Overparameterization Implicitly Regularizes Input-Space Smoothness

Matteo Gamba
KTH
Stockholm
mgamba@kth.se

Hossein Azizpour
KTH
Stockholm
azizpour@kth.se

Mårten Björkman
KTH
Stockholm
celle@kth.se

Abstract

Existing bounds on the generalization error of deep networks assume some form of smooth or bounded dependence on the input variable and intermediate activations, falling short of investigating the mechanisms controlling such factors in practice. In this work, we present an empirical study of the Lipschitz constant of networks trained in practice, as the number of model parameters and training epochs vary. We present non-monotonic trends for the Lipschitz constant, strongly correlating with double descent for the test error. Our findings highlight a theoretical shortcoming in modeling input-space smoothness via uniform bounds.

1 Introduction

A longstanding question towards understanding the remarkable generalization ability of deep networks is characterizing the hypothesis class of models *trained in practice* (Hanin & Rolnick, 2019b; Novak et al., 2018). Indeed, finding a parameterization that accurately describes the class of generalizing trained networks could shed light on the mechanisms controlling model complexity (Neyshabur et al., 2015a,b). At present, constraints have been derived from postulated assumptions on the training data (Wei & Ma, 2019), loss (Bartlett et al., 2017), model architecture (Hanin & Rolnick, 2019b,a) as well as global optima (Ma & Ying, 2021). In this context, empirical studies play an important role in isolating trends that could inform theoretical research (Zhang et al., 2019; Jiang et al., 2019; Zhang et al., 2018; Keskar et al., 2017). Chiefly, a central question is understanding the role of overparameterization (Arora et al., 2018; Zhang et al., 2018), a key property of state of the art models.

Increasing evidence shows that large generalizing networks express smooth functions of the training data (Ma & Ying, 2021), and that input-space smoothness can be predictive of generalization (LeJeune et al., 2019; Novak et al., 2018). Importantly, a recent line of works connects input-space smoothness to the double descent curve of the test error (Belkin et al., 2019), highlighting a non-monotonic dependency of several quantities from model size, namely nonlinearity of the model function (Gamba et al., 2022a), sharpness of the loss landscape (Gamba et al., 2022b), as well as consistency of the decision function and reproducibility of decision boundaries (Somepalli et al., 2022).

At the same time, many theoretical works bounding the test error of neural networks *assume* some form of input-space regularity – via uniformly bounded Lipschitz constant of the model function and layers – in order to prove generalization (Ma & Ying, 2021; Wei & Ma, 2019; Nagarajan & Kolter, 2018), falling short of investigating the mechanisms promoting the assumed regularity. *Specifically, it is unclear whether simple uniform bounds on the Lipschitz constant provide a faithful representation of the hypothesis class of trained networks*. Recently, the problem has received renewed attention, with Bubeck & Sellke (2021) proposing a lower bound for a general class of interpolating models in terms of sample size and number of parameters, and Roth et al. (2020) studying the Lipschitz constant in adversarial training for fixed-size networks. However, no systematic study of implicit regularization driven by overparameterization in practical settings has been carried out so far.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

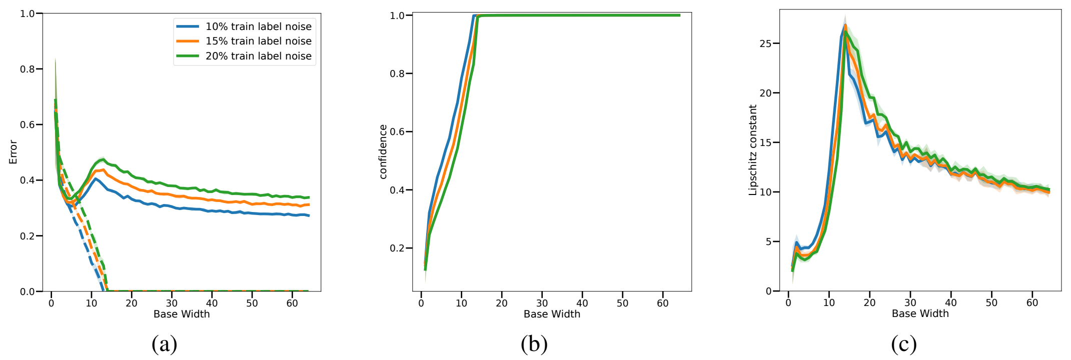


Figure 1: (a) Double descent curve for the test error for a family of ConvNets of increasing model size, trained on CIFAR-10 with corrupted train labels. Solid lines (resp. dashed lines) denote the test error (resp. train error). (b) Average prediction confidence over the training set, for the same networks. (c) Empirical Lipschitz constant of the network function, peaking at the interpolation threshold, and then decreasing for increasing model size. All standard deviations are computed over 3 random seeds.

