THREE FACTORS STRONGLY INFLUENCING MINIMA IN SGD

Anonymous authors
Paper under double-blind review

ABSTRACT

We focus on the importance of noise in stochastic gradient descent (SGD) based training of deep neural networks (DNNs). We develop theory that studies SGD training as a stochastic differential equation and show that its stationary distribution is related to the loss surface. Our analysis suggests that the combination of batch size, learning rate, and the variance of the true loss gradients acts as a hyper-parameter steering the behavior of SGD and determines the trade-offs between the depth and width of the minima that SGD converges to. In a nutshell, a higher ratio of learning rate to batch size leads to wider minima. We validate our theory by examining the correlation between these three factors and the final performance and sharpness of the minimum found. As a verification of our theory, we empirically demonstrate that the learning dynamics is similar between experiments with different learning rates and batch sizes in SGD if the ratio of learning rate to batch size is the same.

1 INTRODUCTION

Despite being massively over-parameterized models (Zhang et al., 2016), deep neural networks (DNNs) have demonstrated good generalization ability and achieved state-of-the-art performances in many application domains such as image (He et al., 2016) and speech recognition (Amodei et al., 2016). The reason for this success has been a focus of research recently but still remains an open question. Our work provides new theoretical insights and useful suggestions for deep learning practitioners.

The standard way of training DNNs corresponds to minimizing a loss function using stochastic gradient descent (SGD) and its variants (Bottou, 1998). In SGD, parameters are updated by taking a small discrete step depending on the learning rate (LR) in the direction of the negative loss gradient, which is approximated based on a small subset of training examples (called mini-batch). Since DNNs are highly non-convex functions with multiple minima, SGD in general converges to different local minima depending on optimization hyper-parameters, initialization and the loss curvature.

Recently, several works (Arpit et al., 2017; Advani & Saxe, 2017; Shirish Keskar et al., 2016) have investigated how SGD impacts generalization in DNNs. It has been argued that local minima with low curvature (wide minima) can generalize better than minima whose curvature is large (sharp minima) (Hochreiter & Schmidhuber, 1997; Shirish Keskar et al., 2016). Specifically, (Shirish Keskar et al., 2016) found that batch size (BS) strongly affects the curvature of minima that are reached by SGD (i.e., smaller batch sizes usually lead to wider minima). We generalize this notion to talk about the controllable noise level in SGD which we theoretically show is determined by the ratio of learning rate over batch size. As such, we discuss how SGD with appropriate noise level tends to converge to deeper and wider minima. In this vein, while (Dinh et al., 2017) discuss the existence of minima with different widths but which behave similarly in terms of predictions, we argue that SGD naturally tends to find wider minima when appropriate noise levels are used.

We approximate SGD as a continuous stochastic differential equation (Bottou, 1998; Mandt et al., 2017). Based on this, we derive the stationary distribution of this stochastic process, and further derive the relative probability of landing in one local minima as compared to another depending on their depth and width. Our main finding is that the ratio of learning rate and batch-size along with noise about the true gradient (due to the choice of batch sample) influences the trade-off between the depth and sharpness of the final minima, with a high ratio of learning rate to batch size

favouring flatter minima. In addition, our analysis provides a theoretical justification for the empirical observation that scaling the learning rate linearly with batch size leads to good performance of DNNs (Krizhevsky, 2014; He et al., 2016).

We verify our theoretical insights experimentally on several models and datasets. In particular, we demonstrate that high learning rate to batch size ratio (due to either high learning rate or low batch-size) leads to wider minima and correlates well with better validation performance. We also observe that a high learning rate to batch size ratio helps to prevent memorization. Furthermore, we show that multiplying each of the learning rate and the batch size by the same scalar factor results in similar training dynamics, but if the scalar factor gets too large, then the assumptions of the theory are violated and performance drops.

2 Related work

The relationship between stochastic gradient descent (SGD) and sampling a posterior distribution via stochastic Langevin methods has been the subject of discussion in a number of papers (Chen et al., 2014; Ding et al., 2014; Vollmer et al., 2015; Welling & Teh, 2011; Shang et al., 2015; Sato & Nakagawa, 2014). In particular, Mandt et al. (2017) describe the dynamics of stochastic gradient descent (SGD) as a stochastic process that can be divided into three distinct phases. In the first phase, weights diffuse and move away from the initialization. In the second phase the gradient magnitude dominates the noise in the gradient estimate. In the final phase, the weights are near the optimum. (Shwartz-Ziv & Tishby, 2017) make related observations from an information theoretic point of view and suggest the diffusion behavior of the parameters in the last phase leads to the minimization of mutual information between the input and hidden representation. In a similar vein, we relate the SGD dynamics to the stationary distribution of the stochastic differential equation. Our derivation bears similarity with Mandt et al. (2017). However, while Mandt et al. (2017) study SGD as an approximate Bayesian inference method in the final phase of optimization in a locally convex setting, our end goal is to analyse the stationary distribution over the entire parameter space reached by SGD. Further, our analysis allows us to compare the probability of SGD ending up in one minima over another, which is novel in our case.

Our work is closely related to the ongoing discussion about the role of large batch size and sharpness of found minima in generalization (Shirish Keskar et al., 2016). Goyal et al. (2017); Hoffer et al. (2017) empirically show that rescaling the learning rate, and training for more epochs, leads to good generalization using large batch size. Our novelty is in explaining from the theoretical point of view the importance of the specific ratio of learning rate to batch size. We also observe that the final minima found by large batch size can be wide.

Our work is also related to other work on the importance of noise in SGD, which have been previously explored. The main inspiration for having a learning rate schedule is to anneal noise (Bottou, 1998). Neelakantan et al. (2015) observe empirically that adding noise can aid optimization of very deep networks. Our analysis allows us to derive the impact of the gradient noise in the SGD stationary distribution. Additionally, our work also provides intuition toward explaining the recently proposed Cyclic Learning Rate (CLR) schedule (Smith, 2015). CLR schedules have demonstrated good optimization and generalization performances, but are grounded on empirical observation rather than on a theoretical understanding. We show that one can replace learning rate annealing with an equivalent batch size schedule. It suggests that the benefit of CLR relates to the noise that it induces and can be thought of as mixing in Monte Carlo Markov Chain (MCMC) methods (Graham & Storkey, 2017).

3 Theoretical Results

Our focus in this section is on finding the relative probability with which we end optimization in a certain minimum, as compared to another minimum. We will find that the relative probability depends on the local geometry of the loss function at each minimum, and on the BS, the LR and the covariance of the stochastic gradients.

3.1 SETUP

We follow a theoretical setup similar to Mandt et al. (2017), approximating SGD with a continuous-time stochastic process, which we now outline.

Let us consider a model parametrized by $\boldsymbol{\theta} = \{\theta_1, \ldots, \theta_q\}$. For N training examples $\boldsymbol{x}_i, i \in \{1, ..., N\}$, the loss function, $L(\boldsymbol{\theta})$, and the corresponding gradient $\mathbf{g}(\boldsymbol{\theta})$, are defined based on the sum over the loss values for all training examples. Stochastic gradients $\mathbf{g}^{(S)}(\boldsymbol{\theta})$ arise when we consider a batch $\mathcal{B}$ of size $S < N$ of random indices drawn uniformly from $\{1, ..., N\}$ and form an (unbiased) estimate of loss and gradient based on the corresponding subset of training examples

$$L^{(S)}(\boldsymbol{\theta}) = \frac{1}{S} \sum_{n \in \mathcal{B}} l(\boldsymbol{\theta}, \boldsymbol{x}_n) \ , \qquad \mathbf{g}^{(S)}(\boldsymbol{\theta}) = \frac{\partial}{\partial \boldsymbol{\theta}} L^{(S)}(\boldsymbol{\theta}) \ .$$

We consider stochastic gradient descent (SGD) with learning rate η, as defined by the update rule

$$\boldsymbol{\theta}(t+1) = \boldsymbol{\theta}(t) - \eta \boldsymbol{g}^{(S)}(\boldsymbol{\theta}) \ .$$

We now make the following assumptions:

(1) By the central limit theorem (CLT), we assume the gradient noise is Gaussian with covariance matrix $\frac{1}{S}\mathbf{C}(\boldsymbol{\theta})$

$$\mathbf{g}^{(S)}(\boldsymbol{\theta}) = \mathbf{g}(\boldsymbol{\theta}) + \frac{1}{\sqrt{S}} \Delta \mathbf{g}(\boldsymbol{\theta}), \text{ where } \Delta \mathbf{g}(\boldsymbol{\theta}) \sim N(0, \mathbf{C}(\boldsymbol{\theta})) \ .$$

We note that the covariance is symmetric positive-semidefinite, and so can be decomposed into the product of two matrices $\mathbf{C}(\boldsymbol{\theta}) = \mathbf{B}(\boldsymbol{\theta})\mathbf{B}^\top(\boldsymbol{\theta})$.

