

端到端单通道睡眠 EEG 自动分期模型

金欢欢¹ 尹海波² 何玲娜¹

(浙江工业大学计算机科学与技术学院 杭州 310023)¹ (哈尔滨工业大学航天学院 哈尔滨 150001)²

摘要 针对现阶段数据和特征决定自动睡眠分期模型分类精度上限的问题,提出一种基于深度混合神经网络的自动睡眠分期模型。在模型主体构建方面,使用多尺度卷积神经网络自动学习高级时不变特征,使用双向门限循环单元构建的循环神经网络对时不变特征中的时间信息进行解码,并用残差连接实现时不变特征与时间信息特征的融合。在模型优化方面,将 MSMOTE(Modified Synthetic Minority Oversampling Technique)重构后的数据集用于预训练,以减少类不平衡对少数类的分类效果的影响,应用 Swish 激活函数加速模型收敛。使用 Sleep-EDF 数据集中 Fpz-Cz 通道的原始 EEG 数据对模型进行 15 折交叉验证,得出 OA(Overall Accuracy)和 MF1(Macro-averaged F1-score)分别为 86.85%和 81.63%。提出的模型可避免特征选取的主观性以及类不平衡小数据集在深度学习中的局限性。

关键词 端到端,单通道,睡眠分期,门限循环单元,Swish,深度学习

中图分类号 TP391 文献标识码 A DOI 10.11896/j.issn.1002-137X.2019.03.036

End-to-End Single-channel Automatic Staging Model for Sleep EEG Signal

JIN Huan-huan¹ YIN Hai-bo² HE Ling-na¹

(College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310023, China)¹

(School of Astronautics, Harbin Institute of Technology, Harbin 150001, China)²

Abstract The classification accuracy of current automatic sleep staging is determined by the small data set of imbalanced classes and hand-engineered features. Aiming at this problem, this paper proposed an automatic sleep staging model based on deep hybrid neural network. For the construction of model's main structure, the multi-scale Convolutional Neural Networks are used to automatically learn the high-level time-invariant features, the Recurrent Neural Networks constructed by bidirectional Gated Recurrent Unit are used to decode the temporal information from the time invariant features, and the residual connection is used to fully combine the time invariant features with the time information features. For model optimization, in order to reduce the impact of the dataset of imbalanced class on the classification effect of minority class, the experimental data set reconstructed by MSMOTE (Modified Synthetic Minority Oversampling Technique) is used for pre-training. The Swish activation function is used to accelerate the training convergence rate. The experiment was set up on the initial single-channel EEG signal of Fpz-Cz in Sleep-EDF Database. The 15-fold cross-validation experiments show that the overall classification accuracy is 86.85% and the Macro-averaged F1-score is 81.63%. This model can effectively avoid the subjectivity of feature selection and the limitation of class imbalanced small dataset of imbalanced class in deep learning.

Keywords End to end, Single-channel, Sleep staging, Gated recurrent unit, Swish, Deep learning

1 引言

长期以来,自动睡眠分期模型多基于特征工程^[1-3],这类模型针对特定数据集的效果较好,但是其由于需要针对数据集特点进行人工特征设计,因此无法实现端到端学习,实际可应用性并不强。

近年来,机器学习在图像识别、自然语言处理以及语音识别领域已取得大量突破性的成果^[4-7]。尤其在医学成像识别中,得益于可直接应用 ImageNet 对模型进行预训练,基于深度神经网络的深度学习模型能够接近并超过人类专家水

平^[8]。当前,已有少量研究将深度学习模型应用于自动睡眠分期^[9-10],实现了特征的自动提取。深度学习模型的引入虽然有效避免了特征选取的主观性,但由于其处理的对象为小型类不平衡睡眠脑电数据集,因此模型的迁移应用效果并不理想。

本文在上述研究的基础上,结合残差连接(Residual Connection)^[11]、Swish 激活函数^[12]、MSMOTE^[13]、两步训练法(Two-Step Training Algorithm)^[10]等深度学习优化技术,以卷积神经网络(Convolutional Neural Networks, CNNs)和双向门限循环单元(Bidirectional Gated Recurrent Unit, BiGRU)

收稿日期:2018-01-18 返修日期:2018-04-10 本文受浙江科技计划公益技术项目(2015C31111)资助。

金欢欢(1990-),女,硕士生,主要研究方向为机器学习、脑机接口;尹海波(1990-),男,硕士生,主要研究方向为神经网络、智能诊断;何玲娜(1978-),女,副教授,主要研究方向为脑机接口, E-mail: 2099376617@qq.com(通信作者)。

