DEEP DOUBLE DESCENT VIA SMOOTH INTERPOLATION

Matteo Gamba*
KTH Royal Institute of Technology
Sweden
mgamba@kth.se

Erik Englesson
KTH Royal Institute of Technology
Sweden
engless@kth.se

Mårten Björkman
KTH Royal Institute of Technology
Sweden
celle@kth.se

Hossein Azizpour
KTH Royal Institute of Technology
Sweden
azizpour@kth.se

ABSTRACT

Overparameterized deep networks are known to be able to perfectly fit the training data while at the same time showing good generalization performance. A common paradigm drawn from intuition on linear regression suggests that large networks are able to interpolate even noisy data, without considerably deviating from the ground-truth signal. At present, a precise characterization of this phenomenon is missing. In this work, we present an empirical study of sharpness of the loss landscape of deep networks as we systematically control the number of model parameters and training epochs. We extend our study to neighbourhoods of the training data, as well as around cleanly- and noisily-labelled samples. Our findings show that the loss sharpness in the input space follows both model- and epoch-wise double descent, with worse peaks observed around noisy labels. While small interpolating models sharply fit both clean and noisy data, large models express a smooth and flat loss landscape, in contrast with existing intuition.

1 Introduction

Recent years have seen increased interest in the study of smoothness of deep networks in relationship to their generalization ability. For networks interpreted as functions of their parameters, smoothness of the loss landscape has been related to improved generalization (Ma & Ying, 2021; Foret et al., 2020; Rosca et al., 2020), increased stability to perturbations (Keskar et al., 2017), reduced minimum description length (Hochreiter & Schmidhuber, 1997), as well as better compression (Chang et al., 2021). Additionally, input-space sensitivity measures of the network's learned function have been connected to generalization performance (LeJeune et al., 2019; Novak et al., 2018). Indeed mounting evidence, both empirical (Gamba et al., 2022; Novak et al., 2018) as well as theoretical (Bubeck & Sellke, 2021; Neyshabur et al., 2018), suggests that large state-of-the-art models achieve robust generalization (Ma & Ying, 2021) via smoothness of the learned function. While overparameterization alone is not enough to guarantee adversarial robustness (Chen et al., 2021; Rice et al., 2020), the large number of parameters of modern networks is thought to promote some form of implicit regularization (Gamba et al., 2022; Bubeck & Sellke, 2021; Neyshabur et al., 2018, 2015).

The ability of neural networks to interpolate training data is related to the double descent phenomenon (Belkin et al., 2019; Nakkiran et al., 2019; Belkin et al., 2018) – most markedly in the presence of noisy labels – with large overparameterized networks showing good generalization performance well past the *interpolation threshold* (Belkin et al., 2019), namely the model size achieving zero training error ($0/1$ loss).

In this work, we study the deep double descent phenomenon (Nakkiran et al., 2019) through the lens of smooth interpolation of the training data, as the model size as well as the number of training epochs vary.

*Corresponding author.

Current intuition from linear regression suggests that, under some hypothesis on the training sample, large overparameterized models are able to perfectly interpolate both cleanly- and noisily-labeled samples, without considerably deviating from the ground-truth signal, hence locally fitting noisy data sharply (Muthukumar et al., 2020; Bartlett et al., 2020). To quantify this notion for deep networks trained in practice, we conduct an empirical exploration of smoothness of the loss landscape w.r.t. the input variable, in relationship to the double descent curve of the test error. We provide explicit measures of smoothness of the loss, and study deep networks trained on image classification.

Due to the inherently noisy nature of Euclidean estimators in pixel space, and following the *manifold hypothesis* Pope et al. (2020); Bengio (2013); Narayanan & Mitter (2010), postulating that natural data lies on a combination of manifolds of lower dimension than the input data's ambient dimension, we constrain our measures to the support of the data distribution, locally to each training point.

Contributions

- We present the first systematic empirical study of smoothness of the loss landscape of deep networks in relation to overparameterization for natural image datasets.
- Starting from infinitesimal smoothness measures from prior work, we introduce volumetric measures that capture loss smoothness when traveling away from training points.
- We develop a method for constraining our measures to a local approximation of the data manifold, in proximity of each training point.
- We present an empirical study of model-wise and epoch-wise double descent for neural networks trained without confounders (explicit regularization, data augmentation, batch normalization), as well as for commonly-found training settings.

2 Related work

Our work presents an empirical study of deep double descent in relationship to smoothness of the loss landscape, with respect to the input variable. Our methodology builds upon input-space sensitivity analyses for neural networks, presenting a first systematic study of the role of overparameterization in promoting smoothness of the network's learned function. The smoothness measures presented in section 3, are inspired by the vast body of work on the loss landscape of neural networks in parameter space. Due to the extensive theoretical literature on double descent in simplified controlled settings such as linear regression (Muthukumar et al., 2020; Bartlett et al., 2020; Belkin et al., 2018), in the following we mainly draw connections to prior work targeting deep networks.

Deep Double Descent The classical bias-variance interpretation states that small low-complexity models are characterized by high bias and low variance, while increasing model size reduces bias at the cost of increased variance. After a threshold of optimal bias-variance tradeoff, model variance grows too large, causing the test performance to degrade, showing the characteristic U-shaped test error curve Geman et al. (1992). However, by increasing model size even further, a second descent in the test error is observed (Belkin et al., 2019). This phenomenon was first observed for several machine learning algorithms with increasing model size (model-wise), but Nakkiran et al. (2019) showed a similar trend during training of deep networks (epoch-wise), as well as for increasing dataset size (sample-wise).

Double descent has been studied from various perspectives: bias-variance decomposition (Yang et al., 2020; Neal et al., 2018), parameter norms (Belkin et al., 2019), and decision boundaries (Somepalli et al., 2022). In this work, we study model-wise and epoch-wise double descent in terms of smoothness of the loss landscape with respect to the input, and separate the analysis in terms of clean and noisily-labeled data points. The most related work to ours is the concurrent one of Somepalli et al. (2022) that studies decision boundaries in terms of reproducibility and double descent. Our works differ in that we study double descent in the loss landscape. Furthermore, our study is more detailed in terms of investigating both model-wise and epoch-wise trends, and takes a closer look at the impact of clean and noisily-labeled data points. Finally, we study the phenomenon without explicit regularization (batch norm, data augmentation) and a simpler optimization procedure (SGD with constant learning rate instead of Adam) to reduce confounding factors.

Loss Landscape of Neural Networks To understand the remarkable generalization ability of deep networks (Xie et al., 2020; Geiger et al., 2019; Keskar et al., 2017), as well as to design better training criteria (Foret et al., 2020), several works study the loss landscape of deep networks in *parameter space*, focusing on solutions obtained by SGD (Kuditipudi et al., 2019), as well as the optimization process (Arora et al., 2022; Li et al., 2021). Inspired by such literature, we quantify smoothness of the loss landscape by estimating the *sharpness* of the loss, as proposed by Foret et al. (2020) and Keskar et al. (2017) for the parameter-space, but we perform our analysis in *input-space*. Importantly, in this work we focus on image classification tasks, and study smoothness of interpolation of training data points.

Input Space Sensitivity and Smoothness Novak et al. (2018) present an empirical sensitivity study of fully-connected networks with piece-wise linear activation functions through the input-output Jacobian norm, which is shown to strongly correlate to the generalization ability of the networks considered. Their study proposes an infinitesimal analysis of the Jacobian norm at training and validation points, as well as the use of input-space trajectories in proximity of the data manifold to probe trained networks. LeJeune et al. (2019) analyse second-order information (the tangent Hessian of a neural network) by using weak data augmentation to constrain their measure to the proximity of the data manifold. Lastly, Gamba et al. (2022) introduce a nonlinearity measure for neural networks with piece-wise linear activations, that strongly correlates with the test error in the second descent for large overparameterized models. Similar to the first two works, we study smoothness of neural networks, using the Jacobian and Hessian norm of neural networks trained in practice, and similar to the latter work, we provide a systematic study of model-wise double descent, which we further extend to epoch-wise trends. Importantly, our study focuses on the loss landscape and average-case robustness, rather than the network's learned function.

Finally, Rosca et al. (2020) investigate how to encourage smoothness, either on the entire input space or around training data points. Interestingly, they postulate a connection between model-wise double descent and smoothness: during the first ascent the model size is large enough to fit the training data at the expense of smoothness, while the second descent happens as the model size becomes large enough for smoothness to increase. Later, Bubeck & Sellke (2021) theoretically prove a universal law of robustness highlighting a trade-off between the model size (number of parameters) and smoothness (Lipschitz constant) of a learning algorithm w.r.t. its input variable. Our work provides empirical evidence supporting the postulate of Rosca et al. (2020) and the law of robustness of Bubeck & Sellke (2021).

3 Methodology

Our leading research question is to quantify smoothness of interpolation of training data for deep networks trained on classification tasks, as the number of model parameters is increased. We interpret a network as a function with input variable $\mathbf{x} \in \mathbb{R}^d$ and learnable parameter $\boldsymbol{\theta}$, incorporating all weights and biases. Our study focuses on the landscape of the loss $\mathcal{L}_{\boldsymbol{\theta}}(\mathbf{x}, y) := \mathcal{L}(\boldsymbol{\theta}, \mathbf{x}, y)$ treated as a function of the input $\mathbf{x}$, with target y. Inspired by the literature on the loss landscape of neural networks in parameter space (Foret et al., 2020; Keskar et al., 2017), we quantify (the lack of) smoothness by devising explicit measures of loss sharpness in a neighbourhood of training points $(\mathbf{x}_n, y_n)$, for $n = 1, \ldots, N$. Crucially, for any given network, we focus on sharpness w.r.t. the input variable $\mathbf{x}$, keeping the parameter $\boldsymbol{\theta}$ fixed.

