A Theory on Deep Neural Network Based Vector-to-Vector Regression With an Illustration of Its Expressive Power in Speech Enhancement

Jun Qi, *Student Member, IEEE*, Jun Du, *Member, IEEE*, Sabato Marco Siniscalchi, *Senior Member, IEEE*, and Chin-Hui Lee, *Fellow, IEEE*

***Abstract*—This paper focuses on a theoretical analysis of deep neural network (DNN) based functional approximation. Leveraging upon two classical theorems on universal approximation, an artificial neural network (ANN) with a single hidden layer of neurons is used. With modified ReLU and Sigmoid activation functions, we first generalize the related concepts to vector-to-vector regression. Then, we show that the width of the hidden layer of ANN is numerically related to the approximation of the regression function. Furthermore, we increase the number of hidden layers and show that the depth of the ANN-based regression function can enhance its expressive power. We illustrate this representation with recently-emerged DNN based speech enhancement. We first compare the expressive power by varying ANN structures and then test its related regression performance under different noisy conditions in various noise types and signal-to-noise-ratio levels. Experimental results verify our theoretical prediction that an ANN of a broader hidden layer and a deeper architecture can jointly ensure a closer approximation of the vector-to-vector regression functions in terms of the Euclidean distance between the log power spectra of noisy and expected clean speech. Moreover, a DNN with a broader width at the top hidden layer can improve the regression performance relative to those with a narrower width at the top hidden layers.**

***Index Terms*—Deep neural network, universal approximation, speech enhancement, vector-to-vector regression, expressive power.**

I. Introduction

VECTOR-TO-VECTOR regression, also known as multivariate regression [1], is the problem of finding a regression model for predicting multiple responses (output vector) from some features (input vector). More formally, given a d-dim input vector space X and a measurable q-dim output vector space Y, the goal is to learn a functional relationship $f : X \to Y$ such that the output vectors approximate some desirable values. The whole process is summarized in Eq. (1), where $\mathbf{e}$ refers to the approximation error vector:

$$\mathbf{y} = f(\mathbf{x}) + \mathbf{e}. \tag{1}$$

In supervised learning, we receive a labeled set of sample vector pairs $S = ((\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_I, \mathbf{y}_I)) \in (X \times Y)^I$ with $\mathbf{x}_1, \ldots, \mathbf{x}_I$ drawn independently and identically distributed (iid) from a fixed distribution D, and real-valued vector targets $\mathbf{y}_i = f(\mathbf{x}_i)$. We need to find a regression function f best mapping input samples to the targets with a minimum error.

The problems of vector-to-vector regression have been studied extensively in the modern engineering community. For example, speech enhancement aims at designing mapping functions to transform vectors of noisy speech features into corresponding vectors of clean speech features [2], [3]. Similarly, image de-noising tries to generate clean vectors of images from the corrupted ones in different natural scenes [4]. Recently, end-to-end spoken language translation even translates spectrograms of one language into another [5]. A common goal of these tasks is to obtain expected output vectors from the input ones in a vector-to-vector mapping. Without loss of generality, this work demonstrates a theoretical analysis concerning the universal approximation of deep models in the context of vector-to-vector regression with an illustration in speech enhancement. Speech enhancement is a useful illustration of a general theoretical analysis because it is an unbounded conversion from $\mathbb{R}^d$ to $\mathbb{R}^q$. However, the classification tasks, such as speech recognition and face recognition, are bounded vector-to-vector regressions from $\mathbb{R}^d \to [0, 1]^q$. In this sense, the classification tasks can be considered as special cases of regression ones.

Over the years, several machine learning approaches based on different vector-to-vector regression functions have been proposed. Linear regression admits a straightforward implementation, which originated from Pearson's research on the theory of evolution [6]; however, the limited expressive power of linear regression limits the utility of its strong generalization guarantee.

Manuscript received January 10, 2019; revised July 1, 2019 and August 12, 2019; accepted August 14, 2019. Date of publication August 19, 2019; date of current version August 30, 2019. This work was supported in part by the National Natural Science Foundation of China under Grant 61671422. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Tuomas Virtanen. *(Corresponding author: Jun Qi.)*

J. Qi and C.-H. Lee are with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332 USA (e-mail: qij41@gatech.edu; chl@ece.gatech.edu).

J. Du is with the National Engineering Laboratory for Speech and Language Information Processing, University of Science and Technology of China, Hefei 230027, China (e-mail: jundu@ustc.edu.cn).

S. M. Siniscalchi is with the Faculty of Architecture and Engineering, University of Enna "Kore", Enna 94100, Italy, and also with the Georgia Institute of Technology, Atlanta, GA 30332 USA (e-mail: marco.siniscalchi@unikore.it).

Digital Object Identifier 10.1109/TASLP.2019.2935891

Support vector regression (SVR) is a kernel-based algorithm and maintains all the features of support vector machine (SVM) for classification because SVR aims at solving a max-margin optimization problem [7]. However, a significant drawback of SVR is its high computational requirements when dealing with large training sets [8]. Lasso [9] and Group Lasso [10] based regularization methods can also be used to define the regression algorithm, where it may be desirable to find a sparse solution that selects or omits entire subsets of features. Unfortunately, the regularization based methods do not admit a natural use of kernels, which prevents their extension to non-linear vector-to-vector regression [11].

With the resurgence of artificial neural networks (ANNs, [12]) in machine learning, our previous studies on speech enhancement [2], [13], [14] and the related works [15]–[18], demonstrate that deep neural networks (DNNs) [19] with multiple hidden layers, offer an efficient and robust solution to dealing with large-scale vector-to-vector regression problems. More specifically, a feed-forward DNN with three hidden layers is taken as a regression function to map high dimensional input vectors to the target ones. Notably, the empirical results of speech enhancement [2], [13] have demonstrated DNNs outperform shallow neural networks.

This work focuses on a theoretical analysis of the universal approximation property of ANNs in the context of deep models used for vector-to-vector regression. Moreover, our theoretical analysis of the expressive power of such a regression will be validated on speech enhancement tasks. Although some related theoretical works [20], [21], [22] have justified the use of ANNs in their universal capability of approximating different function classes, this aspect has not been demonstrated in vector-to-vector regression. This work aims at filling this theoretical gap. Our study builds on classical theories including Kolmogorov representation theorem [23], Cybenko universal approximation theorem [24] and Barron universal approximation bounds [25].

Regarding the theoretical aspect, we generalize the classical theorems to interpret the representation power of the DNN based vector-to-vector regression. First, classical theories ensure that ANN can represent an arbitrary vector-to-vector regression function by associating the width of the sigmoidal based hidden layer with expressive power. Moreover, we present new theories that the depth of a DNN is independently related to the representation power of vector-to-vector regression functions, which offers a justification that DNN with deeper architectures improves experimental results of DNN-based vector-to-vector mapping. Besides, DNN generalization capability is briefly discussed based on a new uniform convergence derived from Rademacher complexity [26], which can be used to obtain data-dependent upper-bounds on the learnability of function classes.

As for the experimental part, the related experiments of speech enhancement aim to verify our theoretical analysis for the following two points: (1) We evaluate the expressive power of neural networks with different architectures by setting up different widths and depths. (2) We further investigate the expressive power by analyzing the regression accuracy of neural structures with different widths and depths.

The remainder of the paper is organized as follows: Section II offers some foundational concepts and symbols used in the paper. Section III introduces the classical universal approximation theorems. Sections IV and V separately present our theoretical analysis of DNN based vector-to-vector regression. Experiments on speech enhancement are given in Section VII and we conclude our work in Section VIII. Moreover, proofs of Corollaries, which are not central to this work, are offered in Appendix A and B. Finally, a brief discussion of the generalization power of DNN based vector-to-vector regression is presented in Appendix C.

II. Preliminaries

A. Asymptotic Analysis

The asymptotic analysis here refers to measuring the running time of any operation in mathematical units of computation when problem scales become sufficiently large. A typical terminology and related notations include:

- Big-Oh $O(\cdot)$: If $T(r) = O(f(r))$, there exist constants $z, r_0 \geq 0$ such that $T(r) \leq zf(r)$ for all $r \geq r_0$.
- Theta $\Theta(\cdot)$: If $T(r) = \Theta(f(r))$, there exist constants z_1, z_2 such that $z_1 f(r) \leq T(r) \leq z_2 f(r)$ for all $r \geq r_0$.

