THE UNIVERSITY OF CHICAGO

Linear Layers Promote Learning Single-/Multiple-Index Models

Suzanna Parkinson [1] Greg Ongie [2] Rebecca Willett [1]

[1]University of Chicago [2]Marquette University

Set-Up

Every shallow neural network f can be described by a collection of weights $\theta = (\boldsymbol{W}, \boldsymbol{a}, \boldsymbol{b}, c)$:

$$h_\theta^{(2)}(\boldsymbol{x}) = \boldsymbol{a}^\top[\boldsymbol{W}\boldsymbol{x}+\boldsymbol{b}]_+ + c = \sum_{k=1}^{K} a_k[\boldsymbol{w}_k^\top \boldsymbol{x} + b_k]_+ + c. \tag{1}$$

Adding linear layers effectively re-parameterizes f:

$$h_\theta^{(L)}(\boldsymbol{x}) = \boldsymbol{a}^\top[\boldsymbol{W}_{L-1}\cdots\boldsymbol{W}_2\boldsymbol{W}_1\boldsymbol{x}+\boldsymbol{b}]_+ + c \tag{2}$$

where now $\theta = (\boldsymbol{W}_1, \boldsymbol{W}_2, ..., \boldsymbol{W}_{L-1}, \boldsymbol{a}, \boldsymbol{b}, c)$. With any θ we associate the cost

$$C_L(\theta) := \frac{1}{L}\left(\|\boldsymbol{a}\|_2^2 + \|\boldsymbol{W}_1\|_F^2 + \cdots + \|\boldsymbol{W}_{L-1}\|_F^2\right), \tag{3}$$

i.e., the "weight decay" penalty on non-bias terms. We recast this cost in function space:

$$R_L(f) := \inf_\theta C_L(\theta) \ \text{ s.t. } \ f = h_\theta^{(L)}. \tag{4}$$

This is the function-space penalty equivalent of the weight decay penalty for interpolation learning or regularized empirical risk minimization.

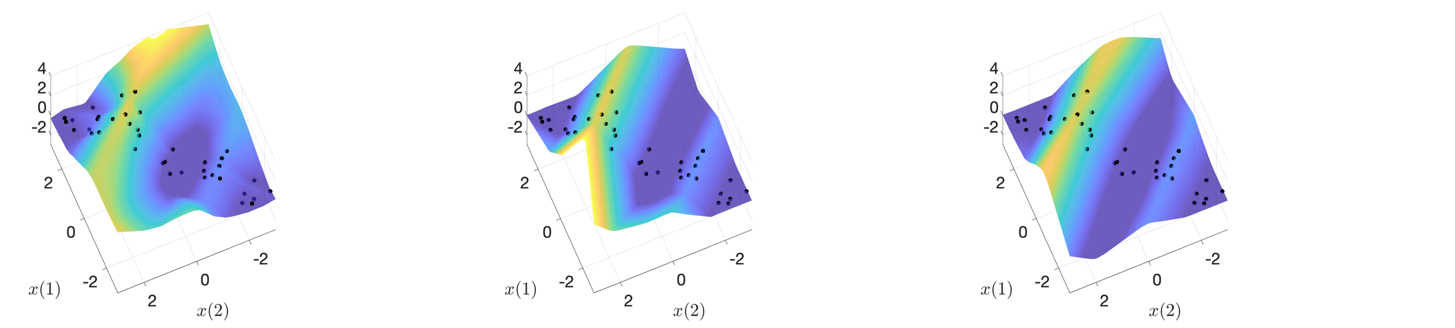

Figure 1. As the number of linear layers increases from left to right, the learned interpolating function will become closer to constant in directions perpendicular to a low-dimensional subspace on which a parsimonious interpolant can be defined.

