The Effect of Network Width on Stochastic Gradient Descent and Generalization: an Empirical Study

Daniel S. Park [1 2] **Jascha Sohl-Dickstein** [1] **Quoc V. Le** [1] **Samuel L. Smith** [3]

Abstract

We investigate how the final parameters found by stochastic gradient descent are influenced by over-parameterization. We generate families of models by increasing the number of channels in a base network, and then perform a large hyper-parameter search to study how the test error depends on learning rate, batch size, and network width. We find that the optimal SGD hyper-parameters are determined by a "normalized noise scale," which is a function of the batch size, learning rate, and initialization conditions. In the absence of batch normalization, the optimal normalized noise scale is directly proportional to width. Wider networks, with their higher optimal noise scale, also achieve higher test accuracy. These observations hold for MLPs, ConvNets, and ResNets, and for two different parameterization schemes ("Standard" and "NTK"). We observe a similar trend with batch normalization for ResNets. Surprisingly, since the largest stable learning rate is bounded, the largest batch size consistent with the optimal normalized noise scale decreases as the width increases.

1. Introduction

Generalization is a fundamental concept in machine learning, but it remains poorly understood (Zhang et al., 2016). Theoretical generalization bounds are usually too loose for practical tasks (Harvey et al., 2017; Neyshabur et al., 2017; Bartlett et al., 2017; Dziugaite & Roy, 2017; Zhou et al., 2018; Nagarajan & Kolter, 2018), and practical approaches to hyper-parameter optimization are often developed in an ad-hoc fashion (Sculley et al., 2018). A number of authors have observed that Stochastic Gradient Descent (SGD) can be a surprisingly effective regularizer (Keskar et al., 2016; Wilson et al., 2017; Sagun et al., 2017; Mandt et al., 2017; Smith & Le, 2017; Chaudhari & Soatto, 2017; Soudry et al., 2018). In this paper, we provide a rigorous empirical study of the relationship between generalization and SGD, which focuses on how both the optimal SGD hyper-parameters and the final test accuracy depend on the network width.

This is a broad topic, so we restrict the scope of our investigation to ensure we can collect thorough and unambiguous experimental results within a reasonable (though still substantial) compute budget. We consider training a variety of neural networks on classification tasks using SGD without learning rate decay ("constant SGD"), both with and without batch normalization (Ioffe & Szegedy, 2015). We define the performance of a network by its average test accuracy at "late times" and over multiple training runs. The set of optimal hyperparameters $\mathcal{H}_{\text{opt}}$ denote the hyper-parameters for which this average test accuracy was maximized. We stress that optimality is defined purely in terms of the performance of the trained network on the test set. This should be distinguished from references to ideal learning rates in the literature, which are often defined as the learning rate which converges fastest during training (LeCun et al., 1996; Karakida et al., 2017). Our use of optimality should also not be confused with references to optimality or criticality in some recent studies (Shallue et al., 2018; McCandlish et al., 2018) where these terms are defined with respect to efficiency of training, rather than final performance.

Given these definitions, we study the optimal hyper-parameters and final test accuracy of networks in the same class but with different widths. Two networks are in the same class if one can be obtained from the other by adjusting the numbers of channels. For example, all three-layer perceptrons are in the same class, while a two-layer perceptron is in a different class to a three-layer perceptron. For simplicity, we consider a "linear family" of networks,

$$\{\mathcal{N}_{w_1}, \cdots, \mathcal{N}_{w_k}\}, \tag{1}$$

each of which is obtained from a base network $\mathcal{N}_1$ by introducing a widening factor, much in the spirit of wide residual networks (Zagoruyko & Komodakis, 2016). That is, network $\mathcal{N}_w$ can be obtained from $\mathcal{N}_1$ by widening every layer by a constant factor of w. We aim to identify a predictive

[1]Google Brain, Mountain View, USA [2]Work done as a member of the Google AI Residency Program. [3]DeepMind, London, UK. Correspondence to: Daniel S. Park <danielspark@google.com>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019.

relationship between the optimal hyper-parameters $\mathcal{H}_{\text{opt}}$ and the widening factor w. We also seek to understand the relationship between network width and final test accuracy.

We will find that a crucial factor governing both relationships is the "normalized noise scale". As observed in (Mandt et al., 2017; Chaudhari & Soatto, 2017; Jastrzebski et al., 2017; Smith & Le, 2017), and reviewed in section 2.2, when the learning rate is sufficiently small, the behaviour of SGD is determined by the noise scale g, where for SGD,

$$g = \frac{\epsilon N_{\text{train}}}{B}, \tag{2}$$

and for SGD with momentum,

$$g = \frac{\epsilon N_{\text{train}}}{B(1-m)}. \tag{3}$$

Here, ϵ is the learning rate, B is the batch size, m is the momentum coefficient, and N_{train} is the size of the training set. Smith & Le (2017) showed that there is an optimal noise scale g_{opt}, and that any setting of the hyper-parameters satisfying $g = g_{\text{opt}}$ will achieve optimal performance at late times, so long as the effective learning rate $\epsilon/(1-m)$ is sufficiently small. We provide additional empirical evidence for this claim in section 4. However in this work we argue that to properly define the noise introduced by SGD, g should be divided by the square of a weight scale. A quick way to motivate this is through dimensional analysis. In a single SGD step, the parameter update is proportional to the learning rate multiplied by the gradient. Assigning the parameters units of [weight], and the loss units of [loss], the gradient has units of [loss]/[weight]. This implies that the learning rate has dimensions of $[\text{weight}]^2/[\text{loss}]$. The scale of the loss is controlled by the choice of cost function and the dataset. However the weight scale can vary substantially over different models in the same class. We hypothesize that this weight scale is controlled by the scale of the weights at initialization, and it will therefore depend on the choice of parameterization scheme (Jacot et al., 2018). Since the noise scale is proportional to the learning rate, we should therefore divide it by the square of this weight scale.

