Analysis of ensemble learning using simple perceptrons based on online learning theory

Seiji Miyoshi,[1,*] Kazuyuki Hara,[2] and Masato Okada[3]

[1]*Department of Electronic Engineering, Kobe City College of Technology, Gakuenhigashi-machi 8-3, Nishi-ku, Kobe 651-2194, Japan*

[2]*Department of Electronics and Information Engineering, Tokyo Metropolitan College of Technology, Higashi-oi 1-10-40, Shinagawa-ku, Tokyo, 140-0011 Japan*

[3]*Department of Complexity Science and Engineering, Division of Transdisciplinary Sciences, Graduate School of Frontier Sciences, The University of Tokyo, 5-1-5, Kashiwanoha, Kashiwa-shi, Chiba, 277-8562 Japan; Laboratory for Mathematical Neuroscience, RIKEN Brain Science Institute, Hirosawa 2-1, Wako, Saitama, 351-0198 Japan; and Intelligent Cooperation and Control, PRESTO, Japan Science and Technology Agency, Hirosawa 2-1, Wako, Saitama, 351-0198 Japan*

(Received 27 May 2004; published 15 March 2005)

Ensemble learning of K nonlinear perceptrons, which determine their outputs by sign functions, is discussed within the framework of online learning and statistical mechanics. One purpose of statistical learning theory is to theoretically obtain the generalization error. This paper shows that ensemble generalization error can be calculated by using two order parameters, that is, the similarity between a teacher and a student, and the similarity among students. The differential equations that describe the dynamical behaviors of these order parameters are derived in the case of general learning rules. The concrete forms of these differential equations are derived analytically in the cases of three well-known rules: Hebbian learning, perceptron learning, and AdaTron (adaptive perceptron) learning. Ensemble generalization errors of these three rules are calculated by using the results determined by solving their differential equations. As a result, these three rules show different characteristics in their affinity for ensemble learning, that is "maintaining variety among students." Results show that AdaTron learning is superior to the other two rules with respect to that affinity.

DOI: 10.1103/PhysRevE.71.036116 PACS number(s): 05.90.+m, 87.10.+e, 05.20.Gg

I. INTRODUCTION

Ensemble learning has recently attracted the attention of many researchers [1–6]. Ensemble learning means to combine many rules or learning machines (students in the following) that perform poorly. Theoretical studies analyzing the generalization performance by using statistical mechanics [7,8] have been performed vigorously [4–6].

Hara and Okada [4] theoretically analyzed the case in which students are linear perceptrons. Their analysis was performed with statistical mechanics, focusing on the fact that the output of a new perceptron, whose connection weight is equivalent to the mean of those of students, is identical to the mean outputs of students. Krogh and Sollich [5] analyzed ensemble learning of linear perceptrons with noises within the framework of batch learning. They showed that the generalization performance can be optimized by choosing the best size of learning samples for a large K limit, where K is the number of students, and that the generalization performance can be improved by dividing learning samples in the noisy situation when K is finite.

On the other hand, Hebbian learning, perceptron learning, and AdaTron (adaptive perceptron) learning are well known as learning rules for a nonlinear perceptron, which decides its output by sign function [9–12]. Urbanczik [6] analyzed ensemble learning of nonlinear perceptrons that decide their outputs by sign functions for a large K limit within the framework of online learning [13]. He treated a generalized learning rule that he termed a "soft version of perceptron learning," which includes both Hebbian learning and perceptron learning as special cases, and discussed it from the viewpoint of generalization error. As a result, he showed that though an ensemble usually has superior performance to a single student, an ensemble has no special advantage in the optimized case within the framework of the soft version of perceptron learning.

Though Urbanczik discussed ensemble learning of nonlinear perceptrons within the framework of online learning, he treated only the case in which the number K of students is large enough. Determining differences among ensemble learnings with Hebbian learning, perceptron learning, and AdaTron learning (three typical learning rules) is a very attractive problem.

Based on the past studies, we discuss ensemble learning of K nonlinear perceptrons, which decide their outputs by sign functions within the framework of online learning and finite K [14,15]. First, we show that an ensemble generalization error of K students can be calculated by using two order parameters: one is a similarity between a teacher and a student, the other is a similarity among students. Next, we derive differential equations that describe dynamical behaviors of these order parameters in the case of general learning rules. After that, we derive concrete differential equations about three well-known learning rules: Hebbian learning, perceptron learning, and AdaTron learning. We calculate the ensemble generalization errors by using results obtained through solving these equations numerically. Two methods are treated to decide an ensemble output. One is the majority vote of students, and the other is an output of a new perceptron whose connection weight equals the mean of those of

*Electronic address: miyoshi@kobe-kosen.ac.jp

students. As a result, we show that these three learning rules have different properties with respect to an affinity for ensemble learning, and AdaTron learning, which is known to have the best asymptotic property [9–12], gives the largest improvement by ensemble among the three learning rules.

II. MODEL

Each student treated in this paper is a perceptron that decides its output by a sign function. An ensemble of K students is considered. Connection weights of students are $\boldsymbol{J}_1, \boldsymbol{J}_2, \ldots, \boldsymbol{J}_K$. $\boldsymbol{J}_k = (J_{k1}, \ldots, J_{kN})^T$, $k = 1, 2, \ldots, K$, and input $\boldsymbol{x} = (x_1, \ldots, x_N)^T$ are N dimensional vectors. Each component x_i of $\boldsymbol{x}$ is assumed to be an independent random variable that obeys the Gaussian distribution $\mathcal{N}(0, 1/N)$. Each component of $\boldsymbol{J}_k^0$, that is the initial value of $\boldsymbol{J}_k$, is assumed to be generated according to the Gaussian distribution $\mathcal{N}(0,1)$ independently. Thus

$$\langle x_i \rangle = 0, \langle (x_i)^2 \rangle = \frac{1}{N}, \tag{1}$$

$$\langle J_{ki}^0 \rangle = 0, \langle (J_{ki}^0)^2 \rangle = 1, \tag{2}$$

where $\langle \cdot \rangle$ denotes the average. Each student's output is $\mathrm{sgn}(u_1 l_1), \mathrm{sgn}(u_2 l_2), \ldots, \mathrm{sgn}(u_K l_K)$ where

$$\mathrm{sgn}(ul) = \begin{cases} +1, & ul \geqslant 0, \\ -1, & ul < 0, \end{cases} \tag{3}$$

$$u_k l_k = \boldsymbol{J}_k \cdot \boldsymbol{x}. \tag{4}$$

Here, l_k denotes the length of student $\boldsymbol{J}_k$. This is one of the order parameters treated in this paper and will be described in detail later. In this paper, u_k is called a normalized internal potential of a student.

The teacher is also perceptron that decides its output by a sign function. The teacher's connection weight is $\boldsymbol{B}$. In this paper, $\boldsymbol{B}$ is assumed to be fixed where $\boldsymbol{B} = (B_1, \ldots, B_N)^T$ is also an N-dimensional vector. Each component B_i is assumed to be generated according to the Gaussian distribution $\mathcal{N}(0,1)$ independently. Thus

$$\langle B_i \rangle = 0, \quad \langle (B_i)^2 \rangle = 1. \tag{5}$$

The teacher's output is $\mathrm{sgn}(v)$ where

$$v = \boldsymbol{B} \cdot \boldsymbol{x}. \tag{6}$$

Here, v represents an internal potential of the teacher. For simplicity, the connection weight of a student and that of the teacher are simply called student and teacher, respectively.

In this paper the thermodynamic limit $N \to \infty$ is also treated. Therefore

$$|\boldsymbol{x}| = 1, \quad |\boldsymbol{B}| = \sqrt{N}, \quad |\boldsymbol{J}_k^0| = \sqrt{N}, \tag{7}$$

where $|\cdot|$ denotes a vector norm. Generally, a norm of student $|\boldsymbol{J}_k|$ changes as the time step proceeds. Therefore the ratio l_k of the norm to $\sqrt{N}$ is considered and is called a length of student $\boldsymbol{J}_k$. That is,

$$|\boldsymbol{J}_k| = l_k \sqrt{N}, \tag{8}$$

where l_k is one of the order parameters treated in this paper.

