

# The Geometric Occam's Razor Implicit in Deep Learning

**Benoit Dherin\***

*Google, Dublin*

DHERIN@GOOGLE.COM

**Michael Munn\***

*Google, New York*

MUNN@GOOGLE.COM

**David G.T. Barrett**

*DeepMind, London*

BARRETTDAVID@DEEPMIND.COM

## Abstract

In over-parameterized deep neural networks there can be many possible parameter configurations that fit the training data exactly. However, the properties of these interpolating solutions are poorly understood. We argue that over-parameterized neural networks trained with stochastic gradient descent are subject to a Geometric Occam's Razor; that is, these networks are implicitly regularized by the geometric model complexity. For one-dimensional regression, the geometric model complexity is simply given by the arc length of the function. For higher-dimensional settings, the geometric model complexity depends on the Dirichlet energy of the function. We explore the relationship between this Geometric Occam's Razor, the Dirichlet energy and other known forms of implicit regularization. Finally, for ResNets trained on CIFAR-10, we observe that Dirichlet energy measurements are consistent with the action of this implicit Geometric Occam's Razor.

## 1. Introduction

Naively, we might expect that over-parameterized models will overfit the training data and that under-parameterized models will be better since they have fewer degrees of freedom. However, it turns out that over-parameterized models can find better solutions than the under-parameterized models - a paradoxical phenomenon known as the double-descent curve [6, 25]. One possible explanation for this behaviour is that over-parameterized models are subject to an Occam's razor that filters out unnecessarily complex solutions in favour of simpler solutions.

Typically we might expect an Occam's razor to take the form of a complexity measure on the number of model parameters or the size of the hypothesis space, for instance [34]. However, for neural networks, the precise form of this hypothesized Occam's razor is not known, since it is not explicitly enforced during training. There has been some progress recently to identify sources of implicit regularization that may play a role here [5, 8, 20, 32]. For instance, recent work has exposed a hidden form of regularization in Stochastic Gradient Descent (SGD) called Implicit Gradient Regularization (IGR) [5, 32] which penalizes learning trajectories that have large loss gradients.

For over-parameterized neural networks trained with SGD, we hypothesize that the hidden Occam's razor takes the form of a geometric complexity measure. Our key contributions are as follows: (1) define this notion of geometric complexity; (2) show that the Dirichlet energy can be used as a proxy for geometric complexity; (3) show that the IGR mechanism from SGD puts a regularization

---

\* equal contribution

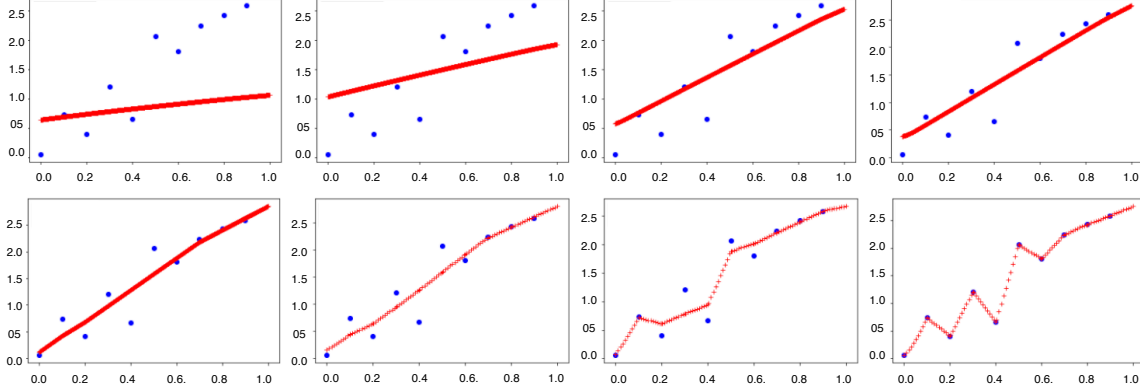


Figure 1: Training sequence for an over-parameterised neural network trained using 10 data points plotted at iteration step 1, 20, 40, 50, 5000, 10000, 20000, and 30000 (from top left to bottom right, respectively). The blue circles represent the data points while the red crosses represent the points predicted by the neural network.

pressure on the geometric complexity; and (4) show that the strength of this pressure increases with the size of the learning rate, which we verify with numerical experiments.