B. Convex Optimization

A few key concepts concerning convex optimization:

- A normed vector space is a vector space U, with a d-dim vector $\mathbf{u} \in U$ with its i-th element u_i, in which each vector has a norm $\|\mathbf{u}\|$ such that (1) for any $\mathbf{u}$, $\|\mathbf{u}\|$ is a unique scalar, and (2) $\|\mathbf{u}\| = 0$ if and only if $\mathbf{u} = \mathbf{0}$. All normed vector spaces in this paper will use variants of the L_p norm, defined by Equation (2). When p go to infinity, $||\mathbf{u}||_\infty = \max(|u_1|, |u_2|, \ldots, |u_d|)$.

$$||\mathbf{u}||_p = \left(\sum_{i=1}^{d} |u_i|^p \right)^{\frac{1}{p}} \tag{2}$$

- A set K is convex if for any $\mathbf{u}, \mathbf{v} \in K$, all points on the line segment connecting $\mathbf{u}$ and $\mathbf{v}$ also belong to K, i.e.,

$$\forall \alpha \in [0, 1], \alpha\mathbf{u} + (1 - \alpha)\mathbf{v} \in K. \tag{3}$$

- A function $f : K \to \mathbb{R}$ is convex if for any $\mathbf{u}, \mathbf{v} \in K$,

$$f(\mathbf{u}) - f(\mathbf{v}) \leq \nabla f(\mathbf{u})^\top (\mathbf{u} - \mathbf{v}), \tag{4}$$

  where we suppose the first-order gradient $\nabla f(\mathbf{u})$ exists.
- A function f is α-strongly convex if for any $\mathbf{u}, \mathbf{v} \in K$,

$$f(\mathbf{u}) - f(\mathbf{v}) \leq \nabla f(\mathbf{u})^\top (\mathbf{u} - \mathbf{v}) - \frac{\alpha}{2}||\mathbf{u} - \mathbf{y}||_2^2. \tag{5}$$

- A function f is β-smooth if for any $\mathbf{u}, \mathbf{v} \in K$,

$$f(\mathbf{u}) - f(\mathbf{v}) \geq \nabla f(\mathbf{u})^\top (\mathbf{u} - \mathbf{v}) - \frac{\beta}{2}||\mathbf{u} - \mathbf{v}||_2^2. \tag{6}$$

  The key property of a smooth function is that it has derivatives of any order everywhere in its domain. Consequently, the smoothness of a function ensures that many optimization algorithms based on gradients can be efficiently conducted without estimating sub-gradients of non-differentiable points. In addition, a β-smooth function

is equivalent to a β-Lipschitz continuous function over the first-order gradients, i.e.,

$$||\nabla f(\mathbf{u}) - \nabla f(\mathbf{v})||_2 \leq \beta ||\mathbf{u} - \mathbf{v}||_2. \tag{7}$$

III. Classical Universal Approximation Theorems

The problem to be addressed here can be informally stated as follows: *How many functions can a layered neural architecture realize?*

In [27], Kolmogorov's theorem states that a scalar function $f(x)$ of d variables can be represented with a three-layer function having $d(2d+1)$ inner functions in the first, and $(2d+1)$ outer functions in the second hidden layer. The inner functions are universal and do not depend on the particular function $f(x)$; whereas, the outer functions do.

Different mathematical proofs establishing standard multilayer feed-forward networks, with as few as one hidden layer having smooth activations, as universal approximation function appeared in the late 80's. Cybenko [24], Hornik *et al.* [28], and Funahashi [29] independently reached the same result, yet Cybenko's universal approximation theorem is mathematically concise and elegant. Cybenko replaced the inner and outer functions with affine transforms and task-independent bounded scalar nonlinearities, respectively, to approximate any continuous scalar function of n real variables with support in the unit hypercube. Cybenko's main result is formally stated in Theorem 1.

Theorem 1: A continuous function $\hat{f} : \mathbb{R}^d \to \mathbb{R}$ can be universally approximated by a feed-forward ANN f as follows,

$$f(\boldsymbol{x}) = \sum_{j=1}^{n} \alpha_j \sigma(\boldsymbol{w}_j^\top \boldsymbol{x} + b_j) \tag{8}$$

where $\boldsymbol{x}, \boldsymbol{w}_j \in \mathbb{R}^d$, $\alpha_j, b_j \in \mathbb{R}$, and σ refers to any bounded activation function satisfying certain smoothness constraints, e.g., a Sigmoid or modified ReLU activation function.

As shown in [24], the definition of Sigmoid activation function (9) faithfully satisfies the approximation requirement of Theorem 1. In the meanwhile, we introduce a modified ReLU activation function denoted as $MReLU$ in Eq. (10), which is derived from the clip function in [30] and will be used in our present work.

$$Sigmoid(x) = \frac{1}{1 + \exp(-x)} \tag{9}$$

$$MReLU(x) = \min(\max(0, x), 1) \tag{10}$$

Cybenko also demonstrated that the ANN universal approximation capability can be attained with an arbitrarily small error ϵ, which more formally implies that $f(\mathbf{x})$ in Theorem 1 is dense in $\mathbb{R}^d$, as shown in Proposition 1.

Proposition 1: For any $\boldsymbol{x} \in \mathbb{R}^d$ and arbitrary small ϵ, a continuous function $\hat{f}(\boldsymbol{x})$ can be approximated by a feed-forward ANN $f(\boldsymbol{x})$ such that Eq. (11) is always valid.

$$|\hat{f}(\boldsymbol{x}) - f(\boldsymbol{x})| \leq \epsilon \tag{11}$$

Proposition 1 suggests that for any given ϵ, there is some ANN that can achieve that ϵ. However, it does not link the number of hidden units, the width of a single hidden layer, with the approximation error, which was first proven by Barron in [25], where the sigmoid activation functions are employed. Theorem 2 discusses an essential bound using Sigmoid activation function [25], and we further extend it to the modified ReLU function which has not been shown in [24], [25], [30]. The constrained conditions for ReLU in our work lie in the fact the input domain is in $\mathbb{R}^d$ rather than $[0, 1]^d$, otherwise a standard ReLU function may grow to infinity without a bound.

Theorem 2: Given a continuous function $\hat{f} : \mathbb{R}^d \to \mathbb{R}$, we can find a function $f : \mathbb{R}^d \to \mathbb{R}$ in the convex hull of n Sigmoid or modified ReLU activation functions such that $\forall \boldsymbol{x} \in \mathbb{R}^d$,

$$|\hat{f}(\boldsymbol{x}) - f(\boldsymbol{x})| \leq 2C \left(\frac{1}{\sqrt{n}} + \delta_\tau \right). \tag{12}$$

where the constants $\tau > 0$, $C > 0$, δ_τ is a distance between the unit step function and the scaled sigmoid activation function which is defined as Eq. (13), where $1_{z>0}$ equals to 1 if $z > 0$ and otherwise becomes 0. In addition, with $\tau \to \infty$, $\delta_\tau \to 0$.

$$\delta_\tau = \min_{0 \leq \epsilon \leq \frac{1}{2}} \left\{ 2\epsilon + \max_{|z| \geq \epsilon} |\sigma(\tau z) - 1_{z>0}| \right\}. \tag{13}$$

The upper bound in Eq. (12) of Theorem 2 suggests that a large n corresponds to a lower approximation error, and it also implies that a large number of neurons of the hidden layer of the ANN can result in sufficiently powerful expressiveness of arbitrary continuous functions. The classical universal approximation theorems focus on an approximated function for the vector-to-scalar regression. Therefore, we still need some investigation in approximating a vector-to-vector regression function in terms of the Euclidean distance.

IV. ANN Based Vector-to-Vector Regression

We now generalize the classical theorems in Section III to associate the width of the hidden layer with the related expressive power in vector-to-vector mapping. The universal approximation upper bounds for the Sigmoid activation function was shown in [25]. We first need to justify that the modified ReLU also realizes the universal approximation bound.

