

中国科学院大学

University of Chinese Academy of Sciences

## 情感计算大作业

成员 1: 李一鸣 202228013229030

成员 2: 陈国鑫 2022E8013282125

成员 3: 张 兆 202228013229029

代码开源地址: <https://github.com/zhangzhao219/UCAS-2023-Spring-Homework/tree/Multimodal-Sentiment-Analysis>

2023 年 06 月

# 目录

1	引言 .....	1
1.1	任务概览 .....	1
1.2	数据描述 .....	1
1.3	评价指标 .....	2
2	基于音频模态的语音情感识别 .....	2
2.1	数据处理 .....	2
2.2	网络架构 .....	3
2.3	性能提升 .....	4
2.3.1	预训练模型 .....	4
2.3.2	数据增强策略 .....	6
3	基于文本模态的语音情感识别 .....	7
3.1	数据处理 .....	7
3.2	传统方法 .....	7
3.2.1	线性回归 .....	8
3.2.2	Logistic 回归 .....	9
3.2.3	朴素贝叶斯 .....	10
3.2.4	决策树 .....	11
3.2.5	随机森林 .....	11
3.2.6	支持向量机 .....	12
3.3	深度学习方法 .....	13
3.3.1	TextCNN & BiLSTM .....	13
3.3.2	预训练模型 .....	14
3.3.3	Prompt Tuning .....	15
3.4	文本情感分析实验 .....	16
4	基于多模态的语音情感识别 .....	17
4.1	网络架构 .....	17
4.2	融合策略 .....	17
4.2.1	前期融合 .....	17
4.2.2	注意力融合 .....	18
5	多模态情感识别系统 .....	19
6	参考文献 .....	22

# 1 引言

## 1.1 任务概览

语音作为语言的第一属性，在语言中起决定性的支撑作用，它不仅包含了说话人所要表达的文本内容，也包含说话人所要表达的情感信息。而情感则与人态度中的内向感受、意向具有协调一致性，是态度在生理上一种较复杂而又稳定的评价和体验，是一种综合了人类行为、思想和感觉的现象。

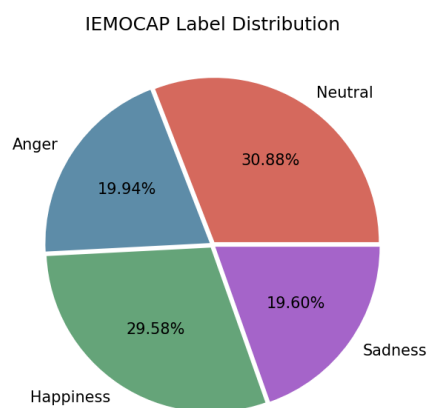
语音情感是指从语音信号中获取相应的情感信息，语音情感识别是计算机对人类上述情感感知和理解过程的模拟，利用计算机分析情感，提取出情感特征值，并利用这些参数进行相应的建模和识别，建立特征值与情感的映射关系，最终对情感进行分类。

语音情感识别是人机情感交互的关键<sup>[1]</sup>，对语音情感的有效识别能够提升语音的通俗性，使各种智能设备最大限度理解用户意图，提高机器人人性化水平，从而更好地为人类服务。

## 1.2 数据描述

本文选择南加州（USC）语音分析和解释实验室（SAIL）收集的交互式情感二元运动捕捉数据库 IEMOCAP<sup>[2]</sup>进行语音情感识别探究。该数据集包含大约十二个小时数据，记录了来自 10 位演员的二元会话信息，它们被要求在假设场景中即兴对话，旨在引发特定的情绪（快乐、愤怒、悲伤、沮丧和中性等状态），此外，数据集当中还包含演员的面部、手部动作等模态信息。

考虑到本文致力于探索语音情感识别，并考虑到平衡数据类别分布等因素，本文选择从数据库当中抽取音频信息（wav 文件夹下）及转录的文本信息（transcription 文件夹下），由于某些类别存在较为严重的长尾现象，因此，本文选择了“neu”（中性）、“sad”（悲伤）、“hap”快乐、“ang”愤怒等四个类别，并将“exc”（激动）类别划分为“hap”类别。划分后的数据集包含 5531 条数据，数据分布如下图所示：



1 数据集标签分布情况

此外，我们将数据集按照 8：1：1 划分训练集、验证集与测试集。

### 1.3 评价指标

IEMOCAP 情感识别任务本质是一类多类别分类任务，本文采用了学界相对常用的加权准确率分数（Weighted Accuracy, WA）及非加权准确率分数（Unweighted Accuracy, UA）作为评价指标。其中加权准确率分数用于评估模型在类别不平衡的数据集上的性能，可以更加充分地考虑各类情感的识别效果。

## 2 基于音频模态的语音情感识别

### 2.1 数据处理

IEMOCAP 数据集音频文件以每一场景下的对话为单位进行录制，每一音频文件对应一份标注文件，标注了该音频以第{start}秒为开始时间，第{end}秒为结束时间的音频片段及其对应的情感类别标签。考虑到神经网络不易处理时间长短不一的音频，本文先按照开始、结束时间戳截取音频片段，并将所有音频的长度统一为 10s：对于少于 10s 的音频，将其重复播放至 10s 止；对于大于 10s 的音频，将其截取至 10s。

对于音频模态，采用对数梅尔频谱图作为输入特征，经预加重、分帧、加窗、短时傅里叶变换、梅尔滤波、对数运算等操作，得到最终的声学特征。频谱图的计算依赖 hop size（决定帧的数目）、window length（决定一帧的窗口内有多少采样点）、mel bins（决定特征维数）等参数，参数依据模型的选择略有不同，将

在下文进行详细说明。此外，计算整个数据集音频在频域维度及时域维度的归一化参数进行特征归一化。

## 2.2 网络架构

我们首先设计了一个 CRNN 网络作为基准网络，其通过 2D 卷积层提取频谱图的特征，并通过池化操作逐步缩小特征图尺寸，直至将频域维度特征由频谱图的 mel bins 压缩至 1，得到的时序特征经过 Bi-GRU 进一步加强时序关联后，最后一个隐状态被送入全连接分类头，通过 Softmax 激活得到音频属于 4 个类别的概率，CRNN 的网络架构如下图所示，在图中，我们标注了网络各个层次输出的张量尺寸，方便读者更好地理解模型设计及音频分类任务处理的大致流程，

