High-order Learning Model via Fractional Tensor Network Decomposition

Chao Li, Zhun Sun, and Qibin Zhao
RIKEN Center for Advanced Intelligence Project (AIP), Tokyo, Japan
{chao.li, qibin.zhao}@riken.jp

Abstract

We consider high-order learning models, of which the weight tensor is represented by symmetric tensor network (TN) decomposition. Although such models have been widely used on various tasks, it is challenging to determine the optimal order in complex systems (e.g., deep neural networks). To tackle this issue, we introduce a new notion of *fractional tensor network (FrTN)* decomposition, which generalizes the conventional TN models by allowing the order to be an arbitrary fraction. Due to the density of fractions in the field of real numbers, the order of the model can be formulated as a learnable parameter and simply optimized by stochastic gradient descent (SGD) and its variants. We apply the proposed model to enhancing the classic ResNet-26/50 [13] and MobileNet-v2 [35] on both CIFAR-10 and ILSVRC-12 classification tasks, and the results demonstrate the effectiveness attributed to the learnable order parameters in FrTN.

1 Introduction

High-order methods are natural extensions to enhance the expressive power of learning models [2, 19, 41]. Taylor's theorem states that a smooth function can be well approximated by a polynomial with a sufficiently high degree [39]. However, there are two drawbacks which severely limit the application of the vanilla high-order models in practice: **(i)** The dimension of learnable weights would exponentially grow with increasing of the model order; **(ii)** Improper model order would break the "bias-variance" [21] balance of the model, yielding unsatisfactory generalization performance.

The issue (i) fortunately can be addressed by leveraging tensor network (TN) decomposition [9], which aims to parameterize the high-order weight tensor by a collection of low-dimensional factors (*a.k.a.*, core tensors) [5]. Recently, TN-based methods are widely applied to important machine learning tasks [16, 23, 28, 30, 38] to name a few, among which various TN decomposition models are exhaustively studied. However, few discussions are focusing on the issue (ii), even though the performance would be dramatically influenced by the order of the model [16, 18]. Therefore, it is of importance to develop methods, which can efficiently select or learn the most suitable (optimal) order parameters in practice.

The issue of order determination. Given the feature vector $\mathbf{x} \in \mathbb{R}^I$, weight tensor $\mathcal{W} \in \mathsf{S}^P(\mathbb{R}^I)$ and bias $b \in \mathbb{R}$, we consider the order-P learning model as

$$y = \Phi\left(\mathcal{W} \times_1 \mathbf{x} \times_2 \cdots \times_P \mathbf{x} + b\right) = \Phi\left(\left\langle \mathcal{W}, \mathbf{x}^{\otimes P} \right\rangle + b\right), \tag{1}$$

where $\Phi(\,\cdot\,)$ denotes a non-linear mapping over $\mathbb{R}$. We see that the dimension of $\mathcal{W}$ would exponentially increase when the order P goes larger. The TN models are therefore applied to representing $\mathcal{W}$ by lower-dimensional core tensors. However, how to determine the optimal order P is still challenging: In Eq. (1), the order P is reflected by sum of exponents of its monomials, which is multivariate and discrete, such that the optimization on the discrete domain is usually NP-hard [36].

First Workshop on Quantum Tensor Networks in Machine Learning, 34th Conference on Neural Information Processing Systems (NeurIPS 2020).

The popularly-used methods, in the existing works, is to apply exhaustive search on all possible candidates [16, 18, 30, 42]. However, the computational cost is apparently prohibited in large-scale applications. Although considerable attention in network architecture search (NAS) reboots the studies on evolutionary algorithms [8, 25, 27, 33], which might be promising on this issue, it remains unexplored to determine the order of TN-based methods by NAS.

Polynomial models. In this paper, we borrow the idea of the polynomial models, which names the essence of the "high-order learning models" in recent works, for instance, the polynomial models usually appear in polynomial network [4, 19, 26], neural tensor network [37], bi-linear attention [20]. Most of the studies exploit the theoretical simplicity or rich interactions among features brought by polynomials. However, there are few studies focusing on how to efficiently determine the optimal order of the polynomial for learning models. In contrast, we extend a class of polynomials into the fractional domain, such that the optimal order parameters for specific tasks can be efficiently learned during the training phase.

Main contribution. In this paper, we establish a new notion of *fractional tensor network* (FrTN) decomposition by generalizing the order of conventional TN models from integer to fractional domain. Since fractions are dense in the field of real numbers, the order of model, as a parameter, can be simply learned by stochastic gradient decent (SGD) and its variants. Based on this, we develop FrTN-based learning models, which can be used as basic building blocks in neural networks, and numerically prove the effectiveness in classification tasks.

