A Mathematical Perspective of Machine Learning

Machine learning from the viewpoint of **numerical analysis in high dimensions**

Weinan E

Princeton University

Joint work with:

Chao Ma, Lei Wu

`www.math.princeton.edu/~weinan`

Outline

Outline

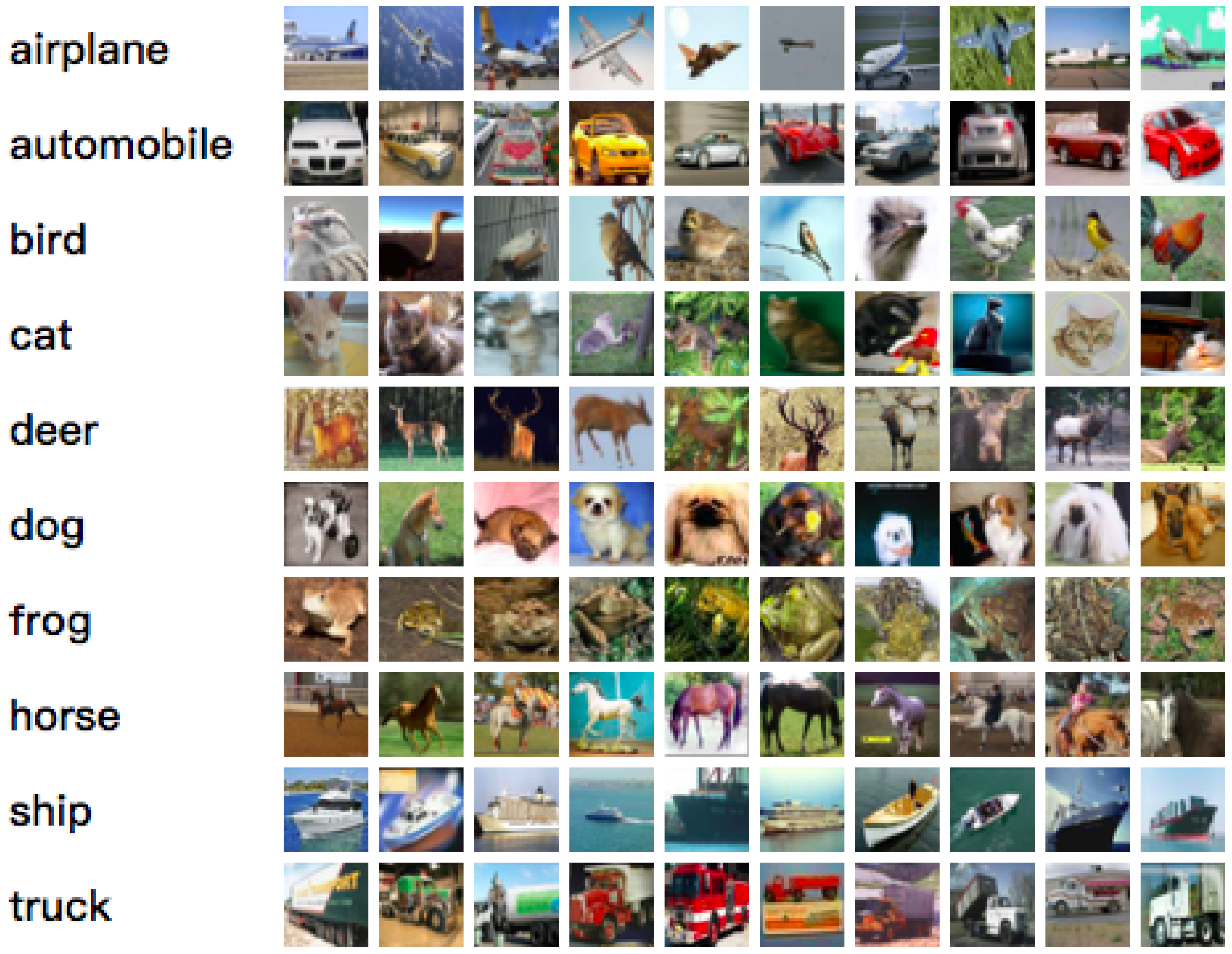
Basic example: Supervised learning

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

1. This is a problem about function approximation.
2. Based on finite pieces of "labeled" data
    - regression (f^* is continuous) vs classification (f^* is discrete)
    - will neglect measurement noise (not crucial for the talk)
    - assume $\boldsymbol{x}_j \in X = [0,1]^d$. d is typically quite large
    - notation: $\mu =$ the distribution of $\{\boldsymbol{x}_j\}$

In practice, one divides S into two subsets, a training set and a testing set.

Classification example: Cifar 10

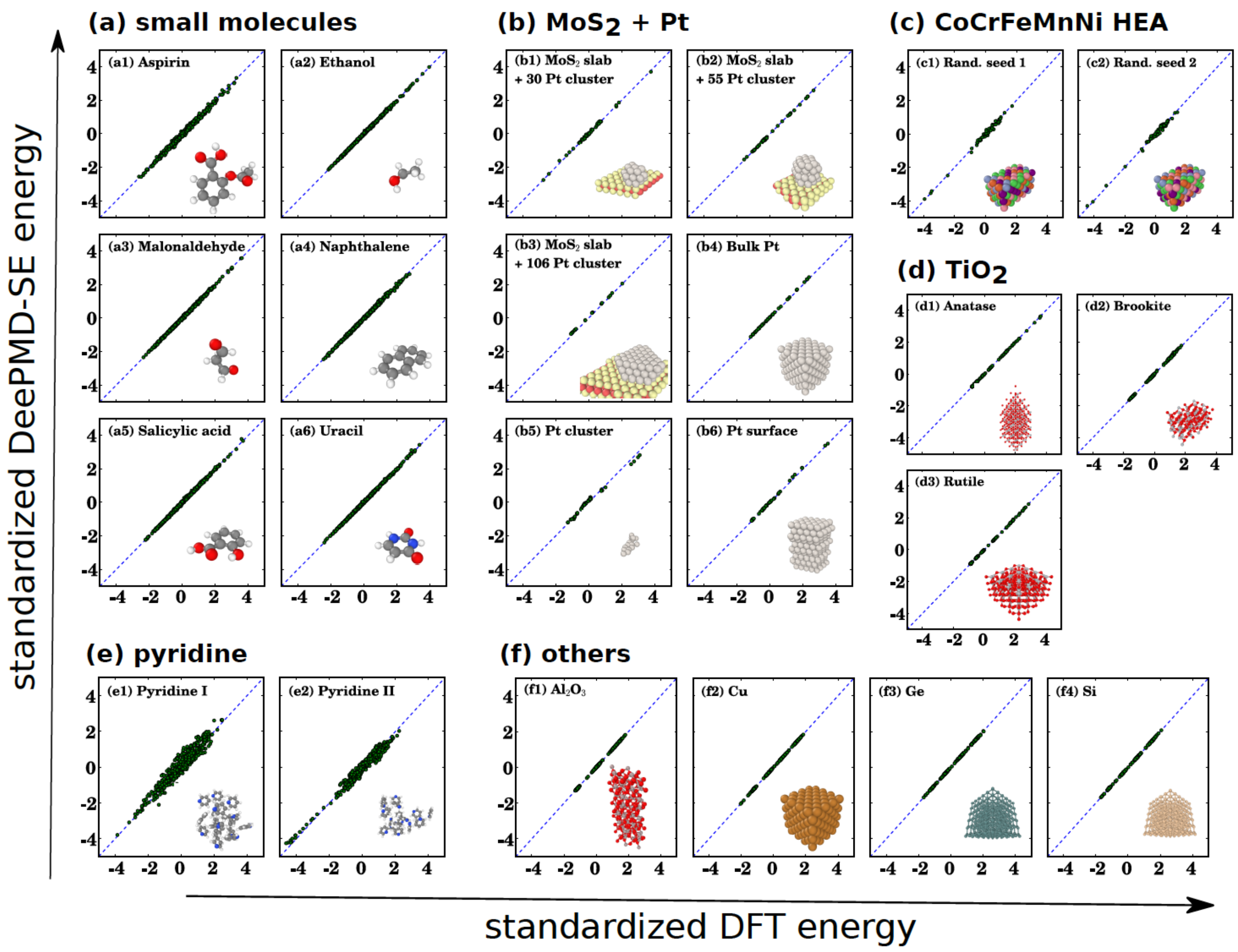


- Input: $\boldsymbol{x} \in [0,1]^d$ with $d = 32 \times 32 \times 3 = 3072$.
- Output: $f^* \in \{0, 1, \ldots, 9\}$.

