Generalization of linearized neural networks: staircase decay and double descent

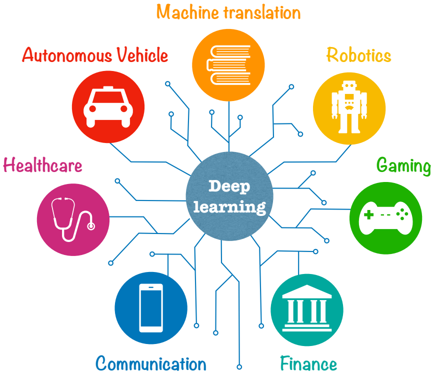
Song Mei

UC Berkeley

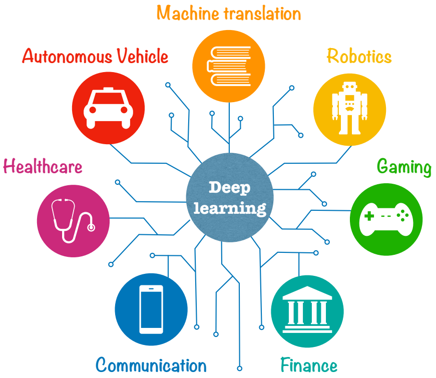
July 23, 2020

Department of Mathematics, HKUST

Deep Learning Revolution



Deep Learning Revolution



“ACM named Yoshua Bengio, Geoffrey Hinton, and Yann LeCun recipients of the 2018 ACM A.M. Turing Award for conceptual and engineering breakthroughs that have made deep neural networks a critical component of computing. ”

But theoretically?

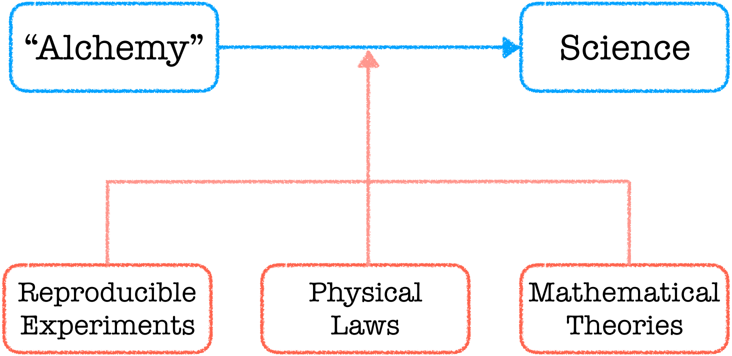
But theoretically?

WHEN and WHY does deep learning work?

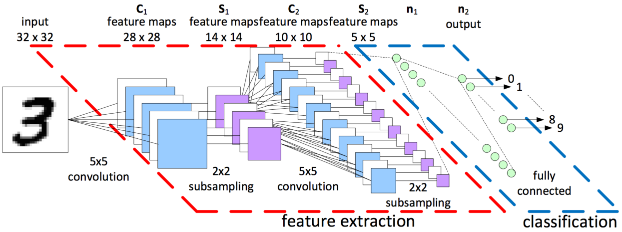
Call for Theoretical understandings

“Alchemy”

Call for Theoretical understandings



What don't we understand?



What don't we understand?

Empirical Surprises [Zhang, et.al, 2015]:

- Over-parameterization: # parameters $\gg$ # training samples.
- Non-convexity.
- Efficiently fit all the training samples using SGD.
- Generalize well on test samples.

What don't we understand?

Empirical Surprises [Zhang, et.al, 2015]:

- Over-parameterization: # parameters $\gg$ # training samples.
- Non-convexity.
- Efficiently fit all the training samples using SGD.
- Generalize well on test samples.

Mathematical Challenges

Non-convexity	$\leftrightarrow$	Why efficient optimization?
Over-parameterization	$\leftrightarrow$	Why effective generalization?

A gentle introduction to

Linearization theory of neural networks

Linearized neural networks (neural tangent model)

- Multi-layers neural network $f(\boldsymbol{x};\boldsymbol{\theta})$, $\boldsymbol{x} \in \mathbb{R}^d$, $\boldsymbol{\theta} \in \mathbb{R}^N$

$$f(\boldsymbol{x};\boldsymbol{\theta}) = \boldsymbol{W}_L \sigma(\cdots \boldsymbol{W}_2 \sigma(\boldsymbol{W}_1 \boldsymbol{x})).$$

- Linearization around (random) parameter θ_0

$$f(x;\theta) = f(x;\theta_0) + \underline{\langle \theta - \theta_0, \nabla_{\boldsymbol{\theta}} f(x;\theta_0)\rangle} + o(\|\theta - \theta_0\|_2).$$

- Neural tangent model: the linear part of f

$$f_{\mathsf{NT}}(x;\beta,\theta_0) = \langle \beta, \nabla_{\boldsymbol{\theta}} f(x;\theta_0)\rangle.$$

[Jacot, Gabriel, Hongler, 2018] [Chizat, Bach, 2018b]

Linearized neural networks (neural tangent model)

- Multi-layers neural network $f(\boldsymbol{x};\boldsymbol{\theta})$, $\boldsymbol{x} \in \mathbb{R}^d$, $\boldsymbol{\theta} \in \mathbb{R}^N$

- Linearization around (random) parameter $\boldsymbol{\theta}_0$

$$f(\boldsymbol{x};\boldsymbol{\theta}) = f(\boldsymbol{x};\boldsymbol{\theta}_0) + \underline{\langle \boldsymbol{\theta} - \boldsymbol{\theta}_0, \nabla_{\boldsymbol{\theta}} f(\boldsymbol{x};\boldsymbol{\theta}_0)\rangle} + o(\|\boldsymbol{\theta} - \boldsymbol{\theta}_0\|_2).$$

- Neural tangent model: the linear part of f

$$f_{\mathsf{NT}}(\boldsymbol{x};\boldsymbol{\beta},\boldsymbol{\theta}_0) = \langle \boldsymbol{\beta}, \nabla_{\boldsymbol{\theta}} f(\boldsymbol{x};\boldsymbol{\theta}_0)\rangle.$$

[Jacot, Gabriel, Hongler, 2018] [Chizat, Bach, 2018b]

Linearized neural networks (neural tangent model)

- Multi-layers neural network $f(\boldsymbol{x}; \boldsymbol{\theta})$, $\boldsymbol{x} \in \mathbb{R}^d$, $\boldsymbol{\theta} \in \mathbb{R}^N$

- Linearization around (random) parameter $\boldsymbol{\theta}_0$

$$f(\boldsymbol{x}; \boldsymbol{\theta}) = f(\boldsymbol{x}; \boldsymbol{\theta}_0) + \underline{\langle \boldsymbol{\theta} - \boldsymbol{\theta}_0, \nabla_{\boldsymbol{\theta}} f(\boldsymbol{x}; \boldsymbol{\theta}_0) \rangle} + o(\|\boldsymbol{\theta} - \boldsymbol{\theta}_0\|_2).$$

- Neural tangent model: the linear part of f

$$f_{\mathsf{NT}}(\boldsymbol{x}; \boldsymbol{\beta}, \boldsymbol{\theta}_0) = \langle \boldsymbol{\beta}, \nabla_{\boldsymbol{\theta}} f(\boldsymbol{x}; \boldsymbol{\theta}_0) \rangle.$$

