Data-Dependence of Plateau Phenomenon in Learning with Neural Network — Statistical Mechanical Analysis

Yuki Yoshida  Masato Okada
Department of Complexity Science and Engineering, Graduate School of Frontier Sciences,
The University of Tokyo
5-1-5 Kashiwanoha, Kashiwa, Chiba 277-8561, Japan
{yoshida@mns, okada@edu}.k.u-tokyo.ac.jp

Abstract
The plateau phenomenon, wherein the loss value stops decreasing during the process of learning, has been reported by various researchers. The phenomenon is actively inspected in the 1990s and found to be due to the fundamental hierarchical structure of neural network models. Then the phenomenon has been thought as inevitable. However, the phenomenon seldom occurs in the context of recent deep learning. There is a gap between theory and reality. In this paper, using statistical mechanical formulation, we clarified the relationship between the plateau phenomenon and the statistical property of the data learned. It is shown that the data whose covariance has small and dispersed eigenvalues tend to make the plateau phenomenon inconspicuous.

1 Introduction
1.1 Plateau Phenomenon
Deep learning, and neural network as its essential component, has come to be applied to various fields. However, these still remain unclear in various points theoretically. The plateau phenomenon is one of them. In the learning process of neural networks, their weight parameters are updated iteratively so that the loss decreases. However, in some settings the loss does not decrease simply, but its decreasing speed slows down significantly partway through learning, and then it speeds up again after a long period of time. This is called as “plateau phenomenon”. Since 1990s, this phenomena have been reported to occur in various practical learning situations (see Figure 1(a) and Park et al. [2000], Fukumizu and Amari [2000]). As a fundamental cause of this phenomenon, it has been pointed out by a number of researchers that the intrinsic symmetry of neural network models brings singularity to the metric in the parameter space which then gives rise to special attractors whose regions of attraction have nonzero measure, called as Milnor attractor (defined by Milnor [1985]; see also Figure 5 in Fukumizu and Amari [2000] for a schematic diagram of the attractor).

1.2 Who moved the plateau phenomenon?
However, the plateau phenomenon seldom occurs in recent practical use of neural networks (see Figure 1(b) for example).

In this research, we rethink the plateau phenomenon, and discuss which situations are likely to cause the phenomenon. First we introduce the student-teacher model of two-layered networks as an ideal system. Next, we reduce the learning dynamics of the student-teacher model to a small-dimensional order parameter system by using statistical mechanical formulation, under the assumption that the
input dimension is sufficiently large. Through analyzing the order parameter system, we can discuss how the macroscopic learning dynamics depends on the statistics of input data. Our main contribution is the following:

- Under the statistical mechanical formulation of learning in the two-layered perceptron, we showed that macroscopic equations can be derived even when the statistical properties of the input are generalized. In other words, we extended the result of Saad and Solla [1995] and Riegler and Biehl [1995].

- By analyzing the macroscopic system we derived, we showed that the dynamics of learning depends only on the eigenvalue distribution of the covariance matrix of the input data, provided that the learning rate is sufficiently small.

- We clarified the relationship between the input data statistics and plateau phenomenon. In particular, it is shown that the data whose covariance matrix has small and dispersed eigenvalues tend to make the phenomenon inconspicuous, by numerically analyzing the macroscopic system.

1.3 Related works

The statistical mechanical approach used in this research is firstly developed by Saad and Solla [1995]. The method reduces high-dimensional learning dynamics of nonlinear neural networks to low-dimensional system of order parameters. They derived the macroscopic behavior of learning dynamics in two-layered soft-committee machine and by analyzing it they point out the existence of plateau phenomenon. Nowadays the statistical mechanical method is applied to analyze recent techniques (Hara et al. [2016], Yoshida et al. [2017], Takagi et al. [2019] and Straat and Biehl [2019]), and generalization performance in over-parameterized setting (Goldt et al. [2019]) and environment with conceptual drift (Straat et al. [2018]). However, it is unknown that how the property of input dataset itself can affect the learning dynamics, including plateaus.