In this work we will consider two parameterization schemes, both defined in section 2.1. In the "standard" scheme most commonly used in deep learning, the weights are initialized from an isotropic Gaussian distribution whose standard deviation is inversely proportional to the square root of the network width. As detailed above, this work will consider families of networks obtained by multiplying the width of every hidden layer in some base network $\mathcal{N}_1$ by a multiplicative factor w. Thus the normalized noise scale,

$$\bar{g}(\mathcal{N}_w) = \frac{g}{(\sigma_0/\sqrt{w})^2} = \frac{gw}{\sigma_0^2} \quad \text{for standard scheme.} \tag{4}$$

The standard deviation σ_0 defines the weight scale of the base network, which for our purposes is just a constant. An alternative parameterization was recently proposed, commonly referred to as "Neural Tangent Kernel" parameterization, or "NTK" (van Laarhoven, 2017; Karras et al., 2017; Jacot et al., 2018; Karras et al., 2018). In this scheme, the weights are initialized from a Gaussian distribution whose standard deviation is constant, while the pre-activations are multiplied by the initialization factor (Glorot & Bengio, 2010; He et al., 2015) *after* applying the weights to the activations in the previous layer. Since the weight scale in this scheme is independent of the widening factor,

$$\bar{g} = g/\sigma_0^2 \quad \text{for NTK scheme.} \tag{5}$$

By finding the optimal normalized noise scale for families of wide residual networks (WRNs), convolutional networks (CNNs) and multi-layer perceptrons (MLPs) for image classification tasks on CIFAR-10 (Krizhevsky, 2009), Fashion-MNIST (F-MNIST) (Xiao et al., 2017) and MNIST (LeCun et al., 2010), we are able to make the following observations:

1. Without batch normalization, the optimal normalized noise is proportional to the widening factor. That is,

   $$\bar{g}_{\text{opt}}(\mathcal{N}_w) \propto w. \tag{6}$$

   See section 4 for plots. This result implies,

   - For the standard scheme, the optimal value of ϵ/B stays constant with respect to the widening factor.
   - For the NTK scheme, the optimal value of ϵ/B is proportional to the widening factor.

2. The definition of the noise scale does not apply to networks with batch normalization, since the gradients of individual examples depend on the rest of the batch. However we have observed that the trend expressed in equation 6 still holds in a weaker sense. Considering networks parameterized using the NTK scheme,

   - When the batch size is fixed, the optimal learning rate increases with the widening factor.
   - When the learning rate is fixed, the optimal batch size decreases with the widening factor.

Residual networks (He et al., 2016) obey the trend implied by equation 6 both with and without batch normalization. In all cases, both with and without batch normalization, wider networks consistently perform better on the test set (Neyshabur et al., 2018; Lee et al., 2018; Novak et al., 2018).

The largest stable learning rate is proportional to $1/w$ in the standard scheme, while it is constant for the NTK scheme (discussed further in section 2.1). This implies that the largest batch size consistent with equation 6 decreases as the network width rises. Since the batch size cannot be smaller than one, these bounds imply that there is a critical network width above which equation 6 cannot be satisfied.

The paper is structured as follows. In section 2, we review the background material and introduce our notation. In section 3 we describe how the experiments were performed, while the empirical results are presented in section 4. In section 5 we discuss our findings and their implications.

2. Background

2.1. Standard vs. NTK Parameterization Schemes

In the standard scheme, the pre-activations z_i^{l+1} of layer $(l+1)$ are related to the activations y_j^l of layer l by

$$z_i^{l+1} = \sum_{j=1}^{n_l} W_{ij}^l y_j^l + b_i^{l+1}, \tag{7}$$

and weights and biases are initialized according to

$$W_{ij}^l \sim \mathcal{N}\left(0,\ \frac{\sigma_0^2}{n_l}\right), \quad b_i^{l+1} = 0. \tag{8}$$

The scalar n_l denotes the input dimension of the weight matrix, and σ_0^2 is a common weight scale shared across all models in the same class. For fully connected layers n_l is the dimension of the input, while for convolutional layers n_l is the filter size multiplied by the number of input channels. By inspecting equation 8, we can see that the weight scale is inversely proportional to the square root of the widening factor w. Following the discussion in the introduction, we arrive at equation 4 by normalizing the learning rate accordingly:

$$\bar{\epsilon}(\mathcal{N}_w) = \frac{\epsilon}{(\sigma_0/\sqrt{w})^2} = \frac{\epsilon w}{\sigma_0^2} \quad \text{for standard scheme.} \tag{9}$$

Meanwhile in the NTK scheme (van Laarhoven, 2017; Jacot et al., 2018), the pre-activations z_i^{l+1} are related to the activations y_j^l of the previous layer by,

$$z_i^{l+1} = \frac{1}{\sqrt{n_l}} \left(\sum_{j=1}^{n_l} W_{ij}^l y_j^l \right) + \beta_l b_i^{l+1}, \tag{10}$$

and weights and biases are initialized according to

$$W_{ij}^l \sim \mathcal{N}(0, \sigma_0^2), \quad b_i^{l+1} = 0. \tag{11}$$

Notice that the scaling factor $\frac{1}{\sqrt{n_l}}$ is introduced after applying the weights, while the parameter β_l controls the effect of bias. We set $\beta_l = 1/\sqrt{n_l}$ in all experiments. The weight scale is independent of the widening factor w, leading to a normalized learning rate which also does not depend on w,

$$\bar{\epsilon} = \epsilon/\sigma_0^2 \quad \text{for NTK scheme.} \tag{12}$$

We therefore arrive at the normalized noise scale of equation 5. The test set performance of NTK and standard networks are compared in section I of the supplemental material.

