On the Hardness of Learning Two-layer Neural Networks

Instructor: Weinan E

Mathematical Introduction to Machine Learning
MAT 490/APC 490

Princeton University, Spring 2021

Motivation

- For two-layer neural networks, we have defined the Barron spaces. Both the approximation and estimation error for target functions in these spaces obey the Monte-Carlo rate, which is free of the curse of dimensionality (CoD). A nature questions is then: Do there exist algorithms can learn these functions efficiently?

Motivation

- For two-layer neural networks, we have defined the Barron spaces. Both the approximation and estimation error for target functions in these spaces obey the Monte-Carlo rate, which is free of the curse of dimensionality (CoD). A nature questions is then: Do there exist algorithms can learn these functions efficiently?
- We say an algorithm $\mathcal{A}$ is efficient in learning a function class $\mathcal{F}$, if for every $\varepsilon > 0$, $f^* \in \mathcal{F}$, the time complexity for returning an solution $\hat{f}$ such that $\|\hat{f} - f^*\| \leq \varepsilon$ satisfies:

$$\text{Time complexity} = \mathrm{poly}(1/\varepsilon, d).$$

Motivation

- For two-layer neural networks, we have defined the Barron spaces. Both the approximation and estimation error for target functions in these spaces obey the Monte-Carlo rate, which is free of the curse of dimensionality (CoD). A nature questions is then: Do there exist algorithms can learn these functions efficiently?
- We say an algorithm $\mathcal{A}$ is efficient in learning a function class $\mathcal{F}$, if for every $\varepsilon > 0$, $f^* \in \mathcal{F}$, the time complexity for returning an solution $\hat{f}$ such that $\|\hat{f} - f^*\| \leq \varepsilon$ satisfies:

$$\text{Time complexity} = \mathrm{poly}(1/\varepsilon, d).$$

- Note that in the highly over-parameterized setting, GD can find a global minimum of the empirical risk exponentially fast. However, this solution is essentially a kernel predictor, whose generalization error suffers from CoD. Hence, the total time complexity still suffers from CoD.

Motivation

- For two-layer neural networks, we have defined the Barron spaces. Both the approximation and estimation error for target functions in these spaces obey the Monte-Carlo rate, which is free of the curse of dimensionality (CoD). A nature questions is then: Do there exist algorithms can learn these functions efficiently?
- We say an algorithm $\mathcal{A}$ is efficient in learning a function class $\mathcal{F}$, if for every $\varepsilon > 0$, $f^* \in \mathcal{F}$, the time complexity for returning an solution $\hat{f}$ such that $\|\hat{f} - f^*\| \leq \varepsilon$ satisfies:

$$\text{Time complexity} = \mathrm{poly}(1/\varepsilon, d).$$

- Note that in the highly over-parameterized setting, GD can find a global minimum of the empirical risk exponentially fast. However, this solution is essentially a kernel predictor, whose generalization error suffers from CoD. Hence, the total time complexity still suffers from CoD.
- In this lecture, we will show that the class of Barron functions is not efficiently learnable.

Learning intersections of halfspaces

- Let $\boldsymbol{x} \in \mathcal{X} = \{-1, 1\}^d$ and consider the binary classification problem, i.e., $f^* : \mathcal{X} \mapsto \{-1, 1\}$.
- Let $\sigma_{0,1}$ be the step function, i.e., $\sigma_{0,1}(t) = 1(t \geq 0)$.

Learning intersections of halfspaces

- Let $\boldsymbol{x} \in \mathcal{X} = \{-1, 1\}^d$ and consider the binary classification problem, i.e., $f^* : \mathcal{X} \mapsto \{-1, 1\}$.
- Let $\sigma_{0,1}$ be the step function, i.e., $\sigma_{0,1}(t) = 1(t \geq 0)$.

We will need the following hardness result for learning the intersection of halfspaces.

Theorem 1 (Theorem 1.2, Kalai, Klivans, 2008)

Let $\mathcal{H} = \{\boldsymbol{x} \mapsto \sigma_{0,1}(\boldsymbol{w}^T \boldsymbol{x} - b - 1/2) : b \in \mathbb{N}, \boldsymbol{w} \in \mathbb{N}^d, |b| + \|\boldsymbol{w}\|_1 \leq \mathrm{poly}(d)\}$. *Define*

$$\mathcal{H}_K = \{\boldsymbol{x} \mapsto h_1(\boldsymbol{x}) \wedge h_2(\boldsymbol{x}) \wedge \cdots \wedge h_K(\boldsymbol{x}) \ : \ h_i \in \mathcal{H}\}.$$

Assume $k \geq d^\rho$ *with* $\rho > 0$. *Then, under a certain* **cryptographic** *assumption,* $\mathcal{H}_K$ *is not efficiently learnable.*

- The proof is to reduce it to some classical hard problems, e.g., k-coloring. The **cryptographic** assumption means that we assume that these hard problems are indeed hard in certain sense. If this assumption does not hold, the modern cryptosystem can be broken in a polynomial time.
- We will show two-layer neural networks can simulate the functions in $\mathcal{H}_K$.