[Jacot, Gabriel, Hongler, 2018] [Chizat, Bach, 2018b]

Linear regression over random features

- NT model: the linear part of f

$$f_{\mathsf{NT}}(\boldsymbol{x}; \boldsymbol{\beta}, \boldsymbol{\theta}_0) = \langle \boldsymbol{\beta}, \phi(\boldsymbol{x}) \rangle = \langle \boldsymbol{\beta}, \nabla_{\boldsymbol{\theta}} f(\boldsymbol{x}; \boldsymbol{\theta}_0) \rangle.$$

- (Random) feature map: $\phi(\cdot) = \nabla_{\boldsymbol{\theta}} f(\cdot; \boldsymbol{\theta}_0) : \mathbb{R}^d \to \mathbb{R}^N$.
- Training dataset: $(\mathcal{X}, \mathcal{Y}) = (\boldsymbol{x}_i, y_i)_{i \in [n]}$.
- Gradient flow dynamics:

$$\frac{\mathrm{d}}{\mathrm{d}t} \boldsymbol{\beta}^t = -\nabla_{\boldsymbol{\beta}} \hat{\mathbb{E}}[(y - f_{\mathsf{NT}}(\boldsymbol{x}; \boldsymbol{\beta}^t, \boldsymbol{\theta}_0))^2], \qquad \boldsymbol{\beta}^0 = \mathbf{0}.$$

- Linear convergence: $\boldsymbol{\beta}^t \to \hat{\boldsymbol{\beta}} = \phi(\mathcal{X})^\dagger \mathcal{Y}$.

Linear regression over random features

- NT model: the linear part of f

$$f_{\mathsf{NT}}(\boldsymbol{x}; \boldsymbol{\beta}, \boldsymbol{\theta}_0) = \langle \boldsymbol{\beta}, \phi(\boldsymbol{x}) \rangle = \langle \boldsymbol{\beta}, \nabla_{\boldsymbol{\theta}} f(\boldsymbol{x}; \boldsymbol{\theta}_0) \rangle.$$

- (Random) feature map: $\phi(\cdot) = \nabla_{\boldsymbol{\theta}} f(\cdot; \boldsymbol{\theta}_0) : \mathbb{R}^d \to \mathbb{R}^N$.
- Training dataset: $(\mathcal{X}, \mathcal{Y}) = (\boldsymbol{x}_i, y_i)_{i \in [n]}$.
- Gradient flow dynamics:

$$\frac{\mathrm{d}}{\mathrm{d}t} \boldsymbol{\beta}^t = -\nabla_{\boldsymbol{\beta}} \hat{\mathbb{E}}[(y - f_{\mathsf{NT}}(\boldsymbol{x}; \boldsymbol{\beta}^t, \boldsymbol{\theta}_0))^2], \qquad \boldsymbol{\beta}^0 = \boldsymbol{0}.$$

- Linear convergence: $\boldsymbol{\beta}^t \to \hat{\boldsymbol{\beta}} = \phi(\mathcal{X})^\dagger \mathcal{Y}$.

Linear regression over random features

- NT model: the linear part of f

$$f_{\mathsf{NT}}(\boldsymbol{x}; \boldsymbol{\beta}, \boldsymbol{\theta}_0) = \langle \boldsymbol{\beta}, \phi(\boldsymbol{x}) \rangle = \langle \boldsymbol{\beta}, \nabla_{\boldsymbol{\theta}} f(\boldsymbol{x}; \boldsymbol{\theta}_0) \rangle.$$

- (Random) feature map: $\phi(\cdot) = \nabla_{\boldsymbol{\theta}} f(\cdot; \boldsymbol{\theta}_0) : \mathbb{R}^d \to \mathbb{R}^N$.
- Training dataset: $(\mathcal{X}, \mathcal{Y}) = (\boldsymbol{x}_i, y_i)_{i \in [n]}$.
- Gradient flow dynamics:

$$\frac{\mathrm{d}}{\mathrm{d}t} \boldsymbol{\beta}^t = -\nabla_{\boldsymbol{\beta}} \hat{\mathbb{E}}[(y - f_{\mathsf{NT}}(\boldsymbol{x}; \boldsymbol{\beta}^t, \boldsymbol{\theta}_0))^2], \qquad \boldsymbol{\beta}^0 = \mathbf{0}.$$

- Linear convergence: $\boldsymbol{\beta}^t \to \hat{\boldsymbol{\beta}} = \phi(\mathcal{X})^\dagger \mathcal{Y}$.

Linear regression over random features

- NT model: the linear part of f

$$f_{\mathsf{NT}}(\boldsymbol{x}; \boldsymbol{\beta}, \boldsymbol{\theta}_0) = \langle \boldsymbol{\beta}, \phi(\boldsymbol{x}) \rangle = \langle \boldsymbol{\beta}, \nabla_{\boldsymbol{\theta}} f(\boldsymbol{x}; \boldsymbol{\theta}_0) \rangle.$$

- (Random) feature map: $\phi(\cdot) = \nabla_{\boldsymbol{\theta}} f(\cdot; \boldsymbol{\theta}_0) : \mathbb{R}^d \to \mathbb{R}^N$.

- Training dataset: $(\mathcal{X}, \mathcal{Y}) = (\boldsymbol{x}_i, y_i)_{i \in [n]}$.

- Gradient flow dynamics:

$$\frac{\mathrm{d}}{\mathrm{d}t} \boldsymbol{\beta}^t = -\nabla_{\boldsymbol{\beta}} \hat{\mathbb{E}}[(y - f_{\mathsf{NT}}(\boldsymbol{x}; \boldsymbol{\beta}^t, \boldsymbol{\theta}_0))^2], \qquad \boldsymbol{\beta}^0 = \mathbf{0}.$$

- Linear convergence: $\boldsymbol{\beta}^t \to \hat{\boldsymbol{\beta}} = \phi(\mathcal{X})^{\dagger} \mathcal{Y}$.

Linear regression over random features

- NT model: the linear part of f

$$f_{\mathsf{NT}}(\boldsymbol{x};\boldsymbol{\beta},\boldsymbol{\theta}_0) = \langle \boldsymbol{\beta}, \phi(\boldsymbol{x})\rangle = \langle \boldsymbol{\beta}, \nabla_{\boldsymbol{\theta}} f(\boldsymbol{x};\boldsymbol{\theta}_0)\rangle.$$

- (Random) feature map: $\phi(\cdot) = \nabla_{\boldsymbol{\theta}} f(\cdot;\boldsymbol{\theta}_0) : \mathbb{R}^d \to \mathbb{R}^N$.

- Training dataset: $(\mathcal{X},\mathcal{Y}) = (\boldsymbol{x}_i, y_i)_{i\in[n]}$.

- Gradient flow dynamics:

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\beta}^t = -\nabla_{\boldsymbol{\beta}}\hat{\mathbb{E}}[(y - f_{\mathsf{NT}}(\boldsymbol{x};\boldsymbol{\beta}^t,\boldsymbol{\theta}_0))^2], \qquad \boldsymbol{\beta}^0 = \mathbf{0}.$$

- Linear convergence: $\boldsymbol{\beta}^t \to \hat{\boldsymbol{\beta}} = \phi(\mathcal{X})^\dagger \mathcal{Y}$.

