International Journal of Innovative
Computing, Information and Control
Volume **8**, Number **3(A)**, March **2012**

 ISSN 1349-4198
pp. 1575–1589

DESIGN OF MULTI-VALUED CELLULAR NEURAL NETWORKS FOR ASSOCIATIVE MEMORIES

Takuma Akiduki[1], Zhong Zhang[2], Takashi Imamura[2] and Tetsuo Miyake[2]

[1]Department of Electronic and Information Engineering
[2]Department of Mechanical Engineering
Toyohashi University of Technology
1-1 Hibarigaoka, Tenpaku-cho, Toyohashi, Aichi, Japan
{ akiduki; zhang; ima; miyake }@is.me.tut.ac.jp

Received December 2010; revised June 2011

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 function. In this paper, we present a multi-valued CNN model in which each non-linear 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 properly selecting the output level q. Moreover, these results also show that the multi-valued associative memory can expand both the flexibility of designing the memory pattern and its applicability.*

Keywords: Cellular neural networks, Multi-valued output function, Associative memory

1. **Introduction.** Cellular neural networks (CNNs), which were proposed by Chua in 1988 [1], have attracted a lot of attention in various fields such as signal/image processing, optimization and associative memory [2, 3, 4, 5, 6, 7]. CNNs differ from the well-known Hopfield model [8, 9] in that each non-linear element (called a *cell*) has self-feedback and a piecewise linear output function. These features provide extremely robust performance as an associative memory [10, 11].

In previous studies, however, the cell output levels have been limited to two levels $\{-1, 1\}$ or three levels $\{-1, 0, 1\}$. Thus, the CNN elements so far have only been cells of low-output resolution, which has limited the design flexibility of the memory patterns. For example, Liu and Michel reported a design procedure of CNNs for associative memories, and applied it to character recognition [12]. Since conventional CNNs have bipolar states, their associative memories are appropriate for binary image processing. Later, Kanagawa et al. proposed a three-valued CNN, and applied it to a liver disease diagnosis problem [13]. In addition, Zhang et al. reported a design method for two- or three-valued CNN output functions to improve the capability of auto-associative recall [11]. In this case, the memory vectors in the three-valued CNNs have three output states that correspond to quantized levels, such as low, middle and high. On the other hand, Yokosawa et al. have proposed a CNN with multi-valued output functions that have arbitrary multiple output states [14]. Their goal has been to extend the CNNs to image processing with multiple gray levels. To the best of our knowledge, however, the design method for associative memories based on a multi-valued CNN has not yet been reported. In this paper, we propose such a design method for associative memories using the CNN with

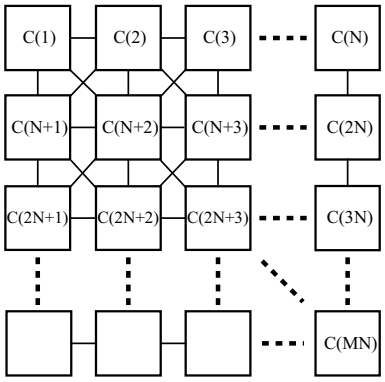


FIGURE 1. A two-dimensional cellular neural network of size $M \times N$ arranged in M rows and N columns. The i^{th} unit in the network, which is called the *cell*, is expressed by $\mathrm{C}(i)$.

a multi-valued saturation function that is called the *q-valued CNN*. The output function is first constructed from a linear combination of piecewise linear functions. Then, the properties of the associative memory using the proposed q-valued CNN are investigated by evaluating the recall capability in computer experiments. In addition, we propose a method for applying the proposed q-valued CNN to pattern classification problems. These results show that the associative memory using the proposed q-valued CNN expands the design flexibility of memory patterns and improves the diagnosis accuracy.

The remainder of this paper is organized as follows. In Section 2, we review previous works of CNN-based associative memories. In Section 3, we describe a design method for the multi-valued output function, which is defined by linear combinations of piecewise linear functions. Then we discuss the stability of the q-valued CNN. In Section 4, we conduct computer experiments to verify the error-correcting capability of associative memories using the q-valued CNN. In Section 5, we present a method for applying the q-valued CNN to a disease diagnosis problem. Finally, the study in this paper is summarized in Section 6.

2. Review of CNN-based Associative Memories.

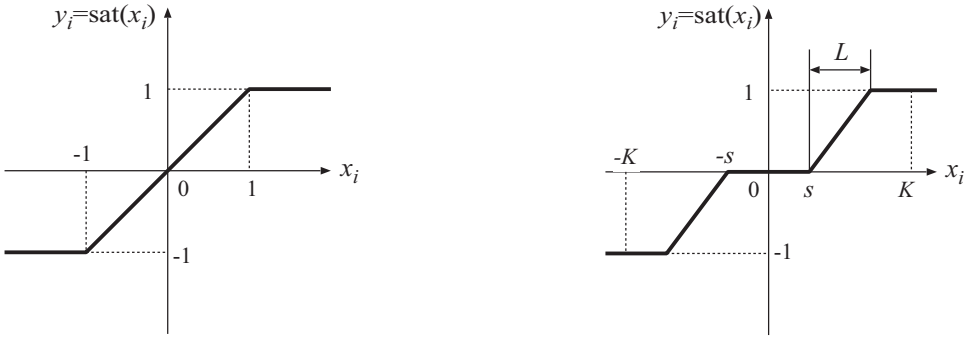
The cellular neural network (CNN) is an artificial neural network, which has a predetermined local interconnection structure. The structure of an associative memory using the CNN is designed by the eigenstructure method shown in references [12, 15]. In the following, note that R^n denotes real n-dimensional space, and $\boldsymbol{a} \in R^n$ denotes an n-dimensional column vector, respectively. Let $\mathcal{B}^n = \{\boldsymbol{y} \in R^n \mid y_i = -1 \text{ or } 1,\ i = 1, \ldots, n\}$. If $\boldsymbol{A} = [A_{ij}]$ is an arbitrary matrix, then $\boldsymbol{A}^\top$ is the transpose of $\boldsymbol{A}$.

2.1. Dynamics of a cell.

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

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

where x_i and I_i represent the state variable and bias value, respectively. T_{ij} are the coupling coefficients between the i^{th} and j^{th} cells. $\mathcal{N}_r(i)$ represents the set of r-neighborhood



(a) Two-valued output function (b) Three-valued output function

FIGURE 2. The piecewise-linear output functions of cell C(i)

cells corresponding to the i^{th} cell, where the integer r is positive. Note that in this paper to simplify the discussion, we only consider fully-interconnected cells.