## 2. The Geometric Occam’s razor in 1-dimensional regression

To build intuition, we begin with a simple 1-dimensional example. Consider a ReLU neural network consisting of 3 layers with 300 units per layer, trained using SGD, without any form of explicit regularization to perform 1-dimensional regression using only 10 data points. In this extreme setting, we should expect the network to overfit the dataset, since the function space described by that neural network is extremely large - consisting of piecewise linear functions with thousands of linear pieces [2]. Yet, if we plot the learned function during training from the first step all the way up to interpolation, as in Figure 1, we observe that the learned function is the ‘simplest’ possible function, in some sense, among all functions with the same training error.

But what do we mean by ‘simple’? Our key intuition in this example is that the arc length of the learned function over the smallest interval containing the data points provides our measure of model complexity. At the end of training, we see that the arc length of the learned function is close to that of the shortest possible path interpolating between the data points, suggesting that this measure of geometric complexity is somehow optimised during training.

### 3. Dirichlet energy as a measure of function complexity

In the previous section, we used the arc length of the learnt 1-dimensional function as a measure of its geometric complexity. What is the corresponding notion for a function in a high-dimensional feature space  $f : \mathbb{R}^d \rightarrow \mathbb{R}$ ? In this case, we can define the *geometric complexity* of a function  $f$  as the volume  $\Omega_D(f)$  of its graph

$$\text{gr}(f_{X_D}) := \{(x, f(x)) : x \in X_D\} \subset \mathbb{R}^d \times \mathbb{R} \quad (1)$$

restricted to the feature polytope  $X_D$ ; that is, the polytope with smallest volume containing all the feature points  $x_i$  of the dataset  $D = \{(x_i, y_i) : i = 1, \dots, n\}$ . From differential geometry [9], for a smooth function  $f : \mathbb{R}^d \rightarrow \mathbb{R}$ , the graph  $\text{gr}(f_{X_D})$  is an  $n$ -dimensional smooth submanifold of  $\mathbb{R}^{n+1}$ . Using the Riemannian metric on  $\text{gr}(f_{X_D})$  induced from the Euclidean metric on  $\mathbb{R}^{n+1}$  and its corresponding Riemannian volume form, the volume of the graph of  $f$  can be expressed as

$$\Omega_D(f) = \int_{X_D} \sqrt{1 + \|\nabla_x f\|^2} dx. \quad (2)$$

This can in turn be approximated using a first-order Taylor series expansion  $\sqrt{1+z} \simeq 1 + \frac{1}{2}z$  so that

$$\Omega_D(f) \approx \int_{X_D} 1 + \frac{1}{2} \|\nabla_x f\|^2 dx = \text{Vol}(X_D) + \text{DE}(f), \quad (3)$$

where

$$\text{DE}(f) = \frac{1}{2} \int_{X_D} \|\nabla_x f\|^2 dx. \quad (4)$$

is the *Dirichlet energy* of the function  $f$  over  $X_D$ . The computation above suggests that both the function volume or its Dirichlet energy can be used as a measure of a function’s geometric complexity.

One way to compute the Dirichlet energy numerically is to use a quadrature formula summing up  $\|\nabla_x f\|^2$  over a number of points in  $X_D$  and multiplying the summands by the volume element of the point. So, if we use the data points themselves for evaluation and  $1/|D|$  as a proxy for the volume element, we obtain a discrete version of the Dirichlet energy which we call the *discrete Dirichlet energy*, denoted by  $\widehat{\text{DE}}(f)$ . This provides an easily computable measure of a function’s geometric complexity:

$$\widehat{\text{DE}}(f) = \frac{1}{2|D|} \sum_{x \in D} \|\nabla_x f(x)\|^2. \quad (5)$$

### 4. How neural networks tame model complexity

We now argue that the geometric complexity, as measured by the discrete Dirichlet energy, is implicitly regularized during the training of neural nets with vanilla SGD.

