$$\frac{du_i(t)}{dt} = -a_i u_i(t) + \sum_{j=1}^{n} w_{ij} f_j(u_j(t)) + \sum_{j=1}^{n} w_{ij}^T f_j(u_j(t-\tau_{ij})) + I_i, i = 1, 2, \ldots, n. \quad (1)$$

$$\left\{\begin{array}{ll} A_I := & \{A = \mathrm{diag}(a_i) : \underline{A} \le A \le \overline{A}, i.e., \underline{a_i} \le a_i \le \overline{a_i}, i = 1, 2, \ldots, n, \forall A \in A_I\} \\ W_I := & \{W = (w_{ij})_{n\times n} : \underline{W} \le W \le \overline{W}, i.e., \underline{w_{ij}} \le w_{ij} \le \overline{w_{ij}}, i, j = 1, 2, \ldots, n, \\ \forall W \in & W_I\} \\ W_I^T := & \{W^T = (w_{ij}^T)_{n\times n} : \underline{W^T} \le W^T \le \overline{W^T}, i.e., \underline{w_{ij}^T} \le w_{ij}^T \le \overline{w_{ij}^T}, i, j = 1, 2, \ldots, n, \\ \forall W^T \in & W_I^T\} \\ \tau_I := & \{\tau = (\tau_{ij})_{n\times n} : \underline{\tau} \le \tau \le \overline{\tau}, i.e., \underline{\tau_{ij}} \le \tau_{ij} \le \overline{\tau_{ij}}, i, j = 1, 2, \ldots, n, \forall \tau \in \tau_I\}. \end{array}\right\}. \quad (2)$$

In general

$$\sum_{i=1}^{n}[|\beta_i a_i u_i| - |\sum_{j=1}^{n}\beta_i(w_{ij}+w_{ij}^T)f_j(u_j)|] \neq \sum_{i=1}^{n}[|\beta_i a_i u_i| - |\sum_{j=1}^{n}\beta_j(w_{ji}+w_{ji}^T)f_i(u_i)|]. \quad (6)$$

In fact, it is not difficult to show that (5) does not hold as illustrated in the following counter-example.

Example: For convenience, let

$$\begin{aligned} \beta_i =& 1, u_i^0 = 1, f_i(u_i^0) = 1, i = 1, 2. a_1 = 3, a_2 = 5 \\ w_{11} =& 2, w_{11}^T = -3, w_{12} = 3, w_{12}^T = 5 \\ w_{21} =& -1, w_{21}^T = 1, w_{22} = -2, w_{22}^T = 2 \end{aligned}$$

then

$$\begin{aligned} &\sum_{i=1}^{2}[|\beta_i a_i u_i^0| - |\sum_{j=1}^{2}\beta_i(w_{ij}+w_{ij}^T)f_j(u_j^0)|] \\ &=3 - |2-3+3+5| + 5 - |-1+1-2+2| = 1 \\ &\sum_{i=1}^{2}[|\beta_i a_i u_i^0| - |\sum_{j=1}^{2}\beta_j(w_{ji}+w_{ji}^T)f_i(u_i^0)|] \\ &=3 - |2-3-1+1| + 5 - |3+5-2+2| = -1. \end{aligned}$$

So

$$\sum_{i=1}^{2}[|\beta_i a_i u_i^0| - |\sum_{j=1}^{2}\beta_i(w_{ij}+w_{ij}^T)f_j(u_j^0)|] \neq \sum_{i=1}^{2}[|\beta_i a_i u_i^0| - |\sum_{j=1}^{2}\beta_j(w_{ji}+w_{ji}^T)f_i(u_i^0)|].$$

That is, (6) holds. Therefore, it is concluded that (5) is not true, and we are sure that the theorem of Liao and Yu is based on this wrong inequality (5) does not in general hold.

Let us take a simple example: all intervals are single point. Then $\omega_{ij} = -\omega_{ij}^T$ implies that $B = \mathrm{diag}(a_1, \ldots, a_n)$ is a M-matrix. However, in this case the theorem in [1] is not valid generality. In fact $n = 1, f(u) = u, I = 0$

$$\begin{aligned} \frac{du}{dt} &= -(1+\epsilon)u(t) + u(t) - u(t-\tau) \\ &= -\epsilon u(t) - u(t-\tau). \end{aligned} \quad (7)$$

It is clear that zero is not a stable equilibrium point of (7).

