AI Magazine Volume 20 Number 3 (1999) (© AAAI)

Machine Learning, Machine Vision, and the Brain

Tomaso Poggio and Christian R. Shelton

■ The problem of learning is arguably at the very core of the problem of intelligence, both biological and artificial. In this article, we review our work over the last 10 years in the area of supervised learning, focusing on three interlinked directions of research—(1) theory, (2) engineering applications (making intelligent software), and (3) neuroscience (understanding the brain's mechanisms of learnings)—that contribute to and complement each other.

Learning is now perceived as a gateway to understanding the problem of intelligence. Because seeing is intelligence, learning is also becoming a key to the study of artificial and biological vision. In the last few years, both computer vision—which attempts to build machines that see—and visual neuroscience—which aims to understand how our visual system works—are undergoing a fundamental change in their approaches. Visual neuroscience is beginning to focus on the mechanisms that allow the cortex to adapt its circuitry and learn a new task. Instead of building a hard-wired machine or program to solve a specific visual task, computer vision is trying to develop systems that can be trained with examples of any of a number of visual tasks. Vision systems that learn and adapt represent one of the most important directions in computer vision research, reflecting an overall trend—to make intelligent systems that do not need to be fully and painfully programmed. It might be the only way to develop vision systems that are robust and easy to use in many different tasks.

Building systems without explicit programming is not a new idea. Extensions of the classical pattern-recognition techniques have provided a new metaphor — learning from examples — that makes statistical techniques more attractive (for an overview of machine learning and other applications, see Mitchell [1997]). As a consequence of this new interest in learning, we are witnessing a renaissance of statistics and function approximation techniques and their applications to domains such as computer vision. In this article, we review our work over the last 10 years in the area of supervised learning, focusing on three interlinked directions of research sketched in figure 1: (1) theory, (2) engineering applications (making intelligent software), and (3) neuroscience (understanding the brain's mechanisms of learning). The figure shows an ideal continuous loop from theory to feasibility demonstrations to biological models feeding back into new theoretical ideas. In reality, the interactions—as one might expect—are less predictable but not less useful. For example in 1990, ideas from the mathematics of learning theory—radial basis function networks—suggested a model for biological object recognition that led to the physiological experiments in cortex described later in the article. It was only later that the same idea found its way into the computer graphics applications described in the conclusions.

Learning and Regularization

In this article, we concentrate on one aspect of learning: supervised learning. *Supervised learning*—or learning from examples—refers to systems that are trained, instead of programmed, by a set of examples that are input-output pairs (x_i, y_i), as sketched in figure 2. At run time, they will hopefully provide a correct output for a new input not contained in the training set. One way to set the problem of learning from examples in a mathematically well-founded framework is the following: Supervised learning can be regarded as the regression problem of interpolating or approximating a multivariate function from sparse data (figure 3). The data are the examples. *Generalization* means estimating the value of the function for points in the input space in which data are not available.

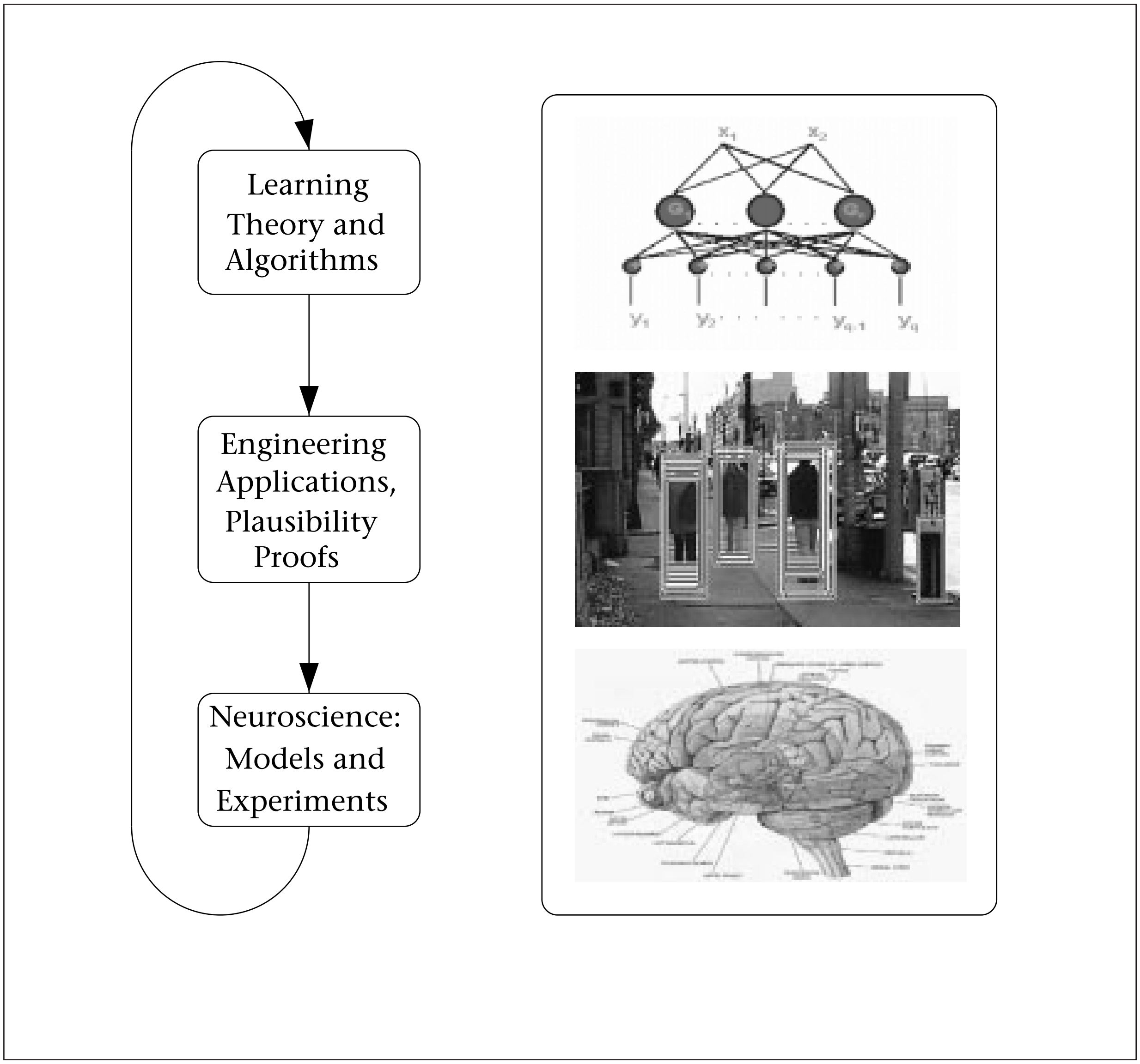


Figure 1. A Multidisciplinary Approach to Supervised Learning.

Once the ill-posed problem of learning from examples has been formulated as a problem of function approximation, an obvious approach to solving it is regularization. *Regularization* solves the problem of choosing among the infinite number of functions that all pass through the finite number of data points by imposing a smoothness constraint on the final solution (as we describe later, it is reasonable to assume that any learnable function is smooth). This results in minimizing the cost functional

$$H[f] = \sum_{i=1}^{N} \left(y_i - f(x_i)\right)^2 + \lambda \| f \|_K^2 \qquad (1)$$

where $\|f\|_K^2$ is a measure of deviation from smoothness of the solution f (see Wahba [1990] and Evgeniou, Pontil, and Poggio [1999]), and the sum is the deviation of the function from the data points (thus we are making a trade-off between accurately modeling the data points and the smoothness of the learned function). For example in the one-dimensional case, using $\|f\|_K^2 = \int dx\, (\partial^2 f(x)/\partial x^2)^2$ in H yields cubic splines as the minimizer $f(x)$ of H.

The use of smoothness stabilizers in the functional equation 1, penalizing nonsmooth functions, can be justified by observing that it would be impossible to generalize for input-output relations that are not smooth, that is, for cases in which "similar" input do not correspond to "similar" output (in an appropriate metric!). Such cases exist: For example, the mapping provided by a telephone directory between names and telephone numbers is usually not "smooth," and it is a safe bet that it would be difficult to learn it from examples!

The functional regularization approach can also be regarded from a probabilistic and Bayesian perspective. In particular, as Girosi, Jones, and Poggio (1995, 1993) (see also Poggio and Girosi [1990a, 1990b] and Wahba [1990]) describe, an empirical Bayes's approach leads to the maximum a posteriori (MAP) estimate of

$$P(f \mid g) \propto P(f)\, P(g \mid f),$$

where the set $g = (x_i, y_i)\ N\ i{=}1$ consists of the input-output pairs of training examples, and f is again the learned function. Under a few assumptions (additive Gaussian noise and a linear Gaussian prior), taking this probabilistic approach to solving the learning problem is equivalent to minimizing equation 1.

