Rademacher Complexity in *Simplex/*$l_\infty$ Set

Yuwei Lu
University of Warwick, Coventry, UK.
u2033107@live.warwick.ac.uk

**Abstract.** When the size of the neural network is too large, calculating the bound of neural network is a difficult problem. Therefore, "size-independent" is what needs to look for here. This paper follows the path of "Size- Independent Sample Complexity of Neural Network ", and tries to get a better expression of Rademacher Complexity of neural networks.

1. **Introduction**
Existing learning theories include VC dimension [1], Rademacher complexity [2] and PAC Bayes boundary [3]. Since Rademacher Complexity can consider data distribution to a certain extent, it is mainstream and popular. This paper finds a way to bound the Rademacher Complexity of neural network in *Simplex/*$l_\infty$ set.

In the learning theory, the generalization ability of a model is established according to the complexity (capacity) of hypothesis class H, where H is a function space. Simply, the "capacity" of a class is put, depending on the number of data sets that can fit well on the class. The larger the size of the class is, the more flexible the class is, but the easier it is to overfit. Learning theory presents that this must be due to some inductive bias, which constrains one to learn machines of specific configurations (either explicitly, e.g., via regularization, or implicitly, via the algorithm used to train them). However, understanding the nature of this inductive bias is still largely an open problem.

In the paper "Size-Independent Sample Complexity of Neural Networks [4]", the authors have figured out the generalization error in network with parameter matrices of Frobenius norm as well as in the $l_1/l_\infty$ setup. However, it is not the general solution to size-dependence. This paper calculates inductive bias via Rademacher complexity and reduces its bound from exponential to polynomial depth dependence. And in order to improve and perfect the algorithm, one of the solutions is found in the set of *Simplex/*$l_\infty$.

Section 2 introduces some fundamental knowledge of neural networks and Rademacher Complexity. In section 3.1, in order to let readers understand the purpose of this paper well, firstly, the process of calculating the complexity of neural networks is shown directly. This section also indicates how much the bound depends on the size of neural networks, for example, its depth and width. Section 3.2 generally introduces how to turn exponential dependent into polynomial in two different sets in [4]. And the result in simplex/*$l_\infty$* set discovered in this paper, inspired by the two previous results, is shown in 3.3, and it will be shown in detail.

2. **Neural Network and Rademacher Complexity**

**Definition 2.1. Neural Network**

Given the domain $X = \{x: ||x|| \leq B\}$ in Euclidean space, we consider (scalar or vector-valued) standard neural network of the form:

$$x \rightarrow W_d\sigma_d - 1(W_d - 1\sigma_d - 2(\ldots \sigma_1(W_1x))))$$  \hspace{1cm} (1)

---

Content from this work may be used under the terms of the Creative Commons Attribution 3.0 licence. Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

Published under licence by IOP Publishing Ltd
where $W_j$ is a parameter and $\sigma_j$ is the activation function, such as $\text{sigmoid}$, $\tanh$, denote $\sigma_j(0) = 0$.

Besides, we use $d$ as the depth of network (the maximal row or column dimensions of $W_1, \ldots, W_d$).

**Definition 2.2. Rademacher Complexity**

Given a real-valued function class $\mathcal{H}$. And $X = \{x_1, \ldots, x_m\}$ is the sample set with a capacity of $m$ from the space of $X$. Then Rademacher complexity $\bar{R}_m(\mathcal{H})$ as:

$$\bar{R}_m(\mathcal{H}) = \mathbb{E}_\varepsilon \left[ \sup_{h \in \mathcal{H}} \frac{1}{m} \sum_{i=1}^{m} \varepsilon_i h(x_i) \right]$$

where $\varepsilon = (\varepsilon_1, \ldots, \varepsilon_m)$ is a vector uniformly distributed in $\{-1, +1\}^m$. And $\varepsilon_i$ is a random variable, obeying the average distribution (50%). The value is $-1$ or $+1$, also known as the Rademacher variable.

3. The Bound of Neural Network

3.1 Directly Bound

Start from a simple "size-independent" example. In the class of linear predictors $(x \rightarrow w^T x)$, we can assume that $||w|| \leq M$ (where $\cdot$ signifies Euclidean norm) and the distribution $||x|| \leq B$, then we can get the generalization error as $O(MB/\sqrt{m})$, which given $m$ scales, which is totally independent of the size of network. However, when it comes to more general case, "size-independent" is not that easy to realize.

