INTERNATIONAL JOURNAL OF QUANTUM CHEMISTRY, VOL. 42, 727–739 (1992)

A Project for an Intelligent System: Vision and Learning

TOMASO POGGIO AND LUIGI STRINGA
I.R.S.T., Povo, 38100 Trento, Italy

Abstract

One of the most critical aspects of a truly intelligent system is the ability to learn, that is, to improve its own functionality by interacting with the environment and exploring it. In this paper, we argue that learning from exploring the environment should be the main goal in developing artificial intelligence. We also argue in favor of an integrated system—combining several state-of-the-art aspects of artificial intelligence, such as speech, vision, natural language, expert systems—as the experimental platform with which to approach this problem. We then describe the main features of a project of this type, MAIA, which is under development at I.R.S.T. The vision components of the system will be discussed in some detail, especially the navigation architecture of the indoor robot available to MAIA. We will conclude outlining some initial learning problems that will be approached within the MAIA project, such as learning to recognize faces and learning to update the map of the Institute used for indoor navigation.

1. A New Definition of Artificial Intelligence

The "Turing test" has represented for several years a definition of intelligence against which most workers in artificial intelligence (AI) have implicitly measured their own goals and achievements. It is an operational definition: If a computer behaves in a way indistinguishable from a human person, then it can be called intelligent. Recent criticisms of AI, for instance by Searle [1], can be summarized neatly by recognizing that they amount to questioning the validity of this definition of intelligence. Criticisms of this type are somewhat moot, since definitions are just definitions. It is, however, interesting to look for definitions of intelligence that are alternative or complementary to Turing's, not in order to claim that the computer can never be intelligent (as Searle and, more recently, Penrose [2] claimed) but in order to better capture the essence of the problem of creating artificial intelligence, at least as it is perceived today after 25 years of work in AI.

Twenty-five years ago, intelligence was mainly reasoning, proving theorems, and playing chess. Today we realize how "intelligent" lower animals are and how complex are the problems that our senses routinely solve. We also realize how intractable is the problem of producing software and how much of it would be needed in order to replicate even just some of the simplest aspects of intelligence (think of the project by Lenat in Austin!). In this perspective, it seems natural to propose a somewhat different definition of intelligence. We suggest that this new definition should emphasize *learning*. Consider an artificial system such as a robot: We may define it as intelligent if it would be able to learn from exploring

 CCC 0020-7608/92/040727-13$04.00

the environment, a little bit like a baby, even if its overall performance would be initially quite low.

It is not completely clear whether our test could be defined in a rigorous way because a system cannot learn from a state of "tabula rasa." The necessity of an adequate combination of "nature" and "nurture" is now widely accepted, also because of overwhelming biological and neurophysiological evidence. In any case, the spirit of our definition should be obvious. To satisfy our "test" and say that we have indeed an intelligent system, it is not enough to build an expert system capable of perfect medical diagnosis or an automatic translator from English to Chinese. A system, on the other hand, that speaks a grammatically incorrect Italian but has learned it in a way similar to how a child learns to speak would certainly pass our "test." We propose that a system that can improve itself with learning and by exploring its environment should be called intelligent. Our definition has the precise goal of introducing in an explicit way the problem of learning as the "new" frontier in the attempt of understanding and synthesizing intelligence.

Most researchers would now agree that AI has reached a barrier that has blocked its progress in recent years. The barrier, we argue, is the inability of the systems developed so far to evolve by themselves, i.e., to learn without explicit programming. Of course, there have been many attempts—some interesting—of mechanizing learning, and it is clear to most that machine learning is a crucial area in AI.

2. Why An Integrated System

One of the most constructive contributions of AI has been the realization that research in intelligence is an experimental science and that computers—whatever their architecture: analog or digital, parallel or serial, silicon-based or protein-based—are the basic tool with which to test theories. From this point of view, we argue that an attack on the problem of learning requires an appropriate experimental platform that can interact with its environment and explore it. It follows that the system should have sensory and probably motor capabilities adequate for dealing with the physical world. A purely electronic agent, interacting with a simulated or nonphysical environment, will probably be inappropriate for exploring the problems of learning that are basic to humanlike intelligence.

In a similar way, it seems important to develop an integrated platform that may acquire and process different types of information such as spoken and written language in addition to visual images. Notice that the sum of relatively simple behaviors can easily originate a behavior that a human observer will likely judge as complex and even intelligent. Furthermore, an integrated platform of the type we envisage is not simply a multisensor system but rather a system that exploits multiple sources of information. After all, any system ought to know a substantial amount in order to learn efficiently.

