SOME CONVERGENCE RESULTS FOR LEARNING IN RECURRENT NEURAL NETWORKS[*]

Chung-Ming Kuan, Kurt Hornik and Halbert White

University of Illinois Champaign-Urbana,
Technical University of Vienna and
University of California, San Diego

ABSTRACT

We give a rigorous analysis of the convergence properties of a recurrent back-propagation algorithm for recurrent networks containing either output or hidden layer recurrence. The conditions permit data generated by stochastic processes with considerable dependence. **The theory suggests restrictions relevant in practical applications.**

1. INTRODUCTION

Artificial neural network models are a class of flexible nonlinear functions developed by cognitive scientists that are useful in forecasting, pattern recognition, signal processing and process control applications. In "feedforward" networks, inputs activate "hidden" units, which in turn determine output activation. In these networks signals flow in only one **direction, without feedback.**

Applications in forecasting, signal processing and control require explicit treatment

[*] This work was supported by NSF Grants SES-8806989 and SES-8921382. The authors have benefited from conversations with José Ferreira Machado, Boyan Jovanovich and Max Stinchcombe. Halbert White wishes especially to express his deep appreciation to the Faculty of Economics, New University of Lisbon (Lisbon, Portugal) whose setting and collegiality provided essential inspiration.

of dynamics. Feedforward networks can accommodate dynamics by including lagged (past) input and target values in an augmented set of inputs. However, a much richer dynamic representation results from also allowing for internal network feedbacks. Such "recurrent" structures were used by Jordan (1986) for controlling and learning smooth robot movements, and by Elman (1988) for learning and representing temporal structure in linguistics. In Jordan's network, lagged values of network output feed back into hidden units; in Elman's network, lagged values of hidden units feed back into themselves.

A leading learning method for feedforward networks is "back-propagation" (Werbos, 1974; Parker, 1982; and Rumelhart, Hinton and Williams, 1986). Its convergence properties have been analyzed rigorously by White (1989) for independent identically distributed (i.i.d.) training examples, and by Kuan and White (1990a) for a class of (time dependent) mixingale processes.

Learning methods for recurrent networks are extensions of the method of back-propagation. A very general algorithm is that of Williams and Zipser (1988). Convergence properties of such learning methods for recurrent networks have not yet been rigorously analyzed. This paper provides a rigorous convergence analysis of an extension of back-propagation for recurrent networks containing Jordan and Elman networks as special cases. This method is a special case of the Williams-Zipser algorithm.

The recurrent networks treated here are related to but distinct from the recurrent networks considered by Pineda (1987a,b) and Almeida (1987). These authors treat networks with instantaneous feedback; here we only permit feedback with a time lag. Their network recurrence structures also differ somewhat from those considered here. Consequently, our results do not cover Almeida's and Pineda's recurrent versions of back-propagation.

Our results follow from a result of Kuan and White (1990b) (KW), derived from fundamental results of Kushner and Clark (1978) (KC). The conditions under which convergence holds suggests some restrictions relevant in practice.

2. HEURISTICS AND THE METHOD OF RECURRENT BACK-PROPAGATION

Suppose that we observe a realization of a sequence $\{Z_t\} = \{Z_t : t = 0, 1, ...\}$ of random vectors, where $Z_t = (Y_t, X_t^T)^T$ (with ^{T} denoting the transposition operator), Y_t is (for simplicity) a scalar, and X_t is a $v \times 1$ vector, $v \in \mathbb{N} \equiv \{1, 2, ...\}$. We interpret Y_t as a target value at time t, and X_t as a vector of input variables influencing Y_t and generated by nature. X_t may contain lagged values of Y_t (e.g. Y_{t-1}, Y_{t-2}, ...) as well as lagged values of other variables. For convenience, we assume throughout that the first element of X_t (i.e. X_{t1}) is always equal to one.