Neural network $\approx$ Neural tangent

Theorem [Jacot, Gabriel, Hongler, 2018] (Informal)

Consider neural networks $f^N(\boldsymbol{x};\boldsymbol{\theta})$ with number of neurons N, and consider

$$
\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\theta}^t &= -\nabla_{\boldsymbol{\theta}}\hat{\mathbb{E}}[(y - f^N(\boldsymbol{x};\boldsymbol{\theta}^t))^2], & \boldsymbol{\theta}^0 &= \boldsymbol{\theta}_0,\\
\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\beta}^t &= -\nabla_{\boldsymbol{\beta}}\hat{\mathbb{E}}[(y - f^N_{\mathsf{NT}}(\boldsymbol{x};\boldsymbol{\beta}^t,\boldsymbol{\theta}_0))^2], & \boldsymbol{\beta}^0 &= \mathbf{0}.
\end{aligned}
$$

Under proper (random) initialization, we have a.s.

$$
\lim_{N\to\infty}|f^N(\boldsymbol{x};\boldsymbol{\theta}^t) - f^N_{\mathsf{NT}}(\boldsymbol{x};\boldsymbol{\beta}^t)| = 0.
$$

Optimization success

Gradient flow of training loss of NN converges to global min ...
... with over-parameterization and proper initialization

[Jacot, Gabriel, Hongler, 2018], [Du, Zhai, Poczos, Singh, 2018], [Du, Lee, Li, Wang, Zhai, 2018], [Allen-Zhu, Li, Song 2018], [Zou, Cao, Zhou, Gu, 2018], [Oymak, Soltanolkotabi, 2018] [Chizat, Bach, 2018b], ...

Optimization success

Gradient flow of training loss of NN converges to global min ...
... with over-parameterization and proper initialization

[Jacot, Gabriel, Hongler, 2018], [Du, Zhai, Poczos, Singh, 2018], [Du, Lee, Li, Wang, Zhai, 2018], [Allen-Zhu, Li, Song 2018], [Zou, Cao, Zhou, Gu, 2018], [Oymak, Soltanolkotabi, 2018] [Chizat, Bach, 2018b], ...

Does linearization fully explain the success of neural networks?

Optimization success

Gradient flow of training loss of NN converges to global min ...
... with over-parameterization and proper initialization

[Jacot, Gabriel, Hongler, 2018], [Du, Zhai, Poczos, Singh, 2018], [Du, Lee, Li, Wang, Zhai, 2018], [Allen-Zhu, Li, Song 2018], [Zou, Cao, Zhou, Gu, 2018], [Oymak, Soltanolkotabi, 2018] [Chizat, Bach, 2018b], ...

Does linearization fully explain the success of neural networks?

Our answer is No

Generalization

Empirically, the generalization of NT models are not as good as NN

Table: Cifar10 experiments

Architecture	Classification error
CNN	4%-
(1) CNTK	23%
(2) CNTK	11%
(3) Compositioal Kernel	10%

(1) [Arora, Du, Hu, Li, Salakhutdinov, Wang, 2019],

(2) [Li, Wang, Yu, Du, Hu, Salakhutdinov, Arora, 2019],

(3) [Shankar, Fang, Guo, Fridovich-Keil, Schmidt, Ragan-Kelley, Recht, 2020].

Performance gap: NN versus NT

Two-layers neural network

$$f_N(\boldsymbol{x};\boldsymbol{\Theta}) = \sum_{i=1}^{N} a_i \sigma(\langle \boldsymbol{w}_i, \boldsymbol{x}\rangle), \quad \boldsymbol{\Theta} = (a_1, \boldsymbol{w}_1, \dots, a_N, \boldsymbol{w}_N).$$

- Input vector $\boldsymbol{x} \in \mathbb{R}^d$.
- Bottom layer weights $\boldsymbol{w}_i \in \mathbb{R}^d$, $i = 1, 2, \dots, N$.
- Top layer weights $a_i \in \mathbb{R}$, $i = 1, 2, \dots, N$.

Linearization around initialization

Linearization

$$f_N(\boldsymbol{x};\boldsymbol{\Theta}) = f_N(\boldsymbol{x};\boldsymbol{\Theta}^0) + \underbrace{\sum_{i=1}^N \Delta a_i \sigma(\langle \boldsymbol{w}_i^0, \boldsymbol{x}\rangle)}_{\text{Top layer linearization}} + \underbrace{\sum_{i=1}^N a_i^0 \sigma'(\langle \boldsymbol{w}_i^0, \boldsymbol{x}\rangle)\langle \Delta \boldsymbol{w}_i, \boldsymbol{x}\rangle}_{\text{Bottom layer linearization}} + o(\cdot).$$

Linearized neural network: $(\boldsymbol{w}_i \sim \mathrm{Unif}(\mathbb{S}^{d-1}))$

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

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

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

[Rahimi, Recht, 2008] [Jacot, Gabriel, Hongler, 2018]

Approximation error

Data distribution:

$$\boldsymbol{x} \sim \mathrm{Unif}(\mathbb{S}^{d-1}(\sqrt{d})), \qquad f_\star \in L^2(\mathbb{S}^{d-1}(\sqrt{d})).$$

Minimum risk (approximation error):

$$R_{\mathsf{M},N}(f_\star) = \inf_{f \in \mathcal{F}_{\mathsf{M},N}(\boldsymbol{W})} \mathbb{E}_{\boldsymbol{x}}\Big[\Big(f_\star(\boldsymbol{x}) - f(\boldsymbol{x})\Big)^2\Big], \quad \mathsf{M} \in \{\mathsf{RF}, \mathsf{NT}\}.$$

Staircase decay

Random features regression

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

$$\boldsymbol{W} = (\boldsymbol{w}_i)_{i \in [N]} \sim_{i.i.d.} \mathrm{Unif}(\mathbb{S}^{d-1}).$$

Theorem (Ghorbani, Mei, Misiakiewicz, Montanari, 2019)

Assume $d^{\ell+\delta} \le N \le d^{\ell+1-\delta}$ and σ satisfies "generic condition", we have

$$\inf_{f \in \mathcal{F}_{\mathsf{RF},N}(\boldsymbol{W})} \mathbb{E}_{\boldsymbol{x}}[(f_\star(\boldsymbol{x}) - f(\boldsymbol{x}))^2] = \|\mathsf{P}_{>\ell} f_\star\|_{L^2}^2 + o_{d,\mathbb{P}}(\|f_\star\|_{L^2}^2).$$

$\mathsf{P}_{>\ell}$: *projection orthogonal to the space of degree-ℓ polynomials.*

With d^ℓ parameters, RF only fit a degree-ℓ polynomial.

Random features regression

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

$$\boldsymbol{W} = (\boldsymbol{w}_i)_{i \in [N]} \sim_{i.i.d.} \mathrm{Unif}(\mathbb{S}^{d-1}).$$

Theorem (Ghorbani, Mei, Misiakiewicz, Montanari, 2019)

Assume $d^{\ell+\delta} \le N \le d^{\ell+1-\delta}$ and σ satisfies "generic condition", we have

$$\inf_{f \in \mathcal{F}_{\mathsf{RF},N}(\boldsymbol{W})} \mathbb{E}_{\boldsymbol{x}}[(f_\star(\boldsymbol{x}) - f(\boldsymbol{x}))^2] = \|\mathsf{P}_{>\ell} f_\star\|_{L^2}^2 + o_{d,\mathbb{P}}(\|f_\star\|_{L^2}^2).$$

$\mathsf{P}_{>\ell}$: *projection orthogonal to the space of degree-ℓ polynomials.*

With d^ℓ parameters, RF only fit a degree-ℓ polynomial.

