Temporally Adaptive Common Spatial Patterns with Deep Convolutional Neural Networks

Mahta Mousavi[1] and Virginia R. de Sa[2]

Abstract— **Brain-computer interface (BCI) systems are proposed as a means of communication for locked-in patients. One common BCI paradigm is motor imagery in which the user controls a BCI by imagining movements of different body parts. It is known that imagining different body parts results in event-related desynchronization (ERD) in various frequency bands. Existing methods such as common spatial patterns (CSP) and its refinement filterbank common spatial patterns (FB-CSP) aim at finding features that are informative for classification of the motor imagery class. Our proposed method is a temporally adaptive common spatial patterns implementation of the commonly used filter-bank common spatial patterns method using convolutional neural networks; hence it is called TA-CSPNN. With this method we aim to: (1) make the feature extraction and classification end-to-end, (2) base it on the way CSP/FBCSP extracts relevant features, and finally, (3) reduce the number of trainable parameters compared to existing deep learning methods to improve generalizability in noisy data such as EEG. More importantly, we show that this reduction in parameters does not affect performance and in fact the trained network generalizes better for data from some participants. We show our results on two datasets, one publicly available from BCI Competition IV, dataset 2a and another in-house motor imagery dataset.**

I. Introduction

Brain-computer interface (BCI) systems read and interpret brain signals directly from the brain and were proposed originally as communication methods for patients suffering from locked-in syndrome [1]. Motor-imagery (MI) based BCIs involve the user imagining moving different body parts to provide different control signals. This results in decreased/increased power in some frequency bands often referred to as event-related desynchronization/synchronization (ERD/S) [2], [3], and the differences between MI of different body parts can be emphasized with spatial filtering. The common spatial patterns (CSP) algorithm is commonly used to find filters that maximize the projected variance for one class while minimizing it for the other class [4]. Filter-bank common spatial patterns (FB-CSP) is a variation of CSP in which optimal sets of filters are sought in multiple frequency bands [5]. However, one downside of both CSP and FB-CSP methods is that supervised feature extraction step (finding the optimal filters) and classification are performed in separate steps.

*This work was supported by NSF IIS 1219200, IIS 1817226, SMA 1041755, and IIS 1528214, FISP G2171, G3155, NIH 5T32MH020002-18 and Mary Anne Fox dissertation year fellowship.

[1]Mahta Mousavi is with the Electrical and Computer Engineering department, UC San Diego, La Jolla, CA 92037, USA. mahta@ucsd.edu

[2]Virginia R. de Sa is with the Department of Cognitive Science and the Halıcıoğlu Data Science Institute, UC San Diego, La Jolla, CA 92037, USA. desa@ucsd.edu

Convolutional neural networks (CNN) have revolutionized the area of computer vision [6]. However, the characteristics of EEG signals are very different from those of images, videos, or speech. EEG signals are time series recorded from electrodes located at multiple sites on the scalp. They are prone to artifacts from non-brain sources such as eye and muscle movements and usually have low signal-to-noise ratio. Therefore, the common architectures of deep convolutional neural networks should be adapted to provide their benefits (end-to-end feature extraction and classification and ability to learn non-linear task specific classification boundaries) while not suffering from the potential drawbacks (overfitting or learning to use class-correlated non-brain artifacts for classification).

Proposed by Maryanovsky et. al [7], CSP-NN implements the CSP algorithm in a convolutional layer (with kernel size $C \times 1$) followed by squared-average pooling to emulate the usual squared post-processing after CSP filtering. It independently computes spatial filters for multiple frequency bands using signals that are already filtered through a bank of band-pass filters just as the original filter-bank CSP algorithm does. Then the features from the filters are merged and go through a fully-connected layer before the output layer.

Shallow and deep ConvNets [8] are two CNN-based architectures proposed for EEG-based BCI data classification. Shallow ConvNet learns a temporal convolution as its first layer and a spatial convolution afterwards. A squared-average pooling layer provides non-linearity; however, the pool that is averaged and squared is shorter than the length of the EEG epoch. Next, the features are concatenated and their logarithm is sent to a dense layer with soft max activation over the number of units equal to the number of motor imagery classes.