构建了一个以循环神经网络(Recurrent Neural Networks, RNNs)为主体的端到端自动睡眠分期模型。该模型首先通过多尺度 CNNs 自动学习不同层次的高级时不变特征;然后使用 BiGRU 构建的 RNNs 学习时不变特征中所蕴含的各睡眠阶段间的长间期依赖关系,通过残差连接实现不同类型特征的融合;最后通过全连接的 Softmax 层对融合后的特征进行识别。除此之外,模型还采用 MSMOTE 对数据集中的少数类进行扩增,由此构建的类平衡数据集则用于预训练,从而提高少数类的识别率;应用 Swish 激活函数替换目前广泛使

用的 ReLU 激活函数,在提高分类效果的同时加快了收敛速度。该模型可在不对原始数据作明显假设,以及没有睡眠专家参与的情况下,完成原始单通道睡眠脑电数据的端到端学习,且总体分类效果目前处于较高水平。

2 模型架构及实验数据

2.1 模型结构

本文提出的端到端自动睡眠分期模型由 4 部分组成,图 1 中依次标示为 Part1,Part2,Part3,Part4。

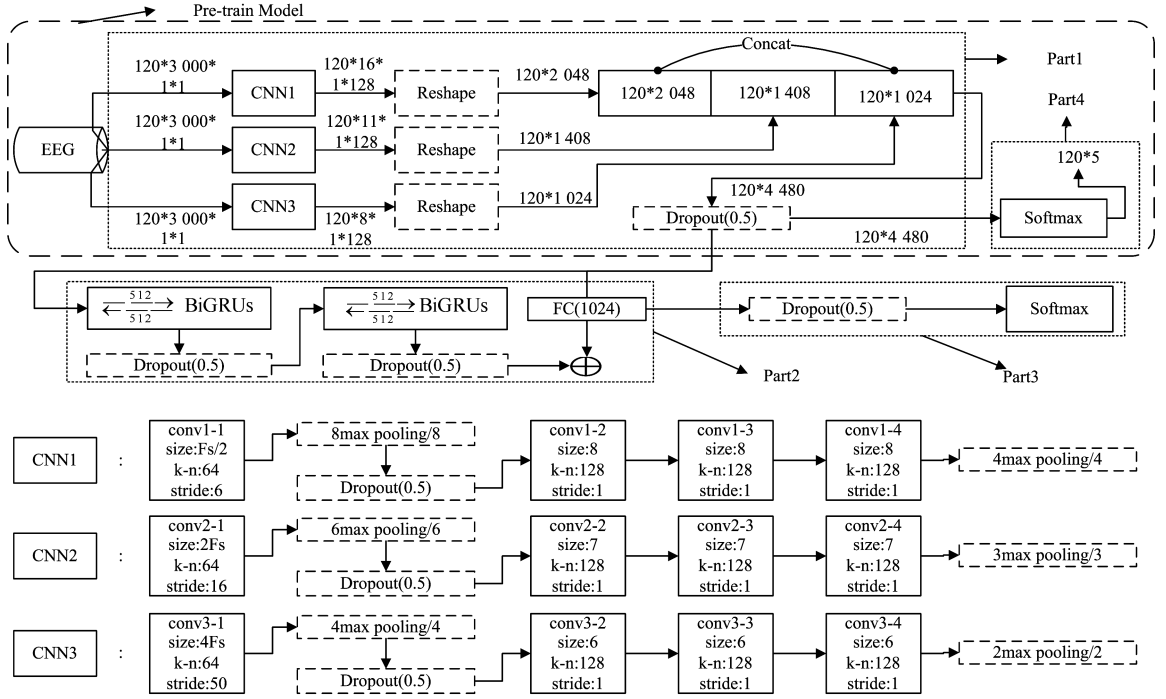


图 1 总体模型结构及其参数

Fig. 1 Overall structure and detailed parameters of proposed model

图 1 详细介绍了模型结构及其参数,以及预训练部分的数据流。其中,CNN1,CNN2,CNN3 为 3 个独立的卷积神经网络,“Reshape”代表平铺操作,“Concat”为特征融合操作,“Dropout(0.5)”代表 Dropout 层以 0.5 的概率进行 Dropout 操作,“conv1-1”表示 CNN1 中的第一个卷积层,“size”表示卷积核尺寸(Fs 表示所使用实验数据的采样率),“k-n”表示卷积核数目,“stride”表示卷积运算步长,“8max pooling/8”表示最大池化操作层,池化层的尺寸为 8,步长为 8,“ $\overleftarrow{512}$ BiGRUs”表示双向 GRU 的前向和后向单元数都为 512。

