Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

Associative Memories with Multi-Valued Cellular Neural Networks and Their Application to Disease Diagnosis

Takuma Akiduki, Zhang Zhong, Takashi Imamura, Tetsuo Miyake
Department of Production Systems Engineering,
Toyohashi University of Technology,
Aichi, Japan
{akiduki, zhang, ima, miyake}@is.pse.tut.ac.jp

***Abstract*—Cellular neural networks (CNNs) are one type of interconnected neural network and differ from the well-known Hopfield model in that each cell has a piecewise linear output characteristic. In this paper, we present a multi-valued CNN model in which each nonlinear element consists of a multi-valued output function. The function is defined by a linear combination of piecewise linear functions. We conduct computer experiments of auto-associative recall to verify our multi-valued CNN's ability as an associative memory. In addition, we also apply our multi-valued CNN to a disease diagnosis problem. The results obtained show that the multi-valued CNN improves classification accuracy by selecting the output level q properly. Moreover, these results also show that the multi-valued associative memory can expand both the flexibility of designing the memory pattern and its applicability.**

***Index Terms*—Multi-valued cellular neural networks, Associative memory, Diagnosis of disease**

I. Introduction

Cellular neural networks (CNNs) were proposed by Chua and Yang in 1988 [1]. They are one type of interconnected neural network and consist of nonlinear elements which are called *cells*. Each cell is coupled to its neighborhood cells and itself. The state of each cell changes in parallel based on a differential equation, and converges to an equilibrium state. CNNs differ from the well-known Hopfield model in that each cell has self-feedback and a piecewise linear output characteristic. These features provide the networks with extremely robust performance when used as an associative memory [2]. For this reason, associative memories using CNNs have been applied in various fields, such as in character and object recognition, medical diagnosis and machine fault diagnosis [3], [4], [5].

Nowadays researchers are interested in the application of associative memories to diagnosis systems. In this system, CNN as an associative memory limits its diagnosis accuracy, and two-level $\{-1, 1\}$ or three-level $\{-1, 0, 1\}$ is used as the conventional output-level. Kanagawa *et al.* applied the CNN to liver disease diagnosis[6]. In this case, the CNN's memory patterns were constructed from actual blood test results into three-level output. For example, an inspection item γ-GTP is usually categorized into the following three states: *normal* $(0-50\mathrm{IU}/l)$, *light excess* $(50-100\mathrm{IU}/l)$ or *heavy excess* (more than $100\mathrm{IU}/l$). However, the resolution of the inspection level does not seem to divide its range of values well enough. In order to increase the detection accuracy and memory capacity of CNNs, and to improve the computational efficiency in these diagnosis systems with associative memories, it is important to have a wide variety of diagnosis patterns. As one solution, CNNs with multi-valued output functions have been proposed in the past [7]. However, a synthesis design method with arbitrary output-level for associative storage mediums has not yet been proposed, and its validation has not yet been conducted.

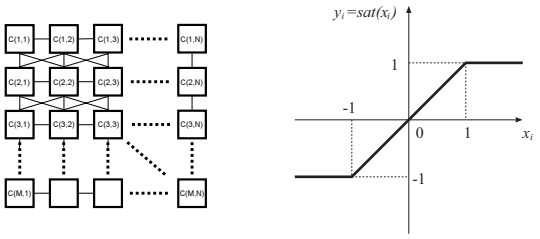
In this paper, we discussed a design method for a CNN with a multi-valued saturation function, which is called a *multi-valued CNN*. The basic idea of our multi-valued CNNs using a piecewise linear function with a saturation range and a sloped linear range was shown in past research [8], and the methodology for deciding their parameters was presented. From these results, we investigate an error-correcting capability with our multi-valued CNN as an associative memory. In addition, to verify the applicability of our multi-valued CNN, we apply it to disease diagnosis.

In Section II we describe a design method of the multi-valued output function which is defined by linear combinations of piecewise linear functions. Then we construct an associative memory with our multi-valued CNN that consists of cells with multi-valued output functions. In Section III we conduct computer experiments to verify our multi-valued CNN's error correcting capability as an associative memory. In Section IV we show a method for applying our multi-valued CNN to a pattern classification problem. The problem as a case study is classification of liver disease from blood test data. Finally, the study in this paper is summarized in Section V.

