Simplicity Bias of Two-Layer Networks beyond Linearly Separable Data

Nikita Tsoy Nikola Konstantinov

INSAIT, Sofia University

ICML 2024

Motivation

- Train and test distributions differ in the real-world (Redman, 2016)

Motivation

- Train and test distributions differ in the real-world (Redman, 2016)
- *Distribution shifts* substantially hurt performance (Gulrajani and Lopez-Paz, 2021; Koh et al., 2021)

Motivation

- Train and test distributions differ in the real-world (Redman, 2016)
- *Distribution shifts* substantially hurt performance (Gulrajani and Lopez-Paz, 2021; Koh et al., 2021)
- Networks often rely on *shortcuts*: spurious rules that holds only on train data (Geirhos et al., 2020)

Motivation

- Train and test distributions differ in the real-world (Redman, 2016)
- *Distribution shifts* substantially hurt performance (Gulrajani and Lopez-Paz, 2021; Koh et al., 2021)
- Networks often rely on *shortcuts*: spurious rules that holds only on train data (Geirhos et al., 2020)
- E.g., texture bias in CV (Geirhos et al., 2019) or heuristics in NLP (McCoy et al., 2019)

Simplicity Bias

- One explanation of shortcuts is *simplicity bias*: the propensity of networks to rely on "simple" features (Shah et al., 2020)

Simplicity Bias

- One explanation of shortcuts is *simplicity bias*: the propensity of networks to rely on "simple" features (Shah et al., 2020)
- Preference for features from simpler datasets (Shah et al., 2020; Hu et al., 2020)

Simplicity Bias

- One explanation of shortcuts is *simplicity bias*: the propensity of networks to rely on "simple" features (Shah et al., 2020)
- Preference for features from simpler datasets (Shah et al., 2020; Hu et al., 2020)
- Preference for linear boundaries for *linearly separable* data (Phuong and Lampert, 2021; Lyu et al., 2021; Wang and Ma, 2023)

Simplicity Bias

- One explanation of shortcuts is *simplicity bias*: the propensity of networks to rely on "simple" features (Shah et al., 2020)
- Preference for features from simpler datasets (Shah et al., 2020; Hu et al., 2020)
- Preference for linear boundaries for *linearly separable* data (Phuong and Lampert, 2021; Lyu et al., 2021; Wang and Ma, 2023)
- Limited theoretical understanding of non-linear cases

Research Question

Does the simplicity bias provably emerge in non-linearly separable datasets?

Setting

- Two-layer networks, $f(\boldsymbol{\theta}, \mathbf{x}) := \sum_{j=1}^{m} u_j \phi(\mathbf{v}_j, \mathbf{x})$, where ϕ is ReLU-like activation

Setting

- Two-layer networks, $f(\boldsymbol{\theta}, \mathbf{x}) := \sum_{j=1}^{m} u_j \phi(\mathbf{v}_j, \mathbf{x})$, where ϕ is ReLU-like activation
- Binary cross entropy loss, $L(\boldsymbol{\theta}) := \frac{1}{n} \sum_{i=1}^{n} \ell(f(\boldsymbol{\theta}, \mathbf{x}_i) y_i)$, where $\ell(z) := \ln(1 + \mathrm{e}^{-z})$

Setting

- Two-layer networks, $f(\boldsymbol{\theta}, \mathbf{x}) := \sum_{j=1}^{m} u_j \phi(\mathbf{v}_j, \mathbf{x})$, where ϕ is ReLU-like activation
- Binary cross entropy loss, $L(\boldsymbol{\theta}) := \frac{1}{n} \sum_{i=1}^{n} \ell(f(\boldsymbol{\theta}, \mathbf{x}_i) y_i)$, where $\ell(z) := \ln(1 + \mathrm{e}^{-z})$
- Small initialization, $\boldsymbol{\theta}(0) = \sigma \boldsymbol{\theta}^0$, where σ is small

