Performance Analysis for Tensor-Train Decomposition to Deep Neural Network Based Vector-to-Vector Regression

Jun Qi, Xiaoli Ma, Chin-Hui Lee
Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, GA
{jqi41,xiaoli}@gatech.edu, chl@ece.gatech.edu

Jun Du
Electrical Engineering
University of Science and Technology
Heifei, China
jundu@ustc.edu.cn

Sabato Marco Siniscalchi
Computer Engineering School
University of Enna
Enna, Italy
marco.siniscalchi@unikore.it

2020 54th Annual Conference on Information Sciences and Systems (CISS) 978-1-7281-4085-8/20/$31.00 ©2020 IEEE 10.1109/CISS48834.2020.1570617364

***Abstract*—This work focuses on a performance analysis of tensor-train decomposition applied to the deep neural network (DNN) based vector-to-vector regression. Tensor-train Network (TTN), obtained through tensor-train decomposition, converts a DNN based vector-to-vector regression into a tensor-to-vector mapping with fewer parameters. We can therefore build an over-parametrized DNN with the tensor-train representation such that the optimization error can be significantly reduced, while the upper bounds on the approximation and estimation errors can be maintained. We compare TTN-based neural architecture against an over-parametrized DNN on the MNIST dataset, and the experimental evidence demonstrates the validity of our conjectures on our proposed performance bounds.**

***Index Terms*—Tensor-train decomposition, deep neural network, vector-to-vector regression, over-parameterization, tensor-to-vector regression**

I. Introduction

Deep neural networks have thrived in various machine learning and signal processing applications. One important application is vector-to-vector regression [1]–[4], which aims at building a functional mapping from input vectors to output vectors. This process is mathematically formulated as

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

where $\mathbf{x} \in \mathbb{R}^d$, $\mathbf{y} \in \mathbb{R}^q$, and a functional mapping $f : \mathbb{R}^d \to \mathbb{R}^q$ is built up to transform the input vectors to the output space such that target vectors can be closely approximated by the output vectors.

Despite the remarkable success in a wide range of real-world vector-to-vector regression tasks, most of the DNN related work focuses on experimental studies, but the development of the related performance analysis is far behind the pace. Our work [5] attempts to offer upper bounds on the representation power of DNN based vector-to-vector regression, and our recent work [6] generalizes the bounds through separately upper bounding the approximation, estimation and optimization errors which arise from the decomposition of mean absolute error (MAE) [7]. Although the approximation and estimation error is well-bounded by our theorems, the optimization error, especially that based on the condition of the over-parametrization, needs further discussion. Over-parametrization is assumed as an asset to the DNN training with global optima being guaranteed [8]–[11], but the condition requires the number of model parameters surpasses the amount of training data, which is not normally adapted to large datasets. In [6], we used the γ-Polyak-Lojasiewicz (γ-PL) condition for non-convex functions [12], [13] to upper bound the optimization error because big datasets, with the size of approximately 10^{10}, were used. Unfortunately, sizes make over-parametrized DNN for vector-to-vector regression difficult to configure. However, our experimental results in [6] suggest that the optimization error dominates the overall MAE score, and it cannot be easily decreased with an increase of training data because its related upper bound solely depends on the selection of optimizers and hyper-parameters like the learning rate.

Hence, this work focuses on the study of lowering the optimization error by employing the tensor-train decomposition [14], [15] to hidden layers of DNN, which forms a Tensor-Train Network (TTN) with a compact tensor-train representation of the fully-connected hidden layers of DNN with much fewer parameters. In other words, a DNN based vector-to-vector regression can be flexibly converted to a TTN based tensor-to-vector regression such that it is possible to relax the parameter required for the over-parametrization condition by applying TTN.

Furthermore, we justify that our bounds on MAE in [6] are robust to the tensor-train representation of DNN for the vector-to-vector regression. Therefore, we can transform an over-parametrized DNN into the corresponding TTN with much fewer parameters than the DNN, and the characteristics of the over-parameterization can be maintained for the TTN.

Experiments based on the noisy MNIST datasets [16] are used to justify our new theorems that TTN with much fewer parameters is capable of maintaining the baseline results of the over-parametrized DNN based vector-to-vector mapping. In this work, two metrics are utilized for evaluating the results: (i) a direct evaluation based on the MAE loss; (ii) an indirect measurement based on the signal-to-noise ratio (SNR).