II. Multi-Valued CNNs

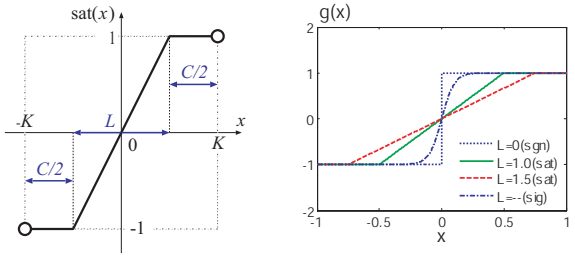
The cellular neural network (CNN) is an artificial neural network which has a predetermined local interconnection structure. The structure of an associative memory CNN is designed by the eigenstructure method shown in [3], [9]. In the following section, we describe a design method of a multi-valued CNN as an associative memory.

978-1-4244-2794-9/09/$25.00 ©2009 IEEE



(a) Network topology (b) Piecewise linear function

Fig. 1. A two-dimensional cellular neural network and its output function



(a) Base waveform (b) Output characteristics

Fig. 2. Proposed base waveform and its parameters. (a) defines the base function of our multi-valued output function. (b) shows output characteristics in (3). If $L = 0$, then the base waveform works as the sign function. If $0 < L < 2Kr_0$, then the base waveform works as a piecewise linear function.

A. Dynamics of a Cell

We first consider a two-dimensional CNN, which is composed of $M \times N$ nonlinear elements that are called *cell*s. The cell $\mathrm{C}(i)$ denotes the elements of the i^{th} cell, where $i = (v-1)N + u$, $v = 1, \ldots, M, u = 1, \ldots, N$, and the number of cells is $n = M \times N$ (refer to Fig. 1 (a)). The dynamics of the i^{th} cell is expressed as follows:

$$\dot{x_i}(t) = -x_i(t) + \sum_{j \in \mathcal{N}_i(r)}^{n} T_{ij} \cdot g(x_j(t)) + I_i, \qquad (1)$$

where x_i and I_i represent, respectively, the state variable and bias value. T_{ij} are coupling coefficients between the i^{th} and j^{th} cells. $\mathcal{N}_r(i)$ represents a set of r-neighborhood cells corresponding to the i^{th} cell, and the integer r is positive. In addition, $g(\cdot)$ represents the nonlinear output function of each cell. The conventional output function of CNNs is expressed as in the following equation:

$$\mathrm{sat}(x_i) = \frac{1}{2}\left(|x_i + 1| - |x_i - 1|\right). \qquad (2)$$

This output function, which is piecewise linear, has two saturated levels (refer to Fig. 1(b)). Thus, the output value of each cell is binary ± 1 for $|x_i| > 1$.

B. Definition of the multi-valued output function

Secondly, we give a design procedure for constructing a multi-valued output function as expressed by the conventional output function in (2).

To extend the output function in (2) to our multi-valued output function, we introduce some notation in Fig. 2 (a). The binary output function in (2) consists of two saturation ranges and one linear slope range. Let L and C be the length of the slope range and the length of the saturation range, respectively. We give a saturation function that consists of both of the above-described parameters C and L as expressed in equation (2) in the following equation:

$$g_0(x_i) = \frac{1}{L}\left(\left|x_i + \frac{L}{2}r_s\right| - \left|x_i - \frac{L}{2}r_s\right|\right), \qquad (3)$$

where $r_s = 1/(q-1)$ and q is the output-level. In addition, we assume equilibrium points of each cell by parameter K. Thus, the values of both parameters C and L are defined by the following equation:

$$K = \frac{1}{2}(C + L), \quad C, L > 0, \qquad (4)$$

and the ratio of parameter L to $C + L$ is represented by $r_0 = L/(C + L)$. In addition, the relation of these parameters is illustrated in Fig. 2. Since a base waveform in (3) has two saturation ranges, the saturation ranges with q-level are generated by linear combinations of $q-1$ *base waveform* with both shrinking and shifting the base waveforms. Therefore, a q-valued output function is defined as follows.

$$g_q(x_i) = g_0(x_i)\delta + \sum_{k=1}^{P}\{g_0(x_i - A_k) + g_0(x_i + A_k)\}, \qquad (5)$$

where $P = (q - 2 + q \bmod 2)/2$, and coefficients (A_k, δ) in (5) are represented as follows:

$$(A_k, \delta) = \begin{cases} ((2k-1)Kr_s, 0) & q \text{ is odd} \\ (2kKr_s, 1) & q \text{ is even.} \end{cases}$$

In addition, a set of output values of a cell with a q-valued output function is defined as follows:

$$\mathcal{M}_q \triangleq \{2ur_s - 1 \mid 0 \le u \le q-1, u \in \mathcal{Z}\}, \qquad (6)$$

where $\mathcal{Z}$ is a set of integer. For example, when the output-level is $q = 4$, the set of output values is $\mathcal{M}_4 = \{-1, -1/3, 1/3, 1\}$ from the definition in (6). Here, Fig. 3 shows examples of the output waveforms which are calculated in (5). The parameters in Fig. 3 are $q = 2, 3, 4, 5$, and $r_0 = 0.5$, respectively.

