BRAIN-STRUCTURED CONNECTIONIST NETWORKS THAT PERCEIVE AND LEARN

by

Vasant Honavar
and
Leonard Uhr

Computer Sciences Technical Report #843

April 1989

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, structuring networks of neuron-like units into successively larger brain-like modules; and examines *Recognition Cone* models of perception from this perspective, as examples of such structures. Neuroanatomical, neurophysiological, and behavioral data on the structure, function, and development of the visual system are briefly summarized to motivate the architecture of brain-structured networks for perceptual recognition. The structural and functional architecture of Recognition Cones, the flow of information and the parallel-distributed nature of processing and control in Recognition Cones are described. The results from the simulation of carefully designed Recognition Cone structures that perceive objects (e.g., houses) in digitized photographs are presented. A framework for perceptual learning, including mechanisms for *generation-discovery*, that involves feedback-guided growth of new links between neuron-like units as needed, within a dynamically emerging network topology, subject to brain-like constraints on the network connectivity (e.g., local receptive fields, global convergence-divergence, retinotopically mapped layered heterarchy) is introduced. The information processing *transforms* discovered through generation are fine-tuned by feedback-guided reweighting of links. A case is made for the need for generation and discarding of transforms in addition to reweighting of links in *Connectionist networks* for perceptual learning. Some preliminary results from the simulation 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 of transforms are presented. Experimental comparisons indicate that such networks can give large improvements over networks that either lack brain-like structure or/and learn by reweighting of links alone. The role of brain-like structures and generation in perceptual learning is examined. Some directions for future research are outlined.

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, which serves as a living existence-proof of achievable perceptual and cognitive capabilities, and as a source of potentially useful mechanisms for achieving its capabili-

---

This paper is a substantially revised and updated version of (Honavar, 1987) and has been submitted to *Connection Science* for publication. Support for this research was provided by the National Science Foundation and by the University of Wisconsin-Madison.

ties by a computer system. The underlying hypothesis is that the larger structures of the brain, how its billions of neurons are connected in complex parallel-serial networks of layered ensembles of neurons, is central to its function.

This is not to deny that 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 than today's computers. This is an estimate, since no existing vision program comes close to perceiving as well as the brain does. Yet, the basic computing unit in the brain, the neuron, is at least 10^5 times slower than a typical computer switch. It has been argued that the only way to achieve adequately fast perception with such slow components is to have large numbers of them cooperating in massively parallel networks performing the necessary computations (Feldman, 1982). Both an analysis of what is needed and observations of the way the brain structures ensembles of neurons strongly suggest local and successively more global brain-like architectures that organize the components, which, in many connectionist systems, are linked quasi-randomly. 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 structures that are suggested by the brain's larger 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 neuron-like 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 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 contrasts with most other characterizations of connectionist models (e.g., Rumelhart, 1986a; Feldman, 1982; Smolensky, 1988) as follows:

[1] In connectionist models, links transmit only weights (typically, real numbers);

[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 connectionist models, assumes a completely or randomly connected graph, an ad-hoc, problem-specific topology, or is left unspecified.

A Neuronal System and its External Environment

It is often convenient to treat a neuronal network as forming a closed system along with the *external environment*. The external environment provides some of the inputs to, and accepts some of the outputs from, the interface nodes in the connectionist network. This closed system may be partitioned into two sub-graphs: an *internal* subgraph (that is, the neuronal system with its input and output links with the external environment removed), and an *external* subgraph (that is, the external environment with its input and output links with the connectionist system removed). The input and output links between the internal and external sub-graphs are functionally similar to input and output links between the nodes of the internal sub-graph. The environment may be part of the real world, linked with the neuronal system via transducers like TV cameras or robot effectors, or a representation of some aspects of the real world, simulated either by a computer program or by human beings interacting via keyboards and monitors.

Specification of a Neuronal Network

In order to completely specify a particular neuronal system, one needs to define the topology of the graph, the processing functions of the nodes, the transfer functions associated with the links, the learning rules (which can, potentially, modify any of the above), and the external environment that, together with the neuronal system, forms a closed system. The behavior of the system results from the dynamics of the system, that is, from the interaction between the large number of units acting in parallel at each moment, over a period of time.

