Supplementary Material

A. Networks and Datasets

We consider MLPs with 1, 2 or 3 hidden layers. Each layer has the same number of hidden units, w. We denote the d-layered perceptron with width w by the label $d\text{LP}_w$. We do not consider batch normalization for these networks, and we consider the range of widths,

$$w \in \{128, 192, 256, 384, 512, 768, 1024\}. \tag{1}$$

We consider a family of convolutional networks CNN_w, obtained from LeNet-5 (figure 2 of Lecun et al. (1998)) by scaling all the channels, as well as the widths of the fully connected layers, by a widening factor of $w/2$ (the factor of $\frac{1}{2}$ allows integer w for all experiments). Thus, LeNet-5 is identified as CNN_2. We also consider batch normalization, which takes place after the dense or convolutional affine transformation, and before the activation function, as is standard. We do not use biases when we use batch normalization. We consider the widening factors,

$$w \in \{2, 3, 4, 6, 8, 12\}. \tag{2}$$

Finally, we consider a family of wide residual networks WRN_w, where WRN_w is equivalent to table 1 of Zagoruyko & Komodakis (2016) if $N = 2$ and $k = w$. For consistency with Zagoruyko & Komodakis (2016) the first wide ResNet layer, a 3 to 16 channel expansion, is not scaled with w. As for CNNs, we study WRNs both with and without batch normalization. We consider the widening factors,

$$w \in \{2, 3, 4, 6, 8\}. \tag{3}$$

The training sets of MNIST and Fashion-MNIST have been split into training-validation sets of size 55000-5000 while for CIFAR-10, the split is given by 45000-5000. We have used the official test set of 10000 images for each dataset. For MNIST and F-MNIST, we normalize the pixels of each image to range from -0.5 to 0.5. For CIFAR-10, we normalize the pixels to have zero mean and unit variance. We do not use data augmentation for any of the experiments presented in the main text.

B. Training Time for Experiments

For experiments with fixed learning rate ϵ_{fixed}, we set the number of training steps T by setting both an epoch bound E_{min} and a step bound T_{min}. So for a given batch size B the number of training steps is set by,

$$T = \max\left(T_{\text{min}}\,,\; E_{\text{min}} \cdot \frac{N_{\text{train}}}{B}\right). \tag{4}$$

After running the batch size search, we may choose a reasonable batch size B_{fixed} to hold fixed during learning search rate. Experiments with fixed batch size and variable learning rate are always paired to such a "parent experiment." When the batch size is fixed and the learning rate varies, we must scale the number of training steps proportional to the learning rate. We pick the reference learning rate ϵ_0 to be the learning rate at which the original batch size search was run, and a reference number of training steps T_0, which is computed at the fixed batch size B_{fixed} using the epoch and step bound provided in equation (4). Then for learning rate ϵ, the number of training steps T is given by

$$T = \max\left(T_0\,,\; T_0 \cdot \frac{\epsilon_0}{\epsilon}\right). \tag{5}$$

That is, for learning rates larger than ϵ_0 we perform T_0 updates, while for learning rates smaller than ϵ_0, we scale the number of updates inversely proportional to the learning rate.

C. Experiment Details and Configurations

In this section we detail the specific configurations of experiments run in this work.

C.1. NTK without Batch Normalization

Table 1. Epoch and step bounds for dataset-network family pairs for various experimental settings.

