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

Statistical Mechanics of Adaptive Weight Perturbation Learning

Ryosuke MIYOSHI[†a)], ***Nonmember***, **Yutaka MAEDA**[††b)], ***Member***, ***and*** **Seiji MIYOSHI**[††c)], ***Senior Member***

SUMMARY Weight perturbation learning was proposed as a learning rule in which perturbation is added to the variable parameters of learning machines. The generalization performance of weight perturbation learning was analyzed by statistical mechanical methods and was found to have the same asymptotic generalization property as perceptron learning. In this paper we consider the difference between perceptron learning and AdaTron learning, both of which are well-known learning rules. By applying this difference to weight perturbation learning, we propose adaptive weight perturbation learning. The generalization performance of the proposed rule is analyzed by statistical mechanical methods, and it is shown that the proposed learning rule has an outstanding asymptotic property equivalent to that of AdaTron learning.

key words: *on-line learning, weight perturbation, simultaneous perturbation, generalization error, statistical mechanics*

1. Introduction

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

Problems exist in which it is necessary to find an adjustable parameter that maximizes or minimizes cost functions. Such problems are widely treated as optimization problems [3]. In this paradigm, the general approach used as a sequential method is the gradient method. In many fields including control theory, methods based on the gradient method are usually used; these methods provide us with an appropriate rule that corrects parameters. However, in cases in which the gradient cannot be used, such methods cannot be used. In this case, if the value of the function can be found, we can consider a method in which the gradient is approximated by using perturbations. However, when the number of adjustable parameters is large, this simple method of calculating the gradient is difficult to apply in most situations since the number of functional values to be calculated is large. To overcome this problem, a simultaneous perturbation optimization method in which perturbations are added to all adjustable parameters was proposed independently by Spall, Alespector et al., Cauwenberghs, and Maeda et al. [4]–[7]. On the other hand, in the machine learning field, methods using perturbations to adjust the connection weights, i.e., the parameters of learning machines, have been proposed. Such methods are known as weight perturbation learning [8]. Applying the simultaneous perturbation optimization method to learning is equivalent to weight perturbation learning.

In the paradigm of analyzing on-line learning by statistical mechanical methods, we consider that inputs with small norms are independently generated at each time step to assume self-averaging [2], [9]. If we use each input as a perturbation, we can directly apply methods for analyzing on-line learning to the analysis of weight perturbation learning [10]. Using this method of analysis, it was shown that weight perturbation learning has the same asymptotic property as perceptron learning.

In this paper, we consider the difference between perceptron learning and AdaTron learning, which are well-known learning rules [11]. By applying this difference to weight perturbation learning, we propose adaptive weight perturbation learning with an improved asymptotic property. We analyze the generalization performance by statistical mechanical methods. Here, for brevity, we refer to weight perturbation and adaptive weight perturbation as WP and AWP, respectively.

2. Model

Two simple perceptrons are considered in this paper: a teacher and a student [2]. Their connection weights are $\boldsymbol{B}$ and $\boldsymbol{J}$, respectively. The same input $\boldsymbol{x}$ is applied to both perceptrons. The teacher $\boldsymbol{B} = (B_1, \ldots B_N)$, the student $\boldsymbol{J}^m = (J_1^m, \ldots J_N^m)$, and the input $\boldsymbol{x}^m = (x_1^m, \ldots x_N^m)$ are N-dimensional vectors, and 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

Manuscript received December 28, 2010.
Manuscript revised May 3, 2011.

[†]The author is with the Graduate School of Science and Engineering, Kansai University, Suita-shi, 564–8680 Japan.
[††]The authors are with the Faculty of Engineering Science, Kansai University, Suita-shi, 564–8680 Japan.
a) E-mail: k329499@kansai-u.ac.jp
b) E-mail: maedayut@kansai-u.ac.jp
c) E-mail: miyoshi@kansai-u.ac.jp