The learning rate has an upper-bound defined by convergence criteria and numerical stability. This upper-bound will also be on the order of the square of the weight scale, which implies that the upper-bound for $\bar{\epsilon}$ is approximately constant with respect to the widening factor. It follows that the stability bound for the bare learning rate ϵ scales like $1/w$ for the standard scheme (Karakida et al., 2017), while it remains constant for the NTK scheme. We provide empirical evidence supporting these stability bounds in section H of the supplementary material. A major advantage of the NTK parameterization is that we can fix a single learning rate and use it to train an entire family of networks $\{\mathcal{N}_w\}$ without encountering numerical instabilities. We therefore run the bulk of our experiments using the NTK scheme.

2.2. Noise in SGD

Smith & Le (2017) showed that for SGD and SGD with momentum, if the effective learning rate is sufficiently small the dynamics of SGD are controlled solely by the noise scale (equations 2 and 3). This implies that the set of hyperparameters for which the network achieves maximal performance at "late times" is well approximated by a level set of g (i.e., the set of hyperparameters for which $g = g_{\text{opt}}$). We define "late times" to mean sufficiently long for the validation accuracy to equilibrate. To verify this claim, in our experiments we will make two independent measurements of g_{opt}. One is obtained by holding the learning rate fixed and sweeping over the batch size, while the other is obtained by holding the batch size fixed and sweeping over the learning rate. We find that these two measures of g_{opt} agree closely, and they obtain the same optimal test performance. We refer the reader to section 3.2 for further discussion of training time, and section G of the supplementary material for experiments comparing the test set performance of an MLP across a two dimensional grid of learning rates and batch sizes.

This analysis breaks down when the learning rate is too large (Yaida, 2018). However empirically for typical batch sizes (e.g., $B \lesssim 1000$ on ImageNet), the optimal learning rate which maximizes the test set accuracy is within the range where the noise scale holds[1] (Goyal et al., 2017; Smith et al., 2017; McCandlish et al., 2018; Shallue et al., 2018). Our experiments will demonstrate that this does not contradict the common observation that, at fixed batch size, the test set accuracy drops rapidly above the optimal learning rate.

When a network is trained with batch normalization, the gradients for individual samples depend on the rest of the batch, breaking the analysis of Smith & Le (2017). Batch normalization also changes the gradient scale. We therefore do not expect equation 6 to hold when batch normalization is introduced. However we note that at fixed batch size, the SGD noise scale is still proportional to the learning rate.

[1] i.e., linear scaling of ϵ and B does not degrade performance.

3. Experiments

3.1. Overview

We run experiments by taking a linear family of networks, and finding the optimal normalized noise scale for each network on a given task. We measure the optimal noise scale in two independent ways—we either fix the learning rate and vary the batch size, or fix the batch size and vary the learning rate. We use fixed Nesterov momentum $m = 0.9$.

We first describe our experiments at fixed learning rate. We train 20 randomly initialized networks for each model in the family at a range of batch sizes, and compute the test accuracy after training "sufficiently long" (section 3.2). We then compute the average trained network performance and find the batch size B with the best average performance μ_B. The "trained performance" refers to the average test accuracy of "trained runs" (runs whose final test accuracy exceeds 0.2). We compute the standard deviation σ_B of the trained accuracy at this batch size and find all contiguous batch sizes to B whose average accuracy is above $\mu_B - 2\sigma_B/\sqrt{n_B}$, where n_B is the number of trained runs at batch size B. This procedure selects all batch sizes whose average accuracy is within two standard error deviations of μ_B, and it defines the "best batch size interval" $[B_0, B_1]$, from which we compute the "best normalized noise scale interval" $[\bar{g}_1, \bar{g}_0]$. We estimate the optimal normalized noise scale by $\bar{g}_{\text{opt}} = (\bar{g}_0 + \bar{g}_1)/2$. When $B_0 \neq B_1$, we include an error bar to indicate the range. The procedure for computing the optimal normalized noise scale in experiments with fixed batch size B is analogous to the procedure above; we train all networks 20 times for a range of learning rates and compute the best learning rate interval $[\epsilon_0, \epsilon_1]$.

Our main result is obtained by plotting the optimal normalized noise scale $\bar{g}_{\text{opt}}(\mathcal{N}_w)$ against the widening factor w (in the absence of batch normalization). When batch normalization is introduced, the definition of the noise scale is not valid. In this case, we simply report the optimal inverse batch size (learning rate) observed when fixing the learning rate (batch size), respectively. To make the plots comparable, we rerun the estimation procedure used for finding the optimal value of $\bar{g}$ when estimating the optimal value of $1/B$ (batch size search) and ϵ (learning rate search).

3.2. Training Time

To probe the asymptotic "late time" behaviour of SGD, we run our experiments with a very large compute budget, where we enforce a lower bound on the training time both in terms of the number of training epochs and the number of parameter updates. When we run learning rate searches with fixed batch size, we take a reference learning rate, for which the training steps are computed based on the epoch/step constraints, and then scale this reference training time accordingly for different learning rates. Although we find consistent relationships between the batch size, learning rate, and test error, it is still possible that our experiments are not probing asymptotic behavior (Shallue et al., 2018).

We terminate early any training run whose test accuracy falls below 0.2 at any time t beyond 20% of the total training time T. We verified that at least 15 training runs completed successfully for each experiment (learning rate/batch size pair). See sections B and C of the supplementary material for a detailed description of the procedure used to set training steps and impose lower bounds on training time.

3.3. Networks and Datasets

We consider three classes of networks; multi-layer perceptrons, convolutional neural networks and residual networks. We use ReLU nonlinearities in all networks, with softmax readout. The weights are initialized at criticality with $\sigma_0^2 = 2$ (He et al., 2015; Schoenholz et al., 2016).

We perform experiments on MLPs, CNNs and ResNets. We consider MLPs with 1, 2 or 3 hidden layers and denote the d-layered perceptron with uniform width w by the label $d\text{LP}_w$. Our family of convolutional networks CNN_w is obtained from the celebrated LeNet-5 (figure 2 of Lecun et al. (1998)) by scaling all the channels, as well as the fully connected layers, by a widening factor of $w/2$. Our family of residual networks WRN_w is obtained from table 1 of Zagoruyko & Komodakis (2016) by taking $N = 2$ and $k = w$. Batch normalization is only explored for CNNs and WRNs.