Acknowledgment

The author is very grateful to the reviewers for their help, comments, and suggestions.

References

[1] X. Liao and J. Yu, "Robust stability for interval Hopfield neural networks with time delay," *IEEE Trans. Neural Networks*, vol. 9, pp. 1042–1045.

[2] J. J. Hopfield, "Neurons with graded response have collective computational properties like these of two-state neurons," in *Proc. Nat. Academy Sci. USA*, 1984, pp. 3088–3092.

A General Backpropagation Algorithm for Feedforward Neural Networks Learning

Xinghuo Yu, M. Onder Efe, and Okyay Kaynak

***Abstract*—In this letter, a general backpropagation algorithm is proposed for feedforward neural networks learning with time varying inputs. The Lyapunov function approach is used to rigorously analyze the convergence of weights, with the use of the algorithm, toward minima of the error function. Sufficient conditions to guarantee the convergence of weights for time varying inputs are derived. It is shown that most commonly used backpropagation learning algorithms are special cases of the developed general algorithm.**

***Index Terms*—Backpropagation, feedforward neural networks, stability, training.**

I. Introduction

Feedforward neural networks (FNN) have been widely used for various tasks, such as pattern recognition, function approximation, dynamical modeling, data mining, and time series forecasting, to name just a few [1], [2]. The training of FNN is mainly undertaken using the backpropagation (BP)-based learning algorithms. A number of different kinds of BP learning algorithms have been proposed, such as an on-line neural-network learning algorithm for dealing with time varying inputs [3], fast learning algorithms based on gradient descent of neuron space [4], and the Levenberg–Marquardt algorithm [5], [6].

Manuscript received August 1, 2001.

X. Yu is with the Faculty of Informatics and Communication, Central Queensland University, Rockhampton, QLD 4702, Australia (e-mail: x.yu@cqu.edu.au).

M. O. Efe is with the Electrical and Computer Engineering Department, Carnegie Mellon University, Pittsburgh, PA 15213-3890 USA.

O. Kaynak is with the Department of Electrical and Electronic Engineering, Bogazici University, Bebek, 80815 Istanbul, Turkey.

Publisher Item Identifier S 1045-9227(02)00348-X.

In this letter, we will develop a general BP learning algorithm for FNN with time varying inputs. This algorithm unifies variations of the BP learning algorithms. The Lyapunov function approach, which has been widely used in analyzing the stability of self-organizing neural networks such as Kohonen and Hopfield types of networks [11], [12], will be used to derive conditions to guarantee the convergence of weights. We will show that trapping into local minima is inherent with the learning algorithms based on the BP principle as they may only enable the weights to converge to global minima if it happens that either the initial weights are near a global minimum or the geometric distribution of weights enables the weights to converge to a global minimum. We will also show that major classes of BP learning algorithms are special cases of the developed learning algorithm.

II. Main Results

Before we proceed, denote the inputs, weights, desired outputs, and actual outputs of the FNN as

$$x(t) = (x_1, x_2, \ldots, x_n)^T \in R^n \tag{1}$$

$$\phi(t) = (\phi_1, \phi_2, \ldots, \phi_l)^T \in R^l \tag{2}$$

$$y_d(t) = (y_{d1}, y_{d2}, \ldots, y_{dm})^T \in R^m \tag{3}$$

$$y(t) = (y_1, y_2, \ldots, y_m)^T \in R^m \tag{4}$$

where $x(t)$ is the input vector, $\phi(t)$ the weight vector, $y_d(t)$ the desired output vector, and $y(t)$ the output vector of the FNN. The error at any instant is represented as

$$\begin{aligned} e(t) &= \tfrac{1}{2}(y(t) - y_d(t))^T (y(t) - y_d(t)) \\ &= \tfrac{1}{2}\|y(t) - y_d(t)\|^2 \end{aligned} \tag{5}$$

where the symbol "T" represents the transpose. Note that here the input $x(t)$ is of a general type, and it can be discrete, continuous, and time varying. The weight vector, $\phi(t)$, represents weights for perceptrons (single-layer FNN) as well as multilayer FNN.