Regression example: Inter-atomic potential

http://www.deepmd.org/database/deeppot-se-data/

$$V = V(\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_N)$$



ML in computational science and scientific computing

1. PDEs and control problems
   - Control problems in high dimension (Han and E, 2016)
   - High dimensional nonlinear parabolic PDEs (Han, Jentzen and E, 2017)
   - Least square methods for PDEs (Sirignano and Spiliopoulos, 2018)
2. Molecular dynamics
   - Neural network models (Behler and Parrinello, 2007)
   - Kernel methods (Bartayne, Kondor, Csányi, 2010)
   - Deep Potential Molecular Dynamics (DPMD, Zhang, Han, Wang, Car and E, 2017): molecular dynamics simulation with quantum accuracy for millions of atoms
3. Quantum many-body problem
   - Schrödinger equation for spins (Carleo and Troyer, 2016)
   - Schrödinger equation for electrons (Han, Zhang and E, 2018)
   - DeepMind group (Pfau, Spencer, Matthews and Foulkes, 2019)
4. Multi-scale modeling
   - Uniformly accurate moment closure models for kinetic equations (Han, Ma, Ma and E, 2019)
   - Hydrodynamic models for polymer fluids (Lei and E 2020)

Standard procedure for supervised learning

Focus on regression problem

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

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

   - regularization
3. choose an optimization algorithm and parameters
   - gradient descent (GD), stochastic gradient descent (SGD), ...
   - hyperparameters (initialization, step size=learning rate, ...)

Objective: Minimize the "population risk" (also known as the "generalization error")

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

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

Approximation and estimation errors

Want to understand $f^* - \hat{f}$, where $\hat{f}$ = the output of the ML model.

$f_m = \text{argmin}_{f \in \mathcal{H}_m} \mathcal{R}(f) =$ "best approx" to f^* in $\mathcal{H}_m$.

Decomposition of the error:

$$f^* - \hat{f} = f^* - f_m + f_m - \hat{f}$$

- $f^* - f_m$ is the approximation error, due entirely to the choice of the hypothesis
- $f_m - \hat{f}$ is the estimation error — additional error caused by the fact that we only have a finite dataset

Approximation error: Basics

How do we approximate functions?

- polynomials, piecewise polynomials, splines
- Fourier,wavelets
- other special basis functions (e.g. atomic orbitals)

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

Appearance of $1/d$ in the exponent of m: **Curse of dimensionality (CoD)**!

If we want $m^{-\alpha/d} = 0.1$, then $m = 10^{d/\alpha} = 10^d$, if $\alpha = 1$.

Dealing with high dimensionality: Monte Carlo integration

Monte Carlo: $X = [0,1]^d$, $\{\boldsymbol{x}_j, j \in [n]\}$ is uniformly distributed in X.

$$I(g) = \int_X g(\boldsymbol{x}) d\mu, \quad I_n(g) = \frac{1}{n} \sum_j g(\boldsymbol{x}_j)$$

$$\mathbb{E}(I(g) - I_n(g))^2 = \frac{\mathrm{var}(g)}{n}, \quad \mathrm{var}(g) = \int_X g^2(\boldsymbol{x}) d\boldsymbol{x} - \left(\int_X g(\boldsymbol{x}) d\boldsymbol{x} \right)^2$$

The $O(1/\sqrt{n})$ rate is (almost) the best we can hope for.

However, var(g) can be very large in high dimension. Variance reduction!

What do we learn from this for our function approximation problem?

Express functions as expectations!

Estimation error

Since we can only work with a finite dataset, what happens to our solution outside of the dataset?

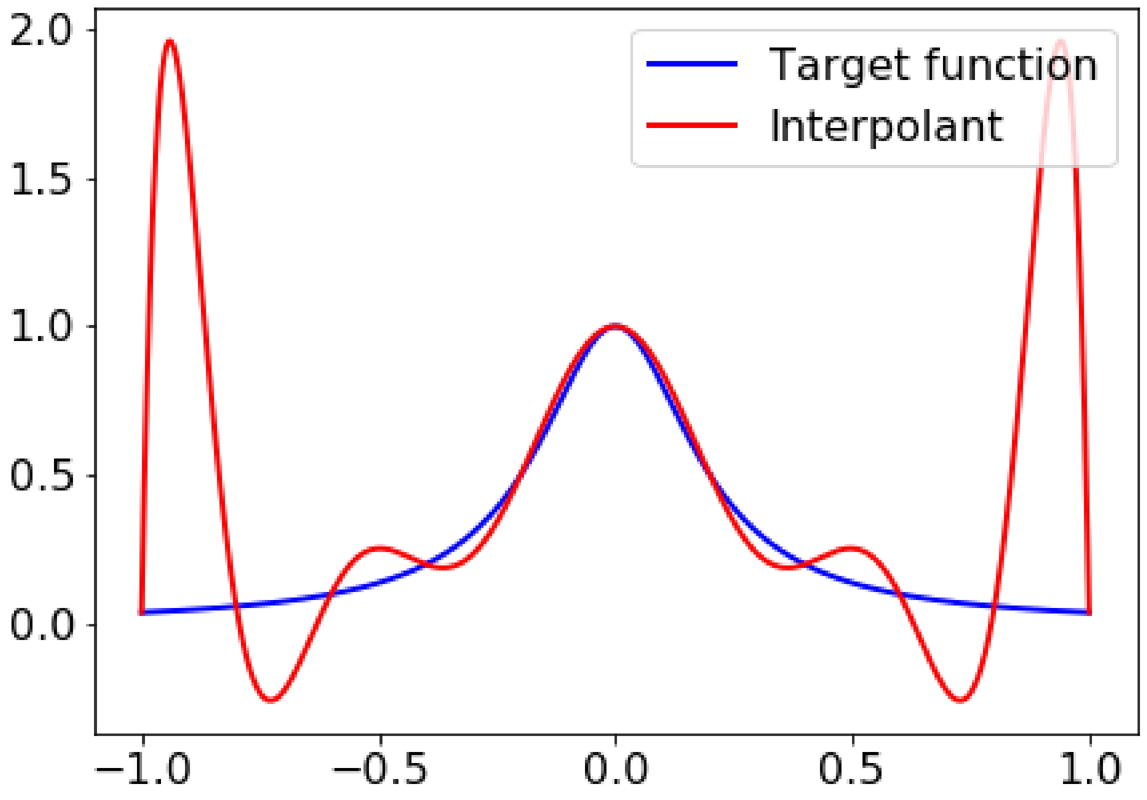


Figure: The Runge phenomenon

The generalization gap

$$\hat{f} = \text{argmin}_{f \in \mathcal{H}_m} \hat{\mathcal{R}}_n(f)$$

$$\text{"Generalization gap"} = |\mathcal{R}(\hat{f}) - \hat{\mathcal{R}}_n(\hat{f})| = |I(g) - I_n(g)|$$

$$\mathcal{R}(f) = \int_{\mathbb{R}^d} (f(\boldsymbol{x}) - f^*(\boldsymbol{x}))^2 d\mu, \quad \hat{\mathcal{R}}_n(f) = \frac{1}{n}\sum_j (f(\boldsymbol{x}_j) - f^*(\boldsymbol{x}_j))^2$$

$$I(g) = \int g(\boldsymbol{x}) d\mu, \quad I_n(g) = \frac{1}{n}\sum_j g(\boldsymbol{x}_j), \quad g(\boldsymbol{x}) = (\hat{f}(\boldsymbol{x}) - f^*(\boldsymbol{x}))^2$$

Difficulty: $\hat{f}$ is highly correlated with $\{\boldsymbol{x}_j\}$.

Should NOT expect: generalization gap $= O(1/\sqrt{n})$ to hold automatically.

Use the trivial bound:

$$|\mathcal{R}(\hat{f}) - \hat{\mathcal{R}}_n(\hat{f})| \le \sup_{f \in \mathcal{H}_m} |\mathcal{R}(f) - \hat{\mathcal{R}}_n(f)| = \sup_{f \in \mathcal{H}_m} |I(g) - I_n(g)|$$

The RHS depends heavily on the nature of $\mathcal{H}_m$.

