

LOSS SPIKE IN TRAINING NEURAL NETWORKS*

Xiaolong Li

*School of Mathematical Sciences, Institute of Natural Sciences, MOE-LSC,
Shanghai Jiao Tong University, Shanghai 200240, China*

Zhi-Qin John Xu¹⁾

*School of Mathematical Sciences, Institute of Natural Sciences, MOE-LSC,
Shanghai Jiao Tong University, Shanghai 200240, China;*

*Key Laboratory of Marine Intelligent Equipment and System, Ministry of Education,
Shanghai 200240, China*

Email: xuzhiqin@sjtu.edu.cn

Zhongwang Zhang

*School of Mathematical Sciences, Institute of Natural Sciences, MOE-LSC,
Shanghai Jiao Tong University, Shanghai 200240, China*

Abstract

In this work, we investigate the mechanism underlying loss spikes observed during neural network training. When the training enters a region with a lower-loss-as-sharper structure, the training becomes unstable, and the loss exponentially increases once the loss landscape is too sharp, resulting in the rapid ascent of the loss spike. The training stabilizes when it finds a flat region. From a frequency perspective, we explain the rapid descent in loss as being primarily influenced by low-frequency components. We observe a deviation in the first eigendirection, which can be reasonably explained by the frequency principle, as low-frequency information is captured rapidly, leading to the rapid descent. Inspired by our analysis of loss spikes, we revisit the link between the maximum eigenvalue of the loss Hessian (λ_{\max}), flatness and generalization. We suggest that λ_{\max} is a good measure of sharpness but not a good measure for generalization. Furthermore, we experimentally observe that loss spikes can facilitate condensation, causing input weights to evolve towards the same direction. And our experiments show that there is a correlation (similar trend) between λ_{\max} and condensation. This observation may provide valuable insights for further theoretical research on the relationship between loss spikes, λ_{\max} , and generalization.

Mathematics subject classification: 65N38, 65N30.

Key words: Neural Network, Loss Spike, Frequency Principle, Maximum Eigenvalue, Flatness, Generalization, Condensation.

1. Introduction

Many experiments have observed a phenomenon, called the edge of stability (EoS) [4,8,10,43, 57], that learning rate (η) and sharpness (i.e. the largest eigenvalue of Hessian) no longer behave as in traditional optimization, sharpness hovers at $2/\eta$ while the loss continues decreasing, albeit non-monotonically. Training with a larger learning rate leads to a solution with smaller λ_{\max} . Since λ_{\max} is often used to indicate the sharpness of the loss landscape, a larger learning rate results in a flatter solution. Intuitively as shown in Fig. 1.1, the flat solution is more robust

* Received April 16, 2024 / Revised version received September 19, 2024 / Accepted December 2, 2024 /

Published online January 13, 2025 /

¹⁾ Corresponding author

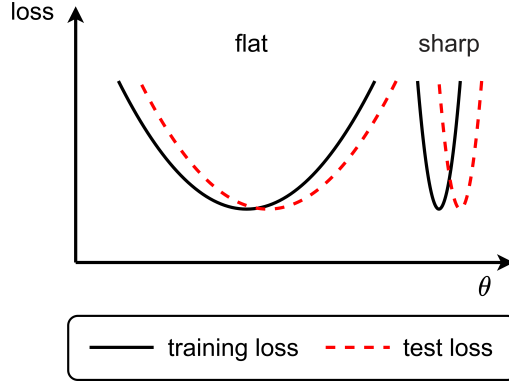


Fig. 1.1. Schematic illustration of an ideal explanation for why flat solutions generalize well [23].

to perturbation and has better generalization performance [19, 23]. Therefore, training with a larger learning rate would achieve better generalization performance. In this work, we argue this intuitive analysis in Fig. 1.1 with λ_{\max} as the sharpness measure, which encounters difficulty in NNs through the study of loss spikes.

In a neural network training process, one may sometimes observe a phenomenon of loss spike (typical examples in Fig. 3.1), where the loss rapidly ascends and then descends to the value before the ascent. We show a special loss landscape structure underlying the loss spike, which is called a lower-loss-as-sharper (LLAS) structure. In the LLAS structure, the training is driven by descending the loss while entering an increasingly sharp region. Once the sharpness is too large, the loss would ascend exponentially fast. To explain why the loss can descend so fast, we provide a frequency perspective analysis. We find that the deviation in the ascending stage is dominated by low-frequency components. Based on the frequency principle [47, 48] that low-frequency converges faster than high-frequency, we rationalize the fast descent.

The study of loss spike provides an important information that the deviation at the first eigendirection is dominated by low-frequency. We then further argue the link between λ_{\max} flatness and generalization. In real-world datasets, low-frequency information is often dominant and well-captured by both the training data and the test data. Therefore, the training can learn low-frequency well. Since the sharpest direction, indicated by the maximum eigenvalue of the loss Hessian, relates more to the low-frequency, a solution with good generalization and a solution with bad generalization have little difference in the sharpest direction, verified by a series of experiments. Hence, λ_{\max} with the intuitive explanation in Fig. 1.1 encounters difficulty in understanding the generalization of neural networks, such as why a larger learning rate results in better generalization for networks with EoS training.

We also find that a loss spike can facilitate condensation, that is, the input weights of different neurons in the same layer evolve towards the same, which would reduce the network's effective size. Condensation is a non-linear feature learning phenomenon in neural networks, which may be the underlying mechanism for why the loss spike improves generalization [18, 22], rather than simply controlling the value of λ_{\max} .

We believe that this work makes contributions in the following aspects:

- (1) Analyzing the loss spike phenomenon and its frequency mechanism,
- (2) Proposing the LLAS structure to explain the ascent stage of loss spikes,

- (3) Revisiting the relationship between flatness and generalization from the frequency perspective,
- (4) Preliminarily revealing the correlation among loss spikes, maximum eigenvalues, and the condensation phenomenon.

2. Related Works

Previous works [2,10,31,42,43,45] conduct an extensive study of the EoS phenomenon under various settings. It has been observed that when the initial sharpness exceeds $2/\eta$, gradient descent “catapults” into a stable region and converges [24]. Progressive sharpening and the edge of stability phenomenon have been analyzed under specific settings, such as normalized gradient descent [4]. It has been shown that the third-order terms bias towards flatter minima to understand EoS [12].

The loss landscape’s subquadratic structure, i.e. the maximum eigenvalue of the loss Hessian being larger when the loss is lower in a direction, has been attributed to the progressive sharpening phenomenon [32]. This work also proposes a flatness-driven motion to study the EoS stage, that is, the training would move towards a flatter minimum, such that the fixed flatness can correspond to points with lower and lower loss values due to the subquadratic property. We call this structure a lower-loss-as-flatter (LLAF) structure. The LLAF structure should expect a continuous decrease in the loss rather than a loss spike. A quadratic regression model with mean squared error (MSE) has been used to study EoS, but in this model, the loss spike cannot occur [1]. The loss spike has been studied from the perspective of adaptive gradient optimization algorithms [33], while in this paper, the focus is on the loss landscape structure and gradient descent training.

A series of works link the generalization performance of solutions to the landscape of loss functions through the observation that flat minima tend to generalize better [15,19,34,44]. Algorithms that favor flat solutions are designed to improve the generalization of the model [7,13,16,20,27,54]. On the other hand, it has been shown that sharp minima can also generalize well by rescaling the parameters at a flat minimum with ReLU activation [14]. In this work, the relationship between flatness and generalization is studied from a new perspective, i.e. the frequency perspective, without the limitation of the activation function.

It has been identified that the linear regime and the condensed regime of parameter initialization for two-layer and three-layer wide ReLU neural networks play a crucial role in determining the network’s final fitting result [30,55]. The training dynamics of NNs are approximately linear and similar to a random feature model. On the contrary, in the condensed regime, active neurons are condensed at several discrete orientations. At this point, the network is equivalent to another network with a reduced width, which may explain why NNs outperform traditional algorithms [6,50]. For the initial stage of training, A series of works [9,35,36,56] study the characteristics of the initial condensation for different activation functions. Du *et al.* [15] proved that gradient descent achieves zero training loss in polynomial time in the linear regime of a deep overparameterized neural networks with residual connections (ResNet).

Usually, loss spikes occur when the learning rate is relatively high. This behavior is often linked to the learning dynamics of the model under different learning rates. A significant amount of work [38] has been dedicated to exploring why neural networks trained with large learning rates for a longer time often lead to better generalization. Andriushchenko *et al.* [3] finds that stochastic gradient descent (SGD) with a large learning rate can facilitate sparse

solutions, attributing this effect to the noise structure inherent in SGD. It has been found [26] that a large learning rate model first learns easier-to-fit patterns and is unable to memorize hard-to-fit patterns, leading to a plateau in accuracy. Once the learning rate is annealed, it is able to fit these patterns, explaining the sudden spike in both train and test accuracy. In our work, we find that for the noise-free full-batch gradient descent algorithm, the loss spike can also facilitate the condensation phenomenon, implying that the noise structure is not the intrinsic cause of condensation.

The frequency principle is examined in extensive datasets and deep neural network models [37, 48, 49]. Subsequent theoretical studies show that the frequency principle holds in the general setting with infinite samples [29]. An overview for frequency principle is referred to Xu *et al.* [46]. Based on the theoretical understanding, the frequency principle inspires the design of deep neural networks to learn a function with high-frequency fast [5, 21, 28].

3. Loss Spike

3.1. Preliminary: Linear stability in training quadratic model

We consider a simple quadratic model with the loss $R(\theta) = \lambda\theta^2/2$ trained by gradient descent with learning rate η , $\theta(t+1) = \theta(t) - \eta \cdot dR(\theta)/d\theta$. To ensure the linear stability of the training, it is required that $|\theta(t+1)| < |\theta(t)|$, which implies $|1 - \lambda\eta| < 1$. Otherwise, the training will diverge. Note that λ is the Hessian of $R(\theta)$. Similarly, to ensure the linear stability of training a neural network, it requires that the maximum eigenvalue of the loss Hessian is smaller than $2/\eta$, i.e. 2 over the learning rate. Therefore, the maximum eigenvalue of the loss Hessian is often used as the measure of the sharpness of the loss landscape. In this section, we study the phenomenon of loss spike, where the loss would suddenly increase and decrease rapidly. For example, as shown in Fig. 3.1(a,d,g,h), we train a tanh fully-connected neural network (FNN) with 20 hidden neurons for a one-dimensional fitting problem, a ReLU convolutional neural network (CNN) for the CIFAR10-1k (a subset of the well-known CIFAR-10 dataset, which includes 1,000 images from 10 different classes) classification problem with MSE, and two VGG-11 [40] models with different learning rates for the CIFAR-10 dataset with datasize 1024. These models experience loss spikes. The red curves, i.e. the λ_{\max} value, show that the loss spikes occur at the EoS stage.