Part1 的主要功能为自动提取特征,其位于模型的最底层。信号处理技巧包括:较小尺寸(0.5Fs)的卷积核能更好地捕获时域特征,中等尺寸(2Fs)的卷积核能更好地捕获时频特征,较大尺寸(4Fs)的卷积核能更好地捕获频间特征^[14]。为全面覆盖数据特征,Part1 采用 3 个独立的 CNNs(CNN1, CNN2, CNN3)对序列进行并行特征提取;3 个 CNNs 初始卷积层(conv1-1, conv2-1, conv3-1)的卷积核大小分别为 0.5Fs, 2Fs, 4Fs。

Part1 的具体工作原理如下:假设有 n 段单通道脑电数据(每段数据时长为 30s),它们共同组成的脑电数据序列为 $\{e_1, e_2, \dots, e_n\}$,使用 3 个并列的 CNNs 对 e_i 的第 i 个特征 r_i 进行

提取,计算式如下:

$$r_i = [h_i^s; h_i^m; h_i^l] \quad (1)$$

$$h_i^s = \text{CNN}_{\theta_s}(e_i) \quad (2)$$

$$h_i^m = \text{CNN}_{\theta_m}(e_i) \quad (3)$$

$$h_i^l = \text{CNN}_{\theta_l}(e_i) \quad (4)$$

其中, $\text{CNN}(e_i)$ 表示通过 CNNs 将 e_i 变换为一个特征向量 h_i ; $\theta_s, \theta_m, \theta_l$ 分别是 3 个独立的 CNNs 的参数;“ $[\]$ ”是将 3 个 CNNs 的输出组合在一起的“Concat”操作; h_i^s, h_i^m, h_i^l 分别表示 3 个 CNNs 最终提取的关于 e_i 的第 i 个时不变特征。

每个卷积层都含有 3 个基本操作:卷积运算、批归一化^[15]以及应用 Swish 自门控单元进行激活。此外,3 个 CNNs 都含有两个最大池化(max-pooling)层和一个 Dropout 层。最大池化操作可在保持特征图数目不变的情况下,使特征图的分辨率降低,以保留高分辨率特征图所承载的特征;通过最大池化操作能够降低神经网络模型的复杂度,以达到减少过拟合的目的。Dropout 操作会按照一定概率随机“丢弃”隐层神经元,以减弱训练期间不同神经元间的共适性,从而达到增强模型泛化能力的目的。Dropout 被大量用于全连接神经网络,通常按照 0.5 或者 0.3 的概率来对神经元进行“丢弃”;在含有 ReLU 激活函数的 CNNs 隐藏层中,由于 ReLU

本身具有较强的稀疏化特性,Dropout 的应用相对较少;鉴于本文的 Swish 激活函数的强制稀疏特性较弱,因此在 3 个 CNNs 中都采用了 Dropout 操作。

CNNs 中激活函数的使用极其重要,即使所使用的激活函数只有微小的性能增强,也会由于它的大量使用而使 CNNs 获得较大的性能提升,目前被广泛使用的激活函数是 ReLU 激活函数,该激活函数是谷歌大脑团队于 2017 年提出的一种新型激活函数。Swish 通过使用相同值进行门控来完成自门控机制的构建;自门控机制使 Swish 激活函数可在 CNN 中轻易置换使用单个标量作为输入的激活函数(如 ReLU),并且不用改变参数的隐藏容量。两个激活函数(即 Swish 和 ReLU)的公式如下:

$$\text{Swish}(x) = \frac{x}{1 + \exp(-x)} \quad (5)$$

$$\text{ReLU}(x) = \max(x, 0) \quad (6)$$

图 2 为 Swish 函数和 ReLU 函数的形态对比图。

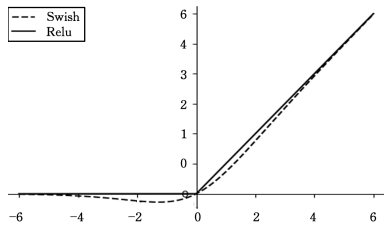


图 2 Swish 函数和 ReLU 函数的曲线对比图

Fig. 2 Curve comparison between Swish and ReLU

如图 2 所示,当 x 为负值时,ReLU 会做强制置零处理,这种强制稀疏处理在一定程度上缩小了模型的有效容量,屏蔽过多特征以致模型无法学习有效特征。Swish 和 ReLU 都有下界而无上界,此外 Swish 还具有非单调特性和平滑特性。谷歌大脑团队通过实验得出:Swish 直接替换 ReLU 激活函数可以在总体上增强模型性能,且这种优势会随着神经网络的加深更加明显^[12]。本文将尝试使用 Swish 函数作为模型中卷积层的激活函数。

Part2 是由 BiGRU 构建的 RNNs 和残差相连接而共同组成的序列残差学习模块^[11],该模块可从 Part1 输出的时不变特征中学习各睡眠阶段间的长周期依赖,并完成不同类型特征的融合。其结构及参数如图 1 中的 Part2 所示。