- For Lipschitz space

$$\sup_{\|h\|_{Lip} \le 1} |I(h) - I_n(h)| \sim \frac{1}{n^{1/d}}$$

  This gives rise to **CoD for the size of the dataset**.

- For the Barron space, to be defined later

$$\sup_{\|h\|_{\mathcal{B}} \le 1} |I(h) - I_n(h)| \sim \frac{1}{\sqrt{n}}$$

"Donsker spaces"

Rademacher complexity

Let $\mathcal{H}$ be a set of functions, and $S = (\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n)$ be a set of data points. Then, the Rademacher complexity of $\mathcal{H}$ with respect to S is defined as

$$\mathrm{Rad}_S(\mathcal{H}) = \frac{1}{n}\mathbb{E}_\xi\left[\sup_{h\in\mathcal{H}}\sum_{i=1}^n \xi_i h(\boldsymbol{x}_i)\right],$$

where $\{\xi_i\}_{i=1}^n$ are i.i.d. random variables taking values ± 1 with equal probability.

Theorem (Rademacher complexity and the generalization gap)

Given a function class $\mathcal{H}$, for any $\delta \in (0,1)$, with probability at least $1-\delta$ over the random samples $S = (\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n)$,

$$\sup_{h\in\mathcal{H}}\left|\mathbb{E}_{\boldsymbol{x}}\left[h(\boldsymbol{x})\right] - \frac{1}{n}\sum_{i=1}^n h(\boldsymbol{x}_i)\right| \le 2\,\mathrm{Rad}_S(\mathcal{H}) + \sup_{h\in\mathcal{H}}\|h\|_\infty\sqrt{\frac{\log(2/\delta)}{2n}}.$$

$$\sup_{h\in\mathcal{H}}\left|\mathbb{E}_{\boldsymbol{x}}\left[h(\boldsymbol{x})\right] - \frac{1}{n}\sum_{i=1}^n h(\boldsymbol{x}_i)\right| \ge \frac{1}{2}\mathrm{Rad}_S(\mathcal{H}) - \sup_{h\in\mathcal{H}}\|h\|_\infty\sqrt{\frac{\log(2/\delta)}{2n}}.$$

Two types of machine learning models

(1). Models that suffer from CoD:

$$\text{generalization error} = O(m^{-\alpha/d}) \text{ and/or } O(n^{-\beta/d})$$

- piecewise polynomial approximation
- wavelets with fixed wavelet basis

(2). Models that don't suffer from CoD: For example

$$\text{generalization error} = O(\gamma_1(f^*)/m + \gamma_2(f^*)/\sqrt{n})$$

These are "Monte-Carlo-like" bounds, γ_1 and γ_2 play the role of variance in Monte Carlo.

- random feature models
- neural network models

Balance of approximation error (prefers large hypothesis space) and estimation error (prefers small hypothesis space).

Outline

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.

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

“New” approach:

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

where π is a probability measure on $\mathbb{R}^d$. Let $\{\boldsymbol{\omega}_j\}$ be an i.i.d. sample of π.

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

where

$$\text{var}(f) = \mathbb{E}_{\boldsymbol{\omega}\sim\pi}|a(\boldsymbol{\omega})|^2 - f(\boldsymbol{x})^2$$

Integral transform-based representation

Let σ be a nonlinear scalar function (the activation function), e.g. $\sigma(z) = \max(z, 0)$

$$f(\boldsymbol{x}) = \int_{\mathbb{R}^{d+1}} a(\boldsymbol{w})\sigma(\boldsymbol{w}^T\tilde{\boldsymbol{x}})\pi(d\boldsymbol{w}) = \mathbb{E}_{\boldsymbol{w}\sim\pi} a(\boldsymbol{w})\sigma(\boldsymbol{w}^T\tilde{\boldsymbol{x}})$$

$$f(\boldsymbol{x}) = \int_{\mathbb{R}^{d+2}} a\sigma(\boldsymbol{w}^T\tilde{\boldsymbol{x}})\rho(da, d\boldsymbol{w}) = \mathbb{E}_{(a,\boldsymbol{w})\sim\rho} a\sigma(\boldsymbol{w}^T\tilde{\boldsymbol{x}})$$

$\tilde{\boldsymbol{x}} = (\boldsymbol{x}, 1) = \boldsymbol{x}$ (standard abuse of notation in ML).

What kind of functions admit such a representation?

Consider the case when $\sigma(z) = \max(z, 0)$, the ReLU (rectified linear units) function.

Theorem (Barron and Klusowski (2016)): If $\int_{\mathbb{R}^d} \|\boldsymbol{\omega}\|_1^2 |\hat{f}(\boldsymbol{\omega})| d\omega < \infty$, where $\hat{f}$ is the Fourier transform of f, then f can be represented as

$$\tilde{f}(\boldsymbol{x}) = f(\boldsymbol{x}) - (f(0) + \boldsymbol{x} \cdot \nabla f(0)) = \int_\Omega a\sigma(\boldsymbol{w}^T \boldsymbol{x}) \rho(da, d\boldsymbol{w})$$

for $\boldsymbol{x} \in [0, 1]^d$. Furthermore, we have

$$\mathbb{E}_{(a,\boldsymbol{w})\sim\rho} |a| \|\boldsymbol{w}\|_1 \leq 2 \int_{\mathbb{R}^d} \|\boldsymbol{\omega}\|_1^2 |\hat{f}(\boldsymbol{\omega})| d\boldsymbol{\omega}$$

Generalizations

1. General integrand

$$f(\boldsymbol{x}) = \int_{\Omega} \phi(\boldsymbol{x}, \boldsymbol{u})\rho(d\boldsymbol{u}) = \mathbb{E}_{\boldsymbol{u}\sim\rho}\phi(\boldsymbol{x}, \boldsymbol{u})$$

Example: $\phi(\boldsymbol{x}, \boldsymbol{u}) = a\sigma(\boldsymbol{w}^T\boldsymbol{x}), \boldsymbol{u} = (a, \boldsymbol{w})$.

2. High co-dimensional case

$$f(\boldsymbol{x}) = \int_{\mathbb{R}^{d+1}\times\mathbb{R}^{d+1}} a(\boldsymbol{w}_1, \boldsymbol{w}_2)\sigma_1(\boldsymbol{w}_1^T\boldsymbol{x})\sigma_2(\boldsymbol{w}_2^T\boldsymbol{x})\pi(d\boldsymbol{w}_1, d\boldsymbol{w}_2)$$

$$= \mathbb{E}_{\boldsymbol{w}\sim\pi}a(\boldsymbol{w}_1, \boldsymbol{w}_2)\sigma_1(\boldsymbol{w}_1^T\boldsymbol{x})\sigma_2(\boldsymbol{w}_2^T\boldsymbol{x})$$

σ_1 and σ_2 are two nonlinear scalar functions.

3. Multi-layers

$$f(\boldsymbol{x}) = \int_{\mathbb{R}^{d_1}} a_1(\boldsymbol{w}_1)\sigma(\boldsymbol{w}_1^T\boldsymbol{z})\pi_1(d\boldsymbol{w}_1) = \mathbb{E}_{\omega_1\sim\pi_1}a_1(\boldsymbol{w}_1)\sigma(\boldsymbol{w}_1^T\boldsymbol{z}) = \mathbb{E}_{(a_1,\boldsymbol{w}_1)\sim\rho_1}a_1\sigma(\boldsymbol{w}_1^T\boldsymbol{z})$$

$$\boldsymbol{z} = \int_{\mathbb{R}^{d_2}} \boldsymbol{a}_2(\boldsymbol{w}_2)\sigma(\boldsymbol{w}_2^T\boldsymbol{x})\pi_2(d\boldsymbol{w}_2) = \mathbb{E}_{\boldsymbol{w}_2\sim\pi_2}\boldsymbol{a}_2(\boldsymbol{w}_2)\sigma(\boldsymbol{w}_2^T\boldsymbol{x}) = \mathbb{E}_{(\boldsymbol{a}_2,\boldsymbol{w}_2)\sim\rho_2}\boldsymbol{a}_2\sigma(\boldsymbol{w}_2^T\boldsymbol{x})$$

Approximation by discretization

Monte-Carlo:

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

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

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

This is a random feature model with features: $\{\sigma(\boldsymbol{w}_j^T\boldsymbol{x}), j = 1, \cdots, m\}$

Model parameters: $\theta = \{a_j\}$

Recovering the two-layer neural network model

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

$$\rho(da, d\boldsymbol{w}) \sim \frac{1}{m}\sum_j \delta_{(a_j, \boldsymbol{w}_j)}$$

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

This is a two-layer neural network model.

