Towards Mastering Tensor Networks: A Comprehensive Guide

Beheshteh T. Rakhshan
Guillaume Rabusseau

Contents

Notations

$\boldsymbol{\mathcal{A}}$	General tensor containing $n \geq 0$ modes
$\boldsymbol{\mathcal{A}} \in \mathbb{R}^{d_1 \times d_2 \times d_3}$	A 3rd-order tensor $\boldsymbol{\mathcal{A}}$ with shape (d_1, d_2, d_3)
$\boldsymbol{\mathcal{A}}_{i,j,k}$	The (i, j, k)'th element of a 3rd-order tensor $\boldsymbol{\mathcal{A}}$
$\boldsymbol{\mathcal{A}}_{i,:,k}$	Mode-2 fiber of a 3rd-order tensor $\boldsymbol{\mathcal{A}}$, equivalent to a vector $v \in \mathbb{R}^{d_2}$
$\boldsymbol{\mathcal{A}}_{(2)}$	Mode-2 flattening of a (3rd-order) tensor $\boldsymbol{\mathcal{A}}$, equivalent to a matrix $M \in \mathbb{R}^{d_2 \times d_1 d_3}$
$\boldsymbol{\mathcal{A}} \circ \boldsymbol{\mathcal{B}}$	The tensor product (equiv. outer product) of tensors $\boldsymbol{\mathcal{A}}$ and $\boldsymbol{\mathcal{B}}$
$\boldsymbol{\mathcal{A}} \simeq \boldsymbol{\mathcal{B}}$	An isomorphism between two tensors $\boldsymbol{\mathcal{A}}$ with $\boldsymbol{\mathcal{B}}$ with compatible shapes
$\mathrm{TN}(G, R)$	Space of all tensors expressable as a tensor network with graph G and rank function R
$\mathrm{TN}(G, R, \mathcal{G})$	A tensor network with graph G, rank function R, and core assignment function $\mathcal{G}$
$\langle \boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{B}} \rangle$	Inner product between tensors $\boldsymbol{\mathcal{A}}$ and $\boldsymbol{\mathcal{B}}$
$\Vert \boldsymbol{\mathcal{A}} \Vert_F$	Frobenius norm of the tensor $\boldsymbol{\mathcal{A}}$
$\boldsymbol{\mathcal{A}} \otimes \boldsymbol{\mathcal{B}}$	The Kronecker product of tensors $\boldsymbol{\mathcal{A}}$ and $\boldsymbol{\mathcal{B}}$
$\mathbf{A}$	Matrix; A rank-2 tensor better: (tensor of order 2)
$\mathbf{A}_{ij}$	The ij-th element of the matrix $\mathbf{A}$
$\mathbf{A}^{-1}$	The inverse matrix of matrix $\mathbf{A}$
$\mathbf{A}^{\mathsf{T}}$	The transposed matrix of matrix $\mathbf{A}$
$\mathbf{I}$	The identity matrix
$\mathbf{A} \odot \mathbf{B}$	The Khatri-Rao product of matrices $\mathbf{A}$ and $\mathbf{B}$
$\mathbf{A} * \mathbf{B}$	The Hadamard product of matrices $\mathbf{A}$ and $\mathbf{B}$
$\mathbf{a}$	Column vector; A rank-1 tensor
$\mathbf{a}_i$	The i-th element of vector $\mathbf{a}$
$\mathrm{vec}(\mathbf{A})$	Column vector obtained by concatenating the columns of the matrix $\mathbf{A}$
a	Scalar; A rank-0 tensor

1 Introduction

To talk about tensor-based methods, we should start with the definition of a tensor. We define an N-th order tensor $\mathcal{T} \in \mathbb{R}^{\mathbf{d}} = \mathbb{R}^{d_1 \times \cdots \times d_N}$ to be a collection of indexed coefficients $\mathcal{T}_{i_1,\ldots,i_N} \in \mathbb{R}$, referred to as the *elements* of $\mathcal{T}$, where each index i_j is associated with the j-th *mode* of $\mathcal{T}$ and varies over the set $[d_j] = \{1, 2, \ldots, d_j\}$. The tuple $\mathbf{d} = (d_1, \ldots, d_N)$ is referred to as the *shape* of $\mathcal{T}$, with $d_j \in [d_j]$ referred to as the *dimension* of the j-th mode of $\mathcal{T}$. Any mode dimension with $d_j = 1$ is referred to as a *singleton mode*, and can be reversably added or removed from a given tensor. The collection of all tensors with a given shape $\mathbf{d}$ form a vector space of dimension $D = \prod_{j=1}^{N} d_j$, where addition of tensors and multiplication by scalars is defined elementwise as $(\mathcal{T} + \mathcal{T}')_{i_1,\ldots,i_N} = \mathcal{T}_{i_1,\ldots,i_N} + \mathcal{T}'_{i_1,\ldots,i_N}$ and $(c\mathcal{T})_{i_1,\ldots,i_N} = c\mathcal{T}_{i_1,\ldots,i_N}$. This vector space is endowed with the inner product $\langle \mathcal{T}, \mathcal{T}' \rangle := \sum_{i_1=1}^{d_1} \cdots \sum_{i_N=1}^{d_N} \mathcal{T}_{i_1,\ldots,i_N} \mathcal{T}'_{i_1,\ldots,i_N}$, as well as the p-norms $\|\mathcal{T}\|_p := \left(\sum_{i_1=1}^{d_1} \cdots \sum_{i_N=1}^{d_N} |\mathcal{T}_{i_1,\ldots,i_N}|^p \right)^{1/p}$ for any $p \geq 1$. We will refer to the case of $p = 2$ as the *Frobenius norm*, denoted by $\|\mathcal{T}\|_F = \|\mathcal{T}\|_2 = \sqrt{\langle \mathcal{T}, \mathcal{T} \rangle}$

N-th order tensors are a natural generalization of vectors and matrices (corresponding to the cases of $N = 1$ and $N = 2$, respectively), and it is useful to reason about any new tensor-based concept by first considering its restriction to these simpler cases. Taking the concepts defined above as an example, with $N = 2$ the two tensor modes correspond to the rows ($j = 1$) and columns ($j = 2$) of a matrix $\mathbf{M} \in \mathbb{R}^{d_1 \times d_2}$, and the presence of a singleton mode means that we can interpret the corresponding matrix as a vector, with is shaped as either a single row ($d_1 = 1$) or column ($d_2 = 2$). In general though, many simple concepts defined for vectors or matrices will tend to admit several different generalizations when extended to general tensors, and we will refer to tensors with order $N > 2$ as *higher-order* tensors.

Although we have defined a tensor $\mathcal{T}$ as the entirety of its elements, this definition should not be taken as a prescription for representing or parameterizing general tensors. This point is important in the context of s on a computer, where the naive representation of a tensor $\mathcal{T} \in \mathbb{R}^{\mathbf{d}}$ as the concatenation of all of its elements $\mathcal{T}_{i_1,\ldots,i_N}$ is referred to as a *dense representation*. Dense representations are commonplace for matrices and vectors, but rapidly become intractable for higher-order tensors, owing to the exponential storage cost of $D = \prod_{j=1}^{N} d_j \geq d^N$, where $d := \min_j d_j$. This survey is primarily concerned with more efficient *implicit* representations of tensors, where a small collection of parameters is sufficient to completely describe a tensor with a much larger number of elements. The details of these implicit representations can vary considerably, but the bare minimum needed from any such representation is the existence of a (parameter-dependent) map $\mathcal{T}_{-} : [d_1] \times \cdots \times [d_N] \to \mathbb{R}$, sending index tuples $(i_1, \ldots, i_N)$ to associated tensor elements $\mathcal{T}_{i_1,\ldots,i_N} := \mathcal{T}_{-}(i_1, \ldots, i_N)$. Such a map ensures that the implicit representation does indeed uniquely specify a tensor.

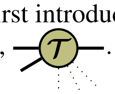
A simple means of implicitly representing higher-order tensors is via the *outer product*, where a pair of tensors $\mathcal{T} \in \mathbb{R}^{d_1 \times \cdots \times d_N}$, $\mathcal{T}' \in \mathbb{R}^{d'_1 \times \cdots \times d'_{N'}}$ of orders N and N' are combined into a single tensor $\mathcal{T} \circ \mathcal{T}' \in \mathbb{R}^{d_1 \times \cdots \times d_N \times d'_1 \times \cdots \times d'_{N'}}$ of order $N + N'$, with elements $(\mathcal{T} \circ \mathcal{T}')_{i_1,\ldots,i_N,i'_1,\ldots,i'_{N'}} = \mathcal{T}_{i_1,\ldots,i_N} \mathcal{T}'_{i'_1,\ldots,i'_{N'}}$. Applying this to N vectors[1] $\{\mathbf{v}^{(j)} \in \mathbb{R}^{d_j}\}_{j=1}^{N}$ gives the *rank-1 tensor* $\mathcal{T} = \mathbf{v}^{(1)} \circ \cdots \circ \mathbf{v}^{(N)}$ use only $\mathcal{O}(dN)$ parameters to describe $\mathcal{O}(d^N)$ tensor elements, via the map $\mathcal{T}_{-} : (i_1, \ldots, i_N) \mapsto \prod_{j=1}^{N} \mathbf{v}^{(j)}_{i_j}$. More generally, we define the *rank* of a general tensor $\mathcal{T}$ to be the smallest number r that allows $\mathcal{T}$ to be written as $\mathcal{T} = \sum_{\alpha=1}^{r} \mathcal{T}_{\alpha}$, where each $\mathcal{T}_{\alpha}$ is a rank-1 tensor. The parameterization of tensors as the sum of r rank-1 tensors is referred to as the *CP decomposition*, and is comparable in efficiency to a rank-1 parameterization, requiring only $\big(rdN\big)$ parameters to describe $\mathcal{O}(d^N)$ tensor elements. We will discuss the CP decomposition in more detail in later sections.

2 Tensor Networks Basics

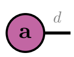
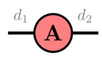
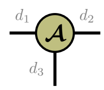
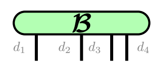
As their order increases, representing and working with tensors becomes more complicated. Tensor networks provide an efficient framework for working with these high-order objects, simplifying both their representation and analysis. The graphical notation of tensor networks offers an intuitive way to visualize and simplify complex tensor operations [Orús, 2014, Biamonte and Bergholm, 2017]. In this section, we introduce basic alphabets for tensors and common operations on them using the language of tensor networks.

[1]The outer product is associative, so there is no ambiguity in applying $\circ$ to more than two tensors.

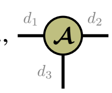
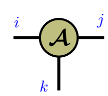
2.1 What are Tensor Networks?

As their name suggests, Tensor Networks (TNs) are simply tensors connected to each other to form a network. The graphical notation for tensor networks was first introduced by [Penrose et al., 1971] where tensors are represented by shapes with legs (edges) attached to them, i.e., . Tensors can have different shapes, such as rectangles, triangles, or circles, and can be in various colors. Each leg of a tensor is called an order. An N-th order tensor $\boldsymbol{\mathcal{T}} \in \mathbb{R}^{d_1 \times \cdots \times d_N}$ is a multidimensional array of scalars ($\boldsymbol{\mathcal{T}}_{i_1,\ldots,i_N}, i_n \in [d_n], n \in [N]$), and each axis represents a mode (order, dimension) of a tensor. A tensor $\boldsymbol{\mathcal{T}} \in \mathbb{R}^{d_1 \times \ldots d_N}$ contains, $d_1 \cdots d_N$ scalars. Tensors can also be seen as a generalization of vectors and matrices to higher-order arrays. As the order increases, representing these arrays becomes more challenging. *Tensor networks* are tools that provide a simple and intuitive way to represent and work with these higher-order objects. Complex operations on tensors can be represented more easily with graphical notations of tensor networks [Biamonte and Bergholm, 2017, Orús, 2014].

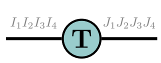
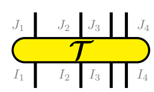
Tensor Network Nodes In tensor networks graphical notations, scalars are shapes with no edges, vectors with a single edge, matrices with two edges, and so on, e.g.,

 $\in \mathbb{R}$,  $\in \mathbb{R}^d$,  $\in \mathbb{R}^{d_1 \times d_2}$,  $\in \mathbb{R}^{d_1 \times d_2 \times d_3}$,  $\in \mathbb{R}^{d_1 \times d_2 \times d_3 \times d_4}$.

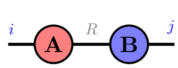
are scalar, vector, matrix, and tensors, respectively. Scalars are also called zero-order tensors, vectors first-order tensors, matrices second-order tensors, and for $N \geq 3$ the object is called a high-order tensor. Throughout this manuscript,

- Tensors are represented by colored circles called nodes, where the colors have no specific meaning.
- Size of each dimension is depicted by gray letters positioned at the top of the edges, e.g.,  $\in \mathbb{R}^{d_1 \times d_2 \times d_3}$, is a 3rd order tensor of size $d_1 \times d_2 \times d_3$.
- Indices are presented by blue letters at the very end of edges, i.e., $\boldsymbol{\mathcal{A}}_{i,j,k} =$  is an element of a 3rd order tensor.

