Progressive Modality Reinforcement for Human Multimodal Emotion Recognition from Unaligned Multimodal Sequences

Fengmao Lv[1,2] Xiang Chen[3] Yanyong Huang[2*] Lixin Duan[4] Guosheng Lin[5*]

摘要

人类多模态情绪识别涉及不同模态的时间序列数据，例如自然语言、视觉运动和听觉行为。因为在采样各个模态时候的采样率不同，所以所以收集的多模态数据流通常是未对齐的。不同模态之间的异步性质会增加多模态融合的难度。因此，作者的这项工作主要关注未对齐多模态序列的融合。为此，作者提出了基于交叉模态 Transformer 的渐进模态强化（PMR）方法。作者的方法引入了一个消息中心来与每个模态交换信息。其中，消息中心向每个模态发送通用消息，并通过跨模态注意力强化每个模态的特征。反过来，它还从每种模态中收集增强后的特征，并使用它们生成增强的通用消息。通过重复循环过程，共同的信息和每个模态的特征可以逐步相互补充。最后，使用增强后的特征对人类情绪进行预测。通过对不同人类多模态情绪识别基准进行全面实验，证明了该方法的优越性。

关键词：多模态；情绪识别；Attention

1 引言

人类多模态情感识别专注于从视频片段中识别人类的情感态度[1]。这个任务涉及时间序列中不同模态的数据，例如自然语言、面部姿势和声学行为。多模态数据可以为深入理解情绪提供丰富的信息。

然而，在实际中，由于来自不同模态的序列的可变采样率，采集的多模态数据流通常是异步的。例如，面部表情沮丧的视频帧可能与过去所说的否定词有关。不同模态之间的异步性会增加进行多模态有效融合的难度。前面的工作通过预定义的字级对齐来解决上述问题。即，首先根据文本单词手动对齐视觉和声学序列。然后在对齐的时间步长上进行多模态融合。然而，手动单词对齐过程通常是费力的，而且需要领域知识。

最近，Tsai 等人提出了 Multimodal Transformer（MulT）方法在未对齐的的数据序列中的跨模态信息进行融合[2]。基于 transformer 的发展[3]，通过学习跨模态数据之间成对的定向注意力，使用一个模态数据对另一个数据模态数据进行增强，从而加强跨模态的数据交互。这样就可以不用明确对齐数据，实现对异步序列的模态数据融合了。方法如图1所示：

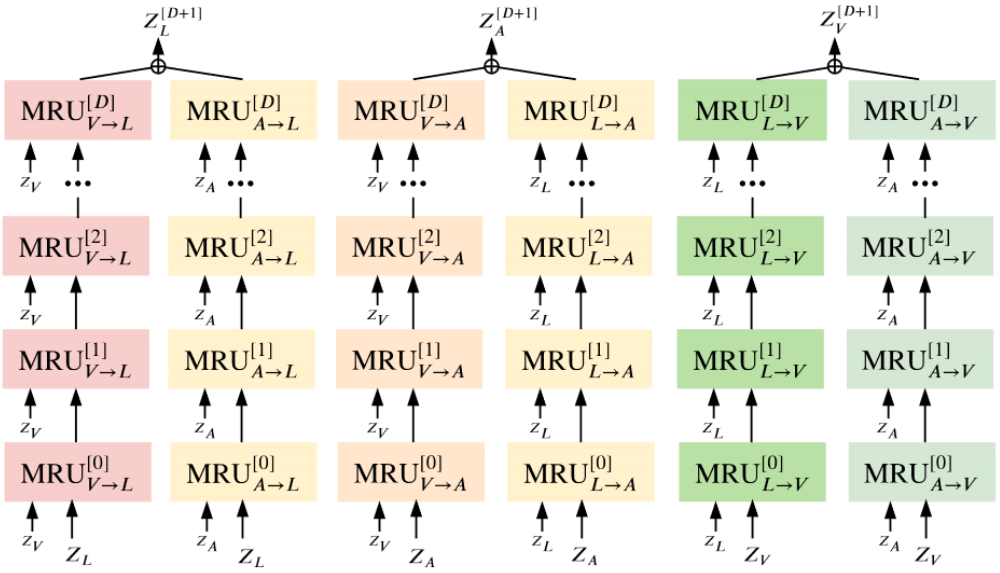

图 1: MulT

上面方法中，存在着一些问题：1）每个方向的模态强化都是独立进行的，只在成对的模态之间交换数据，没有在全部模态之间相互交换信息。2）直接对多个方向后得到的模态数据拼接，会有信息冗余问题。3）成对的融合只能重复利用到源模态的浅层特征，所以后来对源模态加入了前向传播层，获取高层语义信息。方法如图2所示：

