A New Recurrent Neural Network Architecture for Pattern Recognition and Its Convergence Results

Seong-Whan Lee, Young-Joon Kim, and Hee-Heon Song

Abstract

In this paper, we propose a new type of recurrent neural network architecture in which each output unit is connected with itself and fully-connected with other output units and all hidden units. The proposed recurrent neural network differs from Jordan's and Elman's recurrent networks in view of functions and architectures because it was originally extended from the multilayer feedforward neural network for improving the discrimination and generalization power. We also prove the convergence property of learning algorithm of the proposed recurrent neural network and analyze the performance of the proposed recurrent neural network by performing recognition experiments with the totally unconstrained handwritten numeral database of Concordia University of Canada. Experimental results confirmed that the proposed recurrent neural network improves the discrimination and generalization power in recognizing spatial patterns.

I. Introduction

Recently, a number of neural network models have been implemented for pattern recognition [1][2]. Especially, multilayer feedforward neural networks have shown their effectiveness in recognizing spatial patterns of various styles and sizes [3][4][5]. However, these approaches can only provide a partial solution to real-world data because they have shown insufficient learning capability for the similar patterns. In order to overcome this problem, it is needed that the output results of the feedforward neural networks are analyzed and reused in training the input patterns.

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

A recurrent neural network offers a framework suitable for reusing the output values of the network in training. Recently, researches applying recurrent neural network to spatial pattern recognition such as handwritten character recognition are in progress and some of them have shown promising results [8][9][10]. However, these approaches are mostly based on Jordan's and Elman's recurrent neural networks which were proposed for dynamic patterns. Therefore, they may be inefficient for spatial pattern recognition.

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

We also prove the convergence property 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

Manuscript received July, 18, 1995; accepted October 16, 1995.

S. W. Lee is with Department of Computer Science, Korea University, Seoul, Korea.

Y. J. Kim is with BIT Computer Co., Ltd., Seoul, Korea.

H. H. Song is with Electronics and Telecommunications Research Institute.

This research was supported by Directed Basic Research Fund of Korea Science and Engineering Foundation under grant 95-0100-06-01-3.

totally unconstrained handwritten numeral database of Concordia University of Canada. Experimental results confirmed that the proposed recurrent neural network improves the discrimination and generalization power in recognizing spatial patterns.

The rest of this paper is organized as follows. Section II briefly reviews the previous recurrent neural network architectures. A new type of recurrent neural network is proposed and its convergence property is proven in Section III. The experimental results are presented for verifying the effectiveness of the proposed recurrent neural network in Section IV and the 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]. They used the fully-connected neural network called Brain State in a Box(BSB) to model psychological effects seen in probability learning. In this network, each unit which has no self connection is fully-connected with every other units in the network.

Content addressable memory of Hopfield [13] can be viewed as the minimization of an energy function where memories correspond to local minima in the energy spaces. Hopfield's original model was a network of fully-interconnected processing units whose output was computed using a linear threshold. Later, Hopfield developed a continuous version [14]. The new model uses a sigmoid transfer function as the activation function for the processing units and units are updated continuously according to the differential equation.

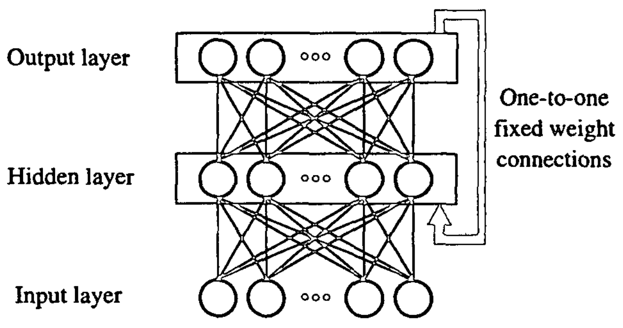


Fig. 1. Jordan's recurrent neural network.

Jordan developed a network model capable of displaying temporal variation and temporal context dependence [15] (Fig. 1). The Jordan's network served as useful role in motor control system. This type of network model differs from traditional views of motor control in that it emphasizes that the processor does not store and retrieve output vector sequences in a linked list or any other abstract data structure. Rather, the trajectories are computed at run-time as the result of a dynamic process. The 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 possess 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 are fed into input units, the activation results of the hidden units are fed into 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 of an extension of backpropagation for recurrent neural networks including Jordan's and Elman's recurrent neural networks as special cases [18].

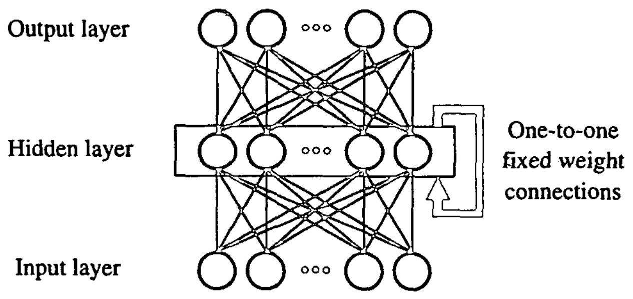


Fig. 2. Elman's recurrent neural network.