Proposition 2: Based on an ANN with n modified ReLU activation units $f : \mathbb{R}^d \to \mathbb{R}$, Barron universal approximation bound Eq. (12) becomes Eq. (14), $\forall \boldsymbol{x} \in \mathbb{R}^d$.

$$|\hat{f}(\boldsymbol{x}) - f(\boldsymbol{x})| = O\left(\frac{1}{\sqrt{n}} \right). \tag{14}$$

Proof: We first show that for $|z| \geq \epsilon$ the modified ReLU activation function can ensure the inequality in Eq. (15).

$$\begin{aligned} |\sigma(\tau z) - 1_{z>0}| &= |MReLU(\tau z) - 1_{z>0}| \\ &= (1 - \tau z) 1_{\{0 \leq \tau z \leq 1\}} \\ &\leq \min_{|z| \geq \epsilon} \exp(-\tau \epsilon) \end{aligned} \tag{15}$$

Based on Eqs. (13) and (15), $\epsilon = \frac{\ln \tau}{\tau}$ yields

$$\delta_\tau \leq \frac{1}{\tau} + \frac{2 \ln \tau}{\tau} \tag{16}$$

Next, we use the inequality Eq. (16) and choose the parameter τ as $\sqrt{n}\ln n$, and the upper bound Eq. (12) becomes

$$\begin{aligned} 2C\left(\frac{1}{\sqrt{n}}+\delta_\tau\right) &\le 2C\left(\frac{1}{\sqrt{n}}+\frac{1}{\tau}+\frac{2\ln\tau}{\tau}\right) \\ &\le 2C\left(\frac{2}{\sqrt{n}}+\frac{2\ln(\ln n)}{\sqrt{n}\ln n}+\frac{1}{\sqrt{n}\ln n}\right) \\ &= O\left(\frac{1}{\sqrt{n}}\right), \end{aligned} \tag{17}$$

which justifies the upper bound Eq. (14). ■

We now extend Barron's theorem explicitly to vector-to-vector mapping and deploy an algorithm to demonstrate how an ANN can achieve the resulting bound. The Barron universal approximation theorem is generalized to the scenario of the vector-to-vector regression as shown in Theorem 3.

Theorem 3: Given a continuous vector-to-vector regression function $\hat{f}:\mathbb{R}^d\to\mathbb{R}^q$, we can find an approximated functional f with q functions $f=[f_n^{(1)},\ldots,f_n^{(q)}]$, where each $f_n^{(i)}:\mathbb{R}^d\to\mathbb{R}$ consists of n Sigmoid or modified ReLU activation functions such that $\forall \boldsymbol{x}\in\mathbb{R}^d$,

$$||\hat{f}(\boldsymbol{x})-f(\boldsymbol{x})||_1 = O\left(\frac{q}{\sqrt{n}}\right). \tag{18}$$

Proof: Suppose the regression function $\hat{f}$ is an q-dimensional functional $\hat{f}=[\hat{f}^{(1)},\hat{f}^{(2)},\ldots,\hat{f}^{(q)}]$, where each function $\hat{f}^{(i)}:\mathbb{R}^d\to\mathbb{R}$. Thus, based on Proposition 2 and the L_p-norm definition in Eq. (2), we obtain the bound Eq. (19) for approximating the ANN based regression functions:

$$\begin{aligned} ||\hat{f}(\mathbf{x})-f(\mathbf{x})||_1 &= \sum_{i=1}^{q}|\hat{f}^{(i)}(\mathbf{x})-f_n^{(i)}(\mathbf{x})| \\ &= \sum_{i=1}^{q} O\left(\frac{1}{\sqrt{n}}\right) = O\left(\frac{q}{\sqrt{n}}\right) \end{aligned} \tag{19}$$

■

Theorem 3 suggests that the functional f corresponds to an ANN for the vector-to-vector regression, and the upper bound in Eq. (18) in Theorem 3 implies that the representation power of an ANN is essentially controlled by the width of the hidden and the output layers.

The back-propagation (BP) algorithm based on stochastic gradient descent (SGD) is applied to update parameters of ANNs, we therefore would like to verify whether SGD can achieve the bound in Eq. (19). We first introduce an iterative approximation algorithm proposed by Barron that can realize the approximation bound in Eq. (14) 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.

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. Furthermore, we assume that g represents the gradient of f, α is a learning rate, and the bounded set G is defined as the set of functions representable by an n-node ANN in Eq (20).

Algorithm 1: Iterative Approximation.

1. Input: A bounded set G, and a target $f\in G$.
2. Choose arbitrary $f_0\in G$.
3. For $t=1,2,\ldots,T$:
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 $f_t:=\alpha_t f_{t-1}+(1-\alpha_t)g_t$.

Algorithm 1 becomes a BP algorithm with momentum [31].

$$G=\{f_{\mathbf{w},b}(\mathbf{x}):\mathbf{w}\in\mathbb{R}^{n\times d},b\in\mathbb{R}^n\}. \tag{20}$$

This section also discusses the convergence rate of different sigmoid activation functions for hidden layers, which are separately concluded in Corollary 1 and Corollary 2. Corollary 1 aims at the modified ReLU based hidden layer, and Corollary 2 refers to the Sigmoid layer. The related proofs are separately shown in Appendix A and Appendix B.

Corollary 1: In an input domain $[-\infty,1]^d$, the modified 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 needs $\Theta(\frac{1}{\epsilon^2})$ iterations for an ϵ-optimal solution.

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

By comparing the convergence rates in the two corollaries, SGD for an ANN with a modified ReLU based hidden layer ensures a faster rate because $\Theta(\frac{1}{\epsilon^2})$ is smaller than $\Theta(\frac{\beta}{\epsilon})$ for all $\beta>1/\epsilon$, as is true of the Sigmoid hidden layer for most reasonable values of ϵ. Furthermore, some new optimization algorithms, such as root mean square propagation (RMSProp), adaptive gradient (AdaGrad), and adaptive moment estimation (Adam), are SGD extensions for speeding up the convergence rate. However, they may fail to converge to an optimal solution under some settings. It is not clear if they can achieve Barron's bound. Therefore, we are only concerned with SGD here.

V. DNN Based Vector-to-Vector Regression

This section establishes a connection between the depth of a DNN and the expressive power of vector-to-vector regression functions. We discuss whether the expressive power can benefit from the increment of depth in terms of the number of hidden layers. Theorem 4 suggests that the depth of a DNN is associated with its mapping capabilities. We then consider the constraints of width and depth together to compose an estimated bound for practical use.

Theorem 4: Let $\hat{f}:\mathbb{R}^d\to\mathbb{R}^q$ refer to a vector-to-vector smooth function, we can find a feed-forward DNN f_{DNN} with k modified ReLU based hidden layers ($k\ge 2$), where the width of each hidden layer is at least $d+2$, to approximate the function $\hat{f}$ with an upper bound as Eq. (21), for $\boldsymbol{x}\in\mathbb{R}^d$ and an integer $r\ge 1$ which depends on the maximum value of the first k derivatives of f, we have

$$||\hat{f}(\boldsymbol{x})-f_{DNN}(\boldsymbol{x})||_1 = O((k-1)^{-\frac{r}{d}}) \tag{21}$$

Proof: Before we demonstrating Theorem 4, we introduce Lemmas 1 and 2. Lemma 1 is based on Theorem 1 in [32], and Lemma 2 is from [33].

Lemma 1: For a smooth function $f : \mathbb{R}^d \rightarrow \mathbb{R}^q$, there exists a modified ReLU based ANN f_{ANN} with a hidden layer of k units, and a constant C_f which depends on the maximum value of the first k derivatives of f. Then, we can find an integer $r \geq 1$, there is a constraint (22) in which $D^k f$ denotes a vector of derivatives as $[\nabla f, \nabla^2 f, \ldots, \nabla^k f]$.

$$||f||_\infty + \sum_{k, 1 \leq \frac{k(k-1)}{2} \leq r} ||D^k f||_\infty \leq C_f \tag{22}$$

such that we obtain (23).

$$||f(\mathbf{x}) - f_{ANN}(\mathbf{x})||_1 = O(k^{-\frac{r}{d}}) \tag{23}$$

Proof: Lemma 1 can be obtained from Theorem 3.1 in [32] where the input dimension is configured as d. Besides, we use the modified ReLU function in our work rather than the standard ReLU used in [32]. However, if all input points beyond $[0, 1]^d$ are taken as the bounded points 0 or 1, then the modified ReLU function does not change the original Theorem 3.1 in [32]. ■

Lemma 2: Let $f : R^d \rightarrow R^q$ be a modified ReLU based ANN with input dimension d and a single hidden layer of width k $(k \geq 1)$. There exists another modified ReLU based DNN f_{DNN}, which has input dimension d and $(k + 1)$ hidden layers with width $(d + 2)$, that computes the same function as f.