Contributions In this work, we revisit input-space smoothness of deep networks and present an empirical study of the Lipschitz constant of model functions parameterized by ReLU networks.

- By controlling the degree of parameterization of trained models, we experimentally show that the empirical Lipschitz constant of the network is predictive of model-wise double descent (Belkin et al., 2019), thereby explaining trends reported in the literature for loss landscape sharpness for the same family of networks.
- We present evidence that the non-monotonic trend of the Lipschitz might be inherently tied to the dynamics of SGD, and furthermore that it may hold globally in the input space.

2 Methodology

We study feed-forward networks composing L linear layers with the continuous piece-wise linear activation ReLU $\phi(x) = \max\{0, x\}$, interpreted as functions $\mathbf{f}(\mathbf{x}, \mathcal{W}) = \mathbf{W}^L\phi(\mathbf{W}^{L-1}\phi(\cdots\phi(\mathbf{W}^1\mathbf{x} + \mathbf{b}^1)) + \mathbf{b}^{L-1}) + \mathbf{b}^L$, which are themselves continuous piece-wise linear[1] in the input variable $\mathbf{x} \in \mathbb{R}^d$. Due to such property, ReLU networks partition their input space into convex polytopes known as activation regions (Hanin & Rolnick, 2019a; Raghu et al., 2017; Montufar et al., 2014), on each of which a linear function is computed. In the notation of Rahaman et al. (2019), $\mathbf{f}$ can be written as

$$\mathbf{f}(\mathbf{x}, \mathcal{W}) = \sum_{\epsilon} \mathbb{1}_{\epsilon}(\mathbf{x})\mathbf{W}_{\epsilon}\mathbf{x} + \mathbf{b}_{\epsilon} \tag{1}$$

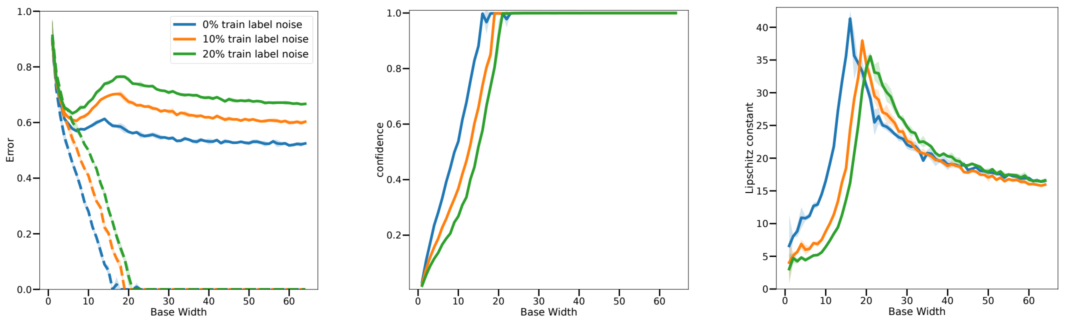
where the index ϵ denotes an activation region, and the indicator function represents conditioning the factorization $\mathbf{W}_{\epsilon} := \prod_{\ell=1}^{L} \operatorname{diag}(\phi_{\mathbf{x}}^{\ell})\mathbf{W}^{\ell}$ by the binary activation pattern $\phi_{\mathbf{x}}^{\ell}$ of each ReLU according to the preactivation of the corresponding layer ℓ, dependent on the input $\mathbf{x}$ to the network.