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, 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 primarily, object recognition. A comprehensive, critical treatment is beyond the scope of this paper. The interested reader is refered to (Kuffler, 1984; Van Essen, 1985; Uhr, 1986b; Zeki, 1988; Livingstone, 1988; DeYoe, 1988) for more details on the anatomy and physiology of the visual system; and (Honavar, 1989b) for a review of anatomical, physiological and behavioral correlates of perceptual development and learning and some of their implications for computational modeling. The original papers are far too numerous to mention here. The reviews cited contain extensive bibliographies that cover the relevant literature. The purpose of this overview is to motivate the architecture of Recognition Cones as examples of neuronal systems that attempt to 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 appears to be a roughly cuboidal aggregate of cells, the so called *hypercolumn*, in which all the possible orientations (in steps of roughly 12^o) for 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 seperating 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

Space does not permit a critical discussion of neuroanatomical, neurophysiological, and behavioral data on perceptual development and learning here. The interested reader is refered to (Honavar, 1989b) for details. There is strong evidence that the structure of the perceptual system is, at best, underspecified at birth. The full complement of 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 offers 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 seems well regulated in the brain, probably under the influence of 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

This section briefly describes the architecture of Recognition Cones (Uhr, 1972; Honavar, 1987; Uhr, 1987) and examines it in the context of the general characteristics of brain-like neuronal models outlined above. The structure described here corresponds to one that is carefully designed (or programmed) for the task of recognition of specific objects or that which emerges as the result of learning, as explained later. The emphasis is on visual perception, although some of the underlying principles appear to be of relevance to other sensory modalities.

Basic Building Blocks of Recognition Cones

Conceptually it is useful to think of the adaptive neuron-like unit as an abstract process that computes one or more probabilistic or fuzzy transforms over its inputs. Such a unit typically has a small set of inputs which gather potentially relevant information, usually over a small compact window (figure 2) 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, the adaptive neuron-like 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.

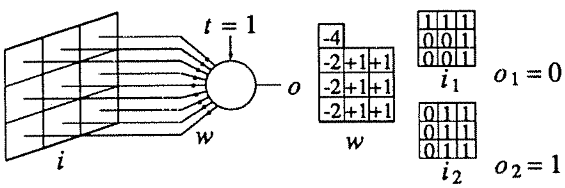

Figure 1: *A fuzzy transform over a* 3×3 *neighborhood:* w *is the mask of weights;* t *is the threshold;* i_1 *and* i_2 *are inputs in two* 3×3 *neighborhoods and* o_1 *and* o_2 *are the corresponding outputs.*

Basic Architectural Features of Recognition Cones

The basic building block of Recognition Cones is an adaptive neuron-like unit described above with a threshold or sigmoid output function which accepts inputs from other units via its input links, does some simple processing of the inputs, and sends out signals over its output links to the units into which it fires. Large numbers of such units are organized, into a layered heterarchy of converging-diverging structures (hence the name Recognition Cones). It is converging because the spatial resolution of the layers decreases logarithmically as one moves up; and also diverging because a unit can link to several others in the layer above. The connectivity between layers is predominantly retinotopic (but effectively encoding translation-invariant features, because of the logarithmically graded resolution as we move up in the heterarchy). Each layer of Recognition Cones has a number of node clusters (the number being determined by the spatial resolution of that layer). 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*. Thus, each node computes a fuzzy transform on the image or some abstractions of the image.

Within each layer, each unit is linked primarily to nearby units in a relatively small surrounding neighborhood. This reflects the property of the real world that nearby points in the scene are likely to influence each other more than those that are farther 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).

Recognition Cones are closely related to *Pyramids* and other hierarchical architectures and algorithms that have been fairly extensively studied for image processing (Uhr, 1983; Rosenfeld, 1983; Burt, 1984; Uhr, 1986a; Dyer, 1987). Often the processors in a Pyramid are more powerful than the basic functional unit of the Recognition Cones. Recognition cones can be thought of as multi-apex multi-pyramids emerging from a common base, possibly augmented with additional links between the pyramids and decision networks at the top.

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 additional links, e.g., for feedback loops, between the different layers, including the output. Computer simulations typically implement 2-way links, so that units can send out signals both upward and downward.

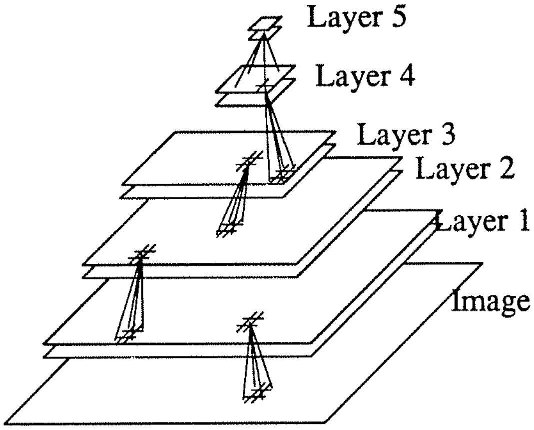

Figure 2: *Recognition cones: A Converging-diverging heterarchy of transforms; Each location in a layer has a cluster of nodes; Each node in a cluster computes a simple function over the outputs of the nodes in clusters within a small window in the layer below, to which it is connected.*