We now develop the criterion for evaluating the performance of the FNN learning. Since the inputs can be time varying, a time window should be used to evaluate the training efficiency [3], [8], that is

$$J = \frac{1}{\tau} \int_{t-\tau}^{t} e(\theta)\, d\theta \tag{6}$$

where τ is the length of the time window. The formulation (6) is particularly useful for on line continuous time learning as it considers evolution of learning in an average sense within a prescribed time window. However, for discrete data sets, since the evaluation of errors can only be done at "discrete moments," (6) can be rewritten as

$$J_k = \lim_{\tau \to 0} \frac{1}{\tau} \int_{t_k-\tau}^{t_k} e(\theta)\, d\theta = e(k) \tag{7}$$

which becomes the usual form for training FNN with discrete data sets.

We now develop the learning algorithm in the following. Most BP based learning algorithms for FNN can be considered as finding zeros of $\partial J/\partial \phi$ which correspond to their local as well as global minima. The search performance of this class of learning algorithms somehow relies on initial weights and, oftentimes, it traps into local minima. To investigate the convergence issue and develop the general algorithm, we propose the following Lyapunov function with respect to J and $\partial J/\partial \phi$:

$$V(J, \phi) = \mu J + \frac{1}{2}\sigma \left\| \frac{\partial J}{\partial \phi} \right\|^2 \tag{8}$$

where $\| \bullet \|$ is the Euclidean norm, the parameters $\mu, \sigma > 0$ determine the relative importance of each term and

$$\left\| \frac{\partial J}{\partial \phi} \right\|^2 = \left(\frac{\partial J}{\partial \phi} \right) \left(\frac{\partial J}{\partial \phi^T} \right) \tag{9}$$

with

$$\frac{\partial J}{\partial \phi} = \left(\frac{\partial J}{\partial \phi_1}, \ldots, \frac{\partial J}{\partial \phi_l} \right)$$

being the gradient represented in a row vector form [7]. For convenience, we also denote

$$\frac{\partial J}{\partial \phi^T} = \left(\frac{\partial J}{\partial \phi_1}, \ldots, \frac{\partial J}{\partial \phi_l} \right)^T.$$

Note that the function (8) is locally positive definite with respect to J and $\partial J/\partial \phi$ (at least around each local/global minimum).

The learning of the weights vector ϕ is considered as a "control" to be determined to minimize the function $V(J, \phi)$. One can easily see that if such a "control" can be found, which minimizes $V(J, \phi)$, that is, the locally positive definite function V with respect to J and $\partial J/\partial \phi$ is minimized, then

$$J = 0 \quad \text{and} \quad \frac{\partial J}{\partial \phi} = 0. \tag{10}$$

Note that (10) does not necessarily guarantee the uniqueness of the global minimal solution, but rather corresponds to a set of solutions which can make (10) hold. Therefore we can say that the optimal learning is accomplished. The question is whether (10) can be realized in theory. In the following, we will first establish the general learning algorithm for FNN. We then show that for BP-based learning algorithms, trapping into local minima is inherent.

Theorem: For a FNN structure whose input–output relationship is $y(t) = f(\phi(t), x(t))$, $\partial J/\partial \phi$ tends to zero asymptotically if

a) $\partial J/\partial t \le 0$, and
b) The weights adaptation is shown in (11) at the bottom of the page where $\varsigma, \eta > 0$ and I_l is an $l \times l$ identity matrix.

$$\frac{d\phi}{dt} = \begin{cases} -\left(\mu I_l + \sigma \dfrac{\partial^2 J}{\partial \phi \partial \phi^T} \right)^{-1} \left(\dfrac{\dfrac{\partial J}{\partial \phi^T}}{\left\| \dfrac{\partial J}{\partial \phi} \right\|^2} \right) \left(\mu \dfrac{\partial J}{\partial t} + \sigma \dfrac{\partial J}{\partial \phi} \dfrac{\partial^2 J}{\partial t \partial \phi^T} + \varsigma \left\| \dfrac{\partial J}{\partial \phi} \right\|^2 + \eta J \right) & \text{if } \left\| \dfrac{\partial J}{\partial \phi} \right\| \neq 0 \\ 0 & \text{Otherwise} \end{cases} \tag{11}$$

