Skoltech

MASTER'S THESIS

Expressive Power of Tensor-Train Networks With Equal TT-cores

Master's Educational Program: Data Science

Student: _____ Emil Alkin
signature

Research Advisor: _____ Ivan Oseledets
signature
Dr. Sc., Professor

Moscow 2024

Skoltech

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

Выразительная сила Tensor-Train сетей с равными TT-ядрами

Магистерская образовательная программа: Науки о данных

Студент: _____ Эмиль Алкин
подпись

Научный руководитель: _____ Иван Оселедец
подпись
д.ф.-м.н., профессор

Москва 2024

Expressive Power of Tensor-Train Networks With Equal TT-cores

Emil Alkin

Submitted to the Skolkovo Institute of Science and Technology on June 18, 2024

ABSTRACT

Deep neural networks show their extreme efficiency in solving a wide range of practical problems. Despite this, a theoretical explanation of this phenomenon is only beginning to emerge in scientific research. For some special kinds of deep neural networks, it has been shown that depth is the key to efficiency. In particular, it has recently been shown that Tensor Train networks, i.e. recurrent neural networks, each layer of which implements a bilinear function, are exponentially more expressive than shallow networks. However, in practice, recurrent neural networks with identical layers are used, the analogue of which are Tensor Train networks with equal TT-cores. For this class of networks, the analogous result is not proved, but formulated as a Hypothesis. We prove this Hypothesis and thus close the question of exponential expressivity of traditional recurrent neural networks. We also conduct a series of numerical experiments to confirm the theoretical result.

Keywords: Deep Learning, Expressive Power, Recurrent Neural Networks, Tensor Decompositions

Research advisor:
Name: Ivan Oseledets
Degree, title: Dr. Sc., Professor

Выразительная сила Tensor-Train сетей с равными TT-ядрами

Эмиль Алкин

Представлено в Сколковский институт науки и технологий
Июнь 18

Аннотация

Глубокие нейронные сети показывают свою чрезвычайную эффективность при решении широкого спектра практических задач. Несмотря на это, теоретическое объяснение этого явления только начинает просматриваться в научных исследованиях. Для некоторых специальных видов глубоких нейронных сетей показано, что глубина это ключ к эффективности. В частности, недавно было показано, что Tensor-Train сети, то есть рекуррентные нейронные сети, каждый слой, которых реализует билинейную функцию, экспоненциально более выразительны чем однослойные сети. Однако, на практике используются рекуррентные нейронные сети с одинаковыми слоями, аналогом которых являются Tensor-Train сети с равными TT-ядрами. Для такого класса сетей аналогичный результат не доказан, а сформулирован в качестве гипотезы. Мы доказываем эту Гипотезу и тем самым закрываем вопрос об экспоненциальной выразительности классических рекуррентных нейронных сетей. Также мы проводим серию численных экспериментов, подтверждающих теоретический результат.

Ключевые слова: глубокое обучение, выразительная сила, рекуррентные нейронные сети, тензорные разложения

Научный руководитель:
Имя: Иван Оселедец
Степень, звание: д.ф.-м.н., профессор

Contents

Chapter 1
Introduction

Deep neural networks solve many practical tasks both in computer vision via Convolutional Neural Networks (CNNs) and in audio and text processing via Recurrent Neural Networks (RNNs). Although many practical problems are solved using deep neural networks (DNNs), the theoretical justification of their effectiveness has not been fully studied.

One approach for justifying the power of depth is to show that deep networks can efficiently express functions that would require shallow networks to have super-polynomial size. Early results related to this approach [4], [5], [3], [8] consider specific network architectures that are not commonly used in practice.

In 2016, Nadav Cohen, Or Sharir, and Amnon Shashua published a paper [1] in which they proposed a deep network architecture based on arithmetic circuits (also known as Sum-Product networks) that inherently uses three features of convolutional network architecture: locality, sharing and pooling. They also showed a connection between the proposed architecture and Hierarchical Tucker decomposition of the parameter tensor $\mathcal{A}^y$ in the following hypotheses space:

$$h_y(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_d) = \sum_{i_1,\ldots,i_d=1}^{m} \mathcal{W}_y^{i_1\ldots i_d} \cdot f_{\theta_{i_1}}(\mathbf{x}_1) \cdot \ldots \cdot f_{\theta_{i_d}}(\mathbf{x}_d),$$

where h_y is a score function for class label y; f_θ is a representation function, selected from a parametric family $\mathcal{F} = \{f_\theta : \mathbb{R}^n \to \mathbb{R}^m\}_{\theta\in\Theta}$; $\mathcal{W}_y^{i_1\ldots i_d}$ is a parameter tensor (see formal definition in Chapter 4). Using this connection the authors proved that deep CNNs are exponentially more expressive than shallow networks (see [1], [2] for details).