Hardness of learning two-layer ReLU networks

Theorem 2 (Livni, et al, 2014)

Let $\mathcal{X} = \{-1, 1\}^d$, and $\mathcal{G}$ be the class of functions with the Barron norm bounded by $\mathrm{poly}(d)$. Then, $\mathcal{G}$ is not efficiently learnable.

- The intuition is that 2-layer neural network can simulate the intersections of hyperspaces.
- The step function can be approximated by two ReLU functions very well:

$$\sigma_{0,1}(t) = \lim_{a \to \infty} \texttt{ReLU}(at) - \texttt{ReLU}(at - 1).$$

Hardness of learning two-layer neural networks

Proof:

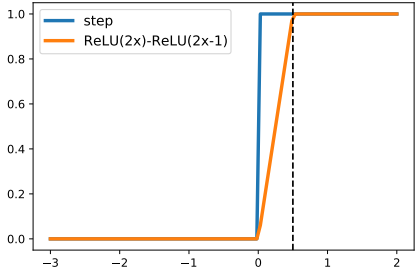
- Let $c(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} - b - 1/2$. Since $\boldsymbol{w} \in \mathbb{N}^d, \boldsymbol{x} \in \{-1, 1\}^d, b \in \mathbb{N}$, we have $|c(\boldsymbol{x})| \geq 1/2$. Assume $\|\boldsymbol{w}\|_1 + |b| \leq \mathrm{poly}(d)$.
- Consider k hyperplanes $\{c_i\}_{i=1}^k$. Let $h_i(\boldsymbol{x}) = \sigma_{0,1}(c_i(\boldsymbol{x})) \in \mathcal{H}$.
- Let

$$g(\boldsymbol{x}) = \frac{1}{2k}\left(\sum_{i=1}^{k}\left(\mathtt{ReLU}(2c_i(\boldsymbol{x})) - \mathtt{ReLU}(2c_i(\boldsymbol{x}) - 1)\right) - k + \frac{1}{3}\right).$$

  Obviously, g is a 2-layer ReLU network with the path norm bounded by

$$\frac{1}{2k}\left(k + \frac{1}{3} + \sum_{i=1}^{k}(2(2\|\boldsymbol{w}_i\|_1 + |b_i| + 1/2) + 1\right) = \mathrm{poly}(d).$$

- The blue part is equal to $\sigma_{0,1}(c_i(\boldsymbol{x}))$ due to $\sigma_{0,1}(z) = \texttt{ReLU}(2z) - \texttt{ReLU}(2z-1)$ for $|z| \geq 1/2$.



- We can verify that

$$\text{sign}(g(\boldsymbol{x})) = h_1(\boldsymbol{x}) \wedge h_2(\boldsymbol{x}) \wedge \cdots \wedge h_k(\boldsymbol{x}), \ \forall \boldsymbol{x} \in \{-1, 1\}^d.$$

Remarks

- Note that similar results also hold for two-layer networks with the sigmoid activation function, since the sigmoid function can approximate the step function as well. See [Livni, et al, 2014] for more details.
- The above results rely on the hardness of certain classical hard problems.
  - Pros: It is implied that the hardness holds for any algorithms.
  - Cons: This perspective is too abstract. It does not provide any concrete examples and intuitions behind the hardness of training.
- In the following, we will provide some understandings from a landscape perspective.

Orthonormal classes

Denote by $\mathcal{D}$ the distribution over the input space $\mathcal{X}$. For any two functions f_1, f_2, define the inner product $\langle f_1, f_2 \rangle = \mathbb{E}_{\boldsymbol{x} \sim \mathcal{D}}[f_1(\boldsymbol{x}) f_2(\boldsymbol{x})]$.

Definition 3 (Orthonormal class)

Let $\mathcal{F}$ be a function class. We say that it is an orthonormal class, if $\langle f_i, f_j \rangle = \delta_{i,j}$ for any $f_i, f_j \in \mathcal{F}$.

Orthonormal classes

Denote by $\mathcal{D}$ the distribution over the input space $\mathcal{X}$. For any two functions f_1, f_2, define the inner product $\langle f_1, f_2 \rangle = \mathbb{E}_{\boldsymbol{x} \sim \mathcal{D}}[f_1(\boldsymbol{x}) f_2(\boldsymbol{x})]$.

Definition 3 (Orthonormal class)

Let $\mathcal{F}$ be a function class. We say that it is an orthonormal class, if $\langle f_i, f_j \rangle = \delta_{i,j}$ for any $f_i, f_j \in \mathcal{F}$.

Definition 4 (Statistical query (SQ) dimension)

Let $\mathcal{F}$ be a function class with $\|f\| = 1$ for any $f \in \mathcal{F}$. The SQ dimension of $\mathcal{F}$ is the largest n such that there is a n-sized subset $\{f_i\}_{i=1}^n \subset \mathcal{F}$ such that $\langle f_i, f_j \rangle \leq \frac{1}{n^2}$ for any $i \neq j$ and $i, j \in [n]$.

