Special Properties of Gradient Descent with Large Learning Rates

Amirkeivan Mohtashami [1] **Martin Jaggi** [1] **Sebastian Stich** [2]

Abstract

When training neural networks, it has been widely observed that a large step size is essential in stochastic gradient descent (SGD) for obtaining superior models. However, the effect of large step sizes on the success of SGD is not well understood theoretically. Several previous works have attributed this success to the stochastic noise present in SGD. However, we show through a novel set of experiments that the stochastic noise is not sufficient to explain good non-convex training, and that instead the effect of a large learning rate itself is essential for obtaining best performance. We demonstrate the same effects also in the noise-less case, i.e. for full-batch GD. We formally prove that GD with large step size—on certain non-convex function classes—follows a different trajectory than GD with a small step size, which can lead to convergence to a global minimum instead of a local one. Our settings provide a framework for future analysis which allows comparing algorithms based on behaviors that can not be observed in the traditional settings.

1 Introduction

While using variants of gradient descent (GD), namely stochastic gradient descent (SGD), has become standard for optimizing neural networks, the reason behind their success and the effect of various hyperparameters is not yet fully understood. One example is the practical observation that using a large learning rate in the initial phase of training is necessary for obtaining well performing models (Li et al., 2019). Though this behavior has been widely observed in practice, it is not fully captured by existing theoretical frameworks.

Recent investigations of SGD's success (Kleinberg et al., 2018; Pesme et al., 2021) have focused on understanding the implicit bias induced by the stochasticity. Note that the effective variance of the trajectory due to the stochasticity of the gradient is moderated by the learning rate (see Appendix G for more intuition). Therefore, using a larger learning rate amplifies the stochasticity and the implicit bias induced by it which can provide a possible explanation for the need for larger learning rates. We show that this explanation is incomplete by demonstrating cases where using stochasticity with arbitrary magnitude but with a small learning rate, can not guarantee convergence to global minimum whereas using a large learning rate can. Furthermore, we provide a practical method to increase stochasticity without changing the learning rate when training neural networks and observe that increased stochasticity can not replace the effects of large learning rates. Therefore, it is important to study how a larger learning rate affects the trajectory beyond increasing the stochasticity.

To that end, in this work we show that randomly initialized full-batch gradient descent with a high learning rate provably escapes local minima and converges to the global minimum over of a class of non-convex functions. In contrast, when using a small learning rate, GD over these functions can converge to a local minimum instead. Such difference is not observable under traditional assumptions such as smoothness. Hence, our settings also provide a framework to compare optimization methods more closely, for example in their ability to escape local minima.

We further show the positive effect of using a high learning rate to increase the chance of completely avoiding undesirable regions of the landscape such as a local minimum. Note that this behavior does not happen when using the continuous version of GD, i.e. gradient flow which corresponds to using infinitesimal step sizes. The difference remains even after adding the implicit regularization term identified in (Smith et al., 2021) in order to bring trajectories of gradient flow and gradient descent closer.

Finally, to show the relevance of our theoretical results in practice, we demonstrate evidence of an escape from local minimum (not to be confused with escaping from saddle points) when applying GD with a high learning rate on a commonly used neural network architecture. Our observations signify the importance of considering the effects of high learning rates for understanding the success of GD.

[1]EPFL, Switzerland [2]CISPA, Germany. Correspondence to: Amirkeivan Mohtashami <amirkeivan.mohtashami@epfl.ch>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

Overall, our contributions can be summarized as follows:

- Demonstrating the exclusive effects of large learning rates even in the stochastic setting both in theory and in practice, showing that they can not be reproduced by increasing stochasticity and establishing the importance of analyzing them.
- Capturing the distinct trajectories of large learning rate GD and small learning rate GD in theory on a class of functions, demonstrating the empowering effect of large learning rate to escape from local minima and providing a framework for future analysis.
- Providing experimental evidence showing that gradient descent escapes from local minima in neural network training when using a large learning rate, establishing the relevance of our theoretical results in practice.

2 Related Work

Extensive literature exists on studying the effect of stochastic noise on the convergence of GD. Several works have focused on the smoothing effect of injected noise (Chaudhari et al., 2017; Kleinberg et al., 2018; Orvieto et al., 2022; Wang et al., 2022a). In (Vardhan & Stich, 2022) it has been shown that by perturbing the parameters at every step (called perturbed GD) it is possible to converge to the minimum of a function f while receiving gradients of $f + g$, assuming certain bounds on g. Other works use different models for the stochastic noise in SGD and use it to obtain convergence bounds or to show SGD prefers certain type (usually flat) of minima (Wu et al., 2018; Xie et al., 2021). In order to better understand the effect of various hyperparameters on convergence, Jastrzebski et al. (2019); Jastrzębski et al. (2017) show the learning rate (and its ratio to batch size) plays an important role in determining the minima found by SGD. In (Pesme et al., 2021) it was shown that SGD has an implicit bias in comparison with gradient flow and its magnitude depends on the learning rate. While this shows one benefit of using large learning rates, in this work, we provide evidence that the effect of learning rate on optimization goes beyond controlling the amount of induced stochastic noise.

Another line of research has been investigating the ability of SGD to avoid saddle points. Lee et al. (2016) show that gradient descent will not converge to saddle points with high probability. Other works have also investigated the time it takes SGD to escape from a saddle point. (Daneshmand et al., 2018; Du et al., 2017; Fang et al., 2019). These results are tangential to ours since we are interested in escaping from local minima not saddle points.

Prior work also experimentally establish existence of different phases during training of a neural network. Cohen et al. (2021) show that initially Hessian eigenvalues tend to grow until reaching the convergence threshold for the used learning rate, a state they call "Edge of Stability". This growth is also reported in (Lewkowycz et al., 2020) for the maximum eigenvalue of the Neural Tangent Kernel (Jacot et al., 2018) where it has also been observed that this value decreases later in training, leading to convergence. Recent works have also investigated GD's behavior at the edge of stability for some settings (Arora et al., 2022) obtaining insights such as its effect on balancing norms of the layers of a two layer ReLU network (Chen & Bruna, 2022). In our results, GD is above the conventional stability threshold while it is escaping from a local minimum but returns to stability once the escape is finished.

In (Elkabetz & Cohen, 2021) it is conjectured that gradient descent and gradient flow have close trajectories for neural networks. However, the aforementioned observations suggest that gradient descent with a large learning rate visits a different set of points in the landscape than GD with a small learning rate. Therefore, this conjecture might not hold for general networks. The difference in trajectory is also supported by the practical observation that a large learning rate leads to a better model (Li et al., 2019).

To bridge this gap and by comparing gradient flow and gradient descent trajectories, Barrett & Dherin (2021) identify an implicit regularization term on gradient norm induced by using discrete steps. Still, this term is not enough to remove a local minimum from the landscape. Other implicit regularization terms specific to various problems have also been proposed in the literature (Ma et al., 2020; Razin & Cohen, 2020; Wang et al., 2022b). In this paper, we provide experimental evidence and showcase the benefits of using large step sizes that are unlikely to be representable through a regularization term, suggesting that considering discrete steps might be necessary to understand the success of GD.

The type of obstacles encountered during optimization of a neural network is a long-standing question in the literature. Lee et al. (2016) show that gradient descent with random initialization almost surely avoids saddle points. However it is still unclear whether local minima are encountered during training. In (Goodfellow & Vinyals, 2015) it was observed that the loss decreases monotonically over the line between the initialization and the final convergence points. However, it was later shown that this observation does not hold when using larger learning rates (Lucas et al., 2021). Swirszcz et al. (2016) also show that it is possible to create datasets which lead to a landscape containing local minima. Furthermore, better visualization of the landscape shows non-convexities can be observed on some loss functions (Li et al., 2018). For the concrete case of two layer ReLU networks, Safran & Shamir (2018) show gradient descent converges to local minima quite often without the help of over-parameterization. Also, it was shown that in the over-parameterized setting, the network is not locally convex around any differentiable global minimum and one-point strong convexity only holds in most but not

all directions (Safran et al., 2020). These observations show the importance of understanding the mechanisms of escaping local minima. We also use these observations to make assumptions that are practically justifiable.

There also exists a body of work on which properties of a minimum leads to better generalization (Dinh et al., 2017; Dziugaite & Roy, 2017; Keskar et al., 2017; Tsuzuku et al., 2020). In this work, our goal is to show the ability of gradient descent to avoid certain minima when using a high learning rate. However, the argument about whether these minima offer better or worse generalization is outside the scope of this work.

3 Main Results

In this section, we present a comprehensive summary of our key findings and their implications. To ensure clarity and prevent technical details from overshadowing the core ideas, we present simplified and less formal statements of our results here. For a more rigorous treatment, we direct readers to Section 4, where we present the formal presentation of our results.

Theoretical Proof of Escaping From Local Minima with a Large Learning Rate The need for a large learning rate in practice is commonly explained based on the intuition of escaping certain local minima[1] . However, a theoretical setting where GD escapes from a local minimum and converges to a global minimum is lacking. Such settings are necessary both for understanding success of GD and for analyzing the effectiveness of other optimizers. In this work, we introduce a class of functions where such behavior can be observed from GD. This is stated in Theorem 1 which we describe here informally and leave the formal version to Section 4.1.

Theorem 1 (Informal)**.** *There exists a class of functions C_l such that for any $f \in C_l$:*

1. *f has at least two minima $\mathbf{x}^\dagger$ and $\mathbf{x}_\star$.*
2. *With a large learning rate, GD with random initialization converges to $\mathbf{x}_\star$ almost surely but using a small learning rate there is a strictly positive probability of converging to $\mathbf{x}^\dagger$.*

Theorem 1 provides a setting where it is possible to distinguish different algorithms based on behaviors that were not observable in the traditional framework, e.g. under L-smoothness of the whole landscape. In particular, in our settings the convergence point will be to a different minimum for the larger learning rate. Furthermore, unlike the traditional settings, it is possible for GD trajectory to go through phases where the gradient norm increases temporarily which has also been observed in practice (Cohen et al., 2021). In Section 5.2 we show that when running GD on a neural network, a similar phenomenon can be observed and show that during this phase GD escapes a minimum and converges to another one. While we do not claim that neural networks are in the class of functions we introduce, GD over neural networks shares more similar behaviors with C_l than with the traditional settings. Hence, it can be expected that analyzing an optimization algorithm over C_l would allow better understanding of its behavior over neural network.

Theoretical Analysis of Avoiding Local Minima As an alternative to escaping from minima, we note that due to discrete steps in GD, it may not visit any point in an arbitrary but small part of the landscape X, such as a local minimum. However, note that there may still exist a set of starting points for which GD iterates reach a point in X. Therefore, assuming the starting point is chosen randomly, not visiting any point in X is a probabilistic event. In this work, we provide a lower bound for the probability of this event in Theorem 2 which we state here informally and postpone the formal statement to Section 4.2.

Theorem 2 (Informal)**.** *For any arbitrary part (subset) of the landscape X sufficiently far from the global minimum, let E_X be the probabilistic event that GD, when initialized randomly from a large enough set, will not iterate over any point in X. Then under certain assumptions on the landscape, $\Pr[E_X]$ can be lower bounded where the bound depends monotonically increasing on the learning rate and inversely on the size of X (as measured by Lebesgue measure). In particular, if X is finite, this probability is 1.*

Theorem 2 highlights an important difference between continuous and discrete optimization. In particular, the skipping behavior can be essential to success of the optimization but does not occur in the continuous regime. Note that as the region can be arbitrarily complex, this problem can not be ratified by adding regularization terms such as those identified in (Smith et al., 2021). This suggests other methods are needed to bridge the gap between gradient flow and GD. We demonstrate an example where both behaviors of escaping and avoiding local minima are required to converge to the desired minimum in Appendix I.

Demonstrating Effects of Large Learning Rate in Neural Networks Traditional analysis of GD's convergence depends on an upper bound on the learning rate which ensures GD gets closer to the minimum at every step. In certain settings, it can be also shown that GD with a learning rate above this upper bound gets further from the minimum at every step. As such, there is a threshold which separates

[1]We would like to note that throughout the paper, we sometimes misuse the terms "global" and "local" minimum to refer to desirable and undesirable minima respectively. For example when discussing generalization, a desirable minimum might not have the lowest objective value but enjoy properties such as flatness.

convergence and divergence of GD. However, this threshold depends on the local curvature and can be different at each point. Therefore, the learning rate can violate the upper bound for some points and satisfy it for others which leads to GD going through different phases of divergence and convergence. Indeed, this alternating phases is how GD converges in the function class introduced in Theorem 1.

