Limitations of Lazy Training of Two-layers Neural Networks

Behrooz Ghorbani [1,*] Song Mei [2,*] Theodor Misiakiewicz [3,*] Andrea Montanari [1,3]

[1]Department of Electrical Engineering, Stanford University [2]ICME, Stanford Unifversity [3]Department of Statistics, Stanford University

[*]Equal contributions

Introduction

Consider the function class of **two-layers neural networks**

$$\mathcal{F}_{\mathsf{NN},N} = \Big\{ f(\boldsymbol{x}) = \sum_{i=1}^{N} a_i \sigma(\langle \boldsymbol{w}_i, \boldsymbol{x}\rangle) : \ a_i \in \mathbb{R},\ \boldsymbol{w}_i \in \mathbb{R}^d \Big\}.$$

- Linearization around (random) parameter $\boldsymbol{\theta}_i^0 = (a_i^0, \boldsymbol{w}_i^0)$
  $$f_{\mathsf{NN}}(\boldsymbol{x};\boldsymbol{\theta}) \approx f_{\mathsf{NN}}(\boldsymbol{x};\boldsymbol{\theta}^0) + \langle \boldsymbol{\theta} - \boldsymbol{\theta}^0, \nabla_{\boldsymbol{\theta}} f_{\mathsf{NN}}(\boldsymbol{x};\boldsymbol{\theta}^0)\rangle$$
- Lazy training [1]: under certain initialization and for a large number of parameters N, the parameters $\boldsymbol{\theta}$ learned by SGD stay close to the initialization $\boldsymbol{\theta}^0$ and the above approximation is accurate [2].
- In this regime, learning the neural network is essentially the same as learning the linearized part:
  $$f_{\mathsf{NN}}(\boldsymbol{x};\boldsymbol{\theta}) \approx 0 + \underbrace{\sum_{i=1}^{N} \Delta a_i \sigma(\langle \boldsymbol{w}_i^0, \boldsymbol{x}\rangle)}_{\text{Second layer linearization}} + \underbrace{\sum_{i=1}^{N} a_i^0 \sigma'(\langle \boldsymbol{w}_i^0, \boldsymbol{x}\rangle)\langle \Delta \boldsymbol{w}_i, \boldsymbol{x}\rangle}_{\text{First layer linearization}}$$

We consider the following two function classes which we will refer to as the random feature model (RF) [6], and the neural tangent model (NT) [4]: for $\boldsymbol{w}_i \overset{i.i.d.}{\sim} \mathsf{N}(0, \boldsymbol{I}_d)$,

$$\mathcal{F}_{\mathsf{RF},N}(\boldsymbol{W}) = \Big\{ f_N(\boldsymbol{x}) = \sum_{i=1}^{N} a_i \sigma(\langle \boldsymbol{w}_i, \boldsymbol{x}\rangle) : \ a_i \in \mathbb{R} \Big\},$$

$$\mathcal{F}_{\mathsf{NT},N}(\boldsymbol{W}) = \Big\{ f_N(\boldsymbol{x}) = \sum_{i=1}^{N} \langle \boldsymbol{a}_i, \boldsymbol{x}\rangle \sigma'(\langle \boldsymbol{w}_i, \boldsymbol{x}\rangle) : \ \boldsymbol{a}_i \in \mathbb{R}^d \Big\}.$$

Blue: random and fixed. Red: parameters to be optimized.

Questions

- Do RF/NT models provide a good approximation to effectively trained NN (e.g. by SGD)?
- Do RF/NT learn good representations of the data?

We provide two simple settings where we can fully characterize the behavior of RF, NT and SGD-trained NN. In these settings, these two questions admit negative answers.