Proof: Clearly, the proposed function (8) is locally positive definite (around each local/global minimum) with respect to J and $\partial J/\partial \phi$. The equilibria of $V(J, \phi)$ are

$$J = 0 \quad \text{and} \quad \frac{\partial J}{\partial \phi} = 0 \tag{12}$$

which, if achieved, correspond to one of the global minima, ϕ^*, such that $J(\phi^*(t), x(t), y(t)) = 0$ and $\partial J/\partial \phi|_{\phi=\phi^*} = 0$. According to the Lyapunov stability theory for nonautonomous systems [7], [9], for the locally positive definite function $V(J, \phi)$, if in the neighborhood around ϕ^* the time derivative of $V(J, \phi)$, $\dot{V}$, is seminegative definite, then the equilibrium point ϕ^* is stable in the sense of Lyapunov. If $\dot{V}$ is negative definite, then the equilibrium point ϕ^* is asymptotically stable in the sense of Lyapunov. Now we check negative definiteness of $\dot{V}$. Differentiating (J, ϕ) with respect to time yields

$$\begin{aligned}\dot{V} &= \mu \left[\frac{\partial J}{\partial \phi}\dot{\phi} + \frac{\partial J}{\partial x}\dot{x} + \frac{\partial J}{\partial t}\right] + \sigma \frac{\partial J}{\partial \phi} \\ &\quad \cdot \left(\frac{\partial^2 J}{\partial t \partial \phi^T} + \frac{\partial^2 J}{\partial \phi \partial \phi^T}\dot{\phi} + \frac{\partial^2 J}{\partial x \partial \phi^T}\dot{x}\right) \\ &= \left(\mu \frac{\partial J}{\partial \phi} + \sigma \frac{\partial J}{\partial \phi}\frac{\partial^2 J}{\partial \phi \partial \phi^T}\right)\dot{\phi} + \mu \frac{\partial J}{\partial t} \\ &\quad + \mu \frac{\partial J}{\partial x}\dot{x} + \sigma \frac{\partial J}{\partial \phi}\frac{\partial^2 J}{\partial t \partial \phi^T} + \sigma \frac{\partial J}{\partial \phi}\frac{\partial^2 J}{\partial x \partial \phi^T}\dot{x} \\ &= \frac{\partial J}{\partial \phi}\left(\mu I_l + \sigma \frac{\partial^2 J}{\partial \phi \partial \phi^T}\right)\dot{\phi} + \mu \frac{\partial J}{\partial t} + \mu \frac{\partial J}{\partial x}\dot{x} \\ &\quad + \sigma \frac{\partial J}{\partial \phi}\frac{\partial^2 J}{\partial t \partial \phi^T} + \sigma \frac{\partial J}{\partial \phi}\frac{\partial^2 J}{\partial x \partial \phi^T}\dot{x}.\end{aligned} \tag{13}$$

Since J is an implicit function of $x(t)$, then $\partial J/\partial x = 0$ and $\partial^2 J/\partial x \partial \phi^T = 0$. Hence (13) becomes

$$\dot{V} = \frac{\partial J}{\partial \phi}\left(\mu I_l + \sigma \frac{\partial^2 J}{\partial \phi \partial \phi^T}\right)\dot{\phi} + \mu \frac{\partial J}{\partial t} + \sigma \frac{\partial J}{\partial \phi}\frac{\partial^2 J}{\partial t \partial \phi^T}. \tag{14}$$

It is sufficient that if the adaptation law (11) is chosen, then

$$\dot{V} = \begin{cases} -\varsigma \left\|\frac{\partial J}{\partial \phi}\right\|^2 - \eta J^2 < 0 & \text{if } \left\|\frac{\partial J}{\partial \phi}\right\| \neq 0 \\ \mu \frac{\partial J}{\partial t} & \text{Otherwise.} \end{cases} \tag{15}$$

From (15), it can be seen that if $\|\partial J/\partial \phi\| = 0$, then $\partial J/\partial \phi = 0$, which means ϕ reaches a minimum which may not be a global minimum, then $\dot{V} = \mu(\partial J/\partial t)$. Therefore if $\partial J/\partial t \leq 0$ then $\dot{V} \leq 0$ at $\partial J/\partial \phi = 0$, which will ensure the convergence trend toward ϕ^*. On the other hand, if $\|\partial J/\partial \phi\| \neq 0$, $\dot{V} = -\varsigma\|\partial J/\partial \phi\|^2 - \eta J^2 < 0$, which is negative definite with respect to J and $\partial J/\partial \phi$ at least locally. This means that $\|\partial J/\partial \phi\| \to 0$ asymptotically but J may not tend to zero even when $\dot{V} = -\varsigma\|\partial J/\partial \phi\|^2 - \eta J^2$ is negative definite with respect to J and $\partial J/\partial \phi$, since this negative definiteness is conditional to $\|\partial J/\partial \phi\| \neq 0$. Once $\|\partial J/\partial \phi\| \to 0$, from (15), $\dot{V} = \mu(\partial J/\partial t)$ and the learning stops. Hence there is no mechanism to further decrease J. The above analysis suggests that the conditions a) and b) are only sufficient for $\partial J/\partial \phi$ to tend to zero asymptotically, but not for J to converge to zero asymptotically. Due to the inherent association of $\partial J/\partial \phi$ and $\dot{\phi}$ with $\dot{V}$, only the stability (not the asymptotical stability) around ϕ^* in the sense of Lyapunov can be guaranteed. QED

