Advances in Applied Mathematics 180 (2026) 103117

Contents lists available at ScienceDirect

Advances in Applied Mathematics

journal homepage: www.elsevier.com/locate/yaama

Product of tensors and description of networks

Luca Chiantini [a], Giuseppe Alessio D'Inverno [b], Sara Marziali [a,*]

[a] *Department of Information Engineering and Mathematics, University of Siena, Via Roma 56, Siena, 58100, Italy*
[b] *International School for Advanced Studies, Via Bonomea 265, Trieste, 34136, Italy*

ARTICLE INFO

Article history:
Received 6 May 2025
Received in revised form 27 April 2026
Accepted 23 May 2026
Available online 1 June 2026

MSC:
15A69

Keywords:
Networks
Tensor algebra
Bhattacharya-Mesner Product

ABSTRACT

We study the characterization of the signal distribution in a network under the lens of tensor algebra. More specifically, we describe every activation function as tensor distributions with respect to the nodes, called *activation tensors*. The distribution of the signal is encoded in the *total tensor* of the network. We formally prove that the total tensor can be obtained by computing the *Bhattacharya-Mesner Product*, an n-ary operation for tensors of order n, on the set of activation tensors properly ordered and processed via two basic operations, that we call *blow* and *forget*. Our theoretical framework can be validated through the related code developed in Python.

The last shall be the first.
Matthew, 16:20

* Corresponding author.
E-mail addresses: luca.chiantini@unisi.it (L. Chiantini), gdinvern@sissa.it (G.A. D'Inverno), sara.marziali@student.unisi.it (S. Marziali).

https://doi.org/10.1016/j.aam.2026.103117

Introduction

In this work, we introduce a description of network behavior based on the Bhattacharya-Mesner Product (BMP) of tensors. The BMP is an n-ary operation on tensors of order n that generalizes the binary matrix multiplication. It was introduced in [10] (see also [11] and [1]) and has been used to describe algebraic and geometric properties of tensors in [4–6,12].

In our setting, a Signal Propagation Network (see Definition 2) is determined by an underlying directed graph, describing the flow of a signal through the set of nodes, and by a list of activation tensors, one for each node. We focus on directed acyclic graphs (DAGs), that is, graphs with no oriented cycles, in order to exclude circular signal transmission. The absence of cycles guarantees the existence of total orderings of the nodes compatible with the partial order induced by the graph. We regard such a total ordering as a constituting aspect of the network. Each activation tensor is responsible for the reaction of the node to the state of its parents.

The dimension of the activation tensor is equal to the number of incoming arrows (*in-degree*) of the node plus one. The size of the activation tensor, in turn, depends on the number of states of each node, which we assume to be constant through the network. More precisely, for a node a whose parents are the nodes $(b_{j_1}, \dots, b_{j_d})$ (in the fixed total order), the entry $A_{i_1,\dots,i_d,i_{d+1}}$ of the activation tensor A represents the probability that a takes the state i_{d+1} when the node b_{j_k} is in state i_k, for $k = 1, \dots, d$.

In our analysis, we transform the collection of activation tensors $\{A_i\}$ into tensors $\{T_{a_i}\}$ of dimension n, where n is the number of nodes of the graph, by means of a procedure based on two operations on the indices called *blow* (or *inflation*, following the terminology of [4]) and *forget* (or *copy*), described in Section 3, Procedure 22.

We observe that these operations are well defined independently of the number of nodes and of the choice of activation tensors, which may be arbitrary and may vary from node to node. Examples of reasonable activation tensors, in the spirit of [2], are given throughout the paper. To better align the model with a statistical interpretation of the network, one could impose stochastic constraints on the activation tensors, for instance, requiring their entries to be positive real numbers in the interval $[0, 1]$ and to sum to 1 over the output state. We do not impose such restrictions here, since they are not needed for our computations.

It turns out that, for a fixed network in which all nodes are observable, the set of total tensors describing the final states of the network forms a toric variety. Since all nodes are observable, the total tensor of the network depends on parameters and fills a toric subvariety of the (projective) space of tensors. The main geometric aspects of this variety are therefore strictly related to the algebraic structure of the BMP, and we will explore the connection, as well as extensions to networks with hidden nodes, via marginalization of the BMP in future investigations.

Our main result (Theorem 23) shows that the final state of the network can be determined by computing the BMP, for a given ordering, of the activation tensors as-

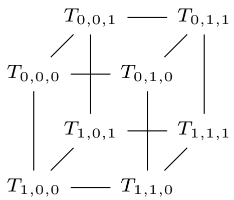


Fig. 1. Labeling convention. The convention used to label the entries of a tensor $T \in \mathbb{C}^2 \otimes \mathbb{C}^2 \otimes \mathbb{C}^2$.

sociated with the nodes suitably modified by the blow and forget operations introduced above.

The paper is organized as follows. In Section 1, we introduce SPNs and present the notation and basic concepts on tensors, with particular emphasis on activation tensors. In Section 2, we define the BMP of tensors and discuss its algebraic properties. Finally, in Section 3, we show how a network can be characterized in terms of the BMP.

1. Basic notation and concepts

1.1. Basic notation on tensors

In this section, we introduce the main notation for tensors and networks. Standard references for tensor theory are [7,8].

Definition 1. Let $V_1, \ldots, V_d$ be complex vector spaces of dimensions $v_1, \ldots, v_d$, respectively. A function

$$f : V_1 \times \cdots \times V_d \to \mathbb{C}$$

is *multilinear* if it is linear in each factor V_l, $l = 1, \ldots, d$. The space of such multilinear functions is denoted by $V_1^* \otimes V_2^* \otimes \cdots \otimes V_d^*$ and it is referred to as *tensor product* of $V_1^*, \ldots, V_d^*$. Elements

$$T \in V_1^* \otimes V_2^* \otimes \cdots \otimes V_d^*$$

are called *tensors*. The integer d is sometimes named the *dimension* of T, while the expression $v_1 \times \cdots \times v_d$ is the *size* of T. We say a tensor is *cubical* of size n if $v_1 = \cdots = v_d = n$.

Our labeling convention for a tensor $T \in \mathbb{C}^2 \otimes \mathbb{C}^2 \otimes \mathbb{C}^2$ is shown in Fig. 1.

1.2. Networks

A basic reference for networks and activation tensors is [9]. Following the notation proposed in [3], let G be a directed acyclic graph (DAG) with $V(G)$ the set of *nodes*, or vertices, and $E(G)$ the set of *edges*. We call *arrows* the oriented edges of G. For any node $a \in V(G)$, we say that a node $b \in V(G)$ is a *parent* of a if there is an arrow $b \to a$ in the graph, that is, if $b \to a \in E(G)$. We denote by $\mathit{parents}(a) = \{ b \in V(G) : b \text{ is a parent of } a \}$ the set of parents of a. The *in-degree* of a, $\mathit{in\text{-}deg}(a)$, is the cardinality of parents(a). We write $|V(G)|$ for the cardinality of G.

Definition 2. In our setting, a *Signal Propagation Network* (SPN) is a pair

$$\mathcal{N} = \mathcal{N}(G, \mathcal{A}),$$

consisting of the following data:

(i) a directed graph $G = (V(G), E(G))$ without (oriented) cycles, whose nodes represent the discrete variables of the network; we say that the SPN is *n-ary* if every node can take n states;

(ii) a set $\mathcal{A}$ of *activation tensors* A_i, one for each node $a_i \in V(G)$, which determines the state of the node based on the states of its parents. More precisely, let a_i be a node of an n-ary SPN $\mathcal{N}(G)$, with $d = \mathit{in\text{-}deg}(a_i)$. The activation tensor $A_i \in \mathcal{A}$ associated to a_i is any tensor of dimension $d+1$ such that

$$A_i \in \underbrace{(\mathbb{C}^n) \otimes \cdots \otimes (\mathbb{C}^n)}_{d+1 \ \mathit{times}};$$

(iii) an ordering of the nodes compatible with the partial ordering given by the graph (topological sort).

Note that the absence of cycles (as in Definition 2(i)) in the graph implies that some topological ordering always exists (see Definition 2(iii)). This can be easily seen by induction.

Unless otherwise stated, we work with n-ary SPNs in which each node can take the same n states, numbered from 0 to $n-1$. In particular, in the next sections we will mainly consider the binary case $n = 2$, interpreting state 0 as inactive and state 1 as active.

By Definition 2, the SPN $\mathcal{N}(G)$ is defined by its underlying DAG together with the activation tensors associated with its nodes. In particular, if $d = 0$, then an activation tensor associated with a node a_i is simply a vector of $\mathbb{C}^n$. If $d = 1$, then an activation tensor at a_i is an $n \times n$ matrix of complex numbers.

1.3. Tensors of networks

We now explain the role of the activation tensors of the network.

Consider an n-ary signal transmitted by the nodes of an SPN $\mathcal{N}$, following the pattern prescribed by the arrows of the underlying DAG. Let d be the in-degree of a node a and $\{b_1, \dots, b_d\}$ be the set of its parents, ordered according to the total ordering of the network. Then, node a decides its state depending on the states of its parents.

