THEORETICAL ERROR PERFORMANCE ANALYSIS FOR DEEP NEURAL NETWORK BASED REGRESSION FUNCTIONAL APPROXIMATION

A Ph.D. Dissertation
Presented to
The Academic Faculty

By

Jun Qi

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Engineering
Department of Electrical and Computer Engineering

Georgia Institute of Technology

May 2022

THEORETICAL ERROR PERFORMANCE ANALYSIS FOR DEEP NEURAL NETWORK BASED REGRESSION FUNCTIONAL APPROXIMATION

Thesis committee:

Prof. Chin-Hui Lee
Electrical and Computer Engineering
Georgia Institute of Technology

Prof. Justin Romberg
Electrical and Computer Engineering
Georgia Institute of Technology

Prof. Xiaoli Ma
Electrical and Computer Engineering
Georgia Institute of Technology

Prof. Sabato Marco Siniscalchi
Computer Engineering
University of Enna Kore

Prof. David Anderson
Electrical and Computer Engineering
Georgia Institute of Technology

Date approved: January 6, 2022

ACKNOWLEDGMENTS

First, I would like to express my most sincere gratitude to my advisors Prof. Chin-Hui Lee and Prof. Xiaoli Ma for their constant support of my Ph.D. study. Special thanks are extended to my co-advisor Prof. Sabato Marco Siniscalchi for his valuable suggestions and collaborative work on my Ph.D. work. Their extraordinary insight, wisdom, and experience greatly benefit my research. For one thing, I acquired a great deal of knowledge, developed a strong research interest, laid solid mathematical and programming skills, and enhanced my research capability in my studying fields. For another, their great personalities and wonderful accomplishments have made them the role model of my life.

I would like to thank the rest of my thesis committee members: Prof. David Anderson, and Prof. Justin Romberg for their precious time reading my thesis, insightful comments, and encouragement. I also thanks Prof. Biing-Hwang Juang for his suggestions in improving the theoretical work in the dissertation.

I also owe a debt of gratitude to my friends during my Ph.D. study at Georgia Tech. I would like to thank our group members Chao-Han Huck Yang and Hu Hu for their collaboration on the projects and research. I would like to thank my other research collaborators Prof. Jun Du, Dr. Min-Hsiu Hsieh, Prof. Javier Tejedor, Prof. Dong Wang, Dr. Pin-Yu Chen, Dr. Yen-Chi Chen, Muhammad Shahmeer Omar, Rui Zhang, and Jing Zhang. Thanks to Pat Dixon, Raquel Plaskett, Tasha Torrence, and Daniela Staiculescu for their great administrative support.

I would like to express my appreciation to Dr. Xiaodong He and Dr. Li Deng whom I worked with at Microsoft Research, and to Dr. Dong Yu at Tencent American AI Lab. I was motivated by their high standards in research and publications and inspired by their great insight into cutting-edge technologies.

Finally, I would like to express my deepest gratitude to my father Jianchang Qi and my mother Yuqing Lian for their boundless love, support, and understanding.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Based on Kolmogorov's superposition theorem and universal approximation theorems by Cybenko and Barron, any vector-to-scalar function can be approximated by a multi-layer perceptron (MLP) within certain bounds. The theorems inspire us to exploit deep neural networks (DNN) based vector-to-vector regression. This dissertation aims at establishing theoretical foundations on DNN based vector-to-vector functional approximation, and bridging the gap between DNN based applications and their theoretical understanding in terms of representation and generalization powers.

Concerning the representation power, we develop the classical universal approximation theorems and put forth a new upper bound to vector-to-vector regression. More specifically, we first derive upper bounds on the artificial neural network (ANN), and then we generalize the concepts to DNN based architectures. Our theorems suggest that a broader width of the top hidden layer and a deep model structure bring a more expressive power of DNN based vector-to-vector regression, which is illustrated with speech enhancement experiments.

As for the generalization power of DNN based vector-to-vector regression, we employ a well-known error decomposition technique, which factorizes an expected loss into the sum of an approximation error, an estimation error, and an optimization error. Since the approximation error is associated with our attained upper bound upon the expressive power, we concentrate our research on deriving the upper bound for the estimation error and optimization error based on statistical learning theory and non-convex optimization. Moreover, we demonstrate that mean absolute error (MAE) satisfies the property of Lipschitz continuity and exhibits better performance than mean squared error (MSE). The speech enhancement experiments with DNN models are utilized to corroborate our aforementioned theorems.

Finally, since an over-parameterized setting for DNN is expected to ensure our theoretical upper bounds on the generalization power, we put forth a novel deep tensor learning framework, namely tensor-train deep neural network (TT-DNN), to deal with an explosive

DNN model size and realize effective deep regression with much smaller model complexity. Our experiments of speech enhancement demonstrate that a TT-DNN can maintain or even achieve higher performance accuracy but with much fewer model parameters than an even over-parameterized DNN.

CHAPTER 1
INTRODUCTION

1.1 Introduction to Vector-to-Vector Regression

The vector-to-vector regression problem is of great interest in machine learning, signal processing, and wireless communications. For example, speech enhancement aims at finding a regression operator to convert noisy speech spectral vectors to clean ones [1, 2]. Similarly, clean images can be estimated from the corrupted ones by leveraging upon image de-noising techniques [3]. Moreover, wireless communication systems can be designed to transmit encrypted and corrupted codes to targeted receivers with the decoding information as correct as possible [4, 5]. Furthermore, vector-to-vector regression is also seen in ecological modeling, natural gas demand forecasting, and drug efficacy prediction domains [6].

To implement a regression function f, several machine learning and signal processing approaches have been proposed. Linear regression [7] admits a straightforward implementation, which originated from Pearson's research in the evolution theory. Support vector regression (SVR) [8] is a kernel-based algorithm and aims at solving a max-margin optimization problem. However, the SVR is computationally expensive when dealing with a large set of training data [9]. Lasso [10] and group Lasso-based regularization [11] admits a sparse solution that selects representative features. Unfortunately, the regularization-based approaches do not allow the natural use of kernels, which prevents their extension to the nonlinear vector-to-vector regression.

With the resurgence of neural networks in many fields of machine learning and signal processing, DNN with multiple hidden layers can provide an efficient and robust solution in dealing with large-scale vector-to-vector regression problems. One typical example is

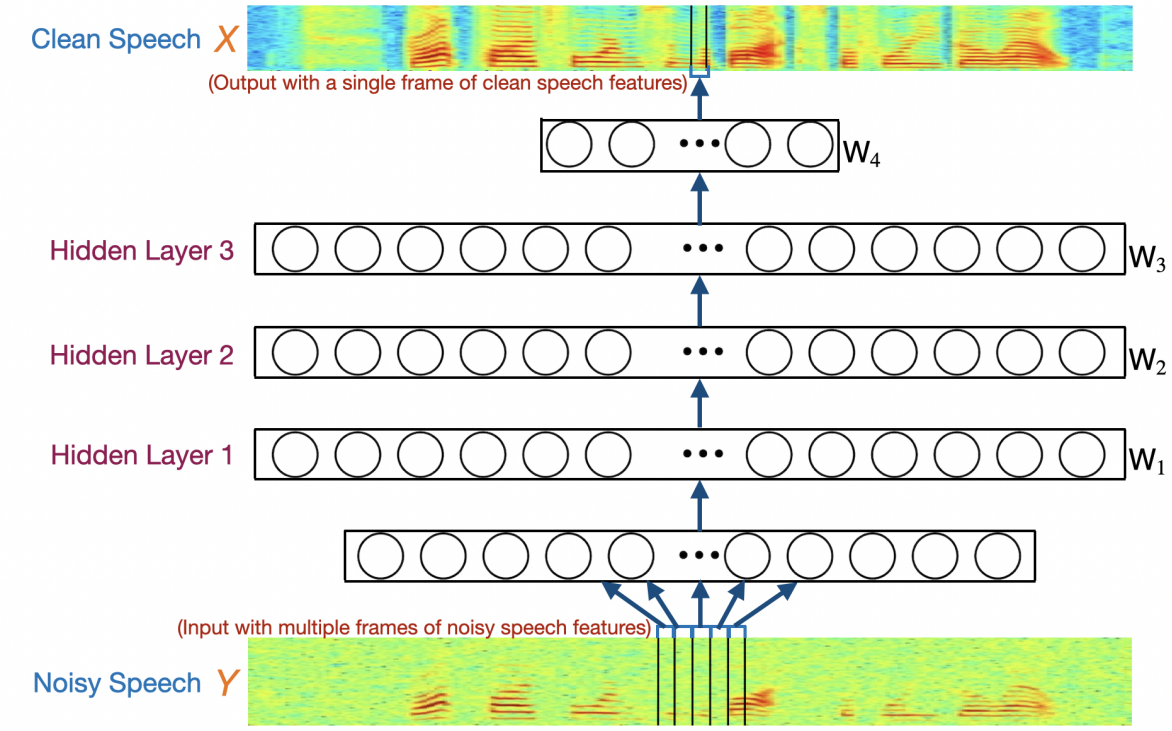

Figure 1.1: *DNN based vector-to-vector regression for speech enhancement.*

from the speech enhancement as shown in Figure 1.1, where the input is high-dimensional vectors of noisy speech, and the output corresponds to the vectors of enhanced speech. A feed-forward DNN composed of 3 hidden layers realizes the vector-to-vector regression for speech enhancement and outperforms the traditional approaches [12]. However, the theoretical understanding of DNN-based vector-to-vector regression is still lacking and this thesis aims at filling the gap.

1.2 Motivations and Scientific Goals

Our theory is inspired by the classical universal approximation theory. The idea of universal approximation first appeared in Kolmogorov's superposition theorem [13, 14], where the representation power of multivariate continuous functions with feed-forward structures are considered in two different Euclidean spaces. However, Kolmogorov's superposition theorem cannot be easily adopted in practice because the exact inner and outer functions are difficult to specify. Later, Cybenko [15] and Barron [16] developed universal approxi-

mation theorems based on Kolmogorov's superposition theorem. They claimed that an artificial neural network (ANN) [17] with a Sigmoid activation function [18] can approximate any continuous vector-to-scalar function. Similarly, Hornik *et al.* [19] also demonstrated that a multi-layer feed-forward network is a universal approximator. More specifically, a well-configured ANN can approximate any vector-to-scalar function with an arbitrarily small error, even though only a non-linear hidden layer is stacked between the input layer and output one.

Inspired by the classical universal approximation theory, Xu *et al.* [2, 1] exploited deep neural network (DNN) [20] based vector-to-vector regression within a machine learning framework, where the DNN parameters can be adjusted by adopting machine learning optimization algorithms, and the DNN based models exhibit advantages over other approaches [1]. However, a theoretical understanding of DNN based vector-to-vector regression has not been systematically studied, and thus this dissertation aims at bridging the gap between the empirical DNN study and the related theoretical analysis. More specifically, we first analyze the representation power of DNN-based regression and illustrate experiments of speech enhancement to justify our theorems. Then, we compare mean absolute error (MAE) [21] with mean squared error (MSE) [22] as an objective function for the DNN-based vector-to-vector regression, and we also highlight the advantages of MAE over MSE for DNN based regression problems. Moreover, the generalization power of DNN-based regression is also discussed given the fact that the potential mismatch between training and test data exists in real machine learning applications. Furthermore, our theory suggests that an over-parameterized DNN is expected to lower the optimization (training) error, we employ the tensor-train (TT) decomposition [23] to each hidden layer of DNN such that the resulting TT-DNN model owns much fewer model parameters. In this work, we assess if the TT technique does not only reduces the DNN parameters but it can also maintain the baseline performance of the original DNN.

1.3 Contributions

The goal of this dissertation is to leverage the technique of error performance analysis for the DNN based vector-to-vector regression. Our contributions can be summarized as follows:

1. To investigate the representation power [24] of the DNN-based regression operator, we upper bound an approximation error by developing the classical universal approximation theory. The derived upper bound can be employed to estimate the practical MSE values, and speech enhancement is utilized to validate our theorems.

2. To compare MAE with MSE for DNN-based vector-to-vector regression, we characterize the objective function. Moreover, we highlight the advantages of MAE as an objective function for the DNN based vector-to-vector regression in terms of the Lipschitz continuity [25], the robustness against additive noises, and the functional connection to the Laplacian distribution.

3. To understand the generalization power [26] of DNN based vector-to-vector regression, we upper bound an empirical Rademacher complexity which is closely related to the estimation error, where a large number of training data is highly expected. Our derived upper bound on the estimation error is assessed by employing a series of experiments on speech enhancement and image de-noising.

4. To minimize the optimization error, we analyze how to avoid bad local optimal points by leveraging certain DNN setups and constraints. In particular, we concentrate on the Polyak-Lojasiewicz (PL) condition [27] and an over-parameterization [28] setup for DNN.

5. To reparameterize an over-parameterized DNN into a TT-DNN, we use TT decomposition to convert a DNN into a TT-DNN with TT formats. We empirically show that a TT-DNN model can even obtain the accuracy of the original DNN with much

fewer parameters. In particular, TT-DNN can be applied to the tasks of multi-channel speech enhancement [29, 30] and speech recognition [31, 32, 33].

1.4 Thesis Outline

The thesis is organized as follows: In Chapter 2, we introduce the necessary background knowledge and the classical universal approximation theorem. More specifically, we discuss Kolmogorov's superposition theorem and Cybenko and Barron's theorems which are associated with the expressive capability of a neural network with only one hidden layer. In Chapter 3, we aim to analyze the representation power of DNN based vector-to-vector regression, and we illustrate the experiments of speech enhancement to corroborate the derived upper bound [34]. In Chapter 4, we exploit the characteristics of MAE for DNN based regression operators and compare it with MSE as a loss function for DNN based regression problems [35]. In Chapter 5, we investigate the generalization power of DNN based vector-to-vector regression and conduct experiments on speech enhancement and image de-noising to justify our derived theorems [36]. In Chapter 6, we characterize TT-DNN and discuss if it can maintain the representation power of DNN with the demonstration of speech enhancement and speech recognition experiments [37, 38]. In Chapter 7, we conclude the thesis and present the future work.

1.5 Notations

We define the notations consistently used throughout this thesis.

- $f \circ g$: The composition of functions f and g.
- $||\mathbf{v}||_p$: L_p norm of the vector $\mathbf{v}$.
- $\langle \mathbf{x}, \mathbf{y} \rangle$ and $\mathbf{x}^\top \mathbf{y}$: Inner product of two vectors $\mathbf{x}$ and $\mathbf{y}$.
- $[Q]$: An integer set $\{1, 2, 3, ..., Q\}$.

- ∇f: A first-order gradient of the function f.
- $\mathbb{E}[X]$: Expectation over a random variable X.
- $\mathbf{1}$: A vector of all ones.
- $\mathbf{1}_m$: Indicator vector of zeros but with the m-th dimension assigned to 1.
- $\mathbb{R}^I$: I-dimensional real coordinate space.
- $\mathbb{R}^{I_1 \times I_2 \times \cdots \times I_K}$: space of K-order tensors.
- $[a, b]$: Closed interval between a and b.
- (a, b): Open interval between a and b.
- $\mathcal{O}(\cdot)$: If $\mathcal{T}(r) = \mathcal{O}(f(r))$, there exist constants $z, r_0 \geq 0$ such that $\mathcal{T}(r) \leq zf(r)$ for all $r \geq r_0$.
- $\Theta(\cdot)$: If $\mathcal{T}(r) = \Theta(f(r))$, there exist constants z_1, z_2, $r_0 \geq 0$, such that $z_1 f(r) \leq \mathcal{T}(r) \leq z_2 f(r)$ for all $r \geq r_0$.
- $o(\cdot)$: If $\mathcal{T}(r) = o(f(r))$, there exist constants $z, r_0 \geq 0$ such that $\mathcal{T}(r) \geq zf(r)$ for all $r \geq r_0$.
- $h^*_{\mathcal{D}}$: The best hypothesis of all functions over the distribution $\mathcal{D}$.
- $f^*_{\mathcal{D}}$: The optimal DNN hypothesis over the distribution $\mathcal{D}$.
- f^*_S: The DNN empirical risk minimizer on the training dataset S.
- $\bar{f}_S$: The returned DNN hypothesis on the training dataset S.
- $\mathbb{F}_K$: The class of DNN hypothesis with K hidden layers.
- $\mathcal{L}_{\mathcal{D}}$: The expected loss function over the distribution $\mathcal{D}$.
- $\mathcal{L}_S$: The empirical loss function over the empirical training dataset S.

CHAPTER 2

BACKGROUND

In this chapter, we review the related literature that inspired us to conduct the research exposed in this thesis. In more detail, we first introduce the universal approximation theory [15, 16] and highlight its significance in analyzing the vector-to-vector regression based on artificial neural networks (ANN) [17] and DNNs [20], respectively. Then, we investigate practical applications of the DNN based vector-to-vector operators, which require further theoretical foundations to interpret the results gathered in the related empirical studies. Finally, tensor-train decomposition [23] and the related tensor-train networks [39] are comprehensively discussed.

2.1 A Review of Classical Universal Approximation Theorems

2.1.1 Kolmogorov's Superposition Theorem

The classical universal approximation theorems were inspired by Kolmogorov's superposition theorem [40]. Theorem 1 states that a real continuous function $f : [0, 1]^D \to \mathbb{R}$ can be exactly represented by a superposition of a two-layer function, where there are $D(2D + 1)$ inner functions at the bottom and $(2D + 1)$ outer functions on the top. The inner functions ψ_{pq} are universal because they do not rely on a particular choice of functional type, whereas the outer functions ϕ_q require a delicate design.

Theorem 1 (Kolmogorov's superposition theorem [13])**.** *Given an D-dimensional input vector $\boldsymbol{x} \in \mathbb{R}^D$ and a real continuous function $f : [0, 1]^D \to \mathbb{R}$, there exist $D(2D + 1)$ continuous and monotone increasing univariate functions, by which f can be reconstructed based on Eq. (2.1) as:*

$$f(\boldsymbol{x}) = \sum_{q=1}^{2D+1} \phi_q \left(\sum_{p=1}^{D} \psi_{pq}(x_p) \right). \tag{2.1}$$

2.1.2 Universal Approximation Theorems

Although several works have been investigated in composing Kolmogorov's superposition theorem, some of the most significant contributions arise from Cybenko and Barron's universal approximation theorems [15, 16, 41]. More specifically, instead of exactly composing the inner and outer functions, Cybenko put forth an approximate function based on ANN, as shown in Theorem 2, to replace the inner and outer functions with affine transformations followed by a nonlinear Sigmoid activation function [42].

Theorem 2 (Cybenko's universal approximation theorem [15])**.** *Given $\boldsymbol{x} \in \mathbb{R}^D$ and an arbitrary small $\epsilon > 0$, a continuous function $\hat{f} : \mathbb{R}^D \to \mathbb{R}$ can be approximated by an ANN f such that*

$$\left| \hat{f}(\boldsymbol{x}) - f(\boldsymbol{x}) \right| \leq \epsilon. \tag{2.2}$$

The approximate function f follows a model structure of ANN as:

$$f(\boldsymbol{x}) = \sum_{j=1}^{J} a_j \sigma(\boldsymbol{w}_j^\top \boldsymbol{x} + b_j), \tag{2.3}$$

where ϵ denotes an approximation bias, J neurons are placed in the hidden layer, $\boldsymbol{x}, \boldsymbol{w}_j \in \mathbb{R}^D$, $a_i, b_i \in \mathbb{R}$, and σ refers to a Sigmoid activation function.

Furthermore, Barron quantifies the approximation error ϵ by connecting the width of the hidden layer to the approximation upper bound, which is shown in Theorem 3.

Theorem 3 (Barron's universal approximation theorem [41])**.** *Given a continuous function $\hat{f} : [0,1]^D \to \mathbb{R}$, we can find a function $f : [0,1]^D \to \mathbb{R}$ in the convex hull of J Sigmoid activation functions such that $\forall \boldsymbol{x} \in \mathbb{R}^D$,*

$$\left| \hat{f}(\boldsymbol{x}) - f(\boldsymbol{x}) \right| \leq 2C \left(\frac{1}{\sqrt{J}} + \delta_\tau \right), \tag{2.4}$$

where the constants $\tau > 0$, $C > 0$, and δ_τ refers to a distance between the unit setup

function and a scaled Sigmoid activation function as:

$$\delta_\tau = \min_{0 \le \epsilon \le \frac{1}{2}} \left(2\epsilon + \max_{|z| \ge \epsilon} |\sigma(\tau z)| - 1_{z>0} \right), \tag{2.5}$$

where δ_τ *goes to* 0 *when* τ *becomes infinity, and* $1_{z>0}$ *equals to* 1 *if* $z > 0$ *and otherwise becomes* 0.

The upper bound in Eq. (2.4) suggests that a larger J is related to a lower approximation error, which implies that a sufficiently large number of hidden neurons results in a powerful representation of an arbitrary continuous function.

2.2 From Universal Approximation Theory to Deep Learning-Based Regression

Although the theory of universal approximation implies that the neural network can serve as a universal approximator to any continuous target function [15], the estimation of model parameters should be placed in the context of machine learning framework, where a set of training data is used to learn the DNN model parameters and another set of test data is utilized to assess the DNN performance. The work [2] first investigates the DNN based vector-to-vector regression for speech enhancement experiments in which noisy speech is synthesized by corrupting clean speech with different types of noise at various signal-to-noise ratios (SNRs), and a randomly initialized DNN model is further refined based on the generated noisy speech corpus. Although the training data are not collected from the practical scenarios, the well-trained model based on the synthesized training data owns a strong generalization capability to enhance the unseen noisy speech data in practice. A class of DNN hypothesis is defined in Eq. (2.6), where $\mathbf{W}_k$ denote the weight matrix for the k-th hidden layer and σ refers to a non-linear activation.

$$f_K(\mathbf{x}) = \mathbf{W}_K(\sigma(\mathbf{W}_{K-1}(\cdots(\mathbf{W}_2(\sigma(\mathbf{W}_1(\mathbf{x})))))) \tag{2.6}$$

Then, a detailed introduction to DNN based vector-to-vector regression is described as follows: DNN based vector-to-vector regression [1, 43] provides an effective way to find underlying relationships between input vectors and their corresponding outputs. More specifically, given a finite training dataset $S = \{(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2), ..., (\mathbf{x}_N, \mathbf{y}_N)\}$ independent and identically drawn from an unknown distribution $\mathcal{D}$, where $\mathbf{x}_i \in \mathbb{R}^D$ and $\mathbf{y}_i \in \mathbb{R}^Q$. In the context of machine learning framework, we are concerned with finding a DNN operator $f : \mathbb{R}^D \rightarrow \mathbb{R}^Q$ that can accurately represent the training instances. The regression process is described as:

$$\mathbf{y} = f(\mathbf{x}) + \mathbf{e}, \tag{2.7}$$

where $\mathbf{x} \in \mathbb{R}^D$, $\mathbf{y} \in \mathbb{R}^Q$, and $\mathbf{e} \in \mathbb{R}^Q$ is the regression error vector.

Moreover, the regression operator f is taken as the parametric model, and an expected loss $\mathcal{L}_{\mathcal{D}}(f)$ is used to estimate the parameters of f over the distribution $\mathcal{D}$ and is defined in Eq. (2.8) as:

$$\mathcal{L}_{\mathcal{D}}(f) = \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\mathcal{D}}[\ell(f(\mathbf{x}), \mathbf{y})], \tag{2.8}$$

where $\ell : \mathbb{R}^D \times \mathbb{R}^Q \rightarrow \mathbb{R}$ is a loss function measuring the difference between the DNN operator $f(\mathbf{x})$ and the target vector $\mathbf{y}$.

In the machine learning framework, instead of calculating the expected loss $\mathcal{L}_{\mathcal{D}}(f)$, an empirical loss $\mathcal{L}_S(f)$, which is defined in Eq. (2.9), is employed to approximate $\mathcal{L}_{\mathcal{D}}(f)$. If the training data are representative enough over the distribution $\mathcal{D}$, $\mathcal{L}_S(f)$ can best approximate $\mathcal{L}_{\mathcal{D}}(f)$.

$$\mathcal{L}_S(f) = \frac{1}{N}\sum_{n=1}^{N} \ell(f(\mathbf{x}_n), \mathbf{y}_n). \tag{2.9}$$

Moreover, given the space of all functions $\mathcal{Y}^{\mathcal{X}}$ and DNN hypothesis class $\mathbb{F}_K$, we need to take into account three types of errors: approximation error, estimation error, and optimization error. Figure 2.1 demonstrates the error performance analysis on DNN based

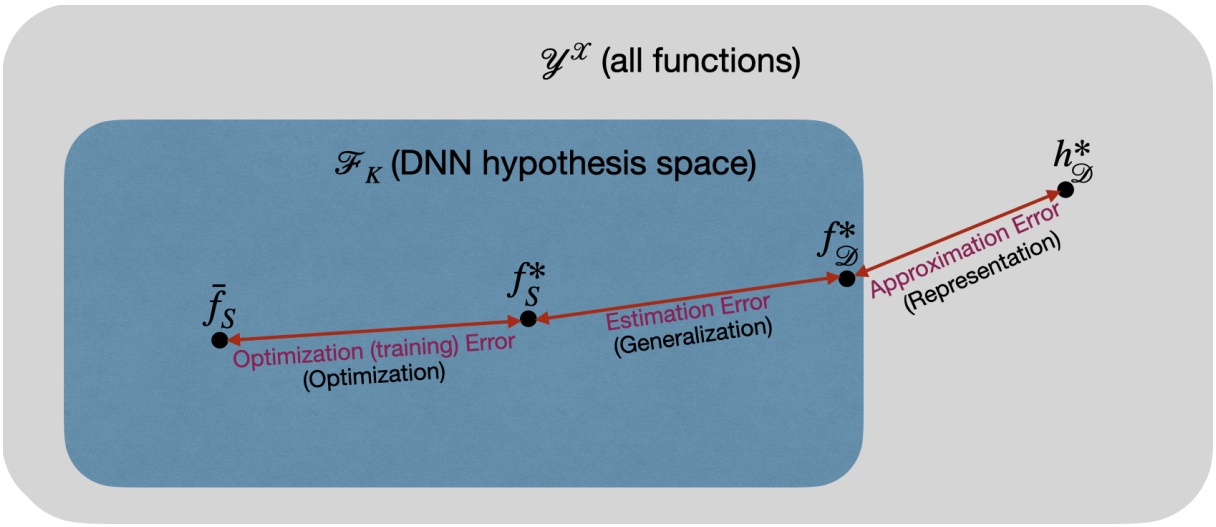

Figure 2.1: *Error performance analysis on DNN based vector-to-vector regression.*

vector-to-vector regression. The difference between the ground truth $h^*_{\mathcal{D}} \in \mathcal{Y}^{\mathcal{X}}$ and the optimal hypothesis $f^*_{\mathcal{D}} \in \mathbb{F}_K$ results in an approximation error, where $h^*_{\mathcal{D}}$ and $f^*_{\mathcal{D}}$ denote the minimizers of the expected loss over $\mathcal{Y}^{\mathcal{X}}$ and the distribution $\mathcal{D}$, respectively.

The estimation error arises from the difference between an empirically optimal hypothesis $f^*_S \in \mathbb{F}_K$. f^*_S refers to a minimizer of the empirical loss over $\mathbb{F}_K$. Furthermore, a returned DNN hypothesis $\bar{f}_S \in \mathbb{F}_K$ is practically attained from training a DNN model based on different DNN initialization and stochastic gradient descent (SGD) algorithms. The bias between $\bar{f}_S$ and f^*_S corresponds to the optimization (training) error.

