笔记 + 心得

JX-Ma

2023 年 10 月 21 日

1 摘要

张量 (多维数组): 可以表示多维视觉数据的数据结构。

张量方法在深度学习中应用的越来越广泛, 能够设计内存和计算效率高的网络架构, 提高对随机噪声和对抗性攻击的鲁棒性, 以及帮助对深度网络的理解。

鲁棒性: 指系统或算法抵抗干扰或意外变化的能力。

这篇文章主要在深度学习的背景下, 对张量和张量方法进行了深入而实用的回顾, 特别关注视觉数据分析和计算机视觉应用。

2 引言

2.1 张量的介绍和应用

张量是矩阵的多维泛化, 它是多维线性代数中的核心对象, 张量的阶数是指寻找它的元素所需要的索引数。如矩阵是 2 阶张量, 他需要两个索引去访问它的元素。

张量在科学和工程的广泛学科中得到了很多应用, 从量子系统到几何代数和理论计算机科学. 张量能被用于表示和分析多维数据中的信息, 也可以用于捕获和利用向量值变量之间的高阶相似性或依赖性.

张量方法主要侧重于扩展基于矩阵的学习模型, 如成分分析、字典学习和回归模型, 以处理由高阶张量表示的数据或通过分析高阶统计量来估计潜在变量模型的参数.

可视化数据中的张量结构

- 视觉测量的张量结构: 例如灰色的图片可以有二阶张量表示, 代表长和高. 多张灰色图片可以由 3 阶张量表示, 第一个索引表示该批次中不同的图像, 彩色图片可以有三阶张量表示, 第一个索引表示颜色通道.

- 图像形成中潜在张量结构: 图像形成依赖于多个潜在因素的相互作用, 通过这些因素产生了丰富的视觉数据结构, 我们保持一个因素外的其他所有因素不变, 那么视觉变化将会是线性的.

传统的计算机视觉和视觉数据分析应用存在的问题

- 结构丢失: 再应用计算机视觉进行工作时, 需要提取一些具有特征的数据, 这样在不同的测量模式下就会丢失一些自然拓扑结构和依赖关系。
- 维数诅咒: 在高维向量训练基于矩阵的机器学习模型时, 所需要的数据样本的数量会随着数据维度成指数增长.

两种张量方法

- **计算机视觉表示学习中的张量方法**: 它可以减轻维数诅咒并且不丢失结构, 张量分解和张量分量分析方法在广泛的计算机视觉应用中产生了变革性的影响, 张量分解可以把视觉数据分解为形状和运动因素, 或者其他因素, 有助于恢复数据。深度学习模型通过局部统计利用数据的统计属性, 讲这些属性给卷积架构利用, 提取图像域共享的局部特征, 这样大大减少了学习参数的数量从而减轻了维数诅咒的影响。
- **计算机视觉深度学习架构中的张量方法**: 虽然深度神经网络的组合结构减轻了维数诅咒, 但是深度学习网络模型的参数依旧很多, 当这些训练参数进行梯度下降时就会遇到过拟合等问题. 张量分解可以显著减少深度模型中未知参数的数量, 进一步缓解维数诅咒。例如在 cnn 中多通道卷积核可以使用低秩张量分解成模式可分卷积的核。

3 关于矩阵和张量的初论

3.1 常用的符号

- 标量:x,y,j
- 向量 (一阶张量):$\mathbf{x}$
- 矩阵 (二阶张量):$\mathbf{X}$
- 相等维数的单位矩阵: $\mathbf{I}$
- 矩阵转置: X^T
- X 的第 i 列: X_i
- 实数: $\mathbf{R}$
- 整数: $\mathbf{Z}$
- M 个不同维数的矩阵 $\{\mathbf{X}_{(m)} \in \mathbb{R}^{I_m \times N}\}_{m=1}^{M}$

- M 个不同维数的向量表示 $\{\mathbf{x}_{(m)} \in \mathbb{R}^{I_m}\}_{m=1}^{M}$
- 多维张量:$\mathcal{X}$
- : 表示某一个索引下的所有值