A recent work (Gamba et al., 2022b) studies the Jacobian norm $\|\nabla_{\mathbf{x}}\mathcal{L}(\mathbf{x}, \mathcal{W}, y)\|$ of crossentropy loss $\mathcal{L} : \mathbb{R}^d \times \{1, \dots, K\} \to \mathbb{R}$ with respect to the input variable as a measure of sharpness, and shows that it correlates with the peak and descent in test error for large interpolating models. In this work, we decompose the Jacobian norm of the loss as the product

$$\|\nabla_{\mathbf{x}}(\mathcal{L} \circ \mathbf{f})\| = \|(\mathbf{e}_y - \hat{\mathbf{y}}) \cdot \nabla_{\mathbf{x}}\mathbf{f}\| \tag{2}$$

where $\mathbf{e}_y$ is a K-dimensional one-hot encoded vector with the y:th component set to 1, and $\hat{\mathbf{y}}$ are the softmax scores of the network logits. We note that the left factor in the product is related to prediction confidence $1 - \|\mathbf{e}_y - \hat{\mathbf{y}}\|^2$, and focus on input smoothness as measured by the Jacobian $\nabla_{\mathbf{x}}\mathbf{f}$.

[1]Convolutional as well as residual networks can be represented this way by vectorizing their weight tensors.



(a) (b) (c)

Figure 2: (a) Double descent curve for the test error for a family of ConvNets of increasing model size, trained on CIFAR-100 with corrupted train labels. Solid lines (resp. dashed lines) denote the test error (resp. train error). (b) Average prediction confidence over the training set, for the same networks. (c) Empirical Lipschitz constant of the network function, peaking at the interpolation threshold, and then decreasing for increasing model size. Standard deviations are computed over 3 random seeds.

Operator norm To measure smoothness of the piece-wise linear $\mathbf{f}$, we study the operator norm of the Jacobian $\nabla_{\mathbf{x}}\mathbf{f}$. For linear operators $A : (\mathbb{R}^d, \|\cdot\|_p) \to (\mathbb{R}^K, \|\cdot\|_q)$, such norm is defined as $\|A\|_{\text{op}} := \sup_{\mathbf{x}:\|\mathbf{x}\|_p \neq 0} \frac{\|A\mathbf{x}\|_q}{\|\mathbf{x}\|_p}$, where the norms $\|\cdot\|_p$ and $\|\cdot\|_q$ are respectively taken in input and logit space. For linear operators, the operator norm measures the maximum change propagated by the function, and can be thought of as a measure of its scale. Crucially, if $p = q = 2$, then the operator norm can be estimated by computing the largest singular value of A.

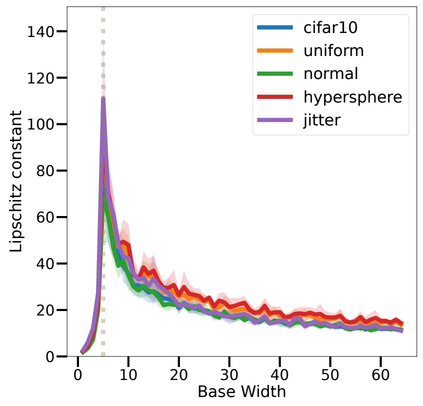
For any data point $\overline{\mathbf{x}} \in \mathbb{R}^d$, evaluating the Jacobian at $\overline{\mathbf{x}}$ yields $\nabla_{\mathbf{x}}\mathbf{f}(\mathbf{x}, \mathcal{W})_{\big|\mathbf{x}=\overline{\mathbf{x}}} = \mathbf{W}_\epsilon$, i.e. the linear function computed by $\mathbf{f}$ on the activation region ϵ of $\overline{\mathbf{x}}$. Hence, at each point, $\|\mathbf{W}_\epsilon\|_{\text{op}}$ provides an estimate of worst-case sensitivity of the linear "piece". We note that, while the supremum $\|A\|_{\text{op}}$ may not be attained within the activation region of $\overline{\mathbf{x}}$, the operator norm upper bounds worst-case sensitivity within the region. Furthermore, activation regions neighbouring training data tend to compute approximately the same linear function (Gamba et al., 2022a; Roth et al., 2020).

Computing the operator norm Computing the operator norm of $\mathbf{W}_\epsilon \in \mathbb{R}^{K\times d}$ entails two steps. First, computing the gradient $\nabla_{\mathbf{x}}\mathbf{f}_{\big|\mathbf{x}=\overline{\mathbf{x}}} = \mathbf{W}_\epsilon$ (via automatic differentiation), and then estimating its largest singular value. To perform the latter, we use a standard power method, detailed in section C.