Information Flow In Recognition Cones

Visual information is input into an array of photoreceptors, which serves a function analogous to that of the retina. This layer samples the continuous input from the environment in space (because of limited, although fairly high, resolution of the retina) and time (just as the photoreceptors respond to light impinging on the retina in the human visual system - once a receptor fires, it cannot fire again for a duration of the refractory period). A whole layer of adaptive neuron-like units are excited and respond by firing in parallel (this is a simplification over the human visual system in which the firing of photoreceptors may not be completely synchronized, although at any given instant a large number of them may fire in parallel). In the process, they compute fuzzy transforms over their inputs of the kind described earlier. The firing of the units in one layer leads to the subsequent firings of units in adjacent layers, which, because of the layered heterarchical converging and diverging structure, naturally results in the computation of successively more complex, increasingly global, transformations (for, e.g., edges, long line segments, corners, contours, and so on).

The rich feedback loops that provide feedback between adjacent layers of neuronal units and, less often, between more distant layers, allow the whole process to take on the nature of a multi-level (involving several levels of abstraction) and multi-modal (involving several different kinds of information, such as shape, color, texture, motion) transformations and *relaxations*, i.e., parallel iterative computations that achieve globally consistent interpretations through a cooperative interplay of several local processes. (Rosenfeld, 1976; Torras, 1989). The outputs of the units at any layer are merged with the outputs of other units computing the same or different transforms at that layer, and converged into the next layer (just as the information the retinal cones and rods sense converges into layers of primary visual cortex after several retinal transformations). Similar processing takes place continually in parallel at every layer. However, it must not be forgotten that a certain serial depth of processing (due to the layered, heterarchical structure of the system) is essential: it is this serial depth that enables the computation of successively more complex, more global, transforms of the input.

Designing Recognition Cones for Visual Perception

This section outlines how Recognition Cones are used for visual perception - the recognition of objects in the environment, given the overall structures and processes described above. Results of simulation of the Recognition Cones model for perception of real-world objects are briefly presented. More detailed descriptions of the actual computer programs are found in (Uhr, 1979; Li, 1987).

Recognition cones are given a specific structure of transforms, as indicated by the following example (any cascaded structure of local processes can be used efficiently): 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 e.g., a high-resolution difference-of-Gaussian operator (Marr, 1982). The next layer then looks for a family of edges at several 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 processes at those layers (which may serve to gather additional evidence to confirm the implied features, initiating a relaxation process).

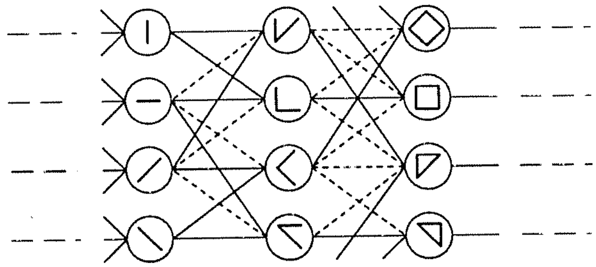

Figure 3: *Heterarchy of transforms detect increasingly complex features*

Performance of Pre-Designed Recognition Cones in Visual Perception

Recognition cone programs that apply sets of local fuzzy transforms, 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 of Recognition Cones which 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 (512x512) 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 (Li, 1987): 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.

Thus, Recognition Cones, although they are highly parallel, and also neuronal and largely connectionist - albeit with additional more global brain-like structures, have been shown able to handle complex vision problems at least as well as do computer vision systems that rely on explicit serial model-matching, and are, as a consequence, much slower and, in most cases, rather brittle and difficult to extend to a full-blown vision system, which must handle the much larger number of object-classes, each with a much larger number of possible variant object-instances.

