Tensor-based two-layer decoupling of multivariate polynomial maps

Konstantin Usevich, Yassine Zniyed, Mariya Ishteva, Philippe Dreesen, André L. F. de Almeida

***Abstract*—In this paper, we introduce a new decomposition of multivariate maps that generalizes the decoupling problem recently proposed in the system identification community. In the context of neural networks, this decomposition can be seen as a two-layer feedforward network with flexible activation functions. We show that for such maps the Jacobian and Hessian tensors admit ParaTuck and CP decompositions respectively. We propose a methodology to perform the two-layer decoupling of the given polynomial maps based on joint ParaTuck and CP decomposition, by combining first and second-order information.**

***Index Terms*—tensor decomposition, polynomial decoupling, ParaTuck, neural networks, coupled decompositions.**

I. Introduction

The problem of learning to imitate and approximate complex nonlinear functions is crucial for solving many scientific challenges, including nonlinear system identification [1] and neural network learning [2]. The decoupling problem formulated in [3] and motivated by system identification problems, aims at decomposing a multivariate map as linear combinations of univariate functions in linear forms of the input variables. From the neural network point of view, the decoupling model of [3] corresponds to the usage of trainable (flexible) activation functions, see e.g., [4], [5]. Flexible activation functions attracted recent interest in the machine learning community since they can improve the expressive power of neural networks (compared to fixed activation functions).

Several approaches relying on linear and multilinear algebra [3], [6], [7] have been proposed to find decoupled representations. The most practically relevant approach of [3] relies on the canonical polyadic decomposition (CP decomposition or CPD) [8]–[10] of a third-order tensor constructed from stacking evaluations at different points of the Jacobian matrix of the function. It proved to be useful in many tasks in block-structured nonlinear dynamical system identification [1], [11]. While formulated for the decoupling of polynomial maps, the approach of [3] can be also adapted to a wider class of differentiable functions [12]. However, the main drawback of the decoupling approach of [3] is that it applies only to a single hidden nonlinear layer.

Konstantin Usevich, Université de Lorraine, CNRS, Nancy, France, e-mail: konstantin.usevich@cnrs.fr; Yassine Zniyed, Univ. de Toulon, Aix Marseille Univ., CNRS, Toulon, France, e-mail: zniyed@univ-tln.fr; Mariya Ishteva, KU Leuven, Leuven, Belgium, e-mail: mariya.ishteva@kuleuven.be; Philippe Dreesen, Maastricht University (DACS), Maastricht, Netherlands, e-mail: philippe.dreesen@gmail.com; André L. F. de Almeida, Federal University of Ceara, Fortaleza, Brazil, e-mail: andre@gtel.ufc.br.

This research was partially supported by the ANR (Agence Nationale de Recherche) grant LeaFleT (ANR-19-CE23-0021). André L. F. de Almeida acknowledges CNPq for its financial support under grant 312491/2020-4.

In this paper, we introduce a novel decoupled representation that includes two hidden layers. For the proposed new representation, we show that the Jacobian tensor follows a ParaTuck decomposition (PTD) [13]–[15], and that the Hessian of the multivariate map at a single point follows a CPD. Using these results, we provide an algorithm that is based on a coupled factorization of Jacobian and Hessian tensors, which allows for retrieval of the two-layer decoupled representation (*i.e.,* the weights and the flexible activation functions in the context of neural networks) in the polynomial case. Proofs and details omitted in this article can be found in the extended version of the paper [16].

Related work. In the machine learning literature, tensor decompositions of tensors of higher-order derivatives have been already used to obtain guarantees for recovery of weights [17]. (This idea, in fact, goes back to earlier works in blind source separation [18].) However, most of these results apply to the case of a single hidden nonlinear layer and fixed activation functions. The authors are aware of only one work [19] that treats two-layer architecture, however, that work concerns single-output maps, fixed activation functions with biases, and also relies on matrix methods (singular value decomposition), rather than tensor decompositions.

II. Notation and Background

The symbols $(\cdot)^\dagger$ and $\mathrm{rank}(\cdot)$ denote, respectively, the pseudo-inverse and the rank of a matrix. The outer (tensor), Hadamard and Khatri-Rao products are denoted by $\otimes$, $\boxdot$, and $\odot$, respectively. Tensors are represented by bold calligraphic capital letters, e.g., $\boldsymbol{\mathcal{X}}$. For an $n_1 \times n_2 \times n_3$ tensor $\boldsymbol{\mathcal{X}}$, the i-th horizontal, j-th lateral and k-th frontal slices are denoted $\boldsymbol{\mathcal{X}}_{i,:,:}$, $\boldsymbol{\mathcal{X}}_{:,j,:}$ and $\boldsymbol{\mathcal{X}}_{:,:,k}$, and are of sizes $n_2 \times n_3$, $n_1 \times n_3$ and $n_1 \times n_2$, respectively. The (Frobenius) norm of a tensor $\boldsymbol{\mathcal{X}}$ is the square root of the sum of the squares of all its elements, *i.e.,* $||\boldsymbol{\mathcal{X}}|| = \sqrt{\sum_{i,j,k=1}^{n_1,n_2,n_3} \mathcal{X}_{i,j,k}^2}$. The contraction on the kth index of a tensor is denoted as $\underset{k}{\bullet}$ [20], while $\mathrm{diag}(\cdot)$ either forms a diagonal matrix from its vector argument or captures the diagonal of its matrix argument. $\mathrm{unfold}_k \boldsymbol{\mathcal{X}}$ refers to the unfolding (flattening) of tensor $\boldsymbol{\mathcal{X}}$ over its k-th mode [21].