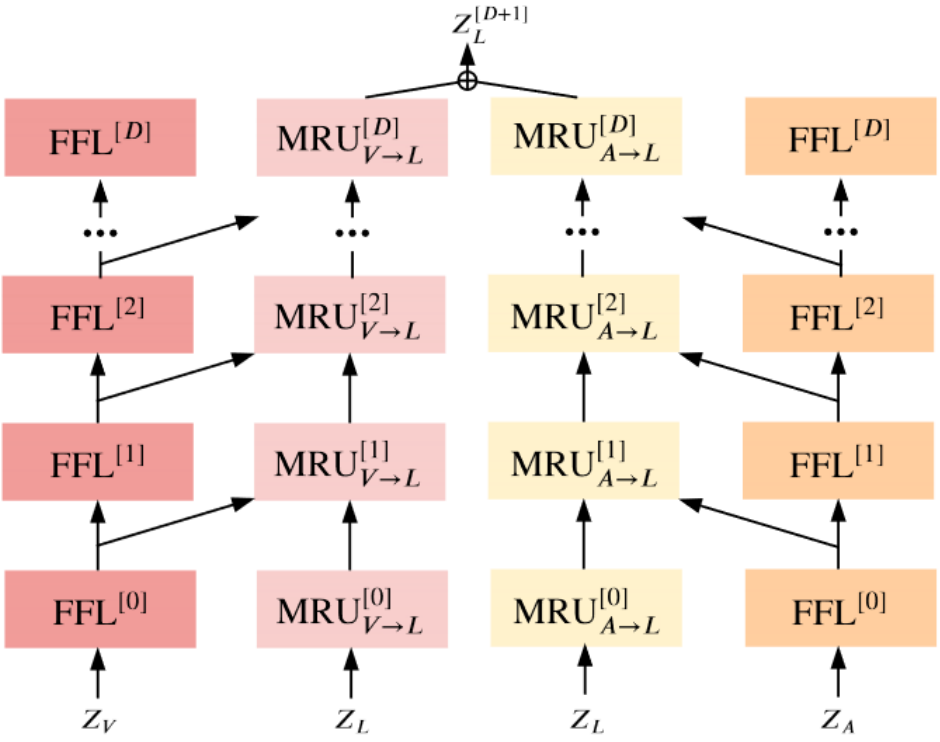

图 2: MulT-FFL

然而第二个问题还是无法解决，因为源模态的 FFL 不参与后面的模型训练，没有收到监督信息去更新，所以它不清楚哪些高层语义特征会比浅层语义特征更好，而且还增加了模型的复杂度。

因此，作者提出了基于交叉模态 Transformer 的渐进模态强化（PMR）方法。作者的方法引入了一个消息中心来与每个模态交换信息。其中，消息中心向每个模态发送共同消息，并通过跨模态注意力强化每个模态的特征。反过来，它还从每种模态中收集增强后的特征，并使用它们生成增强的共有

消息。通过重复循环过程，共同信息和每个模态的特征可以逐步相互补充。最后，我们使用增强的特征对人类情绪进行预测。而且，作者还在强化模态单元中提出了一个动态 filter 的机制，它可以动态决定增强后的特征的通过比例。

两个工作的不同：

1. 共同信息可以促使跨模态之间数据的交流，可以在这三种模态上通过跨模态注意力机制研究元素级的依赖关系。

2. 这种逐步的强化策略有效利用源模态的高层语义特征。和前面的方法不同，这里它也可以得到监督，因为它同时也是目标模态。

文章贡献

1. 对未对齐的多模态数据序列的融合，采用了 message hub 对所有模态进行了交互。

2. 采用了逐步强化策略利用源模态的高层语义信息。

3. 作者的方法可以在不同的人类多模态情绪识别基准上获得 SOTA 效果。

2 相关工作

人类多模态情感识别需要从视频片段中推断出人类的情感态度[1]。关键点在于从自然语言、视频帧和声学信号等不同模态的数据序列中进行多模态融合[4]。这个任务需要从时间序列信号中融合跨模态信息。

2.1 早期工作

早期的工作采用早期融合方法，即拼接不同模态的输入序列进行融合[5-6]；或者晚期融合，将每个模态学习到的高层语义信息结合起来[7-9]。Gan 等人提出通过概率图模型推断不同模态的联合表示[10]。

这些工作虽然比只从一种模态数据中进行学习得到更好的效果，但是它们没有明确考虑来自不同模态的序列元素之间的固有依赖性，而这个关系对于模态融合有很重要的影响作用。

2.2 当前工作

最近的一些工作通过人工在文本的单词基础上对齐视觉序列和语音序列，然后在对齐的时间步长上面进行多模态融合[4,11-12]。其中方法有分级注意力机制[12]，循环转换[11]等。

然而，人工通过单词进行对齐得到过程非常消耗人力资源，而且单词级对齐的多模态融合忽略了来自不同模态的元素之间的长距离依赖性。

所以，基于此，有工作根据最大互信息准则探索不同模态元素之间的依赖关系[13]。然而它的性能由于网络结构层数太少而太低。Tsail 等人提出了跨模态注意力机制学习不同模态之间潜在的关系[2]。他们的方法通过学习不同模态元素之间的定向成对注意，用来自其他模态的信息反复强化这一种模态。

