Inductive Biases of SGD Training

Wes Gurnee
MIT
wesg@mit.edu

Abstract

In this brief survey, we outline the inductive biases of using stochastic gradient descent (SGD) to train deep neural network. Specifically, we focus on biases of SGD towards simple solutions which lie in broad loss basins, two aspects hypothesized to be responsible deep networks' remarkable generalization performance. Our exposition emphasises intuition and breadth over rigor and depth.

1 Introduction

The inductive biases of a deep learning training procedure are the set of factors which bias the process towards specific classes of solutions over others. Understanding these factors enables reasoning about the properties of the learned solutions, critical for advancing the science and safety of artificial intelligence.

We focus on biases hypothesized to be at the heart of deep networks impressive generalization capabilities, that is biases towards solutions which are simple and lie in flat regions of the loss landscape. We discuss several notions of simplicity, why we expect networks trained with SGD to exhibit these simplicity biases, and how simplicity changes over the course of training. We then review the sources of noise within stochastic gradient descent, how this noise biases training towards flat regions of parameter space, and why this implies a bias for data compression.

Finally, we note that the inductive biases of neural network training are complicated and emanate from a variety of sources, including the architecture, hyperparmaters, optimizer, and temporal dynamics. Throughout, we attempt to identify the primary source of the bias as relevant.

2 Simplicity

Motivated by arguments from Occam's Razor and the bias variance tradeoff, a common desiderata of a statistical learning algorithms is a bias towards simplicity. Despite violating classical formalizations of simplicity (like parameter count), deep networks trained via SGD are observed to generalize extremely well. This is hypothesized to be, in part, a consequence of a simplicity bias of SGD. While simplicity is an intuitive concept, it has proved challenging to mathematically operationalize in a sufficiently general manner.

2.1 Types of simplicity biases

Rank Bias One potential definition of complexity is the rank of the weight matrices of a model, given that rank defines the effective dimensionality of the output of a linear operator. [22] show that ReLU networks of sufficient depth are provably biased towards low-rank solutions when optimized with gradient flow (GF) methods. In a more general setting, [7] show that training a network with SGD with small batch size, will induce low rank weights even in networks with convolutional layers, residual connections, and batch normalization layers. Their analysis, based on the observation that $\partial f / \partial W_i$ is a matrix of rank ≤ 1, suggests that the SGD batch size bounds the rank of the network

weight matrices, that weight decay is a necessary condition for low rank bias, and that larger learning rates further biases the network to have smaller rank.

However, the extent to which this low rank bias is a property of the optimizer versus a property of neural networks is not clear. In particular, [10] observe that even at initialization, deep networks are biased towards low rank transformations. Furthermore, these biases continue to exist through training, when using either gradient or non-gradient based optimization (e.g., random search). They conjecture that rank regularization is primarily driven by network depth given how the volume of functions with low effective rank increases with depth and provide empirical evidence for this. The intuition is that the rank of a product of matrices is bounded above by the lowest rank matrix in the product, making it more likely that functions parameterized by the product of many matrices to be of decreasing rank with increasing depth.

Spectral Bias Another form of simplicity is the spectrum of learned features, with low frequency features being more simple than high frequency features. Using tools from Fourier analysis [18], show that deep networks prioritize learning low frequency features first before minimizing the residuals of more complex features, and that the simple features are more robust and generalizable. [24] add more nuance by studying the spectrum of the Conjugate Kernel (CK; what the network looks like at initialization) and the Neural Tangent Kernel (NTK; what the network looks like during and after training). In particular, they analyze the interplay between the hyperparamters and the frequency bias to show that: a) it is not universal (switching ReLU activations for sigmoids mostly negates the effect), b) deeper networks learn more complex features, but that there exists an optimal depth for which it can be detrimental to exceed, c) for complex features, training all layers together is better than just tuning the last layer, but vice versa for simple features, and d) there exists a maximal nondiverging learning rate.

Subsequent work describes how to further modulate this effect with architectural choices. For example [25] amplify, dampen, counterbalance, or reverse the intrinsic frequency biasing by replacing the loss function with a Sobolev norm and [9] show how how using the Hat function as the activation function removes the spectral bias.

