基于局部 Attention 和 CTC 融合的语音情感识别方法研究

孟令源[1] 孙 哲[1] 刘 扬[1*] 赵 振[1] 李永伟[2]

[1](青岛科技大学信息科学技术学院 山东 青岛 266100)

[2](中国科学院自动化研究所模式识别国家重点实验室 北京 100089)

摘 要 针对基于时间序列的语音情感识别方法难以计算情感帧携带的情感信息量的问题,提出一种局部注意力机制(LAM)和结合连接主义时间分类(CTC)融合的语音情感识别模型(LAM-CTC)。提取 VGFCC 情感特征作为共享编码器的输入;CTC 层最小化代价损失并预测情感类别,LAM 层使用局部注意力机制计算上下文向量;通过解码器对上下文向量进行解码;通过平均值法将解码结果融合得到情感预测结果。实验结果表明,提出的模型在 IEMOCAP 数据集上的 UAR 和 WAR 分别达到了 68.1% 和 68.3%。

关键词 语音情感识别 注意力机制 CTC VGFCC IEMOCAP

中图分类号 TP3 **文献标志码** A DOI:10.3969/j.issn.1000-386x.2024.10.030

SPEECH EMOTION RECOGNITION USING LOCAL ATTENTION MECHANISM AND CTC

Meng Lingyuan[1] Sun Zhe[1] Liu Yang[1*] Zhao Zhen[1] Li Yongwei[2]

[1](*School of Information Science and Technology, Qingdao University of Science and Technology, Qingdao 266100, Shandong, China*)

[2](*National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100089, China*)

Abstract Aimed at the problem that time series-based speech emotion recognition methods are difficult to calculate the amount of emotion information carried by emotion frames, a speech emotion recognition model (LAM-CTC) combined with local attention mechanism (LAM) and connectionist temporal classification (CTC) is proposed. The VGFCC emotional features were extracted as the input of the shared encoder. The CTC layer minimized the cost loss and predicted the emotional category, and the LAM layer used the local attention mechanism to calculate the context vector. The decoder decoded the context vector. The average method was used to fuse the decoding results to obtain the emotion prediction results. Experimental results show that the UAR and WAR of the proposed model on the IEMOCAP dataset reach 68.1% and 68.3%, respectively.

Keywords Speech emotion recognition Attention mechanism CTC VGFCC IEMOCAP

0 引 言

语音情感识别是人工智能领域的一个新兴分支,在远程教育、医学辅助和汽车驾驶等人机交互领域有着广泛的应用前景。传统的语音情感识别方法通过人工提取与情感相关的声学特征,利用支持向量机(SVM)[1]、隐马尔可夫模型(HMM)[2]等机器学习模型进行情感分类。但这些传统的方法面对大规模训练样本时难以实施,且由于语音中包含多种情感状态,从而导致模型训练计算量大,情感状态分类困难,最终导致整体识别率较低。

随着深度学习技术的发展,深度神经网络能够轻松地应对大规模训练样本,出色地学习到语音情感的特征,且具有优秀的泛化能力。George 等使用原始的音频样本训练卷积-循环神经网络(CRNN)进行连续

收稿日期:2021-05-20。山东省自然科学基金项目(ZR2020QF007)。**孟令源**,本科生,主研领域:语音信号处理。**孙哲**,本科生。**刘扬**,讲师。**赵振**,副教授。**李永伟**,助理研究员。

性语音情感预测[3]。Zhang 等[4]使用语谱图训练全卷积神经网络,把序列变换问题转化成图像识别问题。文献[5]使用深度残差网络(ResNet)在消除了语音样本环境噪声的同时保留了情感特征,提高了语音情感识别的效果。尽管深度神经网络弥补了传统方法识别语音情感的缺陷,但语音情感的强烈程度随时间连续变化,而绝大多数深度神经网络将所有帧的特征一并提取,无法区分语音中的情感帧与非情感帧,因此在模型训练时会受到大量的非情感帧特征的干扰而降低模型表现。