The error decomposition provides us with a technique of error performance analysis to assess the expected loss on the DNN returned hypothesis $\mathcal{L}_{\mathcal{D}}(\bar{f}_S)$, which is shown as:

$$\mathcal{L}_{\mathcal{D}}(\bar{f}_S) = \underbrace{\mathcal{L}_{\mathcal{D}}(f^*_{\mathcal{D}})}_{\textit{Approximation Error}} + \underbrace{\mathcal{L}_{\mathcal{D}}(f^*_S) - \mathcal{L}_{\mathcal{D}}(f^*_{\mathcal{D}})}_{\textit{Estimation Error}} + \underbrace{\mathcal{L}_{\mathcal{D}}(\bar{f}_S) - \mathcal{L}_{\mathcal{D}}(f^*_S)}_{\textit{Optimization Error}}. \tag{2.10}$$

In Eq. (2.10), $\mathcal{L}_{\mathcal{D}}(\bar{f}_S)$ can be decomposed into the sum of three errors and we focus on the analysis of each error term in this dissertation: the approximation error implies the representation power and measures how well the DNN hypothesis class performs on the distribution $\mathcal{D}$; the estimation error, which is related to the generalization power, results

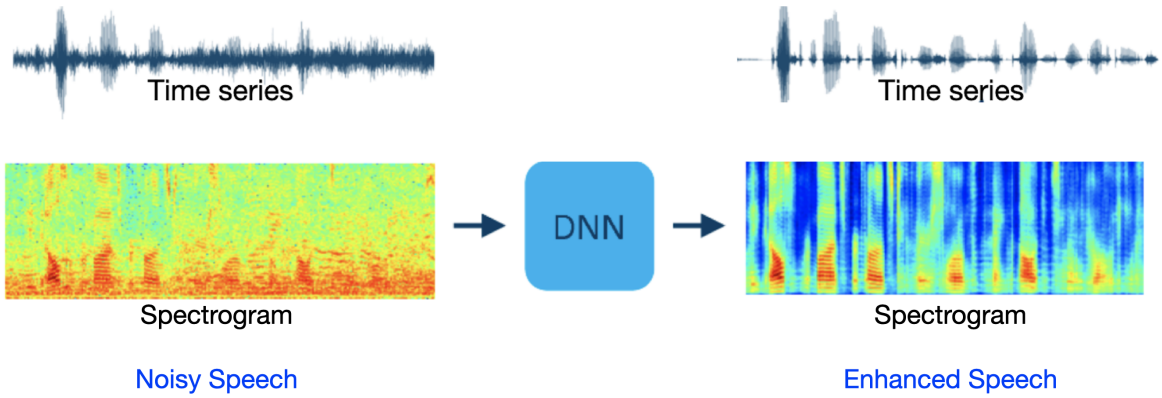

Figure 2.2: *An illustration of speech enhancement for DNN based vector-to-vector regression.*

from the fact that a training dataset is received and the distribution $\mathcal{D}$ cannot be observed; the optimization error measures the difference between the approximation errors concerning the DNN hypothesis $\bar{f}_S$ and f_S^*. This thesis aims at the performance of DNN based vector-to-vector regression in terms of analyzing the three error terms, respectively.

2.2.1 Speech Enhancement

Figure 2.2 illustrates a framework of speech enhancement for DNN based vector-to-vector regression [1], where the input is connected to high-dimensional noisy speech spectrograms and the output corresponds to the clean or enhanced ones. The parametric DNN operator f is designed to map the noisy speech spectrogram to the enhanced one. The DNN framework aims at finding a DNN model which attains the minimum approximation error between the enhanced speech and the reference one.

A feed-forward DNN architecture for speech enhancement was first introduced in the experimental study [1, 2], which achieves the much better performance of speech enhancement. Then, new deep learning models, such as Long Short-Term Memory (LSTM) [44], Generative Adversarial Network (GAN) [45] and Convolutional Neural Network (CNN) [46], was proposed to further improve the speech enhancement performance.

A typical dataset of speech enhancement in our experiments is conducted on the Ed-

inburgh noisy speech database [47], where the noisy backgrounds of the training data are inconsistent with the test ones. More specifically, clean utterances are from 56 speakers including 28 males and 28 females from different accent regions of both Scotland and the United States. Clean materials are randomly split into 23075 training, and 824 test waveforms, respectively. The noisy waveforms at four SNR levels, 15dB, 10dB, 5dB, and 0dB, are set up by mixing the following noises: a domestic noise (inside a kitchen), an office noise (in a meeting room), three public space noises (cafeteria, restaurant, subway station), two transportation noises (car and metro) and a street noise (busy traffic intersection). In summary, 40 different types of noise are synthesized into 23075 noisy training speech utterances. As for the test set, the noisy conditions include a domestic (living room), an office noise (office space), one transport (bus), and two street noises (open area cafeteria and a public square) at 4 SNR levels (17.5 dB, 12.5 dB, 7.5 dB, 2.5 dB). Therefore, 20 various noisy conditions can result in a total of 824 noisy test speech utterances.

Moreover, in the dissertation, the evaluation metrics for speech enhancement are based on three types of criteria: MAE, MSE, perceptual evaluation of speech quality (PESQ) [48], and short-time objective intelligibility (STOI) [49]. MAE and MSE are directly related to the objective loss; PESQ, which ranges from -0.5 to 4.5, is an indirect evaluation that is highly correlated with speech quality; The STOI score lies in the range from 0 to 1 and refers to a measurement of predicting the intelligibility of enhanced speech. A higher PESQ or STOI score corresponds to a better speech perception quality.

2.2.2 Image De-noising

As shown in Figure 2.3, image de-noising [3] is another illustration of DNN based vector-to-vector regression, where an encoder-decoder architecture is applied. The DNN encoder transforms the input images into the abstract features associated with a bottleneck layer, and the decoder reconstructs the features back to the output image. Being similar to the task of speech enhancement, a DNN based vector-to-vector operator f is expected to attain

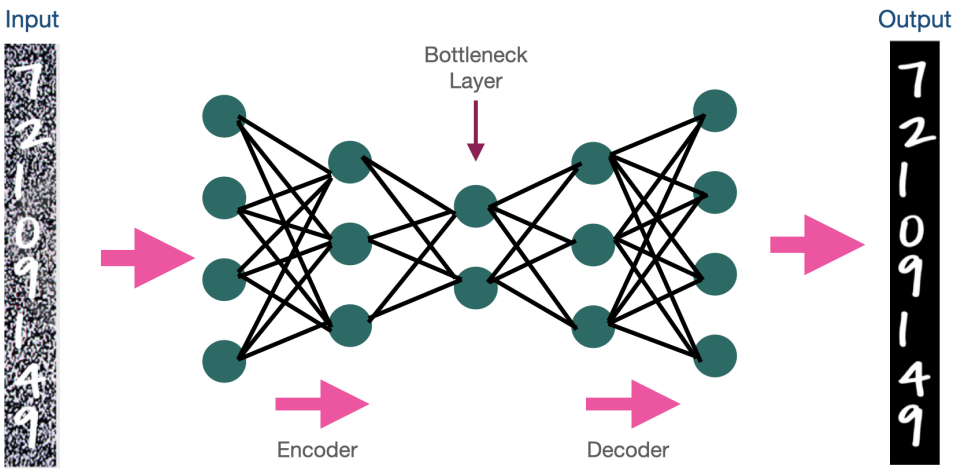

Figure 2.3: *An illustration of image de-noising for DNN based vector-to-vector regression.*

an enhanced image from a noisy input one.

Despite the successful application of DNN based vector-to-vector regression, the related fundamental understanding is still lacking in theory. Therefore, this thesis concentrates on the performance analysis of DNN based vector-to-vector regression. Moreover, we also attempt the use of tensor-train (TT) decomposition [23] for DNN. An introduction to the TT decomposition and the related tensor-train network (TTN) [50] are discussed in the remainder of Chapter 2.

2.3 Tensor-Train Decomposition and Tensor-Train Network

Tensor-Train (TT) [23], also known as matrix product state (MPS) [51], characterizes the representation of a chain-like product of three-index core tensors for a higher-order tensor. More generally, TT denotes a technique of tensor decomposition to factorize a multi-dimensional array into latent factors in a low-dimensional multi-linear space. In doing so, the overhead of memory storage can be greatly reduced. Compared with other tensor decomposition approaches [52] like Tucker decomposition [53] and CANDECOMP/PARAFAC (CP) decomposition [54, 55], TT is referred to a special case of a tree-structured tensor network and can be simply scaled to arbitrarily higher-order tensors. Thus, TT has been widely

used in many signal processing and machine learning domains, such as multi-dimensional harmonic retrieval [56], video classification [50], efficient inference in Markov random field [57], and low-rank tensor decomposition [39]. Moreover, the TT technique can be employed for the channel estimation in wireless communication [58], and it can also apply to cyber-physical-social systems [59].

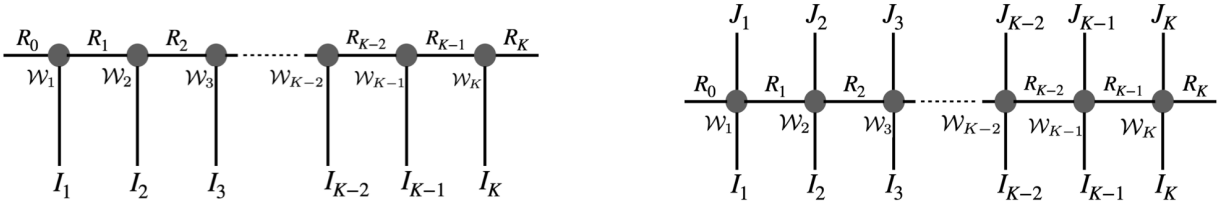

Figure 2.4: *An illustration of TTD and TTN. (a) TTD is a tensor of order K in the TT format and the core tensors are of order 3. A circle represents a core tensor, and each line is associated with the dimension; (b) TTN is a tensor of order K in the TT format and the core tensors are of order 4, a circle represents a core tensor and each line is associated with the dimension.*

2.3.1 Tensor-Train Decomposition

Tensor-train decomposition (TTD) [23] denotes that given a class of positive integer ranks $\{R_0, R_1, ..., R_K\}$, the K-order tensor $\mathcal{W} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_K}$ can be factorized into the multiplication of 3-order tensors. More specifically, given a set of indices $\{m_1, m_2, ..., m_K\}$, $\mathcal{W}(m_1, m_2, ..., m_K)$ is decomposed as:

$$\mathcal{W}(m_1, m_2, ..., m_K) = \prod_{k=1}^{K} \mathcal{W}_k(m_k), \tag{2.11}$$

where $\forall m_k \in I_k$, $\mathcal{W}_k \in \mathbb{R}^{R_{k-1} \times I_k \times R_k}$ and $\mathcal{W}_k(m_k) \in \mathbb{R}^{R_{k-1} \times R_k}$. Since $R_0 = R_K = 1$, the term $\prod_{k=1}^{K} \mathcal{W}_k(m_k)$ is a scalar value. Next, we show that a tensor-train network (TTN) can be generated by employing the TT technique to DNN.

2.3.2 Tensor-Train Network

A TTN refers to a TT representation of a feed-forward neural network with a fully-connected (FC) hidden layer. In more detail, for an input tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_K}$ and the output tensor $\mathcal{Y} \in \mathbb{R}^{J_1 \times J_2 \times \cdots \times J_K}$, we achieve that

$$\begin{aligned} \mathcal{Y}(j_1, j_2, ..., j_K) &= \sum_{i_1=1}^{I_1} \cdots \sum_{i_K=1}^{I_K} \mathcal{W}((i_1, j_1), ..., (i_K, j_K)) \cdot \mathcal{X}(i_1, i_2, ..., i_K) \\ &= \sum_{i_1=1}^{I_1} \cdots \sum_{i_K=1}^{I_K} \left(\prod_{k=1}^{K} \mathcal{W}_k(i_k, j_k) \right) \cdot \mathcal{X}(i_1, i_2, ..., i_K), \end{aligned} \tag{2.12}$$

where $\mathcal{W}((i_1, j_1), (i_2, j_2), ..., (i_K, j_K))$ is closely associated with $\mathcal{W}(m_1, m_2, ..., m_K)$ as defined in Eq. (2.11), if we set the index $m_k = i_k \times j_k, k \in [K]$. In doing so, given the class of ranks $\{R_0, R_1, ..., R_K\}$, the tensor $\mathcal{W}_k \in \mathbb{R}^{R_{k-1} \times I_k \times J_k \times R_k}$ and the matrix $\mathcal{W}_k(i_k, j_k) \in \mathbb{R}^{R_{k-1} \times R_k}$.

Based on the tensor decomposition as shown in Eq. (2.11), given the class of ranks $\{R_0, R_1, ..., R_K\}$, the input tensor $\mathcal{X}$ can be represented as follows:

$$\mathcal{X}(i_1, i_2, ..., i_K) = \prod_{k=1}^{K} \mathcal{X}_k(i_k), \tag{2.13}$$

in which $\mathcal{X}_k \in \mathbb{R}^{R_{k-1} \times I_k \times R_k}$ and $\mathcal{X}_k(i_k) \in \mathbb{R}^{R_{k-1} \times R_k}, \forall i_k \in [I_k]$. Then, Eq. (2.12) can be further derived as:

$$\begin{aligned} \mathcal{Y}(j_1, j_2, ..., j_K) &= \sum_{i_1=1}^{I_1} \cdots \sum_{i_K=1}^{I_K} \left(\prod_{k=1}^{K} \mathcal{W}_k(i_k, j_k) \right) \cdot \prod_{k=1}^{K} \mathcal{X}_k(i_k) \\ &= \prod_{k=1}^{K} \left(\sum_{i_k=1}^{I_k} \mathcal{W}_k(i_k, j_k) \cdot \mathcal{X}_k(i_k) \right) \\ &= \prod_{k=1}^{K} \mathcal{Y}_k(j_k), \end{aligned} \tag{2.14}$$

where $\mathcal{Y}_k(j_k) \in \mathbb{R}^{R_{k-1} \times R_k}$, and thus $\prod_{k=1}^{K} \mathcal{Y}_k(j_k)$ is a scalar because of the ranks $R_0 =$

$R_K = 1$. Eq. (2.14) suggests that K parallel and independent channels are built to compute $\mathcal{Y}(j_1, j_2, ..., j_K)$. To configure a TTN, the ReLU activation is applied to $\mathcal{Y}(j_1, j_2, ..., j_K)$, which is represented as:

$$\hat{\mathcal{Y}}(j_1, j_2, ..., j_K) = \text{ReLU}(\mathcal{Y}(j_1, j_2, ..., j_K)) = \text{ReLU}\left(\prod_{k=1}^{K} \mathcal{Y}_k(j_k)\right). \tag{2.15}$$

The TTN is used in our theoretical work to compactly represent an over-parametrized DNN and explore the empirical performance of tensor-to-vector regression.

CHAPTER 3

ANALYZING REPRESENTATION POWER OF DNN BASED VECTOR-TO-VECTOR REGRESSION

In this chapter, we first analyze the ANN-based vector-to-vector regression and then explore the representation power of DNN architectures. The work has been published in [34].

3.1 Theory on the ANN Based Vector-to-Vector Regression

To begin with, we verify that the ReLU activation defined in ANN can satisfy the condition of Eq. (2.5) in Theorem 3, where given a scalar x, the ReLU function is defined as:

$$\text{ReLU}(x) = \max(0, x). \tag{3.1}$$

Proof. For a given $|z| \geq \epsilon$ and $\tau z \in [0, 1]$, the ReLU activation function can ensure the inequality as:

$$\begin{aligned} |\sigma(\tau z) - 1_{z>0}| &= |\text{ReLU}(\tau z) - 1_{z>0}| \\ &= (1 - \tau z) 1_{\{0 \leq \tau z \leq 1\}} \\ &\leq \min_{|z| \geq \epsilon} \exp(-\tau \epsilon). \end{aligned} \tag{3.2}$$

Based on Eqs. (2.5) and (3.1), given $\epsilon = \frac{\ln \tau}{\tau}$, it yields

$$\delta_\tau \leq \frac{1}{\tau} + \frac{2 \ln \tau}{\tau}. \tag{3.3}$$

Next, we use Eq. (3.2) and choose the parameter τ as $\sqrt{J} \ln J$, the upper bound in Eq.

Algorithm 1 Iterative Approximation [41]

1. Input: A bounded set $\mathbb{F}$, and a target function $f \in \mathbb{F}$.
2. Initialize an arbitrary function $f_0 \in \mathbb{F}$.
3. For $t = 1, 2, ..., T$:
4. Choose the pair (α_t, g_t) to solve
5. $\min_{\alpha \in [0,1], g \in G} ||f - (\alpha f_{t-1} + (1-\alpha)g)||_2^2$.
6. Update $f_t := \alpha_t f_{t-1} + (1-\alpha_t) g_t$.

(2.4) becomes

$$\begin{aligned} 2C\left(\frac{1}{\sqrt{J}} + \delta_\tau\right) &\leq 2C\left(\frac{1}{\sqrt{J}} + \frac{1}{\tau} + \frac{2\ln\tau}{\tau}\right) \\ &\leq 2C\left(\frac{2}{\sqrt{J}} + \frac{2\ln(\ln J)}{\sqrt{J}\ln J} + \frac{1}{\sqrt{J}\ln J}\right) \\ &= O\left(\frac{1}{\sqrt{J}}\right), \end{aligned} \tag{3.4}$$

which justifies the upper bound in the universal approximation theory. □

Furthermore, Barron's universal approximation theory can be generalized to the scenario of the vector-to-vector regression as shown in Theorem 4, and an iterative algorithm is deployed to demonstrate how an ANN achieves the derived upper bound.

Theorem 4. *Given a continuous vector-to-vector regression operator $h_{\mathcal{D}}^* : [0,1]^D \to \mathbb{R}^Q$, we can find an approximate operator $f_{\mathcal{D}}^*$ with Q functions $f_{\mathcal{D}}^* = [f_{\mathcal{D},1}^*, f_{\mathcal{D},2}^*, ..., f_{\mathcal{D},Q}^*]$, where each function $f_{\mathcal{D},i}^* : [0,1]^D \to \mathbb{R}$ with J Sigmoid or ReLU activation functions such that*

$$||h_{\mathcal{D}}^* - f_{\mathcal{D}}^*||_1 = \mathcal{O}\left(\frac{Q}{\sqrt{J}}\right). \tag{3.5}$$

Proof. Suppose that the regression operator $h_{\mathcal{D}}^*$ denotes a Q-dimensional functional $h_{\mathcal{D}}^* = [h_{\mathcal{D},1}^*, h_{\mathcal{D},2}^*, ..., h_{\mathcal{D},Q}^*]$, where each function $h_{\mathcal{D},i}^* : \mathbb{R}^D \to \mathbb{R}$. Thus, based on Theorem 3, we

obtain Eq. (3.5) as:

$$||h^*_{\mathcal{D}} - f^*_{\mathcal{D}}||_1 = \sum_{i=1}^{Q} |h^*_{\mathcal{D},i} - f^*_{\mathcal{D},i}| = \sum_{i=1}^{Q} \mathcal{O}\left(\frac{1}{\sqrt{J}}\right) = \mathcal{O}\left(\frac{Q}{\sqrt{J}}\right).$$

□

Theorem 4 suggests that the operator f corresponds to an ANN-based vector-to-vector regression, and the upper bound in Eq. (3.4) in Theorem 4 implies that the representation power of an ANN is essentially controlled by the width of the hidden layers.

The back-propagation (BP) algorithm [60] based on SGD [61] is applied to update ANN parameters, we consider whether the SGD can achieve the bound in Eq. (3.5). We first introduce an iterative approximation algorithm proposed by Barron [41] that can realize the approximation bound as Eq. (2.4) in Theorem 3 by alternatively solving a minimization problem concerning α and g in Step 5 of Algorithm 1. The minimizer g and α obtained in Step 5 are then used to iteratively update f in Step 6. Furthermore, if f corresponds to a parametric ANN, the update of f_t refers to the update of the related parameters w and b at time t. We assume that g represents the gradient of f, α is a learning rate, and the bounded set $\mathbb{F}$ is defined as the set of functions represented by an J-node ANN in Eq. (3.6). Then, Algorithm 1 becomes the BP algorithm with a momentum [62].

$$\mathbb{F} = \{f_{\mathbf{W},\mathbf{b}} : \mathbb{R}^D \to \mathbb{R} | \mathbf{W} \in \mathbb{R}^{J \times D}, \mathbf{b} \in \mathbb{R}^J\}. \tag{3.6}$$

Corollary 1. *Given an input domain $[0,1]^D$, the ReLU based hidden layer is a convex but not smooth and not strongly convex function. Thus, the SGD algorithm for updating the ReLU based hidden layer requires $\Theta(\frac{1}{\epsilon^2})$ iterations for an ϵ-optimal solution.*

Proof. It has been known that the ReLU activation function is a convex but non-smooth

function with the inequality as:

$$\begin{aligned} f(\mathbf{x}_t) - f(\mathbf{x}^*) &\leq \nabla f(\mathbf{x}_t)^\top (\mathbf{x}_t - \mathbf{x}^*) \\ &\leq \frac{1}{\eta}(\mathbf{x}_t - \mathbf{x}_{t+1})^\top (\mathbf{x}_t - \mathbf{x}^*) \\ &\leq \frac{1}{\eta}(||\mathbf{x}_t - \mathbf{x}^*||_2^2 + ||\mathbf{x}_t - \mathbf{x}_{t+1}||_2^2 - ||\mathbf{x}_{t+1} - \mathbf{x}^*||_2^2) \\ &\leq \frac{1}{2\eta}(||\mathbf{x}_t - \mathbf{x}^*||_2^2 - ||\mathbf{x}_{t+1} - \mathbf{x}^*||_2^2) + \frac{\eta}{2}||\nabla f(\mathbf{x}_t)||_2^2, \end{aligned} \tag{3.7}$$

where $\mathbf{x}^*$ denotes the optimal point, $\mathbf{e}_t$ refers to the sub-gradient of the point $\mathbf{x}_t$, η is the learning rate.

Summing up the resulting inequality over T epochs, and using that $||\mathbf{x}_t - \mathbf{x}^*|| \leq R$ and the sub-gradient of the modified ReLU $||\nabla f(\mathbf{x}_t)||_2^2 \leq 1$ yield a regret Eq. (3.8) at time T as:

$$Regret_T = \sum_{t=1}^{\top} (f(\mathbf{x}_t) - f(\mathbf{x}^*)) \leq \frac{R^2}{2\eta} + \frac{\eta T}{2}. \tag{3.8}$$

By taking $\eta = \frac{R}{\sqrt{T}}$, we obtain

$$Regret_T \leq R\sqrt{T}. \tag{3.9}$$

On the other hand,

$$f\left(\frac{1}{T}\sum_{t=1}^{\top} \mathbf{x}_t\right) - f(\mathbf{x}^*) \leq \frac{1}{T}\sum_{t=1}^{\top} (f(\mathbf{x}_t) - f(\mathbf{x}^*)) \leq \frac{R}{\sqrt{T}}. \tag{3.10}$$

For an ϵ-optimal, we set $\frac{R}{\sqrt{T}} = \epsilon$ so that we obtain $T = \Theta(\frac{1}{\epsilon^2})$ for an ϵ-optimal solution.

□

Corollary 2. *A Sigmoid or ReLU hidden layer is a β-smooth but not a convex function. Thus, the SGD algorithm ensures that it takes $\Theta(\frac{\beta}{\epsilon})$ iterations for an ϵ-optimal solution.*

Proof. To prove Corollary 2, we define a continuously differential function f is β-smooth

if ∇f is β-Lipschitz, which is

$$||\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})||_2 \leq \beta ||\mathbf{x} - \mathbf{y}||_2. \tag{3.11}$$

In addition, let f be a β-smooth function on $\mathbb{R}^D$. Then for any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^D$, one has

$$\begin{aligned}
&|f(\mathbf{x}) - f(\mathbf{y}) - \nabla f(\mathbf{y})^\top (\mathbf{x} - \mathbf{y})| \\
&= \left| \int_0^1 \nabla f(\mathbf{y} + t(\mathbf{x} - \mathbf{y}))^\top (\mathbf{x} - \mathbf{y}) dt - \nabla f(\mathbf{y})^\top (\mathbf{x} - \mathbf{y}) \right| \\
&\leq \int_0^1 ||\nabla f(\mathbf{y} + t(\mathbf{x} - \mathbf{y})) - \nabla f(\mathbf{y})||_2 \cdot ||\mathbf{x} - \mathbf{y}||_2 dt \\
&\leq \int_0^1 \beta t ||\mathbf{x} - \mathbf{y}||_2^2 dt \\
&= \frac{\beta}{2} ||\mathbf{x} - \mathbf{y}||_2^2.
\end{aligned} \tag{3.12}$$

By taking $\mathbf{x} = \mathbf{x}_{t+1}$, $\mathbf{y} = \mathbf{x}_t$, and let f represent a Sigmoid or ReLU function, we obtain:

$$\begin{aligned}
f(\mathbf{x}_{t+1}) &\leq f(\mathbf{x}_t) + \nabla f(\mathbf{x}_t)^\top (\mathbf{x}_{t+1} - \mathbf{x}_t) + \frac{\beta}{2} ||\mathbf{x}_t - \mathbf{x}_{t+1}||_2^2 \\
&\leq f(\mathbf{x}_t) - \eta ||\nabla f(\mathbf{x}_t)||_2^2 + \frac{\beta \cdot \eta^2}{2} ||\nabla f(\mathbf{x}_t)||_2^2.
\end{aligned} \tag{3.13}$$

Summing up the resulting inequality over t, we obtain Eq. (3.14), where we set the learning rate $\eta = \frac{1}{\beta}$.

$$\begin{aligned}
E[||\nabla f(\mathbf{x})||_2^2] &= \frac{1}{T} \sum_{t=1}^{\top} ||\nabla f(\mathbf{x}_t)||_2^2 \\
&= \frac{2(f(\mathbf{x}_1) - f(\mathbf{x}_{T+1}))}{\eta(2 - \eta\beta)T} \\
&\leq \frac{2\beta(f(\mathbf{x}_1) - f(\mathbf{x}_{T+1}))}{T}
\end{aligned} \tag{3.14}$$

which suggests: $E[||\nabla f(\mathbf{x})||_2^2] = \mathcal{O}(\frac{\beta}{T})$. Moreover, by setting

$$\frac{2\beta(f(\mathbf{x}_1) - f(\mathbf{x}_{T+1}))}{T} = \epsilon \tag{3.15}$$

which suggests that $T = \Theta(\frac{\beta}{\epsilon})$ for an ϵ-optimal solution. $\square$

By comparing the convergence rates in corollaries 1 and 2, the SGD algorithm for an ANN with ReLU based hidden layer ensures a faster rate because $\Theta(\frac{1}{\epsilon^2})$ is smaller than $\Theta(\frac{\beta}{\epsilon})$ for all $\beta > \frac{1}{\epsilon}$. Furthermore, some new optimization algorithms, e.g. root mean square propagation (RMSProp) [63], adaptive gradient (AdaGrad) [64], and adaptive moment estimation (Adam) [65], are the SGD extensions to accelerate the training process. However, it is not clear if the optimization algorithms can achieve the Barron's bound. As a result, only the SGD algorithm is considered in this chapter.

3.2 Theory on the DNN Based Vector-to-Vector Regression

This section establishes a connection between the depth of a DNN and the representation power of vector-to-vector regression operators. We discuss whether the DNN expressive power can benefit from the increment of depth in terms of the number of hidden layers. Theorem 5 suggests that a deeper DNN architecture can result in a lower upper bound. On the other hand, Theorem 5 implies that a larger input dimension D or a larger output dimension Q can result in a higher upper bound.

Theorem 5. *Let $h^*_{\mathcal{D}} : [0,1]^D \rightarrow \mathbb{R}^Q$ be a target smooth function, we can find a feed-forward DNN $f^*_{\mathcal{D}}$ with K ReLU based hidden layers ($K \geq 2$), where the width of each hidden layer is at least $D + 2$. The function $f^*_{\mathcal{D}}$ can be approximated with an upper bound as:*

$$||h^*_{\mathcal{D}} - f^*_{\mathcal{D}}||_1 = \mathcal{O}((K-1)^{-\frac{r}{D}}), \tag{3.16}$$

where r depends on the maximum value of the first K derivatives of $h^*_{\mathcal{D}}$.

Proof. Before we demonstrate Theorem 5, we first introduce Lemma 1 and Lemma 2. Lemma 1 is based on Theorem 1 in [66], and Lemma 2 is derived from [67].

