perikarya or fibers within the graft or evidence that NPY fibers were entering the graft from the host. In contrast to implants that contained the SCN, cortical implants never restored rhythmicity to the host. Cortical implants always contained a few NPY perikarya and many fibers. In cortical implants, NPY fibers were always seen to cross the host-graft border, and most crossing fibers seemed to originate from the host.

Although most of our implants contained some portion of extra-SCN tissues (Fig. 3, A and D), the immunocytochemical analysis showed that grafts that restored rhythmicity always contained cells with SCN characteristics (VIP and vasopressin). Therefore, the period of the overt rhythm is determined by cells within, or very close to the SCN. This observation is in agreement with reports showing that the SCN is required for successful restoration of rhythmicity (*11–14, 23*).

In most of our locomotor data, rhythmicity was visually apparent within 6 to 7 days after transplantation. Although surprisingly short, this latency does not preclude the possibility that neural reconnections drive the behavior since dense neural outgrowth has been reported from other transplanted tissue with a similar time course (*24*). Immunocytochemical analysis indicates that neural connections have been made between graft and host brain; however, it was not possible to determine the source of fibers crossing the graft boundary.

The fact that the genotype of the host does not appear to affect significantly the expression of the transplanted rhythm is somewhat surprising, especially in view of evidence for the existence of oscillators outside the SCN in the mammalian brain (*15, 16*). We interpret the absence of a host contribution to the circadian period to mean that either the SCN is essentially autonomous in determining the primary characteristics of rhythmicity in hamsters or that the host brain fails to make the connections with the tissue graft that are required for the brain to influence this period. In either case, our results strengthen the view that the SCN occupies a position at the top of the circadian hierarchy in mammals.