C. Storage of memory patterns

Next, we express the differential equation of (1) in vector notation, so the two-dimensional CNNs which have M rows and N columns are represented as follows:

$$\dot{\boldsymbol{x}}(t) = -\boldsymbol{x}(t) + \boldsymbol{T}\boldsymbol{y} + \boldsymbol{I}, \quad \boldsymbol{y} = g_q(\boldsymbol{x}(t)), \qquad (7)$$

where $\boldsymbol{x} = [x_i]$, $\boldsymbol{y} = [y_i] = [g_q(x_i)]$, and $\boldsymbol{I} = [I_i] \in R^n$ represent respectively a state vector, an output vector, and a

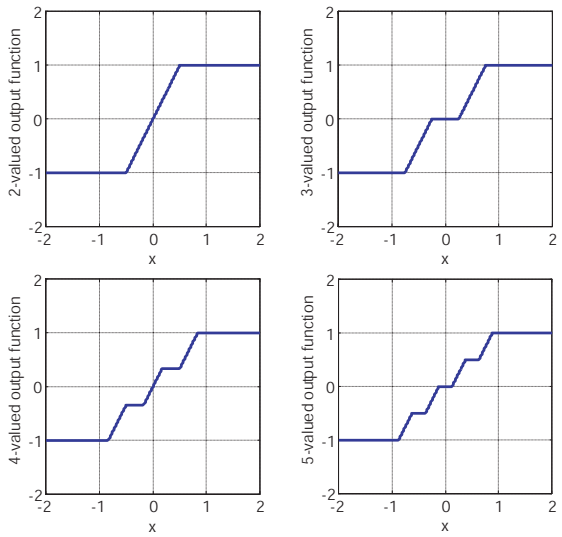


Fig. 3. Examples of multi-valued output functions in (5). Note that these parameters in (5) are $q = 2, 3, 4, 5$, $K = 1.0$, and $r_0 = 0.5$.

bias vector. Here n is the number of cells, and $n = M \times N$. The matrix $\boldsymbol{T} = [T_{ij}] \in R^{n\times n}$ is the template matrix composed of row vectors whose elements are zero when the corresponding cells have no connections. Here, R^n denotes real n-space, $\boldsymbol{x} \in R^n$ the n-dimensional column vector, and $\boldsymbol{x}'$ the transpose of $\boldsymbol{x}$, respectively. The network in (7) has a number of asymptotically stable equilibrium points, since the network in (7) have all their n eigenvalues at -1.

To design a CNN as an associative memory, we compute the coefficient matrix $\boldsymbol{T}$ and the bias vector $\boldsymbol{I}$ in (7) following Liu and Michel's design procedure shown in [3], [9]. Firstly, the matrix $\boldsymbol{Q} = [Q_{ij}] \in R^{n\times n}$ is set as follows:

$$Q_{ij} = \begin{cases} 1, & \text{if } \mathrm{C}(i) \in \mathcal{N}_r(i) \\ 0, & \text{otherwise} \end{cases},$$

and let the matrix $\boldsymbol{S}$ be $\boldsymbol{S} = \boldsymbol{Q} = [S_{ij}] \in R^{n\times n}$. Secondly, let α_i be a saturation value of the output function in (5). In the case of a q-valued output function in (5), the saturation value is $\alpha_i \in \mathcal{M}_q$, and $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)' \in \mathcal{M}_q^n$ is an n-dimensional q-valued vector. We call this vector the *memory vector* of the network in (7). Given that the m desired memory vectors are $\boldsymbol{\alpha}^1, \ldots, \boldsymbol{\alpha}^m$, let us assume the state vectors $\boldsymbol{\beta}^k$ corresponding to $\boldsymbol{\alpha}^k$ are such that:

$$\boldsymbol{\beta}^k = K\boldsymbol{\alpha}^k, \quad k = 1, \ldots, m \tag{8}$$

