

华中科技大学
HUAZHONG UNIVERSITY OF SCIENCE AND TECHNOLOGY

人机交互技术课程设计报告

基于 EEG 的运动想象分类

姓 名： 陈星诺

学 号： U201911366

院 系： 人工智能与自动化学院

班 级： 校交 1901 班

2022 年 6 月 30 日

多种方法下基于 EEG 的运动想象状态分类

陈星诺¹

(1. 华中科技大学 人工智能与自动化学院, 湖北 武汉 430071;

摘要: 基于脑电图的运动图像的分类是脑机接口研究中的一项重要非侵入性应用, 但是由于数据量少、数据间差异大等问题, 设计泛化性能强的模型成为了主要的研究目标。本实验基于传统的机器学习方法 XGBoost, 经典神经网络方法 LSTM 以及针对基于 EEG 的运动想象状态分类任务所设计的小型卷积神经网络 EEG-Inception 等多种不同的机器学习方法利用一个小型的数据集实现运动想象状态分类任务。最后, 采用留一法对实验中所设计的模型进行了验证, 并对比了多种方法在该任务上的表现, 同时简要分析了实验结果以及对进一步的研究思路提出了思考。

关键词: 脑电图; 运动想象分类; 机器学习; 回归任务

1 引言

“脑机接口”研究的核心是在大脑和外部设备之间建立直接的联通通路, 并通过计算机信息处理技术理解人的意图, 并将此意图转化为对外界的控制命令, 实现大脑对外部世界的直接控制 [1]。脑电图 (Electroencephalography, EEG) 信号属于非侵入式脑机接口技术, 作为一种特殊而复杂的生物电信号, 反映了大脑的功能状态, 是由脑内亿万神经元活动而引起的头皮表面电位变化, 检测这些电位的变化对研究大脑的功能状态非常重要。有效提取脑电信号中蕴藏的信息, 可以更深入地了解大脑的功能活动。

近年来, 研究脑电信号的任务不断增长, 主要包括: 运动想象数据, 情绪识别数据, 误差相关电位, 视觉诱发电位, 事件相关电位, 临床脑电图等。其中, 运动想象通过想象身体的运动 (例如, 左手和右手运动) 来触发神经活动。解码正确的 MI 神经活动可使患有运动神经元疾病 (例如中风、帕金森病) 的患者在外部设备的帮助下恢复部分身体运动技能。除了身体康复, 基于脑电图的运动控制流行应用还包括轮椅控制 [2]、机械臂操作 [3] 和四轴飞行器操纵 [4]。

现有的关于基于 EEG 的运动想象分类的方法主要包括传统的机器学习方法以及神经网络方法两大类。其中, 经典机器学习算法过程主要包括信号预处理、特征提取和特征分类三部分。信号预处理的目标是去除噪声和伪影, 典型方法为独

立相关分析、典型相关分析和带通滤波等, 针对特征提取则有研究选用了常用的主成分分析。至于特征分类, 则主要选用了传统分类器, 如 K 近邻 (KNN) [5]、线性判别分析 (LDA) [6] 和支持向量机 (SVM) [7] 等。与基于神经网络的算法相比, 经典机器学习算法相对简单并且有着更快的训练周期。然而, 受限于特征提取能力的局限性, 经典机器学习算法基于 EEG 的运动想象分类结果相对有限。

由于机器学习算法的局限性, 近年来越来越多的研究转向采用神经网络的方法实现运动想象分类任务, 主要的模型选型包括了特征提取能力较强的卷积神经网络 (CNN) 模型以及对序列数据处理表现较好的循环神经网络 (RNN) 和其变种长短期记忆网络 (LSTM) 等。而本实验为讨论各个模型的表现并获得较高正确率的输出, 采用了多种数据处理和分类方法, 其中包括了传统机器学习方法 XGBoost, 经典神经网络方法 LSTM 以及针对基于 EEG 的运动想象分类任务所设计的轻量级卷积神经网络 EEG-Inception 网络, 在一个小型数据集上进行了模型的训练并采用了留一法对模型的表现进行了测试。

