Universitat de Barcelona
Facultat de Matemàtiques i Informàtica

GRAU DE MATEMÀTIQUES

Treball final de grau

OPTIMAL LOW-RANK APPROXIMATION USING TENSOR NETWORK STRUCTURE SEARCH

Autor: Aran Roig

Director: Dr. Nahuel Statuto
Realitzat a: Departament de Matemàtiques i Informàtica

Barcelona, May 15, 2025

Abstract

Tensor network structure search has been interesting research topic since the raise on complexity of deep learning models and quantum mechanics. This is an Undegraduate Thesis whose main goal is to give an automated search of an optimal tensor network structure for representing a given tensor with some fixed error.

For these purpose we first give an introduction to tensors, tensor networks and then we present some examples of well studied tensor networks, including Tucker decomposition, Tensor Train decomposition, Tensor Ring decomposition and Fully Connected Tensor Network decomposition.

Then with some practical experiments we demonstrate that it is possible to find more optimized structures without significant losses on performance and accuracy, and finally we will present an algorithm for finding these optimized structures.

Resum

La recerca de l'estructura òptima de xarxes de tensors ha estat un tema d'interès des de l'augment en la complexitat dels models d'aprenentatge profund i de la mecànica quàntica. Aquest és un treball de final de grau que té com a objectiu principal oferir una cerca automatitzada d'una estructura òptima de xarxa de tensors per representar un tensor donat amb un error fixat.

Per aconseguir aquest propòsit, primer oferim una introducció als tensors, a les xarxes de tensors, i tot seguit presentem alguns exemples de xarxes de tensors ben estudiades, incloent-hi la descomposició de Tucker, la descomposició en tren de tensors (Tensor Train), la descomposició en anell de tensors (Tensor Ring) i la descomposició de xarxa de tensors totalment connectada.

A continuació, amb alguns experiments pràctics, demostrem que és possible trobar estructures més optimitzades sense pèrdues significatives en el rendiment i la precissió de la representació, i finalment presentem un algoritme per trobar aquestes estructures optimitzades.

2020 Mathematics Subject Classification. 11G05, 11G10, 14G10

Agraïments

Vull agrair a ...

Contents

Chapter 1

Introduction

Neural networks are a set of dependant non-linear functions which each individual function corresponds to a neuron (or perceptron). In a fully connected neural network, each neuron applies a linear transformation to the input vector through a weight of matrices.

The output of a neuron j in some layer is given by the formula:

$$y_j(x) = f\left(\sum_{i=1}^{N} w_{ji}x_i + b_j\right)$$

Where w_{ji} is the weight assigned to the connection between the neuron i of the previous layer with the neuron j of the current layer, f is an activation function and b_j is some bias of the neuron j.

We can represent the weights of a fully connected neural network as a matrix, that we call the **weight matrix** since the recent rise on complexity of neural networks, weight matrices can be very large and there is a lot of interest on compressing weight matrices without sacrificing accuracy or performance.

Since we can reshape the weight matrix into a tensor, in this thesis, we will aim to compress a tensor using tensor networks, a concept that originates from the study of many-body quantum systems [6] and recently has gained traction on machine learning.

The objective of this thesis is given a tensor T, finding the tensor network structure that is represented by tensors of smaller rank (called core tensors) that, when the whole network is evaluated, represent T well enough given a fixed error ϵ. We will also minimize the computational cost when it comes to contracting the whole network for retrieving the representation of the original tensor.

We will see that finding the best structure is an integer programming problem, that is NP-hard. We will deduce and give an algorithm that finds a locally best structure by using program synthesis, and finally, we will compare this structure with some other more studied cases of tensor networks, as Tensor Train networks, Tensor Ring networks and fully connected tensor networks

1.1 Thesis structure

First we will present some preliminaries about tensor algebra. Then, we will introduce the diagrammatic notation made by Roger Penrose in the earlies 1970. We will use it for representing both matrix product states and tensor networs.

Then, we will give a formal introduction in tensor networks, based in most part from [13]. We will introduce also well studied tensor networks such as the Tucker decomposition, the tensor train decomposition, the tensor ring decomposition, the fully connected tensor network decomposition.

We will make special emphasis on tree tensor networks, since they have some nice properties that we can later use for splitting tensor networks. We will also describe the alternating least squares algorithm applied to tensor networks for finding approximated states for any tensor network.

Finally, we will explain some algorithms for finding optimal tensor network structures. We will give an algorithm for finding a general tensor network structure based following [4] and we will also describe an algorithm for finding an optimal tensor network tree structure using program synthesis following mainly from the paper [2]

Chapter 2

Preliminaries

2.1 Graph Theory

Definition 2.1 (Graph)**.** [12] A **graph** G is defined as a tuple $G = (V, E)$ where $V = V(G)$ is a set of elements called **vertices** and $E = E(G) \subset \{\{u, v\} : u, v \in V\}$ is a set of elements called **edges**.

Definition 2.2 (Directed Graph)**.** *A* ***directed graph*** *D is a tuple $D = (V, \bar{E})$ where $V = V(D)$ are its vertices and $\bar{E} = \bar{E}(D) \subset \{(u, v) : u, v \in V\}$*

Definition 2.3. *Given a directed graph $G = (V, \bar{E})$ and a vertex $i \in V$ we define*

$$\mathrm{IN}(i) = \{j \in V : (j, i) \in \bar{E}\} \qquad \mathrm{OUT}(i) = \{j \in V : (i, j) \in \bar{E}\}$$

In this chapter we will introduce the basics of tensor algebra and we will present some results that will be useful in the following chapters

2.2 Basic tensor definitions

We will denote $\mathbb{V}_1, \dots, \mathbb{V}_n$ as finite vector spaces over a field $\mathbb{K}$ ($\mathbb{C}$ if unspecified) of dimension $\dim \mathbb{V}_i = N_i \ \forall i = 1, \dots, n$. We denote $\mathcal{B}_1, \dots, \mathcal{B}_n$ with $\mathcal{B}_i = \{e_1^{(i)}, \dots, e_{N_i}^{(i)}\}, i = 1, \dots, n$ the canonical basis of $\mathbb{V}_1, \dots, \mathbb{V}_n$

Definition 2.4 (Tensor)**.** *A* ***tensor*** *is a multilineal map $T : \mathbb{V}_1 \times \cdots \times \mathbb{V}_n \to \mathbb{K}$*

Definition 2.5 (Order of a tensor)**.** *Given a tensor $T \in \mathbb{V}_1 \times \cdots \times \mathbb{V}_n \to \mathbb{K}$ We define the* ***order*** *of the tensor T as n.*

We will allow that some $\mathbb{V}_i$ are dual spaces of some other vector spaces $\mathbb{W}_i$, i.e $\mathbb{V}_i = \mathbb{W}_i^*$.

Definition 2.6. *Let $T \in \mathbb{V}_1 \times \cdots \times \mathbb{V}_p \times \mathbb{W}_1^* \times \cdots \times \mathbb{W}_q^* \to \mathbb{K}$. We say that the tensor T is p-times covariant and q-times contravariant.*

Definition 2.7 (Tensor product)**.** *Let L be the vector space generated by the base $V_1 \times \cdots \times V_n$, i.e the set of linear combinations of the elements $(v_1, \dots, v_n), v_i \in \mathbb{V}_i$. Let $\mathcal{R}$ be the linear subspace of L generated by the relation R defined by:*

$$(v_1, \dots, \alpha v_i, \dots, v_n) \sim \alpha(v_1, \dots, v_n) \ \forall i = 1, \dots, n, \forall \alpha \in \mathbb{K}$$

$$(v_1, \ldots, v_i + u_i, \ldots, v_n) \sim (v_1, \ldots, v_i \ldots, v_n) + (v_1, \ldots, u_i, \ldots, v_n)\ \forall i = 1, \ldots, n$$

The tensor product $\mathbb{V}_1 \otimes \cdots \otimes \mathbb{V}_n$ is defined as the quotient $L/\mathcal{R}$ and its called ***tensor product space****. The image of $(v_1, \ldots, v_n)$ by the quotient is denoted by $v_1 \otimes \cdots \otimes v_n$*

The following theorem gives us a correspondance between each tensor $T : \mathbb{V}_1 \times \cdots \times \mathbb{V}_n \to \mathbb{K}$ and each element of $\mathbb{V}_1 \otimes \cdots \otimes \mathbb{V}_n$:

Theorem 2.8 (Universal property of the tensor product)**.** [9] The tensor product of two vector spaces $\mathbb{V} \otimes \mathbb{W}$ for every bilinear map $h : \mathbb{V} \times \mathbb{W} \to X$ there exists an unique bilinear map $\tilde{h} : \mathbb{V} \otimes \mathbb{W} \to X$ such that the following diagram commutes:

$$\begin{tikzcd}
\mathbb{V} \times \mathbb{W} \arrow[r, "\varphi"] \arrow[dr, "h"'] & \mathbb{V} \otimes \mathbb{W} \arrow[d, "\tilde{h}"] \\
 & X
\end{tikzcd}$$

We can construct a basis for $\mathbb{V}_1 \otimes \cdots \otimes \mathbb{V}_n$. We define

$$\mathcal{B}_\otimes = \{e^{(1)}_{i_1} \otimes \cdots \otimes e^{(n)}_{i_n} : 1 \leqslant i_j \leqslant N_j, 1 \leqslant j \leqslant n\}$$

Where $\{e^{(i)}_1, e^{(i)}_2, \ldots, e^{(i)}_{N_i}\}$ is the canonical basis for $\mathbb{V}_i$. Constructed this way, $\mathcal{B}_\otimes$ is a (canonical) basis of $\mathbb{V}_1 \otimes \cdots \otimes \mathbb{V}_n$.