The remainder of this paper is organized as follows: Section II introduces adequate notations and concepts used in this work. Our previous theorems of upper bounding the DNN based vector-to-vector regression is shown in Section III. Section IV is devoted to the performance analysis of the over-

978-1-7281-4085-8/20/$31.00 ©2020 IEEE

Authorized licensed use limited to: University of Science & Technology of China. Downloaded on August 17,2021 at 01:25:53 UTC from IEEE Xplore. Restrictions apply.

parametrized DNN with the tensor-train representation for the vector-to-vector regression. Our experiments of digit image denoising are presented in Section VI, and the paper is finally concluded in Section VII.

II. Preliminaries

A. Notations

- $f \circ g$: The composition of functions f and g.
- $||\mathbf{v}||_p$: L_p norm of the vector $\mathbf{v}$.
- $\langle \mathbf{x}, \mathbf{y} \rangle$: Inner product of two vectors $\mathbf{x}$ and $\mathbf{y}$.
- $[q]$: An integer set $\{1, 2, 3, ..., q\}$.
- ∇f: A first-order gradient of function f.
- $\mathbb{R}^{I_1 \times I_2 \times \cdots \times I_M}$: set of M-mode real-valued tensors.
- $\mathbb{G}$: Hypothesis space for the MAE loss function.
- $\mathbb{H}$: Space of DNN based vector-to-vector functions.
- $\mathbb{T}$: Space of TTN based tensor-to-vector functions.

B. Over-parametrization condition

Over-parametrization means that the number of model parameters m should be larger than the size of training data N. Specifically, m should be a polynomial function of N as follows:

$$m = poly(N). \tag{2}$$

With the setup of over-parametrization for neural networks, a naive stochastic gradient descent (SGD) can find the global optima when the SGD algorithm gets converged [17].

C. γ-PL condition for non-convex functions

A function $f : \mathbb{R}^d \to \mathbb{R}$ satisfies the γ-Polyak-Lojasiewicz (γ-PL) condition, if there is an inequality

$$||\nabla f(\mathbf{x})||_2^2 \geq \gamma(f(\mathbf{x}) - f^*). \tag{3}$$

where f^* denotes the optimal value over the input domain.

The γ-PL condition can be taken as the generalization of convex functions to non-convex ones because of a local optimum of the non-convex function with the γ-PL condition corresponds to the global one. Unlike the over-parametrization condition, the γ-PL condition is independent of the size of training data. Furthermore, a significant aspect is that a deep neural architecture satisfies the γ-PL condition if its weights are randomly initialized to be near the global points [8].

D. Tensor-Train Decomposition

The tensor-train decomposition can be described as follows: Given a set of ranks $\mathbf{r} = \{r_1, r_2, ..., r_{K+1}\}$, a tensor $W \in \mathbb{R}^{(m_1 n_1) \times \cdots \times (m_K n_K)}$ is decomposed into a multiplication of core tensors according to Eq. (1). More specifically, for the given ranks r_k and r_{k+1}, the k-th core tensor $C^{[k]}(r_k, i_k, j_k, r_{k+1}) \in \mathbb{R}^{m_k \times n_k}$, where $i_k \in [m_k], j_k \in [n_k]$. Besides, r_1 and r_{K+1} are fixed to 1.

$$W((i_1, j_1), ..., (i_K, j_K)) = \prod_{k=1}^{K} C^{[k]}(r_k, i_k, j_k, r_{k+1}). \tag{4}$$

TTN is generated by utilizing the tensor-train factorization to offer a compact representation of a feed-forward DNN with fully-connected hidden layers. Since TTN only stores the low-rank core tensors $\{C_k\}_{k=1}^K$ of the size $\sum_{k=1}^K m_k n_k r_k r_{k+1}$ rather than a much larger storage of $\prod_{k=1}^K m_k n_k$ for DNN parameters. Moreover, instead of fine-tuning a TTN by decomposing a well-trained DNN, core tensors of TTN can be randomly initialized and trained by the SGD algorithms.

III. Our theorems for DNN based Vector-to-Vector Regression

This section briefly presents our previous theorems on the DNN based vector-to-vector regression [6]. Theorem 1 shows the MAE loss can be upper bounded by the sum of approximation error, estimation error, and optimization error.

Theorem 1. *Let $\hat{\mathcal{L}} \in \mathbb{G}$ be the loss function for N training samples drawn i.i.d. according to D, and denote a vector-to-vector function $\hat{f} : \mathbb{R}^d \to \mathbb{R}^q$ as the empirical risk minimization (ERM) [18] for $\hat{\mathcal{L}}$. For a generalized MAE loss function $\mathcal{L} \in \mathbb{G}$, there is f_ϵ such that $\mathcal{L}(f_\epsilon) \leq \inf_{f \in \mathbb{H}} \mathcal{L}(f) + \epsilon$, where $\epsilon > 0$ and $0 < \delta < 1$. We obtain that*

$$\begin{aligned}\mathcal{L}(\hat{f}) &\leq \underbrace{\inf_{f \in \mathbb{H}} \mathcal{L}(f)}_{\text{Approx. error}} + \underbrace{2 \sup_{f \in \mathbb{H}} |\mathcal{L}(f) - \hat{\mathcal{L}}(f)|}_{\text{Estimation error}} + \underbrace{\mathcal{L}(f_\epsilon) - \inf_{f \in \mathbb{H}} \mathcal{L}(f)}_{\text{Optimization error}} \\ &\leq \inf_{f \in \mathbb{H}} \mathcal{L}(f) + 2\hat{\mathcal{R}}_S(\mathbb{H}) + \epsilon,\end{aligned} \tag{5}$$