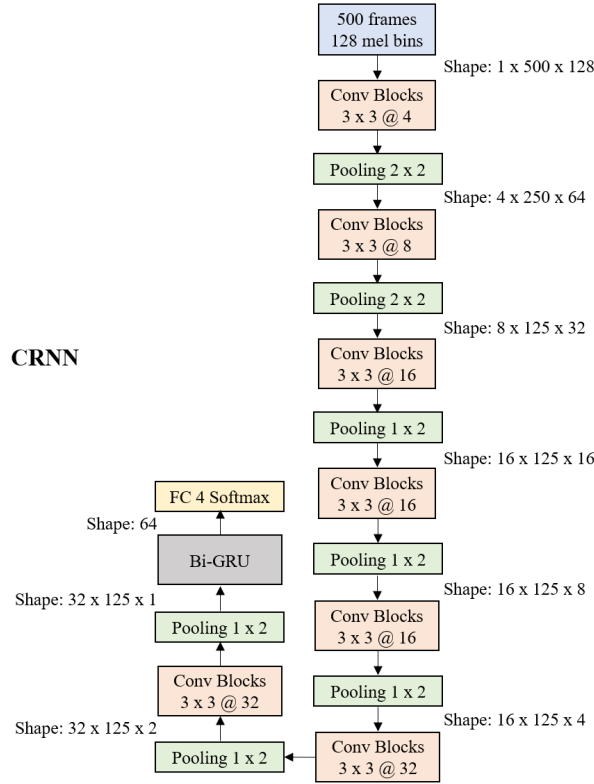


图 2 CRNN 网络架构示意图

网络输入频谱图的 hop size 为 320，window length 为 1024，sample rate 为 16k。模型采用交叉熵损失函数进行训练，优化器为 AdamW，学习率设置为 0.0001，训练迭代周期为 20 epochs。我们保存了在验证集上 UA+WA 最高的模型并在测

试集上进行测试。除输入频谱图外的其它训练设置在 2.3 小节及第 4 章均保持一致。最终，基准模型 CRNN 在验证集上的 UA 为 0.556，WA 为 0.552，测试集上的 UA 为 0.554，测试集 WA 为 0.571。

## 2.3 性能提升

### 2.3.1 预训练模型

近年来，在大规模无标签数据集上进行预训练，再在下游数据集上进行微调的策略逐渐成为自然语言处理、计算机视觉等领域的主流学习范式。基于此，本文考虑使用在 AudioSet 数据集<sup>[3]</sup>上进行音频预训练的网络模型在 IECOMAP 数据集上进行相应下游任务的微调，所考虑的两类模型分别为 CNN 类的网络模型 PANNS<sup>[4]</sup>及 Transformer 类的网络模型 HTSAT<sup>[5]</sup>：

(1) PANNS，为 AudioSet 预训练的 CNN 模型，上游堆叠而成的 ConvBlocks（Conv + BN + ReLU + Pooling）提取音频特征，并在时域与频域进行下采样，通过 Global Temporal Pooling 将时域特征汇聚为整个音频的表征向量，再将该表征向量送入分类头进行分类。本文选择了论文提供的 CNN10、CNN14 两类架构，相较于前者，后者的参数规模更大，二者的网络架构如下图所示：

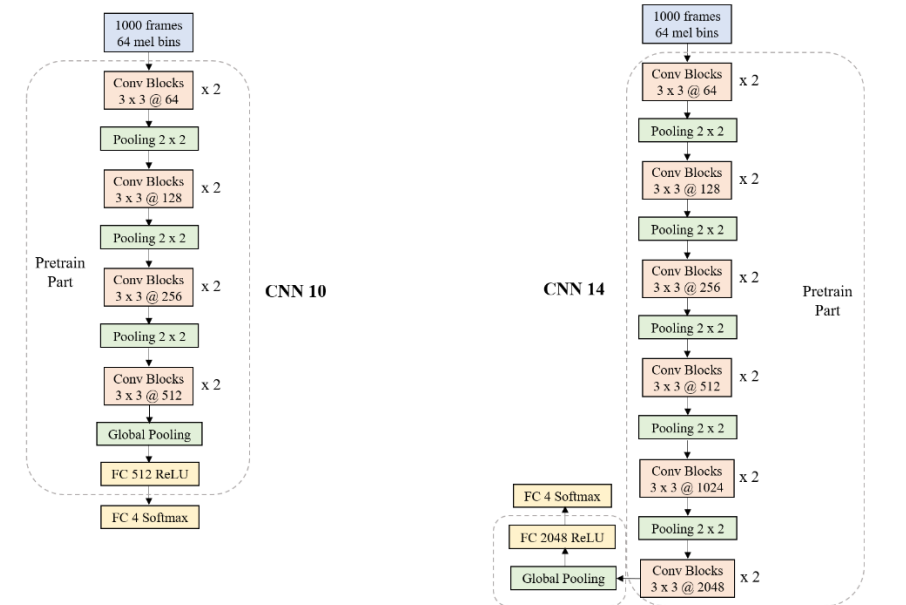


图 3 CNN10 与 CNN14 网络架构示意图

其中, CNN10 的 hop size 为 320, window length 为 1024, sample rate 为 32k; CNN14 的 hop size 为 512, window length 为 160, sample rate 为 16k, 二者均采用 64 个 mel bins。

(2) HTSAT, 为 AudioSet 预训练的 Transformer 模型, 先通过 Patch Embed 将频谱图转成 Patch 序列, 上游堆叠而成的 SwinBlocks 提取层次化的帧级别的音频特征, 之后帧级别特征被送入分类器中得到帧级别概率, 再由平均池化层得到音频片段级别的概率, HTSAT 的特征参数同 CNN10, 其网络架构示意图如下, 其中 T、F 分别代表输入频谱图在时域、频域的尺寸, P 为 Patch 化时的 Patch 大小, D 为 Patch 的特征维度, 最后输出的 Label Prediction 即为 C 个标签的置信分数, HTAST 的网络架构示意图如下:

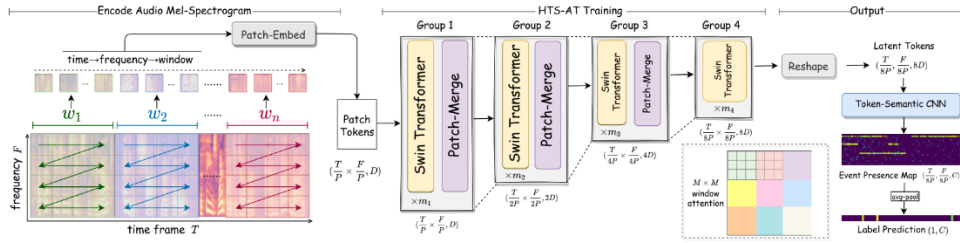


图 4 HTAST 网络架构示意图

我们分别应用上述预训练模型, 比较是否加载预训练权重对于模型性能的影响如下表所示 (汇报了模型在验证集及测试集的各项指标, 指标计算在十次试验下取平均值),

表 1 大规模预训练对于模型性能的影响

Model	Pre-train	Val UA	Val WA	Test UA	Test WA	Params
CRNN		0.556	0.552	0.554	0.571	0.4M
CNN10		0.578	0.588	0.587	0.634	6.1M
CNN10	✓	0.654	0.651	0.628	0.659	6.1M
CNN14		0.452	0.417	0.466	0.423	84.8M
CNN14	✓	0.685	0.682	0.636	0.661	84.8M
HTSAT		0.554	0.548	0.510	0.549	29.2M
HTSAT	✓	0.659	0.664	0.631	0.658	29.2M

从上表可以看出, 当不加载预训练模型时, 参数量较小的 CRNN、CNN10 均

可以在该任务上取得不错的结果，而参数量极大的 CNN14 则由于过拟合表现极差。当加载预训练参数后，CNN10、CNN14、HTSAT 等模型的性能均得到了较大提升，此时，参数量较大的 HTSAT、CNN14 模型的表现要略胜于 CNN10 模型，其中，以 CNN14 模型的分类表现最佳。

### 2.3.2 数据增强策略

为了进一步提升语音情感识别系统的稳定性并扩充数据规模，本文借鉴了两类音频数据增强策略：SpecAugment<sup>[6]</sup>与 FilterAugment<sup>[7]</sup>，前者主要针对频谱图的时域与频域信息进行随机掩膜，后者对特定频段的信息进行扰动，增强的超参数选择与原文保持一致。除此之外，本文采用 Mixup<sup>[8]</sup>来鼓励模型做出插值一致性的预测，具体地，对于两样本  $x_1$ 、 $x_2$ ，假设其对应标签为  $y_1$ 、 $y_2$ ，交叉熵损失函数为  $\mathcal{L}$ ，混合系数  $\lambda$  采样于 Beta 分布，模型映射以  $F$  表示，则 Mixup 损失函数为：

$$\mathcal{L}_m = \lambda \mathcal{L}(F(\lambda x_1 + (1 - \lambda)x_2), y_1) + (1 - \lambda) \mathcal{L}(F(\lambda x_2 + (1 - \lambda)x_1), y_2)$$

我们分别应用上述数据增强策略于上一小节采用的 CNN10 模型（模型加载预训练参数），比较其对于模型性能的影响如下表所示（汇报了模型在验证集及测试集的各项指标，指标计算在 10 次实验下取平均值）：

表 2 数据增强策略对于模型性能的影响

Model	SpecAug	FilterAug	Mixup	Val UA	Val WA	Test UA	Test WA
CNN10				0.654	0.651	0.628	0.659
CNN10	✓			0.672	0.657	0.635	0.661
CNN10		✓		0.651	0.652	0.608	0.640
CNN10			✓	0.654	0.654	0.630	0.671
CNN10	✓		✓	0.670	0.661	0.638	0.664
CNN10		✓	✓	0.638	0.648	0.605	0.646

从上表看出，SpecAugment 与 Mixup 两类增强方法均可以提升模型性能，而将 FilterAugment 应用于模型训练时则会引起模型在测试集上的性能较大幅度的下降。



## 3 基于文本模态的语音情感识别

### 3.1 数据处理

IEMOCAP 数据集文本数据是在每一场景下音频的内容进行转录而得到的，包括两个人物之间的对话内容，包括语音转写、标点符号、发音注释等信息。这些文本数据通常包含了演员的愤怒、快乐、悲伤等情感状态。摘取第二章中音频文件处理后对应的文本片段作为文本数据。

对于文本来说，与结构化的数值数据不同，首先要进行一些预处理操作，提高文本数据的完整性、正确性和一致性，从而提高模型训练和预测的效率。我们在将文本输入到传统机器学习模型之前进行了一系列的文本预处理操作：

（1）数据清洗。从原始的文本数据中删除标签为空或者标签不合理的数据，同时过滤一些明显长度过短的文本。

（2）去除停用词。将一些经常出现在文本中但是语义不强的词汇，如“the”等进行去除，防止其数量过多稀释了重要的模型特征权重，影响模型性能。在本次任务中使用百度的停用词表进行过滤。

（3）保留某些标点符号。一般来说，对于文本的预处理是要移除掉所有的标点符号的，且标点符号也可以作为分词的一个依据。但是对于情感识别任务来说，问号、感叹号等标点符号在一定程度上反映了当前人的一些心理状态。因此需要保留这些比较特殊的标点符号作为文本的一种类别特征。

### 3.2 传统方法

除了进行一系列的数据预处理对文本数据进行清洗外，还需要将文本的特征提取出来，构造模型可以识别的数值型输入，才可以使用机器学习模型进行情感识别。

将文本转换为数字特征的方式有很多，例如词袋模型，n-gram 模型，词嵌入

等等。在本次任务的传统方法部分，我们使用了 TF-IDF 提取文本的特征。

TF-IDF 的主要思想是：如果一个词或短语在一篇文章中出现的频率高，同时在其他文章中很少出现，则认为此词或短语对该文章具有很强的相关性，具有很好的区分度。这体现了词语在语义上的重要性，是信息检索与文本挖掘中常用的评价指标之一。

