The Tensor Cookbook

Thomas Dybdahl Ahle

May 12, 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 Notation and Nomenclature

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	$—A—B—$	$Y_{i,k} = \sum_j A_{i,j} B_{j,k}$	$\begin{bmatrix}\cdot&\cdot&\cdot&\cdot\\\cdot&\cdot&\cdot&\cdot\\\cdot&\cdot&\cdot&\cdot\\\cdot&\cdot&\cdot&\cdot\end{bmatrix}\begin{bmatrix}\cdot&\cdot&\cdot&\cdot\\\cdot&\cdot&\cdot&\cdot\\\cdot&\cdot&\cdot&\cdot\\\cdot&\cdot&\cdot&\cdot\end{bmatrix}$	$= —Y—$

The rank-4 copy tensor, $C_{i,j,k,l} = [i = j = k = l]$, $\times$, 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 $J =$ —○—, and satisfies, for example, $\frac{dX}{dX} = J$.

1.2 Basics

$—A— = —Q—\underset{\lambda}{\circ}—Q^{-1}—$ where λ_i is the ith eigenvalue of A.

More general principles:

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

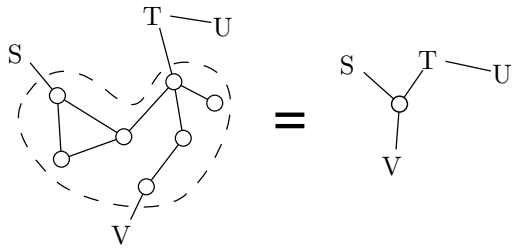


Figure 1.1: Contract spiders

1.2.1 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: 

.

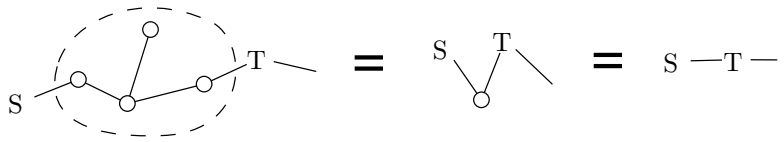


Figure 1.2: Eliminate Identity

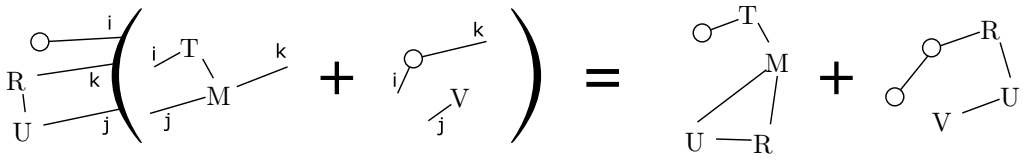


Figure 1.3: Distributive law

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

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

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

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

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

$$\mathrm{Tr}(ABC) = \mathrm{Tr}(BCA) = \mathrm{Tr}(CAB) \qquad \overbrace{A—B—C} = \overbrace{B—C—A} = \overbrace{C—A—B} \tag{16}$$

$$a^T a = \mathrm{Tr}(aa^T) \qquad a—a = \mathrm{Tr}(—a \quad a—) = \overbrace{a \quad a} \tag{17}$$

Chapter 2

Simple Derivatives

2.1 Derivatives of Matrices, Vectors and Scalar Forms

2.1.1 First Order

The Matrix Cookbook defines the single-entry matrix $J^{i,j} \in R^{n \times n}$ as the matrix which is zero everywhere except in the entry (i, j) in which it is 1. Alternatively we could write $J^{i,j}_{n,m} = [i = n][j = m]$.

$$\frac{\partial x^T a}{\partial x} = \frac{\partial a^T x}{\partial x} = a \qquad (x — a) = (x) — a = \diagup — a \tag{69}$$

$$\frac{\partial a^T X b}{\partial X} = a b^T \qquad (a — X — b) = a — (X) — b = a — \diagup — b \tag{70}$$

$$\frac{\partial X}{\partial X_{i,j}} = J^{i,j} \qquad (— X —)^{\text{i}}_{\text{j}} = —\,{}^{\text{i}}\!\diagup\!{}_{\text{j}}\,— \tag{73}$$

$$\frac{\partial (XA)_{i,j}}{\partial X_{m,n}} = (J^{m,n} A)_{i,j} \qquad ({}^{\text{i}}— X — A —{}_{\text{j}})^{\text{m}}_{\text{n}} = — (X) — A — = {}^{\text{i}}\!\diagup^{\text{m}}\!{}_{\text{n}} — A —{}_{\text{j}} \tag{74}$$