Continuing with the definition of Rademacher complexity, we can get $\bar{R}_m(\mathcal{H})$ as:

$$\bar{R}_m(\mathcal{H}) = \mathbb{E}_\varepsilon \left[ \sup_{h \in \mathcal{H}} \frac{1}{m} \sum_{i=1}^{m} \varepsilon_i h(x_i) \right]$$

$$= \mathbb{E}_\varepsilon \left[ \sup_{h \in \mathcal{H}_{d-1}} \sup_{W_d \in F} \frac{1}{m} \sum_{i=1}^{m} \varepsilon_i W_d \sigma(h(x_i)) \right]$$

$$\leq M_F(d) \mathbb{E}_\varepsilon \left[ \sup_{h \in \mathcal{H}_{d-1}} \left\| \frac{1}{m} \sum_{i=1}^{m} \varepsilon_i \sigma(h(x_i)) \right\| \right]$$

$$\leq 2M_F(d) \mathbb{E}_\varepsilon \left[ \sup_{h \in \mathcal{H}_{d-1}} \left\| \frac{1}{m} \sum_{i=1}^{m} \varepsilon_i h(x_i) \right\| \right]$$

$$\leq \cdots \leq \frac{2^d [\prod_{i=1}^{d} M_F(i)]}{\sqrt{m}}$$

Therefore, the generalization error scales as $O\left(\frac{2^d [\prod_{i=1}^{d} M_F(i)]}{\sqrt{m}}\right)$. And $2^d$ here is unavoidable. Therefore, we can try exponential depth as another way.

3.2 From Exponential to Polynomial Depth Dependent

We can rewrite Rademacher complexity $m\bar{R}_m(\mathcal{H}) = \mathbb{E}_\varepsilon [\sup_{h \in \mathcal{H}} \sum_{i=1}^{m} \varepsilon_i h(x_i)]$ as:

$$\frac{1}{\lambda} \log \exp(\lambda \mathbb{E}_\varepsilon [\sup_{h \in \mathcal{H}} \sum_{i=1}^{m} \varepsilon_i h(x_i)]) \leq \frac{1}{\lambda} \log(\mathbb{E}_\varepsilon [\sup_{h \in \mathcal{H}} \exp(\lambda \sum_{i=1}^{m} \varepsilon_i h(x_i))])$$

Running through this idea, we can avoid computing $2^d$ [6].

Then we just need to consider the bound of $\mathbb{E}_\varepsilon [\sup_{h \in \mathcal{H}} \exp(\lambda \sum_{i=1}^{m} \varepsilon_i h(x_i))]$. There are two ways to set the bound for Rademacher complexity in different two conditions:

1. Let $\sigma$ be a 1-Lipschitz positive-homogeneous activation function which is element-wise. Then for any class of vector-valued functions $F$, and any convex and monotonically increasing function $g : \mathbb{R} \rightarrow [0, \infty)$:
\[ \mathbb{E}[\sup_{f \in \mathcal{F}, W} |g(\| \sum_{i=1}^{m} \epsilon_i \sigma(Wf(x_i)) \|) - 2\| \sum_{i=1}^{m} \epsilon_i f(x_i) \|)] \leq 2\mathbb{E}[\sup_{f \in \mathcal{F}, g} (R \| \sum_{i=1}^{m} \epsilon_i f(x_i) \|)] \] (5)

- **Theorem 3.2.1.** Let \( \mathcal{H}_d \) be the class of real-valued networks of depth \( d \) over the domain \( X \), where each parameter matrix \( W_j \) has Frobenius norm at most \( M_j \), and with activation functions satisfy the above condition. Then

\[ \bar{R}_m(\mathcal{H}_d) \leq \frac{1}{m} \prod_{j=1}^{d} M_j \left( \sqrt{2 \log(2) d} + 1 \right) \sqrt{\sum_{i=1}^{m} \| x_i \|_2^2} \leq \frac{\left( j \right)^{\frac{1}{2}} \log(2d+1) \prod_{j=1}^{d} M_j(j)}{\sqrt{m}} \] (6)

(For the simplicity of the paper, we have omitted the proof process.) Therefore, we can conclude the bound as \( O\left( \frac{\sqrt{d} \prod_{j=1}^{d} M_j(j)}{\sqrt{m}} \right) \).