The architectures specified by Jordan and Elman employed first-order connections between units. That is, the activation flowing from one unit to another is merely scaled by the connection strength $(w_{ij}o_j)$.

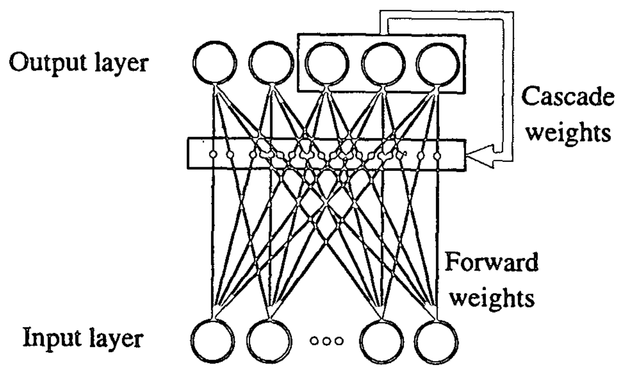


Fig. 3. Pollack's sequential cascaded neural network.

However, the high-order recurrent neural network which has been proposed by Rumelhart *et al.* [19] multiplicatively combine incoming activations $(w_{ijk}o_{j}o_{k})$. One benefit of switching to higher order network is that more functions can be loaded into a network with fewer resources. Just as first-order connections underlie Jordan's and Elman's recurrent neural networks, multiplicative connections have formed 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].

For further information on the 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 property.

1. Architecture of the 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 networks in view of functions and architectures because it was originally extended from the multilayer feedforward neural network for improving discrimination and generalization power in recognizing spatial patterns.

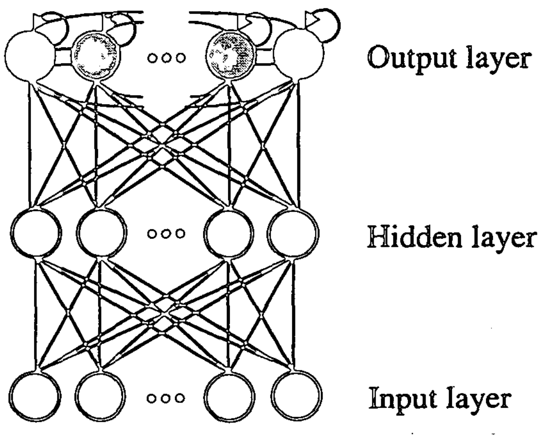


Fig. 4. Proposed recurrent neural network architecture.

Each hidden unit is fully-connected with all input units and each output unit is connected with itself and fully-connected with other output units and all hidden units. Therefore, the output value of 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 jth hidden unit at cycle t, w_{ij} is the weight between jth hidden unit and ith output unit, z_{ik} is the weight between kth output unit and ith output unit, $r_i(t)$ is the recurrent value from 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 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 jth input unit at cycle t, w_{ij} is the weight between jth input unit and ith hidden unit, and n is the number of input units. The input units have linear transfer function.

The proposed recurrent neural netwok operates as follows. The activation values of output units are initially set to be zero. The input feature values are fed into input units which have linear transfer function and the output values of input units are forward propagated until the output units are active. These activation values of output units are used in next cycle in order to provide more information.

Because the output units in a feedforward network are only activated by units in previous layers, the activation values of previous cycle have an effect on the results of output units in current cycle. For the case of training similar spatial patterns, the output units in previous cycle produce ambiguous activation values. However, based on these ambiguous activation values, the weights in output units can be trained to discriminate these ambiguous activation values of previous cycle. Therefore, the discrimination power can be improved.

2. Convergence Property of Learning Algorithm in the Proposed Recurrent Neural Network

We now prove the convergence property of Williams-Zipser learning algorithm [17] in the proposed recurrent neural network. Our result follows from the results of Kuan and White [22] and Kuan *et al.* [18]. Kuan *et al.* provided a rigorous convergence analysis of an extension of back-propagation 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 measure of the network error, the learning recursions, and the limit condition of the weight change of the proposed recurrent neural network are equivalent to those of the Jordan's and Elman's networks in spite of the difference in architecture. Thus, we describe only the assumptions of the *Theorem* which are based on the modified output function and recurrent variable for the proposed recurrent neural network.

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

Suppose that we observe a realization of a sequence $\{Z_t\} = \{Z_t : t = 0, 1, \dots\}$ 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 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 $X^t \equiv (X_0, \dots, X_t)$ 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 $\Theta \subset \boldsymbol{R}^s$, where, s is the number of weights in the network.