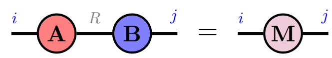
Note that large matrices can sometimes be viewed as high-order tensors through reshaping, e.g.,

$\mathbf{T} =$  $\in \mathbb{R}^{I_1 I_2 I_3 I_4 \times J_1 J_2 J_3 J_4} \equiv \boldsymbol{\mathcal{T}} =$  $\in \mathbb{R}^{I_1 \times I_2 \times I_3 \times I_4 \times J_1 \times J_2 \times J_3 \times J_4}$.

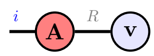
Tensor Network Edges In tensor network diagram, legs are of two types: contracted legs (those connecting two tensors) and un-contracted legs, also called free legs, with one dangling end (i.e., a leg that is not connected to any other tensor). As mentioned above, un-contracted legs correspond to free indices: the number of free legs indicates the order of a tensor: scalars have no free legs, vectors have one, matrices have two, and higher-order tensors have three or more. Contracted legs represent contractions: tensors can be connected along legs of the same sizes, which represents a summation (contraction) the connected modes. We use the terms summation and contraction interchangeably. The most common contraction operation is *matrix multiplication*. For two matrices $\mathbf{A} \in \mathbb{R}^{d_1 \times R}, \mathbf{B} \in \mathbb{R}^{R \times d_2}$, their matrix product is obtained by contraction:

$(\mathbf{AB})_{ij} =$  $= \sum_{r=1}^{R} \mathbf{A}_{ir}\mathbf{B}_{rj}, \quad \text{for } i \in [d_1], j \in [d_2]\ .$ (1)

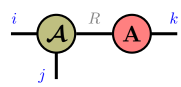
In the diagram above, we can see that there is a connection between the legs with the same size R. This is consistent with matrix multiplication where there is a sum over indices with the same dimension. Finally as the resulting diagram has two free legs, it represents a matrix. Here the final object is a matrix which can be represented as single node, demonstrating how nodes can be merged in tensor networks:

$$\mathbf{M}_{ij} = (\mathbf{AB})_{ij} =$$ 

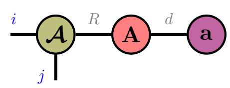
Contraction between two matrices can also be extended to matrix-vector, tensor-matrix, and tensor-matrix-vector products, e.g.,

$$\mathbf{A} \in \mathbb{R}^{d_1 \times d}, \mathbf{a} \in \mathbb{R}^{d}$$ 

$$= \sum_{r=1}^{R} \mathbf{A}_{ir}\mathbf{v}_r, \tag{2}$$

$$\boldsymbol{\mathcal{A}} \in \mathbb{R}^{d_1 \times R \times d_2}, \mathbf{A} \in \mathbb{R}^{R \times d_3} \ :$$ 

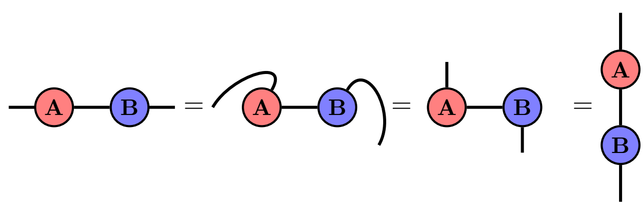
$$= \sum_{r=1}^{R} \boldsymbol{\mathcal{A}}_{ijr}\mathbf{A}_{rk}, \tag{3}$$

$$\boldsymbol{\mathcal{A}} \in \mathbb{R}^{d_1 \times R \times d_2}, \mathbf{A} \in \mathbb{R}^{R \times d_3}, \mathbf{a} \in \mathbb{R}^{d_3} \ :$$ 

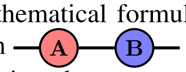
$$= \sum_{r=1}^{R}\sum_{d=1}^{d_3} \boldsymbol{\mathcal{A}}_{ijr}\mathbf{A}_{rd}\mathbf{a}_d. \tag{4}$$

As we can see in all diagrams above, edges between two nodes represent summations. They also indicate that two tensors share dimensions of the same size. According to the settings in 2.1, eqn. (2) represents a vector, eqn. (3) and eqn. (4) represent a matrix.

Note. Tensor networks are simple to work with because there is no strict rules for representing legs and nodes. In tensor network diagrams, legs can be depicted in any order, and nodes can be positioned arbitrarily in the space. For example, when translating matrix multiplication into a tensor network diagram, the key feature is to ensure the indices of the corresponding legs are consistent, e.g., for $\mathbf{A} \in \mathbb{R}^{d_1 \times R}, \mathbf{B} \in \mathbb{R}^{R \times d_2}$ and $i \in [d_1], j \in [d_2]$ all diagrams below illustrate the same matrix multiplication:

$$\mathbf{AB} =$$ 

Note that, because of the sizes of $\mathbf{A} \in \mathbb{R}^{d_1 \times R}$ and $\mathbf{B} \in \mathbb{R}^{R \times d_2}$, there is only one way to do the contraction between the two, hence there is no ambiguity in the diagrams above.

On the other hand, translating tensor network diagrams into mathematical formulations can lead to multiple interpretations. For example, the following matrix multiplication diagram 

can be interpreted differently depending on what we choose the shapes of $\mathbf{A}$, $\mathbf{B}$ and the resulting matrix to be:

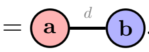
- if $\mathbf{A} \in \mathbb{R}^{d_1 \times R}, \mathbf{B} \in \mathbb{R}^{R \times d_2}$ and the result is of size $d_1 \times d_2$, then the diagram represents the product $\mathbf{AB}$,
- if $\mathbf{A} \in \mathbb{R}^{R \times d_1}, \mathbf{B} \in \mathbb{R}^{R \times d_2}$ and the result is of size $d_1 \times d_2$, then the diagram represents the product $\mathbf{A}^{\top}\mathbf{B}$,
- if $\mathbf{A} \in \mathbb{R}^{R \times d_1}, \mathbf{B} \in \mathbb{R}^{R \times d_2}$ and the result is of size $d_2 \times d_1$, then the diagram represents the product $\mathbf{B}^{\top}\mathbf{A}$,
- ...

Therefore, one should be mindful when translating tensor network diagrams into mathematical formulations.

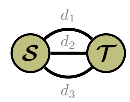
2.2 Inner Product, Outer Product, Trace and Norm

In this section, we introduce inner product, outer product, trace and norm in tensor network diagrams.

Inner product The inner product of two N-th order tensors $\boldsymbol{\mathcal{S}}, \boldsymbol{\mathcal{T}} \in \mathbb{R}^{d_1\times\cdots\times d_N}$ is the sum of the products of their entries, i.e., $\sum_{i_1=1}^{d_1}\cdots\sum_{i_N=1}^{d_N} \boldsymbol{\mathcal{S}}_{i_1\dots i_N}\boldsymbol{\mathcal{T}}_{i_1\dots i_N}$. In a tensor network diagram, the summation over all dimensions is obtained by connecting all the legs of the two tensors. This results in a tensor network with no free legs, representing a scalar. For example, for vectors $\mathbf{a}\in\mathbb{R}^{d\times 1}, \mathbf{b}\in\mathbb{R}^{d\times 1}$ we have

$$\langle \mathbf{a}, \mathbf{b}\rangle = \sum_{i=1}^{d}\mathbf{a}_i\mathbf{b}_i = \tag{5}$$


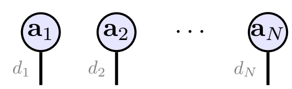
The inner product of two third-order tensors can be depicted by

$$\boldsymbol{\mathcal{S}}, \boldsymbol{\mathcal{T}} \in \mathbb{R}^{d_1\times d_2\times d_3},$$

$$= \langle \boldsymbol{\mathcal{S}}, \boldsymbol{\mathcal{T}}\rangle = \sum_{i_1=1}^{d_1}\sum_{i_2=1}^{d_2}\sum_{i_3=1}^{d_3}\boldsymbol{\mathcal{S}}_{i_1i_2i_3}\boldsymbol{\mathcal{T}}_{i_1i_2i_3}.$$

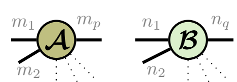
Outer product The outer product is an operation between any number of tensors. For example, the outer product of N vectors, $\mathbf{a}_1\in\mathbb{R}^{d_1},\cdots,\mathbf{a}_N\in\mathbb{R}^{d_N}$ is the tensor of order N defined by

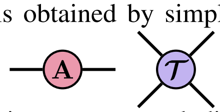
$$(\mathbf{a}_1\circ\mathbf{a}_2\circ\cdots\circ\mathbf{a}_N)_{i_1,i_2,\cdots,i_n} = (\mathbf{a}_1)_{i_1}(\mathbf{a}_2)_{i_2}\cdots(\mathbf{a}_N)_{i_N} \text{ for all } i_1\in[d_1],\cdots,i_N\in[d_N].$$

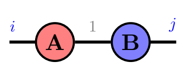
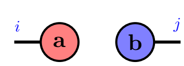
Such a tensor is called a **rank one** tensor. The diagram below illustrates the outer product of N vectors:

$$\mathbf{a}_1\circ\mathbf{a}_2\circ\cdots\circ\mathbf{a}_N =$$

$$\in \mathbb{R}^{d_1\times\cdots\times d_N}.$$

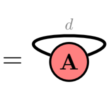
As a special case, for $N=2$, we have $\mathbf{a}_1\mathbf{a}_2^\top\in\mathbb{R}^{d_1\times d_2}$. As we see in the outer product, there are no shared edges, which means there is no summation in this product. The notion of outer product can be extended to tensors with more than one mode. Let $\boldsymbol{\mathcal{A}}\in\mathbb{R}^{m_1\times\cdots\times m_p}$ and $\boldsymbol{\mathcal{B}}\in\mathbb{R}^{n_1\times\cdots\times n_q}$. The outer product of $\boldsymbol{\mathcal{A}}$ and $\boldsymbol{\mathcal{B}}$ is the tensor of order $p+q$ defined by

$$\boldsymbol{\mathcal{A}}\circ\boldsymbol{\mathcal{B}} =$$

$$\in\mathbb{R}^{m_1\times\cdots\times m_p\times n_1\times\cdots\times n_q}.$$

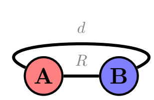
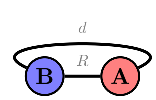
More generally, for any arbitrary number of tensors with arbitrary orders, the tensor network diagram of their outer product is obtained by simply placing them next to each other, e.g., for $\mathbf{A}\in\mathbb{R}^{m\times n}$ and $\boldsymbol{\mathcal{T}}\in\mathbb{R}^{d_1\times d_2\times d_3\times d_4}$ $\mathbf{A}\circ\boldsymbol{\mathcal{T}} =$  $\in\mathbb{R}^{m\times n\times d_1\times\cdots\times d_4}$ and element-wise is $(\mathbf{A}\circ\boldsymbol{\mathcal{T}})_{i_1i_2j_1j_2j_3j_4} = \mathbf{A}_{i_1i_2}\boldsymbol{\mathcal{T}}_{j_1\cdots j_4}$.
Note that in tensor network diagrams legs of size one means there is no edge in the tensor diagrams corresponding to those legs (e.g., vectors). E.g., in the matrix multiplication, if the contraction edge is of size one, it is equivalent to the outer product of vectors.

$$\mathbf{A}\in\mathbb{R}^{p\times 1},\quad \mathbf{B}\in\mathbb{R}^{1\times q},\quad (\mathbf{AB})_{ij} =$$

$$= \sum_{k=1}^{1}\mathbf{A}_{i1}\mathbf{B}_{1j} =$$

$$= (\mathbf{a}\circ\mathbf{b})_{ij}.$$

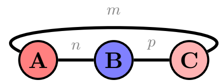
Trace. The trace operation is a special tensor contraction for square matrices, represented by loops (self-edges) connecting the two legs of the matrix.