Consequently, the entry $A_{i_1,\dots,i_d,i_{d+1}}$ of the activation tensor represents the *frequency* with which the node a takes the state i_{d+1} when its parent b_1 is in the state i_1, its parent b_2 is in the state i_2, ..., its parent b_d is in the state i_d.

Remark 3. One could adopt a stochastic point of view by requiring

$$\sum_{l=0}^{n-1} A_{i_1,\dots,i_d,l} = 1,$$

so that $A_{i_1,\dots,i_d,i_{d+1}}$ represents the *probability* that the node a takes the state i_{d+1} when b_1 takes the state i_1, ..., and b_d takes the state i_d. We do not impose this assumption, since it is not necessary for our analysis.

Example 4. Some examples of activation tensors, for different values of d, are listed in the following.

- When $d = 0$, the activation tensor $A = (a_0, \dots, a_{n-1})$ is defined such that node a takes the state i with frequency a_i.
- When $d = 1$, well-known examples of activation tensors are the *Jukes-Cantor* matrices, in which all the entries in the diagonal are equal to a parameter α and all the off-diagonal entries are equal to a parameter β. For instance, in the binary case,

$$A = \begin{pmatrix} \alpha & \beta \\ \beta & \alpha \end{pmatrix},$$

  with $\alpha, \beta \in \mathbb{C}$.
- For $d \geq 1$, a relevant case of activation tensor for binary SPNs is the *threshold of value one*, namely the tensor associated to the Boolean logical operator $\exists$. The node a takes the state 1 if at least one of its parents is in the state 1, while a takes the state 0 otherwise. Hence $A_{i_1,\dots,i_d,i_{d+1}}$ is 0 if $i_1 = \cdots = i_d = 0$ and $i_{d+1} = 1$, or when at least one of $i_1, \dots, i_d$ is equal to 1 and $i_{d+1} = 0$. In all remaining cases, the corresponding entry of A is equal to 1.
  An example, for $d = 2$, is shown in Fig. 2.
- The activation tensor above can be modified by replacing the values 0 and 1 with two parameters $\alpha, \beta \in \mathbb{C}$, yielding the *quantum-threshold of value one*. Specifically, the node a will take the state α if it obeys the rule, and the state β otherwise.
  For $d = 2$ the tensor in Fig. 3 is obtained by modifying the 0's in Fig. 2 with β's and the 1's with α's.

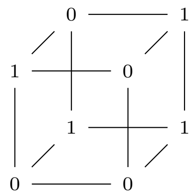


Fig. 2. Activation tensor with threshold of value one. Example of a threshold of value one activation tensor for a node in a binary SPN, with $d = 2$.

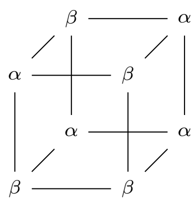


Fig. 3. Activation tensor with quantum-threshold of value one. Example of a quantum-threshold activation tensor for a node in a binary SPN with parameters $\alpha, \beta \in \mathbb{C}$.

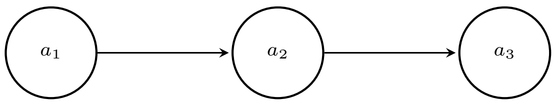


Fig. 4. SPN analogue of a Markov chain with three nodes. Underlying graph of an SPN with three nodes a_1, a_2, a_3. The state of a_2 depends on a_1 and the state of a_3 depends on a_2.

Definition 5. Let $\mathcal{N}(G, \mathcal{A})$ be an n-ary SPN, with $|V(G)| = q$, and nodes $a_1, \ldots, a_q$. The *total tensor* associated with the network is the dimension-q tensor N of size $\underbrace{n \times \cdots \times n}_{q \ times}$ whose entry $N_{i_1,\ldots,i_q}$, where $i_h = 0, \ldots, n-1$, for $h = 1, \ldots, q$, is defined as follows. For each node a_j, $j = 1, \ldots, q$, let $d_j = \textit{in-deg}(a_j) < q$, let A_j be its activation tensor, and let P_j be the d_j-tuple of indices in $\{i_1, \ldots, i_q\}$ corresponding to the parents of a_j, namely $P_j = (i_k : b_k \in \text{parents}(a_j))$ (where the tuple is ordered according to the chosen topological ordering of the SPN). Then

$$N_{i_1,\ldots,i_q} = \Pi_{j=1}^{q} (A_j)_{\mathsf{concat}[P_j, i_j]}. \tag{1}$$

Here, $\mathsf{concat}[x, y]$ denotes the concatenation of tuples x and y.

The following examples, in the binary case, illustrate the meaning of the total tensor.

Example 6. Consider the SPN characterized by the underlying graph in Fig. 4 with three nodes (a_1, a_2, a_3) (in the unique possible ordering) and the following activation tensors: the vector $A_1 = (\alpha, \beta)$ for the source node a_1, the Jukes-Cantor matrix $A_2 = \begin{pmatrix} \alpha & \beta \\ \beta & \alpha \end{pmatrix}$ for a_2 and set $A_3 = A_2$ for a_3. This can be considered as a Signal Propagation Network analogue of a short Markov chain.

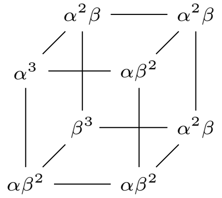


Fig. 5. Total tensor of the SPN in Example 6. Representation of the total tensor of the SPN in Example 6 with parameters $\alpha, \beta \in \mathbb{C}$. A possible reading of the values of the total tensor is the following: $N_{0,0,0}$ means that a_1 takes state 0 (α), then a_2 obeys and takes state 0 (α again), and a_3 obeys and takes state 0 (another α), so $N_{0,0,0} = \alpha^3$; $N_{0,1,1}$ means that a_1 takes state 0 (α), then a_2 disobeys and takes state 1 (β), while a_3 obeys and takes state 1 (another α), so $N_{0,1,1} = \alpha^2\beta$; and so on.

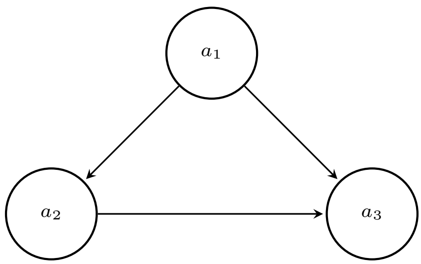


Fig. 6. Acyclic triangle with three nodes. Underlying graph of an SPN with three nodes a_1, a_2, a_3. The state of the node a_2 depends on the state of the node a_1, and the state of the node a_3 depends on both the states of a_1 and a_2.

To compute the total tensor, we determine the indices of the parents of each node. For node a_1, which is the source, we have $\textit{in-deg}(a_1) = 0$ and $P_1 = ()$. For node a_2, we have $\textit{in-deg}(a_2) = 1$, $\text{parents}(a_2) = \{a_1\}$ and $P_2 = (i_1)$. Finally, for node a_3, we have $\textit{in-deg}(a_3) = 1$, $\text{parents}(a_3) = \{a_2\}$ and $P_3 = (i_2)$.

Hence, by Definition 5,

$$N_{i_1,i_2,i_3} = (A_1)_{i_1}(A_2)_{i_1,i_2}(A_3)_{i_2,i_3},$$

and the corresponding tensor is represented in Fig. 5.

Example 7. Consider the SPN associated with the graph in Fig. 6, with three nodes (a_1, a_2, a_3) in the unique possible ordering and, as activation tensors, the vector $A_1 = (\alpha, \beta)$ for the source node a_1, the Jukes-Cantor matrix $A_2 = \begin{pmatrix} \alpha & \beta \\ \beta & \alpha \end{pmatrix}$ for a_2, and the quantum-threshold of value one tensor A_3 of Example 4 for a_3.

For node a_1, which is the source, we have $A_1 = (\alpha, \beta)$, $\textit{in-deg}(a_1) = 0$, and $P_1 = ()$. For node a_2, we have $\textit{in-deg}(a_2) = 1$ and $P_2 = (i_1)$. For node a_3, we have $\textit{in-deg}(a_3) = 2$ and $P_3 = (i_1, i_2)$.

Then, the total tensor N of the SPN, according to Equation (1), can be computed as

$$N_{i_1,i_2,i_3} = (A_1)_{i_1}(A_2)_{i_1,i_2}(A_3)_{i_1,i_2,i_3},$$

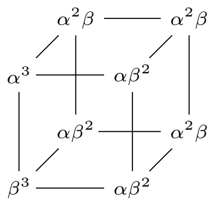


Fig. 7. Total tensor of the acyclic triangle. Representation of the total tensor of a binary acyclic triangle with nodes a_1, a_2, a_3 and parameters $\alpha, \beta \in \mathbb{C}$.

and its representation is shown in Fig. 7.

2. The Bhattacharya-Mesner Product

2.1. Bhattacharya-Mesner Product

The Bhattacharya-Mesner Product of tensors is a d-ary operation in the space of d-dimensional cubical tensors which generalizes the row-by-column product of matrices. It is defined for tensors with coefficients in any field $\mathbb{K}$ (and even when $\mathbb{K}$ is just a ring), although we will apply it mainly to complex tensors.