Similar result for NT

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

$$\boldsymbol{W} = (\boldsymbol{w}_i)_{i\in[N]} \sim_{i.i.d.} \mathrm{Unif}(\mathbb{S}^{d-1}).$$

Theorem (Ghorbani, Mei, Misiakiewicz, Montanari, 2019)

Assume $d^{\ell+\delta} \le N \le d^{\ell+1-\delta}$ and σ satisfies "generic condition", we have

$$\inf_{f\in\mathcal{F}_{\mathsf{NT},N}(\boldsymbol{W})} \mathbb{E}_{\boldsymbol{x}}[(f_\star(\boldsymbol{x}) - f(\boldsymbol{x}))^2] = \|\mathsf{P}_{>\ell+1} f_\star\|_{L^2}^2 + o_{d,\mathbb{P}}(\|f_\star\|_{L^2}^2).$$

$\mathsf{P}_{>\ell+1}$*: projection orthogonal to the space of degree-$(\ell+1)$ polynomials.*

With $d^{\ell+1}$ parameters, NT only fit a degree-$(\ell+1)$ polynomial.

Similar result for NT

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

$$\boldsymbol{W} = (\boldsymbol{w}_i)_{i\in[N]} \sim_{i.i.d.} \mathrm{Unif}(\mathbb{S}^{d-1}).$$

Theorem (Ghorbani, Mei, Misiakiewicz, Montanari, 2019)

Assume $d^{\ell+\delta} \le N \le d^{\ell+1-\delta}$ and σ satisfies "generic condition", we have

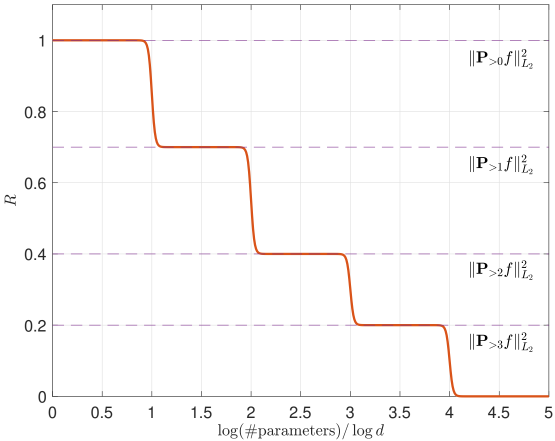
$$\inf_{f\in\mathcal{F}_{\mathsf{NT},N}(\boldsymbol{W})} \mathbb{E}_{\boldsymbol{x}}[(f_\star(\boldsymbol{x}) - f(\boldsymbol{x}))^2] = \|\mathsf{P}_{>\ell+1} f_\star\|_{L^2}^2 + o_{d,\mathbb{P}}(\|f_\star\|_{L^2}^2).$$

$\mathsf{P}_{>\ell+1}$*: projection orthogonal to the space of degree-*$(\ell+1)$ *polynomials.*

With $d^{\ell+1}$ parameters, NT only fit a degree-$(\ell+1)$ polynomial.

The staircase decay (a cartoon)

$$f = \mathsf{P}_0 f + \mathsf{P}_1 f + \mathsf{P}_2 f + \mathsf{P}_3 f + \cdots$$



Approximation gap

$$\text{Function } f : \mathbb{S}^{d-1} \to \mathbb{R},\ f(\boldsymbol{x}) = Q_k(x_1).$$
$$Q_k\text{: degree } k \text{ polynomial.}$$

- NT: $N = \Theta_d(d^{k-1})$;
- NN: $N = \Theta_d(1)$.
- A separation of approximation power.
- Neural network can potentially learn features adaptively.

Related work

Approximation error of two-layers NN and RF:
[Barron, 1993], [Mhaskar, 1996], [Maiorov, 1999], [Caponnetto, de Vito, 2007], [Rahimi, Recht, 2009], [Bach, 2017], [E, Ma, Wu, 2018] ...

Approx bound	$f_\star$ bounded norm	$f_\star \in L^2(\mathbb{R}^d) \cap (d_\star\text{-sparse})$
RF	$\|\|f_\star\|\|_{\mathcal{H}}^2/N$	$\Theta_N(1/N^{1/d})$
NN	$\|\|f_\star\|\|_{\mathcal{B}}^2/N$	$\Theta_N(1/N^{1/d_\star})$

Related work

Approximation error of two-layers NN and RF:
[Barron, 1993], [Mhaskar, 1996], [Maiorov, 1999], [Caponnetto, de Vito, 2007], [Rahimi, Recht, 2009], [Bach, 2017], [E, Ma, Wu, 2018] ...

Approx bound	$f_\star$ bounded norm	$f_\star \in L^2(\mathbb{R}^d) \cap (d_\star\text{-sparse})$
RF	$\|\|f_\star\|\|_{\mathcal{H}}^2 / N$	$\Theta_N(1/N^{1/d})$
NN	$\|\|f_\star\|\|_{\mathcal{B}}^2 / N$	$\Theta_N(1/N^{1/d_\star})$

Difference between:

New results	v.s.	Classical results
$N = d^k$ as $d \to \infty$,	v.s.	fixed d as $N \to \infty$,
Constant asymptotic error,	v.s.	Vanishing upper bound.

Related work

Approximation error of two-layers NN and RF:
[Barron, 1993], [Mhaskar, 1996], [Maiorov, 1999], [Caponnetto, de Vito, 2007], [Rahimi, Recht, 2009], [Bach, 2017], [E, Ma, Wu, 2018] ...

Approx bound	$f_\star$ bounded norm	$f_\star \in L^2(\mathbb{R}^d) \cap (d_\star\text{-sparse})$
RF	$\lVert f_\star \rVert_{\mathcal{H}}^2 / N$	$\Theta_N(1/N^{1/d})$
NN	$\lVert f_\star \rVert_{\mathcal{B}}^2 / N$	$\Theta_N(1/N^{1/d_\star})$

$N = d^k$ as $d \to \infty$, v.s. fixed d as $N \to \infty$,

Which asymptotics makes more sense?

$$d = 100, \qquad N = 10,000,000.$$

$$N = d^{3.5}, \qquad 1/N^{1/d} = 0.85.$$

Related work

Approximation error of two-layers NN and RF:
[Barron, 1993], [Mhaskar, 1996], [Maiorov, 1999], [Caponnetto, de Vito, 2007], [Rahimi, Recht, 2009], [Bach, 2017], [E, Ma, Wu, 2018] ...