The SQ dimension characterizes the number of "nearly" orthonormal functions in $\mathcal{F}$.

Orthonormal classes

Denote by $\mathcal{D}$ the distribution over the input space $\mathcal{X}$. For any two functions f_1, f_2, define the inner product $\langle f_1, f_2 \rangle = \mathbb{E}_{\boldsymbol{x} \sim \mathcal{D}}[f_1(\boldsymbol{x}) f_2(\boldsymbol{x})]$.

Definition 3 (Orthonormal class)

Let $\mathcal{F}$ be a function class. We say that it is an orthonormal class, if $\langle f_i, f_j \rangle = \delta_{i,j}$ for any $f_i, f_j \in \mathcal{F}$.

Definition 4 (Statistical query (SQ) dimension)

Let $\mathcal{F}$ be a function class with $\|f\| = 1$ for any $f \in \mathcal{F}$. The SQ dimension of $\mathcal{F}$ is the largest n such that there is a n-sized subset $\{f_i\}_{i=1}^n \subset \mathcal{F}$ such that $\langle f_i, f_j \rangle \leq \frac{1}{n^2}$ for any $i \neq j$ and $i, j \in [n]$.

The SQ dimension characterizes the number of "nearly" orthonormal functions in $\mathcal{F}$.

- Let $\mathcal{B}_d = \{f_m(\cdot; \theta) : \|\theta\|_{\mathcal{P}} \leq C d^2, m \in \mathbb{N}\}$. Here, f_m is a two-layer neural network and $C > 0$ is sufficiently large. We will show that the SQ dimension of $\mathcal{B}_d$ is exponentially in d.
- We will show that if the SQ dimension of a function class $\mathcal{F}$ is exponentially large, the learning of $\mathcal{F}$ will be hard in some sense.

Parity functions

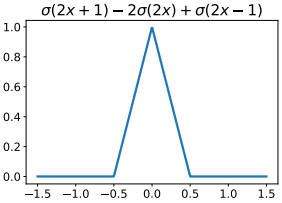
- $\mathcal{F}_1 = \{f_{\boldsymbol{v}}(\boldsymbol{x}) = (-1)^{\langle \boldsymbol{v}, \boldsymbol{x} \rangle} : \boldsymbol{v} \in \{0,1\}^d\}$, where $\boldsymbol{x} \in \{0,1\}^d$.
- Consider $\mathcal{D} = \text{Unif}(\{0,1\}^d)$. Then, we have

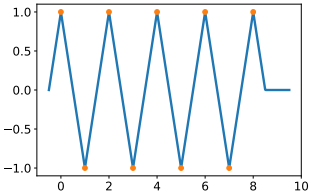
$$\langle f_{\boldsymbol{v}}, f_{\boldsymbol{v}'} \rangle = \mathbb{E}_{\boldsymbol{x}}[(-1)^{(\boldsymbol{v}+\boldsymbol{v}')^T \boldsymbol{x}}] = \mathbb{E}_{\boldsymbol{x}}[\prod_{i=1}^{d} (-1)^{(v_i + v_i')x_i}] \tag{1}$$

$$= \prod_{i=1}^{d} \mathbb{E}_{x_i}[(-1)^{(v_i+v_i')^T x_i}] = \delta_{\boldsymbol{v},\boldsymbol{v}'}. \tag{2}$$

Hence, $\mathcal{F}_1$ is an orthonormal class with $|\mathcal{F}_1| = 2^d$.

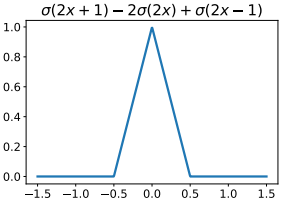
Parity functions as two-layer neural networks

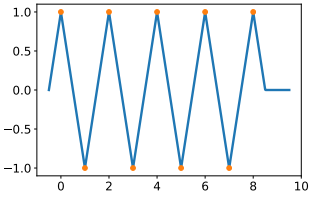




- Observation:
  - The function $(-1)^s$ for $s \in \mathbb{N}$ can be implemented using the triangle wave.
  - The triangle wave can be written as a linear combination of the hat function, which is a linear combination of ReLU function (left figure).

