Towards a Mathematical Understanding of Supervised Learning: what we know and what we don't

Weinan E

Princeton University

Joint work with **Chao Ma, Stephan Wojtowytsch, Lei Wu**

Slides can be found in: `www.math.princeton.edu/~weinan`

Neural network-based machine learning is both very **powerful** and very **fragile**.

- What are the reasons behind?
- How can we do better (more robust formulation)?

Basic problem of supervised learning (regression)

Given $S = \{(\boldsymbol{x}_j, y_j = f^*(\boldsymbol{x}_j)), j \in [n]\}$, learn (i.e. approximate) f^*.

- assume $\boldsymbol{x}_j \in X = [0,1]^d$, $\mu =$ the distribution of $\{\boldsymbol{x}_j\}$

Standard procedure:

1. choose a hypothesis space (set of trial functions) $\mathcal{H}_m$
   - neural network models
2. choose a loss function (to fit the data)
   - "empirical risk"

$$\hat{\mathcal{R}}_n(f) = \frac{1}{n}\sum_j (f(\boldsymbol{x}_j) - y_j)^2 = \frac{1}{n}\sum_j (f(\boldsymbol{x}_j) - f^*(\boldsymbol{x}_j))^2$$

3. choose an optimization algorithm and the hyper-parameters
   - gradient descent (GD), stochastic gradient descent (SGD), ADAM, RMSprop, ...

Objective: Minimize the "population risk" (the "generalization error")

$$\mathcal{R}(f) = \mathbb{E}_{\boldsymbol{x}\sim\mu}(f(\boldsymbol{x}) - f^*(\boldsymbol{x}))^2 = \int_{\mathbb{R}^d} (f(\boldsymbol{x}) - f^*(\boldsymbol{x}))^2 d\mu$$

Three important aspects to study

- hypothesis space: what kind of functions can be approximated efficiently, generalization gap (= difference between training and testing errors)
- loss function (the variational problem): landscape?
- training: can we optimize? does the solution generalize?

Important parameters:

- $m =$: number of free parameters
- $n =$: size of training dataset
- $t =$: training steps
- $d =$: dimensionality

typically interested in the case: $m, n, t \to \infty, d \gg 1$.

Main goal: Error estimates

$$\mathcal{R}(f_{m,n,t}) \lesssim m^{-\alpha} + n^{-\beta} + t^{-\gamma}$$

Want: free of **curse of dimensionality (CoD):** α, β, γ are independent of d.

Given a class of hypothesis space (e.g. two-layer neural networks):

1. What class of functions can be approximated by that model without CoD?

Looking for some function spaces. These are the analog of Besov spaces.

2. Generalization gap for this function class?

Bounds on Rademacher complexity.

Function spaces for ML models

- Random feature model: RKHS (reproducing kernel Hilbert space, ...)
- Two-layer neural networks: Barron space (Bach (2017), E, Ma and Wu (2018, 2019))
- ResNets: Flow-induced space (E, Ma and Wu (2019))
- Multi-layer neural networks: Multi-layer spaces (E and Wojtowytsch (2020))

Looking for the right function spaces

- Given a type of hypothesis space $\mathcal{H}_m$, identify the natural function space associated with them (in particular, identify a norm Identify $\|f^*\|_*$)
  - Direct approximation theorem (with Monte Carlo rate):

$$\inf_{f \in \mathcal{H}_m} \mathcal{R}(f) = \inf_{f \in \mathcal{H}_m} \|f - f^*\|^2_{L^2(d\mu)} \lesssim \frac{\|f^*\|^2_*}{m}$$

  - Inverse approximation theorem: If a function f^* can be approximated efficiently by the functions in $\mathcal{H}_m$, as $m \to$ with some uniform bounds, then $\|f^*\|_*$ is finite.
- Study the generalization gap for this function space. One way to do this is to study the Rademacher complexity of the set $\mathcal{H}_Q = \{f, \|f\|_* \leq Q\}$.
  - Ideally, we would like to have ($\hat{f}$ = output of ML model):

$$\mathrm{Rad}_S(\mathcal{H}_Q) \lesssim \frac{Q}{\sqrt{n}}$$

Combined: Up to log terms, we have

$$\mathcal{R}(\hat{f}) \lesssim \frac{\|f^*\|^2_*}{m} + \frac{\|f^*\|_*}{\sqrt{n}}$$

Two-layer neural network model: Barron spaces

E, Ma and Wu (2018, 2019), Bach (2017) [i]

$$\mathcal{H}_m = \{f_m(\boldsymbol{x}) = \frac{1}{m}\sum_j a_j \sigma(\boldsymbol{w}_j^T \boldsymbol{x})\},\ \theta = \{(a_j, \boldsymbol{w}_j), j \in [m]\}$$

Consider the function $f : X = [0,1]^d \mapsto \mathbb{R}$ of the following form

$$f(\boldsymbol{x}) = \int_\Omega a\sigma(\boldsymbol{w}^T\boldsymbol{x})\rho(da, d\boldsymbol{w}) = \mathbb{E}_{(a,\boldsymbol{w})\sim\rho}[a\sigma(\boldsymbol{w}^T\boldsymbol{x})]\}, \quad \boldsymbol{x} \in X$$

$\Omega = \mathbb{R}^1 \times \mathbb{R}^{d+1}$, ρ is a probability distribution on Ω.

$$\|f\|_{\mathcal{B}} = \inf_{\rho \in P_f} \left(\mathbb{E}_\rho[a^2 \|\boldsymbol{w}\|_1^2]\right)^{1/2}$$

where $P_f := \{\rho : f(\boldsymbol{x}) = \mathbb{E}_\rho[a\sigma(\boldsymbol{w}^T\boldsymbol{x})]\}$.

$$\mathcal{B} = \{f \in C^0 : \|f\|_{\mathcal{B}} < \infty\}$$

[i] Related work in Barron (1993), Klusowski and Barron (2016), E and Wojtowytsch (2020)

Theorem (Direct Approximation Theorem)

$$\|f - f_m\|_{L^2(X)} \lesssim \frac{\|f\|_{\mathcal{B}}}{\sqrt{m}}$$

Theorem (Inverse Approximation Theorem)

Let

$$\mathcal{N}_C \overset{def}{=} \{ \frac{1}{m} \sum_{k=1}^{m} a_k \sigma(\boldsymbol{w}_k^T \boldsymbol{x}) : \frac{1}{m} \sum_{k=1}^{m} |a_k|^2 \|\boldsymbol{w}_k\|_1^2 \leq C^2, m \in \mathbb{N}^+ \}.$$

Let f^* be a continuous function. Assume there exists a constant C and a sequence of functions $f_m \in \mathcal{N}_C$ such that

$$f_m(\boldsymbol{x}) \to f^*(\boldsymbol{x})$$

for all $\boldsymbol{x} \in X$, then there exists a probability distribution ρ^* on Ω, such that

$$f^*(\boldsymbol{x}) = \int a\sigma(\boldsymbol{w}^T \boldsymbol{x}) \rho^*(da, d\boldsymbol{w}),$$

for all $\boldsymbol{x} \in X$ and $\|f^*\|_{\mathcal{B}} \leq C$.