Let $X^t \equiv (X_0, ..., X_t)$ denote the history of the X process from time zero through time t. (Similarly, for any sequence $\{a_t\}$, $a^t \equiv (a_0, ..., a_t)$.) Suppose we are interested in approximating $E(Y_t | X^t)$, the conditional expectation of Y_t given X^t, by a parametric function of X^t, so that $f_t : \mathbb{R}^{v(t+1)} \times \Theta \rightarrow \mathbb{R}$ (say) defines a family of approximations $f_t (X^t, \theta)$ as θ ranges over the parameter space $\Theta \subset \mathbb{R}^s$, $s \in \mathbb{N}$, say.

In this situation we define the approximation error $e_t (\theta) = Y_t - f_t (X^t, \theta)$ and select θ^* such that

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

where min! designates a local minimizer of its argument, we assume limits exist, and $E(\cdot)$ denotes mathematical expectation. To see why this is natural, note that

$$E (e_t(\theta)^2) = E([Y_t - E (Y_t | X^t)]^2) + E ([E (Y_t | X^t) - f_t (X^t, \theta)]^2).$$

It follows that θ^* also satisfies

$$\theta^* = \text{min!} \lim_{t \to \infty} E([E (Y_t | X^t) - f_t (X^t, \theta)]^2),$$

and thus indexes a locally mean-square optimal approximation to the limit of $E (Y_t | X^t)$.

Given the validity of an interchange of limit, derivative and expectation, we have

$$\lim_{t \to \infty} \nabla E (e_t(\theta)^2)/2 = \lim_{t \to \infty} E [\nabla e_t (\theta) \cdot e_t (\theta)] = 0$$

as the necessary first order conditions for θ^*, where ∇ is the gradient operator with respect to θ, producing an $s \times 1$ vector. The expectation above is usually unknown; however, the method of stochastic approximation (Robbins and Monro, 1951; Kushner and Clark, 1978) can approximate a solution to the first order conditions.

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

$$\hat{\theta}_{t+1} = \hat{\theta}_t - \eta_t \, m_t \, (Z^t, \hat{\theta}_t), \quad t = 0, 1, \ldots,$$

where $\lim_{t \rightarrow \infty} E \, (m_t \, (Z^t, \theta)) = M(\theta)$, and $\{\eta_t \in \mathbb{R}^+\}$ is a "learning rate" sequence. For our application, we have

$$m_t \, (Z^t, \theta) = \nabla \, e_t \, (\theta) \, e_t \, (\theta).$$

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

$$O_t = F \, (\alpha + A_t^T \beta), \quad \text{with} \tag{2.1a}$$

$$A_{tj} = G \, (X_t^T \gamma_j + R_t^T \delta_j), \quad j = 1, \ldots, q, \tag{2.1b}$$

$$R_{ti} = \rho_i \, (X_{t-1}, R_{t-1}, \theta), \quad i = 1, \ldots, p, \tag{2.1c}$$

where $F: \mathbb{R} \rightarrow \mathbb{R}$, $G: \mathbb{R} \rightarrow \mathbb{I}$ ($\mathbb{I} \equiv [0, 1]$) are given functions (e.g., the logistic function $G(\lambda) = (1 + e^{-\lambda})^{-1}$); A_t is the $q \times 1$ vector of hidden unit activations; parameters are α (1×1), β $(q \times 1)$, $\gamma \equiv (\gamma_1^T, \ldots, \gamma_q^T)^T$ $(qv \times 1)$ and $\delta \equiv (\delta_1^T, \ldots, \delta_q^T)^T$ $(qp \times 1)$ collected together in the $s \times 1$ network weight vector $\theta = (\alpha, \beta^T, \gamma^T, \delta^T)^T$, with $s = 1{+}q{+}q(v{+}p)$; and R_t is the $p \times 1$ vector of recurrent variables, determined from previous inputs (X_{t-1}), previous recurrent values (R_{t-1}) and network weights (θ) through ρ_i, $i = 1, \ldots, p$.

