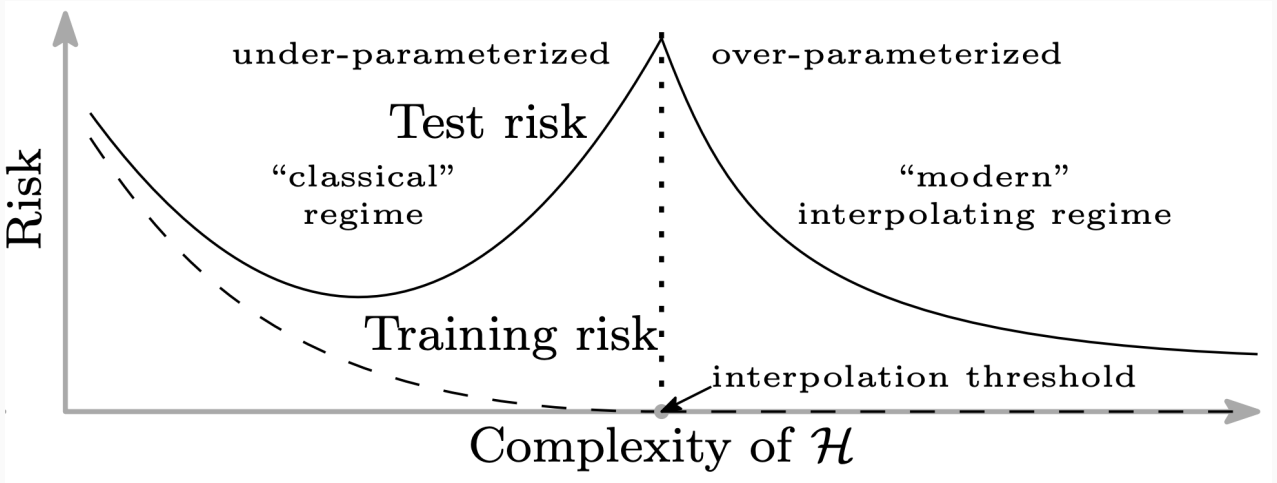
Generalization of Two-layer Neural Networks: An Asymptotic Viewpoint

Jimmy Ba[1,2] Murat A. Erdogdu[1,2] Taiji Suzuki[3,4]
Denny Wu[1,2,4] Tianzong Zhang[2,5]

[1]University of Toronto [2]Vector Institute for Artificial Intelligence
[3]University of Tokyo [4]RIKEN AIP [5]Tsinghua University

International Conference on Learning Representations 2020

Introduction: the Double Descent Phenomenon



Double Descent: second decrease in population risk beyond the "*interpolation threshold*", i.e. when the model interpolates training data.

Previous works provided precise characterization of this phenomenon for the *minimum-norm interpolant* (linear and random features model).

- M. Belkin, D. Hsu, S. Ma, S. Mandal. Reconciling modern machine learning and the bias-variance trade-off.
- T. Hastie, A. Montanari, S. Rosset, R. Tibshirani. Surprises in high-dimensional ridgeless interpolation.

Motivation: Double Descent in Two-layer Networks?

For linear models, the number of parameters is tied to input dimensions.

Motivation of This Work:

- Does this phenomenon generalizes to nonlinear models, in which the model complexity can be controlled **independent of the data**?

Remark: the answer is affirmative for random features model [Mei and Montanari 2019] and principal component regression [Ji and Hsu 2019].

Common mechanism: instability of the *pseudo-inverse*, i.e. the norm of the parameters "blows up" at the interpolation threshold.

Motivation of This Work:

- Does the same mechanism explain the benefit of overparameterization for **neural networks** (in the same proportional limit)?

Remark: "double descent" is empirically observed in neural net optimization.