Proof: Assume that a vector $\mathbf{A}^{(k)} = \{A_1^{(k)}, A_2^{(k)}, \ldots, A_{n_k}^{(k)}\}$ as the output of the k-th hidden layer of width $n_k = d + 2$ based on the modified ReLU function, then we derive Eq. (24) where $\forall j \in [1, 2, \ldots, n_k]$, $\mathbf{w}_j^{(k)} \in \mathbb{R}^{d+2}$, $\mathbf{b}^{(k)} \in \mathbb{R}^{d+2}$, and $\forall l \in [1, 2, \ldots, q]$, $\mathbf{w}_l^{(k+1)} \in \mathbb{R}^q$, $\mathbf{b}^{(k+1)} \in \mathbb{R}^q$.

$$\mathbf{A}^{(k+1)} = MReLU\left(\mathbf{b}^{(k)} + \sum_{j=1}^{n_k} \mathbf{w}_j^{(k)} A_j^{(k)}\right) \tag{24}$$

$$f_{DNN} = \mathbf{b}^{(k+1)} + \sum_{l=1}^{n_{k+1}} \mathbf{w}_l^{(k+1)} A_l^{(k+1)} \tag{25}$$

Then, based on Lemma 6 in [33] and similar Theorems in [34], we know that f can be approximated by $\mathbf{A}^{(k+1)}$, which implies that $\hat{f}$ can be guaranteed to be approximated by f_{DNN} by setting all values in $\mathbf{b}^{(k+1)}$ as 0 and all values in $\mathbf{w}_j^{(k+1)}$ to be $\frac{1}{n_{k+1}}$. ■

Finally, by applying Lemma 2 to Lemma 1, we can find a modified ReLU based DNN f_{DNN} with k hidden layers which can be represented by an ANN f_{ANN} with a hidden layer of $k - 1$ units. Thus, we can obtain Eq. (26) which justifies Theorem 4.

$$\begin{aligned} ||f(\mathbf{x}) - f_{DNN}(\mathbf{x})||_1 &= ||f(\mathbf{x}) - f_{ANN}(\mathbf{x})||_1 \\ &\leq \frac{C_f}{(k-1)^{\frac{r}{d}}} = O\left(\frac{1}{(k-1)^{\frac{r}{d}}}\right) \end{aligned} \tag{26}$$

■

Theorem 4 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 of d and q, a tighter upper bound can be achieved for larger number k. Besides, the width of hidden layers must have at least $(d + 2)$ to obtain the bound in Eq. (21). In other words, a deeper modified ReLU based neural network corresponds to the better expressive power of the vector-to-vector regression function $\hat{f}$.

Although Theorem 4 shows that the upper bound of DNN based vector-to-vector regression depends on the depth k of a DNN structure, the related bound should also be related with the width of hidden layers. Theorem 5 revises the bound in Eq. (21) in Theorem 4 which considers both depth and width.

Theorem 5: For a vector-to-vector regression target function $\hat{f} : \mathbb{R}^d \rightarrow \mathbb{R}^q$, there exists a DNN f_{DNN} with k $(k \geq 2)$ modified ReLU based hidden layers, where the width of each hidden layer is at least $(d + 2)$ and the top hidden layer has n_k $(n_k \geq d + 2)$ units. For an integer $r \geq 1$ associated with the maximum value of derivatives of f, we can derive Eq. (27).

$$||\hat{f} - f_{DNN}||_1 = O\left(\frac{q}{(n_k + k - 1)^{\frac{r}{d}}}\right) \tag{27}$$

Proof: As in the proof of Lemma 2 in [33], the first d hidden nodes, in each hidden layer before the last, are scaled and shifted 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 so far. In that case, the entire network acts like a 2-layer network with $n_k + (k - 1)$ hidden nodes. By Lemma 1, the approximation error is $O(\frac{q}{(n_k+k-1)^{\frac{r}{d}}})$. ■

Note that both Theorems 4 and 5 focus on the modified ReLU based hidden layers. Unfortunately, we have not found if there exists a related bound for the Sigmoid based DNNs.

Finally, we discuss whether the iterative algorithm like SGD can achieve the related bounds. Based on the discussion in Section IV, we understand that for the given depth k and the underlying $(k - 1)$ hidden layers, the approximation bound becomes Eq. (14) which can be achieved via applying the SGD algorithm. However, if the widths of hidden layers are given, some of the recent work, namely [35] and [36], suggest vanilla SGD can converge to local optimal points with provable generalization bounds.

VI. Estimation of MSE Upper Bounds

The mean square error (MSE) [37] is usually taken 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 in Section V and Section IV to estimate MSE upper bounds to the vector-to-vector regression models in our experiments.

Proposition 3 generalizes the theoretical bound in Eq. (19) to a practical bound in Eq. (28), where the number of training data samples N and input dimension d need to be taken into account. Proposition 3 is directly derived from Eq. (18) in Theorem 3 and a vector-to-vector generalization of the bound for MSE in [30].

Proposition 3: For a target function $\hat{f} : \mathbb{R}^d \rightarrow \mathbb{R}^q$, we can use N training data samples to obtain an ANN f with n Sigmoid or modified ReLU activation functions such that the evaluation

loss based on MSE can be bounded as in Eq. (28),

$$MSE(\hat{f}, f) = O\left(\frac{q}{n}\right) + O\left(\frac{qnd}{N}\log N\right) + \nu \tag{28}$$

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

Proposition 3 suggests that for two constants c_1 and c_2, $MSE(\hat{f}, f)$ is upper bounded by Eq. (29).

$$MSE(\hat{f}, f) \leq c_1\frac{q}{n} + c_2\frac{qnd}{N}\log N + \nu \tag{29}$$

For f_1 with l_1 hidden units and f_2 with l_2 hidden units, if Eqs. (30), (31) and (32) are all expected to be satisfied, we derive Eqs. (33) and (34), where $l_2 > l_1$ and $0 < \nu < \frac{\epsilon_1+\epsilon_2}{2}$, where ϵ_1 and ϵ_2 are two lower bounds to $MSE(\hat{f}, f_1)$ and $MSE(\hat{f}, f_2)$, respectively.

$$\epsilon_1 \leq MSE(\hat{f}, f_1) \leq c_1\frac{q}{l_1} + c_2\frac{ql_1d}{N}\log N + \nu \tag{30}$$

$$\epsilon_2 \leq MSE(\hat{f}, f_2) \leq c_1\frac{q}{l_2} + c_2\frac{ql_2d}{N}\log N + \nu \tag{31}$$

$$c_1\frac{q}{l_1} + c_2\frac{ql_1d}{N}\log N + \nu \geq c_1\frac{q}{l_2} + c_2\frac{ql_2d}{N}\log N + \nu \tag{32}$$

$$c_1 \geq \frac{l_1l_2(\epsilon_1+\epsilon_2-2\nu)}{2q(l_1+l_2)} = \hat{c}_1 \tag{33}$$

$$0 < c_2 \leq \frac{Nc_1}{l_1l_2d\log N}, \quad \hat{c}_2 = \frac{N\hat{c}_1}{l_1l_2d\log N} \tag{34}$$

In practical DNN usage, some factors, such as the number of training data and dimensions of some hidden layers, need to be taken into account. Theorem 5 is thus generalized to Proposition 4.

Proposition 4: For a target function $\hat{f} : \mathbb{R}^d \to \mathbb{R}^q$, we use N training data to obtain f_{DNN} with k modified ReLU based hidden layers $(k \geq 2)$, where the width of each hidden layer is at least $(d+2)$. For an integer $r \geq 1$ associated with the maximum value of derivatives of f, the MSE loss is bounded by Eq. (35).

$$MSE(\hat{f}, f_{DNN}) = O\left(\frac{q}{(k-1+n_k)^{\frac{2r}{d}}}\right) + O\left(\frac{qn_kn_{k-1}}{N}\log N\right) + \nu \tag{35}$$

where n_k and n_{k-1} denote the number of units of the k-th and $(k-1)$-th hidden layers, and ν refers to a constant approximation error from the non-deterministic input noise.