Approx bound	$f_\star$ bounded norm	$f_\star \in L^2(\mathbb{R}^d) \cap (d_\star\text{-sparse})$
RF	$\|\|f_\star\|\|_{\mathcal{H}}^2 / N$	$\Theta_N(1/N^{1/d})$
NN	$\|\|f_\star\|\|_{\mathcal{B}}^2 / N$	$\Theta_N(1/N^{1/d_\star})$

$N = d^k$ as $d \to \infty$, v.s. fixed d as $N \to \infty$,

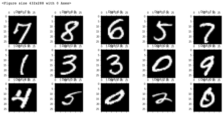
Which asymptotics makes more sense?

$$d_\star = 10, \qquad N = 10,000,000.$$

$$N = {d_\star}^7, \qquad 1/N^{1/d_\star} = 0.20.$$

Double descent

The motivating experiment

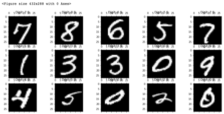


- MNIST: $(\boldsymbol{x}_i, y_i) \in \mathbb{R}^{784} \times [10]$, $i \in [50,000]$.
- Two-layers neural networks f_N:

$$f_N(\boldsymbol{x}; \boldsymbol{\theta}) = \sum_{j=1}^{N} a_j \sigma(\langle \boldsymbol{w}_j, \boldsymbol{x} \rangle).$$

- Square loss without regularization.
- Find a local minimizer, report training and test error.
- Perform a sequence of experiments for different N.
- Plot training and test error vs N.

The motivating experiment



- MNIST: $(\boldsymbol{x}_i, y_i) \in \mathbb{R}^{784} \times [10]$, $i \in [50,000]$.
- Two-layers neural networks f_N:

$$f_N(\boldsymbol{x}; \boldsymbol{\theta}) = \sum_{j=1}^{N} a_j \sigma(\langle \boldsymbol{w}_j, \boldsymbol{x} \rangle).$$

- Square loss without regularization.
- Find a local minimizer, report training and test error.
- Perform a sequence of experiments for different N.
- Plot training and test error vs N.

Increasing # parameters

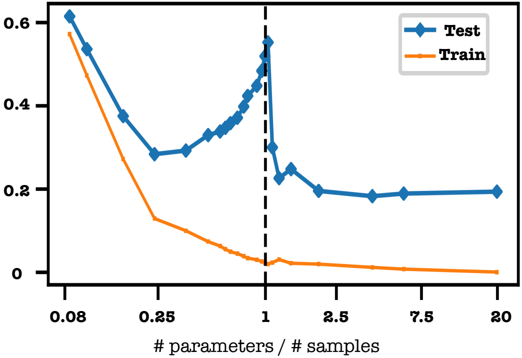


Figure: Experiments on MNIST. [Belkin, Hsu, Ma, Mandal, 2018].

Increasing # parameters

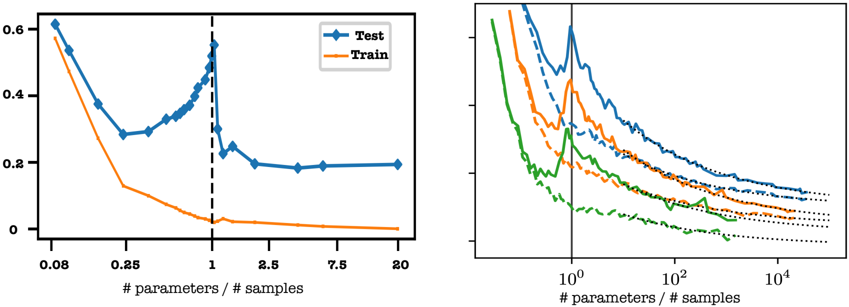
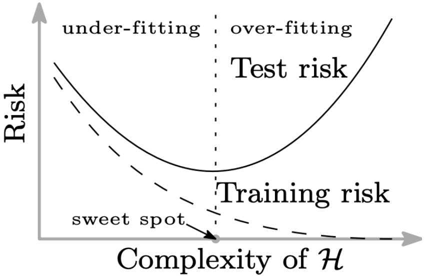


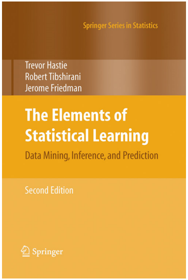
Figure: Experiments on MNIST. Left: [Belkin, Hsu, Ma, Mandal, 2018]. Right: [Spigler, Geiger, Ascoli, Sagun, Biroli, Wyart, 2018].

Similar phenomenon appeared in the literature [LeCun, Kanter, and Solla, 1991], [Krogh and Hertz, 1992], [Opper and Kinzel, 1995], [Neyshabur, Tomioka, Srebro, 2014], [Advani and Saxe, 2017].

U-shaped curve



(a) U-shaped "bias-variance" risk curve



[Belkin, Hsu, Ma, Mandal, 2018]

Double descent

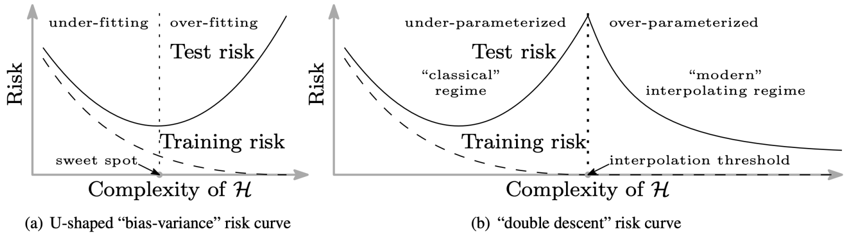


Figure: A cartoon by [Belkin, Hsu, Ma, Mandal, 2018].

- ✓ Peak at the interpolation threshold.
- ✓ Monotone decreasing in the overparameterized regime.
- ✓ Global minimum when the number of parameters is infinity.

Complementary instead of contradictory

U-shaped curve

Test error vs model complexity that tightly controls generalization.

Examples: ℓ_2 norm in linear model, "k" in k nearest-neighbors.

Double-descent

Test error vs number of parameters.

Examples: # parameters in NN.

In NN, # parameters $\neq$ model complexity that tightly controls generalization.

[Bartlett, 1997], [Bartlett and Mendelson, 2002]

Complementary instead of contradictory

U-shaped curve

Test error vs model complexity that tightly controls generalization.

Examples: ℓ_2 norm in linear model, “k” in k nearest-neighbors.

Double-descent

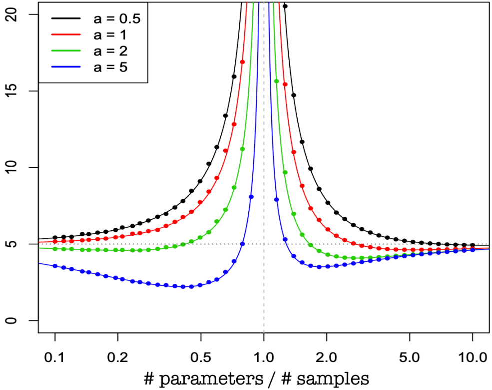
Test error vs number of parameters.

Examples: # parameters in NN.

In NN, # parameters $\neq$ model complexity that tightly controls generalization.

[Bartlett, 1997], [Bartlett and Mendelson, 2002]

Linear model with random covariates



By [Hastie, Montanari, Rosset, Tibshirani, 2019]. See also [Belkin, Hsu, Xu, 2019].

- Under-parameterized: $\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \sum_i (y_i - \langle \boldsymbol{x}_i, \boldsymbol{\beta} \rangle)^2$.
- Over-parameterized: $\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \|\boldsymbol{\beta}\|_2, \quad s.t. \quad y_i = \langle \boldsymbol{x}_i, \boldsymbol{\beta} \rangle + \varepsilon_i, \quad i \in [n]$.

Why singularity?