We begin by describing infinitesimal sharpness in section 3.1, which we compute in proximity of the data manifold local to each training point in section 3.2. Finally, we introduce a method for estimating sharpness over data-driven volumes by exploiting data augmentation in section 3.3, and in section 3.4 we detail the chosen data augmentation strategies. The proposed methodology enables us to measure sharpness of interpolation of the training data, by focusing our study on the support of the data distribution.

3.1 Sharpness at Data Points

To estimate how sharply the loss changes w.r.t. infinitesimal perturbations of the input variable $\mathbf{x}$, we study the Jacobian of the loss,

$$\mathbf{J}(\mathbf{x}, y) := \frac{\partial}{\partial \mathbf{x}} \mathcal{L}_{\boldsymbol{\theta}}(\mathbf{x}, y) \tag{1}$$

To measure sharpness at a point $(\mathbf{x}_n, y_n)$, we follow Novak et al. (2018), and compute the Frobenius norm of $\mathbf{J}(\mathbf{x}_n, y_n)$, which we take in expectation over the training set $\mathcal{D} = \{(\mathbf{x}_n, y_n)\}_{n=1}^N$,

$$J = \mathbb{E}_{\mathcal{D}} \|\mathbf{J}(\mathbf{x}, y)\|_F \tag{2}$$

assuming that the loss is differentiable one time at the points considered. Intuitively, sharpness is measured by how fast the loss $\mathcal{L}_{\boldsymbol{\theta}}(\mathbf{x}, y)$ changes in infinitesimal neighbourhoods of the training data, and a network is said to smoothly interpolate a data point $\mathbf{x}_n$ if the loss is approximately flat locally around the point and the point is classified correctly according to the corresponding target y_n. Throughout our experiments, the Jacobian $\mathbf{J}$ is computed using a backward pass w.r.t. the input variable $\mathbf{x}$.

Equation 2 provides first-order information about the loss landscape. To gain knowledge about curvature, we also study the Hessian of the loss w.r.t. the input variable,

$$\mathbf{H}(\mathbf{x}, y) := \frac{\partial^2}{\partial \mathbf{x} \partial \mathbf{x}^T} \mathcal{L}_{\boldsymbol{\theta}}(\mathbf{x}, y) \tag{3}$$

whose Frobenius norm again we take in expectation over the training set

$$H = \mathbb{E}_{\mathcal{D}} \|\mathbf{H}(\mathbf{x}, y)\|_F \tag{4}$$

The Hessian tensor in equation 3 depends quadratically on the input space dimensionality d, providing a noisy Euclidean estimator of loss curvature in proximity of the input data. Following the *manifold hypothesis* (Bengio, 2013; Narayanan & Mitter, 2010), stating that natural data lies on subspaces of dimensionality lower than the ambient dimension d, we restrict Hessian computation to the tangent subspace of each training point $\mathbf{x}_n$. Starting from equation 1, throughout our experiments, equation 3 is estimated by computing the tangent Hessian, as outlined in the next section.

3.2 Tangent Hessian Estimation

To constrain equation 3 to the support of the data distribution and reduce computational complexity, we adapt the method proposed by LeJeune et al. (2019) and estimate the Hessian norm of the loss projected onto the data manifold local to each training point.

For any input data point $(\mathbf{x}_n, y_n)$ and corresponding Jacobian $\mathbf{J}(\mathbf{x}_n, y_n)$, we generate M augmented data points $\mathbf{x}_n + \mathbf{u}_m$ by randomly sampling a displacement vector $\mathbf{u}_m$ using weak data augmentation. For each sampled $\mathbf{u}_m$, we then estimate the Hessian $\mathbf{H}(\mathbf{x}_n, y_n)$ projected along the direction $\mathbf{x}_n + \mathbf{u}_m$, by computing the finite difference $\frac{1}{\delta}\mathbf{J}(\mathbf{x}_n, y_n) - \mathbf{J}(\mathbf{x}_n + \delta\mathbf{u}_m, y_n)$. Then, following Donoho & Grimes (2003) we estimate the Hessian norm directly by computing

$$H = \frac{1}{M^2\delta^2}\mathbb{E}_{\mathcal{D}}\Big(\sum_{m=1}^{M} \|\mathbf{J}(\mathbf{x}_n, y_n) - \mathbf{J}(\mathbf{x}_n + \delta\mathbf{u}_m, y_n)\|_F^2\Big)^{\frac{1}{2}} \tag{5}$$

which is equivalent to a rescaled version of the rugosity measure of LeJeune et al. (2019). Importantly, different from rugosity, augmentations $\mathbf{x}_n + \mathbf{u}_m$ are generated using weak colour transformations in place of affine transformations (1-pixel shifts), since weak photometric transformations are guaranteed to be fully on-manifold. Details about the specific colour transformations are presented in appendix C.

3.3 Sharpness over Data-Driven Volumes

The measures introduced in equations 2 and 5, capture local sharpness over infinitesimal neighbourhoods of input data points. To study how different networks fit the training data, we devise a method for estimating loss sharpness over volumes centered at each training point $\mathbf{x}_n$, as one moves away from the point. Essentially, we exploit a variant of Monte Carlo (MC) integration to capture sharpness over data-driven volumes, by applying two steps. First, we integrate the Jacobian and Hessian norms along geodesic paths $\boldsymbol{\pi}_p \subset \mathbb{R}^d$ based at $\mathbf{x}_n$, on the data manifold local to each training point, for $p = 1, \ldots, P$. Second, we compute a MC estimation of sharpness over the volume covered by the P paths. The following details each step.

Sharpness along geodesic paths For each training point $(\mathbf{x}_n, y_n) \in \mathcal{D}$, we aim to estimate loss sharpness as we move away from $\mathbf{x}_n$, while traveling on the support of the data distribution. To do so, we exploit a sequence of weak data augmentations of increasing strength to generate paths $\boldsymbol{\pi}_p \subset \mathbb{R}^d$ in the input space, formed by connecting augmentations of $\mathbf{x}_n$.

Formally, let $\mathcal{T}_{\mathbf{s}} : \mathbb{R}^d \to \mathbb{R}^d$, represent a family of smooth transformations (data augmentation) acting on the input space and governed by parameter $\mathbf{s}$, controlling the strength $S = \|\mathbf{s}\|_F$ as well as the direction of the augmentation in $\mathbb{R}^d$. In general, the parameter $\mathbf{s}$, interpreted as a suitably distributed random variable, models the randomness of the transformation. Randomly sampling $\mathbf{s}$, yields a value $\mathbf{s}^{p,k}$ corresponding to a fixed transformation $\mathcal{T}_{\mathbf{s}^{p,k}}$ of strength S^k. For instance, for affine translations, $\mathbf{s}^{p,k}$ models a random radial direction sampled from a hypersphere centered at $\mathbf{x}_n$, with strength S^k denoting the magnitude of the translation (e.g. 4-pixel shift). For photometric transformations, $\mathbf{s}^{p,k}$ may model the change in brightness, contrast, hue, and saturation, with total strength S^k.

To generate on-manifold paths $\boldsymbol{\pi}_p$ starting from $\mathbf{x}_n$, we proceed as follows. First, we fix a sequence of $K+1$ strengths $S^0 < S^1 < \ldots < S^K$, with $S^0 = 0$ denoting the identity transformation $\forall\, p$. Then, for each strength S^k, with $k \geq 1$, p random directions $\mathbf{s}^{p,k}$ are sampled, each with respective fixed magnitude $\|\mathbf{s}^{p,k}\|_F = S^k$. This yields P sequences of transformations $\{\mathcal{T}_{\mathbf{s}^{p,k}}\}_{k=0}^{K}$, each producing augmented versions $\mathbf{x}_n^{p,k}$ of $\mathbf{x}_n$, ordered by strength, $\mathbf{x}_n^{p,1} \prec \ldots \prec \mathbf{x}_n^{p,K}$, and forming a path $\boldsymbol{\pi}_p \subset \mathbb{R}^d$. Specifically, each path $\boldsymbol{\pi}_p$ approximates an on-manifold trajectory by a sequence of Euclidean segments $\mathbf{x}_n^{p,k+1}\mathbf{x}_n^{p,k}$, for $k = 0, \ldots, K$. The maximum augmentation strength S^K controls the distance traveled from $\mathbf{x}_n$, while the number K of strengths used controls how fine-grained the Euclidean approximation is.

Volume integration Once a sequence of paths $\{\boldsymbol{\pi}_p\}_{p=1}^{P}$ is generated for $\mathbf{x}_n$, volume-based sharpness is computed by integrating over all P paths, and normalizing the measure by the length $\mathrm{len}(\boldsymbol{\pi}_p)$ of each path:

$$\frac{1}{P}\sum_{p=1}^{P}\frac{1}{\mathrm{len}(\boldsymbol{\pi}_p)}\int_{\boldsymbol{\pi}_p}\boldsymbol{\sigma}(\mathbf{x},y_n)d\mathbf{x} \tag{6}$$

where $\boldsymbol{\sigma}$ represents an infinitesimal sharpness measure, namely the Jacobian and tangent Hessian norms at $(\mathbf{x}_n, y_n)$.

