ON RADEMACHER COMPLEXITY-BASED GENERALIZATION BOUNDS FOR DEEP LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

We show that the Rademacher complexity-based approach can generate non-vacuous generalisation bounds on Convolutional Neural Networks (CNNs) for classifying a small number of classes of images. The development of new Talagrand's contraction lemmas for high-dimensional mappings between vector spaces and CNNs for general Lipschitz activation functions is a key technical contribution. Our results show that the Rademacher complexity is independent of the network depth, assuming a bounded product of the norms of the weight matrices, for CNNs with common activations such as ReLU, Leaky ReLU, Parametric ReLU, Tanh, or any other odd functions.

1 INTRODUCTION

Deep models are typically heavily over-parametrized, while they still achieve good generalization performance. Despite the widespread use of neural networks in biotechnology, finance, health science, and business, just to name a selected few, the problem of understanding deep learning theoretically remains relatively under-explored. In 2002, Koltchinskii and Panchenko (Koltchinskii & Panchenko, 2002) proposed new probabilistic upper bounds on generalization error of the combination of many complex classifiers such as deep neural networks. These bounds were developed based on the general results of the theory of Gaussian, Rademacher, and empirical processes in terms of general functions of the margins, satisfying a Lipschitz condition. However, bounding Rademacher complexity for deep learning remains a challenging task. In this work, we provide some new upper bounds on Rademacher complexity in deep learning which does not explicitly depend on the length of deep neural networks. In addition, we show that Koltchinskii and Panchenko's approach can be improved to generate non-vacuous bounds for CNNs.

1.1 RELATED PAPERS

The complexity-based generalization bounds were established by traditional learning theory aiming to provide general theoretical guarantees for deep learning. (Goldberg & Jerrum, 1993), (Bartlett & Williamson, 1996), (Bartlett et al., 1998b) proposed upper bounds based on the VC dimension for DNNs. (Neyshabur et al., 2015) used Rademacher complexity to prove the bound with exponential dependence on the depth for ReLU networks. (Neyshabur et al., 2018) and (Bartlett et al., 2017) uses the PAC-Bayesian analysis and the covering number to obtain bounds with polynomial dependence on the depth, respectively. (Golowich et al., 2018) provided bounds with (sub-linear) square-root dependence on the depth for DNNs with positive-homogeneous activations such as ReLU.

The standard approach to develop generalization bounds on deep learning (and machine learning) was developed in seminar papers by (Vapnik, 1998), and it is based on bounding the difference between the generalization error and the training error. These bounds are expressed in terms of the so called VC-dimension of the class. However, these bounds are very loose when the VC-dimension of the class can be very large, or even infinite. In 1998, several authors (Bartlett et al., 1998a; Bartlett & Shawe-Taylor, 1999) suggested another class of upper bounds on generalization error that are expressed in terms of the empirical distribution of the margin of the predictor (the classifier). Later, Koltchinskii and Panchenko (Koltchinskii & Panchenko, 2002) proposed new probabilistic upper bounds on the generalization error of the combination of many complex classifiers such as deep neural networks. These bounds were developed based on the general results of the theory of Gaussian, Rademacher, and empirical processes in terms of general functions of the margins, satisfying a Lipschitz condition. They improved previously known bounds on generalization error of convex combination of classifiers. Generalization bounds for deep learning and kernel learning with Markov dataset based on Rademacher and Gaussian complexity functions have recently analysed in (Truong, 2022a). Analysis of machine learning algorithms for Markov and Hidden Markov datasets already appeared in research literature (Duchi et al., 2011; Wang et al., 2019; Truong, 2022c).

In the context of supervised classification, PAC-Bayesian bounds have been used to explain the generalisation capability of learning algorithms (Langford & Shawe-Taylor, 2003; McAllester, 2004; A. Ambroladze & ShaweTaylor, 2007). Several recent works have focused on gradient descent based PAC-Bayesian algorithms, aiming to minimise a generalisation bound for stochastic classifiers (Dziugaite & Roy., 2017; W. Zhou & Orbanz., 2019; Biggs & Guedj, 2021). Most of these studies use a surrogate loss to avoid dealing with the zero-gradient of the misclassification loss. Several authors used other methods to estimate of the misclassification error with a non-zero gradient by proposing new training algorithms to evaluate the optimal output distribution in PAC-Bayesian bounds analytically (McAllester, 1998; Clerico et al., 2021b;a). Recently, (Nagarajan & Kolter, 2019) showed that uniform convergence might be unable to explain generalisation in deep learning by creating some examples where the test error is bounded by δ but the (two-sided) uniform convergence on this set of classifiers will yield only a vacuous generalisation guarantee larger than $1-\delta$ for some $\delta \in (0,1)$. There have been some interesting works which use information-theoretic approach to find PAC-bounds on generalization errors for machine learning (Xu & Raginsky, 2017; Esposito et al., 2021) and deep learning (Jakubovitz et al., 2018).

In this work, we show that the Rademacher complexity does not explicitly depend on the length of CNNs which uses some special classes of activation functions σ such that $\sigma - \sigma(0)$ belongs to ReLU family $\mathcal{L} = \{\psi_\alpha : \psi_\alpha(x) = ReLU(x) - \alpha ReLU(-x),\ \forall x \in \mathbb{R}, \alpha \in [0,1]\}$, or odd function ones $\mathcal{O} = \{\psi : \psi(-x) = -\psi(x),\ \forall x \in \mathbb{R}\}$. Our result improves Golowich et al.'s bound (Golowich et al., 2018) where the authors showed that the Rademacher complexity is square-root dependent on the depth for DNNs with ReLU activation functions.

1.2 CONTRIBUTIONS

More specifically, our contributions are as follows:

- We develop new contraction lemmas for high-dimensional mappings between vector spaces which extends the Talagrand contraction lemma.
- We apply our new contraction lemmas to each layer of a CNN.
- We validate our new theoretical results experimentally on CNNs for MNIST image classifications, and our bounds are non-vacuous when the number of classes is small.

As far as we know, this is the first result which shows that the Rademacher complexity-based approach can lead to non-vacuous generalisation bounds on CNNs.

1.3 OTHER NOTATIONS

Vectors and matrices are in boldface. For any vector $\mathbf{x} = (x_1, x_2, \cdots, x_n) \in \mathbb{R}^n$ where $\mathbb{R}$ is the field of real numbers, its induced-L^p norm is defined as

$$\|\mathbf{x}\|_p = \left(\sum_{k=1}^{n} |x_k|^p\right)^{1/p}. \tag{1}$$

The j-th component of the vector $\mathbf{x}$ is denoted as $[\mathbf{x}]_j$ for all $j \in [n]$.

