Available at
www.ComputerScienceWeb.com
POWERED BY SCIENCE @ DIRECT®

ELSEVIER

Neurocomputing 51 (2003) 161–180

NEUROCOMPUTING

www.elsevier.com/locate/neucom

Neocognitron for handwritten digit recognition

Kunihiko Fukushima

Tokyo University of Technology, 1404-1, Ktakura, Hachioji, Tokyo 192-0982, Japan

Received 3 July 2001; received in revised form 17 April 2002; accepted 8 May 2002

Abstract

The author previously proposed a neural network model *neocognitron* for robust visual pattern recognition. This paper proposes an improved version of the neocognitron and demonstrates its ability using a large database of handwritten digits (ETL1).

To improve the recognition rate of the neocognitron, several modifications have been applied: such as, the inhibitory surround in the connections from S-cells to C-cells, contrast-extracting layer between input and edge-extracting layers, self-organization of line-extracting cells, supervised competitive learning at the highest stage, staggered arrangement of S- and C-cells, and so on. These modifications allowed the removal of accessory circuits that were appended to the previous versions, resulting in an improvement of recognition rate as well as simplification of the network architecture.

The recognition rate varies depending on the number of training patterns. When we used 3000 digits (300 patterns for each digit) for the learning, for example, the recognition rate was 98.6% for a blind test set (3000 digits), and 100% for the training set.

Keywords: Visual pattern recognition; Neural network model; Multi-layered network; Neocognitron; Handwritten digit

1. Introduction

The author previously proposed a neural network model *neocognitron* for robust visual pattern recognition [4,5]. It acquires the ability to recognize robustly visual patterns through learning. This paper proposes an improved version of the neocognitron and demonstrates its ability using a large database of handwritten digits.

The neocognitron was initially proposed as a neural network model of the visual system that has a hierarchical multilayered architecture similar to the classical

E-mail address: fukushima@karl.teu.ac.jp (K. Fukushima).

0925-2312/03/$ - see front matter © 2002 Elsevier Science B.V. All rights reserved.
PII: S0925-2312(02)00614-8

hypothesis of Hubel and Wiesel [12,13]. They hypothesized a hierarchical structure in the visual cortex: simple cells → complex cells → lower-order hypercomplex cells → higher-order hypercomplex cells. They also suggested that the relation between simple and complex cells resembles that between lower- and higher-order hypercomplex cells. Although physiologists do not use recently the classification of lower- and higher-order hypercomplex cells, hierarchical repetition of similar anatomical and functional architectures in the visual system still seems to be plausible from various physiological experiments.

The architecture of the neocognitron was initially suggested by these physiological findings. The neocognitron consists of layers of S-cells, which resemble simple cells, and layers of C-cells, which resemble complex cells. These layers of S- and C-cells are arranged alternately in a hierarchical manner. In other words, a number of modules, each of which consists of an S- and a C-cell layer, are connected in a cascade in the network.

S-cells are feature-extracting cells, whose input connections are variable and are modified through learning. C-cells, whose input connections are fixed and unmodified, exhibit an approximate invariance to the position of the stimuli presented within their receptive fields.

The C-cells in the highest stage work as recognition cells, which indicates the result of the pattern recognition. After learning, the neocognitron can recognize input patterns robustly, with little effect from deformation, change in size, or shift in position.

Varieties of modifications, extensions and applications of the neocognitron and related networks have been reported elsewhere [1,2,10,11,14,16–27].

This paper proposes a neocognitron of a new version, which shows a further improved performance. The new neocognitron also has a network architecture similar to that of the conventional neocognitron, but several new ideas have been introduced: such as, the inhibitory surround in the connections from S- to C-cells, a contrast-extracting layer followed by an edge-extracting layer, self-organization of line-extracting cells, supervised competitive learning at the highest stage, staggered arrangement of S- and C-cells, and so on.

This paper shows that these modifications allow the removal of accessory circuits appended to the neocognitron of recent versions [8], resulting in an improvement of recognition rate as well as simplification of the network architecture.

2. Architecture of the network

2.1. Outline of the network

Fig. 1 shows the architecture of the proposed network. (See also Fig. 6, which shows a typical response of the network.) As can be seen from the figure, the network has 4 stages of S- and C-cell layers: $U_0 \to U_G \to U_{S1} \to U_{C1} \to U_{S2} \to U_{C2} \to U_{S3} \to U_{C3} \to U_{S4} \to U_{C4}$.

The stimulus pattern is presented to the input layer (photoreceptor layer) U_0.

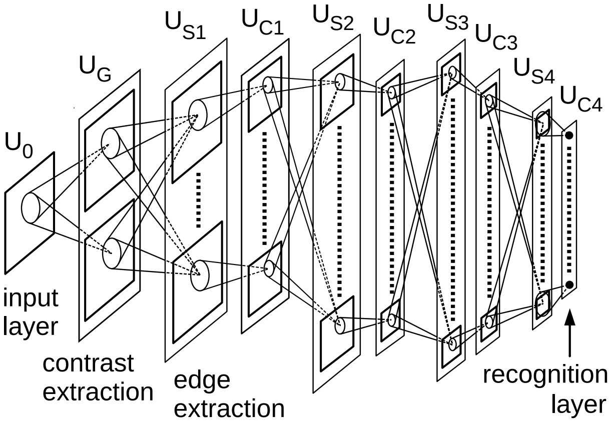


Fig. 1. The architecture of the proposed neocognitron.

A layer of contrast-extracting cells (U_G), which correspond to retinal ganglion cells or lateral geniculate nucleus cells, follows layer U_0. The contrast-extracting layer U_G consists of two cell-planes: one cell-plane consisting of cells with concentric on-center receptive fields, and one cell-plane consisting of cells with off-center receptive fields. The former cells extract positive contrast in brightness, whereas the latter extract negative contrast from the images presented to the input layer.

The output of layer U_G is sent to the S-cell layer of the first stage (U_{S1}). The S-cells of layer U_{S1} correspond to simple cells in the primary visual cortex. They have been trained using supervised learning [4] to extract edge components of various orientations from the input image.

The present model has four stages of S- and C-cell layers. The output of layer U_{Sl} (S-cell layer of the lth stage) is fed to layer U_{Cl}, where a blurred version of the response of layer U_{Sl} is generated. The connections from S- to C-cell layers will be discussed in detail in Sections 2.4 and 2.5. The density of the cells in each cell-plane is reduced between layers U_{Sl} and U_{Cl}.

The S-cells of the intermediate stages (U_{S2} and U_{S3}) are self-organized using unsupervised competitive learning similar to the method used by the conventional neocognitron. The learning method will be discussed later in Section 2.7.

Layer U_{C4}, which is the highest stage of the network, is the recognition layer, whose response shows the final result of pattern recognition by the network. Layer U_{S4} is trained to recognize all training patterns correctly through supervised competitive learning, which will be discussed in Section 2.8.

