1 Tensor Networks

Our goal now is to focus on a subclass of models over Σ^*. To this end, we analyze *tensor networks.*

We denote a tensor T_v with k axes of sizes $D_v = \{d_1, \dots, d_k\}$ as a function

$$T_v : [d_1] \times \dots \times [d_k] \mapsto \mathbb{R} \quad .$$

As a shorthand, we write

$$[D_v] \coloneqq [d_1] \times \dots \times [d_k] \quad .$$

Since indexing is usually clear from context, we treat D_v as a multiset of axis sizes.

Given two tensors T_u and T_v that share a common axis of size d_e, their contraction over this axis produces a new tensor T_C with dimension set

$$D_C = (D_u \setminus \{d_e\}) \cup (D_v \setminus \{d_e\}) \quad ,$$

defined as

$$T_C(i) = \sum_{i_e \in [d_e]} T_u(i_{D_u}, i_e) \cdot T_v(i_{D_v}, i_e) \quad ,$$

where $i \in [D_C]$. Note that $d_e \notin D_C$, which is why we explicitly included index i_e in the summation.

Definition 1.1 (Tensor Network over Σ^n)**.** A *tensor network* $\mathcal{T}$ over Σ^n is defined by a graph $G = (V, E)$ with the following structure:

- V is the set of vertices, where each vertex $v = (\text{layer}, \text{index}) \in V$ corresponds to a tensor T_v with axis sizes $D_v = \{d_1, \dots, d_k\}$. Let $V_{\text{layer}} \subseteq V$ denote the set of all vertices at a given layer.
- The input set $I = (T_{0,1}, \dots, T_{0,n}) \subset V$ consists of tensors each having a single axis of size $|\Sigma|$. These serve as the one-hot-encoded inputs corresponding to a string $w \in \Sigma^n$.
- $E \subseteq \{\{u, v\} \mid u \in V_l,\, v \in V_{l+1}\}$ is the set of edges. Each edge $e = \{u, v\}$ represents a shared index of size d_e between tensors T_u and T_v, which is summed over during contraction.
- The usual tensor network constrains: For each vertex $v \in V$, the degree $\deg(v)$ must match the number of axes $|D_v|$, and shared indices must correspond to same axis sizes.

Once the input tensors are initialized with one-hot encodings derived from a string $w \in \Sigma^n$, the network computes a scalar output $\mathcal{T}(w)$. This induces a probability distribution over Σ^n defined by:

$$S_{n,\mathcal{T}}(w) \coloneqq \frac{f(\mathcal{T}(w))}{\sum_{w' \in \Sigma^n} f(\mathcal{T}(w'))} \quad ,$$

where $f : \mathbb{R} \mapsto \mathbb{R}_{\geq 0}$ is any arbitrary function like $f \equiv \exp$.

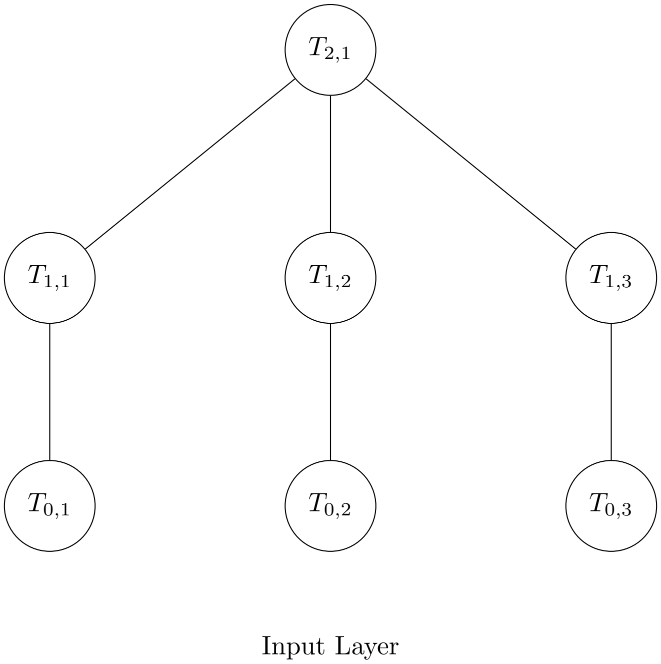


Figure 1: A basic tensor network over Σ^3.

Definition 1.2 (Normalized and Non-Negative Tensor Networks)**.** Let $\mathcal{T}$ be a tensor network over Σ^n with scalar output $\mathcal{T}(w)$ for each $w \in \Sigma^n$. Define the total mass of the network as

$$|\mathcal{T}| \coloneqq \sum_{w \in \Sigma^n} \mathcal{T}(w) \quad .$$

We say $\mathcal{T}$ is *normalized* iff $|\mathcal{T}| = 1$.