In recent work [32], it was shown that the discrete steps of SGD from epoch to epoch closely follow, on average, the gradient flow of a modified loss of the form:

$$\tilde{L} = L + \frac{h}{4|D|} \sum_{(x,y) \in D} \|\nabla_{\theta} L(x, \theta)\|^2,$$

where  $L$  is the original loss and  $L(x, \theta) := E(f_\theta(x), y)$  is the error  $E$  between the prediction  $f_\theta(x)$  and the true label  $y$ . This means that during SGD the quantities  $\|\nabla_\theta L(x, \theta)\|^2$  at each data point  $(x, y) \in D$ , are implicitly regularized, with the learning rate  $h$  acting as an implicit regularization rate.

Now, for models whose losses come from the application of a maximum likelihood estimation on a conditional probability distribution in the exponential family such as the least-square loss or the cross-entropy loss, we obtain loss gradients that have the following form:

$$\nabla_\theta L(x, \theta) = \epsilon_x(\theta) \nabla_\theta f_\theta(x),$$

where  $\epsilon_x(\theta) = (f_\theta(x) - y)$  is the signed residual, yielding

$$\tilde{L} = L + \frac{h}{4|D|} \sum_{(x,y) \in D} \epsilon_x(\theta)^2 \|\nabla_\theta f_\theta(x)\|^2. \quad (6)$$

From that last expression for  $\tilde{L}$ , we see that the terms  $\|\nabla_\theta f_\theta(x)\|^2$  are implicitly regularized at each data point  $(x, y)$  and even more so in the region where the residual errors are large, such as the beginning of training.

We now argue that for *neural networks*, in particular, the regularization pressure on the gradient of the network with respect to the parameters  $\|\nabla_\theta f_\theta(x)\|^2$  acts as a regularization pressure on the gradient of the network with respect to the input  $\|\nabla_x f_\theta(x)\|^2$ . Hence, this creates a pressure for the Dirichlet energy to be implicitly regularized during training. In fact, this follows from the fact that for neural networks their derivatives with respect to the inputs and the parameters can be related as follows (proof in Appendix A):

**Theorem 1** *Consider a neural network with  $l$  layers  $f_\theta(x) = f_1 \circ \dots \circ f_l(x)$ , where  $f_i(z) = a_i(w_i z + b_i)$  with  $\theta = (w_1, b_1, \dots, w_l, b_l)$  being the vector of layer weight matrices  $w_i$  and biases  $b_i$  and the  $a_i$ ’s are the layer activation functions. Then we have that*

$$\|\nabla_x f_\theta(x)\|^2 \left( \frac{1 + \|h_1(x)\|^2}{\|w_1\|^2 \|h'_1(x)\|^2} + \dots + \frac{1 + \|h_l(x)\|^2}{\|w_l\|^2 \|h'_l(x)\|^2} \right) \leq \|\nabla_\theta f_\theta(x)\|^2, \quad (7)$$

where  $h_i(x)$  is the sub-network from input  $x$  to layer  $i$  and  $\|w_i\|$  is the spectral norm of the weight matrix  $w_i$ .

From (7), we see that the regularization pressure from IGR translates into a regularization pressure on the discrete Dirichlet energy when the positive quantities

$$A_i(\theta, x) := \|w_i\| \|h'_i(x)\|, \quad i = 1, \dots, l \quad (8)$$

remain small. Note that this is expected to happen at the beginning of training when the spectral norms of the layers are close to zero, while they tend to grow as the training progresses if no spectral regularization [23] is applied. Furthermore, note here that preventing the  $A_i$ ’s from becoming too large during training may be an important consideration which informs the choice of model architecture and layer regularization.

**Experimental evidence:** From Equation (6), since the strength of IGR is a function of the learning rate, we should expect an increased pressure on the Dirichlet energy as a result of Equation

