On Tensor-Rank and Lower Bounds

A presentation on tensors, some results and Raz's paper

Ali Karim Lalani[1]

[1]University of Toronto

CSC2429 Presentation, March 2026

Table of Contents

Table of Contents

What is a Tensor?

- A tensor T is a multidimensional array of scalars over a field $\mathbb{F}$.
- More formally, it is a function that maps a d-tuple of indices to a scalar in $\mathbb{F}$.

The "Lookup Rule" Perspective

Think of a tensor as a **lookup rule**.

$$T : [n]^d \to \mathbb{F}$$

Given an address $(i_1, i_2, \ldots, i_d)$, the tensor "looks up" and returns the value stored at that location.

Examples by Dimension

The dimension d tells us how many indices we need to look up a value.

Examples

$d = 1$: **Vectors**

$$v : [n] \to \mathbb{F}, \quad i \mapsto v_i$$

$d = 2$: **Matrices**

$$M : [n]^2 \to \mathbb{F}, \quad (i, j) \mapsto M_{ij}$$

$d = 3$: **3D Grids**

$$T : [n]^3 \to \mathbb{F}, \quad (i, j, k) \mapsto T_{ijk}$$

Alternative Approach: Multilinear Algebra

From Landsberg's *Geometry and Complexity Theory*:

- For vector spaces $A_1, \ldots, A_n$, the tensor product $A_1 \otimes \cdots \otimes A_n$ is the space of n-linear maps:

$$A_1^* \times \cdots \times A_n^* \to \mathbb{C}$$

- Alternatively, it is the space of $(n-1)$-linear maps:

$$A_1^* \times \cdots \times A_{n-1}^* \to A_n$$

Element Definition

Let $a_j \in A_j$. The element $a_1 \otimes \cdots \otimes a_n$ is the n-linear map defined by:

$$a_1 \otimes \cdots \otimes a_n(\alpha^1, \ldots, \alpha^n) := \alpha^1(a_1) \cdots \alpha^n(a_n)$$

Rank 1 Tensors

Definition: Rank 1

A tensor A has **rank 1** if there exist vectors $u^{(1)}, u^{(2)}, \ldots, u^{(d)} \in \mathbb{F}^n$ such that every entry is a simple product of the vector components:

$$A(i_1, \ldots, i_d) = u^{(1)}_{i_1} \cdot u^{(2)}_{i_2} \cdots u^{(d)}_{i_d}$$

- These are the "building blocks" of all other tensors.
- **Intuition:** In a rank-1 tensor, the different dimensions do not interact in weird ways.

Defining Tensor Rank $rk(T)$

Tensor Rank

The **rank** of a d-dimensional tensor T, denoted $rk(T)$, is the minimum number of rank-1 tensors needed to represent T as their sum:

$$rk(T) = \min \left\{ r : T = \sum_{i=1}^{r} u_i^{(1)} \otimes u_i^{(2)} \otimes \cdots \otimes u_i^{(d)} \right\}$$

where each $u_i^{(j)} \in \mathbb{F}^n$.

- This is more or less a definition by **complexity measure**, it counts the minimum number of simple blocks required to describe T.
- **Generalization of Matrix Rank:** This is the natural extension of the $d = 2$ case. One characterization of matrix rank is the smallest r such that $M = \sum_{i=1}^{r} u_i v_i^T$.

Concrete Examples: Rank 1 vs. High Rank

Example

Let $u = (1, 2)$, $v = (1, 0)$, and $w = (0, 1)$. The resulting tensor $T(i, j, k) = u_i \cdot v_j \cdot w_k$ has rank 1.

Example of a Non-Simple Tensor

Not all tensors can be written as a single product. Consider $T : [2]^3 \to \mathbb{C}$:

- $T(1, 1, 1) = 1$
- $T(2, 2, 2) = 1$
- All other entries are 0.

This "diagonal" tensor cannot be represented by a single rank-1 block; its rank is 2.

Fundamental Questions

There are two primary questions that drive the research in this field:

1. **The Search Problem:** Given a specific tensor T, how do we efficiently find $rk(T)$?
2. **The Existence Problem:** How do we find or construct *explicit* tensors that have a guaranteed high rank?

