Mathematics in Neural Network **Summer 2019**

Lecture 1: The Mathematical Theory of NN-Based Machine Learning

Lecturer: E Weinan *Scribe: Yihang Chen, Yuyue Wang*

Disclaimer: *These notes have not been subjected to the usual scrutiny reserved for formal publications. They may be distributed outside this class only with the permission of the Instructor.*

1 Basic concepts

1.1 Supervised learning

Object of interest: (f^*, μ),where$f^* : \mathscr{R}^d \to \mathscr{R}^1, \mu$ is a prob measure on $\mathscr{R}^d$.
Given a set of samples from μ,$\{x_j\}_{j=1}^n$,and $\{y_j = f^*(x_j)\}_{j=1}^n$
Task: Approximate f^* using $S = \{(x_j, y_j)\}_{j=1}^n$.
Strategy: Construct some "hypothesis space"(space of function)$\mathscr{H}_m$(m the dimension of $\mathscr{H}_m$).Minimize the "empirical risk":

$$\hat{\mathscr{R}}_n(\theta) = \frac{1}{n}\sum_j (f(x_j) - y_j)^2 = \frac{1}{n}\sum_j (f(x_j) - f^*(x_j))^2$$

What we really want to minimize is the "population risk":

$$\mathscr{R}(\theta) = E(f(x) - f^*(x))^2 = \int_{\mathscr{R}^d} (f(x) - f^*(x))^2 d\mu$$

1.2 Issues that we want to understand

1. some property observed in practice:

   (a) high dimensionality

   (b) models are highly over-parametrized,classical machine learning theory would suggest overfitting

   (c) models are non-convex,yet simple gradient algorithms seem to work(compared with structural optimization in material science and chemistry)

2. Advanced questions:

   (a) Why deep networks seem to perform better than shallow ones?

   (b) Why stochastic gradient descent seems to perform better than gradient discent?
   Lots of other issues,mysterious,e.g. batch normalization,dropout,initialization,...

1.3 Approximating functions in high dimensions

Given (f^*, μ),where $f^* : \mathscr{R}^d \to \mathscr{R}^1$,$\mu$ is a prob measure on $\mathscr{R}^d$,the target is to minimize the "population risk" over "hypothesis space":

$$L(\theta) = \mathscr{E}(f(x) - f^*(x))^2 = \int_{\mathscr{R}^d} (f(x) - f^*(x))^2 d\mu$$

some options to choose the hypothesis space are linear regression,generalized linear models,two-layer neural networks,deep neural networks

1.4 Why neural networks

we can see the reason of using neural network by compare the difference between linear and nonlinear approximations:
for linear: $f(x) \approx f_m(x) = \sum^m a_k cos(2\pi kx), x \in [0,1]^d$,$inf||f - f_m||_2 \geq C(f)m^{-1/d}$
so the number of parameters needed goes up exponentially fast as a function of the accuracy requirement.
while nonlinear:$f(x) \approx f_m(x) = \sum^m a_k cos(2\pi b_k x), x \in [0,1]^d$(two-layer neural network with cos(x) as the activation function),$inf||f - f_m||_2 \leq C(f)m^{-1/2}$.So neural network avoid the curse of dimensionality

1.5 Exploding and vanishing gradients

although neural network avoid the curse of dimensionality, as the depth of network increases, the gradient gets too large or too small, as the formula implies:

$$f(x,\theta) = W_L \sigma(W_{L-1} \sigma(\sigma(W_0 x))), \theta = (W_0, W_1, , W_L)$$

$$\nabla_\theta f = W_L W_{L-1} W_0 \ \kappa^L, L >> 1$$

use residual network can solve the problem.