The common input $\boldsymbol{x}$ is presented to the teacher and all students in the same order.Within the framework of online learning, the update can be expressed as follows:

$$\boldsymbol{J}_k^{m+1} = \boldsymbol{J}_k^m + f_k^m \boldsymbol{x}^m, \tag{9}$$

$$f_k^m = f(\mathrm{sgn}(v^m), u_k^m), \tag{10}$$

where m denotes time step, and f is a function determined by learning rule.

In this paper, two methods are treated to determine an ensemble output. One is the majority vote of K students, which means an ensemble output is decided to be $+1$ if students whose outputs are $+1$ exceed the number of students whose outputs are -1, and -1 in the opposite case.

Another method for deciding an ensemble output is adopting an output of a new perceptron whose connection weight is the mean of the weights of K students. This method is simply called the weight mean in this paper.

III. THEORY

In this paper, the majority vote and the weight mean are treated to determine an ensemble output.We use

$$\epsilon = \Theta\left(-\boldsymbol{B} \cdot \boldsymbol{x} \sum_{k=1}^{K} \mathrm{sgn}(\boldsymbol{J}_k \cdot \boldsymbol{x}) \right), \tag{11}$$

and

$$\epsilon = \Theta\left(-\boldsymbol{B} \cdot \boldsymbol{x} \left(\sum_{k=1}^{K} \boldsymbol{J}_k \right) \cdot \boldsymbol{x} \right) \tag{12}$$

as error ϵ for the majority vote and the weight mean, respectively. Here, $\Theta(z) = 1$ for $z > 0$ and 0 otherwise. ϵ, $\boldsymbol{x}$ and $\boldsymbol{J}_k$ denote ϵ^m, $\boldsymbol{x}^m$, and $\boldsymbol{J}_k^m$, respectively. However, superscripts m, which represent time steps, are omitted for simplicity.

Generalization error ϵ_g is defined as the average of error ϵ over the probability distribution $p(\boldsymbol{x})$ of input $\boldsymbol{x}$. The generalization error ϵ_g can be regarded as the probability that an ensemble output disagrees with that of the teacher for a new input $\boldsymbol{x}$. One purpose of statistical learning theory is to theoretically obtain generalization error. In the case of a majority vote, using Eqs. (4), (6), and (11), we obtain

$$\epsilon = \Theta\left(-v \sum_{k=1}^{K} \mathrm{sgn}(u_k) \right). \tag{13}$$

In the case of a weight mean, using Eqs. (4), (6), and (12), we obtain

$$\epsilon = \Theta\left(-v \sum_{k=1}^{K} u_k \right). \tag{14}$$

That is, error ϵ can be described as $\epsilon = \epsilon(\{u_k\}, v)$ by using a normalized internal potential u_k for the student and an internal potential v for the teacher in both cases. Therefore the

generalization error ϵ_g can be also described as

$$\epsilon_g = \int dx p(x)\epsilon = \int \prod_{k=1}^{K} du_k dv p(\{u_k\},v)\epsilon(\{u_k\},v) \quad (15)$$

by using the probability distribution $p(\{u_k\},v)$ of u_k and v. Since the thermodynamic limit $N\to\infty$ is also considered in this paper, u_k and v obey the multiple Gaussian distribution based on the central limit theorem. The discussion in this paper falls within the framework of online learning. Therefore since an input $\boldsymbol{x}$ and a student $\boldsymbol{J}_k$ have no correlation with each other, from Eq. (4), the mean and the variance of u_k are

$$\langle u_k\rangle = 0, \quad \langle (u_k)^2\rangle = 1, \quad (16)$$

respectively. In the same manner, since an input $\boldsymbol{x}$ and a teacher $\boldsymbol{B}$ have no correlation with each other, from Eq. (6), the mean and the variance of v are

$$\langle v\rangle = 0, \quad \langle v^2\rangle = 1, \quad (17)$$

respectively. From these, all diagonal components of the covariance matrix $\boldsymbol{\Sigma}$ of $p(\{u_k\},v)$ equal unity.

Let us discuss a direction cosine between connection weights as preparation for obtaining nondiagonal components. First, R_k is defined as a direction cosine between a teacher $\boldsymbol{B}$ and a student $\boldsymbol{J}_k$. That is,

$$R_k \equiv \frac{\boldsymbol{B}\cdot\boldsymbol{J}_k}{|\boldsymbol{B}||\boldsymbol{J}_k|} = \frac{1}{l_k N}\sum_{i=1}^{N} B_i J_{ki}. \quad (18)$$

R_k is called the similarity (overlap in other words) between teacher and student in the following. R_k is the second order parameter treated in this paper. Next, $q_{kk'}$ is defined as a direction cosine between a student $\boldsymbol{J}_k$ and another student $\boldsymbol{J}_{k'}$. That is,

$$q_{kk'} \equiv \frac{\boldsymbol{J}_k\cdot\boldsymbol{J}_{k'}}{|\boldsymbol{J}_k||\boldsymbol{J}_{k'}|} = \frac{1}{l_k l_{k'} N}\sum_{i=1}^{N} J_{ki}J_{k'i}, \quad (19)$$

where $k\neq k'$. $q_{kk'}$ is called the similarity among students in the following, and $q_{kk'}$ is the third order parameter treated in this paper.

Covariance between an internal potential v of a teacher $\boldsymbol{B}$ and a normalized internal potential u_k of a student $\boldsymbol{J}_k$ equals a similarity R_k between a teacher $\boldsymbol{B}$ and a student $\boldsymbol{J}_k$ as follows:

$$\langle v u_k\rangle = \left\langle \frac{1}{l_k}\sum_{i=1}^{N} B_i x_i \sum_{j=1}^{N} J_{kj}x_j \right\rangle = R_k. \quad (20)$$

Covariance between a normalized internal potential u_k of a student $\boldsymbol{J}_k$ and a normalized internal potential $u_{k'}$ of another student $J_{k'}$ equals a similarity $q_{kk'}$ among students as follows:

$$\langle u_k u_{k'}\rangle = \left\langle \frac{1}{l_k l_{k'}}\sum_{i=1}^{N} J_{ki}x_i \sum_{j=1}^{N} J_{k'j}x_j \right\rangle = q_{kk'}. \quad (21)$$

Therefore Eq. (15) can be rewritten as

$$\epsilon_g = \int \prod_{k=1}^{K} \mathrm{d}u_k dv p(\{u_k\},v)\epsilon(\{u_k\},v), \quad (22)$$

$$p(\{u_k\},v) = \frac{1}{(2\pi)^{(k+1)/2}|\boldsymbol{\Sigma}|^{1/2}}\exp\left(-\frac{(\{u_k\},v)\boldsymbol{\Sigma}^{-1}(\{u_k\},v)^T}{2}\right), \quad (23)$$

$$\boldsymbol{\Sigma} = \begin{pmatrix} 1 & q_{12} & \cdots & q_{1K} & R_1 \\ q_{21} & 1 & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & q_{K-1,K} & \vdots \\ q_{K1} & \cdots & q_{K,K-1} & 1 & R_K \\ R_1 & \cdots & \cdots & R_K & 1 \end{pmatrix}. \quad (24)$$

