Temporal Computation in Connectionist Models

Erich J. Smythe

Submitted to the faculty of the Graduate School
in partial fulfillment of the requirements
for the degree
Doctor of Philosophy
in the Department of Computer Science
Indiana University

June, 1988

Accepted by the Graduate Faculty, Indiana University, in partial fulfillment of the requirements of the degree of Doctor of Philosophy.

John A. Barnden

John A. Barnden, Ph.D.

Daniel P. Friedman

Daniel P. Friedman

Doctoral Committee:

Conrad Mueller

Conrad Mueller

Robert F. Port

Robert Port

Dirk Van Gucht

Dirk VanGucht

June 7, 1988

To my parents,
Mary and Jack Smythe

Acknowledgements

I want to thank my thesis advisor, John Barnden, for the effort, patience, and wisdom that goes into directing a thesis. Even when at a distance he provided just the right amount of advice to keep me going in the right direction. I am also grateful to Bob Port for insights and direction in speech research that was essential to this project, as well as the final coat of polish on the thesis. To the rest of my committee, Dan Friedman, Dirk VanGucht, and Conrad Mueller, you have my thanks for your thoughtful comments and ideas during the course of this work. I would also like to thank Stan Kwasny, who served on an earlier version of my committee before he left Bloomington.

I am indebted to Diane Kewley–Port for giving me her own thesis data for use in this project. It made all of this possible. The group at the Indiana University Phonetics Laboratory provided essential support during the course of my speech research. I am grateful to John Merrill for the endless discussions, and to Sven Anderson and Jane Hardy–Essid for teaching me to use the facilities of the lab.

I would like to thank the organizers, sponsors, and participants of the Sloane Foundation Connectionist Summer School in June, 1986, where the seeds of this

research were planted.

Without the help of the office staff at the Computer Science Department in Bloomington, Karen Miller, Jenny Mobley, Jill Porter, Kathy Thompson, and Dedaimia Whitney, I would still be lost in the administrative maze. Thank you for telling me where to go. I am also grateful to the systems staff, too numerous to mention, who were always helpful, even with the most insignificant requests.

Finally, I want to thank my family and friends for your support, encouragement, and prodding (when necessary), and for putting up with my occasional frustrations along the way. There are too many of you to mention, or even single out, but without you, this would have never of happened.

During the course of my graduate studies, I was supported financially by the Indiana University Computer Science Department as an associate instructor, and by a grant from the Spencer Foundation as a research assistant. My research was also partly supported by NSF grants DCR–8505635 and DCR–8518725.

Abstract

Temporal computation in connectionist models is studied through the application of a mechanism for visual motion to the detection of acoustic motion. This mechanism is then used in the recognition of isolated stop consonant–vowel (CV) syllables. SYREN is a connectionist model that recognizes 24 CV syllables by tracking the movements of formant center frequencies. SYREN is divided into three parts: a motion detector network to identify the rate and direction of formant transitions, an adaptive network that associates patterns of formant transitions with each syllable, and a veto recognition network that uses patterns from the adaptive network to actually recognize the syllable. The network is dynamic, taking formant center data sequentially in 5 ms time slices.

A veto network is used in the motion detection phase. This is constructed using veto inhibition and characteristics of local dendritic computation from a visual motion mechanism in some vertebrate systems. Neural veto inhibition is a shunting inhibition that blocks the flow of excitatory activation without affecting the membrane potential. A single node in a motion detector performs computations analogous to a small subunit of a dendritic tree of a neuron, allowing the implementation of activation flow, decay, and veto inhibition in a subnet-

work that is analogous to a single nerve cell. Veto inhibition blocks the flow of activation through the subnetwork.

The adaptive network uses a hybrid of the Widrow–Hoff rule with modifications from a mechanism of classical conditioning. It uses formant transition information as input, with transition patterns preserved for 5 time slices through delay lines similar to temporal sequence detection mechanisms. The data corpus consists of formant centers from five repetitions and averaged centers from 24 CV syllables. The network is trained on five tokens from the data corpus and tested on the sixth. A recognition network using veto inhibition is constructed after training and uses activation patterns from the adaptive network to recognize each syllable. The network achieves 100% recognition on its training data and 79% recognition on its testing data, using information from both vowel and consonant portions of the speech signal.

Contents

List of Figures

List of Tables

1. Programming from Neuroscience

Computational methods arise from a number of sources. Mathematics and Logic are two of the fields that have influenced Computer Science. The Predicate Calculus has led to the logic programming language Prolog [25], while the Lambda Calculus [24] has affected the design of a number of other programming languages. Functional optimization has been accomplished through a process called simulated annealing [56], motivated from Chemistry and Physics. When viewed within the scope of Computer Science, Artificial Intelligence (AI) may be thought of as the application of techniques gained from the study of intelligence to difficult problems, and the actual implementation of intelligent systems. Computer Science has benefited from contributions of Artificial Intelligence such as symbolic and rule–based problem solving methods. Most of these techniques attempt to exhibit intelligence at a rather abstract level, far from its actual

implementation details in biological systems.

The brain is a computational device whose processing mechanisms differ from traditional digital sequential computing. Characterized by massive parallelism and interconnectivity, the brain is capable of remarkable computational feats arising from the cooperation of a large number of neurons, whose basic computation is relatively slow compared to conventional digital processors. The properties of the brain combine to form a device noted for its ability to store and retrieve information in an efficient manner. The brain can use this information in a wide variety of ways by capturing the complex interrelationships that characterize human intelligence, and adapt and apply that knowledge to new situations. Almost all biological neural systems are noted for efficient and effective perceptual processes that are unsurpassed by artificial implementations. These neural systems display a property of fault tolerance, the ability to function with damaged or missing elements, that no conventional computer has been able to achieve.

Artificial Intelligence and Computer Science have seen limited success in a few areas of intelligence, but have yet to scratch the surface of the performance of even the most primitive biological neural systems. Traditional AI techniques have been applied to perceptual problems, such as vision and speech, with relatively limited results.

There is a piece, if not an entire section, missing from the puzzle. Neuroscience can provide clues and insights into the nature of the best example of an intelligent computational device available, the brain. It is possible that the nature of these problems requires the application of analog, parallel computational

techniques common to neural devices [64], and that the study of these neural systems may give insights into the construction of an artificial implementation. Computer Science is, in some sense, the study of computational devices, and it is well served by the study of natural systems. Ideas from neurophysiology can be applied not only to problems in intelligence but to problems in other domains as well.

Connectionist models are networks of nodes computing in parallel and communicating information to other nodes through connections. Traditionally a node computes a numerical or binary activation value and transmits this on its output connections. A node's activation is normally determined by the activation values available on input connections. The kind of computation performed by these models is motivated by neurophysiological ideas and, traditionally, a node in a connectionist model is treated as a very simplified analog of a nerve cell or possibly an assembly of several cells. These models offer an alternative to traditional AI programming techniques and have seen some successes in vision, speech, and associative memory [98, 75, 64]. They have also been applied to tasks that do not necessarily fall within the normal domain of Artificial Intelligence such as adaptive process control [37], radar imaging [34], and communication [128]. Many models are adaptive networks in which only the initial configuration need be specified, with the final behavior the result of training.

Connectionist networks offer another computational method not only to the field of Artificial Intelligence but to Computer Science in general. Neuroscience has contributed a class of methods along with fields such as Mathematics and Logic. Most connectionist models, however, use highly idealized abstractions of

neural behavior while leaving out the majority of the rich structure and function known to neuroscience. It is possible that implementations of certain behaviors may require a closer study and modeling of functional properties of the nervous system.

1.1 Temporal and Dynamic Connectionist Models.

Connectionist models can be divided into two classes. A static model is given input all at once and computes its output for that information before the next input is shown. An associative memory is an example of a static model that gives an output pattern B when presented with an input pattern A. Some connectionist speech recognition models are static, and are presented with an entire utterance all at once to compute an output with the entire signal available. Dynamic models are given input in steps or time slices and must accumulate information spanning over a time frame before producing output. In the strongest sense a dynamic model does not preserve the raw input signal or assign parts of its structure to particular moments of the input. An example is a visual recognition system where input receptors are updated after a very short duration, such as a five millisecond refresh cycle.

Dynamic models are important in systems for temporal sequence detection, where several input patterns are presented before an output is expected. The detectors are like associative memories except that the input patterns are spread over a longer temporal window. It may be impractical to store several input frames in full scale applications of vision and speech systems. This requires the use of dynamic models and the need to represent temporal effects. These models

also work in the opposite direction, where a single input produces a sequence of output patterns. This may arise in motor control problems and planning, which require computations in a temporal domain.

The detection of visual motion requires the separation of moving objects from stationary ones and determining the direction and rate that the body is moving. In the visual system this is thought to be accomplished through cells that are responsive to different types of motion. These motion detector cells are believed to exist throughout the visual system from retinal cells whose axons travel in the optic nerve [6, 125, 63] to the visual cortex [47]. These cells have a receptive field spanning a portion of the retinal receptors corresponding to a part of the retinal field. If an object is in that part of the receptive field it will affect the corresponding retinal receptors. Objects moving across the receptive field affect a cell in a way that depends on the direction and rate of movement. Stimuli used in the laboratory in studies of these cells are normally quite simple, such as bars, lines, or grids. The motion of an object must subtend a substantial portion of the cell's receptive field to have an effect, and a motion detector in the retina is not excited until it accumulates information from a number of light receptors. This computation spans a time frame and requires some temporal integration of input.

A speech signal is characterized by dynamic changes of spectral properties and steady–state periods. A formant in a speech spectrogram is a concentration of energy centering on a particular frequency at some point in time. Formants exhibit steady–state frequencies as well as periods when the central frequency is changing. Many parts of speech are characterized by formant transitions, and

research has shown that the nature of these transitions can provide important clues to the identity of these constituents [70, 111, 28].

Formant motion is analogous to visual motion. The detection of the rate and direction of formant transitions requires the temporal integration of spectral information much like the computation required for visual motion. If a mechanism is found for visual motion, it can be applied to the detection for formant transitions, with this process integrated into a speech recognition system.

Through a detailed study of parts of the mammalian visual system, a model has been developed of cells in the retinal ganglion that are sensitive to visual motion [6, 125, 63]. The implementation of this mechanism requires some modifications to current connectionist formalisms. Traditionally, a node in a connectionist network takes a linear weighted sum of the activation values of its input connections. A connection can have an excitatory or inhibitory effect on a node. A value on an inhibitory connection reduces the effect of an excitatory value of another connection by reducing the node's activation. Real neurons, on the other hand, exhibit both spatial and temporal integration of the effects of synapses. This means that the location and temporal effects of a synapse play an important role in the behavior of the nerve cell. If these nodes are to be analogous to nerve cells, input summation must take into account the structural characteristics of the cells and the location of the input synapses. Furthermore, there exist non–linearities in the excitatory and inhibitory results of synapses.

The mechanism for visual motion is a combination of veto or shunting inhibition with the integration of effects of spatial subunits on the dendritic tree of motion detector cells [63]. The effects of veto inhibition differ from the linear

integration of inhibitory connections in traditional connectionist models. With a veto synapse a small amount of inhibition can veto a large amount of excitation. Furthermore, the location of a shunting synapse has an important role in its effect. Excitation flows from synapses on the dendritic tree of a nerve cell to the cell body much as the tributaries of a river flow to its mouth. Veto inhibition can selectively block excitation from specific synapses depending on where the inhibitory synapse is located with respect to the source of the excitation and the body of the cell. If the veto inhibition is located at the confluence of several dendritic branches, the excitatory effect of all the synapses behind that portion of the cell can be vetoed by a single inhibitory synapse.

1.2 Application of the Mechanism to Acoustic Motion

This research explores the effects of temporal computation in connectionist models through the implementation of spatial and temporal synaptic integration in the form of a veto network. Speech recognition serves as the test bed for the application of mechanism of visual motion outlined by Koch, *et al* [63], and this mechanism is applied to the detection of formant center transitions. Formant transition information is then used in the recognition of voiced stop consonant vowel (CV) syllables. Veto networks are used both in the detection of acoustic transitions and in final syllable recognition. An adaptive network serves as an intermediate level of processing, integrating transition information and passing it on to the veto recognition network.

The system, named SYREN for SYllable REcognition Network, is a connectionist–style model consisting of three subnetworks. The first detects the

rates and directions of formant transitions and steady–state frequencies from formant center input. The second subnetwork is a learning model that takes the output of the motion detectors, spreads them out slightly over time, and through an adaptive process associates patterns of acoustic motion with each syllable. This process is not perfect, and another veto network takes the output of the adaptive network and fine tunes the process by eliminating the majority of the errors. This network provides the final output of SYREN.

The data corpus consists of formant centers taken from five repetitions and averaged centers from each of twenty–four naturally produced CV stop consonant syllables [52, 55]. Formant center information is converted to a bit matrix where each row represents the frequency of the formant and each column a 5 ms time slice. The matrix is presented to input nodes of the network one slice at a time. Information from previous presentations is not retained by the input nodes.

There are nineteen different types of motion detectors which preferentially respond to twelve specific rates of transitions, six rates in each direction, and to a steady–state condition. Three faster transition rates are assigned two types of detectors. Each detector is itself a subnetwork whose architecture is analogous to the dendritic tree of a nerve cell. A node in a subnetwork corresponds to a patch of cell membrane whose connections are analogous to exterior synapses and also serve to connect the patch to others in an artificial dendritic tree. Exterior connections are either excitatory or provide shunting inhibition. The activation of a node is computed using equations incorporating saturation and decay properties for nodes on branches. A sigmoid squashing function is used for the node that serves as output for a detector. Veto inhibition is used for both direction selec-

tivity and rate sensitivity. The detectors are copied over the frequency spectrum from 0 to 4000 Hz.

The adaptive network is a single layer of nodes with each node assigned to a particular syllable. The weights of the input connections of the nodes are set by an adaptive method so that the node will respond to transitions associated with its assigned syllable. The adaptive algorithm incorporates ideas from self organizing temporal sequence detection systems as well as classical conditioning methods. The nodes are connected to a delay matrix which preserves transition information for a few time slices.

Detector nodes occasionally fire on the wrong syllable, and the veto recognition network uses patterns of errors to ensure accurate recognition performance. The final output of the network is through twenty–four recognizer nodes, each assigned to a syllable. Although not as complex as the motion detectors, the network uses veto inhibition to prevent a node from firing on the wrong syllable. It is constructed after the performance of the learning phase is evaluated.

1.3 Aim and Focus of the Project

This research primarily focuses on the implementation of a connectionist model which uses temporal properties of input in its computation. Time is not explicitly represented, but is incorporated in many ways. This is accomplished through the construction of a computational tool, the veto network, that is used in a dynamic model that must respond to temporal and sequential characteristics of its input. This tool is developed from a neurophysiological mechanism of visual motion detection, itself a temporal process. The contribution of this method

and its usefulness as a general computational tool is demonstrated through its application to a problem in speech recognition. The use of veto networks at two levels in SYREN indicates that veto inhibition may be useful in many problems requiring the use of temporal and sequential information.

This research also explores the treatment of speech recognition in a dynamic manner. This system takes its input in a left–to–right fashion, a way that is more closely related to human perceptual processing systems. The input signal is divided into short time slices 5 ms in length. The system explores the usefulness of transition information coupled with steady–state properties in recognition, including information that exists at phonemic boundaries. Recognition focuses on the syllable level rather than on phonemes, and incorporates consonant–vowel contextual information.

To understand the goals of the research it is important to understand what it is not attempting to address. From the standpoint of neurophysiology, no claim is made about the mechanism of acoustic motion detection in the auditory nervous system. Evidence is cited supporting the existence of cells sensitive to spectral change. Although veto inhibition is used to implement acoustic motion detectors, this research does not suggest that this is the mechanism used in the auditory system. Any such claim must be made within the results of detailed laboratory experiments which are outside the scope of this research.

The only linguistic claim made is that formant transitions can be useful in the recognition of stop–vowel syllables. They are by no means purported to be the only information that is useful, nor the most important. Furthermore, the syllable serves as the basis for recognition, not the phoneme. Thus the

network is not being trained to search for context–invariant cues for the identity of phonemes, a point of contention in speech research. Syllable recognition, by its very nature, must rely on the interaction of the constituent phonemes of the syllable.

Although learning methods are a popular field of study, this work does not focus on adaptive methods in dynamic connectionist networks. That topic itself is worthy of a separate project. Here, learning is used as a programming tool to aid in the development of the recognition network. The method used is a hybrid of several other methods, and gives acceptable performance in the context of this research. Errors and limitations arise from the fact that the detection network consists of a single layer of nodes, but a multi–layer network was rejected due to the increase in computational requirements of several orders of magnitude.

The bulk of SYREN is constructed by hand rather than appealing to learning techniques. The reason for this is illustrated by examining the continuum between *tabula rasa* learning networks and fully hand–built models. Most learning research concentrates on fully and uniformly interconnected feed–forward networks. These have few initial architectural specifications to guide the early learning process, other than the number of nodes. This is the *tabula rasa* approach, where the final structure and behavior of the network arises from very little initial structure. It is clear from the study of nervous systems that the brain is amenable to a considerable amount of adaptation, but it is also clear that this behavior arises from a very rich initial structure, with a considerable number of features and processes already "pre–wired". By providing more initial structure, an attempt is made to give the learning process a head start, result-

ing in a more powerful learning process. This head start is gained from the study of neural mechanisms, and illustrates the underlying philosophy guiding this research: that the study of neurophysiology provides ideas for more powerful computational tools in Artificial Intelligence and Computer Science.

1.4 Plan of the Thesis

This thesis describes the ideas, motivation, and implementation of SYREN. The next chapter provides a brief overview of neurophysiology, acoustics, and connectionist models. The mechanism of visual motion detection is described, along with its implications. The role of formant transitions is discussed to provide a motivation for the application of the mechanism of visual motion.

Chapter 3 describes the implementation in detail, including the data used, the motion detector network, the adaptive process, and the veto recognition network. It describes how the input is presented and what constitutes the output of the system.

Chapter 4 shows the results of the experiments, describes how the veto recognition network is actually prepared, and discusses the effects of various parameters on the system's performance. The results are analyzed in an attempt to discover what information the network is using in the detection process.

Chapter 5 discusses the results of the experiments and the implications of the research to connectionist models and speech recognition. The motion detectors are evaluated from standpoints of neurophysiology and psychophysics. The performance of the syllable recognition network is discussed in light of similar

connectionist models as well as some linguistic work. The weaknesses of the system are critiqued, and are related to future research problems. The chapter ends with a discussion of the contributions made by this research to computer science and other fields of study.

Finally, Chapter 6 provides a summary and an outline of future directions suggested by this research.

2. Motivation

The implementation of models from neuroscience can serve different purposes. One may propose a neurophysiological or psychological model and use a computer simulation to test whether that model actually produces or predicts a certain behavior. In this case the implementation serves to test a hypothesis and contribute to knowledge of human information processing mechanisms. This research explores another possibility, where the knowledge from neuroscience is used in the development of problem solving methods. These computational metaphors derived from the nervous system may be applied to problems in Artificial Intelligence as well as other fields such as adaptive process control [37] and radar imaging [34].

This chapter describes the neurophysiological ideas and evidence used in SYREN. Traditional connectionist models are discussed as well as how and why

traditional notions are extended in this implementation. This is not intended to be a complete introduction to neurophysiology. It is instead an attempt to show how specific knowledge of the nervous system is used to construct a computational tool, and how evidence from the mechanisms of the auditory system and speech research suggest that it be applied to a problem in speech recognition.

The chapter begins with a discussion of connectionist models, along with the necessary neurophysiology to show their motivation. The Koch, Poggio, and Torre mechanism for motion detector cells is described, and extensions to traditional connectionist models are presented to deal with this mechanism. Acoustic motion is discussed including its importance to speech recognition.

2.1 Connectionist Models

Connectionist Models [35, 43, 98] are neurally inspired information processing methods. Known also as Neural Networks and Parallel Distributed Processing models [98, 75], they have been applied to many problems including vision, speech, and associative memory. Their primary characteristic is that a large number of units or nodes exhibit complex behavior by performing simple computations in parallel. The nodes of a connectionist model are highly interconnected, communicating simple numerical information between nodes. This is similar to the architecture of the brain, which consists of a large number of interconnected cells, also computing on a massively parallel scale. Even though our knowledge of the brain and its constituent nerve cells is in its infancy, we can be sure that current connectionist implementations are highly simplified analogs of the type of computation believed to be performed by neurons. Nevertheless, a brief de-

scription of the brain can serve to show the motivation for these computational models.

2.1.1 Some Neurophysiology

The brain is composed of information processing cells called neurons, and support cells called glia. Functions such as autonomic activity and coordination are handled in the lower regions of the brain. These regions also act as switching stations for perceptual stimuli. Complex tasks normally associated with intelligence are believed to take place at a higher level in the cerebral cortex. The cortex generates the most interest in the study of intelligence, but the processes and mechanisms in the lower sections, such as early visual processing, are important to study as well.

The cortex is divided into various cytoarchitectural regions [91, 133]. The cells of many regions perform some specific function or respond to a particular sensory modality. For example, cells in the visual cortex respond to visual stimuli, and cells in the motor cortex can cause movement. Recordings of the activity of single cells show that many areas in the cortex are arranged topographically. For example, Hubel and Weisel [48, 47] have shown that the representation in the visual cortex is retinotopic, meaning that a neuron responds to stimuli in a specific region of the retina, called the receptive field. Adjacent cells have receptive fields in neighboring retinal regions. Similarly, both somatosensory and motor cortex have been shown to have a somatotopic representation, with neighboring cells responding to stimuli from adjoining regions of the body, or in the case of motor cortex, causing movements in neighboring regions. With careful

recording techniques, something resembling a homunculus may be projected on the somatosensory and motor cortex, although the hands and face receive a disproportionate amount of space. The auditory cortex shows a different sort of topography: a tonotopic representation. Cells in this area are sensitive to stimuli in specific frequency ranges. Adjacent cells have receptive fields of similar frequencies in a sort of frequency map. Some form of spatial map has also been found in the auditory cortex of the bat [89] and the cat [124], with cells responding to stimuli from different regions in space.

Neurons in the cortex and throughout the brain show a functional and morphological diversity. A typical cell may contain arboreal or branching processes, known as dendrites, a cell body or soma, and a longer process with occasional branching known as the axon. These processes come in contact with other cells and allow communication of neural information. One neuron connects to another through a structure called a synapse. In the classical sense, the dendrites serve as the input receptors for the cell and the axons transmit the output, although there are many variations on this type of synapse throughout the nervous system.

A neuron, like other cells in the body, is encased in a cell membrane that forms a barrier between the cell's interior and its environment. A characteristic of this barrier is a membrane potential that is a charge difference across the membrane. This is caused by different concentrations of specific ionic species in the interior of the cell with respect to the exterior, maintained by chemical processes in the cell membrane. When the cell is not active there is a greater concentration of negative ions in the interior of the membrane, causing the membrane to be polarized. The resting potential can be disturbed by applying an electric current

to the membrane. Small negative currents depolarize the membrane, temporarily reducing the membrane potential. If the magnitude of this depolarization exceeds the cell's threshold of excitation, an action potential results and the cell is said to fire. Called a spike from its appearance on an oscilloscope, the action potential is a sudden reversal of the membrane potential that travels from the soma down through the axon. This is believed to be the message transmitted by the nerve cell.

The distal regions of the axon show some branching, and these branches contain structures called axonal or terminal buttons. If the terminal button is in close proximity to another cell's dendrite it forms an axo–dendritic synapse. Synapses may also occur on a cell's soma and axon. The terminal button contains packets called synaptic vesicles, which store neurotransmitter substances. An action potential, upon reaching a terminal button, causes the release of some of this transmitter substance into the synaptic cleft, which affects the postsynaptic membrane in some way. The effect is primarily of an excitatory or inhibitory nature. In the excitatory case, the membrane is depolarized, bringing the cell closer to its threshold of excitation. In the inhibitory case, the membrane may be hyperpolarized, making it more difficult for the cell to generate an action potential. These effects, referred to as excitatory postsynaptic potentials and inhibitory postsynaptic potentials are believed to be the primary means of intercellular communication.

There are many synapses in a cell's dendritic tree. The interaction between excitatory and inhibitory synapses affects the firing behavior of the cell. The combined effects of each synapse is termed neural integration. This integration

has both spatial and temporal characteristics. The interaction of synapses is sometimes viewed as resembling an additive process. The effect of one excitatory synapse is strengthened by the activation of another at close proximity. This combined effect is normally greater than that for synapses located farther apart. This is spatial summation. Temporal summation refers to a stronger reinforcement of synapses if they are activated within the same time frame. The effects on a postsynaptic membrane can decay over time.

From an information processing standpoint, an idealized neuron computes a value representing its current level of activation. This activation is sometimes thought to represent spike frequency when using real activation value, or the presence of a spike when using binary output. This value is transmitted in some way across the synapses by spiking behavior. The activation is computed by combining the effects of the synapses through neural integration. It is generally assumed that the activation value and the information transmitted across the synapse is of a simple nature. More complex behavior is obtained by a large assembly of neurons acting in concert. This idealized notion ignores much of the richness of structure and function found in the nervous system. Nevertheless, it serves as a metaphor for many connectionist models.

2.1.2 Structure and Function of Connectionist Models

A connectionist model is a network of interconnected computing elements called nodes or neurons, computing numerical values that are transmitted to other nodes through connections. A threshold logic unit [77] is a node with a binary activation value. The computation performed by a node is called its transfer

function. One example is the function

$$a_i(t) = \begin{cases} 0, & \text{if } \sum_j^n w_{ij} a_j(t) < \theta \\ 1, & \text{otherwise,} \end{cases}$$

where $a_i(t)$ is the activation value of node i at time t, θ is a threshold, $a_j(t)$ is the activation value of a node j connected to node i, and w_{ij} is the synaptic weight of that connection. The term $\sum_j^n w_{ij} a_j$ is the weighted sum of the values on the node's connections. The synaptic weight is a real value representing that connection's influence on the state of the node. The computational behavior of a node, and hence the entire network, given a specific transfer function, is determined by the network topology and the values of the synaptic weights. In many models input is presented to the network which repeatedly computes activation values until it settles into a stable state where the activations are no longer changing [46, 56, 107].

There are a variety of transfer functions seen in different models. Almost all are based on the linear weighted sum of the connections. In the simplest case, the node's new activation value is the weighted sum. Other models [97, 42, 107] use a sigmoid squashing function

$$a_i(t) = \frac{1}{1 + e^{-(\text{net}_i - \theta)/T}}$$

where net_i is the net input to the node i (usually the weighted sum), θ is a threshold, and T is a parameter called temperature that determines the shape of the sigmoid curve. This function can also serve as a probability function for stochastic connectionist models [1, 8].

Even more complex transfer functions allow activation to decay to a resting value in the absence of input. Grossberg [40] separates excitatory and inhibitory

input in the equation

$$a_i(t+1) = a_i(t)(1-\alpha) + \text{net}_i^E\big(M - a_i(t)\big) - \text{net}_i^I\big(a_i(t) + m\big)$$

where α is a decay constant, M is a maximum activation value, m is a minimum, net^E is the net excitatory input, and net^I is inhibitory input. This equation allows the activation to decay to 0 in the absence of input. McClelland and Rumelhart [76, 99] use the equation

$$a_i(t+1) = a_i(t)(1-\alpha) + \begin{cases} \text{net}_i(t)\big(M - a_i(t)\big) & \text{if } \text{net}_i(t) > 0, \\ \text{net}_i(t)\big(a_i(t) - m\big) & \text{otherwise,} \end{cases}$$

in their letter activation model. In these equations, the effect of the input is to push the activation towards a maximum or minimum value, the rate of which is determined by the magnitude of the input. In each of these functions, the input effect is determined by a linear sum. In the case of this research, non–linear effects of excitation and inhibition are needed to capture direction selectivity. This is developed in the next chapter.

An important feature of many connectionist models is the ability to learn. Adaptive network models modify their synaptic weights and in some cases change the network topology to produce a desired behavior. Although growth in the nervous system is of considerable interest in neuroscience, connectionist models have only recently addressed the issue of topology learning. The foci of topological learning efforts have been on a study of dynamic links [36], programmable networks [73, 93], and stochastic search based on genetic algorithms [44, 80]. Adaptive networks based on weight modification are much more common. Most of these models are based on what is know as the Hebb Rule [41] which has been

paraphrased to say that: "If A and B are simultaneously excited, then increase the weight of the connection between A and B [98]." An extension of this rule which pertains to forms of supervised learning is that: "If A is active, and it is known that B *should* be active, then increase the strength of the connection between A and B." These rules have also been applied in the case of learning inhibition. A more thorough discussion of learning algorithms appears in Chapter 3.

2.1.3 Temporal Pattern Recognition

One feature common to many connectionist models is the static nature of their input. In these models input is presented to the network all at once, with the network performing computations and presenting its output. An intelligent organism, on the other hand, is faced with environmental input that is constantly changing. The visual world is an example where the input to the eyes is in a constant state of flux. Auditory stimuli are processed by organs whose output represents a state at one precise instant. In a few milliseconds the sounds can change drastically. The ability to recognize temporal patterns seems innate to intelligent and even non–intelligent organisms.

Most networks that attempt to recognize speech present the speech utterance all at once, with different parts of the network representing specific times, such as the 200th millisecond *vs.* the 220th millisecond of the utterance. Yet the speech signal is an inherently left–to–right phenomenon, and it would seem informative, if not essential, to approach speech recognition in a stream–like fashion, processing the signal one short time slice at a time. Information is not spread

uniformly in the signal, however, and some recognition can be performed using short portions of the signal. In this case, it boils down to the recognition of a sequential pattern, albeit a complex one.

Imagine a robot attempting to catch a ball. Its visual system records the position of objects at one particular instant. To catch a ball the robot must calculate its velocity and trajectory. This may be accomplished by storing several of the past images in memory and comparing the position of the ball in each. This is an extremely expensive process from a standpoint of both space and time. In practice the robot can only maintain one visual snapshot at a time, and must gather whatever information it needs before the next snapshot wipes out the first.

The image of the ball in the visual field at a particular instant may be thought of as a static pattern, and in many cases it is represented as a matrix representing relative brightness. Successive images of the ball are stored with different matrices. A succession of these matrices is a sequential or temporal pattern. A temporal pattern recognizer is presented with these matrices one at a time, and can recognize or reproduce this pattern. This sort of complex visual interaction, in the case of our robot, has yet to be implemented in neural networks even though some work has been done for simpler tasks.

This section describes some of the work done in temporal pattern recognition by neural networks. The patterns recognized are rather simple, such as the numerical sequence "1, 2, 1, 3, 1, 4" or the letter sequence "A, B, C" presented as a sequence of binary matrices. The mechanisms reviewed provide a background and motivation for some of the ideas used in this research.