Parity functions as two-layer neural networks





- Observation:
  - The function $(-1)^s$ for $s \in \mathbb{N}$ can be implemented using the triangle wave.
  - The triangle wave can be written as a linear combination of the hat function, which is a linear combination of ReLU function (left figure).
- Note that $\boldsymbol{v}^T\boldsymbol{x} \in \{0, 1, \ldots, d\}$. Let σ be the ReLU function. Then,

$$(-1)^{\boldsymbol{v}^T\boldsymbol{x}} = \sigma_{tri}(\boldsymbol{v}^T\boldsymbol{x}) = \sum_{i=0}^{d}(-1)^i\sigma_{hat}(\boldsymbol{v}^T\boldsymbol{x} - i) \tag{3}$$

$$= \sum_{i=0}^{d}(-1)^i\left(\sigma(2\boldsymbol{v}^T\boldsymbol{x} - 2i + 1) - 2\sigma(2\boldsymbol{v}^T\boldsymbol{x} - 2i) + \sigma(2\boldsymbol{v}^T\boldsymbol{x} - 2i - 1)\right) \tag{4}$$

- It is easy to show that the path norm of this network is bounded by Cd^2.

The cosine neuron

Cosine neuron: Let $\mathcal{D} = \mathcal{N}(0, I_d)$. Define

$$\mathcal{F}_2 := \left\{ f_{\boldsymbol{w}}(\boldsymbol{x}) = \cos(2\pi \boldsymbol{w}^T \boldsymbol{x}) : \|\boldsymbol{w}\| = \sqrt{d} \right\}.$$

Then the SQ dimension of $\mathcal{F}_2$ is larger than $C_1 e^{C_2 \sqrt{d}}$, where C_1, C_2 are absolute positive constants.

The cosine neuron (Cont'd)

Proof:

- Let $\boldsymbol{w}_i \overset{iid}{\sim} \mathbb{S}^{d-1}$ with $i = 1, \ldots, n$. Then, taking the union bound gives us

$$\mathbb{P}\left\{ \max_{i \neq j} |\boldsymbol{w}_i^T \boldsymbol{w}_j| \geq t \right\} \leq 2n^2 e^{-C_1 d t^2}.$$

  Let $t = 1/d^{1/4}$ and $2n^2 e^{-C_1 t^2} \leq 1/2$. We have $n = e^{-C_2 d^{1/2}}/2$.

- By the rescaling to $\sqrt{d}\,\mathbb{S}^{d-1}$, it is implied that there exist $n = 0.5e^{-C_2 d^{1/2}}$ points $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_n$ such that

$$\|\boldsymbol{w}_i\|^2 = d, \qquad |\langle \boldsymbol{w}_i, \boldsymbol{w}_j \rangle| \leq \sqrt{d}\sqrt{d}\frac{1}{d^{1/4}} = d^{3/4}, \forall i \neq j.$$

- Denote by $\cos_{\boldsymbol{w}}(\boldsymbol{x}) = \cos(2\pi \boldsymbol{w}^T \boldsymbol{x})$ and φ^2 the density function of the standard Gaussian. Then, we have for $i \neq j$,

$$\begin{aligned}\langle \cos_{\boldsymbol{w}_i}, \cos_{\boldsymbol{w}_j} \rangle &= \int \varphi^2(\boldsymbol{x}) \cos(\boldsymbol{w}_i^T \boldsymbol{x}) \cos(\boldsymbol{w}_j^T \boldsymbol{x}) d\boldsymbol{x} = \langle \cos_{\boldsymbol{w}_i} \varphi, \cos_{\boldsymbol{w}_j} \varphi \rangle_{L^2(\mathbb{R}^d)} \\ &= \langle \widehat{\cos}_{\boldsymbol{w}_i} * \hat{\varphi}, \widehat{\cos}_{\boldsymbol{w}_j} * \hat{\varphi}, \rangle_{L^2(\mathbb{R})} \\ &= \frac{1}{4} \int_{\mathbb{R}^d} (\hat{\varphi}(\boldsymbol{\xi} - \boldsymbol{w}_i) + \hat{\varphi}(\boldsymbol{\xi} + \boldsymbol{w}_i))(\hat{\varphi}(\boldsymbol{\xi} - \boldsymbol{w}_j) + \hat{\varphi}(\boldsymbol{\xi} + \boldsymbol{w}_j)) d\boldsymbol{\xi}.\end{aligned}$$

Here, $\hat{\varphi}(\boldsymbol{\xi}) = \frac{(4\pi)^{d/2}}{(2\pi)^{d/4}} e^{-4\pi^2 \|\boldsymbol{\xi}\|^2}$ is a Gaussian-like function. Since, $\|\boldsymbol{w}_i - \boldsymbol{w}_j\|^2 = 2d - 2\langle \boldsymbol{w}_i, \boldsymbol{w}_j \rangle \geq 2d - 2d^{3/4} \geq d/2$ for sufficiently large d.

