Phase diagram of early training dynamics in deep networks: effect of the learning rate, depth, and width

Anonymous Author(s)
Affiliation
Address
email

Abstract

We systematically analyze optimization dynamics in deep neural networks (DNNs) trained with stochastic gradient descent (SGD) and study the effect of learning rate η, depth d, and width w of the neural network. By analyzing the maximum eigenvalue λ_t^H of the Hessian of the loss, which is a measure of sharpness of the loss landscape, we find that the dynamics can show four distinct regimes: (i) an early time transient regime, (ii) an intermediate saturation regime, (iii) a progressive sharpening regime, and (iv) a late time "edge of stability" regime. The early and intermediate regimes (i) and (ii) exhibit a rich phase diagram depending on $\eta \equiv c/\lambda_0^H$, d, and w. We identify several critical values of c, which separate qualitatively distinct phenomena in the early time dynamics of training loss and sharpness. Notably, we discover the opening up of a "sharpness reduction" phase, where sharpness decreases at early times, as d and $1/w$ are increased.

1 Introduction

The optimization dynamics of deep neural networks (DNNs) is a rich problem that is of great interest. Basic questions about how to choose learning rates and their effect on generalization error and training speed remain intensely studied research problems. Classical intuition from convex optimization has lead to the often made suggestion that in stochastic gradient descent (SGD), the learning rate η should satisfy $\eta < 2/\lambda^H$, where λ^H is the maximum eigenvalue of the Hessian H of the loss, in order to ensure that the network reaches a minimum. However several recent studies have suggested that it is both possible and potentially preferable to have the learning rate *early in training* reach $\eta > 2/\lambda^H$ [63, 47, 68]. The idea is that such a choice will induce a temporary training instability, causing the network to 'catapult' out of a local basin into a flatter one with lower λ^H where training stabilizes. Indeed, during the early training phase, the local curvature of the loss landscape changes rapidly [40, 1, 36, 16], and the learning rate plays a crucial role in determining the convergence basin [36]. Flatter basins are believed to be preferable because they potentially lead to lower generalization error [30, 31, 40, 12, 38, 14] and allow larger learning rates leading to potentially faster training.

From a different perspective, the major theme of deep learning is that it is beneficial to increase the model size as much as possible. This has come into sharp focus with the discovery of scaling laws that show power law improvement in generalization error with model and dataset size [39]. This raises the fundamental question of how one can scale DNNs to arbitrarily large sizes while maintaining the ability to learn; in particular, how should initialization and optimization hyperparameters be chosen to maintain a similar quality of learning as the model size is taken to infinity [33, 45, 46, 11, 66, 55, 67, 65]?

Motivated by these ideas, we perform a systematic analysis of the training dynamics of SGD for DNNs as learning rate, depth, and width are tuned, across a variety of architectures and datasets. We

Submitted to 37th Conference on Neural Information Processing Systems (NeurIPS 2023). Do not distribute.

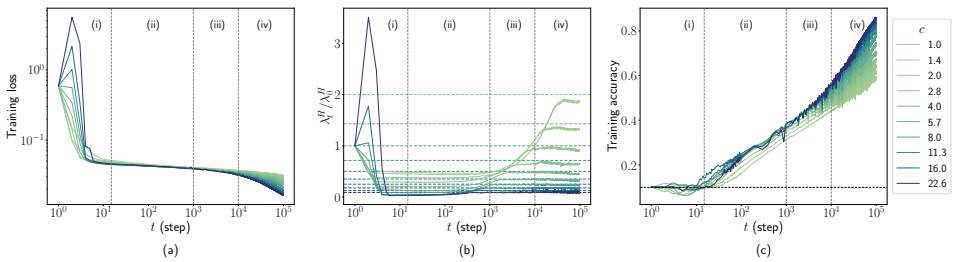


Figure 1: Training trajectories of the (a) training loss, (b) sharpness, and (c) training accuracy of CNNs ($d = 5$ and $w = 512$) trained on CIFAR-10 with MSE loss using vanilla SGD with learning rates $\eta = {}^{c}/\lambda_0^H$ and batch size $B = 512$. Vertical dashed lines approximately separate the different training regimes. Horizontal dashed lines in (b) denote the ${}^{2}/\eta$ threshold for each learning rate.

monitor both the loss and sharpness (λ^H) trajectories during early training, observing a number of qualitatively distinct phenomena summarized below.

1.1 Our contributions

We study SGD on fully connected networks (FCNs) with the same number of hidden units (width) in each layer, convolutional neural networks (CNNs), and ResNet architectures of varying width w and depth d with ReLU activation. For CNNs, the width corresponds to the number of channels. We focus on networks parameterized in Neural Tangent Parameterization (NTP) [33], and Standard Parameterization (SP) [59] initialized at criticality [52, 55], while other parameterizations and initializations may show different behavior. Further experimental details are provided in Appendix A. We study both mean-squared error (MSE) and cross-entropy loss functions and the datasets CIFAR-10, MNIST, Fashion-MNIST. Our findings apply to networks with ${}^{d}/w \lesssim C$, where C depends on architecture class (e.g. for FCNs, $C \approx {}^{1}/16$) and loss function, but is independent of d, w, and η. Above this ratio, the dynamics becomes noise-dominated, and separating the underlying deterministic dynamics from random fluctuations becomes challenging, as shown in Appendix E. We use sharpness to refer to λ_t^H, the maximum eigenvalue of H at time-step t, and flatness refers to ${}^{1}/\lambda_t^H$.

By monitoring the sharpness, we find four clearly separated, qualitatively distinct regimes throughout the training trajectory. Fig. 1 shows an example from a CNN architecture. The four observed regimes are: (i) an early time transient regime where loss and sharpness may drastically change and eventually settle down, (ii) an intermediate saturation regime where the sharpness has lowered and remains relatively constant, (iii) a progressive sharpening regime where sharpness steadily rises, and finally, (iv) a late time regime where the sharpness saturates around ${}^{2}/\eta$ for MSE loss; whereas for cross-entropy loss, sharpness drops after reaching this maximum value while remaining less than ${}^{2}/\eta$ [8]. Note the log scale in Figure 1 highlights the early regimes (i) and (ii); in absolute terms these are much shorter in time than regimes (iii) and (iv).

In this work, we focus on the early transient and intermediate saturation regimes. As learning rate, d and w are tuned, a clear picture emerges, leading to a rich phase diagram, as demonstrated in Section 2. Given the learning rate scaled as $\eta = {}^{c}/\lambda_0^H$, we characterize four distinct behaviors in the training dynamics in the early transient regime (i):

Sharpness reduction phase ($c < c_{loss}$) : *Both the loss and the sharpness* monotonically decrease during early training. There is a particularly significant drop in sharpness in the regime $c_{crit} < c < c_{loss}$, which motivates us to refer to learning rates lower than c_{crit} as sub-critical and larger than c_{crit} as super-critical. We discuss c_{crit} in detail below. The regime $c_{crit} < c < c_{loss}$ opens up significantly with increasing d and ${}^{1}/w$, which is a new result of this work.