In 2018, Valentin Khrulkov, Alexander Novikov and Ivan Oseledets showed in the paper [7] the connection between Tensor-Train tensor decomposition and deep recurrent-type neural networks, and used this connection to prove that deep recurrent-type neural networks in which all layers have their own parameters are exponentially more expressive than shallow neural networks. The authors also hypothesized that their results are also true for traditional RNNs, in which layers share parameters. The research towards this hypothesis is particularly interesting because it refers to the architectures used to solve practical problems.

Relevance. Recurrent neural networks are widely used in audio and text processing. Since the theoretical result that deep recurrent-type neural networks in which all layers have their own parameters are exponentially more expressive than shallow neural networks is not applicable to traditional RNNs, the problem of theoretical estimation of the expressive power of traditional RNNs remains open and the solution of which is of immediate interest.

Main purpose of the research. The main goal of my research is to obtain new theoretical results in the study of the expressive power of certain neural network architectures. I define the main direction of research as the study towards the hypothesis formulated in [7], in particular, the study of the expressive power of traditional recurrent neural networks, as well as conducting experiments with the considered neural network architectures on real-world datasets such as MNIST and CIFAR-10.

Scientific novelty. I formulate and prove the expressive power theorem for the Tensor-Train decomposition with equal TT-cores (see Theorem 1 in Chapter 4). I confirm the obtained theoretical result with original numerical experiments using my own techniques (see Sections 6.1, 6.2).

Statements for defense. The proved theorem can be interpreted as follows:

- to emulate a traditional recurrent neural network, a shallow architecture of exponentially larger width is required.

Chapter 2
Author contribution

All the results presented in the thesis were obtained by me. In particular, my personal contribution is the last part of the proof of Theorem 1, in which I prove the existence of a tensor $\mathcal{X}$ such that the rank of its matricization is at least $q^{\frac{d}{2}}$, the formulation and proof of the Lemma 3, as well as modelling and performing all numerical experiments comparing Tensor-Train networks with different and equal inner TT-cores.

Chapter 3
Literature review

As soon as empirical evidence of the effectiveness of deep neural networks in solving practical tasks was obtained, the question of a theoretical justification for the *power of depth* arose. This question is connected with fundamental notion of *neural network expressivity*.

Early results on this question related to either boolean circuits, or specific neural network architectures that are not commonly used in practice:

- Johan Hastad and Mikael Goldmann in [5] obtained some results on power of depth for boolean circuits computing functions are given;
- Olivier Delalleau and Yoshua Bengio in [3] demonstrated a difference in expressive efficiency between sum-product networks (computation networks, whose individual units compute either products or weighted sums) of depth 3, and those of higher depths;
- James Martens and Venkatesh Medabalimi in [8] generalized results of the paper [3] for approximate computations.

In 2016, Nadav Cohen, Or Sharir, and Amnon Shashua in [1] derived a deep network architecture based on arithmetic circuits (also known as sum-product networks), showed the connection between Hierarchical Tucker tensor decomposition and CNNs, and used this connection to prove that deep CNNs are exponentially more expressive than shallow networks. This paper discovered a new approach for studying expressive power through evaluating the rank of the tensor decomposition corresponding to a specific neural network architecture.

In 2017, Maithra Raghu, Ben Poole, Jon Kleinberg, Surya Ganguli, Jascha Sohl Dickstein in [10] proposed a new approach for studying expressive power using a novel notion of *trajectory length*.

In 2018, Valentin Khrulkov, Alexander Novikov and Ivan Oseledets showed in [7] the connection between Tensor-Train decomposition and deep recurrent-type neural networks, and used this connection to prove that deep recurrent-type neural networks in which all layers have their own parameters are exponentially more expressive than shallow neural networks. The authors also hypothesized that their results are also true for traditional RNNs, in which layers share parameters. The research towards this hypothesis is particularly interesting because it refers to the architectures used to solve practical problems.