(2) We assume the discrete process of SGD can be approximated by the continuous time limit of the following stochastic differential equation (known as a Langevin equation)

$$\frac{d\boldsymbol{\theta}}{dt} = -\eta \boldsymbol{g}(\boldsymbol{\theta}) + \frac{\eta}{\sqrt{S}} \mathbf{B}(\boldsymbol{\theta}) \mathbf{f}(t) \tag{1}$$

where $\mathbf{f}(t)$ is a normalized Gaussian time-dependent stochastic term.

3.2 MAIN RESULTS

The Langevin equation is a stochastic differential equation, and we are interested in its stationary distribution to gain insights into the behavior of SGD and the quality of minima it converges to. It can be derived by finding the stationary solutions of a partial differential equation known as the Fokker-Planck equation, which governs the evolution of the probability density of the value of the parameters with time[1]. The Fokker-Planck equation and its derivation can be found in Appendix A in equation (5). We arrive at the following theorem (all the notations are defined in the setup section above):

Theorem 1 (Stationary Distribution)**.** *Assume that the gradient covariance is proportional to the identity, i.e.* $\mathbf{C}(\boldsymbol{\theta}) = \sigma^2(\boldsymbol{\theta})\mathbf{I}$ *and that* $|\nabla_{\boldsymbol{\theta}}(\sigma^2(\boldsymbol{\theta}))| \ll 2S|\boldsymbol{g}|\eta$*. Then the stationary distribution of the stochastic differential equation 6 is given by*

$$P(\boldsymbol{\theta}) = P_0 e^{-L(\boldsymbol{\theta})/n} \ , \tag{2}$$

where $n \equiv \frac{\sigma^2(\boldsymbol{\theta})\eta}{2S}$ *and* P_0 *is a normalization constant.*

Here $P(\boldsymbol{\theta})$ defines the density over the parameter space. The above result says that if we run SGD long enough (under the assumptions made), then the likelihood of the parameters being in a particular state asymptotically follows the above density. Note, that $n \equiv \frac{\sigma^2(\boldsymbol{\theta})\eta}{2S}$ is a measure of the noise in the system and it depends on two parts: $n_c \equiv \eta/S$, which is the controllable noise set by the choice of hyper-parameters, and the gradient variance $\sigma^2(\boldsymbol{\theta})$, which is not a tunable hyper-parameter.

[1]The Fokker-Plank equation was also used in the context of the stochastic gradient langevin dynamics (SGLD) algorithm in the Bayesian learning setting (Sato & Nakagawa, 2014). While in this setting it was found that the stochastic noise can be ignored, we arrive at a different conclusion.

Given the probability density $P(\boldsymbol{\theta})$, we are now interested in deriving the probability of ending at a given minimum, $\boldsymbol{\theta}_A$, which we will denote by lowercase $p_A = \tilde{p}_A C$, where C is a normalization constant (the unnormalized probability $\tilde{p}_A$ is all we are interested in when estimating the relative probability of finishing in a given minimum compared to another one). This probability is derived in Appendix D, and given in the following theorem, which is the core result of our theory.

Theorem 2 (Probability of ending in minima A)**.** *Assume the loss is locally convex with Hessian $\mathbf{H}_A$ and loss L_A at the minimum $\boldsymbol{\theta}_A$. Then the unnormalized probability of ending in minima $\boldsymbol{\theta}_A$ is*

$$\tilde{p}_A = \sqrt{\frac{(2\pi n_A)^q}{\det \mathbf{H}_A}} \exp\left(-\frac{L_A}{n_A}\right) \tag{3}$$

where q is the number of parameters and $n_A \equiv n_c \sigma^2(\boldsymbol{\theta}_A)$ is the noise at the minimum.

Here we see that the probability of landing in a given minimum depends on the geometry of the loss surface (captured by the height L_A and the determinant of the Hessian $\det \mathbf{H}_A$, which determines the width of the minimum) as well as on the noise n_A at $\boldsymbol{\theta}_A$.

For larger noise, n_A, the factor $\exp\left(-\frac{L_A}{n_A}\right)$ is down-weighted compared to $\sqrt{\frac{(2\pi n_A)^q}{\det \mathbf{H}_A}}$, which grows with larger n_A. Thus, with larger noise the width gets more important than the height of the minimum, while the converse holds if the noise is low enough. Also notice, that with n_A the probability $\tilde{p}_A$ directly depends on the ratio of learning rate and batch size $n_c = \eta/S$. We will explore the implications of this experimentally in Section 4.

To see which kind of minima is preferred, it is instructive to consider the ratio of probabilities p_A and p_B given by

$$\frac{p_A}{p_B} = \sqrt{\frac{n_A^p}{n_B^p}} \sqrt{\frac{\det \mathbf{H}_B}{\det \mathbf{H}_A}} \exp\left(\frac{L_B}{n_B} - \frac{L_A}{n_A}\right) \quad .$$

Let us now look at the special case where $n_A = n = n_B$ (other cases can be found in Appendix E). Without loss of generality we can set $L_A \geq L_B$. If $L_A = L_B$ the minimum with lower determinant of the Hessian (i.e. the flatter minima) is more probable. If $L_A \geq L_B$, it holds that $p_A \geq p_B$ if and only if

$$\frac{1}{n} \leq \frac{\log\left(\sqrt{\frac{\det \mathbf{H}_B}{\det \mathbf{H}_A}}\right)}{(L_A - L_B)} \equiv Y \quad .$$

That is, there is an upper bound on the inverse of the noise for θ_A to be favored in the case that its loss is higher than at θ_B, and this upper bound depends on the difference in the heights compared to the ratio of the widths. In particular we can see that if $\det \mathbf{H}_B < \det \mathbf{H}_A$, then $Y < 0$, and so no amount of noise will result in θ_A being more probable than θ_B – that is, if the minimum at θ_A is both higher and sharper than the minimum at θ_B, it is never reached with higher probability than θ_B, regardless of the amount of noise. However, if $\det \mathbf{H}_B > \det \mathbf{H}_A$ then $Y > 0$, and there is a lower bound on the noise

$$n > \frac{(L_A - L_B)}{\log\left(\sqrt{\frac{\det \mathbf{H}_B}{\det \mathbf{H}_A}}\right)} \tag{4}$$

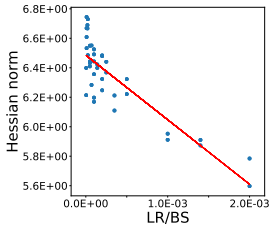
to make $\boldsymbol{\theta}_A$ more probable than $\boldsymbol{\theta}_B$– that is if the minimum at $\boldsymbol{\theta}_A$ is higher but flatter than the minimum at $\boldsymbol{\theta}_B$, it is favored over $\boldsymbol{\theta}_B$, as long as the noise is large enough, as defined by eq. (4).

To summarize, the presented theory shows that the noise level in SGD (which is defined by the learning rate, batch size and gradient covariance) controls the extent to which optimization favors wider over lower minima. Increasing the noise by increasing the ratio of learning rate and batch size, increases the probability of wider compared to lower optima.

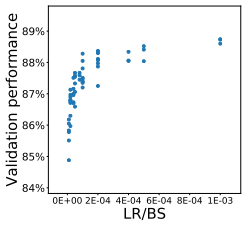
4 Experiments

4.1 Impact of SGD on minima

In this section, we empirically study the impact of the learning rate η, batch size S on the local minimum that SGD finds.

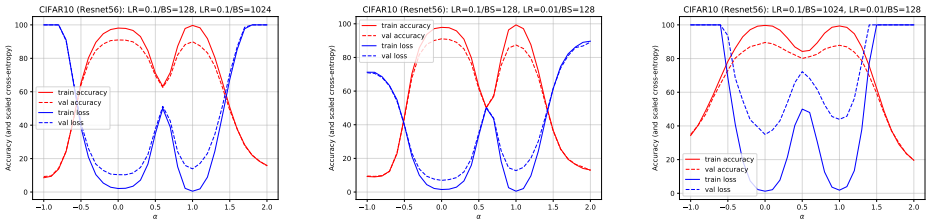


(a) Correlation of $\frac{\eta}{S}$ with norm of Hessian.



(b) Correlation of $\frac{\eta}{S}$ with validation accuracy.