3.2. Typical loss spike experiments

To observe the loss spike clearly, we zoom in on the training epochs around the spike, shown in Fig. 3.1(b,e). The selected epochs are marked green in Fig. 3.1(a,d). When the maximum eigenvalue of Hessian λ_{\max} (red) exceeds $2/\eta$ (black dashed line), the loss increases, and when $\lambda_{\max} < 2/\eta$, the loss decreases, which are consistent with the linear stability analysis.

We then study the parameter space for more detailed characterization. Given t training epochs, and let θ_i denote model parameters at epoch i , we apply principal component analysis (PCA) to the matrix $M = [\theta_1 - \theta_t, \dots, \theta_t - \theta_t]$, and then select the first two eigendirections e_1, e_2 . The two-dimensional loss surface based on e_1 and e_2 can be calculated by $R_S(\theta_t + \alpha e_1 + \beta e_2)$, where α, β are the step sizes, and R_S is the loss function under the dataset S . The trajectory point of parameter θ_i can be calculated by the projection of $\theta_i - \theta_t$ in the PCA directions, i.e. $(\langle \theta_i - \theta_t, e_1 \rangle, \langle \theta_i - \theta_t, e_2 \rangle)$. Parameter trajectories (blue dots) and loss surfaces along PCA directions are shown in Fig. 3.1(c,f,i). In three distinct examples, they

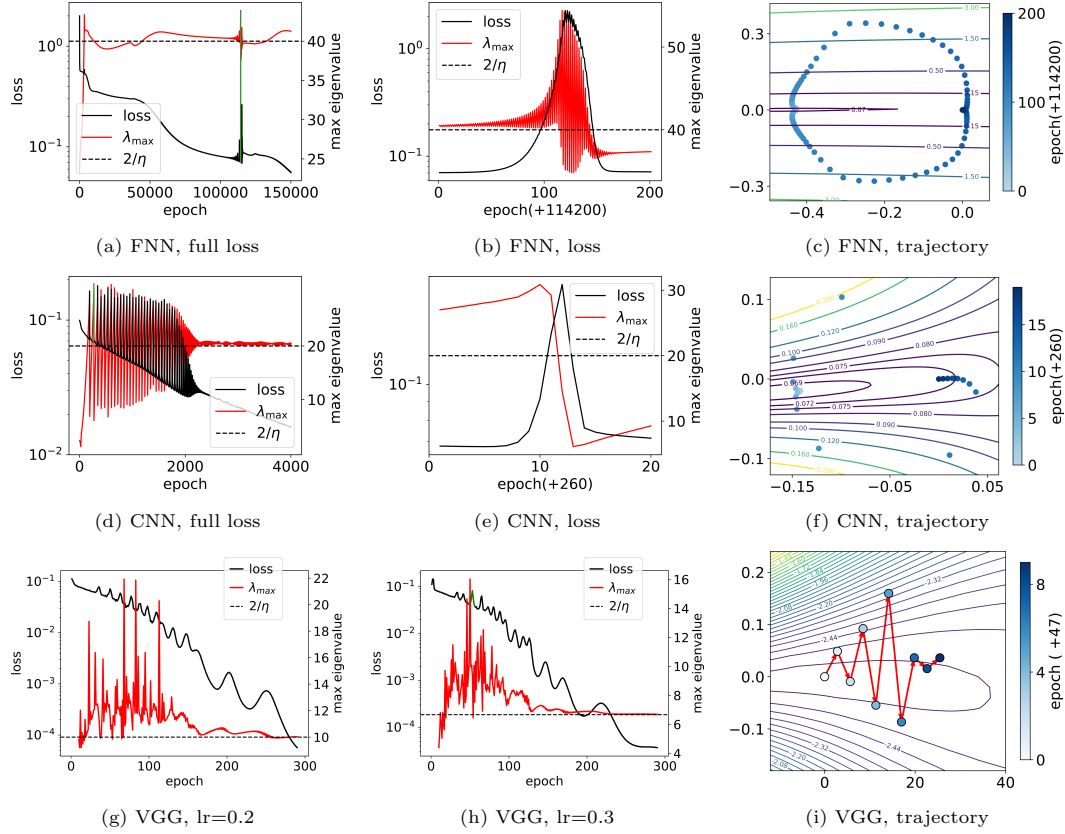


Fig. 3.1. (a, d, g, h) The loss value (black) and λ_{\max} (red) vs. training epoch. (b, e) The loss value and λ_{\max} of a specific epoch interval, which is marked green in (a, d), respectively. (c, f, i) The loss surface and the trajectory of the model parameters along the first two PCA directions. (a, b, c) Two-layer tanh NN with width 20. The sum of the explained variance ratios of the first two PCA directions is 0.9895. (d, e, f) Two-layer ReLU CNN with max pooling. The sum of the explained variance ratios of the first two PCA directions is 0.9882. (g, h, i) VGG-11 [40] with different learning rates. The sum of the explained variance ratios of the first two PCA directions is 0.9999.

exhibit similar behaviors. At the beginning of the ascent stage of the spike, the parameter is at a small-loss region, where the opening of the contour lines is towards the left, indicating a leftward component of descent direction. In the left region, the contour lines are denser, implying a sharper loss surface. Once $\lambda_{\max} > 2/\eta$, the parameters become unstable, and the loss value increases exponentially. In the high-loss region, the opening of the contour shifts to the right, indicating a rightward component of the descent direction, resulting in a sparser contour, i.e. a flatter loss surface. After several steps, when $\lambda_{\max} < 2/\eta$, the training returns to the stable stage.

3.3. Lower-loss-as-sharper structure

The above experiments reveal a common structure that causes a loss spike, namely, the λ_{\max} sharpness increases in the direction of decreasing loss. We call this structure lower-loss-as-sharper structure. The LLAS structure, which is common in the EoS stage as shown in

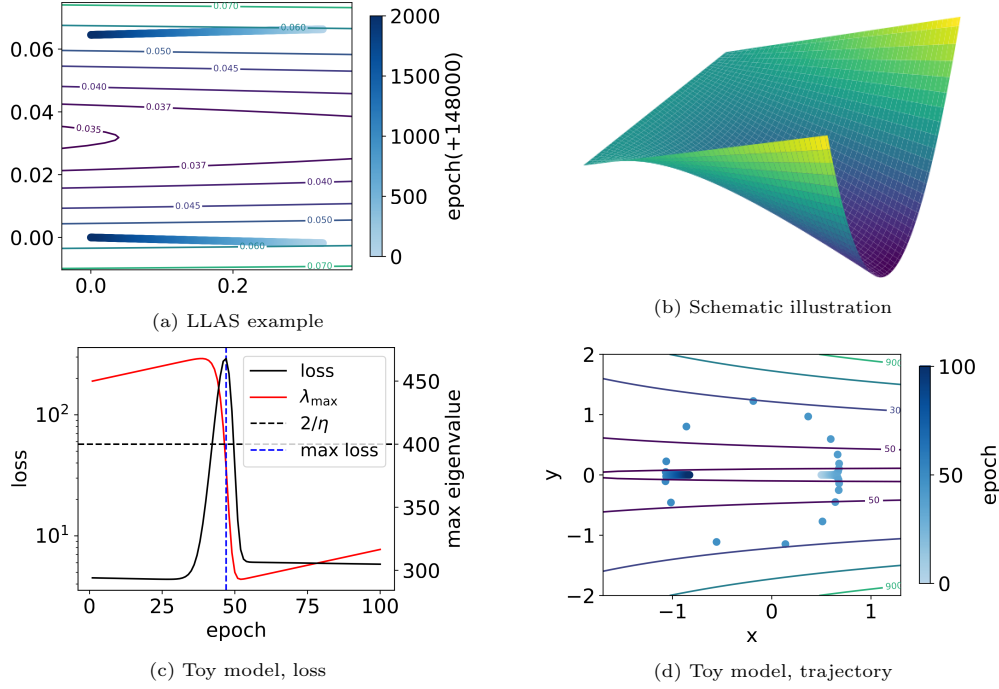


Fig. 3.2. (a) The loss surface and the trajectory of the model parameters along the first two PCA directions in the EoS stage. (b) Schematic illustration of LLAS structure in 3D. (c) The loss value and the maximum eigenvalue of the Hessian matrix of a loss spike process of the toy model. (d) The loss surface and the GD trajectory of the two-dimensional parameters of the toy model.

Fig. 3.2(a), differs from the lower-loss-as-flatter (LLAF) structure studied in [32]. The following quadratic model is a simple example of the LLAS structure:

$$f(x, y) = (50x + 200)y^2 - x + 5, \quad (3.1)$$

where $(x, y) \in (-4, +\infty) \times \mathbb{R}$.

For any constant C , $y = 0$ is the minimum point of $f(C, y)$. The larger x is, the lower the loss value is, and the sharper the loss landscape in the y -direction is. The intuitive explanation for the above phenomenon is that as x increases, $f(x, 0)$ decreases, which means that $f(x, 0)$ has a smaller value at the sharp region, i.e. the LLAS structure, which makes the opening of the contour lines towards different directions at different loss levels.

This model can be considered as being obtained by constraining the parameters of the linear neural network.

The toy example of the LLAS structure is shown in Fig. 3.2(b). As shown in Fig. 3.2(c,d), the loss curve and the trajectory of parameters are similar to the realistic example above, where the parameters move toward the sharp direction at the beginning of the loss spike, and then move toward the flat direction.

For this example, we can exactly compute the derivative of Eq. (3.1) as follows:

$$\begin{aligned} \frac{\partial f(x, y)}{\partial x} &= 50y^2 - 1, \\ \frac{\partial f(x, y)}{\partial y} &= (100x + 400)y. \end{aligned}$$

Thus, we have

$$\frac{\partial f(x, y)}{\partial x} \begin{cases} > 0, & \text{if } f(x, y) > 9, \\ = 0, & \text{if } f(x, y) = 9, \\ < 0, & \text{if } f(x, y) < 9, \end{cases}$$

which indicates that the toy model has a negative gradient component in the x direction when the parameters are in the small-loss region ($f(x, y) < 9$), while a positive gradient component in the x direction when the parameters are in the high-loss region ($f(x, y) > 9$) (more details in Appendix A.1).

