Neural Network Methodologies and their Potential Application to Cloud Pattern Recognition

J. E. Peak
Computer Sciences Corporation
Monterey, CA 93943-5006

Prepared for
Atmospheric Directorate
Monterey, CA 93943-5006

Approved for public release; distribution is unlimited. Naval Oceanographic and Atmospheric Research Laboratory, Stennis Space Center, Mississippi 39529-5004.

ABSTRACT

Six artificial neural network models are explored as potential methodologies for the automated interpretation of satellite cloud images. The Multi-layer Perceptron Network is chosen to be the most applicable to the image interpretation problem. A complete, mathematical description of the methodology is presented. The neural network's classification capability is demonstrated using simple, geometric patterns and alphabetic characters. A more complex test using GOES infrared imagery shows that the neural network can distinguish 53 of 54 large-scale cloud patterns. An architecture for a complete, automated cloud feature recognition system is proposed.

ACKNOWLEDGMENTS

The author gratefully acknowledges the support of the sponsor, Office of Naval Technology, Code 22, Mr. James Cauffman, Program Element 62435N, for making this effort possible.

TABLE OF CONTENTS

NEURAL NETWORK METHODOLOGIES AND THEIR POTENTIAL APPLICATION TO CLOUD PATTERN RECOGNITION

1. Introduction

Artificial neural network models or "neural nets" have been studied for many years as potential methods for solving speech and image recognition problems. These models are composed of many nonlinear computational elements operating in parallel in networks. Artificial neural net structure is based on present understanding of biological nervous systems.

The neural network approach has many advantages. The parallel structure of such networks provides very high computation rates. Neural nets typically provide a greater degree of robustness than sequential methods because they have many more processing units. Thus, poor or missing input data to a few units does not significantly impair the overall performance of the network. Another advantage is that the connection weights can be adapted during use of the algorithm, so that performance is continually improved based on current results. In the interpretation of image features such as clouds, the training sample can never contain examples of every possible feature configuration. Neural networks can continue to adapt as new examples are encountered. Neural net classifiers are nonparametric and make weaker assumptions about the shapes of underlying distributions than traditional statistical classifiers. It is for this reason that neural nets work well when distributions are non-Gaussian.

The automatic interpretation of satellite images would be a powerful aid to Navy forecasters. A necessary step in the interpretation process is to identify the cloud features that are comprised of collections of contiguous and noncontiguous cloudy pixels. To accomplish this goal using standard pattern recognition techniques would require enormous amounts of processing resources and would likely have only limited success. In this paper, neural nets are presented as a possible alternative way to acquire the necessary processing capacity using networks of simple processing elements operating in parallel.

In the next section, six different neural network configurations that accomplish classification are presented. It will be shown that the multi-level perceptron configuration is most applicable to the satellite image feature recognition problem. In Section 3, the mathematical form of the multi-level perceptron network will be presented. An implementation of the algorithm, called "NOARL-NEURAL," has been developed. The program has been tested on simple problems, the results of which are in Section 4. Based on the results of these tests, an architecture is proposed for a comprehensive cloud image analysis system that uses neural network, expert system and statistical methodologies in concert. Finally, the conclusions of this preliminary research will be presented.

2. Types of Neural Networks

There are many different types of neural net models. The various models are designed to accomplish different goals. For example, some are designed to provide a parallel content-addressable memory storage capacity. Similar networks can be used to supply missing information for incomplete input sets. Constraint-satisfaction neural networks find solutions to problems with a large number of constraints. These networks are very useful for problems in which there is no solution that matches all of the constraints. In such cases, the solution satisfies as many constraints as possible. It may be possible to use such a methodology to satisfy the constraints of multi-channel atmospheric profiling from satellite radiance data. Similarly, the solution of a numerical model's cumulus or boundary-layer parametrization could be solved by using embedded neural networks in the model. Pattern-associators are another powerful type of neural network, in which learning from examples is accomplished. Similar to pattern associators are the so-called auto-associators which detect regularity in a set of input patterns and can restore distorted or incomplete input patterns to their original form. Such networks have been used to filter noise from signals or to make generalizations about examples -- a sort of automatic rule generation. Finally, biological neural researchers have used neural methods to simulate human cognitive processes in an attempt to understand human perception.

In this section, six neural models that accomplish pattern classification will be examined with the goal of choosing the best one to apply to the satellite image problem. The different models can be specified by the network topology, node characteristics, and training or learning rules. The network topology specifies how the various artificial neurons, or "nodes," are distributed and interconnected. The node characteristics define how each node processes the signals it receives from other nodes, and what signal it outputs. The learning rules define how the weights are adjusted if the desired output is not attained.

The simplest artificial neuron or node sums N weighted inputs and passes the result through a nonlinear function as shown in Figure 1. The node is characterized by an internal threshold or offset θ and by the type of nonlinearity. Three such nonlinearities, shown in Figure 1, are the hard-limiter, threshold logic and sigmoid functions. Very complex neural models may include a time dependency and a more complex mathematical operation than summation. In an image classifier, the inputs might be the gray scale level of each pixel and the output classes might represent different objects.

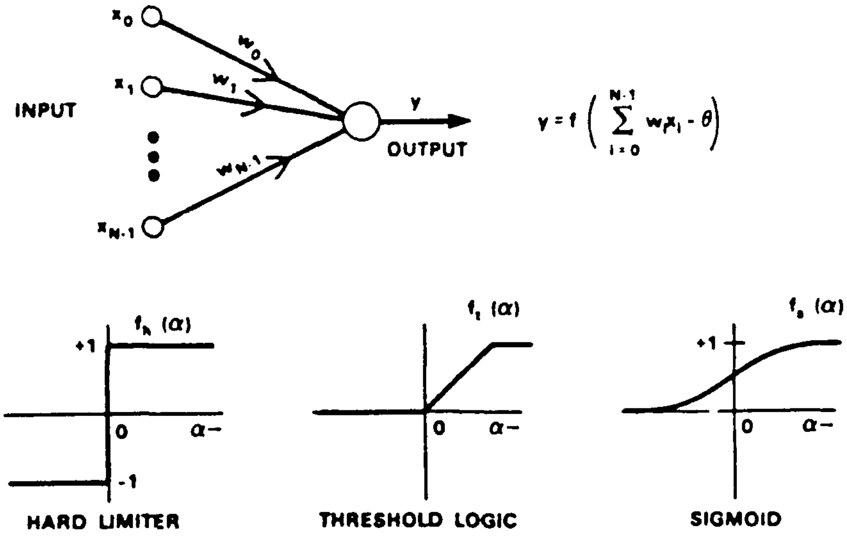

Figure 1. Artificial neuron which forms a weighted sum of N inputs and passes the result through a nonlinear function. Three types of nonlinearities are shown.

In an excellent survey of neural networks, Lippman (1987) summarized six neural network classifiers. A taxonomy of these methodologies is presented in Figure 2. The six net types are first divided into those that use binary versus continuous input data values. Next, the nets are divided between those trained with or without supervision. Supervised training occurs when the correct output patterns are specified for each training case. In unsupervised training, the network itself groups the training set outputs, thus defining the output classes. The classical statistical algorithms listed at the bottom of Figure 2 are those most similar to the corresponding neural net in methodology or function. The six methodologies will now be examined with the goal of choosing the one most applicable to the satellite image interpretation problem.

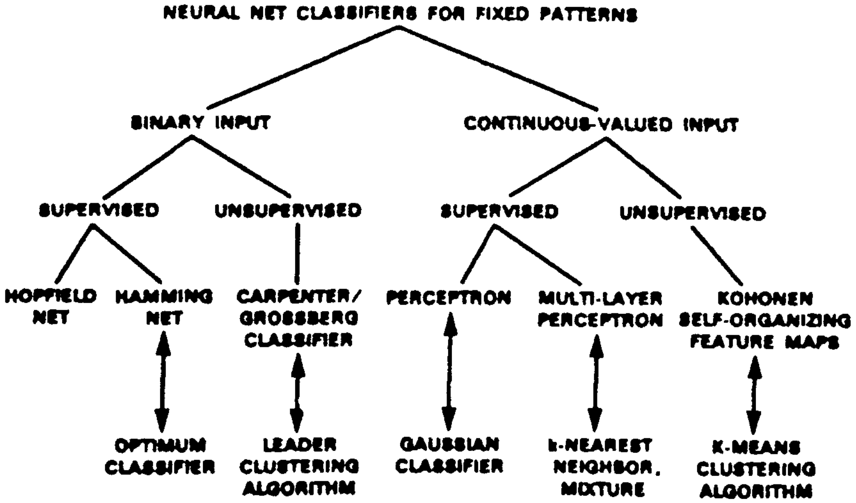
NEURAL NET CLASSIFIERS FOR FIXED PATTERNS
BINARY INPUT
CONTINUOUS-VALUED INPUT
SUPERVISED
UNSUPERVISED
SUPERVISED
UNSUPERVISED
HOPFIELD NET
HAMMING NET
CARPENTER/ GROSSBERG CLASSIFIER
PERCEPTRON
MULTI-LAYER PERCEPTRON
KOHONEN SELF-ORGANIZING FEATURE MAPS
OPTIMUM CLASSIFIER
LEADER CLUSTERING ALGORITHM
GAUSSIAN CLASSIFIER
k-NEAREST NEIGHBOR, MIXTURE
K-MEANS CLUSTERING ALGORITHM

Figure 2. A taxonomy of six neural nets used for classification problems. Classical algorithms that are most similar to the neural net model are shown at the bottom (From Lippman, 1987).

2.1 Hopfield Net

The Hopfield net is one of three nets in Figure 2 that use binary input. Such nets are appropriate for input values from images where the pixel data are determined to be greater than or less than some threshold value. The Hopfield net can be used as a context-addressable memory or to solve optimization problems. This net's nodes use the hard-limiter nonlinearity for binary inputs of values +1 and -1. The output of each node is fed back to all other nodes by a symmetrical weight matrix. By providing a set of examples and specified output classes, the weight matrix can be iteratively derived. When an unknown pattern is presented to the Hopfield net, it iterates in discrete time steps until the outputs no longer change on successive iterations. When used as a memory model, the resultant output is the desired memory information. When used as a classifier, the pattern specified by the outputs is compared to the class outputs to see if an example pattern is matched. If none is matched, a "no match" result occurs.

Hopfield nets can be used to provide memory retrieval or classification of very noisy input data. However, there are two major limitations to these nets. First, the number of output pattern classes that can be recalled is quite limited unless the network size is quite large. If too many patterns are stored for a given net size, the net may converge to a spurious pattern different from all of the example patterns, and the network will not be able to classify at all. This problem occurs when the number of output pattern classes is greater than 0.15 times the number of nodes in the net. Thus, a Hopfield net to distinguish 10 classes would require at least 67 nodes. Since each node is connected to all the others, the weight matrix would be 67x67, or almost 4500 weights. Thus, the training time and storage requirements could quickly become too large for big problems. The second problem occurs when output classes are too similar. In this case, the net may converge to the wrong class given similar input patterns.

2.2 Hamming Net

The Hamming net (Fig. 2) is similar to the Hopfield net except that nodes use the threshold logic nonlinearity (Fig. 1) and the network consists of two layers. The first layer operates much like a Hopfield net. Binary patterns are used as inputs, but the outputs are matching scores defined by the so-called Hamming distance between the input and the correct class for each case. The Hamming distance is the number of bits in the input which do not match the corresponding class example bits. The outputs of the first layer, after convergence, are used as an input layer to the second layer. This layer iterates until only one output node is positive and the rest are negative. This node corresponds to the correct class.

The Hamming net performs as well or better than the Hopfield net while requiring fewer connections, typically only as many nodes as the sum of the number of inputs plus the number of classes. Thus, for bigger problems, the Hopfield net size grows exponentially while the Hamming net size grows linearly. Also, the Hamming net does not have the spurious output problem that the Hopfield net has. The Hamming net is a direct neural net implementation of the "optimum classifier" algorithm used in communications theory to reduce noise in binary signals.

2.3 Carpenter/Grossberg Net

The Carpenter/Grossberg net is an implementation of their Adaptive Resonance Theory. This net forms clusters and is trained without supervision. The net structure is similar to the Hamming net except that a threshold value is used to measure how close the outputs from successive cases are to each other. The first case defines a locus for the first cluster. The next case is processed through the net and is clustered with the first case if the distance between them is less than the threshold. If not, it becomes the locus for a new cluster. The process is repeated for all of the test cases. Thus, the number of classes grows with time, especially when the threshold distance is small. Each new class requires a new node, and as many new weights as two times the number of inputs, to compute matching scores.