Dataset	Networks	E_{min}	T_{min}	$\epsilon_{\text{BS\_search}}$	$B_{\text{LR\_search}}$	$\epsilon_{0,\text{LR\_search}}$	$T_{0,\text{LR\_search}}$
MNIST	1LP	120	80k	10.0	8	10.0	825k
MNIST	2LP	120	80k	10.0	16	10.0	412.5k
MNIST	3LP	120	80k	10.0	16	10.0	412.5k
MNIST	CNN	120	80k	10.0	24	10.0	275k
F-MNIST	1LP	240	160k	10.0	12	10.0	1100k
F-MNIST	2LP	240	160k	10.0	24	10.0	550k
F-MNIST	3LP	240	160k	10.0	48	10.0	275k
F-MNIST	CNN	240	160k	10.0	96	10.0	160k
CIFAR-10	CNN	540	320k	5.0	256	10.0	160k
CIFAR-10	WRN	270	80k	1.0	8	1.0	1500k

We run both batch size search and learning rate search to determine the optimal normalized noise scale for networks trained with NTK parameterization and without batch normalization. The relevant parameters used for the search experiments are listed in table 1. The scalar $\epsilon_{\text{BS\_search}}$ denotes the fixed learning rate used for batch size search, while $B_{\text{LR\_search}}$ denotes the fixed batch size used during learning rate search.

The epoch bound E_{min} and the training step bound T_{min} are defined in section B of the supplementary material. Also as we explained in section B, the training time is scaled with respect to a reference training time and a reference learning rate for the learning rate search experiments. These are denoted $T_{0,\text{LR\_search}}$ and $\epsilon_{0,\text{LR\_search}}$ in the table respectively.

C.2. Standard without Batch Normalization

Table 2. Epoch and step bounds for dataset-network family pairs for various experimental settings.

Dataset	Networks	w	E_{min}	T_{min}	$\epsilon_{\text{BS\_search}}$
MNIST	MLP	All	120	80k	0.02
F-MNIST	CNN	All	480	320k	0.03
CIFAR-10	WRN	2, 3, 4	540	160k	0.0025
CIFAR-10	WRN	6, 8	1080	320k	0.00125

For standard networks without batch normalization, we only carry out batch size search experiments at a fixed learning rate $\epsilon_{\text{BS\_search}}$. For CIFAR-10 experiments on wide residual networks, we chose to use two different learning rates depending on the width of the networks (narrower networks can be trained faster with a bigger learning rate, while wider networks require a smaller learning rate for numerical stability). The experiment configurations are listed in table 2.

C.3. NTK with Batch Normalization

Table 3. Epoch and step bounds for dataset-network family pairs for various experimental settings.

Dataset	Networks	E_{min}	T_{min}	$\epsilon_{\text{BS\_search}}$	$B_{\text{LR\_search}}$	$\epsilon_{0,\text{LR\_search}}$	$T_{0,\text{LR\_search}}$
MNIST	CNN	120	80k	10.0	128	10.0	80k
F-MNIST	CNN	480	320k	10.0	192	10.0	320k
CIFAR-10	CNN	540	320k	10.0	192	10.0	320k
CIFAR-10	WRN	270	80k	30.0	192	30.0	80k

The experiment configurations for networks parameterized using the NTK scheme with batch normalization are listed in table 3. Both batch size search and learning rate search have been carried out. The parameters defined are equivalent to those used for NTK networks without batch normalization in section C.1.

D. Plots from Batch Search and Learning Rate Search

In this section, we present plots of the average test set accuracy vs. batch size/learning rate for batch size/learning rate search experiments with fixed learning rate/batch size respectively. All the learning rate search experiments are paired with batch size search experiments, and share the same color-code and legend describing the network widening factors.