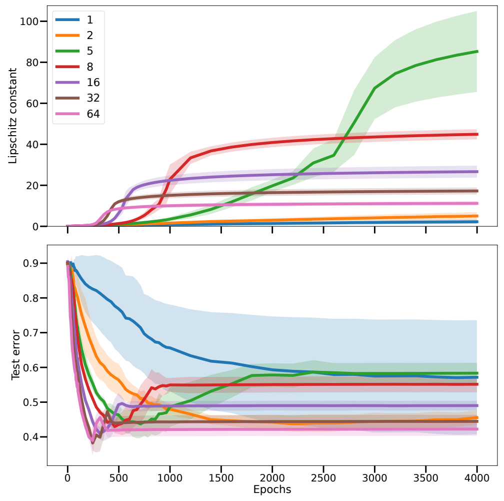
Empirical Lipschitz constant Trivially, the operator norm $\|A\|_{\text{op}}$ corresponds to the smallest Lipschitz constant for A since, by definition, if $\|A\mathbf{u} - A\mathbf{v}\|_q \leq \gamma\|\mathbf{u} - \mathbf{v}\|_p, \forall \mathbf{u}, \mathbf{v} \in \mathbb{R}^d$ and $\gamma > 0$, then $\|A\|_{\text{op}} = \sup_{\mathbf{u}:\|\mathbf{u}\|_p=1} \|A\mathbf{u}\|_q \leq \gamma$. In our experiments, we estimate the Lipschitz constant of the network by its empirical constant, $\gamma_N = \frac{1}{N}\sum_{n=1}^{N} \|\nabla_{\mathbf{x}}\mathbf{f}_{\big|\mathbf{x}=\mathbf{x}_n}\|_{\text{op}}$, for all training points $\mathbf{x}_n \in \{\mathbf{x}_n, y_n\}_{n=1}^N$, as well as validation data. In the next section, we empirically show that the double descent trend observed for the loss function in related works originates from the piece-wise linear function $\mathbf{f}$, and study the empirical Lipschitz constant of networks trained in practice.

3 Experiments

In this section, we present a study of the empirical Lipschitz constant of trained networks as their number of parameters varies. The full experimental setup is detailed in section B. We reproduce the double descent curves of the test error by training a family of 5-layer ConvNets as well as ResNet18s (He et al., 2015) on the CIFAR datasets (Krizhevsky et al., 2009) with 10%, 15% and 20% of the training labels randomly perturbed. Following Nakkiran et al. (2019), we control model size by increasing the base width w (number of learned feature maps) of each convolutional stage in both model families, following the progression $[w, 2w, 4w, 8w]$, and varying $w = 1, \ldots 64$. To isolate the



(a)



(b)

Figure 3: (a) ResNets trained on CIFAR-10 with 20% corrupted labels evaluated using random test data. The results suggest that the non-monotonic trend of the empirical Lipschitz constant might be a global property of the network function. Interestingly, the Lipschitz constant is lower for data lying closer to the training data. (b) Development over epochs of the Lipschitz constant (top) and the test error (bottom) for several model sizes. Underfitting models (low base width) maintain a low Lipschitz throughout training, while models harmfully overfitting the training data keep increasing the Lipschitz constant. Finally, large overparameterized generalizing models undergo a epoch-wise double descent in the test error, associated with plateauing Lipschitz constant for larger models, which is smallest for the largest models. All results averaged over 5 random seeds.

role of overparameterization (and its interaction with the optimizer), we train the networks with SGD with momentum and fixed learning rate, without explicit regularization. Figure 1a and 2a show the double descent curve for the test error for our experimental setting. Analogous results for ResNet are presented in Figure 4a. In the following, we denote with *interpolation threshold* the smallest model size that can perfectly classify the training data (i.e. reaching zero training error) (Belkin et al., 2019).

Lipschitz constant follows model-wise double descent Figures 1a and 1b show that, while the test error undergoes double descent as the model size increases, the mean confidence $1 - \frac{1}{N}\sum_{n=1}^{N} \|\mathbf{e}_{y_n} - \hat{\mathbf{y}}_n\|^2$ (proportional to the left factor in Equation 2) grows monotonically, with all models past interpolation threshold predicting the training targets with very high confidence. At the same time, the Lipschitz constant grows non-monotonically with model size (Figures 1c and 2c), peaking at interpolation threshold and then decreasing as the degree of parameterization increases. This finding highlights the need for better understanding the mechanisms implicitly regularizing smoothness in relation to model size. Postulated monotonic bounds on the Lipschitz therefore do not accurately model the effective complexity of trained networks and may hide important information.