Lemma 1 (Mhaskar and Poggio's approximation theory [66]). *For a smooth function $h^*_{\mathcal{D}}$: $\mathbb{R}^D \rightarrow \mathbb{R}^Q$, there exists a ReLU based ANN $f^*_{\mathcal{D}}$ with a hidden layer of K units and a constant $C_{h^*_{\mathcal{D}}}$ which relies on the maximum value of the first K derivatives of $f^*_{\mathcal{D}}$. Then, we can find an integer $r \geq 1$, so that there is a constraint Eq. (3.17) where $\mathcal{D}^K f^*_{\mathcal{D}}$ denotes a vector of derivatives as $[\nabla f^*_{\mathcal{D}}, \nabla^2 f^*_{\mathcal{D}}, ..., \nabla^K f^*_{\mathcal{D}}]$.*

$$||f^*_{\mathcal{D}}||_\infty + \sum_{K, 1 \leq \frac{K(K-1)}{2} \leq r} ||\mathcal{D}^K f^*_{\mathcal{D}}||_\infty \leq C_{h^*_{\mathcal{D}}}, \tag{3.17}$$

such that we attain Eq. (3.18) as:

$$||h^*_{\mathcal{D}} - f^*_{\mathcal{D}}||_1 = \mathcal{O}(K^{-\frac{r}{D}}). \tag{3.18}$$

Lemma 2 (Hanin's universal function approximation [67]). *Let $h^*_{\mathcal{D}} : \mathbb{R}^D \rightarrow \mathbb{R}^Q$ be a ReLU based ANN with input dimension D and one hidden layer of width $K (K \geq 1)$. There exists another ReLU based DNN $f^*_{\mathcal{D}}$, which owns an input dimension D and $(K + 1)$ hidden layers with the width $(D + 2)$ for each hidden layer that results in the same result as $f^*_{\mathcal{D}}$.*

Proof. Assume a vector $\mathbf{A}^{(k)} = \{A^{(k)}_1, A^{(k)}_2, ..., A^{(k)}_{n_k}\}$ as the output of the k-th hidden layer of the width $J_k = D + 2$ based on the ReLU activation function. Then, we can derive Eq. (3.19) and Eq. (3.20) as:

$$\mathbf{A}^{(k+1)} = \text{ReLU}\left(\mathbf{b}^{(k)} + \sum_{j=1}^{J_k} \mathbf{w}^{(k)}_j A^{(k)}_j\right), \tag{3.19}$$

$$f^*_{\mathcal{D}} = \mathbf{b}^{(k+1)} + \sum_{l=1}^{J_k+1} \mathbf{w}^{(k+1)}_l A^{(k+1)}_l. \tag{3.20}$$

Then, we know that $h^*_{\mathcal{D}}$ can be approximated by $\mathbf{A}^{(k+1)}$, which implies that $h^*_{\mathcal{D}}$ can be guaranteed to be approximated by $f^*_{\mathcal{D}}$ by separately setting the values of $\mathbf{b}^{(k+1)}$ and $\mathbf{w}^{(k+1)}_j$ as 0 and $\frac{1}{J_{k+1}}$. □

Finally, by applying Lemma 2 to Lemma 1, we can find a ReLU based DNN $f_{\mathcal{D}}^*$ with K hidden layers that can be represented by an ANN with a single hidden layer of $(K-1)$ units. Therefore, we can attain

$$||h_{\mathcal{D}}^* - f_D^*||_1 \leq \frac{C_f}{(K-1)^{\frac{r}{D}}} = \mathcal{O}\left(\frac{1}{(K-1)^{\frac{r}{D}}}\right).$$

□

Theorem 5 suggests that the asymptotic upper bound relies on the depth of hidden layers K, the input dimension D, and the output dimension Q. For a fixed pair (D, Q), a tighter upper bound can be obtained by setting a larger K. Besides, the width of hidden layers must have at least $(D+2)$ units to obtain the bound in Eq. (3.16). In other words, a deeper DNN architecture corresponds to a better expressive power for the target operator $h_{\mathcal{D}}^*$.

Although Theorem 5 implies that an upper bound on DNN based vector-to-vector regression depends on the depth K of a DNN architecture, we also explore the relationship between the width of hidden layers and upper bound on the representation power. Theorem 6 revises the upper bound in Theorem 5, where both depth and width are jointly taken into account.

Theorem 6. *For a target regression smooth function $h_{\mathcal{D}}^* : \mathbb{R}^D \to \mathbb{R}^Q$, there exists a DNN $f_{\mathcal{D}}^*$ with $K(K \geq 2)$ ReLU based hidden layers, where the width of each hidden layer is at least $(D+2)$ and the top hidden layer owns $J_K(J_K \geq D+2)$ units. For an integer $r \geq 1$ associated with the maximum value of continuous derivatives of $h_{\mathcal{D}}^*$, we can obtain*

$$||h_{\mathcal{D}}^* - f_{\mathcal{D}}^*||_1 = \mathcal{O}\left(\frac{Q}{(J_K + K - 1)^{\frac{r}{D}}}\right). \tag{3.21}$$

Proof. As in the proof of Lemma 2 in [67], the first D hidden nodes in each hidden layer before the last, are scaled and shifted the exact copies of the input; the $(D+1)$-th node in each hidden layer computes a new ReLU function of the input, and the $(D+2)$-th node

computes the accumulation of all of the ReLU functions computed by layers thus far. In that case, the entire network acts like an ANN with $J_K + (K-1)$ hidden nodes. Based on Lemma 1, the approximation error is upper bounded by $\mathcal{O}\left(\frac{Q}{(J_K+K-1)^{\frac{r}{D}}}\right)$. □

Discussion: Our theorem on DNN based vector-to-vector regression relies on an assumption that the target function is smooth, whereas a continuous target function is assumed for the classical universal approximation. Next, we first compare the property of smoothness and continuity for a target function, and then we connect our theorem to the classical universal approximation when the continuous property is considered.

Lemma 3. *For a continuous function $f : \mathbb{R}^D \to \mathbb{R}$, given a small value $\epsilon_1, \epsilon_2 > 0$, there exists a value $\delta > 0$ such that for a fixed point $\boldsymbol{x}_0$, $\forall \boldsymbol{x} \in (\boldsymbol{x}_0 - \delta\mathbf{1}, \boldsymbol{x}_0 + \delta\mathbf{1})$, the value of $f(\boldsymbol{x})$ satisfies*

$$f(\boldsymbol{x}_0) + \epsilon_1 \leq f(\boldsymbol{x}) \leq f(\boldsymbol{x}_0) + \epsilon_2. \tag{3.22}$$

Definition 1. *Let f be a β-smooth function $f : \mathbb{R}^D \to \mathbb{R}$, then $\forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^D$, we have*

$$\left| f(\boldsymbol{x}) - f(\boldsymbol{y}) - \nabla f(\boldsymbol{y})^\top (\boldsymbol{x} - \boldsymbol{y}) \right| \leq \frac{\beta}{2} ||\boldsymbol{x} - \boldsymbol{y}||_2^2. \tag{3.23}$$

Based on Lemma 3 and Definition 1, we will show that the smoothness is a stronger condition than the continuity. In other words, a smooth function can lead to a continuous one, but the continuous function cannot result in a smooth one. That is because Eq. (3.23) can bring about Eq. (3.24) as:

$$f(\mathbf{y}) + \nabla f(\mathbf{y})^\top (\mathbf{x} - \mathbf{y}) - \frac{\beta}{2} ||\mathbf{x} - \mathbf{y}||_2^2 \leq f(\mathbf{x}) \leq f(\mathbf{y}) + \nabla f(\mathbf{y})^\top (\mathbf{x} - \mathbf{y}) + \frac{\beta}{2} ||\mathbf{x} - \mathbf{y}||_2^2. \tag{3.24}$$

Then, by defining

$$\epsilon_1 = \nabla f(\mathbf{y})^\top (\mathbf{x} - \mathbf{y}) - \frac{\beta}{2} ||\mathbf{x} - \mathbf{y}||_2^2,$$

$$\epsilon_2 = \nabla f(\mathbf{y})^\top (\mathbf{x} - \mathbf{y}) + \frac{\beta}{2} ||\mathbf{x} - \mathbf{y}||_2^2,$$

we can obtain that

$$f(\mathbf{y}) + \epsilon_1 \leq f(\mathbf{x}) \leq f(\mathbf{y}) + \epsilon_2,$$

which is consistent with the definition of a continuous function. However, the property of continuity cannot derive the property of smoothness.

Furthermore, we analyze the theorem for DNN based vector-to-vector regression when the target function is continuous, which is shown in Theorem 7.

Theorem 7. *For a target continuous regression function $h_{\mathcal{D}}^* : \mathbb{R}^D \rightarrow \mathbb{R}^Q$, there exists a feed-forward DNN $f_{\mathcal{D}}^*$ with the ReLU activation and K hidden layers whose width having at least $D + 2$ neurons, such that for a small $\epsilon > 0$,*

$$||h_{\mathcal{D}}^* - f_{\mathcal{D}}^*||_1 \leq \epsilon.$$

Proof. Theorem 7 can be derived from the Lemma 6 in [67]. It means that a ReLU ANN with a single hidden layer of width K. There exists another ReLU DNN $f_{\mathcal{D}}^* : \mathbb{R}^D \rightarrow \mathbb{R}^Q$ that computes the same function as ANN, but it has $K + 2$ hidden layers with the width at least $D + 2$. □

Theorem 7 connects our theorem to the universal approximation theory when a regression target function is assumed to be continuous. Since the smooth target function is always taken as the objective function, Theorem 6 is utilized in our experiments.

3.3 Estimation of Mean Squared Error (MSE) Upper Bounds

MSE is usually taken as the loss function for training DNN based vector-to-vector regression operator. In this section, we discuss how to make use of our derived theorems to

estimate MSE practical values of the DNN based vector-to-vector regression models in our experiments of speech enhancement.

Proposition 1 generalizes our theoretical upper bounds to practical upper bounds, where the amount of training data N and the input dimension D need to be considered.

Proposition 1. *For a target function $h^*_{\mathcal{D}} : \mathbb{R}^D \rightarrow \mathbb{R}^Q$, we can use N training samples to attain an ANN $\bar{f}_S$ with J Sigmoid or ReLU based activation functions such that the evaluation loss of MSE can be upper bounded as:*

$$\textit{MSE}(\bar{f}_S, h^*_{\mathcal{D}}) = \mathcal{O}\left(\frac{Q}{J}\right) + \mathcal{O}\left(\frac{QJD}{N}\log N\right) + \nu, \tag{3.25}$$

where ν represents a constant approximation error from the non-deterministic randomness of the input noise.

Proposition 1 focuses on the representation power and thus do not consider the generalization power associated with the estimation error. Besides, Proposition 1 also suggests that given two constants c_1 and c_2, MSE$(\bar{f}_S, h^*_{\mathcal{D}})$ can be upper bounded as:

$$\text{MSE}(\bar{f}_S, h^*_{\mathcal{D}}) \leq c_1\frac{Q}{J} + c_2\frac{QJD}{N}\log N + \nu. \tag{3.26}$$

For f_1 with J_1 neuron units and f_2 with J_2 neuron units, if Eq. (3.27), Eq. (3.28), and Eq. (3.29) are jointly satisfied, we can separately derive Eq. (3.30) and Eq. (3.31) as:

$$\epsilon_1 \leq MSE(h^*_{\mathcal{D}}, f_1) \leq c_1\frac{Q}{J_1} + c_2\frac{QJ_1D}{N}\log N + \nu, \tag{3.27}$$

$$\epsilon_2 \leq MSE(h^*_{\mathcal{D}}, f_2) \leq c_1\frac{Q}{J_2} + c_2\frac{QJ_2D}{N}\log N + \nu, \tag{3.28}$$

$$c_1\frac{Q}{J_1} + c_2\frac{QJ_1D}{N}\log N + \nu \geq c_1\frac{Q}{J_2} + c_2\frac{QJ_2D}{N}\log N + \nu, \tag{3.29}$$

$$c_1 \geq \frac{J_1 J_2(\epsilon_1 + \epsilon_2 - 2\nu)}{2Q(J_1 + J_2)} = \hat{c}_1, \tag{3.30}$$

$$0 < c_2 \leq \frac{Nc_1}{J_1 J_2 D \log N}, \quad \hat{c}_2 = \frac{N\hat{c}_1}{J_1 J_2 D \log N}. \tag{3.31}$$

In practice, some factors such as the amount of training data and the dimensions of some hidden layers, are necessarily taken into account, which results in our Proposition 2.

Proposition 2. *For a target operator $h^*_{\mathcal{D}} : \mathbb{R}^D \to \mathbb{R}^Q$, we can use N training samples to obtain a $\bar{f}_S$ with $K(K \geq 2)$ ReLU based hidden layers, where the width of each hidden layer is at least $(D + 2)$. For an integer $r > 1$ that is associated with the maximum value of derivatives of $h^*_{\mathcal{D}}$, the MSE loss is bounded as:*

$$MSE(\bar{f}_S, h^*_{\mathcal{D}}) = \mathcal{O}\left(\frac{Q}{(K - 1 + J_K)^{\frac{2r}{D}}}\right) + \mathcal{O}\left(\frac{QJ_K J_{K-1}}{N} \log N\right) + \nu, \tag{3.32}$$

where J_K and J_{K-1} separately denote the numbers of hidden neurons for the K-th and $(K - 1)$-th hidden layers, and ν refers to a constant approximation error arose from a non-deterministic input noise.

Proof. For a DNN with $K(K \geq 2)$ hidden layers, we regard the bottom $(K - 2)$ hidden layers as a feature extractor for the top hidden layer which can be taken as the input to the top hidden layer. By combining Eq. (3.16) in Theorem 5 with Eq. (3.25) in Proposition 1, we derive Eq. (3.32) in Proposition 2. □

Furthermore, Proposition 2 suggests that there exist two constants a_1 and a_2, which results in the following inequality as:

$$MSE(\bar{f}_S, h^*_{\mathcal{D}}) \leq \frac{a_1 Q}{(J_K + K - 1)^{\frac{2r}{D}}} + \frac{a_2 QJ_K J_{K-1}}{N} \log N + \nu. \tag{3.33}$$

Eq. (3.33) suggests that the term $\frac{Q}{(J_K+K-1)^{\frac{2r}{D}}}$ solely relies on the depth, and a deeper

DNN architecture corresponds to a lower upper bound on MSE. Therefore, we can separately set a_1 and a_2 as c_1 and c_2 because the depth of DNN does not impose an additional restriction any more.

Furthermore, $\hat{c}_1$ in Eq. (3.30) and $\hat{c}_2$ in Eq. (3.31) are associated with the minimum estimated MSE values, which correspond to the attained values by SGD. However, a vanilla SGD without the use of some optimization tricks since the technique of dropout generally cannot ensure a closely approximated solution to the global point. Hence, we set an MSE upper bound by setting $c_1 = \hat{c}_1$ in Eq. (3.30) and $c_2 = \frac{N\hat{c}_1}{J_1 J_2 D \log N}$ in Eq. (3.31) to reduce an implicit optimization bias and keep MSE upper bounds as minimum as possible.

As for the setup ϵ_1 and ϵ_2 for the computation of c_1 and c_2, ReLU based ANN can ensure the minimum MSE because of the property of convexity. Thus, we can set ϵ_1 and ϵ_2 as the empirical MSE values of two ReLU based ANN models. Besides, we set the integer r in Eq. (3.33) as D, which keeps the same as the input dimensions. Consequently, the estimation of MSE upper bound can be modified as:

$$MSE(\bar{f}_S, h^*_{\mathcal{D}}) \leq \frac{a_1 Q}{(J_K + K - 1)^2} + \frac{a_2 Q J_K J_{K-1}}{N} \log N + \nu. \tag{3.34}$$

Finally, the configuration for ν varies from different noise types of various SNR levels. Practically, we set the values of ν as 0.1.

3.4 Experiments of Speech Enhancement

3.4.1 Experimental Goals

We now discuss deep learning for speech enhancement with particular attention to linking experimental outcomes with the theorems presented in the previous sections. DNN generalization capability of the vector-to-vector regression has been empirically justified in our earlier efforts [68, 69]. Thus, the present work mainly discusses the expressive power but not the generalization problem and over-fitting problems, which implies that we would not

use very large neural architectures and focus on matched noisy conditions. More specifically, we aim at verifying the following aspects:

- The expressive power of the ANN-based vector-to-vector regression function can be enhanced by enlarging the width of the hidden layer.
- The depth of a DNN can contribute to the improvement of the expressive power of the vector-to-vector regression.
- The above properties can be consistently maintained and verified in various noisy conditions and SNR levels.
- Although the depth and width are two joint parameters affecting the expressive power of vector-to-vector regression, a top hidden layer with a broader width for a deeper DNN architecture contributes to a better expressive capability. This property has also been experimentally verified in [70, 71, 72, 73], where the bottleneck features extracted from a layer closer to the output led to a better abstract representation of the original speech features.

3.4.2 Experimental Setup

The DNN used for speech enhancement is a feed-forward ANN, where inputs were the normalized log-power spectral feature vectors [71] of noisy speech, and outputs referred to the feature vectors of clean or enhanced speech. The reference of clean speech feature vectors associated with the noisy one was assigned to the top layer of DNN in the training process, but the top layer of DNN corresponds to the feature vectors of the enhanced speech during the testing stage. The Sigmoid and ReLU functions were used for hidden layers of neural networks, whereas a linear activation function was used in the output layer for the vector-to-vector regression. Global variance equalization [74] was used to alleviate the problem of over-smoothing by correcting the global variance between estimated feature

vectors and clean reference targets. During the DNN training process, the standard back-propagation algorithm (BP) [75] with MSE was adopted to measure the difference between a normalized log-power spectral feature vector, and the reference one. To enable non-stationary noise awareness, the technique of noise-aware training (NAT) [76] was employed to generate high-dimensional feature vectors of the length of 3 frames via concatenating frames within a sliding window. Moreover, the SGD algorithm with a learning rate of 1×10^{-2} and a momentum rate of 0.4 was used for the update of parameters.

The clean dataset was obtained from the TIMIT speech corpus [77], where 4620 utterances were used for training, and 1600 utterances were selected for testing. Two types of noises, namely M109 and Babble, from the Noise-92 dataset [78] were chosen for synthesizing the noisy training and testing samples at SNR levels of 5dB, and 15dB. The M109 noise is stationary and is collected from the engine of tanks. The Babble noise is more challenging because it involves a mixture of multiple speakers. Since we are interested in assessing the DNN based vector-to-vector expressive power, concerning the theorem discussed in previous chapters, we have deliberately built and evaluated DNN architectures of speech enhancement based on training and testing data covering the same noise types and SNR levels. For example, if a DNN model was trained with noisy speech material corrupted by the Babble noise with an SNR of 15dB, the DNN model would be evaluated with the test data having the same characteristics in terms of noise types and SNR values. Besides, all clean and noise waveforms were downsampled to 8KHz. Both frame and the shift length were separately set to 32 msec and 16 msec, which correspond to 256 samples and 128 samples, respectively. Therefore, the dimension of one feature vector is 257 which involves an additional dimension for the log-power feature. To improve the robustness against noises, long-term features were applied by separately connecting 3 left-and-right neighbors of each frame, which resulted in a dimension of 771. The feature values were further processed by using a mean and variance normalization before they were fed to the DNN inputs. Besides, two evaluation criteria, namely MSE and the perceptual evaluation

of speech quality (PESQ) [79], were employed in our experimental validation.

3.4.3 An Evaluation of the Expressive Power of Layered ANNs

We here present experimental results on speech enhancement by comparing different neural network architectures obtained by varying width and depth of the hidden layers. Table 4.1 lists the model architectures in our experiments, where the structures (the dimension in each layer) follow an order of Input $\rightarrow$ hidden layer $0 \rightarrow$ hidden layer $1 \rightarrow \cdots \rightarrow$ hidden layer $K \rightarrow$ Output.

Table 3.1: Model structures for various vector-to-vector regression models

Model	Structure (Input – hidden_layers – Output)
ANN1 (ReLU)	771-800-257
ANN2 (ReLU)	771-1600-257
ANN1 (Sigmoid)	771-800-257
ANN2 (Sigmoid)	771-1600-257
DNN1 (ReLU)	771-800-800-800-1600-257
DNN2 (ReLU)	771-800-800-800-800-800-1600-257
DNN3 (ReLU)	771-800-800-800-800-800-800-257
DNN4 (ReLU)	771-800-800-800-800-1600-800-257

As shown in Table 3.1, we first compare the regression performance of an ANN with a narrower and broader width. The width of the hidden layer of ANN1 was set equal to 800, which is based on the unit constraint for the hidden layers in Theorem 5 ($D = 771$, $D + 2 = 773 < 800$); whereas, ANN2 had a hidden layer of 1600 neuron units. Next, we studied vector-to-vector regression by increasing the number of hidden layers of DNN1. As shown in Table 3.1, DNN1 had four hidden layers with widths 800-800-800-1600. Two additional hidden layers of width 800 were further appended to DNN2, which resulted in a deeper six hidden layers 800-800-800-800-800-1600.

Table 3.4 shows the experimental results of different neural network architectures. The evaluation of speech enhancement in terms of both MSE and PESQ measures was con-

ducted in a straightforward noisy condition (M109) with a high SNR level ($15dB$). The results show that ANN2 with a broader width can outperform ANN1 with a narrower width, and DNN2 with six hidden layers achieves better results than DNN1 with four hidden nonlinear layers. Moreover, both DNN1 and DNN2 with deeper architectures can result in better regression performance.

Table 3.2: The setup of hyper-parameters for the estimation of MSE upper bounds.

Noises	M109(15dB)	M109(5dB)	Babble(15dB)	Babble(5dB)
ϵ_1	0.2242	0.3977	0.3189	0.4323
ϵ_2	0.2050	0.3607	0.3073	0.3829
l_1	800	800	800	800
l_2	1600	1600	1600	1600
N	5.43×10^8	5.43×10^8	5.43×10^8	5.43×10^8
$\hat{c}_1$	0.2378	0.4422	0.5794	0.6383
$\hat{c}_2$	0.0065	0.0121	0.0159	0.0175
v	0.1	0.1	0.1	0.1

Besides, we estimate the MSE upper bounds based on Eq. (3.27) for ANNs and Eq. (3.34) for DNNs. We also assume that the ReLU based ANNs can achieve ϵ_1 and ϵ_2 based on Eq. (3.28) and Eq. (3.29) by taking

$$\epsilon_1 = \text{Estimated\_MSE (ReLU)} = \text{ANN1 (ReLU)}, \tag{3.35}$$

$$\epsilon_2 = \text{Estimated\_MSE (ReLU)} = \text{ANN2 (ReLU)}. \tag{3.36}$$

Then, we can compute $\hat{c}1$ and $\hat{c}_2$ based on the estimated ϵ_1 and ϵ_2, where the other hyper-parameters for the estimation of MSE upper bounds can be found in Table 3.2. Based on Table 3.3, the results suggest that our estimated MSE (Estimated_MSE) can offer rational upper bounds for DNN based models, but they cannot ensure rational upper bounds for Sigmoid based ANNs because of the non-convexity of Sigmoid functions. Overall, the

experimental results support our theoretical analysis.

Table 3.3: The evaluation results under the M109 noise of SNR 15dB.

Models	MSE	PESQ	Estimated_MSE
ANN1 (ReLU)	0.2242	2.74	0.2242
ANN2 (ReLU)	0.2050	2.77	0.2050
ANN1 (Sigmoid)	0.2332	2.73	0.2146
ANN2 (Sigmoid)	0.2198	2.75	0.2146
DNN1 (ReLU)	0.1662	2.84	0.1793
DNN2 (ReLU)	0.1412	2.86	0.1755

3.4.4 Evaluating the Width at the Top Hidden Layer of DNN

We now analyze the effects of the width of the top hidden layer in a DNN. Although we observe that the width of hidden layers and depth of the neural architecture are two joint factors affecting the expressive power of the DNN based vector-to-vector regression function, it is expected that a broader width at the top of the hidden layer can achieve better regression results based on Theorem 6. Therefore, we compared three DNNs with architectures as shown in Table 3.1, where DNN3 corresponds to a structure of 800-800-800-800-800-800 and the architecture of DNN4 is set up as 800-800-800-800-1600-800.

Table 3.4 shows the results for those three DNNs. It is observed that the top hidden layer with a broader width corresponds to lower MSE and higher PESQ, which suggests that the configuration of a broader width at the top hidden layer is essential to allow for a better expressive power of a DNN based vector-to-vector regression function. However, a broader hidden layer in the middle of DNN cannot contribute to a better result, which is matched with our estimated MSE for DNN4 in Table 3.4, where we also verify the estimated MSE upper bounds for DNN3 are consistent with the results.

Table 3.4: A comparison of the expressive power among DNN2 (800-800-800-800-800-1600), DNN3 (800-800-800-800-800-800), and DNN4 (800-800-800-800-1600-800) under M109 noise of SNR 15dB

Models	MSE	PESQ	Estimated_MSE
DNN2 (ReLU)	0.1412	2.86	0.1755
DNN3 (ReLU)	0.1557	2.84	0.1578
DNN4 (ReLU)	0.1598	2.82	0.1794

3.4.5 Empirical Assessment in Adverse Noisy Conditions

So far, we have analyzed the expressive power of the DNN based vector-to-vector regression function in favorable noisy conditions. In this section, we further evaluated the related expressive power under some noisy adverse conditions. Table 3.5 shows the regression results under a complicated Babble noisy condition. Table 3.6 and Table 3.7 separately list the regression results in the adverse noisy conditions at a low SNR level. We observed that all the conclusions of the DNN based vector-to-vector regression are still valid in the adverse noisy environments at a low SNR level, although the performance becomes worse in such conditions. However, we have only tested on a complicated noisy condition, yet the resulting property can be regarded as a general case because the Babble noise is a typical and one of the most complicated noises in practice.

In adverse conditions, the estimated MSE values based on Eq. (3.27) and Eq. (3.35) are separately shown in Table 3.5, Table 3.6 and Table 3.7. By comparison, the estimated MSE upper bounds provide the rational estimation to the real empirical MSE in all cases except Sigmoid-based ones.

3.4.6 Experimental Summary

The empirical regression results discussed in the previous sections confirm our theoretical claims. More specifically, the experimental results verify that an ANN with a broader hidden width outperforms the one with a narrower width, and a deeper architecture contributes

Table 3.5: Evaluating Results under the Babble noise of SNR 15dB

Models	MSE	PESQ	Estimate_MSE
ANN1 (ReLU)	0.3189	2.65	0.3189
ANN2 (ReLU)	0.3073	2.68	0.3073
ANN1 (Sigmoid)	0.3217	2.65	0.3131
ANN2 (Sigmoid)	0.3098	2.67	0.3131
DNN1 (ReLU)	0.2451	2.74	0.2475
DNN2 (ReLU)	0.2238	2.76	0.2464

Table 3.6: Evaluating Performance under the M109 noise of SNR 5dB

Models	MSE	PESQ	Estimated_MSE
ANN1 (ReLU)	0.3977	2.54	0.3977
ANN2 (ReLU)	0.3607	2.57	0.3607
ANN1 (Sigmoid)	0.4108	2.55	0.3792
ANN2 (Sigmoid)	0.3744	2.56	0.3792
DNN1 (ReLU)	0.3049	2.62	0.3059
DNN2 (ReLU)	0.2895	2.65	0.2932

Table 3.7: Evaluating Performance under the Babble noise of SNR 5dB

Models	MSE	PESQ	Estimated_MSE
ANN1 (ReLU)	0.4323	2.52	0.4323
ANN2 (ReLU)	0.3829	2.55	0.3829
ANN1 (Sigmoid)	0.4384	2.51	0.4076
ANN2 (Sigmoid)	0.3950	2.53	0.4076
DNN1 (ReLU)	0.3415	2.59	0.3528
DNN2 (ReLU)	0.3267	2.61	0.3315

to a better expressive power. Experimental evidence also suggests a configuration with a broader width at the top hidden layer is essential to achieving a better expressive power for DNN based vector-to-vector regression. Moreover, the related properties of DNN based vector-to-vector regression function can be maintained in noisy adverse conditions of various SNR levels. Furthermore, the evaluated MSE upper bound can be closely estimated

based on our Propositions 1 and 2.

Besides, since optimizing a DNN with more than two hidden layers is a non-convex problem, the optimization error may affect the reliability of the estimated MSE strategies discussed in this work. Thus, some theoretical work on the issue of optimization methods for DNN should be essentially considered to discuss the generalization capability of DNN based vector-to-vector regression.

CHAPTER 4
ON MEAN ABSOLUTE ERROR FOR DEEP NEURAL NETWORK BASED VECTOR-TO-VECTOR REGRESSION

4.1 Introduction

Mean absolute error (MAE) [80], originated from a measurement of average error [81], is often employed in assessing vector-to-vector (a.k.a. multivariate) regression models [82]. Another form of average error is a root-mean-squared error (RMSE) [83], but MAE was shown to outperform RMSE for assessing an average model accuracy in most situations except the Gaussian noisy scenarios [84, 85, 86]. An exception occurs when the expected error satisfies Gaussian-distributed and enough training samples are available [84]. Besides, mean squared error (MSE) [87] is the squared form of RMSE and it is commonly adopted as a regression loss function [88, 89, 90, 91].