For $\mathbf{A} \in \mathbb{R}^{m \times n}$ where

$$\mathbf{A} = \begin{bmatrix} a_{11}, & a_{12}, & \cdots, & a_{1n} \\ a_{21}, & a_{22}, & \cdots, & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}, & a_{m2}, & \cdots, & a_{mn} \end{bmatrix} \tag{2}$$

we defined the induced-norm of matrix $\mathbf{A}$ as

$$\|\mathbf{A}\|_{p,q} = \sup_{\mathbf{x}\neq \underline{0}} \frac{\|\mathbf{A}\mathbf{x}\|_q}{\|\mathbf{x}\|_p}. \tag{3}$$

For abbreviation, we also use the following notation

$$\|A\|_p := \|A\|_{p,p}. \tag{4}$$

It is known that

$$\|\mathbf{A}\|_1 = \max_{1\leq j\leq n} \sum_{i=1}^{m} |a_{ij}|, \tag{5}$$

$$\|\mathbf{A}\|_2 = \sqrt{\lambda_{\max}(\mathbf{A}\mathbf{A}^T)}, \tag{6}$$

$$\|\mathbf{A}\|_\infty = \max_{1\leq i\leq m} \sum_{j=1}^{n} |a_{ij}|, \tag{7}$$

where $\lambda_{\max}(\mathbf{A}\mathbf{A}^T)$ is defined as the maximum eigenvalue of the matrix $\mathbf{A}\mathbf{A}^T$ (or the square of the maximum singular value of $\mathbf{A}$).

2 CONTRACTION LEMMAS IN HIGH DIMENSIONAL VECTOR SPACES

First, we recall the Talagrand's contraction lemma.

Lemma 1 *(Ledoux & Talagrand, 1991, Theorem 4.12) Let $\mathcal{H}$ be a hypothesis set of functions mapping from some set $\mathcal{X}$ to $\mathbb{R}$ and ψ be a μ-Lipschitz function from $\mathbb{R} \to \mathbb{R}$ for some $\mu > 0$. Then, for any sample S of n points $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n \in \mathcal{X}$, the following inequality holds:*

$$\mathbb{E}_{\boldsymbol{\varepsilon}}\left[\sup_{h\in\mathcal{H}}\left|\frac{1}{n}\sum_{i=1}^{n}\varepsilon_i(\psi\circ h)(\mathbf{x}_i)\right|\right] \leq 2\mu\mathbb{E}_{\boldsymbol{\varepsilon}}\left[\sup_{h\in\mathcal{H}}\left|\frac{1}{n}\sum_{i=1}^{n}\varepsilon_i h(\mathbf{x}_i)\right|\right], \tag{8}$$

where $\boldsymbol{\varepsilon} = (\varepsilon_1, \varepsilon_2, \cdots, \varepsilon_n)$.

In this section, we introduce some new versions of Talagrand's contraction lemma for the high-dimensional mapping ψ between vector spaces. The proof of the following theorem can be found in Supplementary Materials.

Theorem 2 *Let $\mathcal{H}$ be a set of functions mapping from some set $\mathcal{X}$ to $\mathbb{R}^m$ for some $m \in \mathbb{Z}_+$ and*

$$\mathcal{L} = \left\{\psi_\alpha : \psi_\alpha(x) = ReLU(x) - \alpha ReLU(-x) \ \ \forall x \in \mathbb{R}, \alpha \in [0,1]\right\} \tag{9}$$

where $ReLU(x) = \max(x, 0)$.

For any $\mu > 0$, let $\psi : \mathbb{R} \to \mathbb{R}$ be a μ-Lipschitz function. Define

$$\mathcal{H}_+ = \begin{cases} \mathcal{H} \cup \{-h : h \in \mathcal{H}\}, & \text{if } \ \psi - \psi(0) \ \text{ is odd} \\ \mathcal{H} \cup \{-h : h \in \mathcal{H}\} \cup \{|h| : h \in \mathcal{H}\}, & \text{if } \ \psi - \psi(0) \ \text{ others} \end{cases}. \tag{10}$$

Then, it holds that

$$\begin{aligned} &\mathbb{E}_{\boldsymbol{\varepsilon}}\left[\sup_{h\in\mathcal{H}}\left\|\frac{1}{n}\sum_{i=1}^{n}\varepsilon_i\psi(h(\mathbf{x}_i))\right\|_\infty\right] \\ &\qquad \leq \gamma(\mu)\mathbb{E}_{\boldsymbol{\varepsilon}}\left[\sup_{h\in\mathcal{H}_+}\left\|\frac{1}{n}\sum_{i=1}^{n}\varepsilon_i h(\mathbf{x}_i)\right\|_\infty\right] + \frac{1}{\sqrt{n}}|\psi(0)|, \end{aligned} \tag{11}$$

where

$$\gamma(\mu) = \begin{cases} \mu, & \text{if } \ \psi - \psi(0) \ \text{ is odd or belongs to } \mathcal{L} \\ 2\mu, & \text{if } \ \psi - \psi(0) \ \text{ is even} \\ 3\mu, & \text{if } \ \psi - \psi(0) \ \text{ others} \end{cases}. \tag{12}$$

Here, we define $\psi(\mathbf{x}) := (\psi(x_1), \psi(x_2), \cdots, \psi(x_m))^T$ for any $\mathbf{x} = (x_1, x_2, \cdots, x_m)^T \in \mathbb{R}^m$.