Regularization Networks

A key result for our work since 1990 is that under rather general conditions, the solution of the regularization formulation of the approximation problem can be expressed as the linear combination of basis functions, centered on the data points and depending on the input x. The form of the basis function K depends on the specific smoothness criterion, that is, the functional $\|f\|_K^2$. The simplest solution (for several important K such as the Gaussian) is

$$f(x) = \sum_{i=1}^{l} c_i K(x, x_i) \tag{2}$$

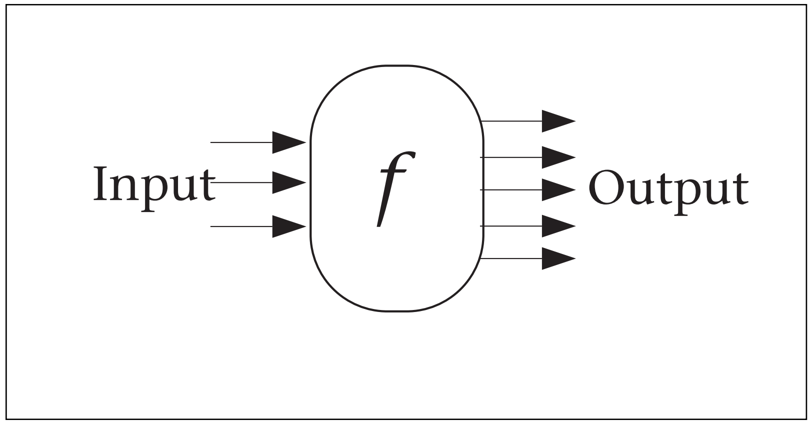


Figure 2. In the Learning-from-Examples Paradigm, We Learn a Function f *from Input-Output Pairs* (x_i, y_i) *Called the Training Set.*

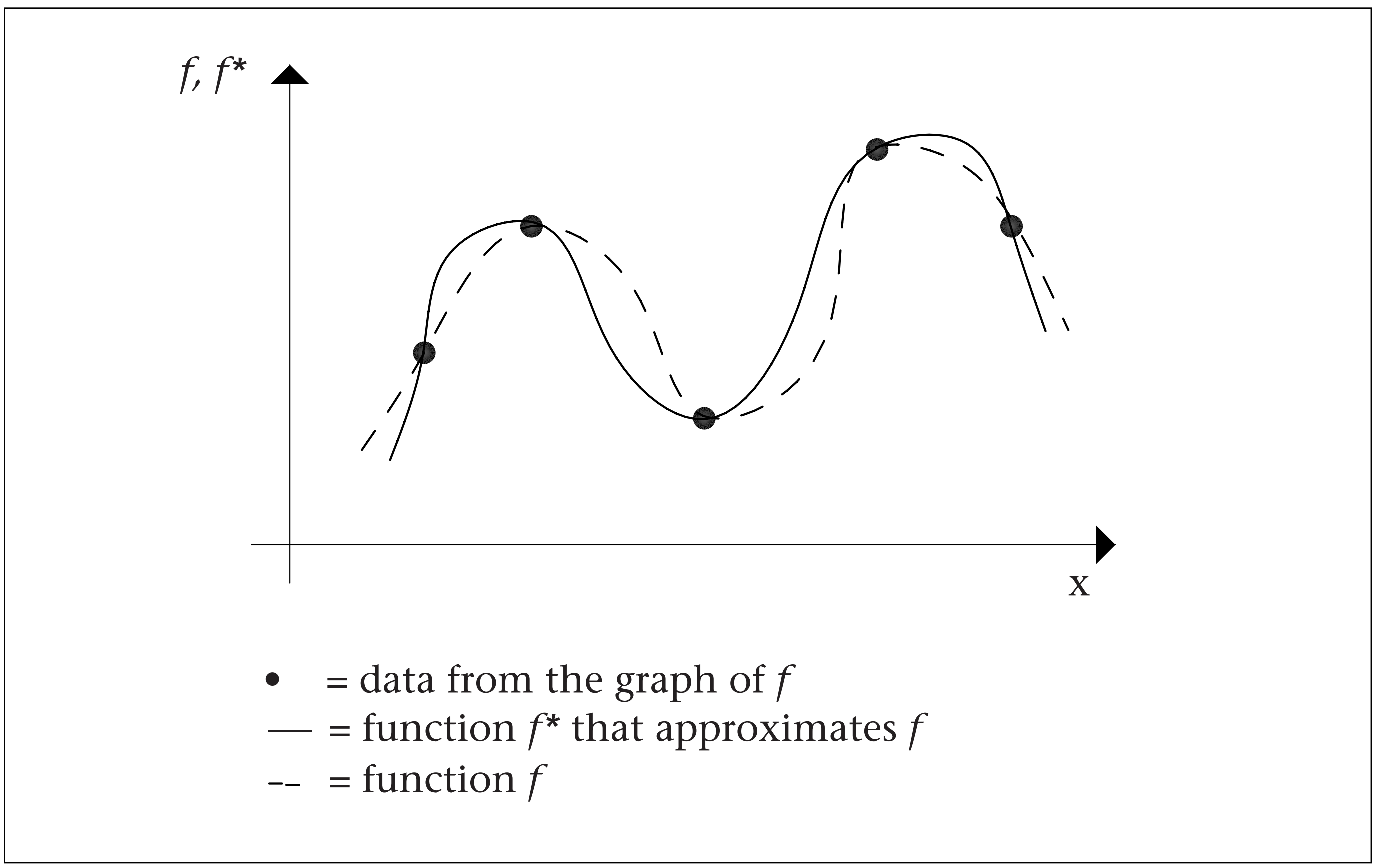


Figure 3. Learning from Examples as Multivariate Function Approximation or Interpolation from Sparse Data.
Generalization means estimating $f^*(x) \approx f(x)$, $\forall x \in X$ from the examples $f^*(x_i) = f(x_i)$, $i = 1, \ldots, N$.

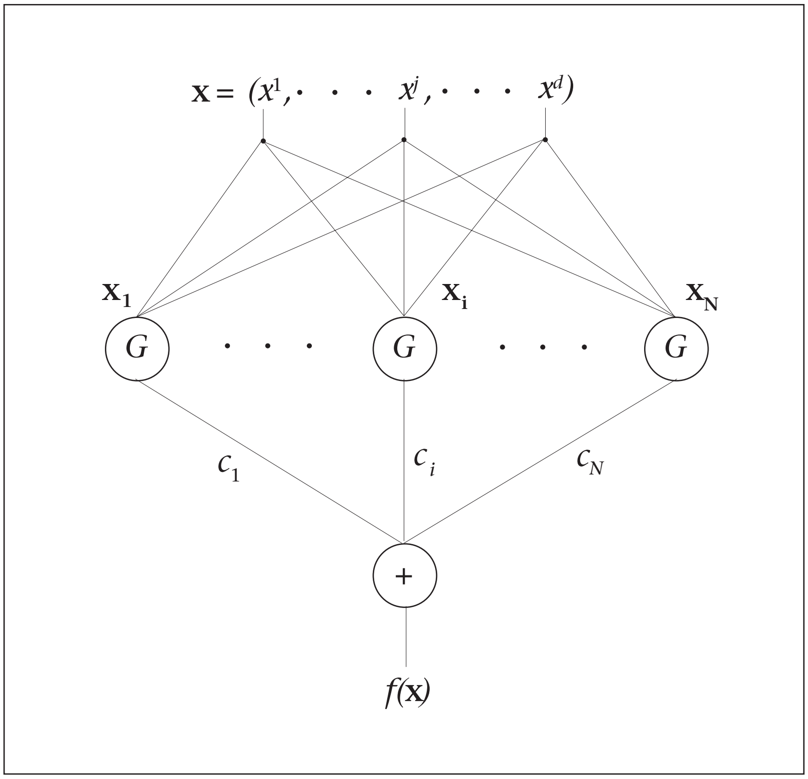


Figure 4. A Regularization Network.

The input vector **x** is *d*-dimensional; there are *N* hidden units, one for each example x_i, and the output is a scalar function $f(x)$.

As observed by Poggio and Girosi (1990b) (see also Broomhead and Lowe [1988]), the solution provided by equation 2 can always be rewritten as a network with one hidden layer containing as many units as examples in the training set (figure 4). We called these networks *regularization networks*. The coefficients c_i that represent the "weights" of the connections to the output are "learned" by minimizing the functional H over the training set (Girosi, Jones, and Poggio 1995).

Radial Basis Functions

An interesting special case arises for radial K. Radial basis function techniques—or radial basis function (RBF) networks—(Girosi, Jones, and Poggio 1995; Poggio and Girosi 1989; Powell 1987; Micchelli 1986) follow from regularization when $K(\mathbf{s}, \mathbf{t})$ is shift invariant and radially symmetric: The best example is a Gaussian $K(\mathbf{s}, \mathbf{t}) = G_\sigma(|\mathbf{s} - \mathbf{t}|^2)$:

$$f(x) = \sum_{i=1}^{l} c_i G_\sigma\left(|\, x - x_i\, |^2\right). \quad (3)$$

In the Gaussian case, these RBF networks consist of units each tuned to one of the examples with a bell-shaped activation curve. In the limit of very small σ for the variance of the Gaussian basis functions, RBF networks become lookup tables. Thus, each unit computes the distance $\|\mathbf{x} - \mathbf{x}_i\|$ of the input vector $\mathbf{x}$ from its center $\mathbf{x}_i$, and in the limiting case of G as a very narrow Gaussian, the network becomes a lookup table, and centers are like templates. Gaussian RBF networks are a simple extension of lookup tables and can be regarded as interpolating lookup tables, providing a very simple interpretation of the result of relatively sophisticated mathematics. The "vanilla" RBF described earlier can be generalized to the case in which there are fewer units than data, and the centers $\mathbf{x}_i$ are to be found during the learning phase of minimizing the cost over the training set. These generalized RBF networks have sometimes been called hyperBF networks (Poggio and Girosi 1990a).

Regularization Provides a General Theory

Several representations for function approximation and regression, as well as several neural network architectures, can be derived from regularization principles with somewhat different prior assumptions on the smoothness of the function space (that is, different stabilizers, defined by different kernels K). They are therefore quite similar to each other.

Figure 5 tries to make the point that regularization networks provide a general framework for a number of classical and new learning techniques. In particular, the radial class of stabilizer is at the root of the techniques on the left branch of the diagram: RBF can be generalized into hyperBF and into so-called kernel methods and various types of multidimensional spline. A class of priors combining smoothness and additivity (Girosi, Jones, and Poggio 1995) is at the root of the middle branch of the diagram: Additive splines of many different forms generalize into ridge regression techniques, such as the representations used in projection pursuit regression (Friedman and Stuetzle 1981); hinges (Breiman 1993); and several multilayer perceptronlike networks (with one hidden layer).

