BRAIN-STRUCTURED CONNECTIONIST NETWORKS THAT PERCEIVE AND LEARN

Vasant Honavar and Leonard Uhr
Computer Sciences Department
University of Wisconsin-Madison

Abstract

This paper specifies the main features of Brain-like, Neuronal, and Connectionist models; argues for the need for, and usefulness of, appropriate successively larger brain-like structures; and examines parallel-hierarchical *Recognition Cone* models of perception from this perspective, as examples of such structures. The anatomy, physiology, behavior, and development of the visual system are briefly summarized to motivate the architecture of brain-structured networks for perceptual recognition. Results are presented from simulations of carefully pre-designed Recognition Cone structures that perceive objects (e.g., houses) in digitized photographs. A framework for perceptual learning is introduced, including mechanisms for *generation-discovery* (feedback-guided growth of new links and nodes, subject to brain-like constraints (e.g., local receptive fields, global convergence-divergence). The information processing *transforms* discovered through generation are fine-tuned by feedback-guided reweighting of links. Some preliminary results are presented of brain-structured networks that learn to recognize simple objects (e.g., letters of the alphabet, cups, apples, bananas) through feedback-guided generation and reweighting. These show large improvements over networks that either lack brain-like structure or/and learn by reweighting of links alone.

Introduction

It is now widely recognized that massively parallel hardware/software structures are needed to perceive and understand the constantly changing environment. The Recognition Cones model of perception (Uhr, 1972; Honavar, 1987; Uhr, 1987) examined in this paper is suggested by the brain, and also the physics of objects and images and the logic of decomposing complex functions (e.g., *recognize house*) into cascades and compounds of simpler ones.

Connectionist networks are general-purpose computing structures (McCulloch, 1943), in the sense that there exist such (sufficiently large) networks that can compute any function computable by Turing machines, finite state automata, or a system of Post Productions. But the problem of developing the necessary, sufficiently powerful, efficient and robust network structures for perceptual recognition tasks remains, just as it does no matter how we try to embody intelligent processes.

Human beings recognize complex objects in less than a second, i.e., at least 10^3 times faster (and far better) than today's computers. Yet the basic computing unit in the brain, the neuron, is at least 10^5 times

This research was partially supported by grants from the Air Force Office of Scientific Research (AFOSR-89-0178), the National Science Foundation (CCR-8720060), and the University of Wisconsin Graduate School. This paper was published in Connection Science 1 139-159 (1989).

slower than a typical computer switch. The only way to achieve adequately fast perception with such slow components is to have large numbers of them cooperating in massively parallel networks (Uhr, 1972; Feldman, 1982). Both the logic of what is needed and observations of the brain's structures strongly suggest local and successively more global, rather than (as in many connectionist systems) completely linked or quasi-random structrures. This approach to machine perception differs significantly in philosophy from that in which *intelligent behavior* is realized in computer vision programs with little reference to underlying neural mechanisms. It also differs from *realistic brain models* (Sejnowski, 1988), wherein a neural net model of a specific portion of the brain is built and simulated, as faithfully as possible, based on the current neurophysiological data, and explanations for, and predictions of, neural phenomena that might take place in that part of the brain are advanced. As do most *simplifying brain models* (Sejnowski, 1988) that are today called *neuronal* or *connectionist*, it uses basic units reminiscent of simplified neurons to build the larger system. But it also employs successively larger brain-like structures, in order to give the system the power needed for real-world perception.

Neuronal, Connectionist, and Brain-Like Models Defined and Characterized

Our characterization of neuronal or connectionist models is somewhat more general than several other formulations (e.g., Rumelhart, 1986a; Feldman, 1982; Smolensky, 1988). In the discussion that follows, the terms *neuronal* and *connectionist* are used interchangeably.

A neuronal or connectionist network is a directed graph whose nodes compute functions on information passed to them via their input links, and send results via their output links.

[1] Each node has, associated with it, an activation level or a state variable.

[2] Each node computes one or more relatively simple local functions: its inputs are integrated or in some other way combined (e.g., by the application of a logic function such as, say, AND), these results might then be evaluated (e.g., against a threshold or a sigmoid function); it then outputs accordingly and updates its activation level.

[3] Each link has, associated with it, a transfer function.

[4] Links transmit signals (e.g., packets of bits, symbols, numbers, etc.) between nodes.

[5] Learning rules modify any of the following: processing functions of the nodes, transfer functions of the links, topology of the graph, and learning rules themselves.

[6] The topology of the graph, along with the functions that the individual nodes and links compute and the information input to them, determine the network's over-all behavior.

[7] The total graph may be (successively) decomposable into relatively regular sub-graphs (e.g., layers, windows, columns, trees). From this, the network's behavior, including output, coordination, control, adaptation, and learning follow and emerge.

This differs somewhat from other characterizations of connectionist models (e.g., Rumelhart, 1986a; Feldman, 1982; Smolensky, 1988) in which:

[1] Links usually have weights (typically real numbers) associated with them.

[2] Nodes usually output a simple sum over the inputs, typically after applying a threshold or a sigmoid function;

[3] Learning rules usually can modify only the weights associated with the links;

[4] Organization into higher-level structures, in many models, assumes a completely or randomly connected graph, an ad-hoc, problem-specific topology, or is left unspecified.

Brain-like Neuronal Models

Brain-Like neuronal models are suggested by the known anatomical, physiological, and behavioral data about the brain. They provide a basis for testing competing theories of perception, development, learning, and cognition; for suggesting neuroanatomical, neurophysiological, and behavioral experiments designed to fill our gaps in our understanding of these phenomena; and for building artificial systems exhibiting comparable perceptual and cognitive abilities. In the hierarchy of brain models, they occupy a place between realistic models and simplifying models (Sejnowski, 1988), capturing some aspects of both. Some basic criteria for brain-like neuronal models could be stated as follows:

[1] The nodes and links should (at least, to a first approximation and without gross violations) model neurons or functional units realizable with neuron-like units and connections between them.

[2] The topology of the graph, processing functions of the nodes, transfer functions, and learning rules should be plausible in terms of the known structure and the function of the brain.

[3] If the total system is decomposable into higher level structures (sub-graphs, columns, areas, etc.), such structures must be reasonably brain-like (to a first approximation and without gross violations, or at least not altogether implausible in terms of the physiology and anatomy of the brain).

The description of connectionist, neuronal, and brain-like models given here is broad enough to include a fairly large class of computational models of perception, ones that can differ significantly from each other in terms of structural and functional details. Thorough empirical and wherever feasible, theoretical analyses of computational abilities, robustness, efficiency, and explanatory powers of such models are prerequisites for our understanding of perceptual and cognitive processes.

The Primate Brain and Visual System: Structure, Function, and Development - An Overview

This section presents a very brief overview of the primate brain and the visual system, emphasizing object recognition. See (Kuffler, 1984; Van Essen, 1985; Uhr, 1986b; Zeki, 1988; Livingstone, 1988; DeYoe, 1988) for more details on vision, and (Honavar, 1989b) for a review of of perceptual development and learning and implications for modeling. The purpose of this overview is to motivate the architecture of Recognition Cones as an example of systems that structure networks of neuron-like units into successively larger brain-like modules.

