A New Recurrent Neural-Network Architecture for Visual Pattern Recognition

Seong-Whan Lee, *Senior Member, IEEE,* and Hee-Heon Song, *Member, IEEE*

***Abstract*—In this paper, we propose a new type of recurrent neural-network architecture, in which each output unit is connected to itself and is also fully connected to other output units and all hidden units. The proposed recurrent neural network differs from Jordan's and Elman's recurrent neural networks with respect to function and architecture, because it has been originally extended from being a mere multilayer feedforward neural network, to improve discrimination and generalization powers. We also prove the convergence properties of learning algorithm in the proposed recurrent neural network, and analyze the performance of the proposed recurrent neural network by performing recognition experiments with the totally unconstrained handwritten numeric database of Concordia University, Montreal, Canada. Experimental results have confirmed that the proposed recurrent neural network improves discrimination and generalization powers in the recognition of visual patterns.**

***Index Terms*— Convergence properties, recurrent neural network, visual pattern recognition.**

I. Introduction

RECENTLY, a number of neural-network models have been implemented for pattern recognition [1], [2]. In particular, multilayer feedforward neural networks have shown their effectiveness in visual pattern recognition in a variety of styles and sizes [3]–[5]. However, these approaches can only provide partial solutions to real-world data handling, because they have shown insufficient learning capability with respect to similar patterns. In order to overcome this problem, it is needed that output results of feedforward neural networks be analyzed and reused in training phases.

In general, in the case of visual pattern recognition with multilayer feedforward neural networks, the hidden units are learned to maximize useful information from input patterns, and the output units are learned to discriminate information given from hidden units [6], [7]. Therefore, it seems reasonable to provide more information to the output units in order to improve discrimination powers in visual pattern recognition.

Recurrent neural networks offer a framework suitable for reusing network output values in training. Recently, researches applying recurrent neural networks to visual pattern recognition, such as handwritten character recognition, have been in progress vigorously, and some of them have shown promising results [8]–[10]. However, these approaches are mostly based on Jordan's and Elman's recurrent neural networks, originally proposed for dynamic pattern recognition. Therefore, they may be inefficient in visual pattern recognition.

In this paper, we propose a new type of recurrent neural-network architecture which is adequate for visual pattern recognition, such as handwritten character recognition. The proposed recurrent neural network differs from Jordan's and Elman's recurrent networks with respect to their functions and architectures, because it has been originally extended from the multilayer feedforward neural-network architecture, to improve discrimination and generalization powers. The proposed recurrent neural network consists of three-layers, in which each output unit is connected to itself, and is also fully connected with other output units and all hidden units.

We also prove the convergence properties of learning algorithm in the proposed recurrent neural network, and analyze the performance of the proposed recurrent neural network by performing recognition experiments with the totally unconstrained handwritten numeric database of Concordia University, Montreal, Canada. Experimental results confirm that the proposed recurrent neural network improves discrimination and generalization powers in visual pattern recognition.

The rest of this paper is organized as follows. Section II briefly reviews previous recurrent neural-network architectures. A new type of recurrent neural network is proposed and its convergence properties proven in Section III. Experimental results are presented, verifying the effectiveness of the proposed recurrent neural network, in Section IV, and concluding remarks are given in Section V.

II. Previous Recurrent Neural-Network Architectures

An early use of a recurrent network can be found in the work of Anderson *et al.* [11], [12]. These used a fully connected neural network called brain state in a box (BSB) to model psychological effects observed in probability learning. In this network, each unit, which has no self-connection, is fully connected to every other unit in the network.

The content addressable memory of Hopfield [13] can be viewed as a minimization of the energy function, where memories correspond to local minima in the energy spaces. Hopfield's initial model was a network of fully-interconnected processing units, whose outputs were computed using a linear threshold. Later, Hopfield developed a continuous version

Manuscript received November 26, 1994; revised January 8, 1996, August 12, 1996, and October 8, 1996. This work was supported by the Directed Basic Research Fund of Korea Science and Engineering Foundation under Grant 95-0100-06-01-3.

Seong-Whan Lee is with the Department of Computer Science and Engineering, Korea University, Anam-dong, Seongbuk-ku, Seoul 136-701, Korea.

Hee-Heon Song is with the AIN Switching Section, Electronics and Telecommunications Research Institute, Kajong-dong, Yusong-ku, Taejon 305-350, Korea.

Publisher Item Identifier S 1045-9227(97)01743-8.

1045–9227/97$10.00 © 1997 IEEE

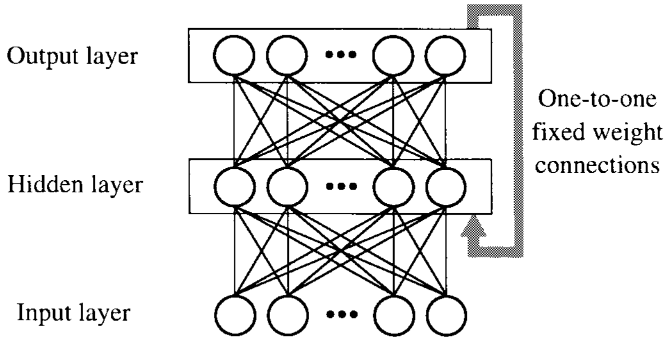


Fig. 1. Jordan's recurrent neural network.

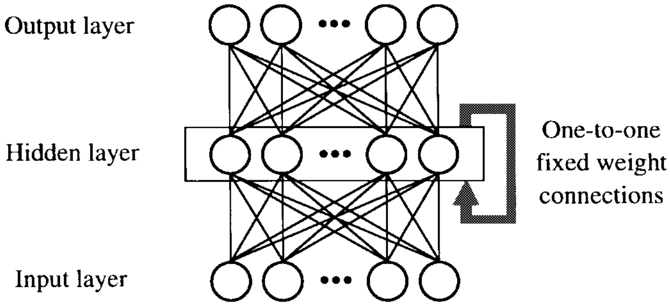


Fig. 2. Elman's recurrent neural network.

[14]. The new model uses a sigmoid transfer function as the activation function for the processing units, and units are updated continuously, using differential equations.

Jordan developed a network model capable of displaying temporal variations and temporal context dependence [15] (Fig. 1). Jordan's network played a useful role in motor control systems. This type of network model differs from the traditional view of motor control, in that it emphasizes that processors do not store and retrieve output vector sequences in linked lists or in any other abstract data structure. Rather, trajectories are computed during run-time as the result of a dynamic process. Those units calculating trajectories can be classified into four types: plan units, state units, hidden units, and output units. The state units possess inputs connected to themselves, and other units within the state layer deploy the standard connections to the output units. Elman developed a simple recurrent neural network [16] (Fig. 2). In this approach, rather than the outputs of the network being fed into the input units, the activation results of the hidden units are fed into the input units. While Jordan's recurrent neural network has appeared in a variety of control applications, Elman's recurrent neural network has been often applied to the problem of symbolic sequence prediction.