In the literature, there have been some discussions on the relationship between MSE and MAE. Berger [92] presented the pros and cons of squared and absolute errors from an estimation point of view. In [93], a better solution to support vector machines could be obtained based on a loss function of an absolute difference instead of the quadratic error. Li *et al.* [94] discussed the effectiveness of MAE and its variations when training a deep model for energy load forecasting; Imani *et al.* [95] investigated distributional losses, including both MAE and MSE, for regression problems from the perspective of efficient optimization. Pandey and Wang [96] exploited the MAE and MSE loss functions for generative adversarial nets (GANs). However, a comparison between MAE and MSE in terms of generalization capabilities [97, 98, 34] is still missing in theory. Thus, we aim at bridging this gap. In particular, we investigate MAE and MSE in terms of performance error bounds and robustness against various noises in the context of the DNN based vector-to-vector re-

gression, since DNNs offer better representation power and generalization capabilities in large-scale regression problems, such as those addressed in [99, 100, 2, 37].

In this chapter, we first prove that the Lipschitz continuity property, which holds for MAE but not for MSE, is a necessary condition to derive the upper bound on the Rademacher complexity of DNN based vector-to-vector regression operators, as we have demonstrated in [36]. Next, we show that the MAE Lipschitz continuity property can result in a new upper bound on the generalization capability of DNN based vector-to-vector regression in the presence of additive noise. Moreover, another contribution of this work is that we establish a connection between the MAE loss function and the Laplacian distribution, which is in contrast to the MSE loss function associated with the Gaussian distribution. In doing so, we can highlight the key advantages of MAE over MSE by comparing the characteristics of those two distributions.

Our experiments of speech enhancement are used as the regression task to evaluate our theoretical derivations and empirically verify the effectiveness of MAE over MSE. We choose regression-based speech enhancement because it is an unbounded mapping from $\mathbb{R}^D \rightarrow \mathbb{R}^Q$, where enhanced speech features are expected to closely approximate the clean speech features in regression.

Our theory and experimental verification of MAE for DNN based vector-to-vector regression have been shown in the publication [35].

4.2 Useful Definitions

4.2.1 Mean Absolute Error (MAE)

Definition 2 (MAE [80])**.** *MAE measures the average magnitude of absolute differences between N predicted vectors $S = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N\}$ and $S^* = \{\boldsymbol{y}_1, \boldsymbol{y}_2, ..., \boldsymbol{y}_N\}$. The corre-*

sponding loss function is defined as:

$$\mathcal{L}_{MAE}(S, S^*) = \frac{1}{N}\sum_{i=1}^{N} ||\boldsymbol{x}_i - \boldsymbol{y}_i||_1, \tag{4.1}$$

where $||\cdot||$ denotes the L_1-norm.

4.2.2 MSE

Definition 3 (MSE [87])**.** *MSE denotes a quadratic scoring rule that measures the average magnitude of N predicted vectors $S = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N\}$ and N actual observations $S^* = \{\boldsymbol{y}_1, \boldsymbol{y}_2, ..., \boldsymbol{y}_N\}$. The corresponding loss function is shown as:*

$$\mathcal{L}_{MSE}(S, S^*) = \frac{1}{N}\sum_{i=1}^{N} ||\boldsymbol{x}_i - \boldsymbol{y}_i||_2^2, \tag{4.2}$$

where $||\cdot||_2$ denotes L_2-norm.

Besides, two useful mathematical tools "Lipschitz function" and "empirical Rademacher complexity" are separately defined as follows:

Definition 4 (Lipschitz Continuity)**.** *A function f is β-Lipschitz continuous if $\forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^D$ and an integer $P \geq 1$,*

$$||f(\boldsymbol{x}) - f(\boldsymbol{y})||_P \leq \beta ||\boldsymbol{x} - \boldsymbol{y}||_P. \tag{4.3}$$

Definition 5 (Empirical Rademacher Complexity)**.** *The empirical Rademacher complexity of a hypothesis space $\mathbb{H}$ of functions $h : \mathbb{R}^D \to \mathbb{R}$ with respect to N samples $S = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N\}$ is:*

$$\hat{\mathcal{R}}_S(\mathbb{H}) := \mathbb{E}_{\alpha_1,...,\alpha_N}\left[\sup_{h\in\mathbb{H}} \frac{1}{N}\sum_{n=1}^{N} \alpha_n h(\boldsymbol{x}_n)\right]. \tag{4.4}$$

where $\boldsymbol{x}_i \in \mathbb{R}^D$, $\alpha_1, \alpha_2, ..., \alpha_N$ are the Rademacher random variables, which are defined by

the uniform distribution as:

$$\alpha_i = \begin{cases} 1, & \textit{with probability } \frac{1}{2} \\ \textit{-1}, & \textit{with probability } \frac{1}{2}. \end{cases} \tag{4.5}$$

In [101, 102, 103], it was shown that a function class with larger empirical Rademacher complexity is more likely to be over-fitted to the training data.

Lemma 4 (Talagrand's Lemma [90])**.** *Let* $\Phi_1, \Phi_2, ..., \Phi_N$ *be* L*-Lipschitz functions and* $\boldsymbol{\alpha} = \{\alpha_1, \alpha_2, ..., \alpha_N\}$ *be Rademacher random variables. Then, for any hypothesis space* $\mathbb{H}$ *of functions* $h : \mathbb{R}^D \rightarrow \mathbb{R}$ *with respect to* N *samples* $S = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N\}$*, the following inequality holds*

$$\frac{1}{N}\mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{h\in\mathbb{H}}\sum_{i=1}^{N}\sigma_i(\Phi_i \circ h)(\boldsymbol{x}_i)\right] \leq \frac{L}{N}\mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{h\in\mathbb{H}}\sum_{n=1}^{N}\sigma_i h(\boldsymbol{x}_i)\right] = L\hat{\mathcal{R}}_S(\mathbb{H}), \tag{4.6}$$

where $\hat{\mathcal{R}}_S(\mathbb{H})$ *refers to the empirical Rademacher complexity.*

4.3 Characterizing MAE for DNN based Vector-to-Vector Regression

4.3.1 MAE Loss Function for Upper Bounding Empirical Rademacher Complexity

The Lipschitz continuity property is fundamental to derive an upper bound of the estimated regression error. In the following Lemma 5, we show that the MAE loss function can ensure the Lipschitz continuity property. In Lemma 6, we instead show that the property does not hold for MSE.

Lemma 5. *The MAE loss function is* 1*-Lipschitz continuous.*

Proof. For two vectors $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^D$, and a target vector $\mathbf{x} \in \mathbb{R}^D$, the absolute value of

MAE loss difference is

$$\begin{aligned} |\mathcal{L}(\mathbf{x}_1, \mathbf{x}) - \mathcal{L}(\mathbf{x}_2, \mathbf{x})| &= |||\mathbf{x}_1 - \mathbf{x}||_1 - ||\mathbf{x}_2 - \mathbf{x}||_1| \\ &\leq ||\mathbf{x}_1 - \mathbf{x}_2||_1 \\ &= \mathcal{L}_{\text{MAE}}(\mathbf{x}_1, \mathbf{x}_2). \end{aligned} \tag{4.7}$$

□

Lemma 6. *The MSE loss function cannot lead to the Lipschitz continuity property.*

Proof. $\forall \mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^D$, and $||\mathbf{x}_2||_2^2 > ||\mathbf{x}_1||_2^2$, there is

$$||\mathbf{x}_1 - \mathbf{x}_2||_2^2 = ||\mathbf{x}_1||_2^2 + ||\mathbf{x}_2||_2^2 - 2\mathbf{x}_1^\top \mathbf{x}_2. \tag{4.8}$$

Next, we assume $\mathbf{x} = 2\mathbf{x}_2$, and we have that

$$\begin{aligned} ||\mathbf{x}_1 - \mathbf{x}||_2^2 - ||\mathbf{x}_2 - \mathbf{x}||_2^2 &= ||\mathbf{x}_1||_2^2 - 2\mathbf{x}_1^\top \mathbf{x} - ||\mathbf{x}_2||_2^2 + 2\mathbf{x}_2^\top \mathbf{x} \\ &= ||\mathbf{x}_1||_2^2 - 4\mathbf{x}_1^\top \mathbf{x}_2 - ||\mathbf{x}_2||_2^2 + 4||\mathbf{x}_2||_2^2 \\ &= ||\mathbf{x}_1||_2^2 - 4\mathbf{x}_1^\top \mathbf{x}_2 + 3||\mathbf{x}_2||_2^2. \end{aligned} \tag{4.9}$$

By reducing Eq. (4.8) from Eq. (4.9),

$$\begin{aligned} &||\mathbf{x}_1 - \mathbf{x}||_2^2 - ||\mathbf{x}_2 - \mathbf{x}||_2^2 - ||\mathbf{x}_1 - \mathbf{x}_2||_2^2 \\ &= 2||\mathbf{x}_2||_2^2 - 2\mathbf{x}_1^\top \mathbf{x}_2 \\ &> ||\mathbf{x}_2||_2^2 + ||\mathbf{x}_1||_2^2 - 2\mathbf{x}_1^\top \mathbf{x}_2 \\ &= ||\mathbf{x}_1 - \mathbf{x}_2||_2^2 \\ &> 0, \end{aligned} \tag{4.10}$$

we derive that

$$\left|||\mathbf{x}_1 - \mathbf{x}||_2^2 - ||\mathbf{x}_2 - \mathbf{x}||_2^2\right| > ||\mathbf{x}_1 - \mathbf{x}_2||_2^2, \tag{4.11}$$

which contradicts the property of Lipschitz continuity. Thus, the MSE loss function is not Lipschitz continuous.

We now discuss the characteristic of Lipschitz continuity derived from the MAE loss function for upper bounding the estimation error $\mathcal{T}$, which is associated with the generalization capability and defined as:

$$\mathcal{T} = \sup_{f \in \mathbb{F}_K} |\mathcal{L}_S(f) - \mathcal{L}_{\mathcal{D}}(f)| \leq \hat{\mathcal{R}}_S(\mathbb{L}_S), \tag{4.12}$$

where $\mathbb{F}_K = \{f : \mathbb{R}^D \to \mathbb{R}^Q\}$ is a family of DNN based vector-to-vector functions and $\mathbb{L}_S$ denotes the family of expected MAE loss functions. In [36], we have shown that the estimation error $\mathcal{T}$ can be upper bounded by the empirical Rademacher complexity $\hat{\mathcal{R}}_S(\mathbb{L}_S)$.

In [36], we have also shown that the estimation error $\mathcal{T}$ can be further upper-bounded as:

$$\mathcal{T} = \sup_{f \in \mathbb{F}_K} |\mathcal{L}_S(f) - \mathcal{L}_{\mathcal{D}}(f)| \leq \hat{\mathcal{R}}_S(\mathbb{L}_S) \leq \hat{\mathcal{R}}_S(\mathbb{F}_K), \tag{4.13}$$

where $\hat{\mathcal{R}}_S(\mathbb{F}_K)$ is defined as:

$$\hat{\mathcal{R}}_S(\mathbb{F}_K) = \frac{1}{N}\mathbb{E}_{\boldsymbol{\alpha}}\left[\sup_{f \in \mathbb{F}_K} \sum_{i=1}^{N} (\alpha_i \mathbf{1})^\top f(\mathbf{x}_i)\right], \tag{4.14}$$

where $\boldsymbol{\alpha} = \{\alpha_1, \alpha_2, ..., \alpha_N\}$ denotes a set of Rademacher random variables. □

4.3.2 MAE Loss Function for DNN Robustness Against Additive Noises

We now show that the MAE loss function can give an upper bound for regression errors to ensure DNN robustness against additive noises.

Theorem 8. *For an objective function $h = \mathcal{L} \circ f : \mathbb{R}^D \to \mathbb{R}$ with the MAE loss function $\mathcal{L} : \mathbb{R}^Q \to \mathbb{R}$ and a vector-to-vector regression function $f : \mathbb{R}^D \to \mathbb{R}^Q$, the difference of*

the objectives for adding noise $\boldsymbol{\eta}$ to signal $\boldsymbol{x}$ is bounded as:

$$|h(\boldsymbol{x}+\boldsymbol{\eta}) - h(\boldsymbol{x})| \leq L_2||\boldsymbol{\eta}||_2, \tag{4.15}$$

where $L_2 = \sum_{i=1}^{Q} L_{2,i}$ is the Lipschitz constant for DNN based vector-to-vector regression, and each $L_{2,i}$ is shown as:

$$L_{2,i} = \sup\{||\nabla f_i(\boldsymbol{x})||_2 : \boldsymbol{x} \in \mathbb{R}^D\}. \tag{4.16}$$

Proof. To prove Theorem 8, we first introduce Lemma 7, which is achieved by the modification of Theorem 1 in [104].

Lemma 7. *For a vector-to-vector regression function $f : \mathbb{R}^D \to \mathbb{R}^Q$ with the property of Lipschitz continuity, $\forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^D$, there exists an inequality as:*

$$||f(\boldsymbol{x}) - f(\boldsymbol{y})||_1 \leq L_P||\boldsymbol{x} - \boldsymbol{y}||_Q, \tag{4.17}$$

where $L_P = \sup\{||\nabla f(\boldsymbol{x})||_P : \boldsymbol{x} \in \mathbb{R}^D\}$ is a Lipschitz constant, and $\frac{1}{P} + \frac{1}{Q} = 1, P, Q \geq 1$.

We employ the fact that DNNs with the ReLU activation function is Lipschitz continuous [105]. Then, based on both triangle inequality and Lemma 7, we can upper bound the difference of objective functions with and without the additive noise $\boldsymbol{\eta}$ as:

$$\begin{aligned} |h(\mathbf{x}+\boldsymbol{\eta}) - h(\mathbf{x})| &= |||f(\mathbf{x}+\boldsymbol{\eta})||_1 - ||f(\mathbf{x})||_1| \\ &\leq ||f(\mathbf{x}+\boldsymbol{\eta}) - f(\mathbf{x})||_1 \quad \text{(triangle ineq.)} \\ &= L_2||\boldsymbol{\eta}||_2 \qquad\qquad\quad \text{(Lemma 2)}, \end{aligned}$$

which completes the proof. □

Theorem 8 holds for the MAE loss function but is not valid for MSE loss because it is not Lipschitz continuous. In other words, the difference of additive noises imposed

upon the DNN based vector-to-vector function is unbounded on the MSE loss function but the MAE can guarantee an upper bound. The upper bound makes more sense when the additive noise is small because the upper bound suggests that the imposed noise cannot lead to significant performance degradation.

4.3.3 Connection of MAE Loss Function to Laplacian Distribution

We now separately link the MAE and MSE loss functions to Laplacian distribution (LD) and Gaussian distribution (GD) based loss functions as defined in [106]. Both LD and GD-based losses for DNN based multivariate regression were experimentally compared and contrasted in [106], and it was shown that the LD loss can attain better vector-to-vector regression accuracies than those obtained optimizing GD losses.

For N input samples $\{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$ and N target vectors $\{\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_N\}$, assuming $f : \mathbb{R}^D \rightarrow \mathbb{R}^Q$ is a vector-to-vector regression function, we change the MAE loss function as:

$$\begin{aligned}\mathcal{L}_{MAE}(S, S^*) &= \frac{1}{N}\sum_{i=1}^{N} ||f(\mathbf{x}_n) - \mathbf{y}_n||_1 \\ &= \frac{1}{N}\sum_{n=1}^{N}\sum_{m=1}^{D} |f_m(\mathbf{x}_n) - y_{n,m}| \\ &= \frac{1}{N}\sum_{n=1}^{N}\sum_{m=1}^{D} \frac{|\hat{f}_m(\mathbf{x}_n) - \hat{y}_{n,m}|}{\alpha_m},\end{aligned} \tag{4.18}$$

where $\hat{f}_m(\mathbf{x}_n) = \alpha_m f_m(\mathbf{x}_n)$, $\hat{y}_{n,m} = \alpha_m y_{n,m}$, and α_m is the variance of dimension m.

To link the LD based loss function $\mathcal{L}_{LD}(S, S^*)$ in [106], an additional term $N\sum_{m=1}^{D} \ln \alpha_m$ is added to $\mathcal{L}_{MAE}(S, S^*)$, and we obtain

$$\mathcal{L}_{LD}(S, S^*) = \mathcal{L}_{MAE}(S, S^*) + N\sum_{m=1}^{D} \ln \alpha_m. \tag{4.19}$$

Moreover, an MSE based loss function can be modified as:

$$\mathcal{L}_{MSE}(S, S^*) = \frac{1}{N}\sum_{n=1}^{N}\sum_{m=1}^{D}\frac{|\hat{f}_m(\mathbf{x}_n) - \hat{y}_{n,m}|^2}{\alpha_m^2}. \tag{4.20}$$

Then, the GD based loss $\mathcal{L}_{GD}(S, S^*)$ can be derived by adding the term $N\sum_{m=1}^{D}\ln\alpha_m$ to the MSE loss $\mathcal{L}_{MSE}(S, S^*)$,

$$\mathcal{L}_{GD}(S, S^*) = \mathcal{L}_{MSE}(S, S^*) + N\sum_{m=1}^{D}\ln\alpha_m. \tag{4.21}$$

We can observe that $\mathcal{L}_{MAE}(S, S^*)$ and $\mathcal{L}_{MSE}(S, S^*)$ are special cases of $\mathcal{L}_{LD}(S, S^*)$ and $\mathcal{L}_{GD}(S, S^*)$ without concerning the variance terms. When $\forall m \in [D]$, the variance α_m is a constant, $\mathcal{L}_{LD}(S, S^*)$ and $\mathcal{L}_{GD}(S, S^*)$ exactly correspond to $\mathcal{L}_{MAE}(S, S^*)$ and $\mathcal{L}_{MSE}(S, S^*)$, respectively.

Since the work [106] suggests that the LD-based loss function can achieve better regression performance than the GD-based one, we show that the MAE-based loss function can also keep the advantage over the MSE when the variance related terms are the same.

4.4 Experiments

This section presents our speech enhancement experiments to corroborate the aforementioned theorems. The goal of the experiments is to verify that MAE can achieve better regression performance than MSE under various noisy conditions because of the ensured upper bounds on the MAE loss functions for DNN based vector-to-vector regression.

4.4.1 Experimental Setup

Our experiments were conducted on the Edinburgh noisy speech database, which has been introduced in Chapter 2. In this work, DNN based vector-to-vector regression models followed feed-forward architectures, where the inputs were normalized log-power spectral

(LPS) feature vectors of noisy speech [107, 108], and the outputs were LPS features of either clean or enhanced speech. At training time, clean LPS vectors were assigned to the top layer of DNN to function as targets. At test time, the top layer of DNN generated enhanced LPS vectors. The architecture of DNN had the structure 771-800-800-800-800-800-1600-257, which corresponds to Input$-$Hidden$-$Output. The ReLU activation function was employed in the hidden neurons, and a linear activation function was used in the top layer for the vector-to-vector regression. The enhanced waveforms were reconstructed based on the overlap-add method as shown in [2]. The technique of global variance equalization [109] was utilized to improve the subjective perception of speech enhancement. At training time, the BP algorithm was adopted to update the model parameters. The MAE and MSE loss functions were separately used to measure the difference between normalized LPS features and the reference ones. The SGD based optimizer with a learning rate of 1×10^{-3} and a momentum rate of 0.4 was set up for the BP algorithm. Moreover, the technique of NAT was also used to enable non-stationary noise awareness. Context information was accounted at the input by using 3 LPS vectors by concatenating frames within a sliding window [73, 110, 111]. During the training time, one-tenth of training data were randomly split into a validation set, and the training process was stopped if the performance of the model on the validation dataset started to degrade. The evaluation metrics were based on four types of criteria: MAE, MSE, PESQ, and STOI, which have been described in Chapter 2.

4.4.2 Evaluation Results

We train two DNN models based on the MAE criterion (DNN-MAE) and the MSE criterion (DNN-MSE), respectively. Table 4.1 presents the MAE values for speech enhancement experiments with test data. The MAE value evaluated for DNN-MAE in the top row is lower than that that in the bottom row evaluated for DNN-MSE under the same noisy condition in the first column. In more detail, DNN-MAE achieves a lower MAE score than DNN-MSE (0.6876 vs. 0.6922). Similarly, even though MSE is utilized for the DNN training,

DNN-MSE can obtain a lower MSE loss score than DNN-MAE (0.8476 vs. 0.8447), but the margin difference is smaller compared with the first column evaluated by MAE.

Table 4.1: The MAE and MSE Values of unseen test data on Edinburgh speech corpus.

Models	MAE	MSE
DNN-MAE	0.6876	0.8476
DNN-MSE	0.6922	0.8447

Table 4.2: The PESQ and STOI scores of unseen test data on Edinburgh speech corpus.

Models	PESQ	STOI
DNN-MAE	2.74	0.8529
DNN-MSE	2.67	0.8420

Table 4.2 shows PESQ and STOI scores obtained with the DNN-MAE and DNN-MSE models. It can be seen that the DNN model trained with the MAE criterion consistently outperforms the models trained with the MSE criterion (2.74 vs. 2.67 for PESQ, and 0.8529 vs. 0.8420 for STOI), which further confirms that MAE is a good objective function to optimize when training DNNs for speech enhancement.

Furthermore, the performance advantages of DNN-MAE over DNN-MSE reflects what is expected by the aforementioned theorems, namely: (1) the upper bound in Eq. (4.15) ensures more robust performance against the additive noise; (2) the performance gain is consistent with the connection between MAE loss function and the Laplacian distribution.

CHAPTER 5

ANALYZING UPPER BOUNDS ON MEAN ABSOLUTE ERRORS FOR DEEP NEURAL NETWORK BASED VECTOR-TO-VECTOR REGRESSION

5.1 Introduction

In Chapter 2, we investigate the representation power of DNN based vector-to-vector regression. In this chapter, we explore the generalization capability of DNN based regression problems. In particular, we focus on an analysis of the generalization power and study the upper bounds on an expected loss of mean absolute error (MAE) for DNN based vector-to-vector regression with mismatched training and testing scenarios. Moreover, we associate the required constraints with DNN models to attain the upper bounds.

In the literature, the recent success of deep learning has inspired many studies on the expressive power of DNNs, which extended the classical universal approximation theory on shallow ANNs to DNNs. As discussed in [101], an approximation error is tightly associated with the DNN expressive power. Moreover, the estimation error represents the DNN generalization power, which can be reflected by error bounds on the out-of-sample error or the testing error. The methods of analyzing DNN generalization power are mainly divided into two classes: one refers to algorithm-independent controls [112, 113, 114] and another one denotes algorithm-dependent [115, 28]. In the class of algorithm-independent controls, the upper bounds for the estimation error are based on the empirical Rademacher complexity [116] for a functional family of certain DNNs. In practice, those approaches concentrate on techniques of how weight regularization affects the generalization error without considering advanced optimizers and the configuration of hyper-parameters. As for the algorithm-dependent approaches [115, 28], several theoretical studies focus on the "over-parameterization" technique [117, 118, 119, 120], and they suggest that a global

optimal point can be ensured if parameters of a neural network significantly exceed the amount of training data during the training process.

Besides, the generalization capability of deep models can also be investigated through the stability of the optimization algorithms. More specifically, an algorithm is stable if a small perturbation to the input does not significantly alter the output, and a precise connection between stability and generalization power can be found in [121, 122].

In this chapter, the aforementioned issues are taken into account by employing the error decomposition technique [123] concerning an empirical risk minimizer (ERM) [124, 125] using three error terms: an approximation error, an estimation error, and an optimization error. Then, we analyze generalized error bounds on MAE for DNN based vector-to-vector regression models. More specifically, the approximation error can be upper bounded by modifying our previous bound on the representation power of DNN based vector-to-vector regression [34]. The upper bound on the estimation error relies on the empirical Rademacher complexity [116] and necessary constraints imposed upon DNN parameters. The optimization error can be upper bounded by assuming γ-Polyak-Lojasiewicz (γ-PL) [27] condition under the "over-parameterization" configuration for neural networks [126, 127]. Putting together all pieces, we attain an aggregated upper bound on MAE by summing the three upper bounds. Furthermore, we exploit our derived upper bounds to estimate practical MAE values in experiments of DNN based vector-to-vector regression. The experiments of image de-noising and speech enhancement are employed to corroborate our derived theorems. Our new theories and experimental verification have been published in [36].

5.2 Error Decomposition of the Empirical Loss Function of MAE

Based on the traditional error decomposition approach [90, 91], we generalize the technique to the DNN based vector-to-vector regression, where the smooth ReLU activation function, the regression loss functions, and their associated hypothesis space are separately defined

in Definition 6.

Definition 6. *Given a set of training samples S, $f_S^* \in \mathbb{F}_K$ is defined as the ERM, and $\mathbb{L}_S = \{\mathcal{L}_S(f, f_S^*) : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R} | f \in \mathbb{F}_k\}$ is taken as the class of empirical loss of MAE over $\mathbb{F}_k$. For simplicity, $\mathcal{L}_S(f, f_S^*)$ is denoted as $\mathcal{L}_S(f)$.*

The following proposition bridges the connection of Rademacher complexity between the class $\mathbb{L}_S$ based on the MAE objective function and the class $\mathbb{F}_K$ for DNN based vector-to-vector functions.

Proposition 3. *For any sample set $S = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N\}$ drawn i.i.d. from a given distribution $\mathcal{D}$, the empirical Rademacher complexity $\hat{\mathcal{R}}_S(\mathbb{L}_S)$ can be upper bounded as:*

$$\hat{\mathcal{R}}_S(\mathbb{L}_S) \leq \hat{\mathcal{R}}_{\mathcal{S}}(\mathbb{F}_K), \tag{5.1}$$

where $\hat{\mathcal{R}}_S(\mathbb{F}_K)$ refers to the empirical Rademacher complexity over the family $\mathbb{F}_K$, and it is defined as:

$$\hat{\mathcal{R}}_S(\mathbb{F}_K) = \mathbb{E}_{\boldsymbol{\alpha}} \left[\frac{1}{N} \sup_{f \in \mathbb{F}_K} \sum_{n=1}^{N} (\alpha_n \boldsymbol{I})^\top f(\boldsymbol{x}_n) \right], \tag{5.2}$$

in which $\boldsymbol{\alpha} = \{\alpha_1, \alpha_2, ..., \alpha N\}$ refer to the empirical Radamacher variables.

Proof. In Chapter 4, we have known that the MAE loss function is 1-Lipschitz continuous. By applying Talagrand's Lemma [90], we obtain that

$$\begin{aligned}
\hat{\mathcal{R}}_S(\mathbb{L}_S) &= \frac{1}{N} \mathbb{E}_{\boldsymbol{\alpha}} \left[\sup_{f \in \mathbb{F}_K} \sum_{n=1}^{N} \alpha_n \mathcal{L}_S(f(\mathbf{x}_n)) \right] \\
&= \frac{1}{N} \mathbb{E}_{\boldsymbol{\alpha}} \left[\sup_{f \in \mathbb{F}_K} \sum_{n=1}^{N} \alpha_n \mathcal{L}_S(\sum_{q=1}^{Q} \langle \mathbf{1}_q, f(\mathbf{x}_n) \rangle \mathbf{1}_q) \right] \\
&\leq \frac{1}{N} \mathbb{E}_{\boldsymbol{\alpha}} \left[\sup_{f \in \mathbb{F}_K} \sum_{n=1}^{N} (\alpha_n \mathbf{1})^\top f(\mathbf{x}_n) \right] \\
&= \hat{\mathcal{R}}_S(\mathbb{F}_K).
\end{aligned} \tag{5.3}$$

□

Since $\hat{\mathcal{R}}_S(\mathbb{F}_K)$ is an upper bound of $\hat{\mathcal{R}}_S(\mathbb{L}_S)$, we can utilize the upper bound on $\hat{\mathcal{R}}_S(\mathbb{L}_S)$ to derive the upper bound for $\hat{\mathcal{R}}_S(\mathbb{F}_K)$. Next, we adopt the error decomposition technique to attain an aggregated upper bound, which consists of three error components.