Although the LLAS structure can explain the mechanism of the ascent stage based on the toy model, it can not explain the reason for the rapid descent of the loss in the descent phase of the loss spike, which takes much fewer steps than the training from the same level loss at the initialization. Moreover, due to the high dimensionality of the parameter space, the parameter trajectory does not always align with the first eigendirection, otherwise, as shown in the toy model, the loss would not decrease continuously. In the following, we take a step toward understanding the rapid decrease from the frequency perspective.

3.4. Frequency perspective for understanding descent stage

In this subsection, we study the mechanism of the rapid loss descent during the descent stage in a loss spike from the perspective of frequency. The “frequency” means response frequency which is the frequency of a general Input-Output mapping [47].

Our analysis is based on the commonly observed phenomenon known as the frequency principle [29, 37, 39, 47, 48, 52], which states that deep NNs often fit target functions from low to high frequencies during the training. Compared to the peak point of the loss spike with the point with the same loss value at the initial training, the descent during the spike should eliminate more low-frequency component with a fast speed while the descent from the initial model should eliminate more high-frequency component with a slow speed. To verify this conjecture, we study the frequency distribution of the converged part during the descent stage.

The parameter at the peak of the loss spike is denoted as θ_{\max} , and the parameter at the initial point which has the similar loss to the loss of θ_{\max} is denoted as $\theta_{\text{ini},m}$, the parameter at the end of the loss spike¹⁾ is denoted as θ_{end} . We then study the frequency distribution of spike output difference $f_{\text{peak,diff}} := f_{\theta_{\max}} - f_{\theta_{\text{end}}}$ and initial output difference $f_{\text{ini,diff}} := f_{\theta_{\text{ini},m}} - f_{\theta_{\text{end}}}$.

For comparison, we also randomly construct a parameter (ensuring that the magnitude of the random perturbation is consistent with the parameter magnitude at the loss spike), and then we use it as a control experiment to compare the frequency of differences in output

$$\theta_{\text{rnd}} := \theta_{\text{end}} + \left(\frac{\|\theta_{\text{end}} - \theta_{\max}\|_2}{\|\varepsilon\|_2} \right) \varepsilon,$$

where $\varepsilon \sim N(0, I)$ is a random variable.

We then study the frequency distribution of the random output difference $f_{\text{rnd,diff}} := f_{\theta_{\text{rnd}}} - f_{\theta_{\text{end}}}$. This comparison is made to demonstrate that the low-frequency proportion of the parameters at points where a loss spike occurs is significantly higher than in situations where no loss spike is observed, even exceeding the range of a random perturbation.

We characterize the frequency distribution by taking different low-frequency thresholds to study low-frequency proportion.

¹⁾ This point is selected roughly during the slow descent because, experimentally, it does not affect the result.

Definition 3.1. For a low-frequency threshold K , a low-frequency proportion (LFP) is defined as follows to characterize the power proportion of the low-frequency component over the whole spectrum:

$$\text{LFP}(K) = \frac{\sum_{k \leq K} \|\hat{f}_{\theta}(k)\|^2}{\sum_k \|\hat{f}_{\theta}(k)\|^2}, \quad (3.2)$$

where \hat{f}_{θ} indicates the Fourier transform of function f_{θ} .

As shown in Fig. 3.3(a), the low-frequency proportion of the spike output difference is significantly larger than the low-frequency proportion of the initial output difference and the random output difference, where we take 100 samples of random variable ε for the mean value and the error bar for each low-frequency threshold. Fig. 3.3(b) shows the training loss and the low-frequency proportion over epochs, where the low-frequency threshold is 2. This illustrates that the low-frequency proportion of the output increases dramatically with the occurrence of the loss spike. Fig. 3.3 verifies that the higher low-frequency proportion of the spike output difference is the key reason for the rapid drop in the loss value during the descent stage, as suggested by the frequency principle.

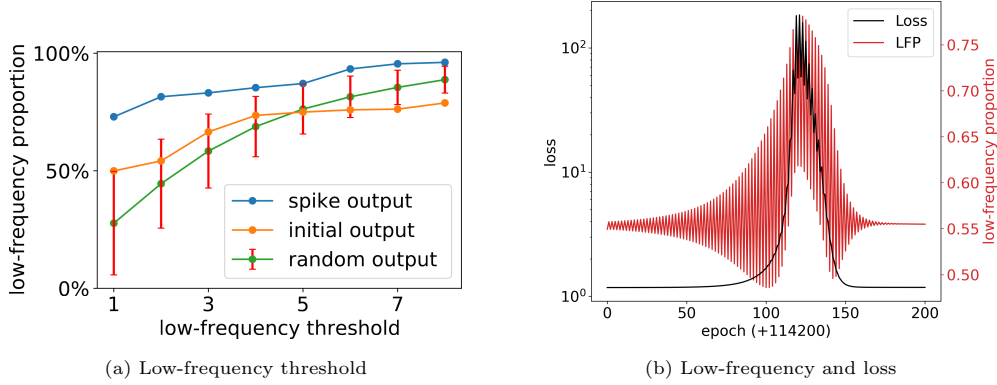


Fig. 3.3. (a) Low-frequency proportion for different low-frequency thresholds. The NN we used is a two-layer tanh NN with width 20. For the random output difference, we calculate the mean value and the error bar with 100 random samples. (b) Train loss and low-frequency proportion for low-frequency threshold is equal to 2 for different epochs.

4. Revisit the Flatness-generalization Picture

Motivated by the loss spike analysis from the frequency perspective, we further revisit the common flatness-generalization picture. A series of previous works [19, 25] attempt to link the flatness of the loss landscape with generalization, so as to characterize the model through flatness conveniently. A classic empirical illustration is shown in Fig. 1.1, which vividly expresses the reason why flat solutions tend to have better generalization. Usually, the training loss landscape and the test landscape do not exactly coincide due to sampling noise. A flat solution would be robust to the perturbation while a sharp solution would not. For such a one-dimensional case, this analysis is valid, but the loss landscape of a NN case is very high-dimensional, and such simple visualization or explanation is yet to be validated.

The first eigendirection of the loss Hessian, i.e. the eigendirection corresponding to the maximum eigenvalue, is the sharpest direction. Based on the flatness-generalization picture, it is natural to use the maximum eigenvalue as the measure for the flatness, which can also indicate generalization. However, this naive analysis is not always correct for neural networks.

4.1. Frequency perspective

The maximum eigenvalue of the loss Hessian can indicate the linear stability of the training process, and is often used as a measure for flatness/sharpness of the loss landscape, where a larger maximum eigenvalue indicates a sharper landscape. As shown by linear stability analysis, once the maximum eigenvalue exceeds $2/\eta$ (η is the learning rate), the training would oscillate and diverge along the first eigendirection. Meanwhile, as the parameters move away from the minimum point along this first eigendirection, the loss spike arises mainly due to the large low-frequency difference, as shown in Fig. 3.3. Therefore, the deviation in the first eigendirection of the loss Hessian primarily leads to the deviation of low-frequency components.

In order to examine the above analysis, we first obtain the model parameter θ_{train} with poor generalization by training the model initialized in the linear regime [30], and then further train the model parameter θ_{train} on the test dataset with a small learning rate to obtain the model parameter θ_{test} .

Let \mathbf{H} be the Hessian matrix of the loss function at the training parameters θ_{train} . λ_i is the i -th eigenvalue of the Hessian matrix \mathbf{H} with $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_N$, and ν_i is the corresponding eigendirection. We study the impact of each eigendirection on the test loss by eliminating the difference between θ_{train} and θ_{test} in the i -th eigendirection ν_i , where i is the index of eigenvalues.

As shown in Fig. 4.1(a), we study the change of the test loss $L(i)$ with the eigenvalue index i as follows to study the effect of eigenvectors on generalization,

$$L(i) = R_{S_{\text{test}}} \left(\theta_{\text{train}} + \sum_{j=1}^i \langle \theta_{\text{test}} - \theta_{\text{train}}, \nu_j \rangle \nu_j \right),$$

where S_{test} is the test dataset. The movement of parameters on the eigenvectors corresponding to large eigenvalues has a weak impact on the test loss, while the movement of parameters on the eigenvectors corresponding to small eigenvalues has a significant impact on the test loss.

A reasonable explanation from the perspective of frequency is as follows. In common datasets, low-frequency components often dominate over high-frequency ones. For noisy sampling, the dominant low-frequency is shared by both the training and the test data. When the parameters move along the eigendirections corresponding to the large eigenvalues, the network output often changes at low-frequency, which is already captured by both θ_{train} and θ_{test} . Therefore, the improvement of model generalization often requires certain high-frequency changes. As shown in Fig. 4.1(b), we move the corresponding θ_{train} along the first nine eigendirections, and show the difference between the network outputs before and after the movement, i.e. $f_{\theta_{\text{train}} + \nu_i / \sqrt{\lambda_i}} - f_{\theta_{\text{train}}}$, where the $1/\sqrt{\lambda_i}$ item is to make the loss of the network moved in different eigendirections approximately the same. From the difference between the outputs before and after the movement, it can be seen that when the parameters move along the eigendirection corresponding to the larger eigenvalue, the change of the model output is often less oscillated, i.e. dominated by the lower-frequency. Since the low-frequency is captured by both θ_{train} and θ_{test} , they should be close in the eigendirections corresponding to large eigenvalues, which is verified in the following subsection.