TF-IDF 通过以下两个指标来衡量一个词语的重要性：

1. 词频 (Term Frequency, TF) :某个词语在一篇文档中出现的次数。词频可以衡量一个词语在该文档中的重要程度，出现次数越多，该词可能越重要。

2. 逆文档频率 (Inverse Document Frequency, IDF) : 某个词语在整个文档集中的词频，用来衡量该词语在所有文档中的常见程度。如果一个词在很多文档中出现，那么该词的区分度较小，IDF 值会较小。相反，如果一个词只在少数文档出现，该词的 IDF 值会较大，说明其具有很好的区分度。

TF-IDF 的值由这两个指标共同决定，通常使用以下公式计算：

$$TF - IDF = TF * IDF = (\text{词频}) * \log(\text{总文档数} / \text{包含该词的文档数})$$

其中，词频 TF 衡量词语在某个文档中的重要性，IDF 通过惩罚在许多文档中出现的常用词，而提高只在少数文档中出现的词的权重，起到平衡的作用。

TF-IDF 可以有效地衡量一个词语的重要性。由于它考虑了词语在该文档和所有文档中的频率信息，所以可以过滤掉常见但语义较弱的词，更强调比较具有区分度的词语。这使得 TF-IDF 特别适用于文本的特征提取与关键词识别等任务。

提取 TF-IDF 特征后，我们使用经典的六种机器学习算法进行文本模态的情感识别。

### 3.2.1 线性回归

线性回归是一种最简单的机器学习算法，假设特征与标签之间存在线性关系。

线性回归的模型表达为：

$$y = w_1x_1 + w_2x_2 + \dots + w_nx_n + b$$

其中， $x_1$ 到 $x_n$ 是 $n$ 个特征， $w_1$ 到 $w_n$ 是每个特征对应的权重， $b$ 是模型的偏置。对任意一个特征 $x_i$ ，如果其权重 $w_i$ 为正，那么 $x_i$ 对 $y$ 有正相关影响；如果 $w_i$ 为负，那么 $x_i$ 对 $y$ 有负相关影响。权重的大小表示该特征对模型预测的重要性。对于本次的任务来说，特征即为通过 TF-IDF 提取的文本特征，标签即为各类别的实际标签。

线性回归旨在找到一组权重 $w$ 和偏置 $b$ ，可以使得模型对训练数据的预测与真实标签 $y^*$ 尽量接近。这是一个优化问题，常用的损失函数是均方误差，通过梯度下降等方法不断调整 $w$ 和 $b$ 的值以最小化损失函数，从而得到最优的模型。

线性回归方法比较简单，容易理解，且计算效率高；缺点是假定了线性关系，因此无法处理非线性问题，学习能力有限，且容易过拟合，对异常值比较敏感。对于本次任务来说，由于标签是离散的值，因此只能将经过模型后输出的值与标签绝对距离进行比较，实际上线性回归并不适用于本次的任务。

### 3.2.2 Logistic 回归

逻辑回归是一种广泛使用的机器学习算法，用于解决二分类问题，也可以进行扩展从而解决多分类问题。

逻辑回归模型表达为：

$$y = \text{sigmoid}(w_1x_1 + w_2x_2 + \dots + w_nx_n + b)$$

其中， $x_1$ 到 $x_n$ 是 $n$ 个特征， $w_1$ 到 $w_n$ 是每个特征对应的权重， $b$ 是模型的偏置。

逻辑回归训练的目的在于最大化训练数据被正确分类的概率。它采用最大似然估计法，通过梯度上升等优化方法不断更新权重 $w$ 和 $b$ 的值，使模型对训练数据的分类结果与真实标签 $y^*$ 尽可能吻合。

多类逻辑回归模型的原理如下：

1. 假设类别之间是互斥的，即每个实例只能属于一个类别。对每个类别 $i$ ，引入 $y_i$ 表示该实例属于类别 $i$ 的概率。

2. 为每个类别构建一个逻辑回归模型,预测属于该类别的概率。具体而言:

$$p(y_i = 1|x) = \exp(w_i^T x) / (1 + \exp(w_i^T x))$$

这里 $w_i$ 是类别 $i$ 对应的权重向量。

3. 对新输入 $x$ , 在每个逻辑回归模型中计算属于对应类别的概率, 然后选择概率最大的类别作为预测类别。

多类逻辑回归的优点在于原理简单, 易于理解和实现, 并且可以在不加修改的情况下, 扩展到更多类别。缺点是无法学习类别之间的相关性, 且计算量比较大, 训练代价高。

### 3.2.3 朴素贝叶斯

朴素贝叶斯是一种简单实用的机器学习算法, 用于解决分类问题。原理如下:

1. 假设特征之间相互独立。这是朴素贝叶斯算法的核心假设, 该假设简化了模型使其易于计算。

2. 根据贝叶斯定理计算后验概率。对于某个类别 $C$ 和特征向量 $X$ , 贝叶斯定理为:

$$P(C|X) = P(X|C)P(C) / P(X)$$

其中 $P(C|X)$ 是后验概率,  $P(X|C)$ 是联合概率,  $P(C)$ 是先验概率

3. 构建分类器。对于新输入 $x$ , 计算它属于每个类别的后验概率, 然后预测类别为先验最大的那个类别。

朴素贝叶斯模型表达为:

$$P(C|x_1, x_2, \dots, x_n) = P(C) * P(x_1|C) * P(x_2|C) * \dots * P(x_n|C) / P(x_1, x_2, \dots, x_n)$$

其中 $x_1$ 到 $x_n$ 是 $n$ 个特征,  $P(C)$ 是类别 $C$ 的先验概率,  $P(x_i|C)$ 是特征 $x_i$ 在类别 $C$ 下的条件概率。