- Model: $\boldsymbol{x}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$, $y_i = \langle \mathbf{0}, \boldsymbol{x}_i \rangle + \varepsilon_i \sim \mathcal{N}(0, 1)$, $i \in [n]$.
- Test risk $\propto \mathbb{E}[||\hat{\beta} - \mathbf{0}||_2^2] \propto \mathbb{E}[||\boldsymbol{X}^\dagger \boldsymbol{y}||_2^2] \propto \mathbb{E}[\mathrm{tr}((\boldsymbol{X}^\top \boldsymbol{X})^\dagger)]$, where $\boldsymbol{X} \in \mathbb{R}^{n \times d}$.
- When $n \neq d$, $\boldsymbol{X}$ is well conditioned.
- When $n \approx d$, $\boldsymbol{X}$ is infinitely ill conditioned.

- The model has marginally enough parameters to interpolate all the data, hence it interpolates in an awkward way.
- To fit the noise, the coefficients $||\hat{\beta}||_2^2 = ||\boldsymbol{X}^\dagger \boldsymbol{y}||_2^2$ blows up.

[Bartlett, Long, Lugosi, Tsigler, 2019], [Muthukumar, Vodrahalli, Sahai, 2019]

Why singularity?

- Model: $\boldsymbol{x}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$, $y_i = \langle \mathbf{0}, \boldsymbol{x}_i \rangle + \varepsilon_i \sim \mathcal{N}(0, 1)$, $i \in [n]$.
- Test risk $\propto \mathbb{E}[||\hat{\boldsymbol{\beta}} - \mathbf{0}||_2^2] \propto \mathbb{E}[||\boldsymbol{X}^\dagger \boldsymbol{y}||_2^2] \propto \mathbb{E}[\mathrm{tr}((\boldsymbol{X}^\top \boldsymbol{X})^\dagger)]$, where $\boldsymbol{X} \in \mathbb{R}^{n \times d}$.
- When $n \neq d$, $\boldsymbol{X}$ is well conditioned.
- When $n \approx d$, $\boldsymbol{X}$ is infinitely ill conditioned.
- The model has marginally enough parameters to interpolate all the data, hence it interpolates in an awkward way.
- To fit the noise, the coefficients $||\hat{\beta}||_2^2 = ||\boldsymbol{X}^\dagger \boldsymbol{y}||_2^2$ blows up.

[Bartlett, Long, Lugosi, Tsigler, 2019], [Muthukumar, Vodrahalli, Sahai, 2019]

Why singularity?

- Model: $\boldsymbol{x}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$, $y_i = \langle \mathbf{0}, \boldsymbol{x}_i \rangle + \varepsilon_i \sim \mathcal{N}(0, 1)$, $i \in [n]$.
- Test risk $\propto \mathbb{E}[\|\hat{\boldsymbol{\beta}} - \mathbf{0}\|_2^2] \propto \mathbb{E}[\|\boldsymbol{X}^\dagger \boldsymbol{y}\|_2^2] \propto \mathbb{E}[\mathrm{tr}((\boldsymbol{X}^\top \boldsymbol{X})^\dagger)]$, where $\boldsymbol{X} \in \mathbb{R}^{n \times d}$.
- When $n \neq d$, $\boldsymbol{X}$ is well conditioned.
- When $n \approx d$, $\boldsymbol{X}$ is infinitely ill conditioned.

- The model has marginally enough parameters to interpolate all the data, hence it interpolates in an awkward way.
- To fit the noise, the coefficients $\|\hat{\boldsymbol{\beta}}\|_2^2 = \|\boldsymbol{X}^\dagger \boldsymbol{y}\|_2^2$ blows up.

[Bartlett, Long, Lugosi, Tsigler, 2019], [Muthukumar, Vodrahalli, Sahai, 2019]

Why singularity?

- Model: $\boldsymbol{x}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$, $y_i = \langle \mathbf{0}, \boldsymbol{x}_i \rangle + \varepsilon_i \sim \mathcal{N}(0, 1)$, $i \in [n]$.
- Test risk $\propto \mathbb{E}[||\hat{\boldsymbol{\beta}} - \mathbf{0}||_2^2] \propto \mathbb{E}[||\boldsymbol{X}^\dagger \boldsymbol{y}||_2^2] \propto \mathbb{E}[\mathrm{tr}((\boldsymbol{X}^\top \boldsymbol{X})^\dagger)]$, where $\boldsymbol{X} \in \mathbb{R}^{n \times d}$.
- When $n \neq d$, $\boldsymbol{X}$ is well conditioned.
- When $n \approx d$, $\boldsymbol{X}$ is infinitely ill conditioned.

- The model has marginally enough parameters to interpolate all the data, hence it interpolates in an awkward way.
- To fit the noise, the coefficients $||\hat{\beta}||_2^2 = ||\boldsymbol{X}^\dagger \boldsymbol{y}||_2^2$ blows up.

[Bartlett, Long, Lugosi, Tsigler, 2019], [Muthukumar, Vodrahalli, Sahai, 2019]

Why singularity?

- Model: $\boldsymbol{x}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$, $y_i = \langle \mathbf{0}, \boldsymbol{x}_i \rangle + \varepsilon_i \sim \mathcal{N}(0, 1)$, $i \in [n]$.
- Test risk $\propto \mathbb{E}[\|\hat{\boldsymbol{\beta}} - \mathbf{0}\|_2^2] \propto \mathbb{E}[\|\boldsymbol{X}^\dagger \boldsymbol{y}\|_2^2] \propto \mathbb{E}[\mathrm{tr}((\boldsymbol{X}^\top \boldsymbol{X})^\dagger)]$, where $\boldsymbol{X} \in \mathbb{R}^{n \times d}$.
- When $n \neq d$, $\boldsymbol{X}$ is well conditioned.
- When $n \approx d$, $\boldsymbol{X}$ is infinitely ill conditioned.

- The model has marginally enough parameters to interpolate all the data, hence it interpolates in an awkward way.
- To fit the noise, the coefficients $\|\hat{\boldsymbol{\beta}}\|_2^2 = \|\boldsymbol{X}^\dagger \boldsymbol{y}\|_2^2$ blows up.

[Bartlett, Long, Lugosi, Tsigler, 2019], [Muthukumar, Vodrahalli, Sahai, 2019]

Why singularity?

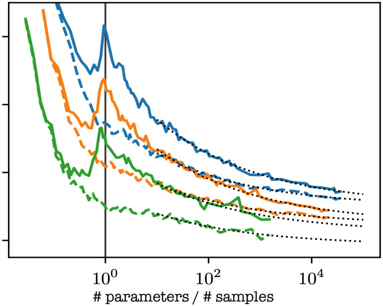
- Model: $\boldsymbol{x}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$, $y_i = \langle \mathbf{0}, \boldsymbol{x}_i \rangle + \varepsilon_i \sim \mathcal{N}(0, 1)$, $i \in [n]$.
- Test risk $\propto \mathbb{E}[||\hat{\boldsymbol{\beta}} - \mathbf{0}||_2^2] \propto \mathbb{E}[||\boldsymbol{X}^\dagger \boldsymbol{y}||_2^2] \propto \mathbb{E}[\mathrm{tr}((\boldsymbol{X}^\top \boldsymbol{X})^\dagger)]$, where $\boldsymbol{X} \in \mathbb{R}^{n \times d}$.
- When $n \neq d$, $\boldsymbol{X}$ is well conditioned.
- When $n \approx d$, $\boldsymbol{X}$ is infinitely ill conditioned.