All the above reasons led to the formulation of MAIA (acronym for *Modello Avanzato di Intelligenza Artificiale*), the main project now starting at I.R.S.T., which has the goal of developing an integrated platform with which to attack the problem of learning.

3. The Project MAIA

An automatic "concierge" for the Institute is the metaphor for the main function of MAIA (see Figure 1). The system will respond to spoken and written questions about the function and the organization of I.R.S.T. It will be able to send a mobile platform to guide a visitor to a certain office in the Institute or to perform functions such as surveillance. It will have "terminals" in the Institute, such as an electronic librarian, capable of recognizing persons from their faces and voices and capable of reading the titles of borrowed books, again from images captured by a CCD camera. The system will interact with people in various ways, such as voice, written text, touch screens, telecontrol, and the usual keyboard. It will explore the physical environment of the Institute through its "tentacles"—i.e., mobile platforms of various sizes and dimensions, capable of navigating in the corridors of the Institute on the basis of several types of sensors and of a map of the building.

Different fields of AI will converge in the realization of MAIA, which is, in fact, an excuse for forcing their integration. Speech understanding, natural language, planning, knowledge-based systems, man–machine interfaces of the hypermedia type, and vision are all technologies that MAIA will require. In the following, we will concentrate on the vision component of MAIA and discuss briefly our initial results.

4. The Vision Components of MAIA

There are three main vision projects within MAIA (Figure 2).

- Indoor navigation.
- Text segmentation and optical character recognition (OCR).
- Face recognition.

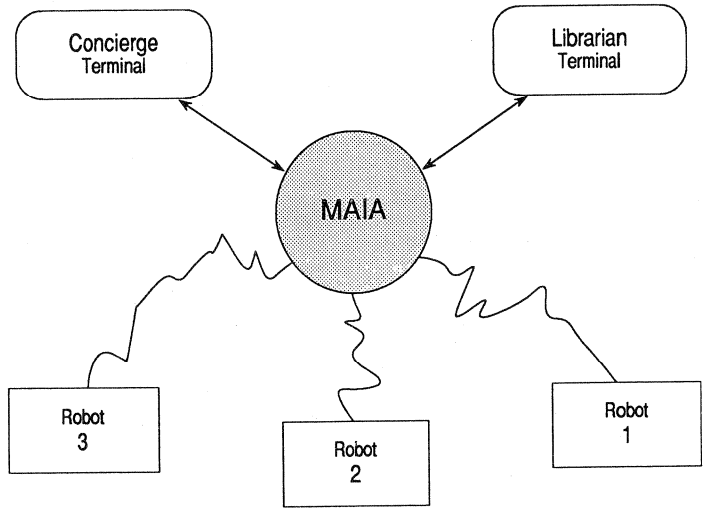

Figure 1. Some of the "terminals" of MAIA.

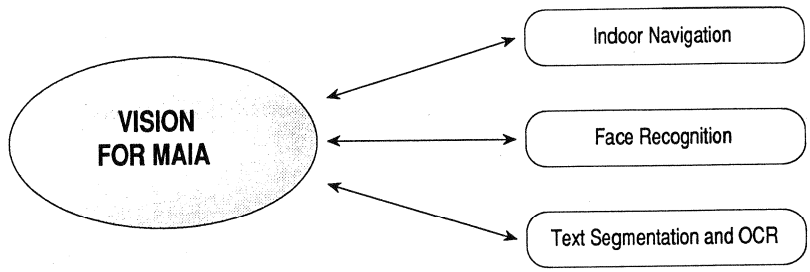

Figure 2. The main vision projects within MAIA.

4.1. The Architecture of the Navigation System of MAIA

The project has a well-defined goal within MAIA: to provide the robot with the capabilities of navigating in the Institute between any location A and any location B in the map. More fundamentally, it is an experimental sandbox for exploring different architectures for indoor navigation and control.