Learning methods for Jordan's and Elman's recurrent neural networks are extensions of the backpropagation learning method. A very general learning algorithm is that of Williams and Zipser [17]. Kuan *et al.* provided a rigorous convergence analysis from an extension of backpropagation for recurrent neural networks containing Jordan's and Elman's recurrent neural networks, as special cases [18]. The architectures specified by Jordan and Elman employ first-order connections between units. That is, the activation flowing from one unit to another is merely scaled by connection strength ($w_{ij}o_j$). However, the high-order recurrent neural network proposed by Rumelhart *et al.* [19], combines multiple incoming activations ($w_{ijk}o_jo_k$). One benefit of switching to a higher-order network is that more functions can be loaded into networks with fewer resources. Just as first-order connections underlie Jordan's and Elman's recurrent neural networks, multiple connections form the foundations of several recurrent networks, such as Pollack's sequential cascaded neural network [20] (Fig. 3) and the higher-order recurrent neural network of Giles *et al.* [21].

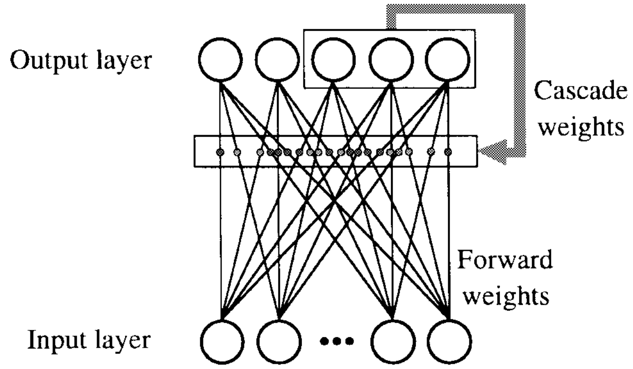


Fig. 3. Pollack's sequential cascaded neural network.

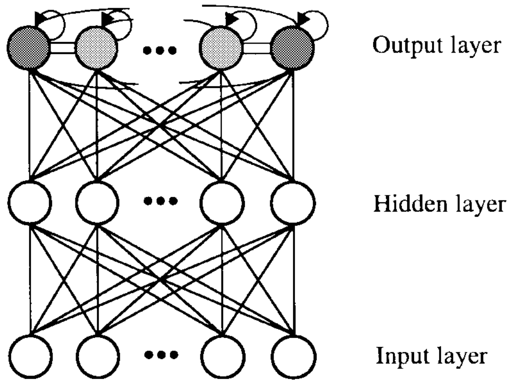


Fig. 4. Proposed recurrent neural-network architecture.

For further information on previous recurrent neural-network architectures, refer to the works of Kolen [12].

III. Proposed Recurrent Neural Network

In this section, we propose a new type of recurrent neural-network architecture, and prove its convergence properties.

A. Architecture of Proposed Recurrent Neural Network

The proposed recurrent neural-network architecture consists of three-layers, as shown in Fig. 4. The proposed recurrent neural network differs from Jordan's and Elman's recurrent neural networks with respect to their function and architecture, because it has been originally extended from the multilayer feedforward neural network to improve discrimination and generalization powers in visual pattern recognition.

Each hidden unit is fully connected to all input units, and each output unit is connected to itself, and also fully connected to other output units and all hidden units. Therefore, the output value of the ith output unit at cycle t is obtained as follows:

$$o_i^o(t) = f(\sum_{j=1}^{p} w_{ij} o_j^h(t) + r_i(t)) \tag{1}$$

$$r_i(t) = \sum_{k=1}^{q} z_{ik} o_k^o(t-1) \tag{2}$$

where $o_j^h(t)$ is the output value of the jth hidden unit at cycle t, w_{ij} is the weight between the jth hidden unit and the ith output unit, z_{ik} is the weight between the kth output unit and the ith output unit, $r_i(t)$ is the recurrent value from the output units at cycle $t-1$, p is the number of hidden units, and q is the number of output units. The activation function f is sigmoidal.

The output value of the ith hidden unit at cycle t is obtained as follows:

$$o_i^h(t) = f(\sum_{j=1}^{n} w_{ij} i_j(t)) \tag{3}$$

where $i_j(t)$ is the output value of the jth input unit at cycle t, w_{ij} is the weight between the jth input unit and ith hidden unit, and n is the number of input units. The input units feature the linear transfer function. The proposed recurrent neural network operates as follows. The activation values of the output units are initially set to zero. The input feature values are fed into the input units, which feature the linear transfer function; and the output values of the input units are forward propagated until the output units become active.

The weight correction of the feedforward neural network in the training phase is as follows:

$$\Delta w_{ij} = \epsilon \cdot (t_i - o_i^o) f_i'(\sum_{j=1}^{p} w_{ij} o_j^h(t)) \cdot o_j^h \tag{4}$$

where ϵ is the positive learning rate and t_i is the target value of output unit i. As shown in this equation, the weight correction is only based on the specific training pattern in a single cycle. Because the output units in a feedforward neural network are only activated by units in previous layers, the activation values of the previous cycle have an effect on the activation values of output units in the current cycle. In particular, in the case of training similar visual patterns, the output units in the previous cycle produce more ambiguous activation values. In this respect, it is required to minimize propagation of ambiguous activation values in the next cycle. On the basis of this property, the weight correction of the proposed recurrent neural network in the training phase is as follows:

$$\Delta w_{ij} = \epsilon \cdot (t_i - o_i^o) f_i'(\sum_{j=1}^{p} w_{ij} o_j^h(t) + \sum_{k=1}^{q} z_{ik} o_k^o(t-1)) \cdot o_j^h. \tag{5}$$

In this equation, the term $\sum_{k=1}^{q} z_{ik} o_k^o(t-1)$, which is the information needed to maximize discrimination powers, is added to (4). Training for the same training pattern is carried out based on these ambiguous activation values in the next cycle. That is, the activation values of each output unit r_i in (2) is used in the next cycle, in order to provide more discriminative information. As a result, the weights in the output units can be trained to discriminate between these ambiguous activation values of the previous cycle and the discrimination powers can be improved.

