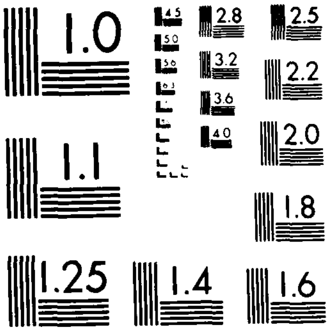
AD-A174 935 APPLICATIONS OF NEURAL NETWORK MODELS IN AUTOMATIC SPEECH RECOGNITION(U) BATTELLE COLUMBUS LABS OH A S NOETZEL ET AL 29 SEP 86 DAAG29-81-D-0100 1/1

UNCLASSIFIED F/G 9/2 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963-A

1

APPLICATIONS OF NEURAL NETWORK MODELS IN AUTOMATIC SPEECH RECOGNITION

by

Andrew Noetzel
Department of EE/CS
Polytechnic University
Brooklyn, New York

and

Thomas Rittenbach
U.S. Army
Communications-Electronics Command
Ft. Monmouth, New Jersey

for

U. S. Army Communications-Electronics Command
ATTN: AMSEL-COM-TN-1
Ft. Monmouth, N. J. 00703-5202

September 29, 1986

Contract No. DAAG29-81-D-0100
Delivery Order 2172
U. S. Army Summer Faculty Research and Engineering Program
Scientific Services Program

The views, opinions, and/or findings contained in this report are those of the authors and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other documentation.

SECURITY CLASSIFICATION OF THIS PAGE

ADA 174935

REPORT DOCUMENTATION PAGE

Form Approved
OMB No 0704-0188
Exp Date Jun 30 1986

1a REPORT SECURITY CLASSIFICATION
Unclassified

1b RESTRICTIVE MARKINGS

2a SECURITY CLASSIFICATION AUTHORITY

2b DECLASSIFICATION/DOWNGRADING SCHEDULE

3 DISTRIBUTION/AVAILABILITY OF REPORT
Approved for public release;
Distribution Unlimited
~~an sponsori~~
f US Army

4 PERFORMING ORGANIZATION REPORT NUMBER(S)
Delivery Order 2172

5. MONITORING ORGANIZATION REPORT NUMBER(S)
TCN 86219

6a. NAME OF PERFORMING ORGANIZATION
Dr. Andrew S. Noetzel

6b. OFFICE SYMBOL *(If applicable)*

7a. NAME OF MONITORING ORGANIZATION
U.S. Army Research Office

6c. ADDRESS (City, State, and ZIP Code)
215 W. 75th St.
New York, N.Y. 10023

7b. ADDRESS (City, State, and ZIP Code)
P.O. Box 12211
Research Triangle Park, NC 27709-2211

8a. NAME OF FUNDING/SPONSORING ORGANIZATION
Mr. Thomas Rittenbach

8b. OFFICE SYMBOL *(If applicable)*
AMSEL-RD-COM-TR-1

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER

8c. ADDRESS (City, State, and ZIP Code)
US Army CECOM
Ft. Monmouth, NJ 07703-5202
(201) 544-5170

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO

11. TITLE *(Include Security Classification)*
Applications of Neural Network Models in Automatic Speech Recognition (Unclassified)

12. PERSONAL AUTHOR(S)
Andrew Noetzel and Thomas Rittenbach

13a. TYPE OF REPORT
FINAL REPORT

13b. TIME COVERED
FROM 6/86 TO 9/86

14. DATE OF REPORT *(Year, Month, Day)*
1986-9-29

15. PAGE COUNT

16. SUPPLEMENTARY NOTATION Task was performed under a Scientific Services Agreement issued by Battelle, Research Triangle Park Office, 200 Park Drive, P.O. Box 12297, Research Triangle Park, NC 27709

17. COSATI CODES

FIELD	GROUP	SUB-GROUP

18. SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*

19. ABSTRACT *(Continue on reverse if necessary and identify by block number)*

Organizations of computing elements that follow the principles of physiological neurons, called neural network models, have been shown to have the capability of learning to recognize patterns and to retrieve complete patterns from partial representations. The implementation of neural network models as VLSI or ULSI chips within a few years is certain. This report reviews a number of published papers on neural network models and their capabilities. Then, an outline of a speech recognition system that uses neural network modules for learning and recognition is proposed. It is based on the layered structure of existing speech recognition systems, and uses forced learning (feedback) for conditioning the neural modules at the various levels.

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT
□ UNCLASSIFIED/UNLIMITED □ SAME AS RPT □ DTIC USERS

21 ABSTRACT SECURITY CLASSIFICATION

22a NAME OF RESPONSIBLE INDIVIDUAL

22b TELEPHONE *(Include Area Code)*

22c OFFICE SYMBOL

83 APR edition may be used until exhausted
All other editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE

APPLICATIONS OF NEURAL NETWORK MODELS IN AUTOMATIC SPEECH RECOGNITION

1. INTRODUCTION

There has long been an interest among biologists, neurophysiologists, and experimental psychologists, in the manner in which the operation of an individual neuron can be made to account for the behavior of the entire nervous system or brain. There has been a parallel and overlapping interest among computer scientists in demonstrating how complex behavior can emerge from primitive computing elements. These interests, taken in the aggregate, constitute an area variously known as self-organizing networks, neural networks, learning networks, or associative memories.

Most of the study in this area has been theoretical. Much has been done by computer simulation. It has long been accepted that in order to demonstrate significant "intelligent" behavior, or behavior resulting from a high level of organization, one would need the parallel operation of millions, if not billions, of these computing elements. Until recently, this was a practical impossibility. But great advances in VLSI technology have been made in the past few years, and the current outlook promises an ultra large-scale integration (ULSI), with even larger chip sizes and greater circuit densities. Some researchers have therefore turned to the practical considerations of implementing learning networks.

This has brought new life to the study of neural networks, particularly from the point of view of Computer Science. Here, the motivation for the study is to determine principles applicable to the construction of computers and computing modules for particular applications, rather than to build faithful models of biological organisms. In fact, computer scientists are to a certain degree being driven to the neural networks by the prospect of ULSI, since it is becoming apparent that the traditional Von Neuman architecture and its multiprocessor generalizations cannot make efficient use of ULSI circuitry.

In the second section of this report, we review some of the works that have been published on self-organizing networks. We summarize the various definitions of learning, review the research results, and outline characteristics that distinguish the approaches to neural network design. Of those studies that were not intended to be precise models of biological systems, some are abstract models of pattern recognition, and some are oriented towards the problems of image recognition. But it is clear that the principles of neural network design can be applied to speech recognition as well, as long as the neural networks are deployed in such a way as to recognize what is known within the complex structure of speech.

Thus, in the third section, we propose and discuss the design of a speech recognition system based on neural networks. We offer some arguments in support of the notion that recognition of speech is a task well suited to neural networks, and that this application will benefit from the earliest practical implementations of the technique.

2. REVIEW OF THE LITERATURE

In this section, we review some of the reports in the literature that describe networks of elements that in some way represent neurons. The elements will be called either neurons, cells or units, depending on how closely their properties are intended to represent those of biological neurons.

