

NEOCOGNITRON'S PARAMETER TUNING BY GENETIC ALGORITHMS

DAMING SHI*

*Department of Electronics and Computer Science,
University of Southampton, Highfield, Southampton, SO17 1BJ UK*

CHUNLEI DONG

*School of Computing, National University of Singapore,
Lower Kent Ridge Road, Singapore 119260*

DANIEL S. YEUNG

*Department of Computing, The Hong Kong Polytechnic University,
Hung Hom, Kowloon, Hong Kong*

Received 13 May 1999

Revised 3 August 1999

Accepted 28 August 1999

The further study on the sensitivity analysis of Neocognitron is discussed in this paper. Fukushima's Neocognitron is capable of recognizing distorted patterns as well as tolerating positional shift. Supervised learning of the Neocognitron is fulfilled by training patterns layer by layer. However, many parameters, such as selectivity and receptive fields are set manually. Furthermore, in Fukushima's original Neocognitron, all the training patterns are designed empirically. In this paper, we use Genetic Algorithms (GAs) to tune the parameters of Neocognitron and search its reasonable training pattern sets. Four contributions are claimed: first, by analyzing the learning mechanism of Fukushima's original Neocognitron, the correlations amongst the training patterns are claimed to affect the performance of Neocognitron, tuning the Neocognitron's number of planes is equivalent to searching reasonable training patterns for its supervised learning; second, a GA-based supervised learning of the Neocognitron is carried out in this way, searching the parameters and training patterns by GAs but specifying the connection weights by training the Neocognitron; third, other than traditional GAs which are unsuitable for the large searching space of training patterns set, the cooperative coevolution is incorporated to play this role; fourth, an effective fitness function is given out when applying the above methodology into numeral recognition. The evolutionary computation in our initial experiments is implemented based on the original training pattern set, e.g. the individuals of the population are generated from Fukushima's original training patterns during initialization of GAs. The results prove that our correlation analysis is reasonable, and show that the performance of a Neocognitron is sensitive to its training patterns, selectivity and receptive fields, especially, the performance is not monotonically increasing with respect to the number of training patterns, and this GA-based supervised learning is able to improve Neocognitron's performance.

1. Introduction

Fukushima's Neocognitron is well-known for its performance in visual pattern recognition.^{1–5} The Neocognitron divides the visual patterns into many

sub-patterns (features) and distributes them into different feature planes which are linked in a hierarchical manner.

The Neocognitron consists of two types of neurons, i.e., S-cells which undertake feature

*Corresponding author
E-mail: ds99r@ecs.soton.ac.uk

extraction, and C-cells which take charge of the toleration of the deformation. There are two important parameters for each S-cell, one is its receptive field which specifies the number of its input interconnection, the other is its selectivity which is a threshold to control its response to features.

Many researchers, including Fukushima himself, have made efforts to improve the performance of the Neocognitron. In our previous work, it was proved that the Neocognitron is sensitive to three parameters, i.e., number of planes, receptive field and selectivity.⁸ This paper is aimed to tune these parameters to reach the goal.

In the conventional neocognitron, the amount of blurring produced by the C-cells is uniform in the receptive field of each S-cell. An S-cell would accept a much larger deformation if a nonuniform blurring could be produced in such a way that a larger blurring is generated in the periphery than at the center of the receptive field. This is desirable as discrepancies between a training pattern and a deformed stimulus pattern usually becomes larger in the periphery than at the center of the receptive field. In order to produce such a nonuniform blurring economically, Fukushima proposed an improved Neocognitron with a dual C-cell layer.⁹

Considering the difficulty of interpreting and manipulating the interlayer connection denoted by connection weights, multi-layered attributed graphs were mapped to this hierarchical structure in our previous work,¹⁰ as well as a rule-based system was embedded in.^{11,12} The rule-based system helps the Neocognitron to initialize its weights instead of training. The parameters in the rule-based system are assigned by domain expert, which are then mapped into the Neocognitron as its connection weights.

Troatin *et al.* propose methods to build the Neocognitron dynamically, as well as to lower the threshold (i.e., the selectivity) automatically.¹³ The whole network architecture of the Neocognitron is built from setting zero planes in each layer at the beginning. The size adaptation of the network for a given application helps users reduce the effort in parameter adjustment. They also take advantage of a feedback signal to evaluate the effect of decreasing threshold, and then to tune thresholds based on the evaluation. The result of recognition after this threshold tuning is claimed to be as good as the re-

sult using manual adjustment of thresholds. However, the detail mechanism of the feedback signal has not been given out, and they confessed that the automatic adjustment of thresholds was difficult in their proceeding research.¹⁴

Regarding the Neocognitron as a linear correlation classifier, Hildebrandt proposed his optimal closed-form training to the Neocognitron, the thresholds of cells were modified, so that the Neocognitron became an optimal classifier.¹⁵ Lovell got failed when applying Hildebrandt's optimal closed-form training,¹⁶ while, Hildebrandt explained that the training should not be applied to every layer of a network but the output layer only. As a matter of fact, Hildebrandt did not take care that the information lost layer by layer when propagating in the network.

Van Ooyen and Niehuis introduced a new parameter to improve the discriminating power of the Neocognitron.¹⁷ In fact, this parameter serves as an inhibitory signal, which suppresses the firing of neuron to the repeatedly stimulated features. Most neural networks memorize the training information in their weights. According to the Hebbian learning rule, the more times the feature is presented in the past, the easier the same feature are detected by the neural networks. However, such an opposite idea, which sets a difficult way for the repeatedly stimulated features, enables the network to identify these distinctive features which help discriminating different patterns.

Simulating biological evolution, Genetic Algorithms offer a class of global search techniques based on the learning process of a population of individuals. Each individual represents a potential solution to a given problem. Some researchers are being engaged in the approaches to construction of a neural networks based on Genetic Algorithms.^{18,19} The general method is to construct a linear chromosome and a 2-dimensional directed graph to represent the network, the chromosome designating the genetic operators are applied to which position, while, the directed graph indicating which genes will be involved in this operation correspondent with this position.

On the basis of the sensitivity analysis of the neocognitron model, we had presented a parameter tuning (namely selectivity) methodology for the Neocognitron in our previous work,²⁰ as well as its application to handwritten numeral

recognition. The performance improvement shows that it is more promising to tune parameters than set them manually. Nevertheless, a more efficient evolutionary algorithm is required to search reasonable parameters and topology which would be difficult by traditional Genetic Algorithms because of the large number of genes involved.

