Journal of the Physical Society of Japan **86**, 044002 (2017)
https://doi.org/10.7566/JPSJ.86.044002

Statistical Mechanical Analysis of Online Learning with Weight Normalization in Single Layer Perceptron

Yuki Yoshida[1], Ryo Karakida[1], Masato Okada[1,2,3], and Shun-ichi Amari[2]

[1]*Department of Complexity Science and Engineering, Graduate School of Frontier Sciences, The University of Tokyo, Kashiwa, Chiba 277-8561, Japan*
[2]*RIKEN Brain Science Institute, Wako, Saitama 351-0198, Japan*
[3]*Artificial Intelligence Research Center, AIST, Koto, Tokyo 135-0064, Japan*

(Received October 20, 2016; accepted January 30, 2017; published online March 13, 2017)

Weight normalization, a newly proposed optimization method for neural networks by Salimans and Kingma (2016), decomposes the weight vector of a neural network into a radial length and a direction vector, and the decomposed parameters follow their steepest descent update. They reported that learning with the weight normalization achieves better converging speed in several tasks including image recognition and reinforcement learning than learning with the conventional parameterization. However, it remains theoretically uncovered how the weight normalization improves the converging speed. In this study, we applied a statistical mechanical technique to analyze on-line learning in single layer linear and nonlinear perceptrons with weight normalization. By deriving order parameters of the learning dynamics, we confirmed quantitatively that weight normalization realizes fast converging speed by automatically tuning the effective learning rate, regardless of the nonlinearity of the neural network. This property is realized when the initial value of the radial length is near the global minimum; therefore, our theory suggests that it is important to choose the initial value of the radial length appropriately when using weight normalization.

1. Introduction

In recent years, large neural networks are commonly used for various tasks in the name of Deep Learning.[1] The success of deep learning is highly indebted to improvements of algorithms for speeding up learning: they include various modifications to gradient descent,[2–6] and several normalization methods.[7–9] In particular, the weight normalization (WN) proposed by Salimans and Kingma,[8] which re-parametrizes the weight vector as explained below, is spotlighted in terms of easiness of implementation and introduction to various conventional network structures. In most neural networks, each neuron's output can be represented as $y = g(\mathbf{W} \cdot \mathbf{x} + b)$, where $\mathbf{x}$ is an input vector and the activation function g is nonlinear in general. The standard steepest descent method updates its weight vector $\mathbf{W}$ as $\Delta\mathbf{W} = -\eta \frac{\partial \varepsilon}{\partial \mathbf{W}}$, where ε is a loss function and $\eta > 0$ is a learning rate. In contrast, in WN, $\mathbf{W}$ is decomposed as $\mathbf{W} = r\frac{\mathbf{V}}{|\mathbf{V}|}$, and then steepest descent optimization proceeds in accordance with the gradient of r and $\mathbf{V}$ instead of $\mathbf{W}$, that is, $\Delta r = -\eta \frac{\partial \varepsilon}{\partial r}$ and $\Delta\mathbf{V} = -\eta \frac{\partial \varepsilon}{\partial \mathbf{V}}$. This newly proposed optimization method is known to speed up the convergence of the loss function in conventional network structures and machine learning tasks such as image recognition and reinforcement learning. However, it remains unclear why this method works well.

Biehl and Schwarze[10] and Saad and Solla[11] established useful techniques on the basis of statistical mechanics, with which we can derive the dynamical equations of order parameters that represent the macroscopic state of the weight vector. Using the techniques, they found analytically and discussed the dynamics of on-line learning in a single layer perceptron and a (two-layered) soft committee machine that learns the input-output relationship of a "teacher network" that has the same structure as a learning one, although their analysis was limited to the conventional steepest descent methods with ordinary parameterization of weight vectors.

In this study, we apply their technique to WN and analyze quantitatively the dynamical evolution of on-line learning in single layer perceptrons. We perform linear stability analysis on a global minimum of the loss function and determine the converging speed of order parameters towards the global minimum. In WN, it shows that an effective learning rate appears that is automatically tuned, and that there exists an optimal initial value of an order parameter for fast converging.

2. Model

2.1 Student–teacher network formulation

In this study, we focus only on single layer perceptron. That is, we consider a neural network that receives input data $\mathbf{x} \in \mathbb{R}^N$, calculates output $s = g(\mathbf{J} \cdot \mathbf{x})$ (we assume the activation function $g : \mathbb{R} \to \mathbb{R}$ is non-constant and weakly monotonous), and learns $\mathbf{J}$ by teacher data. We treat an ideal situation, in which the teacher data t is determined as $t = g(\mathbf{B} \cdot \mathbf{x})$; in other words, the learning network (the "student network") learns the input-output relationship of the "teacher network", which has the same structure as the student one and the original fixed weight $\mathbf{B}$ [Fig. 1(a)]. We use the squared loss function $\varepsilon = \frac{1}{2}(t - s)^2$. (The choice of a loss function is not critical; see Appendix B for general case.)

2.2 Statistical mechanical formulation

2.2.1 Stochastic gradient descent

For the statistical mechanical formulation of on-line learning, we introduce further idealization. We assume that the dimension of input data N is very large, and each element of input data $\mathbf{x}$ is generated in accordance with i.i.d. normal distribution, $\mathcal{N}(x_i|0, 1/N)$. (Note that $|\mathbf{x}| \approx 1$.) We suppose $|\mathbf{B}| = \sqrt{N}$ and define $l(\alpha)$ and $R(\alpha)$ by $|\mathbf{J}| = \sqrt{N}\, l(\alpha)$ and $\mathbf{B} \cdot \mathbf{J} = N\, l(\alpha)R(\alpha)$, where α represents time.[10,11] $l(\alpha)$ and $R(\alpha)$ are the order parameters; the former one is the measure for the length (norm) of the weight vector of the student network, and the latter one is the direction cosine between the weight vectors of student and teacher [Fig. 1(b)]. The initial values of l and R (l_0 and R_0, respectively) depend on how $\mathbf{J}$ is initialized; the value of R_0 converges towards 0 with $N \to \infty$, as long as we choose $\mathbf{J}$ from a spherically symmetric distribution.