Motivation: Impact of Optimization and Initialization

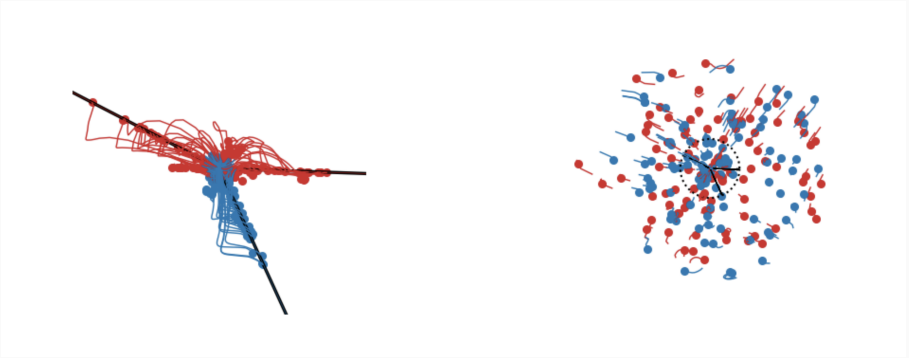
Different optimization procedure:

- Optimizing the **second layer** corresponds to a *random feature model*.
- Optimizing the **first layer** is often *non-convex* due to the nonlinearity.

Different initialization:

- The scale of initialization changes the obtained solution.

Similar analogy: comparison between *kernel* and *mean-field* regime.

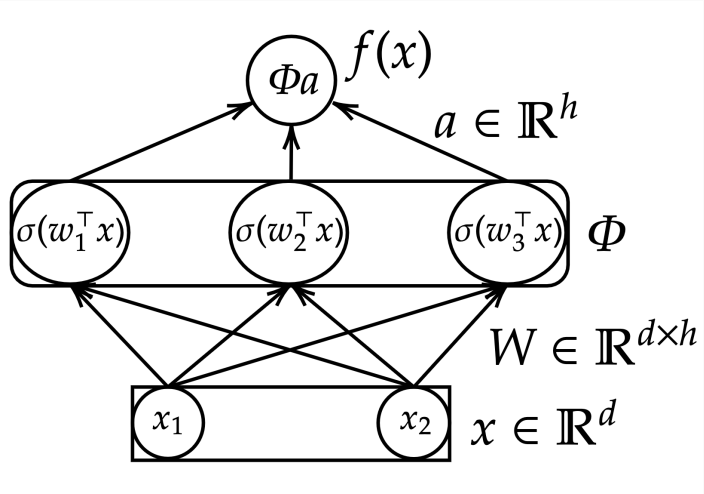


Motivation of This Work:

- How does the **optimization procedure** and the **initialization scale** affect the generalization performance?

- Chizat, L., Oyallon, E. and Bach, F., 2019. On lazy training in differentiable programming.

Problem Setup and Assumptions



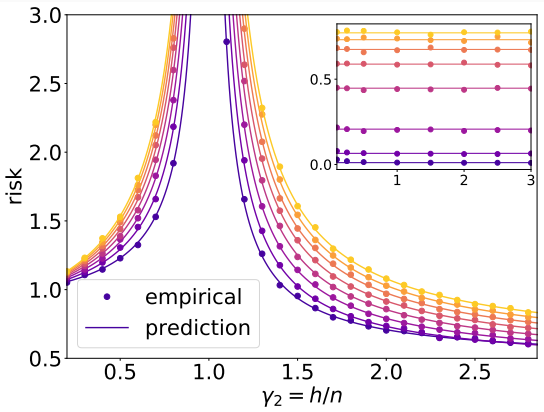
- **Data:** $\boldsymbol{x}_i \sim \mathcal{N}(0, I_d)$.
- **Student:** $f(\boldsymbol{x}) = \sum_{i=1}^{h} a_i \phi(\langle \boldsymbol{x}, \boldsymbol{w}_i \rangle)$.
- **Teacher:** $y_i = \langle \boldsymbol{x}_i, \boldsymbol{\theta}_* \rangle + \varepsilon$. $\|\boldsymbol{\theta}_*\|_2 = r$, $\mathbb{E}[\varepsilon] = 0$, $\mathrm{Var}(\varepsilon) = \sigma^2$.
- **Objective:** minimize (unregularized) MSE: $L(f) = \frac{1}{2n} \sum_{i=1}^{n} (y_i - f(\boldsymbol{x}_i))^2$.