2 Preliminaries and basic setup

2.1 Tensor network (TN) decomposition

Below, we briefly review several tensor network (TN) models used in the following sections, and point out the equivalence between super-symmetric tensors and homogeneous polynomials. Throughout the paper, we define an order-P *tensor* over the complex vector space $\mathbb{C}^I$ as a multi-dimensional array of complex numbers [22] indexed by integer tuples $(i_1, \ldots, i_P)$ with $1 \leq i_j \leq I$, $j \in [P]$, i.e.

$$\mathcal{W} = \mathcal{W}(i_1, \ldots, i_P)_{1 \leq i_1, \ldots, i_P \leq I}. \tag{2}$$

Denote by $\mathsf{T}^P(\mathbb{C}^I)$ the space containing all order-P tensors over $\mathbb{C}^I$. The tensor $\mathcal{W}$ is *super-symmetric* if $\mathcal{W}(i_1, \ldots, i_P)$ is invariant under all permutations of $(i_1, \ldots, i_P)$. Denote by $\mathsf{S}^P(\mathbb{C}^I)$ the linear subspace of all super-symmetric tensors in $\mathsf{T}^P(\mathbb{C}^I)$.

For a vector $\mathbf{x} \in \mathbb{C}^I$, the mode-$m$ *tensor-vector* product with $\mathcal{W}$ is denoted by $\mathcal{Y} = \mathcal{W} \times_m \mathbf{x} \in \mathsf{T}^{P-1}(\mathbb{C}^I)$. Element-wisely, we have [6]

$$\mathcal{Y}(i_1, \ldots, i_{m-1}, i_{m+1}, \ldots, i_P) = \sum_{i_m=1}^{I} \mathcal{W}(i_1, i_2, \ldots, i_P) \cdot \mathbf{x}(i_m). \tag{3}$$

The Pth tensor power of a vector, denoted by $\mathbf{x}^{\otimes P} \in \mathsf{S}^P(\mathbb{C}^I)$, is defined as outer products of $\mathbf{x}$ with itself by P times, i.e., $\mathbf{x}^{\otimes P} = \otimes_{m=1}^{P} \mathbf{x}$ such that for all $1 \leq i_1, \ldots, i_P \leq I$ it obeys $\mathbf{x}^{\otimes P}(i_1, \ldots, i_P) = \mathbf{x}(i_1) \cdots \mathbf{x}(i_P)$.

Symmetric CP decomposition. Given $\mathcal{W} \in \mathsf{S}^P(\mathbb{C}^I)$, there always exists $\{\mathbf{g}_1, \ldots, \mathbf{g}_R\} \in \mathbb{C}^I$ such that

$$\mathcal{W} = (\mathbf{g}_1)^{\otimes P} + \cdots + (\mathbf{g}_R)^{\otimes P} = [\![\underbrace{\mathbf{G}, \cdots, \mathbf{G}}_{P \text{ copies}}]\!], \tag{4}$$

where the matrix $\mathbf{G} = [\mathbf{g}_1, \ldots, \mathbf{g}_R] \in \mathbb{C}^{I \times R}$ denotes the "core-tensor", and the symbol $[\![\cdots]\!]$ denotes the Kruskal product [22]. The smallest R in Eq. (4) is called the symmetric rank [7] of $\mathcal{W}$, and Eq. (4) is called a *symmetric CANDECOMP/PARAFAC (SCP)* decomposition.

Symmetric tensor ring decomposition. Like the SCP model, we introduce the symmetric tensor ring (STR) decomposition based on the work [43]. Specifically, for every $\mathcal{W} \in \mathsf{S}^P(\mathbb{C}^I)$, its STR decomposition is given by

$$\mathcal{W}(i_1, \ldots, i_P) = tr\left(\mathcal{G}(:, i_1, :) \cdot \mathcal{G}(:, i_2, :) \cdots \mathcal{G}(:, i_P, :)\right), \quad \forall 1 \leq i_j \leq I,\ j \in [P], \tag{5}$$

where $tr(\cdot)$ denotes the trace operation, and $\mathcal{G} \in \mathbb{C}^{R\times I\times R}$ denotes the core tensor. Furthermore, *we assume that $\mathcal{G}$ has Hermitian slices*, i.e., $\mathcal{G}(:,i,:) = \mathcal{G}(:,i,:)^H,\ i \in [I]$[1]. We prove the following proposition to guarantee the existence of STR decomposition for all super-symmetric tensors:

Proposition 1 *Given $\mathcal{W} \in \mathsf{S}^P(\mathbb{C}^I)$ of the symmetric rank R, there always an order-3 tensor $\mathcal{G} \in \mathbb{C}^{r\times I\times r}$, where $r \leq R$ and $\mathcal{G}(:,i,:) = \mathcal{G}(:,i,:)^H$, such that Eq. (5) holds.*

Proposition 1 implies that for an arbitrary super-symmetric tensor, it can be decomposed by Eq. (5), of which the "TR-rank" [44] is not larger than its symmetric rank.