While escaping local minima is an intuitive explanation, it is not evident that the alternation between diverging and converging phases also happen in neural networks. In particular, it seems in practice these alternations happen too quick so that the loss usually always has a decreasing trend. This makes it hard to verify the relevance of escaping behavior for neural network landscape. We do so in Section 5.2 by deliberately finding a point close to a minimum that GD would converge to with a small learning rate. In contrast, when applying GD with a large learning rate from this point, we can clearly observe an escape both in the trajectory and in the value of the loss.

Importance of Large Learning Rate Despite the Effects of Stochastic Noise Prior observations in practice that demonstrate the importance of using a large learning rates (Li et al., 2019) are made when applying SGD not full-batch GD. While we establish the importance of escaping from local minima in neural networks landscape in Section 5.2, this behavior can also be a result of stochastic noise since increasing the learning rate magnifies the effect of stochastic noise (see Appendix G for further intuition). As such, one possible explanation for the need of large learning rate is the magnified stochastic noise facilitating escaping from local minima. This explanation makes it questionable whether it is necessary to understand direct effects of learning rate on the trajectory or is it enough to only consider the stochastic noise.

In this work, we show that the effects of using a large learning rate goes beyond magnifying stochastic noise. To that end, we first provide an example in Section 4.3 where escaping from a local minimum and converging to the global minimum can only be achieved with a large learning rate even in presence of stochastic noise. Furthermore, we demonstrate this result in practice in Section 5.1 by decoupling the effect of stochastic noise on the trajectory and the magnitude of the learning rate when training neural networks. Our results show that the effects of large learning rates remain crucial for converging to the correct minimum even in presence of (magnified) stochastic noise.

4 Theoretical Analysis

We now state our results more formally. For our theoretical analysis, we focus on optimizing the minimization problem

$$f_\star := \min_{\mathbf{x}\in\mathbb{R}^d} f(\mathbf{x})$$

using (full-batch) gradient descent with random initialization. For completeness, we provide a pseudo code in the Appendix A, Algorithm 1.

As is done widely in the literature, we assume smoothness (as defined in Definition 1) over regions of the landscape to ensure the gradient does not change too sharply.

Definition 1 (L-smoothness)**.** *A function $f\colon \mathbb{R}^d \to \mathbb{R}$ is L-smooth if it is differentiable and there exists a constant $L > 0$ such that:*

$$\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\| \leq L\|\mathbf{x} - \mathbf{y}\|\,, \qquad \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^d\,. \quad (1)$$

Similarly, we need to ensure sharpness of certain regions, in particular around a local minima, to obtain our results. Therefore, to ensure a lower bound for sharpness in our analysis, we use one-point strong convexity assumption on these regions as defined in the following definition which also commonly appears in the literature:

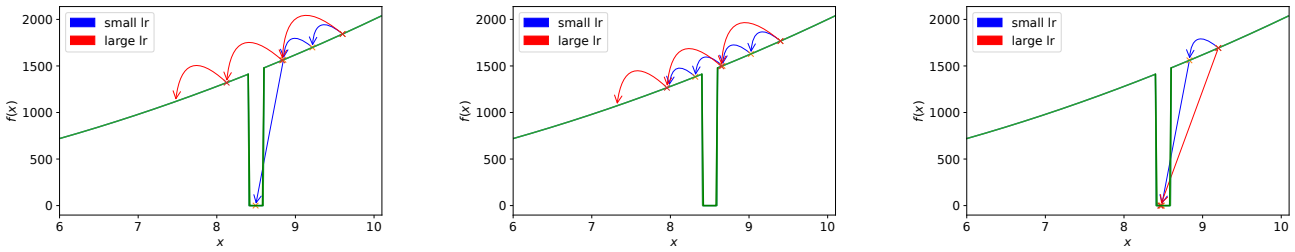
Definition 2 (μ-one-point-strongly-convex (OPSC) with respect to $\mathbf{x}_\star$ over M)**.** *A function $f\colon \mathbb{R}^d \to \mathbb{R}$ is one-point strongly convex with respect to $\mathbf{x}_\star$ if it is differentiable and there exists a constant $\mu > 0$ such that:*

$$\langle \nabla f(\mathbf{x}), \mathbf{x} - \mathbf{x}_\star \rangle \geq \mu\|\mathbf{x} - \mathbf{x}_\star\|^2\,, \qquad \forall \mathbf{x} \in M\,. \quad (2)$$

Assuming OPSC property is common in the literature. When this assumption is applied over the whole landscape, it has been shown to guarantee convergence to $\mathbf{x}_\star$ (Kleinberg et al., 2018; Lee et al., 2016; Safran et al., 2020). However, in this work we make this assumption only over regions around a local minima. Furthermore, we use this assumption to ensure sharpness which we show can result in escaping from the regions where this assumption holds rather than converging to them. Note that recent works have verified both theoretically and empirically that landscapes of neural networks satisfy this property to some extent (Kleinberg et al., 2018; Safran et al., 2020). For example, Safran et al. (2020) show that the condition is satisfied with high probability over the trajectory of perturbed gradient descent on over-parameterized two-layer ReLU networks when initialized in a neighborhood of a global minimum. We also note that there exists other variants of this definition such as quasi-strong convexity (Necoara et al., 2019) or $(1, \mu)$-(strong) quasar convexity (Hinder et al., 2020), which are similar but slightly stronger.

4.1 Escaping From Local Minima with a Large Learning Rate

We first state a lemma which is the key to proving Theorem 1. This lemma defines a set of criteria for the region M around a minimum $\mathbf{x}^\dagger$ as well as the region around M, called $P(M)$, that ensures GD escapes from M moving towards a different minimum $\mathbf{x}_\star$. To build further intuition, Figure 3 provides an example of how GD with large learning



(a) GD escapes with large LR. (b) GD escapes with both LRs. (c) GD does not escape.

Figure 1: Success of GD to avoid a region based on the magnitude of learning rate when initialized from different points. While various cases are possible, it is more likely to avoid the minimum with a higher learning rate.

rate may escape a sharp minimum. We state an overview of the lemma here and provide its formal version and the complete proof in Appendix B.

Lemma 1. *Let f be a function that is L_{global}-smooth with a global minimum $\mathbf{x}_\star$. Assume there exists a local minimum $\mathbf{x}^\dagger$ around which the following holds:*

- *f is $\mu^\dagger$-OPSC with respect to $\mathbf{x}^\dagger$ over a set M containing $\mathbf{x}^\dagger$ with diameter r.*
- *Let $P(M)$ denote the ball around $\mathbf{x}^\dagger$ with radius r_P excluding points M. Over $P(M)$, f is $L < L_{global}$-smooth and $\mu_\star$-OPSC with respect to the global minimum $\mathbf{x}_\star$ such that $\mu^\dagger > \frac{2L^2}{\mu_\star}$. The radius r_P depends on r, γ and L_{global}.*
- *Assume $\mathbf{x}^\dagger$ is sufficiently far from $\mathbf{x}_\star$, i.e. $\|\mathbf{x}^\dagger - \mathbf{x}_\star\|_2 \geq \tau$ where τ depends on $\mu_\star$, r and γ.*

Then, using a suitable learning rate $\frac{2}{\mu^\dagger} < \gamma \leq \frac{\mu_\star}{L^2}$, if GD reaches a point in M, it will escape M and reach a point with distance to $\mathbf{x}_\star$ of less than $\|\mathbf{x}^\dagger - \mathbf{x}_\star\| - r$ almost surely.

Figure 4 provides an illustration of different regions defined in the theorem's statement. Lemma 1 and Theorem 1 provide an improved theoretical setting for future analysis. For example, a second order optimizer may behave differently in this setting and converge to $\mathbf{x}^\dagger$ instead. We point out that Lemma 1 only holds if the learning rate is large enough and GD with small learning rates, i.e. $\gamma \leq \frac{2}{L_{\text{global}}}$, would instead converge to $\mathbf{x}^\dagger$. Therefore, this difference is not observable in the settings of previous works. As we experimentally verify in Sections 5.1 and 5.2 , GD's behavior in the large learning rate settings changes the convergence point in neural networks as well. As such, analyzing new optimizers in this settings can be useful to ensure they work more closely to GD for more complex scenarios such as neural networks.

We emphasize that this lemma allows for multiple local minima to exist on the landscape and only applies constraints around each local minimum. Furthermore, we point out that the lemma only ensures that GD will exit the local minima after some steps. In order to obtain convergence guarantees to the global minimum, it is necessary to assume a convergence property on the rest of the landscape as well. Indeed, this is how we build the class of functions to prove Theorem 1. We provide a more thorough discussion about our assumptions in Appendix F.

Given Lemma 1, it can be seen that if it is possible to ensure the iterations of GD get closer to the global minimum on the rest of the landscape, it is possible to ignore existence of the region M. This is because either the iterations would never cross M or if they do, they will eventually reach a point closer to the global minimum according to Lemma 1. We build the class of functions for Theorem 1 based on this observation. We now state the formal statement of Theorem 1 and leave the proof to Appendix C.

Theorem 1 (Formal)**.** *Let the set C_l be the set of all functions such as f that is L-smooth and $\mu_\star$-OPSC with respect to the global minimum $\mathbf{x}_\star$ in its landscape except on a region M containing a local minimum $\mathbf{x}^\dagger$ satisfying the conditions in Lemma 1. GD initialized randomly inside M converges to $\mathbf{x}^\dagger$ with a small learning rate $\gamma < \frac{\mu^\dagger}{L^2_{global}}$. In contrast, GD initialized randomly over any arbitrary set W with positive Lebesgue measure $\mathcal{L}(W) > 0$ will instead converge to $\mathbf{x}_\star$ with a large learning rate $\frac{2}{\mu^\dagger} < \gamma \leq \frac{\mu_\star}{L^2}$ almost surely.*

Prior results in non-convex settings only hold for fixed learning rates below a common threshold, e.g. $\frac{2}{L}$ for L-smooth functions (Bottou et al., 2016; Vaswani et al., 2019). In contrast, Thoerem 1 extends prior convergence results by proving convergence for learning rate above the traditional threshold $\frac{2}{L_{\text{global}}}$. While the extension comes at the cost of putting additional constraints on the landscape which might be unavoidable as we discuss in Appendix F, we emphasize that this still relaxes the conditions of convergence at least for functions in C_l and is an extension over prior results.

4.2 Avoiding Local Minima

The following key lemma is used to prove Theorem 2.

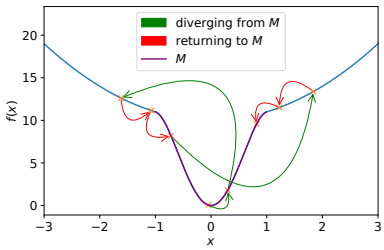
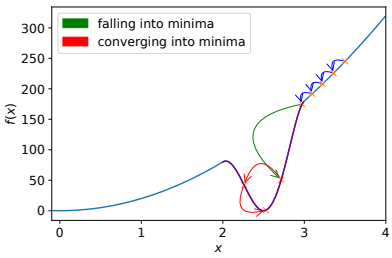
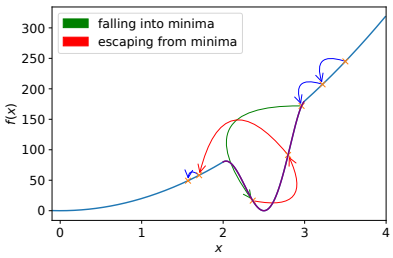


Figure 2: A case where GD keeps returning to a sharp minimum showing that a lower bound on the distance to the global minimum might be necessary to show it can be avoided.



(a) GD with small LR converges.



(b) GD with large LR escapes.

Figure 3: Different behaviors of GD based on the magnitude of learning rate in escaping or converging a sharp minima. GD with a high enough learning rate always escapes the minimum.

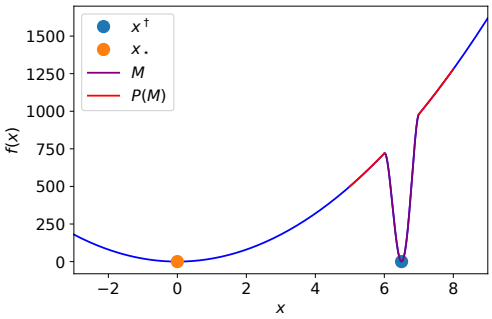


Figure 4: Illustration of different regions defined in Lemma 1.