As a result, a generalization error ϵ_g can be calculated if all similarities R_k and $q_{kk'}$ are obtained. Let us thus discuss differential equations that describe dynamical behaviors of these order parameters. Differential equations regarding l_k and R_k for general learning rules have been obtained based on self-averaging as follows [9]:

$$\frac{dl_k}{dt} = \langle f_k u_k\rangle + \frac{\langle f_k^2\rangle}{2l_k}, \quad (25)$$

$$\frac{dR_k}{dt} = \frac{\langle f_k v\rangle - \langle f_k u_k\rangle R_k}{l_k} - \frac{R_k}{2l_k^2}\langle f_k^2\rangle, \quad (26)$$

where $\langle\cdot\rangle$ stands for the sample average. That is,

$$\langle f_k u_k\rangle = \int du_k dv p_2(u_k,v) f(\mathrm{sgn}(v),u_k)u_k, \quad (27)$$

$$\langle f_k v\rangle = \int du_k dv p_2(u_k,v) f(\mathrm{sgn}(v),u_k)v, \quad (28)$$

$$\langle f_k^2\rangle = \int du dv p_2(u_k,v)[f(\mathrm{sgn}(v),u_k)]^2, \quad (29)$$

$$p_2(u_k,v) = \frac{1}{2\pi|\boldsymbol{\Sigma}_2|^{1/2}}\exp\left(-\frac{(u_k,v)\boldsymbol{\Sigma}_2^{-1}(u_k,v)^T}{2}\right), \quad (30)$$

$$\boldsymbol{\Sigma}_2 = \begin{pmatrix} 1 & R_k \\ R_k & 1 \end{pmatrix}. \quad (31)$$

Next, let us derive a differential equation regarding $q_{kk'}$ for the general learning rule. Considering a student $\boldsymbol{J}_k$ and another student $\boldsymbol{J}_{k'}$ and rewriting as $l_k^m\to l_k$, $l_k^{m+1}\to l_k+dl_k$, $q_{kk'}^m\to q_{kk'}$, $q_{kk'}^{m+1}\to q_{kk'}+dq_{kk'}$, and $1/N\to dt$, a differential equation regarding q is obtained as follows [4]:

$$\frac{dq_{kk'}}{dt} = \frac{\langle f_{k'}u_k\rangle - q_{kk'}\langle f_{k'}u_{k'}\rangle}{l_{k'}} + \frac{\langle f_k u_{k'}\rangle - q_{kk'}\langle f_k u_k\rangle}{l_k} + \frac{\langle f_k f_{k'}\rangle}{l_k l_{k'}} - \frac{q_{kk'}}{2}\left(\frac{\langle f_k^2\rangle}{l_k^2} + \frac{\langle f_{k'}^2\rangle}{l_{k'}^2}\right), \quad (32)$$

from Eqs. (9), (19), and (25) and self-averaging, where

$$\langle f_k u_{k'} \rangle = \int du_k du_{k'} dv p_3(u_k, u_{k'}, v) f(\mathrm{sgn}(v), u_k) u_{k'}, \quad (33)$$

$$\langle f_{k'} u_k \rangle = \int du_k du_{k'} dv p_3(u_k, u_{k'}, v) f(\mathrm{sgn}(v), u_{k'}) u_k, \quad (34)$$

$$\langle f_k f_{k'} \rangle = \int du_k du_{k'} dv p_3(u_k, u_{k'}, v) f(\mathrm{sgn}(v), u_k) f(\mathrm{sgn}(v), u_{k'}), \quad (35)$$

$$p_3(u_k, u_{k'}, v) = \frac{1}{(2\pi)^{3/2} |\mathbf{\Sigma}_3|^{1/2}} \times \exp\left(- \frac{(u_k, u_{k'}, v) \mathbf{\Sigma}_3^{-1} (u_k, u_{k'}, v)^T}{2} \right), \quad (36)$$

$$\mathbf{\Sigma}_3 = \begin{pmatrix} 1 & q_{kk'} & R_k \\ q_{k'k} & 1 & R_{k'} \\ R_k & R_{k'} & 1 \end{pmatrix}. \quad (37)$$

IV. RESULT

A. Conditions of analytical calculations

As described above, in this paper each component of initial value $\boldsymbol{J}_k^0$ of student $\boldsymbol{J}_k$ and teacher $\boldsymbol{B}$ is generated independently according to the Gaussian distribution $\mathcal{N}(0,1)$, and the thermodynamic limit $N \to \infty$ is considered. Therefore all $\boldsymbol{J}_k^0$ and $\boldsymbol{B}$ are orthogonal to each other. That is,

$$R_k^0 = 0, \quad q_{kk'}^0 = 0. \quad (38)$$

From Eq. (38) and symmetry of students, we can write

$$\langle f_k u_{k'} \rangle = \langle f_{k'} u_k \rangle, \quad \langle f_k f_{k'} \rangle = \langle f_{k'} f_k \rangle \quad (39)$$

in Eq. (32). From Eq. (38) and symmetry among students, we omit subscripts k, k' from order parameters l_k, R_k, and $q_{kk'}$ in Eqs. (25)–(37) and write them as l,R, and q. In the following sections, we analytically obtain five sample averages $\langle f_k u_k \rangle$, $\langle f_k v \rangle$, $\langle f_k^2 \rangle$, $\langle f_k u_{k'} \rangle$, and $\langle f_k f_{k'} \rangle$ concretely, which are necessary to solve Eqs. (25)–(37) with respect to typical learning rules under the conditions given in Eqs. (38) and (39). R and q are obtained by solving the above sample averages and Eqs. (25), (26), (32), and (38) numerically. We obtain numerical ensemble generalization errors ϵ_g by calculating Eq. (22) with the obtained R and q.

B. Hebbian learning

The update procedure for Hebbian learning is

$$f(\mathrm{sgn}(v), u) = \mathrm{sgn}(v). \quad (40)$$

Using this expression, $\langle f_k u_k \rangle$, $\langle f_k v \rangle$, and $\langle f_k^2 \rangle$ in the case of Hebbian learning can be obtained by executing Eqs. (27)–(29) analytically [9,17] (see Appendix A). In addition to these results, we have derived $\langle f_k u_{k'} \rangle$ and $\langle f_k f_{k'} \rangle$ (see Appendix A).

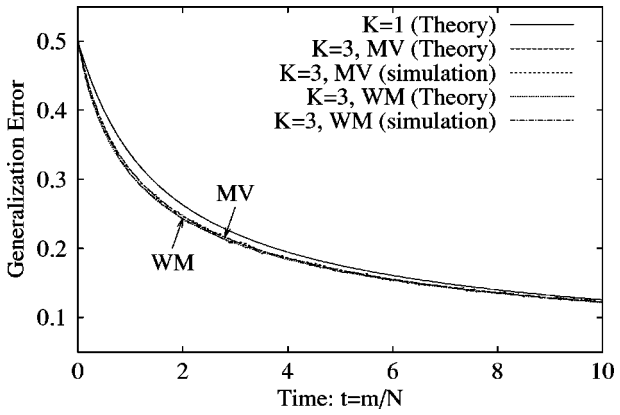

FIG. 1. Dynamical behaviors of ensemble generalization error ϵ_g in Hebbian learning.

R and q have been obtained by solving Eqs. (25), (26), (32), (38), and (39) and the derived sample averages numerically. We have obtained numerical ensemble generalization errors ϵ_g in the case of $K=3$ by using Eqs. (22)–(24) and the above R and q. Figure 1 shows the results. In this figure, MV and WM indicate the majority vote and the weight mean, respectively. Numerical integrations of Eq. (22) in theoretical calculations have been executed by using the six-point closed Newton-Cotes formula. In the computer simulation, $N=10^4$ and ensemble generalization errors have been obtained through tests using 10^5 random inputs at each time step. In this figure, the result of theoretical calculations of $K=1$ is also shown to clarify the effect of the ensemble. This figure shows that the ensemble generalization errors obtained by theoretical calculation explain the computer simulation quantitatively.

Figures 2 and 3 show the results of computer simulations where $N=10^3$, $K=1$, 3, 11, 31 until $t=10^4$ in order to investigate asymptotic behaviors of generalization errors. Asymptotic behavior of generalization error in Hebbian learning in the case of the number K of students at unity is

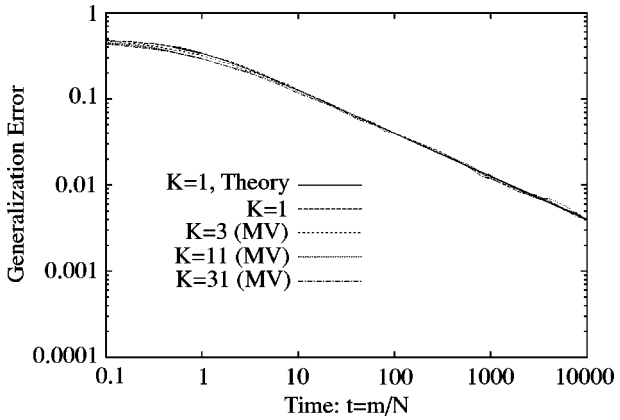

FIG. 2. Asymptotic behavior of generalization error of majority vote in Hebbian learning. Computer simulations, except for the solid line. Asymptotic order of ensemble learning is the same as that at $K=1$.

