On the Context of Simplicity Bias

Jhih-Yi Hsieh, Xuchen Gong, Tianwen Fu
Carnegie Mellon University

Abstract

The success of neural networks depends on the generalization ability. Shah et al. conclude that the inherent bias towards simplistic features, Simplicity Bias, hurts generalization by preferring simple but noisy features to complex yet predictive ones. However, we show that the observance of simplicity bias depends on other factors, including the number of input dimensions, the number of samples in the training set, the model size, the noisiness of simple features, and the variance of positive samples. We hope future work can improve and provide more precise definitions of the context where simplicity bias holds.

1 Introduction

The success of deep neural networks is substantially based on the generalization ability in spite of overparameterization. However, the distinction between "generalization" and "overfitting" on unseen data actually refers to the alignment of the inductive biases of learned models and the prior assumptions of the real world. In this regard, practitioners aim to discover the underlying theory of the model biases and to design algorithms that produce models best match the real-world scenario.

Conventional wisdom, guided by Occam's Razor, restricts the hypotheses classes to models with fewer small-magnitude parameters. Approaches include parameter sharing [8] and regularization [7, 13, 16]. Moreover, recent work [1, 6, 15] shows that neural networks optimized by stochastic gradient descent (SGD) are inherently biased towards simpler features, named Simplicity Bias (SB).

On the other hand, a more preferable prior assumption is the local smoothness of data distribution rather than the simplicity of underlying models. Following the guideline, non-parametric models [3, 5] and adversarial training [9] focus on the local consistency, while support vector machines [2] explicitly prioritize finding largest-margin models. Taking into account the margin of the decision boundary, Shah et al. [12] present the negative impacts of extreme SB: relying exclusively on simple features impairs the robustness of neural networks, leading to poor generalization and out-of-distribution performance. Inspired by the piecewise-linear decision boundary of ReLU-activated multi-layer perceptrons, they proposed synthetic datasets composed of linear slabs, and formulate the notion of "simplicity" by the number of slabs. Experiments show that neural networks neglect complex features even if they are more predictive and have larger margins, while regularization techniques and adversarial training do not mitigate this issue. Methods aiming to resolve the issue, such as gradient regularization for model ensembles [14], lack theoretical guarantee and still have gaps in the optimal performance.

Nevertheless, the context in which simplicity bias [12] is observed is not clear. In this project, we aim to conduct experiments that take into account the different model training contexts to better understand the behavior of such simplicity bias, and we investigate whether the existing definition of *relative complexity of optimal decision boundary* is sufficient. Our goal is not to give a new definition for simplicity; rather, we present empirical evidence showing various ways in which the existing definition cannot explain. We have observed not only the relative complexity of optimal decision boundary, but other training context factors concerning the dataset, model and optimization also affect the simplicity of features and thus the generalization performance of neural networks.

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

2 Related Work

Simplicity bias. The tendency of deep neural networks (DNN) to learn simple hypotheses, called simplicity bias (SB), has been linked to their generalization performance. Some regard SB as the source of regularization and generalization by showing that the learned high-probability functions of DNN tend to have low Lempel-Ziv complexity [15]. Similar arguments are made elsewhere, relying on the evidence that a DNN tends to learn simple patterns first [16, 1]. However, opposite viewpoints arise, suggesting that the extreme SB leads the neural networks to exclusively rely on the simplest features despite the better predictive power of the complex ones [12]. The resultant small margin classifiers, therefore, exhibit poor generalization and robustness performance.

Resolving simplicity bias. Previous experiments concluded that changing hyperparameters, optimizers, architectures, and initializations as well as the conventional cultivation of robustness and generalization, such as adversarial training and ensembles, are shown to be ineffective in alleviating SB to some extend [12]. Therefore, multiple alternatives that encourage a deep model to learn complex features are proposed. Assuming that simple solutions are unlikely to be optimal, one approach utilizes a low-capacity network, which is only capable of capturing simple patterns, to detect and downweigh the shortcuts learned by a high-capacity network [4]. Another group of methods focuses on balanced learning across features, through either regularization techniques [11] or enforcing disagreement among a group of models to make use of diverse features [14, 10].