Lemma 2. *Assume gradient descent is initialized randomly on the set W and is run with learning rate $\gamma \leq \frac{1}{2L}$. Let $X \in \mathcal{R}^d$ be an arbitrary set of points in the landscape. Assume f is L-smooth over $\mathcal{R}^d \setminus X$. Let $\mathcal{L}(S)$ denote the Lebesgue measure of any set S. The probability of encountering any points of X in the first T steps of gradient descent, i.e. $\mathbf{x}_i \in X$ for some $1 \leq i \leq T$ is at most $2^{(T+1)\cdot d} \cdot \frac{\mathcal{L}(X)}{\mathcal{L}(W)}$.*

We provide the complete proof in Appendix D. The dependence of the bound on T seems inevitable in general since the optimization might force the iterations toward certain regions, such as the region around the minimum. Similarly, the exponential dependence on d seems unavoidable in the general case. For example, if the function is $f(\mathbf{x} := (x_1, \ldots, x_d)) := \frac{L}{2}\sum_{i=1}^{d} x_i^2$ with the set X being the region around the minimum, GD converges exponentially towards X in each direction. Therefore the Lebesgue measure set of the points that converge to X within T steps increases exponentially with d (since each direction contributes with an exponential factor).

Lemma 2 does not need any assumptions about the region X . However, the dependency on d and T can be alleviated by making further assumptions. Indeed, we use one such assumption to ensure the iterations of GD move away from the undesired region X and obtain Theorem 2 which we state formally here and leave its proof to Appendix E.

Theorem 2 (Formal)**.** *Let X be an arbitrary set of points. Assume f is L-smooth and $\mu_\star$-OPSC with respect to a minima $x_\star \notin X$ over $\mathbb{R}^d \setminus X$. Define $c_X := \inf\{\|\mathbf{x}-\mathbf{x}_\star\| \mid \mathbf{x} \in X\}$ and $r_W := \sup\{\|\mathbf{x}-\mathbf{x}_\star\| \mid \mathbf{x} \in W\}$. The probability of not encountering any points of X during running gradient descent with $\gamma \leq \frac{\mu_\star}{L^2}$ is at least $1-2^d.\frac{r_W}{c_X}^{-\frac{d}{\log_2(1-\gamma\mu_\star)}} \cdot \frac{\mathcal{L}(X)}{\mathcal{L}(W)}$ when $c_X \leq r_W$ and is 1 otherwise.*

We note that the dependence of the lower bound on the learning rate is intuitive as a larger learning rate allows larger steps and makes it less probable (but not impossible) to visit a small part of the landscape as illustrated in Figure 1. Theorem 2 facilitates applying prior results when assumptions are violated on a part of landscape. In these cases, so long as the area in violation of the assumptions is small, the probability of failure can be small enough to be considered negligible in practice. As such this result can be useful to prove convergence with high probability using conditions that hold mostly but not completely everywhere on the landscape, such as one-point strong convexity on one hidden layer ReLU neural networks (Safran et al., 2020).

4.3 Importance of Large Learning Rate Despite the Effects of Stochastic Noise

We now consider the case of SGD where the stochastic noise is applied as an additive term to the gradient. The update step of SGD in this case would be:

$$\mathbf{x}_{t+1} := \mathbf{x}_t - \gamma(\nabla f(\mathbf{x}_t) + \boldsymbol{\xi}_t)\,. \tag{3}$$

For example, when the global objective $f(\mathbf{x})$ is a finite sum of n different objectives, e.g. one for each data point, we will have $\boldsymbol{\xi}_t := \nabla f_{r_t}(\mathbf{x}) - \nabla f(\mathbf{x})$ where r_t is the index of data point used in t-th step.

In this section we consider the case where the noise $\boldsymbol{\xi}_t$ is drawn from a uniform distribution $\mathrm{Uniform}(-\sigma, \sigma)$ for simplicity. We acknowledge that a uniform additive noise might not accurately approximate the noise induced by sampling from the data. Prior work on convergence results usually consider a bound on the norm of the noise and the research

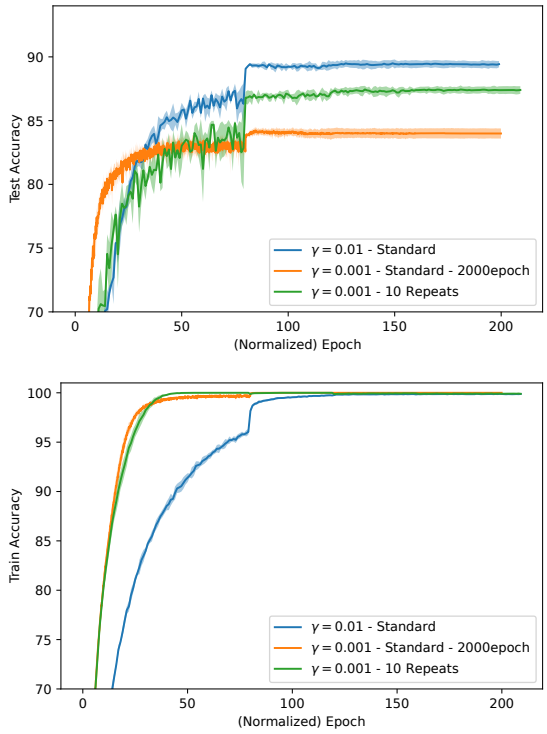


Figure 5: Comparsion between performance of SGD with different learning rates. The gap in performance between large and small learning rates, even after repeatedly using the same batch to maintain the effect of stochastic noise, suggests that learning rate has an effect on trajectory beyond controlling stochastic noise. Repeating batches is turned off at epoch 200 and 10 additional epochs are performed (green). For the experiment with 2000 epochs (orange), the plot is normalized to 200 epochs.

into the exact distribution of the noise is ongoing (). However, the intuition behind our results in this section is that when the region around a local minimum is large, a strong noise is needed to escape from this region. However, such a strong noise would also continuously escape from the region around the global minimum. We use uniform additive noise as means of demonstration but since the same intuition can be extended to many of the more accurate settings, we speculate that our results would extned to those settings as well.

The investigation into the precise distribution and format of the noise is still ongoing (Gürbüzbalaban et al., 2021; Hodgkinson & Mahoney, 2021) but we acknowledge that the use of uniform additive noise may not accurately capture the complexity of noise encountered during data sampling. However, the example in this section is based on the following intuition that we expect to remain consistent across many noise settings. When the region around a local minimum is large, a strong noise is necessary to escape this region. Nevertheless, such a strong noise would also inevitably escape a large region around the global minimum. Thus, while we utilize uniform additive noise here for demonstration, we speculate that our results can potentially be extended to the

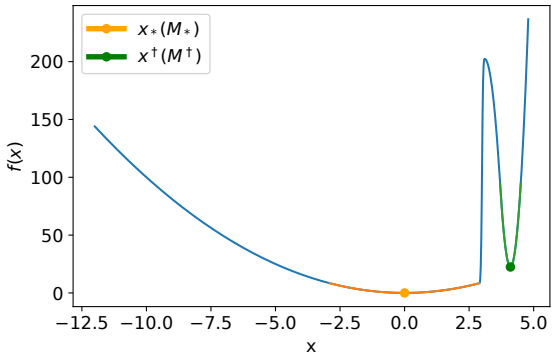


Figure 6: An example where convergence to global minimum can only be achieved by using large learning rates.

more precise scenarios such as multiplicative noise as well, given the shared intuition.

We assess the convergence point of GD on an example function plotted in Figure 6. This function contains a local minima $x^\dagger$ and a global minimum $x_\star$. Optimally, we would like to ensure convergence to the global minimum regardless of the initialization point. The following proposition shows that this happens only when using a large learning rate and is not possible when using a small learning rate regardless of the magnitude of the noise. We provide a formal description of this proposition and its proof in Appendix H.

Proposition 3. *Consider running SGD on the function plotted in Figure 6. If the learning rate is sufficiently small, starting close to $x^\dagger$, the iterates will never converge to the optimum $x_\star$ nor to a small region around it regardless of the magnitude of the noise.On the other hand, by using a large learning rate, given that the stochastic noise satisfies certain bounds, GD will succeed to converge to the optimum $x_\star$ given any starting point.*

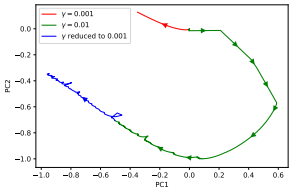
5 Experiments

We now provide practical evidence to show the effects of high learning rate also apply and are essential in optimization of neural networks. In our experiments we train a ResNet-18 (He et al., 2016) without batch normalization on CIFAR10 (Krizhevsky & Hinton, 2009) dataset.

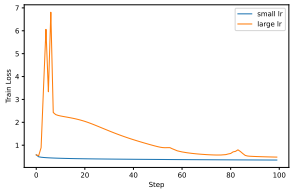
5.1 Disentangling Effects of Stochastic Noise and Learning Rate

As can be seen from (3), reducing the learning rate would also reduce the variance of the effective stochastic noise $\gamma\boldsymbol{\xi}_t$. This entanglement makes it hard to assess the effects of stochastic noise and large learning rate separately. We design the following method to maintain the level of noise when reducing the learning rate.

SGD with Repeats In order to simulate the same magnitude of noise while still using a smaller learning rate, every



(a) Trajectory of (full-batch) GD with small and large learning rate.



(b) The value of train loss when using different values of learning rate.

Figure 7: Behavior of GD for learning rates 0.001 (small) and 0.01 (large). The initialization is obtained by warm-starting the network using SGD with a small learning rate 0.001. Using a large learning rate changes the trajectory sharply and even if the learning rate is reduced again after several steps (blue line) we move toward a different direction in the landscape. This is accompanied by a sharp increase of loss at the beginning that can be attributed to GD escaping from a local sharp region in the landscape.

time a mini-batch is drawn, we use it for k steps before drawing another mini-batch. Note that when $k = 1$, we recover standard SGD. Re-using the same batch k times, allows the bias of the mini-batch to be amplified k times, so when reducing learning rate by $\frac{1}{k}$ the overall magnitude remains unchanged. This is explained in more detail in Appendix G.

We compare standard SGD with learning rate 0.01, standard SGD with learning rate 0.001, and SGD with $k = 10$ and learning rate 0.001. We apply 0.0005 weight decay, 0.9 momentum, and decay the learning rate at epochs 80, 120, and 160 by 0.1. Results without momentum are reported in Appendix N. When training with standard SGD and learning rate 0.001 we train the model for 10 times more epochs (2000 epochs) in order to obtain a fair comparison and rescale its plot to 200 epochs. In this case, learning rate decay happens at epochs 800, 1200, and 1600. Note that when running SGD with $k = 10$ repeats, we perform 10 steps using each batch while going through the whole dataset at each epoch. Therefore, the total number of steps in SGD with $k = 10$ is the same as standard SGD with 2000 epochs. Furthermore, when we have $k > 1$ we train the model for 10 more epochs at the end and use each batch only once (as in standard SGD) during the additional epochs. We perform these additional steps since training for several steps on one batch at the end of training might lead to overfitting on that batch which is not desirable for test performance. In Appendix L we also experiment with turning off repeats earlier in the training and observe no significant improvement. Finally, we ensure that the experiment with $k = 10$ uses the same initialization point and the same ordering of batches used for training with learning rate 0.01.

The results (averaged over 3 runs) are plotted in Figure 5. The first clear observation is that SGD with learning rate 0.01 leads to a much better model than SGD with learning rate 0.001. More importantly, while amplifying the noise through repeats helps lower the gap, it still has a performance below training with the large learning rate.

Explaining the positive effect of using SGD over GD on convergence has been the focus of several prior work. For example, Kleinberg et al. (2018) argue that applying SGD allows optimization to be done over a smoothed version of the function which empirically satisfies good convergence properties, particularly, one-point strong convexity toward a minimum. We argue that our observation provides a more complete overview and suggests that even after applying stochastic noise (which for example can lead to a smoothing of the function), there might be certain regions of the landscape that can only be avoided using a high learning rate. As we described above, one can consider the effect of stochastic noise to be the improvement observed when using repeats with a small learning rate in comparison with training in a standard way which still does not close the gap with training using a high learning rate. Therefore, the effects of using a high learning rate, such as those described in Section 4, are still important in determining the optimization trajectory even in stochastic setting.