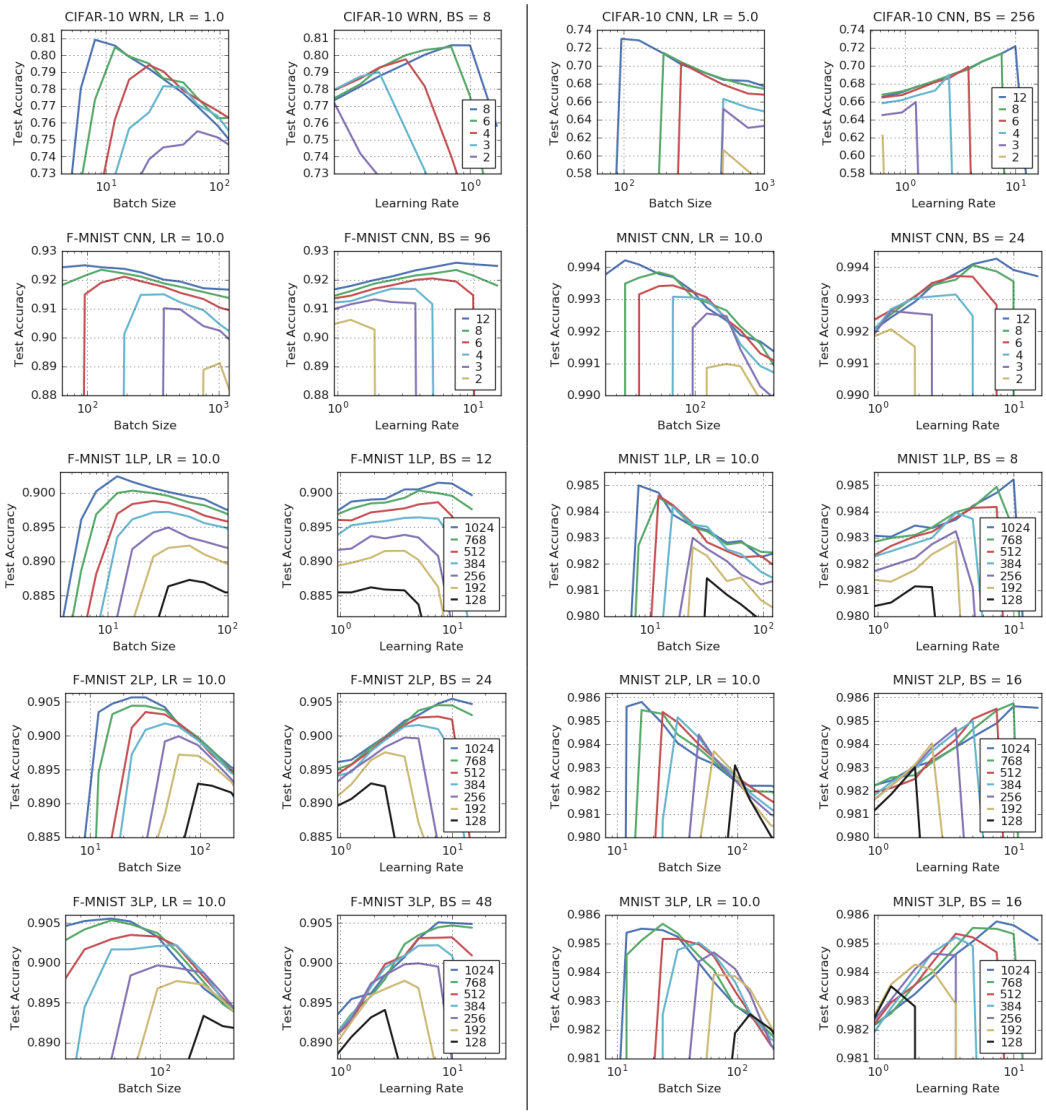

Figure 1. The test accuracy plotted against batch size/learning rate for experiments at fixed learning rate/batch size respectively. These plots are from the experiments performed on NTK-parameterized networks without batch normalization. The legend indicates the widening factor of the plotted networks, and the x-axis is plotted in log scale. The values for the fixed parameters are indicated in the title.

Figure 1 plots the results of batch size/learning rate search experiments run with NTK-parameterized networks without batch normalization. Here, the x-axis is plotted in log-scale. Since $\bar{g} \propto \epsilon/B$, if the performance of the network is determined by the noise scale, then the figures for batch size search and learning rate search experiments on the same dataset-network pair should be symmetric to one another. This symmetry is nicely on display in figure 1.

