Fine-Grained Analysis of Optimization and Generalization for Overparameterized Two-Layer NNs

-Summary-

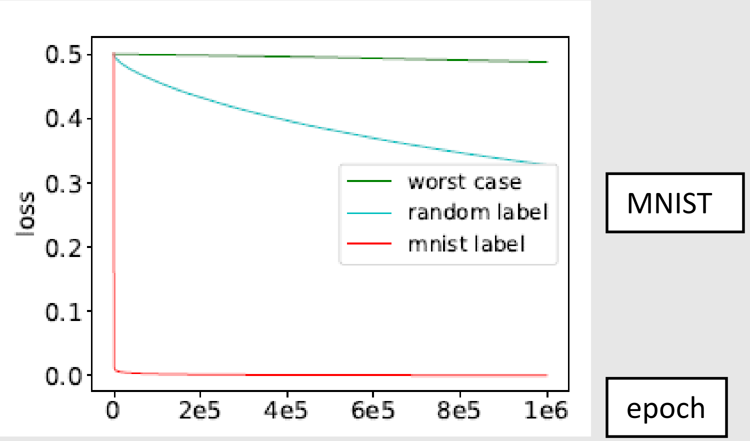
Introduction

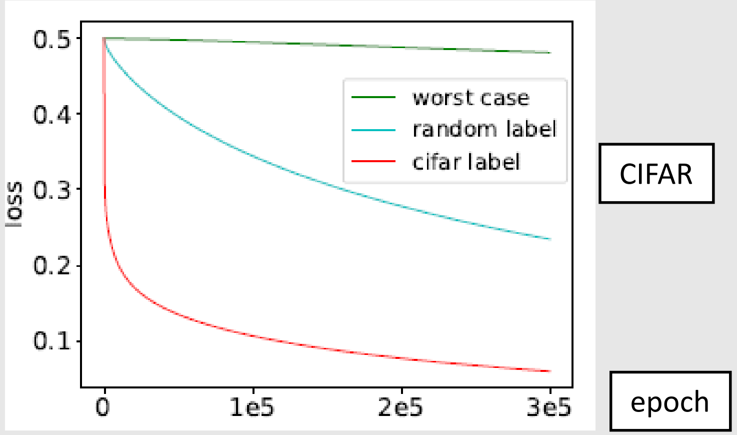
Question

Question 1 :

Why do true labels give faster convergence rate than random labels for gradient descent?

- (Zhang, 2017) : Sufficiently powerful nets(vastly more parameters than number of training samples) can attain zero training error, regardless of properly labeled or randomly labeled data.
- But the convergence rate on randomly labeled data is much slower than properly labeled data.





Question 2 :

Is there an easily verifiable complexity measure that can differentiate true labels and random labels?

- Classical measure : VC-dimension / Rademacher complexity => too pessimistic or weak

  Also, rely on some results of the trained net revealed/computed at the end of training.

Introduction

Notation and setting

Setting : (Two-layer NN trained by randomly initialized GD)

- $x \in \mathbb{R}^d$: input, $y \in \mathbb{R}$: label / $\mathbb{x} = (x_1, \dots, x_n) \in \mathbb{R}^{d \times n}$, $\mathbb{y} = (y_1, \dots, y_n)^T \in \mathbb{R}^n$
- Assume $\|\mathbb{x}\|_2 = 1$ and $|y| \leq 1$
- $S = \{(x_i, y_i)\}_{i=1}^n$: n input- label samples
- $w_1, \dots, w_m \in \mathbb{R}^d$: weights in 1st layer / $\mathbb{W} = (w_1, \dots, w_m) \in \mathbb{R}^{d \times m}$
- $a_1, \dots, a_m \in \mathbb{R}$: weights in 2nd layer / $\mathbb{a} = (a_1, \dots, a_m)^T \in \mathbb{R}^m$
- Output : $f_{\mathbb{W},\mathbb{a}}(\mathbb{x}_i) = u_i = \frac{1}{\sqrt{m}} \sum_{r=1}^m a_r \sigma(w_r^T \mathbb{x}_i)$ / $\mathbb{u} = (u_1, \dots, u_n)^T \in \mathbb{R}^n$
- Loss function : $\Phi(\mathbb{w}) = \frac{1}{2} \sum_{i=1}^n (y_i - f_{\mathbb{W},\mathbb{a}}(\mathbb{x}_i))^2 = \frac{1}{2} \|\mathbb{y} - \mathbb{u}\|_2^2$
- Random initialization : $\mathbb{w}_r(0) \sim N\left(0, \kappa^2 I\right)$, $a_r \sim unif(\{-1,1\})$ for $r \in \{1, \dots m\}$ / note: $\kappa \in (0,1)$
- GD update rule :