本报告的余下部分如下所述: 第二部分介绍了本实验中所采用方法的基本原理, 第三部分介绍了在实验具体内容以及在实验过程中针对研究问题所进行的进一步优化, 第四部分展示了实验结果并总结了结论, 最后一个部分提出了对进一步探索探究

问题的思考。

2 原理说明

2.1 XGBoost

XGBoost 是一种集成学习方法, 其算法思想就是不断地添加树, 不断地进行特征分裂来生长一棵树, 每次添加一个树, 其实是学习一个新函数, 去拟合上次预测的残差。当我们训练完成得到 k 棵树, 如果要预测一个样本, 其实就是根据这个样本的特征, 在每棵树中会落到对应的一个叶子节点, 每个叶子节点就对应一个预测值, 最后只需要将每棵树对应的分数加起来就是该样本的最终预测值。其目标函数考虑正则化项, 定义如下:

$$L(\phi) = \sum_i l(y_i, \hat{y}_i) + \sum_k \Omega(f_k)$$

其中, $\Omega(f_k) = \gamma T + \frac{1}{2} \lambda \|w\|^2$, 其中 T 为叶子结点数, w 为叶子权重值, 可以也通过正则化项防止过拟合问题, 正则化项表示着模型的复杂度, 可以每个树的叶子节点的个数 (叶子节点个数越少模型越简单) 和叶子节点的权重值 w (值越小模型越简单)。由于新生成的树是要拟合上次预测的残差的, 即当生成 t 棵树后, 预测值可以表示成:

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + f_t(x)$$

由此, 目标函数可以改写为

$$L^{(t)} = \sum_{i=1} l\left(y_i, \left(\hat{y}_i^{(t-1)} + f_t(x_i)\right)\right) + \Omega(f_t)$$

所以优化第 t 棵树即为找到一个 f_t 能使得上述目标函数最小, XGBoost 的想法是利用其在 $f_t = 0$ 处的泰勒二阶展开近似, 即目标函数可以近似为:

$$\mathcal{L}^{(t)} \simeq \sum_{i=1}^n \left[l\left(y_i, \hat{y}_i^{(t-1)}\right) + g_i f_t(\mathbf{x}_i) + \frac{1}{2} h_i f_t^2(\mathbf{x}_i) \right] + \Omega(f_t)$$

其中 g_i 和 h_i 分别为一阶和二阶导数。最后 XGBoost 利用顿法进行梯度更新。与 GBDT 不同的是, xgboost 中可以自动处理空缺值, 自动学习空缺值的分裂方向, 而 GBDT 不允许包含空缺值。

2.2 LSTM

LSTM 算法全称为 Long short-term memory, 是一种特定形式的 RNN。RNN 在处理长期依赖 (即时间序列上距离较远的节点) 时会遇到困难, 因为计算距离较远的节点之间的联系时会涉及雅可

比矩阵的多次相乘, 这会带来梯度消失或者梯度膨胀的问题, 在不断的研究过程中, 门限 RNN 作为解决该问题的一个有效手段。

LSTM 是一种门限 RNN, 通过增加输入门限, 遗忘门限和输出门限, 使得自循环的权重是变化的, 所以在模型参数固定的情况下, 不同时刻的积分尺度可以动态改变, 从而避免了梯度消失或者梯度膨胀的问题。LSTM 中状态的更新可以由下面六个公式表示:

$$\begin{aligned} f_t &= \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \\ i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\ \tilde{C}_t &= \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \\ C_t &= f_t * C_{t-1} + i_t * \tilde{C}_t \\ o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \\ h_t &= o_t * \tanh(C_t) \end{aligned}$$

其中 f_t 表示遗忘门限, i_t 表示输入门限, \tilde{C}_t 表示前一时刻 cell 状态、 C_t 表示 cell 状态, o_t 表示输出门限, h_t 表示当前单元的输出, H_{t-1} 表示前一时刻单元的输出。

与前馈神经网络类似, LSTM 网络的训练同样采用的是误差的反向传播算法 (BP), 不过因为 LSTM 处理的是序列数据, 所以在使用 BP 的时候需要将整个时间序列上的误差传播回来。LSTM 本身又可以表示为带有循环的图结构, 在这个带有循环的结构上使用反向传播时称之为 BPTT 算法 (back-propagation through time)。