Loss catapult phase ($c_{loss} < c < c_{sharp}$) : The first few gradient steps take training to a flatter region but with a higher loss. Training eventually settles down in the flatter region as the loss starts to decrease again. The sharpness *monotonically decreases from initialization* in this early time transient regime.

Loss and sharpness catapult phase ($c_{sharp} < c < c_{max}$): In this regime *both the loss and sharpness* initially start to increase, effectively catapulting to a different point where loss and sharpness can

start to decrease again. Training eventually exhibits a significant reduction in sharpness by the end of the early training. The report of a *loss and sharpness catapult* is also new to this work.

Divergent phase ($c > c_{max}$): The learning rate is too large for training and the loss diverges.

The critical values c_{loss}, c_{sharp}, c_{max} are random variables that depend on random initialization, SGD batch selection, and architecture. The averages of c_{loss}, c_{sharp}, c_{max} shown in the phase diagrams show strong systematic dependence on depth and width. In order to better understand the cause of the sharpness reduction during early training we study the effect of network output at initialization by (1) centering the network, (2) setting last layer weights to zero, or (3) tuning the overall scale of the output layer. We also analyze the linear connectivity of the loss landscape in the early transient regime and show that for a range of learning rates $c_{loss} < c < c_{barrier}$, no barriers exist from the initial state to the final point of the initial transient phase, even though training passes through regions with higher loss than initialization.

Next, we provide a quantitative analysis of the intermediate saturation regime. We find that sharpness during this time typically displays 3 distinct regimes as the learning rate is tuned, depicted in Fig. 5. By identifying an appropriate order parameter, we can extract a sharp peak corresponding to c_{crit}. For MSE loss $c_{crit} \approx 2$, whereas for crossentropy loss, $4 \gtrsim c_{crit} \gtrsim 2$. For $c \ll c_{crit}$, the network is effectively in a lazy training regime, with increasing fluctuations as d and/or $1/w$ are increased.

Finally, we show that a single hidden layer linear network – the uv model – displays the same phenomena discussed above and we analyze the phase diagram in this minimal model.

1.2 Related works

A significant amount of research has identified various training regimes using diverse criteria, e.g., [13, 1, 15, 36, 17, 43, 34, 8, 32]. Here we focus on studies that characterize training regimes with sharpness and learning rates. Several studies have analyzed sharpness at different training times [36, 16, 34, 8, 32]. Ref. [8] studied sharpness at late training times and showed how *large-batch* gradient descent shows progressive sharpening followed by the edge of stability, which has motivated various theoretical studies [9, 2, 3]. Ref. [36] studied the entire training trajectory of sharpness in models trained with SGD and cross-entropy loss and found that sharpness increases during the early stages of training, reaches a peak, and then decreases. In contrast, we find a sharpness-reduction phase, $c < c_{loss}$ which becomes more prominent with increasing d and $1/w$, where sharpness only decreases during early training; this also occurs in the catapult phase $c_{loss} < c < c_{sharp}$, during which the loss initially increases before decreasing. This discrepancy is likely due to different initialization and learning rate scaling in their work [32].

Ref. [34] examined the effect of hyperparameters on sharpness at late training times. Ref. [19] studied the optimization dynamics of SGD with momentum using sharpness. Ref. [43] classify training into 2 different regimes using training loss, providing a significantly coarser description of training dynamics than provided here. Ref. [32] studied the scaling of the maximum learning rate with d and w during early training in FCNs and its relationship with sharpness at initialization.

Ref. [47] analyzed the curvature during early training using the top eigenvalue of the neural tangent kernel (NTK) and demonstrated the existence of a new early training phase, which they dubbed the "catapult" phase, $2/\lambda_0^{NTK} < \eta < \eta_{max}$, in wide networks trained with MSE loss using SGD, in which training converges after an initial increase in training loss. The existence of this new training regime was further extended to quadratic models with large widths by [68, 50]. Our work extends the above analysis by studying the combined effect of learning rate, depth, and width for both MSE and cross-entropy loss, demonstrating the opening of a sharpness-reduction phase, the refinement of the catapult phase into two phases depending on whether the sharpness also catapults, analyzing the phase boundaries as d and $1/w$ is increased, analyzing linear mode connectivity in the catapult phase, examining different qualitative behaviors in the intermediate saturation regime (ii) mentioned above.

2 Phase diagram of early transient regime

For wide enough networks trained with MSE loss using SGD, training converges into a flatter region after an initial increase in the training loss for learning rates $c > 2$ [47]. Fig. 2(a, b) shows the first 10 steps of the loss and sharpness trajectories of a shallow ($d = 5$ and $w = 512$) CNN trained on

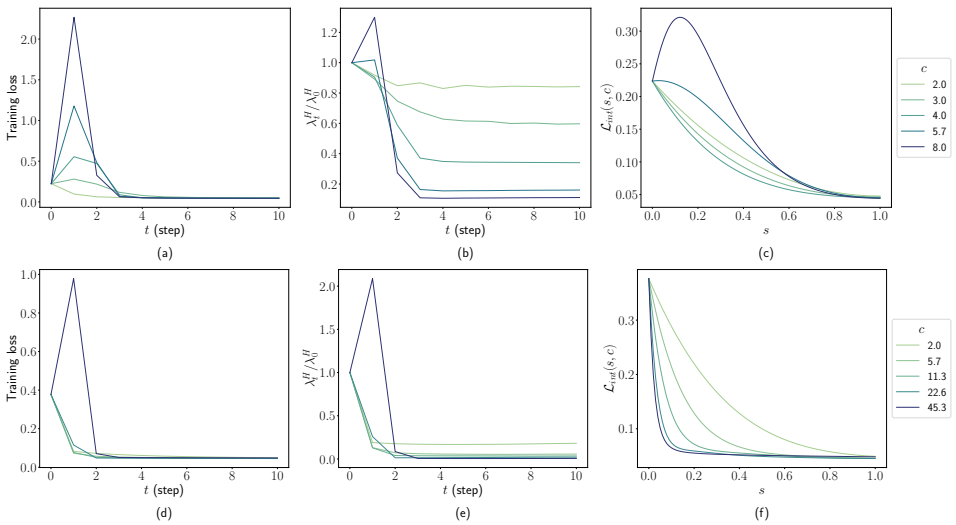


Figure 2: Early training dynamics of (a, b, c) a shallow ($d = 5$, $w = 512$) and (d, e, f) a deep CNN ($d = 10, w = 128$) trained on CIFAR-10 with MSE loss for $t = 10$ steps using SGD for various learning rates $\eta = c/\lambda_0^H$ and batch size $B = 512$. (a, d) training loss, (b, e) sharpness, and (c, f) interpolated loss between the initial and final parameters after 10 steps for the respective models. For the shallow CNN, $c_{loss} = 2.82, c_{sharp} = 5.65, c_{max} = 17.14$ and for the deep CNN, $c_{loss} = 36.75, c_{sharp} = 39.39, c_{max} = 48.50$.

