LETTER *Special Section on Information-Based Induction Sciences and Machine Learning*

Statistical Mechanics of On-Line Learning Using Correlated Examples

Kento NAKAO[†a)], **Yuta NARUKAWA**[††], *Nonmembers*, *and* **Seiji MIYOSHI**[†††b)], *Senior Member*

SUMMARY We consider a model composed of nonlinear perceptrons and analytically investigate its generalization performance using correlated examples in the framework of on-line learning by a statistical mechanical method. In Hebbian and AdaTron learning, the larger the number of examples used in an update, the slower the learning. In contrast, Perceptron learning does not exhibit such behaviors, and the learning becomes fast in some time region.

key words: *nonlinear perceptron, on-line learning, generalization error, correlated examples*

1. Introduction

Learning is the inference of underlying rules that dominate data generation using observed data, where the observed data are pairs of inputs and outputs from a teacher and are called examples. Learning can be roughly classified into batch learning and on-line learning [1]. In batch learning, the given examples are used more than once. In this paradigm, a student gives correct answers after training if the student has adequate freedom. However, a long time and a large memory to store many examples are required. In contrast, in on-line learning, examples are used once and then discarded. In this case, a student cannot give correct answers to all the examples used in training. However, on-line learning has merits, for example, a large memory for storing many examples is not necessary and it is possible to follow a time-variant teacher [2].

Recently, some interesting models have been analyzed with the framework of on-line learning from temporal or spatial viewpoints. However, they have treated cases in which inputs are independently generated [2]. In practical applications, inputs are considered to be mutually correlated in many cases. The treatment of correlated inputs by a model composed of linear perceptrons has already been analyzed [3]. Considering the applications of pattern recognition, it is important to theoretically investigate the generalization performance of a nonlinear learning machine using correlated inputs. In this paper we consider a model composed of simple perceptrons and analytically investigate its generalization performance using correlated inputs in the framework of on-line learning by a statistical mechanical method.

Manuscript received December 30, 2010.
Manuscript revised May 5, 2011.

[†]The author is with the Graduate School of Science and Engineering, Kansai University, Suita-shi, 564–8680 Japan.
[††]The author is with DAIHEN Corporation, Osaka-shi, 532–8512 Japan.
[†††]The author is with the Faculty of Engineering Science, Kansai University, Suita-shi, 564–8680 Japan.
a) E-mail: k156300@kansai-u.ac.jp
b) E-mail: miyoshi@ipcku.kansai-u.ac.jp

DOI: 10.1587/transinf.E94.D.1941

2. Model

In this paper we consider a teacher machine and a student machine as simple perceptrons with connection weights $\boldsymbol{B}$ and $\boldsymbol{J}^m$, respectively. The teacher $\boldsymbol{B} = (B_1, \ldots, B_N)^T$, the student $\boldsymbol{J}^m = (J_1^m, \ldots, J_N^m)^T$ and the input $\boldsymbol{x}_k^m = (x_{k1}^m, \ldots, x_{kN}^m)^T$, $k = 1, \ldots, K$, are N-dimensional vectors. Each component B_i of $\boldsymbol{B}$ is independently drawn from $\mathcal{N}(0, 1)$ and fixed, where $\mathcal{N}(0, 1)$ denotes a Gaussian distribution with a mean of zero and a variance of unity. Each component J_i^0 of the initial student $\boldsymbol{J}^0$ is also independently drawn from $\mathcal{N}(0, 1)$. The direction cosine between $\boldsymbol{B}$ and $\boldsymbol{J}^m$ is R^m. Each component x_{ki}^m of $\boldsymbol{x}_k^m$ is generated as follows:

$$\boldsymbol{\xi}^m = (\xi_1^m, \ldots, \xi_N^m)^T, \qquad \xi_i^m \sim \mathcal{N}(0, 1) \tag{1}$$

$$P\left(x_{ki}^m = \pm\frac{\xi_i^m}{\sqrt{N}}\right) = \frac{1 \pm a}{2}, \tag{2}$$

where $-1 \leq a \leq 1$. m, T and $P(\cdot)$ denote the time step, the transposition and the probability, respectively. Equations (1) and (2) imply that the component ξ_i^m of $\boldsymbol{\xi}^m$ is independently generated and that K inputs $\boldsymbol{x}_k^m$, $k = 1, \ldots, K$, are generated at each time step m. In this manner, K inputs that have direction cosine a with parent vector $\boldsymbol{\xi}^m$ are generated. In this case, $\boldsymbol{x}_k^m \cdot \boldsymbol{x}_{k'}^m = a^2$, $k \neq k'$. The K inputs are used as a set in learning. Note that there is no correlation between $\boldsymbol{J}^0$ and $\boldsymbol{B}^m$, $\boldsymbol{J}^m$ and $\boldsymbol{x}_k^m$, nor $\boldsymbol{B}^m$ and $\boldsymbol{x}_k^m$, although there is a correlation between inputs. Considering the practical applications, temporally uniform correlation is also interesting. However, such correlation is difficult to treat analytically. Therefore, in this paper we treat the correlation considered in [3].

