Optimal Initialization Scale for Neural Networks with Locally Quadratic Loss Landscapes: An SGD Dynamics Perspective

Hiroshi Horii[1,2] and Sothea Has[3]

[1] *Laboratoire Jacques-Louis Lions, Sorbonne Université, Université de Paris et CNRS, Paris, France*
[2] *Université Côte d'Azur, CNRS, Observatoire de la Côte d'Azur, Laboratoire Lagrange, Nice, France*
[3] *Research and Innovation Center, Institute of Technology of Cambodia, Russian Federation Blvd., P.O. Box 86, Phnom Penh, Cambodia*

***Abstract*—Stochastic gradient descent (SGD), one of the most fundamental optimization algorithms in machine learning (ML), can be recast through a continuous-time approximation as a Fokker-Planck equation for Langevin dynamics, a viewpoint that has motivated many theoretical studies. Within this framework, we study the relationship between the quasi-stationary distribution derived from this equation and the initial distribution through the Kullback-Leibler (KL) divergence. As the quasi-steady-state distribution depends on the expected cost function, the KL divergence eventually reveals the connection between the expected cost function and the initialization distribution. Under the assumptions of a locally quadratic neural-network loss landscape and a quasi-stationary SGD regime, we derive an explicit upper bound for the expected loss in terms of the initialization variance. Then, by minimizing this bound, we obtain an optimal condition of the initialization variance in the Gaussian case. This result provides a concrete mathematical criterion, rather than a heuristic approach, to select the scale of weight initialization within the locally quadratic SGD regime. In addition, we experimentally confirm our theoretical results by using classical SGD to train shallow fully connected neural networks on the MNIST dataset and a two-layer CNN model on the CIFAR-10 dataset. The results show that, for the simple network model, if the variance of the initialization distribution satisfies our theoretical optimal condition, then the corresponding network achieves lower final training loss and higher test accuracy than the conventional He-normal initialization. For the two-layer CNN model, He-normal initialization gives the best performance among the tested initializations. This observation is consistent with the proposed optimal condition, since the variance ratio of He-normal initialization remains close to the optimal value not only at the overall level but also approximately at the layer-wise level. This suggests that satisfying the proposed condition at both global and layer-wise scales may be important for achieving lower loss in this architecture. We also discuss the limitation of our optimal condition on complex networks. Our work thus supplies a mathematically grounded indicator for choosing the initialization variance and clarifies its physical meaning in terms of the parameter dynamics in neural network models.**

I. Introduction

Stochastic Gradient Descent (SGD) is a cornerstone optimization algorithm that underpins many advancements in machine learning (ML) and deep learning (DL) [1]–[3]. In general, if L is a differentiable loss function of any ML model with trainable parameters $W \in \mathbb{R}^K$ (all weights and biases of neural networks, for example), then training such a model is equivalent to searching for the optimal parameter $W^* \in \mathbb{R}^K$ that minimizes L, that is,

$$L(W^*) := \inf_{W \in \mathbb{R}^K} L(W). \tag{1}$$

SGD approximates such W^* by generating a sequence of parameter $(W^{(t)})_{t \geq 0}$ such that $W^{(t)} \approx W^*$ for sufficiently large t. The main recurrence relation in SGD is given by

$$W^{(t+1)} := W^{(t)} - \alpha \nabla \hat{L}(W^{(t)}), \tag{2}$$

where α is the *learning rate*, $\hat{L}$ is the estimated loss function computed using a small subset of training data called *minibatch*, and $\nabla \hat{L}$ is the gradient of $\hat{L}$ with respect to the weight W. In practice, the initialization $W^{(0)} \sim p_0$ for some probability distribution p_0 (the common choice is Gaussian distribution $\mathcal{N}(0, \sigma_0^2 I)$ for some $\sigma_0 > 0$). At each step, the algorithm tries to move toward the direction with the steepest loss function at a constant rate of $\alpha > 0$.

Despite its widespread applications, the theoretical understanding of its dynamics and behavior remains an active area of research. For instance, several theoretical studies have been introduced to explore the convergence property of SGD [4], the convergence around flat minima [5]–[7], and heavy-tailed phenomena in SGD [8]–[11]. Other studies focus on the influence of hyperparameters, including the learning rate α and the batch size b, on the behavior of SGD [6], [12]. In addition, the choice of weight-initialization strategy also has a significant impact on both convergence behavior and generalization performance of the model as well (see, for example, [13], [14]). In particular, the widely used Xavier initialization [15] and He-normal initialization [16] have proven effective in practice for efficient learning in deep neural networks. Additionally, [17] briefly discusses how initialization choices are related to network architecture.

This study focuses on analyzing the impact of the initialization parameter using the continuous-time approximation method of SGD. First, we approximate the stochastic sequence generated by equation (2) using a continuous-time framework. This approximation is well established to study the relation between minibatch size and the learning rate in Artificial Neural Networks (ANNs), for example, by [12], [18]–[20]. Next, by applying the Fokker-Planck equation, we derived the steady-state distribution of the converged process under standard conditions of the loss function L. Several recent studies have cast SGD dynamics in terms of a Fokker–Planck formalism (e.g. [21]–[24]). In particular, [23] suggested that the stationary state of SGD can retain dependence on the initialization through a memory effect. Moreover, [22] connected the analytically derived steady state of the associated diffusion process with the empirically observed quasi–steady state by explicitly quantifying their KL divergence. These perspectives are closely related to the methodology developed in our study. However, based on the continuous-time approximation, we establish the main additional contributions, as listed below.

- We derive, through a Kullback-Leibler divergence analysis, the analytical bound for the expected loss function with Gaussian weight initialization in the classical SGD.
- We show that for small initialization variances ($\sigma_0^2 \ll 1$), the expected loss scales as $\mathbb{E}[L(W)] = O(\mathbb{V}(W))$, where $\mathbb{V}(W)$ is the variance of the parameter W at quasi-steady state. Meanwhile, for large variances ($\sigma_0^2 \gg 1$), it behaves like $O(\log \sigma_0^2)$.
- By minimizing the upper bound derived from the expected loss function with respect to σ_0^2, we obtain an optimal condition for the initialization variance σ_0^2.
- We validate our theoretical result through numerical experiments by comparing it against the widely used He-normal initialization. It shows the existence of an optimal hyperparameter in three-layer deep neural network models (DNNs), and a new perspective that confirms our optimal condition can also be observed in a two-layer CNN model initialized using the He-normal distribution.
- Finally, our numerical experiments seem to suggest that better performance is obtained when the optimal condition is satisfied not only for overall layer weights but also for the parameters at each individual layer.