Finally, volume-based sharpness is obtained by averaging over the training set $\mathcal{D}$

$$\frac{1}{P}\mathbb{E}_{\mathcal{D}}\sum_{p=1}^{P}\frac{1}{\mathrm{len}(\boldsymbol{\pi}_p)}\int_{\boldsymbol{\pi}_p}\boldsymbol{\sigma}(\mathbf{x},y_n)d\mathbf{x}=\frac{1}{NP}\sum_{n=1}^{N}\sum_{p=1}^{P}\frac{1}{\mathrm{len}(\boldsymbol{\pi}_p)}\int_{\boldsymbol{\pi}_p}\boldsymbol{\sigma}(\mathbf{x},y_n)d\mathbf{x} \tag{7}$$

Importantly, extending LeJeune et al. (2019), we replace Euclidean integration by geodesic integration over a local approximation of the data manifold, by generating augmentations of increasing strength.

Crucially, the proposed MC integration captures average-case sharpness in proximity of the training data and is directly related to the generalization ability of the studied networks, as opposed to worst-case sensitivity, as typically considered in adversarial settings (Moosavi-Dezfooli et al., 2019). In fact, the random sampling performed in equation 7 is unlikely to hit adversarial directions, which are commonly identified by searching the input space through an optimization process (Goodfellow et al., 2014; Szegedy et al., 2013).

To conclude our methodology, in section 3.4 we present the family of transformations $\mathcal{T}_{\mathbf{s}}$ used for generating trajectories $\boldsymbol{\pi}_p$ throughout our experiments.

3.4 Weak Data Augmentation Strategies

Computing sharpness of interpolation via equation 6 for each data point $\mathbf{x}_n$ requires generating P trajectories $\boldsymbol{\pi}_p$ composed of augmentations of $\mathbf{x}_n$ of controlled increasing strength. Furthermore, the augmented data points $\{\mathbf{x}_n^{p,k}\}_{k=0}^{K}$ should lie in proximity of the base point $\mathbf{x}_n$ in order for the Euclidean approximation to be meaningful. Finally, to correctly estimate correlation between smoothness and the generalization ability of the networks considered, volume-based sharpness should not rely on validation data points, i.e. the augmentations $\mathbf{x}_n^{p,k}$ should be strongly correlated to $\mathbf{x}_n$, for each p, k.

To satisfy the above, we modify a weak data augmentation algorithm introduced by Yu et al. (2018), which allows to efficiently generate augmentations that lie in close proximity to the base training point $\mathbf{x}_n$, for image data. Specifically, each base image $\mathbf{x}_n$, consisting of C input channels (e.g. $C = 3$ for RGB images) and $h \times w$ spatial dimensions, is interpreted as C independent matrices $\mathbf{x}_n[c,:,:\,] \in \mathbb{R}^{h\times w}$, each factorized using Singular Value Decomposition (SVD), yielding a decomposition $\mathbf{x}_n[c,:,:\,] = U^c\Sigma^c V^{cT}$, where Σ^c is a diagonal matrix whose entries are the singular values of $\mathbf{x}_n[c,:,:\,]$ sorted by decreasing magnitude. In the original method, Yu et al. (2018) produce weak augmentations by randomly erasing one singular value from the smallest ones, thereby obtaining a modified matrix $\tilde{\Sigma}^c$, and then reconstructing each channel of the base sample via $U^c\tilde{\Sigma}^c V^{cT}$. In this work, in order to generate P random augmentations of strength k, $\tilde{\Sigma}^c$ is obtained by erasing k singular values $\Sigma^c_{i,i}$, for $i = w-k-p+1,\ldots,w-p$, and $p = 0,\ldots,P-1$[2]. Essentially, the augmentation strength is given by the number k of singular values erased, and P augmentations of similar strength are generated by erasing P subsets of size k from the smallest singular values, for each channel c.

We note that this method produces augmented images that are highly correlated with the corresponding base training sample, and as such they do not directly amount to producing validation data points. We refer the reader to appendix D for further details.

4 Experiments

In this section, we present our empirical exploration of input-space sharpness of the loss landscape of deep networks as the model size and number of training epochs vary. Focusing on *implicit regularization* (Neyshabur et al., 2015) promoted by learning algorithm and model architecture, we evaluate our sharpness measures on a series of networks of increasing number of parameters, trained without any form of explicit regularization (e.g. weight decay, batch normalization, dropout).

[2]Assuming square spatial dimensions $h = w$.

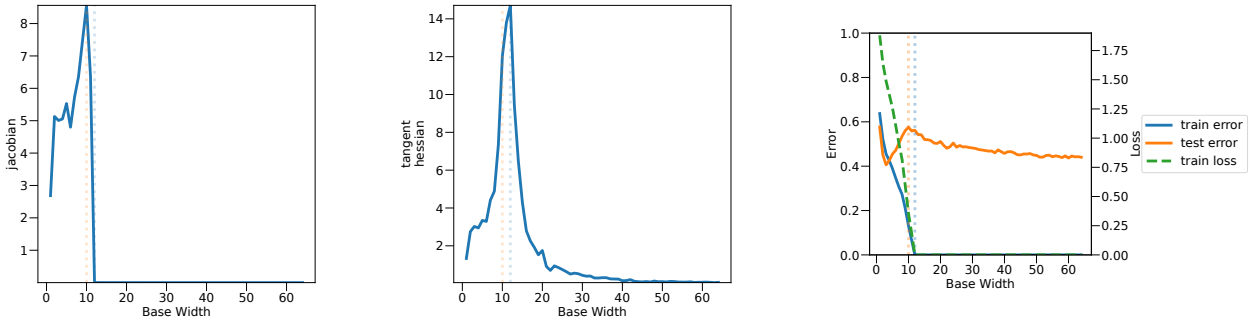


Figure 1: (Left) average Jacobian norm over the CIFAR-10 training set for a family of ConvNets of increasing base width k. (Middle) Average tangent Hessian norm on the training set, for the same ConvNets. (Right) Train and test error (0/1 loss) and training loss (crossentropy) against model size. The dotted vertical lines mark the model width that achieves zero test error (orange) and zero train error (*interpolation threshold*, blue) . All models are trained for 4000 epochs. We observe how loss sharpness increases with model size until the peak in test error is reached, beyond which the Jacobian norm rapidly decreases approaching zero. Similarly, the tangent Hessian norm monotonically increases until the peak in test error, and then decreases at a lower rate, indicating that large models parameterize a flat loss landscape in infinitesimal neighborhoods of the training data.

Experimental setup We reproduce deep double descent by following the experimental setup of Nakkiran et al. (2019). Specifically, we train a family of ConvNets formed by 4 convolutional stages of controlled base width $[k, 2k, 4k, 8k]$, for $k = 1, \ldots, 64$, on the CIFAR-10 dataset with 20% noisy training labels. All models are trained for 4000 epochs using SGD with momentum 0.9 and fixed learning rate. Following Arpit et al. (2019), to stabilize prolonged training, we use a learning rate warmup schedule. Furthermore, we extend our empirical results to training settings more commonly found in practice, and validate our main findings on a series of ResNet18s (He et al., 2015) of increasing base width $k = 1, \ldots, 64$, with batch normalization, trained with the Adam optimizer for 4000 epochs using data augmentation. We refer the reader to section B for a full description of our experimental setting.

4.1 Loss smoothness follows double descent

Figure 1 (right) reproduces deep double descent for ConvNets trained on CIFAR-10 with 20% noisy labels. As predicted by classical machine learning theory, small models underfit the training data, as indicated by high train and test error. As the model size increases, the optimal bias/variance trade-off is reached (Geman et al., 1992). Mid-sized models increasingly overfit the training data – as shown by increasing test error for decreasing train error and loss – until zero training error is achieved, and the training data is interpolated. The smallest interpolating model size is typically referred to as *interpolation threshold* (Belkin et al., 2019). Near said threshold, the test error peaks. Finally, large overparameterized models achieve improved generalization, as marked by decreasing test error, while still interpolating the training set.

Figure 1 establishes an empirical relationship between input-space smoothness and model-wise double descent. The left and middle plots show the average loss smoothness over the training set, as computed by equations 2 and 5 respectively. For increasing model size, input-space sharpness increases until near the interpolation threshold, at which highest sharpness is observed. For large models, all training points become stationary as the Jacobian norm quickly approaches zero. Second order information (Hessian norm) reveals how curvature slowly decreases past the interpolation threshold, until the loss landscape is locally flat around each training point for large model widths.

Under fixed training settings, this trend highlights how overparameterization regularizes the loss landscape, achieving *smooth* interpolation of the training data for large widths, as well as good generalization. Crucially, similarly to the test error, input-space smoothness depends nonlinearly on the model size, peaking near the interpolation threshold. This finding extends the observations of Novak et al. (2018) and LeJeune et al. (2019) from fixed-size networks to a spectrum of model sizes, and establishes a clear correlation with the test error peak in double descent. Finally, the results substantiate the universal law of robustness (Bubeck & Sellke, 2021), showing that at the interpolation threshold highest sensitivity to input perturbations is observed, while overparameterization beyond the threshold promotes smoothness. We refer the reader to section 4.5 for analogous results on ResNets.