The work of Hopfield [1] is useful as a introduction and overview of many neural networks, since it is purposefully abstract. In his model, a cell has only the most elementary properties considered fundamental to computation capability. In particular, the state of cell i is represented only by its output or firing rate V_i.

In biological nervous systems, the average rate at which a cell fires is a nonlinear (and apparently somewhat stochastic) function of its inputs. A model in which V_i can take on continuous values is called a graded-response model [2]. However, in Hopfield's and other elementary models, V_i takes on only the values zero and one.

The state s of the network is a vector of the states of the cells. For a network of n cells, $s=(V_1,V_2,...,V_n)$.

The purpose of most of the reported neural networks is to recognize patterns from the environment. In one class of network, the recognition of a pattern is indicated by the network entering a specific state, or a repeating sequence of closely related states. Thus, let S represent a set of designated states, or (in Hopfield's terms) memories. The network will operate as an associative memory if it can be initialized with some of the cells set to the values of a particular memory s ϵ S, and the rest set randomly, and it will then move toward and enter the memory s. One might then say it retrieves complete information from partial information, or recognizes a complete pattern from a fragmented pattern.

A biological neuron communicates with another through a *synapse*. The strength of the connection from neuron j to neuron i is a measure of the effectiveness of the synapse in communicating V_j to influence V_i. In more abstract terms, the strength of the connection from cell j to cell i is described by the *weighting factor* T_{ij}.

In Hopfield's model, a cell fires if the weighted sum of its inputs is greater than some threshold U_i:

$$\text{If} \quad T_{ij}V_j \begin{cases} > U_i, & V_i=1 \\ < U_i, & V_i=0 \end{cases} \qquad (1)$$

U_i is a threshold value, which could be varied as an elaboration of the model, but is usually zero.

A *plastic* synapse is one whose strength is modified through experience. Modification of synapse strength or connection weight is the central mechanism for learning in neural networks. The *learning rule* describes how the weights T_{ij} are modified. According to the neurological model proposed by Hebb [3], the strength of a synapse is increased only as a result of the simultaneous firing of the pre- and post-synaptic neurons. Hence a learning rule which modifies T_{ij} only as a consequence of simultaneous activation of cells i and j will be termed a *Hebbian* rule.

Hopfield's model does not describe how learning takes place. It assumes that there is a set S of distinguished network states, each of which represents recognition of a pattern, and that the weights have been established on the basis of these states.

$$T_{ij} = \sum (2V_i - 1)(2V_j - 1) \qquad (2)$$

Where the summation is over all distinguished states $s \in S$, and $T_{ii}=0$.

By (2), it is seen that if a distinguished state has one cell active and the other not, its contribution to the connection weight between the cells is negative. If the state has either both cells active or both inactive, its contribution is positive. This rule is consistent with the Hebbian learning rule, in that the connection strength is correlated with simultaneous activity of the cells. But here, the connection strength is specified by the intended cell activity, rather than caused by that activity.

Hopfield considers the case in which the weights are symmetric ($T_{ij}=T_{ji}$) for his analysis. He defines the energy E of the system by

$$E = -0.5 \sum_{i \neq j} [T_{ij} V_i V_j] \qquad (3)$$

The change in overall system energy resulting from the change of the state of cell i is

$$dE = -dV_i \sum_{i \neq j} [T_{ij} V_j] \qquad (4)$$

It can be shown that with symmetric weights, and cell changes given by (1), the energy of the system decreases with every cell state change.

The main result is an estimate of the relationship between the number of designated states N that can reasonably be entered into a network of n cells. The equation (2) places no limit on the number of designated states; however, if N is too large with respect to n, the network may wander in state space and never come to rest at a final state, or else come to rest at a state that has no relation to the initial state.

A simple error measurement is obtained by starting the network in a state $s \in S$, and observing how often it will come to rest at s or a state close to it. The result is that if N exceeds .15n, the error rate exceeds 50 percent. That relationship seemed to hold independently of n. A smaller, *but not negligible error rate could be obtained* with N=.1n, or about ten network cells per designated state.

Networks of the kind defined by Hopfield can be implemented in silicon. The weights on the links are not determined by the operation of the network: they could be established at the time the chip is created, as a ROM. While many circuits may be required, the network will be insensitive to the failure of a few, hence be amenable to ULSI or wafer-scale integration.

Silverman, Shaw and Pearson [4] describe another model in which the network as a whole enters a particular state in response to an environmental stimulus. In their model, each element, called a "trion", is intended to be an abstraction of a cluster of about a hundred neurons. Each trion may assume one of three possible states: +1,0 or -1, respectively corresponding to high, average and low output firing rates. Let g(S) be the initial probability that a particular trion is in state S. The initial firing rates are assumed to be such that g(0) >> g(-1),g(1).

The state of the system at time n is related to the system states at times n-1 and n-2. Let S_j, S'_j, and S''_j be the states of the j^{th} trion at times n, n-1, and n-2 respectively. Then $P_i(S)$, the probability of the i^{th} trion being in state S at time n is given by

$$P_i = g(S)\exp[B\cdot M_i\cdot S]/(\sum_{s} g(s)\ \exp[B\cdot M_i\cdot s]\ ) \qquad (5)$$

where

$$M_i = \sum_j [\ V_{ij}S'_j + W_{ij}S''_j\ ] - V_i^t\ ,$$

V_i^t is a threshold, and B is inversely proportional to noise. Each trion is influenced by the states of its neighbors at the previous two time steps. Hence, V_{ij} and W_{ij} are the interaction weights between trion i (at time n) and trion j at times n-1 and n-2 respectively. The weights may be positive or negative, corresponding to excitatory or inhibitory interaction.

The network used in the simulations consisted of six trions arranged in a circle. Each trion interacted with itself and and its two nearest neighbors on the left and right. The V and W weights were assigned as +1 or -1.

Networks based on various assignments for the weights were usually found to fall into one of a few stable firing patterns, consisting of repeating trion firing sequences of various lengths.

The parameter B may be thought of as thermal noise or random variations in cell output. As B decreases, the noise level increases, resulting in occasional random errors. This may cause the firing sequence to change to a nearby pattern, which could be interpreted as a form of associated recall.

The firing patterns that develop in a network with assigned values for the weights may be enhanced by Hebbian learning rules of the form

$$dV = e\sum[S_i(n)S_j(n-1)] \qquad (6)$$

$$dW = e\sum[S_i(n)S_j(n-2)] \qquad (7)$$

where e>0.

The network might be considered "naive" before the Hebbian learning rules are applied, in the sense that the only connections between cells are those that are prewired, or genetically determined. Once the learning rules take effect, the network learns from experience to choose firing patterns appropriate to a given input pattern.

In the model described by Ackley, Hinton and Sejnowski [6], the network output consists of the state of a relatively small number of higher level cells (which the authors call units) in a hierarchical architecture. The novelty of this model is found in the use of thermal noise as an essential element in the search for optimal global solutions.

Cell i has output S_i, which is either zero or one. The weight on the connection between cell i and cell j is W_{ij}. The overall energy E of the system is defined by

$$E = -\sum_{i<j}[W_{ij}S_iS_j] + \sum_i[QS_i] \; , \qquad (8)$$

where Q is a threshold.