Remark 3 *Some remarks are in order.*

- *Identity, ReLU, Leaky ReLU, Parametric rectified linear unit (PReLU) belong to the class of functions $\mathcal{L}$.*
- *If ψ is odd or belongs to $\mathcal{L}$, then $\psi(0) = 0$. Therefore, Theorem 2 improves Lemma 1 in the special case where $m = 1$. This enhancement is achieved by leveraging the unique properties of certain function classes.*
- *Our results are based on a novel approach, which shows that tighter contraction lemmas can be obtained when both the class of functions $\mathcal{H}$ and the activation functions possess certain special properties. More specifically, in this work, we extend the class of functions $\mathcal{H}$ by adding more functions, resulting in a new class $\mathcal{H}_+$, which possesses certain special properties. Additionally, we restrict the class of activation functions to $\mathcal{L} \cup \{\psi : \mathbb{R} \to \mathbb{R} : \psi(x) - \psi(0) = -(\psi(-x) - \psi(0)),\ \forall x \in \mathbb{R}\}$.*

3 RADEMACHER COMPLEXITY BOUNDS FOR CONVOLUTIONAL NEURAL NETWORKS (CNNS)

3.1 CONVOLUTIONAL NEURAL NETWORK MODELS

Let $d_0, d_1, \cdots, d_L, d_{L+1}$ be a sequence of positive integer numbers such that $d_0 = d$ for some fixed $d \in \mathbb{Z}_+$. We define a class of function $\mathcal{F}$ as follows:

$$\mathcal{F} := \big\{f = f_L \circ f_{L-1} \circ \cdots \circ f_1 \circ f_0 : f_i \in \mathcal{G}_i \subset \{g_i : \mathbb{R}^{d_i} \to \mathbb{R}^{d_{i+1}}\}, \quad \forall i \in \{1, 2, \cdots, L\}\big\}, \tag{13}$$

where $f_0 : [0,1]^d \to \mathbb{R}^{d_1}$ is a fixed function and $d_{i+1} = M$ for some $M \in \mathbb{Z}_+$. A Convolutional Neural Network (CNN) with network-depth L is defined as a composition map $f \in \mathcal{F}$ where

$$f_i(\mathbf{x}) = \sigma_i(\mathbf{W}_i \mathbf{x}), \quad \forall \mathbf{x} \in \mathbb{R}^{d_i}. \tag{14}$$

Here, $\mathbf{W}_i \in \mathcal{W}_i$ where $\mathcal{W}_i$ is a set of matrices in $\mathbb{R}^{d_{i+1} \times d_i}$.

Given a function $f \in \mathcal{F}$, a function $g \in \mathbb{R}^M \times [M]$ predicts a label $y \in [M]$ for an example $\mathbf{x} \in \mathbb{R}^d$ if and only if

$$g(f(\mathbf{x}), y) > \max_{y' \neq y} g(f(\mathbf{x}), y') \tag{15}$$

where $g(f(\mathbf{x}), y) = \mathbf{w}_y^T f(\mathbf{x})$ with $\mathbf{w}_y = \underbrace{(0, 0, \cdots, 0, 1, 0, \cdots, 0)}_{\mathbf{w}_y(y)=1}$.

For a training set $\{\mathbf{x}_i\}_{i=1}^n$, the ∞-norm *Rademacher complexity* for the class function $\mathcal{F}$ is defined as

$$R_n(\mathcal{F}) := \mathbb{E}_{\boldsymbol{\varepsilon}}\left[\sup_{f \in \mathcal{F}} \left\| \frac{1}{n} \sum_{i=1}^n \varepsilon_i f(\mathbf{x}_i) \right\|_\infty \right] \qquad \forall k \in [L], \tag{16}$$

where $\{\varepsilon_i\}$ is a sequence of i.i.d. Rademacher (taking values $+1$ and -1 with probability $1/2$ each) random variables, independent of $\{\mathbf{x}_i\}$.

3.2 SOME CONTRACTION LEMMAS FOR CNNS

Based on Theorem 2, the following versions of Talagrand's contraction lemma for different layers of CNN are derived.

Definition 4 (Convolutional Layer with Average Pooling) *Let $\mathcal{G}$ be a class of μ-Lipschitz function σ from $\mathbb{R} \to \mathbb{R}$ such that $\sigma(0)$ is fixed. Let $C, Q \in \mathbb{Z}_+$, $\{r_l, \tau_l\}_{l \in [Q]}$ be two tuples of positive integer numbers, and $\{W_{l,c} \in \mathbb{R}^{r_l \times r_l}, c \in [C], l \in [Q]\}$ be a set of kernel matrices. A convolutional layer with average pooling, C input channels, and Q output channels is defined as a set of $Q \times C$ mappings $\Psi = \{\psi_{l,c}, l \in [Q], c \in [C]\}$ from $\mathbb{R}^{d \times d}$ to $\mathbb{R}^{\lceil (d - r_l + 1)/\tau_l \rceil \times \lceil (d - r_l + 1)/\tau_l \rceil}$ such that*

$$\psi_{l,c}(\mathbf{x}) = \sigma_{\text{avg}} \circ \sigma_{l,c}(\mathbf{x}), \tag{17}$$

where

$$\sigma_{\text{avg}}(\mathbf{x}) = \frac{1}{\tau_l^2}\Bigg(\sum_{k=1}^{\tau_l^2} x_k, \cdots, \sum_{k=(j-1)\tau_l^2+1}^{j\tau_l^2} x_k, \cdots, \sum_{k=\lceil (d-r_l+1)^2/\tau_l^2\rceil - r_l^2+1}^{\lceil (d-r_l+1)^2/\tau_l^2\rceil \tau_l^2} x_k\Bigg),$$
$$\forall \mathbf{x} \in \mathbb{R}^{\lceil (d-r_l+1)^2/\tau_l^2\rceil \tau_l^2}, \tag{18}$$

and for all $\mathbf{x} \in \mathbb{R}^{d\times d\times C}$,

$$\sigma_{l,c}(\mathbf{x}) = \{\hat{x}_c(a,b)\}_{a,b=1}^{d-r_l+1}, \tag{19}$$

$$\hat{x}_c(a,b) = \sigma\Bigg(\sum_{u=0}^{r_l-1}\sum_{v=0}^{r_l-1} x(a+u, b+v, c) W_{l,c}(u+1, v+1)\Bigg). \tag{20}$$

Lemma 5 (Convolutional Layer with Average Pooling) *Let $\mathcal{F}$ be a set of functions mapping from some set $\mathcal{X}$ to $\mathbb{R}^m$ for some $m \in \mathbb{Z}_+$. Consider a convolutional layer with average pooling defined in Definition 4. Recall the definition of $\mathcal{L}$ in* (9)*. Then, it hold that*

$$\mathbb{E}_{\boldsymbol{\varepsilon}}\Bigg[\sup_{c\in[C]}\sup_{l\in[Q]}\sup_{\psi_l\in\Psi}\sup_{f\in\mathcal{F}}\Bigg\|\frac{1}{n}\sum_{i=1}^{n}\varepsilon_i\psi_{l,c}\circ f(\mathbf{x}_i)\Bigg\|_\infty\Bigg]$$
$$\leq \Bigg[\gamma(\mu)\sup_{c\in[C]}\sup_{l\in[Q]}\Bigg(\sum_{u=0}^{r_l-1}\sum_{v=0}^{r_l-1}\big|W_{l,c}(u+1,v+1)\big|\Bigg)\Bigg]\mathbb{E}\Bigg[\sup_{f\in\mathcal{F}_+}\Bigg\|\frac{1}{n}\sum_{i=1}^{n}\varepsilon_i f(\mathbf{x}_i)\Bigg\|_\infty\Bigg] + \frac{|\sigma(0)|}{\sqrt{n}}, \tag{21}$$

where

$$\gamma(\mu) = \begin{cases} \mu, & \textit{if } \sigma-\sigma(0) \textit{ is odd or belongs to } \mathcal{L} \\ 2\mu, & \textit{if } \sigma-\sigma(0) \textit{ is even} \\ 3\mu, & \textit{if } \sigma-\sigma(0) \textit{ others} \end{cases}. \tag{22}$$