where the real number K is an equilibrium point assignment coefficient, and m is the number of memory patterns. When systems in (7) are in the equilibrium state $\lim_{t\to\infty} \dot{\boldsymbol{x}}(t) = 0$, the state vector $\boldsymbol{x}(\infty)$ is expressed by $\boldsymbol{x}^e = \boldsymbol{\beta}^k$, and so the output vector is $\boldsymbol{y}^e = g_q(\boldsymbol{x}^e) = \boldsymbol{\alpha}^k$. In order to store the desired memory vectors $\boldsymbol{\alpha}^k$, $k = 1, \ldots, m$ in (7), the matrix $\boldsymbol{T}$ and vector $\boldsymbol{I}$ are designed to satisfy the following equations:

$$\begin{cases} -\boldsymbol{\beta}^1 + T\boldsymbol{\alpha}^1 + \boldsymbol{I} = 0 \\ -\boldsymbol{\beta}^2 + T\boldsymbol{\alpha}^2 + \boldsymbol{I} = 0 \\ \quad \vdots \\ -\boldsymbol{\beta}^m + T\boldsymbol{\alpha}^m + \boldsymbol{I} = 0 \end{cases}. \tag{9}$$

Here we set the following matrices:

$$\boldsymbol{Y} = (\boldsymbol{\alpha}^1 - \boldsymbol{\alpha}^m, \boldsymbol{\alpha}^2 - \boldsymbol{\alpha}^m, \ldots, \boldsymbol{\alpha}^{m-1} - \boldsymbol{\alpha}^m),$$
$$\boldsymbol{Z} = (\boldsymbol{\beta}^1 - \boldsymbol{\beta}^m, \boldsymbol{\beta}^2 - \boldsymbol{\beta}^m, \ldots, \boldsymbol{\beta}^{m-1} - \boldsymbol{\beta}^m).$$

After this, the above equations in (9) are transformed as follows:

$$\boldsymbol{Z} = \boldsymbol{T}\boldsymbol{Y}, \tag{10}$$
$$\boldsymbol{I} = \boldsymbol{\beta}^m - \boldsymbol{T}\boldsymbol{\alpha}^m. \tag{11}$$

Here, the matrix $\boldsymbol{Y}, \boldsymbol{Z} \in R^{n\times(m-1)}$ is a known quantity. So matrix $\boldsymbol{T} \in R^{n\times n}$ can be solved from matrix $\boldsymbol{Y}, \boldsymbol{Z}$. Because the matrix $\boldsymbol{Y}$ is a non-square matrix, using singular value decomposition, we decompose the matrix $\boldsymbol{Y}$ as follows:

$$\boldsymbol{Y} = \begin{bmatrix} \boldsymbol{U}_1 & \boldsymbol{U}_2 \end{bmatrix} \begin{bmatrix} \boldsymbol{D} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{V}_1' \\ \boldsymbol{V}_2' \end{bmatrix},$$

where $\boldsymbol{U}_1\boldsymbol{U}_1' = \boldsymbol{V}_1'\boldsymbol{V}_1 = \boldsymbol{E}$ is a unit matrix, and the diagonal matrix $\boldsymbol{D} = \mathrm{diag}(\lambda_1^{1/2}, \ldots, \lambda_{(m-1)}^{1/2})$. Given that $\boldsymbol{z}^i$ is the i^{th} row vector of matrix $\boldsymbol{Z}$, equation (10) gives $\boldsymbol{z}^i = \boldsymbol{t}^i U_1 D V_1'$ and we obtain

$$\boldsymbol{t}^i = \boldsymbol{z}^i \boldsymbol{V}_1 \boldsymbol{D}^{-1} \boldsymbol{U}_1',$$

where vector $\boldsymbol{t}^i = (t_1^i, \ldots, t_n^i)$ is the i^{th} row vector of matrix $\boldsymbol{T}$. Then for $j = 1, \ldots, n$, coupling the coefficient between the i^{th} and j^{th} cells, we get the following:

$$T_{ij} = \begin{cases} t_j^i, & \text{if } S_{ij} = 1 \\ 0, & \text{if } S_{ij} = 0 \end{cases}.$$

From the equations above we obtain both $\boldsymbol{T}$ and $\boldsymbol{I}$ in (7).

III. Experiments

In this section, computer experiments of auto-associative recall are conducted to verify the error-correcting capability of our q-valued CNN. This capability is a basic ability of auto-associative memories, which is to retrieve a memory pattern from a noisy vector.