Let dE_k be the energy difference between a state with cell k on and cell k off:

$$dE_k = -\sum_i[W_{ki}S_i] - Q_k \qquad (9)$$

Global energy is minimized by increasing the weights for co-active cells, a Hebbian rule. If a single cell is assumed to be constantly on, the threshold term can be included in the interactions between cells, simplifying the above equations:

$$E = -\sum_{i<j}[W_{ij}S_iS_j] \qquad (10)$$

$$dE_k = - \sum_i [W_{ki}S_i] \qquad (11)$$

Noise is utilized to help the system escape from local energy minima. The S_k values are not fully determined by the system state and weights. Instead, there is a probabilistic decision. Let P_k be the probability that S_k will be set to one. The cell state transition is described by

$$P_k = 1/(1 + \exp[-dE_k/T]). \qquad (12)$$

As dE gets larger, P_k approaches 1. As T gets larger, P_k approaches zero. Therefore at low T, a relatively small energy gap can cause a change in sta.e, and at high T, a large dE_k is needed to set P_k to one.

At equilibrium,

$$P_a/P_b = \exp[-(E_a-E_b)/T] , \qquad (13)$$

where P_a and P_b are the probabilities of the a and b global states.

Low T favors states with low energy, but the rate at which the optimal state is approached is slow. High T favors low energy less strongly, but the time to reach equilibrium is reduced.

The information gain of the system is defined by

$$G = \sum_a [P(V_a) \ln[P(V_a)/P'(V_a)]]$$

where $P(V_a)$ represents the probability of the network being in state a with the network "clamped" (the states of some of the cells are fixed by the input pattern), and $P'(V_a)$ is the same probability when the network is free running. For example, if for all a, $P(V_a)=P'(V_a)$, the information gain G is zero.

Gradient descent is performed by modifying link weights:

$$dG/DW_{ij} = -(1/T)(p_{ij} - p'_{ij}) \qquad (15)$$

where p_{ij} and p'_{ij} are the probabilities that both cells i and j are active when the network is clamped and free-running, respectively, in thermal equilibrium. Therefore, the network adapts to stimuli by changing link weights in proportion to p_{ij} and p'_{ij} respectively.

$$dW_{ij} = e\cdot(p_{ij} - p'_{ij})$$

where e is a constant.

By the equations above, if a pair of cells have a higher probability of being concurrently active when the network is clamped than when free running, weight will be added to the link between them. This will result in a decrease in system energy.

The weight increment dW_{ij} is determined entirely on the basis of locally available information, while affecting the global energy level. Minimizing G is the process of the network capturing regularities in input patterns.

The authors provided results of simulations involving two layered networks of various sizes. When seeking equilibrium, the system was run at a series of decreasing temperatures, so that the global energy minimum was approached in a reasonably short period of time, while still avoiding capture at local energy minima. The concept of slow reduction of temperature in order to achieve an optimum result is known as annealing. Its use in optimization algorithms in general is discussed in [7].

Because the probability distribution used to determine state changes was the Boltzmann distribution, parameterized by temperature, the network was called the Boltzmann machine. It was run to solve a series of binary encoding problems. As the complexity of the problems increased, the number of learning cycles required to solve the problem also increased, and in some cases the best solution was not found.

Rummelhart and Zipser [7] also describe a multilayered hierarchical architecture with Hebbian learning rules. In their model, the inputs to a cell at one layer come from the cells at the layer below. The weights associated with the inputs to a particular cell sum to one.

The network includes the powerful feature of lateral inhibition, in which an active cell can inhibit the cells at the same level. In the model of [7], cells are arranged in clusters. All cells of a cluster receive the same input signals, but each cell weights them differently. The cells output either zero or one. The cell with the largest weighted input in the cluster outputs one; all others have zero output. The mechanism of the cluster may be thought of as a competition, in which the cell that wins the right to output also inhibits the other cells of the cluster. The cells at the lowest level of the hierarchy represent the input pattern or stimulus.

The learning rule for this model is the following. The weight on the input to cell j from cell i on the lower level is W_{ij}. If pattern S_k is presented to the network, c_{ik} will be the output of cell i (on the lower level). For each pattern S_k, the weights on the inputs to a cell are modified only if the cell wins the competition within its cluster. That is, with pattern S_k,

$$dW_{ij} = 0 \quad \text{if cell j loses,}$$

and

$$dW_{ij} = g(c_{ik}/n_k) - g\,W_{ij} \quad \text{if cell j wins.} \qquad (17)$$

Here, n_k is the number of active cells in pattern S_k;

$$n_k = \sum_i c_{ik} \ .$$

Let V_{jk} be the probability of cell j winning on presentation of stimulus S_k, and let P_k be the probability of S_k being presented on a given trial. At equilibrium,

$$\sum_k dW_{ij} V_{jk} P_k = 0 \ . \qquad (18)$$

As in the works described above, the authors define a global energy term. In this case, the parameter T quantifies system stability, and hence is the negative of energy.

$$T = \sum_k P_k \sum_{j,i} [V_{jk}(a_{jk} - a_{ik})]] \qquad (19)$$

where,

$$a_{jk} = \sum_i [W_{ij} c_{ik}] \ .$$

T is the amount by which the weighted input to winning cells exceeds the weighted input to all other cells, averaged over all stimuli. Since T is the negative of energy, it must be maximized.

A weakness of the learning rule based on competition is that if some cells original (naive) are not related to any stimulus, it may never win the competition, hence never learn. A modification of the learning rule permits "leaky learning" in which all link weights are modified;

$$dW_{ij} = g_l c_{ik} - g_l W_{ij} \quad \text{if cell j loses on } S_k$$

and

$$dW_{ij} = g_w c_{ik} - g_w W_{ij} \quad \text{if cell j wins on } S_k \quad (20)$$

where $g_w >> g_l$. With this rule, cells that constantly lose the competition move slowly towards the active patterns, so as to eventually win.

The authors describe simulations of networks with varying numbers of input cells on the lower level, and one cluster of two units on the upper level. A few letter recognition experiments worked predictably well. Difficulties were exposed in the attempts to train the upper level cells to distinguish between vertical and horizontal lines. Each of the pattern sets that should be recognized as similar (the set of vertical lines and the set of horizontal lines) consist of disjoint elements: parallel lines do not intersect. But by the pattern recognition scheme, patterns are recognized as similar by the number of points they have in common. Therefore, because of the single point they have in common, a vertical line and a horizontal line are considered more alike than any pair of horizontal or vertical lines. The result of exciting the network with a series of vertical lines and horizontal lines is that the upper levels systematically discard exactly what they are intended to capture.

The authors demonstrate that their model can be trained to capture the idea of vertical vs. horizontal in the following way. The higher level clusters are increased from two to four cells each, and a third level, with a single cluster of two units is added. Then each time a vertical line was presenteo on tne right side of the input matrix, it was accompanied by a vertical line in the leftmost column. Similarly, horizontal lines were accompanied by a horizontal line in the uppermost row. The vertical "training patterns" had more points in common with each other than with horizontal training patterns, and so the level two units easily learned to distinguish between them. With four cells on the second level, two would develop weights that would allow the recognition of vertical training patterns: they divided this set between themselves. Two cells would similarly recognize horizontal training patterns. When the training lines were removed from the input patterns, the remaining parts of the patterns were recognized in the same way. And since there are two second-level clusters, the third level would always have two out of eight inputs active, from which it would easily distinguish the vertical from the horizontal patterns.