Figure 2 plots batch size search experiments run with standard parameterization and without batch normalization. Standard-parameterized WRN_6 and WRN_8 were run with a reduced learning rate due to numerical stability issues, and the results of their batch size search experiments have been plotted separately. In contrast to NTK-parameterized networks, the normalized noise is width-dependent for networks parameterized using the standard scheme. Also, we have batch searches conducted over varying learning rates in one instance. Thus it is more informative to put everything together and plot the performance of the network against $(w\epsilon)/B \propto \bar{g}$. This has been done in figure 3. This plot reproduces the qualitative features of figure 1,

which is strong evidence for $\bar{g}$ being the correct indicator of the performance of networks within a linear family.

Figure 4 plots the results of batch size/learning rate search run with NTK-parameterized networks with batch normalization.

When both batch size and learning rate search have been carried out, the y-axes of the plots, along which the network performance is plotted, are aligned so that the maximal performance obtained from the search can be compared.

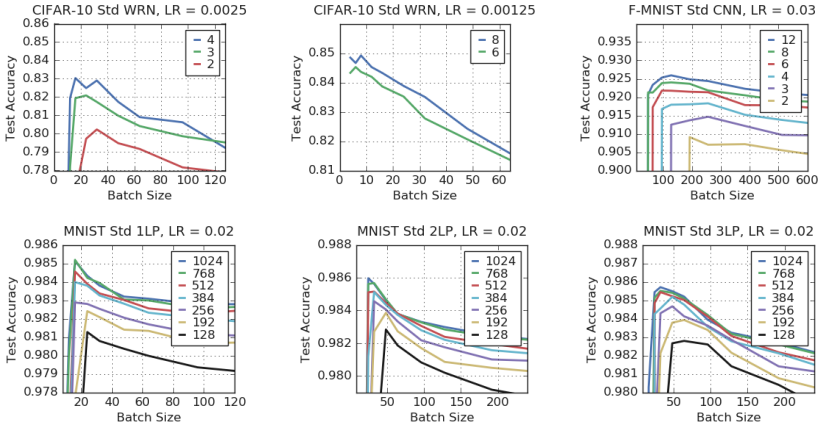

Figure 2. The test accuracy plotted against batch size for experiments at fixed learning rate. The plots are from networks parameterized in the standard scheme without batch normalization. The legend indicates the widening factor of the plotted networks.

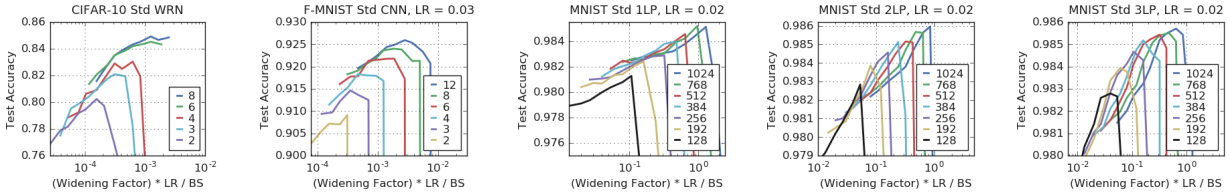

Figure 3. The test accuracy plotted against $w \cdot \epsilon / B \propto \bar{g}$ for networks parameterized in the standard scheme without batch-normalization. The x-axis is log-scaled. The legend indicates the widening factor.