We train these networks for classification tasks on MNIST, Fashion-MNIST (F-MNIST), and CIFAR-10. More details about the networks and datasets used in the experiments can be found in section A of the supplementary material.

We train with a constant learning rate, and do not consider data augmentation, weight decay or other regularizers in the main text. Learning rate schedules and regularizers introduce additional hyper-parameters which would need to be tuned for every network width, minibatch size, and learning rate. This would have been impractical given our computational resources. However we selected a subset of common regularizers (label smoothing, data augmentation and dropout) and ran batch size search experiments for training WRNs on CIFAR-10 in the standard parameterization with commonly used hyper-parameter values. We found that equation 6 still held in these experiments, and increasing network width improved the test accuracy. These results can be found in section F of the supplementary material.

4. Experiment Results

Most of our experiments are run with networks parameterized in the NTK scheme without batch normalization. These

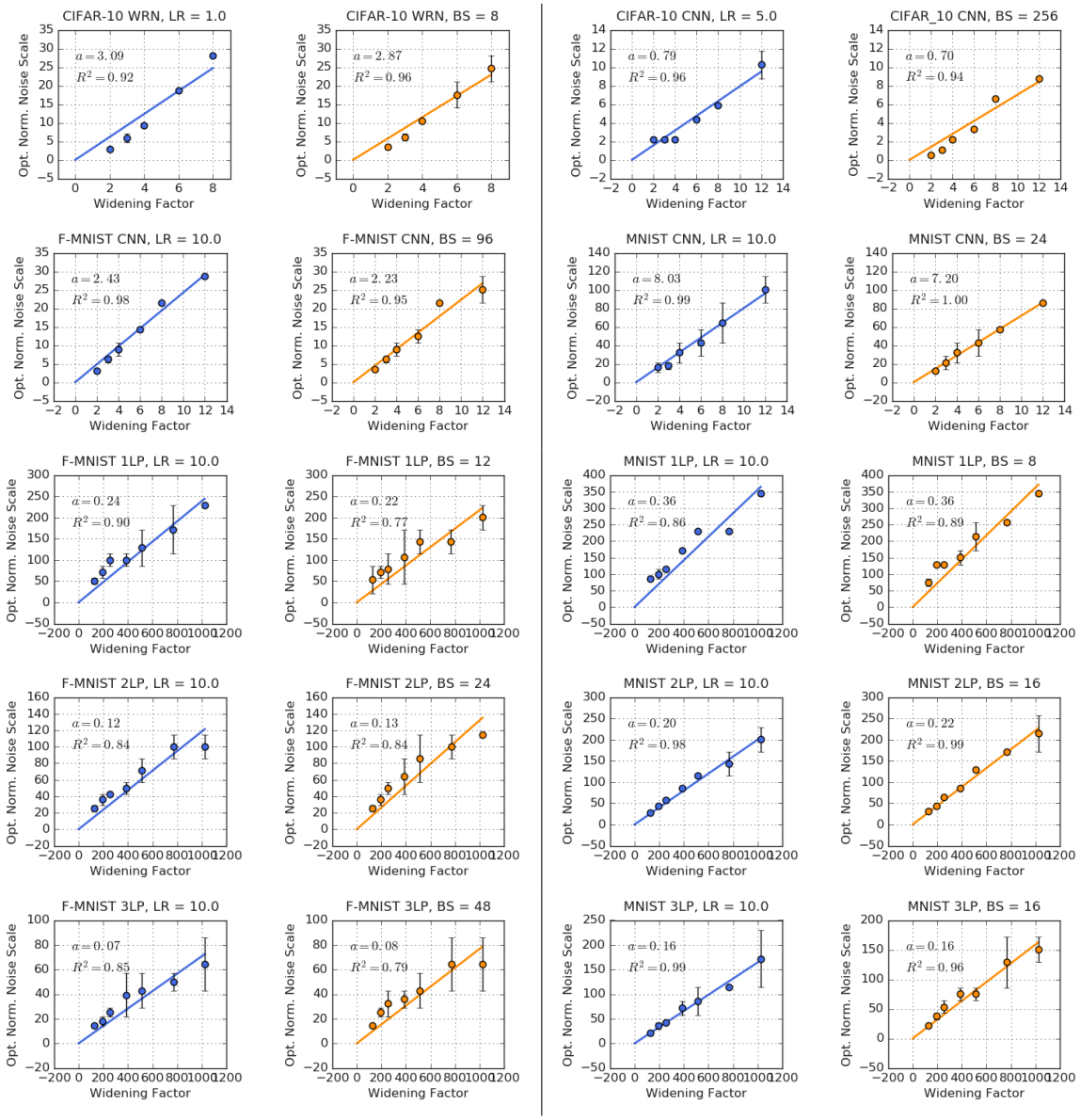

Figure 1. Optimal normalized noise scale vs. widening factor for NTK parameterized networks trained without batch normalization. The optimal normalized noise scale (y-axis) has units of 1000/[loss] while the widening factor (x-axis) is unitless. The proportionality constant a for each plot has the same units as the optimal normalized noise scale. This optimal normalized noise scale is obtained via batch size search for the blue plots, and via learning rate search for the orange plots. The fixed learning rate or batch size used to generate each plot is indicated in the title. Every dataset-network-experiment tuple exhibits a clear linear relationship as predicted by equation 6.

experiments, described in section 4.1, provide the strongest evidence for our main result, $\bar{g}_{\text{opt}}(\mathcal{N}_w) \propto w$ (equation 6). We independently optimize both the batch size at constant learning rate and the learning rate at constant batch size, and confirm that both procedures predict the same optimal normalized noise scale and achieve the same test accuracy.