When $R_t = O_{t-1}$, we have the Jordan (1986) network, and

$$\rho_1 \, (X_{t-1}, R_{t-1}, \theta) = F(\alpha + \sum_{j=1}^{q} \beta_j \, G(X_{t-1}^T \, \gamma_j + R_{t-1} \, \delta_j)).$$

When $R_t = A_{t-1}$, we have the Elman (1988) network, and

$$\rho_i\,(X_{t-1}, R_{t-1}, \theta) = G(X_{t-1}^T\ \gamma_i + R_{t-1}^T\ \delta_i), \quad i = 1, ..., q.$$

Substituting for A_{tj} in (2.1a) gives

$$O_t = F(\alpha + \sum_{j=1}^{q} \beta_j\, G(X_t^T \gamma_j + R_t^T\ \delta_j))$$

for the single hidden layer recurrent net. Because $e_t = Y_t - O_t$, network error depends on Z_t, R_t and θ. Thus, this net is a particular case of a generic class of models with errors

$$e_t = u(Z_t, R_t, \theta),$$

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

Above, the recurrent variables were generated as $R_t = \rho\ (X_{t-1}, R_{t-1}, \theta)$, with $\rho \equiv (\rho_1, ..., \rho_p)^T$. However, much flexibility is gained by including Y_{t-1} as a determinant of R_t, so we write $R_t = \rho\ (Z_{t-1}, R_{t-1}, \theta)$.

Because of R_t, network error is a function of the entire history of targets and inputs, Z^t. For a given θ and a given initial recurrent value, say R_0, the recurrent variables are given in time period t as

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

where we have suppressed the dependence of l_t on R_0. Network error is then

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

The gradient ∇e_t needed for learning is

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

where u_θ is the $1 \times s$ derivative of u with respect to θ ($u_\theta^T = \nabla\ u$), u_r is the $1 \times p$ derivative of u

with respect to recurrent variables, and ∇l_t is the $s \times p$ gradient matrix of l_t with respect to θ.

Any learning algorithm based directly on e_t and ∇e_t will be computationally intensive, as the effect of any change in θ must be propagated through time from period zero up to period t. The required computations grow as t increases, and the entire history Z^t must be kept in memory.

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

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

it follows that

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

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

$$\Delta_t = \rho_\theta\,(Z_{t-1}, R_{t-1}, \theta)^T + \Delta_{t-1}\;\rho_r\,(Z_{t-1}, R_{t-1}, \theta)^T.$$

The recursions for R_t and Δ_t suggest a learning algorithm that updates R_t and Δ_t with the weight update in time t but neglects the effect of weight updates on past values. If the system doesn't have "too long" a memory and if we eventually get "close" to θ^*, then sufficiently little may be lost by ignoring the update effects that we still obtain the desired convergence to θ^*.

Thus, we begin by picking arbitrary initial weights $\hat\theta_0$, recurrent variables $\hat R_0$ and $s \times p$ gradient matrix $\hat\Delta_0$. To update network weights we compute network error

$$\hat e_0 = u\,(Z_0, \hat R_0, \hat\theta_0)$$

and form

$$\nabla \hat e_0 = u_\theta\,(Z_0, \hat R_0, \hat\theta_0)^T + \hat\Delta_0\; u_r\,(Z_0, \hat R_0, \hat\theta_0)^T$$

in order to get period 1 weights

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

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

$$\hat{R}_1 = \rho\ (Z_0, \hat{R}_0, \hat{\theta}_0), \quad \text{and}$$

$$\hat{\Delta}_1 = \rho_\theta\ (Z_0, \hat{R}_0, \hat{\theta}_0)^T + \hat{\Delta}_0\ \rho_r\ (Z_0, \hat{R}_0, \hat{\theta}_0)^T.$$

Now we may compute

$$\hat{e}_1 = u\ (Z_1, \hat{R}_1, \hat{\theta}_1) \quad \text{and}$$

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

to obtain period 2 weights

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

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

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

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

$$\hat{\theta}_{t+1} = \hat{\theta}_t - \eta_t \nabla \hat{e}_t \cdot \hat{e}_t, \tag{2.2}$$

$$\hat{R}_{t+1} = \rho\ (Z_t, \hat{R}_t, \hat{\theta}_t), \quad \text{and}$$

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

Note the modest memory and computation requirements of this algorithm.

