Neural Network Weights Do Not Converge to Stationary Points: An Invariant Measure Perspective

Jingzhao Zhang [1] **Haochuan Li** [2] **Suvrit Sra** [2] **Ali Jadbabaie** [2]

Abstract

This work examines the deep disconnect between existing theoretical analyses of gradient-based algorithms and the practice of training deep neural networks. Specifically, we provide numerical evidence that in large-scale neural network training (e.g., ImageNet + ResNet101, and WT103 + TransformerXL models), the neural network's weights *do not* converge to stationary points where the gradient of the loss is zero. Remarkably, however, we observe that even though the weights do not converge to stationary points, the progress in minimizing the loss function halts and training loss stabilizes. Inspired by this observation, we propose a new perspective based on ergodic theory of dynamical systems to explain it. Rather than studying the evolution of weights, we study the evolution of the distribution of weights. We prove convergence of the distribution of weights to an approximate invariant measure, thereby explaining how the training loss can stabilize without weights necessarily converging to stationary points. We further discuss how this perspective can better align optimization theory with empirical observations in machine learning practice.

1. Introduction

It would not be controversial to claim that currently there exists a wide gulf between theoretical investigations of convergence to (approximate) stationary points for non-convex optimization problems and the empirical performance of popular algorithms used in deep learning practice. Due to the intrinsic intractability of general nonconvex problems, theoretical analysis of nonconvex optimization problems often focuses on the rates of convergence of gradient norm $\|\nabla f(\theta)\|$ instead of the suboptimality $f(\theta) - \min_\theta f(\theta)$. The vast theoretical literature on optimization for machine learning has documented the recent progress in this area. In particular, optimal gradient-based algorithms and rates have been identified in various nonconvex settings, including deterministic, stochastic and finite-sum problems (Carmon et al., 2017; Arjevani et al., 2019; Fang et al., 2018).

In addition to theoretical interest in nonconvex problems, a practical motivation for studying nonconvex convergence analyses is to improve the large-scale optimization methods that are used in machine learning practice, especially in training deep neural networks. As neural network models allow for efficient gradient evaluations, gradient-based algorithms remain the dominant methods to tune network parameters. Naturally, great effort has been dedicated to theoretical understanding of gradient-based optimizers.

But despite the rapid progress in the theory of gradient-based algorithms, this theory has had a limited impact on real-world neural network training. And the gap between theory and practice is as wide as ever. For example, even though the variance reduction technique theoretically accelerates convergence, recent empirical evidence in (Defazio & Bottou, 2018) suggests that it may be ineffective in speeding up neural network training. On the other extreme, ADAM (Kingma & Ba, 2014) is among the most popular algorithms in neural network training, yet its theoretical convergence was proven to be incorrect (Reddi et al., 2019). Despite dubious theoretical properties, ADAM is still among the most effective optimizers.

Our goal is to address the ineffectiveness of applying theoretical convergence rates to stationarity in neural network training by identifying a fundamental gap between theoretical convergence and empirical convergence. First, we provide evidence that in many challenging experiments (e.g., ImageNet, Wiki103) where the model does not overfit the data, gradient-based optimization methods do not converge to stationary points as theory mandates. This mismatch questions applicability of usual theory as applied to training neural networks. The reason for such a surprising divide is that most optimization analyses for deep learning either assume smoothness directly which leads to convergence to station-

[1]IIIS, Tsinghua University [2]Massachusetts Institute of Technology. Correspondence to: Jingzhao Zhang <jingzhaoz@mail.tsinghua.edu.cn>, Haochuan Li <haochuan@mit.edu>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

ary points using classical analyses, or prove smoothness and fast convergence by relying explicitly on overparametrization. However, our empirical investigations reveal that the key premise of the theory–pointwise convergence to a fixed point—*may not happen at all in practice!*

Motivated by this observation, we aim to answer the following question in the rest of this work: *how should one define and analyze the convergence of gradient-based optimization methods, when the training loss seems to converge, yet the gradient norm does not converge to* 0*?*[1]

We propose a new lens through which one should view convergence: rather than convergence of weights, we postulate that the convergence should be viewed in terms of invariant measures as used in the ergodic theory of dynamical systems. Building on classical results from this literature, we then show how this new perspective is also consistent with some curious findings in neural network training, such as relaxed smoothness (Zhang et al., 2019) and edge of stability (Cohen et al., 2021; Wu et al., 2018). More concretely, our contributions are summarized as follows:

- We empirically verify through ResNet training and Transformer-XL training in a wide range of applications that the iterates do not converge to a stationary point as existing theory predicts.
- We propose an invariant measure perspective from dynamical systems to explain why the training loss can converge without the iterates converging to stationary points.
- Most importantly, we show that our theorems on diminishing gain of the loss without vanishing of the gradient apply to neural network training even without standard global Lipschitzness or smoothness assumptions.

It is worth noting that our analysis for deep learning, though holds under a very generic setup without assuming overfitting or bounded Lipschitzness, only states vanishing change in average training loss. It does not comment on the actual loss values. Consequently, much remains to be done based on our proposed view. However, our observations relate to interesting phenomena such as decay of function values, edge of stability, and relative smoothness. We conclude our work with a detailed discussion of the above points. We believe that it provides a paradigm shift in how convergence in deep learning should be defined and studied.

1.1. Related work

Several recent empirical findings discuss the instability of neural network predictions even after training loss has converged, and they inspire us to investigate whether convergence to stationary points actually happens. Henderson et al. (2017) analyze the stability of policy reward in reinforcement learning and observe large variations. Madhyastha & Jain (2019) study the instability for interpretation mechanisms. In (Bhojanapalli et al., 2021), the authors note that though image classification has relatively stable accuracy, the actual prediction on individual images has large variation. A few very recent results report similar large oscillations in Cifar10 training (Li et al., 2020; Kunin et al., 2021; Lobacheva et al., 2021), though the authors focus on SDE approximation or batch normalization. Our work instead focuses on the connection to nonconvex optimization theorems. In addition, we learned from recent studies (Cohen et al., 2021; Zhang et al., 2019; 2020) that assumptions on noise and smoothness not only fail but can further adversarially adapt to the step size choice, further suggesting that optimizers may not find stationary points in deep learning.

On the theory side, two lines of work study convergence beyond finding stationary points, and hence are closely related to this paper. One line studies the non-convergence of dynamics of algorithms in games or multiobjective optimization (Hsieh et al., 2019; Cheung & Piliouras, 2019; Papadimitriou & Piliouras, 2019; Letcher, 2020; Flokas et al., 2020). Another models SGD dynamics via Langevin dynamics (Cheng et al., 2020; Li et al., 2020; Gurbuzbalaban et al., 2021). Our work differs from the Langevin dynamics view in that we do not aim to achieve global mixing. As a consequence, we avoid the unrealistic assumption in Langevin analysis that the noise level is inversely proportional to the step size.

2. A motivating example: ImageNet + ResNet

We start our exposition by providing some experimental result showing that the traditional notion of convergence for nonconvex functions *does not* occur in deep neural network training. Our experiments are based on one of the most popular training schemes, where we train ResNet101 on ImageNet. More details can be found later in Appendix A and in our released code repository.

To explain the quantities of interest, we first define our notation. Let $S = \{(x^i, y^i)\}_{i=1}^N$ be the dataset. Let $f(x, \theta)$ to denote the neural network function with model parameters θ and input data x. We use ℓ to denote the loss function such as cross-entropy after softmax. We would like to investigate, during training, the evolution of the following quantities: training loss, gradient norm, and noise. Mathematically, they are defined as follows respectively,

$$\begin{aligned} &\mathcal{L}_S(\theta_k) := \tfrac{1}{N}\textstyle\sum_{i=1}^N \ell(f(x^i,\theta_k),y^i), \\ &\|\nabla L_S(\theta_k)\|_2 := \|\tfrac{1}{N}\textstyle\sum_{i=1}^N \tfrac{\partial}{\partial\theta}\ell(f(x^i,\theta_k),y^i)\|_2, \\ &\sigma(\theta_k) := \sqrt{\tfrac{1}{N}\textstyle\sum_{i=1}^N \left\|\nabla L_S(\theta_k) - \tfrac{\partial}{\partial\theta}\ell(f(x^i,\theta_k),y^i)\right\|_2^2}, \end{aligned} \tag{1}$$

where $\|\cdot\|_2$ is the standard vector ℓ_2 norm.

[1] More pedantically, the iterates do not converge to even an ϵ-stationary point as predicted by standard theory.