The Carpenter/Grossberg net can work well with perfect input patterns. However, even a small amount of noise can cause problems. With noise, or a threshold value that is too small, all available nodes can quickly be used up and similar patterns will still fail to be clustered. This type of net is very similar to the "sequential leader clustering" statistical algorithm.

2.4 Single-Layer Perceptron Net

The single-layer perceptron and two other nets in Figure 2 can use both continuous-valued and binary input data, which is simply a subset of continuous-valued data. A single perceptron node is depicted in Figure 3. This artificial neuron decides whether an input set belongs to one of two classes denoted A or B. The weighted sum of the input values minus some optional threshold is passed through the hard-limiter nonlinearity, yielding an cutput of +1 or -1. The decision rule is to choose class A if the output is +1 and class B if -1. Lippman (1987) graphically depicts the behavior of such a perceptron by plotting a map of the multidimensional space defined by the input variables. The first two components of that space are depicted in Figure 4. The weighted sum of the input values defines a pair of decision regions that specify which input values result in a class A or a class B response. Thus, the perceptron defines a hyperplane that

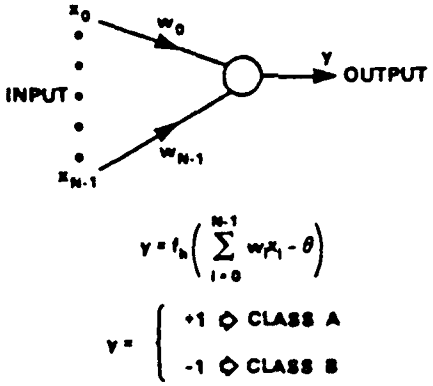
x_0
w_0
INPUT
x_{N-1}
w_{N-1}
Y
OUTPUT
$y = f_h\left(\sum_{i=0}^{N-1} w_i x_i - \theta\right)$
$y = \begin{cases} +1 & \text{◇ CLASS A} \\ -1 & \text{◇ CLASS B} \end{cases}$

Figure 3. A single-layer perceptron that classifies input vectors into two classes denoted A and B.

separates the two decision regions. In the two-dimensional example (Fig. 4) the hyperplane is a line, and inputs above the line lead to class A and those below to class B. Thus, it can be seen that when the classes are not overlapping the perceptron gives a perfect classification of the cases. When there is an overlap, there must be some incorrect classifications. By changing the nonlinearity to the threshold-logic (Fig. 1), and iteratively correcting the weights on every trial by an amount proportional to the difference between the desired and actual outputs, the perceptron converges to the least-mean-square solution in these situations. It will be shown in the next section that the problem of overlapping decision regions is solved by adding more layers of perceptrons. The difficulty in this approach is the question of how to adjust the weights of previous levels in response to errors occurring in higher levels. It was the solution of this problem that renewed interest in the neural network

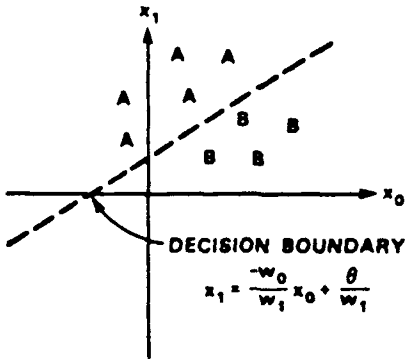

Figure 4. Graphical representation of the space formed by two inputs and the hyperplane (line) formed by a neural net which classifies the cases into two regions.

methodology in recent years. The closest analogy to perceptron nets is the maximum-likelihood Gaussian classifier.

2.5 Multi-layer Perceptron Net

Multi-layer perceptron nets have one or more layers of nodes between the input and output layers. These so-called "hidden layers" solve many of the limitations of single-layer perceptrons, but were impossible to train effectively until recently. The training problem was solved when Rumelhart et al. (1986) devised the "back-propagation" algorithm[1]. In back-propagation, the error between the actual and expected output is used to adjust not only the weights contributing directly to the output, but also the weights contributing to all of the contributing units in previous layers.

The power of multi-level perceptron nets derives from the nonlinearities used in the nodes. If the nodes were linear processors then a single-layer network could do the same calculations as a multi-layer network. The comparative capabilities of single-layer, two-layer and three-layer perceptron nets using hard-limiter nonlinearities can be seen in Figure 5. As in the previous section, the decision regions for various problems are described geometrically. The second column (Fig. 5) lists the types of geometric decision regions that can be defined by the

[1] The author was fortunate last summer to have attended a one-day tutorial on neural networks taught by Dave Rumelhart and the leader in speech-recognition neural networks, Terry Sejnowski.

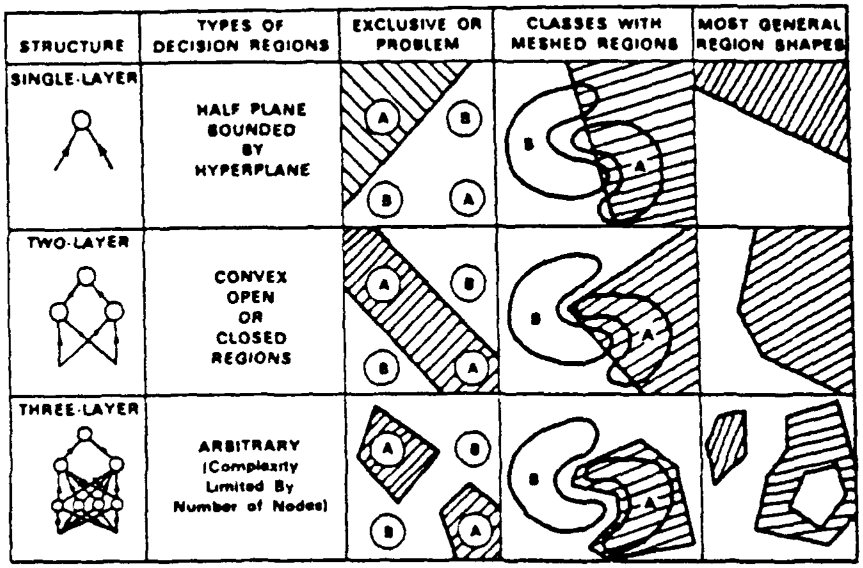
STRUCTURE
TYPES OF DECISION REGIONS
EXCLUSIVE OR PROBLEM
CLASSES WITH MESHED REGIONS
MOST GENERAL REGION SHAPES
SINGLE-LAYER
HALF PLANE BOUNDED BY HYPERPLANE
TWO-LAYER
CONVEX OPEN OR CLOSED REGIONS
THREE-LAYER
ARBITRARY (Complexity Limited By Number of Nodes)
A
B

Figure 5. Types of decision regions that can be formed by single- and multi-layer perceptron nets with one and two layers of hidden units and two inputs. Shading denotes decision region for class A. Smooth, closed contours bound input distributions for classes A and B (from Lippman, 1987).

net structure described in the first column. The third column shows a graphical depiction of the exclusive-or problem and decision regions (shaded vs. nonshaded) that might result for each net. Notice that the single-layer perceptron can not solve the exclusive-or problem because the regions cannot be separated by a hyperplane. The network with one hidden layer, however, can solve the exclusive-or problem because the extra decision layer allows it to define a convex region in the space defined by the inputs (Fig. 5). It turns out that the number of sides on the convex region is determined by the number of nodes in the hidden layer. By adding a second hidden layer, two distinct convex regions can be defined (Fig. 5).

The fourth column in Figure 5 depicts network decision regions for a two-class problem in which a portion of each solution region extends into the concave area formed by the other. For these "meshed" decision regions, the single-layer net again cannot solve the problem. Because the solution space requires a concave decision region, even the net with one hidden layer cannot solve the problem completely (Fig. 5). Only the use of a second hidden layer can accomplish such a concave decision region (Fig. 5). The last column in Figure 5 depicts the most general solution region shape for each net configuration. Since a net with two hidden layers can form an arbitrarily complex solution region, there is never a need to structure a net with more than two hidden layers. Added complexity at this point results from using more nodes in the hidden layers rather than more layers.

The analysis in Figure 5 applies only to perceptron nets using the hard-limiter and having a single output node. In actual practice multiple output nodes and the sigmoidal nonlinearity are used. The behavior of these nets is even more complex, resulting in curved decision region boundaries rather than the straight-line segments depicted here.

Multi-layer perceptron nets can solve a wide range of deterministic problems such as the exculsive-or problem, and have been used in speech synthesis and recognition and visual pattern recognition. The back-propagation method has some convergence problems; notable a tendency to converge to locally-minimum least-squares solutions rather than the global minimum. As will be shown in the next section, there are mathematical ways to overcome this problem. Another problem is that convergence may be slow, requiring many passes through the training set. This problem is especially true when the solution regions are complex in shape and disconnected. Continuing the analogy with conventional statistical methods, the multi-layer perceptron yields results similar to the K-Nearest Neighbor classifier (Lee et al., 1990).

2.6 Kohonen Self-Organizing Feature Maps

This methodology is an unsupervised learning technique that uses continuous-valued inputs. Contrary to the Carpenter/Grossberg net (Sec. 2.3), Kohonen's technique uses a predetermined number of output classes. There is one output node for each class, and the output nodes are extensively interconnected and organized in a two-dimensional array. Input cases are presented sequentially without specifying a desired output. A neighborhood is defined around each output node resulting in weights that cause output nodes to be topologically close to others sensitive to similar inputs. The neighborhood size decreases in time, causing a uniform classification of input patterns across the range of output nodes.

The Kohonen net has been used in speech and pattern recognition problems. It is less sensitive to noise because the number of classes is predetermined. A drawback is that the net tends to be sensitive to the order that the training cases are processed, especially for small training sets. In many respects it is similar to the conventional K-Means Clustering algorithm.

2.7 Network Choice for Satellite Image Interpretation

If the satellite data being used were binary (e.g., cloud/no-cloud) and the classes (i.e., features to be recognized) were well-defined shapes, the Hamming net might be a suitable approach. However, the outputs consist of fronts, cloud lines, vortices, etc., all of which can occur in many shapes and sizes. Thus, the solution regions for this problem are likely complex in shape and numerous. In addition, there might be some important information in the radiance or gray-scale information which would be masked by using binary data. Because of these considerations, it seems clear that the multi-layer perceptron net has the best chance at solving the satellite image interpretation problem. Since we know what the desired output classes are, the Kohonen classification methodology is probably not necessary.

Since the multi-layer perceptron methodology is indicated, the following section presents a mathematical description of the network function and training procedure.

3. Multi-level Perceptron Neural Networks

Multi-layer perceptron nets have one or more layers of "hidden" units between the input and output layers. The activation 'a' of a given unit is a function of the weighted sum of the input units to which it is connected. Thus, for any unit i the activation due to a training pattern p is given by

$$a_i = f(net_{pi})$$

where

$$net_{pi} = \sum_j w_{ij}a_{pj} + bias_i$$

and f is the sigmoid function (Fig. 1). The bias term is analogous to the perceptron offset θ, and can be thought of as a scale factor. The least-mean-square (LMS) learning procedure of Widrow and Hoff (1960) is used to adjust the weights. An error function is defined that depends on the summed square of the error for a set of training cases. The error of an output node is simply the difference between the target output t_{pi} and the network output a_{pi}. Thus, the error E is given by

$$E = \sum_p \sum_i (t_{pi} - a_{pi})^2.$$

The LMS error function defines an error space or surface for the various potential weight values. The network is trained by adjusting the weights such that the network error moves down the error gradient toward a minimum value. This "gradient descent" method makes the change in weight proportional to the negative of the derivative of the error with respect to each weight. Thus, the learning rule (also called the "delta rule") is

$$\Delta w_{ij} = -k \frac{\partial E_p}{\partial w_{ij}}$$

where k is the constant of proportionality. After taking the derivative of E, the delta rule becomes

$$\Delta w_{ij} = \varepsilon \delta_{pi} a_{pj}$$

where $\varepsilon = 2k$ is the learning rate and

$$\delta_{pi} = (t_{pi} - a_{pi}) f'_i(net_{pi})$$

for output units. Here, net_{pi} is defined as above and $f'_i(net_{pi})$

is the derivative of the activation function with respect to a change in the net input to the unit. Since we require this function to exist, the discontinuous hard-limiter and threshold logic nonlinearities (Fig. 1) cannot be used. The form of sigmoid function used is the logistic function

$$a_{pi} = \frac{1}{1 + e^{-(net_{pi})}}$$

The derivative of this function can be shown to be

$$\frac{da_{pi}}{dnet_{pi}} = a_{pi}(1-a_{pi}).$$

Thus, the error δ_{pi} for an output node is given by

$$\delta_{pi} = (t_{pi} - a_{pi})a_{pi}(1-a_{pi}).$$

For hidden units, there is no target value t_{pi}. The back-propagation technique propagates the error backward through the network beginning with the output nodes. Since the error of an output node is partially due to the inputs it receives from nodes in previous layers, an output node error should result in the adjustment of earlier weights. It is assumed that the error propagation should be proportional to the contributing weights; i.e., those nodes contributing more to the error should be adjusted more. Thus, for hidden units we have

$$\delta_{pi} = \sum_{k} \delta_{pk} w_{ki} f'_{i}(net_{pi})$$

where k is the index of the unit to which unit i contributes.