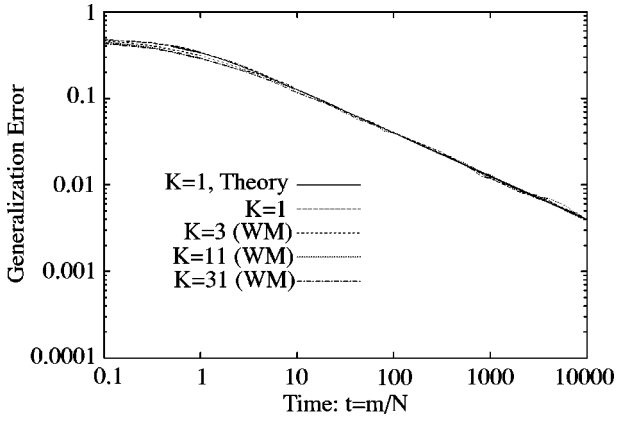

FIG. 3. Asymptotic behavior of generalization error of weight mean in Hebbian learning. Computer simulations, except for the solid line. Asymptotic order of ensemble learning is the same as that at $K=1$.

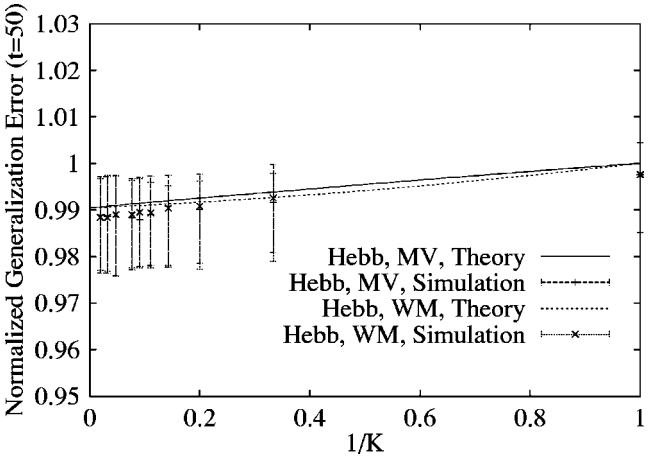

FIG. 4. Relationship between K and effect of ensemble in Hebbian learning. Ensemble generalization error ϵ_g for a large K limit is about 0.99 times that of $K=1$.

$O(t^{-1/2})$ [9]. Asymptotic orders of the generalization error in the case of ensemble learning are considered equal to those of $K=1$, since properties of $K=3$, 11, 31 are parallel to those of $K=1$ in these figures.

To clarify the relationship between K and the effect of ensemble, we have obtained theoretical ensemble generalization errors for various values of K. Here, it is difficult to execute numerical integration of Eq. (22) when $K>3$ by the Newton-Cotes formula used in the calculations for Fig. 1. Therefore the Metropolis method, which is a type of Monte Carlo method, has been used. We then orthogonalized the variables of integration to eliminate the calculation of inverse matrices of Eq. (24). That is,

$$u_k = a\bar{u}_k + b\hat{u} + cv, \quad k = 1,2, \ldots, K, \tag{41}$$

where u_k, $\bar{u}_k$, $\hat{u}$, and v obey the Gaussian distribution $\mathcal{N}(0,1)$, and $\bar{u}_k$, $\hat{u}$, and v have no correlation with each other. Considering that subscripts k,k' have been omitted from order parameters R_k, $q_{kk'}$, and Eq. (24), we obtain

$$a = \sqrt{1-q}, \quad b = \sqrt{q-R^2}, \quad c = R. \tag{42}$$

By using these a, b, and c, we can rewrite Eqs. (22)–(24) as follows:

$$\epsilon_g = \int \prod_{k=1}^{K} d\bar{u}_k p_1(\bar{u}_k) d\hat{u} p_1(\hat{u}) dv p_1(v) \epsilon(\{a\bar{u}_k + b\hat{u} + cv\}, v), \tag{43}$$

$$p_1(u) = \frac{1}{(2\pi)^{1/2}} \exp\left(-\frac{u^2}{2}\right). \tag{44}$$

These operations orthogonalized the variables of integration in exchange for their number having been increased from $K+1$ to $K+2$. The multiple Gaussian distribution function $p(\{u_k\},v)$ can be rewritten as products of simple Gaussian distribution functions $p_1(\cdot)$ by this orthogonalization. Thus calculations of inverse matrices of Eq. (24) become unnecessary. These facts have made it easy to perform the numerical calculations of the generalization error for a large K.

Figure 4 shows the results obtained by the Metropolis method using the values of R and q calculated numerically for Hebbian learning and Eqs. (42)–(44). Calculations have been executed for $K=1$, 3, 5, 7, 9, 11, 13, 21, 31, and 51 in both the majority vote (MV) and the weight mean (WM). The number of MonteCarlo steps is 10^9. These theoretical results are fitted to two quadratic curves. In this figure, the results of computer simulations where $N=10^4$, $K=1$, 3, 5, 7, 9, 11, 13, 21, 31, and 51 have also been drawn for comparison with the theoretical calculations. In the computer simulations, ensemble generalization errors have been obtained through tests using 10^6 random inputs. The figures show the values of $t=50$ for both theoretical calculations and computer simulations, and this is the time for which is considered that the learnings are sufficiently within the asymptotic regions with respect to Fig. 2 and 3. Here, since the relationship between $1/K$ and ensemble generalization errors shows a straight line [4] in the case of linear perceptrons, the abscissa is $1/K$ in Fig. 4. The ordinates have been normalized by the theoretical ensemble generalization error of $K=1$ and $t=50$.

C. Perceptron learning

The update procedure for perceptron learning is

$$f(\mathrm{sgn}(v),u) = \Theta(-uv)\mathrm{sgn}(v). \tag{45}$$

Using this expression, $\langle f_k u_k \rangle$, $\langle f_k v \rangle$, and $\langle f_k^2 \rangle$ in the case of perceptron learning can be obtained by executing Eqs. (27)–(29) analytically [9,17] (see Appendix B). In addition to these results, we have derived $\langle f_k u_{k'} \rangle$ and $\langle f_k f_{k'} \rangle$ (see Appendix B).

In the same manner as Hebbian learning, R and q have been obtained by solving Eqs. (25), (26), (32), (38), and (39), and the derived sample averages numerically. We have obtained numerical ensemble generalization errors ϵ_g in the

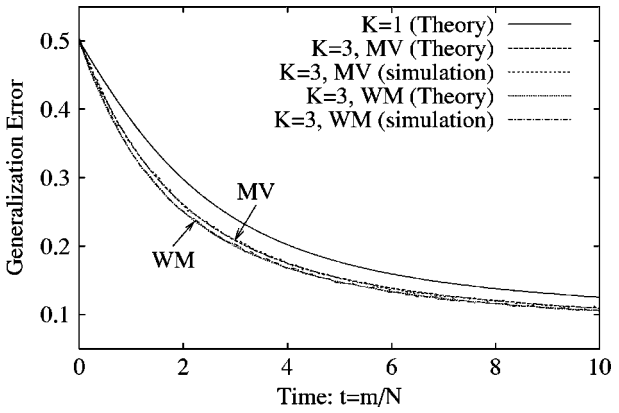

FIG. 5. Dynamical behaviors of ensemble generalization error ϵ_g in perceptron learning.

case of $K=3$ by using Eqs. (22)–(24) and the above R and q. Figure 5 shows the results. This figure shows that the ensemble generalization errors obtained by theoretical calculation explain the computer simulation quantitatively.