$$\mathbb{w}_r(k+1) - \mathbb{w}_r(k) = -\eta \nabla_{\mathbb{w}(k)} \Phi\left(\mathbb{w}(k)\right) = -\eta \frac{a_r}{\sqrt{m}} \sum_{i=1}^n \left(f_{\mathbb{W},\mathbb{a}}(\mathbb{x}_i) - y_i\right) \mathbb{I}\left(w_r(k)^T \mathbb{x}_i \geq 0\right\} \mathbb{x}_i$$

Preliminaries

Definition : Gram matrix

Given $\{\mathbb{x}_i\}_{i=1}^{n}$, we define the following Gram matrix $H^{\infty} \in \mathbb{R}^{n\times n}$ as follows :

$$H_{\mathrm{ij}}^{\infty} = \mathbb{E}_{\mathbb{w}\sim N(0,I)}\left[\mathbb{x}_i^T\mathbb{x}_j\mathbb{I}\left\{\mathbb{w}^T\mathbb{x}_i \geq 0, \mathbb{w}^T\mathbb{x}_j \geq 0\right\}\right] = \frac{\mathbb{x}_i^T\mathbb{x}_j(\pi - \arccos\left(\mathbb{x}_i^T\mathbb{x}_j\right))}{2\pi} \text{ for any } i,j \in \{1,\ldots n\}$$

Note : In two-layer ReLU network, if $H^{\infty}$is positive definite, GD converges to 0 training loss for sufficiently large m (number of neurons on 1st layer) [theorem 3.1]

Theorem 3.1 (Du et al, 2018c)

Assume $\lambda_0 = \lambda_{min}(H^{\infty}) > 0$. For $\delta \in (0,1)$, if $m = \Omega(\frac{n^6}{\lambda_0^4\kappa^2\delta^3})$ and $\eta = O(\frac{\lambda_0}{n^2})$, then with probability at least $1-\delta$ over the random initialization of GD, we have :

1. $\Phi\left(\mathbb{W}(0)\right) = O(\frac{n}{\delta})$
2. $\Phi\left(\mathbb{W}(k+1)\right) \leq \left(1-\frac{\eta\lambda_0}{2}\right)\Phi\left(\mathbb{W}(k)\right)$ for any $k \geq 0$

Results – Analysis of convergence rate

Observation

When the size of initialization κ is small and the network width size m is large, the sequence $\{\mathbb{u}(k)\}_{k=0}^{\infty}$ stays close to another sequence $\{\tilde{\mathbb{u}}(k)\}_{k=0}^{\infty}$, which has a linear update rule [theorem 4.1] :

- $\tilde{\mathbb{u}}(0) = 0$
- $\tilde{\mathbb{u}}(k+1) = \tilde{\mathbb{u}}(k) - \eta H^{\infty}(\tilde{\mathbb{u}}(k) - \mathbb{y})$

Theorem 4.1

Suppose $\lambda_0 = \lambda_{min}(H^{\infty}) > 0\ (positive\ definite)$, $\kappa = O\left(\frac{\epsilon\delta}{\sqrt{n}}\right)$, $m = \Omega(\frac{n^7}{\lambda_0^4\kappa^2\delta^4\epsilon^2})$ and $\eta = O(\frac{\lambda_0}{n^2})$. Then with probability at least $1-\delta$ over the random initialization, for all $k = 0,1,2,\ldots$, we have :

$$\|\mathbb{y} - \mathbb{u}(k)\|_2 = \sqrt{\sum_{i=1}^{n}(1-\eta\lambda_i)^{2k}\left(\mathbb{v}_i^T\mathbb{y}\right)^2} \pm O\left(\frac{\sqrt{n}k}{\delta} + \frac{n^{\frac{7}{2}}}{\sqrt{m}\lambda_0^2\kappa\delta^2}\right) = \underline{\|\mathbb{y} - \tilde{\mathbb{u}}(k)\|_2} \pm O(\frac{\sqrt{n}k}{\delta} + \frac{n^{\frac{7}{2}}}{\sqrt{m}\lambda_0^2\kappa\delta^2})$$

dominating term (indeed?)