Here,

$$\mathcal{F}_+ = \begin{cases} \mathcal{F}\cup\{-f : f\in\mathcal{F}\}, & \textit{if } \sigma-\sigma(0) \textit{ is odd} \\ \mathcal{F}\cup\{-f : f\in\mathcal{F}\}\cup\{|f| : f\in\mathcal{F}\}, & \textit{if } \sigma-\sigma(0) \textit{ others} \end{cases}. \tag{23}$$

For Dropout layer, the following holds:

Lemma 6 *Let $\psi(\mathbf{x})$ is the output of the $\mathbf{x}$ via the Dropout layer. Then, it holds that*

$$\mathbb{E}_{\boldsymbol{\varepsilon}}\Bigg[\sup_{f\in\mathcal{H}}\Bigg\|\frac{1}{n}\sum_{i=1}^{n}\varepsilon_i\psi\circ f(\mathbf{x}_i)\Bigg\|_\infty\Bigg] \leq \mathbb{E}\Bigg[\sup_{f\in\mathcal{H}}\Bigg\|\frac{1}{n}\sum_{i=1}^{n}\varepsilon_i f(\mathbf{x}_i)\Bigg\|_\infty\Bigg]. \tag{24}$$

The following Rademacher complexity bounds for Dense Layers.

Lemma 7 (Dense Layers) *Recall the definition of $\mathcal{L}$ in* (9)*. Let $\mathcal{G}$ be a class of μ-Lipschitz function, i.e.,*

$$\big|\sigma(x)-\sigma(y)\big| \leq \mu|x-y|, \qquad \forall x, y \in \mathbb{R}, \tag{25}$$

such that $\sigma(0)$ is fixed. Let $\mathcal{V}$ be a class of matrices $\mathbf{W}$ on $\mathbb{R}^{d\times d'}$ such that $\sup_{\mathbf{W}\in\mathcal{V}}\|\mathbf{W}\|_\infty \leq \beta$. For any vector $\mathbf{x} = (x_1, x_2, \cdots, x_{d'})$, we denote by $\sigma(\mathbf{x}) := (\sigma(x_1), \sigma(x_2), \cdots, \sigma(x_{d'}))^T$. Then, it holds that

$$\mathbb{E}_{\boldsymbol{\varepsilon}}\Bigg[\sup_{\mathbf{W}\in\mathcal{V}}\sup_{f\in\mathcal{G}}\Bigg\|\frac{1}{n}\sum_{i=1}^{n}\varepsilon_i\sigma(\mathbf{W}f(\mathbf{x}_i))\Bigg\|_\infty\Bigg]$$
$$\leq \gamma(\mu)\beta\mathbb{E}_{\boldsymbol{\varepsilon}}\Bigg[\sup_{f\in\mathcal{G}}\Bigg\|\frac{1}{n}\sum_{i=1}^{n}\varepsilon_i f(\mathbf{x}_i)\Bigg\|_\infty\Bigg] + \frac{|\sigma(0)|}{\sqrt{n}}, \tag{26}$$

where

$$\gamma(\mu) = \begin{cases} \mu, & \textit{if } \sigma-\sigma(0) \textit{ is odd or belongs to } \mathcal{L} \\ 2\mu, & \textit{if } \sigma-\sigma(0) \textit{ is even} \\ 3\mu, & \textit{if } \sigma-\sigma(0) \textit{ others} \end{cases}. \tag{27}$$

3.3 RADEMACHER COMPLEXITY BOUNDS FOR CNNS

Theorem 8 *Let*

$$\mathcal{L} = \big\{\psi_\alpha : \psi_\alpha(x) = ReLU(x) - \alpha ReLU(-x)\ \ \forall x \in \mathbb{R}, \alpha \in [0,1]\big\}. \tag{28}$$

Consider the CNN defined in Section 3.1 where

$$[f_i(\mathbf{x})]_j = \sigma_i\big(\mathbf{w}_{j,i}^T f_{i-1}(\mathbf{x})\big)\ \ \forall j \in [d_{i+1}]$$

and σ_i is μ_i-Lipschitz. In addition, $f_0(\mathbf{x}) = [\mathbf{x}^T, 1]^T$, $\forall \mathbf{x} \in \mathbb{R}^d$ and $\mathbf{x}$ is normalised such that $\|\mathbf{x}\|_\infty \leq 1$. Let

$$\mathcal{K} = \{i \in [L] : \textit{layer } i \textit{ is a convolutional layer with average pooling}\}, \tag{29}$$

$$\mathcal{D} = \{i \in [L] : \textit{layer } i \textit{ is a dropout layer}\}. \tag{30}$$

We assume that there are Q_i kernel matrices $W_i^{(l)}$'s of size $r_i^{(l)} \times r_i^{(l)}$ for the i-th convolutional layer. For all the (dense) layers that are not convolutional, we define $\mathbf{W}_i$ as their coefficient matrices. In addition, define

$$\gamma_{\text{cvl,i}} = \gamma(\mu_i) \sup_{l \in [Q_i]} \sum_{u=1}^{r_{i,l}} \sum_{v=1}^{r_{i,l}} \big|W_i^{(l)}(u,v)\big|, \tag{31}$$

$$\gamma_{\text{dl,i}} = \gamma(\mu_i)\big\|W_i\big\|_\infty \qquad i \notin \mathcal{K}. \tag{32}$$

where

$$\gamma(\mu_i) = \begin{cases} \mu_i, & \textit{if } \ \sigma_i - \sigma_i(0) \ \textit{ is odd or belongs to } \mathcal{L} \\ 2\mu, & \textit{if } \ \sigma_i - \sigma_i(0) \ \textit{ is even} \\ 3\mu, & \textit{if } \ \sigma_i - \sigma_i(0) \ \textit{ others} \end{cases}. \tag{33}$$