Complexity

Håstad (1990) proved that computing the rank of a tensor is **NP-Hard**, and NP-complete for finite fields.

A Basic Upper Bound: $rk(T) \leq n^{d-1}$

Theorem

Any tensor $T : [n]^d \to \mathbb{F}$ has $rk(T) \leq n^{d-1}$.

Proof for $n = 2, d = 3$.

Let $\{e_1, e_2\}$ be the standard basis for $\mathbb{F}^2$. We can decompose T by "freezing" the first $d-1$ indices. For each fixed pair $(i,j) \in \{1,2\}^2$, define the vector $v^{(i,j)} \in \mathbb{F}^2$ as $v^{(i,j)} = \begin{pmatrix} T(i,j,1) \\ T(i,j,2) \end{pmatrix}$ We can then write the entire tensor as a sum of $2^2 = 4$ terms:

$$T = \sum_{i=1}^{2} \sum_{j=1}^{2} e_i \otimes e_j \otimes v^{(i,j)}$$

Since each $e_i \otimes e_j \otimes v^{(i,j)}$ is a rank-1 tensor, $rk(T) \leq 4$.

Rank of a Random Tensor

While the upper bound is n^{d-1}, most tensors (i.e. if you pick random tensor) have a high rank. We can see this via a **dimension counting argument**.

- **General Tensor:** Requires n^d parameters to specify (all entries).
- **Rank-1 Tensor:** Specified by d vectors of size n, totaling $\approx nd$ parameters.

Heuristic for Generic Rank

To span the space of all tensors (n^d degrees of freedom) using blocks of nd parameters, we expect the rank of a generic tensor to be:

$$rk(T) \geq \frac{n^d}{nd} = \frac{n^{d-1}}{d}$$

The "Pathetic" State of Explicit Bounds

While a random tensor probably has rank $\approx n^2/3$, we currently struggle to construct *explicit* tensors with rank even slightly better than n.

Flattening

To use linear algebra, we can "flatten" a d-dimensional tensor into a 2D matrix.

- Any d-dimensional tensor can be viewed as a matrix by partitioning its indices.
- Select a subset of indices for the rows and the remaining indices for the columns.
- **Lower Bound Property:** For any flattening M of T, $rk(T) \geq rank(M)$.

The Mapping

For a tensor $T : [n]^d \to \mathbb{F}$, we map:

$$(i_1, \ldots, i_k, \ldots, i_d) \to (RowIndex, ColumnIndex)$$

This preserves the data while enabling the use of the standard matrix rank.

Example: Flattening $T : [2]^3 \to \mathbb{F}$

A $2 \times 2 \times 2$ tensor into a 2×4 matrix by "freezing" the first index i.

Front Slice ($k = 1$)

$$\begin{pmatrix} 1 & 3 \\ 2 & 4 \end{pmatrix}$$

Back Slice ($k = 2$)

$$\begin{pmatrix} 5 & 7 \\ 6 & 8 \end{pmatrix}$$

Flattened Matrix M

$$\begin{pmatrix} 1 & 3 & 5 & 7 \\ 2 & 4 & 6 & 8 \end{pmatrix}$$

- Rows are indexed by $i \in \{1, 2\}$.
- Columns are indexed by the pairs (j, k).
- M has full rank (2), so $rk(T) \geq 2$.

The "Pathetic" State of Explicit Bounds

The flattening method is currently the only robust way we have to prove explicit lower bounds for tensor rank.

The $n^{\lfloor d/2 \rfloor}$ bound

By taking a "diagonal" tensor and projecting it, we can prove:

$$rk(T) \geq n^{\lfloor d/2 \rfloor}$$

- For $d = 3$, this bound is just n, which is **trivial**.
- We struggle to improve this; we don't even know an explicit 4-tensor with rank $2n^2$.

Tensors as Multilinear Forms

The reason why we're here in this talk is because tensors and tensor rank do have a connection to algebraic complexity. We can represent any d-dimensional tensor as a set-multilinear polynomial.