Figure 1: Impact on SGD with ratio of learning rate (LR) η and batch size (BS) S for 20 layer ReLU Network on FashionMNIST.



(a) left $\frac{\eta=0.1}{S=128}$, right $\frac{\eta=0.1}{S=1024}$

(b) left $\frac{\eta=0.1}{S=128}$, right $\frac{\eta=0.01}{S=128}$

(c) left $\frac{\eta=0.1}{S=1024}$, right $\frac{\eta=0.01}{S=128}$

Figure 2: Interpolation of Resnet56 Networks trained with different with different learning rate to batch size ratio, $\frac{\eta}{N}$. α (x-axis) corresponds to the interpolation coefficient. Lower $\frac{\eta}{N}$ ratio leads to sharper minima.

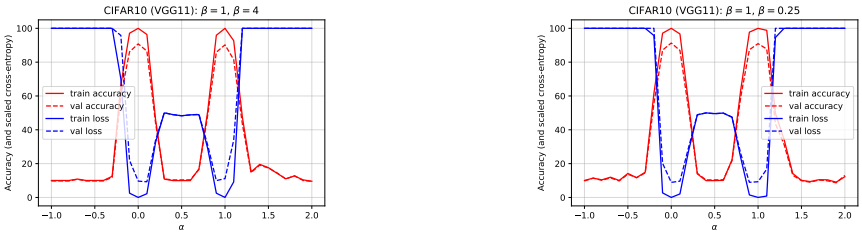
We first focus on an MLP with 20 layers with ReLU activation functions (similar to one used in (Neelakantan et al., 2015)) and trained on FashionMNIST. We study how the controllable noise ratio $n_c = \frac{\eta}{S}$ leads to minima with different curvatures and validation accuracy. To measure the curvature at that minimum, we compute the norm of its Hessian using a finite difference (Wu et al., 2017) (higher Hessian norm imples higher sharpness of minima). In Figure 1a, we report the norm of the Hessian for local minima obtained by SGD for different n_c with $\frac{\eta}{S}$ (where $\eta \in [5e-3, 5e-2]$ and $S \in [50, 1000]$). As $n_c = \frac{\eta}{S}$ grows, we observe that the norm of the Hessian at the minima also decreases, suggesting that higher $\frac{\eta}{S}$ pushes the optimization towards a flatter minimum. This agrees with Theorem 2, (3), that higher $\frac{\eta}{S}$ favors flatter over sharper minima.

In Figure 1b, we explore the impact of $n_c = \frac{\eta}{S}$ on the final validation performance and confirm that better generalization correlates with higher n_c. Taken together, Fig. 1a and Fig. 1b imply wider minima correlate well with better generalization. As $n_c = \frac{\eta}{S}$ increases, SGD finds local minima that generalize better. In the Appendix, we report similar results for Resnet applied on CIFAR10 and the 20 layer ReLU network with bad initialization.

To further illustrate the behavior of SGD with different noise levels, we train three Resnet56 models on CIFAR10 using naive SGD, but with different $\frac{\eta}{S}$. Our baseline model uses $\frac{\eta=0.1}{S=128}$. We define a large batch model $\frac{\eta=0.1}{S=1024}$ and a small learning rate model $\frac{\eta=0.01}{S=128}$. The large batch and the small learning rate model have approximately the same $\frac{\eta}{S}$ ratio. In Figure 2, we follow (Goodfellow et al., 2014) and look at the loss between interpolation of different pairs of models. More specifically, let f_1 and f_2 be the early stop models of two SGD optimizers using different $\frac{\eta}{S}$. In Fig. 2, we report $L((1-\alpha)f_1 + \alpha f_2$ where L is the loss function and $\alpha \in [-1, 2]$.

We observe that both model with large batch or low learning rate end in a sharper minimum relatively to the baseline model, both having a lower $\frac{\eta}{S}$ than the baseline. Each of the three plots in Figure 2 adds empirical weight to our theoretical prediction that higher $n_c = \eta/S$ gives preference to wider minima over sharper minima.

Finally we perform a similar experiment on VGG-11 on CIFAR-10, but in this case, we train all the models with the same noise level but different values of learning rate and batch size. Specifically,



(a) $\beta = 1$ corresponds to model at $\alpha = 0$ and $\beta = 4$ corresponds to model at $\alpha = 1$

(b) $\beta = 1$ corresponds to model at $\alpha = 0$ and $\beta = 4$ corresponds to model at $\alpha = 0.25$

Figure 3: Left: $\frac{\eta=0.1\times\beta}{S=50\times\beta}$, right: $\frac{\eta=0.1\times\beta}{S=50\times\beta}$. Interpolation between parameters of models trained with the same learning rate (η) to batch-size (S) ratio, but different η and S values determined by β. Hence our theory predicts the minima for these models should be qualitatively similar as can be seen by these plots.

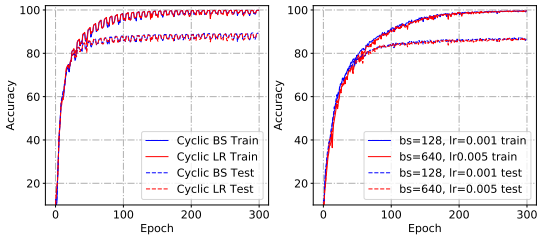


Figure 4: Learning rate schedule can be replaced by an equivalent batch size schedule. The ratio of learning rate to batch size is equal at all times in both red and blue in each plot. Train and test accuracy for experiments involving VGG-11 architecture on CIFAR10 dataset. Left: cyclic batch size schedule (blue) in range 128 to 640, compared to cyclic learning rate schedule (red) in range 0.001 to 0.005. Right: constant batch size 128 and constant learning rate 0.001 (blue), compared to constant batch size 640 and constant learning rate 0.005 (red).

in this case we use $\frac{\eta=0.1\times\beta}{S=50\times\beta}$, where we use $\beta = 0.25, 1, 4$. We then interpolate between the model parameters of the pair of models and record loss and accuracy (shown in figure 3). We find that all the minima in this case have similar width and depth qualitatively showing that for the same noise ratio, SGD ends up in minima of similar quality.

4.2 $\frac{\eta}{S}$ RATIO DETERMINES LEARNING DYNAMICS OF SGD

A prediction of the theory is that the learning rate η and batch size S only appear in the ratio $n_c = \eta/S$ in the expressions for the probability of finishing in a given minimum. Based on this, the endpoint of SGD with a learning rate schedule $\eta \to \eta/a$, for some $a > 0$, and a constant batch size S, should be the same as the endpoint of SGD with a constant learning rate and a batch size schedule $S \to aS$. We see in Fig. 4 that exchanging a learning rate schedule for a batch size schedule leads to similar train and test accuracy, for both cyclic and constant schedules, in the case where the controllable noise $n_c = \eta/S$ is the same in each case. In particular we notice very similar dynamics in the case of a cyclic learning rate when compared to a cyclic batch size. Full loss curves and learning rate and batch size schedules for this experiment can be found in Appendix G.2 in Fig. 10. The test accuracy for cyclic batch size and cyclic learning rate was 89.39% and 89.24% respectively. For constant (batch size, learning rate) = (128, 0.001) and constant (batch size, learning rate) = (640, 0.005) the test accuracy was 87.25% and 86.92% respectively.

Note that exchanging batch size for learning rate such that the ratio η/S remains constant is different to the common assumption that one should trade them so as to keep the ratio $\eta/\sqrt{S}$ constant. This commonly used heuristic of scaling the learning rate with the square root of the batch size is used for example in (Hoffer et al., 2017), as a way of keeping the covariance matrix of the parameter

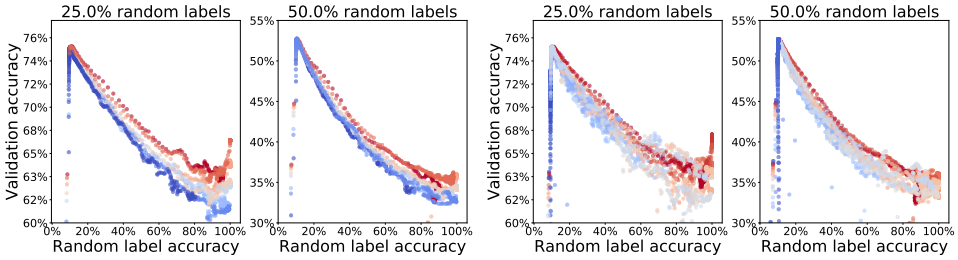


Figure 5: $\frac{\eta}{S}$ impact on memorization on dataset with added 25% and 50% random label noiseon the training set. The two left columns (and right columns) use 0.9 momentum (and 0.0 momentum) respectively. We observe that given a specific level of memorization, high $\frac{\eta}{S}$ leads to better generalization.

update step the same for any batch size. However, our point here is that our theory and experiments suggest exchanging learning rate and batch size in a way that keeps the ratio of controllable noise $n_c = \eta/S$ constant will result in the same stationary distribution.