Remark 2.9. The dimension of $\mathbb{V}_1 \otimes \cdots \otimes \mathbb{V}_n$ is $\dim \mathbb{V}_1 \cdot \dim \mathbb{V}_2 \cdots \dim \mathbb{V}_n$ and its elements can be expressed as

$$T = \sum_{s_1,\ldots,s_n}^{N_1,\ldots,N_n} T_{s_1,\ldots,s_n} \cdot e^{(1)}_{s_1} \otimes \cdots \otimes e^{(n)}_{s_n} \tag{2.2.1}$$

We will define the size of the tensor T as $\mathrm{Size}(T) = \dim \mathbb{V}_1 \cdot \dim \mathbb{V}_2 \cdots \dim \mathbb{V}_n$

Definition 2.10 (Tensor product)**.** *Given two tensors $T \in \mathbb{V}_1 \otimes \cdots \otimes \mathbb{V}_n$ and $U \in \mathbb{W}_1 \otimes \cdots \otimes \mathbb{W}_m$ with $\dim \mathbb{V}_i = N_i$, $i = 1, \ldots, n$ and $\dim \mathbb{W}_j = M_j$, $j = 1, \ldots, m$ and canonical basis $\{v^{(i)}_1, \ldots, v^{(i)}_{N_i}\}$ for each $\mathbb{V}_i$ and $\{w^{(j)}_1, \ldots, w^{(j)}_{M_j}\}$ for each $\mathbb{W}_j$ we define the tensor product $T \otimes U$ as*

$$T \otimes U = \sum_{i_1,\ldots,i_n}^{N_1,\ldots,N_n} \sum_{j_1,\ldots,j_m}^{M_1,\ldots,M_m} T_{i_1,\ldots,i_n} U_{j_1,\ldots,j_m} \cdot v^{(1)}_{i_1} \otimes \cdots \otimes v^{(n)}_{i_n} \otimes w^{(1)}_{j_1} \otimes \cdots \otimes w^{(m)}_{j_m}$$

Note that $T \otimes U \in \mathbb{V}_1 \otimes \cdots \otimes \mathbb{V}_n \otimes \mathbb{W}_1 \otimes \cdots \otimes \mathbb{W}_m$

From Remark 2.9, a tensor $T \in \mathbb{V}_1 \otimes \cdots \otimes \mathbb{V}_n$ can be identified as a "n-dimensional array" $\mathcal{T}$, i.e, a discrete function

$$\begin{aligned}
\mathcal{T} : \prod_{i=1}^{n} \{1, \ldots, N_i\} &\longrightarrow \mathbb{K} \\
T(i_1, \ldots, i_n) &\longmapsto T_{i_1,\ldots,i_n}
\end{aligned}$$

From now on, we will write the discrete function $\mathcal{T}$ as an element of $\mathbb{K}^{N_1 \times \cdots \times N_n}$ [14]. of T as $\mathcal{T}$. Now we will define some definitions from the underlying definition of the tensor viewed as a discrete function:

Now we will define some definitions that will help us establish a formalized method of mapping n-dimensional arrays with vectors and matrices:

Definition 2.11 (Linearization). *Fixed $N_1, \dots, N_n \in \mathbb{N}$, given $i_1, \dots, i_n \in \mathbb{N}$ such that $1 \leqslant i_1, \leqslant N_1, \dots, 1 \leqslant i_n \leqslant N_n$, we define the Linearization of the indices $i_1, \dots, i_n$ as the mapping $\prod_{i=1}^n \{1, \dots, N_i\} \to \{1, \dots, \prod_{i=1}^n N_i\}$:*

$$\overline{i_1, i_2, \dots, i_n} = \sum_{j=2}^{n} \left((i_j - 1) \prod_{k=1}^{j} N_k \right) + i_1$$

Lemma 2.12. *The linearization mapping is bijective*

Definition 2.13 (Vectorization). *Given a tensor $\mathcal{T} \in \mathbb{K}^{N_1 \times \dots \times N_n}$, we define the **vectorization** of $\mathcal{T}$ as the first order tensor $\mathcal{V} \in \mathbb{K}^{N_1 N_2 \cdots N_n}$ defined by:*

$$\mathcal{V}(\overline{i_1 i_2 \dots i_n}) = \mathcal{T}(i_1, i_2, \dots, i_n)$$

We will write $\mathcal{V} = \operatorname{vec} \mathcal{T}$

Definition 2.14 (Tensor unfolding). *Let $\mathcal{T} \in \mathbb{R}^{N_1 \times \dots \times N_n}$, $n \geqslant 2$, $1 \leqslant d \leqslant n$ and p a permutation of the vector $(1, 2, \dots, n)$. We define the **generalized unfolding** of the tensor $\mathcal{T}$ as the 2nd-order tensor $\mathcal{U} \in \mathbb{R}^{\prod_{i=1}^d N_{p_i} \times \prod_{i=d+1}^n N_{p_i}}$:*

$$\mathcal{U}(\overline{i_{p_1}, \dots, i_{p_d}}, \overline{i_{p_{d+1}}, \dots, i_{p_n}}) = \mathcal{T}(i_1, \dots, i_n)$$

We will write $\mathcal{U} = \operatorname{unfold}(\mathcal{T}, (p_1, \dots, p_d), (p_{d+1}, \dots, p_n))$. We also define $\operatorname{unfold}_d T := \operatorname{unfold}(\mathcal{T}, (1, \dots, d), (d+1, \dots, n))$

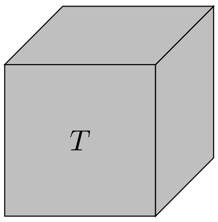


Figure 2.1: Tensor unfolding

Definition 2.15 (Tensor slices). *Let $\mathcal{T} \in \mathbb{K}^{N_1 \times \dots \times N_n}$. Consider $S \subset \{1, \dots, n\}$ a subset of modes (dimensions) of $\mathcal{T}$. Let $\mathbf{i}_S = (i_k)_{k \in S}$. We define the **slice of** $\mathcal{T}$ as the tensor $\mathcal{T}_{\mathbf{i}_S}$ of order $n - \#S$*

$$\begin{aligned} \mathcal{T}_{\mathbf{i}_S} : \prod_{k \notin S} \{1, \dots, N_k\} &\longrightarrow \mathbb{K} \\ \mathcal{T}_{\mathbf{i}_S}(i_{j_1}, \dots, i_{j_m}) &\longmapsto \mathcal{T}(i_1, \dots, i_n) \end{aligned}$$

Where $\{j_1, \dots, j_m\} = \{1, \dots, n\} \setminus S$.

Sometimes we will also implicitly specify $\mathbf{i}_S$ by writing $\mathcal{T}(a_1, \dots, a_n)$ and replacing a_j with i_j if $j \in S$ and ":" otherwise

We will denote $\mathcal{T}^{(m)} := \mathcal{T}(:, \dots, :, i_m, :, \dots, :)$ and $\mathcal{T}^{(\neq m)} := \mathcal{T}(i_1, \dots, i_{m-1}, :, i_{m+1}, \dots, i_n)$

Example 2.16. Consider $\mathcal{M} \in \mathbb{K}^{N_1 \times N_2}$ a second order tensor. We can see this tensor as a bidimensional array (matrix). The slice $\mathcal{M}(i, :)$ results in the i-th row of $\mathcal{M}$. The slice $\mathcal{M}(:, j)$ results in the j-th column of $\mathcal{M}$

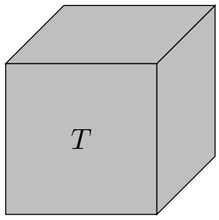


Figure 2.2: Tensor slices

Example 2.17. Consider a 4th-order tensor $\mathcal{T} \in \mathbb{K}^{N_1 \times N_2 \times N_3 \times N_4}$. Fixed i_2, i_3, the tensor slice of 2th-order $\mathcal{T} = \mathcal{A}(:, i_2, i_3, :) \in \mathbb{K}^{N_1 \times N_4}$ with its entries defined by $\mathcal{T}(i_1, i_4) = \mathcal{A}(i_1, i_2, i_3, i_4)$