2.3 EEG-Inception

EEG-Inception [8] 新的卷积神经网络, 可以改进基于时间相关电位 (ERP) 的脑机接口的精度和校准时间。Inception 模块是由文献 [9] 等人为计算机视觉所提出的, 允许通过并行使用不同内核大小的卷积层对输入数据进行多尺度分析。实践证明, 该结构能够以较小的计算量提取出更丰富的特征, 在保持合理的训练和评估时间的前提下提高性能。而 EEG 是由不同时间长度的瞬时和振荡模式组成, 反映了大脑正在进行的活动, 通过 Inception 模块进行多尺度分析特别适合于 EEG 信号。EEG-Inception 的网络结构如下图所示:

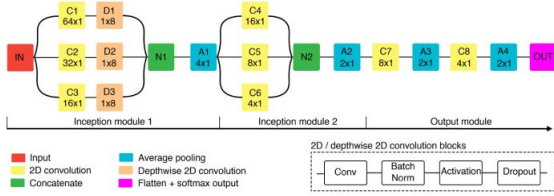


图 1 EEG-Inception 网络结构

其中各层的详细尺寸以及参数如下图所示:

TABLE IV
EEG-INCEPTION ARCHITECTURE DETAILS

Block	Type	Filters	Depth	Kernel	Padding	Output shape	Connected to	Role
IN	Input	—	—	—	—	$128 \times 8 \times 1$	C1, C2, C3	Input
C1	Conv2D	8	—	64×1	Same	$128 \times 8 \times 8$	D1	Temporal analysis
D1	DepthwiseConv2D	—	2	1×8	Valid	$128 \times 1 \times 16$	N1	Spatial analysis
C2	Conv2D	8	—	32×1	Same	$128 \times 8 \times 8$	D2	Temporal analysis
D2	DepthwiseConv2D	—	2	1×8	Valid	$128 \times 1 \times 16$	N1	Spatial analysis
C3	Conv2D	8	—	16×1	Same	$128 \times 8 \times 8$	D3	Temporal analysis
D3	DepthwiseConv2D	—	2	1×8	Valid	$128 \times 1 \times 16$	N1	Spatial analysis
N1	Concatenate	—	—	—	—	$128 \times 1 \times 48$	A1	Concatenation
A1	AveragePooling2D	—	—	4×1	Same	$32 \times 1 \times 48$	C4, C5, C6	Concatenation
C4	Conv2D	8	—	16×1	Same	$32 \times 1 \times 8$	N2	Temporal analysis
C5	Conv2D	8	—	8×1	Same	$32 \times 1 \times 8$	N2	Temporal analysis
C6	Conv2D	8	—	4×1	Same	$32 \times 1 \times 8$	N2	Temporal analysis
N2	Concatenate	—	—	—	—	$32 \times 1 \times 24$	A2	Concatenation
A2	AveragePooling2D	—	—	2×1	—	$16 \times 1 \times 24$	C7	Dimension reduction
C7	Conv2D	12	—	8×1	Same	$16 \times 1 \times 12$	A3	Temporal analysis
A3	AveragePooling2D	—	—	2×1	—	$8 \times 1 \times 12$	C8	Dimension reduction
C8	Conv2D	6	—	4×1	Same	$8 \times 1 \times 6$	A4	Temporal analysis
A4	AveragePooling2D	—	—	2×1	—	$4 \times 1 \times 6$	C7	Dimension reduction
OUT	Dense	—	—	—	—	2	—	Softmax output

Column "Type" describes the class used to implement each block in Keras framework. It should be taken into account that this implementation may vary across different frameworks. All convolutional blocks (i.e., Conv2D and DepthwiseConv2D) include batch normalization, activation and dropout regularization. The model has 15154 parameters, of which 14926 are fitted during training.