Note: Tensor train (TT) [31] is known as a special case of TR when there exist two adjacent core tensors, which degenerate into matrices. Although the topological structure of TT is non-symmetric, we introduce the *partial symmetric tensor train* (pSTT) model induced by the STR model. Specifically, we modify Eq. (5) as

$$\mathcal{W}^\star(i_1,\ldots,i_P) = \mathbf{A}(i_1,:)\cdot\mathcal{G}(:,i_2,:)\cdots\mathcal{G}(:,i_{P-1},:)\cdot\mathbf{A}(:,i_P), \tag{6}$$

where $\mathcal{W}^\star$ denotes a partial-symmetric tensor over $(i_2,\ldots,i_{P-1})$, i.e., entries of $\mathcal{W}$ is invariant under all permutation of indices except i_1 and i_P.

Equivalence to degree-P polynomial. The tensor $\mathcal{W} \in \mathsf{S}^P(\mathbb{C}^I)$ defines an degree-P homogeneous polynomial $f(\mathbf{x}) \in \mathbb{C}[x_1,\ldots,x_P]$ [29, 32]:

$$f(\mathbf{x}) = \left\langle \mathcal{W}, \mathbf{x}^{\otimes P}\right\rangle = \sum_{i_1,\ldots,i_P=1}^{I} \mathcal{W}(i_1,\ldots,i_P)\mathbf{x}(i_1)\cdots\mathbf{x}(i_P), \tag{7}$$

where $\mathbf{x} = [x_1,\ldots,x_P]^\top$, and $\langle\cdot,\cdot\rangle$ denotes the trivial inner product of two tensors. It is easy to see from Eq. (7) that every $\mathcal{W} \in \mathsf{S}^P(\mathbb{C}^I)$ may be uniquely associated with a homogeneous polynomial of degree-P in I variables [3]. In our work, such equivalence allows attacking the problem of TNs as one of homogeneous polynomials.

3 Fractional tensor network (FrTN) induced learning models

To learn the order P, our main idea is to extend the available range of P, such that gradient-based optimization methods can be applied. To do so, we introduce the notion of fractional tensor network (FrTN) decomposition, which allows the order of TN to own fractional components. For each new model, we will also discuss its application in the learning models.

3.1 Fractional SCP decomposition

We introduce the FrTN decomposition by its polynomial form. Given a tensor $\mathcal{W} \in \mathsf{S}^P(\mathbb{C}^I)$, as aforementioned, we have its equivalent homogeneous polynomial form as Eq. (7). In this form, the tensor's order corresponds to the degree of a polynomial. To construct a specific order parameter in the model, we decompose Eq. (7) as a sum of Pth power of linear forms, *i.e.*,

$$f(\mathbf{x}) = \sum_{r=1}^{R}\left(\mathbf{g}_r(1)\mathbf{x}(1) + \mathbf{g}_r(2)\mathbf{x}(2) + \cdots + \mathbf{g}_r(I)\mathbf{x}(I)\right)^P = \sum_{r=1}^{R}\left\langle \mathbf{g}_r, \mathbf{x}\right\rangle^P. \tag{8}$$

It is known that such decomposition does always exist, and the coefficients $\mathbf{g}_r \in \mathbb{C}^I,\ r \in [R]$ correspond the result of SCP decomposition of $\mathcal{W}$ [3, 15]. We see that the order of $\mathcal{W}$, originally defined as the number of indices, is successfully converted into the power function $(\cdot)^P$ in Eq. (8). In the conventional definition of tensor (network), the order P is constrained in non-negative integers [], i.e., $P \in \mathbb{Z}^{\geq 0}$, while in this work we extend the available range of P into all possible fractions:

Definition 1 (FrSCP decomposition) *Given a fractional number $\bar{P} \in \mathbb{Q}^{\geq 0}$, an order-$\bar{P}$ symmetric CP (SCP) decomposition of dimension I is defined by the fractional form of polynomial (8), of which P is replaced by $\bar{P}$.*

[1]Here we apply the Matlab® syntax to representing the slicing operation.