Complexity estimates

Theorem (Bach, 2017)

Let $\mathcal{F}_Q = \{f \in \mathcal{B}, \|f\|_{\mathcal{B}} \leq Q\}$. Then we have

$$\mathrm{Rad}_S(\mathcal{F}_Q) \leq 2Q\sqrt{\frac{2\ln(2d)}{n}}$$

A priori estimates for regularized model

$$\mathcal{L}_n(\theta) = \hat{\mathcal{R}}_n(\theta) + \lambda\sqrt{\frac{\log(2d)}{n}}\|\theta\|_{\mathcal{P}}, \qquad \hat{\theta}_n = \text{argmin } \mathcal{L}_n(\theta)$$

where the path norm is defined by:

$$\|\theta\|_{\mathcal{P}} = \left(\frac{1}{m}\sum_{k=1}^{m}|a_k|^2\|\boldsymbol{w}_k\|_1^2\right)^{1/2}$$

Theorem (E, Ma, Wu, 2018)

Assume $f^* : X \mapsto [0,1] \in \mathcal{B}$. There exist constants C_0, such that for any $\delta > 0$, if $\lambda \geq C_0$, then with probability at least $1-\delta$ over the choice of training set, we have

$$\mathcal{R}(\hat{\theta}_n) \lesssim \frac{\|f^*\|_{\mathcal{B}}^2}{m} + \lambda\|f^*\|_{\mathcal{B}}\sqrt{\frac{\log(2d)}{n}} + \sqrt{\frac{\log(1/\delta)+\log(n)}{n}}.$$

Other models

1. Random feature model

$\{\phi(\cdot;\boldsymbol{w})\}$: collection of random features, e.g. $\phi(\boldsymbol{x},\boldsymbol{w}) = \sigma(\boldsymbol{w}^T\boldsymbol{x})$.
π: prob distribution of the random variable $\boldsymbol{w}$.

Hypothesis space: Given any realization $\{\boldsymbol{w}_j\}_{j=1}^m$, i.i.d. with distribution π

$$\mathcal{H}_m(\{\boldsymbol{w}_j\}) = \{f_m(\boldsymbol{x},\boldsymbol{a}) = \frac{1}{m}\sum_{j=1}^m a_j\phi(\boldsymbol{x};\boldsymbol{w}_j)\}.$$

Corresponding function space: reproducing kernel Hilbert space (RKHS) with kernel:

$$k(\boldsymbol{x},\boldsymbol{x}') = \mathbb{E}_{\boldsymbol{w}\sim\pi}[\phi(\boldsymbol{x};\boldsymbol{w})\phi(\boldsymbol{x}';\boldsymbol{w})]$$

Consider the regularized model:

$$\mathcal{L}_{n,\lambda}(\boldsymbol{a}) = \hat{\mathcal{R}}_n(\boldsymbol{a}) + \frac{\lambda}{\sqrt{n}}\frac{\|\boldsymbol{a}\|}{\sqrt{m}}, \qquad \hat{\boldsymbol{a}}_n = \text{argmin } \mathcal{L}_{n,\lambda}(\boldsymbol{a})$$

Up to log terms

$$\mathcal{R}(\hat{\boldsymbol{a}}_n) \lesssim \frac{\|f^*\|_{\mathcal{H}}^2}{m} + \lambda\frac{\|f^*\|_{\mathcal{H}}}{\sqrt{n}}$$

$$
\begin{aligned}
\boldsymbol{z}_{0,L}(\boldsymbol{x}) &= \boldsymbol{x}, \\
\boldsymbol{z}_{l+1,L}(\boldsymbol{x}) &= z_{l,L}(\boldsymbol{x}) + \frac{1}{L}\boldsymbol{U}_l \sigma \circ (\boldsymbol{W}_l z_{l,L}(\boldsymbol{x})), \quad l = 0, 1, \cdots, L-1 \\
f(\boldsymbol{x}, \theta) &= \alpha \cdot \boldsymbol{z}_{L,L}(\boldsymbol{x})
\end{aligned}
$$

Corresponding function space: "flow-induced function space" (E, Ma and Wu (2019))
Regularized loss function:

$$
\mathcal{L}_{n,\lambda}(\theta) = \hat{\mathcal{R}}_n(\theta) + \lambda \|\theta\|_{\mathcal{P}} \sqrt{\frac{2\log(2d)}{n}}.
$$

Up to logarithmic terms, we have:

$$
\mathcal{R}(\hat{\theta}) \lesssim \frac{\|f^*\|_{\mathcal{D}}^2}{L} + \lambda \frac{\|f^*\|_{\mathcal{D}}^2}{\sqrt{n}}
$$

Variance reduction:

$$
\|f^*\|_{\mathcal{D}} \leq \|f^*\|_{\mathcal{B}} \leq \|f^*\|_{\mathcal{H}}
$$

3. Multilayer networks

$$f(x) = \sum_{i_L=1}^{m_L} a_{i_L}^{L} \sigma \left(\sum_{i_{L-1}=1}^{m_{L-1}} a_{i_L i_{L-1}}^{L-1} \sigma \left(\dots \sigma \left(\sum_{i_1=1}^{m_1} a_{i_2 i_1}^{1} \sigma \left(\sum_{i_0=1}^{d+1} a_{i_1 i_0}^{0} x_{i_0} \right) \right) \right) \right)$$

Corresponding function space: "multilayer space" (E and Wojtowytsch (in preparation))