朴素贝叶斯的“朴素”是因为它假定特征相互独立。这个假设简化了模型, 使其易计算, 但也限制了其表达能力。然而即便特征之间存在依赖, 朴素贝叶斯仍然表现良好。这是因为贝叶斯定理在分类效果上更为重要。朴素贝叶斯通过训练统计特征-类别共现频次, 获得先验概率和条件概率。预测阶段直接应用贝叶斯定

理,以获得每个类别的后验概率。

朴素贝叶斯的优点是计算简单,易理解且易实现。缺点是其对属性独立的假设限制了模型的表达能力,且对小训练集表现不佳。

### 3.2.4 决策树

决策树是一种流行的机器学习算法,用于解决分类和回归问题。其原理如下:

1. 以树形结构表示决策过程。决策树由根节点、内部节点(非叶子节点)和叶子节点组成。
2. 根节点代表完整训练数据集。内部节点表示数据的某个特征,每个子节点代表该特征的一个取值。
3. 叶子节点代表数据集的分类结果或回归结果。分类树的叶子节点包含类别标签,回归树的叶子节点包含连续值。
4. 决策树通过递归地对数据进行切分,直到达到停止条件,逐步生成。
5. 对新数据实例,从根节点开始,依据其特征取值递归地向下移动,最终到达叶子节点,获得预测结果。

决策树主要学习算法有: ID3、C4.5、CART。其中 CART 构建决策树通过贪心算法,按照特征选择和数据切分的方式生成二叉树。其具体步骤为:

1. 选择最优特征和切分点: 计算各个特征和各个切分点的代价,选择代价最小的特征和切分点进行切分。
2. 数据切分: 将节点的数据切分成子节点,根据最优切分点的取值将数据划分到左子节点或右子节点。
3. 生成子节点: 为左右子节点重复以上步骤,直到达到停止条件。
4. 剪枝: 采用 CART 算法可以剪去一些子节点,防止决策树模型过拟合。

决策树算法结果易于理解,对异常值不太敏感。但可能导致过拟合,学习结果随训练数据的变化而变化大。决策树是最简单,使用最广的机器学习算法之一。

### 3.2.5 随机森林

随机森林是一种流行的机器学习算法，用于解决分类和回归问题。它的原理如下：

1. 随机森林由多个决策树组成。每个决策树独立进行训练，最后将多个决策树的结果综合得到最终输出。
2. 随机森林使用 bagging 的思想，通过有放回地从训练数据中抽样获得子数据集来训练每个决策树。
3. 构建每个决策树时，随机选取特征子集。这增加了随机森林的多样性，避免过拟合。
4. 对新输入实例，将其输入到所有的决策树，并统计各类结果出现的次数，最终预测出现次数最多的类。

由于数据、特征的随机抽取，每棵决策树不同，随机森林具有较强的多样性，从而减少了过拟合的风险。随机森林准确性高、对异常值鲁棒、易于理解和实现。缺点是计算开销大、训练时间长。

### 3.2.6 支持向量机

支持向量机（SVM）是一种用于解决分类和回归问题的机器学习算法。它的原理如下：

1. SVM 通过寻找超平面将不同类的训练数据分开，并且使得该超平面与最近的训练数据点之间的距离尽可能大。
2. SVM 使用核技巧隐式地将数据映射到高维空间，使数据在高维空间线性可分，而在原始空间不可分。常用有线性核、多项式核和 RBF 核等。
3. SVM 使用最大间隔分类器，寻找能将不同类的训练数据分开的超平面，使得该超平面与最近的数据点的距离最大。
4. SVM 因此得到一个分类器，能够对新数据判断类别。对于非线性可分数据集，SVM 可以通过核技巧映射得到其在高维空间的表现，仍然使用最大间隔分类器构建超平面，从而实现非线性分类。

SVM 理论基础坚实、泛化能力强，通过核技巧可处理非线性问题，对中间空

间数据不敏感。缺点是核的选择和超参数的调优比较困难。

## 3.3 深度学习方法

### 3.3.1 TextCNN & BiLSTM

文本情感分类是一项重要的自然语言处理任务，常常需要利用深度学习技术进行处理。在这些深度学习技术中，TextCNN<sup>[9]</sup>和双向长短期记忆网络 (BiLSTM)<sup>[10]</sup>已被证明可以有效地应用于文本情感分类任务。

(1) TextCNN: TextCNN 是一种具有多通道输入、多尺度卷积和全局最大池化等特征的卷积神经网络模型。它能够充分利用文本的局部和全局信息，并且能够高效地捕捉文本中的关键特征。因此，在情感分类任务中，TextCNN 能够有效地提高分类准确率和鲁棒性，并且具有良好的效率和可扩展性。TextCNN 的网络架构示意图如下：

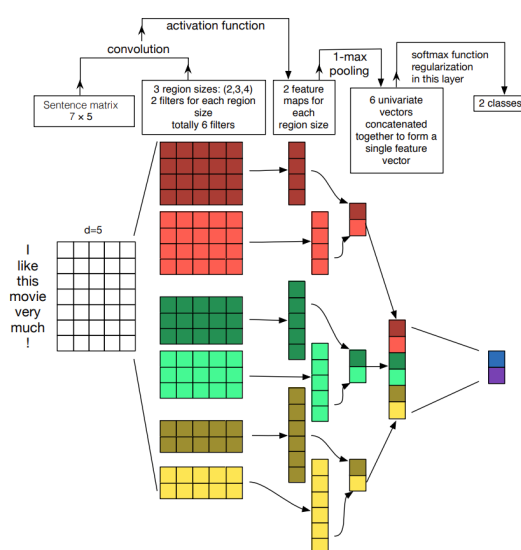


图 5 TextCNN 模型示意图

(2) BiLSTM: BiLSTM 是一种基于循环神经网络 (RNN) 的模型，在处理时序数据上表现出色。BiLSTM 能够通过正向和反向传递信息，充分利用文本的上下文信息，并且具有优秀的建模能力。在情感分类任务中，BiLSTM 能够更好地处理长文本序列，并且对上下文信息的利用更加精细和全面，因此可以取得更