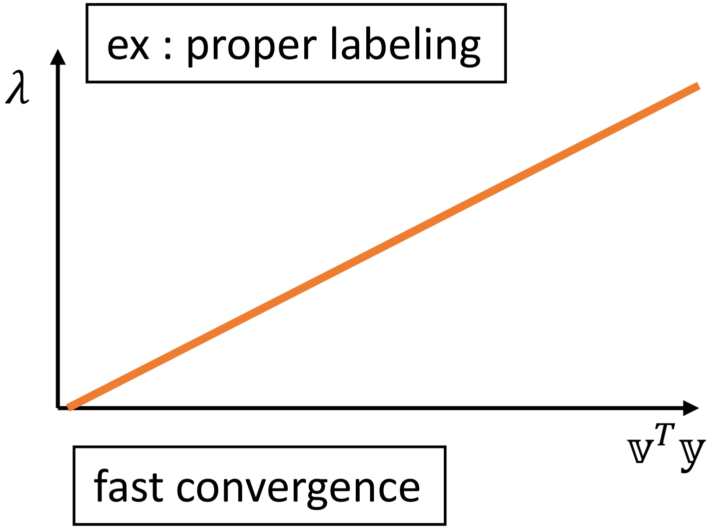
Note : $H^{\infty} = \sum_{i=1}^{n}\lambda_i\mathbb{v}_i\mathbb{v}_i^T$ by spectral decomposition theorem.

Results – Analysis of convergence rate

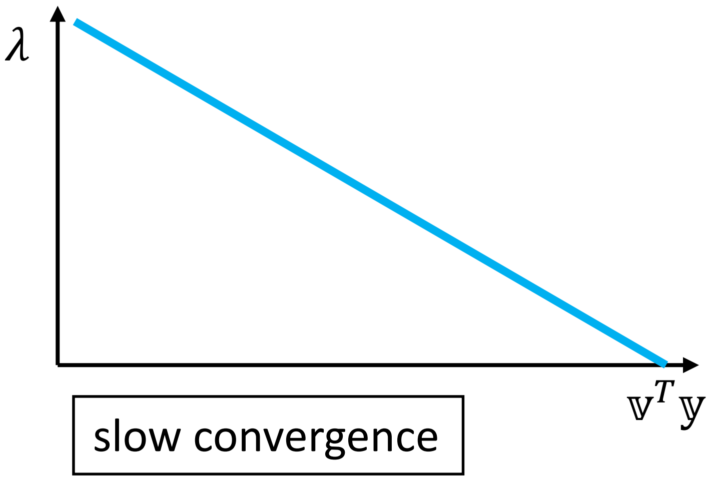
Analysis

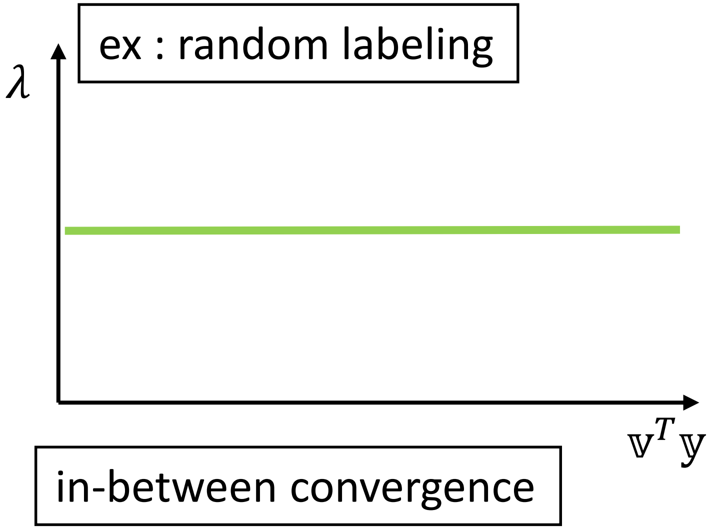
- To answer the question 1, it suffices to understand how fast $\sum_{i=1}^{n}(1-\eta\lambda_i)^{2k}\left(\mathbb{v}_i^T\mathbb{y}\right)^2$[dominating term] converges to 0 as k (=epoch) grows
- Since $\xi_i(k) = \ (1-\eta\lambda_i)^{2k}\left(\mathbb{v}_i^T\mathbb{y}\right)^2$ is a geometric sequence with $\xi_i(0) = \ \left(\mathbb{v}_i^T\mathbb{y}\right)^2$, rate = $(1-\eta\lambda_i)^2$. The larger λ_i is , the faster $\{\xi_i(k)\}_{k=0}^{\infty}$ decreases to 0

