ELSEVIER

Neurocomputing 7 (1995) 159–185

NEUROCOMPUTING

Image Recognition Neural Network: IRNN

Krzysztof J. Cios *, Inho Shin

Department of Electrical Engineering, University of Toledo, Toledo, OH 43606, USA

Received 11 February 1993; accepted 9 November 1993

Abstract

Artificial neural network models are becoming very attractive in image processing where high computational performance and parallel architectures are required. Recently, many papers appeared on applications of neural networks to problems where some degree of intelligence or human-like performance is desired. This paper describes a novel neural network architecture for image recognition and classification. The proposed neural network, called an image recognition neural network (IRNN), is designed to recognize an object or to estimate an attribute of an object. IRNN takes an analog gray level image as an input and produces an appropriate recognition code at the output.

Keywords: Unsupervised learning; Supervised learning; Image recognition

1. Introduction

The basic idea behind the proposed image recognition neural network (IRNN) is related to the work of Fukushima [8–10], whose work, in turn, was based on a hierarchical model of the visual system proposed by Hubel and Wiesel [12].

IRNN consists of a bottom layer, a top layer, and one or more intermediate layers depending on the complexity of the input images. The bottom layer extracts local features from the images and the top layer produces recognition codes. The role of an intermediate layer(s) is to aggregate local features extracted by the bottom layer and generate semi-global features. The top layer associates the semi-global features with the appropriate recognition codes.

The images considered here are constrained to a single object which can vary in size, shape, and intensity. Example is an image of a wheel. Images containing the

* Corresponding author. Email: fac1765@uoft01.utoledo.edu

SSDI 0925-2312(93)E0062-I

same object but differing in some attributes (like number of holes in a wheel) will be referred to as a set of images. Then, the problem is to detect these attributes. We shall deal here only with time invariant, stationary input images.

A number of neuron models have been described in the literature [7,22,1,4]. The artificial neuron consists of a summing junction, an activation function and a number of inputs with variable weights. The x_i inputs correspond to the frequency of incoming pulses in a biological neuron.

The w_i weights correspond to the strengths of the synapses. A function characterizing the behavior of a neuron can be expressed as:

$$y = \sigma\left(\sum_{i=1}^{n} w_i x_i - \theta_1\right), \tag{1}$$

where n is the number of inputs to the neuron, and

$$\sigma(f) = \left(1 + \exp\left(-\frac{t}{\theta_2}\right)\right)^{-1} \tag{2}$$

is the sigmoid activation function, θ_1 is the threshold level and θ_2 represents the steepness of the sigmoid curve.

Since our work is concerned with images, we shall briefly talk about biological neurons in the human eye. There are two types of neurons in the retina: horizontal neurons and amacrine neurons [2,17,20]. These neurons are laterally connected and modify the messages transmitted along the direct pathway. Once rods and cones are stimulated, the receptor potentials induce signals in both bipolar and horizontal neurons. Bipolar neurons transmit the excitatory signal to ganglion neurons and horizontal neurons transmit the excitatory signal to bipolar neurons. This lateral inhibition enhances contrasts in the visual scene between the areas of the retina that are strongly stimulated and adjacent areas that are weakly stimulated.

There are three separate visual signal processing systems. One system is involved in processing information related to the shape of objects, the second system forms a pathway regarding color of objects, and a third system processes information about movement, location, and spatial organization.

We want to emphasize here that the visual system in the retina preprocesses sensory signals before it sends output to the visual cortex in the brain. Part of that information represents features corresponding to the shape, color, and movement of objects. That means that some responses of neurons represent selective features.

Sensory neurons are spatially distributed and their responses are localized with respect to neuron locations. It is believed that a neuron connected to the sensory system receives inputs from only a portion of the sensory neurons.

The above biological considerations will be used to guide the design of the image recognition neural network (IRNN). Two major characteristics of the IRNN will be emphasized: the hierarchical structure of its architecture, and the localiza-

tion of neurons' responses. As a result, local features will be extracted from portions of an image and combined at a higher layer using a hierarchical architecture in order to simulate the structure of a biological sensory system.

There are numerous approaches to competitive learning. A number of researchers have developed various competitive learning mechanisms [3,14]. The main idea behind competitive learning is to have individual units learn to specialize their responses to a set of similar input patterns. As a result, such structures become feature detectors or pattern classifiers.

A significant new idea presented in this paper is that an image pattern will be spatially analyzed at the bottom layer and synthesized at the top layer.

2. New similarity measure for images

A neuron at a specific location of the bottom layer is connected to a part of the image plane and thus receives a subimage as its input. Then, the subimages are classified using some similarity measure. Appropriate choice of the latter is one of the most important factors determining the success or failure of classification and/or competitive learning. In what follows we shall explain how the images are compared. A new approach to measure the similarity of images was first reported in [18].

An image is usually represented by a scalar function of two spatial variables $g = g(p, q)$, where $g(p, q)$ represents the brightness or gray level of the image at the spatial coordinate (p, q) [11]. Another way of representing an image is to plot the image intensities versus the image coordinates. In other words, gray levels are mapped into a third coordinate of a three-dimensional vector space. A vector

$$\bar{g} = \left(p, q, \frac{g}{\delta}\right)^t \tag{3}$$

corresponds to the pixel at location (p, q) whose gray level is g, and δ is a constant which scales the gray level with respect to coordinates p and q. These representations are satisfactory when we deal with pixels alone but are not sufficient when dealing with entire images.

The third way of representing an image is to use vectors. Let us assume that the size of an image is $u \times v$ and that $q(p, q)$ is the gray value of the pixel at location (p, q). Then, the image can be represented by a vector:

$$\bar{x} = (x_1, \ldots, x_i, \ldots, x_n)^t, \quad n = uv, \tag{4}$$

where $x_i = g(p, q)$. The conversion from the spatial coordinate (p, q) to an index i can be achieved by a transformation

$$i = (q-1)u + p, \quad p = 1, \ldots, u, \quad q = 1, \ldots, v. \tag{5}$$

Special attention is given to a neuron's response when input stimulus represents an image. From Eqs. (1) and (2), we see that the response of a neuron is a function

of the input vector $\bar{x}$ and the weight vector $\bar{w}$. Assuming that $\theta_1 = 0$ and $\theta_2 = 1$ response of the neuron can be written as

$$y = \sigma(\bar{x}^t \bar{w}) = \sigma\left(\frac{|\bar{x}|^2 + |\bar{w}|^2 - d^2}{2}\right), \tag{6}$$

where $d^2 = |\bar{x} - \bar{w}|^2 = |\bar{x}|^2 + |\bar{w}|^2 - 2\bar{x}^t\bar{w}$.

As we can see from the above equation, the neuron responds not only to the distance between vectors $\bar{x}$ and $\bar{w}$ but also to their lengths. This may cause inappropriate response in terms of the degree of matching between vectors $\bar{x}$ and $\bar{w}$. For example, let us assume that there are two neurons with weights $\overline{w_1}$ (neuron one) and $\overline{w_2}$ (neuron two), $\overline{w_1}$ being shorter than $\overline{w_2}$. Let us also assume that $\bar{x}$ is closer to $\overline{w_1}$ than to $\overline{w_2}$ and that the angle between $\bar{x}$ and $\overline{w_1}$ is the same as the angle between $\bar{x}$ and $\overline{w_2}$. Formally,

$$|\overline{w_1}| < |\overline{w_2}|, \quad \text{and} \quad |\bar{x} - \overline{w_1}| < |\bar{x} - \overline{w_2}|, \tag{7}$$

and

$$\frac{\bar{x}^t \overline{w_1}}{|\bar{x}|\,|\overline{w_1}|} = \frac{\bar{x}^t \overline{w_2}}{|\bar{x}|\,|\overline{w_2}|} \tag{8}$$

Therefore

$$\bar{x}^t \overline{w_1} < \bar{x}^t \overline{w_2}, \tag{9}$$

which means that the weighted sum of input vector with neuron's two weight vector is greater than that of neuron one. As a result, neuron two responds more strongly than neuron one even though the input pattern $\bar{x}$ is closer to cluster one. A two-dimensional example is depicted in Fig. 1.