Plateau phenomenon and singularity in loss landscape as its main cause have been studied by Fukumizu and Amari [2000], Wei et al. [2008], Cousseau et al. [2008] and Guo et al. [2018]. On the other hand, recent several works suggest that plateau and singularity can be mitigated in some settings. Orhan and Pitkow [2017] shows that skip connections eliminate the singularity. Another work by Yoshida et al. [2019] points out that output dimensionality affects the plateau phenomenon, in that multiple output units alleviate the plateau phenomenon. However, the number of output elements does not fully determine the presence or absence of plateaus, nor does the use of skip connections.

The statistical property of data just can affect the learning dynamics dramatically; for example, see Figure 2 for learning curves with using different datasets and same network architecture. We focus on what kind of statistical property of the data brings plateau phenomenon.

![Figure 1: (a) Training loss curves when two-layer perceptron with 4-4-3 units and ReLU activation learns IRIS dataset. (b) Training loss curve when two-layer perceptron with 784-20-10 units and ReLU activation learns MNIST dataset. For both (a) and (b), results of 100 trials with random initialization are overlaid. Minibatch size of 10 and vanilla SGD (learning rate: 0.01) are used.](image-url)
We consider the situation that the network learns data generated by another network, called “teacher network”, which has fixed weights. Specifically, we consider two-layer perceptron that outputs the input to the network by \( \xi \in \mathbb{R}^N \). Then the output can be written as \( s = \sum_{i=1}^{K} w_i g(J_i; \xi) \in \mathbb{R} \), where \( g \) is an activation function.

We consider the situation that the network learns data generated by another network, called “teacher network”, which has fixed weights. Specifically, we consider two-layer perceptron that outputs \( t = \sum_{n=1}^{M} v_n g(B_n \cdot \xi) \in \mathbb{R} \) for input \( \xi \) as the teacher network. The generated data \( (\xi, t) \) is then fed to the student network stated above and learned by it in the on-line manner (see Figure 3). We assume that the input \( \xi \) is drawn from some distribution \( p(\xi) \) every time independently. We adopt vanilla stochastic gradient descent (SGD) algorithm for learning. We assume the squared loss function \( \varepsilon = \frac{1}{2}(s - t)^2 \), which is most commonly used for regression.

2.2 Statistical Mechanical Formulation

In order to capture the learning dynamics of nonlinear neural networks described in the previous subsection macroscopically, we introduce the statistical mechanical formulation in this subsection.

Let \( x_i := J_i \cdot \xi \) (\( 1 \leq i \leq K \)) and \( y_n := B_n \cdot \xi \) (\( 1 \leq n \leq M \)). Then

\[
(x_1, \ldots, x_K, y_1, \ldots, y_M) \sim N(0, [J_1, \ldots, J_K, B_1, \ldots, B_M]^T \Sigma [J_1, \ldots, J_K, B_1, \ldots, B_M])
\]

holds with \( N \to \infty \) by generalized central limit theorem, provided that the input distribution \( p(\xi) \) has zero mean and finite covariance matrix \( \Sigma \).

Next, let us introduce order parameters as following: \( Q_{ij} := J_i^T \Sigma J_j = \langle x_i x_j \rangle \), \( R_{in} := J_i^T \Sigma B_n = \langle x_i y_n \rangle \), \( T_{nm} := B_n^T \Sigma B_m = \langle y_n y_m \rangle \) and \( D_{ij} := w_i w_j \), \( E_{in} := w_i v_n \), \( F_{nm} := v_n v_m \). Then

\[
(x_1, \ldots, x_K, y_1, \ldots, y_M) \sim N(0, \begin{pmatrix} Q & R^T \\ R & T \end{pmatrix}).
\]