- The model has marginally enough parameters to interpolate all the data, hence it interpolates in an awkward way.
- To fit the noise, the coefficients $||\hat{\boldsymbol{\beta}}||_2^2 = ||\boldsymbol{X}^\dagger \boldsymbol{y}||_2^2$ blows up.

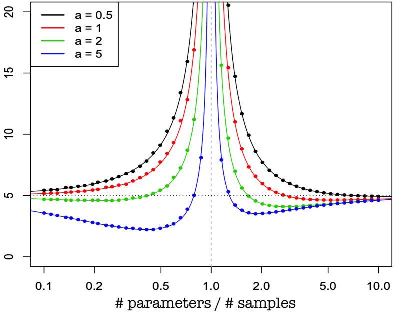
[Bartlett, Long, Lugosi, Tsigler, 2019], [Muthukumar, Vodrahalli, Sahai, 2019]

Comparison

Neural networks [Spigler, *et. al.*, 2018]



Linear model [Hastie, *et. al.*, 2019]



- ✓ Peak at the interpolation threshold.
- ? Monotone decreasing in the overparameterized regime.
- ? Global minimum when the number of parameters is infinity.

Goal: find a tractable model that exhibits all the features of the double descent curve.

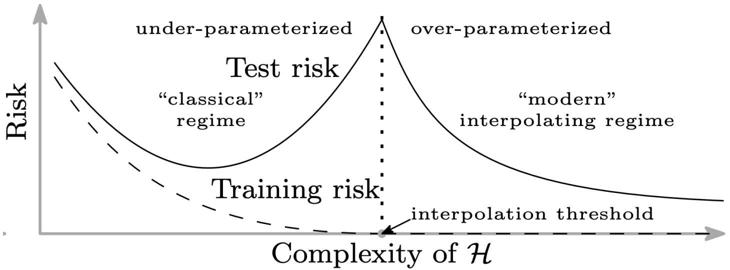


Figure: By [Belkin, Hsu, Ma, Mandal, 2018].

A simple model

The random features model

$$f_{\mathsf{RF}}(\boldsymbol{x}; \boldsymbol{a}) = \sum_{j=1}^{N} a_j \sigma(\langle \boldsymbol{w}_j, \boldsymbol{x} \rangle).$$

Random weights $(\boldsymbol{w}_j)_{j \in [N]}$

$$\boldsymbol{w}_j \sim_{iid} \mathrm{Unif}(\mathbb{S}^{d-1}).$$

A simple model

The random features model

$$f_{\mathsf{RF}}(\boldsymbol{x}; \boldsymbol{a}) = \sum_{j=1}^{N} a_j \sigma(\langle \boldsymbol{w}_j, \boldsymbol{x} \rangle).$$

Random weights $(\boldsymbol{w}_j)_{j \in [N]}$

$$\boldsymbol{w}_j \sim_{iid} \mathrm{Unif}(\mathbb{S}^{d-1}).$$

Data $(\boldsymbol{x}_i, y_i)_{i \in [n]}$

$$\boldsymbol{x}_i \sim \mathrm{Unif}(\mathbb{S}^{d-1}(\sqrt{d})), \qquad y_i = f_\star(\boldsymbol{x}_i) + \varepsilon_i.$$

A simple model

Random features regression: $\hat{\boldsymbol{a}}_\lambda = \arg\min_{\boldsymbol{a}} L_\lambda(\boldsymbol{a})$,

$$L_\lambda(\boldsymbol{a}) = \frac{1}{n}\sum_{i=1}^{n}\Big[\Big(y_i - \sum_{j=1}^{N} a_j \sigma(\langle \boldsymbol{x}_i, \boldsymbol{w}_j\rangle)\Big)^2\Big] + \frac{\lambda N}{d}\|\boldsymbol{a}\|_2^2, \qquad \text{(Train)}$$

$$R(\boldsymbol{a}; f_\star) = \mathbb{E}_{\boldsymbol{x},y}\Big[\Big(f_\star(\boldsymbol{x}) - \sum_{j=1}^{N} a_j \sigma(\langle \boldsymbol{x}, \boldsymbol{w}_j\rangle)\Big)^2\Big]. \qquad \text{(Test)}$$

Assumptions

- n data, N features, d dimension. $N/d \to \psi_1$, $n/d \to \psi_2$, as $d \to \infty$.
- Tech. ass. on $f_\star$ and σ (apply to almost every $f_\star$ and σ).

Precise asymptotics

Theorem (Mei and Montanari, 2019)

Under above assumptions, the test error of RF model is given by

$$R(\hat{\boldsymbol{a}}_\lambda; f_\star) = \|\boldsymbol{\beta}\|_2^2 \cdot \mathscr{B}(\zeta, \psi_1, \psi_2, \lambda/\mu_\star^2) + \tau^2 \cdot \mathscr{V}(\zeta, \psi_1, \psi_2, \lambda/\mu_\star^2) + o_{d,\mathbb{P}}(1),$$

where functions $\mathscr{B}$ *and* $\mathscr{V}$ *are given explicitly below.*

Explicit formulae

Let the functions $\nu_1, \nu_2 : \mathbb{C}_+ \to \mathbb{C}_+$ be the unique solution of

$$\nu_1 = \psi_1\Big(-\xi - \nu_2 - \frac{\zeta^2\nu_2}{1-\zeta^2\nu_1\nu_2}\Big)^{-1},$$
$$\nu_2 = \psi_2\Big(-\xi - \nu_1 - \frac{\zeta^2\nu_1}{1-\zeta^2\nu_1\nu_2}\Big)^{-1};$$

Let

$$\chi \equiv \nu_1(\boldsymbol{i}(\psi_1\psi_2\overline{\lambda})^{1/2}) \cdot \nu_2(\boldsymbol{i}(\psi_1\psi_2\overline{\lambda})^{1/2}),$$

and

$$\begin{aligned}
\mathscr{E}_0(\zeta,\psi_1,\psi_2,\overline{\lambda}) \equiv\; & -\chi^5\zeta^6 + 3\chi^4\zeta^4 + (\psi_1\psi_2 - \psi_2 - \psi_1 + 1)\chi^3\zeta^6 - 2\chi^3\zeta^4 - 3\chi^3\zeta^2 \\
& + (\psi_1 + \psi_2 - 3\psi_1\psi_2 + 1)\chi^2\zeta^4 + 2\chi^2\zeta^2 + \chi^2 + 3\psi_1\psi_2\chi\zeta^2 - \psi_1\psi_2\,, \\
\mathscr{E}_1(\zeta,\psi_1,\psi_2,\overline{\lambda}) \equiv\; & \psi_2\chi^3\zeta^4 - \psi_2\chi^2\zeta^2 + \psi_1\psi_2\chi\zeta^2 - \psi_1\psi_2\,, \\
\mathscr{E}_2(\zeta,\psi_1,\psi_2,\overline{\lambda}) \equiv\; & \chi^5\zeta^6 - 3\chi^4\zeta^4 + (\psi_1 - 1)\chi^3\zeta^6 + 2\chi^3\zeta^4 + 3\chi^3\zeta^2 + (-\psi_1 - 1)\chi^2\zeta^4 - 2\chi^2\zeta^2 - \chi^2\,.
\end{aligned}$$