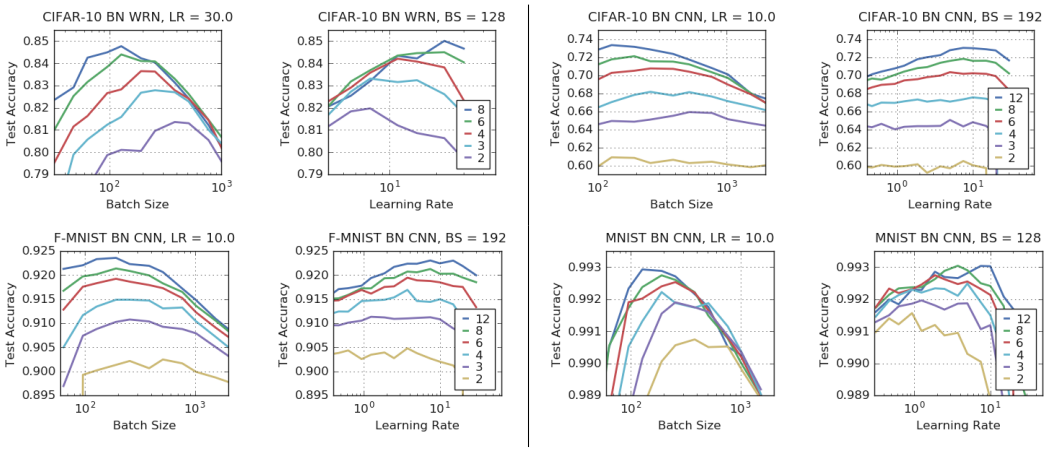

Figure 4. The test accuracy plotted against (learning rate) for experiments at fixed learning rate/batch size respectively. The networks are NTK-parameterized and use batch normalization. The x-axis is log-scaled. The legend indicates the widening factor.

E. Networks with Standard Parameterization and Batch Normalization

In this section, we present results of batch search experiments with WRNs with batch normalization that are parameterized using the standard scheme. We train with a constant schedule with epoch bound $E_{\text{min}} = 270$ and step bound $T_{\text{min}} = 80\text{k}$.

As was with the case with NTK parameterized WRNs, the scaling rule for the optimum batch size coincides with that of the case when batch normalization is absent, i.e., the optimal batch size B_{opt} is constant with respect to the widening factor.

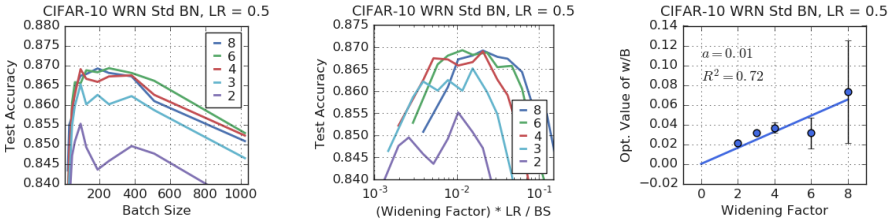

Figure 5. Three plots summarizing the results of the batch search experiments with standard WRNs with batch normalization trained on CIFAR-10. The test accuracy of the networks are plotted against the batch sizes in the first figure, and the value of $w \cdot \epsilon/B$ in the second figure. The x-axis is log-scaled in the second figure. The optimal value of w/B is plotted against the network widths in the third figure.

F. Batch Search Experiments with Regularization

In this section, we present the results of training WRNs on CIFAR-10 with regularization. We use dropout with probability 0.3 and label smoothing with uncertainty 0.9. Data augmentation is also applied by first taking a random crop of the image padded by 4 pixels and then applying a random flip. Note that we have chosen these regularization schemes because we do not anticipate that the associated hyper-parameters will depend strongly on the network width or the noise scale.

We carry out batch search experiments for WRNs parameterized in the standard scheme on CIFAR-10, with epoch bound $E_{\text{min}} = 270$ and step bound $T_{\text{min}} = 80\text{k}$. The results are given in figure 6. We have used the fixed learning rate $\epsilon = 0.03$.