Definition 2.18 (Rank of a tensor)**.** *We say that a tensor is of rank r and we write* $\operatorname{rank} T = r$ *with $r \in \mathbb{N}$ being the minimum value such that we can write T as*

$$T = \sum_{p=1}^{r} v_p^{(1)} \otimes \cdots \otimes v_p^{(n)}$$

where $v_1^{(i)}, \ldots, v_r^{(i)} \in \mathbb{V}_i, i = 1, \ldots, n$

One can easily see that $\operatorname{rank} T \leqslant \prod_{i=1}^{n} N_i$. Unlike matrices, determining the rank of a tensor is an NP-hard problem. [3]. Even finding the maximum rank, (i.e determining $\max_{T \in \mathbb{V}_1 \otimes \cdots \otimes \mathbb{V}_n} \operatorname{rank} T$) this is still an unresolved problem. We will now present an slightly better upper bound for the tensor rank:

Proposition 2.19.

$$\operatorname{rank} T \leqslant \left\lfloor \frac{\prod_{i=1}^{n} N_i}{\sum_{i=1}^{n} N_i} \right\rfloor \tag{2.2.2}$$

Proof. Let $r = \operatorname{rank} T$. We can write $T = \sum_{p=1}^{r} v_p^{(1)} \otimes \cdots \otimes v_p^{(n)}$. Now, each term of this sum has $\sum_{i=1}^{n} N_i$ adjustable parameters, since each $v_p^{(i)}$ is a vector of $\mathbb{V}_i$ with its dimension being N_i. So, in total we will have $r \sum_{i=1}^{n} N_i$ adjustable parameters in our decomposition. Since our tensor T is completly determined by $\prod_{i=1}^{n} N_i$ parameters, we can impose that $r \sum_{i=1}^{n} N_i \leqslant \prod_{i=1}^{n} N_i$ $\square$

Chapter 3

Tensor networks

3.1 Penrose Notation

The Penrose an intuitive graphical language to represent tensor contractions that dates back from at least the early 1970s [8]

Given an nth-order tensor $\mathcal{T} \in \mathbb{K}^{N_1 \times \cdots \times N_n}$ the way we draw it using the Penrose notation is as a circle with as many edges as the order of the tensor, as seen in fig. 3.1

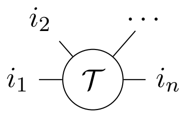


Figure 3.1: Representation of a tensor $\mathcal{T} \in \mathbb{K}^{N_1 \times \cdots \times N_n}$ using the Penrose notation

A lot of times we will not write explicitly the name of the indexes, since we only care about their order. The order of the indexes will be determined by their orientation respect to the circle: to get the order we will start from the left and then following a clockwise rotation. The order in which we encounter the edges will be the order of the indexes. For example, in fig. 3.1, the order would be $i_1, i_2, \ldots, i_n$

3.2 Tensor contraction

We would want to generalize the concept of vector dot product, or matrix product or the trace operation into tensors. The tensor contraction gives us a generalization of all of the above. The concept of the tensor contraction basically arises from the canonical pairing of a vector space and its dual. Since tensors are elments of $T \in \mathbb{V}_1 \otimes \cdots \otimes \mathbb{V}_n \otimes \mathbb{W}_1 \otimes \cdots \otimes \mathbb{W}_m$ with for each $\mathbb{W}_j$ being the dual space of some space $\mathbb{V}_i$, one can contract these tensor by pairing these two spaces.

Formally, the contraction of two tensors $\mathcal{X}, \mathcal{Y}$ is the image of a fixed contraction $\mathcal{C}_k^l$ map where k is an index (or edge) of the first tensor $\mathcal{X}$ and l is another index of the second tensor $\mathcal{Y}$:

Definition 3.1. [1] Let $T \in \left(\bigotimes_{i=1}^{k-1} \mathbb{V}_i \otimes \mathbb{V}_k \otimes \bigotimes_{i=k+1}^{p} \mathbb{V}_i\right) \otimes \left(\bigotimes_{i=1}^{l-1} \mathbb{W}_i^* \otimes \mathbb{W}_l^* \otimes \bigotimes_{i=l+1}^{q} \mathbb{W}_i^*\right)$

Figure 3.2: Representation in the Penrose notation of the contraction between two tensors $\mathcal{X} \in \mathbb{K}^{N_1 \times N_2 \times N_3}, \mathcal{Y} \in \mathbb{K}^{M_1 \times M_2 \times M_3}$ by their indices i_3 and j_1 with $N_1, N_2, N_3, M_1, M_2, M_3 \in \mathbb{N}$ and $N_3 = M_1$

With $\mathbb{V}_k = \mathbb{W}_l$. Consider the mapping

$$\mathcal{C}_k^l : \left(\bigotimes_{i=1}^{p} \mathbb{V}_i\right) \otimes \left(\bigotimes_{i=1}^{q} \mathbb{W}_i^*\right) \longrightarrow \left(\bigotimes_{i=1}^{k-1} \mathbb{V}_i \otimes \bigotimes_{i=k+1}^{p} \mathbb{V}_i\right) \otimes \left(\bigotimes_{i=1}^{l-1} \mathbb{W}_i^* \otimes \bigotimes_{i=l+1}^{q} \mathbb{W}_i^*\right)$$

$$\mathcal{C}_k^l \left(\bigotimes_{i=1}^{p} v_i \otimes \bigotimes_{i=1}^{q} f_q\right) = \left(\bigotimes_{i=1}^{k-1} v_i \otimes \bigotimes_{i=k+1}^{p} v_i \otimes \bigotimes_{i=1}^{l-1} f_i \otimes \bigotimes_{i=l+1}^{q} f_i\right) f_l(v_k)$$

$\mathcal{C}_k^l$ is defined as the tensor contraction mapping of $\mathbb{V}_1 \otimes \cdots \otimes \mathbb{V}_p \otimes \mathbb{V}_{p+1}^* \otimes \cdots \otimes \mathbb{V}_q^*$ by the indices (k, l). We call $\mathcal{C}_k^l(T)$ the contraction of T by indices (k, l). In other words, we apply the dual space $\mathbb{V}_k^*$ with $\mathbb{V}_k$.

Now we will introduce the definition of the contraction between two tensors by defining a contraction over the tensor product of the two original tensors.

Definition 3.2. *Let* $X \in \left(\bigotimes_{i=1}^{k-1} \mathbb{V}_i \otimes \mathbb{V}_k \otimes \bigotimes_{i=k+1}^{p} \mathbb{V}_i\right)$ *and* $Y \in \left(\bigotimes_{i=1}^{l-1} \mathbb{W}_i \otimes \mathbb{W}_l^* \otimes \bigotimes_{i=l+1}^{q} \mathbb{W}_i\right)$ *with* $\mathbb{V}_k = \mathbb{W}_l$. *We define the contraction between* X *and* Y *by the indices* (k, l) *as*

$$X \times_k^l Y := \mathcal{C}_k^l(X \otimes Y)$$

We will represent the contraction between two tensors as their representation in the Penrose notation with the edges that represent the indexes that are contracting by joining them, as seen in fig. 3.2.

Fixing bases for $\mathbb{V}_1, \ldots, \mathbb{V}_p, \mathbb{W}_1, \ldots, \mathbb{W}_q$ and representing X, Y as discrete functions by its representations in those basis, we get a way for computing $\mathcal{C}_k^l(X \otimes Y)$:

Definition 3.3 (Base dependant tensor contraction)**.** [16] Given $\mathcal{X}, \mathcal{Y}$ as the discrete function representations of X, Y from earlier, we can write $\mathcal{C}_k^l(X \otimes Y)$ element-wise as:

$$\begin{aligned} &\mathcal{C}_k^l(i_1, \ldots, i_{k-1}, i_{k+1}, \ldots, i_p, j_1, \ldots, j_{l-1}, j_{l+1}, \ldots, j_q) \\ &= \sum_{s=1}^{N_k} \mathcal{X}(i_1, \cdots, i_{k-1}, s, i_{k+1}, \cdots, i_p) \mathcal{Y}(j_1, \cdots, j_{l-1}, s, j_{l+1}, \cdots, j_q) \end{aligned} \tag{3.2.1}$$

Example 3.4. The tensor contraction for two tensors of order 2 $X \in \mathbb{K}^{N_1 \times N_2}, Y \in \mathbb{K}^{N_2 \times N_3}$ over one edge of each tensor yields the matrix multiplication. Applying eq. (3.2.1) we get that

$$(\mathcal{X} \times_2^2 \mathcal{Y})(i_1, j_2) = \sum_{s=1}^{N_2} X(i_1, s) Y(s, j_2)$$

And it is equivalent to the conventional matrix product. Visually, using the Penrsone notation we have:

Example 3.5. Suppose that $X \in \mathbb{R}^{3\times3\times2}, Y \in \mathbb{R}^{2\times3\times3}$ with their representation in some given basis

$$X = \left[\begin{pmatrix} 1 & 0 & -1 \\ 0 & 2 & -2 \\ -1 & 3 & 1 \end{pmatrix} \quad \begin{pmatrix} 1 & 1 & 0 \\ 2 & 0 & 2 \\ 0 & 3 & 0 \end{pmatrix}\right]$$

$$Y = \left[\begin{pmatrix} 1 & 0 & -1 \\ 0 & 2 & -2 \end{pmatrix} \quad \begin{pmatrix} 1 & 1 & 0 \\ 2 & 0 & 2 \end{pmatrix} \quad \begin{pmatrix} 0 & 1 & 2 \\ 1 & 0 & 2 \end{pmatrix}\right]$$

The contraction described in fig. 3.2 would yield the tensor $\mathcal{Z} = \mathcal{X} \times_3^1 \mathcal{Y} \in \mathbb{R}^{3\times3\times3\times3}$ with each element defined as:

$$\mathcal{Z}(i_1, i_2, j_2, j_3) = \sum_{s=1}^{2} \mathcal{X}(i_1, i_2, s)\mathcal{Y}(s, j_2, j_3)$$

Therefore,

$$\mathcal{Z} = \begin{pmatrix}
\begin{pmatrix} 1 & 2 & -3 \\ 3 & 1 & 2 \\ 1 & 1 & 4 \end{pmatrix} & \begin{pmatrix} 0 & 2 & -2 \\ 2 & 0 & 2 \\ 1 & 0 & 2 \end{pmatrix} & \begin{pmatrix} -1 & 0 & 1 \\ -1 & -1 & 0 \\ 0 & -1 & -2 \end{pmatrix} \\
\begin{pmatrix} 0 & 4 & -4 \\ 4 & 0 & 4 \\ 2 & 0 & 4 \end{pmatrix} & \begin{pmatrix} 2 & 0 & -2 \\ 2 & 2 & 0 \\ 0 & 2 & 4 \end{pmatrix} & \begin{pmatrix} -2 & 4 & 6 \\ 2 & -2 & 4 \\ 2 & -2 & 0 \end{pmatrix} \\
\begin{pmatrix} -1 & 0 & 1 \\ -1 & -1 & 0 \\ 0 & -1 & -2 \end{pmatrix} & \begin{pmatrix} 3 & 6 & -9 \\ 9 & 3 & 6 \\ 3 & 3 & 12 \end{pmatrix} & \begin{pmatrix} 1 & 0 & -1 \\ 1 & 1 & 0 \\ 0 & 1 & 2 \end{pmatrix}
\end{pmatrix}$$

Matrix tensor contraction

Note that eq. (3.2.1) is almost in the form of matrix product. If we reshape accordingly the tensors $\mathcal{X}$ and $\mathcal{Y}$, we can compute $\mathcal{C}_l^k(X \otimes Y)$ as a matrix product.

Corollary 3.6. *Let* $X \in \mathbb{K}^{N_1\times\cdots\times N_k\times\cdots\times N_n}, Y \in \mathbb{K}^{M_1\times\cdots\times M_l\times\cdots\times M_m}$ *with* $1 \leqslant k \leqslant n$, $1 \leqslant l \leqslant m$, $N_k = M_l$. *The matrix product*

$$\mathcal{X}(\overline{i_1, \ldots, i_{k-1}}, \overline{i_k, \ldots, i_n}) \cdot \mathcal{Y}(\overline{j_1, \ldots, j_{l-1}}, \overline{j_l, \ldots, j_m})$$

results in a representation of $\mathcal{T} = \mathcal{X} \times_k^l \mathcal{Y}$

Example 3.7. Following from Example 3.5, we would write $X \in \mathbb{R}^{9\times2}$ and $Y \in \mathbb{R}^{2\times9}$ as:

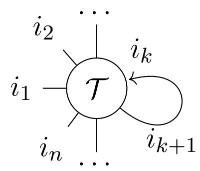


Figure 3.3: A tensor represented with the Penrose Notation with its indexes i_k and i_{k+1} connected as a loop

$$X = \begin{pmatrix} 1 & 1 \\ 0 & 1 \\ -1 & 0 \\ 0 & 2 \\ 2 & 0 \\ -2 & 2 \\ -1 & 0 \\ 3 & 3 \\ 1 & 0 \end{pmatrix} \qquad Y = \begin{pmatrix} 1 & 0 & -1 & 1 & 1 & 0 & 0 & 1 & 2 \\ 0 & 2 & -2 & 2 & 0 & 2 & 1 & 0 & 2 \end{pmatrix}$$

Therefore:

$$Z = XY = \begin{pmatrix} 1 & 2 & -3 & 3 & 1 & 2 & 1 & 1 & 4 \\ 0 & 2 & -2 & 2 & 0 & 2 & 1 & 0 & 2 \\ -1 & 0 & 1 & -1 & -1 & 0 & 0 & -1 & -2 \\ 0 & 4 & -4 & 4 & 0 & 4 & 2 & 0 & 4 \\ 2 & 0 & -2 & 2 & 2 & 0 & 0 & 2 & 4 \\ -2 & 4 & -2 & 2 & -2 & 4 & 2 & -2 & 0 \\ -1 & 0 & 1 & -1 & -1 & 0 & 0 & -1 & -2 \\ 3 & 6 & -9 & 9 & 3 & 6 & 3 & 3 & 12 \\ 1 & 0 & -1 & 1 & 1 & 0 & 0 & 1 & 2 \end{pmatrix}$$

And we can now reshape $Z \in \mathbb{R}^{9\times 9}$ as the tensor in $\mathbb{R}^{3\times 3\times 3\times 3}$ as

$$\mathcal{Z}(i_1, i_2, i_3, i_4) = Z(\overline{i_1, i_2}, \overline{i_3, i_4})$$

which in fact, is identical to $\mathcal{Z}$ in Example 3.5

There may be the case that when contracting a series of tensors, we might end up as what we see as a loop in the Penrose Notation (see fig. 3.3). Contracting over these two indexes we get the trace of the tensor $\mathcal{T}$ respect the indices i_k and i_{k+1} and we denote it as $\mathrm{Tr}(T)$ (See fig. 3.4):

Definition 3.8. *Given a tensor $T \in \mathbb{K}^{N_1\times\cdots\times N_n}$ with $N_k = N_p$ with $1 \leqslant k, p \leqslant n$ and $k \neq p$, we define the trace of $\mathcal{T}$ respect the indices k and p as the tensor $\mathrm{Tr}_k^p(\mathcal{T})$ of order $n-2$ with its entries defined as:*

$$\mathrm{Tr}_k^p(\mathcal{T})(i_1, \dots, i_{k-1}, i_{k+1}, \dots, i_{p-1}, i_{p+1}, \dots, i_n) = \sum_{j=1}^{N_k} \mathcal{T}(i_1, \dots, i_{k-1}, j, i_{k+1}, \dots, i_{p-1}, j, i_{p+1}, \dots, i_n)$$

Therefore, now one can ask that if we have an arbitrary graph G represented with the Penrose Notation and it is connected, we will be able to totally contract it to a single final tensor.

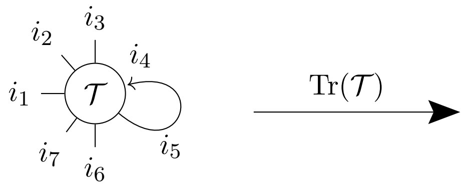


Figure 3.4: Representation of the trace of a tensor using the Penrose notation

3.3 Tensor Network States

The concept of tensor networks originated from a physics background. Roger Penrose described how its diagrammatic language could be used in various applications of physics [8].

Later, in 1992, Steven R. White developed de Density Matrix Renormalization Group (DRMG) algorithm for quantum lattice systems. It was considered the first successfull tensor network application [11].

The commonly use terminology "tensor decomposition" is equivalent to "tensor network" to some extent. After several years of progress accross different research fields, there is no significant distinction between tensor decomposition and tensor networks. Tensor decomposition was employed primarly in signal processing fields [10]. Traditional tensor decomposition models can be viewed as basic kinds of tensor networks. In this thesis we will study some of the properties of different tensor decomposition methods, and their effectivity.

We will define a mathematic definition of a tensor decomposition based on [13], we will see some common examples of tensor decompositions and we will define the tensor network structure space.

Informally, the way we construct a tensor network consists of picking a directed graph $G = (V, \bar{E})$, and for each vertex $i \in V$ we assign a vector space $\mathbb{V}_i$ and for each edge $(i, j) \in \bar{E}$ we assign a vector space $\mathbb{E}_i$ to the tail of the edge and its dual covector space $\mathbb{E}_i^*$ to the head of the edge. We will also demand that the graph G is connected.