In the following, we take a closer look at input-space smoothness and its relationship to double descent. In section 4.2 we explore smoothness over data-driven volumes around training points. In section 4.3 we focus on the role of cleanly-

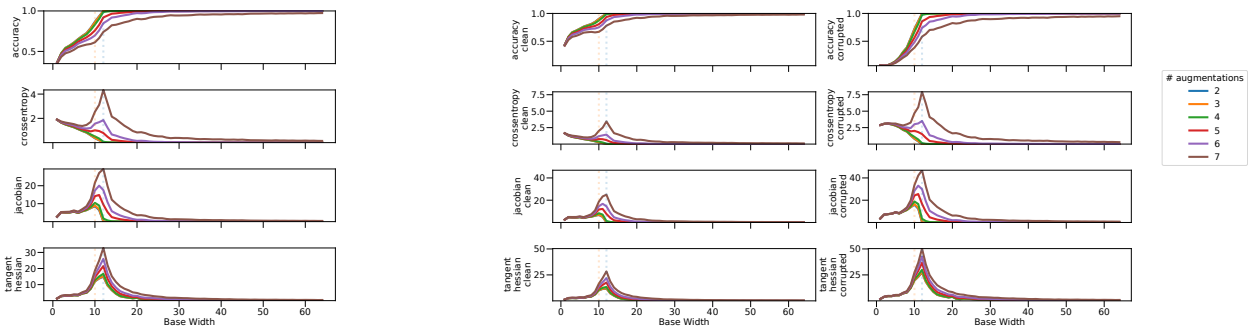


Figure 2: Accuracy, crossentropy, Jacobian and Hessian norms integrated over volumes of different sizes ($K + 1 =$ # augmentations) around all (left), or the clean (middle) and noisy (right) subsets of the CIFAR-10 training points, against varying model size of the family of ConvNets. For models near the interpolation threshold, we observe a large increase in the loss for increasing neighborhood size. Furthermore, we observe bell-shaped curves for the sharpness measures (Jacobian and Hessian norms) with their peaks around the interpolation threshold. This suggests the loss landscape in larger neighborhoods around the training data is non-flat and sharp for models around the interpolation threshold. As the overparameterization increases, the models start to smoothly interpolate the training data, even in the larger neighborhoods. We find that the loss landscape of the noisy examples for models near the interpolation threshold exhibit higher sharpness and larger variations in the loss for different neighborhood sizes (less flat).

and noisily-labelled samples, while in section 4.4 we study epoch-wise double descent. Finally, in 4.5 we shift our focus from implicit regularization to commonly found training settings.

4.2 Exploring the Loss Landscape in Larger Neighborhoods around Training Data Points

To estimate how much the loss varies around a single point, we measure the Jacobian and Hessian norms. However, when we want to measure the variation in the loss landscape in a neighborhood, it is not enough to calculate a weighted mean of the Jacobian and Hessian norms at several points in a neighborhood around a training point, we need to incorporate the variations in the loss values as well. Otherwise, we could have high Jacobian and Hessian norms with only small changes in the height of the loss surface. We make this clear with an example. Consider measuring the variation in a sinusoidal function with high frequency (k): $\sin(kx), k >> 1$. This function has a Jacobian and Hessian of $k\cos(kx)$, $-k^2\sin(kx)$, respectively. The height of the function varies between $[-1, 1]$, while the Jacobian and Hessian norms varies between $[0, k^2]$, and $[0, k^4]$, respectively. Hence, as $k >> 1$, we have small variations in the height of the function, while having large Jacobian and Hessian norms. Therefore, when measuring variations of the loss in a neighborhood, we need to separate small-scale and large-scale variations. We measure small-scale variations using Equation 6 with the Jacobian and Hessian norms, and still call this smoothness/sharpness. We measure large-scale variations in the height of the loss surface by using Equation 6 with the loss and study how it changes for different neighborhood sizes. We call the loss surface *flat* if there is only little change in the volume-based loss for increasing neighborhood sizes.

A Non-Flat and Sharp Loss Landscape for Models Near the Interpolation Threshold For models near the interpolation threshold, we observe large increases in the loss and sharpness as the neighborhood sizes is increased (see Figure 2 left). The increase in loss suggests that the loss landscape varies quickly as one moves away from the training data points, indicating even though these models are able to interpolate the data, the loss landscape is non-flat. Furthermore, the increase in Jacobian and Hessian norms suggests that the loss landscape increases in sharpness away from the training data points.

An Increasingly Flat and Smooth Loss Landscape for Models Past the Interpolation Threshold Increasing the model size past the interpolation threshold, we observe a decrease in the loss in the neighborhood around training points (see Figure 2 left). As the loss goes to zero in the neighborhood, this means that the minima in the loss landscape around the training points becomes flatter. In other words, the models start to confidently predict the given label (including noisy ones) in larger neighborhoods. Furthermore, as the sharpness measures also decrease with increasing model size, the loss surface also becomes smoother. These findings suggests that increased overparameterization makes the models smoothly interpolate the training data in larger neighborhoods.

A related finding in a concurrent work by (Somepalli et al., 2022) showed qualitatively that the distance to the decision surface for data points increased for larger model sizes by visualizing the decision surface *between* data points. Our

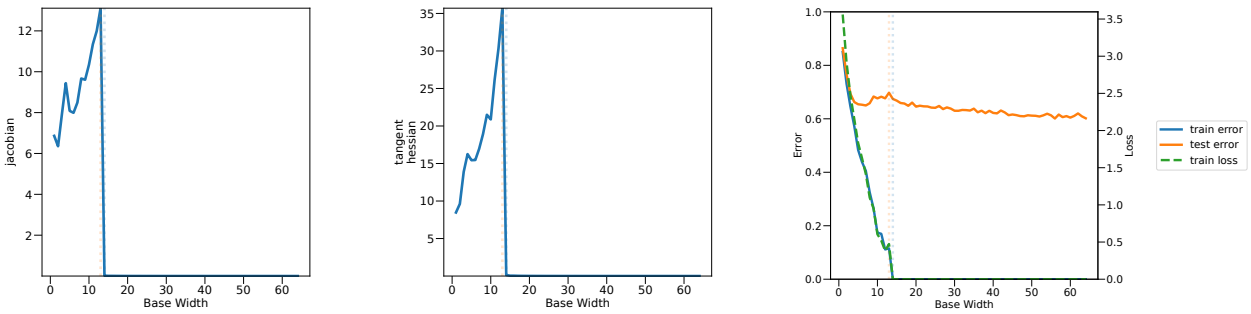


Figure 3: (Left) average Jacobian norm over the *CIFAR-100 training set without any added label noise* for a family of ConvNets of increasing base width k. (Middle) Average tangent Hessian norm on the training set, for the same ConvNets. (Right) Train and test error (0/1 loss) and training loss (crossentropy) against model size. The dotted vertical lines mark the model width that achieves zero test error (orange) and zero train error (*interpolation threshold*, blue). All models are trained for 4000 epochs. We observe the same behavior for C100 without added label noise as we did for C10 with label noise: the loss sharpness increases with model size until the peak in test error is reached, beyond which the Jacobian and Hessian norm rapidly decreases approaching zero. This suggests that our findings about smoothness are general and is not only due to added label noise.

finding is different in that we study the loss surface and we explore neighborhoods around data points while staying on the image manifold via carefully chosen augmentation strategies (see Section 3.4).

Model-wise Double Descent in the Training Loss As the neighborhood size increases, a model-wise double descent pattern in the training loss (crossentropy) becomes more prominent (see Figure 2). Similar to accuracy, the larger the neighborhood the larger the loss is, especially for the models near the interpolation threshold.

4.3 Exploring the Role of Label Noise in Double Descent

Belkin et al. (2019) argued that double descent is not always observable and that this could be due to explicit regularization (e.g., weight decay, early stopping), a too narrow sampling of the increase in model size, etc. Nakkiran et al. (2019) found that double descent was more prominent if label noise was added to the training set, i.e., the labels of some data points are randomly changed. In this section, we take a closer look at how noisy labels affects this phenomenon by studying the loss landscapes of the subsets of the training data with clean and corrupted labels separately.

A Less Flat and Sharper Loss Landscape around Noisy Data Points For models near the interpolation threshold, the sharpness in larger neighborhoods around the noisy labeled examples is much higher than for the clean ones (see Figure 2 middle and right). Interestingly, even the loss landscape of the clean examples show a peak in sharpness around the interpolation threshold.

We find that data points with clean/correct labels on average generalize to larger neighborhood sizes than the noisy/corrupted ones (see Figure 2 middle and right). For example, at the interpolation threshold, the clean training accuracy varies from 100% to ≈80%, while for noisy data points it varies between 100% to ≈60% for the various neighborhood sizes.

Model-wise Increase and Decrease in Sharpness for CIFAR-100 without added Label Noise Nakkiran et al. (2019) did not observe a model-wise double descent in the test error on CIFAR-10 without added label noise. However, the trend was observed for CIFAR-100, which we reproduce here (see Figure 3). We find that on CIFAR-100 without any added label noise, we observe similar behavior in smoothness as for CIFAR-10 with 20% added label noise: the sharpness (Jacobian and Hessian norms) increase until the peak in test error, then quickly goes towards zero as the model size increase (see Figure 3 left and middle). This suggests our findings about smoothness are general as they are not directly dependent on added label noise.

4.4 Exploring the Evolution of Sharpness During Training

In this section, we shift focus from the model-wise analysis of fully trained models, to how the loss landscape of the models evolve during training. For the evolution of Hessian norm as well as volume-based metrics, see Appendix E.1.