3 本文方法

3.1 本文方法概述

本文的目标是对未对齐的多模态数据序列进行有效的多模态融合，以获得好的特征进行情感预测分析任务。

其中的变量声明如下：

对于三种不同的模态：语言（M）、视觉（V）和音频（A）

三种模态数据序列：$X_{\{L,V,A\}} \in \mathbb{R}^{T_{\{L,V,A\}} \times d_{\{L,V,A\}}}$

序列长度：$T_{(.)}$，特征维度：$d_{(.)}$

方法总述：

首先使用一维时序卷积层处理输入数据然后使用位置嵌入对它们进行增强，得到处理后的特征序列 $Z_{\{L,V,A\}} \in \mathbb{R}^{T_{\{L,V,A\}} \times d}$。这个卷积层通过在每个模态设置不同大小的卷积核，将不同模态特征投射到相同的维度。

最初的共同信息 $Z_C = [Z_L, Z_V, Z_A]$，其中 $Z_C \in \mathbb{R}^{T_C \times d}$，$T_C = T_L + L_V + L_A$。初始化是由每个模态的低层特征直接拼接起来的。同时，模态增强层会重复地通过利用不同模态之间的关系来对 Z_C 和 $Z_{\{L,V,A\}}$ 继续增强。

然后将特征进行拼接 $[Z_C, Z_L, Z_V, Z_A] \in \mathbb{R}^{2T_C \times d}$ 传入一个 transformer 层进行交互增强。

最后通过几个全连接层预测情绪分类。

模型概述图如图3所示：

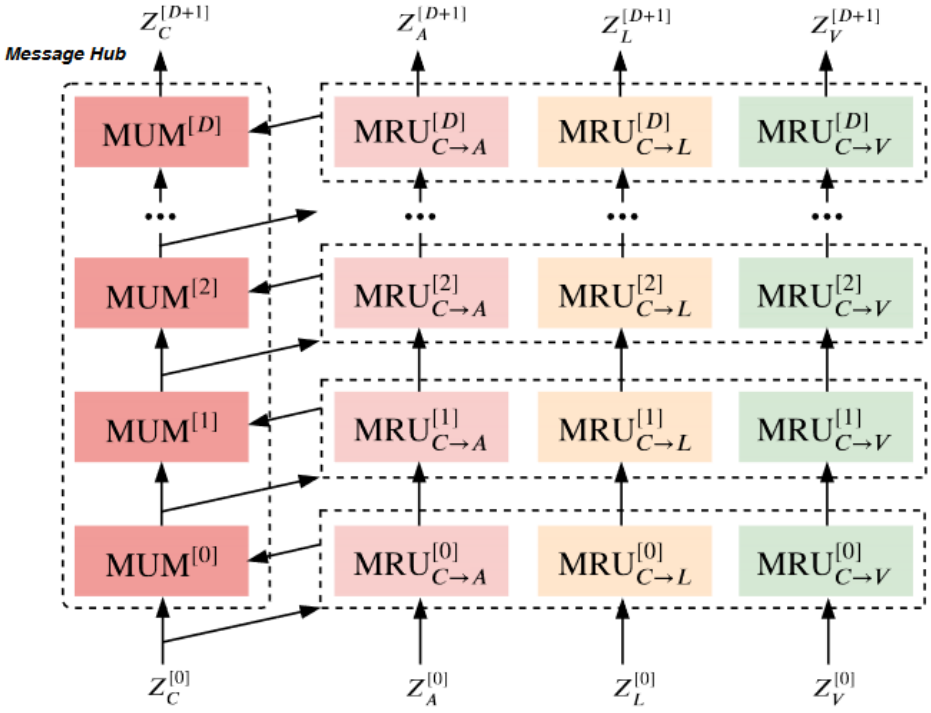

图 3: 模型概述图

3.2 跨模态注意力块

跨模态注意操作通过学习不同模态之间的定向的成对注意，利用来自源模态的信息来增强目标模态。定义，$X_s \in \mathbb{R}^{T_s \times d_s}$，$X_t \in \mathbb{R}^{T_t \times d_t}$ 分别是源模态和目标模态数据序列，$s, t \in \{L, V, A\}$。则跨模态注意力的一个单独头部定义为：

$$
\begin{aligned}
Y_t &= \mathrm{CA}_{s \to t}(X_s, X_t) \\
&= \text{softmax}\left(\frac{Q_t K_s^T}{\sqrt{d_k}}\right) V_s \\
&= \text{softmax}\left(\frac{X_t W_{Q_t} W_{K_s}^T X_s^T}{\sqrt{d_k}}\right) X_s W_{V_s},
\end{aligned}
$$

其中，$Y_t \in \mathbb{R}^{T_t \times d_v}$。如果头部有 h 个时候，跨模态注意力操作表示为 $Y_t = CA^{mul}_{s \to t}(X_s, X_t)$，$s, t \in \{L, V, A\}$ 其中 $Y_t \in \mathbb{R}^{T_t \times h d_v}$