2.2. Contrast extraction

Let the output of a photoreceptor cell of input layer U_0 be $u_0(\boldsymbol{n})$, where $\boldsymbol{n}$ represent the location of the cell. The output of a contrast-extracting cell of layer U_G, whose receptive field center is located at $\boldsymbol{n}$, is given by

$$u_{\mathrm{G}}(\boldsymbol{n},k)=\varphi\left[(-1)^k\sum_{|\boldsymbol{v}|<A_{\mathrm{G}}}a_{\mathrm{G}}(\boldsymbol{v})u_0(\boldsymbol{n}+\boldsymbol{v})\right]\quad(k=1,2),\tag{1}$$

where $\varphi[\ ]$ is a function defined by $\varphi[x]=\max(x,0)$. Parameter $a_{\mathrm{G}}(\boldsymbol{\xi})$ represents the strength of fixed connections to the cell and takes the shape of a Mexican hat. Layer U_{G} has two cell-planes: one consisting of on-center cells ($k=2$) and one of off-center cells ($k=1$). A_{G} denotes the radius of summation range of $\boldsymbol{v}$, that is, the size of spatial spread of the input connections to a cell.

The input connections to a single cell of layer U_{G} are designed in such a way that their total sum is equal to zero. In other words, the connection $a_{\mathrm{G}}(\boldsymbol{\xi})$ is designed so as to satisfy

$$\sum_{|\boldsymbol{v}|<A_{\mathrm{G}}}a_{\mathrm{G}}(\boldsymbol{v})=0.\tag{2}$$

This means that the dc component of spatial frequency of the input pattern is eliminated in the contrast-extracting layer U_{G}. As a result, the output from layer U_{G} is zero in the area where the brightness of the input pattern is flat.

2.3. S-cell layers

We will first explain the characteristics commonly applied to all S-cell layers in the network. The characteristics that are specific to individual layers, especially the learning methods, will be discussed in later sections.

2.3.1. Response of an S-cell

Let $u_{\mathrm{S}l}(\boldsymbol{n},k)$ and $u_{\mathrm{C}l}(\boldsymbol{n},k)$ be the output of S- and C-cells of the kth cell-plane of the lth stage, respectively, where $\boldsymbol{n}$ represents the location of the receptive field center of the cells. Layer $U_{\mathrm{S}l}$ contains not only S-cells but also V-cells, whose output is represented by $v_l(\boldsymbol{n})$. The outputs of S- and V-cells are given by

$$u_{\mathrm{S}l}(\boldsymbol{n},k)=\frac{\theta_l}{1-\theta_l}\,\varphi\left[\frac{1+\sum_{\kappa=1}^{K_{\mathrm{C}l-1}}\sum_{|\boldsymbol{v}|<A_{\mathrm{S}l}}a_{\mathrm{S}l}(\boldsymbol{v},\kappa,k)u_{\mathrm{C}l-1}(\boldsymbol{n}+\boldsymbol{v},\kappa)}{1+\theta_l b_{\mathrm{S}l}(k)v_l(\boldsymbol{n})}-1\right],\tag{3}$$

$$v_l(\boldsymbol{n})=\sqrt{\sum_{\kappa=1}^{K_{\mathrm{C}l-1}}\sum_{|\boldsymbol{v}|<A_{\mathrm{S}l}}c_{\mathrm{S}l}(\boldsymbol{v})\{u_{\mathrm{C}l-1}(\boldsymbol{n}+\boldsymbol{v},\kappa)\}^2}.\tag{4}$$

Parameter $a_{\mathrm{S}l}(\boldsymbol{v},\kappa,k)$ ($\geqslant 0$) is the strength of variable excitatory connection coming from C-cell $u_{\mathrm{C}l-1}(\boldsymbol{n}+\boldsymbol{v},\kappa)$ of the preceding stage. It should be noted here that all cells in a cell-plane share the same set of input connections, hence $a_l(\boldsymbol{v},\kappa,k)$ is independent of $\boldsymbol{n}$. $A_{\mathrm{S}l}$ denotes the radius of summation range of $\boldsymbol{v}$, that is, the size of spatial spread of input connections to a particular S-cell. Parameter $b_l(k)$ ($\geqslant 0$) is the strength of variable inhibitory connection coming from the V-cell. Parameter $c_{sl}(\boldsymbol{v})$ represents the strength of the fixed excitatory connections to the V-cell, and is a monotonically decreasing function of $|\boldsymbol{v}|$. The positive constant θ_l is the threshold of the S-cell and determines the selectivity in extracting features.

In (3) and (4) for $l = 1$, $u_{Cl-1}(\boldsymbol{n}, k)$ stands for $u_G(\boldsymbol{n}, k)$, and we have $K_{Cl-1} = 2$.

It should be noted here that, although the pitch of cells in a cell-plane is the same for S- and C-cells, the locations of S- and C-cells are not necessarily exactly aligned. In some stages, they are staggered by half a pitch, as will be discussed in Section 2.4 and shown in Fig. 5 later. In these stages, if S-cells are located on an integer grid $\boldsymbol{n}$, the locations $\boldsymbol{v}$ of C-cells relative to an S-cell are at the centers of the meshes of the grid. More specifically, if we write $\boldsymbol{v} = (v_x, v_y)$ in Eqs. (3) and (4), v_x and v_y do not take integer values but take integers plus 0.5. In the stage where S- and C-cells are exactly aligned, v_x and v_y take integer values.

2.3.2. Input connections to an S-cell

Training of the network is performed from the lower stages to the higher stages: after the training of a lower stage has been completely finished, the training of the succeeding stage begins. The same set of training patterns is used for the training of all stages except layer U_{S1}.

Although the method for selecting seed cells during learning is slightly different between layers, the rule for strengthening variable connections $a_l(\boldsymbol{v}, \kappa, k)$ and $b_l(k)$ is the same for all layers, once the seed cells have been determined. They are strengthened depending on the responses of the presynaptic cells. The method of selecting seed cells will be discussed later. Let cell $u_{Sl}(\hat{\boldsymbol{n}}, \hat{k})$ be selected as a seed cell at a certain time, the variable connections $a_l(\boldsymbol{v}, \kappa, \hat{k})$ to this seed cell, and consequently to all the S-cells in the same cell-plane as the seed cell, are increased by the following amount:

$$\Delta a_{Sl}(\boldsymbol{v}, \kappa, \hat{k}) = q_l c_{Sl}(\boldsymbol{v}) u_{Cl-1}(\hat{\boldsymbol{n}} + \boldsymbol{v}, \kappa), \tag{5}$$

where q_l is a positive constant determining the learning speed. Although several methods have been proposed for determining the inhibitory connection $b_l(\hat{k})$, we use a method by which $b_l(\hat{k})$ is determined directly from the values of the excitatory connections $a_l(\boldsymbol{v}, \kappa, \hat{k})$ [6]. That is,

$$b_{Sl}(\hat{k}) = \sqrt{\sum_{\kappa=1}^{K_{Cl-1}} \sum_{|\boldsymbol{v}| < A_{Sl}} \frac{\{a_{Sl}(\boldsymbol{v}, \kappa, \hat{k})\}^2}{c_{Sl}(\boldsymbol{v})}}. \tag{6}$$

2.3.3. Vector notation

Now, let us represent the response of an S-cell $u_{Sl}(\boldsymbol{n}, k)$ using a vector notation. Let $\boldsymbol{x}$ be the vector representing input signals to the S-cell. In other words, the responses of the preceding C-cells $u_{Cl-1}(\boldsymbol{n} + \boldsymbol{v}, \kappa)$ for $(|\boldsymbol{v}| < A_{Sl})$ are the elements $x(\boldsymbol{v})$ of the vector $\boldsymbol{x}$.

Similarly, let $\boldsymbol{x}^{(i)}$ be the ith training vector to the S-cell. To be more exact, $\boldsymbol{x}^{(i)}$ be the training vector to the ith seed cell of the cell-plane to which the S-cell belongs.