A. CPD and matrix diagonalization

The CP decomposition (CPD) is a decomposition of a tensor $\boldsymbol{\mathcal{X}}$ of size $n_1 \times n_2 \times n_3$ into a sum of r rank-1 tensors,

$$\boldsymbol{\mathcal{X}} = [\![\lambda;\, \mathbf{A}, \mathbf{B}, \mathbf{C}]\!] \overset{\text{def}}{=} \sum_{k=1}^{r} \lambda_k \mathbf{a}_k \otimes \mathbf{b}_k \otimes \mathbf{c}_k, \qquad (1)$$

with $\lambda \in \mathbb{R}^r$ and the factor matrices $\mathbf{A} \in \mathbb{R}^{n_1 \times r}, \mathbf{B} \in \mathbb{R}^{n_2 \times r}, \mathbf{C} \in \mathbb{R}^{n_3 \times r}$ given by $\mathbf{A} = \begin{bmatrix}\mathbf{a}_1 \cdots \mathbf{a}_r\end{bmatrix}$, $\mathbf{B} = \begin{bmatrix}\mathbf{b}_1 \cdots \mathbf{b}_r\end{bmatrix}$, $\mathbf{C} = \begin{bmatrix}\mathbf{c}_1 \cdots \mathbf{c}_r\end{bmatrix}$. For $\lambda = \mathbf{1}_r \overset{\text{def}}{=} \begin{bmatrix}1 & \cdots & 1\end{bmatrix}^\mathsf{T}$ in (1), we also use a shorthand notation

$$[\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!] \overset{\text{def}}{=} [\![\mathbf{1}_r; \mathbf{A}, \mathbf{B}, \mathbf{C}]\!].$$

The CPD (1) of $\boldsymbol{\mathcal{X}}$ can be also viewed as joint low-rank decomposition (joint diagonalization) [22] of its frontal slices:

$$\boldsymbol{\mathcal{X}}_{:,:,k} = \mathbf{A}\,\mathrm{diag}(\mathbf{C}_{k,:})\mathbf{B}^\mathsf{T}. \tag{2}$$

B. ParaTuck decomposition

The ParaTuck decomposition (PTD) of rank (r, s) of an $n_1 \times n_2 \times n_3$ tensor $\boldsymbol{\mathcal{X}}$ is defined through its frontal slices:

$$\boldsymbol{\mathcal{X}}_{:,:,k} = \mathbf{W}\,\mathrm{diag}(\mathbf{g}_k)\mathbf{F}\,\mathrm{diag}(\mathbf{h}_k)\mathbf{U}^\mathsf{T}, \tag{3}$$

where the five factor matrices are $\mathbf{W} \in \mathbb{R}^{n_1 \times r}$, $\mathbf{U} \in \mathbb{R}^{n_2 \times s}$, $\mathbf{F} \in \mathbb{R}^{r \times s}$, $\mathbf{G} = \begin{bmatrix}\mathbf{g}_1 & \cdots & \mathbf{g}_{n_3}\end{bmatrix} \in \mathbb{R}^{r \times n_3}$, and $\mathbf{H} = \begin{bmatrix}\mathbf{h}_1 & \cdots & \mathbf{h}_{n_3}\end{bmatrix} \in \mathbb{R}^{s \times n_3}$. Thus the PTD (3) can be seen as a two-layer generalization of the CPD (2).

PTD has been proposed in psychometrics literature [13] in 1994 but was not widely used due to a lack of reliable algorithms [15]. It has been exploited in wireless communication problems [23], [24], [25] mostly assuming prior knowledge on some factor matrices.

Alternatively, PTD can be defined in the Tucker [26] form:

$$\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{C}} \underset{1}{\bullet} \mathbf{W} \underset{2}{\bullet} \mathbf{U},$$

where the *ParaTuck core tensor* $\boldsymbol{\mathcal{C}} \in \mathbb{R}^{r \times s \times n_3}$ has structure

$$\mathcal{C}_{ijk} = F_{ij} G_{ik} H_{jk}. \tag{4}$$

If the outer factor matrices $\mathbf{W}$ and $\mathbf{U}$ are known, the remaining factors can be easily recovered from (4) by elementwise division [16]. However, if all factor matrices are unknown, to the best of the authors' knowledge, there are no reliable algorithms that can find the PTD, except for the case $(r, s) = (2, 2)$ [27]. For example, an alternating least squares algorithm, introduced in [15], has been shown to have convergence issues. This is why in many use cases of PTD [24], some of the factor matrices are assumed to be known.