This paper is organized as follows. Section II provides the background on our work, focusing on the continuous-time approximation of the SGD sequence and deriving its steady-state distribution. The main result of this study is presented in Section III. Numerical experiments carried out on the MNIST and CIFAR-10 data sets are provided in Section IV. Finally, Section V concludes the result of this study and discusses possible directions for future investigations.

II. Stochastic differential equation framework for SGD

A. Continuous-time approximation for SGD

In this section, we provide the main theoretical framework for understanding the optimization problems of the SGD in ML. The SGD is one of the most widely adopted and basic optimization algorithms. In addition, the SGD is denoted as a discretized recurrence equation for an optimized parameter in ML. Its time evolution is driven by the loss function gradient, and the step size is defined by the learning rate $\alpha > 0$. In the following, let N denote the total size of the training data. Under the condition that the ratio α/N is sufficiently small, we can obtain a stochastic differential equation (SDE) by rewriting (2) as follows.

$$\begin{aligned} W^{(t+1)} - W^{(t)} &= -\frac{\alpha}{N}\left[\nabla L + (N\nabla\hat{L} - \nabla L)\right] \\ &= -\frac{\alpha}{N}\left[\nabla L + (\nabla\tilde{L} - \nabla L)\right], \end{aligned} \tag{3}$$

where $\nabla L = \sum_i^N \nabla L_i$ is the sum of the true gradient ∇L_i evaluated at observation i, and $\nabla\tilde{L} = \frac{N}{b}\sum_{i\in\Omega_b}\nabla L_i$ is an estimated gradient evaluated by multiplying the average individual gradients (over a minibatch) with the sample size N. Intuitively,

$\nabla\tilde{L}$ approaches ∇L as b approaches N. Then, the equation is rewritten as follows.

$$W^{(t+1)} - W^{(t)} = -\frac{\alpha}{N}\left[\nabla L - \eta(t)\right]. \tag{4}$$

Here, $\eta(t) = \nabla L - \nabla\tilde{L}$ is the difference between the true gradient and the sampled gradient at $W^{(t)}$. It reflects the error introduced when approximating the true gradient of the loss function using a minibatch gradient. If the batch size is sufficiently large, the central limit theorem can be assumed to hold for this gradient error. Consequently, the expected value of η is $\mathbb{E}[\eta] = 0$, and the gradient error can be seen as Gaussian noise for the recurrence equation. Under this condition, the noise is characterized by $\mathbb{E}[\eta(t)\eta(t')] = N(\frac{N}{b} - 1)\Sigma\delta(t - t') \approx \frac{N^2}{b}\Sigma\delta(t - t')$, where Σ is a covariance matrix [12] when N is sufficiently large compared to b. We assume that this covariance matrix has a square root decomposition formula, which is given by $\Sigma = BB^T$. Furthermore, if α is sufficiently small and comparable to the size of the discretization time step Δt ($\Delta t := \alpha$), then the continuous-time approximation can be applied. In the regime where α is sufficiently small, this equation can be interpreted as a stochastic differential equation (SDE), expressed as follows:

$$dW^{(t)} = -\nabla L dt + \sqrt{\epsilon} B dV_t. \tag{5}$$

where ϵ is $\frac{\alpha}{b}$. In addition, $V(t)$ follows $V(t) \sim \mathcal{N}(0, 1)$. Note that the coefficient of the stochastic term $\epsilon^{\frac{1}{2}}$ is obtained by considering $\lim_{\Delta t\to 0}\sqrt{\Delta t}\eta = dV_t$. This equation can also be viewed as an equation for the Wiener process.

B. Fokker-Planck equation

Under the small regime of ϵ, the SGD can be seen as an SDE with the Wiener process. Thus, by using Ito calculus, we can derive the Fokker-Planck equation for the SGD framework. The Fokker-Planck equation allows us to describe the evolution of the probability distribution function over time, which is useful for understanding the distributional dynamics of weights in SGD.

Suppose we have a test function $Y^{(t)} = \phi(W^{(t)}) = (Y_k)$, Ito calculus for the kth stochastic process component Y_k, is calculated as

$$dY_k^{(t)} = \sum_i \frac{\partial\phi_k}{\partial w_i} dW_i^{(t)} + \frac{1}{2}\sum_i\sum_j \frac{\partial^2\phi_k}{\partial w_i w_j} dW_i^{(t)} dW_j^{(t)}. \tag{6}$$

Finally, we get

$$\begin{aligned} d\phi_k &= \sum_i \frac{\partial\phi_k}{\partial w_i}\left(-\nabla L dt + \sqrt{\epsilon} B dV_t\right)_i \\ &\quad + \frac{1}{2}\sum_i\sum_j \frac{\partial^2\phi_k}{\partial w_i^2}(\epsilon\Sigma)\delta_{ij} dt \\ \frac{d\phi_k}{dt} &= \sum_i \frac{\partial\phi_k}{\partial w_i}\left(-\nabla L + \sqrt{\epsilon} B\frac{dV_t}{dt}\right)_i \\ &\quad + \frac{1}{2}\sum_i\sum_j \frac{\partial^2\phi_k}{\partial w_i^2}(\epsilon\Sigma)\delta_{ij}. \end{aligned} \tag{7}$$