The Retina and the Geniculate

The eye's lens focuses images of the scene on the 2-dimensional retinal array of rods (that sense small changes in position and intensity) and cones (that sense color). Each of these light-activated sensors sends excitatory signals straight back to some neurons, and inhibitory signals to surrounding neurons in the local neighborhood in the adjacent layer. There are several such layers consisting of bipolar and ganglion cells, with horizontal and amacrine cells providing rich lateral linkings to greater distances. Only ganglion and amacrine cells in the retina produce impulses of the kind typically found in the cortical neurons. The other cell types respond to illumination or darkness with relatively slow, graded potentials found in most neurons only in the synaptic regions. The rich system of retinal interconnections and its functional significance is only beginning to be understood (Sterling, 1986).

The orderly, layered organization suggests that the visual information processing is carried out in hierarchically arranged levels, going from one functionally related group of cells to the next. Also, the neurons converge and diverge extensively at any stage; i.e., each cell receives inputs from and sends signals to several other cells e.g., the human eye contains over 100 million primary receptors (rods and cones) but only about 1 million optic nerve fibers are sent from the lateral geniculate nucleus (LGN) to the cortex. The retinal ganglion cells appear to enhance differences and emphasize spatial as well as temporal gradients in the dynamic input image in the information conveyed to the cortex (Shapley, 1986). About a million ganglion cells carry signals from the retina via one layer of synapses in the lateral geniculate to the primary visual area of the cortex, retaining the retinotopic nature of the map, much like the original image.

In the monkey, the LGN has 6 layers of cells, 2 *parvocellular* layers and 4 *magnocellular* layers. Different functional properties are correlated with these layers: The parvocellular layers form part of a pathway that is believed to be primarily involved in the perception of form and magnocellular layers form part of the pathway that is primarily involved in perception of motion. The LGN cells, like retinal ganglion cells, respond best to spatial and temporal differences in illumination. Study of receptive field properties of LGN cells is far from complete. However, it seems clear that different functional properties are represented in different layers of the LGN e.g., cells in the parvocellular layers respond to different colors of stimulus and cells in magnocellular layers respond best to moving stimuli.

The Visual Cortex

The visual cortex is a thin (1-2mm in thickness), crumpled, sheet containing a complex structure of six layers of densely linked neurons (that form the grey matter), under which lies a mass of cortico-cortical axons that carry signals from one part of the cortex to another (that constitute the white matter). Several *areas* have been identified as involved primarily in visual information processing (e.g., V1, V2, V3, V4, V5). Each area contains its own representation of the visual field projected in an orderly manner. The significance of these different areas is that they provide for abstraction, enhancement, and integration of information from specific visual submodalities (e.g., color, motion, shape). From area to area there is much variation in the structural and functional properties of cells (Zeki, 1988), as well as the relative thickness of different layers. Characteristically, processes that connect cells in different layers within an area run for the most part, perpendicular to the surface of the cortex. In contrast, the majority of lateral processes are short. Lateral connections between areas are made by axons that run in bundles through the white matter (Kuffler, 1984).

In the monkey, the inputs from the two eyes remain segregated in V1 (like in the LGN), giving rise to the so called *ocular dominance columns*, columns of cells that respond to stimulation from one eye but not the other (Hubel, 1982). The majority of the projections from LGN to V1 end in layer IV of V1.

Another form of columnar organization found in the visual cortex is that of *orientation columns*. Cells responsive to edges at the same orientation are found grouped in columns running perpendicular to the surface of the cortex. It has been suggested that a basic functional unit of V1 is the roughly cuboidal aggregate of cells, the so called *hypercolumn*, that contains all the possible orientations (in steps of roughly 12^o) for a given receptive field area in each of the two eyes (Hubel, 1982). Cells sensitive to simple features such as oriented edges, colors, have been identified in V1. The orientation selective cells are anatomically seggregated from the color selective cells (Livingstone, 1988). There are also cells that respond optimally to specific combinations of simple features e.g., oriented edge separating patches of different colors.

It was originally thought that simple feature detector neurons are followed by ones that respond to more complex features in a more or less strict single hierarchical chain of areas, each carrying out a progressively higher level of analysis over the same image attributes as its predecessor. But it is now established that there are several serial pathways running in parallel, each of which is functionally specialized to a large extent (e.g., color, form, motion), suggesting more complex parallel-serial structures (Zeki, 1988; DeYoe, 1988; Uhr, 1986b), or *heterarchies* rather than a simple hierarchy.

Projections from area V1 lead to V2, and then to V4, and eventually to parietal and temporal cortices, constituting what is thought to be the primary pathway involved in object recognition. (Zeki, 1988; DeYoe, 1988). Most of these projections are bidirectional. Very few systematic studies to date have attempted to identify neurons that respond to successively more complex structures of features found in successively larger regions of the visual field; so at present the existence of such neurons is at best a reasonable conjecture. However, cells that respond in a very specific manner to extremely complex stimuli e.g., hands, faces, or even a specific face have been found by a number of researchers in the monkey temporal cortex (Perret, 1987). Each cell responds to several stimuli, but optimal response is obtained for only a small subset of those stimuli. Similarly, several cells respond to each stimulus. Thus, these cells are not necessarily *grandmother cells* each of which mysteriously somehow responds to a very complex stimulus and none else; they form complex networks of neurons that respond robustly and flexibly, yet specifically enough to a rich variety of objects found in the environment.

To summarize, at least 20 visual areas, and many nonvisual areas involved in perception, have been identified in the brain. Some of these handle different intrinsic scene characteristics such as motion, color and shape. Two major pathways, one processing color and shape, leading to recognition of objects, and the other handling spatial relations between objects and temporal changes (due to motion), are suggested by a large body of anatomical and physiological evidence. Similar evidence has been used to show how the 20 visual areas found to date (in macaque monkey) are wired together by about 40 major and 40 minor pathways. Note that this is far from the *complete connectivity* that is often assumed in some computational models, which would link each of the 20 areas to all 19 others, or worse, each neuron to every other neuron in the cortex; Nor is it a simple hierarchical tree of the sort implemented in many computer vision systems.

The Over-all Design of the Visual System

The brain contains on the order of 10^{11} neurons each of which may be connected to as many as 10^6 others (typically $3x10^4$ in the visual cortex). The extremely complex human visual system forms massively parallel, shallowly serial heterarchies, with functional organizations into larger structures of neurons interconnected by pathways that help to integrate diverse sources of information. Neurons usually (but by no means always) interact with near neighbors, and organize into successively larger structures (e.g., columns, hypercolumns, areas). The brain functions effectively in extremely noisy, distorted, rapidly changing environments. It can tolerate loss of neurons due to damage and aging, and change in thresholds and levels of firing due to drugs and deprivation, indicating large amounts of built-in redundancy and self-regulating mechanisms. It is able to gain knowledge of the environment through a life-long process of learning, suggesting considerable plasticity in its structure.