In the work of Grossberg [8], an "on center/off surround" architecture is described. Each cell receives, in addition to its own (center) input, weighted negative inputs which are the center inputs to neighboring cells. The cells are organized in layers; layer one or v_1 consists of cells $V_{11}, V_{12}, \ldots, V_{1n}$. The output of cell V_{1i} is the continuous variable x_{1i}. (This is a graded-response model.) At level one, the cells are excited by an external pattern. I_i is the part of the pattern presented to V_{1i} to compute x_{1i}.

The equation governing output by a given neuron is

$$dx_{1i} = -Ax_{1i} + (B-x_{1i})I_i - x_{1i} \cdot \sum_{k \neq i} I_k \qquad (21)$$

where $B > x_{1i}$.

The first term in (21) specifies exponential decay of the output, based on the constant A. This permits gradual loss of memory.

The second term is the "on center" part of the expression. If I_i is large, and $x_{1i} \ll B$, then dx_{1i} is strongly positive, and x_{1i} becomes larger.

The third term is the "off-surround". It decreases the output of V_{1i} in proportion to the sum of the input pattern surrounding that part of the pattern presented directly to V_{1i}.

Let $I = \sum_k I_k$ and let $Q_i = I_i I^{-1}$. At equilibrium,

$$x_{1i} = Q_i \cdot B \cdot I/(A+I) \qquad (22)$$

The internal network outputs are adjusted to be relative to the intensity of the input pattern. This mechanism works like an automatic gain control.

To store patterns after input ceases, a reverberating architecture is proposed. Cells on level two excite and inhibit level three cells by the on-center/off-surround technique. Level three cells in turn excite the level two cells in the same way. This may be described by the equation

$$dx_{2j} = -Ax_{2j}+(B-x_{2j})[f(x_{2j})+I_{2j}]-x_{2j}\sum_{k\neq j} f(x_{2k}), \qquad (23)$$

where f(w) is a feedback signal produced by average activity w. In (23), the on-center term is related to both the original input pattern and the feedback signal from level three. The off-surround term is from units on the third level. Since these units are excited by neighbors of V_{2j}, the cell V_{2j} is indirectly inhibited by its neighbors.

Let Z_{ij} be the link weight from V_{1i} to V_{2j}, and D_{ij} be the signal from V_{1i} to V_{2j}. The learning rule is

$$dZ_{ij} = -c_{ij} + D_{ij}x_{2j}, \qquad (24)$$

where c_{ij} is the decay rate of Z_{ij}. The second term of the above expression indicates that the learning rule is Hebbian; the weights are increased whenever the pre- and post-synaptic cells are active.

The model offers two possible scenarios for pattern capture; choice, and partial contrast.

choice:

$$x_{ij} = \begin{cases} 1 & \text{if } S_j > \max\{e, S_k : k<>j\} \\ 0 & \text{if } S_j < \max\{e, S_k : k<>j\} \end{cases}$$

contrast:

$$x_{ij} = \begin{cases} f(S_j)\sum_{S_k>e}[f(S_k)]^{-1} & \text{if } S_j > e \\ 0 & \text{if } S_j < e. \end{cases}$$

In the choice mode, only the cell with the strongest excitation (above threshold e) becomes active. In partial contrast, all cells above threshold respond in proportion to their relative input levels above threshold.

There is a noteworthy degree of underlying similarity between Grossberg's model the Trion model of Silverman, et. al. In Grossberg's model, firing patterns reverberate (and are enhanced) between levels two and three of the network. In the Trion model, patterns reverberate temporally and spatially; each trion may be subject to excitation or inhibition from itself and its neighbors (all on a single level) over the past two time periods. It does not seem unreasonable to suppose that the temporal aspect of the Trion model could in fact just as easily be represented with a reverberating spatial pattern.

Fukushima [9] describes a multilayered network with a modified Hebbian learning rule and a partially stochastic interlayer connection pattern. The learning rule is the following: Connections frcm cell x to cell y are reinforced i: x fires and y is firing more strongly than any other post-synaptic cell in its neighborhood. Among cells in the vicinity of a particular cell, only one is reinforced at a given time.

This learning rule bears a strong similarity to the "competitive learning" mechanism of Rummelhart and Zipser [7], in that only only one cell in a particular grouping is reinforced. But it is not, strictly speaking a lateral inhibition mechanism, since the neighboring cells are not prevented from firing. After learning has taken place, only the one cell responding to a particular pattern will fire.

Suppose u(i) and v(i) are the i^{th} excitatory and inhibitory inputs, respectively, to a particular cell. Let a(i) and b(i) be the respective conductances (weights) of these inputs. The rule for determining the output W of a cell is described as follows.

$$W = f[(1+\sum_j a(j)u(j))/(1+\sum_j b(j)v(j)) - 1] \qquad (25)$$

where $f[x] = 0$ for $x<0$ and $f[x]=x$ for $x \geqslant 0$.

Suppose e and h respectively represent the total excitatory and inhibitory inputs to a cell:

$$e = \sum_j a(j)v(j) \quad \text{and} \quad h = \sum_j b(j)u(j) .$$

Then (25) can be rewritten as

$$W = f[(1+e)/(1+h) -1] = f[(e-h)/(1+h)] \qquad (26)$$

Then the gain control mechanism for the cell output can be seen. If $h \ll 1$, W is approximately f[e-h]. If $e \gg 1$ and $h \gg 1$, W is approximately f[(e/h)-1]. This last condition is the state of the network after learning has occurred. The weights a and b increase indefinitely. For this reason, thresholds do not make sense for this model.

Each cell at level k receives excitatory inputs from cells in a particular area on the next lower level (its "connectable area"), and one inhibitory input from the same area. The index j will span the connectable area. Then $u_{k-1}(n+j)$ will be an excitatory input to cell n at level k, from cell n+j at level k-1. The inhibitory cell on the lower level receives input from the same cells as excite the upper level cell. Thus, the inhibition of cell n at level k is the sum of the excitatory inputs to that cell, multiplied by the (unmodifiable) weights from excitatory to inhibitory cells within level k-1.

$$v_k(n) = \sum_j c_{k-1}(j)\; u_{k-1}(n+j) \qquad (27)$$

The excitatory input to cell n at level k, $u_k(n)$, is greater than zero if the sum of the excitatory inputs from the connectable area on level k-1 is greater than the inhibitory input.

$$u_k = f[(1+\sum_j a_k(j,n)u_{k-1}(n+j)\;)/(1+b_k(n)\; v_{k-1}(n)) -1] \quad (28)$$

The equations for modification of excitatory and inhibitory connection weights are such that if no cell in the neighborhood of n is firing, all cells are (weakly) reinforced. If one cell is firing more strongly than the others, it is the only one to be reinforced.