Let $\boldsymbol{X}$ be the total sum of all training vectors to the cell-plane:

$$\boldsymbol{X} = \sum_i \boldsymbol{x}^{(i)}. \tag{7}$$

We will call $\boldsymbol{X}$ the reference vector of the S-cell (or the cell-plane).

We define a weighted inner product of two vectors $\boldsymbol{X}$ and $\boldsymbol{x}$ by

$$(\boldsymbol{X}, \boldsymbol{x}) = \sum_{|\boldsymbol{v}| < A_{\mathrm{S}l}} c_{\mathrm{S}l}(\boldsymbol{v}) X(\boldsymbol{v}) x(\boldsymbol{v}), \tag{8}$$

using $c_{\mathrm{S}l}(\boldsymbol{v})$, which represents the strength of fixed excitatory connections to a V-cell (see (4) and (5)). We also define the norm of a vector by $\|\boldsymbol{x}\| = \sqrt{(\boldsymbol{x}, \boldsymbol{x})}$.

Substituting (4), (5) and (6) in (3), and using the vector notation defined above, we have

$$u_{\mathrm{S}l}(\boldsymbol{n}, k) = \frac{\alpha}{1 - \theta_l} \varphi(s - \theta_l), \tag{9}$$

where

$$s = \frac{(\boldsymbol{X}, \boldsymbol{x})}{\|\boldsymbol{X}\| \cdot \|\boldsymbol{x}\|} \tag{10}$$

is the similarity between the reference vector $\boldsymbol{X}$ and the test vector $\boldsymbol{x}$. It can also be expressed as the cosine of the angle between two vectors in the multi-dimensional vector space. The variable α, which is defined by

$$\alpha = \frac{\theta_l b_{\mathrm{S}l}(k) v_l(\boldsymbol{n})}{1 + \theta_l b_{\mathrm{S}l}(k) v_l(\boldsymbol{n})}, \tag{11}$$

takes value $\alpha \approx 1$, if $v_l(\boldsymbol{n}) = \|\boldsymbol{x}\| \neq 0$ after finishing learning where $b_{\mathrm{S}l}(k)$ is large enough.

Therefore, we approximately have

$$u_{\mathrm{S}l}(\boldsymbol{n}, k) \approx \frac{\varphi(s - \theta_l)}{1 - \theta_l}. \tag{12}$$

This means that the response of an S-cell takes a maximum value approximately equal to 1 when the test vector is identical to the reference vector, and becomes 0 if the similarity s is less than the threshold θ_l of the cell.

In the multi-dimensional feature space, the area that satisfies $s < \theta_l$ becomes the tolerance area in feature extraction by the S-cell, and the threshold θ_l determines the radius of the tolerance area. The selectivity of an S-cell to its preferred feature (or the reference vector) can thus be controlled by the threshold θ_l.

If the threshold is low, the radius of the tolerance area becomes large, and the S-cell responds even to features largely deformed from the reference vector. This makes a situation like a population coding of features rather than grandmother cell theory: many S-cells respond to a single feature if the response of an entire layer is observed.

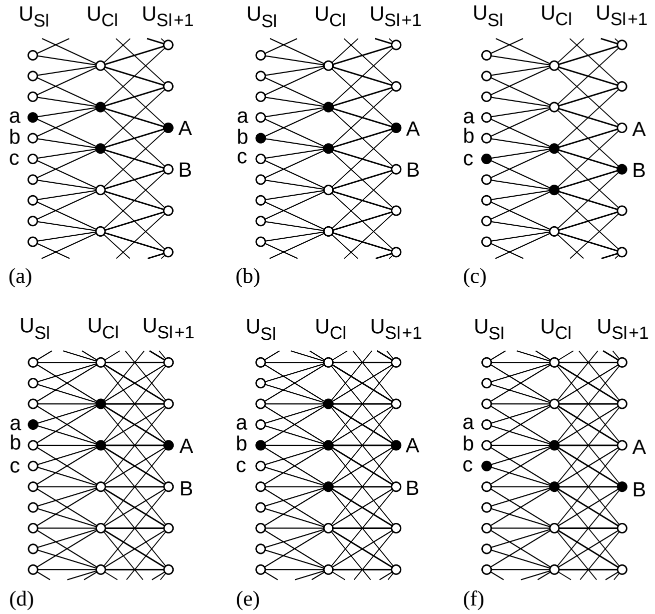


Fig. 2. Effect of thinning-out for two different arrangements of cells. S- and C-cells are staggered by half a pitch in (a)–(c), while they are exactly aligned in (d)–(f).

Empirically, this situation of low threshold produces a better recognition rate of the neocognitron.

2.4. Thinning-out of cells

The response of an S-cell layer U_{Sl} is spatially blurred in the succeeding C-cell layer U_{Cl}, and the density of the cells in each cell-plane is reduced. The reduction of the density is made by a thinning-out of the cells. In our system, we make 2:1 thinning-out in both horizontal and vertical directions. Since there are several kinds of possible thinning-out methods, we will pick up two of them and compare the merits and demerits. To simplify the discussion, Fig. 2 illustrates a one-dimensional case where the number of connections are reduced than in the actual network used for the simulation. In Method 1, the locations of S- and C-cells are staggered by half a pitch as illustrated in (a)–(c) in Fig. 2. In Method 2, S- and C-cells are exactly aligned as shown in (d)–(f). Suppose the number of input connections of each cell be 4 in Method 1, and 5 in Method 2.

Let cell a is active in layer U_{Sl} as shown in (a). The output of this S-cell is fed to two C-cells of the succeeding layer U_{Cl}. If the network has already finished learning, one S-cell in layer U_{Sl+1}, say cell A, will responds to this stimulus. Even if the input stimulus is shifted by 1 pixel and cell b is active instead of a as shown in (b), still the same C-cells will be active, and hence the same S-cell A will responds. Now let the stimulus be shifted by 2 pixels from (a), and cell c be active as shown in (c). The active C-cells will now be shifted by one pitch of C-cells. Since all the S-cells in a cell-plane share the same set of input connections, the response of S-cells will also be shifted in layer U_{Sl+1}, and cell B will be active. We can see that Method 1 causes no problem for this kind of shift.

Now we will consider Method 2. Let cell a is active in layer U_{Sl} as shown in (d). The output of cell a is fed to two C-cells, and cell A will be active in layer U_{Sl+1}. This situation resembles the case (a). If the input stimulus is shifted by 1 pixel and cell b is active, however, the signal will now be sent to three C-cells as shown in (e). To ensure the shift-invariance of the neocognitron, the same S-cell A has to respond to this shifted stimulus. In other words, the threshold of S-cells must be set low enough to allow S-cell A respond to both (d) and (e). If the stimulus be shifted by 2 pixels from (d), we can simply have a shifted response from cell B as shown in (f), and there is no problem as in the case of Method 1.

From these discussions, it might seem that Method 1 is always superior to Method 2. If two adjoining cells are active in layer U_{Sl}, however, Method 2 will produces a better result. Let cells a and b be active simultaneously. Two C-cells will be active by Method 1, and three by Method 2. When cells b and c are active together, three C-cells will be active by both Methods 1 and 2. We can see that the number of active C-cells now changes by Method 1, but not by Method 2.

