Proceedings of the 9th International Conference on Neural Information Processing (ICONIP'02), Vol. 4
Lipo Wang, Jagath C. Rajapakse, Kunihiko Fukushima, Soo-Young Lee, and Xin Yao (Editors)

NEOCOGNITRON CAPABLE OF INCREMENTAL LEARNING

Kunihiko Fukushima

Tokyo University of Technology, Hachioji, Tokyo 192-0982, Japan

ABSTRACT

This paper proposes a new neocognitron that accepts incremental learning, without giving a severe damage to old memories or reducing learning speed. The new neocognitron uses a competitive learning, and the learning of all stages of the hierarchical network progresses simultaneously.

To increase the learning speed, conventional neocognitrons of recent versions sacrificed the ability of incremental learning, and used a technique of sequential construction of layers, by which the learning of a layer started after the learning of the preceding layers had completely finished. If the learning speed is simply set high for the conventional neocognitron, simultaneous construction of layers produces many garbage cells, which become always silent after having finished the learning. The proposed neocognitron with a new learning method can prevent the generation of such garbage cells even with a high learning speed, allowing incremental learning.

1. INTRODUCTION

The author previously proposed a neural network model *neocognitron* for robust visual pattern recognition [1]. It has a hierarchical multilayered architecture similar to the classical hypothesis of Hubel and Wiesel. It acquires the ability to recognize robustly visual patterns through learning. It consists of layers of S-cells, which resemble simple cells in the primary visual cortex, and layers of C-cells, which resemble complex cells. These layers of S-cells and C-cells are arranged alternately in a hierarchical manner. The C-cells in the highest stage work as recognition cells, which indicate the result of the pattern recognition.

To increase the learning speed, the neocognitrons of recent versions have sacrificed the ability of incremental learning. This paper proposes a neocognitron that is capable of incremental learning without reducing the learning speed. The new neocognitron has a modified network architecture and uses a new learning method. The new learning method allows the simultaneous construction of all stages of the network with a fast learning speed, and still accepts an incremental learning. The old memories made by an earlier learning will not be seriously destroyed by subsequent learning.

2. CONVENTIONAL NEOCOGNITRON

2.1. Simultaneous or Sequential Construction

In the neocognitron, the strength of input connections to feature-extracting S-cells is modified during the learning. Each S-cell competes with other cells in its vicinity and has its input connections modified only when it wins the competition. The connections are modified so that the cell responds more strongly to the training stimulus to which the cell becomes a winner. The competition and modification of connections progress for all cells in each layer in parallel.

As for the sequence order of modifying connections of different layers, two alternative methods have been proposed. We will call these methods *simultaneous construction* and *sequential construction*. In the simultaneous construction, which is used in the original versions of the neocognitron [1], learning of all layers in the network progresses simultaneously. In the sequential construction, which is used in most of the recent versions [2], learning starts from the lowest stage and progresses sequentially to higher stages: after the learning of a lower stage has been completely finished, the learning of the succeeding stage begins.

Both simultaneous and sequential constructions have merits and demerits for the conventional neocognitron. The simultaneous construction requires a slow learning speed but can accept incremental learning. On the other hand, the sequential construction can finish learning very fast, but does not accept incremental learning.

The proposed neocognitron allows the simultaneous construction of all stages of the network with a fast learning speed, and still accepts an incremental learning.

2.2. Conventional Learning Methods

Let U_{Sl} be the layer of S-cells in the lth stage of the network. The response of U_{Sl} works as a training stimulus for layer U_{Sl+1} of the succeeding stage.[1] A repeated presentation of training patterns gradually increases the number of cell-planes in U_{Sl}. An increase of cell-planes in U_{Sl} means a change of training stimulus to U_{Sl+1}, even for the same training pattern that have been given previously to the

[1] To be more strict, the response of C-cell layer U_{Cl}, which is a blurred version of the response of U_{Sl}, works as the training stimulus for U_{Sl+1}. To simplify the expression in this section, however, we write as though U_{Sl}, instead of U_{Cl}, is the training stimulus.

input layer of the network. If we express this in a multi-dimensional vector space, the dimension of learning vectors for U_{Sl+1} gradually increases following the progress of learning of U_{Sl}.

Let us first consider the case of simultaneous construction. If the learning speed of U_{Sl} is high, change in response of U_{Sl}, which is caused by the increased number of cell-planes, occurs very fast. Because of the sudden change of signals from presynaptic cells, a cell of U_{Sl+1} often fails to respond to the training pattern, to which the cell used to become a seed cell. Since the cell cannot become a seed cell, the input connections to the cell-plane cannot be modified. The cell-plane fails to adapt to the fast change of layer U_{Sl} and stops responding for ever. Another cell-planes shall be generated now in U_{Sl+1} instead of the silent cell-plane. The silent cell-plane becomes garbage in the network and just consumes a large amount of computation time and memory when the network is installed in a computer.