- Denote by $\cos_{\boldsymbol{w}}(\boldsymbol{x}) = \cos(2\pi \boldsymbol{w}^T \boldsymbol{x})$ and φ^2 the density function of the standard Gaussian. Then, we have for $i \neq j$,

$$\begin{aligned}\langle \cos_{\boldsymbol{w}_i}, \cos_{\boldsymbol{w}_j} \rangle &= \int \varphi^2(\boldsymbol{x}) \cos(\boldsymbol{w}_i^T \boldsymbol{x}) \cos(\boldsymbol{w}_j^T \boldsymbol{x}) d\boldsymbol{x} = \langle \cos_{\boldsymbol{w}_i} \varphi, \cos_{\boldsymbol{w}_j} \varphi \rangle_{L^2(\mathbb{R}^d)} \\ &= \langle \widehat{\cos}_{\boldsymbol{w}_i} * \hat{\varphi}, \widehat{\cos}_{\boldsymbol{w}_j} * \hat{\varphi}, \rangle_{L^2(\mathbb{R})} \\ &= \frac{1}{4} \int_{\mathbb{R}^d} (\hat{\varphi}(\boldsymbol{\xi} - \boldsymbol{w}_i) + \hat{\varphi}(\boldsymbol{\xi} + \boldsymbol{w}_i))(\hat{\varphi}(\boldsymbol{\xi} - \boldsymbol{w}_j) + \hat{\varphi}(\boldsymbol{\xi} + \boldsymbol{w}_j)) d\boldsymbol{\xi}.\end{aligned}$$

  Here, $\hat{\varphi}(\boldsymbol{\xi}) = \frac{(4\pi)^{d/2}}{(2\pi)^{d/4}} e^{-4\pi^2 \|\boldsymbol{\xi}\|^2}$ is a Gaussian-like function. Since, $\|\boldsymbol{w}_i - \boldsymbol{w}_j\|^2 = 2d - 2\langle \boldsymbol{w}_i, \boldsymbol{w}_j \rangle \geq 2d - 2d^{3/4} \geq d/2$ for sufficiently large d.
- Hence, there exist a constant $C_3 > 0$ such that

$$\langle \cos_{\boldsymbol{w}_i}, \cos_{\boldsymbol{w}_j} \rangle \leq \sup_{\|\boldsymbol{w} - \boldsymbol{v}\|^2 \geq C_3 d} \int \hat{\varphi}(\boldsymbol{\xi} + \boldsymbol{w}) \hat{\varphi}(\boldsymbol{\xi} + \boldsymbol{v}) d\boldsymbol{\xi} \tag{5}$$

$$= e^{-2\pi^2 \|\boldsymbol{w} - \boldsymbol{v}\|^2} \leq e^{-2\pi^2 C_3 d} \leq \frac{1}{n^2}. \tag{6}$$

  Here, $n = 0.5 e^{-C_2 \sqrt{d}}$. Therefore, the SQ dimension of $\mathcal{F}_2$ is at least exponentially in $\sqrt{d}$.

Approximating the cosine neuron with two-layer neural networks

- Note that Barron function is defined on a compact domain. We can consider $\mathcal{D} = \text{Unif}(\sqrt{d}\,\mathbb{S}^{d-1})$. This does not change the analysis too much since when $d \gg 1$, $\mathcal{N}(0, I_d)$ is similar to $\text{Unif}(\sqrt{d}\,\mathbb{S}^{d-1})$.

Approximating the cosine neuron with two-layer neural networks

- Note that Barron function is defined on a compact domain. We can consider $\mathcal{D} = \text{Unif}(\sqrt{d}\,\mathbb{S}^{d-1})$. This does not change the analysis too much since when $d \gg 1$, $\mathcal{N}(0, I_d)$ is similar to $\text{Unif}(\sqrt{d}\,\mathbb{S}^{d-1})$.
- Through rescaling, we can consider the target function $f^*(\boldsymbol{x}) = \cos(\sqrt{d}\boldsymbol{w}^T\boldsymbol{x})$ and $\boldsymbol{x} \in [-1, 1]^d$. By the spectral characterization of Barron functions, we have

$$\|f^*\|_{\mathcal{B}} \leq 2\int_{\mathbb{R}^d} \|\boldsymbol{\xi}\|_1^2 |\mathcal{F}(f^*)(\boldsymbol{\xi})| d\boldsymbol{\xi} + 2|f^*(0)| + 2\|\nabla f(0)\|_1 \leq 2\|\sqrt{d}\boldsymbol{w}\|_1^2 + 2 \tag{7}$$

$$\leq Cd^3. \tag{8}$$

Remark

$\mathcal{F}_\phi := \{\phi(\boldsymbol{w}^T\boldsymbol{x}) : \|\boldsymbol{w}\| = \sqrt{d}\}$. $\mathcal{D} = \mathcal{N}(0, I_d)$. [Shamir, 2017] shows that as long as ϕ is periodic, under some mild condition, we have the SQ dimension of $\mathcal{F}_\phi$ is exponentially in d.

Gradients for an orthonormal class

Let $h(\cdot;\theta)$ be any parametric model. Denote by $R^f(\theta) = \mathbb{E}_{\boldsymbol{x}}[(h(\boldsymbol{x};\theta) - f(\boldsymbol{x}))^2]$ the risk. Then, we have the following theorem.