In addition, $g(\cdot)$ represents the non-linear output function of each cell. The output function of the original CNN is expressed as follows:

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

The output function shown in Figure 2(a) is a piecewise-linear function, and has two saturated levels. Thus, the output value of each cell is bipolar ± 1 for $|x_i| > 1$. On the other hand, the three-valued output function shown in Figure 2(b), which was proposed by Zhang et al. in [11], is defined by the following equation:

$$\text{sat}(x_i) = \begin{cases} 1, & s + L \le x_i \\ (x - s)/L, & s < x_i < s + L \\ 0, & |x_i| < s \\ (x + s)/L, & -s - L < x_i < -s \\ -1, & x \le -s - L \end{cases}, \tag{3}$$

where the parameters s and L represent the length of saturation and slope region, respectively. The parameter K in Figure 2(b) represents the convergence point of the cell. In addition, Zhang et al. reported experimentally determined parameters with $s = 0.3$, $L = 0.5$ and $K = 1.335$. The function shown in Figure 2(b) has three saturated levels: $\{-1, 0, 1\}$. In other words, this function is a three-valued CNN output function.

2.2. **Design procedure.** First we express the differential equation of (1) in vector notation. The two-dimensional CNNs, which have M rows and N columns, are represented as follows:

$$\frac{d}{dt}\boldsymbol{x}(t) = -\boldsymbol{x}(t) + \boldsymbol{T}\boldsymbol{y} + \boldsymbol{I}, \tag{4}$$

where $\boldsymbol{x} = [x_i] \in R^n$, $\boldsymbol{y} = [y_i] = [\text{sat}(x_i)] \in R^n$ and $\boldsymbol{I} = [I_i] \in R^n$ represent a state vector, an output vector, and a bias vector, respectively. The matrix $\boldsymbol{T} = [T_{ij}] \in R^{n \times n}$ is a template matrix composed of row vectors whose elements are zero when the corresponding cells have no connections. The network in (4) has a number of asymptotically stable equilibrium points, since all their n eigenvalues in (4) are -1.

To design the CNN as an associative memory, we compute the coefficient matrix $\boldsymbol{T}$ and the bias vector $\boldsymbol{I}$ in (4) by following Liu and Michel's design procedure detailed in [12, 15].

Suppose that we are given m desired memory patterns $\boldsymbol{\alpha}^1, \ldots, \boldsymbol{\alpha}^m$, and assume the asymptotic stable equilibrium points $\boldsymbol{\beta}^k$ corresponding to $\boldsymbol{\alpha}^k$ are such that:

$$\boldsymbol{\beta}^k = K\boldsymbol{\alpha}^k \quad (k = 1, \ldots, m), \tag{5}$$

where the real number K is the convergence point of the cells. When the systems in (4) 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 $\boldsymbol{y}_e = \mathrm{sat}(\boldsymbol{x}_e) = \boldsymbol{\alpha}^k$. By assigning $\boldsymbol{\alpha}^k$ and $\boldsymbol{\beta}^k$ to (4) with $\dot{\boldsymbol{x}}(t) = 0$ the following equation can be obtained.

$$\begin{cases} -\boldsymbol{\beta}^1 + \boldsymbol{T}\boldsymbol{\alpha}^1 + \boldsymbol{I} = 0 \\ -\boldsymbol{\beta}^2 + \boldsymbol{T}\boldsymbol{\alpha}^2 + \boldsymbol{I} = 0 \\ \quad \vdots \\ -\boldsymbol{\beta}^m + \boldsymbol{T}\boldsymbol{\alpha}^m + \boldsymbol{I} = 0 \end{cases}. \tag{6}$$

Here we set matrices $\boldsymbol{Y}$ and $\boldsymbol{Z}$ as follows:

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

and obtain the following equations:

$$\boldsymbol{Z} = \boldsymbol{T}\boldsymbol{Y}, \tag{7}$$
$$\boldsymbol{I} = \boldsymbol{\beta}^m - \boldsymbol{T}\boldsymbol{\alpha}^m. \tag{8}$$

The matrix $\boldsymbol{T} \in R^{n\times n}$ can be solved from matrix $\boldsymbol{Y}$, $\boldsymbol{Z} \in R^{n\times(m-1)}$. Since $\boldsymbol{Y}$ is a non-square matrix, we apply singular value decomposition (SVD) to the matrix $\boldsymbol{Y}$ to obtain the template matrix $\boldsymbol{T}$. Furthermore, $\boldsymbol{I}$ can be obtained by solving (8). In the network designed by this method, all of the desired patterns are guaranteed to be stored as *asymptotically stable* equilibrium points [16].

3. CNNs with a Multi-Valued Output Function.

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

3.1. *q*-valued output function.

To extend the conventional output function in (2) to our multi-valued output function, we introduce a *base wave function* in Figure 3(a). 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:

$$g_0(x_i) = \frac{1}{L}\left(\left|x_i + \frac{L}{2}\Delta_q\right| - \left|x_i - \frac{L}{2}\Delta_q\right|\right), \tag{9}$$

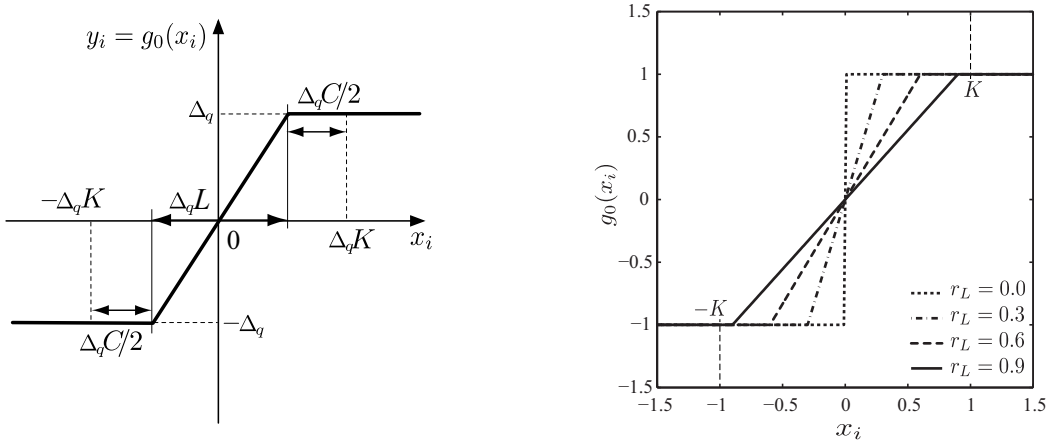
where $\Delta_q = 1/(q-1)$ and q is the output-level. In addition, we assume that the equilibrium points of each cell are determined by a parameter K. Thus, the values of both parameters C and L are defined by the following equation:

$$K = \frac{1}{2}(C + L), \text{ where } C, L > 0, \tag{10}$$

and the ratio of parameter L to $C + L$ is as follows:

$$r_L = \frac{L}{C+L} = \frac{L}{2K}. \tag{11}$$

Figure 3(b) illustrates the relation between the parameter r_L and the output characteristics in (9). If $r_L = 0$, then the function in (9) is the sign function. If $0 < r_L < 1$, then



(a) Definition of $g_0(x_i)$ (b) Output characteristics

FIGURE 3. The base wave function and its parameters, where (a) defines the base function of our multi-valued output function, and (b) shows the output characteristics in (9) with $K = 1.0$ and $q = 2$

the function in (9) is a piecewise-linear function. In addition, if $r_L = 1$, then the function in (9) is a linear function.