A pattern associator is a network that, given a static pattern, produces another as output. Input is presented by switching specific input units on, and specific output units will turn on in response to this input, normally by computing a weighted sum of the input unit values. This is a form of associative memory, with the output pattern being associated with the input pattern. The association is stored in the pattern associator by adjusting the weights on the connections between the input and the output nodes. Many of these networks are self–organizing or adaptive networks and the weights are set automatically. One common way to store a particular pattern is to have the pattern associator reproduce the input pattern as its output. A common feature of this type of pattern associator is that a particular pattern can be reproduced as output even if the input is only partially presented or if there is spurious input in the signal. This is referred to as pattern reconstruction. The books by Kohonen [65] and Hinton and Anderson [43] provide a good outline of pattern associators, associative memories, and their relation to content addressable memories.

A temporal pattern recognizer is much like a pattern associator in that it is presented with a *sequence* of input to produce an output. This output is delayed with respect to the presentation of the input, since the associator must wait to accumulate information about the particular temporal pattern. It is this waiting period, or more specifically, what the recognizer does with its previous input, and whether it stores input in delay lines or activation patterns, that is of interest here.

Kohonen [65, 66] presents a simple neural network model for the storage and reproduction of temporal sequences. It resembles a pattern associator with

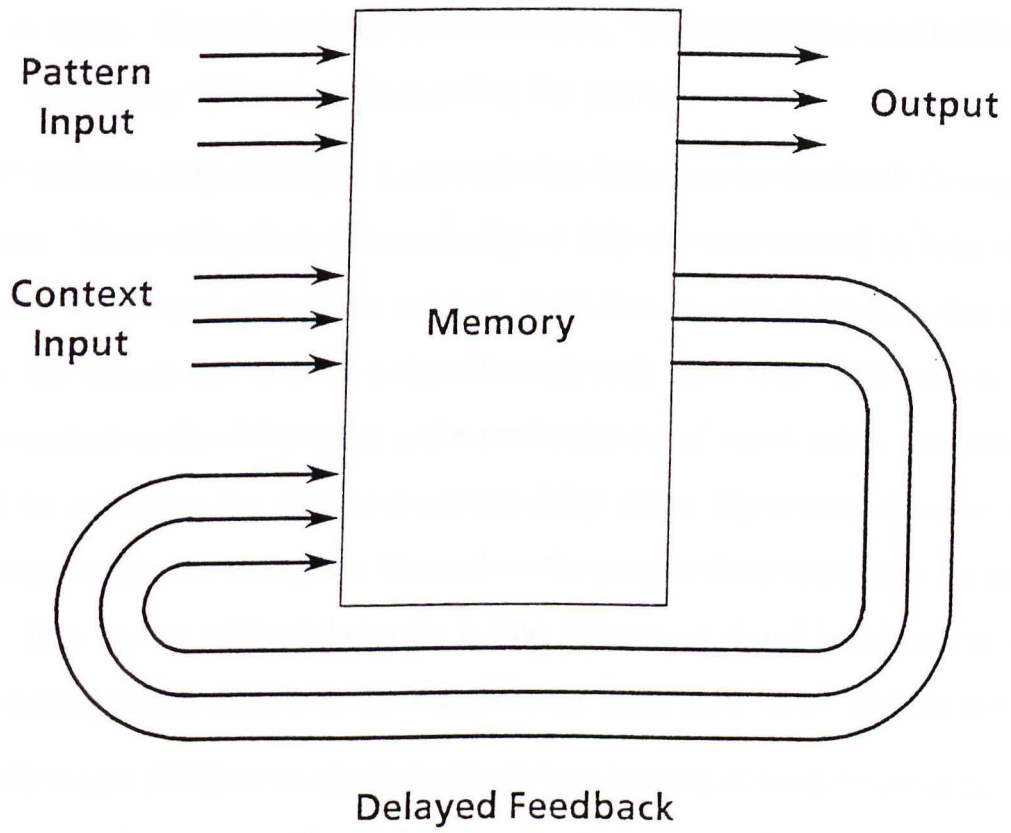


Figure 2.1. Kohonen's sequential pattern recognizing automaton [65]

the addition of a delayed feedback loop (see figure 2.1) in which the output of a pattern associator is delayed and presented as input or context along with the external input. This provides an environment for use in predicting the next item in a sequence. For example, to store the sequence "A, B, C", the "A" is presented and the pattern associator stores it by adapting its internal weights to produce "A" as output, feeding this output back as input. The feedback is presented along with "B" in the next cycle, the weights are set to store "B", and "B" is produced as output. The process is repeated for "C". To reconstruct the pattern, "A" is fed to the network, which produces "A" as output that is fed

back as input. Through pattern reconstruction, "B" is produced and fed back as input, producing "C" and reconstructing the pattern.

Fukushima [38] developed a network that incorporates feedback through delay lines. These delay lines (shown in figure 2.2) are constructed as long chains of nodes. An output unit of the network feeds into its own delay line that propagates the output one cycle at a time through each delay unit. The output units are connected to the delay units and a particular set of input units. Patterns are stored by adjusting the weights from the delay units. By setting these weights the output units can choose for themselves the proper delay necessary for recognition. One feature of the delay units is that the output signal is reduced at each unit, causing recent signals to have more effect than those more distant in time.

Willwacher [129] uses a single feedback loop instead of delay lines for pattern generation and reconstruction. A delay function gates the output of a unit, increasing it to a saturation value and then reducing it to zero. The effect of a particular input is the greatest when the delay function is giving its maximum amplification, with a reduced effect prior to and after the saturation point.

Tank and Hopfield [123] use an idea similar to this delay function to allow time warping, *i.e.,* to recognize sequences that are somehow distorted in the duration between items. Instead of one delay function they use several. Each input detector is gated through several delay functions. The delay functions increase the output of the input detector from zero to a maximum and then back to zero. The functions differ in the amount of time to reach the maximum and decay, varying in a continuous fashion from sharp amplifications to more gradual changes. The recognizer units are connected to the output of each delay function

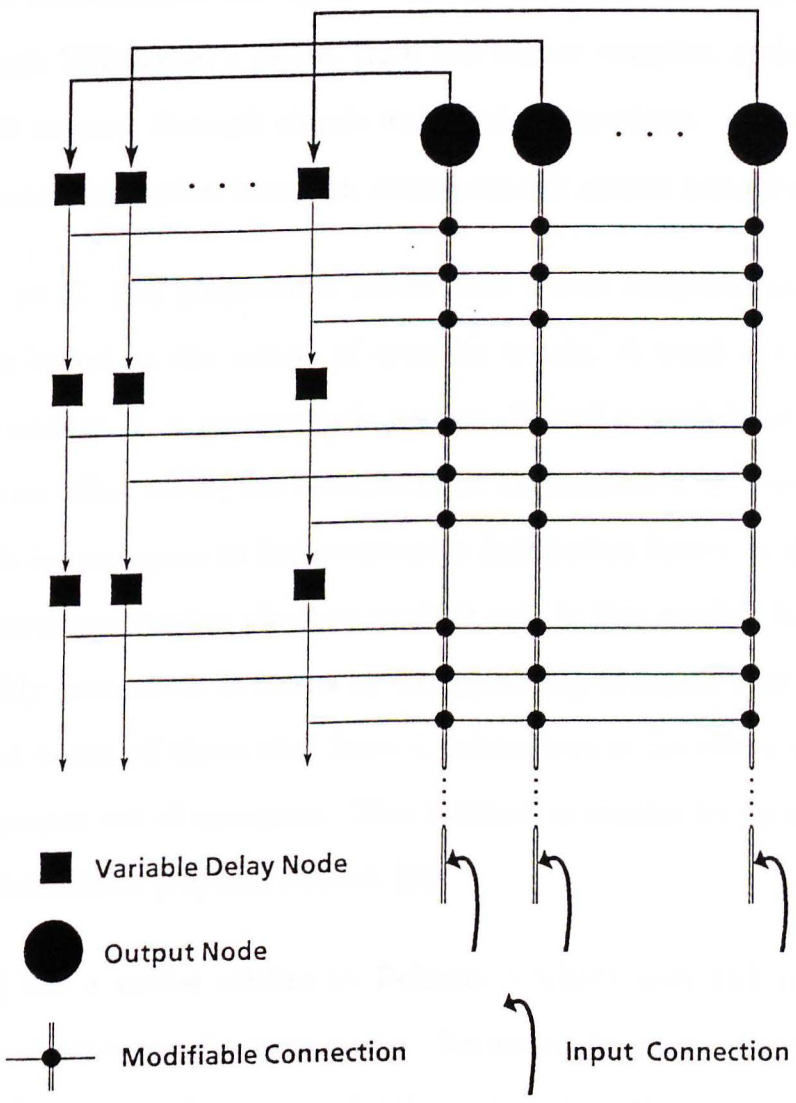


Figure 2.2. Delay lines in Fukushima's sequence detecting model [38]

as well as a simple feedback loop. The effect of this arrangement is to provide a firm anchor at one end of the pattern, allowing more variation in time in the earlier portions of the pattern.

Most of the models above concentrate on classical artificial neural network

behavior. The architectures are regular and the units perform simple computations. Although Willwacher's model [129] has rather complex node dynamics, input was still treated through simple weighted connections. There are other models of sequence detection based on more complex neural behavior.

Dehaene, *et al.* [26] proposed a model that learns temporal sequences by selection. It is based on the notion of synaptic triads. A triad is composed of a presynaptic neuron A, a postsynaptic neuron B, and a modulator neuron C. For A to have an effect on B, the synapse must be facilitated by recent activity of cell C. This is analogous to heterosynaptic facilitation found in the nervous system. Temporal summation plays an explicit role in this model. Initially, the network is highly connected. It learns by strengthening synapses that contribute to activity, and turns off those that have a deleterious or no effect at all, thus selecting the proper set of synapses. This method is similar to programmable networks by McClelland [73] and Pollack [93].

Chun [23] has a model similar to Dehaene's which uses link interactions for gating flow of activation between nodes. Sequence detection is accomplished through link enhancement that primes for the next item in the sequence, and link inhibition that prevents excitation of items not in the sequence. Link interactions also serve to capture durational information of a fixed length.

The methods above concentrate on computer implementations for temporal sequence detection. Neurophysiological studies focus on how the nervous system responds to temporal features or sequences, and provide some support for the direction of the implementations.

Kurogi [67] attempted to describe the behavior of units called P–cells to account for properties of spatio–temporal pattern recognition. The P–cell models the membrane potential and electrodynamics of a cortical pyramidal cell, and although it is too complex for a computational model (and was not intended to be so), it demonstrates some interesting neurodynamic behavior that can be used for temporal pattern recognition. The work shows that proper synchronization of neural signals is required for the cell to fire. This is due to spatial and temporal summation characteristics of the dendritic tree. The excitatory effects of a postsynaptic membrane propagate along the tree, with the rate of propagation determined by the strength of the stimulus. If an excitatory synapse is activated when other excitation reaches that synapse through propagation, its effect is intensified. Strong inhibitory signals can reset the membrane potential of certain regions, canceling the effects of previous excitatory input. Attempts to use the cell in pattern recognition showed that heterosynaptic facilitation may also play a part.

Shepard, *et al.* [103], modeled the behavior of dendritic spines, structures associated with synapses on certain neurons including the pyramidal cell. These findings indicate that synapses on dendritic spines can be facilitated by excitatory synaptic activity on nearby spines, increasing the postsynaptic effect. There is a time course to this facilitation, with the effect decaying over time. In another study, Strehler [116] found evidence of monkey cortical cells that fired only upon receiving a precise, time–coded pattern of synaptic input. They hypothesize that these cells react to precise patterns by the spatial topologies and electrodynamic properties of their dendritic trees.

These studies indicate the important role that non–linear synaptic effects play in sequence detection in neural systems. The spatial and temporal characteristics can be exploited in computational models of sequence detection, and suggest possible extensions and modifications to traditional connectionist ideas to cope with this behavior.

2.2 Visual Motion

One example of temporal sequence detection in biological systems is the detection of motion in a visual field. Cells have been found throughout many visual systems that respond when objects are moved across a receptive field in a specific direction and at a particular speed [47, 48, 132]. These cells detect a series of patterns originating from the receptor cells of the retina, responding with a burst of activity if a pattern is presented in a specific order and within a certain time frame. The presence of direction sensitive cells throughout the visual pathways indicate that this may be one of the representations used for visual information in the cortex [5]. The mechanisms for the detection of visual motion give insights into information processing in the brain and suggest computational methods that can be applied to other problems. This section describes one such mechanism.

2.2.1 A Neural Mechanism for Visual Motion

The lower visual system of vertebrates is composed of cells that translate retinal images into a neural representation that is transmitted from the optic nerve to higher brain regions. Light entering the eyes is focused and projected on the retina. The retina is composed of a layer of receptor cells, the rods and cones, which are activated by photons of light. Horizontal and bipolar cells synapse

with these receptors and send output synapses to the dendritic trees of cells in the retinal ganglion. The axons of the ganglion cells travel along the optic nerve that exits the eye to the rest of the brain. Each cell has an associated receptive field responding to stimuli from roughly circular regions of the retina. One of the reasons these cells have been extensively studied is because their dendritic trees are roughly two dimensional and their receptive fields correspond well with their dendritic arbor.

Barlow and Levick [6] studied retinal ganglion cells in the rabbit that respond to stimuli moving across their receptive fields in a preferred direction, and show no response to motion in the opposite or null direction. They proposed a mechanism based on delayed inhibition, that can veto excitation passing through an AND–NOT gate. Figure 2.3 illustrates this mechanism. Activation of receptor A sends a signal that is passed through veto gate V_1. Activating B sends activation which passes through V_2 and is summed with V_1, and similarly for C, giving the preferred sequence A–B–C. For the sequence C–B–A however, C is activated and sends a delayed inhibitory signal to gate V_2. Receptor B is activated, but veto inhibition blocks it at V_2. Receptor B also sends an inhibitory stimulus to V_1, vetoing the signal from A. The delay is necessary so that inhibition from B arrives at V_1 at the same time as the excitation from A. Delayed inhibition is also believed to be part of mechanisms for directional selectivity in cortex [131] and in the lateral geniculate nucleus [96].

Torre and Poggio [125] attempted to specify the neurophysiology of Barlow and Levick's veto–gates. Their hypothesis involves the electrical circuitry of patches of dendritic membrane. Using circuit equations based on the ionic

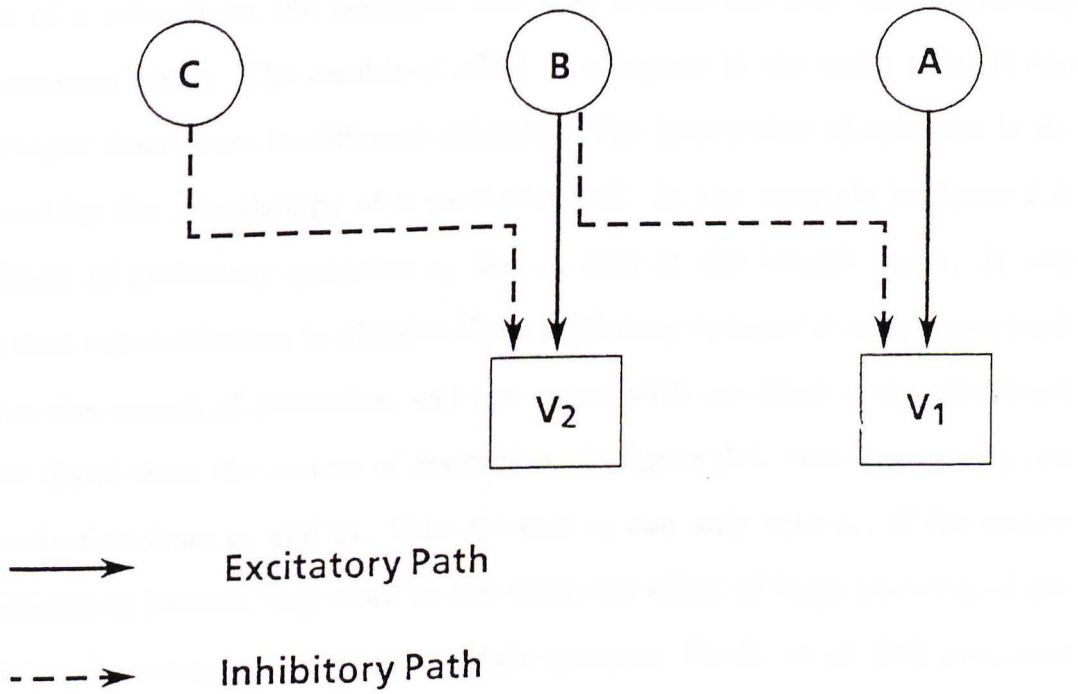


Figure 2.3. Diagram of Barlow and Levick's veto–gate mechanism for motion detection. Nodes V_1 and V_2 will block excitation if the inhibitory path is active.

characteristic of the membrane, they proposed that this veto operation is a post-synaptic phenomenon carried out by closely adjacent synapses. They showed that the veto operation can be implemented by an inhibitory mechanism that prevents excitatory input from having an effect on the membrane potential of the dendritic patch. In this case a small amount of inhibition can cancel a great deal of excitation. The shunting behavior loses effectiveness if the synapses are farther apart. This is an example of non–linear spatial summation, which eventually becomes linear at greater distances.

Koch, Poggio, and Torre [63, 60] refined this model based on detailed circuit analyses of certain cells, called δ cells, of cat retinal ganglion. They defined the

notion of a subunit on the dendritic tree that divides the tree into electrically homogeneous zones. The combined effect of synapses in the same subunit can be stronger than those in different subunits. The interaction of subunits is determined by the morphology of a particular cell. In the example in figure 2.4, the effects of excitatory synapses e_1 and e_2 sum at the branch point. It was found that veto inhibition is effective if the inhibitory synapse is on a direct path between the source of excitation and the soma, with no effect if the inhibition is more distal than the source of excitation. In figure 2.4, veto synapse v_2 can veto excitation from e_1 and e_2. Veto synapse v_1 can only veto e_1. If the source of inhibition is located very close to the soma the effect of large portions of the dendritic arbor may be vetoed by a single synapse. Koch, *et al.* [63] proposed that direction selectivity is implemented by the interaction of dendritic subunits and direct–path inhibition.

Figure 2.5 shows a simplified example of a direction selective cell. In the preferred direction excitation from the rightmost input unit depolarizes the membrane, sending a wave of depolarization flowing downwards towards the soma, which passes the veto synapse before it is activated. Excitation from the next input unit flows and is added to the effect of the first. This continues until sufficient depolarization arrives at the soma to trigger spiking behavior. In the null direction depolarization flows from the excitatory effect of the leftmost input unit, but excitation from the next input unit is blocked by the veto synapse activated by the previous unit, preventing sufficient depolarization from reaching the soma.

The temporal properties of the synapses are important as well. Koch, *et*

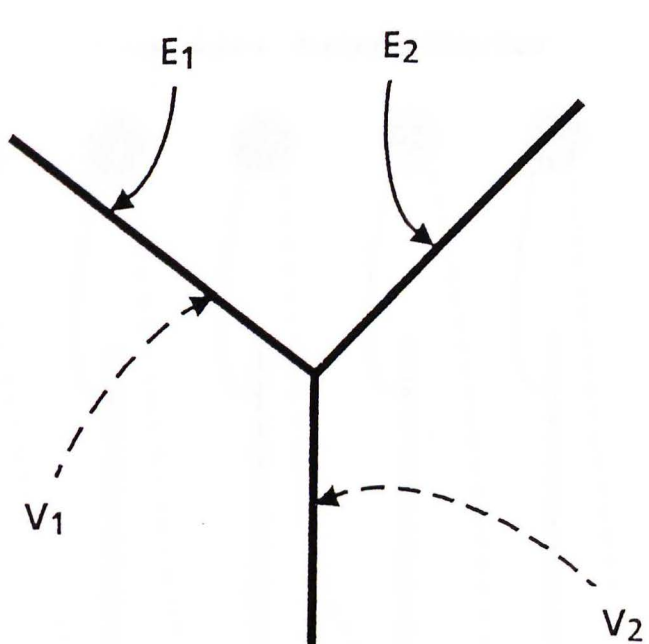


Figure 2.4. Diagram showing spatial integration of excitatory and veto connections. Veto synapse V_1 can only block excitation from excitatory synapse E_1, while V_2 can block both excitatory synapses.

al. [62], have shown that veto inhibition is most effective if it arrives slightly before the wave of depolarization. If it arrives much before the excitation it may have decayed too much to have an effect. If it arrives after the excitation has passed it cannot block the depolarization.

2.2.2 Implications

In the case of a bar moving through the visual field the number of active synaptic inputs to the retinal ganglion cell will be quite large. Direction selectivity in this case requires a highly branched, passive dendritic morphology [63, 62]. The presence of this type of structure has been confirmed to be the case in the retinal

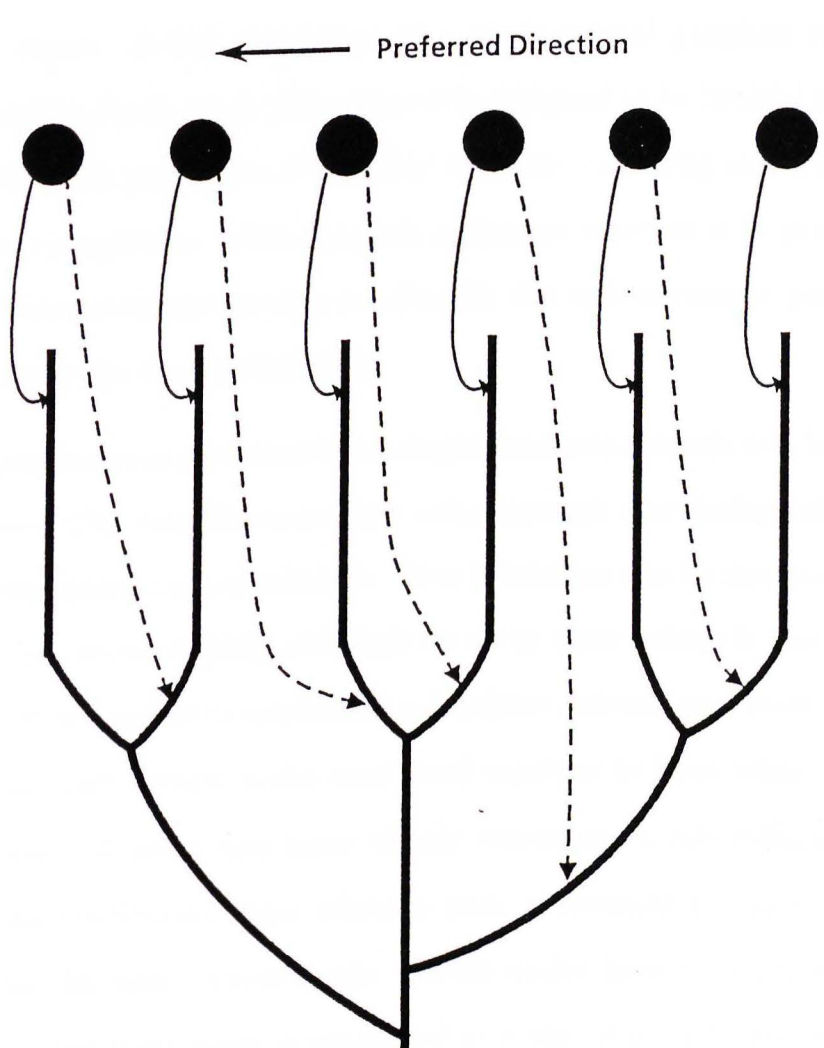


Figure 2.5. Diagram of a one degree motion detector that responds to input moving right to left. Solid lines are excitatory paths and dashed lines are inhibitory.

ganglion through detailed neuroanatomical studies [3]. The spatial and temporal characteristics of this synaptic integration make it impossible for a single node of a connectionist model to capture this behavior through a simple linear weighted

sum of its input. A full simulation of a single retinal ganglion cell has been accomplished by Koch, *et al.* [61]. Since it is designed to be faithful to the known neurophysiological properties it is highly complex, modeling elaborate electrical and chemical properties. This research addresses whether it is possible to use the tools of connectionist models to simplify the neurodynamic properties and yet still capture the desired behavior.

The computations performed in a single dendritic subunit are, by definition, homogeneous. The complications that arise through morphological constraints stem from the interaction of subunits. Veto inhibition can be implemented quite easily as a veto network [108], although its all or none nature is somewhat non–traditional. It is possible to implement a dendritic subunit as a node of a connectionist model, with several nodes connected together to form what is analogous to a single cell. A node now more closely represents a sub–cellular structure, instead of the traditional sense where a node represents a single neuron or a group of cells. In many connectionist models nodes have a rich interconnectivity. In some cases every node is connected to every other with enormous fan–in and fan–out properties. In the veto network, the fan–in is small since subunits normally have few synapses, and there is a single output to the next node on the branch. Further implementational details are discussed in Chapter 3.

The incorporation of veto inhibition and subunits in dendritic integration may be useful as a general mechanism for neural–like computation. It has been proposed as the mechanism for directional selectivity in higher brain areas such as the lateral geniculate nucleus of the cat [60]. The veto inhibitory synapses in retinal ganglion cells are thought to be bicuculline sensitive γ-amino butyric acid

receptors, called $GABA_a$ synapses. These synapses appear to exist in certain directionally sensitive visual cortex cells [112]. Thus veto–type inhibition may be used in other parts of the brain.

Even if it is not found to be used throughout the brain, this type of dendritic computation might be applied to other problems. It has already been suggested as a possible mechanism for binocular disparity [61]. The question remains as to whether it can be useful in other tasks not directly related to vision.

2.3 Auditory Processing and Acoustic Motion

The speech signal contains both dynamic and steady–state events. The dynamic and static nature of the speech signal is seen in figure 2.6, where the phrase "the bad scope" is shown in a wide-band spectrogram with time on the x axis and frequency on the y axis. If this dynamic behavior is useful in speech perception, one would expect to find parts of the auditory nervous system that are sensitive to spectral changes in the speech signal. Research has shown that this may be the case. The question thus is whether units sensitive to spectral change can be constructed, and if they can be useful in speech understanding systems. Before addressing this it is necessary to discuss the ways in which the ear processes acoustic signals.

2.3.1 The Auditory System

The speech signal is transmitted by pressure waves traveling through the air. The acoustic pressure wave is transmitted by mechanical motion of the bones of the middle ear to the oval window of the cochlea which comprises the inner ear. The basilar membrane in the cochlea converts vibrations into neuronal events

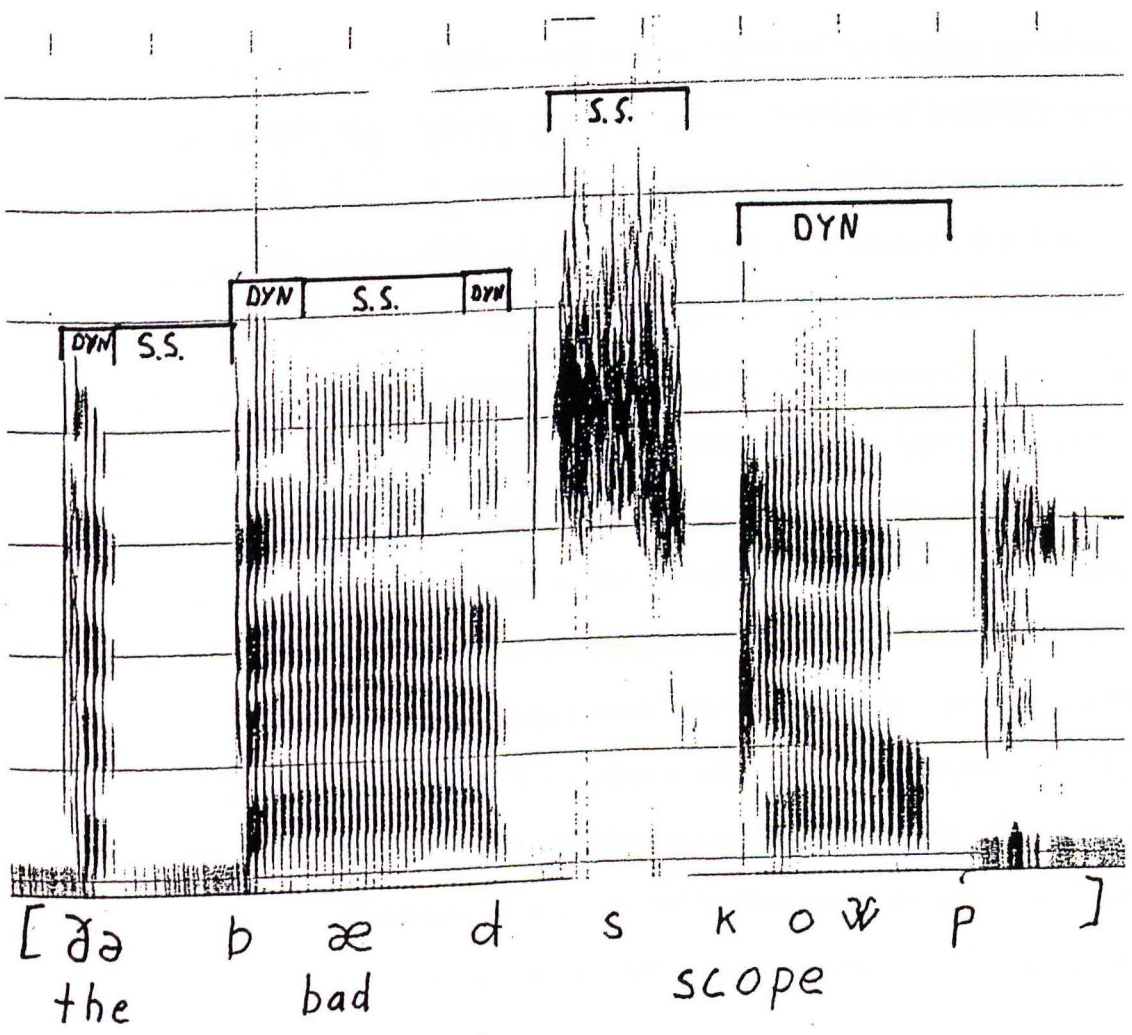


Figure 2.6. Wide band spectrograph of the phrase "The bad scope," showing dynamic and steady–state regions. The horizontal axis is time and the vertical axis is frequency.

by mechanical stimulation of hair cells on the membrane. Stimulation of specific hair cells excites corresponding fibers of the auditory nerve. Transmission of this excitation is sharpened by the effects of lateral inhibition from interneurons at the hair cells and by mechanical filtering on the basilar membrane [106, 100]. The auditory nerve transmits acoustic information to the rest of the central nervous system.

A pure tone excites hair cells in a particular region of the basilar membrane that stimulate specific fibers of the auditory nerve. Studies of auditory nerve fibers show that many respond maximally to tones of a specific frequency, called the characteristic frequency (CF) of the fiber. Output is reduced for frequencies greater or less than the CF due partly to lateral inhibition. This behavior is reflected in the tonotopic representations found in the auditory cortex. The recordings of evoked potentials in the auditory cortex show a type of frequency map with adjacent cells having similar CF's. This tonotopic representation resembles the retinotopic representation in the visual cortex. The auditory system performs something similar to a Fourier separation of complex sounds into their respective pure frequencies. A complex tone consisting of 200, 600, and 1800 Hz will excite nerve cells whose CF's are near those particular frequencies. It is unclear whether all frequency separation takes place in the ear itself. Although it has been believed since Helmholz to occur in the basilar membrane, it is possible that some frequency selectivity may occur higher in the auditory nervous system before the level of the cortex [106].