Furthermore, a tensor network is said to be *non-negative* iff for all $w \in \Sigma^n$ we have $\mathcal{T}(w) \geq 0$.

Remark 1.1. We can enforce all tensor networks of our model space to be non-negative by only allowing for non-negative tensors in the networks.

Definition 1.3 (Normalization of Tensor Networks)**.** Let $\mathcal{T}$ be a tensor network over Σ^n, and let $H := V \setminus I$ be the set of non-input tensors, and define $|H|$ as its cardinality. The *induced normalized tensor network* $\dfrac{\mathcal{T}}{|\mathcal{T}|}$ is the same network as $\mathcal{T}$, but each entry of each tensor in H is scaled by the factor $\dfrac{1}{\sqrt[|H|]{|\mathcal{T}|}}$.

Lemma 1.1. *Let $J \subseteq [n]$ and let $\mathcal{T}$ be a tensor network over Σ^n. Define a modified network $\mathcal{T}_J$ where for all $j \in J$, the input tensor $T_{0,j}$ is initialized to the all-ones vector (i.e., $\mathbf{1} \in \mathbb{R}^{|\Sigma|}$). Then for any $w \in \Sigma^{[n]\setminus J}$:*

$$\sum_{w_J \in \Sigma^{|J|}} \mathcal{T}(w_J, w) = \mathcal{T}_J(w) \quad .$$

Proof. We proceed by induction on the size of the subset $J \subseteq [n]$.

Base case: $|J| = 0$. Then $J = \emptyset$, so $\mathcal{T}_J = \mathcal{T}$, and the sum over $w_J \in \Sigma^{|J|}$ is a sum over a singleton (the empty word), yielding:

$$\sum_{w_J \in \Sigma^0} \mathcal{T}(w_J, w) = \mathcal{T}(w),$$

and since $\mathcal{T}_J(w) = \mathcal{T}(w)$, the base case holds.

Inductive step: Assume the lemma holds for all subsets of size k, and let $J \subseteq [n]$ with $|J| = k + 1$. Pick any $j_0 \in J$, and define $J' = J \setminus \{j_0\}$, which has size k. By the inductive hypothesis, for any $w \in \Sigma^{[n]\setminus J}$, we have:

$$\sum_{w_{J'} \in \Sigma^k} \mathcal{T}(w_{J'}, w, w_{j_0}) = \mathcal{T}_{J'}(w, w_{j_0}) \quad ,$$

where $w_{j_0} \in \Sigma$ varies over its values.

Now consider the sum over all $w_J \in \Sigma^{k+1}$, which we write as:

$$\sum_{w_{J'} \in \Sigma^k} \sum_{w_{j_0} \in \Sigma} \mathcal{T}(w_{J'}, w_{j_0}, w) \quad .$$

By the inductive hypothesis, this equals:

$$\sum_{w_{j_0} \in \Sigma} \mathcal{T}_{J'}(w_{j_0}, w) \quad .$$

Observe that in $\mathcal{T}_{J'}$, the input tensor at position j_0 is still initialized to a one-hot vector, while the inputs at J' have been replaced with the all-ones vector.

Now, note that the inner sum over w_{j_0} is equivalent to replacing the input at j_0 with the all-ones vector, since the sum represents a sum over vector dot products of vector $\boldsymbol{v}_{w_{j_0}} := \mathcal{T}_{J'}(w_{j_0}, w)$ with one-hot encoded vectors. It follows from linearity that we can factor out $\boldsymbol{v}$, and the sum of the one-hot encoded vector yields the all-ones vector. Thus:

$$\sum_{w_{j_0} \in \Sigma} \mathcal{T}_{J'}(w_{j_0}, w) = \mathcal{T}_J(w) \quad .$$

Hence, by induction, the lemma holds for all subsets $J \subseteq [n]$.

□

Corollary 1.1. *Let $\mathcal{T}$ be a tensor network over Σ^n, and let $\mathcal{T}_{[n]}$ be the network where all input tensors are initialized to the all-ones vector. Then:*

$$\mathcal{T} \text{ is normalized} \iff \mathcal{T}_{[n]} = 1 \quad ,$$

i.e., the total contraction of the network with all-one input tensors equals 1.