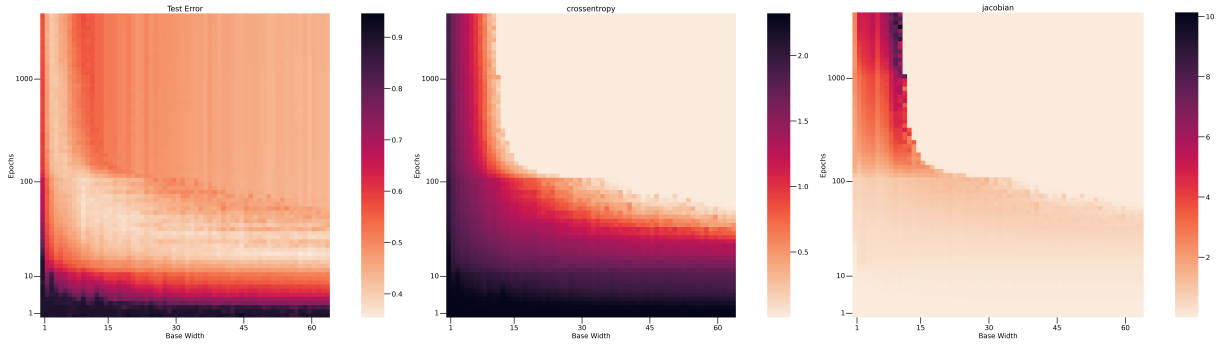
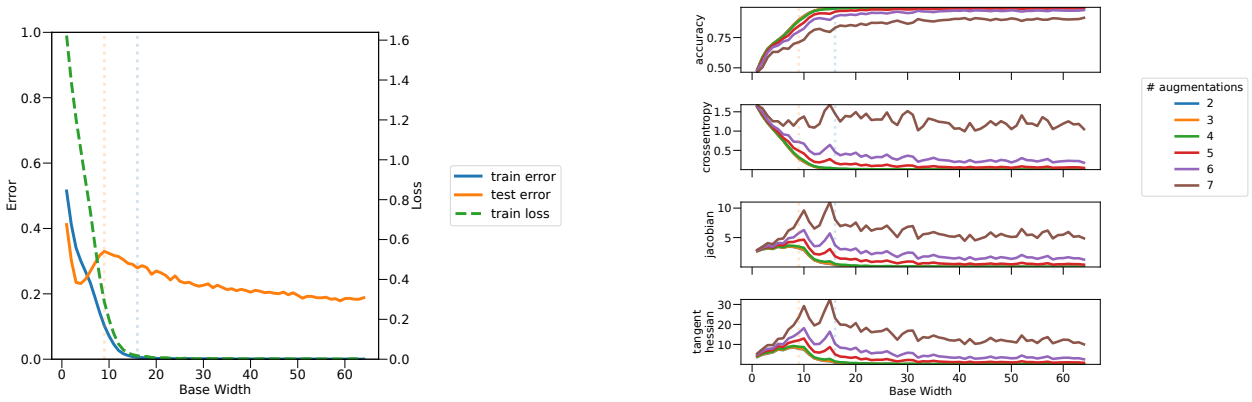


Figure 4: (Left) Test error (color) on CIFAR-10 for increasing training epochs (y-axis) against increasing base width (x-axis) of the family of ConvNets. (Middle & Right) Crossentropy loss (middle) and Jacobian norm (right) evaluated on the training set against epochs and model size. We observe an increase in loss sharpness as the models approach zero loss, which also correlates with an increase in test error. By continuing training after achieving low loss, models with large enough width are able to reduce the sharpness and test error. The sharpness increases during training for models that are not able to achieve low loss.



(a) Point-wise Performance Metrics.

(b) Volume-Based Metrics

Figure 5: (Left) Train and test error (0/1 loss) and training loss (crossentropy) on the CIFAR-10 training set for a family of ResNets of increasing base width k. The dotted vertical lines mark the model width that achieves zero test error (orange) and zero train error (*interpolation threshold*, blue). All models are trained for 4000 epochs. (Right) Accuracy, crossentropy, Jacobian and Hessian norms integrated over volumes of different sizes ($K + 1 =$ # augmentations) around training data points against varying ResNet model size. We observe the same trend of high sharpness around the interpolation threshold. However, in this modern training setup, the overparameterized models struggle to achieve smooth interpolation in larger neighborhoods around the training data.

Sharpness Increases During Training for Models Unable to Interpolate the Data We find that the point-wise sharpness gradually increase during training for the models that cannot interpolate the training data (see Figure 4). This behavior is more prominent for models closer to the interpolation threshold that can get closer to zero loss.

High Test Error and Sharpness When Models Barely Interpolate the Data Nakkiran et al. (2019) observed that in general, the worst test error is achieved when models are barely able to interpolate the training data (model-wise and epoch-wise). We also observe this as the test error increases as the loss becomes small (see Figure 4 left and middle). Interestingly, here we observe that the sharpness behaves as the test error: both increase as the loss becomes small (even during training).

Continued Training Reduces Test Error and Sharpness after Achieving Low Loss In epoch-wise double descent, the test error increases roughly when models achieve low loss and then decrease after continued training. Again, here we observe the same behavior for sharpness (see Figure 4 right).

4.5 Exploring the Role of Model Architecture and Optimization Procedure

To have as few confounding factors as possible, all the results so far have been without any explicit regularization (data augmentation, weight decay, and batch normalization), and with a simple optimizer (SGD). In this section, we investigate how our observations change when using a modern training setting. We use the ResNet architecture with the same training setup as by Nakkiran et al. (2019), i.e., Adam with data augmentation and batch normalization (see Appendix B for more details). Most trends are similar between the two training setups as detailed in Appendix E.2. Here, we focus on the main differences.

A Larger Model Size Difference between Worst Test Error and the Interpolation Threshold First, we observe a double descent in test error in this training setting as well (see Figure 5a). However, there are more model sizes between the model with the worst test error and the one at the interpolation threshold. Interestingly, in this setting, we also observe two peaks in loss and sharpness for models just after the worst test error and just before the interpolation threshold.

A Flatter and Less Sharp Loss Landscape Around the Interpolation Threshold The loss landscape is flatter as there is less variation in the loss for varying neighborhood sizes in this setting (see Figure 5b). As both training settings results in models having close to zero loss for the smallest neighborhoods, the difference in variations come from the maximum loss value, which is around 1.5 compared to 4. Furthermore, the sharpness is lower in this new training setting as well, e.g., the highest Jacobian norm is about half as large. We speculate this smoothness is mostly due to data augmentation.

A Less Flat and Sharper Loss Landscape for Large Models In this setting, the loss landscape is less flat and sharper as the cross entropy and sharpness measures only become close to zero for the neighborhoods of smaller size, but not for the larger ones. Interestingly, for the largest models the metrics seem to plateau and not decrease to zero.

5 Conclusions

Our empirical exploration of the input loss landscape with our proposed method provides observations supporting the claim that double descent and smoothness are related. In particular, we observe a model-wise increase in sharpness for models around the peak in test error which decreases for the better generalizing larger models. We observe that larger models makes the minima in the loss landscape around the training points wider and smoother, potentially providing some initial clues to why these models generalize better. Furthermore, we provide an empirical analysis showing that the loss landscape around noisy examples is less flat and smooth, which could explain why the double descent phenomenon is more apparent with noisy labels.

References

Sanjeev Arora, Zhiyuan Li, and Abhishek Panigrahi. Understanding gradient descent on edge of stability in deep learning. *arXiv preprint arXiv:2205.09745*, 2022.

Devansh Arpit, Víctor Campos, and Yoshua Bengio. How to initialize your network? robust initialization for weightnorm & resnets. *Advances in Neural Information Processing Systems*, 32, 2019.

Peter L. Bartlett, Philip M. Long, Gábor Lugosi, and Alexander Tsigler. Benign overfitting in linear regression. *Proceedings of the National Academy of Sciences*, 117(48):30063–30070, 2020. doi: 10.1073/pnas.1907378117.

Mikhail Belkin, Daniel J Hsu, and Partha Mitra. Overfitting or perfect fitting? risk bounds for classification and regression rules that interpolate. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854, 2019.

Yoshua Bengio. Deep learning of representations: Looking forward. In *International conference on statistical language and speech processing*, pp. 1–37. Springer, 2013.

Sébastien Bubeck and Mark Sellke. A universal law of robustness via isoperimetry. *Advances in Neural Information Processing Systems*, 34, 2021.

Xiangyu Chang, Yingcong Li, Samet Oymak, and Christos Thrampoulidis. Provable benefits of overparameterization in model compression: From double descent to pruning neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 6974–6983, 2021.

Tianlong Chen, Zhenyu Zhang, Sijia Liu, Shiyu Chang, and Zhangyang Wang. Robust overfitting may be mitigated by properly learned smoothening. In *International Conference on Learning Representations*, 2021.

David L. Donoho and Carrie Grimes. Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data. *Proceedings of the National Academy of Sciences*, 100(10):5591–5596, 2003. doi: 10.1073/pnas.1031596100.

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations*, 2020.

Matteo Gamba, Adrian Chmielewski-Anders, Josephine Sullivan, Hossein Azizpour, and Mårten Björkman. Are all linear regions created equal? In *International Conference on Artificial Intelligence and Statistics*, pp. 6573–6590. PMLR, 2022.

Mario Geiger, Stefano Spigler, Stéphane d'Ascoli, Levent Sagun, Marco Baity-Jesi, Giulio Biroli, and Matthieu Wyart. Jamming transition as a paradigm to understand the loss landscape of deep neural networks. *Physical Review E*, 100 (1):012115, 2019.

Stuart Geman, Elie Bienenstock, and René Doursat. Neural Networks and the Bias/Variance Dilemma. *Neural Computation*, 4(1):1–58, 01 1992. ISSN 0899-7667. doi: 10.1162/neco.1992.4.1.1.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1026–1034, 2015.

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

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.