Although each method has thus merits and demerits, we decide to adopt Method 1 in this paper, because a single feature seems to have a larger chance of eliciting a large response from one isolated S-cell than from two adjoining S-cells. Incidentally, in most of the neocognitrons of previous versions, Method 2 was used.

Relative arrangement of cells between U_{Cl} and U_{Sl+1}, however, makes little difference, even if they are staggered or exactly aligned. In the simulation discussed later, they are staggered in layers up to U_{S3}, and exactly aligned between U_{C3} and U_{S4}.

2.5. Inhibitory surround in the connections to C-cells

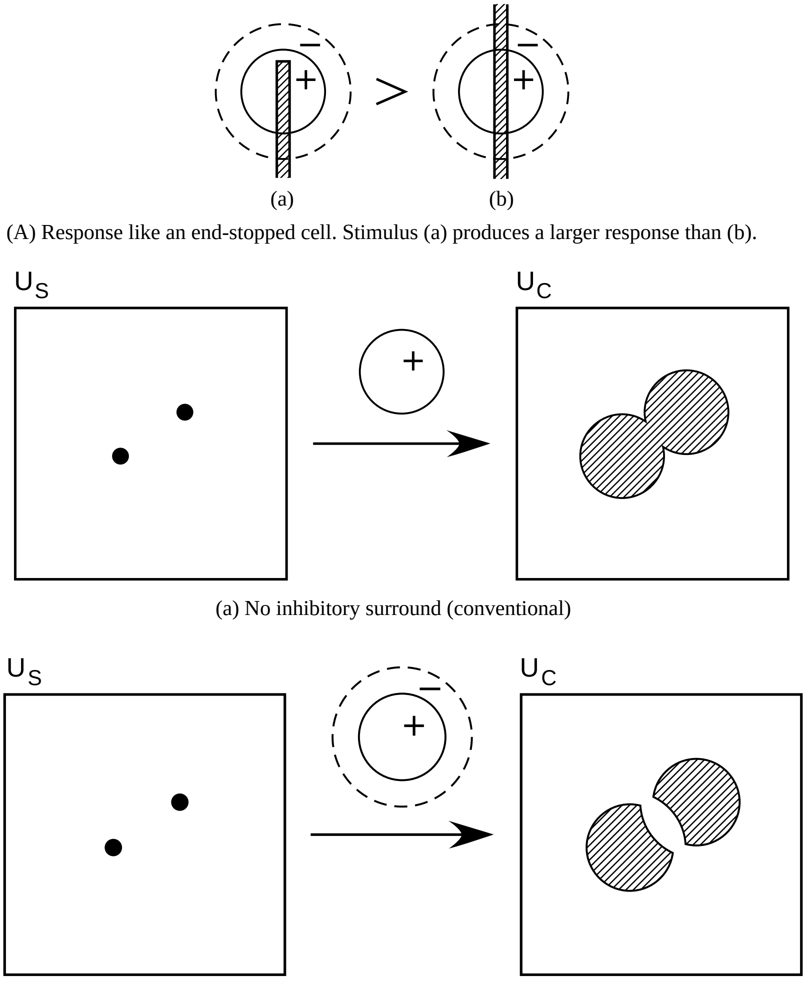
The response of an S-cell layer U_{Sl} is spatially blurred in the succeeding C-cell layer U_{Cl}. Mathematically, the response of a C-cell of U_{Cl}, excluding those in the highest stage U_{C4}, is given by

$$u_{Cl}(\boldsymbol{n},k) = \psi\left[\sum_{|\boldsymbol{v}|<A_{Cl}} a_{Cl}(\boldsymbol{v}) u_{Sl}(\boldsymbol{n}+\boldsymbol{v},k)\right], \tag{13}$$

where $\psi[x] = \varphi[x]/(1+\varphi[x])$. Parameter $a_{Cl}(\boldsymbol{v})$ represents the strength of the fixed excitatory connections converging from a group of S-cells, which spread within a radius of A_{Cl}.

In the conventional neocognitron, the input connections of a C-cell, namely $a_{Cl}(\boldsymbol{v})$, consisted of only excitatory components of a circular spatial distribution.

An inhibitory surround is newly introduced around the excitatory connections. The concentric inhibitory surround endows the C-cells with the characteristics of end-stopped



(a) (b)

(A) Response like an end-stopped cell. Stimulus (a) produces a larger response than (b).

(a) No inhibitory surround (conventional)

(b) Inhibitory surround (proposed)

(B) Separation of the blurred responses produced by two independent features.

Fig. 3. The effect of inhibitory surround in the input connections to a C-cell.

cells, and C-cells behave like hypercomplex cells in the visual cortex (Fig. 3(A)). In other words, an end of a line elicits a larger response from a C-cell than a middle point of the line.

Bend points and end points of lines are important features for pattern recognition. In the network of previous versions (e.g. [8]), an extra S-cell layer, which was called a bend-extracting layer, was placed after the line-extracting stage to extract these feature points. In the proposed network, C-cells, whose input connections have inhibitory surrounds, participate in extraction of bend points and end points of lines while they are making a blurring operation. This allows the removal of accessory layer of bend-extracting cells, resulting in a simplification of the network architecture and an increased recognition rate as well.

The inhibitory surrounds in the connections also have another benefit. The blurring operation by C-cells, which usually is effective for improving robustness against deformation of input patterns, sometimes makes it difficult to detect whether a lump of blurred response is generated by a single feature or by two independent features of the same kind. For example, a single line and a pair of parallel lines of a very narrow separation generate a similar response when they are blurred. The inhibitory surround in the connections to C-cells creates a non-responding zone between the two lumps of blurred responses (Fig. 3(B)). This silent zone makes the S-cells of the next stage easily detect the number of original features even after blurring.

Incidentally, the inhibitory surround has also an effect like lateral inhibition and increases the selectivity to features in the response of C-cells. As a result, the threshold of preceding S-cells has to be slightly lowered to keep the same selectivity in the response of the C-cell layer. In other words, the best threshold of the preceding S-cells becomes lower when inhibitory surround is introduced in the input connections of C-cells.

The inhibitory surround in the input connections, however, is introduced for layers U_{C1} and U_{C2}, but not for layers U_{C3} and higher. In these higher layers, an inhibitory surround seems to have little chance to display its real ability because of two reasons. (1) Since there is a little probability that a single input pattern has two identical global features at different locations, discrimination between one and two features is scarcely required. (2) Since spatial spread of the connections to a single cell becomes large enough to cover a great part of the preceding S-cell layer, most of the surrounding inputs come from outside of the boundary of a cell-plane and are treated as zero in the calculation.

2.6. Edge-extracting layer

Layer U_{S1}, namely, the S-cell layer of the first stage, is an edge-extracting layer. It has 16 cell-planes, each of which consists of edge-extracting cells of a particular preferred orientation. Preferred orientations of the cell-planes, namely, the orientations of the training patterns, are chosen at an interval of 22.5°. The threshold of the S-cells, which determines the selectivity in extracting features, is set low enough to accept edges of slightly different orientations.

The S-cells of this layer have been trained using supervised learning [4]. To train a cell-plane, the “teacher” presents a training pattern, namely a straight edge of a particular orientation, to the input layer of the network. The teacher then points out the location of the feature, which, in this particular case, can be an arbitrary point on the edge. The cell whose receptive field center coincides with the location of the feature takes the place of the seed cell of the cell-plane, and the process of reinforcement occurs automatically. It should be noted here that the process of supervised learning is identical to that of the unsupervised learning except the process of choosing seed cells.