We have conducted experiments on select dataset-network pairs with standard parameterization and without batch normalization in section 4.2. In this section we only perform batch size search at fixed learning rate. Finally we run experiments on NTK parameterized WRN and CNN networks with batch normalization in section 4.3, for which we perform both batch size search and learning rate search. Some additional batch size search experiments for WRNs parameterized in the standard scheme with batch normalization can be found in section E of the supplementary material, while we provide an empirical comparison of the test performance of standard and NTK parameterized networks in section I. We provide a limited set of experiments with additional regularization in section F of the supplementary material.

In section 4.4, we study how the final test accuracy depends on the network width and the normalized noise scale.

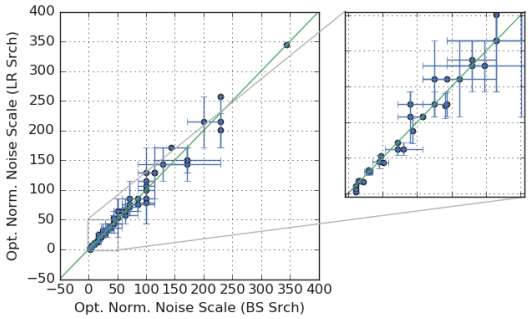

Figure 2. The optimal normalized noise scale obtained via learning rate search plotted against that obtained via batch size search for all experiments with NTK parameterized networks without batch normalization. The optimal normalized noise scale is given in units of 10^3/[loss]. The green line is the line $y = x$.

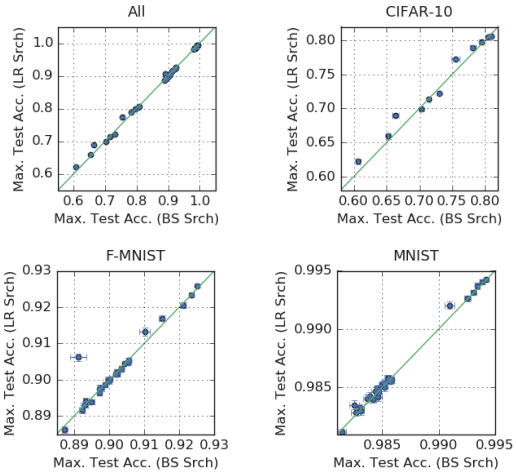

Figure 3. The maximum performance of networks obtained via learning rate search plotted against that obtained via batch size search for all experiments with NTK parameterized networks without batch normalization. The green line is the line $y = x$.

4.1. NTK without Batch Normalization

In figure 1, we plot the optimal normalized noise scale against the widening factor for a wide range of network families. The blue plots were obtained by batch size search with fixed learning rate, while the orange plots were obtained by learning rate search with fixed batch size. The fixed batch size or learning rate is given in the title of each plot alongside the dataset and network family. As explained in section 3.1, the error bars indicate the range of normalized noise scales that yield average test accuracy within the 95% confidence interval of the best average test accuracy. For each plot we fit the proportionality constant a to the equation $\bar{g}_{\text{opt}}(\mathcal{N}_w) = aw$, and provide both a and the R^2 value. We observe a good fit in each plot. The proportionality constant a is computed independently for each dataset-network family pair by both batch size search and learning rate search, and these two constants consistently agree well.

We can verify the validity of our assumption that the set of hyper-parameters yielding optimal performance is given by a level set of $\bar{g}$, by comparing both the optimal normalized noise scale $\bar{g}_{\text{opt}}$ and the maximum test set accuracy obtained by batch size search and learning rate search. The obtained values for both search methods for all experiments have been plotted against each other in figures 2 and 3. Further evidence for our assumption can be found in section G of the supplementary material. Finally, for each triplet of dataset-network-experiment type, we have plotted the test set accuracy against the scanned parameter (either batch size or learning rate) in figure 1 of section D.

4.2. Standard without Batch Normalization

Due to resource constraints, we have only conducted experiments for select dataset-network pairs when using the standard parameterization scheme, shown in figure 4. The optimal normalized noise scale is found using batch size search only. For CIFAR-10 on WRN, networks with $w = 2, 3, 4$ were trained with the learning rate 0.0025, while networks with $w = 6, 8$ were trained with a smaller learning rate of 0.00125 due to numerical instabilities (causing high failure rates for wide models when $\epsilon = 0.0025$). Once again, we observe a clear linear relationship between the optimal normalized noise scale and the widening factor. We provide additional experiments incorporating data augmentation, dropout and label smoothing in section F of the supplementary material, which also show the same linear relationship.

4.3. NTK with Batch Normalization

We have only conducted experiments with batch normalization for families of wide residual networks and CNNs. We perform both batch size search and learning rate search. The results of these experiments are summarized in figure 5. Unlike the previous sets of experiments, we do not report an optimal normalized noise scale, as this term is poorly defined when batch normalization is used. Rather, we report the optimal inverse batch size (learning rate) for the given fixed learning rate (batch size) respectively. However as discussed previously, when the batch size is fixed, the SGD noise is still proportional to the learning rate. A clear linear trend is still present for wide residual networks, however this trend is much weaker in the case of convolutional networks.

4.4. Generalization and Network Width

We showed in section 4.1 that in the absence of batch normalization, the test accuracy of a network of a given width is determined by its normalized noise scale. Therefore in figure 6, we plot the test accuracy as a function of noise scale for a range of widths. We include a variety of networks trained without batch normalization with both standard and NTK parameterizations. In all cases the best observed test

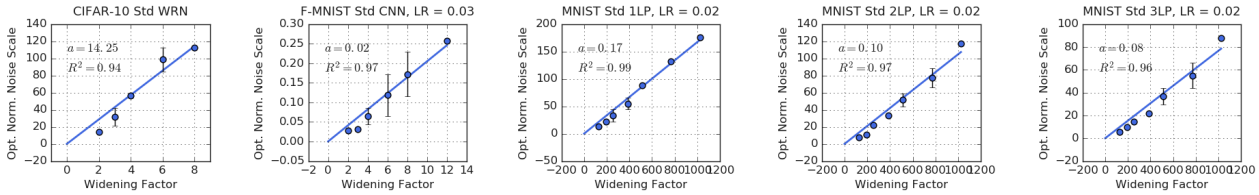

Figure 4. The optimal normalized noise scale vs. widening factor for networks parameterized using the standard scheme without batch normalization. The optimal normalized noise scale and the proportionality constant a are given in units of 10^6/[loss], and the optimal normalized noise scale is obtained via batch size search. All five plots exhibit a clear linear relationship.