Beyond training data We now turn our attention to validation data, and probe a series of ResNet18s trained on CIFAR-10 with 20% corrupted training labels, by using randomly generated data, lying increasingly far from the support of the data distribution. We generate random validation data by sampling i.i.d. pixels with per-channel distribution from the following families: uniform with mean and standard deviation matching the per-pixel distribution of the training split of CIFAR-10, isotropic normal with analogous moments, pixels randomly sampled from within a unit hypersphere, as well as strong random jitter applied to each CIFAR-10 training image. Intriguingly, the emprical Lipschitz constant estimated on validation data closely follows that of standard CIFAR-10, with a sharp peak at the interpolation threshold, suggesting that the observed property might hold globally in the input space for the network function, depending only weakly on training data *at convergence*.

Overparameterization affects epoch-wise double descent Finally, we study the Lipschitz constant throughout prolonged training, for ResNets trained for 4k epochs. Figure 3b reports the test error (bottom) and empirical Lipschitz constant (top) as a function of training epochs. We observe how only large overparameterized models undergo epoch-wise double descent, corresponding to a plateau of the Lipschitz constant, which stably holds throughout the rest of training. At the same time, models at the interpolation threshold (base width 5) present a sharply increasing Lipschitz constant, matching the model-wise trends reported in Figures 1c and 4c. This finding suggests that overparameterization might play a role in biasing the learning dynamics of SGD, and that hypothesis-space studies alone might not be able to explain how smooth interpolating solutions are found in practice.

4 Related Work

Capturing the complexity of deep networks trained in practice is a challenging open problem that may require taking into account the data distribution, the model class, as well as the optimizer (Kawaguchi et al., 2017). Theoretical studies via the Rademacher complexity hinge upon finding a correct parameterization of the hypothesis class of neural networks trained in practice (Neyshabur et al., 2018). Several works constrain parameterizations by assuming upper bounds on the Lipschitz constant in order to bound the norm of layer activations and their gradients (Ma & Ying, 2021; Wei & Ma, 2019; Nagarajan & Kolter, 2018; Bartlett et al., 2017). In this work, we argue that expressing such bounds through an abstract constant may hide the effective complexity of model functions relized by networks trained with stochastic gradient descent. Specifically, we show that by varying the model size, the Lipschitz constant presents non-monotonic behaviour correlating with the test error.

Recently, characterizing robustness in relation to overparameterization has been formally studied by Bubeck & Sellke (2021) for a generic class of interpolating models, providing the first lower bound on the Lipschitz constant that decreases with increasing model size, suggesting that overparameterization is necessary in order to achieve robustness. At present, it is unclear how to express a tight upper bound in terms of the fundamental components of neural networks. In this work, we take a first step in this direction by studying the Lipschitz constant in relation to optimization. Our results empirically substantiate Bubeck & Sellke (2021), as well as highlight the importance of incorporating the optimizer into the hypothesis class of neural networks.

Recent works study nonlinearity of the model function (Gamba et al., 2022a), regularity of decision boundaries (Somepalli et al., 2022), as well as sharpness of the loss (Gamba et al., 2022b) in relationship to double descent, while works on adversarial training study the relationship between curvature and model robustness for fixed-size models (Moosavi-Dezfooli et al., 2019). By studying the Lipschitz constant in the double descent setting, our work explains the trends reported for loss sharpness in (Gamba et al., 2022b) by connecting them to the Lipschitz constant of the underlying network function. Finally, our work extends the observations of Novak et al. (2018) and LeJeune et al. (2019), showing a strong correlation between smoothness and generalization for large models.

5 Conclusions

In this work we present an empirical study of the Lipschitz constant of ReLU networks in hypothesis space (model-wise) as well as throughout training. The Lipschitz constant, largely assumed to be uniformly upper bounded in theoretical works, is behaving non-monotonically in model size, highlighting a theoretical gap in the current understanding of model complexity. Importantly, smoothness alone is neither sufficient nor necessary to ensure generalization, and understanding the emergence of smoothness in interpolating solutions for the hypothesis class of neural networks is still an open question. Our findings highlight that the (model-wise) behaviour observed in hypothesis space is tightly related to the dynamics of SGD, with only large models achieving generalization and bounded Lipschitz. A central question is therefore understanding the ability of SGD to recover smooth solutions among those expressible by a given model class, in relationship to model size. We hope our findings will inspire the development of tighter theory of deep networks, accounting for the non-monotonic behaviour of smoothness of overparameterized models.

