# The Tensor Cookbook

Thomas Dybdahl Ahle

August 28, 2024

# Chapter 1

# Introduction

**What is this?** These pages are a guide to tensors, using the visual language of "tensor diagrams". For illustrating the generality of the approach, I've tried to closely follow the legendary "Matrix Cookbook". As such, most of the presentation is a collection of facts (identities, approximations, inequalities, relations, ...) about tensors and matters relating to them. You won't find many results not in the original cookbook, but hopefully the diagrams will give you a new way to understand and appreciate them.

**It's ongoing:** The Matrix Cookbook is a long book, and not all the sections are equally amendable to diagrams. Hence I've opted to skip certain sections and shorten others. Perhaps in the future, I, or others, will expand the coverage further.

For example, while we cover all of the results on Expectation of Linear Combinations and Gaussian moments, we skip the section on general multi-variate distributions. I have also had to rearrange the material a bit, to avoid having to introduce all the notation up front.

**Complex Matrices and Covariance** Tensor diagrams (or networks) are currently most often seen in Quantum Physics. Here most values are complex numbers, which introduce some extra complexity. In particular transposing a matrix now involves taking the conjugate (flipping the sign of the imaginary part), which introduces the need for co- and contra-variant tensors. None of this complexity is present with standard real valued matrices, as is common e.g. in Machine Learning applications. For simplicity I have decided to not include these complexities.

**Tensorgrad** The symbolic nature of tensor diagrams make the well suited for symbolic computation.

**Advantages of Tensor Diagram Notation:** Tensor diagram notation has many benefits compared to other notations:

Various operations, such as a trace, tensor product, or tensor contraction can be expressed simply without extra notation. Names of indices and tensors can often be omitted. This saves time and lightens the notation, and is especially useful for internal indices which exist mainly to be summed over. The order of the tensor resulting from a complicated network of contractions can be determined by inspection: it is just the number of unpaired lines. For example, a tensor network with all lines joined, no matter how complicated, must result in a scalar.

# Contents

## 1.1 Tensor Diagrams

Tensor diagrams are simple graphs (or "networks") where nodes represent variables (e.g. vectors or matrices) and edges represent contractions (e.g. matrix multiplication or inner products.) The follow table shows how some basic operations can be written with tensor diagrams:

| | | | | |
|---|---|---|---|---|
| Dot product | $a$—$b$ | $y = \sum_i a_i b_i$ | $[\cdot\ \cdot\ \cdot\ \cdot]\begin{bmatrix}\cdot\\\cdot\\\cdot\\\cdot\end{bmatrix}$ | $= y$ |
| Outer product | —$a$ $b$— | $Y_{i,j} = a_i b_j$ | $\begin{bmatrix}\cdot\\\cdot\\\cdot\\\cdot\end{bmatrix}[\cdot\ \cdot\ \cdot\ \cdot]$ | $=$—$Y$— |
| Matrix-Vector | —$A$–$b$ | $y_i = \sum_j A_{i,j} b_j$ | $\begin{bmatrix}\cdot&\cdot&\cdot&\cdot\\\cdot&\cdot&\cdot&\cdot\\\cdot&\cdot&\cdot&\cdot\\\cdot&\cdot&\cdot&\cdot\end{bmatrix}\begin{bmatrix}\cdot\\\cdot\\\cdot\\\cdot\end{bmatrix}$ | $=$—$y$ |
| Matrix-Matrix | —$A$–$B$— | $Y_{i,k} = \sum_j A_{i,j} B_{j,k}$ | $\begin{bmatrix}\cdot&\cdot&\cdot&\cdot\\\cdot&\cdot&\cdot&\cdot\\\cdot&\cdot&\cdot&\cdot\\\cdot&\cdot&\cdot&\cdot\end{bmatrix}\begin{bmatrix}\cdot&\cdot&\cdot&\cdot\\\cdot&\cdot&\cdot&\cdot\\\cdot&\cdot&\cdot&\cdot\\\cdot&\cdot&\cdot&\cdot\end{bmatrix}$ | $=$—$Y$— |

We think of vectors and matrices as tensors of order 1 and 2. The order corresponds to the number of dimensions in their $[\cdots]$ visualization above, e.g. a vector is a 1-dimensional list of numbers, while a matrix is a 2-dimensional grid of numbers. The order also determines the degree of the node representing the variable in the tensor graph.

Diagram notation becomes more interesting when you have tensors of order 3 and higher. An order 3 tensor is a cube or numbers, or stack of matrices. E.g. we can write this as $T \in \mathbb{R}^{n\times m\times k}$, so $T_i \in \mathbb{R}^{m\times k}$ is a matrix for $i = 1 \dots n$. Of course we could slice $T$ along the other axes too, so $T_{:,j} \in \mathbb{R}^{n\times k}$ and $T_{:,:,\ell} \in \mathbb{R}^{n\times m}$ are matrices too.

A matrix having two outgoing edges means there are two ways you can multiply a vector onto it, either on the left: $x^T M$, or on the right: $Mx$. In graph notation we just write $x$–$M$– and –$M$–$x$. An order 3 tensor has three edges, so we can multiply it with a vector in three ways:

$$\begin{array}{c} | \\ T \\ \diagup \quad \diagdown \\ \qquad x \end{array} \quad \text{and} \quad \begin{array}{c} | \\ T \\ \diagup \quad \diagdown \\ x \qquad \end{array} \quad \text{and} \quad \begin{array}{c} x \\ | \\ T \\ \diagup \quad \diagdown \end{array}$$

To be perfectly precise about what each one means, we should give the edges labels. For example we would write $\begin{array}{c} | \\ T \\ \diagup \quad \diagdown_{i} \\ \qquad x \end{array}$ to specify the matrix $\sum_i T_i x_i$. However, often the edge in question will be clear from the context, which is part of what makes tensor diagram notation cleaner than, say, Einstein sum notation.

