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 $D_v = \{d_1, \ldots, d_k\}$ as a function

$$T_v : [d_1] \times \cdots \times [d_k] \to \mathbb{R}_{\geq 0}.$$

As a shorthand, we write

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

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 $D_v = \{d_1, \ldots, d_k\}$. Let $V_{\text{layer}} \subseteq V$ denote the set of all vertices at a given layer.
- The input set $I = (T_{0,1}, \ldots, T_{0,n}) \subset V$ 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:

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

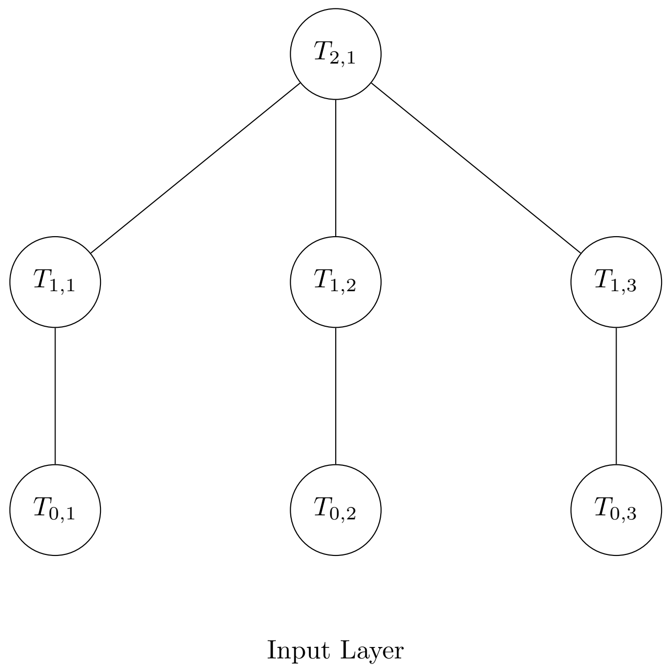

Figure 1: A basic tensor network over Σ^3.

Definition 1.2 (Normalization of 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 $|\mathcal{T}| = 1$.

Let $H \coloneqq V \setminus I$ 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. The tensor network $\mathcal{T}(w)$ evaluates to a scalar obtained by contracting the network, where each input tensor $T_{0,j}$ is initialized with a one-hot vector corresponding to the symbol $w_j \in \Sigma$.

For $j \in J$, replacing the one-hot vector by the all-ones vector is equivalent to summing over all possible $w_j \in \Sigma$. That is, for fixed $w \in \Sigma^{[n]\setminus J}$,

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

since the multilinearity of the network ensures that the contraction distributes over summation in each input.

Formally, each contraction involving an input tensor $T_{0,j}$ with the one-hot vector δ_{w_j} is replaced by a sum over δ_{w_j} for all $w_j \in \Sigma$, i.e., the all-ones vector $\mathbf{1}$. The result of the total contraction is thus the sum over all $w_J \in \Sigma^{|J|}$ of $\mathcal{T}(w_J, w)$, as required. □

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 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 $\alpha = \dfrac{1}{\sqrt[m]{|\mathcal{T}|}}$. 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 softmax 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

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

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. $\square$

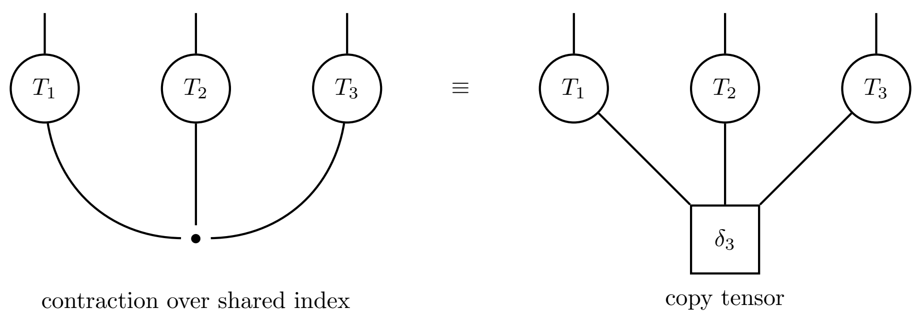

Figure 2: Contracting multiple tensors over one shared index is equivalent to contracting them individually with a single copy tensor.

1.1 Bulk Marginal Property

We are interested in tensor networks that have the bulk marginal property. When further specifying our network structure, we might have a model space for varying word lengths n, but not for every $n \in \mathbb{N}$. Take for example the model space of binary trees as shown in figure 3.

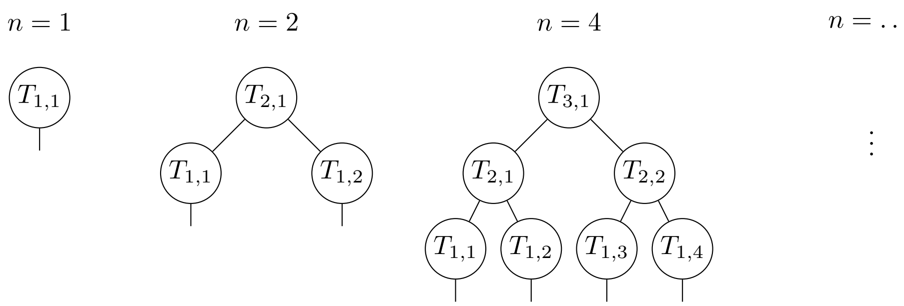

Figure 3: A model space for sequences of length $n = 2^k$.

In definition **??** we saw how to construct a model with the desired bulk marginal property based on the base model. However, we might not always have a base model for ever $n \in \mathbb{N}$ like discussed. Luckily, it turns out that this is not an issue, as there are many ways we can build a new model with the bulk marginal property from a base model even if it is only defined on a subset of $\mathbb{N}$. Without a proof, we might do the same procedure as in definition **??** but with bigger steps (instead of taking always the consecutive model), and induce the in-between models by marginalizing the bigger ones.

Alternatively, if we wanted a model with bulk marginal property that itself is also an element of our specified model space, we might ask ourselves, how we can construct a bigger tensor network while preserving the distribution in its leading random variables.