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

Moreover, we separately derive the upper bounds for the approximation error, estimation error, and optimization error as follows:

(a). **An upper bound on the approximation error**: for a smooth tensor-to-vector regression target function $f^* : \mathbb{R}^d \to \mathbb{R}^q$, there is a DNN $\bar{f}$ with $k (k \geq 2)$ modified smooth ReLU based hidden layers, where the width of each hidden layer is greater than $d+2$ and the top hidden layer has $n_k (n_k \geq d+2)$ units, then we can derive

$$\inf_{f \in \mathbb{H}} \mathcal{L}(f) = ||f^* - \bar{f}||_1 = \mathcal{O}\left(\frac{q}{(n_k + k - 1)^{\frac{r}{d}}}\right) \tag{6}$$

where r is the differential order of f.

(b). **An upper bound on the estimation error**: for a DNN based vector-to-vector mapping function $f(\mathbf{x}) = W_k \circ g \circ W_{k-1} \circ \cdots \circ W_2 \circ g \circ W_1(\mathbf{x}) : \mathbb{R}^d \to \mathbb{R}^q$ with a smooth ReLU function g as

$$g(x) = \lim_{t \to +\infty} \frac{\ln(1 + \exp(tx))}{t}, \tag{7}$$

and regularized weight matrix of $W_i, \forall i \in [k]$ for each hidden layer i. Then, we derive

$$\begin{aligned} &2 \sup_{f \in \mathbb{H}} |\mathcal{L}(f) - \hat{\mathcal{L}}(f)| \leq 2\hat{\mathcal{R}}_S(\mathbb{H}) \leq \frac{2q\Lambda^{'}\Lambda^{k-1}v}{\sqrt{N}} \\ &s.t., \quad ||W_k(i,:)||_1 \leq \Lambda^{'}, \forall i \in [q] \\ &\qquad ||W_j(a,:)||_2 \leq \Lambda, \forall j \in [k-1], a \in [n_j] \\ &\qquad ||\mathbf{x}||_2 \leq v, \end{aligned} \tag{8}$$

Authorized licensed use limited to: University of Science & Technology of China. Downloaded on August 17,2021 at 01:25:53 UTC from IEEE Xplore. Restrictions apply.

where $\hat{R}_S(\mathbb{H})$ is the empirical Rademacher complexity, $W_j(n,:)$ contains all weights from the n-th neuron in the j-th hidden layer to all units in the $(j-1)$-th hidden layer, and the input vector $\mathbf{x}$ is bounded by a constant v.

(c). **An upper bound on the optimization error**: with the assumption of γ-PL condition, we derive

$$\epsilon = \mathcal{L}(f_\epsilon) - \inf_{f\in\mathbb{H}} \mathcal{L}(f) \leq \frac{\mu M^2 \beta}{2\gamma}. \tag{9}$$

where μ denotes the learning rate used for the SGD algorithm, β is the constant associated with the smooth property of the loss function $\mathcal{L}$, and $||\nabla\hat{\mathcal{L}}(f)||_2 \leq M$ is assumed.

To aggregate the upper bounds (6), (8) and (9) for the approximation error, estimation error and optimization error, we derive the upper bound on $\mathcal{L}(\hat{f})$ as

$$\begin{aligned}
\mathcal{L}(\hat{f}) &\leq \inf_{f\in\mathbb{H}} \mathcal{L}(f) + 2\hat{\mathcal{R}}_S(\mathbb{H}) + \epsilon \\
&\leq \mathcal{O}\left(\frac{q}{(n_k+k-1)^{\frac{r}{d}}}\right) + \frac{2q\Lambda^{'}\Lambda^{k-1}v}{\sqrt{N}} + \frac{\mu M^2\beta}{2\gamma} \\
\text{s.t.,}\ & \text{Smooth ReLU: } \lim_{t\to+\infty} \frac{1}{t}\ln(1+\exp(tx)) \\
& \text{Hidden Layers: } n_j \geq d+2, \forall j \in [k] \\
& \text{Regularization: } ||W_k(i,:)||_1 \leq \Lambda^{'}, \forall i \in [q] \\
& \qquad\qquad ||W_j(m,:)||_2 \leq \Lambda, \forall j \in [k-1], m \in [n_j] \\
& \qquad\qquad ||\mathbf{x}||_2 \leq v \\
& \text{Parametrization: The number of parameters is large.}
\end{aligned} \tag{10}$$

Next, we discuss the use of TTN for setting up the vector-to-vector regression and justify that TTN is capable of maintaining the upper bounds on the approximation error and estimation error. Particularly, we can flexibly utilize the over-parametrization condition for the TTN based tensor-to-vector regression.

