Tensor 介绍

> 1、搜索陈列 what is tensor
>
> 2、领域：数学、物理、计算列举几种 tensor
>
> 3、对这个符合的想法如 NamedTensor, 我的想法 TypeTensor
>
> 4、tensor 再 tf、pytorch 的区别，接口陈述
>
> 5、tensor 计算的优化方式

1、各种 Product：

	中文	符号	等效性	意义/计算	特点	其他关系	
inner Product	内积	<a, b> <a | b>					https://en.wikipedia.org/wiki/Inner_product_space
Dot Product	点积	点符号: •	同 内积	Σaibi			
Scalar product	na		同 内积				
Exterior Product	外积	楔形符号：\wedge			3 维/ 7维存在意义。 其他维度不确定	Hodge star	https://en.wikipedia.org/wiki/Exterior_algebra https://math.stackexchange.com/questions/720813/do-four-dimensional-vectors-have-a-cross-product-property
Cross Product	叉积	X 叉符号	同 外积?		3 维下明确的计算和意义	Levi-Civita tensor	https://en.wikipedia.org/wiki/Cross_product

Wedge product	na		同 外积				
Tensor Product	张量积	⊗					
Outer Product	na		同张量积				
Kronecker Product	na		同张量积	tensorflow			https://en.wikipedia.org/wiki/Kronecker_product https://math.stackexchange.com/questions/203947/tensor-product-and-kronecker-product https://www.statlect.com/matrix-algebra/Kronecker-product

matrix multiplication ?

The cross product **a** × **b** is orthogonal to the bivector **a** ∧ **b**

https://en.wikipedia.org/wiki/Bivector

https://math.stackexchange.com/questions/1107690/exterior-product-vs-cross-product

2、Tensor wiki 摘要

https://en.wikipedia.org/wiki/Tensor

1、In mathematics, a **tensor** is an algebraic object that describes a multilinear relationship between sets of algebraic objects related to a vector space.

2、Tensors are defined independent of any basis, although they are often referred to by their components in a basis related to a particular coordinate system.

3、 any tensor with respect to a basis is represented by a multidimensional array. For example, a linear operator is represented in a basis as a two-dimensional square $n \times n$ array. The numbers in the multidimensional array are known as the *scalar components* of the tensor or simply its *components*.

// 往往在固定了基础 base （新base 是基础base 的线性组合）， 那么 tensor 就是可 高维的 array，其实就是已经被数字化建模可用来直接计算。 这里和 tensor 是和 base 无关的并没用冲突。 因为在 array 里面并没有写 base 是具体什么 e， 具体的 e 的物理意义可再进行更换。

// 在 tensor 等 object 进行数字化之后， 高维情况如 3 维，其实描述所谓的 “行向量” "列向量" 失去意义，所谓高坐标和低座标应该有更好的表示方法

3' dual vector sapce

任何线性变化可将 V 空间下的任意向量 v 转换为 scalar in F

V -> F

对任何向量空间的操作，可非常“捷径”的转换为 base 的基础操作，因为此空间下的任何其他元素可表示为 base 的线性组合。 那么 V base {e1, e2 .... en} 那么定义一个 dual space V_star base {<e1|_>, <e2|_> ...<en|_> }

V_star 下的任何元素可线性组合其 base， 任意的一个 g = a1w1 + a2w2 + ...anwn)

可见 g 可将任意原空间向量 v 转换为标量。

（a1w1 + a2w2 + ...anwn） * （b1e1 + b2e2 + ...bnen） // 很多 ‘行’ 元素其实隐含着 加号

从上面看， 所谓的 object 不仅仅是'数'还可'运算', 当进 ‘行多项式乘法’ 这一个基本操作时候, 这个乘法可不是常规的数字乘法，而是先记录一个为抽象的 tensor_product， 然后再具体的 base 上再表明‘具体意义’ 如 a1w1 tensor_product b1e1 定义为 <a1e1|b1e1> = 1 <a1e1|b2e2> = 0

这样的先 tensor_product 再 具体定义线性变化，也正是 Universal property of tensor product：

$$\begin{tikzcd} V \times W \arrow[r, "\varphi"] \arrow[dr, "h"'] & V \otimes W \arrow[d, dashed, "\tilde{h}"] \\ & Z \end{tikzcd}$$

另外这里的也将具体 base （后续可变，所以不是依赖based），来简洁的找到第一步骤本质：多项式展开，第二部本质：base tensor 的具体定义和算法。