The prediction risk achieved within any of the regimes $\mathsf{M} \in \{\mathsf{RF}, \mathsf{NT}, \mathsf{NN}\}$ is defined by

$$R_{\mathsf{M},N}(f_*) = \min_{\hat{f} \in \mathcal{F}_{\mathsf{M},N}(\boldsymbol{W})} \mathbb{E}\Big\{ (f_*(\boldsymbol{x}) - \hat{f}(\boldsymbol{x}))^2 \Big\},$$

$$R_{\mathsf{NN},N}(f_*; \ell, \varepsilon) = \mathbb{E}\Big\{ (f_*(\boldsymbol{x}) - \hat{f}(\boldsymbol{x}; \ell, \varepsilon))^2 \Big\},$$

where $\hat{f}(\,\cdot\,; \ell, \varepsilon)$ is the neural network produced by ℓ steps of stochastic gradient descent (SGD) where each sample is used once, and the stepsize is set to ε

Quadratic Functions (QF)

Setting: $\boldsymbol{x}_i \sim \mathsf{N}(\boldsymbol{0}, \boldsymbol{I}_d)$ and responses

$$y_i = f_*(\boldsymbol{x}_i) \equiv b_0 + \langle \boldsymbol{x}_i, \boldsymbol{B}\boldsymbol{x}_i\rangle, \ \text{with } \boldsymbol{B} \succeq 0.$$

We take a quadratic activation $\sigma(u) = u^2 + c_0$ and consider the high-dimensional regime: $N, d \to \infty$, $N/d \to \rho \in (0, \infty)$.

Results [5]:

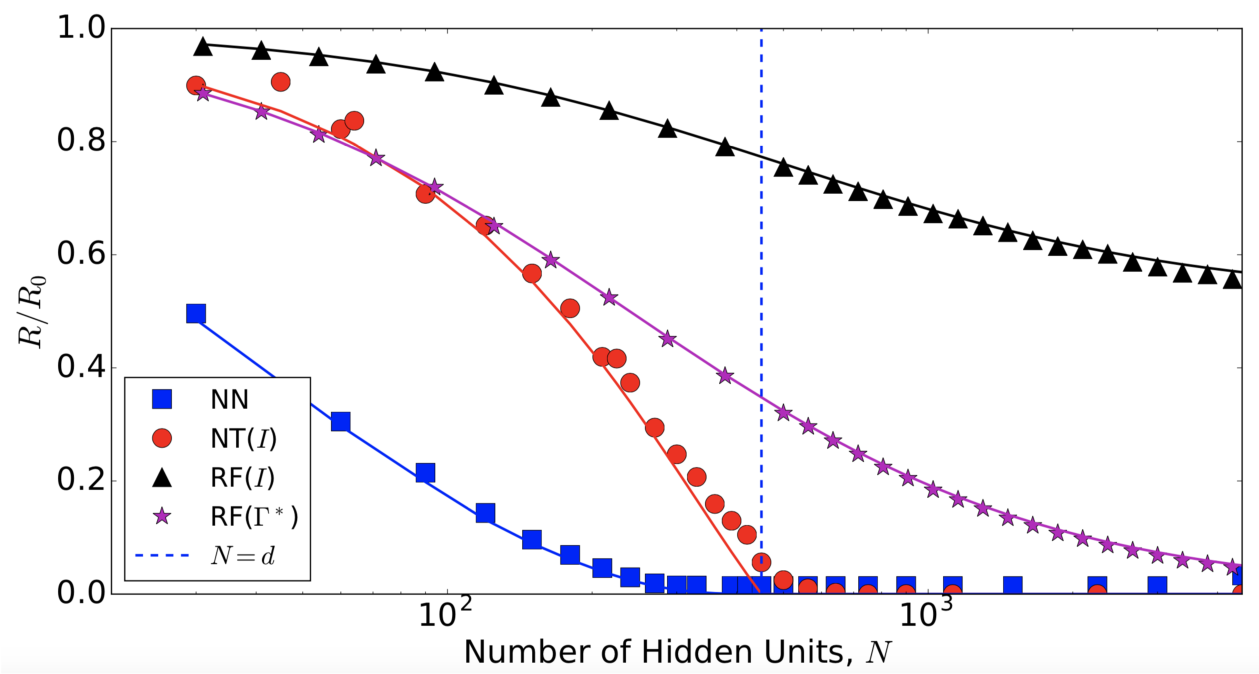

Figure 1: Prediction (test) error in fitting a quadratic function in $d = 450$ dimensions, as a function of the number of neurons N. Lines are analytical predictions obtained in this paper [5], and dots are empirical results.