5.2 Comparing Trajectories of Large and Small Learning Rates

In Section 4, we proved some of the effects of using large step sizes in avoiding or escaping certain minima in the landscape. We now demonstrate that these effects can be observed in real world applications such as training neural networks. To be able to observe the effect of large learning rate more clearly, we first warm-start the optimization by running SGD with a small learning rate 0.001 with $k = 10$ repeats (as described in Section 5.1) for 20 epochs to obtain parameters $\mathbf{x}_{\text{warm}}$. We do this to get near a minimum that would be found when using the small learning rate. Then, we start full-batch GD from $\mathbf{x}_{\text{warm}}$ with two different learning rate 0.001 (small) and 0.01 (large). We do not apply momentum when performing full-batch GD but apply 0.0005 weight decay. However we did not observe any visible difference in the results without weight decay. Similar to (Li et al., 2018), we obtain the first two principal components of the vectors $\mathbf{x}_1 - \mathbf{x}_0, \mathbf{x}_2 - \mathbf{x}_0, \ldots, \mathbf{x}_t - \mathbf{x}_0$ and plot the trajectory along these two directions in Figure 7. We can clearly observe that GD with a large learning rate changes path and moves toward a different place in the landscape. GD continues on the different path even when the learning rate is reduced back to 0.001 after 400 steps. Furthermore, looking at the loss values, we can observe a peak at the beginning of training that closely resembles what we expect to observe when GD is escaping from a local sharp region. This clearly shows that these behaviors of GD are not merely theoretical and are also relevant in real world applications.

Note that while we do not observe similar high spikes in the loss in the next steps, we conjecture that this behavior of

escaping sharp regions is constantly happening throughout training. This is also confirmed by observations in (Cohen et al., 2021) which show sharpness increases throughout training until reaching the threshold $\frac{2}{\mu}$ where μ is the learning rate. GD will then oscillate between areas sharper and smoother than this threshold. As a result of constantly avoiding sharp regions, symptoms of an escape such as a spike in loss is not observed. While we observe smaller increases in the loss, these can be due to oscillations also observed in (Cohen et al., 2021) along the highest eigenvectors which are not the same as escaping. Results in the same work show that in these cases the parameters do not move in these directions and only oscillate around the same center. Developing better visualization techniques or identifying other effects of using a high learning rate on GD's trajectory can help explain this behavior further and both of these directions are ground for future work.

6 Future Work

Developing better methods for visualization of the landscape and trajectory to obtain further insight on how GD avoids locally sharp regions is grounds for future work. Furthermore, various extensions on our theoretical results are also possible, such as showing other effects of using a large learning rate on trajectory that facilitate escaping from local minima. Finally, obtaining similar results with a relaxed set of assumptions would also be an interesting direction of research.

7 Conclusion

We argue that for understanding real world applications such as neural networks training it is crucial to analyze GD in the large learning rate regime and that the behavior stemming from using a large learning rate is irreplaceable by other mechanisms such as stochasticity. In this work, we provide ample evidence to support our argument as well as providing a settings where the special behaviors of this regime can be observed unlike the traditional settings.

In particular, to strengthen prior practical observations on the importance of large learning rate, we design a method to amplify stochastic noise without increasing the learning rate, disentangling the effects of stochastic noise and high learning rates. We observe that while a higher stochastic noise leads to a better model, it is not enough to close the gap with the model obtained using a high learning rate. Therefore, we argue that the effect of learning rate goes beyond controlling the impact of stochastic noise even in SGD. In contrast, recent works on analyzing success of SGD focus on continuous settings (Xie et al., 2021) and only take step size into account when modeling the noise (Pesme et al., 2021). We further demonstrate escaping from sharp regions in training of neural networks that only happens with large learning rates.

More importantly, we introduce a setting which is closer to practice by also imitating the effects of large learning rates widely observed in practice. Such behaviors are not observable under previous assumptions such as smoothness. Therefore, future optimization algorithms can also be evaluated in settings similar to ours ensuring they also benefit from similar escaping mechanisms which seem crucial in practice. We hope that our results will encourage future work on large step size regime.

Acknowledgment

The authors would like to express their gratitude to Maksym Andriushchenko, Hadrien Hendrikx, and Thijs Vogels for their valuable time and effort in reviewing the initial draft of this paper and providing insightful feedback. This project was supported by SNCF grant number 200020_200342.

References

Arora, S., Li, Z., and Panigrahi, A. Understanding Gradient Descent on Edge of Stability in Deep Learning, 2022.

Barrett, D. G. T. and Dherin, B. Implicit gradient regularization. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

Bogachev, V. *Measure Theory*. 2006. ISBN 978-3-540-34513-8.

Bottou, L., Curtis, F. E., and Nocedal, J. Optimization Methods for Large-Scale Machine Learning, 2016.

Chaudhari, P., Choromanska, A., Soatto, S., LeCun, Y., Baldassi, C., Borgs, C., Chayes, J. T., Sagun, L., and Zecchina, R. Entropy-sgd: Biasing gradient descent into wide valleys. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

Chen, L. and Bruna, J. On Gradient Descent Convergence beyond the Edge of Stability, 2022.

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. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

Daneshmand, H., Kohler, J. M., Lucchi, A., and Hofmann, T. Escaping saddles with stochastic gradients. In Dy, J. G. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1163–1172. PMLR, 2018.

Dinh, L., Pascanu, R., Bengio, S., and Bengio, Y. Sharp minima can generalize for deep nets. In Precup, D. and Teh, Y. W. (eds.), *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1019–1028. PMLR, 2017.

Du, S. S., Jin, C., Lee, J. D., Jordan, M. I., Singh, A., and Póczos, B. Gradient descent can take exponential time to escape saddle points. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan, S. V. N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 1067–1077, 2017.

Dziugaite, G. K. and Roy, D. M. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. In Elidan, G., Kersting, K., and Ihler, A. T. (eds.), *Proceedings of the Thirty-Third Conference on Uncertainty in Artificial Intelligence, UAI 2017, Sydney, Australia, August 11-15, 2017*. AUAI Press, 2017.

Elkabetz, O. and Cohen, N. Continuous vs. discrete optimization of deep neural networks. In Ranzato, M., Beygelzimer, A., Dauphin, Y. N., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 4947–4960, 2021.

Fang, C., Lin, Z., and Zhang, T. Sharp analysis for nonconvex SGD escaping from saddle points. In Beygelzimer, A. and Hsu, D. (eds.), *Conference on Learning Theory, COLT 2019, 25-28 June 2019, Phoenix, AZ, USA*, volume 99 of *Proceedings of Machine Learning Research*, pp. 1192–1234. PMLR, 2019.

Goodfellow, I. J. and Vinyals, O. Qualitatively characterizing neural network optimization problems. In Bengio, Y. and LeCun, Y. (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

Gürbüzbalaban, M., Simsekli, U., and Zhu, L. The heavy-tail phenomenon in SGD. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 3964–3975. PMLR, 2021.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pp. 770–778. IEEE Computer Society, 2016. doi: 10.1109/CVPR.2016.90.

Hinder, O., Sidford, A., and Sohoni, N. S. Near-optimal methods for minimizing star-convex functions and beyond. In Abernethy, J. D. and Agarwal, S. (eds.), *Conference on Learning Theory, COLT 2020, 9-12 July 2020, Virtual Event [Graz, Austria]*, volume 125 of *Proceedings of Machine Learning Research*, pp. 1894–1938. PMLR, 2020.

Hodgkinson, L. and Mahoney, M. W. Multiplicative noise and heavy tails in stochastic optimization. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings*

of Machine Learning Research, pp. 4262–4274. PMLR, 2021.

Jacot, A., Hongler, C., and Gabriel, F. Neural tangent kernel: Convergence and generalization in neural networks. In Bengio, S., Wallach, H. M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pp. 8580–8589, 2018.

Jastrzebski, S., Kenton, Z., Ballas, N., Fischer, A., Bengio, Y., and Storkey, A. J. On the relation between the sharpest directions of DNN loss and the SGD step length. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

Jastrzębski, S., Kenton, Z., Arpit, D., Ballas, N., Fischer, A., Bengio, Y., and Storkey, A. Three factors influencing minima in SGD. *ArXiv preprint*, abs/1711.04623, 2017.

Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M., and Tang, P. T. P. On large-batch training for deep learning: Generalization gap and sharp minima. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

Kleinberg, R., Li, Y., and Yuan, Y. An alternative view: When does SGD escape local minima? In Dy, J. G. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2703–2712. PMLR, 2018.

Krizhevsky, A. and Hinton, G. Learning multiple layers of features from tiny images. 2009.

Lee, J. D., Simchowitz, M., Jordan, M. I., and Recht, B. Gradient descent converges to minimizers. *ArXiv preprint*, abs/1602.04915, 2016.

Lewkowycz, A., Bahri, Y., Dyer, E., Sohl-Dickstein, J., and Gur-Ari, G. The large learning rate phase of deep learning: the catapult mechanism. *ArXiv preprint*, abs/2003.02218, 2020.

Li, H., Xu, Z., Taylor, G., Studer, C., and Goldstein, T. Visualizing the loss landscape of neural nets. In Bengio, S., Wallach, H. M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pp. 6391–6401, 2018.

Li, Y., Wei, C., and Ma, T. Towards explaining the regularization effect of initial large learning rate in training neural networks. In Wallach, H. M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E. B., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 11669–11680, 2019.

Lucas, J., Bae, J., Zhang, M. R., Fort, S., Zemel, R., and Grosse, R. Analyzing Monotonic Linear Interpolation in Neural Network Loss Landscapes. *ArXiv preprint*, abs/2104.11044, 2021.

Ma, C., Wang, K., Chi, Y., and Chen, Y. Implicit regularization in nonconvex statistical estimation: Gradient descent converges linearly for phase retrieval, matrix completion, and blind deconvolution. *Foundations of Computational Mathematics*, 20(3):451–632, 2020. ISSN 1615-3375, 1615-3383. doi: 10.1007/s10208-019-09429-9.

Necoara, I., Nesterov, Y., and Glineur, F. Linear convergence of first order methods for non-strongly convex optimization. *Mathematical Programming*, 175(1):69–107, 2019. ISSN 1436-4646.

Orvieto, A., Kersting, H., Proske, F., Bach, F. R., and Lucchi, A. Anticorrelated noise injection for improved generalization. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvári, C., Niu, G., and Sabato, S. (eds.), *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pp. 17094–17116. PMLR, 2022.

Pesme, S., Pillaud-Vivien, L., and Flammarion, N. Implicit bias of SGD for diagonal linear networks: a provable benefit of stochasticity. In Ranzato, M., Beygelzimer, A., Dauphin, Y. N., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 29218–29230, 2021.

Razin, N. and Cohen, N. Implicit regularization in deep learning may not be explainable by norms. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Safran, I. and Shamir, O. Spurious local minima are common in two-layer relu neural networks. In Dy, J. G. and Krause, A. (eds.), *Proceedings of the 35th International*

Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018, volume 80 of *Proceedings of Machine Learning Research*, pp. 4430–4438. PMLR, 2018.

Safran, I., Yehudai, G., and Shamir, O. The effects of mild over-parameterization on the optimization landscape of shallow ReLU neural networks. *ArXiv preprint*, abs/2006.01005, 2020.

Smith, S. L., Dherin, B., Barrett, D. G. T., and De, S. On the origin of implicit regularization in stochastic gradient descent. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

Swirszcz, G., Czarnecki, W. M., and Pascanu, R. Local minima in training of neural networks. *ArXiv preprint*, abs/1611.06310, 2016.

Tsuzuku, Y., Sato, I., and Sugiyama, M. Normalized flat minima: Exploring scale invariant definition of flat minima for neural networks using pac-bayesian analysis. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 9636–9647. PMLR, 2020.

Vardhan, H. and Stich, S. U. Tackling benign nonconvexity with smoothing and stochastic gradients. *ArXiv preprint*, abs/2202.09052, 2022.

Vaswani, S., Bach, F. R., and Schmidt, M. Fast and faster convergence of SGD for over-parameterized models and an accelerated perceptron. In Chaudhuri, K. and Sugiyama, M. (eds.), *The 22nd International Conference on Artificial Intelligence and Statistics, AISTATS 2019, 16-18 April 2019, Naha, Okinawa, Japan*, volume 89 of *Proceedings of Machine Learning Research*, pp. 1195–1204. PMLR, 2019.

Wang, X., Oh, S., and Rhee, C. Eliminating sharp minima from SGD with truncated heavy-tailed noise. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022a.