In traditional GAs, each individual is a potential complete solution to the problem, but in Cooperative Coevolutionary algorithms^{21–24} each individual represents only a partial solution to the problem. The goal of an individual is thus to optimize one piece of the solution and cooperate with other partial solutions that optimize other pieces so as to search for the entire solution effectively. Moreover, by maintaining the different solution pieces in a single population, the population does not converge to a single individual, e.g. diversity is maintained. Moriarty and Miikulainen set up a symbiotic adaptive neuroevolution (SANE) system to coevolve a population of neurons that cooperate to form a functioning network.²⁵ In this process, neurons assume different but overlap roles, resulting in a robust encoding of control behavior. SANE is shown to be more efficient and more adaptive. It also maintain higher levels of diversity than the more common network-based population approaches.

Evolutionary computation for interconnection weights must concern real-valued GA. Maniezzo enhanced the exploration abilities by providing the search-space reducing evolution of both coding granularity and network topology.²⁶ In his contribution, a connectivity is coded by presence/absence bits (the connectivity bits) relative to each possible arc of the complete graph and the first byte of the string specifies the number of bits (the granularity) according to which the weights of the present connections have been codified. However, the Neocognitron is a complicated network which involves thousands of weights, so that, it is unfeasible to search for the weights by GAs.

This paper will incorporate cooperative coevolutionary algorithms into supervised learning of the Neocognitron to search for better parameter values and training patterns. Also regarding the Neocognitron as a correlation classifier, we will evaluate the training patterns by correlation analysis. It is organized as follows: in Sec. 2, two important parameters of the Neocognitron, i.e., receptive field and

selectivity, are discussed in detail. In Sec. 3, after analyzing the learning mechanism of the Neocognitron, we draw a conclusion that tuning the number of S-cell-planes of the Neocognitron is actually searching for the training patterns for supervised learning. In Sec. 4, we point out that the original Neocognitron lacks correlation analysis when designing the training patterns. Then we describe a methodology to incorporate the cooperative coevolution algorithm to find the better Neocognitron in Sec. 5. Finally, the methodology is applied to handwritten numeral recognition in Sec. 6, and the experimental results showed our GA-based supervised learning of Neocognitron is able to improve the performance of Neocognitron.

2. Receptive Field and Selectivity of Neocognitron

In this section, we will introduce the two parameters of Neocognitron, i.e., receptive field and selectivity. However, number of planes, which are closely relevant to the training patterns, will be discussed in the next section.

The Neocognitron tries to model the visual system which can recognize an object even the object shifts its position and is slightly deformed. The Neocognitron can learn from the stimulus patterns with shift invariance ability and can be generalized to recognize any unseen deformed version of these patterns.

The Neocognitron consists of simple cells, called *S-cells*, and complex cells, called *C-cells*. In Fig. 1,

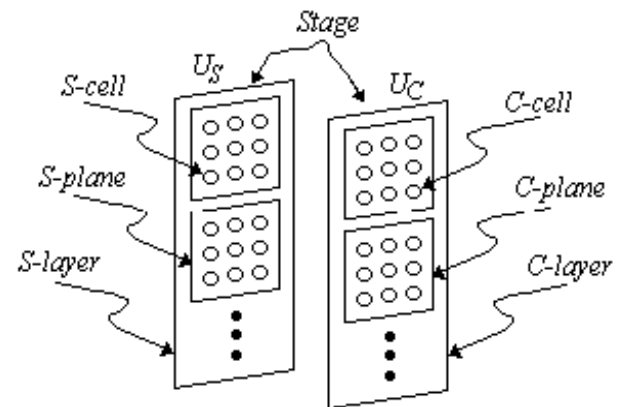


Fig. 1. Hierarchical relationship among cell, plane, layer and stage in the Neocognitron.

S-cells are grouped together to form *S-planes* which are combined to form an *S-layer*, denoted as U_s . The similar relationship exists among C-cells, *C-planes*, *C-layer*(U_c). S-layer and C-layer are put together to form an intermediate stage of the network. In the input stage, the only layer involved is a 2-dimensional plane U_0 which accepts 2-dimensional visual patterns directly. It simplifies the knowledge representation for input. In fact, the feature extraction step has been embedded in the connection scheme of the Neocognitron model. For character recognition, the output stage consists of S-layer and C-layer where every C-plane has only one C-cell which represents a category of characters.

2.1. Receptive field

In the Neocognitron, only the input interconnections to S-cells are variable and modifiable and the input interconnections to other cells are fixed and unmodifiable. Fukushima use the notation $u_{sl}(k_l, \mathbf{n})$ to represent the output of an S-cell in the k_l th S-plane in the l th stage, and $u_{cl}(k_l, \mathbf{n})$ to represent the output of a C-cell in the k_l th C-plane in that stage, where \mathbf{n} is the 2-dimensional coordinates representing the position of these cell's *receptive fields* on the input layer.

From the biological perspective, the visual stimulus light passes through the eyes and hits the retina. The retina ganglion cell collect the signals from a group of photoreceptors which form the receptive field of the cell. The concept of the

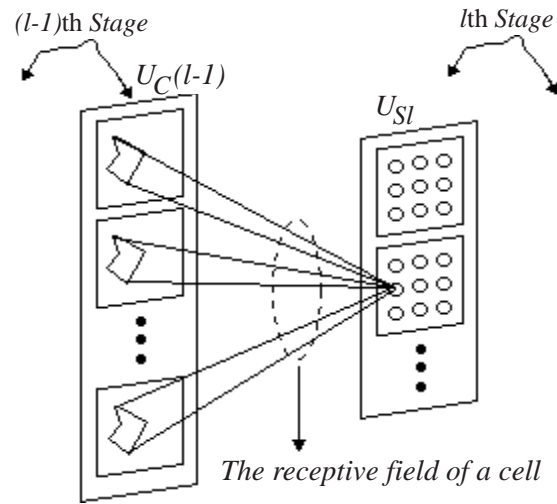


Fig. 2. Illustration of the receptive field of a cell.

receptive field of a cell in the Neocognitron comes from this biological phenomenon. The retina ganglion cell sends signals to lateral geniculate body (LGB) so that the signals are arranged correctly for transmission in the visual cortex. This process is simulated by an input layer in the Neocognitron illustrated in Fig. 2.