- Naive RF/NT do not learn good representations of the data.
- SGD-trained NN learns the most important eigendirections of f_* and fits them, hence surpassing the NT model which remains confined to a random subspace spanned by $\boldsymbol{w}_i$.
- There exists an arbitrary large gap between the SGD-trained networks and the neural tangent model.

Neural networks are superior to linearized model such as RF and NT, because they can learn a good representation of the data.

Mixture of Gaussians

Setting: $y_i = \pm 1$ with equal probability $1/2$, and $\boldsymbol{x}_i | y_i = +1 \sim \mathsf{N}(0, \boldsymbol{I}_d + \boldsymbol{\Delta})$, $\boldsymbol{x}_i | y_i = -1 \sim \mathsf{N}(0, \boldsymbol{I}_d - \boldsymbol{\Delta})$. Take $\sigma(u) = u^2 + c_0$ and $\boldsymbol{w}_i \sim \mathsf{N}(0, \boldsymbol{I}_d/d)$.

$$R_{\mathsf{M},N}(\mathbb{P}_{\boldsymbol{I},\boldsymbol{\Delta}}) \approx \begin{cases} \dfrac{1}{1 + \frac{\rho}{1+2\rho} \cdot \frac{\hat{r}(\boldsymbol{\Delta})^2}{2d}} & \text{for } \mathsf{M} = \mathsf{RF}, \\[2ex] \dfrac{1}{1 + \kappa(\rho, \boldsymbol{\Delta}) \|\boldsymbol{\Delta}\|_F^2/2} & \text{for } \mathsf{M} = \mathsf{NT}, \\[2ex] \dfrac{1}{1 + \sum_{i=1}^{N \wedge d} \lambda_i(\boldsymbol{\Delta})^2/2} & \text{for } \mathsf{M} = \mathsf{NN}. \end{cases}$$

- See Figure 2 for analytical and empirical results.
- We recover a similar behavior as in the QF model.
- Note that the Bayes error is not achieved in this model.
- We do not show convergence of SGD in this setting but we expect a similar result to the QF model to hold.

Analytical Predictions for QF

Random features model

Theorem 1 ([5]) *Take $\sigma(x) = x^2 - 1$, $\boldsymbol{w}_i \sim \mathsf{N}(0, \boldsymbol{\Gamma})$. Then, as $N, d \to \infty$ with $N/d \to \rho$*

$$R_{\mathsf{RF},N}(f_*) = \|f_*\|_{L_2}^2 \left(1 - \frac{\rho d \langle \boldsymbol{B}, \boldsymbol{\Gamma}\rangle^2}{\|\boldsymbol{B}\|_F^2 \big(1 + \rho d \|\boldsymbol{\Gamma}\|_F^2\big)} + o_{d,\mathbb{P}}(1) \right).$$

- See [5] for the Theorem for general activation function σ.
- The risk highly depends on the weight distribution.
- In particular for any activation function,
  $$\lim_{\rho \to \infty} \lim_{d \to \infty, N/d \to \rho} \frac{R_{\mathsf{RF},N}(f_*)}{\|f_*\|_{L_2}^2} = \lim_{d \to \infty} \left(1 - \frac{\langle \boldsymbol{B}, \boldsymbol{\Gamma}\rangle^2}{\|\boldsymbol{B}\|_F^2 \|\boldsymbol{\Gamma}\|_F^2} \right).$$
  The risk vanishes only if $\boldsymbol{\Gamma} \propto \boldsymbol{B}$ is chosen perfectly and $\rho \to \infty$. The asymptotic risk is independent of the non-linearity!

Neural Tangent model

Theorem 2 ([5]) *Take $\sigma(x) = x^2$, $\boldsymbol{w}_i \sim \mathsf{N}(0, \boldsymbol{I}_d/d)$. Then, as $N, d \to \infty$ with $N/d \to \rho$*

$$\frac{\mathbb{E}_{\boldsymbol{W}}[R_{\mathsf{NT},N}(f_*)]}{\|f_*\|_{L^2}^2} = \Big\{ (1-\rho)_+^2 + \rho(1-\rho)_+ \frac{\mathsf{Tr}(\boldsymbol{B})^2}{d\,\|\boldsymbol{B}\|_F^2} + o_d(1) \Big\}.$$

- For $N < d$, NT fits f_* along a random subspace determined by the weights $\boldsymbol{w}_i$ (not the most important subspace).
- For $N \geq d$, weights span the whole space (vanishing risk).