The navigation architecture is shown in Figure 3. It consists of a set of autonomous routines for navigation, controlled by a planner. Each routine is synthesized as a feedback loop between the sensors and the actuators and supported by different sensory modules: the vision module, the ultrasound modules, and the odometer. We plan to explore rather classical control routines in which sensory inputs update the state of the system that is used to control the robot. These rou-

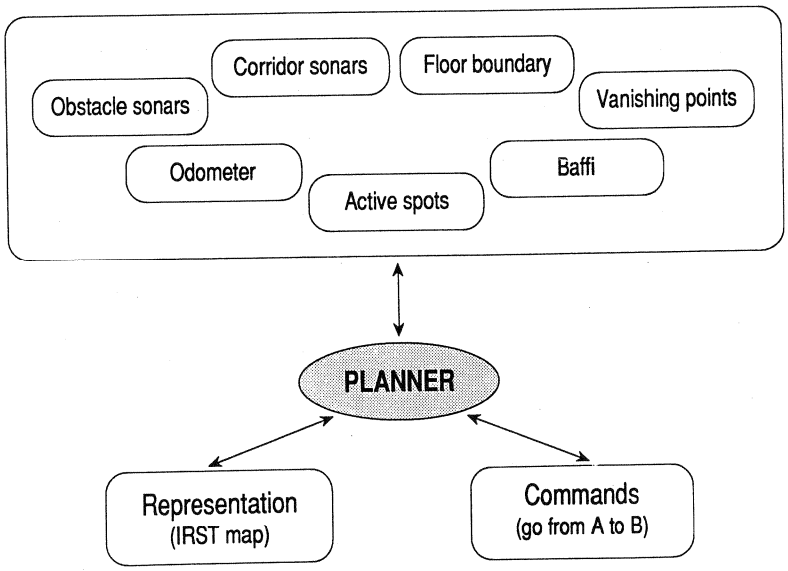

Figure 3. A sketch of the architecture of the navigation system for the mobile platform of MAIA. A set of simple "reflexes" that use visual sensors, ultrasounds, and the odometer is programmed by a simple planner that has access to a map of the Institute.

tines are of the *representational* type, since their characteristic feature is the continuous estimate of a state vector representation of the robot and of the world. We are also exploring more *reflexive* architectures composed of routines in which the sensory inputs are controlling directly and independently the robot without an explicit representation of the state. We expect to be able to define the advantages and disadvantages of these two approaches and come up with the "correct" one, probably a combination of both.

The main vision module is the floor boundary sensor FBS. The FBS [3] uses two CCD cameras to estimate the distance of the robot from the walls from the position in the image of the boundaries between floor and walls. It exploits the a priori knowledge about usual modern indoor environments with their vertical walls and their horizontal floors. The FBS supports routines of the two classes described above (Figure 4):

- The way the apparent boundary floor–wall changes in the image plane (whether it shifts parallel to itself or whether it changes slope) directly con-

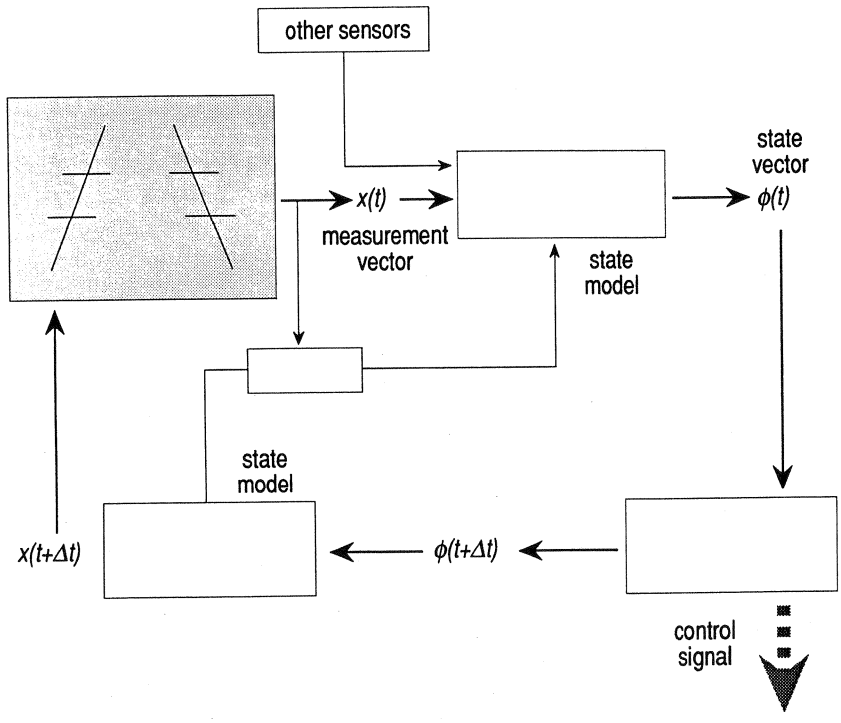

Figure 4. The main visual reflex, based on measurements of the position in the image of the boundaries between floor and walls that drive the estimation through Kalman filtering of the state vector of the platform and the environment. The estimate of the state vector is used to control in real time the platform and to estimate the new position in the image of the boundaries. This last step stabilizes the feature extraction step. The overall "reflex" is similar to the scheme used by Dickmanns and by the BMW group to drive a car.

