Mean Field Theory and Tangent Kernel Theory of Neural Networks

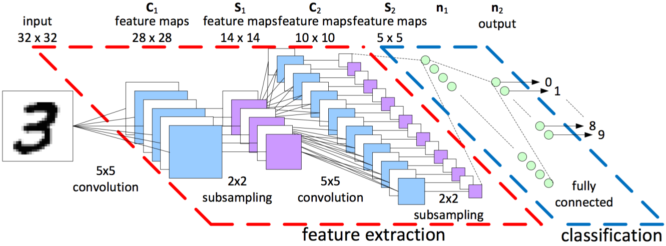
Song Mei

Stanford University

October 22, 2019

Stanford STATS385 guest lecture

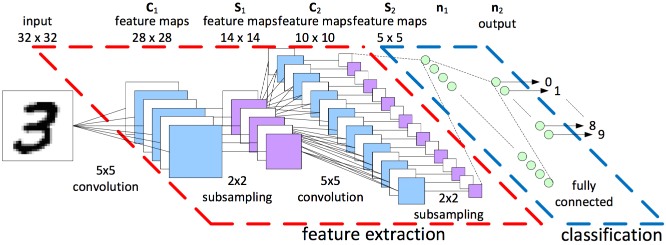
Deep learning applications



Applications

- Computer vision (autonomous vehicles).
- Generative modeling (WaveNet for generating speech).
- Reinforcement learning (Go playing).

Mathematical challenges/mysteries



- Optimization → Non-convexity.
- Generalization → Overparameterization.

Agenda

1. Mean field theory

2. Tangent kernel theory

3. Transitions between mean field and tangent kernel regime

Agenda

1. Mean field theory

2. Tangent kernel theory

3. Transitions between mean field and tangent kernel regime

Two-layers neural networks

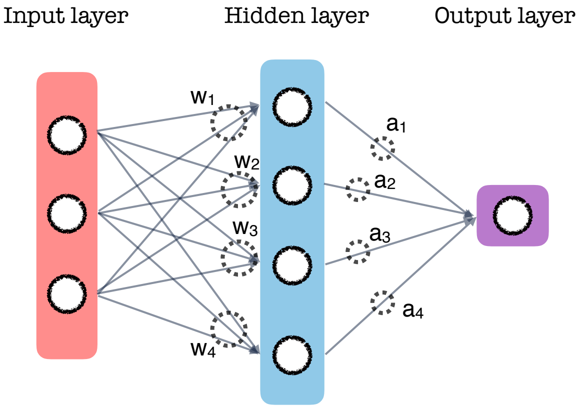


Figure: $\boldsymbol{\theta}_i = (a_i, \boldsymbol{w}_i)$.

Two-layers neural networks

- Parameters: $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N) \in \mathbb{R}^{N \times D}$.
- Prediction function:

$$f(x; \boldsymbol{\theta}) = \frac{1}{N} \sum_{i=1}^{N} \sigma_\star(x; \boldsymbol{\theta}_i) = \frac{1}{N} \sum_{i=1}^{N} a_i \sigma(\langle \boldsymbol{w}_i, x \rangle).$$

- Risk function:

$$R_N(\boldsymbol{\theta}) = \mathbb{E}_{x,y}\Big[\ell\Big(y, \frac{1}{N} \sum_{i=1}^{N} \sigma_\star(x; \boldsymbol{\theta}_i)\Big)\Big].$$

- Gradient flow for R_N:

$$\frac{\mathrm{d}}{\mathrm{d}t} \boldsymbol{\theta}_i^t = -N\xi(t) \nabla_{\boldsymbol{\theta}_i} R_N(\boldsymbol{\theta}^t). \quad \text{(GF)}$$

- Difficulty: non-convexity with local minimizers!

Two-layers neural networks

- Parameters: $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N) \in \mathbb{R}^{N \times D}$.
- Prediction function:

$$f(\boldsymbol{x}; \boldsymbol{\theta}) = \frac{1}{N} \sum_{i=1}^{N} \sigma_\star(\boldsymbol{x}; \boldsymbol{\theta}_i) = \frac{1}{N} \sum_{i=1}^{N} a_i \sigma(\langle \boldsymbol{w}_i, \boldsymbol{x} \rangle).$$

- Risk function:

$$R_N(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{x},y}\Big[\ell\Big(y, \frac{1}{N} \sum_{i=1}^{N} \sigma_\star(\boldsymbol{x}; \boldsymbol{\theta}_i)\Big)\Big].$$

- Gradient flow for R_N:

$$\frac{\mathrm{d}}{\mathrm{d}t} \boldsymbol{\theta}_i^t = -N\xi(t) \nabla_{\boldsymbol{\theta}_i} R_N(\boldsymbol{\theta}^t). \tag{GF}$$

- Difficulty: non-convexity with local minimizers!

Two-layers neural networks

- Parameters: $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N) \in \mathbb{R}^{N \times D}$.
- Prediction function:

$$f(\boldsymbol{x}; \boldsymbol{\theta}) = \frac{1}{N} \sum_{i=1}^{N} \sigma_\star(\boldsymbol{x}; \boldsymbol{\theta}_i) = \frac{1}{N} \sum_{i=1}^{N} a_i \sigma(\langle \boldsymbol{w}_i, \boldsymbol{x} \rangle).$$

- Risk function:

$$R_N(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{x}, y}\Big[\ell\Big(y, \frac{1}{N} \sum_{i=1}^{N} \sigma_\star(\boldsymbol{x}; \boldsymbol{\theta}_i)\Big)\Big].$$

- Gradient flow for R_N:

$$\frac{\mathrm{d}}{\mathrm{d}t} \boldsymbol{\theta}_i^t = -N \xi(t) \nabla_{\boldsymbol{\theta}_i} R_N(\boldsymbol{\theta}^t). \qquad \text{(GF)}$$

- Difficulty: non-convexity with local minimizers!

Two-layers neural networks

- Parameters: $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N) \in \mathbb{R}^{N \times D}$.
- Prediction function:

$$f(\boldsymbol{x}; \boldsymbol{\theta}) = \frac{1}{N} \sum_{i=1}^{N} \sigma_\star(\boldsymbol{x}; \boldsymbol{\theta}_i) = \frac{1}{N} \sum_{i=1}^{N} a_i \sigma(\langle \boldsymbol{w}_i, \boldsymbol{x} \rangle).$$

- Risk function:

$$R_N(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{x},y}\Big[\ell\Big(y, \frac{1}{N} \sum_{i=1}^{N} \sigma_\star(\boldsymbol{x}; \boldsymbol{\theta}_i)\Big)\Big].$$

- Gradient flow for R_N:

$$\frac{\mathrm{d}}{\mathrm{d}t} \boldsymbol{\theta}_i^t = -N\xi(t) \nabla_{\boldsymbol{\theta}_i} R_N(\boldsymbol{\theta}^t). \tag{GF}$$

- Difficulty: non-convexity with local minimizers!

Two-layers neural networks

- Parameters: $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N) \in \mathbb{R}^{N \times D}$.
- Prediction function:

$$f(\boldsymbol{x}; \boldsymbol{\theta}) = \frac{1}{N} \sum_{i=1}^{N} \sigma_\star(\boldsymbol{x}; \boldsymbol{\theta}_i) = \frac{1}{N} \sum_{i=1}^{N} a_i \sigma(\langle \boldsymbol{w}_i, \boldsymbol{x} \rangle).$$

- Risk function:

$$R_N(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{x},y}\Big[\ell\Big(y, \frac{1}{N} \sum_{i=1}^{N} \sigma_\star(\boldsymbol{x}; \boldsymbol{\theta}_i)\Big)\Big].$$

- Gradient flow for R_N:

$$\frac{\mathrm{d}}{\mathrm{d}t} \boldsymbol{\theta}_i^t = -N \xi(t) \nabla_{\boldsymbol{\theta}_i} R_N(\boldsymbol{\theta}^t). \tag{GF}$$

- Difficulty: non-convexity with local minimizers!