Backward propagation involves two steps. First, the input pattern is fed into the net and propagated forward to compute output values a_{pj} for each unit. This value is compared to the target values, resulting in a δ term for each output unit. The second phase is a backward pass through the network during which the δ term is computed for each node. After these two phases, the delta term for each weight can be computed.

It is important to note that the activation derivative, $a_{pi}(1-a_{pi})$, is zero when the activation is a maximum (1.0) or a minimum (0.0), and reaches a maximum when $a_{pi}=0.5$. Thus, weights will change most for activations not yet committed to being on (1.0) or off (0.0) in the network.

Since back-propagation is a gradient-descent procedure, and since the error surfaces are not bowl-shaped, there may be a problem of settling into a local minimum. This problem may be alleviated by the use of many hidden units.

The number of training passes required before the net achieves equilibrium can be quite large if the learning rate is too small. Although a large learning rate causes the weights to change faster, it can lead to instabilities that cause the weights to oscillate rather than converge. One way to increase learning rates without causing oscillation is to include a momentum term in the delta equation:

$$\Delta w_{ij}(n+1) = \varepsilon(\delta_{pi}a_{pj}) + \alpha\Delta w_{ij}(n)$$

where n is the number of trials using a learning sample and α is the momentum constant. Thus, the derivative from the most recent learning trial is used to stabilize the adjustment of weights on the current trial.

A program to derive multi-layer perceptron neural networks has been programmed and tested. The routine, called "NOARL-NEURAL," currently runs on a Z-248 or on the HP-835. A copy of the program and a User's Guide are available from the a''thor.

4. Tests of NOARL-NEURAL

In this section, the pattern-recognition capability of NOARL-NEURAL is demonstrated. First, a neural network to classify simple geometric patterns is derived. For a slightly more complex problem, the network is next trained and tested with alphabetic characters. Finally, a crude application to interpretation of satellite imagery is presented. As will be seen, the surprisingly good results of these tests indicate a strong potential for the use of neural nets to interpret satellite imagery.

4.1 Tests on Geometric Patterns

In this test, three geometric shapes are classified: circles, triangles and squares. The data for these cases were generated by hand-sketching shapes on a 5x5 grid such as the one in Figure 6a. For each grid square, the amount covered by a shape's line was visually estimated. Thus, for the example in Figure 6b, square 1 has zero coverage, square 2 has 0.3 (30%) coverage, etc. The training set included six sketches for each of the three shapes, for a total of 18 cases (Fig. 7). Notice that the circles are not always centered on the grid, and the squares and triangles are presented in different orientations.

The 25 grid-coverage values define an input set for a neural network. This network consists of two layers; 25 inputs feeding

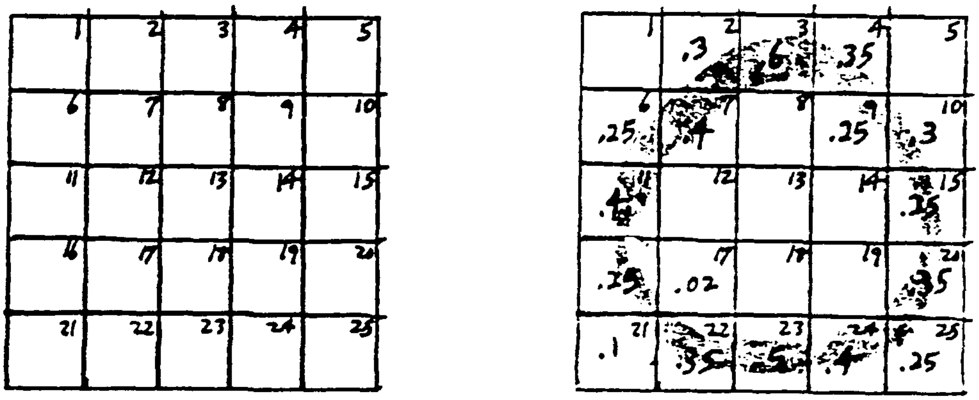

Figure 6. a) Grid of 5x5 squares used as a digitizing background for sketches of geometric shapes, and b) example of a circle drawn on the grid and the estimated coverage in each square.

to a hidden layer of three units and an output layer also of three units. Each output corresponds to one of the shapes. Thus, a circle is assigned the output 0-0-1, a square has the output 0-1-0, and a triangle is given 1-0-0. The network was trained on the 18 learning cases. The net converged very quickly, requiring only 100 passes of the 18 cases before the total sum of the squared error reached very low levels. The 18 dependent sample cases were then processed through the network. As can be seen in Table 1, all 18 were classified correctly.

A test sample of 12 cases was next defined (Fig. 8). There was no conscious effort to make these cases similar to the test cases; in fact, the squares are particularly dissimilar to those in the dependent sample (Fig. 8 vs. Fig. 7). The network classi-

Table 1. Activations of a neural net with three output nodes, and corresponding target activations, for 18 dependent sample geometric shapes. Outputs corresponding to triangles, squares and circles are 1-0-0, 0-1-0 and 0-0-1, respectively.

	Output Node Activations			Target Activations		
Case	1	2	3	1	2	3
1 T1	0.99	0.03	0.00	1.00	0.00	0.00
2 S1	0.00	0.97	0.01	0.00	1.00	0.00
3 C1	0.00	0.02	0.99	0.00	0.00	1.00
4 T2	0.99	0.02	0.00	1.00	0.00	0.00
5 S2	0.01	0.98	0.00	0.00	1.00	0.00
6 C2	0.00	0.05	0.98	0.00	0.00	1.00
7 T3	0.99	0.03	0.00	1.00	0.00	0.00
8 S3	0.01	0.98	0.01	0.00	1.00	0.00
9 C3	0.00	0.03	0.99	0.00	0.00	1.00
10 T4	0.99	0.04	0.00	1.00	0.00	0.00
11 S4	0.01	0.98	0.00	0.00	1.00	0.00
12 C4	0.00	0.02	0.99	0.00	0.00	1.00
13 T5	0.99	0.02	0.00	1.00	0.00	0.00
14 S5	0.01	0.97	0.01	0.00	1.00	0.00
15 C5	0.00	0.01	0.98	0.00	0.00	1.00
16 T6	0.99	0.02	0.00	1.00	0.00	0.00
17 S6	0.01	0.94	0.01	0.00	1.00	0.00
18 C6	0.00	0.02	0.99	0.00	0.00	1.00

fication of the independent cases (Table 2) was very good -- only one incorrect classification. Case ST2 was correct, but the 0.00-0.63-0.31 output values indicate that the net considered this square to be somewhat similar to a circle. The case ST3 outputs were 0.01-0.16-0.15. Thus, the net didn't find a lot of similarity with any of the test shapes, but still just managed to favor the correct classification. The incorrect classification of circle CT4 as a square was clearly wrong, as evidenced by the 0.02-0.96-0.00 output.

This test is admittedly crude in that inexact, hand-generated, visually-estimated input data are used. The small sample sizes also preclude any generalization based on these results.

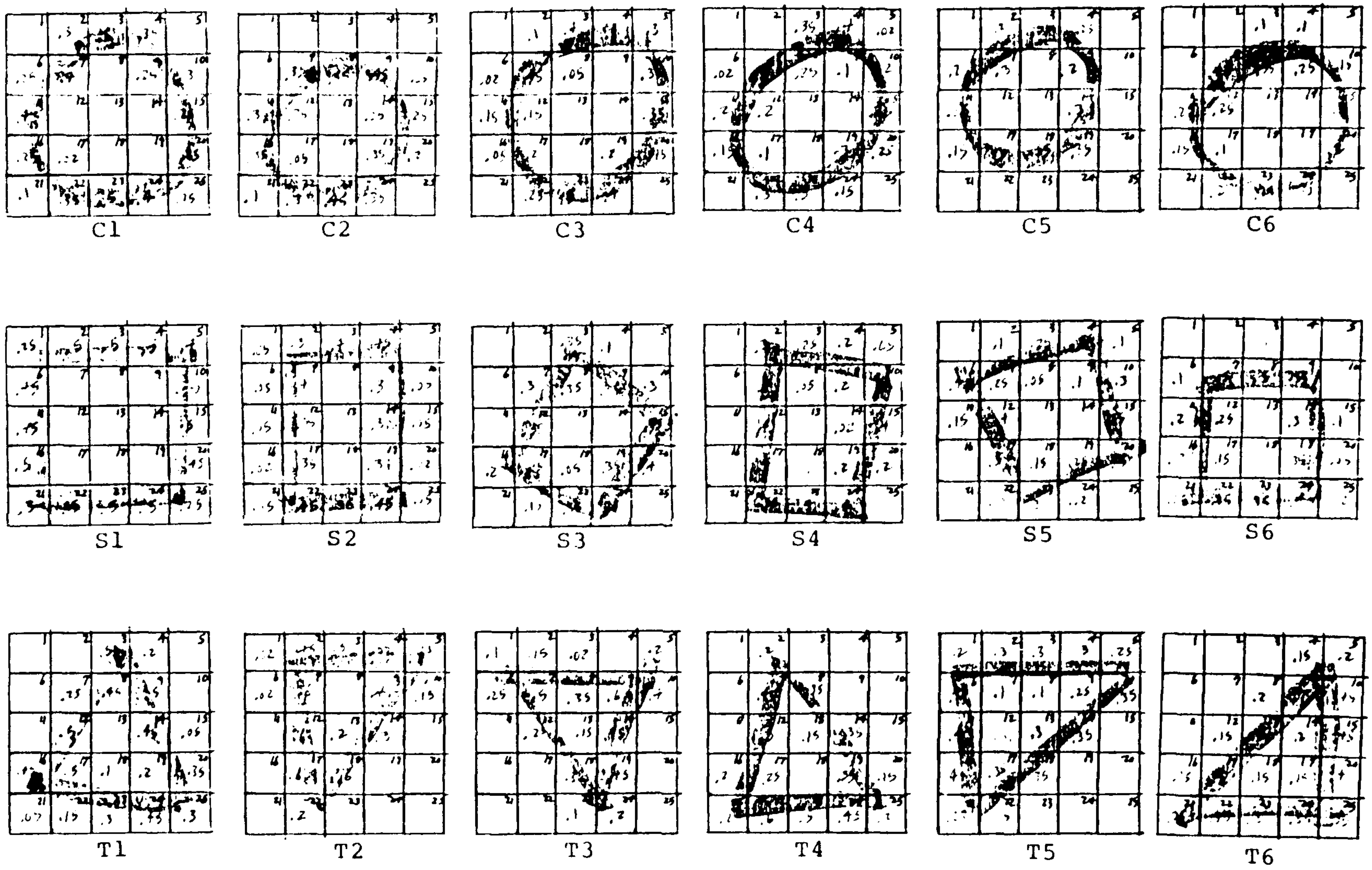

Figure 7. Dependent sample of 18 circles, squares and triangles used to train the neural net. (Labels are case numbers.)