The size of the receptive field is set to be small in the first stage and to increase with the depth l .

2.2. Selectivity

In the Neocognitron, S-cells have inhibitory inputs with a shunting mechanism. The output of an S-cell of the k_l th S-plane in the l th stage is given by:²

$$u_{sl}(k_l, \mathbf{n}) = r_l \bullet \varphi \left[\frac{1 + \sum_{k_{l-1}=1}^{K_{l-1}} \sum_{\gamma \in A_l} a_l(k_{l-1}, \gamma, k_l) \bullet u_{C(l-1)}(k_{l-1}, \mathbf{n} + \gamma)}{1 + \frac{r_l}{1 + r_l} \bullet b_l(k_l) \bullet \nu_{C(l-1)}(\mathbf{n})} - 1 \right] \quad (1)$$

where $\phi[x] = \max(x, 0)$. $a_l(k_{l-1}, \gamma, k_l)$ and $b_l(k_l)$ represent the values of the excitatory and inhibitory variable interconnecting coefficients, respectively. Figure 3 illustrate the input, output, interconnection and activation function of a S-cell.

Parameter γ_l in Eq. (1), called selectivity, controls the intensity of the inhibition. The larger the

value γ_l is, the more selective becomes the cell's response to its specific feature.

The inhibitory cell V-cell, $\nu_{C(l-1)}(\mathbf{n})$, which is sending an inhibitory signal to cell $u_{sl}(k_l, \mathbf{n})$, receives its input interconnections from the same cells as $u_{sl}(k_l, \mathbf{n})$ does, and yields an output proportional

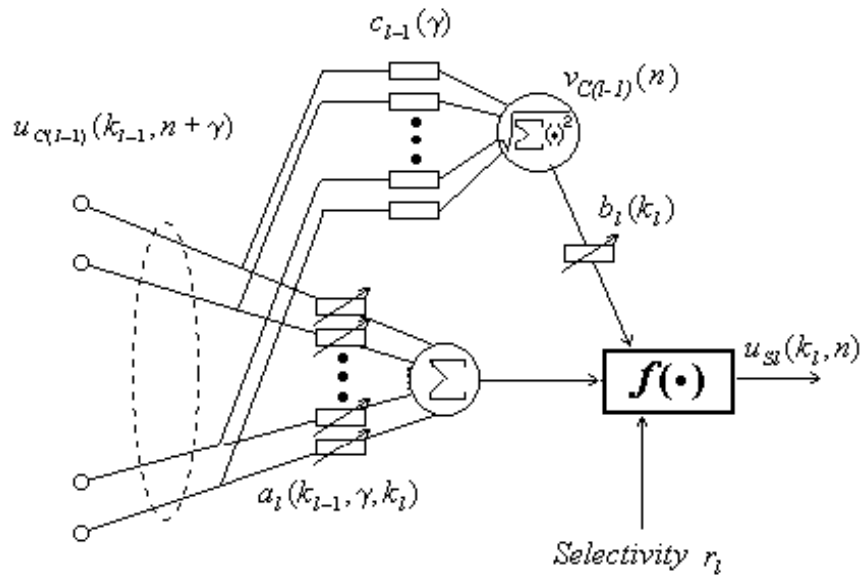


Fig. 3. The input, output, interconnection and activation function of a S-cell.

to the weighted root-mean-square of its inputs:

$$\begin{aligned} & \nu_{C(l-1)}(\mathbf{n}) \\ &= \sqrt{\sum_{k_{l-1}} \sum_{\gamma \in A_l} c_{l-1}(\gamma) \cdot u_{C(l-1)}^2(k_{l-1}, \mathbf{n} + \gamma)}. \end{aligned}$$

The values of fixed interconnections $c_{l-1}(\gamma)$ are determined so as to decrease monotonically with respect to $|\gamma|$ and to satisfy $\sum_{k_{l-1}} \sum_{\gamma \in A_l} c_{l-1}(\gamma) = 1$.

3. Supervised Learning and Training Patterns

In this section, we will analyze the mechanism of supervised learning of Neocognitron, and then we can see that the topology of Neocognitron is actually specified by the training patterns.

3.1. Supervised learning of the Neocognitron

In the early version of the Neocognitron, unsupervised learning (learning-without-a-teacher) predominate. Unsupervised learning which is utterly on the basis of self-organization is performed with two principles: 1. *Reinforcement of Maximum-Output Cells*. This is a winner-take-all strategy, only the one which is responding the strongest has its input connections reinforced. 2. *Development of Iterative Connections*. The maximum-output cell in each cell-plane,

not only has itself reinforced, but also controls the reinforcement of neighboring cells. As a result, connections develop iteratively in a cell-plane, hence, all the S-cells in a cell-plane grow to extract the same feature but in different position.

In the case of the above unsupervised learning, it takes considerable time to automatically find out seed cell which is the maximum-output amongst all of the cells in a layer when training a pattern. On the other hand, handwritten numeral recognition performs the classification not only on the basis of similarity in shape but also on the basis of certain convention, in other words, some case of geometrically similar patterns should not be classified into same categories, and vice versa.

The characteristics of supervised learning can be listed as follows:

1. The training patterns as well as their seed cells are set by the supervisor.
2. Each S-cell-plane is manually mapping to a training pattern which is its feature to be extracted.
3. Which S-cell-plane joining which C-cell-plane is also set by the supervisor.

The first two points will be discussed in detail in the next section. Now, let's explain the last one. In the case of character recognition, input characters have to be classified not only on the basis of geometrical similarity but also on the basis of customs by which some particular kinds of large deformation

are admitted. Sometimes when such deformation is too large, a single S-cell-plane is not enough to extract deformed versions of a feature. In such a case, another S-cell-plane is used to extract a deformed version of the feature, and the output from these S-cell-planes are made to converge to a single C-cell-plane.

3.2. Training patterns

We can see that, the topology of the Neocognitron is closely correspondent to the training patterns in supervised learning. Each S-cell-plane takes charge of extracting its own features, i.e., the training patterns set in advance by the supervisor. Fukushima’s original Neocognitron, which consists of four stages, the cases are simple and clear for U_{s1} and U_{s4} , but for U_{s2} and U_{s3} , requires many patterns to tolerate the deformations and position shift. It is not only labor extensive but also incomplete to select training patterns manually. Figures 4(a) and 4(b) show the original training patterns for U_{s2} and U_{s3} . Obviously, it is unconvincing that such patterns have included all the cases of the deformations and position shift.