Learning

The results presented in the previous section illustrate the usefulness of structuring assemblies of neuron-like units into higher level structures suggested by the brain for building systems capable of visual recognition. Our discussion has so far assumed that the transforms necessary for endowing the Recognition Cones with their perceptual abilities could somehow be put in place: The programs were given sets of carefully chosen transforms by the designers of the system. This is clearly an unreasonable expectation, given the complexity of the environment and

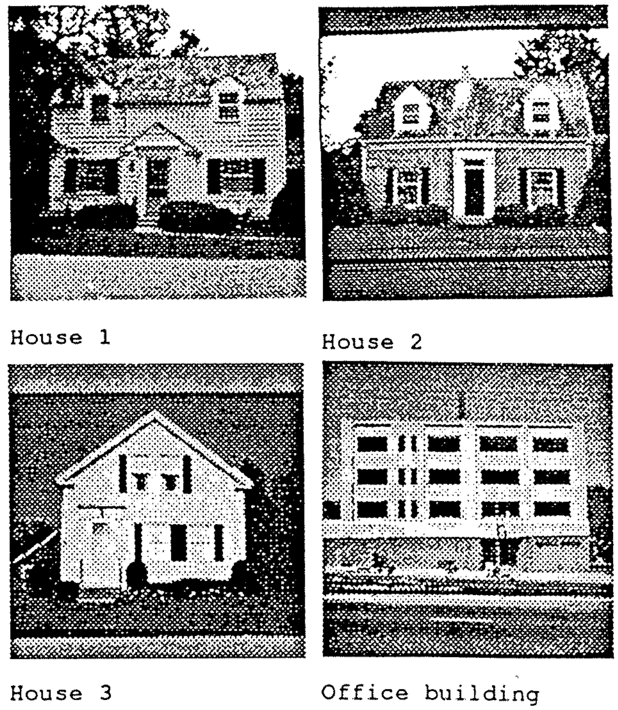

Figure 4: *Digitized 512x512 pictures of buildings that were recognized by the multi-layered image and model-driven Recognition Cones program*

	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.*

the ability of humans to learn successfully in a wide range of environments. How can a system like Recognition Cones develop perceptual abilities by acquiring the necessary transforms as a function of experience?

Learning refers to the acquisition of new knowledge; the development of perceptual, motor, and cognitive skills through instruction or experience; the organization and integration of acquired knowledge into effective representations; and the discovery of new facts, theories, or ideas through observation, experimentation, and thought (Uhr, 1973; Michalski, 1983)

Learning as Constrained Induction

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 is meaningful in modeling the environment. This suggests that the perceptual learning system should be designed so that it is either equipped with, or, develops through learning, structures that enable it to detect and respond to the features, and the relationships among features, in the environment needed to handle the tasks it has to perform.

Given a certain structure, or a set of structural constraints for the development of the perceptual system, knowledge of the environment is gained by a process of induction (constrained by the structure of the system) applied to the information provided by the senses. Induction is the process by which a system develops an understanding of principles or theories that are useful in dealing with the environment by generalization and specialization from specific examples or instances presented to it (Michalski, 1983; Holland, 1986) This includes the process of experimentation and discovery, that is, the setting up of hypotheses and then the accumulating of evidence to confirm or deny their validity.

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). Some of them utilize feedback that allows the network to compute the error between its output and the desired output and use the back-propagated error to change the weights, e.g., the *generalized delta rule* (Rumelhart, 1986b). Some use a form of reinforcement learning that enables the network to utilize feedback in the form of a *reward* for good actions or a *penalty* for bad ones. If a unit can learn to increase the frequency of reward from a noisy *critic*, it can act cooperatively with other units in the network to improve the performance of the entire network (Barto, 1985). Some use a form of *association learning* i.e., a link between two units is strengthened if both of them fire at the same time (Hebb, 1949). Such a scheme tends to sharpen the unit's predisposition, getting its firing to become better and better correlated with a cluster of stimulus patterns.

A learning scheme for [3] that employs a mechanism for activity-dependent, feedback-guided *generation* of new links is described in (Honavar, 1987; Honavar, 1988b).

The Learning of Useful Transforms through Generation and Reweighting

As noted earlier, the adaptive neuron-like unit, the basic building block of Recognition Cones, computes one or more probabilistic or fuzzy transforms over its inputs. In such a system, learning by induction can be viewed as the generation, tuning, and retention of a set of transforms that are adequate for the perceptual tasks demanded of the system. The system learns, or is 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 mechanisms; A particular implementation is explained in detail in the next section. Transforms are modified by changing the weights of their implieds (that is, the output links of the adaptive neuron-like units) and their conditionals (the input links of the units) according to one of the standard reweighting rules; and by changing thresholds of firing or the output functions of the units. New transforms are added by generation, which involves the growth of new links between units (implieds, conditionals, with appropriate weights, which are themselves learned), and recruiting units from a pool of uncommitted units. Thus the network learns, through both generation and reweighting, a set of fuzzy transforms adequate to classify the training patterns correctly, to the desired degree of accuracy; and, because of their probabilistic structures, the much larger set of possible instances the network must handle, and on which it is tested.