To avoid the generation of silent garbage cell-planes, the learning speed of the network has to be very slow. Because of a mechanism of shunting inhibition, the output of an S-cell is small when the connections to it are weak [2]. Hence the response of a cell-plane will stay small for a while after its generation, if the learning speed is slow. Then, the response gradually builds up. In other words, the response of U_{Sl} to a training pattern, which becomes the training stimulus for U_{Sl+1}, does not make a rapid change even after the increase in the number of cell-planes in U_{Sl}. Therefore, each cell-plane of U_{Sl+1} can adapt to the increase in dimension of the training vector by shifting its reference vector gradually to the direction of the new training vector of an increased dimension. Although the generation of garbage cells can thus be avoided by a very slow learning speed, a large number of repeated presentations of the same leaning patterns are required before the learning finishes, because of slow building up of responses of the cells.

To increase the learning speed without generating garbage in the network, sequential construction is often used for the learning of the neocognitron of recent versions. Since the learning of U_{Sl+1} starts after the learning of U_{Sl} has completely finished, garbage cells will not be generated in U_{Sl+1} independent of the learning speed of U_{Sl}.

The sequential construction, however, does not accept incremental learning. Suppose an additional set of training patterns be supplied, after a network has already finished learning a certain set of training patterns. If a layer, which has learned the first training set, additionally learns the second training set, new cell-planes will usually be generated in the layer. Hence the layer comes to show different responses even to the patterns of the first training set. This is the same situation as in the case of simultaneous construction with a fast learning speed. Some of the cell-planes of the succeeding layer fail to respond even to the patterns to which they used to respond, and shall become garbage.

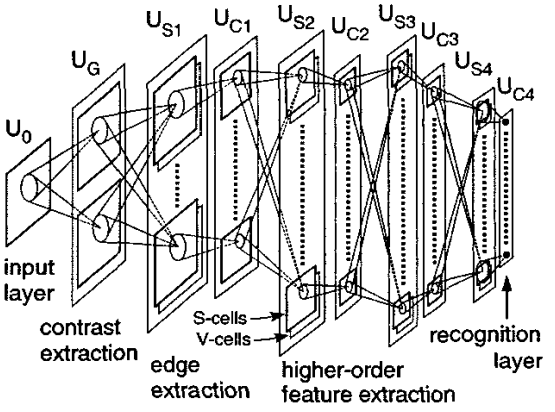

Figure 1: The architecture of the proposed neocognitron.

3. NETWORK ARCHITECTURE

The proposed neocognitron has almost the same network architecture as the neocognitron of a recent version, which is designed for handwritten digit recognition [2]. Only the difference between them resides in V-cell-planes (namely, cell-planes of V-cells). In the conventional neocognitron, each layer of S-cells has only one V-cell-plane, which is used in common for all S-cell-planes of the layer. In the proposed neocognitron, each S-cell-plane has its own V-cell-plane. In other words, each S-cell has its own V-cell.

Figure 1 shows the network architecture of the proposed neocognitron.

The stimulus pattern is presented to the input layer U_0. A layer of contrast-extracting cells (U_G) follows layer U_0. Layer U_G consists of two cell-planes: a cell-plane consisting of cells with concentric on-center receptive fields, and a cell-plane consisting of cells with off-center receptive fields.

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} have been trained using supervised learning [2] to extract oriented edge components 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. An inhibitory surround is introduced around the excitatory connections of the input connections to each C-cell [2].

The S-cells of layers U_{S2}, U_{S3} and U_{S4} are self-organized using the proposed method, which will be discussed later.

Since main difference from the conventional neocognitron [2] resides in the V-cell-planes in S-layers, we will show mathematical expressions of the response of a layer of S-cells only.

