Pattern Classification with Parallel Processing of the Cellular Neural Networks-Based Dynamic Programming

Hyongsuk Kim, Taewan Oh, Sangik Na, and Changbae Yoon

School of Electronics and Information Eng,
Chonbuk National Univ.,
Republic of Korea
hskim@moak.chonbuk.ac.kr,
{control,hunter209}@mail.chonbuk.ac.kr,
guboon@hanmail.net

Abstract. A Cellular Neural Networks (CNN)-based fast pattern classification algorithm utilizing the most likely path finding feature of the dynamic programming is proposed. Previous study shows that the dynamic programming for the most likely path finding algorithm can be implemented with CNN. If exemplars and test patterns are assigned as the goals and the start positions, respectively, on the CNN-based dynamic programming, the paths from test patterns to their closest exemplars are found with the optimality feature of the CNN-based dynamic programming. Such paths are utilized as aggregating keys for the classification. The algorithm is similar to the conventional neural network-based method in the use of the exemplar patterns but quite different in the use of the most likely path finding. Simulation results are included.

1 Introduction

One of the most important superiority human cognition over the conventional digital processing is the parallel processing in the image recognition. The pattern classification is an essential technique for the image recognition.

The pattern classification techniques ever developed can be classified roughly into the statistical probability-based, Fuzzy-based, and the neural network-based ones. The statistical probability-based technique [1][2][3] is the classical method where probabilities to belong to specific classes are computed utilizing the features of each class. However, choosing the adequate features to minimize the classification error is difficult. Also, the order of the function to compute the probability grows high due to the requirement of many features for better classification performance. In the Fuzzy-based technique [4][5][6], the patterns are classified according to fusion of Fuzzy rules. Analysis of exemplar patterns and creation of the Fuzzy rules for the classification are burdensome. The neural network-based technique [7][8] is the one which does not require either analysis of patterns or the effort of creating the Fuzzy rules. If some typical patterns called exemplars are presented for learning, the neural networks create classification rules automatically on the networks, which is the similar way to human learning. The classification is performed with learning of exemplar patterns without prior knowledge about them. The classification principle

X. Zhou et al. (Eds.): APPT 2003, LNCS 2834, pp. 265–273, 2003.

under this approach is the use of nearest pattern classification. The drawbacks are long time taking in learning and unsuccessful learning due to the local minima.

The proposed pattern classification algorithm is with the utilization of fast parallel processing of the Cellular Neural Networks (CNN) [9][10]. The algorithm is similar to the neural networks-based classification in the use of exemplars but quite different in the use of the most likely path finding feature. Also, successful classification is always guaranteed and learning is not required. The basic principle under the proposed algorithm is the closest exemplar finding with the use of optimal computation property of dynamic programming. Since the proposed algorithm utilizes the competition among the classes, the optimality for the classification is kept. Also, since the classification is performed simply with presenting the exemplar patterns, prior knowledge or analysis of classes is not required.

2 Principle of CNN

The CNN is a massively parallel computational structure invented by Chua et al [9][10] in 1988. The CNN is composed of 2-D analog arithmetic cells which are disposed at nodes of 2-D grid as shown in Fig. 1(a). At each cell, there are 3 different sets of information sources: one from outputs of its and its neighboring cells through weighted connections called A template, the other from its and its neighbor's inputs through another set of weighted connections called B template, and the last one from its bias value z. In the CNN, each cell performs same dynamic operation as denoted in (1).

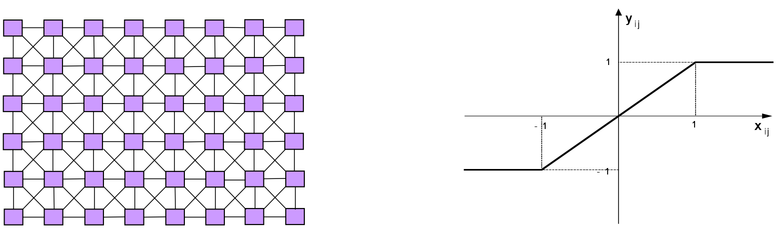
$$\frac{dx(i,j)}{dt} = -x(i,j) + \sum_{kl\in N_r} A(ij,kl)y_{kl} + \sum_{kl\in N_r} B(ij,kl)U(k,l) + z_{ij} \tag{1}$$

where *x(i,j)* is the state of the cell (i,j). *A(ij,kl)* and *B(ij,kl)* are the templates(weights) between cell (i,j) and cell (k,l). Also, Z_{ij} is the bias value of cell (i,j).