Model parameters $\theta = \{(a_j, \boldsymbol{w}_j)\}$.

Outline

Random feature model

$\{\phi(\cdot;\boldsymbol{w})\}$: collection of random features. 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

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

Consider functions of the form

$$\mathcal{F}_p = \{f : f(\boldsymbol{x}) = \int a(\boldsymbol{w})\phi(\cdot;\boldsymbol{w})\pi(\boldsymbol{w})\}, \|f\|_{\mathcal{F}_p} := (\mathbb{E}[|a(\boldsymbol{w})|^p])^{1/p} < \infty\}$$

$\mathcal{F}_2$ is the same as the reproducing kernel Hilbert space (RKHS) with kernel:

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

Direct and Inverse Approximation Theorem

Theorem (Direct Approximation Theorem)

Assume $\|f\|_{\mathcal{F}_\infty} < \infty$. *Then for any* $\delta \in (0,1)$, *with probability at least* $1-\delta$ *over the random sampling of* $\{\boldsymbol{w}_j\}_{j=1}^m$, *there exists* $\boldsymbol{a} \in \mathbb{R}^m$ *such that*

$$\mathcal{R}(\boldsymbol{a}) \le \frac{\|f\|_{\mathcal{F}_\infty}^2}{m}\left(1+\log(2/\delta)\right).$$

Moreover, $\sup_{j\in[m]} |a_j| \le \|f\|_{\mathcal{F}_\infty}$.

Theorem (Inverse Approximation Theorem)

Assume that ϕ *is continuous and bounded, and supp*(π) *is compact. Let* $(\boldsymbol{w}_j)_{j=1}^\infty$ *be a sequence of i.i.d. samples of* π. *Assume that there exist a sequence* $(a_j)_{j=0}^\infty$ *with* $\sup_j |a_j| \le C$, *such that*

$$\lim_{m\to\infty} \frac{1}{m}\sum_{j=1}^m a_j \sigma(\boldsymbol{w}_j \cdot \boldsymbol{x}) = f^*(\boldsymbol{x}),$$

for all $\boldsymbol{x} \in [0,1]^d$. *Then* $\|f^*\|_{\mathcal{F}_\infty} \le C$ *and there exists* $a^*(\cdot) : \Omega \mapsto \mathbb{R}$ *such that*

$$f^*(\boldsymbol{x}) = \int_\Omega a^*(\boldsymbol{w})\sigma(\boldsymbol{w}\cdot\boldsymbol{x})d\pi(\boldsymbol{w}), \quad a.s.$$

A priori estimates of the regularized model

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

Consider the regularized estimator

$$\hat{\boldsymbol{a}}_{n,\lambda} = \text{argmin } \mathcal{L}_{n,\lambda}(\boldsymbol{a})$$

Theorem

Assume that $\|f^*\|_\infty \leq 1$. There exist a constant C_0, such that for any $\delta > 0$, if $\lambda \geq C_0, m \geq \log^2(n/\delta)$, then with probability at least $1-\delta$ over the choice of training set, we have

$$\mathcal{R}(\hat{\boldsymbol{a}}_n) \lesssim \frac{\log(n/\delta)}{m}\|f^*\|_{\mathcal{F}_2}^2 + \frac{\|f^*\|_{\mathcal{F}_2}}{\sqrt{n}} + \sqrt{\frac{\log(n\|f^*\|_{\mathcal{F}_\infty}/\delta)}{n}} + \frac{\log^2(n/\delta)}{m^2}\|f^*\|_{\mathcal{F}_\infty}^2.$$

Two-layer neural network model: Barron spaces

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

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

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

$$\|f\|_{\mathcal{B}} = \inf_{\rho \in P_f} \sqrt{\mathbb{E}_\rho[a^2\|\boldsymbol{w}\|_1^2]},$$

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

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

Barron space and RKHS

Equivalent formulation (taking conditional expectation with respect to $\boldsymbol{w}$):

$$f^*(\boldsymbol{x}) = \int a(\boldsymbol{w})\sigma(\boldsymbol{w}^T\boldsymbol{x})\pi(d\boldsymbol{w}), \quad \boldsymbol{x} = (\boldsymbol{x}, 1)$$

Define:

$$k_\pi(\boldsymbol{x}, \boldsymbol{x}') = \mathbb{E}_{\boldsymbol{w}\sim\pi}\sigma(\boldsymbol{w}^T\boldsymbol{x})\sigma(\boldsymbol{w}^T\boldsymbol{x}')$$

We can write

$$\mathcal{B} = \bigcup_\pi \mathcal{H}_{k_\pi}$$

Shallow neural network can be understood as kernel method with adaptive (learned) kernel.

The ability to learn the right kernel is VERY important.
For example, SVM would be perfect if the right kernel was known.

Theorem (Direct Approximation Theorem)

There exists an absolute constant C_0 such that

$$\|f - f_m\|_{L^2(X)} \le \frac{C_0 \|f\|_{\mathcal{B}}}{\sqrt{m}}$$

Theorem (Inverse Approximation Theorem)

Let

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

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 $\|f^*\|_{\mathcal{B}} \le C$.

Complexity estimates

Theorem

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:

$$\|\theta\|_{\mathcal{P}} = \sqrt{\frac{1}{m}\sum_{k=1}^{m}|a_k|^2\|\boldsymbol{w}_k\|_1^2}.$$

Theorem

Assume that the target function $f^* : X \mapsto [0,1] \in \mathcal{B}$. There exist constants C_0, C_1, C_2, 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

$$\mathcal{R}(\hat{\theta}_n) \leq C_1\left(\frac{\|f^*\|_{\mathcal{B}}^2}{m} + \|f^*\|_{\mathcal{B}}\sqrt{\frac{\log(2d)}{n}}\right) + C_2\sqrt{\frac{\log(4C_2/\delta)+\log(n)}{n}}.$$

Typical results in ML literature: a posteriori estimates

$$\mathcal{R}(\hat{\theta}_n) - \hat{\mathcal{R}}_n(\hat{\theta}_n) \leq C_1\frac{\|\hat{\theta}\|}{\sqrt{n}}$$

Outline

The continuous formulation: Gradient flows

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

$$f(\boldsymbol{x}) = \int_{\mathbb{R}^{d+1}} a(\boldsymbol{w})\sigma(\boldsymbol{w}^T\boldsymbol{x})\pi(d\boldsymbol{w}) = \mathbb{E}_{\boldsymbol{w}\sim\pi}a(\boldsymbol{w})\sigma(\boldsymbol{w}^T\boldsymbol{x})$$

- $a =$ nonconserved, 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):

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

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

Gradient flow for the feature-based model

Fix π, optimize over a.

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

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

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

Decay estimates due to convexity: Let $f^*(\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{w} \sim \pi} a^*(\boldsymbol{w}) \sigma(\boldsymbol{w}^T \boldsymbol{x})$,

$$I(t) = \frac{1}{2} \|a(\cdot, t) - a^*(\cdot)\|^2 + t(\mathcal{R}(a(t)) - \mathcal{R}(a^*))$$

Then we have

$$\frac{dI}{dt} \leq 0, \quad \mathcal{R}(a(t)) \leq \frac{C_0}{t}$$

Conservative gradient flow

$$f(\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{u}\sim\rho}\phi(\boldsymbol{x}, \boldsymbol{u})$$

Example: $\boldsymbol{u} = (a, \boldsymbol{w}), \phi(\boldsymbol{x}, \boldsymbol{u}) = a\sigma(\boldsymbol{w}^T\boldsymbol{x})$

$$V(\boldsymbol{u}) = \frac{\delta\mathcal{R}}{\delta\rho}(\boldsymbol{u}) = \mathbb{E}_{\boldsymbol{x}}[(f(\boldsymbol{x}) - f^*(\boldsymbol{x}))\phi(\boldsymbol{x}, \boldsymbol{u})] = \int K(\boldsymbol{u}, \tilde{\boldsymbol{u}})\rho(d\tilde{\boldsymbol{u}}) - \tilde{f}(\boldsymbol{u})$$

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

- This is the mean-field equation derived by Chizat and Bach (2018), Mei, Montanari and Nguyen (2018), Rotskoff and Vanden-Eijnden (2018), Sirignano and Spiliopoulos (2018), by studying the continuum limit of two-layer neural networks.
- It is the gradient flow of $\mathcal{R}$ under the Wasserstein metric.
- $\mathcal{R}$ is convex but NOT displacement convex.