Definition 8. Let $T_1 = (t^{(1)}_{i_1,\dots,i_d}), T_2 = (t^{(2)}_{i_1,\dots,i_d}), \dots, T_d = (t^{(d)}_{i_1,\dots,i_d})$ be dimension-d tensors of sizes $l \times n_2 \times \cdots \times n_d$, $n_1 \times l \times n_3 \times \cdots \times n_d$, $\cdots, n_1 \times \cdots \times l$, respectively. The *generalized Bhattacharya-Mesner Product* (shortly *BMP*), denoted by $\circ(T_1, T_2, \dots, T_d)$, is a d-ary operation whose result is a tensor $T = (t_{i_1,\dots,i_d})_{i_1,\dots,i_d}$ of size $n_1 \times n_2 \times \cdots \times n_d$ defined by:

$$(T)_{i_1,\dots,i_d} = \sum_{h=0}^{l-1} (T_1)_{h,i_2,\dots,i_d} \cdots (T_d)_{i_1,i_2,\dots,h},$$

that is,

$$t_{i_1,\dots,i_d} = \sum_{h=0}^{l-1} t^{(1)}_{h,i_2,\dots,i_d} \cdots t^{(d)}_{i_1,i_2,\dots,h}.$$

Remark 9. The BMP generalizes the classical matrix multiplication when $d = 2$. Indeed, if we consider two matrices $T_1 \in \mathbb{C}^{n_1 \times l}$ and $T_2 \in \mathbb{C}^{l \times n_2}$, their product is defined as

$$(T_1 \cdot T_2)_{i,j} = \sum_{h=0}^{l-1} (T_1)_{i,h} \cdot (T_2)_{h,j} = \sum_{h=0}^{l-1} (T_2)_{h,j} (T_1)_{i,h} = \circ(T_2, T_1).$$

Hence, $\circ(A_2, A_1)$ coincides with the usual matrix multiplication $A_1 \cdot A_2$, where the order of the factors is reversed.

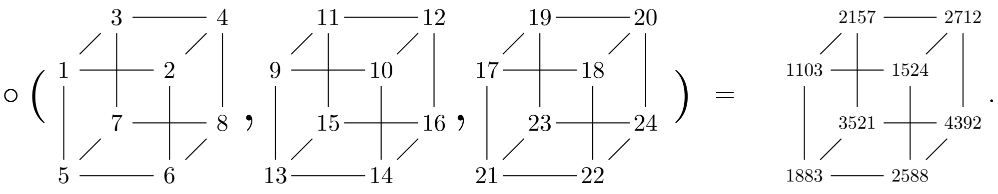
Example 10. Let us give an example of the product of three $2 \times 2 \times 2$ tensors $T_1 = (t^{(1)}_{i_1,i_2}), T_2 = (t^{(2)}_{i_1,i_2}), T_3 = (t^{(3)}_{i_1,i_2})$. The BMP

$$T = (t_{i_1,i_2}) = \circ(T_1, T_2, T_3)$$

can be computed as follows:

$$t_{i_1,i_2} = \sum_{h=0}^{1} t^{(1)}_{h,i_2,i_3} \cdot t^{(2)}_{i_1,h,i_3} \cdot t^{(3)}_{i_1,i_2,h} = t^{(1)}_{0,i_2,i_3} \cdot t^{(2)}_{i_1,0,i_3} \cdot t^{(3)}_{i_1,i_2,0} + t^{(1)}_{1,i_2,i_3} \cdot t^{(2)}_{i_1,1,i_3} \cdot t^{(3)}_{i_1,i_2,1}.$$

For example,



2.2. Some algebraic properties of BMP

It is almost immediate to observe that the BMP is non-commutative. The BMP is also non-associative, in the sense that, for general dimension-3 tensors A, B, C, D, E,

$$\circ(\circ(A, B, C), D, E) \neq \circ(A, B, \circ(C, D, E)).$$

Nevertheless, the BMP is multilinear and satisfies some interesting properties.

Multiple identities.

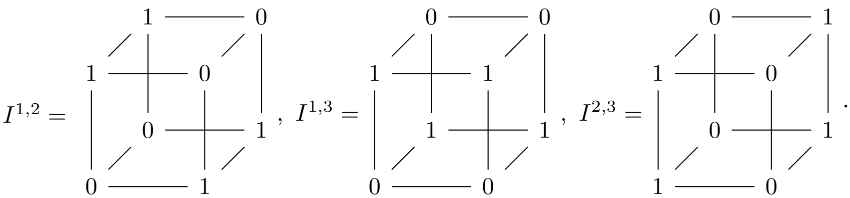
We continue the comparison with matrices (that is, dimension-2 tensors) in order to show how their behavior is generalized in case of dimension-d tensors. When $d = 2$, for any matrix A of size $m \times n$, there exist identity matrices I_m and I_n of size $m \times m$ and $n \times n$, respectively, such that $AI_n = A$ and $I_m A = A$. The role of the identity matrix can be generalized for dimension-n tensors by introducing *identitary tensors*.

Definition 11. Let $n, d \in \mathbb{N}$. For all $j, k = 1, \ldots, d$ with $j < k$, define the dimension-d *identitary* tensors $I^{j,k}$ of size $n \times \cdots \times n$ by

$$(I^{j,k})_{i_1,\ldots,i_d} = \begin{cases} 1 & \text{if } i_j = i_k; \\ 0 & \text{else.} \end{cases}$$

Notice that when $d = 2$ there is only one such tensor, namely $I^{1,2}$, which coincides with the identity matrix.

Example 12. In the case of $2 \times 2 \times 2$ tensors, the dimension-3 identitary tensors are the following:



The role of identitary tensors is clarified by the following proposition.

Proposition 13. *Let A be a cubical dimension-d tensor. Then,*

$$\circ(A, I^{1,2}, \dots, I^{1,d}) = A, \qquad \circ(I^{1,d}, \dots, I^{d-1,d}, A) = A,$$

and, for any $j = 2, \dots, d-1$,

$$\circ(I^{1,j}, \dots, I^{j-1,j}, A, I^{j,j+1}, \dots, I^{j,d}) = A.$$

Proof. Let us prove the case where $j = 2, \dots, d-1$. The proofs for the first two equalities are similar. Let $T = \circ(I^{1,j}, \dots, I^{j-1,j}, A, I^{j,j+1}, \dots, I^{j,d})$. By Definition 8,

$$T_{i_1,\dots,i_d} = \sum_{h=0}^{l-1} (I^{1,j})_{h,i_2,\dots,i_d} \cdots (I^{j-1,j})_{i_1,\dots,i_{j-2},h,i_j,\dots,i_d} \cdot \\ \cdot A_{i_1,\dots i_{j-1},h,i_{j+1},\dots,i_d} \cdot (I^{j,j+1})_{i_1,\dots,i_j,h,i_{j+2},\dots,i_d} \cdots (I^{j,d})_{i_1,\dots i_{d-1},h}.$$

For $j = 2, \dots, d-1$, the first term $I^{1,j}$ appearing in each addend of the summation ensures that only the addend in which $h = i_j$ is different from 0.

Therefore

$$(I^{1,j})_{i_j,i_2,\dots,i_d} = (I^{j,j+1})_{i_j,\dots,i_{j-1},i_j,i_j,i_{j+2},\dots,i_d} = \cdots = (I^{j,d})_{i_1,\dots,i_{d-1},i_j} = 1,$$

and, hence, $T_{i_1,\dots,i_d} = A_{i_1,\dots,i_d}$. $\square$

In particular, from Proposition 13, one obtains that the dimension-3 identitary tensors of Example 12 satisfy

$$\circ(A, I^{1,2}, I^{1,3}) = A$$

for every $2 \times 2 \times 2$ tensor A.

Transposition.

In the case of dimension-2 tensors, the transpose of the product of two matrices is equal to the product of the transpose of the second matrix and the transpose of the first matrix, namely, for matrices A, B properly sized, we have $(AB)^T = B^T A^T$. Again, this behavior can be generalized for dimension-d tensors.

Definition 14. Let $\mathfrak{S}_d$ denote the symmetric group on d elements, and A be a dimension-d tensor. For any $\sigma \in \mathfrak{S}_d$, the *$\sigma$-transpose of A* is the dimension-d tensor A^σ obtained by permuting the d indices of A according to σ.

In the case of matrices, there are 2 indices and hence only 2 permutations, yielding the identity and the usual transpose. For dimension-d tensors, there are $d!$ such permutations, obtaining $(d!-1)$ σ-transposes of an dimension-d tensor. From the definition of the BMP, it is immediate to prove the following.