2.1.2 Second Order

$$\frac{\partial}{\partial X_{i,j}} \sum_{k,l,m,n} X_{k,l} X_{m,n} = (\sum_{k,l} X_{k,l})^2 \qquad \left(\begin{smallmatrix} \circ\!-\!X\!-\!\circ \\ \circ\!-\!X\!-\!\circ \end{smallmatrix}\right)^{\substack{\mathrm{i}\\ \diagup\\ \mathrm{j}}} = \begin{smallmatrix} \circ\!-\!X\!-\!\circ \\ \circ\!-\!(X)\!-\!\circ \end{smallmatrix} + \begin{smallmatrix} \circ\!-\!(X)\!-\!\circ \\ \circ\!-\!X\!-\!\circ \end{smallmatrix} \tag{76}$$

$$= 2 \sum_{k,l} X_{k,l} \qquad = 2 \begin{smallmatrix} \circ\!-\!X\!-\!\circ \\ \circ\!-\!\diagup\!-\!\circ \end{smallmatrix}^{\mathrm{i}}_{\mathrm{j}}$$

$$\frac{\partial b^T X^T X c}{\partial X} = X(bc^T + cb^T) \qquad (b\!-\!X^T\!-\!X\!-\!c)^{\diagup} = b\!-\!X^T\!-\!(X)^{\diagup}\!-\!c + b\!-\!(X^T)^{\diagup}\!-\!X\!-\!c \tag{77}$$

$$= b\!-\!X^T\!-\!\diagup\!-\!c + b\!-\!\diagup\!-\!X\!-\!c = -X\!-\!(-b\,c' + -c\,b')$$

$$\frac{\partial}{\partial x}(Bx+b)^T C(Dx+d) = B^T C(Dx+d) + D^T C^T (Bx+b) \tag{78}$$

$$\frac{\partial}{\partial X_{i,j}}(X^T B X)_{k,l} = \delta_{l,j}(X^T B)_{k,i} + \delta_{k,j}(BX)_{i,l} \qquad (^{\mathrm{k}}\!-\!X^T\!-\!B\!-\!X\!-^{\mathrm{l}})^{\diagup} = {}^{\mathrm{k}}\!-\!X^T\!-\!B\!-\!\diagup^{\mathrm{l}} + {}^{\mathrm{k}}\!-\!\diagup\!-B\!-\!X\!-^{\mathrm{l}} \tag{79}$$

$$\frac{\partial}{\partial X_{i,j}} X^T B X = X^T B J^{i,j} + J^{j,i} B X \qquad \text{(same as above)} \tag{80}$$

$$\frac{\partial}{\partial x} x^T B x = (B + B^T)x \qquad (x\!-\!B\!-\!x)^{\diagup} = -B\!-\!x + x\!-\!B\!- = x\!-\!\left(\begin{smallmatrix} +\!-B\!- \\ -\!-B\!- \end{smallmatrix}\right) \tag{81}$$

$$\frac{\partial}{\partial X} b^T X^T D X c = D^T X b c^T + D X c b^T \qquad \ldots \tag{82}$$

$$\frac{\partial}{\partial X}(Xb+c)^T D(Xb+c) = (D + D^T)(Xb+c)b^T \qquad \ldots \tag{83}$$

Assume W is symmetric, then... (84) - (88)

2.1.3 Higher-order and non-linear

...

2.1.4 Gradient and Hessian

...