Acknowledgments and Disclosure of Funding

This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation, as well as partially funded by Swedish Research Council project 2017-04609. Large-scale experiments were enabled by resources provided by the Swedish National Infrastructure for Computing (SNIC) at Chalmers Centre for Computational Science and Engineering (C3SE) partially funded by the Swedish Research Council through grant agreement no. 2018-05973, as well as by resources provided by the National Supercomputer Centre (NSC), funded by Linköping University.

References

Sanjeev Arora, Nadav Cohen, and Elad Hazan. On the optimization of deep networks: Implicit acceleration by overparameterization. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 244–253. PMLR, 10–15 Jul 2018.

Peter L Bartlett, Dylan J Foster, and Matus J Telgarsky. Spectrally-normalized margin bounds for neural networks. *Advances in neural information processing systems*, 30, 2017.

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.

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

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, 2022a.

Matteo Gamba, Erik Englesson, Mårten Björkman, and Hossein Azizpour. Deep double descent via smooth interpolation. *arXiv preprint arXiv:2209.10080*, 2022b.

Boris Hanin and David Rolnick. Complexity of linear regions in deep networks. In *International Conference on Machine Learning*, pp. 2596–2604, 2019a.

Boris Hanin and David Rolnick. Deep relu networks have surprisingly few activation patterns. In *Advances in Neural Information Processing Systems*, pp. 359–368, 2019b.

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.

Yiding Jiang, Dilip Krishnan, Hossein Mobahi, and Samy Bengio. Predicting the generalization gap in deep networks with margin distributions. In *International Conference on Learning Representations*, 2019.

Kenji Kawaguchi, Leslie Pack Kaelbling, and Yoshua Bengio. Generalization in deep learning. *arXiv preprint arXiv:1710.05468*, 2017.

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.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, 2009.

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

Chao Ma and Lexing Ying. On linear stability of sgd and input-smoothness of neural networks. *Advances in Neural Information Processing Systems*, 34:16805–16817, 2021.

Guido F Montufar, Razvan Pascanu, Kyunghyun Cho, and Yoshua Bengio. On the number of linear regions of deep neural networks. In *Advances in Neural Information Processing Systems*, pp. 2924–2932, 2014.

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.

Vaishnavh Nagarajan and Zico Kolter. Deterministic pac-bayesian generalization bounds for deep networks via generalizing noise-resilience. In *International Conference on Learning Representations*, 2018.

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.

Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. Norm-based capacity control in neural networks. In *Conference on Learning Theory*, pp. 1376–1401. PMLR, 2015a.

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*, 2015b.

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.

Maithra Raghu, Ben Poole, Jon Kleinberg, Surya Ganguli, and Jascha Sohl-Dickstein. On the expressive power of deep neural networks. In *International Conference on Machine Learning*, pp. 2847–2854, 2017.

Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 5301–5310. PMLR, 09–15 Jun 2019.

Kevin Roth, Yannic Kilcher, and Thomas Hofmann. Adversarial training is a form of data-dependent operator norm regularization. *Advances in Neural Information Processing Systems*, 33:14973–14985, 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.

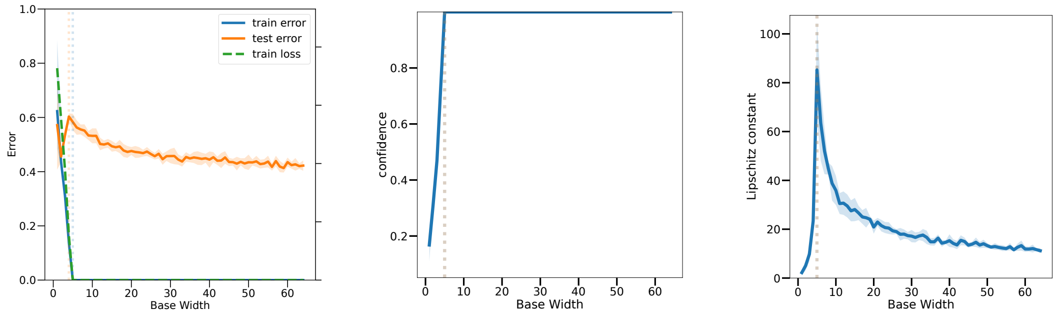
Colin Wei and Tengyu Ma. Data-dependent sample complexity of deep neural networks via lipschitz augmentation. *Advances in Neural Information Processing Systems*, 32, 2019.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *International Conference on Learning Representations*, 2018.