The representation of complex sounds in the lower auditory system seems to be more complicated than that for pure tones. One property commonly found in the responses to complex tones is the notion of critical bands [100]. Component pure tones normally interact one way when they are within the critical band, and show different properties when they are separated on the frequency scale by at least one critical bandwidth. For example, if the frequency separation of two pure tones is increased, there is no perceived increase in loudness until the frequency separation or bandwidth exceeds the critical band. After this point

loudness increases with bandwidth.

Critical bands affect a wide variety of observations of the auditory system. Difference limens, the just noticeable difference between pure tones presented one after another, are about 0.1 Hz in the frequency range of speech. If the pitch of the sound is determined by two simultaneous tones close in frequency, the difference limens are much larger and are related to the width of the critical band. This implies that the frequency resolution of the auditory system may be limited by the width of the critical bands.

The effect of changes of an acoustic signal is not clear. For example, changes in intensity can be coded by changes in firing rates, the number of fibers excited, or some other mechanism. In addition, many cells in the auditory nervous system show phase–locked spiking behavior. This means that the cell will emit a spike at a particular point on the wave of a signal. For example, a cell may spike only when the sinusoidal wave of a pure tone signal begins its descent. It is not known how this information is used as a representation in the auditory system, although it can provide precise timing information for acoustic events.

Some cells in the auditory nervous system show sensitivity to signals of changing frequencies, analogous to the behavior of visual motion detectors. In a study of the effects of changing frequencies, Whitfield and Evans [127] investigated the response properties of neurons in cat auditory cortex. After mapping the characteristic frequencies of a population of cells by their response to pure tones, they found that most of these cells responded stronger to signals of changing frequency. Many cells that were not affected by steady tones responded to changing frequencies instead. Those cells responding to frequency changes would

often show a preferred direction, either to rising or falling frequencies. Mendelson and Cyander [79], also looking at cat auditory cortex, found that cells showed a sensitivity to the rate of change as well as direction of FM sweeps. These cells are tuned to frequency changes of particular slopes in particular directions.

The lower auditory structures show sensitivity to frequency changes in the signal as well. Møller [84] reviewed results of recording studies of various parts of the lower auditory nervous system. In the auditory nerve, fibers respond to frequency changes near their characteristic frequency, but show no direction or rate preference [84, 105]. In the cochlear nucleus it was also found that cells show a preference for particular rates of change near the characteristic frequency, but show no direction selectivity [85, 84]. Britt and Starr [18], on the other hand, found that up to fifty percent of cochlear nucleus cells responded to changing frequencies, with some showing direction selectivity. Direction as well as rate selective cells were also found in the inferior colliculus [84].

The above discussion shows that as a first a approximation the ear is a good microphone and spectrum analyzer. The cells of the auditory nerve exhibit minimum sensitivity to changes in frequency, and cells higher in the auditory pathway begin to show more response specificity to rates and directions of frequency changes.

2.3.2 Speech Sounds

According to the acoustic theory of speech production [33], the spectral qualities of the speech signal are determined by sound generation techniques and the shape of the vocal tract. Different sounds are made by affecting the airflow through the

oral cavity and upper respiratory tract during the rapid motor gestures of the tongue, lips, soft palate, and larynx that are characteristic of human speech. For example, the "sss" sound of the fricative [s] is caused by by directing a jet of air at the teeth at high speeds, and a [p] sound is made by the temporary blocking of the airflow at the lips. Voiced sounds, including the vowels, are composed of a fundamental frequency produced by the vocal chords, and various harmonics whose intensity is affected by the positions of structures including the tongue and the lips. By humming a single note and changing the position of the tongue and lips, a speaker can hear the effects on the amplitude of harmonic or resonant frequencies.

A sound spectrogram separates frequency regions of a complex sound using a wide–band analysis filter and displays their intensities for a short time window, typically 3 ms (see figs. 2.6 and 2.7 for examples). In such a display a formant can be seen as a peak in a frequency *vs.* intensity graph, or a dark band of energy in a wide–band spectrogram consisting of a series of such spectral slices. These dark bands are concentrations of energy produced by several neighboring harmonics of the fundamental that lie near the resonant frequency of the cavity. As the shape of the vocal tract changes, formants rise and fall across the spectrograph. The formants are often referred to by number, with F1 being the lowest frequency, F2, next highest, and so on. Normally little attention is paid to formants higher than F3, since they represent idiosyncratic, not linguistic variables.

Figure 2.7 is an example wide–band spectrogram of the utterance "the bad scope." Formants are seen during periods of voicing for vowels. Formant transitions are best seen just before or just after the closure for [b], which is seen as a

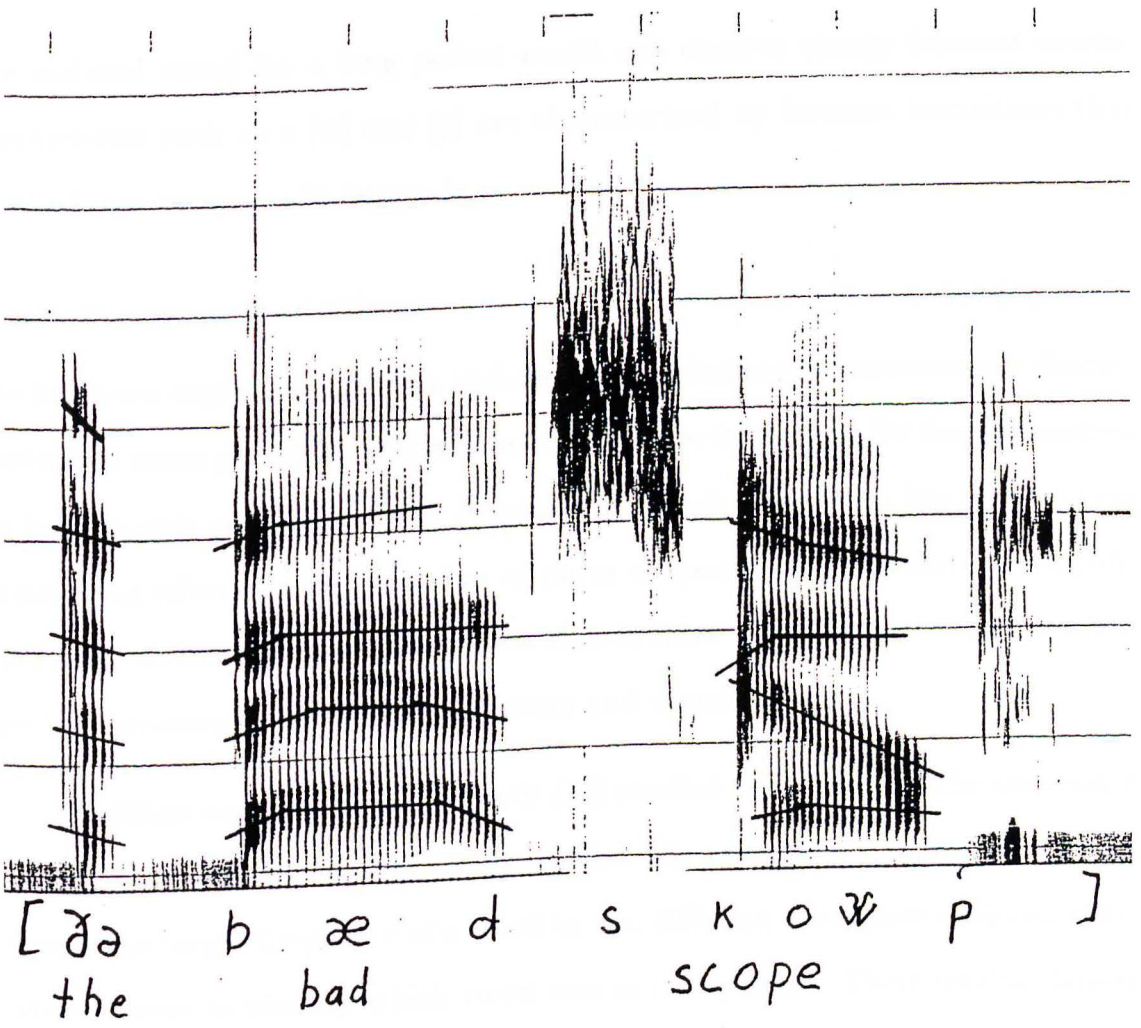


Figure 2.7. Wide band spectrograph of the phrase "The bad scope." Formant center frequencies are shown by straight lines.

silent portion before the vowel. Transitions of formants F2 and F3 can provide some information for phonemes with the same place of articulation. For example, the F2 and F3 transitions for the stop [d] are similar to those for the fricative [s], since both are alveolar consonants with the same place of articulation. The fricative [s] is indicated by the dark patch of high frequency noise. Both the [k] and the [p] show distinctive noise bursts at their release. Only if one holds

an isolated vowel for a long period would one observe steady formant tracks. Semivowels such as a [w] and [y] are characterized by formant transitions that make large sweeps as in "scope."

2.3.3 Formant Transitions

As has been explained, the rate and direction of formant transitions are characteristic of some phonetic units and hence might be important for the recognition of both vowels and consonants. Research in speech perception has shown these transitions affect the identification of parts of speech. Experiments in identification of natural and synthetic speech signals show that formant transitions are used by listeners to recognize consonants and vowels.

Lindblom and Studdert–Kennedy [70] studied the effects of the consonant context in vowel recognition in CVC syllables. They synthesized CVC syllables by varying the target frequency of a vowel in two different consonant environments. Listeners were to identify which vowel was in the syllable. Their results showed that the same vowel frequency was not perceived as the same in different contexts. Subjects would identify the same target frequency as different vowels depending on their surrounding consonants. It was suggested that perceptual information for the vowel was contained in the formant transitions.

Strange, *et al.* [115] used natural speech in CVC vowel identification experiments. They found that subjects showed up to three times better performance in identifying a vowel in a CVC syllable than one produced in isolation. Another study found that more contextual information led to better vowel recognition. Vowel recognition was highest in CVC syllables, followed by CV and VC syllables,

followed by isolated vowels [113]. They proposed that there was considerable information for vowel identification in the formant transitions and the length of the vowel segment. To confirm this, they performed masking studies on naturally produced syllables [114]. CVC syllables were divided into three segments and parts of the signal were masked. In one case the middle of the signal containing the vowel was attenuated to silence, corresponding to a "vowel–less" syllable. The duration of the silence was also varied. In a −V− syllable the initial and final consonant segments were masked and the length of the vowel was varied. Vowel identification scores were not significantly different in the C−C condition compared to the CVC control. Listeners correctly identified the vowel despite the fact that the vowel was "missing" from the signal. The duration information showed no significant effects, except in the −V− case where listeners seemed to use accurate duration information to disambiguate the vowels. This shows that information contained in the formant transitions is sufficient for vowel identification.

The spectra of the voiced stop consonants [b], [d], and [g] in a CV environment are characterized by a burst of noise followed by formant transitions to the steady–state vowel. The consonants differ in their place of articulation, that is, where the airflow is stopped in the vocal tract, with a [b] being bilabial, a [d] alveolar (the ridge just behind the upper teeth), and a [g] velar (the soft palate). Formant transitions can be used to identify stop consonants.

Delattre, *et al.* [28], in an identification study of synthetically produced CV stop consonants, claimed that the F2 transitions would determine the place of articulation with the transition pointing to a locus specific for each consonant.

In the case of [d] this locus was the same no matter what the vowel is. The phonemes [b] and [g] showed less reliable loci, in some cases more than one. Nevertheless they claimed that the F2 transition, preceded by an appropriate silent interval, points back to a locus determining the place of articulation, and hence can determine the identity of the stop consonant.

Stevens and Klatt [111] have shown that a rapid F1 transition is present in voiced stops and absent in voiceless stops (*i.e.*, [p], [t], or [k]). They claim that detection of the F1 transition is the key to the voiced–voiceless distinction, and that detection of this rapid spectral change is an innate property of the auditory system. Evidence for this comes from a study of infants that were shown to be able to distinguish between voiced and voiceless categories, but could not distinguish stops within a category [31].

The role of formant transitions in the perception of stop consonants is somewhat controversial. The argument centers on two notions: whether there are static *vs.* dynamic detectors and whether there must be a dedicated unit that fires everytime a particular phoneme occurs. Stevens assumes that there should be certain cues that can be used to distinguish place of articulation no matter what the vocalic context is. Blumstein and Stevens have argued in studies of synthetic and natural stop consonants that while formant transitions are not an invariant cue to place of articulation, the simple spectrum shape averaged over about 20 ms after the release of the consonant is a good determinant of place of articulation in many contexts [110, 15, 14]. Thus when spectral properties of a noisy burst are added to the synthetic formant signal, recognition is significantly improved over transition–only stimuli. They claimed that formant transitions

are useful only as secondary cues in a vowel environment [110, 15] and are less useful to distinguish place of articulation, since burst information from the first 40 ms of the signal almost always leads to correct recognition [14].

More recent research, while supporting a role for spectral shape, has emphasized that dynamic information is still essential for stop place identification. For example, Kewley–Port [52] measured the formant transitions from 3 voiced stop consonants ([b], [d], [g]) paired with eight vowels. Only weak evidence was found for context invariant cues for place of articulation of stop consonants. The formant measurements did support the claim that F2 transitions are distincitive sources of place information in the contexts of most of the vowels studied. If F2 and F3 onset frequencies were tracked into a F2 $\times$ F3 space for each vowel, context–dependent place of articulation could be achieved statistically.

Further evidence of the usefulness of transitions in CV recognition is provided by Pols and Schouten, in a study of Dutch stop consonants, who showed that the information in CV onsets claimed to be invariant across vowels by Stevens is often masked by the following environment and that formant transitions are necessary. They claim that formant transitions are primary cues to the identity of stop consonants [94]. Suomi found that many context–invariant cues are, in fact, context dependent, and that this context dependence does not necessarily imply that a coarticulated segment cannot be identified until the entire context is heard; there is sufficient information contained at the onset of the signal [117]. This is similar to the CVC phenomenon described above. Furthermore, if one treats the syllable as the basic unit of speech perception, formant transitions can serve to specify place–of–articulation [52].

Formant transitions can be useful in identification of stop consonants. Listeners can identify both consonants and vowels with only transition information present in the signal. Formant transitions form a bridge between the consonant and vowel of the syllable, and must be context dependent by their very nature. They are, of course, not the only information available, and alone may not allow perfect recognition. Nevertheless, they can be useful in recognition of certain parts of speech. In light of the above discussion, it becomes clear that while spectral shape of the burst may be useful place–of–articulation information, formant transitions are also clearly useful in this task. No doubt a combination of features best serves the perception and identification of the segmental units of speech [110, 15, 14]. What emerges is the importance of treating the signal as a series of dynamic events [53, 51, 54].

2.4 Summary

The neurophysiological evidence discussed above shows that the nervous system is predisposed to detect frequency transitions. Gardner [39] has shown psychophysical evidence that there are, in fact, specific channels in the auditory system for detection of the direction of frequency transitions. It seems useful to try to detect formant transitions as they might be used in speech perception. This may also be useful in understanding the nature of dynamic computations in the nervous system.

Formant transitions are analogous to one degree visual motion. The method for detecting them depends on the way the speech signal is processed. Many classical speech recognition systems receive an utterance all at once and begin

processing after the end of the utterance is stored [32, 71, 27, 21]. In these systems the speech signal is treated as a static phenomenon with a long duration window. Many connectionist models of speech recognition also treat the speech signal in a static fashion [45, 30], with the Tank and Hopfield [122] and the Watrous and Shastri [126] models being some of the exceptions. The TRACE model [74] appears at first to be a dynamic model, since the signal is presented one time cycle at a time. Each time cycle is preserved, however, and at the end of the presentation the system can proceed with the entire utterance at its disposal. This is accomplished by allowing each input feature node, currently twenty–seven per segment, to span six time slices. The nodes are copied for the entire length of the utterance, overlapping each other's temporal window. Static and dynamic models for speech recognition can be distinguished by the length of the temporal window available for processing. In the Tank and Hopfield model and the Watrous and Shastri model the temporal window is short, on the order of 3–10 ms. These can be called dynamic models. In the TRACE model, the temporal window covers the entire length of the utterance, and thus is an example of a static model.

It is unlikely that the auditory system can afford to reproduce its structure for each temporal segment of the signal. The output of the cochlea and auditory nerve represents the state of the system at one precise instant. As time progresses the state changes. The auditory system must process the signal dynamically or as a pipeline, much as an assembly line operates, with raw input at one end, and a processed signal appearing at the other. Speech sounds must be represented in various ways and at different levels depending on where they are in the pipeline.

Speech recognition then becomes a problem of temporal sequence detection, with the speech signal comprising a dynamic sequence of tokens at various levels of representation.

Formant transition information may be one such representation. The evidence presented above indicates that the rate and direction of formant transitions are useful in speech perception and suggest that the auditory system is equipped to capture this data. If the speech signal is treated statically the detection of the slopes of formant transitions is rather easy. A more realistic scenario is that the signal is to be treated as a stream, with input to a perceptual system representing the state of affairs at only one particular instant, disappearing when the next time cycle is presented. The detection of formant transitions then becomes complicated, since the effects of previous input must be evaluated. Fortunately a method is available which can accomplish this task: the veto network from visual motion detection. The next chapter describes how a veto network is used in the detection of formant transitions as well as in higher processing levels in the identification of CV syllables.

3. Project Description

The previous chapter showed that the auditory nervous system is sensitive to stimuli of changing frequency and that formant transitions can provide useful information for the perception of coarticulated speech. To use this transition information, it is necessary to recognize both the direction and rate of the acoustic motion. The mechanism demonstrated for visual motion detection can be applied to auditory stimuli using veto inhibition to tune acoustic motion detectors to a specific direction and rate of formant motion.

It was shown in Chapter 2 that formant transitions can provide clues to identification in a consonant–vowel environment. Information about these transitions may be combined with other properties from the speech signal to provide reliable recognition of these consonants and the syllables that contain them. There is no doubt that it is useful for a high performance speech recognition system to detect

properties of formant transitions.

This chapter describes a connectionist model for the detection of formant transitions and shows how the system uses this transition information in the recognition of CV stop consonant syllables. The first section gives an introduction and overview of the system followed by a section on the data used and how it is prepared for input. The next section shows how the mechanism for visual motion is implemented in the construction of formant transition detectors. The final section describes how the information from these motion detectors is used for syllable recognition, including a discussion of the adaptive learning method used as well as how another veto network is used to fine tune the recognition process.

3.1 System Overview

SYREN is designed to recognize CV stop consonant syllables using formant transition data from an experiment by Kewley–Port [52]. There are twenty–four syllables in all, with three voiced stop consonants /b, d, g/ paired with each of eight vowels /ii, ey, ih, eh, ae, ah, ou, uu/ (as in *beat, bait, bit, bet, bat, bottle, boat, boot,* respectively). For reasons of clarity and typesetting the phonemes are presented in this notation rather than using a phonetic alphabet. The syllables were produced by a single male speaker.

Input to the system consists of formant centers constructed from average formant data [52] and the raw data, provided by Kewley-Port [55], that were used to compute the averages. The input is presented to a three–phased network for the recognition of the syllables. As seen in figure 3.1, the first phase is

a subnetwork that detects the direction and rate of change of the first three formants. Veto inhibition and characteristics of local dendritic computation are used to construct detectors for various slopes and directions of spectral change. The detectors are constructed by hand and are tuned to six different rates of change in rising and falling directions as well as for formants remaining in a steady–state condition for a period of time [108].

The output of the detectors is used as input for an adaptive network. This is a single layer network containing twenty–four nodes, one for each syllable in the data corpus. Since there are a large number of input connections it is impossible to set the weights of the nodes by hand, so a learning algorithm is used to allow the nodes to set their own weights. The nature of this algorithm is discussed below.

The nodes of the adaptive recognition network does not do a perfect job of setting the weights and their output must be further processed to achieve good identification. Output from the adaptive network is used as input to a veto network for the final recognition process. It too contains twenty–four output nodes, each for a particular syllable, that serve as the output of the entire system. These nodes are connected to the adaptive network through delay lines and veto connections.

In its current implementation the system consists of three separate computer programs due to computational requirements. A fourth program is used to prepare the data for execution. Each subnetwork is itself a separate program. Implementational details and the source code of SYREN may be found elsewhere [109].

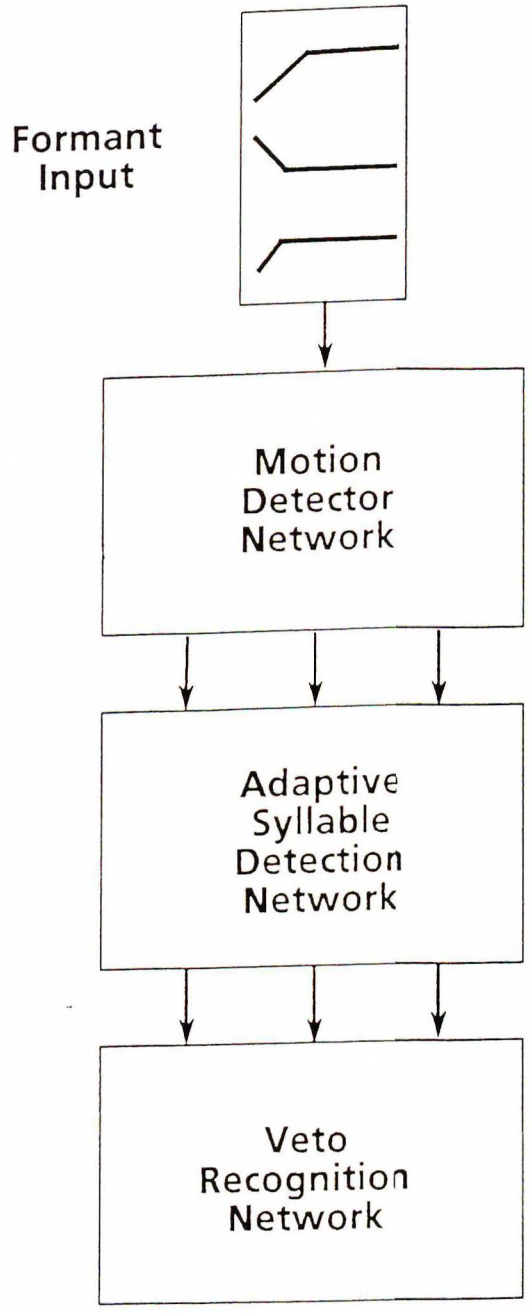


Figure 3.1. Outline of the subnetworks and flow of information in SYREN

Input is presented to the motion detector network sequentially in a stream–like manner, one time slice at a time. Formant centers are received by the network through input nodes, with each node having a non–overlapping receptive field of a particular frequency. An input node is activated if there is a formant center at its characteristic frequency at that time slice. If there is no track at that frequency at the next time slice, as is the case of a rapid formant transition, the node will be deactivated. No record of previous time slices is kept by the input nodes.

Before explaining the operation of the network, the stimuli will be described in greater detail. Two experiments were performed that differed in the training and the testing data. In the first experiment the average formant transition data [52] are used for both training and testing. The network is trained and tested on a single repetition of each syllable. In the second experiment combinations of the raw repetitions of each syllable and the average data are used for training and testing.

3.2 Data Preparation and Presentation

The formant tracks used in the network are derived from formant trajectories and steady–state vowel frequencies provided by Kewley–Port [55]. This information was used in an analysis of formant transitions of stop consonants [52]. Five repetitions of each of 24 test syllables consisting of the voiced stops /b, d, g/ paired with each of eight vowels /ii, ey, ih, eh, ae, ah, ou, uu/ were analyzed. The syllables were embedded in a test sentence "Teddy said CV" recorded by one male speaker. The first 95 ms of the CV portion of the utterance was analyzed.

Formant	Onset of Voicing (ms.)	Onset Frequency (Hz.)	Transition Duration (ms.)	Target Frequency (Hz.)	95 ms. Frequency (Hz.)
F1	9	392	38	669	710
F2	9	1661	65	1166	1154
F3	9	2628	30	2523	2522

Table 3.1. Data used to construct the input matrix for the syllable "dah".

Formant frequencies were measured by eye from sound spectrograms and formant trajectories were calculated. Transitions were defined as starting from the onset of voicing to a steady–state vowel region. Each formant trajectory was simulated with two to four straight lines from the voicing onset [52].

Five sets of raw syllable trajectory data as well as one set of average data from Kewley–Port [52] are used as input to the system. The formant tracks of each of the first three formants in the syllable are specified by a starting frequency, time to end of transition, frequency at the end of the transition, and a frequency at 95 ms, as seen in table 3.1. From these data tables the formant tracks are reconstructed by interpolation for presentation to the network. These tracks are stored in a 200 × 50 bit matrix as shown in figure 3.2. Each row of the matrix represents a small frequency range and each column a time slice. The total frequency range is from 0 to 4000 Hz, with each row representing 20 Hz of the signal. Each column represents a 5 ms time slice. The formant positions at any time slice are represented by an "on" value in the row corresponding to that formant's frequency. The position of the formants at any particular time slice is determined by interpolating from the trajectory data. The 95 ms steady–state frequency was assumed to be the final target of the transition.

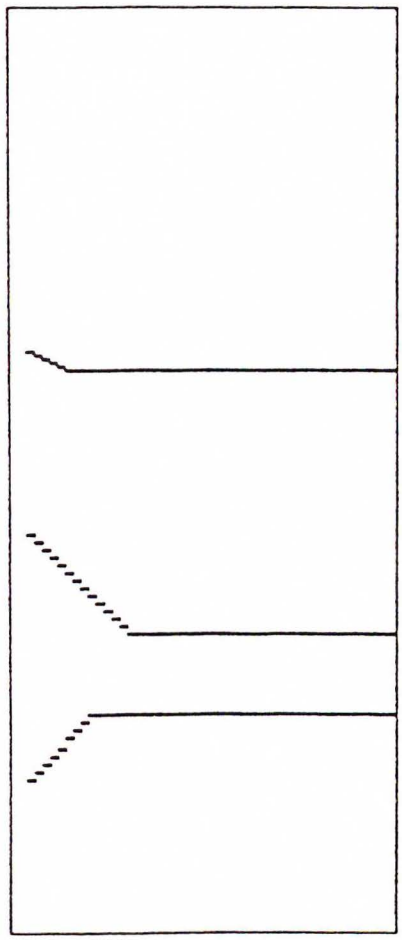


Figure 3.2. Input matrix for syllable "dah". Horizontal axis is time in 5 ms slices. Vertical axis is frequency in 20 Hz units.

The matrix containing the formant centers is presented to the 200 input nodes of the syllable detection network, with each node corresponding to a row of the formant track matrix. Each input node has a receptive field of 20 Hz, and is assigned a value of "1" if there is a formant track within its receptive field at the current time slice, and is assigned a "0" otherwise. The columns of the matrix are presented one at a time, from left to right, with the input nodes

updated at every time slice. These nodes provide excitatory input to the motion detector network and the veto nodes.

There are 200 veto nodes that provide veto inhibition to the motion detector network. Each node is connected to a single input node and is activated when its corresponding input node is active. Its activation is determined by the transfer function

$$a_i^{t+1} = a_i^t(1-\delta) + n_i^t(1-a_i^t),$$

where a_i^t is the activation of veto node i at time t, δ is a decay constant, and n_i^t is the value of input node i. Time steps in the motion detector network correspond to update cycles rather than the 5 ms time slices. Computational behavior of the detector network was found to be optimal if there are seven update cycles per time slice. The activation value quickly rises to its maximum when the input node is on and decreases exponentially to zero when input is turned off. This decay provides a limited memory trace of input node activity for a few time slices, permitting veto inhibition to be used during a few time slices following activation. These nodes provide the delayed veto inhibition required by Barlow and Levick's model [6].

3.3 The Motion Detectors

There are nineteen different motion detectors. One captures steady–state frequencies while the rest are tuned to particular transitions. A detector's preferred transition is determined by its architecture and parameter values. Each of the nine different detectors is tuned to a transition of a specific rate with a mirror image detecting the same transition in the opposite direction. The transitions in

figure 3.3(a-c) have one type of detector assigned while the transitions in figure 3.3(d-f) have two detectors that respond differently depending on how close a transition is to its preferred slope. The reason for this is explained later. The detectors have small receptive fields in the frequency domain corresponding to the number of rows of the transitions in figure 3.3. These receptive fields vary in size depending on the detector's preferred slope, with faster rates requiring larger receptive fields.

3.3.1 Motion Detector Mechanisms

A detector unit is a small network of nodes tuned to a transition of a particular slope and direction. The nodes form a branched architecture analogous to a dendritic tree of a nerve cell. These branches meet at a single output node called an S–node. Some nodes on the branches have excitatory connections to input nodes that make up that detector's receptive field. Each receptive field is determined experimentally and is the minimum size necessary to elicit the desired response from the unit. Branch nodes may also be connected to veto nodes. Veto inhibition completely deactivates a node and is used for direction selectivity as well as for tuning some units to faster slopes.

Two such detectors are shown in figure 3.4. They are tuned to transitions of the same rate but in opposite directions. Their architectures differ only in the pattern of veto connections that determine the direction preference. In the absence of veto connections the detectors would respond to transitions of their preferred rate in either direction, since direction selectivity is accomplished by veto connections. Rate sensitivity is achieved through parametric adjustments.

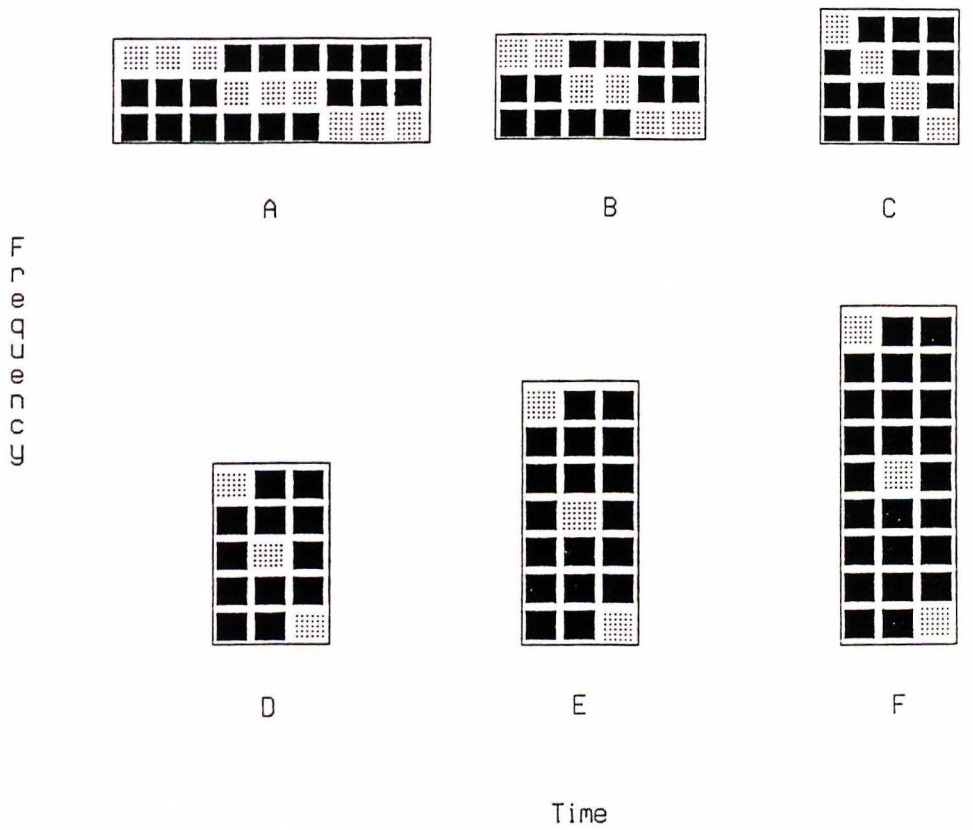


Figure 3.3. These are the transitions identified by the motion detectors. Each column is a 5 ms time slice. A lightly shaded square indicates the presence of a formant center at the frequency corresponding to that row at that time slice. A motion detector is tuned to identify one of these transitions.