2.2 Derivatives of Traces

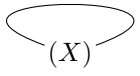
Assume F(X) to be a differentiable function of each of the elements of X. It then holds that

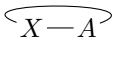
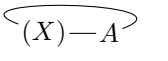
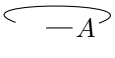
$$\frac{d\mathrm{Tr}(F(x))}{dX} = f(X)^T,$$

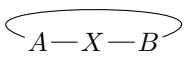
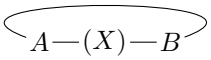
where $f(\cdot)$ is the scalar derivative of $F(\cdot)$.

TODO: To show this with tensor diagrams, we first need to introduce our notation for functions.

2.2.1 First Order

(99) $\frac{\partial}{\partial X}\mathrm{Tr}(X) = I$ () =  =  = $-\circ-$

(100) $\frac{\partial}{\partial X}\mathrm{Tr}(XA) = A^T$ () =  =  = $-A^T-$

(101) $\frac{\partial}{\partial X}\mathrm{Tr}(AXB) = A^T B^T$ () =  = $-A^T-B^T-$

Continues for (102-105). The last one uses the Kronecker product, which we may have to introduce first.

2.2.2 Second Order

2.2.3 Higher Order

2.2.4 Other

Chapter 3

Statistics and Probability

3.0.1 Definition of Moments

Let $x \in \mathbb{R}^n$ is a random variable. We write $m = E[x] \in \mathbb{R}^n$ for the expectation and $M = \mathrm{Var}[x] = E[(x-m)(x-m)^T]$ for the covariance (when these quantities are defined.)

In tensor diagrams, we will use square brackets:

$$m = [—x] \quad \text{and} \quad M = [—(x \ominus m) \;\; (x \div m)—]$$

Note we used the German minus, $\div$, to distinguish subtraction from contraction edges.

We can also define the third and fourth centralized moment tensors

$$M_3 = \begin{bmatrix} (x \div m)— \\ (x \div m)— \\ (x \div m)— \end{bmatrix} \quad \text{and} \quad M_4 = \begin{bmatrix} (x \div m)— \\ (x \div m)— \\ (x \div m)— \\ (x \div m)— \end{bmatrix}.$$

3.0.2 Expectation of Linear Combinations

General principle: The "linearity of expectation" lets you pull out all parts of the graph not involving X.

Linear Forms

$$\mathrm{E}[AXB + C] = A\mathrm{E}[X]B + C \qquad \begin{bmatrix} -A-X-B- \\ + -C- \end{bmatrix} = \begin{matrix} -A-[X]-B- \\ + -C- \end{matrix} \tag{312}$$

$$\mathrm{Var}[Ax] = A\mathrm{Var}[x]A^T \qquad \begin{bmatrix} A-x \div [A-x] \\ A-x \div [A-x] \end{bmatrix} = \begin{bmatrix} A-(x \div m) \\ A-(x \div m) \end{bmatrix} \tag{313}$$

$$= \begin{matrix} A-\Big[(x \div m) \\ A-\Big[(x \div m)\Big] \end{matrix}$$

$$= -A-M_2-A-$$

Quadratic Forms

$$\begin{aligned}
\mathrm{E}[x^T A x] &= \mathrm{Tr}(A\Sigma) + \mu^T A \mu \\
[x - A - x] &= [(x \div \mu) - A - (x \div \mu) + \mu - A - \mu] \\
&= \begin{bmatrix} (x \div m) \\ (x \div m) \end{bmatrix} \geq A \quad + \mu - A - \mu \\
&= \overbrace{\Sigma — A} \quad + \mu - A - \mu
\end{aligned}$$

Cubic Forms

3.0.3 Weighted Scalar Variable

Let $y = w^T x$, and let $m = E[y]$, then

$$\begin{aligned}
\mathrm{E}[y] &= m = w^T \mu \\
\mathrm{E}[(y-m)^2] &= w - M_2 - w \\
\mathrm{E}[(y-m)^3] &= w \overset{w}{\text{-}} M_3 \text{-} w \\
\mathrm{E}[(y-m)^4] &= w \overset{w}{\text{-}} M_4 \text{-} w \\
&\quad\ \ w
\end{aligned}$$

For specific distributions, like x Gaussian, we can often reduce the moment tensors further. Khintchine's inequality also gives a way to bound all of these in terms of $E[(y-m)^2]$.