- Rademacher complexity/generalization gap: Same as for Barron space. Let $\mathcal{F}_Q = \{f \in C^0, \|f\|_{\text{multi-layer}} \leq Q\}$. Then $\mathrm{Rad}_S(\mathcal{F}_Q) \leq 2Q\sqrt{\frac{2\ln(2d+2)}{n}}$
- Inverse approximation theorem holds.
- Direct approximation theorem: holds, but not with Monte-Carlo rate. For f^* in multi-layer space and $m \in \mathbb{N}$, there exists a network f with layers of width $m_\ell = m^{L-\ell+1}$ such that

$$\|f - f^*\|_{L^2(\mathbb{P})} \lesssim \frac{2^L \, \|f^*\|_{\text{multi-layer}}}{m^{1/(4L-2)}}.$$

Representation of functions for different ML model:

- Random feature model:

$$f(\boldsymbol{x}) = \mathbb{E}_{\pi}[a(\boldsymbol{w})\sigma(\boldsymbol{w}^T\boldsymbol{x})]$$

- Two-layer neural networks: Barron space

$$f(\boldsymbol{x}) = \mathbb{E}_{(a,\boldsymbol{w})\sim\rho}[a\sigma(\boldsymbol{w}^T\boldsymbol{x})]$$

- ResNets: Flow-induced space

$$f(\boldsymbol{x}) = \alpha \cdot \boldsymbol{z}(1),\ \dot{\boldsymbol{z}} = \mathbb{E}_{\{(\boldsymbol{a},\boldsymbol{w})\sim\rho_\tau\}}\boldsymbol{a}\sigma(\boldsymbol{w}^T\boldsymbol{z}),\ \boldsymbol{z}(0) = \boldsymbol{x}$$

- Multi-layer neural networks: Multi-layer spaces

$$f(\boldsymbol{x}) = \mathbb{E}_{\theta_L\sim\pi_L} a^{(L)}_{\theta_L}\sigma(\mathbb{E}_{\theta_{L-1}\sim\pi_{L-1}}\dots\sigma(\mathbb{E}_{\theta_1\sim\pi_1} a^1_{\theta_2,\theta_1}\sigma(a^0_{\theta_1}\cdot\boldsymbol{x}))\dots)$$

What's not known?

- Multi-layer spaces: The approximation error is not optimal. Can it be improved?
- Function spaces for CNNs
- Function space for DenseNets

Training dynamics

Optimization:

Does the training process converge to a good solution? How fast?

Generalization:

In particular, is there such thing as "implicit regularization"?

Mean-field formulation

Chizat and Bach (2018), Mei, Montanari and Nguyen (2018), Rotskoff and Vanden-Eijnden (2018), Sirignano and Spiliopoulos (2018)

$$\mathcal{H}_m = \{f_m(\boldsymbol{x}) = \frac{1}{m}\sum_j a_j \sigma(\boldsymbol{w}_j^T \boldsymbol{x})\}$$

Let

$$I(\boldsymbol{u}_1, \cdots, \boldsymbol{u}_m) = \hat{\mathcal{R}}_n(f_m), \quad \boldsymbol{u}_j = (a_j, \boldsymbol{w}_j)$$

GD dynamics:

$$\frac{d\boldsymbol{u}_j}{dt} = -\nabla_{\boldsymbol{u}_j} I(\boldsymbol{u}_1, \cdots, \boldsymbol{u}_m), \quad \boldsymbol{u}_j(0) = \boldsymbol{u}_j^0, \quad j \in [m]$$

Lemma:. Let

$$\rho(\boldsymbol{u}, t) = \frac{1}{m}\sum_j \delta_{\boldsymbol{u}_j(t)}$$

then the GD dynamics described above is equivalent to:

$$\partial_t \rho = \nabla(\rho \nabla V), \quad V = \frac{\delta \hat{\mathcal{R}}_n}{\delta \rho}$$

This is the gradient flow of $\hat{\mathcal{R}}_n$ under the Wasserstein metric.

Convergence to global minima

The functional which is not *displacement convex*, but:

Theorem (Chizat-Bach '18, '20, Wojtowytsch '20)

Let ρ_t be a solution of the Wasserstein gradient flow such that

- *ρ_0 has a density on the cone $\Theta := \{|a|^2 \leq |w|^2\}$.*
- *ρ_0 is* omni-directional*: Every open cone in Θ has positive measure with respect to ρ_0*

Then the following are equivalent.

1. *The velocity potentials $V = \frac{\delta \mathcal{R}}{\delta \rho}(\rho_t, \cdot)$ converge to a unique limit as $t \to \infty$.*
2. *$\mathcal{R}(\rho_t) \to 0$, as $t \to \infty$.*

1. There are further technical conditions for the theorem to hold.
2. Convergence of subsequences of $\frac{\delta \mathcal{R}}{\delta \rho}(\rho_t, \cdot)$ is guaranteed by compactness.

Training two-layer neural networks under conventional scaling

$$f_m(\boldsymbol{x}; \boldsymbol{a}, \boldsymbol{B}) = \sum_{j=1}^{m} a_j \sigma(\boldsymbol{b}_j^T \boldsymbol{x}) = \boldsymbol{a}^T \sigma(\boldsymbol{B}\boldsymbol{x}),$$

"Xavier-like" initialization:

$$a_j(0) \sim \mathcal{N}(0, \beta^2), \qquad \boldsymbol{b}_j(0) \sim \mathcal{N}(0, I/d)$$

$\beta = 0$ or $1/\sqrt{m}$

The associated *random feature model:* $\{\boldsymbol{b}_j\}$ frozen, only train $\{a_i\}$

Define Gram matrix $K = (K_{ij})$:

$$K_{i,j} = \frac{1}{n} \mathbb{E}_{\boldsymbol{b} \sim \pi_0} [\sigma(\boldsymbol{x}_i^T \boldsymbol{b}) \sigma(\boldsymbol{x}_j^T \boldsymbol{b})].$$

Highly over-parametrized regime

- Good news: Exponential convergence (Du et al (2018))
- Bad news: Converged solution is no better than that of the random feature model (E, Ma, Wu (2019), Arora et al (2019), ...)
  Heuristics given in Jacot, Gabriel and Hongler (2018): "neural tangent kernel"