4.3 Impact of SGD on Memorization

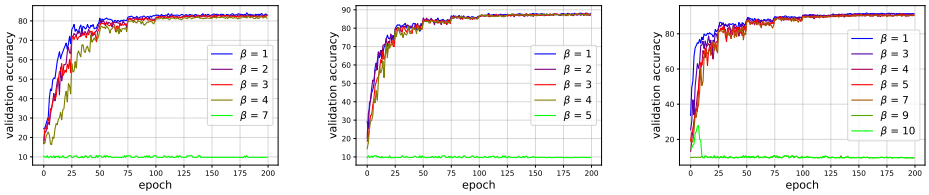
To generalize well, a model must identify the underlying pattern in the data instead of simply perfectly memorizing each training example. An empirical approach to test for memorization is to see if a deep neural network can fit a training set with random noise labels rather than the true labels (Zhang et al., 2016; Arpit et al., 2017). In this section, we highlight that SGD with a sufficient amount of noise reduces the amount of memorization in a network.

Experiments are performed using the MNIST dataset and an MLP similar to the one in (Arpit et al., 2017), but with 256 hidden units. We train the MLP on the dataset with different amounts of random labels in the training set. For each label noise level, we evaluate the impact of $\frac{\eta}{S}$, controlling the amount of noise in the SGD and its impact on the generalization performances. Specifically, we run a grid of batch size in $50, 100, 200, 400, 800$, learning rate in $0.05, 0.1, 0.2$ and momentum in 0.0 and 0.9. Models are trained for 300 epochs. We report in Fig. 5 the MLP performances for both the noisy training set and the clean validation set (without random label, to measure the model generalization ability).

Our analysis highlights that SGD with low controllable noise $n_c = \frac{\eta}{S}$ steers the endpoint of optimization towards a minima with low generalization ability, associated with a sharp minima, as predicted by our theory. We observe in Fig. 5 that larger noise in SGD (regardless if controlled using smaller batch size or larger learning rate) leads to solutions which generalize better for the same amount of memorized random labels on the training set. We also reproduce the observation reported in (Arpit et al., 2017): that randomization roughly starts after reaching maximum generalization - for more details see the curves in Fig. 11 included in Appendix G.3. They show that lower controllable noise $\frac{\eta}{S}$ optimize much slower and do not reach larger than 20% accuracy on random labels. For runs with momentum we exclude higher learning rates than 0.02 as they lead to divergence (see appendix).

4.4 Breaking Point of the Theory in Practice

Our analysis relies on the fact that the gradient step is sufficiently small so that first order approximation of a Taylor expansion is a good estimate of the loss function. In the case where the learning rate becomes too high, the first order approximation of a Taylor expansion is no longer suitable, in which case the continuous limit of the discrete SGD update equation will no longer be valid. In this case, the stochastic differential equation doesn't hold, and hence neither does the Fokker-Planck equation, and so our theory will not be predictive. In particular, we don't expect to arrive at the same stationary distribution as governed just by the ratio η/S. This is exemplified in Fig 6, where similar learning dynamics and final performance is seen when simultaneously multiplying the learning rate and batch size by a factor β up to a certain limit. This is done for different train-set size to see if the breaking point depends on this size. This plots seem to suggest that this happens for smaller β



(a) Train dataset size 12000 (b) Train dataset size 22500 (c) Train dataset size 45000

Figure 6: Breaking point of the theory: Experiments involving VGG-11 architecture on CIFAR10 dataset. Validation accuracy for different dataset sizes, and different β values. In each experiment, we multiply the learning rate (η) and batch size (S) with β such that the ratio $\frac{\eta \times \beta}{S \times \beta}$ is fixed. We observe that for the same ratio increasing the learning rate and batch size yields similar performance up to a β and then performance drops significantly.

values when dataset size is smaller. We highlight other limitations due to our theory assumption in appendix F. A similar experiment is performed on Resnets in figure 7.

4.5 Cyclical Batch and Learning Rate Schedules

It has been observed that a cyclical learning rate schedule leads to better generalization (Smith, 2015). In Sec. 4.2 we demonstrated that one can exchange cyclic learning rate schedule (CLR) with batch size (CBS) and approximately preserve the practical benefit of CLR. This inspired us to hypothesize that CLR by changing between noise levels ($\frac{\eta}{S}$) switches between sharp/deep and wide/shallow minima. To validate that, we run VGG-11 on CIFAR10 using 4 training schedules: CLR with stepsize of 4 and 15 epochs in each stage, CBS with stepsize 4 and 15 epochs in each stage. Each run is repeated 8 times and for each variant we track evolution of sharpness and accuracy. We observe that CBS and CLR with longer stages lead to $89.9 \pm 0.20\%$ and $90.2 \pm 0.10\%$ test accuracy in both cases $0.2 - 0.3\%$ above variance with shorter range. CBS and CLR with long stages times seem to be promising schedules for training DNNs. Finally, we validate that CBS and CLR switch between sharp/deep and wide/shallow minima, suggesting that CLR improves mixing time. Plot included in Appendix G.4.

5 Conclusions

We shed light on the role of noise in SGD optimization of DNNs and argued that three factors (batch size, learning rate and gradient variance) strongly influence the properties (loss and width) of the final minima at which SGD converges. Learning rate and batch size of SGD can be viewed as one effective hyper-parameter that act as a factor controllable noise $n_c = \eta/S$, which, together with the gradient covariance influences the trade-off between the loss and width of the final minima (high noise favors wide minima), which in turn tunes the robustness of the prediction function. [2].

Further, we experimentally verify that the controllable noise $n_c = \eta/S$ determines the width and height of the minima towards which SGD converges. We also show the impact of this controllable noise on the memorization phenomenon. We discussed the limitations of the theory and in what situations it breaks down, exemplified by when the learning rate gets too large. We also experimentally verify that η and S can vary in linear proportion as long as the controllable noise η/S remains the same. In addition, our experiments suggest that cyclical learning rates oscillate between sharp/deep and wide/shallow minima as long as the stage of increased noise is long enough to allow for mixing.

References

Madhu S Advani and Andrew M Saxe. High-dimensional dynamics of generalization error in neural networks. *arXiv preprint arXiv:1710.03667*, 2017.

[2]Robustness of model can also be controlled using regularization methods, e.g. L2, dropout or adversarial training.

Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Qiang Cheng, Guoliang Chen, et al. Deep speech 2: End-to-end speech recognition in english and mandarin. In *International Conference on Machine Learning*, pp. 173–182, 2016.

Devansh Arpit, Stanisław Jastrzębski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S. Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, and Simon Lacoste-Julien. A closer look at memorization in deep networks. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 233–242, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR. URL `http://proceedings.mlr.press/v70/arpit17a.html`.

Léon Bottou. Online learning and stochastic approximations. *On-line learning in neural networks*, 17(9):142, 1998.

T. Chen, E. B. Fox, and C. Guestrin. Stochastic gradient Hamiltonian Monte Carlo. In *Proceedings of the 31st International Conference on Machine Learning*, pp. 1683–1691, 2014.

N. Ding, Y. Fang, R. Babbush, C. Chen, R. D. Skeel, and H. Neven. Bayesian sampling using stochastic gradient thermostats. In *Advances in Neural Information Processing Systems 27*, pp. 3203–3211, 2014.

L. Dinh, R. Pascanu, S. Bengio, and Y. Bengio. Sharp Minima Can Generalize For Deep Nets. *ArXiv e-prints*, March 2017.

Ian J Goodfellow, Oriol Vinyals, and Andrew M Saxe. Qualitatively characterizing neural network optimization problems. *arXiv preprint arXiv:1412.6544*, 2014.

P. Goyal, P. Dollár, R. Girshick, P. Noordhuis, L. Wesolowski, A. Kyrola, A. Tulloch, Y. Jia, and K. He. Accurate, Large Minibatch SGD: Training ImageNet in 1 Hour. *ArXiv e-prints*, June 2017.

Matthew M Graham and Amos J Storkey. Continuously tempered hamiltonian monte carlo. *arXiv preprint arXiv:1704.03338*, 2017.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Sepp Hochreiter and Jürgen Schmidhuber. Flat minima. *Neural Computation*, 9(1):1–42, 1997.