DOI: 10.1587/transinf.E94.D.1937

a variance of unity. Here, m denotes the time step. Each component J_i^0 of the initial value $\boldsymbol{J}^0$ of $\boldsymbol{J}^m$ is also independently drawn from $\mathcal{N}(0,1)$. The norm $\|\boldsymbol{J}^m\|$ of the student changes as the time step proceeds. Therefore, the ratio l^m of the norm to $\sqrt{N}$ is introduced and is called the length of the student. That is, $\|\boldsymbol{J}^m\| = l^m\sqrt{N}$. The direction cosine between $\boldsymbol{J}^m$ and $\boldsymbol{B}$ is R^m. Each component x_i^m of $\boldsymbol{x}^m$ is drawn from $\mathcal{N}(0,1/N)$ independently. Here, $v^m = \boldsymbol{B}\cdot\boldsymbol{x}^m$ and $u^m l^m = \boldsymbol{J}^m\cdot\boldsymbol{x}^m$. In this paper, as we assume the thermodynamic limit $N\to\infty$, u^m and v^m obey a two-dimensional Gaussian distribution,

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

$$\boldsymbol{\Sigma} = \begin{pmatrix} 1 & R^m \\ R^m & 1 \end{pmatrix}. \tag{2}$$

Here, we define the error between the teacher $\boldsymbol{B}$ and the student $\boldsymbol{J}^m$ by $\epsilon^m \equiv \Theta(-u^m v^m)$, where $\Theta(\cdot)$ is a step function. The student $\boldsymbol{J}^m$ is updated by the input $\boldsymbol{x}^m$ and the output of the teacher $\boldsymbol{B}$ for the input. That is,

$$\boldsymbol{J}^{m+1} = \boldsymbol{J}^m + f^m\boldsymbol{x}^m, \tag{3}$$

where f^m is a function determined by the learning rule. Hebbian learning, perceptron learning, and AdaTron learning are well-known learning algorithms with f^m of

$$f^m = \eta\,\mathrm{sgn}(v^m), \tag{4}$$

$$f^m = \eta\,\Theta(-u^m v^m)\mathrm{sgn}(v^m), \tag{5}$$

$$f^m = \eta\,|u^m|\Theta(-u^m v^m)\mathrm{sgn}(v^m), \tag{6}$$

respectively [2]. Here, η denotes the learning rate of the student and is a positive constant.

3. Statistical Mechanical Analysis of On-Line Learning

One of the goals of statistical learning theory is to obtain generalization errors theoretically. The generalization error ϵ_g^m is the mean of the error ϵ^m over the input $\boldsymbol{x}$. In the case of the models considered in this paper,

$$\epsilon_g^m = \frac{1}{\pi}\cos^{-1}R^m \tag{7}$$

[9]. In the following, we suitably omit the time step m. As shown by Eq. (7), ϵ_g is a function of the direction cosine R. Simultaneous differential equations in deterministic forms that describe the dynamical behavior of R can be derived based on self-averaging at the thermodynamic limit as follows [2]:

$$\frac{dl}{dt} = \langle fu\rangle + \frac{\langle f^2\rangle}{2l}, \qquad \frac{dr}{dt} = \langle fv\rangle, \tag{8}$$

where

$$r = Rl. \tag{9}$$

Equation (8) includes three sample means: $\langle fu\rangle$, $\langle f^2\rangle$, and $\langle fv\rangle$. In the cases of Hebbian learning, perceptron learning, and AdaTron learning, these are obtained analytically as follows [2]:

Hebbian learning

$$\langle fu\rangle = \frac{2\eta R}{\sqrt{2\pi}}, \qquad \langle fv\rangle = \eta\sqrt{\frac{2}{\pi}}, \qquad \langle f^2\rangle = \eta^2. \tag{10}$$

Perceptron learning

$$\langle fu\rangle = -\langle fv\rangle = \eta\frac{R-1}{\sqrt{2\pi}}, \qquad \langle f^2\rangle = \frac{\eta^2}{\pi}\cos^{-1}R. \tag{11}$$

AdaTron learning

$$\langle fu\rangle = \frac{\eta}{\pi}(R\sqrt{1-R^2} - \cos^{-1}\mathrm{R}), \tag{12}$$

$$\langle fv\rangle = \frac{\eta}{\pi}(1-R^2)^{3/2} + R\langle fu\rangle, \quad \langle f^2\rangle = -\eta\langle fu\rangle. \tag{13}$$