Branch nodes may have both excitatory and veto inputs. Excitatory activation is computed using a transfer function similar to those used by McClelland and Rumelhart [76, 99] and Grossberg [40]. The equation for excitatory activation e_i,

$$e_i^{t+1} = e_i^t(1 - \delta) + \text{net}_i^t(1 - e_i^t),$$

is the same as that used in the veto nodes with the exception of net_i^t term, which

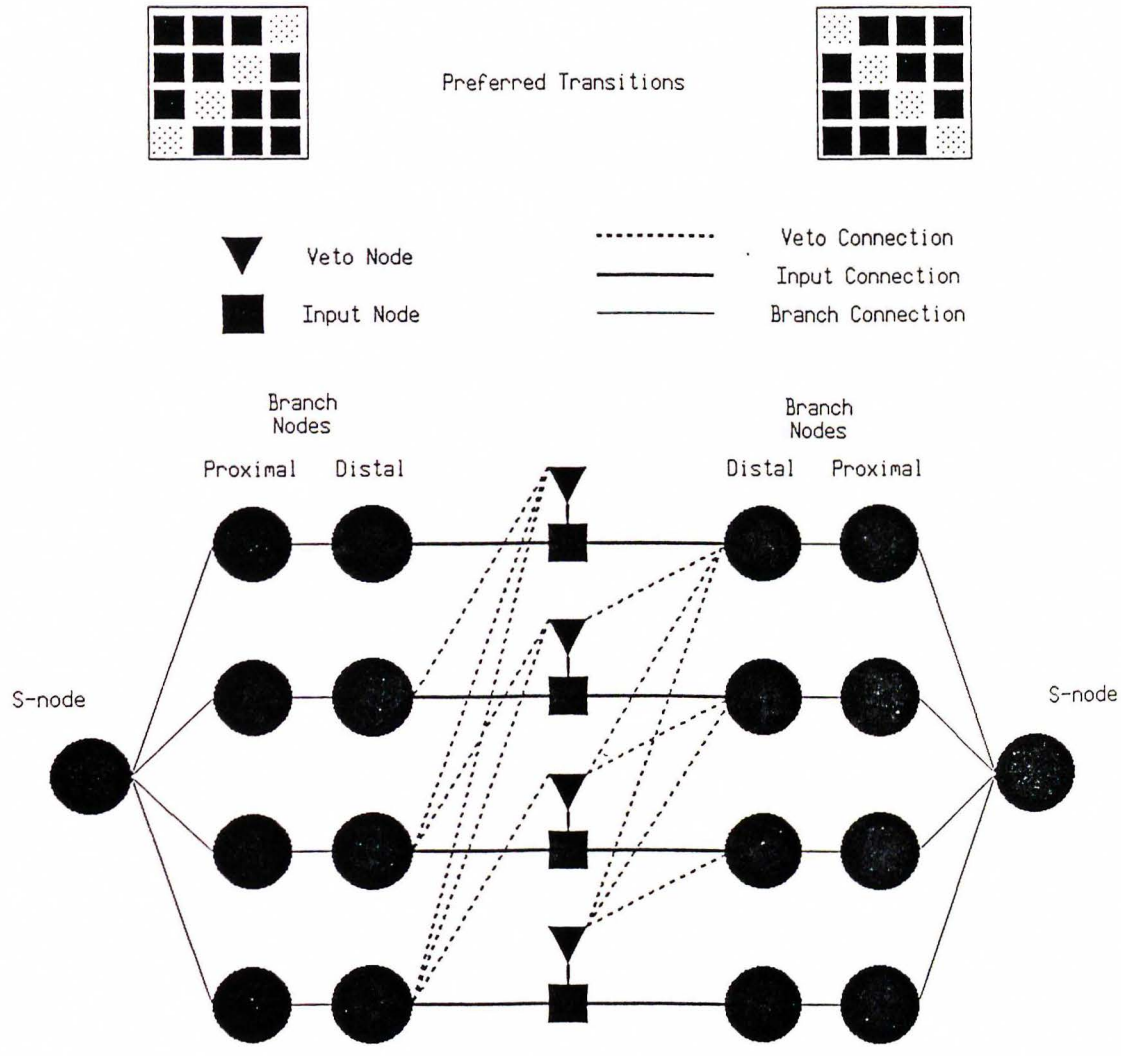


Figure 3.4. Two motion detectors for a rising or falling transition of the same rate. The detector on the right is tuned to the falling transition and the detector on the left is tuned to the rising transition.

is the weighted sum of the excitatory connections from input nodes and other branch nodes. The decay constant, δ, varyies among different types of detectors. Time, t, is again an update cycle rather than a time slice, with an update cycle

corresponding 1/7 of a 5 ms time slice or 5/7 ms. Excitatory input drives the activation value to its maximum (1.0). The rate at which it approaches the maximum is determined by the strength of the input signals. Input strength can be controlled by the weights of the connections. In the absence of input or if net_i is decreasing, the activation value exponentially decays towards 0, at a rate rate that is determined by the magnitude of the decay constant.

After excitatory activation is computed at a branch node, the effect of veto inhibition is calculated to produce the final activation value of the node. This activation value is determined by the equation

$$a_i^{t+1} = \begin{cases} e_i^{t+1} - \text{veto}_i^t, & \text{if } \text{veto}_i^t < \theta \\ 0 & \text{otherwise,} \end{cases}$$

where veto_i^t is the weighted sum of the veto connections at time t, and θ is the veto threshold. The effect of this equation is to eliminate excitatory activation in the presence of strong veto inhibition received by the veto nodes. If the veto node's activation is above threshold, the effect of inhibition is a sudden drop in the branch node's activation, regardless of decay times and excitatory input. This is analogous to the shunting inhibition seen in Chapter 2.

The S–node is connected to each of the proximal branch nodes. Its activation is determined by the sigmoid squashing function

$$a_i^{t+1} = \frac{1}{1 + e^{-(\text{net}_i^t - \theta)/T}},$$

where net_i^t is the weighted sum of the activation of the proximal branch nodes, θ is a threshold, and T is the temperature. When the S-node is activated it means that the unit has detected its preferred transition. The squashing function allows

a unit to respond in a reduced manner to transitions close to its preferred rate, but the units were designed not to respond to a transition preferred by another detector. Thus a unit tuned to respond to the transition in figure 3.3(c) will not respond to an event like those in figure 3.3(b) or (d), but may show a slight response to intermediate rates.

3.3.2 Motion Detector Operation

A demonstration of the operation of a detector with a transition of its preferred slope is shown in figures 3.5(a) – 3.5(c). In the figures, nodes with higher activations are shaded lighter. Figure 3.5(a) is during the seventh update cycle of the first time slice. The input node has activated the distal branch node $B1$ on the topmost branch, and its activation is beginning to spread to the more proximal branch node $B2$. The S-node is not receiving sufficient excitation to activate at this time. The strip charts show that the onset of the rise of activation of node $B2$ trails $B1$ by one update cycle, and the time that $B2$ reaches its maximum lags behind $B1$ by about two update cycles. The synchronous nature of the computation of each node's activation results in such transmission delays on the branches.

The behavior of the unit during the twenty-first update cycle is shown in figure 3.5(b). The third branch behaves just like the top branch in the previous figure, while veto inhibition has deactivated the top two distal branch nodes. Residual activation is present in the proximal branch nodes although those values are beginning to decay. These activations provide another kind of trace of previous activity. If the decay constants are set properly there will still be sufficient

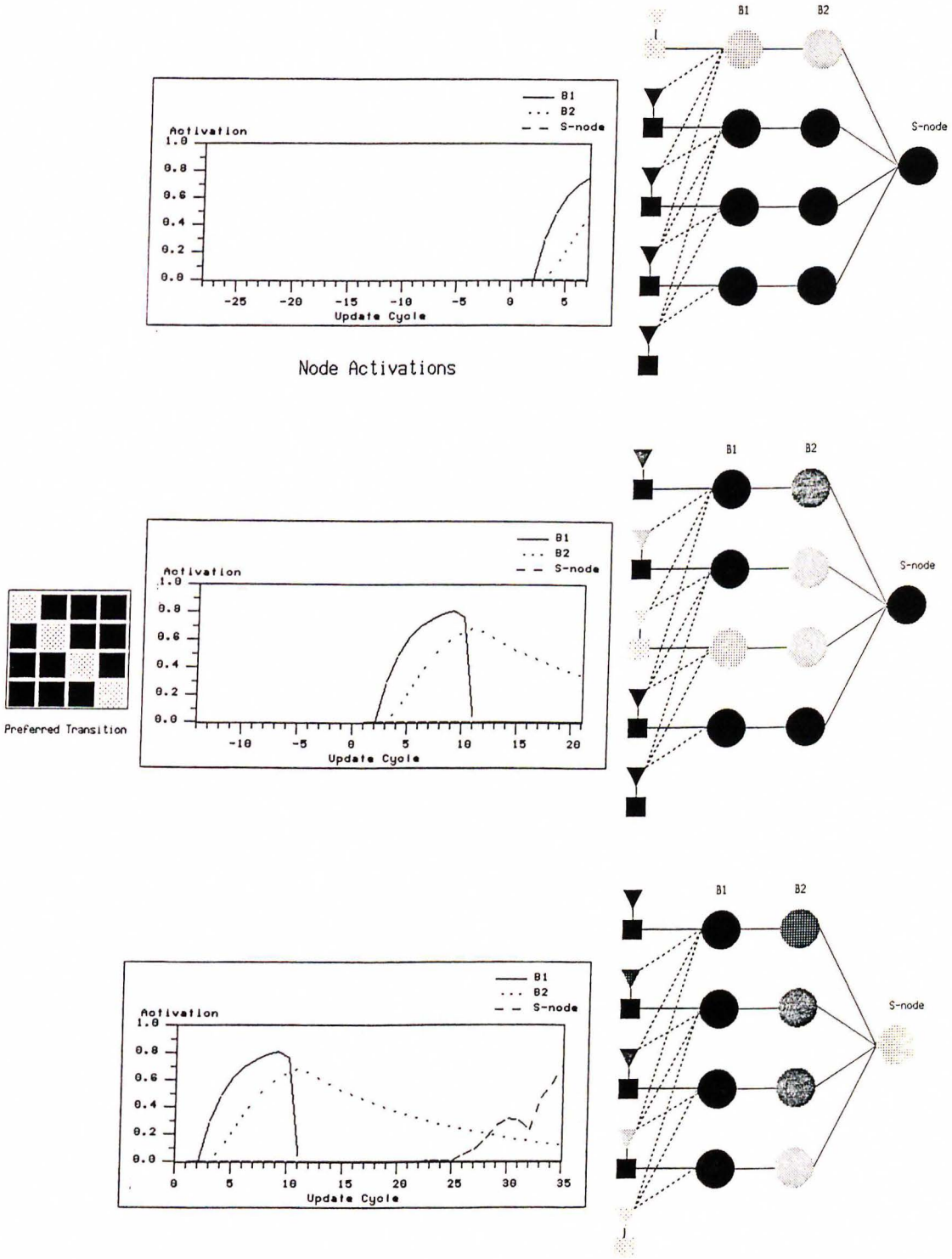


Figure 3.5. Sample run of a motion detector unit on its preferred transition. Part *a* is the top detector at update cycle 7, part *b* is the middle at cycle 21, and part *c* is the bottom at cycle 35. Nodes shaded lighter have a higher activation.

activation to trigger the S–node when the series is complete. In figure 3.5(c) the S–node has fired at the thirty–fifth update cycle, activated by residual excitation in the proximal branches. Transmission delays cause it to fire after the sequence has completed in the fifth time slice.

Although veto inhibition deactivates distal branch nodes for signals in the preferred direction, it occurs after sufficient activation has already propagated beyond the connection and hence has a reduced effect. If the transition is in the opposite direction, however, veto inhibition from the bottom veto node prevents the distal branch node from ever activating. This continues throughout the sequence, and since activation never spreads past the distal branches, the S–node never fires. This is how direction selectivity is implemented.

3.3.3 Designing the Motion Detectors for Specific Transitions

Veto inhibition is used to tune detectors to faster transitions as well as to specific directions. A partially constructed detector for a more rapid transition is shown in figure 3.6. Veto connections for direction selectivity are not shown to illustrate how inhibition is used in rate sensitivity. Veto nodes connect to the next branch in the sequence, ensuring a preference for patterns that skip a frequency or an input node, so as to prevent activation for patterns of contiguous frequencies. The receptive field of this unit is slightly larger than the previous unit, but not large enough to allow activation from even steeper slopes like the one in figure 3.3(e).

The construction of a detector for a faster slope such as that in figure 3.3(e) is accomplished by increasing the receptive field and adding veto connections

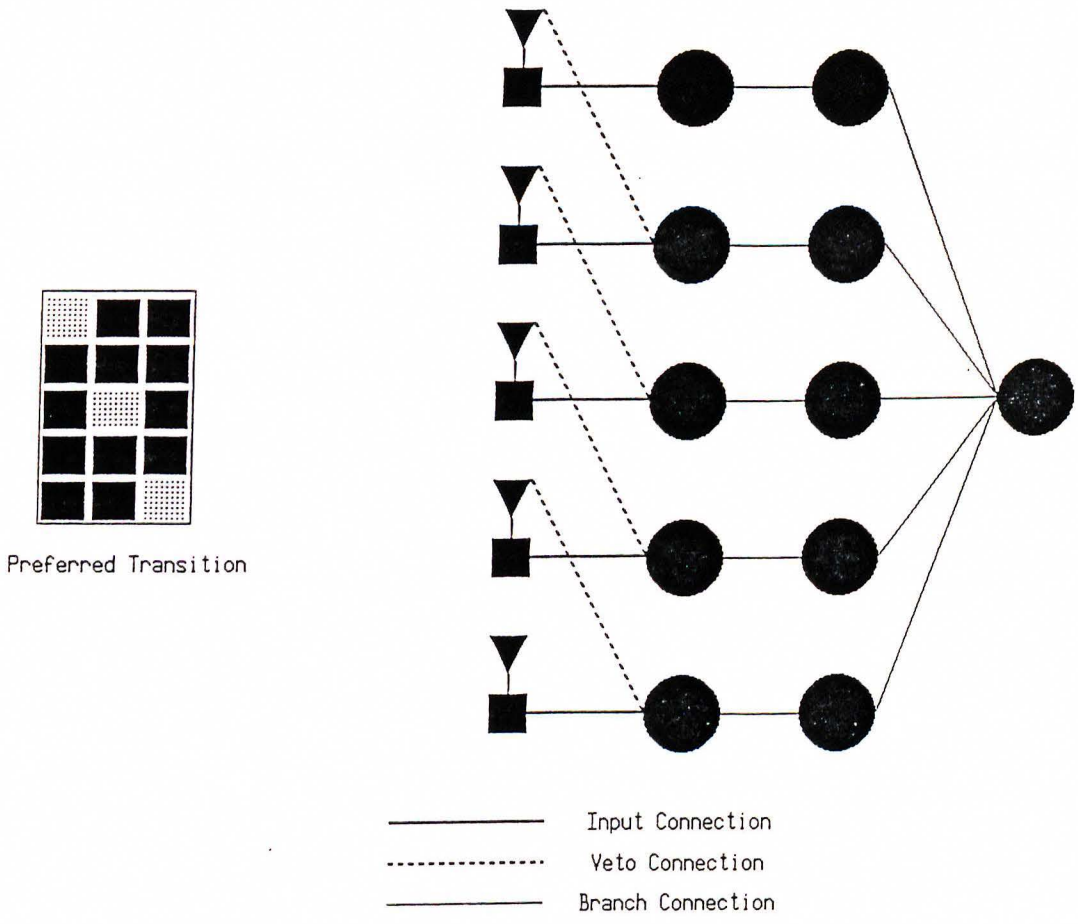


Figure 3.6. Motion detector for a faster transition showing only the veto connections for rate sensitivity. Veto connections for direction selectivity are not shown.

for a preferred pattern that skips two frequency units, or three units as in the case of figure 3.3(f). This can be extended for as fast a transition as needed. To complete the detector it is necessary to add additional veto connections for direction selectivity as seen in figure 3.7.

The identification of slower slopes is accomplished more by architectural and parameter manipulation than by veto inhibition. The architecture used to

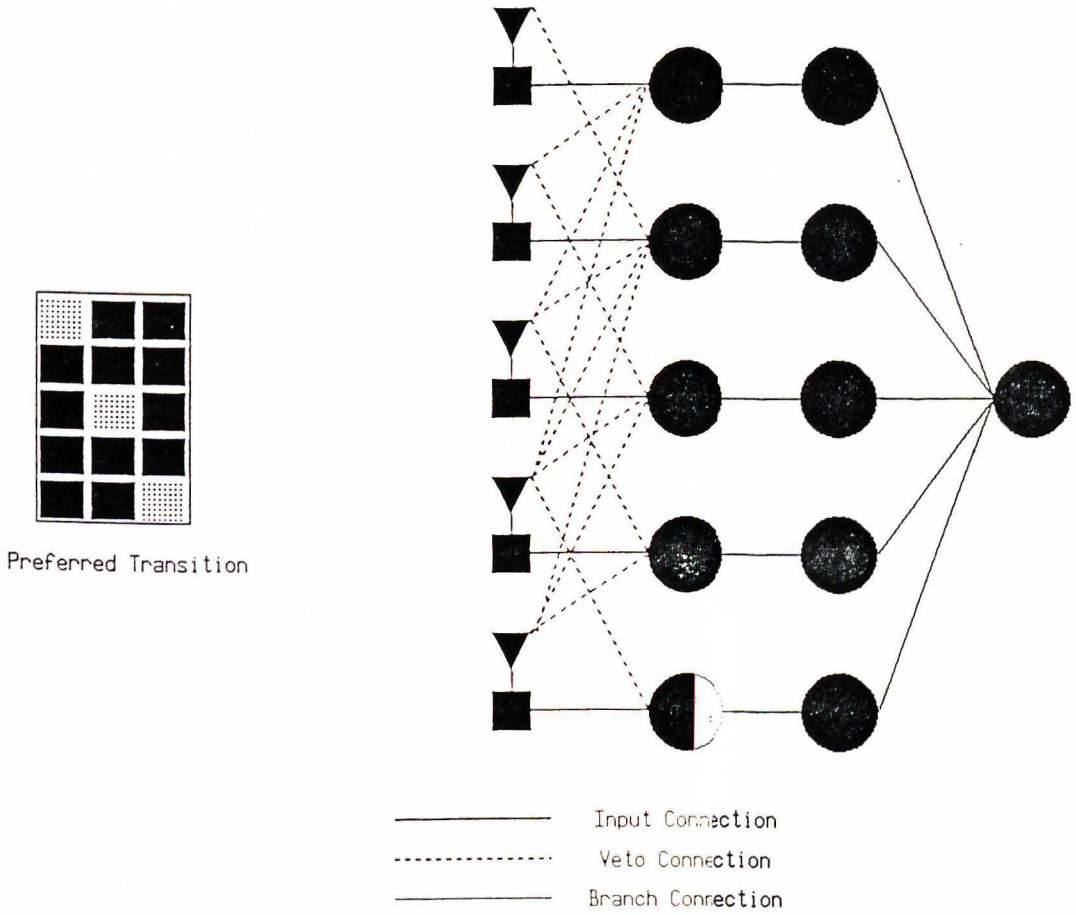
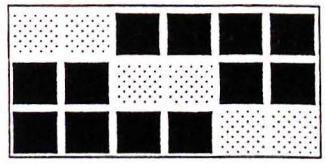


Figure 3.7. Detector for a faster transition with veto connections for both rate and direction sensitivity

detect the slower transitions in figure 3.3(a) and (b) is seen in figure 3.8. It is characterized by longer branches and additional excitatory connections. Veto inhibition is still used for direction selectivity. Discrimination of the two transi-



Preferred Transition

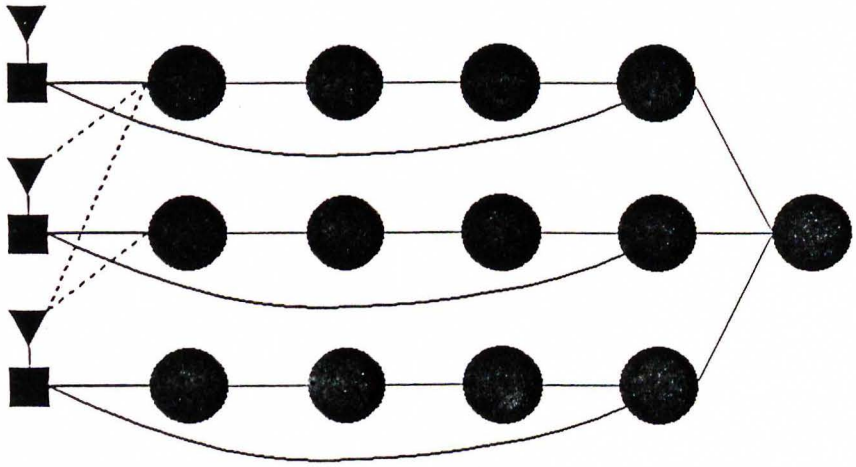


Figure 3.8. Motion detector for a slower transition

tion rates is done by parameters regulating the strength and rate of excitatory propagation along the branches, the strength of reinforcement of excitation at the proximal connections, and the rate of decay of excitation which determines how long residual activation is available to fire the S–node.

As discussed above, the activation function of the branch nodes is driven to a maximum value, the rate of which is determined by the magnitude of the input. A node cannot regulate the value on an input connection since that is set by the other node, but it can still affect the strength of an input connection by regulating its weight. Excitation on a connection with a low weight slowly raises

the node's activation and may never reach the maximum value if the connection is deactivated too soon. The same excitatory value on a connection with a high weight rapidly raises the node's activation to its maximum.

A branch node can have two types of excitatory connections: exterior connections from an input node and interior connections from other branch nodes. Excitation from an input unit on a distal branch node (whose weight is high) causes that node to reach a value near its maximum in seven update cycles. This value is transmitted to the next node in the branch, but since the value is lower than the value of the original input connection the second node activates at a slower rate, and may reach its maximum a few update cycles after the previous node, as demonstrated in the strip charts in figure 3.5(c). This delay lengthens as activation propagates along the branch. Varying the weight on interior connections affects the rate of propagation. Lower weights cause longer delays and *vice versa*.

To detect the transition shown in figure 3.8, the weights on the interior connections are set low enough that activation does not reach the most proximal branch node until well after the first time cycle. That node also contains an exterior connection to the same input unit as the distal end of the branch. The weight of that connection and the decay constant are set so that the node never reaches a high activation from exterior input alone. This can be thought of as a primer that decays if it is not reinforced. For this proximal node to be activated the exterior input must be reinforced by excitation arriving from an interior connection through the branches. In the case of the transition in figure 3.8, the presence of the same frequency for two consecutive time slices causes

reinforcement at the proximal branch node. For a faster slope such as the one in figure 3.3(c), by the time activation from the first time slice propagates to the proximal node the effect from its exterior connection has decayed, and insufficient reinforcement is present to contribute to firing the detector.

Identification of the transition in figure 3.3(a) requires further reduction of the propagation time and a reduction in the decay constant to slow activation decay. Otherwise the architecture is the same. The event required to fire a detector of this type has a duration three time cycles longer than that of the previous slope. If the decay constant on the proximal branch nodes is too high, information from earlier time slices decays too quickly before the final time slice and does not fire the detector. A lower decay constant provides a trace of information over a longer period and allows the detection of slower transitions. Increasing the decay constant tunes the detector to faster transitions.

Detectors have been designed to respond preferentially to each of the transitions in figure 3.3. There are symmetric detectors for transitions in opposite directions. A detector tuned to one of the six transitions will not respond to any of the other five, or to any transition in the opposite direction. There are other types of transitions between the detector's preferred transitions like those shown in figure 3.9. It is desirable for the motion detectors to identify these transitions as well. This is done by allowing detectors to respond to slopes close to their preferred rates. Suppose detectors P and Q are tuned to the transitions in figure 3.3(b) and (c), respectively. One detector will not respond to the other's preferred transition, but both will respond to the transitions of figure 3.9(a). The transition in figure 3.3(b) is signaled by the firing of detector P, the transition

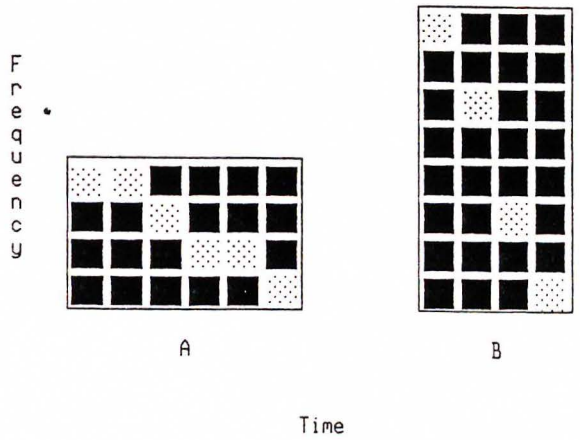


Figure 3.9. Transition rates between those assigned to motion detectors

in figure 3.3(c) is signaled by detector Q, and the transition in figure 3.9(a) is signaled by the combined output of P and Q.

Faster intermediate transitions, where a detector's rate sensitivity is determined by veto connections such as those in figure 3.9(b), pose a different sort of problem. A detector R, like the one in figure 3.7, will not respond to this transition because its receptive field allows only two branch nodes to be activated. A detector S that is tuned to the next higher transition will not respond since veto inhibition allows only two branch nodes to fire. To gain some response, the receptive field of detector R (and all of the faster transition detectors) is increased by one frequency unit. The problem with this is that this detector now responds to three different events, two transitions in figure 3.3(d), each displaced from the other by one frequency unit, and the transition in figure 3.9(b), without any way to distinguish between them. Because of this another type of detector for

precise transitions is constructed and is seen in figure 3.10. This detector, R_p, referred to as a precise detector, responds only to a transition in figure 3.3(d). This transition is signaled by the combined firing of R and R_p. If the transition is shifted by one frequency unit, it is signaled by the combined firing of detectors R and R'_p, if R'_p's receptive field is shifted by one frequency unit from R_p's. Only detector R fires in the case of the intermediate transition. A set of precise detectors is created for each of the transitions requiring veto inhibition for rate discrimination.

A final type of detector is used for the steady–state frequencies. This one fires when a particular frequency unit is on for four consecutive time slices, and continues to fire until that input unit deactivates. These detectors consist of a single node connected to only one input unit. They update once per time slice and are assigned a value of "1" if the steady–state condition is present at that frequency, and a "0" otherwise.