EEGNet [9] was proposed as a general-purpose CNN-based model for EEG-based BCIs and can handle both ERD/ERS type features as well as temporal event-related potentials (ERP). It consists of two blocks, each with convolutional layers, non-linear activation function (exponential linear unit), pooling and dropout. The EEGNet architecture has fewer parameters than shallow ConvNet.

As motor imagery BCIs are known to involve changes in power in different frequency bands, we propose an architecture for ERD/ERS type features. It is important to note that since EEG is generally noisy and artificial neural networks can use many parameters to learn highly non-linear functions, they are prone to overfitting. Inspired by the FB-CSP method, our goal is to propose a CNN-based model for motor imagery classification that keeps the number of parameters small

without compromising performance. We propose to use the temporal convolutional layer from EEGNet together with the spatial feature extraction convolutional layer and activation function from CSP-NN. We call this temporally adaptive common spatial patterns with neural networks (TA-CSPNN). This method uses about half the parameters of EEGNet and yet provides similar or improved results (see Section IV).

II. Proposed Architecture: TA-CSPNN

Let $X_i \in R^{C \times T}$ be the available EEG epochs for each class $i \in \{1, ..., N\}$ where N is the number of imagery classes, C is the number of EEG channels and T the number of time samples. Layers in TA-CSPNN are described in table I and in a block diagram in figure 1. The design parameters are K: length of temporal kernel, F_t: number of temporal filters, F_s: number of spatial filters, and p: dropout layer parameter that indicates the fraction of layer inputs to drop [10].

TABLE I
Description of layers in TA-CSPNN.

Layer	Filters/Units	Size	Output
Input	-	-	$(1, C, T)$
Conv 2D	F_t	$(1, K)$	(F_t, C, T)
Batch Normalization	-	-	(F_t, C, T)
Depthwise Conv 2D	F_s	$(C, 1)$	$(F_t \times F_s, 1, T)$
Batch Normalization	-	-	$(F_t \times F_s, 1, T)$
Activation: x^2	-	-	$(F_t \times F_s, 1, T)$
Average pooling	-	-	$(F_t \times F_s, 1, 1)$
Dropout(p)	-	-	$(F_t \times F_s, 1, 1)$
Flatten	-	-	$F_t \times F_s$
Fully connected	N	-	N
Activation: Softmax	-	-	N

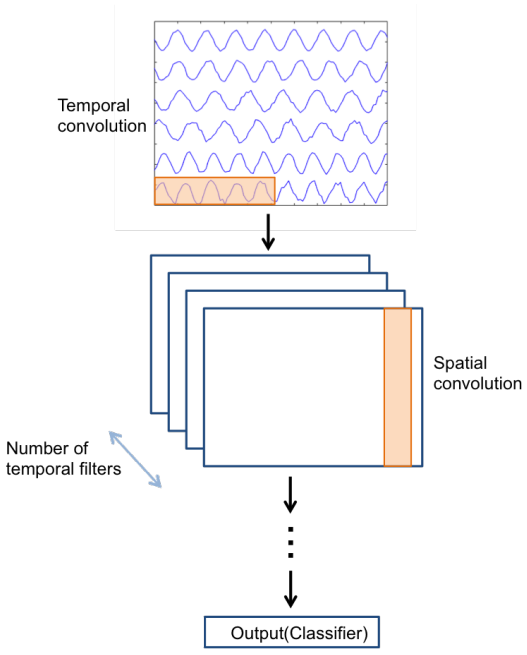

Fig. 1. Block diagram of the proposed TA-CSPNN.

The first convolutional layer filters the input EEG signal with multiple filters. Note that this layer is equivalent to filtering each channel of EEG data with a finite impulse response (FIR) filter:

$$X_i(c,t) = \sum_{n=0}^{K-1} b_n X_i(c, t-n), \quad (1)$$

where K is the length of the temporal filter and b_n $(n \in \{0, ..., K-1\})$ the filter weights. We did not use a bias for the weights trained in this layer to be as close to equation 1 as possible. Note that this temporal Conv2D is implemented with mode 'same' in Keras to pad the input with zeros such that the output has the same number of rows and columns as the input. Batch normalization [11] along the first dimension is applied next. Then a depthwise 2D convolution is applied to extract spatial features. This is equivalent to learning the spatial filters in each temporally filtered (filterbank) separately. We used a norm constraint on the spatial filter weights such that $||w||_2 \leq 1$. This is because the common spatial filters in the CSP algorithm are eigenvectors of a generalized eigenvalue problem and have norm 1. Note that this Conv2D layer is only applied along the channels (and not the time dimension) and is essentially a one-dimensional kernel. Common spatio-temporal filters can be learnt by using a 2D kernel along time and space as discussed in [7].