The Neocognitron implements feature extraction by mapping a group of training patterns to a specified S-cell-plane. Training patterns in supervised learning are so important that they should be studied by theoretical and experimental analysis. Unfortunately, no research has been done on that.

4. Correlation Analysis Amongst the Training Patterns

Neocognitron is essentially a correlation classifier, in which an iterative procedure is used to compute the templates and thresholds corresponding to a given set of training patterns. A good Neocognitron should be satisfied with:

1. The correlations among the training patterns for the same category should be reasonably large.
2. The correlations among the training patterns for the different categories should be reasonably small.
3. A training pattern is unnecessary if the correlation is too large between it and some others.

You can see that designing the training patterns is considerably difficult. Fukushima has given out a set of training patterns based on his many-year studies. Nevertheless, the training patterns must conform to the above criteria.

To compute the correlations among training patterns, let’s give a definition first.

Definition 1

Traverse feature. The number of traverse times when a scan line crosses the image.

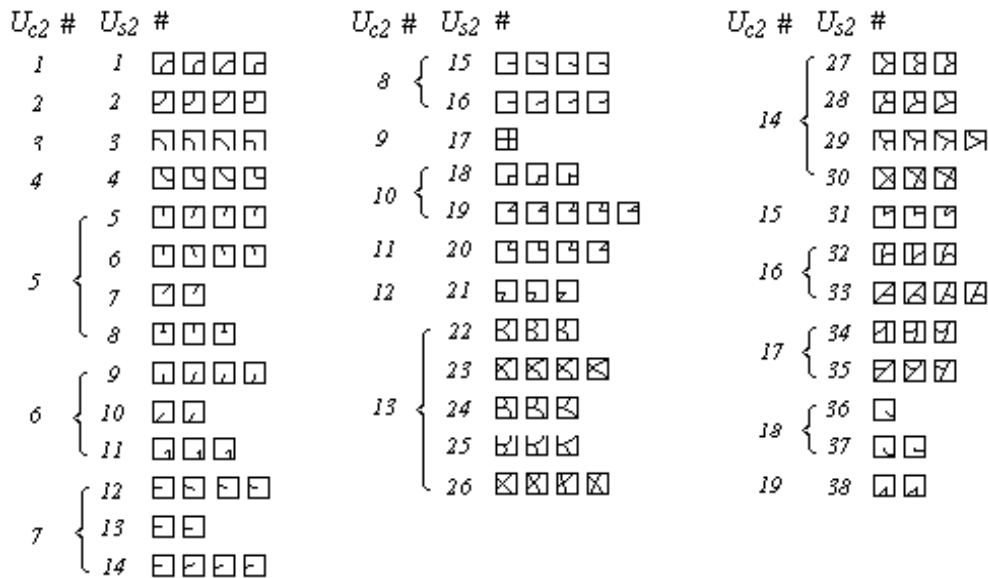


Fig. 4(a). Fukushima’s training patterns for U_{s2} .

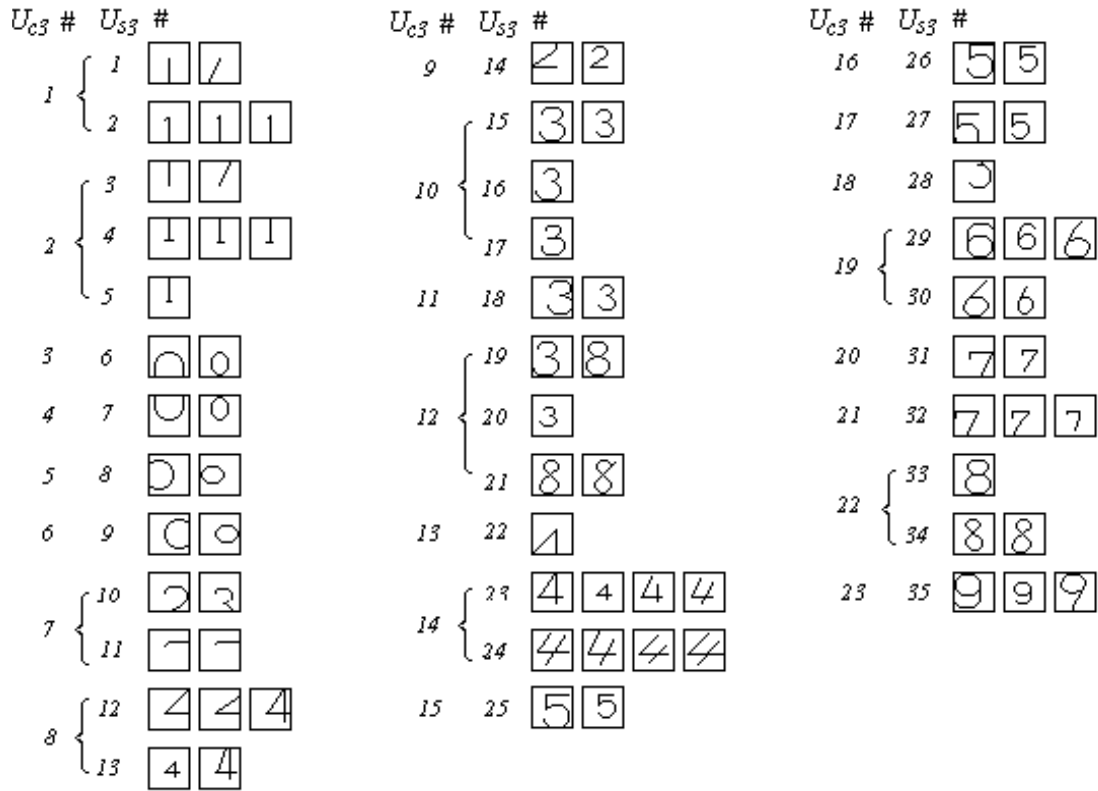


Fig. 4(b). Fukushima's training patterns for U_{s3} .