Theorem

Let $\lambda_n = \lambda_{\min}(K)$ *and assume* $\beta = 0$. *Denote by* $f_m(\boldsymbol{x}; \tilde{\boldsymbol{a}}(t), \boldsymbol{B}_0))$ *the solutions of GD dynamics for the random feature model. For any* $\delta \in (0,1)$, *assume that* $m \gtrsim n^2 \lambda_n^{-4} \delta^{-1} \ln(n^2 \delta^{-1})$. *Then with probability at least* $1 - 6\delta$ *we have*

$$\hat{\mathcal{R}}_n(\boldsymbol{a}(t), \boldsymbol{B}(t)) \leq e^{-m\lambda_n t} \hat{\mathcal{R}}_n(\boldsymbol{a}(0), \boldsymbol{B}(0))$$

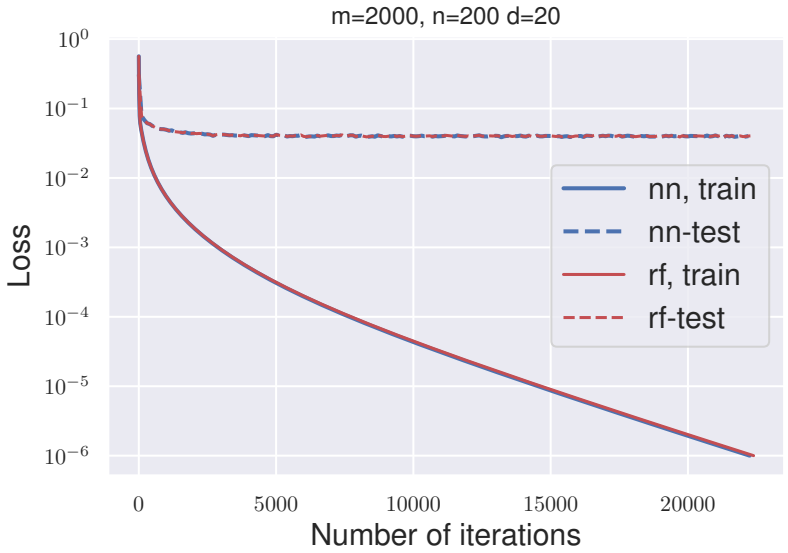
$$\sup_{\boldsymbol{x} \in \mathcal{S}^{d-1}} |f_m(\boldsymbol{x}; \boldsymbol{a}(t), \boldsymbol{B}(t)) - f_m(\boldsymbol{x}; \tilde{\boldsymbol{a}}(t), \boldsymbol{B}_0)| \lesssim \frac{(1 + \sqrt{\ln(\delta^{-1})})^2 \lambda_n^{-1}}{\sqrt{m}}.$$

Disappointing! No "implicit regularization" in this regime, since

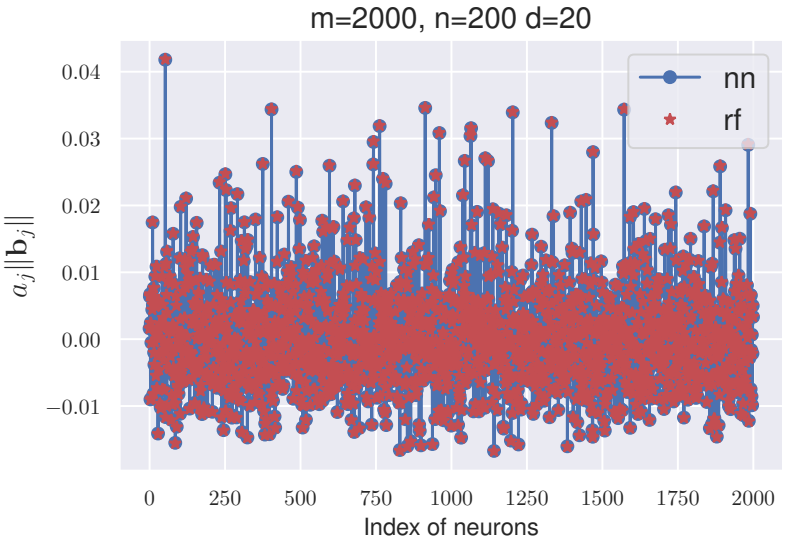
- with explicit regularization, the generalization error is small for all Barron functions.
- without explicit regularization, the generalization error is small only for RKHS functions.

What happens in practice?

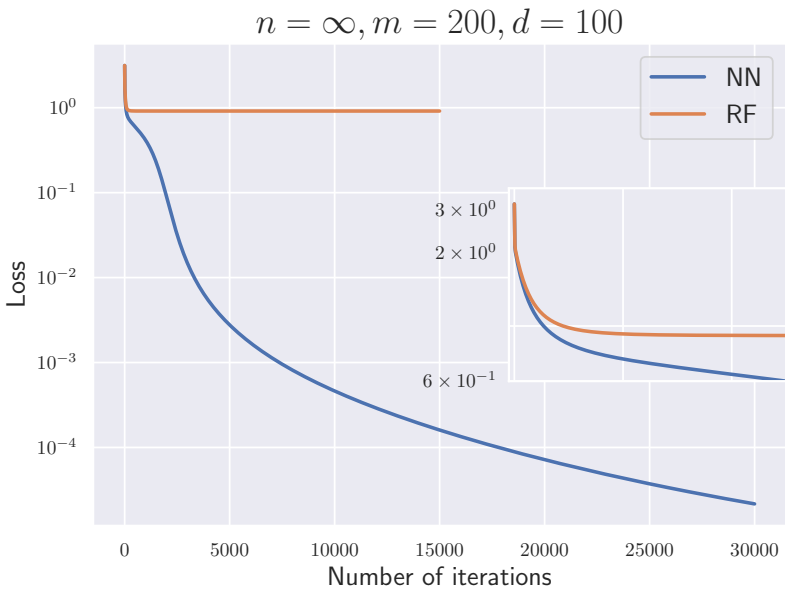
$f^*(\boldsymbol{x}) = \sigma(\boldsymbol{e}_1 \cdot \boldsymbol{x})$ (see Ma, Wu and E (2020) for more results).