Fully-trained NN model

Theorem 3 ([5]) *Take $\sigma(x) = x^2$. Then, as $N, d \to \infty$ with $N/d \to \rho$*

$$\lim_{t \to \infty} \lim_{\varepsilon \to 0} \mathbb{P}\Big(\Big| R_{\mathsf{NN},N}(f_*; \ell = t/\varepsilon, \varepsilon) - R_{\mathsf{NN},N}(f_*) \Big| \geq \delta \Big) = 0,$$

$$R_{\mathsf{NN},N}(f_*) = 2 \sum_{i=N+1}^{d} \lambda_i(\boldsymbol{B})^2,$$

with $\lambda_1(\boldsymbol{B}) \geq \cdots \geq \lambda_d(\boldsymbol{B})$ ordered eigenvalues of $\boldsymbol{B}$.

- Here, we studied a one-pass version of SGD. The probability is over the random initialization $\boldsymbol{W}^0$ and the samples.
- The global convergence is proved by showing convergence of SGD to the gradient flow in the population risk and then proving a strict saddle property for the population risk.
- SGD-learned NN fits f_* along the most important subspace (the N principal eigendirections of $\boldsymbol{B}$).

How General are these Phenomena?

- The separation between NN and NT is established only for $N < d$. We expect the separation to generalize to $N \geq d$ by considering higher order polynomials: for third- or higher-order polynomials, NT does not achieve vanishing risk at any $\rho \in (0, \infty)$ (see [3]).
- While we are only able to provide theory for NN and NT for quadratic activation, we performed extensive experiments with other non-linearities. See Figure 3 for fitting a quadratic function with ReLu activation. In particular, the positive gap between NN and NT is still present for $N < d$.