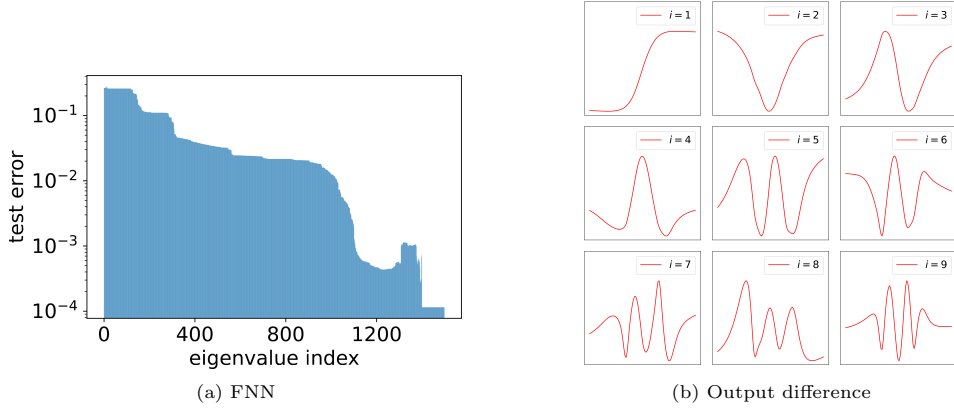


Fig. 4.1. Two-layer tanh FNN with a width of 500. (a) The variation of the test loss with the eigenvalue index i when eliminating the difference between θ_{train} and θ_{test} in the first i eigendirections. (b) The output difference before and after moving θ_{train} in the first nine eigendirections of its Hessian matrix. Each subset corresponds to the case of one eigendirection.

4.2. Difference on each eigendirection

We then examine the projection of $\theta_{\text{test}} - \theta_{\text{train}}$ in each eigendirection of $H(\theta_{\text{train}})$. As shown in Fig. 4.2, we show the projection of $\theta_{\text{test}} - \theta_{\text{train}}$ on each eigenvector ν_i (blue bar) for the FNN on function fitting problem and the CNNs on CIFAR10 classification problem. Due to

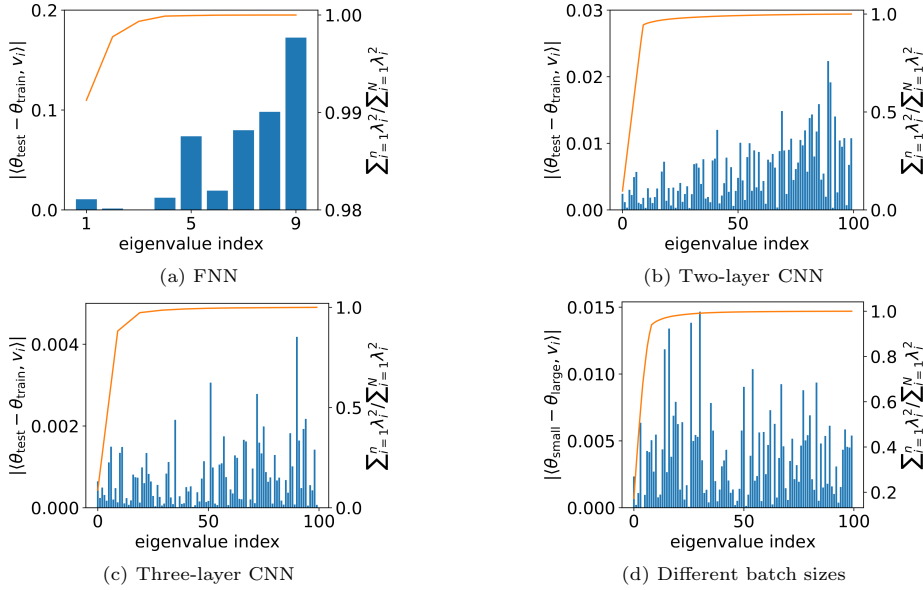


Fig. 4.2. Blue bar: (a, b, c) show the projection values of in each eigendirection of $H(\theta_{\text{train}})$ for $\theta_{\text{test}} - \theta_{\text{train}}$, and (d) for $\theta_{\text{large}} - \theta_{\text{small}}$. Orange line: the sum of the first n eigenvalues over all eigenvalues. (a) Two-layer tanh FNN for the one-dimensional fitting problem. (b) Two-layer ReLU CNN with max pooling for the CIFAR10 classification problem. (c) Three-layer ReLU CNN with max pooling for the CIFAR10 classification problem. (d) Five-layer ReLU CNN with max pooling for the CIFAR10 classification problem.

the high complexity of calculating the eigenvectors of the large-size Hessian matrix, we use the Lanczos method [11] to numerically compute the first N eigenvalues and their corresponding eigenvectors. For $s < N$, we use $\sum_{i=1}^s \lambda_i^2 / \sum_{i=1}^N \lambda_i^2$ to represent the explained variance ratio, i.e. to measure how much flatness information the first n eigendirections (orange line) can explain. For different network structures and model tasks, the projection value of $\theta_{\text{test}} - \theta_{\text{train}}$ on the eigenvector ν_i has a positive correlation with the eigenvalue index i , which confirms that θ_{train} and θ_{test} have little difference on low-frequency part. It has been verified through a series of studies [22] that models with different batch sizes exhibit varying generalization abilities. Note that in Fig. 4.2(d), the two minima, θ_{small} and θ_{large} , found by small and large batch sizes, respectively, have little difference in eigendirections corresponding to the largest s eigenvalues which are enough that $\sum_{i=1}^s \lambda_i^2 / \sum_{i=1}^N \lambda_i^2$ is close to 1. This indicates that the differences in the eigendirections corresponding to the largest eigenvalues are not significant, which is consistent with the conclusions drawn from Fig. 4.2(a,b,c).

4.3. Implications

As shown in Fig. 4.3, we revisit the link between λ_{max} , flatness and generalization. First, we experimentally demonstrate that the difference in model output during the loss spike is mainly dominated by low-frequency components (Section 4.1), and these low-frequency components often have no significant impact on the model’s generalization ability (Section 4.2). Additionally, the loss spike is often accompanied by a decrease in lambda max (as concluded in Section 3), indicating an improvement in flatness based on the maximum eigenvalue. Therefore, our core conclusion is that flatness based on the maximum eigenvalue cannot directly characterize the model’s generalization ability.

The above analysis suggests the following implications:

- (i) The maximum eigenvalue of the loss Hessian is a good measure of sharpness for whether the training is linearly stable but not a good measure for generalization.
- (ii) The common low-dimensional flatness-generalization perspective faces challenges in comprehending the high-dimensional loss landscape of neural networks.

Generalization performance is a combined effect of most eigendirections, including those associated with small eigenvalues.

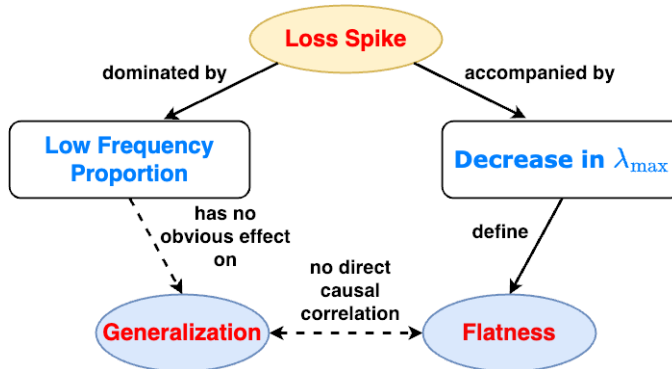


Fig. 4.3. Flatness-generalization relationship.

5. Loss Spike, λ_{\max} and Condensation

5.1. Loss spike experimentally facilitates condensation

In this section, we study the effect of loss spikes on condensation, which may improve the model's generalization in some situations [18, 22]. A condensed network, which refers to a network with neurons condensing in several discrete directions, is equivalent to another smaller network [30, 56]. It has a lower effective complexity than it appears. The embedding principle [17, 41, 51, 53] shows that a condensed network, although equivalent to a smaller one in approximation, has more degeneracy and descent directions that may accelerate the training process. The low effective complexity and simple training process may be underlying reasons for good generalization. We show that the loss spike can facilitate the condensation phenomenon for the noise-free full-batch gradient descent algorithm.

As shown in Fig. 5.1, we train a ReLU with 500 hidden neurons and a Tanh NN with 100 hidden neurons for the one-dimensional fitting problem to fit the data using MSE as the loss function. To clearly study the effect of loss spike on condensation, we take the parameter initialization distribution in the linear regime [30] that does not induce condensation without additional constraints. For NNs with identical initialization, we train the network separately

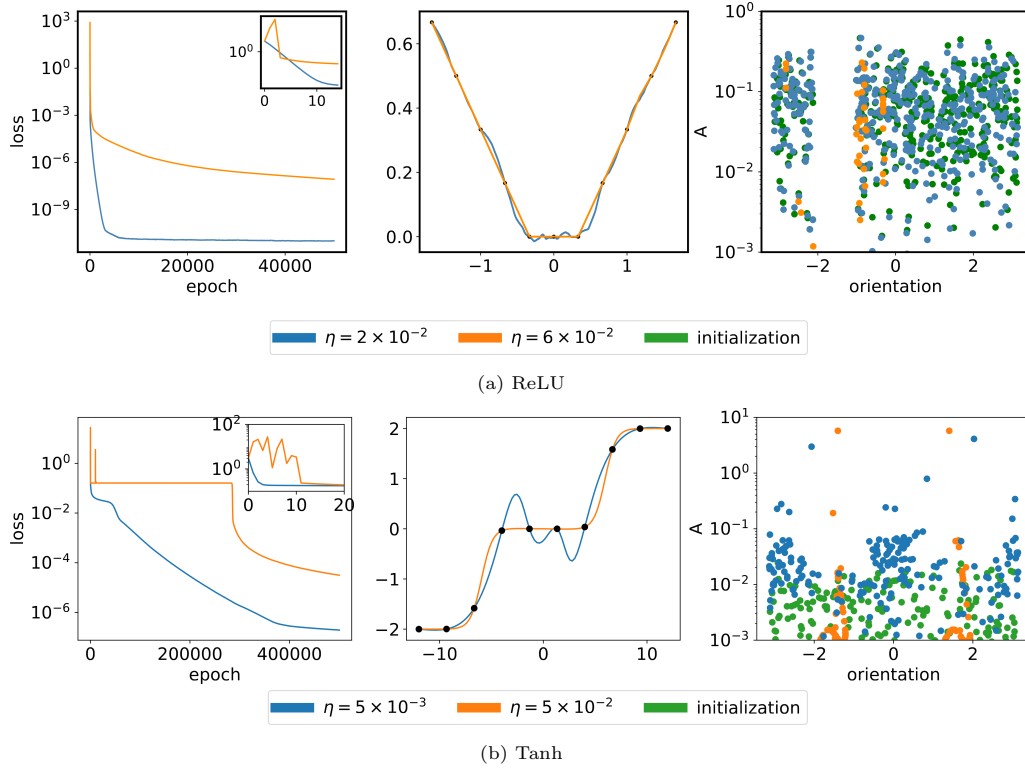


Fig. 5.1. Comparison of two-layer (a) ReLU (b) Tanh NNs with identical initialization but different learning rates η . The loss spike occurs at a large learning rate (orange), while not at a small learning rate (blue). Left: loss vs. epoch. The small picture in the upper right corner shows the occurrence of the loss spike in more detail. Middle: output. Right: the weight feature distribution of the trained models and the initial one.

with a small learning rate (blue) and a large learning rate (orange). For the left subfigure in Fig. 5.1, the loss value has a significant spike for the large learning rate, but not for the small one. At the same time, the middle subfigure reveals that the model output without a loss spike (blue) during the training process has more oscillation than the model output with a loss spike (orange). We study the features of parameters to understand the underlying effect of loss spike better.