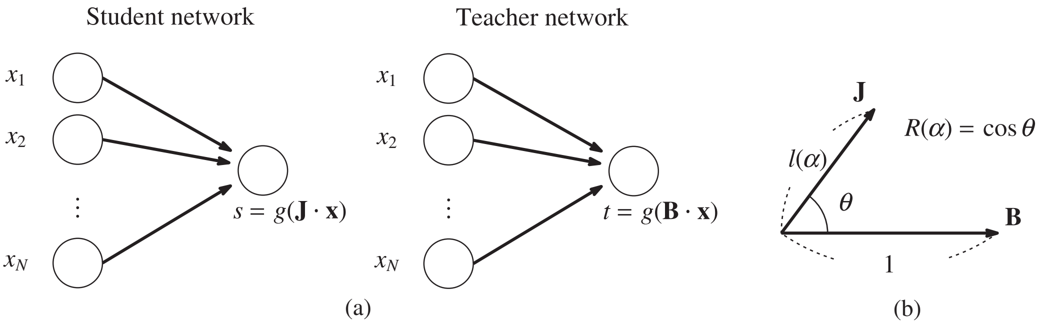

Fig. 1. (a) Student network and teacher network. (b) Geometrical interpretation of order parameters $l(\alpha)$ and $R(\alpha)$.

In the next section, we derive the dynamical equations that govern the order parameters l and R, which capture the macroscopic state of the system, from the dynamics of microscopic variables (the weight vector) of the system.

2.2.2 *Weight normalization*

In WN, weight vector $\mathbf{J}$ is decomposed into radial length r and direction vector $\mathbf{V}$ as $\mathbf{J} = r\frac{\mathbf{V}}{|\mathbf{V}|}$, and the gradients of r and $\mathbf{V}$ are then used to perform gradient descent optimization. Since we put $|\mathbf{J}| = \sqrt{N}\,l(\alpha)$ in the previous section, r equals to $\sqrt{N}\,l(\alpha)$, then we call $l(\alpha)$ "radial length", as well as r. We define $z(\alpha)$ to represent the norm of $\mathbf{V}$, as $|\mathbf{V}| = \sqrt{N}\,z(\alpha)$. This additional order parameter $z(\alpha)$ shows up because of the redundancy in radial parameterization $\mathbf{J} = r\frac{\mathbf{V}}{|\mathbf{V}|}$, where the norm of $\mathbf{V}$ is not normalized. In WN, three order parameters $l(\alpha)$, $R(\alpha)$, and $z(\alpha)$ appear in the following macroscopic dynamical equations.

3. Theory

In this section, we derive the dynamical equations of the parameters described above.

3.1 *Dynamical equations of order parameters in SGD*

We follow the argument by Biehl and Schwarze[10] and Saad and Solla[11] in this subsection. The update rule of online learning based on vanilla SGD is written as

$$\Delta\mathbf{J} = -\eta\frac{d\varepsilon}{d\mathbf{J}} = \eta g'(\mathbf{J}\cdot\mathbf{x})(t-s)\mathbf{x}, \tag{1}$$

which gives the update rule of order parameters l and R:

$$\begin{aligned} N\Delta l^2 &= |\mathbf{J}(\alpha+1)|^2 - |\mathbf{J}(\alpha)|^2 \\ &= |\mathbf{J}(\alpha) + \eta g'(\mathbf{J}(\alpha)\cdot\mathbf{x})(t-s)\mathbf{x}|^2 - |\mathbf{J}(\alpha)|^2 \\ &= 2\eta g'(\mathbf{J}(\alpha)\cdot\mathbf{x})(t-s)\mathbf{J}(\alpha)\cdot\mathbf{x} \\ &\quad + \eta^2 g'(\mathbf{J}(\alpha)\cdot\mathbf{x})^2(t-s)^2|\mathbf{x}|^2 \\ &= 2\eta g'(lu)(t-s)lu + \eta^2 g'(lu)^2(t-s)^2|\mathbf{x}|^2, \end{aligned}$$

$$\begin{aligned} N\Delta(lR) &= \mathbf{B}\cdot\mathbf{J}(\alpha+1) - \mathbf{B}\cdot\mathbf{J}(\alpha) = \mathbf{B}\cdot\Delta\mathbf{J} \\ &= \eta g'(\mathbf{J}(\alpha)\cdot\mathbf{x})(t-s)\mathbf{B}\cdot\mathbf{x} \\ &= \eta g'(lu)(t-s)v, \end{aligned} \tag{2}$$

where we define u and v as $\mathbf{J}\cdot\mathbf{x} = lu$ and $\mathbf{B}\cdot\mathbf{x} = v$. Since the right hand sides of these equations are $O(N^0)$, the difference terms Δl^2 and $\Delta(lR)$ are $O(N^{-1})$, and therefore we can replace these difference equations with differential ones with $N \to \infty$:

$$\begin{aligned} N\frac{d}{d\alpha}l^2 &= 2\eta A_1 + 2\eta^2 A_2, \\ N\frac{d}{d\alpha}lR &= \eta A_3, \end{aligned} \tag{3}$$

where

$$\begin{aligned} A_1(l,R) &= \langle g'(lu)(g(v) - g(lu))lu\rangle, \\ A_2(l,R) &= \frac{1}{2}\langle g'(lu)^2(g(v) - g(lu))^2\rangle, \\ A_3(l,R) &= \langle g'(lu)(g(v) - g(lu))v\rangle. \end{aligned} \tag{4}$$