Convergence results

Chizat and Bach (2018, 2020): If $\{\rho(t)\}$ does converge as $t \to \infty$, then it must converge to a global minimum (and max margin solution for the classification problem if cross entropy is used as the loss function).

A simple one-D example (E, Ma and Wu(2019)):

$$f(x) = \int_0^{2\pi} \sigma(\cos(w-x))\rho(dw), \quad f^*(x) = \int_0^{2\pi} \sigma(\cos(w-x))\rho^*(dw)$$

$$\frac{\partial \rho}{\partial t} = \nabla \cdot (\rho \nabla K * (\rho - \rho^*)) .$$

$$K(w, w') = \frac{1}{2\pi} \int_0^{2\pi} \sigma(\cos(w-x))\sigma(\cos(w'-x))dx$$

Theorem

Assume that ρ^* is the uniform distribution and ρ_0 has C^1 density. Then

$$\lim_{t\to\infty} W_2(\rho(\cdot, t), \rho^*) = 0$$

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

We will see that gradient descent algorithm (GD) for random feature and neural network models are simply the particle method discretization of the gradient flows discussed before.

Particle method for the feature-based model

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

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

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.

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

The over-parametrized regime ($m \gg n$)

Consider the case when $m = \infty, n$ finite

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

$$K_n(\boldsymbol{w}, \tilde{\boldsymbol{w}}) = \frac{1}{n} \sum_j \sigma(\boldsymbol{w}^T \boldsymbol{x}_j) \sigma(\tilde{\boldsymbol{w}}^T \boldsymbol{x}_j), \quad \tilde{f}_n(\boldsymbol{w}) = \frac{1}{n} \sum_j y_j \sigma(\boldsymbol{w}^T \boldsymbol{x}_j)$$

Let $I_n(t) = \frac{1}{2} \|a_n(\cdot, t) - a_n^*(\cdot)\|_{L^2}^2 + t(\hat{\mathcal{R}}_n(a_n(t)) - \hat{\mathcal{R}}_n(a_n^*))$. Then we have

$$\frac{dI_n}{dt} \leq 0$$

$$\|a_n(\cdot, t)\|_{L^2} \leq C_0, \quad \hat{\mathcal{R}}_n(a_n(t)) \leq \frac{C_0}{t}$$

From these, we obtain the bounds for generalization error: with probability $1 - \delta$

$$\mathcal{R}(a_n(t)) \leq C_0(\|f^*\|_{\mathcal{H}_k}^2 + 1) \left(\frac{1 + \sqrt{\ln(1/\delta)}}{\sqrt{n}} + \frac{1}{t} \right)$$

The under-parametrized regime ($m \ll n$)

m finite, $n = \infty$. Hence $\hat{\mathcal{R}}_n = \mathcal{R}$.

Assume $f^*(\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{w}\sim\pi} a^*(\boldsymbol{w})\sigma(\boldsymbol{w}^T\boldsymbol{x})$.
$\{\boldsymbol{w}_j, j \in [m]\}$ = i.i.d. sample of π, $\quad a_m^* = (a^*(\boldsymbol{w}_1), a^*(\boldsymbol{w}_2), \cdots, a^*(\boldsymbol{w}_m))$.
Then Monte-Carlo estimates gives:

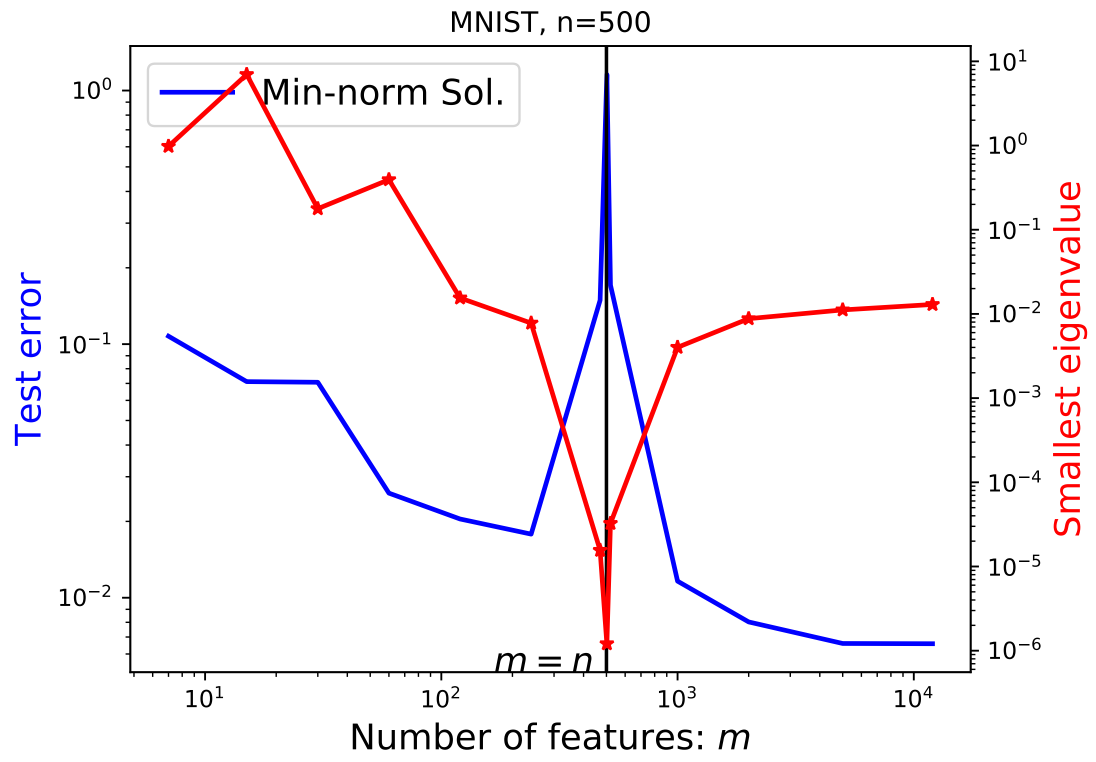
$$\mathcal{R}(a_m^*) \leq C_0 \frac{\|f^*\|_{\mathcal{H}_k}^2}{m}$$

Convexity estimates gives:

$$\mathcal{R}(a(t)) \leq \mathcal{R}(a_m^*) + C_0 \frac{\|f^*\|_{\mathcal{H}_k}^2}{t} \leq C_0 \|f^*\|_{\mathcal{H}_k}^2 \left(\frac{1}{m} + \frac{1}{t}\right)$$

The resonant regime ($m \sim n$)

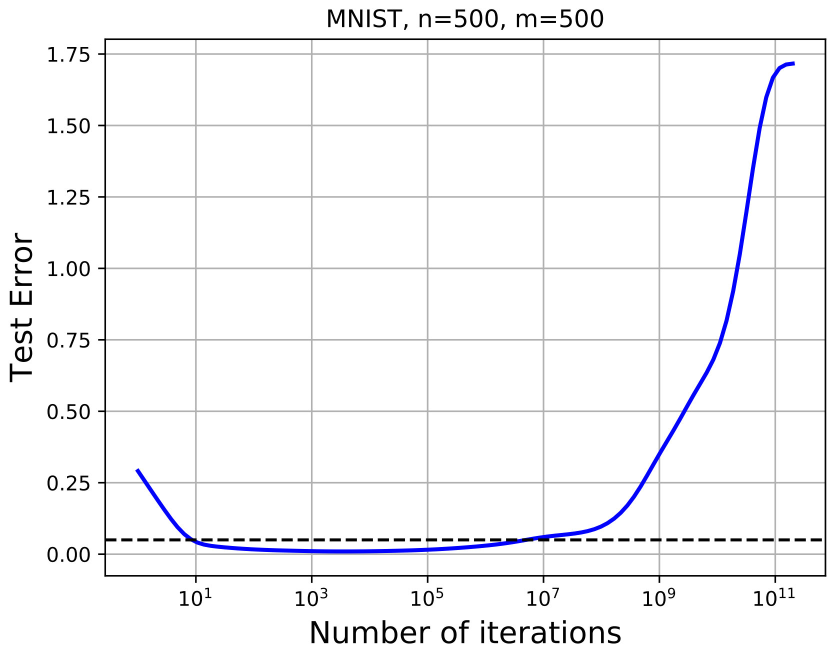
The "double descent" phenomenon (Belkin et al)



The large test error is caused by small eigenvalues of the Gram matrices.