Landscape analysis?

$$R_N(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{x},y}\Big[\Big(y - \frac{1}{N}\sum_{j=1}^{N}\sigma_\star(\boldsymbol{x};\boldsymbol{\theta}_j)\Big)^2\Big].$$

- [Kawaguchi, 2016], [Freeman, Bruna, 2016]: linear network has no spurious local min.
- [Soltanolkotabi, Javanmard, Lee, 2017]: Quadratic two-layers NN has no spurious local min.
- [Zhong, Song, Jain, Bartlett, Dhillon, 2017]: Local strong convexity of two layers NN.
- [Soudry, Carmon, 2016], [Ge, Lee, Ma, 2017], [Tian, 2017], [Soltanolkotabi, 2017], [Li, Yuan, 2017] ...

Mean field perspective: Emp. dist. of weights

- Prediction function

$$f(\boldsymbol{x};\boldsymbol{\theta}) = \frac{1}{N}\sum_{i=1}^{N}\sigma_\star(\boldsymbol{x};\boldsymbol{\theta}_i) = \int \sigma_\star(\boldsymbol{x};\boldsymbol{\theta})\hat{\rho}_N(\mathrm{d}\boldsymbol{\theta}).$$

- Empirical distribution of the weights:

$$\hat{\rho}_N = \frac{1}{N}\sum_{i=1}^{N}\delta_{\boldsymbol{\theta}_i} \in \mathcal{P}(\mathbb{R}^D).$$

- Risk functional $R : \mathcal{P}(\mathbb{R}^D) \to \mathbb{R}$

$$R(\rho) = \mathbb{E}_{\boldsymbol{x},y}\Big[\ell\Big(y, \int \sigma_\star(\boldsymbol{x};\boldsymbol{\theta})\rho(\mathrm{d}\boldsymbol{\theta})\Big)\Big].$$

Induced dynamics on empirical distribution

- Gradient flow on particles $\{\boldsymbol{\theta}_i^t\}_{i\in[N]}$, $\boldsymbol{\theta}_i^t \in \mathbb{R}^D$,

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\theta}_i^t = -N\xi(t)\nabla_{\boldsymbol{\theta}_i} R_N(\boldsymbol{\theta}^t), \qquad \boldsymbol{\theta}_i^0 = \boldsymbol{\theta}_i^0. \tag{GF}$$

- Define dynamics on distribution $\rho_{N,t} \in \mathcal{P}(\mathbb{R}^D)$,

$$\partial_t \rho_{N,t}(\boldsymbol{\theta}) = 2\xi(t)\nabla_{\boldsymbol{\theta}} \cdot \left(\rho_{N,t}(\boldsymbol{\theta})\nabla_{\boldsymbol{\theta}}\Psi(\boldsymbol{\theta};\rho_{N,t})\right), \tag{PDE}$$

with

$$\rho_{N,0} = \frac{1}{N}\sum_{i=1}^N \delta_{\boldsymbol{\theta}_i^0}, \qquad \Psi(\boldsymbol{\theta};\rho) = \frac{\delta R}{\delta \rho}(\boldsymbol{\theta};\rho).$$

- Claim: $\rho_{N,t} = (1/N)\sum_{i=1}^N \delta_{\boldsymbol{\theta}_i^t}$.

Induced dynamics on empirical distribution

- Gradient flow on particles $\{\boldsymbol{\theta}_i^t\}_{i\in[N]}$, $\boldsymbol{\theta}_i^t \in \mathbb{R}^D$,

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\theta}_i^t = -N\xi(t)\nabla_{\boldsymbol{\theta}_i} R_N(\boldsymbol{\theta}^t), \qquad \boldsymbol{\theta}_i^0 = \boldsymbol{\theta}_i^0. \tag{GF}$$

- Define dynamics on distribution $\rho_{N,t} \in \mathcal{P}(\mathbb{R}^D)$,

$$\partial_t \rho_{N,t}(\boldsymbol{\theta}) = 2\xi(t)\nabla_{\boldsymbol{\theta}} \cdot \big(\rho_{N,t}(\boldsymbol{\theta})\nabla_{\boldsymbol{\theta}}\Psi(\boldsymbol{\theta};\rho_{N,t})\big), \tag{PDE}$$

  with

$$\rho_{N,0} = \frac{1}{N}\sum_{i=1}^N \delta_{\boldsymbol{\theta}_i^0}, \qquad \Psi(\boldsymbol{\theta};\rho) = \frac{\delta R}{\delta \rho}(\boldsymbol{\theta};\rho).$$

- Claim: $\rho_{N,t} = (1/N)\sum_{i=1}^N \delta_{\boldsymbol{\theta}_i^t}$.

Induced dynamics on empirical distribution

- Gradient flow on particles $\{\boldsymbol{\theta}_i^t\}_{i \in [N]}$, $\boldsymbol{\theta}_i^t \in \mathbb{R}^D$,

$$\frac{\mathrm{d}}{\mathrm{d}t} \boldsymbol{\theta}_i^t = -N \xi(t) \nabla_{\boldsymbol{\theta}_i} R_N(\boldsymbol{\theta}^t), \qquad \boldsymbol{\theta}_i^0 = \boldsymbol{\theta}_i^0. \tag{GF}$$

- Define dynamics on distribution $\rho_{N,t} \in \mathcal{P}(\mathbb{R}^D)$,

$$\partial_t \rho_{N,t}(\boldsymbol{\theta}) = 2\xi(t) \nabla_{\boldsymbol{\theta}} \cdot \big(\rho_{N,t}(\boldsymbol{\theta}) \nabla_{\boldsymbol{\theta}} \Psi(\boldsymbol{\theta}; \rho_{N,t})\big), \tag{PDE}$$

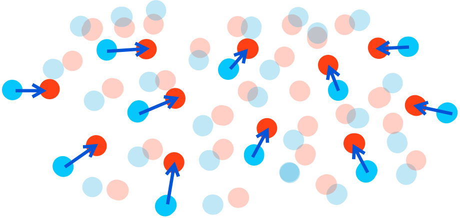
with

$$\rho_{N,0} = \frac{1}{N} \sum_{i=1}^N \delta_{\boldsymbol{\theta}_i^0}, \qquad \Psi(\boldsymbol{\theta}; \rho) = \frac{\delta R}{\delta \rho}(\boldsymbol{\theta}; \rho).$$

- Claim: $\rho_{N,t} = (1/N) \sum_{i=1}^N \delta_{\boldsymbol{\theta}_i^t}$.

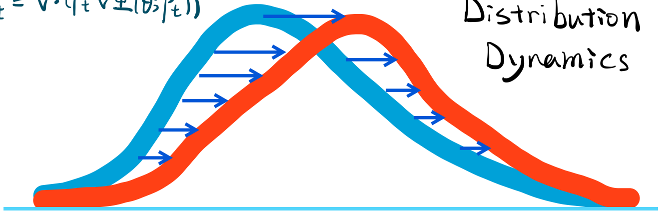
Particle Dynamics

$$\frac{d}{dt}\theta_i^t = -\nabla_{\theta_i} R_N(\theta^t)$$



$$\partial_t \rho_t = \nabla\cdot(\rho_t \nabla \Psi(\theta;\rho_t))$$

Distribution Dynamics



A short proof

Test function + Chain rule + Integration by part.

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\int f(\boldsymbol{\theta})\rho_{N,t}(\mathrm{d}\boldsymbol{\theta}) =& \frac{\mathrm{d}}{\mathrm{d}t}\Big[\frac{1}{N}\sum_{i=1}^{N} f(\boldsymbol{\theta}_i^t)\Big] = -\frac{1}{N}\sum_{i=1}^{N}\langle \nabla f(\boldsymbol{\theta}_i^t), N\nabla_{\boldsymbol{\theta}_i} R_N(\boldsymbol{\theta}^t)\rangle \\ =& -\frac{1}{N}\sum_{i=1}^{N}\Big\langle \nabla f(\boldsymbol{\theta}_i^t), \nabla[\delta R/\delta\rho](\boldsymbol{\theta}_i^t;\rho_{N,t})\Big\rangle \\ =& -\int\Big\langle \nabla f, \nabla[\delta R/\delta\rho](\boldsymbol{\theta};\rho_{N,t})\Big\rangle\rho_{N,t}(\mathrm{d}\boldsymbol{\theta}).\end{aligned}$$

What is this PDE?