Here the brackets $\langle\cdot\rangle$ represent the expectation when $\mathbf{x}$ follows $\mathcal{N}(x_i|0, 1/N)$, that is, when (u, v) follows $\mathcal{N}(\mathbf{0}, (\begin{smallmatrix} 1 & R \\ R & 1 \end{smallmatrix}))$. These differential equations are what we wanted. Note that the generalization error ε_g is represented as

$$\varepsilon_g = \frac{1}{2}\langle(t-s)^2\rangle = \frac{1}{2}\langle(g(v) - g(lu))^2\rangle. \tag{5}$$

All expectation terms appearing in (4), (5) have forms $I_3(y_1, y_2, y_3) := \langle g'(y_1)y_2 g(y_3)\rangle$ or $I_4(y_1, y_2, y_3) := \langle g'(y_1)^2 g(y_2)g(y_3)\rangle$, where y_1, y_2, y_3 is either lu, v, or 0. The I_3 and I_4 can be analytically determined for some activation function g; when (y_1, y_2, y_3) follows a multivariate normal distribution $\mathcal{N}((y_1, y_2, y_3)|0, C)$ where C is covariance matrix, the identity function $g(x) = x$ gives

$$\begin{aligned} I_3 &= \langle y_2 y_3\rangle = C_{23}, \\ I_4 &= \langle y_2 y_3\rangle = C_{23}. \end{aligned} \tag{6}$$

The error function $g(x) = \mathrm{erf}(x/\sqrt{2})$ implies

$$\begin{aligned} I_3 &= \frac{2}{\pi}\cdot\frac{1}{\sqrt{(1+C_{11})(1+C_{33}) - C_{13}^2}}\,\frac{C_{23}(1+C_{11}) - C_{12}C_{13}}{1+C_{11}}, \\ I_4 &= \frac{4}{\pi^2}\cdot\frac{1}{\sqrt{1+2C_{11}}}\arcsin\frac{(1+2C_{11})C_{23} - 2C_{12}C_{13}}{\sqrt{(1+2C_{11})(1+C_{22}) - 2C_{12}^2}\sqrt{(1+2C_{11})(1+C_{33}) - 2C_{13}^2}} \end{aligned} \tag{7}$$

as shown in Saad and Solla.[11] In the case of $g(x) = \mathrm{ReLU}(x)$ $(:= \max\{0, x\})$, which is commonly used as the activation function in recent neural networks, we found the following formula (see Appendix A for derivation):

$$
\begin{aligned}
I_3 &= C_{23}\left(\frac{1}{4} + \frac{1}{2\pi}\arcsin\frac{C_{13}}{\sqrt{C_{11}C_{33}}}\right) + \frac{1}{2\pi}\cdot\frac{C_{12}}{C_{11}}\sqrt{C_{11}C_{33} - C_{13}^2},\\
I_4 &= C_{23}\left[\frac{1}{8} + \frac{1}{4\pi}\left(\arcsin\frac{C_{12}}{\sqrt{C_{11}C_{22}}} + \arcsin\frac{C_{13}}{\sqrt{C_{11}C_{33}}} + \arcsin\frac{C_{23}}{\sqrt{C_{22}C_{33}}}\right)\right]\\
&\quad + \frac{1}{4\pi}\left(\frac{C_{13}}{C_{11}}\sqrt{C_{11}C_{22} - C_{12}^2} + \frac{C_{12}}{C_{11}}\sqrt{C_{11}C_{33} - C_{13}^2} + \sqrt{C_{22}C_{33} - C_{23}^2}\right).
\end{aligned}
\tag{8}
$$

3.2 Dynamical equations of order parameters in WN

In the case of WN, the update rule of direction vector $\mathbf{V}$ and radial parameter r is

$$
\begin{aligned}
\Delta r &= -\eta\frac{d\varepsilon}{dr} = \eta g'(\mathbf{J}\cdot\mathbf{x})(t-s)\frac{\mathbf{J}\cdot\mathbf{x}}{r},\\
\Delta\mathbf{V} &= -\eta\frac{d\varepsilon}{d\mathbf{V}} = \eta g'(\mathbf{J}\cdot\mathbf{x})(t-s)\left(\frac{r}{|\mathbf{V}|}\mathbf{x} - \frac{\mathbf{J}\cdot\mathbf{x}}{|\mathbf{V}|^2}\mathbf{V}\right),
\end{aligned}
\tag{9}
$$

which provides the update rule of order parameters l, R, and z:

$$
\begin{aligned}
N\Delta l &= \sqrt{N}\Delta r = \eta g'(\mathbf{J}\cdot\mathbf{x})(t-s)\frac{\mathbf{J}\cdot\mathbf{x}}{l}\\
&= \eta g'(lu)(t-s)u,\\
N\Delta(Rz) &= \Delta(\mathbf{B}\cdot\mathbf{V}) = \eta g'(\mathbf{J}\cdot\mathbf{x})(t-s)\left(\frac{r}{|\mathbf{V}|}\mathbf{x} - \frac{\mathbf{J}\cdot\mathbf{x}}{|\mathbf{V}|^2}\mathbf{V}\right)\cdot\mathbf{B}\\
&= \eta g'(lu)(t-s)\left(\frac{lv}{z} - \frac{lRu}{z}\right),\\
N\Delta(z^2) &= |\mathbf{V}(\alpha+1)|^2 - |\mathbf{V}(\alpha)|^2\\
&= 2\mathbf{V}(\alpha)\cdot\Delta\mathbf{V} + |\Delta\mathbf{V}|^2 = |\Delta\mathbf{V}|^2\\
&= \eta^2 g'(\mathbf{J}\cdot\mathbf{x})^2(t-s)^2\left[\frac{r^2}{|\mathbf{V}|^2}|\mathbf{x}|^2\right.\\
&\quad\left. - 2\frac{r\mathbf{J}\cdot\mathbf{x}}{|\mathbf{V}|^3}\mathbf{V}\cdot\mathbf{x} + \frac{(\mathbf{J}\cdot\mathbf{x})^2}{|\mathbf{V}|^2}\right]\\
&= \eta^2 g'(lu)^2(t-s)^2\left(\frac{l^2}{z^2}|\mathbf{x}|^2 - \frac{l^2u^2}{Nz^2}\right)
\end{aligned}
\tag{10}
$$