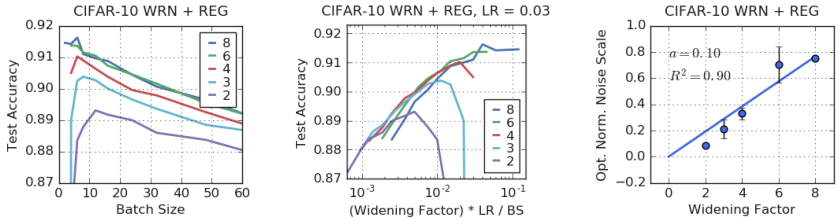

Figure 6. Three plots summarizing the results of the batch search experiments with regularized WRNs trained on CIFAR-10. The test accuracy of the various networks are plotted against the scanned batch sizes in the first figure. In the second figure, the test accuracy is plotted against $w \cdot \epsilon/B \propto \bar{g}$ where the x-axis is log-scaled. The legend indicates the widening factor in the first two figures. The optimal normalized noise scale in units of 10^3/[loss] is plotted against the network widths in the third figure.

The scaling rule $\bar{g}_{\text{opt}} \propto w$ still holds in the presence of these three regularizers. The use of regularization schemes significantly increases the final test accuracies, however these test accuracies still depend strongly on the SGD noise scale.

G. The Performance of 3LP on a 2D Grid of Learning Rates and Batch Sizes

Our main result (equation 6) is based on the theory of small learning rate SGD, which claims that the final performance of models trained using SGD is controlled solely by the noise scale $g \propto \epsilon/B$. To provide further evidence for this claim, here we consider 3LP_{512} and measure its performance on MNIST across a 2D grid of batch sizes $\{12, 16, 24, 32, 48, 64, 96, 128, 192\}$ and a range of learning rates as indicated in figure 7. We run 20 experiments for each learning rate/batch size pair and compute the mean test set accuracy. We set the epoch bound E_{min} to 120 and the training set bound $T_{\text{min}} = T_0 \cdot (\epsilon_0/\epsilon)$ with $T_0 = 412.5\text{k}$ and $\epsilon_0 = 10.0$. The results are shown in figure 7, where we plot the performance curves as a function of the batch size and the noise scale. As expected, the final test accuracy is governed solely by the noise scale.

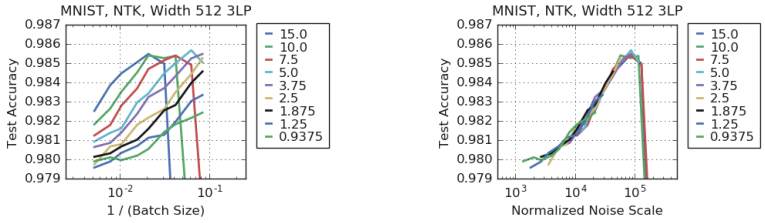

Figure 7. The test set accuracy of a 3-layer perceptron (width 512) for a 2D grid of learning rates and batch sizes. We provide the learning rate in the legend and plot the test accuracy against the batch size (left) and the normalized noise scale (right).

H. Numerical Stability and the Normalized Learning Rate

In this section, we present experiments to verify the claim that (in the absence of batch normalization), numerical instabilities affect training when $\epsilon \gtrsim \epsilon_{\text{unstable}}$, where $\epsilon_{\text{unstable}}$ is constant with respect to the width for NTK parameterization while $\epsilon_{\text{unstable}} \propto 1/w$ as $w \to \infty$ for networks parameterized using the standard scheme. This is equivalent to saying that the scale $\bar{\epsilon}_{\text{unstable}}$ at which the normalized learning rate becomes unstable is constant with respect to the width of the network.

To do so, we take families of NTK and standard parameterized networks, and compute the failure rate after 20 epochs of training with a fixed batch size (64) at a range of learning rates. For each network, we run 20 experiments, and compute the failure rate, which is defined to be the portion of experiments terminated by a numerical error. We run the experiments for CIFAR-10 on WRN_w ($w = 2, 3, 4, 6, 8$), F-MNIST on CNN_w ($w = 6, 8, 12, 16, 24$) and MNIST on $\text{2LP}_{512 \cdot w}$ ($w = 1, 2, 4, 8, 16, 32$).