Here, we use $dV_i dV_j = \delta_{ij} dt$, where δ_{ij} is Kronecker delta. To obtain the Fokker-Planck equation (Kolmogorov forward equation), we take the expectation of (7):

$$\begin{aligned} \mathbb{E}\left[\frac{d\phi_k}{dt}\right] = \mathbb{E}\Bigg[&\sum_i \frac{\partial\phi_k}{\partial w_i}\left(-\nabla L + \sqrt{\varepsilon}\, B\,\frac{dV_t}{dt}\right)_i \\ &+ \frac{1}{2}\sum_{i,j}\frac{\partial^2\phi_k}{\partial w_i\,\partial w_j}\,\varepsilon\,\Sigma\,\delta_{ij}\Bigg]. \end{aligned} \tag{8}$$

Then, we obtain

$$\begin{aligned} \frac{\partial p(W,t)}{\partial t} &= \sum_i \frac{\partial}{\partial w_i}\left(p(W,t)\,\nabla L\right)_i \\ &\quad + \frac{1}{2}\sum_{i,j}\frac{\partial^2}{\partial w_i\,\partial w_j}\left(p(W,t)\,\varepsilon\,\Sigma\right)\delta_{ij}\,. \end{aligned} \tag{9}$$

Equation (9) encodes the dynamics of the weight distribution function and is useful for understanding its behavior. By using a different stochastic process, such as "Levy-process", we can also derive a similar SDE [25]. Note that we used Gaussian noise to derive the Fokker-Planck equation, and this noise comes from the gradient error between the true gradient and the gradient on the minibatch learning.

Our theory is developed around the quasi-convergence of SGD. Therefore, a usual assumption for the existence of such a state is required and stated in the following part. We first assume that around the deep local minimum, the loss function is well approximated by a quadratic form, which is given by

$$L(W) = \frac{1}{2}W^T A W, \tag{10}$$

where A is positive definite and $\nabla L = 0$. Without loss of generality, we assume that a minimum loss is at $W = 0$. Through this assumption, we can describe Gibbs-like quasi-steady state distribution, and it makes sense when the loss function is smooth and the stochastic process reaches a low-variance quasi-stationary distribution around a deep local minimum [22]. Moreover, within the small regime of learning rate, the exit time from the local minimum is very

long [26]. Thus, the local minimum is deep, and a quasi-steady state exists. It is given by solving the Fokker-Planck equation when we assume the distribution converges to a steady state after exploring the state space long enough, under isotropic variance, i.e., $\Sigma = \sigma^2 I$. At the quasi-steady state, the solution $p(W)$ evolves independently of time t. By applying this condition on (9), we obtain the following equation:

$$\nabla p(W) = -p(W)\frac{2}{\epsilon\sigma^2}\nabla L. \tag{11}$$

Therefore, the solution of the weight distribution function is given by

$$p(W) = C_{1,\mathrm{loc}} e^{-\frac{2bL(W)}{\alpha\sigma^2}}. \tag{12}$$

Here, we use $\frac{\alpha}{b} = \epsilon$, and $C_{1,\mathrm{loc}}$ is a normalized constant for a distribution for one local minimum. This formula contains the structure of the loss function and output softmax function through the loss function L. By collecting all the necessary conditions, the following assumptions are made throughout this study:

1) The learning rate α is sufficiently small, and SGD can be written as an SDE in continuous time approximation.
2) The covariance matrix is an isotropic structure $\Sigma = \sigma^2 I$.
3) This stochastic dynamics goes to a deep local minimum with the quadratic loss function, and the escape time is sufficiently large. Then, we have a quasi-steady state for the Fokker-Planck formalization.

From these assumptions, the main results of this study are presented in the following section.

III. Optimization method for initialization variance

In ML, to obtain a high-performance model, we need to choose optimal hyperparameter settings such as the learning rate, batch size, and the properties of the initialization distribution. Building upon the continuous-time framework developed in Section II, this section investigates how the initialization variance σ_0 affects the final loss value in SGD. Our analysis begins by establishing the steady-state distribution of the weights via the Fokker–Planck equation. We also prove the relation between the variance of the steady state and the loss values by optimizing the variance of the initial distribution. We first consider the KL divergence between the stationary distribution and the initial Gaussian density to relate the steady-state loss to the initial variance.

Proposition III.1. *Assume that the quasi-steady state distribution of Equation* (12) *exists, and let* p_0 *be the Gaussian density of the initial distribution of the parameter in the SGD recurrence equation* (4)*, i.e.,*

$$p_0(W) = \frac{1}{(2\sigma_0^2\pi)^{K/2}} e^{-\frac{\|W\|^2}{2\sigma_0^2}}, \tag{13}$$

where K *is the dimension of the parameter and* σ_0 *is the variance of the initial parameter* $W^{(0)}$*. Moreover,* $L(W)$ *is a cost function satisfying the quadratic form assumption with a deep local minimum. Then, the expected value of the loss, with respect to the steady-state distribution, satisfies the following inequality*

$$\mathbb{E}_{\mathrm{ss}}[\bar{L}(W)] \leq \frac{\alpha\,\sigma^2}{2\,b\,K}\Big[\frac{K}{2}\,\log(2\,\sigma_0^2\pi) + \log(C_{1,\mathrm{loc}}) + \frac{\mathbb{E}_{\mathrm{ss}}[\|W\|^2]}{2\,\sigma_0^2}\Big]. \tag{14}$$

where $\mathbb{E}_{ss}$ *refers to the expectation with respect to the steady-state distribution of the parameter* W *and* $\bar{L}(W)$ *is the averaged loss function by the number of elements* K*.*

Proof. We first recall the Fokker-Planck equation, the stationary state is given by

$$p_{\mathrm{ss}}(W) = C_{1,\mathrm{loc}} e^{-\frac{2bL(W)}{\alpha\sigma^2}}. \tag{15}$$

Next, we introduce the Kullback–Leibler (KL) divergence between the initial state and the stationary state. The KL divergence is given by

$$D(p_{\mathrm{ss}}\|p_0) = \int p_{\mathrm{ss}} \log\left(\frac{p_{\mathrm{ss}}}{p_0}\right) dW \geq 0. \tag{16}$$