The parameters \( Q_{ij}, R_{in}, T_{nm}, D_{ij}, E_{in}, \) and \( F_{nm} \) introduced above capture the state of the system macroscopically; therefore they are called as “order parameters.” The first three represent the state of the first layers of the two networks (student and teacher), and the latter three represent their second layers’ state. \( Q \) describes the statistics of the student’s first layer and \( T \) represents that of the teacher’s first layer. \( R \) is related to similarity between the student and teacher’s first layer. \( D, E, F \) is the second layers’ counterpart of \( Q, R, T \). The values of \( Q_{ij}, R_{in}, D_{ij}, \) and \( E_{in} \) change during learning; their dynamics are what to be determined, from the dynamics of microscopic variables, i.e. connection weights. In contrast, \( T_{nm} \) and \( F_{nm} \) are constant during learning.
At each iteration of on-line learning, weights of the student network $J_i$ and $w_i$ are updated with

$$
\Delta J_i = -\frac{\eta}{N} \frac{d\varepsilon}{dJ_i} = \frac{\eta}{N} [(t - s) \cdot w_i] g'(x_i) \xi = \frac{\eta}{N} \left[ \sum_{n=1}^{M} v_n g(y_n) - \sum_{j=1}^{K} w_j g(x_j) \right] \cdot w_i \]

$$

$$
\Delta w_i = -\frac{\eta}{N} \frac{d\varepsilon}{dw_i} = \frac{\eta}{N} g(x_i)(t - s) = \frac{\eta}{N} g(x_i) \left( \sum_{n=1}^{M} v_n g(y_n) - \sum_{j=1}^{K} w_j g(x_j) \right),
$$

(1)
in which we set the learning rate as $\eta/N$, so that our macroscopic system is $N$-independent.

Then, the order parameters $Q^{(e)}_{ij}$ and $R^{(e)}_{in}$ ($e = 0, 1, 2, \ldots$) are updated with

$$
\Delta Q^{(e)}_{ij} = \left( J_i + \Delta J_i \right)^T \Sigma^e \left( J_i + \Delta J_i \right) - J_i^T \Sigma^e J_i = J_i^T \Sigma^e \Delta J_i + J_i^T \Sigma^e \Delta J_i + \Delta J_i^T \Sigma^e \Delta J_i
$$

$$
= \frac{\eta}{N} \sum_{p=1}^{M} \sum_{q=1}^{K} E_{ip} D_{qp} g'(x_i) x^T_j g(x_p) - \sum_{p=1}^{K} D_{ip} g'(x_i) x^T_j g(x_p)

+ \frac{\eta^2}{N^2} \frac{d\varepsilon}{d\xi} \left[ \sum_{p=1}^{K} \sum_{q=1}^{M} D_{ip} D_{pq} g'(x_i) x^T_j g(x_p) g(x_q) \right]

- \frac{\eta^2}{N^2} \frac{d\varepsilon}{d\xi} \sum_{p=1}^{K} \sum_{q=1}^{M} E_{ip} E_{pq} g'(x_i) x^T_j g(x_p) g(y_q),
$$

$$
\Delta R^{(e)}_{in} = \left( J_i + \Delta J_i \right)^T \Sigma^e B_n - J_i^T \Sigma^e B_n = \Delta J_i^T \Sigma^e B_n
$$

$$
= \frac{\eta}{N} \sum_{p=1}^{M} \sum_{q=1}^{K} E_{ip} g'(x_i) y^T_n g(x_p) - \sum_{p=1}^{K} D_{ip} g'(x_i) y^T_n g(x_p),
$$

(2)
in which the higher-order terms $\Sigma^e$ ($e = 1, 2, \ldots$) appear inevitably in the learning dynamics of order parameters.
we have determined the dynamics of order parameters as (3), (5) and (7). However they have 
example,

\[
\xi^T \Sigma \xi \approx N \mu_{e+1} \quad \text{where} \quad \mu_d := \frac{1}{N} \sum_{i=1}^{N} \lambda_i^d, \quad \lambda_1, \ldots, \lambda_N : \text{eigenvalues of } \Sigma
\]

and the right hand sides of the difference equations are \(O(N^{-1})\), we can replace these difference equations with differential ones with \(N \to \infty\), by taking the expectation over all input vectors \(\xi\):