好的分类效果。BiLSTM 的网络架构示意图如下：

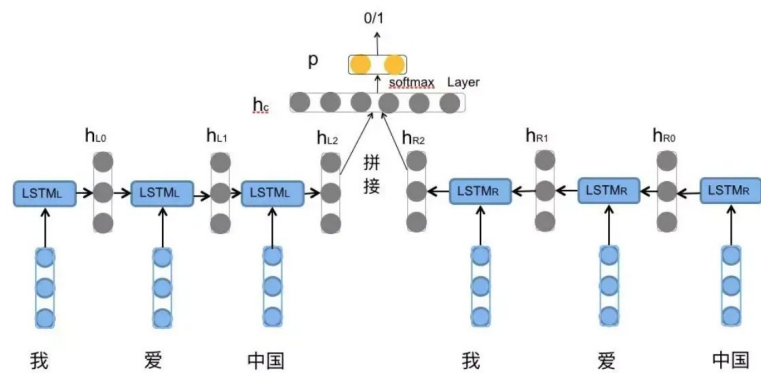


图 6 BiLSTM 模型示意图

TextCNN 通过预训练的词向量模型得到文本数据的词向量矩阵，在本文中包含中文和英文两种语言的数据集，分别采用“tencent-ailab-embedding-zh-d200”和“glove.42B.300d”作为预训练词向量模型。

### 3.3.2 预训练模型

预训练模型是指在大规模无标签语料库上进行预训练，学习到通用的语言知识和表征能力，然后针对具体的下游任务（如文本分类、命名实体识别等），进行有监督的微调。

Bert<sup>[11]</sup>是一种基于 Transformer<sup>[12]</sup>架构的预训练语言模型，被广泛应用于自然语言处理任务中。在情感分类任务中，Bert 可以利用大规模语料库的上下文信息和语义特征，并且能够针对不同类型的情感分类任务进行微调，从而取得较好的分类效果。具体来说，Bert 首先使用无监督的方式进行预训练，通过掩码语言模型和下一句预测等任务，学习到了丰富的语言知识和表征能力。在情感分类任务中，Bert 能够充分利用文本的上下文和语义信息，对于长文本和复杂情感分类任务有很好的应用效果。此外，还可以通过多层抽取和特征融合等技术，进一步提高分类效果。Bert 的网络架构示意图如下：



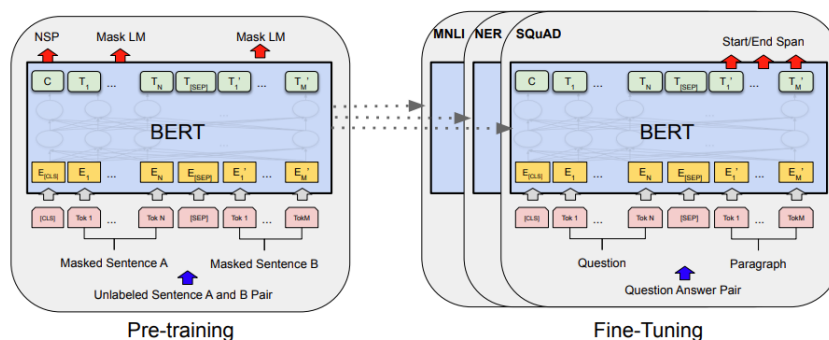


图 7 Bert 模型示意图

在 Bert 中，每一个文本序列都会被添加特殊标记[CLS]和[SEP]，其中[CLS]标记表示整个文本序列的“类别”。在句子级别的情感分类任务中，可以将文本序列作为输入，同时在序列的开头添加[CLS]标记，并将该位置的输出向量作为整个句子的表示向量。在本文实验中，我们尝试对不同规模的 Bert 进行微调，在多个数据集上测试其效果。

### 3.3.3 Prompt Tuning

Prompt 技术<sup>[13]</sup>是一种基于语言模型的自然语言处理技术，它通过在输入文本中插入提示语（Prompt），来引导模型生成与指定任务相关的输出。Prompt 技术可以帮助解决少样本学习、零样本学习等问题，并且具有较好的可解释性和易用性。其中，Soft Prompt<sup>[14]</sup>是一种特殊的 Prompt 技术，它采用软提示方式对模型进行微调。与硬提示不同，Soft Prompt 不会强制指定任务的确切格式和内容，而是使用一组关键词或短语来引导模型生成相关的输出。具体来说，在训练时，Soft Prompt 可以将预定义的提示词或短语加入到输入序列中，作为模型学习的一部分。在推理时，模型将这些提示词或短语视为一种信号或信息，从而更好地完成特定的任务。Soft Prompt 的主要优点是其灵活性和可扩展性。相比硬提示，它能够更准确地匹配输入数据的多样性，从而提高模型的泛化能力。其中较为经典的 Soft Prompt 方法为 P-Tuning v2<sup>[15]</sup>。与之前的 Soft Prompt 方法相比，P-Tuning v2 引入了多层 Prompt 的机制，同时采用了强化学习算法来优化 Prompt 的生成，提高了模型的适应性和泛化能力。P-Tuning v2 的模型示意图如下：