Note that the KL divergence is always non-negative. Finally, substituting the initial distribution and the stationary state, we can obtain

$$\mathbb{E}_{\mathrm{ss}}[\bar{L}(W)] \leq \frac{\alpha\,\sigma^2}{2\,b\,K}\Big[\frac{K}{2}\,\log(2\,\sigma_0^2\pi) + \log(C_{1,\mathrm{loc}}) + \frac{\mathbb{E}_{\mathrm{ss}}[\|W\|^2]}{2\,\sigma_0^2}\Big]. \tag{17}$$

□

Remark III.2. *By following Equation* (14)*, we can consider the behavior of* $\mathbb{E}_{\mathrm{ss}}[L(W)]$ *within two different regions.*

- *When* $\sigma_0^2 \ll 1$*, the loss function behaves as*

$$\mathbb{E}_{\mathrm{ss}}[\bar{L}(W)] \leq K_1 \bar{V}(W), \tag{18}$$

with $\mathbb{E}_{\mathrm{ss}}[W] = 0$*,* K_1 *is a constant and* $\bar{V}(W)$ *is the averaged variance of the steady-state weight distribution, which is divided by* K*.*

- *On a contrary, when $\sigma_0^2 \gg 1$, the expectation of the loss function behaves as*

$$\mathbb{E}_{\rm ss}[\bar{L}(W)] \leq K_2 \log(\sigma_0^2). \tag{19}$$

In the small initial variance range, the behavior of the final state of the loss function depends on the final state of the weight variance $V(W)$. In this sense, when we set the small variances for the initialization, the value of the loss function is proportional to the size of the final state weight distribution. Meanwhile, when $\sigma_0 \gg 1$, the value of loss function proportional to $\log(\sigma_0^2)$. Although at first glance the behavior under $\sigma_0^2 \gg 1$ might seem preferable, the large value of σ_0^2 causes the bound function to increase, offsetting the advantage. In particular, when the steady-state variance $\mathbb{E}_{\rm ss}[\|W\|^2]$ remains on the same scale as σ_0^2, initializing with a smaller σ_0^2 yields a tighter loss bound than using a larger one. Empirically, in the DNN model without vanishing-gradient issues, the observed steady-state variance indeed stays proportional to the chosen σ_0^2 (Section IV). Moreover, by optimizing the r.h.s of (14) with respect to σ_0, we achieve the optimal condition for the initialization variance as stated in the following theorem.

Theorem III.3. *Under the assumptions of Proposition III.1, one has*

$$\mathbb{E}_{\rm ss}[\bar{L}(W)] \leq \frac{\alpha\sigma^2}{4b}\left[\log\left(\frac{\mathbb{E}_{\rm ss}[\|W\|^2]}{K}\right) + \log(2\pi e C_{1,\rm loc})\right], \tag{20}$$

provided that the optimal standard deviation of the initial distribution is given by

$$\sigma_0 = \sqrt{\frac{\mathbb{E}_{\rm ss}[\|W\|^2]}{K}} = \sqrt{\bar{V}(W)}.$$

If we assume $\mathbb{E}_{ss}[W] = 0$, the optimized bound for the expected value of the loss function is:

$$\mathbb{E}_{\rm ss}[L(W)] \leq \frac{\alpha\sigma^2}{4b}\left[\log(\bar{V}(W)) + \log(2\pi e C_{1,\rm loc})\right], \tag{21}$$

where $\bar{V}(W)$ is the averaged variance of the final state.

The direct consequence of our result provides that $\sigma_0 = \sqrt{\mathbb{E}_{\rm ss}[|W|^2]/K}$, where K is the total number of units of the network. For a given structure of the network, the optimal scale of the initialization standard deviation is $O(1/\sqrt{K})$. This result is also consistent with He-normal distribution where the initialization standard deviation at layer ℓ is of order $O(1/\sqrt{K_\ell})$, where K_ℓ is the size of the layer.

Remark III.4. *The inequality in Theorem III.3 depends on the normalizing constant $C_{1,\rm loc}$, which captures the information about the loss landscape and model structure. A common way to approximate $C_{1,\rm loc}$ is to apply the Laplace approximation for each local minimum and sum the resulting Gaussian normalizers, as mentioned in, for instance, [27], [28]. In our theorem, $C_{1,\rm loc}$ should be relatively smaller than $\bar{V}$. If we consider a one-layer model, the normalized constant is larger than $C_{1,\rm loc}$ because it has only a single minimum.*

Remark III.5. *According to [22], under the local quadratic approximation* (10)*, the quasi-steady-state distribution is approximated by a Gaussian*

$$\tilde{p}_{\rm ss} \propto e^{-\frac{1}{2}W^T C^{-1} W}, \tag{22}$$

where C is the quasi stationary covariance of $\tilde{p}_{\rm ss}$. Note that for a general constant anisotropic noise covariance Σ, the quasi-steady state is no longer a Gibbs-type density. With the normalization, we set the quasi-steady state to be

$$\tilde{p}_{\rm ss}(W) = \frac{1}{(2\pi)^{K/2}\det(C)^{1/2}} e^{-\frac{1}{2}W^T C^{-1} W}. \tag{23}$$

In addition, this covariance matrix satisfies the following relation

$$AC + CA^T = \epsilon\Sigma, \tag{24}$$

where A is already introduced in (10)*. Then, the KL divergence is calculated by*

$$D_{\rm KL}(\tilde{p}_{\rm ss}\|p_0) = \frac{1}{2}\left[-\log\det(C) - K + K\log\sigma_0^2 + \frac{\mathbb{E}_{\rm ss}\left[\|W\|^2\right]}{\sigma_0^2}\right]. \tag{25}$$

Moreover, we can optimize the KL divergence to obtain an optimal condition for initialization variance σ_0^2. It is given by

$$\sigma_0 = \sqrt{\frac{\mathbb{E}_{\rm ss}[\|W\|^2]}{K}}. \tag{26}$$

It is the same expression as Theorem. (III.3) *but $\mathbb{E}[|W|^2] = \mathrm{tr(C)}$ with $\mathbb{E}_{\rm ss}[W] = 0$. If C is an isotropic covariance, this result corresponds to the theorem. III.3. For the anisotropic constant covariance case, we take into account* tr(C)*. In this setting, we do not obtain an explicit bound for $\mathbb{E}_{\rm ss}[L(W)]$ as in the isotropic case. However, $L(W)$ and the KL divergence are related through the quasi-steady covariance C, and by optimizing the KL divergence, it still provides an optimal initialization condition $\sigma_0^2 = \mathrm{tr(C)/K}$.*