Definition: The Associated Polynomial (f_T)

Let $T : [k]^d \to \mathbb{F}$ be a d-dimensional tensor. The associated polynomial f_T in variables $\{X_{i,j} : i \in [d], j \in [k]\}$ is defined as:

$$f_T(X_{1,1}, \ldots, X_{d,k}) = \sum_{(i_1,\ldots,i_d)\in[k]^d} T(i_1, \ldots, i_d) \cdot \prod_{j=1}^{d} X_{j,i_j}$$

- Each monomial contains exactly one variable from each of the d groups.
- Tensor rank corresponds exactly to the size of the smallest depth-3 set-multilinear formula computing f_T.

The Ran Raz Breakthrough

Raz's result says strong lower bounds for Tensor Rank imply lower bounds for Arithmetic Formulas.

The main result

Any **explicit** example of a tensor $A : [n]^d \to \mathbb{F}$ with:

$$rk(A) \geq n^{d \cdot (1-o(1))}$$

where $d \leq \log n / \log \log n$ is super-constant, implies an **explicit super-polynomial lower bound** for general arithmetic formulas.

- **Significance:** Depth-3 lower bounds (which are geometric) can prove general computational lower bounds.
- **The Goal:** If we can describe one specific tensor A and prove its rank is very high, we have found a polynomial that is hard for any general formula.

Table of Contents

Nisan and Wigderson's Conjecture

- **The Question:** If a polynomial f has a polynomial-size formula, does it have a polynomial-size *homogeneous* formula?
- **Nisan-Wigderson Conjecture:** They conjectured that if f has a poly-size formula, its homogeneous components have poly-size homogeneous **circuits**.
- **The Naive Method's Failure:**
  - Standard homogenization splits every node v into $r+1$ nodes $(v_0, \ldots, v_r)$.
  - In a formula (tree), these new nodes would need to be duplicated to keep the out-degree 1.
  - This leads to an **exponential blowup** in size, $s^{\Omega(r)}$, destroying the formula's efficiency.

Raz's Theorem 1

Theorem 1 (Raz)

Let Φ be a fan-in 2 formula of size s and product-depth d that computes a homogeneous polynomial of degree r. Then there exists a fan-in 2 homogeneous formula Ψ of size:

$$\text{Size}(\Psi) = O\left(\binom{d+r+1}{r} \cdot s\right)$$

and product-depth d that computes the same polynomial.

Proof.

Done in HW1 Q5 □

Proof Sketch

We construct Ψ by creating nodes (u, D) where $D \in N_u$.

- N_u: Intuitively, the set of all valid degree progression functions along the unique path from u to the root.
- D: A function $D : path(u) \to \{0, \ldots, r\}$ that tracks the target degree.

Notation and Functionality

Each node (u, D) in Ψ is designed to compute:

$$\hat{\Psi}_{(u,D)} = \hat{\Phi}_{u,D(u)}$$

Where $\hat{\Phi}_{u,D(u)}$ is the homogeneous part of degree $D(u)$ of the polynomial computed at node u in the original formula.

- We wire these inductively: Leaves output variables/constants based on $D(u)$, Sum gates pass the degree down, and Product gates perform a convolution.

Combinatorics: Stars and Bars

Why is the blowup $\binom{d+r+1}{r}$?

- A function D is defined by the jumps in degree it makes.
- Degrees only change at **Product Gates**.
- There are at most d product gates on any path.
- We are essentially distributing a total degree "budget" of r into $d+1$ slots (the segments between product gates).

Stars and Bars

By the Stars and Bars formula, the number of ways to choose these non-decreasing degrees is exactly $\binom{d+r+1}{r}$. This is the number of copies we need for each original node u.

Corollary 2: Completing the Lift

Corollary 2

Let f be a homogeneous polynomial of degree $r \leq O(\log n)$. If f has a poly-size formula, then it has a poly-size **homogeneous** formula.

Proof.

1. Restructure Φ to have depth $d = O(\log n)$ (Standard result, HW1 Q4).
2. Apply Theorem 1.
3. The size blowup is $\binom{O(\log n)+O(\log n)+1}{O(\log n)}$, which is $2^{O(\log n)} = \text{poly}(n)$.