Figures 6 and 7 show the results of computer simulations where $N=10^3$, $K=1$, 3, 11, 31 until $t=10^4$ in order to investigate asymptotic behaviors of generalization errors. The effect of ensemble is maintained asymptotically. Asymptotic behavior of generalization error in perceptron learning in the case of the number K of students at unity is $O(t^{-1/3})$ [9]. Note that though this asymptotic behavior is worse than that of Hebbian learning, perceptron learning is robust to the input distribution. On the contrary, Hebbian learning fails whenever the teacher vector of a linearly separable rule is not aligned with one of the principle components of the input distribution [16]. Since properties of $K=3,11,31$ are parallel to those of $K=1$ in Figs. 6 and 7, asymptotic orders of the generalization error in the case of ensemble learning are considered equal to those of $K=1$,

To clarify the relationship between K and the effect of ensemble, we have obtained theoretical ensemble generalization errors for various values of K. In the same manner as Hebbian learning, Fig. 8 shows the results obtained by the

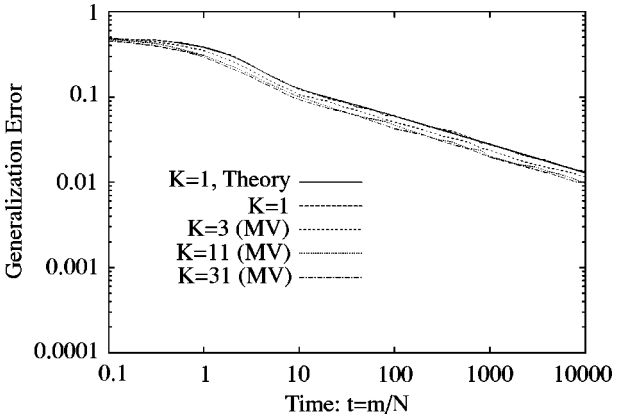

FIG. 6. Asymptotic behavior of generalization error of majority vote in perceptron learning. Computer simulations, except for the solid line. Asymptotic order of ensemble learning is the same as that at $K=1$.

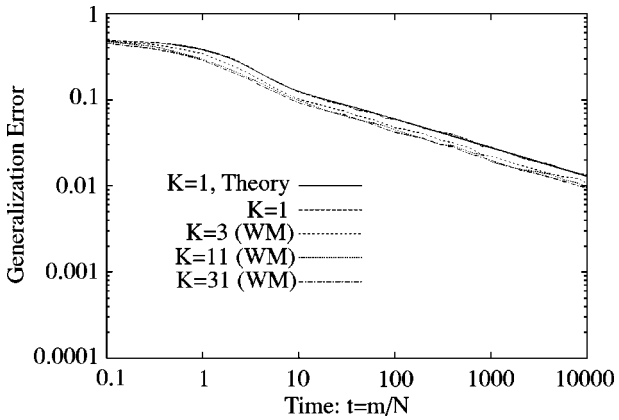

FIG. 7. Asymptotic behavior of generalization error of weight mean in perceptron learning. Computer simulations, except for the solid line. Asymptotic order of ensemble learning is the same as that at $K=1$.

Metropolis method using the values of R and q calculated numerically for perceptron learning and Eqs. (42)–(44).

D. AdaTron learning

The update procedure for AdaTron learning is

$$f(\mathrm{sgn}(v),u) = -u\Theta(-uv). \tag{46}$$

Using this expression, $\langle f_k u_k \rangle$, $\langle f_k v \rangle$, and $\langle f_k^2 \rangle$ in the case of AdaTron learning can be obtained by executing Eqs. (27)–(29) analytically [9,17] (see Appendix C). In addition to these results, we have derived $\langle f_k u_{k'} \rangle$ and $\langle f_k f_{k'} \rangle$. Using Eq. (46), $\langle f_k u_{k'} \rangle \langle f_k f_{k'} \rangle$ and in the case of AdaTron learning are obtained as follows by executing Eqs. (33) and (35) analytically.

In the same manner as Hebbian learning, R and q have been obtained by solving Eqs. (25), (26), (32), (38), and (39), and the derived sample averages numerically. We have obtained numerical ensemble generalization errors ϵ_g in the case of $K=3$ by using Eqs. (22)–(24) and the above R and q.

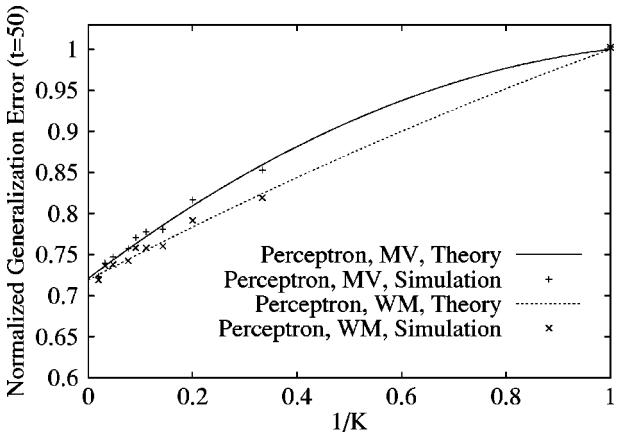

FIG. 8. Relationship between K and effect of ensemble in perceptron learning. Ensemble generalization error ϵ_g for a large K limit is about 0.72 times that of $K=1$.

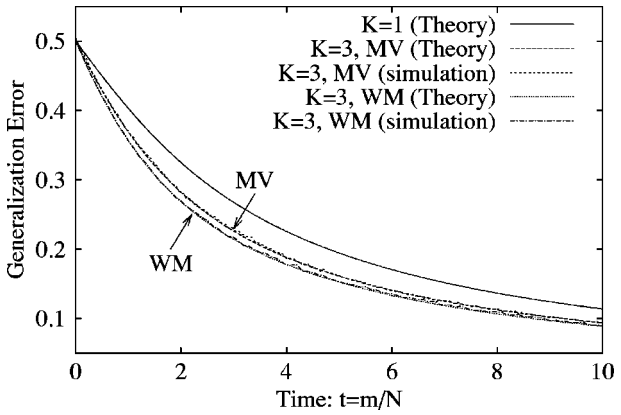

FIG. 9. Dynamical behaviors of ensemble generalization error ϵ_g in AdaTron learning. Improvement of ϵ_g by increasing K from 1 to 3 is largest of the three learning rules.

Figure 9 shows the results. This figure shows that the ensemble generalization errors obtained by theoretical calculation explain the computer simulation quantitatively.

Figures 10 and 11 show the results of computer simulations where $N=10^3$, $K=1$, 3, 11, 31 until $t=10^4$ in order to investigate asymptotic behaviors of generalization errors. Effect of ensemble is maintained asymptotically. Asymptotic behavior of generalization error in AdaTron learning in the case of the number K of students at unity is $O(t^{-1})$ [9,12]. Asymptotic orders of the generalization error in the case of ensemble learning are considered equal to those of $K=1$, since properties of $K=3$, 11, 31 are parallel to those of $K=1$ in these figures.

To clarify the relationship between K and the effect of ensemble, we have obtained theoretical ensemble generalization errors for various values of K. In the same manner as Hebbian learning, Fig. 12 shows the results obtained by the Metropolis method using the values of R and q calculated numerically for perceptron learning and Eqs. (42)–(44).

V. DISCUSSION

Figures 1, 4, 5, 8, 9, and 12 show that the generalization errors of the three learning rules are all improved by ensemble learning. However, the degree of improvement is small in Hebbian learning and large in AdaTron learning. First, we discuss the reason for this difference in the following.

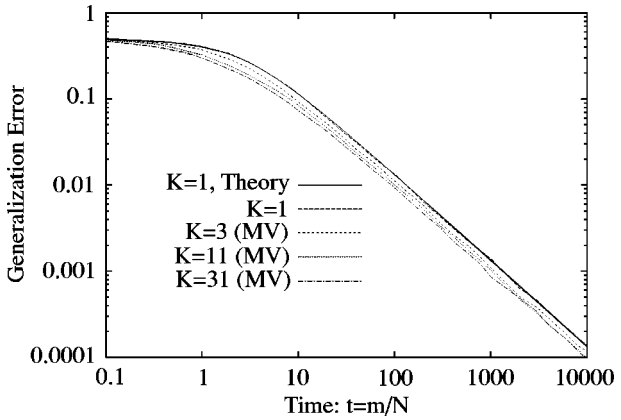

FIG. 10. Asymptotic behavior of generalization error of majority vote in AdaTron learning. Computer simulations, except for the solid line. Asymptotic order of ensemble learning is the same as that at $K=1$.