the CIFAR-10 dataset with MSE loss using SGD. For learning rates, $c \geq 2.82$, the loss catapults and training eventually converges into a flatter region, as measured by sharpness. Additionally, we observe that sharpness may also spike initially, similar to the training loss (see Fig. 2 (b)). However, this initial spike in sharpness occurs at relatively higher learning rates ($c \geq 5.65$), which we will examine along with the loss catapult. We refer to this spike in sharpness as 'sharpness catapult.'

An important consideration is the degree to which this phenomenon changes with network depth and width. Interestingly, we found that the training loss in deep networks on average catapults at much larger learning rates than $c = 2$. Fig. 2(d, e) shows that for a deep ($d = 10, w = 128$) CNN, the loss and sharpness may catapult only near the maximum trainable learning rate. In this section, we characterize the properties of the early training dynamics of models with MSE loss. In Appendix F, we show that a similar picture emerges for cross-entropy loss, despite the dynamics being noisier.

2.1 Loss and sharpness catapult during early training

In this subsection, we characterize the effect of finite depth and width on the onset of the loss and sharpness catapult and training divergence. We begin by defining critical constants that correspond to the above phenomena.

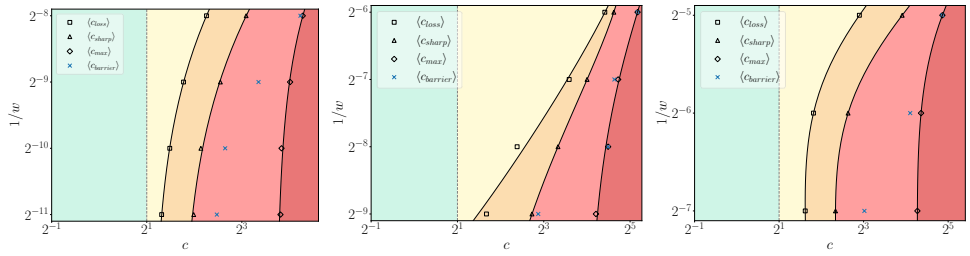
Definition 1. *($c_{loss}, c_{sharp}, c_{max}$) For learning rate $\eta = c/\lambda_0^H$, let the training loss and sharpness at step t be denoted by $\mathcal{L}_t(c)$ and $\lambda_t^H(c)$. We define $c_{loss}(c_{sharp})$ as minimum learning rates constants such that the loss (sharpness) increases during the initial transient period:*

$$c_{loss} = \min_c\{c \mid \max_{t\in[1,T_1]} \mathcal{L}_t(c) > \mathcal{L}_0(c)\}, \qquad c_{sharp} = \min_c\{c \mid \max_{t\in[1,T_1]} \lambda_t^H(c) > \lambda_0^H(c)\},$$

and c_{max} as the maximum learning rate constant such that the loss does not diverge during the initial transient period: $c_{max} = \max_c\{c \mid \mathcal{L}_t(c) < K, \forall t \in [1, T_1]\}$, where K is a fixed large constant.[1]

Note that the definition of c_{max} allows for more flexibility than previous studies [32] in order to investigate a wider range of phenomena occurring near the maximum learning rate. Here, c_{loss}, c_{sharp}, and c_{max} are random variables that depend on the random initialization and the SGD batch sequence, and we denote the average over this randomness using $\langle\cdot\rangle$.

[1]We use $K = 10^5$ to estimate c_{max} In all our experiments, $\mathcal{L}_0 = \mathcal{O}(1)$ (see Appendix A), which justifies the use of a fixed value.



(a) FCNs on MNIST (b) CNNs on Fashion-MNIST (c) ResNets on CIFAR-10

Figure 3: The phase diagrams of the early training of three different types of neural networks trained with MSE loss function using SGD. (a) FCNs ($d = 8$) trained on the MNIST dataset, (b) CNNs ($d = 7$) trained on the Fashion-MNIST dataset, (c) ResNet ($d = 18$) trained on the CIFAR-10 dataset (without batch normalization). Each data point in the figure represents an average of ten distinct initializations, and the solid lines represent a smooth curve fitted to the raw data points. The vertical dotted line shows $c = 2$ for comparison, and various colors are filled in between the various curves for better visualization. For experimental details and additional results for different depths, see Appendices A and C, respectively.

Fig. 3 illustrates the phase diagram of early training for three different architectures trained on various datasets with MSE loss using SGD. These phase diagrams show how the averaged values $\langle c_{loss} \rangle$, $\langle c_{sharp} \rangle$, and $\langle c_{max} \rangle$ are affected by width. The results show that the averaged values of all the critical constants increase significantly with $1/w$ (note the log scale). At large widths, the loss starts to catapult at $c \approx 2$. As $1/w$ increases, $\langle c_{loss} \rangle$ increases and eventually converges to $\langle c_{max} \rangle$ at large $1/w$. By comparison, sharpness starts to catapult at relatively large learning rates at small $1/w$, with $\langle c_{sharp} \rangle$ continuing to increase with $1/w$ while remaining between $\langle c_{loss} \rangle$ and $\langle c_{max} \rangle$. Similar results are observed for different depths as demonstrated in Appendix C. Phase diagrams obtained by varying d are qualitatively similar to those obtained by varying $1/w$. Comparatively, we observe that $\langle c_{max} \rangle$ may increase or decrease with $1/w$ in different settings while consistently increasing with d, as shown in Appendices F and H.