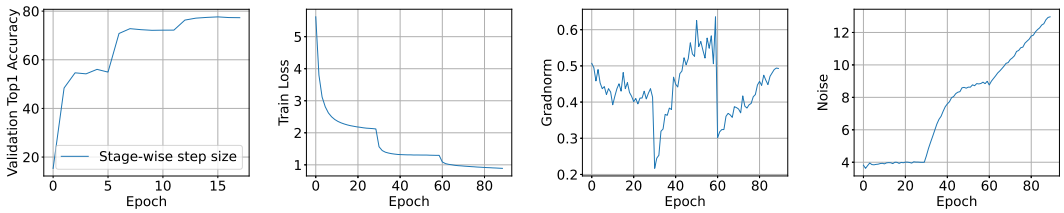

Figure 1. The validation accuracy and the quantities of interest (1) for the default training schedule of ImageNet + ResNet101 experiment.

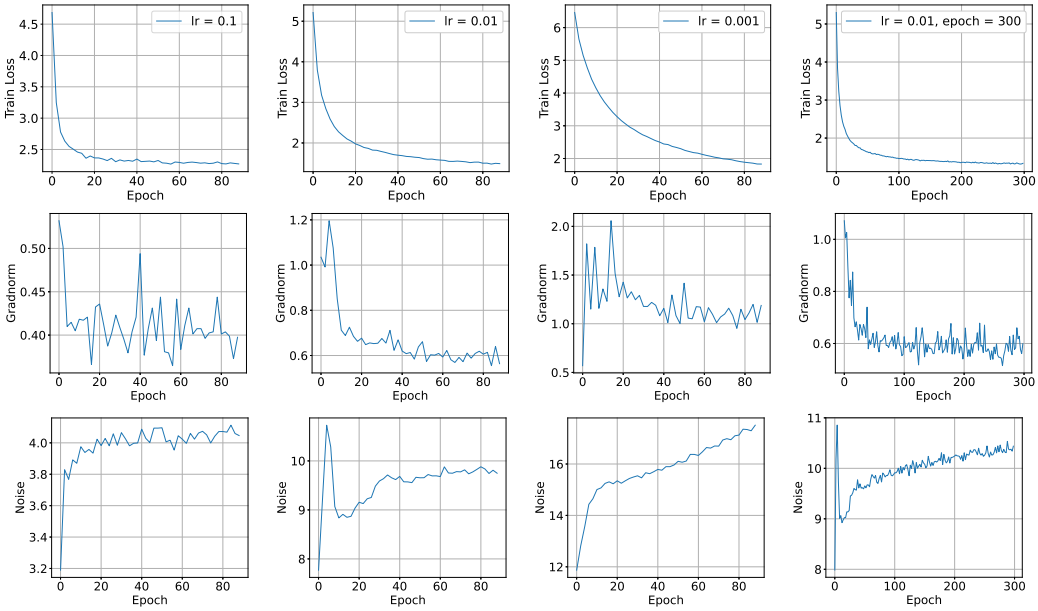

Figure 2. The quantities of interest (1) vs epoch for the constant learning rate training schedule in ImageNet experiments. The learning rate is set to be $0.1, 0.01, 0.001, 0.01$ respectively starting from the left column. All models are trained for 90 epochs, except that the last experiment in the column ran for 300 epochs

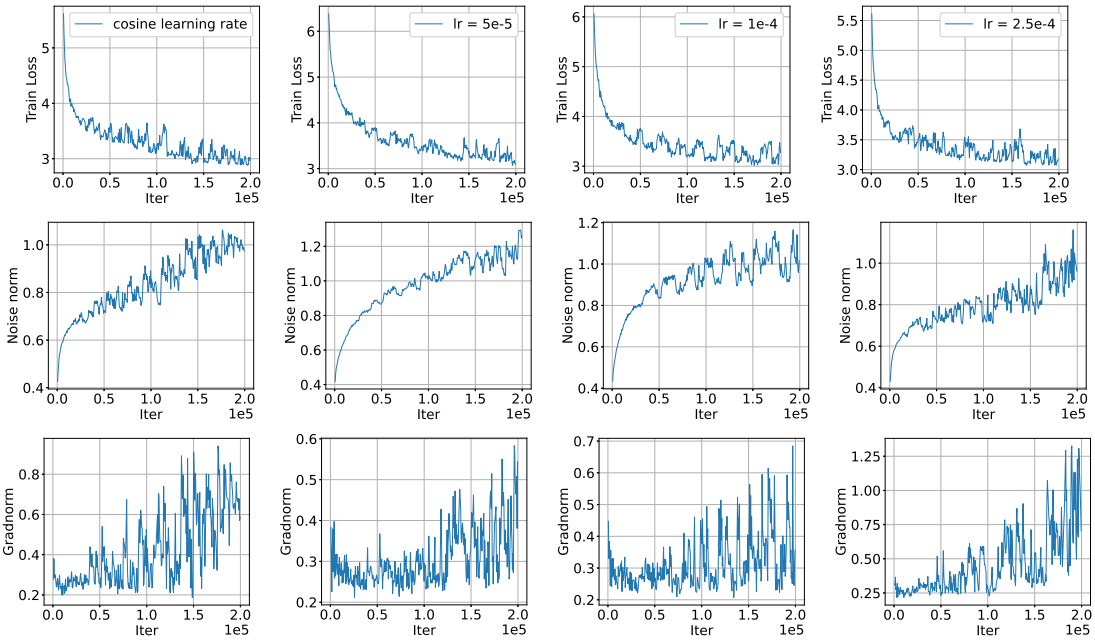

Figure 3. The estimated stats vs epoch for the transformer XL training. The learning rate is set to be cosine learning rate with $\eta = 0.00025$ in the first column. The learning rates are constant learning rates with $\eta = 0.00005, 0.0001, 0.00025$ from the second to the last column.

We adopt the standard training schedule following (He et al., 2016), i.e., the learning rate starts at 0.1 and is decayed by a factor of 10 every 30 epochs. The evolution of the aforementioned quantities is plotted in Figure 1. We make the following immediate observations:

- Within each period where the step size is held constant, the change in loss converges to 0.
- The gradient norm does not converge to 0 despite the fact that the loss function converges. In fact, the gradient norm stays roughly unchanged.
- The noise level (in the stochastic gradient) increases during training.

The above observations suggest that there is *a tremendous gap between theory and practice*. Much of the research on nonconvex optimization theory focuses on the convergence rate of gradient norms under a bounded-smoothness, bounded-noise setup. Faster algorithms are designed under this guidance. However, in practice, we find that the convergence of the training loss *does not* require the convergence of gradient norms. This mismatch may be the reason why techniques such as variance reduction or local regularization combined with Nesterov-momentum have had limited practical use, despite their massive theoretical popularity.

3. A systematic investigation

In this section, we will provide a set of experiments to systematically understand when and (hopefully) why the neural network parameters do not converge to stationary points as theory mandates. In particular, we will try to test the following **hypotheses** in the experiments:

1. The nonconvergence is due to the fact that the step size is not small enough or the model is not trained long enough.
2. This phenomenon is restricted to the ResNet + ImageNet task, or models with non-differentiable ReLUs.
3. The large gradient norm is due to estimation error.

In the end, we will see that these hypotheses **fail** to hold and that the phenomenon is quite common in large-scale tasks.

3.1. Different learning rates and training schedules

One immediate question following the observation in Figure 1 is whether the observed phenomenon holds solely for a particular stage-wise learning rate, which is not very common in theoretical analysis. To address this question, we run the same ResNet101 model on ImageNet just as before, except that we now use a constant learning rate across all 90 epochs of training. The evolutions of the quantities in (1) are summarized in Figure 2. A quick glance at the plots verifies that the gradient norm does not converge to 0 in any of the experiments. We further notice that, surprisingly, a smaller learning rate leads to a larger gradient norm, larger stochastic gradient noise intensity, and larger sharpness as observed in (Cohen et al., 2021). We will further discuss the implications of these observations later in the paper.

As the loss curves in the last two rows of Figure 2 are still decreasing, another question could be that we didn't run the experiment long enough to achieve actual convergence. To address this question, we continue the second row experiment (step size $\eta = 0.01$) for 300 epochs and present the result in the rightmost column of Figure 2. We can see that no clear progress was made after about 50 epochs.

The above experiments show that in ImageNet + ResNet101 experiment, the parameters do not converge to stationary points. In the next section, we test whether this phenomenon is restricted to the particular data set and architecture.

3.2. Transformer XL experiments

We run Transformer-XL training on WT103 dataset for the language modeling task following the implementation of the original authors (Dai et al., 2019b). Our training procedure is exactly the same as the official code, except that we reduce the number of attention layers for the baseline model from 6 to 4. Aside from training with a cosine learning rate schedule with initial learning rate $\eta = 0.00025$, we also experimented with different constant learning rates. The result is summarized in Figure 3. We found that the observations made before also apply to transformer XL.

3.3. Refuted hypotheses from the systematic study

With the above set of experiments, we can already **exclude** the hypotheses at the beginning of this section.

