Neurocomputing 129 (2014) 78–84

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Neocognitron trained by winner-kill-loser with triple threshold

Kunihiko Fukushima [a,*], Isao Hayashi [b], Jasmin Léveillé [c]

[a] Fuzzy Logic Systems Institute, Iizuka, Fukuoka 820-0067, Japan
[b] Faculty of Informatics, Kansai University, Takatsuki, Osaka 569-1095, Japan
[c] Center for Computational Neuroscience and Neural Technology, Boston University, Boston, MA 02215, USA

ARTICLE INFO

Article history:
Received 11 January 2012
Received in revised form
28 April 2012
Accepted 8 May 2012
Available online 25 October 2013

Keywords:
Visual pattern recognition
Neocognitron
Hierarchical network
Winner-kill-loser
Triple threshold

ABSTRACT

The *neocognitron* is a hierarchical, multi-layered neural network capable of robust visual pattern recognition. The neocognitron acquires the ability to recognize visual patterns through learning. The winner-kill-loser is a competitive learning rule recently shown to outperform standard winner-take-all learning when used in the neocognitron to perform a character recognition task. In this paper, we improve over the winner-kill-loser rule by introducing an additional threshold to the already existing two thresholds used in the original version. It is shown theoretically, and also by computer simulation, that the use of a triple threshold makes the learning process more stable. In particular, a high recognition rate can be obtained with a smaller network.

1. Introduction

The *neocognitron*, originally proposed by Fukushima [1], is a hierarchical, multi-layered neural network capable of robust visual pattern recognition. Its architecture was initially inspired from neurophysiological findings about the structure of the visual systems of mammals (e.g., [2,3]). The neocognitron acquires the ability to recognize patterns through learning.

During learning, input connections to feature-extracting cells are modified upon presentation of training patterns. Since the neocognitron is a multi-layered network, the learning method used to train the intermediate stages of the network greatly affects the performance of the neocognitron. Several methods for training have been proposed to date. One of them, the *winner-kill-loser* learning rule, has been shown to be very powerful in this respect [4].

This paper proposes an improved learning rule, in which we use a triple threshold, instead of the dual threshold associated with the original winner-kill-loser. We show theoretically, and also by computer simulation, that the use of a triple threshold makes the learning process more stable. In particular, our simulations show that a high recognition rate can be obtained despite using a smaller network.

2. Outline of the network

2.1. Network architecture

The neocognitron consists of layers of S-cells, which are analogous to simple cells in the visual cortex, and layers of C-cells, which resemble complex cells. Layers of S-cells and C-cells are interleaved in a hierarchical manner.

Fig. 1 shows the architecture of the network discussed in this paper. The label U_{Sl}, for example, stands for the layer of S-cells at the lth stage. The particular network shown in Fig. 1 has four stages of S- and C-cell layers.

Each layer in the network is divided into a number of sub-layers, called *cell-planes*, depending on the feature to which cells respond preferentially. In Fig. 1, each rectangle drawn with thick lines represents a cell-plane. A cell-plane can also be described as a group of cells that are arranged retinotopically and share the same set of input connections [1]. In other words, all cells in a cell-plane have identical receptive fields but at different locations.

2.2. Edge extraction in layer U_{S1}

Stimulus patterns are presented to the input layer, U_0. The output of U_0 is then sent directly to U_{S1} via feedforward projections. An S-cell

* Corresponding author. Permanent address: 634-3, Miwa, Machida, Tokyo 195-0054, Japan. Tel./fax: +81 44 988 5272.
E-mail addresses: fukushima@m.ieice.org (K. Fukushima), ihaya@cbii.kutc.kansai-u.ac.jp (I. Hayashi), jalev51@gmail.com (J. Léveillé).
URL: http://www4.ocn.ne.jp/~fuku_k/index-e.html (K. Fukushima).

0925-2312/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.neucom.2012.05.038

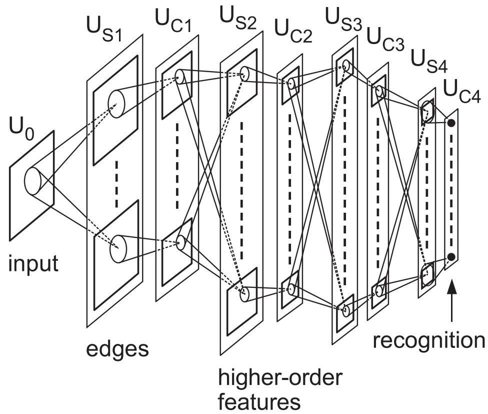

Fig. 1. Architecture of the neocognitron. Image input applied to layer U_0 is projected to interleaved stages of S-cells (U_{Sl}) and C-cells (U_{Cl}). S-cells act as feature detectors tuned to particular edge configurations. C-cells pool the responses of S-cells over a restricted spatial neighborhood, and thereby achieve a certain degree of spatial invariance.

in this layer resembles a simple cell in the primary visual cortex in the sense that it responds selectively to an edge at a particular orientation. To be more specific, U_{S1} has $K_{S1} = 16$ cell-planes, whose preferred orientations are chosen at an interval of 22.5°. Contours in the input image are thus projected onto a full basis set of oriented edges U_{S1}.