\[
\frac{dQ_{ij}^{(e)}}{d\alpha} = \eta \left[ \sum_{p=1}^{2} E_{ip} I_3(x_i, x_j^{(e)}, y_p) - \sum_{p=1}^{2} D_{ip} I_3(x_i, x_j^{(e)}, y_p) \right]
+ \eta^2 \mu_{e+1} \left[ \sum_{p,q}^{2,2} D_{ip} D_{jq} I_4(x_i, x_j, x_p, y_q) + \sum_{p,q}^{2,2} E_{ip} E_{jq} I_4(x_i, x_j, y_p, y_q) \right] - \sum_{p=1}^{2} D_{ip} E_{jq} I_3(x_i, x_j, y_p, y_q) - \sum_{p=1}^{2} E_{ip} D_{jq} I_3(x_i, x_j, y_p, y_q)
\]

\[
\frac{dR_{ip}^{(e)}}{d\alpha} = \eta \left[ \sum_{p=1}^{2} E_{ip} I_3(x_i, y_p^{(e)}, y_p) - \sum_{p=1}^{2} D_{ip} I_3(x_i, y_p^{(e)}, y_p) \right]
\]

where \(I_3(z_1, z_2, z_3) := \langle g'(z_1) z_2 g(z_3) \rangle \) and \(I_4(z_1, z_2, z_3, z_4) := \langle g'(z_1) g'(z_2) g(z_3) g(z_4) \rangle \).

In these equations, \(\tilde{\alpha} := \alpha/N\) represents time (normalized number of steps), and the brackets \(\langle \cdot \rangle\) represent the expectation when the input \(\xi\) follows the input distribution \(p(\xi)\).

The differential equations for \(D\) and \(E\) are obtained in a similar way:

\[
\frac{dD_{ij}}{d\alpha} = \eta \left[ \sum_{p=1}^{M} E_{ip} I_2(x_i, y_p) - \sum_{p=1}^{K} D_{ip} I_2(x_i, y_p) \right] + \sum_{p=1}^{M} E_{ip} I_2(x_i, y_p) - \sum_{p=1}^{K} D_{ip} I_2(x_i, y_p)
\]

\[
\frac{dE_{ip}}{d\alpha} = \eta \left[ \sum_{p=1}^{M} F_{pn} I_2(x_i, y_p) - \sum_{p=1}^{K} E_{pn} I_2(x_i, y_p) \right]
\]

where \(I_2(z_1, z_2) := \langle g(z_1) g(z_2) \rangle\).

These differential equations (3) and (5) govern the macroscopic dynamics of learning. In addition, the generalization loss \(\varepsilon_g\), the expectation of loss value \(\varepsilon(\xi) = \frac{1}{2} \| s - t \|^2 \) over all input vectors \(\xi\), is represented as

\[
\varepsilon_g = \frac{1}{2} \| s - t \|^2 = \frac{1}{2} \left[ \sum_{p,q}^{M,M} F_{pq} I_2(y_p, y_q) + \sum_{p,q}^{K,K} D_{pq} I_2(x_p, x_q) - 2 \sum_{p,q}^{K,M} E_{pq} I_2(x_p, y_q) \right].
\]

### 3.1 Expectation terms

Above we have determined the dynamics of order parameters as (3), (5) and (7). However they have expectation terms \(I_2(z_1, z_2), I_3(z_1, z_2^{(e)}, z_3)\) and \(I_4(z_1, z_2, z_3, z_4)\), where \(z\)s are either \(x_i\) or \(y_n\). By studying what distribution \(\pi\) follows, we can show that these expectation terms are dependent only on 1-st and \((e + 1)\)-th order parameters, namely, \(Q^{(1)}, R^{(1)}, T^{(1)}\) and \(Q^{(e+1)}, R^{(e+1)}, T^{(e+1)}\); for example,

\[
I_3(x_i, x_j^{(e)}, y_p) = \int dz_1 dz_2 dz_3 g'(z_1) z_2 g(z_3) N(z|0, \begin{pmatrix} Q_{ij}^{(1)} & Q_{ij}^{(e+1)} & P_{ip}^{(1)} \\ Q_{ij}^{(e+1)} & R_{ij}^{(e+1)} & P_{ip}^{(e+1)} \\ P_{ip}^{(1)} & P_{ip}^{(e+1)} & T_{ip}^{(1)} \end{pmatrix})
\]
Yoshida et al. [2017].