In summary, our analysis shows that the expected loss is highly sensitive to the scale of the initial weight distribution. In particular, for small initialization variances, the loss is (at most) proportional to $\bar{V}(W)$. In this sense, to obtain the small loss values, we need

to suppress the variances of the steady state. Ideally, if one can set the initialization variance to $\sigma_0 = \sqrt{\bar{V}(W)}$, which implies that the initial distribution is very close to the steady state, the optimized loss bound is achieved. However, satisfying this optimal condition before the learning process is very challenging in practice. In the following numerical section, we further explore the relationship between $\mathbb{E}_{\rm ss}[\bar{L}(W)]$ and $\bar{V}(W)$ and discuss the practical difficulties in meeting this optimal condition.

IV. Numerical experiments

A. Setup

In this section, we perform the simulation on the Deep Neural Network (DNN) model. Our model is a basic DNN model, a fully connected neural network comprising an input layer, hidden layers, and an output layer. The NN model takes training data of size N, (x_i, y_i) for $i = 1, 2, \ldots, N$. Throughout this article, the input $x_i = (x_{i1}, \ldots, x_{id})$ is normalized to be in $[0, 1]^d$, and the output $y_i \in \mathscr{Y} = \{1, \ldots, M\}$, where M is the number of classes of the classification task. In this case, the weight $W = (W_{jm})$ is a $d \times M$ matrix, connecting the input layer to the output layer. To perform classification, we use the softmax function to normalize output values. Moreover, for any input x_i, the transformed value is defined by $z_i = x_i W = (\sum_{j=1}^d x_{ij} W_{j1}, \ldots, \sum_{j=1}^d x_{ij} W_{jM}) \in \mathbb{R}^M$. Then, the soft-max function evaluated at z_i is $\phi(z_i) = (\phi_1(z_i), \ldots, \phi_M(z_i))$ and is defined by

$$\phi_m(z_i) = \frac{e^{z_{im}}}{\sum_k e^{z_{ik}}}, \text{ for } m = 1, \ldots, M. \tag{27}$$

In this study, the loss function is chosen as the cross-entropy. In other words, the loss function of the weight W evaluated at any input x is defined by,

$$L_{\rm cross}(W; x) = -\sum_{m=1}^{M} t_m \log \phi_m(xW), \tag{28}$$

where $t = (t_1, \ldots, t_M)$ is the one-hot coding of the truth label y, i.e.,

$$y = m_0 \Leftrightarrow t_m = \begin{cases} 0, & \text{if } m \neq m_0 \\ 1 & \text{if } m = m_0 \end{cases}. \tag{29}$$

In addition, for minibatch learning, the global loss function is given by,

$$\tilde{L}(W) = \frac{1}{b} \sum_{i \in \Omega_b} L_{\rm cross}(W; x_i), \tag{30}$$

where $\Omega_b \subset \{1, \ldots, n\}$ is a random subset of size b. Then, the SGD optimization step is written as

$$W_{jm}^{(t+1)} = W_{jm}^{(t)} - \frac{\alpha}{b} \sum_{i \in \Omega_b} \frac{\partial L_{\rm cross}(W^{(t)}; x_i)}{\partial W_{jm}}, \quad \text{for } j = 1, \ldots, d, \text{ and } m = 1, \ldots, M, \tag{31}$$

for minibatch learning. In our problem, the optimization method on an NN model with no middle layer is based on the above setup and iterative updates following the SGD optimization algorithm.

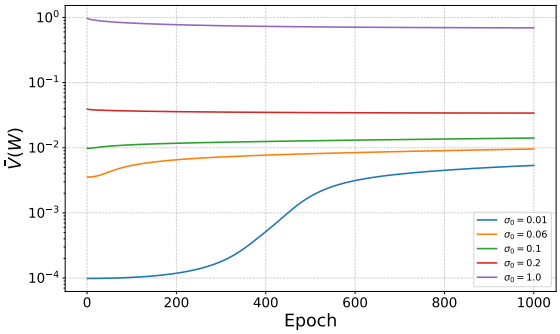
B. Numerical results

We carry out classification experiments on MNIST with a three-layer fully connected network with ReLU activations. All weights are initialized as $W_0 \sim \mathscr{N}(0, \sigma_0^2)$ and the models are trained with stochastic gradient descent (SGD). We aim to examine the relationship between the expected value of the steady-state loss and the variance of the steady-state weight distribution.

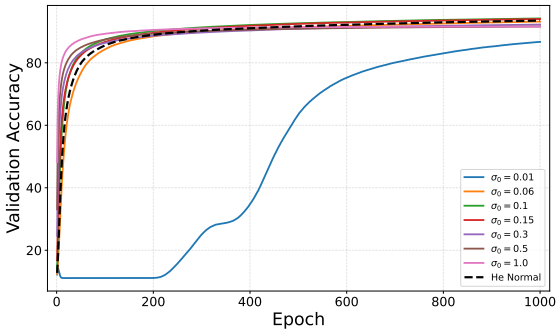
In the three-layer network, the loss function is non-convex, multiple minima coexist, and the theorem predicts a non-trivial dependence between the final loss and the variance of the initial distribution σ_0^2. In what follows, we present numerical results that primarily validate Theorem III.1 and, in addition, illustrate the time evolution of the weight variance. In our theoretical result, we set $V(W) := \mathbb{E}[\|W\|^2]$ under the symmetry assumption $\mathbb{E}[W] = 0$. In our simulations, the empirical mean satisfies $\|\mathbb{E}[W]\|^2 \ll \mathbb{E}[\|W\|^2]$; hence the numerical variance can be compared directly with the theoretical quantity $\bar{V}(W)$. In this numerical experiment, some hyperparameters are fixed, including batch size $b = 100$, the learning rate $\alpha = 0.0001$, and the number of epochs $t = 1000$ for all simulations. Each case is performed 10 times independently, and the averaged loss function $\mathbb{E}_{ss}[\bar{L}(W)]$ is computed from these results. The experiments are conducted using two different settings: (1) A simple three-layer DNN model trained on the MNIST dataset, and (2) CNN-based models including a two-layer CNN and a VGG model [29] trained on the CIFAR-10 dataset.