Remark: The theorem demonstrates that any BP learning algorithm cannot be guaranteed to reach a global minimum because the term $\partial J/\partial \phi$ is inherently associated with the learning law $\dot{\phi}$ as shown in the first term of (14). Any learning will stop once $\partial J/\partial \phi$ becomes zero. It is also shown that for time varying inputs, the condition $\partial J/\partial t \leq 0$ has to be satisfied, especially when $\partial J/\partial \phi = 0$ so that the convergence trend of J can be maintained. It is interesting to note that since $\partial J/\partial t = (1/\tau)(e(t) - e(t-\tau))$, then the condition $\partial J/\partial t \leq 0$ is equivalent to $e(t) \leq e(t-\tau)$ which gives a constraint for the convergence of weights for the time varying inputs when learning is stopped.

The theorem can interpret many existing BP learning algorithms. For example, a common FNN learning task is to train FNN with discrete input data sets. In this case, we have

$$\frac{\partial J}{\partial t} = 0, \qquad \frac{\partial^2 J}{\partial t \partial \phi^T} = 0.$$

Since the learning is in discrete time, the learning algorithm is then generally expressed as

$$\phi(k+1) = \phi(k) - \Delta t \dot{\phi}$$

where the term $\dot{\phi}$ acts as a "gradient" and the approximation $\dot{\phi} = (\phi(k+1) - \phi(k))/\Delta t$ is used with Δt being the sampling time interval. We now demonstrate how to derive several classes of commonly used BP learning algorithms.

The Conventional Gradient Descent Learning Algorithm: The conventional gradient descent learning algorithm can be easily obtained by setting $\sigma = 0$ and $\eta = 0$. Since $\partial^2 J_k/\partial t \partial \phi^T = 0$, then from (11) we have

$$\dot{\phi} = -\mu^{-1}\varsigma \frac{\partial J_k}{\partial \phi^T} = -\lambda \frac{\partial J_k}{\partial \phi^T}, \qquad \lambda = \mu^{-1}\varsigma.$$

The Gauss–Newton Algorithm: The Gauss–Newton algorithm can be obtained by setting $\mu = 0$, $\eta = 0$. Since $\partial J_k/\partial t = 0$ and $\partial^2 J_k/\partial t \partial \phi^T = 0$, then from (11) we have

$$\begin{aligned}\dot{\phi} &= -\left(\sigma \frac{\partial^2 J_k}{\partial \phi \partial \phi^T}\right)^{-1}\left(\varsigma \frac{\partial J_k}{\partial \phi^T}\right) \\ &= -\lambda \left(\frac{\partial^2 J_k}{\partial \phi \partial \phi^T}\right)^{-1}\left(\frac{\partial J_k}{\partial \phi^T}\right) \qquad \text{with } \lambda = \sigma^{-1}\varsigma.\end{aligned}$$

The Levenberg–Marquardt Algorithm: The Levenberg–Marquardt algorithm can be easily obtained by setting $\eta = 0$. Since $\partial J_k/\partial t = 0$ and $\partial^2 J_k/\partial t \partial \phi^T = 0$, then from (11) we have

$$\dot{\phi} = -\left(\mu + \sigma \frac{\partial^2 J_k}{\partial \phi \partial \phi^T}\right)^{-1}\left(\varsigma \frac{\partial J_k}{\partial \phi^T}\right).$$

An On Line Learning BP Algorithm for Time Varying Inputs: In [3], an on line learning BP algorithm for time varying inputs was proposed. This algorithm can be easily derived by setting $\eta = 0$ and $\mu = 0$, which gives rise to exponentially convergent learning.