We refer to this as "recurrent back-propagation", as it generalizes back-propagation to certain recurrent networks. It is a special case of the Williams-Zipser (1988) algorithm. Our main goal is to obtain conditions under which recurrent back-propagation converges as $t \to \infty$ to a desired value, θ^*.

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

3. MAIN RESULTS

In order to state our assumptions, we introduce the notion of a stochastic process near epoch dependent on an underlying mixing process (Billingsley, 1968; McLeish, 1975; Gallant and White, 1988).

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

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

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

where $\mathcal{F}^t_\tau \equiv \sigma(V_\tau, ..., V_t)$. When $\phi_m \to 0$ or $\alpha_m \to 0$ as $m \to \infty$ we say that $\{V_t\}$ is ϕ - mixing or α - mixing. When $\phi_m = O(m^\lambda)$ for some $\lambda < -a$ we say that $\{V_t\}$ is ϕ - mixing of size $-a$, and similarly for α_m. Mixing processes have an asymptotic independence property, although dependence in the short run may be considerable.

Processes formed as functions of infinite histories of mixing processes have longer memories. As long as these functions depend mainly on the "near epoch" of the mixing process, they are still well-behaved enough for our purposes. Let $\|Z_t\|_2 \equiv (E|Z_t|^2)^{1/2}$ and let $L_2(P)$ denote the class of random variables with $\|Z_t\|_2 < \infty$. Let $E^{t+m}_{t-m}(Z_t) \equiv E(Z_t | \mathcal{F}^{t+m}_{t-m})$. We express the dependence of $\{Z_t\}$ on an underlying process $\{V_t\}$ in the following way.

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 $(\Omega, \mathcal{F}, P)$. Then $\{ Z_t$ is near epoch dependent (NED) on $V_t \}$ of size $-a$ if $\upsilon_m \equiv \sup_t \| Z_t - E_{t-m}^{t+m}(Z_t) \|_2$ is of size $-a$. ♦

McLeish (1975, Theorem 3.1) establishes that NED functions of mixing processes are "mixingales", which possess convergence properties that suffice for the convergence conditions of Kushner and Clark (1978).

We may now describe the data generating process.

ASSUMPTION A.1: $(\Omega, \mathcal{F}, P)$ is a complete probability space on which is defined the sequence of $\mathcal{F}$ - measurable functions $Z_t : \Omega \to \mathbb{R}^{v+1}, t = 0, 1, 2,... \}$, $v \in \mathbb{N}$ with $\sup_{t \geq 0}$ $Z_t | \leq \varepsilon^{-1} < \infty$. $\{ Z_t \}$ is NED on V_t of size -1/2 where $V_t, t = 0, \pm 1, \pm 2,...$ is a mixing process on $(\Omega, \mathcal{F}, P)$ with ϕ_m of size -1/2 or α_m of size -1. For each $t = 0, 1,..., Z_t$ is measurable $\mathcal{F}^t \equiv \sigma(..., V_{t-1}, V_t)$. Partition Z_t as $Z_t = (Y_t, X_t^T)^T, X_t : \Omega \to \mathbb{R}^v$, with $X_{t1} = 1, t = 0, 1,....$ ♦

The process generating the input and target sequences is thus bounded and may have a moderately long memory. By convention, $| Z_t | \equiv (\sum_{i=1}^{v+1} Z_{ti}^2)^{1/2}$, and ε is a generic small constant. Let supp Z_t denote the support of Z_t, i.e. the closure of the complement of the largest Borel set B such that $P[Z_t \in B] = 0$, and let jsupp $\{ Z_t \} \equiv cl(\cup_{t=0}^{\infty}$ supp $Z_t)$ denote the "joint support" of $\{ Z_t \}$. Assumption A.1 implies that jsupp $Z_t \} \subset K_z \equiv \times_{i=1}^{v+1} [-\varepsilon^{-1}, \varepsilon^{-1}]$.