First, in the rightmost column of Figure 2, we find that after running 300 epochs with a smaller step size, though the training loss dropped significantly, the gradient norm did not decrease. *This confirms that even the qualitative analysis (let alone the quantitative convergence rates) on when gradient norm gets smaller from canonical optimization theory is not applicable to neural network training.*

Second, we see that the nonzero-gradient phenomenon in TransformerXL (Dai et al., 2019a) training is even more prominent. In addition, as TransformerXL is differentiable, this also excludes the chance that oscillation is caused by non-differentiability. The **third** conjecture is refuted due to our estimation precision discussed later in Appendix A.2 with additional experimental details. We also observed that in our Cifar10 experiment in Appendix A.1, the gradient norm can indeed go to zero.

Given the above evidence, we believe that the convergence of training loss without reaching stationary points is caused

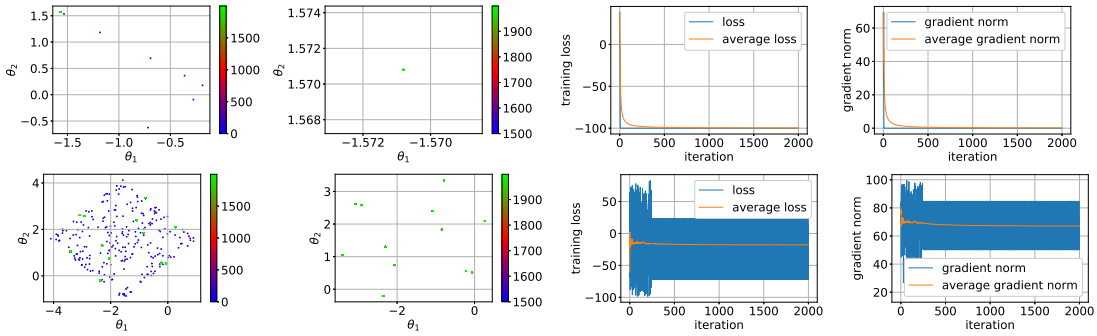

Figure 4. Synthetic experiment. The learning rate is set to be 0.01 and 0.04 for the first and second row respectively. Column I: the whole trajectory in 2000 iterations, where the scatter points correspond to iterates and the color of a point represents which iteration it is at; Column II: the trajectory in the last 500 iterations to show the convergence behavior; Column III: training loss and average training loss vs iteration, where the average is taken over iterations; Column IV: gradient norm and average gradient norm vs iteration.

by more fundamental and nontrivial reasons. To understand this phenomenon, we later will develop a notion of convergence based on the theory of dynamical systems.

Before diving into our theorems, we start with an interesting observation in Figure 5. Here, we evaluate the *full batch* training loss in two ways. The left one is to compute a moving average during the training epoch:

$$\text{Loss} = \frac{1}{N}\sum_{i=1}^{N} \ell(\theta_i, x_i), \tag{2}$$

where i denotes the iteration number within an epoch of N minibatches, x_i denotes data from i_{th} minibatch and θ_i denotes the network parameter at iteration i. The other is to compute the full batch loss at the last iteration N,

$$\text{Loss} = \frac{1}{N}\sum_{i=1}^{N} \ell(\theta_N, x_i). \tag{3}$$

We notice that though both evaluations consumed the entire dataset, averaging the minibatch losses across all training iterations leads to a much more smooth loss curve than evaluating all the minibatch losses at a fixed iteration. Hence, one explanation is that the **time average of the loss** rather than the **iteration-wise loss** converges, while the gradient norm is nonzero due to nonsmoothness, and that the actual weight iterates keep oscillating. To provide more intuition, in the next subsection, we provide a conceptual explanation through a synthetic experiment.

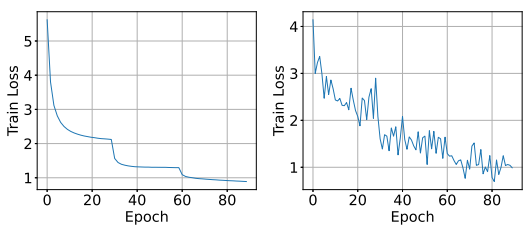

Figure 5. The **full batch** training loss vs epochs in default ImageNet training. The left plot computes loss in the usual way as an moving average during training (2), where as the right plot computes the loss at the last iteration of a training epoch using (3).

3.4. An explanatory synthetic experiment

The curious phenomenon discussed above is not limited to neural network training. In what follows we present a simple synthetic example to illustrate the intuition behind the convergence behavior to unstable cycles rather than stationary points.

To this end, we simulate gradient descent on the objective function $f(\theta_1, \theta_2) = 100 \sin\theta_1 \sin\theta_2$ whose smoothness and Lipschitzness parameters are both $L_f = 100$. It is well known that gradient descent with a learning rate $\eta < 2/L_f = 0.02$ provably converges to stationary points for such a smooth function. As shown in the first row of Figure 4, the iterates converge to a fixed point very fast with $\eta = 0.01$. Moreover, the gradient norm converges to zero, which means a stationary point is reached at convergence.

However, when $\eta > 2/L_f$, which is often the case for neural network training, gradient descent no longer converges to stationary points as shown in the second row of Figure 4 with $\eta = 0.04$. During the last 500 iterations, the iterates only take values around a few points and keep oscillating among them. As a result, the training loss and gradient norm also oscillate and do not converge in the usual sense. However, the oscillation among these points follows some periodic pattern. If we collect all the iterates during a long enough training process, their empirical distribution will converge to a discrete distribution over those points that capture the periodic pattern. Then if we take an average of the training losses or gradient norms over time, it must converge to the average value of the periodic points, as shown in the last two images in Figure 4. However, although the average gradient norm converges, the convergence value can not be zero in presence of oscillation, as gradient descent makes no updates if the gradient is zero.

The above example shows that the key to function value convergence could be that a time average rather than a spatial average is taken in evaluating the loss. Hence, the convergence only happens in **time average** sense.

4. Convergence beyond stationary points

We saw above that even though the per-iteration loss does not converge, the time average with a long enough window size can converge. In this section, we provide a simple mathematical analysis to explain why that happens. In particular, we prove that the change in training loss evaluated as a time average converges to 0 for neural networks. Our analysis is motivated by, and follows the proof of the celebrated Krylov-Bogolyubov theorem. As a result, we refer to our interpretation as the *invariant measure perspective*.

Particularly, we say a measure μ is an **invariant measure** for the map $F : \mathcal{X} \to \mathcal{X}$ if for any measurable set A

$$\mu(A) = \mu(F^{-1}(A)) = \int_\theta \mathbb{1}\{F(\theta) \in A\} d\mu(\theta),$$

where $F^{-1}(A) = \{\theta | F(\theta) \in A\}$. Notice that if F is a stochastic update, then this should be read as

$$\mu(A) = \mu(F^{-1}(A)) = \int_\theta \mathbb{P}\{F(\theta) \in A\} d\mu(\theta). \tag{4}$$

In other words, the pushforward of μ under F stays unchanged, $F\#\mu = \mu$.

Invariance of measure is closely related to convergence of function values. To see this, consider the dynamical system

$$\theta_{t+1} = F(\theta_t).$$

In such a scenario, for any continuous function ϕ, the function value does not change after update when the variable is sampled from an invariant measure,

$$\mathbb{E}_{\theta\sim\mu}[\phi(\theta)] = \mathbb{E}_{\theta\sim\mu}[\phi(F(\theta))].$$

Recall that our key insight is that the convergence of the training loss occurs in a time-average sense, which naturally leads to the following notion of empirical measure:

$$\mu_k := \tfrac{1}{k} \textstyle\sum_{t=1}^k \delta_{\theta_t}, \tag{5}$$

where δ_θ denotes the Dirac measure supported on the value θ, i.e., $\delta_\theta(A) = 1$ if and only if $\theta \in A$, and $\{\theta_1, \theta_2, \cdots\}$ are the sequence of iterates generated by the dynamical system. With this notation, we can conveniently write the time average of a scalar function $\phi : \mathcal{X} \to \mathbb{R}$ as

$$\mu_k(\phi) = \mathbb{E}_{\theta\sim\mu_k}[\phi(\theta)]. \tag{6}$$

We focus on the case when the dynamic system $F(\theta_t)$ denotes the SGD update, i.e.,

$$F(\theta_t) = \theta_t - \eta g(\theta_t),$$

where $g(\theta_t)$ denotes the stochastic gradient and η denotes the step size. Next, we will show that the empirical measure converges to an approximately invariant measure as the number of iterations grows.