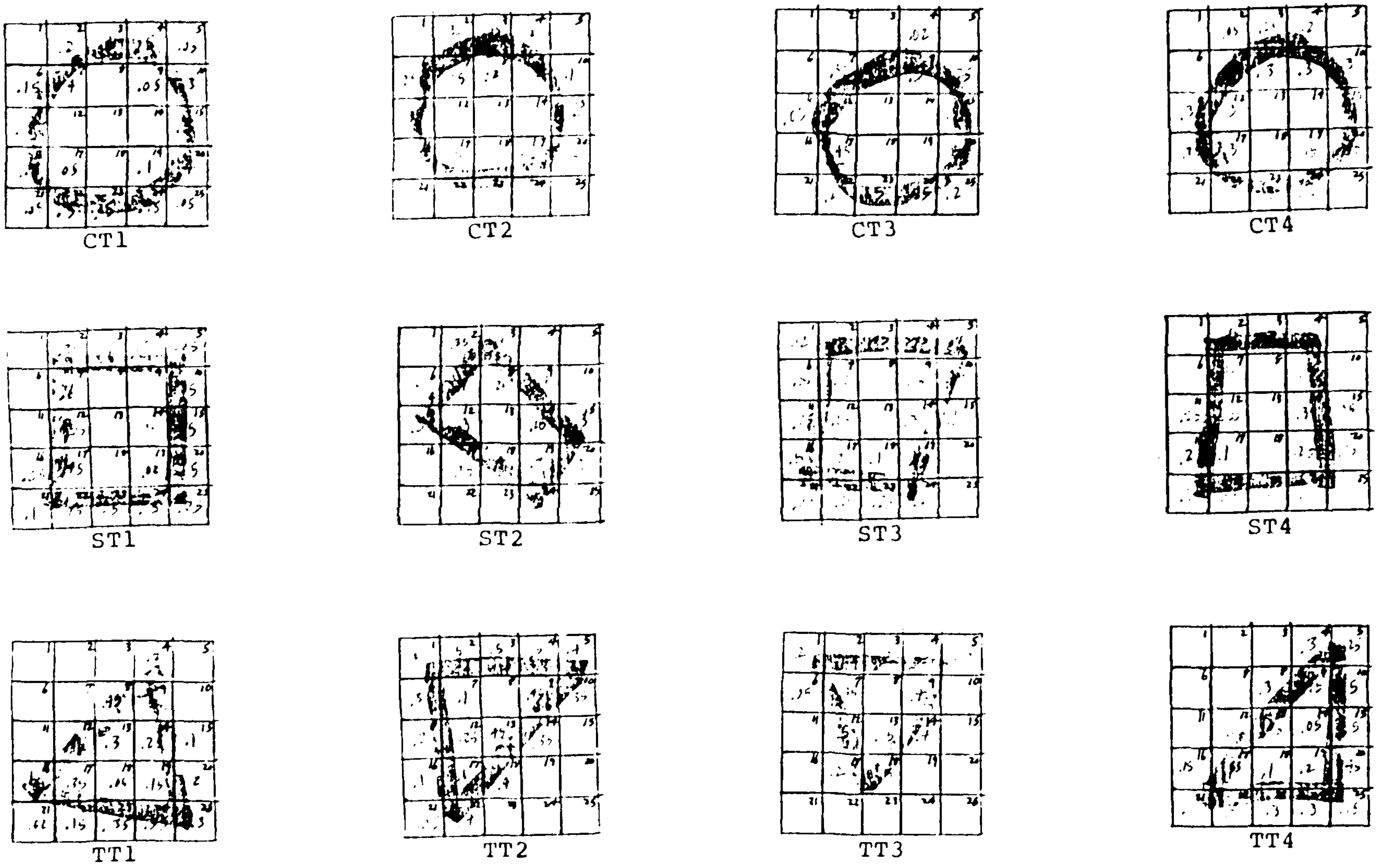

Figure 8. Independent sample of 12 geometric shapes used to test the neural net. (Labels are case numbers.)

Table 2. As in Table 1 except for 12 independent sample cases.

	Output Node Activations			Target Activations		
Case	1	2	3	1	2	3
1 TT1	0.99	0.02	0.00	0.10	0.00	0.00
2 ST1	0.02	0.96	0.00	0.00	1.00	0.00
3 CT1	0.00	0.02	0.99	0.00	0.00	1.00
4 TT2	0.99	0.02	0.00	1.00	0.00	0.00
5 ST2	0.00	0.63	0.31	0.00	1.00	0.00
6 CT2	0.00	0.03	0.99	0.00	0.00	1.00
7 TT3	0.99	0.02	0.00	1.00	0.00	0.00
8 ST3	0.01	0.16	0.15	0.00	1.00	0.00
9 CT3	0.02	0.96	0.00	0.00	0.00	1.00
10 TT4	0.99	0.03	0.00	1.00	0.00	0.00
11 ST4	0.01	0.97	0.00	0.00	1.00	0.00
12 CT4	0.00	0.14	0.92	0.00	0.00	1.00

However, one goal of this test was to verify the function of the NOARL-NEURAL program. The good independent sample results indicate that the program is indeed deriving a skillful neural network. Despite the low number of hidden layer units, the net still shows the ability to classify most of the cases correctly.

4.2 Tests on Alphabetic Characters

Although they are presented in different orientations, all of the geometric shapes in each of the previous example classes are basically the same. To present a more difficult task, the net was given four alphabetic characters to recognize: 'A', 'a', 'B' and 'b'. What makes this test more difficult is that there are only two output classes; that is, the net must determine whether the input pattern indicates the first or the second letter of the alphabet, regardless of upper or lower case. Thus, the network must accomplish the same outputs for the very dissimilar input patterns 'A' and 'a', and for 'B' and 'b'.

As in the previous test, sketches of the letters were made on the 5x5 grid (Fig. 9). There are 6 upper- and 6 lower-case occurrences for each letter, for a total of 24 examples in the training set. Notice that the letters are of different sizes and shapes, and many occupy different locations in the grid. The letters 'A' and 'a' were assigned the outputs 1-0, while 'B' and 'b' were assigned 0-1.

The network for this test consisted of 25 inputs, 4 hidden units and 2 output units. Again, the net converged very quickly and training was stopped at 100 iterations. When the 24 dependent sample cases were processed through the net, all were correctly classified (Table 3).

An independent sample of 12 cases (Fig. 10) was used to test the network. All but one were correctly classified (Table 4). In the incorrectly classified case (AT4), a lower-case 'a' was classified as a 'b', probably due to a somewhat large lower loop size compared to the lower loops of the training set examples.

Again, this test is for a small sample size and the net configuration is small and only two-layered. Even so, the net has the ability to discern almost all of the different letters regardless of letter case. These encouraging results set the stage for a test of neural nets on actual satellite imagery which will be presented in the next section.

4.3 Tests on Satellite Imagery

In this test, the more ambitious goal of distinguishing between major cloud patterns on a satellite image will be at-

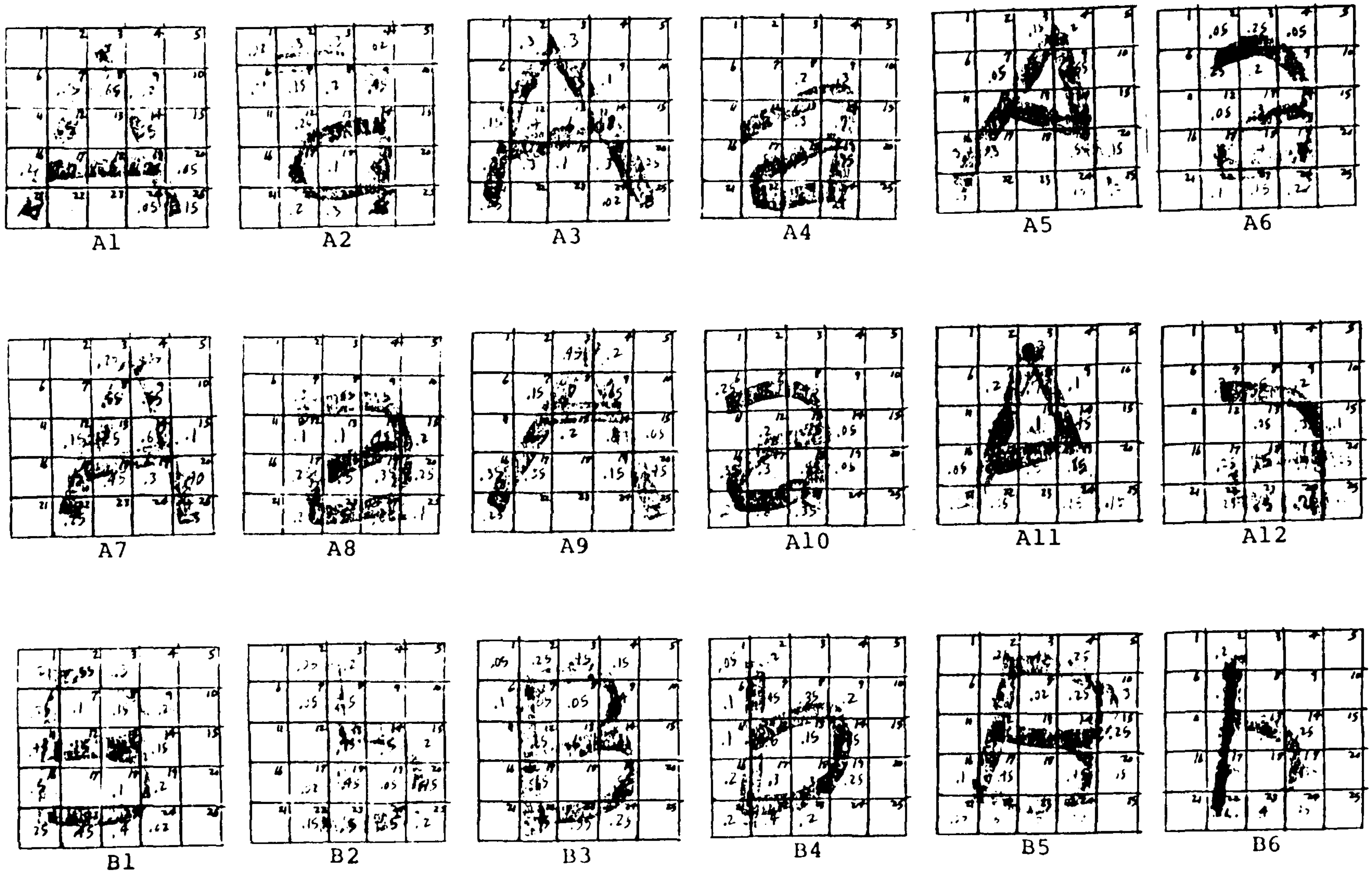

Figure 9. Dependent sample of 24 alphabetic letters (continued next page). (Labels are case numbers.)

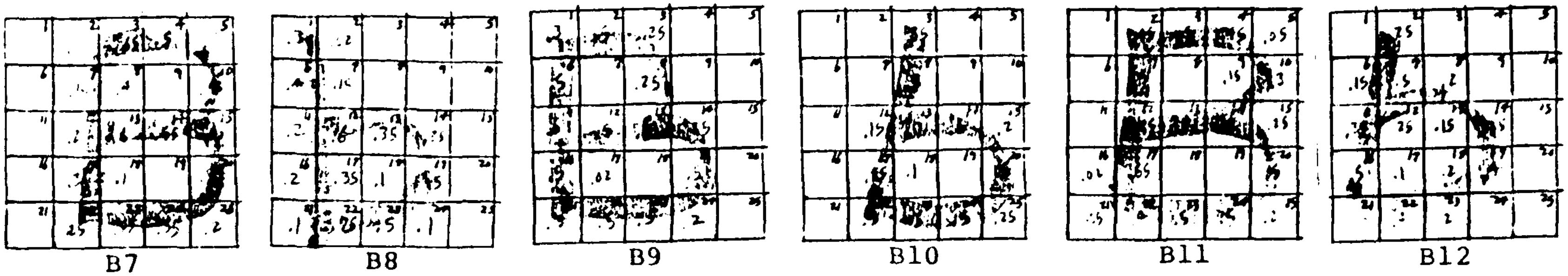

Figure 9 (continued).

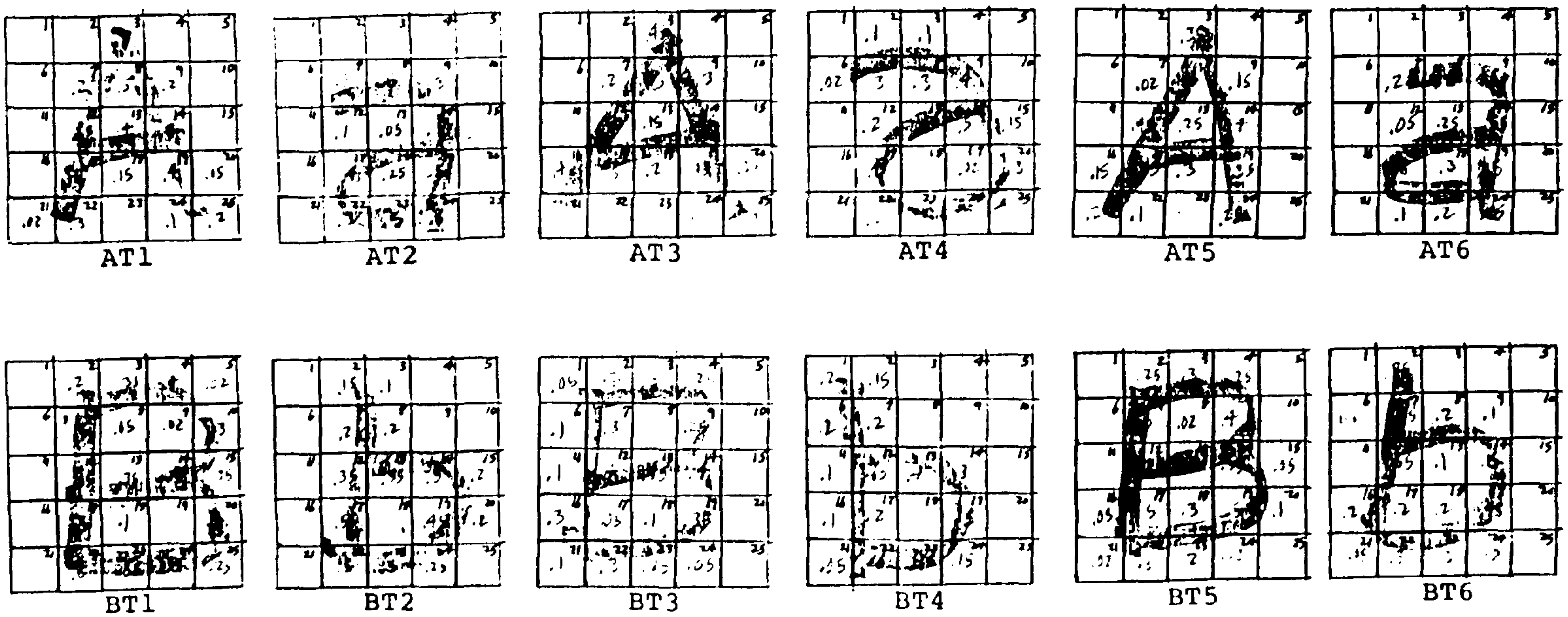

Figure 10. Independent sample of 12 alphabetic letters. (Labels are case numbers.)