图 2 EEG-Inception 网络详细参数

可以看出, EEG-Inception 网络结构上包含了 2 个 Inception 模块后接了两个卷积层, 最后全连接层进行分类。第一个 Inception 模块对信号进行三种时间维度的特征提取, 对应三个卷积核 C1、C2 和 C3 大小为 64, 32 和 16。输入数据的采样率为 $128Hz$, 对应每个卷积分别对应 $500ms$ 、 $250ms$ 和 $125ms$ 的时间窗。紧接着 D1、D2 和 D3 卷积层为导联维度的深度卷积。之后将不同卷积的输出拼接, 进行平均池化降低维度。第二个 Inception 模块三个卷积分支同样对应 $500ms$ 、 $250ms$ 和 $125ms$ 的时间窗, 对应的卷积核大小分别为 16, 8 和 4。该模块在更高的特征抽象层次提取所有导联的时序特征。之后将不同卷积的输出拼接, 进行平均池化降低维度。最后的两个卷积层用于提取特征中对于分类有用的模式, 同时使用平均池化降低特征维度, 最后全连接以及 softmax 进行分类输出。

在文献 [8] 发表后, 同一时间段的另一篇文章 [10] 也将 Inception 模块运用到了 EEG 网络中, 并讨论运动想象分类任务, 但是由于数据集的采样问题, 该研究中的数据频率成分更丰富, 模型采用了更多的卷积层特征提取, 不适用于本次实验数据, 但是为提高模型的泛化性能, 本次实验借鉴了该篇文献中的数据增强思路, 具体方法将在第四部分详细叙述。

3 实验说明

3.1 数据集说明

3.1.1 数据采集说明

采集过程中, 受试者坐在电脑前的椅子上。采集开始时, 电脑屏幕上会出现一个固定的叉, 提示对象准备, 持续 3s; 然后, 一个指向某一个方向的箭头作为视觉提示在屏幕上出现 5s, 在此期间, 受试者根据箭头的方向执行特定的运动想象任务; 然后, 视觉提示从屏幕上消失, 受试者短暂休息 2s。紧接着下一个测试过程开始。

数据来自 8 个健康的受试者 (训练受试者 S1 ~ S4, 测试受试者 S5 ~ S8), 每一个受试者执行两类运动想象任务: 右手和双脚, 脑电信号由一个 13 通道的脑电帽以 $512Hz$ 的频率记录得到。我们提供了经过预处理后的数据: 下采样到了 $250Hz$, 带通滤波至 $8 - 32Hz$, 划分每一次视觉提示出现后的 $0.5 - 3.5s$ 之间的 EEG 信号作为一个测试过程。每个用户包含 200 个测试过程 (右手和双脚各 100 个)。

3.1.2 数据格式说明

数据以 .npz 和 .mat 格式提供, 包含: X——预处理后的 EEG 信号, 维度为 [测试过程 * 通道 * 采样点], 即 $[200 \times 13 \times 750]$ 以及 y——类别标签向量, 值得注意的是, 测试数据 (S5 ~ S8) 不包含此变量。

3.2 传统机器学习方法: XGBoost

3.2.1 数据预处理

XGBoost 模型直接调用了 xgboost 包中的 XGBRegressor 类, 为了能将数据传入模型中进行训练, 首先将每一个测试过程的 13×750 大小的数据展平为 1×9750 大小的一维数据, 对训练集的标签数据没有进行处理。在数据集的划分问题上, 选用 S1 受试者数据作为测试集进行留一测试, 余下三位受试者 S2 ~ S4 作为训练数据对模型进行训练。

EEG 的运动想象问题的难点在于如何解决模型的过拟合问题, 为了避免过拟合现象过于严重, 在数据预处理过程中, 首先选用了 K-Means 聚类算法排除掉离群数据, 将离群数据视作数据集中的噪声。K-Means 算法选取 K 个点做为初始聚集的簇心 (也可选择非样本点), 分别利用下述公式计

算每个样本点到 K 个簇核心的距离, 找到离该点最近的簇核心, 将它归属到对应的簇。

$$d(x, y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \cdots + (x_n - y_n)^2}$$

$$= \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

所有点都归属到簇之后, M 个点就分为了 K 个簇。之后重新计算每个簇的重心 (平均距离中心), 将其定为新的“簇核心”。反复迭代以上步骤, 直到达到某个中止条件。