(we set $\mathbf{J}\cdot\mathbf{x} = lu$ and $\mathbf{B}\cdot\mathbf{x} = v$ again). These difference terms are $O(N^{-1})$; thus, we can replace these equations with differential equations with $N \to \infty$:

$$
\begin{aligned}
N\frac{d}{d\alpha}l^2 &= 2\eta A_1,\\
N\frac{d}{d\alpha}Rz &= \eta\left(\frac{l}{z}A_3 - \frac{R}{z}A_1\right),\\
N\frac{d}{d\alpha}z^2 &= \eta^2\frac{2l^2}{z^2}A_2.
\end{aligned}
\tag{11}
$$

The definition of terms A_i are the same as (4). Again, all expectation terms have forms like I_3 or I_4, which can be determined for some g. Note that z is monotonously increasing since $A_2 \geq 0$.

3.3 Linear stability analysis

In our system of student and teacher single layer perceptrons, the generalization error ε_g equals to 0 if and only if $\mathbf{J} = \mathbf{B}$; in other words, $\mathbf{J} = \mathbf{B}$ is the unique global minimum. Values of order parameters at this global minimum are $(l, R) = (1, 1)$. For the system (3) for SGD, $(l, R) = (1, 1)$ is a steady state, and for the system (11) for WN, $(l, R, z) = (1, 1, z)$ are steady states for all z. We perform stability analysis at these steady points corresponding to the global minimum to evaluate quantitatively the speed of converging towards global optimality.

For the system (3) for SGD, the stability matrix at $(l, R) = (1, 1)$, and its eigenvalues and eigenvectors are given as

$$
P_{\mathrm{SGD}} = \begin{pmatrix} \eta\dfrac{\partial A_1}{\partial l} & \eta\dfrac{\partial A_1}{\partial R} + \eta^2\dfrac{\partial A_2}{\partial R}\\ 0 & \eta\left(\dfrac{\partial A_3}{\partial R} - \dfrac{\partial A_1}{\partial R}\right) - \eta^2\dfrac{\partial A_2}{\partial R} \end{pmatrix},
$$

$$
\lambda_1 = \eta\frac{\partial A_1}{\partial l}, \quad \mathbf{e}_1 = \begin{pmatrix} 1\\ 0 \end{pmatrix},
$$

$$
\lambda_2 = \eta\left(\frac{\partial A_3}{\partial R} - \frac{\partial A_1}{\partial R}\right) - \eta^2\frac{\partial A_2}{\partial R} = \frac{\partial A_2}{\partial R}\eta(\eta_c - \eta),
$$

$$
\mathbf{e}_2 = \begin{pmatrix} \dfrac{\partial A_1}{\partial R} + \eta\dfrac{\partial A_2}{\partial R}\\ -\dfrac{\partial A_1}{\partial l} + \dfrac{\lambda_2}{\eta} \end{pmatrix},
$$

where

$$
\eta_c := \frac{\partial(A_3 - A_1)}{\partial R}\bigg/\frac{\partial A_2}{\partial R},
$$

where the derivatives of A_i are evaluated at $(l, R) = (1, 1)$. For the system (11) for WN, the stability matrix at $(l, R, z) = (1, 1, z_\infty)$, and its eigenvalues and eigenvectors are calculated as

$$
P_{\mathrm{WN}} = \begin{pmatrix} \eta\dfrac{\partial A_1}{\partial l} & \eta\dfrac{\partial A_1}{\partial R} & 0\\ 0 & \dfrac{\eta}{z_\infty^2}\left(\dfrac{\partial A_3}{\partial R} - \dfrac{\partial A_1}{\partial R}\right) - \dfrac{\eta^2}{z_\infty^4}\dfrac{\partial A_2}{\partial R} & 0\\ 0 & \dfrac{\eta^2}{z_\infty^3}\dfrac{\partial A_2}{\partial R} & 0 \end{pmatrix},
$$

$$
\lambda_1 = \eta\frac{\partial A_1}{\partial l}, \quad \mathbf{e}_1 = \begin{pmatrix} 1\\ 0\\ 0 \end{pmatrix},
$$

$$
\lambda_2 = \frac{\eta}{z_\infty^2}\left(\frac{\partial A_3}{\partial R} - \frac{\partial A_1}{\partial R}\right) - \frac{\eta^2}{z_\infty^4}\frac{\partial A_2}{\partial R} = \frac{\partial A_2}{\partial R}\frac{\eta}{z_\infty^2}\left(\eta_c - \frac{\eta}{z_\infty^2}\right),
$$

$$
\mathbf{e}_2 = \begin{pmatrix} \dfrac{\partial A_1}{\partial R}\\ -\dfrac{\partial A_1}{\partial l} + \dfrac{\lambda_2}{\eta}\\ \dfrac{\eta^2}{z_\infty^3}\dfrac{\partial A_2}{\partial R}\left(-\dfrac{1}{\lambda_2}\dfrac{\partial A_1}{\partial l} + \dfrac{1}{\eta}\right) \end{pmatrix},
$$

$$
\lambda_3 = 0, \quad \mathbf{e}_3 = \begin{pmatrix} 0\\ 0\\ 1 \end{pmatrix},
$$

Table I. Converging speeds towards global minimum of order parameters l and R. All derivatives of A_i are evaluated at global minimum.