Note that **(a)** Def. 1 naturally degenerate into the conventional SCP decomposition when assuming $\bar{P}$ as non-negative integers, where the coefficients $\mathbf{g}_r$'s construct the corresponding core tensor of the decomposition, and **(b)** we cannot precisely represent the SCP decomposition when $\bar{P}$ is fractional.

FrSCP-based learning model The corresponding learning model is obtained by representing the weight tensor $\mathcal{W}$ in Eq. (1) using its FrSCP form:

$$y^{FrSCP} = \Phi\left(\sum_{r=1}^{R} \langle \mathbf{g}_r, \mathbf{x}\rangle^{\bar{P}} + b\right). \tag{9}$$

Compared to Eq. (1), the output of Eq. (9) is calculated by sum of $\bar{P}$th power of linear forms. Since models in learning are generally defined over $\mathbb{R}$, we need to further assume the *non-negativity* of the inner product in Eq. (9) to guarantee the existence of gradient. In practice like neural networks, the assumption can be simply satisfied by applying a rectifier activation [10] before the power function.

3.2 Fractional STR decomposition

Below, we prove that the STR-induced polynomial also has the similar form as the SCP model, i.e., the sum of powers. Specifically,

Proposition 2 *Assume that tensors $\mathcal{W} \in \mathsf{S}^P(\mathbb{C}^I)$ and $\mathcal{G} \in \mathbb{C}^{R\times I\times R}$ obey the STR decomposition defined as Eq. (5), then the polynomial (7) has the equivalent form:*

$$f(\mathbf{x}) = \left\langle \mathcal{W}, \mathbf{x}^{\otimes P}\right\rangle = \sum_{r=1}^{R} \sigma_r\left(\mathcal{G} \times_2 \mathbf{x}\right)^P, \tag{10}$$

where $\sigma_r(\,\cdot\,)$, $r \in [R]$ denotes the rth largest eigenvalue of a matrix.

In contrast to Eq. (8), as shown in Proposition 2, the Pth power function is employed on eigenvalues, which are always real numbers because of the Hermitian structure of core tensor $\mathcal{G}$. Thus, we define the fractional extension of STR as following:

Definition 2 (FrSTR decomposition) *Given a fractional number $\bar{P} \in \mathbb{Q}^{\geq 0}$, an order-$\bar{P}$ symmetric tensor ring (STR) decomposition of dimension I is defined by the fractional form of polynomial (10), of which P is replaced by $\bar{P}$.*

FrSTR-based learning model The model is given by representing $\mathcal{W}$ by its FrSTR form (10):

$$y^{FrSTR} = \Phi\left(\sum_{r=1}^{R} \sigma_r(\mathcal{G} \times_2 \mathbf{x})^{\bar{P}} + b\right), \tag{11}$$

following a positive semi-definite assumption on the slices $\mathcal{G}(:,i,:)$, $i \in [I]$. Compared to the FrSCP model, FrSTR has the property of unitary invariance over the tensor $\mathcal{G}$. Specifically,

Proposition 3 *Assume $\mathcal{G}_0 \in \mathbb{R}^{R\times I\times R}$ with positive semidefinite slices $\mathcal{G}(:,i,:)$, $i \in [I]$, then all tensors $\mathcal{G} \in \left\{\mathcal{G}_0 \times_1 \mathbf{U} \times_3 \mathbf{V} | \forall \mathbf{U}, \mathbf{V} \in \mathbb{R}^{R\times R}, \mathbf{U}\mathbf{U}^\top = \mathbf{U}^\top\mathbf{U} = \mathbf{V}\mathbf{V}^\top = \mathbf{V}^\top\mathbf{V} = \mathbf{I}\right\}$ do not change the value of Eq. (11) for all possible $\mathbf{x} \in \mathbb{R}^I$ and $P \in \mathbb{Q}^{\geq 0}$.*

As shown in Proposition 3, the prediction by Eq. (11) would not be changed by arbitrary rotation of $\mathcal{G}$ along the first and third indices.

Eq. (11) also have a norm-like form as the FrSCP-based model. With the same assumptions on Φ and b, we can rewrite Eq. (11) as $y^{FrSTR} = \|\mathcal{G} \times_2 \mathbf{x}\|_{S_{\bar{P}}}$, where $\|\,\cdot\,\|_{S_{\bar{P}}}$ denotes the matrix Schatten $\bar{P}$-(quasi)norm [1].