1) Three-layer DNN model: In this section, we show the numerical results for a deep neural network model with three layers. First, we show the results of the time evolution of the averaged variance. In Figure 1(a), by examining the time evolution of these variances in the three-layer DNN, we observe in this case that when $\sigma_0 \leq 0.1$, the steady-state variance at the end of the training epoch is larger than the initial variance, whereas for $\sigma_0 \geq 0.2$, the final variance falls below the initial value. Intuitively, the DNN appears

to adjust its weights throughout the training process to converge toward a locally optimal variance.



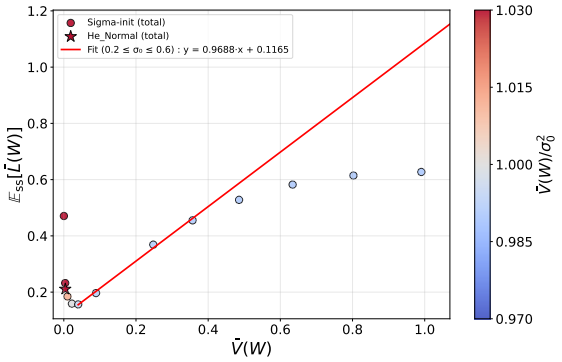
(a) Time evolution of the averaged weight variance.



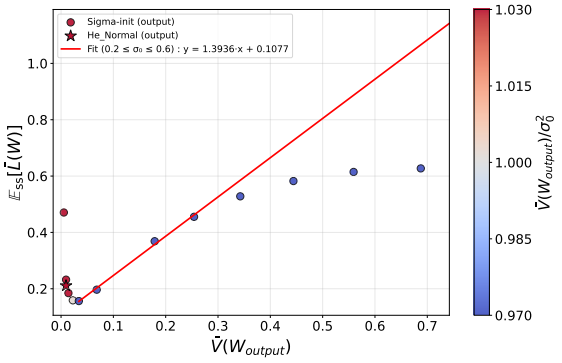
(b) Validation accuracy for different σ_0 values.

Fig. 1: (a) Time evolution of the averaged weight variance $V(W,t)$ over $N = 10$ runs of a three-layer deep neural network trained with SGD under different initial variances σ_0. Results are shown for MNIST. (b) The figure shows the accuracy for various σ_0 values on MNIST. When $\sigma_0 = 0.15$, the validation accuracy takes the highest value. The black dashed line represents the result using He-Normal initialization.

Note that for tiny values of σ_0, the steady-state variance $\bar{V}(W)$ seems to violate the bound of Equation (18). This issue likely stems from the *vanishing gradient problem*, a common challenge in training deep neural networks using too small initial weights. This is a technical problem; thus, we have to discuss our theoretical analysis without this region. In Figure 2(a), with a reasonably small region of σ_0 implemented on MNIST data, the result shows that the optimization process gives a lower loss value than a linear bound $K_1\bar{V}(W)$ on the right-hand side of (18). In this sense, (18) holds for this numerical simulation. In particular, the optimized σ_0, the one satisfying the optimal condition (III.3), provided a slightly lower loss function than the case of using He-normal initialization, which is a common initialization distribution.



(a) Overall variance for the network.



(b) Variance of the output layer.

Fig. 2: The relationship between the expected training loss $\mathbb{E}[L(W)]$ and the steady-state variance $\bar{V}(W)$ in Equation (18) for a DNN network trained using SGD under different initial variances σ_0^2. The red linear function is the most relaxed bound for small σ_0 shown in Equation (18), and a star marker represents the result using He-Normal initialization. For each σ_0, weights are independently sampled from the same Gaussian distribution $\mathcal{N}(0, \sigma_0^2)$, and the average variance $\bar{V}(W)$ over 10 runs is plotted on the x-axis with the MNIST dataset. The figures show (a) the overall variance for this network, (b) the variance of the output layer. Each point is colored according to the ratio $\bar{V}(W)/\sigma_0^2$.

Figure 2(a) highlights the ratio $\bar{V}(W)/\sigma_0^2$ as a function of the overall converged variance $\bar{V}(W)$. At $\sigma_0 = 0.15$, this ratio approaches unity, which corresponds to the point in Figure 2(a) (loss vs variance of overall layer) and 2(b) (loss vs variance of the output layer) attaining the minimum expected loss. Hence, satisfying $\sigma_0^2 = \bar{V}(W)$ realizes the lowest loss, which again confirms our theoretical result.

Finally, we illustrate the time evolution of the validation accuracy for a learning process with our three-layer DNN model. Figure 1(b) shows that when the optimal $\sigma_0 = 0.15$ is selected for the initialization,

the model achieves the highest accuracy in the final epoch. Furthermore, the trajectories corresponding to the standard deviations around this optimal σ_0 also achieve better accuracies than the case of He normal initialization. In summary, our numerical experiments for the DNN network under SGD confirm the following key points:

- In Figure 1(a), we observe that the learning dynamics tend to converge toward the ideal scale of the weight distribution. Specifically, the algorithm dynamically modulates the spread of the weight distribution by adapting the scale of the parameters around the optimal σ_0 to align itself with the theoretically optimal size.
- For initial variances in the range $\sigma_0^2 \leq 1$, the expected value of the loss function satisfies our bound of Equation (18):

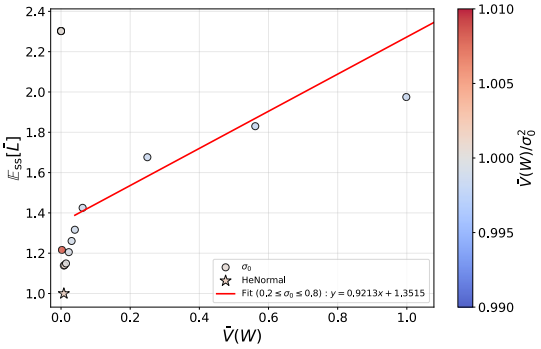
$$\mathbb{E}_{\text{ss}}[L(W)] \leq K_1 \bar{V}(W)$$