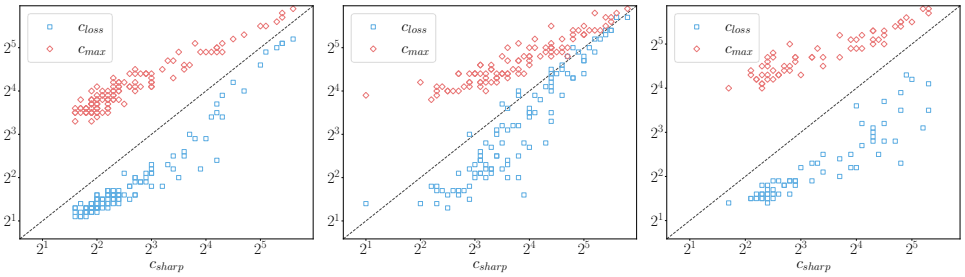
While we plotted the averaged quantities $\langle c_{loss} \rangle$, $\langle c_{sharp} \rangle$, $\langle c_{max} \rangle$, we have observed that their variance also increases significantly with d and $1/w$; in Appendix C we show standard deviations about the averages for different random initializations. Nevertheless, we have found that the inequality $c_{loss} \leq c_{sharp} \leq c_{max}$ typically holds, for any given initialization and batch sequences, except for some outliers due to high fluctuations when the averaged critical curves start merging at large d and $1/w$. Fig. 4 shows evidence of this claim. The setup is the same as in Fig. 3. Appendix D presents extensive additional results across various architectures and datasets.

In Appendix F, we show that cross-entropy loss shows similar results with some notable differences. The loss catapults at a relatively higher value $\langle c_{loss} \rangle \gtrsim 4$ and $\langle c_{max} \rangle$ consistently decreases with $1/w$, while still satisfying $c_{loss} \leq c_{sharp} \leq c_{max}$.

2.2 Loss connectivity in the early transient period

In the previous subsection, we observed that training loss and sharpness might quickly increase before decreasing ("catapult") during early training for a range of depths and widths. A logical next step is to analyze the region in the loss landscape that the training reaches after the catapult. Several works have analyzed loss connectivity along the training trajectory [20, 49, 61]. Ref. [49] report that training traverses a barrier at large learning rates, aligning with the naive intuition of a barrier between the initial and final points of the loss catapult, as the loss increases during early training. In this section, we will test the credibility of this intuition in real-world models. Specifically, we linearly interpolate the loss between the initial and final point after the catapult and examine the effect of the learning rate, depth, and width. The linearly interpolated loss and barrier are defined as follows.

Definition 2. $(\mathcal{L}_{int}(s, c), U(c))$ *Let θ_0 represent the initial set of parameters, and let θ_{T_1} represent the set of parameters at the end of the initial transient period, trained using a learning rate constant c. Then, we define the linearly interpolated loss as $\mathcal{L}_{int}(s, c) = \mathcal{L}[(1 - s)\ \theta_0 + s\ \theta_{T_1}]$, where $s \in [0, 1]$*



(a) FCNs on MNIST (b) CNNs on Fashion-MNIST (c) ResNets on CIFAR-10

Figure 4: The relationship between critical constants for (a) FCNs, (b) CNNs, and (c) ResNets. Each data point corresponds to a randomly initialized model. The dashed line represents the $y = x$ line.

is the interpolation parameter. The interpolated loss barrier is defined as the maximum value of the interpolated loss over the range of s: $U(c) = \max_{s\in[0,1]} \mathcal{L}_{int}(s) - \mathcal{L}(\theta_0)$.

Here we subtracted the loss's initial value such that a positive value indicates a barrier to the final point from initialization. Using the interpolated loss barrier, we define $c_{barrier}$ as follows.

Definition 3. *($c_{barrier}$) Given the initial (θ_0) and final parameters (θ_{T_1}), we define $c_{barrier}$ as the minimum learning rate constant such that there exists a barrier from θ_0 to θ_{T_1}:* $c_{barrier} = \min_c\{c \mid U(c) > 0\}$.

Here, $c_{barrier}$ is also a random variable that depends on the initialization and SGD batch sequence. We denote the average over this randomness using $\langle . \rangle$ as before. Fig. 2(c, f) shows the interpolated loss of CNNs trained on the CIFAR-10 dataset for $t = 10$ steps. The experimental setup is the same as in Section 2. For the network with larger width, we observe a barrier emerging at $c_{barrier} = 5.65$, while the loss starts to catapult at $c_{loss} = 2.83$. In comparison, we do not observe any barrier from initialization to the final point at large d and $1/w$. Fig. 3 shows the relationship between $\langle c_{barrier} \rangle$ and $1/w$ for various models and datasets. We consistently observe that $c_{sharp} \leq c_{barrier}$, suggesting that training traverses a barrier only when sharpness starts to catapult during early training. Similar results were observed on increasing d instead of $1/w$ as shown in Appendix C. We chose not to characterize the phase diagram of early training using $c_{barrier}$ as we did for other critical c's, as it is somewhat different in character than the other critical constants, which depend only on the sharpness and loss trajectories.

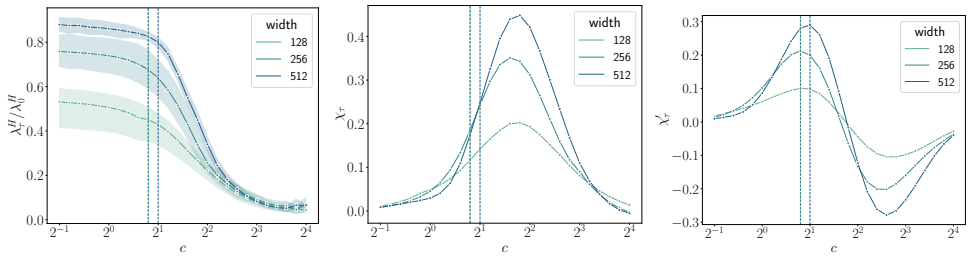
These observations call into question the intuition of catapulting out of a basin for a range of learning rates in between $c_{loss} < c < c_{barrier}$. These results show that for these learning rates, the final point after the catapult already lies in the same basin as initialization, and even *connected through a linear path*, revealing an inductive bias of the training process towards regions of higher loss during the early time transient regime.

3 Intermediate saturation regime

In the intermediate saturation regime, sharpness does not change appreciably and reflects the cumulative change that occurred during the initial transient period. This section analyzes sharpness in the intermediate saturation regime by studying how it changes with the learning rate, depth, and width of the model. Here, we show results for MSE loss, whereas cross-entropy results are shown in Appendix F.

We measure the sharpness λ_τ^H at a time τ in the middle of the intermediate saturation regime. We choose τ so that $c\tau \approx 200$.[2] For further details on sharpness measurement, see Appendix I.1. Fig. 5(a) illustrates the relationship between λ_τ^H and the learning rate for 7-layer deep CNNs trained on the CIFAR-10 dataset with varying widths. The results indicate that the dependence of λ_τ^H on learning rate can be grouped into three distinct stages. (1) At small learning rates, λ_τ^H remains relatively constant, with fluctuations increasing as d and $1/w$ increase ($c < 2$ in Fig. 5(a)). (2) A crossover regime where λ_τ^H is dropping significantly ($2 < c < 2^3$ in Fig. 5(a)). (3) A saturation

[2] time-step $\tau = 200/c$ is in the middle of regime (ii) for the models studied. Normalizing by c allows proper comparison for different learning rates.