The speed of reinforcement of variable input connections of a cell [i.e. the value of q_l in (5)] is set so large that the training of a seed cell (and hence the cell-plane) is completed by only a single presentation of each training pattern.

The optimal value of threshold θ_1 can be determined as follows. A low threshold θ_1 reduces the orientation selectivity of the S-cells and increases the tolerance for rotation of edges to be extracted. Computer simulation shows that a lower threshold usually produces a greater robustness against deformation of the input patterns. (Some discussions on the merit of low threshold appear also in [7].) If the threshold becomes too low, however, S-cells of this layer come to yield spurious outputs, responding to features other than desired edges. For example, an S-cell responds even to an edge of 180° apart from the preferred orientation, when the threshold is too low. Such spurious responses usually reduce the recognition rate of the neocognitron. Hence it can be concluded that the optimal value of the threshold θ_1 is the lower limit of the value that does not generate spurious responses from the cells.

The optimal threshold value, however, changes depending whether an inhibitory surround is introduced in the connections to the C-cells or not. The inhibitory surround produces an effect like a lateral inhibition, and small spurious responses generated in layer U_{S1} can be suppressed in layer U_{C1}. Hence the threshold θ_1 can be lowered down to the value by which no spurious responses are observed, not in U_{S1}, but in U_{C1}.

2.7. Competitive learning for intermediate layers

The S-cells of intermediate stages (U_{S2} and U_{S3}) are self-organized using unsupervised competitive learning similar to the method used in the conventional neocognitron [4,5]. Seed cells are determined by a kind of winner-take-all process. Every time a training pattern is presented to the input layer, each S-cell competes with the other cells in its vicinity, which is called the competition area and has the shape of a hypercolumn. If and only if the output of the cell is larger than any other cells in the competition area, the cell is selected as the seed cell. As can be seen from Eq. (5), each input connection to a seed cell is increased by an amount proportional to the response of the cell from which the connection leads. Because of the shared connections within each cell-plane, all cells in the cell-plane come to have the same set of input connections as the seed cell.

Line-extracting S-cells, which were created by supervised learning in the previous version [8], are now automatically generated (or, self-organized) together with cells extracting other features in the second stage (U_{S2}). The cells in this stage extract features using information of edges that are extracted in the preceding stage.

The neural networks' ability to recognize patterns robustly is influenced by the selectivity of feature-extracting cells, which is controlled by the threshold of the cells. Fukushima and Tanigawa [7] have proposed the use of higher threshold values for feature-extracting cells in the learning phase than in the recognition phase, when unsupervised learning with a winner-take-all process is used to train neural networks.

This method of dual threshold is used for the learning of layers U_{S2} and U_{S3}.

2.8. Learning method for the highest stage

S-cells of the highest stage (U_{S4}) are trained using a supervised competitive learning.[1] The learning rule resembles the competitive learning used to train U_{S2} and U_{S3}, but the class names of the training patterns are also utilized for the learning. When the network learns varieties of deformed training patterns through competitive learning, more than one cell-plane for one class is usually generated in U_{S4}. Therefore, when each cell-plane first learns a training pattern, the class name of the training pattern is assigned to the cell-plane. Thus, each cell-plane of U_{S4} has a label indicating one of the 10 digits.

Every time a training pattern is presented, competition occurs among all S-cells in the layer. (In other words, the competition area for layer U_{S4} is large enough to cover all cells of the layer.) If the winner of the competition has the same label as the training pattern, the winner becomes the seed cell and learns the training pattern in the same way as the seed cells of the lower stages. If the winner has a wrong label (or if all S-cells are silent), however, a new cell-plane is generated and is put a label of the class name of the training pattern.

During the recognition phase, the label of the maximum-output S-cell of U_{S4} determines the final result of recognition. We can also express this process of recognition as follows. Recognition layer U_{C4} has 10 C-cells corresponding to the 10 digits to be recognized. Every time a new cell-plane is generated in layer U_{S4} in the learning phase, excitatory connections are created from all S-cells of the cell-plane to the C-cell of that class name. Competition among S-cells occur also in the recognition phase, and only one maximum output S-cell within the whole layer U_{S4} can transmit its output to U_{C4}.

In the recognition phase, the threshold (θ_4^{R}) of U_{S4} is chosen so low that any input pattern usually elicits responses from several S-cells. Hence the process of finding the largest-output S-cell is equivalent to the process of finding the nearest reference vector in the multi-dimensional feature space. Each reference vector has its own territory determined by the Voronoi partition of the feature space. The recognition process in the highest stage resembles the vector quantization [9,15] in this sense.

To get a good recognition rate, the threshold in the learning phase, θ_4^{L}, can be set either very high or equal to θ_4^{R}, which is low. The behavior of the network differs, however, depending on whether $\theta_4^{\mathrm{L}} \gg \theta_4^{\mathrm{R}}$ or $\theta_4^{\mathrm{L}} = \theta_4^{\mathrm{R}}$.

If we take $\theta_4^{\mathrm{L}} \gg \theta_4^{\mathrm{R}}$, the learning time can be shorter, but a larger number of cell-planes K_{S4} are generated in the learning, resulting in a longer recognition time. Since the high threshold θ_4^{L} produces a small tolerance area around each reference vector, there is little chance that training vectors of different classes drop in the same tolerance area. It should be noted here that a training vector outside the tolerance area does not elicit any response from the cell. If a training vector drops in the territory of a reference vector, which does not extend outside its small tolerance area, and if the reference vector has the same class name as the training vector, the training vector is added

[1] Although the learning rule used in a previous version [6] was also called a supervised competitive learning, it is slightly different from the one discussed in this paper.