3 Methods

3.1 Generation of Synthetic Data

The synthetic dataset [12] contains N samples $\{(x^{(i)}, y^{(i)})\}_{i=0}^{N-1}$ of d-dimensional features $\{x^{(i)} = (x_0^{(i)}, x_1^{(i)}, \cdots, x_{d-1}^{(i)})\}_{i=0}^{N-1}$ and binary labels $\{y^{(i)}\}_{i=0}^{N-1}$ where $y^{(i)} \in \{-1, +1\}$. Each dimension $X_j \in [-w_j, w_j]$ has width w_j and is a s_j-slab feature, different s_j's bringing different levels of simplicity. Any consecutive slab pieces are separated by margin γ_j, possibly filled with noise, making up ζ_j proportion of data. For each sample in the dataset,Y is sampled independently from Bernoulli distribution, where $P(Y = -1) = P(Y = +1) = 1/2$. For each data point and corresponding label $y^{(i)}$ we then sample each feature dimension X_j independently based on the feature class.

For each dimension X_j, we first split the interval $[-w_j, w_j]$ into s_j intervals with margin γ_j. Let

$$l_j = \frac{2w_j - 2\gamma_j(s_j - 1)}{s}$$

denote the width of each slab; we may then find the intervals $\{U_j^{(k)}\}_{k=0}^{s_j-1}$ to be

$$U_j^{(k)} = [-w + k(l_j + 2\gamma_j), -w + k(l_j + 2\gamma_j) + l_j]\,.$$

Then, we define the alternating intervals of positive slabs and negative slabs by

$$U_j^+ = U_j^{(0)} \cup U_j^{(2)} \cup \cdots \cup U_j^{2\left(\lceil \frac{s_j}{2} \rceil - 1\right)},\ U_j^- = U_j^{(1)} \cup U_j^{(3)} \cup \cdots \cup U_j^{2\lfloor \frac{s_j}{2} \rfloor - 1}.$$

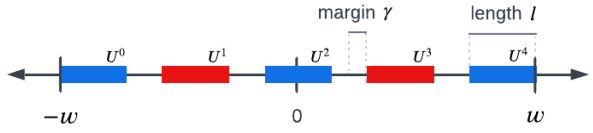


Figure 1: An illustrated example with annotations of a 5-slab feature. The blue slabs (intervals) are in U^+, and the red slabs are in U^-.

Based on the slab intervals, we sample $x_j^{(i)}$ based on the conditional probability density $p(x_j^{(i)}|y^{(i)})$. To ensure equal variances $Var[X_j|Y = 1] = Var[X_j|Y = -1]$ for both positive and negative samples, we define the side slabs as

$$U_j^{+,s} = U_j^{(0)} \cup U_j^{2\left(\lceil \frac{s_j}{2} \rceil - 1\right)},\ U_j^{-,s} = U_j^{(1)} \cup U_j^{2\lfloor \frac{s_j}{2} \rfloor - 1},$$

and let $\mathcal{P}_j^+, \mathcal{P}_j^-$ denote hyperparameters denoting the probability of sampling from the slabs on both farthest sides [12]. The probability density functions are shown in equation (1).

$$f_{x_j^{(i)}|y^{(i)}=-1}(x) = \begin{cases} \frac{\mathcal{P}_j^-}{2l_j} & \text{if } x \in U_j^{-,s}, \\ \frac{1-\mathcal{P}_j^-}{\left(\lfloor \frac{s_j}{2} \rfloor - 2\right) l_j} & \text{if } x \in U_j^- \setminus U_j^{-,s}, \\ 0 & \text{otherwise.} \end{cases}, \; f_{x_j^{(i)}|y^{(i)}=+1}(x) = \begin{cases} \frac{\mathcal{P}_j^+}{2l_j} & \text{if } x \in U_j^{+,s}, \\ \frac{1-\mathcal{P}_j^+}{(\lceil \frac{s_j}{2} \rceil - 2) l_j} & \text{if } x \in U_j^+ \setminus U_j^{+,s}, \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