The mathematical results (Girosi, Jones, and Poggio 1995) summarized in figure 5 are useful because they provide an understanding of what many different neural networks do, the function of their hidden units, an approximate equivalence of many different schemes for regression while providing insights into their slightly different underlying (smoothness) assumptions, and a general theory for a broad class of supervised learning architectures.

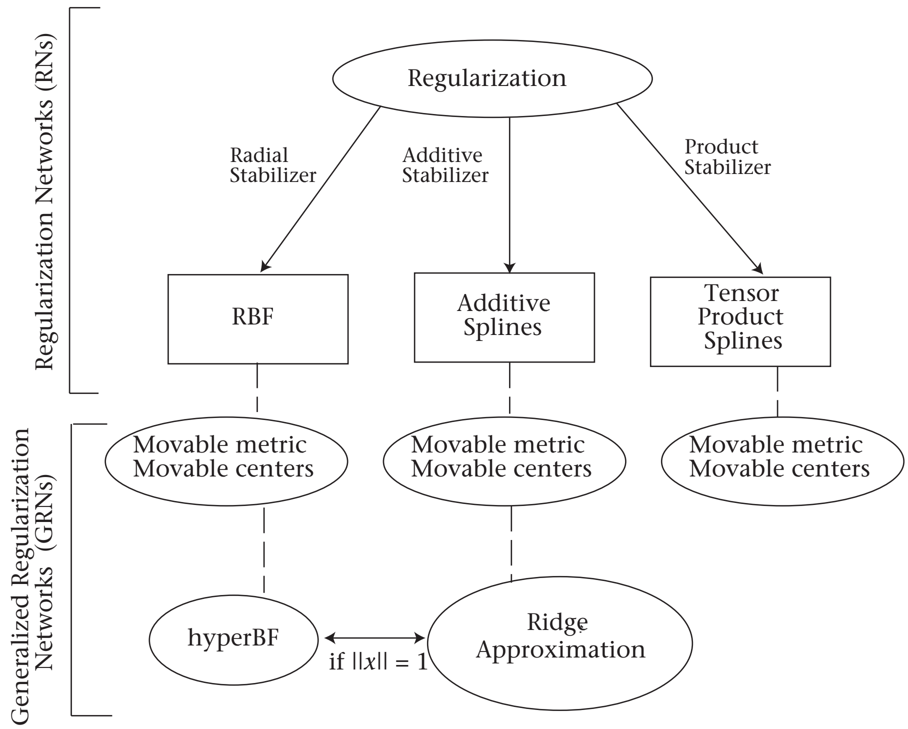


Figure 5. Several Classes of Approximation Schemes and Corresponding Network Architectures Can Be Derived from Regularization with the Appropriate Choice of Smoothness Priors and Associated Stabilizers and Basis Functions, Showing the Common Bayesian Roots (Girosi, Jones, and Poggio 1993).

Several representations for function approximation and regression, as well as several neural network architectures, can be derived from regularization principles with somewhat different prior assumptions on the smoothness of the function space ….

Support Vector Machines and Regularization

Recently, a new learning technique has emerged and become quite popular because of its good performance and its deep theoretical foundations: support vector machines (SVMs), proposed by Vapnik (1995). It is natural to ask the question of its relation with regularization networks. The answer is that it is very closely connected to regularization (Evgeniou, Pontil, and Poggio 1999): It can be regarded as the same type of network, corresponding to exactly the same type of solution f (that is, equation 2) but "trained" in a different way and, therefore, with different values of the weight c_i after the training (Engeniou, Pontil, and Poggio 1999). In particular, in SVM many of the coefficients c_i are usually zero: The x_i corresponding to the nonzero coefficients are called *support vectors* and capture all the relevant information of the full training set.

Support Vector Machines and Sparsity

In recent years, there has been a growing interest in using sparse function approximators. An analogy to human speech owed to Stefan Mallat (of wavelet fame) provides the right intuition. If one were to describe a concept using a small dictionary of only three thousand English words, the description of most concepts would require long sentences using all of most of the three thousand words. However, if one were to describe a concept using a large dictionary of 100,000 words, only a small number of the words would be required for most concepts.

As we mentioned, in SVMs many of the weights c in the sum of equation 2 are zero. The link to sparsity can be made formal: Girosi (1998) proved that, loosely speaking, the sparsest representation (in a certain sense, see Girosi [1998]) is also the one with the best prediction and generalization abilities. The result suggests that a sparse representation of a signal (for

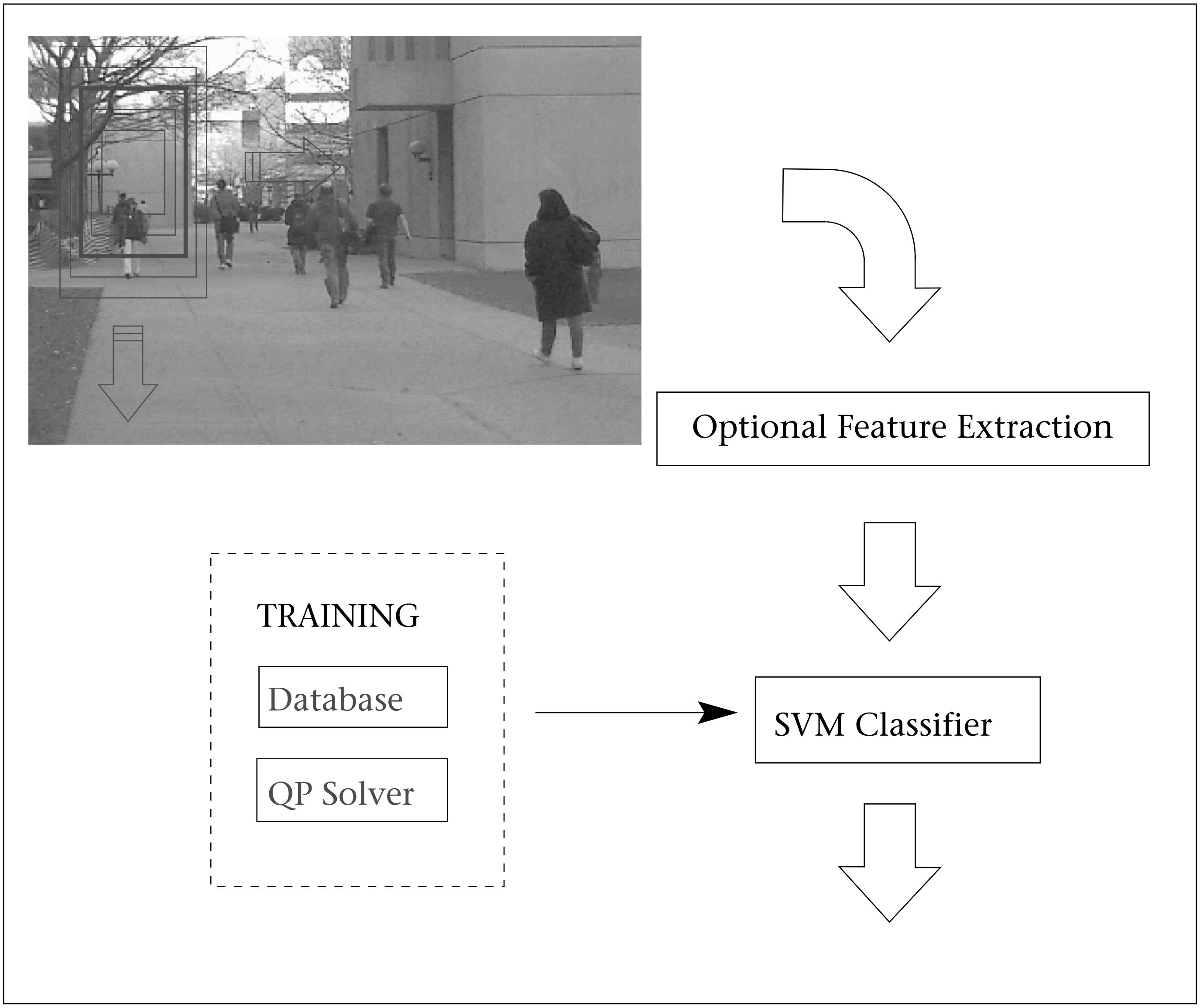


Figure 6. Architecture of Support Vector Machine System for Object Detection.

example, images) from a large dictionary of features is optimal for generalization.

Finally, it is important to observe that until now the functionals of classical regularization have lacked a rigorous justification for a finite set of training data. Vapnik's seminal work has laid the foundations for a more general theory that justifies a broad range of regularization functionals for learning from finite sets, including classical regularization and support vector machines for regression and classification. The basic idea is that for a finite set of training examples, the search for the best model or approximating function has to be constrained to an appropriately small hypothesis space (which can also be thought of as a space of machines or models or network architectures). Vapnik's theory characterizes and formalizes these concepts in terms of the capacity of a set of functions and capacity control depending on the training data: For example, for a small training set, the capacity of the function space in which f is sought has to be small, whereas it can increase with a larger training set. A key part of the theory is to define and bound the capacity of a set of functions. Evgeniou, Pontil, and Poggio (1999) show how different learning techniques based on the minimization of the H functionals listed earlier can be justified using a slight extension of the tools and results of Vapnik's statistical learning theory.

Object Detection with Support Vector Machines

One can only ask, "Does all the theory mean anything?" The mathematics of the previous section suggest that a sparse regularization network (such as a support vector machine) will perform well in classification tasks.

We present here two systems based on the theory outlined in the previous sections—they use support vector machine (SVM) classifiers of the form of figure 4 and equation 2—that learn to detect and classify objects of a specific class in complex image and video sequences. In both systems, the goal is to take an image and find whether and where the object of interest is in the image.