3.3 What if tensor train (TT)?

We cannot define "fractional tensor train (TT) decomposition" trivially using the aforementioned framework because of the non-symmetric topology of the TT decomposition. Therefore, for an order-P learning model, we assume an order-$(P+2)$ tensor $\mathcal{W}^\star \in \mathsf{T}^{P+2}(\mathbb{R}^I)$ with the pSTT format as Eq. (6). We can prove that the following equation holds:

Proposition 4 *Assuming the weight in Eq. (1) to be an order-$(P+2)$ tensor $\mathcal{W}^\star \in \mathsf{T}^{P+2}(\mathbb{R}^I)$ with the pSTT format as Eq. (6), the following equation holds:*

$$y^{pSTT} = \Phi\left(\left\langle \mathcal{W}^\star, \mathbf{x}^{\otimes P}\right\rangle + b\right) = \Phi\left(\sum_{r=1}^{R} a_r(\mathbf{x})\sigma_r(\mathcal{G}\times_2 \mathbf{x})^P + b\right). \tag{12}$$

In the equation, the scalar $a_r(\mathbf{x}) \geq 0$, $\forall r \in [R]$ denotes a non-negative weight on each eigenvalue, where it satisfies $a_r(\mathbf{x}) = \left(\mathbf{u}_r^\top \mathbf{A}\mathbf{x}\right)^2$ and $\mathbf{u}_r \in \mathbb{R}^R$ denotes the rth eigenvector w.r.t. $\sigma_r(\mathcal{G}\times_2 \mathbf{x})$.

Proposition 4 implies that an order-$(P+2)$ pSTT model results in an order-P "STR-based" learning model, of which an adaptive weighting trick is exploited on each eigenvalues, and the weights $a_r(\mathbf{x})$, $\forall r$ come from the marginal order-2 core tensor $\mathbf{A}$ in Eq. (6). Similarly, the fractional form of *pSTT-based model* can be obtained by extending the range of P in Eq. (12) into fractional domain. More interestingly, if we further assume the root operation on Φ and omit the bias, Eq. (12) can be rewritten as a weighted Schatten (quasi)norms [40], which are popularly used in the convex low-rank approximation of matrices and tensors.

3.4 A general form of FrTN in neural networks

In a nutshell, FrTN brings us three inspiring operations for DNNs compared to the linear models: **(a)** *multi-branch structures*, **(b)** *learnable power functions*, and **(c)** *adaptive weighting on branches*. Inspired by those operations, we generalize the FrTN-based models as a basic building block for both fully-connected (FC) and convolutional neural networks (CNNs). Given the input feature $\mathbf{x}$, the output features $\mathbf{y}_j$, $j \in [J]$ are given by

$$\mathbf{y}_j = \sum_{r=1}^{R} A_{r,\mathbf{x}} \cdot \left[\delta\left(\mathcal{G}_r \star \mathbf{x}\right) + 1\right]^{\bar{P}_r}, \tag{13}$$

where the operation $\star$ denotes the inner product in the FC layers or various convolutions in CNN, and $\delta(\cdot)$ denotes the rectified linear unit (Relu) [10]. In contrast to the orignal FrTN-based models, the power function $[\cdot]^{\bar{P}_r}$ in Eq. (13) is element-wise, and we assign different powers $\bar{P}_r$'s for different branches. Note that the rectifier δ guarantees that the gradient over the power $\bar{P}_r$'s do always exist, and the additional constant bias 1 avoids extreme gradient of $\bar{P}_r$ when the ouput of δ is close to zero.

4 Experiments and Discussions

Goal. Below, we apply the model (13) to two classic image recognition tasks, i.e., CIFAR-10 [24] and ILSVRC-12 [34]. The goal of the experiments is to verify whether the proposed learnable order parameters can boost the performance of DNNs in specific tasks.

Models	**Cifar-10** Res-26	**ILSVRC-12** Res-50	**ILSVRC-12** Mobile-v2
	err.	top 1 | 5 err.	top 1 | 5 err.
Baseline	11.10	23.91 | 7.11	27.55 | 10.23
CPD$_{P=2}$	10.63	25.19 | 9.07	27.41 | 9.29
CPD$_{P=3}$	13.53	24.04 | 7.26	66.75 | 41.18
$\bar{P}$	11.41	23.08 | 6.67	27.73 | 9.32
$\bar{P}_{\mathbf{x}}$	**8.17**	**22.77** | 6.54	**26.69** | **8.70**
$\bar{P} + A_{\mathbf{x}}$	9.50	23.05 | 6.68	26.88 | 8.97
$\bar{P}_{\mathbf{x}} + A_{\mathbf{x}}$	8.35	23.01 | 6.70	26.82 | 9.94
Maxout	10.59	23.75 | 6.98	27.68 | 9.56
ℓ_p-pooling	13.20	23.25 | 6.88	27.45 | 9.21
SE	10.71	22.90 | **6.45**	28.00 | 9.68

Table 1: Classification (%) error.