Then, the Rademacher complexity, $\mathcal{R}_n(\mathcal{F})$, satisfies

$$\begin{aligned} \mathcal{R}_n(\mathcal{F}) &:= \mathbb{E}_{\boldsymbol{\varepsilon}}\left[\sup_{f \in \mathcal{F}_+} \left\|\frac{1}{n}\sum_{i=1}^n \varepsilon_i f(\mathbf{x}_i)\right\|_\infty\right] \\ &\leq F_L, \end{aligned} \tag{34}$$

where F_L is estimated by the following recursive expression:

$$F_i = \begin{cases} F_{i-1}\gamma_{\text{cvl,i}} + \frac{|\sigma_i(0)|}{\sqrt{n}}, & i \in \mathcal{K} \\ F_{i-1}\gamma_{\text{dl,i}} + \frac{|\sigma_i(0)|}{\sqrt{n}}, & i \notin (\mathcal{K} \cup \mathcal{D}) \\ F_{i-1}, & i \in \mathcal{D} \end{cases} \tag{35}$$

and $F_0 = \sqrt{\frac{d+1}{n}}$.

Remark 9 *For some special CNNs where all the activation functions belong to ReLU family or odd functions, Theorem 8 shows that the Rademacher complexity does not depend on the network length under the assumption of a bounded product of norms of weight matrices. This result improves Golowich et al.'s bound (Golowich et al., 2018) where the authors showed that the Rademacher complexity is square-root dependent on the depth. It also improve Neyshabur et al.'s bound Neyshabur et al. (2015) where the authors show that the Rademacher complexity depends exponentially on the network-length.*

Proof This is a direct application of Lemmas 5, 6, and 7. By the modelling of CNNs in Section 3.1, it holds that

$$\mathcal{F}_k := \big\{f = f_k \circ f_{k-1} \circ \cdots \circ f_1 \circ f_0 : f_i \in \mathcal{G}_i \subset \{g_i : \mathbb{R}^{d_i} \to \mathbb{R}^{d_{i+1}}\}, \quad \forall i \in \{1, 2, \cdots, k\}\big\} \tag{36}$$

and $\mathcal{F} := \mathcal{F}_L$.

For CNNs, $f_l(\mathbf{x}) = \sigma_l(W_l\mathbf{x}))$ for all $l \in [L]$ where $W_l \in \mathcal{W}_l$ (a set of matrices) and $\sigma_l \in \Psi_l$ where

$$\Psi_l = \big\{\sigma_l : \big|\sigma_l(x) - \sigma_l(y)\big| \le \mu_l|x-y|, \quad \forall x, y \in \mathbb{R}\big\}. \tag{37}$$

Then, since $|\sigma_l|, -\sigma_l \in \Psi_l$, it is easy to see that

$$\mathcal{F}_{l,+} \subset \Psi_l(\mathcal{W}_l\mathcal{F}_{l-1,+}), \qquad \forall l \in [L], \tag{38}$$

where $\mathcal{F}_{l,+}$ is a supplement of $\mathcal{F}_l$ defined in (23).

Therefore, by peeling layer by layer we finally have

$$\mathbb{E}_{\boldsymbol{\varepsilon}}\left[\sup_{f\in\mathcal{F}}\left\|\frac{1}{n}\sum_{i=1}^n \varepsilon_i f(\mathbf{x}_i)\right\|_\infty\right] \le F_L, \tag{39}$$

where for each $i \in [L]$

$$F_i = \begin{cases} F_{i-1}\gamma_{\text{cvl,i}} + \frac{|\sigma_i(0)|}{\sqrt{n}}, & i \in \mathcal{K} \\ F_{i-1}\gamma_{\text{dl,i}} + \frac{|\sigma_i(0)|}{\sqrt{n}}, & i \notin (\mathcal{K}\cup\mathcal{D}) \\ F_{i-1}, & i \in \mathcal{D} \end{cases} \tag{40}$$

and

$$F_0 = \mathbb{E}_{\boldsymbol{\varepsilon}}\left[\sup_{f\in\mathcal{H}_+}\left\|\frac{1}{n}\sum_{i=1}^n \varepsilon_i f(\mathbf{x}_i)\right\|_\infty\right]. \tag{41}$$

Here, $\mathcal{H}_+$ is the extended set of inputs to the CNN, i.e.,

$$\mathcal{H}_+ = \begin{cases} f_0 \cup \{-f_0\}, \ \text{ if } \ \sigma_1 - \sigma_1(0) \ \text{ is odd} \\ f_0 \cup \{-f_0\} \cup \{|f_0|\}, \ \text{ if } \ \sigma_1 - \sigma_1(0) \ \text{ others} \end{cases}. \tag{42}$$

Now, for the case $\sigma_1 - \sigma_1(0)$ is odd, it is easy to see that

$$\sup_{f\in\mathcal{H}_+}\left\|\frac{1}{n}\sum_{i=1}^n \varepsilon_i f(\mathbf{x}_i)\right\|_\infty = \left\|\frac{1}{n}\sum_{i=1}^n \varepsilon_i f_0(\mathbf{x}_i)\right\|_\infty \tag{43}$$

$$\le \left\|\frac{1}{n}\sum_{i=1}^n \varepsilon_i f_0(\mathbf{x}_i)\right\|_2. \tag{44}$$

On the other hand, for the case $\sigma_1 - \sigma_1(0)$ is general, we have

$$\sup_{f\in\mathcal{H}_+}\left\|\frac{1}{n}\sum_{i=1}^n \varepsilon_i f(\mathbf{x}_i)\right\|_\infty \le \max\left\{\left\|\frac{1}{n}\sum_{i=1}^n \varepsilon_i f_0(\mathbf{x}_i)\right\|_\infty, \left\|\frac{1}{n}\sum_{i=1}^n \varepsilon_i \big|f_0(\mathbf{x}_i)\big|\right\|_\infty\right\}. \tag{45}$$

On the other hand, we have

$$\mathbb{E}_{\boldsymbol{\varepsilon}}\left[\left\|\frac{1}{n}\sum_{i=1}^n \varepsilon_i f_0(\mathbf{x}_i)\right\|_2\right]$$

$$\le \frac{1}{n}\sqrt{\mathbb{E}_{\boldsymbol{\varepsilon}}\left[\left\|\frac{1}{n}\sum_{i=1}^n \varepsilon_i f_0(\mathbf{x}_i)\right\|_2^2\right]} \tag{46}$$

$$\le \frac{1}{n}\sqrt{\sum_{j=1}^{d+1}\sum_{i=1}^n [f_0(\mathbf{x}_i)]_j^2} \tag{47}$$

$$\le \frac{1}{n}\sqrt{(d+1)n} \tag{48}$$

$$= \sqrt{\frac{d+1}{n}}, \tag{49}$$

where (48) follows from $|[f_0(\mathbf{x}_i)]_j| \le 1$ for all $i \in [n], j \in [d_1]$ when the data is normalised by using the standard method.