Theorem 5

Let $\mathcal{F}$ be an orthonormal class. Let P denote the uniform distribution over the space of $\mathcal{F}$ and $g(\theta) = \mathbb{E}_{f\sim P}[\nabla R^f(\theta)]$. We have

$$\mathbb{E}_{f\sim P}[(\nabla R^f(\theta) - g(\theta))^2] \leq \frac{\mathbb{E}_{\boldsymbol{x}}[\|\nabla_\theta h(\boldsymbol{x};\theta)\|^2]}{|\mathcal{F}|}. \tag{9}$$

Gradients for an orthonormal class

Let $h(\cdot;\theta)$ be any parametric model. Denote by $R^f(\theta) = \mathbb{E}_{\boldsymbol{x}}[(h(\boldsymbol{x};\theta) - f(\boldsymbol{x}))^2]$ the risk. Then, we have the following theorem.

Theorem 5

Let $\mathcal{F}$ be an orthonormal class. Let P denote the uniform distribution over the space of $\mathcal{F}$ and $g(\theta) = \mathbb{E}_{f\sim P}[\nabla R^f(\theta)]$. We have

$$\mathbb{E}_{f\sim P}[(\nabla R^f(\theta) - g(\theta))^2] \leq \frac{\mathbb{E}_{\boldsymbol{x}}[\|\nabla_\theta h(\boldsymbol{x};\theta)\|^2]}{|\mathcal{F}|}. \tag{9}$$

- If $|\mathcal{F}|$ is exponentially in d, e.g. the parity functions. The variance of gradients is exponentially small.

Gradients for an orthonormal class

Let $h(\cdot;\theta)$ be any parametric model. Denote by $R^f(\theta) = \mathbb{E}_{\boldsymbol{x}}[(h(\boldsymbol{x};\theta) - f(\boldsymbol{x}))^2]$ the risk. Then, we have the following theorem.

Theorem 5

Let $\mathcal{F}$ be an orthonormal class. Let P denote the uniform distribution over the space of $\mathcal{F}$ and $g(\theta) = \mathbb{E}_{f\sim P}[\nabla R^f(\theta)]$. We have

$$\mathbb{E}_{f\sim P}[(\nabla R^f(\theta) - g(\theta))^2] \leq \frac{\mathbb{E}_{\boldsymbol{x}}[\|\nabla_\theta h(\boldsymbol{x};\theta)\|^2]}{|\mathcal{F}|}. \tag{9}$$

- If $|\mathcal{F}|$ is exponentially in d, e.g. the parity functions. The variance of gradients is exponentially small.
- This theorem implies that the "information" about the target function contained in the gradient is exponentially small. Therefore, one would expect that gradient-based methods will be unlikely to learn the function class $\mathcal{F}$.

Proof:

- First, the gradient can be written as follows

$$\nabla_\theta R^f = \mathbb{E}_{\boldsymbol{x}}[(h(\boldsymbol{x};\theta) - f)\nabla_\theta h(\boldsymbol{x};\theta)] = C_\theta - \langle f, \nabla_\theta h(\boldsymbol{x};\theta)\rangle,$$

  where C_θ is independent of the target function f.

Proof:

- First, the gradient can be written as follows

$$\nabla_\theta R^f = \mathbb{E}_{\boldsymbol{x}}[(h(\boldsymbol{x};\theta) - f)\nabla_\theta h(\boldsymbol{x};\theta)] = C_\theta - \langle f, \nabla_\theta h(\boldsymbol{x};\theta)\rangle,$$

  where C_θ is independent of the target function f.

- Hence,

$$\mathbb{E}_f[(\nabla_\theta R^f - g(\theta))^2] \leq \mathbb{E}_f[\langle f, \nabla_\theta h(\boldsymbol{x};\theta)\rangle^2] \tag{10}$$

$$\leq \frac{1}{|\mathcal{F}|}\sum_f \langle f, \nabla_\theta h(\boldsymbol{x};\theta)\rangle^2 \tag{11}$$

$$\leq \frac{\mathbb{E}_{\boldsymbol{x}}[\|\nabla_\theta h(\boldsymbol{x};\theta)\|^2]}{|\mathcal{F}|}. \tag{12}$$

Extension to the nearly orthonormal classes

For the nearly orthonormal classes. Let n be the SQ dimension. Then, there exist $f_1, \ldots, f_n$ such that $\langle f_i, f_j \rangle \leq 1/n^2$ for $i \neq j$. Let $g = \nabla_\theta h(\cdot; \theta)$. WLOG, assume $\|g\| \leq 1$.
Then,

$$0 \leq \|g - \sum_i \langle g, f_i \rangle f_i\|^2 \tag{13}$$

$$= \|g\|^2 - \sum_i \langle g, f_i \rangle^2 + \sum_{i \neq j} \langle g, f_i \rangle \langle g, f_j \rangle \langle f_i, f_j \rangle \tag{14}$$

$$\leq \|g\|^2 - \sum_i \langle g, f_i \rangle^2 + \frac{n^2}{n^2}. \tag{15}$$

Therefore, let P be the uniform distribution over $f_1, \ldots, f_n$, we have

$$\mathbb{E}_{f \sim P}[(\nabla R^f(\theta) - g(\theta))^2] \leq \frac{1}{n} \sum_{i=1}^{n} \langle f_i, g \rangle^2 \leq \frac{2}{n}.$$

Therefore, we prove similar results for the nearly orthonormal classes, such as the cosine neuron class.