The output *y(i,j)* of the cell (i,j) is

$$y(i,j) = f(x(i,j)) \tag{2}$$

where *f* is a nonlinear output function defined as in Fig. 1(b).



(a) (b)

Fig. 1. Cellular Neural Networks(CNN) (a) cell disposition of the CNN (b) nonlinear output function of a CNN cell.

Since every CNN cell has same weighted connections and dynamic mechanism, every cell performs the same actions as long as input and sate value are the same. However, if different template sets are employed, different processing results are expected even though same input or initial state values are given. Many sets of templates for different type of image processing have already been developed [11].

3 Implementation of the DP with CNN Nonlinear Templates

The Dynamic Programming (DP) is an efficient optimal path finding algorithm. Let *D(i,j)* and *D(k,l)* be the shortest distance to the goal from a node (i,j) and a node (k,l), respectively. Then, *D(i,j)* is computed utilizing *D(k,l)* as

$$D(i,j) = \min\left\{\ d_{ij,kl} + D(k,l)\ ,\ (k,l) \in R(i,j)\right\} \tag{3}$$

where $d_{ij,kl}$ is the shortest distance between the node (i,j) and (k,l), and *R(i,j)* is the set of neighboring nodes of (i,j). Note that $d_{ij,kl}$ equals to zero if $ij=kl$. By setting the initial value of *D* at a goal (k,l) with zero and all other *D*s with a big value (bigger value than shortest distance between the most far nodes), *D(i,j)* in (3) computes the minimum distance to the goal (k,l) at a node (i,j). If an arithmetic cell is arranged to compute (3) at each node, the processing of (3) is confined to the local operation of *min* and *summation*. Since the *min* circuit is known to be more complicated than *max* circuits for the practical implementation [12], some arrangement of (3) enables *min* to be replaced by *max* operation.

Let's *y(i,j)* be the complement value of *D(i,j)* from a dummy constant I_{max} for the node at (i,j) as

$$y(i,j) = I_{\max} - D(i,j) \tag{4}$$

Plugging *D(i,j)* of (4) into (3) and rearranging its result, we have

$$y(i,j) = \max\ \left\{y(k,l) - d_{ij,kl}\ ;\ (k,l) \in R(i,j)\right\} \tag{5}$$

In the proposed algorithm, (5) is computed by a cell at each node. On the networks with these cells, I_{max} is set for the state of the goal node and zero is set for all other states. At the goal cell, *y(k,l)* and $d_{ij,kl}$ in the parenthesis of (5) are I_{max} and zero, respectively since (k,l) = (i,j). Therefore, the operation of (5) can be summarized as

$$y(i,j) = \begin{cases} I_{\max} & ;\ \textit{if}\ \ (i,j)\ \ \textit{is designated as the goal.} \\ \max\{\ y(k,l) - d_{ij,kl}\ ;\ (k,l) \in R(i,j)\ \}\ ; & \textit{otherwise.} \end{cases} \tag{6}$$

With *y(i,j)*, the shortest distance to the goal, *D(i,j)*, can easily be computed with (4).

The physical meaning of the modified dynamic programming of (5) is associated with the propagation of reference information from its goal while its magnitude is reduced by the amount of the distance weight value between cells.