4.1. Vanishing change in neural network training

We are now ready to provide a theoretical analysis to prove the vanishing gain of training losses in neural network training, and thus explain how the training loss can stabilize even when the norm of the loss function gradient is non-zero. Our analysis is distinct from previous ones (e.g. (Chizat & Bach, 2018; Mei et al., 2019; Jacot et al., 2018)) in the literature in that it *does not* assume global Lipschitzness or smoothness, *does not* rely on bounded noise assumptions, and it *does not* require perfectly fitting the data as in Neural tangent kernel models or mean-field style arguments. Instead, it builds upon minimal practical conditions. The downside of this generality is that we only prove convergence of function values and do not comment on local or global optimality or generalization. We believe much remains to be done here and we have just scratched the surface. We will make more comments on this in Section 5.

To start the discussion, we define the following L-layer deep neural network $f(x, \theta)$, where x is the input and $\theta = (W_0, \ldots, W_{L-1})$ is the network weights:

$$x_{l+1} = \sigma_{l+1}(W_l x_l), \quad l = 0, \ldots L-1 \tag{7}$$
$$f(x, \theta) = x_L, \tag{8}$$

where σ_l is a coordinate-wise activation function (e.g., ReLU or sigmoid). In practice, the last layer usually does not use any activation function so σ_L is the identity mapping. We do not consider pooling layers, convolutional layers, or skip connections for now and it should be easy to extend our analysis to these settings. Iteration (7) does not include batch normalization layers which we will analyze later in this section. Given a training dataset $S = \{(x^i, y^i)\}_{i=1}^N$, the empirical training loss is defined as

$$L_S(\theta) := \tfrac{1}{N} \textstyle\sum_{i=1}^N \ell(f(x^i, \theta), y^i),$$

where $\ell : \mathbb{R}^d \times [d] \to \mathbb{R}$ is a loss function and we assume $\left\|x^i\right\|_2 \leq 1$. The network is trained by SGD with weight decay, which is equivalent to running SGD on the following regularized loss

$$L_S^\gamma(\theta) := L_S(\theta) + \frac{\gamma}{2} \left\|\theta\right\|_2^2,$$

where $\left\|\theta\right\|_2$ denotes the ℓ_2 norm of vectorized θ. We will focus on the most widely used loss function for classification tasks, the cross-entropy after softmax, defined as follows.

$$\ell(x, y) = x_y - \log\left(\textstyle\sum_{j=1}^d e^{x_j}\right), \tag{9}$$

which has the following properties that we will use later.

Lemma 4.1. *The cross-entropy after softmax loss $\ell : \mathbb{R}^d \times [d] \to \mathbb{R}$ defined in (9) satisfies*

1. *If $\max_i x_i - \min_i x_i \leq c$, we have $\ell(x, y) \leq c + \log d$ for any $y \in [d]$.*

2. $\ell(x, y)$ is c_ℓ Lipschitz w.r.t. x for a global constant c_ℓ.

Next, we make the following assumption for the activation function. It holds for ReLU and tanh activation functions.

Assumption 4.2. Each activation function σ_l is (sub)-differentiable and c_σ coordinate-wise Lipschitz for some numerical constant $c_\sigma > 0$. Also assume $\sigma_l(0) = 0$.

Now we can prove the vanishing gain of the function values.

Theorem 4.3. *Suppose Assumption 4.2 holds and θ is initialized within the compact set $C_w := \{(W_0, \ldots, W_{L-1}) : \|W_l\|_{op} \le w\}$ for some $w \le (\gamma/c_\ell c_\sigma^L)^{1/(L-2)}$. Then the iterate θ_k for every k lies in C_w and the empirical measure generated by SGD with a stepsize $\eta \le 1/\gamma$ satisfies with probabiblity $1 - \delta$ that*

$$\mathbb{E}_{\theta\sim\mu_n}[L_S(\theta) - L_S(F(\theta))] = \mathcal{O}(\tfrac{\log(1/\delta)}{\sqrt{n}}).$$

The above theorem states that the change in loss vanishes in a time average sense. The key step in the previous proof is to show that all iterates lie in a compact region almost surely even though the function may not be smooth or Lipschitz continuous. One may suspect that a stronger result should hold that the limit will exist. We show in Section 5.3 that such a statement is highly nontrivial and sometimes false.

We notice that the initialization choice may not always hold in practice, especially when there is batch normalization design. We further note that similar to the above theorem, all (piece-wise) continuous scalar functions including the noise norm are bounded by compactness, and hence should stabilize after long enough training. However, in the third row of Figure 2, the noise norm does not really converge. To explain this observation, we propose the following theorem that studies neural networks with batch normalization.

For simplicity of analysis, we assume the last layer is one of the layers with batch normalization. For a vector x, we use x^2, $|x|$ and $\sqrt{x}$ to denote its coordinate-wise square, absolute value, and square root respectively. In the l-th layer, if it uses batch normalization, given a batch $\mathcal{B} = \{(x^i, y^i)\}_{i=1}^m$ sampled from a distribution $\mathcal{P}_\mathcal{B}$, batch normalization makes the following update from $\{x_{l-1}^i\}_{i\in\mathcal{B}}$ to $\{x_l^i\}_{i\in\mathcal{B}}$:

$$\begin{aligned}
\mu_{\mathcal{B},l-1} =& \tfrac{1}{m}\textstyle\sum_{i\in\mathcal{B}} x_{l-1}^i,\\
\sigma_{\mathcal{B},l-1}^2 =& \tfrac{1}{m}\textstyle\sum_{i\in\mathcal{B}} \left(x_{l-1}^i - \mu_{\mathcal{B},l-1}\right)^2,\\
\hat{x}_l^i =& \tfrac{x_{l-1}^i - \mu_{\mathcal{B},l-1}}{\sqrt{\sigma_{\mathcal{B},l-1}^2 + \epsilon}}, \quad x_l^i = a_l \cdot \hat{x}_l^i + b_l,
\end{aligned}$$

where a_l and b_l are the scale and shift parameters to be trained. We also use SGD with weight decay in training.

Theorem 4.4 (With batch normalization)**.** *Suppose the parameter of batch normalization layer a_L is initialized within the compact set $|a_L| \le 2\sqrt{m}/\gamma$. Then the empirical measure generated by SGD with $\eta \le 1/\gamma$ satisfies with probability $1 - \delta$ that,*

$$\mathbb{E}_{\theta\sim\mu_n,\mathcal{B}\sim\mathcal{P}_\mathcal{B}}[L_\mathcal{B}(\theta) - L_\mathcal{B}(F(\theta))] = \mathcal{O}(\tfrac{\log(1/\delta)}{\sqrt{n}}).$$

where $\mathcal{P}_\mathcal{B}$ denotes the distribution of random minibatches.

We have shown in this section how the expected change of the training loss in per iterate update converges to zero for neural network training without any smoothness or Lipschitzness assumptions. One weakness of our analysis is that the limit of the training loss may not exist. Another caveat is that our gain is measured in terms of empirical measure instead of the last iterate distribution. In the next section, we discuss the many implications and open problems that the invariant measure view brings us.

5. Theorems implications and open questions

We showed that in neural network training, the change in training loss gradually converges to 0, even if the full gradient norm does not vanish. In this section, we will show how this result explains and connects to several observed phenomenons that were not captured by the canonical optimization framework. We then conclude by discussing several limitations of our result and important future directions.

5.1. Edge-of-stability and relaxed smoothness

The invariant measure perspective can also provide insight into the edge-of-stability observation and relaxed smoothness phenomenon. Our argument is heuristic, and somewhat speculative. We believe a rigorous analysis is both interesting and challenging and leave them as future directions.

We start from the equation (15) in the proof of the Theorem 5.2 in Appendix G. If the variable θ follows the distribution of an invariant measure, then by the fact that the expected loss does not change after one SGD update,

$$\begin{aligned}
&\mathbb{E}_{\theta,g}[\|\nabla L_S(\theta)\|_2^2] =\\
&\mathbb{E}_{\theta,g}\left[\eta \iint_0^1 \langle g(\theta), \nabla^2 L_S(\gamma_{\theta,g}(t\tau\eta)) g(\theta)\rangle dt d\tau\right],
\end{aligned}$$

where $g(\theta)$ is the stochastic gradient and $\gamma_{\theta,g(\theta)}(r) = \theta - rg(\theta)$ denotes the line segment. Then we boldly extract an equation that holds in expectation

$$(\nabla)^2 = \eta\mathcal{L}G^2, \tag{10}$$

where ∇ denotes the gradient norm, $\mathcal{L}$ denotes the sharpness in the update direction and G^2 denotes the second moment of stochastic gradients. The only approximation we made is that we replaced the hessian integral along the line segment

$\gamma_{\theta,g(\theta)}$ by sharpness. This equation has some interesting connections to the following two observations.

First, we recall the edge-of-stability framework (Cohen et al., 2021), which observes that the actual smoothness constant during training neural network has an inverse relation to step size. This is true from the above equation if we hold ∇, G constant. Second, in (Zhang et al., 2019), the authors identified a positive correlation between the gradient norm and the smoothness constant. This relation can also be extracted from the equation if η, G are held constant.