Similarly, we also have

$$\mathbb{E}_{\boldsymbol{\varepsilon}}\left[\left\|\frac{1}{n}\sum_{i=1}^n \varepsilon_i \big|f_0(\mathbf{x}_i)\big|\right\|_2\right] \le \sqrt{\frac{d+1}{n}}. \tag{50}$$

4 GENERALIZATION BOUNDS FOR CNNS

4.1 GENERALIZATION BOUNDS FOR DEEP LEARNING

Definition 10 *Recall the CNN model in Section 3.1. The margin of a labelled example* $(\mathbf{x}, y)$ *is defined as*

$$m_f(\mathbf{x}, y) := g(f(\mathbf{x}), y) - \max_{y' \neq y} g(f(\mathbf{x}), y'), \tag{51}$$

so f *mis-classifies the labelled example* $(\mathbf{x}, y)$ *if and only if* $m_f(\mathbf{x}, y) \leq 0$*. The generalisation error is defined as* $\mathbb{P}(m_f(\mathbf{x}, y) \leq 0)$*. It is easy to see that* $\mathbb{P}(m_f(\mathbf{x}, y) \leq 0) = \mathbb{P}\big(\mathbf{w}_y^T f(\mathbf{x}) \leq \max_{y' \in \mathcal{Y}} \mathbf{w}_{y'}^T f(\mathbf{x})\big)$*.*

Remark 11 *Some remarks:*

- *Since* $g(f(\mathbf{x}), y) > \max_{y' \neq y} g(f(\mathbf{x}), y')$*, it holds that* $\tilde{g}(f_k(\mathbf{x}, y)) > \max_{y' \neq y} \tilde{g}(f_k(\mathbf{x}, y'))$ *for some* $k \in [L]$ *where* $\tilde{g}$ *is an arbitrary function. Hence,* $\mathbb{P}(m_f(\mathbf{x}, y) \leq 0) \leq \mathbb{P}(\tilde{g}(f_k(\mathbf{x}, y)) > \max_{y' \neq y} \tilde{g}(f_k(\mathbf{x}, y')))$*, so we can bound the generalisation error by using only a part of CNN networks (from layer* 0 *to layer* k*). However, we need to know* $\tilde{g}$*. If the last layers of CNN are softmax, we can easily know this function.*
- *When testing on CNNs, it usually happens that the generalisation error bound becomes smaller when we use almost all layers.*

Now, we prove the following lemma.

Lemma 12 *Let* $\mathcal{F}$ *be a class of function from* $\mathcal{X}$ *to* $\mathbb{R}^m$*. For CNNs for classification, it holds that*

$$\mathbb{E}_{\boldsymbol{\varepsilon}}\left[\sup_{f \in \mathcal{F}} \left|\frac{1}{n}\sum_{i=1}^n \varepsilon_i m_f(\mathbf{x}_i, y_i)\right|\right] \leq \beta(M) \mathbb{E}_{\boldsymbol{\varepsilon}}\left[\sup_{f \in \mathcal{F}} \left\|\frac{1}{n}\sum_{i=1}^n \varepsilon_i m_f(\mathbf{x}_i)\right\|_\infty\right], \tag{52}$$

where

$$\beta(M) = \begin{cases} M(2M-1), & M > 2 \\ 2M, & M = 2 \end{cases}. \tag{53}$$

For $M > 2$, (52) is a result of (Koltchinskii & Panchenko, 2002, Proof of Theorem 11). We improve this constant for $M = 2$. Based on the above Rademacher complexity bounds and a justified application of McDiarmid's inequality, we obtains the following generalization for deep learning with i.i.d. datasets.

Theorem 13 *Let* $\gamma > 0$ *and define the following function (the* γ*-margin cost):*

$$\zeta(x) := \begin{cases} 0, & \gamma \leq x \\ 1 - x/\gamma, & 0 \leq x \leq \gamma \\ 1, & x \leq 0 \end{cases}. \tag{54}$$

Recall the definition of the average Rademacher complexity $\mathcal{R}_n(\mathcal{F})$ *in* (34) *and the definition of* $\beta(M)$ *in* (53)*. Let* $\{(\mathbf{x}_i, y_i)\}_{i=1}^n \sim P_{\mathbf{x}y}$ *for some joint distribution* $P_{\mathbf{x}y}$ *on* $\mathcal{X} \times \mathcal{Y}$*. Then, for any* $t > 0$*, the following holds:*

$$\mathbb{P}\bigg\{\exists f \in \mathcal{F} : \mathbb{P}\big(m_f(\mathbf{x}, y) \leq 0\big) > \inf_{\gamma \in (0,1]} \bigg[\frac{1}{n}\sum_{i=1}^n \zeta(m_f(\mathbf{x}_i, y_i)) + \frac{2\beta(M)}{\gamma}\mathcal{R}_n(\mathcal{F}) + \frac{2t + \sqrt{\log\log_2(2\gamma^{-1})}}{\sqrt{n}}\bigg]\bigg\} \leq 2\exp(-2t^2). \tag{55}$$

Corollary 14 *(PAC-bound) Recall the definition of the average Rademacher complexity* $\mathcal{R}_n(\mathcal{F})$ *in* (34) *and the definition of* $\beta(M)$ *in* (53)*. Let* $\{(\mathbf{x}_i, y_i)\}_{i=1}^n \sim P_{\mathbf{x}y}$ *for some joint distribution* $P_{\mathbf{x}y}$ *on*

$\mathcal{X} \times \mathcal{Y}$*. Then, for any* $\delta \in (0, 1]$*, with probability at least* $1 - \delta$*, it holds that*

$$\mathbb{P}\big(m_f(\mathbf{x}, y) \leq 0\big) \leq \inf_{\gamma \in (0,1]} \Bigg[\frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\big\{m_f(\mathbf{x}_i, y_i) \leq \gamma\big\} + \frac{2\beta(M)}{\gamma}\mathcal{R}_n(\mathcal{F}) + \sqrt{\frac{\log\log_2(2\gamma^{-1})}{n}} + \sqrt{\frac{2}{n}\log\frac{3}{\delta}}\Bigg], \qquad \forall f \in \mathcal{F}. \tag{56}$$

Proof This result is obtain from Theorem 13 by choosing $t > 0$ such that $3\exp(-2t^2) = \delta$.