- **Proportional Asymptotics:** $n, d, h \to \infty$, $d/n = \gamma_1$, $h/n = \gamma_2$.
- **Optimization:** *gradient flow* on either the first or second layer.
- **Goal:** derive *prediction risk* $R(f) = \mathbb{E}_{\boldsymbol{x}}[(\langle \boldsymbol{x}, \boldsymbol{\theta}^* \rangle - f(\boldsymbol{x}))^2]$.

Remark: overparameterization corresponds to *increasing* $\gamma_2 = h/n$.

Warm Up: Linear Network

Two-layer linear network: $f(\boldsymbol{x}) = \boldsymbol{x}^\top \boldsymbol{W}\boldsymbol{a}$; Optimize either $\boldsymbol{W}$ or $\boldsymbol{a}$.



$\gamma_1 = d/n > 1$.

- **Main figure:** when only the **2nd layer** is optimized (from zero init.), *double descent w.r.t.* γ_2 occurs when $\gamma_1 > 1$ i.e. $d > n$.
- **Subfigure:** when only the **1st layer** is optimized (for fixed non-zero 2nd layer), risk is *independent to* γ_2 *(overparameterization)*.

Note: darker color corresponds to larger γ_1.

Observation: double descent observed when the *2nd layer* is optimized, but **not** when the *1st layer* is optimized.

Question: is this phenomenon also present in nonlinear networks?

Nonlinear Network: Trained Second Layer

Learning the 2nd layer from zero initialization yields the least squares solution $\hat{\boldsymbol{a}} = \phi(\boldsymbol{X}\boldsymbol{W})^{\dagger}\boldsymbol{y}$, i.e. RF model.

Bias-variance Decomposition:

Variance – quantitative characterization:

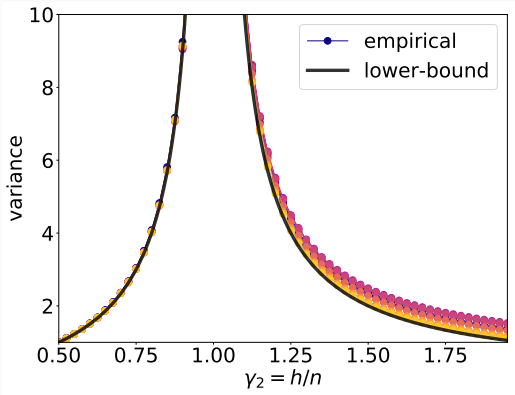
- Independent to $\gamma_1 = d/n$ when $\gamma_2 = h/n < 1$.
- Peaks at $\gamma_2 \to 1$ and then decreases.

Remark: result largely follows from [Cheng and Singer 2013] and [Hastie et al. 2019].

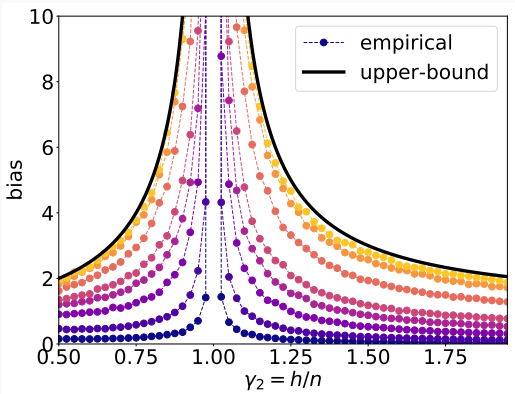
Bias – qualitative characterization:

- Peaks at $\gamma_2 \to 1$ and bounded for $\gamma_2 > 1$.

Remark: [Mei and Montanari 2019] provided a complete characterization for both the bias and variance.



(a) variance (ReLU).



(b) bias (ReLU).

Observation: double descent observed in *both the bias and variance*.

Nonlinear Network: Trained First Layer

2nd Layer: $a_i \sim \{-1/\sqrt{h}, 1/\sqrt{h}\}$ and *fixed* throughout optimization.

Challenge: stationary solution of gradient flow (under *empirical risk*) is often difficult to characterize due to nonlinearity.

Solution: analyze specific initializations that allow the training dynamics to be "**linearized**" (e.g. 1st order Taylor expansion is accurate).