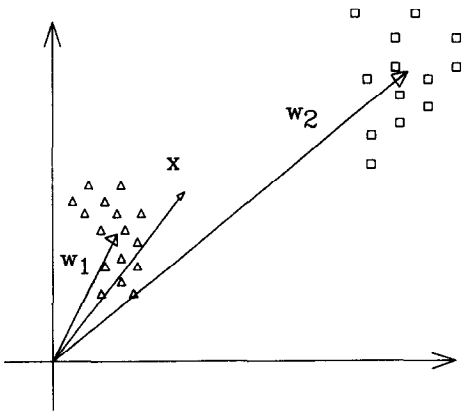


Fig. 1. Misclassification due to vector lengths differences.

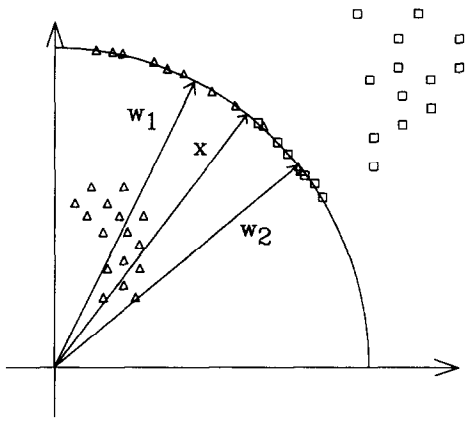


Fig. 2. Misclassification due to normalization.

In order to eliminate the problem associated with the lengths of the vectors, we shall utilize the well-known concept of normalized vector spaces. By substituting $|\bar{x}| = |\bar{w}| = 1$ into Eq. (6), the output response becomes:

$$y = \sigma\left(1 - \frac{d^2}{2}\right), \tag{10}$$

where $d^2 = |\bar{x} - \bar{w}|^2$. Thus, the smaller the distance between the input and the weight vector, the greater the response of the neuron. Fig. 2 illustrates the normalized case. Therefore, if normalization of pattern vectors is allowable, the response of a neuron can be used to calculate the matching between the pattern vectors.

Multitude of similarity measures have been developed but their applicability vary depending on the application and characteristics of the patterns involved [13]. Moreover, some of them are restricted only to binary images. The best known similarity measure, the Euclidean distance, may not be appropriate to characterize the similarity among image patterns since it does not take into account the spatial information. Let us consider the input patterns shown in Fig. 3. The Euclidean distance between any two of the three patterns is the same, namely $\sqrt{2}$.

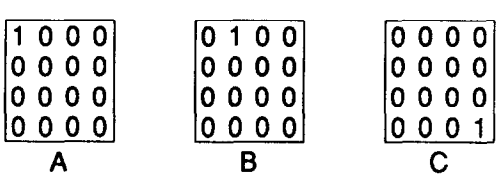


Fig. 3. Three 'images'.

However, we can see that image A is more similar to image B than it is to image C. This illustrates that the Euclidean distance cannot be used for discriminating between the images to determine their visual similarity. Thus, in a vector representation, the spatial local information is either lost or deformed. Therefore, a new measure that takes into account the spatial information is proposed next.

For the new similarity measure between two images $g_1(p, q)$, $g_2(p, g)$, we need to calculate their absolute difference:

$$\Delta g(p, q) = |g_1(p, q) - g_2(p, q)|. \tag{11}$$

In order to utilize the spatial information present in the pixels, we first define intensity center $\bar{m} = (m_p, m_q, 0)$ which satisfies the following relations:

$$\sum_{i=1}^{m_p}\sum_{j=1}^{v}\Delta g(i, j) = \sum_{i=m_p}^{u}\sum_{j=1}^{v}\Delta g(i, j), \tag{12}$$

$$\sum_{i=1}^{u}\sum_{j=1}^{m_q}\Delta g(i, j) = \sum_{i=1}^{u}\sum_{j=m_q}^{v}\Delta g(i, j). \tag{13}$$

The main property of the intensity center is that an image can be divided into two parts by a line passing through the center so that the sum of intensities of pixels in each part is the same.

Let us define a pixel vector as:

$$\bar{p} = \frac{\delta}{1+\delta}(\Delta\bar{g} - \bar{m}) = \frac{1}{1+\delta}\begin{pmatrix}\delta(p - m_p)\\ \delta(q - m_q)\\ \Delta g(p, q)\end{pmatrix}, \tag{14}$$

where δ is defined as a ratio of significance between the location and the intensity (gray) level. When δ becomes zero, the pixel vector contains only gray-level

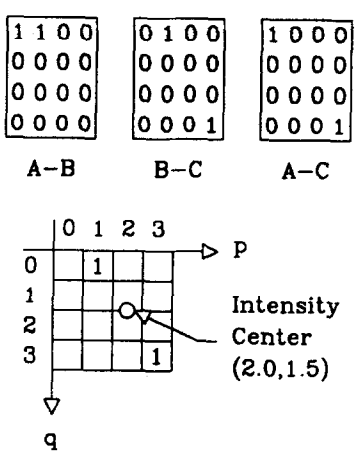


Fig. 4. Image differences and the intensity center.

Table 1
Comparison of distances ($\delta = 1$).

	A-B	B-C	A-C
Intensity center	(0.5, 0.0)	(2.0, 1.5)	(1.5, 1.5)
Dot product	0	0	0
Euclidean	1.414214	1.414214	1.414214
S	2.236068	4.123106	4.690416

information. When δ reaches infinity, the pixel vector becomes just a coordinate position of that pixel.

Now, a similarity between the two images is defined as:

$$S = \sum_{p=1}^{u} \sum_{q=1}^{v} |\bar{p}| \Delta g(p, q)$$

$$= \sum_{p} \sum_{q} \frac{\Delta g(p, q)}{1+\delta} \sqrt{\delta^2 (p - m_p)^2 + \delta^2 \left(q - m_q^2\right) + \Delta g(p, q)^2}. \tag{15}$$

We shall apply both the Euclidean measure and the new similarity measure, S, to the previously shown example. Fig. 4 shows the image differences and the intensity center. The results are shown in Table 1. Dot products between the images are also included to demonstrate the response of neurons. The image distance between the sample images demonstrates that images A and B do form the most similar pair of images. Also, images B-C are more similar than images A-C. All distances are normalized with respect to the distance (A-B) and the results are shown in Table 2.

The second example compares the new similarity measure, S, with the Euclidean metric using images of tire wheels. The measurement process of taking these images is described later in the *Results* section. Fig. 5 shows portions of the tire wheels image used. The size of the images is 64×64 with maximum gray level of 256. There are three visually separable different types of wheels: the first two images are more similar than any other combination. The third image looks more different than the other two.

The calculated similarities for the tire wheels are shown in Table 3. Using the image similarity, S, we notice that image 031A is far away from images 050A and

Table 2
Comparison of normalized distances ($\delta = 1$).