To study the parameter features, we measure each parameter pair (a_j, \mathbf{w}_j) by the feature direction $\hat{\mathbf{w}}_j = \mathbf{w}_j / \|\mathbf{w}_j\|_2$ and amplitude¹⁾ $A_j = |a_j| \|\mathbf{w}_j\|_2$. For a NN with one-dimensional input, after incorporating the bias term, \mathbf{w}_j is two-dimensional, and we use the angle between \mathbf{w}_j and the unit vector $(1, 0)$ to indicate the orientation of each neuron. The scatter plots of $\{(\hat{\mathbf{w}}_j, |a_j|)\}_{j=1}^m$ and $\{(\hat{\mathbf{w}}_j, \|\mathbf{w}_j\|_2)\}_{j=1}^m$ of tanh activation are presented in Appendix B to eliminate the impact of the non-homogeneity of tanh activation.

The scatter plots of $\{(\hat{\mathbf{w}}_j, A_j)\}_{j=1}^m$ of the NN is shown in the right subfigure of Fig. 5.1. Parameters without loss spikes (blue) are closer to the initial values (green) than those with loss spikes (orange). For the case with loss spikes, non-zero parameters tend to condense in several discrete orientations, showing a tendency to condensation.

Fig. 5.1 shows that as the learning rate η increases, the occurrence of loss spikes becomes more pronounced. At the same time, the constraints on λ_{\max} are also increasing, resulting in a decrease in the maximum eigenvalue λ_{\max} of the network. It is worth noting that this decrease is accompanied by an increase in the condensation level of network parameters, indicating a clear correlation between the two phenomena.

The variation in learning rates leads to the occurrence or absence of spikes, and we observe that this is accompanied by the model exhibiting either a condensed or non-condensed state. Therefore, it is crucial to investigate the relationship between the two under different learning rates to understand how these dynamics influence the model's training stability and parameter distribution.

5.2. The correlation between λ_{\max} and the condensation

The previous experiments shows that λ_{\max} reaches a lower value during the spike process which might be the potential mechanism by which spikes promote condensation. We further wonder whether there is a correlation between λ_{\max} and the condensation. To quantitatively study this correlation, we define a measure to quantify the degree of condensation between neurons in a neural network.

Definition 5.1. Assume we have a set containing n two-dimensional data points, where each data point is represented as (θ_i, r_i) for $i = 1, 2, \dots, n$. Each data point has two attributes: direction (represented by the angle of the vector) and magnitude (the length of the vector). Each point in our case can be obtained from a neuron parameter (w_i, b_i) in a specific hidden layer of the neural network

$$\theta_i = \arctan \frac{b_i}{w_i}, \quad r_i = \sqrt{w_i^2 + b_i^2}.$$

¹⁾ The amplitude accurately describes the contribution of ReLU neurons due to the homogeneity. For tanh neurons, there is a positive correlation between their amplitude and contribution. Appendix B provides a more refined characterization of tanh network features.

The weighted average direction $\bar{\theta}$, where the weights are the magnitudes r_i of each data point can be defined as

$$\bar{\theta} = \arctan\left(\frac{\sum_{i=1}^n r_i \sin(\theta_i)}{\sum_{i=1}^n r_i}, \frac{\sum_{i=1}^n r_i \cos(\theta_i)}{\sum_{i=1}^n r_i}\right),$$

where $\arctan(y, x)$ is the arctangent function that returns the angle whose tangent is the quotient of its arguments y and x .

Next, using this weighted average direction $\bar{\theta}$, we calculate the variance of the weighted magnitudes.

Definition 5.2. We define the condensed variance is

$$V_{\text{cond}} = \frac{\sum_{i=1}^n r_i (\theta_i - \bar{\theta})^2}{\sum_{i=1}^n r_i}.$$

This definition is based on the condensation phenomenon we observed in experiments: the closer the neuron directions are, i.e. the smaller the variance, the higher the condensation level. At the same time, neurons with larger magnitudes should have greater weights. Based on this, we designed a weighted variance-based condensed variance. This provides the mathematical definitions for the degree of condensation between neurons in a neural network. Based on this, we conducted a series of experiments to investigate the correlation between λ_{max} and condensed variance.

We make some observations using ReLU neural networks with identical settings to those employed in Fig. 5.1. We also conducted experiments (see Appendix B.3) using other activation functions, such as Sigmoid and LeakyReLU, under identical settings.

As shown in Fig. 5.2. For varying learning rate, we observe condensed variance and λ_{max} at the end of training in ReLU NNs. To more intuitively observe the threshold for the occurrence of spikes and condensation, we define the spike threshold. The spike threshold, determined through

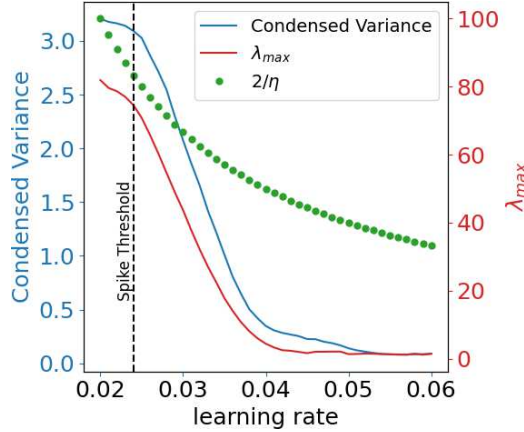


Fig. 5.2. Condensed variance and λ_{max} with varying learning rate. The condensed variance (blue) decreases as the learning rate increases. The λ_{max} (red) value observed at the end of the training epoch exhibits a decreasing trend with respect to increasing learning rates, mirroring the behavior of the condensed variance. $2/\eta$ (green) always remains higher than λ_{max} (red). Learning rates greater than the threshold indicated by the vertical dashed line (black) will result in the observation of loss spikes. The training is conducted using ReLU NNs with the same settings as in Fig. 5.1.

experimental observation, is marked in the figure by the vertical dashed line (black), indicates that when the learning rate exceeds this threshold, a clear observation of loss spikes¹⁾ can be made. Also, the value of λ_{\max} (red) is always lower than $2/\eta$ (green). We have observed similar trends in the changes of condensed variance and λ_{\max} , which inspires us that the phenomenon of “loss spikes promoting condensation” may be related to λ_{\max} . To further investigate the relationship between condensed variance and λ_{\max} , we conducted experiments under different settings.

In our experimental setup, the changes in λ_{\max} and condensed variance occur primarily at the beginning of training and stabilize towards the end. Therefore, we study the trend of their values in the early stages of the training process. We investigate the behavior of condensed variance and λ_{\max} during the initial stages of training using learning rates of 0.03 and 0.05 under the neural network initialization settings of the linear regime [30] (Fig. 5.3, left and right, respectively). Notably, a loss spike was observed when training with the high learning rate of 0.05 (right), but not with the low learning rate of 0.03 (left). Throughout the training process, whether or not loss spikes occurred, λ_{\max} (red) consistently exhibited the same trend as the condensed variance (blue), they both decrease during the same epoch (ignoring the temporary increase in λ_{\max}).

Remark 5.1. We conduct the same experiments using the initialization settings of the condensed regime [30] (see Appendix B.1).

At the same time, we also investigated the trend changes of condensed variance and λ_{\max} under the neural network initialization settings of the condensed regime [30]. We found that, for different initialization settings, both still exhibit similar trends. By fixing the learning rate at 0.01 and taking 10 random seeds, we obtained the average values as shown in Fig. 5.4, where γ is defined in [30] (see Appendix A).

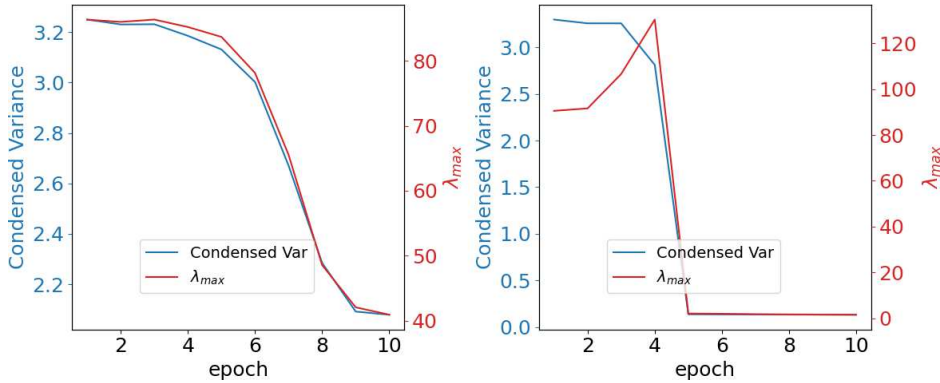


Fig. 5.3. Condensed variance and λ_{\max} at the initial stages of training with learning rate = 0.03 (left) and 0.05 (right). The loss spike occurs at a large learning rate (right) but not a small learning rate (left). Similarly, the spikes in λ_{\max} (red) also follow this pattern, occurring at higher learning rates (right) but not at lower learning rates (left). Throughout the training process, λ_{\max} (red) consistently displays the same trend as the condensed variance (blue), irrespective of the occurrence of spikes. The training is conducted using ReLU NNs with the same settings as in Fig. 5.1.

¹⁾ When the current loss is significantly greater than the loss from the previous epoch, it is considered a loss spike.