Vanishing initialization: $\|\boldsymbol{W}(t) - \boldsymbol{W}(0)\|_F \gg \|\boldsymbol{W}(0)\|_F$.

- Satisfied by $\boldsymbol{w}_i \sim \mathcal{N}(0, \mathbf{I}/dh^{1+\epsilon})$. Neurons stay close to one another.
- Training can be *linearized around the* ***origin***.

Non-vanishing initialization: $\|\boldsymbol{W}(t) - \boldsymbol{W}(0)\|_F \ll \|\boldsymbol{W}(0)\|_F$.

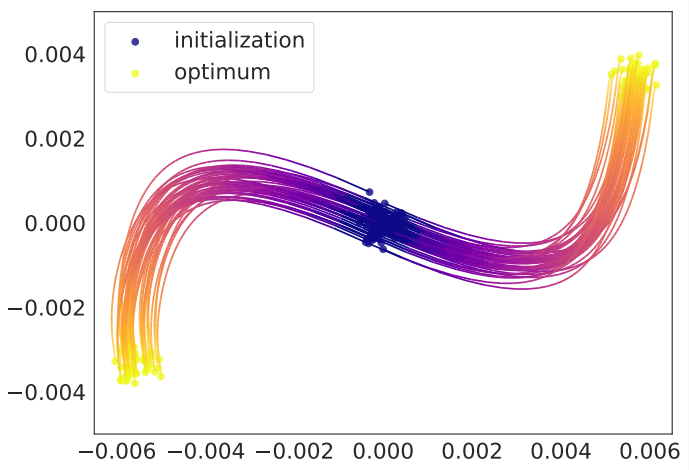
- Satisfied by $\boldsymbol{w}_i \sim \mathcal{N}(0, \mathbf{I}/d^{1-\epsilon})$. Neurons stay close to initialization.
- Training can be *linearized around the* ***initialization***.

Vanishing and Non-vanishing Initialization

Vanishing Initialization

- Model is asymptotically equivalent to that of a *two-layer linear network*.

Remark: smooth activation is required due to 1st order Taylor expansion.



(a) vanishing initialization.

Non-vanishing Initialization

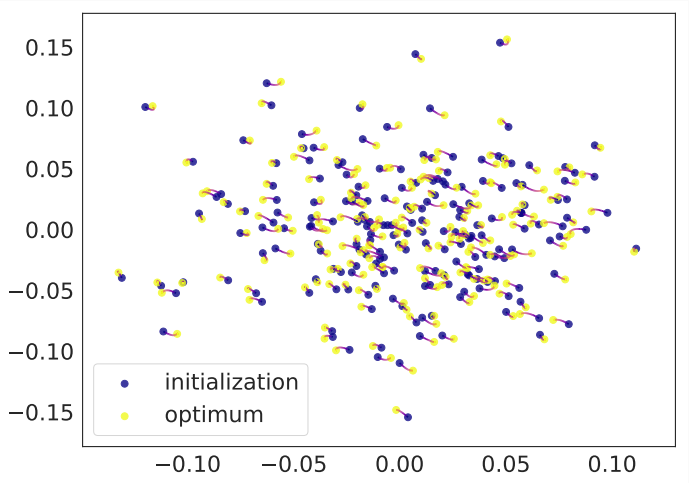
- Model described by the *neural tangent kernel*: $f(\boldsymbol{x}) \approx (\boldsymbol{\theta}_t - \boldsymbol{\theta}_0)^\top \nabla_{\boldsymbol{\theta_0}} f_0(\boldsymbol{x})$.

Remark: "doubling trick" to ensure $f_0(\boldsymbol{x}) = 0$.