Chapter 4
Problem statement

We consider the task of classification of a collection of vectors $(\mathbf{x}_1, \ldots, \mathbf{x}_d), \mathbf{x}_i \in \mathbb{R}^n$, into one of the categories $\mathcal{Y} := \{1, \ldots, Y\}$. Representing instances as a collection of vectors is natural in many applications. We assume that the components $\mathbf{x}_1, \ldots, \mathbf{x}_d$ of an instance X can be transformed by a function f_θ from the parametric family $\mathcal{F} = \{f_\theta : \mathbb{R}^n \to \mathbb{R}^m\}_{\theta \in \Theta}$ into vectors $f_\theta(\mathbf{x}_1), \ldots, f_\theta(\mathbf{x}_d)$, which we call *lower-dimensional representations* of $\{\mathbf{x}_k\}_{k=1}^d$. As usual, an instance X will be assigned to class y if and only if the value $h_y(X)$ of *score function* h_y for class label y is maximal among the values $h_1(X), \ldots, h_Y(X)$ of score functions $h_1, \ldots, h_Y$ for all class labels. We define our hypotheses space by the following formula for the score function h_y for a class label y:

$$h_y(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_d) = \sum_{i_1, \ldots, i_d = 1}^{m} \mathcal{W}_y^{i_1 \ldots i_d} \cdot f_{\theta_{i_1}}(\mathbf{x}_1) \cdot \ldots \cdot f_{\theta_{i_d}}(\mathbf{x}_d), \tag{4.1}$$

where $\mathcal{W}_y$ is a *coefficient tensor* of order d and dimension m in each mode (see the motivation for this definition of hypotheses space in [1]).

Denote $[n] := \{1, \ldots, n\}$ for any positive integer n. Let us recall some known tensor decompositions.

Canonical decomposition, or *CP-decomposition* for short, of a tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times \ldots \times n_d}$ is defined as follows

$$\mathcal{X}^{i_1 i_2 \ldots i_d} = \sum_{\alpha=1}^{r} \mathbf{v}_{1,\alpha}^{i_1} \cdot \mathbf{v}_{2,\alpha}^{i_2} \cdot \ldots \cdot \mathbf{v}_{d,\alpha}^{i_d}, \qquad \mathbf{v}_{i,\alpha} \in \mathbb{R}^{n_i}. \tag{4.2}$$

Canonical rank, or *CP-rank* for short, of a tensor $\mathcal{X}$ is the minimal r such that canonical decomposition of $\mathcal{X}$ with r summands exists.

Tensor-Train decomposition, or *TT-decomposition* for short, of a tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times \ldots \times n_d}$ is defined as follows

$$\mathcal{X}^{i_1 i_2 \ldots i_d} = \sum_{\alpha_1=1}^{r_1} \sum_{\alpha_2=1}^{r_2} \ldots \sum_{\alpha_{d-1}=1}^{r_{d-1}} G_1^{i_1 \alpha_1} \cdot G_2^{\alpha_1 i_2 \alpha_2} \cdot \ldots \cdot G_d^{\alpha_{d-1} i_d}, \tag{4.3}$$

where $G_k \in \mathbb{R}^{r_{k-1} \times n_k \times r_k},\ k \in [d]$ $(r_0 := 1, r_d := 0)$, are tensors which we call *TT-cores*.

Tensor-Train ranks, or *TT-ranks* for short, of a tensor $\mathcal{X}$ is the element-wise minimal ranks $\mathbf{r} = (r_1, r_2, \ldots, r_{d-1})$ such that Tensor-Train decomposition of $\mathcal{X}$ with such $r_1, r_2, \ldots, r_{d-1}$ exists.

Let us denote $\mathbf{n} := (n_1, n_2 \ldots n_d)$. Set of all tensors $\mathcal{X}$ with mode sizes $\mathbf{n}$ representable in TT-format with

$$\text{rank}_{TT}\, \mathcal{X} \leqslant \mathbf{r},$$

for some vector of positive integers $\mathbf{r}$ (inequality is understood entry-wise) forms an *irreducible algebraic variety* (see details in [11]), which we denote by $\mathcal{M}_{\mathbf{n},\mathbf{r}}$.