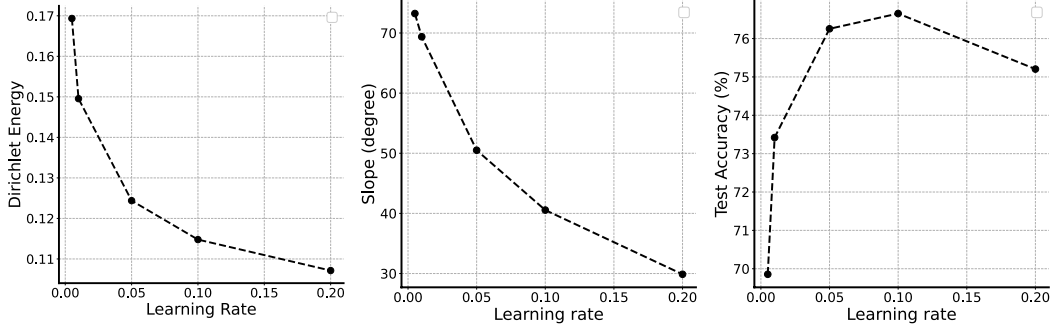


Figure 2: For a ResNet-18 trained on CIFAR-10, we observe that the Dirichlet energy and the loss-surface slope (essentially the IGR term  $\|\nabla_{\theta} L\|^2$ ) decrease as the learning rate increases. Each dot corresponds to a fully trained model at time of maximal test accuracy.

(7) when training with higher learning rates. We verify this prediction for a ResNet-18 trained to classify CIFAR-10 images. Measuring the discrete Dirichlet energy at the time of maximal test accuracy for a range of learning rates, we observe this predicted behaviour, consistent with our theory; see Figure 2.

**Remark 2** Note also that for linear models (i.e., neural networks with a single linear layer), the Dirichlet energy of the network coincides in this case with the L2-norm of the parameters. Therefore, this results recovers the already known fact that linear models trained with SGD have an inductive bias towards low L2-norm solutions (see [34]). This also points toward the fact that the Dirichlet energy may be the right generalization of the L2-norm for a general network.

## 5. Related work, Future directions, and Discussion

**Splines and connections to harmonic function theory:** The Dirichlet energy (4) is well-known in harmonic function theory [3] where it can be shown using calculus of variations that harmonic functions subject to a boundary condition minimize the Dirichlet energy over the space of differentiable functions. This is known as Dirichlet’s principle. The minimization of the Dirichlet energy itself is also related to the theory of splines [17]. Our work seems to indicate that neural networks are biased towards (a notion of) harmonic functions with the dataset acting as the boundary condition. **Complexity theory:** The notion of geometric complexity introduced has similarities to the Kolmogorov complexity [30] as well as the minimum description length given in [15]. **Smoothness regularization:** The notion of geometric complexity considered here is related to the notion of smoothness with respect to the input as discussed in [27] as well as to the Sobolev regularization effect of SGD discussed in [20], where inequalities similar to (7) but involving only the first layer are considered. In particular, various forms of gradient penalties, reminiscent of the Dirichlet energy, have been devised to achieve Lipschitz smoothness [1, 10, 11, 14, 18]. It has been shown that the discrete

Dirichlet energy (evaluated at the data points) is a powerful regularizer [16, 33] and in [27] that it has advantages over other form of smoothness regularization (such as spectral norm regularization [19, 23]). Our analysis shows that we can control this form of regularization cheaply through the learning rate. In image processing, the Dirichlet energy is also called the Rudin–Osher–Fatemi total variation and it as been introduced as a powerful explicit regularizer for image denoising; see [29] and [13]. It may be possible that these various forms of smoothness regularization are useful because they provide implicit control over the model geometric complexity. **Regularization through noise:** The discrete Dirichlet energy is reminiscent of the Tikhonov regularizer which is implicitly regularized with added input noise [7]. The modified loss in (6) is also very reminiscent of the modified loss in [8], which is argued to be implicitly minimized by SGD when a random white noise is added to the labels. In Section 3 of [22], it is argued that explicit gradient regularization with respect to input and noise instance produce similar types of regularization. Altogether, this suggests that feature noise, label noise, and the optimization scheme all conspire to implicitly tame the geometric complexity in the case of neural networks trained with gradient-based optimization schemes. **Regularization through the number of layers:** In Equation (7), one sees that each layer contributes an additional positive term, increasing the pressure on the Dirichlet energy. This suggests that the pressure on the model geometric complexity may increase with the neural network depth in a similar spirit as [12]. **Training of GANs:** For GANs, explicit gradient regularization both with respect to the input [1, 11, 14, 18, 23] and the parameters [4, 21, 24, 26, 28] has been proven to be beneficial and related to smoothness. Our main theorem provides a way to relate gradient penalties with respect to the input and with respect to the parameters for neural networks, in a way where the spectral norm of the weight matrices plays a key role. This points toward geometric complexity being a useful notion to relate and understand these different forms of regularization (including spectral normalization as in [23] and [19]).