Effective Depth Given the increase of expressively when adding more layers to a deep network, another potential measure of simplicity is the "effective depth" of the network. This notion is motivated by the phenomena of neural collapse (NC) where the representations of the second-to-last layer of a classification network cluster to their class means [16]. To formalize effective depth, [6] measure the first layer for which sample embeddings are separable using the nearest-class center classifier. They further hypothesize that SGD has an implicit bias for networks with smaller effective depth, and provide positive empirical evidence. This agrees with a natural intuition of SGD: it is easier to learn smaller circuits as it is difficult to get many layers to coordinate together, hence SGD should bias towards circuits of smaller intrinsic depth. Such an intuition is supported by evidence from [23], who show that residual networks behave like ensemble of shallow networks with short path lengths (i.e., small effective depth).

Geometric Complexity Recently, [5] proposed what is currently the most comprehensive measure: geometric complexity. Drawing from the theory of harmonic functions and minimal surfaces, they attempt measure complexity as the variability of the model function, computed using a discrete Dirichlet energy.

Definition (Geometric Complexity) [5]. Let $g_\theta : \mathbb{R}^d \to \mathbb{R}^k$ be a neural network parameterized by θ. We can write $g_\theta(x) = a\left(f_\theta(x)\right)$ where a denotes the last layer activation, and f_θ its logit network. The GC of the network 2 over a dataset D is defined to be the discrete Dirichlet energy of its logit network:

$$\langle f_\theta, D\rangle_G = \frac{1}{|D|}\sum_{x\in D}\left\|\nabla_x f_\theta(x)\right\|_F^2,$$

where $\left\|\nabla_x f_\theta(x)\right\|_F$ is the Frobenius norm of the network Jacobian.

This metric captures the basic intuition that, for simple functions, small changes in input (around the training data) should have a small change in output. For linear transformations defined by A the GC is dataset independent and exactly $\|A\|_F^2$. This coincides exactly with an l_2 norm penalty, hence l_2 regularization corresponds to regularizing the geometric complexity in the linear setting. For ReLU

networks, the GC over the whole dataset is simply the GC over a particular batch $B \subset D$. This is helpful as it enables measuring GC batchwise rather than on the whole dataset provided the batches are sufficiently large.

The authors go on to show how many popular training techniques explicitly or implicitly regularize the geometric complexity. This includes

1. Common initialization schemes: sampling from truncated normal Gaussians with variance inversely proportional to the number of input units has initial GC near 0, with distance to 0 being inversely proportional to network depth.

2. Common explicit regularization: increased l_2, spectral, and flatness regularization are all shown to empirically also decrease the GC.

3. Recently identified forms of SGD implicit regularization, such as step-size regularization [2], Sobolev regularization [12], and batch effect regularization [21] also implicitly regularize GC.

Again, geometric complexity highlights the complicated interplay between the inductive biases of the network, the optimization algorithm, and the training procedure.

Pitfalls of simplicity Finally, while a simplicity bias (SB) can have appealing statistical properties, this simplicity is not without its drawbacks. After defining simplicity in terms of the learned decision boundary, [20] make four observations (in their words):

> (i) SB of SGD and variants can be extreme: neural networks can exclusively rely on the simplest feature and remain invariant to all predictive complex features.
> (ii) The extreme aspect of SB could explain why seemingly benign distribution shifts and small adversarial perturbations significantly degrade model performance.
> (iii) Contrary to conventional wisdom, SB can also hurt generalization on the same data distribution, as SB persists even when the simplest feature has less predictive power than the more complex features.
> (iv) Common approaches to improve generalization and robustness—ensembles and adversarial training—can fail in mitigating SB and its pitfalls.

2.2 Training Dynamics

Part of what makes neural networks so challenging to study is that they cannot be understood as a fixed artifact, but rather the product of complex learning dynamics occurring during training. Therefore, an important aspect of understanding complexity, is understanding how it changes over the course of training. [15] show that in the initial epochs of training, almost all of the performance improvement of the classifier obtained by SGD can be explained by a linear classifier, but with additional iterations, SGD learns functions of increasing complexity. Importantly, they also that the linear classifier learned in the initial stages is retained throughout training, giving evidence to the hypothesis of implicit gradient boosting where the network learns an initial weak model, and then iteratively fits the residual errors with more complex functions.

Similar to learning functions of increasing complexity, [19] show that deep networks trained with SGD learn to model distributions of increasing complexity. In particular, they show that such networks "classify their inputs using lower-order input statistics, like mean and covariance, and exploit higher-order statistics only later during training."