A subset of this set consisting only of those tensors whose inner cores are equal, or in other words $G_2 = G_3 = \ldots = G_{d-1}$, is denoted by $\mathcal{M}^{eq}_{\mathbf{n},\mathbf{r}}$.

Remark 1. *The condition* $G_2 = G_3 = \ldots = G_{d-1}$ *implies that* $n_2 = n_3 = \ldots = n_{d-1}$ *and*

$r_1 = r_2 = \ldots = r_{d-1}$, *i.e.* $\mathcal{M}^{eq}_{\mathbf{n},\mathbf{r}}$ *is defined only for such* $\mathbf{n}, \mathbf{r}$ *that* $\mathbf{n} = (n_{first}, \underbrace{n_{inner}, \ldots, n_{inner}}_{d-2 \text{ times}}, n_{last})$ *and* $\mathbf{r} = (\underbrace{r, r, \ldots, r}_{d-1 \text{ times}})$.

Let us denote $\mathcal{G}[n_{first}, n_{inner}, n_{last}, r] := \mathbb{R}^{n_{first} \times r} \times \mathbb{R}^{r \times n_{inner} \times r} \times \mathbb{R}^{r \times n_{last}}$. *Then*

$$\mathcal{M}^{eq}_{\mathbf{n},\mathbf{r}} = \left\{ G_{first} \mathbin{@} \underbrace{G_{inner} \mathbin{@} \ldots \mathbin{@} G_{inner}}_{d-2 \text{ times}} \mathbin{@} G_{last} : (G_{first}, G_{inner}, G_{last}) \in \mathcal{G}[n_{first}, n_{inner}, n_{last}, r] \right\},$$

where @ is a tensor dot product.

In paper [7], Khrulkov, V. et al. showed that a recurrent-type neural network with d layers, each implementing a bilinear function, lies in the hypotheses space defined by formula 4.1, with the parameters of each of the d layers being exactly equal to the TT-cores $G_1, \ldots, G_d$ of the corresponding Tensor-Train decomposition of the coefficient tensor $\mathcal{W}_y$. Moreover, the TT-ranks $r_1, r_2, \ldots, r_{d-1}$ of the corresponding Tensor-Train decomposition of $\mathcal{W}_y$ are equal to the dimensions of the hidden states after, respectively, the first layer, the second layer and so on up to the $(d-1)$-th layer. Further we call such networks *Tensor-Train networks*. One of the main results of paper [7] is the theorem about the lower bound on the CP-rank for almost all tensors from $\mathcal{M}_{\mathbf{n},\mathbf{r}}$. A similar result for $\mathcal{M}^{eq}_{\mathbf{n},\mathbf{r}}$ is of particular interest because it is about Tensor-Train networks with equal inner layers, which is closest to traditional recurrent neural networks. We claim to have proved this result, which is formulated as the following theorem.

Theorem 1. *Suppose that* $d = 2k$ *is even. Define the following set*

$$B := \left\{ \mathcal{X} \in \mathcal{M}^{eq}_{\mathbf{n},\mathbf{r}} : \text{rank}_{CP}\, \mathcal{X} < q^{\frac{d}{2}} \right\},$$

where $q = \min\{n, r-1\}$.

Then

$$\mu(B) = 0,$$

where μ *is the standard Lebesgue measure on* $\mathcal{M}^{eq}_{\mathbf{n},\mathbf{r}}$.

To prove this theorem we need to formulate Lemma 2, which is proved in [7].

Lemma 2 (rank of matricization)**.** *Let* $\mathcal{X}^{i_1 i_2 \ldots i_d}$ *and* $\text{rank}_{CP}\, \mathcal{X} = r$. *Then for any matricization* $\mathcal{X}^{(s,t)}$ *we have* $\text{rank}\, \mathcal{X}^{(s,t)} \leqslant r$, *where the ordinary matrix rank is assumed.*