Both use the same architecture (depicted in figure 6). A window is translated across the image. At each translation step, the subwindow of the image masked by the sliding window is fed into a feature extractor (which can return features of the image or just the raw pixel values) whose output is then given to a support vector classifier. This classifier was previously trained using labeled examples of subimages. To achieve detection at multiple scales, the image is rescaled to different sizes and the translation rerun at the new scales. Thus, the output of the classifier on a particular subimage indicates whether the object exists at that location and scale.

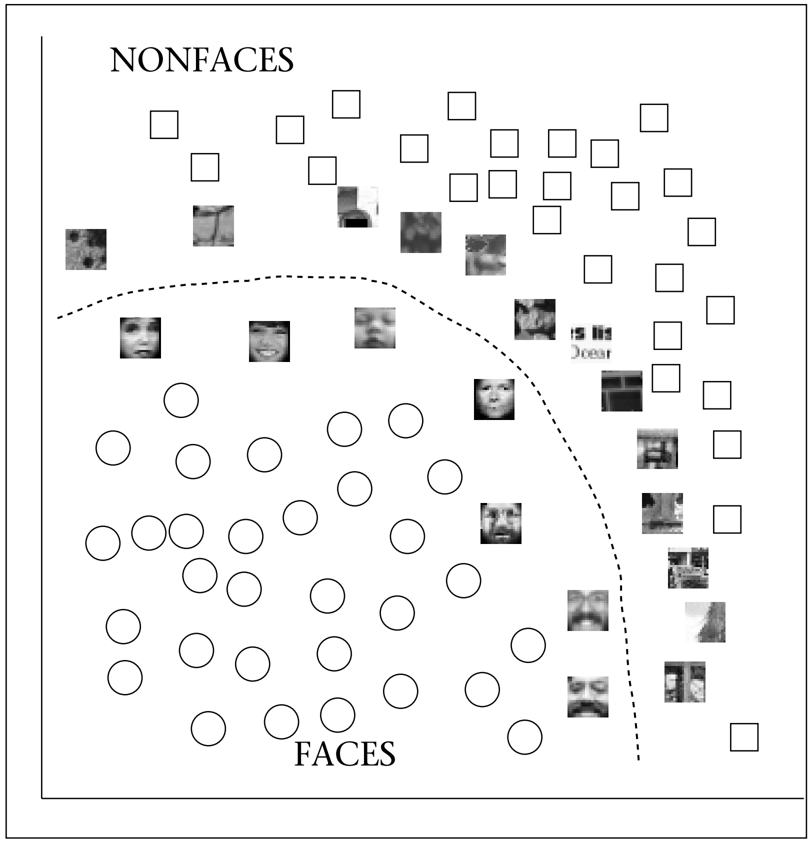


Figure 7. Some of the Support Vectors Found by Training for Face Detection (Osuna, Freund, and Girosi 1997).

Face Detection

For face detection, the goal is to identify the position and scale of all the faces in the image. The subwindow for this task was 19 x 19 pixels, and no feature extraction was used (the gray-scale intensity values from the subimage were fed directly to the classifier). The full system details are described in Osuna, Freund, and Girosi (1997). Here, we just quote some of the results from their experiment.

After training an SVM, most of the examples are automatically discarded because many of the c_i of equation 2 are zero, which is related to the theoretical connection between the SVM framework and sparsity and results in a network that depends only on a few boundary examples (the support vectors). Theoretically, these examples helped to define the decision boundary. Figure 7 shows a few examples from the face-detection system of Osuna et. al. It is interesting to note that they appear to be the most "unfacelike" of the face images and the most "facelike" of the nonface images. Put another way, they are the most difficult training examples and the ones mostly likely to be confused later and therefore the ones that should be remembered to classify new examples correctly.

These learned support vectors and their associated weights were used in a network, as shown in figure 4, to do classification. Some examples of the results of the system are shown in figure 8.

Pedestrian Detection

Using the same system architecture, we can attempt to learn to detect pedestrians. Unfortunately, because pedestrians are a far more varied class of objects, using a subwindow of the pixel values is not sufficient for good performance.

To solve this problem, we add a feature-extraction step (as shown in figure 6) to build an overcomplete, multiscale set of the absolute values of Haar wavelets as the basic dictionary with which to describe shape. These wavelets are simple differencing filters applied to the image at different resolutions, resulting in roughly 1300 coefficients for each subwindow. The full system is described in depth in Papageorgiou, Evgeniou, and Poggio (1998); Papageorgiou, Oren, and Poggio (1998); Oren et al. (1997); and Papageorgiou (1997).

Figure 8. Results of the Face-Detection System (Osuna, Freund, and Girosi 1997).

Because the sensitivity of the system to pedestrians can be adjusted, we can trade off the number of undetected pedestrians (false negatives) against the number of incorrect detected nonpedestrians (false positives). Figure 9 plots a curve showing the performance of the system for various settings of the sensitivity. The upper left corner represents an ideal system that classifies all pedestrians correctly and does not signal nonpedestrian image patches as pedestrians. These ROC curves were computed over an out-of-sample test set gathered around the Massachusetts Institute of Technology and over the internet.

The different plots in figure 9 correspond to different sets of features. Shown are the receiver operating characteristics (ROC) curves for three systems: (1) color processing with all 1326 features, (2) color processing with 29 features, and (3) gray-level processing with 29 features.

The ROC curve shows the difference in the performance resulting from the choice of features. What is not shown (for clarity) is the impact of changing the kernel function. Changes to the kernel used in the SVM had little effect on the final performance (those shown are for polynomials of degree 3).

As expected, using color features results in a more powerful system. The curve of the system with no feature selection is clearly superior to all the others, indicating that for the best accuracy, using all the features is optimal. When classifying using this full set of features, we pay for the accuracy through a slower system. It might be possible to achieve the same performance as the 1326 feature system with fewer features; this is an open question, however. Reducing the number of features is important to reducing the running time of the final detection system. Examples of processed images are shown in figure 10; these images were not part of the training set.

The system has also been extended to allow detection of frontal, rear, and side views of pedestrians. It is currently installed in an experimental car at Daimler. Figure 11 shows the results of processing a video sequence from this car driving in downtown Ulm, Germany. The results shown here are without using any motion or tracking information; adding this information to the system would improve results. From the sequence, we can see that the system generalizes extremely well; this test sequence was gathered with a different camera, in a different location, and in different lighting conditions than our training data.

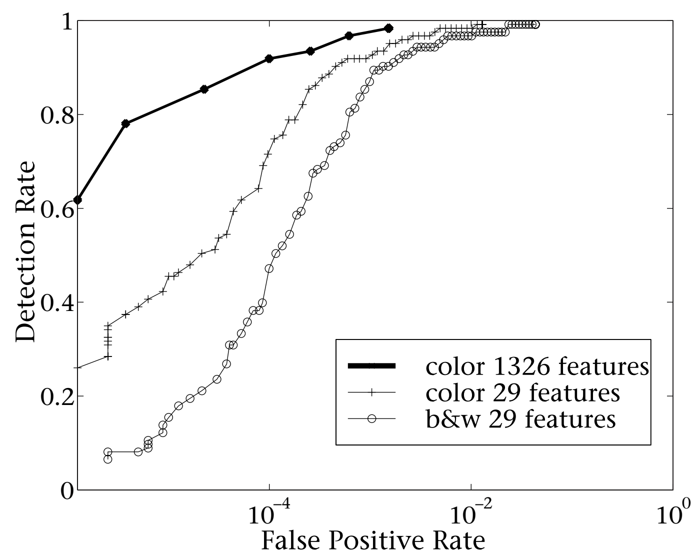


Figure 9. Receiver Operarting Characteristics (ROC) Curves for Different Detection Systems.

The detection rate is plotted against the false positive rate, measured on a logarithmic scale. The false detection rate is defined as the number of false detections for each inspected window (Papageorgiou, Evgeniou, and Poggio 1998).

Object Recognition in the Inferotemporal Cortex

As an example of neuroscience research, in this section, we present results from the inferotemporal cortex, believed responsible for some forms of object recognition.

View-Based Object Recognition

As we mentioned in the introduction, 10 years ago a learning approach to object recognition—based on Gaussian radial basis functions—suggested a view-based approach to recognition (Poggio and Edelman 1990). Regularization networks store a number of examples in the hidden nodes and compare the current input to each of those stored examples in parallel. Instead of having an explicit three-dimensional (3D) model of the object we want to recognize, we instead have a number of 2D examples of what the object looks like, and we compare a current view against each of the stored examples. Different simulations with artificial (Poggio and Edelman 1990) and real "wire-frame" objects (Brunelli and Poggio 1991), and also with images of faces (Beymer 1993; Romano 1993), showed that a view-based scheme of this type can be made to work well.

It was not surprising that one of the first questions we asked was whether a similar approach might be used by our brain. As Poggio and Girosi (1989) and Poggio (1990) argued, networks that learn from examples have an obvious appeal from the point of view of neural mechanisms and available neural data. In a certain sense, networks such as Gaussian RBFs are an extension of a very simple device: lookup tables. The idea of replacing computation with memory is appealing, especially from the point of view of biological and evolutionary plausibility. Interpolating or approximating memory devices such as RBF avoids many of the criticisms of pure lookup table theories. It was therefore natural for our group to try to see how far we could push this type of brain theory.

Somewhat surprising to us, over the last 10 years many psychophysical experiments (for the first such work see Bülthoff and Edelman [1992]) have supported the example-based and view-based schemes that we suggested as one of the mechanisms of object recognition. More recent physiological experiments have provided a suggestive glimpse on how neurons in inferotemporal cortex (the area of the brain responsible for object recognition) can represent objects. The experimental results seem again to agree (so far!) to a surprising extent with the model (Logothetis, Pauls, and Poggio 1995). We are now developing a more detailed model of the circuitry and the mechanisms underlying the properties of the view-tuned units of the model (Riesenhuber and Poggio 1998).