	A-B	B-C	A-C
Euclidean	1.000	1.000	1.000
S	1.000	1.844	2.098

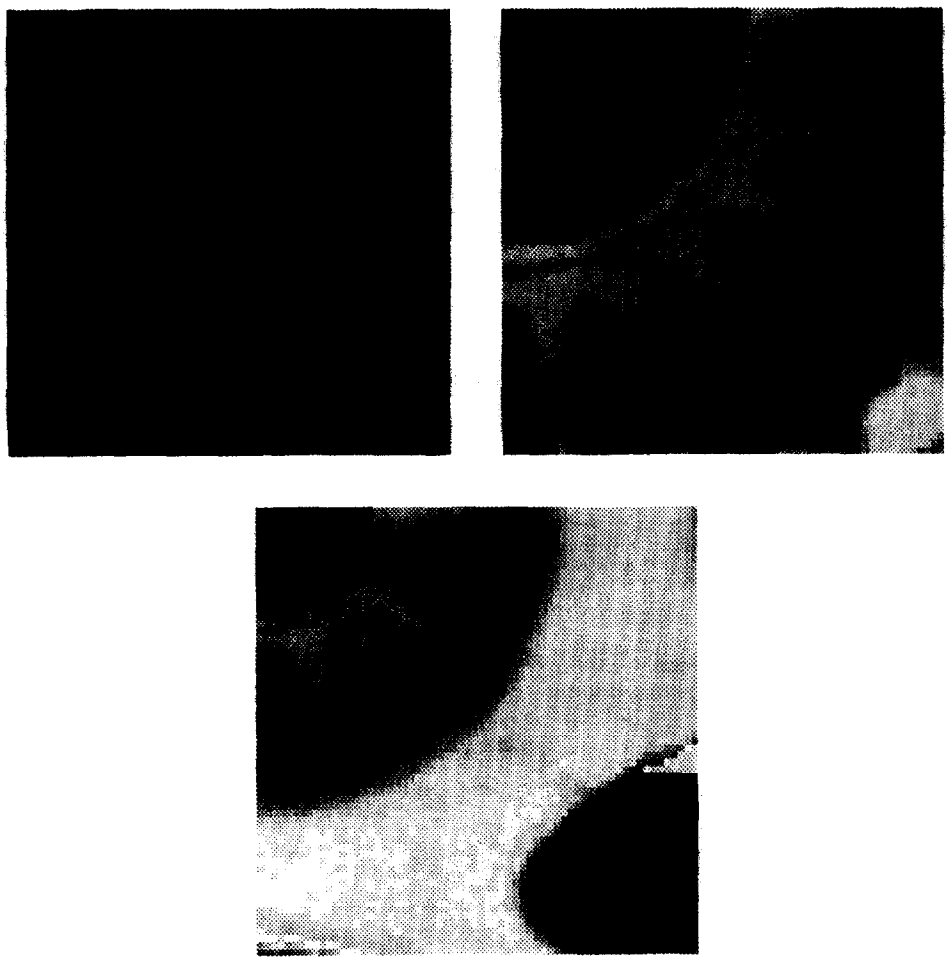


Fig. 5. Portions of tire wheel images. The identification codes for the tire wheels are 050A, 052A and 031A, shown clockwise from upper left.

052A. As can be seen, the image similarity, S, produces more realistic results because the wheels 050A and 052A are identical except that wheel 052A is silver-painted and wheel 050A is not painted.

Table 3
Comparison of the tire wheel image similarities ($\delta = 1$).

	050A-052A	052A-031A	050A-031A
Intensity center	(29, 29)	(36, 38)	(33, 35)
Dot product	36643936	43795326	47958148
Euclidean	4234	5784	5620
S	18998612	50351584	63152160

3. Image Recognition Neural Network

The image recognition neural network (IRNN) described here consists of a hierarchy of several layers of artificial neurons, arranged in planes to form layers. The IRNN consists of layers divided into top, middle and bottom layers. There can be one or more middle layers. The architecture of the IRNN is divided into two segments: visual and associative, shown in Fig. 6.

The visual segment operates on an input image and generates features which are then processed by the associative segment. The visual segment may contain one or two layers of neurons. An image is divided by the visual segment into subimages. A set of neurons is assigned to each subimage for classification and in order to produce appropriate codes depending on local features. The extraction of local features is based on the similarity among subimages.

The associative segment combines the local features and associates them with correct recognition codes. The associative segment consists of one or two layers of neurons, namely, a global recognition layer and (possibly) a local recognition layer. The output of the visual segment constitutes input to the associative segment. The main role of the associative segment is to relate the features generated by the visual segment to the desired recognition code. Before we discuss the IRNN in detail, the basic ideas of how it works are explained by means of the following example.

Fig. 7 shows the structure of the simple IRNN. The network consists of seven neurons. Each neuron corresponds to a processing unit, $N_1, \ldots, N_7$. Let us assume

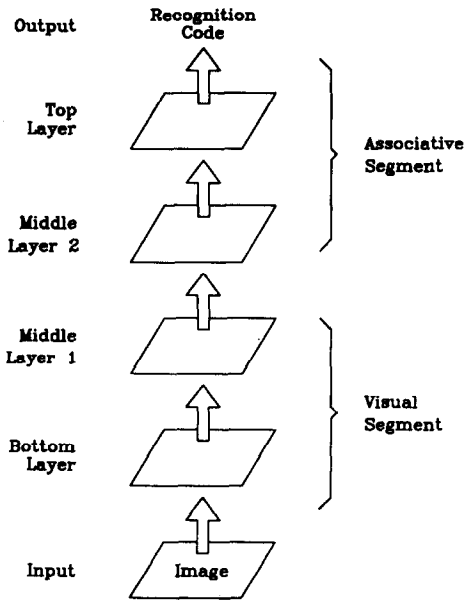


Fig. 6. Structure of the IRNN. The arrows denote the direction of signal flows.

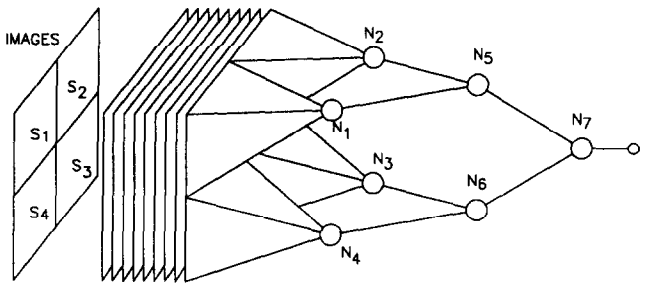


Fig. 7. IRNN with seven neurons $N_1,\ldots,N_7$. $S_1,\ldots,S_4$ represent subimages.

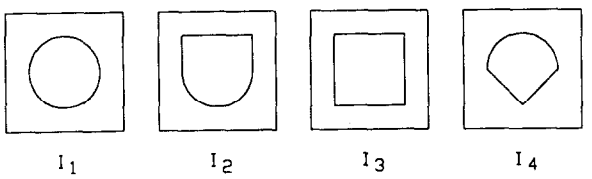


Fig. 8. Images for classification by the IRNN shown in Fig. 7.

that a task is to recognize four different images shown in Fig. 8. Each image is divided by the neural network into four subimages, $S_1,\ldots,S_4$. Neurons N_1 through N_4 'see' subimages S_1 through S_4, respectively. These four neurons will classify the images I_1 through I_4 into categories using the similarity between the subimages as a criterion.

Neurons N_1 through N_4 operate in a self-organized mode to classify the subimages. For instance, the input patterns for N_1 are two portions of a circle and two portions of a rectangle as shown in Fig. 9. Because there are two distinct categories (circle and rectangle), N_1 will classify the input images into two groups. This operation is called self-organization. The resulting classes, for neurons N_1 through N_4, and their outputs are shown in Fig. 10.

The outputs from neurons N_1 and N_2 are combined to form a feature which will be classified by neuron N_5. The outputs of N_3 and N_4 form a feature for neuron N_6. The role of neurons N_5 and N_6 is to generate a combined code corresponding to the combined regions S_{12} and S_{34}, where S_{12} represents the combined region of S_1 and S_2. Fig. 11 shows the classified input patterns and output codes for neurons N_5 and N_6.

Finally, neuron N_7 learns the relation between the combined local features and the desired recognition. The resulting relations for neuron N_7 are shown in Fig. 12. In this example, neurons N_1 through N_6 constitute the visual segment, and neuron N_7 constitutes the associative segment of the IRNN architecture.

3.1 Visual segment

As described above, neurons in the visual segment take an image and produce local features as their output. The visual segment consists of one or two layers of

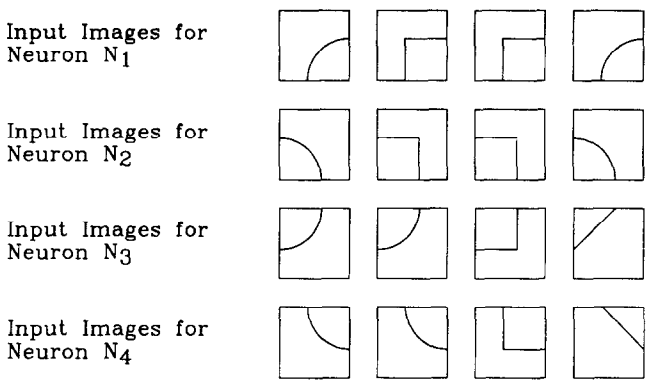


Fig. 9. Sequences of input images for neurons $N_1, \ldots, N_4$.

specially designed neurons. Neurons in the bottom layer are called sensory neurons, while the neurons in the second layer (if required) are called feature-aggregating neurons.