Proof: For a DNN with k hidden layers $(k > 2)$, we regard the bottom $(k-2)$ hidden layers as a feature extractor for the $(k-1)$-th hidden layer which can be taken as the input to the top hidden layer. Since the values of top hidden layer based on the modified ReLU smoothly lie in $(0, 1)$ with derivatives of all orders, Barron's bound continues to hold [30]. Therefore, by combining Eq. (27) in Theorem 5 with Eq. (28) in Proposition 3, we derive Eq. (35) in Proposition 4. ■

Proposition 4 suggests that there exist two constants a_1 and a_2, and we obtain the following:

$$MSE(\hat{f}, f_{DNN}) \leq \frac{a_1q}{(n_k+k-1)^{\frac{2r}{d}}} + \frac{a_2qn_kn_{k-1}}{N}\log N + \nu \tag{36}$$

By comparing Eq. (36) with Eq. (29), only the term $\frac{q}{(n_k+k-1)^{\frac{2r}{d}}}$ relies on the depth and larger depth further reduces MSE. Thus, we separately set a_1 and a_2 as c_1 and c_2 because the factor of depth does not impose more additional restrictions.

Further, $\hat{c}_1$ in Eq. (33) and $\hat{c}_2$ in Eq. (34) are associated with a minimum estimated MSE, which correspond to a global optimum point found by SGD. However, a vanilla SGD without the use of some optimization tricks like dropout generally cannot ensure a closely approximated solution to the global one. Thus, we set an MSE upper bound by setting $c_1 = \hat{c}_1$ in (33) and $c_2 = \frac{N\hat{c}_1}{l_1l_2d\log N}$ in (34) for the concern of implicit optimization bias from SGD [38] and obtaining MSE upper bounds as minimum as possible.

As to the setup of ϵ_1 and ϵ_2 for computing c_1 and c_2, Corollary 1 suggests that the modified ReLU based ANNs can ensure the minimum MSE because of the property of convexity, if the input domain lies in $[-\infty, 1]^d$. Thus, we can set ϵ_1 and ϵ_2 as the empirical MSE values of two modified ReLU based ANN models.

Besides, we set the integer r in Eq. (36) as d, which is the same as the input dimensions. Thus, the estimation of MSE upper bound can be modified as (37).

$$MSE(\hat{f}, f_{DNN}) \leq \frac{a_1q}{(n_k+k-1)^2} + \frac{a_2qn_kn_{k-1}}{N}\log N + \nu \tag{37}$$

Finally, the configuration for v varies from various noisy types of different SNR levels. Empirically, we set the values of v as 0.1 under all noisy conditions as shown in Table II.

VII. Experiments on Speech Enhancement

A. 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 [2], [13], [14]; therefore, the present work mainly discusses the expressive power but not the generalization problem and overfitting problems, and that implies that we would not use very large neural architectures and focus on matched noisy conditions. 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. Interesting, this property has also been experimentally verified in [39]–[42], where the authors noticed that bottleneck features extracted from a layer closer to the output lead to a better and abstract representation of original speech features.

The experimental configurations and data preparation are briefly summarized in the next section. More technical detail of the related C++/Python codes can be found in [2], [13].

B. Experimental Setup

The DNN used for speech enhancement is a feed-forward ANN, where inputs were the normalized log-power spectral feature vectors [40] 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 phase. The Sigmoid and modified ReLU functions were used for hidden layers of neural networks, whereas the linear function is used as the output layer for the vector-to-vector regression. To improve the subjective score in a voice conversion task, the global variance equalization [43] was used to alleviate the problem of over-smoothing by correcting the global variance between estimated feature vectors and clean reference targets. During DNN training, the standard back-propagation algorithm [44] 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) [45] 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 [46], 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 [47] were chosen for synthesizing the noisy training and testing samples at SNR levels of 5 dB, and 15 dB. The M109 noise is a stationary noise 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 sections, 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 15 dB, the DNN model would be evaluated with the test data having the same characteristics in terms of noise types and SNR values. Besides, all the clean speech and noise waveforms were downsampled to 8 KHz. The frame length and the shift length were separately set to 32 msec and 16 msec which correspond to 256 samples and 128 samples, respectively. So, the dimension of one feature 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.

Since the outputs of modified ReLU function over inputs of $(1, \infty)$ are exactly 1, Corollary 1 should be still valid if all of inputs whose values greater than 1 are set to 1. It is simple to justify that the outputs of modified ReLU based ANN are the same over the two input domains. Thus, our experimental results of modified ReLU based ANN can reasonably correspond to Corollary 1.

Two evaluation criteria, namely MSE and the perceptual evaluation of speech quality (PESQ) [48], were employed in our experimental validation. The MSE assessment is directly associated with the expressive power of vector-to-vector regression functions because MSE is set as the objective loss function in the DNN training process. A lower MSE value corresponds to better representativeness. On the other hand, PESQ is an indirect evaluation which is highly correlated with subjective objective scores. 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.

C. 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 I demonstrates 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.

As shown in Table I, 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 4 (Here, $d = 771$, $d + 2 = 773 < 800$); 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 I, 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 III shows the experimental results of different neural network architectures. The evaluation of speech enhancement in terms of both MSE and PESQ measures was conducted in a straightforward noisy condition (M109) with a high SNR level (15 dB). 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 by DNN1 with four hidden non-linear layers. Moreover, both DNN1 and DNN2 with a deeper architecture can lead to better regression performance.

TABLE I
MODEL STRUCTURES FOR VARIOUS VECTOR-TO-VECTOR REGRESSION

Models	Structures (Input – hidden_layers – Output)
ANN1 (MReLU)	771-800-257
ANN2 (MReLU)	771-1600-257
ANN1 (Sigmoid)	771-800-257
ANN2 (Sigmoid)	771-1600-257
DNN1 (MReLU)	771-800-800-800-1600-257
DNN2 (MReLU)	771-800-800-800-800-800-1600-257
DNN3 (MReLU)	771-800-800-800-800-800-800-257
DNN4 (MReLU)	771-800-800-800-800-1600-800-257

TABLE II
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

TABLE III
THE EVALUATION RESULTS UNDER THE M109 NOISE OF SNR 15 DB

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

Besides, we estimate the MSE upper bounds based on Eqs. (29) for ANNs and (36) for DNNs. As discussed in Section V, we assume that the modified ReLU based ANNs can closely achieve ϵ_1 and ϵ_2 in Eqs. (33) and (34) by taking ϵ_1 = Estimate_MSE (ReLU) = ANN1 (MReLU) and ϵ_2 = Estimate_MSE (ReLU) = ANN2 (MReLU). Then, we can compute $\hat{c}1$ and $\hat{c}_2$ based on ϵ_1 and ϵ_2 used in Eqs. (33) and (34), where the other hyper-parameters for the estimation of MSE upper bounds can be found in Table II. Based on Table III, the results suggest that our estimated MSE (Estiamte_MSE) can offer rational upper bounds for DNN based models, but they can not ensure rational upper bounds for Sigmoid ANNs because of non-convexity of Sigmoid functions. Overall, the experimental results well correspond to the theoretical analysis in Section VII.

D. A Width Evaluation at the Top Hidden Layer of DNN

We now analyze the effects of the width of the top hidden non-linear layer of a DNN. Although we observe that 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 3 in Section IV and Theorem 5 in Section V. Thus, we compared three DNNs with architectures shown in Table I, 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 IV
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 15 DB

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

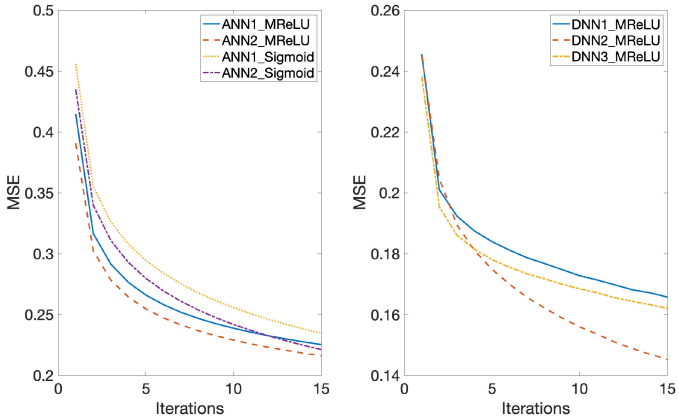


Fig. 1. A comparison of convergence rates on the training set under the M109 noise of SNR 15 dB.