The slow deterioration

The test accuracy deteriorates slowly along the GD trajectory:



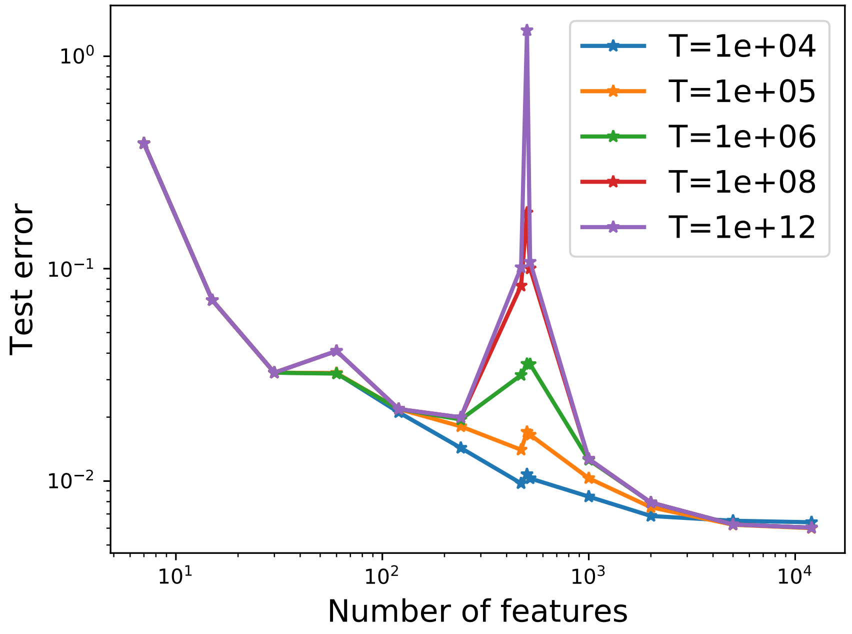


Figure: **Left:** Test error along the GD path for $m = n = 500$. The black dashed line corresponds to test error 0.05. **Right:** The test error of GD solutions obtained by running different number of iterations.

Because the large test error is caused by very small eigenvalues of the Gram matrix, which also lead to slow convergence.

Analysis

1. Let $\mathcal{K}$ be the kernel operator defined as

$$\mathcal{K}f(\boldsymbol{x}) = \int K(\boldsymbol{x}, \boldsymbol{x}')f(\boldsymbol{x}')\pi(d\boldsymbol{x}'),$$

   and $(\mu_i, \psi_i(\boldsymbol{x}))$ be the eigenvalues and eigenfunctions of $\mathcal{K}$.
2. To be simple, let $f^*(\boldsymbol{x}) = \psi_1(\boldsymbol{x})$.
3. Let $\Phi = \sigma(X^T B)$, with $X = [\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n]$ and $B = [\boldsymbol{b}_1, \boldsymbol{b}_2, ..., \boldsymbol{b}_m]$. Let the SVD decomposition of Φ be $\Phi = U\Sigma V^T$. Assume $m = n$ and $\Phi\Phi^T \succ 0$. Let $U = [\boldsymbol{u}_1, ..., \boldsymbol{u}_n]$, $V = [\boldsymbol{v}_1, ..., \boldsymbol{v}_m]$, and $\lambda_i = \Sigma_{ii}$, $i = 1, 2, ..., n$.
4. The prediction function at time t is

$$\hat{f}_t(\boldsymbol{x}) = \sum_{i=1}^{n} \frac{1 - e^{-\frac{\lambda_i^2 t}{n^2}}}{\lambda_i} (\boldsymbol{u}_i^T \boldsymbol{y})(\boldsymbol{v}_i^T \sigma(B^T \boldsymbol{x})).$$

Analysis

The following theorem estimates the test error $\|\hat{f}_t - f^*\|$.

Theorem

For any $\delta > 0$, there exists a constant $C(\delta)$, such that with probability no less than $1-\delta$ over the choice of X and B, we have

$$\|\hat{f}_t - f^*\| \leq e^{-\frac{\lambda_1^2 t}{n^2}} + C(\delta)n^{-\frac{1}{4}} + C(\delta)Mn^{\frac{3}{4}}d(t),$$

where $M = \int \|\sigma(B^T\boldsymbol{x})\|^2\pi(d\boldsymbol{x})$, and

$$d(t) = \min\left\{\sqrt{\frac{t}{n^2}}, \frac{\lambda_{\lfloor\sqrt{n}\rfloor}t}{n^2}, \frac{1}{\lambda_n}\right\}.$$

Runge phenomenon

Under the gradient flow, Runge phenomenon only shows up after very long time

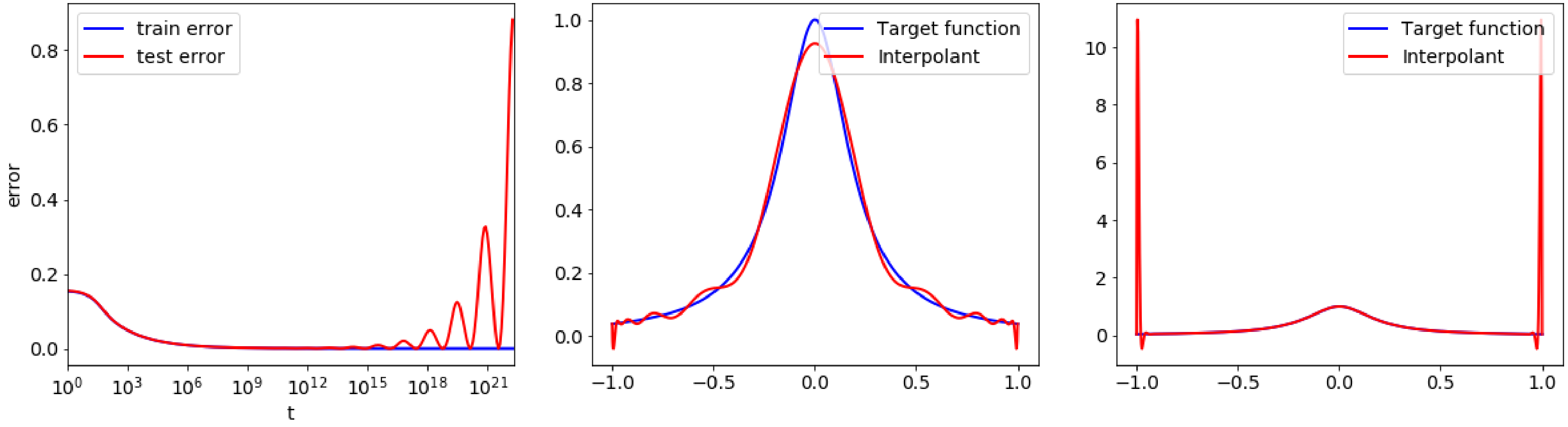


Figure: The function $\frac{1}{1+25x^2}$ is approximated by high-order polynomials on 101 equi-spaced points on $[-1,1]$. Gradient flow is used to minimize the empirical l^2 loss. **Left:** Interpolant by GD at $t = 2 \times 10^{10}$; **Middle:** Interpolant by GD at $t = 2 \times 10^{20}$; **Right:** Training and testing error along the GD trajectory.