The experimental procedure of the error-correcting capability of our q-valued CNN is as follows: Firstly, we prepare an initial pattern in (7) from the following equation:

$$\boldsymbol{x}(0) = K\tilde{\boldsymbol{y}}, \quad \tilde{\boldsymbol{y}} \equiv \boldsymbol{\alpha}^k + \boldsymbol{\varepsilon}, \ k \in \{1, \ldots, m\}, \tag{12}$$

where $\boldsymbol{\alpha}^k \in \mathcal{M}_q^k$ is a memory pattern, and $\boldsymbol{\varepsilon} = [\varepsilon_i] \in R^n$ is a noise vector which corresponds to the normal distribution: $\varepsilon_i \sim N(0, \sigma_e^2)$ for each $i = 1, \ldots, n$. However, it is assumed that the equilibrium point assignment coefficient is constant with value $K = 1.0$ for the sake of simplicity in this paper. The generated pattern from (12) is used as the initial value of the CNN in (7), and then we compute $\boldsymbol{x}(t)$ in (7) by

using a numeric calculation method (e.g. the Runge-Kutta method). Finally, we evaluate the similarity between the initial pattern $\boldsymbol{x}(0)$ and the recalled pattern that is represented by $\boldsymbol{y}_e = \lim_{t\to\infty} g_q(\boldsymbol{x})$, and $g_q(\cdot)$ is a q-valued output function in (5). Here, the similarity between vector $\boldsymbol{a}$ and $\boldsymbol{b}$ is calculated as in the following equation:

$$s_{\mathrm{in/out}}(\boldsymbol{a}, \boldsymbol{b}) = \frac{\langle \boldsymbol{a}, \boldsymbol{b} \rangle}{\|\boldsymbol{a}\| \cdot \|\boldsymbol{b}\|}, \tag{13}$$

where $\langle , \rangle$ denotes the dot product on R^n. If $s_{\mathrm{in/out}} = 1.0$, then both vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ are equal, and if $s_{\mathrm{in/out}} = 0.0$, then both vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ are mutually orthogonal. In addition, s_{in} represents the similarity between the memory pattern $\boldsymbol{\alpha}^k$ and the initial pattern $\boldsymbol{x}(0)$, and s_{out} represents the similarity between the memory pattern $\boldsymbol{\alpha}^k$ and the recalled pattern $\boldsymbol{y}_{\mathrm{e}}$ that corresponds to $\boldsymbol{x}(0)$.

In this experiment, we assume $\langle \boldsymbol{\varepsilon}, \boldsymbol{\varepsilon} \rangle = 0$, $\langle \boldsymbol{\alpha}^k, \boldsymbol{\varepsilon} \rangle = 0$ from both (12) and (13). Next, the standard deviation σ_{e} in (12) is decided by the following equation:

$$\sigma_e = \left\{ \frac{1}{n} \left(\frac{1}{s_{\mathrm{in}}^2} - 1 \right) \|\boldsymbol{\alpha}^k\| \right\}^{1/2}, \quad k \in \{1, \ldots, m\}. \tag{14}$$

In the following experiments, when we prepare an initial pattern in (12), we set the noise vector $\boldsymbol{\varepsilon}$ corresponding to the desired similarity s_{in} from both (13) and (14).

A. Experimental Conditions

The conditions of this experiment are as follows: the network size is $n = 100$, the neighborhood size is $r = n - 1$, and the number of memory patterns is $m = 0.3n = 30$, The range of output levels is $2 \le q \le 20$ and the ratio of parameter L to $C + L$ is $r_0 = 0.95$ (Note that $K = 1.0$, $L = 1.9$, and $C = 0.1$ in these conditions from (4)). In addition, we set the range of s_{in} in (13) to $0.05 \le s_{\mathrm{in}} \le 1.0$ with 0.05 step size.

B. Results and Discussion

An evaluation experiment of our q-valued CNN as an associative memory was performed. Fig. 4 shows the result of the error correcting capability with our q-valued CNN. Since the noisy vector in each s_{in} and each output level q are improved for s_{out}, the validity of our q-valued CNN as an auto-associative memory was confirmed from this result. However, if $s_{\mathrm{out}} = 1.0$ in Fig. 4, then both the recalled pattern $\boldsymbol{y}_e$ and the desired memory pattern $\boldsymbol{\alpha}^k$ are an *exact match*. Thus, when output level q is increasing, an exact match between $\boldsymbol{y}_e$ and $\boldsymbol{\alpha}^k$ is difficult.