Table 3. Activations of a neural net with two output nodes, and corresponding target activations, for 24 dependent sample alphabetic letters. Outputs corresponding to 'A' or 'a' are 1-0; outputs corresponding to 'B' or 'b' are 0-1.

	Output Node Activations		Target Activations	
Case	1	2	1	2
1 A1	0.99	0.00	1.00	0.00
2 B1	0.00	0.99	0.00	1.00
3 A2	0.99	0.00	1.00	0.00
4 B2	0.00	0.99	0.00	1.00
5 A3	0.99	0.01	1.00	0.00
6 B3	0.00	0.99	0.00	1.00
7 A4	0.99	0.01	1.00	0.00
8 B4	0.02	0.99	0.00	1.00
9 A5	0.99	0.00	1.00	0.00
10 B5	0.00	0.99	0.00	1.00
11 A6	0.99	0.00	1.00	0.00
12 B6	0.00	0.99	0.00	1.00
13 A7	0.99	0.00	1.00	0.00
14 B7	0.00	0.99	0.00	1.00
15 A8	0.99	0.01	1.00	0.00
16 B8	0.00	0.99	0.00	1.00
17 A9	0.99	0.00	1.00	0.00
18 B9	0.00	0.99	0.00	1.00
19 A10	0.99	0.00	1.00	0.00
20 B10	0.00	0.99	0.00	1.00
21 A11	0.99	0.00	1.00	0.00
22 B11	0.00	0.99	0.00	1.00
23 A12	0.99	0.00	1.00	0.00
24 B12	0.00	0.99	0.00	1.00

tempted. The images used will be a sequence of GOES infrared (IR) pictures presented in the Navy Tactical Applications Guide (NTAG) Vol. 4 (Fett et al., 1984). In Section 1B of that NTAG, an eastern North Pacific blocking situation is depicted. In choosing the types of cloud patterns to distinguish, it was considered desirable to include patterns of approximately the same size and that occur somewhat in isolation; i.e., the pattern is distinct and is not contiguous with an adjacent pattern. Upon examination, three major cloud patterns that meet these criteria

Table 4. As in Table 3 except for 12 independent sample cases.

	Output Node Activations		Target Activations	
Case	1	2	1	2
1 AT1	0.99	0.00	1.00	0.00
2 AT2	0.98	0.02	1.00	0.00
3 BT1	0.00	0.99	0.00	1.00
4 BT2	0.00	0.99	0.00	1.00
5 AT3	0.99	0.00	1.00	0.00
6 AT4	0.00	0.99	1.00	0.00
7 BT3	0.00	0.99	0.00	1.00
8 BT4	0.00	0.99	0.00	1.00
9 AT5	0.99	0.00	1.00	0.00
10 AT6	0.99	0.00	1.00	0.00
11 BT5	0.48	0.55	0.00	1.00
12 BT6	0.00	0.99	0.00	1.00

were determined -- frontal bands, cirrus clouds and vortices. The last category includes only those vortical clouds not associated with an accompanying frontal band; e.g., those associated with troughs or cut-off lows. In choosing cases, the description from the NTAG text was used to determine the correct feature classification.

As in the previous tests, a 5x5 input grid was used. In this case, however, a template conforming to the GOES image was defined (Fig. 11) such that each grid square was 5^{o}x5^{o}. The template was converted to acetate such that it could be overlaid on the GOES image (Fig. 12). The grid square that falls at the center of each cloud feature defines the center of a 5x5 square region. For each cloud pattern, the amount of upper (e.g., bright on IR) cloud cover in each square was visually estimated. For this network a 26th input parameter, the northernmost latitude of the grid centered on each feature, was added. A total of

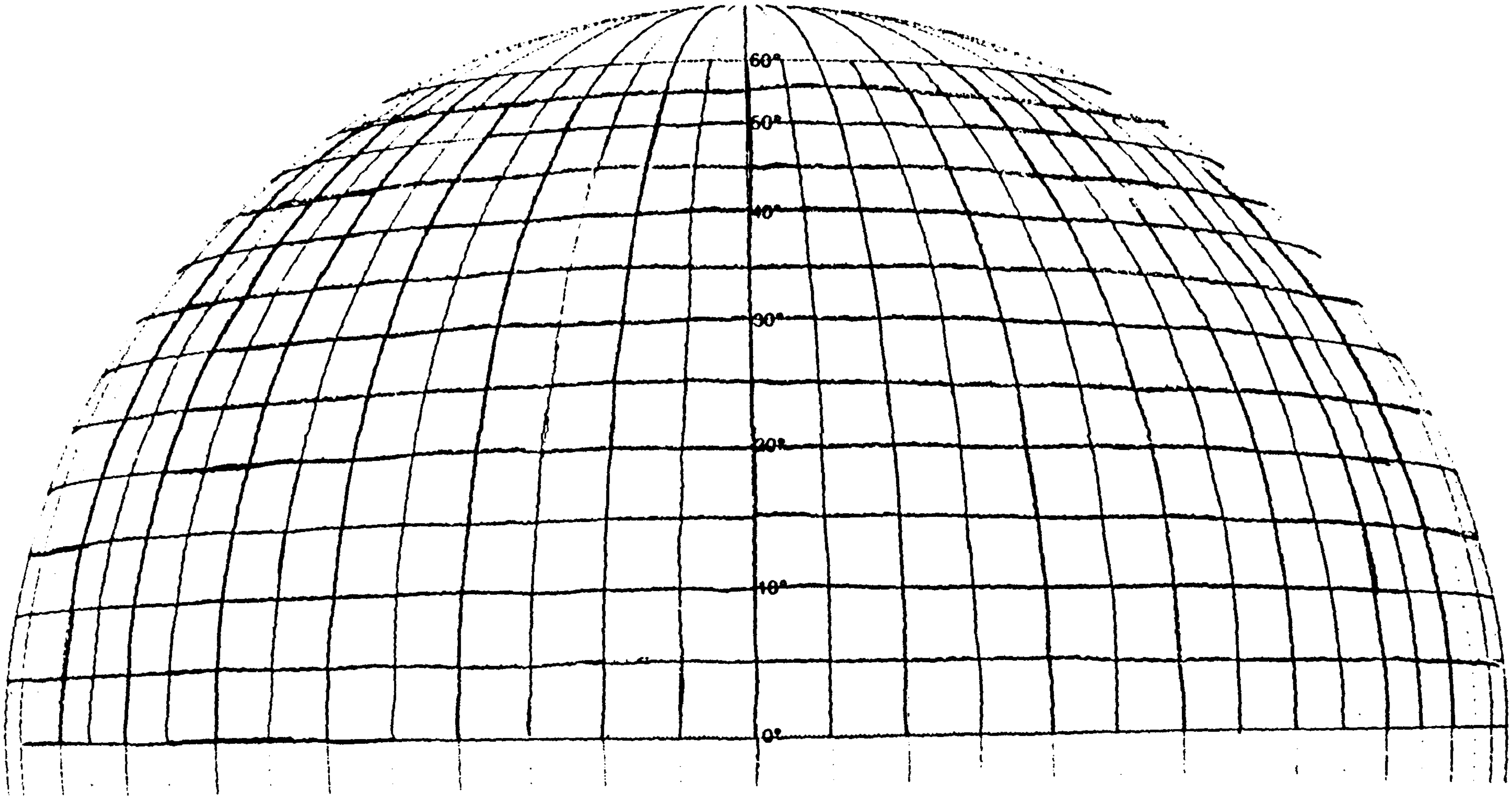

Figure 11. Template which, after transfer to acetate, forms a grid of $5^{\circ}\times5^{\circ}$ squares used to digitize GOES infrared cloud amounts.

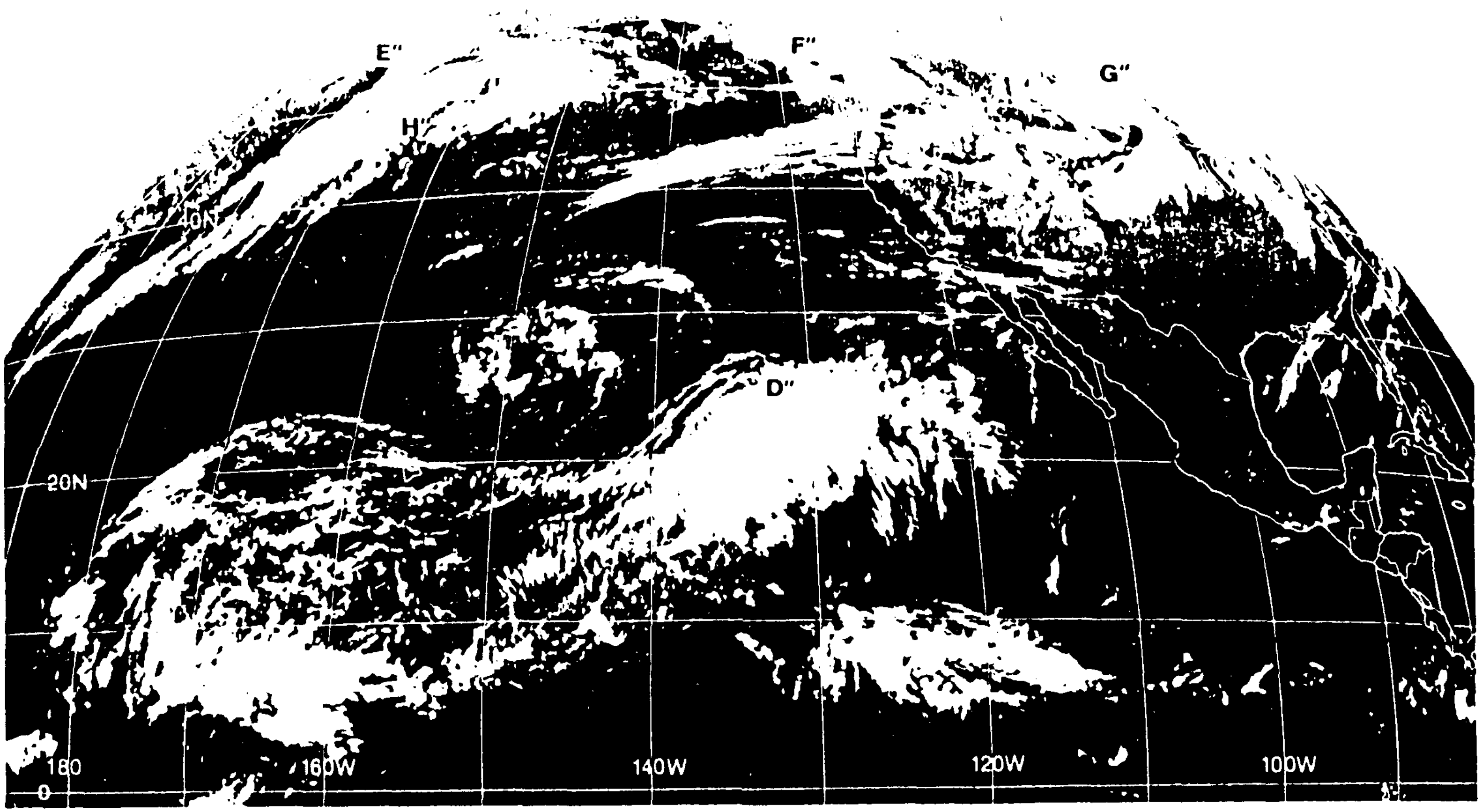

Figure 12. GOES infrared image from Fett et al. (1984), which includes several cloud features used in the neural cloud pattern recognizer (see Table 5).