门限循环单元(Gated Recurrent Unit, GRU)是长短期记忆网络(Long Short Term Memory, LSTM)的一个变体。GRU 除了拥有 LSTM 的优点外,其参数比 LSTM 更少,可有效抑制过拟合,且在处理同样的问题时收敛速度更快,因此在训练大量数据时 GRU 能节约大量时间^[16]。

相对于单向 GRU, BiGRU 可以通过使用两个 GRU 独立地处理前后两个方向的输入序列;两个方向的 GRU 的输出彼此无连接,因此能够利用前后两侧的时间信息。综合考虑模型对收敛速度、减少过拟合以及充分利用时间点前后两侧时间信息的需求,模型使用 BiGRU 构建的 RNNs 对睡眠 EEG 信号中所蕴含的时间信息进行解码。

Part3 和 Part4 都是完全连接的 Softmax 层,分别用于模型的预训练和微调。其中,Part4 仅用于模型的预训练,预训练结束后其将被“丢弃”。

2.2 实验数据

在实验数据选取方面,本文使用了 PhysioBank 的公开基准睡眠 EEG 数据:Sleep-EDF 数据集。该数据集包含 25~34 岁的 10 名健康女性和 10 名健康男性的睡眠多导图记录,实验过程没有使用任何药物,实验周期为两个昼夜(约 40 小时),采样率为 100 Hz。所有的 PSG 被划分为 30 s 每段(含有 3000 个数据点),然后由训练有素的睡眠专家根据 R&K 标准进行人工分类标记。使用 Fpz-Cz 通道的原始 EEG 信号对模型的性能进行评估。由于原数据集中的文件(SC4132 E0.edf)已经被销毁,现存实验数据集共有 39 个文件。R&K 标准将整个睡眠阶段分为 6 期(即 W, REM, S1, S2, S3, S4),其中 W 期、REM 期、S1 期、S2 期依次对应 AASM 规则中的 W 期、REM 期、N1 期、N2 期,R&K 标准中的 S3 期和 S4 期对应 AASM 标准中的 N3 期。本文使用现行通用的 AASM 标准。表 1 列出了睡眠专家的分类结果。

表 1 睡眠专家对数据的分类及各期比例

Table 1 Sleeping experts' classification and proportion results

阶段	W	N1	N2	N3	REM	Total
段数	7726	2699	17258	5467	7350	40500
比例/%	19.08	6.66	42.61	13.50	18.15	100

由表 1 可直观地看出,本文所使用的实验数据集为类不平衡数据集,其中 N1 期和 N2 期的比例约为 1:6,最小类 N1 约占总数的 6.66%。原始数据中出现的类不平衡现象是分类任务中的典型问题,通常选择少数类占总比例的 10%~20%为类不平衡研究对象;实际得到的数据集中的少数类占比通常低于 10%,本文实验数据集中的最小类占比仅为 6.66%。一般情况下,若实验数据集的样本量较大,各类都会有足量的可训练样本,类不平衡问题是可以忽略的。但是本文所使用的实验数据集较小,类不平衡会使模型对于少数类的识别效果较差。通常通过简单复制来对少数类进行过采样,以减少类不平衡对少数类分类效果造成的负面影响。但采用复制过采样技术生成的类平衡数据集用于训练时极易出现过拟合现象,本文使用 MSMOTE 取代传统的复制过采样技术,来完成数据集中少数类的过采样。通过 MSMOTE 将数据集中少数类划分为噪声类、边界类、安全类,并对边界类和安全类做过采样处理,进而完成少数类的过采样,最终将实验数据集重构为类平衡数据集。

3 模型训练

针对实验数据集的特点以及 MSMOTE 重构生成的类平衡数据集所存在的缺陷,本文采用两步训练法对模型进行训练。在训练过程中使用 Adam 优化器对网络权值进行更新。图 3 给出了具体的训练流程。

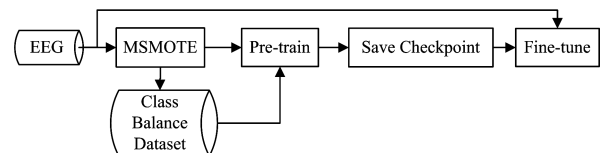


图 3 模型训练方法解析图

Fig. 3 Analysis diagram for model training method

预训练结构以及数据流已标注在图 1 中,微调模型结构

由图1中的 Part1, Part2, Part3 组成。预训练(Pre-train)可以使卷积层权值靠近较优的(保持在高梯度区间内)局部最优解,以便能够有效地对权值进行进一步的微调(Fine-tune)。

