周报（2022. 12.14-2022.12.27）姓名：孙瑞阳

每日小结

	周一	周二	周三	周四	周五
早	信令数据代码	信令数据代码	上课，信令数据	阅读文献	上课，跑模型
中	上课，结课论文	听讲座	论文阅读	上课，信令数据	上课，信令数据
晚	信令数据代码	信令数据，上课	新生讲课学习	上课，结课论文	上课，跑模型

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：TRANS-BLSTM: Transformer with Bidirectional LSTM for Language Understanding

作者：Zhiheng Huang, Peng Xu Amazon, Davis Liang, Ajay Mishra, Bing Xiang

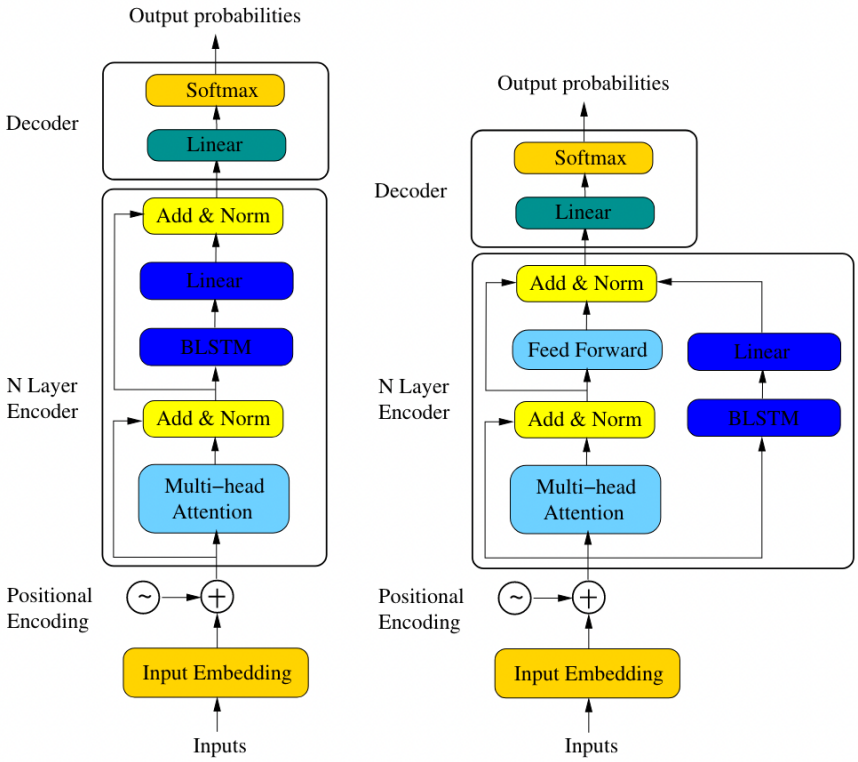
出处：arXiv 2020

方法：

在本文中，研究了如何结合 BiLSTM 和 Transformer 来创建更强大的模型架构，称为 Transformer with BLSTM (TRANS-BLSTM)，它有一个 BLSTM 层集成到每个 transformer 块，从而形成了 transformer 和 BLSTM 的联合建模框架。

TRANS-BLSTM-1 每个 BERT 层用双向 LSTM 层替换前馈层。

TRANS-BLSTM-2 添加了一个双向 LSTM 层，它采用与原始 BERT 层相同的输入。双向 LSTM 层的输出与原始 BERT 层输出（在 LayerNorm 之前）相加。



启发：

1. 如果使用与 BERT 模型 H 中相同数量的 BLSTM 隐藏单元，将获得维度为 2H 的 BLSTM 输出，因此需要一个线性层来投影 BLSTM 的输出（维度为 2H）到 H，以匹配 transformer 输出。或者，如果将 BLSTM 隐藏单元的数量设置为 H/2，则不需要包括额外的投影层。

文献2

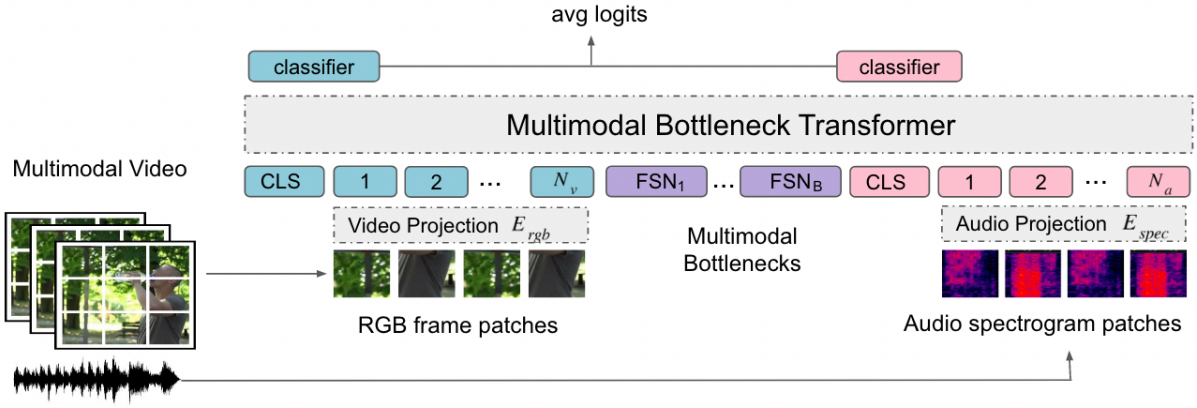
题目：Attention Bottlenecks for Multimodal Fusion

