Multimodal Learning with Transformers：A Survey

引言部分论述了多模态的由来：模仿人类的感知，形成共同利用多种感知数据模态，动态且不受约束的前提下与周围环境互动的基本机制。模态通常与创建独特通信渠道的特定传感器相关联，每种模态都作为具有不同统计特征的独特的信息源。

transformer主要多模态建模实践

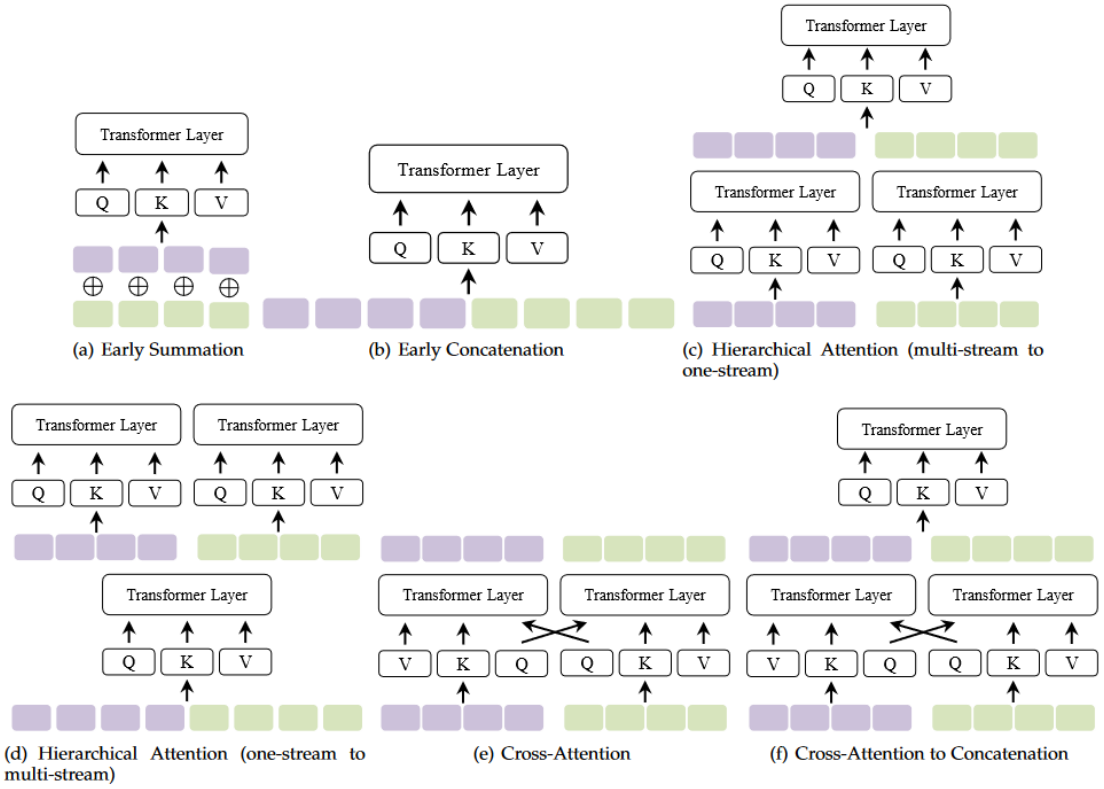

Fig. 4. Transformer-based cross-modal interactions. "Q": Query embedding; "K": Key embedding; "V": Value embedding. Best viewed in colour. See text for details.