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. At the output layer, a node that made the wrong choice releases a *transform down-weight signal* that weakens the links that fired into it from the nodes at the next lower layer. This down-weight signal is propagated back through the network until the input layer is reached. Every node that receives a down-weight signal, weakens the links that fired into it on the training presentation. In a similar fashion, a node at the output layer that would have been correct (had it fired) releases a *transform up-weight signal* that strengthens the links that fired into it from the next lower layer and this up-weight signal is propagated back through the network. This is the form of learning that has been studied widely in connectionist systems, and several algorithms are available for this purpose (Hinton, 1987a). The one we use is very similar to the the *error back-propagation* algorithm (Rumelhart, 1986b).

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 theoretically suffices to enable the network to learn the desired input-output mappings, the fan-in and fan-out required of the units is arbitrarily large. 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 for trivial cases, to foresee the necessary connectivity, and the number and the depth of transformations (which corresponds to the number of layers in the network if no cycling between layers is permitted) needed for a particular task on which the network is to be trained; this is especially true when, in dynamic real-world environments, the task changes over time and learning can never cease.

Only if the network has *an adequate number of appropriately linked nodes* to start with, feedback-guided reweighting of links has a chance of eventually finding the right set of weights that would result in correct classification of patterns in the training set. 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 and to start with a network that is sufficiently large enough to subsume G_E. Thus, generation of new transforms is an essential learning mechanism in networks where the necessary connectivity cannot be established in advance, because no amount of reweighting would enable such a network to learn all the pattern classes on which it is trained. Given mechanisms to generate new links and recruit new nodes as needed, the network can gradually grow, until the number of nodes in the network approaches N_E, and the network topology approaches G_E, whatever N_E and G_E may be.

The Generation of New Transforms

The generation of a new transform is triggered by negative feedback under certain appropriate conditions which will be explained later. Suppose the feedback indicates that the system implied the *wrong* thing. This triggers the release of a *transform generation signal* by units that received negative feedback, which is transmitted to units successively in the layers below that contributed inputs to the units in question, just as the error signal is propagated back for 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. The effect of this is to add a new transform to the existing set. The transforms so added participate in the learning process according to the same principles as those described above. The conditions under which new transforms are added through generation, instead of simply reweighting the existing transforms, will be explained later.

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 the influence of appropriate regulatory mechanisms. 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. The conditions under which it is appropriate for the network to discard transforms are discussed later.

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 these two learning mechanisms. If learning is restricted to reweighting alone, the network may never be able to achieve the desired performance level of recognition, because of reasons outlined earlier. On the other hand, if new transforms are generated each time negative feedback is received by the network the essential process of tuning of transforms by reweighting is disturbed and the network is likely to end up with a large set of transforms most of which are only rarely useful. Similarly, discarding existing transforms has to be done when appropriate. Regulatory mechanisms that decide when to generate new transforms and when to discard existing ones are therefore needed.

One possible mechanism to decide when to generate a new transform is suggested by the need to guide (and possibly goad) the network in the direction of developing into the simplest possible structure (according to some criteria of complexity), that does not violate the topological constraints placed on the network (such as, e.g., the size of a receptive field) that is adequate to perform the pattern classification tasks for which the system is being trained. We call this the *minimal complexity heuristic*. Several measures of network complexity suggest themselves, e.g., the number of transforms (number of nodes, links, or both), the depth of transformations from input to output (number of layers, which in turn determines the time for processing a given input pattern).

One such regulatory mechanism, based on the minimal complexity heuristic (Honavar, 1987; Honavar, 1988b) 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). This requires the network to have some mechanism to keep a (recent) portion of the learning curve for each pattern class on which the network is being trained. The details of implementation are described later. Other regulatory mechanisms based on different measures of network complexity are possible, and are being investigated.

Regulatory mechanisms are needed to decide when to discard a transform. Discarding a transform by breaking its input and output links may appear to be a drastic step, but it is necessary if the network's performance remains consistently poor over intolerably long periods of time (and reweighting and prior generation have failed to give the desired improvement in performance), or when it becomes difficult to grow new links needed to generate new transforms, without violating the topological constraints on the network (because most of the allowed units and links have been used up).

Additional regulatory mechanisms may be needed to determine where in the network to generate new transforms (e.g., in which layer); or to decide which transforms to discard (a transform with fewer output links leaves the network relatively undisturbed). These issues are interesting in their own right and deserve further examination.