Chiyuan Zhang, Samy Bengio, and Yoram Singer. Are all layers created equal? *ICML Workshop Deep Phenomena*, 2019.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [No]
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]



(a) (b) (c)

Figure 4: (a) Double descent curve for the test error for a family of ResNet18s of increasing model size, trained on CIFAR-10 with 20% corrupted train labels. (b) Average prediction confidence over the training set, for the same networks. Confidence monotonically increases with model size, with models past the interpolation threshold matching the targets with high confidence. (c) Empirical Lipschitz constant of the network function, peaking at the interpolation threshold, and then decreasing for increasing model size. All results are averaged over 5 random seeds.

A Appendix

Section B fully details our experimental setup. Section C presents a standard algorithm for estimating the operator norm, while section D describes the distributions used for generating random validation data for Figures 3a and 5a. Section E presents double descent curves for the test error and the operator norm for ResNet18. Finally, section F presents epoch-wise trends for a family of ConvNets.

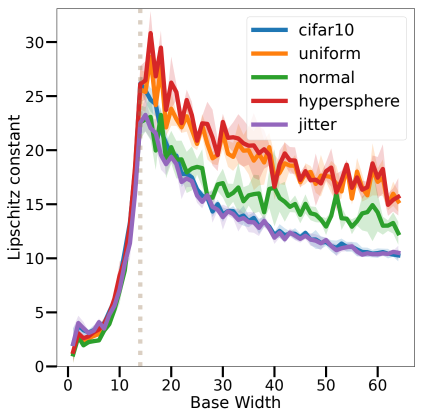
B Experimental Setup

Following Nakkiran et al. (2019) we train a family of ConvNets composed of 4 convolutional stages, each composed of a `[Conv, ReLU]` block, followed by maxpooling with stride 2, and 1 final linear classification layer. We also train a family of ResNet18s (He et al., 2015) without batch normalization layers. Both network architectures are composed of 4 convolutional stages, in which the spatial dimensions are reduced by factor of 2 and the number of learned feature maps doubles. More precisely, the convolutional stages respectively follow the progression $]w, 2w, 4w, 8w]$, where w is the base width of the network. In our experiments, we vary the base width in the range $w = 1, \dots, 64$. To tune hyperparameters, we take a random validation split of size 1000 from the CIFAR-10 (resp. CIFAR-100) training set. We train all networks with SGD with momentum 0.9, batch size 128, a fixed learning rate, set at $\mu = 5\text{e} - 3$ for the ConvNets and $\mu = 1\text{e} - 4$ for the ResNets. We train the ConvNets for 500 epochs, and the ResNets for 4000. To stabilize prolonged training, we use learning rate warmup over the first 5 epochs of training, starting from a learning rate $\mu_0 = 10^{-1} \times \mu$.

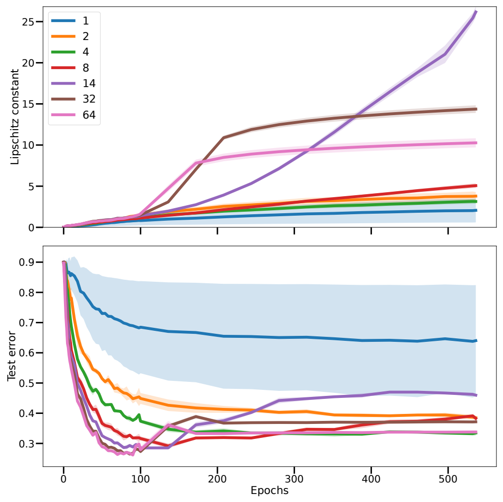
Our codebase is implemented in Pytorch 1.11, running on a local cluster equipped with NVIDIA A100 GPUs with 40GB onboard memory. We use 3 random seeds for the ConvNets and 5 for the ResNets, controlling network initialization and the shuffling and sampling of mini-batches from the training set. We use a dedicated random seed for generating the validation split used for hyperparameter tuning, fixed for all networks, as well as a fixed seed for corrupting the CIFAR training labels.