Perceptual Development and Learning in the Brain

There is strong evidence that the structure of the perceptual system is, at best, underspecified at birth. It appears that all the neurons as well as the layered, topographically mapped structure of the visual cortex are present at the time of birth. However, the connectivity among neurons undergoes significant changes (at least partly) as a function of experience throughout the life of the animal, although the exact mechanisms and locus of plasticity may vary with age. Connections are overproduced during early postnatal development. Connections are pruned, perhaps as a result of competition to somehow represent the input. Neuroanatomical and physiological evidence suggests that the plasticity of information processing structures involves both changes in existing connections (analogous to reweighting of links in a connectionist network) as well growth of new connections in response to environmental input practically throughout life (Greenough, 1988).

Development of certain visual processes (e.g., binocular fusion) relies heavily on the availability of certain kinds of stimuli (e.g., similar patterns to both eyes) during appropriate stages of development. Behavioral studies of infants offer evidence that at least suggests the possibility that the emergence of certain perceptual abilities (e.g., form discrimination) occurs only after the development of certain other requisite abilities (e.g., discrimination of line segments at different orientations). In this context, anatomical and physiological evidence for a phased development of the visual pathways from the retina to successively deeper cortical layers is tantalizing. Further, there is some evidence for the *gestalt principles of perceptual organization* (Hochberg, 1978) e.g., proximity (image features that are relatively close together tend to be grouped together) and similarity (image segments that have similar brightness, color, or texture tend to be grouped together) among very young infants. The initiation, maintenance and termination of plasticity are probably regulated by slowly diffusing neuromodulators, peptides, and hormones. All of this is a rich source of suggestions for the construction of artificial systems that learn from experience.

Brain-Structured, Parallel-Serial, Distributed, Heterarchical, Architecture of Recognition Cones

Basic Building Blocks of Recognition Cones

Conceptually it is useful to think of Recognition Cones' adaptive neuron-like elements as abstract processors that compute neuron-like functions (call them *fuzzy transforms*) over its inputs. A set of inputs gathers potentially relevant information, usually over a small compact window (figure 1) - e.g., a region large enough to extract a local feature like an edge, angle, or (at a higher level where abstracted image arrays form the inputs to the arrays of processing units) contours, enclosures, and other higher-level features.

Alternatively, each unit may be thought of as a simple *finite state automaton*; or, as a device embodying *IF [conditions] THEN [implieds]|[actions]* type *production rules.*

These differ from standard production rules in their parallel execution, and their use of weights and thresholds, and, in the case of Recognition Cones, of spatial interrelations derived from arrays containing images and their successive abstractions.

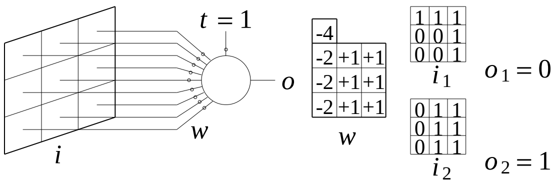

Figure 1: *A fuzzy transform over a* 3×3 *neighborhood:* w *is the mask of weights;* t *is a constant input that is multiplied by the threshold weight (-4 in this case);* i_1 *and* i_2 *are inputs in two* 3×3 *neighborhoods and* o_1 *and* o_2 *are the corresponding outputs.*

Architectural Features of Recognition Cones

Large numbers of basic units are organized into converging-diverging structures (converging logarithmically moving up; and also diverging because a unit can link to several others in the layer above). The connectivity between layers is predominantly retinotopic (but it effectively encodes translation-invariant features, because of the logarithmically graded resolution moving up). Each layer has a cluster of nodes at each location. Each node in layer *L* is linked to an n-tuple of nodes drawn from node clusters within a small window in layer *L-1*.

Within each layer, each unit is linked primarily to nearby units in a relatively small surrounding neighborhood, reflecting the stronger interaction between nearby points in the scene (as compared to points that are further apart). For simplicity, some regularity could be imposed on the size and shape (and to make implementations on today's computers feasible it typically is - e.g., each unit can link to its 4, 8 or 24 nearest neighbors in a square grid) of the neighborhoods. Or the connectivity patterns could model, albeit in a simplified manner, the structure of the retina, the lateral geniculate and the visual cortex (Uhr, 1986b). Several implementation alternatives are examined in detail in (Uhr, 1986a; Uhr, 1987). Thus, Recognition Cones make massively parallel use of *Pyramid* structures (Hanson, 1980; Uhr, 1983; Rosenfeld, 1983; Burt, 1984; Uhr, 1986a; Dyer, 1987; Levitan, 1987).

The input to the system is the image of the scene sensed by transducers (e.g., TV cameras) at the base layer of the Recognition Cones. The total system is made of several cone-like structures emerging from the retinal layer. There are several outputs from the system, typically, but not necessarily, from the higher levels (Figure 2 shows a schematic diagram of one such cone). Further, there may be a rich set of 2-way links, e.g., for feedback loops, between the different layers, including the output.

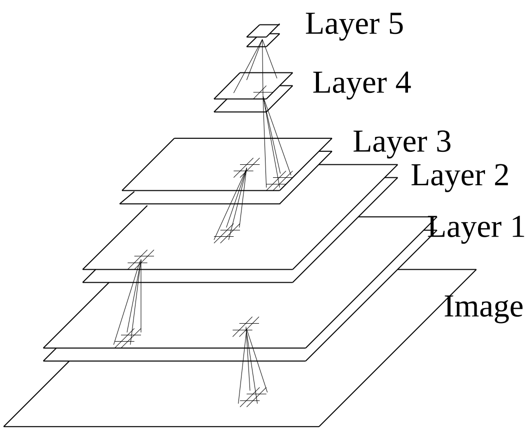

Figure 2: *Recognition cones: A Converging-diverging heterarchy of transforms. Each location in a layer has a cluster of nodes. Each node in a cluster computes a simple function over the outputs of the nodes in clusters within a small window in the layer below, to which it is connected.*

Designing Recognition Cones for Visual Perception

Recognition cones must be given a specific structure of transforms, for example: The image is input into the retinal layer at the base of the pyramid. It is processed there with local smoothing (noise suppression) transforms and then by local gradient detectors. The next layer then looks for a family of edges at different orientations, as well as color and textural features. The next layer combines oriented edges into corners, longer lines, curves, etc; colored regions into contrast-corrected larger regions; and so on.

This process of successive transformation and merging of information to detect more and more complex features (figure 3) continues, possibly all the way to the top, until enough information is gathered so that specific objects are sufficiently highly implied by the features detected. In addition, continuing feedback from higher to lower layers activates other processes at those layers. These serve to gather additional evidence to confirm the implied features, initiating a multi-level *relaxation* (Rosenfeld, 1976; Torras, 1989), i.e., parallel, iterative computations that achieve globally consistent interpretations through a cooperative interplay of several local processes.