In this situation, the approximation error $e_t(\theta)$ is defined as $e_t(\theta) = Y_t - f_t(X^t, \theta)$ and θ^* is selected as follows:

$$\theta^* = \min!\ \lim_{t\to\infty} E(e_t(\theta)^2)/2 \tag{4}$$

where, min! designates a local minimizer of its argument and $E(\cdot)$ denotes mathematical expectation.

In general, stochastic approximation estimates a solution to the equation $M(\theta) = 0$ (with $M: \Theta \to \boldsymbol{R}^s$) as

$$\hat\theta_{t+1} = \hat\theta_t - \eta_t m_t(Z^t, \hat\theta_t), \qquad t = 0, 1, \dots, \tag{5}$$

where, $M(\theta) = \lim_{t\to\infty} E(m_t(Z^t, \theta))$, η_t is a "learning rate" sequence, and

$$m_t(Z^t, \theta) = \nabla e_t(\theta)\, e_t(\theta). \tag{6}$$

We take f_t as the output function of a recurrent neural network with output given in cycle t by

$$O_t = F(\alpha + A_t^T \beta + R^{T_t} \delta) \tag{7}$$

with

$$A_{tj} = G(X_t^T \gamma_j), \qquad j = 1, \dots, q \tag{8}$$

$$R_t = \rho(X_{t-1}, R_{t-1}, \theta) \tag{9}$$

where, $F: \boldsymbol{R} \to \boldsymbol{R}$, $G: \boldsymbol{R} \to \boldsymbol{R}$ are given functions, parameter α, β, γ, and δ collected together in the network weight vector $\theta = (\alpha, \beta^T, \gamma^T, \delta^T)^T$, and R_t is the recurrent values determined from previous input (X_{t-1}), previous recurrent values (R_{t-1}), and network weight (θ) through ρ. In the proposed recurrent neural network, the function ρ is defined as follows:

$$\rho(X_{t-1}, R_{t-1}, \theta) = F(\alpha + A_{t-1}^T \beta + R^{T_{t-1}} \delta). \tag{10}$$

Because e_t is defined as $e_t = Y_t - O_t$, network error depends on Z_t, R_t, and θ. Thus, this network is a particular case of a generic class of models with network errors

$$e_t = u(Z_t, R_t, \theta) \tag{11}$$

where, the function u results from the assumed network output function, and R_t is determined by network recurrence and given as

$$R_t = \rho(Z_{t-1}, \rho(Z_{t-2}, \dots, \theta), \theta) \equiv l_t(Z^{t-1}, \theta). \tag{12}$$

Network error is then

$$e_t(\theta) = u(Z_t, l_t(Z^{t-1}, \theta), \theta). \tag{13}$$

The gradient ∇e_t needed for learning is

$$\nabla e_t(\theta) = u_\theta(Z_t, l_t(Z^{t-1}, \theta), \theta)^T + \nabla l_t(Z^{t-1}, \theta)\, u_r(Z_t, l_t(Z^{t-1}, \theta), \theta)^T \tag{14}$$

where, U_θ is the derivative of u with respect to θ $(u_\theta^T = \nabla u)$, u_r is the derivative of u with respect to recurrent variables, and ∇l_t is the gradient matrix of l_t with respect to θ.

A computationally convenient alternative results from exploiting the recursive structure of R_t. Because

$$R_t = l_t(Z^{t-1}, \theta) = \rho(Z_{t-1}, l_{t-1}(Z^{t-2}, \theta), \theta), \tag{15}$$

it follows that

$$\nabla l_t(Z^{t-1}, \theta) = \rho_\theta(Z_{t-1}, R_{t-1}, \theta)^T + \nabla l_{t-1}(Z^{t-2}, \theta)\, \rho_r(Z_{t-1}, R_{t-1}, \theta)^T \tag{16}$$

where, ρ_θ is the Jacobian matrix of ρ with respect to θ $(\rho_\theta^T = \nabla_\rho)$ and ρ_r is the Jacobian matrix of ρ with respect to recurrent variables. With $\triangle_t = \nabla l_t(Z^{t-1}, \theta)$, we have a recursion as follows:

$$\triangle_t = \rho_\theta(Z_{t-1}, R_{t-1}, \theta)^T + \triangle_{t-1}\rho_r(Z_{t-1}, R_{t-1}, \theta)^T. \tag{17}$$

The recursion for R_t and $\triangle_t$ suggests a learning algorithm that updates R_t and $\triangle_t$ with the weight update in cycle t but neglects the effect of weight updates on past values. If the system does not have "too long" memory and if we eventually get "close" to θ^*, then sufficiently little may be lost by ignoring the update effects.