Definitions

Fix a bounded density ρ such that $\rho(\boldsymbol{x}) > 0$ for all $\boldsymbol{x} \in \mathbb{R}^d$ and consider the uncentered covariance matrix of the gradient of a function:

$$\boldsymbol{C}_{f,\rho} := \mathbb{E}_\rho[\nabla f(\boldsymbol{x})\nabla f(\boldsymbol{x})^\top] = \int \nabla f(\boldsymbol{x})\nabla f(\boldsymbol{x})^\top \rho(\boldsymbol{x})\, d\boldsymbol{x} \tag{5}$$

- The function f is constant in the direction of $\boldsymbol{v} \in \text{null}(\boldsymbol{C}_{f,\rho})$ because
$$\|\boldsymbol{v}^\top \nabla f\|_{L_2(\rho)}^2 = \boldsymbol{v}^\top \boldsymbol{C}_{f,\rho}\boldsymbol{v} \ \ \forall \boldsymbol{v}.$$
- The *active subspace* of a function f is $\text{range}(\boldsymbol{C}_{f,\rho})$.
- The *rank* of a function is $\text{rank}(f) = \text{rank}(\boldsymbol{C}_{f,\rho})$.
- If f is a multi-index model of the form $f(\boldsymbol{x}) = g(\boldsymbol{V}^\top \boldsymbol{x})$ then
$$\boldsymbol{C}_{f,\rho} = \boldsymbol{V}\left[\mathbb{E}_\rho[\nabla g(\boldsymbol{V}^\top\boldsymbol{x})\nabla g(\boldsymbol{V}^\top\boldsymbol{x})^\top]\right]\boldsymbol{V}^\top.$$

Definitions (cont.)

- Let $\sigma_k(f;\rho) := \sigma_k(\boldsymbol{C}_{f,\rho}^{1/2})$
- Define the *mixed variation* of order $q \in (0,1]$ of f with respect to ρ as

$$\mathcal{MV}(f;\rho,q) := \|\boldsymbol{C}_{f,\rho}^{1/2}\|_{S^q} = \left(\sum_{k=1}^{d} \sigma_k(f;\rho)^q\right)^{1/q}.$$

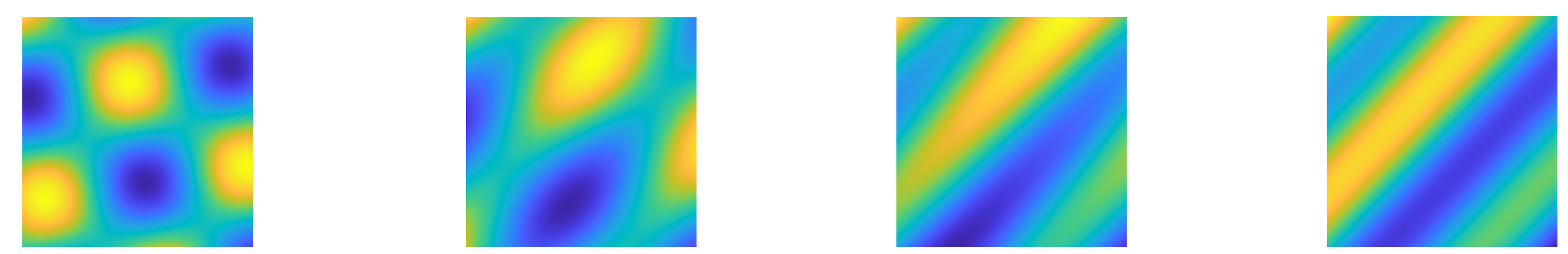

Figure 2. Illustration of four functions in $d = 2$ with mixed variation decreasing from left to right.

Lemma

$$R_2(f)^{2/L} \le R_L(f) \le \text{rank}(f)^{\frac{L-2}{L}} R_2(f)^{2/L} \tag{6}$$

$$\mathcal{MV}\left(f;\rho,\frac{2}{L-1}\right) \le R_L(f)^{L/2} \tag{7}$$

Theorem

For all $f_l, f_h \in \mathcal{N}_2(\mathbb{R}^d)$ such that $\text{rank}(f_l) < \text{rank}(f_h)$, there is a value L_0 such that $L > L_0$ implies $R_L(f_l) < R_L(f_h)$.