Setting

- Two-layer networks, $f(\boldsymbol{\theta}, \mathbf{x}) := \sum_{j=1}^{m} u_j \phi(\mathbf{v}_j, \mathbf{x})$, where ϕ is ReLU-like activation
- Binary cross entropy loss, $L(\boldsymbol{\theta}) := \frac{1}{n} \sum_{i=1}^{n} \ell(f(\boldsymbol{\theta}, \mathbf{x}_i) y_i)$, where $\ell(z) := \ln(1 + \mathrm{e}^{-z})$
- Small initialization, $\boldsymbol{\theta}(0) = \sigma \boldsymbol{\theta}^0$, where σ is small
- Balanced initialization, $|u_j(0)| = \|\mathbf{v}_j(0)\|$

Setting

- Two-layer networks, $f(\boldsymbol{\theta}, \mathbf{x}) := \sum_{j=1}^{m} u_j \phi(\mathbf{v}_j, \mathbf{x})$, where ϕ is ReLU-like activation
- Binary cross entropy loss, $L(\boldsymbol{\theta}) := \frac{1}{n} \sum_{i=1}^{n} \ell(f(\boldsymbol{\theta}, \mathbf{x}_i) y_i)$, where $\ell(z) := \ln(1 + \mathrm{e}^{-z})$
- Small initialization, $\boldsymbol{\theta}(0) = \sigma \boldsymbol{\theta}^0$, where σ is small
- Balanced initialization, $|u_j(0)| = \|\mathbf{v}_j(0)\|$
- Training with gradient flow, $\frac{\mathrm{d}\boldsymbol{\theta}}{\mathrm{d}t} = -\nabla L(\boldsymbol{\theta})$

Initial Condensation

First, consider the phase where scale grows from small scale σ to small scale $r := \sigma^{\frac{1}{1+\kappa^*}}$. During that phase:

Initial Condensation

First, consider the phase where scale grows from small scale σ to small scale $r := \sigma^{\frac{1}{1+\kappa^*}}$. During that phase:

- Neurons divide into two types: *prominent* and *non-prominent*

Initial Condensation

First, consider the phase where scale grows from small scale σ to small scale $r := \sigma^{\frac{1}{1+\kappa^*}}$. During that phase:

- Neurons divide into two types: *prominent* and *non-prominent*
- **Prominent** neurons align around several directions that do not depend on the network width

Initial Condensation

First, consider the phase where scale grows from small scale σ to small scale $r := \sigma^{\frac{1}{1+\kappa^*}}$. During that phase:

- Neurons divide into two types: *prominent* and *non-prominent*
- **Prominent** neurons align around several directions that do not depend on the network width
- **Non-prominent** neurons do not contribute to the decision boundary

Initial Condensation

First, consider the phase where scale grows from small scale σ to small scale $r := \sigma^{\frac{1}{1+\kappa^*}}$. During that phase:

- Neurons divide into two types: *prominent* and *non-prominent*
- **Prominent** neurons align around several directions that do not depend on the network width
- **Non-prominent** neurons do not contribute to the decision boundary
- Thus, the network learns only few data-dependent features

Initial Condensation

Consider $G(\mathbf{v}_j) := \frac{1}{n}\sum_{i=1}^{n}(-\ell'(0))\phi(\mathbf{v}_j, \mathbf{x}_i)y_i$ (notice that G does not depend on network width) and $\sigma = r^{1+\kappa^*}$