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

Existence and uniqueness: [Sznitman, 1991].

- Physics: nonlinear transport equation describing motions of particles with pairwise interaction (mean field approach).
- Math: Gradient flow of $R(\rho)$ in the metric space $(\mathcal{P}(\mathbb{R}^D), W_2)$. [Jordan, Kinderlehrer, Otto, 1998], [Ambrosio, Gigli, Savaré, 2006], [Carrillo, McCann, Villani, 2013]

What is this PDE?

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

Existence and uniqueness: [Sznitman, 1991].

- Physics: nonlinear transport equation describing motions of particles with pairwise interaction (mean field approach).
- Math: Gradient flow of $R(\rho)$ in the metric space $(\mathcal{P}(\mathbb{R}^D), W_2)$. [Jordan, Kinderlehrer, Otto, 1998], [Ambrosio, Gigli, Savaré, 2006], [Carrillo, McCann, Villani, 2013]

Converge of $\rho_{N,t}$ to ρ_t as $N \to \infty$

Let $\rho_{N,t}$ be the solution of ($\boldsymbol{\theta}_i^0 \sim_{iid} \rho_0 \in \mathcal{P}(\mathbb{R}^D)$)

$$\partial_t \rho_{N,t} = \nabla_{\boldsymbol{\theta}} \cdot \Big(\rho_{N,t} \nabla_{\boldsymbol{\theta}} \Psi(\boldsymbol{\theta}; \rho_{N,t}) \Big), \quad \rho_{N,0} = \frac{1}{N} \sum_{i=1}^{N} \delta_{\boldsymbol{\theta}_i^0}.$$

Let ρ_t be the solution of

$$\partial_t \rho_t = \nabla_{\boldsymbol{\theta}} \cdot \Big(\rho_t \nabla_{\boldsymbol{\theta}} \Psi(\boldsymbol{\theta}; \rho_t) \Big), \quad \rho_t \big|_{t=0} = \rho_0.$$

Converge of $\rho_{N,t}$ to ρ_t as $N \to \infty$

Let $\rho_{N,t}$ be the solution of $(\boldsymbol{\theta}_i^0 \sim_{iid} \rho_0 \in \mathcal{P}(\mathbb{R}^D))$

$$\partial_t \rho_{N,t} = \nabla_{\boldsymbol{\theta}} \cdot \Big(\rho_{N,t} \nabla_{\boldsymbol{\theta}} \Psi(\boldsymbol{\theta}; \rho_{N,t})\Big), \quad \rho_{N,0} = \frac{1}{N} \sum_{i=1}^{N} \delta_{\boldsymbol{\theta}_i^0}.$$

Let ρ_t be the solution of

$$\partial_t \rho_t = \nabla_{\boldsymbol{\theta}} \cdot \Big(\rho_t \nabla_{\boldsymbol{\theta}} \Psi(\boldsymbol{\theta}; \rho_t)\Big), \quad \rho_t|_{t=0} = \rho_0.$$

Converge of $\rho_{N,t}$ to ρ_t as $N \to \infty$

Theorem (M., Montanari, and Nguyen, 2018)

Under some assumptions. Let $(\theta_i^0)_{i \le N} \sim_{iid} \rho_0$. *Denote* $\rho_{N,t}$: *emp. dist. of sol. of grad. flow with init.* θ^0. *Denote* ρ_t: *sol. of PDE with init.* ρ_0. *Then,* $\forall f$ *bounded Lipschitz,*

$$\sup_{t \le T} \left| \int f(\theta) \rho_{N,t}(\mathrm{d}\theta) - \int f(\theta) \rho_t(\mathrm{d}\theta) \right| \le K e^{KT} \sqrt{\frac{\log(N/\delta)}{N}}$$

with probability at least $1 - \delta$.

- See also [Rotskoff, Vanden-Eijnden, 2018], [Sirignano, Spiliopoulos, 2018].

Immediate implication

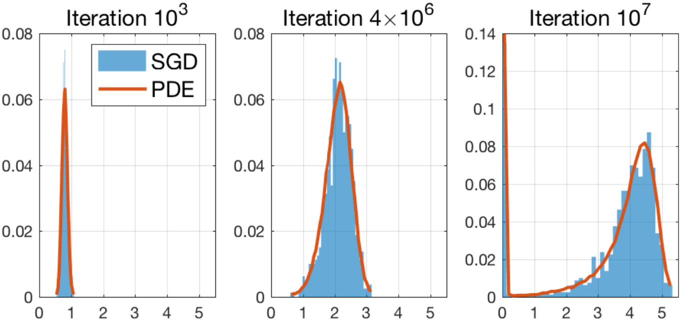
$$\rho_{N,t} \to \rho_t, \qquad N \to \infty$$

$$\partial \rho_t = \nabla \cdot \Big(\rho_t \nabla \Psi(\boldsymbol{\theta}; \rho_t) \Big), \qquad \rho_t \big|_{t=0} = \rho_0.$$

Convergence speed of N-neuron gradient flow is independent of N!

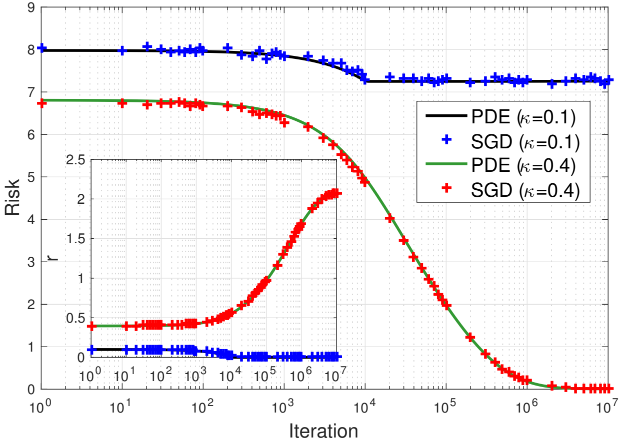
- Effective dimension from $N \times D$ to D.

SGD v.s. PDE



Iteration 10^3
Iteration 4×10^6
Iteration 10^7
SGD
PDE
0.08
0.06
0.04
0.02
0
0.14
0.12
0.1
0 1 2 3 4 5

SGD v.s. PDE



Risk
Iteration
r
PDE (κ=0.1)
SGD (κ=0.1)
PDE (κ=0.4)
SGD (κ=0.4)

Does $R(\rho_t) \to \min_\rho R(\rho)$ as $t \to \infty$?

In general, no convergence guarantees.

But sometimes, yes.

[M., Montanari, Nguyen, 2018]

- Case by case: a special mixture of two Gaussians.
- Noisy SGD (PDE with diffusion term).

PDE with diffusion term

Noisy gradient flow

$$\mathrm{d}\boldsymbol{\theta}_i^t = -N\nabla_{\boldsymbol{\theta}_i} R_N(\boldsymbol{\theta}^t)\mathrm{d}t + \frac{1}{\sqrt{\beta}}\mathrm{d}\boldsymbol{W}_i^t.$$

PDE with diffusion term

$$\partial_t \rho_t = \nabla_{\boldsymbol{\theta}} \cdot \Big(\rho_t \nabla_{\boldsymbol{\theta}} \Psi(\boldsymbol{\theta}; \rho_t)\Big) + \frac{1}{\beta}\Delta \rho_t. \qquad (\star)$$

Wasserstein grad. flow of the free energy

$$F_\beta(\rho) = R(\rho) + \frac{1}{\beta}\int \rho(\boldsymbol{\theta}) \log \rho(\boldsymbol{\theta}) \mathrm{d}\boldsymbol{\theta}.$$

Theorem (M., Montanari, Nguyen, 2018)

$(\star)$ converges to the minimizer of $F_\beta(\rho)$ as $t \to \infty$.