B. Convergence Properties of Learning Algorithm in Proposed Recurrent Neural Network

We now prove the convergence properties of the Williams–Zipser learning algorithm [17] in the proposed recurrent neural network. Our results follow from the results of Kuan and White [22] and Kuan *et al.* [18]. Kuan *et al.* provided rigorous convergence analysis of an extension of backpropagation for recurrent neural networks containing Jordan's and Elman's recurrent neural networks, as special cases.

We considered the same conditions and convergence results as those of the theorem in the work of Kuan *et al.* [18], because the stochastic process with respect to input sequences, the measurements of network error, the learning recursions, and the limit conditions of the weight changes of the proposed recurrent neural network are equivalent to those of Jordan's and Elman's networks, in spite of the difference in architecture. Thus, we describe only those assumptions of the theorem which are based on modified output functions and recurrent variables for the proposed recurrent neural network.

We now introduce some mathematical notations and a stochastic process [18], which are necessary for these assumptions.

Suppose that we observe realization of a sequence $\{Z_t\} = \{Z_t : t = 0, 1, \cdots\}$ of random vectors, where $Z_t = (Y_t, X_t^T)^T$ (with T denoting the transposition operator). We interpret Y_t as a target value at cycle t and X_t as a vector of those input variables influencing Y_t. X_t may contain the lagged values of Y_t (e.g., Y_{t-1}, Y_{t-2}), as well as the lagged values of other variables.

Let $X^t \equiv (X_0, \cdots, X_t)$ denote the history of the process X from cycle zero through cycle t and $f_t(X^t, \theta)$ denote the approximation function as θ ranges over the parameter space $\Theta \subset \mathbf{R}^s$, where s is the number of weights in the network.

On the basis of the stochastic process used to prove the convergence properties of the Williams–Zipser learning algorithm [17], [18], we begin by picking arbitrary initial weights $\hat{\theta}_0$, recurrent variables $\hat{R}_0$, and gradient matrix $\hat{\triangle}_0$. To update network weights, we compute network error and gradient as follows:

$$\hat{e}_0 = u(Z_0, \hat{R}_0, \hat{\theta}_0) \tag{6}$$

$$\nabla \hat{e}_0 = u_\theta(Z_0, \hat{R}_0, \hat{\theta}_0)^T + \hat{\triangle}_0 u_r(Z_0, \hat{R}_0, \hat{\theta}_0)^T. \tag{7}$$

Then, the weights in cycle 1 are calculated as follows:

$$\hat{\theta}_1 = \hat{\theta}_0 - \eta_0 \nabla \hat{e}_0 \cdot e_0. \tag{8}$$

The recurrent variables and gradient matrix are updated in cycle 1 to

$$\hat{R}_1 = \rho(Z_0, \hat{R}_0, \hat{\theta}_0) \tag{9}$$

and

$$\hat{\triangle}_t = \rho_\theta(Z_0, \hat{R}_0, \hat{\theta}_0)^T + \hat{\triangle}_0 \rho_r(Z_0, \hat{R}_0, \hat{\theta}_0)^T. \tag{10}$$

Now, network error and gradient are calculated as follows:

$$\hat{e}_1 = u(Z_1, \hat{R}_1, \hat{\theta}_1) \tag{11}$$

$$\nabla \hat{e}_1 = u_\theta(Z_1, \hat{R}_1, \hat{\theta}_1)^T + \hat{\triangle}_1 u_r(Z_1, \hat{R}_1, \hat{\theta}_1)^T. \tag{12}$$

Then, the weights in cycle 2 are obtained as follows:

$$\hat{\theta}_2 = \hat{\theta}_1 - \eta_1 \nabla \hat{e}_1 \cdot \hat{e}_1. \tag{13}$$

At cycle t, we have targets and inputs Z_t, recurrent variables $\hat{R}_t$, weights $\hat{\theta}_t$, and gradient matrix $\hat{\triangle}_t$, permitting us to compute

$$\begin{aligned} \hat{e}_t &= u(Z_t, \hat{R}_t, \hat{\theta}_t), \\ \nabla \hat{e}_t &= u_\theta(Z_t, \hat{R}_t, \hat{\theta}_t)^T + \hat{\triangle}_t u_r(Z_t, \hat{R}_t, \hat{\theta}_t)^T \\ \hat{\theta}_{t+1} &= \hat{\theta}_t - \eta_t \nabla \hat{e}_t \cdot \hat{e}_t, \\ \hat{R}_{t+1} &= \rho(Z_t, \hat{R}_t, \hat{\theta}_t) \end{aligned} \tag{14}$$

and

$$\hat{\triangle}_{t+1} = \rho_\theta(Z_t, \hat{R}_t, \hat{\theta}_t)^T + \hat{\triangle}_t \rho_r(Z_t, \hat{R}_t, \hat{\theta}_t)^T. \tag{15}$$

A potential difficulty is that nothing prevents $\hat{\theta}_t \to \infty$. To avoid this, we employ a projection operator $\pi : \mathbf{R}^s \to \Theta$, where Θ is a compact subset of $\mathbf{R}^s$. The projected process $\{\pi(\hat{\theta}_t)\}$ is bounded, and $\hat{\theta}_t$ is defined as $\hat{\theta}_t = \pi(\hat{\theta}_t)$ whenever $\hat{\theta}_t \in \Theta.\{\theta_t\}$ also denotes the projected process, for notational convenience.

In order to describe our assumptions of the theorem, we introduce the notion of near epoch dependent (NED) on an underlying mixing process [18].

Let $\{V_t\}$ be a stochastic process in a probability space $(\Omega, \mathcal{F}, P)$, and define the mixing coefficients

$$\phi_m \equiv \sup_t \sup_{\{F \in \mathcal{F}_{-\infty}^t, G \in \mathcal{F}_{t+m}^\infty : P(F) > 0\}} |P(G|F) - P(G)| \tag{16}$$

$$\alpha_m \equiv \sup_t \sup_{\{F \in \mathcal{F}_{-\infty}^t, G \in \mathcal{F}_{t+m}^\infty\}} |P(G \cap F) - P(G)P(F)| \tag{17}$$

where $\mathcal{F}_\tau^t \equiv \sigma(V_\tau, \cdots., V_t)$, and the σ-field is generated by $\mathcal{F}_\tau^t$: $V_\tau, \cdots, V_t$. When $\phi_m \to 0$ or $\alpha_m \to 0$ as $m \to \infty$, $\{V_t\}$ is termed ϕ-mixing or α-mixing [18]. When $\phi_m = O(m^\lambda)$ for some $\lambda < -a$, $\{V_t\}$ is termed ϕ-mixing of size $-a$, and similarly for α_m.