Unlike the S-cells in subsequent layers, S-cells in U_{S1} are made of analog threshold elements. An analog threshold element calculates a weighted sum of its inputs and cuts off the negative component by a threshold operation.

Mathematically, an S-cell in U_{S1} extracts an oriented edge directly from U_0 using a linear filter followed by a half-wave rectification. A mechanism of weak lateral inhibition among S-cells further enhances the selectivity of each cell to their preferred orientation. The shape of the linear filter is encoded in the input connections to an S-cell and is implemented as a directional derivative of a two-dimensional Gaussian.

The neocognitron used here thus differs from previous versions [5] in which the S-cells were defined in the same way across all layers and where an additional contrast-extracting layer, U_G, was present between U_0 and U_{S1}.

2.3. *S-cell layers* U_{S2}, U_{S3} *and* U_{S4}

S-cells in the later stages (U_{S2} to U_{S4}) differ from those in U_{S1}, and are hence described separately.

We now use vector notation to represent the response of an S-cell. Let $\boldsymbol{x} = (x_1, x_2, \ldots, x_N)$ be the input vector from a set of presynaptic C-cells to the S-cell, and $\boldsymbol{X} = (X_1, X_2, \ldots, X_N)$ be the corresponding vector of synaptic connections from the C-cells to the S-cells. We can interpret $\boldsymbol{X}$ as the preferred (optimal) feature of the S-cell in a multi-dimensional feature space. For this reason, $\boldsymbol{X}$ is typically referred to as the *reference vector* of the S-cell.

Each S-cell is accompanied by an inhibitory V-cell. The V-cell computes the intensity of the input signal to its corresponding S-cell using the L_2-norm (root-mean-square) $\|\boldsymbol{x}\|$. The V-cell uses this quantity to inhibit the S-cell. In the previous formulations of the neocognitron, the V-cell performed a divisive normalization operation. In the current model, however, the V-cell inhibits the S-cell in a subtractive manner, since this has been shown to increase robustness to background noise [6].

When an input vector $\boldsymbol{x}$ is presented, the S-cell computes the similarity s between $\boldsymbol{x}$ and its own reference vector $\boldsymbol{X}$ with the following normalized inner-product:

$$s = \frac{(\boldsymbol{X}, \boldsymbol{x})}{\|\boldsymbol{X}\| \cdot \|\boldsymbol{x}\|}. \quad (1)$$

If similarity s is larger than a certain threshold θ $(0 < \theta < 1)$, the S-cell yields a non-zero response [6]:

$$u = \|\boldsymbol{x}\| \cdot \frac{\varphi[s-\theta]}{1-\theta}, \quad (2)$$

where $\varphi[]$ is defined as $\varphi[x] = \max(x, 0)$. The value of threshold θ is determined by the strength of inhibitory connections from a V-cell to its S-cell.

The range of similarity values for which $s > \theta$ is called the *tolerance area* of the S-cell. This situation is illustrated in Fig. 2 for the case of a three-dimensional feature space. The S-cell thus yields a non-zero response, only when $\boldsymbol{x}$ is within the tolerance area around $\boldsymbol{X}$.

2.4. *C-cell layers*

Except at the final stage, the C-cell layer U_{Cl} has the same number of cell-planes as the S-cell layer U_{Sl} at the same stage.

The output of a given cell-plane of S-cell layer U_{Sl} is forwarded to a corresponding cell-plane of C-cell layer U_{Cl}. C-cells have fixed input connections. By averaging their input signals across a fixed spatial neighborhood, C-cells exhibit some level of translation invariance. As a result of averaging across position, C-cells encode a blurred version of their input. The blurring operation is essential for making the neocognitron robust to deformation, change in size, or shift in the position of input patterns. Unlike previous versions of the neocognitron that used the arithmetic mean, in the current model pooling is done via root-mean-square [6].

As in previous versions of the neocognitron, excitatory connections to the C-cells in U_{C1} and U_{C2} are surrounded by inhibitory connections, thereby forming an on-center off-surround connectivity pattern.