3.1.1 Localized response of a sensory neuron

In the process of designing the IRNN we wanted to achieve localization of a neuron's response by the bottom layer. This means that a neuron should generate output corresponding to only a part of an input image.

The response of a neuron corresponding to only a part of its input is called a local response. The part of the input image which activates a neuron will be called a subimage. The localization of the response is achieved by adding a synapse with a fixed weight (strength) between the input nodes, and the variable weight connections. The value of the fixed weight is used to control the localization of a neuron's response. This synapse can be viewed as lateral inhibition of a neuron.

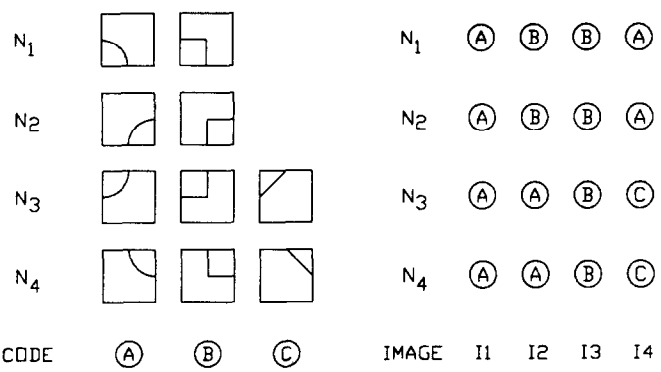


Fig. 10. Classification by N_1 through N_4 neurons of I_1 through I_4 images, and the corresponding codes.

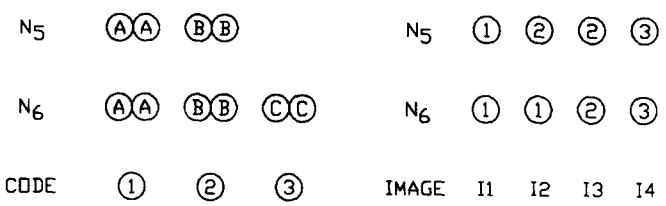


Fig. 11. Images classified by neurons N_5 and N_6, and the output responses with respect to input images I_1 through I_4.

The fixed weights do not have to be uniform. Examples are shown in Fig. 13. The only requirement is that the central point must have the strongest connection. The weights of the connections decrease with increasing the distance from the center. Zero weight represents no connection.

3.1.2 Contrast enhancement

Classification of an image can be severely affected by either brightness or contrast. Here, by contrast we mean the variation of intensity within an image, and by brightness the average of all intensities. For a given set of images brightness and contrast are fixed. The relative importance of brightness and contrast of a subimage varies with the size of the image. When the subimage becomes smaller, the local image loses texture information. In the extreme case a single pixel may contain intensity information corresponding only to the brightness with no texture information in it. Therefore, the relative importance of brightness is emphasized in the subimages.

For the purpose of relative enhancement of contrast in the subimages, a neuron responding to brightness will be added. This neuron will inhibit the sensory neuron depending on the degree of brightness. Fig. 14 illustrates the sensory neuron model which contains three types of cells: A, B and C. Cell A can be seen as a neuron corresponding to a horizontal cell in the human retina since it also responds to the brightness of an input image. Cell A inhibits cell B. It is a summing node with one inhibitory input and one excitatory input. Cell C produces an output value corresponding to the degree of match between the input and

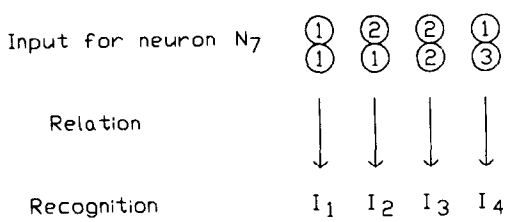


Fig. 12. Neuron N_7 find the relation between the desired recognition and the input features.

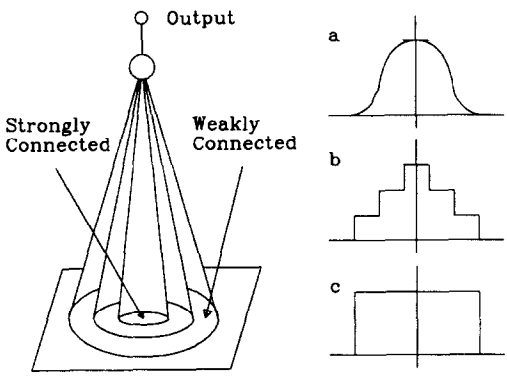


Fig. 13. Examples of different distributions of fixed weights.

weight vectors. Sensory neurons of this type will be used in the bottom layer of the visual segment to extract local features.

The sensory neurons form the bottom layer of the IRNN. We shall use a three dimensional example to explain the architecture of the bottom layer. The arrangement of the neurons in the bottom layer will be discussed from two points of view: vertical and horizontal.

The vertical arrangement is used for self-organization of subimages. Each neuron in this vertical arrangement represents a distinct class of images. All neurons in a vertical arrangement have the same connections, with fixed weights, with the input image. The set of neurons in a vertical arrangement will be called a block. The output of a block represents the result of identifying a specific feature in the subimage.

If a number of blocks is distributed over the image plane to form a planar architecture of the bottom layer, then we talk about the horizontal arrangement. The areas connected by the blocks in the bottom layer depend on the distribution

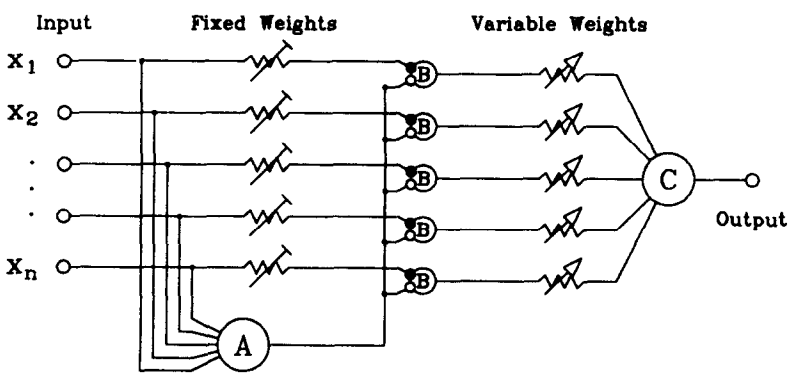


Fig. 14. Structure of a sensory neuron.

of the fixed weights. Some portions of the input image might overlap. Fig. 15 illustrates a layer of sensory neurons with overlapping subimages.

3.1.3. Feature aggregating layer

In addition to the layer of sensory neurons, a layer of feature-aggregating neurons is also a part of the visual segment. The feature aggregating layer constitutes the second layer of visual segment. Fig. 16 shows details of a feature-aggregating neuron. The main role of this layer is to combine the information coming from the subimages. The arrangement of neurons in this layer is the same as in the layer of sensory neurons. In other words, this layer consists of many blocks and each block consists of a set of neurons. Fig. 17 illustrates the layer of feature aggregating neurons.

3.2. Associative segment

The second part of the IRNN is called associative segment. Inputs to this segment are the combined local features extracted by the visual segment, and the output is a recognition code corresponding to the input image. The role of the associative segment of the IRNN is to relate extracted features to the recognition codes. Here, recognition code usually represents a binary number, with 1 corresponding to the firing state of a neuron.

