80 to 100 for the full viewing sphere. This figure harmonizes nicely with Rock and DiVita's finding[3] that people have trouble in recognizing a wire-frame object if it is rotated more than about 30° away from any position in which they saw it before, coupled with the fact that it takes 72 30° × 30° patches to cover the viewing sphere. Finally they point out that their GRBF recognition scheme requires the combination of 'receptive fields' by network units somewhat similar to the 'grandmother cells' beloved by some neurophysiologists. So perhaps we do, after all, have brain cells that fire when and only when Grannie makes her appearance. □

H. Christopher Longuet-Higgins is in the Centre for Research on Perception and Cognition, University of Sussex, Brighton BN1 9QG, UK.

1. Poggio, T. & Edelman, S. *Nature* **343**, 263 – 266 (1990).
2. Basri, R. & Ullman, S. *Artif. Intell. Lab. Memo No. 1152* (Artificial Intelligence Laboratory, MIT, Cambridge 1989).
3. Rock, I. & DiVita, J. *Cognitive Psychol.* **19**, 280 – 293 (1987).

NOVA EXPLOSIONS

The long hot summer

Sumner Starrfield and R. Mark Wagner

THE explosion of a recurrent, classical or X-ray nova is one of the most violent events that can occur in a galaxy. The discovery of such an event will stimulate astronomers worldwide to observe it with many techniques and at various wavelengths. This explains the excitement, evident at a recent meeting*, generated last summer when an X-ray nova, a recurrent nova, two classical novae and the bizarre Cygnus X-3 were all found to be in outburst.

The series of outbursts began on 22 May 1989 when the All Sky Monitor on board the Japanese Ginga satellite discovered an X-ray source in the constellation Cygnus[1]. Upon learning of the discovery through the International Astronomical Union (IAU) telegram network (IAU Circ. No. 4782), we immediately set out to identify the optical counterpart of the source. This task was complicated because the actual position lay outside the initial X-ray error box. However, B. Marsden, editor of the IAU circulars, pointed out to us the close correspondence in the X-ray position to that of a nova, V404 Cygni, last known to be in outburst in 1938, listed in the *Atlas of Galactic Novae* by Duerbeck[2]. An electronic-mail message to A. Cassatella at the International Ultraviolet Explorer (IUE) satellite observatory in Spain alerted him to re-direct the satellite to observe V404 Cyg (the mid-afternoon Sun in Arizona left us temporarily helpless). Indeed, V404 Cyg was brighter than catalogued (see figure), but interstellar extinction rendered the IUE spectra blank. Within a few hours, our optical spectra confirmed that the X-ray source was identical with V404 Cyg (IAU Circ. 4783).

Our initial observations were followed rapidly by multiwavelength observations by several astronomers at infrared, radio and γ-ray wavelengths (IAU Circs 4786, 4790, 4794, 4797, 4800, 4816, 4879). For example, the radio studies of V404 Cyg, as reported by R. M. Hjellming and X. Xan (IAU Circs 4790, 4796, 4879) and at the meeting showed that the radio behaviour of this outburst was completely unlike that of the other X-ray novae. They found very short-timescale variations and quasi-periodic oscillations. The data gathered over the next few months showed that this outburst was completely different from the other well-studied long-period X-ray novae — A0620-00 (V616 Mon) and Cen X-4.

Negative images of V404 Cygni before (13 July 1951, from the Palomar Sky Survey; left) and after (2 June 1989, by T. J. Kreidl and S. B. Howell; right) the nova explosion of 22 May 1989.

The outburst of V404 Cyg was soon followed by outbursts of Cygnus X-3 in June and July (IAU Circs 4798, 4817, 4826). Then in August, a recurrent nova, V745 Scorpii (IAU Circs 4820, 4821, 4822, 4825, 4826, 4844, 4853, 4885), and a classical nova, Scorpii 1989 (4836, 4838, 4839, 4840), were discovered to be in outburst. Finally, in September another classical nova, Scuti 1989, was found in outburst (IAU Circs 4861, 4862, 4865).

The discovery of V404 Cyg and V745 Sco demonstrate the need to study those outbursts that have been recorded in the past as having low amplitudes at optical

*11th North American Workshop on Cataclysmic Variables, Santa Fe, 9–13 October 1989.