Since the base function in (9) has two saturation ranges, the saturation ranges with q-level are generated by linear combinations of (9) with both shrinking and shifting of the waveforms. Therefore, a q-valued output function is defined as follows:

$$g_q(x_i) = g_0(x_i) \times \delta + \sum_{j=1}^{P} \{g_0(x_i - \gamma_j) + g_0(x_i + \gamma_j)\}, \tag{12}$$

where $P = (q - 2 + q \bmod 2)/2$, and the coefficients (δ, γ_j) in (12) are represented as follows:

$$(\delta, \gamma_j) = \begin{cases} (0, (2j-1)K\Delta_q), & \text{if } q \text{ is odd} \\ (1, 2jK\Delta_q), & \text{if } q \text{ is even.} \end{cases}$$

If $q = 2$, the output function in (12) is the same as the function in (9): $g_2(x_i) = g_0(x_i)$. In addition, we introduce the following notation.

For integer q as the output level of CNNs, let

$$\mathcal{M}_q^n = \{\boldsymbol{y} \in R^n \mid y_i = (-q + 2k - 1)\Delta_q,\ k = 1, \ldots, q;\ i = 1, \ldots, n\}. \tag{13}$$

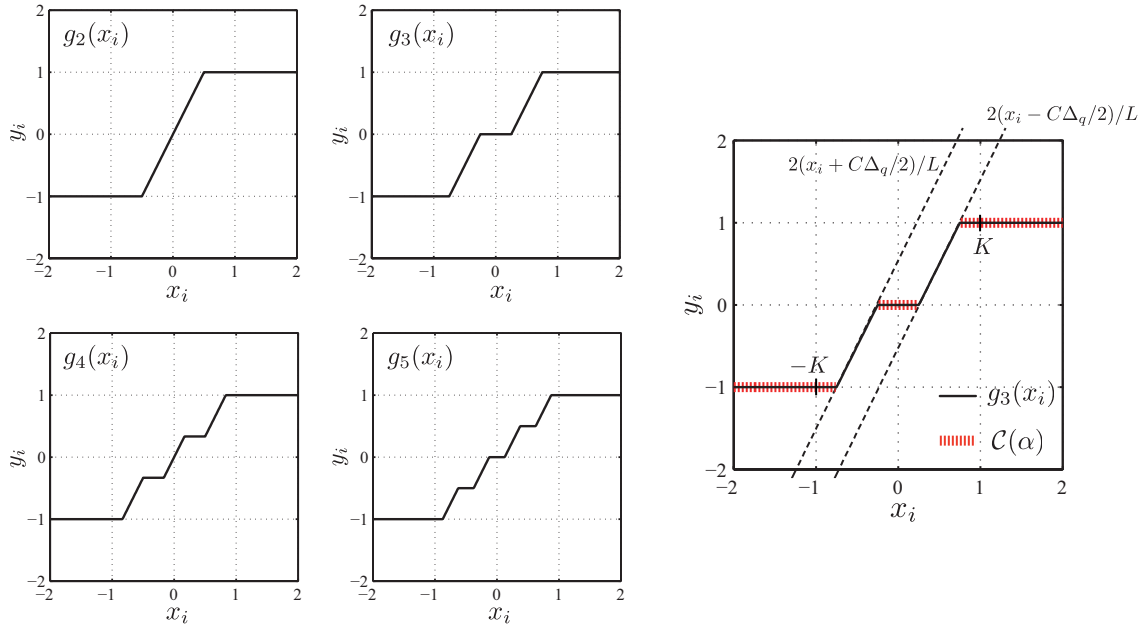
For example, if the output-level is $q = 4$, $\mathcal{M}_4 = \{y \in R \mid y = \pm 1 \text{ or } \pm 1/3\}$. Also, if $q = 3$ as shown in Figure 4(b), then $\mathcal{M}_3 = \{y \in R \mid y_i = \pm 1 \text{ or } 0\}$, and for the memory vector $\boldsymbol{\alpha} \in \mathcal{M}_3^n$, let

$$\mathcal{C}(\boldsymbol{\alpha}) = \left\{ \boldsymbol{x} \in R^n \;\middle|\; \begin{array}{ll} x_i > (C/2 + L)\Delta_q & \text{if } \alpha_i = 1 \\ |x_i| < C\Delta_q/2 & \text{if } \alpha_i = 0, \quad i = 1 \ldots, n \\ x_i < -(C/2 + L)\Delta_q & \text{if } \alpha_i = -1 \end{array} \right\},$$

where $\mathcal{C}(\boldsymbol{\alpha})$ is the saturation region in the multi-valued output function in (12). In the saturation region, we have $g_3(\boldsymbol{x}) = \boldsymbol{\alpha}$ for $\boldsymbol{x} \in \mathcal{C}(\boldsymbol{\alpha})$ because $\alpha_i = \pm 1$ or 0 for all i. Thus the system (4) of $\boldsymbol{x}$ for $\boldsymbol{x} \in \mathcal{C}(\boldsymbol{\alpha})$ is

$$\dot{\boldsymbol{x}} = -\boldsymbol{x} + \boldsymbol{T}\boldsymbol{\alpha} + \boldsymbol{I}, \quad \boldsymbol{\alpha} = g_q(\boldsymbol{x}). \tag{14}$$

As $\boldsymbol{\beta} = \boldsymbol{T}\boldsymbol{\alpha} + \boldsymbol{I}$ is constant, this equation has an asymptotic equilibrium point at $\boldsymbol{x}_e = \boldsymbol{\beta}$. If $\boldsymbol{\beta} \in \mathcal{C}(\boldsymbol{\alpha})$, this equilibrium point is also asymptotically stable since all the eigenvalues



(a) q-valued output functions (b) Output characteristics of $g_3(\cdot)$

FIGURE 4. Examples of multi-valued output functions in (12), where (a) shows waveforms with $q = 2, 3, 4, 5$, having parameters $K = 1.0$ and $r_L = 0.5$, and (b) illustrates saturation and slope regions in $g_3(\cdot)$

of (14) are -1. In Section 2.2, it is shown $\boldsymbol{T}$ and $\boldsymbol{I}$ can be constructed such that $\boldsymbol{\beta} = \boldsymbol{T}\boldsymbol{\alpha} + \boldsymbol{I} \in \mathcal{C}(\boldsymbol{\alpha})$ for a given $\boldsymbol{\alpha}$. So the CNN using the three-valued output function in (12) has *three*-levels as the cell states: $\{-1, 0, 1\}$. In the case of general multiple q-valued output, it can be proved in the same manner that the CNN has the corresponding number of convergence points. We call this network a *q-valued CNN*.

3.2. **Stability of q-valued CNNs.** To design an associative memory, the CNN must always converge to a constant steady state, which is initialized by a given input pattern. For analyzing the convergence properties of the proposed q-valued CNN, the Lyapunov function [1, 14] is as follows:

$$E(t) = -\frac{1}{2}\sum_i\sum_j T_{ij}y_i(t)y_j(t) - \sum_i I_i y_i(t) + \sum_i \int_0^{y_i(t)} g_q^{-1}(v)dv, \tag{15}$$

where $g_q^{-1}(s)$ is the inverse function of the multi-valued output function shown in (12). In the following theorem, we prove that $E(t)$ in (15) is a monotone decreasing function.