In this paper, the thermodynamic limit is also treated. Therefore, $\|\boldsymbol{B}\| = \sqrt{N}$, $\|\boldsymbol{J}^0\| = \sqrt{N}$ and $\|\boldsymbol{x}_k^m\| = 1$. Generally, since the norm $\boldsymbol{J}^m$ of the student changes with time, the ratio l^m of the norm to $\|\boldsymbol{J}^0\|$ is introduced and is called the length of the student. That is, $\|\boldsymbol{J}^m\| = l^m\sqrt{N}$. In the case of simple perceptrons, the outputs of the teacher and student are $\mathrm{sgn}(v_k^m)$ and $\mathrm{sgn}(u_k^m l^m)$, respectively. Here, $v_k^m = \boldsymbol{B} \cdot \boldsymbol{x}_k^m$ and $u_k^m l^m = \boldsymbol{J}^m \cdot \boldsymbol{x}_k^m$. Thus, v_k^m and u_k^m obey Gaussian distributions with a mean of zero and a variance of unity, and the covariance between v_k^m and u_k^m is R^m.

The update rule for student $\boldsymbol{J}$ is given by

$$\boldsymbol{J}^{m+1} = \boldsymbol{J}^m + \sum_{k=1}^{K} f_k^m \boldsymbol{x}_k^m, \tag{3}$$

where f_k^m is a function determined by the learning rule. Hebbian, Perceptron and AdaTron learning are well-known learning rules for simple perceptrons.

3. Theory

3.1 Generalization Error

One purpose of statistical learning theory is to theoretically obtain the generalization error ϵ_g, which is the mean of errors over the distribution of a new input. We define the error ϵ to be 0 if the outputs of the teacher and student agree, and ϵ to be 1 if the outputs disagree. Thus, ϵ_g is the probability that the outputs of the teacher and student disagree. ϵ_g can be calculated as follows [2], [4]:

$$\epsilon_g = \langle \epsilon \rangle = \int d\boldsymbol{x} P(\boldsymbol{x}) \epsilon = \int du dv P(u,v) \epsilon = \frac{1}{\pi} \cos^{-1} R. \tag{4}$$

Here,

$$P(u,v) = \frac{1}{2\pi |\boldsymbol{\Sigma}_2|^{\frac{1}{2}}} \exp\left(-\frac{(u,v)\boldsymbol{\Sigma}_2^{-1}(u,v)^T}{2} \right), \tag{5}$$

$$\boldsymbol{\Sigma}_2 = \begin{pmatrix} 1 & R \\ R & 1 \end{pmatrix}. \tag{6}$$

3.2 Simultaneous Differential Equations for Dynamical Behaviors of Order Parameters

Equation (4) shows that ϵ_g is calculated using the direction cosine R. Therefore, it is desirable to determine the order parameter R. Simultaneous differential equations with deterministic forms that describe the dynamical behaviors of order parameters are obtained by self-averaging in the thermodynamic limit as follows [2]–[4]:

$$\frac{dl^2}{dt} = \left\langle f_k^2 \right\rangle + (K-1) \langle f_k f_{k'} \rangle a^2 + 2l \langle f_k u_k \rangle, \tag{7}$$

$$\frac{dr}{dt} = \langle f_k v_k \rangle, \tag{8}$$

where $\langle \cdot \rangle$ denotes the sample average and $t = Km/N$. Here, to simplify the analysis, the auxiliary order parameter $r \equiv Rl$ has been introduced. $\langle f_k f_{k'} \rangle$ denotes the sample average of the product of the two f for $\boldsymbol{x}_k$ and $\boldsymbol{x}_{k'}$. On the other hand, $\left\langle f_k^2 \right\rangle$, $\langle f_k u_k \rangle$ and $\langle f_k v_k \rangle$ do not depend on k since the $\boldsymbol{x}_k$ are generated from identical distributions, although we include the subscript k to match the notation with $\langle f_k f_{k'} \rangle$. Therefore, r and l do not depend on k. It is necessary to calculate four sample averages for the specific learning rule.