E. Hoffer, I. Hubara, and D. Soudry. Train longer, generalize better: closing the generalization gap in large batch training of neural networks. *ArXiv e-prints*, May 2017.

Alex Krizhevsky. One weird trick for parallelizing convolutional neural networks. *arXiv preprint arXiv:1404.5997*, 2014.

S. Mandt, M. D. Hoffman, and D. M. Blei. Stochastic Gradient Descent as Approximate Bayesian Inference. *ArXiv e-prints*, April 2017.

A. Neelakantan, L. Vilnis, Q. V. Le, I. Sutskever, L. Kaiser, K. Kurach, and J. Martens. Adding Gradient Noise Improves Learning for Very Deep Networks. *ArXiv e-prints*, November 2015.

Issei Sato and Hiroshi Nakagawa. Approximation analysis of stochastic gradient langevin dynamics by using fokker-planck equation and ito process. In Eric P. Xing and Tony Jebara (eds.), *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pp. 982–990, Bejing, China, 22–24 Jun 2014. PMLR. URL `http://proceedings.mlr.press/v32/satoa14.html`.

Xiaocheng Shang, Zhanxing Zhu, Benedict Leimkuhler, and Amos J Storkey. Covariance-controlled adaptive Langevin thermostat for large-scale Bayesian sampling. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 28*, pp. 37–45, 2015. URL `http://papers.nips.cc/paper/5978-covariance-controlled-adaptive-langevin-thermostat-for-large-scale-bayesian-s` `pdf`.

N. Shirish Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang. On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima. *ArXiv e-prints*, September 2016.

Ravid Shwartz-Ziv and Naftali Tishby. Opening the black box of deep neural networks via information. *CoRR*, abs/1703.00810, 2017. URL `http://arxiv.org/abs/1703.00810`.

L. N. Smith. Cyclical Learning Rates for Training Neural Networks. *ArXiv e-prints*, June 2015.

S. J. Vollmer, K. C. Zygalakis, and Y. W. Teh. (Non-) asymptotic properties of stochastic gradient Langevin dynamics. *arXiv preprint arXiv:1501.00438*, 2015.

M. Welling and Y. W. Teh. Bayesian learning via stochastic gradient Langevin dynamics. In *Proceedings of the 28th International Conference on Machine Learning*, pp. 681–688, 2011.

Lei Wu, Zhanxing Zhu, et al. Towards understanding generalization of deep learning: Perspective of loss landscapes. *arXiv preprint arXiv:1706.10239*, 2017.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016.

APPENDIX

A DERIVATION OF THE FOKKER-PLANCK EQUATION

In this appendix we derive the Fokker-Planck equation. For any stochastic differential equation, since the evolution is noisy we can't say exactly where in parameter space the parameter values will be at any given time. But we can talk about the probability $P(\boldsymbol{\theta}, t|\boldsymbol{\theta}_0, t_0)$ that a parameter takes a certain value $\boldsymbol{\theta}$ at a certain time t, given that it started at $\boldsymbol{\theta}_0, t_0$. That is captured by the Fokker-Planck equation, which reads

$$\frac{\partial P(\boldsymbol{\theta}, t)}{\partial t} = \nabla_{\boldsymbol{\theta}} \cdot \left[\eta \boldsymbol{g}(\boldsymbol{\theta}) P(\boldsymbol{\theta}, t) + \frac{\eta^2}{2S} \nabla_{\boldsymbol{\theta}} \cdot [\mathbf{C}(\boldsymbol{\theta}) P(\boldsymbol{\theta}, t)] \right]. \tag{5}$$

In this appendix we will derive the above equation from the stochastic differential equation (6). We will not be interested in pure mathematical rigour, but intend the proof to add intuition for the machine learning audience.

For brevity we will sometimes write the probability just as $P(\boldsymbol{\theta}, t)$. We will sometimes make use of tensor index notation, where a tensor is denoted by its components, (for example θ_i are the components of the vector $\boldsymbol{\theta}$), and we will use the index summation convention, where a repeated index is to be summed over.

We start with the stochastic differential equation

$$\frac{d\boldsymbol{\theta}}{dt} = -\eta \boldsymbol{g}(\boldsymbol{\theta}) + \frac{\eta}{\sqrt{S}} \mathbf{B}(\boldsymbol{\theta}) \mathbf{f}(t). \tag{6}$$

A formal expression for the probability for a given noise function $\mathbf{f}$ is given by $P(\boldsymbol{\theta}, t) = \delta(\boldsymbol{\theta} - \boldsymbol{\theta}_{\mathbf{f}})$, but since we don't know the noise, we instead average over all possible noise functions to get

$$P(\boldsymbol{\theta}, t) = \mathbb{E}[\delta(\boldsymbol{\theta} - \boldsymbol{\theta}_{\mathbf{f}})]. \tag{7}$$

While this is just a formal solution we will make use of it later in the derivation.

We now consider a small step in time δt, working at first order in δt, and ask how far the parameters move, denoted $\delta \boldsymbol{\theta}$, which is given by integrating (6)

$$\delta \boldsymbol{\theta} = -\eta \boldsymbol{g} \delta t + \frac{\eta}{\sqrt{S}} \mathbf{B} \int_t^{t+\delta t} \mathbf{f}(t') dt' \tag{8}$$

where we've assumed that $\delta\boldsymbol{\theta}$ is small enough that we can evaluate $\boldsymbol{g}$ at the original $\boldsymbol{\theta}$. We now look at expectations of $\delta\boldsymbol{\theta}$. Using the fact that the noise is normalized Gaussian, $\mathbb{E}(\mathbf{f}(t)) = 0$, we get (switching to index notation for clarity)

$$\mathbb{E}(\delta\theta_i) = -\eta g_i \delta t \tag{9}$$

and using that the noise is normalized Gaussian again, we have that $\mathbb{E}(\mathbf{f}(t)\mathbf{f}(t')) = \mathbf{I}\delta(t-t')$, leading to

$$\mathbb{E}(\delta\theta_i \delta\theta_j) = \frac{\eta^2}{S} C_{ij}\delta t + \mathcal{O}(\delta t^2). \tag{10}$$

If at time t we are at position $\boldsymbol{\theta}'$ and end up at position $\boldsymbol{\theta} = \boldsymbol{\theta}' + \delta\boldsymbol{\theta}$ at time $t + \delta t$, then we can take (7) with $\boldsymbol{\theta}_{\mathbf{f}} = \boldsymbol{\theta}'$ and Taylor expand it in $\delta\theta_i$

$$P(\boldsymbol{\theta}, t+\delta t|\boldsymbol{\theta}', t) = \left(1 + \mathbb{E}(\delta\theta_i)\frac{\partial}{\partial\theta'_i} + \frac{1}{2}\mathbb{E}(\delta\theta_i\delta\theta_j)\frac{\partial^2}{\partial\theta'_i\partial\theta'_j} + \mathcal{O}(\delta t^2)\right)\delta(\boldsymbol{\theta} - \boldsymbol{\theta}'). \tag{11}$$

We deal with the derivatives of the delta functions in the following way. We will use the following identity, called the Chapman-Kolmogorov equation, which reads

$$P(\boldsymbol{\theta}, t+\delta t|\boldsymbol{\theta}_0, t_0) = \int_{-\infty}^{+\infty} d\boldsymbol{\theta}' P(\boldsymbol{\theta}, t+\delta t|\boldsymbol{\theta}', t')P(\boldsymbol{\theta}', t'|\boldsymbol{\theta}_0, t_0) \tag{12}$$

for any t', such that $t_0 \leq t' \leq t + \delta t$. This identity is an integral version of the chain rule of probability, stating there are multiple paths of getting from an initial position $\boldsymbol{\theta}_0$ to $\boldsymbol{\theta}$ at time $t + \delta t$ and one should sum all of these paths.