Let $||Z_t||_2 \equiv (E|Z_t|^2)^{1/2}$ and $E_{t-m}^{t+m}(Z_t) \equiv E(Z_t|\mathcal{F}_{t-m}^{t+m})$, and let $L_2(P)$ denote that class of random variables having $||Z_t||_2 < \infty$. The dependence of $\{Z_t\}$ on an underlying process $\{V_t\}$ is expressed as follows [18].

Definition 1: Let $\{Z_t\}$ be a sequence of random variables belonging to $L_2(P)$, and let $\{V_t\}$ be a stochastic process on $(\Omega, \mathcal{F}, P)$. Then $\{Z_t\}$ is NED on $\{V_t\}$ of size $-a$ if $v_m \equiv \sup_t ||Z_t - E_{t-m}^{t+m}(Z_t)||_2$ is of size $-a$.

The data generating process is described as follows.

Assumption A.1: $(\Omega, \mathcal{F}, P)$ is a complete probability space, which is defined by the sequence of $\mathcal{F}$-measurable functions $\{Z_t : \Omega \to \mathbf{R}^{v+1}, t = 0, 1, 2, \cdots\}$ with $\sup_{t \geq 0} |Z_t| \leq \epsilon^{-1} < \infty$, where $v \in \mathbf{N}$ is the size of input vector. $\{Z_t\}$ is NED on $\{V_t\}$ of size -$\frac{1}{2}$, where $\{V_t, t = 0, \pm 1, \pm 2, \cdots\}$ is a mixing process on $(\Omega, \mathcal{F}, P)$ with ϕ_m of size $-\frac{1}{2}$, or α_m of size -1. For each $t = 0, 1, \cdots, Z_t$ is measurable $\mathcal{F}^t \equiv \sigma(\cdots, V_{t-1}, V_t)$.

The following condition restricts the network error function.

Assumption A.2: Network output is given by

$$o = F\left(\alpha + \sum_{i=1}^q (\beta_i G(x^T \gamma_i) + r^T \delta_i)\right) \tag{18}$$

and network error is given by

$$u(z, r, \theta) = y - F\left(\alpha + \sum_{i=1}^q (\beta_i G(x^T \gamma_i) + r^T \delta_i)\right). \tag{19}$$

Then, network recurrence is obtained as follows:

$$\rho(z, r, \theta) = F\left(\alpha + \sum_{i=1}^q (\beta_i G(x^T \gamma_i) + r^T \delta_i)\right). \tag{20}$$

The mean value theorem for such functions ensures

$$\begin{aligned} &|\rho(z, r_1, \theta) - \rho(z, r_2, \theta)| \\ &\leq \left(\sup_{z \in K_z, r \in K_r, \theta \in \Theta} |\rho_r(z, r, \theta)|\right) |r_1 - r_2|. \end{aligned} \tag{21}$$

The following condition restricts network recurrence.

Assumption A.3: Network recurrence is determined by (20). Let $c_F = sup_{b \in K_F} |F'(b)|$. Then, Θ is such that $\sum_{i=1}^q |\delta_i| \leq c_F^{-1}(1 - \epsilon)$ for some $\epsilon > 0$.

Because the Jacobian matrix of ρ with respect to recurrent variables is calculated as follows:

$$\begin{aligned} \rho_r(z, r, \theta) = F'(\alpha + \sum_{i=1}^q \beta_i G(x^T \gamma_i) \\ + \sum_{i=1}^q r^T \delta_i) \cdot \sum_{i=1}^q \delta_i \end{aligned} \tag{22}$$

network recurrence restriction is given by

$$\begin{aligned} |\rho_r(z, r, \theta)| \leq |F'(\alpha + \sum_{i=1}^q \beta_i G(x^T \gamma_i) \\ + \sum_{i=1}^q r^T \delta_i)| \sum_{i=1}^q |\delta_i| \\ \leq c_F \sum_{i=1}^q |\delta_i|. \end{aligned} \tag{23}$$

Now, the learning recursions are formally described as follows.

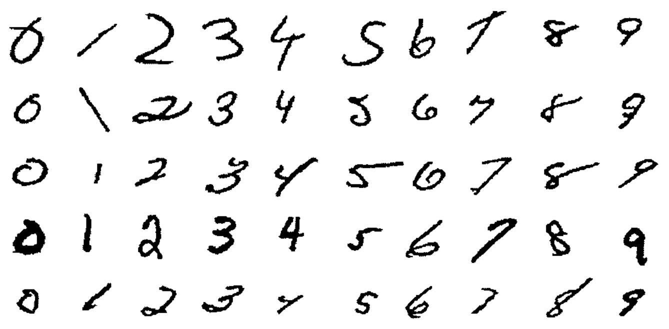


Fig. 5. Representative samples from totally unconstrained handwritten numeric database.

Assumption A.4: 1) Let $K_\triangle$ be a compact subset of $\mathbf{R}^{s \times p}$, where p is the size of recurrent variable, and let $\hat{R}_0 \in K_r, \hat{\triangle}_0 \in K_\triangle$, and $\hat{\theta}_0$ be chosen arbitrarily and independently of $\{Z_t\}$. For $t = 0, 1, 2, \cdots$, define

$$\hat{e}_t = u(Z_t, \hat{R}_t, \hat{\theta}_t), \tag{24}$$

$$\nabla \hat{e}_t = u_\theta(Z_t, \hat{R}_t, \hat{\theta}_t)^T + \hat{\triangle}_t u_r(Z_t, \hat{R}_t, \hat{\theta}_t)^T \tag{25}$$

$$\hat{\theta}_{t+1} = \pi[\hat{\theta}_t - \eta_t \nabla \hat{e}_t \hat{e}_t] \tag{26}$$

$$\hat{R}_{t+1} = \rho(Z_t, \hat{R}_t, \hat{\theta}_t) \tag{27}$$

and

$$\hat{\triangle}_{t+1} = \rho_\theta(Z_t, R_t, \hat{\theta}_t)^T + \hat{\triangle}_t \rho_r(Z_t, \hat{R}_t, \hat{\theta}_t)^T \tag{28}$$

where $\pi : \mathbf{R}^s \to \Theta$ is a projection operator restricting $\{\hat{\theta}_t\}$ to the compact set Θ.

2) $\{\eta_t\}$ is a sequence of positive real numbers, such that $\sum_{t=0}^{\infty} \eta_t^2 < \infty$ and $\sum_{t=0}^{\infty} \eta_t = \infty$.

