Supplement to DMSACNN: Deep Multiscale Attentional Convolutional Neural Network for EEG-Based Motor Decoding

Ke Liu, Xin Xing, Tao Yang, Zhuliang Yu, *Member, IEEE*, Bin Xiao, Guoyin Wang*, Wei Wu*, *Senior Member, IEEE*

I. Experiments

A two-stage training strategy was adopted to prevent overfitting and reduce required epochs. The training procedure is detailed in Algorithm 1, while the trainable parameters for all deep learning algorithms are shown in Table I. Specifically, we implemented Deep ConvNet [1], EEGNet [2], and FBCNet [3] using code from `https://github.com/ravikiran-mane/FBCNet`, EEG-Inception [4] from `https://github.com/esantamariavazquez/EEG-Inception`, and EEG Conformer [5] from `https://github.com/eeyhsong/EEG-Conformer`. However, the three public datasets we use have a sampling frequency of 250Hz, whereas EEG-Inception and EEGNet are designed for data with a sampling frequency of 128Hz. Therefore, we adjust the length of time convolution kernels and pooling layers in both architectures by a factor of 2 to approximate the 250Hz sampling rate, consistent with previous work [3]. As shown in Table II, the specific parameter settings of the EEGNet and EEG-Inception models used in the experimental analysis are presented, with all parameter names referenced from the original paper.

TABLE I

Number of learnable parameters for different algorithms across three datasets.

Dataset	Deep ConvNet	EEGNet	EEG-Inception	FBCNet	EEG Conformer	DMSACNN
BCI-IV-2a	282879	4028	29816	11812	789572	35884
HGD	296629	4380	30872	18148	824772	38524
OpenBMI	278827	3002	29706	8930	786306	20762

TABLE II

Parameter settings for the EEGNet and EEG-Inception models.

Model	Block	Layer	Filters	Size	Activation	Options
EEGNet	Block 1	Conv 2D	F1=8	(1,125)	Linear	mode=same
		BatchNorm				
		DepthwiseConv2D	(D=2)*F1	(C,1)	Linear	mode=valid,depth=D,max norm=1
		BatchNorm				
		Activation			ELU	
		AveragePool2D		(1,4)		
		Dropout				p=0.25
	Block 2	SeparableConv2D	F2=16	(1,22)	Linear	mode=same
		BatchNorm				
		Activation			ELU	
		AveragePool2D		(1,8)		
		Dropout				p=0.25
EEG-Inception	Inception Module 1	Conv2D	C1=C2=C3=8	(128, 1), (64, 1), (32, 1)	ELU	mode=same
		DepthwiseConv2D	D1=D2=D3=2	(1,C)	ELU	mode=valid,depth=2,max norm=1
		AveragePooling2D		(8,1)		
	Inception Module 2	Conv2D	C4=C5=C6=8	(32, 1), (16, 1), (8, 1)	ELU	mode=same
		AveragePooling2D		(4, 1)		
	Output Module	Conv2D	C7=12	(16, 1)	ELU	mode=same
		AveragePooling2D		(4, 1)		
		Conv2D	C8=6	(8,1)	ELU	mode=same
		AveragePooling2D		(4,1)		

Algorithm 1: Two-stage Training Strategy

Input: N trials of training data $X_{\text{train}} \in \mathbb{R}^{N \times C \times T}$, true labels $Y_{\text{train}} \in \mathbb{R}^{N \times 1}$, initialized parameters of DMSACNN Θ and the maximum number of training epochs.

Output: The parameters of DMSACNN Θ.

Stage 1:
X_{sub}, X_{val} = func_split(X_{train});
Y_{sub}, Y_{val} = func_split(Y_{train});
while $epoch < max\_epochs\_1st$ **do**
    Train Network $(X_{sub}, Y_{sub}, \Theta)$;
    $pred_{sub}, loss_{sub}$ = func_predict(X_{sub}, Y_{sub}, Θ);
    $pred_{val}, loss_{val}$ = func_predict(X_{sub}, Y_{sub}, Θ);
    if $loss_{val} < min\_loss$ **then**
        $min\_loss = loss_{val}$;
        best_model = Θ;
    end
    $doStop$ = func_stopCheck($loss_{val}, min\_loss$);
    if $doStop$ **then**
        $\epsilon = loss_{sub}$;
        break;
    end
    $epoch = epoch + 1$;
end
Stage 2:
while $epoch < max\_epochs\_2st$ **do**
    Train Network $(X_{train}, Y_{train}, \Theta)$;
    $pred_{val}, loss_{val}$ = func_predict(X_{val}, Y_{val}, Θ);
    if $loss_{val} < min\_loss$ **then**
        $min\_loss = loss_{val}$;
        best_model = Θ;
    end
    $doStop$ = func_stopCheck($loss_{val}, \epsilon$);
    if $doStop$ **then**
        Θ_{best} = best_model;
        break;
    end
    $epoch = epoch + 1$;
end

References

[1] R. T. Schirrmeister, J. T. Springenberg, L. D. J. Fiederer, M. Glasstetter, K. Eggensperger, M. Tangermann, F. Hutter, W. Burgard, and T. Ball, "Deep learning with convolutional neural networks for EEG decoding and visualization," *Human Brain Mapping*, vol. 38, no. 11, pp. 5391–5420, 2017.

[2] V. J. Lawhern, A. J. Solon, N. R. Waytowich, S. M. Gordon, C. P. Hung, and B. J. Lance, "EEGNet: a compact convolutional neural network for EEG-based brain–computer interfaces," *Journal of Neural Engineering*, vol. 15, no. 5, p. 056013, 2018.

[3] R. Mane, E. Chew, K. Chua, K. K. Ang, N. Robinson, A. P. Vinod, S.-W. Lee, and C. Guan, "FBCNet: A multi-view convolutional neural network for brain-computer interface," *arXiv preprint arXiv:2104.01233*, 2021.

[4] E. Santamaria-Vazquez, V. Martinez-Cagigal, F. Vaquerizo-Villar, and R. Hornero, "EEG-inception: a novel deep convolutional neural network for assistive ERP-based brain-computer interfaces," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 28, no. 12, pp. 2773–2782, 2020.

[5] Y. Song, Q. Zheng, B. Liu, and X. Gao, "EEG conformer: Convolutional transformer for EEG decoding and visualization," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 31, pp. 710–719, 2022.