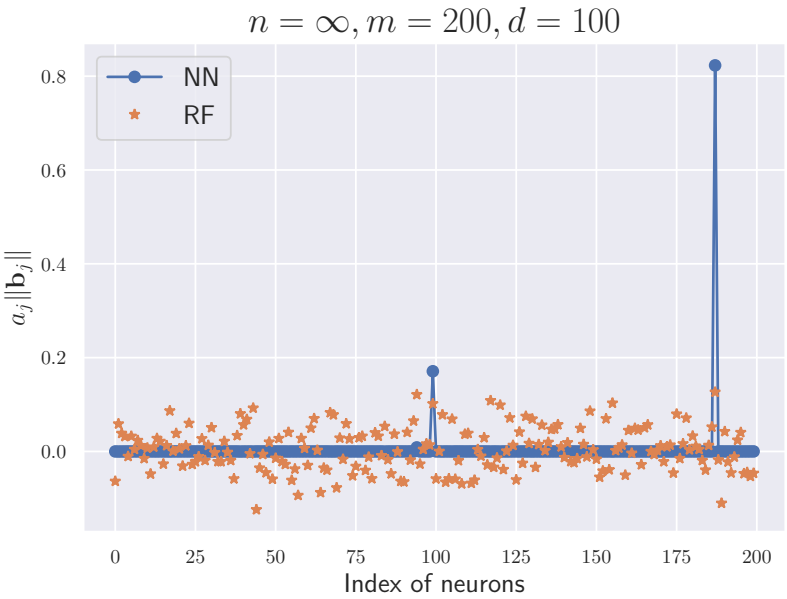


(a)

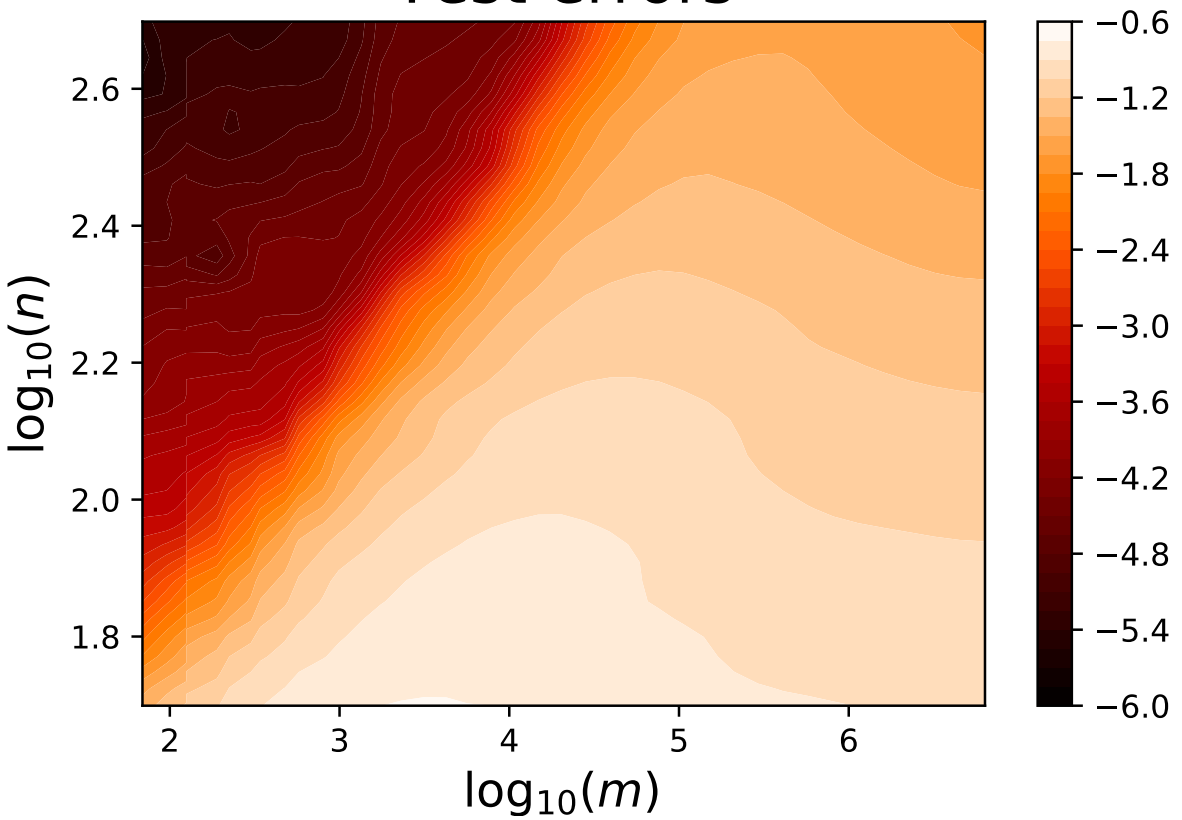


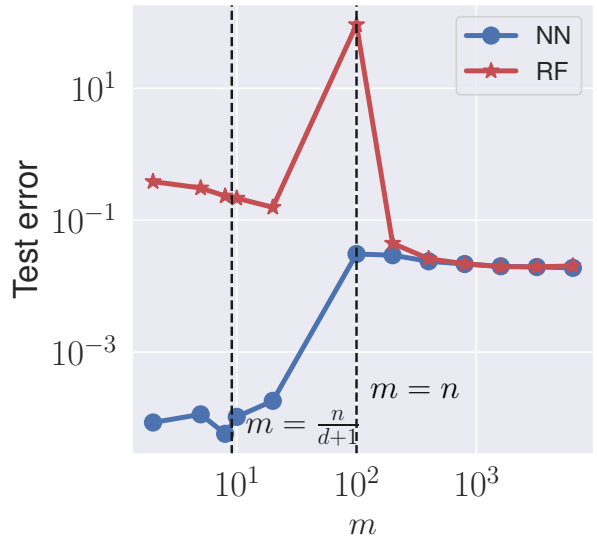
(b)