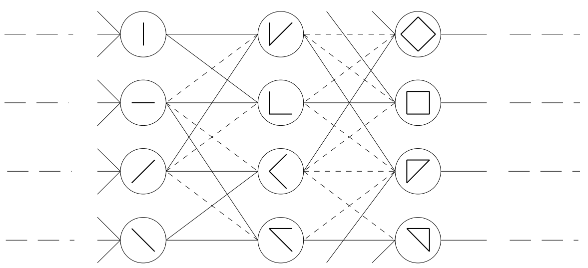

Figure 3: *Heterarchy of transforms detects increasingly complex features*

Performance of Pre-Designed Recognition Cones in Visual Perception

Recognition cones, when given a small set of transforms (such as edges, angles and curves) recognize a variety of simple objects (squares, circles, etc.). When given a large enough set of carefully chosen transforms distributed over 4-7 layers, such programs have demonstrated the capability to identify hand-printed letters (with gaps, small distortions and other forms of noise) as well as stylized hand-drawn sketches of place settings consisting of plates, spoons, knives and forks, and also some of the major structural features (e.g., doors, roofs, windows) of photographed houses (Uhr, 1979).

Simulations that combine data-driven, bottom-up processing where many feature-detecting transform are applied in parallel with model-driven, top-down processes which are activated when certain transforms respond to the image with sufficiently high weights (Li, 1987) recognize complex real-world objects such as windows, shutters, doors, houses, etc. from digitized (grey values range from 0 to 255), high resolution (512×512) TV images of outdoor scenes. The program was tested on three scenes, each containing a different house (two of the scenes were used by the programmer in determining the set of transforms to be provided to the program and the third was used to evaluate the generality of the transforms) and a fourth scene containing an office building (Figure 4 shows these scenes) with good results in identifying the building and its major structural components. Figure 5 shows some of the results: W1 through W12 correspond to the 12 windows in the office building; N4 and N5 correspond to 2 of the several regions in the scene that do not contain a window. *Bel(X)* is the output of transform *X*; *Bel(window)* and *Bel'(window)* correspond respectively, to evidence for a window before and after relaxation triggered by the model-driven, top-down processes (that use feedback loops between adjacent layers).

Thus, Recognition Cones, although they are highly parallel, and also neuronal and connectionist - albeit with additional more global brain-like structures - handle complex vision problems at least as well as do computer vision systems that rely on explicit serial model-matching. The latter are also much slower and appear to be rather brittle and difficult to extend to handle larger numbers of object-classes, each with much larger numbers of possible variant object-instances.

Learning

How can a system like Recognition Cones develop perceptual abilities by learning the necessary transforms as a function of experience, rather than having them built in?

Learning as Constrained Induction

Learning refers to the acquisition (Uhr, 1973; Michalski, 1983). Learning entails the building of usable models of the environment in which the learner-perceiver (whether natural or artificial) operates. Given a sufficiently rich environment, one that captures at least a significant portion of the great complexity and variety present in the real world, the number of possible inputs and the number of possible structures relating and combining them is enormous (If there are N inputs, each capable of taking V values, the number of possible structures is V^N). Only a small fraction of these associations are meaningful in modeling the environment. The perceptual learning system must be able to abstract the meaningful.

Knowledge of the environment is gained by a process of induction (Michalski, 1983; Holland, 1986) - that is, the setting up of hypotheses and then the accumulating of evidence to confirm or deny their validity.

Figure 4: *Digitized 512x512 pictures of buildings that were recognized by the multi-layered image and model-driven Recognition Cones program*

Possible window areas					
-	W1-6	W10	W11-12	N4	N5
$Bel(elong)$	0.50	0.50	0.50	0.30	0.50
$Bel(text)$	0.40	0.40	0.40	0.40	0.00
$Bel(left-bound)$	0.60	0.60	0.60	0.00	0.10
$Bel(right-bound)$	0.60	0.10	0.60	0.60	0.30
$Bel(window)$	0.45	0.38	0.45	0.34	0.20
$Bel(v-sibling)$	0.60	0.60	0.60	0.00	0.00
$Bel(h-sibling)$	0.60	0.60	0.60	0.00	0.60
$Bel'(window)$	0.49	0.46	0.49	0.13	0.20

Figure 5: *Results of identifying windows in the office building. W1 thro' W12 correspond to the 12 windows in the office building (W7 thro' W9 are not shown in the table); N4 and N5 correspond to 2 of the several regions in the scene that do not contain a window;* $Bel(window)$ *and* $Bel'(window)$ *correspond to the evidence for a window before and after relaxation triggered by the model-driven, top-down processes.*

Basic Neuronal Mechanisms for Learning

Learning in a neuronal system may involve modification of any of the following:

[1] The processing functions of the nodes (e.g., change in the threshold or output function),

[2] The weights (or transfer functions) of the links,

[3] The topology of the network (addition and deletion of links), and

[4] The learning rules themselves

Most of the work on learning in connectionist networks to date has concentrated on [2]. Several algorithms for changing weights associated with the links are available (Hinton, 1987a). A learning scheme for [3] that employs a mechanism for activity-dependent, feedback-guided *generation* of new links is described in (Honavar, 1987; Honavar, 1988).

The Learning of Useful Transforms through Generation and Reweighting

In a system like Recognition Cones, learning by induction can be viewed as the generation, tuning (by reweighting), and retention of a set of fuzzy transforms that are adequate for the perceptual tasks demanded of the system. The system may be initialized with (as though by evolution) a set of low-level transforms such as edge detectors, color detectors, etc. What follows is a general description of the learning mechanisms; a particular implementation is explained in detail in the next section.

Modification (Reweighting) of Existing Transforms

Feedback is used to weaken the links of the transforms that implied the *wrong* thing (and, optionally, to strengthen those that implied the *right* thing), by propagating the information from the unit or units that made the choice, usually moving from the output layer of the network, backward through the network, down to the input layer. This form of learning has been studied widely in connectionist systems, and several reweighting algorithms are available (Hinton, 1987a). We use one that is quite similar to the *error back-propagation* algorithm (Rumelhart, 1986b) (see below).

The Need for the Capability to Generate New Transforms

The input to the network represents a certain encoding of the environment. A single layer of neuron-like units computing fuzzy transforms over this encoding is combinatorially explosive and not always sufficient to produce the desired input-output mapping (Minsky, 1969). Internal representations that capture non-linear relationships between features in the input encoding must be created to overcome this problem. While it is true that one layer of hidden units between the input and output layers in theory suffices to enable the network to learn any desired input-output mappings, the fan-in and fan-out required of the units is arbitrarily large. That is, all input units must link directly each hidden unit that computes a global function. Large fan-in and fan-out imply longer links, a higher density of links, and hence a much greater, combinatorially explosive, cost/complexity. In the worst case, NV^N links are needed for N pixel input images, a quite impossible number even for toy images (e.g., 8x8) much less the 256x256 to 4,092x4,092 arrays needed for real-world images. To fend off this combinatorial explosion it is essential to restrict the links to relatively small *receptive fields*. When receptive field sizes are limited, multiple layers of hidden units become necessary to compute global functions and to represent the non-linear relationships between features in the input encoding. It is difficult, and in practice impossible except in trivial problems, to foresee the necessary connectivity, and the number and the depth of transformations needed for a particular task on which the network is to be trained; this is especially true when, in dynamic real-world environments, in which the system has to constantly adapt, as its tasks change over time.