One more condition is required to state Kuan and White's convergence results [22]; it guarantees the existence of the limit of $E(\nabla e_t(\theta) \cdot e_t(\theta))$. We define the function h as

$$h(\lambda, \theta) = -[u_\theta(z, r, \theta)^T + \triangle u_r(z, r, \theta)^T] u(z, r, \theta) \tag{29}$$

where $\lambda \equiv (z^T, r^T, \mathrm{vec}^T \triangle)^T$. We also define $\lambda_t(\theta)$ as $\lambda_t(\theta) = (\lambda_t^z(\theta)^T, \lambda_t^r(\theta)^T, \lambda_t^\triangle(\theta)^T)^T$, where $\lambda_t^z(\theta) \equiv Z_t$, $\lambda_t^r(\theta) \equiv l_t(Z^{t-1}, \theta)$, and $\lambda_t^\triangle(\theta) \equiv \mathrm{vec} \nabla l_t(Z^{t-1}, \theta)$.

Our final condition is given as follows.

Assumption A.5: For each $\theta \in \Theta, \bar{h}(\theta) \equiv \lim_{t \to \infty} E (h(\lambda_t(\theta), \theta))$ exists.

Kushner and Clark's results [23] establish certain properties of piecewise linear interpolations of $\{\hat{\theta}_t\}$ with interpolation intervals $\{\eta_t\}$. Define $\tau_t \equiv \sum_{i=0}^{t-1} \eta_i$, $t \geq 1$, $\tau_0 \equiv 0$. The interpolated process is defined as

$$\begin{aligned} \tilde{\theta}_0(\tau) =& \eta_t^{-1}(\tau_{t+1} - \tau)\hat{\theta}_t \\ &+ \eta_t^{-1}(\tau - \tau_t)\hat{\theta}_{t+1}, \tau \in [\tau_t, \tau_{t+1}] \end{aligned} \tag{30}$$

and its leftward shifts are defined as

$$\tilde{\theta}_t(\tau) = \begin{cases} \tilde{\theta}_0(\tau_t + \tau) & \tau \geq -\tau_t, \\ \hat{\theta}_\tau & \tau < -\tau_t, \quad t = 0, 1, 2, \cdots. \end{cases} \tag{31}$$

We thus have a sequence $\{\tilde{\theta}_t(\cdot)\}$ of continuous function on $(-\infty, \infty)$. In stating this result, we write $\hat{\theta}_t \to \Theta^*$ as $t \to \infty$ if $\inf_{\theta \in \Theta^*} |\hat{\theta}_t - \theta| \to 0$ as $t \to \infty$. Furthermore, for a continuous vector field $v(\cdot)$ on Θ, define the vector field $\bar{\pi}[v(\cdot)]$ as

$$\bar{\pi}[v(\theta)] = \lim_{\delta \to 0} [\pi(\theta + \delta v(\theta)) - \theta]/\delta, \quad \theta \in \Theta \tag{32}$$

when the limit is unique. When θ is in Θ, but not on its boundaries, sufficiently small $\theta + \delta v(\theta)$ is in Θ for δ, so that $\bar{\pi}[v(\theta)] = v(\theta)$.

The desired convergence of the proposed recurrent neural network now follows immediately.

Theorem 1: Suppose that Assumptions A.1–A.5 hold. Then,

1) There exists a P-null set Ω_0 such that for $\omega \notin \Omega_0$, $\{\bar{\theta}_t(\cdot)\}$ is bounded and equicontinuous at bounded intervals, and $\{\tilde{\theta}_t(\cdot)\}$ has a convergent subsequence, whose limit $\bar{\theta}_t(\cdot)$ satisfies the limit expectation $\dot{\theta} = \bar{\pi}[\bar{h}(\theta)]$.
Let Θ^* be a set of locally asymptotically stable (in the sense of Liapunov) equilibria in Θ for this limit expectation with a domain of attraction $d(\Theta^*) \subseteq \mathbf{R}^s$.
2) If $\Theta \subseteq d(\Theta^*)$, then $\hat{\theta}_t \to \Theta^*$ as $t \to \infty$, with probability one.
3) If Θ is not contained in $d(\Theta^*)$, but for each $\omega \notin \Omega_0, \hat{\theta}_t(\omega)$ enters a compact subset of $d(\Theta^*)$ infinitely often, then $\hat{\theta}_t \to \Theta^*$ as $t \to \infty$, with probability one.
4) Given the conditions in (c), if Θ^* contains only a finite number of points, then there exists a measurable mapping $\theta^* : \Omega \times \Theta \times K_r \times K_\triangle \to \Theta^*$, such that $\hat{\theta}_t - \theta^*(\cdot, \hat{\theta}_0, \hat{R}_0, \hat{\triangle}_0) \to 0$ as $t \to \infty$, with probability one.

Proof: The proof follows immediately from the proof of the Theorem in the work of Kuan and White [22], itself derived from fundamental results of Kushner and Clark [23].

IV. Experimental Results

In this section, we present experimental results and analyze the performance of the proposed recurrent neural network. In order to verify the performance of the proposed recurrent neural network, recognition experiments using the totally unconstrained handwritten numeric database of Concordia University were performed [2].

A. Database

The handwritten numeric database of Concordia University consists of a totally unconstrained 6000 numerals originally collected from dead letter envelopes by the U.S. Postal Service from different locations in the United States. The numerals of this database were digitized in bilevel on a 64 × 224 grid of 0.153 mm square elements, giving a resolution of approximately 166 PPI [24]. Training used 4000 numerals and testing used 2000.

Fig. 5 shows some representative samples taken from the numeric database used in this paper. Many different writing styles are apparent, as well as different numeral sizes and stroke widths.

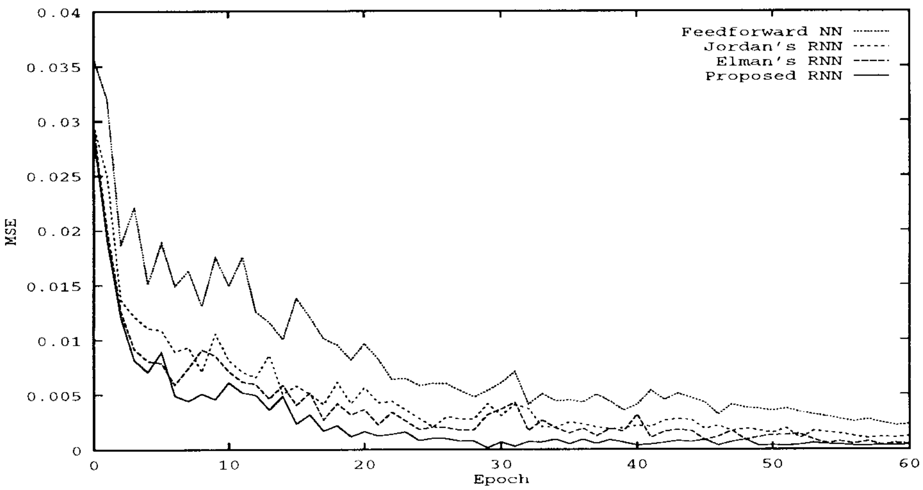


Fig. 6. Learning curves for the four different neural networks.