Theorem

In the limit $r \to 0$, $\exists P \subseteq [m], (u_j^* \in \mathbb{R}, \hat{\mathbf{v}}_j^* \in \mathbb{S}^{d-1})_{j=1}^m$ such that for $T_1 := \frac{1}{\lambda}\ln\left(\frac{r}{\sigma}\right)$, we get

$$\forall j \in P \qquad |u_j(T_1) - ru_j^*| = o(r)$$

$$\|\hat{\mathbf{v}}_j(T_1) - \hat{\mathbf{v}}_j^*\| = o(1), \ |G(\hat{\mathbf{v}}_j^*)| = \max_{\hat{\mathbf{v}} \in \mathbb{S}^{d-1}} |G(\hat{\mathbf{v}})|$$

$$\forall j \in [m] \setminus P \quad |u_j(T_1)| = \|v_j(T_1)\| = o(r)$$

Further Growth

Now, consider the phase where scale grows from small scale r to constant data-dependent scale ε. During that phase:

Further Growth

Now, consider the phase where scale grows from small scale r to constant data-dependent scale ε. During that phase:

- **Prominent** neurons remain inside their alignment clusters

Further Growth

Now, consider the phase where scale grows from small scale r to constant data-dependent scale ε. During that phase:

- **Prominent** neurons remain inside their alignment clusters
- **Non-prominent** neurons still do not contribute to the decision boundary

Further Growth

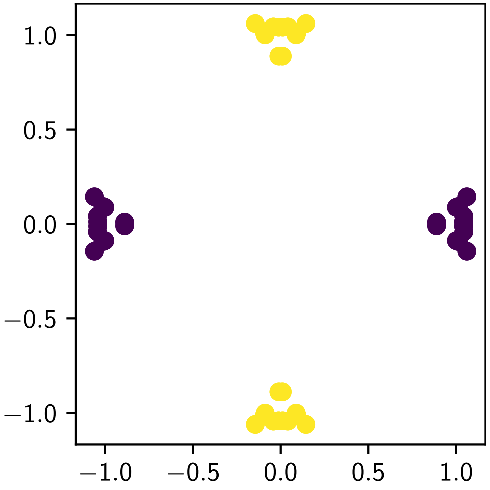
Now, consider the phase where scale grows from small scale r to constant data-dependent scale ε. During that phase:

- **Prominent** neurons remain inside their alignment clusters
- **Non-prominent** neurons still do not contribute to the decision boundary
- Thus, the simplicity bias persists even when the weights grow to a constant scale

Simplicity Bias on XOR-data

Assume that

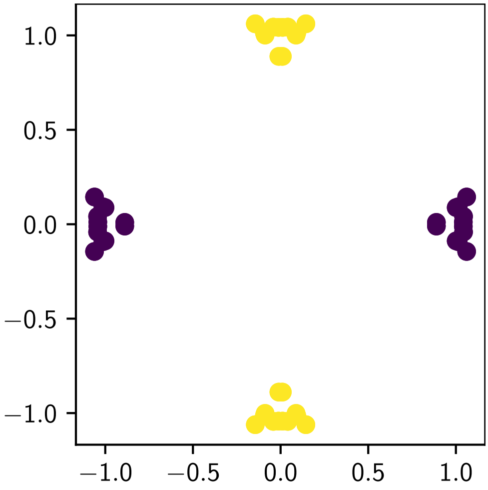
1. Positive points cluster around $\mathbf{e}_1$ and $-\mathbf{e}_1$



Simplicity Bias on XOR-data

Assume that

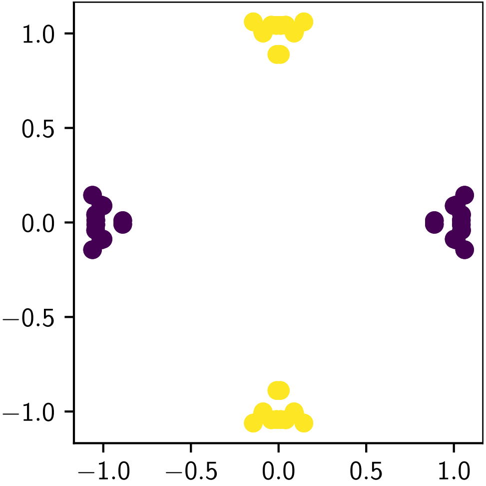
1. Positive points cluster around $\mathbf{e}_1$ and $-\mathbf{e}_1$
2. Negative points cluster around $\mathbf{e}_2$ and $-\mathbf{e}_2$