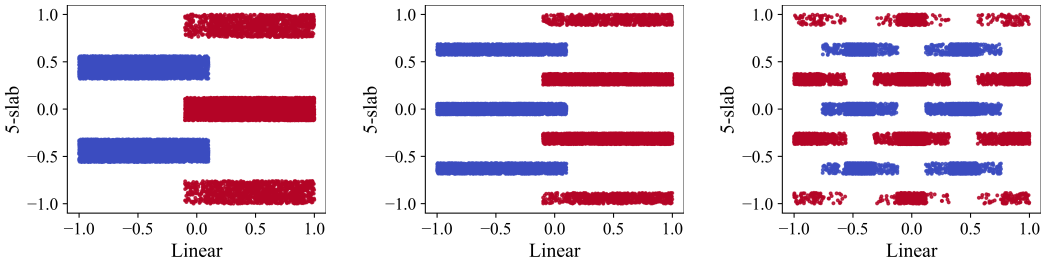
Noisy Features. For the features X_j with noise proportion $\zeta_j > 0$, we generate noise labels Z_j independently of Y by $P(Z_j = 1) = \zeta_j, P(Z_j = 0) = 1 - \zeta_j$. For $Z_j = 0$, X_j is generated according to Y exactly by Equation (1) in the last paragraph. For $Z_j = 1$, X_j is uniformly distributed in the gaps $[-w, w] \setminus (U_j^+ \cup U_j^-)$ regardless of Y.

3.2 Specification of Our Datasets

Following the convention in [12], we mainly use these datasets for our experiments:

- $\hat{L}MS - k$: one linear (2-slab) block and multiple k-slab blocks. For $k = 5$, $\mathcal{P}_j^+ = 1/4$ and $\mathcal{P}_j^- = 1$ (note that $U_j^{-,s} = U_j^-$ in this case); for $k = 7$, $\mathcal{P}_j^+ = 1/8$ and $\mathcal{P}_j^- = 1/2$. The linear feature is noised with $\zeta_0 = 0.1$, and the more complex k-slab features have 100% predictive power. S-randomized accuracy and S^C-randomized accuracy denotes the accuracy where the linear and k-slab dimension is corrupted respectively.
- $\hat{M}S - (m, n)$: one noisy m-slab block and multiple n-slab blocks without noise. $\mathcal{P}$ are the same as in $\hat{L}MS - k$.

A visualization of the datasets is shown in Figure 2.



(a) $\hat{L}MS - 5$ (b) $\hat{L}MS - 7$ (c) $\hat{M}S - (5, 7)$

Figure 2: Visualization of the datasets. On each of the above figures, the x-axis visualizes the first dimension and the y-axis visualizes the second dimension.

4 Results

Baseline Setups. The baseline setups are the same as mentioned in [12]. A summary of the datasets and hyperparameters:

- We have two baseline datasets as specified in 3.2: (1) $\hat{L}MS - 5$ dataset and (2) $\hat{L}MS - 7$ dataset. They each have $d = 50$ input dimensions (one linear dimension, and 49 $\{5, 7\}$-slab dimensions) and 50000 training samples.
- The baseline model is a fully connected neural network with ReLU activations. The model has one hidden layer and the latent dimension is 300.
- SGD with learning rate $\alpha = 0.3$, weight decay $\lambda = 5.0 \cdot 10^{-4}$ and no momemtum.