The following condition restricts the network error function.

ASSUMPTION A.2: Let D_z, D_r and D_θ be Borel subsets of $\mathbb{R}^{v+1}$, $\mathbb{R}^p$ and $\mathbb{R}^s$ respectively, $p, s, \in \mathbb{N}$, with $K_z \subset D_z$. Then $u : D_z \times D_r \times D_\theta \to \mathbb{R}$ is continuously differentiable of order 2 on $D_z \times D_r \times D_\theta$. ♦

We let u_θ and u_r denote the $1 \times s$ and $1 \times p$ partial derivative functions of u with respect to θ and r.

The next condition restricts network recurrence.

ASSUMPTION A.3: With D_z, D_r and D_θ as in Assumption A.2, let K_r be a compact subset of D_r and let Θ be a compact subset of D_θ.

(i) $\rho : D_z \times D_r \times D_\theta \rightarrow K_r$ is continuously differentiable of order 2 on $D_z \times D_r \times D_\theta$.

(ii) For each (z, θ) in $K_z \times \Theta$, $\rho(z, \cdot, \theta)$ is a contraction mapping on K_r, i.e. $|\rho(z, r_1, \theta) - \rho(z, r_2, \theta)| \leq c_0 | r_1 - r_2 |, c_0 < 1, r_1, r_2, \in K_r$. ♦

We let ρ_θ and ρ_r denote the $p \times s$ and $p \times p$ Jacobian matrices of ρ with respect to θ and r. The contraction property keeps the internal network feedbacks under proper control.

We now state formally the learning recursions.

ASSUMPTION A.4: (i) let K_Δ be a compact subset of $\mathbb{R}^{s \times p}$ and let $\hat{R}_0 \in K_r, \hat{\Delta}_0 \in K_\Delta$ and $\hat{\theta}_0 \in \Theta$ be chosen arbitrarily and independently of $\{Z_t\}$ For $t = 0, 1, 2, ...,$ define

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

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

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

$$\hat{R}_{t+1} = \rho(Z_t, \hat{R}_t, \hat{\theta}_t) \quad \text{and}$$

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

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

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

An important condition is the restriction on the learning rate sequence $\{\eta_t\}$. This condition holds whenever $\eta_t \propto t^{-\mu}$, $1/2 < \mu \leq 1$. The larger values for μ lead to faster convergence. The projection device applied to $\{\hat{\theta}_t\}$ ensures that $\{\hat{\theta}_t$ is bounded. Assumption A.3 ensures that $\{\hat{\Delta}_t\}$ is bounded.

One condition is required KW gence resul guarantees the existence the limit of $E(\nabla$ $(\theta$ $(\theta$ We define functi

$$(\lambda \quad (z, \quad ')^{T} \quad \Delta u_r(z, \quad u(z$$

where λ $\mathrm{vec}^T \Lambda)^T$ and define $\lambda_t(\theta$ $(\lambda\ (\theta$ $\lambda_t^T(\theta)^T$ λ $(\theta$ where $\lambda\ (\theta)$ Z_t $\lambda_t^{\prime}(\theta)$ $l_t(Z$ $')$, and λ (θ) $'ec\ \nabla l_t(Z$ $,\theta)$. Our final condition

ASSUMPTION A.5 Fo each Θ $(\theta$ lim $(h(\lambda_t(\theta\ ,\theta$ exists

With this assumption, $-\bar{h}(\theta$ lim $E(\nabla$ $(\theta$ $e_t(\theta))$, the least squares gradient functi