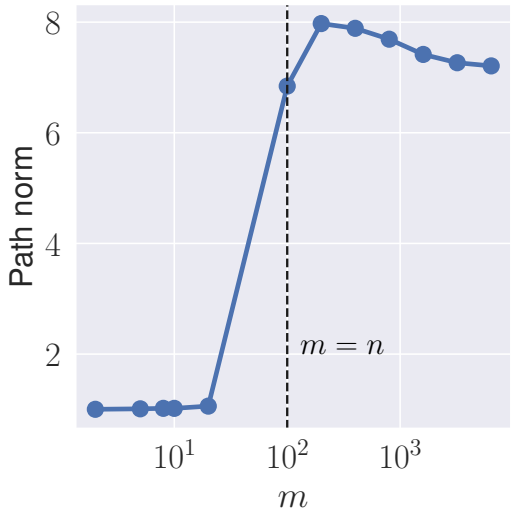




Test errors


$\log_{10}(n)$
2.6
2.4
2.2
2.0
1.8
2
3
4
5
6
$\log_{10}(m)$
−0.6
−1.2
−1.8
−2.4
−3.0
−3.6
−4.2
−4.8
−5.4
−6.0


Test error
10^{1}
10^{-1}
10^{-3}
NN
RF
$m = n$
$m = \frac{n}{d+1}$
10^{1}
10^{2}
10^{3}
m


Path norm
8
6
4
2
$m = n$
10^{1}
10^{2}
10^{3}
m

Neural network-like vs. random feature-like behavior

- Neural network-like behavior
  - two phases: initial random feature-like phase, followed by a second phase with quenching and activation

    Neurons are divided into two groups: activated ones and background neurons
  - testing error continues to decay after the first phase
- Random feature-like behavior
  - simpler dynamics (e.g. no division into two groups)
  - testing error saturates quickly while training error continues to decay

This is observed for target functions which can be accurately approximated by a small number of neurons (effectively "over-parametrized").

What we don't know?

Why does GD/SGD converge to good minima?

- mean-field for continuum of neurons: This is a clean analysis problem.
- mean-field for finite neurons: No rigorous results yet.
- conventional scaling: Does there exist a regime with implicit regularization?

"Well-posed" formulations of ML?

Start with a "nice" continuous problem and discretize to get practical ML algorithms

- representation of functions
- the variational problem for minimizing the population risk (loss function)
- gradient flow for the variational problem (training, a PDE-like problem)

Key: The variational problem should be "nice".

Representation of functions: An illustrative example

Traditional approach:

$$f(\boldsymbol{x}) = \int_{\mathbb{R}^d} a(\boldsymbol{\omega}) e^{i(\boldsymbol{\omega},\boldsymbol{x})} d\boldsymbol{\omega}, \quad f_m(\boldsymbol{x}) = \frac{1}{m} \sum_j a(\boldsymbol{\omega}_j) e^{i(\boldsymbol{\omega}_j,\boldsymbol{x})}$$

$\{\boldsymbol{\omega}_j\}$ is a fixed grid, e.g. uniform.

$$\|f - f_m\|_{L^2(X)} \leq C_0 m^{-\alpha/d} \|f\|_{H^\alpha(X)}$$

"New" approach (π = probability measure on $\mathbb{R}^d$):

$$f(\boldsymbol{x}) = \int_{\mathbb{R}^d} a(\boldsymbol{\omega}) e^{i(\boldsymbol{\omega},\boldsymbol{x})} \pi(d\boldsymbol{\omega}) = \mathbb{E}_{\boldsymbol{\omega}\sim\pi} a(\boldsymbol{\omega}) e^{i(\boldsymbol{\omega},\boldsymbol{x})}$$

Let $\{\boldsymbol{\omega}_j\}$ be an i.i.d. sample of π.

$$\mathbb{E}|f(\boldsymbol{x}) - \frac{1}{m} \sum_{j=1}^{m} a(\boldsymbol{\omega}_j) e^{i(\boldsymbol{\omega}_j,\boldsymbol{x})}|^2 = \frac{\mathsf{var}(f)}{m}$$

$\frac{1}{m} \sum_{j=1}^{m} a(\boldsymbol{\omega}_j) e^{i(\boldsymbol{\omega}_j,\boldsymbol{x})}$ = two-layer neural network with activation function $\sigma(z) = e^{iz}$.

Integral transform-based representation and the variational problem

Consider the (parametric) representation:

$$f(\boldsymbol{x},\theta)=\int_{\mathbb{R}^d} a(\boldsymbol{w})\sigma(\boldsymbol{w}^T\boldsymbol{x})\pi(d\boldsymbol{w})=\mathbb{E}_{\boldsymbol{w}\sim\pi}a(\boldsymbol{w})\sigma(\boldsymbol{w}^T\boldsymbol{x})=\mathbb{E}_{(a,\boldsymbol{w})\sim\rho}a\sigma(\boldsymbol{w}^T\boldsymbol{x})$$

$\theta=$ parameter:

- $\theta=\{a(\cdot)\}$, π is given
- $\theta=\{a(\cdot),\pi(\cdot)\}$, or equivalently $\theta=\{\rho(\cdot)\}$.

Given a target function f^*, the variational problem for minimizing the population risk:

$$\min_{\theta}\mathcal{R},\quad \mathcal{R}(\theta)=\mathbb{E}_{\boldsymbol{x}\sim\mu}(f(\boldsymbol{x},\theta)-f^*(\boldsymbol{x}))^2$$

Conjecture: This is a "nice (convex-like)" variational problem.

Gradient flow for the population risk

Population risk: $\mathcal{R}(f) = \mathbb{E}_{\boldsymbol{x}\sim\mu}(f(\boldsymbol{x}) - f^*(\boldsymbol{x}))^2 =$ "free energy"

$$f(\boldsymbol{x}) = \int a(\boldsymbol{w})\sigma(\boldsymbol{w}^T\boldsymbol{x})\pi(d\boldsymbol{w}) = \mathbb{E}_{\boldsymbol{w}\sim\pi}a(\boldsymbol{w})\sigma(\boldsymbol{w}^T\boldsymbol{x})$$

Follow Halperin and Hohenberg (1977)

- $a =$ non-conserved, use "model A" dynamics (Allen-Cahn):

$$\frac{\partial a}{\partial t} = -\frac{\delta \mathcal{R}}{\delta a}$$

- $\pi =$ conserved (probability density), use "model B" (Cahn-Hilliard):

$$\frac{\partial \pi}{\partial t} + \nabla \cdot \mathbf{J} = 0$$

$$\mathbf{J} = \pi \boldsymbol{v},\ \boldsymbol{v} = -\nabla V,\ V = \frac{\delta \mathcal{R}}{\delta \pi}.$$

Examples

- $\theta = \{a\}$ (fix π, non-conservative).

$$\partial_t a(\boldsymbol{w}, t) = -\frac{\delta \mathcal{R}}{\delta a}(\boldsymbol{w}, t) = -\int a(\tilde{\boldsymbol{w}}, t) K(\boldsymbol{w}, \tilde{\boldsymbol{w}}) \pi(d\tilde{\boldsymbol{w}}) + \tilde{f}(\boldsymbol{w})$$

$$K(\boldsymbol{w}, \tilde{\boldsymbol{w}}) = \mathbb{E}_{\boldsymbol{x}}[\sigma(\boldsymbol{w}^T \boldsymbol{x}) \sigma(\tilde{\boldsymbol{w}}^T \boldsymbol{x})], \quad \tilde{f}(\boldsymbol{w}) = \mathbb{E}_{\boldsymbol{x}}[f^*(\boldsymbol{x}) \sigma(\boldsymbol{w}^T \boldsymbol{x})]$$

This is an integral equation with a symmetric positive definite kernel.