As shown in Fig. 5, from the $N \times N$ dot-matrix F , we can obtain the N -dimensional horizontal traverse feature vector T_H and N -dimensional vertical traverse feature vector T_V :

$$T_H = [t_H(0), t_H(1), \dots, t_H(N-1)]^T$$

$$T_V = [t_V(0), t_V(1), \dots, t_V(N-1)]^T.$$

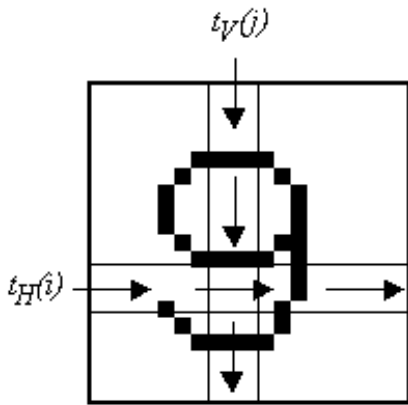


Fig. 5. Traverse feature.

Taking into consideration the translation and revolution of the image, a Rapid transformation should be implemented. Let the image size N be normalized to 2^l , and $T = \{T_H, T_V\}$, then the Rapid transformation of T can be divided into $M = l + 1$ steps, while the R th and $(R-1)$ th steps meet the relationship as below:

$$\begin{cases} T_{2i}^{(R)} = |T_i^{(R-1)} + T_{i+M/2}^{(R-1)}| \\ T_{2i+1}^{(R)} = |T_i^{(R-1)} - T_{i+M/2}^{(R-1)}| \end{cases}$$

where, $i = 0, 1, 2, \dots, (M/2 - 1)$.

Now, $\{R(0), R(1), \dots, R(M-1)\}$ is the feature sequence after Rapid transformation. For each $R(u)$ ($u = 0, 1, \dots, M-1$), it is a linear combination of original functions $T(i)$, and it can comprehensively express the features of a image. For example, $R(0) = \frac{1}{M} \sum_{i=0}^{M-1} T(i)$ reflects the complexity of a numeral image. Therefore, Rapid transformed traverse features can describe a numeral image, and are insensitive to translation and slight revolution, hence, they are suitable for correlation analysis among the training patterns of Neocognitron.

Correlation between two vectors X and Z can be defined to be the inner product of them divided by their magnitudes: $S(X, Z) = \frac{X^T Z}{\|X\| \cdot \|Z\|}$. It is in fact the cosine of the angle between these two vectors. More $S(X, Z)$ tending to the value 1, more similar the two vectors.

Correlation analysis amongst the training patterns is the factor why we should improve Fukushima's original Neocognitron, and it also can be used to evaluate the performance of a Neocognitron.

5. GA-Based Supervised Learning of the Neocognitron

In traditional GAs, each individual in the population represents a complete solution that is independently of others in the population. Applying traditional GAs to the Neocognitron to search for a better topology is shown as Fig. 6.

There are two problems with this method: 1. GAs tend to focus the search toward a single dominant individual. 2. The length of a chromosome is too long to undertake an efficient evolution.

This section describes how to incorporate cooperative coevolution into the supervised learning of

the Neocognitron. When searching for the training patterns by cooperative coevolution, a training pattern is represented by its own chromosome which is a partial solution of the problem. And several such chromosomes cooperate to form a complete solution.

5.1. Work mode

The work mode of supervised learning incorporated with cooperative coevolution can be illustrated as Fig. 7.

As aforementioned, tuning the number of S-cell planes of the Neocognitron is equal to searching for reasonable training patterns in supervised learning. Since no single training pattern can perform the whole task alone, the training pattern must optimize one aspect of the Neocognitron. In other words, a single training pattern will specify only one case of the deformation or positional shift of the visual image.

It can be divided into two level when cooperative coevolution is incorporated into training the Neocognitron, i.e., training pattern level and network level. At network level, each individual select some training patterns from the population of training patterns to

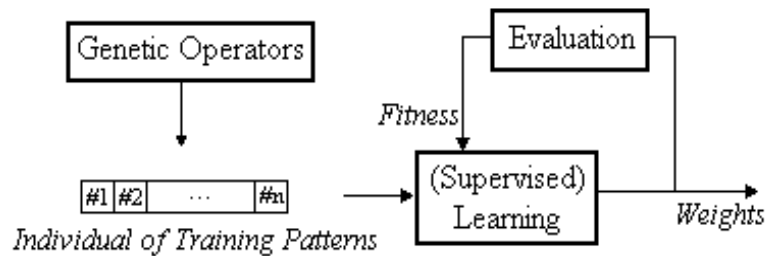


Fig. 6. Illustration of supervised learning for the neocognitron by traditional GAs.

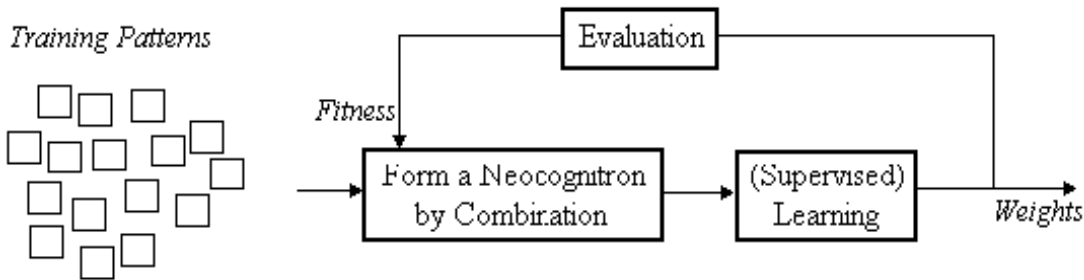


Fig. 7. Illustration of supervised learning for the Neocognitron incorporated with cooperative coevolution.

form a potential network; at training pattern level, there is cooperation amongst the individuals to form a good network, as well as competition because of limited resource (limited number of S-planes and C-planes).

Moriarty and Miikkulainen's basic steps in one generation²⁵ can be applied to our GA-based supervised learning of Neocognitron as follows:

1. Clear the fitness value of each training pattern and network.
2. Select ξ individuals from the training pattern population to form a Neocognitron.
3. Training layer by layer like conventional method to obtain the interconnection weights.
4. Evaluate the performance indicator of this Neocognitron by testing sample set.
5. Assign the performance indicator as the fitness of the individual at network level.
6. Repeat steps 2 to 4 for each individual in network population.
7. Assign each training pattern the evaluation of the best five networks in which it participated.
8. Perform crossover and mutation operations on populations at training pattern level and network level, respectively.