  => To have faster convergence, it would be good if the projection of $\mathbb{y}$ onto top eigenvectors to be larger. (Answer to question 1)


ex : proper labeling

fast convergence

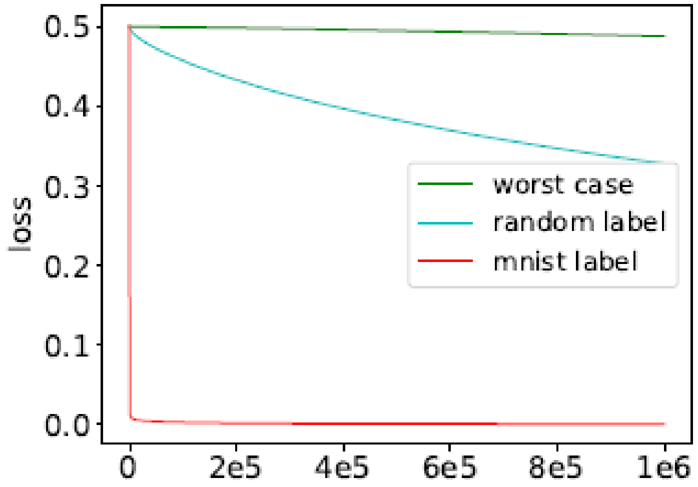

slow convergence

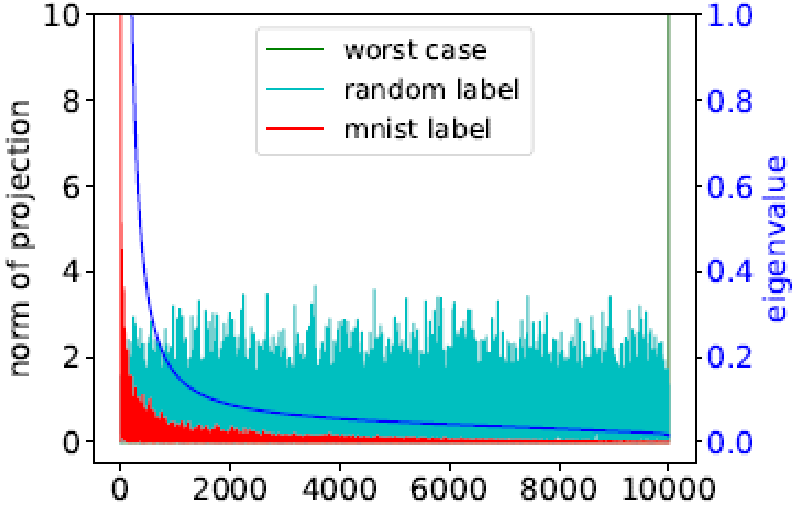

ex : random labeling

in-between convergence

Results – Analysis of convergence rate

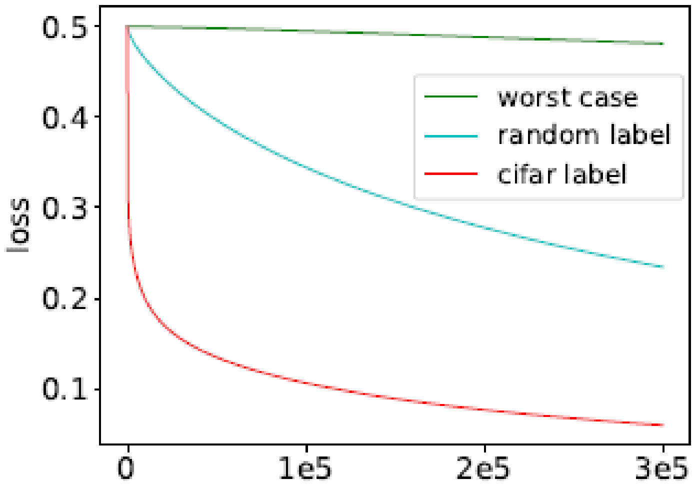
MNIST

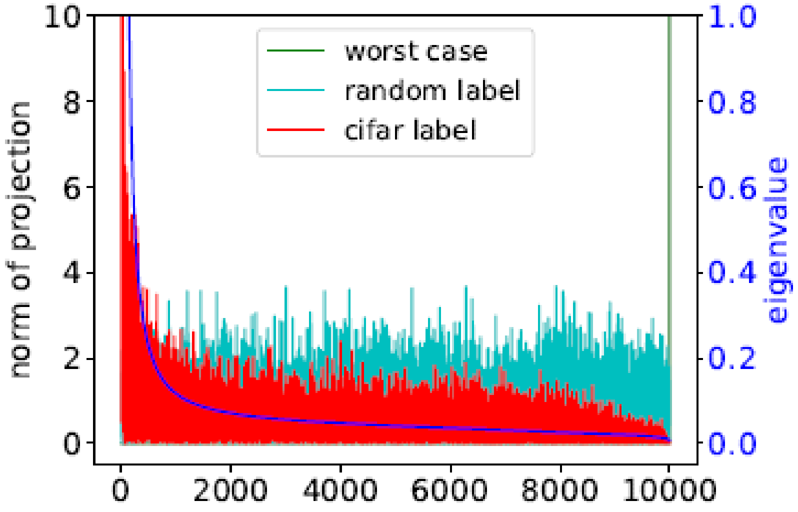




Question:
Why true labels align well with top eigenvectors of H^{∞}?

CIFAR





epoch

index i

Results – Analysis of Generalization

Definition : non-degenerate distribution

A distribution $\mathcal{D}$ over $\mathbb{R}^d \times \mathbb{R}$ is (λ_0, δ, n)-non-degenerate, if for $n\ i.i.d.$ samples $\{(\mathbb{x}_i, y_i)\}_{i=1}^{n}$ from $\mathcal{D}$, with probability at least $1-\delta$, we have $\lambda_{min}(H^{\infty}) \geq \lambda_0 > 0$

Remark

- (Du et al., 2018c) : As long as no two $\mathbb{x}_i$ and $\mathbb{x}_j$ are parallel to each other, we have $\lambda_{min}(H^{\infty}) > 0$.
- For most real-world distributions, any two training inputs are not parallel.

Results – Analysis of Generalization

Theorem 5.1