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

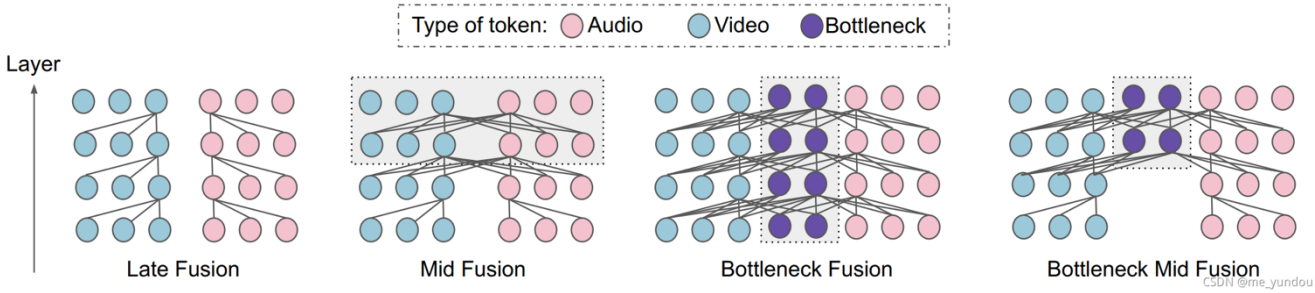
出处：arXiv:2022

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. 问题：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
2. 大多数基于transformer的工作堆叠的多个 transformer layer都使用相同的操作（比如ViT）。然而在多模态transformer中，一个共识是在前期先让各个模态分别学习自己的特征，后期再进行多模态的融合。因为我们通常认为前面的层用来学low-level的特征，后面的层学习high-

level的特征，而low-level的不同模态特征之间可能还没有出现明显的关联关系，所以融合要放在后面层进行。

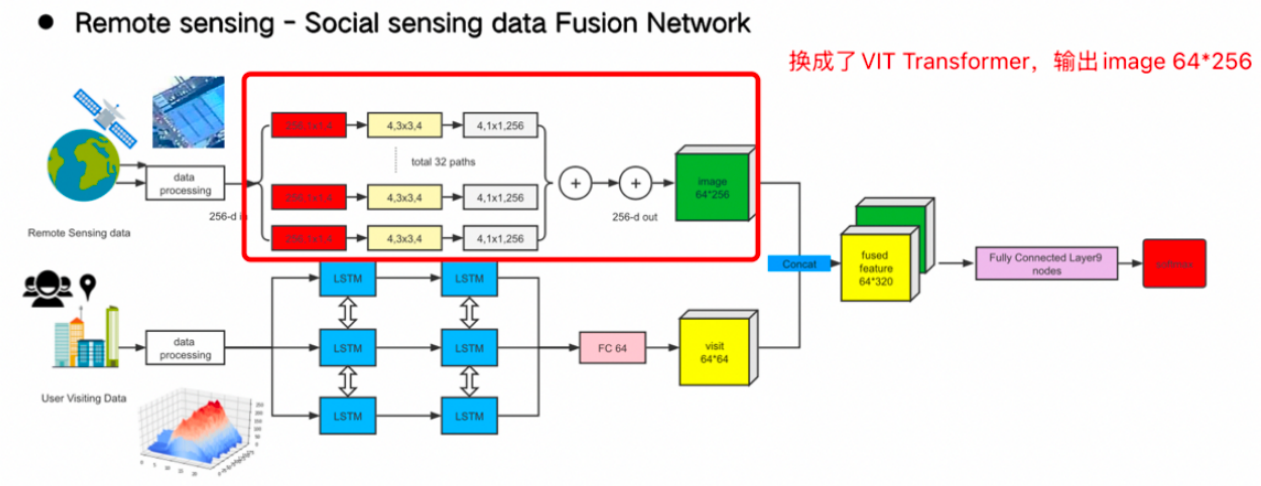
3. 原始transformer里面的attention层能够freely接触和处理每个token之间的关系，这样对于模态内的冗余信息会造成计算量浪费。所以本文将原始transformer中attention层修改为模态内attention（保持self-attention结构不变）+ 模态间attention（设计cross-attention只在每个模态的部分token之间做attention，避免过度计算冗余信息，降低计算量。并且选择了mid fusion，探讨了fusion层在模型early，mid，late部分的影响。

4. 本文的融合方法可以作为借鉴，有开源代码，正在学习。

工作进展

1：阅读文献；寻找transformer和lstm层级间进行权重共享或者约束的论文

2: **在自己的网络里把遥感编码器换成了预训练的 VIT Transformer，社交用的还是 2 层 BiLSTM：准确度 63.6%（毕设 66.8%），感觉还能调参。**
（无预训练 transformer+ BiLSTM 准确度 53.3%）



3: Daas平台上的郑州信令数据（下载了2021年7月**24-31日**每小时的数据）。

下周计划

1：阅读文献；寻找 transformer 和 lstm 层级间进行权重共享或者约束的论文，**遥感和社交数据可以在层级对比权重信息，最大程度互补，**想办法改模型

2: 完成预训练VIT transformer➕卷积LSTM代码