Theorem 9 (Error decomposition)**.** *Let $\mathcal{L}_S \in \mathbb{L}_S$ denote an empirical loss function of MAE for a set of training samples S drawn i.i.d. from a given distribution $\mathcal{D}$. For an expected loss function of MAE $\mathcal{L}_\mathcal{D}$, $\epsilon > 0$, and $0 < \delta < 1$, there exists a returned DNN hypothesis $\bar{f}_S \in \mathbb{F}_K$ such that with a probability of δ, we can attain that*

$$\begin{aligned}
\mathcal{L}_\mathcal{D}(\bar{f}_S) = & \underbrace{\mathcal{L}_\mathcal{D}(f_\mathcal{D}^*)}_{\textit{Approximation Error}} + \underbrace{\mathcal{L}_\mathcal{D}(f_S^*) - \mathcal{L}_\mathcal{D}(f_\mathcal{D}^*)}_{\textit{Estimation Error}} + \underbrace{\big(\mathcal{L}_\mathcal{D}(\bar{f}_S) - \mathcal{L}_\mathcal{D}(f_S^*)\big)}_{\textit{Optimization Error}} \\
& \leq \mathcal{L}_\mathcal{D}(f_\mathcal{D}^*) + 2 \sup_{f \in \mathbb{F}_K} |\mathcal{L}_D(f) - \mathcal{L}_S(f)| + \big(\mathcal{L}_\mathcal{D}(\bar{f}_S) - \mathcal{L}_\mathcal{D}(f_S^*)\big) \\
& \leq \mathcal{L}_\mathcal{D}(f_\mathcal{D}^*) + 2\hat{\mathcal{R}}_S(\mathbb{F}_K) + \big(\mathcal{L}_\mathcal{D}(\bar{f}_S) - \mathcal{L}_\mathcal{D}(f_S^*)\big)
\end{aligned} \tag{5.4}$$

Proof. To proof Eq. (5.4), we need to show that

$$\mathcal{L}_\mathcal{D}(f_S^*) - \mathcal{L}_\mathcal{D}(f_\mathcal{D}^*) \leq 2 \sup_{f \in \mathbb{F}_K} |\mathcal{L}_\mathcal{D}(f) - \mathcal{L}_S(f)| \leq 2\hat{\mathcal{R}}_S(\mathbb{F}_K). \tag{5.5}$$

The first inequality comes from the fact that

$$\begin{aligned}
\mathcal{L}_\mathcal{D}(f_S^*) - \mathcal{L}_\mathcal{D}(f_\mathcal{D}^*) &= \mathcal{L}_\mathcal{D}(f_S^*) - \mathcal{L}_S(f_S^*) + \mathcal{L}_S(f_S^*) - \mathcal{L}_\mathcal{D}(f_\mathcal{D}^*) \\
&\leq \mathcal{L}_\mathcal{D}(f_S^*) - \mathcal{L}_S(f_S^*) + \mathcal{L}_S(f_\mathcal{D}^*) - \mathcal{L}_\mathcal{D}(f_\mathcal{D}^*) \\
&\leq 2 \sup_{f \in \mathbb{F}_K} |\mathcal{L}_S(f) - \mathcal{L}_\mathcal{D}(f)|
\end{aligned} \tag{5.6}$$

Then, we continue to upper bound the term $2 \sup_{f \in \mathbb{F}_K} |\mathcal{L}_S(f) - \mathcal{L}_\mathcal{D}(f)|$. We first define μ

as the expected value of $\sup_{f \in \mathbb{F}_K} |\mathcal{L}_S(f) - \mathcal{L}_{\mathcal{D}}(f)|$, and then introduce the fact that

$$\mu = \mathbb{E}\left[\sup_{f \in \mathbb{F}_K} |\mathcal{L}_S(f) - \mathcal{L}_{\mathcal{D}}(f)|\right] \leq 2\hat{\mathcal{R}}_S(\mathbb{L}_S). \tag{5.7}$$

For a small δ $(0 < \delta < 1)$, we apply the Hoeffding's bound [128] as:

$$\begin{aligned} \mathrm{P}\left(2 \sup_{f \in \mathbb{F}_K} |\mathcal{L}_S(f) - \mathcal{L}_{\mathcal{D}}(f)| \leq \nu\right) &\geq 1 - 2\exp\left(-2N(\nu - \mu)^2\right) \\ &\geq 1 - 2\exp\left(-2N(\nu - 2\hat{\mathcal{R}}_S(\mathbb{L}_S))^2\right) \\ &= \delta, \end{aligned}$$

which can derive ν as:

$$\nu = 2\hat{\mathcal{R}}_S(\mathbb{L}_S) + \sqrt{\frac{1}{2N}\ln\left(\frac{2}{1-\delta}\right)},$$

and we thus obtain that

$$2 \sup_{f \in \mathbb{F}_K} |\mathcal{L}_S(f) - \mathcal{L}_{\mathcal{D}}(f)| \leq 2\hat{\mathcal{R}}_S(\mathbb{L}_S) + \sqrt{\frac{1}{2N}\ln\left(\frac{2}{1-\delta}\right)}.$$

Therefore, for a sufficient large N, we attain that

$$2 \sup_{f \in \mathbb{F}_K} |\mathcal{L}_S(f) - \mathcal{L}_{\mathcal{D}}(f)| \leq 2\hat{\mathcal{R}}_S(\mathbb{L}_S).$$

□

In the following parts of this chapter, each of the error components needs to be upper bounded and an aggregated upper bound is attained to estimate practical MAE values of DNN based vector-to-vector regression.

5.3 Theoretical Upper Bounding on MAE based Vector-to-Vector Regression

5.3.1 An Upper Bound for Approximation Error

The upper bound for the approximation error is shown in Theorem 10, which is based on the modification of our previous theorem for the representation power of DNN based vector-to-vector regression [34].

Theorem 10. *For a smooth vector-to-vector regression target function $h_S^* : \mathbb{R}^D \rightarrow \mathbb{R}^Q$, there exists a DNN $f_{\mathcal{D}}^* \in \mathbb{F}_K$ with $K(K \geq 2)$ modified smooth ReLU based hidden layers, where the width of each hidden layer is at least $D + 2$ and the top hidden layer has n_K units. Then, we derive the upper bound for the approximation error as:*

$$\inf_{f \in \mathbb{F}_K} \mathcal{L}_{\mathcal{D}}(f) = ||h_{\mathcal{D}}^* - f_{\mathcal{D}}^*||_1 = \mathcal{O}\left(\frac{Q}{(n_K + K - 1)^{\frac{r}{D}}}\right), \tag{5.8}$$

where r refers to the maximum value of the differential order of $h_{\mathcal{D}}^*$.

Theorem 10 is a direct theoretical outcome derived from Lemma 2 in [129], where the standard ReLU is employed. Moreover, Theorem 10 requires at least $D + 2$ neurons for a D-dimensional input vector to achieve the upper bound.

5.3.2 An Upper Bound for Estimation Error

Since the estimation error in Eq. (5.5) is upper bounded by the empirical Rademacher complexity $\hat{\mathcal{R}}_S(\mathbb{F}_K)$, we derive Theorem 11 to present an upper bound on $\hat{\mathcal{R}}_S(\mathbb{F}_K)$. Our derived upper bound is explicitly controlled by the constraints of weights in the hidden layers, inputs, and the number of training data. In particular, the constraint of L_1-norm is set to the top hidden layer, and L_2-norm is imposed upon the other hidden layers.

Theorem 11. *For a DNN based vector-to-vector mapping function $f(\boldsymbol{x}) = \boldsymbol{W}_K \circ \sigma \circ \boldsymbol{W}_{K-1} \circ \cdots \circ \boldsymbol{W}_2 \circ \sigma \circ \boldsymbol{W}_1(\boldsymbol{x}) : \mathbb{R}^D \rightarrow \mathbb{R}^Q$ with a ReLU function σ and $\forall i \in [K]$, $\boldsymbol{W}_i$ being the weight*

matrix of the i-th hidden layer, we obtain an upper bound for the empirical Rademacher complexity $\hat{\mathcal{R}}_S(\mathbb{F}_K)$ with regularized constraints of the weights in each hidden layer, and the L_2-norm of input vectors $\boldsymbol{x}$ is bounded by s.

$$
\begin{aligned}
&2 \sup_{f \in \mathbb{F}_K} |\mathcal{L}_S(f) - \mathcal{L}_{\mathcal{D}}(f)| \leq 2\hat{\mathcal{R}}_S(\mathbb{F}_K) \leq \frac{2Q\Lambda^{'}\Lambda^{K-1}s}{\sqrt{N}} \\
&s.t., \quad ||\boldsymbol{W}_K(i,:)||_1 \leq \Lambda^{'}, \forall i \in [Q] \\
&\qquad\quad ||\boldsymbol{W}_j(a,:)||_2 \leq \Lambda, \forall j \in [K-1], a \in [J_j] \\
&\qquad\quad ||\boldsymbol{x}||_2 \leq s,
\end{aligned}
\tag{5.9}
$$

where $\boldsymbol{W}_j(m,n)$ is an element associated with the j-th hidden layer of DNN where m is indexed to neurons in the j-th hidden layer and n is pointed to units of the $(j-1)$-th hidden layer, and $\boldsymbol{W}_j(m,:)$ contains all weights from the m-th neuron to all units in the $(j-1)$-th hidden layer.

Proof. We first consider an ANN with one hidden layer of J neuron units with the ReLU function g_u, and also denote $\mathbb{F}_K$ as a family of ANN based vector-to-vector regression functions. $\mathbb{F}_K$ can be decomposed into the sum of Q subspaces $\sum_{q=1}^{Q} \mathbb{F}_{K,q}$ and each subspace $\mathbb{F}_{K,q}$ is defined as:

$$
\mathbb{F}_{K,q} = \left\{ \mathbf{x} \to \sum_{j=1}^{J} w_j \sigma(\mathbf{u}_j^{\top}\mathbf{x}) \cdot \mathbf{1}_q : ||\mathbf{w}||_1 \leq \Lambda^{'}, ||\mathbf{u}_j||_2 \leq \Lambda \right\},
$$

where J is the number of hidden neurons, $\forall j \in [J]$, $\mathbf{w}$ and $\mathbf{u}_j$ separately correspond to $\mathbf{W}_2(m,:)$ and $\mathbf{W}_1(j,:)$ in Equation 5.9. Given N data samples $\{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$ and N empirical Rademacher variables α_i, the empirical Rademacher complexity of $\mathbb{F}_{K,q}$ is bounded

as:

$$
\begin{aligned}
\hat{\mathcal{R}}_S(\mathbb{F}_{K,q}) &= \frac{1}{N}\mathbb{E}_{\boldsymbol{\alpha}}\left[\sup_{||\mathbf{w}||_1\leq\Lambda^{'},||\mathbf{u}_j||_2\leq\Lambda}\sum_{n=1}^{N}\alpha_n\sum_{j=1}^{J}w_j\sigma(\mathbf{u}_j^{\top}\mathbf{x}_n)\right] \\
&= \frac{1}{N}\mathbb{E}_{\boldsymbol{\alpha}}\left[\sup_{||\mathbf{w}||_1\leq\Lambda^{'},||\mathbf{u}_j||_2\leq\Lambda}\sum_{j=1}^{J}w_j\sum_{n=1}^{N}\alpha_n\sigma(\mathbf{u}_j^{\top}\mathbf{x}_n)\right] \\
&\leq \frac{\Lambda^{'}}{N}\mathbb{E}_{\boldsymbol{\alpha}}\left[\sup_{||\mathbf{u}_j||_2\leq\Lambda}\max_{j\in[J]}\left|\sum_{n=1}^{N}\alpha_n\sigma(\mathbf{u}_j^{\top}\mathbf{x}_n)\right|\right] \qquad \text{(Hölder's ineq.)} \\
&= \frac{\Lambda^{'}}{N}\mathbb{E}_{\boldsymbol{\alpha}}\left[\sup_{||\mathbf{u}||_2\leq\Lambda}\left|\sum_{n=1}^{N}\alpha_n\sigma(\mathbf{u}^{\top}\mathbf{x}_n)\right|\right] \\
&\leq \frac{\Lambda^{'}}{N}\mathbb{E}_{\boldsymbol{\alpha}}\left[\sup_{||\mathbf{u}||_2\leq\Lambda}\left|\sum_{n=1}^{N}\alpha_n\mathbf{u}^{\top}\mathbf{x}_n\right|\right] \qquad \text{(c.f. Telegram's Lemma)} \\
&\leq \frac{\Lambda\Lambda^{'}}{N}\mathbb{E}_{\boldsymbol{\alpha}}\left[||\sum_{n=1}^{N}\alpha_n\mathbf{x}_n||_2\right] \qquad \text{(Cauchy-Schwartz ineq.)} \\
&\leq \frac{\Lambda\Lambda^{'}}{N}\sqrt{\mathbb{E}_{\boldsymbol{\alpha}}\left[||\sum_{n=1}^{N}\alpha_n\mathbf{x}_n||_2^2\right]} \qquad \text{(Jensen's inequality).}
\end{aligned}
\tag{5.10}
$$

The last term in the inequality Eq. (5.10) can be further simplified based on the independence of α_n. Thus, we finally derive the upper bound as:

$$
\begin{aligned}
\hat{\mathcal{R}}_S(\mathbb{F}_{K,m}) &\leq \frac{\Lambda\Lambda^{'}}{N}\sqrt{\mathbb{E}_{\boldsymbol{\alpha}}\left[||\sum_{n=1}^{N}\alpha_n\mathbf{x}_n||_2^2\right]} \\
&= \frac{\Lambda\Lambda^{'}}{N}\sqrt{\sum_{n,m=1}^{N}\mathbb{E}_{\boldsymbol{\alpha}}[\alpha_n\alpha_m](\mathbf{x}_n^{\top}\mathbf{x}_m)} \\
&= \frac{\Lambda\Lambda^{'}}{N}\sqrt{\sum_{n=1}^{N}||\mathbf{x}_n||_2^2} \qquad \text{(independence of } \sigma_n\text{s)} \\
&\leq \frac{\Lambda\Lambda^{'}s}{\sqrt{N}}.
\end{aligned}
\tag{5.11}
$$

The upper bound for $\hat{\mathcal{R}}_S(\mathbb{F}_K)$ is derived based on the fact that for Q families of func-

tions $\mathbb{F}_{K,q}, q \in [Q]$, there is $\hat{\mathcal{R}}_S(\mathbb{F}_K) = \hat{\mathcal{R}}_S(\sum_{q=1}^{Q} \mathbb{F}_{K,q}) = \sum_{q=1}^{Q} \hat{\mathcal{R}}_S(\mathbb{F}_{K,q})$, and thus

$$\hat{\mathcal{R}}_S(\mathbb{F}_K) = \sum_{q=1}^{Q} \hat{\mathcal{R}}_S(\mathbb{F}_{K,q}) \leq \frac{Q\Lambda\Lambda' s}{\sqrt{N}}, \tag{5.12}$$

which is an extension of the empirical Rademacher identities [90]. Then, for the family of DNNs $\mathbb{F}_K$ with k hidden layers activated by the smooth ReLU function, we iteratively apply Talagrand's Lemma and end up deriving the upper bound as:

$$\begin{aligned}
\hat{\mathcal{R}}_S(\mathbb{F}_K) &= \mathbb{E}_{\boldsymbol{\alpha}} \left[\sup_{\forall l, w_{j_l} \in \mathbb{U}} \sum_{q=1}^{Q} \sum_{n=1}^{N} \alpha_n \sum_{j_K=1}^{J_K} w_{j_K} \sigma(\cdots \sum_{j_1=1}^{J_1} w_{j_1} \sigma(\mathbf{u}_j^\top \mathbf{x}_n)) \right] \\
&\leq \mathbb{E}_{\boldsymbol{\alpha}} \left[\sup_{\forall l, w_{j_l} \in \mathbb{U}} \sum_{q=1}^{Q} \sum_{n=1}^{N} \alpha_n \sum_{j_K=1}^{J_K} w_{j_K} \cdots \sum_{j_1=1}^{J_1} w_{j_1} \mathbf{u}_j^\top \mathbf{x}_n \right] \\
&\leq \frac{Q\Lambda' \Lambda^{K-1} s}{\sqrt{N}},
\end{aligned}$$

where $w_{j_1}, ..., w_{j_K}$ are selected from the hypothesis space

$$\mathbb{U} = \left\{ w_{j_1}, ..., w_{j_K} : \sum_{j_K=1}^{J_K} |w_{j_K}| \leq \Lambda', \sqrt{\sum_{j_i=1}^{J_i} w_{j_i}^2} \leq \Lambda, \forall i \in [K-1] \right\}. \tag{5.13}$$

□

5.3.3 An Upper Bound for Optimization Error

Next, we derive an upper bound for the optimization error. A recent work [130] has shown that the γ-PL property can be ensured if neural networks are configured with the setup of the "over-parametrization" [127], which is induced from the two facts as follows:

- Neural networks can satisfy γ-PL condition, when the weights of hidden layers are initialized near the global minimum point [127, 98].
- As the neural network involves more parameters, the update of parameters moves

less, and there exists a global minimum point near the random initialization [119, 120].

Thus, the upper bound on the optimization error can be tractably derived in the context of the γ-PL condition for the expected MAE loss $\mathcal{L}_{\mathcal{D}}$. The ReLU activation function admits smooth DNN based vector-to-vector functions, which can result in an upper bound on the optimization error as:

$$\mathcal{L}_{\mathcal{D}}(\bar{f}_S) - \mathcal{L}_{\mathcal{D}}(f_S^*) \leq \frac{\mu M^2 \beta}{2\gamma}, \tag{5.14}$$

where M and β refer to two constants introduced in the following.

To achieve the upper bound in Eq. (5.14), we assume that the SGD algorithm can result in an approximately equal optimization error for both the expected MAE loss $\mathcal{L}_{\mathcal{D}}$ and the empirical MAE loss $\mathcal{L}_S$, which is

$$\mathcal{L}_{\mathcal{D}}(\bar{f}_S) - \mathcal{L}_{\mathcal{D}}(f_S^*) \approx \mathcal{L}_{\mathcal{S}}(\bar{f}_S) - \mathcal{L}_{\mathcal{S}}(f_S^*). \tag{5.15}$$

Therefore, we focus on analyzing $\mathcal{L}_S(f)$ because it can be updated during the training process. We assume that $\mathcal{L}_S(f)$ is β-smooth with $||\nabla \mathcal{L}_S(f)||_2 \leq M$ and it also satisfies the γ-PL condition from an early iteration t_0. Besides, the learning rate of SGD is set to μ.

Moreover, we define $f_{\mathbf{w}_t} \in \mathbb{F}$ as the function with an updated parameter $\mathbf{w}_t$ at the iteration t, and denote $f_{\mathbf{w}_*} \in \mathbb{F}$ as the function with the optimal parameter $\mathbf{w}_*$. The smoothness of $\mathcal{L}_S$ implies that

$$\mathcal{L}_S(f_{\mathbf{w}_{t+1}}) - \mathcal{L}_S(f_{\mathbf{w}_t}) - \langle \nabla \mathcal{L}_S(f_{\mathbf{w}_t}), \mathbf{w}_{t+1} - \mathbf{w}_t \rangle \leq \frac{\beta}{2} ||\mathbf{w}_t - \mathbf{w}_{t+1}||_2^2. \tag{5.16}$$

Then, we apply the SGD algorithm to update model parameters at the iteration t as:

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \mu \nabla \mathcal{L}_S(f_{\mathbf{w}_t}). \tag{5.17}$$

Next, we substitute $-\mu\nabla\mathcal{L}_S(f_{\mathbf{w}_t})$ in Eq. (5.17) for $\mathbf{w}_{t+1}-\mathbf{w}_t$ in Eq. (5.15), and we have that

$$\mathcal{L}_S(f_{\mathbf{w}_{t+1}})-\mathcal{L}_S(f_{\mathbf{w}_t})+\mu||\nabla\mathcal{L}_S(f_{\mathbf{w}_t})||_2^2\leq\frac{\beta\mu^2}{2}||\nabla\mathcal{L}_S(f_{\mathbf{w}_t})||_2^2. \tag{5.18}$$

By employing the condition $||\nabla\mathcal{L}_S(f_{\mathbf{w}_t})||_2^2\leq M^2$, we further derive that

$$\mathcal{L}_S(f_{\mathbf{w}_{t+1}})-\mathcal{L}_S(f_{\mathbf{w}_t})+\mu||\nabla\mathcal{L}_S(f_{\mathbf{w}_t})||_2^2\leq\frac{\mu^2M^2\beta}{2}. \tag{5.19}$$

Furthermore, we employ the γ-PL condition to Eq. (5.18) and obtain the inequalities as:

$$\begin{aligned}
&\mathcal{L}_S(f_{\mathbf{w}_{t+1}})-\mathcal{L}_S(f_{\mathbf{w}_*})\\
&\leq\left(\mathcal{L}_S(f_{\mathbf{w}_t})-\mathcal{L}_S(f_{\mathbf{w}_*})-\gamma\mu(\mathcal{L}_S(f_{\mathbf{w}_t})-\mathcal{L}_S(f_{\mathbf{w}_*}))\right)+\frac{\mu^2M^2\beta}{2}\\
&\leq\left(1-\mu\gamma\right)\left(\mathcal{L}_S(f_{\mathbf{w}_t})-\mathcal{L}_S(f_{\mathbf{w}_*})\right)+\frac{\mu^2M^2\beta}{2}\\
&\leq\left(1-\mu\gamma\right)^2\left(\mathcal{L}_S(f_{\mathbf{w}_{t-1}})-\mathcal{L}_S(f_{\mathbf{w}_*})\right)+\sum_{i=0}^{1}(1-\gamma\mu)^i\frac{\mu^2M^2\beta}{2}\\
&\leq\cdots\\
&\leq\left(1-\mu\gamma\right)^{t-t_0+1}\left(\mathcal{L}_S(f_{\mathbf{w}_{t_0}})-\mathcal{L}_S(f_{\mathbf{w}_*})\right)+\sum_{i=0}^{t-t_0}(1-\gamma\mu)^i\frac{\mu^2M^2\beta}{2}\\
&\leq\left(1-\mu\gamma\right)^{t-t_0+1}\left(\mathcal{L}_S(f_{\mathbf{w}_{t_0}})-\mathcal{L}_S(f_{\mathbf{w}_*})\right)+\frac{\mu M^2\beta}{2\gamma}\\
&\leq\exp(-\mu\gamma(t-t_0+1))\left(\mathcal{L}_S(f_{\mathbf{w}_{t_0}})-\mathcal{L}_S(f_{\mathbf{w}_*})\right)+\frac{\mu M^2\beta}{2\gamma}.
\end{aligned} \tag{5.20}$$

By connecting the optimization error in Eq. (5.14) to our derived Eq. (5.20), we further

have that

$$\begin{aligned}\mathcal{L}_{\mathcal{D}}(\bar{f}_S) - \mathcal{L}_{\mathcal{D}}(f_S^*) &\approx \mathcal{L}_S(\bar{f}_S) - \mathcal{L}_S(f_S^*) \\ &\leq \exp(-\mu\gamma(T+1))\left(\mathcal{L}_S(f_{\mathbf{w}_0}) - \mathcal{L}_S(f_S^*)\right) + \frac{\mu M^2\beta}{2\gamma} \\ &\approx \frac{\mu M^2\beta}{2\gamma},\end{aligned} \tag{5.21}$$

where $T = t - t_0$ and $f_{\mathbf{w}_0} \in \mathbb{F}$ denotes a function with an initial parameter $\mathbf{w}_0$. The inequality in Eq. (5.21) suggests that when the number of iterations T is sufficiently large, we eventually attain the upper bound as Eq. (5.14).

Remark 1: The "over-parametrization" condition becomes difficult to be configure in practice when large datasets have to be dealt with. In such cases, the upper bound on the optimization error cannot be always guaranteed, but we can relax the configuration of "over-parametrization" for DNNs and assume the γ-PL condition to derive the upper bound on the optimization error. In doing so, our proposed upper bound can be applied to more general DNN based vector-to-vector regression operators.

5.3.4 An Aggregated Bound for MAE

Based on the upper bounds for the approximation error, estimation error, and optimization error, we can derive an upper bound for $\mathcal{L}_{\mathcal{D}}(\bar{f}_S)$. Besides, the constraints as shown in Eq. (5.21), which arise from the upper bounds on the approximation, estimation, and optimization errors, are necessary conditions to derive the upper bound with a probability

$\delta \in (0,1)$ as:

$$
\begin{aligned}
\mathcal{L}_{\mathcal{D}}(\bar{f}_S) &\leq \mathcal{L}_{\mathcal{D}}(f^*_{\mathcal{D}}) + 2\hat{\mathcal{R}}_S(\mathbb{F}_K) + \mathcal{L}_{\mathcal{D}}(\bar{f}_S) - \mathcal{L}_{\mathcal{D}}(f^*_S) \\
&\leq \mathcal{O}\left(\frac{Q}{(J_k+K-1)^{\frac{r}{D}}}\right) + \frac{2Q\Lambda^{'}\Lambda^{K-1}s}{\sqrt{N}} + \frac{\mu M^2 \beta}{2\gamma} \\
\text{s.t.,}\ & \text{Hidden Layers: } J_j \geq D+2, \forall j \in [K] \\
& \text{Regularization: } ||\mathbf{W}_k(i,:)||_1 \leq \Lambda^{'}, \forall i \in [Q] \\
& \qquad\qquad ||\mathbf{W}_j(m,:)||_2 \leq \Lambda, \forall j \in [K-1], m \in [J_j] \\
& \text{Bounded Inputs: } ||\mathbf{x}||_2 \leq s \\
& \text{Optimization constraints: Over-parameterization + PL condition}
\end{aligned}
\tag{5.22}
$$

Eq. (5.22) suggests that several hyper-parameters are required to derive the upper bound, which makes it difficult to be utilized in practice because of the prior setup of μ, M, β and γ are strong assumptions in use. The term $\frac{Q\Lambda^{'}\Lambda^{K-1}s}{\sqrt{N}}$ in Eq. (5.22) may become arbitrarily large when a large K and $\Lambda > 1$ are concerned. Thus, we set Λ as 1 to ensure normalized weights of the first $K-1$ layers, and the amount of training data N could be large enough to ensure a small estimation error.

The configuration of "over-parametrization" requires that the number of model parameters exceeds the amount of training data such that the γ-PL condition can be guaranteed and consequently the upper bound on the optimization error can be attained. However, when the setup of "over-parametrization" cannot be strictly satisfied, the γ-PL condition does not always hold. Then, we can still assume the γ-PL condition to obtain the upper bound as shown in Eq. (5.14), which allows the derived upper bound applicable for more general DNN based vector-to-vector regression functions.

Remark 2: Our work employs MAE as the loss function instead of MSE for the following reasons: (i) MSE does not satisfy the Lipschitz continuity such that the inequality Eq. (5.4) cannot be guaranteed [35]; (ii) The MAE loss function for vector-to-vector regression tasks

can achieve better performance than MSE in experiments [85].

5.4 Estimation of MAE Upper Bounds

MAE can be employed as the loss function for training an ANN or DNN based vector-to-vector regression function. In this section, we discuss how to make use of our theorems to estimate MAE upper bounds for the vector-to-vector regression models in our experiments.

Proposition 4 provides an upper bound on MAE based on our theorem in Eq. (5.22), where c and b are two non-negative hyper-parameters to be estimated from the experimental MAE losses of the ANN-based vector-to-vector regression. An ANN with the ReLU activation function is a convex and smooth function, which implies that the local optimum point returned by the SGD algorithm corresponds to a global one. Then, the estimated hyper-parameters c and b can be used to estimate the MAE values of DNN based vector-to-vector regression.

Proposition 4. *For a smooth target function $h^*_{\mathcal{D}} : \mathbb{R}^D \rightarrow \mathbb{R}^Q$, we use N training data samples to obtain a DNN based vector-to-vector regression function $\bar{f}_S \in \mathbb{F}$ with K ReLU based hidden layers ($K \geq 2$), where the width of each hidden layer is at least $D+2$. Then, we can derive an upper bound for MAE as:*

$$MAE(\bar{f}_S, h^*_{\mathcal{D}}) \leq \frac{cQ}{(n_K + K - 1)^{\frac{r}{D}}} + \frac{2Q\Lambda'\Lambda^{K-1}s}{\sqrt{N}} + b, \tag{5.23}$$

where the hyper-parameters c and b are separately set as:

$$c = \frac{(MAE_1 - MAE_2)J_1^{r/D}J_2^{r/D}}{Q(J_2^{r/D} - J_1^{r/D})}, \tag{5.24}$$

and

$$b = \max\left(MAE_1 - \frac{(MAE_1 - MAE_2)J_2^{r/D}}{J_2^{r/D} - J_1^{r/D}} - \frac{2Q\Lambda' s}{\sqrt{N}}, 0\right). \tag{5.25}$$

Note that MAE_1 and MAE_2 are two practical MAE loss values associated with two ANNs

with hidden units J_1 and J_2, respectively.