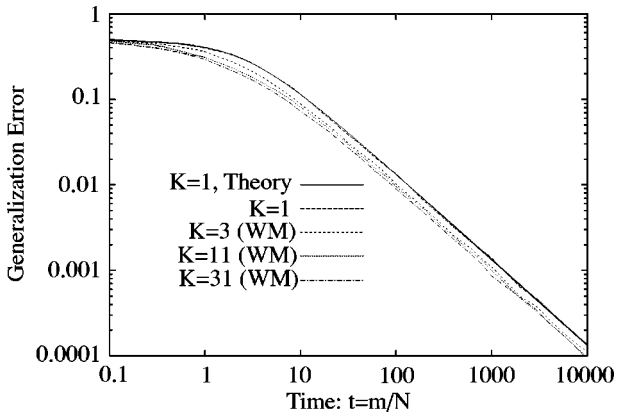

FIG. 11. Asymptotic behavior of generalization error of weight mean in AdaTron learning. Computer simulations, except for the solid line. Asymptotic order of ensemble learning is the same as that at $K=1$.

Each student moves towards teacher as learning proceeds. Therefore similarities R_k and $q_{kk'}$ increase and approach unity, leading to R_k and $q_{kk'}$ becoming less irrelevant to each other. For example when $R_k=R_{k'}=1$, $q_{kk'}$ cannot be $\neq 1$ since a teacher $\boldsymbol{B}$, a student $\boldsymbol{J}_k$, and another student $\boldsymbol{J}_{k'}$ have the same direction. Thus R_k and $q_{kk'}$ are under a certain relationship restraint with each other. When $q_{kk'}$ is relatively smaller when compared with R_k, variety among students is further maintained and the effect of the ensemble can be considered as large. On the contrary, after $q_{kk'}$ becomes unity, a student $\boldsymbol{J}_k$ and another student $\boldsymbol{J}_{k'}$ are the same and there is no merit in combining them.

Let us explain these considerations intuitively by using Fig. 13. Both (a) and (b) show the relationship among two students $\boldsymbol{J}_1$, $\boldsymbol{J}_2$ and a teacher $\boldsymbol{B}$ when learning has proceeded to some degree from the condition that the students and the teacher have no correlation. Then, as shown in Fig. 13, stu-

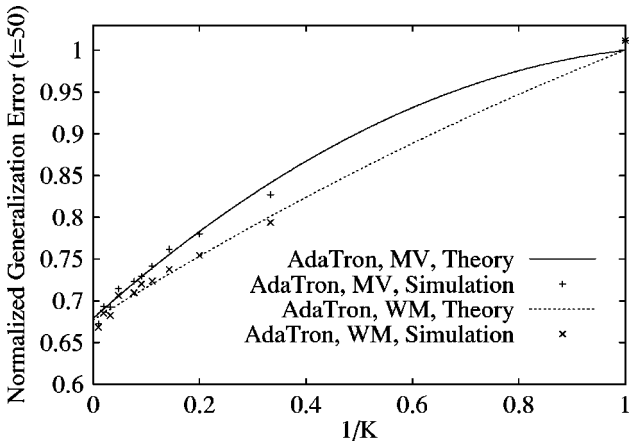

FIG. 12. Relationship between K and effect of ensemble in AdaTron learning. Ensemble generalization error ϵ_g for a large K limit is about 0.68 times that of $K=1$.

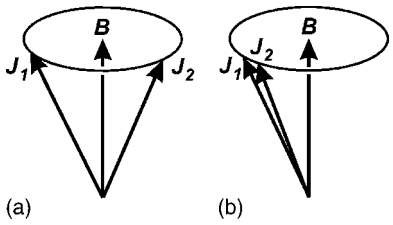

FIG. 13. Variety among students.

dents must distribute to points the same distance from the teacher. That is, the similarity R_1 of the teacher and a student J_1 equals the similarity R_2 of the teacher and a student $\boldsymbol{J}_2$ in both (a) and (b). Here, (a) shows the case in which students are unlike each other—in other words the variety among students is large, that is, q is small. In this case, it is obvious that a mean vector of $\boldsymbol{J}_1$ and $\boldsymbol{J}_2$ is closer to the teacher B than either $\boldsymbol{J}_1$ or $\boldsymbol{J}_2$. Therefore, a mean vector $(1/K)\Sigma_{k=1}^{K}\boldsymbol{J}_k$ of the students' connection weights can closely approximate the connection weight vector $\boldsymbol{B}$ of the teacher in cases like (a). In addition, a combination method other than a mean of students, e.g., the majority vote of students, must approximate the teacher better than each student can do alone in cases like (a). In this case, the effect of ensemble learning is strong. On the contrary, Fig. 13(b) shows the case in which students are similar to each other—in other words, the variety among students is small, meaning q is large. In this case, the significance of combining two students is small since their outputs are almost always the same. Therefore the effect of ensemble learning is small when q is large, as in Fig. 13(b). Thus the relationship between R_k and $q_{kk'}$ is essential to know in ensemble learning.

Figure 14 shows a comparison between the theoretical results regarding the dynamical behaviors of R and q of Hebbian learning, which are obtained by solving Eqs. (25), (26), (32), (38), (39), and (A2)–(A4) numerically and by computer simulation ($N=10^5$). In the same manner, Fig. 15 shows a comparison between the theoretical results regarding the dynamical behaviors of R and q of perceptron learning, which are obtained by solving Eqs. (25), (26), (32), (38), (39), and (B2)–(B5) numerically and by computer simulation ($N=10^5$). Figure 16 shows a comparison between the theoretical results regarding the dynamical behaviors of R and q of AdaTron learning, which are obtained by solving Eqs. (25), (26), (32), (38), (39), and (C3)–(C7) numerically and by computer simulation ($N=10^5$). In these figures, the theoretical results and the computer simulations closely agree with each other. That is, the derived theory explains the computer simulation quantitatively. Figure 14 shows that q rises more rapidly than R in Hebbian learning; in other words, q is relatively large when compared with R, meaning the variety among students disappears rapidly in Hebbian learning. Figure 15 shows that q is smaller than R in the early period of learning ($t<4.0$), which means perceptron learning maintains the variety among students for a longer time than Hebbian learning. Figure 16 shows that q is relatively smaller when compared with R than in the cases of Hebbian learning and perceptron learning. This means AdaTron learning maintains variety among students most out of these three learning rules.

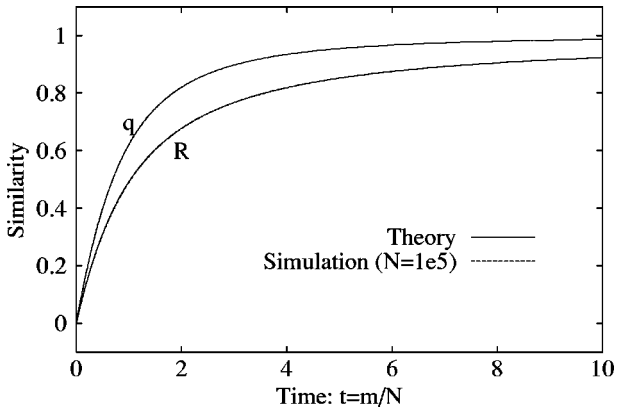

FIG. 14. Dynamical behaviors of R and q in Hebbian learning. Here, q rises more rapidly than R, which means the variety among students disappears rapidly in Hebbian learning.