View-Based Model

Here, we review briefly our model and the physiological evidence for it. Figure 12 shows our basic module for object recognition. Classification of a visual stimulus is accomplished

Figure 10. Results from the Pedestrian Detection System.

Typically, missed pedestrians are the result of occlusion or lack of contrast with the background. False positives can be eliminated with further training (Papageorgiou, Evgeniou, and Poggio 1998).

by a network of units. Each unit is broadly tuned to a particular view of the object.

We refer to this optimal view as the *center* of the unit and to the unit as a *view-tuned* unit. One can think of it as a template to which the input is compared. The unit is maximally excited when the stimulus exactly matches its template but also responds proportionately less to similar stimuli. The weighted sum of activities of all the units represents the output of the network. The simplest recognition scheme of this type is the Gaussian RBF network (equation 3): Each center stores a sample view of the object and acts as a unit with a Gaussianlike recognition field around the view. The unit performs an operation that could be described as "blurred" template matching. At the output of the network, the activities of the various units are combined with appropriate weights, found during the learning stage.

Consider how the network "learns" to recognize views of the object shown in figure 13. In this simplified and nonbiological example, the input of the network are the x, y positions of the vertexes of the wire-frame object in the image. Four training views are used. After training, the network consists of four units, each one tuned to one of the four views, as in figure 13. The weights of the output connections are determined by minimizing misclassification errors on the four views and using as negative examples views of other similar objects ("distractors").

The figure shows the tuning of the four units for images of the "correct" object. The tuning is broad and centered on the center of the unit, that is, the training view. Somewhat surprisingly, the tuning is also quite selective: The thinly dotted line shows the average response of each of the unit to 300 similar dis-

Figure 11. Processing the Downtown Ulm Sequence with the Frontal, Rear, and Side-View Detection System.

The system performs the detection frame by frame: It uses no motion or tracking. Adding motion information and the capability to integrate detection over time improves results (Papageorgiou, Evgeniou, and Poggio 1998).

tractors (paper clips generated by the same mechanisms as the target; for further details about the generation of paperclips, see Edelman and Bülthoff [1992]).

Even the maximum response to the best distractor is in this case always less than the response to the optimal view. The output of the network, a linear combination of the activities of the four units, is essentially view invariant and still very selective. Notice that each center can be regarded as the conjunction of all the features represented: The Gaussian can be, in fact, decomposed into the product of one-dimensional Gaussians, each for each input component, that is, for each feature. The activity of the unit measures the global similarity of the input vector to the center: For optimal tuning, all features have to be closed to the optimum value. Even the mismatch of a single component of the template can set to zero the activity of the unit. Thus, the rough rule implemented by a view-tuned unit is the conjunction of a set of predicates, one for each input feature, measuring the match with the template. However, the output of the network is performing an operation more similar to the Or of the output of the units.

This example is clearly a caricature of a view-

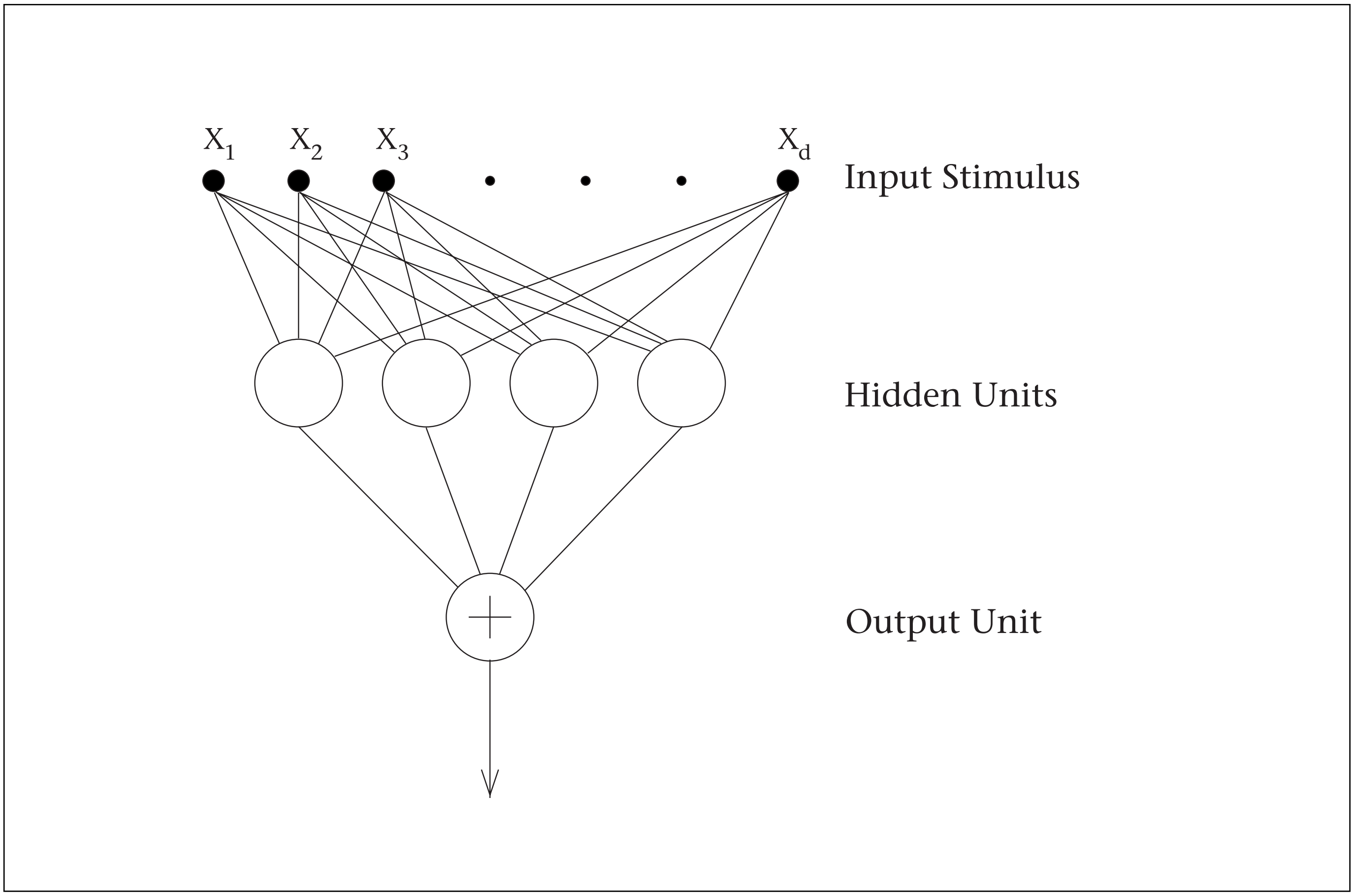


Figure 12. A Gaussian Radial Basis Function Network with Four View-Tuned Units That, after Training, Are Each Tuned to One of the Four Training Views Shown in the Next Figure.

The resulting tuning curve of each of the units is also in the next figure. The units are view dependent and selective, relative to distractor objects of the same type.

based recognition module, but it helps to make the main points of the argument. Of course, biologically plausible features are different from the coordinates of the corners used by the toy network described earlier. We (Riesenhuber and Poggio 1998; Bricolo, Poggio, and Logothetis 1997) recently performed simulations of a biologically more plausible network in which we first filter the image through a bank of directional filters of various orders and scale, similar to V1 neurons (cells in the part of the brain through which the visual information first passes). Before describing in more detail the model work on the circuitry underlying the properties of view-tuned cells, we summarize the physiological findings (Logothetis and Pauls, 1995; Logothetis, Pauls, and Poggio 1995).

Experimental Evidence

Two monkeys were trained to recognize computer-rendered objects irrespective of position or orientation. The monkeys first were allowed to inspect an object, the target, presented from a given viewpoint and subsequently were tested for recognizing views of the same object generated by rotations. In some experiments, the animals were tested for recognizing views around either the vertical or the horizontal axis, and in some other experiments, the animals were tested for views around all three axes. The images were presented sequentially, with the target views dispersed among a large number of other objects, the distractors. Two levers were attached to the front panel of the chair, and reinforcement was contingent on pressing the right lever each time the target was presented. Pressing the left lever was required on presentation of a distractor. Correct responses were rewarded with fruit juice.