Table IV 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 maintain 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 IV, where we also verify the estimated MSE upper bounds for DNN3 are consistent with the results.

E. A Comparison of Convergence Rates

In Section IV, Corollary 1 and Corollary 2 analyze convergence rates of SGD for the modified ReLU and Sigmoid activation functions, respectively. This section provides some empirical MSE results under the M109 noise of SNR 15 dB to verify the related theories. As shown in Figure 1, the modified ReLU based ANNs perform faster convergence rates and lower MSE values compared with the Sigmoid based ones, which consistently corresponds to Corollary 1 and Corollary 2. Besides, Figure 1 also shows that DNNs with more hidden layers and a

TABLE V
THE EVALUATION RESULTS UNDER THE BABBLE NOISE OF SNR 15 DB

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

TABLE VI
THE EVALUATION RESULTS UNDER THE M109 NOISE OF SNR 5 DB

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

TABLE VII
THE EVALUATION RESULTS UNDER THE BABBLE NOISE OF SNR 5 DB

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

broader top hidden layer can achieve lower MSE values with much faster convergence rates.

F. Empirical Assessment in Adverse Noisy Conditions

Thus 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 V shows the regression results under a complicated Babble noisy condition. Table VI and VII 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 in Section VII-C 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 Babble noise is a typical and one of the most complicated noises in practice.

In adverse conditions, the estimated MSE values based on Eqs. (29) and (36) are separately shown in Table V, VI and VII. By comparison, the estimated MSE upper bounds provide rational estimation to the real empirical MSE in all cases except Sigmoid based ones.

G. Experimental Summary

The empirical regression results discussed in the previous sections confirm our theoretical claims, respectively. More specifically, the experimental results verify that an ANN with a broader hidden width outperforms the one with a narrower one, and a deeper architecture contributes to better expressive power. Also, experimental evidence also suggests a configuration of a broader width at the top hidden layer is essential to achieving better expressive power of DNN based vector-to-vector regression function. Moreover, the related properties of DNN based vector-to-vector regression function can be even maintained in noisy adverse conditions of various SNR levels. Furthermore, the evaluated MSE upper bound can be closely estimated based on our Propositions 3 and 4.

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

VIII. CONCLUSION

This work focused on a theoretical analysis of DNN-based vector-to-vector regression. We have started from the classical universal approximation theorems for an ANN and then generalized the related theorems to DNN. We have shown that the width of the hidden layers of ANN is associated with the approximation of the vector-to-vector regression function. The experiments on speech enhancement verify the related theoretical properties that a broader width at the top hidden layer and a deeper DNN architecture contribute to the better expressive power of DNN based vector-to-vector regression functions. Moreover, the related properties of expressive power still hold even in noisy adverse conditions.

APPENDIX A
PROOF OF COROLLARY 1

First to prove Corollary 1, it is certainly known that the ReLu function is a convex but non-smooth function with the inequality as Eq. (38), where $\mathbf{x}^*$ denotes the optimal point, g_t refers to the sub-gradient of the point $\mathbf{x}_t$ and η is the learning rate, the function f represents a modified ReLU activation function.

$$f(\mathbf{x}_t) - f(\mathbf{x}^*) \leq g_t^\top(\mathbf{x}_t - \mathbf{x}^*) \leq \frac{1}{\eta}(\mathbf{x}_t - \mathbf{x}_{t+1})^\top(\mathbf{x}_t - \mathbf{x}^*)$$
$$\leq \frac{1}{\eta}\left(||\mathbf{x}_t - \mathbf{x}^*||_2^2 + ||\mathbf{x}_t - \mathbf{x}_{t+1}||_2^2 - ||\mathbf{x}_{t+1} - \mathbf{x}^*||_2^2\right)$$
$$\leq \frac{1}{2\eta}\left(||\mathbf{x}_t - \mathbf{x}^*||_2^2 - ||\mathbf{x}_{t+1} - \mathbf{x}^*||_2^2\right) + \frac{\eta}{2}||g_t||_2^2 \quad (38)$$

Summing the resulting inequality over t, and using that $||\mathbf{x}_t - \mathbf{x}^*|| \leq R$ and the sub-gradient of the modified ReLU $||g||_2^2 \leq 1$

yield a regret Eq. (39) at time T.

$$Regret_T = \sum_{t=1}^{\top}(f(\mathbf{x}_t) - f(\mathbf{x}^*)) \leq \frac{R^2}{2\eta} + \frac{\eta T}{2}. \tag{39}$$

By taking $\eta = \frac{R}{\sqrt{T}}$, we obtain Eq. (40).

$$Regret_T \leq R\sqrt{T} \tag{40}$$

On the other hand,

$$f\left(\frac{1}{T}\sum_{t=1}^{\top}\mathbf{x}_t\right) - f(\mathbf{x}^*) \leq \frac{1}{T}\sum_{t=1}^{\top}(f(\mathbf{x}_t) - f(\mathbf{x}^*)) \leq \frac{R}{\sqrt{T}} \tag{41}$$

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

Appendix B
Proof of Corollary 2

To prove Corollary 2. We say a continuously differentiable function f is β-smooth if ∇f is β-Lipschitz, that is

$$||\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})||_2 \leq \beta||\mathbf{x} - \mathbf{y}||_2 \tag{42}$$

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

$$\begin{aligned}
&|f(\mathbf{x}) - f(\mathbf{y}) - \nabla f(\mathbf{y})^\top(\mathbf{x} - \mathbf{y})| \\
&= \left|\int_0^1 \nabla f(\mathbf{y} + t(\mathbf{x} - \mathbf{y}))^\top(\mathbf{x} - \mathbf{y})dt - \nabla f(\mathbf{y})^\top(\mathbf{x} - \mathbf{y})\right| \\
&\leq \int_0^1 ||\nabla f(\mathbf{y} + t(\mathbf{x} - \mathbf{y})) - \nabla f(\mathbf{y})||_2 \cdot ||\mathbf{x} - \mathbf{y}||_2 dt \\
&\leq \int_0^1 \beta t||\mathbf{x} - \mathbf{y}||_2^2 dt = \frac{\beta}{2}||\mathbf{x} - \mathbf{y}||_2^2.
\end{aligned} \tag{43}$$

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

$$\begin{aligned}
f(\mathbf{x}_{t+1}) &\leq f(\mathbf{x}_t) + \nabla f(\mathbf{x}_t)^\top(\mathbf{x}_{t+1} - \mathbf{x}_t) + \frac{\beta}{2}||\mathbf{x}_t - \mathbf{x}_{t+1}||_2^2 \\
&\leq f(\mathbf{x}_t) - \eta||\nabla f(\mathbf{x}_t)||_2^2 + \frac{\beta \cdot \eta^2}{2}||\nabla f(\mathbf{x}_t)||_2^2.
\end{aligned} \tag{44}$$

Summing the resulting inequality over t, we obtain Eq. (45), where we set the learning rate $\eta = \frac{1}{\beta}$.

$$\begin{aligned}
E[||\nabla f(\mathbf{x})||_2^2] = \frac{1}{T}\sum_{t=1}^{\top}||\nabla f(\mathbf{x}_t)||_2^2 &= \frac{2(f(\mathbf{x}_1) - f(\mathbf{x}_{T+1}))}{\eta(2 - \eta\beta)T} \\
&\leq \frac{2\beta(f(\mathbf{x}_1) - f(\mathbf{x}_{T+1}))}{T}
\end{aligned} \tag{45}$$

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

$$\frac{2\beta(f(\mathbf{x}_1) - f(\mathbf{x}_{T+1}))}{T} = \epsilon \tag{46}$$

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

Appendix C
The Generalization Bound for DNN based Vector-to-Vector Regression

Finally, we briefly discuss the uniform convergence bound derived from the Rademacher complexity of neural networks for the generalization power of the vector-to-vector regression, which builds on the recent work in [49]. The Rademacher complexity, which definition is introduced via Definitions 1 and 2, is a measure of how rich a class of hypothesis is. It does so by measuring how well the class can fit random noise using Rademacher random variables.