5.2. Representation

There are two types of chromosome corresponding to the above two levels respectively. At training pattern

level, any training pattern can be represented by a chromosome illustrated in Fig. 8, which consists of the coordinate of the receptive field of its seed-cells and its binary dot matrix.

At network level, each chromosome consists of receptive field, selectivity and training patterns for every layer, such like Fig. 9.

5.3. Fitness function

Before we derive the fitness function of GAs, some variables are defined as follows: (i) c which presents the number of correctly recognized inputs. (ii) r which represents the number of rejected inputs. For a given input, if the winner of the output layer scores quite small, the input will be rejected. (iii) m which represents the number of misclassified inputs. Furthermore, the rules for designing a fitness function which represents a performance indicator ρ are:

- (1) When all digits are correctly recognized, ρ should be 1.
- (2) When all digits are rejected, ρ should be 0.
- (3) When all digits are misclassified, ρ should be 0.
- (4) When no digits is correctly recognized, ρ should be 0.
- (5) When c increases, ρ increases, and when c decreases, ρ decreases.
- (6) When r increases, ρ decreases, and when r decreases, ρ increases.

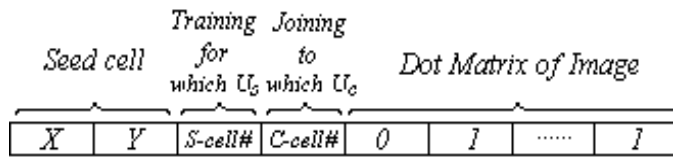


Fig. 8. Chromosome at training pattern level.

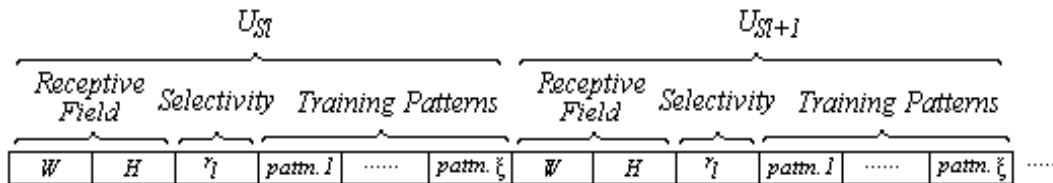


Fig. 9. Chromosome at network level.

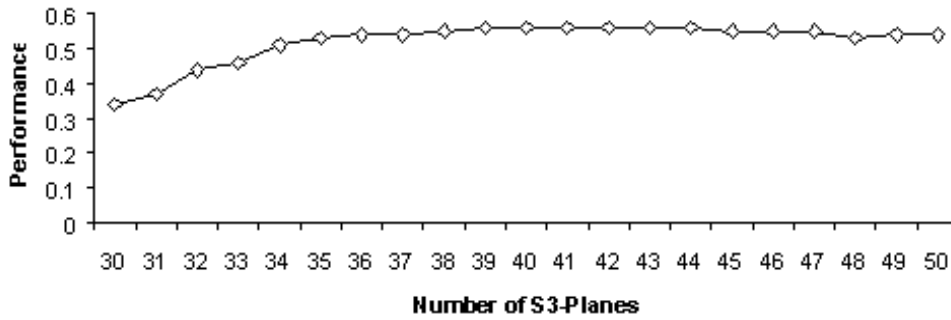


Fig. 10. Number of planes versus the performance of Neocognitron.

- (7) When m increases, ρ decreases, and when m decreases, ρ increases.
- (8) The preference order of three types which an input digit belongs to $c \succ r \succ m$, where $a \succ b$ means the user prefers a to b . Why the user prefers r to m is that the misclassification results in a fatally wrong decision, but rejection is good for other decision system to work in this case.

i.e.

$$\begin{cases} \rho \propto c \\ \rho \propto (1 - r) \\ \rho \propto (1 - m) \end{cases} \quad (2)$$

or

$$\rho \propto c(1 - r)(1 - m). \quad (3)$$

Since the range of the terms on the right hand side of Eq. (2) is $[0, 1]$, ρ could become unnecessarily small. For example, when half of the digits are correctly recognized and the other half are all rejected, the value of ρ is expected near 0.5, but it yields 0.25 only. When $c \succ r \succ m$ is taken into consideration, a possible proportional constant term could be $(1 + \frac{c-1}{2})(1 - \frac{r}{2})$. Then Eq. (3) becomes:

$$\rho = \frac{c(1 - r)(1 - m)}{\left(1 + \frac{c-1}{2}\right)\left(1 - \frac{r}{2}\right)}. \quad (4)$$

The performance indicator ρ will play the role of fitness function of GAs.

6. Experiments

Our experiments implement the evolution on the basis of the Fukushima's original training patterns set,

that is, during the procedure of initializing the population, any individual is randomly selected from the original training patterns with mutation in random positions.

The performance can be assessed by these two sets of testing samples, one is from Fukushima,⁶ the other is from Lovell.¹⁶ In fact, the first set is used to calculate the fitness during evolutionary computation, while, the second one is used to assess the whole performance. Because the training patterns for U_{s1} and U_{s4} are clear and simple, they are not concerned during evolution.

To analyze the relationship between the number of the plane and the performance of Neocognitron, let's observe the performance of the individuals at network level, in which the number of U_{s3} plane is from 30 to 50, as shown in Fig. 10.

We can see that the performance doesn't grow with the increasing the number of planes. Because in the supervised learning of Neocognitron, every training pattern will be presented to the input layer to train for the same times. If a training pattern is very similar to another one, it will make the Neocognitron more sensitive to this kind of feature.

So far, out of 6,000 different individuals at network level, five solutions have been found better than original one, and the best solution is given in Table 1 and Fig. 11.

Now, let's analyze the correlations amongst our new training patterns versus Fukushima's training patterns for U_{s3} . Provided that we have got M training patterns, in which there are N_W ones belonging to a S-cell-plane W , the correlation amongst the N_W training patterns of W is called S_i , whereas, the correlation between the patterns of W and the others belonging to different cell-planes is called S_o .