Let $u_{Sl}(\boldsymbol{n}, k)$, $v_l(\boldsymbol{n}, k)$ and $u_{Cl}(\boldsymbol{n}, k)$ be the output of S-, V- 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. The outputs of S-cells are given by

$$u_{Sl}(\boldsymbol{n}, k) = \frac{\theta_l}{1-\theta_l} \cdot \varphi\left[\frac{1+\sum_{\kappa=1}^{K_{Cl-1}} \sum_{|\boldsymbol{\nu}|<A_{Sl}} a_{Sl}(\boldsymbol{\nu}, \kappa, k) \cdot u_{Cl-1}(\boldsymbol{n}+\boldsymbol{\nu}, \kappa)}{1+\theta_l \cdot b_{Sl}(k) \cdot v_l(\boldsymbol{n}, k)} - 1\right] \quad (1)$$

Parameter $a_{Sl}(\boldsymbol{\nu}, \kappa, k)$ (≥ 0) is the strength of variable excitatory connection coming from C-cell $u_{Cl-1}(\boldsymbol{n}+\boldsymbol{\nu}, \kappa)$ of the preceding stage. For $l=1$, however, $u_{Cl-1}(\boldsymbol{n}, k)$ stands for $u_G(\boldsymbol{n}, k)$, and we have $K_{Cl-1}=2$. It should be noted here that all cells in a cell-plane share the same set of input connections, hence $a_l(\boldsymbol{\nu}, \kappa, k)$ is independent of $\boldsymbol{n}$. A_{Sl} denotes the radius of summation range of $\boldsymbol{\nu}$, that is, the size of spatial spread of input connections to a particular S-cell. Parameter $b_l(k)$ (≥ 0) is the strength of variable inhibitory connection coming from the V-cell. The positive constant θ_l is the threshold of the S-cell and determines the selectivity in extracting features. Incidentally, equation (1) is the same as that for the conventional neocognitron.

On the other hand, the outputs of V-cells are given by

$$v_l(\boldsymbol{n}, k) = \sqrt{\sum_{\kappa=1}^{K_{Cl-1}} \sum_{|\boldsymbol{\nu}|<A_{Sl}} c_{Sl}(\boldsymbol{\nu}) \cdot \left\{u_{Cl-1}(\boldsymbol{n}+\boldsymbol{\nu}, \kappa)\right\}^2 \cdot \frac{w_l(\kappa, k)}{w_l(1, k)}} \quad (2)$$

where $w_l(\kappa, k)$ represents a weight for signals from the κth C-cell-plane, which will be explained below. Parameter $c_{Sl}(\boldsymbol{\nu})$ represents the strength of the fixed excitatory connections to the V-cell, and is a monotonically decreasing function of $|\boldsymbol{\nu}|$.

It should be noted here that K_{Cl-1}, the number of C-cell-planes, does not stay constant, but increases, during the learning, because simultaneous construction progresses for all stages of the hierarchical network. Each cell-plane of S- and V-cells receives signals only from the C-cell-planes that have been created by that time. As will be discussed later in Section 4, excitatory connections $a_{Sl}(\boldsymbol{\nu}, \kappa, k)$ from layer U_{Cl-1} are increased by an amount proportional to the response of C-cells presynaptic to the seed cell. Hence a cell-plane that was created earlier (say, the cell-plane with $\kappa=1$) has a larger contribution to strengthening the connections than a newly created cell-plane, because it has been presented more frequently to the seed cell. The difference in contribution to the excitatory connections is also reflected to the connections to the inhibitory V-cell by weighting factor $w_l(\kappa, k)$, which is also modified by the learning.

If we express this situation in a multi-dimensional vector space, the output of an S-cell represents the similarity between the reference vector $a_{Sl}(\boldsymbol{\nu}, \kappa, k)$ and input vector $u_{Cl-1}(\boldsymbol{n}+\boldsymbol{\nu}, \kappa)$, where the reference vector is a vector sum of the training vectors [2]. The similarity is measured by an inner product of the reference vector and the input vector normalized by the norm of the two vectors. In an intermediate stage of the network, however, the dimension of the training vectors gradually increases during the learning. A training vector presented earlier usually has a smaller dimension. When measuring the similarity, vector components of the missing dimensions should not be treated as zero, but should be treated as unknown.

By the proposed learning method, which will be discussed below in Section 4, however, vector components of missing dimensions are treated as zero, when a training vector of a smaller dimension is summed to the reference vector. To compensate the missing dimensions of training vectors, the similarity is measured by a *weighted* inner product, and the weight for each component is proportional to the number of presentations of training vectors to the corresponding component of the reference vector.

4. LEARNING METHOD

We adopt simultaneous construction, by which learning of all layers progresses simultaneously in the network.

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 [2].

During the learning, each S-cell competes with other cells in its vicinity, and the winners of the competition become seed cells. Once the seed cells are determined, variable connections $a_l(\boldsymbol{\nu}, \kappa, k)$ and $b_l(k)$ are strengthened depending on the responses of the C-cells presynaptic to the seed cells.