Simplicity Bias on XOR-data

Assume that

1. Positive points cluster around $\mathbf{e}_1$ and $-\mathbf{e}_1$
2. Negative points cluster around $\mathbf{e}_2$ and $-\mathbf{e}_2$
3. Points are symmetric under reflections



Simplicity Bias on XOR-data

In this setting,

- Initially, the network will behave like 4-neuron network

Simplicity Bias on XOR-data

In this setting,

- Initially, the network will behave like 4-neuron network
- If the underlying 4-neuron network converges, the original network will behave like 4-neuron network even at the end of training

Simplicity Bias on XOR-data

In this setting,

- Initially, the network will behave like 4-neuron network
- If the underlying 4-neuron network converges, the original network will behave like 4-neuron network even at the end of training
- Thus, our alignment results might hold even in the later stages of training

Simplicity Bias on MNIST-CIFAR10 data

- Train ResNet-18 on the train part of MNIST-CIFAR10 domino data (Shah et al., 2020)
  - MNIST image above, CIFAR10 image below
  - Labels come from CIFAR10
  - Classes are perfectly correlated

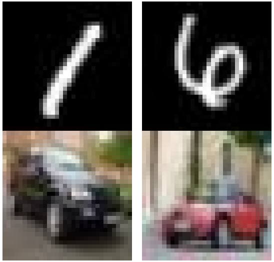


Figure: Examples of train (left) and test (right) inputs

Simplicity Bias on MNIST-CIFAR10 data

- Train ResNet-18 on the train part of MNIST-CIFAR10 domino data (Shah et al., 2020)
  - MNIST image above, CIFAR10 image below
  - Labels come from CIFAR10
  - Classes are perfectly correlated
- Periodically, use last layer to classify OOD portion of the domino dataset
  - The input structure is the same
  - However, top MNIST image may come from any class

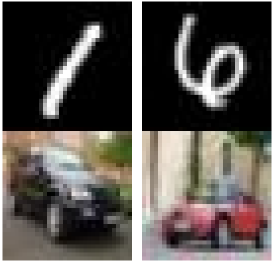


Figure: Examples of train (left) and test (right) inputs

Simplicity Bias on MNIST-CIFAR10 data

- Bad accuracy on the OOD task

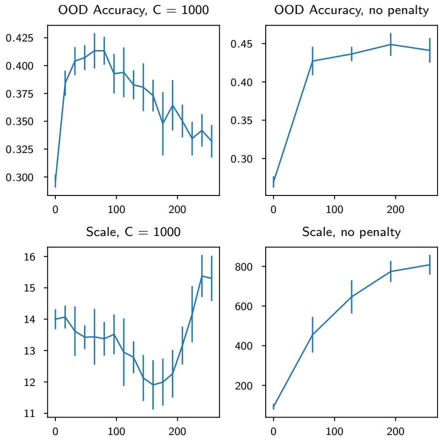


Figure: Accuracy and scale of the logistic regression on the OOD task

Simplicity Bias on MNIST-CIFAR10 data

- Bad accuracy on the OOD task
- The network relies on "simple" MNIST-related features

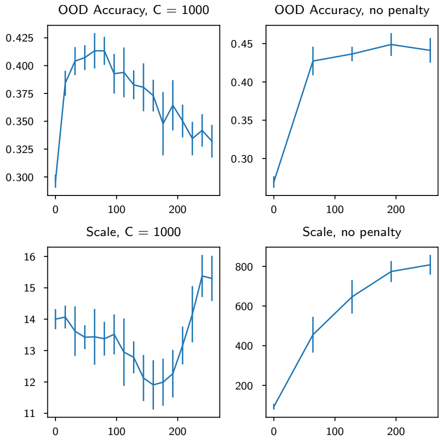


Figure: Accuracy and scale of the logistic regression on the OOD task

Simplicity Bias on MNIST-CIFAR10 data

- Bad accuracy on the OOD task
- The network relies on "simple" MNIST-related features
- Simplicity bias becomes stronger towards the end of training