(a) (b) (c)

Figure 5: (a) Normalized sharpness measured at $c\tau = 200$ against the learning rate constant for 7-layer CNNs trained on the CIFAR-10 dataset, with varying widths. Each data point is an average over 5 initializations, where the shaded region depicts the standard deviation around the mean trend. (b, c) Smooth estimations of the first two derivatives, χ_τ and χ'_τ, of the averaged normalized sharpness wrt the learning rate constant. The vertical lines denote c_{crit} estimated using the maximum of χ'_τ. For smoothening details, see Appendix I.2.

stage where λ_τ^H stays small and constant with learning rate ($c > 2^3$) in Fig. 5(a)). In Appendix I, we show that these results are consistent across architectures and datasets for varying values of d and w. Additionally, the results reveal that in stage (1), where $c < 2$ is sub-critical, λ_τ^H decreases with increasing d and $1/w$. In other words, for small c and in the intermediate saturation regime, the loss is locally flatter as d and $1/w$ increase.

We can precisely extract a critical value of c that separates stages (1) and (2), which corresponds to the onset of an abrupt reduction of sharpness λ_τ^H. To do this, we consider the averaged normalized sharpness over initializations and denote it by $\langle \lambda_\tau^H/\lambda_0^H \rangle$. The first two derivatives of the averaged normalized sharpness, $\chi_\tau = -\frac{\partial}{\partial c}\langle \lambda_\tau^H/\lambda_0^H \rangle$ and $\chi'_\tau = -\frac{\partial^2}{\partial c^2}\langle \lambda_\tau^H/\lambda_0^H \rangle$, characterize the change in sharpness with learning rate. The extrema of χ'_τ quantitatively define the boundaries between the three stages described above. In particular, using the maximum of χ'_τ, we define $\langle c_{crit} \rangle$, which marks the beginning of the sharp decrease in λ_τ^H with the learning rate.

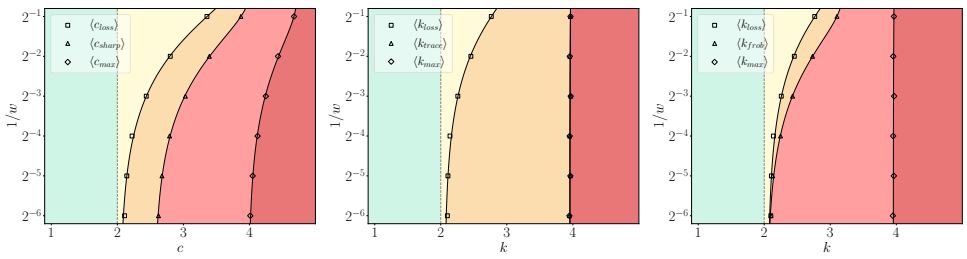
Definition 4. *($\langle c_{crit} \rangle$) Given the averaged normalized sharpness $\langle \lambda_\tau^H/\lambda_0^H \rangle$ measured at τ, we define c_{crit} to be the learning rate constant that minimizes its second derivative: $\langle c_{crit} \rangle = \arg\max_c \chi_\tau$.*

Here, we use $\langle . \rangle$ to denote that the critical constant is obtained from the averaged normalized sharpness. Fig. 5(b, c) show χ_τ and χ'_τ obtained from the results in Fig. 5(a). We observe similar results across various architectures and datasets, as shown in Appendix I. Our results show that $\langle c_{crit} \rangle$ has slight fluctuations as d and $1/w$ are changed but generally stay in the vicinity of $c = 2$. The peak in χ'_τ becomes wider as d and $1/w$ increase, indicating that the transition between stages (1) and (2) becomes smoother, presumably due to larger fluctuations in the properties of the Hessian H at initialization. In contrast to $\langle c_{crit} \rangle$, $\langle c_{loss} \rangle$ increase with d and $1/w$, implying the opening of the sharpness reduction phase $\langle c_{crit} \rangle < c < \langle c_{loss} \rangle$ as d and $1/w$ increase. In Appendix F, we show that cross-entropy loss shows qualitatively similar results, but with $4 \gtrsim \langle c_{crit} \rangle \gtrsim 2$.

4 Effect of network output at initialization on early training

Here we discuss the effect of network output $f(x; \theta_t)$ at initialization on the early training dynamics. x is the input and θ_t denotes the set of parameters at time t. In Appendix G, we consider setting the network output to zero at initialization, $f(x; \theta_0) = 0$, by either (1) considering the "centered" network: $f_c(x; \theta) = f(x; \theta) - f(x; \theta_0)$, or (2) setting the last layer weights to zero at initialization. Remarkably, both (1) and (2) removing the opening up of the sharpness reduction phase with $1/w$. The average onset of the loss catapult, diagnosed by $\langle c_{loss} \rangle$, becomes independent of $1/w$ and d.

We also study empirically the impact of the output scale [18, 5, 4] on early training dynamics. Given a network function $f(x; \theta)$, we define the scaled network as $f_s(x; \theta) = \alpha f(x; \theta)$, where α is a scalar, fixed throughout training. In Appendix H, we show that a large (resp. small) value of $\|f(x; \theta_0)\|$ relative to the one-hot encodings of the labels causes the sharpness to decrease (resp. increase) during



(a) (b) (c)

Figure 6: The phase diagram of the uv model trained with MSE loss using gradient descent with (a) the top eigenvalue of Hessian λ_t^H, (b) the trace of Hessian $\mathrm{tr}(H_t)$ and (c) the square of the Frobenius norm $\mathrm{tr}(H_t^T H_t)$ used as a measure of sharpness. In (a), the learning rate is scaled as $\eta = c/\lambda_0^H$, while in (b) and (c), the learning rate is scaled as $\eta = k/\mathrm{tr}(H_0)$. The vertical dashed line shows $c = 2$ ($k = 2$) for reference. Each data point is an average over 500 random initializations.

early training. Interestingly, we still observe an increase in $\langle c_{loss} \rangle$ with d and $1/w$, unlike the case of initializing network output to zero, highlighting the unique impact of output scale on the dynamics.

5 Insights from a simple model

Here we analyze a two-layer linear network [53, 57, 47], the uv model, which shows much of the phenomena presented above. Define $f(x) = \frac{1}{\sqrt{w}} v^T u x$, with $x, f(x) \in \mathbb{R}$. Here, $u, v \in \mathbb{R}^w$ are the trainable parameters, initialized using the normal distribution, $u_i, v_i \sim \mathcal{N}(0, 1)$ for $i \in \{1, \ldots, w\}$. The model is trained with MSE loss on a single training example $(x, y) = (1, 0)$, which simplifies the loss to $\mathcal{L}(u, v) = f^2/2$, and which was also considered in Ref. [47]. Our choice of $y = 0$ is motivated by the results of Sec. 4, which suggest that the empirical results of Sec. 2 are intimately related to the model having a large initial output scale $\|f(x; \theta_0)\|$ relative to the output labels. We minimize the loss using gradient descent (GD) with learning rate η. The early time phase diagram also shows similar features to those described in preceding sections (compare Fig. 6(a) and Fig. 3). Below we develop an understanding of this early time phase diagram in the uv model.