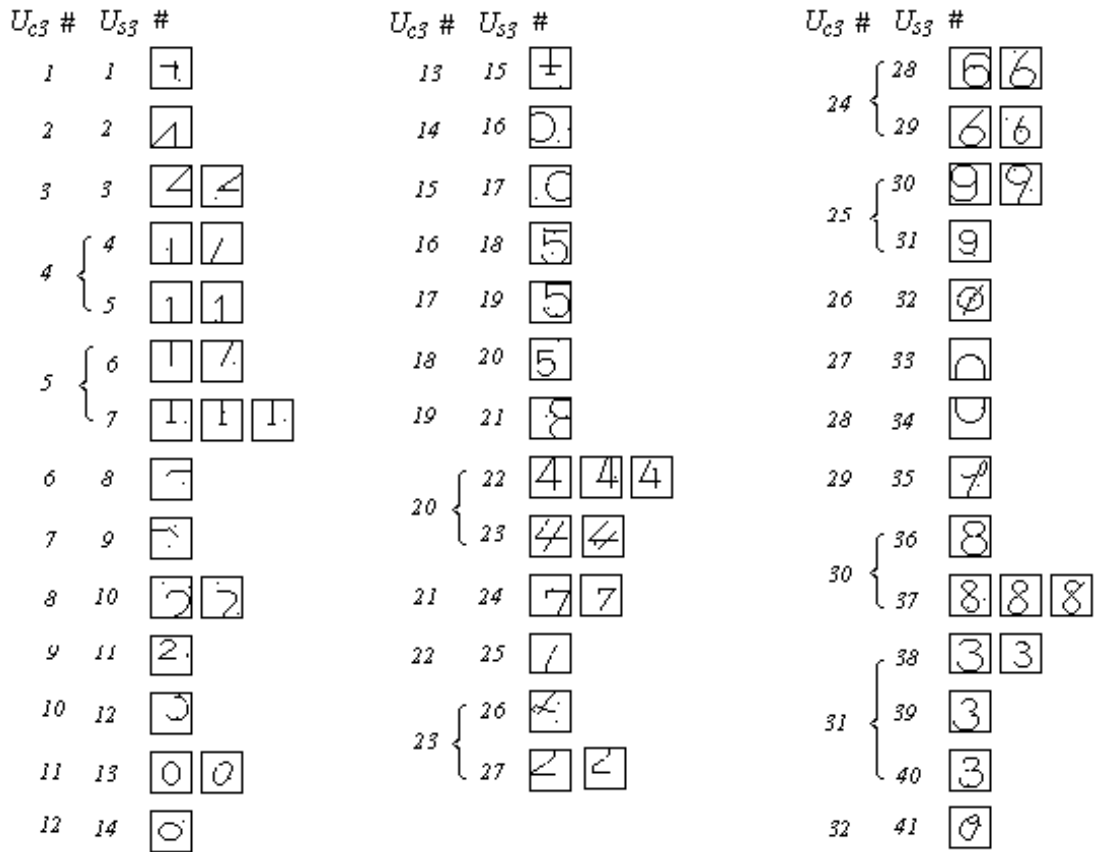


Fig. 11. Training patterns for U_{s3} affiliated to Table 1.

Table 1. The best solution out of 6,000 different individuals at network level.

Neocognitron	Receptive fields	Selectivity	Training patterns	Performance to Fukushima's test set ⁵	Performance to Lovell's test set ¹⁶
	A_1 A_2 A_3 A_4	γ_1 γ_2 γ_3 γ_4	for U_{s2} for U_{s3}	ρ c r m	ρ c r m
"Original"	3*3 5*5 5*5 5*5	1.7 4.0 1.0 1.0	Fig. 4(a) Fig. 4(b)	0.43 0.53 0.13 0.34	0.43 0.50 0.26 0.24
"GA-based"	3*3 5*7 3*5 5*5	0.8 4.0 1.7 1.1	Fig. 4(a) Fig. 11	0.50 0.59 0.15 0.26	0.56 0.63 0.18 0.19

S_i and S_o can be computed as follows:

$$S_i = \begin{cases} 1, & \text{if } (N_W = 1) \\ \frac{1}{(N_W - 1)!} \sum_i \sum_j S(X_i, Z_j)|_{X_i, Z_j \in W}, & \text{otherwise} \end{cases} \quad (5)$$

$$S_o = \frac{1}{N_W \bullet (M - N_W)} \sum_{i=1}^{N_W} \sum_{j=1}^{M-N_W} S(X_i, Z_j)|_{X_i \in W, Z_j \notin W} \quad (6)$$

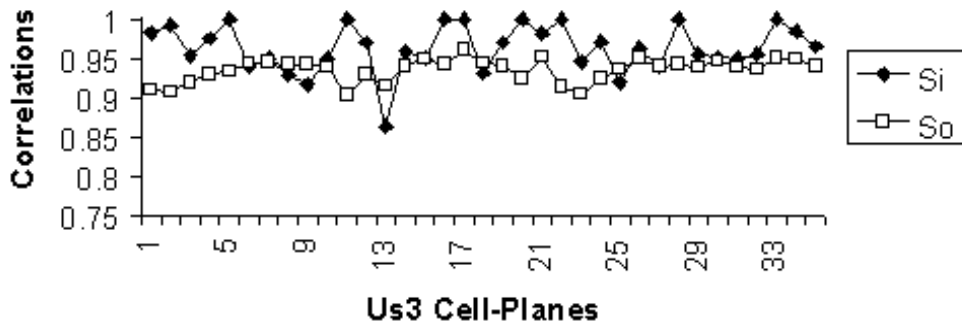


Fig. 12. Correlations amongst Fukushima's training patterns.

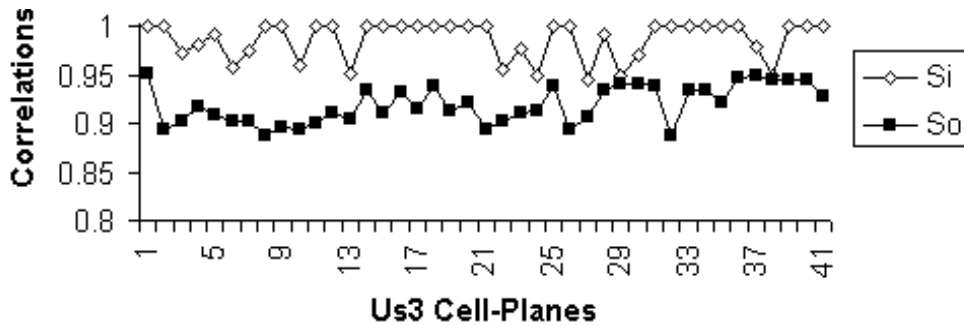


Fig. 13. Correlations amongst the training patterns in Fig. 11.