- $\theta = \{\rho\}$ (conservative)

$$\partial_t \rho = \nabla(\rho \nabla V)$$

$$V(\boldsymbol{u}) = \frac{\delta \mathcal{R}}{\delta \rho}(\boldsymbol{u}) = \int \tilde{K}(\boldsymbol{u}, \tilde{\boldsymbol{u}}) \rho(d\tilde{\boldsymbol{u}}) - \tilde{f}(\boldsymbol{u})$$

This is the same as the mean-field equation.

Discretizing the gradient flows

- Discretizing the population risk (into the empirical risk) using data
- Discretizing the gradient flow
  - particle method – the dynamic version of Monte Carlo
  - smoothes particle method – analog of vortex blob method
  - spectral method – very effective in low dimensions

Particle method for the feature-based model

Continuous problem:

$$\partial_t a(\boldsymbol{w}, t) = -\frac{\delta \mathcal{R}}{\delta a}(\boldsymbol{w}) = -\int a(\tilde{\boldsymbol{w}}, t) K(\boldsymbol{w}, \tilde{\boldsymbol{w}}) \pi(d\tilde{\boldsymbol{w}}) + \tilde{f}(\boldsymbol{w})$$

$$\pi(d\boldsymbol{w}) \sim \frac{1}{m} \sum_j \delta_{\boldsymbol{w}_j}, a(\boldsymbol{w}_j, t) \sim a_j(t)$$

Discretized version:

$$\frac{d}{dt} a_j(t) = -\frac{1}{m} \sum_k K(\boldsymbol{w}_j, \boldsymbol{w}_k) a_k(t) + \tilde{f}(\boldsymbol{w}_j)$$

This is exactly the GD for the random feature model.

$$f(\boldsymbol{x}) \sim f_m(\boldsymbol{x}) = \frac{1}{m} \sum_j a_j \sigma(\boldsymbol{w}_j^T \boldsymbol{x})$$

Discretization of the conservative flow

Consider the integral differential equation (IDE):

$$\partial_t \rho = \nabla(\rho \nabla V)$$

Particle method discretization:

$$\rho(a, \boldsymbol{w}, t) \sim \frac{1}{m} \sum_j \delta_{(a_j(t), \boldsymbol{w}_j(t))} = \frac{1}{m} \sum_j \delta_{\boldsymbol{u}_j(t)}$$

gives rise to

$$\frac{d\boldsymbol{u}_j}{dt} = -\nabla_{\boldsymbol{u}_j} I(\boldsymbol{u}_1, \cdots, \boldsymbol{u}_m)$$

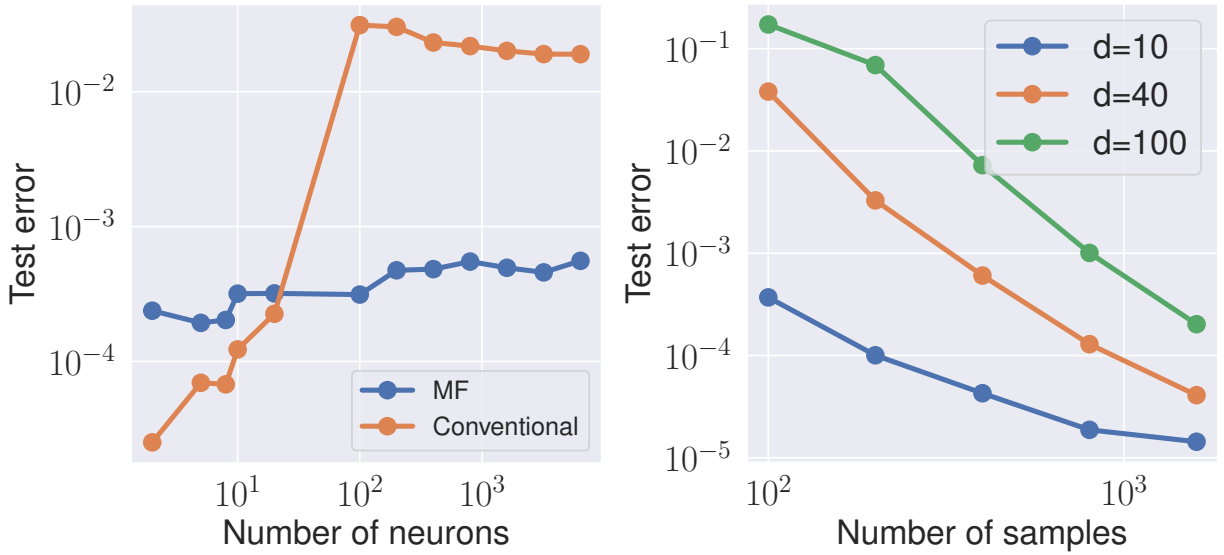
where

$$I(\boldsymbol{u}_1, \cdots, \boldsymbol{u}_m) = \mathcal{R}(f_m), \quad \boldsymbol{u}_j = (a_j, \boldsymbol{w}_j), \quad f_m(\boldsymbol{x}) = \frac{1}{m} \sum_j a_j \sigma(\boldsymbol{w}_j^T \boldsymbol{x})$$