Let $g_k(n)$ be a function that takes on the valte one if no cell in its vicinity is firing more strongly (i.e., $u_k(n)>=u_k(n+j)$, and value zero otherwise. If $u_k(n)=0$ then

$$da_k(j,n) = q_0 c_{k-1}(j)u_{k-1}(n+j)\cdot g_k(n) \qquad (29)$$

$$db_k(n) = q_0 \cdot v_{k-1}(n)\cdot g_k(n). \qquad (30)$$

If $u_k(n) > 0$ then

$$da_k(j,n) = q_1 \cdot c_{k-1}(j)\cdot u_{k-1}(j+n)\cdot g_k(n) \qquad (31)$$

$$db_k(n) =[(\sum_j a_k(j,n)u_{k-1}(j+n))/(2\ v_{k-1}(n))]\cdot g_k(n), (32)$$

Excitatory reinforcement from a cell on level k-1 to a cell on level k is a constant times the output level of the level k-1 cell, provided that the total excitation to the level k cell is is higher than to any other cell in its vicinity. If the level k cell is receiving subthreshold excitation, the reinforcement is weaker than if it had suprathreshold excitation.

The equations for reinforcement of inhibitory connections, similarly, only apply for the most strongly excited level k cell in its vicinity. For the subthreshold case, reinforcement is a constant times the output of the level k-1 inhibitory cell. For the suprathreshold case, the reinforcement equation is more complex, but has the following implications.

Let $r(j,n)$ be the ratio of excitatory to inhibitory reinforcement;

$$r(j,n) = da(j,n)/(c_{k-1}(j)\ db(n)).$$

It can then be shown that

$$r>1 \text{ if } u_k(n)=0 \text{ and } u_{k-1}(n+j)>v_{k-1}(n+j) \text{ or}$$
$$\text{if } u_k(n)>0 \text{ and } u_{k-1}(n+j)>\sum_{z} c_{k-1}(z)u^2_{k-1}(n+z)/(2v_{k-1}(n)).$$

The first condition means that if cell n is not receiving suprathreshold levels of excitation, then only if the level of output from the excitatory cell on level k-1 exceeds the (weighted) output of all cells in its vicinity (which is the definition of $v_{k-1}(n)$) will excitatory connections from level k-1 to level k be more strongly reinforced than inhibitory connections. The equation for $u_k(n)>0$ shows that (in the simplified binary case) if a cell responds stronger than 1/2, its excitatory connections are more strongly reinforced than its inhibitory connection.

The result of these equations is that cells in a given vicinity responding most strongly to stimuli from the previous layer have their excitatory connections enhanced while weakly responding cells have their inhibitory connections enhanced. Over time, the network evolves so that the number of cells in a vicinity responding to a particular stimulus approaches unity.

Fukushima discusses different strategies for interconnecting network layers. For the case where cells in all layers have equal connectable areas on the previous layer, many layers are required to cover a large area on the lowest layer. If the connectable areas decrease with increasing depth into the network layers, the uppermost level will be connected to a larger area of the input layer. The problem with this is that too much overlap occurs, so that only one or a few cells fire on the uppermost layer. Fukushima considers this to be undesirable, since a "concept" should consist of a more complex pattern. And, no further processing on higher network levels could occur since cells on the next higher level would have only a single firing cell in their connectable areas a given time.

Fukushima uses bifurcating excitatory connections. One goes directly to the cell in the corresponding position on the next lower level, and the other is probabilistically connected in a manner such that large spatial deviations are less prone to occur than smaller deviations. with this method, each cell on the uppermost level is connected (in a somewhat stochastic manner) to the entire input field.

Simulations were run using four 12x12 layers with letters as patterns at the lowest level. Fukushima shows how individual units on the fourth layer capture the entire pattern presented to the first layer, while cells on the third and second levels capture smaller parts of the pattern.

class cells. This is a form of lateral inhibition.

The matrix functions as follows. The initial transfer weights of all cells are assumed to be small, so that the initial excitation by a pattern impinging on the imaging matrix produce only feeble, random firing of the filter cells. But when such firing occurs, the connections of the active mosaic cells to the filter cells will be reinforced, thus making the filter cells that fired more sensitive to the pattern. Whenever the pattern is again presented, the sensitized filter cells fire at a higher rate, eventually triggering a class cell. The first class cell to fire also causes the inhibitory reset cell to fire, thereby preventing all other class cells from firing. Thus, after the Hebbian conditioning, only one class cell will be active for a given pattern. But the bifurcating branch of the class cell is adaptively linked to the mosaic cells. The synapses on those mosaic cells that were excited by the pattern are reinforced. Now consider the possibility that the class cell that responded to the pattern is stimulated by some external signal, such as might result from some higher intellectual function, or just randomness. Then the original pattern will be regenerated in the mosaic cells. This is the basis for the function of "imagination" in this neural network.

Trehub also discusses a "novelty detecting" cell. This cell slowly accumulates the excitatory input from the imaging matrix. If a class cell fires, the novelty detector is reset. But if no class cell responds to the input in a certain period of time, the novelty cell would reach its threshold and fire. The consequences of the novelty cell firing would be to lower thresholds throughout the set of filter cells, allowing an unmodified filter cell to fire, capturing the pattern.

Trehub describes an organization of special purpose interacting networks called "retinoids", which perform transformations such as translation, rotation and scaling of the input pattern. Each of these subnetworks would be activated by a particular neuron, and the transformed pattern would then be re-presented to the class cells.

3. TECHNICAL DISCUSSION

OUTLINE FOR NEURAL-NETWORK BASED SPEECH RECOGNITION

Neural networks have been studied either for the abstract problem of pattern recognition, in which the nature of the patterns is not specified, or else for the specific application area of image analysis. None of the works surveyed explicitly addressed the problem of speech recognition. Yet the body of general a1d image- oriented neural network research illuminates several possibilities for neural-network based recognition systems for speech. In this section, we will outline such a system. It is based on both the neural-network research and traditional speech-recognitions systems.

At the theoretical level, speech recognition is no different than image recognition. A given (isolated) utterance can take the form of the two-dimensional pattern x(f,t), representing the magnitude of the frequency component f at time t. But without taking the specific nature of the speech signal into account, one is presented with immense computational difficulties. In order to make the necessary distinctions between utterances, the number of points required for x(f,t) would be in the thousands, and the number of cells required for successful recognition would be beyond the scope of any conceivable implementation. Furthermore, the approach would work (if at all) only for utterances of fixed length, and would not be generalizable to utterances of greater length.

Yet some of the techniques or network designs used in image analysis may be helpful in specific subtasks of speech recognition problem. Consider the well-known techniques of adjusting for variations in amplitude (normalization) and time (dynamic time warping) in a speech signal presented to a traditional recognition system based on matching stored templates. There are analogous operations for image recognition: rotation, translation, and scaling. These analogous operations have been studied, and neural networks have been proposed to effect them. See, for example, [10]. Variations of these techniques should be accessible where needed for the speech recognition system. The details of these mechanisms will not be developed in the outline of the speech recognition shown here.

The most successful speech recognition systems employ the sort of layering that is necessary to reduce the amount of learning that must be performed by a single neural network. Each layer attempts the recognition of successively larger sound units, and passes the conditionally recognized items to the next higher layer.