## 6. Conclusion

In conclusion, we have found that neural networks trained with SGD are subject to an implicit Geometric Occam’s razor, which selects parameter configurations that have low geometric complexity ahead of configurations with high geometric complexity. This geometric complexity is given by the arc length in 1-dimensinal regression; is linearly related to the Dirichlet energy in higher-dimensional settings; and has many intriguing similarities to other known quantities, including various forms of implicit and explicit regularisation and model complexity. More generally, our work develops promising new theoretical connections between optimization and the geometry of over-parameterised neural networks.

## ACKNOWLEDGMENTS

We would like to thank Mihaela Rosca, Maxim Neumann, Yan Wu, Samuel Smith, and Soham De for helpful discussion and feedback. We would also like to thank Patrick Cole and Shakir Mohamed for their support.

## References

- [1] Michael Arbel, Danica J Sutherland, Mikolaj Binkowski, and Arthur Gretton. On gradient regularizers for mmd gans. In *Advances in Neural Information Processing Systems*, volume 31,

2018.

- [2] Raman Arora, Amitabh Basu, Poorya Mianjy, and Anirbit Mukherjee. Understanding deep neural networks with rectified linear units. In *International Conference on Learning Representations*, 2018.
- [3] Sheldon Axler, Paul Bourdon, and Ramey Wade. *Harmonic Function Theory*, volume 137. Springer, 2013.
- [4] David Balduzzi, Sebastien Racaniere, James Martens, Jakob Foerster, Karl Tuyls, and Thore Graepel. The mechanics of n-player differentiable games. In *International Conference on Machine Learning*, 2018.
- [5] David G.T. Barrett and Benoit Dherin. Implicit gradient regularization. In *International Conference on Learning Representations*, 2021.
- [6] Mikhail Belkin. Fit without fear: remarkable mathematical phenomena of deep learning through the prism of interpolation. 2021. URL <https://arxiv.org/abs/2105.14368>.
- [7] Chris M. Bishop. Training with noise is equivalent to tikhonov regularization. *Neural Computation*, 7(1), 1995.
- [8] Guy Blanc, Neha Gupta, Gregory Valiant, and Paul Valiant. Implicit regularization for deep neural networks driven by an ornstein-uhlenbeck like process. In *Conference on Learning Theory*, 2020.
- [9] Manfredo P. do Carmo. *Differential geometry of curves and surfaces*. Prentice Hall, 1976.
- [10] Gamaleldin Elsayed, Dilip Krishnan, Hossein Mobahi, Kevin Regan, and Samy Bengio. Large margin deep networks for classification. In *Advances in Neural Information Processing Systems*, volume 31, 2018.
- [11] William Fedus, Mihaela Rosca, Balaji Lakshminarayanan, Andrew M. Dai, Shakir Mohamed, and Ian J. Goodfellow. Many paths to equilibrium: Gans do not need to decrease a divergence at every step. In *International Conference on Learning Representations*, 2018.
- [12] Tianxiang Gao and Vladimir Jojic. Degrees of freedom in deep neural networks. In *Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence*, 2016.
- [13] Pascal Getreuer. Rudin-Osher-Fatemi Total Variation Denoising using Split Bregman. *Image Processing On Line*, 2012.
- [14] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in neural information processing systems*, 2017.
- [15] Geoffrey E Hinton and Richard Zemel. Autoencoders, minimum description length and helmholtz free energy. In J. Cowan, G. Tesauro, and J. Alspector, editors, *Advances in Neural Information Processing Systems*, volume 6. Morgan-Kaufmann, 1994.