Proposition 15. *Let $A_1, \dots, A_d$ be properly sized tensors of dimension d and, given $\sigma \in \mathfrak{S}_d$ a permutation of d indices, call $A_1^\sigma, \dots, A_d^\sigma$ their σ-transposes. Then*

$$(\circ(A_1, \dots, A_d))^\sigma = \circ(A^\sigma_{\sigma(1)}, \dots, A^\sigma_{\sigma(d)}).$$

Proof. We report the computation for $d = 3$. Let A_1, A_2, A_3 be 3-tensors of sizes $l \times n_2 \times n_3$, $n_1 \times l \times n_3$, $n_1 \times n_2 \times l$, respectively, and consider the permutation $\sigma \in \mathfrak{S}_3$ defined as

$$\sigma(1) = 1, \qquad \sigma(2) = 3, \qquad \sigma(3) = 2.$$

We define $T := \circ(A_1, A_2, A_3) = \sum_{h=0}^{l-1} (A_1)_{h,j,k} \cdot (A_2)_{i,h,k}(A_3)_{i,j,k}$, and, by Definition 14,

$$\begin{aligned}(T^\sigma)_{i,j,k} &= \Big(\sum_{h=0}^{l-1} (A_1)_{h,j,k} \cdot (A_2)_{i,h,k} \cdot (A_3)_{i,j,h}\Big)^\sigma = \\ &= \sum_{h=0}^{l-1} (A_1)_{h,k,j} \cdot (A_2)_{i,h,j} \cdot (A_3)_{i,k,h} = T_{i,k,j}.\end{aligned}$$

On the other hand, we denote $T' = \circ(A^\sigma_{\sigma(1)}, A^\sigma_{\sigma(2)}, A^\sigma_{\sigma(3)})$, that is

$$T'_{i,j,k} = \circ(A_1^\sigma, A_3^\sigma, A_2^\sigma)_{i,j,k}.$$

Since

$$(A_1^\sigma)_{i,j,k} = (A_1)_{i,k,j}, \qquad (A_2^\sigma)_{i,j,k} = (A_2)_{i,k,j}, \qquad (A_3^\sigma)_{i,j,k} = (A_3)_{i,k,j},$$

T' can be computed as

$$T'_{ijk} = \circ(A_1^\sigma, A_3^\sigma, A_2^\sigma)_{i,j,k} = \sum_{h=0}^{l-1} (A_1)^\sigma_{h,j,k} \cdot (A_3)^\sigma_{i,h,k} \cdot (A_2)^\sigma_{i,j,h} =$$
$$= \sum_{h=0}^{l-1} (A_1)_{h,k,j} \cdot (A_3)_{i,k,h} \cdot (A_2)_{i,h,j}.$$

Finally,

$$(\circ(A_1, A_2, A_3)^\sigma)_{ijk} = T_{i,k,j} = \sum_{h=0}^{l-1} (A_1)_{h,k,j} \cdot (A_2)_{i,h,j} \cdot (A_3)_{i,k,h} =$$
$$= \sum_{h=0}^{l-1} (A_1)_{h,k,j} \cdot (A_3)_{i,k,h} \cdot (A_2)_{i,h,j} = T'_{i,j,k} =$$
$$= \circ(A_1^\sigma, A_3^\sigma, A_2^\sigma)_{i,j,k}. \quad \square$$

3. Network characterization in terms of BMP

3.1. Expansion of tensors

The following operations are fundamental in transforming the activation tensor of each node into a tensor which, through the BMP, yields the total tensor of the SPN.

We call these operations *blow* and *forget*. The former increases the dimension of a tensor from d to $d+1$, while the latter increases the dimension of a tensor from d to an arbitrary dimension $d' > d$.

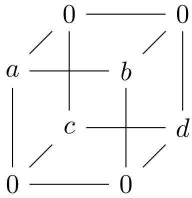
The *blow* operation.

Definition 16. Let $T \in V^{\otimes d}$. The *blow*, or *inflation*, of T is the tensor $b(T)$ of dimension $d+1$ defined by

$$(b(T))_{i_1,\dots,i_{d+1}} = \begin{cases} T_{i_1,\dots,i_d} & \text{if } i_1 = i_{d+1}; \\ 0 & \text{else.} \end{cases} \tag{2}$$

We observe that this definition of inflation (or blow) extends the notion introduced in [4, Section 2.4].

Example 17. The blow $b(M)$ of the matrix $M = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ is the following dimension-3 tensor:

$b(M) =$ 

Notice that the blow is defined only in coordinates.

Remark 18. If $T = v_1 \otimes \cdots \otimes v_d$ is a tensor of rank 1, then in general its blow is no longer of rank 1. Indeed, if $\{e_1, \dots, e_n\}$ is a basis of V and

$$v_1 = a_1 e_1 + \cdots + a_n e_n,$$

then

$$b(T) = \sum_{i=1}^{n} a_i e_i \otimes v_2 \otimes \cdots \otimes v_d \otimes e_i.$$

The *forget* operation.

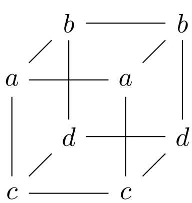
Definition 19. Let $T \in V^{\otimes d}$. For $d' > d$, let $J \subset \{1, \dots, d'\}$ be such that $|J| = d' - d$. The J-th *forget*, or J-th *copy*, of T is the tensor $f_J(T)$ of dimension d' defined by

$$(f_J(T))_{i_1,\dots,i_{d'}} = T_{j_1,\dots,j_d}, \tag{3}$$

where $(j_1, \dots, j_d)$ is the list obtained from $(i_1, \dots, i_{d'})$ by deleting the entries i_k for all $k \in J$.

For instance, the forget operation $f_{\{k\}}(T)$ creates a tensor whose entry with indices $i_1, \dots, i_{d+1}$ corresponds to the entry of T where $i_h = j_h$ for $h = 1, \dots, k-1$, and $i_h = j_{h-1}$ for $h = k+1, \dots, d+1$.

Example 20. The forget $f_{\{2\}}(M)$ of the matrix $M = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ is the following dimension-3 tensor:

$f_{\{2\}}(M) =$ 

In this example, $(f_{\{2\}}(M))_{i_1,i_3} = M_{j_1,j_2}$ where $i_1 = j_1$, $i_3 = j_2$. Indeed, $(f_{\{2\}}(M))_{0,0,0} = (f_{\{2\}}(M))_{0,1,0} = M_{0,0} = a$.

Remark 21. If $T = v_1 \otimes \cdots \otimes v_d$ is a tensor of rank 1, then its J-th forget tensor $f_J(T)$ is again a tensor of rank 1. Indeed, if $\{e_1, \ldots, e_n\}$ is a basis of V for which $v_1 = a_1e_1 + \cdots + a_ne_n$, then

$$f_{\{k\}}(T) = w_1 \otimes \cdots \otimes w_{d'},$$

where $w_i = v_i$ if $i \notin J$, while $w_i = e_1 + \cdots + e_n$ if $i \in J$.

Notice that when $J = \emptyset$, one has $f_J(T) = T$.

3.2. The theorem

We are now ready to explain how the total tensor of an SPN can be obtained as the BMP of tensors constructed from the activation tensors of the network.

Let $\mathcal{N}(G, \mathcal{A})$ be an SPN with $|V(G)| = q$ nodes and total ordering $(a_1, a_2, \ldots, a_q)$. We call a_1 the *source* and a_q the *sink* of the network.

For each node $a_i \in V(G)$, we define P_i as the set of indices corresponding to the parents of a_i (see Definition 5), so that $P_i \subset \{1, \ldots, i-1\}$. Let p_i be the cardinality of P_i ($p_i = |P_i|$) and let T_{a_i} be the activation tensor of the node a_i. According to Definition 2(*ii*), tensor A_i has dimension $p_i + 1$.

Procedure 22. For each node $a_i \in V(G)$, we define the associated dimension-q tensor T_{a_i} by the following procedure.

Step 1. Start with the activation tensor A_i.

Step 2. If $i > 1$, compute the dimension-i forget tensor $A'_i = f_J(A_i)$, where $J = \{1, \ldots, i-1\} \setminus P_i$.

Step 3. If $i \neq q$, compute the blow tensor $A''_i = b(A'_i)$. Thus A''_i has dimension $i + 1$.

Step 4. Define T_{a_i} as the dimension-q forget tensor $T_{a_i} = f_H(A''_i)$, where $H = \{i + 2, \ldots, q\}$.

This construction, based on Equation (2) and Equation (3), ensures that each activation tensor is expanded to a dimension-q tensor. More precisely:

- the initial dimension of A_i is $p_i + 1 \leq i$;
- since $|J| = i - 1 - p_i$, after *Step 2* the tensor has dimension i;
- if $i < q$, after *Step 3* the tensor has dimension $i + 1$;
- if $i + 1 < q$, *Step 4* adds the remaining indices and yields a dimension-q tensor.

Theorem 23. *Let $\mathcal{N}(G, \mathcal{A})$ be an SPN with $|V(G)| = q$ nodes and total ordering $(a_1, a_2, \ldots, a_q)$, such that a_1 is the source and a_2 is the sink. Moreover, assume that $T_{a_1}, \ldots, T_{a_q}$ are the dimension-q tensors obtained from the activation tensors $A_1, \ldots, A_q$ by applying Procedure 22. Then the BMP*

$$\circ(T_{a_q}, T_{a_1}, \ldots, T_{a_{q-1}}) \tag{4}$$

computes the total tensor N of the network.

We stress that the tensor T_{a_q} associated with the sink node must appear in the first position of the BMP in Equation (4).

Before proving the main theorem, let us illustrate the procedure with some examples.