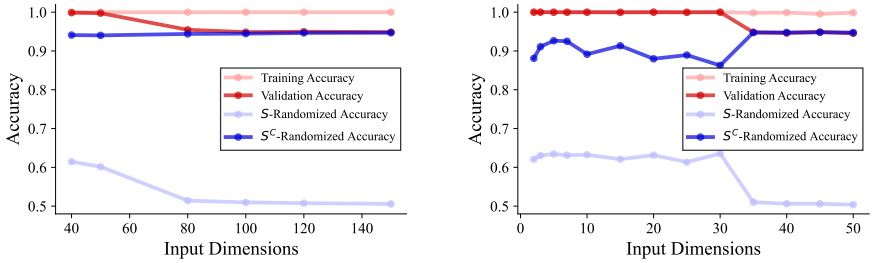
In our experiments, this setup is assumed unless otherwise specified.

Metrics. We use the same metrics as proposed in [12].

- **Training accuracy.** Classification accuracy of the training set.
- **Validation accuracy.** Classification accuracy of the validation set.
- **S-Randomized accuracy.** Classification accuracy of the validation set with simple feature values randomly shuffled.
- **S^c-Randomized accuracy.** Classification accuracy of the validation set with complex feature values randomly shuffled.

4.1 Number of Input Dimensions

This experiment shows model performances under different input dimension sizes. The experiments conducted by Shah et al. [12] are all based on $d = 50$. We found, however, models of the same size tend to suffer more from simplicity bias for high input dimensionality, as shown in Figure 3, although the features are independently generated.



(a) $\hat{L}MS - 5$ (b) $\hat{L}MS - 7$

Figure 3: Training and validation accuracy with different input dimensions. As the input dimension increases, the model tends to focus more exclusively on the simple but noisy feature. Dots aligned vertically represent different metrics on the same model.

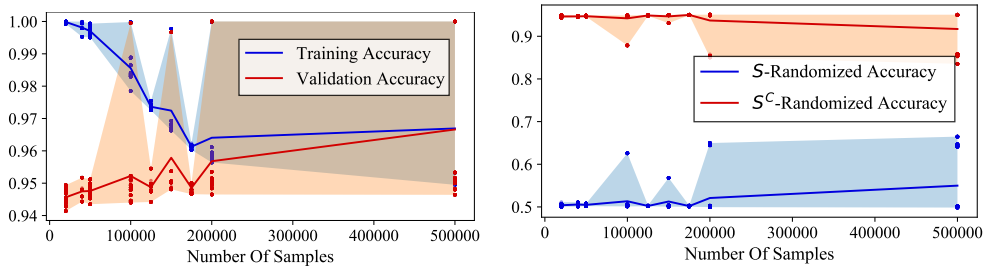
4.2 Number of Samples in Training Dataset

In this experiment (shown in Figure 4b), we tried to observe the model performances under different training dataset sizes in terms of the number of samples. As the number of training samples grows, converged models are less prone to simplicity bias and can reach optimal validation accuracy; however, along with the trend, there are also an increasing number of experiments where the model cannot achieve 100% training accuracy due to the failure of "memorizing" the multiple-slab features of the training samples.

In both cases, the generalization gap decays with the number of training samples, and the training-validation alignment benefits from larger training sets. Therefore, contrary to the claim by Shah [12] that simplicity bias hurts generalization, we hypothesize that the poor generalization performance is due to the insufficiency of the training dataset.

