A Theory of How the Brain Might Work

T. POGGIO

I.R.S.T., Povo, 38100 Trento, Italy; Thinking Machines Co., Artificial Intelligence Laboratory, Cambridge, Massachusetts 02142; Center for Biological Information Processing, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139

I wish to propose a quite speculative new version of the grandmother cell theory to explain how the brain, or parts of it, may work. In particular, I discuss how the visual system may learn to recognize three-dimensional objects. The model would apply directly to the cortical cells involved in visual face recognition. I also outline the relationship of our theory to existing models of the cerebellum and of motor control. Specific biophysical mechanisms can be readily suggested as part of a basic type of neural circuitry that can learn to approximate multidimensional input/output mappings from sets of examples and that is expected to be replicated in different regions of the brain and across modalities. The main points of the theory are:

1. The brain uses modules for multivariate function approximation as basic components of several of its information processing subsystems.
2. These modules are realized as HyperBF networks (Poggio and Girosi 1990a,b).
3. HyperBF networks can be implemented in terms of biologically plausible mechanisms and circuitry.

The theory predicts a specific type of population coding that represents an extension of schemes such as look-up tables. I conclude with some speculations about the trade-off between memory and computation and the evolution of intelligence.

I. THE GRANDMOTHER NEURON THEORY

A classic theme in the neurophysiological literature, at least since the work of Hubel and Wiesel (1962), is the idea of information processing in the brain as leading to "grandmother" neurons responding selectively to the precise combination of visual features that are associated with one's grandmother. Even when not explicitly stated, this notion seems to capture how many neuroscientists believe that the brain works. The grandmother neuron theory is of course not restricted to vision and applies as well to other sensory modalities and even to motor control under the form of cells corresponding to elemental movements. Why is this idea so attractive? The idea is attractive because of its simplicity: It replaces possibly complex information processing with the superficially simpler task of accessing a memory. The problem of recognition and motor control would be solved by simply accessing look-up tables containing appropriate descriptions of objects and of motor actions. The human brain can probably exploit a vast amount of memory with its 10^{14} or so synapses, making attractive any scheme that replaces computation with memory. In the case of vision, the apparent simplicity of this solution hides the difficult problems of an appropriate representation of an object and of how to extract it from complex images. Even assuming that these problems of representation, feature extraction, and segmentation could be solved by other mechanisms, a fundamental difficulty seems to be intrinsic to the grandmother cell idea. The difficulty consists of the combinatorial explosion in the number of cells that any scheme of the look-up table type would reasonably require for either vision or motor control. In the case of three-dimensional object recognition, for instance, there should be for each object as many entries in the look-up table as there are two-dimensional views of the object, in principle an infinite number!

The difficulty of a combinatorial explosion lies at the heart of theories of intelligence that attempt to replace information processing with look-up tables of precomputed results. In this paper, we suggest a scheme that avoids the combinatorial problem, while retaining the attractive features of the look-up table. The basic idea is to use only a few entries and interpolate or approximate among them. A mathematical theory based on this idea leads to a powerful scheme of learning from examples that is equivalent to a parallel network of simple processing elements. The scheme has an intriguingly simple implementation in terms of plausible biophysical mechanisms. We discuss in particular the case of three-dimensional object recognition but propose that the scheme is possibly used by the brain for several different information-processing tasks. Many information-processing problems can be represented as the composition of one or more multivariate functions that map an input signal into an output signal in a smooth way. These modules could be synthesized from a sufficient set of input/output pairs—the examples—by the scheme described here. Because of the power and general applicability of this mechanism, we speculate that a part of the machinery of the brain, including perhaps some of the cortical circuitry that is somewhat similar across the different modalities, may be dedicated to the task of function approximation.

Cold Spring Harbor Symposia on Quantitative Biology, Volume LV. 0-87969-059-3/90 $1.00

II. HOW TO SYNTHESIZE THROUGH LEARNING THE BASIC APPROXIMATION MODULE: REGULARIZATION NETWORKS

This section describes a technique for synthesizing the approximation modules discussed above through learning from examples. I first explain how to rephrase the problem of learning from examples as a problem of approximating a multivariate function. The material in this section is from Poggio and Girosi (1989, 1990a,b), where more details can be found.

To illustrate the connection, let us draw an analogy between learning an input/output mapping and a standard approximation problem, two-dimensional surface reconstruction from sparse data points. *Learning* simply means collecting the *examples*, i.e., 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, i.e., no data. This requires interpolating or, more generally, approximating the surface (i.e., 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* (Omohundro 1987; Poggio et al. 1988, 1989).

From this point of view, learning a smooth mapping from examples is clearly ill-posed, in the sense that the information in the data is not sufficient to reconstruct uniquely the mapping at places 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. 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* and have interesting Bayesian interpretations (Tikhinov and Arsenin 1977; Poggio et al. 1985; Bertero et al. 1988). We have recently shown 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). Our main result (Poggio and Girosi 1989) is that the regularization approach is equivalent to an expansion of the solution in terms of a certain class of functions:

$$f(\mathbf{x}) = \sum_{i=1}^{N} c_i G(\mathbf{x}; \xi_i) + p(\mathbf{x}) \qquad (1)$$

where $G(\mathbf{x})$ is one such function and the coefficients c_i satisfy a linear system of equations that depend on the N "examples," i.e., the data to be approximated. The term $p(\mathbf{x})$ is a polynomial that depends on the smoothness assumptions. In many cases, it is convenient to include up to the constant and linear terms. Under relatively broad assumptions, the Green's function G is radial and therefore the approximating function becomes

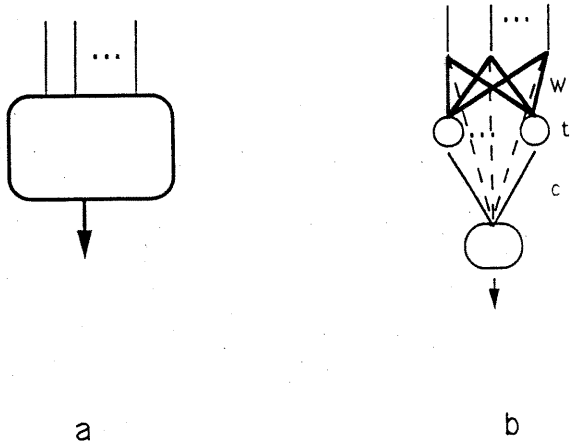

Figure 1. (*a*) The basic learning module that (we conjecture) is used by the brain for a number of tasks. The module learns to approximate a multivariate function from a set of examples (i.e., a set of input/output pairs). (*b*) A HyperBF network equivalent to a module for approximating a scalar function of three variables from sparse and noisy data. The data, a set of points where 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 are determined during learning are the coefficients c_n, the centers $\mathbf{t}_n$ and the norm-weights $\mathbf{W}$. In the radial case $G = G(\|\mathbf{x} - \mathbf{t}_n\|_{\mathbf{W}}^2)$ and the hidden units simply compute the radial basis functions G at the "centers" $\mathbf{t}_n$. The radial basis functions 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). There may be also connections computing the polynomial term of Fig. 1b: Constant and linear terms (the dotted lines in *b*) may be expected in most cases.

$$f(\mathbf{x}) = \sum_{i=1}^{N} c_i G(\|\mathbf{x} - \xi_i\|^2) + p(\mathbf{x}) \qquad (2)$$

which is a sum of radial functions, each with its *center* ξ_i on a distinct data point and of constant and linear terms (from the polynomial, when restricted to be of degree one). 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 specific a priori assumption of smoothness. Depending on it, 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 and not. As observed by Broomhead and Lowe (1988) in the radial case, a superposition of functions like Equation 1 is equivalent to a network of the type shown in Figure 1b. The interpretation of Equation 2 is simple: in the two-dimensional case, for instance, the surface is approximated by the superposition of, say, several two-dimensional Gaussian distributions, each centered on one of the data points.