III. Conclusion

A general FNN training algorithm has been proposed. Its convergence has been completely analyzed using the Lyapunov stability theory. It has been shown that the proposed algorithm covers major classes of commonly used BP learning algorithms. However, it should be emphasized that the strength of the general learning algorithm lies in its ability to handle time varying inputs. Sufficient conditions for the convergence of FNN weights have been given.

References

[1] J. M. Zurada, *Introduction to Artificial Neural Systems*. St. Paul, MN: West, 1992.

[2] P. Mehra and B. W. Wah, *Artificial Neural Networks: Concepts and Theory*: IEEE Comput. Society Press, 1992.

[3] Y. Zhao, "On-line neural network learning algorithm with exponential convergence rate," *Electron. Lett.*, vol. 32, no. 15, pp. 1381–1382, July 1996.

[4] G. Zhou and J. Si, "Advanced neural network training algorithm with reduced complexity based on Jacobian deficiency," *IEEE Trans. Neural Networks*, vol. 9, pp. 448–453, May 1998.

[5] R. Parisi, E. D. Di Claudio, G. Orlandi, and B. D. Rao, "A generalized learning paradigm exploiting the structure of feedforward neural networks," *IEEE Trans. Neural Networks*, vol. 7, pp. 1450–1459, Nov. 1996.

[6] M. T. Hagan and M. B. Menhaj, "Training feedforward neural networks with the Marquardt algorithm," *IEEE Trans. Neural Networks*, vol. 5, pp. 989–993, Nov. 1994.

[7] J.-J. Slotine and W. Li, *Applied Nonlinear Control*. Englewood Cliffs, NJ: Prentice-Hall, 1991.

[8] H. Bersini and V. Gorrini, "A simplification of the backpropagation through time algorithm for optimal neurocontroller," *IEEE Trans. Neural Networks*, vol. 8, pp. 437–441, Mar. 1997.

[9] M. Krstic, I. Kanellakopoulos, and P. Kokotovic, *Nonlinear and Adaptive Control Design*. New York: Wiley, 1995.

[10] Z. Artstein, "Stabilization with relaxed controls," *Nonlinear Anal.*, vol. TMA-7, pp. 1163–1173, 1983.

[11] X. B. Liang and J. Wang, "Absolute exponential stability of neural networks with a general class of activation functions," *IEEE Trans. Circuits Syst. Part I*, vol. 47, pp. 1258–1263, 2000.

[12] Z. Guan, G. Chen, and Y. Yin, "On equilibra, stability, and instability of Hopfield neural networks," *IEEE Trans. Neural Networks*, vol. 11, pp. 534–540, 2000.

Errata to "Learning Efficiency of Redundant Neural Networks in Bayesian Estimation"

S. Watanabe

***Abstract*—This paper proves that the Bayesian stochastic complexity of a layered neural network is asymptotically smaller than that of a regular statistical model if it contains the true distribution. We consider a case when a three-layer perceptron with M input units, H hidden units and N output units is trained to estimate the true distribution represented by the model with H_0 hidden units and prove that the stochastic complexity is asymptotically smaller than $(1/2)\{H_0(M+N)+R\}\log n$ where n is the number of training samples and R is a function of $H-H_0$, M, and N that is far smaller than the number of redundant parameters. Since the generalization error of Bayesian estimation is equal to the increase of stochastic complexity, it is smaller than $(1/2n)\{H_0(M+N)+R\}$ if it has an asymptotic expansion. Based on the results, the difference between layered neural networks and regular statistical models is discussed from the statistical point of view.**

In the above paper,[1] the manuscript received and revised dates, as well as the author's biography, were inadvertently omitted. They are as follows. Manuscript received September 3, 1999; revised February 26, 2001.

Sumio Watanabe (M'90) was born in Japan in 1959. He received the B. S. degree in physics in 1982 from the University of Tokyo, Tokyo, Japan, the M.S. degree in mathematics in 1984 from Kyoto University, Kyoto, Japan, and the Ph.D. degree in applied electronics in 1993 from Tokyo Institute of Technology, Tokyo, Japan.

He is currently an Associate Professor at the Precision and Intellegence Laboratory, Tokyo Institute of Technology. His research interests include probability theory, mathematical statistics, and neural-network learning theory.

Dr. Watanabe is a member of IEICE and the Japanese Neural Network Society.

Publisher Item Identifier S 1045-9227(02)01468-6

[1]S. Watanabe, *IEEE Trans. Neural Networks*, vol. 12, pp. 1475–1486, Nov. 2001.