IV. Applications

In this section, we consider a method for applying the q-valued CNN to pattern classification problems. These problems are aimed at diagnosis problems in the fields of process plant and medical care. In these fields, however, trouble is very rare in practice. For this reason it is difficult to obtain anomalous value data that occurs when there is some trouble such as hardware failure or an indication of disease. So, many engineers and doctors diagnose failures and diseases with

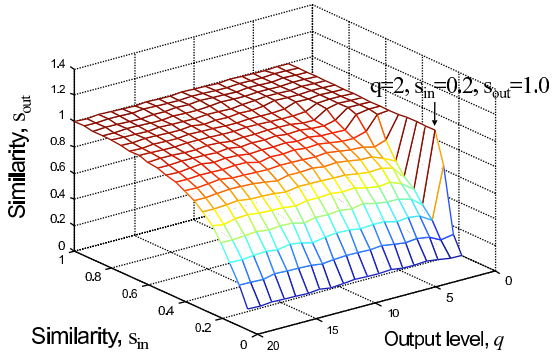


Fig. 4. Error correcting capability with our q-valued CNN as an auto-associative memory ($n = 100, q = 2-20, r_0 = 0.95$).

qualitative knowledge of the observed value of the trend of variation. For example, γ-GTP, which is an inspection item in blood tests, is usually categorized into the following three states: *normal* ($0 - 50\mathrm{IU}/l$), *light excess* ($50 - 100\mathrm{IU}/l$) or *heavy excess* (more than $100\mathrm{IU}/l$). These states can be represented by discrete values such as $\{-1, 0, 1\}$. In this regard, an associative memory CNN is suitable for representing this qualitative knowledge.

In the following section, we firstly introduce a classification method by using a q-valued CNN that is called *associative classification*. Then, the associative classification is applied to a diagnosis of liver disease as a typical example in the medical field.

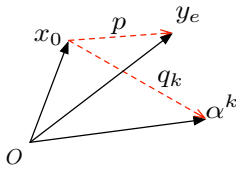
A. The method

The procedure of the associative classification is as follows:

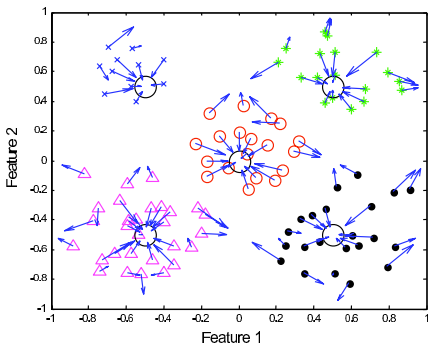
1) Compute the centroids of each cluster in the feature space from the training data, and then discretize these centroid patterns corresponding to the output level q. These created patterns $\boldsymbol{\alpha}^k, k = 1, \ldots, m$ are called *memory pattern*s, where m is the number of class labels. In other words, each centroid pattern serves as a memory pattern of our q-valued CNN.
2) Store these memory patterns in (7) by using Liu and Michel's design procedure in Section II-C.
3) Set the input vector in the test data to be the initial value $\boldsymbol{x}_0$ of the q-valued CNN in (7). Then the recalled pattern $\boldsymbol{y}_e$ is obtained by solving its q-valued CNN by using numeric calculation such as the Runge-Kutta method .
4) Discriminate class label c from the following similarity measurement between recalled pattern $\boldsymbol{y}_e$ and memory patterns $\boldsymbol{\alpha}^k$. First, we calculate both vectors $\boldsymbol{p}$ and $\boldsymbol{q}_k$ by the following equations.

$$\boldsymbol{p} = \boldsymbol{y}_e - \boldsymbol{x}_0, \quad \boldsymbol{q}_k = \boldsymbol{\alpha}^k - \boldsymbol{x}_0.$$

Fig. 5 (a) illustrates the relationship between vectors $\boldsymbol{p}$ and $\boldsymbol{q}_k$. Next, we also calculate a measure of distance d_k



(a) discriminant vector



(b) auto-associative classification

Fig. 5. An image of auto-associative classification. (a) shows the discriminant criterion for class label. (b) shows the process of auto-associative classification in 2-dimensional space. Note that each non-filled black circle represents the position of class centroids, and the other plotted dots represent the positions of each input vector in this data set. Moreover each arrow in the above figure represents both the convergent direction and the distance between each data point and memory pattern.

between $\boldsymbol{p}$ and $\boldsymbol{q}_k$ and a measure of angle s_k between $\boldsymbol{p}$ and $\boldsymbol{q}_k$. Both measures are calculated from the following equations, respectively.

$$d_k = \frac{\|\boldsymbol{q}_k - \boldsymbol{p}\|}{2\sqrt{N}}, \quad s_k = \frac{1}{\pi}\cos^{-1}\frac{\langle \boldsymbol{p}, \boldsymbol{q}_k \rangle}{\|\boldsymbol{p}\|\|\boldsymbol{q}_k\|},$$

where $d_k, s_k \in [0, 1]$. Finally, a class label c is predicted as follows:

$$c = \arg\min_k M(d_k, s_k), \quad M(d_k, s_k) = \frac{2d_k s_k}{d_k + s_k}. \quad (15)$$

Fig. 5 (b) also illustrates an image that each input vector is classified to corresponding to the memory pattern by the dynamics of the q-valued CNN.