K resul establish certain properti of the pi ise linear interpolations with interpolation inte als $\{\eta_t$ Define $\sum_{i=0}^{t-1} \eta_i$ The rpolated process defined

$$(\tau \quad \eta_t^{-1}(\tau_t \quad) \quad {}_t^{-1}(\tau \quad [\tau \quad),$$

and leftward shifts defined

$$.(\tau \quad (\tau \quad -\tau_t \quad O$$

$$t_t \quad -\tau_t \quad O$$

In stating the resul ri $\to \Theta$ $\to$ if $\inf_\theta$ $\to$

THE REM .2 (Kuan and White 990b Suppose that Assumpti A. A.: hold. Then

a) there exists -null Ω_0 su that fo Ω_0 $\{$ $\cdot)\}$ is bounded anc equicontinuous bounded intervals and $\{$ $\}$ has convergent sul sequence hose lim satisfies the DE $[\bar{h}(\theta)]$.

Let Θ^* be the set of locally asymptoticall able (in the sense Liap equilibria in fo this ODE with domain $(\Theta^* \subset I\!R$

(b) If Θ (Θ^*), then $\to \Theta$ as $\to$ with probability (w

(c) If Θ is not contained in $d(\Theta^*)$, but for each $\omega \in \Omega_0, \hat{\theta}_t(\omega)$ enters a compact subset of $d(\Theta^*)$ infinitely often, then $\hat{\theta}_t \to \Theta^*$ as $t \to \infty$ w.p.1.

(d) Given the conditions in (c), if Θ^* contains only finitely many points, then $\hat{\theta}_t \to \theta^* \in \Theta^*$ as $t \to \infty$ w.p.1. ♦

The path of the recurrent back-propagation algorithm behaves asymptotically like the solution trajectory of an appropriate ODE. Thus, θ^* satisfies $\bar{h}(\theta^*) = 0$ when $\bar{h}$ has a zero in Θ. These conclusions are identical to those of Kuan and White (1990a) using Theorem 2.4.2 of KC for single hidden layer feedforward networks, except that there θ^* indexes a locally mean square optimal approximation to $E(Y_t \mid X_t)$, while here the approximation is to $E(Y_t \mid X^t)$.

Because output in single hidden layer recurrent networks is given by

$$o = F(\alpha + \sum_{j=1}^{q} \beta_j G(x^T \gamma_j + r^T \delta_j)), \tag{3.1}$$

the network error function is

$$u(z, r, \theta) = y - F(\alpha + \sum_{j=1}^{q} \beta_j G(x^T \gamma_j + r^T \delta_j)). \tag{3.2}$$

For Jordan nets, network recurrence is

$$\rho(z, r, \theta) = F(\alpha + \sum_{j=1}^{q} \beta_j G(x^T \gamma_j + r\, \delta_j)). \tag{3.3}$$

For Elman nets, network recurrence is

$$\rho_i(z, r, \theta) = G(x^T \gamma_i + r^T \delta_i), \quad i = 1, \ldots, q. \tag{3.4}$$

It is now simple to state conditions sufficient for those of Theorem 3.2; we maintain Assumptions A.1, A.4 and A.5, and choose F, G and Θ so that Assumption A.2 and A.3 hold.

The following suffices for Assumption A.2.

ASSUMPTION B.2: Network output is given by (3.1) and network error by (3.2), where

$F: I\!R \rightarrow I\!R$ and $G: I\!R \rightarrow I\!I$ are twice continuously differentiable on $I\!R$. ♦

For example, F may be the identity function, or F and G may be the logistic squasher or tanh squasher. We denote the first derivatives of F and G as F' and G'.

The differentiability conditions of Assumption A.3(i) are satisfied for both Jordan and Elman nets under Assumption B.2. It remains to guarantee that in each case $\rho\,(z, \cdot\,, \theta)$ is a contraction mapping.