Discretization of the conservative flow

$$\partial_t \rho = \nabla(\rho \nabla V) \tag{1}$$

$$\rho(da, d\boldsymbol{w}) \sim \frac{1}{m} \sum_j \delta_{(a_j, \boldsymbol{w}_j)}$$

$$I(\boldsymbol{u}_1, \cdots, \boldsymbol{u}_m) = \mathcal{R}(f_m), \quad \boldsymbol{u}_j = (a_j, \boldsymbol{w}_j), j =\in [m]$$

where $f_m(\boldsymbol{x}) = \frac{1}{m} \sum_j a_j \sigma(\boldsymbol{w}_j^T \boldsymbol{x})$.
Lemma: Given a set of initial data $\{\boldsymbol{u}_j^0 = (a_j^0, \boldsymbol{w}_j^0), j \in [m]\}$. The solution of (**??**) with initial data $\rho(0) = \frac{1}{m} \sum_{j=1}^m \delta_{\boldsymbol{u}_j^0}$ is given by

$$\rho(t) = \frac{1}{m} \sum_{j=1}^m \delta_{\boldsymbol{u}_j(t)}$$

where $\{\boldsymbol{u}_j(\cdot), j =\in [m]\}$ solves the following systems of ODEs:

$$\frac{d\boldsymbol{u}_j(t)}{dt} = -\nabla_{\boldsymbol{u}_j} I(\boldsymbol{u}_1, \cdots, \boldsymbol{u}_m), \quad \boldsymbol{u}_j(0) = \boldsymbol{u}_j^0, \quad j \in [m] \tag{2}$$

Note that (**??**) is exactly the GD dynamics for two-layer neural networks.

Outline

Flow-based representation

Dynamical system viewpoint (E (2017), Haber and Ruthotto (2017) ...)

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

The flow-map at time 1: $\boldsymbol{x} \to \boldsymbol{z}(1)$.

Trial functions:

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

The correct form of g (E, Ma and Wu, 2019):

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

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

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

As before, we can also use the model:

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

$$f(\boldsymbol{x}) = \mathbf{1}^T \boldsymbol{z}_1^{\boldsymbol{x}}$$

Discretize: We obtain the residual neural network model:

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

$$f_L(\boldsymbol{x}) = \mathbf{1}^T \boldsymbol{z}_L$$

Compositional law of large numbers

Consider the following compositional scheme:

$$\begin{aligned} \boldsymbol{z}_{0,L}(\boldsymbol{x}) &= \boldsymbol{x}, \\ \boldsymbol{z}_{l+1,L}(\boldsymbol{x}) &= z_{l,L}(\boldsymbol{x}) + \frac{1}{LM}\sum_{k=1}^{M} \boldsymbol{a}_{l,k}\sigma(\boldsymbol{w}_{l,k}^T z_{l,L}(\boldsymbol{x})), \end{aligned}$$

$(\boldsymbol{a}_{l,k}, \boldsymbol{w}_{l,k})$ are pairs of vectors i.i.d. sampled from a distribution ρ.

Theorem

Assume that

$$\mathbb{E}_\rho \| |\boldsymbol{a}||\boldsymbol{w}^T| \|_F^2 < \infty$$

where for a matrix or vector $\boldsymbol{A}$, $|\boldsymbol{A}|$ means taking element-wise absolute value for $\boldsymbol{A}$. Define $\boldsymbol{z}$ by

$$\begin{aligned} \boldsymbol{z}(\boldsymbol{x}, 0) &= \boldsymbol{V}\boldsymbol{x}, \\ \frac{d}{d\tau}\boldsymbol{z}(\boldsymbol{x}, t) &= \mathbb{E}_{(\boldsymbol{a},\boldsymbol{w})\sim\rho}\boldsymbol{a}\sigma(\boldsymbol{w}^T\boldsymbol{z}(\boldsymbol{x}, \tau)). \end{aligned}$$

Then we have

$$\boldsymbol{z}_{L,L}(\boldsymbol{x}) \to \boldsymbol{z}(\boldsymbol{x}, 1)$$

almost surely as $L \to +\infty$.

Compositional function spaces

Extension: Let $\{\rho_\tau\}$ be a family of prob distributions (for $(\boldsymbol{a}, \boldsymbol{w})$) such that $\mathbb{E}_{\rho_\tau} g(\boldsymbol{a}, \boldsymbol{w})$ is integrable as a function of τ for any continuous function g. Define:

$$\begin{aligned}
\boldsymbol{z}(\boldsymbol{x}, 0) &= \boldsymbol{V}\boldsymbol{x}, \\
\frac{d}{d\tau}\boldsymbol{z}(\boldsymbol{x}, \tau) &= \mathbb{E}_{(\boldsymbol{a},\boldsymbol{w})\sim\rho_\tau} \boldsymbol{a}\sigma(\boldsymbol{w}^T \boldsymbol{z}(\boldsymbol{x}, \tau))
\end{aligned}$$

Let $f_{\alpha,\{\rho_\tau\},\boldsymbol{V}}(\boldsymbol{x}) = \alpha^T \boldsymbol{z}(\boldsymbol{x}, 1)$. Define "compositional norm":

$$\begin{aligned}
\frac{d}{d\tau}\mathbf{N}(t) &= \mathbb{E}_{\rho_\tau} |\boldsymbol{a}||\boldsymbol{w}|^T \mathbf{N}(\tau), \\
\mathbf{N}(0) &= \mathbf{I} \\
\|f\|_{\mathcal{D}_1} &= \inf_{f = f_{\alpha,\{\rho_\tau\},\boldsymbol{V}}} \|\alpha\|_1 \, \|\mathbf{N}(1)\|_{1,1} \, \|\boldsymbol{V}\|_{1,1},
\end{aligned}$$

$\|\cdot\|_{\mathcal{D}_2}$ is defined similarly.

Barron space and the compositional function space

Theorem

$\mathcal{B} \subset \mathcal{D}_2$. There exists constant $C > 0$, such that

$$\|f\|_{\mathcal{D}_2} \le \sqrt{d+1}\|f\|_{\mathcal{B}}$$

holds for any $f \in \mathcal{B}$,

Theorem (Direct approximation theorem)

Let $f \in L^2(X) \cap \mathcal{D}_2$. There exists a residue-type neural network $f_L(\cdot;\tilde{\theta})$ of input dimension $d+1$ and depth L such that $\|f_L\|_P \lesssim \|f\|_{c_1}^3$ and

$$\int_X |f(\boldsymbol{x}) - f_L((\boldsymbol{x});\tilde{\theta})|^2 dx \to 0 \lesssim \frac{\|f\|_{c_2}^2}{L}$$

Furthermore, if $f = f_{\alpha,\{\rho_\tau\},\boldsymbol{V}}$ and ρ_τ is Lipschitz continuous in τ, then

$$\int_X |f(\boldsymbol{x}) - f_L((\boldsymbol{x});\tilde{\theta})|^2 dx \lesssim \frac{\|f\|_{\mathcal{D}_2}^2}{L}$$

Theorem (Inverse approximation theorem)

Let $f \in L^2(X)$. Assume that there is a sequence of residual networks $\{f_L(\boldsymbol{x})\}_{L=1}^{\infty}$ with increasing depth such that $\|f(\boldsymbol{x}) - f_L(\boldsymbol{x})\| \to 0$ as $L \to \infty$. Assume further that the parameters are (entry-wise) bounded, then there exists α, $\{\rho_\tau\}$ and $\boldsymbol{V}$ such that

$$f(\boldsymbol{x}) = f_{\alpha,\{\rho_\tau\},\boldsymbol{V}}(\boldsymbol{x}).$$

Complexity control

Rademacher complexity bound for path norm

Let $\mathcal{F}_{L,Q} = \{f_L : \|f_L\|_{\mathcal{D}_1} \leq Q\}$. Assume $\boldsymbol{x}_i \in [0,1]^d$. Then, for any data set $S = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$, we have

$$\mathrm{Rad}_S(\mathcal{F}_{L,Q}) \leq 3Q\sqrt{\frac{2\log(2d)}{n}}.$$

Regularized model and a priori estimates

Regularized loss function:

$$\mathcal{L}_{n,\lambda}(\theta) = \hat{\mathcal{R}}_n(\theta) + \lambda(\|\theta\|_{\mathcal{D}_1} + 1)\sqrt{\frac{2\log(2d)}{n}}.$$

Theorem (a-priori estimates)

Assume that $f^\star : [0,1]^d \to [-1,1]$ *such that* $f^* \in \mathcal{D}_2$. *Let*

$$\hat{\theta} = \textit{argmin}\ _\theta \mathcal{L}_{n,\lambda}(\theta)$$

then if λ *is larger than some constant, and the depth* L *is sufficiently large, for any* $\delta > 0$, *with probability at least* $1-\delta$,

$$\mathcal{R}(\hat{\theta}) \lesssim \frac{\|f^*\|_{\mathcal{D}_2}^2}{L} + \lambda(\|f^*\|_{\mathcal{D}_1}^3 + 1)\sqrt{\frac{\log(2d)}{n}} + \sqrt{\frac{\log(1/\delta)}{n}}.$$