An observation period began with the presentation of a small fixation spot. Successful fixation was followed by the learning phase,

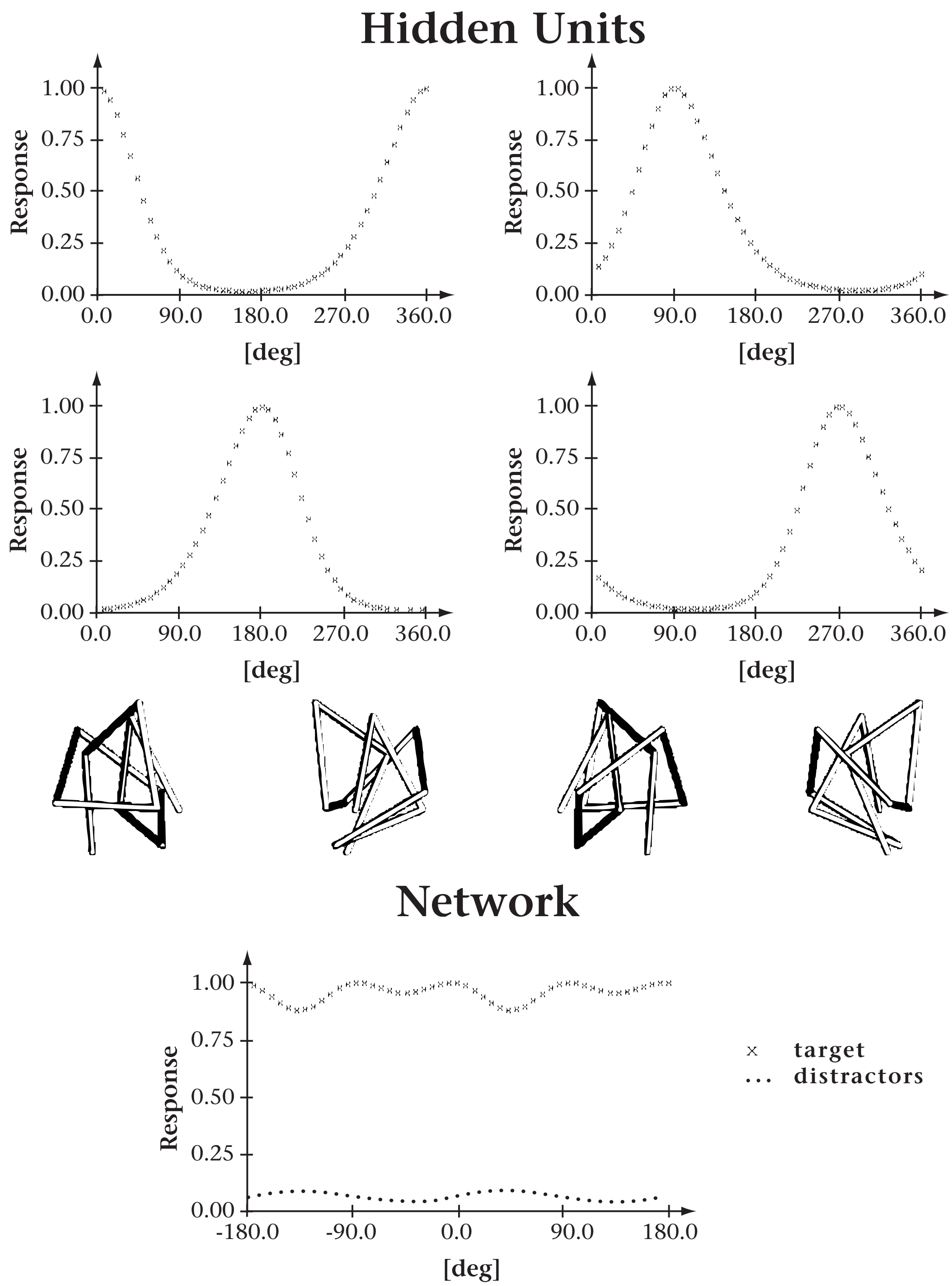


Figure 13. Tuning Each of the Four Hidden Units of the Network of the Previous Figure for Images of the "Correct" Three-Dimensional Objects.

The tuning is broad and selective: The dotted lines indicate the average response to 300 distractor objects of the same type. The bottom graphs show the tuning of the output of the network of the previous figure after learning (that is, computation of the weights *c*): It is view invariant and object specific. Again, the dotted curve indicates the average response of the network to the same 300 distractors (Vetter and Poggio 1992).

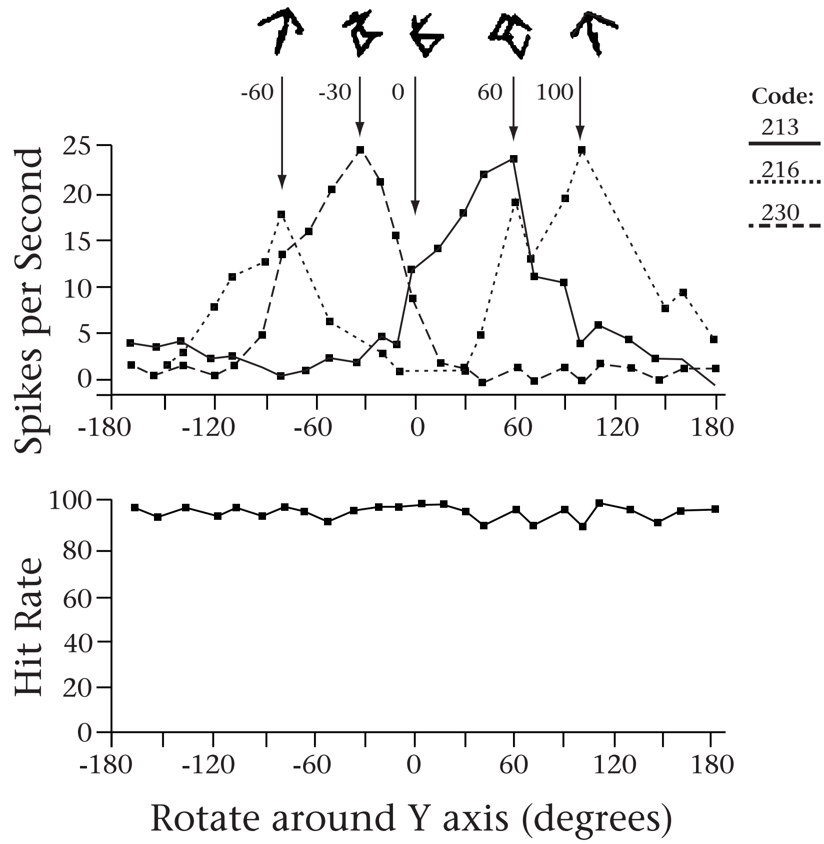


Figure 14. The Top Graph Shows the Activity of Three Units in IT Cortex, as a Function of the Angle of the Stimulus View.

The three neurons are tuned to four different views of the same object, in a similar way to the units of the model of figure 12 and figure 13. One of the units shows two peaks for two mirror symmetric views. The neurons firing rate was significantly lower for all distractors (not shown here). The bottom graph represents the almost perfect, view-invariant behavioral performance of the monkey for this particular object for which he was extensively trained (Logothetis and Pauls 1995).

whereby the target was inspected for two seconds from one viewpoint, the training view. The learning phase was followed by a short fixation period after which the testing phase started. Each testing phase consisted of as many as 10 trials, in each of which the test stimulus, a shaded, static view of either the target or a distractor, was presented.

A total of 970 IT cells were recorded from two monkeys during combined psychophysical-electrophysiological experiments. Logothetis and coworkers found a significant number of units that showed a remarkable selectivity for individual views of wire objects that the monkey was trained to recognize.

Figure 14 shows the responses of three units that were found to respond selectively to four different views of a wire object (wire 71). The animal had been exposed repeatedly to this object, and its psychophysical performance remains above 95 percent for all tested views, as can be seen in the lower plot of figure 14.

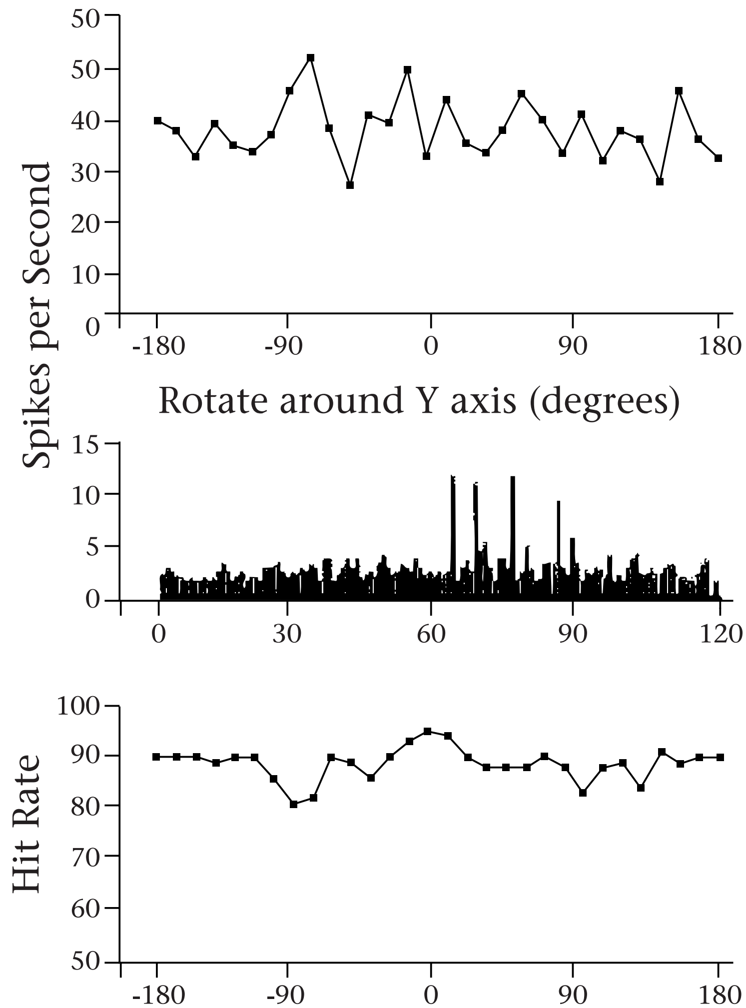


Figure 15. The Monkey Performed Quite Well on This Particular Wire after Extensive Training (Bottom Graph).

A neuron in IT was found that shows a view-invariant response, with about 30 spikes/second to any view of the wire object (top). The response of the cell to any of the 120 distractors is lower, as shown in the middle graph (Logothetis and Pauls 1995). This response is similar to the output unit of the model of figure 12 (see figure 13).

Notice that one of the three neurons is tuned to a view and its mirror image, consistent with other theoretical and psychophysical work. Figure 14 is surprisingly similar to figure 13 showing the response of the view-tuned hidden units of the model of figure 12.