19 images was used, resulting in 18 examples of each cloud pattern (Fig. 13). The images will not be reproduced here; a table of the 54 cases (Table 5) can be used in conjunction with the NTAG to determine which cloud patterns were used. In this study, every third case of each type was separated to form an independent test sample.

Again, a two-layer network was used, this time consisting of 26 inputs feeding to three hidden and three output units. Cirrus was assigned the output 1-0-0, fronts were assigned 0-1-0 and vortices were given 0-0-1. When trained on the dependent sample cases, the net was comparatively slow to converge. Several tests of 500-1000 iterations were tried with only limited success. At the time of these tests, only the Z-248 version was available. It was finally decided to initiate the program at the end of a work day and let it iterate overnight. By morning, it had completed 30,000 iterations. When this network was tested on the dependent sample, all but one of the 36 cases were correctly classified (Table 6). The one error was in classifying a front (case 29, F-14) as a vortex, although the front in case 9 (F-4) was just barely classified correctly.

On the independent sample, only limited success was expected due to the small dependent sample size, the coarse resolution of the input grid, the inexact method of determining the cloud cover amount, the use of only one radiance channel, and the goal of distinguishing somewhat similar patterns that occur in a wide variety of shapes. However, on these 18 test cases, the net was

.4	.3	.15	.15	.02
.35	.75	.75	.5	.2
.9	.85	.75	.55	.2
.85	.95	.95	.65	.25
.02	.02	.35	.15	.01

C1

.2	.15	.25	.3	.1
.15	.65	1.0	.9	.55
.45	.15	.8	.4	.2
.15	.15	.05	.05	.05
.5	[illegible]	.05	.65	.65

C2

.05	0	.35	.55	.05
.45	.75	.9	.35	.15
.98	.98	.95	.8	.35
.35	.9	.2	.15	.2
.2	.1	.02	.01	.01

C3-I

0	.15	.6	.98	.95
.4	.6	.7	.5	.9
.4	.2	.35	.25	.05
			.02	
.1				

C4

.4	.65	.15	.05	
.02	.45	.7	.75	.75
.45	.9	.85	.75	.55
.3	.5	.5	.65	.15
.02	.02	.05		.05

C5

			.05	.1
	.2	.85	.85	.85
.02	.7	.99	.35	.02
.6	.55	.5	.02	
.1	.05			

C6-I

	.05	.02	.25	.45
.2	.65	.9	.8	.3
.75	1.0	.65	.2	
.95	.95	.2	.05	
.3	.4			.2

C7

	.5	.7	.35	.5
	.65	.70	.15	
.15	.8	.75	.3	.3
.6	.85	.3	.12	.02
.25	.25	.1		

C8

			.2	.65
		.6	.95	.99
	.7	1.0	.6	.6
.15	.45	.65	.35	.1
			.05	.2

C9-I

	.45	.25	.4	.7
	.05	.6	1.0	.9
.05	.75	1.0	.95	.85
.5	.95	.75	.8	.25
.3	.8	.75	.35	

C10

.05	.9	.55	.05	
.45	1.0	.9	.2	.2
.65	.9	.85	.75	.65
.99	.75	.8	.65	.35
.45	.45	.4	.3	

C11

	.75	.95	.6	.55
	.7	.95	.55	.65
.02	.75	1.0	.95	.6
.35	.9	.65	.4	.4
.6	.4	.2	.2	.25

C12-I

.7	1.0	.98	.55	
.5	.95	.98	.4	.15
.2	.8	.7	.75	.3
.3	.65	.5	.45	.3
.3	.3	.3	.25	05

C13

	.45	.8	.8	.35
.02	.15	.7	.6	.3
.15	.5	.9	.4	.1
	.5	.65	.25	.15
.15	.2	.2		

C14

.15	.35	.55	.95	.65
	.2	.6	.7	.45
	.35	.85	.3	.1
.02	.6	.35	.15	.2
.25	.2			.02

C15-I

.4	.85	.8	.75	.95
.1	.4	.85	.85	.75
.05	.15	.95	.9	.35
.05	.35	.35	.25	
.1	.05	.02		.05

C16

.45	.45	.5	.35	.25
.35	.05	.95	.9	.1
	.35	.8	.35	.3
	.25	.05		.05
			.02	

C17

.25	.4	.75		.45
.15	.1	.7	.35	.95
.6	.6	.65	.65	.45

C18-1

Figure 13a. 18 digitized cirrus GOES cases. Label beneath each grid is the NTAG reference (see Table 5). Cases marked "-I" form the independent test sample.

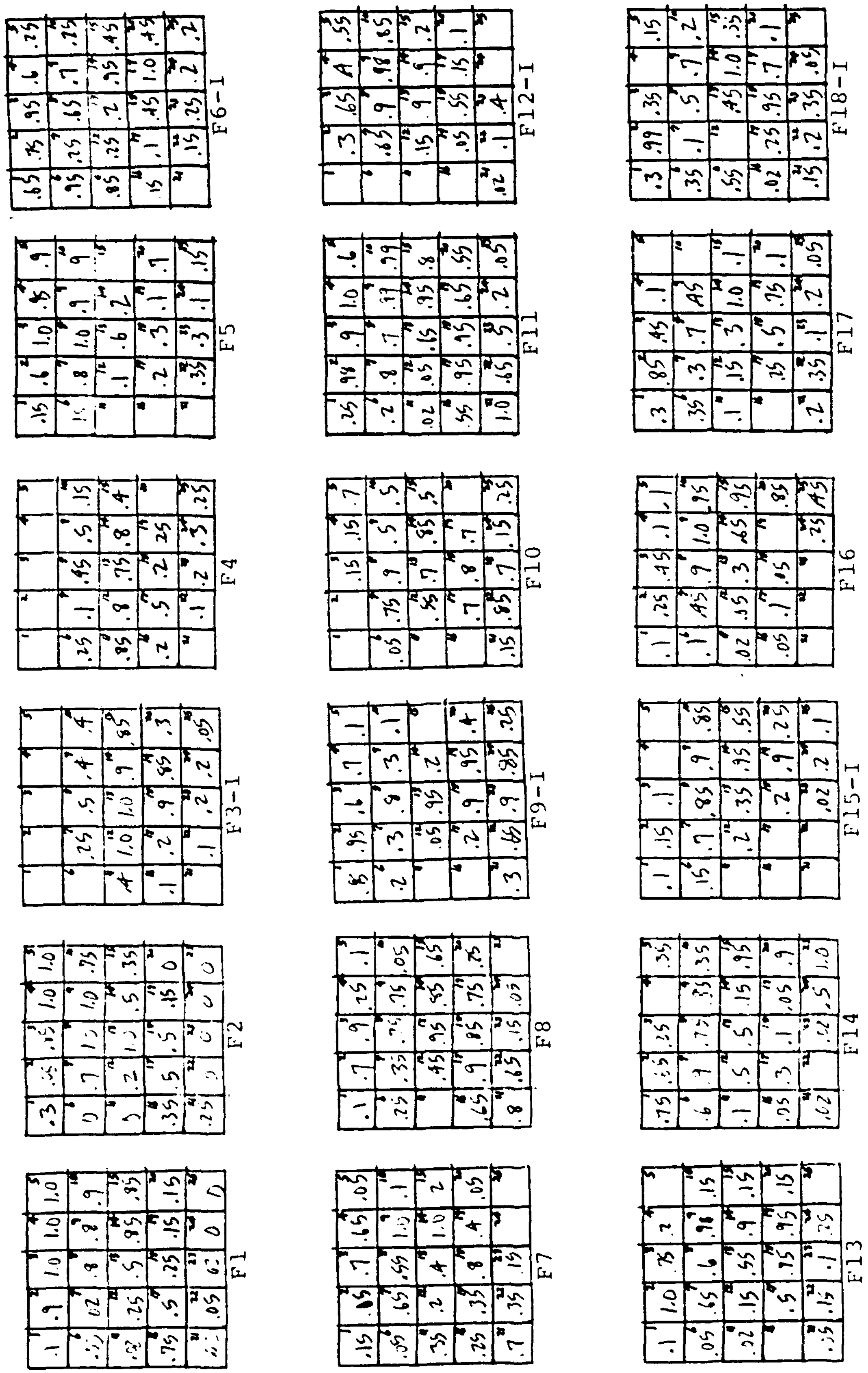

Figure 13b. As in Fig. 13a except for 18 digitized GOES fronts.

.02	.05	.05	.05	.02
0	0	.1	.25	.25
0	.5	.4	.35	.15
.15	.02	.02	.15	.65
.15	.1	.2	.5	.9

V1

.25				
	.02	.2	.4	.05
	.15	.3	.75	.3
		.1	.6	.95
.1	.5	.55	.75	.4

V2

.1	.02			
.15	.7	.8	.65	.1
.1	.2	.45	.75	.1
		.02	.4	.6
		.3	.9	.9

V3-I

	.05	.02		
.35	.5	.65	.6	.15
.25	.65	.75	.95	.5
.45	.35	.35	.35	.6
.15	.02	.2	.45	.3

V4

.2	.2	.5	.4	.12
.65	.35	.2	.25	.7
.2	.35	.75	.35	.1
	.35	.4	.4	.3
		.35	.7	.4

V5

.15	.25	.15	.6	.85
	.25	.75	.25	.25
	.4	.15	.8	.85
	.05	.6	.35	.5
		.05	.1	1.0

V6-I

.95	.15	.05	.5	.85
.85	.05	.75	.75	.9
.2	.05	.25	.35	.85
.15	.02	.05	.1	.85
			.4	1.0

V7

.85	.8	.25	.2	.95
.5	.3	.65	.1	.5
		.95	.6	.7
		.35	.35	.15
		.1	.75	.75

V8

.45	.9	.4		.05
.75	.1	.15	.85	1.0
.65	.05	.4	.9	.85
.1		.15	.05	.45
			.02	.75

V9-I

.35	.85	.65	.3	.6
.2	.25	.35	.5	.15
.05	.02	.14	.6	.2
.1			.25	.05

V10

.98	.7	.25	.35	.45
.45	.4	.8	.3	.05
.8	.1	.4	.45	.55
		.4	.50	.25
	.05	.15	.2	

V11

.05	.15	.2		.4
.55	.95	1.0	.35	
.3	.85	.95	.45	
.3	.4	.05		
	.05			

V12-I

.4	.65	.15	.7	.55
	.55	.35	.35	.75
	.05	.35	.7	.25
	.1	.25	.35	.35
			.05	.15

V13

.2	.35	.45	.65	.6
	.5	.65	.2	.15
	.15	.8	.15	.2
	.05	.1	.65	.3
		.05		

V14

.15	.15	.3	.1	.5
.05	.6	.4	.5	.05
.02	.45	.9	.55	.15
	.25	.35	.65	.15
		.1	.1	

V15-I

.25	.7	.3	.1	.5
.25	1.0	.6	.6	.5
.1	.15	.7	.1	.35
.15	.25	.35	.4	.35
	.1	.2	.2	.1

V16

.25	.75	.6	.2	.9
1.0	.6	.5	.65	.65
.1	.9	.85	.3	.65
		.15	.35	.35
.65		.2	.3	.65

V17

.95	.8	.95	.6	.6
.2	.35	.65	.3	.5
	.3	.25	.7	.25
.1		.15	.25	.35
.65	.2	.05	.25	.35

V18-I

Figure 13c. As in Fig. 13a except for 18 digitized GOES cloud vortices.

Table 5. 54 cirrus, front and vortex cloud patterns taken from NTAG Vol. 4 (Fett, et al., 1984) for training and testing a neural net. Labels used in NTAG are included for reference, as well as latitude and longitude of grid center used in digitizing the cloud amounts.