More formally, let $\mathbb{V}_1, \ldots, \mathbb{V}_d$ be vector spaces with $\dim \mathbb{V}_i = N_i, i = 1, \ldots, d$. Let $\mathbb{E}_1, \ldots, \mathbb{E}_c$ be finite vector spaces with $\dim \mathbb{E}_i = R_i, i = 1, \ldots, c$. For each $i \in V$ we associate the tensor product space

$$\left(\bigotimes_{j\in \mathrm{IN}(i)} \mathbb{E}_j\right) \otimes \mathbb{V}_i \otimes \left(\bigotimes_{j\in \mathrm{OUT}(i)} \mathbb{E}_j^*\right)$$

and a contraction map κ_G defined by contracting factors in $\mathbb{E}_j$ with factors of $\mathbb{E}_j^*$

$$\kappa_G : \bigotimes_{i=1}^{d} \left[\left(\bigotimes_{j\in \mathrm{IN}(i)} \mathbb{E}_j\right) \otimes \mathbb{V}_i \otimes \left(\bigotimes_{j\in \mathrm{OUT}(i)} \mathbb{E}_j^*\right)\right] \to \bigotimes_{i=1}^{d} \mathbb{V}_i$$

Note that we have given this shapes to the tensors that we fix onto each vertex $i \in V$ because when we contract the whole graph, we will get a tensor of $\mathbb{V}_1 \otimes \cdots \otimes \mathbb{V}_n$. Since every directed edge (i, j) must point out of a vertex i and point into a vetex j, each copy of $\mathbb{E}_j$ is paired with one copy of $\mathbb{E}^*$, so the contraction κ_G is well defined and it results in a tensor of $\mathbb{V}_1 \otimes \cdots \otimes \mathbb{V}_n$ (See fig. 3.5)

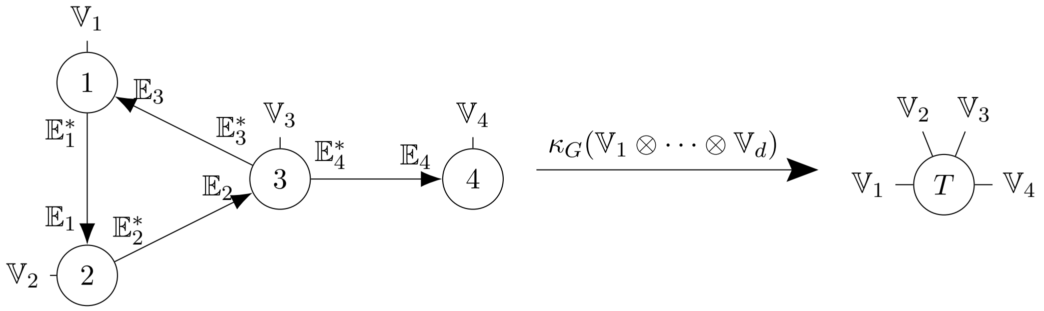


Figure 3.5: Example of the correspondance between some directed graph G and the vector spaces $\mathbb{V}_1, \ldots, \mathbb{V}_d, \mathbb{E}_1, \ldots, \mathbb{E}_c, \mathbb{E}_1^*, \ldots, \mathbb{E}_c^*$ using the Penrose notation

Definition 3.9 (Tensor network state)**.** *If a tensor $T \in \mathbb{V}_1 \otimes \cdots \otimes \mathbb{V}_n$ can be written as $T = \kappa_G(\mathcal{G}_1 \otimes \cdots \otimes \mathcal{G}_n)$ where*

$$\mathcal{G}_i \in \left(\bigotimes_{j \in \mathrm{IN}(i)} \mathbb{E}_j\right) \otimes \mathbb{V}_i \otimes \left(\bigotimes_{j \in \mathrm{OUT}(i)} \mathbb{E}_j^*\right)$$

*then we will say that T is a **tensor network state** associated to the graph G with cores $\mathcal{G}_i \in \mathbb{V}_i, i = 1, \ldots, n$*

Definition 3.10 (Tensor network)**.** *We will define all the resulting tensors that are possible by varying $\mathcal{G}_1, \ldots, \mathcal{G}_n$ and then contracting through κ_G as the tensor network associated to G and the vector spaces $\mathbb{V}_1, \ldots, \mathbb{V}_n, \mathbb{E}_1, \ldots, \mathbb{E}_c$ and we will write this set as* $\mathrm{TNS}(G; \mathbb{E}_1, \ldots, \mathbb{E}_c, \mathbb{V}_1, \ldots, \mathbb{V}_n)$*, i.e*

$$\begin{aligned}\mathrm{TNS}(G; \mathbb{E}_1, \ldots, \mathbb{E}_c, \mathbb{V}_1, \ldots, \mathbb{V}_n) := \Bigg\{ & \kappa_G(\mathcal{G}_1 \otimes \cdots \otimes \mathcal{G}_n) \in \mathbb{V}_1 \otimes \cdots \otimes \mathbb{V}_n : \\ & \mathcal{G}_i \in \left(\bigotimes_{j \in \mathrm{IN}(i)} \mathbb{E}_j\right) \otimes \mathbb{V}_i \otimes \left(\bigotimes_{j \in \mathrm{OUT}(i)} \mathbb{E}_j^*\right) \Bigg\}\end{aligned}$$

Now, since all vector spaces are determined up to isomorphism by its dimension, when the vector spaces $\mathbb{E}_1, \ldots, \mathbb{E}_c, \mathbb{V}_1, \ldots, \mathbb{V}_n$ are unimportant, we will write the tensor network as $\mathrm{TNS}(G; R_1, \ldots, R_c, N_1, \ldots, N_n)$. On the Penrose Notation representation of the network, we will not write the vector spaces since they are not important.

Also, since n is equal to the number of vertices of G, we will sometimes write $\mathrm{TNS}(G; R)$ for a more compact notation.

Definition 3.11. *Given a tensor network state* $\mathrm{TNS}(G, R)$*, from the graph given by its Penrose Notation, we will call the edges with a dangling end free edges, and the edges that connect two vertex contracted edges*

For example, in fig. 3.5, the edges labeled as $\mathbb{V}_1, \mathbb{V}_2, \mathbb{V}_3, \mathbb{V}_4$ are free edges and the rest are contracted edges.

The following theorem gives us that if we make $R_1, \ldots, R_c$ big enough, every tensor T can be a state of $\mathrm{TNS}(G; R_1, \ldots, R_c, N_1, \ldots, N_n)$. In fact, these values that guarantees that T is an state are $R_1 = \ldots R_c = \operatorname{rank} T$ where $\operatorname{rank} T$ is the *traditional rank* of a tensor. We will give now the definition of the traditional rank:

Definition 3.12. *We say that a tensor has* ***traditional rank*** r *and we write* $\operatorname{rank} T = r$ *with* $r \in \mathbb{N}$ *being the minimum value such that we can write* T *as*

$$T = \sum_{p=1}^{r} v_p^{(1)} \otimes \cdots \otimes v_p^{(n)}$$

where $v_1^{(i)}, \ldots, v_r^{(i)} \in \mathbb{V}_i, i = 1, \ldots, n$

We can immediatly see that $\operatorname{rank} T \leqslant \prod_{i=1}^n N_i$. Unlike matrices, determining the rank of a tensor is an NP-hard problem. [3]. Even finding the maximum traditional rank of a tensor space (that means finding $\max_{T \in \mathbb{V}_1 \otimes \cdots \otimes \mathbb{V}_n} \operatorname{rank} T$) it still remains as an open problem. We can find an slightly better upper bound for the tensor rank:

Proposition 3.13.

$$\operatorname{rank} T \leqslant \left\lfloor \frac{\prod_{i=1}^n N_i}{\sum_{i=1}^n N_i} \right\rfloor \tag{3.3.1}$$

Proof. Let $r = \operatorname{rank} T$. We can write $T = \sum_{p=1}^r v_p^{(1)} \otimes \cdots \otimes v_p^{(n)}$. Now, each term of this sum has $\sum_{i=1}^n N_i$ adjustable parameters, since each $v_p^{(i)}$ is a vector of $\mathbb{V}_i$ with its dimension being N_i. So, in total we will have $r \sum_{i=1}^n N_i$ adjustable parameters in our decomposition. Since our tensor T is completly determined by $\prod_{i=1}^n N_i$ parameters, we can impose that $r \sum_{i=1}^n N_i \leqslant \prod_{i=1}^n N_i$ $\square$

Theorem 3.14. [13] Let $T \in \mathbb{V}_1 \otimes \cdots \otimes \mathbb{V}_n$ and let G be a connected graph with n vertices and c edges. There exists $R_1, \ldots, R_c \in \mathbb{N}$ such that

$$T \in \mathrm{TNS}(G; R_1, \ldots, R_c, N_1, \ldots, N_d)$$

in fact, we can choose $R_1 = \cdots = R_c = \operatorname{rank} T$

Proof. Let $r = \operatorname{rank} T$. Then there exist $v_1^{(i)}, \ldots, v_r^{(i)} \in \mathbb{V}_i, i = 1, \ldots, n$ such that

$$T = \sum_{p=1}^{r} v_1^{(p)} \otimes \cdots \otimes v_n^{(p)}$$

We take $R_1 = \cdots = R_c = r$ we take for each $i = 1, \ldots, n$

$$\mathcal{G}_i = \sum_{p=1}^{r} \left(\bigotimes_{j \in \mathrm{IN}(i)} e_p^{(j)} \right) \otimes v_p^{(i)} \otimes \left(\bigotimes_{j \in \mathrm{OUT}(i)} e_p^{(j)*} \right)$$