模型训练的具体步骤如下。

1)使用经 MSMOTE($K=8$)重构后生成的类平衡数据集对图1中的“Pre-train Model”进行有监督预训练,保留预训练最后一个 Epoch 后生成的神经网络权值,并将其作为微调结构中 Part1 的参数初值。预训练的批量大小为 120,学习率为 0.001。

2)考虑到经 MSMOTE 重构后生成的类平衡数据集中会出现不属于任何分类的数据,因此使用未经处理的、按照时间顺序排列的原始顺序数据集对微调模型进行有监督微调。微调的批量大小为 250(每次输入 10 个序列,每个序列的长度为 25)。鉴于 Part1 已经被预先训练,为防止其参数被过度调整,微调时使用两个具有不同学习率的 Adam 优化器:Part1 使用较低的学习率(0.00001),Part2 使用较高的学习速率(0.0001)。Adam 优化器的其他参数则使用默认值。

4 模型评估及实验分析

4.1 模型评估设计

合理地选取评价指标是客观评价一个模型的前提,本文从每类指标和总体指标这两方面分别对实验结果进行评估。每类指标包括每类精度、每类召回率、每类 F1 值,总体指标包括总体精度和宏平均 F1 值。每类评估是通过将待评估类作为正例,将其他各类作为负例进行计算的。各指标公式如下:

$$PR_i = \frac{TP_i}{N_i'} \quad (7)$$

$$RE_i = \frac{TP_i}{N_i} \quad (8)$$

$$F1_i = \frac{2PR_i \times TP_i}{PR_i + TP_i} \quad (9)$$

$$OA = \frac{\sum_{i=1}^I TP_i}{N} \quad (10)$$

$$MF1 = \frac{\sum_{i=1}^I F1_i}{I} \quad (11)$$

其中, $PR_i, RE_i, F1_i$ 分别表示第 i 类精度、召回率、F1 值; TP_i 为 i 类真阳数(实际为 i 类又被预测为 i 类), N_i' 为 i 类的预测数, N_i 为 i 类的实际数。OA 和 MF1 分别为总体精度和宏平均 F1 值, I 为睡眠阶段总数, N 为所有类的总样本数。

为最大化地利用现有小样本数据对模型的效果进行合理评估,使用 K 折交叉验证来评估本文模型。

4.2 实验与结果分析

4.2.1 实验方案设计

为确定交叉验证的 K 值以及验证本文所提各种结构的合理性,实验设计如表 2 所列。表 2 中,“S”表示未处理的原始顺序数据集,“M”表示采用 MSMOTE 重构后生成的类平衡数据集,“CY”表示采用复制过采样技术生成的类平衡数据集。实验 A1, A2, A, A3 使用的都是理论最优模型,其交叉验证 K 值分别为 5, 10, 15, 20。B, C, D, E 为模型优化验证实验。

表 2 实验设计

Table 2 Experimental design

实验编号	RNNs	激活函数	预训练		微调	
			数据集	Epoch	数据集	Epoch
A1, A2, A, A3	Bi-GRU	Swish	M	20	S	160
B	Bi-GRU	ReLU	M	20	S	260
C	GRU	Swish	M	20	S	160
D	Bi-GRU	Swish	M	20	M	160
E	Bi-GRU	Swish	CY	20	S	160

4.2.2 实验结果

图 4 给出了 8 组实验所获得的实验结果,评价指标为 OA 和 MF1。

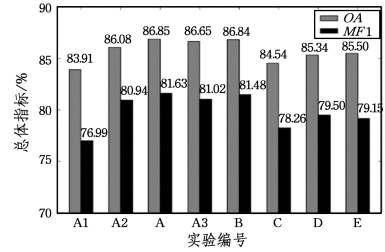


图 4 实验结果对比

Fig. 4 Comparison of experimental results

在 8 组实验中,首先进行的是 A1, A2, A, A3 4 组实验,由 4 组实验的结果可知 15 折交叉验证的效果最好。A1(5 折)和 A2(10 折)的效果比 A(15 折)差的原因是可用于训练的数据相对较少;A3(20 折)的效果较 A 略微下降的原因是实验数据相关性增大从而造成过拟合。综合上述分析,为合理评估模型的性能,本文采用 15 折交叉验证,因此 B, C, D, E 4 组对比实验都采用 15 折交叉验证。为控制对比实验中预训练结果的随机性,部分与预训练无关的对比实验(C, D)直接使用实验 A 中的预训练结果。

4.2.3 激活函数性能的对比分析

实验 A 和实验 B 分别使用 Swish 和 ReLU 作为卷积层的激活函数。由图 4 中实验所获得的总体指标可知,卷积层采用 Swish 激活函数获得的分类性能略优于 ReLU,但优势并不明显。