NTAG			
Figure	Cirrus	Front	Vortex
1B3a	C" (12.5N,162.5W)	F" (47.5N,127.5W)	A" (27.5N,147.5W)
	D" (17.5N,132.5W)	H" (47.5N,172.5W)	
1B7a	jet(22.5N,127.5W)	I" (42.5N,172.5W)	A" (27.5N,147.5W)
	jet(12.5N,142.5W)	F" (42.5N,97.5W)	
1B11a	jet(22.5N,132.5W)	I" (42.5N,157.5W)	A" (32.5N,142.5W)
	jet(12.5N,157.5W)		
1B15a	PJS(22.5N,142.5W)	J" (42.5N,172.5W)	K" (42.5N,137.5W)
1B19a		J"1(42.5N,157.5W)	Q" (42.5N,172.5W)
		P" (27.5N,127.5W)	
1B23a	jet(17.5N,117.5W)	Q" (47.5N,152.5W)	
		P" (27.5N,117.5W)	
1B27a	jet(17.5N,152.5W)	S" (27.5N,142.5W)	
		P" (27.5N,117.5W)	
1B30a	jet(22.5N,142.5W)		S" (37.5N,147.5W)
1B31a	jet(22.5N,132.5W)		S" (37.5N,147.5W)
			U" (42.5N,172.5W)
1B31c	jet(22.5N,132.5W)		U" (42.5N,167.5W)
1B32a	jet(27.5N,127.5W)		U" (37.5N,157.5W)
1B33a	jet(12.5N,127.5W)		U" (37.5N,152.5W)
1B35a	jet(22.5N,122.5W)		
1B38a	jet(27.5N,117.5W)		V" (37.5N,172.5W)
			U"3(37.5N,152.5W)
1B39a	jet(27.5N,112.5W)	V" (37.5N,167.5W)	U"3(32.5N,142.5W)
1B39c		V" (42.5N,167.5W)	X" (32.5N,132.5W)
1B40a		X" (32.5N,127.5W)	V" (47.5N,172.5W)
		V"1(37.5N,152.5W)	
1B41a		X" (32.5N,122.5W)	V" (47.5N,172.5W)
		V"1(37.5N,147.5W)	
1B43a	Y"3(27.5N,152.5W)		V" (47.5N,172.5W)

able to correctly classify every case (Table 7). The likely reason for this surprising success is that the cases are selected from a time sequence of images, so that there is some similarity between instances of a pattern on images separated by a few hours. Nevertheless, these results are a very encouraging indi-

Table 6. Activations of a neural net with three output nodes, and corresponding target activations, for 36 dependent sample cloud patterns. Outputs corresponding to cirrus, fronts and vortices are 1-0-0, 0-1-0 and 0-0-1, respectively.

	Output Node Activations			Target Activations		
Case	1	2	3	1	2	3
1 C1	0.99	0.06	0.00	1.00	0.00	0.00
2 F1	0.00	0.99	0.00	0.00	1.00	0.00
3 V1	0.00	0.03	0.99	0.00	0.00	1.00
4 C2	0.99	0.06	0.00	1.00	0.00	0.00
5 F2	0.00	0.99	0.00	0.00	1.00	0.00
6 V2	0.00	0.03	0.99	0.00	0.00	1.00
7 C4	0.99	0.06	0.00	1.00	0.00	0.00
8 F4	0.03	0.05	0.00	0.00	1.00	0.00
9 V4	0.00	0.03	0.99	0.00	0.00	1.00
10 C5	0.99	0.06	0.00	1.00	0.00	0.00
11 F5	0.00	0.99	0.00	0.00	1.00	0.00
12 V5	0.00	0.03	0.99	0.00	0.00	1.00
13 C7	0.99	0.06	0.00	1.00	0.00	0.00
14 F7	0.00	0.99	0.00	0.00	1.00	0.00
15 V7	0.00	0.03	0.99	0.00	0.00	1.00
16 C8	0.99	0.06	0.00	1.00	0.00	0.00
17 F8	0.00	0.99	0.00	0.00	1.00	0.00
18 V8	0.00	0.03	0.99	0.00	0.00	1.00
19 C10	0.99	0.06	0.00	1.00	0.00	0.00
20 F10	0.00	0.99	0.00	0.00	1.00	0.00
21 V10	0.00	0.03	0.99	0.00	0.00	1.00
22 C11	0.99	0.06	0.00	1.00	0.00	0.00
23 F11	0.00	0.99	0.00	0.00	1.00	0.00
24 V11	0.00	0.03	0.99	0.00	0.00	1.00
25 C13	0.99	0.06	0.00	1.00	0.00	0.00
26 F13	0.00	0.99	0.00	0.00	1.00	0.00
27 V13	0.00	0.03	0.99	0.00	0.00	1.00
28 C14	0.99	0.06	0.00	1.00	0.00	0.00
29 F14	0.00	0.03	0.99	0.00	1.00	0.00
30 V14	0.00	0.03	0.99	0.00	0.00	1.00
31 C16	0.99	0.06	0.00	1.00	0.00	0.00
32 F16	0.00	0.99	0.31	0.00	1.00	0.00
33 V16	0.00	0.03	0.99	0.00	0.00	1.00
34 C17	0.92	0.05	0.00	1.00	0.00	0.00
35 F17	0.00	0.99	0.00	0.00	1.00	0.00
36 V17	0.00	0.03	0.99	0.00	0.00	1.00

cation that the neural pattern recognition approach can be applied to the cloud pattern identification problem.

Table 7. As in Table 6 except for 18 independent sample cases.

	Output Node Activations			Target Activations		
Case	1	2	3	1	2	3
1 C3-I	0.99	0.07	0.00	1.00	0.00	0.00
2 F3-I	0.00	0.99	0.00	0.00	1.00	0.00
3 V3-I	0.00	0.03	0.99	0.00	0.00	1.00
4 C6-I	0.99	0.06	0.00	1.00	0.00	0.00
5 F6-I	0.00	0.99	0.00	0.00	1.00	0.00
6 V6-I	0.00	0.03	0.99	0.00	0.00	1.00
7 C9-I	0.99	0.06	0.00	1.00	0.00	0.00
8 F9-I	0.00	0.99	0.00	0.00	1.00	0.00
9 V9-I	0.00	0.03	0.99	0.00	0.00	1.00
10 C12-I	0.99	0.06	0.00	1.00	0.00	0.00
11 F12-I	0.00	0.99	0.00	0.00	1.00	0.00
12 V12-I	0.00	0.03	0.99	0.00	0.00	1.00
13 C15-I	0.99	0.06	0.00	1.00	0.00	0.00
14 F15-I	0.00	0.99	0.00	0.00	1.00	0.00
15 V15-I	0.00	0.03	0.99	0.00	0.00	1.00
16 C18-I	0.96	0.06	0.00	1.00	0.00	0.00
17 F18-I	0.00	0.99	0.00	0.00	1.00	0.00
18 V18-I	0.00	0.03	0.99	0.00	0.00	1.00

5. Proposed Cloud Image Feature Recognition System

These results are a very encouraging indication that neural cloud pattern recognition is indeed possible. This study, however, was designed only to prove the concept, not to provide an operational pattern recognition capacity. The types of patterns distinguished here are quite restrictive compared to the range of patterns seen on the day-to-day imagery. There are cloud features of both large and small scales, and instances where features may be contiguous or overlap. These problems must be solved before neural cloud image interpretation is possible.

An additional problem that would be crucial to automated, real-time image interpretation is the huge amount of pixel data in each image. For each pixel value to be processed, even if only to be accessed as an input to a neural net, would likely require an unacceptably large amount of computer resources if done sequentially, or expensive hardware if done in parallel.

Based on the results of this study, a potential processing architecture for automated cloud feature pattern recognition will now be presented. The envisioned system is called the Cloud Image Feature Recognition System, or CIFRS (pronounced "cy-fers").

As shown in this study, large-scale patterns can be recognized based on fairly coarse input data from the image. Thus, it may not be necessary to process data from every pixel, but rather from every Nth pixel. The acceptable value of N could probably be determined statistically from information about how much of such a scene is cloudy and the total number of pixels it contains. Given such a spot-sampling of the image, cloudy vs. non-cloudy regions could be determined. Because of the robustness of the neural interpretation methods, this initial cloud/no-cloud determination could be from a simple radiance threshold method. Alternately, a neural net using inputs from many channels might prove superior for this task. Adjacent cloudy pixels would then be considered to define a continuously-cloudy region. Again, the validity of this assumption depends on the number of pixels omitted from the sample. Even if multiple features are erroneously combined at this step, a later process will separate them.

Once the cloudy regions have been determined, a large-scale pattern-recognition neural net could be used to provide an initial classification. Such a net would likely be quite similar to the one presented in the previous section. The net would have to be trained on a data set that includes any large-scale cloud features, including those apparently large-scale features that are in reality contiguous smaller-scale features. Consider, however, that the image area that must be analyzed has been reduced by the omission of predominantly noncloudy regions. In addition, it seems likely that the analysis of a large, cloudy region would still not require the processing of every pixel in the region. Thus, the CIFRS approach would establish a significant reduction in the amount of data to be processed.

Given a large-scale identification of cloudy regions, a logic-based system would be used to expand the analysis to include smaller-scale features. An example of such a system is the SIAMES expert system (Peak, 1989). SIAMES uses information about the known feature classifications to initiate a search for associated smaller-scale features that might be present. For example, once SIAMES knows there is a front, it looks for open cells behind the front. This top-down approach accomplishes a further reduction of the data because specific locations are chosen for the search. In CIFRS, a neural net would be trained for analysis of each such subregion. For example, a post-frontal neural net would look for open or closed cells, small-scale vortices, etc. An additional task at this stage would be the analysis of details of the large-scale features. Examples might be the sharpness of a frontal band edge or the presence of transverse bands in cirrus.

The envisioned architecture is presented in Figure 14. The pixel data is first reduced by some statistical method in the Data Reducer. This reduced data set forms the input to the Cloud Determiner. This module might attempt a cloud/no-cloud categorization, or possibly a cloud classification such as does the neural network methodology of Lee et al. (1990). The output of this analysis step becomes the input to the Large Feature Identifier. This module would be very similar to that presented in this study. However, the system should be designed to have its results checked by the next module, the Analysis Supervisor. Since neural nets never have to stop learning, if subsequent analysis proves that a large-feature classification was wrong, that network could learn from its mistake.

The Analysis Supervisor would be responsible for overseeing the more detailed analysis. Like SIAMES, it would use contextual information and preliminary conclusions to generate hypotheses about the remaining image features and also generate more detailed analysis of the identified features. It is unclear whether the Supervisor would be a knowledge-based system or a neural net. To accomplish the verification of these hypotheses, a suite of Small Feature Identifiers would be available. The Supervisor would determine which to use and at what locations. Based on the growing knowledge provided by the continuing analysis, the Supervisor might generate further uses for the Small Feature Identifi-

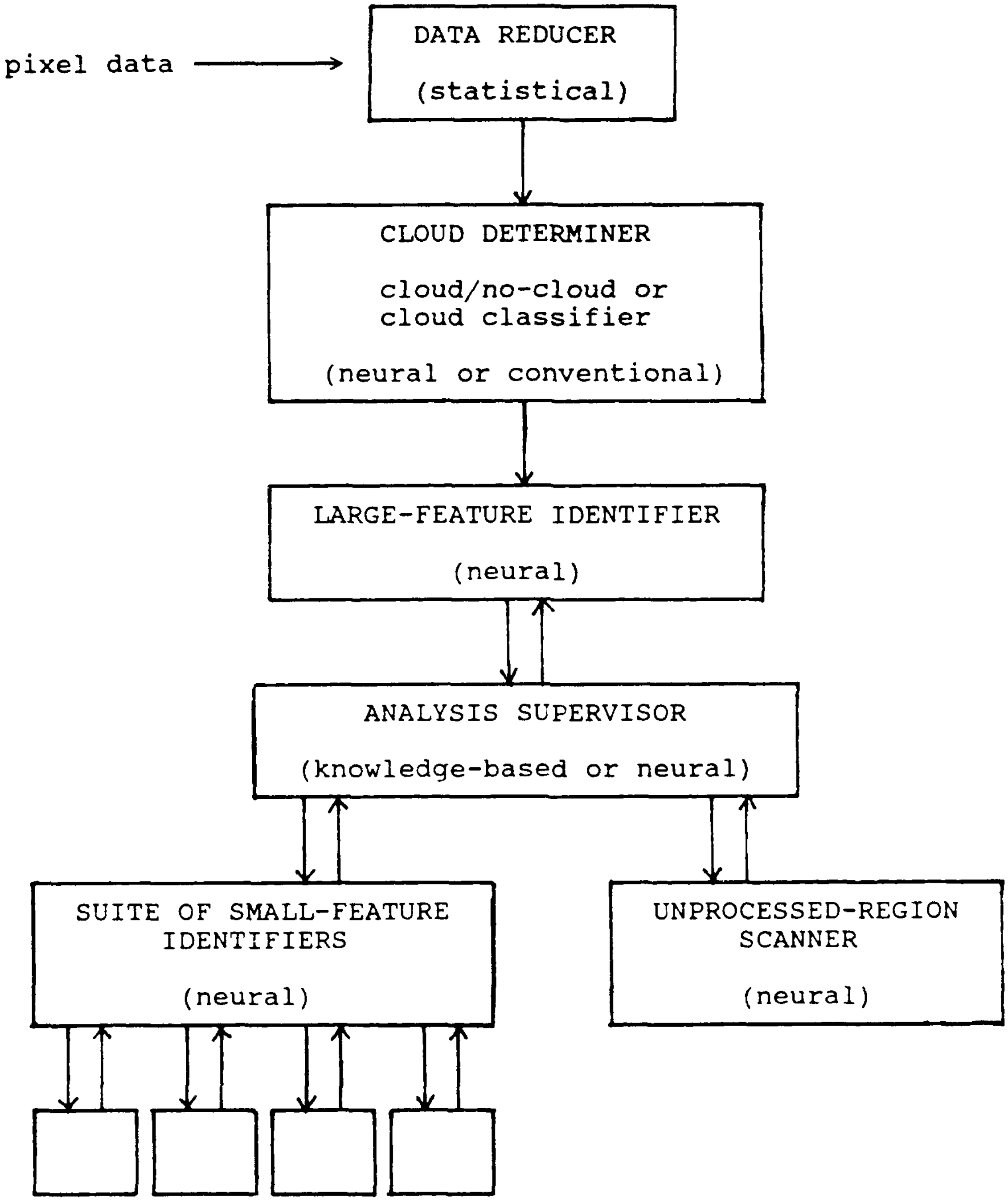

Figure 14. Proposed CIFRS architecture.

ers, or might alter its earlier conclusions. In particular, large-scale features that are actually connected smaller features would be identified as such.