Proof. Our proof is based on applying Lemma 2 to a particular matricization of $\mathcal{X}$. Namely, we would like to show that for $s = \{1, 3, \ldots, d-1\}$, $t = \{2, 4, \ldots, d\}$ the following set

$$B^{(s,t)} := \left\{ \mathcal{X} \in \mathcal{M}^{eq}_{\mathbf{n},\mathbf{r}} : \text{rank}\, \mathcal{X}^{(s,t)} \leqslant q^{\frac{d}{2}} - 1 \right\}$$

has measure 0. Indeed, by Lemma 2 we have

$$B \subset B^{(s,t)},$$

so if $\mu(B^{(s,t)}) = 0$ then $\mu(B) = 0$ as well. Note that $B^{(s,t)}$ is an algebraic subset of $\mathcal{M}^{eq}_{\mathbf{n},\mathbf{r}}$ given by the conditions that the determinants of all $q^{\frac{d}{2}} \times q^{\frac{d}{2}}$ submatrices of $\mathcal{X}^{(s,t)}$ are equal to 0. Thus to show that $\mu(B^{(s,t)}) = 0$ we need to find at least one $\mathcal{X}$ such that $\text{rank}\, \mathcal{X}^{(s,t)} \geqslant q^{\frac{d}{2}}$. This follows from the fact that because $B^{(s,t)}$ is an algebraic subset of the irreducible algebraic variety $\mathcal{M}^{eq}_{\mathbf{n},\mathbf{r}}$, it is either equal to $\mathcal{M}^{eq}_{\mathbf{n},\mathbf{r}}$ or has measure 0, as was explained before.

One way to construct such tensor is as follows. Let us define the following tensors:

$$G_1^{i_1\alpha_1} = [i_1 = \alpha_1], \quad G_1 \in \mathbb{R}^{1\times n\times r}$$

$$G_k^{\alpha_{k-1}i_k\alpha_k} = \begin{cases} [\alpha_{k-1} = i_k]\cdot[\alpha_k = q+1], & \text{if } \alpha_{k-1} \leqslant q \\ [i_k = \alpha_k], & \text{if } \alpha_{k-1} = q+1 \\ 0, & \text{if } \alpha_{k-1} > q+1 \end{cases}, \quad G_k \in \mathbb{R}^{r\times n\times r}, \quad k = 2,3,\ldots,d-1$$

$$G_d^{\alpha_{d-1}i_d} = [\alpha_{d-1} = i_d], \quad G_d \in \mathbb{R}^{r\times n\times 1},$$

where $[\,\cdot\,]$ is the Iverson bracket notation.

The TT-ranks of the tensor $\mathcal{X}$ defined by the TT-cores are equal to $\text{rank}_{TT}\,\mathcal{X} = (r, r, \ldots, r)$.

Lemma 3 (structure of non-zero summands)**.** *Consider* $(i_1, i_2, \ldots, i_d) \in [q]^d$ *and* $(\alpha_1, \alpha_2, \ldots, \alpha_{d-1}) \in [r]^{d-1}$ *such that*

$$G_1^{i_1\alpha_1} \cdot \ldots \cdot G_d^{\alpha_{d-1}i_d} \neq 0.$$

Then

$$\alpha_k = \begin{cases} i_k, & \textit{if } k \textit{ is odd} \\ q+1, & \textit{if } k \textit{ is even} \end{cases} \qquad \textit{for any } k \in [d-1].$$

Proof. Let us prove the lemma by induction over k.

- *Base of the induction:* $k = 1$.

  Since $G_1^{i_1\alpha_1} = [i_1 = \alpha_1] \neq 0$, then $\alpha_1 = i_1$.

- *The induction step:* $k \to k+1$, $k \in [d-2]$.

  If k is odd, then $\alpha_k = i_k \in [q]$. Hence $G_{k+1}^{\alpha_k i_{k+1}\alpha_{k+1}} = [\alpha_k = i_{k+1}]\cdot[\alpha_{k+1} = q+1]$. Since $G_{k+1}^{\alpha_k i_{k+1}\alpha_{k+1}} \neq 0$, then $[\alpha_{k+1} = q+1] \neq 0$, so $\alpha_{k+1} = q+1$.

  If k is even, then $\alpha_k = q+1$. Hence $G_{k+1}^{\alpha_k i_{k+1}\alpha_{k+1}} = [i_{k+1} = \alpha_{k+1}]$. Since $G_{k+1}^{\alpha_k i_{k+1}\alpha_{k+1}} \neq 0$, then $[i_{k+1} = \alpha_{k+1}] \neq 0$, so $\alpha_{k+1} = i_{k+1}$.