(2) Let \( \sigma \) be a 1-Lipschitz positive-homogeneous activation function with \( \sigma(0) = 0 \), applied element-wise. Then for any vector-valued class \( \mathcal{F} \), and convex and monotonically increasing function \( g : \mathbb{R} \to [0, \infty) \):

\[ \mathbb{E}[\sup_{f \in \mathcal{F}, W} |g(\| \sum_{i=1}^{m} \epsilon_i \sigma(Wf(x_i)) \| \infty) - 2\| \sum_{i=1}^{m} \epsilon_i f(x_i) \|_\infty)] \leq 2\mathbb{E}[\sup_{f \in \mathcal{F}, g} (R \| \sum_{i=1}^{m} \epsilon_i f(x_i) \|_\infty)] \] (7)

where \( \| \cdot \|_{\infty} \) denotes the vector infinity norm.

- **Theorem 3.2.2.** Let \( \mathcal{H}_d \) be the class of real-valued networks of depth \( d \) over the domain \( X \), where \( \|W_j\|_{1,\infty} \leq M_j \) for all \( j \in \{1, \ldots, d\} \), and with activation functions satisfying the above condition. Then,

\[ \bar{R}_m(\mathcal{H}_d) \leq 2 \frac{2^d}{m} \prod_{j=1}^{d} M_j \left( \sqrt{d+1 + \log(n)} \right) \left[ \max_{j \in \{1, \ldots, n\}} \sum_{i=1}^{m} x_{i,j}^2 \right] \]

\[ \leq \frac{2^d \sqrt{d+1 + \log(n)} \prod_{j=1}^{d} M_j(j)}{\sqrt{m}} \] (8)

where \( x_{i,j} \) is the \( j \)-th coordinate of the vector \( x_i \).

Therefore, we can conclude the bound as \( O\left( \frac{2^d \sqrt{d+1 + \log(n)} \prod_{j=1}^{d} M_j(j)}{\sqrt{m}} \right) \) [4].

### 3.3 In Simplex/l\( \infty \) set

**Lemma 3.3.1.** Let \( \sigma \) be a 1-Lipschitz positive-homogeneous activation function with \( \sigma(0) = 0 \), applied element-wise. Then for any vector-valued class \( F \), and convex and monotonically increasing function \( g : \mathbb{R} \to [0, \infty) \):

\[ \mathbb{E}[\sup_{f \in \mathcal{F}, W} |g(\| \sum_{i=1}^{m} \epsilon_i \sigma(Wf(x_i)) \| \infty) - 2\| \sum_{i=1}^{m} \epsilon_i f(x_i) \|_\infty)] \leq 2\mathbb{E}[\sup_{f \in \mathcal{F}, g} (R \| \sum_{i=1}^{m} \epsilon_i f(x_i) \|_\infty)] \] (9)

where \( \| \cdot \|_\infty \) denotes the vector infinity norm.

Proof. Let \( W_j \) denote the \( j \)-th row of a matrix \( W \), we have

\[ \sup_{f \in \mathcal{F}, W} |g(\| \sum_{i=1}^{m} \epsilon_i \sigma(Wf(x_i)) \| \infty) = \sup_{f \in \mathcal{F}, W} \| \sum_{i=1}^{m} \epsilon_i \sigma(Wf(x_i)) \| \infty) \]

\[ = \sup_{f \in \mathcal{F}, W} \max_{k} g\left( \sum_{i=1}^{m} \epsilon_i \sigma(w_kf(x_i)) \right) \]

\[ = \sup_{f \in \mathcal{F}, W} \| \sum_{i=1}^{m} \epsilon_i \sigma(Wf(x_i)) \| \infty) \] (10)
Since \( g(|z|) \leq g(z) + g(-z) \),