	SGD	WN
Direction cosine R	$-\frac{\partial A_2}{\partial R}\eta(\eta_c-\eta)$	$-\frac{\partial A_2}{\partial R}\frac{\eta}{z_\infty^2}\left(\eta_c-\frac{\eta}{z_\infty^2}\right)$
Radial length l	$\min\left\{-\frac{\partial A_1}{\partial l}\eta, -\frac{\partial A_2}{\partial R}\eta(\eta_c-\eta)\right\}$	$\min\left\{-\frac{\partial A_1}{\partial l}\eta, -\frac{\partial A_2}{\partial R}\frac{\eta}{z_\infty^2}\left(\eta_c-\frac{\eta}{z_\infty^2}\right)\right\}$

where the derivatives of A_i are evaluated at $(l, R, z) = (1, 1, z_\infty)$. Since $A_2 = \frac{1}{2}\langle[g'(lu)(g(v)-g(lu))]^2\rangle \geq 0$ and $A_3 - A_1 = \langle g'(lu)(g(v)-g(lu))(v-lu)\rangle \geq 0$ (note that g is monotonous) equal to 0 at global minimum (1, 1), it holds that $\frac{\partial A_2}{\partial R}, \frac{\partial(A_3-A_1)}{\partial R} \leq 0$. Thus, for SGD system, the necessary and sufficient condition for the stability of the global minimum is $\eta < \eta_c = \frac{\partial(A_3-A_1)}{\partial R} \,/\, \frac{\partial A_2}{\partial R}$. The value of η_c depends on activation function g; for example, $\eta_c = 2$ for $g(x) = x$ and $g(x) = \mathrm{ReLU}(x)$, and $\eta_c = \sqrt{5/3}\pi$ for $g(x) = \mathrm{erf}(x/\sqrt{2})$. In contrast, the stability condition of the global minimum for WN system is $\eta/z_\infty^2 < \eta_c$.

We can evaluate the speeds of order parameters l and R converging towards the global optimum when it is linearly stable, using eigenvalues and eigenvectors of the stability matrix calculated above. For SGD and WN cases, the R-component of the first eigenvector $\mathbf{e}_1$ equals to 0, therefore R converges towards the global optimum 1 at the speed of $-\lambda_1$. [We call a variable behaving like $O(e^{\lambda\alpha})$ as "converging at the speed of λ".] In contrast, l converges towards 1 at the speed of $\min\{-\lambda_1, -\lambda_2\}$. The converging speeds of l and R while performing SGD and WN are summarized in Table I.

Comparing WN and SGD, the converging speed of R, the direction cosine, is different; its effective learning rate is η in SGD and η/z_∞^2 in WN. Since the initial value of z is usually small (Salimans and Kingma[8]) recommend 0.05), the value of η/z^2 is initially large and then decreases during learning because z is monotonously increasing. In SGD, since $-\lambda_2 \propto \eta(\eta_c - \eta)$, a learning rate larger than η_c prevents convergence, while too small a learning rate takes a long time to converge. The optimal learning rate which maximizes $-\lambda_2$ is $\eta = \frac{\eta_c}{2}$, although the value of η_c is generally unknown. Contrary to this, while performing WN the effective learning rate η/z^2 automatically decreases to near the optimal learning rate $\frac{\eta_c}{2}$. This mechanism seems to realize η-independent fast converging.

Meanwhile, the converging speed of l, radial length, in WN cannot be larger than $-\frac{\partial A_1}{\partial l}\eta$—that in SGD. Hence, if the initial value of the radial length is near optimal ($l = 1$), WN seems to greatly outperform SGD; otherwise, it is probable that WN yields the same converging speed as SGD does. If the initial value of l is too far from optimal, WN might be slower than SGD because η/z^2 may become too small and turn into a bottleneck in converging.

In the next section, we confirm the correspondence between numerical solutions of differential equations (3) and (11) about order parameters and simulation results about the original microscopic system, then examine the hypotheses described in this section.

4. Experimental Results

4.1 *Consistency between simulation and numerical solution*

We performed numerical simulations of original microscopic systems ($N = 10000$) for SGD and WN. We set the weight vector of teacher $\mathbf{B}$, the initial weight vector of student $\mathbf{J}(0)$, and the initial direction vector $\mathbf{V}(0)$ by sampling their elements in accordance with $\mathcal{N}(B_i|0, 1)$, $\mathcal{N}(J_i|0, l_0^2)$, and $\mathcal{N}(V_i|0, z_0^2)$, then normalizing them so that they suffice $|\mathbf{B}| = \sqrt{N}$, $|\mathbf{J}(0)| = \sqrt{N}\,l_0$, and $|\mathbf{V}(0)| = \sqrt{N}\,z_0$. We plotted the time course of order parameters l, R, and z and compared them with numerical solutions of differential equations (3) and (11) about order parameters (Fig. 2). We confirmed their good agreement.

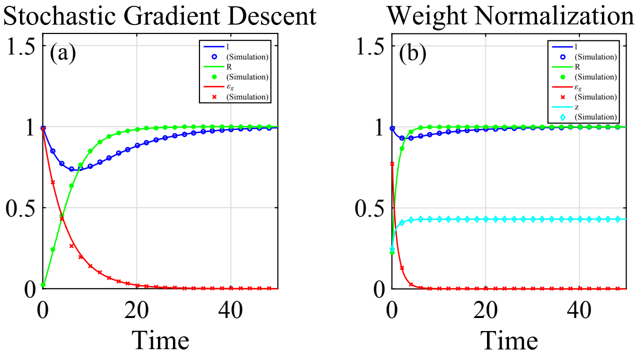

Fig. 2. (Color online) Time course of order parameters and generalization error in (a) SGD and (b) WN. blue: l, green: R, red: ε_g, cyan [in (b)]: z. Solid lines represent numerical solutions of differential equations (3) and (11). Markers represent simulation results ($N = 10000$). Initial value of l is $l_0 = 1.0$. In (b), initial value of z is $z_0 = 0.05$. For all cases $\eta = 0.1$ and $g(x) = x$.