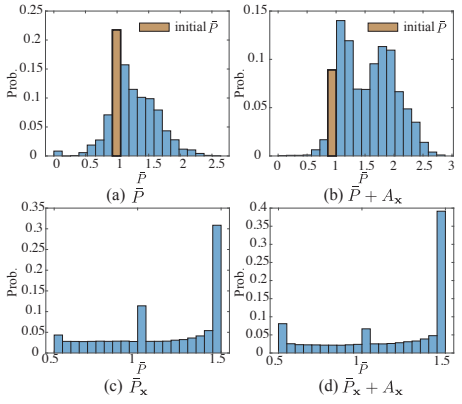

Figure 1: Histogram of $\bar{P}$ in MoibleNet-v2.

Setup. We evaluate our methods by two well-known building blocks in CNNs, i.e., the bottle-neck in ResNet [13], and its reversed variant in MobileNet [35]. In such units, we implement the model (13)

to replace the original 1×1 convolutional layers, which are exploited for the channel reduction. We totally consider 4 variants of Eq. (13) in the experiments, denoted by $\bar{P}$, $\bar{P}_{\mathbf{x}}$, $\bar{P} + A_{\mathbf{x}}$ and $\bar{P}_{\mathbf{x}} + A_{\mathbf{x}}$, where $\bar{P}$ and $\bar{P}_{\mathbf{x}}$ mean that only the order parameters are learned, while "$+A_{\mathbf{x}}$" means that the weights $A_{r,\mathbf{x}}$ in Eq. (13) is also taken into account. The subscript $\mathbf{x}$ in $\bar{P}_{\mathbf{x}}$ denotes that we also apply two FC layers sub-network to adaptively learning the order parameters. For stable training of the networks, we further clip $\bar{P}$'s within $[0.0,\ 6.0]$, and $\bar{P}_{\mathbf{x}}$ in $[0.5,\ 1.5]$. For comparison, we employ the CP decomposition (CPD) models with fixed orders $P = 2, 3$.

In the ILSVRC-12 task, we implement both ResNet50 (as Res-50 in Table 1) and MobileNet-v2 ((as Mobile-v2 in Table 1)) as baselines, while in Cifar-10 we construct a "Res-26" to represent a relatively shallow architecture. For training, we employ the SGD with moment equaling 0.9, SGD with Nestrov moment 0.9, and RMSProp ($\epsilon = 1.0$) [14] optimizers for Res-26, Res-50 and Mobile-v2, respectively.

Results. The classification error for the two tasks is shown in Table 1. As shown in Table 1, the variant $\bar{P}_{\mathbf{x}}$ with adaptively learnable order parameters, outperforms its counterparts. Note that the results of $CPD_{P=3}$ diverge in our experiment. We empirically find that models with higher orders generally lead to more unstable training dynamics. We also employ several related methods including ℓ_p-pooling [12], Maxout [11], SE [17], the connections are discussed below.

Fig. 1 shows the histogram of the order parameters $\bar{P}$ in MobileNet-v2, where the brown bars in sub-fig. (a,b) denote the initial value of $\bar{P}$. As shown in Fig. 1, the values of $\bar{P}$ and $\bar{P}_{\mathbf{x}}$ spread in the fractional domain. Note that the values of the order parameters obey a Gaussian-like distribution in cases of $\bar{P}$ and $\bar{P} + A_{\mathbf{x}}$, while in $\bar{P}_{\mathbf{x}}$ and $\bar{P}_{\mathbf{x}} + A_{\mathbf{x}}$ the values concentrate at the boundary ($\bar{P}_{\mathbf{x}} = 0.5$ and 1.5) and uniformly spread between. We infer that the adaptive method used in $\bar{P}_{\mathbf{x}}$ more encourages the order parameters to fully spread all available range, even going beyond. On the other hand, the $\bar{P}$ by directly learning tends to concentrate around the initial values.

5 Discussion

Connection to ℓ_p-pooling [12]. As shown in Eq. (9), the SCP-based model results in the similar architecture to ℓ_p-pooling among multiple branches and its special case "Maxout" [10] when assuming the $\bar{P}$th root of Φ. For each output feature, the SCP-based model first applies a linear projection of $\mathbf{G}$, and then compute the $\bar{P}$-norm of the projection as the output. However, the FrSTR and pSTT-based models generalize ℓ_P-pooling into a higher-order form. Unlike the SCP-based model, STR and pSTT consider the matrix Schatten $\bar{P}$-norm as pooling. It implies that more structural information like low-rankness can be obtained by matrix norm.