通过激励模型关注元素之间的跨模态交互，利用源模态数据增强了目标模态。

3.3 Modality reinforcement unit

模块如图4所示：

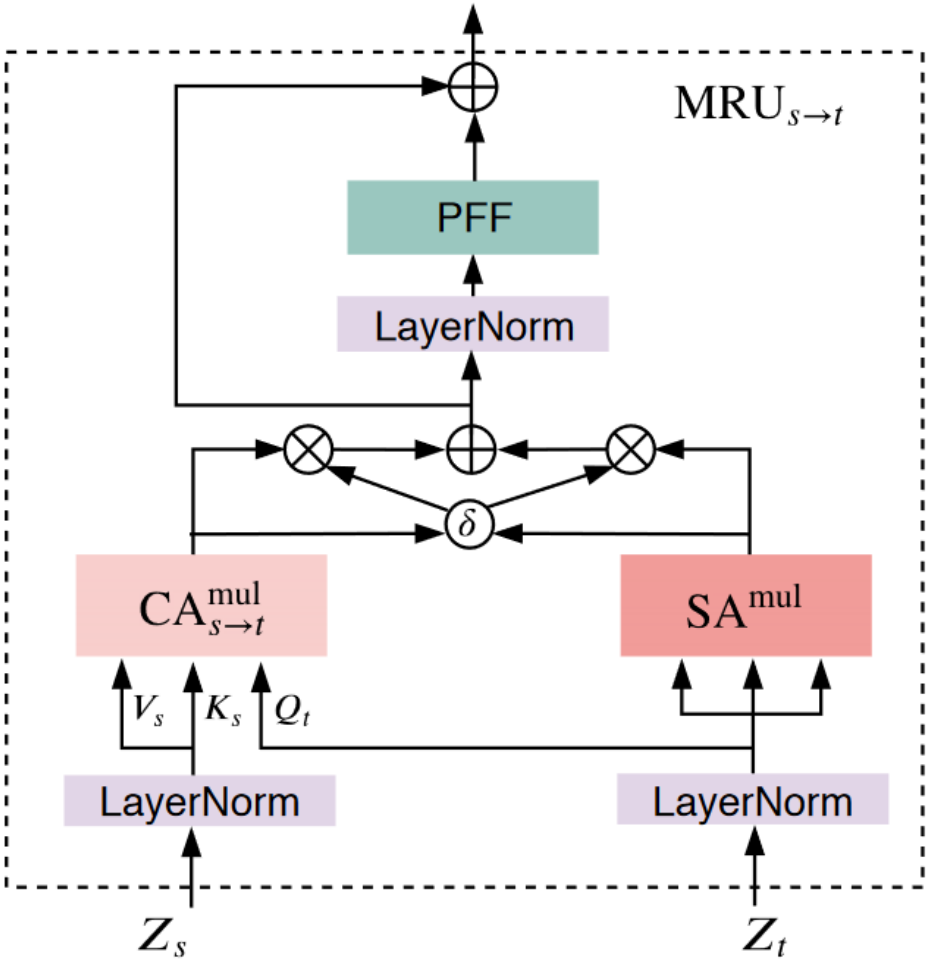

图 4: Modality Reinforcement Unit

总公式为：

$$Z_*^{[i+1]} = \text{MRU}_{C\to *}^{[i]}\left(Z_C^{[i]}, Z_*^{[i]}\right)$$

其中，$* \in \{L, V, A\}$,i 指代当前是强化层的第几层。

具体是通过两个分支，分别是自注意力和交叉注意力来通过 $MRU_{C\to *}^{[i]}$ 强化 $Z_*^{[i]}$

$$Z_{C\to *}^{[i]} = \text{CA}_{C\to *}^{\text{mul}}\left(\text{LN}\left(Z_C^{[i]}\right), \text{LN}\left(Z_*^{[i]}\right)\right)$$

$$Z_*^{[i]} = \text{SA}^{\text{mul}}\left(\text{LN}\left(Z_*^{[i]}\right)\right)$$

其中，CA^{mul} 表示多头注意力操作，LN 表示层规范操作。

然后，强化后的特征 $Z_*^{[i]} \square Z_{C\to *}^{[i]}$ 通过动态 filter 机制进行处理：

$$G_*^{[i]} = \text{sigmoid}\left(Z_*^{[i]} \cdot W_*^{[i]} + Z_{C\to *}^{[i]} \cdot W_{C\to *}^{[i]} + b_*^{[i]}\right),$$

$$Z_*^{[i]} = G_*^{[i]} \odot Z_*^{[i]} + \left(1 - G_*^{[i]}\right) \odot Z_{C\to *}^{[i]},$$