2.5. *Training S-cells*

Input connections to S-cells are modified via learning. After learning, S-cells become tuned to specific feature configurations. S-cells at higher stages of the hierarchy extract more global features than S-cells at lower stages.

Learning is performed layerwise, from lower layers to higher layers, such that the training of a given stage can start only after the training of the preceding stage is complete. In order to train S-cells in U_{Sl}, the responses of C-cells in the preceding layer

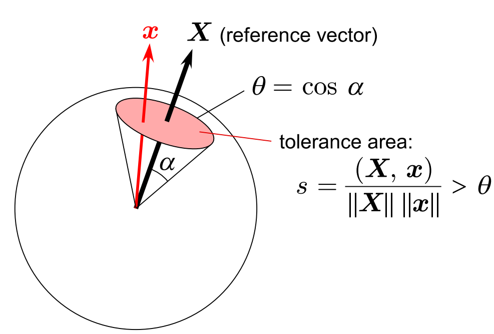

Fig. 2. Tolerance area of an S-cell in the multi-dimensional feature space. Here, $\boldsymbol{x}$ is an input vector, $\boldsymbol{X}$ is an vector representing the S-cell's synaptic connections, and θ is a threshold that defines the tolerance area for the corresponding S-cell. Only input vectors in the tolerance area (pink shaded zone) are able to activate the S-cell. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

U_{Cl-1} are used as a training stimulus. All S-cell layers – except for U_{S1} – are trained on the same training set.

We use unsupervised competitive learning to train intermediate layers U_{S2} and U_{S3}, and supervised competitive learning to train the final layer U_{S4}. In competitive learning, winners are chosen by competition among neighboring S-cells in a given layer, and only the winners learn the training stimulus.

As mentioned above, each layer of the neocognitron is divided into cell-planes. All cells in a given cell-plane share the same set of input connections. Shared connections are maintained throughout learning. When a winner is chosen in a given cell-plane, its input connections are first renewed based on the presynaptic C-cell responses. The connections to all other cells in the cell-plane are then adjusted accordingly so as to maintain the same connections as the winner. The winner thus works like a seed in crystal growth. Hence we call it a *seed-cell*.

We use the *winner-kill-loser* rule, a variant of competitive learning, to train intermediate layers U_{S2} and U_{S3} [4]. We propose to use a triple threshold to govern learning, instead of the dual threshold that was used in the original winner-kill-loser rule. This is the main topic of this paper and is discussed in more detail in the next section.

2.6. Final stage

S-cells at the final stage (U_{S4}) are trained by supervised competitive learning using labeled training data [5].

S-cells in the final layer compete upon presentation of each training pattern. 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. However, if the winner has a wrong label (or if all S-cells are silent), a new cell-plane is generated. The new cell-plane hence learns the current training pattern simply by being assigned its corresponding label. Each cell-plane of U_{S4} thus has a label indicating one of the 10 digits. As the network learns varieties of deformed training patterns, more than one cell-plane per class is usually generated in U_{S4}.

Layer U_{C4} at the final stage has 10 C-cells corresponding to the 10 digits, to which input patterns are to be classified. During the recognition phase, the class of the input pattern is determined by the most active S-cell in U_{S4}. The output of the S-cell is sent only to the corresponding C-cell of the inferred label.

3. Competitive learning with the winner-kill-loser rule

3.1. Winner-kill-loser with dual threshold

The winner-kill-loser (WKL) rule is used in order to train S-cells in layers U_{S2} and U_{S3} [4]. We first describe the original winner-kill-loser rule.

Fig. 3 illustrates the learning process with the original winner-kill-loser rule [4], and compares it to other learning rules.

The Hebbian rule, shown at the top of Fig. 3(a), is one of the most commonly used learning rules. In the learning phase, each synaptic connection is strengthened by an amount proportional to the product of the responses of the pre- and post-synaptic cells.

In the winner-take-all rule (WTA), shown in the middle of Fig. 3(a), post-synaptic cells compete with each other, and the most active cell becomes the winner. Only the winner can have its input connections renewed. Input connections are modified by adding the training vector $\boldsymbol{x}$ to the vector of synaptic connections $\boldsymbol{X}$. The magnitude of the weight change is proportional to the response of the pre-synaptic cell. Most formulations of the neocognitron use this learning rule [1,5].

The winner-kill-loser rule (WKL), shown at the bottom of Fig. 3(a), resembles the winner-take-all rule in the sense that only the winner learns the training stimulus. In the winner-kill-loser rule, however, not only does the winner learn the training stimulus, but also the losers are simultaneously removed from the network. Losers are defined as cells whose responses to the training stimulus are smaller than that of the winner, but whose activations are nevertheless greater than zero.