The output of the spatial convolution is squared and summed (average pooling layer) across time which is equivalent to calculating the power of the spatially filtered EEG data. The dropout layer [10] comes after. We use dropout to prevent overfitting and allow better generalization. Finally the features are sent through a dense layer of the size of the number of imagery classes (output). The activation is chosen to be softmax such that the network outputs probabilities.

As mentioned earlier, activity from the spatial convolution layer is squared and passed through an average pooling layer. This is different from EEGNet's use of the exponential linear unit (ELU) activation function. Figure 2 plots both of these functions. We chose x^2 as the non-linear activation function since ERS/ERD features are variations in the power of the EEG signal.

Code was implemented in Keras [12] with Tensorflow backend [13] and is available at `https://github.com/mahtamsv/TA-CSPNN`. We used Adam optimization with default parameters [14] to minimize the categorical cross-entropy loss function. In all experiments, 10% of the training data were used for validation. To avoid overfitting, the accuracy of the validation set was monitored for early stopping [15] with a patience of 50 and maximum of 500 epochs. The model was then evaluated on the test set.

III. Datasets

We report the performance of our proposed architecture on two datasets:

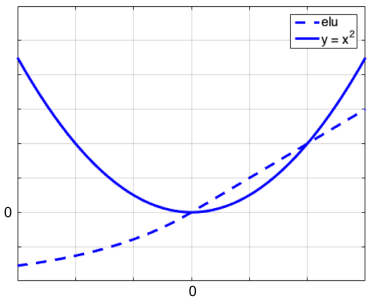

Fig. 2. Comparison of ELU and x^2 activations.

A. Dataset I

This is a publicly available dataset (BCI Competition IV, Dataset 2a) [16] from 9 participants performing left hand, right hand, both feet, and tongue motor imagery. The dataset consists of data from two sessions (train and evaluation sets) for each participant recorded on two different days. Each session has 288 trials total (72 for each imagery class). Data were recorded from 22 EEG and 2 EOG channels but we only used the EEG channels.

We used code provided by [8] to import, epoch and filter the data from 4-40 Hz and epoch from 0.5-2.5 seconds after the onset of the cue. The number of classes is $N = 4$. Data on each channel were downsampled to 125 Hz.

B. Dataset II

Motor imagery data were recorded from 10 participants who signed a consent form approved by the UC San Diego Human Research Protections Program prior to participating in the experiment. Data were recorded using a 64-channel BrainAmp system (Brain Products GmbH). Participants were instructed to perform right/left hand motor imagery to move a cursor on the screen in front of them right/left towards a target at either the right/left side of the screen. The cursor moved at the speed of one step per second towards/away from the target. After visual inspection, we applied ICA and removed eye and muscle artifacts. Then the data were downsampled to 100 Hz, bandpassed from 1-40 Hz with an FIR filter of order 100 and epoched 0.1-1 second after each cursor movement. For more details about the experiment please refer to [17].

For the CSP-NN method, we used FIR filters of order 100 to filter the EEG data on each channel in 1-3, 2-5, 4-7, 6-10, 7-12, 10-15, 12-19, 18-25, 19-30, 25-35, and 30-40 Hz frequency bands to cover low and high theta, mu and beta bands while compensating for individual differences [17].

We randomly selected 5 divisions of train-validation-test sets in which right and left imagery classes were balanced in each set. Performance is reported as the average of the classification accuracy on the test sets.