The nonlinear function of the *max* and the linear function of subtraction are involved in operation (5). It is not easy to be directly implemented with the CNN linear template due to the nonlinearity of these operations. The proposed strategy to implement the functions (5) with CNN is associated with the decomposition into several simple functions. Arranging them to be processed by cells at different layer, the CNN to compute (5) is feasible. Adding y_{ij} to both side of (5), (5) becomes

$$2y_{ij} = \max\left\{y_{ij} + y_{kl} - d_{ij,kl}\ ,\ (k,l) \in R(i,j)\right\} \tag{7}$$

or

$$y_{ij} = \max\left\{\frac{y_{ij} + y_{kl} - d_{ij,kl}}{2}\ ,\ (k,l) \in R(i,j)\right\} \tag{8}$$

Let the *max* in (8) be processed by a cell on a layer called Distance Computation layer and the terms inside of the parenthesis be processed by a cell on the other layer called Intermediate layer. Then,

$$y_{DC}(i,j) = \max\left\{y_I(i,j)\right\} \tag{9}$$

and

$$y_I(i,j) = \frac{1}{2}(y_{DC}(i,j) + y_{DC}(k,l)) - \frac{d_{ij,kl}}{2} \tag{10}$$

where $y_I(ij)$ and $y_{DC}(ij)$ are the output of the cell (i,j) in the I layer and DC layer, respectively. The structure to find the most likely path through such processing is as in Figure 2. The cells of I layer are placed at the locations corresponding to the links between the nodes and the metric information is provided externally to its input. The cells in DC and PF layer have connections only with their adjacent cells in I layer.

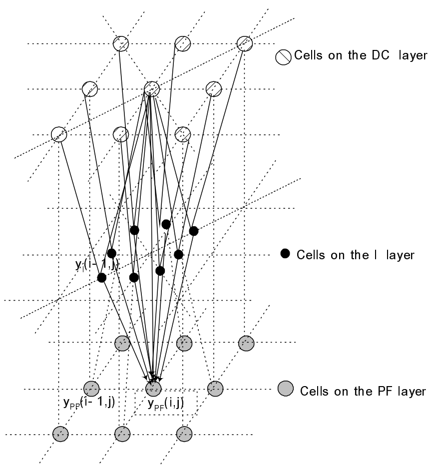


Fig. 2. Connections of a cell in the proposed multi-layer CNN structure.

Thus, the *max* computation in (9) is performed with the interaction between the cells in DC layer and I layer. Path decision criterion can also be more simplified utilizing the cells in the I layer as

$$y_I(i, j) = y_{DC}(i, j) \tag{11}$$

The average operation and the subtraction in (10) can easily be performed with the linear template [11]. The nonlinear operation of *max* in (9) and comparison in (11) can be implemented with nonlinear templates as in [13].

4 Patten Classification Utilizing the Optimization Property of the Dynamic Programming

The proposed pattern classification algorithm is the utilization of both benefits of parallel processing property of the CNN and the optimization property of the dynamic programming. The CNN determines appropriate group for each test pattern based on the most likely path finding mechanism where the exemplars and the test patterns are treated as the goals and the start points, respectively.

Let j th pattern in the ith class be denoted as $P_i(j)$ and P_s be the whole set of the patterns. That is

$$p_i(j) \in \mathrm{P_s}, \qquad 1 \le i \le M, 1 \le j \le N_m \tag{12}$$

where M is the number of classes and N_m is the number of exemplar patterns in each class. Also, let the distance between the test pattern P_t and an exemplar $P_i(j)$ (j th pattern in the i th class) be the distance $d_{t\text{-}ij}$. If the test pattern has the closest distance to the lth pattern in the kth class such as

$$d_{t-kl} = \min\{d_{t-ij}, \quad 1 \le i \le M, 1 \le j \le N_m\} \tag{13}$$

then, k is the class to which the test pattern has the shortest path among M classes. In this case, the pattern P_t is classified into the class k.

The algorithm to implement above concept with CNN is shown in Figure 3. For this algorithm, the location of each pattern is expressed with the cell position in the multilayer CNN. The first step of the algorithm is the distance weight setting on the I layer of the CNN. The information is utilized for computing shortest distance to the nearest exemplar. The next step is the exemplar and test pattern setting on the DC and PF layer, respectively. If the identical reference values I_{max}s are given at the exemplar positions as goals, the reference values propagate to all directions while they are reduced by the amount of the inter cell distance value provided by I layer. When the reference value reaches each test pattern, the path connection from each test pattern to the closest exemplar begins to be made. The path information obtained after this procedure is bi-levels with zero and infinitive big value. Providing such paths as distance information on I layer again and presenting class identification (ID) values at the exemplar cell positions, the identification values are propagated without alteration through the paths with zero distance weighting. When the class identification values reach the positions of the test patterns, the classification is performed by simply reading the class identification values which appear at the cells corresponding to test patterns.