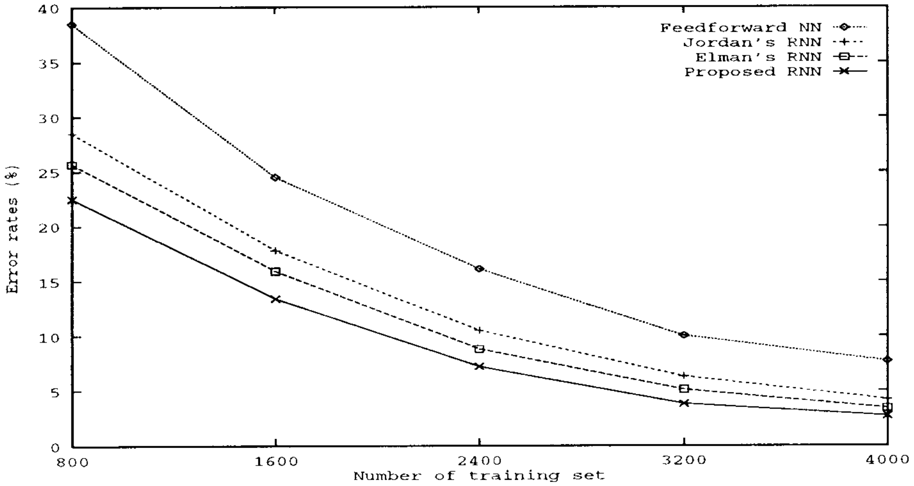


Fig. 7. Error rate versus training set size for the four different neural networks.

TABLE I
ERROR RATES ON THE TRAINING SET

Method	Feedforward NN	Jordan's RNN	Elman's RNN	Proposed RNN
Error rate	0.850%	0.650%	0.625%	0.575%

TABLE II
ERROR RATES ON THE TESTING SET

Method	Feedforward NN	Jordan's RNN	Elman's RNN	Proposed RNN
Error rate	3.7%	3.1%	2.9%	2.7%

B. Recognition Experiments

In order to demonstrate the performance of the proposed recurrent neural network, four kinds of neural-network classifiers have been considered. These are as follows:

Feedforward NN: Simple three-layer feedforward neural network;
Jordan's RNN: Jordan's recurrent neural network;
Elman's RNN: Elman's recurrent neural network;
Proposed RNN: Proposed recurrent neural network.

The input pattern has been size-normalized to 16 × 16, and then, in order to train the spatial dependencies in a character image, directional feature vectors for horizontal, vertical, right-diagonal, and left-diagonal directions are calculated from a size-normalized image by using Kirsch masks [25]. Additionally, each 16 × 16 directional feature vector is compressed to 4 × 4 features. Furthermore, in order to consider the global characteristics of input image, we compressed the 16 × 16 normalized input image into a 4 × 4 image, and used this compressed image as a global feature. As a result, final features consist of 5 × 4 × 4 features; 4 × 4 × 4 local features, and 1 × 4 × 4 global features. These features have been used as input values to neural networks, in which the input layer and hidden layer consist of 80 units and the output layer consists

TABLE III
REDUCTIONS IN ERROR RATE FOR EACH RECURRENT NEURAL NETWORK COMPARED TO THE SIMPLE FEEDFORWARD NEURAL NETWORK

Method	Jordan's RNN	Elman's RNN	Proposed RNN
Reduction in error rate	18.0%	20.0%	24.1%

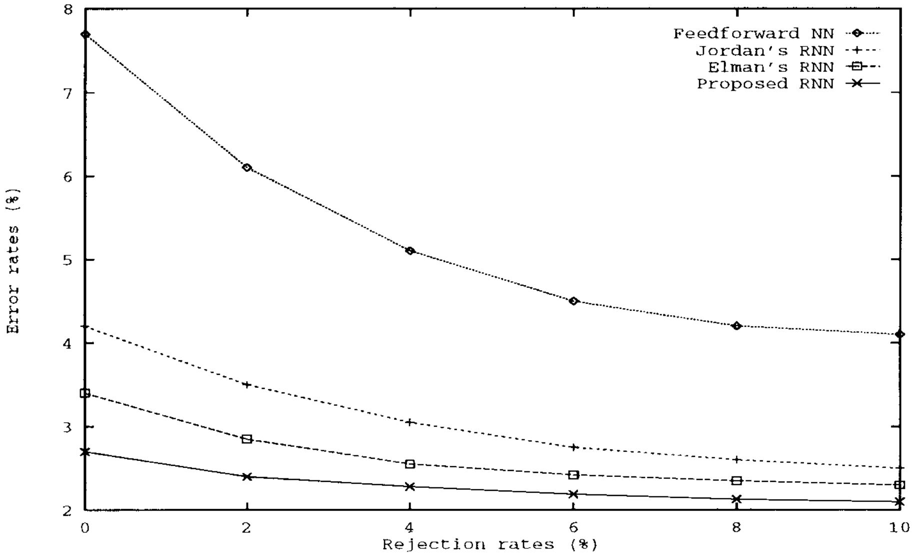


Fig. 8. Error rate versus rejection rate for the four different neural networks.

TABLE IV
CONFUSION MATRIX FOR THE FEEDFORWARD NN

Class	0	1	2	3	4	5	6	7	8	9	Substituted	Recognized
0	192					1	2		5		4.0%	96.0%
1		195	2			1		2			2.5%	97.5%
2			191	5				1	3		4.5%	95.5%
3			3	192					4	1	4.0%	96.0%
4		2	1		193			3		1	3.5%	96.5%
5	1	1		4		193	1				3.5%	96.5%
6	2	1			1	1	195				2.5%	97.5%
7			2	1	3			192	1	1	4.0%	96.0%
8	1	5		3					191		4.5%	95.5%
9	1		1	1				3	2	192	4.0%	96.0%
								Average			3.7%	96.3%

of 10 units. In order to activate the output units of the recurrent neural networks the feature values for a character are accepted twice. This has the effect of preserving spatial dependencies for effective discrimination of similar numerals.