3.0.4 Gaussian Moments

Mean and covariance of linear forms

Mean and variance of square forms

Cubic forms

Mean of Quartic Forms

Gaussian Integration by Parts

General principle for Gaussian expectations.

Chapter 4

Kronecker and Vec Operator

4.1 Flattening

Flattening is a common operation for programmers. In the language of numpy, we may write `np.ones((2,3,4)).reshape(2, 12)` to flatten a shape (2,3,4) tensor into a shape (2,12) matrix. Similarly, in mathematical notation, $\mathrm{vec}(X)$ is commonly used to denote the flattening of a matrix into a vector.

Typically the main reason to do this is as a cludge for dealing with bad general notation for tensors. Hence, with tensor diagrams, we can avoid this operation entirely. However, it is still interesting to see how tensor diagrams can make a lot of properties of flattening much more transparent.

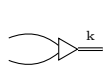
To begin with we note that flattening is a linear operation, and hence can be represented as a simple tensor. We'll use a triangle to denote this:

$$\triangleright_{i,j,k} = \quad = [i + jn = k].$$

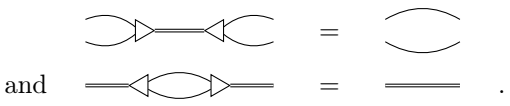
Here n is the dimension of the i edge. Note we use a double line to denote the output of the flattening operation. This is simply a syntactic choice to remind ourselves that the output is a bundle of two edges.

Using this notation we can write

$$\mathrm{vec}(X)_k = \sum_{i,j} \triangleright_{i,j,k} X_{i,j} = \quad .$$



The basic property of $\triangleright$ is that opposing triangles cancel:



.

4.2 The Kronecker Product

The Kronecker product of an $m \times n$ matrix A and an $r \times q$ matrix B, is an $mr \times nq$ matrix, $A \otimes B$ defined as

$$A \otimes B = \begin{bmatrix} A_{1,1}B & A_{1,2}B & \cdots & A_{1,n}B \\ A_{2,1}B & A_{2,2}B & \cdots & A_{2,n}B \\ \vdots & \vdots & \ddots & \vdots \\ A_{m,1}B & A_{m,2}B & \cdots & A_{m,n}B \end{bmatrix}.$$

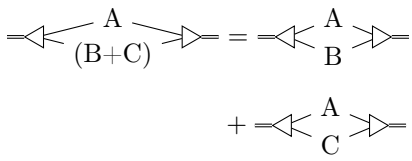
Using index notation we can also write this as $(A \otimes B)_{p(r-1)+v,q(s-1)+w} = A_{rs}B_{vw}$, but it's pretty hard to read.

In tensor notation the Kronecker Product is simply the outer product of two matrices, flattened "on both sides": $A \otimes B =$

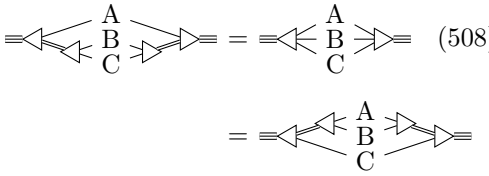
.

The Kronecker product has the following properties:

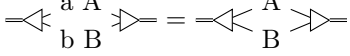
$$A \otimes (B + C) = A \otimes B + A \otimes C \tag{506}$$



$$A \otimes (B \otimes C) = (A \otimes B) \otimes C \tag{508}$$



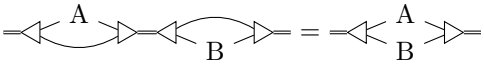
$$aA \otimes bB = ab(A \otimes B) \tag{509}$$