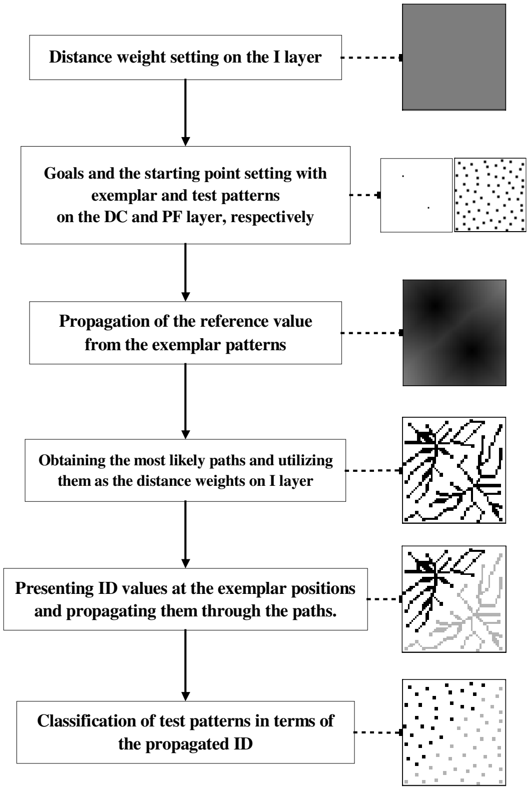


Fig. 3. Flow diagram of the proposed algorithm.

5 Simulation

Pattern classification capability of the proposed algorithm has been shown about patterns with multiple classes. Also, its classification capability for data with highly nonlinear class boundary has been demonstrated.

5.1 Classification of Patterns with Multiple Classes

Figure 4 shows the data with multiple classes, where figures 4(a) shows exemplar patterns of 4 classes and Figure 4(b) is test patterns. It is assumed that the inter cell distances are all identical. Being provided four identical reference inputs at the exemplar positions, the reference values propagate through the CNN networks while they are reduced by the amount of inter-cell distance as in Figure 4(c). If the reduced reference values reach the locations of the test patterns, the paths initiated at the locations of test patterns proceeds along the direction which the strongest reference value comes from. The most likely paths obtained with such procedure are shown in Figure 4(d). These paths are provided again on the I layer as the inter-cell distance of zero and 1 for path and non-path, respectively. With such bi-leveled inter-cell distance, the identification values presented at exemplar positions are propagated through the paths. Figure 4(e) shows the propagated identification values on the paths. Note that the distance weights on weights are zero. The classification is performed simply with reading the propagated class identification value at each pattern position. Figure 4(f) is the classified results with this procedure. It can be seen that the test patterns are all reasonably classified in terms of the Euclidian distance to the closest classes of exemplars.