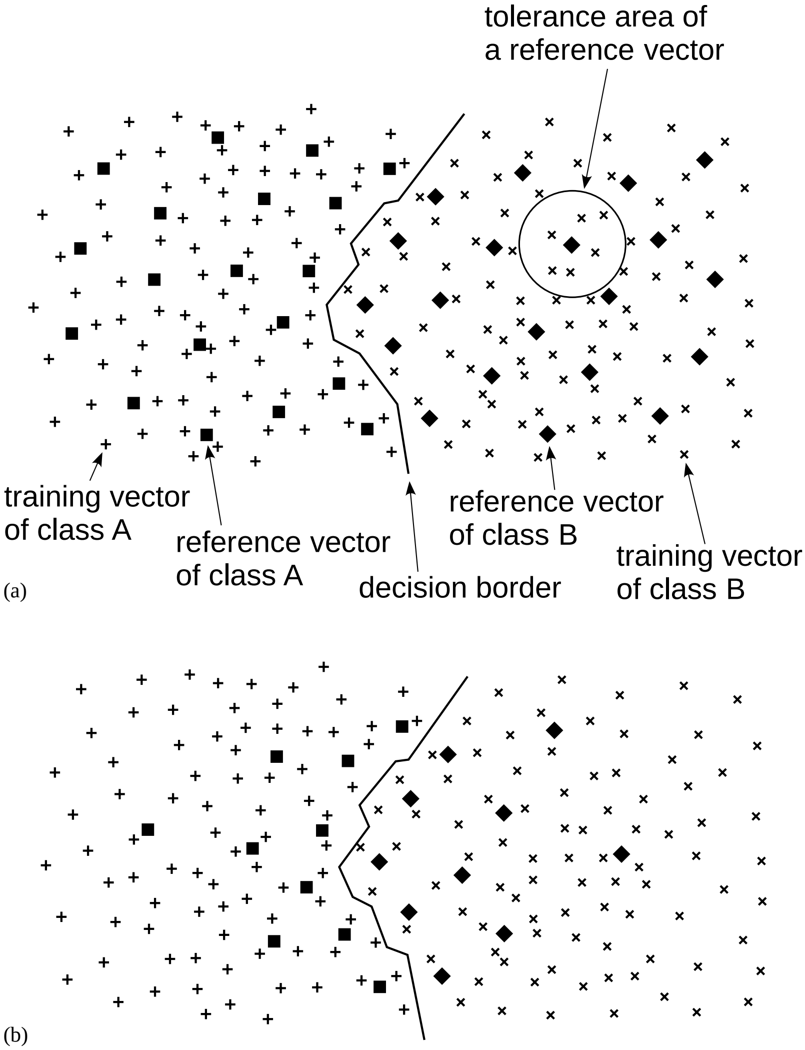


Fig. 4. Distribution of training and reference vectors in a multi-dimensional feature space. (a) A high threshold in the learning ($\theta_4^{\mathrm{L}} \gg \theta_4^{\mathrm{R}}$). (b) A low threshold in the learning ($\theta_4^{\mathrm{L}} = \theta_4^{\mathrm{R}}$).

to the reference vector. Since the reference vector is the vector sum of all training vectors presented in its territory, the territory gradually changes during the progress of learning, but the change is not so large in the case of a high threshold value. If the training vector does not drop within tolerance areas of any cells, a new reference vector (or cell-plane) is generated. Hence reference vectors will be distributed almost uniformly in the multi-dimensional feature space independent of the distance from the borders between classes (Fig. 4(a)). One presentation of each training vector is almost enough to finish the learning, but a large number of reference vectors (or cell-planes) are generated because the average distance between adjacent reference vectors becomes of the order of the radius of the small tolerance area.

On the other hand, if we take $\theta_4^{\mathrm{L}} = \theta_4^{\mathrm{R}}$, namely, a low threshold, the size of tolerance area of each reference vector become very large in the learning phase. As a result, training vectors that are distant from class borders can be represented by a small number of reference vectors. Training vectors that are misclassified in the learning phase usually come from near class borders. Suppose a particular training vector is misclassified in the learning. The reference vector of the winner, which caused a wrong recognition for this training vector, is not modified this time. A new cell-plane is generated instead, and the misclassified training vector is adopted as the reference vector of the new cell-plane. Generation of a new reference vector causes a shift of decision borders in the feature space, and some of the training vectors, which have been recognized correctly before, are now misclassified and additional reference vectors have to be generated again to readjust the borders. Hence a repeated presentation of the same set of training vectors is required before the learning converges. Thus, the decision borders are gradually adjusted to fit the real borders between classes. During this learning process, reference vectors come to be distributed more densely near the class borders. Since the density of the reference vectors is much lower in the locations distant from class borders, the number of reference vectors (or cell-planes) generated is smaller than in the case of $\theta_4^{\mathrm{L}} \gg \theta_4^{\mathrm{R}}$, although some of the reference vectors are still redundant.

Although a repeated presentation of a training pattern set is required before the learning converges when a low threshold is used, the required number of repetition is not so large in usual cases. In the computer simulation discussed below in the next section, only four rounds of presentation of the training set was enough. Every time when each round of presentation has finished in the learning phase, the number of newly generated cell-planes during that round, which is equal to the number of erroneously recognized patterns, is counted. If it reaches zero, the learning process ends.

This does not always guarantee that all training patterns will be recognized correctly after finishing the learning, because reference vectors drift slightly even during the last round of the presentation of the training set, where each training vector is summed to the reference vector of the winner. Erroneous recognition for the training patterns, however, occurs very seldom after finishing the learning. In usual situations, like the computer simulation shown in the next section, the recognition rate for the training set is 100%.

The recognition rate for unlearned patterns (blind test patterns) does not differ so much whether a very high or very low threshold is used in the learning. An intermediate value between them, however, usually produces a worse recognition rate. In the simulation discussed below in Section 3, we chose a low threshold ($\theta_4^{\mathrm{L}} = \theta_4^{\mathrm{R}}$) to get a smaller scale of the network and a shorter recognition time.

Another merit in choosing a low threshold is that, when designing the network, we need not be serious in determining the exact threshold value, which can be any small value. If the size of tolerance areas, which is already large enough, is increased by lowering the threshold value, a larger number of reference vectors will participate in the competition. The learning process, however, is scarcely affected by the change in the threshold, because reference vectors distant from the training vector does not influence the result of competition.

3. Computer simulation

3.1. Scale of the network

The arrangement of cells in each layer of the network is illustrated in Fig. 5, which shows a one-dimensional cross-section of connections between cell-planes. The reduction in density of cells in cell-planes, namely the thinning-out of the cells, is made from a S-cell layer to the C-cell layer of the same stage: The ratio of the thinning-out from U_{Sl} to U_{Cl} is 2:1 (in both horizontal and vertical directions) in all stages except U_{C4}. The pitch of cells in a cell-plane is the same between layers U_{Cl-1} and U_{Sl}, but the locations of C- and S-cells are staggered by half a pitch for $l \leqslant 3$.

The total number of cells (not counting inhibitory V-cells) in each layer is also shown in Fig. 5. Although the number of cells in each cell-plane has been pre-determined for all layers, the number of cell-planes in an S-cell layer (K_{Sl}) is determined automatically in the learning phase depending on the training set. In each stage except the highest

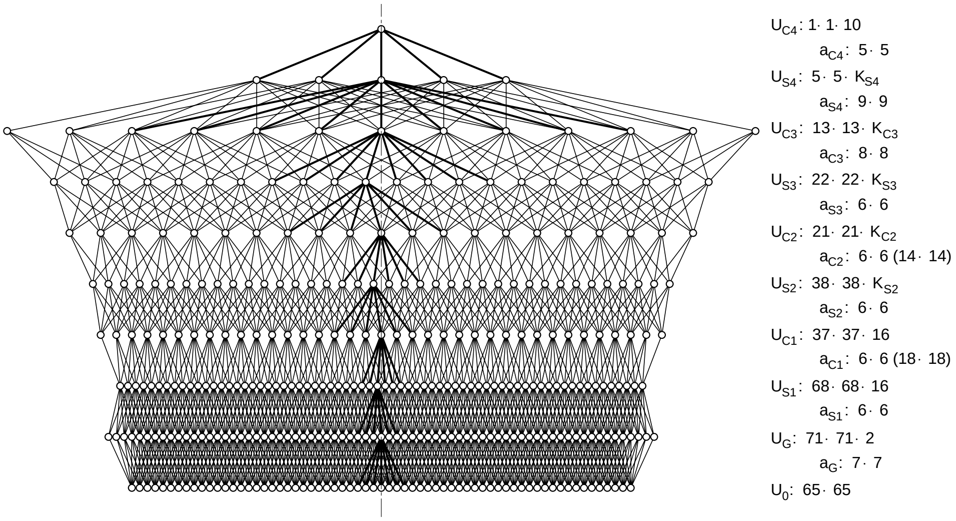


Fig. 5. Arrangement of cells and connections in the network. A one-dimensional cross-section of connections between cell-planes is drawn. Since the spatial spread of connections converging to a cell is not square but actually is circular, only approximate numbers of connections are shown in the figure. Only positive connections are drawn for a_{C1} and a_{C2}. To clearly show the input connections converging to a single cell, a cell is arbitrarily chosen from each layer, and its input connections are drawn with heavy lines.

one, the number of cell-planes of the C-cell layer (K_{Cl}) is the same as K_{Sl}. The recognition layer U_{C4} has only $K_{C4} = 10$ cell-planes, and each cell-plane contains only one C-cell.