IV. TTN BASED VECTOR-TO-VECTOR REGRESSION

In this section, we first justify that TTN can keep the upper bounds on approximation error and estimation error, respectively. Then, we show that TTN can maintain the characteristic of over-parametrized DNNs and own fewer parameters than the DNN.

To begin with, we reformulate the generalized loss of the tensor-to-vector regression over the ERM $\hat{h} \in \mathbb{T}$, that is

$$\begin{aligned}
\mathcal{L}(\hat{h}) &\leq \inf_{h\in\mathbb{T}} \mathcal{L}(h) + 2\sup_{h\in\mathbb{T}} |\mathcal{L}(h) - \hat{\mathcal{L}}(h)| + \mathcal{L}(h_\epsilon) - \inf_{h\in\mathbb{T}} \mathcal{L}(h) \\
&\leq \inf_{h\in\mathbb{T}} \mathcal{L}(h) + 2\hat{\mathbb{R}}_S(\mathbb{T}) + \epsilon.
\end{aligned} \tag{11}$$

where h_ϵ denotes a TTN based tensor-to-vector function with the optimization bias ϵ.

A. Analyzing the upper bound on the approximation error

As for the upper bound on the approximation error, we first justify Theorem 2, where we substitute TTN based tensor-to-vector regression for DNN based vector-to-vector regression.

Theorem 2. *For a smooth tensor-to-vector regression target function h^*, we can find a TTN h associated with a DNN $\bar{f}$ with $k(k \geq 2)$ modified smooth ReLU based hidden layers in which 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, then we can derive that*

$$\inf_{h\in\mathbb{T}} \mathcal{L}(h) = ||h^* - h||_1 = \mathcal{O}\left(\frac{q}{\prod_{t=1}^{M}(n_k^{[t]}+k-1)^{\frac{r}{d}}}\right),$$

where r is the differential order of $\bar{f}$, and $n_k = \prod_{t=1}^{M} n_k^{[t]}$.

Proof. $\forall t \in [M]$, we assume $h^* = \prod_{t=1}^{M} h_t^*$, and $h = \prod_{t=1}^{M} h_t$ in which h_t corresponds to the t^{th} core tensor of the top hidden layer of h. Thus, $||h^* - h||_1$ is decomposed to a multiplication of $||h_t^* - h_t||_1, t \in [M]$. From the upper bound on the approximation error, the tensor-train representation of $\bar{f}$ is invariant to the depth k, and we know that

$$||h_t^* - h_t||_1 = \mathcal{O}\left(\frac{q_i}{(n_k^{[t]}+k-1)^{\frac{r}{d}}}\right), \tag{12}$$

Thus, we derive

$$\begin{aligned}
||h^* - h||_1 &= \left|\left|\prod_{t=1}^{M} h_t^* - \prod_{t=1}^{M} h_t\right|\right|_1 \\
&\leq \prod_{t=1}^{M} ||h_t^* - h_t||_1 \\
&= \mathcal{O}\left(\prod_{t=1}^{M} \frac{q_i}{(n_k^{[t]}+k-1)^{\frac{r}{d}}}\right) \\
&= \mathcal{O}\left(\frac{q}{\prod_{t=1}^{M}(n_k^{[t]}+k-1)^{\frac{r}{d}}}\right).
\end{aligned} \tag{13}$$

where we assume that $q = \prod_{t=1}^{M} q_i$. □

B. Analyzing the upper bound on the estimation error

The bound on the estimation error was derived as

$$\begin{aligned}
& 2\sup_{h\in\mathbb{T}} |\mathcal{L}(h) - \hat{\mathcal{L}}(h)| \leq 2\hat{\mathcal{R}}_S(\mathbb{T}) \leq \frac{2q\Lambda'\Lambda^{k-1}v}{\sqrt{N}} \\
s.t.,\quad & ||\hat{W}_k(i,:)||_1 \leq \Lambda^{'}, \forall i \in [q] \\
& ||\hat{W}_j(a,:)||_2 \leq \Lambda, \forall j \in [k-1], a \in [n_j] \\
& ||\hat{\mathbf{x}}||_2 \leq v,
\end{aligned} \tag{14}$$

where $\hat{W}_k$ and $\hat{W}_i$ refer to the matricization of weight tensors of the top hidden layer k and the other hidden layer $i \in [k-1]$, respectively. Besides, $\hat{\mathbf{x}}$ denotes the tensorization of the input tensor.