\[
\mathbb{E}_z \left[ \sup_{f \in \mathcal{F}, W : \|W\|_1 \leq R} \left( \| \sum_{i=1}^{m} \varepsilon_i \sigma(W f(x_i)) \|_\infty \right) \right] \leq \mathbb{E}_z \left[ \sup_{f \in \mathcal{F}, W : \|W\|_1 \leq R} \left( \sum_{i=1}^{m} \varepsilon_i \sigma(w^i f(x_i)) \right) \right]
\]
\[
+ \mathbb{E}_z \left[ \sup_{f \in \mathcal{F}, W : \|W\|_1 \leq R} \left( - \sum_{i=1}^{m} \varepsilon_i \sigma(w^i f(x_i)) \right) \right]
\]
\[
= 2 \mathbb{E}_z \left[ \sup_{f \in \mathcal{F}, W : \|W\|_1 \leq R} \left( \sum_{i=1}^{m} \varepsilon_i \sigma(w^i f(x_i)) \right) \right]
\]
\[
\leq 2 \mathbb{E}_z \left[ \sup_{f \in \mathcal{F}, W : \|W\|_1 \leq R} \left( \| W \| \sum_{i=1}^{m} \varepsilon_i f(x_i) \| \right) \right]
\]
\[
= 2 \mathbb{E}_z \left[ \sup_{f \in \mathcal{F}, g : \|g\| \leq R} \left( \sum_{i=1}^{m} \varepsilon_i f(x_i) \| \right) \right]
\]

(\( \varepsilon_i \) is symmetry distributed)

From this Lemma [6], we can get our theorem.

**Theorem 3.3.2** Let \( \mathcal{H}_d \) be the class of real-valued networks of depth \( d \) over the domain \( X \), where \( \|W\|_1 = 1 \), and with activation functions satisfying above Lemma, then

\[
\hat{R}_m(\mathcal{H}_d) \leq \frac{2}{m} \left( \sqrt{d + 1 + \log(n)} \right) \max_j \sum_{i=1}^{m} x_{ij}^2
\]
\[
\leq \frac{2R \sqrt{d+1 + \log(n)}}{\sqrt{m}}
\]

(12)

Proof. Fixing \( \lambda > 0 \), then we can upper bound the Rademacher complexity as

\[
m \hat{R}_m(\mathcal{H}) = \mathbb{E}_z \left[ \sup_{N_{w_{d-1}} : \|N_{w_{d-1}}\|_1 \leq R} \sum_{i=1}^{m} \varepsilon_i W_{d} \sigma_{d-1}(N_{w_{d-1}}(x_i)) \right]
\]
\[
\leq \frac{1}{\lambda} \log \left( \mathbb{E}_z \left[ \sup_{\lambda} \sum_{i=1}^{m} \varepsilon_i W_{d} \sigma_{d-1}(N_{w_{d-1}}(x_i)) \right] \right)
\]
\[
\leq \frac{1}{\lambda} \log \left( \mathbb{E}_z \left[ \sup_{\lambda} (M(d) \| \lambda \sum_{i=1}^{m} \varepsilon_i \sigma_{d-1}(N_{w_{d-1}}(x_i)) \|_\infty) \right] \right)
\]
\[
= \frac{1}{\lambda} \log \left( \mathbb{E}_z \left[ \sup_{f} (M(d) \| \lambda \sum_{i=1}^{m} \varepsilon_i \sigma_{d-1}(W_{d-1}(x_i)) \|_\infty) \right] \right)
\]
Letting $x_{i,j}$ denote the $j$-th coordinate of $x_i$, we can calculate the expectation as:

$$
\mathbb{E}_e \left[ \exp (\lambda \sum_{i=1}^{m} \varepsilon_i x_{i,j}) \right] = \mathbb{E}_e \left[ \exp (\lambda \max_j \sum_{i=1}^{m} \varepsilon_i x_{i,j}) \right]
$$

$$
\leq \sum_{j=1}^{n} \mathbb{E}_e \left[ \exp (\lambda \sum_{i=1}^{m} \varepsilon_i x_{i,j}) \right] 
$$

$$
\leq \sum_{j=1}^{n} \mathbb{E}_e \left[ \exp (\lambda \sum_{i=1}^{m} \varepsilon_i x_{i,j}) + \exp (-\lambda \sum_{i=1}^{m} \varepsilon_i x_{i,j}) \right] 
$$