If a training stimulus elicits non-zero responses from two or more S-cells, it means that the preferred features of these cells resemble each other, and that they work redundantly in the network. To reduce this redundancy, only the winner has its input connections renewed so as to fit more to the training vector, while the other active cells, namely the losers, are removed from the network.

Since silent S-cells (namely, the S-cells whose responses to the training stimulus are zero) do not join the competition, they are not removed. These cells are expected to work toward extracting other features.

If all S-cells are silent as depicted in Fig. 3(b), a new S-cell is generated, and the training vector $\boldsymbol{x}$ becomes the reference vector for that new S-cell. In other words, the initial value of the input

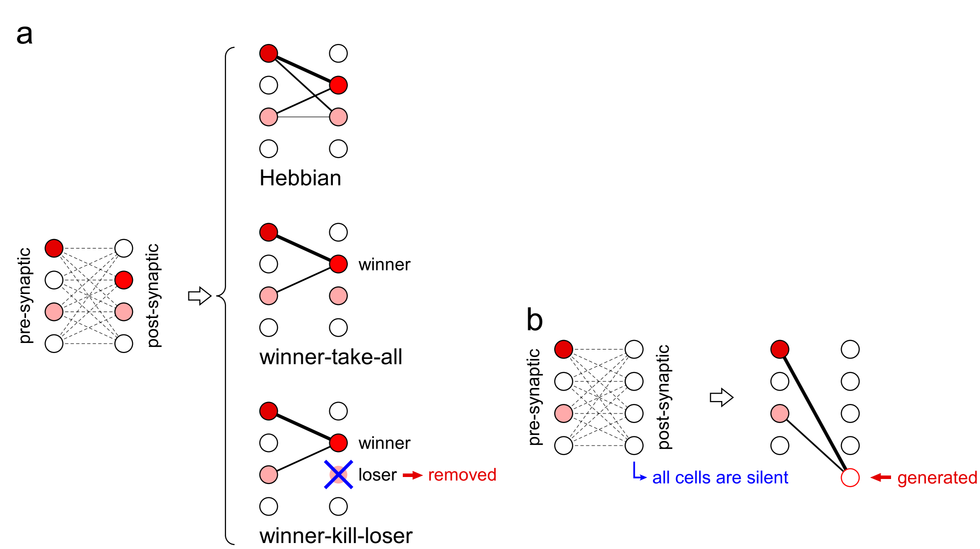

Fig. 3. Winner-kill-loser rule with dual threshold, in comparison with other learning rules. In this figure, the response of each cell is represented by the level of color saturation. (a) Several rules of learning. (b) A new cell is generated when all postsynaptic cells are silent. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

connections of the newly generated S-cell is proportional to the response of the pre-synaptic C-cells. Incidentally, the generation of new S-cells was also a feature of the winner-take-all rule implemented in previous versions of the neocognitron.

In the learning phase, a number of training stimuli are presented sequentially to the network. During this process, generation of new cells and removal of redundant cells occurs repeatedly in the network. In particular, new cells are generated to cover areas of the multi-dimensional feature space that were not previously covered by existing cells. In the areas where similar cells exist in duplicate, redundant cells are removed. By repeating this process for a long enough time, the preferred features (reference vectors) of S-cells gradually become distributed uniformly over the multi-dimensional feature space.

In the original winner-kill-loser rule, a dual threshold is used to guide learning and recognition in S-cells [4]. Namely, S-cells, whose response is given by Eq. (2), have a higher threshold during learning (θ^{L}) than during recognition (θ^{R}).

During learning, S-cells join the competition only when their responses are above zero under the high threshold θ^{L}. After the learning with the high threshold, each stimulus will tend to activate only one S-cell. This contributes toward reducing redundancy and covering uniformly the multidimensional input space.

In order to recognize deformed patterns robustly, however, population coding is necessary in intermediate layers. In other words, each stimulus needs to activate a small set of S-cells, such that these S-cells jointly represent the stimulus. To promote population coding, we use a lower threshold θ^{R} for the recognition phase than the threshold θ^{L} for the learning phase.

When applying this learning rule to the neocognitron, a slight modification is required due to the fact that each layer of the network consists of a number of cell-planes such that all cells in a given cell-plane must share the same set of input connections both during learning and recognition.

At first, the S-cell whose response is the largest in the layer is chosen as a seed-cell. The seed-cell has its input connections renewed depending on the training vector presented to it. Once the connections to the seed-cell are renewed, all cells in the cell-plane from which the seed-cell is chosen are modified so as to have the same set of input connections as the seed-cell. All non-silent cells at the same spatial location as the seed cell are tagged as losers, and the cell-planes to which they belong are removed from the layer.