Theorem

For all constants $C \ge 1$, $\eta > 0$ and all integers $s \ge 1$ and $k \ge 0$ such that $s + k \le d$, if

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(\boldsymbol{x}_i))^2 + \eta R_L(\hat{f}) \le C\left(\inf_{f\in\mathcal{N}_2(\mathbb{R}^d)} \frac{1}{n}\sum_{i=1}^{n}(y_i - f(\boldsymbol{x}_i))^2 + \eta R_L(f)\right)$$

or $\hat{f}(\boldsymbol{x}_i) = y_i$ and

$$R_L(\hat{f}) \le C\left(\inf_{f\in\mathcal{N}_2(\mathbb{R}^d): f(\boldsymbol{x}_i)=y_i} R_L(f)\right)$$

then

$$\sigma_{s+k}(\hat{f};\rho) = O\left(\left(\frac{s}{s+k}\right)^{(L-1)/2} C^{L/2}\right).$$

Numerical Experiments

To see how adding linear layers affects performance in practice, we performed numerical experiments with and without adding linear layers. All models are of the form (2) with varying values of L.

- For $r = 1, 2$, **Ground Truth** $f_r(\boldsymbol{x}) = \boldsymbol{a}_r^\top[\boldsymbol{W}_r\boldsymbol{x} + \boldsymbol{b}_r]_+$ is a rank-r function with active subspace $\text{range}(\boldsymbol{V})$.
- **Train** and **Test Samples** are generated as $\{(\boldsymbol{x}_i, f_r(\boldsymbol{x}_i))\}_{i=1}, \boldsymbol{x}_i \sim U([-\frac{1}{2},\frac{1}{2}]^{20})$
- **Train** from random initialization using Adam with weight decay parameter $\lambda = 10^{-3}$
- **Estimate $\boldsymbol{C}_{\hat{f},\rho}$** and **Active Subspace Basis $\hat{\boldsymbol{V}}_r$** of $\hat{f}$ and report subspace distance $\|\hat{\boldsymbol{V}}_r\hat{\boldsymbol{V}}_r^\top - \boldsymbol{V}\boldsymbol{V}^\top\|_{op}$.

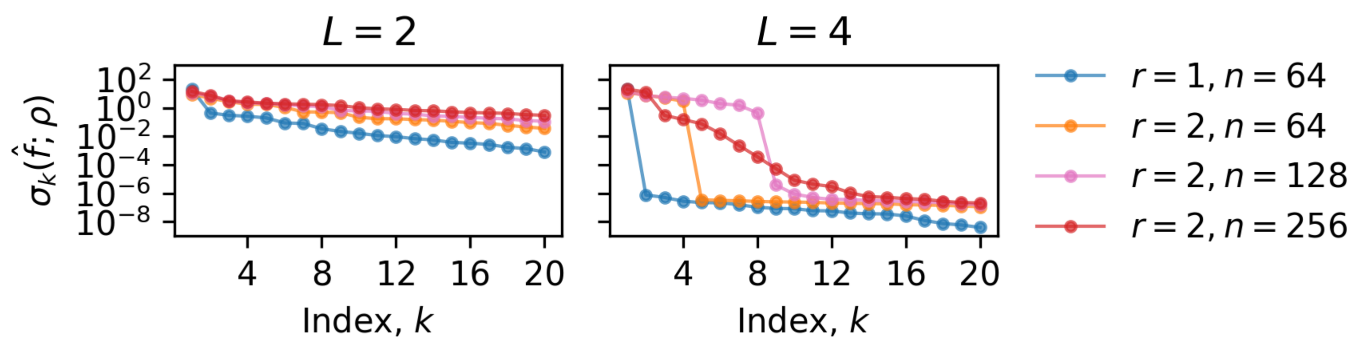

Figure 3. Adding linear layers causes learned networks to have low effective rank. Singular values of trained networks with $L = 2$ (left, no linear layers) vs. $L = 4$ (right, two linear layers). The singular values of the $L = 4$ networks exhibit a sharp dropoff.

r	n	L	Train MSE	Generalization MSE	Out of Distribution MSE	Active Subspace Distance
1	64	2	3.38e-06	1.24e-01	1.09e+00	3.95e-02
		4	8.19e-05	**8.86e-04**	**5.39e-03**	**2.48e-03**
2	64	2	2.69e-07	**1.04e+01**	**4.23e+01**	**7.59e-01**
		4	4.95e-07	1.25e+01	5.02e+01	9.57e-01
2	128	2	7.78e-05	**5.97e+00**	**2.68e+01**	**4.97e-01**
		4	1.74e-05	8.04e+00	3.92e+01	5.88e-01
2	256	2	4.36e-04	4.05e+00	1.87e+01	2.73e-01
		4	9.97e-04	**2.35e-02**	**2.39e-01**	**1.10e-02**

Table 1. With enough data, adding linear layers improves generalization and aligns models with the true active subspace.