In fact, as we observe that in practice, the relation between the sharpness and step size is not a direct inverse but indeed has some negative correlation. Therefore, we believe that through a more rigorous analysis of the property of invariant measures, one could understand why many counter-intuitive behaviors can happen, and provide a more accurate model of the interaction between different quantities.

5.2. Decreasing stepsize leads to smaller objective values

One well-known observation in neural network training is that when the training loss plateaus, reducing the learning rate can further reduce the objective. This phenomenon can be proved in theory if the function has globally bounded noise and smoothness constant. However, as we showed, the smoothness and noise level change adversarially to the step size. In this section, we provide a partial explanation on when a smaller step size can decrease the function value. In particular, we consider the neural network setup introduced in Section 4.1. We make the following assumption:

Assumption 5.1. The neural network is continuously second-order differentiable, though it need not necessarily have bounded smoothness.

Then we can prove that reducing the step size would result in a decrease in function value.

Theorem 5.2. *Consider the stochastic gradient update $F : \mathcal{X} \to \mathcal{X}$ on a compact set defined as $F(\theta) = \theta - \eta g(\theta)$ for a fixed step size $\eta > 0$. Let μ be the invariant distribution such that $\mathbb{E}_{F,\theta\sim\mu}[L_S(\theta)] = \mathbb{E}_{F,\theta\sim\mu}[L_S(F(\theta))]$. If μ is not supported on stationary points (i.e. $\mathbb{E}_{\theta\sim\mu}[\|\nabla f(\theta)\|_2^2] > 0$), then there exists a small enough $c \in (0, 1)$ such that for any positive step size $\eta' < c\eta$, the update $F'(\theta) = \theta - \eta' g(\theta)$ will lead to a smaller function value, i.e.*

$$\mathbb{E}_{F',\theta\sim\mu}[L_S(F'(\theta))] < \mathbb{E}_{\theta\sim\mu}[L_S(\theta)].$$

The above theorem states that once the change in loss vanishes, by selecting a smaller step size, one could further reduce the loss. This reflects the observation in Figure 1. The challenge in the proof is that reducing the step size might lead to worse smoothness that is too large for the step size, and hence may increase the training objective. The proof can be found in Appendix G.

However, Theorem 5.2 only depicts what happens after one-step update rather than the long-term behavior after the iterates generated by a smaller step size converge. We believe that characterizing the shift from one invariant measure to another due to step size update could lead to a better understanding of the convergence rates of optimization algorithms, and is worth future studies.

5.3. Vanishing change vs existence of a limit

Theorems 4.3 and 4.4 show that the update vanishes to zero, yet they do not imply whether the limit $\lim_{t\to\infty} \mu_t(\phi)$ exists. In fact, an explicit counterexample shows that the limit may not exist even for a dynamic system defined on a compact domain with Lipschitz maps.

Theorem 5.3 ((Yoccoz)). *There exist a compact set $\mathcal{X}$, a dynamic system with deterministic continuous map $F : \mathcal{X} \to \mathcal{X}$ and a scalar function $\phi \in C^\infty : \mathcal{X} \to \mathbb{R}$, such that sequence $\frac{1}{n}\sum_{k\le n} \phi(\theta_k)$ has no limit, where $\theta_{k+1} = F(\theta_k)$.*

Given the above negative result, we only know that a subsequence of the series of empirical measures will converge to the invariant set.

Theorem 5.4 (convergence of distribution). *Assume that F maps a compact set $\mathcal{X}$ to itself. Then the empirical distribution has a subsequence converging weakly to an ergodic distribution. In other words, there exists an invariant distribution μ, and a **subsequence** of positive integers $\{n_k\}_{k\in\mathbb{Z}}$ such that $\mu_{n_k} \to_w \mu$.*

The proof of the above theorem is similar to the proof for the Krylov-Bogolyubov theorem. We include the proof in Appendix C for completeness. However, we note that the above two theorems do not make use of the gradient descent update or the neural network architecture. Whether the dynamic system resulting from gradient descent has exactly the same property is left as a challenging future problem.

5.4. Discussion

Our work introduces a paradigm shift in how the convergence of weights and loss function should be analyzed and defined. It suggests neural network training converges to approximate invariant measures *when iterates fail to converge to a single point and the distribution of weights does not converge to a globally unique stationary distribution.* Our results, however, lead to more questions than answers. For example, what do we know about the last iterate instead of empirical distribution? How do the invariant measures of different neural network structures and gradient-based optimizers differ from one another? What kind of dynamics converge to invariant measures faster? We believe that answering these questions will require vastly different techniques compared to standard optimization theory.

Acknowledgments

SS acknowledges support from an NSF CAREER award (number 1846088), and NSF CCF-2112665 (TILOS AI Research Institute). JZ acknowledges support from a IIIS young scholar fellowship.

References

Arjevani, Y., Carmon, Y., Duchi, J. C., Foster, D. J., Srebro, N., and Woodworth, B. Lower bounds for non-convex stochastic optimization. *arXiv preprint arXiv:1912.02365*, 2019.

Bhojanapalli, S., Wilber, K., Veit, A., Rawat, A. S., Kim, S., Menon, A., and Kumar, S. On the reproducibility of neural network predictions. February 2021.

Carmon, Y., Duchi, J. C., Hinder, O., and Sidford, A. Lower bounds for finding stationary points i. *arXiv preprint arXiv:1710.11606*, 2017.

Cheng, X., Yin, D., Bartlett, P., and Jordan, M. Stochastic gradient and langevin processes. In *International Conference on Machine Learning*, pp. 1810–1819. PMLR, 2020.

Cheung, Y. K. and Piliouras, G. Vortices instead of equilibria in minmax optimization: Chaos and butterfly effects of online learning in zero-sum games. In *Conference on Learning Theory*, pp. 807–834. PMLR, 2019.

Chizat, L. and Bach, F. On the global convergence of gradient descent for over-parameterized models using optimal transport. *arXiv preprint arXiv:1805.09545*, 2018.

Cohen, J. M., Kaur, S., Li, Y., Kolter, J. Z., and Talwalkar, A. Gradient descent on neural networks typically occurs at the edge of stability. February 2021.

Dai, Z., Yang, Z., Yang, Y., Carbonell, J., Le, Q. V., and Salakhutdinov, R. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*, 2019a.

Dai, Z., Yang, Z., Yang, Y., Carbonell, J. G., Le, Q. V., and Salakhutdinov, R. Transformer-XL: Attentive language models beyond a fixed-length context. *CoRR*, abs/1901.02860, 2019b. URL http://arxiv.org/abs/1901.02860.

Defazio, A. and Bottou, L. On the ineffectiveness of variance reduced optimization for deep learning. *arXiv preprint arXiv:1812.04529*, 2018.

Fang, C., Li, C. J., Lin, Z., and Zhang, T. Spider: Near-optimal non-convex optimization via stochastic path integrated differential estimator, 2018.

Flokas, L., Vlatakis-Gkaragkounis, E.-V., Lianeas, T., Mertikopoulos, P., and Piliouras, G. No-regret learning and mixed nash equilibria: They do not mix. *arXiv preprint arXiv:2010.09514*, 2020.

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

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Henderson, P., Islam, R., Bachman, P., Pineau, J., Precup, D., and Meger, D. Deep reinforcement learning that matters. September 2017.

Hsieh, Y.-G., Iutzeler, F., Malick, J., and Mertikopoulos, P. On the convergence of single-call stochastic extra-gradient methods. *arXiv preprint arXiv:1908.08465*, 2019.

Jacot, A., Gabriel, F., and Hongler, C. Neural tangent kernel: Convergence and generalization in neural networks. *arXiv preprint arXiv:1806.07572*, 2018.

Kingma, D. P. and Ba, J. ADAM: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kunin, D., Sagastuy-Brena, J., Gillespie, L., Margalit, E., Tanaka, H., Ganguli, S., and Yamins, D. L. Rethinking the limiting dynamics of sgd: modified loss, phase space oscillations, and anomalous diffusion. *arXiv preprint arXiv:2107.09133*, 2021.

Letcher, A. On the impossibility of global convergence in multi-loss optimization. *arXiv preprint arXiv:2005.12649*, 2020.

Li, Z., Lyu, K., and Arora, S. Reconciling modern deep learning with traditional optimization analyses: The intrinsic learning rate. *Advances in Neural Information Processing Systems*, 33, 2020.

Lobacheva, E., Kodryan, M., Chirkova, N., Malinin, A., and Vetrov, D. On the periodic behavior of neural network training with batch normalization and weight decay. *arXiv preprint arXiv:2106.15739*, 2021.

Madhyastha, P. and Jain, R. On model stability as a function of random seed. September 2019.

Mei, S., Misiakiewicz, T., and Montanari, A. Mean-field theory of two-layers neural networks: dimension-free bounds and kernel limit. In *Conference on Learning Theory*, pp. 2388–2464. PMLR, 2019.