Rohith Kuditipudi, Xiang Wang, Holden Lee, Yi Zhang, Zhiyuan Li, Wei Hu, Rong Ge, and Sanjeev Arora. Explaining landscape connectivity of low-cost solutions for multilayer nets. *Advances in neural information processing systems*, 32, 2019.

Daniel LeJeune, Randall Balestriero, Hamid Javadi, and Richard G Baraniuk. Implicit rugosity regularization via data augmentation. *arXiv preprint arXiv:1905.11639*, 2019.

Zhiyuan Li, Tianhao Wang, and Sanjeev Arora. What happens after sgd reaches zero loss?–a mathematical framework. In *International Conference on Learning Representations*, 2021.

Chao Ma and Lexing Ying. On linear stability of sgd and input-smoothness of neural networks. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 16805–16817. Curran Associates, Inc., 2021.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Jonathan Uesato, and Pascal Frossard. Robustness via curvature regularization, and vice versa. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Vidya Muthukumar, Kailas Vodrahalli, Vignesh Subramanian, and Anant Sahai. Harmless interpolation of noisy data in regression. *IEEE Journal on Selected Areas in Information Theory*, 1(1):67–83, 2020.

Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. Deep double descent: Where bigger models and more data hurt. In *International Conference on Learning Representations*, 2019.

Hariharan Narayanan and Sanjoy Mitter. Sample complexity of testing the manifold hypothesis. *Advances in neural information processing systems*, 23, 2010.

Brady Neal, Sarthak Mittal, Aristide Baratin, Vinayak Tantia, Matthew Scicluna, Simon Lacoste-Julien, and Ioannis Mitliagkas. A modern take on the bias-variance tradeoff in neural networks. *arXiv preprint arXiv:1810.08591*, 2018.

Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. In search of the real inductive bias: On the role of implicit regularization in deep learning. In *International Conference on Learning Representations Workshop Track*, 2015.

Behnam Neyshabur, Zhiyuan Li, Srinadh Bhojanapalli, Yann LeCun, and Nathan Srebro. The role of over-parametrization in generalization of neural networks. In *International Conference on Learning Representations*, 2018.

Roman Novak, Yasaman Bahri, Daniel A Abolafia, Jeffrey Pennington, and Jascha Sohl-Dickstein. Sensitivity and generalization in neural networks: an empirical study. In *International Conference on Learning Representations*, 2018.

Phil Pope, Chen Zhu, Ahmed Abdelkader, Micah Goldblum, and Tom Goldstein. The intrinsic dimension of images and its impact on learning. In *International Conference on Learning Representations*, 2020.

Leslie Rice, Eric Wong, and Zico Kolter. Overfitting in adversarially robust deep learning. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 8093–8104. PMLR, 13–18 Jul 2020.

Mihaela Rosca, Theophane Weber, Arthur Gretton, and Shakir Mohamed. A case for new neural network smoothness constraints. *NeurIPS Workshops*, 2020.

Gowthami Somepalli, Liam Fowl, Arpit Bansal, Ping Yeh-Chiang, Yehuda Dar, Richard Baraniuk, Micah Goldblum, and Tom Goldstein. Can neural nets learn the same model twice? investigating reproducibility and double descent from the decision boundary perspective. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13699–13708, 2022.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Zeke Xie, Issei Sato, and Masashi Sugiyama. A diffusion theory for deep learning dynamics: Stochastic gradient descent exponentially favors flat minima. In *International Conference on Learning Representations*, 2020.

Zitong Yang, Yaodong Yu, Chong You, Jacob Steinhardt, and Yi Ma. Rethinking bias-variance trade-off for generalization of neural networks. In *International Conference on Machine Learning*, pp. 10767–10777. PMLR, 2020.

Tao Yu, Huan Long, and John E Hopcroft. Curvature-based comparison of two neural networks. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pp. 441–447. IEEE, 2018.

A Appendix

B Network Architectures and Training Setup

This section details the training set up, as well as the network architectures used.

Network architectures The ConvNets and ResNets used follow the experimental settings of Nakkiran et al. (2019), with the only difference that we disable batch normalization in order to focus our study on implicit regularization. In summary, the ConvNets are composed of 4 convolutional stages (each with a single conv + ReLU block) with kernel size 3×3, stride 1, padding 1, each followed by maxpooling of stride 2 and kernel size 2×2. Finally, a max pooling layer of stride 2 and kernel size 2×2 is applied, followed by a linear layer. The Residual networks used in this study are ResNet18s (He et al., 2015) without batch normalization.

Both ConvNets and ResNets are formed by 4 convolutional stages at which the number of learned feature maps doubles, i.e. the base width of each stage follows the progression $[k, 2k, 4k, 8k]$, with $k = 64$ denoting a standard ResNet18. To control the number of parameters in each network, the base width k varies from 1 to 64.

Dataset splits To tune the training hyperparameters of all networks, a validation split of 1000 samples was drawn uniformly at random from the training split of CIFAR-10 and CIFAR-100.

ConvNet Training setup The training settings are the same for CIFAR-10 and CIFAR-100. All ConvNets are trained for 4000 epochs with SGD with momentum 0.9, fixed learning rate $1\mathrm{e}-3$, batch size 128, and no weight decay. All learned layers are initialized with Pytorch's default weight initialization (version 1.11.0). To stabilize prolonged training in the absence of batch normalization, we use learning rate warmup: starting from a base value of $1\mathrm{e}-4$ the learning rate is linearly increased to $1\mathrm{e}-3$ during the first 5 epochs of training, after which it remains constant at $1\mathrm{e}-3$.

ResNet training setup All ResNets are trained for 4000 epochs using Adam with base learning rate $1\mathrm{e}-4$, batch size 128, and no weight decay. All learned layers are initialized with Pytorch's default initialization (version 1.11.0). All residual networks are trained with data augmentation, consisting of $4-pixel$ random shifts, and random horizontal flips.

C Tangent Hessian Computation

To estimate the tangent Hessian norm at a point $\mathbf{x}_n$ through equation 5, we approximate the tangent space to the data manifold local to $\mathbf{x}_n$ by using a set of random weak augmentations of $\mathbf{x}_n$. To guarantee that all augmentations $\mathbf{x}_n + \mathbf{u}_m$, as well as the displacements $\mathbf{x}_n + \delta\mathbf{u}_m$ lie on the data manifold, we use weak colour augmentations as follows.

For each sample $\mathbf{x}_n$, we apply in random order the following photometric transformations:

- random brightness transformation in the range $[0.9, 1.1]$, with 1. denoting the identity transformation.
- random contrast transformation in $[0.9, 1.1]$, with 1. denoting the identity transformation.
- random saturation transformation in $[0.9, 1.1]$, with 1. denoting the identity transformation.
- random hue transformation in $[-0.05, 0.05]$, with 0. denoting the identity transformation.

Furthermore, a step size $\delta = 0.1$ is used for computing the finite differences in equation 5. 4 augmentations $\mathbf{x}_n + \mathbf{u}_m$ are sampled for each point. All randomness is controlled to ensure reproducibility. Figure 6 (left) shows a visualization of the colour augmentations used.

D SVD Augmentation

The SVD augmentation method presented in section 3.4 allows for generating images that lie in close proximity to the base sample $\mathbf{x}_n$. Figure 6 shows an illustration of the original image (first column) and several augmented images, as the augmentation strength (number of erased singular values) increases. Figure 7 shows the average (over the CIFAR-10 training set) Euclidean distance of augmented samples from their respective base sample, as well as the length of the polygonal path formed by connecting augmentations of increasing strength. We note that for $k < 30$, in expectation, augmentations lie in close proximity to the original base sample in Euclidean space.

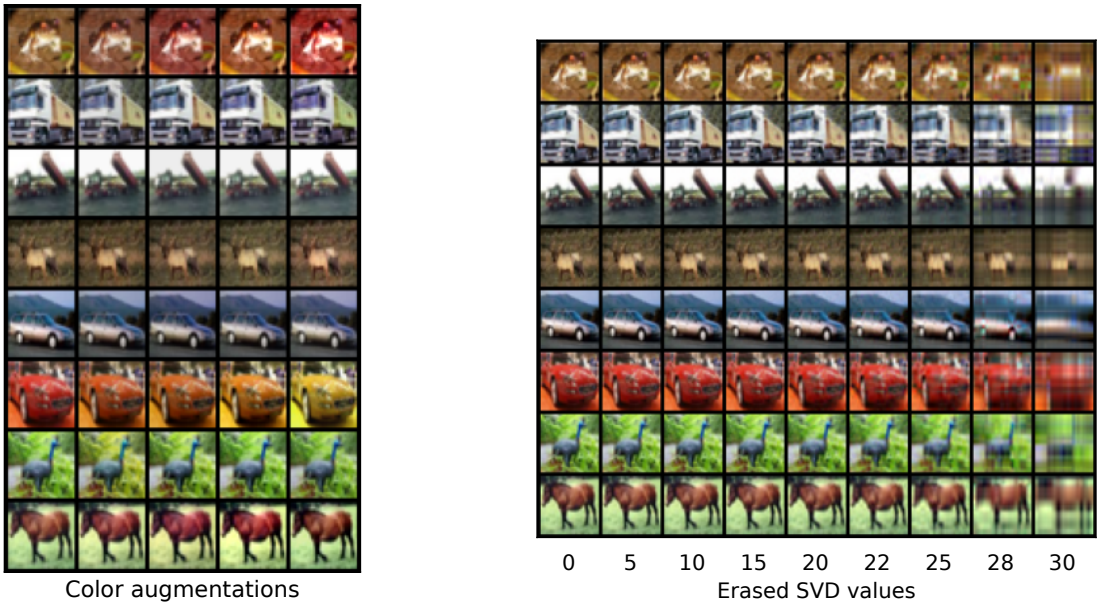


Figure 6: (Left) Visualization of random colour augmentations used to estimate the tangent Hessian norm. Each row represents a set of random augmentation, with the first image per-row showing the corresponding base sample. (Right) Each row represents SVD augmentations of increasing strength. Also in this case, the first column represents the base sample used to generate the corresponding augmentations in each row.

