

每日小结

	周一	周二	周三	周四	周五
早	对比实验补充， 论文撰写	代码，多尺 度，审稿子	论文撰写， 对比实验补充， 复现	Loss 代码修改， 跑代码	
中	审稿子	多尺度， 论文撰写， 审稿子	论文撰写，审稿 子	网络图， 跑代码	Loss 代码和公 式，跑代码
晚	实验补充，跑代 码	代码调参	整理结果	网络图绘制，跑 代码	论文撰写，跑代 码

注：简单表述当前时间段工作，如看文献1，整理数据等

科研详情

文献阅读

文献 1

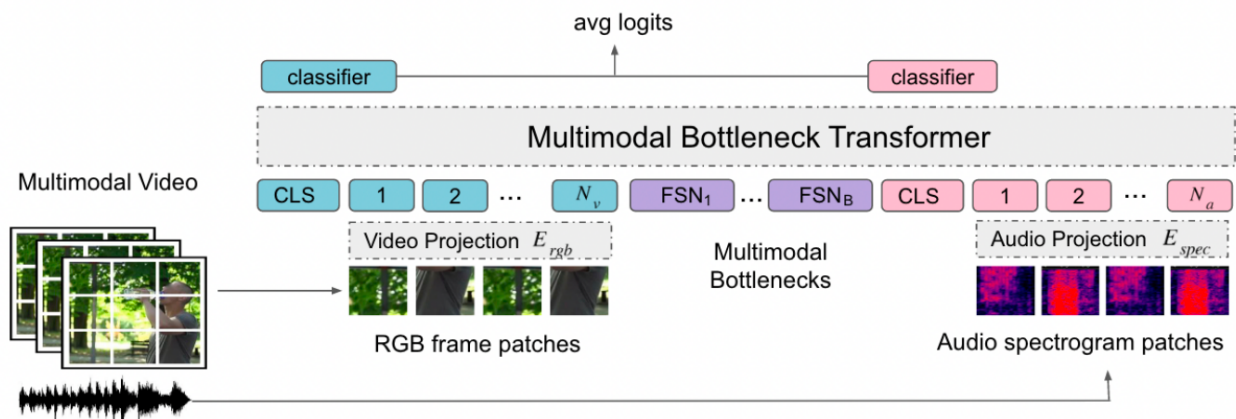
题目：Attention Bottlenecks for Multimodal Fusion

作者：Arsha Nagrani Shan Yang Anurag Arnab Aren Jansen Cordelia Schmid Chen Sun

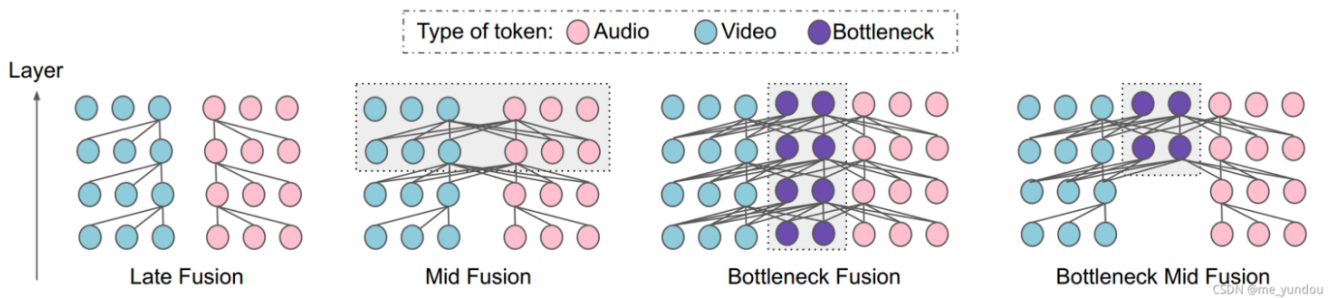
出处：arXiv:2022

方法：

人们对世界的认知，对信息的处理是多模态的，而大多的机器学习模型却是仅针对单模态的。同时，处理多模态问题的模型，大多还是使用 late-stage 的 fusion 方法，先分别处理单个模态数据之后 fusion 为多模态结果。本文提出一种基于 transformer 的多层 fusion 方法，借助于“fusion bottlenecks”。本文让不同模态的信息穿过许多小的 bottlenecks，迫使模型 collate 和 share 不同模态中最重要的信息。作者发现通过这种方式，模型的 fusion 性能更好，且计算消耗降低。本文做了完整的消融实验，在多个音视频分类基准数据集上取得了 SOTA 的性能，包括 Audioset, Epic-Kitchens and VGGSound。代码和模型已公开。



本文设计了两种方式来解决原始 transformer 中 attention 的问题。1. 如同多数多模态 fusion 模型一样，将 fusion 部分往后推移，先让模型单独处理单个模态的信息，然后再做 fusion（做 mid fusion，而不是 early fusion）。这样能够充分提取单模态内部的信息，毕竟不同模态的数据结构和分布差距很大，使用一样的处理方式是不合理的。2. 在 layer 内的不同模态的 tokens 之间做跨模态的 attention。单模态内部仍然是原始的 self-attention，但是跨模态的 fusion 使用每个模态的部分 tokens 信息来做 cross-attention。这样就能降低计算量并且处理部分冗余信息。



启发:

1. 单模态内部直接使用 self-attention, 那么其冗余信息就没有处理? 或者说在提取单模态信息做 fusion 的时候, 避开冗余的信息, 只提取有效的, 这样也算是成功避免了单模态内部冗余信息的影响? 毕竟最终的目的是做 fusion

文献2

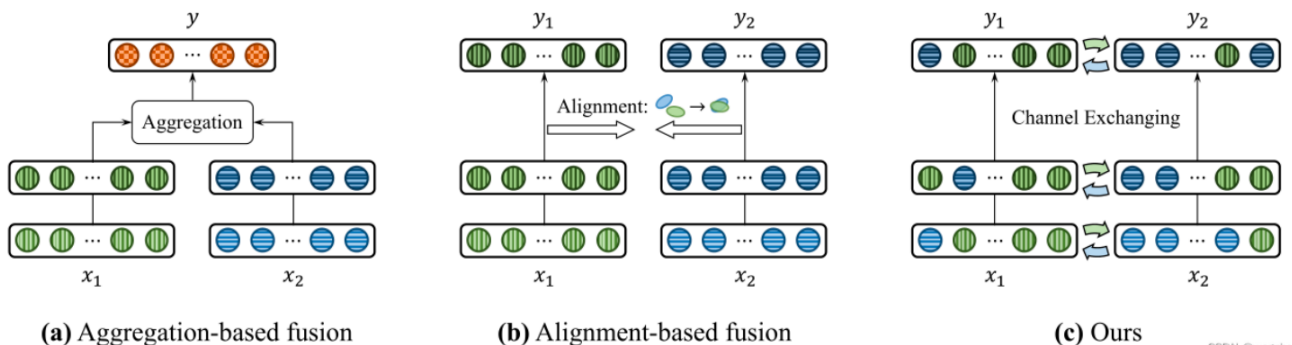
题目: Deep Multimodal Fusion by Channel Exchanging

作者: Yikai Wang, Fuchun Sun, Wenbing Huang, Fengxiang He, Dacheng Tao

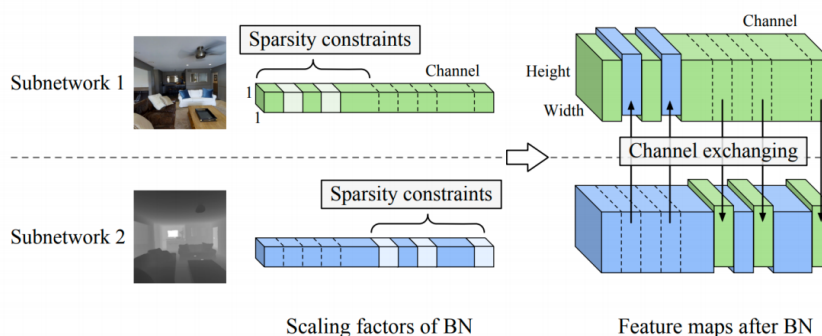
出处: TPAMI-2022

方法:

本文提出了通道交换网络 (CEN), 一个无参数的多模态融合框架, 在不同模态的子网络之间动态地交换通道。具体来说, 信道交换过程是由单个信道的重要性自我引导的, 这个重要性是由训练期间的批量标准化 (BN) 缩放因子的大小来衡量的。这种交换过程的有效性也是通过共享卷积滤波器, 但在不同的模式下保持独立的 BN 层来保证的



CEN 自适应地在子网络之间交换信道。CEN 的核心在于其受网络修剪启发的较小范数信息量较少的假设。具体而言, 利用批量归一化 (BN) 或实例归一化 (IN) 的缩放因子 (即 γ) 作为每个相应信道的重要性度量, 并用其他子网络的平均值替换与每个子网络的接近零因子相关的信道。CEN 的另一个特点是, 除了所有子网络的 BN 层之外, 参数是彼此共享的。

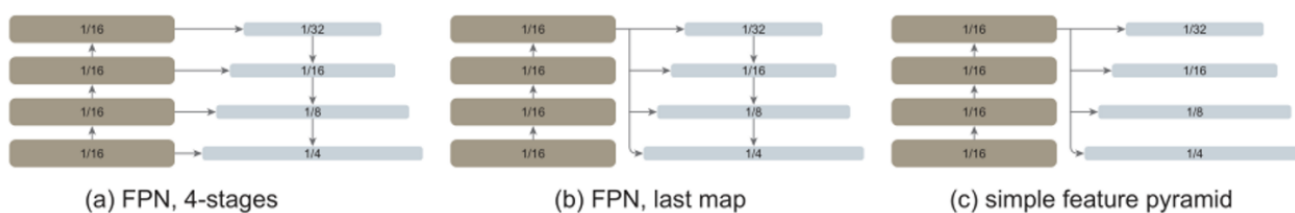


启发:

1. 文章公式多，代码较为简单，还在一边看代码一边看公式，看能否套用在特征融合部分

工作进展

- 1: 阅读文献;
- 2: 论文撰写制作; 网络图绘制
- 3: 整理实验结果
- 4: 多尺度模块换了参数，有微小提升



- 5: 尝试了几种简单的融合方法，效果都不好
- 6: 审稿子，目前提了 5 条，还差方法的提问

下周计划

1. 网络修改: 继续看论文想 idea:
2. self-attention 换多头注意力代码
3. 类似双重融合，用花里胡哨的简单fusion方法修改一下fusion的concat，还想看看变化检测里做的一些简单花里胡哨的fusion，增加创新点
4. 论文撰写