This layered organization is quite similar to the structure of the perceptron - an early neural network model [11]. The perceptron and related models performed a unidirectional transmission of signals from input layer to output layer. They employed unsupervised learning: feedback from the external world determined the modification of link weights. Specifically, weights on the active cells were increased if the network produced the correct response, and were decreased if the response was incorrect. The perceptron was subjected to mathematical analysis that showed its power to De less than what had been expected, and so the research interest moved on to more general models. [12] However, the experience with the perceptron and related unidirectional hierarchical models showed the difficulty of effecting supervised learning in a layered structure. Whenever such a model produces the wrong response, the learning, or modification of excitatory weights, affects the cells that operated correctly as well as those responsible for the errors. The problem is that when wrong responses occur, it is impossible to tell which level is responsible.

But if the system is layered in such a way that the outputs of each layer are understandable units, feedback-conditioning loops can be applied for each layer. This provides the basis for the proposed neural-network based speech recognition system.

Figure 1 is an outline of a neural-network based speech recognition system. The recognition system is comprised of several successive "modules" of cells with plastic links. Each of the neural modules is comprised of several layers of cells. The cell links are modified in the "supervised learning" mode: the weights of active cells are either increased or decreased depending upon an external signal that indicates whether the response of the module was correct. Thus, the output of each module is converted into a signal that can be displayed to the user of the system, who provides the response.

The need for presentation of the module output to the user does not mean that the module output must itself be a unique code indicating the object recognized. It is

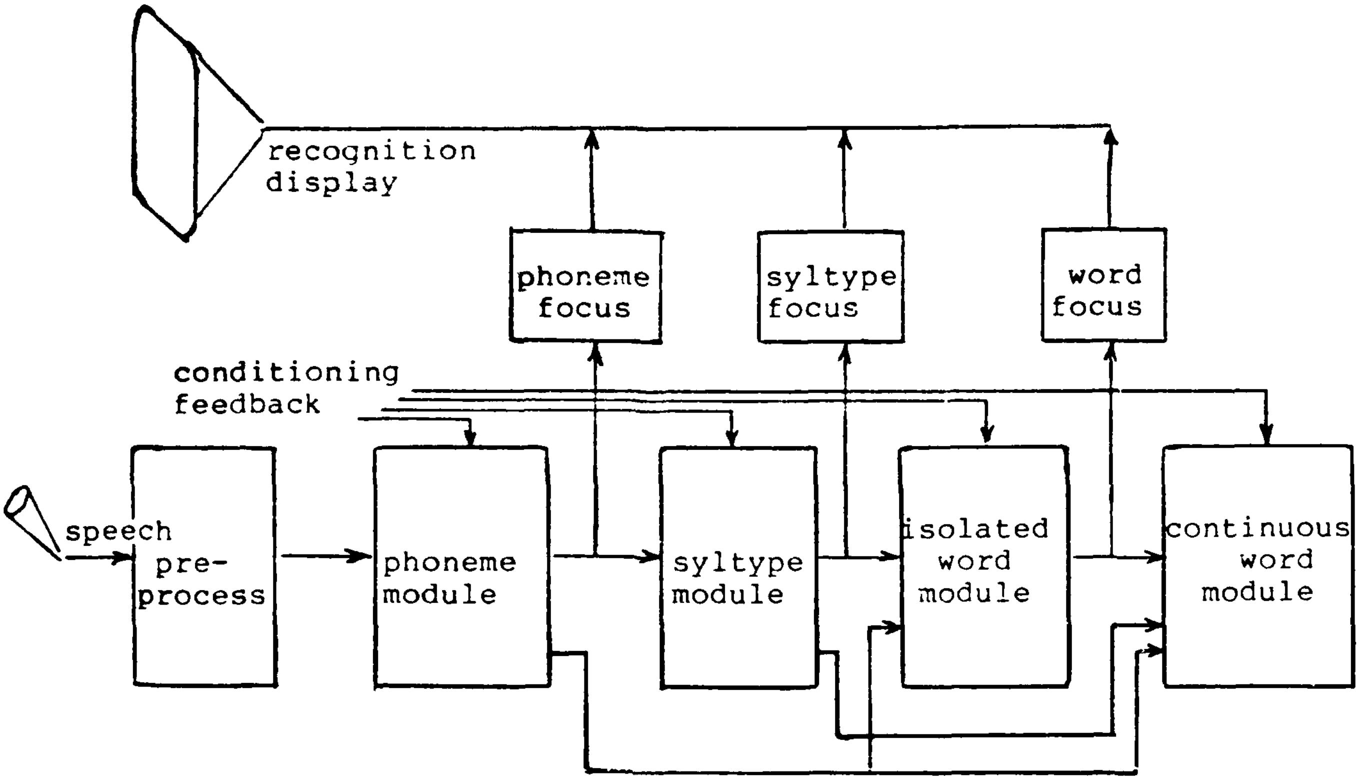


Figure 1
Outline of Neural-Network Based Speech Recognition

the nature of neural networks to process patterns in "fuzzy" way. A small increment in the weight of any link in the network could result in the variation of a single bit of the output: but this should not be considered a different result. Hence, the patterns representing the desired results of each module should be separated by Hamming distances of at least three, allowing for one-bit error correction. (For integration of the modules into one fluid system, the separation of the output codes should be much greater.) The units marked "focus" in the figure are fixed combinational logic whose purpose is to provide a single unique code for each output, from the fuzzy variations produced by the module.

It should be emphasized that the training mode, with the feedback loops, exists only to assist the system to make the correct responses, by training its operation at the appropriate level. In the operating mode, it will not be necessary for the user to provide feedback, and the modification of link weights will be turned off, or drastically attenuated.

Each of the modules operates at a different level of organization of the input signals. The input signal is digitized, accumulated into frames (at intervals of 5-30 ms.) and filtered to obtain either a set of LPC coefficients or a representation of the frequency spectrum. These, along with an indication of the amplitude of the speech signal, provide the input to the first neural module.

The first neural module determines phonemes. Phonemes are the linguistic units (or sound elements) that are defined by their ability to make the critical distinctions between words. Linguists identify about 40 phonemes for English. A dictionary of phonemes, and common words in which they occur, will be made available to the user, so that he can pronounce them (in his own voice), and train the system to recognize them. The phoneme identification (output of the "focus" unit) need be no more than six bits. However, the phoneme module output should be significantly greater. For example, a sixteen-bit representation would allow for a Hamming distance of seven between phoneme codes, providing three-bit error correction.

The second neural module recognizes linguistic units called syltypes. The syltype, which is defined and used in the Hearsay-II speech understanding system, represents a class of syllables. [13,14] The number of distinct syltypes in a given vocabulary is only a small fraction of the number of distinct syllables found in the vocabulary. The set of syltypes is determined by grouping the phonemes into classes based on similarity of sound: there may be seven such classes. Then, for a given vocabulary, a state transition diagram is developed. The states are the phoneme classes, and the transitions through the diagram (involving one to about six phoneme classes) are the syltypes.