We will now substitute (11) into the first term on the right hand side of (12) with t' set to be t, and apply integration by parts (assuming vanishing boundary conditions at infinity) to move the derivatives off of the delta functions and onto the other terms. We end up with

$$\begin{aligned} P(\boldsymbol{\theta}, t+\delta t|\boldsymbol{\theta}', t) = P(\boldsymbol{\theta}, t|\boldsymbol{\theta}_0, t_0) &- \frac{\partial}{\partial\theta_i}\left(\mathbb{E}(\delta\theta_i)P(\boldsymbol{\theta}, t|\boldsymbol{\theta}_0, t_0)\right) \\ &+ \frac{1}{2}\frac{\partial^2}{\partial\theta'_i\partial\theta'_j}\left(\mathbb{E}(\delta\theta_i\delta\theta_j)P(\boldsymbol{\theta}, t|\boldsymbol{\theta}_0, t_0)\right). \end{aligned} \tag{13}$$

We can then take the first term on the right hand side of (13) to the other side, insert (9) and (10), divide by δt and take the limit $\delta t \to 0$, getting a partial derivative with respect to time on the left hand side, leading directly to the Fokker-Planck equation quoted in the text (5) (where we have reverted back to vector notation from index notation for conciseness)

$$\frac{\partial P(\boldsymbol{\theta}, t)}{\partial t} = \nabla_{\boldsymbol{\theta}} \cdot \left[\eta \boldsymbol{g}(\boldsymbol{\theta})P(\boldsymbol{\theta}, t) + \frac{\eta^2}{2S}\nabla_{\boldsymbol{\theta}} \cdot [\mathbf{C}(\boldsymbol{\theta})P(\boldsymbol{\theta}, t)]\right]. \tag{14}$$

B Intuitions from Fokker-Planck

In this appendix we will give some supplementary comments about the intuition we can gain from the Fokker-Planck equation (5). If the learning rates, batch size and covariance all change slowly, and the covariance is proportional to the identity $\mathbf{C} = \sigma^2\mathbf{I}$ then we can rewrite the Fokker-Planck equation in the following form

$$\frac{\partial P(\boldsymbol{\theta}, t_\eta)}{\partial t_\eta} = \nabla_{\boldsymbol{\theta}} \cdot [\mathbf{g}(\boldsymbol{\theta})P(\boldsymbol{\theta}, t_\eta)] + \frac{\eta\sigma^2}{2S}\nabla^2_{\boldsymbol{\theta}} P(\boldsymbol{\theta}, t_\eta)\,, \tag{15}$$

where we have rescaled the time coordinate to be $t_\eta \equiv t\eta$. One can now see that in terms of this rescaled time coordinate, the ratio between the drift and diffusion terms is governed by the following quantity

$$n \equiv \frac{\eta\sigma^2}{2S} \tag{16}$$

which we call the noise. In terms of the balance between drift and diffusion, we see that higher n gives rise to a more diffusive evolution, while a lower n allows the potential drift term to dominate. In the next section we will see how n controls the stationary distribution at which SGD converges to. For now we highlight that in terms of this rescaled coordinate, only n controls the evolution towards this stationary distribution (not just its endpoint), in terms of the rescaled time t_η. That is, learning rate and batch size are interchangable in the sense that n is invariant under transformations $S \to aS$, $\eta \to \eta/a$, for $a > 0$. But note that the time it takes to reach the stationary distribution depends on η as well, because of the rescaled time variable. For example, for a higher learning rate, but constant n, one arrives at the same stationary distribution, but in a quicker time by a factor of $1/\eta$.

However, a caution here is necessary. The first order SGD update equation only holds for small enough η that a first order approximation to a Taylor expansion is valid, and hence we expect the first order approximation of SGD as a continuous stochastic differential equation to break down for high η. Thus, we expect learning rate and batch size to be interchangable up to a maximum value of η at which the approximation breaks.

C STATIONARY DISTRIBUTION

In this appendix we will prove equation (2) quoted in the main text, which claims that the stationary (ie late time, independent of time) solution of the Fokker-Planck equation has the following form

$$P(\boldsymbol{\theta}) = P_0 e^{-L(\boldsymbol{\theta})/n}. \tag{17}$$

where $n \equiv \frac{\sigma^2 \eta}{2S}$, under the assumptions that the covariance is proportional to the identity $\mathbf{C}(\boldsymbol{\theta}) = \sigma^2(\boldsymbol{\theta})\mathbf{I}$ and further that $|\nabla_{\boldsymbol{\theta}}(\sigma^2)| \ll 2S|\boldsymbol{g}|\eta$. Here P_0 is a normalization constant.

In order to prove this is the stationary distribution, we need to solve the Fokker-Planck equation (5) with the left hand side set equal to zero (since we are looking for the stationary solution). To do this, we begin by writing the Fokker-Planck equation in a slightly different form, making use of a probability current $\boldsymbol{J}$, defined

$$\boldsymbol{J} \equiv \eta \boldsymbol{g}(\boldsymbol{\theta}) P(\boldsymbol{\theta}, t) + \frac{\eta^2}{2S} \nabla_{\boldsymbol{\theta}} \cdot (\mathbf{C}(\boldsymbol{\theta}) P(\boldsymbol{\theta}, t)) \tag{18}$$

in which the Fokker-Planck equation becomes

$$\frac{\partial P(\boldsymbol{\theta}, t)}{\partial t} = \nabla_{\boldsymbol{\theta}} \cdot \boldsymbol{J}. \tag{19}$$

At this point we use the assumptions that $\mathbf{C}(\boldsymbol{\theta}) = \sigma^2(\boldsymbol{\theta})\mathbf{I}$ and that $|\nabla_{\boldsymbol{\theta}}(\sigma^2)| \ll 2S|\boldsymbol{g}|\eta$ in order to bring $\mathbf{C}(\boldsymbol{\theta})$ outside of the divergence in (18) and turn the divergence into a gradient of P. We get

$$\boldsymbol{J} = \eta \boldsymbol{g}(\boldsymbol{\theta}) P(\boldsymbol{\theta}, t) + \frac{\eta^2 \sigma^2}{2S} \nabla_{\boldsymbol{\theta}} P(\boldsymbol{\theta}, t) \tag{20}$$

Now we notice we can write this in the following way

$$\boldsymbol{J} = \eta n e^{-L(\boldsymbol{\theta})/n} \nabla_{\boldsymbol{\theta}} \left(e^{L(\boldsymbol{\theta})/n} P \right) \tag{21}$$

where we've defined $n = \frac{\eta \sigma^2}{2S}$. Then we consider that $\nabla_{\boldsymbol{\theta}} \cdot \boldsymbol{J} = 0$ and hence

$$\nabla_{\boldsymbol{\theta}} \cdot \left[\eta n e^{-L(\boldsymbol{\theta})/n} \nabla_{\boldsymbol{\theta}} \left(e^{L(\boldsymbol{\theta})/n} P \right) \right] = 0. \tag{22}$$

This equation has the rather easy to spot solution, simply

$$P(\boldsymbol{\theta}) = P_0 e^{-L(\boldsymbol{\theta})/n} \tag{23}$$

which is the desired stationary solution we intended to find.

D DERIVATION OF PROBABILITY OF ENDING IN A GIVEN MINIMA

In this appendix we derive the discrete set of probabilities of ending at each minima, as given in (3). We work locally near $\boldsymbol{\theta}_A$ and take the following approximation of the loss function, since it is near a minimum,

$$L(\boldsymbol{\theta}) \approx L_A + \frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}_A)^\top \mathbf{H}_A (\boldsymbol{\theta} - \boldsymbol{\theta}_A). \tag{24}$$

where $\mathbf{H}_A$ is the Hessian, and is positive definite for a minimum.

The distribution $P_A(\boldsymbol{\theta})$ is a probability density, while we are interested in the discrete set of probabilities of ending at a given minimum, which we will denote by lowercase p_A. To calculate this discrete set of probabilities, for each minimum we need to integrate this stationary distribution over an interval containing the minimum. We take σ^2 to take the constant value σ_A^2 within this minimum and hence denote n by n_A. Integrating (2) in some region R_A around $\boldsymbol{\theta}_A$ and using (24)

$$p_A \approx P_0 \int_{R_A} \exp\left(-\frac{1}{n_A} L(\boldsymbol{\theta})\right) \tag{25}$$

$$\approx P_0 \int_{R_A} \exp\left(-\frac{1}{n_A}\left[L_A + \frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}_A)^\top \mathbf{H}_A (\boldsymbol{\theta} - \boldsymbol{\theta}_A)\right]\right) \tag{26}$$

$$\approx P_0 \exp\left(-\frac{L_A}{n_A}\right) \sqrt{(2\pi n_A)^p \frac{1}{\det \mathbf{H}_A}} \tag{27}$$

where in the last line we assume the region is large enough that an approximation to the full Gaussian integral can be used (i.e. that the tails don't contribute, which is fine as long as the region is sufficiently larger than $\det \mathbf{H}_A$) – note that the region can't be too large, otherwise we would invalidate our local assumption. The picture is that the minima are sufficiently far apart that the region can be taken sufficiently large for this approximation to be valid.

Since we are interested in relative probabilities between different minima, so we can consider the unnormalized probability, dropping factors that are common amongst all minima we get

$$\tilde{p}_A = \sqrt{\frac{n_A^p}{\det \mathbf{H}_A}} \exp\left(-\frac{L_A}{n_A}\right) \tag{28}$$

This is the required expression given in the main text.