First consider the Jordan net. Define the compact sets $K_F \equiv \{b \in I\!R : b = \alpha + \sum_{j=1}^{q} \beta_j \, a_j,\, a_j \in I\!I, \theta \in \Theta\}$ and $K_G \equiv \{\, a \in I\!R : a = x^T \gamma_j + r^T \delta_j,$ $z \in K_z, r \in K_r, \theta \in \Theta\}$, where here $K_r = co\, F(K_F)$, the convex hull of the image of K_F under F. The mean value theorem ensures that in the convex compact set K_r

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

We have that

$$\rho_r(z, r, \theta) = F'(\alpha + \sum_{i=1}^{q} \beta_i \, G\,(x^T \gamma_i + r\, \delta_i))\, [\sum_{j=1}^{q} G'(x^T r_j + r\, \delta_j) \beta_j \, \delta_j]$$

is a scalar, so

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

The continuity of F' and G' and the compactness of K_F and K_G imply the existence of constants c_F and c_G bounding $F'(b)$ and $G'(a)$ for all $b \in K_F$, $a \in K_G$, so

$$|\rho_r(z, r, \theta)\,| \leq c_F \, c_G \sum_{j=1}^{q} |\beta_j \quad \delta_j\,|.$$

This is less than 1 as we require if $\sum_{j=1}^{q} |\beta_j \quad \delta_j\,| < (c_F\, c_G)^{-1}$, so we impose the following condition.

ASSUMPTION B.3(a) (Jordan): Network recurrence is determined by (3.3). Put $c_F = \sup_{b \in K_r} | F'(b) |$, $c_G = \sup_{a \in K_a} | G'(a) |$. Then Θ is such that $\sum_{j=1}^{q} \beta_j \delta_j \leq (c_F\, c_G)^{-1} (1-\varepsilon)$ for some $\varepsilon > 0$. ♦

For example, if $F(a) = G(a) = (1 + e^{-a})^{-1}$ (logistic squashing at both hidden and output layers) then $c_F = c_G = 1/4$, so contraction is ensured by imposing $\sum_{j=1}^{q} |\beta_j| \, |\delta_j| \leq 16\,(1-\varepsilon)$. The theory thus provides a concrete benefit, insofar as this restriction aids practical implementations.

For the Elman net, set $K_r = \mathbb{I}^q$ in defining K_G. Now ρ is a vector-valued function. The mean value theorem for such functions again ensures

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

where now $| \rho_r(z, r, \theta)$ is the square root of the maximum eigenvalue of $\rho_r(z, r, \theta)\overset{T}{\rho}_r(z, r, \theta)^{\cancel{T}}$. Now

$$\rho_{ri}(z, r, \theta) = G'(x^T \gamma_i + r^T \delta_i)\, \delta_i^T \quad i = 1, \; ., q,$$

so

$$| \rho_r(z, r, \theta) | \leq (tr\, \rho_r(z, r, \theta)\overset{T}{\rho}_r(z, r, \theta)^{\cancel{T}})^{1/2}$$

$$= (\sum_{i=1}^{q} G'(x^T \gamma_i + r^T \delta_i)^2\, \delta_i^T \delta_i)^{1/2}$$

$$\leq c_G (\sum_{i=1}^{q} \delta_i^T \delta_i)^{1/2}.$$

We obtain the contraction property using

ASSUMPTION B.3(b) (Elman): Network recurrence is determined by (3.4). Put $c_G = \sup_{a \in K_a} | G'(a) |$. Then Θ is such that $(\sum_{i=1}^{q} \delta_i^T \delta_i)^{1/2} \leq c_G^{-1} (1-\varepsilon)$ for some $\varepsilon > 0$. ♦

The desired convergence now follows immediately.

COROLLARY 3.3: Given Assumptions A.1, B.2, B.3(a) or B.3(b), A.4 and A.5, the conclusions of Theorem 3.2 hold. ♦

Thus, recurrent back-propagation in the Jordan or Elman nets converges in the precise sense established by Theorem 3.2, provided that network weights are sufficiently restricted as to ensure a contraction mapping property for network recurrence.