Thus, 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 the network error and the gradient as follows:

$$\hat e_0 = u(Z_0, \hat R_0, \hat\theta_0) \tag{18}$$

$$\nabla \hat e_0 = u_\theta(Z_0, \hat R_0, \hat\theta_0)^T + \hat\triangle_0 u_r(Z_0, \hat R_0, \hat\theta_0)^T. \tag{19}$$

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

$$\hat\theta_1 = \hat\theta_0 - \eta_0 \nabla \hat e_0 \cdot e_0. \tag{20}$$

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

$$\hat R_1 = \rho(Z_0, \hat R_0, \hat\theta_0) \tag{21}$$

and

$$\widehat{\triangle}_t = \rho_\theta(Z_0, \hat{R}_0, \hat{\theta}_0)^T + \widehat{\triangle}_0 \rho_r(Z_0, \hat{R}_0, \hat{\theta}_0)^T. \qquad (22)$$

Now the network error and the gradient are calculated as follows:

$$\hat{e}_1 = u(Z_1, \hat{R}_1, \hat{\theta}_1) \qquad (23)$$

$$\nabla \hat{e}_1 = u_\theta(Z_1, \hat{R}_1, \hat{\theta}_1)^T + \widehat{\triangle}_1 u_r(Z_1, \hat{R}_1, \hat{\theta}_1)^T. \qquad (24)$$

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

$$\hat{\theta}_2 = \hat{\theta}_1 - \eta_1 \nabla \hat{e}_1 \cdot \hat{e}_1. \qquad (25)$$

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

$$\begin{aligned} \hat{e}_t &= u(Z_t, \hat{R}_t, \hat{\theta}_t) \\ \nabla \hat{e}_t &= u_\theta(Z_t, \hat{R}_t, \hat{\theta}_t)^T + \widehat{\triangle}_t u_r(Z_t, \hat{R}_t, \hat{\theta}_t)^T \\ \hat{\theta}_{t+1} &= \hat{\theta}_t - \eta_t \nabla \hat{e}_t \cdot \hat{e}_t \\ \hat{R}_{t+1} &= \rho(Z_t, \hat{R}_t, \hat{\theta}_t) \end{aligned} \qquad (26)$$

and

$$\widehat{\triangle}_{t+1} = \rho_\theta(Z_t, \hat{R}_t, \hat{\theta}_t)^T + \widehat{\triangle}_t \rho_r(Z_t, \hat{R}_t, \hat{\theta}_t)^T. \qquad (27)$$

A potential difficulty is that nothing prevents $\hat{\theta}_t \to \infty$. To avoid this, we employ a projection operator $\pi : \boldsymbol{R}^s \to \Theta$, where Θ is a compact subset of $\boldsymbol{R}^s$. 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 on a probability space (Ω, cf, P) and define the mixing coefficients

$$\phi_m \equiv sup_t \; sup_{\{F \in cf_{-\infty}^t, G \in cf_{t+m}^{\infty}: P(F) > 0\}} |P(G|F) - P(G)| \qquad (28)$$

$$\alpha_m \equiv sup_t \; sup_{\{F \in cf_{-\infty}^t, G \in cf_{t+m}^{\infty}\}} |P(G \cap F) - P(G)P(F)| \qquad (29)$$

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

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

Definition 3.1 *Let* $\{Z_t\}$ be a sequence of random variables belonging to $L_2(P)$, and let $\{V_t\}$ be a stochastic process on $(\Omega, calF, P)$. 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 (Ω, cf, P) is a complete probability space which is defined by the sequence of cf-measurable functions $\{Z_t : \Omega \to \boldsymbol{R}^{v+1}, t = 0, 1, 2, \ldots\}$ with $\sup_{t \geq 0} |Z_t| \leq \varepsilon^{-1} < \infty$, where $v \in \boldsymbol{N}$ is the size of input vector. $\{Z_t\}$ is NED on $\{V_t\}$ of size $-\frac{1}{2}$ where $\{V_t, t = 0, \pm 1, \pm 2, \ldots\}$ is a mixing process on (Ω, cf, P) with ϕ_m of size $-\frac{1}{2}$ or α_m of size -1. For each $t = 0, 1, \ldots, Z_t$ is measurable $cf^t \equiv \sigma(\ldots, V_{t-1}, V_t)$.

The following condition restricts the network error function.

Assumption A.2 Network output is given by

$$o = F\Big(\alpha + \sum_{i=1}^{q} (\beta_i G(x^T \gamma_i) + r^T \delta_i)\Big) \qquad (30)$$

and network error is given by

$$u(z, r, \theta) = y - F\Big(\alpha + \sum_{i=1}^{q} (\beta_i G(x^T \gamma_i) + r^T \delta_i)\Big). \qquad (31)$$

Then, the network recurrence is obtained as follows:

$$\rho(z, r, \theta) = F\Big(\alpha + \sum_{i=1}^{q} (\beta_i G(x^T \gamma_i) + r^T \delta_i)\Big). \qquad (32)$$

The mean value theorem for such functions ensures

$$|\rho(z, r_1, \theta) - \rho(z, r_2, \theta)| \leq \Big(\sup_{z \in K_z, r \in K_r, \theta \in \Theta} |\rho_r(z, r, \theta)| \Big) |r_1 - r_2|. \qquad (33)$$

The following condition restricts network recurrence.