We have used the backpropagation learning algorithm [19] for simple feedforward neural network and the Williams–Zipser algorithm [17] for recurrent neural networks. Because training sequence size is two, namely, t_0 and t_1, per character in training the recurrent neural network, the modification of weights occurs only in cycle t_1.

C. Experimental Results and Analysis

Fig. 6 shows learning curves for the four different neural networks. As shown in Fig. 6, the proposed recurrent neural network greatly improved convergence speed.

We also have examined error rate versus training set size, in order to verify the generalization powers of the proposed recurrent neural network. The number of training sets has been varied from 800 to 4000, fixing the number of testing set at 2000. Fig. 7 shows the changes in error rate without rejection on testing sets as the size of training set increases.

As can be observed from Fig. 7, the proposed recurrent neural network and Elman's recurrent neural network showed superior generalization powers to the other neural networks.

Tables I and II show error rate without rejection on the training sets and testing sets, respectively.

As indicated in Table II, in the case of simple feedforward neural network, the error rate is 3.7%. In the cases of using the other three types of recurrent neural network, error rates are 3.1%, 2.9%, and 2.7%, respectively. These results confirm that the proposed recurrent neural network has very good discrimination powers when compared to the other recurrent neural networks.

Table III shows the reduction of error rate for each recurrent neural network compared to the simple feedforward neural network in Table II.

TABLE V
CONFUSION MATRIX FOR JORDAN'S RNN

Class	0	1	2	3	4	5	6	7	8	9	Substituted	Recognized
0	193					1	2		4		3.5%	96.5%
1		196	2			1		1			2.0%	98.0%
2			193	4				1	2		3.5%	96.5%
3			2	193					4	1	3.5%	96.5%
4		2	1		194			2		1	3.0%	97.0%
5	1	1		3		194	1				3.0%	97.0%
6	1	1			1	1	196				2.0%	98.0%
7			2	1	2			193	1	1	3.5%	96.5%
8	1	4		3					193		3.5%	96.5%
9	1		1	1				3	1	193	3.5%	96.5%
								Average			3.1%	96.9%

TABLE VI
CONFUSION MATRIX FOR ELMAN'S RNN

Class	0	1	2	3	4	5	6	7	8	9	Substituted	Recognized
0	194					1	2		3		3.0%	97.0%
1		196	2			1		1			2.0%	98.0%
2			194	3				1	2		3.0%	97.0%
3			2	194					3	1	3.0%	97.0%
4		2	1		194			2		1	3.0%	97.0%
5	1	1		3		194	1				3.0%	97.0%
6	1	1			1	1	196				2.0%	98.0%
7			2	1	2			193	1	1	3.5%	96.5%
8	1	3		3					194		3.0%	97.0%
9	1		1	1				3	1	193	3.5%	96.5%
								Average			2.9%	97.1%

TABLE VII
CONFUSION MATRIX FOR THE PROPOSED RNN

Class	0	1	2	3	4	5	6	7	8	9	Substituted	Recognized
0	195					1	2		2		2.5%	97.5%
1		196	2			1		1			2.0%	98.0%
2			195	2				1	2		2.5%	97.5%
3			2	194					3	1	3.0%	97.0%
4		2	1		194			2		1	3.0%	97.0%
5	1	1		2		195	1				2.5%	97.5%
6	1	1			1	1	196				2.0%	98.0%
7			2		2			194	1	1	3.0%	97.0%
8	1	3		3					194		3.0%	97.0%
9	1		1	1				3	1	193	3.5%	96.5%
								Average			2.7%	97.3%

As shown in Table III, the use of the proposed recurrent neural network brings about 24.1% reduction in error rate compared to the simple feedforward neural network. However, for the cases of using Jordan's and Elman's recurrent neural networks, the reductions in error rate are 20.0% and 18.0%, respectively. The 24.1% reduction in error rate is of statistical significance in unconstrained handwritten numeric recognition.

We have also analyzed error rate versus rejection rate to evaluate discrimination performance of the four different neural networks on testing sets. The results are described in Fig. 8.

Table IV through Table VII show confusion matrices without rejection for each neural network. In these tables, we can easily see the discrimination performance of each architecture. As shown in Table VII, the proposed recurrent neural network can classify similar numerals efficiently.

V. CONCLUDING REMARKS

In this paper, we proposed a new type of recurrent neural network architecture, in which each output unit is connected to itself and is also fully connected to other output units and all hidden units. The proposed recurrent neural network differs from Jordan's and Elman's recurrent networks with respect to function and architecture, because it was originally extended from the multilayer feedforward neural network to improve discrimination and generalization powers.

In general, in cases of visual pattern recognition using multilayer feedforward neural networks, the hidden units are learned to maximize the useful information from input patterns and the output units are learned to discriminate the information given from the hidden layers. Therefore, providing more information to output units in order to improve discrimination powers seems a natural step.

In this paper, we also proved the convergence properties of learning algorithm in the proposed recurrent neural network and analyzed the performance of the proposed architecture by performing recognition experiments with the totally unconstrained handwritten numeric database of Concordia University. Experimental results confirmed that the proposed recurrent neural network improves discrimination and generalization powers in visual pattern recognition.

Further investigation should be made, however, to design an optimal recurrent neural-network architecture which offers good generalization powers and to implement it on parallel hardware.

VI. Acknowledgment

The authors wish to thank the anonymous reviewers for their helpful comments leading to improvements of the earlier draft of this paper.

References

[1] R. P. Lippmann, "An introduction to computing with neural nets," *IEEE Acoust., Speech, Signal Processing Mag.*, vol. 4, pp. 4–22, Apr. 1987.

[2] C. Y. Suen, C. Nadal, R. Legault, T. A. Mai, and L. Lam, "Computer recognition of unconstrained handwritten numerals," *Proc. IEEE*, vol. 80, no. 7, pp. 1162–1180, 1992.

[3] Y. Le Cun *et al.*, "Constrained neural-network for unconstrained handwritten digit recognition," in *Proc. 1st Int. Wkshp. Frontiers Handwriting Recognition*, Montreal, Canada, 1990, pp. 145–154.

[4] A. Krzyzak, W. Dai, and C. Y. Suen, "Unconstrained handwritten character classification using modified backpropagation model," in *Proc. 1st Int. Wkshp. Frontiers Handwriting Recognition*, Montreal, Canada, 1990, pp. 155–166.

[5] S. Knerr, L. Personnaz, and G. Dreyfus, "Handwritten digit recognition by neural networks with single-layer training," *IEEE Trans. Neural Networks*, vol. 3, pp. 962–968, 1992.

