Neural Comput & Applic (1998)7:260–272
© 1998 Springer-Verlag London Limited

Neural
Computing
& Applications

Neocognitron with Improved Bend-Extractors: Recognition of Handwritten Digits in the Real World

K. Fukushima., E. Kimura and H. Shouno

Graduate School of Engineering Science, Osaka University, Toyonaka, Osaka, Japan

We have reported previously that the performance of a neocognitron can be improved by a built-in bend-extracting layer. The conventional bend-extracting layer can detect bend points and end points of lines correctly, but not always crossing points of lines. This paper shows that an introduction of a mechanism of disinhibition can make the bend-extracting layer detect not only bend points and end points, but also crossing points of lines correctly. This paper also demonstrates that a neocognitron with this improved bend-extracting layer can recognise handwritten digits in the real world with a recognition rate of about 98%. We use the technique of dual thresholds for feature-extracting S-cells, and higher threshold values are used in the learning than in the recognition phase. We discuss how the threshold values affect the recognition rate.

Keywords: Bend-extraction; Disinhibition; Handwritten digits; Neocognition; Neural network model; Pattern recognition; Vision

1. Introduction

Extraction of appropriate local features is very important for the robust recognition of visual patterns. End points, bend points and crossing points of lines are examples of such important features. This paper offers a new network architecture for the bend-extracting cells, and proposes an improved neocognitron that has this new bend-extracting cells. Incidentally, the neocognitron [1,2] is an Artificial Neural Network (ANN) that has the ability to recognise visual patterns robustly. It has a hierarchical multi-layered architecture similar to the classical hierarchy hypothesis by Hubel and Wiesel [3].

We have reported previously that the performance of a neocognitron can be improved by a built-in bend-extracting layer, which is placed after line-extracting layer [4]. The conventional bend-extracting layer can detect bend points and end points of lines correctly, but not always crossing points of lines. This paper shows that an introduction of a mechanism of disinhibition can make the bend-extracting layer detect not only bend points and end points, but also crossing points of lines correctly.

We then train a neocognitron with this new bend-extracting layer by unsupervised learning to recognise handwritten digits in the real world. Generally speaking, the neural networks' ability to robustly recognise patterns is influenced by the selectivity of feature-extracting cells in the networks. This selectivity can be controlled by the threshold values of the cells [5]. Fukushima and Tanigawa [6] have proposed previously to use higher threshold values for feature-extracting cells in the learning than in the recognition phase, when an unsupervised learning with a winner-take-all process is used to train neural networks. We apply this technique of dual thresholds to the neocognitron, and discuss how the thresholds affect the recognition rate.

We modify the learning method for the cells in the highest stage, in order to reconcile the unsupervised learning with the use of information of the category names of the training patterns [7,8].

We then show that the new neocognitron can acquire a recognition rate of about 98% for handwritten digits in a large scale real-world database ETL-1, without making any preprocessing like normalising scale and deformation of input patterns. ETL-1 is published by the Electrotechnical Labora-

Correspondence and offprint requests to: Kunihiko Fukushima, Graduate School of Engineering Science, Osaka University, Toyonaka, Osaka 560–8531, Japan.

tory in Japan, and contains handwritten digits freely written by many different persons.

2. Standard Neocognitron

2.1. Network Architecture

The neocognitron [1,2] is an ANN with a hierarchical multi-layered architecture similar to the classical hierarchy hypothesis by Hubel and Wiesel [3].

We first briefly explain the architecture of a standard neocognitron. The neocognitron has a multilayered architecture, as shown in Fig. 1, in which each rectangle represents a two-dimensional array of cells. Each cell receives its input connections from only a limited number of cells situated in a small area on the preceding layer. The density of cells in each layer is designed to decrease with the order of the stage.

The lowest stage of the hierarchical network is an input layer U_0, consisting of a two-dimensional array of photoreceptor cells. Each succeeding stage has a layer consisting of 'S-cells' followed by another layer consisting of 'C-cells'. Thus, in the whole network, layers of S-cells and C-cells are arranged alternately. We use the notation U_{Sl} and U_{Cl} to represent the S-cell layer and the C-cell layer of the lth stage, respectively.

Each layer of S-cells or C-cells is divided into sublayers, called 'cell-planes', according to the features (for example, line orientations) to which they respond. The cells in each cell-plane are arranged in a two-dimensional array. Each rectangle drawn with heavy lines in Fig. 1 represents a cell-plane. The connections converging to the cells in a cell-plane are homogeneous and topographically ordered. In other words, the connections have a translational symmetry, such that each of the cells of a cell-plane shares the same set of input connections. This condition of translational symmetry holds for both fixed and variable connections. The modification of variable connections is always done under this condition.

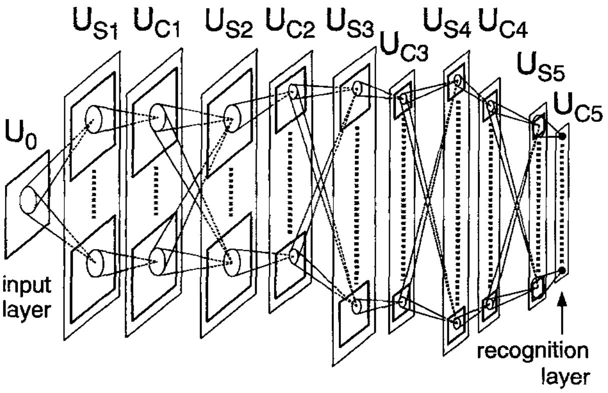

Fig. 1. The architecture of a typical neocognitron.

S-cells are feature-extracting cells. They resemble simple cells in the visual cortex in their response. Connections converging to these cells are modified by learning. After learning, S-cells are able to extract features from input patterns. In other words, an S-cell is activated only when a particular feature is presented in its receptive field. The features extracted by the S-cells are determined during the learning process. Generally speaking, local features, such as lines in particular orientations, are extracted in lower stages. More 'global' features, such as parts of a training pattern, are extracted in higher stages.

C-cells, which resembles complex cells in the visual cortex, are inserted in the network to allow for positional errors in the features of the stimulus. The connections from S-cells to C-cells are fixed and invariable. Each C-cell receives signals from a group of S-cells that extract the same feature, but from slightly different positions. The C-cell is activated if at least one of these S-cells is active. Even if the stimulus feature is shifted in position and another S-cell is activated instead of the first one, the same C-cell keeps responding. Hence, the C-cell's response is less sensitive to shifts in the position of the input pattern. In another interpretation, we can also express that C-cells make a blurring operation: the output of an S-cell layer is blurred spatially and appears in the succeeding C-cell layer. Shift-invariance in the response of a layer can be greatly increased by the blurring operation.

Incidentally, importance of blurring operation for robust pattern recognition has been reported from several research groups. For example, LeCun et al. used a hierarchical network to recognise handwritten zip code recognition [9]. Their network, like a neocognitron, uses a blurring operation to accept deformation, but is trained by backpropagation.

Thus, in the hierarchical network of the neocognitron, the process of feature-extraction by the S-cells and toleration of positional shift by the C-cells is repeated. The C-cells of the highest stage work as recognition cells: the response of the C-cells of the highest stage represents the final result of pattern recognition by the neocognitron.