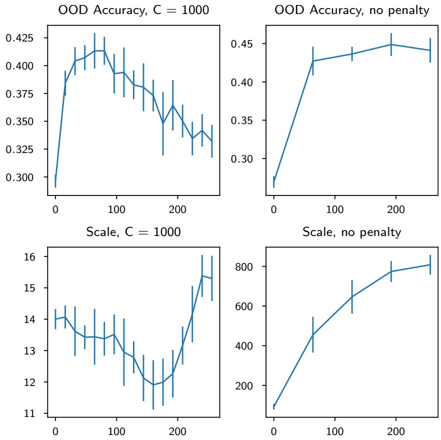


Figure: Accuracy and scale of the logistic regression on the OOD task

Simplicity Bias on MNIST-CIFAR10 data

Discussion

- Simplicity bias exists even in non-linearly separable datasets

Discussion

- Simplicity bias exists even in non-linearly separable datasets
- It manifests as the alignment of features in few data-dependent directions

Discussion

- Simplicity bias exists even in non-linearly separable datasets
- It manifests as the alignment of features in few data-dependent directions
- It can be observed even in real-world datasets and architectures

References I

Geirhos, R., Jacobsen, J.-H., Michaelis, C., Zemel, R., Brendel, W., Bethge, M., and Wichmann, F. A. (2020). Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673.

Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F. A., and Brendel, W. (2019). ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations*.

Gulrajani, I. and Lopez-Paz, D. (2021). In Search of Lost Domain Generalization. In *International Conference on Learning Representations*.

Hu, W., Xiao, L., Adlam, B., and Pennington, J. (2020). The Surprising Simplicity of the Early-Time Learning Dynamics of Neural Networks. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 17116–17128. Curran Associates, Inc.

References II

Koh, P. W., Sagawa, S., Marklund, H., Xie, S. M., Zhang, M., Balsubramani, A., Hu, W., Yasunaga, M., Phillips, R. L., Gao, I., Lee, T., David, E., Stavness, I., Guo, W., Earnshaw, B., Haque, I., Beery, S. M., Leskovec, J., Kundaje, A., Pierson, E., Levine, S., Finn, C., and Liang, P. (2021). WILDS: A Benchmark of in-the-Wild Distribution Shifts. In Meila, M. and Zhang, T., editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 5637–5664. PMLR.

Lyu, K., Li, Z., Wang, R., and Arora, S. (2021). Gradient Descent on Two-layer Nets: Margin Maximization and Simplicity Bias. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W., editors, *Advances in Neural Information Processing Systems*, volume 34, pages 12978–12991. Curran Associates, Inc.

References III

McCoy, T., Pavlick, E., and Linzen, T. (2019). Right for the Wrong Reasons: Diagnosing Syntactic Heuristics in Natural Language Inference. In Korhonen, A., Traum, D., and Màrquez, L., editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.

Phuong, M. and Lampert, C. H. (2021). The inductive bias of ReLU networks on orthogonally separable data. In *International Conference on Learning Representations*.

Redman, T. C. (2016). Bad data costs the us $3 trillion per year. *Harvard Business Review*, 22:11–18.

Shah, H., Tamuly, K., Raghunathan, A., Jain, P., and Netrapalli, P. (2020). The Pitfalls of Simplicity Bias in Neural Networks. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 9573–9585. Curran Associates, Inc.

References IV

Wang, M. and Ma, C. (2023). Understanding Multi-phase Optimization Dynamics and Rich Nonlinear Behaviors of ReLU Networks. In Oh, A., Neumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S., editors, *Advances in Neural Information Processing Systems*, volume 36, pages 35654–35747. Curran Associates, Inc.