Papadimitriou, C. and Piliouras, G. Game dynamics as the meaning of a game. *ACM SIGecom Exchanges*, 16(2): 53–63, 2019.

Reddi, S. J., Kale, S., and Kumar, S. On the convergence of ADAM and beyond. *arXiv preprint arXiv:1904.09237*, 2019.

Wu, L., Ma, C., et al. How sgd selects the global minima in over-parameterized learning: A dynamical stability perspective. *Advances in Neural Information Processing Systems*, 31:8279–8288, 2018.

Yoccoz, J.-C. An example of non convergence of birkhoff sums. https://www.college-de-france.fr/media/jean-christophe-yoccoz/UPL54030_birkhoff.pdf.

Zhang, J., He, T., Sra, S., and Jadbabaie, A. Why gradient clipping accelerates training: A theoretical justification for adaptivity. May 2019.

Zhang, J., Lin, H., Das, S., Sra, S., and Jadbabaie, A. Stochastic optimization with non-stationary noise. *arXiv preprint arXiv:2006.04429*, 2020.

A. Additional experiments details

In this section, we add some additional experiments and experimental details that supplement the results in Section 2. We showed that the observed phenomenon happens in large scale tasks. To supplement the result, we briefly comment on how smaller dataset presents different behavior by taking Cifar experiment as an example. In the end, we will discuss some experimental details on how the quantities in (1) are estimated.

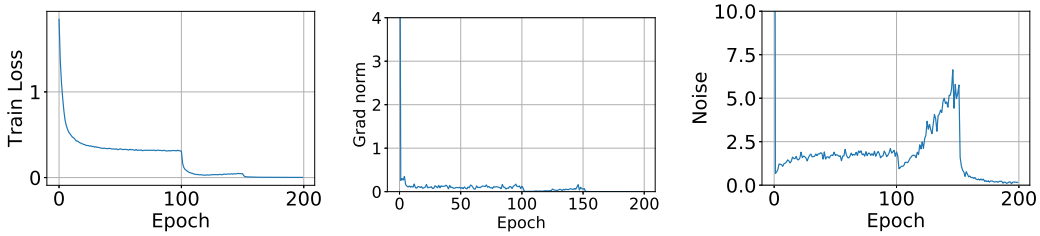

Figure 6. The estimated stats vs epoch for Cifar10 training. The learning rate starts at 0.1 and decay by a factor of 10 at epoch 100 and epoch 150.

A.1. Cifar10 Experiment

In this section, we show how noise, gradient norm and training loss evolve in Cifar10 with ResNet training. Our training procedure is based on the implementation[2]. The key result is demonstrated in Figure 6. We observe that in this case, the gradient norm indeed converges to 0. In fact, this is expected, as for cross entropy loss, the train loss could bound the gradient norm when weights are bounded.

The implications of the above observations are many. First, this separation behavior between small overfitting model on Cifar10 and larger model on ImageNet shows that the study of overparametrization and convergenece to stationary point may still be true in many cases. However, we should be careful that these analysis does not apply to larger models that do not overfit the data. Second, this shows that the SDE modeling in (Li et al., 2020; Lobacheva et al., 2021) can also be valid. It also shows that our work studies a problem of a different nature (non-zero grad norm).

A.2. Estimating the statistics

Here we provide additional details on how the values in (1) are estimated. Notice that these quantities are defined using all N data points in the entire dataset, which is too large in practice. Therefore, we use a random batch $m < N$ to estimate the quantities. For training loss, gradient norm, and noise norm, the estimation is straight-forward. For the sharpness, we follow the implementation in (Wu et al., 2018)[3] and estimate the sharpness via power iterations.

By Jensen's inequality, the estimated norms would be larger than the true value. However, the value should converge as the sampled batch size m converges to the total data number N. We show in Figure 7 and Figure 8 how these estimator values converge in practice. Based on these plots, we select the batch size to be 1.6×10^5 for ImageNet training and the token size to be 9×10^5 for the WT103 training. These sample sizes give a high enough precision level for making the observations in previous sections. Note that the estimated smoothness for the ImageNet experiment has very large variations, and hence we didn't make many comments on that plot throughout this work.

[2] https://github.com/kuangliu/pytorch-cifar
[3] https://github.com/leiwu0/sgd.stability

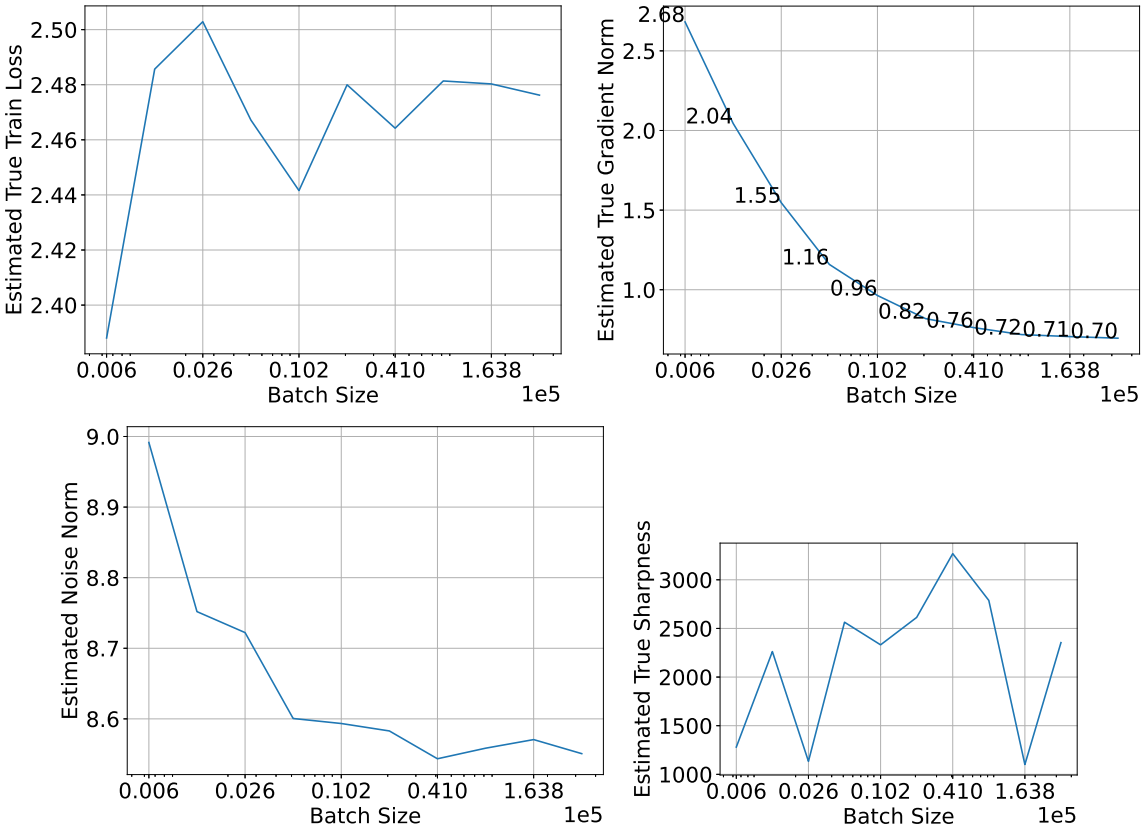

Figure 7. The estimated stats vs batch size for ImageNet training.

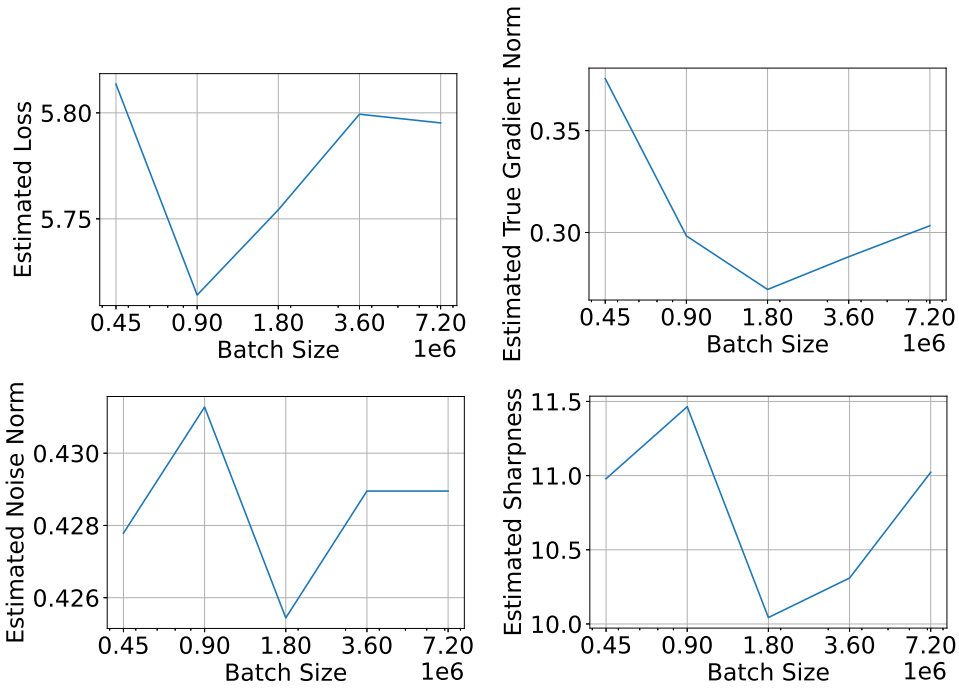

Figure 8. The estimated stats vs batch size for WT103 training.