2.2. Principles of Deformation-Resistant Recognition

In the whole network, with its alternate layers of S-cells and C-cells, the process of feature-extraction by the S-cells and toleration of positional shift by the C-cells is repeated. During this process, local features extracted in lower stages are gradually integrated into more 'global' features. Finally, each C-cell of the recognition layer at the highest stage integrates all the information of the input pattern, and responds only to one specific pattern.

Tolerating positional error a little at a time at each stage, rather than all in one step, plays an important role in endowing the network with the ability to recognise even distorted patterns. Figure 2 illustrates this situation. Let an S-cell in an intermediate stage of the network have already been trained to extract a global feature consisting of three local features of a training pattern 'A', as shown in Fig. 2(a). Because of the function of the C-cells before it, the S-cell tolerates a positional error of each local feature if the deviation falls within the dotted circle. Hence, the S-cell responds to any of the deformed patterns shown in Fig. 2(b). The toleration of positional errors should not be too large at this stage. If large errors are tolerated at any one step, the network may come to respond erroneously, such as by recognising a stimulus like Fig. 2(c) as an 'A' pattern.

Since errors in the relative position of local features are thus tolerated in the process of extracting and integrating features, the same C-cell responds in the recognition layer at the highest stage, even if the input pattern is deformed, changed in size, or shifted in position.

2.3. Unsupervised Learning

The neocognitron can be trained to recognise patterns through learning. During the learning, connections converging to S-cells are modified, and S-cells come to extract features from input patterns. Various training methods have been proposed, but they are, in principle, modifications from an unsupervised learning, which will be explained below.

In the case of the basic unsupervised learning, the self-organisation of the network is performed using two principles.

2.3.1. Winner-Take-All. The first principle is a kind of 'winner-take-all' rule: among the cells situated in a certain small area, only the one responding most strongly has its input connections reinforced. The amount of reinforcement of each input connection to this maximum-output cell (i.e. the winner) is proportional to the intensity of the response of the cell from which the relevant connection leads.

Figure 3 illustrates the connections converging to an S-cell. The S-cell receives variable excitatory connections from a group of C-cells of the preceding stage. The S-cell also receives a variable inhibitory connection from an inhibitory cell, called a V-cell. The V-cell receives fixed excitatory connections from the same group of C-cells, as does the S-cell, and always responds with the average intensity of the output of the C-cells.

The initial strength of the variable connections is very weak and nearly zero. Suppose the S-cell responds most strongly of the S-cells in its vicinity when a training stimulus is presented. According to the winner-take-all rule described above, variable connections leading from activated C- and V-cells are reinforced. Mathematical description of this process will appear in Section 2.4. The variable excitatory connections to the S-cell grow into a 'template' that exactly matches the spatial distribution of the response of the C-cells in the preceding layer. The inhibitory variable connection from the V-cell is also reinforced at the same time, but not strongly, because the output of the V-cell is not as large.

After the learning, the S-cell acquires the ability to extract a feature of the stimulus presented during the learning period. Through the excitatory connections, the S-cell receives signals indicating the existence of the relevant feature to be extracted. If an irrelevant feature is presented, the inhibitory signal from the V-cell becomes stronger than the direct excitatory signals from the C-cells, and the response of the S-cell is suppressed.

Once an S-cell is thus selected and reinforced to respond to a feature, the cell usually loses its responsiveness to other features. When a different feature is presented, a different cell usually yields

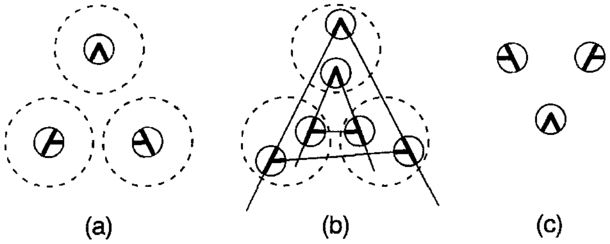

Fig. 2. Principle for recognising deformed patterns.

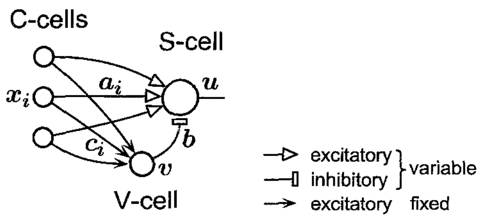

Fig. 3. The connections converging to an S-cell.

the maximum output and has its input connections reinforced. Thus, a 'division of labour' among the cells occurs automatically.

2.3.2. Seed Cell. The second principle for the learning is introduced so that the connections being reinforced always preserve translational symmetry. The maximum-output cell not only grows by itself, but also controls the growth of neighbouring cells, working, so to speak, like a seed in crystal growth. To be more specific, all of the other S-cells in the cell-plane, from which the 'seed cell' is selected, follow the seed cell and have their input connections reinforced by having the same spatial distribution as those of the seed cell. Hence, all the S-cells in the cell-plane come to respond selectively to a particular feature, and differences between these cells arise only from differences in position of the feature to be extracted. This rule might sound complicated, but it can be realised very easily in computer programming. Since all the cells of each cell-plane share the same set of input connections, modification of input connections to a seed cell automatically modifies the input connections to all cells in the cell-plane in the same manner as the seed cell.

We now explain the process of selecting seed cells in more detail. Competition among S-cells is performed within each 'hypercolumn' in the layer. Incidentally, a hypercolumn is defined as a group of S-cells in a layer whose receptive fields are situated approximately at the same location [1]. In other words, each hypercolumn contains all kinds of feature extracting cells (i.e. cells from all the cell-planes) in it, and these cells extract features from approximately the same place in the input layer.

Now, let a training pattern be presented to the input layer of the network. The response of the C-cells of the preceding stage is calculated from this training pattern. From each hypercolumn, the S-cell which happens to respond most strongly is chosen as a candidate for seed cells. When two candidates or more appear in one and the same cell-plane, only the one whose response is the largest is selected as the seed cell for that cell-plane. When only one candidate appears in a cell-plane, the candidate automatically becomes the seed cell for that cell-plane. If no candidate appears in a cell-plane, no seed cell is selected from that cell-plane this time. Thus, at most, one seed cell is selected from each cell-plane at a time. Usually, a different cell becomes a seed cell when a different training pattern is given.

2.4. Response of an S-cell

This section analyses the response of an S-cell and discusses how the threshold affects the selectivity for feature extraction by an S-cell (Fig. 3). Let a_i be the strength of the excitatory variable connection to an S-cell from the ith C-cell, whose output is x_i, and b be the inhibitory variable connection from the V-cell, whose output is v. Also let c_i be the strength of the fixed excitatory connection to the inhibitory V-cell from the ith C-cell.