Finally, [13] empirically relate the parameter norm of full scale transformer models to the dynamics of training with SGD. In addition to simply showing that parameters grow in magnitude, they prove that the network approximates a discretized network with "saturated" activation functions. A saturated network is a restricted network variant whose discretized representations are understandable in terms of formal languages and automata, potentially offering a more interpretable formalism to study large language models.

3 Broad Minima

Moving beyond simplicity, we describe SGD's bias for converging to extreme points in broad loss basins, how this is helpful for generalization, and why this implies an inductive bias towards data compression.

3.1 Breadth Bias

Broad optima, that is local minima where the loss hessian has many 0 or near zero eigenvalues, have long been hypothesized for explaining the generalization performance of neural nets [4, 3]. Perhaps the simplest intuition for this, is to consider the distribution shift between the training and test sets. For broader training loss basins, just as we can change the parameters more while preserving performance, it is likely we can change the data distribution more while preserving performance. Recently, [3] formalize this intuition with the notion of shift curvature, that is the amount of curvature of the loss surface in the direction between the train and test minima. Of course, this direction is unknown *a priori*, and therefore the best policy is to minimize the loss curvature in all directions, in other words, to find a broad basin. Many, including [3], have identified the noise induced by SGD, as the crucial factor that drives SGD's bias for broad minima.

Batch noise The basic intuition is that with minibatch SGD, any local minima for one batch is unlikely to be a local minima for another batch, unless it is an especially broad minima, that is, a minima which generalizes across batches. Therefore, if SGD converges, we should expect the optima to be broad, that is to generalize to all of the different batches (and likely also out of distribution). A simple experiment to test this intuition is to study the effect of batch size on generalization. [11] run this experiment and find "when using a larger batch there is a degradation in the quality of the model, as measured by its ability to generalize...present numerical evidence that supports the view that large-batch methods tend to converge to sharp minimizers of the training and testing functions... In contrast, small-batch methods consistently converge to flat minimizers."

[27] study the conditions for which this noise is helpful for escaping local minima (the Hessian being ill conditioned and the noise covariance being aligned with the Hessian) and show that the anisotropic noise in SGD satisfies both these properties. Similarly, [8] show that the multiplicative noise caused by variance in the local rates of convergence leads to heavy tailed stationary behavior in the parameters enabling more efficient exploration of the loss surface for nonconvex functions. By connecting the loss curvature with respect to the parameters to the curvature with respect to the input data, [12] show that flat minima regularize the gradient of the learned function (i.e., the geometric complexity), and that SGD implicitly regularizes the Sobolev seminorms of the learned function with respect to the data leading to improved generalization and robustness. Finally, [26] show that the escaping time of SGD depends on the Radon measure of basin positivity and the heaviness of the gradient noise negatively. They then use this to explain why SGD escapes basins faster than ADAM: (a) by adaptively scaling each gradient, ADAM reduces the anisotropic structure of the gradient noise and (b) by smoothing past gradients, ADAM dampens the gradient noise tails compared to SGD.

Step noise Stochasticy of the minibatch is not the only source of noise is SGD. SGD takes non optimal step sizes in discrete time, rather than an idealized continuous gradient flow. In addition to making learning computationally tractable, these discrete steps insert further noise into the training process. That is, after each step, gradient descent actually steps off the exact continuous path that minimizes the loss at each point. [2] show that this divergence induces a form of implicit regularization by penalizing gradient descent trajectories that have large loss gradients. Furthermore, they show this implicit regularization is proportional to the square of the loss surface slope, enabling the design of an explicit regularization penalty when the organic effect is not strong enough. Both the implicit and explicit regularization biases training towards optimization paths with shallower slopes and optima in flatter loss basins.

This implies that the learning rate (step size) should have a large effect on the optimization. Indeed, folk wisdom in the ML community has it that there exists a Goldilocks step size: too small and training will be too slow or get stuck in a local optima, too large and training may diverge. [1] show that these middle ground step sizes lead the iterates to jump from one side of a valley to the other causing loss stabilization, as opposed to too large of a step which causes the iterate to jump wholly out of the valley. They show that this loss stabilization "induces a hidden stochastic dynamics orthogonal

to the bouncing directions that biases it implicitly toward simple predictors," where simplicity is operationalized as sparsity of representations. This is suggestive of an intriguing connection between SGD inductive biases and compositional sparsity [17]

3.2 Compression Bias

Next we describe how this bias towards broad optima implies a bias towards data compression in the overparameterized regime. When optimizing overparameterized functions, there are many global optima that lie on high dimensional ridges or subspaces, rather than isolated peaks. The natural question then, is which point does SGD converge to?