5 NUMERICAL RESULTS

In this experiment, we use a CNN (cf. Fig. 1) for classifying MNIST images (class 0 and class 1), i.e., $M = 2$, which consists of $n = 12665$ training examples.

For this model, the sigmoid activation σ satisfies $\sigma(x) - \sigma(0) = \frac{1}{2}\tanh\left(\frac{x}{2}\right)$ which is odd and has the Lipschitz constant $1/4$. In addition, for the dense layer, the sigmoid activation satisfies

$$\big|\sigma(x) - \sigma(y)\big| \leq \frac{1}{4}|x - y|, \qquad \forall x, y \in \mathbb{R}. \tag{57}$$

Hence, by Theorem 8 it holds that $\mathcal{R}_n(\mathcal{F}) \leq F_3$, where

$$F_3 \leq \underbrace{\frac{1}{4}\|\mathbf{W}\|_\infty F_2 + \frac{1}{2\sqrt{n}}}_{\text{Dense layer}}, \tag{58}$$

$$F_2 \leq \underbrace{\left(\frac{1}{4}\sup_{l \in [64]}\sum_{u=1}^{3}\sum_{v=1}^{3}\big|W_2^{(l)}(u,v)\big|\right)F_1 + \frac{1}{2\sqrt{n}}}_{\text{The second convolutional layer}}, \tag{59}$$

$$F_1 \leq \underbrace{\left(\frac{1}{4}\sup_{l \in [32]}\sum_{u=1}^{3}\sum_{v=1}^{3}\big|W_1^{(l)}(u,v)\big|\right)F_0 + \frac{1}{2\sqrt{n}}}_{\text{The first convolutional layer}}, \tag{60}$$

$$F_0 = \sqrt{\frac{d+1}{n}}. \tag{61}$$

Numerical estimation of F_3 gives $\mathcal{R}_n(\mathcal{F}) \leq 0.00859$.

By Corollary 14 with probability at least $1 - \delta$, it holds that

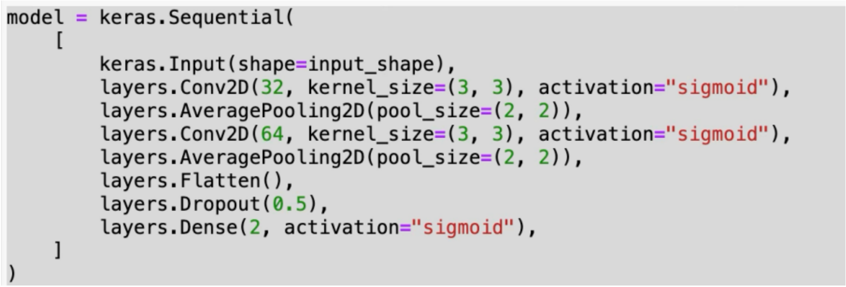
$$\mathbb{P}\big(m_f(\mathbf{x}, y) \leq 0\big) \leq \inf_{\gamma \in (0,1]} \Bigg[\frac{1}{n}\sum_{i=1}^{n} \zeta\big(m_f(\mathbf{x}_i, y_i)\big) + \frac{4M}{\gamma}\mathcal{R}_n(\mathcal{F}) + \sqrt{\frac{\log\log_2(2\gamma^{-1})}{n}} + \sqrt{\frac{2}{n}\log\frac{3}{\delta}}\Bigg] \tag{62}$$

By setting $\delta = 5\%$, $\gamma = 0.5$, the generalisation error can be upper bounded by

$$\mathbb{P}\big(m_f(\mathbf{x}, y) \leq 0\big) \leq 0.189492. \tag{63}$$

For this model, the reported test error is 0.0028368.

Two extra experiments are given in Supplementary Materials.



```
model = keras.Sequential(
    [
        keras.Input(shape=input_shape),
        layers.Conv2D(32, kernel_size=(3, 3), activation="sigmoid"),
        layers.AveragePooling2D(pool_size=(2, 2)),
        layers.Conv2D(64, kernel_size=(3, 3), activation="sigmoid"),
        layers.AveragePooling2D(pool_size=(2, 2)),
        layers.Flatten(),
        layers.Dropout(0.5),
        layers.Dense(2, activation="sigmoid"),
    ]
)
```

Figure 1: CNN model with sigmoid activations

6 COMPARISION WITH GOLOWICH ET AL.'S BOUND (GOLOWICH ET AL., 2018)

In (Golowich et al., 2018, Section 4), the authors present an upper bound on Rademacher complexity for DNNs with ReLU activation functions as follows:

$$\mathcal{R}_n(\mathcal{F}) = O\left(\prod_{j=1}^{L} \|\mathbf{W}_j\|_F \max\left\{1, \log\left(\prod_{j=1}^{L} \frac{\|\mathbf{W}_j\|_F}{\|\mathbf{W}_j\|_2}\right)\right\} \min\left\{\frac{\max\{1,\log n\}^{3/4}}{n^{1/4}}, \sqrt{\frac{L}{n}}\right\}\right) \tag{64}$$

where $\mathbf{W}_1, \mathbf{W}_2, \cdots, \mathbf{W}_L$ are the parameter matrices of the L layers.
Now, let Γ be the term inside the bracket in (64), and define

$$\beta = \min_j \frac{\|\mathbf{W}_j\|_F}{\|\mathbf{W}_j\|_2} \geq 1. \tag{65}$$

Then, from (64) we have

$$\Gamma \geq \prod_{j=1}^{L} \|\mathbf{W}_j\|_F \min\left\{\frac{\max\{1,\log n\}^{3/4}\sqrt{\max\{1, L\log\beta\}}}{n^{1/4}}, \sqrt{\frac{L}{n}}\right\}. \tag{66}$$