Eventually, the goal is to provide a meteorological interpretation of the image features. At this early stage, it is somewhat difficult to determine how much of the CIFRS processing will proceed top-down or bottom-up. It seems likely that data will be processed in both directions repetitively; as more features are identified, the increased information will generate a search for more detail about what is known and for new features not yet recognized. The CIFRS process appears to be similar to the way humans interpret an image. We certainly do not begin by analyzing every tiny detail -- rather, a few large features catch our attention first. Based on what these features might be, we generate in out minds hypotheses about the overall image and begin to search for evidence to verify or disprove those hypotheses. Thus, we only examine in detail the portions of the image that give us the information that we need. Granted, sometimes scanning an area that we have no particular interest in will uncover an interesting feature. After the CIFRS knowledge-based interpretation is complete, the system could then include such scanning of areas not yet processed using neural networks (Fig. 14). The time and detail devoted to this search would depend on the computer resources available, but since much of the image will already be interpreted the scanning should not have to cover a large area.

The CIFRS system would involve many neural nets tailored toward different tasks. The research involved in constructing such a complete system is extensive. However, the results of the initial tests presented here provide a good starting point in developing CIFRS. The first task should probably be to develop a more sophisticated neural identifier of large features. If possible, actual satellite image data should be used. Thus, the problem of determining how much pixel data to ignore should also be addressed. Once the large-scale features are being recognized, the system can be expanded to include more and more detailed analysis.

6. Conclusions and Recommendations

Neural networks are a powerful type of computer model in which many nonlinear processing elements are arranged in parallel networks. These networks, based on current understanding of biological nervous systems, have proven useful in pattern recognition problems.

In this paper, six neural networks that perform classification have been investigated. The multi-level perceptron network has been chosen to be most applicable to the classification of satellite cloud images. The mathematical basis for this neural network model has been presented and a version of the model, called "NOARL-NEURAL," has been programmed.

In simple tests on geometric and alphabetic shapes, the multi-layer perceptron neural nets were shown to have high levels of pattern classification skill. A more complex test using GOES infrared imagery showed that a neural network can be used to distinguish between fronts, cirrus and cloud vortices. Of the 54 cloud patterns in this test, only one dependent sample pattern was incorrectly classified.

The surprisingly good results of this limited experiment have led to a proposed architecture for automated cloud feature recognition. The system, called CIFRS (Cloud Image Feature Recognition System), would involve several neural networks tailored to the interpretation of different types of cloud features. These network types would be based on feature size and location relative to other known features. The complete system would likely include statistical and knowledge-based components to accomplish data reduction and to direct the search process. The proposed architecture includes both top-down and bottom-up processing such that the problem is reduced to a manageable size. The continuous feedback between the CIFRS modules should make the system less sensitive to classification errors made during any one step in the process.

The next step in the process should be to refine the large-scale feature identification experiment presented in this study. It should be attempted to derive a network that classifies a wider range of large-scale features. Possible inputs modifications would include different resolutions, multiple-channel information and cloud classifications. Since CIFRS requires a skillful large-feature classification, the development of this module is a natural first step.

REFERENCES

Fett, R.W., W.A. Bohan and J. Rosenthal, 1984: Navy Tactical Applications Guide, Vol. 4, Eastern North Pacific Weather Analysis and Forecast Applications; Naval Environmental Prediction Research Facility* technical report TR 83-01. 180 pp.

Lee, J., R.C. Weger, S.K. Sengupta and R.M. Welch, 1990: A neural network approach to cloud classification. Unpublished manuscript.

Lippman, R.P., 1987: An introduction to computing with neural nets. IEEE ASSP Mag., April, p.4-22.

Peak, J.E., 1989: Initial expert evaluation and resultant modifications to the satellite image analysis meteorological expert system (SIAMES) prototype. Naval Oceanographic and Atmospheric Research Laboratory, Atmospheric Directorate, Monterey, CA 93943-5006. 12 pp plus appendices. Unpublished internal report.

Rumelhart, D.E., G.E. Hinton and R.J. Williams, 1986: Learning internal representations by error propagation; in Parallel Distributed Processing - Explorations in the Microstructure of Cognition. MIT Press.

Widrow, B. and M.E. Hoff, 1960: Adaptive switching circuits. IRE WESCON Conv. Record, Part 4. pp 96-104.

*Now the Naval Oceanographic and Atmospheric Research Laboratory, Atmospheric Directorate, Monterey, CA 93943-5006.

DISTRIBUTION LIST

SPAWARSYSCOM
ATTN: PMW-141
WASHINGTON, DC 20363-5100

NOARL (10)
ATTN: CODE 125L
JCSSC, MS 39529-5004

NOARL
ATTN: CODE 125P
JCSSC, MS 39529-5004

NOARL
ATTN: CODE 300
JCSSC, MS 39529-5004

OFFICE OF NAVAL RESEARCH
ATTN: CODE 10
800 N. QUINCY ST.
ARLINGTON, VA 22217-5000

WOODS HOLE OCEANOGRAPHIC INST.
ATTN: AI APPLICATIONS
P.O. BOX 32
WOODS HOLE, MA 02543

SCRIPOPS INST. OF OCEANOGRAPHY
ATTN: AI APPLICATIONS
BOX 6049
SAN DIEGO, CA 92106

OFFICE OF NAVAL TECHNOLOGY
ATTN: DR. P. SELWYN, CODE 20
800 N. QUINCY ST.
ARLINGTON, VA 22217-5000

OFFICE OF NAVAL TECHNOLOGY
DR. M. BRISCOE, CODE 228
800 N. QUINCY ST.
ARLINGTON, VA 22217-5000

OFFICE OF NAVAL RESEARCH
ATTN: CODE 12
800 N. QUINCY ST.
ARLINGTON, VA 22217-5000

OFFICE OF NAVAL RESEARCH
ATTN: HEAD, OCEAN SCIENCES DIV
CODE 1122
ARLINGTON, VA 22217-5000

U.S. NAVAL ACADEMY
ATTN: LIBRARY REPORTS
ANNAPOLIS, MD 21402

U.S. NAVAL ACADEMY
ATTN: OCEANOGRAPHY DEPT.
ANNAPOLIS, MD 21402

NAVAL POSTGRADUATE SCHOOL
ATTN: CODE MR
MONTEREY, CA 93943-5000

NAVAL POSTGRADUATE SCHOOL
ATTN: 0142
MONTEREY, CA 93943-5002

SPAWARSYSCOM
ATTN: CODE 312
NAT. CTR. #1
WASHINGTON, DC 20363-5100

SPAWARSYSCOM
ATTN: CODE PMW-141
NAT. CTR. #1
WASHINGTON, DC 20363-5100

NAVOCEANSYSCEN
ATTN: J. RICHTER, CODE 54
SAN DIEGO, CA 92152-5000

PACMISTESTCEN
ATTN: GEOPHYSICS OFFICER
PT. MUGU, CA 93042

AFGWC/DAPL
ATTN: TECH. LIBRARY
OFFUTT AFB, NE 68113

AFGL/LY
ATTN: MET. OFFICER
HANSCOM AFB, MA 01731

COMMANDER/DIRECTOR
ASL, WHITE SANDS
ATTN: SLCAS-AE
WSMR, NM 88002-5501

NOAA-NESDIS LIAISON
ATTN: CODE SC2
NASA-JOHNSON SPACE CENTER
HOUSTON, TX 77058

DIRECTOR
NATIONAL EARTH SAT. SERV/SEL
ATTN: FB-4, S321B
SUITLAND, MD 20233

FEDERAL COORD. FOR METEORO.
SERVS. & SUP. RSCH. (OFCM)
ATTN: OPERATIONS SECTION
11426 ROCKVILLE PIKE
ROCKVILLE, MD 20852

DIRECTOR, NOAA
ATTN: TECH. DEVELOPMENT LAB
GRAMAX BLDG.
8060 13TH ST.
SILVER SPRING, MD 20910

NCAR
ATTN: LIBRARY ACQUISITIONS
P.O. BOX 3000
BOULDER, CO 80307

NATIONAL SCIENCE FOUNDATION
ATTN: ATMOS. SCIENCES DIV.
1800 G STREET, NW
WASHINGTON, DC 20550

SCRIPPS INSTITUTION OF
OCEANOGRAPHY, LIBRARY
ATTN: DOCUMENTS/REPORTS SECT.
LA JOLLA, CA 92037

WOODS HOLE OCEANO. INST.
ATTN: DOCUMENT LIBRARY LO-206
WOODS HOLE, MA 02543

COLORADO STATE UNIVERSITY
ATTN: ATMOS. SCIENCES DEPT
ATTN: LIBRARIAN
FT. COLLINS, CO 80523

UNIVERSITY OF WASHINGTON
ATTN: ATMOS. SCIENCES DEPT.
SEATTLE, WA 98195

PENNSYLVANIA STATE UNIV.
ATTN: METEOROLOGY DEPT.
503 DEIKE BLDG.
UNIVERSITY PARK, PA 16802

AMERICAN METEORO. SOCIETY
ATTN: METEOR. & GEOASTRO. ABST
P.O. BOX 1736
WASHINGTON, DC 20013

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. Agency Use Only *(Leave blank).*

2. Report Date.
March 1991

3. Report Type and Dates Covered.
Final

4. Title and Subtitle.
Neural Network Methodologies and their Potential Application to Cloud Pattern Recognition

5. Funding Numbers.
Program Element No. 62435N
Project No. RM35G82
Task No. 4
Accession No. DN651750

6. Author(s).
J.E. Peak

7. Performing Organization Name(s) and Address(es).
Computer Sciences Corporation, Monterey, CA 93943-5006

Naval Oceanographic and Atmospheric Research Laboratory
Atmospheric Directorate, Attn Dr. P. Tag
Monterey, CA 93943-5006

8. Performing Organization Report Number.
NOARL Technical Note 103

9. Sponsoring/Monitoring Agency Name(s) and Address(es).
Office of Naval Research (ONT)
800 N. Quincy St.
Arlington, VA 22217-5000

10. Sponsoring/Monitoring Agency Report Number.
NOARL Technical Note 103

11. Supplementary Notes.

12a. Distribution/Availability Statement.
Approved for public release; distribution is unlimited.

12b. Distribution Code.

13. Abstract *(Maximum 200 words).*

Six artificial neural network models are explored as potential methodologies for the automated interpretation of satellite cloud images. The Multi-layer Perceptron Network is chosen to be the most applicable to the image interpretation problem. A complete, mathematical description of the methodology is presented. The neural network's classification capability is demonstrated using simple geometric patterns and alphabetic characters. A more complex test using GOES infrared imagery shows that the neural network can distinguish 53 of 54 large-scale cloud patterns. An architecture for a complete, automated cloud feature recognition system is proposed.

14. Subject Terms.
Neural network
Artificial intelligence
Cloud recognition
Satellite imagery

15. Number of Pages.
52

16. Price Code.

17. Security Classification of Report.
UNCLASSIFIED

18. Security Classification of This Page.
UNCLASSIFIED

19. Security Classification of Abstract.
UNCLASSIFIED

20. Limitation of Abstract.
Same as report