Hardness of learning with GD: Setup

Setup:

- Assume $\mathcal{F}$ to be an orthonormal class with $|\mathcal{F}| = 2^d$. Consider the binary classification with the hinge loss. The risk is given by

$$R^f(\theta) := \mathbb{E}_{\boldsymbol{x}}[\max(0, 1 - h(\boldsymbol{x}; \theta) f(\boldsymbol{x}))]. \tag{16}$$

- Assume $|h(\boldsymbol{x}; \theta)| \leq 1$ and $|f(\boldsymbol{x})| \leq 1$ for any $\boldsymbol{x} \in \mathcal{X}$. Then we have

$$\begin{aligned}\mathbb{E}_f[\|\nabla_\theta R^f(\theta)\|^2] &= \mathbb{E}_f \left(\mathbb{E}_{\boldsymbol{x}}[f(\boldsymbol{x}) \nabla_\theta h(\boldsymbol{x}; \theta)]\right)^2 \\ &= \frac{1}{|\mathcal{F}|} \sum_i \langle f_i, \nabla h(\cdot; \theta) \rangle^2 \leq \frac{\|\nabla_\theta h\|^2}{|\mathcal{F}|} \leq \frac{G_\theta}{2^d}.\end{aligned}$$

Remark: the above assumption holds for parity functions.

Hardness of learning with GD

Theorem 6

Assume the model satisfies that $\sup_{\boldsymbol{x}\in X}|h(\boldsymbol{x};\theta)|\leq 1$ *and* $\mathbb{E}_{\boldsymbol{x}}[\|\nabla_\theta h(\boldsymbol{x};\theta_1)-\nabla_\theta h(\boldsymbol{x};\theta_2)\|^2]\leq L\|\theta_1-\theta_2\|^2$. *Let* θ_0,θ_t^f *be the GD solution at time* 0 *and time* t, *respectively. Then, there exist* C_1,C_2 *such that*

$$\mathbb{E}_f[\|\theta_t^f-\theta_0\|^2]\leq C_1(e^{\frac{C_2 t}{2^{d/2}}}-1), \tag{17}$$

where C_1,C_2 *only depend on* L *and* θ_0.

Hardness of learning with GD

Theorem 6

Assume the model satisfies that $\sup_{\boldsymbol{x}\in X}|h(\boldsymbol{x};\theta)|\leq 1$ *and* $\mathbb{E}_{\boldsymbol{x}}[\|\nabla_\theta h(\boldsymbol{x};\theta_1)-\nabla_\theta h(\boldsymbol{x};\theta_2)\|^2]\leq L\|\theta_1-\theta_2\|^2$. *Let* θ_0,θ_t^f *be the GD solution at time* 0 *and time* t, *respectively. Then, there exist* C_1,C_2 *such that*

$$\mathbb{E}_f[\|\theta_t^f-\theta_0\|^2]\leq C_1(e^{\frac{C_2 t}{2^{d/2}}}-1), \tag{17}$$

where C_1,C_2 *only depend on* L *and* θ_0.

The above theorem implies that GD solution is exponentially close to the initialization in polynomial time. More rigorously, we have the following corollary.

Corollary 7

For any $T=poly(d)$, *there exists a* $f\in\mathcal{F}$ *such that*

$$\|\theta_t^f-\theta_0\|\leq C\frac{poly(d)}{2^d},\ \forall t\in[0,T]$$

where C *only depends on* L *and* θ_0.

Hardness of learning with GD

Proof:

- $G(\theta) = \mathbb{E}_{\boldsymbol{x}}[\|\nabla_\theta h(\boldsymbol{x};\theta)\|^2]$ satisfies

$$G(\theta) \le G(\theta_0) + 2L\|\theta - \theta_0\|^2. \tag{18}$$

- Therefore,

$$\frac{d\mathbb{E}_f[\|\theta_t^f - \theta_0\|^2]}{dt} = 2\mathbb{E}_f[\langle \theta_t^f - \theta_0, -\nabla_\theta R^f(\theta_t^f)\rangle] \tag{19}$$

$$\le \frac{1}{2^{\frac{d}{2}-1}} \sqrt{\mathbb{E}_f[\|\theta_t^f - \theta_0\|^2]\mathbb{E}_f[G(\theta_t^f)]} \tag{20}$$

$$\le \frac{1}{2^{\frac{d}{2}-1}} \sqrt{\mathbb{E}_f[\|\theta_t^f - \theta_0\|^2]\mathbb{E}_f[G(\theta_0) + 2L\|\theta_t^f - \theta_0\|^2]}. \tag{21}$$