4.2 *Dependence of converging speed on learning rate and initial radial length*

Next, we computed numerical solutions of differential equations (3) and (11), investigated how the convergence speed of the generalization error towards 0 depends on η and l_0, and compared the results in the cases of SGD and WN. [Fig. 3 for $g(x) = x$, and Fig. 4 for $g(x) = \mathrm{erf}(x/\sqrt{2})$; in the case of $g(x) = \mathrm{ReLU}(x)$, we got almost the same results as Fig. 3.] [See Fig. A·1 for dependence on (η, l_0).]

4.2.1 *Dependence on learning rate*

When the learning rate η is too large ($\eta > \eta_c$), SGD does not allow the weight vector to converge towards the global minimum, while WN realizes the η-independent converging speed as the same as that in a case of $\eta \approx \eta_c$.

When the weight vector converges to the global optimum, the "effective learning rate" η/z^2 of direction cosine R decreases to near $\eta_c/2$ for a wide range of parameter values [yellow regions of Fig. A·1(d)]. This demonstrates that the "automatic tuning of the learning rate" indeed serves for the η-independent converging speed.

4.2.2 *Dependence on initial radial length*

Only when the initial value of the radial length is near

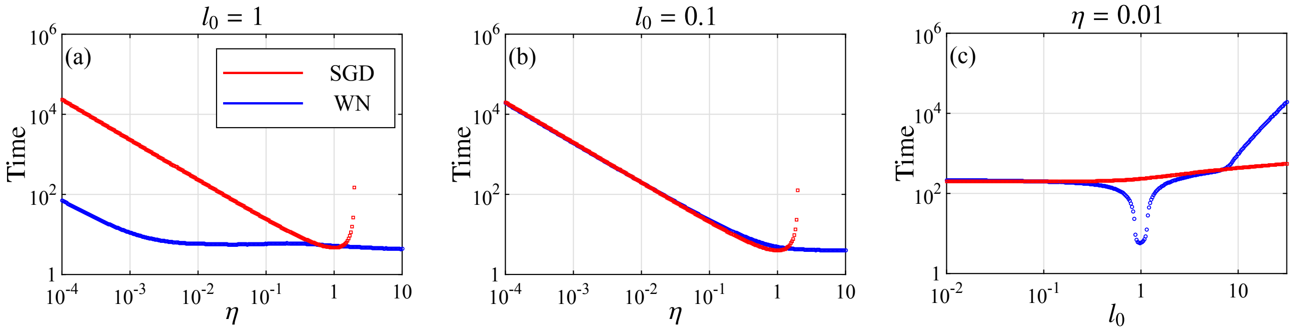

Fig. 3. (Color online) (a, b) Dependence of elapsed time until generalization error ε_g falls below 0.01, on (a, b) learning rate η and (c) initial radial length l_0. In (a) and (b), l_0 is fixed to 1 (global minimum) and 0.1, respectively. In (c), η is fixed to 0.01. Red symbols: SGD. Blue symbols: WN. Activation function is $g(x) = x$.

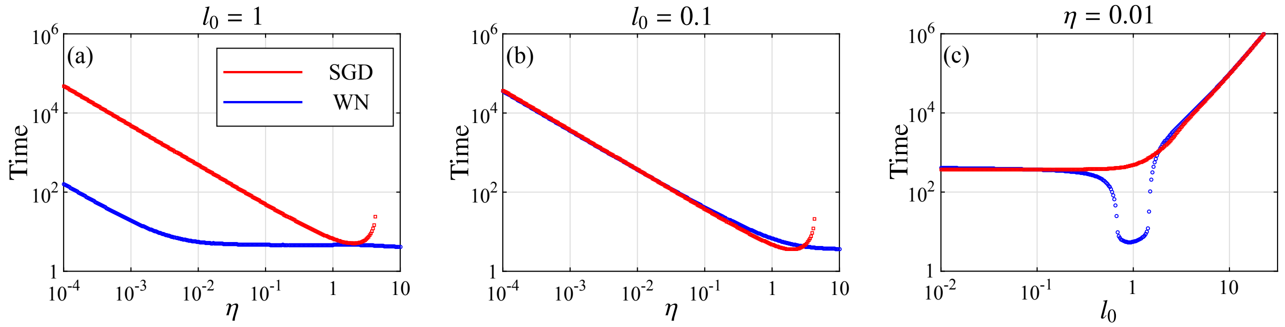

Fig. 4. (Color online) (a, b) Dependence of elapsed time until generalization error ε_g falls below 0.01, on (a, b) learning rate η and (c) initial radial length l_0. In (a) and (b), l_0 is fixed to 1 (global minimum) and 0.1, respectively. In (c), η is fixed to 0.01. Red symbols: SGD. Blue symbols: WN. Activation function is $g(x) = \mathrm{erf}(x/\sqrt{2})$.

optimal, a wide range of values of η [$10^{-2} < \eta < 10$ in Figs. 3(a) and 4(a)] yields the same converging speed. Learning with p times smaller η takes $1/p$ times longer in the case of SGD, while in the case of WN such elongation does not occur (within the range of η mentioned above).

In contrast, when the initial value of the radial length is significantly larger than the global minimum, there are cases in which WN takes more time than SGD to converge [$l_0 > 10$ in Fig. 3(c) and yellow region in Fig. A·1(c)]. Figure 3(d) indicates that, in such cases, the effective learning rate η/z^2 of the direction cosine parameter R reaches a value below η that regulates l's converging speed, showing that an effective learning rate decreased too much turns into a bottleneck in converging. On the contrary, smaller initial values of the radial length than optimum do not cause such a delay. Therefore, when performing WN we should carefully choose the initial value of the radial length; it seems that one near the optimum is best, and we should at least avoid one much larger than the optimum.