Definition 1: A Rademacher random variable takes on values ± 1 and is defined by the uniform distribution as Eq. (47).

$$\sigma_i = \begin{cases} 1, & w.p.\ \frac{1}{2} \\ -1, & w.p.\ \frac{1}{2} \end{cases} \tag{47}$$

Definition 2: The empirical Rademacher complexity of a class G of functions $g : X \to \mathbb{R}$ with respect to a sample $S = (x_1, x_2, \ldots, x_q)$ is

$$\hat{R}_S := E_{\epsilon_1,\ldots,\epsilon_q}\left[\sup_{g \in G}\frac{1}{q}\sum_{i=1}^{q}\epsilon_i g(x_i)\right] \tag{48}$$

where $\epsilon_1, \epsilon_2, \ldots, \epsilon_q$ are iid Rademacher random variables.

Note that $\sup(\cdot)$ in Definition 2 is an abbreviation of supremum. The supremum of a sample S of a partially ordered set T is the largest element in T that is less than or equal to all elements of S, if such elements exist.

Next, we employ the uniform convergence bound associated with the Rademacher complexity to measure the DNN generalization power for vector-to-vector regression. Specifically, we first define an ANN class $\mathbb{H}$ as Eq. (49), where $\mathbf{W} = \{\mathbf{w}^{(1)}, \mathbf{w}^{(2)}\}$ in which $\mathbf{w}^{(1)} \in \mathbb{R}^{n \times d}$, $\mathbf{w}^{(2)} \in \mathbb{R}^{q \times n}$. In addition, $B_2' > 0, B_2 > 0$ are two constants, and the empirical and expected losses are defined as $\hat{L}(f)$ and $L(f)$, respectively, the Rademacher complexity of neural networks $R_N(\mathbb{H})$ offers an upper bound Eq. (50) of $||\hat{L}(f) - L(f)||_2$.

$$\mathbb{H} = \{f_{\mathbf{W}} : ||\mathbf{w}_i^{(1)}||_2 \leq B_2', ||\mathbf{w}_j^{(2)}||_2 \leq B_2, i \in [n], j \in [m]\} \tag{49}$$

$$||\hat{L}(f) - L(f)||_2 \leq 2R_N(\mathbb{H}). \tag{50}$$

Theorem 6 presents an upper bound Eq. (51) for the uniform convergence based on the Rademacher complexity. The upper bound does not rely on the width of the hidden layer of ANN, but the number of training data N is necessarily large enough to lower the bound.

Theorem 6: Define $B(\mathbf{W}) = \sum_{i=1}^{n}\sum_{j=1}^{q}||\mathbf{w}_i^{(1)}||_2||\mathbf{w}_j^{(2)}||_2$ and $H = \{f_{\mathbf{W}} : B(\mathbf{W}) \leq B_1\}$. If $||\mathbf{x}_i||_2 \leq C, \forall i \in [n]$, then

$$R_N(\mathbb{H}) \leq \frac{2B_1 C}{\sqrt{N}}. \tag{51}$$

where constants $B_1 > 0$ and $C > 0$.

Proof: The empirical Rademacher complexity can be obtained by Eq. (52), where we use the fact that L_2-norm is self-dual, and let σ refer to the Rademacher variables which

take $\{-1, 1\}$ with an equal probability.

$$\begin{aligned} R_S(\mathbb{H}) &= E_\sigma \left[\sup_{||\mathbf{w}||_2 \le B_1} \frac{1}{N} \sum_i \sigma_i \mathbf{w}^\top \mathbf{x}_i \right] \\ &= E_\sigma \sup_{||\mathbf{w}||_2 \le B_1} \mathbf{w}^\top \left(\frac{1}{N} \sum_i \sigma_i \mathbf{x}_i \right) \\ &= \frac{B_1}{N} E_\sigma || \sum_i \sigma_i \mathbf{x}_i ||_2 \\ &= \frac{B_1}{N} \left(E_\sigma \left(|| \sum_i \sigma_i \mathbf{x}_i ||_2^2 \right) \right)^{\frac{1}{2}} \end{aligned} \tag{52}$$

Then, based on Jensen's inequality [50], we obtain:

$$R_S(\mathbb{H}) \le \frac{B_1}{N} E_\sigma \sqrt{\sum_i \sigma_i^2 ||\mathbf{x}_i||_2^2} = \frac{B_1}{N} \sqrt{\sum_i ||\mathbf{x}_i||_2^2}. \tag{53}$$

Thus, the Rademacher complexity can be derived as Eq. (54).

$$R_N(\mathbb{H}) = E[R_S(\mathbb{H})] \le \frac{B_1}{N} \sqrt{\sum_i E||\mathbf{x}_i||_2^2} \le \frac{B_1 C}{\sqrt{N}}. \tag{54}$$

■

As discussed in Section V, the hidden layers of DNN are taken as better feature extraction, and an ANN is responsible for the vector-to-vector regression. The bottom hidden layers of DNN offer a better feature representation of inputs, and the abstracted feature can be taken as the inputs of an ANN which corresponds to the top two layers of DNN. Thus, the generalization power of ANN can be simply generalized to DNN with a deep learning architecture.

REFERENCES

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

[2] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee, "A regression approach to speech enhancement based on deep neural networks," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 23, no. 1, pp. 7–19, Jan. 2015.

[3] J. Qi, D. Wang, Y. Jiang, and R. Liu, "Auditory features based on gammatone filters for robust speech recognition," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2013, pp. 305–308.

[4] J. Xie, L. Xu, and E. Chen, "Image denoising and inpainting with deep neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 341–349.

[5] A. Bérard, L. Besacier, A. C. Kocabiyikoglu, and O. Pietquin, "End-to-end automatic speech translation of audiobooks," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2018, pp. 6224–6228.

[6] K. Pearson, "VII. mathematical contributions to the theory of evolution. III. regression, heredity, and panmixia," *Philosoph. Trans. Roy. Soc. London. Series A, Containing Papers Math. Phys. Character*, no. 187, pp. 253–318, 1896.

[7] H. Drucker, C. J. Burges, L. Kaufman, A. J. Smola, and V. Vapnik, "Support vector regression machines," in *Proc. Adv. Neural Inf. Process. Syst.*, 1997, pp. 155–161.

[8] S. R. Gunn *et al.*, "Support vector machines for classification and regression," *ISIS Tech. Report*, vol. 14, no. 1, pp. 5–16, 1998.

[9] P. Zhao and B. Yu, "On model selection consistency of lasso," *J. Mach. Learn. Res.*, vol. 7, no. Nov, pp. 2541–2563, 2006.

[10] N. Simon, J. Friedman, T. Hastie, and R. Tibshirani, "A sparse-group lasso," *J. Comput. Graph. Statist.*, vol. 22, no. 2, pp. 231–245, 2013.

[11] B. Scholkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA, USA: MIT Press, 2001.

[12] S. S. Haykin *et al.*, *Neural Networks and Learning Machines*, vol. 3. London, U.K.: Pearson, 2009.

[13] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee, "An experimental study on speech enhancement based on deep neural networks," *IEEE Signal Process. Lett.*, vol. 21, no. 1, pp. 65–68, Jan. 2014.

[14] Y. Xu, J. Du, Z. Huang, L.-R. Dai, and C.-H. Lee, "Multi-objective learning and mask-based post-processing for deep neural network based speech enhancement," in *16th Annu. Conf. Int. Speech Commun. Assoc.*, pp. 1508–1512, 2015.

[15] B. Wu *et al.* "An end-to-end deep learning approach to simultaneous speech dereverberation and acoustic modeling for robust speech recognition," *IEEE J. Sel. Topics Signal Process.*, vol. 11, no. 8, pp. 1289–1300, Dec. 2017.

[16] D. Wang and J. Chen, "Supervised speech separation based on deep learning: An overview," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 26, no. 10, pp. 1702–1726, Oct. 2018.

[17] X.-L. Zhang and D. Wang, "Boosting contextual information for deep neural network based voice activity detection," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 24, no. 2, pp. 252–264, Feb. 2016.

[18] X.-L. Zhang and D. Wang, "A deep ensemble learning method for monaural speech separation," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 24, no. 5, pp. 967–977, May 2016.

[19] G. Hinton *et al.*, "Deep neural networks for acoustic modeling in speech recognition," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 82–92, Nov. 2012.

[20] Z. Lu, H. Pu, F. Wang, Z. Hu, and L. Wang, "The expressive power of neural networks: A view from the width," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 6231–6239.