Theorem 3.1. *If $T_{ij} = T_{ji}$, the energy function in* (15) *is monotonously decreasing.*

Proof: From the symmetry assumption $T_{ij} = T_{ji}$, the differentiation of $E(t)$ in (15) with respect to time t is as follows:

$$\frac{dE(t)}{dt} = -\sum_i\sum_j T_{ij}\frac{dy_i}{dx_i}\frac{dx_i(t)}{dt}y_j - \sum_i I_i\frac{dy_i}{dx_i}\frac{dx_i(t)}{dt} + \sum_i \frac{dy_i}{dx_i}\frac{dx_i(t)}{dt}\frac{d}{dy_i}\int_0^{y_i} g_q^{-1}(\tau)d\tau. \tag{16}$$

Here,

$$\frac{dy_i}{dx_i} = \begin{cases} 2/L & \text{if } x_i \notin \mathcal{C}(\boldsymbol{\alpha}) \\ 0 & \text{if } x_i \in \mathcal{C}(\boldsymbol{\alpha}) \end{cases}.$$

The inverse function $g_q^{-1}(y)$ of the multi-valued output function in (15), however, does not exist because $g_q^{-1}(y)$ is underspecified for x. Thus, according to the Yokosawa et al. definition in [14], we consider the following pseudoinverse function:

$$\frac{d}{dy_i}\int_0^{y_i} g_q^{-1}(v)dv \approx \frac{d}{dy_i}\left(\frac{1}{2}Ky_i^2\right) = x_i.$$

For (16), if $x_i(t)$ is in the slope region, we obtain

$$\frac{dE(t)}{dt} = -\sum_i \frac{2}{L}\frac{dx_i(t)}{dt}\left(\sum_j T_{ij}y_j + I_i - x_i(t)\right). \tag{17}$$

Substituting the cell dynamics (1) into (17), and recalling $L > 0$ in assumption (10), we obtain

$$\frac{dE(t)}{dt} = -\sum_i \frac{2}{L}\left(\frac{dx_i(t)}{dt}\right)^2 \le 0. \tag{18}$$

Theorem 3.2. *If the parameter K satisfies*

$$K > \frac{L}{2}$$

the CNN has a stable equilibrium point at any constant region of the output function $g_q(x)$.

Proof: If the matrix $\boldsymbol{T}$ in (4) satisfies (7), given m vectors $\boldsymbol{\alpha}^1, \ldots, \boldsymbol{\alpha}^m$ are stored as memory vectors for (4). From $\boldsymbol{Z} = K\boldsymbol{Y}$, we can rewrite (7) as

$$(K\boldsymbol{E} - \boldsymbol{T})\boldsymbol{Y} = 0, \tag{19}$$

where $\boldsymbol{E}$ is a unit matrix. The indeterminate Equation (19) has the following trivial solution:

$$T_{ij} = \begin{cases} K, & i = j \\ 0, & i \ne j \end{cases}. \tag{20}$$

To simplify the discussion, we consider the i^{th} cell of the CNN given by (20). Then the cell dynamics in (1), which has no connections with its neighborhood cells, are rewritten in the following equation:

$$\frac{dx_i(t)}{dt} = -x_i(t) + T_{ii}y_i + h(t), \text{ where } h(t) = \sum_{j\ne i} T_{ij}y_j + I_i.$$

If $h(t) = 0$, then

$$\frac{dx_i(t)}{dt} = -x_i(t) + Ky_i, \text{ where } y_i(t) = \begin{cases} \dfrac{2}{L}x_i(t) + \dfrac{C}{L}\Delta_q\gamma & \text{if } x_i \notin \mathcal{C}(\boldsymbol{\alpha}) \\ \xi & \text{if } x_i \in \mathcal{C}(\boldsymbol{\alpha}) \end{cases}. \tag{21}$$

Here, the intercept coefficient $\gamma = -q + 2k$, $k \in \{1, 2, \ldots, q-1\}$ corresponds to $x_i(t)$ and the output value in the saturation region, $\xi \in \mathcal{M}_q$. Then, the solution of (21) is obtained as follows:

$$x_i(t) = \begin{cases} -K\Delta_q\gamma + (x_i(0) + K\Delta_q\gamma)\exp\{(2K/L - 1)\,t\} & \text{if } x_i(t) \notin \mathcal{C}(\boldsymbol{\alpha}) \quad (22) \\ (x_i(0) - K\xi)\exp(-t) + K\xi & \text{if } x_i(t) \in \mathcal{C}(\boldsymbol{\alpha}). \quad (23) \end{cases}$$

Since $2K/L > 1$, the cell (21) has $q-1$ unstable equilibrium points at $x_i(t) = K\Delta_q\gamma$, and q asymptotic stable equilibrium points at $x_i(t) = K\xi$. Hence, if any equilibrium point exists in the linear regions, it is unstable. Therefore, all cells of the CNN have a stable equilibrium point at the saturation regions of $g_q(\cdot)$.

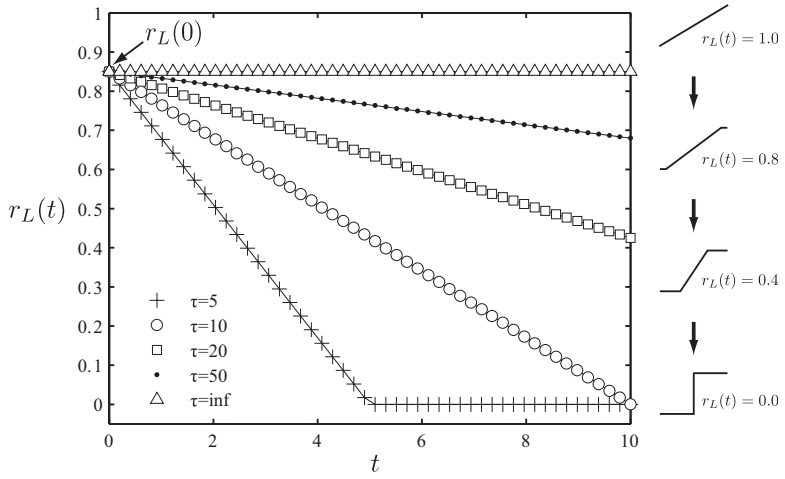


FIGURE 5. An example of the time-varying parameter $r_L(t)$ with $r_L(0) = 0.85$

From the above theorems, the steady-state behavior of the q-valued CNN is summarized by the following theorem.

Theorem 3.3. *Under the assumptions*

$$T_{ij} = T_{ji} \quad \textit{and} \quad K > \frac{L}{2}$$

and constraint condition

$$|x_i(0)| \leq K$$

all the cells of the q-valued CNN have a stable equilibrium point in the saturation regions.