As an aside, we note that the derivation used above talked about true minima, with a positive definite Hessian, i.e. a minima in all directions. In the real world application to deep neural networks with a large number of parameters, it is unrealistic to expect the endpoint of training to be in a true minimum. Instead, it is more likely to be a point in which the Hessian has some positive eigenvalues λ_{Ai} in just a few directions, say $d \ll q$, with the other eigenvalues (approximately) zero. In this case, the relative probabilities are given by

$$\tilde{p}_A = \sqrt{\frac{(2\pi n_A)^d}{\lambda_{A1} \cdots \lambda_{Ad}}} \exp\left(-\frac{L_A}{n_A}\right). \tag{29}$$

In the main paper we only consider the simplest case of true minima, though much of what we say can be extended to the case of points with some positive Hessian eigenvalues with some flat directions.

E FURTHER SPECIAL CASES

In this appendix we explore other special cases in which the ratio of probabilities for ending in one or another minima simplifies, allowing for an intuitive picture of the way SGD favours some minima over others. We begin this appendix with the special case of equal height minima, and then look at equal width minima.

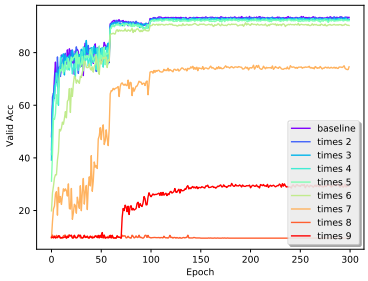


Figure 7: Experiments involving Resnet56 architecture on CIFAR10 dataset. In each curve, we multiply the $\frac{\eta}{S}$ ratio by a given factor (increasing both batch size and learning rate). We observe that multiplying the ratio by a factor up to 5 results in similar performances. However, the performances degrades for factor superior to 5.

E.1 Equal Height Minima

Consider now the case where the minima are the same height $L_A = L = L_B$. The simplest case is to look at equal width minima, in which case A is favoured over B if and only if $n_A > n_B$, with the relation being exponential – ie a noisier minima is exponentially more probable than a less noisy one.

If on the other hand the minima are not equal width, but do have equal height, we can without loss of generality take $\det \mathbf{H}_B > \det \mathbf{H}_A$. We find that $p_A > p_B$ if and only if the following inequality is satisfied

$$\log\left(\sqrt{\frac{n_A^p}{n_B^p}}\right) > L\left(\frac{1}{n_A} - \frac{1}{n_B}\right) + \log\left(\frac{\det \mathbf{H}_B}{\det \mathbf{H}_A}\right) > L\left(\frac{1}{n_A} - \frac{1}{n_B}\right) \tag{30}$$

This places an interesting constraint on the noises and the height L.

E.2 Equal Width Minima

The case of equal width minima has been covered in the previous sections, but with either equal width or equal height presupposed. Here, if we don't assume the noise or height are equal, but that the widths are the same, then $p_A > p_B$ if and only if

$$\sqrt{n_A^p} e^{-\frac{L_A}{n_A}} > \sqrt{n_B^p} e^{-\frac{L_B}{n_B}}. \tag{31}$$

It is not clear how to simplify this further, but one can see that if $L_A < L_B$ then this inequality will be satisfied for most reasonable values of n_A and n_B, unless n_B is exponentially larger than n_A.

F Limitation of our Analysis

Due to our assumptions, we expect the theory to become unreliable when the discrete to continuous approximation fails, when the variance in gradients changes as fast as the loss, when the covariance in gradients cannot be written as proportional to the identity matrix, the finite size of the training set and when momentum is considered.

We discussed limit of the discrete to continous approximation in 4.4 (further illustrated in 7).

When the gradient variance changes rapidly compared to the loss, there will be a different, more complicated stationary solution that arises when solving the stationary Fokker-Planck equation. We expect this to happen in certain datasets and in using certain architectures, though it is hard to say an exact relation between datasets, architectures and gradient variances.

When the covariance in gradients can't be written as proportional to the identity matrix, the stationary solution to the Fokker-Planck equation is the solution to a complicated partial differential

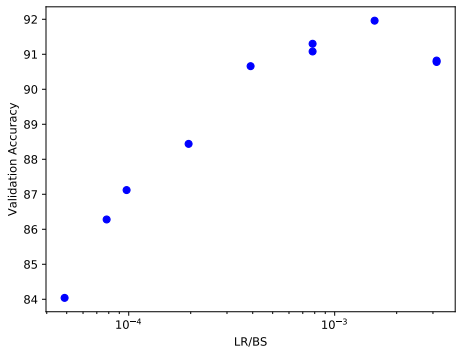


Figure 8: Validation accuracy of Resnet56 networks against different ratios learning rate (LR) to batch size (BS) on CIFAR10. Trained with naive SGD.

equation, and one can't easily spot the solution by inspection as we do in Appendix C. We expect this approximation to break down especially in the case of complicated architectures where different gradient directions will be have very different gradient covariances.

Our theory does not involve the finite size of the training set, which is a drawback of the theory. This may be especially apparent when the batch size becomes large, compared to the training set size, and we expect the theory to break down at this point.

Finally, we mention that momentum is used in practical deep learning optimization algorithms. Our theory does not consider momentum, which is a drawback of the theory and we expect it to break down in models in which momentum is important. We can write a Langevin equation for the case of momentum, with momentum damping coefficient μ,

$$\frac{d\boldsymbol{v}}{dt} = -\mu\boldsymbol{v} - \eta\boldsymbol{g} + \frac{\eta\boldsymbol{B}}{\sqrt{S}}\boldsymbol{f}(t)$$

where the velocity is $\boldsymbol{v} = d\boldsymbol{\theta}/dt$.

The Fokker-Planck equation corresponding to this Langevin equation with momentum is

$$\left(\frac{\partial}{\partial t} + \boldsymbol{v}\cdot\nabla_{\boldsymbol{\theta}}\right)P = \eta\nabla_{\boldsymbol{\theta}}\cdot\left(\frac{\mu}{\eta}\boldsymbol{v}P + \boldsymbol{g}P\right) + \frac{\eta^2\sigma^2}{2S}\nabla^2_{\boldsymbol{\theta}}P$$

where P is now a function of velocity $\boldsymbol{v}$ as well as $\boldsymbol{\theta}$ and t, and we've again assumed the gradient covariance varies slowly compared to the loss.

From this more complicated Fokker-Planck equation, it is hard to spot the stationary distribution. We leave the study of this for further work. For now, we just note that a factor of η can be taken from the right hand side, and then the ratio between the diffusion and drift terms will again be the noise $n = \eta\sigma^2/(2S)$

G Other Empirical Results

G.1 More SGD Impacts on Minima

Figure 8 reports the validation accuracy for Resnet models trained on CIFAR with different LR/BS ratio. Figure 9 reports 20 layer ReLU experiment variant with scaled initialization by 0.7 , which explores case in which model is driven to bad local minima (and is not addressed by our theory).

G.2 More Learning Rate & Batch Size Exchangeability

In this appendix we show in more detail the experiments of Section 4.2 which show the exchangeability of learning rate and batch size. In Fig. 10 we show the log cross entropy loss, the epoch-averaged cross entropy loss, train, validation and test accuracy as well as the batch size schedule,

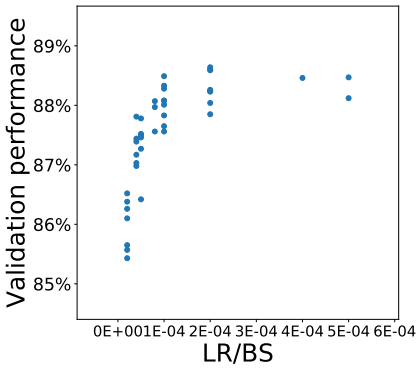


Figure 9: Correlation of $\frac{\eta}{S}$ with validation accuracy for 20 layer ReLU network on Fashion MNIST with bad initialization.

learning rate schedule. We see in the lr/bs plot that the orange and blue lines (cyclic schedules) have the same ratio of lr/bs as each other throughout, and that their dynamics are very similar to each other. The same holds for the green and red lines, of constant batch size and learning rate. This supports the theoretical result of this paper, that the ratio S/η governs the stationary distribution of SGD.

We note that it is not just in the stationary distribution that the exchangeability holds in these plots - it appears throughout training, highlighted especially in the cyclic schedules. We postulate that due to the scaling relation in the Fokker-Planck equation, the exchangeability holds throughout learning as well as just at the end, as long as the learning rate does not get so high as to ruin the approximations under which the Fokker-Planck equation holds.