trols the robot. This is an example of *reflexive control* that works without any representation of the state of the robot, directly on measurements on the image.

- Measurements of the position of the boundary in the image plane are used to estimate through a calibration function and Kalman filtering the position of the robot in the corridor relative to the wall. Control is performed on the basis of the state vector. This is an example of *representational control.*

The FBS can support several specific *navigation routines,* such as to go straight in a corridor keeping the midline or to go straight until a corner is detected. The other modules can support similar routines. The problem of integrating different sensors within single routines has an elegant solution within the representational approach: Kalman filtering (see [4]) can be used to integrate the different types of measurements into an estimate of the state vector. The reflexive approach does not offer—it seems—a general solution to this problem but only the heuristic suggestion of simple interactions between sensors such as reciprocal activation and inhibition.

An important component of the navigation architecture is the *planner,* which in our approach is very simple. It has access to a map of the Institute, which in the version now implemented is a bitmap of free spaces and obstacles and which will soon represent the connection between commands in natural language and the navigation system. One of the most interesting goals of this project is to explore experimentally the boundary between reflexes and a planner. How well will the system function without a planner? Where is the boundary between a pure "insect" and conscious planning? How much can the spinal cord do without the cortex? The answers we expect are what common sense suggests: Reflexes alone can only support very simple behaviors but not more complex and goal-oriented ones; reflexes *with* a planner can support a robust control system. Experiments will tell us whether we are correct.

In addition to a standard, rather large mobile platform that we are modifying extensively with the addition of visual sensors, new ultrasound devices and a 386-based on-board processor, we plan the development of a minirobot with a smaller and cheaper platform of about $40 \times 40 \times 40$ cm. The minirobot will have several functions: the development of sensor and control subsystems of the main platform, the development of cost-effective platforms for automatic surveillance, and the possibility to study in an experimental way modes of interaction between autonomous agents (how to avoid each other, how to cooperate, how to coordinate).

4.2. *Text Segmentation and* OCR

The OCR project capitalizes on a wealth of experience at I.R.S.T. in the field. Its focus will be text-reading from a rather unconstrained image of the cover of a book that will not be at a constant distance from the camera, will not be parallel to the image plane, and may contain characters with poor contrast against different types of background, possibly containing graphics. Therefore, the main

focus of the project will be segmentation of text from background. Beyond its specific relevance to OCR, this problem is of a far more general interest: We believe that "segmentation" of an image in regions likely to represent different objects' surfaces is, at present, the key problem in visual recognition.

The specific task is to read titles and author names from the cover of books presented to the camera in a somewhat unrestricted way: The book will not be perfectly perpendicular to the image plane, and its location in the image will be variable and so, also, its size and distance. Fonts and their background and location will be highly variable. Illumination is assumed to be constant. This specific system goal is also an excuse for exploring a problem of fundamental importance in vision tasks and especially in the recognition of 3D objects: the problem of segmenting and grouping regions in the image that are likely to be associated with distinct objects. Here we are dealing with the problem of segmenting the image of the book from the rest and with the problem of finding and grouping the title and author names against the background of the cover (see [5]).

4.3. Face Recognition

Face recognition is a specific instance of object recognition, possibly the most important visual task. The project has a specific goal—to develop a system for recognizing isolated, frontally viewed faces under controlled illumination—and the more general motivation of exploring the basic problem of 3D object recognition. We divide the problem into two basic steps:

- Extraction of several features from images of faces, such as the color of the hair and the position of the eyes. This part of the project should define the set of needed features and how to compute them reliably (see [6]).
- Use of the hyper basic functions (Hyper BF) technique for learning from examples (described in the section on learning) to synthesize modules that can recognize each face. The first step is to extend the work of Poggio and Edelman [7], described in the next section, from simulated objects to real objects. Since they used simulated wire frames, we plan to test our extended technique at first on real wire-frame objects (paper clips).