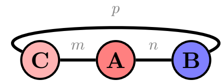
$$\mathbf{A}\in\mathbb{R}^{d\times d},\qquad \mathrm{tr}(\mathbf{A}) = \sum_{i=1}^{d}\mathbf{A}_{ii} =$$


Since there are no free legs, it is consistent with the fact that the trace is a scalar. Tensor networks diagrams offers a very simple proof of the invariance of the trace under cyclic permutation:

$$\mathbf{A}\in\mathbb{R}^{d\times R}, \mathbf{B}\in\mathbb{R}^{R\times d},\quad \mathrm{tr}(\mathbf{AB}) = \sum_{i=1}^{R}\sum_{j=1}^{d}\mathbf{A}_{ij}\mathbf{B}_{ij} =$$

$$=$$

$$= \mathrm{tr}(\mathbf{BA}),$$

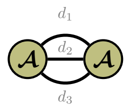
and more generally for three matrices:

$$\mathbf{A} \in \mathbb{R}^{m\times n}, \mathbf{B} \in \mathbb{R}^{n\times p}, \mathbf{C} \in \mathbb{R}^{p\times m}, \quad \mathrm{tr}(\mathbf{ABC}) =$$ 

$$=$$ 

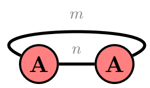
$$= \mathrm{tr}(\mathbf{CAB}).$$

Note that the equality between the two tensor network diagrams is a trivial one: we just changed the nodes' position without changing the underlying graph's structure. We thus simply interpreted the same diagram in two ways, leading to a less trivial equality between $\mathrm{tr}(\mathbf{ABC}) = \mathrm{tr}(\mathbf{CAB})$. **Tensor Norm.** The Frobenius norm of a tensor $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{d_1\times\cdots\times d_N}$ is the square root of the sum of the squares of all its elements. In tensor networks, the norm of a 3rd-order tensor is represented as

$$\boldsymbol{\mathcal{A}} \in \mathbb{R}^{d_1\times d_2\times d_3},$$ 

$$= \langle \boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{A}} \rangle = \sum_{i_1=1}^{d_1}\sum_{i_2=1}^{d_2}\sum_{i_3=1}^{d_3} \boldsymbol{\mathcal{A}}^2_{i_1 i_2 i_3} = \|\boldsymbol{\mathcal{A}}\|_F^2.$$

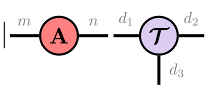
where tensor norm can be seen as a special case of the inner product of a tensor with itself. It is also analogous to the matrix Frobenius norm:

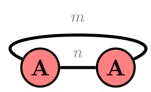
$$\mathbf{A} \in \mathbb{R}^{m\times n},$$ 

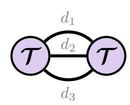
$$= \mathrm{tr}(\mathbf{A}\mathbf{A}^\top) = \mathrm{tr}(\mathbf{A}^\top\mathbf{A}) = \langle \mathbf{A}, \mathbf{A} \rangle = \|\mathbf{A}\|_F^2.$$

Below is an example of proving one of the norm and outer product identities with tensor network diagrams:

Example. Given $\mathbf{A} \in \mathbb{R}^{m\times n}$ and $\boldsymbol{\mathcal{T}} \in \mathbb{R}^{d_1\times d_2\times d_3}$, we show how $\|\mathbf{A} \circ \boldsymbol{\mathcal{T}}\|_F^2$ can be computed using tensor diagrams:

$$\|\mathbf{A} \circ \boldsymbol{\mathcal{T}}\|_F^2 = \|$$ 

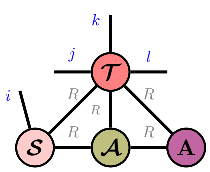
$$\|_F^2 =$$ 

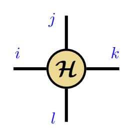
$$\cdot$$ 

$$= \|\mathbf{A}\|_F^2 \|\boldsymbol{\mathcal{T}}\|_F^2.$$

As we can see the norm of the outer product of tensors is the product of their norms.

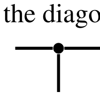
Note. Tensor network graphical notations are very useful for presenting more complicated operations on tensors. For example, we can contract two or more tensors over the same-size edges to produce a new tensor (Einsum operation), e.g.,

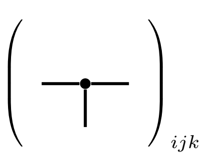


$$= \sum_{r_1=1}^{R}\sum_{r_2=1}^{R}\sum_{r_3=1}^{R}\sum_{r_4=1}^{R}\sum_{r_5=1}^{R} \boldsymbol{\mathcal{S}}_{ir_1r_2}\boldsymbol{\mathcal{A}}_{r_2r_3r_5}\boldsymbol{\mathcal{T}}_{iklr_1r_3r_4}\mathbf{A}_{r_4r_5} =$$ 

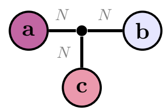
$$. \tag{6}$$

2.3 Copy Tensors (Hyperedges)

One sometimes nees to represent contractions between more than two indices. Consider for example the following contraction between there vectors to obtain a scalar: $\sum_{i=1}^{N} \mathbf{a}_i\mathbf{b}_i\mathbf{c}_i$. This operation can be depicted in tensor networks using a special tensor called *copy tensor* (also known as spider tensor). The copy tensor is equivalent to a Kronecker delta, i.e. a hyper-diagonal tensor with ones on the diagonal and zeros elsewhere, and is represented as a black dot in tensor network diagrams. E.g., the copy tensor  is defined element-wise as

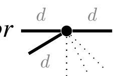

$$= \delta_{ijk} = \begin{cases} 1 & \text{if } i = j = k \\ 0 & \text{otherwise} \end{cases}.$$

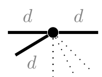
Using the copy tensor, the 3-way contraction mentioned above can be represented as

$$\sum_{i=1}^{N} \mathbf{a}_i \mathbf{b}_i \mathbf{c}_i =$$ 

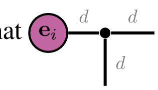
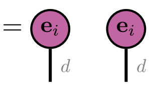
$$= \sum_{i,j,k=1}^{N} \delta_{ijk} \mathbf{a}_i \mathbf{b}_j \mathbf{c}_k.$$

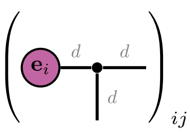
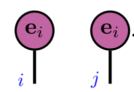
We now give a more formal definition of the copy tensor.

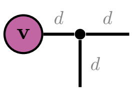
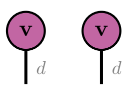
Definition 1. *The N-th order copy tensor*  *is the tensor of shape* $\underbrace{d \times d \cdots \times d}_{N \text{ times}}$ *defined by*


$$= \sum_{i=1}^{N} \underbrace{\mathbf{e}_i \circ \mathbf{e}_i \circ \ldots \circ \mathbf{e}_i}_{N \text{ times}},$$

where $\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_N \in \mathbb{R}^N$ *are the vectors of the canonical basis.*

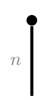
The name copy tensor comes from the fact that one can "copy" the canonical vectors by contraction with the copy tensor. Indeed, let $\mathbf{e}_i \in \mathbb{R}^d$ be a canonical basis vector, then one can check that  $=$  . Here is a shot proof of this fact:

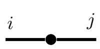

$$= \sum_s (\mathbf{e}_i)_s \delta_{sjk} = \sum_s \delta_{is}\delta_{sjk} = \delta_{ii}\delta_{ijk} = \delta_{ii} = \begin{cases} 1 & \text{if } i = j = k \\ 0 & \text{otherwise.} \end{cases} = \mathbf{e}_i \mathbf{e}_i^\top =$$ .

It is important to observe, that the "copy" property of the copy tensor only holds for canonical basis vector: it is not true that  $=$  for an arbitrary vector $\boldsymbol{v}$.

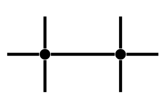
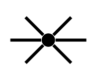
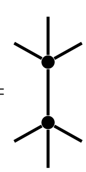
In the following, we list some useful properties of copy tensors.

Remark.

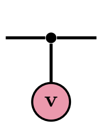
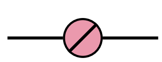
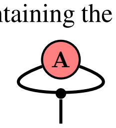
1. The simplest version of the copy tensor is an all one vector, i.e.,  $= \sum_{i=1}^{n} \mathbf{e}_i = \overrightarrow{\mathbf{1}}$.

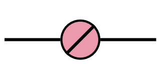
2. The order 2 copy tensor is the identity matrix, where we can simply omit the black node in the middle and assume it as a line, i.e.,  $=$  $= \delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise,} \end{cases} = \mathbf{I}_{ij}.$

3. Copy tensors can also be contracted and combined as needed for specific representation. Using summations we can play with copy tensors:

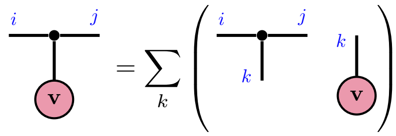
$$\sum_{l_1} \delta_{i_1 j_1 k_1 l_1} \delta_{i_2 j_2 k_2 l_1} =$$  $= \delta_{i_1 j_1 k_1 i_2 j_2 k_2} =$  $=$  $= \sum_{i_1} \delta_{i_1 j_1 k_1 l_1} \delta_{i_1 j_2 k_2 l_2}.$

4. For any vector $\mathbf{v} \in \mathbb{R}^n$, let $\text{diag}(\mathbf{v}) \in \mathbb{R}^{n \times n}$ denote the diagonal matrix having the entries of $\mathbf{v}$ on the diagonal, and for any matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$, let $\text{diag}(\mathbf{A}) \in \mathbb{R}^n$ denote the vector containing the diagonal entries of $\mathbf{A} \in \mathbb{R}^{n \times n}$.

   Then, $\text{diag}(\mathbf{v}) =$  $=$  and $\text{diag}(\mathbf{A}) =$  ,

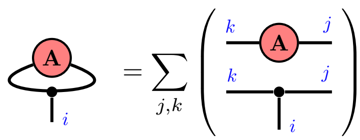
   where  represents a diagonal matrix.

Proof. For the first claim, for any $i, j \in [n]$, we have



$$= \sum_k \delta_{ijk} \mathbf{v}_k = \delta_{ij} \mathbf{v}_i = \text{diag}(\mathbf{v})_{ij}$$

For the second claim, for any $i \in [n]$, we have

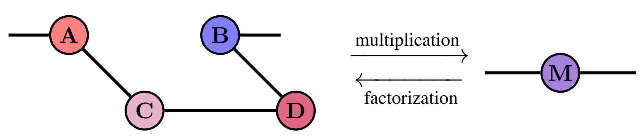


$$= \sum_{j,k} \delta_{ijk} \mathbf{A}_{kj} = \mathbf{A}_{ii} = \text{diag}(\mathbf{A})_i.$$

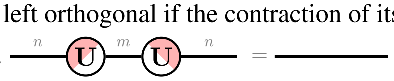
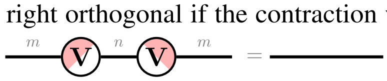
□

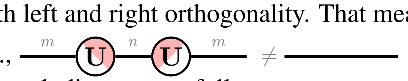
2.4 Matrix Factorization in Tensor Networks

The tensor factorizations can be pictured in tensor networks diagrams as any other operation. In this section, we only introduce graphical diagrams of matrix factorizations. More general factorizations on tensors will be covered in Chapter 4. In tensor networks, factorizing means splitting a single node into multiple nodes, while multiplying refers to combining multiple nodes into a single node. This process can be clearly illustrated using tensor networks diagrams:

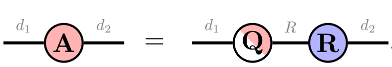
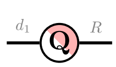


Therefore, we can represent any matrix decomposition in tensor networks diagrams like the celebrated QR and singular value decompositions (SVD). But before presenting these decompositions in tensor network notations we need to introduce orthogonal matrices:
Convention.

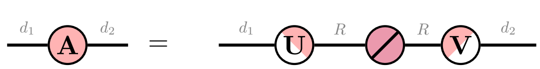
- The matrix $\mathbf{U} \in \mathbb{R}^{m \times n}$ is left orthogonal if the contraction of its transpose with itself from left yields the identity matrix ($\mathbf{U}^\mathsf{T}\mathbf{U} = \mathbf{I}_n$), e.g.,  $= \mathbf{I}_n$.
- The matrix $\mathbf{V} \in \mathbb{R}^{m \times n}$ is right orthogonal if the contraction with its transpose from right yields the identity matrix ($\mathbf{V}\mathbf{V}^\mathsf{T} = \mathbf{I}_m$), e.g.,  $= \mathbf{I}_m$.

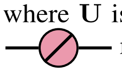
Note that the colored area points towards the identity edge in both left and right orthogonality. That means if the matrix $\mathbf{U}$ is left orthogonal then $\mathbf{U}\mathbf{U}^\mathsf{T}$ is not necessary the identity, i.e.,  $= \mathbf{I}_m$.
The QR decomposition and SVD can be presented in tensor network diagrams as follows:

- **QR Decomposition** For $\mathbf{A} \in \mathbb{R}^{d_1 \times d_2}$, $\mathbf{Q} \in \mathbb{R}^{d_1 \times R}$ and $\mathbf{R} \in \mathbb{R}^{R \times d_2}$,

  , where  is the left-orthogonal matrix, i.e., $\mathbf{Q}^\mathsf{T}\mathbf{Q} = \mathbf{I}_R$

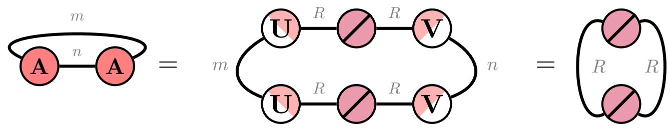
- **Singular Value Decomposition** For $\mathbf{A} \in \mathbb{R}^{d_1 \times d_2}$, $\mathbf{U} \in \mathbb{R}^{d_1 \times R}$, $\Sigma \in \mathbb{R}^{R \times R}$ and $\mathbf{V} \in \mathbb{R}^{R \times d_2}$,

  ,

  where $\mathbf{U}$ is the left-orthogonal ($\mathbf{U}^\mathsf{T}\mathbf{U} = \mathbf{I}_R$), $\mathbf{V}$ is the right-orthogonal ($\mathbf{V}\mathbf{V}^\mathsf{T} = \mathbf{I}_R$) and $\mathbf{\Sigma} \in \mathbb{R}^{R \times R} =$  represents the diagonal matrices, respectively.

Assuming $\mathbf{A}$ has the SVD represented in the previous diagram, we can give a short proof in tensor networks of $\mathrm{tr}(\mathbf{A}^\mathsf{T}\mathbf{A}) = \sum_i \sigma_i^2$, where σ_i are the singular values of matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$:

$\mathrm{tr}(\mathbf{A}^\mathsf{T}\mathbf{A}) =$



$= \mathrm{tr}(\boldsymbol{\Sigma}^2) = \sum_i \sigma_i^2,$

where we use left and right orthogonal property of matrices $\mathbf{U}$ and $\mathbf{V}$.