The relationship between the number of syltypes and the vocabulary size is given in [14]. A 1000-word vocabulary requires about 250 syltypes. Thus, the output of the syltype recognition module need be only eight bits. But for seven-bit separation between syltype codes, allowing 3-bit error correction, the output of this unit should be 21 bits. The syltype focus unit will then be able to provide the user with a unique code for the recognized syltype, and allow the user to return the conditioning signal. It is assumed that the user will have an online dictionary of syltypes and associated syllables, so that he may easily determine whether the recognition is correct.

Some additional structure, apart from the neural modules, will be required to resolve the differences in time scale at each level. The output of the phoneme module, for example, is a single phoneme, but a syltype is determined by a sequence of phonemes. Therefore, some sort of memory is required at the input to the syltype unit. This memory may be a sequence of one-phoneme latches, each of which is set upon the appearance of a phoneme. (See Figure 2.)

One output of the phoneme unit is a "phoneme recognized" signal, activated whenever there is a sufficiently strong similarity between the input signal and the signals of the training set, encoded into the module link weights. Then successive activations of the "phoneme recognized" signal cause successive phonemes to be latched. And, upon the recognition of a syltype, a "syltype recognized" signal will reset the phoneme counter.

The form of synchronization implied in the structure of Figure 2 is an attempt to overcome one of the greatest difficulties of the neural network approach to speech recognition: the variation in the speed of speech. The solution shown in the figure, which reverts to sequential digital logic, is certain to be one of the weakest links

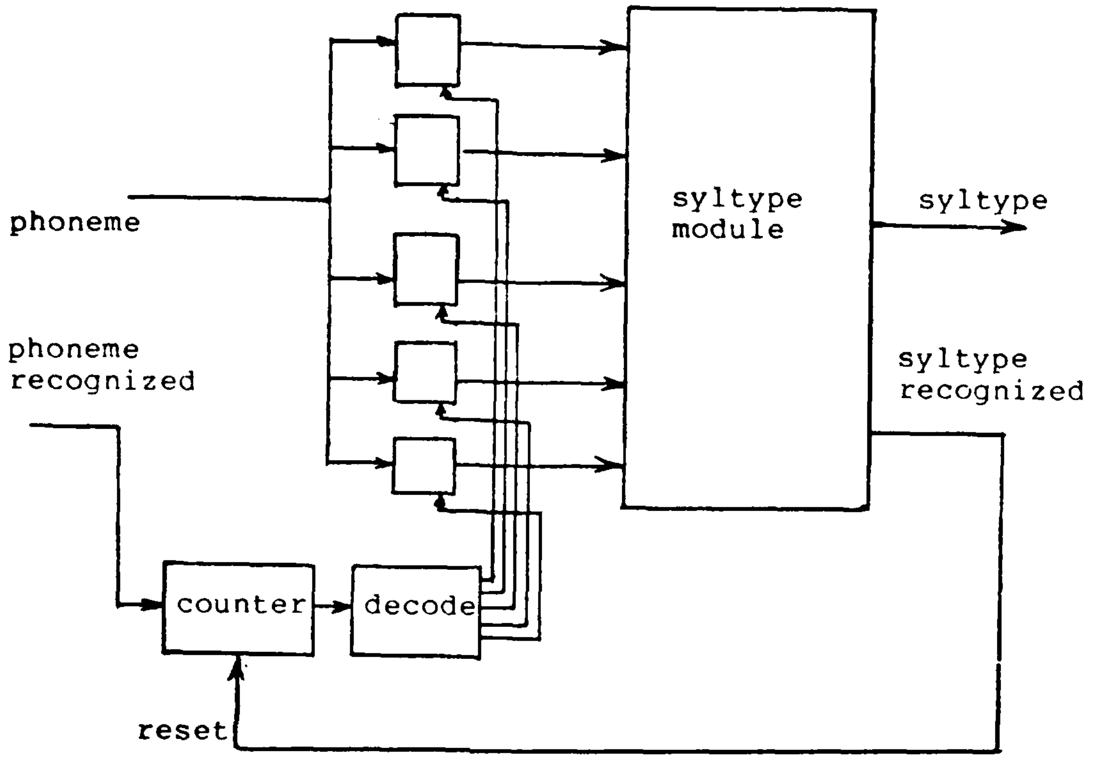


Figure 2

Accumulation of Phonemes for Syltype Recognition of the system. A single-bit error, such as the failure or inappropriate activation of the "phoneme recognized" or "syltype recognized" signals would terminate successful recognition. However, there are modifications of the proposed outline that will ameliorate this weakness. One possibility is the operation of several recognition banks in parallel, activated by recognized signals of various degrees of sensitivity. Another is the design of special-purpose neural networks that will be trained to provide the latching internal to the network.

The third neural module in Figure 1 is the isolated word recognition module. The input is a sequence of syltypes, latched in the same manner as the phonemes are at the input to the syltype recognition unit. The word recognition unit requires only a ten-bit output for a 1000-word vocabulary. A 27-bit output will provide three-bit error correction. To enable word recognition at the third neural module, the particular phonemes used in the syltype, that is, the exact syllable, should be employed. Therefore, the path from the phoneme unit to the word recognition unit is provided as well.

As it is shown in Figure 1, the continuous speech recognition module takes its input from the isolated word recognition module. This is an oversimplification. The greatest difficulty of continuous speech recognition is precisely that words cannot be isolated. Even if the word boundaries were marked, a unit that recognizes words in isolation would not necessarily recognize the word as it appears in continuous speech, because of the modification of the beginning and ending phonemes to merge with those of the adjacent words, and the variations in accent and intonation induced by the context. However, a neural network that has been conditioned to recognize words in isolation would be expected to provide some response on the "word recognized" output line, as syllables and phonemes are presented to it, even if it does not exceed the threshold required for recognition.

Thus, the scheme shown in Figure 3 is proposed. It uses the partial recognition capability of isolated word recognition modules in conjunction with a network that is conditioned only in the continuous speech recognition mode. Each of the three isolated word recognition modules is operated in parallel, to receive the identical modifications to the link weights, when operated in the isolated word (training) mode. In the continuous speech recognition mode, the elements recognized at lower levels (syltypes and phonemes) are latched at the input to the lower isolated word recognition module. Upon some activity of the "isolated word recognized" signal, the entire group of latched inputs is shifted up to the next isolated word module. Then further syltypes and phonemes are presented to the lower module, and both inputs are shifted up upon some combined activity of the "isolated word recognized" signals.