Here, in the case of Hebbian learning, the simultaneous differential equations obtained by substituting Eq. (10) into Eq. (8) can be solved analytically to obtain [2], [12]

$$r = \eta\sqrt{\frac{2}{\pi}}t, \qquad l^2 = \frac{2\eta^2}{\pi}t^2 + \eta^2 t + 1. \tag{14}$$

4. Weight Perturbation Learning

A method that uses perturbations to correct the connection weights of a learning machine is called WP learning [8]. As stated in the preceding section, in the case of analyzing on-line learning by statistical mechanical methods, we consider that inputs with small norms are generated independently at each time step. If we use these inputs as perturbations, we can directly apply the analysis method to WP learning [10]. An updating function of WP learning is

$$f^m = -\frac{\eta}{2c}g^m, \tag{15}$$

$$g^m = \Theta(-v^m(\boldsymbol{J}^m + c\boldsymbol{x}^m)\cdot\boldsymbol{x}^m) - \Theta(-v^m(\boldsymbol{J}^m - c\boldsymbol{x}^m)\cdot\boldsymbol{x}^m). \tag{16}$$

Here, c denotes a positive constant. The sample averages are

$$\langle fu\rangle = -\frac{\eta}{\sqrt{2\pi}c}\left(\exp\left(-\frac{c^2}{2l^2}\right)\left(1-2H\left(\frac{Rc}{l\sqrt{1-R^2}}\right)\right) - R\left(1-2H\left(\frac{c}{l\sqrt{1-R^2}}\right)\right)\right), \tag{17}$$

$$\langle fv\rangle = \frac{\eta}{\sqrt{2\pi}c}\left(1-2H\left(\frac{c}{l\sqrt{1-R^2}}\right) - R\exp\left(-\frac{c^2}{2l^2}\right)\left(1-2H\left(\frac{Rc}{l\sqrt{1-R^2}}\right)\right)\right), \tag{18}$$

$$\langle f^2\rangle = \frac{\eta^2}{2c^2}\int_{-\frac{c}{l}}^{\frac{c}{l}} DuH\left(-\frac{Ru}{\sqrt{1-R^2}}\right). \tag{19}$$

Here, $H(u) \equiv \int_u^\infty Dx$, $Dx \equiv \frac{dx}{\sqrt{2\pi}}\exp\left(-\frac{x^2}{2}\right)$. Equations (17) and (18) are new results obtained in this paper.

5. Adaptive Weight Perturbation Learning

The asymptotic property of AdaTron learning is $\epsilon_g \sim O(t^{-1})$ [13], which is a superior property. In AdaTron learning, the length l of a student decreases monotonically. AdaTron learning can be considered as an adaptive version of perceptron learning. In this section, we propose AWP learning by considering the difference between perceptron learning and AdaTron learning. In addition, we analyze the generalization performance of the proposed learning rule by statistical mechanical methods. From Eqs. (5) and (6), in perceptron learning and AdaTron learning, an update occurs when the outputs of the teacher and the student are different. In the case of perceptron learning, the norm of the update vector is constant regardless of whether the update direction is that causing the student to become longer or shorter. On the other hand, in the case of AdaTron learning, the norm of the update vector is shorter in the direction in which the length of the student increases, and longer in the direction in which the length of the student decreases since $|u^m|$ is included as a multiplier in Eq. (6). This difference between perceptron learning and AdaTron learning is the reason why the asymptotic properties of the generalization errors are different. To correct WP learning and to obtain a learning rule that greatly updates the student in the direction that the student length decreases, we should set the update function to

$$f^m = -\frac{\eta}{2c}\,|u^m|\,g^m\,\Theta(u^m g^m), \tag{20}$$

$$g^m = \Theta(-v^m(\boldsymbol{J}^m + c\boldsymbol{x}^m)\cdot\boldsymbol{x}^m) - \Theta(-v^m(\boldsymbol{J}^m - c\boldsymbol{x}^m)\cdot\boldsymbol{x}^m). \tag{21}$$

Here, $\Theta(u^m g^m)$ in Eq. (20) means that this rule updates in the direction in which the update vector of the student becomes shorter, and it does not update in the direction in which the update vector of the student becomes longer.