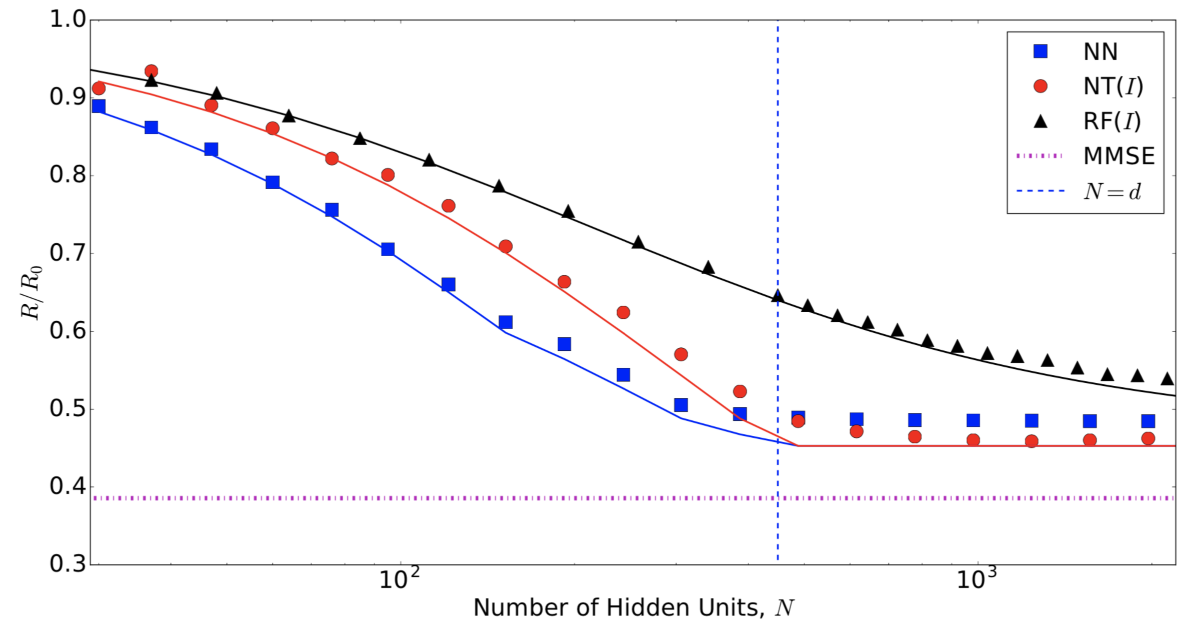

Figure 2: Prediction (test) error in fitting a mixture of Gaussians in $d = 450$ dimensions, as a function of N. Lines are analytical predictions obtained in this paper [5], and dots are empirical results. Dotted line is the Bayes error.

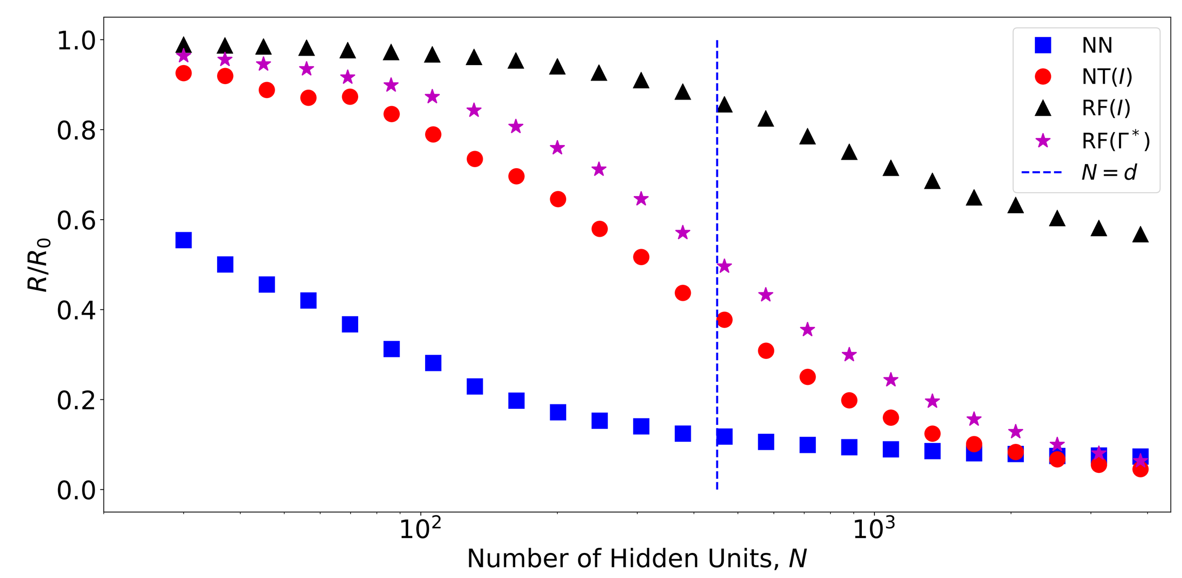

Figure 3: Empirical prediction (test) error in fitting a quadratic function in $d = 450$ dimensions with ReLu activation, as a function of N.

Bibliography

[1] L. Chizat and F. Bach.
A note on lazy training in supervised differentiable programming.
arXiv:1812.07956, 2018.

[2] S. S. Du, J. D. Lee, H. Li, L. Wang, and X. Zhai.
Gradient descent finds global minima of deep neural networks.
arXiv:1811.03804, 2018.

[3] B. Ghorbani, S. Mei, T. Misiakiewicz, and A. Montanari.
Linearized two-layers neural networks in high dimension.
arXiv:1904.12191, 2019.

[4] A. Jacot, F. Gabriel, and C. Hongler.
Neural tangent kernel: Convergence and generalization in neural networks.
In *Advances in neural information processing systems*, pages 8571–8580, 2018.

[5] S. Mei, T. Misiakiewicz, B. Ghorbani, and A. Montanari.
Limitations of lazy training of two-layers neural network.
In *Advances in Neural Information Processing Systems*, pages 9108–9118, 2019.

[6] A. Rahimi and B. Recht.
Random features for large-scale kernel machines.
In *Advances in neural information processing systems*, pages 1177–1184, 2008.