S_i and S_o of Fukushima's original training patterns are shown in Fig. 12, and those of our new training patterns are shown in Fig. 13.

Generally speaking, the correlations in Fig. 13 amongst the patterns of the same categories are larger than those in Fig. 12, but the correlations amongst the patterns belonging to different S-cell-planes are smaller than those in Fig. 12. This is a reason why we can get a better Neocognitron than the original one. From Table 1, we can also see that the performance of Neocognitron is sensitive to the receptive fields and selectivity.

7. Conclusion and Future Work

The Neocognitron is proved to be sensitive to three parameters, i.e., receptive fields, selectivity and number of planes. This paper discussed how to tune these parameters to improve the performance of the Neocognitron.

First of all, by analyzing the mechanism of Neocognitron's supervised learning in detail, this paper draws a conclusion that tuning the number of

planes of Neocognitron is absolutely equivalent to searching appropriate training patterns for supervised learning. And the correlation analysis can be used to evaluate the training patterns. On the basis of original training patterns, the methodology and application is given out to incorporate cooperative coevolution into the Neocognitron to search training patterns and tune parameters. Our experimental results prove that correlation amongst the training patterns is a critical factor to affect the performance of Neocognitron, which has been concluded could be improved by the proposed GA-based supervised learning.

Regarding the low recognition rate, we will search training patterns for U_{s2} in our future work, as well as add one more stage into the Neocognitron. The other problem is, GAs' running tends to be more and more difficult to find out better solution. Our future work will study how to maintain the diversity of the population at network level. The correlation amongst the training pattern will be applied to pre-selection with respect to the population at training pattern level.

References

1. K. Fukushima, *et al.* 1982, "Neocognitron: A new algorithm for pattern recognition tolerant of deformations and shifts in position," *Pattern Recognition* **15**(6), 455–469.
2. K. Fukushima and S. Miyake 1987, "Neural network model for selective attention in visual pattern recognition and associative recall," *Applied Optics* **26**(23), 4985–4992.
3. K. Fukushima 1988, "A neural network for visual pattern recognition," *IEEE Trans. On Computer*, March, 65–75.
4. K. Fukushima *et al.* 1993, "Recognition and segmentation of connected selective attention," *Neural Networks* **6**, 33–41.
5. K. Fukushima 1988, "Neocognitron: A hierarchical neural network capable of visual pattern recognition," *Neural Networks* **1**, 119–130.
6. K. Fukushima and N. Wake 1991, "Handwritten alphanumeric character recognition by the neocognitron," *IEEE Trans. On Neural Networks* **2**(3), 355–365.
7. K. Fukushima and N. Wake, "An improved learning algorithm for the neocognitron," in *Artificial Neural Network* eds. Aleksander and J. Taylor, **1**, 497–504, Amsterdam, North-Holland (1992).
8. Y. T. Cheng and D. S. Yeung 1999, "Sensitivity analysis of Neocognitron," *IEEE Trans. On Systems, Man, and Cybernetics* **29**(2), 238–249.
9. K. Fukushima, *et al.* 1994, "Neocognitron with dual C-cell Layer," *Neural Networks* **7**(1), 41–47.
10. H. Y. Chan and D. S. Yeung 1991, "Mapping multi-layer attributed graphs onto neocognitron network," *Int. Joint Conference on Neural Networks 1991*, Vol. 2, 1470–1475.
11. D. S. Yeung and H. Y. Chan 1994, "A hybrid cognitive system using production rules to synthesize neocognitrons," *Neural Systems* **5**(4), 345–355.
12. D. S. Yeung, *et al.* 1994, "Incorporating production rules with spatial information onto a neocognitron neural network," *Neural Systems* **5**(2), 131–142.
13. Anne Trotin, *et al.* 1991, "Acquired structure, adapted parameters: Modifications of the neocognitron," in *Artificial Neural Network, Proceedings of ICANN-91*, Vol. 2, 1107–1110, Amsterdam, Elsevier, June.
14. Nicholas Darbel, Anne Trotin and Francis Jutand 1991, "A neural net sized by data," in *Algorithms and Parallel VLSI Architectures*, eds. E. F. Deprettere, Vol. B, 203–212, Elsevier Science Publishers.
15. T. H. Hildebrandt 1991, "Optimal training of thresholded linear correlation classifiers," *IEEE Transactions on Neural Networks* **2**(6), 577–588.
16. D. Lovell, *et al.* 1993, Comments on "Optimal training of threshold linear correlation classifiers," *IEEE Trans. On Neural Networks* **4**(2), 367–369.
17. A. van Ooyen and B. Nienhuis 1993, "Pattern recognition in the neocognitron is improved by neuronal adaptation," *Biological Cybernetics* **70**, 47–53.
18. J. Pujol 1998, "Evolving the topology and the weights of neural networks using a dual representation," *Applied Intelligence* **8**, 73–84.
19. D. Dasgupta 1998, "Evolving neuro-controllers for a dynamic system using structured genetic algorithms," *Applied Intelligence* **8**, 113–121.
20. D. S. Yeung, *et al.* 1998, "Neocognitron based handwriting recognition system performance tuning using genetic algorithm," *Proc. of IEEE Int. Conf. on Systems, Man and Cybernetics*, San Diego, pp. 4228–4233.
21. J. Horn, D. E. Goldberg and K. Deb 1994, "Implicit niching in a learning classifier system: Nature's Way," *Evolutionary Computation* **2**(1), 37–66.
22. D. E. Moriarty and R. Miikkulainen 1996, "Efficient reinforcement learning through symbiotic evolution," *Machine Learning* **22**, 11–32.
23. J. Paredis 1995, "Coevolutionary computation," *Artificial Life* **2**, 355–375.
24. R. E. Smith, *et al.* 1993, "Searching for diverse, cooperative populations with genetic algorithms," *Evolutionary Computation* **1**(2), 127–149.
25. D. E. Moriarty and R. Miikkulainen 1998, "Forming neural networks through efficient and adaptive coevolution," *Evolutionary Computation* **5**(4), 373–399.
26. V. Maniezzo 1994, "Genetic evolution of the topology and weight distribution of neural networks," *IEEE Trans. On Neural Networks* **5**(1), 39–53.