5. Conclusion

Using the statistical mechanical method for online learning established by Biehl and Schwarze[10] and Saad and Solla,[11] we analyzed the weight normalization proposed by Salimans and Kingma[8] in the simplest situation: that of single layer perceptrons. We revealed quantitatively that the "automatic tuning" of the effective learning rate of the direction cosine plays a critical role in fast convergence. Our theoretical argument does not depend on any specific form of the activation function. Thus, the mechanism of WN we discussed is not dependent on activation and seems to be a proper characteristic of WN.

In this study, we considered single layer perceptrons, although two-or-more-layered networks are often used for practical applications. Unlike a single layer perceptron, a multilayer perceptron has singular points and plateaus in its loss surface, which prevent converging towards its global minimum.[12] The fact that WN exhibits faster convergence even in multilayer networks[8] suggests that WN performs well even under the existence of singular points and plateaus. The statistical mechanical approach was used for a multilayer network by Biehl and Schwarze[10] and Saad and Solla[11] who addressed a two-layered soft committee machine and Riegler and Biehl[12] who analyzed the learning of two-layered perceptrons. Their methods seem to be applicable to the analysis of learning in two-or-more-layered perceptrons with WN.

The advantage of the statistical mechanical method is that we can briefly inspect the behavior of essential quantities like radial length and direction cosine. This statistical mechanical method is also used for theoretical analysis of natural gradients[13] and analysis of learning in networks with ReLU activation.[14] It might be useful for analysis of other recent optimization algorithms.[2–5,7,9] Further success of this approach is expected.

Acknowledgment

We would like to thank Professor Kazuyuki Hara for his thoughtful comments on the paper.

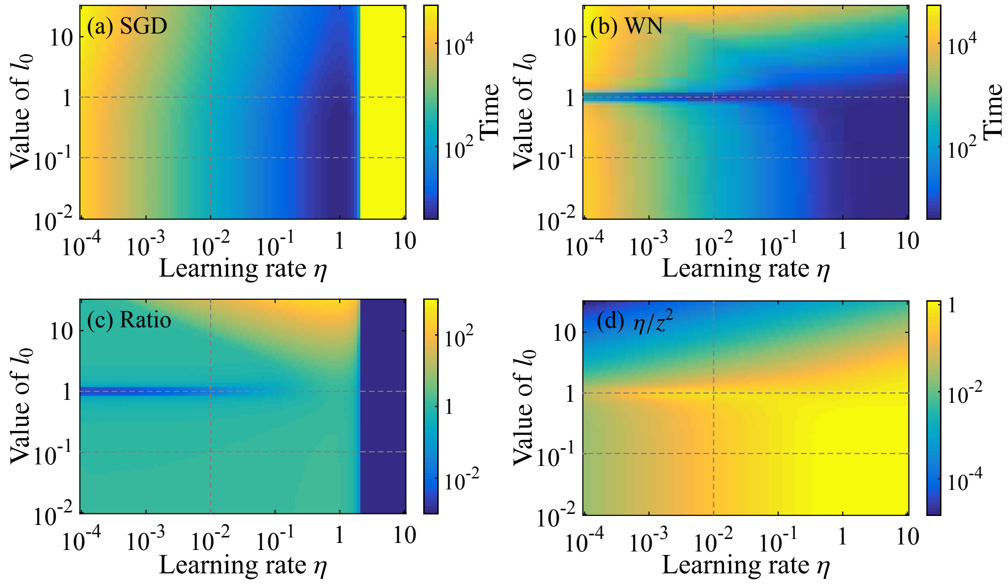

Fig. A·1. (Color online) (a, b) Elapsed time until generalization error ε_g falls below 0.01, shown as function of learning rate η and initial radial length l_0. (a) SGD, (b) WN. (c) Ratio of (a) and (b) (WN/SGD). (d) Value of effective learning rate η/z^2 when ε_g reaches <0.01 in WN. Activation function is $g(x) = x$ in all cases. Three dashed lines correspond to Figs. 3(a), 3(b), and 3(c).

Appendix A: Calculation of Expectation Terms in the Case of ReLU Activation

We calculate expectation terms $I_3(y_1, y_2, y_3) := \langle g'(y_1)y_2g(y_3)\rangle$ and $I_4(y_1, y_2, y_3) := \langle g'(y_1)^2g(y_2)g(y_3)\rangle$, where (y_1, y_2, y_3) follows multivariate normal distribution $\mathcal{N}((y_1, y_2, y_3)|0, C)$ and $g(x) = \mathrm{ReLU}(x)$ $(:= \max\{0, x\})$.

For I_3,

$$\begin{aligned}I_3 &= \langle g'(y_1)y_2g(y_3)\rangle\\ &= \int_{y_1,y_3>0} y_2y_3\mathcal{N}(\mathbf{y}|\mathbf{0}, C)\,d\mathbf{y}\\ &= \frac{1}{(2\pi)^{3/2}\sqrt{|C|}}\int_{y_1,y_3>0} y_2y_3\exp\left(-\frac{1}{2}\mathbf{y}^{\mathrm{T}}C^{-1}\mathbf{y}\right)d\mathbf{y}\\ &= -\frac{1}{(2\pi)^{3/2}\sqrt{|C|}}\frac{\partial}{\partial C_{23}^{-1}}\int_{y_1,y_3>0}\exp\left(-\frac{1}{2}\mathbf{y}^{\mathrm{T}}C^{-1}\mathbf{y}\right)d\mathbf{y}\\ &= -\frac{1}{\sqrt{|C|}}\frac{\partial}{\partial C_{23}^{-1}}(\sqrt{|C|}\,P_3(C)). \qquad (\mathrm{A{\cdot}1})\end{aligned}$$