IV. Results and Discussion

As the EEGNet architecture for ERD/ERS data outperforms deep ConvNet [9], we restrict comparison of our TA-CSPNN to shallow ConvNet and EEGNet. For dataset I, performance of TA-CSPNN with $F_t = 8$ temporal filters and $F_s = 2$ spatial filters is presented in table II and is compared with EEGNet(8,2) [9] and shallow ConvNet (Sh-ConvNet) [8]. For this dataset, the length of the temporal kernel and the dropout parameter were set to $K = 63$ and $p = 0.25$ respectively for both EEGNet and TA-CSPNN. Shallow ConvNet was originally proposed for a sampling rate of 250 Hz. Since we downsampled the data by two, we also divided the lengths of the temporal kernels and pooling layers by two: temporal kernel length was set to 13, pool size in the average pooling layer to (1,35) with a stride of size (1,7) as suggested by [9].

For each participant, all models were trained on the train set with 10% for validation and tested on the evaluation set (as originally distributed for the purpose of the BCI competition). Performance is reported as the average of the classification accuracy on the evaluation set for a model trained with 10 different initializations. Shallow ConvNet performs significantly better than TA-CSPNN for A2 and A4 but does significantly worse for A1, A3, A8 and A9 (paired-sample t-test, $p < 0.05$). However, the difference between EEGNet and TA-CSPNN is not statistically significant (paired-sample t-test, $p > 0.1$).

Table III reports the length of EEG epochs (L), the sampling rate (SRate) and the number of trainable parameters in each architecture for dataset I. Note that the number of parameters in TA-CSPNN is about half the number of parameters in EEGNet and less than 2.5% of the number of parameters in shallow ConvNet.

TABLE II
CLASSIFICATION RATES FOR DATASET I. NOTE THAT THIS IS A FOUR-CLASS CLASSIFICATION.

PID	Sh-ConvNet	EEGNet(8,2)	TA-CSPNN(8,2)
A1	0.61	0.69	0.71
A2	0.39	0.40	0.36
A3	0.70	0.79	0.79
A4	0.55	0.49	0.44
A5	0.38	0.38	0.39
A6	0.42	0.46	0.44
A7	0.70	0.71	0.72
A8	0.62	0.73	0.72
A9	0.68	0.78	0.76

TABLE III
NUMBER OF PARAMETERS FOR DATASET I: TRIAL LENGTH IS 2 SECONDS AT SAMPLING RATE OF 125 HZ.

L	SRate	Sh-ConvNet	EEGNet(8,2)	TA-CSPNN(8,2)
2 s	125 Hz	40644	1900	972

For dataset II, the length of the temporal kernel and dropout parameter were set to $K = 50$ and $p = 0.25$ for both EEGNet and TA-CSPNN. Since CSP-NN has 11 temporal filters and 6 spatial filters (equivalent to the top three filters for each class in CSP), we set the number of temporal and

TABLE IV

CLASSIFICATION RATES FOR DATASET II.

PID	CSP-NN	EEGNet(11,6)	TA-CSPNN(11,6)
P1	0.87	0.87	0.87
P2	0.66	0.68	0.74
P3	0.71	0.70	0.71
P4	0.74	0.78	0.76
P5	0.83	0.91	0.89
P6	0.74	0.68	0.74
P7	0.87	0.82	0.86
P8	0.82	0.77	0.88
P9	0.69	0.72	0.75
P10	0.71	0.67	0.78

TABLE V

NUMBER OF PARAMETERS FOR DATASET II: TRIAL LENGTH IS 0.9 SECONDS AT SAMPLING RATE OF 100 HZ.

L	SRate	CSP-NN	EEGNet(11,6)	TA-CSPNN(11,6)
0.9 s	100 Hz	6560	10738	5062

spatial filters in both EEGNet and TA-CSPNN to $F_t = 11$ and $F_s = 6$ respectively.

CSP-NN was implemented slightly different than [7]: we used batch normalization [11] after the Conv2D layer for each frequency band and also used dropout with $p = 0.25$ after the squared-average pooling layer. Then the merged features from different frequency bands were passed to a dense layer with 30 hidden units and ELU activation function before the output layer.

Table IV reports the classification accuracy for each participant in dataset II. TA-CSPNN does significantly better than EEGNet (paired-sample t-test, $p < 0.05$) in P2, P6, P8 and P10. For the rest of the participants, the difference between EEGNet and TA-CSPNN is not statistically significant.

Table V reports the length of EEG epochs (L) for dataset II as well as the sampling rate (SRate) and the number of trainable parameters for each model applied to dataset II. Note that TA-CSPNN uses less than half the number of parameters in EEGNet.