Fix a failure probability $\delta \in (0,1)$. Suppose our data $S = \{(\mathbb{x}_i, y_i)\}_{i=1}^{n}$ are $i.i.d.$ samples from a $\left(\lambda_0, \frac{\delta}{3}, n\right)$-non-degenerate distribution $\mathcal{D}$, and $\kappa = O(\frac{\lambda_0 \delta}{n})$, $m \geq k^{-2} poly(n, \lambda_0^{-1}, \delta^{-1})$. Consider any loss function $l : \mathbb{R} \times \mathbb{R} \rightarrow [0,1]$ that is 1-Lipschitz in the first argument such that $l(y, y) = 0$.

Then with probability at least $1 - \delta$ over the random initialization GD and the training samples, the two-layer NN $f_{\mathbb{W}(k),\mathbb{a}}$ trained by GD for $k \geq \Omega(\frac{\log\left(\frac{n}{\delta}\right)}{\eta \lambda_0})$ epochs has population loss $L_{\mathcal{D}}\left(f_{\mathbb{W}(k),\mathbb{a}}\right) = \mathbb{E}_{(\mathbb{x},y)\sim\mathcal{D}}[l\left(f_{\mathbb{W}(k),\mathbb{a}}(\mathbb{x}), y\right)]$ bounded as :

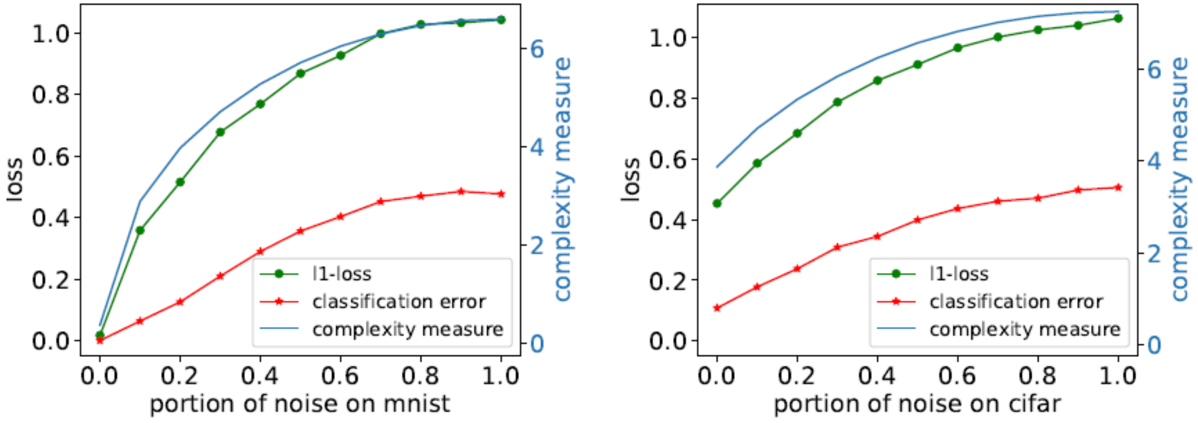
$$L_{\mathcal{D}}\left(f_{\mathbb{W}(k),\mathbb{a}}\right) \leq \overset{\boxed{\text{dominating term}}}{\sqrt{\frac{2\mathbb{y}^T (H^\infty)^{-1} \mathbb{y}}{n}}} + O\left(\sqrt{\frac{\log(\frac{n}{\lambda_0 \delta})}{n}}\right)$$