We then have

$$\mathscr{B}(\zeta,\psi_1,\psi_2,\overline{\lambda}) \equiv \frac{\mathscr{E}_1(\zeta,\psi_1,\psi_2,\overline{\lambda})}{\mathscr{E}_0(\zeta,\psi_1,\psi_2,\overline{\lambda})}\,, \qquad \mathscr{V}(\zeta,\psi_1,\psi_2,\overline{\lambda}) \equiv \frac{\mathscr{E}_2(\zeta,\psi_1,\psi_2,\overline{\lambda})}{\mathscr{E}_0(\zeta,\psi_1,\psi_2,\overline{\lambda})}\,.$$

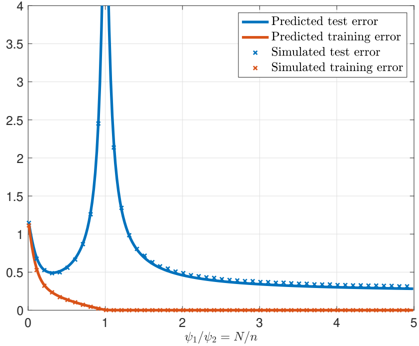
Proof strategy

Random matrix theory for the random kernel inner product matrices

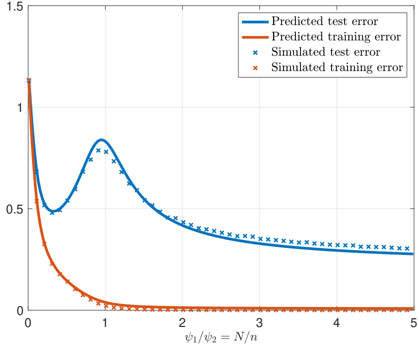
$$\boldsymbol{Z} = \Big(\sigma(\langle \boldsymbol{w}_i, \boldsymbol{x}_j \rangle)\Big)_{i \in [N], j \in [n]}.$$

[El Karoui, 2010], [Cheng, Singer, 2013], [Do, Vu, 2013], [Fan, Montanari, 2019], [Hastie, Montanari, Rosset, Tibshirani, 2019].

Analytical prediction



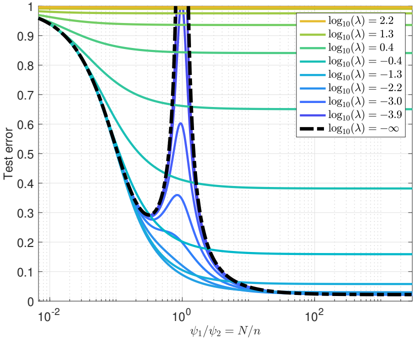
$\lambda = 0+$



$\lambda = 3 \times 10^{-4}$

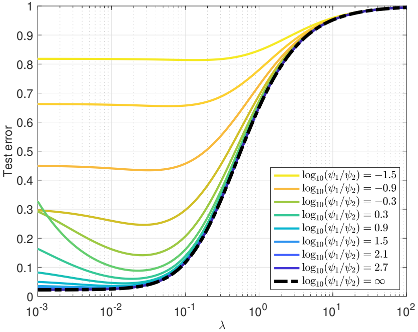
- ✓ Peak at the interpolation threshold.
- ✓ Monotone decreasing in the overparameterized regime.
- ✓ Global minimum when the number of parameters is infinity.

Insights

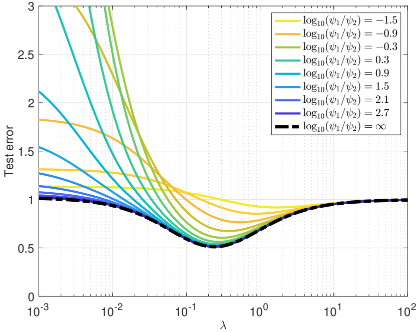


- For any λ, the min prediction error is achieved at $N/n \to \infty$.
- For optimal λ, the prediction error is monotonically decreasing.

Insights



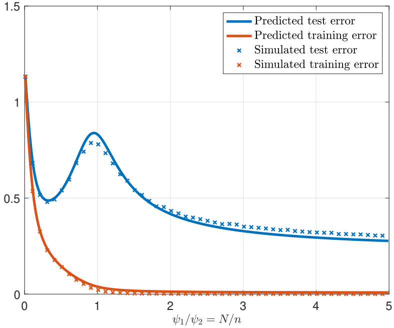
SNR $= 5$



SNR $= 1/10$

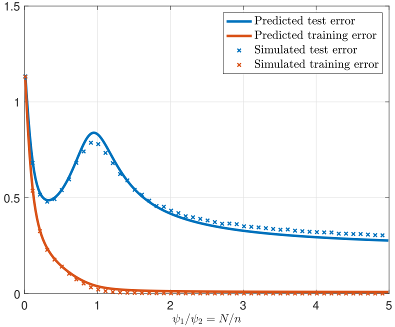
- High SNR: minimum at $\lambda = 0+$;
- Low SNR: minimum at $\lambda > 0$.

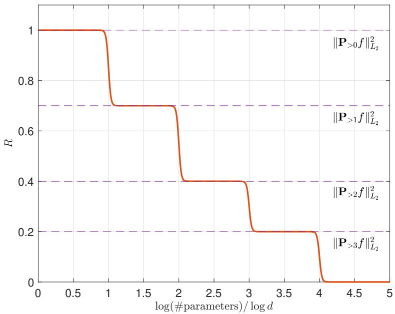
Summary of linearization of neural networks



- # parameters $\neq$ model complexity that controls generalization.
- Double descent also exists in linearized neural networks.

Summary of linearization of neural networks





- # parameters $\neq$ model complexity that controls generalization.
- Double descent also exists in linearized neural networks.
- Gap between NN and NT.
  NT models cannot fully explain the generalization efficacy of NN.

Going beyond linearization?

Mean field theory

- SGD of two layers neural networks

$$\boldsymbol{\theta}_i^{k+1} = \boldsymbol{\theta}_i^k - \varepsilon \nabla_{\boldsymbol{\theta}_i} \ell\Big(y_k, \frac{1}{N}\sum_{i=1}^N \sigma_\star(\boldsymbol{x}_k, \boldsymbol{\theta}_i^k)\Big).$$

- Consider empirical distribution of weights

$$\hat{\rho}_{N,k\varepsilon} = \frac{1}{N}\sum_{i=1}^N \delta_{\boldsymbol{\theta}_i^k}$$

- Then $\hat{\rho}_{N,t} \to \rho_t$ as $N \to \infty$ and $\varepsilon \to 0$, and ρ_t satisfies

$$\partial_t \rho_t = \nabla \cdot (\nabla \Psi(\boldsymbol{\theta}; \rho_t)\rho_t).$$

- Difference from linearization theory: A different scaling limit.

[**Mei**, Montanari, Nguyen, 2018], [Rotskoff, Vanden-Eijnden, 2018]

Future directions

- Distribution of features x matter.
  - Images $\leftrightarrow$ Convolutional neural network.
  - Graph $\leftrightarrow$ Graph neural network.
  - Exploring data and network invariance.
- Neural networks as function/distribution approximation?
  - Generative modeling.
  - Reinforcement learning.
- Uncertainty quantification in neural network systems.
  - Robustness and adversarial examples.
  - Approximate inference for Bayesian neural networks.
  - Predictive inference.

Thanks!