3 Operations on Tensors

As in the first chapter, a tensor $\mathcal{T} \in \mathbb{R}^{d_1\times\cdots\times d_N}$ can be seen as a multi-dimensional array with size (order) N. An N-th order or N-way tensor has N modes, where each mode represents one dimension [Kolda and Bader, 2009].

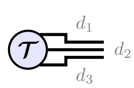
Tensor Fibers For any mode i (where $i = 1, \ldots, N$), tensor *fibers* are obtained by keeping all indices fixed except the i-th one. For example, a matrix column is a mode-1 fiber and a matrix row is a mode-2 fiber. In a third-order tensor $\mathcal{T} \in \mathbb{R}^{d_1\times d_2\times d_3}$, we have $d_2 d_3$ mode-1 fibers which are vectors of size d_1, i.e., $\mathcal{T}_{:,i_2,i_3} \in \mathbb{R}^{d_1}$, for $i_2 \in [d_2]$ and $i_3 \in [d_3]$, where the colon indicates varying the first index while i_2 and i_3 remain fixed. Simply, fibers are vectors.

Tenor Slices Slices of a tensor are two-dimensional arrays matrices, obtained by fixing all but two indices. For a third-order tensor $\mathcal{T} \in \mathbb{R}^{d_1\times d_2\times d_3}$ there are horizontal, lateral, and frontal slides denoted by $\mathcal{T}_{i_1,:,:} \in \mathbb{R}^{d_2\times d_3}$, $\mathcal{T}_{:,i_2,:} \in \mathbb{R}^{d_1\times d_3}$, and $\mathcal{T}_{:,:,i_3} \in \mathbb{R}^{d_1\times d_2}$, respectively. Therefore, slices are matrices.

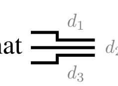
3.1 Permute and Reshape Tensors

Permute and reshape are two fundamental operations on tensors. **Permute** rearranges the indices of a tensor without changing the overall order of a tensor. An example of a permutation is a matrix transpose. **Reshape** combines indices into larger indices, reducing the total number of indices while keeping the tensor size unchanged. In the following, we introduce the *vectorization* and *matricization* of a tensor, which are two main reshaping operations on a tensor.

Definition 2. *(Vectorization) Let $\mathcal{T} \in \mathbb{R}^{d_1\times d_2\times d_3}$. The* vectorization *of $\mathcal{T}$ is the vector obtained by concatenating its mode-1 fibers, e.g.,*

$\mathrm{vec}(\mathcal{T}) =$ 

$\in \mathbb{R}^{d_1 d_2 d_3}.$

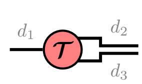
We can also see vectorization as a flattening operator of any order of a tensor into a vector. Note that 

presents an edge of size $d_1 d_2 d_3$. In general, convergent edges represent an edge whose size is the product of the sizes of all associated edges.

Definition 3. *(Matricizitation) Let $\mathcal{T} \in \mathbb{R}^{d_1\times d_2\times d_3}$. A matricization of $\mathcal{T}$ is obtained by unfolding it into a matrix by taking all fibers along one direction and stacking them together. A matricization of a tensor $\mathcal{T}$ along mode i is represented by $\mathcal{T}_{(i)}$ where for the 3rd order tensor $i \in [3]$. For example, the mode-1 matricization of $\mathcal{T}$ is*

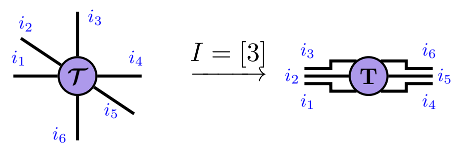
$$\mathcal{T}_{(1)} = \begin{bmatrix} | & | & | & | \\ \mathcal{T}_{:,11} & \mathcal{T}_{:,12} & \ldots & \mathcal{T}_{:,d_2 d_3} \\ | & | & | & | \end{bmatrix} \in \mathbb{R}^{d_1\times d_2 d_3},$$

Observe that in this matricization, the two indices corresponding to the 2nd and 3rd modes of $\mathcal{T}$ are grouped *together to form a new index ranging from 1 to $d_2 d_3$. In tensor network diagrams, we will represent such a grouping of indices by grouping the corresponidng legs together:*

$\mathcal{T}_{(1)} =$ 

The mode-2 and mode-3 matricization $\mathcal{T}_{(2)} \in \mathbb{R}^{d_2 \times d_1 d_3}$ *and* $\mathcal{T}_{(3)} \in \mathbb{R}^{d_3 \times d_1 d_2}$ *are defined similarly. More generally, this definition can be extended to any arbitrary order of tensors, i.e.,* $\mathcal{T}_{(n)} \in \mathbb{R}^{d_n \times d_1 \dots d_{n-1} d_{n+1} \dots d_N}$.

Matricization can also be seen as the flattening or unfolding of a tensor into a matrix. In general, the notion of matricization can be extended to any subset $I \subset [N]$ of the modes of $\mathcal{T}$ which maps I modes of $\mathcal{T}$ to the rows of $\mathbf{T}$ resulting in a matrix $\mathbf{T}_{([I])}$ of size $\prod_{i \in I} d_i \times \prod_{j \in [N] \setminus I} d_j$. For instance, for a 6-th order tensor $\mathcal{T} \in \mathbb{R}^{d_1 \times \dots \times d_6}$ we can group the indices as follows

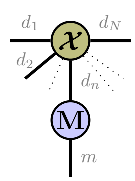
$\mathcal{T}_{i_1,\dots,i_6} =$  $= (\mathcal{T}_{([3])})_{i_1 i_2 i_3, i_4 i_5 i_6} \in \mathbb{R}^{d_1 d_2 d_3 \times d_4 d_5 d_6}$

3.2 Products

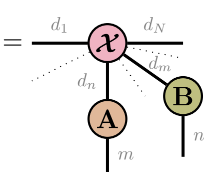
Tensors can be multiplied through different operations, similar to contraction for matrix multiplication as we see in Section 2.1. In this section, we provide graphical illustrations for different product operations. The most general tensor multiplication is the Einsum operation pictured in chapter (6), which performs summation (contraction) over same-size tensor indices.

Mode-n Product Tensor mode-n products, where a tensor is multiplied by a matrix along a specific mode, are a special case of Einsum and can be seen as a generalization of matrix products. These include mode-n products between tensors and matrices, as well as tensors and vectors.

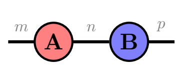
1. **Mode-n product (matrix).** The mode-n product of a tensor $\mathcal{X} \in \mathbb{R}^{d_1 \times \dots \times d_N}$ with the matrix $\mathbf{M} \in \mathbb{R}^{m \times d_n}$ is denoted by $\mathcal{X} \times_n \mathbf{M} \in \mathbb{R}^{d_1 \times \dots \times d_{n-1} \times m \times d_{n+1} \times \dots \times d_N}$ and defined as

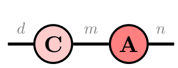
   $\mathcal{X} \times_n \mathbf{M} =$  $\in \mathbb{R}^{d_1 \times \dots \times d_{n-1} \times m \times d_{n+1} \times \dots \times d_N}$

   The operation contracts the tensor's n-th mode with the matrix's second mode, replacing the original dimension d_n with a new dimension m. For $\mathbf{A} \in \mathbb{R}^{d_n \times m}$, $\mathbf{B} \in \mathbb{R}^{d_m \times n}$ and $\mathcal{X} \in \mathbb{R}^{d_1 \times \dots \times d_N}$ with distinct modes $n \neq m$, the order of multiplication does not matter, i.e.,

   $\mathcal{X} \times_n \mathbf{A} \times_m \mathbf{B} =$  $= \mathcal{X} \times_m \mathbf{B} \times_n \mathbf{A}, \text{ For } (n \neq m).$

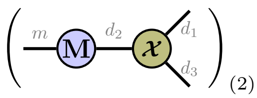
   The mode-n tensor product can be seen as a generalization of matrix multiplication:
   Example.
   Let $\mathbf{A} \in \mathbb{R}^{m \times n}$, $\mathbf{B} \in \mathbb{R}^{p \times n}$ and $\mathbf{C} \in \mathbb{R}^{d \times m}$, then

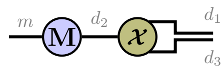
   $\mathbf{A} \times_2 \mathbf{B} =$  $= \mathbf{A}\mathbf{B}^\top \in \mathbb{R}^{m \times p}$

   $\mathbf{A} \times_1 \mathbf{C} =$  $= \mathbf{C}\mathbf{A} \in \mathbb{R}^{d \times n}$

   Proposition 1. *Let* $\mathcal{X} \in \mathbb{R}^{d_1 \times \dots \times d_N}$ *and* $\mathbf{M} \in \mathbb{R}^{m \times d_n}$ *then* $(\mathcal{X} \times_n \mathbf{M})_{(n)} = \mathbf{M}\mathcal{X}_{(n)}$.

Proof. We show this identity for the special case $n = 2$ and $N = 3$. The extension to the general case is straightforward. We have

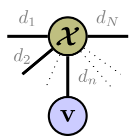
$$(\boldsymbol{\mathcal{X}} \times_2 \mathbf{M})_{(2)} =$$


$$=$$


$$= \mathbf{M}\boldsymbol{\mathcal{X}}_{(2)} \in \mathbb{R}^{m \times d_1 d_3}.$$

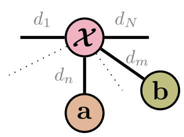
□

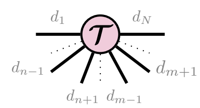
2. **Mode-n product (vector).** The mode-n product of a tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{d_1 \times \cdots \times d_N}$ with a vector $\mathbf{v} \in \mathbb{R}^{d_n}$, is denoted by $\boldsymbol{\mathcal{X}} \times_n \mathbf{v}$ and is a tensor $\boldsymbol{\mathcal{S}} \in \mathbb{R}^{d_1 \times \cdots \times d_{n-1} \times d_{n+1} \times \cdots \times d_N}$. The result is a tensor of order $N - 1$. It can be pictured in a tensor network diagram as

$$\boldsymbol{\mathcal{S}} = \boldsymbol{\mathcal{X}} \times_n \mathbf{v} =$$


$$\in \mathbb{R}^{d_1 \times \cdots \times d_{n-1} \times d_{n+1} \times \cdots \times d_N}.$$

In mode-n vector multiplication, unlike mode-n matrix multiplication, the order of multiplication matters because it affects intermediate results. Let $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{d_1 \times \cdots \times d_m \times \cdots \times d_n \times \cdots \times d_N}$ and $\mathbf{a} \in \mathbb{R}^{d_n}$, $\mathbf{b} \in \mathbb{R}^{d_m}$, then

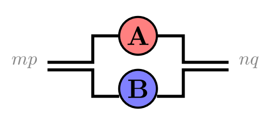
$$\boldsymbol{\mathcal{X}} \times_n \mathbf{a} \times_m \mathbf{b} =$$


$$=$$


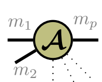
$$\neq (\boldsymbol{\mathcal{X}} \times_m \mathbf{b}) \times_n \mathbf{a},$$

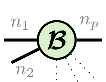
as mode-n vector multiplication in the right-hand side, drops the m-th dimension [Bader and Kolda, 2006].

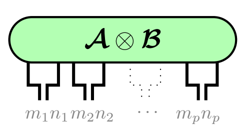
Kronecker product Let $\mathbf{A} \in \mathbb{R}^{m \times n}$ and $\mathbf{B} \in \mathbb{R}^{p \times q}$ then the Kronecker product, $\mathbf{A} \otimes \mathbf{B} \in \mathbb{R}^{mp \times nq}$ is defined by

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & a_{12}\mathbf{B} & \dots & a_{1n}\mathbf{B} \\ a_{21}\mathbf{B} & a_{22}\mathbf{B} & \dots & a_{2n}\mathbf{B} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}\mathbf{B} & a_{m2}\mathbf{B} & \dots & a_{mn}\mathbf{B} \end{bmatrix} = \tag{7}$$


More generally, Kronecker product can be defined for any two tensors with the same order, e.g.,

$$\boldsymbol{\mathcal{A}} \otimes \boldsymbol{\mathcal{B}} =$$


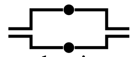
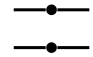
$$\otimes$$


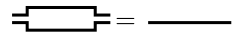
$$=$$


$$\in \mathbb{R}^{m_1 n_1 \times \cdots \times m_p n_p}.$$

In the following, we list some useful properties of copy tensors.
Remarks.

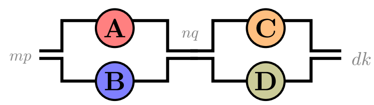
1. Kronecker product is not commutative, i.e., $\mathbf{A} \otimes \mathbf{B} \neq \mathbf{B} \otimes \mathbf{A}$
2. $\mathbf{I} \otimes \mathbf{A} = \begin{bmatrix} \mathbf{A} & 0 & \dots & 0 \\ 0 & \mathbf{A} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \mathbf{A} \end{bmatrix}$ is a block diagonal matrix.
3. The Kronecker product of two tensors of the same order results in a tensor of the same order, while their outer product produces a tensor with double the order.
4. By reshaping the Kronecker product $\mathbf{A} \otimes \mathbf{B}$, the outer product $\mathbf{A} \circ \mathbf{B}$ can be obtained.