3.3 Hebbian Learning

In the case of Hebbian learning, the update function is $f_k^m = \eta \mathrm{sgn}(v_k^m)$, where η denotes the learning rate of the student and is a constant positive number. The four sample averages can be analytically calculated as follows:

$$\langle f_k u_k \rangle = \frac{2\eta R}{\sqrt{2\pi}}, \quad \langle f_k v_k \rangle = \eta \sqrt{\frac{2}{\pi}}, \quad \left\langle f_k^2 \right\rangle = \eta^2, \tag{9}$$

$$\langle f_k f_{k'} \rangle = \eta^2 \left(1 - \frac{2}{\pi} \cos^{-1} a^2 \right). \tag{10}$$

Substituting Eqs. (9) and (10) into Eqs. (7) and (8) and using $R(0) = 0$ and $l(0) = 1$ as initial conditions, we can analytically solve the simultaneous differential equations as follows:

$$l = \eta \sqrt{\frac{2}{\pi}} t \left(1 + \frac{\pi}{2\eta^2} t^{-2} + \frac{\pi}{2} \left(1 + a^2 (K-1) \left(1 - \frac{2}{\pi} \cos^{-1} a^2 \right) \right) t^{-1} \right)^{\frac{1}{2}}, \tag{11}$$

$$R = \left(1 + \frac{\pi}{2\eta^2} t^{-2} + \frac{\pi}{2} \left(1 + a^2 (K-1) \left(1 - \frac{2}{\pi} \cos^{-1} a^2 \right) \right) t^{-1} \right)^{-\frac{1}{2}}. \tag{12}$$

From Eqs. (4) and (12), ϵ_g can be analytically obtained. If we substitute a sufficiently large t into these equations, then we obtain

$$\epsilon_g \simeq \sqrt{\frac{1}{2\pi} \left(1 + a^2 (K-1) \left(1 - \frac{2}{\pi} \cos^{-1} a^2 \right) \right)} t^{-\frac{1}{2}}. \tag{13}$$

If we substitute a sufficiently large K into Eq. (13), then we obtain

$$\epsilon_g \simeq a \sqrt{\frac{1}{2\pi} \left(1 - \frac{2}{\pi} \cos^{-1} a^2 \right)} \left(K^{-1} t \right)^{-\frac{1}{2}}. \tag{14}$$

Equation (14) implies that the learning speed is proportional to K^{-1}. Meanwhile, if we substitute $a = 0$ into Eqs. (4) and (12), then we obtain

$$\epsilon_g = \frac{1}{\pi} \cos^{-1} \left(1 + \frac{\pi}{2} t^{-1} + \frac{\pi}{2\eta} t^{-2} \right)^{-\frac{1}{2}}. \tag{15}$$

From Eq. (15), ϵ_g does not depend on K in the case of uncorrelated inputs.

3.4 Perceptron Learning

In the case of Perceptron learning, the update function is $f_k^m = \eta \Theta(-u_k^m v_k^m) \mathrm{sgn}(v_k^m)$, where $\Theta(\cdot)$ denotes a step function. The four sample averages can be calculated as follows:

$$\langle f_k u_k \rangle = -\langle f_k v_k \rangle = \eta \frac{R-1}{\sqrt{2\pi}}, \quad \left\langle f_k^2 \right\rangle = \frac{\eta^2}{\pi} \cos^{-1} R, \quad (16)$$

$$\langle f_k f_{k'} \rangle = \eta^2 \int dv_k du_k dv_{k'} du_{k'} P(v_k, u_k, v_{k'}, u_{k'}) \times \Theta(-u_k v_k) \mathrm{sgn}(v_k) \Theta(-u_{k'} v_{k'}) \mathrm{sgn}(v_{k'}). \quad (17)$$

Here,

$$P(u_k, v_k, u_{k'}, v_{k'}) = \frac{1}{(2\pi)^2 |\mathbf{\Sigma}_4|^{\frac{1}{2}}} \times \exp\left(-\frac{(u_k, v_k, u_{k'}, v_{k'}) \mathbf{\Sigma}_4^{-1} (u_k, v_k, u_{k'}, v_{k'})^T}{2} \right), \quad (18)$$

$$\mathbf{\Sigma}_4 = \begin{pmatrix} 1 & R & a^2 & a^2 R \\ R & 1 & a^2 R & a^2 \\ a^2 & a^2 R & 1 & R \\ a^2 R & a^2 & R & 1 \end{pmatrix}. \quad (19)$$

Since the integral in Eq. (17) cannot be analytically executed, it must be numerically executed.