Connection to the "squeeze-and-excitation" operation [17]. Compared to FrSTR-based model, the pSTT-based model (12) incorporates additional non-negative weights $a_r,\ r \in [R]$ by "squeezing" $\mathbf{x}$ into scalars on each "eigenvalue-monomial" $\sigma_r(\mathcal{G} \times_2 \mathbf{x})^{\bar{P}}$. Because of its non-negativity, it implies a gate-like operation to control the contribution of each "monomial" for specific input features. A similar idea is applied to "squeeze-and-excitation" network (SE-Net) [17], in which each channel is also weighted. More notably, the weights in both SE-Net and the the pSTT-based model are calculated by a bottleneck with two "fully-connected (FC)" layers. As shown in Proposition 4, a_r is calculated by first projecting $\mathbf{x}$ into a R-dimensional latent space, which is generally lower-dimensional than i due to the low-rank fact of TT decomposition. Because of such similar structures, claim that we show a new insight for the SE operation from the tensor-algebraic perspective, i.e., *a learning unit with SE operations can be neatly modeled by the pSTT model.* In this paper, we extend the conventional tensor network (TN) decomposition into a novel framework, *i.e.*, FrTN, by assigning the order parameters with fractional components. The new models are defined by their polynomial forms, and applied to tackling the order-determination issue in learning problems. Meanwhile, we also reveal that FrTN can be used to interpret well-known methods including ℓ_p-pooling and SE-Net in deep learning by a novel yet unified tensor perspective.

References

[1] Rajendra Bhatia. *Matrix analysis*, volume 169. Springer Science & Business Media, 2013.

[2] Mathieu Blondel, Akinori Fujino, Naonori Ueda, and Masakazu Ishihata. Higher-order factorization machines. In *Advances in Neural Information Processing Systems*, pages 3351–3359, 2016.

[3] Jerome Brachat, Pierre Comon, Bernard Mourrain, and Elias Tsigaridas. Symmetric tensor decomposition. *Linear Algebra and its Applications*, 433(11-12):1851–1872, 2010.

[4] Grigorios G Chrysos, Stylianos Moschoglou, Giorgos Bouritsas, Jiankang Deng, Yannis Panagakis, and Stefanos Zafeiriou. $\backslash\pi$- nets: Deep polynomial neural networks. *arXiv preprint arXiv:2003.03828*, 2020.

[5] Andrzej Cichocki, Namgil Lee, Ivan Oseledets, Anh-Huy Phan, Qibin Zhao, Danilo P Mandic, et al. Tensor networks for dimensionality reduction and large-scale optimization: Part 1 low-rank tensor decompositions. *Foundations and Trends® in Machine Learning*, 9(4-5):249–429, 2016.

[6] Andrzej Cichocki, Rafal Zdunek, Anh Huy Phan, and Shun-ichi Amari. *Nonnegative matrix and tensor factorizations: applications to exploratory multi-way data analysis and blind source separation*. John Wiley & Sons, 2009.

[7] Pierre Comon, Gene Golub, Lek-Heng Lim, and Bernard Mourrain. Symmetric tensors and symmetric tensor rank. *SIAM Journal on Matrix Analysis and Applications*, 30(3):1254–1279, 2008.

[8] Thomas Elsken, Jan Hendrik Metzen, and Frank Hutter. Neural architecture search: A survey. *Journal of Machine Learning Research*, 20(55):1–21, 2019.

[9] Mark Fannes, Bruno Nachtergaele, and Reinhard F Werner. Finitely correlated states on quantum spin chains. *Communications in mathematical physics*, 144(3):443–490, 1992.

[10] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 315–323, 2011.

[11] Ian J Goodfellow, David Warde-Farley, Mehdi Mirza, Aaron Courville, and Yoshua Bengio. Maxout networks. *arXiv preprint arXiv:1302.4389*, 2013.

[12] Caglar Gulcehre, Kyunghyun Cho, Razvan Pascanu, and Yoshua Bengio. Learned-norm pooling for deep feedforward and recurrent neural networks. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 530–546. Springer, 2014.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[14] Geoffrey Hinton, Nitish Srivastava, and Kevin Swersky. Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. *Cited on*, 14(8), 2012.

[15] Frank L Hitchcock. The expression of a tensor or a polyadic as a sum of products. *Journal of Mathematics and Physics*, 6(1-4):164–189, 1927.

[16] Ming Hou, Jiajia Tang, Jianhai Zhang, Wanzeng Kong, and Qibin Zhao. Deep multimodal multilinear fusion with high-order polynomial pooling. In *Advances in Neural Information Processing Systems*, pages 12113–12122, 2019.