The network associated with Equation 2 can be made more general in terms of the following extension

$$f^*(\mathrm{x}) = \sum_{\alpha=1}^{n} c_\alpha G(\|(\mathrm{x} - \mathbf{t}_\alpha)\|_W^2) + p(\mathrm{x}) \qquad (3)$$

where the parameters $\mathbf{t}_\alpha$, which we call "centers," and the coefficients c_α are unknown, and are in general many fewer than the data points ($n \leq N$). The norm is a *weighted norm*

$$\|(\mathrm{x} - \mathbf{t}_\alpha)\|_W^2 = (\mathrm{x} - \mathbf{t}_\alpha)^T W^T W(\mathrm{x} - \mathbf{t}_\alpha) \qquad (4)$$

where W is an unknown square 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, determining in fact the units of measure and the importance of each feature (the matrix W is especially important in cases in which the input features are of a different type and their relative importance is unknown). Equation 3 can be implemented by the network of Figure 1. Notice that a sigmoid function at the output may sometimes be useful without increasing the complexity of the system (see Poggio and Girosi 1989). Notice also that there could be more than one set of Green's functions, for instance, a set of multiquadrics and a set of Gaussians, each with its own **W**. Notice that two or more sets of Gaussians, each with a diagonal **W**, are equivalent to sets of Gaussians with their own σs.

Learning

Iterative methods can be used to find the optimal values of the various sets of parameters, the c_α, the w_i, and the $\mathbf{t}_\alpha$, that minimize an error functional on the set of examples. Steepest descent is *the* standard approach that requires calculations of derivatives. An even simpler method that does not require calculation of derivatives (suggested and found surprisingly efficient in preliminary work by B. Caprile and F. Girosi, pers. comm.) is to look for random changes (controlled in appropriate ways) in the parameter values that reduce the error. We define the error functional—also called energy—as

$$H[f^*] = H_{\mathrm{c,t,W}} = \sum_{i=1}^{N} (\Delta_i)^2$$

with

$$\Delta_i \equiv y_i - f^*(\mathrm{x}) = y_i - \sum_{\alpha=1}^{n} c_\alpha G(\|\mathrm{x}_i - \mathbf{t}_\alpha\|_{\mathrm{W}}^2)$$

In the first method, the values of c_α, $\mathbf{t}_\alpha$, and **W** that minimize $H[f^*]$ are regarded as the coordinates of the stable fixed point of the following dynamical system:

$$\dot{c}_\alpha = -\omega \frac{\partial H[f^*]}{\partial c_\alpha}, \qquad \alpha = 1, \ldots, n$$

$$\dot{\mathbf{t}}_\alpha = -\omega \frac{\partial H[f^*]}{\partial \mathbf{t}_\alpha}, \qquad \alpha = 1, \ldots, n$$

$$\dot{\mathbf{W}} = -\omega \frac{\partial H[f^*]}{\partial \mathbf{W}}$$

where ω is a parameter. The derivatives are rather complex (see Poggio and Girosi 1990a; and Notes section).

The second method is simpler: Random changes in the parameters are made and accepted if $H[f^*]$ decreases. Occasionally, changes that increase $H[f^*]$ may also be accepted (similarly to the Metropolis algorithm).

Interpretation of the Network

The interpretation of the network of Figure 1 is as follows. *After learning*, the centers of the basis 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, that 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 basis functions in the network, plus direct, weighted connections from the inputs (the linear terms of $p[\mathrm{x}]$) and from a constant input (the constant term). Notice that in the limit case of the basis functions approximating delta functions, the system becomes equivalent to a look-up table. *During learning*, the weights c are found 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 **W** for the norm is equivalent to transforming appropriately, for instance scaling, the input coordinates and corresponds to task-dependent dimensionality reduction.

Regularization networks, of which HyperBFs are the most general and powerful version, represent a general framework for learning smooth mappings that rigorously connects approximation theory, generalized splines, and regularization with feedforward multilayer networks. They also contain as special cases the radial basis functions (RBF) technique (Micchelli 1986; Powell 1987; Broomhead and Lowe 1988) and several well-known algorithms, especially in the pattern recognition literature.

III. A PROPOSAL FOR A BIOLOGICAL IMPLEMENTATION

In this section, we point out some remarkable properties of Gaussian HyperBF, which may have implications for neurobiology.

Factorizable Radial Basis Functions

The synthesis of (weighted) RBFs in high dimensions may be easier if they are factorizable. It is easily seen that the *only RBF which is factorizable is the Gaussian*

(with diagonal **W**). A multidimensional Gaussian function can be represented as the product of lower dimensional Gaussians. For instance, a two-dimensional Gaussian radial function centered in **t** can be written as

$$G(\|\mathbf{x}-\mathbf{t}\|^2_{\mathbf{W}}) \equiv e^{-\|\mathbf{x}-\mathbf{t}\|^2_{\mathbf{W}}} = e^{-(x-t_x)^2/2\sigma_x^2}\, e^{-(y-t_y)^2/2\sigma_y^2} \tag{5}$$

with $\sigma_x = 1/w_1$ and $\sigma_y = 1w_2$, where w_1 and w_2 are the elements of the matrix **W** assumed, in this section, to be diagonal.

This dimensionality factorization is especially attractive from the physiological point of view, since it is difficult to imagine how neurons could compute $G(\|\mathbf{x}-\mathbf{t}_\alpha\|^2)$. The scheme of Figure 2, on the other hand, is physiologically plausible. Gaussian radial functions in one, two, and possibly three dimensions can be implemented as *receptive fields* by weighted connections from the sensor arrays (or some retinotopic array of units representing with their activity the position of features). Gaussians in higher dimensions can then be synthesized as products of one- and two-dimensional receptive fields.