Hardness of learning with GD

Proof: Let $\delta_t = \sqrt{\mathbb{E}_f[\|\theta_t^f - \theta_0\|^2]}$. Then, we have

$$\dot{\delta}_t \leq 2^{2-\frac{d}{2}}(\sqrt{2L}\delta_t + \sqrt{G(\theta_0)}). \tag{22}$$

By Gronwall's inequality, we obtain

$$\delta_t \leq \sqrt{\frac{G(\theta_0)}{2L}}(e^{2^{2-\frac{d}{2}}\sqrt{L}t} - 1).$$

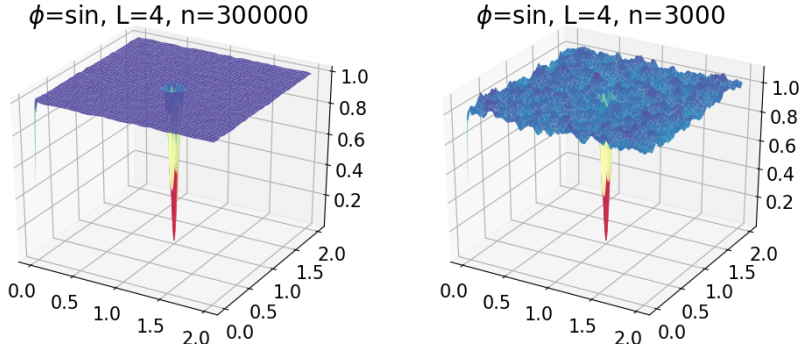
Numerical evidence

Consider learning parity functions with online SGD. Fig. 1 shows the convergence of SGD. Here, the model is two-layer neural networks with width being 2000. The hinge loss $\ell(y, y') = \max(0, 1 - yy')$ is used, batch size is 2000 and learning rate is 0.002. We see clearly that when $d = 20$, the training process does not show any improvement in a reasonable time.

Figure 1: Learning party functions with SGD and two-layer neural networks.

An illustration of the landscape of sine neuron

Consider a two-dimensional case. $R(\boldsymbol{w}) = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}[(\sin(L2\pi\boldsymbol{w}^T\boldsymbol{x}) - \sin(L2\pi\boldsymbol{w}^T\boldsymbol{x}))^2]$, where L can be viewed as a proxy of the dimension d.



- For the population landscape, the global minima locate in a deep well with other place is extremely flat. This confirms Theorem 5.
- The empirical landscape is full of bad local minima.

Summary

- Learning a subset of two-layer neural networks, whose path norms are bounded by $\mathrm{poly}(d)$, can be reduced to certain classical hard problems, whose hardness is assumed to be true. Otherwise, the modern cryptosystem can be broken in polynomial time. This type of hardness results hold for any algorithms.

Summary

- Learning a subset of two-layer neural networks, whose path norms are bounded by $\mathrm{poly}(d)$, can be reduced to certain classical hard problems, whose hardness is assumed to be true. Otherwise, the modern cryptosystem can be broken in polynomial time. This type of hardness results hold for any algorithms.
- For orthonormal classes, we show that the variance (wrt the target function) of gradients is exponentially small. Hence, *gradient-based* algorithms are unlikely to succeed. This observation hold for any parametric model as long as it is satisfies some smooth condition.

Summary

- Learning a subset of two-layer neural networks, whose path norms are bounded by $\mathrm{poly}(d)$, can be reduced to certain classical hard problems, whose hardness is assumed to be true. Otherwise, the modern cryptosystem can be broken in polynomial time. This type of hardness results hold for any algorithms.
- For orthonormal classes, we show that the variance (wrt the target function) of gradients is exponentially small. Hence, *gradient-based* algorithms are unlikely to succeed. This observation hold for any parametric model as long as it is satisfies some smooth condition.
- Typical examples include the parity function and the cosine neuron: $f(\boldsymbol{x}) = \cos(\boldsymbol{w}^T \boldsymbol{x})$. These functions can be represented as two-layer ReLU networks with the path norm bounded by Cd^2.

Summary

- Learning a subset of two-layer neural networks, whose path norms are bounded by $\mathrm{poly}(d)$, can be reduced to certain classical hard problems, whose hardness is assumed to be true. Otherwise, the modern cryptosystem can be broken in polynomial time. This type of hardness results hold for any algorithms.
- For orthonormal classes, we show that the variance (wrt the target function) of gradients is exponentially small. Hence, *gradient-based* algorithms are unlikely to succeed. This observation hold for any parametric model as long as it is satisfies some smooth condition.
- Typical examples include the parity function and the cosine neuron: $f(\boldsymbol{x}) = \cos(\boldsymbol{w}^T \boldsymbol{x})$. These functions can be represented as two-layer ReLU networks with the path norm bounded by Cd^2.
- These hardness results suggest that the Barron space is very likely too large to study the training of two-layer neural networks.