Only when the network has *an adequate number of appropriately linked nodes* to start with does feedback-guided reweighting of links have a chance of eventually finding the right set of weights that would result in correct classification of patterns in the training set. But the only way to ensure that the network has the needed connectivity is to either build it that way, using a-priori knowledge, or to make some guess as to the necessary number of nodes N_E and the necessary topology G_E, trying to start with a network that is sufficiently large to subsume G_E. Given mechanisms to generate new links and recruit new nodes as needed, the network can start with an arbitrarily small number of nodes and links and can gradually grow, until it approximates N_E and G_E.

The Generation of New Transforms

The generation of a new transform is triggered by feedback, through the release of a *transform generation signal*, under certain conditions to be explained later. This signal is propagated back through the network very much like the error signal used in reweighting. At one or more layers, a subset of the units receiving the transform generation signal recruit one or more uncommitted units from the next-higher layer by growing a link to that unit. Growth of these links takes place without violating the topological constraints (such as those of local receptive fields, retinotopy, convergence, and layered organization) imposed by the architecture of Recognition Cones (see below). Thus, generation adds new transforms to the existing set.

The Discarding of Poor or Useless Transforms

Transforms get discarded either by a gradual lowering of weights as a consequence of negative feedback (when the weight on the link reaches a value close to zero, the link is broken) or by an abrupt breaking of some of the links, under certain conditions to be explained later. The discarding of transforms that are deemed poor or useless creates space in the system, by freeing up units that may then be used in the generation of new (and hopefully better) transforms to replace them.

Regulatory Mechanisms that Decide When to Generate and When to Discard Transforms

A network that both reweights and generates must somehow strike a reasonable balance between the two. Restricted to reweighting alone, the network may never achieve desired performance. But if transforms are generated too often, the essential process of tuning them by reweighting is aborted, and the result can be a large set of transforms, most of which are only rarely useful.

One possible mechanism to decide when to generate a new transform is suggested by the need to guide (and possibly goad) the network to develop the simplest possible structure to learn the required tasks (call this the *minimal complexity heuristic*).

One such regulatory mechanism, based on the minimal complexity heuristic (Honavar, 1987; Honavar, 1988) that decides when to add new transforms has been implemented in the simulation to be described below. This uses the number of transforms in the network as a measure of complexity. Thus, we seek networks with the smallest number of transforms adequate for classifying the patterns in the training set correctly. This suggests that the network should continue to reweight existing transforms so long as its performance is improving; and generate a new transform when performance ceases to improve with reweighting alone (and the network has not yet reached the target performance). Other regulatory mechanisms using different measures of complexity are under investigation.

Regulatory mechanisms are needed to decide when to discard a transform. Discarding is necessary if performance remains poor, or when space must be freed to generate new links and nodes.

Additional regulatory mechanisms are needed to determine where to place new generations (e.g., in which layer); and to decide which transforms to discard with minimal disturbance to others.

Regulatory mechanisms that guide the generation, modification, and retention of transforms require structures that maintain, update and transmit information (accumulated through local computations) concerning the performance of the network at the tasks being learned. History of the recent performance is used to determine what changes to make. All of this is done by the network itself (see below). a

The mechanisms of reinforcement of *good* transforms, and the generation of new transforms explained above, over a period of time result in the development and retention of useful transforms. On the other hand, transforms found not useful will fade away, finally being discarded, since they are negatively reinforced. Several questions remain. For example: What are good sets of regulatory mechanisms? Can networks exercise appropriate higher order control on the nature of the particular regulatory mechanisms used?

Simulation of Perceptual Learning through Generation and Reweighting of Transforms

This section briefly outlines the simulation of a brain-like network that generates as well as re-weights the transforms necessary for perception The way that generations build on prior generations (or pre-wired transforms) within the topological constraints of local receptive fields and layered, logarithmically converging structure, yields a Recognition Cone architecture.

Topological Constraints and Network Structure

The input layer (the retina) is a $d \times d$ square array of pixels (where $d = 2^m$; m = 2, 3, 4, ...). In most of the simulations to date, the input layer is a 32×32 array. Layer L contains 1/4th the number of node-clusters found in the adjacent layer $L-1$, giving a 2×2 logarithmic decrease in resolution as one moves up from the retina. Each node in a node cluster at layer L can only link to pairs of nodes drawn from 4 node clusters located directly below it in layer $L-1$. Thus, the mapping between layers is retinotopic. In the current implementation, layer 2 is an exception in that each node in layer 2 receives input from 9 nodes in the input layer. In some of our simulations, layer 2 contained 8 pre-wired edge detectors (that are extremely simplified versions of the oriented edge detectors found in the primary visual area (V1) of the living primate brains). It is not necessary to provide these pre-wired transforms since the learning mechanisms (as simulations demonstrate) are capable of discovering the edge detectors and any other transforms that may be useful.

At each layer, the nodes, with the exception of those that are already linked into the network as part of either pre-wired or learned transforms, form a pool of uncommitted units which get used in the generation of new transforms (see below).

Reweighting and Generation

Re-weighting of links is a function of the back-propagated error signal. Suppose a pattern class C_W is implied by the network with a weight W_W, and the pattern class indicated by the feedback, C_R is implied with a weight W_R, the amount of reweighting at the output layer is given by $(K \times (W_W - W_R))$ where K is a parameter related to the rate of learning. Our current implementation has K set equal to 0.25. This weight change is distributed equally among all the links firing into the node implying C_W. At internal nodes, the weight changes are computed in a similar fashion. Other variants of the error back-propagation, or even other reweighting schemes could be used.

In addition, the network occasionally generates a new transform, when it determines this to be appropriate - on the basis of information provided by regulatory substructures that monitor the network's performance on each pattern class on which it is being trained. The design of these substructures is motivated by the minimal complexity heuristic outlined earlier. The current implementation generates at most 2 new transforms if the performance improvement on any given pattern class is less than 1% over 10 consecutive training epochs (while the target performance has not yet been reached). This is accomplished entirely by simple networks of neuron-like units, through local computations performed incrementally following each training presentation. An implementation of such structures is described elsewhere (Honavar, 1988). Details of a particular implementation are unimportant, since a variety of local computations of this kind can be performed by appropriate microcircuits (e.g., counters, comparators) built from neuron-like units.

Generation proceeds as follows: In the 1st layer, a 3x3 sub-array is extracted from the raw input image (this is done only when feedback indicates an error was made, and the history of the recent past indicates that performance is levelling off rather than improving). These 9 links fire into a new node placed directly above it in the next layer.