□

- This refutes the conjecture that homogenization ought to be expensive for formulas!

Multilinearization: New Notation

To transition from homogeneous polynomials to **set-multilinear** ones, we need to track which variable sets are being used.

- **Bitwise Comparison:** For vectors $a, b \in \{0,1\}^r$, we say $b \leq a$ if $b(i) \leq a(i)$ for all $i \in \{1, \ldots, r\}$.
- **Variable Sets:** Let $X_1, \ldots, X_r$ be disjoint sets of variables.

Type of a Monomial

A monomial q is **set-multilinear of type** a if it is multilinear and contains:

- Exactly one variable from X_i if $a(i) = 1$.
- No variables from X_i if $a(i) = 0$.

- **Notation:** f_a denotes the set-multilinear part of polynomial f of type a. Similarly,for an arithmetic formula Φ, let $\hat{\Phi}_{u,a}$ denote the type-a part of the polynomial computed at node u.

Theorem 3: The Multilinearization Theorem

Just as Theorem 1 homogenized a formula, Theorem 3 extracts its set-multilinear part.

Theorem 3

Let Φ be a fan-in 2 formula of size s and product-depth d that computes a set-multilinear polynomial over the disjoint sets $X_1, \ldots, X_r$. Then there exists a fan-in 2 set-multilinear formula Ψ of size

$$\text{Size}(\Psi) = O\left((d+2)^r \cdot s\right)$$

and product-depth d that computes the same polynomial.

- **Contrast with Homogenization:** The blowup is now bounded by $(d+2)^r$ instead of $\binom{d+r+1}{r}$.

Proof Sketch: Tracking Types

The proof structure is identical to Theorem 1, but instead of tracking an integer degree, we track the **type vector** $a \in \{0,1\}^r$.

- **State Space N_u:** The set of functions $D : \text{path}(u) \to \{0,1\}^r$.
- **Constraints:**
  - Sum gate v: $D(v) = D(child)$
  - Product gate v: $D(v) \geq D(child)$ (bitwise)

Inductive Construction

We construct nodes (u, D) computing $\hat{\Psi}_{(u,D)} = \hat{\Phi}_{u,D(u)}$. For a product gate $u = v \cdot w$, we sum over all valid type partitions:

$$\hat{\Psi}_{(u,D)} = \sum_{a \leq D(u)} \hat{\Psi}_{(v,D_{v,a})} \cdot \hat{\Psi}_{(w,D_{w,D(u)-a})}$$

- **Size Bound:** Each of the r coordinates can flip from 0 to 1 at most once along the $\leq d$ product gates. This bounds $|N_u|$ by $(d+2)^r$.

Corollary 4

Corollary 4

Let f be a set-multilinear polynomial over sets $X_1, \ldots, X_r$ of size n each. If $r \leq O(\log n / \log \log n)$ and f has a poly-size formula, then it has a poly-size **set-multilinear** formula.

Proof.

1. Assume WLOG that the initial poly-size formula Φ has depth $d = O(\log n)$.
2. Apply Theorem 3. The new size is bounded by $O((d+2)^r \cdot s)$.
3. Look at the blowup factor:

$$(d+2)^r \approx (O(\log n))^{O\left(\frac{\log n}{\log \log n}\right)} = 2^{O\left(\log \log n \cdot \frac{\log n}{\log \log n}\right)} = 2^{O(\log n)}$$

4. $2^{O(\log n)} = \text{poly}(n)$. Thus, the final formula is polynomial size!

Theorem 5: The Main Connection

Raz ties formula size directly to an upper bound on tensor rank.

Theorem 5

Let $A : [n]^r \to \mathbb{F}$ be a tensor where $r \leq O(\log n / \log \log n)$. If the associated polynomial f_A has a formula of size n^c, then:

$$rk(A) \leq n^{r \cdot (1 - 2^{-O(c)})}$$

- **The Crucial First Step:** Where do we use our previous theorems?
- We start by assuming Φ is a formula of size n^c. By **Theorem 3** (Multilinearization) and our bound on r, we can assume WLOG that Φ is a **set-multilinear** formula of size $n^{O(c)}$.
- Without this step, the rest of the structural analysis would be impossible!