Let cell $u_{Sl}(\hat{\boldsymbol{n}}, \hat{k})$ be selected as a seed cell at a certain time. Variable connections $a_l(\boldsymbol{\nu}, \kappa, \hat{k})$ and weight $w_l(\kappa, \hat{k})$ 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:

$$\Delta a_{Sl}(\boldsymbol{\nu}, \kappa, \hat{k}) = q_l \cdot c_{Sl}(\boldsymbol{\nu}) \cdot u_{Cl-1}(\hat{\boldsymbol{n}}+\boldsymbol{\nu}, \kappa) \quad (3)$$

$$\Delta w_l(\kappa, \hat{k}) = 1 \qquad (1 \leq \kappa \leq K_{Cl-1}) \quad (4)$$

where q_l is a positive constant determining the learning speed. It should be noted here that K_{Cl-1} is not a constant but increases with the progress of learning. The inhibitory connection $b_l(\hat{k})$ is determined directly from the values of the excitatory connections $a_l(\boldsymbol{\nu}, \kappa, \hat{k})$:

$$b_{Sl}(\hat{k}) = \sqrt{\sum_{\kappa=1}^{K_{Cl-1}} \sum_{|\boldsymbol{\nu}|<A_{Sl}} \frac{\left\{a_{Sl}(\boldsymbol{\nu}, \kappa, \hat{k})\right\}^2}{c_{Sl}(\boldsymbol{\nu})}} \quad (5)$$

For other cell-planes from which no seed cell is selected ($k \neq \hat{k}$), all of these values, namely, $a_{Sl}(\nu, \kappa, k)$, $w_l(\kappa, k)$ and $b_{Sl}(k)$, do not changed at this moment.

The method of dual threshold [3] is also used for the learning: Competition among S-cells is based on the responses with a high threshold value θ_l^L, and signals that are sent to the succeeding stage are calculated with a lower threshold value θ_l^R.

S-cells of the highest stage (U_{S4}) are trained using a supervised competitive learning [2]. 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 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. 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. 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.

Competition among S-cells occurs also in the recognition phase, and the label of the maximum-output S-cell of U_{S4} determines the final result of recognition.

5. COMPUTER SIMULATION

5.1. Scale of the Network

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 [4].

The network has the same scale and parameters as the one reported in [2], except the number of V-cell-planes. That is, the total number of cells (not counting inhibitory V-cells) in each layer is: U_0: 65×65, U_G: $71 \times 71 \times 2$, U_{S1}: $68 \times 68 \times 16$, U_{C1}: $37 \times 37 \times 16$, U_{S2}: $38 \times 38 \times K_{S2}$, U_{C2}: $21 \times 21 \times K_{C2}$, U_{S3}: $22 \times 22 \times K_{S3}$, U_{C3}: $13 \times 13 \times K_{C3}$, U_{S4}: $5 \times 5 \times K_{S4}$, U_{C4}: $1 \times 1 \times 10$. 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 during the learning depending on the training set. In each stage except the highest 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 $K_{C4} = 10$ cell-planes corresponding to ten digits, and each cell-plane contains only one C-cell.

The thresholds of S-cells were chosen 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 for the recognition (namely, thresholds for calculating responses of S-cells) were $\theta_2^R = 0.51$, $\theta_3^R = 0.58$ and $\theta_4^R = 0.30$. Those for the learning (namely, thresholds used for the competition) were: $\theta_2^L = 0.66$, $\theta_3^L = 0.67$. As for the highest stage, however, we used $\theta_4^L = 0.75$, instead of $\theta_4^L = 0.30$ that was used for the previous network [2].

5.2. Recognition Rate

To demonstrate that the network accepts incremental learning, we will show how the recognition rate changes depending on two different ways of pattern presentation in the learning.

Experiment A: The network learns a single training set consisting of patterns of all classes, namely, handwritten digits from '0' to '9'. The training set has the same number of patterns from each class. The patterns are randomly sampled from the ETL1 database. To test how the recognition rate changes depending on the total number of patterns in the training set, we prepared training sets consisting of 500, 1000, 2000 and 3000 patterns. The training set of 500 patterns is a subset of 1000, which in tern is a subset of 2000, and so on. The training set is presented to the network repeatedly until the increase in K_{S4} stops. After having finished the learning, the recognition rate of the network is measured using a blind test set. The blind test set consists of 3000 digits (300 patterns from each class), which are also randomly sampled from the ETL1 database, but there is no overlapping of patterns between the training and the test sets.

Experiment B: The training set used for experiment A is divided into two: one containing digits from '0' to '4', and the other from '5' to '9'. The network initially learns the first training set. After having finished learning the first training set, the network learns the second training set. After having finished the first and the second half of the learning, the recognition rate of the network was measured using the blind test set. It should be noted here that no more additional learning for the first training set is made after starting the second learning.