Wang, Y., Chen, M., Zhao, T., and Tao, M. Large learning rate tames homogeneity: Convergence and balancing effect. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022b.

Wu, L., Ma, C., and E, W. How SGD selects the global minima in over-parameterized learning: A dynamical stability perspective. In Bengio, S., Wallach, H. M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pp. 8289–8298, 2018.

Xie, Z., Sato, I., and Sugiyama, M. A diffusion theory for deep learning dynamics: Stochastic gradient descent exponentially favors flat minima. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

A Gradient Descent with Random Initialization

Algorithm 1 Gradient Descent with Random Initialization

1: Pick $\mathbf{x}_0$ randomly from the set W.
2: **for** $t = 1 \dots T$ **do**
3: $\quad \mathbf{x}_t \leftarrow \mathbf{x}_{t-1} - \gamma \nabla f(\mathbf{x}_{t-1})$
4: **end for**

B Proof of Lemma 1

Lemma. *Let f be a function that is L_{global}-smooth and consider running GD with learning rate γ randomly initialized over a set W with $\mathcal{L}(W) > 0$. Let M be a set with diameter r, containing a local minimum $\mathbf{x}^\dagger$ and define $P(M) := \{\mathbf{x} \notin M \mid \|\mathbf{x} - \mathbf{x}^\dagger\|_2 \leq r\sqrt{\gamma^2 L_{global}^2 - 3}\}$ to be the set surrounding M. Assume f is $L < L_{global}$-smooth and $\mu_\star$-OPSC over $P(M)$ with respect to a (global) minimum $\mathbf{x}_\star$ that is sufficiently far from M, formally, $\|\mathbf{x}_\star - \mathbf{x}^\dagger\|_2 \geq r \cdot \frac{1+\sqrt{(\gamma^2 L_{global}^2 - 3)(1-\gamma\mu_\star)}}{1-\sqrt{1-\gamma\mu_\star}}$. Finally, assume f is $\mu^\dagger$-OPSC with respect to $x^\dagger$ over M where $\mu^\dagger > \frac{2L^2}{\mu_\star}$. Then, using a suitable learning rate $\frac{2}{\mu^\dagger} < \gamma \leq \frac{\mu_\star}{L^2}$, if GD reaches a point M, it will escape M and reach a point with distance to $x_\star$ of less than $\|\mathbf{x}^\dagger - \mathbf{x}_\star\| - r$ almost surely.*

Proof. Let t be the smallest step where $\mathbf{x}_t \in M$. Using Corollary 1, $\mathbf{x}_t \neq \mathbf{x}^\dagger$ almost surely. Therefore $\|\mathbf{x}_t - \mathbf{x}^\dagger\| > 0$. Since $\gamma > \frac{2}{\mu^\dagger}$, we have

$$\begin{aligned}
\|\mathbf{x}_{t+1} - \mathbf{x}^\dagger\|_2^2 &= \|\mathbf{x}_t - \gamma\nabla f(\mathbf{x}_t) - \mathbf{x}^\dagger\|_2^2 \\
&= \|\mathbf{x}_t - \mathbf{x}^\dagger\|_2^2 - 2\gamma\left\langle \nabla f(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x}^\dagger \right\rangle + \gamma^2\|\nabla f(\mathbf{x}_t)\|_2^2 \\
&\geq \|\mathbf{x}_t - \mathbf{x}^\dagger\|_2^2 - 2\gamma\|\nabla f(\mathbf{x}_t)\|_2\|\mathbf{x}_t - \mathbf{x}^\dagger\|_2 + \gamma^2\|\nabla f(\mathbf{x}_t)\|_2^2 \\
&= \|\mathbf{x}_t - \mathbf{x}^\dagger\|_2^2 + \gamma\|\nabla f(\mathbf{x}_t)\|_2(\gamma\|\nabla f(\mathbf{x}_t)\|_2 - 2\|\mathbf{x}_t - \mathbf{x}^\dagger\|_2) \\
&\geq \|\mathbf{x}_t - \mathbf{x}^\dagger\|_2^2 + \gamma\|\nabla f(\mathbf{x}_t)\|_2(\gamma\mu^\dagger\|\mathbf{x}_t - \mathbf{x}^\dagger\|_2 - 2\|\mathbf{x}_t - \mathbf{x}^\dagger\|_2) \\
&\geq \|\mathbf{x}_t - \mathbf{x}^\dagger\|_2^2 + \gamma\|\nabla f(\mathbf{x}_t)\|_2\|\mathbf{x}_t - \mathbf{x}^\dagger\|_2(\gamma\mu^\dagger - 2) \\
&\overset{(A)}{\geq} \|\mathbf{x}_t - \mathbf{x}^\dagger\|_2^2 + \gamma\mu^\dagger\|\mathbf{x}_t - \mathbf{x}^\dagger\|_2^2(\gamma\mu^\dagger - 2) \\
&\overset{(B)}{\geq} (2\gamma\mu^\dagger - 3)\|\mathbf{x}_t - \mathbf{x}^\dagger\|_2^2 ,
\end{aligned}$$

where (A) holds because $\gamma\mu^\dagger - 2 > 0$ and (B) is obtained by using the lower bound $\gamma\mu^\dagger > 2$. Therefore, the distance to $\mathbf{x}^\dagger$ grows at least with the rate $2\gamma\mu^\dagger - 3 > 1$. Hence, GD is guaranteed to reach a point $\mathbf{x}_{t+k}$ outside M for some $k > 0$. If $\|\mathbf{x}_{t+k} - \mathbf{x}_\star\| \leq \|\mathbf{x}^\dagger - \mathbf{x}_\star\| - r$, we are done. Otherwise, we verify that this condition holds for $\mathbf{x}_{t+k+1}$.

First note that

$$\begin{aligned}
\|\mathbf{x}_{t+k} - \mathbf{x}^\dagger\|_2^2 &= \|\mathbf{x}_{t+k-1} - \gamma\nabla f(\mathbf{x}_{t+k-1}) - \mathbf{x}^\dagger\|_2^2 \\
&= \|\mathbf{x}_{t+k-1} - \mathbf{x}^\dagger\|_2^2 - 2\gamma\left\langle \nabla f(\mathbf{x}_{t+k-1}), \mathbf{x}_{t+k-1} - \mathbf{x}^\dagger \right\rangle + \gamma^2\|\nabla f(\mathbf{x}_{t+k-1})\|_2^2 \\
&\leq \|\mathbf{x}_{t+k-1} - \mathbf{x}^\dagger\|_2^2(1 - 2\gamma\mu^\dagger + \gamma^2 L_{\text{global}}^2) \\
&\leq r^2(1 - 2\gamma\mu^\dagger + \gamma^2 L_{\text{global}}^2) \\
&\leq r^2(\gamma^2 L_{\text{global}}^2 - 3) ,
\end{aligned}$$

where the last inequality holds because $\gamma > \frac{2}{\mu^\dagger}$.

$$\begin{aligned}\|\mathbf{x}_{t+k+1} - \mathbf{x}_\star\|_2^2 &= \|\mathbf{x}_{t+k} - \gamma\nabla f(\mathbf{x}_{t+k}) - \mathbf{x}_\star\|_2^2 \\ &= \|\mathbf{x}_{t+k} - \mathbf{x}_\star\|_2^2 - 2\gamma\left\langle\nabla f(\mathbf{x}_{t+k}), \mathbf{x}_{t+k} - \mathbf{x}_\star\right\rangle + \gamma^2\|\nabla f(\mathbf{x}_{t+k})\|_2^2 \\ &\leq \|\mathbf{x}_{t+k} - \mathbf{x}_\star\|_2^2(1 - 2\gamma\mu_\star + \gamma^2 L^2) \\ &\leq \|\mathbf{x}_{t+k} - \mathbf{x}_\star\|_2^2(1 - \gamma\mu_\star)\,,\end{aligned}$$

where in the last inequality we used $\gamma \leq \frac{\mu_\star}{L^2}$. We can now write

$$\begin{aligned}\|\mathbf{x}_{t+k+1} - \mathbf{x}_\star\|_2 &\leq (\|\mathbf{x}^\dagger - \mathbf{x}_\star\|_2 + \|\mathbf{x}_{t+k} - \mathbf{x}^\dagger\|_2)\sqrt{1 - \gamma\mu_\star} \\ &\leq \left(\|\mathbf{x}^\dagger - \mathbf{x}_\star\|_2 + r\sqrt{\gamma^2 L_{\text{global}}^2 - 3}\right)\sqrt{1 - \gamma\mu_\star}\,.\end{aligned}$$

Given the lower bound on distance $\|\mathbf{x}^\dagger - \mathbf{x}_\star\|$, we have

$$r(\sqrt{(\gamma^2 L_{\text{global}}^2 - 3)(1 - \gamma\mu_\star)} + 1) \leq \|\mathbf{x}^\dagger - \mathbf{x}_\star\|_2(1 - \sqrt{1 - \gamma\mu_\star})\,.$$

This yields

$$\|\mathbf{x}_{t+k+1} - \mathbf{x}_\star\|_2 \leq \|\mathbf{x}^\dagger - \mathbf{x}_\star\|_2 - r\,,$$

completing the proof. □

C Proof of Theorem 1

Theorem. *Consider any function f that is L-smooth and $\mu_\star$-OPSC with respect to some minimum $\mathbf{x}_\star$ in its landscape except on a region M containing a local minimum $\mathbf{x}^\dagger$ satisfying the conditions in Lemma 1. GD initialized randomly inside M converges to $\mathbf{x}^\dagger$ with a small learning rate $\gamma < \frac{\mu^\dagger}{L_{global}^2}$ but will instead converge to $\mathbf{x}_\star$ with a large learning rate $\frac{2}{\mu^\dagger} < \gamma \leq \frac{\mu_\star}{L^2}$ almost surely.*

Proof. When GD is initialized inside M and the learning rate is small satisfying $\gamma < \frac{\mu^\dagger}{L_{\text{global}}^2}$, since we have $\mu^\dagger$-OPSC and L_{global}-smoothness inside M, the iterates will satisfy

$$\begin{aligned}\|\mathbf{x}_{t+1} - \mathbf{x}^\dagger\|_2^2 &= \|\mathbf{x}_t - \gamma\nabla f(\mathbf{x}_t) - \mathbf{x}^\dagger\|_2^2 \\ &= \|\mathbf{x}_t - \mathbf{x}^\dagger\|_2^2 - 2\gamma\left\langle\nabla f(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x}^\dagger\right\rangle + \gamma^2\|\nabla f(\mathbf{x}_t)\|_2^2 \\ &\leq (1 - 2\gamma\mu^\dagger + \gamma^2 L_{\text{global}}^2)\|\mathbf{x}_t - \mathbf{x}^\dagger\|_2^2 \\ &\leq (1 - \gamma(2\mu^\dagger - \gamma L_{\text{global}}^2))\|\mathbf{x}_t - \mathbf{x}^\dagger\|_2^2 \\ &\leq (1 - \gamma\mu^\dagger)\|\mathbf{x}_t - \mathbf{x}^\dagger\|_2^2\,,\end{aligned}$$

Therefore, GD will converge to $\mathbf{x}^\dagger$. Let us now consider the case when GD is instead applied using a large learning rate satisfying $\frac{2}{\mu^\dagger} < \gamma \leq \frac{\mu_\star}{L^2}$. Furthermore, we allow initialization over any arbitrary set (instead of only subsets of M) as long as they satisfy $\mathcal{L}(W) > 0$. In this case, for each iterate, if $\mathbf{x}_t \notin M$, similar to above we have

$$\|\mathbf{x}_{t+1} - \mathbf{x}_\star\|_2^2 \leq (1 - \gamma\mu_\star)\|\mathbf{x}_t - \mathbf{x}_\star\|_2^2\,.$$

If $\mathbf{x}_t \in M$, Lemma 1 shows that there exists $k > 0$ such that $\mathbf{x}_{t+k} \notin M$ and $\|\mathbf{x}_{t+k} - \mathbf{x}_\star\|_2^2$ is less than the distance of any point in M to $\mathbf{x}_\star$. Since $\mathbf{x}_{t+k} \notin M$ the above argument holds and the distance to $\mathbf{x}_\star$ decreases. Therefore, for any $t' > t + k$ this distance $\|\mathbf{x}_{t'} - \mathbf{x}_\star\|_2^2$ remains less than the distance of any point in M to $\mathbf{x}_\star$. This guarantees that $\mathbf{x}_{t'} \notin M$. Hence the distance to $\mathbf{x}_\star$ keeps decreasing which means GD will converge to $\mathbf{x}_\star$. □