3.3. **Time-varying parameter.** To improve the performance of auto-associative recall of the q-valued CNN, we introduce a time-varying parameter; it is a slope parameter with time-variant $r_L(t)$. This parameter $r_L(t)$ produces an annealing effect on the energy function in the state space of CNNs. The parameter r_L in (11) changes the characteristics of the output function shown in Figure 3(b). If $r_L \to 0$, the output function in (9) forms an approximate sign function. This means the peaks of the energy function in (15) have steep gradients. On the other hand, if $r_L \to 1$, the output function in (9) forms a linear function. This means the energy function (15) has gently-sloping peaks. Therefore, adjusting the slope parameter L can be expected to improve convergence to the desired equilibrium points [17]. For time step t, like in the annealing schedule, an update role of the slope parameter $r_L(t)$ is defined as the following equation:

$$r_L(t) = \max\left(r_L(0) \times (1 - t/\tau), 10^{-6}\right), \tag{24}$$

where $r_L(0)$ is an initial value of the $r_L(t)$, and τ is a time constant. If $\tau = \inf$, r_L remains constant. For the time constant τ, Figure 5 shows an example of the time-varying parameter. Note that the interval of numerical integration is set from $t_0 = 0$ to $t_f = 10$ in this paper.

4. **Numerical Examples.** We evaluate the error-correcting performance of the proposed q-valued CNN by using computer simulation.

Each q-valued CNN in these examples is constructed with $n = 400$ cells in order to memorize the linearly-independent vectors $\boldsymbol{\alpha}^k \in \mathcal{M}_q^{400}$ $(k = 1, \ldots, 120)$. Both the template matrix $\boldsymbol{T}$ and bias vector $\boldsymbol{I}$ are designed following the procedure in Section 2.2.

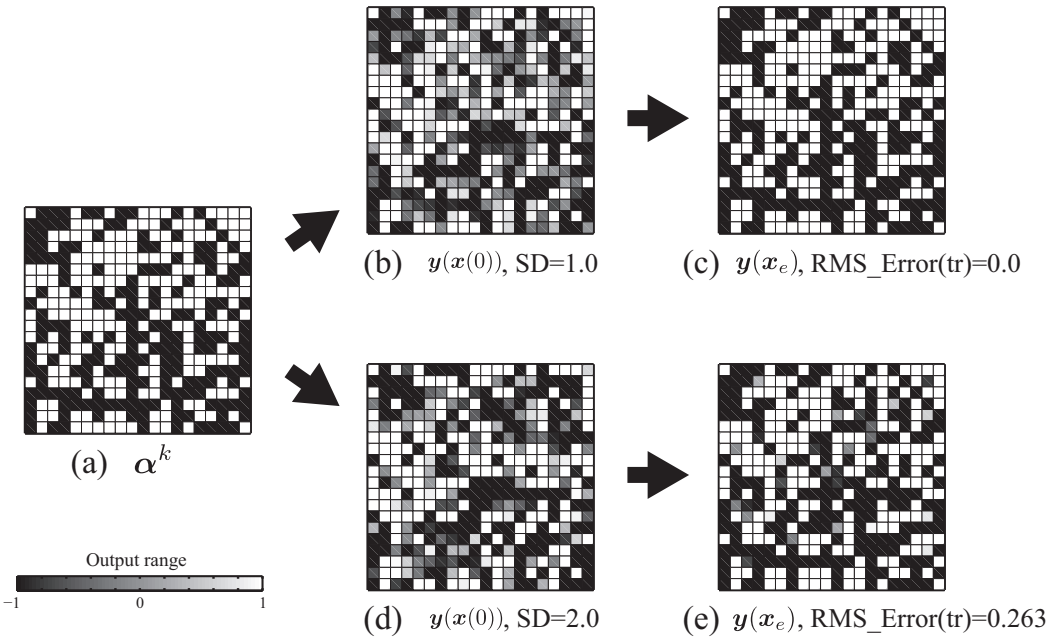


FIGURE 6. Example of the error correction in the q-valued CNN ($q = 2$), where (a) shows the memory pattern with 20×20 cells, (b) the initial pattern that was made by adding white noise with SD = 1.0 to the memory pattern (a), (c) the recalled pattern obtained by the q-valued CNN from the initial pattern (b), (d) the initial pattern that was made by adding white noise with SD = 2.0 to the memory pattern (a) and (e) the recalled pattern obtained by the q-valued CNN from the initial pattern (d)

Noisy memory patterns are generated as inputs by the following equation:

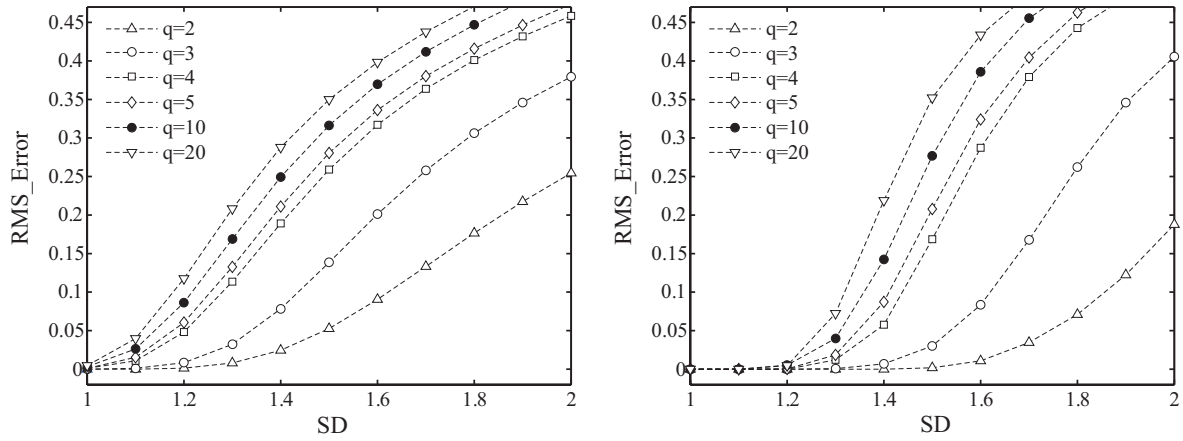
$$\boldsymbol{x}(0) = K(\boldsymbol{\alpha}^k + \boldsymbol{\varepsilon}), \quad k \in \{1, \ldots, 120\}, \tag{25}$$

where $\boldsymbol{\varepsilon} = [\varepsilon_i \times \Delta_q] \in R^n$ is the noise vector, whose elements ε_i correspond to the normal distribution of a certain standard deviation (SD). Then the elements are scaled by the output level q; $\Delta_q = 1/(q-1)$. The inputs are amplified by the parameter $K = 1.0$ for both the randomly-selected memory pattern and the noise vector in (25). The states of network $\boldsymbol{x}(t)$ in (4) are solved by using numerical integration [1]. By recording the final state of the network, the root mean squared error is computed by the following equation.

$$RMS\_Error(i_{tr}) = \frac{1}{2\Delta_q\sqrt{n}}\|\boldsymbol{\alpha}^k - \boldsymbol{y}_e\|, \quad i_{tr} = 1, 2, \ldots N_{tr}. \tag{26}$$

The average of $RMS\_Error(i_{tr})$: RMS_Error is computed to follow (26) among $N_{tr} = 300$ trials, where the k^{th} vector $\boldsymbol{\alpha}^k$ is selected by sampling with repetition from the set of memory patterns $\{1, \ldots, 120\}$. Figure 6 illustrates examples of auto-associative recall, which are evaluated by the $RMS\_Error(i_{tr})$ in (26). Figure 6(a) shows an example memory pattern, (b) the initial patterns that were made by adding white noise with SD = 1.0 to the memory pattern (a), (c) the recalled pattern obtained by the q-valued CNN from the initial pattern (b), (d) the initial pattern that was made by adding white noise with SD = 2.0 to the memory pattern (a), and (e) the recalled pattern obtained by the q-valued CNN from the initial pattern (d).