Mathematically, the output of the S-cell is given by

$$u = \frac{\theta}{1-\theta}\,\varphi\left[\frac{1+\sum\limits_i a_i x_i}{1+\theta b v} - 1\right] \tag{1}$$

where $\varphi[\ ]$ is a function defined by $\varphi[x] = \max(x, 0)$. θ is a constant $(0 < \theta < 1)$ determining the threshold of the S-cell. (In our previous papers [1,2,5], parameter r is used instead of θ, where $\theta = r/(1+r)$). Variable v represents the output of the V-cell:

$$v = \sqrt{\sum_i c_i \{x_i\}^2} \tag{2}$$

We use vector notation $\boldsymbol{x}$ to represent the response of the C-cells, x_i. Similarly, vectors $\boldsymbol{a}$ and $\boldsymbol{c}$ are used to represent connections a_i and c_i. We define a weighted inner product of two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ by $(\boldsymbol{x},\boldsymbol{y}) = \Sigma_i c_i x_i y_i$, where the strength of the connections converging to the inhibitory V-cell is used as the weighting vector $\boldsymbol{c}$. We also define the norm of a vector $\boldsymbol{x}$ by $\|\boldsymbol{x}\| = \sqrt{(\boldsymbol{x},\boldsymbol{x})}$. It should be noted that we have $v = \|\boldsymbol{x}\|$ from Eq. (2).

During the learning phase, a large number of training patterns are presented to the neocognitron, but only a portion of them makes this particular S-cell a winner (or a 'seed cell'). The vector $\boldsymbol{x}$ that makes this S-cell a winner is called the training vector for this S-cell, and the nth training vector for this S-cell is represented by $\boldsymbol{x}^{(n)}$. The vector sum of all training vectors for the S-cell is called the reference vector of the cell and is denoted by $\boldsymbol{X}$. That is, $\boldsymbol{X} = \Sigma_n \boldsymbol{x}^{(n)}$. To simplify the discussion, we assume here, without loosing generality, that the same S-cell is always selected as the seed cell from the cell-plane.

Every time when the S-cell becomes a winner, the strength of the excitatory connection a_i is reinforced by the amount

$$\Delta a_i = q\, c_i\, x_i \tag{3}$$

where q is a positive constant determining the speed of reinforcement. Hence, a_i after finishing the learning becomes

$$a_i = q\, c_i\, X_i \tag{4}$$

Although two different methods have been proposed for determining inhibitory connection b, we introduce here a new method [7], in which b is determined directly from the values of the excitatory connections a_i. That is,

$$\begin{aligned} b &= \sqrt{\sum_i \frac{\{a_i\}^2}{c_i}} \\ &= \sqrt{\sum_i c_i \{X_i\}^2} \\ &= q\|X\| \end{aligned} \tag{5}$$

(Incidentally, in the old method [1,2,5], b is reinforced in the same manner as the excitatory connections. That is, $\Delta b = qv = q\ \|x^{(n)}\|$. Hence, $b = q\ \Sigma_n \|x^{(n)}\| = q\ \|X\| / \lambda$, where $\lambda = \|X\| / \{\Sigma_i \|x^{(n)}\|\} \leq 1$.)

By substituting Eqs (4) and (5) in Eq. (1), the response u of the S-cell to an arbitrary test vector x can be described as follows [7,5]:

$$u = \frac{\alpha}{1-\theta}\, \varphi[s-\theta] \tag{6}$$

where

$$s = \frac{(X, x)}{\|X\| \cdot \|x\|} \tag{7}$$

α is a variable defined by $\alpha = \theta bv/(1 + \theta bv)$, but can be considered as a constant (≈ 1) after some progress of learning, in which the inhibitory connection b has become large enough to satisfy $\theta bv \gg 1$. If the input to the S-cell is completely zero ($x = 0$), however, we have $\alpha = 0$. Incidentally, parameters α in this paper corresponds to $\gamma / (1 + r)$ in Fukushima [5]. (If the network has been trained by the old method, Eq. (6) has to be replaced by $u = \{\alpha / (1 - \theta)\}\ \varphi[\lambda s - \theta]$. Qualitatively, the difference between the two is not so large, because we usually have $\lambda \approx 1$ after finishing the learning.)

s represents a kind of similarity between the test vector x and the reference vector X. The S-cell yields a non-zero output for $s > \theta$. In the vector space of x, the conical area determined by $s > \theta$ around the reference vector X becomes the 'tolerance area' of the S-cell. The S-cell responds if and only if the test vector x falls in the tolerance area. The higher the threshold of the cell is, the smaller the tolerance area becomes.

2.5. Seed-Selecting Plane

Although the training of the neocognitron can be started simultaneously in all stages of the hierarchical network, we usually process it sequentially from lower stages to higher stages, for the sake of efficiency of the training in computer programming. After the training of lower stages has been completely finished, the training of the succeeding stage begins. The same set of training patterns is used for the training of all the stages. We now discuss the process of training a certain cell-plane when the training has been finished up to the preceding stage.

For the sake of efficient coding of computer program, a technique is used for selecting seed cells. We introduced a virtual cell-plane, called a 'seed-selecting plane'. Each layer of S-cells has one such plane. The plane represents all of the cell-planes which have not been reinforced yet. Each cell of the plane has very weak and diffused excitatory input connections from the C-cells in the preceding stage. (These weak input connections, which are fixed, correspond to the S-cell's initial input connections, which appear only when self-organisation is going to start in the conventional unsupervised learning.) Besides these weak input connections, each cell of the seed-selecting plane receives inhibitory connections from S-cells of the same layer. The initial values of the input connections of S-cells in all cell-planes other than the seed-selecting plane are zero.

The process of selecting seed cells progresses as follows (Fig. 4). A training pattern is presented to the input layer, and the response of the C-cells of the preceding stage is calculated. This can be done easily because the learning has already been finished up to the preceding stage. Using the response of these C-cells, the response of the S-cells of this layer are calculated. In the example of Fig. 4, cell-planes $k = 1$ through 3 have already been created in this layer, and S-cells in cell-planes $k = 1$ and 3 have responded to this training pattern ($t = 1$). (Of course, no response is elicited from any S-cells if this is the first presentation of training patterns to this layer.) Through competition among S-cells within each hypercolumn, seed-cells are selected. They are the S-cells marked with × in Fig. 4 ($t = 1$). The input connections of the cell-planes, from which seed cells are selected, are reinforced through the process discussed in Section 2.3.

Then, the responses of the cells of the seed-

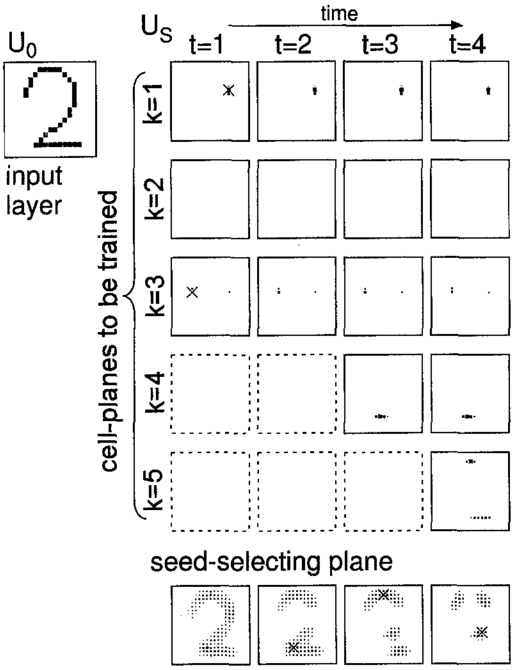

Fig. 4. Process of selecting seed cells using the seed-selecting plane.