Proof. For two ANNs with hidden layers with units J_1 and J_2, we set K to 2 and then estimate their corresponding MAE losses as:

$$\frac{cQ}{J_1^{r/D}} + \frac{2Q\Lambda' s}{\sqrt{N}} + b = MAE_1, \tag{5.26}$$

$$\frac{cQ}{J_2^{r/D}} + \frac{2Q\Lambda' s}{\sqrt{N}} + b = MAE_2, \tag{5.27}$$

which can result in hyper-parameters c and b. In particular, we substitute $\frac{\mu M^2 \beta}{2\gamma}$ for b in Eq. (5.23) and then subtract two sides of Eq. (5.26) by Eq. (5.27), which can result in Eq. (5.24). By replacing c in Eq. (5.26) with Eq. (5.24), we finally obtain Eq. (5.25). □

Compared with our previous approaches to estimating practical MAE values in [34] where the DNN representation power is mainly considered, Eq. (5.23) results from the upper bound on the DNN generalization capability such that it can be used to estimate MAE values in more general experimental settings.

5.5 Experiments

5.5.1 Experimental Goals

Our experiments separately employ the DNN based vector-to-vector regression for both image de-noising and speech enhancement with particular attention to linking empirical results with our proposed theorems. Being different from our analysis on the representation power of the DNN based regression task in [34], we here focus on the generalization capability of the DNN based vector-to-vector regression based on our derived upper bounds. More specifically, we employ the tasks of image de-noising and speech enhancement, where inconsistent noisy conditions are mixed to the clean training and testing data,

to validate our theorems by comparing the estimated MAE upper bound (MAE_B) with the practical one.

It should also be remarked that the image de-noising experiment corresponds to an "over-parametrization" setting, in which the number of DNN parameters is much larger than the amount of training data. We cannot set up the "over-parametrization" for speech enhancement tasks due to a significantly large amount of training data. However, we assume the γ-PL condition and evaluate our derived upper bounds on the speech enhancement tasks.

Therefore, our experiments of image de-noising and speech enhancement aim at verifying the following points:

- The estimated MAE upper bound (MAE_B) matches with experimental MAE values.
- A deeper DNN structure corresponds to a lower approximation error (AE).
- A significantly small optimization error can be achieved if the "over-parametrization" configuration is satisfied. Otherwise, the optimization error could be large enough to dominate MAE losses, even if the γ-PL condition is assumed.

5.5.2 Experiments of Image De-noising

Data Preparation

This section presents the image de-noising experiments on the MNIST dataset [131]. The MNIST dataset consists of 60000 images for training and 10000 ones for testing. We added additive Gaussian random noise (AGRN), with mean 0 and variance 1, to both training and testing data. The synthesized noisy data were then normalized such that for each image the condition $||\mathbf{x}_{noisy}||_2 \leq 1$ is satisfied.

Experimental Setup

The experiments of DNN based vector-to-vector regression were carried out using a feed-forward neural network architecture, where the inputs were 784-dimensional feature vectors of the noisy images and the outputs were 784-dimensional features of either clean or enhanced images. The reference of clean image features associated with the noisy inputs was assigned to the top layer of DNN in the training process, but the top layer corresponded to the features of the enhanced images during the testing stage. Table 5.1 lists the structures of the neural networks used in our experiments. In more detail, the vector-to-vector regression model was first built based on an ANN. The width of the hidden layer of ANN1 was set to 1024, which satisfies the constraint of the number of neurons in hidden layers based on both the inequality Eq. (5.22) ($D = 784$, $D + 2 = 786 < 1024$) and the "over-parametrization" ($784 \times 1024 = 802816 > 60000$) condition; On the other hand, ANN2 had a width of 2048 neurons, which was twice larger than that of ANN1. Next, we studied the DNN based vector-to-vector regression by increasing the number of hidden layers of DNN1. Specifically, DNN1 was equipped with 4 hidden layers with widths 1024-1024-1024-2048. Additional two hidden layers of width 1024 were further appended to DNN2, which brings an architecture with 6 hidden layers 1024-1024-1024-1024-1024-2048.

Table 5.1: Model structures for various vector-to-vector regression

Models	Structures (Input – Hidden_layers – Output)
ANN1	784-1024-784
ANN2	784-2048-784
DNN1	784-1024-1024-1024-2048-784
DNN2	784-1024-1024-1024-1024-1024-2048-784

Moreover, the SGD optimizer with a learning rate of 0.02 and a momentum rate of 0.2 was used to update model parameters based on the standard BP algorithm. The weights of the $K - 1$ hidden layers were normalized by dividing the L_2 norm, which corresponds

to the term Λ^{K-1} configured to 1 in Eq. (5.23). The weights of the top hidden layer were normalized by dividing the L_1 norm such that Λ' is set to 1. Besides, MAE was employed as the evaluation metric in our experimental validation because the MAE metric is directly connected to the objective loss function of MAE.

Experimental Results

We present our experimental results on the noisy MNIST dataset, where the AGRN was added to the clean images. Table 5.2 shows the setup of hyper-parameters J_1, J_2, N, and r in Eq. (5.23) to estimate MAE_B. Table 5.3 shows that the estimated MAE values are in line with the practical MAE values. Specifically, DNN2 attains a lower MAE (0.1278 vs. 0.1263) than DNN1. Moreover, our estimated MAE_B score for DNN2 is also lower than that for DNN1, namely 0.1438 vs. 0.1434, which arises from the decreasing AE score for DNN2 with a deeper architecture. Since we keep Λ and Λ' equal to 1, estimation error (EE) and optimization error (OE) for both DNN1 and DNN2 share the same values. Furthermore, although the OE values are comparatively larger than AE and EE, they also stay at a small level because of the "over-parametrization" technique adopted in our experiments.

Table 5.2: Hyper-parameters for the estimation of MAE upper bounds.

J_1	J_2	N	r	ANN1_MAE	ANN2_MAE
1024	2048	6×10^4	1176	0.1318	0.1292

Table 5.3: The evaluation results under the AGRN noise.

Models	MAE	AE	EE	OE	MAE_B
DNN1	0.1278	0.0172	0.0261	0.1005	0.1438
DNN2	0.1263	0.0168	0.0261	0.1005	0.1434

5.5.3 Experiments of Speech Enhancement

Experimental Setup

Our experiments of speech enhancement were conducted on the Edinburgh noisy speech database, which has been introduced in Chapter 2. The DNN based vector-to-vector regression for speech enhancement also followed the feed-forward ANN architecture, where the input was a normalized LPS feature vector of noisy speech, and the output was LPS feature vectors of either clean or enhanced speech. The references of clean speech feature vectors associated with the noisy inputs were assigned to the top layer of DNN in the training process, but the top layer of DNN corresponds to the feature vectors of the enhanced speech during the testing phase. The ReLU function was employed in the hidden nodes of the neural architectures assessed in this work, whereas a linear function was used at the output layer. To improve the subjective perception in the speech enhancement tasks, the global variance equalization [132] was applied to alleviate the problem of over-smoothing by correcting a global variance between estimated features and clean reference targets [133]. In the training stage, the BP algorithm was adopted to update the model parameters, and the MAE loss was used to measure the difference between a normalized LPS vector, and the reference one. NAT was also employed to enable non-stationary noise awareness, and feature vectors of 3-frame size were obtained by concatenating frames within a sliding window [110]. Moreover, the SGD optimizer with a learning rate of 1×10^{-3} and a momentum rate of 0.4 was used for the update of parameters. The weights of the first $k-1$ hidden layers are normalized by dividing the L_2 norm of each row of weights, which correspond to the term Λ^{k-1} equal to 1 in Eq. (5.22). Moreover, we set s in Eq. (5.22) as the maximum value of L_2 norm of the input, and assume $\Lambda^{'}$ in Eq. (5.22) as the maximum value of $(||\mathbf{W}_k(1,:)||_1, ..., ||\mathbf{W}_k(Q,:)||_1)$, which are different from the setup of image de-noising.

Table 5.4 shows the neural architectures used in our speech enhancement experiments. Two ANN models (ANN1 and ANN2) were utilized to estimate the hyper-parameters in

Eq. (5.23), which were then used to estimate the MAE values of DNN models based on Eq. (5.23).

Table 5.4: Model structures for various vector-to-vector regressions

Models	Structures (Input – Hidden_layers – Output)
ANN1	771-800-257
ANN2	771-1600-257
DNN1	771-800-800-800-1600-257
DNN2	771-800-800-800-800-800-1600-257

The MAE and PESQ metrics are used to assess the quality of enhanced speech in the experiments. All of the evaluation results on the test datasets are presented in Table 5.6.

Experimental Results

We now present our experimental results on the Edinburgh speech database. Table 5.5 shows the parameters used in the experiments to estimate the upper bound based on Eq. (5.22). The experimental results as shown in Table 5.6 are in line with those observed in the consistent noisy conditions. Specifically, DNN2 attains a lower MAE (0.6859 vs. 0.7060) and higher PESQ values (2.85 vs. 2.82) than DNN1. Moreover, the MAE_B score for DNN2 is also lower than that for DNN1, namely 0.7124 vs. 0.7236. Furthermore, DNN2 owns a better representation power in terms of AE scores (0.0081 vs. 0.0161) and a better power generalization capability because of a lower (EE + OE) score. More significantly, the OE term is the key contributor to the MAE_B score, which suggests that the MAE loss is primarily from OE, as expected. Optimization plays an important role when it comes to training large neural architectures [2, 1], which in turn shows that the proposed upper bounds are in line with current research efforts [28, 119, 127, 126] on the optimization strategies.

Table 5.5: Hyper-parameters for the estimation of MAE upper bounds.

l_1	l_2	N	r
800	1600	1.04×10^{10}	771
ANN1_MAE	ANN2_MAE	Λ'(ANN1)	Λ' (ANN2)
0.7409	0.7328	8.9543	10.1542

Table 5.6: The MAE Results on the Edinburgh speech database

Models	MAE	PESQ	AE	EE	OE	MAE_B
DNN1	0.7060	2.82	0.0161	0.0579	0.6496	0.7236
DNN2	0.6859	2.85	0.0081	0.0728	0.6315	0.7124

5.5.4 Discussion

The experimental results of the image de-noising and speech enhancement suggest that our proposed upper bounds on the generalized loss of MAE can tightly estimate the practical MAE values. Unlike our previous work on the analysis of the representation power, which is strictly constrained to consistent noisy environments, our MAE bounds aim at the generalization power of DNN based vector-to-vector regression and can be generalized to more general noisy conditions.

Experimental results are based on our aggregated bound in Eq. (5.22), and the related practical methods in Eq. (5.22). The decreasing AE scores of DNN2 correspond to Eq. (5.8), where a deeper depth K can lead to smaller AE values. In the meanwhile, Eq. (5.24) and Eq. (5.25) suggest that a smaller EE is associated with a larger OE, which also corresponds to our estimated results. Furthermore, deeper DNN structures can result in a larger Λ', which slightly escalates the AE scores and also decreases OE values. With the setup of "over-parametrization" for neural networks in image de-noising experiments, OE can be lowered to a small scale compared to AE and EE. However, OE becomes much larger than AE and EE without the "over-parametrization" configuration in the speech enhancement tasks.

CHAPTER 6

VECTOR-TO-VECTOR REGRESSION BASED ON TENSOR-TRAIN DEEP NEURAL NETWORK

In the preceding chapters, we focus on a vector-to-vector regression based on DNN, where our theoretical analysis suggests that an over-parametrized DNN is preferred. The over-parameterized DNN requires several model parameters greater than the amount of training data. Fortunately, TT-DNN provides a compact tensor representation for a DNN. Thus, in this chapter, we investigate if a tensor-train deep neural network (TT-DNN) with much fewer parameters is capable of maintaining the empirical baseline performance of the DNN counterpart. In particular, we investigate the deployment of the TT-DNN in the fields of speech processing including multi-channel speech enhancement and spoken command recognition. The related work has been published in [37, 38].

6.1 Tensor-Train Deep Neural Network

Tensor-Train Network (TTN) [23], as discussed in Chapter 2, can be generalized to a deeper architecture and it is closely associated with the TT representation for DNN, namely TT-DNN. Figure 6.4 illustrates that a DNN model can be converted into a TT-DNN structure, where the input vector X is decomposed into the TT format as $\mathcal{X}$, and all FC hidden layers are represented as the TT ones. An additional softmax operation is employed for classification. More specifically, $\forall k \in [K]$ and $\forall l \in [L]$, the DNN matrix $\mathbf{W}_l \in \mathbb{R}^{J_l \times I_l}$ can be decomposed into K core tensors $\{W_{l,1}, W_{l,2}, ..., W_{l,K}\}$, where $\mathcal{W}_{l,k} \in \mathbb{R}^{R_k \times J_{l,k} \times I_{l,k} \times R_{k-1}}$, $J_l = J_{l,1} \times J_{l,2} \times \cdots \times J_{l,K}$ and $I_l = I_{l,1} \times I_{l,2} \times \cdots I_{l,K}$.

Figure 6.4 illustrates the TT-DNN model, which is a TT representation for DNN. Although TT-DNN is associated with DNN, TT-DNN can be independently set up and learned from scratch. On the other hand, the TTD admits a TT-DNN model with much fewer model

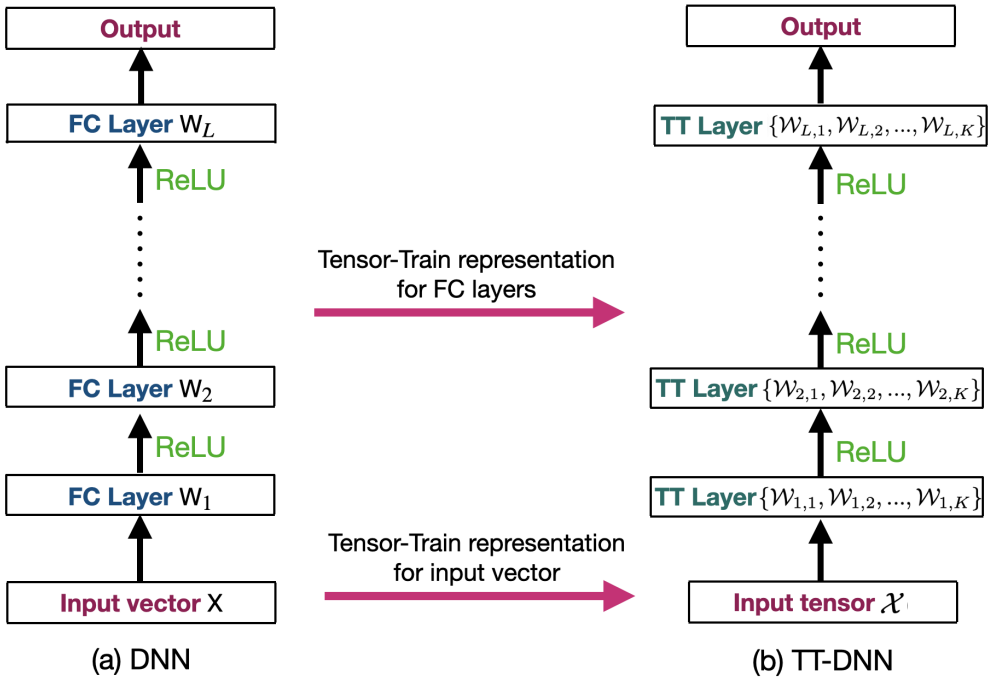

Figure 6.1: *An illustration of converting DNN into TT-DNN, where each FC layer of DNN is converted to K core tensors of TT-DNN.*

parameters than the related DNN. More specifically, a DNN with $\sum_{l=1}^{L} J_l I_l$ parameters could be converted into a TT-DNN with fewer parameters such as $\sum_{l=1}^{L} \sum_{k=1}^{K} J_{l,k} I_{l,k} R_{k-1} R_k$.

For the task of speech enhancement, we also put forth a new hybrid vector-to-vector regression framework, namely CNN+(TT-DNN). The CNN+(TT-DNN) model is composed of a convolutional neural network (CNN) at the bottom for feature extraction and TT-DNN for enhancing speech vectors. In this chapter, we first consider the deployment of TT-DNN and CNN+(TT-DNN) for single speech enhancement, and then we extend the TT models to multi-channel speech enhancement.

6.1.1 TT-DNN based tensor-to-vector regression for speech enhancement

A framework of TT-DNN based multi-channel speech enhancement is shown in Figure 6.2, where the input refers to a multi-dimensional tensor corresponding to speech features from multiple microphones, the output is connected to enhanced speech vectors, and several hidden TT layers are stacked. The TT-DNN architecture is contrasted to the framework of

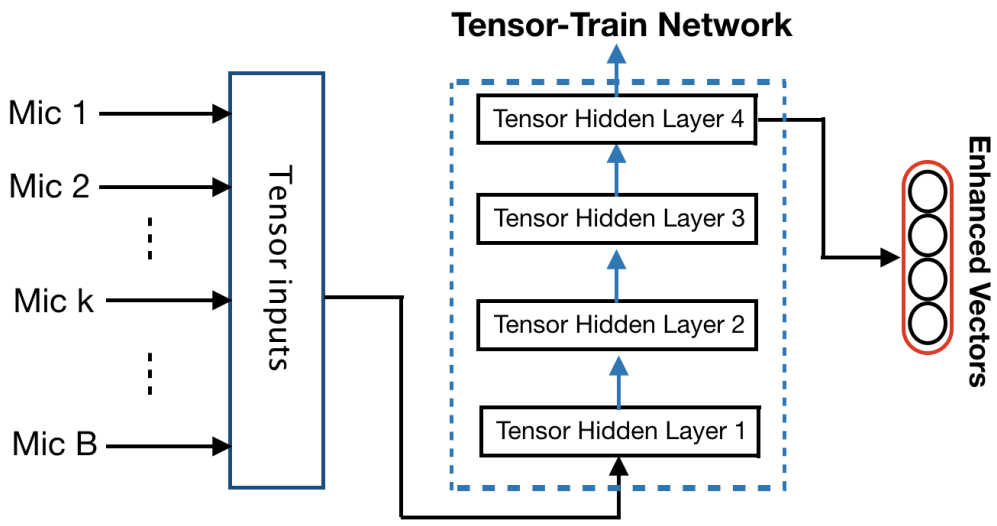

Figure 6.2: *A TT-DNN based multi-channel speech enhancement.*

vector-to-vector regression as shown in Figure 6.3. An array of microphones is exploited from multiple microphones into a single high-dimensional vector so that the vector-to-vector regression approach can be employed for speech enhancement by appending multi-channel feature vectors together into a high-dimensional vector. Such a simple solution clashes with Theorem 6 in Chapter 3, which claims that the width of each DNN hidden layer is greater than the input dimension plus two. In doing so, although the expressive power of DNN based vector-to-vector regression can be guaranteed, a huge amount of computational resources and memory storage are required. Fortunately, the TT-DNN model can significantly reduce the number of model parameters, and our experiments of speech enhancement are investigated whether TT-DNN can maintain the baseline performance of DNN.

6.1.2 Deep hybrid tensor-to-vector regression for speech enhancement

A hybrid tensor-to-vector regression based on CNN+(TT-DNN) is further proposed to improve the performance of TT-DNN. In addition to the DNN baseline as shown in Figure 6.4 (a), Figure 6.4 (b), (c) and (d) demonstrate the tensor-to-vector regression models employed

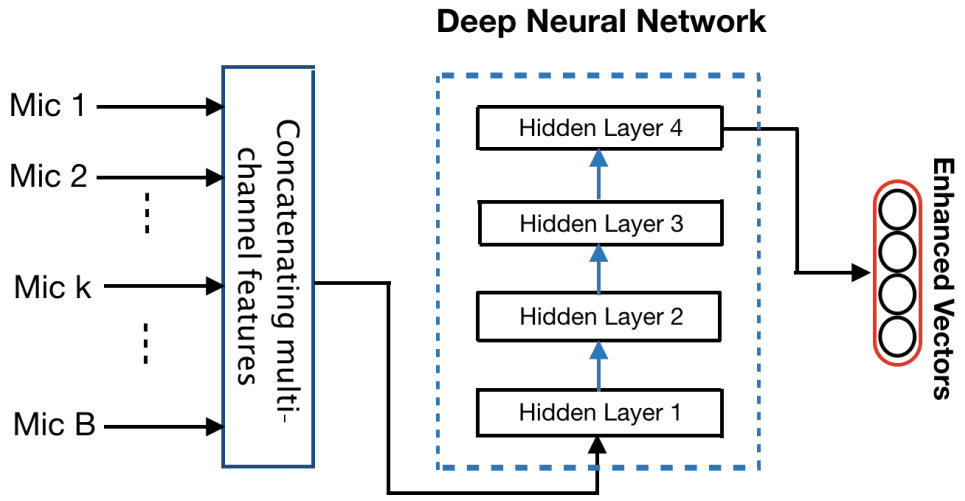

Figure 6.3: *Conventional multi-channel DNN based vector-to-vector regression for speech enhancement.*

in this work. More specifically, Figure 6.4 (b) represents a CNN model where the hidden layers are composed of 2D convolutional layers; Figure 6.4 (c) refers to the model of TT-DNN with all hidden layers stacked with TT layers; Figure 6.4 (d) presents our proposed CNN+(TT-DNN) model, where the 2D convolutional layers are placed at the bottom and the TT layers are appended on the top of convolutional layers. The convolutional layers are used to extract CNN features before going through the FC or TT layers for regressing the enhanced speech.

6.2 Experiments

6.2.1 Single-channel Speech Enhancement

Experimental setup

Our proposed experimental results were assessed on the Edinburgh noisy speech dataset, which has been introduced in Chapter 2. In all experiments, we use 257-dimensional LPS feature vectors as inputs. LPS features were generated by computing 512 points Fourier transform on a speech segment of 32 milliseconds. For each input frame, M neighboring

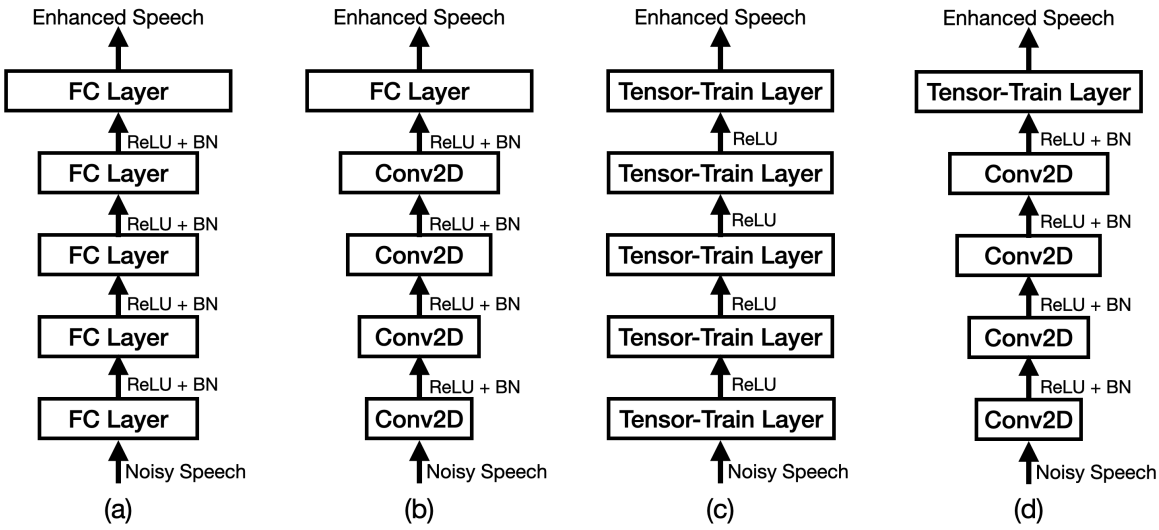

Figure 6.4: *The speech enhancement models utilized in this study, where BN denotes the batch normalization.*

adjacent frames were concatenated together, which results in a total $257 \times (2M + 1) \times B$ dimensional feature, where B is the channel number of the input signal. As for the setup of TT-DNN, we ignore the first dimension of the input LPS features because it corresponds to the direct-current component. After the regression, the first dimension of the input was concatenated back to the 256-dimensional output without any change. The clean speech features were assigned to the top layers of tensor-to-vector regression models as the reference during the training stage.

The DNN based regression model was adopted as a baseline model. The DNN model consisted of 4 hidden layers with hidden dimensions configured to 1024-1024-1024-2048, respectively. Moreover, the CNN models kept similar deep tensor-to-vector structures in all experiments and were composed of four convolutional layers with gradually increasing the number of channels according to the setup of 32-64-128-128. Furthermore, the ReLU activation function and batch normalization (BN) [134] were utilized in each convolutional layer, and two FC layers with 2048 neurons were stacked on the top hidden layer to generate output vectors. Moreover, to improve the subjective perception in the speech enhancement tasks, the global variance equalization was applied to alleviate the problem of

Table 6.1: PESQ comparisons of single-channel deep speech enhancement models on the Edinburgh noisy speech database. The average PESQ score for unprocessed noisy speech is 1.97.

Model	Parameters #	PESQ
DNN	5.5M	2.82
CNN	9.1M	3.04
TT-DNN	0.55M	**2.81**
CNN+(TT-DNN)	**0.73M**	3.02
CNN+(TT-DNN)	2.9M	3.09
CNN+(TT-DNN)	5.1M	**3.13**

over-smoothing by correcting a global variance between estimated features and clean reference targets, and a technique of NAT was also employed to enable non-stationary awareness. Besides, the MSE loss was applied. Adam optimizer with an initial learning rate of 0.002 was utilized during the training stage, and the BP algorithm was used to update the model parameters.

Perceptual evaluation of speech quality (PESQ) [48] was used as the evaluation criterion. The PESQ score, which ranges from -0.5 to 4.5, is calculated by comparing the enhanced speech with the clean one. A higher PESQ score corresponds to a higher quality of speech perception.

Experimental results

Table 6.1 demonstrates our experimental results on the Edinburgh noisy speech data set. The tensor-to-vector regression based on CNN outperforms the DNN baseline results in terms of a higher PESQ score (3.04 vs. 2.82). TT-DNN with much fewer model parameters (0.55M vs. 5.51M) can maintain the same empirical performance of DNN, where the TT transformation was applied in the FC layers. More importantly, compared with the combined CNN and TT layers, the proposed CNN+(TT-DNN) can attain the highest PESQ score. If we allow the size of the CNN+(TT-DNN) model to increase up to 5.05Mb, a better speech enhancement quality can be attained with a PESQ score as high as 3.13.

6.2.2 Multi-Channel Speech Enhancement

Data preparation

Our proposed TT-DNN based models are evaluated on the simulated data from WSJ0, which contains additive noise, interfering speakers, and reverberation. The dataset is created by corrupting the WSJ0 corpus with OSU-100-noise. When simulating the noisy data, each waveform is mixed with one type of background noise, which results in 30 hours of training materials and 5 hours of testing data. The target and additional interfering speech with their corresponding RIRs are convoluted to generate our required waveform. In particular, our training and testing datasets are created from different noisy utterances of various speakers. As for the training dataset, a 5-minute clean speech from each of the targeted speakers is randomly mixed with 73 interfering speakers and 90 types of additive noises. The targeted an-echoic speech is generated by combining clean speech with the direct path response between the targeted speakers and the reference channel. To generate the test dataset, another 5-minute unseen speech of targeted speakers are mixed with 10 unseen interfering speakers and 10 types of unseen noise. The signal-to-interfered-noise-ratio (SINR) level of each utterance is set as follows: when SINR is 5dB, the signal-to-noise-ratio (SNR) is set to $15dB$; when SINR is $15dB$, the SNR level is increased to $20dB$. The proportion of each SINR level is equally set. Besides, some utterances of SINR $30dB$ are included in the training set to cover some very high SINR conditions.