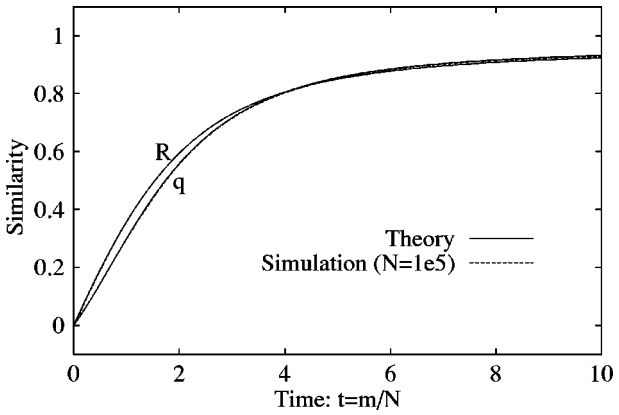

FIG. 15. Dynamical behaviors of R and q in perceptron learning. Here, q is smaller than R in the early period of learning ($t<4.0$). Perceptron learning maintains the variety among students for a longer time than Hebbian learning.

Figures 14–16 show that q is relatively small when compared with R in the case of AdaTron learning than in Heb-

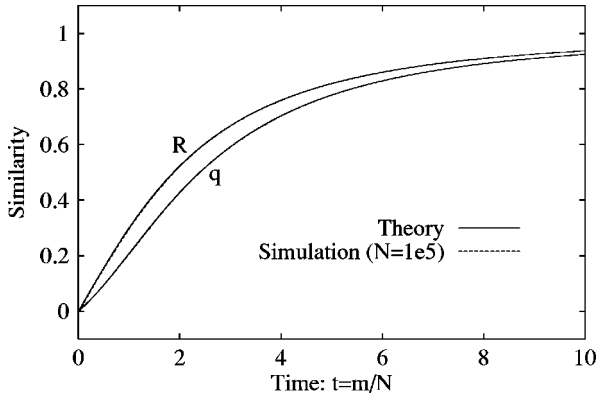

FIG. 16. Dynamical behaviors of R and q in AdaTron learning. Here, q is relatively smaller when compared with R than in the cases of Hebbian learning or perceptron learning. AdaTron learning maintains variety among students most out of these three learning rules.

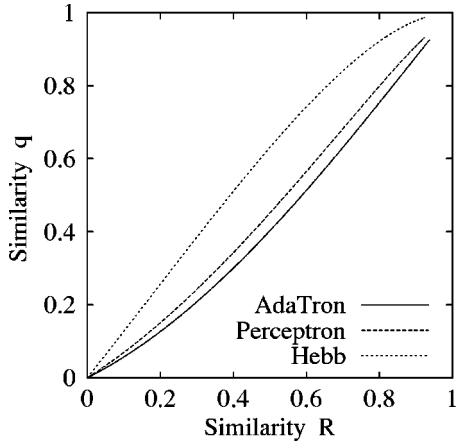

FIG. 17. Relationship between R and q (theory). Here, q of AdaTron learning is the smallest when compared with R. The rising of q is the slowest and variety among students is best maintained in AdaTron learning.

bian learning and perceptron learning. As described before, the relationship between R and q is essential in ensemble learning. To illustrate this, Fig. 17 shows the relationship more clearly by taking R and q as axes. In this figure, the curve for AdaTron learning is located in the bottom. That is, of the three learning rules, the one offering the smallest q when compared with R is AdaTron learning. In other words, the learning rule in which the rising of q is the slowest and the variety among students is maintained best is AdaTron learning.

These characteristics can be understood from the update expression of each rule. Equation (40) means that an update by Hebbian learning depends on only the output $\mathrm{sgn}(v)$ of a teacher. That is, all students are updated identically at all time steps. Therefore the similarity of students increases rapidly in Hebbian learning. On the other hand, the update by perceptron learning equals that of Hebbian learning times $\Theta(-uv)$, as shown in Eq. (45). Students whose outputs are opposite to that of a teacher change their connection weights. At least in the initial period of learning, students whose output is opposite to that of a teacher and students whose output is the same as that of a teacher both exist. As a result, students that change their connection weights and students who do not change their connection weights both exist, leading to the fact that variety among students by perceptron learning is better maintained than by Hebbian learning. The update by AdaTron learning is given in Eq. (46). This can be rewritten as $f(\mathrm{sgn}(v),u)=|u|\Theta(-uv)\mathrm{sgn}(v)$. That is, the update by AdaTron learning equals that of perceptron learning times $|u|$, which depends on the students. Therefore the variety among students by AdaTron learning is still better maintained.

In the discussion above, the reason why the degree of improvement by ensemble learning is small in Hebbian learning and large in AdaTron learning as shown in Fig. 1, 4, 5, 8, 9, and 12 has been explained. AdaTron learning originally featured the fastest asymptotic characteristic of the three learning rules [9]. However, it has a disadvantage that the learning is slow at the beginning; that is, the generalization error is larger than for the other two learning rules in the period of $t<6$. This paper shows that the fastest asymptotic characteristic of AdaTron learning is maintained in ensemble learning and that AdaTron learning has a good affinity with ensemble learning in regard to "the variety among students" and the disadvantage of the early period can be improved by combining it with ensemble learning.

From the perspective of the difference between the majority vote and the weight mean, Figs. 1, 4, 5, 8, 9, and 12 show that the improvement by weight mean is larger than that by majority vote in all three learning rules. Improvement in the generalization error by averaging connection weights of various students can be understood intuitively because the mean of students is close to that of the teacher in Fig. 13(a). The reason why the improvement in the majority vote is smaller than that in the weight mean is considered to be that the variety among students cannot be utilized as effectively by the majority vote as by the weight mean. However, the majority vote can determine an ensemble output only using outputs of students, and is easy to implement. It is therefore significant that the effect of an ensemble in the case of the majority vote has been analyzed quantitatively.

Figures 4, 8, and 12 also show that the ensemble generalization errors ϵ_g by the majority vote are larger than those by the weight mean in the case of $K<\infty$. In both perceptron learning and AdaTron learning, the relationship between $1/K$ and ϵ_g shows a straight line and an upwards-convex curve in the case of the weight mean and the majority vote, respectively. The ensemble generalization errors ϵ_g in the cases of the majority vote and the weight mean agree with each other at a large K limit. This fact agrees with the description in Ref. [6]. Therefore the weight mean is superior than the majority vote especially in the case of a small K. Moreover, it is shown that ϵ_g for a large K limit compared with that of $K=1$ is about 0.99, 0.72, and 0.68 times in Hebbian, perceptron and AdaTron learning, respectively. It has been confirmed that ensemble has the strongest effect in AdaTron learning among three learning rules.

VI. CONCLUSION

This paper discussed ensemble learning of K nonlinear perceptrons, which determine their outputs by sign functions within the framework of online learning and statistical mechanics. One purpose of statistical learning theory is to theoretically obtain the generalization error. In this paper, we have shown that the ensemble generalization error can be calculated by using two order parameters, that is the similarity between the teacher and a student, and the similarity among students. The differential equations that describe the dynamical behaviors of these order parameters have been derived in the case of general learning rules. The concrete forms of these differential equations have been derived analytically in the cases of three well-known rules: Hebbian learning, perceptron learning, and AdaTron learning. We calculated the ensemble generalization errors of these three rules by using the results determined by solving their differential equations. As a result, these three rules have different characteristics in their affinity for ensemble learning, that is,

"maintaining variety among students." The results show that AdaTron learning is superior to the other two rules with respect to that affinity.

ACKNOWLEDGMENT

This research was partially supported by the Ministry of Education, Culture, Sports, Science and Technology, Japan, by Grant-in-Aid for Scientific Research Grant Nos.13780313, 14084212, 15500151 and 16500093.