For the general case, it holds that $\beta > 1$. Hence, from (66) we have

$$\mathcal{R}_n(\mathcal{F}) = O\left(\sqrt{\frac{L}{n}}\prod_{j=1}^{L} \|\mathbf{W}_j\|_F\right) \tag{67}$$

which depends on the square-root of the network-length under the assumption of a bounded product of weight matrices $\prod_{j=1}^{L} \|\mathbf{W}_j\|_F$. As shown in (Golowich et al., 2018), this bound improves many previous bounds, including Neyshabur et al.'s bound Neyshabur et al. (2015), Bartlett et al. (2017), Neyshabur et al. (2018).
By using Theorem 8 and Lemma 7, we can show that

$$\mathcal{R}_n(\mathcal{F}) = O\left(\sqrt{\frac{1}{n}}\prod_{j=1}^{L} \mu_j \|\mathbf{W}_j\|_\infty\right) \tag{68}$$

for DNNs with some special classes of activation functions, including ReLU family and classes of old activation functions, where μ_j is the Lipschitz constant of the j-layer activation function. Hence, under the assumption of a bounded product of norms of weight matrices $\prod_{j=1}^{L} \|\mathbf{W}_j\|_\infty$, our derived bound is independent of the network depth. This represents a significant improvement over the bound established by Golowich et al., particularly for DNNs with a large network depth L.

REFERENCES

E. Parrado-Hern’’andez A. Ambroladze and J. ShaweTaylor. Tighter PAC-Bayes bounds. In *NIPS*, 2007.

Peter Bartlett and John Shawe-Taylor. *Generalization Performance of Support Vector Machines and Other Pattern Classifiers*, pp. 43–54. MIT Press, 1999.

Peter Bartlett, Yoav Freund, Wee Sun Lee, and Robert E. Schapire. Boosting the margin: a new explanation for the effectiveness of voting methods. *The Annals of Statistics*, 26(5):1651 – 1686, 1998a.

Peter L. Bartlett and Robert C. Williamson. The vc dimension and pseudodimension of two-layer neural networks with discrete inputs. *Neural Computation*, 8:625–628, 1996.

Peter L. Bartlett, Vitaly Maiorov, and Ron Meir. Almost linear vc-dimension bounds for piecewise polynomial networks. *Neural Computation*, 10:2159–2173, 1998b.

Peter L. Bartlett, Dylan J. Foster, and Matus Telgarsky. Spectrally-normalized margin bounds for neural networks. In *NIPS*, 2017.

F. Biggs and B. Guedj. Differentiable PAC-Bayes objectives with partially aggregated neural networks. *Entropy*, 23, 2021.

Eugenio Clerico, George Deligiannidis, and Arnaud Doucet. Conditional Gaussian PAC-Bayes. *Arxiv: 2110.1188*, 2021a.

Eugenio Clerico, George Deligiannidis, and Arnaud Doucet. Wide stochastic networks: Gaussian limit and PACBayesian training. *Arxiv: 2106.09798*, 2021b.

John C. Duchi, Alekh Agarwal, Mikael Johansson, and Michael I. Jordan. Ergodic mirror descent. *2011 49th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 701–706, 2011.

G. K. Dziugaite and D. M. Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. In *Uncertainty in Artificial Intelligence (UAI)*, 2017.

Amedeo Roberto Esposito, Michael Gastpar, and Ibrahim Issa. Generalization error bounds via Rényi-f-divergences and maximal leakage. *IEEE Transactions on Information Theory*, 67(8): 4986–5004, 2021.

Paul W. Goldberg and Mark Jerrum. Bounding the vapnik-chervonenkis dimension of concept classes parameterized by real numbers. *Machine Learning*, 18:131–148, 1993.

Noah Golowich, Alexander Rakhlin, and Ohad Shamir. Size-independent sample complexity of neural networks. In *COLT*, 2018.

D. Jakubovitz, R. Giryes, and M. R. D. Rodrigues. Generalization Error in Deep Learning. *Arxiv: 1808.01174*, 30, 2018.

V. Koltchinskii and D. Panchenko. Empirical Margin Distributions and Bounding the Generalization Error of Combined Classifiers. *The Annals of Statistics*, 30(1):1 – 50, 2002.

J. Langford and J. Shawe-Taylor. PAC-Bayes and Margins. In *Advances of Neural Information Processing Systems (NIPS)*, 2003.

M. Ledoux and M. Talagrand. *Probability in Banach Spaces*. Springer, New York., 1991.

A. McAllester. Some PAC-Bayesian theorems. In *Conference on Learning Theory (COLT)*, 1998.

D. A. McAllester. PAC-Bayesian stochastic model selection. *Machine Learning*, 51, 2004.

Mehryar Mohri and Andrés Muñoz Medina. Learning theory and algorithms for revenue optimization in second price auctions with reserve. In *ICML*, 2014.

V. Nagarajan and Z. Kolter. Uniform convergence may be unable to explain generalization in deep learning . In *Advances of Neural Information Processing Systems (NeurIPS)*, 2019.

Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. Norm-based capacity control in neural networks. In *COLT*, 2015.

Behnam Neyshabur, Srinadh Bhojanapalli, David A. McAllester, and Nathan Srebro. A PAC-bayesian approach to spectrally-normalized margin bounds for neural networks. *ArXiv*, abs/1707.09564, 2018.

M. Raginsky and I. Sason. *Concentration of measure inequalities in information theory, communications and coding*, volume 10 of *Foundations and Trends in Communications and Information Theory*. Now Publishers Inc, 2013.

H. Royden and P. Fitzpatrick. *Real Analysis*. Pearson, 4th edition, 2010.

Lan V. Truong. Generalization Bounds on Multi-Kernel Learning with Mixed Datasets. *ArXiv*, 2205.07313, 2022a.

Lan V. Truong. Generalization Error Bounds on Deep Learning with Markov Datasets. In *Advances of Neural Information Processing Systems (NeurIPS)*, 2022b.

Lan V. Truong. On linear model with markov signal priors. In *AISTATS*, 2022c.

V. N. Vapnik. *Statistical Learning Theory*. Wiley, New York, 1998.

M. Austern R. P. Adams W. Zhou, V. Veitch and P. Orbanz. Non-vacuous generalization bounds at the imagenet scale: a PAC-Bayesian compression approach. In *The International Conference on Learning Representations (ICLR)*, 2019.

Gang Wang, Bingcong Li, and Georgios B. Giannakis. A multistep lyapunov approach for finite-time analysis of biased stochastic approximation. *ArXiv*, abs/1909.04299, 2019.

A. Xu and M. Raginsky. Information-theoretic analysis of generalization capability of learning algorithms. In *Advances of Neural Information Processing Systems (NIPS)*, 2017.