C Operator norm estimation

To estimate the operator norm of the Jacobian $\nabla_{\mathbf{x}} \mathbf{f} = \mathbf{W}_\epsilon$ evaluated at a training point, we use a power iteration algorithm. Starting at iteration $t = 0$ with randomly initialized vectors $\tilde{\mathbf{u}}_0 \in \mathbb{R}^K$, $\tilde{\mathbf{v}}_0 \in \mathbb{R}^d$, with corresponding normalized vectors $\mathbf{u}_0 = \tilde{\mathbf{u}}_0 / \|\tilde{\mathbf{u}}_0\|_q$, $\mathbf{v}_0 = \tilde{\mathbf{v}}_0 / \|\tilde{\mathbf{v}}_0\|_p$, at step t we compute $\tilde{\mathbf{u}}_t \leftarrow \nabla_{\mathbf{x}} \mathbf{f}\ \mathbf{v}_{t-1}$, $\tilde{\mathbf{v}}_t \leftarrow \mathbf{u}_t^T \nabla_{\mathbf{x}} \mathbf{f}$, $\sigma_t \leftarrow \mathbf{u}_t^T \nabla_{\mathbf{x}} \mathbf{f}\ \mathbf{v}_t$, with σ_t storing the largest singular value at convergence. The algorithm is iterated until $|\sigma_t - \sigma_{t-1}| \leq \text{tol}$, with relative tolerance $\text{tol} = 1\text{e} - 6$.



(a)



(b)

Figure 5: (a) ConvNets trained on CIFAR-10 with 20% corrupted labels evaluated using random test data. The results suggest that the non-monotonic trend of the empirical Lipschitz constant might be a global property of the network function. Interestingly, the Lipschitz constant is lower for data lying closer to the training data. (b) Development over epochs of the Lipschitz constant (top) and the test error (bottom) for several model sizes. Underfitting models (low base width) maintain a low Lipschitz throughout training, while models harmfully overfitting the training data keep increasing the Lipschitz constant. Finally, large overparameterized generalizing models undergo a epoch-wise double descent in the test error, associated with plateauing Lipschitz constant for larger models, which is smallest for the largest models. All results averaged over 3 random seeds.)

D Generating Random Validation Data

To generate random validation data for the experiments reported in Figures 3a and 5a, we define several distributions over RGB pixels, and sample each pixel independently. We consider the following distributions:

- $\mathbf{x}_n \sim \mathcal{U}([\boldsymbol{\mu}_{\text{CIFAR}} - \boldsymbol{\sigma}_{\text{CIFAR}}, \boldsymbol{\mu}_{\text{CIFAR}} + \boldsymbol{\sigma}_{\text{CIFAR}}])$ pixel-wise
- $\mathbf{x}_n \sim \mathcal{N}([\boldsymbol{\mu}_{\text{CIFAR}}, \mathcal{I}_3 \boldsymbol{\sigma}_{\text{CIFAR}}])$ pixel-wise
- $\mathbf{x}_n \sim \mathcal{U}(S_{d-1})$ (pixel-wise) hypersphere
- $\mathbf{x}_n + \boldsymbol{\epsilon}_n$, with $\boldsymbol{\epsilon}$ strong random jitter

where $\boldsymbol{\mu}_{\text{CIFAR}}$ and $\boldsymbol{\sigma}_{\text{CIFAR}}$ respectively denote the per-channel mean and standard deviation computed on the CIFAR-10 training set. For each distribution, we generate a validation set of 50k i.i.d. samples, and probe networks trained on CIFAR-10 with 20% corrupted labels.

E Additional Results for ResNet

Figure 4a reports the double descent curve for the test error for the family of ResNets considered, the average confidence computed on the training set (Figure 4b), as well as the empirical Lipschitz constant (Figure 4c), matching the trends described in section 3 for the ConvNets.

F Additional Results for ConvNets

Figure 5a presents non-monotonic trends for the Lipschitz constant for ConvNets trained on CIFAR-10 with 20% corrupted training labels, and evaluated using random test data. We observe how random data lying closely to the support of the training data distribution, namely CIFAR-10 training images with random jitter, closely follow the Lipschitz constant estimated on clean CIFAR-10 data, while other validation sets present a higher value for the constant, but the same non-monotonic trend. We believe the distance between curves – larger than that observed for ResNets in Figure 3a – originates from the lower training budget (500 epochs vs 4k).