C. ParaTuck ambiguities and uniqueness

Similarly to the CPD, the PTD possesses trivial ambiguities (due to permutations and rescaling), *i.e.,* multiple decompositions may exist for the same tensor. It has been shown in [14], that a third-order tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ with PTD (3), admits a family of alternative PTDs

$$\boldsymbol{\mathcal{X}}_{:,:,k} = \tilde{\mathbf{W}}\,\mathrm{diag}(\tilde{\mathbf{g}}_k)\tilde{\mathbf{F}}\,\mathrm{diag}(\tilde{\mathbf{h}}_k)\tilde{\mathbf{U}}^\mathsf{T}, \tag{5}$$

with the factor matrices given as

$$\begin{aligned}
&\mathbf{W} = \tilde{\mathbf{W}} \cdot (\boldsymbol{\Pi}_\mathbf{W} \cdot \boldsymbol{\Lambda}_\mathbf{W}), \quad \mathbf{U} = \tilde{\mathbf{U}} \cdot (\boldsymbol{\Pi}_\mathbf{U} \cdot \boldsymbol{\Lambda}_\mathbf{U}),\\
&\mathbf{F} = (\bar{\boldsymbol{\Lambda}}_\mathbf{W} \cdot \boldsymbol{\Lambda}_\mathbf{W}^{-1} \cdot \boldsymbol{\Pi}_\mathbf{W}^\mathsf{T}) \cdot \tilde{\mathbf{F}} \cdot (\boldsymbol{\Pi}_\mathbf{U} \cdot \boldsymbol{\Lambda}_\mathbf{U}^{-1} \cdot \bar{\boldsymbol{\Lambda}}_\mathbf{U}),\\
&\mathbf{g}_k = (\alpha_k \cdot \bar{\boldsymbol{\Lambda}}_\mathbf{W}^{-1} \boldsymbol{\Pi}_\mathbf{W}^\mathsf{T}) \cdot \tilde{\mathbf{g}}_k, \quad \mathbf{h}_k = (\alpha_k^{-1} \cdot \bar{\boldsymbol{\Lambda}}_\mathbf{U}^{-1} \boldsymbol{\Pi}_\mathbf{U}^\mathsf{T}) \cdot \tilde{\mathbf{h}}_k,
\end{aligned}$$

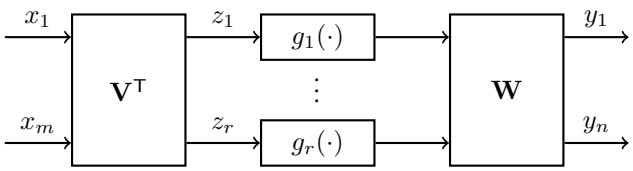

Fig. 1. Decoupled representation of $\mathbf{y} = \mathbf{f}(\mathbf{x})$ into a single-layer model as in (6). This naturally leads to the CPD of the corresponding Jacobian tensor.

where $\boldsymbol{\Lambda}_\mathbf{W}$, $\boldsymbol{\Lambda}_\mathbf{U}$, $\bar{\boldsymbol{\Lambda}}_\mathbf{W}$ and $\bar{\boldsymbol{\Lambda}}_\mathbf{U}$ are diagonal matrices, $\boldsymbol{\Pi}_\mathbf{W}$ and $\boldsymbol{\Pi}_\mathbf{U}$ are permutation matrices, and α_k are nonzero scalars.

Similar to the CPD, the PTD is called essentially unique if it is unique up to trivial ambiguities; essential uniqueness can happen under mild conditions [14]. One of the main distinctions between the CPD and PTD ambiguities lies in the slice-wise ambiguities (coefficients α_k), which should be handled with care in decoupling problems, as shown later.

III. Decoupling polynomial functions

The problem of decoupling refers to the representation of a multivariate polynomial function as a linear combination of univariate polynomials in terms of the input variables (Fig. 1). In this paper, we take the classical decoupling problem one step further and generalize the representation to a two-layer model, in order to enhance its versatility and expressive power.

A. Reminder: one-layer structure

Let $\mathbf{f} : \mathbb{R}^m \to \mathbb{R}^n$ be a multivariate map $\mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}) \cdots f_n(\mathbf{x})]^\mathsf{T}$, with $\mathbf{x} = [x_1 \cdots x_m]^\mathsf{T}$. It is said that $\mathbf{f}$ has a decoupled representation (Fig. 1), if

$$\mathbf{f}(\mathbf{x}) = \mathbf{W}\mathbf{g}(\mathbf{V}^\mathsf{T}\mathbf{x}), \tag{6}$$

where $\mathbf{W} \in \mathbb{R}^{n \times r}$, $\mathbf{V} \in \mathbb{R}^{m \times r}$ are transformation matrices, $\mathbf{w}_k$ and $\mathbf{v}_k$ are respectively their columns, and $\mathbf{g} : \mathbb{R}^r \to \mathbb{R}^r$ follows $\mathbf{g}(z_1, \cdots, z_r) = \begin{bmatrix}g_1(z_1) & \cdots & g_r(z_r)\end{bmatrix}^\mathsf{T}$, with $g_k : \mathbb{R} \to \mathbb{R}$ univariate differentiable functions. From a neural network point of view, (6) can be viewed as a model with one nonlinear layer composed of flexible activation functions [5] sandwiched between nonlinear layers.

It was shown in [3] that the Jacobian of a function of form (6) can be factorized as $\mathbf{J}_\mathbf{f}(\mathbf{x}) = \mathbf{W}\mathbf{D}(\mathbf{x})\mathbf{V}^\mathsf{T}$ where $\mathbf{D}(\mathbf{x})$ is a diagonal matrix that depends on $\mathbf{x}$. The idea of [3] was to stack evaluations of Jacobians in different points $\mathbf{x}^{(p)}$, for $p = 1, \ldots, P$ into $n \times m \times P$ tensor $\boldsymbol{\mathcal{J}}$, and use the connection to joint matrix factorization (2) of its frontal slices. Thus decoupling a given multivariate function from first order information can be achieved using the CPD of the Jacobian tensor. Initially proposed for exact decoupling of polynomial maps [3], the approach is applicable for a wider range of scenarios, and found many applications, in particular, in block-structured system identification.