$$
= 2 \sum_{j=1}^{n} \mathbb{E}_e \left[ \exp (\lambda \sum_{i=1}^{m} \varepsilon_i x_{i,j}) \right] 
$$

$$
= 2 \sum_{j=1}^{n} \prod_{i=1}^{m} \mathbb{E}_e \left[ \exp (\lambda \varepsilon_i x_{i,j}) \right] 
$$

$$
= 2 \sum_{j=1}^{n} \prod_{i=1}^{m} \exp (\lambda x_{i,j} + \exp (-\lambda x_{i,j})) \frac{2}{2} 
$$

$$
\leq 2 \sum_{j=1}^{n} \exp \left( \lambda^2 \sum_{i=1}^{m} x_{i,j}^2 \right) 
$$

$$
\leq 2n \max_j \exp (\lambda^2 \sum_{i=1}^{m} x_{i,j}^2) \quad (14) 
$$

Therefore, we get:

$$
\frac{1}{\lambda} \log \left( 2^d \mathbb{E}_e \left[ \exp (\lambda \sum_{i=1}^{m} \varepsilon_i x_{i,j}) \right] \right) \leq \frac{1}{\lambda} \log \left( 2^d 2n \max_j \exp (\lambda^2 \sum_{i=1}^{m} x_{i,j}^2) \right) 
$$

$$
= \frac{d+1+\log(n)}{\lambda} + \lambda \max_j \sum_{i=1}^{m} x_{i,j}^2 \quad (15) 
$$

Then we choose $\lambda = \frac{d+1+\log(n)}{\sqrt{\max_j \sum_{i=1}^{m} x_{i,j}^2}}$, such that:
\[
\hat{R}_m(\mathcal{H}_d) \leq \frac{2}{m} \left( \sqrt{d + 1 + \log(n)} \right) \sqrt{\max_j \sum_{i=1}^m x_{ij}^2}
\]

\[
\leq \frac{2B\sqrt{d + 1 + \log(n)}}{\sqrt{m}}
\]

From this result, we can have a guess of a more general case:
Let \( \mathcal{H}_d \) be the class of real-valued networks of depth \( d \) over the domain, where \( ||w||_1 = C \), and \( C \) is a constant, then:

\[
\hat{R}_m(\mathcal{H}_d) \leq \frac{2}{m} \left( \sqrt{d + 1 + \log(n)} \right) \sqrt{\max_j \sum_{i=1}^m x_{ij}^2}
\]

\[
\leq \frac{2Bc\sqrt{d+1+\log(n)}}{\sqrt{m}}
\]

4. Conclusion
Depending on Rademacher complexity, the capacity of neural networks can be better estimated. This paper firstly has a general understanding of neural networks and Rademacher complexity. Then the bound of Rademacher complexity is calculated directly. However, the answer is hard to calculate with \( 2^d \). Therefore, a log is used to translate exponential into polynomial depth dependent. Finally, in \( \text{Simple}/l_\infty \) set, we work out the bound in polynomial depth dependent level as \( \mathcal{O}\left(\frac{2Bc\sqrt{d+1+\log(n)}}{\sqrt{m}}\right) \).

References
[1] Martin Anthony and Peter L Bartlett. Neural network learning: Theoretical foundations. Cambridge university press, 2009.
[2] Yin, D., Kannan, R. and Bartlett, P., May. Rademacher complexity for adversarially robust generalization. In International Conference on Machine Learning (pp. 7085-7094). PMLR, 2019.
[3] Germain, P., Lacasse, A., Laviolette, F. and Marchand, M., June. PAC- Bayesian learning of linear classifiers. In Proceedings of the 26th Annual International Conference on Machine Learning (pp. 353-360), 2009.
[4] Golowich, N., Rakhlin, A. and Shamir, O., July. Size-independent sample complexity of neural networks. In Conference On Learning Theory (pp. 297-299). PMLR, 2018.
[5] Bartlett, P. L., Foster, D. J., & Telgarsky, M. J. Spectrally-normalized margin bounds for neural networks. In Advances in Neural Information Processing Systems (pp. 6240-6249), 2017.
[6] Michel Ledoux and Michel Talagrand.(1991) Probability in Banach Spaces. Springer,1991.
[7] Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. Norm-based capacity control in neural networks. In Conference on Learning Theory, pages 1376–1401, 2015.