为了进一步对比应用两种激活函数所获得的收敛速度,分别用每个实验所有折微调的训练集损失函数的平均值和训练集总体精度平均值作为两个实验微调收敛效果的指标,评价范围为前 160 个 Epoch。图 5 为两个实验微调收敛效果的对比图。通过图 5 可直观地看出,Swish 作为激活函数时收敛速度较快。

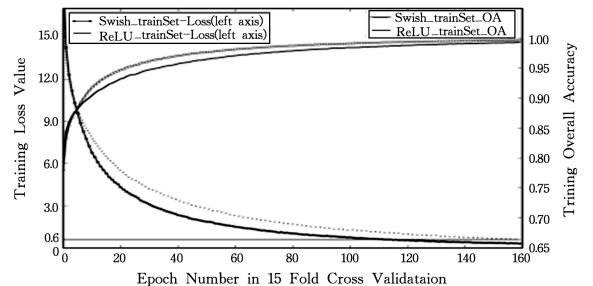


图 5 A 和 B 两组实验微调收敛速度的对比图

Fig. 5 Comparison of fine-tune convergence rate between experiment A and experiment B

由图 5 底端的贯穿直线可知,ReLU 作为激活函数时第 160 个 Epoch 的收敛效果与 Swish 作为激活函数时第 100 个 Epoch 的收敛效果相当。综合考虑分类效果和收敛速度,选用效果更好的 Swish 函数作为激活函数。

4.2.4 GRU 和 BiGRU 效果对比分析

表 2 中,实验 C 和实验 A 分别使用了 GRU 构建的 RNNs 和 BiGRU 构建的 RNNs。由图 4 中的总体指标对比可知,BiGRU 的效果更佳。选择两组实验的各折 OA 和 MF1 对实验结果进行进一步分析。

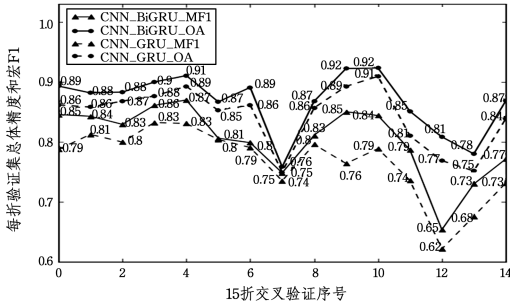


图 6 A 和 C 两组实验每折总体指标的对比图

Fig. 6 Per-fold comparison between experiment A and experiment C using overall metrics

图 6 中,“CNN-BiGRU”代表 A 实验。由图 6 可知,A 实验每折效果都优于 C 实验,这主要是由于单向 GRU 构建的 RNNs 只能从时序信号的一端逐个读取输入数据,因此对于任意时间节点,单向 GRU 所储存的数据只含有现在和以前的信息。事实上,每个睡眠阶段与其前后的睡眠阶段都具有密切的联系,BiGRU 构建的 RNNs 可以充分利用当前阶段与其前后两侧各阶段所存在的依赖关系。

4.2.5 不同微调数据集的对比分析

表 2 中实验 A 和实验 D 分别采用原始顺序数据集和 MSMOTE 重构后生成的类平衡数据集进行微调。由图 4 中的总体指标对比可知,原始顺序数据集用于微调时效果更佳。选择两组实验的各折 OA 和 MF1 对实验结果进行进一步分析。

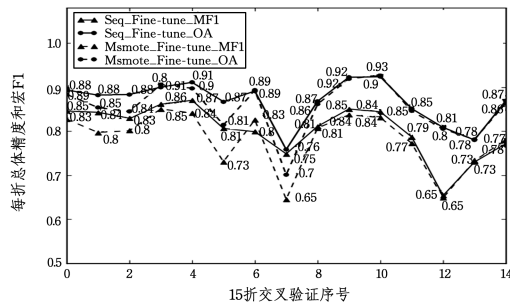


图 7 A 和 D 两组实验每折总体效果的对比图

Fig. 7 Per-fold comparison between experiment A and experiment D using overall metrics

图 7 中,“Seq_Fine-tune”代表 A 实验,由图 7 中各折总体指标的对比结果可知,A 实验各折效果较优,初步分析是由于 MSMOTE 生成的类平衡数据集中含有部分不属于任何类的数据,影响每折微调效果。并且在本文实验硬件(处理器:i5-7300HQ,显卡:NVIDIA GTX1050,显存:4GB,内存:8GB,

笔记本电脑)环境下,MSMOTE 生成的类平衡数据集用于微调时,单个 Epoch 所需的时间(约为 77 s)大于原始顺序数据集用于微调所需的时间(约为 43 s)。综上所述,本文选用原始顺序数据集用于微调。