Lemma 1.2. *Let $\mathcal{T}$ be a tensor network over Σ^n. The induced normalized tensor network $\frac{\mathcal{T}}{|\mathcal{T}|}$ is indeed normalized, and if additionally $\mathcal{T}$ is non-negative and $f \equiv id$, we have for all $w \in \Sigma^n$:*

$$S_{n,\mathcal{T}}(w) = S_{n,\frac{\mathcal{T}}{|\mathcal{T}|}}(w) \quad .$$

Proof. Let H be the set of non-input tensors in $\mathcal{T}$, and let $|H| = m$. In the induced normalized network, every tensor in H is scaled by a factor $\alpha = \frac{1}{\sqrt[m]{|\mathcal{T}|}}$. Since the final output $\mathcal{T}(w)$ is a multilinear contraction over the tensors, this means the scalar output for any $w \in \Sigma^n$ becomes:

$$\left(\prod_{v \in H} \alpha\right) \cdot \mathcal{T}(w) = \alpha^m \cdot \mathcal{T}(w) = \frac{1}{|\mathcal{T}|} \cdot \mathcal{T}(w) \quad .$$

Hence,

$$\left(\frac{\mathcal{T}}{|\mathcal{T}|}\right)(w) = \frac{\mathcal{T}(w)}{|\mathcal{T}|} \quad .$$

Summing over all $w \in \Sigma^n$,

$$\left|\frac{\mathcal{T}}{|\mathcal{T}|}\right| = \sum_{w \in \Sigma^n} \frac{\mathcal{T}(w)}{|\mathcal{T}|} = \frac{1}{|\mathcal{T}|} \sum_{w \in \Sigma^n} \mathcal{T}(w) = \frac{|\mathcal{T}|}{|\mathcal{T}|} = 1 \quad .$$

Moreover, since the normalization rescales all outputs by the same constant, the ratio of the terms to the total sum remains unchanged:

$$S_{n,\frac{\mathcal{T}}{|\mathcal{T}|}}(w) = \frac{\left(\frac{\mathcal{T}(w)}{|\mathcal{T}|}\right)}{\sum_{w' \in \Sigma^n} \left(\frac{\mathcal{T}(w')}{|\mathcal{T}|}\right)} = \frac{\mathcal{T}(w)}{|\mathcal{T}|} \cdot \frac{1}{1} = S_{n,\mathcal{T}}(w) \quad .$$

This completes the proof. □

One might ask whether our definition for tensor networks is bit restrictive, as it only allows for contraction over *pairs* of tensors. But what if we wanted to contract, say, three tensors at once over a common index?

Proposition 1.1. *Let $V' \subseteq V$ be a set of tensors in a tensor network, each containing an axis of dimension d labeled by a shared index i. Contracting all tensors in V' over the shared index i is equivalent to contracting each tensor individually with a single tensor*

$$\delta_{|V'|} : [d]^{|V'|} \mapsto \mathbb{R}_{\geq 0}$$

defined by

$$\delta_{|V'|}(i_1, \ldots, i_{|V'|}) = \begin{cases} 1 & \textit{if } i_1 = \cdots = i_{|V'|} \ , \\ 0 & \textit{otherwise.} \end{cases} \quad .$$

That is, a full contraction over a shared index can be implemented by introducing a single copy tensor connected to each tensor in V'.

Proof. Each tensor T_v for $v \in V'$ has an index $i \in [d]$ corresponding to the shared axis. The contraction over this index is defined by summing over the common value of i across all tensors:

$$\sum_{i=1}^{d} \prod_{v \in V'} T_v(\ldots, i, \ldots) \quad .$$

Now consider a new tensor $\delta_{|V'|}$ of order $|V'|$, defined as 1 if all indices are equal and 0 otherwise. Let each tensor T_v maintain its original indices, but connect to $\delta_{|V'|}$ via the position corresponding to v.

The contraction over this shared structure gives:

$$\sum_{i_1, \ldots, i_{|V'|}} \left(\prod_{v \in V'} T_v(\ldots, i_v, \ldots) \right) \delta_{|V'|}(i_1, \ldots, i_{|V'|}) \quad .$$

By definition of $\delta_{|V'|}$, this enforces $i_1 = \cdots = i_{|V'|}$, reducing the above to:

$$\sum_{i=1}^{d} \prod_{v \in V'} T_v(\ldots, i, \ldots) \quad ,$$

which is exactly the original contraction. Hence, the two constructions are equivalent. □

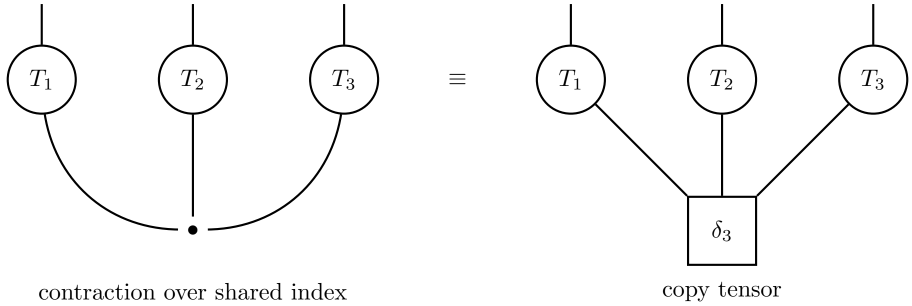


Figure 2: Contracting multiple tensors over one shared index is equivalent to contracting them individually with a single copy tensor.