B. Diagnosis of liver disease

In this section, we apply the associative classification method to a diagnosis problem. Kanagawa *et al.* applied a 3-valued CNN to this problem [6], where they represent the state of each inspection item in a blood test as a three-level state. However, the resolution of each inspection item's level does not seem to divide its range of values well enough. Fig. 6 shows the trend of variation of each inspection item about each disease, i.e., Normal, AH (Acute Hepatitis), CH (Chronic Hepatitis), HC (Hepatic Cirrhosis), and Hepatoma. And each

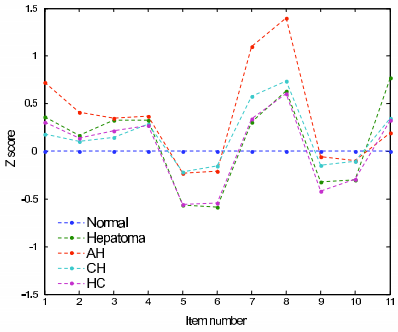


Fig. 6. Trend of variation of blood test items between normal mean value and each disease mean value. Note that each item is a Z-score using both the mean value and the standard deviation that were obtained from healthy people's blood test data.

TABLE I
INSPECTION ITEMS IN BLOOD TEST

Item #	Attribute	Item #	Attribute	Item #	Attribute
1	TBiL	5	Abl-G	9	PLT
2	DBiL	6	CHE	10	ALB
3	ALP	7	GPT	11	AFP
4	γ-GTP	8	GOT		

inspection item's name is summarized in Table I. Fig. 6 means that the resolution of each inspection level required more than three-levels to improve the diagnosis accuracy. So we apply a q-valued CNN to this diagnosis problem of liver disease.

The selected problem as a case study is the diagnosis of liver disease from blood test data. The information of this data set is summarized in Table II. Moreover, Fig. 7 shows a scatter plot of this data set in 2-D feature space, which is obtained by reducing the dimension from 11^{dim} to 2^{dim} using LDA (Linear Discrimination Analysis, refer to [10]). In this paper, we set the mean vectors as the centroid patterns in the above-described procedure.

C. Results and Discussion

The liver disease classification task using our q-valued CNN was performed. Fig. 8 shows the prediction accuracy of our q-valued CNN corresponding to output level q, which was obtained by using a 10-cross validation method. The accuracy increases as the output level q increased, and saturates from almost $q = 10$ to higher output levels. From these results,

TABLE II
SUMMARY OF THE BLOOD TEST DATA

Task	Classification
Data Set Characteristics	Multivariate
Attribute Characteristics	Real
Number of Attributes	11
Number of Instances	399
Number of Class	5

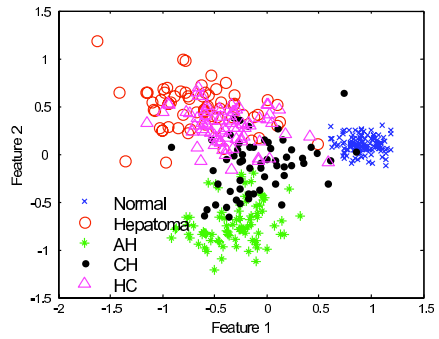


Fig. 7. Scatter plot of the blood test data with 2-dimentions, which was obtained by using LDA as a dimension reduction method.

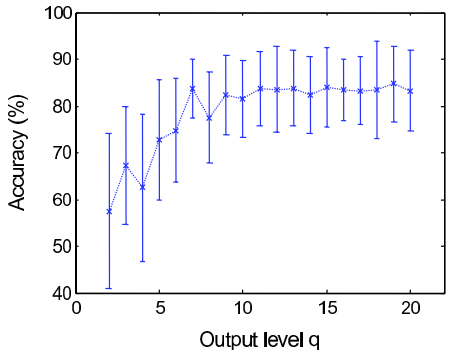


Fig. 8. Prediction accuracy with q-valued CNN corresponding to output level q ($n = 7$, $m = 5$, $q = 2 - 20$, $r_0 = 0.95$).

our q-valued CNN improved the classification accuracy by selecting the output level q properly. In addition, the optimal output level q in this task is $q = 19$ in Fig. 8.