□

Lets consider the following matricization of the tensor $\mathcal{X}$

$$\mathcal{X}^{(i_1,i_3,\ldots,i_{d-1}),(i_2,i_4,\ldots,i_d)}$$

The following identity holds true for any values of indices such that $i_k = 1,\ldots,q$, $k = 1,\ldots,d$.

$$\mathcal{X}^{(i_1,i_3,\ldots,i_{d-1}),(i_2,i_4,\ldots,i_d)} = \sum_{\alpha_1,\ldots,\alpha_{d-1}} G_1^{i_1\alpha_1}\ldots G_d^{\alpha_{d-1}i_d} =$$

$$\text{(by Lemma 3)} = G_1^{i_1 i_1} G_2^{i_1 i_2 (q+1)} G_3^{(q+1) i_3 i_3} \ldots G_d^{i_{d-1} i_d} =$$

$$\Big([i_1 = i_1]\Big)\cdot\Big([i_1 = i_2]\cdot[q+1 = q+1]\Big)\cdot\Big([i_3 = i_3]\Big)\cdot\ldots\cdot\Big([i_{d-1} = i_d]\Big) =$$

$$[i_1 = i_2]\cdot[i_3 = i_4]\cdot\ldots\cdot[i_{d-1} = i_d].$$

We obtain that

$$\mathcal{X}^{(i_1,i_3,\ldots,i_{d-1}),(i_2,i_4,\ldots,i_d)} = [i_1 = i_2]\cdot[i_3 = i_4]\cdot\ldots\cdot[i_{d-1} = i_d] = I^{(i_1,i_3,\ldots,i_{d-1}),(i_2,i_4,\ldots,i_d)},$$

where I is the identity matrix of size $q^{\frac{d}{2}} \times q^{\frac{d}{2}}$.

To summarize, we found an example of a tensor $\mathcal{X}$ such that $\text{rank}_{TT}\, \mathcal{X} \leqslant \mathbf{r}$ and the matricization $\mathcal{X}^{(i_1,i_3,\ldots,i_{d-1}),(i_2,i_4,\ldots,i_d)}$ has a submatrix being equal to the identity matrix of size $q^{\frac{d}{2}} \times q^{\frac{d}{2}}$, and hence $\text{rank}\, \mathcal{X}^{(i_1,i_3,\ldots,i_{d-1}),(i_2,i_4,\ldots,i_d)} \geqslant q^{\frac{d}{2}}$. This means that the canonical $\text{rank}_{CP}\, \mathcal{X} \geqslant q^{\frac{d}{2}}$ which concludes the proof. □

Theorem 1 gives a lower bound for almost all tensors of order d with modes $\mathbf{n}$ with Tensor-Train rank at most r. Using the connection between Tensor-Train decomposition and recurrent neural networks presented in [7], we have that traditional RNNs, which corresponds to tensors from $\mathcal{M}^{eq}_{\mathbf{n},\mathbf{r}}$, are exponentially more expressive than shallow neural networks.

Chapter 5
Methodology

The main results presented in the thesis are theoretical, which do not require additional equipment. The results associated with experiments on various neural network architectures may require an execution of programs in Python, for which it is enough to use such a tool for interactive execution of programs in Python as Jupyter Notebook. Some experiments uses Tensorlab [12] which is a MATLAB [6] package for tensor computations.

Chapter 6
Numerical experiments

I distinguish two different purposes of numerical experiments:

- to confirm numerically the result of the Theorem 1, i.e., to generate a random tensor from $\mathcal{M}^{eq}_{\mathbf{n},\mathbf{r}}$, evaluate numerically its CP-rank and compare it with the theoretical bound (see Section 6.1);
- to show that Tensor-Train network with equal inner TT-cores copes with classification tasks on real data.

All performed numerical experiments are available via the link to the Github repository.

6.1 Numerical verification of Theorem 1

Any experiment in this section consists of several steps:

1. fixing the parameters n_{first}, n_{inner}, n_{last}, r and the upper bound $d_{\max}$ for depth d;
2. selection of "typical" TT-cores G_{first}, G_{inner}, G_{last} from $\mathcal{G}[n_{\text{first}}, n_{\text{inner}}, n_{\text{last}}, r]$ (see Remark 1);
3. numerical evaluation of the CP-rank of the tensors from $\mathcal{M}^{eq}_{\mathbf{n},\mathbf{r}}$ generated by the TT-cores G_{first}, G_{inner}, G_{last} for each d from $\{1, \ldots, d_{\max}\}$;
4. aggregation of the results and plotting of the graph.