Example 24. Consider the basic case of an SPN in which there are only two nodes, a_1 the source and a_2 the sink. The activation tensor of a_1 is a vector A_1. The activation tensor of a_2 depends on whether the arrow $a_1 \to a_2$ is present.

Assume that the arrow $a_1 \to a_2$ exists. Then, the activation tensor $A_2 = (a_{ij})$ of a_2 is a matrix describing the reaction of a_2 to the state of a_1. Applying Procedure 22 to A_1, we obtain $T_{a_1} = b(A_1)$, since here $J = H = \emptyset$. By Definition 16, T_{a_1} is the diagonal matrix with the vector A_1 in the diagonal. If we apply Procedure 22 to A_2, since $J = H = \emptyset$, we simply obtain the matrix

$$T_{a_2} = A_2.$$

Hence, the BMP $\circ(T_{a_2}, T_{a_1})$, which in this case is just the usual matrix multiplication $T_{a_1} \cdot T_{a_2}$, gives the matrix N with entries

$$N_{ij} = (A_1)_i (A_2)_{i,j},$$

that is the total tensor of the network.

Assume now that the arrow $a_1 \to a_2$ does not exist. Then, the activation tensor of a_2 is a vector A_2. As above, applying Procedure 22 to A_1, we obtain $T_{a_1} = b(A_1)$, that is the diagonal matrix with the vector A_1 in the diagonal. If we apply Procedure 22 to A_2, we get the forget tensor $T_{a_2} = f_{\{1\}}(A_2)$, which is the $n \times n$ matrix whose rows are all equal to A_2. Then, the BMP $N = \circ(T_{a_2}, T_{a_1})$, which corresponds to the matrix multiplication $T_{a_1} \cdot T_{a_2}$, has entries

$$N_{ij} = (A_1)_i (A_2)_j,$$

expressing the independence of the two nodes. Again, N is the tensor of the network.

Example 25. Consider the SPN of Example 6 with activation tensor A_1 of the source node a_1 equal to (α, β) and let the activation tensors A_2 and A_3 of nodes a_2 and a_3 be both equal to the Jukes-Cantor matrix $A = \begin{pmatrix} \alpha & \beta \\ \beta & \alpha \end{pmatrix}$ of Example 4.

The total tensor of the network is thus given by the BMP of the tensors $T_{a_3}, T_{a_1}, T_{a_2}$, where

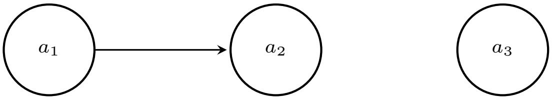


Fig. 8. SPN with three nodes, two of which are sources. Underlying graph of an SPN with three nodes a_1, a_2, a_3. The state of node a_2 depends on a_1, nodes a_1 and a_3 are sources.

$$T_{a_1} = f_{\{3\}}(b(A_1)), \qquad T_{a_2} = b(A_2), \qquad T_{a_3} = f_{\{1\}}(A_3).$$

One computes

$$T_{a_1} = \begin{tikzcd}
& \alpha \arrow[rr, dash] \arrow[dl, dash] \arrow[dd, dash] & & 0 \arrow[dl, dash] \arrow[dd, dash] \\
\alpha \arrow[rr, dash] \arrow[dd, dash] & & 0 \arrow[dd, dash] & \\
& 0 \arrow[rr, dash] \arrow[dl, dash] & & \beta \arrow[dl, dash] \\
0 \arrow[rr, dash] & & \beta &
\end{tikzcd}, T_{a_2} = \begin{tikzcd}
& 0 \arrow[rr, dash] \arrow[dl, dash] \arrow[dd, dash] & & 0 \arrow[dl, dash] \arrow[dd, dash] \\
\alpha \arrow[rr, dash] \arrow[dd, dash] & & \beta \arrow[dd, dash] & \\
& \beta \arrow[rr, dash] \arrow[dl, dash] & & \alpha \arrow[dl, dash] \\
0 \arrow[rr, dash] & & 0 &
\end{tikzcd}, T_{a_3} = \begin{tikzcd}
& \beta \arrow[rr, dash] \arrow[dl, dash] \arrow[dd, dash] & & \alpha \arrow[dl, dash] \arrow[dd, dash] \\
\alpha \arrow[rr, dash] \arrow[dd, dash] & & \beta \arrow[dd, dash] & \\
& \beta \arrow[rr, dash] \arrow[dl, dash] & & \alpha \arrow[dl, dash] \\
\alpha \arrow[rr, dash] & & \beta &
\end{tikzcd},$$

and the product $\circ(T_{a_3}, T_{a_1}, T_{a_2})$ gives

$$N = \begin{tikzcd}
& \alpha^2\beta \arrow[rr, dash] \arrow[dl, dash] \arrow[dd, dash] & & \alpha^2\beta \arrow[dl, dash] \arrow[dd, dash] \\
\alpha^3 \arrow[rr, dash] \arrow[dd, dash] & & \alpha\beta^2 \arrow[dd, dash] & \\
& \beta^3 \arrow[rr, dash] \arrow[dl, dash] & & \alpha^2\beta \arrow[dl, dash] \\
\alpha\beta^2 \arrow[rr, dash] & & \alpha\beta^2 &
\end{tikzcd},$$

which parameterizes a twisted cubic curve in the projective space of tensors of size $2 \times 2 \times 2$, lying in the linear subspace of equations

$$N_{0,1,0} = N_{1,0,0} = N_{1,1,0}, \qquad N_{0,0,1} = N_{0,1,1} = N_{1,1,1}.$$

Note that both the left and right faces of the tensor have rank 1, corresponding to the fact that a_3 is independent of a_1 given a_2.

If we remove from the network the arrow that joins a_2 and a_3, and we consider as activation tensors A_1 and A_2 as above, and $A_3 = (\alpha, \beta)$, we obtain a new SPN whose graph is represented in Fig. 8.

The transformed tensors are

$$T_{a_1} = f_{\{3\}}(b(A_1)), \qquad T_{a_2} = b(A_2), \qquad T_{a_3} = f_{\{1,2\}}(A_3),$$

that is

$$T_{a_1} = \begin{tikzcd}[row sep=small, column sep=small]
& \alpha \arrow[rr, dash] \arrow[dl, dash] \arrow[dd, dash] & & 0 \arrow[dl, dash] \arrow[dd, dash] \\
\alpha \arrow[rr, dash] \arrow[dd, dash] & & 0 \arrow[dd, dash] & \\
& 0 \arrow[rr, dash] \arrow[dl, dash] & & \beta \arrow[dl, dash] \\
0 \arrow[rr, dash] & & \beta &
\end{tikzcd}, T_{a_2} = \begin{tikzcd}[row sep=small, column sep=small]
& 0 \arrow[rr, dash] \arrow[dl, dash] \arrow[dd, dash] & & 0 \arrow[dl, dash] \arrow[dd, dash] \\
\alpha \arrow[rr, dash] \arrow[dd, dash] & & \beta \arrow[dd, dash] & \\
& \beta \arrow[rr, dash] \arrow[dl, dash] & & \alpha \arrow[dl, dash] \\
0 \arrow[rr, dash] & & 0 &
\end{tikzcd}, T_{a_3} = \begin{tikzcd}[row sep=small, column sep=small]
& \beta \arrow[rr, dash] \arrow[dl, dash] \arrow[dd, dash] & & \beta \arrow[dl, dash] \arrow[dd, dash] \\
\alpha \arrow[rr, dash] \arrow[dd, dash] & & \alpha \arrow[dd, dash] & \\
& \beta \arrow[rr, dash] \arrow[dl, dash] & & \beta \arrow[dl, dash] \\
\alpha \arrow[rr, dash] & & \alpha &
\end{tikzcd}.$$

We get that the total tensor of the SPN is the BMP $\circ(T_{a_3}, T_{a_1}, T_{a_2})$, corresponding to the tensor

$$N = \begin{tikzcd}[row sep=small, column sep=small]
& \alpha^2\beta \arrow[rr, dash] \arrow[dl, dash] \arrow[dd, dash] & & \alpha\beta^2 \arrow[dl, dash] \arrow[dd, dash] \\
\alpha^3 \arrow[rr, dash] \arrow[dd, dash] & & \alpha^2\beta \arrow[dd, dash] & \\
& \beta^3 \arrow[rr, dash] \arrow[dl, dash] & & \alpha\beta^2 \arrow[dl, dash] \\
\alpha\beta^2 \arrow[rr, dash] & & \alpha^2\beta &
\end{tikzcd},$$

which parameterizes a different twisted cubic curve.

Note that not only both the left and right faces of the tensor have rank 1, that is a_2 is independent of a_1 given a_3, but also the upper and lower faces have rank 1, since a_2 and a_3 are independent.

Example 26. It is obvious that in a network without arrows all the activation tensors of the nodes are vectors, and the tensor product of these vectors defines a rank one tensor N which is the total tensor of the network.