- [16] Judy Hoffman, Daniel A. Roberts, and Sho Yaida. Robust learning with jacobian regularization. 2020. URL <https://arxiv.org/abs/1908.02729>.
- [17] Hui Jin and Guido Montufar. Implicit bias of gradient descent for mean squared error regression with wide neural networks. 2021. URL <https://arxiv.org/abs/2006.07356>.
- [18] Naveen Kodali, James Hays, Jacob Abernethy, and Zsolt Kira. On convergence and stability of GANs. In *Advances in Neural Information Processing Systems*, 2018.
- [19] Zinan Lin, Vyas Sekar, and Giulia C. Fanti. Why spectral normalization stabilizes gans: Analysis and improvements. In *Advances in neural information processing systems*, 2021.
- [20] Chao Ma and Lexing Ying. The sobolev regularization effect of stochastic gradient descent. 2021. URL <https://arxiv.org/abs/2105.13462>.
- [21] Lars Mescheder, Sebastian Nowozin, and Andreas Geiger. The numerics of gans. In *Advances in Neural Information Processing Systems*, 2017.
- [22] Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for GANs do actually converge? In *Proceedings of the 35th International Conference on Machine Learning*, 2018.
- [23] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*, 2018.
- [24] Vaishnavh Nagarajan and J Zico Kolter. Gradient descent gan optimization is locally stable. In *Advances in neural information processing systems*, 2017.
- [25] 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*, 2020.
- [26] Chongli Qin, Yan Wu, Jost Tobias Springenberg, Andy Brock, Jeff Donahue, Timothy Lillicrap, and Pushmeet Kohli. Training generative adversarial networks by solving ordinary differential equations. In *Advances in Neural Information Processing Systems*, volume 33, 2020.
- [27] Mihaela Rosca, Theophane Weber, Arthur Gretton, and Shakir Mohamed. A case for new neural network smoothness constraints. In *Proceedings on "I Can't Believe It's Not Better!" at NeurIPS Workshops*, volume 137 of *Proceedings of Machine Learning Research*, 2020.
- [28] Mihaela Rosca, Yan Wu, Benoit Dherin, and David G.T. Barrett. Discretization drift in two-player games. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, 2021.
- [29] Leonid Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1), 1992.



- [30] J. Schmidhuber. Discovering neural nets with low kolmogorov complexity and high generalization capability. *Neural networks : the official journal of the International Neural Network Society*, 10 5, 1997.
- [31] Sihyeon Seong, Yegang Lee, Youngwook Kee, Dongyoon Han, and Junmo Kim. Towards flatter loss surface via nonmonotonic learning rate scheduling. In *Proceedings of the Thirty-Fourth Conference on Uncertainty in Artificial Intelligence*, 2018.
- [32] Samuel L Smith, Benoît Dherin, David G.T. Barrett, and Soham De. On the origin of implicit regularization in stochastic gradient descent. In *International Conference on Learning Representations*, 2021.
- [33] Dániel Varga, Adrián Csizsárik, and Zsolt Zombori. Gradient regularization improves accuracy of discriminative models. 2018. URL <https://arxiv.org/abs/1712.09936>.
- [34] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations*, 2017.

## Appendix A. Proof of Theorem 1

Consider a neural network with  $l$  layers  $f_\theta(x) = f_1 \circ \dots \circ f_l(x)$ , where  $f_i(z) = a_i(w_i z + b_i)$  with  $\theta = (w_1, b_1, \dots, w_l, b_l)$  being the vector of layer weight matrices  $w_i$  and biases  $b_i$  and the  $a_i$ 's are the layer activation functions. We will use the notation  $f_{w_i}$  or  $f_{b_i}$  instead of  $f_\theta$  when we want to consider  $f_\theta$  as dependent on the  $w_i$ 's or  $b_i$ 's only.

For this model structure and following Pythagoras, we have:

$$\|\nabla_\theta f_\theta(x)\|^2 = \|\nabla_{w_1} f_\theta(x)\|^2 + \|\nabla_{b_1} f_\theta(x)\|^2 + \dots + \|\nabla_{w_l} f_\theta(x)\|^2 + \|\nabla_{b_l} f_\theta(x)\|^2 \quad (9)$$

For each layer  $i$ , we can rewrite the network function as

$$f_{w_i}(x) = g_i(w_i h_i(x) + b_i),$$

where  $g_i(z)$  consists of the deeper layers above  $i$  and  $h_i(x)$  consists of the shallower layers below  $i$ . The idea now, inspired from [31], is to show that a small perturbation  $\delta x$  of the input is equivalent to a small perturbation  $u(\delta x)$  of the weights of layer  $i$ . We will use this idea to prove the following two lemmas.