每个分支的权重是通过可以学习的 $W_*^{[i]}$ 和 $b_*^{[i]}$ 来动态决定，这样可以过滤不正确的跨模态交互产

生的信息。最后，类似于 transformer 模型，结合后的特征会经过一个前向传播层处理得到 Z_*^{i+1}，进行下一层的模态增强。

3.4 Message update module

模块如图5所示：

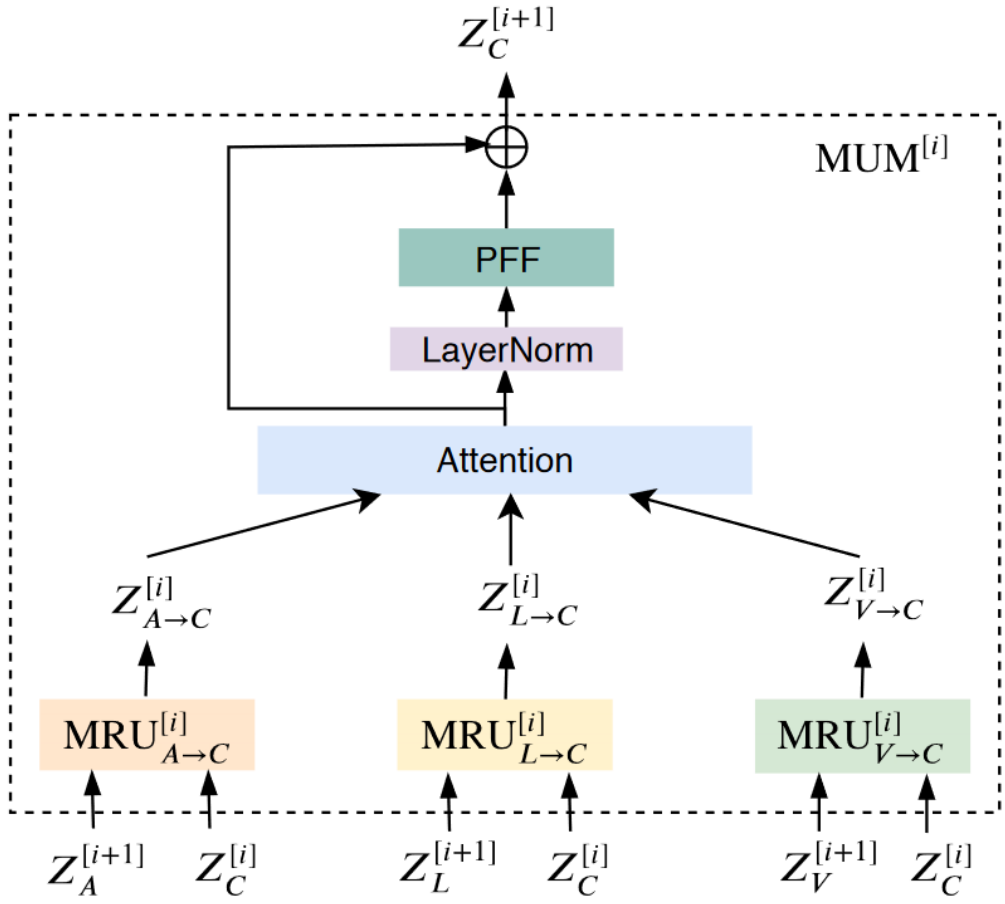

图 5: Message Update Module

使用得到的增强后的特征 $Z_{\{L,V,A\}}^{[i+1]}$ 在前面的 Modality reinforcement layer 中对共同信息 $Z_C^{[i]}$ 进行增强，公式为

$$Z_C^{[i+1]} = \mathrm{MUM}^{[i]}\left(Z_V^{[i+1]}, Z_L^{[i+1]}, Z_A^{[i+1]}, Z_C^{[i]}\right)$$

每个 message update module 由三个 modality reinforcement unit 组成，其中每个 unit 是使用一种模态增强共同信息 $Z_C^{[i]}$：

$$Z_{*\to C}^{[i]} = \mathrm{MRU}_{*\to C}^{[i]}\left(Z_*^{[i+1]}, Z_C^{[i]}\right)$$

然后，通过一个 attention layer 将 $Z_{*\to C}^{[i]}$ 融合成 $Z_C^{[i]}$。首先 reshape$Z_{*\to C}^{[i]}$ 为 $\hat{Z}_{*\to C}^{[i]} \in \mathbb{R}^{\{T_C\cdot d\times 1\}}$，然后进行下面的处理：

$$\mu_{*\to C}^{[i]} = U^T \tanh\left(W_{*\to C}^{[i]} \cdot \hat{Z}_{*\to C}^{[i]} + b_{*\to C}^{[i]}\right),$$

$$\alpha_{*\to C}^{[i]} = \frac{\exp\left(\mu_{*\to C}^{[i]}\right)}{\sum_{*\in\{L,V,A\}} \exp\left(\mu_{*\to C}^{[i]}\right)},$$

$$Z_C^{[i]} = \sum_{*\in\{V,L,A\}} \alpha_{*\to C}^{[i]} \odot Z_{*\to C}^{[i]},$$

通过注意力层，可以动态控制 $Z_{*\to C}^{[i]}$ 传入的信息和产生一个全面的共同信息。然后，将 $Z_C^{[i]}$ 通过

一个前向传播层处理得到输出 $Z_C^{[i+1]}$。

3.5 算法流程

Procedure 1 The forward propagation procedure of the modality reinforcement layers.