However, the approach [3] can only handle representations with a single nonlinear layer, which limits its applicability and expressive power. We are not aware of existing tensor-based decoupling approaches for the case of several hidden layers.

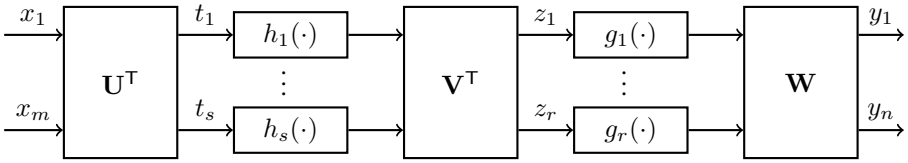

Fig. 2. Decoupled representation of $\mathbf{y} = \mathbf{f}(\mathbf{x})$ into a two-layer model, as in (7). This naturally leads to a PTD of the corresponding Jacobian tensor.

B. Proposed two-layer structure

We propose to extend the decoupled representation in (6) to a representation with two layers as follows (Fig. 2):

$$\mathbf{f}(\mathbf{x}) = \mathbf{W}\mathbf{g}\big(\mathbf{V}^\mathsf{T} \cdot \mathbf{h}(\mathbf{U}^\mathsf{T}\mathbf{x})\big), \tag{7}$$

where $\mathbf{W} \in \mathbb{R}^{n\times r}$, $\mathbf{V} \in \mathbb{R}^{s\times r}$, $\mathbf{U} \in \mathbb{R}^{m\times s}$, are transformation matrices, and $\mathbf{h} : \mathbb{R}^s \to \mathbb{R}^s$ and $\mathbf{g} : \mathbb{R}^r \to \mathbb{R}^r$ follow $\mathbf{g}(z_1, \cdots, z_r) = [g_1(z_1) \cdots g_r(z_r)]^\mathsf{T}$ and $\mathbf{h}(t_1, \cdots, t_s) = [h_1(t_1) \cdots h_s(t_s)]^\mathsf{T}$, respectively, with differentiable h_k and g_k. This two-layer generalization allows having more flexibility in the decoupling of multivariate nonlinear functions. As we will show next, it is intricately connected to the ParaTuck and CP decompositions when considering first- and second-order information.

IV. Tensor-based function decomposition

A. Jacobian and ParaTuck decomposition

The main idea to find the decomposition (7) of a nonlinear function $\mathbf{f}$ relies on the evaluation of the Jacobian matrix in different points $\mathbf{x}^{(p)}$, for $p = 1, \ldots, P$. This idea mirrors [3], where it has been applied to the classical decoupling model. In the sequel, we will replicate the procedure with the new proposed structure in (7) and will derive the new expression of the Jacobian tensor.

Lemma 1: The first-order derivatives of the parameterization (7) are given by

$$\begin{aligned}\mathbf{J}_\mathbf{f}(\mathbf{x}) :&= \begin{bmatrix} \frac{\partial f_1}{\partial x_1}(\mathbf{x}) & \cdots & \frac{\partial f_1}{\partial x_m}(\mathbf{x}) \\ \vdots & & \vdots \\ \frac{\partial f_n}{\partial x_1}(\mathbf{x}) & \cdots & \frac{\partial f_n}{\partial x_m}(\mathbf{x}) \end{bmatrix} \qquad (8)\\ &= \mathbf{W} \cdot \operatorname{diag}\Big(\big[g_1'(z_1) \cdots g_r'(z_r)\big]\Big) \cdot \mathbf{V}^\mathsf{T} \\ &\quad \cdot \operatorname{diag}\Big(\big[h_1'(t_1) \cdots h_s'(t_s)\big]\Big) \cdot \mathbf{U}^\mathsf{T}. \qquad (9)\end{aligned}$$

Proof: Proof follows by applying the chain rule to (7). □

Based on Lemma 1, we can see that Jacobian of (7) evaluated at the points $\mathbf{x}^{(p)}$ follows

$$\mathbf{J}_\mathbf{f}(\mathbf{x}^{(p)}) = \mathbf{W} \cdot \operatorname{diag}\big(\mathbf{g}_p\big) \cdot \mathbf{V}^\mathsf{T} \cdot \operatorname{diag}\big(\mathbf{h}_p\big) \cdot \mathbf{U}^\mathsf{T}, \tag{10}$$

where the vectors $\mathbf{g}_p \in \mathbb{R}^r$ and $\mathbf{h}_p \in \mathbb{R}^s$ are given by

$$\mathbf{g}_p = \mathbf{g}'(\mathbf{z}^{(p)}) = \big[g_1'(z_1^{(p)}) \cdots g_r'(z_r^{(p)})\big]^\mathsf{T}, \tag{11}$$

$$\mathbf{h}_p = \mathbf{h}'(\mathbf{t}^{(p)}) = \big[h_1'(t_1^{(p)}) \cdots h_s'(t_s^{(p)})\big]^\mathsf{T}, \tag{12}$$

with $\mathbf{t}^{(p)} = \big[t_1^{(p)} \cdots t_s^{(p)}\big]^\mathsf{T} = \mathbf{U}^\mathsf{T}\mathbf{x}^{(p)}$ and $\mathbf{z}^{(p)} = \big[z_1^{(p)} \cdots z_r^{(p)}\big]^\mathsf{T} = \mathbf{V}^\mathsf{T}\mathbf{h}(\mathbf{U}^\mathsf{T}\mathbf{x}^{(p)})$.