The bound (14) is consistent with the bound on the estimation error of the DNN based vector-to-vector regression. We can simply justify the bound by folding all the weight

Authorized licensed use limited to: University of Science & Technology of China. Downloaded on August 17,2021 at 01:25:53 UTC from IEEE Xplore. Restrictions apply.

tensors into matrices such that the TTN was transformed back to the corresponding DNN. Based on the upper bound on the estimation error for the DNN based vector-to-vector regression, we obtain the bound (14).

C. Analyzing the upper bound on the optimization error

As shown in the introduction section, the tensor-train representation of DNN is that it can keep the benefits of the over-parametrization, while significantly lowering the number of parameters required for the model setup. In particular, the upper bound (9) for the optimization error still holds for TTN, which is

$$\epsilon = \mathcal{L}(h_\epsilon) - \inf_{h\in\mathbb{T}} \mathcal{L}(h) \leq \frac{\mu M^2 \beta}{2\gamma}. \tag{15}$$

However, the γ-PL condition is ensured to be satisfied for a TTN with the characteristic of DNN over-parametrization. Unlike an over-parametrized DNN, the requirement of over-parametrization for the number of TTN parameters can be much lower because of the tensor-train compact representation for the DNN.

D. Aggregating the three error bounds

By aggregating the bounds on the approximation, estimation, and optimization errors, we can finally derive the upper bound on $\mathcal{L}(\hat{h})$ as

$$\begin{aligned}
&\mathcal{L}(\hat{h}) \leq \inf_{h\in\mathbb{T}} \mathcal{L}(h) + 2\hat{\mathcal{R}}(\mathbb{T}) + \epsilon \\
&\quad \leq \mathcal{O}\left(\frac{q}{\prod_{t=1}^{M}(n_k^{[t]}+k-1)^{\frac{r}{d}}}\right) + \frac{2q\Lambda'\Lambda^{k-1}v}{\sqrt{N}} + \frac{\mu M^2\beta}{2\gamma}. \\
&\text{s.t., Smooth ReLU: } \lim_{t\to+\infty} \frac{1}{t}\ln(1+\exp(tx)) \\
&\quad \text{Hidden Layers: } \prod_{t=1}^{M} n_j^{[t]} \geq d+2, \forall j \in [k] \\
&\quad \text{Regularization: } ||\hat{W}_k(i,:)||_1 \leq \Lambda', \forall i \in [q] \\
&\qquad\qquad ||\hat{W}_j(m,:)||_2 \leq \Lambda, \forall j \in [k-1], m \in [n_j] \\
&\qquad\qquad ||\hat{\mathbf{x}}||_2 \leq v \\
&\quad \text{Over-parametrization: The number of parameters of the} \\
&\qquad\qquad \text{matricization of TTN is greater than } N.
\end{aligned} \tag{16}$$

V. MAE ESTIMATION

This section presents how to use the bounds (10) and (16) to estimate MAE values of DNN based vector-to-vector regression and TTN based tensor-to-vector regression, respectively.

A. MAE estimation for DNN based vector-to-vector regression

As shown in Proposition 2 in our work [6], we can estimate the approximation error, estimation error, and optimization error based on Proposition 1 as follows:

Proposition 1. *For a smooth function $\hat{f}: \mathbb{R}^d \to \mathbb{R}^q$, we use N training data to well-train a f_{DNN} with k smooth ReLU based hidden layers ($k \geq 2$). Then, the MAE loss can be derived as*

$$MAE(\hat{f}, f) \leq \frac{cq}{(n_k+k-1)^{\frac{r}{d}}} + \frac{2q\Lambda'\Lambda^{k-1}v}{\sqrt{N}} + b, \tag{17}$$

where constants c and b are set based on Eqs. (18) and (19), respectively.

$$c = \frac{(MAE_1 - MAE_2) l_1^{r/d} l_2^{r/d}}{q(l_2^{r/d} - l_1^{r/d})}, \tag{18}$$

$$b = \max(MAE_1 - \frac{(MAE_1 - MAE_2) l_2^{r/d}}{l_2^{r/d} - l_1^{r/d}} - \frac{2q\Lambda' v}{\sqrt{N}}, 0). \tag{19}$$

where MAE_1 and MAE_2 are two practical MAE loss values of two artificial neural networks (ANNs) with hidden units l_1 and l_2, respectively.