General convergence for noisy SGD

Theorem (M., Montanari, Nguyen, 2018)

Under certain assumptions. Initialization $(\boldsymbol{\theta}_i^0)_{i \le N} \sim_{iid} \rho_0$*. Then there exists* $\beta_0 = \beta_0(D, U, V, \eta)$*, such that, for* $\beta \ge \beta_0$*, there exists* $T = T(D, U, V, \beta, \eta)$ *such that for any* $k \in [T/\varepsilon, 10T/\varepsilon]$*,* $N \ge C_0 D \log D$*,* $\varepsilon \le 1/(C_0 D)$*, we have, w.h.p.*

$$R_{\lambda,N}(\boldsymbol{\theta}^k) \le \inf_{\boldsymbol{\theta} \in \mathbb{R}^{D \times N}} R_{\lambda,N}(\boldsymbol{\theta}) + \eta.$$

- Cautious: no polynomial convergence rate.

General convergence for noisy SGD

Theorem (M., Montanari, Nguyen, 2018)

Under certain assumptions. Initialization $(\theta_i^0)_{i \le N} \sim_{iid} \rho_0$*. Then there exists* $\beta_0 = \beta_0(D, U, V, \eta)$*, such that, for* $\beta \ge \beta_0$*, there exists* $T = T(D, U, V, \beta, \eta)$ *such that for any* $k \in [T/\varepsilon, 10T/\varepsilon]$*,* $N \ge C_0 D \log D$*,* $\varepsilon \le 1/(C_0 D)$*, we have, w.h.p.*

$$R_{\lambda,N}(\boldsymbol{\theta}^k) \le \inf_{\boldsymbol{\theta} \in \mathbb{R}^{D \times N}} R_{\lambda,N}(\boldsymbol{\theta}) + \eta.$$

- Cautious: no polynomial convergence rate.

Other references

Other references

- [Chizat, Bach, 2018a]: Global convergence with homogenuity.
- [Ma, Lee, Liu, Wei, 2019]: Quantitative rate for modified PDE.
- [Rotskoff, Jelassi, Bruna, Vanden-Eijnden, 2019]: Semi-quantitative rate for the birth-death process.
- [Nguyen, 2019], [Sirignano, Spiliopoulos, 2019], [Araújo, Oliveira, Yukimura, 2019]: Extension to multilayers.

Still many open problems: establish global convergence, extend to multi-layers.

Agenda

1. Mean field theory

2. Tangent kernel theory

3. Transitions between mean field and tangent kernel regime

The neural tangent model

- Multi-layers neural network $f(\boldsymbol{x};\boldsymbol{\theta})$

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

- Linearization around random parameter θ_0

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

- NT model: the linear part of f

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

The neural tangent model

- Multi-layers neural network $f(\boldsymbol{x};\boldsymbol{\theta})$

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

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

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

- NT model: the linear part of f

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

The neural tangent model

- Multi-layers neural network $f(\boldsymbol{x};\boldsymbol{\theta})$

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

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

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

- NT model: the linear part of f

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

The neural tangent model

- NT model: the linear part of f

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

- Random feature map: $\nabla_{\boldsymbol{\theta}} f(\boldsymbol{x};\boldsymbol{\theta}_0)$.
- Neural tangent kernel: $\mathcal{K}_{\mathsf{NT}}(\boldsymbol{x},\boldsymbol{y}) = \langle \nabla_{\boldsymbol{\theta}} f(\boldsymbol{x};\boldsymbol{\theta}_0), \nabla_{\boldsymbol{\theta}} f(\boldsymbol{y};\boldsymbol{\theta}_0)\rangle$. [Jacot, Gabriel, Hongler, 2018], [Du, Zhai, Poczos, Singh, 2018], [Chizat, Bach, 2018b], ...
- Successful optimization: under certain conditions (different from the mean field theory), the trajectory (of GF on empirical risk) of NT and NN is uniformly close.

Does NT models fully explain the success of neural networks?

The neural tangent model

- NT model: the linear part of f

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

- Random feature map: $\nabla_{\boldsymbol{\theta}} f(\boldsymbol{x};\boldsymbol{\theta}_0)$.
- Neural tangent kernel: $\mathcal{K}_{\mathsf{NT}}(\boldsymbol{x},\boldsymbol{y}) = \langle \nabla_{\boldsymbol{\theta}} f(\boldsymbol{x};\boldsymbol{\theta}_0), \nabla_{\boldsymbol{\theta}} f(\boldsymbol{y};\boldsymbol{\theta}_0)\rangle$. [Jacot, Gabriel, Hongler, 2018], [Du, Zhai, Poczos, Singh, 2018], [Chizat, Bach, 2018b], ...
- Successful optimization: under certain conditions (different from the mean field theory), the trajectory (of GF on empirical risk) of NT and NN is uniformly close.

Does NT models fully explain the success of neural networks?

The neural tangent model

- NT model: the linear part of f

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

- Random feature map: $\nabla_{\boldsymbol{\theta}} f(\boldsymbol{x};\boldsymbol{\theta}_0)$.
- Neural tangent kernel: $\mathcal{K}_{\mathsf{NT}}(\boldsymbol{x},\boldsymbol{y}) = \langle \nabla_{\boldsymbol{\theta}} f(\boldsymbol{x};\boldsymbol{\theta}_0), \nabla_{\boldsymbol{\theta}} f(\boldsymbol{y};\boldsymbol{\theta}_0)\rangle$. [Jacot, Gabriel, Hongler, 2018], [Du, Zhai, Poczos, Singh, 2018], [Chizat, Bach, 2018b], ...
- Successful optimization: under certain conditions (different from the mean field theory), the trajectory (of GF on empirical risk) of NT and NN is uniformly close.

Does NT models fully explain the success of neural networks?

The neural tangent model

- NT model: the linear part of f

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

- Random feature map: $\nabla_{\boldsymbol{\theta}} f(\boldsymbol{x};\boldsymbol{\theta}_0)$.
- Neural tangent kernel: $\mathcal{K}_{\mathsf{NT}}(\boldsymbol{x},\boldsymbol{y}) = \langle \nabla_{\boldsymbol{\theta}} f(\boldsymbol{x};\boldsymbol{\theta}_0), \nabla_{\boldsymbol{\theta}} f(\boldsymbol{y};\boldsymbol{\theta}_0)\rangle$. [Jacot, Gabriel, Hongler, 2018], [Du, Zhai, Poczos, Singh, 2018], [Chizat, Bach, 2018b], ...
- Successful optimization: under certain conditions (different from the mean field theory), the trajectory (of GF on empirical risk) of NT and NN is uniformly close.

Does NT models fully explain the success of neural networks?

Generalization

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

Table: Cifar10 experiments

Architecture	Classification error
Best convolutional NN	5%-
Best convolutional NT	23%
CNN of best CNT	19%

[Arora, Du, Hu, Li, Salakhutdinov, Wang, 2019]

Theoretical analysis of generalization gap

Two-layers neural network

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

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

Random features model and Neural tangent model

Linearization

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

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

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

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

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

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 lower bound

Lower bound for random features regression

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

Theorem (Ghorbani, M., Misiakiwics, Montanari, 2019)

Assume $N = O_d(d^{\ell+1-\delta})$, *and* $(\boldsymbol{w}_i)_{i \in [N]} \sim \mathrm{Unif}(\mathbb{S}^{d-1})$, *we have*

$$\inf_{f \in \mathcal{F}_{\mathsf{RF},N}(\boldsymbol{W})} \mathbb{E}_{\boldsymbol{x}}[(f_\star(\boldsymbol{x}) - f(\boldsymbol{x}))^2] \geq \|\mathsf{P}_{>\ell} f_\star\|_{L^2}^2 + o_{d,\mathbb{P}}(\|f_\star\|_2^2).$$

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

With $N = O_d(d^k)$ parameters, one can only fit a degree k polynomial.