The extraction is got from a *busy* part of the input image, one where the network judges there may be useful information. The present simple system insists that a gradient be present, but potentially more powerful mechanisms that enable the system to evaluate a certain region (e.g., a 3x3 window) of the input for its information content, and their possible connectionist network implementations are being investigated.

In layers other than the 1st, extraction adds links into a new node from 2 nodes that actively responded to the present (incorrectly identified) input image in the 2x2 window of node-clusters directly below it in the previous layer. For example, two oriented edge detector nodes may be linked by this mechanism to generate a transform that is responsive to an angle; and such generation is triggered when the existing transforms, say, edge detectors, by themselves have proved inadequate to achieve the desired performance, given the working of regulatory substructures that initiate generation described earlier.

Whenever a node is recruited, it is put into a node-cluster at that location, and also at every other location in that layer of the network. This seems logically complelling for a general purpose recognizer since a transform useful at some location in the visual field is likely to be useful at all locations, to enable translation-invariant recognition. All the links added to the network through generation get tuned through reweighting as a function of feedback. The current implementation does not include mechanisms for discarding poor transforms (except through a gradual lowering of weights).

Performance of Brain-Structured Networks That Learn by Generation and Reweighting at Perceptual Recognition of Simple Patterns

This section summarizes some preliminary simulation results.

Training and Test Data

Simple 2-dimensional patterns such as letters of the alphabet (T, D, E) and simple objects (apple, cup, banana) were used for training the networks. The training and test sets were obtained by randomly dividing the set of drawings of each pattern provided by 3 different volunteers into two subsets. The drawings were made using the *Xgremlin* graphics utility on a Digital VAXstation-3200, in a 24x24 subarray of a 32x32 grid (to allow for some translation). A sample subset of patterns used for training and testing is shown in figure 6. Figure 7 gives a summary of the pattern classes used in the runs, i.e., (T, D, E), (apple, banana, cup) and the combined set (T, D, E, apple, banana, cup).

A *run* consists of several *epochs* of training interspersed with epochs of testing, repeated until the desired accuracy of recognition (currently set to 100 percent) is attained or the performance clearly levels off, as indicated by the learning curve. An *epoch* of training (or testing) involves cycling through the entire training set (or test set) once, in some arbitrary order.

The figures 8A and 8B show the results of these runs on the pattern set (T, D, E). With no pre-wired edge detectors, the network attained 100% accuracy of recognition on the test set in 26 epochs of training; 14 new transforms were generated and they were replicated at each location in the corresponding layers. When 8 oriented edge detectors were provided at the first layer to start with, 100% accuracy of recognition was attained in about 8 epochs of training, and 6 new transforms generated in the process.

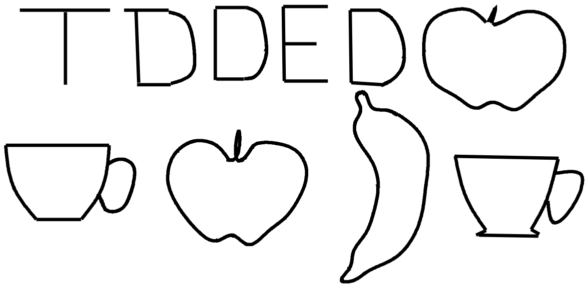

Figure 6: *Sample images used in the simulation of learning*

Pattern Set	# of classes	# of training instances per classs	# of test instances per class
T, D, E	3	4	3
apple, cup, banana	3	4	4
T, D, E, apple, cup, banana	6	4	3

Figure 7: *Summary of pattern sets used in the experiments: the pattern sets were obtained from instances provided by 3 volunteers. Training and test instances for each class were obtained by randomly partitioning the set of instances for a given class into two subsets, one for training, and the other for testing.*

The results with pattern sets (apple, cup, banana) were qualitatively similar in all the cases (the runs needed about 10% more epochs, and about 10% more links were generated). The runs were repeated with 6 pattern classes (T, D, E, apple, banana, cup) and the results were qualitatively similar, but there were more generations (about twice as many) at the higher layers, and about twice as many epochs of training were needed to reach 100% accuracy. Repeating the runs with a different randomly chosen translation (between 0 to 8 pixels from the center of the 32x32 grid) during each training and test presentation, as expected, had little effect on performance.

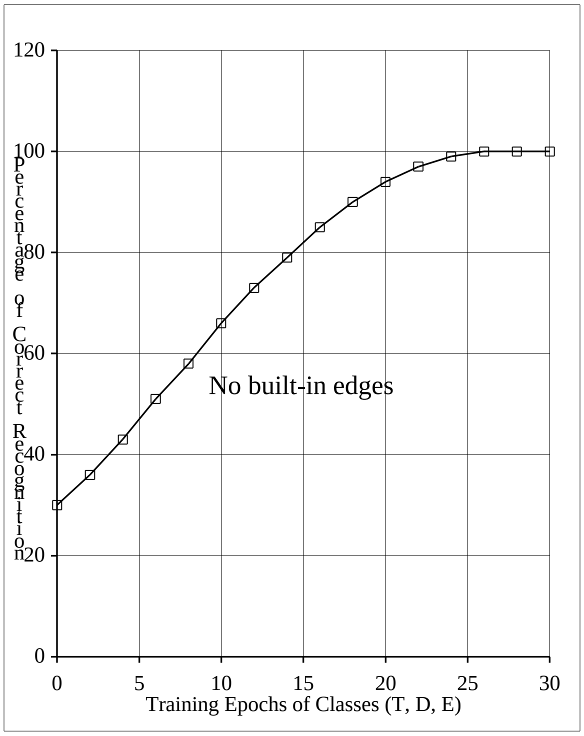

Figure 8A: *Performance on test set with pyramid convergence, generation, local receptive fields, without built-in edges*

The performance of networks that both generate and re-weight have been compared, using the same training and test data, with that of networks that learn by reweighting alone (that have the same structure but varying amounts of connectivity); with networks that lack locality of receptive fields; and with networks with varying numbers of hidden units with complete connectivity between layers (Honavar, 1989a). Although the results are preliminary in nature, they suggest that generation and local structure substantially improve learning, need fewer training epochs, and produce smaller networks.