3.2 将张量分解为矩阵或者向量

- 张量展开: 张量可以按照不同的维度展开, 按照一些运算库期待的样式展开运算效率会大幅度提高.
- 张量矢量化: 将张量分解为有顺序的向量

3.3 矩阵与张量积

- Kronecker 积: A 矩阵中每一个元素去乘 B 矩阵, 例如 $A \in \mathbb{R}^{I_1 \times I_2}, B \in \mathbb{R}^{J_1 \times J_2}$ 得到矩阵 $C \in \mathbb{R}^{I_1 \cdot J_1 \times I_2 \cdot J_2}$ 定义为 $A \otimes B = C$
- Khatri-Rao 积: 给定两个列相同的矩阵如 $A \in \mathbb{R}^{I \times R}, B \in \mathbb{R}^{J \times R}$ 将 A 中的每一列和 B 中的每一列做 Kronecker 积, 最后得到 $C \in \mathbb{R}^{I \cdot J \times R}$ 定义为 $A \odot B = C$
- Hadamard 积: 元素积给定两个行列都相同的矩阵 $A, B \in I \times J$, 让这两个矩阵的元素各自相乘得到矩阵 C, 定义为 $A \times B = C$
- 外积: 给定一组 N 个的向量集合 $\{\mathbf{x} \in \mathbb{R}^{I_n}\}_{(n=1)}^{N}$, 将他们组成 N 阶秩为 1 的张量. 记为 $\mathcal{X} = x^{(1)} \circ x^{(2)} \circ \ldots \circ x^{(N)}$
- n 模积: 给定一个张量 $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ 和一个矩阵 $M \in \mathbb{R}^{R \times I_n}$,n 模积记为 $\mathcal{X} \times_n M = \mathbf{MX}_{[n]} \in \mathbb{R}^{I_1 \times \ldots \times I_{n-1} \times R \times I_{n+1} \times \ldots \times I_N}$; 该张量与向量 $x \in \mathbb{R}^{I_n}$ 做 n 模积得到 $\mathcal{X} \prod_{n=1}^{N} \times_n x_{(n)} = \mathbb{R}^{I_1 \times \ldots \times I_{n-1} \times I_{n+1} \times \ldots \times I_N}$
- 广义内积: 两个相同的多维张量 $\mathcal{X}, \mathcal{Y} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$
- 卷积:

3.4 张量图

顶点圆表示张量, 边上的线代表张量的阶数, 标量的旁边有一条线, 矩阵 (二阶张量) 旁边就有两条线, 三阶四阶的张量也是一样的. 两个张量公共维度的张量收缩可以采用连接公共维度的边表示他们的乘积

3.5 矩阵的秩

定义一个矩阵 $A \in \mathbb{R}^{I \times J}$, 代表一个 $I \times J$ 的矩阵, 矩阵的秩表示为 $rank(A)$ 它也可以定义为 A 线性无关列的个数, 或者 A 线性无关行的个数.

且存在 $rank(\mathbf{A}) \leqq min(I, J)$ 如果 $rank(\mathbf{A}) = min(I, J)$, 则称 A 为全秩

3.6 范数

在任意阶数中，张量范数的主要家族之一是逐元素范数。

- $l_p norm$: $\mathcal{X}$ 的 $l_p norm$: $\|\mathcal{X}\|_p = \|vec(\mathcal{X})\|_p = (\sum_{i_1,i_2,\ldots,i_N} |\mathcal{X}_{i_1,i_2,\ldots,i_N}|^p)^{\frac{1}{p}}$

3.7 心得

本次学习主要了解了本文大概讲的内容, 本文讲的是张量的一些介绍和表示, 以及张量方法在计算机视觉和数据分析上的内容.

第一章主要介绍了张量和它的一些应用场景, 首先, 张量也可以称为多维数组, 它是矩阵的多维泛化, 是多维线性代数中的核心对象, 张量的阶数代表着访问张量中的某个元素需要的索引数量, 例如一阶张量可以理解为一维数组, 这个时候访问它其中的每个元素就需要提供一个索引值, 同样的 n 阶张量就需要 n 个索引值去访问. 之后介绍了张量在一些学科中的应用, 张量主要可以用于表示和分析多维数据中的信息, 也可以用于捕获和利用向量值变量之间的高阶相似性或依赖性. 张量方法主要侧重于扩展基于矩阵的学习模型, 比如成分分析和回归模型一类的. 之后介绍了一些可视化数据中的张量结构, 比如在视觉分析图片时, 如果图片是灰白的, 也就意味着不需要颜色去描述图片, 这个时候一张图片可以用二阶张量表示, 代表图片的宽和高, 如果图片是有颜色则需要三阶张量, 需要存储颜色通道, 我之前接触到的直接卷积算法中的四阶张量中, 各索引分别表示图片的个数, 颜色通道数, 宽和高. 然后又说了传统的计算机视觉和视觉中的数据分析存在的两个问题, 一个是结构丢失的问题, 另一个是维数诅咒. 结构丢失主要是因为计算机视觉进行工作时, 需要提取一些具有特征的数据, 这样在不同的测量模式下就会丢失一些自然拓扑结构和依赖关系。维数诅咒是指在高维向量训练基于矩阵的机器学习模型时, 所需要的数据样本的数量会随着数据维度成指数增长. 之后提出了两种张量方法去解决这些问题, 大致是根据数据的某些特征进行张量分解, 通过降维等方式去优化算法。

第二章中主要介绍了一些在使用张量时常见的符号, 如多维张量如何表示, 标量如何表示... 之后介绍了将张量按照不同的维度展开, 让张量中的数据变为有利于运算库提高计算速度的样子提高整体的运行效率, 接着介绍了 7 种矩阵张量之间的积运算, 如 Kronecker 积,Khatri-Rao 积和如何将张量用图表示, 将来一些基本张量怎样使用无向图表示, 接着介绍了矩阵的秩和范数.

这周看完这篇文章前面所讲的内容, 让我了解了张量这个表示高维数据的数据结构, 相当于补了一下高性能计算一些基本的概念, 了解了张量的数据结构组成, 在阅读文献

时也能看懂一些关于介绍张量常用的符号, 在看到矩阵和张量积这部分时,n 模积和广义内积的具体做法还有范数这部分不是很了解。