Lower bound for random features regression

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

Theorem (Ghorbani, M., Misiakiwics, Montanari, 2019)

Assume $N = O_d(d^{\ell+1-\delta})$, *and* $(\boldsymbol{w}_i)_{i \in [N]} \sim \mathrm{Unif}(\mathbb{S}^{d-1})$, *we have*

$$\inf_{f \in \mathcal{F}_{\mathsf{RF},N}(\boldsymbol{W})} \mathbb{E}_{\boldsymbol{x}}[(f_\star(\boldsymbol{x}) - f(\boldsymbol{x}))^2] \geq \| \mathsf{P}_{>\ell} f_\star \|_{L^2}^2 + o_{d,\mathbb{P}}(\| f_\star \|_2^2).$$

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

With $N = O_d(d^k)$ parameters, one can only fit a degree k polynomial.

Similar result for NT

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

Theorem (Ghorbani, M., Misiakiwics, Montanari, 2019)

Assume $N = O_d(d^{\ell+1-\delta})$, *and* $(\boldsymbol{w}_i)_{i\in[N]} \sim \mathrm{Unif}(\mathbb{S}^{d-1})$, *we have*

$$\inf_{f\in\mathcal{F}_{\mathsf{NT},N}(\boldsymbol{W})} \mathbb{E}_{\boldsymbol{x}}[(f_\star(\boldsymbol{x}) - f(\boldsymbol{x}))^2] \geq \|\mathsf{P}_{>\ell+1} f_\star\|_{L^2}^2 + o_{d,\mathbb{P}}(\|f_\star\|_2^2),$$

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

With $O_d(d^{k+1})$ parameters, one can only fit a degree $k+1$ polynomial.

Similar result for NT

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

Theorem (Ghorbani, M., Misiakiwics, Montanari, 2019)

Assume $N = O_d(d^{\ell+1-\delta})$, *and* $(\boldsymbol{w}_i)_{i\in[N]} \sim \mathrm{Unif}(\mathbb{S}^{d-1})$, *we have*

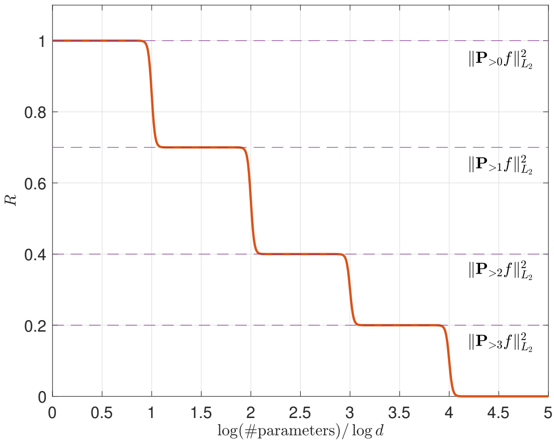
$$\inf_{f\in\mathcal{F}_{\mathsf{NT},N}(\boldsymbol{W})} \mathbb{E}_{\boldsymbol{x}}[(f_\star(\boldsymbol{x}) - f(\boldsymbol{x}))^2] \geq \|\mathsf{P}_{>\ell+1} f_\star\|_{L^2}^2 + o_{d,\mathbb{P}}(\|f_\star\|_2^2),$$

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

With $O_d(d^{k+1})$ parameters, one can only fit a degree $k+1$ polynomial.

The staircase lower bound

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



Implication

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

- NT: $N \geq \Theta_d(d^{k-1})$;
- NN: $N = O_d(1)$.
- Different from the RKHS theory [Bach, 2017], [E, Ma, Wu, 2018].
- Difference 1:
$$f_\star \in L^2, \quad \text{v.s.} \quad f_\star \in \text{RKHS}.$$
- Difference 2:
$$N = d^k \text{ as } d \to \infty, \quad \text{v.s.} \quad \text{fixed } d \text{ as } N \to \infty.$$

Implication

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

- NT: $N \geq \Theta_d(d^{k-1})$;
- NN: $N = O_d(1)$.
- Different from the RKHS theory [Bach, 2017], [E, Ma, Wu, 2018].
- Difference 1:

$$f_\star \in L^2, \qquad \text{v.s.} \qquad f_\star \in \text{RKHS}.$$

- Difference 2:

$$N = d^k \text{ as } d \to \infty, \qquad \text{v.s.} \qquad \text{fixed } d \text{ as } N \to \infty.$$

double descent curve

Experiment setup

- MNIST dataset: $(\boldsymbol{x}, y) \in \mathbb{R}^{100} \times [10]$.
  Training/test data size: 50000/10000.

- Two-layers neural networks:

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

  Bottom layer weights $\boldsymbol{w}_i \in \mathbb{R}^{100}$. Top layer weights $a_i \in \mathbb{R}$.

- Loss function: cross entropy. Training algorithm: SGD.

- N: model complexity, to be varied.

Experiment setup

- MNIST dataset: $(\boldsymbol{x}, y) \in \mathbb{R}^{100} \times [10]$.
  Training/test data size: 50000/10000.

- Two-layers neural networks:

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

  Bottom layer weights $\boldsymbol{w}_i \in \mathbb{R}^{100}$. Top layer weights $a_i \in \mathbb{R}$.

- Loss function: cross entropy. Training algorithm: SGD.

- N: model complexity, to be varied.

Experiment setup

- MNIST dataset: $(\boldsymbol{x}, y) \in \mathbb{R}^{100} \times [10]$.
  Training/test data size: 50000/10000.

- Two-layers neural networks:

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

  Bottom layer weights $\boldsymbol{w}_i \in \mathbb{R}^{100}$. Top layer weights $a_i \in \mathbb{R}$.

- Loss function: cross entropy. Training algorithm: SGD.

- N: model complexity, to be varied.

Experiment setup

- MNIST dataset: $(\boldsymbol{x}, y) \in \mathbb{R}^{100} \times [10]$.
  Training/test data size: 50000/10000.

- Two-layers neural networks:

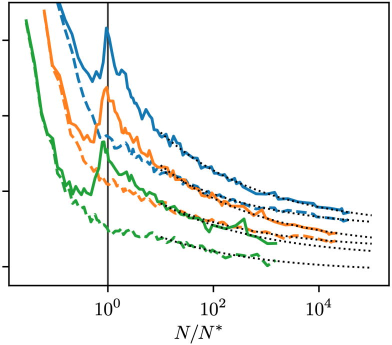
$$f_N(\boldsymbol{x}; \boldsymbol{\Theta}) = \sum_{i=1}^{N} a_i \sigma(\langle \boldsymbol{w}_i, \boldsymbol{x} \rangle), \quad \boldsymbol{\Theta} = (a_1, \boldsymbol{w}_1, \ldots, a_N, \boldsymbol{w}_N).$$

  Bottom layer weights $\boldsymbol{w}_i \in \mathbb{R}^{100}$. Top layer weights $a_i \in \mathbb{R}$.

- Loss function: cross entropy. Training algorithm: SGD.

- N: model complexity, to be varied.

Experimental results



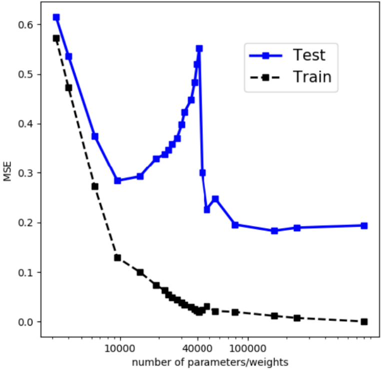
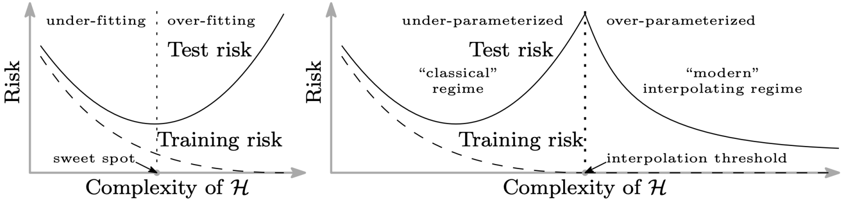


Figure: Experiments on MNIST. Left: [Spigler, Geiger, Ascoli, Sagun, Biroli, Wyart, 2018]. Right: [Belkin, Hsu, Ma, Mandal, 2018]. See also: [Neyshabur, Tomioka, Srebro, 2014a].