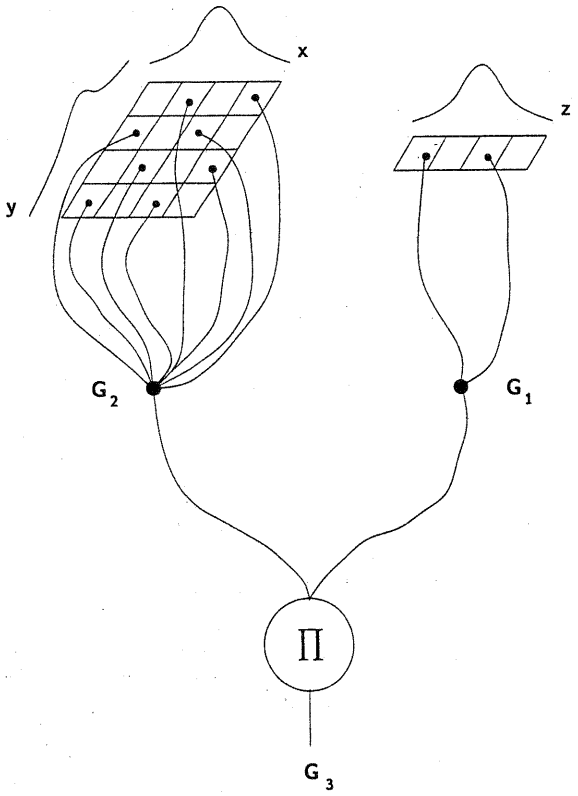

Figure 2. A three-dimensional radial Gaussian implemented by multiplying two-dimensional Gaussian and one-dimensional Gaussian receptive fields. The latter two functions are synthesized directly by appropriately weighted connections from the sensor arrays, as neural receptive fields are usually thought to arise. Notice that they transduce the implicit position of stimuli in the sensor array into a number (the activity of the unit). They thus serve the dual purpose of providing the required "number" representation from the activity of the sensor array and of computing a Gaussian function. Two-dimensional Gaussians acting on a retinotopic map can be regarded as representing two-dimensional "features," whereas the radial basis function represents the "template" resulting from the conjunction of those lower-dimensional features.

This scheme has three additional interesting features:

1. The multidimensional radial functions are synthesized directly by appropriately weighted connections from the sensor arrays, without any need of an explicit computation of the norm and the exponential.
2. Two-dimensional Gaussians operating on the sensor array or on a retinotopic array of features extracted by some preprocessing transduce the implicit position of features in the array into a number (the activity of the unit).
3. Two-dimensional Gaussians acting on a retinotopic map can be regarded each as representing one two-dimensional "feature," i.e., a component of the input vector, whereas each center represents the "template," resulting from the conjunction of those lower-dimensional features. Notice that in this analogy the RBF is the AND of several features and could also include the negation of certain features, that is the AND NOT of them. **W** weights the importance of the different features.

Biophysical Mechanisms

The network. The multiplication operation required by the previous interpretation of Gaussian GRBFs to perform the "conjunction" of Gaussian receptive fields is not too implausible from a biophysical point of view. It could be performed by several biophysical mechanisms (see Koch and Poggio 1987). Here we mention three mechanisms:

1. Inhibition of the silent type and related circuitry (see Poggio and Torre 1978; Torre and Poggio 1978)
2. The AND-like mechanism of NMDA receptors
3. A logarithmic transformation, followed by summation, followed by exponentiation. The logarithmic and exponential characteristic could be implemented in appropriate ranges by the sigmoid-like pre-to-postsynaptic voltage transduction of many synapses.

If the first or the second mechanism is used, the product of Figure 3 can be performed directly on the dendritic tree of the neuron representing the corresponding radial function (alternatively, each dendritic tree may perform pairwise products only, in which case a logarithmic number of cells would be required). The scheme also requires a certain amount of memory per basis unit, in order to store the center vector. In the case of Gaussian receptive fields used to synthesize Gaussian RBFs, the center vector is effectively stored in the position of the two-dimensional (or one-dimensional) receptive fields and in their connections to the product unit(s). This is plausible physiologically.

The linear terms (the direct connections from the inputs to the output in Fig. 1) can be realized directly as