A small percentage of cells (8 of 773) responded to wirelike objects presented from any viewpoint, thereby showing view-invariant response characteristics, superficially similar to the output unit of the model of figure 12. An example of such a neuron is shown in figure 15. The upper plot shows the monkey's hit rate and the middle plot the neuron's average spike rate. The cell fires with a rate of about 40 Hertz for all target's views. The lower plot shows the responses of the same cell to 120 distractors. With four exceptions, activity was

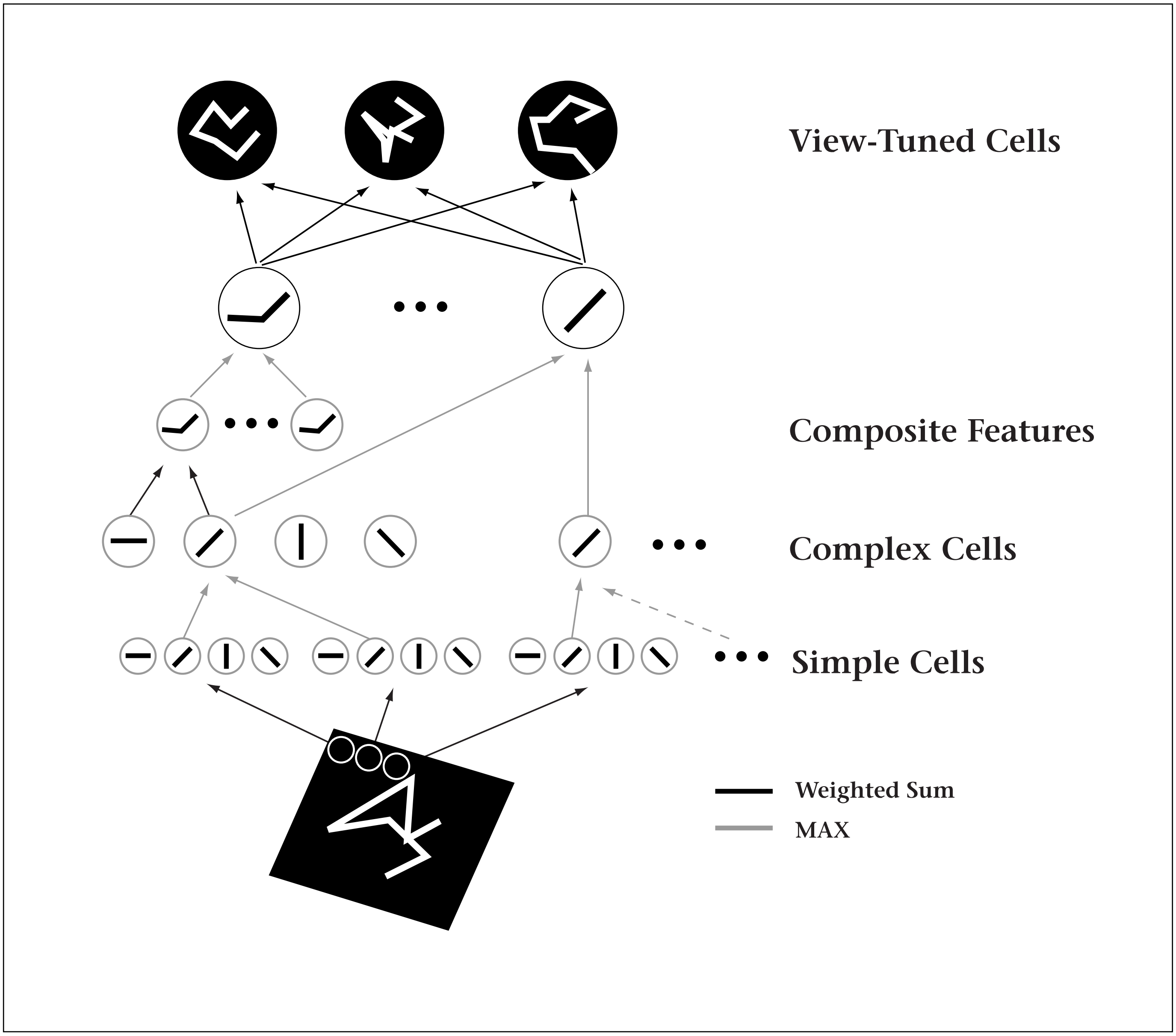


Figure 16. Model to Explain Receptive Field Properties of the View-Tuned Units of Figure 12 Found in Experiments (Riesenhuber and Poggio 1999).

uniformly low for all distractor objects presented. In all cases, even the best response to a distractor, however, remains about one-half the worst response to a target view. This neuron seems to behave as the output of the model of figure 12. Of the 773 (9 percent) analyzed cells, 71 showed view-selective responses similar to those illustrated in figures 12 and 13. In their majority, the rest of the neurons were visually active when plotted with other simple or complex stimuli, including faces.

The main finding of this study is that there are neurons in the IT cortex with properties intriguingly similar to the cartoon model of figure 12, which is itself supported by psychophysical experiments in humans and primates. Several neurons showed a remarkable selectivity for specific views of a computer-rendered object that the monkey had learned to recognize. A much smaller number of neurons were object specific but view invariant, as expected in a network in which “complex”-like view-invariant cells are fed by view-centered “simple”-like units. Furthermore, we

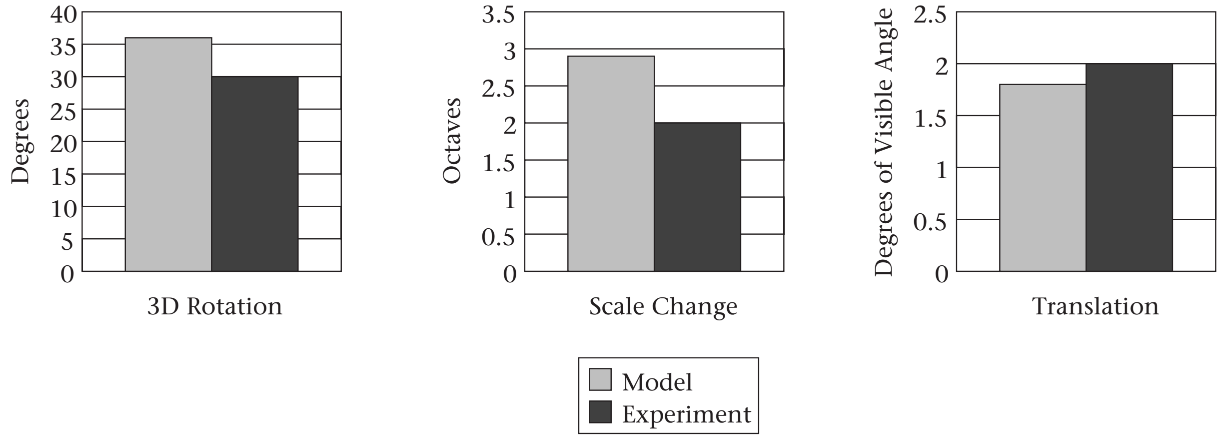


Figure 17. Comparison of Theoretical Model and Experimental Data.

believe that our results reflect experience-dependent plasticity in IT neurons and quite possibly also much earlier in the visual pathway. First, the neurons we found responded selectively to novel visual objects that the monkey had learned to recognize during the training. None of these objects had any prior meaning to the animal, and none of them resembled anything familiar in the monkey's environment. In addition, no selective responses were ever encountered for views that the animal systematically failed to recognize. Thus, it seems that neurons in this area can develop a complex selectivity as a result of training in the recognition of specific objects. Notice that view tuning was observed only for those views that the monkey could recognize.

A back-of-the-envelope extrapolation of the available data suggests an estimate of the number of cells whose tuning was determined by the training. In the region of IT from which recordings were made, which contains around 10 million neurons, we estimate that for each of the about 12 objects that the monkeys had learned to recognize, there were, at the time of the recordings, a few hundred view-tuned cells and on the order of 40 or so view-invariant cells.

A New Model

Models like the one of figure 12 leave open the issue of the mechanisms and circuitry underlying the properties of the view-tuned cells, from their view tuning to their invariance to image-based transformations such as scaling and translation. In fact, the invariance of the view-tuned neurons to image-plane transformation and to changes in illumination has been tested experimentally by Logothetis, Pauls, and Poggio (1995), who report an average rotation invariance over 30 degrees, translation invariance over 2 degrees, and size invariance to 1 octave around the training view.

These recent data put in sharp focus and in quantitative terms the question of the circuitry underlying the properties of the view-tuned cells. The key problem is to explain in terms of biologically plausible mechanisms their viewpoint invariance obtained from just one object view, which arises from a combination of selectivity to a specific object and tolerance to viewpoint changes.

Riesenhuber and Poggio (1998) described a model that conforms to the main anatomical and physiological constraints, reproduces all the data obtained by Logothetis et al., and makes several predictions for experiments on a subpopulation of IT cells. A key component of the model is a cortical mechanism that can be used to either provide the sum of several afferents to a cell or enable only the strongest one. The model explains the receptive field properties found in the experiment based on a simple hierarchical feed-forward model. The structure of the model reflects the idea that invariance and specificity must be built up through separate mechanisms. Figure 16 shows connections to invariance units with light arrows and to specificity units with dark arrows.

This new model is an expansion of the previous model to include nonlinear maximum (MAX) operation (similar to nearest-neighbor classification) to allow a high degree of invariance. This new model in simulations shows agreement with several physiological experiments from different labs. In particular, figure 17 shows the predictions of the model in comparison with experimental data.

Conclusions

The diagram in figure 1 shows our research process as a continuous loop. Because of the linearity of the print medium, we have mainly been able to show how theory has inspired applications and how their success has influenced research in neuroscience.

However, the flows of ideas also go the other way. The results shown for pedestrian detection of the influence of the number of features on the final classification results beg a very important theoretical question: "What is the optimal feature set, and how can we find it?" Furthermore, applications seem to beat the theoretical upper bounds by quite a large margin, which is especially true when one considers that the data-independent bounds derived for structural risk minimization should be overly optimistic because, as actually implemented, support-vector machines require data-dependent bounds (which should be worse). Thus, the applications pose another theoretical question: "Can we find good data-dependent bounds for machine-learning algorithms?"