More explicitly, let an n-ary SPN with no arrows have q nodes $a_1, \dots, a_q$, with total ordering $(a_1, \dots, a_q)$, and let $w_i \in \mathbb{C}^n$ be the activation vector of node a_i, $i = 1, \dots, q$. Then, Theorem 23 implies that the BMP of the associated dimension-q tensors $\{T_{a_i}\}_{i=1}^q$, obtained from $\{w_i\}_{i=1}^q$ by applying Procedure 22, is equal to

$$w_1 \otimes \cdots \otimes w_q.$$

This can also be verified directly. Namely, we have:

$$T_{a_1} = f_{\{3,\dots,q\}}(b(w_1)), \quad T_{a_j} = f_{\{j+2,\dots,q\}}(b(f_{\{1,\dots j-1\}}(w_j))) \text{ for } j = 2, \dots, q-2;$$

and

$$T_{a_{q-1}} = b(f_{\{1,\dots,q-2\}}(w_{q-1})), \quad T_{a_q} = f_{\{1,\dots,q-1\}}(w_q).$$

Thus, assuming $w_i = (w_{i1}, \dots, w_{in})$ for all $i = 1, \dots, q$, the computation of the product $N = \circ(T_{a_q}, T_{a_1}, \dots, T_{a_{q-1}})$ goes as follows:

$$N_{i_1,\dots,i_q} = \sum_{h=0}^{n-1} (T_{a_q})_{h,i_2,\dots,i_q} \cdot (T_{a_1})_{i_1,h,i_3,\dots,q} \cdots (T_{a_{q-1}})_{i_1,\dots,i_{q-1},h}.$$

Since the last operation used to obtain $T_{a_{q-1}}$ is a blow, the only surviving summand is the one with $h = i_1$. Hence

$$N_{i_1,\dots,i_q} = (T_{a_q})_{i_1,i_2,\dots,i_q} \cdot (T_{a_1})_{i_1,i_1,i_3,\dots,i_q} \cdots (T_{a_{q-1}})_{i_1,\dots,i_{q-1},i_1}.$$

By construction, one gets

$$(T_{a_q})_{i_1,i_2,\dots,i_q} = (f_{\{1,\dots,q-1\}}(w_q))_{i_1,i_2,\dots,i_q} = w_{qi_q}.$$

Analogously, for $j = 2, \dots, q-2$,

$$(T_{a_j})_{i_1,i_2,\dots,i_q} = (f_{\{j+2,\dots,q\}}(b(f_{\{1,\dots,j-1\}}(w_j))))_{i_1,i_2,\dots,i_{j-1},i_1,i_{j+1},\dots,i_q},$$

where the index i_1 in the middle appears in position j. Thus,

$$(f_{\{j+2,\dots,q\}}(b(f_{\{1,\dots,j-1\}}(w_j))))_{i_1,i_2,\dots,i_{j-1},i_1,i_{j+1}\dots,i_q} = (b(f_{\{1,\dots,j-1\}}(w_j)))_{i_1,i_2,\dots,i_{j-1},i_1},$$

and

$$(f_{\{1,\dots,j-1\}}(w_j))_{i_1,i_2,\dots,i_{j-1}} = w_{ji_j}.$$

With the same argument, one proves that

$$(T_{a_1})_{i_1,i_1,i_3,\dots,i_q} = w_{1i_1}, \text{ and } (T_{a_{q-1}})_{i_1,i_2,\dots,i_{q-1},i_1} = w_{q-1\ i_{q-1}}.$$

Then

$$N_{i_1,\dots,i_q} = w_{1i_1} \cdots w_{qi_q},$$

that is,

$$N = w_1 \otimes \cdots \otimes w_q.$$

Proof of Theorem 23. We work by induction on the number of nodes, the case $q = 1$ being trivial. We could also use Example 24 as a basis for the induction, starting with $q = 2$.

Assume the statement holds for every SPN with at most $q-1$ nodes, and pick an SPN $\mathcal{N}(G, \mathcal{A})$ with $|V(G)| = q$ and total ordering $(a_1, \dots, a_q)$.

If we exclude the sink a_q, then we obtain an SPN $\mathcal{N}'(G', \mathcal{A}')$ for which the theorem holds, by induction. Use on the nodes of G' the ordering defined for the nodes of G.

Let N be the total tensor of $\mathcal{N}$, and let N' be the total tensor of $\mathcal{N}'$. Then, the relation between N and N' can be described as follows. Call $P = P_q$ the set of indices corresponding to the parents of a_q, sorted by the ordering defined for G. If A_q is the activation tensor of a_q, then

$$N_{i_1,\dots,i_q} = N'_{i_1,\dots,i_{q-1}}(A_q)_{P(i_1,\dots,i_q)} \tag{5}$$

where $P(i_1,\dots,i_q)$ denotes the tuple obtained by selecting from $(i_1,\dots,i_q)$ the indices corresponding to the elements of P, and then appending i_q.

After applying Procedure 22, let T'_{a_i} be the tensor associated with a_i for the SPN $\mathcal{N}'$, and let T_{a_i} be the tensor associated with the same node for the SPN $\mathcal{N}$, with $i = 1,\dots q-1$. Then, T'_{a_i} and T_{a_i} are linked by the following relations:

$$\begin{cases} T_{a_i} & = f_{\{q\}}(T'_{a_i}) \quad \text{if } i = 1,\dots,q-2; \\ T_{a_{q-1}} & = b(T'_{a_{q-1}}). \end{cases}$$

Moreover, applying Procedure 22 to the activation tensor A_q, we obtain the tensor T_{a_q} associated with node a_q:

$$T_{a_q} = f_J(A_q) \quad \text{where } J = \{1\dots,q-1\} \setminus P. \tag{6}$$

Now, we only need to perform the required calculations. The entries of the tensor $M = \circ(T_{a_q}, T_{a_1},\dots,T_{a_{q-1}})$ have the following structure:

$$M_{i_1,\dots,i_q} = \sum_{h=0}^{n-1} (T_{a_q})_{h,i_2,\dots,i_q} \cdot (T_{a_1})_{i_1,h,i_3,\dots,i_q} \cdots (T_{a_{q-1}})_{i_1,\dots,i_{q-1},h}.$$

Since $T_{a_{q-1}}$ is constructed according to Equation (2), the only nonzero summand is the one with $h = i_1$. Therefore,

$$M_{i_1,\dots,i_q} = (T_{a_q})_{i_1,i_2,\dots,i_q} \cdot (T_{a_1})_{i_1,i_1,i_3,\dots,i_q} \cdots (T_{a_{q-1}})_{i_1,\dots,i_{q-1},i_1}.$$

By Equation (6),

$$(T_{a_q})_{i_1,i_2,\dots,i_q} = (f_J(A_q))_{i_1,i_2,\dots,i_q} = (A_q)_{P(i_1,\dots,i_q)}.$$

Moreover, for $j = 1,\dots,q-2$,

$$(T_{a_j})_{i_1,\dots,i_{j-1},i_1,i_{j+1},\dots,i_q} = (T'_{a_j})_{i_1,\dots,i_{j-1},i_1,i_{j+1},\dots,i_{q-1}}, \tag{7}$$

and

$$(T_{a_{q-1}})_{i_1,\dots,i_{q-1},i_1} = (T'_{a_{q-1}})_{i_1,\dots,i_{q-1}} \tag{8}$$

for the last factor in the product.

By the inductive hypothesis, N' can be computed as $\circ(T'_{q-1}, T'_1,\dots,T'_{q-2})$, thus

$$N'_{i_1,\dots,i_q} = \sum_{h=0}^{n-1} (T'_{a_{q-1}})_{h,i_2,\dots,i_{q-1}} \cdot (T'_1)_{i_1,h,i_3,\dots,i_{q-1}} \cdots (T_{a_{q-2}})_{i_1,\dots,i_{q-2},h}.$$

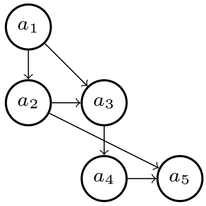


Fig. 9. SPN with five nodes. Underlying graph of an SPN with nodes a_1, a_2, a_3, a_4, a_5. The source is a_1 and the sink is a_5.

With the same blow argument already introduced in Example 26, the unique surviving summand has $h = i_1$, so

$$N'_{i_1,\dots,i_q} = (T'_{a_{q-1}})_{i_1,\dots,i_{q-1}} (T'_{a_1})_{i_1,i_1,i_3,\dots,i_{q-1}} \cdots (T'_{a_{q-2}})_{i_1,\dots,i_{q-2},i_1}.$$

Comparing with Equation (7) and Equation (8), we obtain

$$N'_{i_1,\dots,i_q} = (T_{a_{q-1}})_{i_1,\dots,i_{q-1},i_1} \cdot (T_{a_1})_{i_1,i_1,i_3,\dots,i_q} \cdots (T_{a_{q-2}})_{i_1,\dots,i_{q-1},i_1}. \tag{9}$$

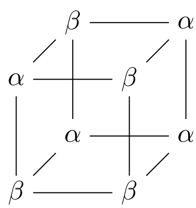
Finally, from Equation (5) and Equation (9), we get

$$\begin{aligned} N_{i_1,\dots,i_q} &= N'_{i_1,\dots,i_{d-1}} (A_q)_{P(i_1,\dots,i_q)} = \\ &= (T_{a_{q-1}})_{i_1,\dots,i_{q-1},i_1} \cdot (T_{a_1})_{i_1,i_1,i_3,\dots,i_q} \cdots (T_{a_{q-2}})_{i_1,\dots,i_{q-1},i_1} \cdot (T_q)_{i_1,i_2,\dots,i_q} = M_{i_1,\dots,i_q}. \end{aligned}$$

The claim follows. $\square$

Example 27. Consider the binary SPN, whose graph is displayed in Fig. 9, with the (unique) ordering $(a_1, a_2, a_3, a_4, a_5)$. The activation tensors A_1 at node a_1, A_2 at node a_2, and so on, which depend on two parameters, are defined as follows:

$$A_1 = (\alpha, \beta), \qquad A_2 = A_4 = \begin{pmatrix} \alpha & \beta \\ \beta & \alpha \end{pmatrix}, \qquad A_3 = A_5 =$$



.