Let us fix $n_{\text{first}} = n_{\text{inner}} = n_{\text{last}} = 2$, $r = 3$. Theorem 1 gives us the following lower bound on CP-rank of almost all tensors from $\mathcal{M}^{eq}_{\mathbf{n},\mathbf{r}}$: $q^{\frac{d}{2}} = \min\{n_{\text{first}}, n_{\text{inner}}, n_{\text{last}}, r-1\}^{\frac{d}{2}} = 2^{\frac{d}{2}}$.

Since the standard Lebesgue measure on $\mathcal{G}[n_{\text{first}}, n_{\text{inner}}, n_{\text{last}}, r]$ is not a probability, we will choose TT-cores G_{first}, G_{inner}, G_{last} using arbitrary distribution on $\mathcal{G}[n_{\text{first}}, n_{\text{inner}}, n_{\text{last}}, r]$, for example, standard normal.

A direct way to estimate the canonical rank of tensor $\mathcal{X}$ is to search for a low-rank approximation of tensor $\mathcal{X}$ by increasing the CP-rank of an approximation. If at some CP-rank the low-rank approximation differs from tensor $\mathcal{X}$ by a negligible error, then the number of summands in the low-rank approximation will be equal to the CP-rank of $\mathcal{X}$. The search for the low-rank approximation was performed both using gradient descent by minimising the Frobenius norm of the difference between $\mathcal{X}$ and the approximation, and using a function `cpd` (Canonical polyadic decomposition) from Tensorlab [12] which is a MATLAB [6] package.

The results of these experiments are shown in the following graph (see Figure 6.1).

The graph shows that the theoretical lower bound derived from Theorem 1 is lower than estimated CP-rank of chosen tensor for all considered values of depth d.

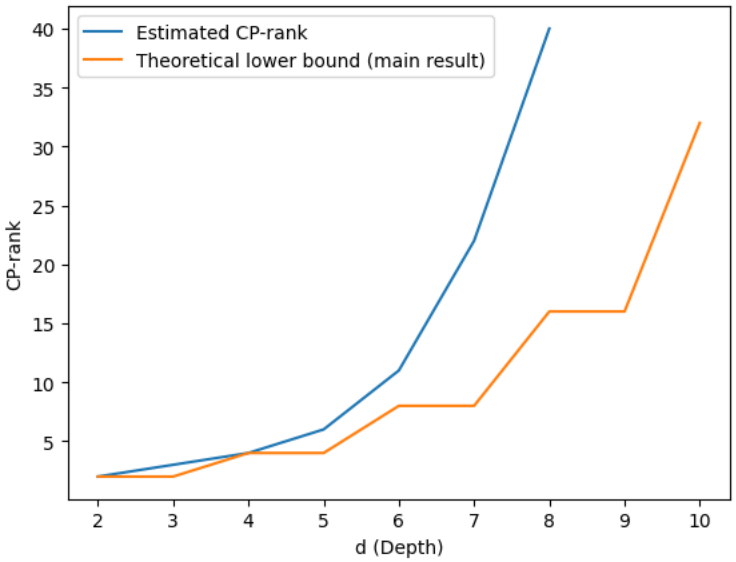

Figure 6.1: The graph compares the theoretical lower bound on CP-rank (orange line) and the numerical CP-rank estimate of the random tensor from $\mathcal{M}_{2,3}^{eq}$ (blue line).

6.2 Experiments on the MNIST dataset

In this section we consider standard computer vision datasets MNIST which is a collection of 70000 handwritten digits.

We have implemented Tensor-Train networks with both different and the same inner TT-cores using PyTorch [9]. We use Adam optimizer with batch size 64 and learning rate $5\text{e}{-4}$. Each picture of size 28×28 pixels is split into 16 non-overlapping 7×7 patches, whose low-dimensional representations are alternately fed to the Tensor-Train network of depth 16.

The training process of these networks was as follows: in the initial stage, we added `BatchNorm1d` layer after each recurrent layer of the Tensor-Train network and trained such a network for 20 epochs; in the second stage, we removed the normalisation layers and further trained the Tensor-Train network for four epochs. This two-step approach avoided the problem of stopping training at the initial stage.