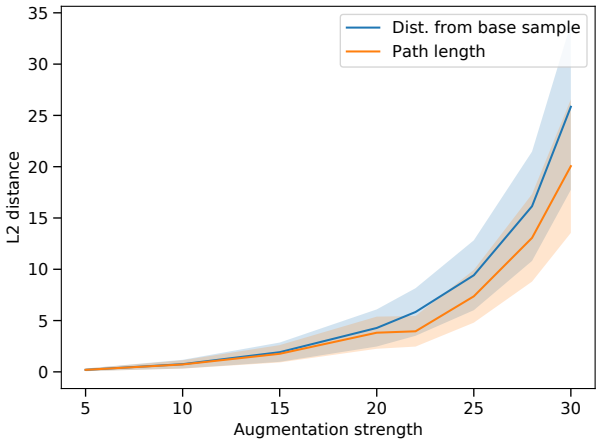


Figure 7: Average L2 distance from the base samples, for augmentations of increasing strength.

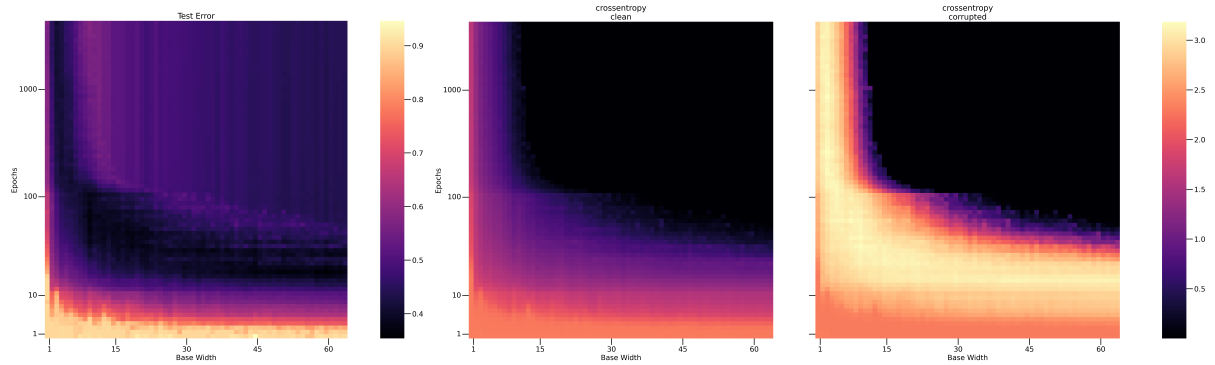


Figure 8: Test error (left) and loss at the data points over the course of training (y-axis) for different model sizes (x-axis).

E Additional Experiments

E.1 Exploring the Evolution of the Loss Landscape During Training

E.1.1 The Evolution of the Loss at Data Points during Training

Epoch-wise Double Descent in Test Error We reproduce the epoch-wise double descent in test error for large enough models at the data points (no neighborhood) as observed by Nakkiran et al. (2019) (see Figure 8 left).

Clean and Noisy Samples are Consistently Learnt at Different Speeds We find that clean data points are prioritized early in training and only later is the noisy ones fit (see Figure 8 middle and right). Only after the models have a low loss for the clean data points, the large enough models start to overfit to the noisy ones. Interestingly, there seem to be at least a correlation between when the models start to overfit to the noisy data points and when the test error start to increase.

The Loss of Noisy Samples First Increase then Decrease During Early Training One would expect that the loss at the training data points would decrease during training, which is what we see for clean examples (see Figure 8 middle). However, interestingly, the loss of the noisy examples first increase as the clean examples are fit, and then decrease.

E.1.2 The Evolution of Volume-Based Sharpness of the Loss Landscape during Training

For completeness, we also provide epoch-wise results for cross-entropy as well as Jacobian and Hessian norm evaluated in a neighborhood (see Figure 9 right).

E.2 Experiments with ResNets and Explicit Regularization

The goal of this section is to go over most of the observations we made about the training setup for the CNN models, but now for the training setup with the ResNet models (except for the ones already covered in Section 4.5).

E.2.1 Model-wise Observations

Sharpness at Data Points Increases Until Peak in Test Error then Decreases. The sharpness of the ResNet models also increase and decrease as the model size increases (see Figure 10). However, in this setting, both the sharpness measures are peaking just before the test error, not at it.

Model-wise Double Descent in Volume-based Training Loss. We can still observe this phenomenon, but the peak around the interpolation threshold is less clear in this setting (see Figure 13 top right).

Volume-based Jacobian Norm Moves from Worst Test Error to Interpolation Threshold for Increasing Neighborhood Size. Here, a more complex pattern emerges. The peak in sharpness around the test error does move towards larger widths, but not all the way to the interpolation threshold (see Figure 11 right). However, a second peak close to the interpolation threshold appears as the neighborhood size increase.

A Flat and Quite Smooth Loss Landscape for Small Models. Observed (see Figure 11).

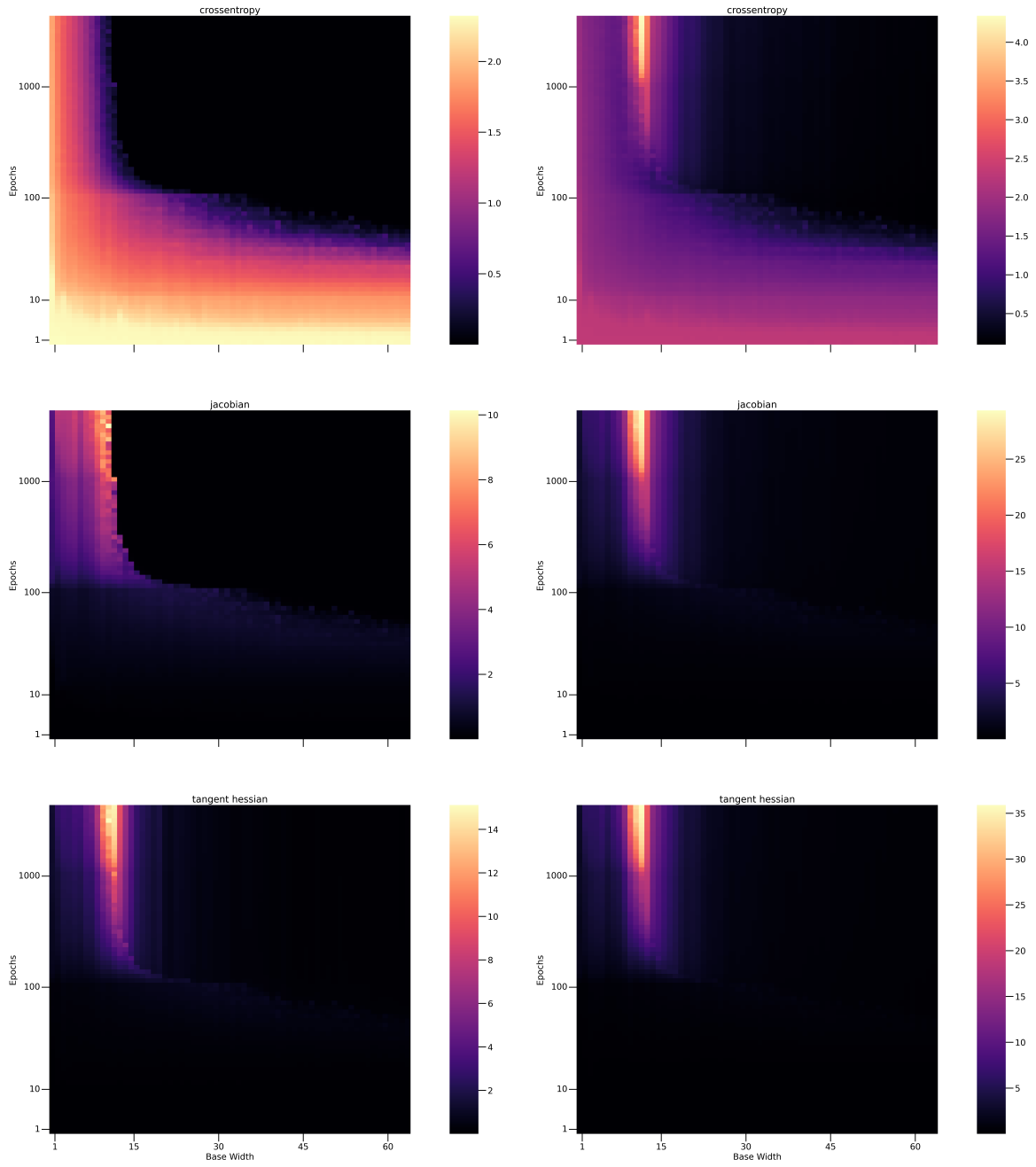


Figure 9: Left: Metrics evaluated at all the training data points. Right: Metrics in a neighborhood (7 augmentations) around all the training data points. Note that each heatmap has its own range of values.

The Loss Landscape is Less Flat and Sharper around Noisy Data Points. Observed (see Figure 11).

Small Models Prioritize Fitting Clean Data Points. Observed (see Figure 11).