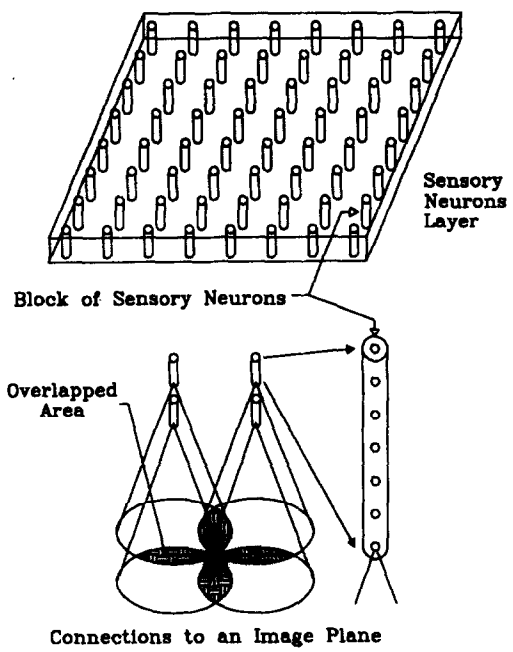


Fig. 15. A layer of sensory neurons.

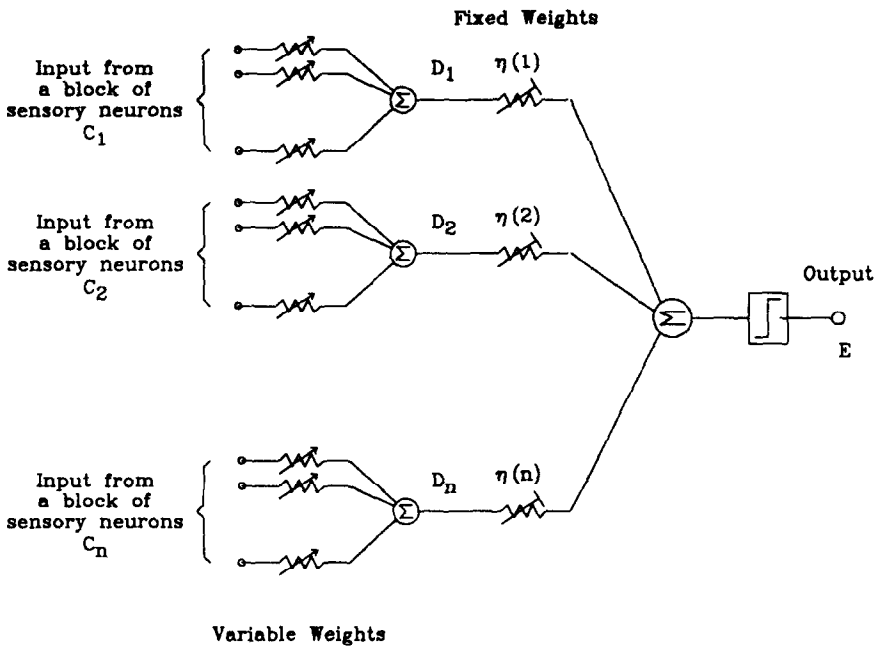


Fig. 16. Diagram of a feature aggregating neuron.

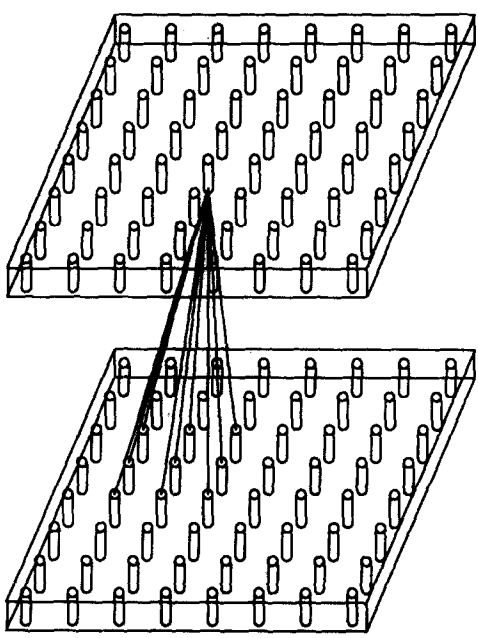


Fig. 17. Layer of feature aggregating neurons.

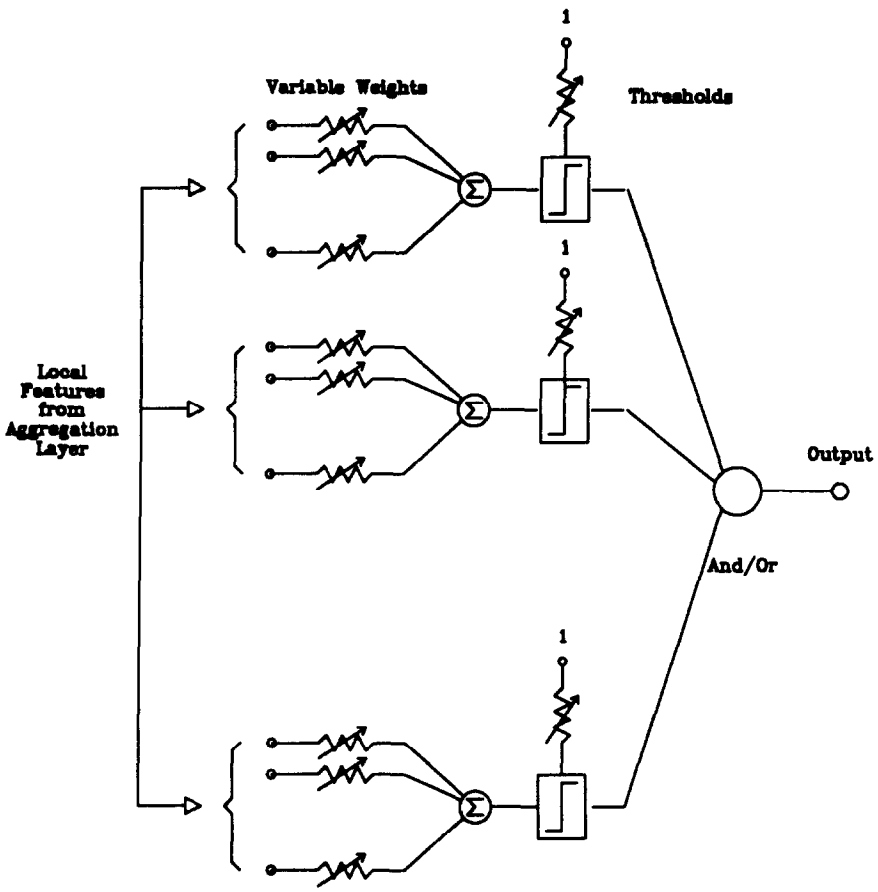


Fig. 18. Model of a neuron for local recognition.

The associative segment consists of two layers of neurons as shown in Fig. 6. The first layer achieves local recognition and the second makes a decision.

A model of a neuron for subimage recognition is a Madaline-type [23] neural network which finds decision boundaries for a set of inputs belonging to different categories. As shown in Fig. 18, inputs to the neuron come from several blocks of the visual segment. Each block represents a combined local feature. Therefore, the neuron associates a set of combined features with a certain category. The set of such neurons produces a set of outputs which, if taken together, can be thought of as a recognition code.

4. Learning algorithms and training methods

The goal of training is to find weight vectors so that an appropriate recognition code for an image can be found. Each layer of the IRNN will be treated differently

during training. The IRNN has a number of learning algorithms and corresponding training methods, since different types of neurons require different learning algorithms. As said before, there are three types of neurons used in the IRNN: Sensory Neurons, Feature-Aggregating Neurons and Global Recognition Neurons.

The sensory neurons and the feature-aggregating neurons are similar in the sense that both work in an unsupervised learning mode. Global recognition neurons work in a supervised learning mode. The algorithms for the unsupervised learning are developed based on the competitive learning algorithm [15]. The supervised learning algorithms are based on the error correction method [16].

The terms 'learning rule', 'learning algorithm' and 'training method' are used interchangeably in neural networks literature. In this paper, however, we distinguish between the meanings of these terms. The term 'learning rule' will be used to represent the method of weight updating for a single neuron. The term 'learning algorithm' will be concerned with a sequence of processes to update the weights of a set of neurons in a layer. The term 'training method' will refer to the way of training the whole IRNN for a given recognition problem.