  This bound is the most relaxed bound function for the small σ_0 region.
- The theory predicts that the optimal initialization should satisfy

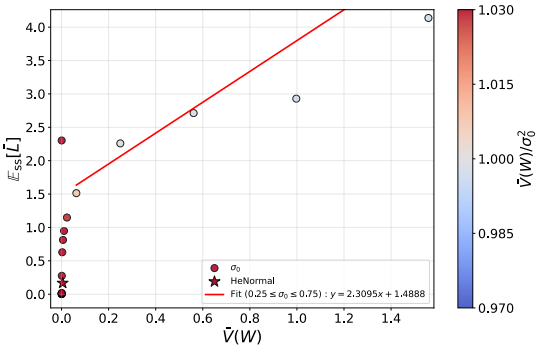
$$\sigma_0^2 = \bar{V}(W).$$

  In practice, DNN models are capable of achieving this optimal condition if they start from an ideal scale of the initialization distribution, especially in the Gaussian case. In particular, although the He-normal scheme remains a strong baseline, our results show that an alternative initialization variance σ_0, derived from the proposed optimal condition, achieves even better training performance, particularly in terms of lower loss, than He-normal initialization. Physically, this optimal condition means that SGD begins its search in exactly the region of parameter space that will eventually contain the minimum-loss solution. Because the exploration range is already well-matched to the final steady-state spread, every weight is updated inside this region from the outset, enabling the network to converge more quickly and reliably to well-optimized values.
- While the theoretical optimal condition is expressed in terms of the overall ratio $\bar{V}(W)/\sigma_0^2 = 1$, our numerical experiments suggest that lower loss is achieved when this condition is also approximately satisfied in individual layers. In particular, Figure 2(b) indicates that the variance ratio of the output layer affects the loss. This suggests that satisfying the optimal condition is important not only for the overall variance but also for each individual layer.

2) Modern network models: In this section, we present numerical results for more realistic architectures trained on the CIFAR-10 dataset.



(a) Two-layer CNN model.



(b) VGG model.

Fig. 3: The relationship between the expected training loss $\mathbb{E}[L(W)]$ and the steady-state variance $\bar{V}(W)$ in Equation (18) for a modern complex network trained using SGD under different initial variances σ_0^2. The red linear function is the most relaxed bound for small σ_0 shown in Equation (18), and a star marker represents the result using He-Normal initialization. For each σ_0, weights are independently sampled from the same Gaussian distribution $\mathcal{N}(0, \sigma_0^2)$, and the average variance $\bar{V}(W)$ over 10 runs is plotted on the x-axis with the CIFAR-10 dataset. The figures show (a) the CNN two-layer model and (b) the VGG Net model. Each point is colored according to the ratio $\bar{V}(W)/\sigma_0^2$.

Figure 3(a) shows the results for the two-layer CNN. In this case, He-normal initialization gives the lowest loss among the tested initializations. This appears to be closely related to the fact that the variance ratio $\bar{V}_\ell(W)/\sigma_{0,\ell}^2$ in He-normal initialization remains close to unity across all layers, including the convolutional layers. In this sense, He-normal provides an initialization scale that is already well aligned with the practically favorable condition ob-

served in our earlier experiments. By contrast, the VGG results in Figure 3(b) do not exhibit the same trend as in the DNN and two-layer CNN cases. In particular, the loss is no longer organized simply by the ratio $\bar{V}(W)/\sigma_0^2$, and the relationship predicted by the theory is not clearly observed for this model. This may indicate that, for a deeper and more complex architecture such as VGG, the assumptions underlying our analysis are no longer sufficiently reasonable. In particular, our theoretical argument relies on a local quadratic approximation of the loss function around a local minimum, whereas such an approximation may not well describe the optimization landscape relevant to VGG training.

V. Conclusion

In this article, we derived an analytic optimal condition for the initial variance σ_0^2 of Gaussian weight initialization under SGD and examined it numerically using a fully connected three-layer neural network model on MNIST and modern convolutional architectures on the CIFAR-10 dataset. For the three-layer model, our experiments showed that the theoretically optimal σ_0 consistently achieves lower training loss and higher test accuracy than the widely used He-normal initialization. For the two-layer CNN model, He-normal initialization gave the best performance among the initializations tested in our experiments, which appears to be related to the fact that it satisfies the optimal variance condition well in this case. In particular, our results indicate that He-normal satisfies the optimal condition at each layer, and this provides a mathematical perspective on why He-normal initialization performs well in practice. In some shallow DNN models, the loss function structure usually consists of multiple quadratic local minima, and therefore, the learning process is practically non-ergodic. In this sense, studying the ideal condition for the initialization is important because the influence of the initial state persists in the steady state, and starting the learning process from an ideal situation likely yields the most efficient result. By contrast, for the VGG model, we observed that the predicted variance-loss relation is no longer clearly observed. This suggests that, in deeper and more complex architectures, the assumptions underlying our analysis, in particular the local quadratic approximation around a local minimum, may no longer provide a sufficient description of the effective optimization landscape. Physically, choosing the initialization distribution is equivalent to choosing the exploration region in the learning process, especially for small learning rates. This optimal condition means that the exploration region of SGD already matches the region where the final state is located. Every weight is updated inside a productive basin from the first step, yielding more reliable optimization in landscapes riddled with multiple local minima. Although a lower training loss does not always translate into the highest test accuracy, our experiments show that the optimal Gaussian parameter σ_0 improves both metrics in practice. It is still meaningful for the process to achieve the lowest loss value in ML framework.