For CNNs and 2-layer perceptrons, we consider much wider networks than are studied in the main text. This is because we are ultimately interested in observing numerical instabilities which occur when w is large. For the purpose of studying this break-down of numerical stability, we can afford to use much wider networks. The width dependence of $\epsilon_{\text{unstable}}$ becomes more evident by focusing on these wide networks, as the behaviour of narrow networks is less predictable.

Figure 8 depicts 3 figures for each dataset-network combination. The first figure shows the failure rate plotted against the learning rate for networks using the standard parameterization. The second is the failure rate plotted against the product of the learning rate and the widening factor—i.e., twice the normalized learning rate—for the same networks (trained using the standard parameterization). The third figure shows the failure rate plotted against the learning rate for networks parameterized using the NTK scheme. Here, the normalized learning rate is simply half the learning rate.

It is clear from these plots that $\bar{\epsilon}_{\text{unstable}}$ is independent of the widening factor for WRNs and CNNs, while the definition of $\bar{\epsilon}_{\text{unstable}}$ for the 2LP seems more subtle. Nevertheless for all three network families, we see similar stability curves as width increases for both parameterization schemes, when measured as a function of the normalized learning rate.

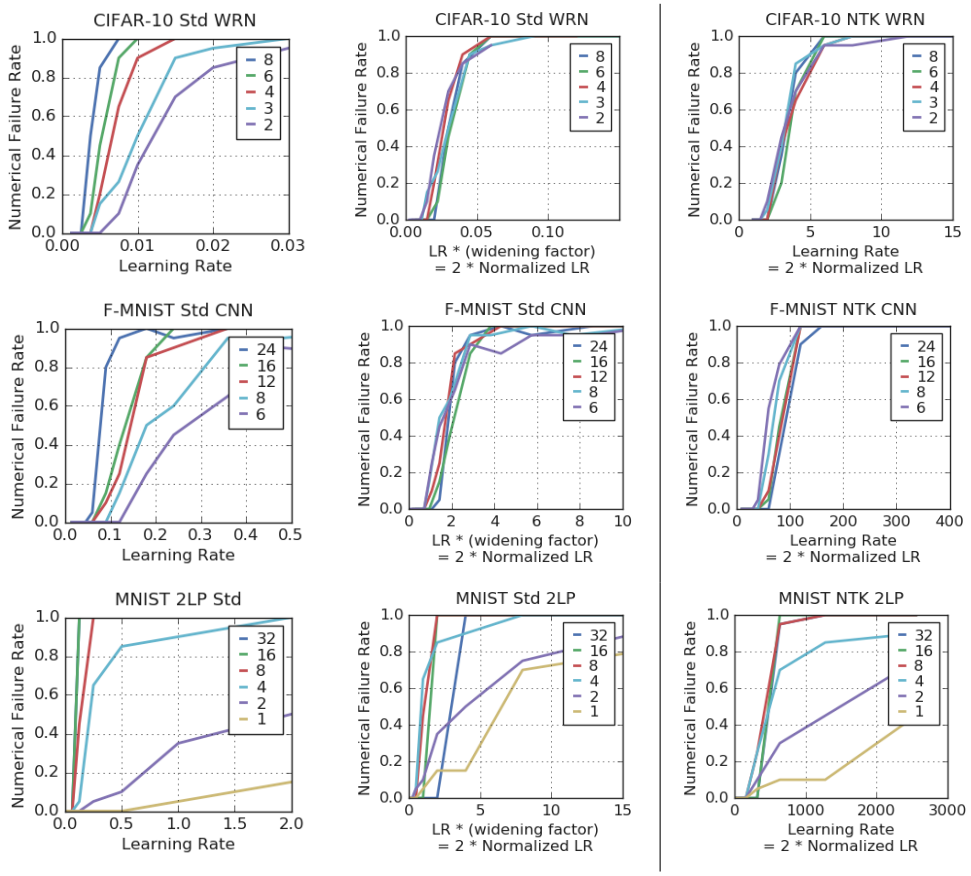

Figure 8. Failure rate plotted against the learning rate or the normalized learning rate for various dataset-network pairs.