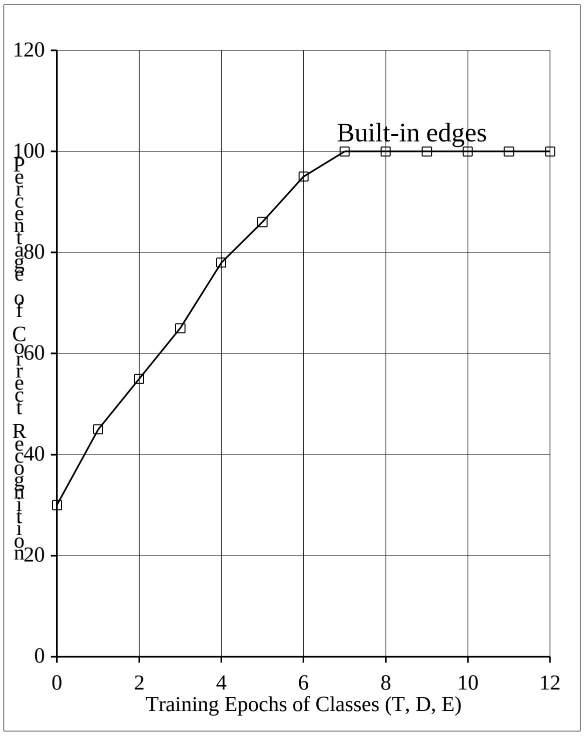

Figure 8B: *Performance on test set with pyramid convergence, generation, local receptive fields, with built-in edges*

Discussion

Distributed Processing, Memory and Control in Brain-Like Computing Structures

Recognition of a pattern is the result of utilizing knowledge distributed over many successively more complex transforms. Each transform processes a *micro-feature* in the input or an encoding or abstraction of the input. Such a micro-feature on its own conveys very little information about the pattern to be perceived. Thus processing and memory are distributed over a large number of neuron-like units.

Control is locally but collectively exerted, by individual units providing the inputs for other units. This is supplemented by the regulatory micro-circuits, e.g., those that decide when to generate new transforms. Some control is exerted by the structure of the system itself, since the connectivity determines the direction and order of information flow, and this in turn determines what transforms get applied. Thus control is distributed; the need for a *central controller*, analogous to that used in conventional von Neumann computers is eliminated.

To function effectively in the real world (as the human visual system clearly does), networks must be immune to noise and redundant to faults. This is accomplished by the fuzzy nature of the transforms, the use of windows and samplings, thresholds that accept a variety of different patterns of firings, using several transforms that look for similar features (e.g., different edge detectors all of whose results are combined). Different but closely related transforms that serve overlapping functions also provide the redundancy needed to ensure robustness and graceful degradation under random failures, as well as noise immunity.

Role of Brain-Like Structures in Learning

Retinotopic mapping and local receptive fields exploit spatio-temporal contiguity in the environment. This favors the discovery-learning of relations between sub-patterns that are imaged onto neighboring regions of the retina. For example, a single object's sub-parts (e.g., legs, seat, back-rest) are projected into neighboring regions, with all spatial relations intact. Our results suggest that constraints on network connectivity are important: Random connectivity is unlikely to work in most practical problems. Similar conclusions were reached in a study involving the training of a connectionist network to solve random-dot stereograms (Qian, 1988). Since each layer fires into the next, the learning of structure is a hierarchical, repeated operation. Generation ensures that successively more complex non-linear relations between features in the input encoding of patterns are discovered at higher layers, to be assessed by the new transforms that are added. For example, the lower layers might learn the associations between several vertical edges more or less aligned with each other and thus discover (learn) the concept of a long vertical line. At higher levels, intersecting horizontal and vertical line segments facilitate the learning of the more complex concept of a corner, and so on. Thus learning of simpler relations preceeds learning of more complex relations and successively more global relations are learned at successively higher layers. An examination of transforms actually generated confirms this.

Generation, Reweighting, and Discarding of Transforms

Intuition suggests that good system performance requires a proper match between the *entropy* of the source of external stimuli and the connectivity, both between the source and the system (Abu-Mostafa, 1988), as well as within the system itself. Since generation relies on the environmental stimuli to develop the connectivity of the system, the resulting network is likely to have a better match with the entropy of the environment, than a network with a fixed random subset of connections and learns only by reweighting links.

Networks that learn by reweighting alone run the risk of getting stuck in a local minimum (Rumelhart, 1986b) of the error surface. Generation provides a potential means of getting out (by modifying the error surface as a result of generated transforms).

Network structures that maintain, update, and transmit appropriate information about performance (e.g., a portion of the learning curve, used to trigger generation) offer several interesting mechanisms to influence learning. They can be used to alter learning strategies, rates of learning, thresholds of firing - all improving *plasticity*.

The extent of generalization in networks is sensitive to the number of hidden units links (Hinton, 1987b; Rumelhart, 1988). If too many, the network may generalize rather poorly; if too few, it may never learn. Generation favors the discovery of the smallest necessary number, hence (we conjecture) better generalization.

The simulations described here do not discard bad nodes or place any limit on the number of nodes generated. (Neither capability was needed for the test runs reported here.) But to handle larger sets of more complex patterns, the ability to discard is almost certainly necessary; to keep from clogging the network.

There are a number of promising improvements to be made, including the processes that make better generations, learn to improve upon these, evaluate the generations for their usefulness, discard poor generations, narrow and broaden the tolerance-threshold for matching, and generate sets of alternate possible transforms that are placed in competition with one another. There are a number of other issues to be investigated, including the development of good sub-nets that decide whether to further re-weight or to generate, the optimal number of nodes in a node-cluster, whether to make nodes within a cluster compete, and whether and how to combine generation with other reweighting rules e.g., some variant of the Hebbian rule (Hebb, 1949). Generation is also being used to learn speech, boolean as well as continuous functions, and simple arithmetic.

Conclusions

Connectionist networks built from simple neuron-like units arranged in brain-like topologies can be constructed to yield relatively good perceptual recognition of complex real-world objects (e.g., houses) in large (512x512) images.

The preliminary results presented in this paper suggest the possibility of discovering such networks, by realizing significantly more powerful and potentially more practical learning than that given by reweighting alone, through a combination of:

[1] Different learning mechanisms: generation, reweighting, and (when necessary) discarding of transforms,

[2] Regulatory mechanisms that alter network plasticity in a controlled fashion, choose between different learning strategies e.g., minimal complexity heuristics, and

[3] Brain-like constraints on the network topology e.g., local receptive fields, retinotopy, layered converging-diverging heterarchy.

Extensive and systematic evaluations of networks incorporating one or more of these aspects for perceptual learning of pattern sets of varying complexity are needed in order to judge how they perform individually as well as collectively, toward the dual goals of understanding information processing in the brain and of designing artificial systems of comparable perceptual and cognitive abilities.

References

Abu-Mostafa, Y., ‘‘Connectivity and Entropy,’’ *Advances in Neural Information Processing systems*, Morgan Kaufmann, San Mateo, California, 1988.

Barto, A. G. and Anandan, P., ‘‘Pattern recognizing stochastic learning automata,’’ *IEEE Trans. Systems, Man and Cybernetics*, vol. 15, pp. 360-375, 1985.

DeYoe, E. A. and Van Essen, D. C., ‘‘Concurrent processing streams in monkey visual cortex,’’ *Trends in Neuroscience*, vol. 11-5, pp. 219-226, May 1988.

Dyer, C. R., ‘‘Multiscale image understanding,’’ in *Parallel Computer Vision*, ed. L. Uhr, Academic Press, New York, 1987.

Feldman, J. A. and Ballard, D. H., ‘‘Connectionist models and their properties,’’ *Cognitive Science*, vol. 6, pp. 205-264, 1982.