Here the term $P_3(C) := \int_{y_1,y_3>0}\mathcal{N}(\mathbf{y}|\mathbf{0}, C)\,d\mathbf{y}$ is a "quadrant probability" of multivariate distribution with zero mean and is calculated with the formula:[15]

$$P_3(C) = \frac{1}{4} + \frac{1}{2\pi}\arcsin\frac{C_{13}}{\sqrt{C_{11}C_{33}}}. \qquad (\mathrm{A{\cdot}2})$$

With

$$\begin{aligned}&-\frac{1}{\sqrt{|C|}}\frac{\partial}{\partial C_{23}^{-1}}\sqrt{|C|}\\ &\quad= -\frac{1}{\sqrt{|C|}}\frac{\partial}{\partial C_{23}^{-1}}\left(\sqrt{|C|}\int_{\mathbb{R}^3}\mathcal{N}(\mathbf{y}|\mathbf{0}, C)\,d\mathbf{y}\right)\\ &\quad= \int_{\mathbb{R}^3} y_2y_3\mathcal{N}(\mathbf{y}|\mathbf{0}, C)\,d\mathbf{y} = C_{23},\end{aligned}$$

Eq. (A·1) implies

$$I_3 = C_{23}P_3(C) - \frac{\partial P_3(C)}{\partial C_{23}^{-1}},$$

and by substituting (A·2) for $P_3(C)$, we obtain the first row of Eq. (8).

In the same way, for I_4 we get

$$I_4 = C_{23}P_4(C) - \frac{\partial P_4(C)}{\partial C_{23}^{-1}},$$

where

$$P_4(C) := \int_{y_1,y_2,y_3>0}\mathcal{N}(\mathbf{y}|\mathbf{0}, C)\,d\mathbf{y},$$

and by using the formula[15]

$$\begin{aligned}P_4(C) = \frac{1}{8} + \frac{1}{4\pi}\Bigg(&\arcsin\frac{C_{12}}{\sqrt{C_{11}C_{22}}} + \arcsin\frac{C_{13}}{\sqrt{C_{11}C_{33}}}\\ &+ \arcsin\frac{C_{23}}{\sqrt{C_{22}C_{33}}}\Bigg),\end{aligned}$$

we can derive the second row of Eq. (8).

Appendix B: Case of Arbitrary Loss Functions

In the main text, we assumed the squared loss function $\varepsilon = \frac{1}{2}(t - s)^2$. However, there are many other loss functions which are commonly used, such as the softmax cross entropy loss in classification tasks. In this section, we discuss the case of arbitrary loss function $\varepsilon(s, t)$, where s is the student's output, and t is the teacher's output.

The differential equations (3) and (11), which describe the dynamics of order parameters, still hold if we replace the definition of terms A_i with

$$A_1(l, R) = -\left\langle g'(lu)lu\left.\frac{\partial\varepsilon}{\partial s}\right|_{(s,t)=(g(lu),g(v))}\right\rangle,$$

$$A_2(l, R) = \frac{1}{2}\left\langle g'(lu)^2\left[\left.\frac{\partial\varepsilon}{\partial s}\right|_{(s,t)=(g(lu),g(v))}\right]^2\right\rangle,$$

$$A_3(l, R) = -\left\langle g'(lu)v\left.\frac{\partial\varepsilon}{\partial s}\right|_{(s,t)=(g(lu),g(v))}\right\rangle.$$

Whether these new expectation terms can be determined analytically depends on both the form of the activation function $g(x)$ and loss function $\varepsilon(s,t)$.

In the subsection "Linear stability analysis", we required $A_2 \geq 0$, $A_3 - A_1 \geq 0$, and $A_2 = A_3 - A_1 = 0$ at the global minimum $(l,R) = (1,1)$. These properties are still true, provided that the activation function g is monotonous and the loss function $\varepsilon(s,t)$ with respect to s is single-peaked at the minimum $s = t$, for arbitrary t. In particular, the softmax cross entropy loss

$$\varepsilon(s,t) = H(\mathrm{Ber}(t), \mathrm{Ber}(s)) = -t \log s - (1-t)\log(1-s),$$

where Ber: Bernouli distribution, H: cross entropy, with the sigmoidal activation $g(x) = 1/(1+e^{-x})$ satisfies the conditions above. Hence, our discussion suggests that WN with softmax cross entropy loss also works well.

Therefore, our theoretical argument can be extended to a wide range of loss functions, and WN seems to be advantageous in more general situation.

1) Y. LeCun, Y. Bengio, and G. Hinton, Nature **521**, 436 (2015).
2) J. Duchi, E. Hazan, and Y. Singer, J. Mach. Learn. Res. **12**, 2121 (2011).
3) M. D. Zeiler, arXiv:1212.5701.
4) D. Kingma and J. Ba, arXiv:1412.6980.
5) T. Tieleman and G. Hinton, COURSERA: Neural Networks Mach. Learn. **4** (2012).
6) S. Amari, Neural Comput. **10**, 251 (1998).
7) S. Ioffe and C. Szegedy, arXiv:1502.03167.
8) T. Salimans and D. P. Kingma, Advances in Neural Information Processing Systems 29, 2016, p. 901.
9) J. L. Ba, J. R. Kiros, and G. E. Hinton, arXiv:1607.06450.
10) M. Biehl and H. Schwarze, J. Phys. A **28**, 643 (1995).
11) D. Saad and S. A. Solla, Phys. Rev. E **52**, 4225 (1995).
12) P. Riegler and M. Biehl, J. Phys. A **28**, L507 (1995).
13) H. Park, M. Inoue, and M. Okada, Prog. Theor. Phys. Suppl. **157**, 275 (2005).
14) K. Hara, D. Saito, and H. Shouno, Int. Joint Conf. Neural Networks (IJCNN), 2015, p. 1.
15) M. G. Kendall, A. Stuart, and J. K. Ord, *Kendall's Advanced Theory of Statistics* (Arnold, London, 1994) Vol. 1.