Basic Properties of Tensor Rank

To analyze the formula, Raz uses these straightforward properties of tensor rank.

For any tensors $A_1, A_2, \ldots$ over $\mathbb{F}$:

1. **Trivial Upper Bound:** For a tensor of order $r' > 0$,

$$rk(A') \leq n^{r'-1}$$

2. **Subadditivity:** The rank of a sum is at most the sum of the ranks.

$$rk\left(\sum_{i=1}^{k} A_i\right) \leq \sum_{i=1}^{k} rk(A_i)$$

3. **Submultiplicativity:** For $A_1 : [n]^{r_1} \to \mathbb{F}$ and $A_2 : [n]^{r_2} \to \mathbb{F}$,

$$rk(A_1 \otimes A_2) \leq rk(A_1) \cdot rk(A_2)$$

where $A_1 \otimes A_2(i_1, \ldots, i_{r_1+r_2}) = A_1(i_1, \ldots, i_{r_1}) \cdot A_2(i_{r_1+1}, \ldots, i_{r_1+r_2})$.

The Technical Machinery: A High-Level View

This proof is very techical. Raz introduces several specialized tools:

- **Normal Form:** He forces the set-multilinear formula into a strict binary tree where sum gates are collapsed and leaf labels are highly standardized.
- **Syntactic-Rank:** Instead of calculating the true tensor rank (which is hard), he defines an inductive upper-bound calculated directly from the formula's topology (using the three basic properties).
- **Extended-Formulas:** He assigns real-valued "weights" to nodes to track sub-formula duplication.

The Optimization Argument

The core of the proof is an analytical argument. By viewing the space of these "Extended-Formulas" as a compact topological space, he maximizes the Syntactic-Rank against the Size, proving the upper bound holds for all formulas.

Corollary 6: Super-Polynomial Lower Bounds

Corollary 6

Let $A : [n]^r \to \mathbb{F}$ be a tensor such that $r \leq O(\log n / \log \log n)$. If

$$rk(A) \geq n^{r \cdot (1 - o(1))}$$

then there is **no polynomial-size formula** for the polynomial f_A.

- This is essentially the contrapositive of Theorem 5.
- This lower bound is non-trivial only when $r = \omega(1)$ is super-constant. If r were constant, the rank is naturally capped by n^{r-1}, making the hypothesis impossible to satisfy.

Table of Contents

The Matrix Multiplication Tensor

One of the biggest drivers of tensor research is the complexity of Matrix Multiplication (MM).

The MM Tensor

The multiplication of an $a \times b$ matrix and a $b \times c$ matrix can be encoded as a specific 3-tensor $\langle a, b, c \rangle$:

$$\sum_{i=1}^{a} \sum_{k=1}^{b} \sum_{j=1}^{c} X_{i,k} \cdot Y_{k,j} \cdot Z_{i,j}$$

- The tensor rank of this specific tensor corresponds exactly to the number of multiplication gates needed to multiply the matrices!
- Naive multiplication gives $O(n^3)$, but Strassen (1969) showed $O(n^{2.807})$ by divide and conquer.
- **Open Question:** Can we get it down to n^2?

The Set of Low-Rank Tensors

Consider the set of all tensors of rank at most r:

$$V_r = \{T : rk(T) \leq r\}$$

What is this object? Is it a variety?

- **Yes, for matrices ($d = 2$):** We can define it by picking every possible $(r+1) \times (r+1)$ minor and finding the subdeterminants.
- These subdeterminants act as our "laws" (polynomials that vanish on the set).

Example

Rank ≤ 1 matrices must have 0 determinant. Thus,
$V_1 = \{\text{points in } \mathbb{F}^4 : ad - bc = 0\}$.

Making it into a Variety for $d > 2$

For general tensors ($d \geq 3$), V_r is **NOT** a variety! To make it into a variety, we must take its closure:

- Let $P_r = \{\text{all polynomials vanishing on } V_r\}$.
- We set $\overline{V}_r = \text{Zero}(P_r) = \{T : f_1(T) = 0, f_2(T) = 0, \dots\}$. This represents all tensors satisfying the required laws.