REFERENCES AND NOTES

1. R. Y. Moore, in *Biological Rhythms and Their Central Mechanisms*, M. Suda, O. Hayaishi, H. Hakagawa, Eds. (North-Holland, Amsterdam, 1979), pp. 343–354.
2. B. Rusak and Z. Boulos, *Photochem. Photobiol.* **34**, 267 (1981).
3. W. J. Schwartz and H. Gainer, *Science* **197**, 1089 (1977).
4. R. Y. Moore and V. B. Eichler, *Brain Res.* **42**, 201 (1972).
5. F. K. Stephan and I. Zucker, *Proc. Natl. Acad. Sci. U.S.A.* **69**, 1583 (1972).
6. B. Rusak and I. Zucker, *Physiol. Rev.* **59**, 449 (1979).
7. S. T. Inouye and H. Kawamura, *Proc. Natl. Acad. Sci. U.S.A.* **76**, 5962 (1979).
8. D. J. Green and M. U. Gillette, *Brain Res.* **245**, 198 (1982).
9. G. Groos and J. Hendricks, *Neurosci. Lett.* **34**, 283 (1982).
10. D. J. Earnest and C. D. Sladek, *Brain Res.* **382**, 129 (1986).
11. Y. Sawaki *et al.*, *Neurosci. Res.* **1**, 67 (1984).
12. R. Drucker-Colin *et al.*, *Brain Res.* **311**, 353 (1984).
13. M. N. Lehman *et al.*, *J. Neurosci.* **7**, 1626 (1987).
14. P. J. DeCoursey and J. Buggy, *Soc. Neurosci. Abstr.* **12**, 210 (1986).
15. K.-I. Honma, S. Honma, T. Hiroshige, *Physiol. Behav.* **40**, 767 (1987).
16. K. Abe, J. Kroning, M. A. Greer, V. Critchlow, *Neuroendocrinology* **29**, 119 (1979); F. K. Stephan, J. M. Swann, C. L. Sisk, *Behav. Neural. Biol.* **25**, 346 (1979).
17. In the avian pineal [S. A. Binkley, J. B. Riebman, K. B. Reilly, *Science* **202**, 1198 (1978); N. H. Zimmerman and M. Menaker, *Proc. Natl. Acad. Sci. U.S.A.* **76**, 999 (1979)], in the lizard pineal [M. Menaker and S. Wisner, *ibid.* **80**, 6119 (1983)], and in the amphibian retina [J. C. Besharse and P. M. Iuvone, *Nature* **305**, 133 (1983)].
18. M. R. Ralph and M. Menaker, *Science* **241**, 1225 (1988).
19. Tissue for implantation was obtained in the following manner. Pregnant females were anesthetized and prepared for surgery on day 13.5 of gestation. Fetuses were located and decapitated in utero, and the heads were removed to a sterile petri dish containing BSS. The entire litter was collected at one time. Fetal brains were then removed under dissecting microscope and placed in a second dish. After all of the brains had been collected, each was oriented with the ventral surface visible under the microscope. All dissections were performed with the tissue immersed in MEM-BSS. This procedure allowed the ends of the developing optic nerves to float in the media so that the optic chiasm could be located easily. A coronal incision was made where the two nerves fused, and a second, parallel cut was made 1 to 1.5 mm caudal to this. Two parasagittal cuts were then made about 1.5 mm on either side of midline, with the scissors held at a 45° angle so that these cuts passed into the third ventricle. This resulted in the excision of two small blocks of tissue connected by the optic chiasm. Neural tissue for implantation was then teased away from the chiasm and pia mater.
20. Tissue blocks for implantation were placed in a group at the tip of a Wiretrol micropipette. The pipette was graduated in increments of 1.0 μl so that the total volume to be injected could be estimated. The tissue occupied about 1 μl of the 1.5 to 2 μl volume that was injected into the host brain.
21. E. Balaban, M.-A. Teillet, N. Le Douarin, *Science* **241**, 1339 (1988).
22. Animals were perfused intracardially with 350 ml of 4% paraformaldehyde in 0.01*M* phosphate buffer containing 15% picric acid; the brain was removed and placed in the same fixative for an additional 24 to 48 hours. Frontal sections (80 μm) were cut on a vibratome and processed for immunocytochemistry as described [R. G. Foster, G. Plowman, A. Goldsmith, B. Follett, *J. Endocrinol.* **115**, 211 (1987)] with antibodies directed against NPY (1:1000; Peninsula Laboratories, Belmont, CA), vasopressin (1:500; Incstar, Stillwater, MN), and VIP (1:1000; Peninsula Laboratories). Controls were performed by incubating sections in absorbed primary antisera [40 nm of peptide (Peninsula Laboratories) added to 1 ml of diluted primary antisera for 24 hours at 4°C] or omitting the primary antisera. In both controls, immunostaining was abolished in the graft and host brain, except for faint staining within necrotic tissue and astrocytes around the lesion site, suggesting artifactual staining within these tissues. Necrotic tissue and astrocytes around the lesion site also showed immunostaining with VIP, NPY, and vasopressin antibodies (Fig. 3).
23. R. Aguilar-Roblero, L. P. Morin, R. Y. Moore, *Soc. Neurosci. Abstr.* **14**, 49 (1988).
24. M. K. Floeter and E. G. Jones, *Dev. Brain Res.* **22**, 19 (1985).

25. Supported by PHS grants MH09483 to M.R.R., HD13162 to M.M., and HD18686 to F.C.D.

28 June 1989; accepted 28 November 1989

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

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

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

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

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

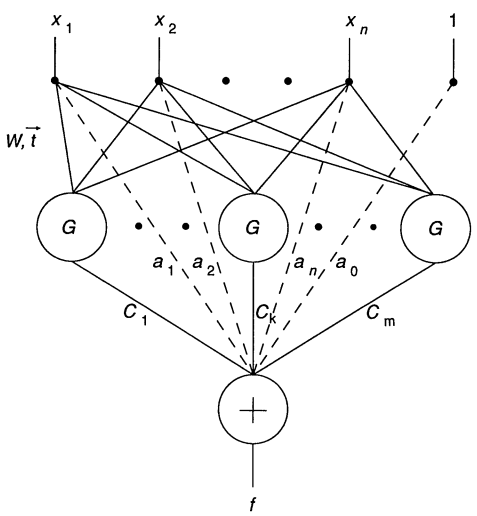

Fig. 1. The HyperBf network used to approximate a mapping between x_1, x_2, . . ., x_n and f, given a set of sparse, noisy data. The data, a set of points for which the value of the function is known, can be considered as examples to be used during learning. The hidden units evaluate the function $G(\mathbf{x};\mathbf{t}_n)$, 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 $\mathbf{t}_\alpha$ 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, . . . 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:

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

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:

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

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 $\|\mathbf{x} - \boldsymbol{\xi}_i\|$ 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

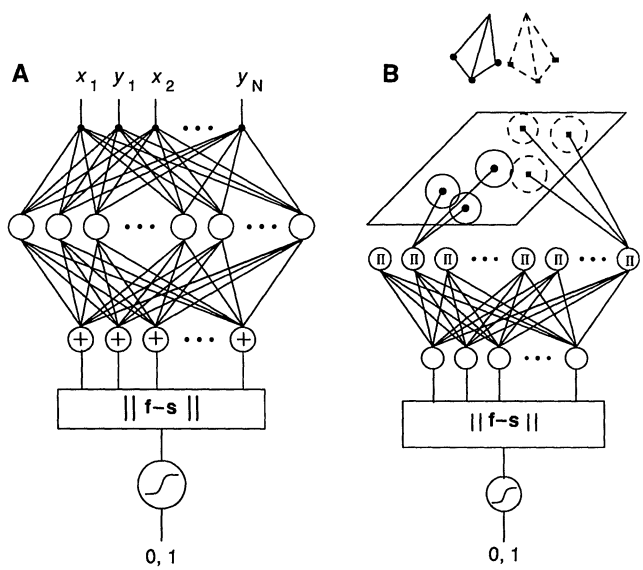

Fig. 2. (**A**) The HyperBF network proposed for the recognition of a 3-D object from any of its perspective views. The network attempts to map any view (as defined in the text) into a standard view, arbitrarily chosen. The norm of the difference between the output vector **f** and the standard view **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 $f(\boldsymbol{\xi}_i)$ (*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 $(G)_{ij} = G(\boldsymbol{\xi}_i;\boldsymbol{\xi}_j)$ (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

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

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 $\{\mathbf{t}_\alpha\}_{\alpha=1}^n = \{\boldsymbol{\xi}_i\}_{i=1}^N$ 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

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

where the set of parameters c_i^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 $\boldsymbol{\xi}_\alpha$, the c_α can be found as $\mathbf{c} = (G^TG + \lambda g)^{-1}G^T\mathbf{d}$, where $G_{i\alpha} = G(\|\boldsymbol{\xi}_i - \mathbf{t}_\alpha\|^2)$, $g_{\alpha\beta} = G(\|\mathbf{t}_\alpha - \mathbf{t}_\beta\|^2)$, and G^T is the transpose of G. We consider the case of movable centers $\mathbf{t}_\alpha$ 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 $\mathbf{t}_\alpha$ are (in the case of $\lambda \to 0$):

$$c_\alpha^{(k+1)} = c_\alpha^{(k)} + 2\omega \sum_{i=1}^{N} \Delta_i G(\|\boldsymbol{\xi}_i - \mathbf{t}_\alpha\|_W^2) + \mu_\alpha^k, \quad \alpha = 1, \ldots n \qquad (12)$$

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

$$\mathbf{t}_\alpha^{(k+1)} = \mathbf{t}_\alpha^{(k)} - 4\omega c_\alpha \sum_{i=1}^{N} \Delta_i G'(\|\boldsymbol{\xi}_i - \mathbf{t}_\alpha\|_W^2) \times W^TW(\boldsymbol{\xi}_i - \mathbf{t}_\alpha) + \epsilon_\alpha^k, \quad \alpha = 1, \ldots n \qquad (14)$$

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; $(\boldsymbol{\xi}_i - \mathbf{t}_\alpha)_j$ is the jth component of the vector $(\boldsymbol{\xi}_i - \mathbf{t}_\alpha)$,

$$\Delta_i = y_i - \sum_{\alpha=1}^{K} c_\alpha G(\|\boldsymbol{\xi}_i - \mathbf{t}_\alpha\|_W^2) \qquad (15)$$

is the error between the desired output and the network's output for example i, and $Q_{i\alpha} = (\boldsymbol{\xi}_i - \mathbf{t}_\alpha)(\boldsymbol{\xi}_i - \mathbf{t}_\alpha)^T$. 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