Detectors for each transition are reproduced throughout the entire frequency range for all of the 200 input nodes. Suppose detectors are tuned to the transition in figure 3.3(b) (their design is in figure 3.8). If $\{i_1, \ldots, i_{200}\}$ are the input nodes for the 200 frequency units, detector d_{30} would detect the transition from i_{30}–i_{32} (corresponding to the frequencies 600–640Hz), d_{31} for i_{31}–i_{33}, and so on. The branch nodes of the two detectors are isomorphic, with the same pattern of interior and exterior connections and the same parameters (this is not entirely true at the ends of the frequency scale, but since the data in this project never fall in those ranges the problem is ignored). The middle branch of detector d_{30} and the bottom branch of d_{31} have exactly the same exterior connections, with

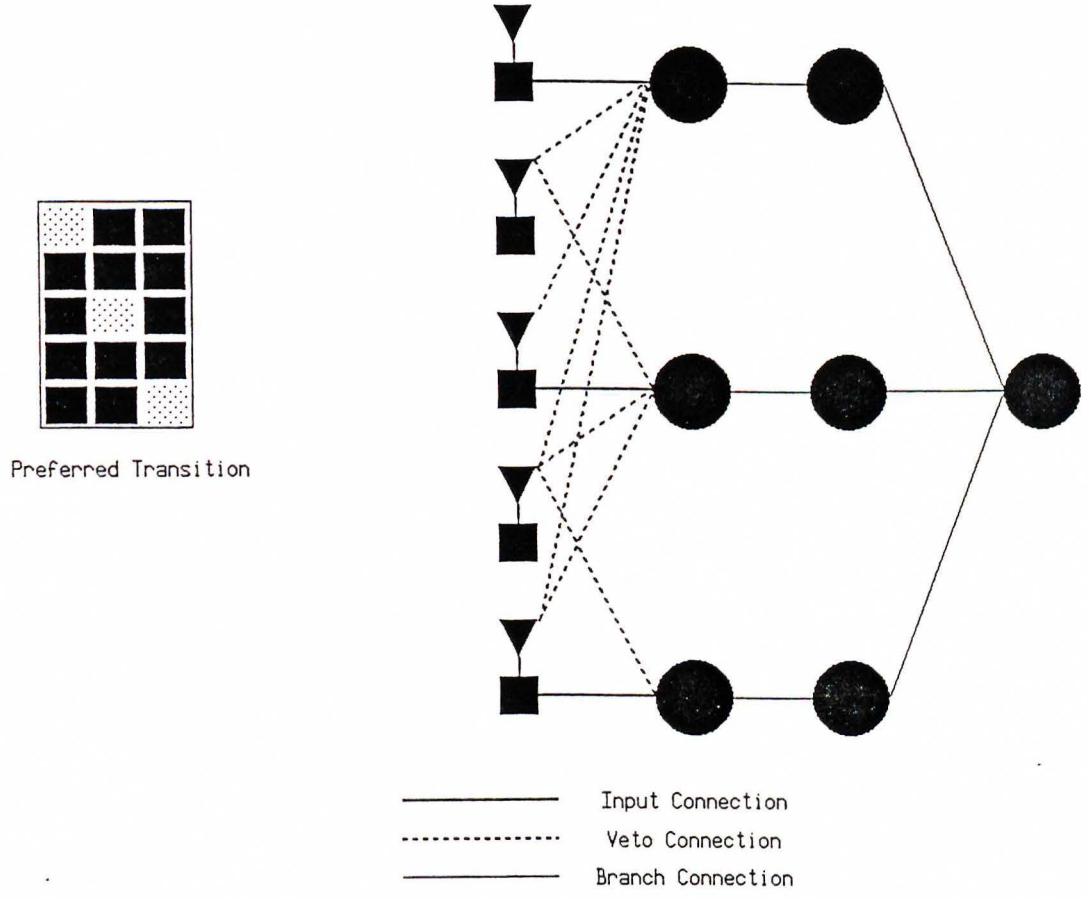


Figure 3.10. Motion detector tuned to a precise transition

excitatory connections to input node i_{31} and veto connections to veto nodes v_{29} and v_{30}. Because the nodes of these branches have the same exterior connections, isomorphic interior connections, and the same decay constants and connection weights, the activation values for corresponding nodes in each branch is exactly the same at any time slice. Thus it is possible to combine branch nodes of the detector, as in figure 3.11, for the sake of computational efficiency. The detector units differ only in the connections of the S–nodes. If $\{B_1, \ldots, B_{200}\}$ are branches

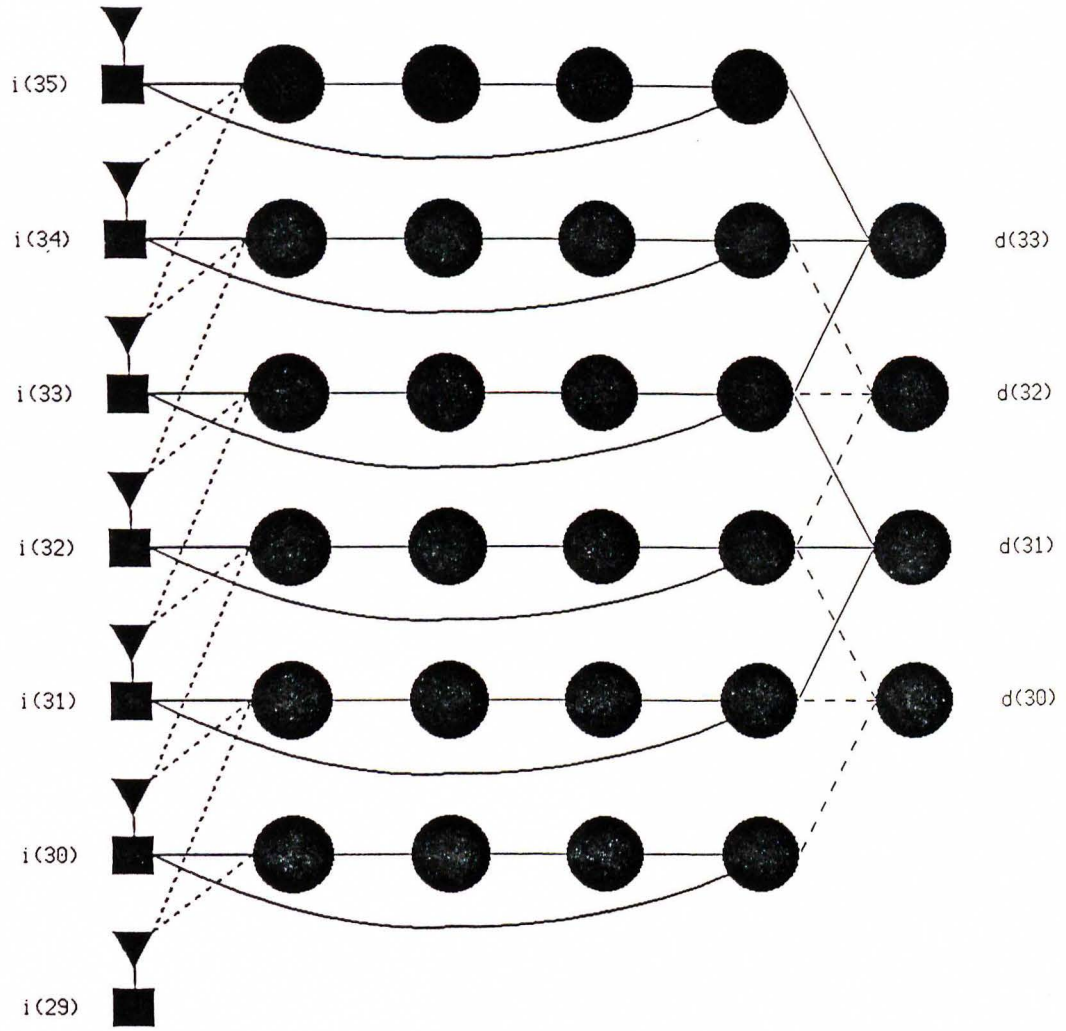


Figure 3.11. This shows four motion detector S–nodes and how they share branch nodes and connections.

connected to input nodes $\{i_1, \ldots, i_{200}\}$, the output of d_{30} is determined by the fact that its S–node is connected to the proximal branches B_{30}–B_{32}.

This architecture results in an array of detector S–nodes, each node respond-

ing to a transition beginning at a particular frequency or input node. There are nineteen such arrays. In figure 3.3 there are two arrays for each transition (a)–(c), one for rising and another for falling slopes, four arrays for each of the transitions (d)–(f), a regular and precise detector for each direction. There is one array of steady–state detectors. These arrays represent the output of the motion detector network.

3.4 Syllable Detection and Recognition

The output of the motion detectors is used in the detection and recognition of syllables. In this thesis the term syllable detection refers to the detection of a pattern of formant transitions associated with a syllable, with the possibility that a pattern is shared among different syllables. A syllable detector unit may be trained to respond to one particular syllable, but may also respond to others if they have patterns in common. A recognizer unit, on the other hand, is designed to respond only to one syllable and be silent for all the others.

The operation of the recognition network is divided into several phases, as shown in figure 3.12. Output from the motion detectors is captured and fed into a delay matrix which functions as a brief memory for transition events. Syllable detector nodes are connected to the delay matrix and are trained to respond to a specific syllable by a connection weight updating method. There are twenty–four nodes, one for each of the syllables used in the experiments. The detector nodes feed into another type of veto network for syllable recognition. Here the output of the detector nodes is normalized and, through the use of veto inhibition, the detection of shared or ambiguous patterns is resolved. The activity levels of the

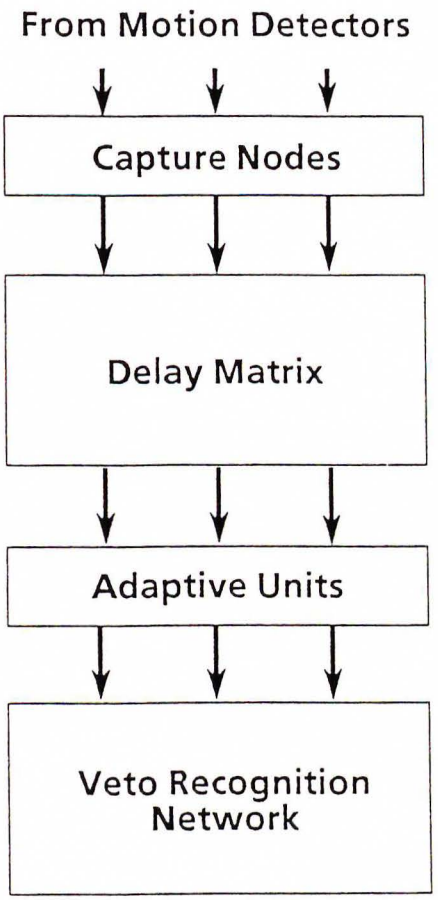


Figure 3.12. Outline of the subnetworks used in syllable recognition

twenty–four recognizer nodes comprises the output of the entire system.

Nodes in the motion detector network are updated seven times per 5 ms time slice. Output of the S–node of a motion detector is highly transient and may last for only a few update cycles. The nodes of the recognition network, on the other hand, are updated once every time slice. Because of this, every S–node is connected to its own capture node which captures and holds the S–node's maximum activation for each time cycle. This ensures that an S–node's output

will be seen even if it fires for an interval shorter than a time slice.

Syllable detection involves the recognition of temporal patterns of formant transitions. Many of the temporal pattern recognition systems discussed in Chapter 2 share a common feature of a delay line, used to preserve the input sequences or output behavior of the network over a few time slices. This allows the network to accumulate parts of the sequence before making a decision. If the length of the delay line corresponds to the entire duration of the input, the problem of temporal recognition is reduced to spatial recognition. Patterns of a long duration or those composed of a large number of input features make long delay lines computationally impractical. Reducing the length of the delay line causes the recognition unit to focus on the parts of the sequence containing the most useful information for discrimination. This has motivated the construction of a delay matrix to serve as input to the syllable detector units.

The output of each capture node is fed into a delay line, seen in figure 3.13, which propagates the signal across each node one time slice at a time. This allows the transition events to be preserved for five time slices as it passes along the delay nodes. These delay lines form a 200×5 matrix for specific transition events over the entire frequency range. Each row corresponds to the output of one S–node. The first column of the matrix is made up of the capture nodes, and the delay nodes comprise the rest of the columns. There are nineteen such matrices, one for each type of transition detector. The delay lines preserve the sequence of transition events without reference to the specific time of occurrence.

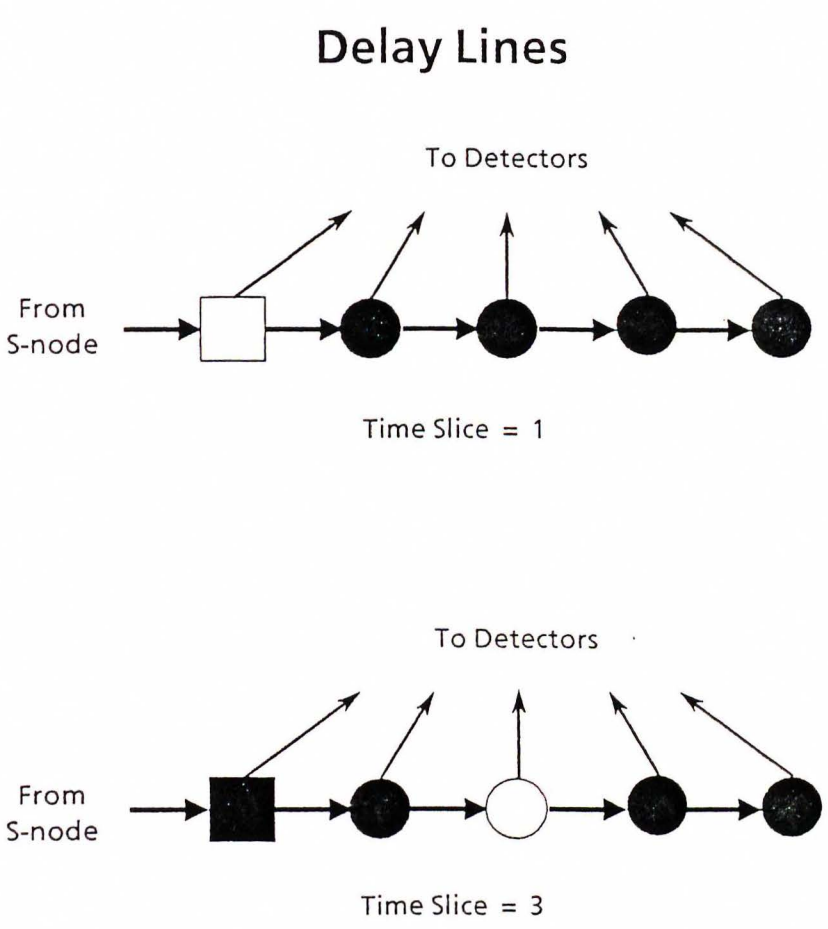


Figure 3.13. Diagram of a row of the delay line matrix at two different time slices. The lighter nodes are activated.

3.4.1 Syllable Detection Phase and Adaptation Method

Each of the twenty–four detector nodes is trained to fire when the patterns of a specific syllable are present. The nodes are connected to every node of the delay matrices, forming $200 \times 5 \times 19$ input connections for each detector node. The activation of the detector i is computed by the sigmoid squashing function

$$a_i^{t+1} = \frac{1}{1 + e^{-(\mathrm{net}_i^t - \theta)/T}},$$

where $\text{net}_i^t = \sum_j w_{ij} a_j^t$ is the weighted sum of the input connections to node i, θ is the threshold, and T is the temperature. The weights on the input connections determine whether or not a node is activated by a particular input pattern. Setting these weights by hand is an impractical task, and thus an adaptive mechanism for weight modification is needed.

Algorithms for determining the weights of a node, so–called connectionist learning methods, have been widely studied. The methods for temporal sequence detection discussed in Chapter 2 are some examples. For the most part these methods may be classified as self-organizing, meaning that their output nodes get no form of performance feedback from the environment. In supervised learning a "teacher" specifies the output behavior of the network and can provide feedback used in training. In the strongest form of supervised learning the teacher provides the correct output for each node, and training involves having the node learn to achieve its specified behavior. In other cases the network learns by receiving a performance measure from the environment. This is sometimes referred to as unsupervised learning since the *a priori* behavior is not specified by a teacher, but instead the environment provides some sort of global performance measure. The term is somewhat misleading since the environment itself can actually be thought to be supervising the learning process, taking the place of an omniscient teacher.

Widrow and Hoff [128] presented what has become the classic supervised learning method. Weights are updated by the equation

$$w_{ij}^{t+1} = w_{ij}^t + \alpha^t (z_i^t - y_i^t) x_j^t,$$

where w_{ij}^t is the weight for node i's connection to node j at time cycle t, α is a small real valued learning constant, y_i is the output value of node i, z_i is the expected output of the node provided by the teacher, and x_j is the value on connection j. This equation, sometimes referred to as the Widrow–Hoff rule [120], updates a node's weights based on how close that node's actual output is compared to its expected output provided by the teacher at each time step. Weight modification can be viewed as a gradient descent search through weight space in an attempt to reduce the error term δ, where

$$\delta_i^t = z_i^t - y_i^t.$$

If α is constant, the rate of weight change is greatest when y_i is far from z_i, giving a large δ value, and decreases as y_i approaches z_i. This rule is sometimes referred to as the delta rule. It can be shown that this rule will converge to a unique set of weights if the input is linearly independent [65, 97]. The learning constant α must be sufficiently small for this convergence to occur.

Classical conditioning is a type of animal learning behavior. This is the familiar Pavlovian conditioning where an unconditioned stimulus (US) such as the sight of food is paired with a conditioned stimulus (CS) such as the ringing of a bell to elicit salivation, a conditioned response. A common characteristic of this behavior is that the US may be delayed with respect to the CS, a duration called the inter–stimulus interval (ISI). Sutton and Barto [120, 119] have developed a real–time neural network model from the Rescorla and Wagner model [95] which incorporates ISI effects. Their weight update rule [119]

$$w_{ij}^{t+1} = w_{ij}^t + \alpha(y_i^t - y_i^{t-1})e_{ij}^t,$$

involves a temporal difference [118] error term, $y^t - y^{t-1}$, and an eligibility trace e_{ij}^t, which gives the eligibility of that weight for modification, also called the canonical eligibility [130].

The eligibility trace provides a limited memory so that input behavior can be spread out somewhat over time. The eligibility of input pathway j is given by the recurrence

$$e_{ij}^t = \beta e_{ij}^{t-1} + (1 - \beta) y_j^{t-1},$$

where $0 \leq \beta \leq 1$ is a decay constant. This trace shows exponential decay behavior. The ISI phenomenon is similar to the delayed input common to temporal pattern recognition. The notion of canonical eligibility has also been applied to a pole balancing task where the effect of certain control decisions may not be realized until some time has elapsed [11].

There is a striking similarity between the Rescorla–Wagner model of classical conditioning and the Widrow–Hoff method for solving linear equations. Sutton and Barto [120] have stated that the similarity between these two methods may be illustrative in the study of associative learning, even though they were designed for completely different tasks. These methods could be applied to other domains as well.

Multi–layer learning networks are able to learn more complex behaviors [83]. These networks are characterized by a number of hidden nodes between an input and an output layer. The use of the Widrow–Hoff rule is problematic since it is normally not possible to provide an expected value for any of the hidden nodes. This is similar to the credit assignment problem in classical Artificial

Intelligence literature [82]. One attempt to address the credit assignment problem has been with adaptive critic elements that allow a network to predict its optimal output behavior based on a single feedback signal from the environment [12, 118] which can be used to provide reinforcement or training signals to multi–layer networks [11, 9, 4]. During the learning process the network must determine the preferred output of the network and use this information to provide explicit training information to each node. Based on this information the network then tries to adjust its weights to achieve the predicted behavior.

The back–propagation method [97, 69, 90] that is seeing wide use in PDP models computes an error gradient in feed forward nets and sends this error measure back along its input connections. For each time step the value computed by each output node is compared to an expected value and the node adjusts its weights based on this value. The error term is back propagated along the node's input connections to the hidden nodes, which compute their own error term based on their output value and the error measures received from their output connections. The process of computing an error term, updating weights, and back propagating the error term is repeated for each layer of nodes in the network. Barto and Jordan [10] have claimed that computing an exact error gradient by hidden nodes may be unnecessary and have combined a supervised learning procedure for output nodes with the Associative Reward–Penalty algorithm A_{R-P} [8, 7] to estimate the error gradient in supervised learning tasks.

One problem associated with multilayer learning methods is the large amount of time needed to converge to a solution, caused by the computations of error gradients. Although suggestions have been made to improve efficiency

[130, 10] the computation time can still be prohibitive in networks with a large number of nodes and connections.

3.4.2 Implementation of the Learning Mechanism

The detector network is limited to a single layer of nodes partially because of the large number of input connections. The behavior of each of the twenty–four nodes can be exactly specified allowing the use of a single–layer supervised learning method, so a method similar to the Widrow–Hoff rule is applied. The weight update equation used is

$$w_{ij}^{t+1} = w_{ij}^{t} + \alpha(z_i^t - s_i^t)e_j^t,$$

where s_i is the weighted sum, $\sum_j w_{ij}x_j$, of the values of the input connections j. Eligibility is computed by

$$e_j^t = \beta e_j^{t-1} - (\beta - 1)x_j^{t-1}.$$

The use of an eligibility trace allows some temporal flexibility to the learning process. Functioning similar to the variable delay lines of Tank and Hopfield [123, 122], the delay lines coupled with the eligibility trace allow the duration and separation of events to vary somewhat between training and testing. The use of the connection sum rather than the actual activation value in the error term provides more control over the learning rate and final weight values. The output of the activation function of the recognizer nodes falls within the range $0 < a_i < 1$ and asymptotically approaches the extreme values. Consequently, if the expected value, z_i, is 1, excitatory weights will grow unbounded, since the actual output never reaches the expected value, and conversely for $z_i = 0$. By using the sum,

the expected value can actually reflect the values of the connection weights, keeping them from growing (or shrinking) unbounded and allowing variability in the onset of excitatory and inhibitory influences in the learning process. This greatly improves learning efficiency.

The actual operation of the network involves a training phase using certain tokens of input data followed by a testing phase on other tokens. During training, an input matrix for one syllable is presented to the motion detector network, which passes its output to the recognition layer. At each time cycle the detector units compute their activation and then update their weights. The expected value is set for each node depending on the identity of the syllable, with a high value for the node being trained for that syllable and a low value for all others. After the training phase the network is tested on different utterances of the same syllables. Network operation is the same in the training phase with the exclusion of the weight update operation.

3.4.3 Final Recognition with a Veto Network

The testing phase gives an idea of the accuracy of the training. A node can make two types of errors, one where it fails to fire for its trained syllable, called a miss, and another when it fires on the wrong syllable, called a false alarm. As will be described in the next chapter, these errors arise in experiments where the network is tested on utterances that it has never seen before. Similarities in the patterns of various syllables can cause a node to mistakenly identify one syllable for another. A final part of the system, the veto recognition network, is used to eliminate some of the latter types of errors. Consider two nodes, d_{bah}

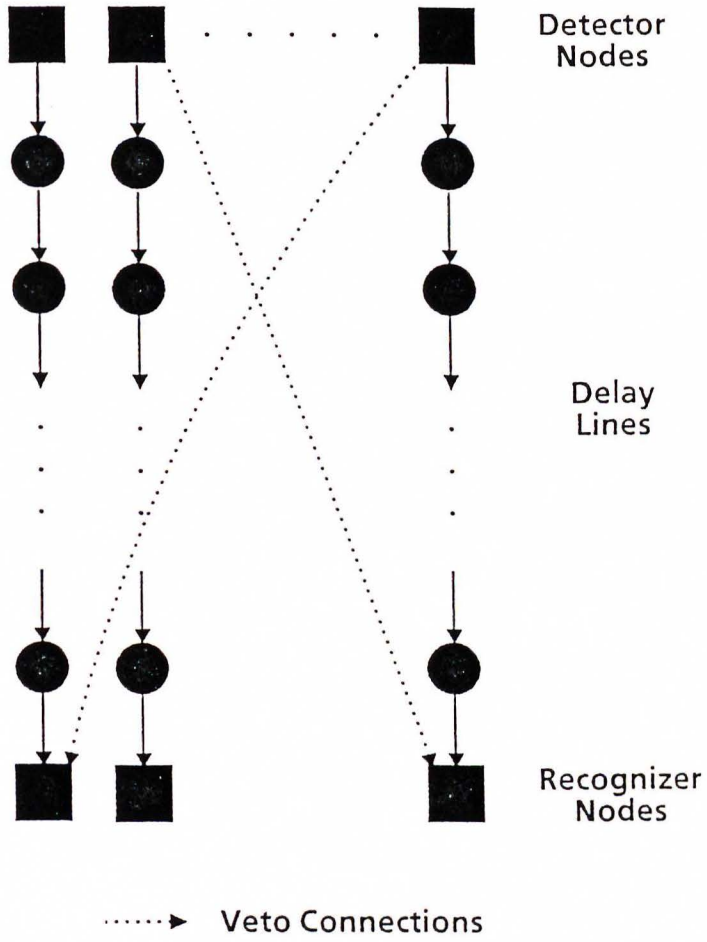


Figure 3.14. Part of the veto recognition network

trained to detect a "bah", and d_{bae} trained to detect a "bae". Suppose both nodes fire correctly when presented with their trained syllable, but d_{bah} also fires on a "bae" by mistake. Is there any way to eliminate this error?

The veto recognition network seen in figure 3.14 takes input from the detec-

tor nodes and uses it to compute the output of a set of twenty–four recognizer nodes. Each node is connected to a delay line which takes input from a single detector node, with each recognizer node corresponding to a particular detector node. Recognizer nodes also have veto connections from certain syllable detectors. The node r_{bah} can correctly discriminate a "bah" from a "bae" based on the output of d_{bah} and d_{bae}. When "bah" is presented only d_{bah} fires, activating r_{bah}. In the case of a "bae" d_{bah} fires mistakenly while d_{bae} fires correctly. A veto connection from d_{bae} to r_{bah} prevents it from firing on the wrong syllable, eliminating the error. Detector nodes fire at different times during the presentation of the testing stimulus. A veto connection, once activated, will inhibit firing of a node for the entire presentation of the stimulus. If d_{bae} fires first it will prevent the firing of r_{bah}. If it fires after d_{bah} the veto inhibition will arrive too late to prevent the firing of r_{bah} unless activation from d_{bah} is delayed. This is the function of the delay lines. They allow some time to elapse before the excitation from a detector node reaches a recognizer node. This allows the inhibition to block the firing of a node at the proper time.

A recognizer node has a single excitatory input connection at the end of its delay line. The weight is set according to the strength of the signal received from a detector node firing on the correct syllable. Excitatory activation is computed by the familiar sigmoid squashing function, and is then subject to veto inhibition. Each recognizer node may have some of its veto connections enabled with a particular threshold. If the value on an enabled veto connection is greater than its threshold, activation is completely shunted and the node's output is set to 0, otherwise the excitatory activation is the output value.

3.5 Summary

The mechanism of visual motion detection [63] is used to construct detectors sensitive to specific types of formant transitions. This is a dynamic form of neural network computation and provides the first layer of processing for syllable recognition. Formant transition information is provided to the second layer for syllable detection. This process involves a learning method which combines ideas from temporal sequence detection, supervised learning, and classical conditioning. Since the detection level has many false alarms, a final level of processing is used to actually recognize the syllable. Veto inhibition is used to fine tune the output of the detectors, and delay lines allow for the time–varying nature of the network's behavior and the task itself. The entire system must detect events that may take some time to develop or are separated by some delay.

Veto inhibition has played two roles, both at the lowest level of motion detection and at the recognition level. Coupled with a set of adaptive units the system detects the movement of formants and then decides how to use this movement data in the recognition of syllables. How well it performs is described in the next chapter.

4. Results

The performance of SYREN and the usefulness of formant transition information was evaluated through syllable recognition experiments. The data corpus contains samples of twenty–four syllables. Each syllable's corpus consists of formant centers from five "raw" repetitions from a single speaker and one set of averaged centers from the five raw syllables. Experiments differed in which tokens were used in training and testing.

In the first experiment the network was trained and tested on the average data. This was a preliminary experiment to study the performance of the transition detectors and the learning algorithm, and it provided a set of initial parameters for use in the other studies. In the next task the network was trained on the five raw repetitions and then tested on the average data. This tests the network's ability to detect common features of a syllable and use those features

to recognize another token from samples that it has never seen before. The final experiments gathered more performance data by testing the network on each raw repetition in the data corpus while using the others for training. This makes up for the fact that the averaged data could be considered an idealized sample of the syllable. Testing on the raw repetitions as well may give a more reliable performance measure. In many samples, however, the average formant centers did not closely resemble any of its raw data for some syllable types.

In each experiment, the performance of the network is analyzed by looking at both the percent correct recognition performance and the behavior of the detector nodes during the recognition process. This yields an overall recognition score and provides clues to what information the network is using in recognition. In any task where input is spread out over time, an adaptive system must decide what information is important for recognition. In analyzing the behavior of the syllable detection network it is possible to determine what information it is using in its operation.

The first section of this chapter gives the result of the average only experiment. The next section discusses the results of the raw–average experiment and shows in detail how the veto recognition network is used to tidy up the learning process. Following that, the experiments that tested on the raw data are presented. Finally, the effect of various parameters on the learning process is discussed.

4.1 Average–Only Experiment

For the average–only experiment, the network was presented with the averaged

formant centers for both training and testing. Twenty–four detector nodes were trained to respond to one of the twenty–four syllables. Due to computational requirements each detector was trained separately. Detectors were trained by presenting each syllable of the averaged data as input. If the input is the syllable that the node is supposed to detect, its preferred syllable, the expected value of the node was set to a reward value, otherwise it was set to a penalty value. The number of time slices presented as input is called the exposure window of a trial. It reflects the duration of the presentation. In this experiment the exposure window was twenty–five time cycles, meaning that the first 125 ms of each input syllable was used. This is sufficient to allow all formants of each syllable to reach a steady–state condition. Steady–state is defined as a formant center at one frequency for four time cycles (20 ms). A training cycle is the presentation of each syllable of the training corpus one time. Each node was trained for four training cycles.

After training was completed the network was tested on the average data (the same as the training corpus). Each syllable was presented and the nodes calculated their activation at each time slice, using the weights determined during the training phase. Each node's activation value, a, falls within the range of $0 < a \leq 1$. In this experiment detection was successful if the node achieved an $a \geq .9$ at some point during the input presentation and no other nodes responded to that syllable with an $a \geq .1$.

Correct detection was accomplished 100% of the time with each detector node responding to its correct syllable and to no others in the corpus. All nodes signaled their correct syllable with an activation value greater than 0.97. Because

recognition was perfect there was no need to construct a veto recognition network.

Table 4.1 shows how each detector node responded to its preferred syllable, displaying the time of the onset of activation and, in some cases, where the node begins to deactivate. The temporal characteristics of the activation behavior can shed some light on the information that the node is using for recognition. By comparing activation values with the formant input it is possible to tell what is actually firing the node and what may be preventing its firing.

A good example is provided by the detector for the syllable "beh". The formant tracks are shown in figure 4.2, and the performance of the detector of the syllable with that input is shown in figure 4.1. The onset of activation begins at the seventh time cycle (35 ms). At this point on the spectrum no formant has reached a steady–state condition. The first formant has reached its final frequency but the steady–state detector will not fire for three more time slices. The detector is firing on transition data alone. The formants do not appear until after the second time slice, corresponding to the onset of voicing of the stop consonant. Transition events are not signaled until the fifth time slice for F1, and the sixth time slice for F2 and F3. These initial transitions are not sufficient to fire the node, but the delay matrix preserves a record of these events for five time slices. As more transition data arrive in the delay matrix, the node fires. This node is characterized by a deactivation beginning at the sixteenth time slice. In figure 4.2, steady–state frequencies are reached for formants F1, F2, and F3 at the seventh, tenth, and ninth time cycle, respectively. This leads to the firing of steady–state detectors at the eleventh, fourteenth, and thirteenth time cycle. When a steady–state detector fires, no more transitions

Average–Only Results		
Syllable Detector	A–onset (time–cycle)	A–decline (time–cycle)
bae	7	19
dae	11	22
gae	9	24
bah	7	19
dah	7	22
gah	10	–
bee	7	21
dee	9	–
gee	11	–
beh	7	16
deh	11	–
geh	10	24
bei	8	21
dei	9	–
gei	11	–
bih	7	15
dih	9	–
gih	10	24
bou	7	16
dou	11	24
gou	11	–
buu	7	17
duu	9	–
guu	11	–

Table 4.1. Performance of the twenty–four syllable detectors when presented with their preferred syllable, showing the time cycle of activation onset and decline. Dashes mean the activation does not decline.

are signaled for that formant and previous transitions are at the end of their delay lines. Both transition and final vowel target information is available. As time passes, transition information is lost and is no longer available to fire the node. In figure 4.1, the activation of the node begins to decline at the sixteenth time cycle. At this time the delay matrix is filling with information from the

steady–state detectors and is losing the last of the transition data. Activation is reduced completely when only the vowel target information is present. There are two possible causes for this: either the weights on those particular connections are excitatory, but not enough to fire the node, or the connections have been inhibited during the learning process, telling the node to ignore this information. Most of the nodes that exhibit a decline in activation show some combination of these conditions.