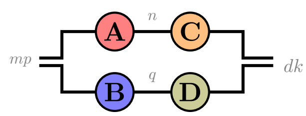
5. **Kronecker product of copy tensors** is  while the **outer product of copy tensors** is , where it reveals that they are not equal, but reshaping of one another.

6. The Kronecker product of two **identity matrices** is another identity matrix, i.e., . In general, the legs of a Kronecker product can be represented as straight lines, with their size corresponding to the product of the sizes of the respective legs.

7. Kronecker product has a *mixed product* property, i.e., $(\mathbf{A} \otimes \mathbf{B})(\mathbf{C} \otimes \mathbf{D}) = \mathbf{AC} \otimes \mathbf{BD}$ where $\mathbf{A} \in \mathbb{R}^{m \times n}, \mathbf{B} \in \mathbb{R}^{p \times q}, \mathbf{C} \in \mathbb{R}^{n \times d}$ and $\mathbf{D} \in \mathbb{R}^{q \times k}$

   Proof.

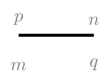
   $(\mathbf{A} \otimes \mathbf{B})(\mathbf{C} \otimes \mathbf{D}) =$ 

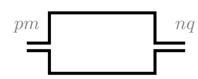
   $=$ 

   $= \mathbf{AC} \otimes \mathbf{BD}.$

   □

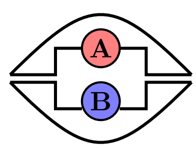
   As a special case, we can see $(\mathbf{A} \otimes \mathbf{I})(\mathbf{I} \otimes \mathbf{B}) = \mathbf{A} \otimes \mathbf{B}$.

8. As we can see in all tensor network diagrams above, 

   can be reshaped as 

   and vice versa.

9. For $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n \times n}$, we can write $\operatorname{tr}(\mathbf{A} \otimes \mathbf{B}) = \operatorname{tr}(\mathbf{A}) \operatorname{tr}(\mathbf{B})$

   $\operatorname{tr}(\mathbf{A} \otimes \mathbf{B}) =$ 

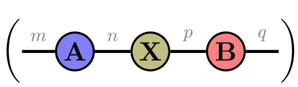
   $= \operatorname{tr}(\mathbf{A}) \operatorname{tr}(\mathbf{B}).$

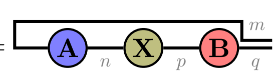
10. **Sylvester Identity**

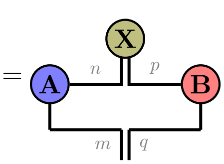
$$\operatorname{vec}(\mathbf{AXB}) = (\mathbf{B}^\top \otimes \mathbf{A}) \operatorname{vec}(\mathbf{X}),$$

   where $\mathbf{A} \in^{m \times n}, \mathbf{X} \in \mathbb{R}^{n \times p}$ and $\mathbf{B} \in \mathbb{R}^{p \times q}$.

   Proof.

   $\operatorname{vec}(\mathbf{AXB}) = \operatorname{vec}$ 

   $=$ 

   $=$ 

   $= (\mathbf{A} \otimes \mathbf{B}^\top) \operatorname{vec}(\mathbf{X}).$

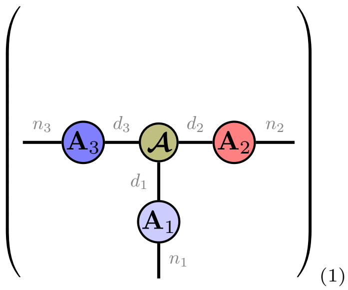
   □

Proposition 2. *Let* $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{d_1 \times d_2 \times \cdots \times d_p}$, $\mathbf{A}_1 \in \mathbb{R}^{d_1 \times n_1}, \mathbf{A}_2 \in \mathbb{R}^{d_2 \times n_2}, \ldots, \mathbf{A}_p \in \mathbb{R}^{d_p \times n_p}$, *then*

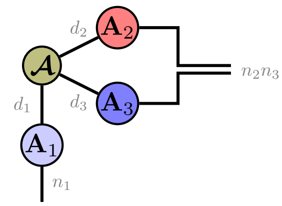
$$(\boldsymbol{\mathcal{A}} \times_1 \mathbf{A}_1 \times_2 \mathbf{A}_2 \times_3 \mathbf{A}_3 \times_4 \ldots \times_p \mathbf{A}_p)_{(n)} = \mathbf{A}_n \boldsymbol{\mathcal{A}}_{(n)} (\mathbf{A}_1 \otimes \ldots \otimes \mathbf{A}_{n-1} \otimes \mathbf{A}_{n+1} \otimes \ldots \otimes \mathbf{A}_p)^\top$$

Proof. For $p = 3$ and $n = 1$,

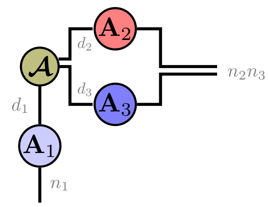
$$(\boldsymbol{\mathcal{A}} \times_1 \mathbf{A}_1 \times_2 \mathbf{A}_2 \times_3 \mathbf{A}_3)_{(1)} =$$



$$=$$



$$=$$

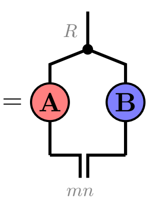


$$= \mathbf{A}_1 \boldsymbol{\mathcal{A}}_{(1)} (\mathbf{A}_2 \otimes \mathbf{A}_3)^\mathsf{T}.$$

□

Khatri-Rao product. Let $\mathbf{A} \in \mathbb{R}^{m \times R}$ and $\mathbf{B} \in \mathbb{R}^{n \times R}$ then the Khatri-Rao product, $\mathbf{A} \odot \mathbf{B} \in \mathbb{R}^{mn \times R}$ is defined by

$$\mathbf{A} \odot \mathbf{B} = \begin{bmatrix} | & | & & | \\ \mathbf{a}_1 \otimes \mathbf{b}_1 & \mathbf{a}_2 \otimes \mathbf{b}_2 & \dots & \mathbf{a}_R \otimes \mathbf{b}_R \\ | & | & & | \end{bmatrix} =$$



,

where $\mathbf{a}_1, \dots, \mathbf{a}_R \in \mathbb{R}^m$ are the columns of $\mathbf{A}$, $\mathbf{b}_1, \dots, \mathbf{b}_R \in \mathbb{R}^n$ are the columns of $\mathbf{B}$ and the columns of $\mathbf{A} \odot \mathbf{B}$ is the subset of the Kronecker product. In the corresponding tensor network diagram, the copy tensor captures the fact that the second indices are the same.

Remarks.

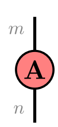
1. Like Kronecker product, Khatri-Rao product is not commutative, i.e., $\mathbf{A} \odot \mathbf{B} \neq \mathbf{B} \odot \mathbf{A}$.
2. Khatri-Rao product is associative, i.e., $\mathbf{A} \odot (\mathbf{B} \odot \mathbf{C}) = (\mathbf{A} \odot \mathbf{B}) \odot \mathbf{C}$, where $\mathbf{A} \in \mathbb{R}^{m \times R}, \mathbf{B} \in \mathbb{R}^{n \times R}$ and $\mathbf{C} \in \mathbb{R}^{s \times R}$.

Hadamard product Let $\mathbf{A} \in \mathbb{R}^{m \times n}$ and $\mathbf{B} \in \mathbb{R}^{m \times n}$ be the matrices of the same dimension then the Hadamard product $\mathbf{A} * \mathbf{B} \in \mathbb{R}^{m \times n}$ is the matrix of the same dimension defined element-wise by

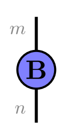
$$(\mathbf{A} * \mathbf{B})_{ij} = \mathbf{A}_{ij} \mathbf{B}_{ij},$$

and by using tensor diagrams

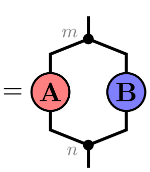
$$\mathbf{A} * \mathbf{B} =$$



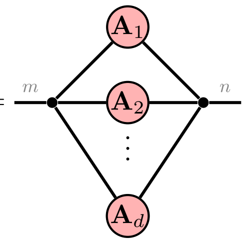
$$*$$



$$=$$



More generally, for any $\mathbf{A}_1 \in \mathbb{R}^{m\times n} \dots \mathbf{A}_N \in \mathbb{R}^{m\times n}$ we have

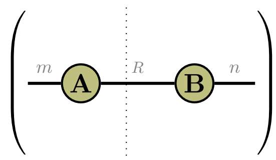
$\mathbf{A}_1 * \mathbf{A}_2 \dots * \mathbf{A}_d =$ 

Note. The Hadamard product of two copy tensors is a copy tensor:

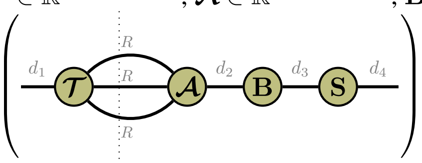
3.3 (temporary) Useful TN results

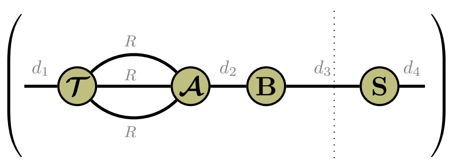
3.3.1 Rank of matricization of TN

Theorem 3. *For an arbitrary tensor network, the rank of any matricization is bounded by the weight of the cut of the graph.*

1. For $\mathbf{A} \in \mathbb{R}^{m\times R}$ and $\mathbf{B} \in \mathbb{R}^{R\times n}$, $\mathrm{rank}(\mathbf{AB}) =$ rank  $\leq R$

2. Let $\boldsymbol{\mathcal{T}} \in \mathbb{R}^{d_1\times R\times R\times R}$, $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{R\times R\times R\times d_2}$, $\mathbf{B} \in \mathbb{R}^{d_2\times d_3}$ and $\mathbf{S} \in \mathbb{R}^{d_3\times d_4}$, then

   rank  $\leq R^3$ and

   rank  $\leq d_3$

3.3.2 cutting edge

4 Tensor Decompositions

Working with high-order tensors is computationally expensive because the number of elements grows exponentially with the tensor's order. Tensor decompositions have emerged as powerful and efficient tools to address this issue. Similar to matrix factorizations, tensor decompositions break down a high-order tensor into smaller components with lower order and fewer entries, making them easier to work with. However, unlike matrices, there are many different ways to decompose a tensor, each associated with a distinct concept of rank. In this chapter, we the most well-known tensor decomposition models.

4.1 CANDECOMP/PARAFAC (CP) Decomposition

The CP decomposition [Hitchcock, 1927] of an N-th order tensor $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{d_1\times\cdots\times d_N}$ is the sum of a finite number of rank-one tensors. Equivalently, it is a linear combination of R rank-one tensors where R is called the rank of the decomposition:

$$\boldsymbol{\mathcal{A}} = \sum_{r=1}^{R} \mathbf{a}_1^{(r)} \circ \dots \circ \mathbf{a}_N^{(r)},$$

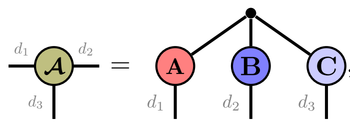
where $\mathbf{a}_n^{(1)}, \ldots, \mathbf{a}_n^{(R)} \in \mathbb{R}^{d_n}$ for each $n \in [N]$.

By grouping all the vectors $\mathbf{a}_n^{(r)}$ in factor matrices,

$$\mathbf{A}_1 = \begin{pmatrix} \mathbf{a}_1^{(1)} & \ldots & \mathbf{a}_1^{(R)} \end{pmatrix} \in \mathbb{R}^{d_1 \times R}, \ldots, \mathbf{A}_N = \begin{pmatrix} \mathbf{a}_N^{(1)} & \ldots & \mathbf{a}_N^{(R)} \end{pmatrix} \in \mathbb{R}^{d_N \times R},$$

the CP decomposition is concisely noted as $\boldsymbol{\mathcal{A}} = [\![\mathbf{A}_1, \cdots, \mathbf{A}_N]\!]$.

In tensor networks, a CP decomposition $\boldsymbol{\mathcal{A}} = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$ is represented by



where the black dot is the third order copy tensor introduced in 2.3.

CP Rank of a Tensor. The most fundamental concept of rank for tensors, and also the oldest, is the CP rank, which was first introduced by [Hitchcock, 1927]. The CP rank of a tensor is defined as the minimum number of rank-one tensors needed to express the tensor as their sum. This definition of tensor rank is similar to the definition of matrix rank, but the properties of tensor rank differ significantly from those of matrix rank. From a computational standpoint, one key difference is that, unlike matrix rank, there is no straightforward polynomial algorithm to determine a tensor's CP rank. In fact, computing the rank of a tensor is an NP-hard problem [Hillar and Lim, 2013]. There are several variations of tensor rank, each linked to a specific tensor decomposition. We will introduce some of these different types of ranks as we proceed through this chapter.