The update equations of the uv model in function space can be written in terms of the trace of the Hessian $\mathrm{tr}(H)$

$$f_{t+1} = f_t \left(1 - \eta\, \mathrm{tr}(H_t) + \frac{\eta^2 f_t^2}{w}\right), \qquad \mathrm{tr}(H_{t+1}) = \mathrm{tr}(H_t) + \frac{\eta f_t^2}{w} \left(\eta\, \mathrm{tr}(H_t) - 4\right). \tag{1}$$

From the above equations, it is natural to scale the learning rate as $\eta = k/\mathrm{tr}(H_0)$. Note that $c = \eta \lambda_0^H = k\lambda_0^H/\mathrm{tr}(H_0)$. Also, we denote the critical constants in this scaling as k_{loss}, k_{trace}, k_{max} and k_{crit}, where the definitions follow from Definitions 1 and 4 on replacing sharpness with trace and use $\langle . \rangle$ to denote an average over initialization. Figure 6(b) shows the phase diagram of early training, with $\mathrm{tr}(H_t)$ replaced with λ_t^H as the measure of sharpness and with the learning rate scaled as $\eta = k/\mathrm{tr}(H_0)$. Similar to Figure 6(a), we observe a new phase $\langle k_{crit} \rangle < k < \langle k_{loss} \rangle$ opening up at small width. However, we do not observe the loss-sharpness catapult phase as $\mathrm{tr}(H)$ does not increase during training (see Equation 1). We also observe $\langle k_{max} \rangle = 4$, independent of width.

In Appendix B.3, we show that the critical value of k for which $\langle \mathcal{L}_1/\mathcal{L}_0 \rangle > 1$ increases with $1/w$, which explains why $\langle k_{loss} \rangle$ increases with $1/w$. Combined with $\langle k_{crit} \rangle \approx 2$, this implies the opening up of the sharpness reduction phase as w is decreased.

To understand the loss-sharpness catapult phase, we require some other measure as $\mathrm{tr}(H)$ does not increase for $0 < k < 4$. As λ_t^H is difficult to analyze, we consider the Frobenius norm $\|H\|_F = \sqrt{\mathrm{tr}(H^T H)}$ as a proxy for sharpness. We define k_{frob} as the minimum learning rate such that $||H_t||_F^2$ increases during early training. Figure 6(c) shows the phase diagram of the uv model, with $||H_t||_F^2$ as the measure of sharpness, while the learning rate is scaled as $\eta = k/\mathrm{tr}(H_0)$. We observe the loss-sharpness catapult phase at small widths. In Appendix B.4, we show that the critical value of k for which $\langle ||H_1||_F^2 - ||H_0||_F^2 \rangle > 0$ increases from $\langle k_{loss} \rangle$ as $1/w$ increases. This explains the opening up of the loss catapult phase at small w in Fig. 6 (c).

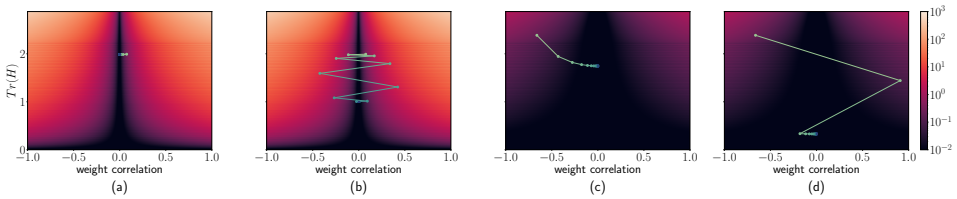


Figure 7: Training trajectories of the uv model trained on $(x, y) = (1, 0)$, with (a, b) large and (c, d) small width, in a two-dimensional slice of the parameters defined by the trace of Hessian $\mathrm{tr}(H)$ and weight correlation, trained with (a, c) small ($c = 0.5$) and (b, d) large ($c = 2.5$) learning rates. The colors correspond to the training loss $\mathcal{L}$, with darker colors representing a smaller loss.

Fig. 7 shows the training trajectories of the uv model with large ($w = 512$) and small ($w = 2$) widths in a two-dimensional slice of parameters defined by $\mathrm{tr}(H)$ and weight correlation $\langle v, u \rangle / ||u|| ||v||$. The above figure reveals that the first few training steps of the small-width network take the system in a flatter direction (as measured by $\mathrm{tr}(H)$) as compared to the wider network. This means that the small-width network needs a relatively larger learning rate to get to a point of increased loss (loss catapult). We thus have the opening up of a new regime $\langle k_{crit} \rangle < k < \langle k_{loss} \rangle$, in which the loss and sharpness monotonically decrease during early training.

The uv model trained on an example (x, y) with $y \neq 0$ provides insights into the effect of network output at initialization observed in Section 4. In Appendix G, we show that setting $f_0 = 0$ and $y \neq 0$ in the dynamical equations results in loss catapult at $k = 2$, implying $\langle k_{loss} \rangle \approx \langle k_{crit} \rangle \approx 2$, irrespective of w.

The loss landscape of the uv model shown in Fig. 7 reveals interesting insights into the loss landscape connectivity results in Section 2.2 and the presence of $c_{barrier}$. Fig. 7 shows how even when there is a loss catapult, as long as the learning rate is not too large, the final point after the catapult can be reached from initialization by a linear path without increasing the loss and passing through a barrier. However if the learning rate becomes large enough, then the final point after the catapult may correspond to a region of large weight correlation, and there will be a barrier in the loss upon linear interpolation.

6 Discussion

We have studied the effect of learning rate, depth, and width on the early training dynamics in DNNs trained using SGD with learning rate scaled as $\eta = c/\lambda_0^H$. We analyzed the early transient and intermediate saturation regimes and presented a rich phase diagram of early training with learning rate, depth, and width. We report two new phases, sharpness reduction and loss-sharpness catapult, which have not been reported previously. Furthermore, we empirically investigated the underlying cause of sharpness reduction during early training. Our findings show that setting the network output to zero at initialization effectively leads to the vanishing of sharpness reduction phase at supercritical learning rates. We further studied loss connectivity in the early transient regime and demonstrated the existence of a regime $\langle c_{loss} \rangle < c < \langle c_{barrier} \rangle$, in which the final point after the catapult lies in the same basin as initialization, connected through a linear path. Finally, we study these phenomena in a 2-layer linear network (uv model), gaining insights into the opening of the sharpness reduction phase.