Greenough, W. T. and Bailey, C. H., ‘‘The anatomy of a memory: convergence of results across a diversity of tests,’’ *Trends in Neuroscience*, vol. 11, pp. 142-147, April 1988.

Hebb, D. O., *The Organization of Behavior,* Wiley, New York, 1949.

Hinton, G. E., ‘‘Connectionist learning procedures,’’ *Technical Report CMU-CS-87-115*, Computer Science Department, Carnegie Mellon University, Pittsburgh, Pennsylvania, 1987a.

Hinton, G. E., ‘‘Learning translation-invariant recognition in a massively parallel network,’’ in *PARLE: Parallel Architectures and Languages, Europe. Lecture Notes in Computer Science*, ed. G. Goos and J. Hartmanis, Springer-Verlag , Berlin , 1987b.

Hochberg, J., *Perception*, Prentice Hall, Englewood Cliffs, NJ, 1978.

Holland, J. H., Holyoak, K. J., Nisbett, R. E., and Thagard, P. R., *Induction: Processes of Inference, Learning, and Discovery,* The MIT Press, Cambridge, Massachusetts, 1986.

Honavar, V. and Uhr, L., ‘‘Recognition Cones: A neuronal architecture for perception and learning,’’ *Computer Sciences Technical Report #717,* University of Wisconsin-Madison, Madison, WI, 1987.

Honavar, V. and Uhr, L., ‘‘A network of neuron-like units that learns to perceive by generation as well as reweighting of its links,’’ in *Proceedings of the 1988 Connectionist Models Summer School*, ed. G. E. Hinton, T. J. Sejnowski and D. S. Touretzky, Morgan Kaufmann, San Mateo, CA, 1988.

Honavar, V. and Uhr, L., ‘‘Generation, Local Receptive Fields, and Global Convergence Improve Perceptual Learning in Connectionist Networks,’’ in *Proceedings of the 1989 International Joint Conference on Artificial Intelligence,* Morgan Kaufmann, San Mateo, CA, 1989a .

Honavar, V., ‘‘Perceptual development and learning: From behavioral, neurophysiological, and morphological evidence to computational models,’’ *Computer Sciences Technical Report #818,* University of Wisconsin-Madison, Madison, WI, 1989b.

Hubel, D. H., ‘‘Explorations of the primary visual cortex, 1955-1978,’’ *Nature* , vol. 299, pp. 515-524, 1982.

Kuffler, S. W., Nicholls, J. G., and Martin, A. R., *From Neuron To Brain,* Sinauer Associates Inc., Sunderland, Massachusetts, 1984.

Li, Z. N. and Uhr, L., ‘‘Pyramid vision using key features to integrate image-driven bottom-up and model-driven top-down processes,’’ *Systems, Man and Cybernetics*, vol. 17, pp. 250-263 , 1987.

Livingstone, M. and Hubel, D., ‘‘Segregation of form, color, movement, and depth: anatomy, physiology, and perception,’’ *Science*, vol. 240, pp. 740-749, 1988.

Marr, D., *Vision,* W. H. Freeman and Co., New York, NY, 1982.

Michalski, R. S., Carbonell, J. G., and Mitchell, T. M. (eds.), *Machine Learning - An Artificial Intelligence Approach, vol. 1,* Tioga , Palo Alto, California, 1983.

Minsky, M. and Papert, S., *Perceptrons: An Introduction to Computational Geometry,* The MIT Press, Cambridge, Massachusetts, 1969.

Perret, D. I., Mistlin, A. J., and Chitty, A. J., ‘‘Visual neurones responsive to faces,’’ *Trends in Neuroscience*, vol. 10-9, September 1987.

Rosenfeld, A., Hummel, R. A., and Zucker, S. W., ‘‘Scene labeling by relaxation operations,’’ *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 6, pp. 430-440, 1976.

Rosenfeld, A., ‘‘Pyramids: multiresolution image analysis,’’ *Proceedings of the Third Scandinavian Conference on Image Analysis,* pp. 23-28, 1983.

Rumelhart, D. E., Hinton, G. E., and McClelland, J. L., ‘‘A general framework for parallel distributed processing,’’ in *Parallel Distributed Processing vol. 1: Explorations into the Microstructure of Cognition*, The MIT Press, Cambridge, Massachusetts, 1986a.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J., ‘‘Learning internal representations by error propagation,’’ in *Parallel Distributed Processing vol. 1: Foundations*, The MIT Press, Cambridge, Massachusetts, 1986b.

Rumelhart, D. E., ‘‘Parallel Distributed Processing,’’ *Plenary Lecture given at the IEEE International Conference on Neural Networks,* San Diego, 1988.

Sejnowski, T. J., Koch, C., and Churchland, P. S., ‘‘Computational neuroscience,’’ *Science*, vol. 240, pp. 1299-1305, 1988.

Shapley, R. and Perry, V. H., ‘‘Cat and monkey retinal ganglion cells,’’ *Trends in Neuroscience*, pp. 229-235, 1986.

Smolensky, P., ‘‘On the proper treatment of connectionism,’’ *Behavioral and Brain Sciences*, vol. 11, pp. 1-23, 1988.

Sterling, P., Freed, M., and Smith, R. G., ‘‘Microcircuitry and functional architecture of the cat retina,’’ *Trends in Neuroscience*, pp. 186-198, 1986.

Torras, C., ‘‘Relaxation and neural learning: Points of convergence and divergence,’’ *Journal of Parallel and Distributed Computing*, vol. (In Press), 1989.

Uhr, L., ‘‘Layered recognition cone networks that preprocess, classify, and describe,’’ *IEEE Transactions on Computers*, vol. 21, pp. 758-768, 1972.

Uhr, L., *Pattern Recognition, Learning and Thought*, Prentice-Hall, Englewood Cliffs, New Jersey, 1973.

Uhr, L. and Douglass, R., ‘‘A parallel-serial recognition cone system for perception,’’ *Pattern Recognition*, vol. 11, pp. 29-40, 1979.

Uhr, L., ‘‘Pyramid multi-computer structures, and augmented pyramids,’’ in *Computing Structures for Image Processing*, ed. M. J. B. Duff, Academic Press, London, 1983.

Uhr, L., ‘‘Multiple image and multi-modal augmented pyramid networks,’’ in *Intermediate Level Image Processing*, ed. M. J. B. Duff, Academic Press, London, 1986.

Uhr, L., ‘‘Toward a computational information processing model of object perception,’’ *Computer Sciences Technical Report #651*, University of Wisconsin-Madison, Madison, WI, 1986.

Uhr, L., ‘‘Highly parallel, hierarchical, recognition cone perceptual structures,’’ in *Parallel Computer Vision*, ed. L. Uhr, Academic Press, New York, 1987.

Van Essen, D. C., ‘‘Functional organization of the primate visual cortex,’’ in *Cerebral Cortex*, ed. A. Peters and E. G. Jones, vol. 3, Plenum, New York, NY, 1985.

Zeki, S. and Shipp, S., ‘‘The functional logic of cortical connections,’’ *Nature*, vol. 335, pp. 311-317, 1988.