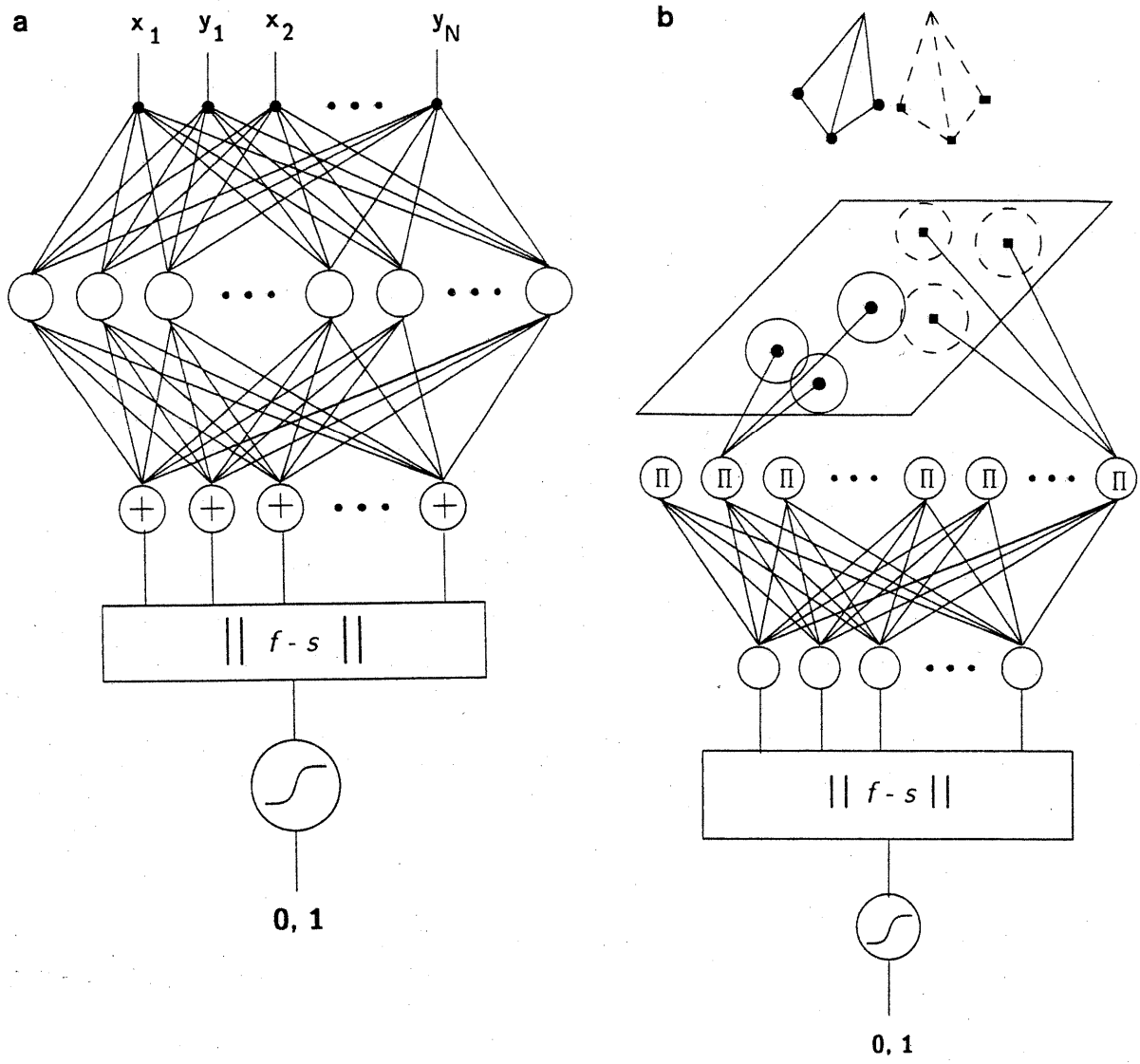

Figure 3. (*a*) The HyperBF network proposed for the recognition of a three-dimensional object from any of its perspective views (Poggio and Edelman 1990). The network attempts to map any view (as defined in the text) into a standard view, arbitrarily chosen. The norm of the difference between the output 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 radial basis functions is initially centered on one of a subset of the M views used to synthesize the system ($K < M$). During training, each of the M inputs in the training set is associated with the desired output, i.e., the standard view **s**. (*b*) A completely equivalent interpretation of *a* for the special case of Gaussian radial basis functions. Gaussian functions can be synthesized by multiplying the outputs of two-dimensional Gaussian receptive fields, that "look" at the retinotopic map of the object point features. The solid circles in the image plane represent the two-dimensional Gaussians associated with the first radial basis function, which represents the first view of the object. The dotted circles represent the two-dimensional receptive fields that synthesize the Gaussian radial function associated with another view. The two-dimensional 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. (Reprinted, with permission, from Poggio and Girosi 1990b.)

inputs to the output neuron that summates linearly its synaptic inputs (an output nonlinearity is allowed and will not change the basic form of the model, see Poggio and Girosi 1989). They may also be realized through intermediate linear units.

Mechanisms for learning. Do the update schemes have a physiologically plausible implementation? Consider first the steepest descent methods, which require derivatives. Equation 6 or a somewhat similar, quasi-Hebbian scheme is not too unlikely and may require only a small amount of neural circuitry. Equation 7 seems more difficult to implement for a network of real neurons.

Methods such as the random descent method, which do not require calculation of derivatives, are biologically much more plausible and seem to perform very well in preliminary experiments. In the Gaussian case, with

basis functions synthesized through the product of Gaussian receptive fields, moving the centers means establishing or erasing connections to the product unit. A similar argument can be made also about the learning of the matrix **W**. Notice that in the diagonal Gaussian case, the parameters to be changed are exactly the σ of the Gaussians, i.e., the spread of the associated receptive fields. Notice also that the σ for all centers on one particular dimension is the same, suggesting that the learning of w_i may involve the modification of the scale factor in the input arrays rather than a change in the dendritic spread of the postsynaptic neurons. In all these schemes, the real problem consists in how to provide the "teacher" input (but see Fig. 5).

IV. VISUAL RECOGNITION OF THREE-DIMENSIONAL OBJECTS AND FACE-SENSITIVE NEURONS

We have recently suggested and demonstrated how to use a HyperBF network to learn to recognize a three-dimensional object. This section reviews very briefly this work (Poggio and Edelman 1990) and then suggests that the brain may use a similar strategy. Face-sensitive neurons are discussed as a specific instance.

HyperBF Networks for Recognizing Three-dimensional Objects

A three-dimensional object gives rise to an infinite variety of two-dimensional images or views, because of the infinite number of possible poses relative to the viewer, and because of arbitrarily different illumination conditions. Is it possible to synthesize a module that can recognize an object from any viewpoint, after it learns its three-dimensional structure from a small set of perspective views? We have recently shown (Poggio and Edelman 1990) that the HyperBF scheme may provide a solution to the problem provided that relatively stable and uniquely identifiable features (that we will call "labeled" features) can be extracted from the image.

In our scheme, 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 labeled and visible feature points on the object. We assume that a view of an object is a vector of this type (instead of position in the image of feature points, we have also used angles between corners and length of segments or both), in general augmented by components that represent other properties of the object not necessarily related to its geometric shape, such as color or texture. We also assume that the function that maps the views into 0, 1 (0 if the view is of another object, 1 if the view is of the correct object) can be approximated by a smooth function (if this were false, one could approximate the mapping from the view to a "standard" view and then apply a radial function to the result, see Poggio and Edelman 1990).

The network used for this task is shown in Figure 3 (see also Fig. 4). In the simplest version (fixed centers), the centers correspond to some of the examples, i.e., some views of the object. Updating the centers is equivalent to modifying the corresponding "prototypical views." Updating the weights of the matrix **W** corresponds to changing the relative importance of the various features that define the views of an object. This is important in the case in which these features are of a completely different type: A large w indicates a larger weight in the feature in the measure of similarity and is equivalent to a small σ in the Gaussian function. Features with a small role have a very large σ: Their exact position or value does not matter much.

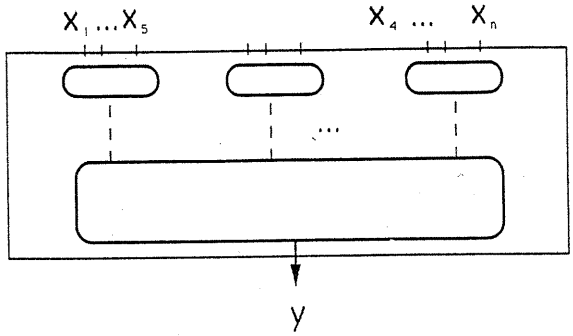

Figure 4. A hierarchical scheme in which HyperBF modules are inputs to another HyperBF module. As an example, a scheme of this type may be used for three-dimensional object recognition in the general case of spurious and missing features. Instead of encoding all n features, one encodes only subsets of dimensions d, where $d < n$. The input to each of the first row of modules is a different set of features of the object; the output is a value between 0,1 that indicates the degree of certainty that the input is the sought object. The last module is a decision module that integrates the various inputs. Notice that all modules could be synthesized by learning through independent sets of examples.

An interesting conclusion of this work consists of the small number of views required to recognize an object from the infinite number of possible views. The results clearly show that the scheme avoids the main problem of look-up table schemes, the explosion in the number of entries. Furthermore, the performance of the HyperBF recognition scheme resembles human performance in a related task. As discussed in Poggio and Edelman (1990), 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 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° × 30° patches to cover the viewing sphere).

Recently, H. Buelthoff and S. Edelman (in prep.) have obtained interesting psychophysical results that support this model for human recognition of a certain class of three-dimensional objects against other possible models. In general, the experimental results fit closely the prediction of theories of the two-dimensional interpolation variety and appear to contradict theories that involve three-dimensional models.