The continuous speech recognition module is the network of cells that fills the areas of Figure 3 that are not occupied by an isolated word recognition unit. It takes its input from the results of each of the isolated word recognition modules, as well as the inputs to those modules. In the training mode, continuous speech module links are modified by correct recognition of sequences of words. The isolated word units receive a lesser

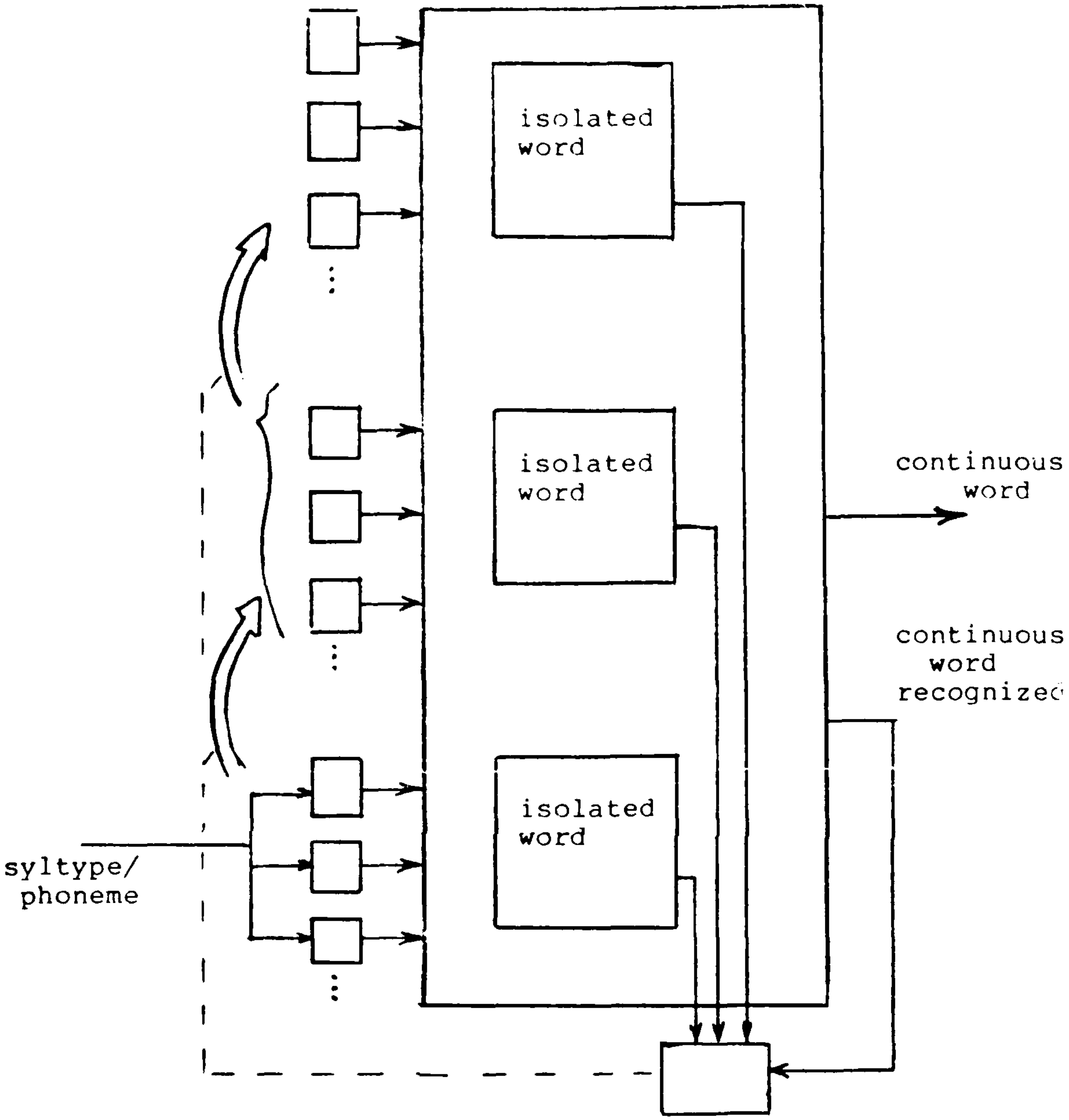


Figure 3

Continuous Word Recognition

modification during continuous speech training. The output of the continuous speech module is derived primarily from the center isolated word recognition module. But it is also influenced by cells that conditioned by the preceding and following words, and the phonemes of the transition between them. Thus, the continuous speech recognition module is trained to respond precisely to the differences between words as they appear in isolation, and as they appear in continuous speech.

4. CONCLUSIONS

The research papers that were reviewed showed that neural network models can be defined in a variety of ways, but that networks with lateral inhibition seem to offer greater discrimination capability per network cell. From Hopfield's model [1], one would expect tnat a module that can recognize 40 phonemes at a low error rate would have at least 400 cells. But the experiments by Rummelhart and Zipser [7], using competitive learning (a form of lateral inhibition), indicate that 40 distinct and fixed patterns could be recognized with just 40 cells. Because of the noisy nature of the data, the number of cells required to recognize the phonemes in real speech will be greater than that implied by these elementary results. But with the benefit of lateral inhibition, this number should not be more than a few thousand at most.

The number of cells required in the remaining speech recognition modules is comparable to that of the phoneme module. Thus, the proposed model of neural-network based speech recognition system is amenable to development and validation through simulation modelling. Within a few years, chips will be available for the implementation of the neural networks. Systems based on these hardware components, possibly organized according to the outline given here, will be built. They will be simpler, and have superior learning and recogintion capability than the sequential systems in use today. Considerable work remains to be done to realize this promise.

REFERENCES

[1] Hopfield, J.J. Neural networks and physical systems with emergent collective computational abilities. Proc. Natl. Acad. Sci. (1982) 79, pp. 2554-2558.

[2] Hopfield, J.J. and D.W. Tank. Computing with neural circuits: A model. Science (1986) 233, pp. 625-633.

[3] Hebb, D.O. The organization of behavior. (1949) Wiley, New York.

[4] Silverman, D.J., G.L. Shaw and J.C. Pearson. Associative recall properties of the trion model of cortical organization. Biol. Cyern. (1986) 53, pp. 259-271.

[5] Ackley, D.H., G.E. Hinton and T.J. Sejnowski. A learning algorithm for Boltzman machines. Cognitive Sci. (1985) 9, pp. 147-169.

[6] Kirkpatrick, S., C.D. Gelatt, Jr. and M.P. Vecchi. Optimization by simulated annealing. Science (1983) 220, pp. 671-680.

[7] Rummelhart D.E. and D. Zipser. Feature discovery by competitive learning. Cognitive Sci. (1985) 9, pp. 75-112.

[8] Grossberg, S. Adaptive pattern classification and universal recoding: I. Parallel development and coding of neural feature detectors. Biol. Cybern. (1976) 23, pp. 121-134.

[9] Fukushima, K. Cognitron: A self-organizing multilayered neural network. Bolo. Cybern. (1975) 20, pp. 121-136.

[10] Trehub, A. Neural models for cognitive processes: Networks for learning, perception and imagination. J. Theor. Biol. (1977) 65, pp. 141-169.

[11] Rosenblatt, F. Principles of neurodynamics: Perceptrons and the theory of brain mechanisms. Spartan Books, Washington, D.C. (1961).

[12] Minsky, M.L. and S. Papert. Perceptrons: An introduction to computational geometry. MIT Press, Cambridge, Mass. (1969).

[13] Erman, L.D., F. Hayes-Roth, V.P. Lesser and D.R. Reddy. The Hearsay-II speech-understanding system: integrating knowledge to resolve uncertainty. Computing Surveys (1960) 12, pp. 213-252.

[14] Smith, A.R. Word hypothesization in the Hearsay-II speech system. Proc. IEEE Conf. Acoustics, Speech and Signal Processing, (1976) pp. 549-552.

END

1-87

DTIC