V. CONCLUSIONS AND FUTURE WORK

In this work, we proposed a temporally adaptive convolutional neural network-based implementation of the widely used FB-CSP to classify ERD/ERS: TA-CSPNN. Our model uses about half the number of parameters in EEGNet [9] and less than 2.5% of that used by shallow ConvNet [8] and shows comparable or improved results for motor imagery classification on a publicly available dataset (BCI Competition IV, dataset 2a) and another motor imagery dataset [17].

Our proposed architecture is easily generalized to incorporate spatio-temporal features in each filterbank by changing the spatial convolution kernel size [7], [18]. Also, regularization methods that have been proposed to improve CSP [19] can be incorporated as regularization terms on the weights in the spatial convolution layer. Moreover, the hidden layers may provide additional useful feedback signals for online motor imagery training [20].

REFERENCES

[1] L. A. Farwell and E. Donchin, "Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials," *Electroencephalography and clinical Neurophysiology*, vol. 70, no. 6, pp. 510–523, 1988.

[2] D. J. McFarland, L. A. Miner, T. M. Vaughan, and J. R. Wolpaw, "Mu and beta rhythm topographies during motor imagery and actual movements," *Brain topography*, vol. 12, no. 3, pp. 177–186, 2000.

[3] G. Pfurtscheller and C. Neuper, "Motor imagery and direct brain-computer communication," *Proceedings of the IEEE*, vol. 89, no. 7, pp. 1123–1134, 2001.

[4] H. Ramoser, J. Muller-Gerking, and G. Pfurtscheller, "Optimal spatial filtering of single trial EEG during imagined hand movement," *IEEE transactions on rehabilitation engineering*, vol. 8, no. 4, pp. 441–446, 2000.

[5] K. K. Ang, Z. Y. Chin, H. Zhang, and C. Guan, "Filter bank common spatial pattern (FBCSP) in brain-computer interface," in *Neural Networks, 2008. IJCNN 2008.(IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on*. IEEE, 2008, pp. 2390–2397.

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[7] D. Maryanovsky, M. Mousavi, N. G. Moreno, and V. R. de Sa, "CSP-NN: A convolutional neural network implementation of common spatial patterns," in *Proceedings of the 7th Graz Brain-Computer Interface Conference 2017*, 2017, pp. 302–307.

[8] 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.

[9] V. Lawhern, A. Solon, N. Waytowich, S. M. Gordon, C. Hung, and B. J. Lance, "EEGNet: a compact convolutional neural network for EEG-based brain–computer interfaces," *Journal of neural engineering*, 2018.

[10] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[11] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[12] F. Chollet *et al.*, "Keras," https://keras.io, 2015.

[13] M. Abadi *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/

[14] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[15] R. Caruana, S. Lawrence, and C. L. Giles, "Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping," in *Advances in neural information processing systems*, 2001, pp. 402–408.

[16] C. Brunner, R. Leeb, G. Müller-Putz, A. Schlögl, and G. Pfurtscheller, "BCI competition 2008–Graz data set A," *Institute for Knowledge Discovery (Laboratory of Brain-Computer Interfaces), Graz University of Technology*, vol. 16, 2008.

[17] M. Mousavi, A. S. Koerner, Q. Zhang, E. Noh, and V. R. de Sa, "Improving motor imagery BCI with user response to feedback," *Brain-Computer Interfaces*, vol. 4, no. 1-2, pp. 74–86, 2017.

[18] D. Krusienski, E. Sellers, and T. Vaughan, "Common spatio-temporal patterns for the P300 speller," in *Neural Engineering, 2007. CNE'07. 3rd International IEEE/EMBS Conference on*. IEEE, 2007, pp. 421–424.

[19] F. Lotte and C. Guan, "Regularizing common spatial patterns to improve BCI designs: unified theory and new algorithms," *IEEE Transactions on biomedical Engineering*, vol. 58, no. 2, pp. 355–362, 2011.

[20] M. Mousavi and V. R. de Sa, "Towards elaborated feedback for training motor imagery brain computer interfaces," in *Proceedings of the 7th Graz Brain-Computer Interface Conference 2017*, 2017, pp. 332–337.