Graves 等[6]提出连接主义时间分类(Connectionist Temporal Classfication,CTC)算法,通过将语音转化为序列,能够学习整个样本的误差从而使网络自动收敛到情感特征最明显的时间点[7-8]。但 CTC 算法仅考虑当前帧是否属于情感帧,并未考虑每个语音帧包含的情感信息量的差异。注意力机制(Attention Mechanism,AM)[9]通过计算和比较语音信号中情感特征与各个时域的相关权重,并选取权重较大的时域信号进行识别,使神经网络能够更专注于捕获语音的情感,从而确保关键信息不会丢失[10]。陈晓敏[11]提出了一种 AM 与 CTC 的融合模型,AM 和 CTC 共享同一个编码器,其中 CTC 模块利用编码器输出的情感语义编码序列计算损失,AM 模块对编码器输出的情感语义编码序列进行解码,CTC 模块和 AM 模块的融合使神经网络能够同时在情感关键帧和辅助帧上进行学习。但是此方法中 AM 是全局注意力方法,并未考虑样本序列在时间轴上局部注意力的变化结果,从而限制了语音情感识别精度的提高。

因此本文提出了一种基于局部注意力机制(Local Attention Mechanism,LAM)和 CTC 融合的语音情感识别方法。首先,使用 SVM 提取输入语音的 VGFCC 特征;然后,将提取的特征输入共享编码器,编码器隐藏层输出作为 LAM 层以及 CTC 层的输入;其中,CTC 层通过 LSTM 网络 Softmax 层的激活值与标签向量计算交叉熵误差输出并进行反向传播,匹配语音中的情感关键帧并输出情感预测值概率;LAM 层根据注意力机制算法将编码输出匹配度以及编码输出加入上下文相关度计算,并通过 Softmax 函数将其归一化得到输出,在不同情感帧中抽取不同程度的信息进行学习;然后解码器对 LAM 层输出的上下文向量进行解码,输出情感预测值概率。最后,对 CTC 层与解码器输出的概率值取平均值,通过 Sigmoid 函数锐化预测概率值得到最终的情感预测类别。

本文通过局部注意力机制减少无关语音帧对当前语音帧权重计算的干扰,从而提高模型的性能,同时也能够更好地适应不同大小的数据集,减少网络模型参数,显著缩短模型训练时间。通过对解码器网络结构进行了优化并将 CTC 和 LAM 的结果进行融合,使模型更加充分地对情感语音中的情感特征进行学习。

1 基于 LAM-CTC 的语音情感识别模型

本文提出的基于 LAM-CTC 的语音情感识别模型包括 VGFCC 特征提取模块、编码模块、解码模块和融合输出模块,如图 1 所示。

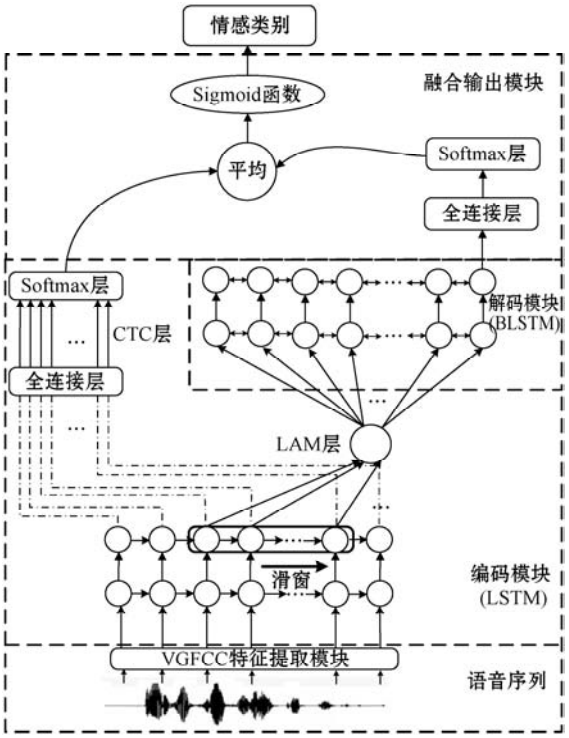

图 1 LAM-CTC 模型结构图

1.1 VGFCC 特征提取

VGFCC 考虑了语音信号的非线性和非平稳特性,能够准确反映信号的局部特征,同时对噪声具有鲁棒性[12]。VGFCC 特征的提取包含五个步骤:

(1) 通过裁减或末端补零的方式将输入语音信号 $x(n)$ 的时间长度统一设置为 7.5 秒。

(2) 将 $x(n)$ 的采样率设置为 16 000 Hz,并进行预加重、分帧、加窗处理得到 $x'(n)$。

(3) 将 $x'(n)$ 分解成 K 个固有模式函数(Intrinsic Modal Function,IMF)分量,并进行快速傅里叶变换得到频谱幅度 $X_k(r)$。其中 r 是每帧信号点数,N 为傅里叶变换点数,$k=1,2,\cdots,K$。

(4) 通过对 IMF 分量的频谱幅度取模平方并求和获得信号的能量谱:

$$E_k(r)=|X_k(r)|^2,k=1,2,\cdots,K \tag{1}$$

(5) 通过 Gammatone 滤波器对能谱进行滤波;对滤波后的结果进行离散余弦变换得到 VGFCC 特征。

1.2 编码模块

1.2.1 共享编码器

CTC 层和 LAM 层共享一个具有 256 个神经元节点的双隐层 LSTM 编码器。将提取的 VGFCC 特征输入到共享编码器中,LSTM 通过门结构让信息有选择性地影响神经网络每个时刻的状态,每个 LSTM 单元的输出都作为 CTC 层和 LAM 层的输入。对于一个输入编码器的 VGFCC 特征 $\boldsymbol{X}=(x_1,x_2,\cdots,x_t)$,$x_i\in\mathbf{R}^m$,$i=1,2,\cdots,t$,编码器输出层隐藏状态 $\boldsymbol{H}=(H_1,H_2,\cdots,H_t)$即为 X 编码后得到的情感语义编码序列。

共享编码器 t 时刻的输出 H_t 的计算过程如下:

(1) 通过遗忘门对编码器先前时刻的信息进行选择性过滤:

$$\boldsymbol{f}_t=\sigma(\boldsymbol{W}_{\mathrm{f}}\cdot[\boldsymbol{H}_{t-1},\boldsymbol{x}_t]+\boldsymbol{b}_{\mathrm{f}}) \tag{2}$$

式中:$\sigma(\cdot)$为 sigmoid 激活函数,$\boldsymbol{W}_{\mathrm{f}}$和 $\boldsymbol{b}_{\mathrm{f}}$为编码器 LSTM 网络遗忘门的连接权值与偏置项。

(2) 通过输入门将当前输入信息选择性地记录到神经元状态中:

$$\boldsymbol{i}_t=\sigma(\boldsymbol{W}_{\mathrm{i}}\cdot[\boldsymbol{H}_{t-1},\boldsymbol{x}_t]+\boldsymbol{b}_{\mathrm{i}}) \tag{3}$$

$$\tilde{\boldsymbol{C}}_t=\tanh(\boldsymbol{W}_{\mathrm{C}}\cdot[\boldsymbol{H}_{t-1},\boldsymbol{x}_t]+\boldsymbol{b}_{\mathrm{C}}) \tag{4}$$

式中:$\boldsymbol{W}_{\mathrm{i}}$和 $\boldsymbol{W}_{\mathrm{c}}$为 LSTM 输入门的权值,$\boldsymbol{b}_{\mathrm{i}}$和 $\boldsymbol{b}_{\mathrm{c}}$为偏置项。

(3) 通过状态更新门确定神经元的最终状态:

$$\boldsymbol{C}_t=\boldsymbol{f}_t\cdot\boldsymbol{C}_{t-1}+\boldsymbol{i}_t\cdot\tilde{\boldsymbol{C}}_t \tag{5}$$