Now observe that for each $i = 1, \ldots, n$ there exists an unique h such that whenever $j \in \mathrm{IN}(i) \cap \mathrm{OUT}(i)$, $e_p^{(j)}$ and $e_p^{(j)*}$ contract and give δ_{pq}, therefore the summand vanishes except when $p = q$. This together with the assumption that G is a connected graph implies that $\kappa_G(\mathcal{G}_1 \otimes \cdots \otimes \mathcal{G}_n)$ reduces to a sum of terms of the form $v_p^{(1)} \otimes \cdots \otimes v_p^{(d)}$ for $p = 1, \ldots, r$, which is of course T $\square$

Definition 3.15 (Tensor G-rank)**.** *Given a graph G, we define the tensor rank respect to a G as the mapping*

$$\begin{aligned}\operatorname{rank}_G : \mathbb{V}_1 \otimes \cdots \otimes \mathbb{V}_n &\longrightarrow \mathcal{P}(\mathbb{N}^c) \\ T &\longmapsto \min\{(R_1, \ldots, R_c) \in \mathbb{N}^c : T \in \operatorname{TNS}(G; R_1, \ldots, R_c, N_1, \ldots, N_d)\}\end{aligned}$$

Where $\min(S)$ with $S \subset \mathbb{N}^c$ denotes the minimal elements of S. We treat $\mathbb{N}^c$ with its usual partial order:

$$(a_1, \ldots, a_c) \leqslant (b_1, \ldots, b_c) \iff a_1 \leqslant b_1, a_2 \leqslant b_2, \ldots, a_c \leqslant b_c$$

So for example if $S = \{(3,4,5), (2,1,3), (1,3,2)\}$, then $\min(S) = \{(2,1,3), (1,3,2)\}$ Note that by Theorem 3.14 $\operatorname{rank}_G(T)$ is always a finite set.

We will say that if $(r_1, \ldots, r_c) \in \operatorname{rank}_G(T)$, we will say that $(r_1, \ldots, r_c)$ is a G-rank of T. We will later see that in some tensor network structures, for example in tree tensor networks, the G-rank of a tensor T is unique

3.4 Common Tensor network structures

Example 3.16. [Tensor Train decomposition] [7] Let $\mathcal{T} \in \mathbb{R}^{N_1 \times \cdots \times N_n}$. A tensor train decomposition or matrix product state of $\mathcal{T}$ are a set of 3th-order tensors $\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_n$ with $\mathcal{G}_i \in \mathbb{R}^{R_{i-1} \times N_i \times R_i}$ and $R_0 = R_n = 1$ such that every element of $\mathcal{T}$ is written in the form

$$\mathcal{T}(i_1, i_2, \ldots, i_n) = \sum_{r_0, \ldots, r_n}^{R_0, \ldots, R_n} \mathcal{G}_1(r_0, i_1, r_1)\mathcal{G}_2(r_1, i_2, r_2) \cdots \mathcal{G}_n(r_{n-1}, i_n, r_n) \tag{3.4.1}$$

We denote $R_0, R_1, \ldots, R_n$ as the ranks of the tensor train decomposition, or TT-ranks.

We can easily see that the tensor train decomposition (or TT) is obtained by our definition of a tensor network when G is a path, also the contraction of the whole network yields eq. (3.4.1)

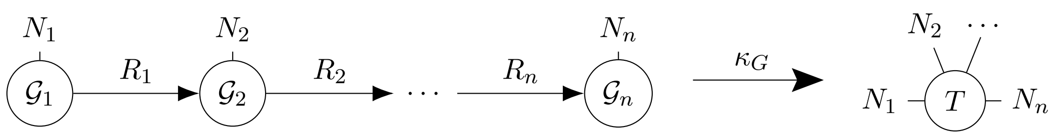


Figure 3.6: Tensor Train decomposition

Example 3.17. [Tensor Ring decomposition] [15] Tensor ring decomposition (or TR) or also known a matrix product state with periodic boundary conditions, is obtained when G is a cycle.

Tensor Ring decomposition is considered generalization of Tensor Train decomposition, it's contraction is the same as eq. (3.4.1) but removing the condition $R_0 = R_1 = 1$.

Theorem 3.18 (Circular dimensional permutation invariance)**.** *Let $\mathcal{T} \in \mathbb{R}^{N_1 \times \cdots \times N_n}$ be a nth-order tensor with its corresponding tensor ring decomposition $\mathcal{T} = \mathcal{R}(\mathcal{U}^{(1)}, \mathcal{U}^{(2)}, \ldots, \mathcal{U}^{(n)})$*

Example 3.19. [Fully connected tensor network] The fully connected tensor network decomposition is obtenied when G is a complete graph.

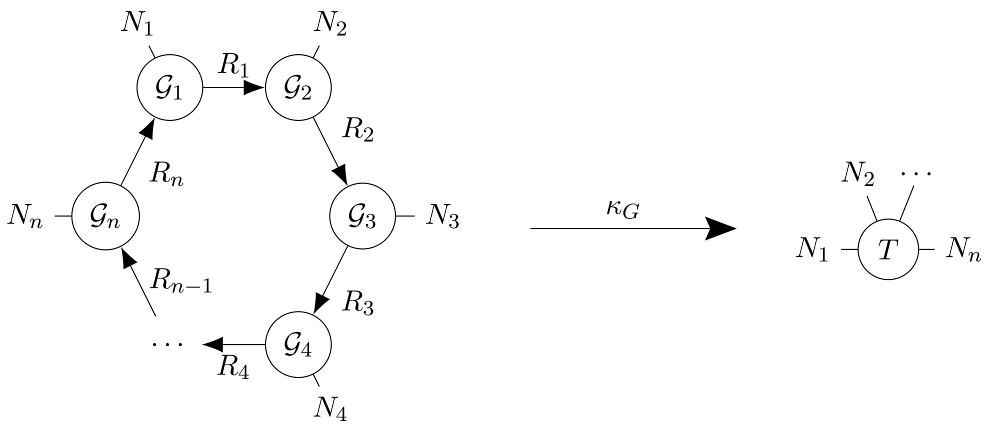


Figure 3.7: Tensor Ring (TR) decomposition

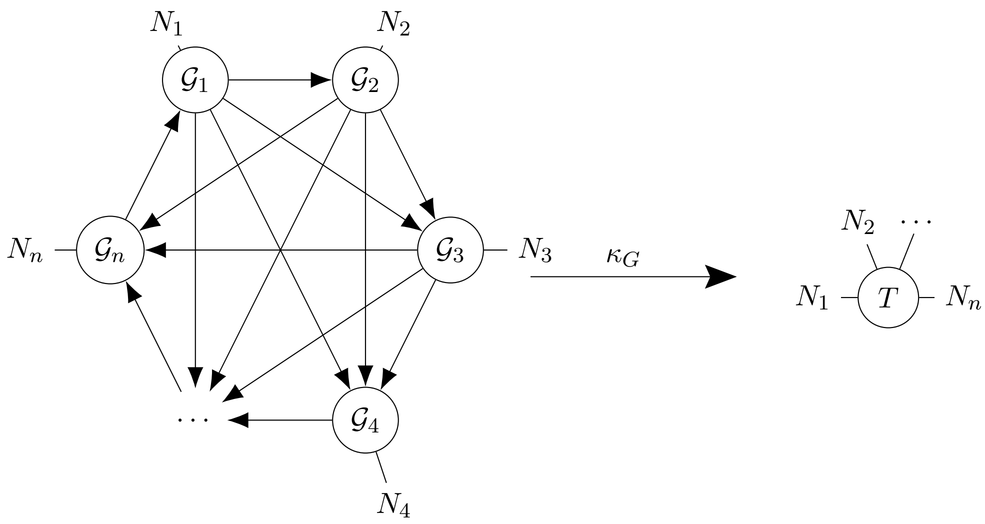


Figure 3.8: Fully connected tensor network decomposition (FCTN)

This theorem serves as an upper bound for our problem. But either way it is still very big. We can take a different approach: we can fix $G, R_1, \ldots, R_c$ and then find the cores of the tensor network that approximate better to our objective tensor T. That means finding

$$\underset{\mathcal{G}_1,\ldots,\mathcal{G}_n}{\arg\min} \|T - \kappa_G(\mathcal{G}_1, \ldots, \mathcal{G}_n)\|_F \tag{3.4.2}$$

Later, we will discuss some algorithms that gives us a solution to eq. (3.4.2).