Face-sensitive Neurons

The HyperBF recognition scheme we have outlined has suggestive similarities with some of the data about visual neurons responding to faces obtained by Perrett and co-workers recording from the temporal association cortex (see Perrett et al. 1987 and references therein; Poggio and Edelman 1990). Let us consider the network of Figure 3 as the skeleton for a model of the circuitry involved in the recognition of faces. One expects different modules, one for each different object of the type of the network of Figure 3. One also expects hierarchical organizations: For instance, a network of the HyperBF type may be used to recognize certain types of eyes and then may serve as input to another network involved in recognizing a certain class of faces, which may be itself one of the inputs to a network for a specific face. Different types of cells may then be expected. The overall output of a network for a specific face may be identified with the behavioral responses associated with recognition and may or may not coincide with an individual neuron. There should be cells or parts of cells corresponding to the centers, i.e., to the prototypes used by the networks. The response of these neurons should be a Gaussian function of the distance of the input to the template. 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, two-dimensional Gaussian receptive fields looking at a retinotopic map of features. 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.

Some of these expectations are consistent with the experimental findings of Perret et al. (1987). Some of the neurons described have several of the properties expected from the units of a HyperBF network with a center, i.e., a prototype that corresponds to a view of a specific face. Some of the main data (from Perret et al. 1987 and references therein) follow.

1. The majority of cells responsive to faces are sensitive to the general characteristics of the face, and they are somewhat invariant to its exact position and attitude.
2. Presenting parts of the face in isolation revealed that some of the cells responded to different subsets of features: Some cells are more sensitive to parts of the face such as eyes or mouth.
3. There are cells selective for a particular view of the head. Some cells were maximally sensitive to the front view of a face, and their response fell off as the head was rotated into the profile view, and others were sensitive to the profile view with no response to the front view of the face.
4. There are cells that are specific to the views of one individual. It seems that for each known person there would be a set of "face recognition units." Our model applies most directly to these neurons.

V. THEORIES OF THE CEREBELLUM AND OF MOTOR CONTROL

Cerebellum Models of Marr and Albus

The cerebellum is a part of the brain that is important in the coordination of complex muscle movements. The neural organization of the cerebellum is highly regular and well known (see Fig. 5). Marr (1969) and Albus (1971) modeled the cerebellum as a look-up table. The critical part of their theories is the assumption that the synapses between the parallel fibers and the Purkinje cells are modified as a function of the Purkinje cell activity *and* the climbing fibers input. I suggest (see Fig. 5) that the cerebellum is a HyperBF network or set of networks (one for each Purkinje cell). Instead of a simple look-up table, the cerebellum would be a *function approximation module* (in a sense, "an approximating look-up table"). In our conjecture, basket and Golgi cells would have different roles from the roles assumed in the Marr-Albus theory. In particular, the

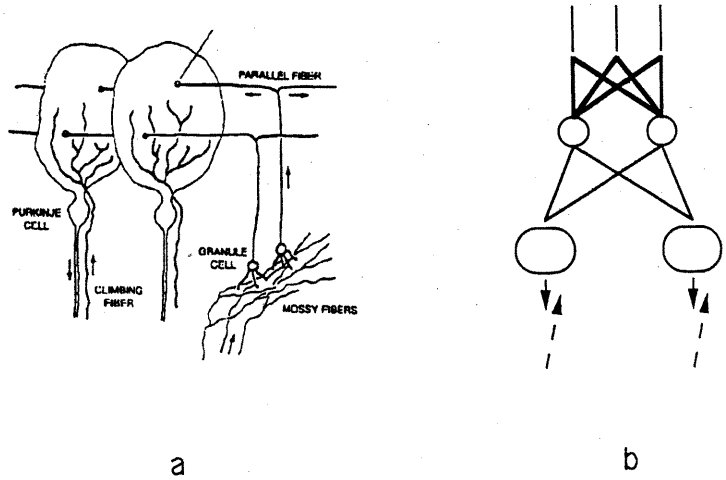

Figure 5. (*a*) A sketch of the neurons of the cerebellum and their connections. In our conjecture, these would be the basic elements of a HyperBF network: The mossy fibers are the inputs, the granule cells correspond to the various centers and basis functions $G(x, x_i)$, the Purkinje cells correspond to the output units that summate the weighted activities of the basis units, whereas the climbing fibers carry the "teacher" signal y_i. The strength of the synapses between the parallel fibers and the Purkinje cells would correspond to the c_α. (*b*) The corresponding HyperBF network has two basis functions corresponding to the two granule cells in *a* and two output summation units corresponding to the two Purkinje cells in *a*.

Golgi cells, which receive inputs from the parallel fibers and whose axons synapse on the granule cells/mossy fibers clusters, may be used to change the norm weights W.

Key assumptions include: (1) granule cells correspond to basis units (there may be as many as 200,000 granule cells per Purkinje cell) representing as many "examples"; (2) Purkinje cells are the outputs of the network; (3) climbing fibers are responsible for modifying synapses from granule cells to the Purkinje cell.

Theories of Motor Control

There are at least two aspects of motor control in which HyperBF modules could be used: (1) to compute smooth, time-dependent trajectories (for instance arm trajectories) given sparse points such as initial, final, and intermediate positions; (2) to associate to each position in the trajectory the appropriate field of muscle forces. These two problems may be solved by two modules that can be used in series, the first one providing the input to the second one (see Fig. 6a,b). I first consider the problem of computing appropriate smooth trajectories from sparse points in space-time. An interesting question is: Are HyperBFs a plausible implementation for Flash and Hogan's minimum jerk principle for the coordination of arm movements? Flash and Hogan (1985) found experimental evidence that arm trajectories minimize jerk, i.e., $C = \|x^{(3)}\|^2 + \|y^{(3)}\|^2$, where $x^{(3)}$ is the third temporal derivative of x. This suggests a regularization principle with a stabilizer corresponding to additive quintic splines. HyperBF could implement it using basis units recruited for the specific motion (as many as there are constrained points) with Gaussian-like or spline-like time-dependent activities (boundary conditions may have to be taken into account). The weights would be learned during training. As Morasso and Mussa-Ivaldi (1982) implied, approximation schemes of this type amount to composition of elemental movements. It is interesting to observe that jerk is automatically minimized by the linear superposition of the appropriate elemental movements, i.e., the appropriate Green's functions. Thus, a scheme of the Morasso-Mussa-Ivaldi type can be made to be perfectly equivalent to the Flash-Hogan minimization principle. The fact that the minimum jerk principle can be implemented directly by a HyperBF network is attractive from the point of view of a biological implementation, since biologically implausible direct minimization procedures are not required anymore. The minimization is implicit in the form of the elemental movements; weighted superposition of the elemental movements seems a much easier operation to implement in the motor system than explicit minimization.