The Topological View

Equivalently, this closure contains all limit points:

$$\overline{V}_r = \{T : \exists \text{ family } \{T_\epsilon\} \text{ s.t. } T_\epsilon \xrightarrow{\epsilon \to 0} T, \text{ and } rk(T_\epsilon) \leq r\}$$

Conclusion: $\overline{V}_r$ contains not just low-rank tensors, but also limit points of low-rank tensors.

Border Rank: A Concrete Example

Definition: Border Rank

The **Border Rank**, denoted $brk(T)$, is the smallest r such that T is a limit of rank-r tensors.

Consider the following $2 \times 2 \times 2$ tensor T (written as two slices):

$$T = \left(\begin{array}{cc|cc} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{array}\right)$$

- The true tensor rank of this specific tensor is $rk(T) = 3$.
- However, look at the following limit as $\epsilon \to 0$:

$$T = \lim_{\epsilon \to 0} \underbrace{\left(\begin{array}{cc|cc} \frac{1}{\epsilon} & 1 & 1 & \epsilon \\ 1 & \epsilon & \epsilon & \epsilon^2 \end{array}\right)}_{\text{Rank 1}} - \underbrace{\left(\begin{array}{cc|cc} \frac{1}{\epsilon} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{array}\right)}_{\text{Rank 1}}$$

Since T is the limit of the difference of two rank-1 tensors, $brk(T) \leq 2$

Why Border Rank Matters

We have a major open question regarding the relationship between border rank and the standard tensor rank.

Open Question

How far apart can they be? Is there a nice f such that:

$$brk(T) \leq rk(T) \leq f(d) \cdot brk(T)$$

- **Matrix Multiplication (MM):** For the complexity of MM, it turns out it is sufficient to understand border rank.
- One way to interpret Strassen's famous $O(n^{2.807})$ result is actually that $brk(T_2^{MM}) \leq 7$.
- Landsberg proved that 7 is a tight bound for the 2×2 MM tensor. To prove things like this, we need to understand the polynomials that define this variety.

Uniqueness of Tensor Decompositions?

A major difference between matrices and tensors is the uniqueness of their decompositions.

The Question

Given a tensor $T = \sum_{i=1}^{r} u_i \otimes v_i \otimes w_i$, is this decomposition unique?

- **Jennrich's Algorithm** provides a powerful algorithmic guarantee.
- It requires two conditions:
  1. The sets $\{u_i\}$ and $\{v_i\}$ are linearly independent.
  2. The set $\{w_i\}$ does not contain parallel vectors.
- If these hold, the decomposition is unique (up to scaling and permutations) and can be found efficiently by a probabilistic algorithm.
- This uniqueness is used around in machine learning!

Kruskal's Condition for Uniqueness

What if the vectors aren't fully linearly independent? Kruskal provided a combinatorial condition.

- **Notation:** For a matrix U, let $i(U)$ be the maximum integer k such that *every* k columns of U are linearly independent.

Theorem (Kruskal)

Let $T = \sum_{i=1}^{r} u_i \otimes v_i \otimes w_i$. If we assume:

$$r \leq \frac{1}{2}(i(U) + i(V) + i(W)) - 1$$

Then $rk(T) = r$ and the decomposition is completely unique.

- The proof is clever and combinatorial, but it is entirely non-algorithmic.

Explicit Tensors with High Rank

We can leverage Kruskal's condition to explicitly construct a tensor with a provably high rank.

- Take U, V, W to be $n \times r$ matrices, where we set $r = \frac{3}{2}n - 1$.
- We choose U, V, and W to be **Vandermonde matrices**.
- A key property of the Vandermonde matrix is that any n columns are linearly independent.
- This perfectly satisfies Kruskal's requirement!

The Result

Thus, the tensor $T = \sum_{i=1}^{r} u_i \otimes v_i \otimes w_i$ formed by these matrices has rank exactly $r = \frac{3}{2}n - 1$.

Thank You!

Questions?

- **References:**
  - Raz, R. (2013). *Tensor-rank and lower bounds for arithmetic formulas.*
  - Wigderson, A. *Tensor Rank* (Lecture from IAS)