Then, the total tensor N is given by the BMP

$$N = \circ(T_{a_5}, T_{a_1}, T_{a_2}, T_{a_3}, T_{a_4}),$$

where $T_{a_1}, T_{a_2}, T_{a_3}, T_{a_4}, T_{a_5}$ are the dimension-5 tensors constructed according to Procedure 22. One computes:

$$T_{a_1} = f_{\{3,4,5\}}(b(A_1)) \quad T_{a_2} = f_{\{4,5\}}(b(A_2)), \quad T_{a_3} = f_{\{5\}}(b(A_3)),$$

$$T_{a_4} = b(f_{\{1,2\}}(A_4)), \quad T_{a_5} = f_{\{1,3\}}(A_5).$$

It follows that $T_{a_1}, T_{a_2}, T_{a_3}, T_{a_4}, T_{a_5}$ are given by

$$\begin{array}{llll}
(T_{a_1})_{0,0,0,0,0} = \alpha, & (T_{a_1})_{0,0,0,0,1} = \alpha, & (T_{a_1})_{0,0,0,1,0} = \alpha, & (T_{a_1})_{0,0,0,1,1} = \alpha, \\
(T_{a_1})_{0,0,1,0,0} = \alpha, & (T_{a_1})_{0,0,1,0,1} = \alpha, & (T_{a_1})_{0,0,1,1,0} = \alpha, & (T_{a_1})_{0,0,1,1,1} = \alpha, \\
T_{a_1})_{0,1,0,0,0} = 0, & (T_{a_1})_{0,1,0,0,1} = 0, & (T_{a_1})_{0,1,0,1,0} = 0, & (T_{a_1})_{0,1,0,1,1} = 0, \\
(T_{a_1})_{0,1,1,0,0} = 0, & (T_{a_1})_{0,1,1,0,1} = 0, & (T_{a_1})_{0,1,1,1,0} = 0, & (T_{a_1})_{0,1,1,1,1} = 0, \\
(T_{a_1})_{1,0,0,0,0} = 0, & (T_{a_1})_{1,0,0,0,1} = 0, & (T_{a_1})_{1,0,0,1,0} = 0, & (T_{a_1})_{1,0,0,1,1} = 0, \\
(T_{a_1})_{1,0,1,0,0} = 0, & (T_{a_1})_{1,0,1,0,1} = 0, & (T_{a_1})_{1,0,1,1,0} = 0, & (T_{a_1})_{1,0,1,1,1} = 0, \\
(T_{a_1})_{1,1,0,0,0} = \beta, & (T_{a_1})_{1,1,0,0,1} = \beta, & (T_{a_1})_{1,1,0,1,0} = \beta, & (T_{a_1})_{1,1,0,1,1} = \beta, \\
(T_{a_1})_{1,1,1,0,0} = \beta, & (T_{a_1})_{1,1,1,0,1} = \beta, & (T_{a_1})_{1,1,1,1,0} = \beta, & (T_{a_1})_{1,1,1,1,1} = \beta;
\end{array}$$

$$\begin{array}{llll}
(T_{a_2})_{0,0,0,0,0} = \alpha, & (T_{a_2})_{0,0,0,0,1} = \alpha, & (T_{a_2})_{0,0,0,1,0} = \alpha, & (T_{a_2})_{0,0,0,1,1} = \alpha, \\
(T_{a_2})_{0,0,1,0,0} = 0, & (T_{a_2})_{0,0,1,0,1} = 0, & (T_{a_2})_{0,0,1,1,0} = 0, & (T_{a_2})_{0,0,1,1,1} = 0, \\
(T_{a_2})_{0,1,0,0,0} = \beta, & (T_{a_2})_{0,1,0,0,1} = \beta, & (T_{a_2})_{0,1,0,1,0} = \beta, & (T_{a_2})_{0,1,0,1,1} = \beta, \\
(T_{a_2})_{0,1,1,0,0} = 0, & (T_{a_2})_{0,1,1,0,1} = 0, & (T_{a_2})_{0,1,1,1,0} = 0, & (T_{a_2})_{0,1,1,1,1} = 0, \\
(T_{a_2})_{1,0,0,0,0} = 0, & (T_{a_2})_{1,0,0,0,1} = 0, & (T_{a_2})_{1,0,0,1,0} = 0, & (T_{a_2})_{1,0,0,1,1} = 0, \\
(T_{a_2})_{1,0,1,0,0} = \beta, & (T_{a_2})_{1,0,1,0,1} = \beta, & (T_{a_2})_{1,0,1,1,0} = \beta, & (T_{a_2})_{1,0,1,1,1} = \beta, \\
(T_{a_2})_{1,1,0,0,0} = 0, & (T_{a_2})_{1,1,0,0,1} = 0, & (T_{a_2})_{1,1,0,1,0} = 0, & (T_{a_2})_{1,1,0,1,1} = 0, \\
(T_{a_2})_{1,1,1,0,0} = \alpha, & (T_{a_2})_{1,1,1,0,1} = \alpha, & (T_{a_2})_{1,1,1,1,0} = \alpha, & (T_{a_2})_{1,1,1,1,1} = \alpha;
\end{array}$$

$$\begin{array}{llll}
(T_{a_3})_{0,0,0,0,0} = \alpha, & (T_{a_3})_{0,0,0,0,1} = \alpha, & (T_{a_3})_{0,0,0,1,0} = 0, & (T_{a_3})_{0,0,0,1,1} = 0, \\
(T_{a_3})_{0,0,1,0,0} = \beta, & (T_{a_3})_{0,0,1,0,1} = \beta, & (T_{a_3})_{0,0,1,1,0} = 0, & (T_{a_3})_{0,0,1,1,1} = 0, \\
(T_{a_3})_{0,1,0,0,0} = \beta, & (T_{a_3})_{0,1,0,0,1} = \beta, & (T_{a_3})_{0,1,0,1,0} = 0, & (T_{a_3})_{0,1,0,1,1} = 0, \\
(T_{a_3})_{0,1,1,0,0} = \alpha, & (T_{a_3})_{0,1,1,0,1} = \alpha, & (T_{a_3})_{0,1,1,1,0} = 0, & (T_{a_3})_{0,1,1,1,1} = 0, \\
(T_{a_3})_{1,0,0,0,0} = 0, & (T_{a_3})_{1,0,0,0,1} = 0, & (T_{a_3})_{1,0,0,1,0} = \beta, & (T_{a_3})_{1,0,0,1,1} = \beta, \\
(T_{a_3})_{1,0,1,0,0} = 0, & (T_{a_3})_{1,0,1,0,1} = 0, & (T_{a_3})_{1,0,1,1,0} = \alpha, & (T_{a_3})_{1,0,1,1,1} = \alpha, \\
(T_{a_3})_{1,1,0,0,0} = 0, & (T_{a_3})_{1,1,0,0,1} = 0, & (T_{a_3})_{1,1,0,1,0} = \beta, & (T_{a_3})_{1,1,0,1,1} = \beta, \\
(T_{a_3})_{1,1,1,0,0} = 0, & (T_{a_3})_{1,1,1,0,1} = 0, & (T_{a_3})_{1,1,1,1,0} = \alpha, & (T_{a_3})_{1,1,1,1,1} = \alpha;
\end{array}$$

$$\begin{array}{llll}
(T_{a_4})_{0,0,0,0,0} = \alpha, & (T_{a_4})_{0,0,0,0,1} = 0, & (T_{a_4})_{0,0,0,1,0} = \beta, & (T_{a_4})_{0,0,0,1,1} = 0, \\
(T_{a_4})_{0,0,1,0,0} = \beta, & (T_{a_4})_{0,0,1,0,1} = 0, & (T_{a_4})_{0,0,1,1,0} = \alpha, & (T_{a_4})_{0,0,1,1,1} = 0, \\
(T_{a_4})_{0,1,0,0,0} = \alpha, & (T_{a_4})_{0,1,0,0,1} = 0, & (T_{a_4})_{0,1,0,1,0} = \beta, & (T_{a_4})_{0,1,0,1,1} = 0, \\
(T_{a_4})_{0,1,1,0,0} = \beta, & (T_{a_4})_{0,1,1,0,1} = 0, & (T_{a_4})_{0,1,1,1,0} = \alpha, & (T_{a_4})_{0,1,1,1,1} = 0, \\
(T_{a_4})_{1,0,0,0,0} = 0, & (T_{a_4})_{1,0,0,0,1} = \alpha, & (T_{a_4})_{1,0,0,1,0} = 0, & (T_{a_4})_{1,0,0,1,1} = \beta, \\
(T_{a_4})_{1,0,1,0,0} = 0, & (T_{a_4})_{1,0,1,0,1} = \beta, & (T_{a_4})_{1,0,1,1,0} = 0, & (T_{a_4})_{1,0,1,1,1} = \alpha, \\
(T_{a_4})_{1,1,0,0,0} = 0, & (T_{a_4})_{1,1,0,0,1} = \alpha, & (T_{a_4})_{1,1,0,1,0} = 0, & (T_{a_4})_{1,1,0,1,1} = \beta, \\
(T_{a_4})_{1,1,1,0,0} = 0, & (T_{a_4})_{1,1,1,0,1} = \beta, & (T_{a_4})_{1,1,1,1,0} = 0, & (T_{a_4})_{1,1,1,1,1} = \alpha;
\end{array}$$