We performed a preliminary analysis on the effect of batch size on the presented results in Appendix J. The sharpness trajectories of models trained with a smaller batch size ($B = 32$ vs. $B = 512$) show similar early training dynamics. In the early transient regime, we observe a qualitatively similar phase diagram. In the intermediate saturation regime, the effect of reducing the batch size is to broaden the transition around c_{crit}.

The early training dynamics is sensitive to the initialization scheme and optimization algorithm used, and we leave it to future work to explore this dependence and its implications. In this work, we focused on models initialized at criticality [52] as it allows for proper gradient flow through ReLU networks at initialization [22, 55], and studied vanilla SGD for simplicity. However, other initializations [44], parameterizations [66, 67], and optimization procedures [21] may show dissimilarities with the reported phase diagram of early training.

References

[1] Alessandro Achille, Matteo Rovere, and Stefano Soatto. Critical learning periods in deep networks. In *International Conference on Learning Representations*, 2019.

[2] Atish Agarwala, Fabian Pedregosa, and Jeffrey Pennington. Second-order regression models exhibit progressive sharpening to the edge of stability. *ArXiv*, abs/2210.04860, 2022.

[3] Sanjeev Arora, Zhiyuan Li, and Abhishek Panigrahi. Understanding gradient descent on edge of stability in deep learning. In *International Conference on Machine Learning*, 2022.

[4] Alexander Atanasov, Blake Bordelon, Sabarish Sainathan, and Cengiz Pehlevan. The onset of variance-limited behavior for networks in the lazy and rich regimes. In *The Eleventh International Conference on Learning Representations*, 2023.

[5] Blake Bordelon and Cengiz Pehlevan. Self-consistent dynamical field theory of kernel evolution in wide neural networks. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

[6] Blake Bordelon and Cengiz Pehlevan. Dynamics of finite width kernel and prediction fluctuations in mean field neural networks. *ArXiv*, abs/2304.03408, 2023.

[7] James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018.

[8] Jeremy Cohen, Simran Kaur, Yuanzhi Li, J Zico Kolter, and Ameet Talwalkar. Gradient descent on neural networks typically occurs at the edge of stability. In *International Conference on Learning Representations*, 2021.

[9] Alex Damian, Eshaan Nichani, and Jason D. Lee. Self-stabilization: The implicit bias of gradient descent at the edge of stability. In *OPT 2022: Optimization for Machine Learning (NeurIPS 2022 Workshop)*, 2022.

[10] Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

[11] Ethan Dyer and Guy Gur-Ari. Asymptotics of wide networks from feynman diagrams. In *International Conference on Learning Representations*, 2020.

[12] Gintare Karolina Dziugaite and Daniel M. Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. In Gal Elidan, Kristian Kersting, and Alexander Ihler, editors, *Proceedings of the Thirty-Third Conference on Uncertainty in Artificial Intelligence, UAI 2017, Sydney, Australia, August 11-15, 2017*. AUAI Press, 2017.

[13] Dumitru Erhan, Yoshua Bengio, Aaron Courville, Pierre-Antoine Manzagol, Pascal Vincent, and Samy Bengio. Why does unsupervised pre-training help deep learning? *Journal of Machine Learning Research*, 11(19):625–660, 2010.

[14] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations*, 2021.

[15] Stanislav Fort, Gintare Karolina Dziugaite, Mansheej Paul, Sepideh Kharaghani, Daniel M. Roy, and Surya Ganguli. Deep learning versus kernel learning: an empirical study of loss landscape geometry and the time evolution of the neural tangent kernel. In *NeurIPS*, 2020.

[16] Stanislav Fort and Surya Ganguli. Emergent properties of the local geometry of neural loss landscapes. *ArXiv*, abs/1910.05929, 2019.

[17] Jonathan Frankle, Gintare Karolina Dziugaite, Daniel M. Roy, and Michael Carbin. Linear mode connectivity and the lottery ticket hypothesis. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

[18] Mario Geiger, Stefano Spigler, Arthur Jacot, and Matthieu Wyart. Disentangling feature and lazy training in deep neural networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2020(11):113301, nov 2020.

[19] Justin Gilmer, Behrooz Ghorbani, Ankush Garg, Sneha Kudugunta, Behnam Neyshabur, David Cardoze, George Edward Dahl, Zachary Nado, and Orhan Firat. A loss curvature perspective on training instabilities of deep learning models. In *International Conference on Learning Representations*, 2022.

[20] Ian J. Goodfellow and Oriol Vinyals. Qualitatively characterizing neural network optimization problems. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[21] Priya Goyal, Piotr Dollár, Ross B. Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *ArXiv*, abs/1706.02677, 2017.

[22] Boris Hanin. Which neural net architectures give rise to exploding and vanishing gradients? In *Neural Information Processing Systems*, 2018.

[23] Boris Hanin and Mihai Nica. Finite depth and width corrections to the neural tangent kernel. In *International Conference on Learning Representations*, 2020.

[24] Boris Hanin and David Rolnick. How to start training: The effect of initialization and architecture. In *Neural Information Processing Systems*, 2018.

[25] Soufiane Hayou, A. Doucet, and Judith Rousseau. On the selection of initialization and activation function for deep neural networks. *ArXiv*, abs/1805.08266, 2018.

[26] Soufiane Hayou, A. Doucet, and Judith Rousseau. Exact convergence rates of the neural tangent kernel in the large depth limit. 2019.

[27] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1026–1034, 2015.

[28] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[29] Jonathan Heek, Anselm Levskaya, Avital Oliver, Marvin Ritter, Bertrand Rondepierre, Andreas Steiner, and Marc van Zee. Flax: A neural network library and ecosystem for JAX, 2020.

[30] Sepp Hochreiter and Jürgen Schmidhuber. Simplifying neural nets by discovering flat minima. *Advances in neural information processing systems*, 7, 1994.

[31] Sepp Hochreiter and Jürgen Schmidhuber. Flat minima. *Neural computation*, 9(1):1–42, 1997.

[32] Gaurav Iyer, Boris Hanin, and David Rolnick. Maximal initial learning rates in deep relu networks. *ArXiv*, abs/2212.07295, 2022.

[33] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: convergence and generalization in neural networks (invited paper). *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, 2018.

[34] Stanislaw Jastrzebski, Maciej Szymczak, Stanislav Fort, Devansh Arpit, Jacek Tabor, Kyunghyun Cho*, and Krzysztof Geras*. The break-even point on optimization trajectories of deep neural networks. In *International Conference on Learning Representations*, 2020.