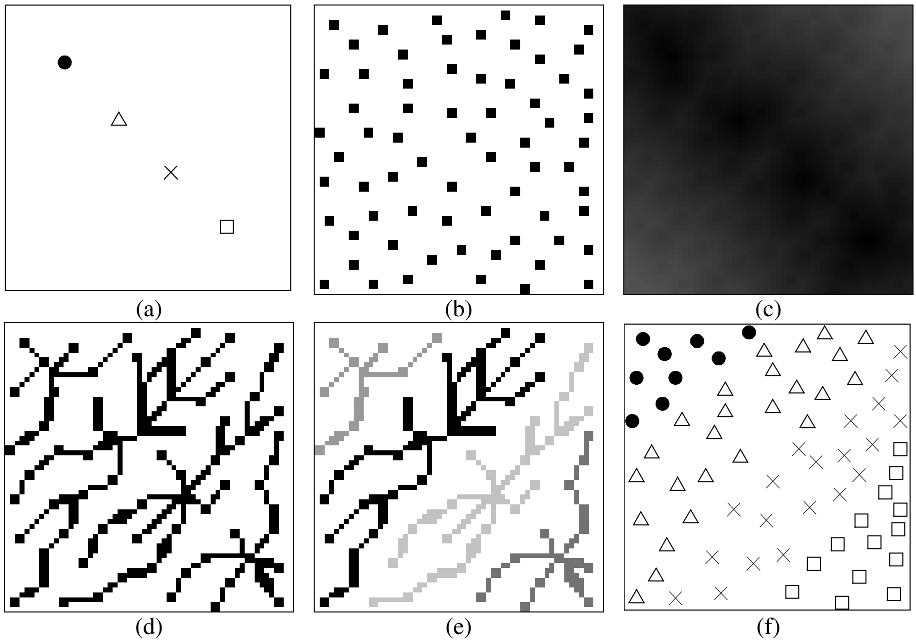


Fig. 4. Classification of patterns of multiple classes (a) exemplar patterns (b) test pattern (c) propagated reference values (d) paths from the test patterns to the closest exemplars (e) class identification values propagated from the exemplar patterns through the paths in (d) (f) classified patterns.

5.2 Classification of Data with Nonlinear Class Boundaries

The pattern classification about data with the nonlinear class boundary is a difficult task in classification. Figure 5 is the data with very highly nonlinear class boundaries where Figure 5(a) is the exemplar patterns and Figure 5(b) is test patterns. Also, the Figures 5(c)-5(e) shows the propagated reference values, the paths to the nearest exemplar, and the propagated identification values, respectively. The classified result is as shown in Figure 5(f). No extra difficulty is experienced in classifying patterns with highly nonlinear class boundaries.

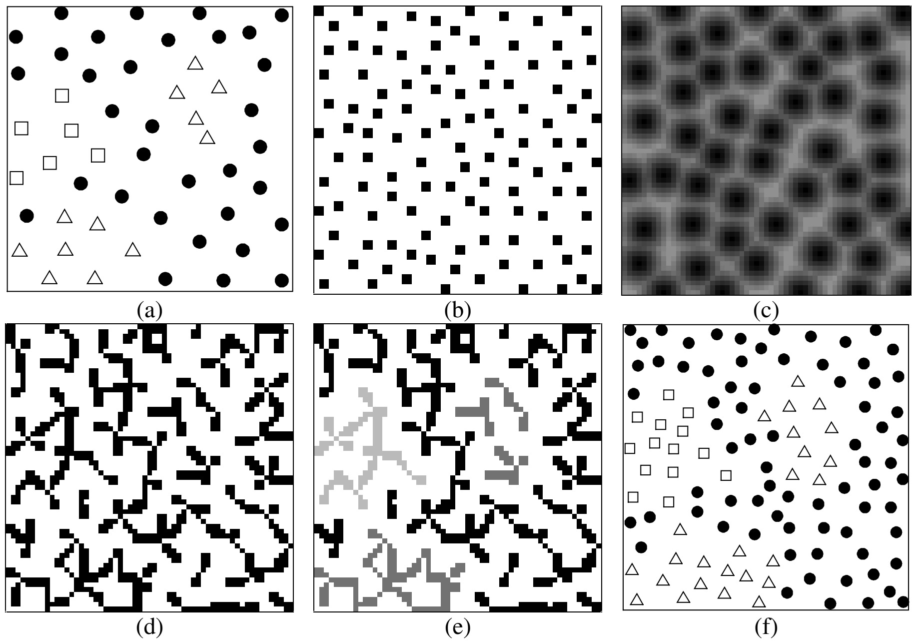


Fig. 5. Classification of the patterns with highly nonlinear class boundaries (a) exemplar patterns (b) test pattern (c) propagated reference values (d) paths from the test patterns to the closest exemplars (e) class identification values propagated from the exemplar patterns through the paths in (d) (f) classified patterns.

6 Conclusion

A pattern classification algorithm using the analog parallel processing of the CNN technology has been proposed. The algorithm utilizes the both benefits of parallel processing of the CNN and the optimization of the dynamic programming. Another feature of the proposed classification algorithm is the use of exemplars as in the conventional neural networks.