Remark 4. *We list here some interesting properties of the CP rank and the CP decomposition.*

1. *From Theorem 3 and Remark 2.3 (item 3), one can show that if a tensor $\boldsymbol{\mathcal{A}}$ admits a rank R CP decomposition, then all its matricizations have rank upper bounded by R. The following tensor network illustrates this result:*

2. *The CP rank of a tensor $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{d_1 \times \cdots \times d_N}$ can easily be upper bounded as*

$$\mathrm{rank}_{\mathrm{CP}}(\boldsymbol{\mathcal{A}}) \leq \min_{n \in [N]} \prod_{i \neq n} d_i.$$

3. *For order 2 tensors $\mathbf{A} \in \mathbb{R}^{d_1 \times d_2}$, we can recover the classical notion of rank R factorization.*

$$\mathbf{A} = \sum_{r=1}^{R} \mathbf{a}_1^{(r)} \circ \mathbf{a}_2^{(r)} = \begin{pmatrix} \mathbf{a}_1^{(1)} & \ldots & \mathbf{a}_1^{(R)} \end{pmatrix} \begin{pmatrix} \mathbf{a}_2^{(1)} & \ldots & \mathbf{a}_2^{(R)} \end{pmatrix}^\top.$$

4. *A smaller CP rank R results in a more efficient CP decomposition.*

5. *The CP decomposition $\boldsymbol{\mathcal{A}} = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$ can be expressed using the Kronecker delta*

$$\boldsymbol{\mathcal{A}}_{i,j,k} = \sum_{r_1, r_2, r_3} \delta_{r_1, r_2, r_3} \mathbf{A}_{i, r_1} \mathbf{B}_{j, r_2} \mathbf{C}_{k, r_3},$$

*as well as with mode-n products (see **??**)*

$$\boldsymbol{\mathcal{A}} = \boldsymbol{\mathcal{I}} \times_1 \mathbf{A} \times_2 \mathbf{B} \times_3 \mathbf{C}$$

where $\boldsymbol{\mathcal{I}}$ is the 3rd order copy tensor.

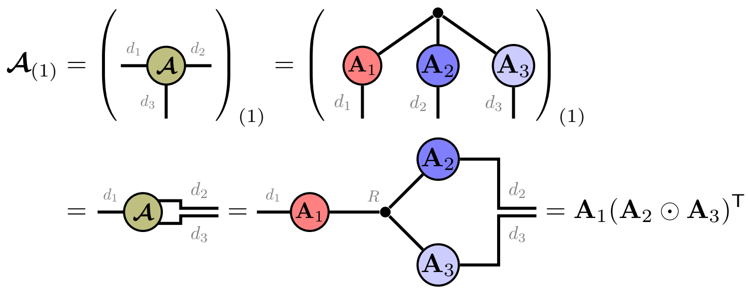
6. *The rank of the second-order tensors (matrices), over the fields $\mathbb{R}$ and $\mathbb{C}$ is the same. However, for higher order tensors (N-th order tensors with $N > 3$) the rank can be different depending on the decomposition field [Kruskal, 1989].*

7. *The CP of N-th order d-dimensional tensors ($d_1 = \cdots = d_N = d$) using only $\mathcal{O}(NdR)$ parameters instead of d^N. If R is small then the number of parameters can be considerably reduced.*

Proposition 5. *Let $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{d_1 \times d_2 \times \cdots \times d_N}$. If $\boldsymbol{\mathcal{A}} = [\![\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_N]\!]$, then*

$$\boldsymbol{\mathcal{A}}_{(n)} = \mathbf{A}_n \left(\mathbf{A}_1 \odot \ldots \odot \mathbf{A}_{(n-1)} \odot \mathbf{A}_{(n+1)} \odot \ldots \odot \mathbf{A}_N\right)^{\mathsf{T}}.$$

Proof. For $N = 3$ and $n = 1$,



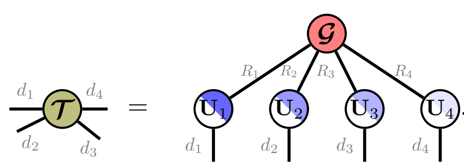
□

4.2 Tucker Decomposition

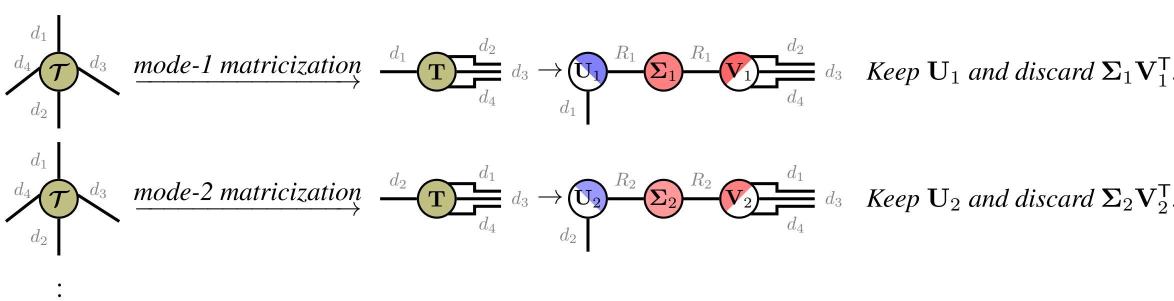
Tucker [Tucker, 1966] introduced the Tucker decomposition which factorizes an N-th order tensor into a smaller tensor and N factor matrices. The smaller tensor is called a core tensor in this decomposition. The Tucker decomposition is a mode-n product (see, e.g., 1) between a core tensor and the factors matrices. Let $\boldsymbol{\mathcal{T}} \in \mathbb{R}^{d_1 \times \cdots \times d_N}$, then the Tucker decomposition of tensor $\boldsymbol{\mathcal{T}}$ is the decomposition of the form $\boldsymbol{\mathcal{T}} = \mathbf{G} \times_1 \mathbf{U}_1 \times_2 \ldots \times_{N-1} \mathbf{U}_{N-1} \times_N \mathbf{U}_N$, where $\mathbf{G} \in \mathbb{R}^{R_1 \times \cdots \times R_N}$ and $\mathbf{U}_i \in \mathbb{R}^{d_i \times R_i}, i \in [N]$. The tuple $(R_i), i \in [N]$ that contains the dimensions of the core tensor along all its modes is called the Tucker rank. It is not difficult to show that the factor matrices $\mathbf{U}_i \in \mathbb{R}^{d_i \times R_i}$ can always be set as unitary.

Remark 6. *We list here some interesting properties of the Tucker decomposition and the HOSVD algorithm.*

1. *The Tucker decomposition of a 4-th order tensor can be illustrated in a tensor networks notation as follows:*

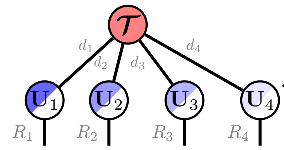


2. ***Computing The Tucker Decomposition.*** *The basic idea of the Tucker decomposition is finding the R_n leading left singular vectors of in mode n, independent of the other modes. This algorithm is depicted below and known as a Higher Order SVD (HOSVD). We start by using the fact that there exists SVD for any mode-n matricization of the tensor $\boldsymbol{\mathcal{T}} \in \mathbb{R}^{d_1 \times \cdots \times d_N}$. For simplicity, we picture $N = 4$,*



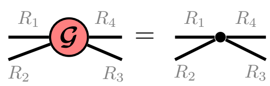
Construct tensor $\mathcal{G} \in \mathbb{R}^{R_1 \times \cdots \times R_4}$ by performing a mode-n product with the transpose of retained factor matrices for each corresponding mode (for $n \in [4]$), i.e.,

$$\mathcal{G} = \mathcal{T} \times_1 \mathbf{U}_1^\mathsf{T} \times_2 \ldots \times_4 \mathbf{U}_4^\mathsf{T} =$$



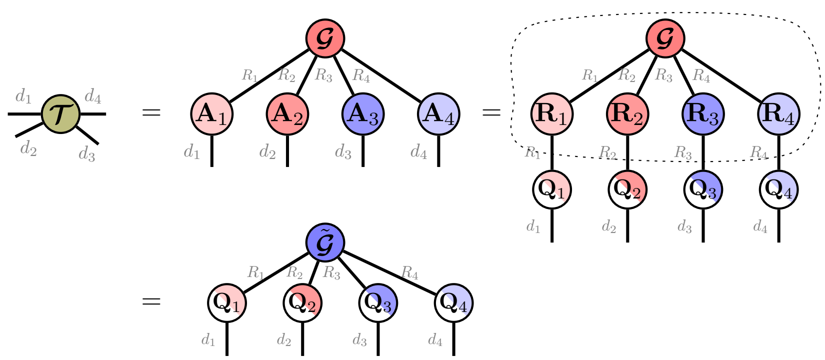
Now contract the smaller tensor $\mathcal{G} \in \mathbb{R}^{R_1 \times \cdots \times d_4}$ with the factor matrices $\mathbf{U}_1, \ldots, \mathbf{U}_4$ retained from the previous part.

3. *If $\mathcal{T} = \mathcal{G} \times_1 \mathbf{U}_1 \times_2 \ldots \times_{N-1} \mathbf{U}_{N-1} \times_N \mathbf{U}_N$ with $\mathbf{U}_i$s orthogonal, then $\|\mathcal{T}\|_F = \|\mathcal{G}\|_F$.*

4. *If we ignore the orthogonality and choose*  *, the CP decomposition is recovered. Moreover, as CP decomposition always exists for an arbitrary tensor, we conclude that the Tucker decomposition also exists.*

5. *If $\mathcal{T} = \mathbf{G} \times_1 \mathbf{A} \times_2 \mathbf{B} \times_3 \mathbf{C} \times_4 \mathbf{D}$, with $\mathbf{A}, \mathbf{B}, \mathbf{C}$ and $\mathbf{D}$ not necessarily orthogonal, then there exists $\tilde{\mathcal{G}}$ and $\mathbf{Q}_1, \mathbf{Q}_2, \mathbf{Q}_3$ and $\mathbf{Q}_4$ orthogonal such that $\mathcal{T} = \tilde{\mathcal{G}} \times_1 \mathbf{Q}_1 \times_2 \mathbf{Q}_2 \times_3 \mathbf{Q}_3 \times_4 \mathbf{Q}_4$.*

Proof. By using QR decomposition for each factor matrix, we obtain

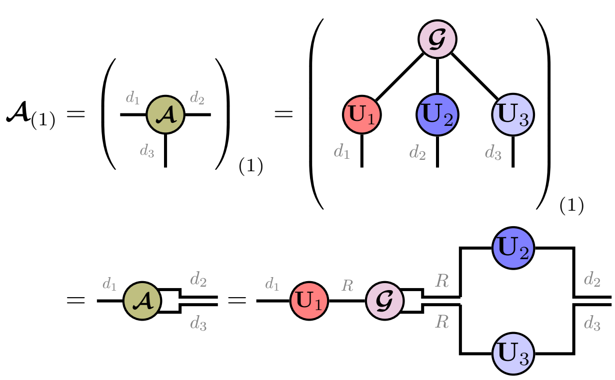


□

Proposition 7. *Let $\mathcal{A} \in \mathbb{R}^{d_1 \times d_2 \times \cdots \times d_N}$. If $\mathcal{A} = \mathcal{G} \times_1 \mathbf{U}_1 \times_2 \ldots \times_N \mathbf{U}_N$ then*

$$\mathcal{A}_{(i)} = \mathbf{U}_i \mathcal{G}_{(i)} \left(\mathbf{U}_1 \otimes \ldots \otimes \mathbf{U}_{(i-1)} \otimes \mathbf{U}_{(i+1)} \otimes \ldots \otimes \mathbf{U}_N \right)^\mathsf{T}.$$

Proof. For $N = 3$ and $n = 1$,



$$= \mathbf{U}_1 \mathcal{G}_{(1)} (\mathbf{U}_2 \otimes \mathbf{U}_3)^\mathsf{T}.$$

□

6. *The Tucker rank of a tensor $\mathcal{T}$ is determined by the rank of its matricizations, i.e., rank$(\mathcal{T}_{(i)})$ for $i \in [N]$.*

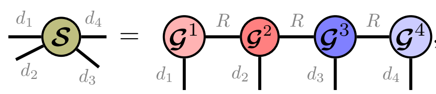
7. *For the N-th order d-dimensional tensor, the number of parameters for its Tucker decomposition is $\mathcal{O}(R^N + NdR)$ with assumption $R_1 = \cdots = R_N = R$ and $d_1 = \cdots = d_N = d$.*

4.3 Tensor Train (TT) Decomposition

The Tensor Train (TT) decomposition [Oseledets, 2010] is one of the significant tensor factorizations method that decomposes an N-th order tensor in to N smaller third-order tensors. Let $\mathcal{S} \in \mathbb{R}^{d_1 \times \cdots \times d_N}$ be an N-dimensional array. A rank-$(R_1, \ldots, R_{N-1})$ *tensor train decomposition* of a tensor $\mathcal{S} \in \mathbb{R}^{d_1 \times \cdots \times d_N}$ factorizes it into a product of N third-order tensors $\mathcal{G}^n \in \mathbb{R}^{R_{n-1} \times d_n \times R_n}$ for $n \in [N]$ (with $R_0 = R_N = 1$):

$$\mathcal{S}_{i_1, \cdots, i_N} = \sum_{r_0, \cdots, r_N} \prod_{n=1}^{N} \mathcal{G}_n(r_{n-1}, i_n, r_n),$$