1.1 Reshaping a Tensor by a Partition (Matricization)

A tensor network defines a probability tensor with $|\Sigma|^n$ entries. However, we can reshape this tensor by grouping together some axes. To this end, we first define a *partition*:

Definition 1.4. Let $I = \{1, 2, \ldots, n\}$ be a set of indices. A *partition* of I is a collection of non-empty, pairwise disjoint subsets $\{I_1, I_2, \ldots, I_k\}$ of I whose union is I. That is:

1. $I_j \neq \emptyset$ for all $j \in \{1, \ldots, k\}$
2. $I_j \cap I_l = \emptyset$ for all $j \neq l$
3. $\bigcup_{j=1}^{k} I_j = I$

The subsets I_j are called the *parts* of the partition.

In the context of a tensor with n axes, a partition of its indices allows for grouping these axes together, effectively reshaping the tensor. Partitions with two parts allows us to represent a tensor by a matrix. To this end, we use the notation $[[T]]_{A,B}$ to represent the reshaping of a tensor T into a matrix based on the partition $\{A, B\}$. This operation, also known as *matricization*, *unfolding*, or *flattening*, relies on partitioning the tensor's indices into two distinct sets, one for the rows and one for the columns of the new matrix. Formally:

Definition 1.5 (Tensor Matricization)**.** Let T be an order-k tensor with indices in the set $I = \{1, 2, \ldots, k\}$, and corresponding dimensions $d_1, d_2, \ldots, d_k$. Let $\{A, B\}$ be a bipartition of I, such that $A \cup B = I$ and $A \cap B = \emptyset$. Let $A = \{a_1, \ldots, a_{|A|}\}$ and $B = \{b_1, \ldots, b_{|B|}\}$ be ordered sets of indices.

The matricization of T with respect to the partition $\{A, B\}$ is the matrix $M =$ of size $D_A \times D_B$, where:

- The number of rows is $D_A = \prod_{j \in A} d_j$.
- The number of columns is $D_B = \prod_{j \in B} d_j$.

An element $T_{i_1, i_2, \ldots, i_k}$ of the tensor is mapped to an element $M_{\alpha,\beta}$ of the matrix, where the row index α and column index β are "super-indices" formed from the indices in A and B, respectively. Assuming a lexicographical ordering, the mapping is given by:

$$\alpha = 1 + \sum_{j=1}^{|A|} (i_{a_j} - 1) \left(\prod_{l=j+1}^{|A|} d_{a_l} \right)$$

$$\beta = 1 + \sum_{j=1}^{|B|} (i_{b_j} - 1) \left(\prod_{l=j+1}^{|B|} d_{b_l} \right)$$

1.1.1 An Upper Bound for the Rank of Matrices in Tensor Networks

We can analyze these matrices, which depend on the partition $\{A, B\}$, of our tensor network. When picturing it as a graph, we can separate the states in A from the states in B with a cut. Metaphorically, we see that A and B *communicate* over this cut. Hence, we might assume that if the associated bond dimensions are small, so is the mutual information between the two sets. In fact, this intuition is backed up by mathematical evidence:

Theorem 1.1. *Let $\mathcal{T}$ be a tensor represented by a tensor network over a set of physical indices $\mathcal{I} = \{i_1, \ldots, i_N\}$. For any bipartition of these indices into sets $\mathcal{I}_A$ and $\mathcal{I}_B$, which induces a cut through a set of internal bonds $E_{cut} =$*

$\{e_1, \ldots, e_c\}$ *with dimensions* $\{D_1, \ldots, D_c\}$*, the rank of the reshaped matrix* M *corresponding to this partition is bounded by:*

$$rank(M) \leq \prod_{k=1}^{c} D_k$$

Proof. A tensor network represents a tensor $\mathcal{T}$ by contracting a set of smaller tensors. Let's visualize a generic tensor network. A bipartition of the physical indices $(\mathcal{I}_A, \mathcal{I}_B)$ divides the graph of the network into two subgraphs.