Double descent

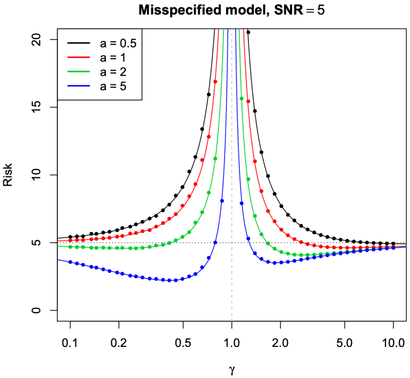


(a) U-shaped "bias-variance" risk curve (b) "double descent" risk curve

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

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

Linear model with random covariates



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

Model: $y = \langle \boldsymbol{x}, \boldsymbol{\beta}_\star \rangle + \varepsilon$, $\boldsymbol{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$.
Loss: $L(\boldsymbol{\beta}) = \hat{\mathbb{E}}[(y - \langle \boldsymbol{x}, \boldsymbol{\beta} \rangle)^2]$

Why singularity?

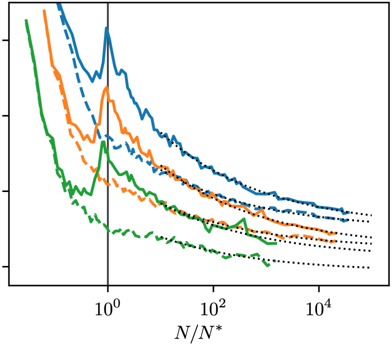
- Model: $\boldsymbol{x}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$, $y_i = \langle \boldsymbol{x}_i, \boldsymbol{\beta}_\star \rangle + \varepsilon_i$, $\boldsymbol{\beta}_\star = \mathbf{0}$, $i \in [n]$.
- Test risk $\propto \mathbb{E}[\|\hat{\beta}\|_2^2] \propto \mathbb{E}[\|\boldsymbol{X}^\dagger \boldsymbol{y}\|_2^2] \propto \mathbb{E}[\mathrm{tr}((\boldsymbol{X}^\top \boldsymbol{X})^\dagger)]$.
- 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.

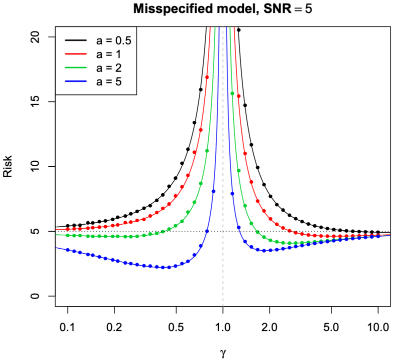
Why singularity?

- Model: $\boldsymbol{x}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$, $y_i = \langle \boldsymbol{x}_i, \boldsymbol{\beta}_\star \rangle + \varepsilon_i$, $\boldsymbol{\beta}_\star = \mathbf{0}$, $i \in [n]$.
- Test risk $\propto \mathbb{E}[\|\hat{\boldsymbol{\beta}}\|_2^2] \propto \mathbb{E}[\|\boldsymbol{X}^\dagger \boldsymbol{y}\|_2^2] \propto \mathbb{E}[\mathrm{tr}((\boldsymbol{X}^\top \boldsymbol{X})^\dagger)]$.
- 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.

Comparison





✓ Peak at the interpolation threshold.

? Monotonic 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.

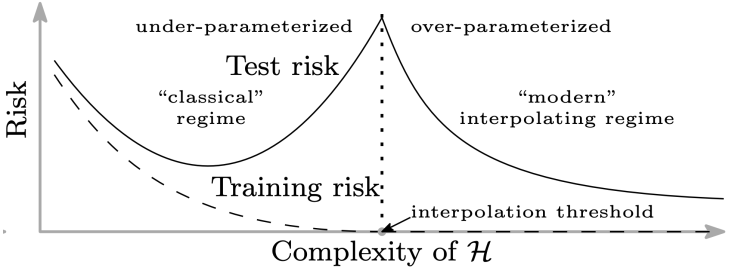


Figure: By [Belkin, Hsu, Ma, Mandal, 2018].

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

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

Setting

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

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

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

- Data: $(\boldsymbol{x}_i)_{i\in[n]} \sim \mathrm{Unif}(\mathbb{S}^{d-1}(\sqrt{d}))$, $y_i = f_\star(\boldsymbol{x}_i) + \varepsilon_i$, $\mathbb{E}[\varepsilon_i^2] = \tau^2$.
- Weights: $(\boldsymbol{w}_j)_{j\in[N]} \sim_{iid} \mathrm{Unif}(\mathbb{S}^{d-1})$.
- Activation σ: $\|\mathrm{P}_1\sigma\|_{L^2}^2 = \mu_1^2$, and $\|\mathrm{P}_{>1}\sigma\|_{L^2}^2 = \mu_\star^2$.
- Tech. ass. on $f_\star$ and σ. Almost every nonlinear $f_\star$ and σ.
- n data, N features, d dimension. $N/d \to \psi_1$, $n/d \to \psi_2$, as $d \to \infty$.

Setting

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

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

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

- Data: $(\boldsymbol{x}_i)_{i\in[n]} \sim \mathrm{Unif}(\mathbb{S}^{d-1}(\sqrt{d}))$, $y_i = f_\star(\boldsymbol{x}_i) + \varepsilon_i$, $\mathbb{E}[\varepsilon_i^2] = \tau^2$.
- Weights: $(\boldsymbol{w}_j)_{j\in[N]} \sim_{iid} \mathrm{Unif}(\mathbb{S}^{d-1})$.
- Activation σ: $\|\mathsf{P}_1\sigma\|_{L^2}^2 = \mu_1^2$, and $\|\mathsf{P}_{>1}\sigma\|_{L^2}^2 = \mu_\star^2$.
- Tech. ass. on $f_\star$ and σ. Almost every nonlinear $f_\star$ and σ.
- n data, N features, d dimension. $N/d \to \psi_1$, $n/d \to \psi_2$, as $d \to \infty$.

Setting

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

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

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

- Data: $(\boldsymbol{x}_i)_{i\in[n]} \sim \mathrm{Unif}(\mathbb{S}^{d-1}(\sqrt{d}))$, $y_i = f_\star(\boldsymbol{x}_i) + \varepsilon_i$, $\mathbb{E}[\varepsilon_i^2] = \tau^2$.
- Weights: $(\boldsymbol{w}_j)_{j\in[N]} \sim_{iid} \mathrm{Unif}(\mathbb{S}^{d-1})$.
- Activation σ: $\|\mathrm{P}_1\sigma\|_{L^2}^2 = \mu_1^2$, and $\|\mathrm{P}_{>1}\sigma\|_{L^2}^2 = \mu_\star^2$.
- Tech. ass. on $f_\star$ and σ. Almost every nonlinear $f_\star$ and σ.
- n data, N features, d dimension. $N/d \to \psi_1$, $n/d \to \psi_2$, as $d \to \infty$.

Setting

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

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

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

- Data: $(\boldsymbol{x}_i)_{i\in[n]} \sim \mathrm{Unif}(\mathbb{S}^{d-1}(\sqrt{d}))$, $y_i = f_\star(\boldsymbol{x}_i) + \varepsilon_i$, $\mathbb{E}[\varepsilon_i^2] = \tau^2$.
- Weights: $(\boldsymbol{w}_j)_{j\in[N]} \sim_{iid} \mathrm{Unif}(\mathbb{S}^{d-1})$.
- Activation σ: $\|\mathsf{P}_1\sigma\|_{L^2}^2 = \mu_1^2$, and $\|\mathsf{P}_{>1}\sigma\|_{L^2}^2 = \mu_\star^2$.
- Tech. ass. on $f_\star$ and σ. Almost every nonlinear $f_\star$ and σ.
- n data, N features, d dimension. $N/d \to \psi_1$, $n/d \to \psi_2$, as $d \to \infty$.