Table 1 summarizes the recognition rates under these various conditions. The recognition rates for '0'–'4' and '5'–'9' have been separately counted, as well as for all '0'–'9' patterns, in the table. The numbers of cell-planes generated when the learning has finished are also listed.

As can be seen from the table, the recognition rate does not decrease so much even if the training is divided in to two steps (experiment B). When we used 3000 training patterns, for example, recognition rates after experiment A and B were 98.1% and 98.3%, respectively. The numbers of cell-planes are smaller for the latter case.

These results show that the memory of the first learning is not seriously destroyed by the second learning. In other words, the proposed neocognitron accepts incremental learning without giving a serious damage to old memories.

Table 1: Recognition rates of the network for experiments A and B using training sets of different sizes.

Training set (number of training patterns)		Recognition rate (%) for Test set (for Training set)			Scale of the network		
		'0'–'4'	'5'–'9'	'0'–'9'	K_{S2}	K_{S3}	K_{S4}
500	together	93.9 (100.)	95.0 (100.)	94.5 (100.)	29	77	83
1000	together	95.3 (100.)	97.7 (100.)	96.5 (100.)	37	88	128
2000	together	97.3 (100.)	97.9 (100.)	97.6 (100.)	44	105	189
3000	together	97.9 (100.)	98.4 (100.)	98.1 (100.)	46	118	205
500	after 1/2	98.3 (100.)	—	—	24	49	29
	after 2/2	93.9 (98.4)	95.9 (100.)	94.9 (99.2)	28	60	64
1000	after 1/2	98.7 (100.)	—	—	26	60	37
	after 2/2	94.7 (98.8)	98.1 (99.8)	96.4 (99.3)	30	76	85
2000	after 1/2	99.1 (100.)	—	—	32	72	43
	after 2/2	96.2 (98.5)	98.5 (100.)	97.4 (99.3)	42	91	116
3000	after 1/2	99.3 (100.)	—	—	35	78	52
	after 2/2	98.0 (99.3)	98.7 (98.7)	98.3 (99.6)	48	98	125

Incidentally, if simultaneous construction was applied to the conventional network with the same parameters, extremely large number of cell-planes were generated: (K_{S2}, K_{S3}, K_{S4}) = (46, 244, 452). Such a large number of cell-planes would not be practically acceptable. Probably many garbage cells were generated in the network. The recognition rate was 98.3% and was almost the same as that for the proposed network,

Fig. 2 shows a response of the network that has finished the learning by experiment B using the training set of 3000 patterns. The responses of layers U_0, U_G, U_{C1}, U_{C2}, U_{C3} and U_{C4} 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.

Acknowledgement The author thanks Katsuyoshi Yanagawa, graduate student at the University of Electro-Communications (now at Hitachi Software Engineering Co.), for his great contribution to this work. He first proposed the use a separate V-cell-plane for each S-cell-plane [5]. This work was partially supported by Grant-in-Aid-for-Scientific-Research #14380169, and Special Coordination Fund for Promoting Science and Technology, both from the Ministry of Education, Culture, Sports, Science and Technology, Japan.

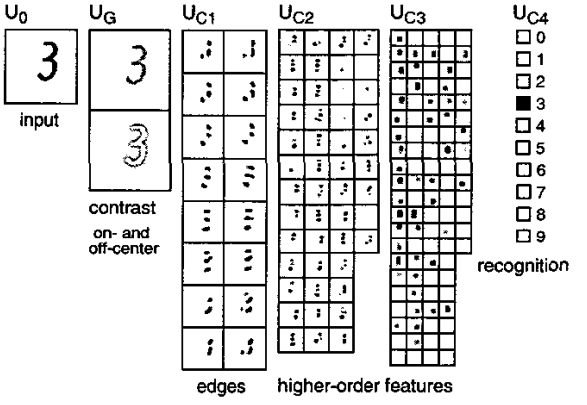

Figure 2: An example of the response of the neocognitron. It has learned a training set consisting of '0'–'4' first, then a training set consisting of '5'–'9'. The input pattern is recognized correctly as '3'.

6. REFERENCES

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

[2] K. Fukushima: "Neocognitron for handwritten digit recognition", *Neurocomputing*, in print (2002).

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

[4] ETL1 database:
`http://www.etl.go.jp/~etlcdb/index.htm`

[5] K. Yanagawa, K. Fukushima, T. Yoshida: "Additional learnable neocognitron" (in Japanese), *Technical Report of IEICE*, No. **NC2001**-176 (2002).