To simulate reverberated speech, a reverberated acoustic environment is built: a microphone array of 8-circular channel microphone is arranged in a room of size $6.5m \times 5.5m \times 3m$ in terms of length-width-height. As for the single-channel scenario, the microphone is placed at the center of the array. To avoid unnecessary combinations of multiple interferences, we deliberately constrain the conditions that the microphone array exactly aims at one targeted speaker, and it received one type of additive noise. Specifically, a horizontal distance of a targeted speaker to the center of the microphone array is strictly fixed to

$3m$. Besides, we set both the targeted speaker and the interfering speaker keeping the same distance to the microphone array, and the angle of them is configured as 40°. Before we build the training and test sets, an important image-source method (ISM) is used to generate RIRs of reverberation time (RT60) (from $0.2s$ to $0.3s$) and the corresponding direct path response for each microphone channel. For both training and testing datasets, the setting of RIRs is fixed to the same conditions, such as the room size, RT60, and all of the distances and directions.

Experimental settings

In our experiments, 257-dimensional normalized LPS features were taken as the inputs to the DNNs. The LPS features were generated by computing 512 points Fourier transformation on a speech segment of 32 milliseconds. For B-channel data, the inputs of all channels were concatenated together for the model training. For each input frame, the adjacent context of size M was combined with the current frame. The input size for TTN was $256 \times (2M+1) \times B$. After the regression, the first dimension of the input was concatenated back to the 256-dimensional output without any change. The clean speech features of the first channel were assigned to the top layers of DNN and TT-DNN, as the reference during the training stage.

Our baseline DNN model is composed of 6 hidden layers, and each hidden layer owns 2048 neurons. The ReLU activation function was used for all hidden layers. A linear function was employed in the top hidden layer. As for the setup of TT-DNN, each hidden dense layer is decomposed and replaced by a TT layer. Both DNN and TT-DNN are jointly trained from scratch based on the standard back-propagation algorithm, and both models adopt the same training configuration. During the training phase, Adam optimizer is adopted, and the initial learning rate is set to 0.0002. The MSE is utilized as the objective function. The context window size at the input layer was set to 5 for all models, in which the current frame was concatenated with the previous 5 frames and the following 5 frames

within the same channel.

Experimental results

Our TT-DNN was first assessed on the single-channel speech enhancement task. In Table 6.2, we observe that a 6-layer DNN model, which is taken as a baseline system, achieves a PESQ score of 2.86 with 27 million parameters. A TT-DNN architecture is generated by applying tensor-train decomposition to the DNN baseline model. Each weight matrix of DNN is decomposed into two four-dimension tensors using tensor decomposition. For example, a weight matrix of the size 2048×2048 can be decomposed to two tensors with the size of $1 \times 32 \times 32 \times 4$ and $4 \times 64 \times 64 \times 1$. The TT-DNN core tensors are randomly initialized and then trained from scratch using the Adam optimizer.

Table 6.2 shows that a substantial parameter reduction when the TT model is employed. A drop in the PESQ value, from 2.86 (DNN) to 2.66 (TT-DNN), is observed because of the parameter reduction. Nonetheless, TT-DNN consistently delivers better speech enhancement results as its number of parameters increases. As shown in Table 6.2, the TT-DNN model with 5 million parameters can achieve nearly the same PESQ scores as the DNN baseline model (2.84 vs 2.86). However, the TT-DNN model uses only 18% of the number of parameters in the DNN, which suggests that TT-DNN can significantly reduce the number of parameters while keeping the baseline performance. Furthermore, if we further increase the parameter number of the TT-DNN model, we can even obtain a better TT-DNN model achieving a 0.1 absolute PESQ improvement using only 74% of the DNN parameters, namely 20 million.

6.2.3 Exploring Hybrid Models of Tensor-Train Networks for Spoken Command Recognition

In this part, we focus on the employment of the hybrid tensor-train model for spoken command recognition. Figure 6.5 illustrates the proposed SCR system. The CNN framework is used to convert speech signals into spectral features in Figure 6.5 (a). The entire CNN

Table 6.2: PESQ results for multi-channel speech enhancement.

Model	Channel #	Parameter #	PESQ
DNN	1	27M	2.86
DNN	2	33M	3.00
DNN	8	68M	3.06
TT-DNN	1	0.6M	2.66
TT-DNN	1	5M	2.84
TT-DNN	1	20M	2.96
TT-DNN	2	5M	2.96
TT-DNN	8	5M	3.06
TT-DNN	8	20M	3.12

framework consists of 4 components, each of which is constructed by stacking a 1D convolutional layer with BN and the ReLU activation, which is followed by a max-pooling layer with a kernel size of 4. Particularly, the first component owns a kernel size of 80 and 16 strides, while the kernel sizes of 3 and the stride of 1 are assigned to the other CNN components. Moreover, the number of channels for the CNN framework follows the pipeline of 1-32-64-64.

The spectral features associated with the outputs of the CNN framework are fed into the FC layers or TT layers, which are shown in Figure 6.5 (b), (c), and (d), respectively. We set our baseline system as the CNN+DNN architecture in which several FC layers are stacked on top of the CNN layers. On the other hand, the FC layers are changed to the TT layers in our proposed CNN+(TT-DNN) model. Besides, two training methods for the CNN+(TT-DNN) model are considered: one refers to setting up the CNN+(TT-DNN) model with a random initialization for model parameters, and another one is derived from the TT decomposition of a well-trained CNN+DNN model. Moreover, the output of the SCR system is connected to the classification labels.

Data profile

Our spoken command recognition experiments were conducted on the Google Speech Command dataset [135], which includes 35 spoken commands, e.g. [‘left’, ‘go’, ‘down’,

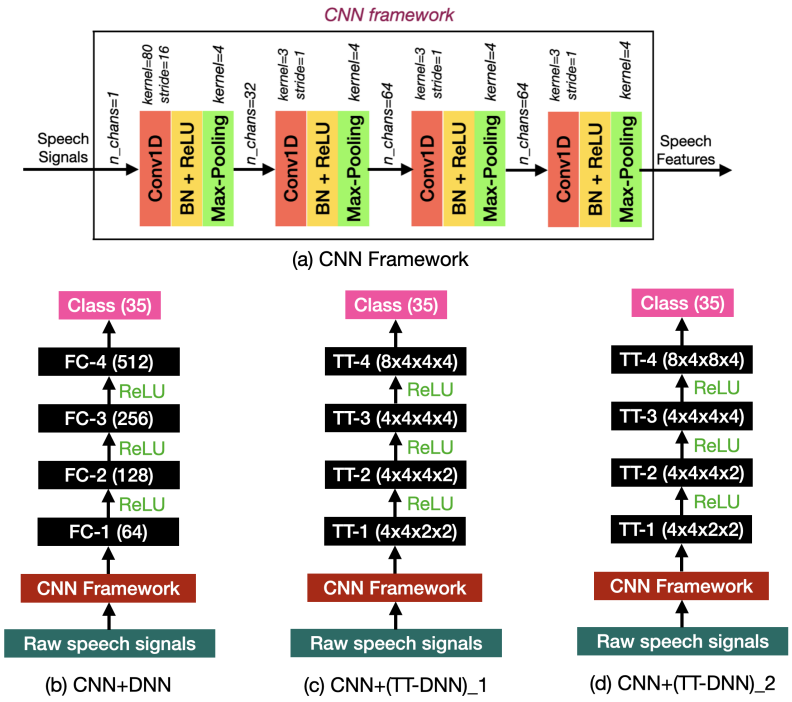

Figure 6.5: *The CNN+DNN and CNN+(TT-DNN) models for spoken command recognition, where CNN+(TT-DNN)_1 and CNN+(TT-DNN)_2 differ in the tensor shape of the top hidden layer.*

'up', 'on', 'right', ...]. There are a total of $11,165$ development and $6,500$ test utterances. The development data are randomly split into two parts: 90% is used for model training and 10% is used for validation. All the audio files are about 1 second long, down-sampled from $16KHz$ to $8KHz$. The batch size was set to 256 in the training process, and the speech signals in a batch were configured as the same length by zero padding.

Experimental Setup

The model architectures for the tested SCR systems are shown in Figure 6.5, where we take three acoustic models into account. Figure 6.5 (b) shows our baseline SCR system in which 4 FC layers (64-128-256-512) are stacked and 35 classes are appended as the label layer. Figure 6.5 (c) illustrates our CNN+(TT-DNN) model where 4 TT layers follow the tensor shape of $4 \times 4 \times 2 \times 2 - 4 \times 4 \times 4 \times 2 - 4 \times 4 \times 4 \times 4 - 8 \times 4 \times 4 \times 4$, whereas in Figure 6.5 (d) the shape of the top TT layer is modified as $8 \times 4 \times 4 \times 4$, because a larger

CNN+(TT-DNN) model is considered to compare SCR performances.

The loss function was based on the criterion of cross-entropy (CE), and the Adam optimizer with a learning rate of 0.01 was used in the training stage. The loss value of CE is a direct assessment for the model performance of SCR, and Accuracy (Acc.) is an indirect measurement to evaluate speech recognition performance. There are a total of 100 epochs used in the training process. We report the best average accuracy for each SCR architecture with 10 runs.

Experimental Results

The CNN+(TT-DNN) models are randomly initialized, and CNN+(TT-DNN)_1 and CNN+(TT-DNN)_2 correspond to the models in Figure 6.5 (c) and (d), respectively. Our models are compared with neural network models available in the literature, namely: DenseNet-121 benchmark used in [136] for SCR, Attention-RNN [137, 138], which refers to the neural attention model, and QCNN [139], which refers to the use of quantum convolutional features for the task. We extend the 10 classes training setup used in [139] to 35 classes to report its final results. All deployed models are trained with the same SCR dataset from scratch without any data augmentation [140] or pre-training techniques [141] to make a fair architecture-wise study.

Table 6.3: The experimental results on the test dataset. Params. represents the number of model parameters; CE means the cross-entropy; and Acc. refers to the classification accuracy.

Models	Params (Mb)	CE	Acc. (%)
DenseNet-121 [136]	7.978	0.473	82.11
Attention-RNN [137]	0.170	0.291	93.90
QCNN [139]	0.186	0.280	94.23
CNN+DNN	0.216	0.251	94.42
CNN+(TT-DNN)_1	**0.056**	0.137	96.31
CNN+(TT-DNN)_2	0.083	**0.124**	**97.20**

Experimental Summary

Our experiments of speech recognition assess different model settings on the task of spoken command recognition. Our experimental results show that the use of TTD can maintain and even obtain better performance than the given baseline models (DenseNet, Attention-RNN, QCNN). In particular, the performance of the CNN+(TT-DNN) model can be boosted by increasing the number of model parameters.

CHAPTER 7
CONCLUSIONS AND FUTURE WORK

Inspired by Cybenco and Barron's universal approximation theory, we formulate the vector-to-vector regression based on the DNN architectures in the machine learning setting. In this thesis, we focus on the representation and generalization powers of DNN based regression operators. Leveraging upon the technique of MAE decomposition, the representation power is related to the approximation error and the generalization capability is associated with the estimation error in addition to optimization error. Our contributions can be summarized as follows:

1. We first analyze the representation power of DNN based vector-to-vector regression, which is related to the upper bound on the approximation error. More specifically, we upper bound the approximation error by developing the universal approximation theory adapting to the deep learning architectures. Moreover, the experiments of speech enhancement are designed to corroborate our theorems. In particular, our derived upper bound on the approximation error can be used to estimate the practical MSE values, and the experimental results of speech enhancement can corroborate our theoretical analysis.

2. We next compare MAE with MSE as the loss function for the DNN based vector-to-vector regression. We highlight the advantages of MAE over MSE in terms of Lipschitz continuity, noise robustness, and the connections to the Laplacian distribution. Our experiments of speech enhancement are conducted to show that the DNN models equipped with the MAE loss function can result in better empirical results than the MSE as a loss function for DNN.

3. We further exploit the generalization power by employing the error decomposition

technique, where the expected loss of MAE is upper bounded by the sum of the approximation error, estimation error, and optimization error. The approximation error is associated with the representation power discussed in Chapter 3, and the estimation error and optimization error are separately discussed in Chapter 5. The estimation error can be upper bounded by deriving an upper bound on the empirical Rademacher complexity, and the optimization error relies on the setup of over-parameterization and the PL condition for DNN. Our derived upper bounds can help to analyze how to set up DNN models for regression problems, and our derived bounds can also be used to estimate practical MAE loss values. The experiments of vector-to-vector regression for image de-noising and speech enhancement to verify our theorems.

4. We also investigate how to apply TT decomposition to re-parameterize the DNN based vector-to-vector regression such that model parameters of an over-parameterized DNN can be substantially reduced. Our TT-based regression models include TT-DNN based tensor-to-vector regression and a more advanced hybrid tensor-to-vector model with CNN. Our empirical experiments of single and multi-channel speech enhancement and speech recognition demonstrate that TT-DNN with much fewer model parameters can maintain the DNN baseline performance.

Finally, our theoretical analysis of DNN-based vector-to-vector regression can be extended to future work as described in the following list:

1. Our theoretical analysis of DNN can be exploited to supervise the model pruning of DNN parameters. In particular, the existing approach of lottery ticket hypothesis (LTH) [142] exhibit the fact that a small subnet of DNN is capable of achieving even better performance than the over-parameterized DNN.

2. Although tensor-train decomposition is an efficient way to reparameterize DNN into a compressed model, it is still worth further exploration in finding the tensor networks for convolutional neural networks.

3. The theoretical study of DNN can help to facilitate the development of quantum neural networks [143], which are based on quantum computers and are composed of universal quantum circuits.

Appendices

APPENDIX A

SUPPLEMENTARY PROOFS FOR CHAPTER 4

A.1 More Discussions on the Cross-Entropy Loss Function

Although the MAE and MSE based loss functions are compared for DNN based vector-to-vector regression, the loss function based on the cross-entropy (CE) for DNN based classification still requires further discussion.

Theorem 12. *Given the set of training data* $S = \{(\boldsymbol{x}_1, \boldsymbol{y}_1), (\boldsymbol{x}_2, \boldsymbol{y}_2), ..., (\boldsymbol{x}_N, \boldsymbol{y}_N)\}$*, the CE loss function* $\mathcal{L}(S; \boldsymbol{w})$ *is* $(\frac{1}{4N} || \sum_{n=1}^{N} \boldsymbol{x}_n \boldsymbol{x}_n^T ||)$*-Lipschitz continuous function.*

Proof.

$$\begin{aligned}
\mathcal{L}(S; \mathbf{w}) &= \frac{1}{N} \sum_{n=1}^{N} \mathcal{L}\left(\mathbf{x}_n, y_n; \mathbf{w}\right) \\
&= -\frac{1}{N} \sum_{n=1}^{N} \sum_{c=1}^{C} \left(y_n \log \left(\frac{\sigma(\mathbf{w}_c^T \mathbf{x}_n)}{\sum_{c=1}^{C} \sigma(\mathbf{w}_c^T \mathbf{x}_n)} \right) \right) \\
&= -\frac{1}{N} \sum_{n=1}^{N} \sum_{c=1}^{C} \left(y_n \log \left(\sigma(\mathbf{w}_c^T \mathbf{x}_n) \right) - y_n \log \left(\sum_{c=1}^{C} \sigma(\mathbf{w}_c^T \mathbf{x}_n) \right) \right).
\end{aligned} \tag{A.1}$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}_c} = -\frac{1}{N} \sum_{n=1}^{N} y_n (1 - \sigma(\mathbf{w}_c^T \mathbf{x}_n)) \left(1 - \frac{\sigma(\mathbf{w}_c^T \mathbf{x}_n)}{\sum_{c=1}^{C} \sigma(\mathbf{w}_c^T \mathbf{x}_n)} \right) \mathbf{x}_n. \tag{A.2}$$

$$\frac{\partial^2 \mathcal{L}}{\partial \mathbf{w}_c^2} = \frac{1}{N}\sum_{n=1}^{N} y_n(1-\sigma(\mathbf{w}_c^T\mathbf{x}_n))\sigma(\mathbf{w}_c^T\mathbf{x}_n)\cdot \left[1 - \frac{\sigma(\mathbf{w}_c^T\mathbf{x}_n)}{\sum_{c=1}^{C}\sigma(\mathbf{w}_c^T\mathbf{x}_n)} + (1-\sigma(\mathbf{w}_c^T\mathbf{x}_n))\frac{\sum_{k=1,k\neq c}^{C}\sigma(\mathbf{w}_k^T\mathbf{x}_n)}{\sum_{c=1}^{C}\sigma(\mathbf{w}_c^T\mathbf{x}_n)^2}\right]\mathbf{x}_n\mathbf{x}_n^T \tag{A.3}$$

Since $t_n = (1-\sigma(\mathbf{w}_c^T\mathbf{x}_n))\sigma(\mathbf{w}_c^T\mathbf{x}_n)\left[1 - \frac{\sigma(\mathbf{w}_c^T\mathbf{x}_n)}{\sum_{c=1}^{C}\sigma(\mathbf{w}_c^T\mathbf{x}_n)} + (1-\sigma(\mathbf{w}_c^T\mathbf{x}_n))\frac{\sum_{k=1,k\neq c}^{C}\sigma(\mathbf{w}_k^T\mathbf{x}_n)}{\sum_{c=1}^{C}\sigma(\mathbf{w}_c^T\mathbf{x}_n)^2}\right] \leq \frac{1}{4}$, for any unit-length vector $\mathbf{z}$,

$$\begin{aligned}\mathbf{z}^T\frac{\partial^2 \mathcal{L}}{\partial \mathbf{w}_c^2}\mathbf{z} &= \mathbf{z}^T\left(\frac{1}{N}\sum_{n=1}^{N} y_n t_n \mathbf{x}_n\mathbf{x}_n^T\right)\mathbf{z} \\ &\leq \frac{1}{4N}\mathbf{z}^T\left(\sum_{n=1}^{N}\mathbf{x}_n\mathbf{x}_n^T\right)\mathbf{z} \\ &= \frac{1}{4N}\sum_{n=1}^{N}(\mathbf{z}^T\mathbf{x}_n)^2 \geq 0.\end{aligned} \tag{A.4}$$

Since the maximum value $\mathbf{z}^T(\sum_{n=1}^{N}\mathbf{x}_n\mathbf{x}_n^T)\mathbf{z}$ corresponds to the maximum eigenvalue of the matrix $\sum_{n=1}^{N}\mathbf{x}_n\mathbf{x}_n^T$, a Lipschitz constant for the cross-entropy is $\frac{1}{4N}||\sum_{n=1}^{N}\mathbf{x}_n\mathbf{x}_n^T||_2^2$. □

Theorem 12 suggests that the CE loss is a data-dependent Lipschitz continuous function. Given the set of training data, the Lipschitz constant is fixed and the related analysis in this chapter can be transferred to the analysis of the CE loss for DNN based classification.

APPENDIX B

SUPPLEMENTARY PROOFS FOR CHAPTER 5

We append the necessary theorems and experimental evidence in Chapter 5.

Lemma 8. *Let $\mathcal{L}_S$ denote the empirical loss function given N training samples $S = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N\}$ drawn i.i.d. from a distribution $\mathcal{D}$. For an expected MAE loss function $\mathcal{L}_\mathcal{D}$, we have that*

$$\mathbb{E}\left[\sup_{f\in\mathbb{F}_K} |\mathcal{L}_\mathcal{D}(f) - \mathcal{L}_S(f)|\right] \leq 2\hat{\mathcal{R}}_S(\mathbb{F}_K). \tag{B.1}$$

Proof. We utilize the symmetrization [144] to upper bound $\mathbb{E}\left[\sup_{f\in\mathbb{F}_K} |\mathcal{L}_\mathcal{D}(f) - \mathcal{L}_S(f)|\right]$. The symmetrization introduces a ghost dataset $S' = \{\mathbf{x}'_1, \mathbf{x}'_2, ..., \mathbf{x}'_N\}$ drawn i.i.d. from D. Let $\mathcal{L}'_S$ be the empirical risk with respect to the ghost dataset, and we assume $\mathcal{L}_{S'}(f) = \mathbb{E}_{S'}[\mathcal{L}'_{S'}(f)]$. Assuming $\mathcal{L}(f) \geq \mathcal{L}(f), \forall f \in \mathbb{F}_K$, we derive that

$$\begin{aligned}
\mathbb{E}_S\left[\sup_{f\in\mathbb{F}_K} |\mathcal{L}_\mathcal{D}(f) - \mathcal{L}_S(f)|\right] &= \mathbb{E}_S\left[\sup_{f\in\mathbb{F}_K} (\mathcal{L}_\mathcal{D}(f) - \mathcal{L}_\mathcal{S}(f))\right] \\
&= \mathbb{E}_S\left[\sup_{f\in\mathbb{F}_K} \Big(\mathbb{E}_{S'}[\mathcal{L}'_{S'}(f)] - \mathcal{L}_{S'}(f)\Big)\right] \\
&\leq \mathbb{E}_S\left[\mathbb{E}_{S'}\left[\sup_{f\in\mathbb{F}_K} \Big(\mathcal{L}'_{S'}(f) - \mathcal{L}_{S'}(f)\Big)\right]\right] \\
&\leq \mathbb{E}_{S,S'}\left[\sup_{f\in\mathbb{F}_K} \frac{1}{N}\sum_{n=1}^{N} \sigma_n(\mathcal{L}'_{S'}(f(\mathbf{x}'_n)) - \mathcal{L}_{S'}(f(\mathbf{x}_n)))\right] \\
&\leq 2\hat{\mathcal{R}}_S(\mathbb{F}_K),
\end{aligned}$$

where $\sigma_1, \sigma_2, ..., \sigma_N$ are Rademacher random variables. Similarly, the assumption of $\mathcal{L}_\mathcal{D}(f) \leq \mathcal{L}_{S'}(f), \forall f \in \mathbb{F}_K$ also brings the same result. Thus, we finish the proof of Lemma 8. □

Lemma 9 (An extension of empirical Rademacher identities)**.** *Given any sample set $S = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N\}$, and hypothesis sets $\mathbb{F}_{K,1}$, $\mathbb{F}_{K,2}$, ..., $\mathbb{F}_{K,Q}$ of functions $f^{(1)} \in \mathbb{F}_{K,1}$, $f^{(2)} \in$*

$\mathbb{F}_{K,2}$, ..., $f^{(Q)} \in \mathbb{F}_{K,Q}$ *mapping from* $\mathbb{R}^D$ *to* $\mathbb{R}^Q$, *we have that*

$$\begin{aligned}
\hat{\mathcal{R}}_S\left(\sum_{q=1}^{Q}\mathbb{F}_{K,q}\right) &= \frac{1}{N}\mathbb{E}_\sigma\left[\sup_{f^{(1)}\in\mathbb{F}_{K,1},\ldots,f^{(Q)}\in\mathbb{F}_{K,Q}}\sum_{n=1}^{N}\sigma_n\left(\sum_{q=1}^{Q}f^{(q)}(\boldsymbol{x}_n)\right)\right] \\
&= \frac{1}{N}\sum_{q=1}^{Q}\mathbb{E}_\alpha\left[\sup_{f^{(1)}\in\mathbb{F}_{K,1},\ldots,f^{(Q)}\in\mathbb{F}_{K,Q}}\sum_{n=1}^{N}\alpha_n f^{(q)}(\boldsymbol{x}_n)\right] \\
&= \sum_{q=1}^{Q}\hat{\mathcal{R}}_S(\mathbb{F}_{K,q}).
\end{aligned}$$

REFERENCES

[1] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee, "An Experimental Study on Speech Enhancement Based on Deep Neural Networks," *IEEE Signal Processing Letters*, vol. 21, no. 1, pp. 65–68, 2013.

[2] ——, "A Regression Approach to Speech Enhancement Based on Deep Neural Networks," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 23, no. 1, pp. 7–19, 2015.

[3] A. Buades, B. Coll, and J.-M. Morel, "A Review of Image Denoising Algorithms," *Multiscale modeling & simulation*, vol. 4, no. 2, pp. 490–530, 2005.

[4] M. Sánchez-Fernández, M. de-Prado-Cumplido, J. Arenas-Garcıa, and F. Pérez-Cruz, "SVM Multiregression for Nonlinear Channel Estimation in Multiple-Input Multiple-Output Systems," *IEEE Transactions on Signal Processing*, vol. 52, no. 8, pp. 2298–2307, 2004.

[5] A. El Gamal and Y.-H. Kim, *Network Information Theory*. Cambridge University Press, 2011.

[6] H. Borchani, G. Varando, C. Bielza, and P. Larranaga, "A Survey on Multi-Output Regression," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 5, no. 5, pp. 216–233, 2015.

[7] K. Pearson, "Mathematical Contributions to The Theory of Evolution," *Philosophical Transactions of the Royal Society of London*, no. 187, pp. 253–318, 1896.

[8] V. Vapnik, S. E. Golowich, A. Smola, *et al.*, "Support Vector Method for Function Approximation, Regression Estimation, and Signal Processing," *Proc. Advances in Neural Information Processing Systems*, pp. 281–287, 1997.

[9] H. Drucker, C. J. Burges, L. Kaufman, A. Smola, V. Vapnik, *et al.*, "Support Vector Regression Machines," *Proc. Advances in Neural Information Processing Systems*, vol. 9, pp. 155–161, 1997.

[10] S. R. Gunn *et al.*, "Support Vector Machines for Classification and Regression," *ISIS Technical Report*, vol. 14, no. 1, pp. 5–16, 1998.

[11] N. Simon, J. Friedman, T. Hastie, and R. Tibshirani, "A Sparse-Group Lasso," *Journal of Computational and Graphical Statistics*, vol. 22, no. 2, pp. 231–245, 2013.

[12] D. Wang and J. Chen, "Supervised Speech Separation Based on Deep Learning: An Overview," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 10, pp. 1702–1726, 2018.

[13] A. Kolmogoro, "On the Representation of Continuous Functions of Several Variables as Superpositions of Functions of Smaller Number of Variables," in *Soviet. Math. Dokl*, vol. 108, 1956, pp. 179–182.

[14] V. Tikhomirov, "On the Representation of Continuous Functions of Several Variables as Superpositions of Continuous Functions of One Variable and Addition," *Selected Works of AN Kolmogorov*, pp. 383–387, 1991.

[15] G. Cybenko, "Approximation by Superpositions of A Sigmoidal Function," *Mathematics of control, signals and systems*, vol. 2, no. 4, pp. 303–314, 1989.

[16] A. R. Barron, "Universal Approximation Bounds for Superpositions of A Sigmoidal Function," *IEEE Transactions on Information Theory*, vol. 39, no. 3, pp. 930–945, 1993.

[17] J. J. Hopfield, "Artificial Neural Networks," *IEEE Circuits and Devices Magazine*, vol. 4, no. 5, pp. 3–10, 1988.

[18] H. K. Kwan, "Simple Sigmoid-Like Activation Function Suitable for Digital Hardware Implementation," *Electronics Letters*, vol. 28, no. 15, pp. 1379–1380, 1992.

[19] K. Hornik, M. Stinchcombe, and H. White, "Multilayer Feed-Forward Networks Are Universal Approximators," *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989.

[20] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, *et al.*, "Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups," *IEEE Signal processing magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[21] M. Gabbouj and E. J. Coyle, "Minimum Mean Absolute Error Stack Filtering With Structural Constraint and Goals," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 38, no. 6, pp. 955–968, 1990.

[22] B. V. Kumar, A. Mahalanobis, S. Song, S. R. F. Sims, and J. F. Epperson, "Minimum Squared Error Synthetic Discriminant Functions," *Optical Engineering*, vol. 31, no. 5, pp. 915–922, 1992.

[23] I. V. Oseledets, "Tensor-Train Decomposition," *SIAM Journal on Scientific Computing*, vol. 33, no. 5, pp. 2295–2317, 2011.

[24] J. Takeuchi and Y. Kosugi, "Neural Network Representation of Finite Element Method," *Neural Networks*, vol. 7, no. 2, pp. 389–395, 1994.

[25] T. Abatzoglou, "The Lipschitz continuity of the metric projection," *Journal of Approximation Theory*, vol. 26, no. 3, pp. 212–218, 1979.

[26] V. Vapnik, *The Nature of Statistical Learning Theory*. Springer Science & Business Media, 1999.

[27] H. Karimi, J. Nutini, and M. Schmidt, "Linear Convergence of Gradient and Proximal-Gradient Methods Under The Polyak-Łojasiewicz Condition," in *Proc. Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, Springer, 2016, pp. 795–811.

[28] L. Chizat and F. Bach, "On the Global Convergence of Gradient Descent for Over-Parameterized Models Using Optimal Transport," in *Proc. Advances in Neural Information Processing Systems*, 2018, pp. 3036–3046.

[29] P. C. Loizou, *Speech Enhancement: Theory and Practice*. CRC press, 2007.

[30] J. Benesty, S. Makino, and J. Chen, *Speech Enhancement*. Springer Science & Business Media, 2006.