(4) 通过输出门计算出当前神经元隐层状态:

$$\boldsymbol{o}_t=\sigma(W_{\mathrm{o}}\cdot[\boldsymbol{H}_{t-1},x_t]+b_{\mathrm{o}}) \tag{6}$$

$$\boldsymbol{H}_t=\boldsymbol{o}_t\cdot\tanh(\boldsymbol{C}_t) \tag{7}$$

式中:W_{o}和 b_{o}为 LSTM 输入门连接权值和偏置项。

1.2.2 LAM 层

结构化的 LAM 层会聚合来自共享编码器输出层隐藏状态 $\boldsymbol{H}$ 的信息,并生成固定长度的向量作为情感特征的编码。

将隐藏层状态 $\boldsymbol{H}$ 作为输入,首先确定语音帧的数量 M,然后确定可选语音帧数量 G 的大小。

当 M 小于 G 时,模型将使用全局注意力机制。注意力权重向量 $\boldsymbol{c}$ 和隐藏层状态加权和 h 的计算式为:

$$\boldsymbol{c}=\mathrm{softmax}(\boldsymbol{w}_{\mathrm{s}}\times\tanh(\boldsymbol{W}_{\mathrm{s}}\times\boldsymbol{H}^t)) \tag{8}$$

$$\boldsymbol{h}=\sum_{i=1}^{t}\boldsymbol{c}\times\boldsymbol{H} \tag{9}$$

式中:$\boldsymbol{W}_{\mathrm{s}}$和 $\boldsymbol{w}_{\mathrm{s}}$为训练时网络的超参数。

当 M 大于 G 时,模型通过使用滑动窗口来限制注意力权重向量的计算范围,并根据当前计算的位置滑动窗口。如图 2 所示,x_n为编码器神经元的隐藏层输出当前计算位置 u 在滑动窗口的正中间。如果滑动窗口的下限小于 x_1的位置 1,则从位置 1 开始。如果滑动窗口的上限大于序列的最后位置,则以最后位置结束。滑动窗口 D 的长度设置为 $G/2$,注意力权重向量 $\boldsymbol{c}_{\mathrm{s}}$ 和隐藏层状态加权和 $\boldsymbol{h}_{\mathrm{s}}$ 的计算式为:

$$\boldsymbol{c}_{\mathrm{s}}=\mathrm{softmax}(\boldsymbol{w}_{\mathrm{s}}\times\tanh(\boldsymbol{W}_{\mathrm{s}}\times\boldsymbol{H}^G)) \tag{10}$$

$$\boldsymbol{h}_{\mathrm{s}}=\sum_{i=1}^{t}\boldsymbol{c}_{\mathrm{s}}\times\boldsymbol{H} \tag{11}$$

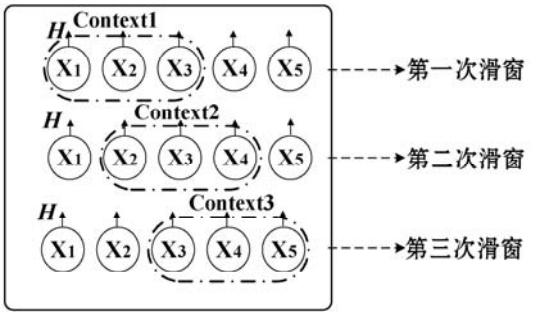

图 2　LAM 滑窗示意图

1.2.3 CTC 层

CTC 层利用编码器输出的情感语义编码序列,通过交叉熵损失函数计算 CTC 代价损失并进行反向传播,匹配语音中的情感关键帧。

首先,定义中间标签序列 $\boldsymbol{\pi}=(\pi_1,\pi_2,\cdots,\pi_{\mathrm{T}})$。假设 y'是添加了分隔符后的扩展,定义一个多对一的映射:

$$B:L'->L^{\leqslant T} \tag{12}$$

式中:$L^{\leqslant T}$是输出的中间标签序列 $\boldsymbol{\pi}$ 的集合,得到输出标签 $P(\boldsymbol{\pi}\mid x)$的概率的计算式为:

$$P(\boldsymbol{\pi}\mid x)=\sum_{\boldsymbol{\pi}\in B^{-1}(y')}p(\boldsymbol{\pi}\mid x) \tag{13}$$