Input: the sequences processed by 1D temporal convolution and positional embedding: $Z_{\{L,V,A\}} \in \mathbb{R}^{T_{\{L,V,A\}} \times d}$; the layer number: D.

Output: the reinforced modalities' features: $Z_{L,V,A}$; the reinforced common message: Z_C

Initialize the modalities' features: $Z_{\{L,V,A\}}^{[0]} = Z_{\{L,V,A\}}$;

Initialize the common message:$Z_C^{[0]} = [Z_L, Z_V, Z_A]$;

i=0;

while $i \leqslant D$ **do**

  Update the modalities' features:$Z_*^{[i+1]} = \text{MRU}_{C\to *}^{[i]}\left(Z_C^{[i]}, Z_*^{[i]}\right)$;

  $i = i + 1$;

  Update the common message:$Z_C^{[i+1]} = \text{MUM}^{[i]}\left(Z_V^{[i+1]}, Z_L^{[i+1]}, Z_A^{[i+1]}, Z_C^{[i]}\right)$

end

$Z_C = Z_C^{[D+1]}$; $Z_{\{L,V,A\}} = Z_{\{L,V,A\}}^{[D+1]}$;

return Z_C *and* $Z_{\{L,V,A\}}$;

4 复现细节

4.1 与已有开源代码对比

4.1.1 引用代码

本篇论文引用的代码来源是https://github.com/yaohungt/Multimodal-Transformer

本片论文工作是基于 MulT 这篇文章开展，作者在文中针对 MulT 这篇文章提出了它里面存在的问题，然后在此基础上对其进行改进，提出更好的多模态数据融合方法，但是没有开源代码。

MulT 这篇文章开源了它的代码，本次复现是基于此开源代码进行修改。论文中作者针对 MulT 提出了改进方法，在代码中我也针对论文中提出的改进方案进行实现。

4.1.2 代码改进

Modality Reinforcement Unit

作者该论文中，引入了模态增强块（MRU）。在该模块中，通过源模态增强目标模态数据。

首先通过两个分支来进行数据处理，其中自注意力分支对目标模态进行长距离依赖关系的学习，跨模态注意力分支将目标模态和源模态数据作为输入，目标模态从源模态中获取信息。

然后处理后的两个分支的特征，通过一个动态的过滤器机制进行处理，每个分支的权重可以通过这个可学习的机制来动态决定，这样可以过滤不正确的跨模态交互产生的信息。

最后，结合后的特征会经过一个前向传播层和跳跃连接处理得到强化后的特征，进行下一层的模态增强。

```
# CA_x (sequence_length, batchsize, dimension)
CA_x = self.get_attetion(x, x_k, x_v)

# SA_x (sequence_length, batchsize, dimension)
SA_x = self.get_attetion(x)

# Dynamic filter
```

```
weight_G = torch.sigmoid(self.CA_affline(CA_x) + self.SA_affline(SA_x))
combined_x = SA_x * weight_G + CA_x * (1 - weight_G)

# PFF & residual
residual = combined_x
combined_x = self.maybe_layer_norm(1, combined_x, before=True)
combined_x = F.relu(self.fc1(combined_x))
combined_x = F.dropout(combined_x, p=self.relu_dropout, training=self.training)
combined_x = self.fc2(combined_x)
combined_x = F.dropout(combined_x, p=self.res_dropout, training=self.training)
combined_x = residual + combined_x
updated_x = self.maybe_layer_norm(1, combined_x, after=True)
return updated_x
```

Message Update Module

在前一步得到的增强后的特征 $Z_{\{L,V,A\}}^{[i+1]}$ 后，首先在前面的 Modality reinforcement layer 中对共同信息 $Z_C^{[i]}$ 进行增强。每个 message update module 由三个 modality reinforcement unit 组成，其中每个 unit 是使用一种模态增强共同信息 $Z_C^{[i]}$。

然后，通过一个 attention layer 将 $Z_{*\to C}^{[i]}$ 融合成 $Z_C^{[i]}$。首先 reshape$Z_{*\to C}^{[i]}$ 为 $\hat{Z}_{*\to C}^{[i]} \in \mathbb{R}^{\{T_C \cdot d \times 1\}}$，然后通过注意力层，可以动态控制 $Z_{*\to C}^{[i]}$ 传入的信息和产生一个全面的共同信息。