在本数据集中, $K = 2$, 即有两类标签, 通过 K-Means 算法可以得到两类样本, 其中每个簇都会存在一个簇中心 $X_c = \{x_0, x_1, \dots, x_n\}$, 计算簇中每个点到簇中心的欧式距离, 不满足下述条件的样本点剔除出训练集。

$$d(X, X_c) < \alpha \times d(X, X_c)_{max}$$

其中 α 为一个超参数, 取值范围为 $(0, 1)$, 如果取值过大则聚类清洗数据的效果较差, 反之则训练样本过少训练效果不理想, 本实验中 $\alpha = 0.9$ 。

同样, 神经网络中经常使用 Dropout 层解决过拟合问题, 其作用是在前向传播的时候, 让某个神经元的激活值以一定的概率 p 停止工作, 通过这样可以使模型泛化性更强, 因为它不会太依赖某些局部的特征。针对于隐层, Dropout 的实际效果是减少部分神经元数量从而降低模型的复杂度达到过拟合效果, 而本实验将 Dropout 层运用在了数据预处理过程中, 即依概率 p 忽略部分维度的数据特征。本实验中 $p = 0.05$

3.2.2 正则化项

正如第二部分所述, XGBoost 中采用了正则化项防止过拟合, 其正则化的目的是限制生成树的复杂度, 主要限制角度为树里面叶子节点个数和对参数进行经典的二阶正则化, XGBoost 也支持一阶正则化, 但容易优化叶子节点权重为 0, 一阶正则化的正则化项可表示为:

$$\Omega = \gamma T + \frac{1}{2} \alpha \sum_{j=1}^T |w_j|$$

本实验中, 同时使用了一阶正则化和二阶正则化, 其参数分别为 $\alpha = 0.5$ $\lambda = 1$, 正则化项表达如下:

$$\Omega = \gamma T + \frac{1}{2} \alpha \sum_{j=1}^T |w_j| + \frac{1}{2} \lambda \sum_{j=1}^T w_j^2$$

除正则化项以外, 本实验还限制了树的深度 (最大深度为 4 层) 以及树的个数 (一共 300 个) 等方式以解决过拟合问题, 提高模型的泛化性能。

3.3 经典神经网络: LSTM

针对 LSTM 网络的数据, 本实验采用了与 XGBoost 方法一样的方式在训练之前对数据进行了展平、聚类、Dropout 等处理以防止模型过拟合。模型直接基于 PyTorch 深度学习框架中内置的 LSTM 模型构建, 只包括一层隐层, 其中神经元个数为 128 个。模型采用 Adam 算法进行梯度更新, 学习率为 0.001, 所选用的损失函数为均方损失函数, 其计算方式表达如下:

$$\text{loss}(\mathbf{x}_i, \mathbf{y}_i) = (\mathbf{x}_i - \mathbf{y}_i)^2$$

针对 XGBoost 和 LSTM 两个模型, 对标签都没有进行预处理 (如 EEG-Inception 将进行的 one-hot 编码), 其标签为集合 $\{0, 1\}$ 中的一个整数, 而模型的输出为区间 $[0, 1]$ 中的一个数, 由数据集的特点可以知道, 两类样本的个数应该相等, 所以在进行最终的打标时, 把 0 和 1 的标签个数相等作为先验知识进行启发式输出, 输出逻辑如下所示, 其中 mid 为模型输出的中位数:

$$y_{pred} = 1 \quad \text{if} \quad y_{pred} > mid \quad \text{else} \quad 0$$

3.4 针对 EEG 设计的小型卷积神经网络: EEG-Inception

3.4.1 数据预处理

与前两个模型不同, 由于卷积神经网络可以处理一个样本的多通道数据, 所以针对于输入数据并没有提前进行展平, 而是保留其 13×750 维的格式输入进神经网络。而对输出则进行了 one-hot 编码, 其编码逻辑如下:

$$0 \rightarrow [1, 0]$$

$$1 \rightarrow [0, 1]$$

即模型的输出为一个二维向量, 其每个维度的值均通过模型最后的 Softmax 层归一化于 $[0, 1]$ 区间内。

正如第二部分所述, 文献 [10] 中对输入数据进行了增强, 其基本思想为以扩展数据大小来最小化过度拟合问题, 由于实验过程繁琐冗长并且有一定风险, 所以数据量很小一直是影响着 EEG 相关模型的泛化性能的重要因素, 与标准计算机视觉数据