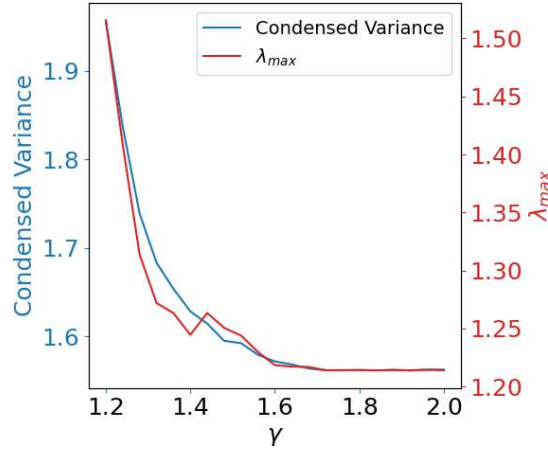


Fig. 5.4. Condensed variance and λ_{\max} with varying γ in condensed regime. The condensed variance (blue) decreases as γ increases. The λ_{\max} (red) value observed at the end of the training epoch exhibits a decreasing trend with respect to increasing γ , mirroring the behavior of the condensed variance. The training is conducted using ReLU NNs with the same settings as in Fig. 5.1.

In our case, $\gamma > 1$ is the condensed regime. As the initialization variance decreases, both the condensed variance and λ_{\max} exhibit a similar decaying trend. In other words, at the end of the training phase, the network becomes more condensed and converges to a flatter loss landscape.

These experiments demonstrate a consistent correlation between condensed variance and λ_{\max} across various settings, strengthening our hypothesis that the loss spike phenomenon, which favors flatter minima (smaller λ_{\max}), may play an important role in promoting weight condensation. The observed relationship between condensed variance and λ_{\max} provides valuable insights for understanding the underlying mechanisms of loss spikes and their impact on the learning dynamics of neural networks. Further theoretical and empirical investigations are needed to unravel the precise nature of this relationship and its implications for generalization performance.

6. Conclusion and Discussion

In this work, we focus on loss spikes observed during neural network training and revisiting the relationship between flatness and generalization. We explain the ascent stage based on the landscape structure, specifically the LLAS structure. For the descent stage, we offer an explanation from the perspective of frequency analysis. We revisit the common understanding of the relationship between flatness and generalization through this frequency analysis. Additionally, we observe in our experiments that noise-free gradient descent with loss spikes can facilitate feature condensation, accompanied by flatter solutions, which may explain good generalization performance in some situations. We observe that there is a certain correlation between λ_{\max} and condensation, which might be the potential mechanism by which spikes promote condensation.

Clearly, many questions remain open. For example, why is the eigendirection corresponding to a large eigenvalue dominated by low-frequency components? Why can loss spikes facilitate feature condensation? We leave the discussion of these important questions for future work.

Appendix A. Experimental Setups

For Figs. 3.1(a-c), 3.2(a), 3.3, we use the two-layer tanh FNN with a width of 20 to fit the target function using full-batch gradient descent as follows:

$$f(x) = \sin(x) + \sin(4x).$$

The initialization of the parameters $\theta \sim N(0, m^{-1})$, where m is the width of the NN, and the learning rate $\eta = 0.05$. For Fig. 3.1(a), the λ_{\max} is calculated every 100 epochs. Figs. 3.1(c) and 3.2(a) show the parameter trajectories of different epoch intervals, which are indicated on the label of the color bar. For Fig. 3.3, the θ_{\max} is selected at epoch 114320, and the θ_{end} is selected at epoch 114400.

For Fig. 3.1(d-f), we use the two-layer ReLU CNN with a Max Pooling layer behind the activation function for the CIFAR10-1k classification problem, i.e. using the first 1000 training data of the CIFAR10 as the training data. The number of the convolution kernels is 16 and the size is 3×3 . We use the MSE as the loss function with learning rate $\eta = 0.1$.

For Fig. 3.1(g-i), we use the VGG-11 [40] model with batch normalization layer for the CIFAR10 classification problem. We conducted experiments with two different settings: a learning rate of 0.2 and another with a learning rate of 0.3. Furthermore, we performed PCA analysis on the spikes from the model trained with a learning rate of 0.3 to gain deeper insights into the spike dynamics.

For Fig. 3.2, we use the following quadratic model as the toy model to illustrate the LLAS structure:

$$f(x, y) = (50x + 200)y^2 - x + 5,$$

where $(x, y) \in (-4, +\infty) \times \mathbb{R}$. The training uses the gradient descent algorithm with learning rate $\eta = 5 \times 10^{-3}$ and the initial value $(x, y) = (0.5, 0.00001)$.

For Figs. 4.1(a, b) and 4.2(a), we use the two-layer tanh FNN with a width of 500 to fit the target function using full-batch gradient descent as follows:

$$f(x) = \tanh(x - 6) + \tanh(x + 6).$$

The initialization of the parameters $\theta \sim N(0, m^{-0.4})$, where m is the width of the NN, and the learning rate $\eta = 0.001$. The training dataset is obtained by sampling 15 points equidistantly in the $[-12, 12]$ interval, and the test dataset is obtained by sampling 14 points equidistantly in the $[-11.14, 11.14]$ interval, which is approximately the midpoint of the pairwise data of the training set.

For Fig. 4.2(b-c), we use the CNNs for the CIFAR10-1k classification problem with structures shown in Tables A.1-A.2, respectively. We use ReLU as the activation function, added behind each convolutional layer. We use the Xavier initialization and the MSE loss function. The learning rate is 0.005. For Fig. 4.2(d), we use the CNNs for the CIFAR10-2k classification problem with structures shown in Table A.3. We use ReLU as the activation function, added behind each convolutional layer. We use the Xavier initialization and the cross-entropy loss function. The learning rate is 0.01. The large batch size we used is 1000, while the small one is 32.

For Figs. 5.1(b), B.1, we use the two-layer tanh FNN with a width of 200 to fit the target function using full-batch gradient descent as follows:

$$f(x) = \tanh(x - 6) + \tanh(x + 6).$$

Table A.1: The architecture of the three-layer CNN used in Fig. 4.2(b).

Layer	Output size
input	$32 \times 32 \times 3$
$3 \times 3 \times 16$, conv	$32 \times 32 \times 16$
2×2 , maxpool	$16 \times 16 \times 16$
flatten	4096
$4096 \rightarrow 10$, linear	10

Table A.2: The architecture of the three-layer CNN used in Fig. 4.2(c).

Layer	Output size
input	$32 \times 32 \times 3$
$3 \times 3 \times 16$, conv	$32 \times 32 \times 16$
2×2 , maxpool	$16 \times 16 \times 16$
$3 \times 3 \times 32$, conv	$16 \times 16 \times 32$
2×2 , maxpool	$8 \times 8 \times 32$
flatten	2048
$2048 \rightarrow 10$, linear	10

Table A.3: The architecture of the five-layer CNN used in Fig. 4.2(d).

Layer	Output size
input	$32 \times 32 \times 3$
$3 \times 3 \times 16$, conv	$32 \times 32 \times 16$
2×2 , maxpool	$16 \times 16 \times 16$
$3 \times 3 \times 32$, conv	$16 \times 16 \times 32$
2×2 , maxpool	$8 \times 8 \times 32$
$3 \times 3 \times 64$, conv	$8 \times 8 \times 64$
2×2 , maxpool	$4 \times 4 \times 64$
flatten	1024
$1024 \rightarrow 500$, linear	500
$500 \rightarrow 10$, linear	10

The initialization of the parameters $\theta \sim N(0, m^{-1})$, where m is the width of the NN. We train the NN with loss spikes using the learning rate $\eta = 0.05$ while using $\eta = 0.005$ for the training without loss spikes. The training dataset is obtained by sampling 10 points equidistantly in the $[-12, 12]$ interval.

For Fig. A.1, we use the two-layer ReLU FNN with a width of 500 to fit the target function using full-batch gradient descent as follows:

$$f(x) = \frac{1}{2}\text{ReLU}\left(-x - \frac{1}{3}\right) + \frac{1}{2}\text{ReLU}\left(x - \frac{1}{3}\right).$$

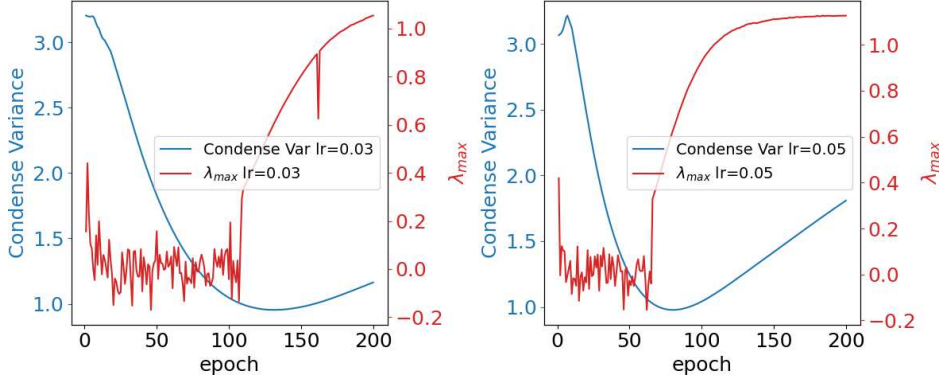


Fig. A.1. Condensed variance and λ_{\max} at the initial stages of training with learning rate = 0.03 (left) and 0.05 (right).

The initialization of the parameters $\theta \sim N(0, m^{-0.8})$, where m is the width of the NN. We train the NN with loss spikes using the learning rate $\eta = 0.03$ and $\eta = 0.05$. The training dataset is obtained by sampling 11 points equidistantly in the $[-5/3, 5/3]$ interval.

For Fig. 5.4, we refer to a special version (with the same initialization for network parameters) of the definition of gamma in [30].

We consider a two-layer NN with m hidden neurons:

$$f_{\theta}(\mathbf{x}) = \sum_{k=1}^m a_k \sigma(\mathbf{w}_k^{\top} \mathbf{x}),$$

where $\mathbf{x} \in \mathbb{R}^d$, $\theta = (\theta_a, \theta_w)$ with $\theta_a = (\{a_k\}_{k=1}^m)$, $\theta_w = (\{\mathbf{w}_k\}_{k=1}^m)$ is the set of parameters initialized by $a_k^0 \sim N(0, \beta_1^2)$, $\mathbf{w}_k^0 \sim N(0, \beta_1^2 \mathbf{I}_d)$.