E.2.2 Epoch-wise Observations

Clean and Noisy Samples are Consistently Learnt at Different Speeds. Observed (see Figure 12).

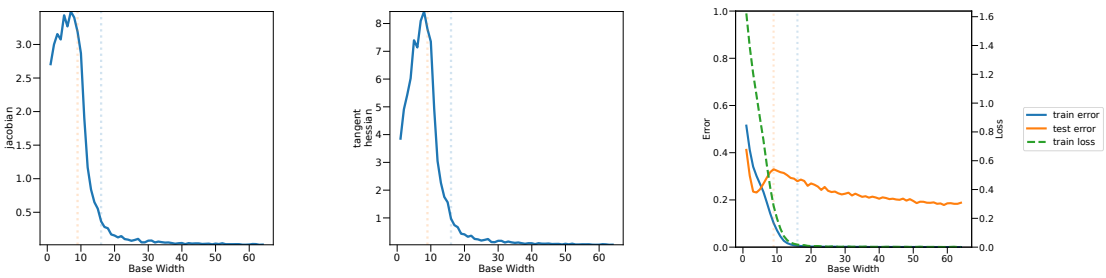


Figure 10: Sharpness (left & middle) metrics at the training points and train and test losses (right) for ResNets of varying model sizes.

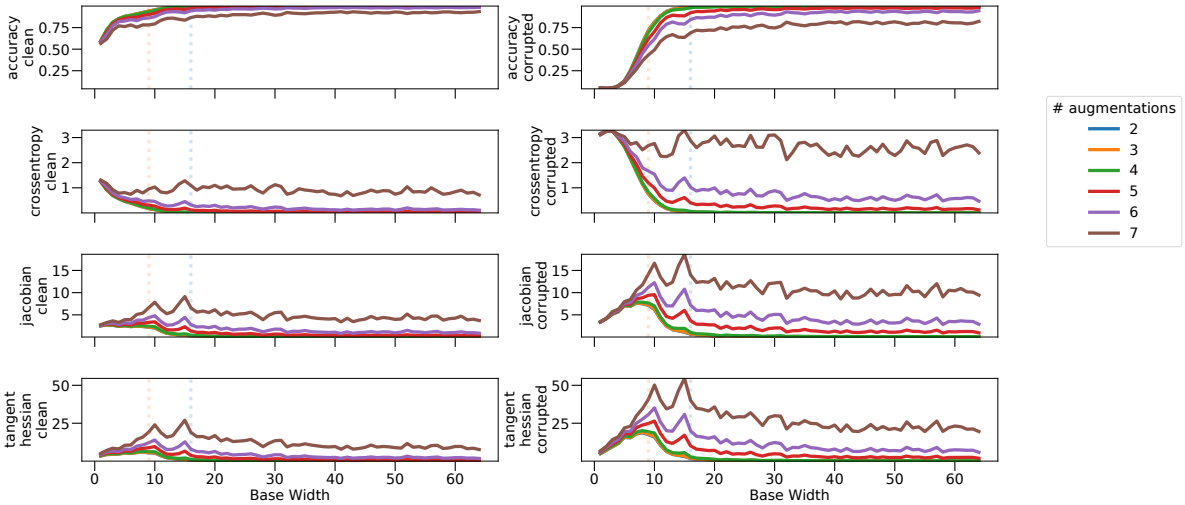


Figure 11: The ResNet models evaluated with volume-based metrics for the clean (left) and noisy (right) data points of the training set.

The Loss of Noisy Samples First Increase Then Decrease During Early Training. Observed, but not as clear because the increase in loss happens during fewer epochs (see Figure 12 right). We speculate that this is because no learning rate warm-up is used for the ResNet models.

Sharpness of Small and Intermediate Models Increase During Training. With this training setting, the ResNet models show a more complex behavior. The models do increase in sharpness during the later stages of training (see Figure 13). However, a difference is that early in training these models have a high sharpness that decreases before it increases again. Past the peak in sharpness, some models even show an epoch-wise double descent in sharpness.

Large Models Are Not Always Smooth During Training. Observed more clearly in this setting, especially for the point-wise sharpness (see Figure 13). The sharpness first increases then decreases as the training progresses.

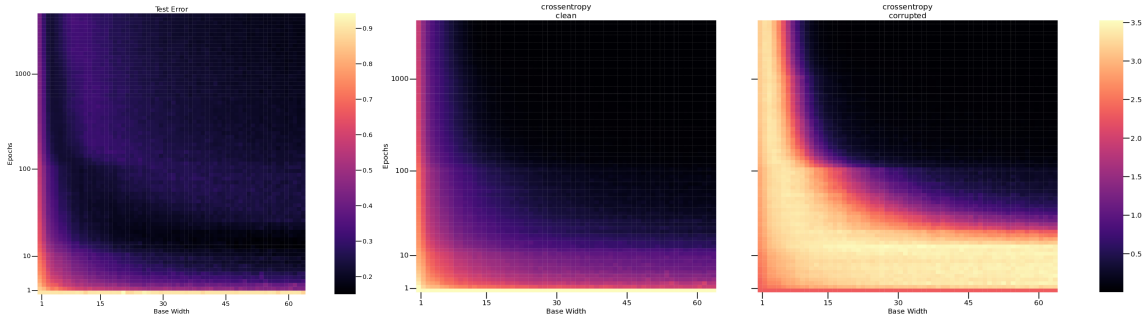


Figure 12: Test error (left) and loss at the data points over the course of training (y-axis) for different ResNet model sizes (x-axis).

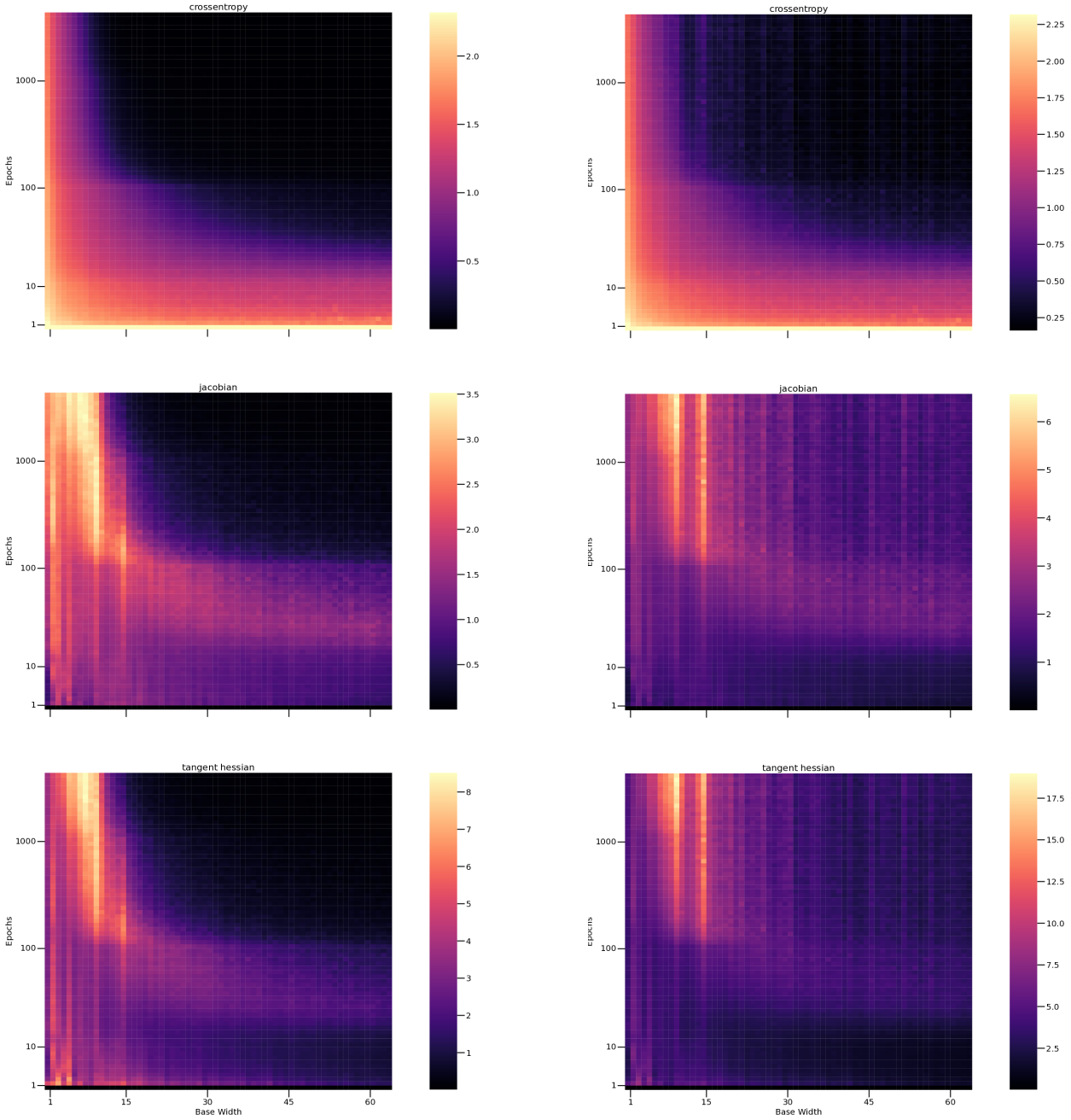


Figure 13: Metrics (y-axis) evaluated at (left) and in neighborhoods (6 augmentations) around (right) all the training data points for different ResNet model sizes (x-axis). Note that each heatmap has its own range of values.