APPENDIX A: SAMPLE AVERAGES OF HEBBIAN LEARNING

The update procedure for Hebbian learning is

$$f(\mathrm{sgn}(v),u) = \mathrm{sgn}(v). \tag{A1}$$

Using this expression, $\langle f_k u_k \rangle$, $\langle f_k v \rangle$, and $\langle f_k^2 \rangle$ in the case of Hebbian learning can be obtained as follows by executing Eqs. (27)–(29) analytically [9,17]:

$$\langle f_k u_k \rangle = \frac{2R}{\sqrt{2\pi}}, \quad \langle f_k v \rangle = \sqrt{\frac{2}{\pi}}, \quad \langle f_k^2 \rangle = 1. \tag{A2}$$

In addition to these results, we have derived $\langle f_k u_{k'} \rangle$ and $\langle f_k f_{k'} \rangle$. Since Eq. (40) is independent of u, we obtain

$$\langle f_k u_{k'} \rangle = \langle f_k u_k \rangle = \frac{2R}{\sqrt{2\pi}}, \tag{A3}$$

$$\langle f_k f_{k'} \rangle = \langle [\mathrm{sgn}(v)]^2 \rangle = 1. \tag{A4}$$

APPENDIX B: SAMPLE AVERAGES OF PERCEPTRON LEARNING

The update procedure for perceptron learning is

$$f(\mathrm{sgn}(v),u) = \Theta(-uv)\mathrm{sgn}(v). \tag{B1}$$

Using this expression, $\langle f_k u_k \rangle$, $\langle f_k v \rangle$, and $\langle f_k^2 \rangle$ in the case of perceptron learning can be obtained as follows by executing Eqs. (27)–(29) analytically [9,17]:

$$\langle f_k u_k \rangle = \frac{R-1}{\sqrt{2\pi}}, \quad \langle f_k v \rangle = \frac{1-R}{\sqrt{2\pi}}, \tag{B2}$$

$$\langle f_k^2 \rangle = 2\int_0^\infty Dv H\left(\frac{Rv}{\sqrt{1-R^2}}\right) = \frac{1}{\pi}\tan^{-1}\frac{\sqrt{1-R^2}}{R}. \tag{B3}$$

In addition to these results, we have derived $\langle f_k u_{k'} \rangle$ and $\langle f_k f_{k'} \rangle$. Using Eq. (B1), $\langle f_k u_{k'} \rangle$ and $\langle f_k f_{k'} \rangle$ in the case of perceptron learning are obtained as follows by executing Eqs. (33) and (35) analytically:

$$\langle f_k u_{k'} \rangle = \int du_k du_{k'} dv p_3(u_k,u_{k'},v)\Theta(-u_k v)\mathrm{sgn}(v)u_{k'}$$

$$= \frac{R-q}{\sqrt{2\pi}}, \tag{B4}$$

$$\langle f_k f_{k'} \rangle = \int du_k u_{k'} dv p_3(u_k,u_{k'},v)\Theta(-u_k v)\Theta(-u_{k'} v)$$

$$= 2\int_0^\infty Dv \int_{Rv/\sqrt{1-R^2}}^\infty DxH(z), \tag{B5}$$

where

$$z \equiv \frac{-(q-R^2)x + R\sqrt{1-R^2}v}{\sqrt{(1-q)(1+q-2R^2)}} \tag{B6}$$

and the definitions of $H(u)$ and Dx are

$$H(u) \equiv \int_u^\infty Dx, \tag{B7}$$

$$Dx \equiv \frac{dx}{\sqrt{2\pi}}\exp\left(-\frac{x^2}{2}\right). \tag{B8}$$

APPENDIX C: SAMPLE AVERAGES OF ADATRON LEARNING

The update procedure for AdaTron learning is

$$f(\mathrm{sgn}(v),u) = -u\Theta(-uv). \tag{C1}$$

Using this expression, $\langle f_k u_k \rangle$, $\langle f_k v \rangle$, and $\langle f_k^2 \rangle$ in the case of AdaTron learning can be obtained as follows by executing Eqs. (27)–(29) analytically [9,17]:

$$\langle f_k u_k \rangle = -2\int_0^\infty Duu^2 H\left(\frac{Ru}{\sqrt{1-R^2}}\right) \tag{C2}$$

$$= -\frac{1}{\pi}\cot^{-1}\left(\frac{R}{\sqrt{1-R^2}}\right) + \frac{1}{\pi}R\sqrt{1-R^2}, \tag{C3}$$

$$\langle f_k v \rangle = \frac{(1-R^2)^{3/2}}{\pi} + R\langle f_k u_k \rangle, \tag{C4}$$

$$\langle f_k^2 \rangle = -\langle f_k u_k \rangle. \tag{C5}$$

In addition to these results, we have derived $\langle f_k u_{k'} \rangle$ and $\langle f_k f_{k'} \rangle$. Using Eq. (C1), $\langle f_k u_{k'} \rangle$ and $\langle f_k f_{k'} \rangle$ in the case of AdaTron learning are obtained as follows by executing Eqs. (33) and (35) analytically:

$$\langle f_k u_{k'} \rangle = -\int du_k du_{k'} dv p_3(u_k,u_{k'},v)\Theta(-u_k v)u_k u_{k'}$$

$$= \frac{1+q}{\pi}R\sqrt{1-R^2} - 2q\int_0^\infty Dv \int_{Rv/\sqrt{1-R^2}}^\infty Dx x^2, \tag{C6}$$

$$\langle f_k f_{k'} \rangle = \int dv du_k u_k du_{k'} u_{k'} p_3(u_k, u_{k'}, v), \Theta(-u_k v)\Theta(-u_{k'} v)$$

$$= \frac{(1-q)^2(1+q-2R^2)}{2\pi(1-R^2)^{3/2}}\left(\sqrt{\frac{(1+q)(1-R^2)}{1-q}} - R\right)$$

$$+ 2(q-R^2)\int_0^\infty Dv \int_{Rv/\sqrt{1-R^2}}^\infty Dx x^2 H(z)$$

$$- \frac{2R(1+q-2R^2)}{\sqrt{1-R^2}}\int_0^\infty Dvv \int_{Rv/\sqrt{1-R^2}}^\infty DxxH(z)$$

$$+ 2R^2\int_0^\infty Dvv^2 \int_{Rv/\sqrt{1-R^2}}^\infty DxH(z), \qquad \text{(C7)}$$

where the definitions of z, $H(u)$, and Dx are Eqs. (B6), (B7), and (B8), respectively.

[1] Y. Freund and R. E. Schapire, J. Jpn. Soc. Artif. Intell. **14**, 771 (1999) (in Japanese, translation by N. Abe).
[2] L. Breiman, Mach. Learn. **26**, 123 (1996).
[3] Y. Freund and R. E. Shapire, J. Comput. Syst. Sci. **55**, 119 (1997).
[4] K. Hara and M. Okada, cond-mat/0402069
[5] A. Krogh and P. Sollich, Phys. Rev. E **55**, 811 (1997).
[6] R. Urbanczik, Phys. Rev. E **62**, 1448 (2000).
[7] J. A. Hertz, A. Krogh, and R. G. Palmer, *Introduction to the Theory of Neural Computation* (Addison-Wesley, Redwood City, CA, 1991).
[8] M. Opper and W. Kinzel, in *Physics of Neural Networks III*, edited by E. Domany, J. L. van Hemmen, and K. Schulten (Springer, Berlin, 1995).
[9] H. Nishimori, *Statistical Physics of Spin Glasses and Information Processing: An Introduction* (Oxford University Press, Oxford, 2001).
[10] J. K. Anlauf and M. Biehl, Europhys. Lett. **10**, 687 (1989).
[11] M. Biehl and P. Riegler, Europhys. Lett. **28**, 525 (1994).
[12] J. Inoue and H. Nishimori, Phys. Rev. E **55**, 4544 (1997).
[13] D. Saad, *On-line Learning in Neural Networks* (Cambridge University Press, Cambridge, England, 1998).
[14] S. Miyoshi, K. Hara, and M. Okada, IEICE Technical Report No. 103, 2003 (in Japanese), p.13.
[15] S. Miyoshi, K. Hara, and M. Okada, in Proceedings of the 2003 Annual Conference of the Japanese Neural Network Society, 104 (in Japanese).
[16] P. Riegler, M. Biehl, S. A. Solla, and C. Marangi, in Proceedings of the 7th Italian Workshop on Neural Nets. Vol. 87.
[17] A. Engel and C. V. Broeck, *Statistical Mechanics of Learning* (Cambridge University Press, Cambridge, England, 2001).