For the vanilla gradient flow training dynamics of NN, the hyperparameter β_1 is a function of m . To follow the analysis of initialization [30], our initialization parameter is:

$$\gamma = \lim_{m \rightarrow \infty} -\frac{\log \beta_1^2}{\log m}.$$

A.1. Detailed features of LLAS structure

We calculate the Hessian matrix and its eigenvalues as follows:

$$\begin{aligned} \frac{\partial^2 f(x, y)}{\partial x^2} &= 0, & \frac{\partial^2 f(x, y)}{\partial y^2} &= 100x + 400 > 0, \\ \frac{\partial^2 f(x, y)}{\partial xy} &= 100y, & \frac{\partial^2 f(x, y)}{\partial yx} &= 100y, \\ \text{Hessian} &= \begin{bmatrix} \frac{\partial^2 f(x, y)}{\partial x^2} & \frac{\partial^2 f(x, y)}{\partial xy} \\ \frac{\partial^2 f(x, y)}{\partial yx} & \frac{\partial^2 f(x, y)}{\partial y^2} \end{bmatrix} = \begin{bmatrix} 0 & 100y \\ 100y & 100x + 400 \end{bmatrix}. \end{aligned}$$

The eigenvalues of the Hessian matrix when $y = 0$ are $100x + 400 > 0$ and 0.

Appendix B. Experimental Results

B.1. Condensed variance and λ_{\max} in condensed regime

We investigated the behavior of condensed variance and λ_{\max} during the initial stages of training using learning rates of 0.03 and 0.05 under the neural network initialization settings of the condensed regime [30].

However, the results do not display similar trends, but it is reasonable. When neural network parameters are initialized with extremely small values in the condensed regime, λ_{\max} starts off very small and even close to 0. After a period of training, λ_{\max} increases to a non-zero constant. This increase is necessarily accompanied by a rise in λ_{\max} . Meanwhile, the degree of condensation of the neural network increases i.e. the condensed variance decreases since it is under the condensed regime. There is bound to be an inconsistency in the trends here. On the other hand, small initialization implies that λ_{\max} is very small and will not be affected by $2/\eta$, so overall, it may not have a strong correlation with condensation at the beginning of training under the neural network initialization settings of the condensed regime.

B.2. Detailed features of tanh NNs

In order to eliminate the influence of the inhomogeneity of the tanh activation function on the parameter features of Fig. 5.1(b), we plot the normalized scatter figures between $\|a_j\|$, $\|w_j\|$ and the orientation, as shown in Fig. B.1. Obviously, for the network with loss spikes, both the input weight and the output weight have weight condensation, while the network without loss spikes does not have weight condensation.

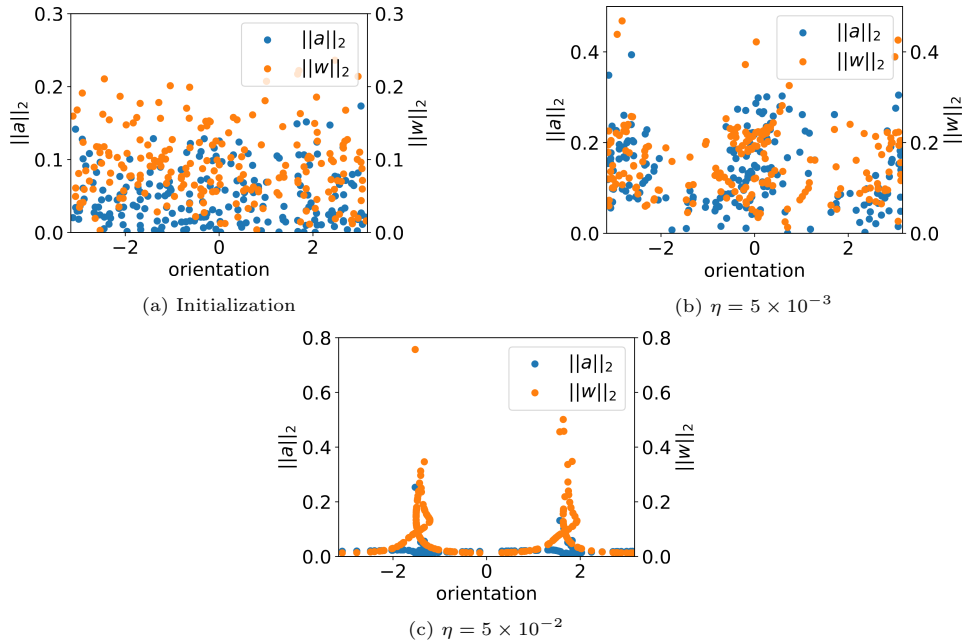


Fig. B.1. The normalized scatter diagrams between $\|a_j\|$, $\|w_j\|$ and the orientation of tanh NNs for the initialization parameters and the parameters trained with and without loss spikes. Blue dots and orange dots are the output weight distribution and the input weight distribution, respectively.

B.3. Experimental results with different activation functions

We observed the phenomena shown in Figs. 5.2 and 5.3 under the experimental setup, conducting experiments with both the Sigmoid activation function and leakyReLU. All other experimental settings were consistent with those in Figs. 5.2 and 5.3. The results obtained are as Figs. B.2 and B.3.

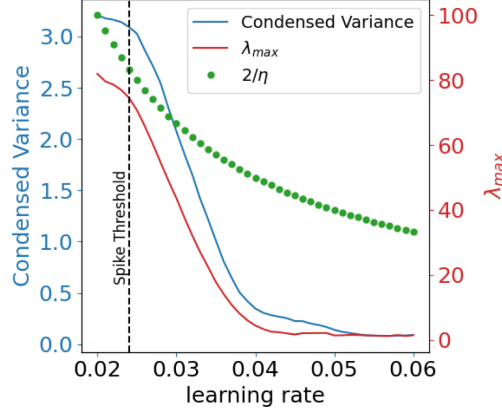


Fig. B.2. Condensed variance and λ_{\max} with varying learning rate. The condensed variance (blue) decreases as the learning rate increases. The λ_{\max} (red) value observed at the end of the training epoch exhibits a decreasing trend with respect to increasing learning rates, mirroring the behavior of the condensed variance. The training is conducted using Sigmoid NNs with the same settings as in Fig. 5.1.

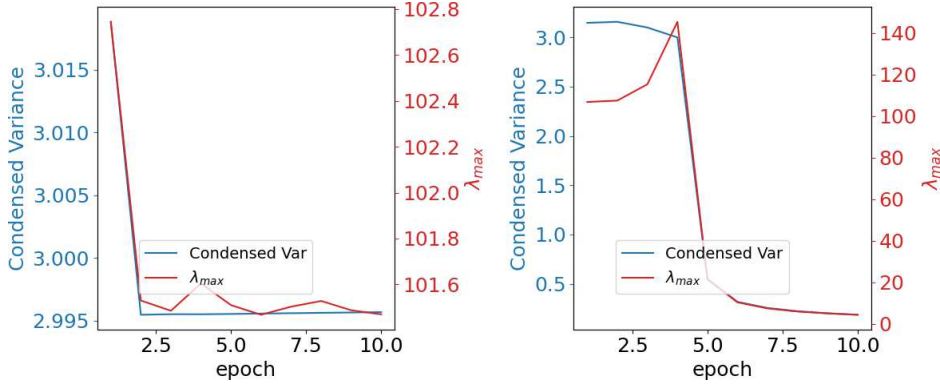


Fig. B.3. Condensed variance and λ_{\max} at the initial stages of training with learning rate = 0.01 (left) and 0.04 (right). Throughout the training process, λ_{\max} (red) consistently displays the same trend as the condensed variance (blue), irrespective of the occurrence of spikes. The training is conducted using LeakyReLU NNs with the same settings as in Fig. 5.1.

Acknowledgments. The authors are listed in alphabetic order. This work is sponsored by the National Key R&D Program of China (Grant No. 2022YFA1008200), by the National Natural Science Foundation of China (Grant Nos. 92270001, 12371511, 12422119), by the Shanghai Municipal of Science and Technology Major Project (Grant No. 2021SHZDZX0102), and by the HPC of School of Mathematical Sciences and the Student Innovation Center, and by the

Siyuan-1 cluster supported by the Center for High Performance Computing at Shanghai Jiao Tong University, Key Laboratory of Marine Intelligent Equipment and System, Ministry of Education, P.R. China. This work was partially supported by the SJTU Kunpeng&Ascend Center of Excellence.

References

- [1] A. Agarwala, F. Pedregosa, and J. Pennington, Second-order regression models exhibit progressive sharpening to the edge of stability, in: *Proceedings of the 40th International Conference on Machine Learning*, PMLR, **202** (2023), 169–195.
- [2] K. Ahn, J.Z. Zhang, and S. Sra, Understanding the unstable convergence of gradient descent, in: *Proceedings of the 39th International Conference on Machine Learning*, PMLR, **162** (2022), 247–257.
- [3] M. Andriushchenko, A. Varre, L. Pillaud-Vivien, and N. Flammarion, SGD with large step sizes learns sparse features, in: *Proceedings of the 40th International Conference on Machine Learning*, PMLR, **202** (2023), 903–925.
- [4] S. Arora, Z.Y. Li, and A. Panigrahi, Understanding gradient descent on the edge of stability in deep learning, in: *Proceedings of the 39th International Conference on Machine Learning*, PMLR, **162** (2022), 948–1024.
- [5] S. Biland, V.C. Azevedo, B. Kim, and B. Solenthaler, Frequency-aware reconstruction of fluid simulations with generative networks, *arXiv:1912.08776*, 2019.
- [6] L. Breiman, Reflections after refereeing papers for NIPS, in: *The Mathematics of Generalization*, CRC Press, (1995), 11–15.
- [7] P. Chaudhari, A. Choromanska, S. Soatto, Y. LeCun, C. Baldassi, C. Borgs, J. Chayes, L. Sagun, and R. Zecchina, Entropy-SGD: Biasing gradient descent into wide valleys, *J. Stat. Mech. Theory Exp.*, **2019**:12 (2019), 124018.
- [8] L. Chen and J. Bruna, On gradient descent convergence beyond the edge of stability, *arXiv:2206.04172*, 2022.
- [9] Z.G. Chen, Y.Q. Li, T. Luo, Z.C. Zhou, and Z.Q.J. Xu, Phase diagram of initial condensation for two-layer neural networks, *arXiv:2303.06561*, 2023.
- [10] J.M. Cohen, S. Kaur, Y.Z. Li, J.Z. Kolter, and A. Talwalkar, Gradient descent on neural networks typically occurs at the edge of stability, *arXiv:2103.00065*, 2021.
- [11] J.K. Cullum and R.A. Willoughby, *Lanczos Algorithms for Large Symmetric Eigenvalue Computations: Vol. I: Theory*, SIAM, 2002.
- [12] A. Damian, E. Nichani, and J.D. Lee, Self-stabilization: The implicit bias of gradient descent at the edge of stability, *arXiv:2209.15594*, 2022.
- [13] L.J. Ding, D. Drusvyatskiy, M. Fazel, and Z. Harchaoui, Flat minima generalize for low-rank matrix recovery, *Inf. Inference J. IMA*, **13**:2 (2024), iaae009.
- [14] L. Dinh, R. Pascanu, S. Bengio, and Y. Bengio, Sharp minima can generalize for deep nets, in: *Proceedings of the 34th International Conference on Machine Learning*, PMLR, **70** (2017), 1019–1028.
- [15] S. Du, J. Lee, H.C. Li, L.W. Wang, and X.Y. Zhai, Gradient descent finds global minima of deep neural networks, in: *Proceedings of the 36th International Conference on Machine Learning*, PMLR, **97** (2019), 1675–1685.
- [16] P. Foret, A. Kleiner, H. Mobahi, and B. Neyshabur, Sharpness-aware minimization for efficiently improving generalization, *arXiv:2010.01412*, 2020.
- [17] K. Fukumizu, S. Yamaguchi, Y. Mototake, and M. Tanaka, Semi-flat minima and saddle points by embedding neural networks to overparameterization, in: *Advances in Neural Information Processing Systems*, Curran Associates, Inc., **32** (2019), 13801–13809.