The sizes of connections converging to single cells (namely, radii A_G, A_{Sl}, A_{Cl}, etc.) are determined as follows, where the pitch of cells in the cell-plane of the preceding layer is taken as the unit of length.[2] In the contrast-extracting layer U_G, the radius of input connections $a_G(\boldsymbol{\xi})$, namely A_G in (1), is 3.3, and the positive center of the Mexican-hat is 1.2 in radius. For S-cell layers, the radius of input connections A_{Sl} is 3.4 for $l = 1, 2, 3$, and 4.9 for $l = 4$. As for C-cell layers, $A_{C1} = 9.4$, $A_{C2} = 7.4$, $A_{C3} = 4.4$ and $A_{C4} = \infty$ ($A_{C4} = \infty$ means that each C-cell of U_{C4} receive input connections from all S-cells of U_{S4}). The excitatory center of the connections $a_{Cl}(\boldsymbol{v})$ is 3.4 in radius for the lower layers ($l = 1, 2$). For the higher layers (U_{C3} and U_{C4}), connections $a_{Cl}(\boldsymbol{v})$ do not have inhibitory surrounds and consist of only excitatory components.

The radius of the hypercolumn used for the competition area in the learning is 3.1 for U_{S2} and U_{S3}, and ∞ for U_{S4} where competition is made among all cells in the layer.

3.2. Recognition rate

We tested the behavior of the proposed network by computer simulation using handwritten digits (free writing) randomly sampled from the ETL1 database. Incidentally, the ETL1 is a database of segmented handwritten characters and is distributed by Electrotechnical Laboratory, Tsukuba, Japan [3].

For S-cells of layers U_{S2} and U_{S3}, the method of dual thresholds is used for the learning and recognition phases, as mentioned in Section 2.7. Each training pattern of the training set was presented once for the learning of layers U_{S2} and U_{S3}.

For the learning of layer U_{S4} at the highest stage, the same training set was presented repeatedly until all the patterns in the training set were recognized correctly. Although the required number of repetition changes depending on the training set, it usually is not so large. In the particular case shown below, it was only 4.

We searched the optimal thresholds that produce the best recognition rate. Since there are a large number of combinations in the threshold values of four layers, a complete search for all combinations has not been finished yet. We show here a result with a set of threshold values that seems to be nearly optimal.

The recognition rate varies depending on the number of training patterns. When we used 3000 patterns (300 patterns for each digit) for the learning, for example, the recognition rate was 98.6% for a blind test sample (3000 patterns), and 100% for the training set. The thresholds of S-cells used in this case were as follows. For the edge-extracting layer U_{S1}, we chose $\theta_1 = 0.55$. For the higher layers U_{S2}, U_{S3} and U_{S4}, the thresholds in the recognition phase were $\theta_2^R = 0.51$, $\theta_3^R = 0.58$ and $\theta_4^R = 0.30$.

[2] It should be noted here that, if two layers have the same radius based on the pitch of the cells in their respective layers, the actual size of the connections measured with the scale of the input layer is larger in the higher stage, because the density of cells is lower in the higher stage.

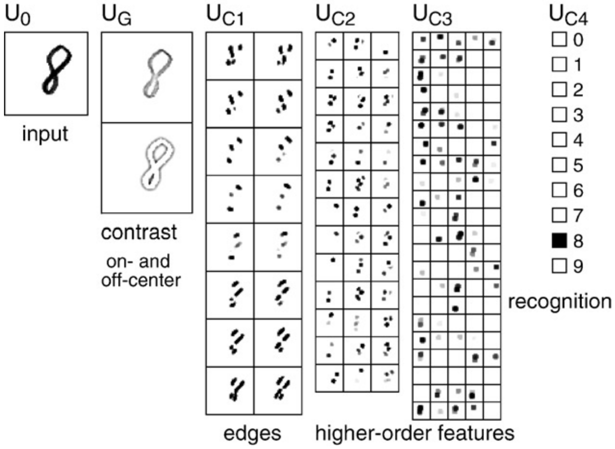


Fig. 6. An example of the response of the neocognitron. The input pattern is recognized correctly as ‘8’.

Those in the learning phase were: $\theta_2^L = 0.66$, $\theta_3^L = 0.67$ and $\theta_4^L = 0.30$. The numbers of cell-planes generated by this training set with these threshold values were $K_{S2} = 39$, $K_{S3} = 110$ and $K_{S4} = 103$.

Fig. 6 shows a typical response of the network that has finished the learning. The responses of layers U_0, U_G and layers of C-cells of all stages are displayed in series from left to right. The rightmost layer, U_{C4}, is the recognition layer, whose response shows the final result of recognition. The responses of S-cell layers are omitted from the figure but can be estimated from the responses of C-cell layers: a blurred version of the response of an S-cell layer appears in the corresponding C-cell layer, although it is slightly modified by the inhibitory surround in the connections.

Incidentally, we measured how the recognition rate changes depending on the size of the training set, using the same threshold values (optimal values that produced the recognition rate of 98.6% for the 3000 blind test set). The recognition rates for the same blind test sample (3000 patterns) were 92.8%, 94.8%, 97.0% and 98.0%, for 200, 500, 1000 and 2000 training patterns, respectively. The numbers of cell-planes generated by these training sets were: $(K_{S2}, K_{S3}, K_{S4}) = (21, 50, 22)$, $(27, 74, 45)$, $(33, 89, 63)$ and $(38, 109, 114)$, respectively.

4. Discussions

To improve the recognition rate of the neocognitron, several modifications have been applied: such as, the inhibitory surround in the connections from S- to C-cells, contrast-extracting layer followed by edge-extracting cells, self-organization of line-extracting cells, supervised competitive learning at the highest stage, staggered arrangement of S- and C-cells, and so on. These modifications allowed the removal of accessory circuits appended in the previous versions, resulting in an improvement of recognition rate as well as simplification of the network architecture.

Some people claim that a neocognitron is a complex network, but it is not correct. The mathematical operation between adjacent cell-planes can be interpreted as a kind of two-dimensional filtering operation because of shared connections. If we count the number of processes performed in the network, assuming that one filtering operation corresponds to one process, the neocognitron is a very simple network compared to other artificial neural networks. The required number of repeated presentation of a training set is much smaller for the neocognitron than for the network trained by backpropagation. In the computer simulation shown in this paper, for example, only 6 times of presentation was enough for the learning of the whole network (that is, 1 for U_{S2}, 1 for U_{S3}, and 4 for U_{S4}).

Although the phenomenon of overlearning (or overtraining) has not been observed in the simulation shown in this paper, the possibility cannot be completely excluded. If some patterns in a training set had wrong class names, the learning of the highest stage might be affected by the erroneous data. A serious overlearning, however, does not seem to occur in the intermediate stages, if the thresholds for the learning phase are properly chosen. Since the selectivity (or the size of the tolerance area) of a cell is determined by a fixed threshold value and does not change, all training patterns within the tolerance area of the winner cell contribute to the learning of the cell, and not for the generation of new cell-planes. In other words, an excessive presentation of training patterns does not necessarily induce the generation of new cell-planes. Although a repeated presentation of similar training patterns to a cell increases the values of the connections to the cell, the behavior of the cell almost stops changing after having finished some degrees of learning. This is because the response of the cell, which receives a shunting inhibition, is determined by the ratio, not by the difference, of excitatory and inhibitory inputs, if the connections have already been large enough (see the discussions in Section 2.3.3).