The second problem requires a neural circuit that associates an equilibrium position to an appropriate activation. Bizzi (see, e.g., E. Bizzi et al., in prep.) suggests that a group of spinal cord interneurons specify the limb's final position and configuration

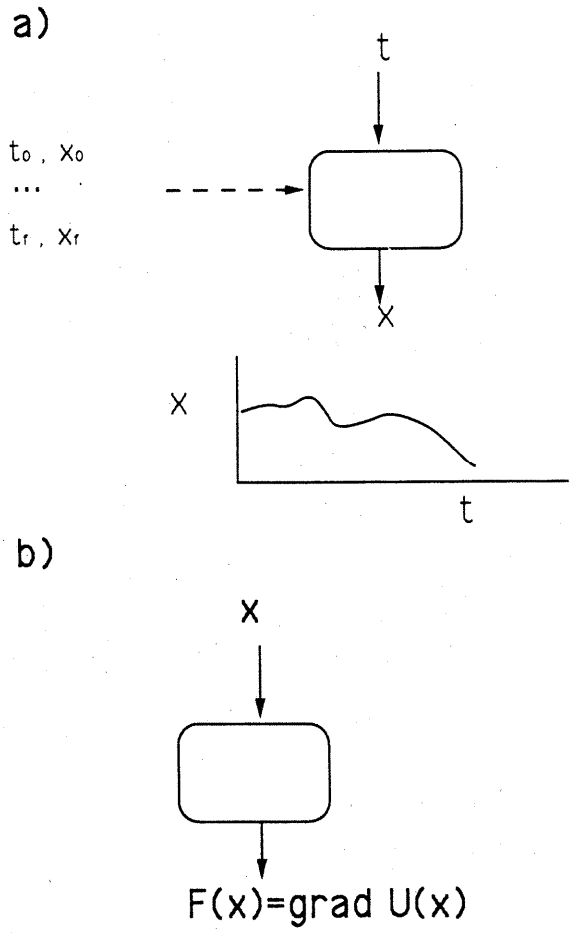

Figure 6. Two problems in motor control: (*a*) determining the trajectory x(*t*) from a small set of points (t_i, x_i) on the desired trajectory and (*b*) computing the field of muscle forces for each of the points on the trajectory. The figure suggests that two different HyperBF modules may be used to perform both tasks. In *a*, a HyperBF module approximates the trajectory from the sparse points by superimposing Gaussian distributions with the appropriate weights in such a way as to satisfy some minimum-jerk-like principle. In *b*, a module of the HyperBF type has been synthesized during development and continuously adapted to generate the appropriate field of forces for each equilibrium position x. It is similar to an approximating look-up table. A behavior of the look-up table type was suggested by Bizzi because of very recent experimental data (see E. Bizzi et al., in prep.).

through a field of muscle forces that have the appropriate equilibrium point. E. Bizzi et al. (in prep.) propose that the spinal cord contains aspects of motor behavior reminiscent of a look-up table. Their findings extend several results in the area of oculomotor research, where investigators have described neural structures whose activation brings the eyes or the head to a unique position. I suggest that the required look-up table behavior may be implemented through a HyperBF module that requires the storage of only a few equilibrium positions (or correspondingly, a few conservative-like fields, i.e., appropriate activation coefficients for the motoneurons) and can interpolate between them (see Fig. 6). Notice that the synthesis of a conservative field

of muscle force could be achieved through the superposition (with arbitrary weights, over the index α) by the motor system of appropriate elementary motor fields of the form (see F. Mussa-Ivaldi and S. Giszter, in prep.)

$$\phi(x, \alpha) = \frac{x - x_\alpha}{r_\alpha} G'(r_\alpha)$$

with $r_\alpha = \|x - x_\alpha\|$ and G is a radial basis function such as the Gaussian.

VI. SUMMARY: A PROPOSAL FOR HOW THE BRAIN WORKS

The theory proposed in this paper consists of three main points:

1. It assumes that the brain may use *modules that approximate multivariate functions and that can be synthesized from sparse examples* as basic components for several information-processing tasks.
2. It proposes that these modules are realized in terms of HyperBF networks, of which a rigorous theory is now available.
3. It shows how HyperBF networks can be implemented in terms of plausible biophysical mechanisms.

The theory is in a sense a modern version of the grandmother neurons idea, made computationally plausible by eliminating the combinatorial explosion in the number of required cells, which was the main problem in the old idea.

The proposal that much information processing in the brain is performed through modules that are similar to *enhanced look-up tables* is attractive for many reasons. It also promises to bring closer apparently orthogonal views, such as the *immediate perception* of Gibson and the *representational theory* of Marr, since almost iconic "snapshots" of the world may allow the synthesis of computational mechanisms completely equivalent to vision algorithms such as, say, structure-from-motion. The idea seems to change significantly the computational perspective on several vision tasks. As a simple example, consider the different specific tasks of hyperacuity, invented by the psychophysicists. The theory developed here would suggest that an appropriate module for the task, somewhat similar to a new "routine," may be synthesized by learning in the brain.

Notice that the theory makes two independent claims: The first is that the brain can be explained in part in terms of approximation modules, the second is that these modules are of the HyperBF type. The second claim implies that the modules are an extension of look-up tables. Notice that there are schemes other than HyperBF that could be used to extend look-up tables. Notice also that multilayer Perceptrons, typically used in conjunction with back-propagation, can also be considered as approximation schemes, albeit still without a convincing mathematical foundation. Unlike HyperBF networks, they cannot be interpreted as direct extensions of look-up tables (they are more similar to an extension of multidimensional Fourier series).

The theory suggests that population coding (broadly tuned neurons combined linearly) is a consequence of extending a look-up table scheme (corresponding to interval coding) to yield interpolation (or more precisely approximation, since the examples may be noisy), that is, generalization.

The theory suggests some possibly interesting ideas about the evolution of intelligence. It also makes a number of predictions for physiology and psychophysics. More work is needed to specify sufficiently the details and some of the basic assumptions of the theory in order to make it useful to biologists. The next sections deal with these last three points.

Evolution of Intelligence: From Memory to Computation

There is a duality between *computation* and *memory*. Given infinite resources, the two points of view are equivalent: For instance, I could play chess by precomputing winning moves for every possible state of the chessboard. More to the point, notice that basic logical operations can be defined in terms of *truth tables* and that all Boolean predicates can be represented in disjunctive normal form, i.e., as a look-up table.

Given that the brain probably has a prodigious amount of memory and given that one can build powerful approximating look-up tables using techniques such as HyperBF, is it possible that part of intelligence may be built from a set of souped-up look-up tables? One advantage of this point of view is to make it perhaps easier to understand how intelligence may have evolved from simple associative reflexes. In more than one sense (biophysical and computational), HyperBF-like networks are a natural and rather straightforward development of very simple systems of a few neurons showing basic learning phenomena such as classic conditioning.

Predictions and Remarks

General Predictions

1. Computation, as generalization from examples, emerges from the superposition of receptive fields in a multidimensional input space.
2. Computation is performed by *Gaussian receptive fields* and their combination (through some approximation to multiplication), rather than by threshold functions.
3. The theory predicts the existence of low-dimensional feature-like cells and multidimensional Gaussian-like receptive fields, somewhat similar to template-like cells, a fact that could be tested experimentally on cortical cells.
4. The HyperBF scheme is a general-purpose circuitry, used in the brain to synthesize module that can be regarded as approximating look-up tables. If this

point of view is correct, we expect the same basic kind of neural machinery to be replicated in different parts of the brain across different modalities (in particular in different cortical areas).
5. The "programming style" used by the brain in solving specific perceptual and motor problems is to synthesize appropriate architectures from modules of the type shown in Figure 1 (a very simple architecture built from the basic module of Fig. 1 is shown in Fig. 4).