G.3 Details on memorization experiment

We report learning curves from memorization experiment with 0.0 momentum, see Fig. 11. We additionally confirm similar result as in previous experiments and show correlation between batch size and learning rate ratio and norm of Hessian, see Fig. 12.

G.4 Cyclic batch and learning schedule

We report sharpness and accuracy evolution for learning schedules discussed in Sec. 4.5, see Fig. 13.

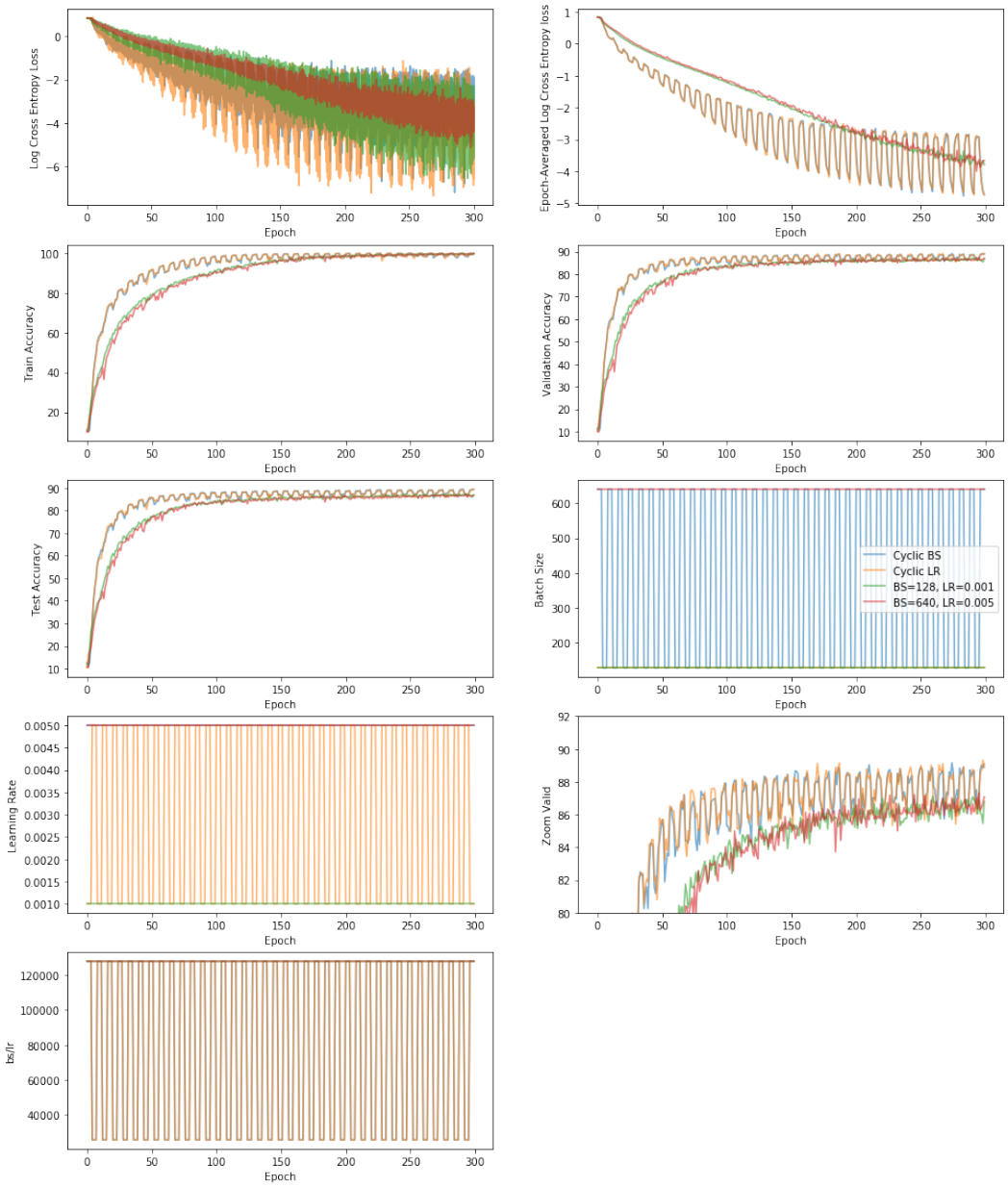


Figure 10: We show in more detail exchangeability of batch size and learning rate in a one-to-one ratio. In blue, cyclic batch size schedule between size 128 and 640 and fixed learning rate 0.005, is exchangeable with orange cyclic learning rate schedule between learning rates 0.001 and 0.005 with fixed batch size 128. In green, constant batch size 640 and constant learning rate 0.005 is exchangeable with, in red, constant batch size 128 and constant learning rate 0.005.

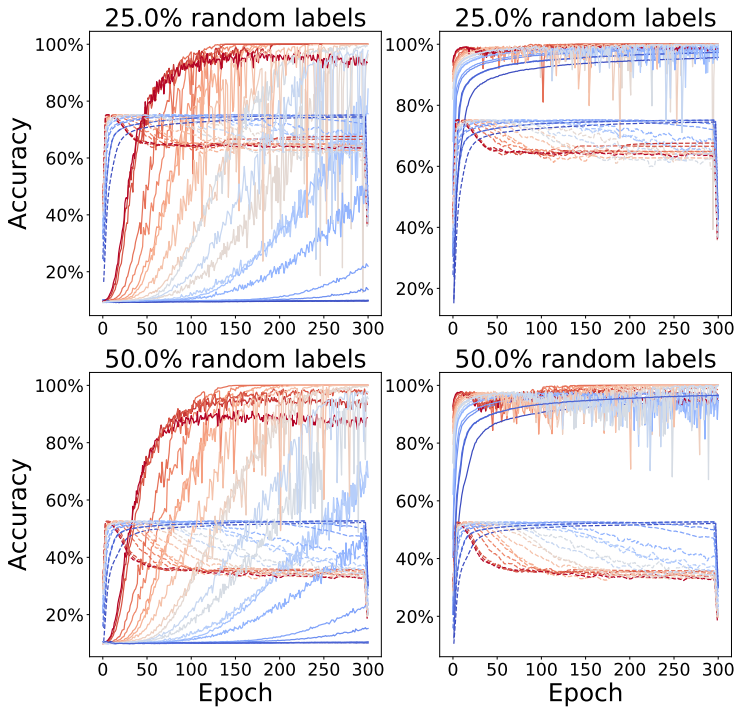


Figure 11: Learning curves for memorization experiment with momentum 0.0. Solid lines represent training accuracy, dotted validation accuracy. Warm color indicates higher $\frac{\eta}{S}$ ratio.

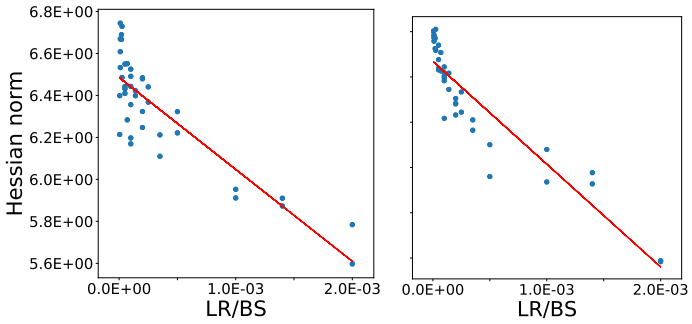


Figure 12: Correlation between (approximate) norm of Hessian of best validation minima and LR/BS for 0.0 (left) and 0.9 (right) momentum.

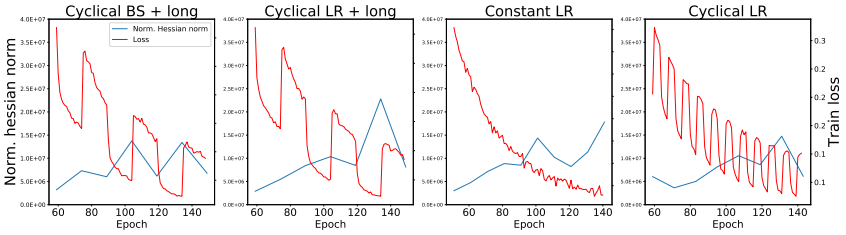


Figure 13: Cyclical learning rate switches between sharp/deep and wide/shallow minima if stage of increased noise are long enough to allow for mixing. Plots from left to right: cyclical BS with long stage (15 epochs), cyclical LR with long stage, constant learning rate, cyclical LR with short stage (5 epochs).