Asymptotic equivalent of NTK:

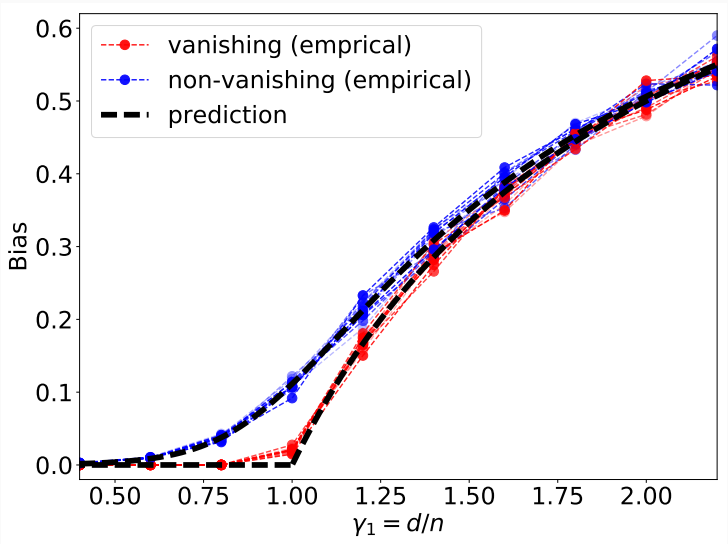
$$\boldsymbol{K} \approx b_0 \boldsymbol{X}\boldsymbol{X}^\top + b_1 \boldsymbol{I}_n,$$

where b_0, b_1 are obtained from orthogonal decomposition of the activation.

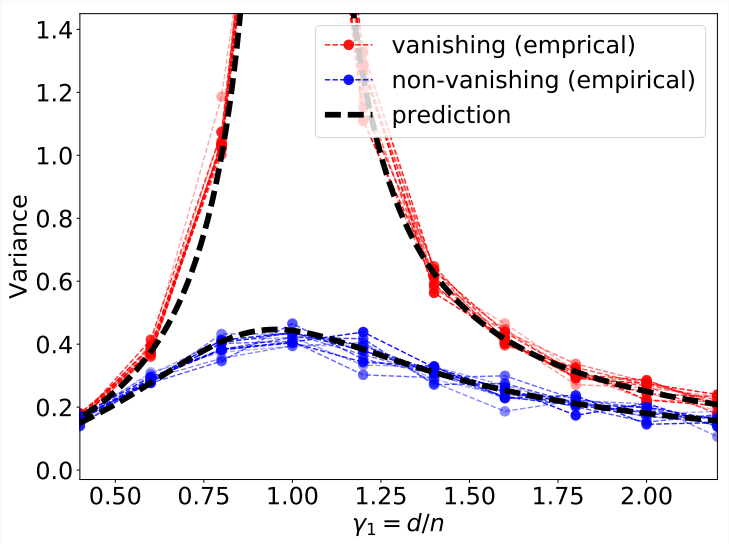


(b) non-vanishing initialization.

Exact Risk for Two Initializations



(a) bias (sigmoid).



(b) variance (sigmoid).

Note: individual dotted lines are different $\gamma_2 = h/n$, which **does not affect the risk**.

- For both initializations, population risk is **independent to** γ_2, i.e. double descent does not occur as a result of overparameterization.
- Two initializations lead to models with **contrasting properties**: large initialization results in higher bias but lower variance.

Discussion and Conclusion

Conclusion: "double descent" in neural networks is *more nuanced* compared to linear models (i.e. minimum norm interpolants).

- Optimizing **different layers** of the model results in different behaviors.
- **Scale of initialization** leads to different inductive bias.
- Proportional limit *may not* be the right regime to analyze double descent in neural networks?

Future Directions:

- Relax assumptions (e.g. universality of random matrix results).
- Consider different initializations (e.g. the mean-field $1/h$ scaling).
- Characterize the impact of loss function and regularization (both explicit and algorithmic).

Additional Reference

- Krogh, A. and Hertz, J. A., 1992. A simple weight decay can improve generalization.
- Cheng, X. and Singer, A., 2013. The spectrum of random inner-product kernel matrices.
- Mei, S., Montanari, A. and Nguyen, P.M., 2018. A mean field view of the landscape of two-layer neural networks.
- Jacot, A., Gabriel, F. and Hongler, C., 2018. Neural tangent kernel: Convergence and generalization in neural networks.
- Xu, J. and Hsu D., 2019. On the number of variables to use in principal component regression.
- Mei, S. and Montanari, A., 2019. The generalization error of random features regression: Precise asymptotics and double descent curve.