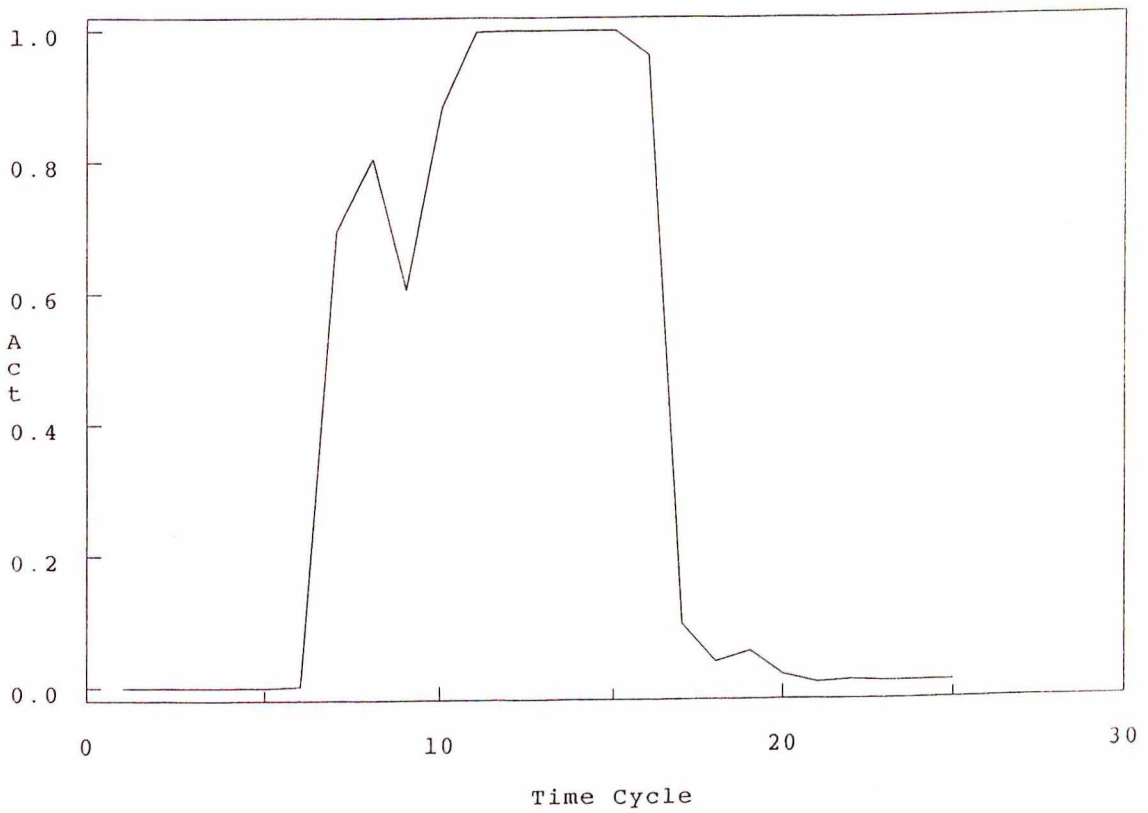


Figure 4.1. Activation curve for detector node trained for syllable "beh" when presented with that syllable in the Average–Only experiment.

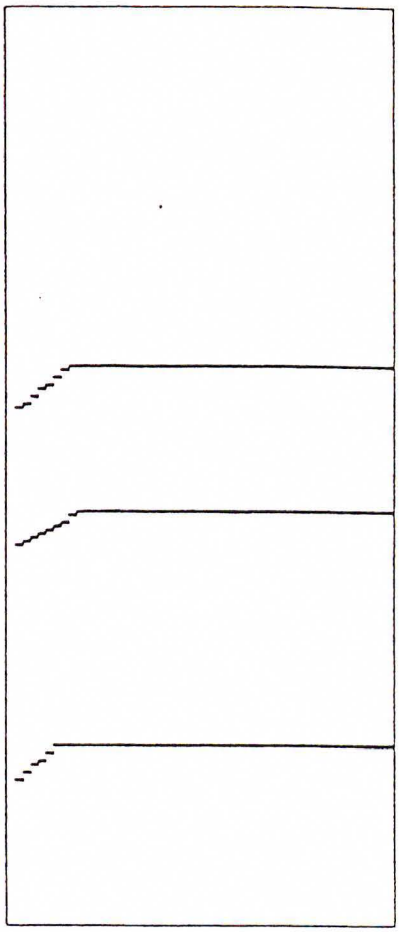


Figure 4.2. Input matrix for syllable "beh". Horizontal axis is 5 ms time slices and the vertical axis is frequency.

Performance of the detector for the syllable "deh", when presented with that syllable, is shown in figure 4.3. This detector was characterized by no reduction in activation, meaning that it was using steady–state information in the later part of the presentation and transition information earlier. The onset of activation was three time cycles later than in "beh", caused by a longer onset of voicing (beginning at the fourth cycle) and a slower F3 transition. Correlation of

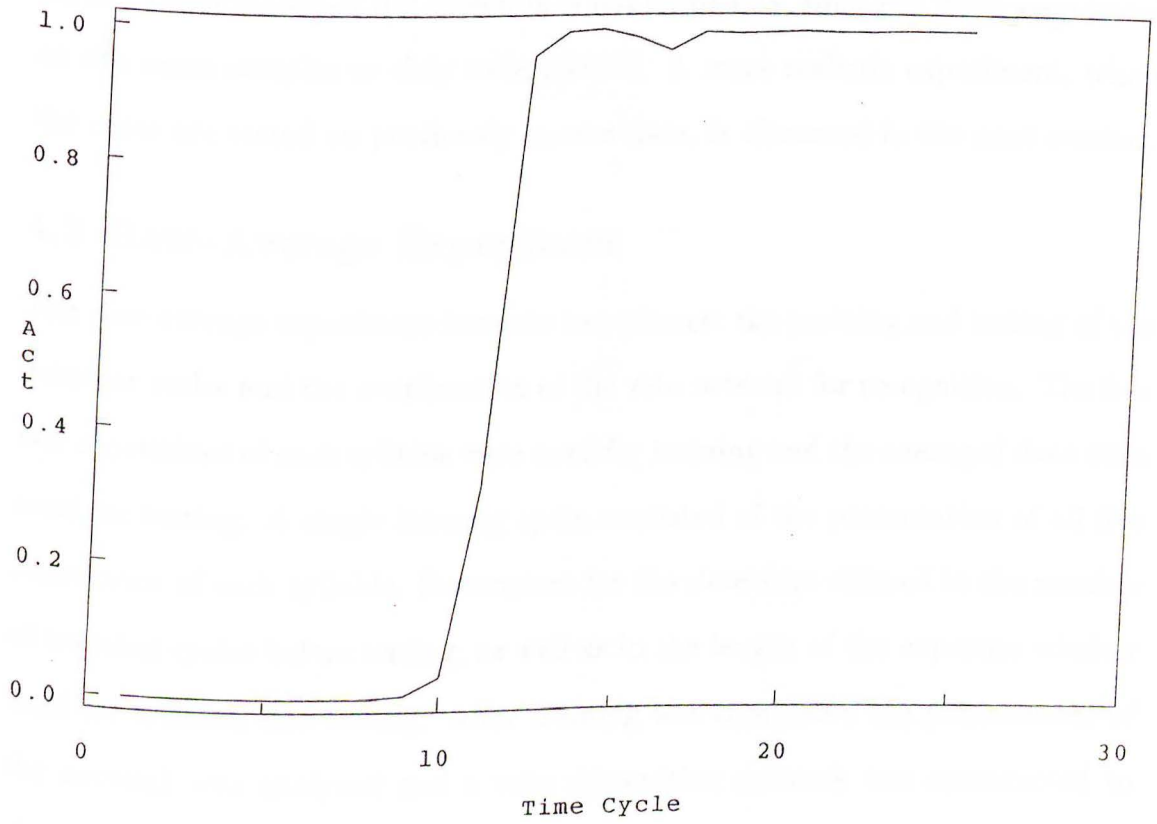


Figure 4.3. Activation curve for the detector trained for syllable "deh" when presented with that syllable in the Average–Only experiment.

activation onset with voicing onset and transition type is a common property in the average–only experiment. Onset of activation depends on the onset of voicing and the nature of the initial transitions. This means that syllables beginning with a /b/, which has a small delay in the onset of voicing, should cause activation sooner than a /d/ or a /g/.

The detector nodes use formant transition information in the recognition of the syllables. Twelve of the twenty–four detectors depend primarily on the transitions, while the others can use both transition and steady–state data. The

perfect performance of the detectors is not surprising, since the units were tested on the same samples as they were trained. A more realistic experiment, where the units are tested on previously unseen data, is discussed in the next section.

4.2 Raw–Average Experiment

The raw–average experiment involves two phases: the training and testing of the detector nodes and the construction of the veto network for recognition. The five raw repetitions of each syllable were used for training and the averaged data were used for testing. A single learning cycle consisted of the presentation of all five repetitions of each syllable. Parameters for the detectors differed in the number of training cycles before testing, as well as in the length of the exposure window used for training and testing. After training was completed the performance of the network was analyzed and a veto recognition network was constructed to clean up the deficiencies from training.

The performance of the twenty–four detector nodes on the averaged data after training is seen in table 4.2. This table gives the output of each detector node when the network is presented with that detector's preferred syllable. As in the average–only experiment, the onset of activation of the node can occur at any time during firing, and the table gives the activation onset and the maximum value reached by the node. The training process is not perfect and the table also shows the other detectors that fire mistakenly on a given syllable. For example, when a "bae" is shown to the network the detector for a "dah" also fired.

The shapes of the activation curves are markedly different from the average–only experiment. Many show a spike shape as in figure 4.4, or multiple spikes

Syllable Presented	Preferred Detection Onset	Maximum	Shape	Others Firing
bae	7	0.49	spike	dah
dae	12	1.00	plateau	
gae	17	0.12	spike	*gah*
bah	10	0.26	spike	bae *bou*
dah	7	1.00	cliff	deh
gah	12	0.81	spike	*dah*
bee	14	0.09	cliff	*buu*
dee	16	0.34	spike	gee
gee	14	1.00	spike	*dee*
beh	10	0.70	spike	
deh	17	1.00	cliff	dei dou
geh	12	0.17	spike	*dou*
bei	16	1.00	plateau	dei
dei	12	0.90	cliff	
gei	17	1.00	cliff	*geh*
bih	9	0.73	plateau	bee
dih	15	0.48	spike	dei
gih	12	1.00	plateau	geh
bou	8	0.40	spike	beh
dou	12	1.00	cliff	
gou	16	0.68	spike	
buu	9	1.00	cliff	
duu	–	–	–	
guu	–	–	–	*duu*

Table 4.2. Network performance when presented with each syllable. Preferred detector response is shown along with other detectors that misfire on each syllable. Italicized syllables are false alarms with a value greater than the detector's maximum activation on its preferred syllable. Dashes indicate no response to the preferred syllable.

as in figure 4.5. Other shapes can be classified as plateaus, shown in figure 4.6, or cliffs which extend to the end of the exposure in figure 4.7. The shapes of the activation curves can shed some light on the detection process just as in the average-only experiment. An examination of these curves reveals what

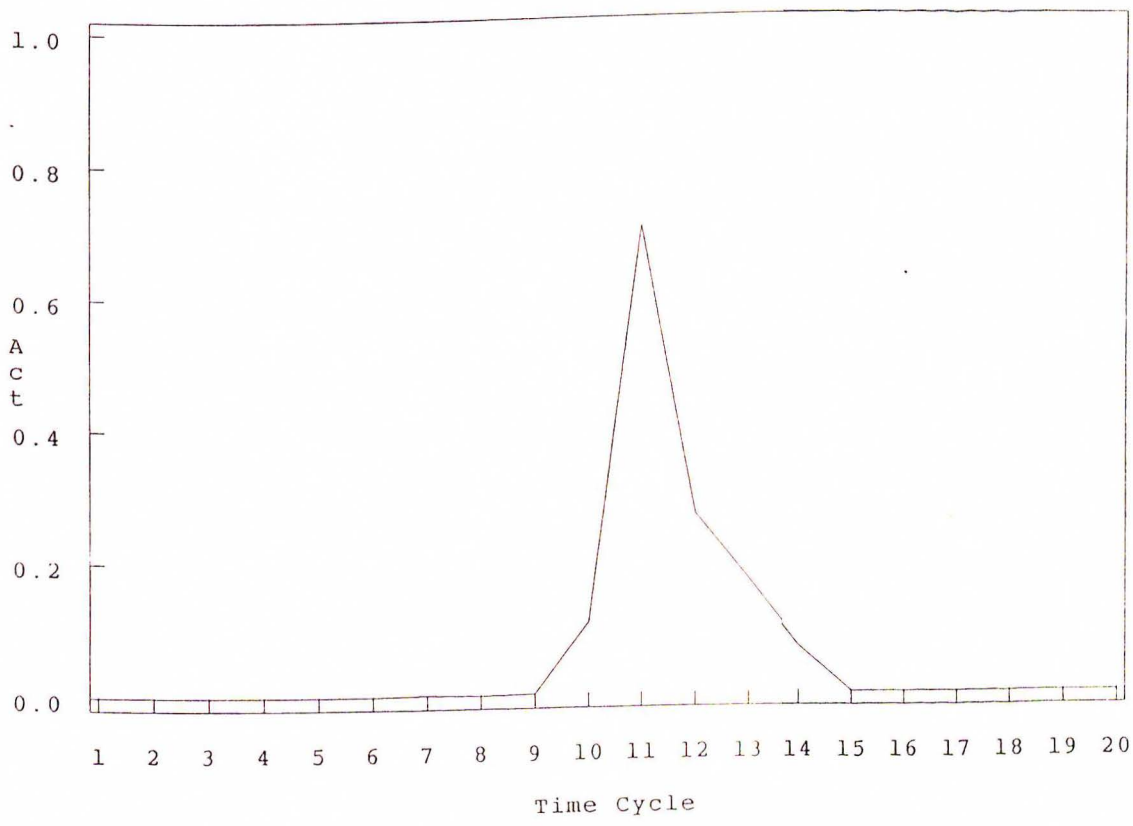


Figure 4.4. Activation curve for detector node trained on syllable "beh" with that syllable as input in the Raw–Average experiment.

information is being used by the node during the detection process.

Table 4.3 shows what information is available when a node fired on its preferred syllable. Eight nodes fired during the period when only formant transitions are present in the data. Of those eight, three nodes fired only on transition information. These three nodes fired early, reached a comparatively low activation value, and deactivated before steady–state information is present. Of more interest is that sixteen of the twenty–two nodes that fired on their correct syllable did so while both transition and steady–state events were present in the delay lines.

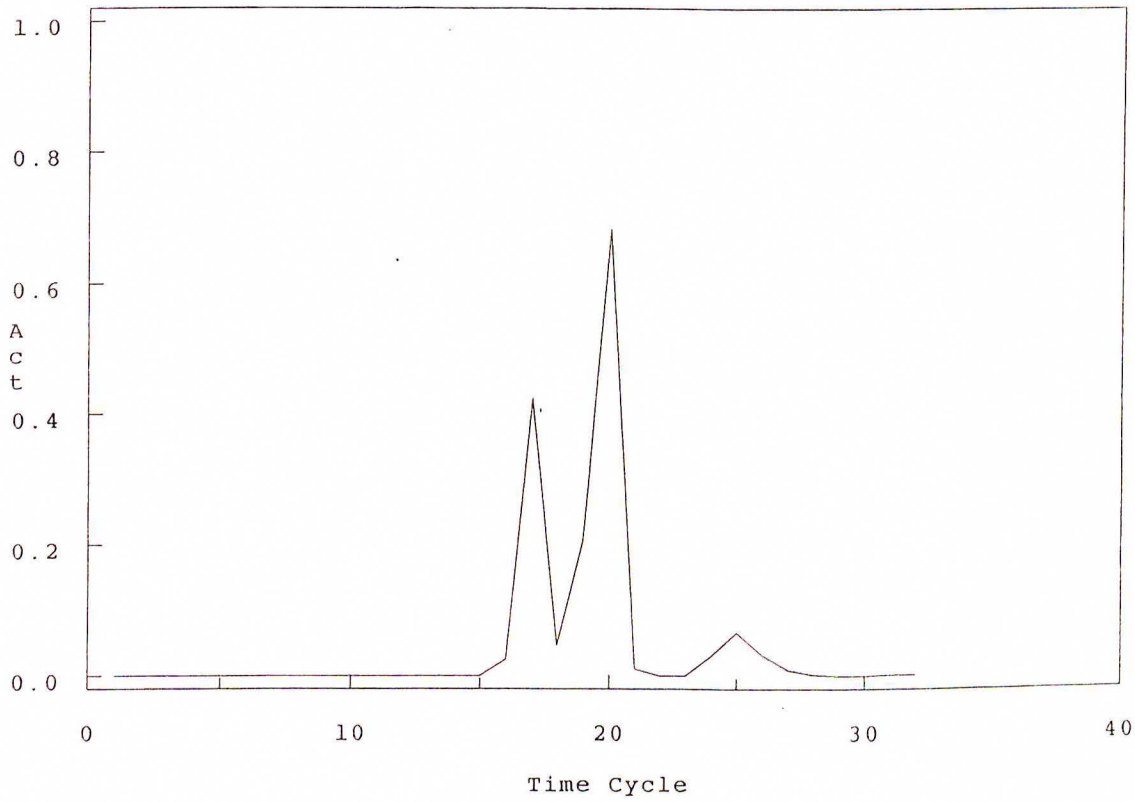


Figure 4.5. Activation curve for detector node trained on syllable "gou" with that syllable as input in the Raw–Average experiment.

Of those sixteen, eight fired exclusively with this combined information. Due to the four cycle delay in signaling a steady–state frequency, these two events were present together in the delay matrix for only about two time cycles. This can result in a sharp spike shape of a node's activation curve. This is most clearly illustrated in figure 4.5. The three spikes in the curve correspond to the activation of steady–state detectors for F3, F1, and F2 at time cycle sixteen, twenty, and twenty–five, respectively. Plateau–shaped curves normally represent the use of combined information spanning two or three formants.

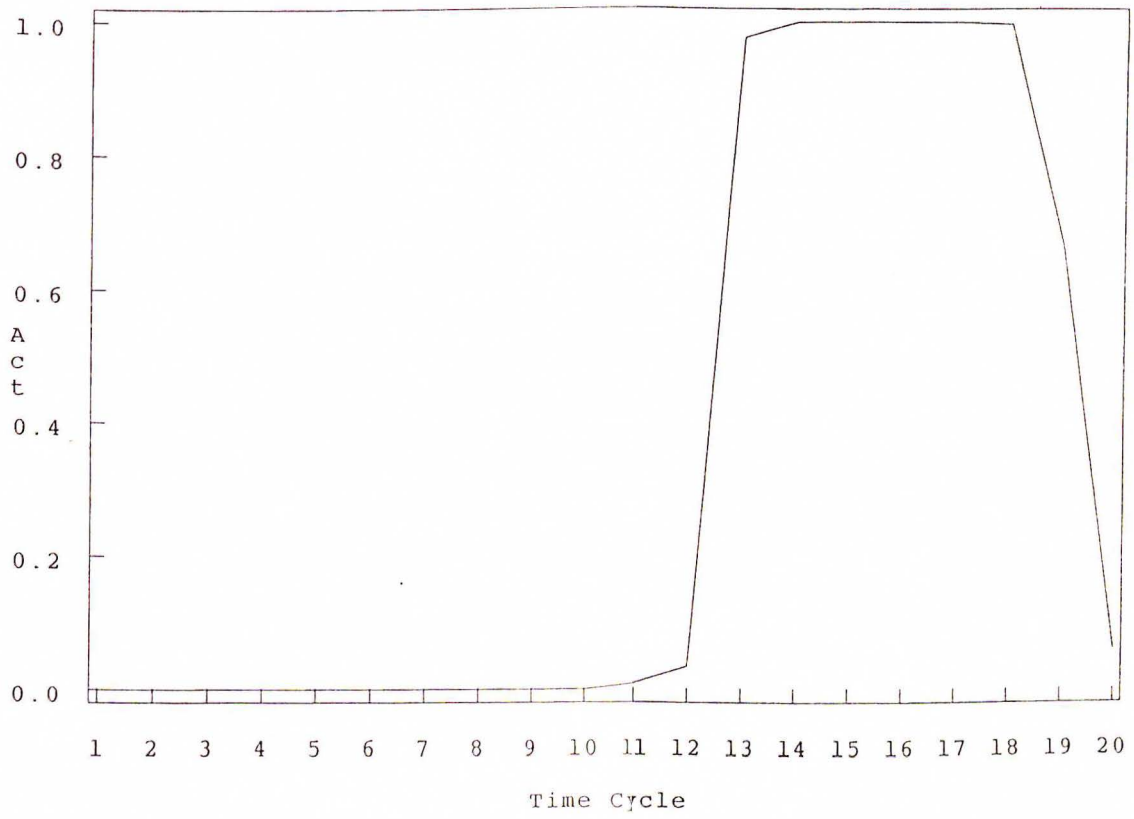


Figure 4.6. Activation curve for detector node trained on syllable "dae" with that syllable as input in the Raw–Average experiment.

Eight nodes showed strong activation in the presence of steady–state information alone and are characterized by cliff–like activation curves. Of these, three fired only in the absence of transition information. Three other nodes showed a slight activation in the presence of the vowel alone.

The majority of the nodes fired with combined consonant and vowel events, and one third of the nodes fired exclusively with this information. For syllable detection, at least, a combination of the transitions from the consonant to the vowel and the vowel target itself is needed. This shows that CV syllable detection

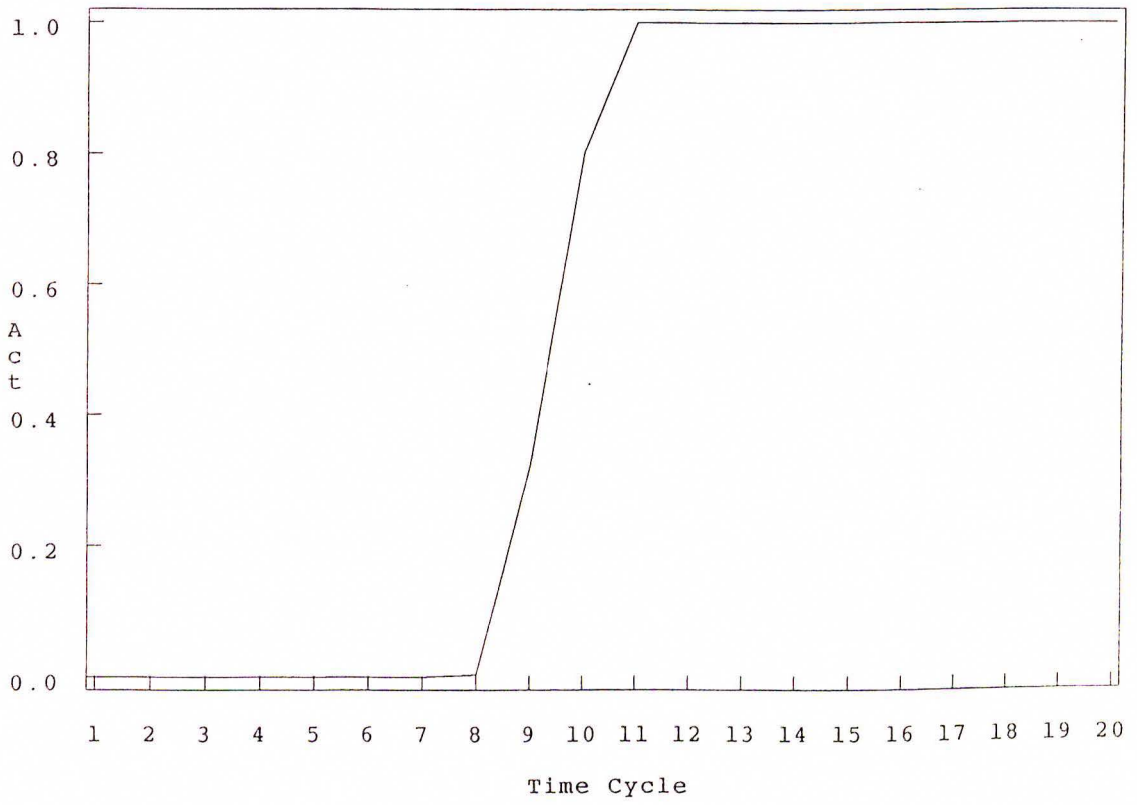


Figure 4.7. Activation curve for detector node trained on syllable "buu" with that syllable as input in Raw–Average experiment.

relied on properties from both the consonant and the vowel. Nineteen of the twenty–four detectors, roughly 86%, used formant transition information at some point when they fired on their preferred syllable.

A detector can make two types of errors: a miss error, where the detector fails to fire on its preferred syllable, and a false alarm, where a detector fires on an incorrect syllable. Table 4.4 shows the performance of each detector node on its preferred syllable and the incorrect syllable, if any, that caused it to misfire. The failure of the detectors for "duu" and "guu" to fire on their preferred syllable

Syllable Detector	Transition Only	Transition & Steady–State	Steady–State Only
bae	•		
dae		•	
gae	•		
bah		•	
dah	•	•	•
gah	•	•	
bee			•
dee		•	
gee		•	
beh		•	
deh			•
geh	•		
bei		•	
dei			•
gei		•	•
bih		•	
dih		•	•
gih	•	•	
bou	•	•	
dou	•	•	•
gou		•	
buu		•	•
duu			
guu			

Table 4.3. Table shows what information is available when a detector fires on its preferred syllable.

is caused by the lack of common transitions between the training corpus and the averaged data. These samples share common steady–state frequencies, but apparently that information was inhibited during training due to other syllables with the same target frequencies. The lack of common transitions in the training and testing corpus seems to be a feature of detectors that fire only on the vowel

Syllable Detector	Preferred Detection Onset	Maximum	Others Firing
bae	7	0.49	bah
dae	12	1.00	
gae	17	0.12	
bah	10	0.26	
dah	7	1.00	bae gah
gah	12	0.81	*gae*
bee	14	0.09	bih
dee	16	0.34	*gee*
gee	14	1.00	dee
beh	10	0.70	*bou*
deh	17	1.00	dah
geh	12	0.17	*gei gih*
bei	16	1.00	
dei	12	0.90	bei deh dih
gei	17	1.00	
bih	9	0.73	
dih	15	0.48	
gih	12	1.00	
bou	8	0.40	*bah*
dou	12	1.00	deh *geh*
gou	16	0.68	
buu	9	1.00	bee
duu	–	–	guu
guu	–	–	

Table 4.4. Figure shows the performance of each detector node. Activation values and onset are given for the preferred syllable, and false alarms are shown. False alarms with an activation greater than the preferred syllable are in italics.

information from their preferred syllable.

A false alarm occurs when a detector fires with an activation value within an order of magnitude of its maximum activation on its preferred syllable, or with an $a > 0.1$ if it does not fire at all. Nineteen false alarms occurred. Of these, twelve were the result of firing on a syllable with the same stop consonant as the

preferred syllable, five shared a common vowel, and two showed no relationship. This would seem to indicate that the detectors were primarily confusing the stop consonant part of the utterance. These results should be viewed with caution since the number of both detection events and errors is small.

Analysis of the performance of the training and testing phase suggested that the majority of the false alarms could be corrected by the veto recognition network. As mentioned in Chapter 3, the recognition network consists of twenty–four recognizer nodes, each assigned a preferred syllable. A node contains an excitatory connection to the end of the delay line originating from its detector node, and a set of veto connections to the other detector nodes. The weights for the excitatory connections are set by looking at the maximum activation value of the corresponding detector node. The excitatory activation of a recognizer node is computed by a sigmoid squashing function that approaches a value of $a = 1$ if the excitatory input is above a threshold set to $\theta = 1$. The weight of the excitatory connection for a recognizer node is normally set to $1/(max - \delta)$, where max is the maximum activation of the detector node on its preferred syllable and δ is a value that ensures the excitatory input will exceed the threshold, giving a value of $a > 0.9$ for a recognizer node. For example, the excitatory weight for the recognizer node for "bae" is set to 1/0.4.

The thresholding effect of the recognizer nodes effectively cancels out any false alarms whose activation values are less than $max - \delta$. For example, the detector for "bae" misfires on the syllable "bah" with a maximum value of 0.3. The recognizer for "bae" correctly fires on its preferred syllable, and will not fire on a "bah" even though the detector is firing because the excitatory input never

exceeds the threshold. The weights for the nodes "duu" and "guu" are set low enough so that the recognizers never fire.

The syllables in italics in table 4.4 are false alarms whose maximum activations are greater than that detector's maximum activation for its preferred syllable. These false alarms are not eliminated by thresholding and must be vetoed. For example, the detector for "gah" fires on that syllable with a value of 0.81, but also misfires on "gae" with a value of 0.9. This error can be corrected if the recognizer node is prevented from firing when a "gae" is input. Since the detector for "gae" fires on that syllable with a value of 0.12, it can prevent the recognizer for "gah" from firing and the error can be eliminated. This is done by enabling the veto connection between the detector node for "gae" and the recognizer node for "gah". A veto connection is enabled with a threshold value, in this case 0.1, corresponding to the activation of the vetoing detector. If the value on that veto connection exceeds the threshold the recognizer is vetoed and cannot fire. It is essential that the node be vetoed before it receives its excitatory input. This is accomplished by the delay lines between a detector node and its corresponding recognizer, which gives the detector on the veto connection time to reach its maximum before the excitation arrives.

False alarms are eliminated by enabling a veto connection for each of the italicized false alarms in table 4.4, and the network correctly recognized twenty–two of the twenty–four syllables from the testing corpus for a score of 91.7%. When the network was tested on the training corpus it achieved 100% detection with no errors (no veto network was constructed). There are situations where the veto network will not be able to eliminate some false alarms, and these will

Test Repetition	Correct Detections		False Alarms	False Alarms After Veto Network	Recognition Score
Avg	22	91.7%	19	0	91.7%
1	22	91.7%	21	1	87.5%
2	17	70.8%	25	2	62.5%
3	20	83.3%	27	0	83.3%
4	20	83.3%	25	0	83.3%
5	15	62.5%	35	0	62.5%
Overall	19.3	80.6%	25.5	0.5	78.5%

Table 4.5. Detection, recognition, and misfire scores for average–only and raw–test expermiments, as well as mean scores for all six runs. Recognition score is $(correct - errors)/24$.

be discussed in the following section.

4.3 Raw–Test Experiments

The five raw–test experiments tested the network on each of the raw repetitions. In each experiment one of the raw repetition samples was selected for testing and the network was trained on the remainder of the data corpus. All parameters were the same as the raw–average experiment. The results of the raw–test and raw–average experiments are summarized in table 4.5. Detector nodes fired on their preferred syllable in all six experiments with a mean of 80.6%. Recognition scores after the construction of the veto recognition network had a mean of 78.5%.

It is clear from the results that the veto recognition network cannot eliminate all errors from the detection phase. There are two ways that this can happen, and both arise in tests on repetitions one and two. The first type occurs when two detectors misfire on each others' preferred syllable. In testing on repetition one, this occurs when the detector for "bey" fires on a "dey" and *vice versa*. If

the false alarm cannot be eliminated by thresholding, as is the case here since the maximum activation for the errors is greater than that for the correct syllable, the recognizer nodes cannot enable veto connections for either detector. If both veto connections are enabled the detectors would prevent each other's recognizer from firing on the correct syllable. There occasionally is a way out, however, when another detector misfires on one of the syllables. In this experiment the detector for "bou" misfires on a "bey", and by enabling the veto connection between the "bou" detector and the "dey" recognizer the error is eliminated. No other detector misfires on a "dey", and this error remains.

A second way that the veto network can fail to block an error occurs in the experiment testing the second repetition. In this experiment, the detector for "deh" misfires on a "bih". The problem is that the detector for "bih" does not fire at all on that syllable, and the recognizer for "deh" has no way of vetoing the error. No other node misfires on a "bih", and the error is left uncorrected. This is the cause of the two remaining errors in the experiment.