4.1. Learning rule for neurons in the associative segment

The learning rule used for neurons in the associative segment is a delta rule [21] and is expressed as follows:

$$\bar{w}(t+1) = \bar{w}(t) + \alpha e_t \frac{\bar{x}(t)}{|\bar{x}(t)|^2}. \tag{16}$$

4.2. Learning in the visual segment

We shall formulate the equations modeling the neuron response and the training procedures for each layer in the visual segment.

The input to the layer of sensory neurons is an input image divided into subimages. The equations and the learning processes for each subimage are identical.

(a) Response of the sensory neuron

A sensory neuron, as shown in Fig. 14, consists of three types of cells A, B and C. The outputs of the cells are denoted by A, B_i, and C, respectively. The output of cell A

$$A = \sum_{i=1}^{n} x_i \tag{17}$$

represents brightness. The output of cell B_i:

$$B_i = \gamma(\eta(i)x_i - \delta A), \quad i = 1, \ldots, n \tag{18}$$

corresponds to the gray value of the pixel x_i (after performing relative contrast enhancement and localization). The $\eta(i)$ is a fixed weight associated with the ith

pixel. The constant δ is proportional to the amount of contrast enhancement. Therefore, the output of a sensory neuron C can be expressed as:

$$C = \sigma\left(\sum_{i=1}^{n} w_i B_i\right) = \sigma\left(\sum_{i=1}^{n} w_i \gamma\left(\eta(i)\, x_i - \sum_{j=1}^{n} x_j\right)\right). \tag{19}$$

Several sensory neurons are arranged to form a vertical cylinder in the bottom layer of the IRNN. This forms a block of neurons. Such a block takes a subimage as an input and produces a number of scalar values at the output.

The response of neurons to a subimage $\bar{x}$ depends on the individual weight vectors $\bar{w}_1, \bar{w}_2, \ldots, \bar{w}_m$. The outputs $C_1, C_2, \ldots, C_m$, when taken together, represent a local feature corresponding to subimage $\bar{x}$.

$$\bar{C} = (C_1 \ldots, C_i, \ldots, C_m)^t \quad \text{and} \quad C_i = f(\bar{w}_i, \bar{x}), \quad i = 1, \ldots, m. \tag{20}$$

Fig. 19 illustrates the neural network with a block of sensory neurons that produces vector $\bar{C}$ (a local feature) in a self-organized manner.

Eq. (20) represents a subimage. In a case where multiple subimages are presented to a block of sensory neurons a set of weight vectors has to be modified so that the output of a block represents classification which can be considered as a local feature. To achieve this goal a modified form of competitive learning will be used by modifying the way in which a winner is selected.

(b) Learning algorithm

The algorithm to find a set of weight vectors for a block of sensory neurons corresponding to a set of local images $\bar{x}_i$, $i = 1,2,\ldots,n$, follows:

Step 1. Create a neuron in a vertical arrangement and set its weight vector equal to an input vector. Without loss of generality, the first input vector can be used:

$$p = m = 1 \quad \text{and} \quad \bar{w}_1 = \bar{x}_1$$

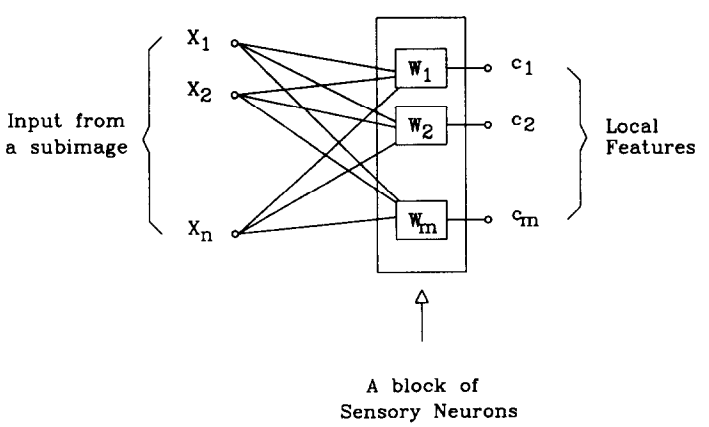


Fig. 19. A block of sensory neurons.

where m is the number of output nodes in the block and p is an index counter.

Step 2. Increment p: if p is less than or equal to n go to Step 3. Otherwise, stop. Reset p for next iteration.

Step 3. Present image x_p and calculate the neuron responses:

$$C_i,\ i = 1, \ldots, m.$$

Step 4. Choose the node responding with maximum value

$$C_j \geq C_i, \quad \text{for all } i \neq j.$$

Step 5. Calculate similarity between the input image $\bar{x}_p$, and the weight vector $\bar{w}_j$,

$$S = \sum_{p=1}^{u} \sum_{q=1}^{v} |\bar{p}|\, \Delta g(p, q).$$

Step 6. If the difference is small, update the weights, and go to Step 2,

$$\bar{w}_j(t+1) = \alpha \bar{x}_p + (1-\alpha)\bar{w}_j(t).$$

Step 7. If the difference is big, set neuron response C_j temporarily to zero.

Step 8. If all outputs are zero then increase the number of nodes in the block and initialize the corresponding weight vector with the current input vector:

$$m = m + 1 \quad \text{and } \bar{w}_m = \bar{x}_p$$

and go to Step 2. If there exists at least one non-zero response go to Step 4.

4.2.1. Second layer of the visual segment

The general learning procedure for the second layer is the same as for the bottom layer since both self-organize input patterns based on their similarities. Neurons in the second layer also use a modified competitive learning. However, the Euclidean metric is used here to measure the similarity between the features generated by the bottom layer.

(a) Response of feature-aggregating neurons

Self-organization of the subimages is obtained within a block of neurons in the bottom layer using the modified competitive learning algorithm. The bottom layer consists of many blocks and each block produces vector $\bar{C}$ at its output.

A set of sensory blocks is connected to a neuron in the second layer, with outputs of the sensory blocks denoted by

$$\bar{C}_i = (C_{i1}, C_{i2}, \ldots, C_{im})^t, \quad i = 1, \ldots, n, \tag{21}$$

where m_i is the number of neurons in the ith block of the bottom layer and n is the number of blocks connected to a neuron E as shown in Fig. 20.

The $\bar{C}_i$'s constitute the inputs to the feature-aggregating layer. First, the case when only one neuron is responding to input $\bar{C}_i$ is considered. To obtain formulas for such a neuron there is a need to examine the structure of the neurons in the

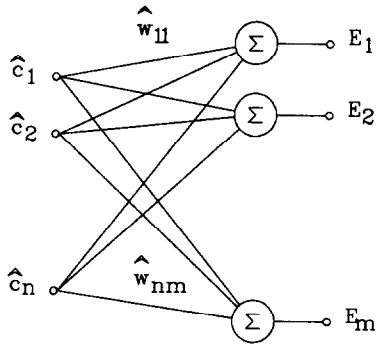


Fig. 20. Illustration of the signal flow within a block of neurons in the second layer of the IRNN. Vector $\bar{E} = (E_1, \ldots, E_m)$ represents a local feature.

second layer. The output of neuron E is the weighted sum of D_1 through D_n, followed by an activation function ϕ:

$$E = \phi\left(\sum_{i=1}^{n} \eta(i) D_i\right), \tag{22}$$

where $\eta(i)$ is a fixed weight associated with the cell output D_i, and

$$D_i = \sum_{j=1}^{m_i} C_{ij} w_{ij}, \quad i = 1, \ldots, n. \tag{23}$$

Therefore, the neuron output becomes

$$E = \sum_{i=1}^{n} \eta(i) \sum_{j=1}^{m_i} C_{ij} w_{ij} = \sum_{i=1}^{n} \sum_{j=1}^{m_i} \eta(i) C_{ij} w_{ij}. \tag{24}$$

The learning algorithm applied to the second layer is similar to that of the bottom layer. Again we have to build a block of neurons to achieve self-organization of local features.