[6] A. Wieland and R. Leighton, "Geometric analysis of neural-network capabilities," in *Proc. IEEE Int. Conf. Neural Networks*, vol. III, San Diego, CA, 1987, pp. 385–392.

[7] I. K. Sethi, "Entropy nets: From decision trees to neural networks," *Proc. IEEE*, vol. 78, no. 10, pp. 1605–1613, 1990.

[8] A. W. Senior, "Off-line handwriting recognition: A review and experiments," Engineering Dep., Cambridge Univ., Cambridge, U.K., Tech. Rep. TR105, 1992.

[9] S. S. Kim and S. I. Chien, "Improving generalization capability and noise cancelling with the partially connected recurrent neural network," in *Proc. Int. Joint Conf. Neural Networks*, Beijing, China, 1992, pp. 166–171.

[10] R. Urbanczik, "A recurrent neural network inverting a deformable template model of handwritten digits," in *Proc. Int. Conf. Artificial Neural Networks*, Sorrento, Italy, 1994, pp. 961–964.

[11] J. A. Anderson *et al.*, "Distinctive features, categorical perception, and probability learning: Some applications of a neural model," *Psych. Rev.*, vol. 84, pp. 413–451, 1977.

[12] J. F. Kolen, "Exploring the computational capabilities of recurrent neural networks," Ph.D. dissertation, Ohio State Univ., Columbus, 1994.

[13] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," in *Proc. Nat. Academy Sci. USA*, vol. 79, 1982, pp. 2554–2558.

[14] ——, "Neurons with graded response have collective computational properties like those of two-state neurons," in *Proc. Nat. Academy Sci. USA*, vol. 81, pp. 3088–3092, 1984.

[15] M. Jordan, "Serial order: A parallel distributed processing approach," Univ. California San Diego, Inst. Cognitive Sci., ICS Rep. 8604, 1986.

[16] J. L. Elman, "Finding structure in time," *Cognitive Sci.*, vol. 14, pp. 179–211, 1990.

[17] R. J. Williams and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks," *Neural Computa.*, vol. 1, pp. 270–280, 1989.

[18] C.-M. Kuan, K. Hornik, and H. White, "A convergence result for learning in recurrent neural networks," *Neural Computa.*, vol. 6, pp. 420–440, 1994.

[19] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," in *Parallel Distributed Processing*, vol. 1. Cambridge, MA: MIT Press, 1986, pp. 318–362.

[20] J. B. Pollack, "The induction of dynamical recognizers," *Machine Learning*, vol. 7, pp. 227–252, 1991.

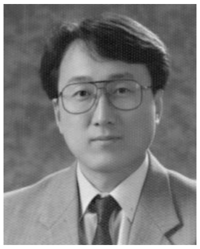
[21] C. L. Giles, G. Z. Sun, H. H. Chen, Y. C. Lee, and D. Chen, "Higher-order recurrent networks and grammatical inference," *Advances in Neural Information Processing Systems 2*, pp. 380–387, 1990.

[22] C.-M. Kuan and H. White, "Adaptive learning with nonlinear dynamics driven by dependent processes," Univ. California San Diego, Dep. Economics Discussion Paper, 1992.

[23] H. J. Kushner and D. S. Clark, *Stochastic Approximation Methods for Constrained and Unconstrained Systems*. New York: Springer-Verlag, 1978.

[24] C. Y. Suen, C. Nadal, T. A. Mai, R. Legault, and L. Lam, "Recognition of handwritten numerals based on the concept of multiple experts," in *Proc. 1st Int. Wkshp. Frontiers Handwriting Recognition*, Montreal, Canada, 1990, pp. 131–144.

[25] S.-W. Lee, "Off-line recognition of totally unconstrained handwritten numerals using multilayer cluster neural network," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 18, pp. 648–652, 1996.



Seong-Whan Lee (S'84–M'91–SM'96) received the B.S. degree in computer science and statistics from Seoul National University, Seoul, Korea, in 1984 and the M.S. and Ph.D. degrees in computer science from the Korea Advanced Institute of Science and Technology in 1986 and 1989, respectively.

In 1987, he worked as a Visiting Researcher at the Pattern Recognition Division, Delft University of Technology, Delft, The Netherlands. He was a Visiting Scientist at the Centre for Pattern Recognition and Machine Intelligence, Concordia University, Montreal, Canada, during the winter of 1989 and the summer of 1990. From 1989 to 1994, he was an Assistant Professor in the Department of Computer Science, Chungbuk National University, Cheongju, Korea. In March 1995, he joined the faculty of the Department of Computer Science and Engineering, Korea University, Seoul, Korea, as an Associate Professor. He has more than 100 publications on pattern recognition and neural networks in international journals and conference proceedings, holds two Korean patents, and has authored two Korean books: *Theory and Practice of Character Recognition*, (Hongneung Press, 1993) and *Principles of Pattern Recognition*, (Hongneung Press, 1994). His research interests include document image analysis, large-set character recognition, face recognition, hidden Markov models, and neural networks.

Dr. Lee was the winner of the Annual Best Paper Award of the Korea Information Science Society in 1986. He obtained the First Outstanding Young Researcher Award at the Second International Conference on Document Analysis and Recognition in 1993, and the First Distinguished Research Professor Award from Chungbuk National University in 1994. In 1996, he also obtained the Outstanding Research Award from the Korea Information Science Society. He is the Guest Editor of the *Pattern Recognition Journal*, and the Associate Editor of the *Pattern Recognition Journal*, *International Journal of Pattern Recognition and Arificial Intelligence* and *International Journal of Computer Processing of Oriental Languages*. He is the Program Chairman of the 17th International Conference on Computer Processing of Oriental Languages and the Sixth International Workshop on Frontiers in Handwriting Recognition, and the Program Cochairman of the Fifth International Conference on Document Analysis and Recognition and the Second INternational Conference on Multimodal Interface. He served on the program committees of several well-known international conferences. He is a Senior Member of the IEEE Computer Society and a Member of the Korea Information Science Society, the Pattern Recognition Society, the International Neural Network Society, and the Oriental Language Computer Society.

Hee-Heon Song (M’96) received the B.S. degree in computer science from Dongkook University, Seoul, Korea, in 1986, the M.S. degree in computer science from Chungnam National University, Taejon, Korea, in 1992, and the Ph.D. degree in computer science from Chungbuk National University, Cheongju, Korea, in 1995, respectively.

Since 1986, he has been working as a Senior Research Engineer at Electronics and Telecommunications Research Institute, Taejon, Korea. His research interests include neural networks, pattern recognition, and speech recognition.

Dr. Song is a member of the IEEE Computer Society and the Korea Information Science Society.