The neuroscience work also raises theoretical and application questions. For example, the model depicted in figure 16 is inspired by invariances found in neurons; so, the next step is to try to use such results to direct new theories that might explain why such a model is a good one and use similar banks of filters as feature detectors in applications.

The supervised learning paradigm outlined here can be applied to other domains beyond the area of vision. For example, over the years, we have applied it to computer graphics. By analogy to the view-based paradigm for computer vision, we were led to the paradigm of image-based rendering that is just now becoming an important research direction in the graphics community (Ezzat and Poggio 1998; Beymer and Poggio 1996; Librande 1992). Other applications of our learning techniques have been in the domain of time series and finance (see, for example, Hutchinson, Lo, and Poggio [1994]) control, and search engines.

Despite a number of interesting and useful applications, it is clear that the problem of building machines that learn from experience and the problem of understanding how our brain learns are still wide open. Most of the really challenging questions are unsolved. There are still gaps between theory and applications and between machine learning and biological learning. Such comparisons raise a number of interesting questions, including the following:

Why is there a large difference between the number of examples a machine-learning algorithm needs (usually thousands) and the number of examples the human brain requires (just a few)?

What is the best way of naturally incorporating unlabeled examples into the supervised learning framework?

Can supervised learning methods be used to attack or solve other types of learning problem such as reinforcement learning and unsupervised learning?

To what extent can supervised learning explain the adaptive systems of the brain?

We hope that the work we described represents a few small steps in the right direction, in addition to providing a lot of fun for the mathematicians, the engineers, and the neuroscientists who are involved.

References

Beymer, D. J. 1993. Face Recognition under Varying Poses. Memo, 1461, Artificial Intelligence Laboratory, Massachusetts Institute of Technology.

Beymer, D., and Poggio, T. 1996. Image Representations for Visual Learning. *Science* 272(5270): 1905–1909.

Breiman, L. 1993. Hinging Hyperplanes for Regression, Classification, and Function Approximation. *IEEE Transactions on Information Theory* 39(3): 999–1013.

Bricolo, E.; Poggio, T.; and Logothetis, N. 1997. 3D Object Recognition: A Model of View-Tuned Neuron. In *Advances in Neural Information Processing Systems 3,* eds. M. I. Jordan and T. Petsche, 41–47. Cambridge, Mass.: MIT Press.

Broomhead, D., and Lowe, D. 1988. Multivariable Functional Interpolation and Adaptive Networks. *Complex Systems* 2:321–355.

Brunelli, R., and Poggio, T. 1991. HyberBF Networks for Real Object Recognition. In Proceedings of the Twelfth International Joint Conference on Artificial Intelligence, 1278–1284. Menlo Park, Calif.: International Joint Conferences on Artificial Intelligence.

Bülthoff, H. H., and Edelman, S. 1992. Psychophysical Support for a 2-D View Interpolation Theory of Object Recognition. In Proceedings of the National Academy of Science 89, 60–64. Washington, D.C.: National Academy of Science.

Edelman, S., and Büthoff, H. H. 1992. Orientation Dependence in the Recognition of Familiar and Novel Views of 3D Objects. *Vision Research* 32(2): 2385–2400.

Evgeniou, T.; Pontil, M.; and Poggio, T. 1999. A Review of a Unified Framework for Regularization Networks and Support Vector Machines. AI Memo, 1654, Artificial Intelligence Lab, Massachusetts Institute of Technology.

Ezzat, T., and Poggio, T. 1998. MIKETALK: A Talking Facial Display Based on Morphing Visemes. In *Proceedings of the Computer Animation Conference,* 96–102. San Francisco, Calif.: Morgan Kaufmann.

Friedman, J., and Stuetzle, W. 1981. Projection Pursuit Regression. *Journal of the American Statistical Association* 76(376): 817–823.

Girosi, F. 1998. An Equivalence between Sparse Approximation and Support Vector Machines. *Neural Computation* 10(6): 1455–1480.

Girosi, F.; Jones, M.; and Poggio, M. 1995. Regularization Theory and Neural Networks Architectures. *Neural Computation* 7(2): 219–269.

Girosi, F.; Jones, M.; and Poggio, T. 1993. Priors, Stabilizers, and Basis Functions: From Regularization to Radial, Tensor, and Additive Splines. Memo, 1430, Artificial Intelligence Laboratory, Massachusetts Institute of Technology.

Hutchinson, J.; Lo, A.; and Poggio, T. 1994. A Nonparametric Approach to Pricing and Hedging Derivative Securities via Learning Networks. *The Journal of Finance* 49(3): 851–889.

Librande, S. 1992. Example-Based Character Drawing. Master's thesis, Media Arts and Science Section, School of Architecture and Planning, Massachusetts Institute of Technology.

Logothetis, N., and Pauls, J. 1995. Personal communication, May, Center for Biological and Computational Learning, Massachusetts Institute of Technology.

Logothetis, N.; Pauls, J.; and Poggio, T. 1995. Shape Representation in the Inferior Temporal Cortex of Monkeys. *Current Biology* 5(5): 552–563.

Micchelli, C. A. 1986. Interpolation of Scattered Data: Distance Matrices and Conditionally Positive Definite Functions. *Constructive Approximation* 2(1): 11–22.

Mitchell, T. 1997. *Machine Learning*. Boston: McGraw-Hill.

Oren, M.; Papageorgiou, C.; Sinha, P.; Osuna, E.; and Poggio, T. 1997. Pedestrian Detection Using Wavelet Templates. Paper presented at the Computer Vision and Pattern Recognition Conference, June, Puerto Rico.

Osuna, E.; Freund, R.; and Girosi, F. 1997. Training Support Vector Machines: An Application to Face Detection. Paper presented at the Computer Vision and Pattern Recognition Conference, June, Puerto Rico.

Papageorgiou, C. 1997. Object and Pattern Detection in Video Sequences. Master's thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology.

Papageorgiou, C.; Evgeniou, T.; and Poggio, T. 1998. A Trainable Pedestrian Detection System. Paper presented at the Intelligent Vehicles Conference, October, Stuttgart, Germany.

Papageorgiou, C.; Oren, M.; and Poggio, T. 1998. A General Framework for Object Detection. Paper presented at the International Conference on Computer Vision, January, Bombay, India.

Poggio, T. 1990. 3D Object Recognition: On a Result by Basri and Ullman. Technical Report, 9005–03, IRST, Povo, Italy.

Poggio, T., and Edelman, S. 1990. A Network That Learns to Recognize 3D Objects. *Nature* 343(6255): 263–266.

Poggio, T., and Girosi, F. 1990a. Networks for Approximation and Learning. *Proceedings of the IEEE* 78(9): 1481–1497.

Poggio, T., and Girosi, F. 1990b. Regularization Algorithms for Learning That Are Equivalent to Multilayer Networks. *Science* 247:978–982.

Poggio, T., and Girosi, F. 1989. A Theory of Networks for Approximation and Learning. Memo, 1140, Artificial Intelligence Laboratory, Massachusetts Institute of Technology.

Powell, M. J. D. 1987. Radial Basis Functions for Multivariable Interpolation: A Review. In *Algorithms for Approximation,* eds. J. C. Mason and M. G. Cox, 113–167. Oxford: Clarendon Press.

Riesenhuber, M., and Poggio, T. 1999. Personal communication, January, Center for Biological and Computational Learning, Massachusetts Institute of Technology.

Riesenhuber, M., and Poggio, T. 1998. Modeling Invariances in Inferotemporal Cell Tuning. Memo, 1629, Artificial Intelligence Laboratory, Massachusetts Institute of Technology.

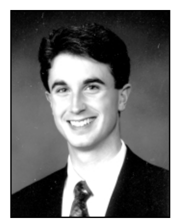
Romano, R. 1993. Real-Time Face Verification. Master's thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology.

Vapnik, V. 1995. *The Nature of Statistical Learning Theory.* New York: Springer-Verlag.

Vetter, T., and Poggio, T. 1992. Personal communication, April, Center for Biological and Computational Learning, Massachusetts Institute of Technology.

Wahba, G. 1990. Splines Models for Observational Data. In *Series in Applied Mathematics, Volume 59.* Philadelphia, Pa.: Society of Industrial and Applied Mathematics.

Tomaso A. Poggio is the Uncas and Helen Whitaker Professor in the Department of Brain and Cognitive Sciences at the Massachusetts Institute of Technology (MIT) and a member of the Artificial Intelligence Laboratory. He is doing research in computational learning and vision at the MIT Center for Biological and Computational Learning, where he is codirector. He is the author of more than 200 papers in areas ranging from psychophysics and biophysics to information processing in man and machine, AI, machine vision, and learning. His main research activity at present is learning from the perspective of statistical learning theory, engineering applications, and neuroscience. Poggio received his doctorate in theoretical physics from the University of Genoa in 1970 and had a tenured research position at the Max Planck Institute from 1971 to 1981 when he became a professor at MIT. He has received a number of distinguished international awards in the scientific community, is on the editorial board of a number of interdisciplinary journals, is a fellow of the American Association for Artificial Intelligence and the American Academy of Arts and Sciences, and is an honorary associate of the Neuroscience Research Program at Rockefeller University.



Christian R. Shelton is a Ph.D. candidate in the Department of Electrical Engineering and Computer Science at the Massachusetts Institute of Technology (MIT) and a member of the Artificial Intelligence Laboratory. He received his B.S. in computer science from Stanford University (1996) and his S.M. from MIT (1998). His research interests include surface correspondence and reinforcement learning.

Insert AI Communications Advertisement