[1]In this paper, we used the fourth-order Runge-Kutta algorithm with an adaptive step-size routine: `ode45.m`, which is MATLAB's standard solver for ordinary differential equations.



(a) $\tau = \inf$, $r_L(0) = 0.85$ (b) $\tau = 20$, $r_L(0) = 0.85$

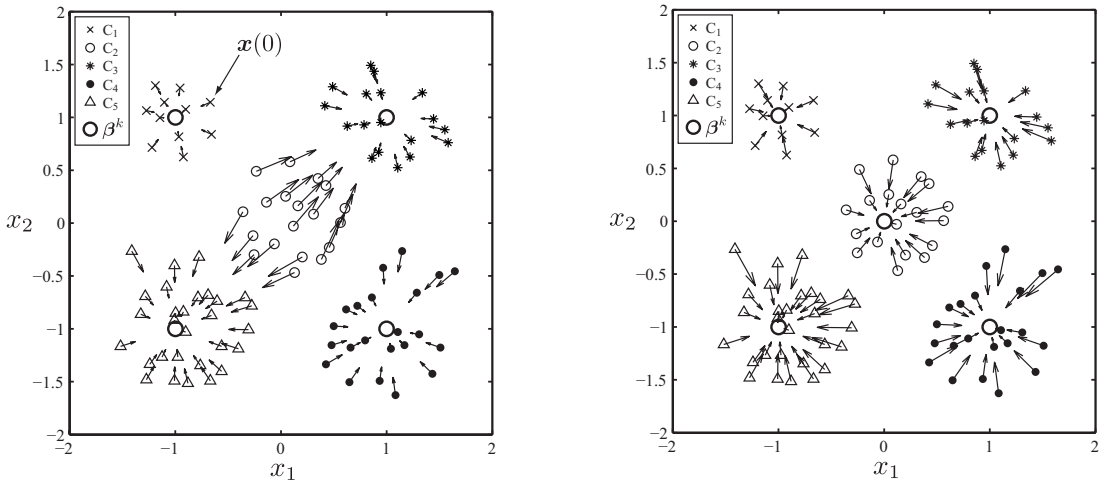
FIGURE 7. Error correcting capabilities for the white noise of the certain standard deviation (SD)

Figure 7 shows that the associative memories based on the proposed q-valued CNN succeed in recalling multi-valued patterns. The effect of the time constant τ for the time-varying slope parameter $r_L(t)$ can be seen by comparing Figures 7(a) and 7(b). The initial value of slope $r_L(0) = 0.85$ was empirically selected to be the best-performing in terms of error correction capability. Figure 7(b) shows that the proposed q-valued CNNs with the time-varying $r_L(t)$ improve the error correcting capability of the conventional CNNs with the *time-invariant* r_L which include the three-valued CNNs shown in [11]. From the result of our preliminary experiments, the time constant τ should be selected within the range of approximately 2.0 times the interval of integration, i.e., $\tau = 20$ in the case of this paper. The result of our experiment corresponds with related work [17].

5. **Application.** In this section, we consider a method for applying the proposed q-valued CNN to pattern classification problems. These problems are aimed at diagnosis problems in the fields of medicine or mechanical faults. In the following, we firstly introduce a classification method by using the proposed q-valued CNN that is called *associative classification*. Then, the associative classification is applied to the diagnosis of liver disease as a typical example in the medical field.

5.1. **The method.** We consider the classification method based on associative memory neural networks. The data to classify, which is a set of feature values, are expressed as a bit map pattern, and stored as a *memory pattern* in the network. The network classifies the inputs by recalling a correlated memory pattern using network dynamics. The associative classification procedure using the proposed q-valued CNN is as follows:

1) Consider a q-valued CNN with $n = M \times N$ cells to allocate n features $f_1, \ldots, f_n$ to each cell.
2) Compute the centroids for each cluster C_k $(k = 1, 2, \ldots, m)$ in the feature space from a set of training data, and then discretize the centroid vectors corresponding to the output level q using (12) with $r_L = 0$. These computed vectors $\boldsymbol{\alpha}^k \in R^n$, $k = 1, \ldots, m$ are called memory patterns, where m is the number of classes.
3) Store the memory patterns in the network of (4) by using Liu and Michel's design procedure in Section 2.2.



(a) $q = 2$ (b) $q = 3$

FIGURE 8. An image of process of auto-associative classification in two-dimensional state space with the proposed q-valued CNN. (a) Illustrates the stored four vectors in 2-valued CNN ($[1,1]^\top$, $[-1,1]^\top$, $[1,-1]^\top$ and $[-1,-1]^\top$); (b) illustrates the stored five vectors in 3-valued CNN ($[1,1]^\top$, $[-1,1]^\top$, $[1,-1]^\top$, $[-1,-1]^\top$ and $[0,0]^\top$).

4) Set an input vector in the set of test data to be the initial state $\boldsymbol{x}(0) = \boldsymbol{x}_0$ of (4). Then the output of limit state $\boldsymbol{y}_e$ is obtained as a recalled pattern by solving its q-valued CNN by using the Runge-Kutta algorithm.
5) Discriminate a class of the input vector from the following similarity measurement between the recalled pattern $\boldsymbol{y}_e$ and the memory patterns $\boldsymbol{\alpha}^k$. First, we calculate both vectors $\boldsymbol{p}$ and $\boldsymbol{q}_k$ by the following equations:

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

Then we also calculate a measure of distance d_k between $\boldsymbol{p}$ and $\boldsymbol{q}_k$ and 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 $d_k, s_k \in [0,1]$. As a result, the class of the input vector is predicted as follows:

$$\textit{Predicted\_Label} = \arg\min_k M(d_k, s_k), \text{ where } M(d_k, s_k) = \frac{2d_k s_k}{d_k + s_k}. \tag{27}$$

Figure 8 illustrates an image of associative classification. Each input vector is classified to correspond to the memory pattern by the dynamics of the proposed q-valued CNN. Note that each unfilled black circle represents the position of the memory pattern, and the other plotted dots represent the positions of each input vector. Each arrow represents the convergent direction and distance to the memory pattern. Moreover, from Figures 8(a) and 8(b), we can see that it is possible to improve the probability of correct classification by a proper choice of the output level q.