4.2.6 不同预训练数据集的对比分析

表 2 中实验 A 和实验 E 分别采用 MSMOTE 和复制过采样技术生成的类平衡数据集进行预训练。由图 4 中 A 和 E 两组实验获得的总体分类效果可知,由 MSMOTE 重构后生成的类平衡数据集用于预训练的效果较优。分别对两组实验的每类实验结果进行进一步分析。表 3、表 4 为两组实验得出的总混淆矩阵及每类指标。图 8 为 A 实验的每类指标相对于 E 实验相同指标的变动百分比,正值代表实验 A 相对于实验 E 有所提升,负值代表下降。

表 3 实验 E 的总混淆矩阵及每类指标

Table 3 Overall confusion matrix and per-class metrics of experiment E

	预测类					每类指标/%		
	W	N1	N2	N3	REM	PR	RE	F1
实际类	7120	337	154	27	88	86.97	92.16	89.49
W	474	1023	777	21	404	53.06	37.90	44.22
N1	386	293	15195	809	575	88.51	88.05	88.28
N2	38	0	439	4986	4	85.30	91.20	88.15
N3	169	275	602	2	6302	85.47	85.74	85.61
REM								

表 4 实验 A 的总混淆矩阵及每类指标

Table 4 Overall confusion matrix and per-class metrics of experiment A

	预测类					每类指标/%		
	W	N1	N2	N3	REM	PR	RE	F1
实际类	7158	276	129	24	139	87.71	92.65	90.11
W	422	1206	661	15	395	65.33	44.68	53.07
N1	380	250	15194	810	624	90.10	88.04	89.06
N2	39	3	388	5037	0	85.52	92.13	88.70
N3	162	111	492	4	6581	85.04	89.54	87.23
REM								

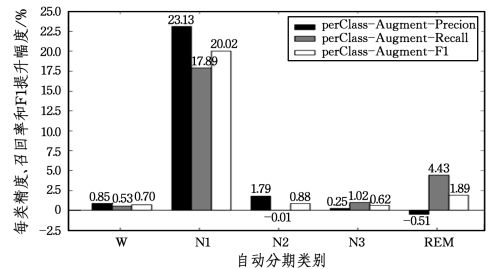


图 8 应用 MSMOTE 对各类指标的提升比例

Fig. 8 Augmentation of per-class metrics using MSMOTE

通过对比表 3 和表 4 的总混淆矩阵可知,MSMOTE 的应用使 N1 期被误判为其他各期的数目都有所下降,并且所有少数类的真阳数都在增加。由图 8 可直观地看出所有指标的升降比例,除 REM 期的精度和 N2 期的召回率略微下降(N2 期、REM 期的总体效果也是提升的)外,其他各类的每项指标都有所提升,其中占数据集总数最少的 N1 期的各指标提升幅度最大。综上所述,对于预训练数据集的重构,相较于传统的复制过采样技术,采用 MSMOTE 可在提升总体分类效果的同时大幅提升 N1 期的分类效果。

通过实验 D 和实验 E 分别与实验 A 的对比可知,由

MSMOTE 构建的类平衡数据只能用于预训练而不能用于微调。本文认为原因如下:预训练学习的是数据集结构,使用 MSMOTE 生成的类平衡数据能较好地满足学习要求;微调需要数据的精确性,若存在不属于任何类的干扰数据,则会学到本不存在的特征,从而影响训练效果。

4.3 模型评估

为论证本文模型的有效性,将本文实验 A 的结果与近两年使用相同数据集的研究成果进行比较。对比结果如表 5 所列。

表 5 最新成果与本文实验结果 A 的对比

Table 5 Comparison of experiment A with state-of-the-art results

方法	验证段数	总体指标/%		每类 F1/%				
		OA	MF1	W	N1	N2	N3	REM
文献[1]的方法	106376	89.73	62.44	97.89	10.67	80.10	52.48	71.08
文献[2]的方法	37022	78.94	73.73	71.57	47.04	84.60	84.03	81.40
文献[9]的方法	37022	74.77	69.82	65.41	43.69	80.59	84.88	74.54
文献[10]的方法	41950	82.02	76.87	84.71	46.56	85.89	84.81	82.38
本文实验结果 A	40500	86.85	81.63	90.11	53.07	89.06	88.70	87.23

表 5 中,文献[1]的总体精度较高的原因是其验证集中含有过多的 W 期(占比 70%),较大比例的 W 期在一定程度上提升了总体精度,但这种做法并没有使综合指标 MF1 值增大。综合来说,文献[1]的分类效果最差,本文方法的分类效果最优,尤其在少数类的分类效果上提升明显。