4.3. Learning in the associative segment

The learning rule used here is the error-correction method [23]. The associative segment needs to extract a recognition code or to estimate an attribute from local features. IRNN executes this task in two layers. The learning algorithm for the first layer to recognize the input image from local features follows.

(a) Neuron response

The input of the neuron in the associative segment is $\bar{E} = (E_{i1}, E_{i2}, \ldots, E_{in})^t$ and the output is $\bar{y} = (y_1, y_2, \ldots, y_m)^t$, where

$$y_i = \sigma\left(\sum_{j=1}^{n} w_{ij} E_{ij} - \theta_i\right), \quad i = 1, \ldots, m. \tag{25}$$

Let us assume that the desired output is $\bar{D} = (D_1, D_2, \ldots, D_m)^t$

(b) Learning algorithm

Step 1. Set the weights w_{ji} and thresholds θ_j to small random values (this is done only once). Set index p to zero.

Step 2. Increase p; If p is less than or equal to n present a new training pair (input $\bar{x}_p$, desired output $\bar{D}_p$) and go to Step 5.

Step 3. Calculate outputs and the system error.

$$y_i = \sigma\left(\sum_{j=1}^{n} w_{ij}E_{ij} - \theta_i\right), \quad i = 1, \ldots, m, \tag{26}$$

$$e_p = \sum_{i=1}^{m} (D_{ip} - y_{ip})^2, \tag{27}$$

Step 4. Update the weight using delta rule

$$\bar{w}(t+1) = \bar{w}(t) + \alpha e_p \frac{\bar{x}(t)}{|\bar{x}(t)|^2} \tag{28}$$

and go to Step 2.

Step 5. Set p to 0 and calculate the system error

$$E = \sum_{i=1}^{n} e_i. \tag{29}$$

If the error is small enough or ceases to decrease; stop. Otherwise go to Step 2.

4.4. Training method

The IRNN requires a specific training method so that the system performs the required recognition in an efficient manner. The performance of the network will vary considerably depending on the training method, even if the data and learning rules are the same. Without proper training the IRNN may fail to learn the required recognition problem. The method is summarized as follows:

Step 1. Train the bottom layer alone while disabling the learning ability of all other layers. Present randomly ordered sequence of images until all sensory neuron responses are stable.

Step 2. Train the second layer only. The bottom layer provides input for the second layer by propagating its response to the second layer. Iterate until the outputs of the second layer are stabilized.

Step 3. The associative segment is trained using the output of the visual segment, and the known desired recognition code. The weights in the visual segment are not modified. Training of the associative segment continues until the system error is reduced to some acceptable value.

After the IRNN is completely trained by the above processes, there is a possibility of having to train the neural network again on new data. If the new data consists of training pairs (inputs and the desired outputs) then additional training should start without initializing the weights. The current weights are treated as initial values and training continues with the new data. When the desired outputs are not available, only the visual segment should be trained.

5. Experiment

In a production process, polycast car wheels of different models and shapes are placed randomly on a conveyor. The process requires that the type of a wheel is identified before being sent for a subsequent operation. Manual wheel identification by an operator is error prone and leads to unnecessary scrap and reworks. Using a vision system together with a neural network in place of the operator is thus of great interest.

Images of thirteen types of car wheels were obtained using a CCD matrix video camera, a frame grabber and a PC. For each wheel type, eleven images were taken. In each of the eleven images, the position of the wheel with relation to the camera was altered. This was done by rotating the wheel slightly about the wheel's axis and by changing the angle at which the camera faces the wheel by a small amount. For all the images, a black background was used and the distance between the camera and the wheel was, for practical purposes, constant.

The size of each image was 320 by 230 pixels, with pixels having 256 gray levels. Such an image would be too large for a neural network to process. Hence, reduction in dimensionality was performed on the images before they were used. The reduction process is described later.

The architecture of the IRNN for recognition of tire wheels consists of four layers of neurons. The bottom layer has 15×15 blocks of neurons, each block containing a maximum of 20 neurons. Therefore, the total number of neurons in the bottom layer is 4,500. Each of the neurons in this layer takes as an input subimages of size 8×8. This results in a total of 288,000 connections between the

Table 4
Fixed weights for the sensory neurons (8×8).

0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25
0.25	0.5	0.5	0.5	0.5	0.5	0.5	0.25
0.25	0.5	1	1	1	1	0.5	0.25
0.25	0.5	1	1	1	1	0.5	0.25
0.25	0.5	1	1	1	1	0.5	0.25
0.25	0.5	1	1	1	1	0.5	0.25
0.25	0.5	0.5	0.5	0.5	0.5	0.5	0.25
0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25

sensory neuron layer and the input image plane. The weights for the localization of a neuron's response connecting it to the 8 × 8 input window are shown in Table 4. All of the 8 × 8 windows overlap with each other by four pixels in both vertical and horizontal directions.

There are four blocks of neurons in the second layer with again a maximum of 20 neurons in each block. Each neuron in the second layer takes its inputs from 10 × 10 blocks of sensory neurons. Therefore, the total number of neurons in this layer is 80 and the number of connections between the sensory neuron layer and the second layer is 160,000. The weights for the 10 × 10 window are shown in Table 5.

The threshold values for the visual segment are selected experimentally. First, from a few samples the image similarity among 8 × 8 subimages is calculated to obtain a rough value for a threshold to be used in competitive learning algorithm. Depending on the threshold the number of self-organized classes expanded in each block varies. During the learning phase the threshold value is adjusted so that the maximum number of nodes expanded within a block is less than 20.

The number of outputs in the top layer of the IRNN is equal to the number of classes to be identified. In the case of wheel images the number of outputs is thirteen.

As said above, the size of the original image was 320 × 230. Since this size was too large for processing by the IRNN it was reduced to 64 × 64. There were several reasons for this reduction, the most important being that the computational load and the memory requirements for simulation were too high for the available hardware.

The reduction of the size was done by cutting-off a portion of an object from the original image. This process was performed by taking into account the following:

(1) The original images have different resolutions in vertical and horizontal directions. The number of pixels in a vertical and a horizontal lines (within a

Table 5
Fixed weights for the feature-aggregating neurons (10 × 10).

0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25
0.25	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.25
0.25	0.5	1	1	1	1	1	1	0.5	0.25
0.25	0.5	1	1	1	1	1	1	0.5	0.25
0.25	0.5	1	1	1	1	1	1	0.5	0.25
0.25	0.5	1	1	1	1	1	1	0.5	0.25
0.25	0.5	1	1	1	1	1	1	0.5	0.25
0.25	0.5	1	1	1	1	1	1	0.5	0.25
0.25	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.25
0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25

unit rectangle) is not the same. The number of pixels in both vertical and horizontal diameters of a tire wheel image is counted. The approximate ratio is 198:173. This ratio needs to be adjusted in order to obtain equal significance in both directions before calculating the image similarity.

(2) The position and orientation of the wheel within the 320×230 image frame is random. Because the IRNN calculates the similarity between images, rotated and shifted versions of the same image are considered to be different. Thus, image samples need to have similar orientation and location. To deal with this problem the locations of the center of a wheel and the air valve hole are identified visually. Then, the image is rotated with respect to its center so that the air valve hole is placed on a horizontal line.

(3) One quarter of an image contains enough information for recognition of tire wheels, because the wheels are symmetrical with respect to a line passing through the center of the wheel.

The obtained 64×64 images include only the first quadrant of the tire wheel with respect to the wheel center.

Thirteen samples of wheels are used in this experiment. Some of the wheels are exactly the same except for color. Six wheels are not painted and five are silver painted. There are two wheels which have highlighted rings and edges. All thirteen wheels must be identified as belonging to different categories.