B. MAE estimation for TTN based tensor-to-vector regression

As to the TTN based tensor-to-vector regression, we change the bound (17) to (20).

$$MAE(\hat{f}, f) \leq \frac{cq}{\prod_{t=1}^{M}(n_k^{[t]}+k-1)^{\frac{r}{d}}} + \frac{2q\Lambda'\Lambda^{k-1}v}{\sqrt{N}} + b. \tag{20}$$

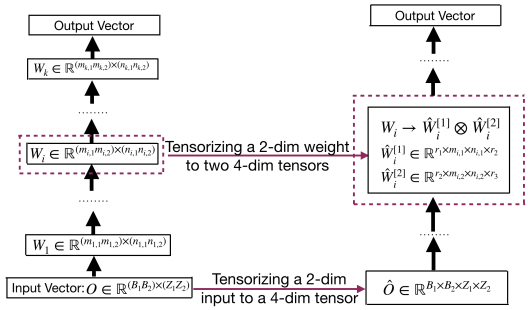
Besides, we use Eqs. (18) and (19) where $l_1 = \prod_{t=1}^{M}(l_1^{[t]} + k - 1)$ and $l_2 = \prod_{t=1}^{M}(l_2^{[t]} + k - 1)$.

VI. EXPERIMENTS

A. Experimental Goals

This section discusses TTN based vector-to-vector regression for digit image de-noising and associates the empirical results with our theorems shown in the previous sections. In particular, we aim at verifying the following conjectures:

- An over-parametrized DNN can reduce the training MAE loss down to 0, while the testing MAE loss can be also lowered to 0.
- TTN with the tensor-train representation of the corresponding DNN can maintain the performance of DNN based vector-to-vector regression, but TTN owns fewer parameters.
- The performance of TTN for tensor-to-vector regression is consistent under different noisy SNR levels.



(a) DNN for Vector-to-Vector Regression (b) TTN for Tensor-to-Vector Regression

Figure 1. *Transforming a DNN-based vector-to-vector regression into a TTN-based tensor-to-vector regression.*

Authorized licensed use limited to: University of Science & Technology of China. Downloaded on August 17,2021 at 01:25:53 UTC from IEEE Xplore. Restrictions apply.

B. Data preparation

This section presents our experiments of the image denoising based on the MNIST dataset. The original MNIST dataset consists of 60000 and 10000 for clean digit images for training and testing, respectively. To create two different noisy datasets, we separately mix the datasets with the additive Gaussian noises (AGN) with 10dB and 5dB SNR levels.

The DNN based vector-to-vector regression used in our experiments employs a feed-forward neural network architecture, where the inputs were 784-dimensional feature vectors of the noisy images, and the outputs were 784-dimensional feature vectors of either clean or enhanced images. The references of clean image 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 corresponded to the feature vectors of the enhanced images for an evaluation during the testing phase. Besides, there were 4 hidden layers configured for DNN, and the hidden layers followed the structure 1024-1024-1024-2048, where the smooth ReLU activation as Eq. (7) was utilized. The related setups of hidden layers satisfy the condition of over-parametrization because $1024 \times 1024 + 1024 \times 1024 + 1024 \times 2048 + 768 \times 1024 + 2048 \times 768 = 6553600 > 60000$. The SGD optimizer with a learning rate 10^{-3} and a momentum rate of 0.4 was used for the update of model parameters. The weights of the first $k-1$ hidden layers were 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. (12), and we assume Λ' as the maximum value of $(||W_k(1,:)||_1, ..., ||W_k(q,:)||_1)$.

As shown in Figure 1, we employed tensor-train decomposition to the used DNN for composing a 4-order TTN, where $\forall i \in [k]$ and the given ranks $\{1, r_1, r_2, r_3, 1\}$, the weight $W_i \in \mathbb{R}^{(m_{i,1}m_{i,2})\times(n_{i,1}\times n_{i,2})}$ was factorized into two tensors $\hat{W}_i^{[1]} \in \mathbb{R}^{r_1\times m_{i,1}\times m_{i,1}\times r_2}$ and $\hat{W}_i^{[2]} \in \mathbb{R}^{r_2\times m_{i,2}\times m_{i,2}\times r_3}$. Accordingly, the input vector $O \in \mathbb{R}^{(B_1B_2)\times(Z_1Z_2)}$ was decomposed to a 4-order tensor format as $\hat{O} \in \mathbb{R}^{B_1\times B_2\times Z_1\times Z_2}$. To be more specific, given the ranks $\{1, 2, 2, 2, 1\}$, the hidden layers of TTN were configured as $4\times8\times4\times8-4\times8\times4\times8-4\times8\times4\times8-8\times4\times8\times8$, and the input tensors follow a tensor format of $4 \times 7 \times 4 \times 7$.