B. Convergence of function values

One key intermediate result for the proof main theorem relies on the convergence of empirical measures when the iterates are updated by a compact continuous function F.

Theorem B.1 (convergence of function values)**.** *Consider a continuous scalar function $\phi : \mathcal{X} \to \mathbb{R}$. Assume that the update map F has the property that $\phi \circ F : \mathcal{X} \to [-M, M]$ has a bounded value for any $\theta \in \mathcal{X}$, then with probability $1 - \delta$ over the randomness of F,*

$$|\mathbb{E}_{\theta\sim\mu_n}[\phi(\theta) - \phi(F(\theta))]| = \mathcal{O}\left(\frac{\log(1/\delta)}{\sqrt{n}}\right). \tag{11}$$

Proof. The proof can be found in Appendix B. □

Proof. By the fact that $\phi \circ F : \mathcal{X} \to \mathbb{R}$ has bounded value $[-M, M]$, we can denote the subgaussian norm at θ as

$$\sigma(\theta) = \inf\{\sigma > 0 | \mathbb{P}(\|\phi(F(\theta)) - \mathbb{E}[\phi(F(\theta))]\| \geq t) \leq 2e^{-t^2/2\sigma^2}\}.$$

In fact, $\forall\theta, \sigma(\theta) \leq M < \infty$. Hence, we can further denote the upperbound on the sub-Gaussian norm as

$$\sigma = \sup_\theta \sigma(\theta).$$

Then we consider two distributions. One is the empirical distribution of a sampled trajectory,

$$\mu_n = \frac{1}{n}\sum_{t=1}^{n} \delta_{\theta_t}.$$

The other one is the pushforward distribution $\mu_k(F^{-1})$ as defined in (4). Then,

$$\begin{aligned}
\mathbb{E}_{\theta\sim\mu_n}[\phi(\theta) - \phi(F(\theta))] &= \frac{1}{n}\sum_{t=1}^{n}\phi(\theta_t) - \frac{1}{n}\sum_{t=1}^{n}\phi(F(\theta_t)) \\
&= \frac{1}{n}(\phi(\theta_0) - \phi(F(\theta^n))) + \frac{1}{n}\sum_{t=1}^{n}\phi(\theta_t) - \phi(F(\theta_{-1})) \\
&= \mathcal{O}\left(\frac{1}{n}\right) + \frac{1}{n}\sum_{t=1}^{n}\phi(\theta_t) - \phi(F(\theta_{t-1})).
\end{aligned}$$

Then the claim follows by applying Hoeffding's inequality on the second term. □

C. Proof of Theorem 5.4

Proof. Since $\mathcal{X}$ is a compact metric space, we can find a dense countable set of the family of continuous functions $C(\mathcal{X})$, denoted as $\{\phi_1, \phi_2...\}$. Since $\mathcal{X}$ is compact, we have that $\mu_k(\phi_j)$ exists for any k, j. Therefore, by the diagonal argument, there exists a subsequence $\{n_k\}_k$ such that for all $j = 1, 2, ...,$

$$\lim_{k\to\infty}\frac{1}{n_k}\sum_{l\leq n_k}\phi_j(\theta_l) = J(\phi_j).$$

Then by denseness of the set $\{\phi_1, \phi_2...\}$, we know that the above limit also exists for any $\phi \in C(\mathcal{X})$. Denote the functional as

$$J(\phi) = \lim_{k\to\infty}\frac{1}{n_k}\sum_{l\leq n_k}\phi(\theta_l). \tag{12}$$

Since J is obviously linear and bounded, there exist a unique probability measure ϕ such that $J(\phi) = \mu(\phi)$.

The invariance of μ follows by the fact that for any continuous ϕ,

$$\begin{aligned}
\lim_{k\to\infty} |\mathbb{E}_{\theta\sim\mu_k}\left[\phi(\theta) - \phi(F(\theta))\right]| &= \lim_{k\to\infty} \frac{1}{n_k}\sum_{l\le n_k}\phi(\theta_l) - \frac{1}{n_k}\sum_{l\le n_k}\phi(F(\theta_l)) \\
&= \lim_{k\to\infty} \frac{1}{n_k}\sum_{l\le n_k}\phi(\theta_l) - \frac{1}{n_k}\sum_{l\le n_k}\phi(\theta_{l+1}) \\
&\quad + \lim_{k\to\infty} \frac{1}{n_k}\sum_{l\le n_k}\phi(\theta_{l+1}) - \frac{1}{n_k}\sum_{l\le n_k}\phi(F(\theta_l)) \\
&= \lim_{k\to\infty} \frac{1}{n_k}(\phi(\theta_1) - \phi(\theta_{n_k+1})) \\
&\quad + \lim_{k\to\infty} \frac{1}{n_k}\sum_{l\le n_k}(\phi(\theta_{l+1}) - \phi(F(\theta_l))) \to 0. \qquad (13)
\end{aligned}$$

In the last line, the first term goes to zero by boundedness of function value on the compact set. The second term goes to zero by noticing that the sequence

$$M_n = \frac{1}{n}\sum_{l\le n}(\phi(\theta_{l+1}) - \phi(F(x_l))$$

is a martingale sequence. By the fact that each the induced martingale difference sequence has uniformly bounded sub-Gaussian norm, we can apply Hoeffding's inequality and know that M_n converge in probability to 0, which implies convergence in distribution.

□

D. Proof of Lemma 4.1

Proof.

1. Let $x_m = (\max_i x_i + \min_i x_i)/2$ and define $z_i = x_i - x_m$. We know $|z_i| \le c/2$. Then we have

$$\begin{aligned}
|\ell(x,y)| &= \left|z_y + x_m - \log\left(\textstyle\sum_{j=1}^d e^{z_j + x_m}\right)\right| \\
&= \left|z_y - \log\left(\textstyle\sum_{j=1}^d e^{z_j}\right)\right| \\
&\le c/2 + \log\left(de^{c/2}\right) \\
&= c + \log d.
\end{aligned}$$

2. As ℓ is differentiable, it suffices to bound its gradient norm. For any fixed $1 \le k \le d$, we have

$$\frac{\partial \ell(x,y)}{\partial x_k} = \delta_{y,k} - \frac{e^{x_k}}{\sum_{j=1}^d e^{x_j}}.$$

   Then we can bound

$$\begin{aligned}
\left\|\frac{\partial \ell(x,y)}{\partial x}\right\|_2 &= \sqrt{\textstyle\sum_{k=1}^d \left(\frac{\partial \ell(x,y)}{\partial x_k}\right)^2} \\
&\le \sqrt{1 + \frac{\sum_{k=1}^d e^{2x_k}}{\left(\sum_{j=1}^d e^{x_j}\right)^2}} \\
&\le \sqrt{2}.
\end{aligned}$$

□

E. Proof of Theorem 4.3

Proof. Denote $\rho = wc_\sigma$. Then it is easy to show that within C_w, we have $\|x_l\|_2 \le \rho^l$ for every l. We define $z_{l+1} = W_l\theta_l$ and thus $x_l = \sigma_l(z_l)$. For any mini-batch $\mathcal{B} = \{(x^i, y^i)\}_{i=1}^m$, we can bound the gradient norm of the loss.

$$\left|\frac{\partial L_\mathcal{B}}{\partial W_l}\right| = \left|\frac{1}{m}\sum_{i\in\mathcal{B}} x_l^i(\nabla_x \ell(x_L^i, y^i))^\top D_L^{(i)}\left(\prod_{s=l+1}^{L-1} W_s D_s^{(i)}\right)\right| \le c_\ell c_\sigma \rho^{L-1} \le \gamma w,$$

where we define $D_l^{(i)} = \mathrm{Diag}(\sigma_l'(z_l^i))$. By the SGD rule, we have

$$W_l^{k+1} = (1-\eta\gamma)W_l^k - \eta\frac{\partial L_\mathcal{B}}{\partial W_l}.$$