- [18] F.X. He, T.L. Liu, and D.C. Tao, Control batch size and learning rate to generalize well: Theoretical and empirical evidence, in: *Advances in Neural Information Processing Systems*, Curran Associates, Inc., **32** (2019), 1143–1152.
- [19] S. Hochreiter and J. Schmidhuber, Flat minima, *Neural Comput.*, **9**:1 (1997), 1–42.
- [20] P. Izmailov, A.G. Wilson, D. Podoprikin, D. Vetrov, and T. Garipov, Averaging weights leads to wider optima and better generalization, in: *34th Conference on Uncertainty in Artificial Intelligence 2018*, Curran Associates, Inc., (2018), 876–885.
- [21] A.D. Jagtap, K. Kawaguchi, and G.E. Karniadakis, Adaptive activation functions accelerate convergence in deep and physics-informed neural networks, *J. Comput. Phys.*, **404** (2020), 109136.
- [22] S. Jastrzebski, Z. Kenton, D. Arpit, N. Ballas, A. Fischer, Y. Bengio, and A. Storkey, Three factors influencing minima in SGD, *arXiv:1711.04623*, 2017.
- [23] N.S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P.T.P. Tang, On large-batch training for deep learning: Generalization gap and sharp minima, *arXiv:1609.04836*, 2016.
- [24] A. Lewkowycz, Y. Bahri, E. Dyer, J. Sohl-Dickstein, and G. Gur-Ari, The large learning rate phase of deep learning: The catapult mechanism, *arXiv:2003.02218*, 2020.
- [25] H. Li, Z. Xu, G. Taylor, C. Studer, and T. Goldstein, Visualizing the loss landscape of neural nets, in: *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, Curran Associates Inc., (2018), 6391–6401.
- [26] Y.Z. Li, C. Wei, and T.Y. Ma, Towards explaining the regularization effect of initial large learning rate in training neural networks, in: *Advances in Neural Information Processing Systems*, **32** (2019), 11642–11653.
- [27] T. Lin, S.U. Stich, K.K. Patel, and M. Jaggi, Don’t use large mini-batches, use local SGD, *arXiv:1808.07217*, 2018.
- [28] Z.Q. Liu, W. Cai, and Z.Q.J. Xu, Multi-scale deep neural network (MscaledNN) for solving Poisson-Boltzmann equation in complex domains, *Commun. Comput. Phys.*, **28**:5 (2020), 1970–2001.
- [29] T. Luo, Z. Ma, Z.Q.J. Xu, and Y.Y. Zhang, Theory of the frequency principle for general deep neural networks, *CSIAM Trans. Appl. Math.*, **2**:3 (2021), 484–507.
- [30] T. Luo, Z.Q.J. Xu, Z. Ma, and Y.Y. Zhang, Phase diagram for two-layer ReLU neural networks at infinite-width limit, *J. Mach. Learn. Res.*, **22**:71 (2021), 1–47.
- [31] K.F. Lyu, Z.Y. Li, and S. Arora, Understanding the generalization benefit of normalization layers: Sharpness reduction, *arXiv:2206.07085*, 2022.
- [32] C. Ma, D. Kunin, L. Wu, and L.X. Ying, Beyond the quadratic approximation: The multiscale structure of neural network loss landscapes, *J. Mach. Learn.*, **1**:3 (2022), 247–267.
- [33] C. Ma, L. Wu, and W.N. E, A qualitative study of the dynamic behavior for adaptive gradient algorithms, in: *Proceedings of the 2nd Mathematical and Scientific Machine Learning Conference*, PMLR, **145** (2022), 671–692.
- [34] C. Ma and L.X. Ying, On linear stability of SGD and input-smoothness of neural networks, in: *Advances in Neural Information Processing Systems*, **34** (2021), 16805–16817.
- [35] H. Maennel, O. Bousquet, and S. Gelly, Gradient descent quantizes ReLU network features, *arXiv:1803.08367*, 2018.
- [36] F. Pellegrini and G. Biroli, An analytic theory of shallow networks dynamics for hinge loss classification, in: *Advances in Neural Information Processing Systems*, Curran Associates, Inc., **33** (2020), 5356–5367.
- [37] N. Rahaman, A. Baratin, D. Arpit, F. Draxler, M. Lin, F.A. Hamprecht, Y. Bengio, and A. Courville, On the spectral bias of neural networks, in: *Proceedings of the 36th International Conference on Machine Learning*, PMLR, **97** (2019), 5301–5310.
- [38] Y.N. Ren, C. Ma, and L.X. Ying, Understanding the generalization benefits of late learning rate decay, in: *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, PMLR, **238** (2024), 4465–4473.
- [39] B. Ronen, D. Jacobs, Y. Kasten, and S. Kritchman, The convergence rate of neural networks for

- learned functions of different frequencies, in: *Advances in Neural Information Processing Systems*, Curran Associates, Inc., **32** (2019), 4761–4771.
- [40] K. Simonyan and A. Zisserman, Very deep convolutional networks for large-scale image recognition, *arXiv:1409.1556*, 2014.
 - [41] B. Simsek, F. Ged, A. Jacot, F. Spadaro, C. Hongler, W. Gerstner, and J. Brea, Geometry of the loss landscape in overparameterized neural networks: Symmetries and invariances, in: *International Conference on Machine Learning*, PMLR, (2021), 9722–9732.
 - [42] Z.X. Wang, Z.Z. Li, and J. Li, Analyzing sharpness along gd trajectory: Progressive sharpening and edge of stability, in: *Advances in Neural Information Processing Systems*, Curran Associates, Inc., **35** (2022), 9983–9994.
 - [43] L. Wu, C. Ma, and W.N. E, How SGD selects the global minima in over-parameterized learning: A dynamical stability perspective, in: *Advances in Neural Information Processing Systems*, Curran Associates, Inc., **31** (2018), 8279–8288.
 - [44] L. Wu, Z. Zhu, W. E, Towards understanding generalization of deep learning: Perspective of loss landscapes, *arXiv:1706.10239*, 2017.
 - [45] C. Xing, D. Arpit, C. Tsirigotis, and Y. Bengio, A walk with SGD, *arXiv:1802.08770*, 2018.
 - [46] Z.Q.J. Xu, Y.Y. Zhang, and T. Luo, Overview frequency principle/spectral bias in deep learning, *arXiv:2201.07395*, 2022.
 - [47] Z.Q.J. Xu, Y.Y. Zhang, T. Luo, Y.T. Xiao, and Z. Ma, Frequency principle: Fourier analysis sheds light on deep neural networks, *Commun. Comput. Phys.*, **28**:5 (2020), 1746–1767.
 - [48] Z.Q.J. Xu, Y.Y. Zhang, and Y.Y. Xiao, Training behavior of deep neural network in frequency domain, in: *International Conference on Neural Information Processing*, Springer, (2019), 264–274.
 - [49] Z.Q.J. Xu and H.X. Zhou, Deep frequency principle towards understanding why deeper learning is faster, in: *Proceedings of the AAAI Conference on Artificial Intelligence*, AAAI Press, **35** (2021), 10541–10550.
 - [50] C.Y. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, Understanding deep learning (still) requires rethinking generalization, *Commun. ACM*, **64**:3 (2021), 107–115.
 - [51] Y.Y. Zhang, Y.Q. Li, Z.W. Zhang, T. Luo, and Z.Q.J. Xu, Embedding principle: A hierarchical structure of loss landscape of deep neural networks, *J. Mach. Learn.*, **1** (2022), 60–113.
 - [52] Y.Y. Zhang, T. Luo, Z. Ma, and Z.Q.J. Xu, A linear frequency principle model to understand the absence of overfitting in neural networks, *Chin. Phys. Lett.*, **38**:3 (2021), 038701.
 - [53] Y.Y. Zhang, Z.W. Zhang, T. Luo, and Z.Q.J. Xu, Embedding principle of loss landscape of deep neural networks, in: *Advances in Neural Information Processing Systems*, Curran Associates, Inc., **34** (2021), 14848–14859.
 - [54] Y.W. Zheng, R.C. Zhang, and Y.Y. Mao, Regularizing neural networks via adversarial model perturbation, In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, (2021), 8156–8165.
 - [55] H.X. Zhou, Q.X. Zhou, Z.Y. Jin, T. Luo, Y.Y. Zhang, and Z.Q.J. Xu, Empirical phase diagram for three-layer neural networks with infinite width, in: *Advances in Neural Information Processing Systems*, Curran Associates, Inc., **35** (2022), 26021–26033.
 - [56] H.X. Zhou, Q.X. Zhou, T. Luo, Y.Y. Zhang, and Z.Q.J. Xu, Towards understanding the condensation of neural networks at initial training, *arXiv:2105.11686*, 2021.
 - [57] X.Y. Zhu, Z.X. Wang, X. Wang, M. Zhou, and R. Ge, Understanding edge-of-stability training dynamics with a minimalist example, *arXiv:2210.03294*, 2022.