最后，将 $Z_C^{[i]}$ 通过一个前向传播层和跳跃连接层处理得到输出 $Z_C^{[i+1]}$。

```
a_t_oc = self.MRUblock(common_m, individual_a, individual_a)
l_t_oc = self.MRUblock(common_m, individual_l, individual_l)
v_t_oc = self.MRUblock(common_m, individual_v, individual_v)
common_m = self.MUMblock(v_t_oc, l_t_oc, a_t_oc)

def MUMblock(self, x_v, x_l, x_a):
   posi_0 = x_v.size(0)
   posi_1 = x_v.size(1)
   posi_2 = x_v.size(2)
   embed_dim = x_v.size(0) * x_v.size(2)

   # 先对各个模态特征进行展平处理
   x_v = x_v.view(1, x_v.size(1), embed_dim)
   x_l = x_l.view(1, x_l.size(1), embed_dim)
   x_a = x_a.view(1, x_a.size(1), embed_dim)

   # 计算每种模态的权重，控制通过量，对应于论文中的注意力层
   linear_proj = nn.Linear(embed_dim, embed_dim, bias=True)
   trans_proj = nn.Linear(embed_dim, 1, bias=False)

   single_v = trans_proj(torch.tanh(linear_proj(x_v)))
   single_v = F.dropout(single_v, p=self.embed_dropout, training=self.training)
   single_l = trans_proj(torch.tanh(linear_proj(x_l)))
   single_l = F.dropout(single_l, p=self.embed_dropout, training=self.training)
   single_a = trans_proj(torch.tanh(linear_proj(x_a)))
   single_a = F.dropout(single_a, p=self.embed_dropout, training=self.training)

   sum_atten = torch.exp(single_l) + torch.exp(single_v) + torch.exp(single_a)
   atten_v = torch.exp(single_v) / sum_atten
   atten_l = torch.exp(single_l) / sum_atten
   atten_a = torch.exp(single_a) / sum_atten

   # 合并得到全面的共同信息特征
   common_message = atten_a * (x_a.view(posi_0, posi_1, posi_2)) + atten_l * (
      x_l.view(posi_0, posi_1, posi_2)) + atten_v * (x_v.view(posi_0, posi_1,
```

```
    posi_2))

# 通过PFF层和LN对其进行处理，得到最终输出
residual = common_message
common_message = self.layerNorm(common_message)
common_message = F.relu(self.fc1(common_message))
common_message = F.dropout(common_message, p=self.relu_dropout, training=
    self.training)
common_message = self.fc2(common_message)
common_message = F.dropout(common_message, p=self.res_dropout, training=self
    .training)
common_message = residual + common_message

return common_message
```

4.2 实验环境搭建

本篇论文的运行在 RTX 3090 上，所需的环境是 Python 3.6/3.7，Pytorch (>=1.0.0) and torchvision，CUDA 10.0 或者以上。

4.3 创新点

本人工作基于论文中所提出的方法，复现了其对应的代码。

5 实验结果分析

该工作是基于 MulT 的开源代码进行复现，所以会在实验结果中对比 MulT 的运行效果，和在该代码上复现的本篇论文的运行效果。两者分别在三个数据集上进行，每个数据集都有对齐和未对齐两个版本。其中，三项指标都是指数越大，说明模型的效果越好。

Metric	Acc_7 (%)	Acc_2 (%)	$F1$ (%)
(Word Aligned)CMU-MOSI Sentiment			
MulT	31.5(40.0)	75.5(83.0)	75.3(82.8)
PMR	**35.7(40.6))** ↑	**77.1(83.6))** ↑	**77.1(83.4))** ↑
(Unaligned)CMU-MOSI Sentiment			
MulT	31.6(39.1)	71.3(81.1)	71.2(81.0)
PMR	**34.5(40.6)** ↑	**76.2(82.4)** ↑	**76.2(82.1)** ↑

图 6: CMU-MOSI RESULT

Metric	Acc_7 (%)	Acc_2 (%)	$F1$ (%)
(Word Aligned)CMU-MOSEI Sentiment			
MulT	48.9(51.8)	80.4(82.5)	80.8(82.3)
PMR	**49.2(52.5)** ↑	80.0(83.3)	80.0(82.6)
(Unaligned)CMU-MOSEI Sentiment			
MulT	50.0(50.7)	79.3(81.6)	79.8(81.6)
PMR	49.3(51.8)	**80.4(83.1)** ↑	**80.3(82.8)** ↑