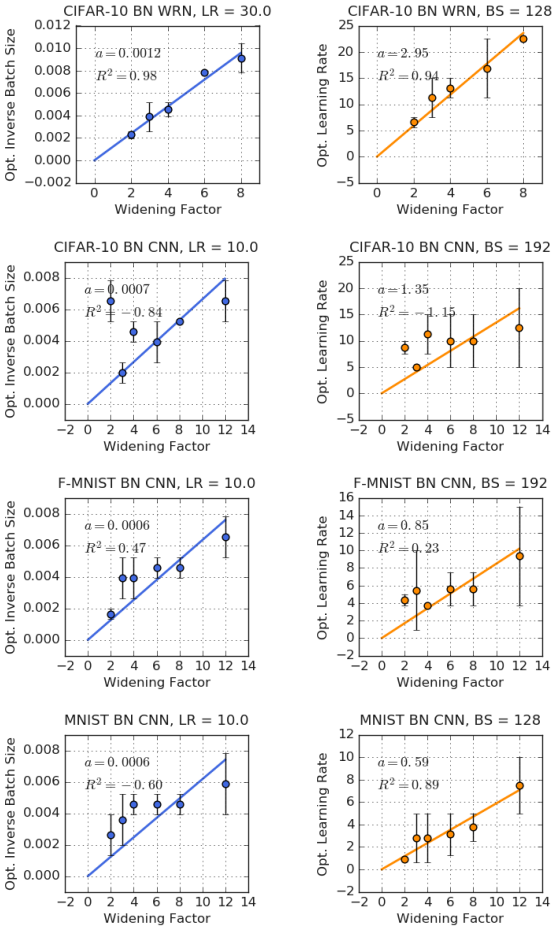

Figure 5. The optimal inverse batch size/learning rate vs. widening factor for NTK parameterized networks with batch normalization.

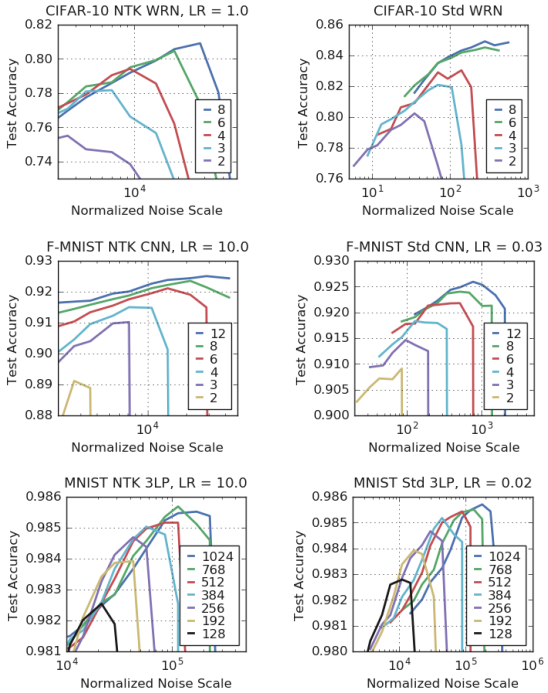

Figure 6. The test set accuracy plotted against normalized noise scale (in units of $[\text{loss}]^{-1}$) from experiments with fixed learning rate without batch normalization. The datasets, networks, parameterization schemes, and learning rates are indicated in the plot titles, except for the second plot where separate learning rates were used for different networks (see section 4.2). The legend indicates the widening factor, and the x-axis is in log-scale.

accuracy increases with width. This is consistent with previous work showing that test accuracy improves with increasing over-parameterization (Neyshabur et al., 2018; Lee et al., 2018; Novak et al., 2018). See section D of the supplementary material for plots of test accuracy in terms of the raw learning rate and batch size instead of noise scale.

More surprisingly, the dominant factor in the improvement of test accuracy with width is usually the increased optimal normalized noise scale of wider networks. To see this, we note that for a given width the test accuracy often improves slowly below the optimum noise scale and then falls rapidly above it. Wider networks have larger optimal noise scales, and this enables their test accuracies to rise higher before they drop. Crucially, when trained at a fixed noise scale below the optimum, the test accuracy is very similar across networks of different widths, and wider networks do not consistently outperform narrower ones. This suggests the empirical performance of wide over-parameterized networks is closely associated with the implicit regularization of SGD.

In figure 7 we examine the relationship between generalization and network width for experiments with batch normalization. Since the normalized noise scale is not well-defined, we plot the test accuracy for a range of widths as a function of both the batch size at fixed learning rate, and the learning rate at fixed batch size. We provide plots for WRNs on

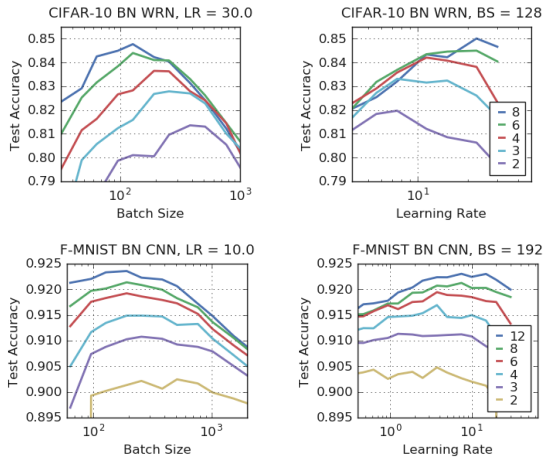

Figure 7. The test accuracy plotted against batch size/learning rate for experiments with fixed learning rate/batch size respectively. The networks are NTK-parameterized and use batch normalization. The x-axis is log-scaled, and the legend indicates the widening factor of the plotted networks. The dataset, network architecture and values for the fixed parameters are indicated in the title.