In this classification algorithm, the exemplars and test patterns are considered as goals and starting points in the most likely path finding algorithm, respectively. The classification is composed of two steps of information propagation; the distance reference signal propagation and the class identification value propagation. Since principle of the proposed algorithm is the use of competition among the class, the optimality for the classification is kept. Also, prior knowledge or analysis of classes is not necessary in this algorithm.

Simulations have been done to test the properties of the proposed algorithm. With the algorithm, the classification has been performed successfully for all the problems in the simulations. Perhaps the strongest feature of the proposed algorithm is the classification capability about the data with highly nonlinear class boundaries. No extra difficulty is encountered in classifying patterns with highly nonlinear class boundaries.

References

1. Patra, P.K. Nayak, M. Nayak, S.K. Gobbak, N.K.: Probabilistic Neural Network for Pattern Classification. Neural Networks, 2002. IJCNN '02. Proceedings of the 2002 International Joint Conference on , vol. 2, pp.1200–1205, 2002.
2. Lee Luan Ling, Cavalcanti, H.M.: Fast and efficient feature extraction based on Bayesian decision boundaries. Pattern Recognition, 2000. Proceedings. 15th International Conference on , vol. 2, pp. 390–393, 2000.
3. Horiuchi, T.: Decision rule for pattern classification by integrating interval feature values. Pattern Analysis and Machine Intelligence, IEEE Transactions on , vol. 20, pp. 440–448, 1998, Apr.
4. Hahn-Ming Lee, Chih-Ming Chen, Jyh-Ming Chen, Yu-Lu Jou.: An efficient fuzzy classifier with feature selection based on fuzzy entropy, Systems, Man and Cybernetics, Part B, IEEE Transactions on , vol. 31, pp. 426–432, 2001, June.
5. Ishibuchi, H. Nakashima, T. Murata, T.: Performance evaluation of fuzzy classifier systems for multidimensional pattern classification problems. Systems, Man and Cybernetics, Part B, IEEE Transactions on , vol. 29, pp. 601−618, 1999, Oct.
6. Xi Chen, Dongming Jin, Zhijian Li.: Fuzzy petri nets for rule-based pattern classification, Communications, Circuits and Systems and West Sino Expositions, IEEE 2002 International Conference on , vol. 2 , pp. 1218–1222, 2002.
7. Jinwook Go, Gunhee Han, Hagbae Kim, Chulhee Lee.: Multigradient: A New Neural Network Learning Algorithm for Pattern Classification, Geoscience and Remote Sensing, IEEE Transactions on , vol. 39, pp. 986–993, 2001, May.
8. Murphey, Y. L. Yun Luo,: Feature Extraction for a Multiple Pattern Classification Neural Network System. Pattern Recognition, 2002. Proceedings. 16th International Conference on , vol. 2, pp. 220–223, 2002.
9. Chua, L.O. and Yang, L.: Cellular Neural Networks: theory. IEEE Tr. on Circuits Systems, vol.35, pp.1257–1272, 1988.
10. Chua, L.O. and Yang, L.: Cellular Neural Networks: applications. IEEE Tr. on Circuits Systems, vol. 35, pp. 1273–1290, 1988.
11. Analogical and Neural Computing Lab, SimCNN-Multi-layer CNN Simulator for Visual Mouse Platform: Reference Manual version 2.2. Computer and Automation Institute(MTA SzTAKI) of the Hungarian Academy of Science, 1998.
12. Espejo, S., Dominguez-Castro, R., Linan, G.., Rodriguez-Vazquez, A.: A 64×64 CNN universal chip with analog and digital I/O. Electronics, Circuits and Systems, 1998 IEEE International Conference on , vol. 1, pp. 203–206, 1998.
13. Kim, H., Son, H., Roska, T., Chua, L.O.: Optimal path finding with space- and time-variant metric weights with Multi-layer CNN. International Journal of Circuit Theory and Applications, vol. 30, pp. 247–270, 2002, 2.
14. Roska, T. and Chua, L.O.: The CNN universal machine: an analogic array computer. IEEE Tr. on Circuits Systems II, CAS–40, pp. 163–173, 1993.
15. Bellman, R.: Dynamic Programming, Princeton, NJ: Princeton Univ. Press, 1957.