**4.14 两阶段模型训练方法（brainda.algorithms.utils.model\_selection）**

**4.14.1 Functions: model\_training\_two\_stage**

使用带有K折交叉验证的两阶段训练策略。此函数为神经网络模型实现了两阶段训练过程[1]，通常用于脑机接口（BCI）任务。在第一阶段，模型在K折交叉确认分割上训练，并根据验证损失或准确性提前停止。在第二阶段，使用训练和验证数据对第一阶段的最佳模型进行微调。训练后的模型为每个折叠保存。

[1] Schirrmeister RT, Springenberg JT, Fiederer LDJ, Glasstetter M, Eggensperger K, Tangermann M, Hutter F, Burgard W, Ball T. Deep learning with convolutional neural networks for EEG decoding and visualization. Hum Brain Mapp. 2017 Nov;38(11):5391-5420.

**model\_training\_two\_stage***(model, criterion, optimizer, lr\_scheduler, frist\_epochs, eary\_stop\_epoch, second\_epochs, batch\_size, X\_train, Y\_train, kfolds, device, model\_name, subject, model\_savePath)*

带有K折交叉验证的两阶段训练策略

**Parameters**

**model :** *torch.nn.Module*

要训练的PyTorch模型（例如TCNet\_Fusion或EEGNet）。

**criterion:***callable*

训练的损失函数（例如torch.nn.CrossEntropyLoss）。

**optimizer:***torch.optim.Optimizer*

用于更新模型参数的优化器（例如torch.optim.Adam）。

**lr\_scheduler:***torch.optim.lr\_scheduler.\_LRScheduler or None*

训练期间调整学习率（例如CosineAnnealingLR）。如果为“无”，则不应用调度程序。

**frist\_epochs:** *int*

第一个训练阶段的最大迭代次数。

**eary\_stop\_epoch:** *int*

如果验证准确性没有提高，则等待提前停止的迭代次数。

**second\_epochs:** *int*

第二训练阶段的最大迭代次数。

**batch\_size:** *int*

DataLoader每批的样本数。

**X\_train:** *torch.Tensor or np.ndarray*

具有[n\_samples，n\_channels，n\_samples]形状的训练数据。 **Y\_train:** *torch.Tensor or np.ndarray*

使用形状[n\_samples]训练标签。 **kfolds:** *int*

K折交叉验证的折数。 **device:** *torch.device*

用于执行计算的设备（CPU或GPU）。 **model\_name:** *str*

用于保存目的的模型名称（例如，“TCNet\_Fusion”）。 **subject:** *str or int*

保存模型文件的主题标识符。 **model\_savePath:** *str*

保存训练模型文件的目录路径。

**Returns**

None。

将模型训练到位，并将每个折叠的模型状态保存到model\_savePath。打印训练进度和各折叠的平均评估精度。

**4.14.2 Functions: model\_training\_two\_stage\_up**

使用具有K倍交叉验证的改进两阶段训练策略。该函数为神经网络模型实现了两阶段训练过程，针对脑机接口（BCI）任务进行了优化。在第一阶段，模型在K倍交叉验证分割上进行训练，根据验证损失和准确性提前停止，以选择最佳模型。在第二阶段，使用训练和验证数据对最佳模型进行微调。每个折叠都会保存训练好的模型。与标准的两阶段训练相比，该版本优先考虑验证损失较低的模型，同时确保高精度。

[1] Liang G, Cao D, Wang J, Zhang Z, Wu Y. EISATC-Fusion: Inception Self-Attention Temporal Convolutional Network Fusion for Motor Imagery EEG Decoding. IEEE Trans Neural Syst Rehabil Eng. 2024;32:1535-1545.