6. Results and Discussion

Figure 1 shows the dynamical behaviors of the generalization error ϵ_g obtained theoretically using Eqs. (7)–(14), (17)–(19), and the corresponding simulation results. Here, $\eta = 1$ and $c = 1$. In this figure, the curves represent the theoretical results and the symbols represent simulation results. For Hebbian learning, theoretical calculations were performed using Eqs. (7), (9), and (14) to obtain the analytical solutions. For perceptron learning and AdaTron learning, we numerically solved Eqs. (7)–(9) and (11)–(13) by the Runge-Kutta method. For WP learning, we numerically calculated the sample average given by Eq. (19) by Simpson's method [14]. For AWP learning, we calculated the sample averages

$$\langle fu\rangle = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} fuP(u,v)dudv, \tag{22}$$

$$\langle fv\rangle = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} fvP(u,v)dudv, \tag{23}$$

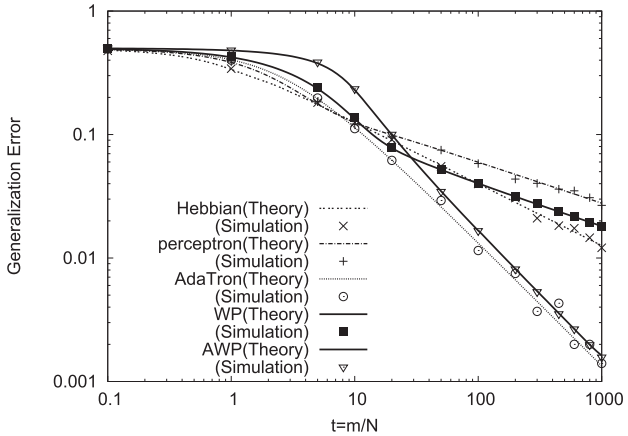

Fig. 1 Dynamical behaviors of generalization error ϵ_g obtained theoretically and by simulation.

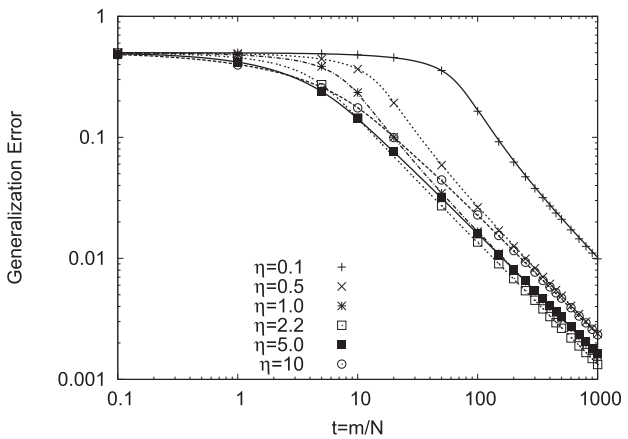

Fig. 2 Dynamical behaviors of generalization error ϵ_g for various η obtained theoretically and by simulation.

$$\langle f^2\rangle = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f^2P(u,v)dudv \tag{24}$$

numerically by Simpson's method [14].

The simulations were performed with $N = 10^4$. The generalization error ϵ_g was obtained through tests using 10^4 random inputs at each time step when the outputs of the teacher and the student were different. Figures 1–3 show that the theoretical results are in good agreement with the simulation results and support the theoretical analysis in this paper. Figure 1 shows that the asymptotic property of AWP learning is $\epsilon_g \sim O(t^{-1})$, while that of WP learning is $\epsilon_g \sim O(t^{-\frac{1}{3}})$. It can be observed that AWP learning has an outstanding asymptotic property equivalent to that of AdaTron learning. Figures 2 and 3 show the dynamical behaviors of ϵ_g for various η and c in AWP learning. It is clear from Fig. 2 that η has an optimum value, which is approximately $\eta = 2.2$. Figure 3 shows that ϵ_g exhibits strange behavior when $c < 1.0$. On the basis of the concept of weight perturbation, it appears that the approximation for the gradient is improved as c decreases. However, from Eq. (21), this causes a decrease in the update probability. Owing to