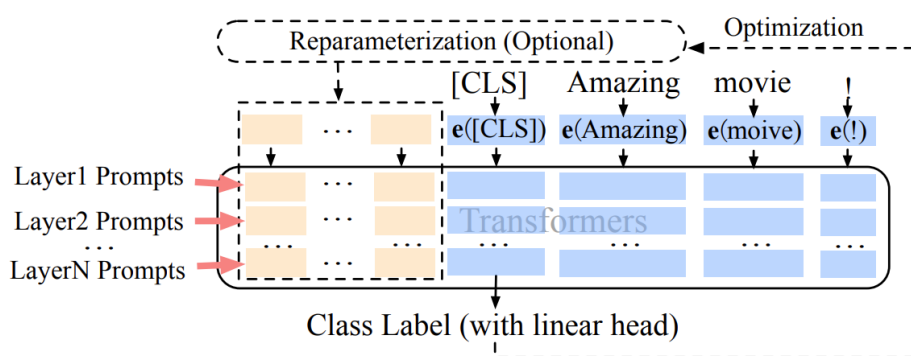


图 8 P-Tuning v2 的模型示意图

在文本实验中，我们设置 Prompt token 数为 20，并且采用多层感知机 (MLP) 进行重参数化，提升模型效果。

### 3.4 文本情感分析实验

通过传统方法和深度学习方法进行文本模态的情感识别的结果如下：

表 3 文本模态的情感识别结果

Method	Val UA	Val WA	Test UA	Test WA
Linear	0.349	0.361	0.451	0.423
Logistic Regression	0.577	0.569	0.628	0.637
Naïve Bayes	0.573	0.549	0.588	0.575
Decision Tree	0.436	0.447	0.518	0.523
Random Forest	0.380	0.404	0.511	0.489
SVM	0.575	0.562	0.612	0.620
TextCNN	0.634	0.626	0.626	0.631
BiLSTM	0.639	0.634	0.634	0.648
bert-base	0.712	0.706	0.706	0.733
bert-large	0.721	0.724	0.724	0.730
base-P-Tuning v2	0.684	0.684	0.684	0.704
large-P-Tuning v2	0.720	0.724	0.724	0.736

从测试结果可以看出，传统机器学习方法中 Logistic Regression 和 SVM 比较适合，其中 Logistic Regression 对大规模数据有很好的扩展性，SVM 的模型泛化能力强，两者在测试集上达到了 0.62 左右的性能。但深度学习方法可以达到更高的识别精度，最高的精度与机器学习方法相比要高出 0.1 左右，这是因为深度

学习模型可以自动学习文本的语义特征表示，而不再依赖手工提取的特征。相比之下，传统机器方法需要人工选取特征，依赖于人工经验和专业知识。

但是，深度学习模型在其他方面也有不足之处，深度学习模型通常具有较高的模型复杂度，需要大数据集进行训练，计算资源要求也更高，而传统机器学习模型相对简单，计算资源需求较小，且具有更好的解释性。

综上所述，深度学习方法在文本情感识别任务上具有更高的识别精度和更好的泛化能力等优势。但其模型较为复杂，需要大量数据和计算资源。相比之下，传统机器学习方法更加简单和可解释，但识别效果有限，且需要专业知识进行特征工程。

## 4 基于多模态的语音情感识别

### 4.1 网络架构

采用双流网络架构，音频编码器由第 2 章 CNN 模型（如 CNN10 模型）构成，语言编码器由第 3 章 BERT 模型（如 BERT-Tiny 模型）构成，二者提取的特征经过特征投影及融合机制融合后送入分类头进行分类。具体地，考虑前期融合、注意力机制融合两类融合策略。

### 4.2 融合策略

#### 4.2.1 前期融合

CNN 音频编码器通过堆叠的卷积层及最后的 Temporal Pooling 层得到音频的  $D$  维表征，BERT 语言编码器的 CLS Token Embedding 也可以看作文本的  $C$  维表征，二者分别经过各自的 Projector（线性层）将表征投影到  $D'$  维空间，再将投影后的向量进行拼接，作为多模态表征送入分类头中，前期融合的原理示意图如下：

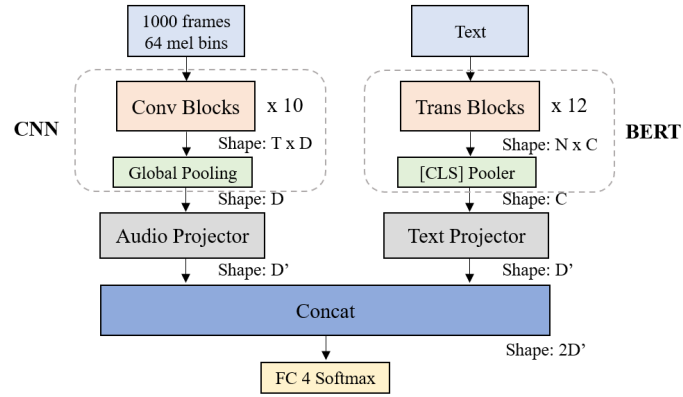


图 9 多模态特征前期融合原理示意图

#### 4.2.2 注意力融合

CNN 音频编码器通过堆叠的卷积层得到帧级别的特征（不经过 Temporal Pooling），BERT 语言编码器的 CLS Token Embedding 也可以看作文本的 C 维表征。此时，将 BERT 输出的 CLS Token Embedding 作为交叉多头注意力的 Q，而将 CNN 输出的帧级别特征作为 K、V，此时相当于利用文本特征对音频模态的帧级别特征进行加权而不是直接取平均或者最大值，注意力融合的原理示意图如下：

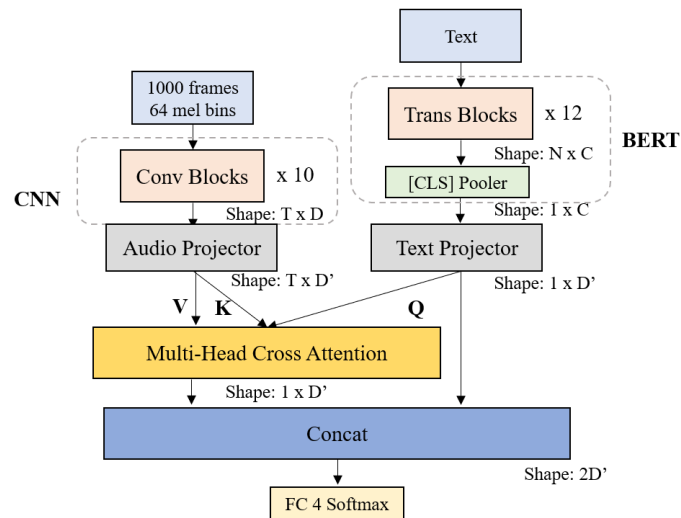


图 10 多模态特征注意力融合原理示意图

利用第 2、3 章提出的单模态模型进行组合验证，为了匹配各模态的特征提取能力，将 CNN10 模型与 BERT-Tiny 搭配使用，比较其对于模型性能的影响如下表所示（汇报了模型在验证集及测试集的各项指标，指标计算在十次试验下取平均值），