We can then define the matrices $\mathbf{H}$ and $\mathbf{G}$ as

$\mathbf{G} = \begin{bmatrix}\mathbf{g}_1 & \cdots & \mathbf{g}_P\end{bmatrix} \in \mathbb{R}^{r\times P}$, $\mathbf{H} = \begin{bmatrix}\mathbf{h}_1 & \cdots & \mathbf{h}_P\end{bmatrix} \in \mathbb{R}^{s\times P}$.

Thus Lemma 1 shows that the expression of the Jacobian tensor for the two-layer model (7) corresponds to the frontal slices of a PTD (3) of rank (r, s). Indeed, a Jacobian tensor $\boldsymbol{\mathcal{J}} \in \mathbb{R}^{n\times m\times P}$ constructed by stacking the Jacobian evaluations at sampling points $\mathbf{x}^{(p)} \in \mathbb{R}^m$, for $p = 1, \ldots, P$:

$$\boldsymbol{\mathcal{J}}_{:,:,p} = \mathbf{J}_\mathbf{f}(\mathbf{x}^{(p)}), \tag{13}$$

admits a PTD (3) with factors $\mathbf{U}$, $\mathbf{F} = \mathbf{V}^\mathsf{T}$, $\mathbf{W}$, $\mathbf{G}$ and $\mathbf{H}$ (where $\mathbf{U}$, $\mathbf{V}$ and $\mathbf{W}$ do not depend on p).

B. Second-order information and structured CPD

To improve the usefulness of the PT formulation, we will examine the second-order information of (7), and show that it is very helpful for the decoupling problem since the PTD lacks reliable algorithms for the moment. In this subsection, we derive an expression for the Hessians at each point. The Hessian tensor $\boldsymbol{\mathcal{H}}(\mathbf{x}) \in \mathbb{R}^{n\times m\times m}$ at a point $\mathbf{x}$ is defined as

$$\mathcal{H}_{ijk}(\mathbf{x}) = \frac{\partial^2 f_i}{\partial x_j \partial x_k}(\mathbf{x}).$$

We will show the $\boldsymbol{\mathcal{H}}(\mathbf{x})$ admits a CPD of a special form. For that, we introduce the following notation for the factors of the Jacobians in (9):

$$\mathbf{A}(\mathbf{x}) = \mathbf{W} \cdot \operatorname{diag}\Big(\mathbf{g}'(\mathbf{z}(\mathbf{x}))\Big) \cdot \mathbf{V}^\mathsf{T} \in \mathbb{R}^{n\times s}, \tag{14}$$

$$\mathbf{B}(\mathbf{x}) = \mathbf{V}^\mathsf{T} \cdot \operatorname{diag}\Big(\mathbf{h}'(\mathbf{t}(\mathbf{x}))\Big) \cdot \mathbf{U}^\mathsf{T} \in \mathbb{R}^{r\times m}, \tag{15}$$

where

$$\mathbf{t}(\mathbf{x}) = \mathbf{U}^\mathsf{T}\mathbf{x} \text{ and } \mathbf{z}(\mathbf{x}) = \mathbf{V}^\mathsf{T}\mathbf{h}(\mathbf{U}^\mathsf{T}\mathbf{x}),$$

so that we can rewrite (9) as

$$\begin{aligned}\mathbf{J}_\mathbf{f}(\mathbf{x}) &= \mathbf{A}(\mathbf{x}) \operatorname{diag}\Big(\mathbf{h}'(\mathbf{t}(\mathbf{x}))\Big)\mathbf{U}^\mathsf{T} \\ &= \mathbf{W} \operatorname{diag}\Big(\mathbf{g}'(\mathbf{z}(\mathbf{x}))\Big)\mathbf{B}(\mathbf{x}).\end{aligned} \tag{16}$$

Armed with this notation, we are ready to formulate the following result on the structure of the Hessian tensor.

Lemma 2: The Hessian tensor has the following $(r+s)$-term CPD:

$$\begin{aligned}\boldsymbol{\mathcal{H}}(\mathbf{x}) = &\llbracket \mathbf{g}''(\mathbf{z}(\mathbf{x})); \mathbf{W}, \mathbf{B}^\mathsf{T}(\mathbf{x}), \mathbf{B}^\mathsf{T}(\mathbf{x}) \rrbracket \\ &+ \llbracket \mathbf{h}''(\mathbf{t}(\mathbf{x})); \mathbf{A}(\mathbf{x}), \mathbf{U}, \mathbf{U} \rrbracket.\end{aligned} \tag{17}$$

Proof: Follows by applying the Leibniz rule to (16). □

It is worth noting that (i) the Hessian tensor is partially symmetric (*i.e.,* $\mathcal{H}_{ijk} = \mathcal{H}_{ikj}$), thus the factors in (17) cannot be recovered from the CPD due to loss of uniqueness.