Assumption A.3 Network recurrence is determined by equation (32). Put

$$c_F = sup_{b \in K_F} |F'(b)|.$$

Then Θ is such that $sum_{i=1}^{q} |\delta_i| \leq c_F^{-1}(1 - \varepsilon)$ for some $\varepsilon > 0$. Because the Jacobian matrix of ρ with respect to recurrent variables is calculated as follows:

$$\rho_r(z, r, \theta) = F'\Big(\alpha + \sum_{i=1}^{q} \beta_i G(x^T \gamma_i) + \sum_{i=1}^{q} r^T \delta_i\Big) \cdot \sum_{i=1}^{q} \delta_i, \qquad (34)$$

the restriction of network recurrence is given by

$$|\rho_r(z, r, \theta)| \leq |F'(\alpha + \sum_{i=1}^{q} \beta_i G(x^T \gamma_i) + \sum_{i=1}^{q} r^T \delta_i)| \sum_{i=1}^{q} |\delta_i| \qquad (35)$$

$$\leq c_F \sum_{i=1}^{q} |\delta_i|. \qquad (36)$$

Now, the learning recursions are formally described as follows.

Assumption A.4 (i) Let $K_\triangle$ be a compact subset of $\boldsymbol{R}^{s \times p}$ where p is the size of recurrent variable, and let $\hat{R}_0 \in K_r$, $\widehat{\triangle}_0 \in K_\triangle$, and $\hat{\theta}_0$ be chosen arbitrarily and independently of

$\{Z_t\}$. For $t=0,1,2,\ldots,$ define

$$\hat{e}_t = u(Z_t, \hat{R}_t, \hat{\theta}_t) \tag{37}$$

$$\nabla \hat{e}_t = u_\theta(Z_t, \hat{R}_t, \hat{\theta}_t)^T + \hat{\triangle}_t u_r(Z_t, \hat{R}_t, \hat{\theta}_t)^T \tag{38}$$

$$\hat{\theta}_{t+1} = \pi[\hat{\theta}_t - \eta_t \nabla \hat{e}_t \hat{e}_t] \tag{39}$$

$$\hat{R}_{t+1} = \rho(Z_t, \hat{R}_t, \hat{\theta}_t) \tag{40}$$

and

$$\hat{\triangle}_{t+1} = \rho_\theta(Z_t, R_t, \hat{\theta}_t)^T + \hat{\triangle}_t \rho_r(Z_t, \hat{R}_t, \hat{\theta}_t)^T \tag{41}$$

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

(ii) $\{\eta_t\}$ 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 a function h as

$$h(\lambda, \theta) = -[u_\theta(z, r, \theta)^T + \triangle u_r(z, r, \theta)^T] u(z, r, \theta) \tag{42}$$

where, $\lambda \equiv (z^T, r^T, \text{vec}^T \triangle)^T$. We also define $\lambda_{K\theta})$ 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 $\lambda_t^\triangle(\theta) \equiv \text{vec} \nabla l_t(Z^{t-1}, \theta)$.

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 the 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

$$\hat{\theta}_0(\tau) = \eta_t^{-1}(\tau_{t+1} - \tau)\, \hat{\theta}_t + \eta_t^{-1}(\tau - \tau_t)\, \hat{\theta}_{t+1}, \quad \tau \in [\tau_t, \tau_{t+1}] \tag{43}$$

and its leftward shifts are defined as

$$\hat{\theta}_t(\tau) = \begin{cases} \hat{\theta}_0(\tau_t + \tau) & \tau \geq -\tau_t \\ \hat{\theta}_\tau & \tau < -\tau_t \end{cases} \quad t = 0, 1, 2, \ldots \tag{44}$$

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

$$\bar{\pi}[v(\theta)] = \lim_{\delta \to 0} [\pi(\theta + \delta v(\theta)) - \theta] / \delta, \quad \theta \in \Theta \tag{45}$$

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

The desired convergence of proposed recurrent network now follows immediately.

Theorem 3.2 *Suppose that Assumption* $A.1 \sim A.5$ *hold. Then*

(a) There exists a P-null set Ω_0' such that for $\omega \notin \Omega_0$, $\{\bar{\theta}_t(\cdot)\}$ is bounded and equicontinuous on bounded intervals, and $\{\hat{\theta}_t(\cdot)\}$ has a convergent subsequence whose limit $\bar{\theta}_t(\cdot)$ satisfies the limit expectation $\theta = \bar{\pi}[\bar{h}(\theta)]$.

Let Θ^* be the set of locally asymptotically stable (in the sense of Liapunov) equilibria in for this limit expectation with domain of attraction $d(\Theta^*) \subseteq \boldsymbol{R}^s$.