The types of detector errors for all experiments are shown in table 4.6. A detector can misfire on a syllable that has the same consonant, vowel, or shows nothing in common with a preferred syllable. These are shown in their respective columns. This table shows that the identity of the phonemes can have a slight influence on whether a detector will make an error. Both the number of false alarms related to consonants and those related to vowels were slightly greater than random chance for each type of error (chance being 33.3% for consonants and 12.5% for vowels). This shows that the network is able to extract features from both phonemes in a syllable and that syllables showing one or another of

Repetition	Total Errors	Consonants		Vowels		Unrelated	
Avg	19	12	63.2%	5	26.3%	2	10.5%
1	21	7	33.3%	7	33.3%	7	33.3%
2	25	12	48.0%	6	24.0%	7	28.0%
3	27	11	40.7%	6	22.2%	10	37.0%
4	25	9	36.0%	4	16.0%	12	48.0%
5	35	14	40.0%	9	25.7%	12	34.3%
Overall			43.5%		24.6%		31.8%

Table 4.6. Types of errors from all six experiments

these phonemic features were more likely to cause a detection error.

4.4 Parametric Effects

The effects of the parameters of the learning method were studied to give some idea of the robustness of the procedure. The exposure window and the number of training cycles can vary within an experiment. The reward and penalty constants that serve as expected values remain constant in an experiment but are different for the average–only experiment. Their effects will be discussed as well.

The exposure window varies the most within an experiment, depending on the length of time required to reach a steady–state value. The particular values for both the exposure window and the number of training cycles for all but the average–only experiment are seen in table 4.7. The length of the exposure window can have an effect on the strength of a detector node's activation and, in some cases, the errors that it makes. The length of the window also affects the amount of time needed for the learning process, and an attempt was made to keep it as short as possible for the sake of efficiency.

An example of the effects of the exposure window can be seen with the

Syllable Detector	Exposure Window	Training Cycles
bae	20	4
dae	20	4
gae	20	4
bah	20	4
dah	20	4
gah	25	4
bee	15	10
dee	25	4
gee	30	4
beh	20	4
deh	20	8
geh	20	10
bei	25	4
dei	20	8
gei	30	4
bih	18	4
dih	18	4
gih	25	4
bou	18	4
dou	32	4
gou	32	4
buu	20	4
duu	25	4
guu	30	4

Table 4.7. Exposure window length and number of training cycles used to train each detector in all six experiments

detector for "gae". With the exposure window set to twenty time cycles, this detector reaches a maximum activation of 0.12 at the eighteenth time cycle. When the exposure window is increased to twenty–five, the maximum activation becomes 0.8 at time cycle twenty, with no errors in either case. The cause of this lies with the eligibility trace mechanism in the learning algorithm. A particular connection, when activated, is eligible for weight modification as the

eligibility trace decays for a few time cycles after its deactivation. Increasing the exposure window allows the weights of the connections to be enhanced for a few more time cycles within a training cycle, increasing that connection's effect. The length of time required to reach steady–state is longer for syllables beginning with /d/ or /g/, and for those detectors that use both transition and steady–state information, the extra few time cycles can make a difference.

Increasing the exposure window can either reduce or increase the number of errors. In some cases the window must be lengthened for a detector to fire at all, eliminating the miss error. The effect on false alarms can vary. The detector for "dah" fires with a maximum value of one, and it misfires on "gah" and "bae" with an exposure window of twenty time cycles. Increasing the exposure window to twenty–five does nothing to the maximum activation for "dah", increases the false alarm activation for "gah", and eliminates the false alarm for "bae".

The number of training cycles affects the strength of activation of a detector node. A few nodes show a barely measurable activation after four training cycles, but this value may be increased with more presentations of the input data. The length of the learning process normally enhances both excitatory and inhibitory effects, and seems to amplify the behavior of the node. After about twelve to fifteen training cycles, further presentations of the training corpus have little effect, as the weights are changing much more slowly, if at all. The learning process normally was continued until either acceptable behavior or quiescence was achieved.

The reward and penalty constants that serve as the expected value to the learning method have some effect on the learning process, although they remain

unchanged during an experiment. Increasing the reward value enhances excitatory effects and can cause the node to activate earlier in the learning process. The reward value is smaller in the average–only experiment, and is increased in the other experiments to enhance the detectors' activation values. Since in these experiments the network has never seen the test data, the larger reward value allows the detector to fire on a smaller number of common features than is available when it is trained and tested on the same data. The effect of the reward is tied to the threshold of the node, and if it is too low the node will never fire. If the reward is too high inhibitory connections will have little effect, and the node will fire on almost every input. A similar argument can be made for the penalty value. If it penalizes the node too much, it will wash out excitatory connections and the node will never fire. Since the node is exposed in training to more penalty situations than rewards, the magnitude of the penalty constant is less than that of the reward. Acceptable values for the constants were found after two to three experimental trials.

In summary, SYREN was able to recognize 100% of the testing corpus in the average–only experiment. Each of the detector nodes was using transition information, and about half used only this information. Recognition was accomplished 91.7% of the time in the raw–average experiment, and the majority of the detector nodes were concentrating on information that contained both transition and steady–state information. The veto recognition network was able to eliminate all 19 false alarms in this experiment. The overall recognition rate from both the raw–test and the raw–average experiments was 78.5%, and the veto recognition network was able to eliminate 113 of the 116 false alarms in all

experiments. An analysis of the types of false alarms shows that similarities in either a vowel or consonant increased the probability that a detector would make an error.

5. Discussion and Implications

The previous chapter discussed the performance of SYREN on the syllable recognition experiments. To understand the implications of the system's behavior it is necessary to compare it to other work of this nature. Unfortunately, there are few models for comparison due to the many different ideas that make up SYREN. It is possible, however, to divide the system into a few parts and evaluate those parts from computational, neurophysiological, and psychophysical standpoints. The evaluation will look at two parts of SYREN. One involves the performance of the motion detector mechanism, including how the veto network mechanism performs in detecting acoustic motion. The performance of SYREN in syllable identification and its relation to speech recognition will serve as another point for analysis.

The first section of the chapter deals with the implications of the motion de-

tector mechanism. The next discusses the performance of the syllable detection apparatus in relation to both connectionist and linguistic models, followed by a discussion of the importance of the ideas of this project to speech recognition. The limitations of the system and the weaknesses of some of the initial assumptions are discussed in light of possible improvements to SYREN. The final section discusses the implications of this work to other problems in computer science and neuroscience.

5.1 Discussion of the Motion Detectors

The motion detectors effectively determine the direction of the change of formant center frequencies and discriminate specific rates of change through the use of veto inhibition and characteristics of dendritic computation. Detectors discriminate transition rates of 1.33, 2.0, 4.0, 8.0, 12.0 and 16.0 Hz/ms, corresponding to the transitions slopes in figure 3.3. In fact, as mentioned in Chapter 3, cooperative firing patterns of detectors for different slopes allow detection of intermediate rates between the transitions assigned to different detectors. The fine frequency and temporal selectivity of the input data allows for very sensitive detection of transition rates.

Computationally this appears to be a highly effective mechanism. The question remains whether this is sufficient for the task of syllable detection. There are a few ways to answer this question. One involves psychophysical studies of listeners' abilities to discriminate the rate and direction of frequency change, and another looks at neurophysiological studies of the auditory nervous system to see if neurons can respond to similar transitions. Both types of studies in-

volve the presentation of steady–state and frequency modulated tones to subjects in most cases. In psychophysical experiments, subjects' responses to the tones are measured, while the behavior of neurons in (animal) subjects is the focus of neurophysiological studies.

5.1.1 Psychophysical Studies

Psychophysical studies center around subjects' abilities to detect if a stimulus is changing in frequency, to detect if it is rising or falling, and in a few cases, to detect the rate–of–change of the frequency. Responses indicate whether they can identify differences in the frequency transitions as well as finding ranges of stimuli where detection is optimal.

Brady and House [17] tested whether subjects were merely averaging initial and final frequencies or were using some strategy involving tracking movement when responding to transition events. Subjects were presented with stimuli consisting of upward and downward frequency glides of rates of 10 and 25 Hz/ms, and were asked to adjust a steady–state tone to match the pitch of a transition. They found that subjects tended to adjust the matching frequency to the terminal portion of the transition, indicating that they were responding to something other than the average of the initial and final frequencies. They mentioned that this could be significant to speech, emphasizing the better detection of transitions in CV compared to VC syllables. This conclusion must be regarded with caution since a single tone bears only a superficial resemblance to a speech–like sound. What is significant is that a subject's response is affected by frequency transitions.

Sergeant and Harris [101] examined listeners' abilities to detect a variety of frequency glides, from slow transitions on the order of several seconds for a small frequency change, to fast transitions with durations of less than 30 ms. They found that subjects could detect a transition and identify its direction over a wide range of glide rates. They show that the auditory system can detect transitions of rates and durations common to formant transitions of stop consonants. This led them to conclude that there exists a mechanism in the auditory system for the detection of frequency transitions, and that it is most effective for high rates ($>$ 1Hz/ms) of short duration.

Further support of the notion of specific mechanisms for the detection of acoustic motion comes from Gardner and Wilson [39] who found strong evidence for the existence of direction–specific channels in the auditory system. Their experiments concentrated on the effects of repeated presentations of upward or downward FM sweeps on the ability to detect sweeps in the same or opposite direction. They argued that if direction–specific channels did not exist, the adaptation resulting from presentation of sweeps in one direction would affect the ability to detect sweeps in both directions. They found that repeated presentation of a sweep in one direction makes it more difficult to detect a sweep only in that direction, and has no effect in the opposite direction.

There have been relatively few studies on listeners' abilities to distinguish between different rates of transitions with the sensitivity exhibited by the motion detectors in SYREN. This is partially a result of a confound in evaluating the results of stimuli of different rates. If two transition rates are presented, they must differ either in the duration of the transition or in the frequency range

swept. This makes it more difficult to assess the effects relative to the length of the frequency window or the duration of the stimuli, both of which affect a subject's ability to discriminate the transitions.

Pollack [92] raised this point and attempted to study the combination of the two factors. He found that for short transition times, those corresponding to the formant transitions in SYREN, rate sensitivity was primarily a function of the differences in frequency, or the range of the frequency window. This correlates well with the motion detection mechanism in SYREN, which measures frequency differences over a fixed time frame for each detector.

Nabelek and Hirsh [87] studied listeners' abilities to discriminate transitions of different rates by determining if a test stimulus is the same as a reference stimulus. They found that optimum discrimination scores depended on the duration of the stimulus, with smaller transitions requiring longer durations. Rates similar to those for the CV syllable experiments had optimum durations of between 20–30 ms. This is close to the time frames of the transitions required to fire SYREN's motion detectors. These durations were relatively insensitive to where in the frequency range the transitions took place.

5.1.2 Neurophysiological Studies

In the discussion in the previous section, no study was found relating a listener's ability to discriminate the rates of frequency change with the sensitivity of SYREN's motion detectors. Neurophysiological studies also do not provide enough information to compare the overall performance. This is due to a lack of neurons available for any one project. Quantitative studies using intracellular

recordings are hard to reproduce, since the sample size, the number of neurons recorded, is too small to yield reliable results. Nevertheless, it is possible to make a few statements on the performance of the motion detectors and the mechanism of veto inhibition.

Most cells that show selectivity to rates and directions of frequency changes are not as sharply tuned to their preferred transitions as the detectors of SYREN, when responses are measured in firing rates. These studies are with animals, however, and comparisons to speech processing should be made with caution since the speech signal is a distinctly human phenomenon. In cat auditory cortex, cells were found that preferentially responded to frequency glides of pure tones of 0.05, 0.1, 0.2, 0.4, and 0.8 kHz/ms [79]. Many cells showed a limited response to transitions for the entire range, and responses to preferred rates could differ by as little as 30% to the next rate measured. The rates of these transitions are roughly two orders of magnitude faster than those found in speech, however. Studies of cells in the auditory nerve and cochlear nucleus were done with rates more closely resembling those of speech sounds [84, 105, 104, 18] and found similar effects.

Mechanisms for frequency transition detection are still under investigation. Many proposals require some form of inhibitory process for direction and rate sensitivity, especially for transition rates found in speech [18, 68]. This seems to fit a pattern of increased asymmetry and specificity of cell responses to direction and rate of movement as one moves higher in the auditory pathway [18]. Interneurons are required to deliver inhibition, and the cochlear nucleus is the first major structure where synapses occur. As one moves higher in the pathway, there is a greater chance to be affected by inhibition and a greater probability of

response asymmetry and selectivity [18].

5.1.3 Discussion of the Mechanisms in SYREN

Although none of the evidence above indicates that veto inhibition is the mechanism for motion detection in the auditory system, there is no reason not to use it in the construction of the acoustic motion detectors. It may be that the detectors used in SYREN are much more sensitive and selective to acoustic motion than those found in natural systems. This is not necessarily something to avoid since the goal of this research is not to faithfully model the nervous system but to use ideas from it in the implementation of artificial systems. The extra power available may lead to better recognition.

One result from of psychophysical studies of the detection of frequency transitions is that responses are relatively insensitive to variations in intensity [86]. In SYREN the intensity of the formant centers could be varied by using continuous rather than binary input values. Small changes in intensity should have little effect on the motion detectors since only the time course of the rise of activation is affected by variations in the input level. As long as the variations are within, say, 80% of the normal values, a node will still reach a maximum value in time to activate the next node in the branch.

An interesting property arises when looking at the time course of activation of nodes in the motion detectors. Strong excitation causes a node to reach a value near its maximum in four to seven update cycles, corresponding to a time course of from 2.8 to 5 ms of time (there are 7 update cycles for each 5 ms time slice). Nodes decay to a value near their resting activation in about 24 update cycles

or 17 ms. Koch has calculated timings for membrane potentials for patches of dendritic membrane in cat lateral geniculate nucleus [58]. He found that the rise time for the effect of excitatory synaptic potentials, that is, the time required for the membrane to react to the synapse, was on the order of 1.6 to 3.7 ms, and the decay times were about 9 to 25 ms. This corresponds fairly well to the behavior of the nodes in the motion detectors. Although these nodes are meant to be analogous to patches of dendritic membrane, whether or not this is coincidental, or a form of convergent evolution, is open to speculation.

5.2 Discussion of Syllable Recognition Performance

As with the motion detector analysis, there are few systems similar to SYREN to use to compare syllable detector performance. Dynamic connectionist models for speech recognition are only now becoming a popular research topic. There are, however, a few systems which bear discussion. Linguistics and psychology offer another area of comparison, although it is hard to relate results of studies of human subjects to results from artificial implementations. Nevertheless, the performance of SYREN will be compared to results from both areas.

5.2.1 Connectionist Models for Speech

The only model found in the literature that is applied to a problem similar to this research is the temporal flow model of Watrous and Shastri [126]. Their model is adaptive, using the back propagation method generalized to feedback connections on the hidden units and the output nodes. In one of their experiments, the data consisted of five repetitions of eighteen voiced stop consonant CV syllables produced by a single (presumably male) speaker. These use the same consonants

and six of the vowels tested with SYREN. Syllables were presented in a left–to-right fashion with input nodes updated once per time slice. Input consisted of fast Fourier transform coefficients, with the region of formant transitions for the stop consonants apparently segmented by hand from the vowel. The network is trained to recognize the consonants and vowels rather than the entire syllable.

For consonant identification, the network achieved an error rate of 0.11 errors per output unit per token for each time slice. For vowels, it achieved an error rate of 0.004 after 290 learning trials [126]. The average error rate for all experiments in SYREN was 0.008, with the lowest error rate of 0.004 for the Raw–Average experiment. This is after a maximum of only 12 learning trials. The error rate in SYREN may be skewed higher from the method of scoring a miss error. If a unit fails to fire on its syllable for an exposure window of 25 time slices, it scores 25 errors. It is unclear how Watrous and Shastri scored errors in this situation. They also segmented formant transitions by hand "to decrease the computational load of the optimization algorithm, [126]" whereas SYREN concentrated on the formant transitions by design.

A system by Homma, *et al.* [45], presented thirty synthetic phonemes as a temporal pattern to an adaptive network consisting of what they call dynamic formal neurons. This dynamic neuron differs from the formal neuron of McCulloch and Pitts [77] in the use of transfer functions in the place of connection weights and the use of correlation in the place of multiplication. The synthetic phonemes were combined into strings representing the first ten digits, and presented as input to the network one time slice at a time, in the same manner as SYREN. The network was trained and tested on this input matrix. It was also

tested on a noisy matrix which was made by subjecting the original matrix to random noise. One variant of their model failed to recognize three phonemes and misfired once on the noisy matrix and failed to recognize one phoneme with one additional false alarm on the training matrix. A second model achieved perfect recognition on the training matrix and had one false alarm on the noisy data. It is hard to compare this performance to SYREN, however, since Homma's network is trained and tested on the same data.

The TRACE model by Elman and McClelland [74, 29] is a combination of the blackboard architecture of HEARSAY [32] and ideas from connectionist processing. Its input consists of acoustic features sampled at 5 ms intervals of an utterance. In some senses it can be considered a dynamic model, since input is presented one time slice at a time, and previous time slices affect the processing of future input. Unfortunately the input feature nodes are copied for each time slice, and higher features, such as phonemes or syllables, are copied to span a slightly larger temporal window. These separate input nodes as well as phoneme and word level nodes stand for a particular moment in the utterance. Comparisons of the performance of TRACE are not appropriate since TRACE is designed to model psychological and linguistic behavior, not to serve as a general speech recognition system.

A final system to consider was designed by Elman and Zipser [30]. Although not a dynamic model, it can serve as a baseline for performance of static models in CV syllable recognition. Their model is a feed–forward adaptive network trained using back–propagation. Stimuli consisted of 505 input tokens of nine CV syllables from a single male speaker. The nine syllables were composed of the

consonants /b, d, g/ paired with the vowels /ii, ah, uu/. The entire utterance was presented at one time as input to the network. Training consisted of 100,000 learning cycles. The model, when trained for syllable recognition, achieved a score of 84% for clean data and 90% for data with added noise. It achieved 100% recognition on the training data. This is a static model, however, with the entire input signal available for the duration of the learning cycle.

5.2.2 Speech Perception Experiments

Speech Perception experiments measure a listener's ability to identify linguistic sound units in test stimuli. Although few conclusions can be drawn between artificial and human recognition systems, linguistic data can give a reference point for evaluating an artificial recognition system's performance as well as comparing the qualitative effects of different types of stimuli. The experiments discussed here tested recognition of consonants and vowels in real and synthetic CV stop consonant syllables. These experiments manipulated the availability of transition information in isolation and in conjunction with other acoustic properties.

Pols and Schouten [94] originally attempted to determine whether listeners could identify certain synthetic frequency changes similar to formant transitions. Pilot results convinced them that evaluation of listeners' performance on this task is difficult, so they instead proceeded to investigate how well subjects could use formant transitions excised from natural syllables in a speech recognition task. In one experiment, they studied Dutch listeners' performance in initial stop–consonant recognition by presenting the stimuli consisting of burst information only, transitions only, or a combination of both. They found that listeners

correctly identified approximately 70% of the initial stop consonants with transition information alone, 85% with only the burst, and 98% with both cues. They concluded that transition information can be used alone to a limited extent, and that it greatly augments performance with burst cues.

Formant transitions seem to help in the identification of vowels. Strange, *et al.*, tested listeners' abilities to recognize vowels in a variety of environments including bV syllables (syllables beginning with a "b") [113]. Vowel recognition scores averaged 85% in bV syllables compared with 75% for vowels that were pronounced in isolation. In another study [114] vowels were correctly recognized 94% of the time even when the vowel portions were masked by replacement with noise in a bVb syllable. This further supports the view that dynamic properties of the rapid transitions carry useful information about the entire syllable.

The above studies demonstrate the difficulty experienced by the most proficient speech recognition systems, humans, in isolated syllable recognition tasks. They also demonstrate how performance is improved with the presence of additional information compared with the presentation of only one type of cue. In the experiments with SYREN, recognition is achieved with only minimal acoustic information for each spectral slice. No information is available from formant bandwidths and intensities, and no cues are available as to burst onset and the onset of voicing. Instead, dynamic information is provided that is known to be useful in speech perception.

5.3 Limitations of the Research

As with any research, there are some points in the design of SYREN that are

open to question. Two major problems are apparent. The first involves the nature of the input, and the availability of smooth formant centers in a real–time speech processing system, as well as the appropriateness of formant–based representations at all. This is mostly a linguistic issue. The second problem centers on the structure of the syllable detection network, the learning algorithm, and the *ad hoc* nature of the veto recognition network. Each of these problems is discussed and possible solutions are suggested. These problems are not major setbacks, however, but provide starting points for future research that further addresses the nature of temporal processing in dynamic connectionist models as well as computational methods used in speech recognition.

5.3.1 Linguistic Issues

One assumption made in this project is that formant center data could be reliably obtained from an analysis of the speech signal. The formant data used in this project were visually measured from sound spectrograms [52]. There are at least two reasons why this might not hold. The first is the difficulty in extracting the formant centers. The second is whether formants are appropriate at all for speech recognition. Fortunately, there are alternative input representations available that still allow the use of the veto network in the detection of spectral transitions.

The importance of formants in a phonetic analysis arises from their appearance in wide band spectrograms and their effectiveness as control parameters in speech synthesis. They are prominent features in wide–band spectrograms for vowel and sonorant stretches, but are usually absent during fricatives and stops.

When they are present, their measurement and detection can be extremely difficult, both visually in spectrograms, as well as automatically. If two formants are close together, they may appear as a single wide band. The resolution of both spectrograms and the frequency sensitivity in the nervous system make it difficult to track and identify formants in this case [13]. Furthermore, their detection is affected by low intensity and high noise level. Even visual extraction of formant centers is difficult in cases of recordings of low signal to noise ratio.

If a speech signal at a particular instant is plotted in a frequency *vs.* intensity graph, formants can be seen as mountain ranges or wide bands of high signal intensity. Spectral peaks emerge as dominant frequencies possessing the highest intensity in a given region. Many times they can be seen near the formant centers, but they are also present at frequencies that do not contain formants. Several methods exist for the identification of spectral peaks. An interesting method that could be convenient for SYREN is the *dominant frequency* technique. Carlson and Grandström [20] have devised a method to identify prominent spectral peaks and sharpen the representation into what appears to be tight dark bands in the spectrogram. There are normally more peaks than formants in a signal, and during periods of voicing the transitions and steady–state portions produced by this method appear much like the formant centers used in SYREN. This is a useful input representation for study.

Klatt has observed that "changes of formant frequencies are the most important characteristic that causes subjects to report changes in phonetic quality [57]." Frequency changes are even more important then changes in intensity or bandwidth of frequency regions. Spectral peak measurement gives a represen-

tation that is relatively insensitive to bandwidth but may depend on intensity. The effect of intensity can be reduced in the Carlson and Grandström method through proper thresholding techniques, forming a binary representation of the presence of peaks at a particular frequency band.

Connectionist methods exist as well for the extraction of certain features of the acoustic signal. Shamma [102] has devised a mechanism based on cochlear mechanics. Dominant frequencies can be identified through a hand–constructed connectionist model using lateral inhibition and feedback connections, with a fine resolution in both space and time. Output of this type of network could serve as input to the motion detector network. The motion detector network would then detect the changes of dominant frequencies rather than formant centers. The veto network is relatively insensitive to small variations in intensity, and adaptive thresholding techniques can eliminate the effect of large changes in intensity.

5.3.2 Issues in Representation and Learning

Decisions were made concerning the target representation for the syllable detectors, the learning algorithm, and the use of a veto network to recover from deficiencies in the learning process. These decisions were made based on constraints of the problem, computational resources, and the focus of the project. The rationale behind some of these decisions is discussed below to provide a motivation for future projects to improve performance.

SYREN uses a single–layered adaptive network for the first level of syllable detection. This was chosen primarily to reduce the computational demands of the training phase. Single–layer networks have inherent limitations in processing

abilities [83] and although the network gave adequate performance in the task it certainly can be improved. One obvious modification is to use a multi–layer network for improved computational capacity. This would require a learning method different from the one currently used.

The back propagation method [97] has seen some success in multi–layer feed–forward networks. The temporal nature of the recognition process imposes certain problems on a totally feed–forward network, however. Hidden units, those nodes in the middle layers of a multi–layer network, form an intermediate representation of the pattern of activation of input nodes. The output nodes use this information to decide whether or not to activate. The characteristics of speech imply that the hidden nodes may need to respond to intermediate features of the input signal at different times. These responses must be preserved somehow for simultaneous use in the output nodes.

Delay lines and feedback from recurrent connections are two methods used to address the problem, but they impose additional computational complexity to the learning method in both space and time. These solutions are common in temporal sequence detection methods, but most are only single layer, self–organizing systems. Work is being done to extend these methods to multi–layer recurrent nets in supervised systems [2, 50] and merit further study.

A question can be raised as to whether any form of supervised learning is appropriate for speech recognition problems. In supervised learning algorithms, the behavior of each output node must be known in advance, and is given to the network at each time slice of the learning cycle. This was possible in SYREN's task since the identity of each syllable in the data corpus was known. In a

less limited task where an utterance is composed of several syllables, supervised learning requires that the signal be pre–segmented so that the identity of constituent parts is available during the training process. This is unreasonable in complex tasks with large training sets. Self–organizing methods do not suffer from this problem, but designers of a recognition system may still want to impose a particular output representation. Still, this is impossible without external interaction during the learning process. Algorithms do exist to allow a network to learn its appropriate output behavior [11, 118], but these methods need to be incorporated in feedback network architectures.

A localized representation for syllables was used here for both the adaptive network and the veto recognition network, where each node stands for one syllable. This was chosen to simplify the determination of the expected value for the learning method. In a distributed representation the identity of a syllable is determined by patterns of activity of output nodes. In such a case a single output node may be active for more than one syllable. Even though SYREN was trained to use a local representation, the results of the training process showed many characteristics of a distributed representation. Each syllable resulted in a firing pattern composed of nodes that fire both correctly and incorrectly. Since the nodes did not fire at the same time, these patterns exhibited temporal variation. Thus, the patterns show signs of a distributed representation for a syllable, even though the network was trained to use a local representation.

The veto recognition network in SYREN exploits many of the distributed properties of the output of the adaptive network. Firing patterns over time of various detector nodes determine whether or not a recognizer node will activate.

This representation takes on a spatial dimension, determined by which node is firing, as well as a temporal dimension, determined by the onset of activation. A recognizer node can be vetoed by other detector nodes in the adaptive network. Normally, if a recognizer's detector node is firing on an incorrect syllable, it is vetoed by the activation of the detector node assigned to that syllable. In the case where the correct detector is not firing and cannot veto a node, another node that may be firing incorrectly can supply the veto inhibition. In this case detector errors are not necessarily detrimental, but may be useful and informative. This behavior may also be used to eliminate miss errors. A recognizer could be set up to fire when other detectors are firing incorrectly on a syllable and when the correct detector is silent. This is not currently implemented.

The temporal dimension of the output of the syllable detectors motivates the use of veto inhibition. In the nervous system this type of inhibition can veto excitation that arrives after the inhibitory synapse is activated. In syllable detection the node supplying the inhibition may activate before the excitation arrives. The veto mechanism allows for this behavior.

5.4 Contributions of the Project

Insights from neurophysiological mechanisms led to development of a computational tool for use in dynamic connectionist models. Neural veto inhibition and related characteristics of local dendritic computation have led to a novel sort of connectionist implementation that has been shown to be effective in the detection and identification of acoustic motion. This project has been useful in the study of temporal computation in connectionist models, and should promise insights

relevant to any system that processes data as a continuous stream.

5.4.1 Applications of Veto Inhibition

Speech recognition is not the only area that can benefit from the tools of this project. Veto inhibition has been implicated not only as part of the mechanism for visual motion, but has been found at higher levels of the visual system, such as surface smoothing in the cortex [59]. Shunting inhibition has been shown to be part of the motion detection mechanism in the lateral geniculate nucleus [60] and even in the cortex [112]. It is claimed to help select retinal input in the lateral geniculate nucleus [58]. So it is clear that this mechanism is useful in a wide variety of visual tasks.

To be useful in real computing systems, the veto network from this project must be implemented to allow real time computation. The nature of connectionist models implies that they should be implemented as massively parallel networks of simple computational units. Hardware implementations are complicated by the large amount of interconnectivity required in traditional connectionist models. These veto networks actually show much less connectivity, making their implementation in very large scale integration (VLSI) hardware seem tractable.

The implementation of some of these ideas has, in fact, been accomplished in a VLSI implementation for visual motion detection [49]. Using an architecture from a resistive network [78] and incorporating shunting inhibitory circuitry originally suggested for surface smoothing [59], a visual motion detector network has been implemented in hardware. This allows for low–level visual processing in real time, partially as a result of ideas like those used in this project. This

hardware implementation could be modified and applied to the task of acoustic motion detection as well. This could facilitate the development of real–time speech recognition systems.

This project has explored the use of veto inhibition primarily in perceptual processing, but this is not the only area of application. Inhibitory mechanisms similar to shunting inhibition may be useful in fine control of motor behavior in the cerebellum [72]. Higher cognitive functions can benefit as well. Minsky has illustrated the use of censors in controlling the flow of information between cognitive agents in the mind [81]. Essentially a censor blocks the transmission of information from one agent to another. This is an ideal task for the on-path, shunting characteristics of veto inhibition. The mechanism might also be used for blocking the execution of inappropriate alternatives in planning tasks.

The implementation of the veto mechanism has led to some ideas in the neurological study of aphasia. Some language–impaired children have difficulty recognizing syllables with rapidly varying dynamic cues such as CV syllables [121]. This implies separate mechanisms for the detection of fast and slow transitions. The veto network has been applied to rapidly changing stimuli, but may not be appropriate for small frequency changes over a period of seconds. Impairment of neurological structures that provide veto inhibition may be the cause of some types of aphasia if this is, in fact, a mechanism for the detection of rapidly varying acoustic stimuli. The veto network may provide a basis for neurophysiological experiments that can yield information both on language impairment and the detection of spectral change [22].

5.4.2 Temporal Processing and Speech Recognition

The processing mechanisms of SYREN are clearly useful in the recognition of stop consonant CV syllables. This project additionally provides insights that are necessary for full scale speech recognition systems.

A connectionist model for general speech recognition will need to deal with the speech signal in short temporal windows. The TRACE model [74, 29] accomplishes this by copying the network for each time slice of the signal. This is unreasonable on at least two fronts. The detection of input features and higher level features requires complex processing mechanisms. Replication of this mechanism throughout the network places a burden on computer architects who must fit these copies on VLSI hardware. Furthermore, this sort of method limits the system to processing utterances of a fixed length. In running speech a signal must be chopped into fixed–size pieces for digestion in the system, yet information can be lost across the boundaries of the pieces. Clearly a system must store information from previous time slices, but to do so requires the exploration of alternative mechanisms.

Continuous processing of stream input seems to be the best solution, where the input interface to the system is updated at each time slice. The storage of previous time slices and the representation of their temporal effects is a difficult problem in connectionist models. The issue is whether or not to explicitly represent time, or to incorporate some of its effects in the network architecture.