To study this, lets consider two possible parameter settings θ_1 and θ_2 of N parameters in a deep network that achieve perfect training loss. θ_1 uses k_1 parameters to fit the data with $k_1 \approx N$ where as θ_2 uses $k_2 \ll N$ parameters, implying that θ_2 does compression of the data to learn a smaller model. Now if we consider the first order Taylor approximation around the the minimum, we have $N - k_1$ basis directions that preserve optimality for θ_1, while for θ_2 there are $N - k_1 \gg N - k_2$ basis directions. Of course the volume of the space grows exponentially with dimensionality so the optimal subspace surrounding θ_2 is exponentially bigger than θ_1. Hence, in general, there are exponentially more solutions which compress the data. [14] show that all of these solutions are approximately equally likely to be found by gradient descent conditioned on optimality. Therefore, conditioned on finding a zero loss solution, the probably of it compressing the data approaches one as we increase model size. Finally to bring things full circle, solutions which compress the data have Hessians with many zero eigenvalues, and therefore lie in broad loss basins, and therefore also generalize.

However the only place where SGD was implicitly used in this argument was to motivate the assumption that we might actually find a solution with perfect training loss in practice. Therefore, the degree to which this is a bias of SGD or overparameterized optimization in general is not clear. One might make the claim that SGD does not introduce any bias of the solution conditioned on it being optimal, but that SGD is biased towards finding optimal solutions, over what one might naively expect for high dimensional non-convex optimization problems.

Ultimately further research is needed.

4 Conclusion

In conclusion, we explore two major classes of inductive biases of SGD training: simplicity and broad basins. For simplicity, we explore the many notions of simplicity and their observed biases, in particular rank bias, spectral bias, effective depth, and geometric complexity, as well as the dynamics of simplicity over the course of training. We then show that various aspects of the noise of SGD, both from batches and from discrete steps, help SGD escape local minima and bias the optimization towards solutions in broad loss basins, where such solutions generalize better and imply a bias towards data compression.

References

[1] Maksym Andriushchenko et al. “SGD with large step sizes learns sparse features”. In: (Oct. 2022). arXiv:2210.05337 [cs, stat]. URL: http://arxiv.org/abs/2210.05337.

[2] David G. T. Barrett and Benoit Dherin. *Implicit Gradient Regularization*. arXiv:2009.11162 [cs, stat]. July 2022. DOI: 10.48550/arXiv.2009.11162. URL: http://arxiv.org/abs/2009.11162.

[3] Arwen V. Bradley, Carlos Alberto Gomez-Uribe, and Manish Reddy Vuyyuru. *Shift-Curvature, SGD, and Generalization*. arXiv:2108.09507 [cs, stat] version: 3. July 2022. DOI: 10.48550/arXiv.2108.09507. URL: http://arxiv.org/abs/2108.09507.

[4] Pratik Chaudhari et al. *Entropy-SGD: Biasing Gradient Descent Into Wide Valleys*. arXiv:1611.01838 [cs, stat]. Apr. 2017. URL: http://arxiv.org/abs/1611.01838.

[5] Benoit Dherin et al. *Why neural networks find simple solutions: the many regularizers of geometric complexity*. arXiv:2209.13083 [cs, stat]. Sept. 2022. DOI: 10.48550/arXiv.2209.13083. URL: http://arxiv.org/abs/2209.13083.

[6] Tomer Galanti, Liane Galanti, and Ido Ben-Shaul. *On the Implicit Bias Towards Minimal Depth of Deep Neural Networks*. arXiv:2202.09028 [cs] version: 9. Sept. 2022. DOI: 10.48550/arXiv.2202.09028. URL: http://arxiv.org/abs/2202.09028.

[7] Tomer Galanti and Tomaso Poggio. *SGD Noise and Implicit Low-Rank Bias in Deep Neural Networks*. Tech. rep. Center for Brains, Minds and Machines (CBMM), 2022.