(b) If $\Theta \subseteq d(\Theta^*)$, then $\hat{\theta}_t \to \Theta^*$ as $t \to \infty$ with probability 1.

(c) 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 1.

(d) Given the conditions in (c), if Θ^* contains only finitely many 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 1.

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

IV. Experimental Results

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

1. Database

The handwritten numeral database of Concordia University consists of totally unconstrained 6,000 numerals originally collected from dead letter envelopes by the U. S. Postal Services at different locations in the U. S. 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]. 4,000 numerals were used for training and 2,000 numerals were used for testing.

2. 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:

Feed forward NN: Simple three layer feedforward neural network
Jordan RNN: Jordan's recurrent neural network
Elman RNN: Elman's recurrent neural network
Proposed RNN: Proposed recurrent neural network

The input pattern has been size-normalized by 16×16 and then directional feature vectors for horizontal, vertical, right-diagonal, and left-diagonal directions are calculated from a size-normalized image by using Kirsch-like masks [25]. And each 16×16 directional feature vector is compressed to 4×4 features. Also, in order to consider the global characteristics of input image, we compressed the 16×16 normalized input image into 4×4 image and used this compressed image as global features. As a result, final features consist of $5 \times 4 \times 4$ features; $4 \times 4 \times 4$ local features and $1 \times 4 \times 4$ global features. These features have been used as input values of neural networks in which the input layer and the hidden layer consist of 80 units respectively, and the output. layer consists of 10 units. The recurrent neural network accepts the feature values in twice for a character in order to activate output units.

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

3. Experimental Results and Analysis

Fig. 5 shows the learning curves for the four different neural networks. As shown in Fig. 5, the proposed recurrent neural network greatly improved the convergence speed.

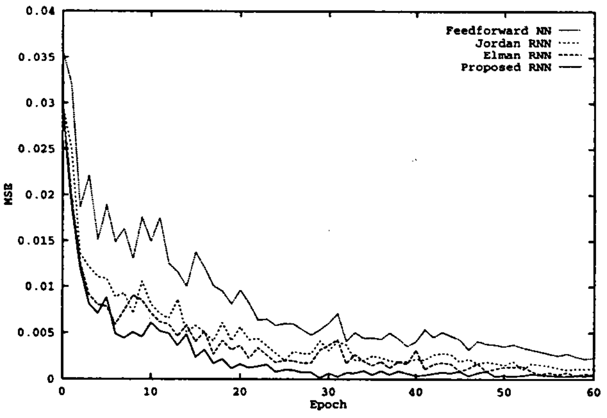


Fig. 5. Learning curves for the four different neural networks.

We also have examined the error rates vs. training set size in order to verify the generalization power of the proposed network. The number of training set has been varied from 800 to 4,000, fixing the number of testing set to 2,000. Fig. 6 shows the changes of the error rate without rejection on the testing set as thd size of training set increases. As can be observed from Fig. 6, the proposed recurrent neural network and Elman recurrent neural network showed superior generalization power to the other neural networks.

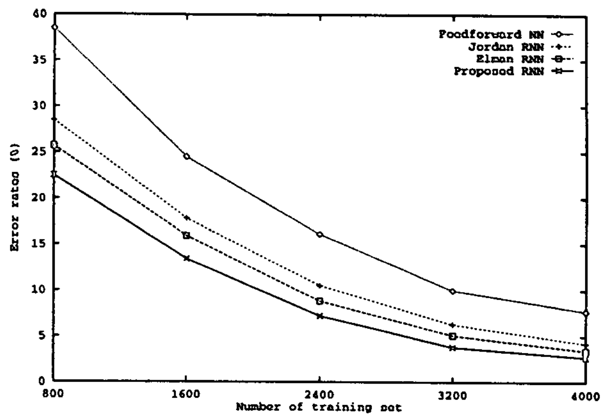


Fig. 6. Error rates vs. training set size for the four different neural networks.

Table 1 and 2 show the error rates without rejection on the training set and testing set, respectively. As indicated in Table 2, for the case of using the simple feedforward neural network, the error rate was 3.7%. And, for the case of using the other three types of recurrent neural networks, the error rates were 3.1%, 2.9%, and 2.7%, respectively. These results showed that the proposed recurrent neural network has very good discrimination power compared to the other recurrent neural networks.

Table 1. Error rates on the training set.

Method	Feedforward NN	Jordan RNN	Elman RNN	Proposed RNN
Error rates	0.850%	0.650%	0.625%	0.575%

Table 2. Error rates on the testing set.