The realization of regulatory mechanisms that guide the generation, modification, and retention of transforms requires structures that maintain, update and transmit information (accumulated through local computations) concerning the performance of the network at the task that it is being trained for. Such information may include some measure of the history of the network performance which is used to determine the nature and the extent of changes to be made with learning. If the system has been responding correctly most of the time in the recent past, it perhaps should be conservative in adding, deleting and modifying transforms as a result of feedback. On the other hand, if the system has been responding incorrectly most of the time in the recent past, it should perhaps be more radical in making those changes.

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 sets of fuzzy transforms that are useful for recognizing the objects in the environment. On the other hand, transforms found not useful will fade away, since they are negatively reinforced, or, if necessary, discarded by the learning mechanism. Several questions remain to be answered. For example: What is an appropriate set of regulatory mechanisms? Can higher order control be exercised by the network, as a function of its performance, on the nature of the particular regulatory mechanisms that come into play?

Simulation of Perceptual Learning through Generation and Reweighting of Transforms

This section briefly outlines the simulation of a connectionist network with brain-like structures (e.g., local receptive fields, global convergence) that discovers the transforms necessary for perception of simple 2-dimensional visual patterns through feedback-guided modification of weights as well as the relatively less frequent generation of new transforms. The networks that result from such learning are identical in their architecture to Recognition Cone networks designed for pattern perception described earlier. Much more detailed description of the implementation of the learning program can be found in (Honavar, 1988b).

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 2-tuples 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 several 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 shown by simulations, 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. Generation grows new links and recruits uncommitted nodes into the network as the network learns guided by the input patterns and feedback. The transforms in the network are fine-tuned through feedback-guided reweighting of links. Because generation does not violate the topological constraints of layered, logarithmically converging structure as well as local receptive fields, the networks that are discovered through generation and reweighting resemble Recognition Cones (see figure 1).

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 sub-structures that monitor the network's performance on each pattern class on which it is being trained. The design of these sub-structures is motivated by the need to discover the *simplest* networks capable of the desired accuracy of recognition. The rationale behind the design is as follows: Continue to reweight existing links so long as the network's performance is improving. When it is observed that the network's performance has leveled off (before reaching the desired accuracy of recognition), generate a new transform. This is accomplished easily 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, 1988b). The details of the particular implementation are unimportant for our purposes here. Suffices it to point out that a variety of local computations of this kind may be performed by highly structured microcircuits of neuron-like units (e.g., counters, comparators) as part of information processing, learning, or control functions in brain-like networks

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 randomly 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, as though it is immediately broadcast (either laterally through that layer or up to the apex of the pyramid and then showered back down). 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 results of simulation of the model described above. Experimental comparisons of networks that learn by generation and reweighting, under the topological constraints of local receptive fields and global convergence with networks with varying degrees of structure that learn by reweighting of links alone are given in (Honavar, 1988a; Honavar, 1989a) and are mentioned only briefly here.

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. 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. The results with pattern sets (apple, cup, banana) were qualitatively similar in all the cases (the runs were slightly longer (took 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 for attaining 100% accuracy of recognition.

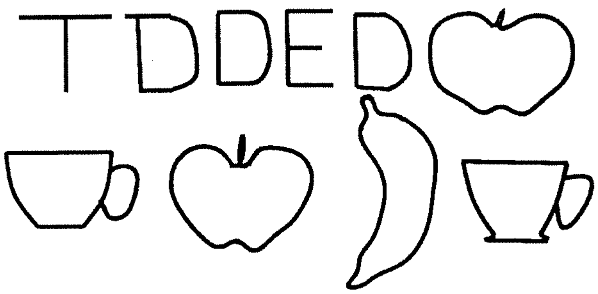

Figure 6: *Sample images used in the simulation of learning*

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 performance of networks with brain-like structure (e.g., local receptive fields, global convergence) that learn by generation and reweighting of the sort discussed here have been compared, using the same training and test data, with that of networks that learn by reweighting of links alone that have the same structure but varying amounts