I. Performance Comparison between NTK Networks and Standard Networks

Here we provide a brief comparison of the performance of both parameterization schemes on the test set. In figure 9, the peak test accuracy of a network parameterized with the standard scheme is plotted against the peak test accuracy obtained when the same network is parameterized with the NTK scheme. The dataset-network pairs are indicated in the title—CIFAR-10 on WRN_w, F-MNIST on CNN_w and MNIST on MLPs. We see that standard parameterization consistently out-performs NTK parameterization on WRNs and CNNs, although the performance is comparable for MNIST on MLPs.

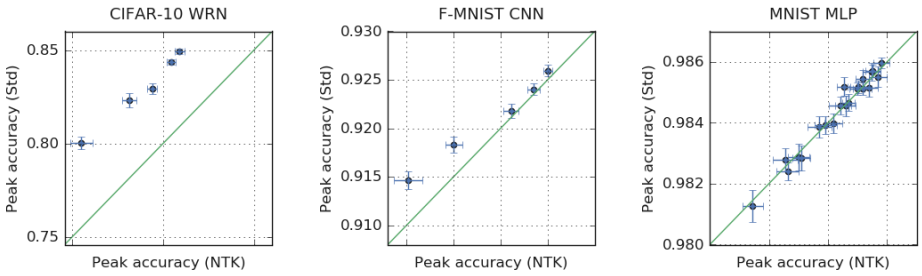

Figure 9. Performance of networks with standard parameterization plotted against performance of networks with NTK parameterization. The green line is the line $y = x$.

The reason that performance agrees well for the particular MLPs investigated in the main text of the paper, is because all the hidden layers have equal width. In this limit, NTK parameterization and standard parameterization are essentially identical. However by varying the network architecture, we can observe a discrepancy between the performance of MLPs as well. As an example, we consider the following bottom-heavy (BH) and top-heavy (TH) 3LPs with hidden layer widths,

$$\text{BH}_w \;:\; [4w,\, 2w,\, w]\,, \qquad \text{TH}_w \;:\; [w,\, 4w,\, 4w]. \tag{6}$$

We consider the widening factors $w \in \{128, 256, 512\}$. In figure 10, we display the discrepancy between the performance between both bottom-heavy and top-heavy networks parameterized in the NTK and standard schemes.

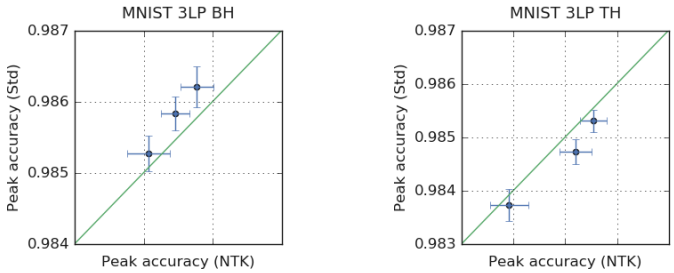

Figure 10. Performance of networks with standard parameterization plotted against performance of networks with NTK parameterization for 3LP networks BH_w and TH_w. The green line is the line $y = x$.

In the bottom heavy case, MLPs parameterized in the standard scheme appear to outperform MLPs parameterized in the NTK scheme. However in the top heavy case, MLPs parameterized in the NTK scheme appear to outperform MLPs parameterized in the standard scheme. These results suggest that neither scheme is superior to the other, but that the final performance will depend on the combination of parameterization scheme, initialization conditions and network architecture. We note that we initialize all our networks at critical initialization ($\sigma_0^2 = 2$), and that this overall weight scale was not tuned.

References

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.

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