Computer simulations show that the use of the winner-kill-loser rule for competitive learning greatly increases the recognition rate in small networks [4]. However, some problems still remain that make the winner-kill-loser inefficient when using the dual threshold procedure described above.

3.1.1. Problems with the dual threshold formulation

The underlying motivation behind the winner-kill-loser learning process is to promote a uniform covering of the multidimensional feature space by the S-cell reference vectors (Fig. 4(a)).

Let us now observe how the distribution of reference vectors changes during learning. We assume that, at a certain moment during learning, we have a uniform distribution of reference vectors as shown in Fig. 4(a). If a training vector is presented at ✚ in Fig. 4(b), the cell centered on ■ is the winner and learns the training vector as shown in Fig. 4(c). This is all right.

If a training vector is presented at ✚ as shown in Fig. 4(d), however, all cells are silent and a new cell, whose reference vector is at ♦, is generated as shown in Fig. 4(e). After that, if another training vector is presented at ✚ in Fig. 4(f), the cell at ■ becomes a winner and the cell at ▲ becomes a loser. Removal of the loser leads to the situation depicted in Fig. 4(g). Thus, further training can actually destroy the desirable uniform distribution that was initially present in Fig. 4(a).

In other words, the removal and generation of cell-planes in the neocognitron do not stabilize during learning. Since the number of cell-planes continues to increase and decrease, the final number of cell-planes obtained after learning is determined largely by the duration of the learning phase. This in turn will strongly affect the network's recognition rate and scale.

3.2. Use of triple threshold for winner-kill-loser

In the winner-kill-loser with a dual threshold, which is discussed above, we use a high threshold θ^{L} for learning and a low threshold θ^{R} for recognition. That is, in the learning phase, only one threshold θ^{L} is used.

We now propose to add one more threshold during learning. Specifically, we use thresholds θ^{W} and θ^{G}, instead of only one threshold θ^{L} (Fig. 5).

Threshold θ^{W} is used to separate the winner from the losers: non-zero responses are only elicited from S-cells whose similarity s (namely, the similarity between reference vector $\boldsymbol{X}$ of the cell and the training vector $\boldsymbol{x}$) is larger than θ^{W}. Among these S-cells, the S-cell that has the largest response becomes the winner and learns $\boldsymbol{x}$. As before, other non-silent S-cells are tagged as losers and removed from the network.

The threshold θ^{G} works like a kind of subliminal threshold and controls the generation of a new S-cell (namely, the generation of a new reference vector in the vector space). If there exists at least one S-cell whose similarity s is larger than θ^{G}, no new S-cell is generated. In other words, not only an active S-cell with $s > \theta^{\mathrm{W}}$, but also a silent S-cell whose similarity s is in the range $\theta^{\mathrm{W}} \geq s > \theta^{\mathrm{G}}$ can prevent the generation of a new S-cell. A new

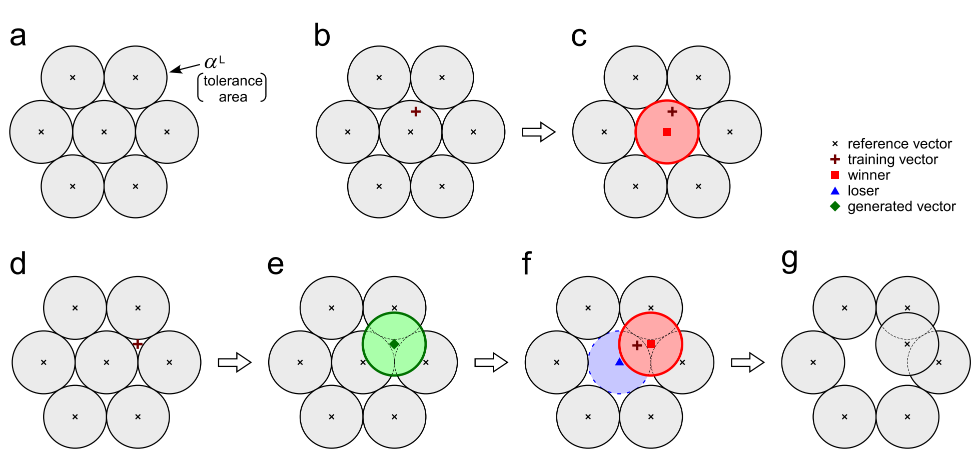

Fig. 4. Progress of learning by winner-kill-loser with a dual threshold. The dual threshold means that, in the learning phase, only one threshold, θ^{L} ($=\cos\alpha^{\mathrm{L}}$), is used.