A key principle of this project is not to explicitly represent time in any single manner, but instead to place its effects at many levels. There is an explicit temporal representation in the delay lines used in several parts of the system,

but in each case these are used to preserve the sequence of abstract events, not simply to relate the events to particular point in real time. Since information in the delay lines has been subjected to various kinds of processing, it is not simply a buffer for the input signal. The two levels of delay lines store different kinds of information in order to support different kinds of decisions.

Temporal effects are also represented in the activation of nodes in the motion detector network. The effects of time are realized in the dynamics of the onset and decay of activation as well as in the communication throughout nodes in the network. Veto inhibition plays an important role in this by providing inhibitory effects that can begin and end at specific points in time. This allows flexibility in the temporal onset of informative events, both at the first level of the motion detectors and at the last level of syllable recognition. Simple inhibitory mechanisms from traditional models do not give this power and flexibility.

The effects of time in SYREN are found throughout the network in many different forms. The knowledge gained from these different temporal representations may be applied to connectionist implementations requiring dynamic interaction of components and information. This is useful in speech recognition, which requires the processing of many dynamic events. In the most general case the study of these processing mechanisms should lead to the development of a number of connectionist tools that can be applied to problems requiring computation in time.

6. Conclusions and Future Directions

This research has shown some of the methods that can be used in a connectionist model that must deal with temporal properties of its input. Speech recognition is a problem that requires temporal processing, and the mechanisms implemented have proven useful in the problem of syllable recognition. Neuroscience has contributed the principal mechanism that provides for the implementation of dynamic behavior. The contributions of this project need not be limited to speech. The ideas used here can be applied to other problems in perception and higher levels of cognition, and can provide a starting point for further research into the nature of temporal computation in connectionist models.

This chapter summarizes the achievements of the project and outlines possibilities for future research.

6.1 Achievements

A possible mechanism for visual motion sensitivity in the retina has been implemented for the detection of the rate and direction of transitions of formant centers. This mechanism incorporates a veto inhibitory mechanism coupled with properties found in small areas of dendritic membrane in cells of the nervous system. It is implemented as a veto network.

This implementation changes the neurological analogy of nodes and links in a connectionist model. Previously a node has been thought of as a simplified analog of a single neuron or even an assembly of cells. Links between nodes form connections between cells or assemblies. In this model a node in the motion detector network corresponds to a portion of a dendritic tree. Connections between nodes represent synapses from other cells and provide communication pathways along the branches of the dendrites. This change in neurological focus allows the implementation of more complex behaviors of a single cell, while still appealing to the computational guidelines of connectionist models, where nodes perform simple computations of activation levels and transmit this to other nodes in the network.

The structure of the network and the computations performed by the nodes permits the implementation of these mechanisms in VLSI architectures. This could lead to applications of the motion detector mechanism to real time speech recognition systems.

The formant motion detectors were used in the recognition of stop–vowel syllables. The system identifies the rates and directions of formant transitions from

formant center data from repetitions of twenty–four CV syllables. Transition information is fed to a single layer adaptive network which learns to associate this information with the identities of different syllables. Output from the adaptive network forms a sort of intermediate representation that is used by a final network for syllable recognition. This recognition network also uses veto inhibition that is similar to the mechanism used for acoustic motion detection.

Formant center tracks are divided into 200 frequency regions of 20 Hz each, and presented to the network in 5 ms time slices. Input nodes of the network see only one time slice at a time, and do not retain any information from previous time slices. This forces the network to maintain traces of previous information by properties of activation onset and decay, and through the dynamics of activation along delay lines. The network sees only short durations of the signal at any one time and must concentrate on those parts most useful for recognition. This is part of the temporal computation performed by the network.

Veto inhibition and delay lines are used in the final stages of processing for syllable recognition. The system uses patterns of activation from nodes in the adaptive network to determine the identity of particular syllables. These patterns have a temporal character, since nodes activate at different times during the presentation of the input. Veto inhibition permits the temporal integration of these patterns for effective syllable identification.

The network achieved perfect recognition when tested on its training corpus. When tested on previously unseen repetitions, it recognized an average of 79% of the syllables ranging from 92% to 63% in different experiments. The veto recognition network eliminated all but three of the 116 false alarms from the

adaptive network. This performance comes after a small number of training cycles. The majority of the detector units concentrate on a brief, early portion of the signal that contained both vowel and consonant information.

The number of training cycles used in each experiment depends on which syllable is being trained, but ranges from only four to ten presentations of the training corpus. Although the adaptive method used in the system is simple, it accomplishes its task quickly.

6.2 Where Do We Go From Here?

The initial successes of this project imply other avenues for exploration. These range from the development of better methods for representation and learning of temporal properties in connectionist models, to hardware implementations of the mechanisms of motion detection, and even to neurophysiological experiments.

Clearly the representation and architecture of the adaptive network needs to be explored. A multi–layer network is most certainly the answer, and sequence detection research indicates that feedback methods must be implemented as well. This will require more sophisticated learning algorithms to deal with this type of architecture. The previous chapter mentioned some of the possible methods, and there is much active research in this type of learning. Some of these methods, or possibly hybrids, are worth exploring to see if recognition behavior can be improved.

The representation of syllables in the adaptive network is another topic for study. Currently, a local representation is used for simplicity, while the final behavior of the subnetwork after training shows that it is developing a more

complex, distributed scheme. It may be fruitful to impose fewer initial constraints and allow the network more freedom to develop its own representations. This places a greater burden on the learning mechanism since supervised learning methods require the behavior of each output node to be known in advance. Methods which allow the network to predict its behavior are mentioned in Chapter 3, and provide a promising starting point.

The nature of the input to the network must be must be modified if application of the veto network to general speech recognition systems is envisioned. Methods discussed in Chapter 5 for isolating spectral peaks seem to be a promising starting point. Connectionist models outlined for the extraction of spectral qualities of the speech signal seem to be the best method for accomplishing this task. Availability of this type of information means that few modifications need to be made to the motion detector network.

The hardware implementation of the motion detector network and signal processing methods seems to be a promising line of research. These will allow real time processing of the speech signal in larger applications. The architecture of the motion detector network is fairly regular and simple and can be specified without the need for adaptive methods. This allows the implementation of the motion detector network in the same manner as the motion detection mechanisms for vision. These vision networks use many of the same mechanisms used in the formant motion detectors, and need only be implemented to detect one dimensional motion characteristic of acoustic motion. Hardware implementations of early acoustic signal processing methods, including motion detection, might provide a powerful tool for speech processing.

The ideas from the motion detectors can be applied to other areas as well. Tasks such as planning and motor control can benefit from notions of veto inhibition and more complex, neurally inspired processing methods. Here tools developed for vision and speech might be applied to tasks in different domains including those requiring higher–level cognitive processing.

Finally, ideas from this project can lead to experiments in neuroscience and speech science. The mechanisms studied have a few implications in the neural representation of the acoustic signal and the processes used in human speech understanding. The goal of this research is to use neurophysiological mechanisms to give insights for computational solutions. It is not intended to suggest that these specific mechanisms are used in the auditory system. Nevertheless, if similar mechanisms are used, it would imply the existence of certain structures and functional properties in the auditory nervous system. Thus, these results suggest experiments that look for this behavior in the areas of neurophysiology, and as mentioned, the study of language impairment. Experimental confirmation of these mechanisms would be rewarding, but even disconfirmation will yield insights into the complexities of the nervous system.

This project shows the promise in the application of neurophysiological mechanisms as computational techniques. Connectionist models are already reaping the benefits from the use of rather simple analogs of neural behavior, and a deeper understanding of the nervous system can yield rewards in computer science. This project shows the benefits of just one mechanism and reveals the potential available from the study of the mechanisms of intelligence.

References

1. Ackley, D. H., Hinton, G. E., Sejnowski, T. J., "A learning algorithm for Boltzmann machines," *Cognitive Science*, **9**, 1985, 147–169.

2. Almeida, L. B., "A learning rule for asynchronous perceptrons with feedback in a combinatorial environment," *Proceedings of the IEEE International Conference on Neural Networks*, San Diego, 1987, II–609–618.

3. Amthor, F. R., Oyster, C. W., Takahashi, E. S., "Morphology of on–off direction–selective ganglion cells in the rabbit retina," *Brain Research*, **298**, 1984, 187–190.

4. Anderson, C. W., "Learning and problem solving with multilayer connectionist systems," PhD. Thesis, COINS Technical Report 86–50, University of Massachusetts, Amherst, 1986.

5. Ballard, D. H., "Cortical connections and parallel processing: Structure and Function," *Behavioral and Brain Sciences*, **9**, 1986, 67–120.

6. Barlow, H. B., Levick, W. R., "The mechanism of directionally selective units in rabbit's retina," *Journal of Physiology*, **173**, 1965, 477–504.

7. Barto, A. G., "Learning by statistical cooperation of self interested computing elements," *Human Neurobiology*, **4**, 1985, 229–256.

8. Barto, A. G., Anaden, P., "Pattern recognizing stochastic learning automata," *IEEE Transactions on Systems, Man, and Cybernetics*, **SMC-15**, 1985, 360–374.

9. Barto, A. G., Anderson, C. W., Sutton, R. S., "Synthesis of nonlinear control surfaces by a layered associative search network," *Biological Cybernetics*, **43**, 1982, 175–185.

10. Barto, A. G., Jordan, M. I., "Gradient following without back–propagation in layered networks," *Proceedings of the IEEE International Conference on Neural Networks*, San Diego, 1987, II-629–636.

11. Barto, A. G., Sutton, R. S., Anderson, C. W., "Neuronlike adaptive elements that solve difficult learning control problems," *IEEE Transactions on Systems, Man, and Cybernetics*, **SMC-13**, 1983, 834–846.

12. Barto, A. G., Sutton, R. S., Brouwer, P. S., "Associative Search Networks: A reinforcement learning associative memory," *Biological Cybernetics*, **40**, 1981, 201–211.

13. Bladon, A., "Arguments against formants in the auditory representation of speech," In: Carlson, R., and Grandström, B., *The Representation of Speech in the Peripheral Auditory System*, Elsevier, 1982, 95–102.

14. Blumstein, S. E., and Stevens, K. N., "Perceptual invariance and onset spectra for stop consonants in different vowel environments," *Journal of the Acoustical Society of America*, **67**, 1980, 648–662.

15. Blumstein, S. E., and Stevens, K. N., "Acoustic invariance in speech production: Evidence from measurements of the spectral characteristics of stop consonants," *Journal of the Acoustical Society of America*, **66**, 1979, 1001–1017.

16. Borden, G. J., Harris, K. S., *Speech Science Primer*, Second Edition, Williams and Wilkins, 1984.

17. Brady, P. T., House, A. S., Stevens, K. N., "Perception of sounds characterized by rapidly changing frequency," *Journal of the Acoustical Soceity of America*, **33**, 1961, 1357–1362.

18. Britt, K., and Starr, A., "Synaptic events and discharge patterns of cochlear nucleus cells. II. Frequency–modulated tones. *Journal of Neurophysiology*, **39**, 1978, 179–184.

19. Carlson, N. R., *Physiology of Behavior*, Allyn and Bacon, 1986.

20. Carlson, R., Grandström, B., "Towards and auditory spectrograph," In: Carlson, R., and Grandström, B., *The Representation of Speech in the Peripheral Auditory System*, Elsevier, 1982, 109–114.

21. Charot, F., Frison, P., Quinton, P., "Systolic Architectures for connected speech recognition," *Rapport de Recherche*, No. 332 INRIA:Rennes, 1984.

22. Chase, C., Personal Communication.

23. Chun, H. W., "A representation for temporal sequence and duration in massively parallel networks: exploiting link interactions," *Proceedings of the 5th National Conference on Artificial Intelligence* (AAAI–86), Morgan Kaufman, 1986.

24. Church, A., *The Calculi of Lambda–Conversions*, Princeton University Press, Princeton, N.J., 1941.

25. Colmerauer, A., Kanoui, H., Pasero, R., Roussel, P., "Un systeme de communication homme–machine en Francais," Research Report, Groupe Intelligence Artificielle, Universite d'Aix–Marseille II, France, 1973.

26. Dehaene, S., Changeux, J.-P., and Nadal, J.-P., "Neural networks that learn temporal sequences by selection," *Proceedings of the National Academy of Sciences: USA*, **84**, 1987, 2727–2731.

27. De Mori, R., Lam, L., and Gillous, M., "Learning and plan refinement in a knowledge–based system for automatic speech recognition" *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI–9, 1987, 289–305.

28. Delattre, P. C., Liberman, A. M., Cooper, F. S., "Acoustic loci and transitional cues for consonants," *Journal of the Acoustical Society of America*, **27**, 1955, 769–773.

29. Elman, J. L., McClelland, J. A., "Exploiting lawful variables in the speech wave," in: Perkell, J. S., and Klatt, D. H., eds., *Invariance and Variability in Speech Processing*, Erlbaum, 1986, 360–380.

30. Elman, J. L., and Zipser, D., "Learning the hidden structure of speech," ICS Report 8701, UCSD, San Diego, 1987.

31. Eimas, P. D., Sequeland, E. R., Jusczyk, P., and Vigorito, J., "Speech perception in infants," *Science*, **171**, 1971, 303–306.

32. Erman, L. D., Hayes–Roth, F., Lesser, V. R., and Reddy, D. R., "The HEARSAY–II speech understanding system: Integrating knowledge to resolve uncertainty," *Computing Surveys*, **12**, 1980, 213–253.

33. Fant, G. M., *Acoustic Theory of Speech Production*, Mouton, 1960.

34. Farhat, N. H., and Miyahara, S., "Optical analog of two dimensional neural networks and their applications in recognition of radar targets," In: Denker, J. S., ed. *Neural Networks for Computing*, AIP conference Proceedings No. 151, 1986.

35. Feldman, J. A., Ballard, D. H., "Connectionist models and their properties," *Cognitive Science*, **6**, 1982, 205–254.

36. Feldman, J. A., "Memory and change in connectionist networks," Rochester University Computer Science Department Technical Report No. 196. 1981,

37. Fu, K. S., "Learning control systems–review and outlook," *IEEE Transactions on Automated Control*,

38. Fukushima, K., "A model of associative memory in the brain," *Kybernetik*, **12**, 1973, 58–63.

39. Gardner, R. B., and Wisler, J. P., "Evidence for direction–specific channels in the processing of frequency modulation," *Journal of the Acoustical Society of America*, **66**, 1979, 704–709.

40. Grossberg, S., "How does the brain build a cognitive code?," *Psychological Review*, **87**, 1980, 1–51.

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

42. Hinton, G. E., "Learning distributed representations of concepts," Proceedings of the Eighth Conference of the Cognitive Science Society, Erlbaum. 1986.

43. Hinton, G. E., Anderson, J. A., *Parallel Models of Associative Memory*, Erlbaum. 1981.

44. Holland, J., *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, 1975.

45. Homma, T., Atlas, L. E., Marks II, R. J., "An artificial neural network for spatio–temporal bipolar patterns: Applications to phoneme classification," In: *Neural Information Processing Systems, Natural and Synthetic*, (to appear).

46. Hopfield, J. J., "Neural networks and physical systems with emergent collective computational abilities," *Proceedings of the National Academy of Sciences: USA*, **79**, 1982, 2554–2558.

47. Hubel, D. H., and Weisel, T. N., "Receptive fields and functional architecture of monkey striate cortex," *Journal of Physiology*, **195**, 1968, 215–243.

48. Hubel, D. H., and Weisel, T. N., "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," *Journal of Physiology*, **160**, 1962, 106–154.

49. Hutchinson, J., Koch, C., Luu, J., Mead, C., "Computing motion using analog and binary resistive networks," *IEEE Computer*, March, 1988, 52–63.

50. Jordan, M. I., "Serial order: A parallel distributed processing approach," ICS Report 8604, UCSD, San Diego, 1986.

51. Kewley–Port, D., "Time–varying features as correlates of place of articulation in stop consonants," *Journal of the Acoustical Society of America*, **73**, 1983, 322–335.

52. Kewley–Port, D., "Measurement of formant transitions in naturally produced consonant–vowel syllables," *Journal of the Acoustical Society of America*, **72**, 1982, 379–389.

53. Kewley–Port, D., and Luce, P. A., "Time–varying features of initial stop consonants in auditory running spectra: A first report," *Perception and Psychophysics*, **35**, 1984, 353–360.

54. Kewley–Port, D., Pisoni, D. B., and Studdert–Kennedy, M., Perception of static and dynamic acoustic cues to place of articulation in initial stop consonants," *Journal of the Acoustical Society of America*, **73**, 1983, 1779–1793.

55. Kewley–Port, D., Personal communication, 1987.

56. Kirkpatrick, S., Gelatt, C. D., Vecchi, M. P., "Optimization by simulated annealing," *Science*, **220**, 1983, 671–680.

57. Klatt, D. H., "Speech processing strategies based on auditory models," In: Carlson, R., and Grandström, B., *The Representation of Speech in the Peripheral Auditory System*, Elsevier, 1982, 181–196.

58. Koch, C., "Understanding the intrinsic circuitry of the cat's lateral geniculate nucleus: electrical properties of the spine–triad arrangement," *Proceedings of the Royal Society of London: Series B*, **225**, 1985, 365–390.

59. Koch, C., Marroquin, J., Yuille, A., "Analog neuronal networks in early vision," *Proceedings of the National Academy of Sciences*, **83**, 1986, 4263–4267.

60. Koch, C., Poggio, T., "Biophysics of computation: Neurons, synapses, and membranes," In: Edelman, G.M., Gall, W.E., Cowan, W.M., eds, *New Insights in Synaptic Function*, Neurosciences Research Foundations, Inc., Wiley, 1986.

61. Koch, C., Poggio, T., Torre, V., "Computation in the vertebrate retina: Gain enhancement, differentiation, and motion discrimination," *Trends in Neurosciences*, **9**, 1986, 204–211.

62. Koch, C., Poggio, T., Torre, V., "Nonlinear interactions in a dendritic tree: Localization, timing, and role in information processing," *Proceedings of the National Academy of Sciences: USA*, **80**, 1983, 2799–2802.

63. Koch, C., Poggio, T., Torre, V., "Retinal ganglion cells: A functional interpretation of dendritic morphology," *Philosophical Transactions of the Royal Society of London: Series B*, **298**, 1982, 227–264.

64. Kohonen, T., "An introduction to neural computing," *Neural Networks* **1**, 1988, 3–16.

65. Kohonen, T., *Self Organization and Associative Memory*, Springer–Verlagg, 1984.

66. Kohonen, T., Oja, E., Pelka, L., "Storage and processing of information in distributed associative memory in the brain," In: Hinton, G. E., Anderson, J. A., *Parallel Models of Associative Memory*, Erlbaum. 1981.

67. Kurogi, S., "A model neural network for spatiotemporal pattern recognition," *Biological Cybernetics*, **57**, 1987, 103–114.

68. Lacerda, F., Moreira, H. O., "How does the peripheral auditory system represent formant transitions," In: Carlson, R., and Grandström, B., *The Representation of Speech in the Peripheral Auditory System*, Elsevier, 1982, 89–94

69. LeCun, Y., "Une procedure d'apprentissage pour reseau a sequil assymetrique," *Proceedings of Cognitiva*, **85**, 1985, 599–604.

70. Lindblom, B. E. F., Studdert–Kennedy, M., "On the role of formant transitions in vowel recognition," *Journal of the Acoustical Society of America*, **42**, 1967, 830–843.

71. Lowerre, B., and Reddy, R., "The HARPY speech understanding system," In: Lea, *Trends*, 1980, 340–360.

72. Marr, D., "A theory of cerebellar cortex," *Journal of Physiology*, **202**, 1969, 437–470.

73. McClelland, J. L., "Putting knowledge in its place: A scheme for programming parallel processing structures on the fly," *Cognitive Science*, **9**, 1985, 113–146.

74. McClelland, J. L., and Elman, J. L., "Interactive processes in speech perception: The TRACE model," In: McClelland, J. L., Rumelhart, D. E., eds. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, Vol 2: Psychological and Biological Models, MIT Press, 1986.

75. McClelland, J. L., Rumelhart, D. E., *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, Vol 2: Psychological and Biological Models, MIT Press, 1986.

76. McClelland, J. L., Rumelhart, D. E., "An interactive activation model of context effects in letter perception Part 1: An account of basic findings," *Psychological Review*, **88**, 1981, 375–407.

77. McCulloch, W. S., and Pitts, W. H., "A logical calculus of ideas immanent in nervous activity," *Bulletin of Mathematical Biophysics*, **5**, 1943, 115–133.

78. Mead, C. A., Mahowald, M. A., "A silicon model of early visual processing," *Neural Networks*, **1**, 1988, 91–97.

79. Mendelson, J. R., Cynader, M. S., "Sensitivity of cat primary auditory cortex (AI) neurons to the direction and rate of frequency modulation," *Brain Research*, **327**, 1985, 331–335.

80. Merrill, J. W. L., and Port, R. F., "A new stochastic algorithm for neural networks," TR No. 236, Indiana University Department of Computer Science, Bloomington, 1987.

81. Minsky, M., *The Society of Mind*, Simon and Schuster, 1986.

82. Minsky, M. L., "Steps toward Artificial Intelligence," in: Feigenbaum, E. A., and Feldman, J., eds. *Computers and Thought*, McGraw Hill, NY, 1963, 406–450

83. Minsky, M., Papert, S., *Perceptrons*, MIT Press, 1969.

84. Møller, A. R., "Neurophysiological basis for perception of complex sounds," In: Carlson, R., and Grandström, B., *The Representation of Speech in the Peripheral Auditory System*, Elsevier, 1982, 43–60.

85. Møller, A. R., "Coding of complex sounds in the auditory nervous system," in: Creutzfeldt, O., Scheich, H., Schreiner, Chr., Eds, "Hearing Mechanisms and Speech," *Experimental Brain Research*, Supp. 2. Springer Verlagg, 1979.

86. Moore, B. C. J., Glasberg, B. R., "The role of frequency selectivity in the perception of pitch, loudness, and timbre," in: Moore, B. C. J. (ed), *Frequency Selectivity in Hearing*, Academic Press, 1986, 251–308.

87. Nabelek, I., Hirsch, I. J., "On the discrimination of frequency transitions," *Journal of the Acoustical Soceity of America*, **45**, 1969, 1510–1519.

88. Noordmark, J. O., "Time and frequency analysis," in: Tobias, J. V., ed., *Foundations of Modern Auditory Theory*, Vol. I, Academic Press, 1970.

89. O'Neill, W. E., Suga, N., "Encoding of target range and its representation in the auditory cortex of the mustached bat," *Journal of Neuroscience*, **2**, 17–31.

90. Parker, D. B., "Learning Logic," Technical Report TR–47 Massachusetts Institute of Technology, 1985.

91. Peters, A., Jones, E. G., *Cerebral Cortex*, Vol. 4, Plenum, 1985.

92. Pollack, I., "Detection of rate of change of auditory frequency," *Journal of Experimental Psychology: Human Perception and Performance*, **77**, 1968, 535–541.

93. Pollack, J. P., "Cascaded back–propagation on dynamic connectionist networks," *Proceedings of the 9th Annual Conference of the Cognitive Science society*, 1987.

94. Pols L. C. W., and Schouten, M. F. H., "Perceptual relevance of coarticulation," in: Carlson, R., and Grandström, B., *The Representation of Speech in the Peripheral Auditory System*, Elsevier, 1982, 203–208.

95. Rescorla, R. A., Wagner, A. R., "A theory of Pavlovian conditioning: Variations in the effectiveness of reinforcement and nonreinforcement," in: Black, A. H., and Proskay, W. F., eds. *Classical Conditioning II*, Apelton–Century–Crofts, NY, 1972, 64–99.

96. Ruff, P. I., Rauschecker, J. P., and Palm, G., "A model of direction–selective "simple" cells in the visual cortex based on inhibition asymmetry," *Biological Cybernetics*, **57**, 1987, 147–157.

97. Rumelhart, D. E., Hinton, G. E., Williams, R. J., "Learning internal representations by error propagation," In: Rumelhart, D. E., McClelland, J. L., *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, Vol. 1: Foundations, MIT Press. 1986.

98. Rumelhart, D. E., McClelland, J. L., *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, Vol. 1: Foundations, MIT Press. 1986,

99. Rumelhart, D. E., McClelland, J. L., "An interactive activation model of context effects in letter perception: Part 2: The contextual enhancement effect and some tests and extensions of the model," *Psychological Review*, **89**, 1982, 60–94.

100. Scharf, B., "Critical Bands," in: Tobias, J. V., ed., *Foundations of Modern Auditory Theory*, Vol. I, Academic Press, 1970, 157–202.

101. Sergeant, R. L., Harris, J. D., "Sensitivity to unidirectional frequency modulation," *Journal of the Acoustical Soceity of America*, **34**, 1962, 1625–1628.

102. Shamma, S. A., "Neural networks for speech processing and recognition," *Proceedings of the IEEE International Conference on Neural Networks*, San Diego, 1987, IV–397–406.

103. Shepard, G. M., Brayton, R. K., Miller, J. P., Segev, I., Rinzel, J., and Rall, W., "Signal enhancement in distal cortical dendrites by means of interaction between active dendritic spines," *Proceedings of the National Academy of Sciences: USA*, **82**, 1985 2192–2195.

104. Sinex, D. G., Geisler, C. D., "Responses of auditory nerve fibers to consonant–vowel syllables," *Journal of the Acoustical Soceity of America*, **73**, 1983, 602–615.

105. Sinex, D. G., and Geisler, C. D., "Auditory nerve fiber responses to frequency–modulated tones," *Hearing Research*, **4**, 1981, 127–148.

106. Small, A. M., "Periodicity pitch," in: Tobias, J. V., ed., *Foundations of Modern Auditory Theory*, Vol. I, Academic Press, 1970, 1–54.

107. Smolensky, P., "Information processing in dynamical systems: Foundations of harmony theory," In: Rumelhart, D. E., McClelland, J. L., eds. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, Vol. 1: Foundations, MIT Press. 1986.

108. Smythe, E. J., "The detection of formant transitions by a connectionist network," *Proceedings of the IEEE International Conference on Neural Networks*, San Diego, 1987, IV-495–503.

109. Smythe, E. J., "The source code of SYREN: A connectionist syllable recognition network," TR-252, Computer Science Department, Indiana University, Bloomington, Indiana, 1988

110. Stevens, K. N., and Blumstein, S. E., "Invariant cues for place of articulation in stop consonants," *Journal of the Acoustical Society of America*, **64**, 1978, 1358–1368.

111. Stevens, K. N., Klatt, D. H., "Role of formant transitions in voiced–voiceless distinction for stops," *Journal of the Acoustical Society of America*, **55**, 1974, 653–659.

112. Stillito, A. M., "Inhibitory processes underlying the directional specificity of simple, complex, and hypercomplex cells in the cat's visual cortex," *Journal of Physiology*, **271**, 1977, 699–720.

113. Strange, W., Edman, T. R., Jenkins, J. J., "Acoustic and phonological features in vowel identification," *Journal of Experimental Pychology: Human Perception and Performance*, **5**, 1979, 643–656.

114. Strange, W., Jenkins, J. J., Johnson, T. L., "Dynamic specification of coarticulated vowels," *Journal of the Acoustical Society of America*, **74**, 1983, 695–705.

115. Strange, W., Verbrugge, R. R., Shankweiller, D. P., Edman, T. R., "Consonant environment specifies vowel identity," *Journal of the Acoustical Society of America*, **60**, 1976, 213–224.

116. Strehler, B. L., Lestienne, R., "Evidence on precise time–coded symbols and memory of patterns in monkey cortical neuronal spike trains," *Proceedings of the National Academy of Sciences: USA*, **83** 1986, 9812–9816.

117. Suomi, K., "The vowel–dependence of gross spectral cues to place of articulation of stop consonants in CV syllables," *Journal of Phonetics*, **13**, 1985, 267–285.

118. Sutton, R. S., "Learning to predict by methods of temporal differences," GTE TR87–509.1, 1987.

119. Sutton, R. S., Barto, A. G., "A temporal difference model of classical conditioning," GTE TR87–509.2 1987.

120. Sutton, R. S., Barto, A. G., "Toward a modern theory of adaptive networks: Expectation and prediction," *Psychological Review*, **88**, 1981, 135–170.

121. Tallal, P., Stark, R. E., "Speech acoustic–cue discrimination abilities of normally developed and language impaired children," *Journal of the Acoustical Soceity of America*, **69**, 1981, 568–574.

122. Tank, D. W., Hopfield, J. J., "Concentrating information in time: Analog neural networks with applications to speech recognition problems," *Proceedings of the IEEE International Conference on Neural Networks*, San Diego, 1987, IV-455–468.

123. Tank, D. W., and Hopfield, J. J., "Neural computation by concentrating information in time," *Proceedings of the National Academy of Sciences: USA* **84**, 1987, 1896–1900.

124. Travis, B. J., "A layered neural network model applied to the auditory system," In: Denker, J. S., ed. *Neural Networks for Computing*, AIP conference Proceedings No. 151, 1986, 432–439.

125. Torre, V., Poggio, T., "A synaptic mechanism possibly underlying directional selectivity to motion," *Proceedings of the Royal Society of London: Series B*, **202**, 1978, 409–416.

126. Watrous, R. L., Shastri, L., "Learning phonetic features using a connectionist network, an experiment in speech recognition," *Proceedings of the IEEE International Conference on Neural Networks*, San Diego, 1987, IV-381–388.

127. Whitfield, I. C., Evans, E. F., "Responses of auditory cortical neurons to stimuli of changing frequency," *Journal of Neurophysiology*, **28**, 1965, 655–672.

128. Widrow, B., Hoff, M. E., "Adaptive switching circuits" *IRE WESCON Convention Record*, pt. 4, 1960, 96–104.

129. Willwacher, G., "Storage of temporal pattern sequences in a network," *Biological Cybernetics* **43**, 1982, 115–126.

130. Williams, R. J., "Reinforcement–learning connectionist systems," Technical Report NU–CCS–87–3, Northeastern University, Boston, 1987.

131. Wilson, H. R., "A model for direction selectivity in threshold motion perception," *Biological Cybernetics*, **51**, 1985, 213–222.

132. Wyatt, H. J., and Daw, N. W., "Directionally sensitive ganglion cells in the rabbit retina: Specificity for stimulus direction, size, and speed," *Journal of Neurophysiology*, **38**, 1975, 613–626.

133. Zeki, S. M., "Uniformity and diversity of structure and function in rhesus monkey prestriate visual cortex," *Journal of Physiology*, **277**, 1978, 273–290.

Vita

Erich Smythe received his B.S. in Biology from Indiana University, Bloomington, in May 1982. Not wanting to leave Bloomington just yet, he received his M.S. in Computer Science from Indiana in May, 1986. He received his Ph.D. in Computer Science from Indiana in June, 1988.

During his graduate education, Erich was a Lecturer in Computer Science during the 1987–88 academic year. From 1984–87 he was an associate instructor in computer science, except for the 1985–86 academic year when he was a research assistant for a Spencer Foundation grant.