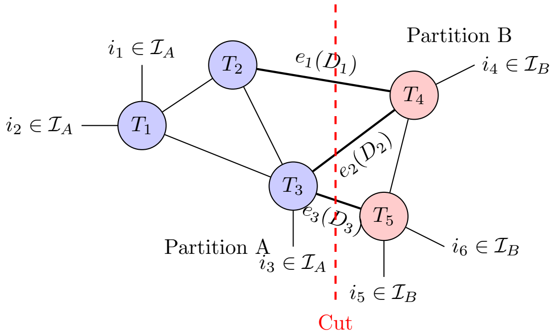
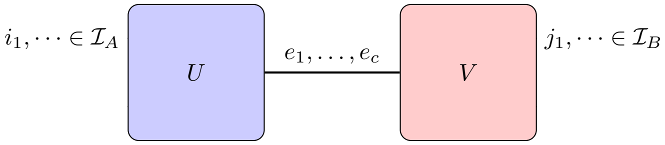


Figure 3: A generic tensor network with a cut partitioning the physical indices into $\mathcal{I}_A$ (blue) and $\mathcal{I}_B$ (red). The cut crosses bonds e_1, e_2, e_3.

We can group all tensors on the left of the cut (Partition A) and contract their internal bonds among themselves. This results in a single, large tensor, which we call U. The indices of U are the physical indices in $\mathcal{I}_A$ and the cut bond indices $\{e_1, \ldots, e_c\}$. Likewise, we contract all tensors in Partition B to form a tensor V, with indices from $\mathcal{I}_B$ and $\{e_1, \ldots, e_c\}$.



$i_1, \cdots \in \mathcal{I}_A$
U
$e_1, \ldots, e_c$
V
$j_1, \cdots \in \mathcal{I}_B$

Figure 4: Contracting the subgraphs results in two tensors, U and V, connected by the cut bonds.

The original tensor $\mathcal{T}$ is recovered by contracting U and V over the cut indices. Now, we reshape U and V into matrices. Let α be a multi-index that combines all cut indices $\{e_1, \ldots, e_c\}$. The dimension of this multi-index is $D_{cut} = \prod_{k=1}^{c} D_k$. Let I_A be the multi-index for all physical indices in $\mathcal{I}_A$. Let I_B be the multi-index for all physical indices in $\mathcal{I}_B$.

We reshape the tensor U into a matrix $\mathbf{U}$ of size $(\dim(I_A) \times D_{cut})$ where the rows are indexed by I_A and the columns by α. We reshape the tensor V into a matrix $\mathbf{V}$ of size $(D_{cut} \times \dim(I_B))$ where the rows are indexed by α and the columns by I_B.

The reshaped tensor matrix M (with rows I_A and columns I_B) is exactly the matrix product of $\mathbf{U}$ and $\mathbf{V}$:

$$M = \mathbf{U} \cdot \mathbf{V}$$

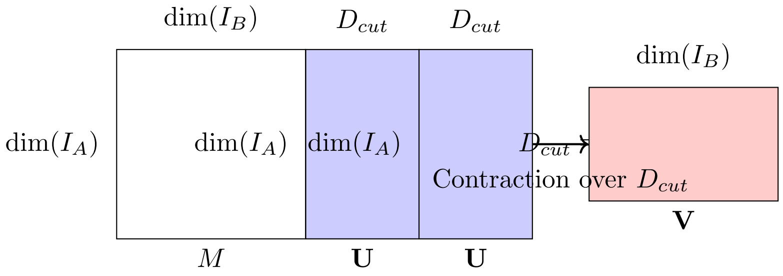


Figure 5: The matrix M is a product of matrices $\mathbf{U}$ and $\mathbf{V}$, with inner dimension D_{cut}.

A fundamental property of linear algebra states that the rank of a product of matrices is at most the minimum of their individual ranks:

$$\text{rank}(M) = \text{rank}(\mathbf{U} \cdot \mathbf{V}) \leq \min(\text{rank}(\mathbf{U}), \text{rank}(\mathbf{V}))$$

The rank of a matrix is also bounded by its dimensions. For $\mathbf{U}$ and $\mathbf{V}$, we have:

$$\text{rank}(\mathbf{U}) \leq \min(\dim(I_A), D_{cut}) \leq D_{cut}$$

$$\text{rank}(\mathbf{V}) \leq \min(D_{cut}, \dim(I_B)) \leq D_{cut}$$

Therefore, the rank of M is bounded by the total dimension of the cut:

$$\text{rank}(M) \leq D_{cut} = \prod_{k=1}^{c} D_k$$

This completes the proof. □