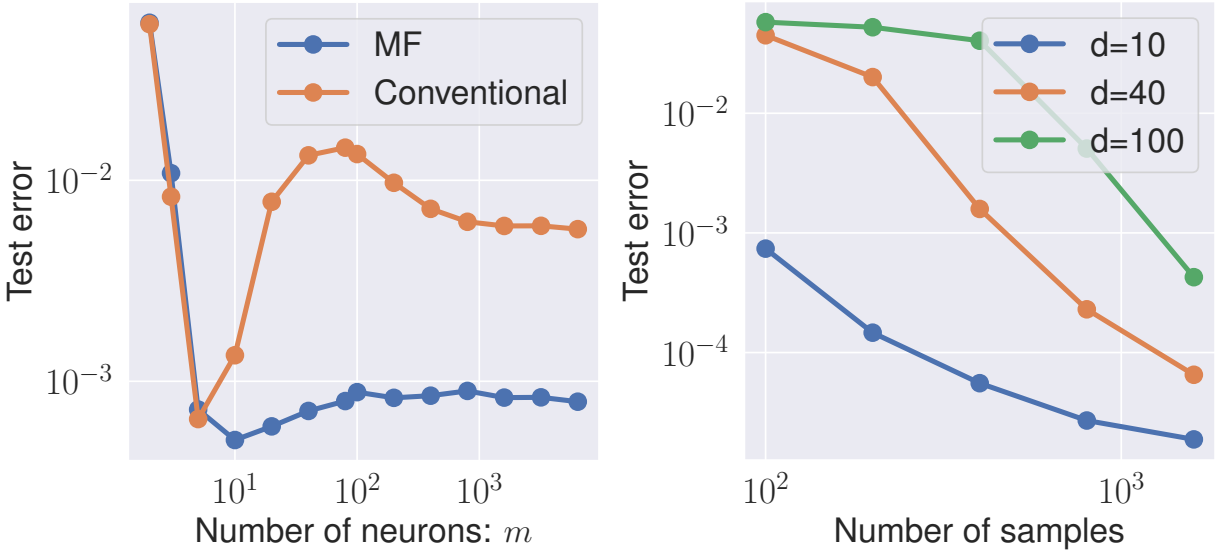
Note that this is exactly the GD dynamics for two-layer neural networks.

Why is "mean-field" or "continuous formulation" better?

Their performance is more robust.



(a) Single neuron f_1^*.



(b) Circle neuron f_2^*.

However, "mean-field" and "continuous" are different viewpoints:

- mean field: discrete $\rightarrow$ continuous by taking the limit (more like interacting particles in stat phys)
- continuous formulation: continuous $\rightarrow$ discrete by discretization (more like the usual numerical analysis situation)
  The crucial technical work is in the statics (particularly the representation of functions).

"Continuous" formulation tries to formulate the "first principles" of ML.
It allows us to think "outside the box" about ML.

- Indeed, there are other natural discretizations that lead to "non-neural network-like models" (see E, Ma and Wu (2019)).

Flow-based representation

ODE viewpoint (E (2017), Haber/Ruthotto (2017), "Neural ODEs" (Chen et al (2018)))

$$\frac{d\boldsymbol{z}}{d\tau} = \mathbf{g}(\tau, \boldsymbol{z}),\ \boldsymbol{z}(0) = \boldsymbol{x}$$

The flow-map at time 1:

$$\boldsymbol{x} \to \boldsymbol{z}(1)$$

Hypothesis space (or trial functions):

$$f = \alpha^T \boldsymbol{z}(1)$$

What form of $\mathbf{g}$ should we choose?

A natural choice of $\mathbf{g}$ (E, Ma and Wu, 2019):

$$\mathbf{g}(\tau, \boldsymbol{z}) = \mathbb{E}_{\boldsymbol{w}\sim\pi_\tau} \boldsymbol{a}(\boldsymbol{w}, \tau)\sigma(\boldsymbol{w}^T\boldsymbol{z}) = \mathbb{E}_{(\boldsymbol{a},\boldsymbol{w})\sim\rho_\tau} \boldsymbol{a}\sigma(\boldsymbol{w}^T\boldsymbol{z})$$

where $\{\pi_\tau\}$ or $\{\rho_\tau\}$ is a family of probability distributions.

$$\frac{d\boldsymbol{z}}{d\tau} = \mathbb{E}_{\boldsymbol{w}\sim\pi_\tau} \boldsymbol{a}(\boldsymbol{w}, \tau)\sigma(\boldsymbol{w}^T\boldsymbol{z}) = \mathbb{E}_{(\boldsymbol{a},\boldsymbol{w})\sim\rho_\tau} \boldsymbol{a}\sigma(\boldsymbol{w}^T\boldsymbol{z})$$

$$f(\boldsymbol{x}) = \alpha^T \boldsymbol{z}(\boldsymbol{x}, 1)$$

Discretize: We obtain the residual neural network model:

$$\boldsymbol{z}_{l+1} = \boldsymbol{z}_l + \frac{1}{LM}\sum_{j=1}^{M} \boldsymbol{a}_{j,l}\sigma(\boldsymbol{z}_l^T\boldsymbol{w}_{j,l}), l = 1, 2, \cdots, L-1, \quad \boldsymbol{z}_0 = V\tilde{\boldsymbol{x}}$$

$$f_L(\boldsymbol{x}) = \alpha^T \boldsymbol{z}_L$$

Particle discretization of the continuous GD flow recovers GD for ResNets.

What we don't know?

Other general representations of functions?

- It has to be some kind of expectation.
- Each such representation may lead to a new kind of ML model.

In a way, neural network models are also very natural: They arise naturally when considering continuous formulations in high dimensions.

Not covered:

- Other approximation results
- Estimates of the generalization gap
- Global minima selection for different optimization algorithms
- Adaptive optimization algorithms (Adam, RMSprop, ...)
- Landscapes, structure of the set of critical points
- .....

Summary: What do we know and what we don't know?

- approximation/generalization properties of hypothesis space
  - function spaces (RKHS, Barron, flow-induced, multi-layer)– the key is representation of functions
  - generalization error estimates of regularized model
  - CNNs, DenseNets, improvements?
- highly over-parametrized NNs
  - optimizes well
  - generalizes badly
- other regimes under the conventional scaling
  - qualitative behavior (better characterization of NN-like bahavior)?
  - behavior for different number of layers?
  - implicit regularization?
- mean-field training dynamics
  - Chizat and Bach and extensions
  - stronger results?
- continuous formulation
  - other representations of functions?
  - other discretizations?
- classification? GAN? RNN? RL?