V. Constrained coupled decomposition approach

We propose to rephrase the new decoupling problem as a constrained coupled tensor decomposition, using both the first and second-order information. Before that, we specify the assumptions considered in our approach:

1) $m \geq s$ and $n \geq r \geq s$,
2) $\mathbf{W}$ is known and has full column rank r,
3) $\mathbf{U}$ and $\text{unfold}_2\mathcal{J}$ have rank s.

Under the conditions above, we can always reduce the problem to the case

$$r = n, \quad s = m, \quad \text{and} \quad \mathbf{W} = \mathbf{I}_r. \tag{18}$$

It is important to mention that despite these simplifying assumptions, the PTD remains a challenging problem, for example, for ALS-type algorithms [15].

A. Reformulation as a constrained CPD

We assume that we are given Jacobians and Hessians of $\mathbf{f}$ at P evaluation points, and that $\mathbf{f}$ satisfies the assumption (18). Additionally, we impose that the elements of $\mathbf{G}$ are nonzero. Then the following proposition can be proved.

Proposition 1: Let us take all the Hessians at P points and stack them into a third-order $Pn \times m \times m$ tensor $\mathcal{T}^{hess}$ as

$$(\mathcal{T}^{hess})_{1+(p-1)n:pn,:,:} = \mathcal{H}(\mathbf{x}^{(p)}). \tag{19}$$

We also stack Jacobians in one matrix $\mathbf{J}^{all} \in \mathbb{R}^{Pn \times m}$ as

$$\mathbf{J}^{all}_{1+(p-1)n:pn,:} = \mathbf{J}_{\mathbf{f}}(\mathbf{x}^{(p)}). \tag{20}$$

Then, $\mathcal{T}^{hess}$ has the following CPD with structured factors:

$$\mathcal{T}^{hess} = [\![\text{diag}(\mathbf{c}), (\mathbf{J}^{all})^{\mathsf{T}}, (\mathbf{J}^{all})^{\mathsf{T}}]\!] + [\![\mathbf{E}, \mathbf{U}, \mathbf{U}]\!], \tag{21}$$

where $\mathbf{E} \in \mathbb{R}^{Pn \times m}$ and $\mathbf{c} \in \mathbb{R}^{Pn}$.

Proof: The proof of Proposition 1 as well as the expression for $\mathbf{E}$ and $\mathbf{c}$ can be found in [16]. □

B. Reformulation as structured low-rank matrix completion

In order to find the structured CPD of the tensor $\mathcal{T}^{hess}$, we reformulate the problem as low-rank matrix recovery. To do so, consider the unfolding

$$\mathbf{T}^{hess} = (\text{unfold}_3(\mathcal{T}^{hess}))^{\mathsf{T}} \in \mathbb{R}^{m^2 \times nP},$$

which admits the factorization

$$\mathbf{T}^{hess} = \left((\mathbf{J}^{all})^{\mathsf{T}} \odot (\mathbf{J}^{all})^{\mathsf{T}}\right) \text{diag}(\mathbf{c}) + (\mathbf{U} \odot \mathbf{U})\mathbf{E}^{\mathsf{T}}.$$

Assuming the exact model (21), we just need to find vector $\mathbf{c} \in \mathbb{R}^{Pn}$ so that the following matrix has rank s

$$\mathscr{S}(\mathbf{c}) = \mathbf{T}^{hess} - \left((\mathbf{J}^{all})^{\mathsf{T}} \odot (\mathbf{J}^{all})^{\mathsf{T}}\right) \text{diag}(\mathbf{c}).$$

We pose this problem as rank minimization of $\mathscr{S}(\mathbf{c})$, which can be solved as the minimization over the low-rank manifold:

$$\min_{\mathbf{P},\mathbf{L}} \|\Pi_{\mathscr{S}}(\mathbf{PL} - \mathbf{T}^{hess})\|_F, \tag{22}$$

where $\Pi_{\mathscr{S}}$ is the projection on the set of structured matrices, see e.g., [28] for more details on the reformulation (22). We summarize our approach in the following algorithm.

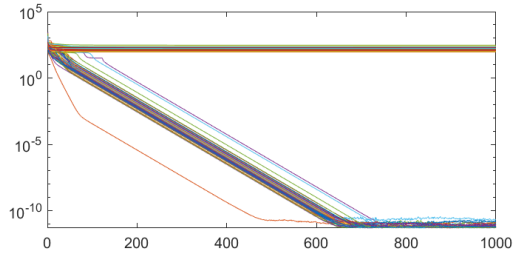

Fig. 3. The convergence plot shows a linear convergence of the cost function in (22) to an error of order 10^{-10} in 46 out of 100 random initializations.

Algorithm 1: Two-layer decoupling using PTD

assume eqn. (18)
input: Jacobian evaluations $\mathbf{J}_f(\mathbf{x}^{(1)}), \ldots, \mathbf{J}_f(\mathbf{x}^{(P)})$
Hessian evaluations $\mathcal{T}(\mathbf{x}^{(1)}), \ldots, \mathcal{T}(\mathbf{x}^{(P)})$
output: factors $\mathbf{U}$, $\mathbf{V}$; coefficients of $\mathbf{g}$, $\mathbf{h}$