其次,在输入序列的每个时间步 t 计算相应输出 π_t,假设每个时间步之间的输出顺序是有条件且独立的,中间标签序列中单个标签的概率 $P(\boldsymbol{\pi}\mid x)$的计算式为:

$$P(\boldsymbol{\pi}\mid x)\approx\prod_{t=1}^{T}P(\pi_1\mid\pi_1,\pi_2,\cdots,\pi_{t-1},x),\forall\boldsymbol{\pi}\in L' \tag{14}$$

CTC 损失函数的负对数概率的值的计算式为:

$$L_{\mathrm{CTC}}(S)=-\sum_{(x,y')\in S}\ln\sum_{\boldsymbol{\pi}\in B^{-1}(y')}\prod_{t=1}^{T}q_t(\pi_t),\forall\boldsymbol{\pi}\in L' \tag{15}$$

CTC 算法使用向前传播和向后传播算法提高计算速度:

$$p_{\mathrm{CTC}}(y\mid x)=\sum_{t=1}^{T}\sum_{u=1}^{|y'|}\frac{\alpha_t(u)\beta_t(u)}{q_t(\pi_t)} \tag{16}$$

式中:p_{CTC}为当前输入下标签 y 的概率,$\alpha_t(u)$为第 u 个

标签在时间步 t 的前向的概率，$\beta_t(u)$ 为第 u 个标签在时间步 t 的后向的概率。

1.3 解码模块

LSTM 只能按顺序处理语音序列，而 BLSTM 能够处理全局语音序列。因此，本文选择双隐层的 BLSTM 作为解码器单元。解码器接收两个输入，包括根据上一时刻注意力计算得到的注意力权重向量和上一时刻 BLSTM 单元的隐层输出 y_j。在时间步 j 时，解码器的隐层输出 k_j 计算过程如下：

$$\boldsymbol{k}_j = \boldsymbol{o}_j \cdot \tanh(\boldsymbol{C}_J) \tag{17}$$

$$\boldsymbol{o}_j = \sigma(\boldsymbol{W}_o \cdot [\boldsymbol{c}_{sj}, \boldsymbol{k}_{j-1}] + \boldsymbol{b}_o) \tag{18}$$

$$\boldsymbol{C}_J = \boldsymbol{f}_j \cdot \boldsymbol{C}_{j-1} + \boldsymbol{i}_j \cdot \boldsymbol{C}_j \tag{19}$$

$$\boldsymbol{f}_j = \sigma(\boldsymbol{W}_F \cdot [\boldsymbol{c}_{sj}, \boldsymbol{k}_{j-1}] + \boldsymbol{b}_F) \tag{20}$$

$$\boldsymbol{i}_j = \sigma(\boldsymbol{W}_I \cdot [\boldsymbol{c}_{sj}, \boldsymbol{k}_{j-1}] + \boldsymbol{b}_I) \tag{21}$$

式中：$\boldsymbol{W}^*$ 和 $\boldsymbol{b}^*$ 为注意力模型参数，$\boldsymbol{c}_{sj}$ 为时间步 j 时的注意权重向量。

解码器中 BLSTM 单元的隐藏层状态和细胞状态初始化为编码器最后一次输入时相应的 LSTM 单元的隐藏层状态和细胞状态。

1.4 融合输出模块

将解码模块输出的序列输入到带有 8 个神经元的全连接层神经网络当中，然后通过 Softmax 层输出 LAM 模型的情感类别概率值。采用梯度下降法进行学习，直到交叉熵损失下降到最小。