图 7: CMU-MOSEI RESULT

Task	Happy		Sad		Angry		Neutral	
Metric	Acc (%)	$F1$ (%)	Acc (%)	$F1$ (%)	Acc (%)	$F1$ (%)	Acc (%)	$F1$ (%)
(Word Aligned)IEMOCAP Sentiment								
MulT	86.7(90.7)	85.3(88.6)	84.2(86.7)	82.7(86.0)	82.1(87.4)	83.1(87.0)	69.9(72.4)	69.2(70.7)
PMR	**87.2(91.3)** ↑	84.7(89.2)	83.4(87.8)	**83.2(87.0)** ↑	**84.6(88.1)** ↑	**85.0(87.5)** ↑	68.3(73.0)	67.5(71.5)
(Unaligned)IEMOCAP Sentiment								
MulT	85.6(84.8)	79.6(81.9)	79.2(77.7)	70.4(74.1)	75.8(73.9)	65.4(70.2)	55.1(62.5)	55.3(59.1)
PMR	**85.6(86.4)** ↑	79.0(83.3)	**79.4(78.5)** ↑	70.3(75.3)	**75.8(75.0)** ↑	**65.4(71.3)** ↑	**59.3(63.7)** ↑	46.9(60.9)

图 8: IEMOCAP RESULT

实验结果如上图所示。在 CMU-MOSI 数据集上，复现的效果各项都比在 MulT 原论文代码上的效果好。在 CMU-MOSI 和 IEMOCP 数据集上，都有一半或以上的指标超过在 MulT 原论文代码上的效果好。基于此，证实说明了作者在这篇论文中，针对 MulT 论文的问题提出的改进方案的确是有一定的效果，得到了更好的性能。说明在模态融合时候，利用源模态增强目标模态时候，也要利用源模态的高层语义信息，在监督下对其进行更新；而且融合时候，如果能够充分利用三种模态一起进行融合，效果会比两两模态定向融合的效果会更好。

6 总结与展望

在该工作中，首先介绍了当前多模态情绪识别的背景，并且提出了其中存在的问题。然后发展过程中前人提出了各种解决方法，从人工对齐模态再融合的方法到利用模态之间的跨模态注意力机制直接在未对齐的模态上进行融合。为此，作者提出了一个信息中心和各个模态数据之间进行信息交流融合的方法。信息中心可以通过共有信息探索所有模态之间的内在联系，从而实现更有效的多模态融合，而且，共有模态和每个模态的特征之间通过元素之间的跨模态交互进行逐渐的相互补充。逐渐补充策略中，可以有效地利用到来自于源模态地高级特征，而且在不同的实验中体现了作者提出的方法的优越性。

未来研究方向：在此工作中，多模态数据之间直接进行融合。但是不同模态之间存在着巨大差异，如果直接进行融合效果得不到很大的提高。所以可以在模态融合之前，可以先对不同的模态进行模态空间的对齐，缩小模态之间的差距，然后再进行融合。aa

参考文献

[1] LIANG T, LIN G, FENG L, et al. Attention is not Enough: Mitigating the Distribution Discrepancy in Asynchronous Multimodal Sequence Fusion[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021: 8148-8156.

[2] TSAI Y H H, BAI S, LIANG P P, et al. Multimodal Transformer for Unaligned Multimodal Language Sequences[C]//Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Florence, Italy: Association for Computational Linguistics, 2019.

[3] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need[J]. Advances in neural information processing systems, 2017, 30.

[4] TSAI Y H H, LIANG P P, ZADEH A, et al. Learning factorized multimodal representations[J]. arXiv

preprint arXiv:1806.06176, 2018.

[5] LAZARIDOU A, PHAM N T, BARONI M. Combining language and vision with a multimodal skip-gram model[J]. arXiv preprint arXiv:1501.02598, 2015.

[6] NGIAM J, KHOSLA A, KIM M, et al. Multimodal deep learning[C]//ICML. 2011.

[7] RANGANATHAN H, CHAKRABORTY S, PANCHANATHAN S. Multimodal emotion recognition using deep learning architectures[C]//2016 IEEE Winter Conference on Applications of Computer Vision (WACV). 2016: 1-9.

[8] NGUYEN D, NGUYEN K, SRIDHARAN S, et al. Deep spatio-temporal feature fusion with compact bilinear pooling for multimodal emotion recognition[J]. Computer Vision and Image Understanding, 2018, 174: 33-42.

[9] NGUYEN D, NGUYEN K, SRIDHARAN S, et al. Deep spatio-temporal features for multimodal emotion recognition[C]//2017 IEEE winter conference on applications of computer vision (WACV). 2017: 1215-1223.

[10] GAN Q, WANG S, HAO L, et al. A multimodal deep regression bayesian network for affective video content analyses[C]//Proceedings of the IEEE International Conference on Computer Vision. 2017: 5113-5122.

[11] PHAM H, LIANG P P, MANZINI T, et al. Found in translation: Learning robust joint representations by cyclic translations between modalities[C]//Proceedings of the AAAI Conference on Artificial Intelligence: vol. 33: 01. 2019: 6892-6899.

[12] WANG Y, SHEN Y, LIU Z, et al. Words can shift: Dynamically adjusting word representations using nonverbal behaviors[C]//Proceedings of the AAAI Conference on Artificial Intelligence: vol. 33: 01. 2019: 7216-7223.

[13] ZENG Z, TU J, PIANFETTI B, et al. Audio-visual affect recognition through multi-stream fused HMM for HCI[C]//2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05): vol. 2. 2005: 967-972.