3.5 Tree Tensor Networks

When G is a tree, i.e when there are no cycles inside G, we say that $\mathrm{TNS}(G; R)$ is a tree tensor network. This includes all tensor train networks. There are a lot of interesting results in tree tensor networks. One very useful property is that given a tensor T, his G rank when G is a tree, is always unique.

3.6 The Alternating Least Squares algorithm

[5] Suppose we want to only optimize the function eq. (3.4.2) for only one variable core $\mathcal{G}_m$ leaving the rest fixed. Then our problem would become

$$\arg\min_{\mathcal{G}_m} \|T - \kappa_G(\mathcal{G}_1, \ldots, \mathcal{G}_n)\|_F$$

Now, we could apply our contraction mapping κ_G for all cores excluding $\mathcal{G}_m$ (figs. 3.9 and 3.10). We will call this tensor $\mathcal{G}^{\neq m}$. Now, if we consider appropiate matricizations $T^{(m)}$, $G^{\neq m}$ and G_m of T, $\mathcal{G}^{\neq m}$ and $\mathcal{G}_i$ respectively, evaluating the whole tensor network is equal to computing the product $G^{\neq m}G_m$, so our problem is equivalent to solve the following linear least squares problem:

$$\arg\min_{G_m} \|G^{\neq m}G_m - T^{(m)}\|_2 \tag{3.6.1}$$

Let $x^{(i)}$ be the i-th column of G_m and $y^{(i)}$ the i-th column of $T^{(m)}$. Solving 3.6.1 means solving for each i

$$\arg\min_{x^{(i)}} \|G^{\neq m}x^{(i)} - y^{(i)}\|_2 \tag{3.6.2}$$

Since we can't assure that there exists an exact solution to $G^{\neq m}x^{(i)} = y^{(i)}$, we can use the solution to the normal equation $(G^{\neq m})^T G^{\neq m}x^{(i)} = (G^{\neq m})^T y^{(i)}$.

Now we can iteratively change the varying core tensor $\mathcal{G}_i$ until the contraction of the whole tensor network is T with some fixed error ϵ:

Algorithm 1 Tensor Network ALS

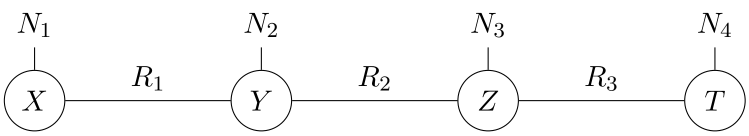
Input: A tensor $T \in \mathbb{K}^{N_1 \times \cdots \times N_n}$ and some fixed error ϵ
Output: Core tensors $\mathcal{G}_1, \ldots, \mathcal{G}_n$

```
Initialize tensors G_1, ..., G_n
while ||T − κ_G(G_1, ..., G_n)||_F > ε do
    for k = 1, ..., n do
        G_m ← argmin_{G_m} ||G^{≠m} G_m − T^{(m)}||_2
return G_1, ..., G_n
```

3.7 Contracting the tensor network

The ALS algorithm requires us to contract the network when we update the cores and also when we do a contraction for checking for the stopping condition. The order in which we select to contract the cores affects directly the computational cost of the contraction.

For example, consider the following tensor train network:



The contraction of the whole network is given by

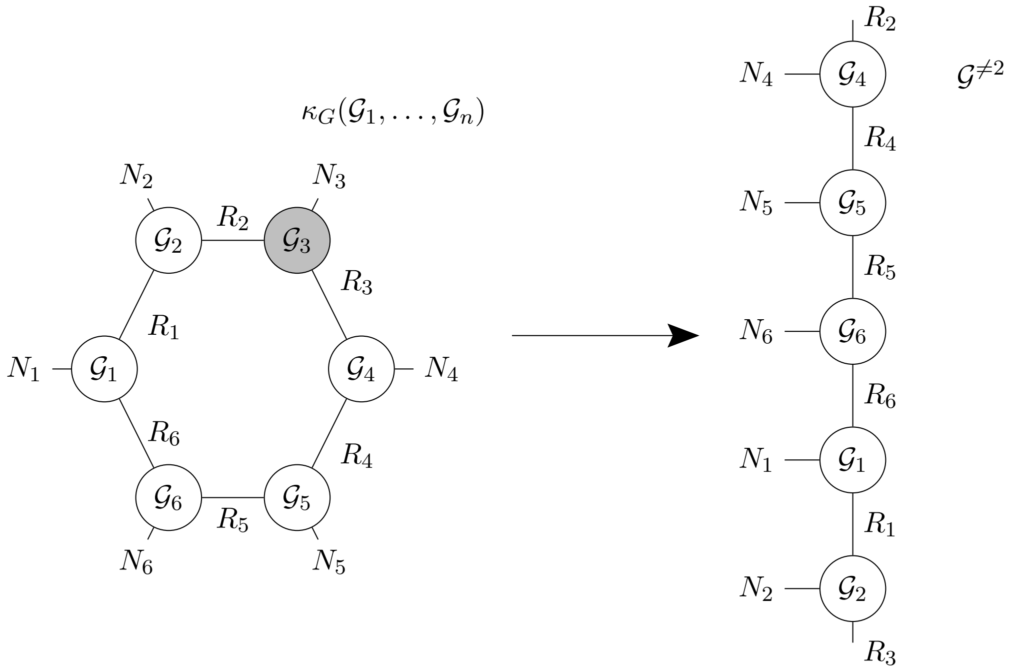


Figure 3.9: The representation of $\mathcal{G}^{\neq m}$ on the TR decomposition, with $m = 2$

$$T_{ijkl} = \sum_{p,q,r}^{R_1,R_2,R_3} X_{ip} Y_{pjq} Z_{qkr} T_{rl}$$

We could contract first R_1, then R_2 and finally R_3. That means computing:

$$A_{ijq} = \sum_{p}^{R_1} X_{ip} Y_{pjq} \qquad B_{ijkr} = \sum_{q}^{R_2} A_{ijq} Z_{qkr} \qquad T_{ijkl} = \sum_{r}^{R_3} B_{ijkr} T_{rl}$$

For this computation, we would need $R_1R_2N_1N_2 + R_2R_3N_1N_2N_3 + R_3N_1N_2N_3N_4$ products, and we would need to store first A, then B and then T

If we consider a different ordering for the contraction, for example first R_1, then R_3 and finally R_2 we get a different computation cost. We would need to compute the matrices

$$A'_{ijq} = \sum_{p}^{R_1} X_{ip} Y_{pjq} \qquad B'_{qkl} = \sum_{r}^{R_3} Z_{qkr} T_{rl} \qquad T_{ijkl} = \sum_{q}^{R_2} A'_{ijq} B'_{qkl}$$

The result of the contraction is the same, but now instead the number of computations is changed to $R_1R_2N_1N_2 + R_3N_1N_3N_4 + R_2N_1N_2N_3N_4$.

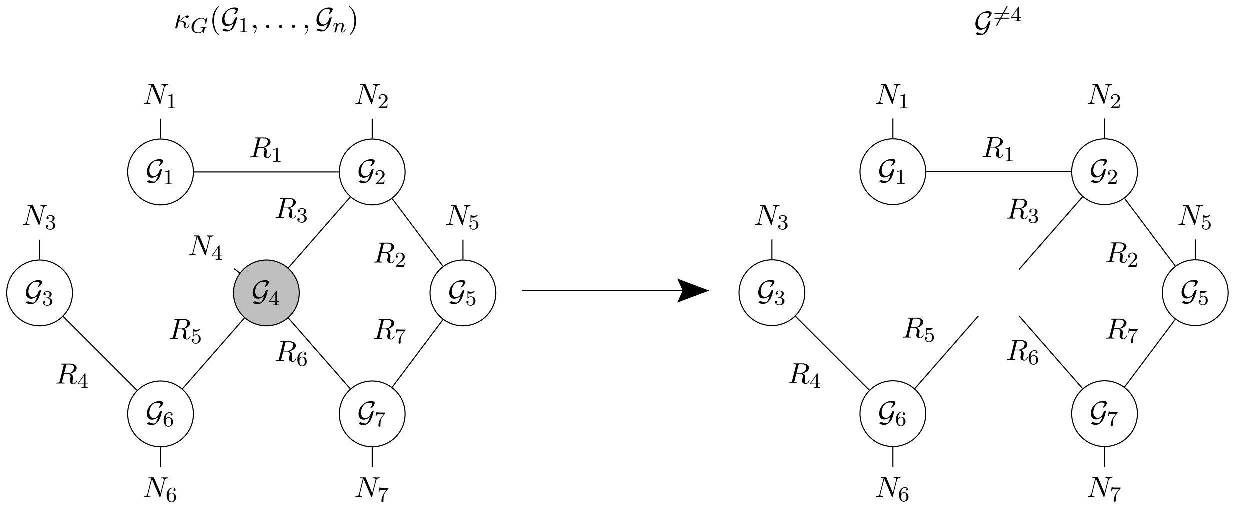


Figure 3.10: The representation of $\mathcal{G}^{\neq m}$ on the TR decomposition, with $m = 2$

Chapter 4

Topological search on the tensor network structure space

By now, we know that for any tensor T, if we choose a graph $G = (V, E)$ we can choose $R_1, \dots, R_c \leqslant \operatorname{rank} T \leqslant \left\lfloor \frac{\prod_{i=1}^{n} N_i}{\sum_{i=1}^{n} N_i} \right\rfloor$ such that $T \in \mathrm{TNS}(G, R)$. We can also find thanks to the ALS algorithm the cores $\mathcal{G}_1, \dots, \mathcal{G}_n$ of $\mathrm{TNS}(G, R)$. The only thing that remains is how we pick an optimal G for compressing T with a fixed error ϵ, since we know that the optimal graph for a tensor network state of a given tensor T depends on the underlying data of T itself.