CTC 层和 LAM 的情感预测概率值通过平均值法进行融合。将融合的结果输入 Sigmoid 函数进行数值锐化，概率最大的情感类别作为最终的情感识别结果。

2 实验与分析

2.1 数据集介绍

实验的数据集采用长达 12 小时的来自于美国南加州大学工程学院发布的公开数据库 Interactive Emotional Dyadic Motion Capture (IEMOCAP)[12]。数据集中的音频被切分成许多短句，并由三个专家进行离散情感类别标注，其中标注的类别有“angry”“excited”“happy”“neutral”和“sad”。由于“excited”和“happy”这两种情感非常类似，故将“excited”和“happy”数据合并为“happy”类，然后将这些语音数据作为本文实验的情感数据集，最终得到 5 531 个情感语音样本(1 636 个“happy”、1 103 个“angry”、1 084 个“sad”和 1 708 个“neutral”)。

2.2 实验参数设置

实验以 Keras 为开发框架，在 GeForce1080-Ti 显卡和 Ubuntu 18.04 LST 系统上完成。实验统一将语音信号转换为 16 kHz，并使用 16 bit 量化语音信号。编码模块和解码模块选择交叉熵作为损失函数，选择 Adam 作为优化器。编码器和解码器选择的神经元个数设置为 256，学习率设置为 0.01，epoch 设置为 1 024，batch_size 设置为 32。将 5 531 条语音情感数据随机分为 80% 训练设置和 20% 的测试设置。数据集共有 4 种情感类别，且在整个训练/测试集中的比例仍与整个语料库相同。

2.3 对比实验

本文在 IEMOCAP 数据集上进行十折交叉验证实验，并计算模型训练五次的平均训练时间、加权平均召回率(Weight Average Recall，WAR)和未加权平均召回率(Unweighted Average Recall，UAR)作为性能衡量指标，实验结果如表 1 所示。为了减少数据集各类情感样本数目不均衡的影响，本文在训练过程中以最大化 UAR 为目标。

表 1 本文模型与 4 个经典模型对比实验结果

情感识别方法	UAR/%	WAR/% s	平均训练时间/min
LSTM	58.2	57.1	76
LSTM-CTC	63.7	62.9	109
LSTM-SelfAtt	66.1	65.8	114
AttRNN-RNN	67.6	67.5	105
LAM-CTC	68.1	68.3	89

本文选择 4 个经典模型进行比较，包括基于 LSTM 的语音情感识别模型、基于 LSTM-SelfAtt 的语音情感识别模型[10]、基于 LSTM-CTC 的语音情感识别模型和基于 AttRNN-RNN 的语音情感识别模型[12]。

由表 1 可知，本文提出的 LAM-CTC 的模型在 IEMOCAP 数据集上的 UAR 和 WAR 均优于其他 4 种经典模型。与 4 种经典模型中性能表现最好的 AttRNN-RNN 模型相比，本文提出的模型的 UAR 和 WAR 分别提升了 0.5 百分点和 0.8 百分点。由于 LAM 有效地提高了长句语音情感的上下文的紧密度，改善了全局注意力无法针对局部语音进行情感权重计算，从而忽略了局部情感随时间变化的问题。另外，当使用 BLSTM 作为解码器的单元时，与仅使用 RNN 相比，当前计算向量包含了更多的后导信息。

在训练时间方面，本文模型的平均训练时间为 89 分钟，与 LSTM-selfAtt 模型的平均训练时间 114 分钟相比，本文模型仅花费 LSTM + selfAtt 模型平均训练时间的 78%；与 AttRNN-RNN 模型的 105 分钟相比，本文

提出的模型仅使用了 84.7% 的平均训练时间。由于全局注意力机制在进行样本权重计算时一次性计算每一帧的权重,而 LAM 仅需计算当前滑窗内 LSTM 隐藏层输出的权重,且窗口随着时间的推移而向后滑动。在同样的时间内,LAM 计算的参数量更少,因此使用 LAM 能够减少训练时间。

2.4　消融实验

通过消融实验对 LAM 和 CTC 模块的有效性进行验证。消融实验的设置如下:(1) AM-CTC:将 LAM 替换为全局注意力,以验证 LAM 对模型性能的贡献。(2) LAM:将 CTC 模块剔除掉,即原始的 LAM-CTC 融合模型被单独的 LAM 模型取代,以验证 CTC 对模型性能的贡献。消融实验结果如表 2 所示。