C. Experimental results

Figure 2 and 3 show the trend of MAE losses of both DNN and TTN based vector-to-vector regression on the noisy MNIST datasets during the first 12 training epochs. Table I and III compare DNN and TTN in terms of number of model parameters, SNR levels, and final MAE loss on both training and testing datasets after 100 training epochs.

Table I
THE PERFORMANCE OF DNN AND TTN ON THE NOISY MNIST DATASET UNDER A 5DB SNR LEVEL AFTER 100 ITERATIONS.

Model	MAE (Train)	MAE (Test)	Parameter	SNR
DNN	0.0204	0.0229	5.56M	11.29
TTN	0.0171	0.0170	0.0019M	11.54

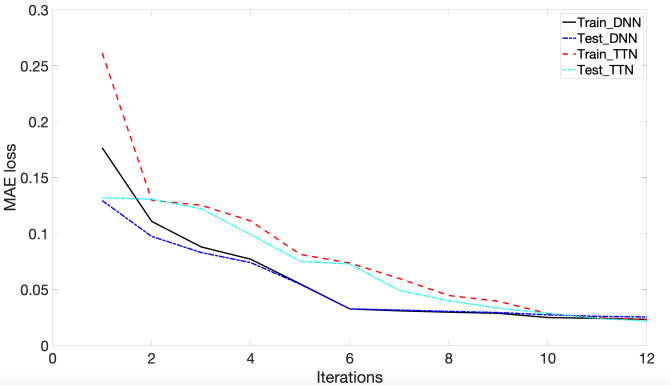


Figure 2. *The MAE losses on noisy MNIST dataset with a 10dB SNR level.*

Table II
THE MAE ESTIMATION FOR DNN AND TTN ON THE NOISY MNIST DATASET UNDER A SNR AT 5DB.

Model	AE	EE	OE	MAE_B
DNN	0.0034	0.0287	0.0031	0.0352
TTN	2.776×10^{-6}	0.0282	0.0036	0.0318

The curves in Figure 2 and 3 demonstrates that DNN and TTN converge to close MAE values after 10 iterations, but both DNN and TTN eventually converges to the similarly low values on the training datasets after 10 iterations. Moreover, the MAE loss on the testing datasets exhibits a similar trend. The MAE loss on the testing datasets for DNN separately goes down from 0.1298 and 0.1352 to 0.0256 and 0.0332 under SNRs of 10dB and 5dB, while the related values for TTN decrease from 0.1321 and 0.1322 to 0.0220 and 0.0315, respectively.

Experimental results for DNN based vector-to-vector regression exactly correspond to the over-parametrization condition, which can lower the training loss close to 0. More importantly, the results for TTN maintain the over-parametrization characteristic, which verifies our main theory in this work.

Furthermore, the results in Table I and III show that TTN can achieve even better performance in terms of the lower

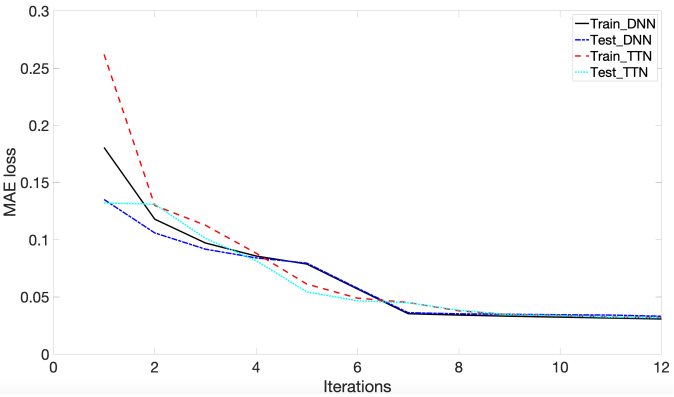


Figure 3. *The MAE losses on noisy MNIST dataset with a 5dB SNR level.*

Authorized licensed use limited to: University of Science & Technology of China. Downloaded on August 17,2021 at 01:25:53 UTC from IEEE Xplore. Restrictions apply.

Table III
THE PERFORMANCE OF DNN AND TTN ON THE NOISY MNIST DATASET UNDER A 10DB SNR LEVEL AFTER 100 ITERATIONS.

Model	MAE (Train)	MAE (Test)	Parameter	SNR
DNN	0.0269	0.0327	5.56M	15.43
TTN	0.0300	0.0299	0.0019M	15.89

Table IV
THE MAE ESTIMATION FOR DNN AND TTN ON THE NOISY MNIST DATASET UNDER A SNR AT 10DB.