D Proof of Lemma 2

Lemma. *Assume gradient descent is initialized randomly on the set W and is run with learning rate $\gamma \leq \frac{1}{2L}$. Let $X \in \mathcal{R}^d$ be the set of points that should not be encountered by GD and assume f is L-smooth over $\mathcal{R}^d \setminus X$. Let $\mathcal{L}(S)$ denote the Lebesgue measure of any set S. The probability of encountering any points of X in the first T steps of gradient descent, i.e. $\mathbf{x}_i \in X$ for some $1 \leq i \leq T$ is at most $2^{(T+1)\cdot d} \cdot \frac{\mathcal{L}(X)}{\mathcal{L}(W)}$.*

Proof. Define $g(\mathbf{x}) := \mathbf{x} - \gamma \nabla f(\mathbf{x})$. When $\gamma < \frac{1}{L}$, since f is L-smooth over $\mathcal{D}_g := \mathcal{R}^d \setminus X$, results of Lee et al. (2016) show $g(\mathbf{x})$ is a diffeomorphism over $\mathcal{D}_g$. As a result, the function g^T obtained by applying g for T times is also a diffeomorphism over the set

$$\mathcal{D}_{g^T} := \mathcal{R}^d \setminus (X \cup g^{-1}(X) \cup \ldots \cup (g^{(T-1)})^{-1}(X)) \,.$$

According to the change of variable formula for Lebesgue measure (for example, see (Bogachev, 2006, Eq. (3.7.2))), for any measurable set $Y \subset \mathcal{D}_{g^T}$

$$\mathcal{L}(g^T(Y)) = \int_Y |\det \nabla g^T(\mathbf{y})| \mathbf{dy} \,.$$

Since $\gamma \leq \frac{1}{2L}$, we have for any $\mathbf{y} \in \mathcal{D}_g$,

$$|\det \nabla g(\mathbf{y})| = |\det(I - \gamma \nabla^2 f(\mathbf{y}))| \geq 2^{-d} \,.$$

The last equality holds because smoothness ensures all eigenvalues of $\nabla^2 f(\mathbf{x})$ are at most L. So for any eigenvalue λ_i, $1 - \gamma\lambda_i \geq \frac{1}{2}$. Using this result, we also can obtain $|\det \nabla g^T(\mathbf{y})| \geq 2^{-Td}$ for any $\mathbf{y} \in \mathcal{D}_{g^T}$. Thus, we have

$$\mathcal{L}(g^T(Y)) \geq 2^{-Td} \mathcal{L}(Y) \,,$$

which means,

$$\mathcal{L}((g^T)^{-1}(X) \cap \mathcal{D}_{g^T}) \leq 2^{Td} \mathcal{L}(X) \,.$$

Note that while the above argument works for $T \geq 1$, the former inequality also trivially holds for $T = 0$. Hence

$$\begin{aligned}
\mathcal{L}(\cup_{t=0}^T ((g^t)^{-1}(X) \cap W)) &\leq \mathcal{L}(\cup_{t=0}^T (g^t)^{-1}(X)) \\
&= \mathcal{L}(\cup_{t=0}^T ((g^t)^{-1}(X) \cap \mathcal{D}_{g^t})) \\
&\leq \sum_{t=0}^T \mathcal{L}((g^t)^{-1}(X) \cap \mathcal{D}_{g^t}) \\
&\leq \sum_{t=0}^T 2^{td} \mathcal{L}(X) \\
&\leq \mathcal{L}(X) (\sum_{t=0}^T 2^t)^d \\
&\leq 2^{(T+1)d} \mathcal{L}(X) \,,
\end{aligned}$$

where in the last inequality we used $2^0 + 2^1 + \ldots 2^T < 2^{T+1}$. The theorem follows directly from this result. □

The following corollary directly follows from Lemma 2. We use this corollary in proving Lemma 1 to avoid cases where we directly land on a minimum with $\nabla f(\mathbf{x}) = 0$.

Corollary 1. *Let f be L smooth. If X is a set with $\mathcal{L}(X) = 0$, for example when it is a finite set of points, the probability of encountering X throughout training with gradient descent using $\gamma \leq \frac{1}{2L}$ and random initialization over a set W with $\mathcal{L}(W) > 0$ is 0.*

E Proof of Theorem 2

Theorem. *Let f be L-smooth and $\mu_\star$-OPSC with respect to a minima $x_\star$ over $\mathbb{R}^d \setminus X$. Define $c_X := \inf\{\|\mathbf{x}-\mathbf{x}_\star\| \mid \mathbf{x} \in X\}$ and $r_W := \sup\{\|\mathbf{x}-\mathbf{x}_\star\| \mid \mathbf{x} \in W\}$. The probability of encountering any points of X during running gradient descent with $\gamma \leq \frac{\mu_\star}{L^2}$ is upper bounded by $2^d \cdot \frac{r_W}{c_X}^{-\frac{d}{\log_2(1-\gamma\mu_\star)}} \cdot \frac{\mathcal{L}(X)}{\mathcal{L}(W)}$ when $c_X \leq r_W$ and is zero otherwise.*

Proof. Due to $\mu_\star$-OPSC property of the landscape over $\mathbb{R}^d \setminus X$, as long as $\mathbf{x}_t \notin X$, we have

$$\begin{aligned}
\|\mathbf{x}_{t+1}-\mathbf{x}_\star\|_2^2 &= \|\mathbf{x}_t - \gamma\nabla f(\mathbf{x}_t) - \mathbf{x}_\star\|_2^2 \\
&= \|\mathbf{x}_t-\mathbf{x}_\star\|_2^2 - 2\gamma\langle\nabla f(\mathbf{x}_t), \mathbf{x}_t-\mathbf{x}_\star\rangle + \gamma^2\|\nabla f(\mathbf{x}_t)\|_2^2 \\
&\leq (1-2\gamma\mu_\star+\gamma^2L^2)\|\mathbf{x}_t-\mathbf{x}_\star*\|_2^2 \\
&\leq (1-\gamma(2\mu_\star-\gamma L^2))\|\mathbf{x}_t-\mathbf{x}_\star\|_2^2 \\
&\leq (1-\gamma\mu_\star)\|\mathbf{x}_t-\mathbf{x}_\star\|_2^2\,,
\end{aligned}$$

where the last inequality holds because $\gamma \leq \frac{\mu_\star}{L^2}$. Hence, if $\mathbf{x}_t \notin X$ for $t \in [T-1]$, we have

$$\begin{aligned}
\|\mathbf{x}_T-\mathbf{x}_\star\|_2^2 &\leq (1-\gamma\mu_\star)^T\|\mathbf{x}_t-\mathbf{x}_\star\|_2^2 \\
&\leq (1-\gamma\mu_\star)^T r_W\,.
\end{aligned}$$

Let $T_0 := \frac{\log_2\frac{c_X}{r_W}}{\log_2(1-\gamma\mu_\star)}$. For $T > T_0$, we have

$$\|\mathbf{x}_T-\mathbf{x}_\star\|_2^2 \leq (1-\gamma\mu_\star)c_X < c_X\,,$$

which means $\mathbf{x}_T \notin X$. Therefore, if GD does not reach any point in X in the first T_0 steps, it will not reach any point in X afterwards neither. Therefore, the probability of encountering any point in X is the same as the probability of encountering such points in the first T_0 steps. According to Lemma 2, this value is bounded as:

$$\begin{aligned}
2^{(T_0+1)d}\cdot\frac{\mathcal{L}(X)}{\mathcal{L}(W)} &= 2^d\cdot\frac{c_X}{r_W}^{\frac{d}{\log_2(1-\gamma\mu_\star)}}\cdot\frac{\mathcal{L}(X)}{\mathcal{L}(W)} \\
&= 2^d\cdot\frac{r_W}{c_X}^{-\frac{d}{\log_2(1-\gamma\mu_\star)}}\cdot\frac{\mathcal{L}(X)}{\mathcal{L}(W)}\,.
\end{aligned}$$

□

F Discussion about Lemma 1's Assumptions

OPSC condition inside M We assume f is OPSC with respect to a different minima inside M in order to ensure GD will escape from M. However, other conditions might also ensure the same effect. The theorem would also hold with those assumptions. Note that the sharpness of M with respect to the rest of landscape is reflected through the lower bound on $\mu^\dagger$ and is necessary so we can set the learning rate in the given range. As an example, when f is a quadratic function everywhere except M (such as in Figure 4), we have $\mu_\star = L$ and the lower bound becomes $\mu^\dagger > 2L$.

OPSC condition around M We combine this assumption with the assumption on M being sufficiently far from the global minimum in order to ensure that once GD escapes from a local minima, the gradient points strongly towards $\mathbf{x}_\star$. This ensures that GD will reach a point closer to the global minimum after escaping M. While the OPSC assumption is not necessary to show GD will never converge to M and may be replaceable by alternatives, an assumption on the distance to $\mathbf{x}_\star$ might be necessary to show GD will not return to M. For example, consider a quadratic function where the region around minimum is replaced by a sharper quadratic function, as plotted in Figure 2. In this case, GD with a high learning rate will keep returning to M though it will never converge to it. As alternatives to OPSC assumption on $P(M)$, one can assume GD converges in at most a fixed number of steps (which Corollary 1 states can not be to any point in M almost surely) or assume directly that the gradient points strongly away from M. Finding similar assumptions is grounds for future work.

G Effect of Learning Rate on Stochasticity

Let us focus on the case where $f(\mathbf{x})$ is the finite-sum $\frac{1}{N}\sum_{i=1}^{N} f_i(\mathbf{x})$. Then, using a large learning rate $k\gamma$, the iterates would satisfy

$$\frac{\mathbf{x}_{t+1}-\mathbf{x}_t}{\gamma} = -k\nabla f_{r_t}(\mathbf{x}_t) = -k\nabla f(\mathbf{x}_t) - k(\nabla f_{r_t}(\mathbf{x}_t) - \nabla f(\mathbf{x}_t))\,. \tag{4}$$

Let us assume that the deviation direction of each data point from the true gradient changes very slowly, i.e. the functions $f_i - f$ are extremely smooth. Then, using a smaller learning rate we instead have

$$\begin{aligned}
\frac{\mathbf{x}_{t+k}-\mathbf{x}_t}{\gamma} &= -\sum_{i=0}^{k-1} \nabla f_{r_{t+i}}(\mathbf{x}_{t+i}) \\
&= -\sum_{i=0}^{k-1} \nabla f(\mathbf{x}_{t+i}) - \sum_{i=0}^{k-1} (\nabla f_{r_{t+i}}(\mathbf{x}_{t+i}) - \nabla f(\mathbf{x}_{t+i})) \\
&\approx -\sum_{i=0}^{k-1} \nabla f(\mathbf{x}_{t+i}) - \sum_{i=0}^{k-1} (\nabla f_{r_{t+i}}(\mathbf{x}_t) - \nabla f(\mathbf{x}_t))\,.
\end{aligned}$$

To compare the strength of noise in each case we can for example compare the variance of the right hand side. Let $\sigma^2 := \frac{1}{N}\sum_{i=1}^{N}\|\nabla f_i(\mathbf{x}_t) - \nabla f(\mathbf{x}_t)\|_2^2$. Then, the variance when using the large learning rate would be $k^2\sigma^2$. When using a smaller learning rate and sampling at each step to obtain r_t the variance is instead $k\sigma^2$ and is therefore reduced. However, using SGD with repeats, i.e. using $r_{t+i} = r_t$ for $1 \le i \le k-1$, we recover the same variance as the large learning rate. Therefore, using SGD with repeats, allows maintaining the same level of noise while still using a smaller learning rate.

H Proof of Proposition 3

We first state the following key theorem which describes criteria ensuring escaping from or staying around a minimum:

Theorem 4. *Let M be a ball with radius r centered at a minimum $\mathbf{x}^\dagger$ and assume f is L_M-smooth over M and $\mu^\dagger$-OPSC with respect to $\mathbf{x}^\dagger$. Consider running SGD with a small learning rate $\gamma \le \frac{\mu^\dagger}{2L_M^2}$. Assume that when running SGD such that the oracle noise is bounded as*

$$\mathbb{E}\|\mathbf{g}_t - \nabla f(\mathbf{x}_t)\|_2^2 \le \sigma^2\,.$$

Furthermore assume that for some $c \le 1$ we have $\Pr\left[\|\mathbf{g}_t - \nabla f(\mathbf{x}_t)\|_2^2 > c^2\sigma^2\right] > 0$ for all $\mathbf{x}_t \in M$. Assume SGD reaches a point in M. If $\sigma^2 \le \frac{\mu^\dagger}{\gamma} r^2$ it will remain in M with probability at least $\frac{2}{2-\gamma\mu^\dagger}$. On the other hand, if $c^2\sigma^2 \ge \frac{2L_M}{\gamma} r^2 + \epsilon$ for some $\epsilon > 0$ it will escape M almost surely.