4.3.1. Application of the HyperBF Technique to Object Recognition: Poggio and Edelman have applied the HyperBF technique to the problem of 3D object recognition with promising results [7]. They have been able to synthesize a module that can recognize an object from any viewpoint, after it learns its 3D structure from a small set of 2D perspective views, using the HyperBF network scheme. Their results were obtained so far with simulated wire-frame objects and assumed that the problems of feature extraction and matching were already solved. We summarize in the following the main point of their work.

Shape-based visual recognition of 3D objects requires the solution of at least two difficult problems. The first problem is the variability of object appearance due to changing illumination, which may be addressed by working with relatively stable features, such as intensity edges. The remaining problem, the removal of the variability due to the unknown pose of the object, may be solved by first hy-

pothesizing the viewpoint (e.g., using information on feature correspondences between the image and a model), then computing the appearance of the model of the object to be recognized from that viewpoint and comparing it with the actual image. Generally, recognition schemes of this type employ 3D models of objects. Automatic learning of 3D models is in itself a difficult problem. Consequently, few present schemes learn to recognize objects from examples and most use 3D models acquired through user interaction.

Is the need for 3D range-based or manually specified models real? Structure from motion theorems indicate that full information about the 3D structure of an object represented as a set of feature points (at least five to eight) is present in just two of their perspective views, provided that corresponding points are identified in each view. A view is represented as a $2N$ vector $x_1, y_1, x_2, y_2, \ldots, x_N, y_N$ of the coordinates on the image plane of N labeled and visible feature points on the object.

Poggio and Edelman have approached this problem by assuming that all features are visible, as they are in wire-frame objects [7]. In principle, therefore, having enough 2D views of an object is equivalent to having its 3D structure specified. This line of reasoning, together with properties of perspective projection, suggested to them (a) that for each object there exists a smooth function mapping any perspective view into a "standard" view of the object and (b) that this multivariate function may be synthesized, or at least approximated, from a small number of views of the object. Such a function would be object-specific, with different functions corresponding to different 3D objects. Furthermore, the application of the function that is specific for one object to the views of a different object is expected to result in a "wrong" standard view that can be easily detected as such.

Synthesizing an approximation to a function from a small number of sparse data—the views—can be considered as learning an input–output mapping from a set of examples and fits well the HyperBF scheme described above.

Poggio and Edelman [7] demonstrated a successful application of HyperBF to the recognition problem. They considered the special case of recognizing a wire-frame 3D object from any of its perspective views with N feature points (we mainly used $N = 6$). A HyperBF module, trained on several tens of random views, was shown capable of mapping any new view of the same object into a standard view (e.g., into one of the training views, chosen initially).

Poggio and Edelman [7] also explored the use of a smaller number of basis functions than training views and used gradient descent to look for the optimal locations of the centers $\mathbf{t}_\alpha$ in addition to the optimal value of the c_α. They found satisfactory performance with just two basis units (for 10–40 training views and with the attitude of the object limited to one octant of the viewing sphere). This suggests that a small number of units are needed for each aspect of an opaque object.

Notice that the HyperBF approach to recognition does not require as inputs the x, y coordinates of image features: Other parameters of appropriate features could also be used, such as corner angles or segment lengths, or the color and the

texture of the object. Recognition of noisy and partially occluded objects, using realistic feature identification schemes, requires an extension of the scheme, even if the problems of object segmentation and selection are addressed separately. A possible extension of the scheme involves a hierarchical composition of HyperBF 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.

5. Learning and MAIA

As we said at the beginning of the paper, the main long-term goal of the MAIA project is to attack the problem of learning from experience. There are several instances of this general problem that will be approached in the immediate future. In the area we have considered, vision, two learning problems are being considered right now:

- The synthesis of a module capable of recognizing faces from a series of examples of that face (see previous section).
- The update of the map of the Institute by the robot exploring the Institute.

Obviously, the problem of learning does not have a single solution, i.e., a single algorithm for all cases. The two examples sketched above show that there are different types of learning that probably require different solutions. A technique that seems relatively general is the HyperBF technique that we plan to use for object recognition and for synthesizing control modules of the robot. In the following, we summarize the main features of the method. More details can be found in the papers by Poggio et al. [8–13].