selecting planes are calculated for the same training pattern. Because of the inhibitory connections from S-cells, the response of the seed-selecting plane is suppressed around the places where S-cells of other cell-planes are strongly responding ($t = 2$). The position of the maximum output cell in the seed-selecting plane is marked as the location of the seed cell to be selected. However, the seed cell is not actually chosen from the seed-selecting plane, but chosen from a new cell-plane. More specifically, among the cell-planes which have not been reinforced yet, the cell-plane with the smallest serial number is chosen to be reinforced (cell-plane $k = 4$ in Fig. 4). The input connections of the selected cell-plane are reinforced with the help of the seed cell, and the response of these S-cells, which now have the reinforced input connections, is calculated ($t = 3$). The response of the seed-selecting plane is further suppressed by the inhibition from the S-cells of this newly generated cell-plane $k = 4$. This process of selecting and reinforcing seed cells is continued under the presentation of the same training pattern, until the response (above a certain threshold value) of the seed-selecting plane disappears. After that, the same process of training is repeated for the next training pattern.

2.6. One-Shot Learning

Although the features that S-cells in each layer should extract can thus be determined automatically by unsupervised learning, manual determination of features is still useful in some layers for efficient training of the network, especially when we know what features should be extracted in the layers. For example, we know, from neurophysiological experiments, that the visual system of monkeys and cats always has edge- and line-extracting cells. We have shown by computer simulation that neocognitrons with edge- and line-extracting cells exhibit a large ability to recognise patterns [7,8,10].

When we explicitly know what features should be extracted, 'one-shot learning'[1] becomes useful for efficient training of the network.

We now explain the process of one-shot learning. Assume that we want to generate in a certain layer a cell-plane that extracts a particular feature. Since the training is made sequentially from lower stages to higher stages, the training has already been completely finished up to the preceding stage. The 'teacher' prepare a training pattern that contains the feature, and presents the training pattern to the input layer of the network. He then points out the location of the feature, and the cell whose receptive field centre coincides with the location of the feature takes the place of the seed cell of the layer. The other process of reinforcement is identical to that of the unsupervised learning, and occurs automatically. The speed of reinforcement of variable input connections of a cell (that is, the value of q in Eq. (3)) can be so large that the training of a cell (and hence the cell-plane) is completed by only a single presentation of a training pattern. Therefore, the only task that the 'teacher' has to do is to present training patterns to the input layer, and to point out the locations of the features that should be extracted.

3. Different Thresholds in Learning and Recognition

The neural networks' ability to robustly recognise patterns is influenced by the selectivity of feature-extracting cells, and the selectivity is controlled by the threshold θ of the cells. Fukushima and Tanigawa [6] have proposed previously to use higher threshold values for feature-extracting cells in the learning than in the recognition phase, when an

[1] The 'one-shot learning' was previously called 'supervised learning of the neocognitron' [1]. The naming 'supervised learning', however, sometimes lead to misunderstanding, as though it is a time-consuming process like the supervised learning used in backpropagation. Therefore, we use a new naming 'one-shot learning' in this paper.

Table 1. Effect of inappropriate threshold in competitive learning.

Threshold	Recognition phase	Learning phase
Too high	Reduced ability to generalise • oversensitive to deformation • oversensitive to thinning-out (Shifted versions of the same pattern may be classified as different.)	Excessive expansion of the network (too many feature-extracting cells)
	overfitting to the training patterns	
Too low	Reduced ability to discriminate (confusion of resembling patterns)	Lack of feature-extracting cells (insufficient number of cells generated)

unsupervised learning with a winner-take-all process is used to train neural networks.

This method of dual threshold is used for layers U_{S3} and U_{S4} of our neocognitron. We discuss here how the threshold values affect the ability of the neocognitron.

During the recognition phase, a too high threshold makes the neocognitron oversensitive to deformation, and reduces the generalisation ability. A better performance is achieved when the thresholds are set low enough to maintain the generalisation ability [6]. A bad effect of 'thinning-out' on shifted patterns, which will be discussed later, also emerges under the condition of a too high threshold. A too low threshold, however, is not good either, because it reduces the ability to discriminate similar patterns of different categories.

If a competitive learning with a winner-take all process is used to train the network, the thresholds in the learning phase should be kept higher than in the recognition phase. If the thresholds in the learning phase are made as low as in the recognition phase, a sufficient number of feature-extracting cells (or cell-planes) cannot be generated in the network because of the competition among the cells [6]. The reason can be explained as follows. Suppose a cell-plane has been created for a particular feature. Even if a deformed version of the feature, to which a new cell-plane is desired to be created, is presented as a second training feature, no cell-plane can be generated if a cell of the first cell-plane responds to the second training feature in the competition area. Therefore, the threshold of the S-cells has to be set high enough to prevent the first cell-plane from responding to the second training feature.

A too high threshold in the learning phase, however, expands the scale of the network more than necessary, because each deformed training pattern generates a different cell-plane. To be more specific, a new cell-plane is always generated if all S-cells in a competition area are silent in spite of non-zero inputs. This occurs when the training vector does not fall in the tolerance area of any of the S-cells in the competition area. If an infinite number of training vectors exist and are distributed randomly in the vector space, the number of cell-planes will increase until the vector space has been completely covered with tolerance areas of different S-cells. The increase in number of cell-planes expands the scale of the network, and makes a serious problem on the computation cost.

A network with a higher threshold also requires a higher density of cells in each cell-plane to suppress the bad effect of thinning-out. In the neocognitron, thinning-out of cells is performed between layers so that the density of cells decreases with the order of the stages. If there is no thinning-out operation, a shifted input pattern simply produces a shifted response. In the actual neocognitron, however, a shifted input pattern does not always produce the same response pattern, but generates a slightly different response pattern, because of the thinning-out. According to the design principle of the neocognitron, the deformation of the response caused by the shift is expected to be absorbed by the blurring operation by the C-cells. If the threshold of the S-cells of the succeeding stage is too high, however, they become oversensitive to the effect of thinning-out, and a shifted pattern comes to elicit responses from different cell-planes [8]. This does not only reduce the recognition rate of the neocognitron, but also increase the scale of the network, because each shifted training pattern will generate a new cell-plane during the learning. In any case, the expansion of the scale of the network is inevitable if the threshold during the learning is too high.

If the threshold is too high for both learning and recognition phases, a so-called overfitting to the training data occurs.

Table 1 summarises how the threshold values affect the performance of the neocognitron.

4. Neocognitron with a New Bend-Extracting Layer

4.1. Network Architecture

The neocognitron used in this paper has the architecture shown in Fig. 5. U_0 is the input layer. Layer U_{S1} consists of edge-extracting S-cells. The S-cells of layer U_{S2} extract line components using the edge information extracted in U_{S1}. These two layers are trained by one-shot learning. A detailed process of line extraction will be explained in Section 4.2. Layers U_{S1}, U_{C1} and U_{S2} consist of cell-planes of various preferred orientations.