4. SUMMARY AND CONCLUDING REMARKS

We have applied a result of Kuan and White (1990b) to establish the almost sure convergence of recurrent back-propagation for Jordan and Elman nets. Other recurrent structures are readily handled by applying KW's result, for example, recurrent networks with several hidden layers feeding back into one another in various ways. The key condition is that network recurrence have a contraction mapping property. We also draw attention to the learning rate restriction of Assumption A.4(ii).

For simplicity, we did not permit control or manipulation of the system generating the data; however KW's results apply directly to this case also. Specifically, the present convergence results extend to situations in which a recurrent network is learning while controlling an unknown system with internal feedback and output subject to exogenous noise. Some interesting difficulties arise in this context due to the network's ignorance of the recurrence structure of the system. In particular, the convergence is no longer necessarily to a locally mean square optimal approximation to expected system behavior. The interested reader is referred to KW for additional detail.

REFERENCES

Almeida, L.B. (1987): "A Learning Rule for Asynchronous Perceptrons with Feedback in a Combinatorial Environment," in *Proceedings of the IEEE First International Conference on Neural Networks*. New York: IEEE Press, pp. II:609-618.

Billingsley, P. (1968): *Convergence of Probability Measures*. New York: John Wiley and Sons.

Elman, J.L. (1988): "Finding Structure in Time," University of California, San Diego, Center for Research in Language, CRL Report 8801.

Gallant, A.R. and H. White (1988): *A Unified Theory of Estimation and Inference for Nonlinear Dynamic Models*. Oxford: Basil Blackwell.

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

Kuan, C.M. and H. White (1990a): "Recursive *m*-Estimation, Nonlinear Regression and Neural Network Learning with Dependent Observations," University of California, San Diego, Department of Economics Discussion Paper.

Kuan, C.M. and H. White (1990b): "Convergence to Learning Equilibria with Misspecified Nonlinear Dynamic Models," University of California, San Diego, Department of Economics Discussion Paper.

Kushner, H.J. and D.S. Clark (1978): *Stochastic Approximation Methods for Constrained and Unconstrained Systems*. New York: Springer Verlag.

McLeish, D.L. (1975): "A Maximal Inequality and Dependent Strong Laws," *Annals of Probability* 3, 829-839.

Parker, D.B. (1982): "Learning Logic," Invention Report 581-64 (File 1), Stanford University, Office of Technology Licensing.

Pineda, F.J. (1987a): "Generalization of Back-Propagation to Recurrent Neural Networks," *Physical Review Letters* 59, 2229-2232.

Pineda, F.J. (1987b): "Generalization of Back-Propagation to Recurrent and Higher Order Neural Networks," in *Proceedings of the IEEE Conference on Neural Information Processing Systems*. New York: IEEE Press.

Robbins, H. and S. Monro (1951): "A Stochastic Approximation Method," *Annals of Mathematical Statistics* 22, 400-407.

Rumelhart, D.E., G.E. Hinton and R.J. Williams (1986): "Learning Internal Representations by Error Propagation," in D.E. Rumelhart, J.L. McClelland, and the PDP Research Group, *Parallel Distributed Processing: Explorations in the Microstructures of Cognition*, vol. 1. Cambridge: MIT Press, pp. 318-362.

Werbos, P. (1974): "Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences," unpublished Ph.D. dissertation, Harvard University, Department of Applied Mathematics.

White, H. (1989): "Some Asymptotic Results for Learning in Single Hidden Layer Feedforward Network Models," *Journal of the American Statistical Association* 84, 1003-1013.

Williams, R.J. and D. Zipser (1988): "A Learning Algorithm for Continually Running Fully Recurrent Neural Networks," University of California, San Diego, Institute for Cognitive Science, ICS Report 8805.