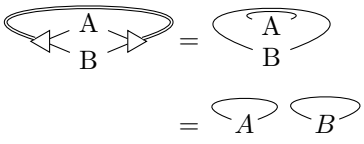
$$(A \otimes B)^T = A^T \otimes B^T \qquad \ldots \tag{510}$$

$$(A \otimes B)(C \otimes D) = AC \otimes BD \tag{511}$$

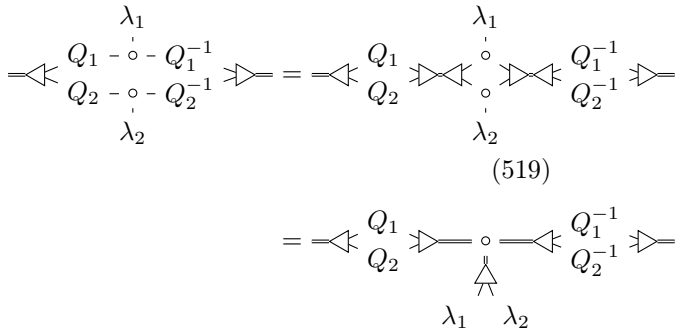
$$(A \otimes I)(I \otimes B) = A \otimes B \tag{511b}$$



$$\mathrm{Tr}(A \otimes B) = \mathrm{Tr}(A)\mathrm{Tr}(B) \tag{515}$$

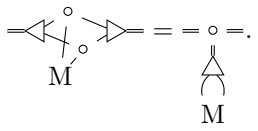


$$\text{eig}(A \otimes B) = \text{eig}(A)\text{eig}(B)$$



(519)

Here the last equation shows an interesting general property of $\triangleright$:



While the right side is just $\text{diag}(\text{vec}(M))$, the left side is harder to write in classical notation. We can write it with index notation as

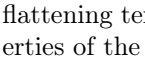
$$v_{k,\ell} = \sum_{i,j} \triangleright_{i,j,k} \triangleright_{i,j,\ell} M_{i,j} = \sum_{i,j} [i + jn = k][i + jn = \ell] M_{i,j},$$

but it is hardly enlightening.

4.3 The Vec Operator

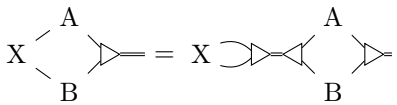
The vec-operator applied on a matrix A stacks the columns into a vector, i.e. for a 2×2 matrix

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \quad \text{vec}(A) = \begin{bmatrix} A_{11} \\ A_{21} \\ A_{12} \\ A_{22} \end{bmatrix}$$

At the start of the chapter we showed how to represent the vec-operator using the flattening tensor: $\text{vec}(X) =$ 

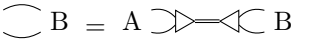
. The Matrix Cookbook gives the following properties of the vec-operator:

$$\text{vec}(A^T X B) = \text{vec}(X)^T (A \otimes B)$$



(520)

$$\text{Tr}(A^T B) = \text{vec}(A)^T \text{vec}(B)$$

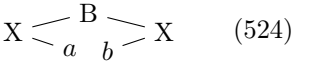


(521)

$$\text{vec}(A + B) = \text{vec}(A) + \text{vec}(B)$$

(522)

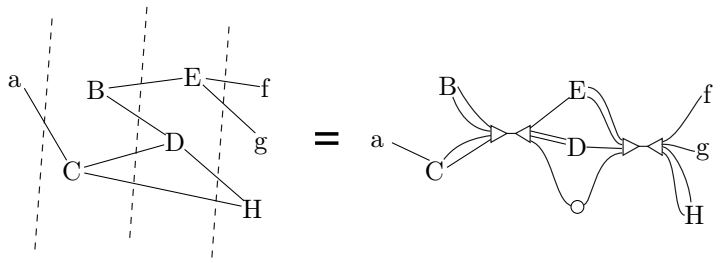
$$\mathrm{vec}(aA) = a\,\mathrm{vec}(A) \qquad (523)$$



$$a^T X B X^T c = \mathrm{vec}(X)^T (B \otimes c a^T) \mathrm{vec}(X) \qquad (524)$$

4.4 General Matrification

The last equation is an example of a general idea: Any tensor network can be transformed into a series of matrix multiplications by applying the vec-operator to all tensors and the flattening tensor to all edges. For example, the following complicated graph:



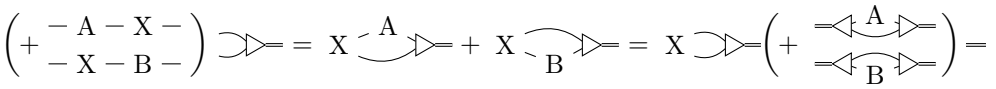
Can be written as a simple vector-matrix-matrix-vector product, aM_1M_2b, where $M_1 = \mathrm{vec}(B) \otimes C'$, $M_2 = E' \otimes D' \otimes I$ and $b = f \otimes g \otimes \mathrm{vec}(H)$, where C', D' and E' are rank 3 tensors flattened on one side, and $\mathrm{vec}(B)$ is interpreted as a matrix with a single column.

4.4.1 The Lyapunov Equation

A nice application of Kronecker product rewritings is to solve equations like

$$AX + XB = C. \qquad (272)$$

We use the rewriting $\mathrm{vec}(AX + XB) = (I \otimes A + B^T \otimes I)\mathrm{vec}(X)$, which follows from the tensor diagram massaging:



after which we can take the normal matrix inverse to get

$$\mathrm{vec}(X) = (I \otimes A + B^T \otimes I)^{-1}\mathrm{vec}(C). \qquad (273)$$

4.4.2 Encapsulating Sum

This is a generalization of the previous equation.

$$\sum_n A_n X B_n = C \qquad (274)$$

$$\mathrm{vec}(X) = (\sum_n B_n^T \otimes A_n)^{-1}\mathrm{vec}(C) \qquad (275)$$

4.5 The Hadamard Product

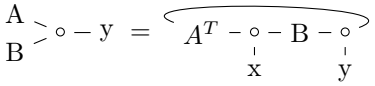
The Hadamard product, also known as element-wise multiplication, is not described in the Matrix Cookbook. Yet, it is a very useful operation, and has some interesting properties in connection with the Kronecker product.

We define the Hadamard product of two 2×2 matrices A and B as

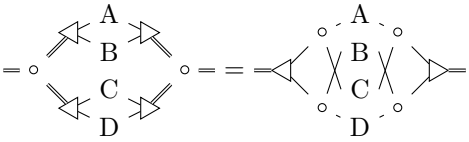
$$A \circ B = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \circ \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} = \begin{bmatrix} A_{11}B_{11} & A_{12}B_{12} \\ A_{21}B_{21} & A_{22}B_{22} \end{bmatrix}.$$

In tensor notation, the Hadamard product can be represented using two rank-3 copy tensors: $- \circ \overset{\text{A}}{\underset{\text{B}}{<}} \circ -$. Some properties of the Hadamard product are:

$$x^T(A \circ B)y = \mathrm{tr}(A^T D_x B D_y)$$



$$(A \otimes B) \circ (C \otimes D) = (A \circ C) \otimes (B \circ D)$$



To see why the last equation holds, it suffices to follow the double lines to see that A and C both use the "upper" part of the double edge, while B and D use the lower part.

4.6 Khatri–Rao product

Also known as the column-wise Kronecker, row-wise Kronecker or "Face-splitting Product". We use the symbols $*$ and $\bullet$ for the column and row-wise Kronecker products, respectively.

$$A * B = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} * \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} = \begin{bmatrix} A_{11}B_{11} & A_{12}B_{12} \\ A_{11}B_{21} & A_{22}B_{22} \\ A_{21}B_{11} & A_{22}B_{12} \\ A_{21}B_{21} & A_{22}B_{22} \end{bmatrix}$$

$$A \bullet B = \ldots = \begin{bmatrix} A_{11}B_{11} & A_{11}B_{12} & A_{12}B_{11} & A_{12}B_{12} \\ A_{21}B_{21} & A_{21}B_{22} & A_{22}B_{21} & A_{22}B_{22} \end{bmatrix}$$

In terms of tensor diagrams, these products correspond simply to flattening the product on one side, and using a copy tensor on the other:

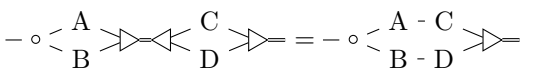
$$A * B =$$

$$A \bullet B = -$$

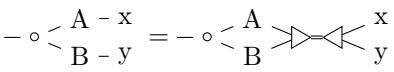
Clearly the two are identical up to transpose. Indeed, $(A*B)^T = B^T \bullet A^T$ and $(A\bullet B)^T = B^T * A^T$.