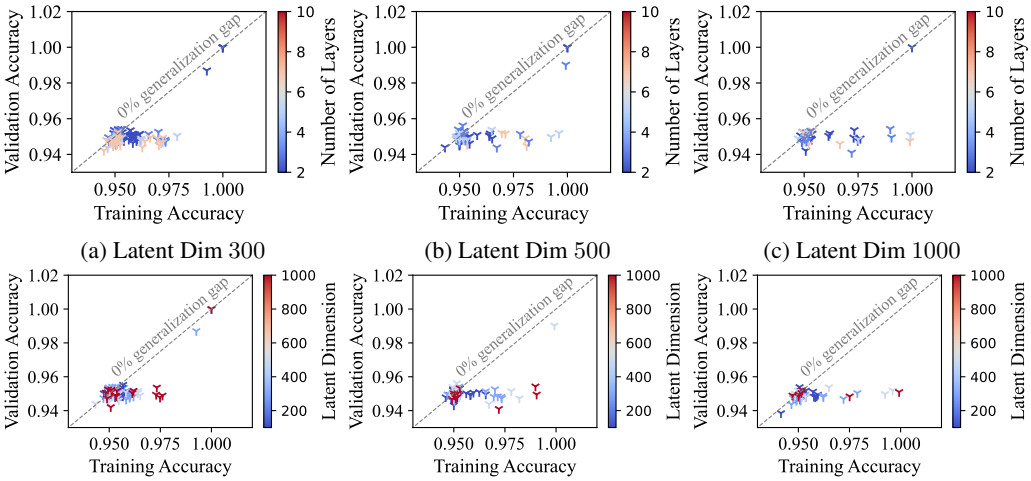
4.3 Model Size (Depth and Latent Dimension)

This experiment aims to investigate the performances of models of varying sizes, in terms of both model depth and the size of latent dimensions. The number of samples is set to 200000, where the model is hard to "memorize and overfit". In figure 5, we show that in terms of the $\hat{L}MS$ datasets, wide and shallow networks are preferable to narrow and deep networks. Although a number of models fall below the 0% generalization gap line, we still observe some models with proper hyperparameters can reach 100% accuracy on both the training set and the validation set. Therefore, the exhibition of simplicity bias may depend on the initialization and hyperparameters.



(a) Training and validation accuracy. (b) $S-$ and S^C-randomized accuracy.

Figure 4: Accuracies with respect to varying number of samples in $\hat{L}MS-7$. We conduct experiments for each setting with different random seeds and plot the average accuracy (lines) and individual results (dots). As shown in the figure, the generalization gap decreases as the number of samples increases, and whether the validation accuracy can converge to 100% depends on the initialization seed.



(a) Latent Dim 300 (b) Latent Dim 500 (c) Latent Dim 1000

(d) 2 Layers (e) 3 Layers (f) 5 Layers

Figure 5: The training and validation accuracy for model sizes with respect to model depth and latent dimension. Each figure contains models trained with a variety of learning rates, momentum, and random seeds. Experiments that fail to train, with a training accuracy of around 0.5, are omitted. Note for shallow and wide networks, there are some initializations that can achieve 100% validation accuracy, which may suggest that the simplicity bias is more of an initialization problem.

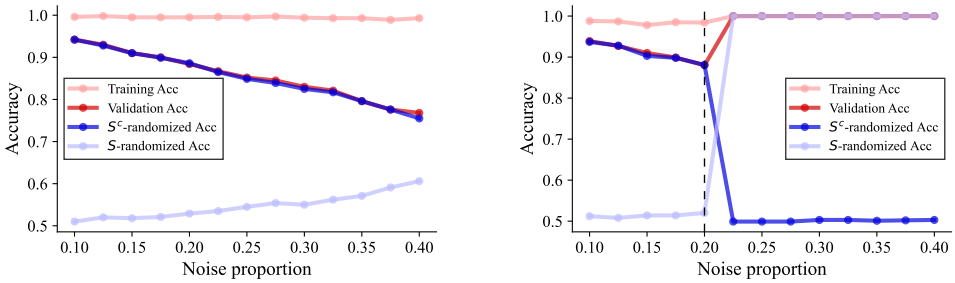
4.4 Noisiness of Simple Features

To investigate how noise proportion affects a model's preferred features for prediction, we vary the noise proportion (i.e., the proportion of the data points that are noisy) in the range $[0.1, 0.4]$ on $\hat{M}S - (5, 7)$, as illustrated in Figure 6.