$$\begin{array}{llll}
(T_{a_5})_{0,0,0,0,0} = \alpha, & (T_{a_5})_{0,0,0,0,1} = \beta, & (T_{a_5})_{0,0,0,1,0} = \beta, & (T_{a_5})_{0,0,0,1,1} = \alpha, \\
(T_{a_5})_{0,0,1,0,0} = \alpha, & (T_{a_5})_{0,0,1,0,1} = \beta, & (T_{a_5})_{0,0,1,1,0} = \beta, & (T_{a_5})_{0,0,1,1,1} = \alpha, \\
(T_{a_5})_{0,1,0,0,0} = \beta, & (T_{a_5})_{0,1,0,0,1} = \alpha, & (T_{a_5})_{0,1,0,1,0} = \beta, & (T_{a_5})_{0,1,0,1,1} = \alpha, \\
(T_{a_5})_{0,1,1,0,0} = \beta, & (T_{a_5})_{0,1,1,0,1} = \alpha, & (T_{a_5})_{0,1,1,1,0} = \beta, & (T_{a_5})_{0,1,1,1,1} = \alpha, \\
(T_{a_5})_{1,0,0,0,0} = \alpha, & (T_{a_5})_{1,0,0,0,1} = \beta, & (T_{a_5})_{1,0,0,1,0} = \beta, & (T_{a_5})_{1,0,0,1,1} = \alpha, \\
(T_{a_5})_{1,0,1,0,0} = \alpha, & (T_{a_5})_{1,0,1,0,1} = \beta, & (T_{a_5})_{1,0,1,1,0} = \beta, & (T_{a_5})_{1,0,1,1,1} = \alpha, \\
(T_{a_5})_{1,1,0,0,0} = \beta, & (T_{a_5})_{1,1,0,0,1} = \alpha, & (T_{a_5})_{1,1,0,1,0} = \beta, & (T_{a_5})_{1,1,0,1,1} = \alpha, \\
(T_{a_5})_{1,1,1,0,0} = \beta, & (T_{a_5})_{1,1,1,0,1} = \alpha, & (T_{a_5})_{1,1,1,1,0} = \beta, & (T_{a_5})_{1,1,1,1,1} = \alpha;
\end{array}$$

and, by computing the BMP $\circ(T_{a_5}, T_{a_1}, T_{a_2}, T_{a_3}, T_{a_4})$, we obtain the correct evaluation of N, whose entries are

$$
\begin{array}{cccc}
N_{0,0,0,0,0} = \alpha^5 & N_{0,0,0,0,1} = \alpha^4\beta & N_{0,0,0,1,0} = \alpha^3\beta^2 & N_{0,0,0,1,1} = \alpha^4\beta \\
N_{0,0,1,0,0} = \alpha^3\beta^2 & N_{0,0,1,0,1} = \alpha^2\beta^3 & N_{0,0,1,1,0} = \alpha^3\beta^2 & N_{0,0,1,1,1} = \alpha^4\beta \\
N_{0,1,0,0,0} = \alpha^2\beta^3 & N_{0,1,0,0,1} = \alpha^3\beta^2 & N_{0,1,0,1,0} = \alpha\beta^4 & N_{0,1,0,1,1} = \alpha^2\beta^3 \\
N_{0,1,1,0,0} = \alpha^2\beta^3 & N_{0,1,1,0,1} = \alpha^3\beta^2 & N_{0,1,1,1,0} = \alpha^3\beta^2 & N_{0,1,1,1,1} = \alpha^4\beta \\
N_{1,0,0,0,0} = \alpha^2\beta^3 & N_{1,0,0,0,1} = \alpha\beta^4 & N_{1,0,0,1,0} = \beta^5 & N_{1,0,0,1,1} = \alpha\beta^4 \\
N_{1,0,1,0,0} = \alpha^2\beta^3 & N_{1,0,1,0,1} = \alpha\beta^4 & N_{1,0,1,1,0} = \alpha^2\beta^3 & N_{1,0,1,1,1} = \alpha^3\beta^2 \\
N_{1,1,0,0,0} = \alpha^2\beta^3 & N_{1,1,0,0,1} = \alpha^3\beta^2 & N_{1,1,0,1,0} = \alpha\beta^4 & N_{1,1,0,1,1} = \alpha^2\beta^3 \\
N_{1,1,1,0,0} = \alpha^2\beta^3 & N_{1,1,1,0,1} = \alpha^3\beta^2 & N_{1,1,1,1,0} = \alpha^3\beta^2 & N_{1,1,1,1,1} = \alpha^4\beta.
\end{array}
$$

Observe that, in the projective space of tensors of type $2\times2\times2\times2\times2$, the tensors defined by N, when α, β vary, describe a rational normal curve of degree 5 (highly degenerate).

3.3. Software

A Python software has been implemented for computing the BMP of tensors of arbitrary dimension d, as in Definition 8, and for performing the procedure described in Procedure 22. Given an SPN, the software performs the blow and forget operations for the tensor of each node, computing the transformed tensors T_{a_i}. By Theorem 23, the BMP of these tensors yields the total tensor of the network. The only data required to run the code are the adjacency matrix of a DAG with a source and a sink, and the list of activation tensors assigned to the nodes (compatible with their number of parents); alternatively, random tensors can be generated according to the adjacency matrix of the DAG, for any number of nodes. All the Examples presented in this paper and BPMs for random networks up to 12 nodes can be computed in less than 1 second on a Macbook Pro with M1 Pro processor.

The software is open-source and freely available at: https://github.com/MarzialiS/BMP-Network, along with a notebook to run all the Examples presented in this work.

Acknowledgments

The authors want to warmly thank Luke Oeding, who introduced them to the early works of Bhattacharya and Mesner, Fulvio Gesmundo, and Edinah Gnang, for useful discussions on the subject of the paper. G.A.D. acknowledges the support provided by the European Union - NextGenerationEU, in the framework of the iNEST - Interconnected Nord-Est Innovation Ecosystem (iNEST ECS00000043 – CUP G93C22000610007) project and its CC5 Young Researchers initiative. The views and opinions expressed are solely those of the author and do not necessarily reflect those of the European Union, nor can the European Union be held responsible for him. In addition, G.A.D. would like to acknowledge INdAM–GNCS. L.C. is member of INdAM-GNSAGA.

References

[1] Prabir Bhattacharya, A new 3-D transform using a ternary product, IEEE Trans. Signal Process. 43 (12) (1995) 3081–3084.

[2] Cristiano Bocci, Luca Chiantini, et al., An Introduction to Algebraic Statistics with Tensors, vol. 1, Springer, 2019.
[3] Reinhard Diestel, Graph Theory, Sixth Edition, Graduate Texts in Mathematics, vol. 173, Springer-Verlag Heidelberg, 2025.
[4] Edinah K. Gnang, A combinatorial approach to hypermatrix algebra, arXiv preprint, arXiv:1403.3134, 2014.
[5] Edinah K. Gnang, Yuval Filmus, On the spectra of hypermatrix direct sum and Kronecker products constructions, Linear Algebra Appl. 519 (2017) 238–277.
[6] Edinah K. Gnang, Yuval Filmus, On the Bhattacharya-Mesner rank of third order hypermatrices, Linear Algebra Appl. 588 (2020) 391–418.
[7] Leslie Hogben, Handbook of Linear Algebra, CRC Press, 2006.
[8] Joseph M. Landsberg, Tensors: Geometry and Applications: Geometry and Applications, vol. 128, American Mathematical Soc., 2011.
[9] Sara Marziali, Boolean operators and neural networks, Ann. Univ. Ferrara 70 (4) (2024/10/01) 1767–1783.
[10] Dale M. Mesner, Prabir Bhattacharya, Association schemes on triples and a ternary algebra, J. Comb. Theory, Ser. A 55 (2) (1990) 204–234.
[11] Dale M. Mesner, Prabir Bhattacharya, A ternary algebra arising from association schemes on triples, J. Algebra 164 (3) (1994) 595–613.
[12] Fan Tian, Misha E. Kilmer, Eric Miller, Abani Patra, Tensor BM-decomposition for compression and analysis of spatio-temporal third-order data, arXiv preprint, arXiv:2306.09201, 2023.