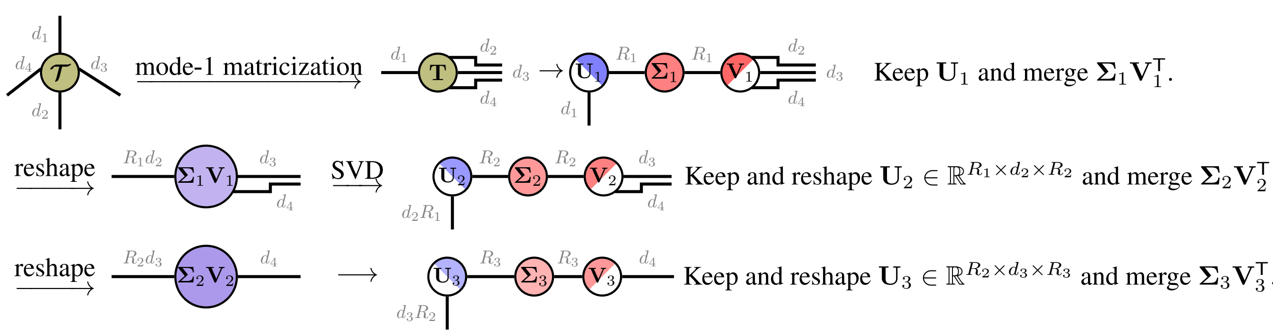
for all $i_1 \in [d_1], \cdots, i_N \in [d_N]$, where each r_n ranges from 1 to R_n, for $n \in [N]$. The *TT-rank* decomposition of $\mathcal{S}$ is the smallest $(R_1, R_2, \ldots, R_{N-1})$ such that $\mathcal{S} = \langle\!\langle \mathcal{G}^1, \mathcal{G}^2, \cdots, \mathcal{G}^{N-1}, \mathcal{G}^N \rangle\!\rangle$, where $\langle\!\langle \mathcal{G}^1, \mathcal{G}^2, \cdots, \mathcal{G}^{N-1}, \mathcal{G}^N \rangle\!\rangle$ represents a TT decomposition with core tensors $\mathcal{G}^1, \ldots, \mathcal{G}^N$. The TT decomposition can be represented in a tensor networks notation, i.e., for a 4-th order tensor:



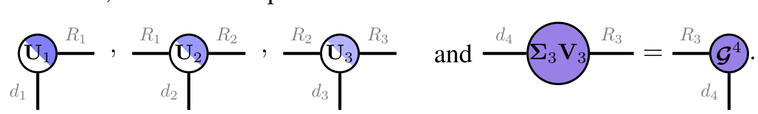
The intermediate edges are also known as bond dimensions and the free legs as physical dimensions. The above representation is also known as a TT vector. Next, we explain how to construct the TT tensor from a tensor $\mathcal{T} \in \mathbb{R}^{d_1 \times \cdots \times d_N}$ by an SVD which is called TT-SVD algorithm [Oseledets, 2010]. The theorem illustrates the existence of the minimal tensor train decomposition for any arbitrary tensor.

Theorem 8. ***(Computing (Orthogonal) Tensor Train Decomposition.)*** *For any $\mathcal{T} \in \mathbb{R}^{d_1 \times \cdots \times d_N}$, let $\mathbf{T}_{(n)} \in \mathbb{R}^{d_1 \ldots d_n \times d_{n+1} \ldots d_N}$ be the matricization obtained by mapping the first n modes of $\mathcal{T}$ to the rows of $\mathbf{T}$. Then the TT rank of $\mathcal{T}$ is given by $R_n =$ rank $(\mathbf{T}_{(n)})$ for any $n \in [N]$.*

Proof. Leaving $R_n =$ rank $(\mathbf{T}_{(n)})$, we will construct the TT tensor by SVD (or QR) decomposition as follows.



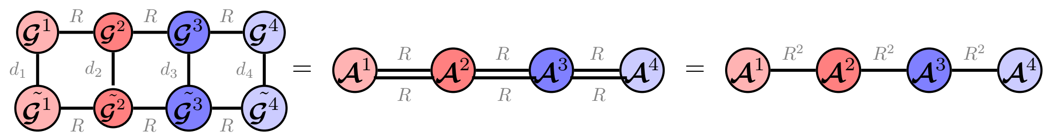
Therefore, the TT decomposition cores are:



We can now contract all obtained cores from R_n part, for $n \in \{1, 2, 3, 4\}$. Note that the TT form obtained through the algorithm above is known as the left-orthogonal TT, which we will define later in this section. Alternatively, the TT-SVD algorithm can start with mode-4 matricization and result in the right-orthogonal TT decomposition. ☐

4.4 Efficient Operations in TT Format.

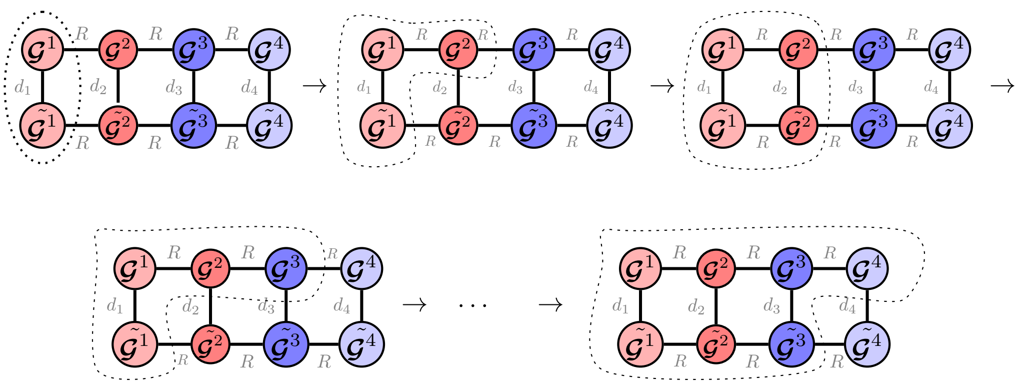
- **Inner Product.** As mentioned in section 2.1, for tensors, as well as vectors, the inner product can be represented by connecting all corresponding indices. Suppose that we have two 4-th order tensors $\mathcal{T}, \widetilde{\mathcal{T}} \in \mathbb{R}^{d_1 \times \ldots \times d_4}$ in TT formats. Then the inner product can also be represented in a TT format, i.e.,

$\langle \mathcal{T}, \widetilde{\mathcal{T}} \rangle =$ 

We can see that this representation is correct by simply using the definition,

$$\langle \mathcal{T}, \widetilde{\mathcal{T}} \rangle = \sum_{i_1=1}^{d_1} \cdots \sum_{i_4=1}^{d_4} \mathcal{T}_{i_1, \ldots i_4} \widetilde{\mathcal{T}}_{i_1, \ldots i_4}.$$

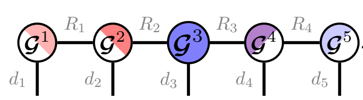
It is important to note that the total complexity to compute the inner product of two N-th order tensors in a TT format is $\mathcal{O}(NdR^4)$. This is a huge improvement compared to the complexity of computing inner product in an standard way which is $\mathcal{O}(d^N)$. Therefore, the TT format is a useful and efficient way to perform the operations on high-dimensional tensors. Moreover, we can contract cores in much more efficient way, i.e.



As we can see above, the complexity of an inner product between two vectors of size $\mathcal{O}(d^N)$ can be reduced to $\mathcal{O}(NdR^3)$. In general, finding the optimal order of contraction of an arbitrary tensor network is an NP-hard problem [Chi-Chung et al., 1997]. Note that inner product operation increases the size of the bond dimensions.

We conclude this subsection by introducing the canonical form of the TT decomposition [Holtz et al., 2012, Evenbly, 2018, 2022].

Tensor Train Canonical Form. A TT decomposition $\mathcal{S} = \langle\langle \mathcal{G}^1, \mathcal{G}^2, \cdots, \mathcal{G}^{N-1}, \mathcal{G}^N \rangle\rangle \in \mathbb{R}^{d_1 \times \cdots \times d_N}$ is in a canonical format with respect to a fixed index $j \in [N]$ if $\mathbf{G}_{(n)}{}^\top \mathbf{G}_{(n)} = \mathbf{I}_{R_n}$ for all $n < j$, and $\mathbf{G}_{(n)} \mathbf{G}_{(n)}{}^\top = \mathbf{I}_{R_{n-1}}$ for all $n > j$.



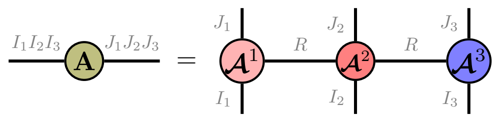
The cores $\mathcal{G}^1, \mathcal{G}^2$ are referred to left-orthogonal, while $\mathcal{G}^4, \mathcal{G}^5$ are referred to as right-orthogonal in the representation above. The core $\mathcal{G}^3$ is called the center of the orthogonality. Note that any TT decomposition can efficiently be converted to canonical form w.r.t. any index $j \in [N]$ by performing a series of QR decompositions on the core tensors.

4.5 TT Decomposition Generalizations

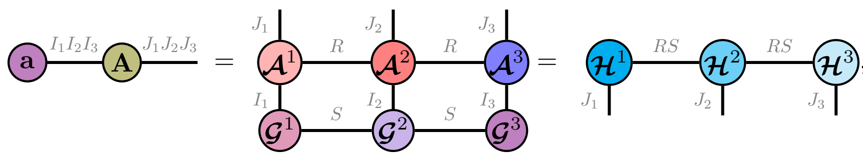
- **Matrix Product Operator decomposition.** As a generalization of the TT decomposition, we introduce the Matrix Product Operator (MPO) decomposition [Oseledets, 2010]. An MPO, also known as a TT-matrix, is a chain of four-way tensors used to represent a matrix. It was originally developed to describe operators acting on multi-body quantum systems. Simply put, an MPO is a method of representing a matrix using tensors. Suppose that we have a matrix of size $\mathbf{A} \in \mathbb{R}^{I_1 I_2 \ldots I_N \times J_1 J_2 \ldots J_N}$. For $n \in [N]$, let $\boldsymbol{\mathcal{A}}^n \in \mathbb{R}^{R_{n-1} \times I_n \times J_n \times R_n}$ with $R_0 = R_N = 1$ and $R_1 = \cdots = R_{N-1} = R$. A rank R MPO decomposition of $\mathbf{A}$ is given by

$$\mathbf{A}_{i_1 i_2 \cdots_N, j_1 j_2 \ldots j_N} = (\boldsymbol{\mathcal{A}}^1)_{i_1, j_1, :} (\boldsymbol{\mathcal{A}}^2)_{:, i_2, j_2, :} \ldots (\boldsymbol{\mathcal{A}}^{N-1})_{:, i_{N-1}, j_{N-1}, :} (\boldsymbol{\mathcal{A}}^N)_{:, i_N, j_N},$$

for all indices $i_1 \in [I_1], \cdots, i_N \in [I_N]$ and $j_1 \in [J_1], \ldots, j_N \in [J_N]$; we will use the notation $\mathbf{A} = \mathrm{MPO}((\boldsymbol{\mathcal{A}}^n)_{n=1}^N)$ to denote the MPO format. The MPO decomposition for a matrix $\mathbf{A} \in \mathbb{R}^{I_1 I_2 I_3 \times J_1 J_2 J_3}$ in a tensor network notation can be represented by:



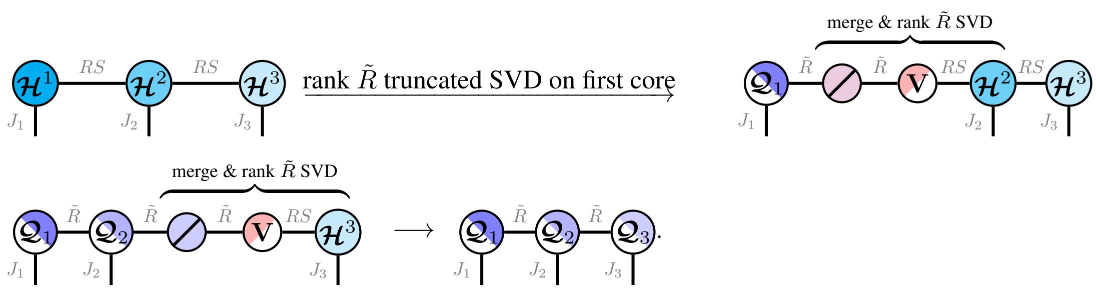
Mat-vec Product. The product between a matrix $\mathbf{A} \in \mathbb{R}^{I_1 I_2 I_3 \times J_1 J_2 J_3}$ and a vector $\mathbf{a} \in \mathbb{R}^{I_1 I_2 I_3}$ can be computed efficiently in the TT format directly by decomposing a vector to a TT vector and a matrix to a TT-matrix, e.g.,



(8)

where the final tensor is a TT vector of rank RS since multiplication of two TT tensors increases the rank to the multiplication of ranks.