[8] Liam Hodgkinson and Michael W. Mahoney. *Multiplicative noise and heavy tails in stochastic optimization*. arXiv:2006.06293 [cs, math, stat]. June 2020. DOI: 10.48550/arXiv.2006.06293. URL: http://arxiv.org/abs/2006.06293.

[9] Qingguo Hong et al. *On the Activation Function Dependence of the Spectral Bias of Neural Networks*. arXiv:2208.04924 [cs]. Sept. 2022. DOI: 10.48550/arXiv.2208.04924. URL: http://arxiv.org/abs/2208.04924.

[10] Minyoung Huh et al. *The Low-Rank Simplicity Bias in Deep Networks*. arXiv:2103.10427 [cs]. Apr. 2022. DOI: 10.48550/arXiv.2103.10427. URL: http://arxiv.org/abs/2103.10427.

[11] Nitish Shirish Keskar et al. "On large-batch training for deep learning: Generalization gap and sharp minima". In: *arXiv preprint arXiv:1609.04836* (2016).

[12] Chao Ma and Lexing Ying. "The sobolev regularization effect of stochastic gradient descent". In: *arXiv preprint arXiv:2105.13462* (2021).

[13] William Merrill et al. *Effects of Parameter Norm Growth During Transformer Training: Inductive Bias from Gradient Descent*. arXiv:2010.09697 [cs]. Sept. 2021. URL: http://arxiv.org/abs/2010.09697.

[14] Chris Mingard et al. "Is SGD a Bayesian sampler? Well, almost". In: *Journal of Machine Learning Research* 22 (2021).

[15] Preetum Nakkiran et al. "Sgd on neural networks learns functions of increasing complexity". In: *arXiv preprint arXiv:1905.11604* (2019).

[16] Vardan Papyan, XY Han, and David L Donoho. "Prevalence of neural collapse during the terminal phase of deep learning training". In: *Proceedings of the National Academy of Sciences* 117.40 (2020), pp. 24652–24663.

[17] Tomaso Poggio. *Compositional Sparsity: a framework for ML*. Tech. rep. Center for Brains, Minds and Machines (CBMM), 2022.

[18] Nasim Rahaman et al. "On the Spectral Bias of Neural Networks". In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, Sept. 2019, pp. 5301–5310. URL: https://proceedings.mlr.press/v97/rahaman19a.html.

[19] Maria Refinetti, Alessandro Ingrosso, and Sebastian Goldt. *Neural networks trained with SGD learn distributions of increasing complexity*. arXiv:2211.11567 [cond-mat, stat]. Nov. 2022. URL: http://arxiv.org/abs/2211.11567.

[20] Harshay Shah et al. *The Pitfalls of Simplicity Bias in Neural Networks*. arXiv:2006.07710 [cs, stat]. Oct. 2020. DOI: 10.48550/arXiv.2006.07710. URL: http://arxiv.org/abs/2006.07710.

[21] Samuel L Smith et al. "On the origin of implicit regularization in stochastic gradient descent". In: *arXiv preprint arXiv:2101.12176* (2021).

[22] Nadav Timor, Gal Vardi, and Ohad Shamir. "Implicit regularization towards rank minimization in relu networks". In: *arXiv preprint arXiv:2201.12760* (2022).

[23] Andreas Veit, Michael J Wilber, and Serge Belongie. "Residual networks behave like ensembles of relatively shallow networks". In: *Advances in neural information processing systems* 29 (2016).

[24] Greg Yang and Hadi Salman. "A fine-grained spectral perspective on neural networks". In: *arXiv preprint arXiv:1907.10599* (2019).

[25] Annan Yu, Yunan Yang, and Alex Townsend. *Tuning Frequency Bias in Neural Network Training with Nonuniform Data*. arXiv:2205.14300 [cs] version: 2. Sept. 2022. DOI: 10.48550/arXiv.2205.14300. URL: http://arxiv.org/abs/2205.14300.

[26] Pan Zhou et al. *Towards Theoretically Understanding Why SGD Generalizes Better Than ADAM in Deep Learning*. arXiv:2010.05627 [cs, math, stat]. Nov. 2021. DOI: 10.48550/arXiv.2010.05627. URL: http://arxiv.org/abs/2010.05627.

[27] Zhanxing Zhu et al. “The anisotropic noise in stochastic gradient descent: Its behavior of escaping from sharp minima and regularization effects”. In: *arXiv preprint arXiv:1803.00195* (2018).