Face Neurons

1. Some of the face cells correspond to basis functions with centers in a high-dimensional input space and are somewhat similar to prototypes or coarse "grandmother cells."
2. They could be synthesized as the conjunctions of features with Gaussian-like distance from the prototype.
3. Face cells are *not* detectors; often several may be active simultaneously. The output of the network is a combination of several prototypes.
4. From our preliminary experiments (Poggio and Edelman 1990), the number of basis cells that are required per object is about 40–80 for the full viewing sphere, but much less (10–20) for each aspect (e.g., frontal views). I conjecture that a similar estimate holds for faces.
5. Input to the face cells are features such as eye positions, mouth position, and hair color.
6. Eye features cells may be themselves the output of HyperBF networks specialized for eyes.

Cerebellum

1. The cerebellum is a set of approximation modules for learning to perform motor skills (both movements and posture).
2. Its neurons are elements of a HyperBF network: The mossy fibers are the inputs, the granule cells correspond to the basis functions $G(\mathrm{x}, \mathrm{x}_i)$, the Purkinje cells correspond to the output units that summate the weighted activities of the basis units, whereas the climbing fibers carry the "teacher" signal y_i.
3. The strength of the modifiable synapses between the parallel fibers and the Purkinje cells corresponds to the c_α.
4. Golgi cells may be involved in modifying during learning the center positions $\mathbf{t}_\alpha$ and the norm weights $\mathbf{W}$.

Motor Control

The qualitative expectation is to find cells and circuits corresponding to the two stages shown in Figure 6. Spinal cord neurons, according to very recent data by E. Bizzi et al. (in prep.), specify the limb's final position and configuration.

Future

The proposal of this paper is just a rough sketch of a theory. Many details (some of them critical) need to be filled in. Some basic questions remain: For instance, how reasonable is the idea of supervised learning schemes? To say it in a different and perhaps more constructive way, what are the systems that can be synthesized from building blocks that are just function approximation modules? What types of tasks can be solved by systems of that type? On the biological side of the theory, the obvious next task is to develop detailed proposals for the circuitries underlying face recognition and motor control (including the circuitry of the cerebellum) that take into account up-to-date physiological and anatomical data.

Notes to Section I

1. Segmentation of an image in parts that are likely to correspond to separate objects is probably the most difficult problem in vision. Remember that already in the Perceptron book (Minsky and Papert 1969) recognition-in-context was shown to be significantly harder than recognition of isolated patterns. We assume here that this problem has been "solved," at least to a reasonable extent.
2. The same basic machinery in the brain may be used for synthesizing many different, "small" learning modules, as components of many different systems. This is very different from suggesting a single giant network that learns everything.

Notes to Section II

The relevant derivatives for optimization methods that need them are for the c_α

$$\frac{\partial H[f^*]}{\partial c_\alpha} = -2 \sum_{i=1}^{N} \Delta_i G(\|\mathrm{x}_i - \mathbf{t}_\alpha\|_{\mathrm{W}}^2) \tag{6}$$

for the centers $\mathbf{t}_\alpha$

$$\frac{\partial H[f^*]}{\partial \mathbf{t}_\alpha} = 4c_\alpha \sum_{i=1}^{N} \Delta_i G'(\|\mathrm{x}_i - \mathbf{t}_\alpha\|_{\mathrm{W}}^2)\mathbf{W}^{\mathrm{T}}\mathbf{W}(x_i - \mathbf{t}_\alpha) \tag{7}$$

and for $\mathbf{W}$

$$\frac{\partial H[f^*]}{\partial \mathbf{W}} = -4\mathbf{W} \sum_{\alpha=1}^{n} c_\alpha \sum_{i=1}^{N} \Delta_i G'(\|\mathrm{x}_i - \mathbf{t}_\alpha\|_{\mathrm{W}}^2) Q_{i,\alpha} \tag{8}$$

where $Q_{i,\alpha} = (\mathrm{x}_i - \mathbf{t}_\alpha)^T$ *is a dyadic product and* G' is the first derivative of G (for details, see Poggio and Girosi 1990a).

Notes to Section III

1. There are many nonradial functions derived from our regularization formulation, such as tensor product splines, that are factorizable.

2. I have assumed here that all centers have the same **W**. It is possible to have sets of different Green's functions, each set with its own **W** (see Poggio and Girosi 1990a).
3. It is natural to imagine hierarchical architectures based on the HyperBF scheme: A multidimensional Gaussian "template" unit may be a "feature" input for another radial function (again because of the factorization property of the Gaussian). Of course, a whole HyperBF network may be one of the inputs to another HyperBF network.
4. I conjecture that Equation 8 could be approximated by a Hebbian-like rule for the elements of the diagonal **W** such as

$$w_k(t+1) = w_i(t) - \sum_{\alpha=1}^{n} c_\alpha \gamma(x_k(t) - (t_\alpha)_k) y_k(t) \tag{9}$$

where **y** is the output of the upper layer of Figure 1a, i.e., $\mathbf{y} = \mathbf{W}\mathbf{x}$ and γ is

$$\gamma = \Delta_i G'(\|\mathbf{x}_i - \mathbf{t}_\alpha\|_{\mathbf{W}}^2) \tag{10}$$

and i labels the ith example. Such a Hebbian rule requires back-connections from later stages in the network to the upper layer—where **W** is updated—in order to broadcast quantities such as the error of the overall network relative to the ith example and the derivative of G' of the activation units.
5. The mechanisms and especially the connections needed to implement the learning equations or some equivalent scheme are an open question, in terms of biological plausibility. More work is needed.

Notes to Section IV

1. The HyperBF scheme 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 the network.
2. Notice that the HyperBF approach to recognition does not require as inputs the x, y coordinates of image features: Other parameters of appropriate features can also be used.
3. In a similar vein, notice that the HyperBF network can provide, with the same centers (but different **c**), other parameters of the object, such as its pose, instead of simply a *yes*, *no* recognition signal.
4. Recognition of noisy and partially occluded objects, using realistic feature identification schemes, requires an extension of the scheme. A natural extension of the scheme is based on the use of multiple lower-dimensional centers, corresponding to different subsets of detected features, instead of one $2N$-dimensional center for each view in the example set. This corresponds to a set of networks capable of recognizing different parts of an object. It is equivalent to a set of networks each with a diagonal **W** with some zero entries in the diagonal, instead of one network with **W** with nonzero diagonal elements.
5. Not all features may be always labeled correctly. In general, one expects a significant "correspondence" problem. Possibly the easiest solution is to generate all reasonable sequence of labels for a given input vector and simply try them out on the network. This is, of course, equivalent to trying in parallel the given input on many networks each with a different labeling of its inputs.
6. An obvious use of these learning/approximation modules based on the HyperBF technique is based on a hierarchical composition of GRBF modules, in which the outputs of lower-level modules assigned to detect object parts and their relative disposition in space are combined to allow recognition of complex structured objects. Figure 4 is an example of this architecture.