Furthermore, we also compared the prediction accuracy of this task between the q-valued CNN and other pattern classification methods, such as template matching, k-nearest neighbor, multi-layer perceptron, and support vector machine. For more details of these methods refer to [10]. Table III shows the prediction accuracy of each method, which is obtained by using a 10-cross validation method. From this result, our q-valued CNN shows the same level of prediction accuracy. Therefore, this result also shows that multi-valued associative memory can expand both its applicability and flexibility of designing its memory patterns.

V. Conclusion

In this paper, we firstly described a design method of the multi-valued CNN which consists of cells that include a multi-valued saturation function as the output function. The multi-valued output function is defined by a linear combination of piecewise linear functions, and can construct a q-valued output level. Secondly, we conducted computer experiments to verify the error-correcting capability with our q-valued CNN designed by the eigenstructure method. These results confirm the validity of our q-valued CNN as an auto-associative memory which can store and retrieve desired q-valued patterns. Finally, we applied our q-valued CNN to disease diagnosis. This task was to classify the data of blood tests as five liver diseases: normal, AH, CH, HC, and Hepatoma. In order to solve this task, an associative classifier, which stored the centroids of each class as memory patterns, was designed. The results of the associative classifier were evaluated as its classification accuracy. These results showed that the q-valued CNN improved classification accuracy by selecting the output level q properly. Moreover, they also showed that the multi-valued associative memory could expand both the applicability and flexibility of designing its memory patterns.

TABLE III
Comparison results of other methods

Method	Model	Accuracy (%)
Template matching	Templates: mean vectors, measured by Euclid distance	85.25 ±8.24
k-Nearest neighbor	$k = 20$ measured by Euclid distance	83.00 ±8.40
Multi-layer perceptron	7-21-5 structure, $\eta = 0.05$, learned by BP method	75.75 ±14.0
Support vector machine	Gaussian ker., $\sigma = 2^{0.5}$, $C = 2^{-1}$, learned by SMO	83.50 ±8.51
q-Valued CNN	$q = 19$, designed by eigenstructure method	84.75 ±8.12

Acknowledgment

This work was supported by the Global COE Program "Frontiers of Intelligent Sensing" from the Ministry of Education, Culture, Sports, Science and Technology, Japan.

References

[1] L. O. Chua and L. Yang, "Cellular neural networks: Theory," *IEEE Trans. Circuits & Syst.*, vol. CAS-35, no. 10, pp. 1257–1272, 1988.

[2] ——, "Cellular neural networks: Applications," *IEEE Trans. Circuits & Syst.*, vol. CAS-35, no. 10, pp. 1273–1290, 1988.

[3] D. Liu and A. N. Michel, "Cellular neural networks for associative memories," *IEEE Trans. Circuits & Syst.*, vol. CAS-40, no. 2, pp. 119–121, 1993.

[4] R. Tetzlaff, *Cellular neural networks and their applications*. World scientific, 2002.

[5] Z. Zhang, M. Namba, H. Kawabata, and A. Kanagawa, "Cellular neural networks and its application for abnormal detection," *T. SICE*, vol. 39, no. 3, pp. 209–217, 2003.

[6] A. Kanagawa, H. Kawabata, and H. Takahashi, "Cellular neural networks with multiple-valued output and its application," *IEICE Trans. on Fundamentals of Electronics*, vol. E79-A, no. 10, pp. 1658–1663, 1996.

[7] K. Yokosawa, T. Nakaguchi, Y. Tanji, and M. Tanaka, "Cellular neural networks with output function having multiple constant regions," *IEEE Trans. Circuits & Syst. I*, vol. 50, pp. 847– 857, 2003.

[8] Z. Zhang, T. Akiduki, T. Imamura, and T. Miyake, "Design of multi-valued cellular neural networks for associative memory," in *Proc. of IEEE Int. Conf. on Systems, Man and Cybernetics*, 2007, pp. 1205–1210.

[9] D. Liu and A. N. Michel, "Sparely interconnected neural networks for associative memories with applications to cellular neural networks," *IEEE Trans. Circuits & Syst.*, vol. CAS-41, no. 4, pp. 295–307, 1994.

[10] P. H. Richard Duda and D. Stork, *Pattern classification 2nd ed.* Wiley-Interscience, 2001.