表 2　消融实验结果

模型	UAR/%	WAR/%	平均训练时间/min
AM-CTC	67.7	67.4	121
LAM	66.5	66.3	**70**
LAM-CTC	**68.1**	**68.3**	89

可以看出,与 AM-CTC 相比,本文提出模型的 UAR 提高了 0.4 百分点,WAR 提高了 0.9 百分点,因为 LAM 能够提高语音情感向量的紧密度。与 LAM 相比,本文提出模型的 UAR 和 WAR 分别提高了 1.6 百分点和 2 百分点。通过融合 CTC 模块,本文提出的模型能够学习语音关键情感帧的信息,从而提高了模型的表现能力。

LAM-CTC 的训练时间仅消耗 AM-CTC 模型的训练时间的 73.5%,而 LAM 模型训练仅消耗 AM-CTC 模型的训练时间的 57.9%。因为 LAM 在计算上下文向量权重时仅需要计算当前滑窗中 LSTM 隐藏层输出的权重,且注意力窗口会向后滑动,因此 LAM 计算时的参数更少,使用 LAM 可以减少训练时间。消融实验结果表明,LAM 和 CTC 均对语音情感识别的性能起到重要作用。

3　结　语

本文提出了一种融合 LAM 和 CTC 的语音情感识别深度学习模型。本文通过 LAM 根据语音序列的长度来匹配注意力模型,并更改长句子上语音情感向量的计算范围;通过 CTC 使模型能够自动收敛到情感特征最明显的时间点上。在 IEMOCP 数据集上实验结果表明,相比于目前最先进的语音情感识别模型,本文模型的 UAR 和 WAR 分别提高了 0.5 百分点 ~0.8 百分点。

参　考　文　献

[1] 朱菊霞,吴小培,吕钊. 基于 SVM 的语音情感识别算法[J]. 计算机系统应用,2011,20(5):87 - 91.

[2] Foo S W, Nwe T L, De Silva L C. Speech emotion recognition using hidden Markov models[J]. Speech communication, 2003, 41(4): 603 - 623.

[3] Ringeval F, Trigeorgis G, Brueckner R. Adieu features end-to-end speech emotion recognition using a deep convolutional recurrent network [C]//2016 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE,2016: 5200 - 5204.

[4] Zhang Y, Du J, Wang Z, et al. Attention based fully convolutional network for speech emotion recognition[C]//2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC). IEEE, 2018: 1771 - 1775.

[5] Triantafyllopoulos A, Keren G, Wagner J, et al. Towards robust speech emotion recognition using deep residual networks for speech enhancement [C]//INTERSPEECH, 2019: 1691 - 1695.

[6] Graves A, Fernández S, Gomez F,et al. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks[C]//Proceedings of International Conference on Machine Learning,2006:25 - 26.

[7] Miao Y, Gowayyed M, Metze F. EESEN: End-to-end speech recognition using deep RNN models and WFST-based decoding[C]//2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU). IEEE, 2015: 167 - 174.

[8] Amodei D, Ananthanarayanan S, Anubhai R, et al. Deep speech 2: End-to-end speech recognition in english and mandarin[C]//ICML'16: Proceedings of the 33rd International Conference on International Conference on Machine Learning. ACM,2015.

[9] Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate[EB]. arXiv:1409.0473,2014.

[10] 陈巧红,于泽源,孙麒,等. 基于注意力机制与 LSTM 的语音情绪识别[J]. 浙江理工大学学报(自然科学版),2020,43(6):815 - 822.

[11] 陈晓敏. 基于时序深度学习模型的语音情感识别方法研究[D]. 哈尔滨:哈尔滨工业大学,2018.

[12] 刘雨柔,张雪英,陈桂军,等. VMD 改进 GFCC 的情感语音特征提取[J]. 计算机工程与设计,2020,41(8):2265 - 2270.

[13] Busso C, Bulut M, Lee C C, et al. IEMOCAP: Interactive emotional dyadic motion capture database[J]. Language Resources and Evaluation, 2008, 42(4): 335 - 359.