5.2. **Diagnosis of liver disease.** In previous studies [11, 13], the data of medical blood tests shown in Table 1 were used as the confirmation data set. In Table 1, f_i denotes i^{th} inspection item after normalizing the blood test result u_i by the scaling function.

Each q-valued CNN in this problem is constructed with $n = 12$ cells to memorize five vectors $\boldsymbol{\alpha}^k \in \mathcal{M}_q^{12}$ $(k = 1, \ldots, 5)$ corresponding to classes of liver disease. The memory

TABLE 1. Inspection items in blood test

Item	Attribute	Scaling function
f_1	TBiL	$\log(u_1/50) + 3.71$
f_2	DBiL	$(u_2 - 50)/26$
f_3	ALP	$(u_3 - 80)/30$
f_4	γ-GTP	$u_4/35 - 1.2$
f_5	Abl-G	$(u_5 - 3.8)/0.55$
f_6	CHE	$(u_6 - 215)/93.75$
f_7	GPT	$\log(u_7/90)$
f_8	GOT	$\log(u_8/90)$
f_9	PLT	$(u_9 - 23)/13$
f_{10}	ALB	$(u_{10} - 54.8)/6.5$
f_{11}	AFP	$\log(u_{11}/50)$
f_{12}	AFP	$= f_{11}$

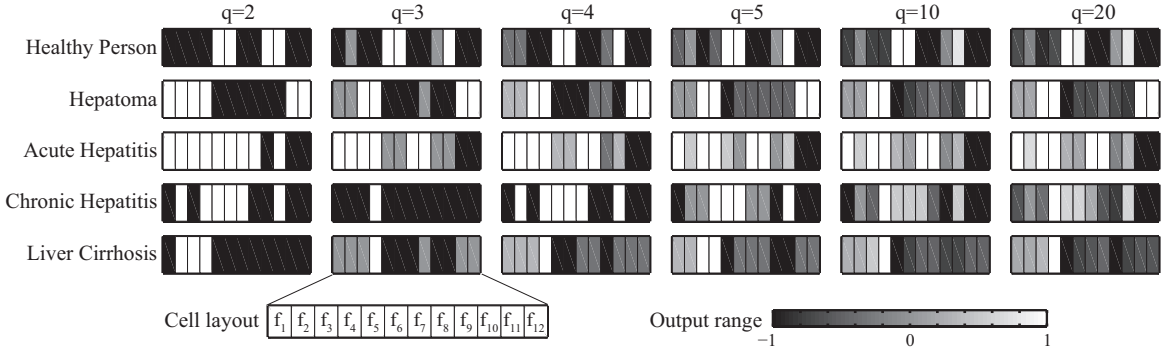


FIGURE 9. Patterns of five liver diseases for q-valued CNN

patterns are computed by the following equation:

$$\boldsymbol{\alpha}^k = g_q(\boldsymbol{f}^k_{ave}) \mid_{r_L=0,\, K=1.2}, \text{ where } \boldsymbol{f}^k_{ave} = \frac{1}{N_k} \sum_{j \in C_k} \boldsymbol{f}_j \text{ for each } k. \tag{28}$$

Here, N_k is the number of data $\boldsymbol{f}_j$ that is a sample of class C_k. The inputs are amplified by $K = 1.2$ before being fed into these networks. The slope of the output in (24) is defined by $r_L(0) = 0.85$ and $\tau = 20$.

Figure 9 illustrates the memory patterns for the five classes, i.e., "Healthy person", "Hepatoma", "Acute Hepatitis", "Chronic Hepatitis" and "Liver Cirrhosis". As indicated in Figure 9, each box represents the q-valued memory pattern of the disease, with each cell corresponding to a vector component which is allowed to assume values between -1 and 1. Here, -1 represents black, 1 represents white and the intermediate values correspond to intermediate grey levels.

To evaluate the result of diagnosis, we define a classification accuracy, which depends on the number of correctly classified samples.

$$Accuracy = T_{correct}/N_{all} \times 100(\%), \tag{29}$$

where $T_{correct}$ is the number of sample cases correctly classified, and N_{all} is the total number of sample cases. In this paper, we used 399 sample data ($N_{all} = 399$). The accuracy was computed by counting the number of all correctly classified samples obtained by the 10-cross validation method [18].

5.3. **Results and discussion.** By using the proposed q-valued CNN, the liver disease classification task shown in [11, 13] was performed. Figure 10 shows that the proposed q-valued CNN improves the classification accuracy of conventional CNNs at $q = 5$. The accuracy increases as the output level q increases, and saturates from approximately $q = 10$ to higher output levels. Hence, the q-valued CNN improved the classification accuracy by selecting the output level q properly. Moreover, the optimal output level q in this task was $q = 5$ in Figure 10.

On the other hand, we also compared the classification accuracy of the same task between the proposed q-valued CNN and other pattern classification methods, such as template matching (TMP), k-nearest neighbor (k-NN), multi-layer perceptron (MLP), support vector machine (SVM), and the three-valued CNN proposed by Zhang et al. in [11] (3-CNN). The TMP uses the mean vectors for each class as the templates. The k-nearest neighbor uses $k = 20$ prototype vectors. The MLP is constructed by two hidden layers and four total layers, which has 11 input neurons, $33 + 17$ hidden neurons, and 5 output neurons. The weights in the MLP are trained by the back propagation algorithm with learning rate $\eta = 0.005$. The SVM uses the Radial Basis Function (RBF) $ker(\boldsymbol{x}_1, \boldsymbol{x}_2) = \exp(-0.5\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|^2/\sigma^2)$ with $\sigma = 2^{0.5}$ and regularization constant $C_{reg} = 2^1$. Moreover, the 3-CNN in [11] uses the output function of (3). For each classifier, the accuracy of (29) is computed by using the 10-cross validation method. The parameters of each classifier are selected to maximize the accuracy. For more details of these methods and implementations using MATLAB refer to [18, 19].

From Figure 11, we can see that the proposed q-valued CNN improves the classification accuracy of the 3-CNN in [11] by more than 8%. Moreover, the classification accuracy of the proposed q-valued CNN is higher than those for typical classification methods, i.e., MLP and SVM, although it is smaller than those for the TMP and k-NN. The reasons being considered are that the local partition approaches (TMP, k-NN and q-CNN) are superior to the global partition approaches (SVM and MLP) for this example since the dataset has high within-class variance relative to between-class variance. On the other hand, one of the main advantages of associative classification using the proposed q-valued CNN is that the amount of calculation for solving (4) shows no further increase with subsequent additions of data. The associative memory using the proposed q-valued CNNs can expand both the applicability and flexibility of designing its memory vectors. Since the CNNs can perform parallel signal processing in real time through VLSI implementation, the results in this paper suggest that it may be possible to obtain a novel architecture for a high-speed and fault-tolerant pattern recognition system by using the proposed q-valued CNN.