1.Early Summation

Early Summation（早期求和）是一个简单且高效的多模态交互方式，它将来自多个模态下的token embedding可以在每个token position(标记位置)进行加权求和，然后再由Transformer层进行处理。具体来说，假设X_A与X_B是来自任意两个模态的输入，Z_A与Z_B分别代表了这两个输入的token embedding，并且设Z表示通过多模态交互作用生成的token embedding，$T_f(.)$代表transformer模块。用数学公式来表示上述过程为：

$$Z \longleftarrow T_f(\alpha Z_A + \beta Z_B) = MHSA(Q_{AB}, K_{AB}, V_{AB})$$

$$Q_{AB} = (\alpha Z_A + \beta Z_B) W_{AB}^{Q}$$

$$K_{AB} = (\alpha Z_A + \beta Z_B) W_{AB}^{K}$$

$$V_{AB} = (\alpha Z_A + \beta Z_B) W_{AB}^{V}$$

该方法的优点在于其计算复杂度为$O(N_A^2)$（N_A表示A模态下token序列长度），缺点在于其需要人为手动设置权重。

2.Early Concatenation

Early Concatenation（早期连接）是另外一个直接的多模态交互方式，它将来自多个模态下的token embedding序列拼接起来，再将其送入Transformer层进行处理。用数学公式来表示为：

$$Z \longleftarrow T_f(C(Z_A, Z_B)) = MHSA(Q_{AB}, K_{AB}, V_{AB})$$

在这种模式下，所有的多模态下的token position可以作为一个整体序列被处理，这样每个模态的位置可以通过调节其他模态的上下文来很好的编码（？），但是拼接起来的较长序列会增加计算复杂度。

3.Hierarchical Attention(multi-stream to one-stream)

Hierarchical Attention（分层注意（多流到单流））中transformer层可以分层组合以实现跨模态的融合。一个常见的做法就是首先多模态输入是由独立的 transformer 流编码，然后将它们各自的输出送入另一个 transformer 层进行连接和融合。用数学公式来表示为：

$$Z \longleftarrow T_{f_3}(C(T_{f_1}(Z_A), T_{f_2}(Z_B))) = MHSA(Q_{AB}, K_{AB}, V_{AB})$$

这种分层注意力是后期交互/融合的一种实现。

4.Hierarchical Attention(one-stream to multi-stream)

Hierarchical Attention（分层注意（单流到多流））是先连接多模态输入，再将拼接过后的token embedding由共享的单流 Transformer 编码，最后再分别送入到两个单独的 Transformer 流中处理。用数学公式来表示为：

$$\begin{gathered} Z_A,\ Z_B \longleftarrow T_{f_1}(C(Z_A,\ Z_B)) \\ Z_A \longleftarrow T_{f_2}(Z_A) \\ Z_B \longleftarrow T_{f_3}(Z_B) \end{gathered}$$

这种方法既感知跨模态交互，同时保持单模态表示的独立性。

5.Cross-Attention

Cross-Attention（交叉注意）是指对于双流的transformer，Q(query)是以跨流的方式交换。用数学公式来表示为：

$$\begin{gathered} Z_A \longleftarrow MHSA(Q_B, K_A, V_A) \\ Z_B \longleftarrow MHSA(Q_A, K_B, V_B) \end{gathered}$$

Cross-attention 以其他模态为条件对每个模态进行关注，并且不会导致更高的计算复杂度，但是如果同时考虑每个模态，该方法无法全局执行跨模态注意，因此会丢失整个上下文。两流交叉注意可以学习跨模态的相互作用，而对每个模态内部的自我背景不存在自我注意。

6.Cross-Attention to Concatenation

Cross-Attention to Concatenation（交叉注意并连接）是在两个交叉注意力流的基础上进一步连接并由另一个 Transformer 处理以对全局上下文进行建模，减轻了5中所述的交叉注意的缺点。用数学公式来表示为：

$$\begin{gathered} Z_A \longleftarrow MHSA(Q_B, K_A, V_A) \\ Z_B \longleftarrow MHSA(Q_A, K_B, V_B) \\ Z \longleftarrow T_f(C(Z_A,\ Z_B)) \end{gathered}$$

7.一点思考

事实上上述各种建模方式也可以灵活地组合和嵌套，比如说在分层注意(一流到多流)中使用多个交叉注意流。

并且对于上述两模态的融合可推广到多模态的融合上，比如说现有三个模态的信息， 当采用交叉注意（拼接）的方式进行融合时，我们可以给定一个模态的Q（query），而Key和Value是来自其他两个模态的拼接。