S-cell can be generated only when the similarity s is under the threshold θ^{G} for all S-cells. This situation is illustrated in Fig. 6.

3.2.1. Optimal values of the thresholds

We now discuss how to choose threshold values θ^{W} and θ^{G}. To represent the radius of the tolerance area based on the threshold θ, we define the angle α between two vectors in the multi-dimensional feature space as follows: $\theta^{W}=\cos\alpha^{W}$ and $\theta^{G}=\cos\alpha^{G}$ (see also Fig. 2). Although the feature space is actually of dimensionality greater than two, for purpose of illustration our discussion assumes a two-dimensional plane.

The goal of training is to make reference vectors distribute uniformly in the feature space. Once a desirable uniform distribution of reference vectors has emerged during learning, it should not be destroyed upon further training. To prevent reference vectors from becoming losers and being removed, adjacent tolerance areas of radius α^{W} should not overlap (Fig. 7). In a feature space of dimensionality greater than one, however, some vacant gaps are generated between non-overlapping disks of radius α^{W}. To prevent generation of a new reference vector, the feature space is covered by disks of radius α^{G} that can overlap with each other. The smallest α^{G} that can fill vacant gaps can be determined from α^{W}, as illustrated in the right of Fig. 7. Namely,

$$\alpha^{W}=\alpha^{G}\cos(\pi/6),\quad \text{or}\quad \cos^{-1}\theta^{W}=\cos(\pi/6)\cos^{-1}\theta^{G}. \tag{3}$$

	dual threshold	triple threshold	
learning phase	θ^{L}	θ^{W}	for choosing a winner & losers
		θ^{G}	for generating a new cell-plane
recognition phase	θ^{R}	θ^{R}	

Fig. 5. Comparison of dual and triple thresholds.

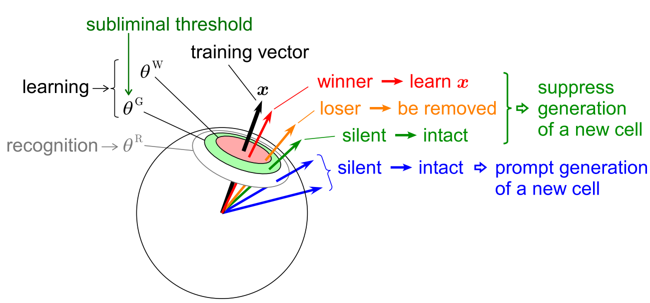

Fig. 6. Learning by winner-kill-loser with triple threshold in the multi-dimensional vector space. Here we propose to use a subliminal threshold, θ^{G}, such that the presence of a cell whose similarity s is greater than θ^{G} prevents the generation of a new cell.

4. Computer simulation

Simulations were conducted to compare the impacts of the dual and triple thresholds in the neocognitron trained with the winner-kill-loser rule. The training set we use consists of 3000 handwritten digits (300 patterns for each digit) randomly sampled from the ETL1 database [7]. This training set is presented only once to train layers U_{S2} and U_{S3}.

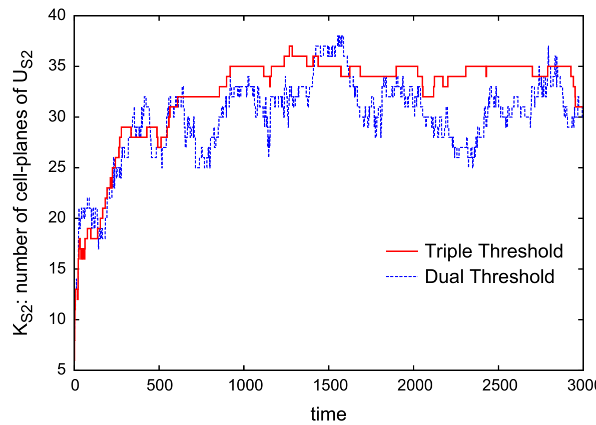

Fig. 8. Number of cell-planes (K_{S2}) during learning. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

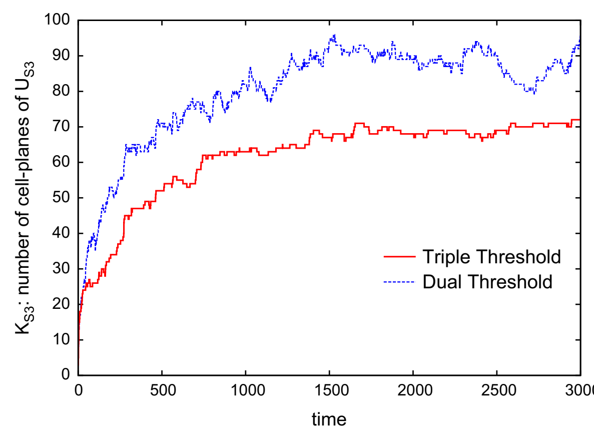

Fig. 9. Number of cell-planes (K_{S3}) during learning. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