Setting

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

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

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

- Data: $(\boldsymbol{x}_i)_{i\in[n]} \sim \mathrm{Unif}(\mathbb{S}^{d-1}(\sqrt{d}))$, $y_i = f_\star(\boldsymbol{x}_i) + \varepsilon_i$, $\mathbb{E}[\varepsilon_i^2] = \tau^2$.
- Weights: $(\boldsymbol{w}_j)_{j\in[N]} \sim_{iid} \mathrm{Unif}(\mathbb{S}^{d-1})$.
- Activation σ: $\|\mathsf{P}_1\sigma\|_{L^2}^2 = \mu_1^2$, and $\|\mathsf{P}_{>1}\sigma\|_{L^2}^2 = \mu_\star^2$.
- Tech. ass. on $f_\star$ and σ. Almost every nonlinear $f_\star$ and σ.
- n data, N features, d dimension. $N/d \to \psi_1$, $n/d \to \psi_2$, as $d \to \infty$.

Setting

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

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

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

- Data: $(\boldsymbol{x}_i)_{i\in[n]} \sim \mathrm{Unif}(\mathbb{S}^{d-1}(\sqrt{d}))$, $y_i = f_\star(\boldsymbol{x}_i) + \varepsilon_i$, $\mathbb{E}[\varepsilon_i^2] = \tau^2$.
- Weights: $(\boldsymbol{w}_j)_{j\in[N]} \sim_{iid} \mathrm{Unif}(\mathbb{S}^{d-1})$.
- Activation σ: $\|\mathsf{P}_1\sigma\|_{L^2}^2 = \mu_1^2$, and $\|\mathsf{P}_{>1}\sigma\|_{L^2}^2 = \mu_\star^2$.
- Tech. ass. on $f_\star$ and σ. Almost every nonlinear $f_\star$ and σ.
- n data, N features, d dimension. $N/d \to \psi_1$, $n/d \to \psi_2$, as $d \to \infty$.

Precise asymptotics

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

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

Theorem (M. and Montanari, 2019)

Under above assumptions, for any $\lambda > 0$, we have

$$\begin{aligned} R(\hat{\boldsymbol{a}}_\lambda; f_\star) = {}& \|\mathsf{P}_{\rm lin} f_\star\|_{L^2}^2 \cdot \mathscr{B}(\zeta, \psi_1, \psi_2, \lambda/\mu_\star^2) \\ & + (\tau^2 + \|\mathsf{P}_{\rm nl} f_\star\|_{L^2}^2) \cdot \mathscr{V}(\zeta, \psi_1, \psi_2, \lambda/\mu_\star^2) + o_{d,\mathbb{P}}(1), \end{aligned}$$

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

(Similar result for the training error.)

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

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

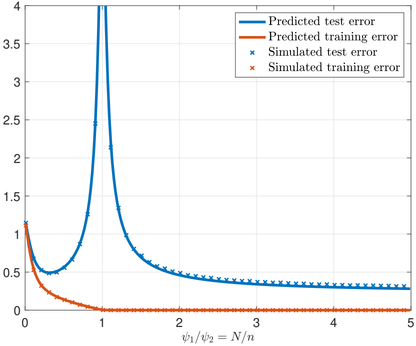
and

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

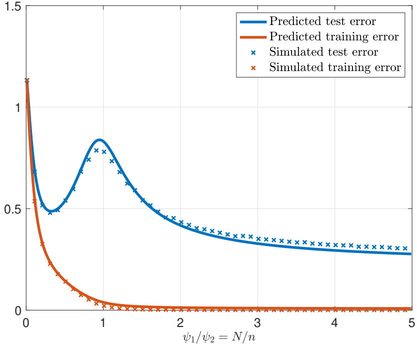
We then have

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

Insights

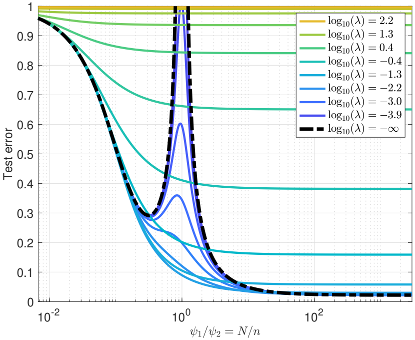


$\lambda = 0+$

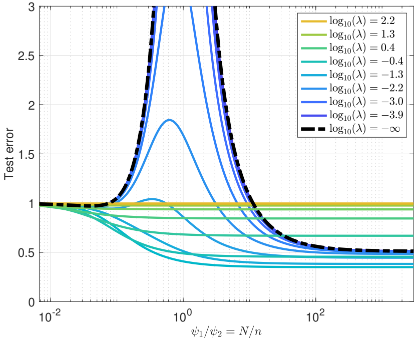


$\lambda = 3 \times 10^{-4}$

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

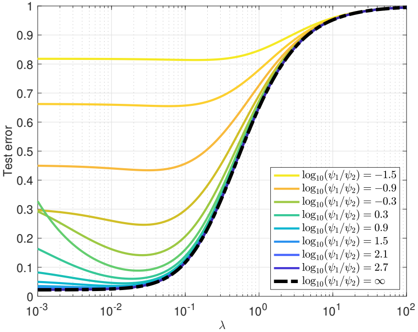


SNR = 5

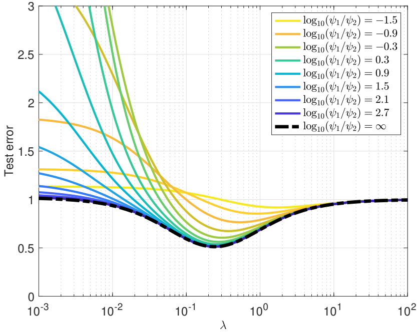


SNR = 1/5

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



SNR = 5



SNR = 1/10

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

Agenda

1. Mean field theory

2. Tangent kernel theory

3. Transitions between mean field and tangent kernel regime

Connections of mean field and tangent kernel

Setup: α controls the speed of change of emp. dist.

$$\text{Prediction function:} \quad \hat{f}_{\alpha,N}(\boldsymbol{x};\boldsymbol{\theta}) = \frac{\alpha}{N}\sum_{j=1}^{N}\sigma_\star(\boldsymbol{x};\boldsymbol{\theta}_j),$$

$$\text{Risk function:} \quad R_{\alpha,N}(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{x}}\Big[\Big(f(\boldsymbol{x}) - \hat{f}_{\alpha,N}(\boldsymbol{x};\boldsymbol{\theta})\Big)^2\Big],$$

$$\text{Gradient flow:} \quad \frac{\mathrm{d}\boldsymbol{\theta}_j^t}{\mathrm{d}t} = -\frac{N}{2\alpha^2}\nabla_{\boldsymbol{\theta}_j} R_{\alpha,N}(\boldsymbol{\theta}^t).$$

The coupled dynamics

Denote $\rho_t^{\alpha,N} = (1/N)\sum_{j=1}^N \delta_{\boldsymbol{\theta}_j^t}$. Distributional dynamics:

$$\partial_t \rho_t^{\alpha,N} = (1/\alpha)\nabla_{\boldsymbol{\theta}} \cdot (\rho_t^{\alpha,N} \nabla_{\boldsymbol{\theta}} \Psi(\boldsymbol{\theta}; \rho_t^{\alpha,N})).$$

Denote $u_t^{\alpha,N}(z) = f(z) - \hat{f}_{\alpha,N}(z;\boldsymbol{\theta}^t)$. Residual dynamics:

$$\partial_t \|u_t^{\alpha,N}\|_{L^2}^2 = -\langle u_t^{\alpha,N}, \mathcal{H}_{\rho_t^{\alpha,N}} u_t^{\alpha,N}\rangle.$$