We assume that the second layers’ weights of the student and the teacher, namely \( w \) and \( i \), can be updated with SGD iteratively, and (ii) by solving numerically the differential equations \( \Omega^2 \) which govern the order parameters, and we confirmed that they accord with each other very well (Figure 4). Note that we set the initial values of order parameters in (ii) as values corresponding to initial weights used in (i). For dependence of the learning trajectory on the initial condition, see Supplementary Material A.3.

### 3.2 Dependency on input data covariance \( \Sigma \)

Surprisingly, the differential equation system we derived depends on \( \Sigma \), through only the coefficient \( \mu_{e1} \) of \( O(\eta^2) \)-term. That is, the system only depends on the eigenvalue distribution of \( \Sigma \). If the learning rate \( \eta \) is sufficiently small and \( O(\eta^2) \)-term can be negligible, the system becomes not to depend on \( \Sigma \) at all. However, note that \( \Sigma \) affects the initial state of the system (i.e. initial values of the order parameters and values of order constants).

### 3.3 Evaluation of expectation terms for specific activation functions

Expectation terms \( I_2, I_3 \) and \( I_4 \) can be analytically determined for some activation functions \( g \), including sigmoid-like \( g(x) = \text{erf}(x/\sqrt{2}) \) (see Saad and Solla [1995]) and \( g(x) = \text{ReLU}(x) \) (see Yoshida et al. [2017]).

### 4 Analysis of numerical solutions of macroscopic differential equations

In this section, we analyze numerically the order parameter system, derived in the previous section. We assume that the second layers’ weights of the student and the teacher, namely \( w_t \) and \( v_t \), are fixed to 1, for simplicity (i.e. we consider the learning of a committee machine). Here we think of sigmoid-like activation \( g(x) = \text{erf}(x/\sqrt{2}) \).

### 4.1 Consistency between macroscopic system and microscopic system

First of all, we confirmed the consistency between the macroscopic system we derived and the original microscopic system. That is, we computed the dynamics of the generalization loss \( \varepsilon_g \) in two ways: (i) by updating weights of the network with SGD \( \Omega^1 \) iteratively, and (ii) by solving numerically the differential equations \( \Omega^2 \) which govern the order parameters, and we confirmed that they accord with each other very well (Figure 4). Note that we set the initial values of order parameters in (ii) as values corresponding to initial weights used in (i). For dependence of the learning trajectory on the initial condition, see Supplementary Material A.3.

---

\(^1\) We executed all computations on a standard PC.
4.2 Case of scalar input covariance $\Sigma = \sigma I_N$

As the simplest case, here we consider the case that the covariance matrix $\Sigma$ is proportional to unit matrix. In this case, $\Sigma$ has only one eigenvalue $\lambda = \mu_1$ of multiplicity $N$, then our order parameter system contains only parameters whose order is 0 $(\epsilon = 0)$. For various values of $\mu_1$, we solved numerically the differential equations of order parameters (5) and plotted the time evolution of generalization loss $\varepsilon_g$ (Figure 5(a)). From these plots, we quantified the lengths and heights of the plateaus as following: we regarded the system is plateauing if the decreasing speed of log-loss is smaller than half of its terminal converging speed, and we defined the height of the plateau as the median of loss values during plateauing. Quantified lengths and heights are plotted in Figure 5(b)(c). It indicates that the plateau length and height heavily depend on $\mu_1$, the input scale. Specifically, as $\mu_1$ decreases, the plateau rapidly becomes longer and lower. Though smaller input data lead to longer plateaus, it also becomes lower and then inconspicuous. This tendency is consistent with Figure 2(a)(b), since IRIS dataset has large $\mu_1$ ($\approx 15.9$) and MNIST has small $\mu_1$ ($\approx 0.112$). Considering this, the claim that the plateau phenomenon does not occur in learning of MNIST is controversy; this suggests the possibility that we are observing quite long and low plateaus.