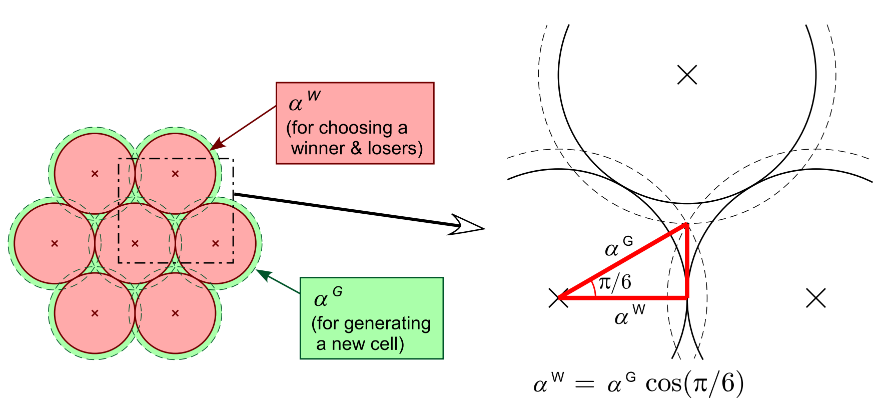

Fig. 7. Winner-kill-loser with triple threshold. Using thresholds $\theta^{G}(=\cos\alpha^{G})$ and $\theta^{W}(=\cos\alpha^{W})$ for learning allows covering the gaps that may emerge between the tolerance areas of neighboring units.

Figs. 8 and 9 show how the number of cell-planes in layers U_{S2} and U_{S3} changes during learning. Results for the triple and dual thresholds are depicted as a red solid line and a blue dotted line, respectively. It can be seen from these figures that the fluctuation of the number of cell-planes is much smaller with the triple threshold and that the learning is more stable (as seen from the fact that the curve is smoother). The number of cell-planes created after learning is usually smaller with the triple threshold than with the dual threshold. In this particular case, however, the final number of K_{S3} is smaller with the triple threshold, but K_{S2} is almost the same.

Fig. 10 shows a typical response of the cells after learning with the same triple threshold used for the network in Figs. 8 and 9. For simplicity, only the responses of layers of C-cells are shown here. The leftmost layer, U_0, is the input layer. The rightmost layer, U_{C4}, shows the final recognition result. In this example, it is clear that the input pattern '8' presented to U_0 is recognized correctly.

Fig. 11 shows how the error rate of the neocognitron changes with the size of the training set. The test set consists of 5000 patterns (500 patterns for each digit). Experiments were repeated twice for each condition, using different training and test sets randomly sampled from the ETL1 database, and the results were averaged across the two experiments. We can see that the recognition error itself does not differ much either using the dual (blue dotted line) or triple threshold (red solid line). Although the recognition error of the neocognitron depends on the final number of cell-planes that have been created after learning, we usually have almost the same recognition rate with a smaller number of cell-planes when the network is trained with the triple threshold.

As an example, the final number of cell-planes in each layer for the simulations of Figs. 8–10 was $(K_{S2}, K_{S3}, K_{S4}) = (31, 72, 73)$, and the recognition error was 1.22%, under the triple threshold. On the other hand, under the dual threshold, we had $(K_{S2}, K_{S3}, K_{S4}) = (30, 96, 82)$ and 1.26%, respectively. It should be noted here that the computational cost for calculating the response of U_{Sl} is approximately proportional to $K_{Sl-1} \times K_{Sl}$.

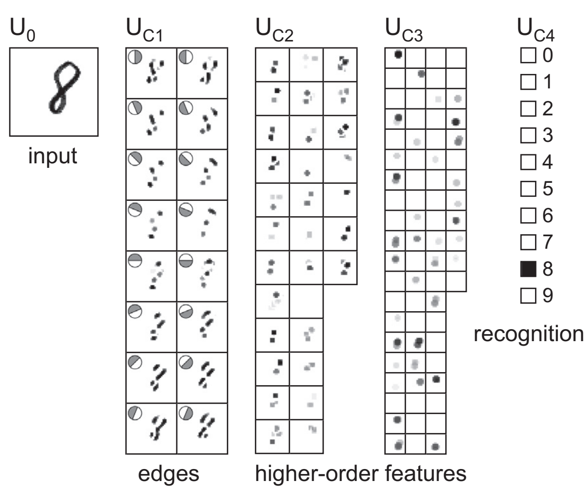

Fig. 10. An example of the response of the neocognitron trained by winner-kill-loser with triple threshold. The half disk drawn in each cell-plane of U_{C1} shows the orientation of the edge extracted by the cell-plane. The rightmost layer, U_{C4}, shows that the input pattern is recognized correctly as '8'.