Notes to Section V

Zipser and Andersen (1988) have presented intriguing simulations suggesting that a back-propagation network trained to solve the problem of converting visual stimuli in retinal coordinates to head-centered coordinates generates receptive fields similar to the ones experimentally found in cortical area 7 of the monkey. We conjecture that Andersen's data may be better accounted for by a HyperBF network. For simplicity, let us consider the one-dimensional version of the problem Zipser and Andersen propose is solved by neurons in area 7. The position of a spot of light on the retina is given as r; the eye position relative to the head is also known as e. The problem is to compute the position of the spot of light relative to the head, i.e., $h = r + e$. Stated in these terms, the problem is computationally trivial, and its solution simply requires the addition of the two inputs r and e. The situation is, however, more complicated due to the actual representation in which r and e are given. In the equation, r and e are represented as numbers. Zipser and Andersen assume, in accordance with physiology, a different representation: They assume that the position r of a spot of light is coded by the presence or absence of activity of one or more cells in a retinotopic array. From this point of view, the goal of the computation carried out by the network is to change representation from *array representation* to *number representation*.

The simplest solution to the problem of changing from an array representation to a number representation is the following. Assume that only one cell in the array $f(x)$ is excited at any given position, i.e., $f(x) = \delta(r - x)$. Simplifying somewhat the situation assumed by Zipser and Andersen, but not altering it in any significant way, let us assume that e is represented directly as a number or a firing rate. The problem then is to convert the *array representation* $f(x) = \delta(r - x)$ for the retinal position into a number *(or a firing rate)*

representation. Consider a linear unit that summates linearly all inputs with the "receptive field" $w(x)$. The output l is given by $l = \int w(x)f(x)dx$. For $f(x) = \delta(x - r)$, the choice $w(x) = x$ yields $l = r$. Thus, a simple solution to our problem of converting an array representation into a number representation only needs receptive fields that increase linearly with eccentricity (notice that $w[x] = ax$ may also be acceptable; simply a monotonic dependence on x may be a sufficient approximation).

If a Gaussian HyperBF network with a polynomial term of degree one is used to approximate the relation of the equation from a set of input/output examples, some of the basis functions will be linear units such as the ones described above, and some will be the product of two-dimensional Gaussians representing the visual receptive fields and two-dimensional Gaussians representing the eye position. These latter cells would probably account for the multiplicative property of the area 7 cells found by Andersen. We conjecture that other features of the cells could be replicated in a HyperBF simulation.

ACKNOWLEDGMENTS

The ideas of this paper about the biological implications of new function approximation techniques can be found, to a good extent, in Poggio and Girosi (1989). They depend critically on the work done together with Federico Girosi, Shimon Edelman, and Bruno Caprile on the theory and the applications of regularization networks. It is very likely that, although plausible, the ideas are wrong—unlike the work on which they rest. Anya Hurlbert, Sandro Mussa-Ivaldi, and Robert Thau read the manuscript and suggested several good ways to improve it, which I managed to implement only in part. This paper describes research done in part within the Artificial Intelligence (A.I.) Laboratory and the Center for Biological Information Processing in the Department of Brain and Cognitive Sciences. Support for this research is provided by a grant from the ONR Cognitive and Neural Sciences Division, and by the NATO Scientific Affairs Division (0403/87). Support for the A.I. Laboratory's artificial intelligence research is provided by the Advanced Research Projects Agency of the Department of Defense under Army contract DACA-76-85-C-0010 and in part under ONR contract N-00014-85-K-0124. T.P. is supported by the Uncas and Ellen Whitaker chair.

REFERENCES

Albus, J.S. 1971. A theory of cerebellar functions. *Math. Biosci.* **10:** 25.

Bertero, M., T. Poggio, and V. Torre. 1988. Ill-posed problems in early vision. *Proc. IEEE* **76:** 869.

Broomhead, D.S. and D. Lowe. 1988. Multivariable functional interpolation and adaptive networks. *Complex Syst.* **2:** 321.

Flash, T. and N. Hogan. 1985. The coordination of arm movements: An experiment confirmed mathematical model. *J. Neurosci.* **5:** 1688.

Hubel, D.H. and T.N. Wiesel. 1962. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *J. Physiol.* **160:** 106.

Koch, C. and T. Poggio. 1987. Biophysics of computational systems: Neurons, synapses, and membranes. In *Synaptic function* (ed. G.M. Edelman et al.), p. 637. Wiley, New York.

Marr, D. 1969. A theory of cerebellar cortex. *J. Physiol.* **202:** 437.

Micchelli, C.A. 1986. Interpolation of scattered data: Distance matrices and conditionally positive definite functions. *Constr. Approx.* **2:** 11.

Minsky, M.L. and S. Papert. 1969. *Perceptrons.* MIT Press, Cambridge, Massachusetts.

Morasso, P. and F.A. Mussa-Ivaldi. 1982. Trajectory formation and handwriting: A computational model. *Biol. Cybern.* **45:** 131.

Omohundro, S. 1987. Efficient algorithms with neural network behaviour. *Complex Syst.* **1:** 273.

Perrett, D.I., A.J. Mistlin, and A.J. Chitty. 1987. Visual neurones responsive to faces. *Trends Neurosci.* **10:** 358.

Poggio, T. and S. Edelman. 1990. A network that learns to recognize 3D objects. *Nature* **343:** 263.

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

———. 1990a. Extension of a theory of networks for approximation and learning: Dimensionality reduction and clustering. In *A.I. Memo No. 1167.* Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge.

———. 1990b. Theory of networks for learning. *Science* **247:** 978.

Poggio, T. and V. Torre. 1978. A theory of synaptic interactions. In *Theoretical approaches in neurobiology* (ed. W.E. Reichardt and T. Poggio), p. 28. MIT Press, Cambridge, Massachusetts.

Poggio, T., V. Torre, and C. Koch. 1985. Computational vision and regularization theory. *Nature* **317:** 314.

Poggio, T. and the staff. 1988. MIT progress in understanding images. In *Proceedings image understanding workshop,* Cambridge, Massachusetts, April 1988. Morgan Kaufmann, San Mateo, California.

Poggio, T. and the staff. 1989. MIT progress in understanding images. In *Proceedings image understanding workshop,* Palo Alto, California, May 1989, p. 56. Morgan Kaufmann, San Mateo, California.

Powell, M.J.D. 1987. Radial basis functions for multivariable interpolation: A review. In *Algorithms for approximation* (ed. J.C. Mason and M.G. Cox). Clarendon Press, Oxford.

Tikhonov, A.N. and V.Y. Arsenin. 1977. *Solutions of ill-posed problems,* Winston, Washington, D.C.

Torre, V. and T. Poggio. 1987. An application: A synaptic mechanism possibly underlying motion detection. In *Theoretical approaches in neurobiology* (ed. W.E. Reichardt and T. Poggio), p. 39. MIT Press, Cambridge, Massachusetts.

Zipser, D. and R.A. Andersen. 1988. A back-propagation programmed network that simulates response properties of a subset of posterior parietal neurons. *Nature* **331:** 679.