Choosing $\eta \le 1/\gamma$, if $\left\|W_l^k\right\|_{\text{op}} \le w$, we also have

$$\left\|W_l^{k+1}\right\|_{\text{op}} \le (1-\eta\gamma)w + \eta\gamma w \le w.$$

By induction on k, the iterates of SGD optimizing the above objective always lie in C_w if the stepsize satisfies $\eta \le 1/\gamma$. Then we have $\left\|x_L^i\right\|_2 \le \rho^L$ and can bound that for any k

$$|L_S(\theta_k)| \le \log d + c_\ell \left\|x_L^i\right\|_2 \le \log d + \left(\frac{\gamma}{c_\ell c_\sigma^2}\right)^{L/(L-2)}.$$

The claim then follows by applying Theorem B.1. □

F. Proof of Theorem 4.4

Proof. We first show that the coordinates of $\hat{x}_L^i$ are bounded.

$$\left|\hat{x}_L^i\right| = \frac{\left|x_{L-1}^i - \mu_{\mathcal{B},L-1}\right|}{\sqrt{\frac{1}{m}\sum_{i\in\mathcal{B}}\left(x_{L-1}^i - \mu_{\mathcal{B},L-1}\right)^2 + \epsilon}} \le \sqrt{m}.$$

As in Theorem 4.3, we show that during the training process, a_L always satisfies $|a_L| \le 2\sqrt{m}/\gamma$. Note that

$$\begin{aligned}
\left|\frac{\partial L_\mathcal{B}}{\partial a_L}\right| &= \left|\frac{1}{m}\sum_{i\in\mathcal{B}} \hat{x}_L^i \cdot \nabla_x \ell(x_L^i, y^i)\right| \\
&= \left|\frac{1}{m}\sum_{i\in\mathcal{B}}\sum_{k=1}^d (\hat{x}_L^i)_k\left(\delta_{y,k} - \frac{e^{(x_L^i)_k}}{\sum_{j=1}^d e^{(x_L^i)_j}}\right)\right| \\
&\le \max_{i\in\mathcal{B}}\left|(\hat{x}_L^i)_y - \frac{\sum_{k=1}^d (\hat{x}_L^i)_k e^{(x_L^i)_k}}{\sum_{j=1}^d e^{(x_L^i)_j}}\right| \\
&\le 2\sqrt{m}.
\end{aligned}$$

Therefore if $\left|a_L^k\right| \le 2\sqrt{m}/\gamma$, we have

$$\left|a_L^{k+1}\right| = \left|(1-\eta\gamma)a_L^k - \eta\frac{\partial L_\mathcal{B}}{\partial a_l}\right| \le (1-\eta\gamma)\cdot 2\sqrt{m}/\gamma + 2\eta\sqrt{m} \le 2\sqrt{m}/\gamma.$$

Then by induction, the above is true for every k. By Lemma 4.1, we have for every k

$$L_\mathcal{B}(\theta_k) \le 4m/\gamma + \log d.$$

Then the training loss $|L_\mathcal{B}(\theta_k)| \le 4m/\gamma + \log d$ is bounded during the training process if the stepsize satisfies $\eta \le 1/\gamma$. The theorem follows by applying Theorem B.1. □

G. Proof of Theorem 5.2

Proof. For simplicity, we denote

$$f(\theta) := L_S(\theta),$$
$$\delta := \mathbb{E}_{\theta\sim\mu}[\|\nabla f(\theta)\|_2^2] > 0.$$

By compactness of $\mathcal{X}$, we could denote the following quantities:

$$G = \sup_{\theta\in\mathcal{X}} \|g(\theta)\|_2 < \infty,$$
$$M^2 = \sup_{\theta,\zeta\in\mathcal{X}} \mathbb{E}_{z\sim\text{unif}[\theta,\zeta]}[\|\nabla^2 f(z) - \mathbb{E}_{z'\sim\text{unif}[\theta,\zeta]}[\nabla^2 f(z')]\|_{\text{op}}^2] < \infty,$$

For clarity, note that for any function $f : \mathcal{X} \to \mathbb{R}^d$,

$$\mathbb{E}_{z\sim\text{unif}[\theta,\zeta]}[f(z)] = \int_0^1 f(t\theta + (1-t)\zeta)dt.$$

Therefore, we have that for any $c \in (0, 1)$

$$\begin{aligned}
&\int_0^1 \|\nabla^2 f(ct\theta + (1-ct)\zeta) - \mathbb{E}_{z\sim\text{unif}[\theta,\zeta]}[\nabla^2 f(z)]\|_{\text{op}}^2 dt \\
=&\frac{1}{c}\int_0^c \|\nabla^2 f(t\theta + (1-t)\zeta) - \mathbb{E}_{z\sim\text{unif}[\theta,\zeta]}[\nabla^2 f(z)]\|_{\text{op}}^2 dt \\
\leq&\frac{1}{c}\mathbb{E}_{z\sim\text{unif}[\theta,\zeta]}[\|\nabla^2 f(z) - \mathbb{E}_{z'}[\nabla^2 f(z')]\|_{\text{op}}^2].
\end{aligned}$$

Therefore, by Jensen's inequality, we have

$$\int_0^1 \|\nabla^2 f(ct\theta + (1-ct)\zeta) - \mathbb{E}_{z\sim\text{unif}[\theta,\zeta]}[\nabla^2 f(z)]\|_{\text{op}} dt \leq \sqrt{\frac{M^2}{c}}. \tag{14}$$

By applying Taylor expansion twice we get the following equations,

$$\begin{aligned}
\mathbb{E}_{\theta,F}[f(F(\theta)) - f(\theta)] &= \mathbb{E}_{\theta,g}[f(\theta - \eta g(\theta)) - f(\theta)] \\
&= \mathbb{E}_{\theta,g}[-\eta \int_0^1 \langle g(\theta), \nabla f(\gamma_{\theta,g(\theta)}(\eta t))\rangle dt] \\
&= \mathbb{E}_{\theta,g}[-\eta\|\nabla f(\theta)\|_2^2 - \eta \int_0^1 \langle g(\theta) - \nabla f(\theta), \nabla f(\gamma_{\theta,g(\theta)}(\eta t))\rangle dt] \\
&- \mathbb{E}_{\theta,g}[\eta \int_0^1 \langle \nabla f(\theta), \nabla f(\gamma_{\theta,g(\theta)}(\eta t)) - \nabla f(\theta)\rangle dt] \\
&= \mathbb{E}_{\theta,g}[-\eta\|\nabla f(\theta)\|_2^2 - \eta \int_0^1 \langle g(\theta) - \nabla f(\theta), \nabla f(\gamma_{\theta,g(\theta)}(\eta t)) - \nabla f(\theta)\rangle dt] \\
&- \mathbb{E}_{\theta,g}[\eta \int_0^1 \langle \nabla f(\theta), \nabla f(\gamma_{\theta,g(\theta)}(\eta t)) - \nabla f(\theta)\rangle dt] \\
&= \mathbb{E}_{\theta,g}[-\eta\|\nabla f(\theta)\|_2^2 - \eta^2 \int_0^1 \int_0^1 \langle g(\theta), \nabla^2 f(\gamma_{\theta,g}(t\tau\eta))g(\theta)\rangle dt d\tau].
\end{aligned}$$

where $\gamma_{\theta,g(\theta)}(r) = \theta - rg(\theta)$ denotes the line segment. In the second line, we applied the fundamental theorem of calculus. In the second line, we add and subtracted same terms. In the fourth equality, we used the unbiasedness of noise. In the last line, we combined the last two terms and applied Taylor expansion again.

By invariance of the function value, we get that

$$\mathbb{E}_{\theta,g}[-\eta\|\nabla f(\theta)\|_2^2 - \eta^2 \int_0^1 \int_0^1 \langle g(\theta), \nabla^2 f(\gamma_{\theta,g}(t\tau\eta))g(\theta)\rangle dt d\tau] = 0$$
$$\implies \mathbb{E}_{\theta,g}[\|\nabla f(\theta)\|_2^2] = \mathbb{E}_{\theta,g}[\eta \int_0^1 \int_0^1 \langle g(\theta), \nabla^2 f(\gamma_{\theta,g}(t\tau\eta))g(\theta)\rangle dt d\tau]. \tag{15}$$

Therefore we have that

$$\begin{aligned}
&\mathbb{E}_{\theta,F'}[f(F'(\theta)) - f(\theta)] \\
=&\mathbb{E}_{\theta,g}[-c\eta\|\nabla f(\theta)\|_2^2 - c^2\eta^2 \int_0^1 \int_0^1 \langle g(\theta), \nabla^2 f(\gamma_{\theta,g}(t\tau\eta))g(\theta)\rangle dt d\tau] \\
\leq& c\eta\left(-\delta + cG^2\sqrt{\tfrac{M}{c}}\right),
\end{aligned}$$

where in the last line we used (14). The claim follows by setting c small enough.

□