Proof. Let t denote the parameters at an iteration such that $\mathbf{x}_t \in M$. We have

$$\begin{aligned}
\mathbb{E}\|\mathbf{x}_{t+1} - \mathbf{x}^\dagger\|^2 &= \|\mathbf{x}_t - \mathbf{x}^\dagger\|^2 - 2\gamma\left\langle \mathbb{E}\mathbf{g}_t, \mathbf{x}_t - \mathbf{x}^\dagger\right\rangle + \gamma^2\mathbb{E}\|\mathbf{g}_t\|^2 \\
&= \|\mathbf{x}_t - \mathbf{x}^\dagger\|^2 - 2\gamma\left\langle \nabla f(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x}^\dagger\right\rangle + \gamma^2\|\nabla f(\mathbf{x}_t)\|^2 + \gamma^2\sigma^2 \\
&\le \|\mathbf{x}_t - \mathbf{x}^\dagger\|^2(1 - 2\gamma\mu^\dagger + \gamma^2 L_M^2) + \gamma r^2\mu^\dagger \\
&\le r^2(1 - \gamma\mu^\dagger + \gamma^2 L_M^2) \\
&\le r^2(1 - \frac{\gamma\mu^\dagger}{2})\,.
\end{aligned}$$

Thus, using Markov inequality we have

$$\Pr\left[\|\mathbf{x}_{t+1} - \mathbf{x}^\dagger\|^2 > r^2\right] \le \frac{1}{1 - \frac{\gamma\mu^\dagger}{2}}\,,$$

which shows the claim. On the other hand, let $p := \Pr\left[\|\mathbf{g}_t - \nabla f(\mathbf{x}_t)\|_2^2 > c^2\sigma^2\right]$. Then with probability at least $p > 0$ we have

$$\begin{aligned}
\|\mathbf{x}_{t+1} - \mathbf{x}^\dagger\|_2^2 &= \|\mathbf{x}_t - \gamma\nabla f(\mathbf{x}_t) - \mathbf{x}^\dagger - \gamma(\mathbf{g}_t - \nabla f(\mathbf{x}_t))\|_2^2 \\
&= \|\mathbf{x}_t - \mathbf{x}^\dagger\|_2^2 - 2\gamma\left\langle\nabla f(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x}^\dagger\right\rangle + \gamma^2\|\nabla f(\mathbf{x}_t)\|_2^2 + \gamma^2\|\mathbf{g}_t - \nabla f(\mathbf{x}_t)\|_2^2 \\
&\geq \|\mathbf{x}_t - \mathbf{x}^\dagger\|_2^2 - 2\gamma\|\nabla f(\mathbf{x}_t)\|_2\|\mathbf{x}_t - \mathbf{x}^\dagger\|_2 + \gamma^2\|\nabla f(\mathbf{x}_t)\|_2^2 + c^2\gamma^2\sigma^2 \\
&= \|\mathbf{x}_t - \mathbf{x}^\dagger\|_2^2 + \gamma\|\nabla f(\mathbf{x}_t)\|_2(\gamma\|\nabla f(\mathbf{x}_t)\|_2 - 2\|\mathbf{x}_t - \mathbf{x}^\dagger\|_2) + c^2\gamma^2\sigma^2 \\
&\geq \|\mathbf{x}_t - \mathbf{x}^\dagger\|_2^2 + \gamma\|\nabla f(\mathbf{x}_t)\|_2(\gamma\mu^\dagger\|\mathbf{x}_t - \mathbf{x}^\dagger\|_2 - 2\|\mathbf{x}_t - \mathbf{x}^\dagger\|_2) + c^2\gamma^2\sigma^2 \\
&\geq \|\mathbf{x}_t - \mathbf{x}^\dagger\|_2^2 + \gamma\|\nabla f(\mathbf{x}_t)\|_2\|\mathbf{x}_t - \mathbf{x}^\dagger\|_2(\gamma\mu^\dagger - 2) + c^2\gamma^2\sigma^2 \\
&\geq \|\mathbf{x}_t - \mathbf{x}^\dagger\|_2^2 + \gamma L_M\|\mathbf{x}_t - \mathbf{x}^\dagger\|_2^2(\gamma\mu^\dagger - 2) + c^2\gamma^2\sigma^2 \\
&\geq \|\mathbf{x}_t - \mathbf{x}^\dagger\|_2^2 + \gamma L_M r^2(\gamma\mu^\dagger - 2) + 2\gamma L_M r^2 + \epsilon \\
&\geq \|\mathbf{x}_t - \mathbf{x}^\dagger\|_2^2 + \gamma^2 L_M\mu^\dagger r^2 + \epsilon\,.
\end{aligned}$$

This means the distance to $\mathbf{x}^\dagger$ grows at least with the constant ϵ. Let $q := \frac{r^2}{\epsilon}$. Therefore, with probability at least p^q, one of $\mathbf{x}_{t+1}, \ldots, \mathbf{x}_{t+q}$ will be out of M. This holds for any consecutive q iterates. Partitioning the iterates to parts of consecutive iterates of size q, each part has a positive probability of containing a point outside M. Therefore SGD will reach a point outside of M almost surely. □

The function plotted in Figure 6 can be formally defined as follows:

$$\begin{aligned}
f_1(x) &:= 86400((x-\alpha)^3 - (2.9-\alpha)^3) + 2.9^2 \\
f_2(x) &:= \beta((x-3.1)^3 + 0.001) + f_1(3) \\
f_3(x) &:= -300(x-3.1)^2 + f_2(3.1) \\
f_{\text{tm}}(x) &:= \begin{cases} x^2 & x \leq 2.9 \\ f_1(x) & 2.9 < x \leq 3 \\ f_2(x) & 3 < x \leq 3.1 \\ f_3(x) & 3.1 < x \leq 3.7 \\ 450 * ((x-4.1)^2 - 0.16) + f_3(3.7) & 3.7 < x \end{cases}
\end{aligned}$$

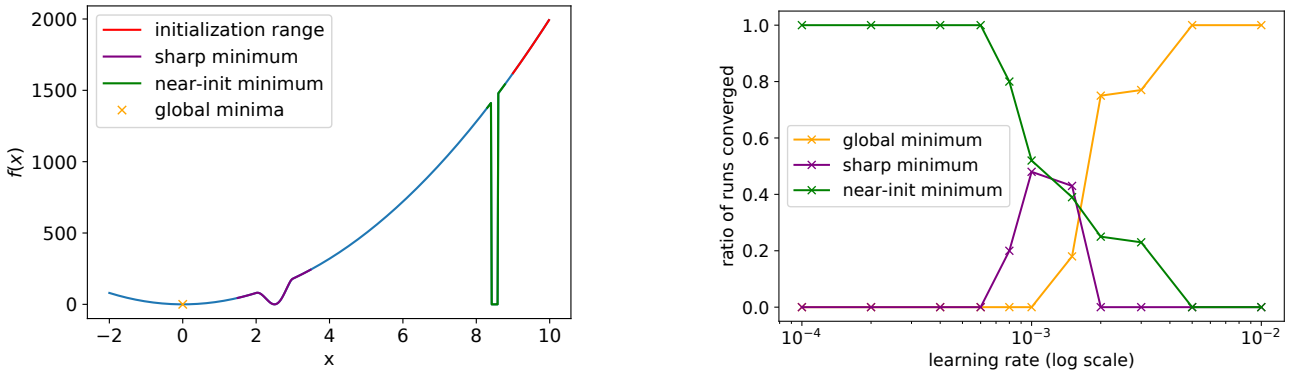
with $\alpha = 2.9 - \frac{\sqrt{2.9}}{360}$ and $\beta = 8640000(3-\alpha)^2$. This function satisfies the following properties:

- f_{tm} is 2-smooth over $\{x \mid x < 2.9\}$.
- f_{tm} is 900-OPSC towards 4.1 and 900-smooth over $\{x \mid 3.7 < x\}$.
- f_{tm} is 4.5-OPSC towards 4.1 over $\{x \mid 3.108 < x\}$.

We now proceed to proving Proposition 3, stating it formally here:

Proposition 3. *Consider running SGD on f_{tm} with stochastic noise ξ_t drawn i.i.d. at each step from the uniform distribution $Uniform(-\sigma, \sigma)$. If the learning rate is small such that it satisfies $\gamma < \frac{1}{30^2}$ the algorithm will not converge to $x_\star$ for some set of initialization points with positive Lebesgue measure. In contrast, with a large learning rate satisfying $0.4 \leq \gamma \leq 0.5$ it is possible to choose σ such that the algorithm will converge to $x_\star$ almost surely.*

Proof. Assume $\gamma < \frac{1}{900}$. Consider the case where the algorithm is initialized inside $M_1 := \{x \mid 3.7 < x < 4.5\}$. Then if σ satisfies $\sigma^2 \leq \frac{900}{\gamma} \cdot 0.4^2$ it will remain in M_1 with positive probability according Theorem 4. According to the same theorem, If this bound is not satisfied, then we have $\frac{\sigma^2}{2} \geq \frac{2 \cdot 2 \cdot 2.9^2}{\gamma}$ which means any time SGD reaches a point in $M_\star := \{x \mid -2.9 < x < 2.9\}$, it will escape from it almost surely within a constant number of steps. This means that the algorithm will never stay close to $x_\star = 0$ forever or for an arbitrarily long number of steps.



(a) Landscape of function

(b) Distribution of convergence to each minima

Figure 8: The function used in the toy experiments which has two local minima, a mostly flat minima near the initialization points and a sharp minima further away. It can be clearly observed how as the learning rate grows the two effects of avoiding parts of the landscape and escaping sharp minimas allow GD to converge to the global minimum.

Now consider the case where the learning rate is large enough. Choose σ such that $5.1 < \sigma < 5.5$. Note that if the iterates reach the set $\{x \mid -2.9 < x < 2.9\}$ they will never exit it since we have

$$\begin{aligned} |x - \gamma(2x + \xi_t)| &= |(1 - 2\gamma)x - \gamma\xi_t| \\ &\leq (1 - 2\gamma)(2.9) + \gamma(5.5) \\ &\leq (1 - 2\gamma)(2.9) + 2\gamma(2.9) \\ &\leq 2.9\,. \end{aligned}$$

We will now show that from any other point there is a positive probability of reaching the range $[-2.9, 2.9]$. This fact combined with the almost sure guarantee of not escaping from $[-2.9, 2.9]$, proves that the algorithm will converge to this set almost surely.

Note that f_{tm} is 4.5-OPSC towards $x = 4$ over the set $\{x \mid 3.108 < x\}$. Since $\gamma > 0.45$, it can be seen from the proof of Lemma 1 that SGD will continue to get further from $x = 4$ while it is in this set. Furthermore, given the direction of the gradients it is clear that the iterates would alternate between being less and more than 4. Therefore, at some point, the iterates will exit this set reaching a point $x_t < 3.108$. Note that with a positive probability, the noise will not interfere with this escape since there is at least 0.5 probability that the noise is aligned with the gradient direction.

If $3.1 < x < 3.108$, the gradient value is less than 5. Since $\sigma > 5.1$ there is a positive probability of moving to the region $x < 3.1$. When $2.9 < x < 3.1$, because of the positive probability of alignment between the gradient and the noise, SGD will move to $x < 2.9$ with positive probability. Finally, given the smoothness of the region $x < 2.9$, if $x < -2.9$ SGD will converge toward $x_\star = 0$, ultimately reaching the region $-2.9 < x < 2.9$ with positive probability. This completes the proof. □

I Toy Example

In order to demonstrate the effects discussed in Section 4, we experiment with running GD over a simple function. The landscape of this function is plotted in Figure 8a and its formula is presented in Appendix J. The function has two minima, one near the initialization and one further away. Since the near-init minimum is almost completely flat, i.e. gradient is constant and equal to zero (except for the edges which are extremely sharp lines in order to ensure the function remains continuous), if GD reaches a point in this region, it will remain there. However, as this region is very close to the initialization, Lemma 2 (more particularly Corollary 2) suggests that GD with large enough learning rate, will probably not reach any points in this region. To demonstrate this more clearly, we plot the trajectory of GD from several initialization points in Figure 1. It is worth noting that even with large learning rate it is possible for GD to get stuck in this region while it is possible to avoid this region even with a small learning rate. However, as suggested by our theoretical upper bound, the probability of this phenomenon increases with the learning rate.