As shown in Figure 6a, as noise proportion increases, S-randomized accuracy increases, meaning that the model's predictions rely less and less on the simple features. Similar tendencies can also be observed in Figure 6b, where the model totally relies on complex features to make predictions when simple features become noisy to some extent.

Therefore, we conclude that a model has SB when data has a small noise proportion; a large noise proportion (> 0.2 in this case) gives no SB. We hypothesize that noisiness influences simplicity - the noisier, the more complex the simple feature becomes, which motivates the model to learn S^c.

Additionally, we observe that the "inflection point" appears only when there are a large number of training samples: Unlike in the case of the 100k dataset, when there are 40k training samples, S



(a) $\hat{M}S - (5, 7)$ with 40k samples. (b) $\hat{M}S - (5, 7)$ with 100k samples.

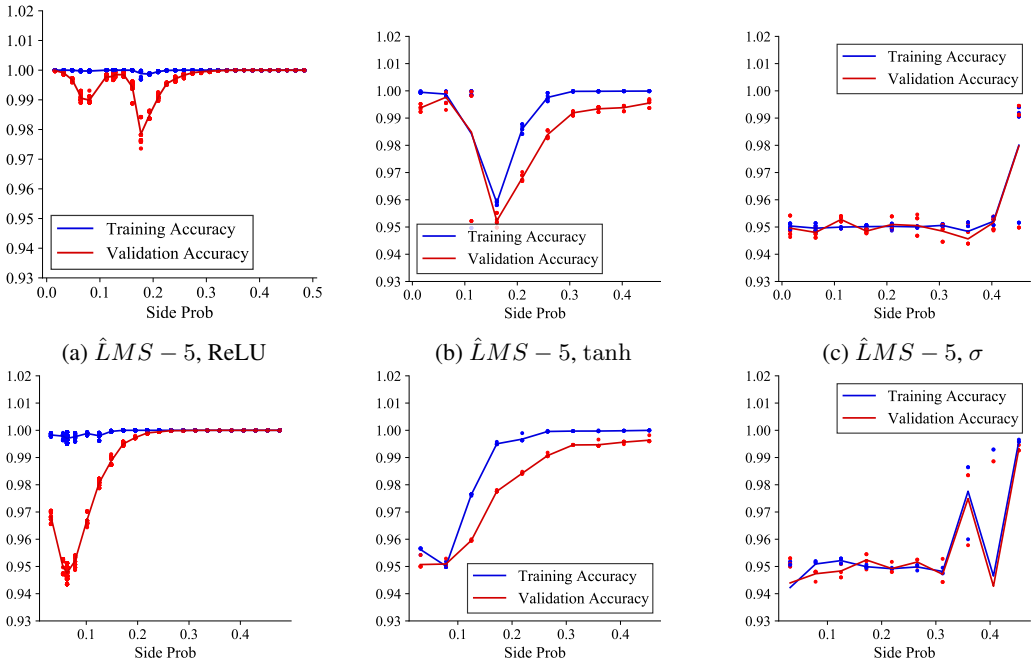
Figure 6: Accuracies vs. noise proportion on $\hat{M}S - (5, 7)$ of different sizes.

is always "biased for". Therefore, we conjecture that the point where a model's preferred feature changes is related to the size of the dataset.

4.5 Variance of Positive Samples and Activation Functions

Denoting the probability of sampling from the slabs on the two farthest sides as $\mathcal{P}^+$, in this experiment, we evaluate the training and validation performance with different values of $\mathcal{P}^+$ (and thus the variance of positive samples) for the 5-slab or 7-slab features for $\hat{L}MS-5$ and $\hat{L}MS-7$ datasets respectively, as shown in Figure 7a - 7f. Experiments show that the simplicity of a feature may also be influenced by the distribution of data rather than merely the complexity of the decision boundary.

In this experiment, we also tried $tanh(x)$ (Figure 7b and 7e) and sigmoid (Figure 7c and 7f) activation functions apart from the baseline ReLU under the same settings.