There are multiple "mixed product" identities:

$$(A \bullet B)(C \otimes D) = (AC) \bullet (BD)$$



$$(Ax) \circ (By) = (A \bullet C)(x \otimes y)$$



Chapter 5

Determinant and Inverses

5.1 Determinant

It's convenient to write the determinant in tensor notation as

$$\det(A) = \frac{1}{n!} \overline{\underline{A \quad \cdots \quad A}}$$

where $\underline{i_1 \quad i_2 \quad \cdots \quad i_n} = \varepsilon_{i_1,\ldots,i_n}$ is the rank-n Levi-Civita tensor defined by

$$\varepsilon_{i_1,\ldots,i_n} = \begin{cases} \operatorname{sign}(\sigma) & \sigma = (i_1,\ldots,i_n) \text{ is a permutation} \\ 0 & \text{otherwise.} \end{cases}$$

To see that the definition makes sense, let's first consider

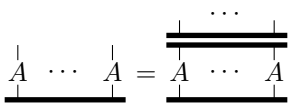
$$\det(I) = \frac{1}{n!} \underline{\square \quad \cdots} = \frac{1}{n!} \sum_{i_1,\ldots,i_n,j_1,\ldots,j_n} \varepsilon_{i_1,\ldots,i_n}\varepsilon_{j_1,\ldots,j_n}[i=j] = \frac{1}{n!} \sum_{i_1,\ldots,i_n} \varepsilon^2_{i_1,\ldots,i_n} = 1.$$

In general we get from the permutation definition of the determinant:

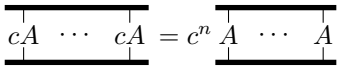
$$\begin{aligned} \overline{\underline{A \quad \cdots \quad A}} &= \sum_{i_1,\ldots,i_n,j_1,\ldots,j_n} \varepsilon_{i_1,\ldots,i_n}\varepsilon_{j_1,\ldots,j_n} A_{i_1,j_1}\cdots A_{i_n,j_n} \\ &= \sum_{\sigma,\tau} \operatorname{sign}(\sigma)\operatorname{sign}(\tau) A_{\sigma_1,\tau_1}\cdots A_{\sigma_n,\tau_n} \\ &= \sum_{\sigma} \operatorname{sign}(\sigma) \sum_{\tau} \operatorname{sign}(\tau) A_{\sigma_1,\tau_1}\cdots A_{\sigma_n,\tau_n} \\ &= \sum_{\sigma} \operatorname{sign}(\sigma)^2 \det(A) \\ &= n!\det(A). \end{aligned}$$

The definition generalizes to Cayley's "hyper determinants" by ....

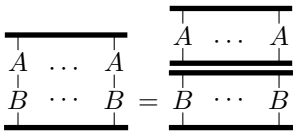
A curious property is that



(18) $\det(A) = \prod_i \lambda_i$ ...

(19) $\det(cA) = c^n \det(A)$ 

(20) $\det(A) = \det(A^T)$ ...

(21) $\det(AB) = \det(A)\det(B)$ 

(22) $\det(A^{-1}) = 1/\det(A)$ ...

(23) $\det(A^n) = \det(A)^n$ ...

(24) $\det(I + uv^T) = 1 + u^T v$ ...

5.2 Inverses

Might be reduced, unless cofactor matrices have a nice representation?

Chapter 6

Functions

Stuff about analytical matrix functions. Such as Exponential Matrix Function. I'd rather talk about my general function notation. And maybe about taylor series?