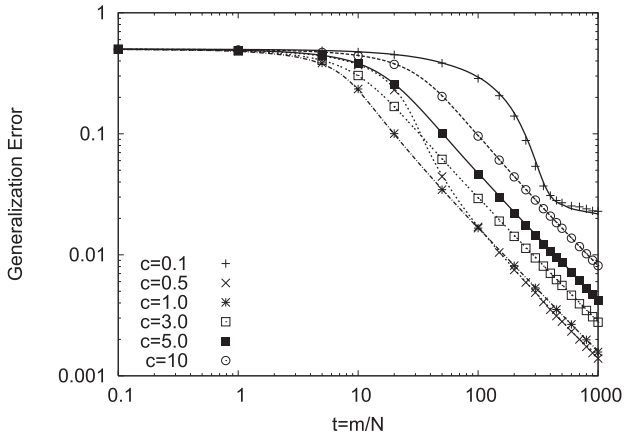

Fig. 3 Dynamical behaviors of generalization error ϵ_g for various c obtained theoretically and by simulation.

the trade-off between these two factors, it does not follow inevitably that the asymptotic property is improved as c decreases. This explains why c has an optimum value in Fig. 3.

7. Conclusion

In this paper, we first proposed AWP learning, which is an adaptive version of WP learning, by considering the difference between perceptron learning and AdaTron learning. As a result of statistical mechanical analysis, it was shown that the asymptotic property of the generalization error is $\epsilon_g \sim O(t^{-1})$, which is an outstanding asymptotic property equivalent to that of AdaTron learning. Moreover, it was observed that the learning rate η has an optimum value of approximately 2.2 and that the dynamical behavior exhibits strange behavior when $c < 1.0$.

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.216–223, 2007.
[3] Y. Maeda, "Simultaneous perturbation optimization methods and their applications," Systems, Control and Information, vol.52, no.2, pp.47–53, 2008.
[4] J.C. Spall, "A stochastic approximation technique for generating maximum likelihood parameter estimates," Proc. American Control Conference, pp.1161–1167, 1987.
[5] J. Alespector, R. Meir, B. Yuhas, A. Jayakumar, and D. Lippe, "A parallel gradient descent method for learning in analog VLSI neural networks," in Advances in Neural Information Processing Systems 5, ed. S.J. Hanson, J.D. Cowan, and C. Lee, pp.836–844, Morgan Kaufmann Publishers, 1993.
[6] G. Cauwenberghs, "A fast stochastic error-descent algorithm for supervised learning and optimization," in Advances in Neural Information Processing Systems 5, ed. S.J. Hanson, J.D. Cowan, and C. Lee, pp.244–251, Morgan Kaufmann Publishers, 1993.
[7] Y. Maeda and R.J.P. De Figueiredo, "Learning rules for neurocontroller via simultaneous perturbation," IEEE Trans. Neural Netw., vol.8, no.5, pp.1119–1130, Sept. 1997.
[8] J. Werfel, X. Xie, and H. Seung, "Learning curves for stochastic gradient descent in linear feedforward networks," Neural Comput., vol.17, pp.2699–2718, 2005.
[9] H. Nishimori, Statistical Physics of Spin Glasses and Information Processing, Oxford University Press, 2001.
[10] S. Miyoshi, H. Hikawa, and Y. Maeda, "Statistical mechanical analysis of simultaneous perturbation learning," IEICE Trans. Fundamentals, vol.E92-A, no.7, pp.1–4, July 2009.
[11] J.K. Anlauf and M. Biehl, "The AdaTron: An adaptive perceptron algorithm," Europhys. Lett., vol.10, no.1, pp.687–692, 1989.
[12] E. Domany, J.L. van Hemmen, and K. Shulten, eds., Models of Neural Networks III, Springer, 1996.
[13] M. Biel and P. Riegler, "On-line learning with a perceptron," Europhys. Lett., vol.28, no.7, pp.525–530, 1994.
[14] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery, Numerical Recipes in C: The Art of Scientific Computing, Cambridge University Press, 2002.