Here

$$\mathcal{H}_\rho(x,z) \equiv \int \langle \nabla_{\boldsymbol{\theta}}\sigma_\star(x;\boldsymbol{\theta}), \nabla_{\boldsymbol{\theta}}\sigma_\star(z;\boldsymbol{\theta})\rangle \rho(\mathrm{d}\boldsymbol{\theta}),$$

$$\Psi_\alpha(\boldsymbol{\theta};\rho^{\alpha,N}) = -\mathbb{E}_x[u_t^{\alpha,N}(x)\sigma_\star(x;\boldsymbol{\theta})].$$

[Rotskoff, Vanden-Eijnden, 2018], [Chizat, Bach, 2018b]

The coupled dynamics

Denote $\rho_t^{\alpha,N} = (1/N)\sum_{j=1}^N \delta_{\theta_j^t}$. Distributional dynamics:

$$\partial_t \rho_t^{\alpha,N} = (1/\alpha)\nabla_{\theta} \cdot (\rho_t^{\alpha,N} \nabla_{\theta} \Psi(\theta; \rho_t^{\alpha,N})).$$

Denote $u_t^{\alpha,N}(z) = f(z) - \hat{f}_{\alpha,N}(z; \theta^t)$. Residual dynamics:

$$\partial_t \|u_t^{\alpha,N}\|_{L^2}^2 = -\langle u_t^{\alpha,N}, \mathcal{H}_{\rho_t^{\alpha,N}} u_t^{\alpha,N}\rangle.$$

Here

$$\mathcal{H}_\rho(x, z) \equiv \int \langle \nabla_\theta \sigma_\star(x; \theta), \nabla_\theta \sigma_\star(z; \theta)\rangle \rho(\mathrm{d}\theta),$$

$$\Psi_\alpha(\theta; \rho^{\alpha,N}) = -\mathbb{E}_x[u_t^{\alpha,N}(x)\sigma_\star(x; \theta)].$$

[Rotskoff, Vanden-Eijnden, 2018], [Chizat, Bach, 2018b]

The coupled dynamics

Denote $\rho_t^{\alpha,N} = (1/N)\sum_{j=1}^N \delta_{\theta_j^t}$. Distributional dynamics:

$$\partial_t \rho_t^{\alpha,N} = (1/\alpha)\nabla_{\boldsymbol{\theta}} \cdot (\rho_t^{\alpha,N} \nabla_{\boldsymbol{\theta}} \Psi(\boldsymbol{\theta}; \rho_t^{\alpha,N})).$$

Denote $u_t^{\alpha,N}(\boldsymbol{z}) = f(\boldsymbol{z}) - \hat{f}_{\alpha,N}(\boldsymbol{z}; \boldsymbol{\theta}^t)$. Residual dynamics:

$$\partial_t \|u_t^{\alpha,N}\|_{L^2}^2 = -\langle u_t^{\alpha,N}, \mathcal{H}_{\rho_t^{\alpha,N}} u_t^{\alpha,N}\rangle.$$

Here

$$\mathcal{H}_\rho(\boldsymbol{x}, \boldsymbol{z}) \equiv \int \langle \nabla_{\boldsymbol{\theta}} \sigma_\star(\boldsymbol{x}; \boldsymbol{\theta}), \nabla_{\boldsymbol{\theta}} \sigma_\star(\boldsymbol{z}; \boldsymbol{\theta})\rangle \rho(\mathrm{d}\boldsymbol{\theta}),$$

$$\Psi_\alpha(\boldsymbol{\theta}; \rho^{\alpha,N}) = -\mathbb{E}_{\boldsymbol{x}}[u_t^{\alpha,N}(\boldsymbol{x})\sigma_\star(\boldsymbol{x}; \boldsymbol{\theta})].$$

[Rotskoff, Vanden-Eijnden, 2018], [Chizat, Bach, 2018b]

The mean field limit and tangent kernel limit

$$\begin{aligned}\partial_t \rho_t^{\alpha,N} =& (1/\alpha)\nabla_{\boldsymbol{\theta}} \cdot (\rho_t^{\alpha,N}[\nabla_{\boldsymbol{\theta}}\Psi(\boldsymbol{\theta};\rho_t^{\alpha,N})]),\\ \partial_t \|u_t^{\alpha,N}\|_{L^2}^2 =& -\langle u_t^{\alpha,N}, \mathcal{H}_{\rho_t^{\alpha,N}} u_t^{\alpha,N}\rangle.\end{aligned}$$

- The mean field limit: fix $\alpha = O(1)$ and let $N \to \infty$.
- The tangent kernel limit: let $\alpha = \sqrt{N} \to \infty$.
- In tangent kernel limit: the kernel will not change. The res. dynamics becomes self contained. The emp. risk converges to 0.

The mean field limit and tangent kernel limit

$$\partial_t \rho_t^{\alpha,N} = (1/\alpha)\nabla_{\boldsymbol{\theta}} \cdot (\rho_t^{\alpha,N}[\nabla_{\boldsymbol{\theta}}\Psi(\boldsymbol{\theta};\rho_t^{\alpha,N})]),$$
$$\partial_t \|u_t^{\alpha,N}\|_{L^2}^2 = -\langle u_t^{\alpha,N}, \mathcal{H}_{\rho_t^{\alpha,N}} u_t^{\alpha,N}\rangle.$$

- The mean field limit: fix $\alpha = O(1)$ and let $N \to \infty$.
- The tangent kernel limit: let $\alpha = \sqrt{N} \to \infty$.
- In tangent kernel limit: the kernel will not change. The res. dynamics becomes self contained. The emp. risk converges to 0.

The mean field limit and tangent kernel limit

$$\partial_t \rho_t^{\alpha,N} = (1/\alpha)\nabla_{\boldsymbol{\theta}} \cdot (\rho_t^{\alpha,N}[\nabla_{\boldsymbol{\theta}}\Psi(\boldsymbol{\theta};\rho_t^{\alpha,N})]),$$
$$\partial_t \|u_t^{\alpha,N}\|_{L^2}^2 = -\langle u_t^{\alpha,N}, \mathcal{H}_{\rho_t^{\alpha,N}} u_t^{\alpha,N}\rangle.$$

- The mean field limit: fix $\alpha = O(1)$ and let $N \to \infty$.
- The tangent kernel limit: let $\alpha = \sqrt{N} \to \infty$.
- In tangent kernel limit: the kernel will not change. The res. dynamics becomes self contained. The emp. risk converges to 0.

Benefits and limitations of the mean field theory

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

Benefits:

- It captures the non-linear behavior of neural networks that potentially give better generalization.
- People believe in practice we are in this regime.

Limitations:

- Hard to prove convergence of PDE.
- Hard to generalize to multi-layers.

Benefits and limitations of the mean field theory

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

Benefits:

- It captures the non-linear behavior of neural networks that potentially give better generalization.
- People believe in practice we are in this regime.

Limitations:

- Hard to prove convergence of PDE.
- Hard to generalize to multi-layers.

Benefits and limitations of the tangent kernel theory

Benefits:

- Give provable convergence results for multi-layers neural networks.
- Exhibit many similar behaviors of neural networks: double-descent.
- Easy to use in many scenarios: ResNet, convolutional NN, graph NN, recurrent NN.

Limitations

- The intuition of fixed weights distribution is not realistic.
- The generalization is not as good as fully trained neural networks.
- Everything is just a kernel?

Benefits and limitations of the tangent kernel theory

Benefits:

- Give provable convergence results for multi-layers neural networks.
- Exhibit many similar behaviors of neural networks: double-descent.
- Easy to use in many scenarios: ResNet, convolutional NN, graph NN, recurrent NN.

Limitations

- The intuition of fixed weights distribution is not realistic.
- The generalization is not as good as fully trained neural networks.
- Everything is just a kernel?

Thanks!