1) stack the Jacobians into matrix $\mathbf{J}^{all}$, see (20)
2) stack the Hessians into tensor $\mathcal{T}^{hess}$, see (19)
3) find $\mathbf{PL}$ from the rank minimization problem (22)
4) reshape $\mathbf{P}$ into a $m \times m \times m$ tensor
5) compute its rank-m CPD $[\![\mathbf{U}, \mathbf{U}, \mathbf{Q}]\!]$ and extract $\mathbf{U}$
6) find the ParaTuck core tensor $\mathcal{C} = \mathcal{J} \bullet_2 \mathbf{U}^{\dagger}$
7) from $\mathcal{C}$ and (4), recover $\mathbf{V}$, $\mathbf{G}$, and $\mathbf{H}$
8) fix α_k ambiguities (see the next section)

VI. Numerical examples

A. Example of decoupling with $r = s = 3$

We consider a concrete example of decoupling for polynomial functions that are shifts of the same function

$$h_1(t) = \phi(t - 0.5), \quad h_2(t) = \phi(t + 0.2), \quad h_3(t) = \phi(t)$$
$$g_1(t) = \phi(t + 0.1), \quad g_2(t) = \phi(t + 0.4), \quad g_3(t) = \phi(t)$$

where $\phi(t) = t^2 - 0.25t^4 + t^3 - 3t$. In our experiment, the matrices $\mathbf{V}$ and $\mathbf{U}$ were generated randomly, with i.i.d. elements from the uniform distribution on $[-1; 1]$. The $P = 100$ sampling points $\mathbf{x}^{(p)}$ are drawn uniformly from $[-0.5; 0.5]$.

In Fig. 3, we show the cost function (22) as a function of iteration (out of maximum 1000 iterations). We run 100 random initializations of $\mathbf{P}$ as $\mathbf{P}_0 = \mathbf{U}_0 \odot \mathbf{U}_0$ with $\mathbf{U}_0$ drawn from standard Gaussian i.i.d. distribution. We observe that in 46 cases out of 100, the algorithm shows linear convergence to an error of order 10^{-10}. Taking one of the runs with the best cost function value, we are able to recover the original $\mathbf{U}$ and the nonlinearities as explained in the next subsection.

B. Ambiguities in the problem and recovering the functions

The problem which remains for the decoupling approach is the reconstruction of functions. The issue is that the approach suggested in [3] (regression of $\mathbf{H}_{k,:}$ versus $(t_k^{(1)}, \ldots, t_k^{(P)})$) does not directly work, due to the presence of the ambiguities α_p for each of the columns of $\mathbf{G}$. To recover the functions $h_k(\cdot)$, we need more assumptions, for example, impose that

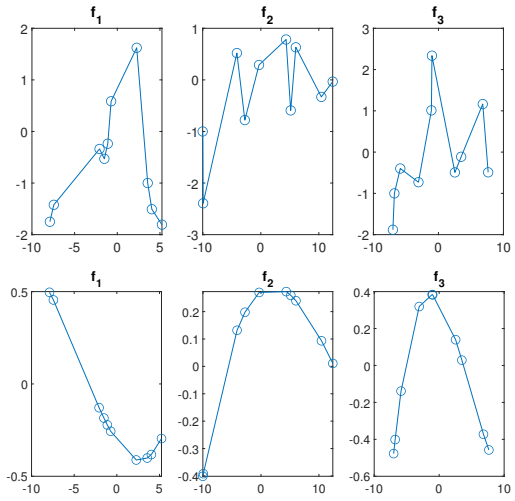

Fig. 4. Plot of $\mathbf{H}_{k,:}$ versus $(t_k^{(1)}, \ldots, t_k^{(P)})$ for $k = 1, \ldots, 3$ before (top) and after (bottom) slice ambiguity correction (step 8 of Algorithm 1).

the functions are polynomials of low order. To estimate the ambiguities, we solve the following system of equations

$$\mathbf{a}_k^{\mathsf{T}} \mathbf{X}(\mathbf{t}, d) \approx \mathbf{H}_{k,:} \operatorname{diag}(\boldsymbol{\alpha}), \quad k = 1, \ldots, s,$$

where d is the polynomial degree, $\mathbf{X}(\mathbf{t}, d)$ is the Vandermonde matrix (for points t and up to degree d), and we solve for $\mathbf{a}_k^{\mathsf{T}}$ (coefficients of polynomials) and $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_P)$ (vector of scalings). This is a problem of intersection of two linear subspaces and can be solved with alternating projections.

As shown in Fig. 4, the proposed estimation of slice ambiguities α_p, helps to recover the functions $\mathbf{h}$ (and $\mathbf{g}$), which is otherwise not possible to estimate from the factors of the PTD without additional assumptions. Note that the proposed correction is not limited to polynomials, but can be applied to other bases of functions (see, *e.g.*, [5]).

VII. Conclusion and Outlook

We presented a new method for multivariate function approximation that couples the PT and CP decompositions. Our approach utilizes both first and second-order information of the original function and has been shown to be effective through numerical simulations on a simple synthetic example. Although the PT decomposition remains a challenging problem, our results demonstrate the potential of the proposed method for addressing this issue and provide a promising direction for future work in the field of multivariate function approximation.

References

[1] J. Schoukens and L. Ljung, "Nonlinear system identification: A user-oriented road map," *IEEE Control Systems Magazine*, vol. 39, no. 6, pp. 28–99, 2019.

[2] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[3] P. Dreesen, M. Ishteva, and J. Schoukens, "Decoupling multivariate polynomials using first-order information and tensor decompositions," *SIAM Journal on Matrix Analysis and Applications*, vol. 36, no. 2, pp. 864–879, 2015.