Method	Feedforwad NN	Jordan RNN	Elman RNN	Proposed RNN
Error rates	3.7%	3.1%	2.9%	2.7%

Table 3 shows the reductions of error rate for each recurrent neural network compared to the simple feedforward neural network in Table 2. As shown in Table 3, the use of proposed recurrent neural network brought about 24.1% reduction of error rate compared to the simple feedforward neural network. However, for the case of using Jordan recurrent neural network and Elman's recurrent neural network, the reductions of error rate are 20.0% and 18.0%, respectively. The 24.1% reduction of error rate represents statistical significance in unconstrained handwritten numeral recognition.

We also have analyzed the error rates vs. rejection rates to evaluate the discrimination performances of each network on testing set. The results are described in Fig. 7.

Table 3. Reductions of error rate for each recurrent neural network compared to the simple feed-forward neural network.

Method	Jordan RNN	Elman RNN	Proposed RNN
Reductions of error rate	18.0%	20.0%	24.1%

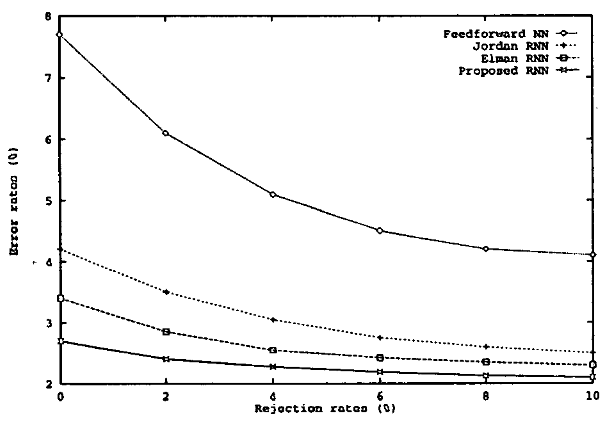


Fig. 7. Error rates vs. rejection rates for the four different neural networks architectures.

Table 4 shows that the proposed neural network can classify similar numerals efficiently.

Table 4. 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%

V. Concluding Remarks

In this paper, we proposed a new type of recurrent neural network architecture in which each output unit is connected with itself and fully-connected with other output units and all hidden units. The proposed recurrent neural network differs from Jordan's and Elman's recurrent networks in view of functions and architectures because it was originally extended from the multilayer feedforward neural network for improving the discrimination and generalization power.

In general, for the case of recognizing spatial patterns with multilayer feedforward neural network, the hidden units are learned to maximize the useful information from input pattern and the output units are learned to discriminate the information given from the hidden layer. Therefore, providing more information to output units in order to improve the discrimination power seems natural.

In this paper, we also proved the convergence property of learning algorithm in the proposed recurrent neural network and analyzed the performance of the proposed architecture by performing the recognition experiments with the totally unconstrained handwritten numeral database of Concordia University of Canada. Experimental results confirmed that the proposed recurrent neural network improves the discrimination and generalization power in recognizing spatial patterns.

Further investigation should be made, however, to design an optimal recurrent neural network architecture which has good generalization power and to implement it on parallel hardwares.

References

[1] Lippmann, R. P., "An Introduction to Computing with Neural Nets," *IEEE ASSP Magazine*, vol. 4, 1987, pp. 4-22.

[2] Suen, C. Y., Nadal, C., Legault, R., Mai, T. A., and Lam, L., "Computer Recognition of Unconstrained Handwritten Numerals," *Proceedings of the IEEE*, vol. 7, 1992, pp. 1162-1180.

[3] Le Cun, Y. *et al.*. "Constrained Neural Newortk for Unconstrained Handwritten Digit Recognition," *Proc. 1st Int. Workshop on Frontiers in Handwriting Recognition*, Montreal, Canada, 1990, pp. 145-154.

[4] Krzyzak, A., Dai, W., and Suen, C. Y., "Unconstrained Handwritten Character Classification Using Modified Backpropagation Model," *Proc. 1st Int. Workshop on Frontiers in Handwriting Recognition*, Montreal, Canada, 1990, pp. 155-166.

[5] Knerr, S., Personnaz, L., and Dreyfus, G., "Handwritten Digit Recognition by Neural Networks with Single-Layer Training," *IEEE Trans. on Neural Networks*, vol. 3, no. 6, 1992, pp. 962-968.

[6] Wieland, A. and Leighton, R., "Geometric Analysis of Neural Network Capabilities," *Proc. of IEEE Int. Conf. on Neural Networks*, vol. III, San Diego, USA, 1987, pp. 385-392.

[7] Sethi, I. K., "Entropy Nets: From Decision Trees to Neural Networks," *Proceedings of the IEEE*, vol. 78, no. 10, 1990, pp. 1605-1613.

[8] Senior, A. W., "Off-line Handwriting Recognition: A

Review and Experiments," Technical Report, TR105, Engineering Department, Cambridge University, 1992.

[9] Kim, S. S. and Chien, S. I., "Improving Generalization Capability and Noise Cancelling with the Partially Connected Recurrent Neural Network," *Proc. of Int. Joint Conf. on Neural Networks*, vol. II, Beijing, 1992, pp. 166-171.