Model	AE	EE	OE	MAE_B
DNN	0.0057	0.0274	0.0098	0.0416
TTN	4.653×10^{-6}	0.0261	0.0111	0.0372

MAE loss on the testing datasets (0.0170 vs. 0.0229 (10dB), 0.0299 vs. 0.0327 (5dB)) and higher SNR levels (11.54 vs. 11.29 (10dB), 15.89 vs. 15.43 (5dB)). In the meanwhile, the parameters of TTN is much less than the DNN (0.0019Mb vs. 5.56Mb), which suggests that the characteristics of over-parametrized DNN can be maintained by the corresponding TTN with a few numbers of parameters. Furthermore, the TTN with significantly fewer parameters can obtain even better performance.

Besides, Table II and IV shows the estimated MAE values on the testing datasets under SNR levels at 10dB and 5dB, respectively. We separately list the approximation error (AE), estimation error (EE), optimization error (OE), and an overall MAE bounded value (MAE_B). Our estimated results show that TTN can achieve lower MAE_B scores (0.0318 vs. 0.0352 (10dB), 0.0372 vs. 0.0416 (5dB)). This is mainly because AE values for TTN can be significantly reduced, while EE and OE scores are kept to be quite close. Hence, our estimated MAE scores based on our theorems can offer tight upper bound to the practical MAE scores and they also correspond to our experiments that TTN can be used for relaxing the over-parametrization condition required for DNN.

VII. CONCLUSIONS

This work discusses a performance analysis of tensor-train decomposition applied to DNN based vector-to-vector regression. We first discuss our findings on the DNN based vector-to-vector regression [6], and then we generalize and justify them for the TTN case. In particular, we show that the benefits of the over-parametrization condition for DNN can be transferred to TTN because a TTN owns fewer parameters than its related DNN, and the characteristics of an over-parametrized DNN can be shared with the TTN.

Our experiments of digit image de-noising on the MNIST datasets verify our theorems. On one hand, the MAE losses of both DNN and TTN can consistently follow the condition of the over-parametrization in different noisy backgrounds. On the other hand, TTN with fewer parameters can obtain even better performance than DNN in terms of the MAE losses and the enhanced SNR levels. Furthermore, our estimated MAE values based on our bounds in theorems correspond to the experimental results and verify the advantages of TTN.

REFERENCES

[1] J. Xie, L. Xu, and E. Chen, "Image denoising and inpainting with deep neural networks," in *Proc. Advances in Neural Information Processing Systems (NIPS)*, 2012, pp. 341–349.

[2] 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.

[3] A. Bérard, L. Besacier, A. C. Kocabiyikoglu, and O. Pietquin, "End-to-end automatic speech translation of audiobooks," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 6224–6228.

[4] 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 (TASLP)*, vol. 23, no. 1, pp. 7–19, 2015.

[5] 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.

[6] J. Qi, J. Du, S. M. Siniscalchi, X. Ma, and C.-H. Lee, "Analyzing upper bounds on mean absolute errors for deep neural network based vector-to-vector regression," *submitted to IEEE Transactions on Signal Processing (TSP)*.

[7] 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.

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

[9] S. Vaswani, F. Bach, and M. Schmidt, "Fast and faster convergence of sgd for over-parameterized models and an accelerated perceptron," *arXiv preprint arXiv:1810.07288*, 2018.

[10] R. Bassily, M. Belkin, and S. Ma, "On exponential convergence of SGD in non-convex over-parametrized learning," *arXiv preprint arXiv:1811.02564*, 2018.

[11] 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 (NIPS)*, 2018, pp. 3036–3046.

[12] 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.

[13] D. Csiba and P. Richtárik, "Global convergence of arbitrary-block gradient methods for generalized polyak-łojasiewicz functions," *arXiv preprint arXiv:1709.03014*, 2017.

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

[15] A. Novikov, D. Podoprikhin, A. Osokin, and D. P. Vetrov, "Tensorizing neural networks," in *Advances in Neural Information Processing Systems (NIPS)*, 2015, pp. 442–450.

[16] 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.

[17] Y. Li and Y. Liang, "Learning overparameterized neural networks via stochastic gradient descent on structured data," in *Proc. Advances in Neural Information Processing Systems (NIPS)*, 2018, pp. 8157–8166.

[18] V. Vapnik, "Principles of risk minimization for learning theory," in *Proc. Advances in Neural Information Processing Systems (NIPS)*, 1992, pp. 831–838.

Authorized licensed use limited to: University of Science & Technology of China. Downloaded on August 17,2021 at 01:25:53 UTC from IEEE Xplore. Restrictions apply.