Linear Layers Promote Learning Single-/Multiple-Index Models

Suzanna Parkinson [1] Greg Ongie [2] Rebecca Willett [1]

[1]University of Chicago [2]Marquette University

References

[1] Sanjeev Arora, Nadav Cohen, and Elad Hazan. On the optimization of deep networks: Implicit acceleration by overparameterization. In *International Conference on Machine Learning*, pages 244–253. PMLR, 2018. `http://proceedings.mlr.press/v80/arora18a/arora18a.pdf`.

[2] Sanjeev Arora, Nadav Cohen, Wei Hu, and Yuping Luo. Implicit regularization in deep matrix factorization. *Advances in Neural Information Processing Systems*, 32:7413–7424, 2019.

[3] Peter L Bartlett. For valid generalization the size of the weights is more important than the size of the network. pages 134–140, 1997.

[4] Alberto Bietti, Joan Bruna, Clayton Sanford, and Min Jae Song. Learning single-index models with shallow neural networks. *arXiv preprint arXiv:2210.15651*, 2022.

[5] Sitan Chen and Raghu Meka. Learning polynomials in few relevant dimensions. In *Conference on Learning Theory*, pages 1161–1227. PMLR, 2020.

[6] Albert Cohen, Ingrid Daubechies, Ronald DeVore, Gerard Kerkyacharian, and Dominique Picard. Capturing ridge functions in high dimensions from point queries. *Constructive Approximation*, 35:225–243, 2012.

[7] Paul G Constantine. *Active subspaces: Emerging ideas for dimension reduction in parameter studies*. SIAM, 2015.

[8] Paul G Constantine, Eric Dow, and Qiqi Wang. Active subspace methods in theory and practice: applications to kriging surfaces. *SIAM Journal on Scientific Computing*, 36(4):A1500–A1524, 2014.

[9] Amit Daniely. Depth separation for neural networks. In *Conference on Learning Theory*, pages 690–696. PMLR, 2017.

[10] David L Donoho et al. High-dimensional data analysis: The curses and blessings of dimensionality. *AMS math challenges lecture*, 1(2000):32, 2000.

[11] Massimo Fornasier, Karin Schnass, and Jan Vybiral. Learning functions of few arbitrary linear parameters in high dimensions. *Foundations of Computational Mathematics*, 12:229–262, 2012.

[12] Ravi Ganti, Nikhil Rao, Laura Balzano, Rebecca Willett, and Robert Nowak. On learning high dimensional structured single index models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

[13] Ravi Sastry Ganti, Laura Balzano, and Rebecca Willett. Matrix completion under monotonic single index models. *Advances in neural information processing systems*, 28, 2015.

[14] Anna Golubeva, Behnam Neyshabur, and Guy Gur-Ari. Are wider nets better given the same number of parameters? *arXiv preprint arXiv:2010.14495*, 2020. `https://arxiv.org/pdf/2010.14495.pdf`.

[15] Suriya Gunasekar, Blake Woodworth, Srinadh Bhojanapalli, Behnam Neyshabur, and Nathan Srebro. Implicit regularization in matrix factorization. In *2018 Information Theory and Applications Workshop (ITA)*, pages 1–10. IEEE, 2018.

[16] Arthur Jacot. Implicit bias of large depth networks: a notion of rank for nonlinear functions. *arXiv preprint arXiv:2209.15055*, 2022.

[17] Ziwei Ji and Matus Telgarsky. Gradient descent aligns the layers of deep linear networks. *arXiv preprint arXiv:1810.02032*, 2018.