结束语 自动睡眠分期是脑电信号分类中的一个重要课题。本文提出一种以深度学习技术为基础的端到端自动睡眠分期模型。该模型中的多尺度 CNNs 可直接从原始脑电数据中提取时不变特征。同时,由于序列残差学习框架的嵌入,模型可以充分利用时不变特征中所蕴含的阶段间的依赖关系,来实现不同类型特征的融合。实验表明,Swish 激活函数的应用在保证模型分类精度略微提升的同时加快了训练收敛速度;MSMOTE 构建的类平衡数据用于预训练可大幅提高模型对少数类的识别率;模型能够从少量原始睡眠脑电数据中自动学习高级特征,并完成高精度的端到端分期,且分类效果较近年高水平的研究有所提升。

事实上,在数据量充足的情况下,少数类和多数类都可以提供足量的标签数据用于训练,不需要使用数据集扩增技术即可获得模型的最优性能。鉴于以上理论,本文模型可从以下两方面进行进一步优化:一方面,在现存睡眠数据集样本数不足的情况下,可通过对不同采样率的睡眠数据集进行降采样处理,来将不同睡眠数据集中的数据进行融合,以获取更多的可用于本文模型训练的标签数据集;另一方面,数据集扩充导致无法进一步提升模型性能时,可考虑增加卷积神经网络的深度对模型进行优化。

参考文献

[1] DA S T, KOZAKEVICIUS A J, RODRIGUES C R. Single-channel EEG sleep stage classification based on a streamlined set of statistical features in wavelet domain[J]. Medical & Biological Engineering & Computing, 2017, 55(2): 1-10.

[2] TSINALIS O, MATTHEWS P M, GUO Y K. Automatic sleep stage scoring using time-frequency analysis and stacked sparse autoencoders [J]. Annals of Biomedical Engineering, 2016, 44(5): 1587-1597.

[3] MONIKA P, POLAK A G. Effect of Feature Extraction on Automatic Sleep Stage Classification by Artificial Neural Network [J]. Metrology & Measurement Systems, 2017, 24(2): 229-240.

[4] LI C, CHAI Y M, NAN X F, et al. Research on Problem Classification Method Based on Deep Learning [J]. Computer Science, 2016, 43(12): 115-119. (in chinese)
李超, 柴玉梅, 南晓斐, 等. 基于深度学习的问题分类方法研究 [J]. 计算机科学, 2016, 43(12): 115-119.

[5] WANG Z X, TENG S H, LIU G D, et al. Hierarchical sparse representation with deep dictionary for multi-modal classification [J]. Neurocomputing, 2017, 253(C): 65-69.

[6] WANG Z G, ZHAO Z S, WENG S F, et al. Incremental multiple instance outlier detection [J]. Neural Computing & Applications, 2015, 26(4): 957-968.

[7] ZHAO Z S, FENG X, WEI F, et al. Learning Representative Features for Robot Topological Localization [J]. International Journal of Advanced Robotic Systems, 2013, 10(4): 1-12

[8] ZHANG Q L, ZHAO D, CHI X B. Review for Deep Learning Based on Medical Imaging Diagnosis [J]. Computer Science, 2017, 44(Z11): 1-7. (in Chinese)
张巧丽, 赵地, 迟学斌. 基于深度学习的医学影像诊断综述 [J]. 计算机科学, 2017, 44(Z11): 1-7.

[9] TSINALIS O, MATTHEWS P M, GUO Y K, et al. Automatic sleep stage scoring with single-channel EEG using convolutional neural networks [EB/OL]. [2016-10-05]. <https://arxiv.org/abs/1610.01683>.

[10] SUPRATAK A, DONG H, WU C, et al. DeepSleepNet: a model for automatic sleep stage scoring based on raw single-channel EEG [EB/OL]. [2017-08-03]. <https://arxiv.org/abs/1703.04046v2>.

[11] HE K M, ZHANG X Y, REN S Q, et al. Deep Residual Learning for Image Recognition [C] // Computer Vision and Pattern Recognition. IEEE, 2016: 770-778.

[12] RAMACHANDRAN P, ZOPH B, LE Q V. Swish: a self-gated activation function [EB/OL]. [2017-10-27]. <https://arxiv.org/abs/1710.05941v2>.

[13] HU S G, LIANG Y F, MA L T, et al. MSMOTE: Improving Classification Performance When Training Data is Imbalanced [C] // International Workshop on Computer Science & Engineering. IEEE Computer Society, 2009: 13-17.

[14] COHEN M X. Analyzing neural time series data: theory and practice [M]. Massachusetts: MIT Press, 2014.

[15] IOFFE S, SZEGEDY C. Batch normalization: accelerating deep network training by reducing internal covariate shift [C] // International Conference on Machine Learning. 2015: 448-456.

[16] JOZEFOWICZ R, ZAREMBA W, SUTSKEVER I. An empirical exploration of recurrent network architectures [C] // International Conference on Machine Learning. 2015: 2342-2350.