增强不同, EEG 信号是非平稳的, 不能旋转、拉伸或缩放, 因为这些方法会改变时间序列信号的属性, 所以针对 EEG 数据的增强一般选用噪声添加的方式。文献 [10] 的做法是从一次试验中提取 $100Hz$ 以上的信号 (视作噪声), 然后将其应用于另一次试验, 这是因为文献 [11] 认为大多数大脑活动存在于 $0 - 100Hz$ 的频率范围内, 而 $100Hz$ 以上的频率可以被认为是伪影和噪音。增强后的数据可以表示为:

$$S_{aug}(i) = S_0(i) - S_n(i) + S_n(k) \quad (1)$$

其中, $S_n(i)$ 为从原始信号中提取出的噪声, $S_n(k)$ 为从其他信号中提取出来的噪声, 噪声的提取通过一个 8 阶巴特沃斯高通滤波器完成。

但是正如数据集说明部分所述, 数据已经被规定在了 $8Hz$ 到 $32Hz$ 区间内, 所以本实验不严格按照文献 [10] 的方式将 $100Hz$ 视作截止频率, 而是将非 $11Hz - 29Hz$ 以外的部分视作噪声, 按照上述公式进行数据增强, 将增强后的数据加入训练数据集丰富数据量, 讨论数据增强前后对模型的影响。

3.4.2 模型训练

模型同样采用 Adam 优化器在 0.001 的学习率下进行训练, 选用了针对多分类问题设计的交叉熵损失函数作为训练过程的损失函数, 其函数计算公式如下:

$$\text{Loss} = - \sum_{i=1}^n y_i \log y'_i$$

其中 y_i 是真实值 y'_i 是预测值。

3.5 集成学习: 投票器

通过模型融合, 可以取长补短, 综合个体学习器的优势, 能降低预测误差、优化整体模型性能。而且个体学习器的准确性越高、多样性越大, 模型融合的提升效果越好。如前文所述, XGBoost 和 LSTM 两个模型的输出格式相同, 可以对模型直接进行融合获得性能更好的模型。而针对最后一个模型, 需要先将输出从 one-hot 编码重新映射回 $\{0, 1\}$ 集合中再进行集成, 投票器的输出表示如下:

$$\hat{y} = \alpha_0 y_{xgboost} + \alpha_1 y_{lstm} + \alpha_2 y_{eeg}$$

其中, α 为各个模型的权重, 可以由各个模型的正确率等性能指标确定, 这种投票器的方式称为软投票, 相比于“少数服从多数”的硬投票, 软投

票可以设置模型权重。

4 实验结果

4.1 模型训练

实验中除了投票器之外, 一共涉及了 XGBoost、LSTM 和 EGG-Inception 三种方法, 其中 XGBoost 调用了开源的模型进行训练和测试, 其余两个模型均通过 Adam 算法获得神经网络中的各个参数, 为说明模型的收敛性, 利用 tensorboard 输出了模型在训练过程中的损失函数的变化, 结果如下:

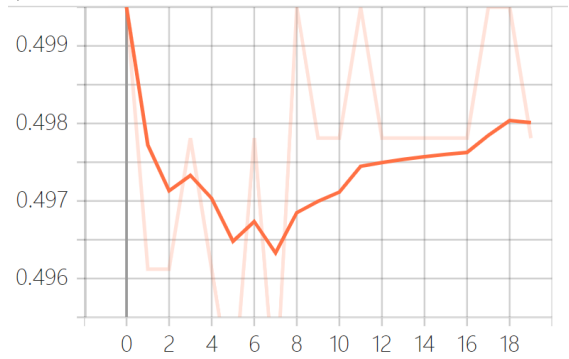


Figure 1: LSTM

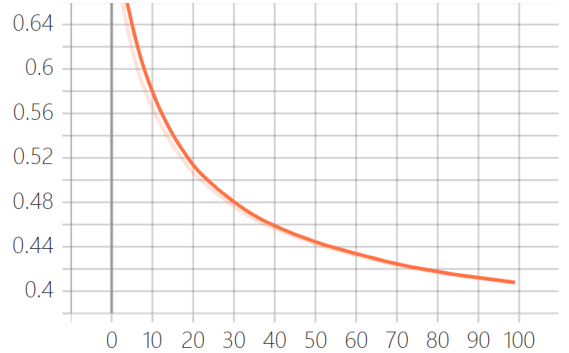


Figure 2: EGG-Inception

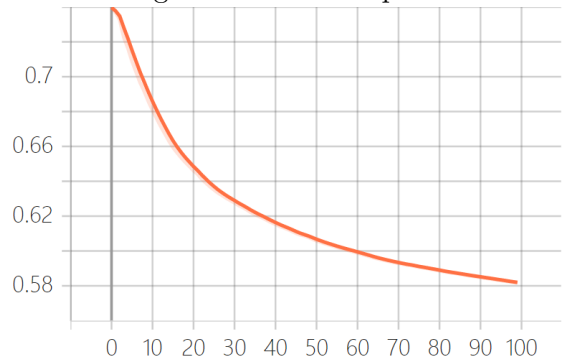


Figure 3: Data_augmentation+EGG-Inception

可以看出随着迭代轮次的增加, 三个模型均已基本收敛, 值得一提的是, LSTM 模型的随着训练轮次的进一步增加, 模型的损失函数会在几个

epoch 中突然增大,最后会稳定在 0.45 左右。

4.2 实验结果

如数据集说明所述,模型在测试阶段均采用留一法,将 S1 受试者作为测试样本,使用利用 S2、S3 和 S4 三个样本所训练的模型进行测试,最后的结果如下:

表 1 实验结果

Table 1 Results

XGBoost	LSTM	EEG	EEG_aug	集成学习
0.555	0.52	0.58	0.57	0.58

可以看出,传统的机器学习方法和神经网络孰优孰劣并没有明确的关系,在数据量较小的情况下未经过具体问题设计的神经网络可能由于模型过于复杂表达能力过强而发生拟合的情况导致在测试集上的表现更差。

为了说明数据预处理的有效性,即聚类算法和 Dropout 层的有效性,表中所给的结果是在预处理后的结果,而完全不预处理即同时放弃聚类和 Dropout 层,以 XGBoost 模型为例,其正确率只有 0.515,相比于预处理后的数据有着明显的下降。

而就数据增强而言,在本例上数据增强并没有给到模型正向提升,究其原因是因为本例数据已经通过了滤波成分限制在了 $8Hz$ — $32Hz$,而原始文献是根据信号有效成分的频率对原始信号进行有针对性地滤波从而能获得质量更高的样本,而本例滤波后的频率成分本来较少同时均处在信号的有效区间,所以虽然通过这样的方式充盈了样本量,但是样本中的有效信息也受到了影响。

此外,利用先验知识(即两类的比例为 1:1)所提出的结果输出方式能一定程度上提高正确率,以 LSTM 模型为例,在采用传统的四舍五入法输出结果是,0 和 1 两类样本的正确个数分别为 13 和 87,正确率是 50%,明显可以看出在两类样本上的表现出入较大。而加入先验知识后,将大于中位数的结果视为 1,反之则为 0,两类样本的正确个数均为 52,正确率也是 52%,效果得到了显著提升。

其中集成学习采用的是 XGBoost、LSTM 和 EEG 三个模型,集成学习的表达式如下,其权重由各自单独的正确率确定。

$$\hat{y} = 0.35 * y_{xgboost} + 0.65 * y_{lstm} + 0.1 * y_{eeg}$$

4.3 测试数据

对 S5 到 S8 四个受试者的测试数据在 EEG 模型上进行测试并输出结果文件为.csv 文件,CSV 结果文件只包含 4 列,每一列即预测的测试用户上的 200 个测试过程的标签。选用 EEG-Inception 模型的原因是因为它模型结构不复杂,并且表现较好。值得说明的是,S1 也作为训练数据参与了训练,测试结果详见附件。训练过程中的损失函数变化如下:

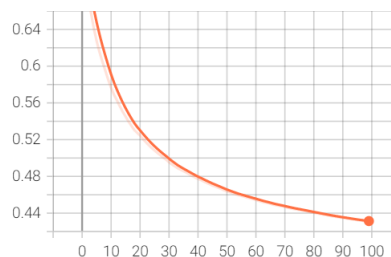


图 4 利用前四个样本数据训练模型时损失函数的变化

5 总结与展望

整个实验主要利用给定的数据集研究了基于 EEG 的运动想象分类任务,实验的主要工作以及创新点在于:

1. 分别测试了传统机器学习、经典神经网络以及 EEG 中的神经网络中的经典方法: XGBoost、LSTM 和 EEG-Inception,并对模型进行了训练以及在测试数据上验证,最后对结果的合理性进行了分析;
2. 利用聚类/Dropout 层等方式进行了数据清洗,测试了在给定数据集上的数据增强方式,将数据增强前后的结果进行了对比,分析了现象的原因;
3. 提出了新的结果输出方式,针对数据集的总体分布修改了传统的结果输出方式,一定程度上提高了正确率;
4. 利用提出的模型对测试样本输出了结果;

整个研究方向的主要阻力即为模型的过拟合问题,为了解决这个问题,首先可以从数据出发,研究领域需要规范数据格式,得到更为规范化的统一的数据集,如计算机视觉领域的 ImageNet,这样各个实验室所采集的数据可以结合在一起利用,大

大提高了数据的利用效率。此外需要结合实际问题的, 设计出泛化性能更强的模型, 也可以从时域、频域等多个角度结合起来研究问题从而获得更好的结果。

参考文献:

- [1] Krucoff M O , Shervin R , Slutzky M W , et al. Enhancing Nervous System Recovery through Neurobiologics, Neural Interface Training, and Neurorehabilitation[J]. *Frontiers in Neuroscience*, 2016, 10(Pt 3).
- [2] Yu Y , Zhou Z , Liu Y , et al. Self-Paced Operation of a Wheelchair Based on a Hybrid Brain-Computer Interface Combining Motor Imagery and P300 Potential[J]. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 2017, PP(99):1-1.
- [3] Rakshit A , Konar A , Nagar A K . A hybrid brain-computer interface for closed-loop position control of a robot arm[J]. *IEEE/CAA Journal of Automatica Sinica*, 2020, 7(5):1344-1360.
- [4] Khan M J , Zafar A , Hong K S . Hybrid EEG-NIRS based active command generation for quadcopter movement control[C]// 2016 International Automatic Control Conference (CACCS). IEEE, 2016.
- [5] Bhaduri S , Khasnobish A , Bose R , et al. Classification of lower limb motor imagery using K Nearest Neighbor and Na e-Bayesian classifier[C]// 2016 3rd International Conference on Recent Advances in Information Technology (RAIT). IEEE, 2016.
- [6] Wu S L , Wu C W , Pal N R , et al. Common spatial pattern and linear discriminant analysis for motor imagery classification[C]// *IEEE Symposium on Computational Intelligence*. IEEE, 2013.
- [7] Ma Y , Ding X , She Q , et al. Classification of Motor Imagery EEG Signals with Support Vector Machines and Particle Swarm Optimization[J]. *Computational and Mathematical Methods in Medicine*, 2016, (2016-5-30), 2016, 2016:1-8.
- [8] Santamaria-Vazquez E , Martinez-Cagigal V , Vaquerizo-Villar F , et al. EEG-Inception: A Novel Deep Convolutional Neural Network for Assistive ERP-based Brain-Computer Interfaces[J]. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 2020, PP(99):1-1.
- [9] Szegedy C , Liu W , Jia Y , et al. Going Deeper with Convolutions[J]. *IEEE Computer Society*, 2014.
- [10] Zhang C , Kim Y K , Eskandarian A . EEG-Inception: An Accurate and Robust End-to-End Neural Network for EEG-based Motor Imagery Classification[J]. 2021.
- [11] Muthukumaraswamy S D . High-frequency brain activity and muscle artifacts in MEG/EEG: a review and recommendations[J]. *Frontiers in Human Neuroscience*, 2013, 7.