1.6 Some methods to study deep learning

we can construct nonlinear approximations through the flow map of a dynamic system:

$$\frac{dz(x,t)}{dt} = F(z(x,t)), z(0,x) = Vx$$

flow map is a nonlinear mapping, and the simplest choice of F is $F(z; U, W) = U\sigma(Wz)$. Choose the optimal $U, W(), \alpha$ to approximate f^* by $f^*(x) \ \alpha z(x,1)$
we can also observe deep learning from a control theory viewpoint.

1.7 Classical numerical analysis(approximation theory)

1. Define a "well posed" math model

(a) splines:hypothesis space $=C^1$ piece-wise cubic polynomials the data

$$I_n(f) = \frac{1}{n}\sum_{j=1}^{n}(f(x_j) - y_j)^2 + \lambda \int |D^2 f(x)|^2 dx$$

(b) finite element:hypothesis space$=C^0$ piece-wise polynomials

2. Identify the right function spaces,e.g. Sobolev/Besov spaces

   (a) direct and inverse approximation theorem (Bernstein and Jackson type theorems): f can be approximated by trig polynomials in L^2 to order s iff $f \in H^s, ||f||_{H^s}^2 = \sum_{k=0}^{s} ||\nabla^k f||_{L^2}^2$

   (b) functions of interest are in the right spaces(PDE theory,real analysis,etc)

3. Optimal error estimates

   (a) A priori estimates(for piece-wise linear finite elements,$\alpha = 1/d, s = 2$):$||f_m - f^*||_{H^1} \leq Cm^{-\alpha}||f^*||_{H^s}$

   (b) A posteriori estimates(say in finite elements):$||f_m - f^*||_{H^1} \leq Cm^{-\alpha}||f_m||_h$

1.8 Another benchmark:High dimensional integration

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

$$I(g) = \int_x g(x)d\mu, I_n(g) = \frac{1}{n}\sum_j g(x_j)$$

$$E(I(g) - I_n(g))^2 = \frac{1}{n}Var(g)$$

The $O(1/\sqrt{n})$ rate is the best we can hope for. However,var(g) can be very large in high dimension. That's why variance reduction is important.

1.9 The data we have is limited

We work with the empirical risk, but our final target is to minimize the population risk. Here are some difficulty we need to overcome: First,the parameter set $\hat{\theta}$ is far from being unique when $m > n$(such $\hat{\theta}$ form a $m - n$ dimensional manifold). Second, $\hat{\theta}$ is highly correlated with the data set S.

1.10 Estimate the generalization gap

"Generalization gap"$=\hat{R}(\hat{\theta}) - \hat{R}_n(\hat{\theta}) = I(g) - I_n(g)$, $g(x) = (f(x,\theta) - f^*(x))^2$

$$I(g) = \int_{X=[-1,1]^d} g(x)d\mu, I_n(g) = \frac{1}{n}\sum_j g(x_j)$$

For fixed g=h,we have $|I(h) - I_n(h)|\ \frac{1}{\sqrt{n}}$
For Lipschitz functions,$sup_{||h||\leq 1}|I(h) - I_n(h)|\ \frac{1}{n^{1/d}}$
For functions in Barron space, to be defined later: $sup_{||h||\leq 1}|I(h) - I_n(h)|\ \frac{1}{\sqrt{n}}$

1.11 Rademacher complexity

Let $\mathscr{H}$ be a set of functions, and $S = (x_1, x_2, ..., x_n)$ be a set of data points. Then, the Rademacher complexity of$\mathscr{H}$ with respect to S is defined as

$$\hat{R}_S(\mathscr{H}) = \frac{1}{n}\mathrm{E}_\zeta[sup_{h\in\mathscr{H}} \sum_{i=1}^{n} \zeta_i h(x_i)]$$