(a) $\hat{L}MS - 5$, ReLU (b) $\hat{L}MS - 5$, tanh (c) $\hat{L}MS - 5$, σ

(d) $\hat{L}MS - 7$, ReLU (e) $\hat{L}MS - 7$, tanh (f) $\hat{L}MS - 7$, σ

Figure 7: Training and validation accuracy with different $\mathcal{P}^+$. $\mathcal{P}^-$ is fixed at the value in Section 3.2. The suboptimal generalization performance is mostly exhibited only in some middle range of $\mathcal{P}^+$.

5 Conclusion

In our experiments, we have observed that the context under which models are trained can greatly influence the notion of simplicity and the resulting generalization performance in neural networks. Contrary to some previous conclusions, the number of samples, input dimensions, the relative variance of feature space, the number of features, and the noisiness of simple features have all been shown to have influences on feature simplicity and generalization. Under some settings, the Extreme SB [12] behavior is not observed in our experiments. This is supported by evidence showing S^c-Randomized accuracy catching up with and even exceeding S-Randomized accuracy in some experiments (Figures 4b,6).

Moreover, we observed in experiment 4.2 that when the number of training samples is high, the models can achieve 0% generalization gap regardless of simplicity bias, and hyperparameters even including the random seed for initialization may lead to completely different conclusions. We conclude that our results are not well explained under the theories developed in the literature. This cast doubts on whether simplicity bias under the current definition of the relative complexity of optimal decision boundary is complete and can explain poor generalization performances in neural networks.

References

[1] Devansh Arpit, Stanisław Jastrzębski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. A closer look at memorization in deep networks. In *International conference on machine learning*, pages 233–242. PMLR, 2017.

[2] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[3] Thomas Cover and Peter Hart. Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27, 1967.

[4] Nikolay Dagaev, Brett D Roads, Xiaoliang Luo, Daniel N Barry, Kaustubh R Patil, and Bradley C Love. A too-good-to-be-true prior to reduce shortcut reliance. *arXiv preprint arXiv:2102.06406*, 2021.

[5] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018.

[6] Dimitris Kalimeris, Gal Kaplun, Preetum Nakkiran, Benjamin Edelman, Tristan Yang, Boaz Barak, and Haofeng Zhang. Sgd on neural networks learns functions of increasing complexity. *Advances in neural information processing systems*, 32, 2019.

[7] Jan Kukačka, Vladimir Golkov, and Daniel Cremers. Regularization for deep learning: A taxonomy. *arXiv preprint arXiv:1710.10686*, 2017.

[8] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[9] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[10] Matteo Pagliardini, Martin Jaggi, François Fleuret, and Sai Praneeth Karimireddy. Agree to disagree: Diversity through disagreement for better transferability. *arXiv preprint arXiv:2202.04414*, 2022.

[11] Mohammad Pezeshki, Oumar Kaba, Yoshua Bengio, Aaron C Courville, Doina Precup, and Guillaume Lajoie. Gradient starvation: A learning proclivity in neural networks. *Advances in Neural Information Processing Systems*, 34:1256–1272, 2021.

[12] Harshay Shah, Kaustav Tamuly, Aditi Raghunathan, Prateek Jain, and Praneeth Netrapalli. The pitfalls of simplicity bias in neural networks. *Advances in Neural Information Processing Systems*, 33:9573–9585, 2020.

[13] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[14] Damien Teney, Ehsan Abbasnejad, Simon Lucey, and Anton van den Hengel. Evading the simplicity bias: Training a diverse set of models discovers solutions with superior ood generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16761–16772, 2022.

[15] Guillermo Valle-Perez, Chico Q Camargo, and Ard A Louis. Deep learning generalizes because the parameter-function map is biased towards simple functions. *arXiv preprint arXiv:1805.08522*, 2018.

[16] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021.