The neocognitron has several parameters that have to be predetermined before learning. Among them, parameters that critically affect the performance of the network are thresholds of S-cell, that is, θ_1, θ_2^{R}, θ_2^{L}, θ_3^{R}, θ_3^{L}, θ_4^{R} and θ_4^{L}. Threshold θ_1 of the edge-extracting layer can be determined by the method discussed in Section 2.6. Thresholds $\theta_4^{\mathrm{R}} = \theta_4^{\mathrm{L}}$ for the highest stage can take an arbitrary small value, as was discussed in Section 2.8. However, a good method for determining the thresholds of the intermediate stages, θ_2^{R}, θ_2^{L}, θ_3^{R} and θ_3^{L}, has not been fully established yet. Their optimal values vary depending on the characteristics of the pattern set to be recognized. We need to search optimal thresholds by experiments. To find out a good method for determining these thresholds is a problem left to be solved in the future.

References

[1] E. Barnard, D. Casasent, Shift invariance and the neocognitron, Neural Networks 3 (4) (1990) 403–410.

[2] M.C.M. Elliffe, E.T. Rolls, S.M. Stringer, Invariant recognition of feature combinations in the visual system, Biol. Cybernet. 86 (2002) 59–71.

[3] ETL1 database: http://www.etl.go.jp/∼etlcdb/index.htm.

[4] K. Fukushima, Neocognitron: a hierarchical neural network capable of visual pattern recognition, Neural Networks 1 (2) (1988) 119–130.

[5] K. Fukushima, S. Miyake, Neocognitron: a new algorithm for pattern recognition tolerant of deformations and shifts in position, Pattern Recognition 15 (6) (1982) 455–469.

[6] K. Fukushima, K. Nagahara, H. Shouno, Training neocognitron to recognize handwritten digits in the real world, pAs'97 (2nd Aizu International Symposium on Parallel Algorithms/Architectures Synthesis), IEEE Computer Society Press, Silver Spring, MD, 1997, pp. 292–298.

[7] K. Fukushima, M. Tanigawa, Use of different thresholds in learning and recognition, Neurocomputing 11 (1) (1996) 1–17.

[8] K. Fukushima, N. Wake, Improved neocognitron with bend-detecting cells, IJCNN'92, Vol. IV, Baltimore, MD, USA, 1992, pp. 190–195.

[9] R.M. Gray, Vector quantization, IEEE Trans. Acoust. Speech Signal Process. Mag. 1 (2) (1984) 4–29.

[10] T.H. Hildebrandt, Optimal training of thresholded linear correlation classifiers, IEEE Trans. Neural Networks 2 (6) (1991) 577–588.

[11] G.S. Himes, R.M. Iñigo, Automatic target recognition using a neocognitron, IEEE Trans. Knowledge Data Eng. 4 (2) (1992) 167–172.

[12] D.H. Hubel, T.N. Wiesel, Receptive fields, binocular interaction and functional architecture in the cat's visual cortex, J. Physiol. (London) 106 (1) (1962) 106–154.

[13] D.H. Hubel, T.N. Wiesel, Receptive fields and functional architecture in two nonstriate visual areas (18 and 19) of the cat, J. Neurophysiol. 28 (2) (1965) 229–289.

[14] E.J. Kim, Y. Lee, Handwritten hangul recognition using a modified neocognitron, Neural Networks 4 (6) (1991) 743–750.

[15] T. Kohonen, The self-organizing map, Proc. IEEE 78 (9) (1990) 1464–1480.

[16] Y. LeCun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard, L.J. Jackel, Backpropagation applied to handwritten zip code recognition, Neural Comput. 1 (1989) 541–551.

[17] S.B. Lo, H. Chan, J. Lin, H. Li, M.T. Freedman, S.K. Mun, Artificial convolution neural network for medical image pattern recognition, Neural Networks 8 (7/8) (1995) 1201–1214.

[18] M.M. Menon, K.G. Heinemann, Classification of patterns using a self-organizing neural network, Neural Networks 1 (3) (1988) 201–215.

[19] C. Neubauer, Evaluation of convolutional neural networks for visual recognition, IEEE Trans. Neural Networks 9 (4) (1998) 685–696.

[20] A. van Ooyan, B. Nienhuis, Pattern recognition in the neocognitron is improved by neuronal adaptation, Biol. Cybernet. 70 (1) (1993) 47–53.

[21] S. Sato, J. Kuroiwa, H. Aso, S. Miyake, Recognition of rotated patterns using a neocognitron, in: L.C. Jain, B. Lazzerini (Eds.), Knowledge Based Intelligent Techniques in Character Recognition, CRC Press, Boca Raton, FL, 1999, pp. 49–64.

[22] D. Shi, C. Dong, D.S. Yeung, Neocognitron's parameter tuning by genetic algorithms, Internat. J. Neural Systems 9 (6) (1999) 495–509.

[23] C.H. Ting, Magnocellular pathway for rotation invariant neocognitron, Internat. J. Neural Systems 4 (1) (1993) 43–54.

[24] C. Ting, K. Chuang, An adaptive algorithm for neocognitron to recognize analog images, Neural Networks 6 (2) (1993) 285–299.

[25] D.S. Yeung, H. Chan, A hybrid cognitive system using production rules to synthesize neocognitrons, Internat. J. Neural Systems 5 (4) (1994) 345–355.

[26] D.S. Yeung, H.S. Fong, A knowledge matrix representation for a rule-mapped neural network, Neurocomputing 7 (2) (1995) 123–144.

[27] B.A. White, M.I. Elmasry, The digi-neocognitron: a digital neocognitron neural network model for VLSI, IEEE Trans. Neural Networks 3 (1) (1992) 73–85.

Kunihiko Fukushima is a Full Professor, Katayanagi Advanced Research Laboratories, Tokyo University of Technology, Tokyo, Japan. He received a B.Eng. degree in electronics in 1958 and a Ph.D. degree in electrical engineering in 1966 from Kyoto University, Japan. He was a full professor at Osaka University from 1989 to 1999, at the University of Electro-Communications from 1999 to 2001. Prior to his Professorship, he was a Senior Research Scientist at the NHK Science and Technical Research Laboratories. He is one of the pioneers in the field of neural networks and has been engaged in modeling neural networks of the brain since 1965. His special interests lie in modeling neural networks of the higher brain functions, especially, the mechanism of the visual system, learning and memory. He invented the *Neocognitron* for deformation invariant pattern recognition, and the *Selective Attention Model*, which can recognize and segment overlapping objects in the visual fields. One of his recent research interests is in modeling neural networks for active vision in the brain. He is the author of many books on neural networks, including “Neural Networks and Information Processing”, “Neural Networks and Self-Organization”, and “Physiology and Bionics of the Visual System”. Prof. Fukushima is the founding President of JNNS (the Japanese Neural Network Society) and is a founding member on the Board of Governors of INNS (the International Neural Network Society). He serves as an editor for many international journals.