The other minimum is much sharper than the rest of the function and therefore we can expect an escaping behavior similar to the one described by Lemma 1. This behavior is demonstrated in Figure 3. Note that unlike the previous case, GD with large learning rate always (except when landing directly at the minimum) escapes the sharp minimum while GD with small learning rate converges.

We measure rate of convergence of GD for 100 different random initialization to each of these three regions for different learning rates. The results are plotted in Figure 8b. We observe that as the learning increases, the rate of avoiding the near initialization minimum increases. While the learning rate is not still high enough, GD will converge to the sharp minimum while as the learning rate increases further, it is also able to escape the sharp minimum and converge to the global minimum. This behavior is completely compatible with what can be expected based on the results and effects discussed in Section 4.

J Function for Toy Example

$$f(x) := \begin{cases} -1600(x-2.5)^5 - 2000(x-2.5)^4 + 800(x-2.5)^3 + 1020(x-2.5)^2 & 2 \leq x \leq 3, \\ 1411.2 \times (1 - 10^4(x-8.4)) & 8.4 \leq x \leq 8.40001, \\ 0 & 8.40001 \leq x \leq 8.59999, \\ 1479.2 \times (10^4(x-8.6)+1) & 8.59999 \leq x \leq 8.6, \\ 20x^2 & \text{otherwise.} \end{cases}$$

K 2D Toy Example

To build more intuition and show the effect of large learning rate extends to multi-dimensions, we also provide a toy example on 2D. Figure 9 shows the landscape of our toy example which contains four local minima that are also sharp. Consider GD initialized randomly on the region $W := \{(x, y) \mid 3 \leq x, y \leq 4\}$. Then, using a small learning rate GD will converge to the minimum in the region $[1, 2] \times [1, 2]$. However, using a larger learning rate allows escaping that minimum. Increasing the magnitude, GD can also jump over the minimum completely. In these cases, GD will converge towards the global minimum at $(0, 0)$.

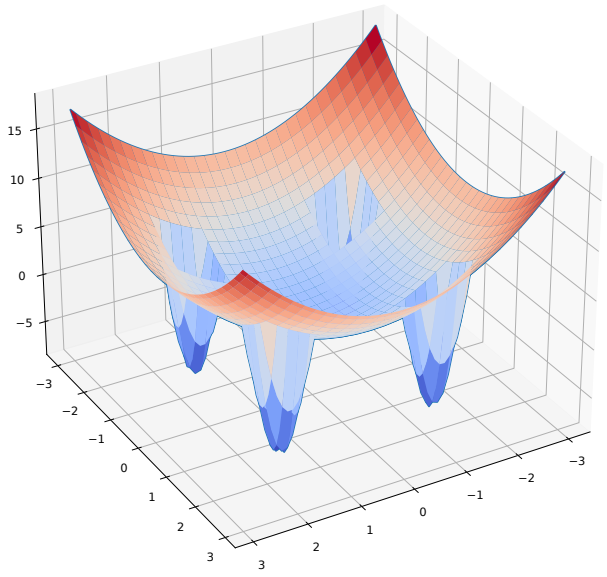


Figure 9: The landscape of the function $f(x, y) := x^2 + y^2 - 200ReLU(|x| - 1)ReLU(|y| - 1)ReLU(2 - |x|)ReLU(2 - |y|)$.

L Results of Stopping Repeats from Different Epochs

In Section 5.1, we explained that at the end of training we stop using the same batch for k steps and train in the standard way (each batch used just once) for additional 10 epochs. This was done to make sure the model that is used to obtain the accuracy on the test data is not overfitted on one batch which might be more likely to happen at the end of the training. In this section, we also experiment with stopping repeats, i.e. using the same batch for k steps, earlier in the training. The result is plotted in Figure 10. No significant improvement is observed.

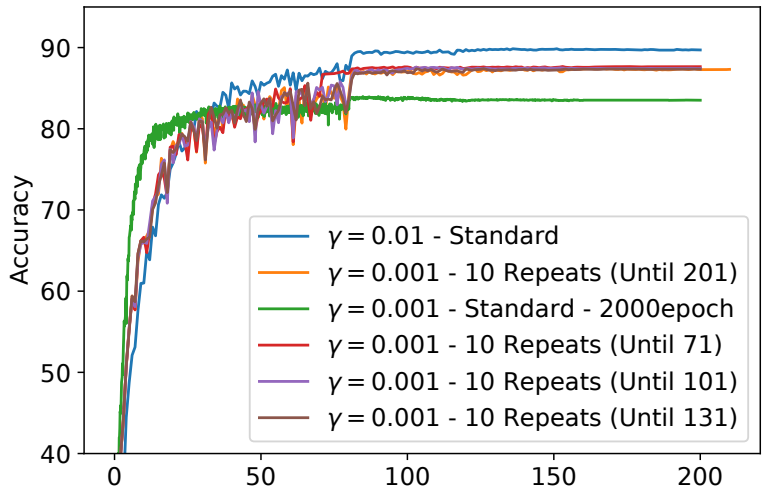


Figure 10: Plot of test accuracy when we stop using the same batch several times (doing repeats) at different epochs. It can be clearly observed that the stopping epoch does not affect the final accuracy and the gap with the case of GD with a large learning rate can be clearly observed.

M Experiments on CIFAR100

In order to make sure our results extend to other scenarios, we repeat the experiments in Section 5.1 on CIFAR100 and observe a similar behavior. The accuracy on the train and test datasets during training are plotted in Figure 11.

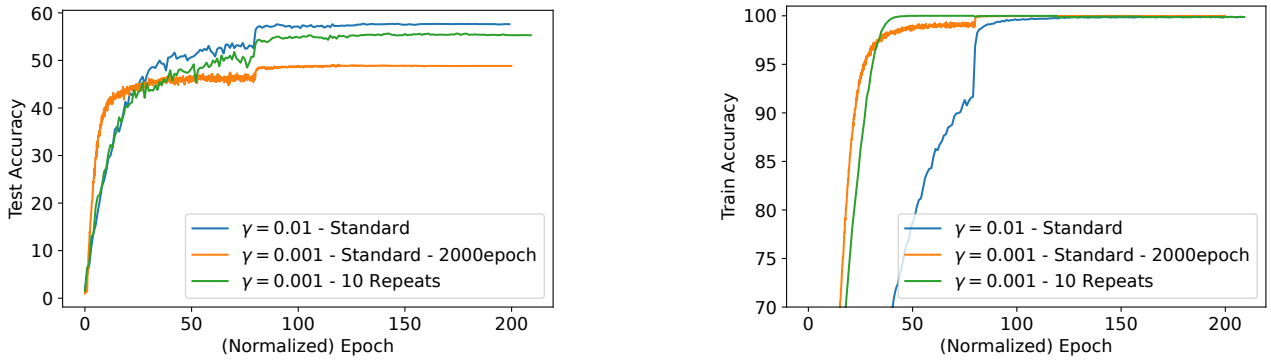


Figure 11: Comparsion between performance of SGD with different learning rates on CIFAR100. Repeating batches is turned off at epoch 200 and 10 additional epochs are performed (green). For the experiment with 2000 epochs (orange), the plot is normalized to 200 epochs. For more explanations refer to Figure 5 and Section 5.1.

N Experiments on SGD without Momentum

In Section 5.1, we designed an experiment to show the effect of large learning rate is important and goes beyond controlling the effect of stochastic noise on the trajectory. Since our goal was to demonstrate the relevance and importance of analyzing

these effects for the practical scenarios, we used the standard training settings including momentum and weight decay. For completeness, in this section we also include the results of applying SGD with repeats without momentum and without weight decay. We compare standard SGD with learning rate 0.05. standard SGD with learning rate 0.005, and SGD with $k = 10$ repeats and learning rate 0.005. Accuracy on test and train datasets throughout training is plotted in Figure 12. The figure also contains the accuracy during training with momentum to allow comparison. As expected, applying SGD without momentum performs worse than SGD with momentum. The gap between small and large learning rate can be observed in this case as well. However, we do not observe an improvement when applying repeats.

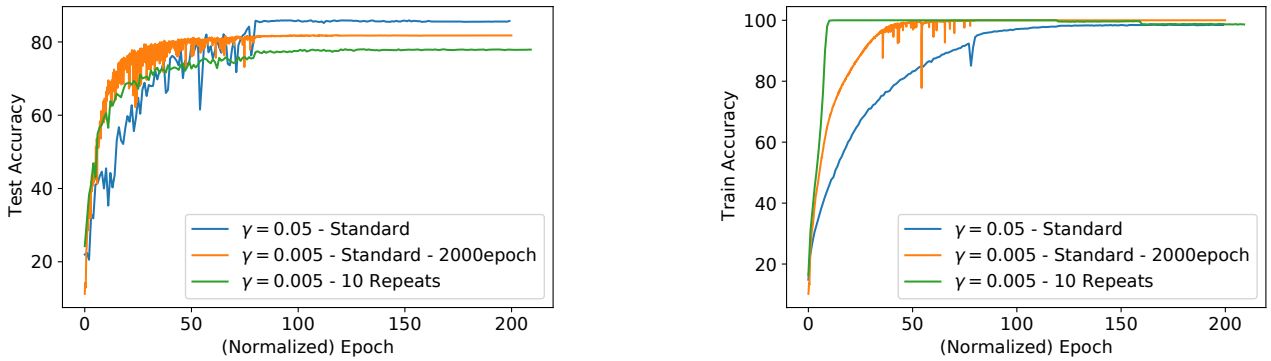


Figure 12: Comparsion between performance of SGD without momentum and weight decay and with different learning rates on CIFAR10. Repeating batches is turned off at epoch 200 and 10 additional epochs are performed (green). For the experiment with 2000 epochs (orange), the plot is normalized to 200 epochs. For more explanations refer to Figure 5 and Section 5.1.

O Loss on the line between large and small learning rate trajectories

In Section 5.2, we observed that GD with a large learning rate shows behavior similar to escaping and follows a different trajectory than GD with the small learning rate. In this section, we plot the loss along the line between the first point in the trajectory of GD with small learning rate (hereafter called the origin) and different points along the trajectory of GD with the large learning rate. Figure 13 shows the loss based on the norm of the distance to the origin. As expected the loss increases along the line between the origin and points at the beginning of the trajectory. This is when GD is showing escaping behaviors. However, interestingly, the loss is decreasing along the line between the origin and points encountered later in the trajectory.

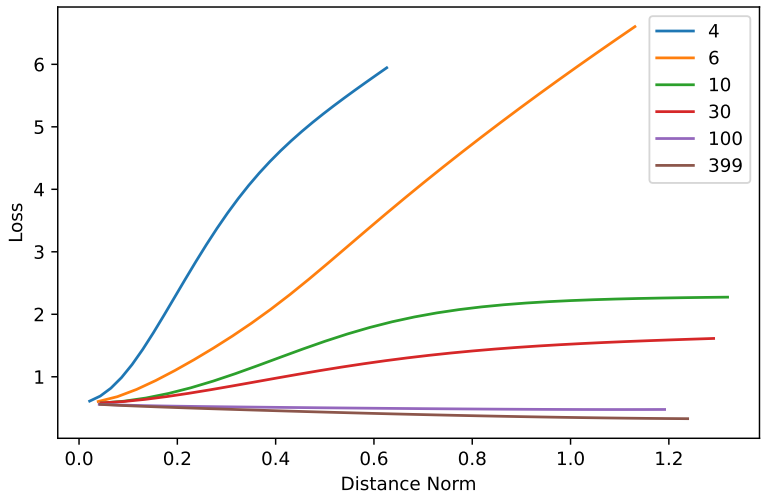


Figure 13: The value of loss along the line between the first point in the trajectory of GD with small learning rate and different points in the trajectory of GD with a large learning rate. For more detailed explanation of the settings, refer to Section 5.2. Each line corresponds to the value of loss measured on 30 points along the line between the initialization and the parameters after an step. The step number for each line is written in the box located on the top-right of the plot.