Layer $U_{S2'}$, which is inserted after layers U_{S2}, is the new bend-extracting layer, which is the topic of this paper. C-cells of layer U_{C2} receives signals only from S-cells of layer $U_{S2'}$.

The S-cells in layers U_{S3}, U_{S4} and U_{S5} have variable input connections, which are modified through unsupervised learning discussed in Section 2.3.

Dual thresholds are used for S-cells, as explained in Section 3. In other words, the method of learning for these layers is, in principle, the same as the one used in our previous system [7,8]. The method of learning for U_{S5} is slightly different, and will be discussed in Section 6.

The total number of S- or C-cells (not counting inhibitory V-cells) in each layer is indicated at the bottom of Fig. 5. In each stage except the highest one, the number of cell-planes of the C-cell layer, K_{Cl}, is the same as that of the S-cell layer, K_{Sl}. The values of K_{S3}, K_{S4} and K_{S5} are determined by the learning. Layer U_{C5} at the highest stage is the recognition layer, representing the final result of pattern recognition. In this layer, each cell-plane contains only one C-cell, and K_{C5} is equal to 10, the number of categories of the patterns to be recognised.

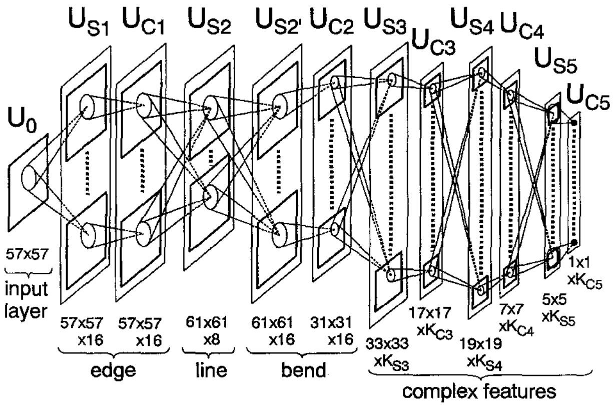

Fig. 5. The neocognitron used in the experiment.

4.2. Edge and Line Extraction

Layer U_{S1} is trained through one-shot learning to extract edges of various orientations. When we want to train a cell-plane, we first choose an arbitrary cell near the centre of the cell-plane, and determine it as the seed cell. We then present a straight edge stimulus of a particular orientation at a location near the centre of the receptive field (connectable area) of the seed cell, and train the cell-plane through the process of one-shot learning discussed in Section 2.6.

Sixteen different cell-planes are thus created in U_{S1}. In other words, preferred orientations of the cell-planes are chosen at an interval of 22.5°. The threshold (θ in Eq. (1)) of the S-cells of U_{S1} is set low enough to accept edges of slightly different orientations.

The output of layer U_{S1} is fed to layer U_{C1}, and a blurred version of the response of layer U_{S1} appears in layer U_{C1}.

The S-cells of layer U_{S2} extract line components using the edge information sent from layer U_{C1} [4]. Figure 6 illustrates the principle of line extraction. Layer U_{S1} extracts the edges on both side of a line. A change in thickness of a line causes a shift in position of both edges. However, the effect of the positional shift of the edges is absorbed by the

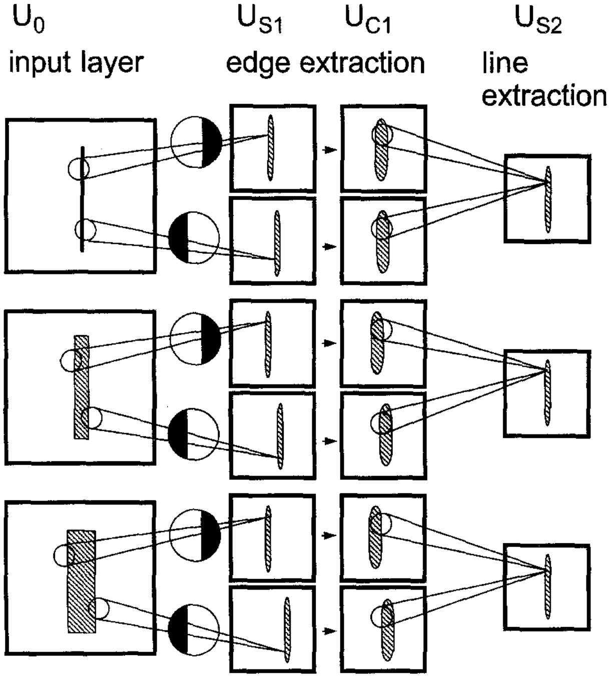

Fig. 6. The process of line extraction.

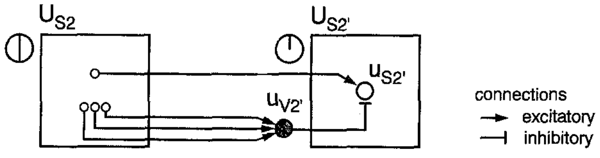

Fig. 7. The network related to a conventional bend-extracting cell.

blurring operation by the C-cells of U_{C1}. The S-cells of U_{S2} can then extract a line with little effect from the variation in thickness.

Layer U_{S2} is also trained by one-shot learning. A straight line of a particular orientation is used to train a cell-plane of U_{S2}. Only a single presentation of a line of a standard thickness is needed to train the cells to extract lines of different thickness. Eight cell-planes, whose preferred orientations are chosen at an interval of 22.5°, are thus created in U_{S2}. After finishing the training, the S-cells can flexibly extract line components of different thickness.

It is important to note here that network architecture illustrated in Fig. 6 need not be designed manually, but can be automatically generated in a standard neocognitron network through one-shot learning.

5. Bend Extraction

5.1. Conventional Bend-Extracting Cells

The bend-extracting cells installed in the conventional neocognitron [4,7,8] extract end points and bend points of lines in a similar way to the classical model of hypercomplex cells proposed by Hubel and Wiesel [3]. In other words, they resemble end-stopped cells in the physiological term. Figure 7 illustrates the network related to a conventional bend-extracting cell in a neocognitron. It shows only the connections converging to one bend-extracting cell $u_{S2'}$, which extracts the lower end of a vertical line. The cell $u_{S2'}$ is accompanied by an inhibitory cell $u_{V2'}$. U_{S2} in the figure shows a cell-plane consisting of cells u_{S2} that extract vertical line components. The cells are arranged retiontopically in each cell-plane.

If a straight vertical line is presented as shown in Fig. 8(a), cell $u_{S2'}$ shown in Fig. 7 is silent because the direct (monosynaptic) excitatory input is inhibited by the indirect (disynaptic) input through inhibitory cell $u_{V2'}$. If a line-end is presented at the location shown in Fig. 8(b), cell $u_{S2'}$ yields a large output because no inhibitory input comes from $u_{V2'}$. Cell $u_{S2'}$ also responds to a bend of a vertical line like Fig. 8(c). To this stimulus, the cells u_{S2} presynaptic to $u_{V2'}$ respond only weakly, because the orientation of the line is slightly different from the preferred orientation of the cells. As a result, the disynaptic inhibitory input to $u_{S2'}$ through $u_{V2'}$ becomes weaker than the monosynaptic excitatory input from cells u_{S2}. Therefore, the magnitude of the response of $u_{S2'}$ reflects the degree of bend or the curvature of the line.