For MNIST, both Tensor-Train networks and Tensor-Train networks with equal cores show reasonable performance (see Table 6.1).

Table 6.1: The results of training Tensor-Train networks on MNIST dataset during 20+4 epochs.

	TT-net with different cores	TT-net with equal cores
Train accuracy	98.5	96.3
Test accuracy	97.3	95.7

6.3 Experiments on the CIFAR-10 dataset

In this section we consider computer vision datasets CIFAR-10 which is a collection of 60000 32×32 color images in 10 different classes.

We use Adam optimizer with batch size 64 and learning rate $5\text{e}{-4}$. Each picture of size 32×32 pixels is split into 16 non-overlapping 8×8 patches, whose low-dimensional representations are alternately fed to the Tensor-Train network of depth 16.

The training process of these networks was as follows: in the initial stage, we added `BatchNorm1d` layer after each recurrent layer of the Tensor-Train network and trained such a network for 40 epochs; in the second stage, we removed the normalisation layers and further trained the Tensor-Train network for ten epochs. This two-step approach avoided the problem of stopping training at the initial stage.

Despite the fact that we failed to train Tensor-Train networks with high accuracy to distinguish classes, a network with equal TT-cores is not worse in accuracy than a network with different TT-cores, but even better (see Table 6.2).

Table 6.2: The results of training Tensor-Train networks on CIFAR-10 dataset during 40+10 epochs.

	TT-net with different cores	TT-net with equal cores
Train accuracy	16.1	37.3
Test accuracy	15.9	37.6

Chapter 7
Discussion and conclusion

We have proved a theorem on the lower bound of rank of almost all tensors of fixed size and bounded Tensor-Train rank having equal inner TT-cores. We validated the obtained theoretical result using numerical experiment (see Section 6.1). Thus, we close the question of the exponential expressiveness of traditional recurrent neural networks.

Through experiments described in Sections 6.2, 6.3, we also established that Tensor-Train networks are expressive enough to solve the classification problem on real-world datasets. Moreover, our results can be interpreted that Tensor-Train networks with equal inner TT-cores are as expressive as Tensor-Train networks with different TT-cores, despite the fact that the first ones have significantly fewer trainable parameters than the second ones.

The unexpectedly higher accuracy of the Tensor-Train network with equal inner TT-cores compared to the Tensor-Train network with different TT-cores on the CIFAR-10 dataset (see Section 6.3) leaves room for further research.

Bibliography

[1] Cohen, N., Sharir, O., and Shashua, A. On the expressive power of deep learning: A tensor analysis. *In Conference on Learning Theory* (2016), 698–728.

[2] Cohen, N., and Shashua, A. Convolutional rectifier networks as generalized tensor decompositions. *In International Conference on Machine Learning* (2016), 955–963.

[3] Delalleau, O., and Bengio, Y. Shallow vs. deep sum-product networks. *In Advances in Neural Information Processing Systems* (2011), 666–674.

[4] Hastad, J. Almost optimal lower bounds for small depth circuits. *In Proceedings of the eighteenth annual ACM symposium on Theory of computing* (1986), 6–20.

[5] Hastad, J., and Goldmann, M. On the power of small-depth threshold circuits. *Computational Complexity* (1991), 1(2):113–129.

[6] Inc., T. M. Matlab version: 9.14.0 (r2023a), 2022.

[7] Khrulkov, V., Novikov, A., and Oseledets, I. Expressive power of recurrent neural networks. *arXiv preprint arXiv:1711.00811* (2018).

[8] Martens, J., and Medabalimi, V. On the expressive efficiency of sum product networks. *arXiv preprint arXiv:1411.7717* (2014).

[9] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems 32* (2019).

[10] Raghu, M., Poole, B., Kleinberg, J., Ganguli, S., and Sohl-Dickstein, J. On the expressive power of deep neural networks. *Proceedings of Machine Learning Research 70* (06–11 Aug 2017), 2847–2854.

[11] Shafarevich, I. R., and Hirsch, K. A. *Basic algebraic geometry, vol. 2*. Springer, 1994.

[12] Vervliet, N., Debals, O., Sorber, L., Van Barel, M., and De Lathauwer, L. Tensorlab 3.0, Mar. 2016. Available online.