[18] M. J. Kearns. *Computational Complexity of Machine Learning*. PhD thesis, Department of Computer Science, Harvard University, 1989.

[19] Hao Liu and Wenjing Liao. Learning functions varying along a central subspace. *arXiv preprint arXiv:2001.07883*, 2020.

[20] Kaifeng Lyu and Jian Li. Gradient descent maximizes the margin of homogeneous neural networks. *arXiv preprint arXiv:1906.05890*, 2019.

[21] Alireza Mousavi-Hosseini, Sejun Park, Manuela Girotti, Ioannis Mitliagkas, and Murat A Erdogdu. Neural networks efficiently learn low-dimensional representations with SGD. *arXiv preprint arXiv:2209.14863*, 2022.

[22] Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. In search of the real inductive bias: On the role of implicit regularization in deep learning. *arXiv preprint arXiv:1412.6614*, 2014.

[23] Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. Norm-based capacity control in neural networks. pages 1376–1401, 2015.

[24] Behnam Neyshabur, Srinadh Bhojanapalli, David Mcallester, and Nati Srebro. Exploring generalization in deep learning. *Advances in Neural Information Processing Systems*, 30:5947–5956, 2017.

[25] Behnam Neyshabur, Ryota Tomioka, Ruslan Salakhutdinov, and Nathan Srebro. Geometry of optimization and implicit regularization in deep learning. *arXiv preprint arXiv:1705.03071*, 2017.

[26] Greg Ongie, Rebecca Willett, Daniel Soudry, and Nathan Srebro. A function space view of bounded norm infinite width ReLU nets: The multivariate case. *arXiv preprint arXiv:1910.01635*, 2019.

[27] Rahul Parhi and Robert D Nowak. Banach space representer theorems for neural networks and ridge splines. *J. Mach. Learn. Res.*, 22(43):1–40, 2021.

[28] Rahul Parhi and Robert D Nowak. Near-minimax optimal estimation with shallow ReLU neural networks. *IEEE Transactions on Information Theory*, 2022.

[29] Noam Razin and Nadav Cohen. Implicit regularization in deep learning may not be explainable by norms. *arXiv preprint arXiv:2005.06398*, 2020.

[30] Pedro Savarese, Itay Evron, Daniel Soudry, and Nathan Srebro. How do infinite width bounded norm networks look in function space? In *Conference on Learning Theory*, pages 2667–2690. PMLR, 2019. `http://proceedings.mlr.press/v99/savarese19a/savarese19a.pdf`.

[31] Fanhua Shang, Yuanyuan Liu, Fanjie Shang, Hongying Liu, Lin Kong, and Licheng Jiao. A unified scalable equivalent formulation for Schatten quasi-norms. *Mathematics*, 8(8):1325, 2020.

[32] Bo-Ying Wang and Bo-Yan Xi. Some inequalities for singular values of matrix products. *Linear Algebra and its Applications*, 264:109–115, 1997. ISSN 0024-3795. doi:https://doi.org/10.1016/S0024-3795(97)00020-7. URL `https://www.sciencedirect.com/science/article/pii/S0024379597000207`. Sixth Special Issue on Linear Algebra and Statistics.

[33] Colin Wei, Jason D Lee, Qiang Liu, and Tengyu Ma. Regularization matters: Generalization and optimization of neural nets v.s. their induced kernel. *arXiv preprint arXiv:1810.05369*, 2019.

[34] Ke Ye and Lek-Heng Lim. Schubert varieties and distances between subspaces of different dimensions. *SIAM Journal on Matrix Analysis and Applications*, 37(3):1176–1197, 2016. doi:10.1137/15M1054201. URL `https://doi.org/10.1137/15M1054201`.

[35] Xiangrong Yin, Bing Li, and R Dennis Cook. Successive direction extraction for estimating the central subspace in a multiple-index regression. *Journal of Multivariate Analysis*, 99(8):1733–1757, 2008.

[36] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016.

[37] Yu Zhu and Peng Zeng. Fourier methods for estimating the central subspace and the central mean subspace in regression. *Journal of the American Statistical Association*, 101(476):1638–1651, 2006.