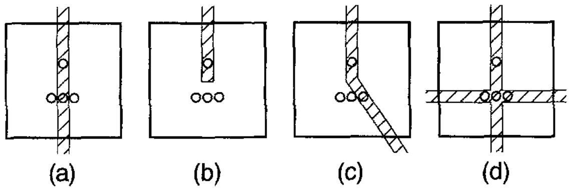

Fig. 8. Various stimuli presented to a bend-extracting cell.

When a line crossing is presented at the location shown in Fig. 8(d), however, the response of the cell $u_{S2'}$ becomes uncertain. The reason for the uncertainty can be explained as follows. In the cell-plane consisting of line-extracting cells u_{S2} of vertical preferred orientation, the cell at the crossing point sometimes responds to the vertical line-component of the cross, but sometimes not, because the cell's response is depressed by the horizontal line-component. The amount of depression is influenced by the strength (for example, thickness) of the horizontal line-component. Therefore, the output of the bend-extracting cell $u_{S2'}$ becomes either positive or zero, depending on the value of the response of the line-extracting cell u_{S2} at the crossing point.

5.2. New Bend-Extracting Cells

5.2.1. Principles of Bend Extraction. We propose here an improved version of the bend-extracting cell. It always responds to line crossing regardless of the angle of the crossing line. Of course, the cell responds also to the end and bend points of a line.

To get this type of response from the $u_{S2'}$ cell shown in Fig. 7, the inhibitory $u_{V2'}$ cell must be silent when a line-crossing is presented at the location shown in Fig. 8(d). In the new network, another inhibitory cell $u_{W2'}$, that inhibits the former inhibitory cell $u_{V2'}$, is introduced as shown in Fig. 9. Cell $u_{W2'}$ receives excitatory signals from line-extracting cells u_{S2} of all preferred orientations except vertical. This $u_{W2'}$ cell responds to all non-vertical lines. Once the $u_{W2'}$ cell responds to the horizontal line component of the cross, it inhibits the $u_{V2'}$ cell. As a result, the inhibition from $u_{V2'}$ cell is removed, and the $u_{S2'}$ cell becomes active receiving only monosynaptic excitatory input from the u_{S2} cell, which is responding to the vertical line-component just above the cross. In other words, the $u_{S2'}$ cell becomes active by disinhibition.

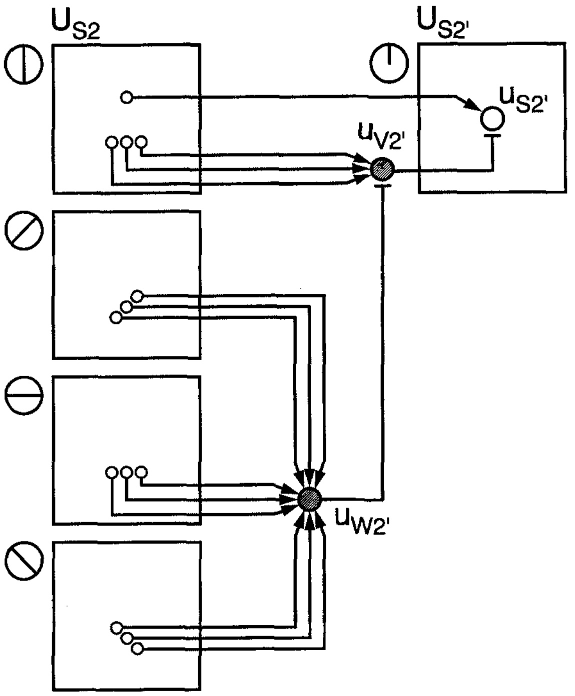

Fig. 9. The network related to a new bend-extracting cell.

Although the cell $u_{S2'}$ responds at a location near a line crossing, the response of the cell does not always mean the existence of a crossing. The $u_{S2'}$ cell also responds to the end and bend points of a line, and to a T-junction. A line-crossing has to be extracted by an S-cell of the succeeding stage U_{S3}. Near a line-crossing, bend-extracting cells of four different preferred orientations will become active in layer $U_{S2'}$ as shown in Fig. 10. These responses are transmitted to the C-cell layer U_{C2}, and are blurred. A cell-plane responding to this quadruplet is usually generated in U_{S3} by unsupervised learning. We can interpret that this cell-plane extracts line-crossings. Similarly, cell-planes extracting T-junctions, L- or V-shaped bends, curvatures, and also features created by a combination of three or more lines, are easily generated in U_{S3} if they are repeatedly presented during the learning phase.

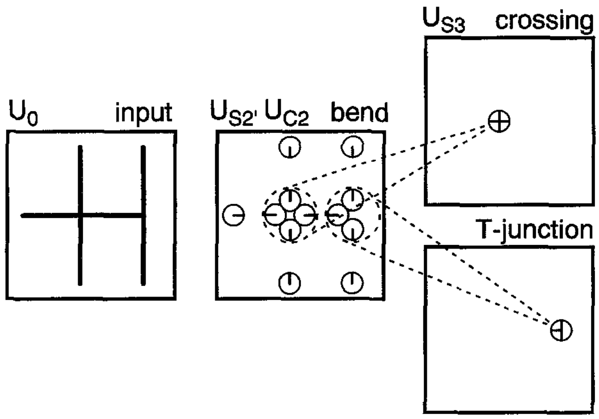

Fig. 10. The process of extracting line crossing and T-junction.

If we use a bend-extracting network of the old version, however, layer U_{S3} of the network shown in Fig. 5 cannot extract line-crossing stably, because the bend-extracting cells of $U_{S2'}$ do not always respond near the crossing point. Therefore, the conventional neocognitron has been designed in such a way that layer U_{S3} receives direct connections (via a C-cell layer) not only from bend-extracting layer $U_{S2'}$, but also from line-extracting layer U_{S2}. Although the direct signals from the line-extracting layer U_{S2} are useful for extracting line-crossing, they sometimes do harm to the robustness in the recognition when unsupervised learning is used.

Generally speaking, feature extracting cells that respond only to long straight lines (long-line cells) are not desirable for robust recognition of deformed patterns [10]. This makes the neocognitron oversensitive to change in size of the input patterns: A long-line cell that has been responding to a large input pattern might stop responding to a diminished version of the same pattern, because long-line cells do not usually respond to short lines. When we use supervised learning, we can easily prevent the generation of long-line cells in layer U_{S3} by not choosing seed cells in the middle of long straight lines [10]. If a competitive unsupervised learning is used, however, seed cells have a large tendency to be selected in a middle of a long straight line, and hence long-line cells are usually generated in layer U_{S3}.

If we use the new bend-extracting layer, direct connections from line-extracting layer U_{S2} to layer U_{S3} can be eliminated, because line crossing can be extracted stably without having these direct connections. Layer U_{S3} can thus receive signals only from bend-extracting layer $U_{S2'}$, which does not respond to middle points of long-lines. This can be expected to increase the robustness in pattern recognition.