6.0.1 Pseudo-linear form

Maybe this should just be an example in a table?
Derivation of Peyman Milanfar's gradient

$$\begin{aligned}
\mathrm{d}[\mathbf{f}(\mathbf{x})] &= \mathrm{d}[\mathbf{A}(\mathbf{x})\mathbf{x}] \\
&= \mathrm{d}[\mathbf{A}(\mathbf{x})]\mathbf{x} + \mathbf{A}(\mathbf{x})\mathrm{d}\mathbf{x} \\
&= \mathrm{vec}\{\mathrm{d}[\mathbf{A}(\mathbf{x})]\mathbf{x}\} + \mathbf{A}(\mathbf{x})\mathrm{d}\mathbf{x} \\
&= \mathrm{vec}\{\mathbf{I}\mathrm{d}[\mathbf{A}(\mathbf{x})]\mathbf{x}\} + \mathbf{A}(\mathbf{x})\mathrm{d}\mathbf{x} \\
&= \left(\mathbf{x}^T \otimes \mathbf{I}\right) \mathrm{vec}\{\mathrm{d}[\mathbf{A}(\mathbf{x})]\} + \mathbf{A}(\mathbf{x})\mathrm{d}\mathbf{x} \\
&= \left(\mathbf{x}^T \otimes \mathbf{I}\right) \mathrm{D}\,\mathrm{vec}[\mathbf{A}(\mathbf{x})]\mathrm{d}\mathbf{x} + \mathbf{A}(\mathbf{x})\mathrm{d}\mathbf{x} \\
&= \left[\left(\mathbf{x}^T \otimes \mathbf{I}\right) \mathrm{D}\,\mathrm{vec}[\mathbf{A}(\mathbf{x})] + \mathbf{A}(\mathbf{x})\right] \mathrm{d}\mathbf{x}
\end{aligned}$$

6.1 Taylor Series

I've always wanted to write these out properly with tensors.

Chapter 7

Advanced Derivatives

7.1 Derivatives of vector norms

7.1.1 Two-norm

$$\frac{d}{dx}\|x-a\|_2 = \frac{x-a}{\|x-a\|_2} \tag{7.1}$$

$$\frac{d}{dx}\frac{x-a}{\|x-a\|_2} = \frac{I}{\|x-a\|_2} - \frac{(x-a)(x-a)^T}{\|x-a\|_2^3} \tag{7.2}$$

$$\frac{d}{dx}\|x\|_2^2 = \frac{d}{dx}\|x^T x\|_2 = 2x \tag{7.3}$$

7.2 Derivatives of matrix norms

7.3 Derivatives of Structured Matrices

7.3.1 The Chain Rule

Sometimes the objective is to find the derivative of a matrix which is a function of another matrix.

7.3.2 The Hessian Chain Rule

See main.tex

Chapter 8

Special Matrices

8.0.1 Block matrices

Stuff like Schur complements is interesting. But can we say anything useful using tensor diagrams?

8.0.2 The Discrete Fourier Transform Matrix

I think FFT can be nicely described with diagrams

Let's start with the Hadamard matrix: $H_n = \left[\begin{smallmatrix} 1 & 1 \\ 1 & -1 \end{smallmatrix}\right]^{\otimes n}$. Hm. It's just a bunch of matrices below each other, kinda boring.

What about the FFT? Does that require a bit more?

8.0.3 Hermitian Matrices and skew-Hermitian

Complex. Skip

8.0.4 Idempotent Matrices

Skip

8.0.5 Orthogonal matrices

Skip

8.0.6 Positive Definite and Semi-definite Matrices

Skip

8.0.7 Singleentry Matrix, The

Describes the matrix J. All of this is trivial with diagrams.

8.0.8 Symmetric, Skew-symmetric/Antisymmetric

Could introduce Penrose's symmetric tensors here?

8.0.9 Toeplitz Matrices

Could talk about the convolution tensor here...

8.0.10 Units, Permutation and Shift

Not that interesting...

8.0.11 Vandermonde Matrices

Does this have a nice description? Not a lot of properties are given in the Cookbook.

Chapter 9

Machine Learning Applications

Chapter 10

Tensorgrad

Implementation details

10.1 Simplification Rules

10.2 Functions

10.3 Isomorphisms

Chapter 11

Appendix

Contains some proofs, such as of equation 524 or 571. They are pretty long and could be useful for contrasting with the diagram proofs.