3.5 AdaTron Learning

In the case of AdaTron learning, the update function is $f_k^m = -\eta u_k^m \Theta(-u_k^m v_k^m)$. The four sample averages can be calculated as follows:

$$\left\langle f_k^2 \right\rangle = -\eta \langle f_k u_k \rangle = -\eta^2 \frac{R\sqrt{1-R} - \cos^{-1} R}{\pi}, \quad (20)$$

$$\langle f_k v_k \rangle = \eta \frac{(1-R^2)^{\frac{3}{2}}}{\pi} + R \langle f_k u_k \rangle, \quad (21)$$

$$\langle f_k f_{k'} \rangle = \eta^2 \int dv_k du_k dv_{k'} du_{k'} P(v_k, u_k, v_{k'}, u_{k'}) \times u_k \Theta(-u_k v_k) \mathrm{sgn}(v_k) u_{k'} \Theta(-u_{k'} v_{k'}) \mathrm{sgn}(v_{k'}). \quad (22)$$

Here, $P(u_k, v_k, u_{k'}, v_{k'})$ is given by Eq. (18). Since the integral in Eq. (22) cannot be analytically executed, it must be numerically executed.

4. Results and Discussion

The dynamical behaviors of ϵ_g were obtained by solving Eqs. (4), (7) and (8), and obtaining the sample averages for each learning rule. Figures 1–4 show the theoretical results and the corresponding simulation results when $\eta = 1$. In the computer simulations, $N = 10^4$ and ϵ_g was measured through tests using 10^6 random inputs at each time step. To illustrate the theoretical results for Hebbian learning, we plotted Eqs. (4) and (12). For Perceptron learning, we numerically solved the simultaneous differential equations obtained by substituting Eqs. (16) and (17) into Eqs. (7) and (8) by the Runge-Kutta method. The integration of Eq. (17) was numerically executed by Simpson's method for $K = 1, 10$ and 10^2 and also $K = 10^3$ with $a = 0.0$, and by the Monte Carlo method [5] for $a = 0.6$ and $K = 10^3$. For AdaTron learning, we numerically solved the simultaneous differential equations obtained by substituting Eqs. (20)–(22) into Eqs. (7) and (8) by the Runge-Kutta method. The integration of Eq. (22) was numerically executed by Simpson's method. Figures 1–4 show that the theoretical results and simulation results agree well. This means that the theory is obtained correctly.

Figure 1 shows the results when uncorrelated inputs are used. In this case, ϵ_g does not depend on K for all three learning rules, which is also implied by Eqs. (4), (7) and (8).

Figures 2–4 show the results when correlated inputs are used. Figures 2 and 4 show that, in Hebbian and AdaTron learning, the learning speed with $K = 1000$ is ten times lower than that with $K = 100$. That is, in the case of a large K, the learning speed is in proportion to K^{-1}. In contrast, Figure 3 shows that ϵ_g in the asymptotic region does not depend on K in Perceptron learning. These results illustrate the qualitative differences between Perceptron learning and the other two types of learning. These are very interesting properties. Furthermore, ϵ_g with $K = 100$ and 1000 are smaller than that with $K = 1$ in some time region. This means the generalization capability is better for correlated inputs than for uncorrelated inputs. This is a very interesting phenomenon.

In the case of linear perceptrons, the learning be-

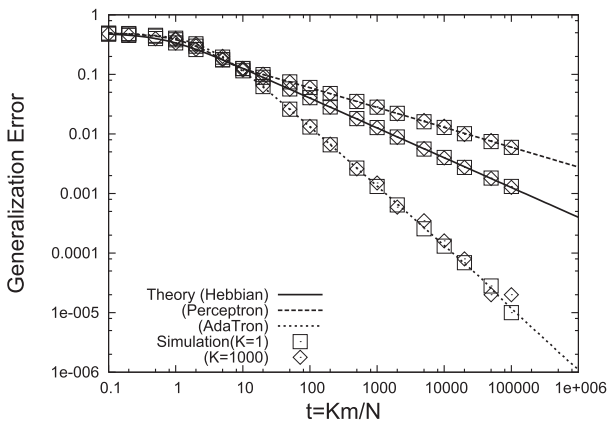

Fig. 1 Dynamical behaviors of generalization error ϵ_g. (Hebbian, Perceptron and AdaTron learning, $a = 0.0$)