We will aim to search the tensor network that gives the best compression ratio for T, for that, we will define the size of a tensor network:

Definition 4.1. *[2] Given a tensor network state* $\mathrm{TNS}(G, R)$*, we define*

$$\mathrm{Size}(G, R) = \sum_{i=1}^{c} \mathrm{Size}\, \mathcal{G}_i$$

We would want to solve

$$\arg\min_{G,R} \ \mathrm{Size}(G, R) \quad s.t \quad \|\kappa_G(\mathcal{G}_1, \dots, \mathcal{G}_c) - T\|_F \leqslant \epsilon \|T\|_F$$

Let $\mathcal{X} \in \mathbb{R}^{N_1 \times \cdots \times N_n}$ an n-order tensor. Let the following problem:

$$\min_{r \in \mathbb{K}_{K_N}} \phi(K_N, r) \quad s.t\, \mathcal{X} \in \mathrm{TNS}\,(K_N, r)$$

TODO: Suposo que començar per dir quines parts del graf caldria tallar maybe??? Fer més representacions gràfiques de segons quina demostració com més clar quedi tot millor

- Descriure G-ranks
- Algorismes per aproximar TNS per G-ranks propers i mínims si es pot fer
- Algun algorisme per trobar heuristicament els G-ranks adequats? (suposo q depen de compressió ratio i l'error relatiu)

- Com podem trobar un G adequat?
- Estratègies per contraure tensors més ràpidament? (DRMG?)
- Algorismes, part pràctica en C/C++
- Fer moltes gràfiques
- Fer aplicacions per machine learning, etc.
- Fixar la mathematical subject classification

Chapter 5

Conclusions

TODO

Fent servir un símil geomètrico-cartogràfic, aquesta memòria constitueix un mapa a escala planetària de la demostració de la conjectura feble de Goldbach presentada per Helfgott i un mapa a escala continental de la verificació numèrica d'aquesta. Estudis posteriors i més profunds haurien de permetre elaborar mapes de menor escala.

La naturalesa dels nombres primers ens ha portat per molts racons diferents de les Matemàtiques; en no imposar-nos restriccions en la forma de pensar, hem pogut gaudir del viatge i assolir els objectius que ens vam plantejar a l'inici del projecte i anar més enllà, sobretot en el camp de la computació i la manipulació de grans volums de dades numèriques.

Una gran part dels coneixements bàsics que hem hagut de fer servir han estat treballats en les assignatures de Mètodes analítics en teoria de nombres i d'Anàlisi harmònica i teoria del senyal, que són optatives de quart curs del Grau de Matemàtiques. Altres els hem hagut d'aprendre durant el desenvolupant del projecte. S'ha realitzat una tasca de recerca bibliogràfica important, consultant recursos antics i moderns, tant en format digital com en format paper.

Bibliography

[1] Christopher Terence John Dodson and Timothy Poston. "Tensors and Multilinear Forms". In: *Tensor Geometry: The Geometric Viewpoint and Its Uses*. Ed. by Christopher Terence John Dodson and Timothy Poston. Berlin, Heidelberg: Springer, 1991, pp. 98–113. ISBN: 978-3-642-10514-2. DOI: 10.1007/978-3-642-10514-2_6. URL: https://doi.org/10.1007/978-3-642-10514-2_6 (visited on 03/28/2025).

[2] Zheng Guo et al. *Tensor Network Structure Search Using Program Synthesis*. Feb. 22, 2025. DOI: 10.48550/arXiv.2502.02711. arXiv: 2502.02711 [cs]. URL: http://arxiv.org/abs/2502.02711 (visited on 05/12/2025). Pre-published.

[3] Christopher Hillar and Lek-Heng Lim. *Most Tensor Problems Are NP-hard*. July 1, 2013. DOI: 10.48550/arXiv.0911.1393. arXiv: 0911.1393 [cs]. URL: http://arxiv.org/abs/0911.1393 (visited on 03/31/2025). Pre-published.

[4] Chao Li et al. *Permutation Search of Tensor Network Structures via Local Sampling*. June 14, 2022. DOI: 10.48550/arXiv.2206.06597. arXiv: 2206.06597 [cs]. URL: http://arxiv.org/abs/2206.06597 (visited on 03/24/2025). Pre-published.

[5] Osman Asif Malik, Vivek Bharadwaj, and Riley Murray. *Sampling-Based Decomposition Algorithms for Arbitrary Tensor Networks*. Oct. 7, 2022. DOI: 10.48550/arXiv.2210.03828. arXiv: 2210.03828 [math]. URL: http://arxiv.org/abs/2210.03828 (visited on 04/30/2025). Pre-published.

[6] Román Orús. "Tensor Networks for Complex Quantum Systems". In: *Nature Reviews Physics* 1.9 (Sept. 2019), pp. 538–550. ISSN: 2522-5820. DOI: 10.1038/s42254-019-0086-7. URL: https://www.nature.com/articles/s42254-019-0086-7 (visited on 05/14/2025).

[7] I. V. Oseledets. "Tensor-Train Decomposition". In: *SIAM Journal on Scientific Computing* 33.5 (Jan. 2011), pp. 2295–2317. ISSN: 1064-8275. DOI: 10.1137/090752286. URL: https://epubs.siam.org/doi/10.1137/090752286 (visited on 03/24/2025).

[8] Roger Penrose. "Applications of Negative Dimensional Tensors". In: *Combinatorial Mathematics and its Applications* (1971), pp. 221–244. URL: https://homepages.math.uic.edu/~kauffman/Penrose.pdf (visited on 03/04/2025).

[9] Steven Roman. "Tensor Products". In: *Advanced Linear Algebra*. Ed. by Steven Roman. New York, NY: Springer, 2008, pp. 355–409. ISBN: 978-0-387-72831-5. DOI: 10.1007/978-0-387-72831-5_14. URL: https://doi.org/10.1007/978-0-387-72831-5_14 (visited on 03/25/2025).

[10] Maolin Wang et al. *Tensor Networks Meet Neural Networks: A Survey and Future Perspectives*. May 8, 2023. DOI: 10.48550/arXiv.2302.09019. arXiv: 2302.09019 [cs]. URL: http://arxiv.org/abs/2302.09019 (visited on 03/09/2025). Pre-published.

[11] Steven R. White. "Density Matrix Formulation for Quantum Renormalization Groups". In: *Physical Review Letters* 69.19 (Nov. 9, 1992), pp. 2863–2866. DOI: 10.1103/PhysRevLett.69.2863. URL: https://link.aps.org/doi/10.1103/PhysRevLett.69.2863 (visited on 03/27/2025).

[12] Robin J. Wilson. *Introduction to Graph Theory*. 4. ed., [Nachdr.] Harlow Munich: Prentice Hall, 2009. 171 pp. ISBN: 978-0-582-24993-6.

[13] Ke Ye and Lek-Heng Lim. *Tensor Network Ranks*. Feb. 9, 2019. DOI: 10.48550/arXiv.1801.02662. arXiv: 1801.02662 [math]. URL: http://arxiv.org/abs/1801.02662 (visited on 03/24/2025). Pre-published.

[14] Tatsuya Yokota. *Very Basics of Tensors with Graphical Notations: Unfolding, Calculations, and Decompositions*. Nov. 25, 2024. DOI: 10.48550/arXiv.2411.16094. arXiv: 2411.16094 [cs]. URL: http://arxiv.org/abs/2411.16094 (visited on 02/24/2025). Pre-published.

[15] Qibin Zhao et al. *Tensor Ring Decomposition*. June 17, 2016. DOI: 10.48550/arXiv.1606.05535. arXiv: 1606.05535 [cs]. URL: http://arxiv.org/abs/1606.05535 (visited on 03/04/2025). Pre-published.

[16] Yu-Bang Zheng et al. "Fully-Connected Tensor Network Decomposition and Its Application to Higher-Order Tensor Completion". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 35.12 (12 May 18, 2021), pp. 11071–11078. ISSN: 2374-3468. DOI: 10.1609/aaai.v35i12.17321. URL: https://ojs.aaai.org/index.php/AAAI/article/view/17321 (visited on 03/04/2025).

Appendix A

Chapter 1