[21] R. Eldan and O. Shamir, "The power of depth for feedforward neural networks," in *Proc. Conf. Learn. Theory*, 2016, pp. 907–940.

[22] S. Liang and R. Srikant, "Why deep neural networks for function approximation?" in *Proc. Int. Conf. Learn. Represent.*, 2016.

[23] A. Kolmogoro, "On the representation of continuous functions of several variables as superpositions of functions of smaller number of variables," *Soviet Math. Doklady*, vol. 108, 1956, pp. 179–182.

[24] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Math. Control, Signals Syst.*, vol. 2, no. 4, pp. 303–314, 1989.

[25] A. R. Barron, "Universal approximation bounds for superpositions of a sigmoidal function," *IEEE Trans. Inf. Theory*, vol. 39, no. 3, pp. 930–945, May 1993.

[26] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of Machine Learning*. Cambridge, MA, USA: MIT Press, 2018.

[27] C. M. Bishop, *et al.*, *Neural Networks for Pattern Recognition*. London, U.K.: Oxford Univ. Press, 1995.

[28] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Netw.*, vol. 2, no. 5, pp. 359–366, 1989.

[29] K.-I. Funahashi, "On the approximate realization of continuous mappings by neural networks," *Neural Netw.*, vol. 2, no. 3, pp. 183–192, 1989.

[30] A. R. Barron, "Approximation and estimation bounds for artificial neural networks," *Mach. Learn.*, vol. 14, no. 1, pp. 115–133, 1994.

[31] D. C. Plaut and G. E. Hinton, "Learning sets of filters using back-propagation," *Comput. Speech Lang.*, vol. 2, no. 1, pp. 35–61, 1987.

[32] H. N. Mhaskar and T. Poggio, "Deep vs. shallow networks: An approximation theory perspective," *Anal. Appl.*, vol. 14, no. 6, pp. 829–848, 2016.

[33] B. Hanin, "Universal function approximation by deep neural nets with bounded width and relu activations," 2017, *arXiv:1708.02691*.

[34] R. Arora, A. Basu, P. Mianjy, and A. Mukherjee, "Understanding deep neural networks with rectified linear units," in *Proc. Int. Conf. Learn. Represent.*, 2018.

[35] Z. Allen-Zhu and Y. Li, "Can sgd learn recurrent neural networks with provable generalization?" 2019, *arXiv:1902.01028*.

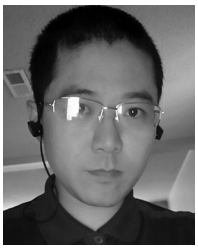
[36] Z. Allen-Zhu, Y. Li, and Y. Liang, "Learning and generalization in overparameterized neural networks, going beyond two layers," 2018, *arXiv:1811.04918*.

[37] Y. Ephraim and D. Malah, "Speech enhancement using a minimum mean-square error log-spectral amplitude estimator," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-33, no. 2, pp. 443–445, Apr. 1985.

[38] S. Gunasekar, J. Lee, D. Soudry, and N. Srebro, "Characterizing implicit bias in terms of optimization geometry," in *Proc. 35th Int. Conf. Mach. Learn.*, 2018, vol. 80, pp. 1832–1841.

[39] M. McLaren, L. Ferrer, and A. Lawson, "Exploring the role of phonetic bottleneck features for speaker and language recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2016, pp. 5575–5579.

[40] X. Hou and L. Zhang, "Saliency detection: A spectral residual approach," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2007, pp. 1–8.
[41] 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.
[42] J. Qi, D. Wang, and J. Tejedor Noguerales, "Subspace models for bottleneck features," in *Proc. Interspeech*, 2013, pp. 1746–1750.
[43] T. Toda, A. W. Black, and K. Tokuda, "Spectral conversion based on maximum likelihood estimation considering global variance of converted parameter," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2005, vol. 1, pp. I–9.
[44] Y. Hirose, K. Yamashita, and S. Hijiya, "Back-propagation algorithm which varies the number of hidden units," *Neural Netw.*, vol. 4, no. 1, pp. 61–66, 1991.
[45] M. L. Seltzer, D. Yu, and Y. Wang, "An investigation of deep neural networks for noise robust speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2013, pp. 7398–7402.
[46] V. Zue, S. Seneff, and J. Glass, "Speech database development at mit: Timit and beyond," *Speech Commun.*, vol. 9, no. 4, pp. 351–356, 1990.
[47] 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 Commun.*, vol. 12, no. 3, pp. 247–251, 1993.
[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. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2001, vol. 2, pp. 749–752.
[49] C. Wei, J. D. Lee, Q. Liu, and T. Ma, "On the margin theory of feedforward neural networks," 2018, *arXiv:1810.005369*.
[50] J. L. W. V. Jensen *et al.*, "Sur les fonctions convexes et les inégalités entre les valeurs moyennes," *Acta Mathematica*, vol. 30, pp. 175–193, 1906.



Jun Qi received the Master of Engineering degree from Tsinghua University, Beijing, China, in 2013 and the graduate study in electrical engineering with MSEE from the University of Washington, Seattle, WA, USA, in 2017. He is currently working toward the Ph.D. degree with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, USA. Also, he was a research intern in Deep Learning Technology Center with Microsoft Research. His study focuses on deep Gaussian process and non-Convex optimization particularly based on submodular functions, and tensor decomposition applied to high-dimensional signal processing in big data, end-to-end speech processing, and natural language processing.

Jun Du received the B.Eng. and Ph.D. degrees from the Department of Electronic Engineering and Information Science, University of Science and Technology of China (USTC), Hefei, China, in 2004 and 2009, respectively. From 2004 to 2009, he was with iFlytek Speech Lab of USTC. During the above period, he worked as an Intern twice for nine months with Microsoft Research Asia (MSRA), Beijing. In 2007, he also worked as a Research Assistant for six months with the Department of Computer Science, The University of Hong Kong. From July 2009 to June 2010, he worked with iFlytek Research on speech recognition. From July 2010 to January 2013, he joined MSRA as an Associate Researcher, working on handwriting recognition, OCR, and speech recognition. Since February 2013, he has been with the National Engineering Laboratory for Speech and Language Information Processin of USTC.

Sabato Marco Siniscalchi received the Laurea and Doctorate degrees in computer engineering from the University of Palermo, Palermo, Italy, in 2001 and 2006, respectively. He is a Professor with the University of Enna "Kore," Enna, Italy and affiliated with the Georgia Institute of Technology. In 2001, he was employed by STMicroelectronics where he designed optimization algorithms for digital image processing. In 2002, he was an Adjunct Professor with the University of Palermo. In 2006, he was engaged as a Post Doctoral Fellow with the Georgia Institute of Technology, Atlanta, under the guidance of Prof. C.-H. Lee. From 2007 to 2009, he joined the Norwegian University of Science and Technology, Trondheim, Norway, as a Research Scientist under the guidance of Prof. T. Svendsen. From 2010 to 2015, he was an Assistant Professor, first, and an Associate Professor, after, at the University of Enna "Kore." In October 2017, he was on one-year leave from his academic appointment and joined as Senior Speech Researcher the Siri Speech Group, Apple Inc., Cupertino CA, USA. He currently acts as an Associate Editor in the IEEE/ACM TRANSACTIONS ON AUDIO, SPEECH AND LANGUAGE PROCESSING, and he was the recipient of a certificate of merit for outstanding editorial board in April, 2018. He is an elected member of the IEEE Speech and Language Technical Committee (2019–2021).

Chin-Hui Lee is a Professor with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, USA. Before joining academia in 2001, he had 20 years of industrial experience ending in Bell Laboratories, Murray Hill, NJ, USA as a Distinguished Member of Technical Staff and the Director of the Dialogue Systems Research Department. He has authored more than 400 papers and 30 patents, and was highly cited for his original contributions with an h-index of 66. He is a Fellow of ISCA. He was the recipient numerous awards, including the Bell Labs President Gold Award in 1998. He won the SPS 2006 Technical Achievement Award for Exceptional Contributions to the Field of Automatic Speech Recognition. In 2012, he gave an ICASSP plenary talk on the future of speech recognition. In the same year, he was awarded the ISCA Medal in scientific achievement for pioneering and seminal contributions to the principles and practice of automatic speech and speaker recognition.