[35] Stanisław Jastrzębski, Zac Kenton, Devansh Arpit, Nicolas Ballas, Asja Fischer, Amos Storkey, and Yoshua Bengio. Three factors influencing minima in SGD, 2018.

[36] Stanisław Jastrzębski, Zachary Kenton, Nicolas Ballas, Asja Fischer, Yoshua Bengio, and Amost Storkey. On the relation between the sharpest directions of DNN loss and the SGD step length. In *International Conference on Learning Representations*, 2019.

[37] Stanisław Jastrzębski, Zachary Kenton, Nicolas Ballas, Asja Fischer, Yoshua Bengio, and Amost Storkey. On the relation between the sharpest directions of DNN loss and the SGD step length. In *International Conference on Learning Representations*, 2019.

[38] Yiding Jiang*, Behnam Neyshabur*, Hossein Mobahi, Dilip Krishnan, and Samy Bengio. Fantastic generalization measures and where to find them. In *International Conference on Learning Representations*, 2020.

[39] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

[40] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. In *International Conference on Learning Representations*, 2017.

[41] Günter Klambauer, Thomas Unterthiner, Andreas Mayr, and Sepp Hochreiter. Self-normalizing neural networks. In *NIPS*, pages 972–981, 2017.

[42] Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009.

[43] Guillaume Leclerc and Aleksander Madry. The two regimes of deep network training. *ArXiv*, abs/2002.10376, 2020.

[44] Yann LeCun, Léon Bottou, Genevieve B. Orr, and Klaus-Robert Müller. Efficient backprop. In *Neural Networks*, 2012.

[45] Jaehoon Lee, Jascha Sohl-dickstein, Jeffrey Pennington, Roman Novak, Sam Schoenholz, and Yasaman Bahri. Deep neural networks as gaussian processes. In *International Conference on Learning Representations*, 2018.

[46] Jaehoon Lee, Lechao Xiao, Samuel S. Schoenholz, Yasaman Bahri, Roman Novak, Jascha Narain Sohl-Dickstein, and Jeffrey S. Pennington. Wide neural networks of any depth evolve as linear models under gradient descent. *Journal of Statistical Mechanics: Theory and Experiment*, 2020, 2019.

[47] Aitor Lewkowycz, Yasaman Bahri, Ethan Dyer, Jascha Sohl-Dickstein, and Guy Gur-Ari. The large learning rate phase of deep learning: the catapult mechanism. *ArXiv*, abs/2003.02218, 2020.

[48] Chunrui Liu, Wei Huang, and Richard Yi Da Xu. Implicit bias of deep learning in the large learning rate phase: A data separability perspective. *Applied Sciences*, 13(6), 2023.

[49] James Lucas, Juhan Bae, Michael R Zhang, Stanislav Fort, Richard Zemel, and Roger Grosse. Analyzing monotonic linear interpolation in neural network loss landscapes. *arXiv preprint arXiv:2104.11044*, 2021.

[50] David Meltzer and Junyu Liu. Catapult dynamics and phase transitions in quadratic nets. *ArXiv*, abs/2301.07737, 2023.

[51] Roman Novak, Lechao Xiao, Jiri Hron, Jaehoon Lee, Alexander A. Alemi, Jascha Sohl-Dickstein, and Samuel S. Schoenholz. Neural tangents: Fast and easy infinite neural networks in python. In *International Conference on Learning Representations*, 2020.

[52] Ben Poole, Subhaneil Lahiri, Maithra Raghu, Jascha Sohl-Dickstein, and Surya Ganguli. Exponential expressivity in deep neural networks through transient chaos. In *NIPS*, 2016.

[53] Jasson D. M. Rennie and Nathan Srebro. Fast maximum margin matrix factorization for collaborative prediction. In *Proceedings of the 22nd International Conference on Machine Learning*, ICML '05, page 713–719, New York, NY, USA, 2005. Association for Computing Machinery.

[54] Jasson D. M. Rennie and Nathan Srebro. Fast maximum margin matrix factorization for collaborative prediction. In *Proceedings of the 22nd International Conference on Machine Learning*, ICML '05, page 713–719, New York, NY, USA, 2005. Association for Computing Machinery.

[55] Daniel A. Roberts, Sho Yaida, and Boris Hanin. *The Principles of Deep Learning Theory*. Cambridge University Press, 2022. https://deeplearningtheory.com.

[56] Abraham. Savitzky and M. J. E. Golay. Smoothing and differentiation of data by simplified least squares procedures. *Analytical Chemistry*, 36(8):1627–1639, 1964.

[57] Andrew M. Saxe, James L. McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *CoRR*, abs/1312.6120, 2014.

[58] Vaishaal Shankar, Alexander W. Fang, Wenshuo Guo, Sara Fridovich-Keil, Ludwig Schmidt, Jonathan Ragan-Kelley, and Benjamin Recht. Neural kernels without tangents. In *ICML*, 2020.

[59] Jascha Narain Sohl-Dickstein, Roman Novak, Samuel S. Schoenholz, and Jaehoon Lee. On the infinite width limit of neural networks with a standard parameterization. *ArXiv*, abs/2001.07301, 2020.

[60] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.

[61] Tiffany J. Vlaar and Jonathan Frankle. What can linear interpolation of neural network loss landscapes tell us? In *International Conference on Machine Learning*, 2021.

[62] Zixuan Wang, Zhouzi Li, and Jian Li. Analyzing sharpness along GD trajectory: Progressive sharpening and edge of stability. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

[63] Lei Wu, Chao Ma, and Weinan E. How sgd selects the global minima in over-parameterized learning: A dynamical stability perspective. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[64] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017. cite arxiv:1708.07747Comment: Dataset is freely available at https://github.com/zalandoresearch/fashion-mnist Benchmark is available at http://fashion-mnist.s3-website.eu-central-1.amazonaws.com/.

[65] Sho Yaida. Meta-principled family of hyperparameter scaling strategies. *ArXiv*, abs/2210.04909, 2022.

[66] Greg Yang and Edward J. Hu. Tensor programs iv: Feature learning in infinite-width neural networks. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 11727–11737. PMLR, 18–24 Jul 2021.

[67] Greg Yang, Edward J. Hu, Igor Babuschkin, Szymon Sidor, Xiaodong Liu, David Farhi, Nick Ryder, Jakub W. Pachocki, Weizhu Chen, and Jianfeng Gao. Tensor programs v: Tuning large neural networks via zero-shot hyperparameter transfer. In *Neural Information Processing Systems*, 2022.

[68] Libin Zhu, Chaoyue Liu, Adityanarayanan Radhakrishnan, and Mikhail Belkin. Quadratic models for understanding neural network dynamics. *ArXiv*, abs/2205.11787, 2022.