CIFAR-10 and CNNs on F-MNIST, both in the NTK parameterization. Once again, we find that the best observed test accuracy consistently increases with width. However the qualitative structure of the data differs substantially for the two architectures. In the case of WRNs, we observe similar trends both with and without batch normalization. In the NTK parameterization, wider networks have larger optimal learning rates (smaller optimal batch sizes), and this is the dominant factor behind their improved performance on the test set. For comparison, see figure 1 in section D of the supplementary material for equivalent plots without batch normalization. However in the case of CNNs the behaviour is markedly different, wider networks perform better across a wide range of learning rates and batch sizes. This is consistent with our earlier observation that WRNs obey equation 6 with batch normalization, while CNNs do not.

5. Discussion

Speculations on main result: The proportionality relation (equation 6) between the optimal normalized noise scale and network width holds remarkably robustly in all of our experiments without batch normalization. This relationship provides a simple prescription which predicts how to tune SGD hyper-parameters as width increases. We do not have a theoretical explanation for this phenomenon. However intuitively it appears that noise enhances the final test performance, while the amount of noise a network can tolerate is proportional to the network width. Wider networks tolerate more noise, and thus achieve higher test accuracies. Why SGD noise enhances final performance remains a mystery (Keskar et al., 2016; Sagun et al., 2017; Mandt et al., 2017; Chaudhari & Soatto, 2017; Smith & Le, 2017).

Implications for very wide networks: As noted in the introduction, the largest batch size consistent with equation 6 decreases with width (when training with SGD + momentum without regularization or batch normalization). To clarify this point, we consider NTK networks and standard networks separately. For NTK networks, the learning rate can stay constant with respect to the width without introducing numerical instabilities. As the network gets wider equation 6 requires $B \propto \epsilon/w$, which forces the batch size of wide networks to have a small value to achieve optimality. For standard networks, $B \propto \epsilon$. However in this case the learning rate $\epsilon \propto 1/w$ must decay as the width increases in order for the SGD to remain stable (Karakida et al., 2017). We provide empirical evidence for these stability bounds in section H of the supplementary material. In both cases the batch size must eventually be reduced if we wish to maintain optimal performance as width increases.

Unfortunately we have not yet been able to perform additional experiments at larger widths. However if the trends above hold for arbitrary widths, then there would be a surprising implication. Since the batch size is bounded from below by one, and the normalized learning rate is bounded above by some value $\bar{\epsilon}_{\text{max}}$ due to numerical stability, there is a maximum noise scale we can achieve experimentally. Meanwhile the optimal noise scale increases proportional to the network width. This suggests there may be a critical width for each network family, at which the optimal noise scale exceeds the maximum noise scale, and beyond which the test accuracy does not improve as the width increases, unless additional regularization methods are introduced.

Comments on batch normalization: The analysis of small learning rate SGD proposed by Smith & Le (2017) does not hold with batch normalization, and we therefore anticipated that networks trained using batch normalization might show a different trend. Surprisingly, we found in practice that residual networks trained using batch normalization do follow the trend implied by equation 6, while convolutional networks trained with batch normalization do not.

Conclusion: We introduce the normalized noise scale, which extends the analysis of small learning rate SGD proposed by Smith & Le (2017) to account for the choice of parameterization scheme. We provide convincing empirical evidence that, in the absence of batch normalization, the normalized noise scale which maximizes the test set accuracy is proportional to the network width. We also find that wider networks perform better on the test set. A similar trend holds with batch normalization for residual networks, but not for convolutional networks. We consider two parameterization schemes and three model families including MLPs, ConvNets and ResNets. Since the largest stable learning rate is bounded, the largest batch size consistent with the optimal noise scale decreases as the width increases.

Acknowledgements

We thank Yasaman Bahri, Boris Hanin, Simon Kornblith, Jaehoon Lee, Luke Metz, Roman Novak, George Philipp, Ben Poole, Chris Shallue, Ola Spyra, Olga Wichrowska, Soham De and Sho Yaida for helpful discussions.

References

Bartlett, P. L., Foster, D. J., and Telgarsky, M. J. Spectrally-normalized margin bounds for neural networks. In *Advances in Neural Information Processing Systems*, pp. 6240–6249, 2017.

Chaudhari, P. and Soatto, S. Stochastic gradient descent performs variational inference, converges to limit cycles for deep networks. *CoRR*, abs/1710.11029, 2017. URL http://arxiv.org/abs/1710.11029.

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

Glorot, X. and Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS 2010*, 2010.

Goyal, P., Dollár, P., Girshick, R. B., Noordhuis, P., Wesolowski, L., Kyrola, A., Tulloch, A., Jia, Y., and He, K. Accurate, large minibatch SGD: training imagenet in 1 hour. *CoRR*, abs/1706.02677, 2017. URL http://arxiv.org/abs/1706.02677.

Harvey, N., Liaw, C., and Mehrabian, A. Nearly-tight vc-dimension bounds for piecewise linear neural networks. In *Conference on Learning Theory*, pp. 1064–1068, 2017.

He, K., Zhang, X., Ren, S., and Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *CoRR*, abs/1502.01852, 2015. URL http://arxiv.org/abs/1502.01852.

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

Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*, abs/1502.03167, 2015. URL http://arxiv.org/abs/1502.03167.