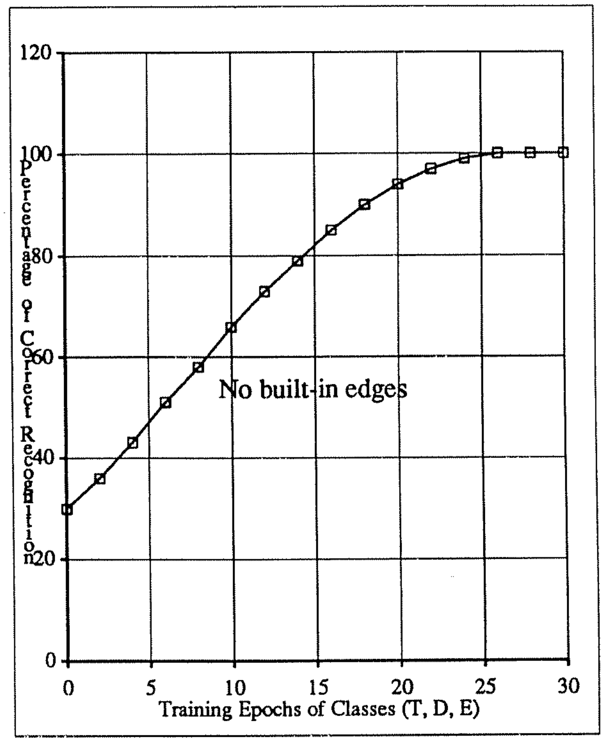

Figure 8A: *Performance on test set with pyramid convergence, generation, local receptive fields, without built-in edges*

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, 1988a; Honavar, 1989a). Although the results are preliminary in nature, they do suggest that other factors being equal, generation and local structure substantially improve learning, both in terms of the number of training epochs needed as well as the size of the networks necessary to attain the desired accuracy of translation-invariant recognition.

Discussion

This section discusses briefly: the distributed nature of information processing in brain-like computing structures (local receptive fields, global convergence-divergence); the role of brain-like structures, generation,

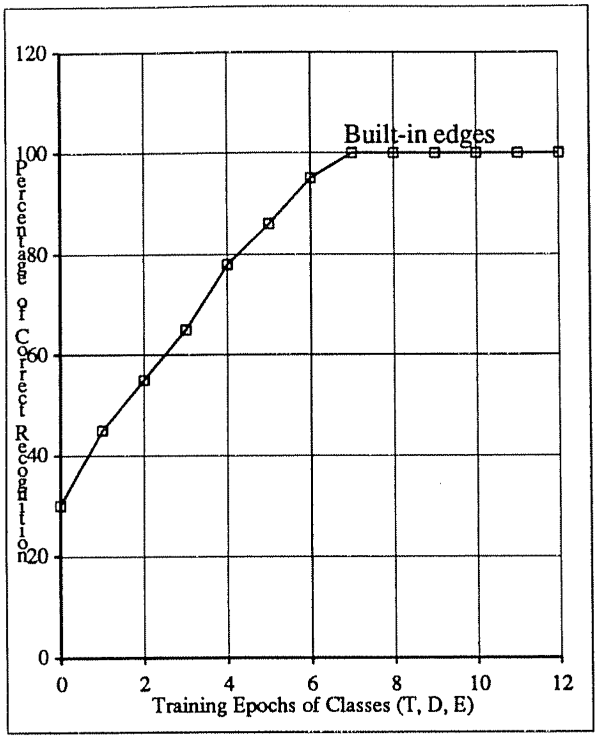

Figure 8B: *Performance on test set with pyramid convergence, generation, local receptive fields, with built-in edges*

and regulatory structures on learning; and outlines some future directions for research.

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

An information processing system is typically thought of in terms of three functions, viz., *processing, memory,* and *control,* each of which may be *distributed* to varying degrees, in several different ways, over the system. It is interesting to examine the nature of this distribution in Recognition Cones and some of the properties that emerge as a result.

First, each of the many micro-modular units computes only a tiny part of the large and complex function that perceives. Thus, processing is clearly distributed. Recognition of a pattern is the result of utilizing knowledge distributed over several successively more complex transforms. Each transform extracts 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 is distributed over a large number of neuron-like units. Memory in brain-like computing structures is distributed over the large number of units (and each unit's output functions, thresholds, activation levels, etc.) and the links (the weights and learning rules associated with them).

Control, that is, the decision to execute one instruction (or apply a transform) or another - in a neuronal system is locally but collectively exerted by individual units providing the inputs for other units. This is supplemented by micro-structures (e.g., the regulatory structures that decide when to generate new transforms). Thus, control of the information processing functions performed by the system is the result of the consorted action of a large number of functional modules. Some control is exerted by the structure of the system itself, for the connectivity determines the direction and order of information flow, and this in turn governs what transforms get applied. Thus 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), connectionist networks must incorporate sufficient noise immunity as well as redundancy. In Recognition Cones, this is accomplished by the fuzzy nature of the transforms. Also, windows and samplings are used, with thresholds that accept a variety of different patterns of firings. Redundancy and robustness may be increased by using several transforms that look the same feature (say long vertical line) in approximately different edge detectors are used and all their results combined; to detect angles, several different detectors, one concentrating on properly interrelated edges, another on the vertex, and others on various aspects of the interior and exterior regions, and so on. Thus the different but closely related processes that serve overlapping purposes provide the redundancy that is needed to ensure robustness (graceful degradation under damage - failure of a few random units or loss of a few random links, analogous to the death of a few neurons) as well as noise immunity.