5.1. The HyperBF technique

This section describes a technique for synthesizing approximation modules through learning from examples. These modules could be used for the task of face recognition and other tasks, such as control problems. 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 and input–output mapping and a standard approximation problem: 2-D 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.*

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 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. 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.* 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 HyperBF (see Figure 5). The main result [8] is that the regularization approach is equivalent to an expansion of the solution in terms of a certain class of functions:

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

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

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

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 function 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 [14] in the radial case, a superposition of functions like Eq. (1) is equivalent to a network of the type shown in Figure 5. The interpretation of Eq. (2) is simple: In the 2D case, for instance, the surface is approximated by the superposition of, say, several 2D Gaussian distributions, each centered on one of the data points.

The network associated with Eq. (2) can be made more general in terms of the following extension:

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

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

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

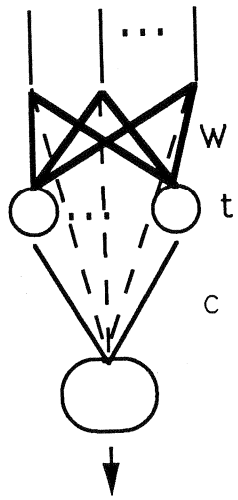

Figure 5. A HyperBF network equivalent to a module for approximating a scalar function of three variables from sparse and noisy data. The data, a set of points where the value of the function is known, can be considered as examples to be used during learning. The hidden units evaluate the function $G(\mathbf{x}; \mathbf{t}_n)$, 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, $G = G(\|\mathbf{x} - \mathbf{t}_n\|_W^2)$ 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 also be connections computing the polynomial term of three constant and linear terms (the dotted lines in the figure) that may be expected in most cases.

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 5. Notice that a sigmoid function at the output may be sometimes useful without increasing the complexity of the system (see Poggio and Girosi [8]). 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 $\mathbf{W}$. Notice that two or more sets of Gaussians, each with its own (diagonal) $\mathbf{W}$, are equivalent to sets of Gaussians with their own σ's.

5.1.1. The Learning Equations: 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_i, and the $\mathbf{t}_\alpha$, that minimize an error functional on the set of examples.

Gradient-descent is probably the simplest approach for attempting to find the solution to this problem, though, of course, it is not guaranteed to converge. We define

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

with

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

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

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

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

$$\dot{\mathbf{W}} = -\omega \frac{\partial H[f^*]}{\partial \mathbf{W}} + \boldsymbol{\Omega}(t),$$

where $\eta_\alpha(t)$, $\boldsymbol{\mu}_\alpha(t)$, and $\boldsymbol{\Omega}(t)$ are the white noise of the zero mean and ω is a parameter. The important quantities—which can be used in more efficient schemes than gradient descent—are

- for the c_α:

$$\frac{\partial H[f^*]}{\partial c_\alpha} = -2 \sum_{i=1}^{N} \Delta_i G(\|\mathbf{x}_i - \mathbf{t}_\alpha\|^2_{\mathbf{W}}), \tag{5}$$

- for the centers t_α:

$$\frac{\partial H[f^*]}{\partial \mathbf{t}_\alpha} = 4c_\alpha \sum_{i=1}^{N} \Delta_i G'(\|\mathbf{x}_i - \mathbf{t}_\alpha\|^2_{\mathbf{W}}) \mathbf{W}^{\mathrm{T}}\mathbf{W}(x_i - \mathbf{t}_\alpha), \tag{6}$$

- and for $\mathbf{W}$:

$$\frac{\partial \mathrm{H}[\mathrm{f}^*]}{\partial \mathbf{W}} = -4\mathbf{W} \sum_{\alpha=1}^{\mathrm{n}} \mathrm{c}_\alpha \sum_{\mathrm{i}=1}^{\mathrm{N}} \Delta_\mathrm{i} \mathrm{G}'(\|\mathbf{x}_i - \mathbf{t}_\alpha\|^2_{\mathbf{W}}) Q_{i,\alpha}, \tag{7}$$

where $Q_{i,\alpha} = (\mathbf{x}_i - \mathbf{t}_\alpha)(\mathbf{x}_i - \mathbf{t}_\alpha)^T$ is a dyadic product and G' is the first derivative of G (for details see Poggio and Girosi [9]).

5.1.2. Interpretation of the Network: The interpretation of the network of Figure 5 is the following: *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, i.e., a measure of their similarity, and applies to it the radial function. In the case