[31] B. H. Juang and L. R. Rabiner, "Hidden Markov Models for Speech Recognition," *Technometrics*, vol. 33, no. 3, pp. 251–272, 1991.

[32] C.-H. Lee, F. K. Soong, and K. K. Paliwal, *Automatic speech and speaker recognition: advanced topics*. Springer Science & Business Media, 2012, vol. 355.

[33] D. Yu and L. Deng, *Automatic Speech Recognition*. Springer, 2016.

[34] J. Qi, J. Du, S. M. Siniscalchi, and C.-H. Lee, "A Theory on Deep Neural Network Based Vector-to-Vector Regression With an Illustration of Its Expressive Power in Speech Enhancement," *IEEE/ACM Transactions on Audio, Speech, and Language Processing (TASLP)*, vol. 27, no. 12, pp. 1932–1943, 2019.

[35] J. Qi, J. Du, S. M. Siniscalchi, X. Ma, and C.-H. Lee, "On Mean Absolute Error for Deep Neural Network based Vector-to-Vector Regression," *IEEE Signal Processing Letters*, vol. 27, pp. 1485–1489, 2020.

[36] ——, "Analyzing Upper Bounds on Mean Absolute Errors for Deep Neural Network based Vector-to-Vector Regression," *IEEE Transactions on Signal Processing*, vol. 68, pp. 3411–3422, 2020.

[37] J. Qi, H. Hu, Y. Wang, C. H. Yang, S. M. Siniscalchi, and C.-H. Lee, "Tensor-Train Network based Tensor-to-Vector Regression for Multi-Channel Speech Enhancement," in *Proc. International Conference on Acoustics, Speech, and Signal Processing*, 2020, pp. 7504–7508.

[38] J. Qi, H. Hu, Y. Wang, C.-H. H. Yang, S. M. Siniscalchi, and C.-H. Lee, "Exploring Deep Hybrid Tensor-to-Vector Network Architectures for Regression Based Speech Enhancement," in *Proc. Interspeech*, 2020, pp. 76–80.

[39] M. Imaizumi, T. Maehara, and K. Hayashi, "On Tensor-Train Rank Minimization: Statistical Efficiency and Scalable Algorithm," *arXiv preprint arXiv:1708.00132*, 2017.

[40] J. Schmidt-Hieber, "The Kolmogorov–Arnold Representation Theorem Revisited," *Neural Networks*, vol. 137, pp. 119–126, 2021.

[41] A. R. Barron, "Approximation and Estimation Bounds for Artificial Neural Networks," *Machine learning*, vol. 14, no. 1, pp. 115–133, 1994.

[42] J. Han and C. Moraga, "The Influence of The Sigmoid Function Parameters On The Speed of Backpropagation Learning," in *International Workshop on Artificial Neural Networks*, Springer, 1995, pp. 195–201.

[43] J. Du and Y. Xu, "Hierarchical deep neural network for multivariate regression," *Pattern Recognition*, vol. 63, pp. 149–157, 2017.

[44] F. Weninger, H. Erdogan, S. Watanabe, E. Vincent, J. Le Roux, J. R. Hershey, and B. Schuller, "Speech Enhancement with LSTM Recurrent Neural Networks and Its Application to Noise-Robust ASR," in *Proc. International Conference on Latent Variable Analysis and Signal Separation*, Springer, 2015, pp. 91–99.

[45] S. Pascual, A. Bonafonte, and J. Serra, "SEGAN: Speech Enhancement Generative Adversarial Network," *arXiv preprint arXiv:1703.09452*, 2017.

[46] H. Zhao, S. Zarar, I. Tashev, and C.-H. Lee, "Convolutional-Recurrent Neural Networks for Speech Enhancement," in *Proc. International Conference on Acoustics, Speech and Signal Processing*, 2018, pp. 2401–2405.

[47] C. Valentini-Botinhao, X. Wang, S. Takaki, and J. Yamagishi, "Investigating RNN-based Speech Enhancement Methods for Noise-Robust Text-to-Speech," in *SSW*, 2016, pp. 146–152.

[48] A. W. Rix, J. G. Beerends, M. P. Hollier, and A. P. Hekstra, "Perceptual Evaluation of Speech Quality (PESQ)-A New Method for Speech Quality Assessment of

Telephone Networks and Codecs," in *Proc. International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, 2001, pp. 749–752.

[49] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen, "A Short-Time Objective Intelligibility Measure for Time-Frequency Weighted Noisy Speech," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, 2010, pp. 4214–4217.

[50] Y. Yang, D. Krompass, and V. Tresp, "Tensor-Train Recurrent Neural Networks for Video Classification," in *Proc. International Conference on Machine Learning*, 2017, pp. 3891–3900.

[51] D. Perez-Garcia, F. Verstraete, M. Wolf, and J. Cirac, "Matrix Product State Representations," *Quantum Information & Computation*, vol. 7, no. 5-6, pp. 401–430, 2007.

[52] N. D. Sidiropoulos, L. De Lathauwer, X. Fu, K. Huang, E. E. Papalexakis, and C. Faloutsos, "Tensor Decomposition for Signal Processing and Machine Learning," *IEEE Transactions on Signal Processing*, vol. 65, no. 13, pp. 3551–3582, 2017.

[53] Y.-D. Kim and S. Choi, "Nonnegative Tucker Decomposition," in *Proc. Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–8.

[54] A. H. Phan and A. Cichocki, "PARAFAC Algorithms for Large-Scale Problems," *Neurocomputing*, vol. 74, no. 11, pp. 1970–1984, 2011.

[55] N. K. M. Faber, R. Bro, and P. K. Hopke, "Recent Developments in CANDECOMP / PARAFAC Algorithms: A Critical Review," *Chemometrics and Intelligent Laboratory Systems*, vol. 65, no. 1, pp. 119–137, 2003.

[56] Y. Zniyed, R. Boyer, A. L. de Almeida, and G. Favier, "Multidimensional Harmonic Retrieval Based on Vandermonde Tensor-Train," *Signal Processing*, vol. 163, pp. 75–86, 2019.

[57] A. Novikov, A. Rodomanov, A. Osokin, and D. Vetrov, "Putting MRFs on A Tensor-Train," in *Proc. International Conference on Machine Learning*, 2014, pp. 811–819.

[58] J. Zhang, X. Ma, J. Qi, and S. Jin, "Designing Tensor-Train Deep Neural Networks For Time-Varying MIMO Channel Estimation," *IEEE Journal of Selected Topics in Signal Processing*, vol. 15, no. 3, pp. 759–773, 2021.

[59] X. Wang, L. T. Yang, Y. Wang, X. Liu, Q. Zhang, and M. J. Deen, "A Distributed Tensor-Train Decomposition Method for Cyber-Physical-Social Services," *ACM Transactions on Cyber-Physical Systems*, vol. 3, no. 4, pp. 1–15, 2019.

[60] R. Hecht-Nielsen, "Theory of The Backpropagation Neural Network," in *Neural Networks for Perception*, Elsevier, 1992, pp. 65–93.

[61] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proc. COMPSTAT*, Springer, 2010, pp. 177–186.

[62] D. C. Plaut and G. E. Hinton, "Learning Sets of Filters Using Back-Propagation," *Computer Speech & Language*, vol. 2, no. 1, pp. 35–61, 1987.

[63] M. C. Mukkamala and M. Hein, "Variants of RMSProp and Adagrad with Logarithmic Regret Bounds," in *Proc. International Conference on Machine Learning*, PMLR, 2017, pp. 2545–2553.

[64] R. Ward, X. Wu, and L. Bottou, "AdaGrad Stepsizes: Sharp Convergence Over Nonconvex Landscapes," in *Proc. International Conference on Machine Learning*, 2019, pp. 6677–6686.

[65] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[66] H. N. Mhaskar and T. Poggio, "Deep vs. Shallow Networks: An Approximation Theory Perspective," *Analysis and Applications*, vol. 14, no. 06, pp. 829–848, 2016.

[67] B. Hanin, "Universal Function Approximation by Deep Neural Nets with Bounded Width and ReLU Activations," *Mathematics*, vol. 7, no. 10, 2019.

[68] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee, "A Regression Approach to Speech Enhancement Based on Deep Neural Networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 1, pp. 7–19, 2014.

[69] ——, "An Experimental Study on Speech Enhancement Based on Deep Neural Networks," *IEEE Signal processing letters*, vol. 21, no. 1, pp. 65–68, 2013.

[70] M. McLaren, L. Ferrer, and A. Lawson, "Exploring the Role of Phonetic Bottleneck Features for Speaker and Language Recognition," in *Proc. International Conference on Acoustics, Speech and Signal Processing*, 2016, pp. 5575–5579.

[71] X. Hou and L. Zhang, "Saliency Detection: A Spectral Residual Approach," in *Proc. Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–8.

[72] J. Qi, D. Wang, J. Xu, and J. Tejedor Noguerales, "Bottleneck Features based on Gammatone Frequency Cepstral Coefficients," in *Proc. Interspeech*, 2013, pp. 1751–1755.

[73] J. Qi, D. Wang, and J. Tejedor Noguerales, "Subspace Models for Bottleneck Features," in *Proc. Interspeech*, 2013, pp. 1746–1750.

[74] T. Toda, A. W. Black, and K. Tokuda, "Spectral Conversion Based on Maximum Likelihood Estimation Considering Global Variance of Converted Parameter," in *Proc. International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, 2005, pp. I–9.

[75] Y. Hirose, K. Yamashita, and S. Hijiya, "Back-propagation Algorithm Which Varies the Number of Hidden Units," *Neural Networks*, vol. 4, no. 1, pp. 61–66, 1991.

[76] M. L. Seltzer, D. Yu, and Y. Wang, "An Investigation of Deep Neural Networks for Noise Robust Speech Recognition," in *Proc. International Conference on Acoustics, Speech and Signal Processing*, 2013, pp. 7398–7402.

[77] V. Zue, S. Seneff, and J. Glass, "Speech Database Development at MIT: TIMIT and Beyond," *Speech communication*, vol. 9, no. 4, pp. 351–356, 1990.

[78] A. Varga and H. J. Steeneken, "Assessment for Automatic Speech Recognition: II. NOISEX-92: A Database and An Experiment to Study the Effect of Additive Noise on Speech Recognition Systems," *Speech Communication*, vol. 12, no. 3, pp. 247–251, 1993.

[79] A. W. Rix, J. G. Beerends, M. P. Hollier, and A. P. Hekstra, "Perceptual Evaluation of Speech Quality (PESQ)-A New Method for Speech Quality Assessment of Telephone Networks and Codecs," in *Proc. International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, 2001, pp. 749–752.

[80] E. J. Coyle and J.-H. Lin, "Stack Filters and The Mean Absolute Error Criterion," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 36, no. 8, pp. 1244–1254, 1988.

[81] C. Willmott, S. Ackleson, R. Davis, J. Feddema, K. Klink, D. Legates, J. O'donnell, and C. Rowe, "Statistics for The Evaluation of Model Performance," *J. Geophys. Res*, vol. 90, no. C5, pp. 8995–9005, 1985.

[82] H. Borchani, G. Varando, C. Bielza, and P. Larrañaga, "A Survey on Multi-Output Regression," *Mathematical Methods in the Applied Sciences*, vol. 5, no. 5, pp. 216–233, 2015.

[83] E. V. Slud and T. Maiti, "Mean-Squared Error Estimation in Transformed Fay–Herriot Models," *Journal of the Royal Statistical Society: Series B*, vol. 68, no. 2, pp. 239–257, 2006.

[84] T. Chai and R. R. Draxler, "Root Mean Square Error (RMSE) or Mean Absolute Error (MAE)?–Arguments Against Avoiding RMSE in the Literature," *Geoscientific Model Development*, vol. 7, no. 3, pp. 1247–1250, 2014.

[85] C. J. Willmott and K. Matsuura, "Advantages of The Mean Absolute Error (MAE) Over The Root Mean Square Error (RMSE) in Assessing Average Model Performance," *Climate Research*, vol. 30, no. 1, pp. 79–82, 2005.

[86] C. J. Willmott, K. Matsuura, and S. M. Robeson, "Ambiguities Inherent in Sums-of-Squares-Based Error Statistics," *Atmospheric Environment*, vol. 43, no. 3, pp. 749–752, 2009.

[87] D. Wallach and B. Goffinet, "Mean Squared Error of Prediction as A Criterion for Evaluating and Comparing System Models," *Ecological Modelling*, vol. 44, no. 3-4, pp. 299–306, 1989.

[88] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. Springer New York Inc., 2001.

[89] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[90] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of Machine Learning*. MIT Press, 2018.

[91] S. Shalev-Shwartz and S. Ben-David, *Understanding Machine Learning: From Theory to Algorithms*. Cambridge university press, 2014.

[92] J. O. Berger, *Statistical Decision Theory and Bayesian Analysis*. Springer Science & Business Media, 2013.

[93] V. N. Vapnik, *The Nature of Statistical Learning Theory*. Springer-Verlag New York, Inc., 1995.

[94] N. Li, L. Wang, X. Li, and Q. Zhu, "An Effective Deep Learning Neural Network Model for Short-Term Load Forecasting," *Concurrency and Computation: Practice and Experience*, vol. 32, Jan. 2020.

[95] E. Imani and M. White, "Improving Regression Performance with Distributional Losses," in *Prof. International Conference on Machine Learning*, 2018, pp. 2157–2166.

[96] A. Pandey and D. Wang, "On Adversarial Training and Loss Functions for Speech Enhancement," in *Proc. International Conference on Acoustics, Speech and Signal Processing*, 2018, pp. 5414–5418.

[97] V. N. Vapnik and A. Y. Chervonenkis, "On the Uniform Convergence of Relative Frequencies of Events to Their Probabilities," *Theory of Probability and Its Applications*, vol. 16, no. 2, pp. 264–280, 2018.

[98] Z. Charles and D. Papailiopoulos, "Stability and Generalization of Learning Algorithms That Converge to Global Optima," *arXiv preprint arXiv:1710.08402*, 2017.

[99] A. Lorencs, I. Mednieks, and J. Sinica-Sinavskis, "Biomedical Image Processing Based on Regression Models," in *Proc. Nordic-Baltic Conference on Biomedical Engineering and Medical Physics*, 2008, pp. 536–539.

[100] H. Takeda, S. Farsiu, and P. Milanfar, "Kernel Regression for Image Processing and Reconstruction," *IEEE Transactions on Image Processing*, vol. 16, no. 2, pp. 349–366, 2007.

[101] J. Fan, C. Ma, and Y. Zhong, "A Selective Overview of Deep Learning," *Statistical Science*, vol. 36, no. 2, pp. 264–290, 2021.

[102] J. Zhu, B. R. Gibson, and T. T. Rogers, "Human Rademacher Complexity," in *Proc. Advances in Neural Information Processing Systems*, 2009, pp. 2322–2330.

[103] M. J. Wainwright, *High-Dimensional Statistics: A Non-Asymptotic Viewpoint*. Cambridge University Press, 2019, vol. 48.

[104] R. Paulavičius and J. Žilinskas, "Analysis of Different Norms and Corresponding Lipschitz Constants for Global Optimization," *Technological and Economic Development of Economy*, vol. 12, no. 4, pp. 301–306, 2006.

[105] M. Fazlyab, A. Robey, H. Hassani, M. Morari, and G. Pappas, "Efficient and Accurate Estimation of Lipschitz Constants for Deep Neural Networks," in *Proc. Advances in Neural Information Processing Systems*, 2019, pp. 11 423–11 434.

[106] L. Chai, J. Du, Q.-F. Liu, and C.-H. Lee, "Using Generalized Gaussian Distributions to Improve Regression Error Modeling for Deep Learning-Based Speech Enhancement," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 12, pp. 1919–1931, 2019.

[107] L. Deng, J. Droppo, and A. Acero, "Enhancement of Log-Mel Power Spectra of Speech Using A Phase-Sensitive Model of The Acoustic Environment and Sequential Estimation of The Corrupting Noise," *IEEE Transactions on Speech and Audio Processing*, vol. 12, no. 2, pp. 133–143, 2004.

[108] J. Qi, D. Wang, Y. Jiang, and R. Liu, "Auditory Features Based on Gammatone Filters for Robust Speech Recognition," in *Proc. IEEE International Symposium on Circuits and Systems*, 2013, pp. 305–308.

[109] H. Silén, E. Helander, J. Nurminen, and M. Gabbouj, "Ways to Implement Global Variance in Statistical Speech Synthesis," in *Proc. Interspeech*, 2012, pp. 1436–1439.

[110] J. Qi, D. Wang, J. Xu, and J. Tejedor Noguerales, "Bottleneck Features Based on Gammatone Frequency Cepstral Coefficients," in *Proc. Interspeech*, 2013, pp. 1751–1755.

[111] J. Qi and J. Tejedor, "Robust Submodular Data Partitioning for Distributed Speech Recognition," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, 2016, pp. 2254–2258.

[112] B. Neyshabur, R. Tomioka, and N. Srebro, "Norm-based Capacity Control in Neural Networks," in *Proc. Conference on Learning Theory*, 2015, pp. 1376–1401.

[113] P. L. Bartlett, D. J. Foster, and M. J. Telgarsky, "Spectrally-Normalized Margin Bounds for Neural Networks," in *Proc. Advances in Neural Information Processing Systems*, 2017, pp. 6240–6249.

[114] N. Golowich, A. Rakhlin, and O. Shamir, "Size-Independent Sample Complexity of Neural Networks," in *Proc. International Conference On Learning Theory*, vol. 75, 2018, pp. 297–299.

[115] S. Mei, A. Montanari, and P.-M. Nguyen, "A Mean Field View of The Landscape of Two-layer Neural Networks," *the National Academy of Sciences*, vol. 115, no. 33, E7665–E7671, 2018.

[116] P. L. Bartlett and S. Mendelson, "Rademacher and Gaussian Complexities: Risk Bounds and Structural Results," *Journal of Machine Learning Research*, vol. 3, no. Nov, pp. 463–482, 2002.

[117] S. Du and J. Lee, "On the Power of Over-parametrization in Neural Networks with Quadratic Activation," in *Proc. International Conference on Machine Learning*, vol. 80, 2018, pp. 1329–1338.

[118] B. Neyshabur, Z. Li, S. Bhojanapalli, Y. LeCun, and N. Srebro, "Towards understanding the role of over-parametrization in generalization of neural networks," in *Proc. International Conference of Learning Representation*, 2019.

[119] Z. Allen-Zhu, Y. Li, and Z. Song, "A Convergence Theory for Deep Learning via Over-Parameterization," in *Proc. International Conference on Machine Learning*, vol. 97, 2019, pp. 242–252.

[120] Y. Li and Y. Liang, "Learning Over-Parameterized Neural Networks via Stochastic Gradient Descent on Structured Data," in *Proc. Advances in Neural Information Processing Systems*, 2018, pp. 8157–8166.

[121] O. Bousquet and A. Elisseeff, "Stability and Generalization," *Journal of Machine Learning Research*, vol. 2, pp. 499–526, 2002.

[122] C. Yang, J. Qi, P. Chen, X. Ma, and C. Lee, "Characterizing Speech Adversarial Examples Using Self-Attention U-Net Enhancement," in *Proc. International Conference on Acoustics, Speech and Signal Processing*, 2020, pp. 3107–3111.

[123] L. Devroye, L. Györfi, and G. Lugosi, *A Probabilistic Theory of Pattern Recognition*. Springer Science & Business Media, 2013, vol. 31.

[124] V. Vapnik, *The Nature of Statistical Learning Theory*. Springer science & business media, 2013.

[125] ——, "Principles of Risk Minimization for Learning Theory," in *Proc. Advances in Neural Information Processing Systems*, 1992, pp. 831–838.

[126] Z. Allen-Zhu, Y. Li, and Y. Liang, "Learning and Generalization in Over-parameterized Neural Networks, Going Beyond Two Layers," in *Proc. Advances in Neural Information Processing Systems*, 2019, pp. 6158–6169.

[127] S. Vaswani, F. Bach, and M. Schmidt, "Fast and Faster Convergence of SGD for Over-Parameterized Models and An Accelerated Perceptron," in *Proc. International Conference on Artificial Intelligence and Statistics*, 2019.

[128] W. Hoeffding, "Probability Inequalities for Sums of Bounded Random Variables," in *The Collected Works of Wassily Hoeffding*, 1994, pp. 409–426.

[129] B. Hanin, "Universal Function Approximation by Deep Neural Nets With Bounded Width and ReLU Activations," *Mathematics*, vol. 7, no. 10, p. 992, 2019.

[130] R. Bassily, M. Belkin, and S. Ma, "On Exponential Convergence of SGD in Non-Convex Over-parametrized Learning," *arXiv preprint arXiv:1811.02564*, 2018.

[131] L. Deng, "The MNIST Database of Handwritten Digit Images for Machine Learning Research," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, 2012.

[132] T. Toda, A. W. Black, and K. Tokuda, "Spectral Conversion Based on Maximum Likelihood Estimation Considering Global Variance of Converted Parameter," in *Proc. International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, 2005, pp. I–9.

[133] J. Qi, D. Wang, and J. Tejedor Noguerales, "Subspace Models for Bottleneck Features," in *Proc. Interspeech*, 2013, pp. 1746–1750.

[134] S. Loffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," in *International Conference on Machine Learning*, 2015, pp. 448–456.

[135] P. Warden, "Speech commands: A dataset for limited-vocabulary speech recognition," *arXiv preprint arXiv:1804.03209*, 2018.

[136] B. McMahan and D. Rao, "Listening to The World Improves Speech Command Recognition," in *Proc. AAAI Conference on Artificial Intelligence*, vol. 32, 2018.

[137] D. C. de Andrade, S. Leo, M. L. D. S. Viana, and C. Bernkopf, "A Neural Attention Model for Speech Command Recognition," *arXiv preprint arXiv:1808.08929*, 2018.

[138] C.-H. Yang, J. Qi, P.-Y. Chen, X. Ma, and C.-H. Lee, "Characterizing Speech Adversarial Examples Using Self-Attention U-Net Enhancement," in *Proc. International Conference on Acoustics, Speech and Signal Processing*, 2020, pp. 3107–3111.

[139] C.-H. H. Yang, J. Qi, S. Y.-C. Chen, P.-Y. Chen, S. M. Siniscalchi, X. Ma, and C.-H. Lee, "Decentralizing Feature Extraction with Quantum Convolutional Neural Network for Automatic Speech Recognition," in *Proc. International Conference on Acoustics, Speech and Signal Processing*, 2021, pp. 6523–6527.

[140] A. Berg, M. O'Connor, and M. T. Cruz, "Keyword transformer: A self-attention model for keyword spotting," *arXiv preprint arXiv:2104.00769*, 2021.

[141] D. Seo, H.-S. Oh, and Y. Jung, "Wav2kws: Transfer learning from speech representations for keyword spotting," *IEEE Access*, vol. 9, pp. 80 682–80 691, 2021.

[142] J. Frankle and M. Carbin, "The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks," *arXiv preprint arXiv:1803.03635*, 2018.

[143] K. Beer, D. Bondarenko, T. Farrelly, T. J. Osborne, R. Salzmann, D. Scheiermann, and R. Wolf, "Training Deep Quantum Neural Networks," *Nature Communications*, vol. 11, no. 1, pp. 1–6, 2020.

[144] R. Vershynin, *High-dimensional Probability: An Introduction with Applications in Data Science*. Cambridge University Press, 2018, vol. 47.

VITA

Jun Qi received his B.S. degree from Applied Mathematics, Beijing Normal University in 2010 and his M.S. degree from the Department of Electronic Engineering, Tsinghua University in 2013. He also obtained another Master's degree in Electrical Engineering from the University of Washington in 2017. He was a graduate research intern with Microsoft Research, Tencent American AI Lab, and Mitsubishi Electric Research Labs in 2017, 2019, and 2020, respectively.

In addition to the "theoretical error performance analysis for deep neural networks based regression functional approximation" work presented in this thesis, Jun Qi is also actively engaged in the research regarding quantum tensor networks for machine learning and speech processing. The publications during his Ph.D. study at the Georgia Institute of Technology are listed below.

1. **Jun Qi**, Javier Tejedor, "Classical-to-Quantum Hybrid Transfer Learning for Spoken Command Recognition based on Quantum Neural Networks," in Proc. IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing (ICASSP), 2022.

2. **Jun Qi**, Javier Tejedor, "Exploiting Hybrid Models of Tensor-Train Networks for Spoken Command Recognition," in Proc. IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing (ICASSP), 2022.

3. Huck Yang, **Jun Qi**, Yen-Chi Chen, Yu Tsao, Pin-Yu Chen, "When BERT Meets Quantum Temporal Convolutional Learning for Text Classification in Heterogeneous Computing," in Proc. IEEE Intl. Conf. Acoustic, Speech, and Signal Processing (ICASSP), 2022.

4. **Jun Qi**, Huck Yang, Pin-Yu Chen, "QTN-VQC: An End-to-End Learning Framework for Quantum Neural Networks," in NeurIPS Workshop on Quantum Tensor

Networks in Machine Learning, 2021.

5. Jing Zhang, Xiaoli Ma, **Jun Qi**, Shi Jin, "Tensor-Train Deep Neural Network Based Channel Estimation Over Time-Varying MIMO Channels," in IEEE Journal of Selected Topics in Signal Processing (JSTSP), 2021.

6. Huck Yang, **Jun Qi**, Yen-Chi Chen, Pin-Yu Chen, Sabato Marco Siniscalchi, Xiaoli Ma, Chin-Hui Lee, "Decentralizing Feature Extraction with Quantum Convolutional Neural Network for Automatic Speech Recognition," in Proc. IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing (ICASSP), 2021.

7. **Jun Qi**, Jun Du, Sabato Marco Siniscalchi, Xiaoli Ma, Chin-Hui Lee, "Analyzing Upper Bounds on Mean Absolute Errors for Deep Neural Network Based Vector-to-Vector Regression," in IEEE Transactions on Signal Processing (TSP), Vol 68, pp. 3411-3422, 2020.

8. **Jun Qi**, Jun Du, Sabato Marco Siniscalchi, Xiaoli Ma, Chin-Hui Lee, "On Mean Absolute Error for Deep Neural Network-based Vector-to-Vector Regression," in IEEE Signal Processing Letters (SPL), Vol. 27, pp. 1485-1489, 2020.

9. Yeh-Chi Chen, Huck Yang, **Jun Qi**, Pin-Yu Chen, Xiaoli Ma, Hsi-Sheng Goan, "Variational Quantum Circuits for Deep Reinforcement Learning," IEEE Access, Vol. 8, pp. 141007-141024, 2020.

10. **Jun Qi**, Hu Hu, Yannan Wang, Chao-Han Huck Yang, Sabato Marco Siniscalchi, Chin-Hui Lee, "Tensor-to-Vector Regression for Multi-Channel Speech Enhancement based on Tensor-Train Network," in Proc. IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing (ICASSP), 2020.

11. **Jun Qi**, Huck Yang, Javier Tejedor, "Submodular Rank Aggregation for Score-based Permutations for Distributed Automatic Speech Recognition," in Proc. IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing (ICASSP), 2020.

12. Huck Yang, **Jun Qi**, Pin-Yu Chen, Xiaoli Ma, Chin-Hui Lee, "Characterizing Speech Adversarial Examples Using Self-Attention U-Net Enhancement," in Proc. IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing (ICASSP), 2020.

13. **Jun Qi**, Hu Hu, Huck Yang, Sabato Marco Siniscalchi, Chin-Hui Lee, "Exploring Deep Hybrid Tensor-to-Vector Network Architectures for Regression-based Speech Enhancement," in Proc. INTERSPEECH, 2020.

14. **Jun Qi**, Jun Du, Sabato Marco Siniscalchi, Chin-Hui Lee, "A Theory on Deep Neural Network-based Vector-to-Vector Regression with an Illustration of Its Expressive Power in Speech Enhancement," in IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP), Vol 27, no. 12, pp. 1932-1943, 2019.

15. Huck Yang, **Jun Qi**, Pin-Yu Chen, Xiaoli Ma, "Enhanced Adversarial Strategically-Timed Attacks on Deep Reinforcement Learning," in NeurIPS Workshop on Robot Learning, 2019.

16. **Jun Qi**, Xu Liu, Shunsuke Kamijo, Javier Tejedor, "Distributed Submodular Maximization for Large-scale Vocabulary Continuous Speech Recognition," in Proc. IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing (ICASSP), 2018.