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

Theorem 1.1 (Rademacher complexity and the generalization gap) *Given a function class*$\mathscr{H}$*,for any* $\delta \in (0, 1)$*, with probability at least* $1 - \delta$ *over the random samples* $S = (x_1, ..., x_n)$

$$sup_{h\in\mathscr{H}}|E_x[h(x)] - \frac{1}{n}\sum_{i=1}^{n} h(x_i)| \leq 2\hat{R}_S(\mathscr{H}) + sup_{h\in\mathscr{H}}||h||_\infty \sqrt{\frac{\log{(2/\delta)}}{2n}}$$

$$sup_{h\in\mathscr{H}}|E_x[h(x)] - \frac{1}{n}\sum_{i=1}^{n} h(x_i)| \geq \frac{1}{2}\hat{R}_S(\mathscr{H}) - sup_{h\in\mathscr{H}}||h||_\infty \sqrt{\frac{\log{(2/\delta)}}{2n}}$$

If $\mathscr{H}$ contains a single function, then $\hat{R}_S(\mathscr{H})$ $O(1/\sqrt{n})$
If $\mathscr{H}$ contains functions that can fit any random values on S, then $\hat{R}_S(\mathscr{H})$ $O(1)$
If $\mathscr{H}$ = unit ball in Barron space, then $\hat{R}_S(\mathscr{H})$ $O(1/\sqrt{n})$
If $\mathscr{H}$ = unit ball in Lipschitz space, then $\hat{R}_S(\mathscr{H})$ $O(1/n^{1/d})$
If $\mathscr{H}$ = unit ball in C^0: $\hat{R}_S(\mathscr{H})$ $O(1)$

2 Curse of Dimensionality

2.1 Models that suffer from the curse of dimensionality.

$$\text{generational error} = \mathcal{O}(m^{-\alpha d} + n^{-\beta d})$$

1. piece-wise polynomial approximation.

2. wavelets with fixed wavelet basis.

2.2 Models that do not suffer from the curse of dimensionality.

1. random feature model.

2. two-layer neural network, i.e. adaptive random feature model.

3. residual neural network.

3 Random feature model

Let $\{\phi(,w)\}$ be a collection of random features, π be a probability distribution of the random variable ω.

$$\mathcal{H}_k - \{f : f(x) = \int a(\omega)\phi(x;\omega)d\pi(\omega)\}$$

with norm defined as $||f||^2_{\mathcal{H}_k} = \mathbb{E}_{\omega\sim\pi}(|a(\omega)|^2)$.
This is related to RKHS with kernel:

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

Since $\mathcal{H}_k$ is the completion of $\{\sum_{i=1}^m \alpha_i k(x_i,x)\}$ w.r.t. inner product given by

$$(\sum_i \alpha_i k(x_i,x), \sum_j \beta_j k(\hat{x_j},x)) = \sum_{i,j} \alpha_i \beta_j k(x_i,\hat{x_j})$$

To approximate function in space $\mathcal{H}_k$, we can sample $\{\omega_k\}$ w.r.t. π, then construct

$$\mathcal{H}_m(\{\omega_k\}) = \{f(x;a) = \frac{1}{m}\sum_{i=1}^m \alpha_i \phi(x,\omega_i)\}$$

To get a prior estimate, we need to minimize

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

Suppose $\hat{\theta}_n = \text{argmin} L_n(\theta)$. Then we can prove the generalization error $\mathcal{R}(\hat{\theta}_n) = \mathcal{O}(\frac{1}{m} + \sqrt{\frac{1}{n}})$.

4 Shallow neural networks

We generalize the two layer neural network to Barron space, whose function $f : [0,1]^d \to \mathbb{R}$ is in the following form:

$$f(x) = \int_\Omega a\sigma(b^T x + c)\rho(da,db,dc), \quad x \in [0,1]^d$$

where $\Omega = \mathbb{R}^1 \times \mathbb{R}^d \times \mathbb{R}^1$, and ρ is a prob distribution on Ω. We can then define $\mathcal{B}_p = \{f \in C^0 : ||f||_{\mathcal{B}_p} < \infty$, where $||f||_{\mathcal{B}_p}\} = \inf_\rho (E_\rho[|a|^p(||b||_1 + |c|)^p])^{1/p}$.
According to Barron and Klusowski, if $\int_{\mathbb{R}^d} ||\omega||_1^2 |\hat{f}(\omega)| d\omega < \infty$, then f admits such a representation.
Next we consider approximation by functions in the Barron space. For any f, we can construct $f_m(x) = \frac{1}{m}\sum_{j=1}^m a_j \sigma(b_j^T x + c_j)$ s.t. $||f - f_m||_2$ can be bounded by $\mathcal{O}(\sqrt{\frac{1}{m}})$. In the other direction, if $f_m \to f \in C^0$, $f_m \in \mathcal{N}_{p,C} := \{\frac{1}{m}\sum_{j=1}^m a_j\sigma(b_j^T x + c_j) : \frac{1}{m}\sum_{j=1}^m |a_j|^p(||b_j||_1 x + ||c_j||)^p \leq C, m \in \mathbb{N}^+\}$, then f lies in the Barron space.
Furthermore, the complexity of the unit ball can be bounded by $\mathcal{O}(\sqrt{\frac{1}{n}})$.
To get a prior estimate, we need to minimize

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

Suppose $\hat{\theta}_n = \text{argmin} L_n(\theta)$, where $||\theta||_p = ||f(\cdot,\theta)||_{\mathcal{B}_1}$. Then we can prove the generalization error $\mathcal{R}(\hat{\theta}_n) = \mathcal{O}(\frac{1}{m} + \sqrt{\frac{1}{n}})$.

A posterior estimate can be written as $|\mathcal{R}(\theta) - \hat{\mathcal{R}}_n(\theta)| \leq |||\theta|||\sqrt{\frac{1}{n}}$, where $|||\cdot|||$ is some norm, yet this bound requires a lot refinement.

4.1 Relation to random feature model.

Since f in Barron space can be reformulated as

$$f(x) = \int a(w)\sigma(w^T x)\pi(dw).w = (b, c)$$

Define

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

Compare it to the definition of RKHS $\mathcal{H}_k$, we discover the main difference is that distribution of w in random feature is predetermined, yet distribution of w in neural network can be learned. We can write this observation as

$$\mathcal{B}_2 = \bigcup_\pi \mathcal{H}_{k_\pi}$$

In summary, shallow neural network can be understood as adaptive kernel method.

5 Deep network

Consider the following schemes of residual neural network:

$$z_{0,L}(x) = Vx$$

$$z_{l+1,L}(x) = z_{l,L}(x) + \frac{1}{L}U_l\sigma(W_l z_{l,L}(x)), \quad (U, W) \sim \rho$$

which can be rewritten in the continuous form:

$$z(x, 0) = Vx$$

$$\frac{d}{dt}z(x,t) = \mathbb{E}_{(U,W)\sim\rho}U\sigma(Wz(x))$$

We want $z_{L,L}(x) \to z(x, 1)$ a.s. as $L \to \infty$. By the composition law of large number, we need only to assure $\mathbb{E}_\rho |||U||W|||_F^2 < \infty$, where $|\cdot|$ means taking element-wise absolute value.
Define $f_{\alpha,\rho,V}(x) = \alpha^T z(x, 1), \quad \alpha \in \mathbb{R}^D$. Notice that here f is defined by input layer V, output layer α, and the distribution of hidden layers ρ.
Define spaces $\mathcal{D}_1, \mathcal{D}_2$ separately, according to some norm as follow:

$$||f||_{\mathcal{D}_1} = \inf_f ||\alpha||_1 || \exp(E|U||W|)||_{1,1}||V||_{1,1}$$

$$||f||_{\mathcal{D}_2} = \inf_f ||\alpha||_F || \exp(\sqrt{E(|U||W|)^2}||_F||V||_F$$

where $|\cdot|, (\cdot)^2$ and $\sqrt{\cdot}$ are element-wise operations. Let

$$\mathcal{D}_i = \{f = f_{\alpha,\rho,V}, ||f||_{\mathcal{D}_i} < \infty\}, \quad i = 1, 2.$$