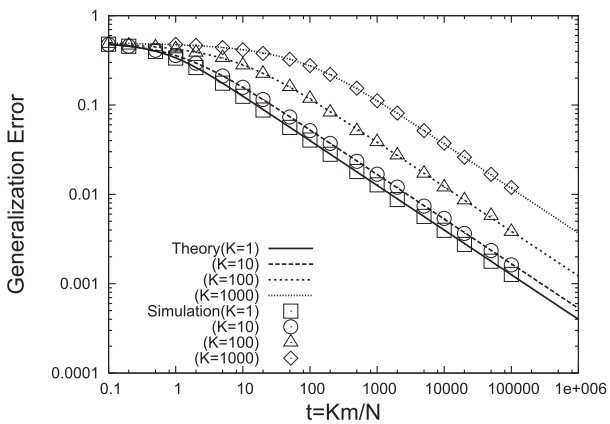

Fig. 2 Dynamical behaviors of generalization error ϵ_g. (Hebbian learning, $a = 0.6$)

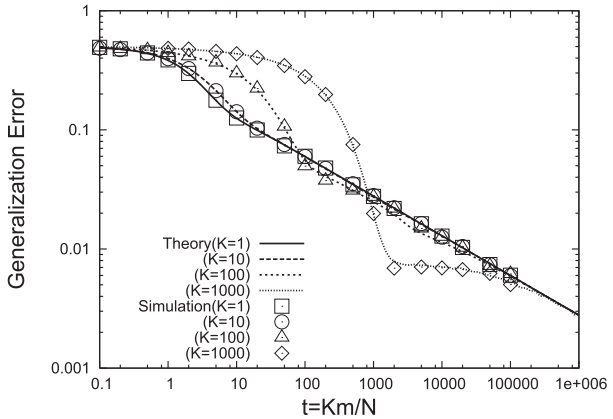

Fig. 3 Dynamical behaviors of generalization error ϵ_g. (Perceptron learning, $a = 0.6$)

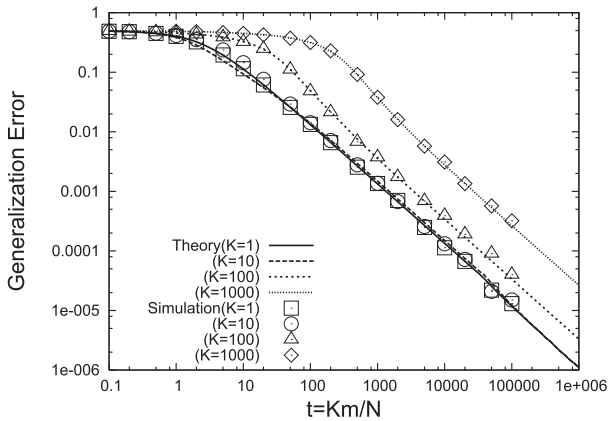

Fig. 4 Dynamical behaviors of generalization error ϵ_g. (AdaTron learning, $a = 0.6$)

comes slower when the inputs are correlated. Therefore, the block orthogonal projection learning was proposed as a learning rule whose learning speed is not affected by the correlation [3]. In the case of nonlinear perceptrons, the different behaviors described in this section are exhibited by different learning rules. In Hebbian and AdaTron learning, the greater the correlation between the inputs, the slower the learning. On the other hand, in Perceptron learning, the learning becomes fast in some time region.

5. Conclusion

In this paper we considered a model composed of simple perceptrons and analytically investigated its generalization performance using correlated inputs in the framework of online learning by a statistical mechanical method. In Hebbian and AdaTron learning, the learning speed is in proportion to K^{-1} when K is large. In contrast, the learning speed of Perceptron learning becomes fast in some time region.

Acknowledgments

This research was partially supported by the Ministry of Education, Culture, Sports, Science and Technology of Japan, with Grants-in-Aid for Scientific Research (21500228) and the Kansai University Grant-in-Aid for Progress of Research in Graduate Course, 2010.

References

[1] D. Saad, ed., On-Line Learning in Neural Networks, Cambridge University Press, 1998.
[2] S. Miyoshi, "Statistical mechanical analysis of on-line learning," Systems, Control and Information, vol.51, no.5, pp.216–233, May 2007.
[3] C. Seki, S. Sakurai, M. Matsuno, and S. Miyoshi, "A theoretical analysis of on-line learning using correlated examples," IEICE Trans. Fundamentals, vol.E91-A, no.9, pp.2663–2670, Sept. 2008.
[4] H. Nishimori, Statistical Physics of Spin Glass and Information Processing, Oxford University Press, 2009.
[5] K. Hukushima and K. Nemoto, "Exchange Monte Carlo method and application to spin glass simulations," J. Physical Society of Japan, vol.65, no.6, pp.1604–1608, June 1996.