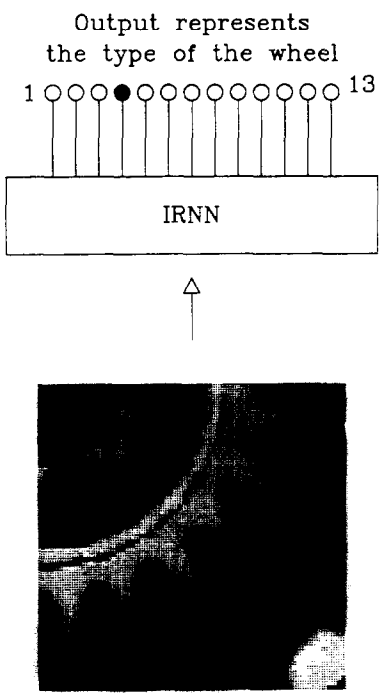


Fig. 21. IRNN for the wheels images.

There are nine images for each of the thirteen types of wheels. Six are used for learning. The remaining three images are used for testing. Therefore, a total of 78 learning samples and 39 testing samples are available. Fig. 21 shows the IRNN for the wheels images and output assignment for each category.

The IRNN classifies all categories correctly. The threshold value was chosen to be 210,000. The maximum number of classes produced by the blocks of sensory neurons was seven with the given threshold value. Two additional simulations were carried out with rotated learning and testing samples. It was done by swapping learning and testing data. The recognition results were again 100% correct.

6. Conclusions

An image recognition neural network (IRNN) for recognition of an object and/or estimation of attributes of an object has been proposed in the paper. A new similarity measure for images was proposed in the paper and tested on images of tire wheels. Comparison with the Euclidean metric demonstrated significant differences between these two measures of similarity.

The IRNN was developed for real-life image recognition problems; it effectively deals with problems of extracting appropriate information from an image relevant to the desired classification. The time it takes for the IRNN to produce output is equal to the time it takes to propagate a signal through the network, which is almost negligible. Future work may include the use of entropy function to decide the size and composition of subimages along the lines proposed in [5].

IRNN has two important properties. First, it exploits the spatial information present in the images. Because the IRNN takes gray level images as inputs and operates on them using the similarity among images, there are many areas of applications. For instance, the IRNN can be applied to the characterization of materials [6,19], quality control in manufacturing systems, identification of sequence of parts in a factory automation system, and identification of objects for an automatic guidance of robots, etc. All these applications require real-time processing.

The second property of the IRNN is its relative low connectivity with respect to the number of neurons used in the network. For example, the number of connections between the bottom layer and the image plane for the tire wheels is 288,000 for 4,500 neurons. For comparison, Kohonen's network would require 18,432,000 connections for the same number of neurons, and the ART network would require even larger number of connections. The low connectivity was made possible by localization of neuron responses. Neurons in the visual segment respond to a small portion of input image thus allowing for the reduction in the number of connections between the layers. This property is important because the speed of simulation is dependent on the number of weights. The lower the number of weights (connectivity) the smaller the time required for simulation. This also relates to the size of data which can be processed on a given computer.

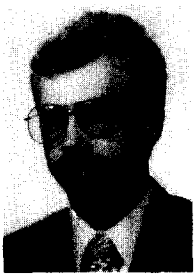
Acknowledgements

This work was partially supported by National Science Foundation, grant number DDM-9015333. Thanks also go to Edison Industrial Systems Center in Toledo for supplying us with the wheel images.

References

[1] M.A. Arbib, *Brains, Machines, and Mathematics*, 2nd ed. (Springer-Verlag, Berlin, 1990).
[2] C. Bruce, R. Desimone and C.G. Gross, Visual properties of neurons in a polysensory area in superior temporal sulcus of the macaque, *J. Neurophysiol.* 46 (2) (Aug. 1981) 369–384.
[3] G.A. Carpenter and S. Grossberg, A massively parallel architecture for a self-organizing neural pattern recognition machine, *Computer Vision Graphics Image Processing* (1987) 54–115.
[4] E.R. Calaniello, Outline of theory of thought processes and thinking machine, *Theoret. Biol.* 2 (1961) 204.
[5] K.J. Cios and N. Liu, Machine learning in generation of a neural network architecture: A continuous ID3 approach, *IEEE Trans. Neural Networks* (March 1992) 280–291.
[6] K.J. Cios, A. Vary, L. Berke and H.E. Kautz, Application of neural networks to prediction of advanced composite structures mechanical response and behavior, *Comput. Syst. in Eng.* 3 (1-4) (Pergamon Press, Oxford, 1992) 539–544.
[7] B.G. Farley and W.A. Clark, Simulation of self-organizing systems by digital computer, *IRE Trans. Inform. Theory* IT-4 (1954) 76.
[8] K. Fukushima, Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position, *Biol. Cybernet.* 36, (4) (April 1980) 193–202.
[9] K. Fukushima, A neural network model for selective attention in visual pattern recognition and associative recall, *Applied Optics* 26 (23) (Nov. 1987) 4985–4992.
[10] K. Fukushima, Neocognitron: A hierarchical neural network capable of visual pattern recognition, *Neural Networks*, 1 (2) (April 1988) 119–130.
[11] R.C. Gonzales and P. Wintz, *Digital Image Processing* (Addison-Wesley, Reading, MA, 1977).
[12] D.H. Hubel and T.N. Wiesel, Receptive fields and functional architecture in two nonstriate visual areas (18 and 19) of the cat, *J. Neurophysiol.* 28 (1965) 229–289.
[13] J. Kittler, Feature selection and extraction, *Handbook of Pattern Recognition and Image Processing* (Academic Press, New York, 1986) 59–83.
[14] T. Kohonen, Adaptive, associative, and self-organizing functions in neural computing, *Applied Optics* 26 (23) (Dec. 1987) 4910–4918.
[15] D.E. Rumelhart and D. Zipser, Feature discovery by competitive learning, *Cognitive Sci.* 9, (1) (January/March 1985) 75–112.
[16] D.E. Rumelhart, G.E. Hinton and R.J. Williams, Learning representations by back-propagating errors, *Nature* 323 (Oct. 1986) 533–536.
[17] G.M. Shepherd and C.A. Greer, Olfactory bulb: The synapic organization of the brain, third ed. (G.M. Shepherd, ed.)(Oxford University Press, 1990) 133–169.
[18] I. Shin and K.J. Cios, A neural network application for image pattern recognition, *Proc. Fourth Conf. on Neural Networks and Parallel Distributed Processing*, Indiana Univ. Purdue Univ. at Fort Wayne (1992) 143–151.
[19] I. Shin and K. Cios, Piecewise linear estimation of mechanical properties of materials with neural networks, *Proc. Third Conf. on Neural Networks and Parallel Distributed Processing*, Fort Wayne (1991) 58–66.
[20] P. Sterling, Retina: The synapic organization of the brain, third ed. (G.M. Shepherd ed.) (Oxford University Press, 1990) 170–213.
[21] P. Werbos, Beyond regression: New tools for prediction and analysis in the behavioral sciences, Ph.D. Dissertation, Harvard University.

[22] B. Widrow and M.E. Hoff, Adaptive switching circuits, *Human Neurobiol.*, 4 (1985) 229.
[23] B. Widrow and R. Winter, Neural nets for adaptive filtering and adaptive pattern recognition, *Computer* (March 1988) 25–39.



Krzysztof J. Cios received an M.S. degree in electrical engineering and a Ph.D. degree in computer science from Technical University (AGH), Krakow, Poland. Currently he is completing his MBA degree at the University of Toledo.

Dr. Cios is an Associate Professor of Electrical Engineering at the University of Toledo, Toledo, OH, USA. His research interests are in the area of fuzzy and neural systems, machine learning, pattern recognition and applications. He is a senior member of the IEEE and a member of Sigma Xi.

Dr. Cios is an associate editor of IEEE Transactions on Fuzzy Systems and on the editorial board of the Handbook of Neural Computation to be published by Oxford Press and IOP Publ. Inc.

Inho Shin received a B.S. degree in electronics engineering from the Sogang University, Korea, and his M.S. and a Ph.D. degrees in electrical engineering from the University of Toledo, Toledo, OH, USA.

Dr. Shin is a senior member of engineering staff in the Switching Systems Department of Electronics and Telecommunications Research Institute, Daejeon Korea. His research interests include parallel and distributed computing, switching systems and neural networks.