表 4 多模态特征融合策略对于模型性能的影响

A-Model	T-Model	Fusion	Val UA	Val WA	Test UA	Test WA
CNN10	-	-	0.654	0.651	0.631	0.659
-	BERT-Tiny	-	0.671	0.688	0.676	0.696
CNN10	BERT-Tiny	Early	0.707	0.722	0.714	0.730
CNN10	BERT-Tiny	Attention	0.709	0.726	0.723	0.741

从上表可以看出，相较于单模态，多模态模型由于可以同时融合多种来源的信息，在性能上普遍胜过各个单模态。此外，基于注意力的多模态特征融合优于早期融合，原因可能是其更多地考虑了模态之间的交互而不是简单地使用模态特征的拼接。

## 5 多模态情感识别系统

在提升算法性能的同时，为了方便查看端到端的情感识别效果，我们实现了一个简单的情感识别系统，支持用户在本机自行输入文本与音频，然后在网页上实时查看仅使用音频模态进行情感识别的效果、仅使用文本模态进行情感识别的效果，和多模态情感识别的效果。系统的后端使用 Python 的 Flask 框架，前端使用简单的 HTML+CSS+Javascript 进行搭建。

在后端调用我们训练好的模型进行推理的时候，其中的文本模型我们使用 Mindspore 框架进行重新部署。Mindspore 框架相对于 Pytorch 框架来说，对 CPU 推理的支持更好，部署简单且性能优异。部署流程使用 Docker 进行自动化控制，部署的服务器为华为云 2 核 4G 的云耀云服务器，硬盘大小为 40G。

系统的效果展示如下：

# Multimodal Sentiment Analysis


Please Input Text:

Oh See, I don't care what you do, see. You can paint yourself green and run naked through the Place Vendome and run off with all of the men of the world. I shan't say a word, just as long as you love me best.

UPLOAD TEXT

SELECT AUDIO

UPLOAD AUDIO



infer.wav

Classification Result

1

开心

Text: 开心

Audio: 开心

SUBMIT

图 11 多模态情感识别系统界面示意图

左上角为文本的输入框，左下角为音频的上传位置，图片展示的是音频的波形图。上传好文字和音频后，点击下方按钮即可以在右侧展示情感分类的结果。其中的粉色圆形内部是多模态推理的数值形式的标签，粉色文字是多模态推理的实际标签。下方的两行分别是通过文字和音频进行单模态推理的结果。可以看出，对于这个文本与音频来说，三种模型的推理效果均正确。

在实际测试的过程中，我们找到了一些单模态推理错误但是多模态推理正确的例子，从而更加直观展示了多模态情感识别系统相较于单模态情感识别系统的优势。

# Multimodal Sentiment Analysis


Please Input Text:

Well, sometimes you have to do things that you don't necessarily like to do to have a job. But the job affords you money, you won't have to be taking the bus right now.

UPLOAD TEXT

SELECT AUDIO

UPLOAD AUDIO



infer.wav

SUBMIT

Classification Result

0

中性

Text: 生气

Audio: 中性

图 12 文本模态识别错误，语音模态与多模态识别正确

# Multimodal Sentiment Analysis

Please Input Text:

But I mean she could, but it was- so it was kind of sitting at her feet holding her up and she had these poems she had memorized. And I remember looking up, she had curly red hair and there was this palm tree right behind her and it was like a crown growing right out of her head. It was cool.

UPLOAD TEXT

SELECT AUDIO

UPLOAD AUDIO



infer.wav

SUBMIT

Classification Result

3

悲伤

Text: 悲伤

Audio: 中性

图 13 语音模态识别错误，文本模态与多模态识别正确

## 6 参考文献

- [1] El Ayadi M, Kamel M S, Karray F. Survey on speech emotion recognition: Features, classification schemes, and databases[J]. Pattern recognition, 2011, 44(3): 572-587.
- [2] Busso C, Bulut M, Lee C C, et al. IEMOCAP: Interactive emotional dyadic motion capture database[J]. Language resources and evaluation, 2008, 42: 335-359.
- [3] Gemmeke J F, Ellis D P W, Freedman D, et al. Audio set: An ontology and human-labeled dataset for audio events[C]//2017 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, 2017: 776-780.
- [4] Kong Q, Cao Y, Iqbal T, et al. Panns: Large-scale pretrained audio neural networks for audio pattern recognition[J]. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2020, 28: 2880-2894.
- [5] Chen K, Du X, Zhu B, et al. HTS-AT: A hierarchical token-semantic audio transformer for sound classification and detection[C]//ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2022: 646-650.
- [6] Park D S, Chan W, Zhang Y, et al. Specaugment: A simple data augmentation method for automatic speech recognition[J]. arXiv preprint arXiv:1904.08779, 2019.
- [7] Nam H, Kim S H, Park Y H. Filteraugument: An acoustic environmental data augmentation method[C]//ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2022: 4308-4312.
- [8] Zhang H, Cisse M, Dauphin Y N, et al. mixup: Beyond empirical risk minimization[J]. arXiv preprint arXiv:1710.09412, 2017.
- [9] Chen Y. Convolutional neural network for sentence classification[D]. University of Waterloo, 2015.
- [10] Huang Z, Xu W, Yu K. Bidirectional LSTM-CRF models for sequence tagging[J].



arXiv preprint arXiv:1508.01991, 2015.

[11] Devlin J, Chang M W, Lee K, et al. Bert: Pre-training of deep bidirectional transformers for language understanding[J]. arXiv preprint arXiv:1810.04805, 2018.

[12] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[J]. Advances in neural information processing systems, 2017, 30.

[13] Liu P, Yuan W, Fu J, et al. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing[J]. ACM Computing Surveys, 2023, 55(9): 1-35.

[14] Lester B, Al-Rfou R, Constant N. The power of scale for parameter-efficient prompt tuning[J]. arXiv preprint arXiv:2104.08691, 2021.

[15] Liu X, Ji K, Fu Y, et al. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks[J]. arXiv preprint arXiv:2110.07602, 2021.