Jacot, A., Gabriel, F., and Hongler, C. Neural tangent kernel: Convergence and generalization in neural networks. *CoRR*, abs/1806.07572, 2018. URL http://arxiv.org/abs/1806.07572.

Jastrzebski, S., Kenton, Z., Arpit, D., Ballas, N., Fischer, A., Bengio, Y., and Storkey, A. J. Three factors influencing minima in SGD. *CoRR*, abs/1711.04623, 2017. URL http://arxiv.org/abs/1711.04623.

Karakida, R., Akaho, S., and ichi Amari, S. Universal statistics of fisher information in deep neural networks: Mean field approach. *arXiv preprint arXiv:*, abs/1806.01316, 2017. URL http://arxiv.org/abs/1806.01316.

Karras, T., Aila, T., Laine, S., and Lehtinen, J. Progressive growing of gans for improved quality, stability, and variation. *CoRR*, abs/1710.10196, 2017. URL http://arxiv.org/abs/1710.10196.

Karras, T., Laine, S., and Aila, T. A style-based generator architecture for generative adversarial networks. *CoRR*, abs/1812.04948, 2018. URL http://arxiv.org/abs/1812.04948.

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. *arXiv preprint arXiv:1609.04836*, 2016.

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

LeCun, Y., Bottou, L., Orr, G. B., and Müller, K.-R. Efficient backprop. In *Neural Networks: Tricks of the Trade*, 1996.

Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, pp. 2278–2324, 1998.

LeCun, Y., Cortes, C., and Burges, C. J. MNIST handwritten digit database. 2010. URL http://yann.lecun.com/exdb/mnist/.

Lee, J., Bahri, Y., Novak, R., Schoenholz, S., Pennington, J., and Sohl-dickstein, J. Deep neural networks as gaussian processes. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=B1EA-M-0Z.

Mandt, S., Hoffman, M. D., and Blei, D. M. Stochastic gradient descent as approximate bayesian inference. *arXiv preprint arXiv:*, abs/1704.04289, 2017. URL http://arxiv.org/abs/1704.04289.

McCandlish, S., Kaplan, J., Amodei, D., and Team, O. D. An empirical model of large-batch training. *CoRR*, abs/1812.06162, 2018. URL http://arxiv.org/abs/1812.06162.

Nagarajan, V. and Kolter, Z. Deterministic pac-bayesian generalization bounds for deep networks via generalizing noise-resilience. 2018.

Neyshabur, B., Bhojanapalli, S., and Srebro, N. A pac-bayesian approach to spectrally-normalized margin bounds for neural networks. *arXiv preprint arXiv:1707.09564*, 2017.

Neyshabur, B., Li, Z., Bhojanapalli, S., LeCun, Y., and Srebro, N. Towards understanding the role of over-parametrization in generalization of neural networks. *arXiv preprint arXiv:1805.12076*, 2018.

Novak, R., Bahri, Y., Abolafia, D. A., Pennington, J., and Sohl-Dickstein, J. Sensitivity and generalization in neural networks: an empirical study. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=HJC2SzZCW.

Sagun, L., Evci, U., Guney, V. U., Dauphin, Y., and Bottou, L. Empirical analysis of the hessian of over-parametrized neural networks. *arXiv preprint arXiv:1706.04454*, 2017.

Schoenholz, S. S., Gilmer, J., Ganguli, S., and Sohl-Dickstein, J. Deep information propagation. *CoRR*, abs/1611.01232, 2016. URL http://arxiv.org/abs/1611.01232.

Sculley, D., Snoek, J., Wiltschko, A. B., and Rahimi, A. Winner's curse? on pace, progress, and empirical rigor. In *ICLR (Workshop)* Sculley et al. (2018). URL http://dblp.uni-trier.de/db/conf/iclr/iclr2018w.html#SculleySWR18.

Shallue, C. J., Lee, J., Antognini, J. M., Sohl-Dickstein, J., Frostig, R., and Dahl, G. E. Measuring the effects of data parallelism on neural network training. *CoRR*, abs/1811.03600, 2018. URL http://arxiv.org/abs/1811.03600.

Smith, S. L. and Le, Q. V. A bayesian perspective on generalization and stochastic gradient descent. *CoRR*, abs/1710.06451, 2017. URL http://arxiv.org/abs/1710.06451.

Smith, S. L., Kindermans, P., and Le, Q. V. Don't decay the learning rate, increase the batch size. *CoRR*, abs/1711.00489, 2017. URL http://arxiv.org/abs/1711.00489.

Soudry, D., Hoffer, E., Nacson, M. S., Gunasekar, S., and Srebro, N. The implicit bias of gradient descent on separable data. *The Journal of Machine Learning Research*, 19(1):2822–2878, 2018.

van Laarhoven, T. L2 regularization versus batch and weight normalization. *CoRR*, abs/1706.05350, 2017. URL http://arxiv.org/abs/1706.05350.

Wilson, A. C., Roelofs, R., Stern, M., Srebro, N., and Recht, B. The marginal value of adaptive gradient methods in machine learning. In *Advances in Neural Information Processing Systems*, pp. 4148–4158, 2017.

Xiao, H., Rasul, K., and Vollgraf, R. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *CoRR*, abs/1708.07747, 2017. URL http://arxiv.org/abs/1708.07747.

Yaida, S. Fluctuation-dissipation relations for stochastic gradient descent. *arXiv preprint arXiv:*, abs/1810.00004, 2018. URL http://arxiv.org/abs/1810.00004.

Zagoruyko, S. and Komodakis, N. Wide residual networks. *CoRR*, abs/1605.07146, 2016. URL http://arxiv.org/abs/1605.07146.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Understanding deep learning requires rethinking generalization. *CoRR*, abs/1611.03530, 2016. URL http://arxiv.org/abs/1611.03530.

Zhou, W., Veitch, V., Austern, M., Adams, R. P., and Orbanz, P. Non-vacuous generalization bounds at the imagenet scale: a pac-bayesian compression approach. 2018.