[10] Urbanczik, R., "A Recurrent Neural Network Inverting a Deformable Template Model of Handwritten Digits," *Proc. of Int. Conf. on Artificial Neural Networks*, vol. 2, Sorrento, Italy, 1994, pp. 961-964.

[11] Anderson, J. A. *et al.*, "Distinctive Features, Categorical Perception, and Probability Learning: Some Applications of a Neural Model," *Psychological Review*, vol. 84, 1977, pp. 413-451.

[12] Kolen, J. F., "Exploring the Computational Capabilities of Recurrent Neural Networks," A Thesis for the Degree Doctor of Philosophy, Ohio State University, 1994.

[13] Hopfield, J. J., "Neural Networks and Physical Systems with Emergent Collective Computational Abilities," *Proceedings of the National Academy of Sciences USA*, vol. 79, 1982, pp. 2554-2558.

[14] Hopfield, J. J., "Neurons with Graded Response have Collective Computational Properties like Those of Two-State Neurons," *Proceedings of the National Academy of Sciences USA*, vol. 81, 1984, pp. 3088-3092.

[15] Jordan, M., "Serial Order: A Parallel Distributed Processing Approach," ICS Report 8604, University of California San Diego, Institute for Cognitive Science, 1986.

[16] Elman, J. L., "Finding Structure in Time," *Cognitive Science*, vol. 14, 1990, pp. 179-211.

[17] Williams, R. J. and Zipser, D., "A Learning Algorithm for Continually Running Fully Recurrent Neural Networks," *Neural Computation*, vol. 1, 1989, pp. 270-280.

[18] Kuan, C.-M., Hornik, K., and White, H., "A Convergence Result for Learning in Recurrent Neural Networks," *Neural Computation*, vol. 6, 1994, pp. 420-440.

[19] Rumelhart, D. E., Hinton, G. E., and Williams, R. J., "Learning Internal Representations by Error Propagation," *Parallel Distributed Processing*, vol. 1, Cambridge, USA: MIT Press, 1986, pp. 318-362.

[20] Pollack, J. B., "The Induction of Dynamical Recognizers," *Machine Learning*, vol. 7, 1991, pp. 227-252.

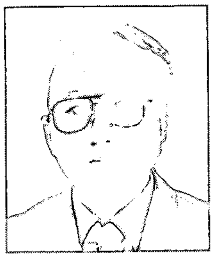
[21] Giles, C. L., Sun, G. Z., Chen, H. H., Lee, Y. C., and Chen, D., "Higher Order Recurrent Networks and Grammatical Inference," *Advances in Neural Information Processing Systems 2*, 1990, pp. 380-387.

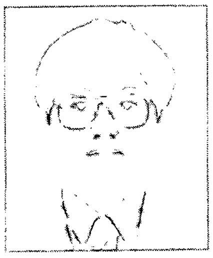
[22] Kuan, C.-M. and White, H., "Adaptive Learning with Nonlinear Dynamics Driven by Dependent Processes," Department of Economics Discussion Paper, University of California San Diego, 1992.

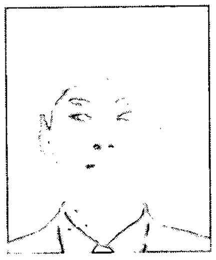
[23] Kushner, H. J. and Clark, D. S., *Stochastic Approximation Methods for Constrained and Unconstrained Systems*, Springer-Verlag, New York, USA, 1978.

[24] Suen, C. Y., Nadal, C., Mai, T. A., Legault, R., and Lam, L., "Recognition of Handwritten Numerals Based on the Concept of Multiple Experts," *Proc. of 1st Int. Workshop on Frontiers in Handwriting Recognition*, Montreal, Canada, 1990, pp. 131-144.

[25] Lee, S.-W., "Multilayer Cluster Neural Network for Off-Line Recognition of Totally Unconstrained Handwritten Numerals," *Neural Networks*, vol. 8, no. 5, 1995, pp. 783-792.


Seong-Whan Lee 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, Korea University, Seoul, Korea. His research interests include document analysis and recognition, pattern recognition, neural networks, and human-computer interaction. He was the winner of the 5th Annual Best Paper Award of the Korea Information Science Society in 1986. He obtained the First Outstanding Young Researcher Award at the 2nd International Conference on Document Analysis and Recognition in 1993.


Hee-Heon Song 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, Daejon, 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 Researcher at Electronics and Telecommunications Research Institute, Daejon, Korea. His research interests include pattern recognition, speech recognition, and neural networks.


Young-Joon Kim received the B. S. degree in physics and the M. S. degree in computer science from Chungbuk National University, Cheounju, Korea, in 1993 and 1995 respectively. Since 1995, he has been a research engineer at BIT Computer Co., Ltd., Seoul, Korea. His research interests include handwritten digit recognition and neural networks.