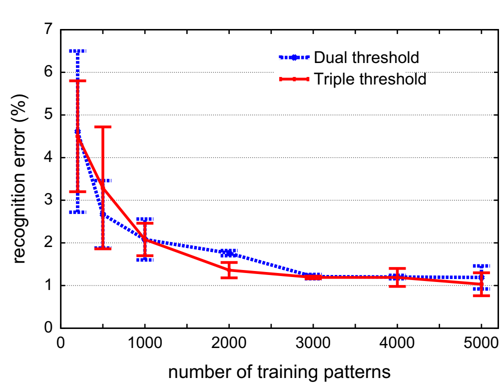

Fig. 11. Recognition error vs. number of training patterns. The error bars represent the standard deviation. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

5. Discussion

In this paper we introduce a new triple threshold to be used for competitive learning with the winner-kill-loser rule. We show by computer simulation that the use of the triple threshold makes the learning process more stable than when using the dual threshold. Although the triple threshold does not seem to improve the recognition rate, it nevertheless significantly reduces the network scale (with a smaller number of cell-planes in each layer). One of the greatest merits of the triple threshold formulation is the stability of learning, which in turn makes the neocognitron less sensitive to the duration of the learning phase.

Acknowledgments

This work was partially supported by Kansai University by Strategic Project to Support the Formation of Research Bases at Private Universities: Matching Fund Subsidy from MEXT, 2008–2012.

References

[1] K. Fukushima, Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position, Biol. Cybern. 36 (1980) 193–202.
[2] D.H. Hubel, T.N. Wiesel, Receptive fields, binocular interaction and functional architecture in the cat's visual cortex, J. Physiol. (London) 160 (1962) 106–154.
[3] 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 (1965) 229–289.
[4] K. Fukushima, Neocognitron trained with winner-kill-loser rule, Neural Networks 23 (2010) 926–938.
[5] K. Fukushima, Neocognitron for handwritten digit recognition, Neurocomputing 51 (2003) 161–180.
[6] K. Fukushima, Increasing robustness against background noise: visual pattern recognition by a neocognitron, Neural Networks 24 (2011) 767–778.
[7] ETL1 Database: 〈http://projects.itri.aist.go.jp/etlcdb/〉.

Kunihiko Fukushima 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 professor at Osaka University, Japan, from 1989 to 1999, then, at the University of Electro-Communications, and at Tokyo University of Technology. He was then a visiting professor at Kansai University from 2006 to 2010. Prior to his Professorship, he was a Senior Research Scientist at the NHK Science and Technical Research Laboratories. After retirement from universities, he had part-time positions as research scientist at several laboratories, and he is now a Senior Research Scientist at Fuzzy Logic Systems Institute. 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. He was the founding President of JNNS and was a founding Member on the Board of Governors of INNS. He is a former President of APNNA.

Isao Hayashi is a full professor, Faculty of Informatics at Kansai University, Osaka, Japan. He received a B.Eng. degree in 1981, a M.Eng. degree in 1985 and a Ph.D. degree in Industrial Engineering in 1991 from Osaka Prefecture University. He was a full professor at Hannan University from 1993 to 2004. Prior to his associate professorship and professorship, he was a senior researcher at Matsushita Electric Industrial (Panasonic) Co., Ltd, and an engineer at Sharp Corporation. He is a pioneer in Neuro-Fuzzy Systems and was the one who promoted the spread of fuzzy application in the industry of the 1990s. His interests lie in modeling of knowledge using soft computing. His recent research interests are in modeling of knowledge in brain-computer interface and human skill. He is an Editorial Member of International Journal of Hybrid Intelligent Systems, Journal of Advanced Computational Intelligence and Intelligent Informatics (JACIII), and is a Member of Kansai chapter of Japan Society for Fuzzy Theory and Intelligent Informatics (SOFT), and an Executive General of the technical group on Brain Computing of SOFT.

Jasmin Léveillé recently completed postdoctoral studies at the Center for Computational Neuroscience and Neural Technology, and in the Center of Excellence for Learning in Education, Science, and Technology at Boston University. His research interests include self-supervised learning, computer vision and neural architectures for high-performance computing. Léveillé received a Ph.D. degree in cognitive and neural systems from Boston University.