6. **Conclusions and Remarks.** In this paper, we proposed a new q-valued CNN for associative memories that includes 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 the q-valued output level. The stability of the CNN with the multi-valued output function was then discussed based on the theorems of [1, 14].

Numerical examples showed that the proposed q-valued output function with time-varying parameter $r_L(t)$ improved the error correction capability of the associative memory using q-valued CNNs. Finally, we showed the method of associative classification with the proposed q-valued CNN and demonstrated the diagnosis of liver disease. For output level q, these results showed that the proposed q-valued CNN improved classification accuracy by properly selecting the output level q.

As further work we need to investigate the capacity and limits of associative memory in the proposed q-valued CNN and explore the relationship between the output level q

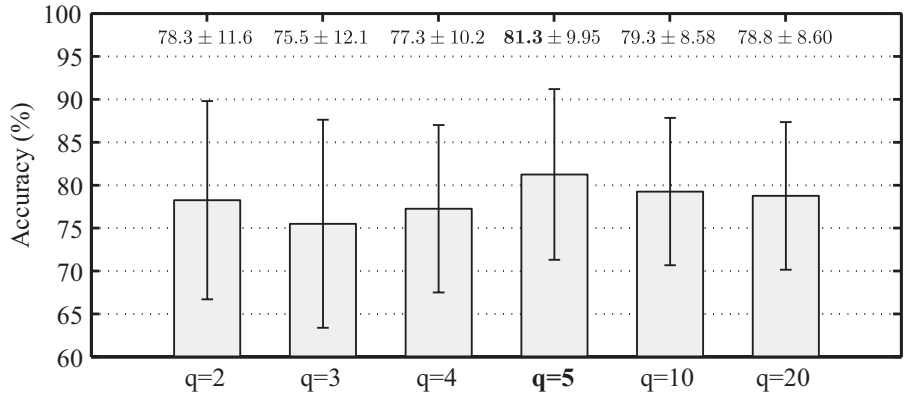


FIGURE 10. Classification results of the proposed q-valued CNN for output level q

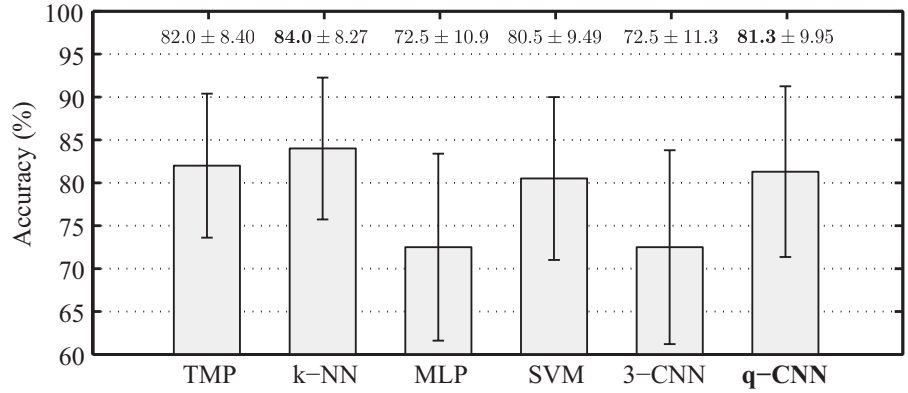


FIGURE 11. Comparison results of the conventional classification method (TMP, k-NN, MLP, SVM and 3-CNN) and the proposed q-valued CNN (q-CNN)

and the number of memory vectors stored for the network. Moreover, we also need to try to develop a design procedure of associative memories, which uses a convex optimization method, i.e., LMI (linear matrix inequality) in [20], to further improve the recall capability of the proposed q-valued CNN.

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. The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

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

[2] L. O. Chua and L. Yang, Cellular neural networks: Applications, *IEEE Trans. Circuits & Systems*, vol.35, no.10, pp.1273-1290, 1988.

[3] K. R. Crounse and L. O. Chua, Methods for image processing and pattern formation in cellular neural networks: A tutorial, *IEEE Trans. Circuits & Systems I*, vol.42, no.10, pp.583-601, 1995.

[4] M. Ercsey-Ravasz, T. Roska and Z. Néda, Cellular neural networks for NP-hard optimization, *EURASIP Journal on Advances in Signal Processing*, 2009.

[5] J. Park, H. Y. Kim, Y. Park and S. W. Lee, A synthesis procedure for associative memories based on space-varying cellular neural networks, *Neural Networks*, vol.14, no.1, pp.107-113, 2001.

[6] R. Tetzlaff, *Cellular Neural Networks and Their Applications*, World Scientific Publishing, Singapore, 2002.

[7] M. Namba, Design of tri-valued output cellular neural network for associative memory for self-directed e-learning, *ICIC Express Letters, Part B: Applications*, vol.2, no.3, pp.583-588, 2011.

[8] J. J. Hopfield, Neural networks and physical systems with emergent collective computational abilities, *Proc. of National Academy of Sciences of U.S.A*, vol.79, no.8, pp.2554-2558, 1982.

[9] J. J. Hopfield, Neurons with graded response have collective computational properties like those of two-state neurons, *Proc. of National Academy of Sciences of U.S.A*, vol.81, no.10, pp.3088-3092, 1984.

[10] Z. Zhang and H. Kawabata, Examinations of associative memories in Hopfield and cellular neural networks, *Journal of the Japanese Society of Computational Statistics*, vol.9, no.2, pp.93-103, 1996 (in Japanese).

[11] Z. Zhang, M. Namba, H. Kawabata and A. Kanagawa, Cellular neural networks and its application for abnormal detection, *Trans. on the Society of Instrument and Control Engineers*, vol.39, no.3, pp.209-217, 2003 (in Japanese).

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

[13] 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.

[14] K. Yokosawa, T. Nakaguchi, Y. Tanji and M. Tanaka, Cellular neural networks with output function having multiple constant regions, *IEEE Trans. Circuits & Systems I*, vol.50, no.7, pp.847-857, 2003.

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

[16] A. N. Michel and J. A. Farrell, Associative memories via artificial neural networks, *IEEE Control Systems Magazine*, vol.10, no.3, pp.6-17, 1990.

[17] Y. Akiyama, Three techniques for energy minimization on Hopfield-type neural networks, *IEICE Technical Report*, vol.NC90, no.40, pp.73-79, 1990 (in Japanese).

[18] R. Duda, P. Hart and D. Stork, *Pattern Classification*, 2nd Edition, John Wiley & Sons, New York, 2001.

[19] V. Franc and V. Hlavac, *Statistical Pattern Recognition Toolbox for Matlab*, Czech Technical University, http://cmp.felk.cvut.cz/cmp/software/stprtool/, 2004.

[20] Z. Zhang, R. Taniai, T. Akiduki, T. Imamura and T. Miyake, A new design method of multi-valued cellular neural networks for associative memory, *Proc. of the 4th International Conference on Innovative Computing, Information and Control*, Kaohsiung, Taiwan, pp.1-4, 2009.