[4] A. Apicella, F. Donnarumma, F. Isgrò, and R. Prevete, "A survey on modern trainable activation functions," *Neural Networks*, vol. 138, pp. 14–32, 2021.

[5] Y. Zniyed, K. Usevich, S. Miron, and D. Brie, "Tensor-based approach for training flexible neural networks," in *55th Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, USA, Oct. 2021.

[6] M. Schoukens and Y. Rolain, "Cross-term elimination in parallel Wiener systems using a linear input transformation," *IEEE Transactions on Instrumentation and Measurement*, vol. 61, pp. 845–847, 2012.

[7] A. Van Mulders, L. Vanbeylen, and K. Usevich, "Identification of a block-structured model with several sources of nonlinearity," in *European Control Conference (ECC)*, Strasbourg, France, 2014.

[8] F. L. Hitchcock, "Multiple invariants and generalized rank of a p-way matrix or tensor," *Journal of Mathematics and Physics*, vol. 7, pp. 39–79, 1927.

[9] R. A. Harshman, "Foundations of the PARAFAC procedure: Models and conditions for an 'explanatory' multi-modal factor analysis," *UCLA Working Papers in Phonetics*, vol. 16, pp. 1–84, 1970.

[10] R. Bro, "PARAFAC. tutorial and applications," *Chemometrics and Intelligent Laboratory Systems*, vol. 38, no. 2, pp. 149–171, 1997.

[11] G. Hollander, "Multivariate polynomial decoupling in nonlinear system identification," Ph.D. dissertation, Vrije Universiteit Brussel, 2018.

[12] J. Decuyper, K. Tiels, S. Weiland, M. C. Runacres, and J. Schoukens, "Decoupling multivariate functions using a nonparametric filtered tensor decomposition," *Mechanical Systems and Signal Processing*, vol. 179, p. 109328, 2022.

[13] R. A. Harshman and M. E. Lundy, "PARAFAC: Parallel factor analysis," *Computational Statistics & Data Analysis*, vol. 18, no. 1, pp. 39–72, 1994.

[14] ——, "Uniqueness proof for a family of models sharing features of Tucker's three-mode factor analysis and PARAFAC/candecomp," *Psychometrika*, vol. 61, no. 1, pp. 133–154, March 1996.

[15] R. Bro, "Multi-way analysis in the food industry — models, algorithms & applications," Ph.D. dissertation, Universiteit van Amsterdam, 1998.

[16] K. Usevich, Y. Zniyed, M. Ishteva, P. Dreesen, and A. de Almeida, "Two-layer decoupling of multivariate polynomials with coupled Paratuck-2 and CP decompositions," 2023, hAL preprint, https://hal.science/hal-03968630.

[17] M. Janzamin, H. Sedghi, and A. Anandkumar, "Beating the perils of non-convexity: Guaranteed training of neural networks using tensor methods," *arXiv preprint arXiv:1506.08473*, 2015.

[18] P. Comon and M. Rajih, "Blind identification of under-determined mixtures based on the characteristic function," *Signal Processing*, vol. 86, no. 9, pp. 2271–2281, 2006.

[19] M. Fornasier, T. Klock, and M. Rauchensteiner, "Robust and resource-efficient identification of two hidden layer neural networks," *Constructive Approximation*, pp. 1–62, 2019.

[20] N. D. Sidiropoulos, L. De Lathauwer, X. Fu, K. Huang, E. E. Papalexakis, and C. Faloutsos, "Tensor decomposition for signal processing and machine learning," *IEEE Transactions on Signal Processing*, vol. 65, no. 13, pp. 3551–3582, 2017.

[21] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM review*, vol. 51, no. 3, pp. 455–500, 2009.

[22] R. André, X. Luciani, and E. Moreau, "Joint eigenvalue decomposition algorithms based on first-order Taylor expansion," *IEEE Transactions on Signal Processing*, vol. 68, pp. 1716–1727, 2020.

[23] A. L. de Almeida, G. Favier, and J. C. Mota, "Space–time spreading–multiplexing for MIMO wireless communication systems using the PARATUCK2 tensor model," *Signal Processing*, vol. 89, no. 11, pp. 2103–2116, 2009.

[24] L. R. Ximenes, G. Favier, A. L. F. de Almeida, and Y. C. B. Silva, "PARAFAC-PARATUCK semi-blind receivers for two-hop cooperative MIMO relay systems," *IEEE Transactions on Signal Processing*, vol. 62, no. 14, pp. 3604–3615, 2014.

[25] P. Marinho R. de Oliveira, C. A. R. Fernandes, G. Favier, and R. Boyer, "PARATUCK semi-blind receivers for relaying multi-hop MIMO systems," *Digital Signal Processing*, vol. 92, pp. 127–138, 2019.

[26] L. R. Tucker, "Some mathematical notes on three-mode factor analysis," *Psychometrika*, vol. 31, no. 3, pp. 279–311, 1966.

[27] K. Usevich, "An algebraic algorithm for rank-2 ParaTuck-2 decomposition," 2023, hAL preprint, https://hal.science/hal-03966869.

[28] M. Ishteva, K. Usevich, and I. Markovsky, "Factorization approach to structured low-rank approximation with applications," *SIAM Journal on Matrix Analysis and Applications*, vol. 35, no. 3, pp. 1180–1204, 2014.