Note that Figure 5(b) shows that the speed of growing of plateau length is larger than $O(1/\mu_1)$. This is contrast to the case of linear networks which have no activation; in that case, as $\mu_1$ decreases the speed of learning gets exactly $1/\mu_1$-times larger. In other words, this phenomenon is peculiar to nonlinear networks.

Figure 4: Example dynamics of generalization error $\varepsilon_g$ computed with (a) microscopic and (b) macroscopic system. Network size: $N$-2-1. Learning rate: $\eta = 0.1$. Eigenvalues of $\Sigma$: $\lambda_1 = 0.4$ with multiplicity $0.5N$, $\lambda_2 = 1.2$ with multiplicity $0.3N$, and $\lambda_3 = 1.6$ with multiplicity $0.2N$. Black lines: dynamics of $\varepsilon_g$. Blue lines: $Q_{11}, Q_{12}, Q_{22}$. Green lines: $R_{11}, R_{12}, R_{21}, R_{22}$.

Figure 5: (a) Dynamics of generalization error $\varepsilon_g$ when input variance $\Sigma$ has only one eigenvalue $\lambda = \mu_1$ of multiplicity $N$. Plots with various values of $\mu_1$ are shown. (b) Plateau length and (b) plateau height, quantified from (a).
4.3 Case of different input covariance $\Sigma$ with fixed $\mu_1$

In the previous subsection we inspected the dependence of the learning dynamics on the first moment $\mu_1$ of the eigenvalues of the covariance matrix $\Sigma$. In this subsection, we explored the dependence of the dynamics on the higher moments of eigenvalues, under fixed first moment $\mu_1$.

In this subsection, we consider the case in which the input covariance matrix $\Sigma$ has two distinct nonzero eigenvalues, $\lambda_1 = \mu_1 - \Delta \lambda/2$ and $\lambda_2 = \mu_1 + \Delta \lambda/2$, of the same multiplicity $N/2$ (Figure 6). With changing the control parameter $\Delta \lambda$, we can get eigenvalue distributions with various values of second moment $\mu_2 = \langle \lambda_1^2 \rangle$.

\[
\begin{array}{c}
\Delta \lambda \\
\hline
\mu_1 - \frac{\Delta \lambda}{2} & \mu_1 + \frac{\Delta \lambda}{2} \\
\end{array}
\]

Figure 6: Eigenvalue distribution with fixed $\mu_1$ parameterized by $\Delta \lambda$, which yields various $\mu_2$.

Figure 7(a) shows learning curves with various $\mu_2$ while fixing $\mu_1$ to 1. From these curves, we quantified the lengths and heights of the plateaus, and plotted them in Figure 7(b)(c). These indicate that the length of the plateau shortens as $\mu_2$ becomes large. That is, the more the distribution of nonzero eigenvalues gets broaden, the more the plateau gets alleviated.

![Figure 7](image)

Figure 7: (a) Dynamics of generalization error $\varepsilon_g$ when input variance $\Sigma$ has two eigenvalues $\lambda_{1,2} = \mu_1 \pm \Delta \lambda/2$ of multiplicity $N/2$. Plots with various values of $\mu_2$ are shown. (b) Plateau length and (c) plateau height, quantified from (a).

5 Conclusion

Under the statistical mechanical formulation of learning in the two-layered perceptron, we showed that macroscopic equations can be derived even when the statistical properties of the input are generalized. We showed that the dynamics of learning depends only on the eigenvalue distribution of the covariance matrix of the input data, in case that the learning rate is sufficiently small. By numerically analyzing the macroscopic system, it is shown that the statistics of input data dramatically affect the plateau phenomenon.

Through this work, we explored the gap between theory and reality; though the plateau phenomenon is theoretically predicted to occur by the general symmetrical structure of neural networks, it is seldom observed in practice. However, more extensive researches are needed to fully understand the theory underlying the plateau phenomenon in practical cases.
Acknowledgement

This work was supported by JSPS KAKENHI Grant-in-Aid for Scientific Research(A) (No. 18H04106).

References