To get a prior estimate, we need to minimize

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

Suppose $\hat{\theta}_n = \text{argmin}J(\theta)$. Then we can prove the generalization error $\mathcal{R}(\hat{\theta}_n) = \mathcal{O}(\frac{1}{L} + \sqrt{\frac{1}{n}})$.
The form of posterior estimate and Rademacher complexity is just the same as the former case. To consider different parameter distribution across the layers, we can relate ρ to time t, i.e. the order of layers, and modify above definition to this general case. We can introduce direct and inverse approximation theorems under this space:

1. Direct: Let $f \in L^2(D) \cap \mathcal{D}_2$. There exists a residue-type neural network $f_L(\cdot, \theta)$ of input dimension $d+1$ and depth L such that $||f_L||_P \lesssim ||f||_{c1}^3$ and

$$\int_{D_0} |f(x) - f_L((x,1),\theta)|^2 dx \to 0 \lesssim \frac{||f||_{c2}^2}{L}$$

Furthermore, if $f = f_{\alpha,\rho_t,V}$ and ρ_t is a Lipschitz continuous in t, then

$$\int_{D_0} |f(x) - f_L((x,1),\theta)|^2 dx \lesssim \frac{||f||_{\mathcal{D}_2}^2}{L}$$

2. Inverse: Let $f \in L^2(D)$. Assume that there is a sequence of residual networks $\{f_L(x)\}_{L=1}^{\infty}$ with increasing depth such that $||f - f_L|| \to 0$. Assume further that the parameters are (entry-wise) bounded, then there exist $\alpha, \{\rho_t\}, V$, s.t. $f(x) = f_{\alpha,\rho_t,V}(x)$.

5.1 Relation of shallow neural network.

Define $\hat{f}(x)$ by

$$\hat{f}(x) = e_1^T z(x,1)$$

$$\frac{d}{dt} z(x,t) = \mathbb{E}_{\rho(a,b,c)} \begin{bmatrix} a \\ 0 \\ 0 \end{bmatrix} \sigma([0, b^T, c] z(x,t))$$

$$z(x,0) = \begin{bmatrix} 0 \\ x \\ 1 \end{bmatrix}$$

Then $\hat{f}(x) = \int_\Omega a\sigma(b^T x + c)\rho(da, db, dc)$, which indicates wider representation range of residual networks. To make this idea concrete, we can prove $\mathcal{B}_2 \subset \mathcal{D}_2$, and for $f \in \mathcal{B}_2$, norm satisfies $||f||_{\mathcal{D}_2} \leq \sqrt{d+1}||f||_{\mathcal{B}_2}$.

6 Summary

Inspired by Monte Carlo method:

$$(I(g) - I_n(g))^2 \sim \frac{\gamma(g)}{n}$$

$\gamma(g)$ is some kind of variance of g, depending on the details of the Monte Carlo.
For the machine learning model, we want to prove:

$$\mathcal{R}(\theta) \leq \frac{\gamma_1(f^*)}{m} + \frac{\gamma_2(f^*)}{\sqrt{n}}$$

where γ_1, γ_2 is related to the model and target function f^*. For this purpose, we need to:

1. Find probablistic interpretation of the machine learning model.

2. Identify the right function spaces/norms for approximation theory.

3. Study the Rademacher complexity.

4. Should think in terms of approximating probability distributions $(\rho, \{\rho_t\})$, not specific parameters or weights.

5. Regularization is important for obtaining well-posed models. Relying on ”implicit regularization” might NOT be the way to go. (All the generation bounds requires explicit regulation.)

A main mathematical question is to identify and study low complexity spaces in high dimension.
We can consider open problems

1. What about these function spaces? How big are they? Size of the norms classification problems.

2. Other regularization, other network structures, other activation functions?