**model\_training\_two\_stage\_up***(model, criterion, optimizer, lr\_scheduler, frist\_epochs, eary\_stop\_epoch, second\_epochs, batch\_size, X\_train, Y\_train, kfolds, device, model\_name, subject, model\_savePath)*

带有K折交叉验证的改进两阶段训练策略

**Parameters**

**model :** *torch.nn.Module*

要训练的PyTorch模型（例如TCNet\_Fusion或EEGNet）。

**criterion:***callable*

训练的损失函数（例如torch.nn.CrossEntropyLoss）。

**optimizer:***torch.optim.Optimizer*

用于更新模型参数的优化器（例如torch.optim.Adam）。

**lr\_scheduler:***torch.optim.lr\_scheduler.\_LRScheduler or None*

训练期间调整学习率（例如CosineAnnealingLR）。如果为“无”，则不应用调度程序。

**frist\_epochs:** *int*

第一个训练阶段的最大迭代次数。

**eary\_stop\_epoch:** *int*

如果验证准确性没有提高，则等待提前停止的迭代次数。

**second\_epochs:** *int*

第二训练阶段的最大迭代次数。

**batch\_size:** *int*

DataLoader每批的样本数。

**X\_train:** *torch.Tensor or np.ndarray*

具有[n\_samples，n\_channels，n\_samples]形状的训练数据。 **Y\_train:** *torch.Tensor or np.ndarray*

使用形状[n\_samples]训练标签。 **kfolds:** *int*

K折交叉验证的折数。 **device:** *torch.device*

用于执行计算的设备（CPU或GPU）。 **model\_name:** *str*

用于保存目的的模型名称（例如，“TCNet\_Fusion”）。 **subject:** *str or int*

保存模型文件的主题标识符。 **model\_savePath:** *str*

保存训练模型文件的目录路径。

**Returns**

None。

将模型训练到位，并将每个折叠的模型状态保存到model\_savePath。打印训练进度和各折叠的平均评估精度。

**4.14.3 Functions: test\_with\_cross\_validate**

使用K倍交叉验证模型在测试数据集上评估模型。此函数从K倍交叉确认中加载预训练的模型权重，在测试数据集中评估模型，并计算性能指标，包括准确性、精确度、召回率、F1分和Cohen的Kappa分数。它跨倍数聚合结果以计算平均准确性和Kappa分数，适用于脑机接口（BCI）任务，如EEG信号分类。

**test\_with\_cross\_validate***(model, device, X\_test, Y\_test, model\_path, kfolds, subject)*

使用K倍交叉验证模型在测试数据集上评估模型

**Parameters**

**model :** *torch.nn.Module*

要测试的PyTorch模型（例如TCNet\_Fusion或EEGNet）。

**device:** *torch.device*

用于执行计算的设备（CPU或GPU）。

**X\_test:** *torch.Tensor or np.ndarray*

具有[n\_samples，n\_channels，n\_samples]形状的训练数据。 **Y\_test:** *array-like*

使用形状[n\_samples]训练标签。

**model\_path:** *str*

保存训练模型文件的目录路径。 **kfolds:** *int*

K折交叉验证的折数。 **subject:** *str or int*

保存模型文件的主题标识符。

**Returns**

**avg\_acc:** *float*

所有folds的平均分类准确率（百分比）。 **avg\_Kscore:** *float*

所有*folds*的Kappa平均得分。

**4.14.4 Functions: validate\_model**

在验证数据集上评估模型。此函数使用指定的损失函数计算所提供数据集上模型的平均损失和准确性。该模型设置为评估模式，在推理过程中禁用梯度以节省内存并提高性能。

**validate\_model***(model, dataset, device, losser, batch\_size=128)*

在验证数据集上评估模型

**Parameters**

**model :** *torch.nn.Module*

要验证的PyTorch模型（例如TCNet\_Fusion或EEGNet）。

**device:** *torch.device*

用于执行计算的设备（CPU或GPU）。

**dataset:** *torch.utils.data.Dataset*

用于评估模型的验证数据集。 **losser:** *callable*

计算模型损失的损失函数（例如torch.nn.CrossEntropyLoss）。

**batch\_size:** *int, optional*

DataLoader的每批样本数（默认值：128）。

**Returns**

**loss\_val:** *float*

验证数据集的平均损失。 **accuracy\_val:** *float*

验证数据集的平均准确度，计算为正确分类样本的分数。