While previous works showed a few mathematically grounded criteria for choosing initialization hyperparameters, our method provides a concrete guideline by evaluating the information distance between initialization and steady state in a learning process. For future perspective, discovering algorithms that actively search for the optimal initialization variance could further improve the efficiency of deep learning optimization. Moreover, although our current analysis relies on the Kullback–Leibler divergence, replacing it with metrics such as the Wasserstein distance would naturally recast the problem within the framework of optimal transport and may lead to richer theoretical insights. Moreover, we can also consider the heavy-tailed distribution for initialization. This aspect has been studied, for example, by [30]. However, in our theorem, heavy-tailed initializations do not deliver uniformly high accuracy across trials. Indeed, their divergent second moment often makes performance less stable than in the Gaussian case. Nevertheless, they sometimes yield surprisingly good results in individual runs, suggesting that, under the right conditions, the broader exploration they induce can land the optimizer in favorable regions in the loss function space. Identifying when and why these rare successes occur remains an interesting area for future work.

References

[1] J. Wang, Y. Chen, S. Hao, X. Peng, and L. Hu, "Deep learning for sensor-based activity recognition: A survey," *Pattern recognition letters*, vol. 119, pp. 3–11, 2019.

[2] W. Zhan, Y. Chen, Q. Liu, J. Li, M. D. Sacchi, M. Zhuang, and Q. H. Liu, "Simultaneous prediction of petrophysical properties and formation layered thickness from acoustic logging data using a modular cascading residual neural network (mcarnn) with physical constraints," *Journal of Applied Geophysics*, vol. 224, p. 105362, 2024.

[3] Y. Song and A. Zhang, "From black box to physically interpretable: Trustworthy computing for ai-driven decision-making and control," *Journal of Automation and Intelligence*, 2025.

[4] A. Rakhlin, O. Shamir, and K. Sridharan, "Making gradient descent optimal for strongly convex stochastic optimization," *arXiv preprint arXiv:1109.5647*, 2011.

[5] S. Hochreiter and J. Schmidhuber, "Flat minima," *Neural computation*, vol. 9, no. 1, pp. 1–42, 1997.

[6] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang, "On large-batch training for deep learning: Generalization gap and sharp minima," *arXiv preprint arXiv:1609.04836*, 2016.

[7] G. K. Dziugaite and D. M. Roy, "Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data," *arXiv preprint arXiv:1703.11008*, 2017.

[8] M. Gurbuzbalaban, U. Simsekli, and L. Zhu, "The heavy-tail phenomenon in sgd," in *International Conference on Machine Learning*. PMLR, 2021, pp. 3964–3975.

[9] L. Hodgkinson and M. Mahoney, "Multiplicative noise and heavy tails in stochastic optimization," in *International Conference on Machine Learning*. PMLR, 2021, pp. 4262–4274.

[10] U. Simsekli, O. Sener, G. Deligiannidis, and M. A. Erdogdu, "Hausdorff dimension, heavy tails, and generalization in neural networks," *Advances in Neural Information Processing Systems*, vol. 33, pp. 5138–5151, 2020.

[11] K. Lehman, A. Durmus, and U. Simsekli, "Approximate heavy tails in offline (multi-pass) stochastic gradient descent," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[12] S. L. Smith, P.-J. Kindermans, C. Ying, and Q. V. Le, "Don't decay the learning rate, increase the batch size," *arXiv preprint arXiv:1711.00489*, 2017.

[13] D. Mishkin and J. Matas, "All you need is a good init," *arXiv preprint arXiv:1511.06422*, 2015.

[14] D. Arpit, V. Campos, and Y. Bengio, "How to initialize your network? robust initialization for weightnorm & resnets," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[15] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 2010, pp. 249–256.

[16] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

[17] B. Hanin and D. Rolnick, "How to start training: The effect of initialization and architecture," *Advances in neural information processing systems*, vol. 31, 2018.

[18] P. Goyal, P. Dollár, R. Girshick, P. Noordhuis, L. Wesolowski, A. Kyrola, A. Tulloch, Y. Jia, and K. He, "Accurate, large minibatch sgd: Training imagenet in 1 hour," *arXiv preprint arXiv:1706.02677*, 2017.

[19] E. Hoffer, I. Hubara, and D. Soudry, "Train longer, generalize better: closing the generalization gap in large batch training of neural networks," *Advances in neural information processing systems*, vol. 30, 2017.

[20] Q. Li, C. Tai, and E. Weinan, "Stochastic modified equations and dynamics of stochastic gradient algorithms i: Mathematical foundations," *Journal of Machine Learning Research*, vol. 20, no. 40, pp. 1–47, 2019.

[21] X. Dai and Y. Zhu, "On large batch training and sharp minima: a fokker–planck perspective," *Journal of Statistical Theory and Practice*, vol. 14, no. 3, p. 53, 2020.

[22] S. Mandt, M. D. Hoffman, and D. M. Blei, "Stochastic gradient descent as approximate bayesian inference," *Journal of Machine Learning Research*, vol. 18, no. 134, pp. 1–35, 2017.

[23] L. Ziyin, H. Li, and M. Ueda, "Noise balance and stationary distribution of stochastic gradient descent," *Physical Review E*, vol. 111, no. 6, p. 065303, 2025.

[24] P. Chaudhari and S. Soatto, "Stochastic gradient descent performs variational inference, converges to limit cycles for deep networks," in *2018 Information Theory and Applications Workshop (ITA)*. IEEE, 2018, pp. 1–10.

[25] U. Simsekli, L. Sagun, and M. Gurbuzbalaban, "A tail-index analysis of stochastic gradient noise in deep neural networks," in *International Conference on Machine Learning*. PMLR, 2019, pp. 5827–5837.

[26] H. A. Kramers, "Brownian motion in a field of force and the diffusion model of chemical reactions," *physica*, vol. 7, no. 4, pp. 284–304, 1940.

[27] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin, *Bayesian data analysis*. Chapman and Hall/CRC, 1995.

[28] S. Jastrzebski, Z. Kenton, D. Arpit, N. Ballas, A. Fischer, Y. Bengio, and A. Storkey, "Three factors influencing minima in sgd," *arXiv preprint arXiv:1711.04623*, 2017.

[29] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[30] M. Gurbuzbalaban and Y. Hu, "Fractional moment-preserving initialization schemes for training deep neural networks," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2021, pp. 2233–2241.