[17] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.

[18] Zihao Huang, Chao Li, Feng Duan, and Qibin Zhao. H-owan: Multi-distorted image restoration with tensor 1x1 convolution. *arXiv preprint arXiv:2001.10853*, 2020.

[19] Joe Kileel, Matthew Trager, and Joan Bruna. On the expressive power of deep polynomial neural networks. In *Advances in Neural Information Processing Systems*, pages 10310–10319, 2019.

[20] Jin-Hwa Kim, Jaehyun Jun, and Byoung-Tak Zhang. Bilinear attention networks. In *Advances in Neural Information Processing Systems*, pages 1564–1574, 2018.

[21] Ron Kohavi, David H Wolpert, et al. Bias plus variance decomposition for zero-one loss functions. In *ICML*, volume 96, pages 275–83, 1996.

[22] Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.

[23] Jean Kossaifi, Adrian Bulat, Georgios Tzimiropoulos, and Maja Pantic. T-net: Parametrizing fully convolutional nets with a single high-order tensor. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7822–7831, 2019.

[24] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[25] Jason Liang, Elliot Meyerson, and Risto Miikkulainen. Evolutionary architecture search for deep multitask networks. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 466–473, 2018.

[26] Roi Livni, Shai Shalev-Shwartz, and Ohad Shamir. On the computational efficiency of training neural networks. In *Advances in neural information processing systems*, pages 855–863, 2014.

[27] Zhichao Lu, Ian Whalen, Vishnu Boddeti, Yashesh Dhebar, Kalyanmoy Deb, Erik Goodman, and Wolfgang Banzhaf. Nsga-net: A multi-objective genetic algorithm for neural architecture search. 2018.

[28] Xindian Ma, Peng Zhang, Shuai Zhang, Nan Duan, Yuexian Hou, Ming Zhou, and Dawei Song. A tensorized transformer for language modeling. In *Advances in Neural Information Processing Systems*, pages 2229–2239, 2019.

[29] Jiawang Nie. Low rank symmetric tensor approximations. *SIAM Journal on Matrix Analysis and Applications*, 38(4):1517–1540, 2017.

[30] Alexander Novikov, Dmitrii Podoprikhin, Anton Osokin, and Dmitry P Vetrov. Tensorizing neural networks. In *Advances in neural information processing systems*, pages 442–450, 2015.

[31] Ivan V Oseledets. Tensor-train decomposition. *SIAM Journal on Scientific Computing*, 33(5):2295–2317, 2011.

[32] Liqun Qi. Eigenvalues of a real supersymmetric tensor. *Journal of Symbolic Computation*, 40(6):1302–1324, 2005.

[33] Esteban Real, Alok Aggarwal, Yanping Huang, and Quoc V Le. Regularized evolution for image classifier architecture search. In *Proceedings of the aaai conference on artificial intelligence*, volume 33, pages 4780–4789, 2019.

[34] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[35] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

[36] IV Sergienko and VP Shylo. Problems of discrete optimization: Challenges and main approaches to solve them. *Cybernetics and Systems Analysis*, 42(4):465–482, 2006.

[37] Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. Reasoning with neural tensor networks for knowledge base completion. In *Advances in neural information processing systems*, pages 926–934, 2013.

[38] Edwin Stoudenmire and David J Schwab. Supervised learning with tensor networks. In *Advances in Neural Information Processing Systems*, pages 4799–4807, 2016.

[39] Brook Taylor. *Methodus incrementorum directa & inversa*. Inny, 1717.

[40] Yuan Xie, Shuhang Gu, Yan Liu, Wangmeng Zuo, Wensheng Zhang, and Lei Zhang. Weighted schatten p-norm minimization for image denoising and background subtraction. *IEEE transactions on image processing*, 25(10):4842–4857, 2016.

[41] Jiyan Yang and Alex Gittens. Tensor machines for learning target-specific polynomial features. *arXiv preprint arXiv:1504.01697*, 2015.

[42] Rose Yu, Stephan Zheng, Anima Anandkumar, and Yisong Yue. Long-term forecasting using higher order tensor rnns. *arXiv preprint arXiv:1711.00073*, 2017.

[43] Qibin Zhao, Masashi Sugiyama, Longhao Yuan, and Andrzej Cichocki. Learning efficient tensor representations with ring-structured networks. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8608–8612. IEEE, 2019.

[44] Qibin Zhao, Guoxu Zhou, Shengli Xie, Liqing Zhang, and Andrzej Cichocki. Tensor ring decomposition. *arXiv preprint arXiv:1606.05535*, 2016.