The *key principle* of tensor diagrams is that *edge contraction is associative.* This means you can contract any edge in any order you prefer. To see this, it suffices to note that an

### 1.1.1 Trace

The "trace" of a square matrix is defined $\mathrm{Tr}(A) = \sum_i A_{i,i}$. In tensor diagram notation, that corresponds to a self-edge: $\overset{\frown}{A}$ .

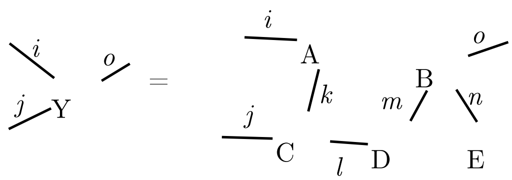


Figure 1.1: contract spiders

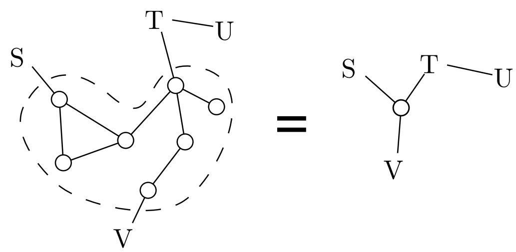


Figure 1.2: contract spiders

$$\sum_{i=1}^{n} A_{ii} = \mathrm{Tr}(A) = \mathrm{Tr}(AI) \qquad A = A\text{—}\circ \tag{11}$$

$$\mathrm{Tr}(A) = \sum_i \lambda_i = \langle 1, \lambda \rangle \qquad A = Q\text{—}\underset{\lambda}{\circ}\text{—}Q^{-1} \tag{12}$$

$$\mathrm{Tr}(A) = \mathrm{Tr}(A^T) \qquad A = A \tag{13}$$

$$\mathrm{Tr}(AB) = \mathrm{Tr}(BA) \qquad A\text{—}B = B\text{—}A \tag{14}$$

$$\mathrm{Tr}(A+B) = \mathrm{Tr}(A) + \mathrm{Tr}(B) \qquad (A+B)\text{—}\circ = A\text{—}\circ + B\text{—}\circ \tag{15}$$

$$\begin{aligned} \mathrm{Tr}(ABC) &= \mathrm{Tr}(BCA) & A\text{—}B\text{—}C &= B\text{—}C\text{—}A \\ &= \mathrm{Tr}(CAB) & &= C\text{—}A\text{—}B \end{aligned} \tag{16}$$

$$\begin{aligned} a^T a = \mathrm{Tr}(a a^T) \qquad a\text{—}a &= \mathrm{Tr}(\text{—}a \quad a\text{—}) \\ &= a \quad a \end{aligned} \tag{17}$$

### 1.1.2 The Copy Tensor

We define $\circ\!-$ to be the all-ones vector. That is $\circ_i = 1$. We generalize $\circ$ to rank-n tensors by $\circ_{i,j,k,\ldots} = [i = j = k = \ldots]$. That is, the tensor with 1 on the diagonal, and 0 everywhere else. This is also known as the "copy" or "spider" tensor, or "generalized Kronecker delta".

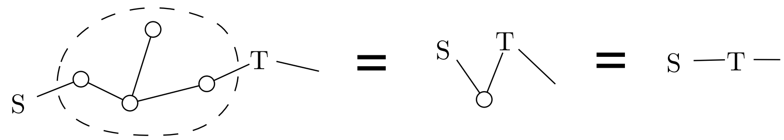


Figure 1.3: Eliminate Identity

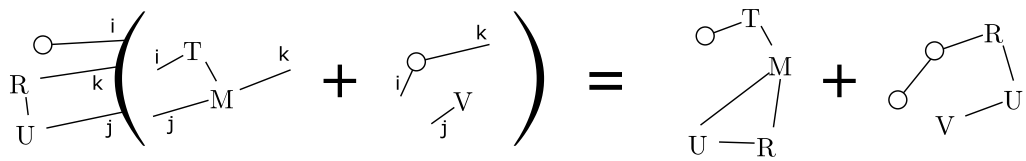


Figure 1.4: Distributive law

For rank-2 tensors, $-\circ- = I$, the identity matrix.

The rank-4 copy tensor, $C_{i,j,k,l} = [i = j = k = l]$, $\times\!\!\circ$, differs from the outer product of two identity matrices (in the Cookbook denoted $J$), which satisfies $J_{i,j,k,l} = [i = k][j = l]$ and which we'd write as $J = \substack{-\circ-\\-\circ-}$, and satisfies, for example, $\frac{dX}{dX} = J$.

### 1.1.3 Eigen values

$-A- = -Q-\underset{\lambda}{\circ}-Q^{-1}-$ where $\lambda_i$ is the $i$th eigenvalue of $A$.

More general principles:

1. Contractions are associative: You can contract edges in any order.
2. distributive law of sums/products.
3. Sums and broadcasting.
4. You can always contract connected subgraphs of Copy tensors.