5.2.2. Computer Simulation of Bend-Extraction. Figures 11 and 12 show examples of the response of a network with new bend-extracting layer. They display the response of input layer U_0, line-extracting layer U_{S2}, bend-extracting layer $U_{S2'}$, and layer U_{S3} of the network after finishing the learning. The mark enclosed in the circle beside each cell-plane indicates the preferred feature of the cell-plane.

Layer U_{S3} of this network has been trained with patterns ⊤, ⊣, ⊥, ⊢, ⋋, ⋎, ≺, ⋌, × and + through unsupervised learning. Dual thresholds are also used for layer U_{S3} in this simulation: a lower threshold value is used after finishing the learning. The thresholds are set to the optimal values for the recognition of handwritten digits, which will be discussed in Section 7.

Figure 11 shows the response to one of the training patterns. In layer U_{S2}, the input pattern is decomposed into two rectangular line-components: large responses are elicited from two cell-planes corresponding to the orientations of the lines contained in the input pattern. Small responses are seen also in the cell-planes of adjoining preferred orientations, because the tuning to line-orientation is set broad enough to ensure the tolerance for deformation.

In the bend-extracting layer $U_{S2'}$, we can see not only the responses to the four line-ends of the cross-shape pattern, but also large responses to the line-crossing at the centre, from which four radial line-components start. As in layer U_{S2}, small responses are seen also in the cell-planes of adjoining preferred orientations.

The response of layer U_{S3} shows that the line-crossing and four line-ends of the input pattern are correctly extracted. Although some faint response to the line-crossing are also seen in the cell-planes that extract T-junctions, the magnitude of these responses are much smaller than the response from the cell-plane that extract line-crossings. If required, these responses can be eliminated by the use of a slightly higher threshold. However, the threshold value as low as this gives a better performance in handwritten digit recognition, as discussed later. It can also be seen from the figure that troublesome long-line cells have not been generated in this layer.

Figure 12 shows the response of the same network to a deformed version of one of the training patterns. Although the response is not so strong, the T-junction and line-ends are correctly extracted in layer U_{S3}, irrespective of large deformations in the input pattern.

6. Learning Method for the Highest Stage

S-cells of U_{S5} are trained with a competitive learning method similar to that used to train U_{S3} and U_{S4}, but the category names of the training patterns are also used for the learning. Since the network learns varieties of deformed training patterns, more than one cell-plane for one category are usually generated in U_{S5}. Therefore, when each cell-plane of U_{S5} first learns a training pattern, the category name of the training pattern is assigned to the cell-plane. At the same time, connections are created from all S-cells of the cell-plane to the C-cell of that category name. Incidentally, layer U_{C5} has ten C-cells corresponding to the ten digits to be recognised. Each cell-plane of U_{S5} is thus determined to process exclusively one of the ten digits. Every time a training pattern is presented, the competition occurs only among the S-cells of the same category name as the training pattern. However, large the responses S-cells of different category names yield, they do not participate in the competition.

During the recognition phase, the maximum out-

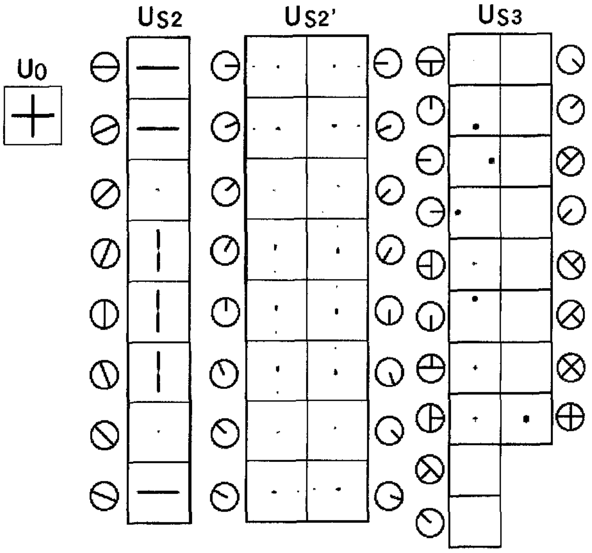

Fig. 11. The response of the network with the new bend-extracting layer. One of the patterns used to train layer U_{S3} is presented to the input layer.

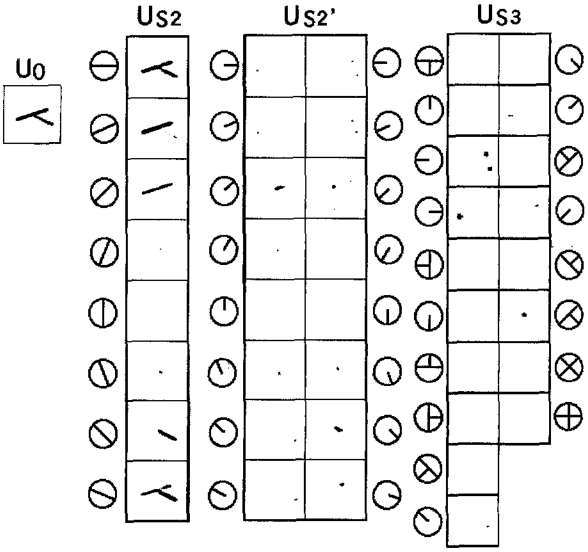

Fig. 12. The response of the network with the new bend-extracting layer. The input is a deformed version of a pattern used to train layer U_{S3}.

put S-cell of U_{S5} determines the final result of recognition by the neocognitron. Only one maximum output S-cell within the whole layer U_{S5} can transmit its output to U_{C5}.

7. Recognition of Handwritten Digits

To test the performance of the new bend-extracting cells, we trained the neocognitron discussed in Section 4 to recognise handwritten digits in the real world. We used a large-scale database (ETL-1) of handwritten digits freely written by many different persons. Figure 13 shows some examples of the characters in the ETL-1.

The network has the architecture shown in Fig. 5. The variable connections that are modified through unsupervised learning are the input connections to the S-cells of layers U_{S3}, U_{S4} and U_{S5}. Dual thresholds are used for these S-cells: a higher threshold is used for the learning phase than for the recognition phase.

Using handwritten digits from the ETL-1 database, we searched the optimum threshold values for the learning and the recognition phases. Threshold θ of the S-cells of the lth stage in the learning and the recognition phases are represented by θ_l^L and θ_l^R, respectively. We want to find the optimum values of θ_3^L, θ_3^R, θ_4^L, θ_4^R and θ_5^L. The value of θ_5^R does not cause any effect on the search process, because the final result of recognition is decided by the maximum-output S-cell of layer U_{S5}, and the sequence order of the magnitudes of the responses of the S-cells does not change with θ_5^R. The value of θ_5^R affects the rejection rate only.