**Lemma 3** *In the notation above, for each layer  $i$ , we have:*

$$\|\nabla_x f_\theta(x)\|^2 \left( \frac{\|h_i(x)\|}{\|w_i\| \|h'_i(x)\|} \right)^2 \leq \|\nabla_{w_i} f_\theta(x)\|^2. \quad (10)$$

**Proof** Consider a small perturbation  $x + \delta x$  of the input  $x$ . We start by showing that we can always find a corresponding perturbation  $w_i + u(\delta x)$  of the weight matrix in layer  $i$  such that

$$f_{w_i}(x + \delta x) = f_{w_i + u(\delta x)}(x). \quad (11)$$

Namely, because  $f_{w_i}(x) = g_i(w_i h_i(x) + b_i)$ , to show this, it is enough to find  $u(\delta x)$  such that

$$w_i(h_i(x) + h'_i(x)\delta x) + b_i = (w_i + u(\delta x))h_i(x) + b_i, \quad (12)$$

where we identify  $h_i(x + \delta x)$  with its linear approximation around  $x$  for small  $\delta x$ . Then (12) is always satisfied if we set

$$u(\delta x) := \frac{(w_i h'_i(x)\delta x) h_i^T(x)}{\|h_i(x)\|^2}, \quad (13)$$

since  $h_i^T(x)h_i(x) = \|h_i(x)\|^2$ . Now taking the derivative with respect to  $\delta x$  at  $\delta x = 0$  on both sides of Equation (16), and using the chain rule and that  $u(\delta x)$  is linear in  $\delta x$ , we obtain a relation between the network derivative with respect to the weight matrices and w.r.t the input:

$$\nabla_x f_\theta(x) = \nabla_{w_i} f_\theta(x) \frac{(w_i h'_i(x)) h_i^T(x)}{\|h_i(x)\|^2}. \quad (14)$$

Taking the norm on both sides, squaring, and rearranging the terms yields (10). ■

Following the same strategy, we now prove a corresponding lemma for the biases at each layer:

**Lemma 4** *In the notation above, for each layer  $i$ , we have:*

$$\|\nabla_x f_\theta(x)\|^2 \left( \frac{1}{\|w_i\| \|h'_i(x)\|} \right)^2 \leq \|\nabla_{b_i} f(x)\|^2. \quad (15)$$

**Proof** Consider a small perturbation  $x + \delta x$  of the input  $x$ . We start again by showing that we can always find a corresponding perturbation  $b_i + u(\delta x)$  of the biases in layer  $i$  such that

$$f_{b_i}(x + \delta x) = f_{b_i + u(\delta x)}(x). \quad (16)$$

Namely, because  $f_{b_i}(x) = g_i(w_i h_i(x) + b_i)$ , to show this, it is enough to find  $u(\delta x)$  such that

$$w_i(h_i(x) + h'_i(x)\delta x) + b_i = w_i h_i(x) + b_i + u(\delta x), \quad (17)$$

where we again identify  $h_i(x + \delta x)$  with its linear approximation around  $x$  for small  $\delta x$ . Then (17) is always satisfied if we set this time

$$u(\delta x) := w_i h'_i(x) \delta x. \quad (18)$$

Now taking the derivative with respect to  $\delta x$  at  $\delta x = 0$  on both sides of Equation (16), and using the chain rule and that  $u(\delta x)$  is linear in  $\delta x$ , we obtain a relation between the network derivative w.r.t. the biases and w.r.t the input:

$$\nabla_x f_\theta(x) = \nabla_{w_i} f(x) w_i h'_i(x). \quad (19)$$

Taking the norm on both sides, squaring, and rearranging the terms yields (15). ■

Using this in quality for each layer in the decomposition given by Equation (9), we obtain that

$$\|\nabla_x f_\theta(x)\|^2 \left( \left( \frac{k_1(x)}{\|w_1\|} \right)^2 + \dots + \left( \frac{k_l(x)}{\|w_l\|} \right)^2 \right) \leq \|\nabla_\theta f(x)\|^2 \quad (20)$$

Now if we use (10) and (15) in (9), we finally obtain that

$$\|\nabla_x f_\theta(x)\|^2 \left( \frac{1 + \|h_1(x)\|^2}{\|w_1\|^2 \|h'_1(x)\|^2} + \dots + \frac{1 + \|h_l(x)\|^2}{\|w_l\|^2 \|h'_l(x)\|^2} \right) \leq \|\nabla_\theta f(x)\|^2. \quad (21)$$