Role of Brain-Like Structures in Learning

Constraints on the network topology determine the space of transforms that can be learned, and bias the network so that the learning of certain relations is favored. Retinotopic mapping and local receptive fields exploit spatio-temporal contiguity in the environment. This favors the discovery and learning of relations between subpatterns that are imaged onto neighboring regions of the retina. For example, if the object imaged is a chair, its sub-parts (e.g., legs, seat, back-rest) are projected onto the neighboring parts of the visual field with all the spatial relations between them intact. Our results suggest that the choice of constraints on network connectivity is 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 (in the layers as well as over several presentations of the patterns). Generation, under brain-like constraints on connectivity, 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. The effects of this are two-fold: Learning of simpler relations preceeds the learning of more complex relations; Successively more global relations are learned at successively higher layers. This is confirmed by an examination of the transforms generated by network simulations.

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 that starts out with a random subset of the possible connections and is constrained not to change the initial topology, and whose learning is restricted to reweighting of links alone.

Feedback-guided reweighting of links by small amounts changes the pattern descrimination properties of the network gradually. Reweighting tends to minimize the error between the actual and the desired outputs of the network for the various pattern classes, by effectively performing a *gradient descent* on the error function. However, there is a risk of getting caught in a local minimum, a shallow trough, or a valley in the (Rumelhart, 1986b) of the error surface. Generation and discarding of transforms can be thought of as providing the network some means of getting out of such local minima.

Network structures that maintain, update, and transmit as appropriate, information about the network's performance over time (e.g., a portion of the learning curve, used to trigger generation) offer several interesting mechanisms to influence learning that may be worth examining. Such structures may be used to alter learning strategies, rates of learning, thresholds of firing, each of which has an impact on the *plasticity* of the network. Future work will address some of these issues.

The extent of generalization i.e., building of meaningful internal representations by discarding uninteresting details, is an important property of connectionist systems that learn. More compact representations result from better generalization. There is reason to believe that the extent of generalization in connectionist networks is sensitive to the number of hidden units as well as the connectivity (Hinton, 1987b; Rumelhart, 1988). If the hidden units or connections are too many, the network may generalize rather poorly; if they are too few, the network may never learn. Thus, finding the optimal number of hidden units and/or weights is of interest. Generation and deletion of links can be seen in this context as providing mechanisms that dynamically determine the number of hidden units and connections needed in the network. Thus, such networks may exhibit good generalization properties as well. Generation makes possible the linking up of an adequate number of units to solve a given problem; minimal generation favors the discovery of the smallest necessary number, and hence, better generalization. Future work will examine this conjecture experimentally.

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, since these programs learned to recognize the pattern-sets they were tested on in relatively small number of training epochs. But to handle larger sets of more complex patterns, the ability to discard is almost certainly necessary; otherwise the network will get bogged down with many poor or worthless links. Evaluation of the performance of brain-structured networks that learn by generation, reweighting and discarding of transforms on more complex pattern sets (e.g., faces and other real-world objects) is in progress.

There are a number of promising improvements to be made, including the addition of networks that make better assessments of potential generations, that learn to improve upon these assessments, that evaluate the generations for their usefulness for recognition, that discard poor generations to make room for new ones, that narrow and broaden the tolerance-threshold for matching, and that 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-networks that realize functions for deciding whether to further re-weight or to generate, the optimal number of nodes in a node-cluster, and whether it helps to put nodes within a cluster into direct competition. It may be informative to explore generation as part of an *unsupervised* learning scheme, and/or in combination with other reweighting rules e.g., some variant of the Hebbian rule (Hebb, 1949). Performance of generation learning on problems other than vision (e.g., speech) is also under study.

The Role of Reciprocal Connections in Information Processing and Learning

In a brain-structured network, reciprocal connections between layers provide a means of *model-driven, expectation-driven* or *top-down* processing without the need for explicit top-down control. Partial evidence that is passed on to a layer from its predecessor in the bottom-up pathway can activate processes, that through the reciprocal connections, intiate an iterative relaxation that would eventually find a globally consistent solution through an interplay between several local sources of information. Future work will examine the role of such reciprocal connections in learning in brain-structured networks.

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 in large 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., "Experimental results indicate that generation, local receptive fields and global convergence improve perceptual learning in connectionist networks," *Computer Sciences Technical Report #805*, University of Wisconsin-Madison,

Madison, WI, 1988a.

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, 1988b.

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.