Gradient flow for flow-based models

- Consider a generalized flow-based model

$$\begin{aligned} \boldsymbol{z}_0^{\boldsymbol{x}} &= V\boldsymbol{x} \\ \frac{d\boldsymbol{z}_\tau^{\boldsymbol{x}}}{d\tau} &= \mathbb{E}_{\boldsymbol{w}\sim\rho_\tau}[\boldsymbol{\phi}(\boldsymbol{z}_\tau^{\boldsymbol{x}}, \boldsymbol{w})] \\ f(\boldsymbol{x};\rho) &= \mathbf{1}^T\boldsymbol{z}_1^{\boldsymbol{x}}. \end{aligned}$$

- Denote by $W := \{\rho : [0,1] \mapsto \mathcal{P}_2(\Omega)\}$ the space of all feasible parameters. For any $\rho^1, \rho^2 \in W$, define the following metric:

$$d(\rho^1, \rho^2) := \sqrt{\int_0^1 W_2^2(\rho_\tau^1, \rho_\tau^2)d\tau}.$$

- Consider minimizing the following risk

$$\mathcal{R}(f) = \mathbb{E}_{\boldsymbol{x}}(f(\boldsymbol{x}) - f^*(\boldsymbol{x}))^2].$$

Define the Hamiltonian $H : \mathbb{R}^d \times \mathbb{R}^d \times \mathcal{P}_2(\Omega) :\mapsto \mathbb{R}$ as

$$H(\boldsymbol{z}, \boldsymbol{p}, \mu) = \mathbb{E}_{\boldsymbol{w}\sim\mu}[\boldsymbol{p}^T \boldsymbol{\phi}(\boldsymbol{x}, \boldsymbol{w})].$$

Theorem

The gradient flow in the metric space (W, d) is given by

$$\partial_t \rho_\tau(\boldsymbol{w}, t) = \nabla \cdot (\rho_\tau(\boldsymbol{w}, t) \nabla V(\boldsymbol{w}; \rho)), \quad \forall \tau \in [0, 1],$$

where

$$V(\boldsymbol{w}; \rho) = \mathbb{E}_{\boldsymbol{x}}[\frac{\delta H}{\delta \mu}\left(\boldsymbol{z}_\tau^{t,\boldsymbol{x}}, \boldsymbol{p}_\tau^{t,\boldsymbol{x}}, \rho_\tau(\cdot; t)\right)],$$

and for each $\boldsymbol{x}$, $(\boldsymbol{z}_\tau^{t,\boldsymbol{x}}, \boldsymbol{p}_\tau^{t,\boldsymbol{x}})$ are defined by the forward and backward equations, respectively:

$$\frac{d\boldsymbol{z}_\tau^{t,\boldsymbol{x}}}{d\tau} = \nabla_{\boldsymbol{p}} H = \mathbb{E}_{\boldsymbol{w}\sim\rho_\tau(\cdot;t)}[\boldsymbol{\phi}(\boldsymbol{z}_\tau^{t,\boldsymbol{x}}, \boldsymbol{w})]$$
$$\frac{d\boldsymbol{p}_\tau^{t,\boldsymbol{x}}}{d\tau} = -\nabla_{\boldsymbol{z}} H = \mathbb{E}_{\boldsymbol{w}\sim\rho_\tau(\cdot;t)}[\nabla_{\boldsymbol{z}}^T \boldsymbol{\phi}(\boldsymbol{z}_\tau^{t,\boldsymbol{x}}, \boldsymbol{w}) \boldsymbol{p}_\tau^{t,\boldsymbol{x}}].$$

with the boundary conditions:

$$\boldsymbol{z}_0^{t,\boldsymbol{x}} = V\tilde{\boldsymbol{x}}$$
$$\boldsymbol{p}_1^{t,\boldsymbol{x}} = \mathbf{1}2(f(\boldsymbol{x}; \rho(\cdot; t)) - f^*(\boldsymbol{x})).$$

Discretization

- forward Euler for the flow in τ variable, step size $1/L$.
- particle method for the GD dynamics, M samples in each layer

$$\boldsymbol{z}_{l+1}^{t,\boldsymbol{x}} = \boldsymbol{z}_{l}^{t,\boldsymbol{x}} + \frac{1}{LM}\sum_{j=1}^{M}\boldsymbol{\phi}(\boldsymbol{z}_{l}^{t,\boldsymbol{x}}, \boldsymbol{w}_{l}^{j}(t)), \quad l = 0, \dots, L-1$$

$$\boldsymbol{p}_{l}^{t,\boldsymbol{x}} = \boldsymbol{p}_{l+1}^{t,\boldsymbol{x}} + \frac{1}{LM}\sum_{j=1}^{M}\nabla_{\boldsymbol{z}}\boldsymbol{\phi}(\boldsymbol{z}_{l+1}^{t,\boldsymbol{x}}, \boldsymbol{w}_{l+1}^{j}(t))\boldsymbol{p}_{l+1}^{t,\boldsymbol{x}}, \quad l = 0, \dots, L-1$$

$$\frac{d\boldsymbol{w}_{l}^{j}(t)}{dt} = -\mathbb{E}_{\boldsymbol{x}}[\nabla_{\boldsymbol{w}}^{T}\boldsymbol{\phi}(\boldsymbol{z}_{l}^{t,\boldsymbol{x}}, \boldsymbol{w}_{l}^{j}(t))\boldsymbol{p}_{l}^{t,\boldsymbol{x}}].$$

This recovers the gradient descent algorithm (with back-propagation) for the ResNet:

$$\boldsymbol{z}_{l+1} = \boldsymbol{z}_{l} + \frac{1}{LM}\sum_{j=1}^{M}\boldsymbol{\phi}(\boldsymbol{z}_{l}, \boldsymbol{w}_{l}).$$

Outline

Numerical analysis viewpoint of ML

Basic strategy for formulating models/algorithms:

1. Start with continuous formulation (representation of functions, loss function, gradient flows for optimization)
2. Discretization: Most important existing ML models can be recovered as discretizations of the continuous formulation

Instead of specific neural networks, think about representations of functions.
Besides integral transforms and flows, are there other ways of representing functions?

Basic strategy for the analysis of models/algorithms:

1. At the level of the hypothesis space:
   - Function norms, direct and inverse approximation theorems
   - Complexity estimates (Rademacher)
   - A priori and a posteriori error estimates

   Goal: get Monte-Carlo like error estimates
2. At the level of the optimization algorithm:
   - The three regimes (over- and under-parametrized, resonant)
   - Resonance might be bad news (large generalization gap)
   - Slow deterioration phenomenon comes to the rescue
   - Stability of the back-propagation algorithm

Overall picture that has emerged: Why thing work?

1. The target functions can be represented as expectations in various forms, and this is the fundamental reason why things can work in such high dimension.
2. The risk functionals are nice functionals. Even if not convex, they share many features of convex functionals (not always true).
3. The different gradient flows are nice flows, and dynamics helps to allievate overfitting, basically because they are integral equations.
4. Known category-2 ML models are simply particle method discretizations of the corresponding continuous problems.
5. They are naturally numerically stable.

These should be our "design principles" for ML models/algorithms.

Overall picture that has emerged: Why thing don't work?

1. Fully connected neural networks are "unstable": gradient grows exponentially as the number of layers increases (Lyapunov exponent, see the work of Hanin)
2. Resonance: When $m \sim n$, the minimum-norm solution of the random feature model does not generalize due to the presence of small eigenvalues in the Gram matrix. In other regimes, this phenomenon may also manifest at some level.
3. GAN really get stuck at local min
4. RNNs can also be unstable

Violations of the design principles.

Open questions

1. Asymptotic (large time) convergence of the gradient flows (PDE-like problem)
2. Resonance for neural network models
3. Classification problem (need to define the relevant space of probability measures in high dimensions)
4. Density estimation in high dimension, similar issue + regularization

New function norms and spaces

From a numerical analysis viewpoint, function spaces (such as Besov) are classes of functions that have a particular approximation property (specific order of convergence for a specific approximation scheme).

RKHS, Barron space and compositional function spaces are the high dimensional analog, each is associated with a particular approximation scheme.

Solutions of PDEs belong to these spaces? Size of their norms?