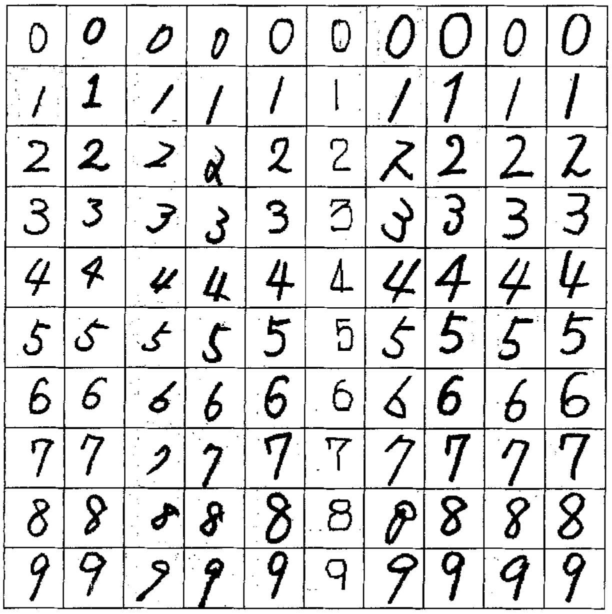

Fig. 13. Examples of handwritten characters contained in the ETL-1 database.

We chose 1000 patterns (100 patterns for each digit) randomly from the ETL-1 database, and used them as the training set. We also prepared a validation set of randomly chosen 1000 patterns, which does not overlap with the training set.

We measured the recognition rate for many different combinations of threshold values. More specifically, we trained the neocognitron using the training set and measured the recognition rate using the validation set for each combination of threshold values. During the learning phase, 1000 training patterns were presented five times for the training of each layer.

We repeated this process for many different combinations of threshold values, and searched the best combination of thresholds, which produces the maximum recognition rate for the validation set.

After having estimated the best combination of the threshold values using the validation set, we measured the recognition rate of the network using a test set consisting of randomly chosen 3000 patterns, which neither overlap with the training set nor with the validation set. Table 2 shows the recognition rate for the training, validation and test sets. We can see from this table that the recognition rate of 97.9% has been obtained for the blind test data that have not been presented during the learning phase. There is no rejection in the recognition. It should be noted here that any preprocessing for normalising scale and deformation of input patterns is not required at all for the neocognitron.

Figure 14 lists all the patterns that are erroneously recognised in this experiment. Figures 14(a), (b) and (c) show the misrecognised patterns in the learning, validation and test sets, respectively. The numeral in the lower right in each box indicates the result of misrecognition. Some of the errors shown in this figure seem to be inevitable. For example, patterns '9' with a style of writing like the one listed at the end of Fig. 14(c) is not common in the ETL-1 database. Most samples of pattern '9' have straight bar at the bottom instead of a curved hook. As a

Table 2. Recognition rate of the neocognitron with the optimal thresholds

	Recognition rate (%)	Error rate (%)
Training set	99.8	0.2
Validation set	98.2	1.8
Test set	97.9	2.1

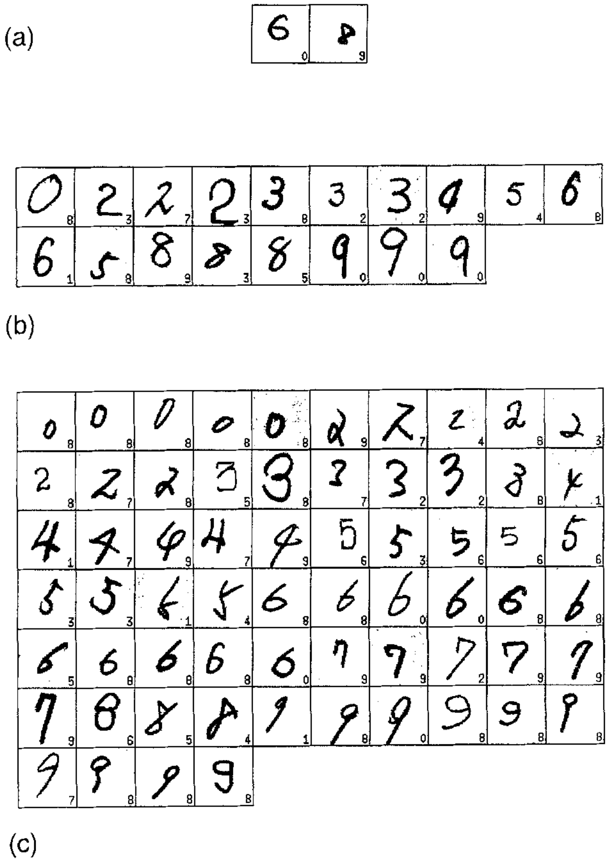

Fig. 14. Examples of handwritten characters contained in the ETL-1 database. (a) Misrecognised patterns in the learning set of 1000 patterns; (b) misrecognised patterns in the validation set of 1000 patterns; (c) misrecognised patterns in the test set of 3000 patterns.

result, it happened that no pattern '9' with a curved hook has been included in the randomly chosen 1000 learning patterns. Since the network has not seen '9' with a curved hook during the learning, the misrecognition will be unavoidable. Although we have used this learning set for evaluating the new system with the same criteria as for our previous systems, we expect to have a better score if we use another learning set, which has less bias in sampling.

8. Discussion

This paper proposed using a mechanism of disinhibition to improve the performance of bend-extracting network. The improved bend-extracting cells can detect not only bend points and end points, but also crossing points of lines correctly.

The paper also demonstrated that a neocognitron with these improved bend-extracting cells can recognise handwritten digits in the real world (ETL-1 database) with a recognition rate of about 98%. For the training of this neocognitron, we used the technique of dual thresholds: higher threshold values are used for the feature-extracting cells in the learning than in the recognition phase.

Acknowledgements. This work was supported in part by Grants-in-Aids #09308010 and #09268223 for Scientific Research from the Ministry of Education, Science, Sports and Culture of Japan.

References

1. Fukushima K. Neocognitron: a hierarchical neural network capable of visual pattern recognition. Neural Networks 1988; 1: 119–130.
2. Fukushima K, Miyake S. Neocognitron: a new algorithm for pattern recognition tolerant of deformations and shifts in position. Pattern Recognition 1982; 15: 455–469.
3. Hubel DH, Wiesel TN. Receptive fields and functional architecture in nonstriate areas (18 and 19) of the cat. J Neurophysiology 1965; 28: 229–289.
4. Fukushima K, Wake N. Improved neocognitron with bend-detecting cells. Int Joint Conf Neural Networks (IJCNN'92), Baltimore, vol IV. 1992; 190–195.
5. Fukushima K. Analysis of the process of visual pattern recognition by the neocognitron. Neural Networks 1989; 2: 413–420.
6. Fukushima K, Tanigawa M. Use of different thresholds in learning and recognition. Neurocomputing 1996; 11: 1–17.
7. Fukushima K, Nagahara K, Shouno H, Okada M. Training neocognitron to recognise handwritten digits in the real world. World Congress on Neural Networks (WCNN'96), San Diego, 1996; 21–24.
8. Fukushima K, Nagahara K, Shouno H. Training neocognitron to recognise handwritten digits in the real world. Proc Second Aizu Int Symposium on Parallel Algorithms/Architectures Synthesis (pAs'97), Aizu-Wakamatsu, Japan, 1997; 292–298.
9. LeCun Y, Boser B, Denker J S, Henderson D, Howard RE, Hubbard W, Jackel LD. Backpropagation applied to handwritten zip code recognition. Neural Computation 1989; 1: 541–551.
10. Fukushima K, Wake N. Handwritten alphanumeric character recognition by the neocognitron. IEEE Trans Neural Networks 1991; 2: 355–365.