Results – Analysis of Generalization

Analysis

- The dominating term $\sqrt{\frac{2\mathbb{y}^T(H^\infty)^{-1}\mathbb{y}}{n}}$ can be viewed as a complexity measure of data, which can be used to predict the test accuracy of the learned NN. (Answer to question 2)
- Advantages of this measure :

1. Directly computed from given data $\{(\mathbb{x}_i, y_i)\}_{i=1}^n$ (without the need of training of NN)
2. Independent of the network width m ($\because H^\infty$ is independent on m)



(a) MNIST Data.

(b) CIFAR Data.

complexity measure follows tendency of test error

Results – Provable learning using Two-Layer ReLU NN

What type of functions can be learned by using the new complexity measure?

Theorem 6.1 [Two-layer NN with polynomial activation]

Suppose we have $y_i = g(\mathbb{x}_i) = \sum_j \alpha_j \left(\beta_j^T \mathbb{x}_i\right)^{p_j}$ for any $i \in \{1, \dots, n\}$,where for each j, $p_j = 1$ or $p_j = 2l$ ($l \in \mathbb{N}$), $\beta_j \in \mathbb{R}^d$ and $\alpha_j \in \mathbb{R}$. Then we have

$$\sqrt{\mathbb{y}^T (H^\infty)^{-1} \mathbb{y}} \leq 3 \sum_j p_j \left|\alpha_j\right| \cdot \left\|\beta_j\right\|_2^{p_j}$$

Results – Provable learning using Two-Layer ReLU NN

Learnable function examples

- [Linear functions] : $g(\mathbb{x}) = \beta^T \mathbb{x}$ ---> $\sqrt{\mathbb{y}^T (H^\infty)^{-1} \mathbb{y}} \leq 3\|\beta\|_2$
- [Quadratic functions] : $g(\mathbb{x}) = \mathbb{x}^T A \mathbb{x}$, where $A \in \mathbb{R}^{d\times d}$ is symmetric and $A = \sum_{j=1}^{d} \alpha_j \beta_j \beta_j^T \ \ (spectral - decomposition)$,

  then $g(\mathbb{x}) = \sum_{j=1}^{d} \alpha_j \left(\beta_j^T \mathbb{x}\right)^2$ ---> $\sqrt{\mathbb{y}^T (H^\infty)^{-1} \mathbb{y}} \leq 6 \sum_j \left|\alpha_j\right| = O(\|A\|_*)$ [trace norm]
- [Cosine activation] : $g(\mathbb{x}) = \cos\left(\beta^T \mathbb{x}\right) - 1$ Then,

  (using taylor series) $g(\mathbb{x}) = \sum_{j=1}^{\infty} \frac{(-1)^j \left(\beta^T \mathbb{x}\right)^{2j}}{(2j)!}$ ---> $\sqrt{\mathbb{y}^T (H^\infty)^{-1} \mathbb{y}} \leq O\left(\sum_{j=1}^{\infty} \frac{j}{(2j)!} \|\beta\|_2^{2j}\right) = \mathrm{O}(\|\beta\|_2 \cdot \sinh(\|\beta\|_2))$

Note : Broad class of functions can be approximated using taylor series, which have forms $g(\mathbb{x}_i) = \sum_j \alpha_j \left(\beta_j^T \mathbb{x}_i\right)^{p_j}$

Thus, we can probably guarantee the learning of those functions using theorem 6.1/5.1