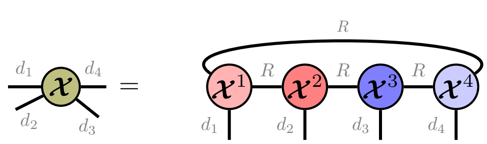
TT-rounding. In the operations on TT format (e.g., summation, inner products, etc.), the rank of a final tensor is increased. To avoid this growth, we can reduce the rank while maintaining the accuracy. For this purpose, we can take the TT tensor obtained by eqn. (4.5) and apply the SVD decomposition as in Theorem 8. To obtain the rank $\tilde{R} \leq R$ TT decomposition, we can use truncated SVD with rank $\tilde{R}$ on the first core of the below tensor:



- **Tensor Ring Decomposition.** The Tensor Ring (TR) decomposition is another generalization of the TT decomposition [Zhao et al., 2016]. Originally introduced in quantum physics, it has recently gained popularity in the machine learning community [Wang et al., 2017, 2018, Yuan et al., 2018]. Although the TR decomposition is known to have certain numerical stability issues, it generally requires less storage and achieves better compression ratios compared to the TT decomposition in practice. Let $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{d_1 \times \cdots \times d_N}$ be an N-th order tensor. For $n \in [N]$, let $\boldsymbol{\mathcal{X}}^n \in \mathbb{R}^{R_{n-1} \times d_n \times R_n}$ with $R_0 = R_1 = \cdots = R_{N-1} = R_N = R$. A rank R *tensor ring decomposition* of the tensor $\boldsymbol{\mathcal{X}}$ is given by

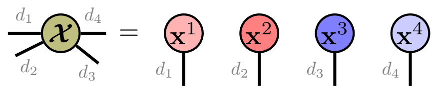
$$\boldsymbol{\mathcal{X}}_{i_1, \ldots, i_N} = \sum_{r_0=1}^{R} \cdots \sum_{r_n=1}^{R} (\boldsymbol{\mathcal{X}}^1)_{r_0, i_1, r_1} (\boldsymbol{\mathcal{X}}^2)_{r_1, i_2, r_2} \ldots (\boldsymbol{\mathcal{X}}^{n-1})_{r_{n-2}, i_{n-1}, r_{n-1}} (\boldsymbol{\mathcal{X}}^n)_{r_{n-1}, i_n, r_n},$$

for all indices $i_1 \in [d_1], \cdots, i_N \in [d_N]$. The TR decomposition can be represented in a tensor network notation, i.e., for a 4-th order tensor:



.

• **Note.** As a special case of tensor train decomposition, we can also obtain a rank-1 decomposition



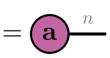
$$= \mathbf{x}^1 \circ \mathbf{x}^2 \circ \mathbf{x}^3 \circ \mathbf{x}^4.$$

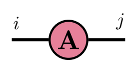
5 Computing Gradients with Tensor Networks

Optimizing tensor networks in a general setting is a key challenge in many research areas. While optimization techniques for two-dimensional matrices have seen significant success, extending these methods to tensor networks in three or more dimensions remains difficult [Liao et al., 2019]. This complexity arises from the substantial computational cost of tensor contractions and the lack of efficient optimization algorithms for higher-dimensional cases. Moreover, manually calculating gradients using the chain rule is feasible only for specially designed and simple tensor network structures [Wang et al., 2011]. In this chapter, we present an elegant and intuitive way to compute (higher-order) derivatives in tensor networks graphical notations efficiently.

5.1 Jacobians

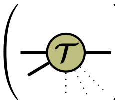
- For $f : \mathbb{R}^n \mapsto \mathbb{R}$ and $g : \mathbb{R}^n \mapsto \mathbb{R}^p$ the gradient of f and the Jacobian of g are respectively,

Gradient of f $\quad \nabla_\theta f = \left[\frac{\partial f(\theta)}{\partial \theta_1}, \frac{\partial f(\theta)}{\partial \theta_2}, \ldots, \frac{\partial f(\theta)}{\partial \theta_n}\right]^\top =$  For each $\theta \in \mathbb{R}^n$,

Jacobian of g $\quad \frac{\partial g(\theta)}{\partial \theta} = \left(\frac{\partial g(\theta)_i}{\partial \theta_j}\right)_{i,j} =$  For each $\theta \in \mathbb{R}^n$.

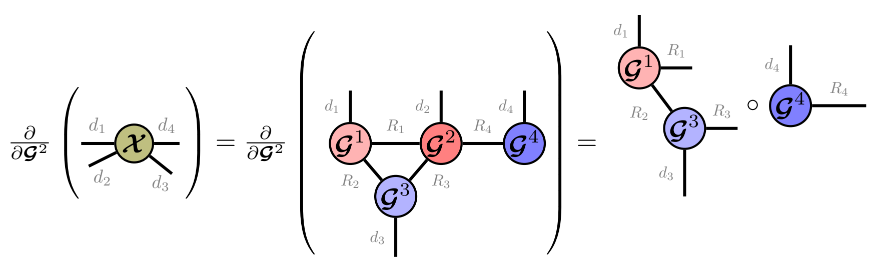
- **Jacobian of Tensor Networks.** If $f : \mathbb{R}^{n_1 \times \cdots \times n_N} \mapsto \mathbb{R}^{m_1 \times \cdots \times m_M}$, then the Jacobian tensor of f for each $\theta \in \mathbb{R}^{n_1 \times \cdots \times n_N}$, is of size $\frac{\partial f}{\partial \theta} \in \mathbb{R}^{m_1 \times \cdots \times m_M \times n_1 \times \cdots \times n_N}$ and defined by

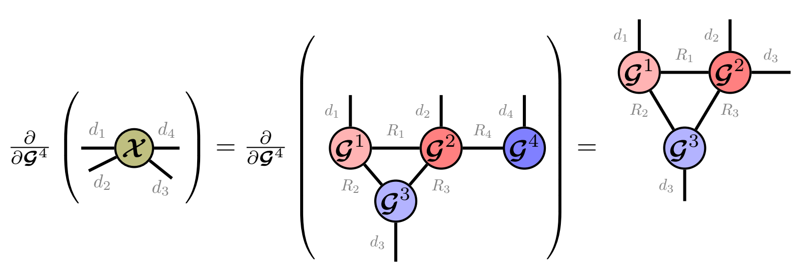
$$\left(\frac{\partial f}{\partial \theta}\right)_{i_1,\ldots,i_M,j_1,\ldots,j_N} = \frac{\partial f(\theta)_{i_1,\ldots,i_M}}{\partial \theta_{j_1,\ldots,j_N}} = \Bigg(\quad \Bigg)_{i_1,\ldots,i_M,j_1,\ldots,j_N}.$$



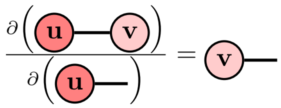
Theorem 9. *Let $\mathcal{T}$ be a tensor given as a tensor network, where $\mathcal{G}$ is a core tensor appearing only once in the tensor network. Then $\frac{\partial \mathcal{T}}{\partial \mathcal{G}}$ is obtained by removing $\mathcal{G}$ from the tensor network of $\mathcal{T}$.*

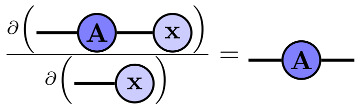
Examples.

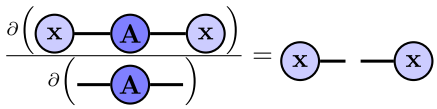


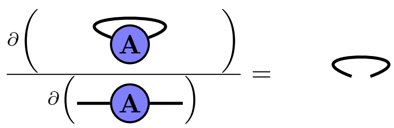


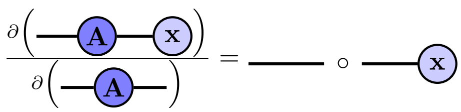
Some Identities.

1. $\frac{\partial\langle \mathbf{u},\mathbf{v}\rangle}{\partial \mathbf{u}} =$  $= \mathbf{v}$

2. $\frac{\partial \mathbf{A}\mathbf{x}}{\partial \mathbf{x}} =$  $= \mathbf{A}$

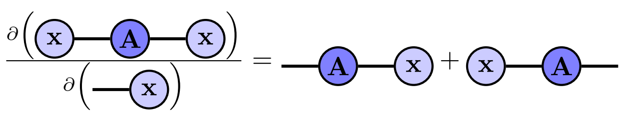
3. $\frac{\partial \mathbf{x}^\top \mathbf{A}\mathbf{x}}{\partial \mathbf{A}} =$  $= \mathbf{x} \circ \mathbf{x}$

4. $\frac{\partial\, \mathrm{tr}(\mathbf{A})}{\partial \mathbf{A}} =$  $= \mathbf{I}$

5. $\frac{\partial \mathbf{A}\mathbf{x}}{\partial \mathbf{A}} =$  $= \mathbf{I} \circ \mathbf{x}$

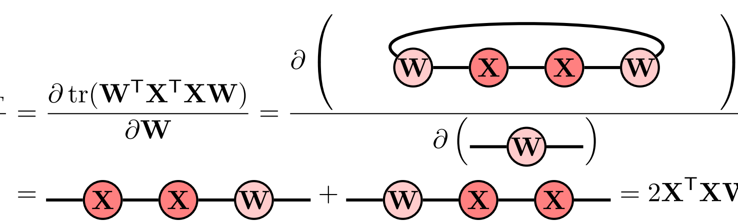
Theorem 10. *Let $\mathcal{T}$ be a tensor network where $\mathcal{G}$ is a core tensor. If $\mathcal{G}$ appears k times in the tensor network of $\mathcal{T}$, then $\frac{\partial \mathcal{T}}{\partial \mathcal{G}}$ is obtained by summing k copies of the tensor network of $\mathcal{T}$, where the different occurrence of $\mathcal{G}$ is removed in each copy.*

Examples.

- $\frac{\partial \mathbf{x}^\top \mathbf{A}\mathbf{x}}{\partial \mathbf{x}} =$  $= \mathbf{A}\mathbf{x} + \mathbf{A}^\top\mathbf{x} = (\mathbf{A}\mathbf{x} + \mathbf{A}^\top)\mathbf{x}$

- $\mathbf{X} \in \mathbb{R}^{n\times m},\ \mathbf{W} \in \mathbb{R}^{\mathbf{W}\in\mathbb{R}^{m\times n}}$

$$\frac{\partial \|\mathbf{X}\mathbf{W} - \mathbf{Y}\|_F^2}{\partial \mathbf{W}} = \frac{\partial\, \mathrm{tr}(\mathbf{W}^\top\mathbf{X}^\top\mathbf{X}\mathbf{W})}{\partial \mathbf{W}} =$$



$= 2\mathbf{X}^\top\mathbf{X}\mathbf{W}.$

6 Probability and Random Vectors

7 Tensor Networks for Machine Learning

References

Brett W Bader and Tamara G Kolda. Algorithm 862: Matlab tensor classes for fast algorithm prototyping. *ACM Transactions on Mathematical Software (TOMS)*, 32(4):635–653, 2006.

Jacob Biamonte and Ville Bergholm. Tensor networks in a nutshell. *arXiv preprint arXiv:1708.00006*, 2017.

Lam Chi-Chung, P Sadayappan, and Rephael Wenger. On optimizing a class of multi-dimensional loops with reduction for parallel execution. *Parallel Processing Letters*, 7(02):157–168, 1997.

Glen Evenbly. Gauge fixing, canonical forms, and optimal truncations in tensor networks with closed loops. *Physical Review B*, 98(8):085155, 2018.

Glen Evenbly. A practical guide to the numerical implementation of tensor networks i: Contractions, decompositions and gauge freedom. *arXiv preprint arXiv:2202.02138*, 2022.

Christopher J Hillar and Lek-Heng Lim. Most tensor problems are np-hard. *Journal of the ACM (JACM)*, 60(6):1–39, 2013.

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.

Sebastian Holtz, Thorsten Rohwedder, and Reinhold Schneider. The alternating linear scheme for tensor optimization in the tensor train format. *SIAM Journal on Scientific Computing*, 34(2):A683–A713, 2012.

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

Joseph B Kruskal. Rank, decomposition, and uniqueness for 3-way and n-way arrays. In *Multiway data analysis*, pages 7–18. 1989.

Hai-Jun Liao, Jin-Guo Liu, Lei Wang, and Tao Xiang. Differentiable programming tensor networks. *Physical Review X*, 9(3):031041, 2019.

Román Orús. A practical introduction to tensor networks: Matrix product states and projected entangled pair states. *Annals of physics*, 349:117–158, 2014.

Ivan V Oseledets. Approximation of 2ˆd\times2ˆd matrices using tensor decomposition. *SIAM Journal on Matrix Analysis and Applications*, 31(4):2130–2145, 2010.

Roger Penrose et al. Applications of negative dimensional tensors. *Combinatorial mathematics and its applications*, 1: 221–244, 1971.

Ledyard R Tucker. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3):279–311, 1966.

Ling Wang, Ying-Jer Kao, and Anders W Sandvik. Plaquette renormalization scheme for tensor network states. *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics*, 83(5):056703, 2011.

Wenqi Wang, Vaneet Aggarwal, and Shuchin Aeron. Efficient low rank tensor ring completion. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5697–5705, 2017.

Wenqi Wang, Yifan Sun, Brian Eriksson, Wenlin Wang, and Vaneet Aggarwal. Wide compression: Tensor ring nets. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9329–9338, 2018.

Longhao Yuan, Jianting Cao, Xuyang Zhao, Qiang Wu, and Qibin Zhao. Higher-dimension tensor completion via low-rank tensor ring decomposition. In *2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 1071–1076. IEEE, 2018.

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