Proceedings of the
7th Graz Brain-Computer Interface Conference 2017 DOI: 10.3217/978-3-85125-533-1-54

DEEP RECURRENT CONVOLUTIONAL NEURAL NETWORKS FOR CLASSIFYING P300 BCI SIGNALS

R. K. Maddula[1], J. Stivers[2], M. Mousavi[3], S. Ravindran[1], V. R. de Sa[2]

[1]Computer Science and Engineering, UC San Diego, La Jolla, CA, United States
[2]Cognitive Science, UC San Diego, La Jolla, CA, United States
[3]Electrical and Computer Engineering, UC San Diego, La Jolla, CA, United States

E-mail: desa@ucsd.edu

ABSTRACT: We develop and test three deep-learning recurrent convolutional architectures for learning to recognize single trial EEG event related potentials for P300 brain-computer interfaces (BCI)s. One advantage of the neural network solution is that it provides a natural way to share a lower-level feature space between subjects while adapting the classifier that works on that feature space. We compare the deep neural networks with the standard methods for P300 BCI classification.

INTRODUCTION

Brain-computer interfaces (BCIs) are being developed as communication methods for people with locked-in syndrome who have lost the ability to control their muscles and thus can't move, speak, or eventually even move their eyes in a well-controlled way. Electroencephalography (EEG) provides a cheap non-invasive monitoring channel with excellent temporal precision and is the most used signal for brain-computer interfaces. Classifying the EEG signals however is a difficult task as the signals are subject to poor spatial resolution, sensitivity to other electromagnetic sources as well as to the impedance of the electrode-scalp interface. The accuracy of classification algorithms for interpreting EEG data is a major barrier to the improvement of the EEG based BCI systems.

P300 based BCIs attempt to recognize a single-trial P300-like response that occurs when a subject attends to a presented rare or meaningful item. P300 responses have been studied in Cognitive Neuroscience for years and are referred to as event related potentials as they are time-locked to the presentation of the stimulus (event) [1]. Modern P300 BCIs classify the single-trial version of the event-related potential. Single-trial processing of EEG data has to date largely been most successful with very simple algorithms due to the large amounts of noise in the data and the paucity of data.

Many studies [2,3] have generally found that linear classification methods such as linear support vector machines (SVM) and linear discriminant analysis (LDA) worked better for classifying P300 BCIs than simple shallow non-linear methods such as multilayer perceptron and Gaussian kernel support vector machines. However, these classifiers treat every spatio-temporal sample equally (for example they perform identically if channel data are consistently permuted), and therefore do not benefit from the potential assumption that local temporal and spatial patterns exist in the data. By preserving the spatial organization of electrodes in the representation of the data, we could attempt to learn local spatial filters present in the data. Overall, a classifier should consider both spatial and temporal information while classifying the P300 EEG signal.

Recently deep neural networks have transformed the fields of handwriting recognition, speech recognition [4], large scale image recognition [5] and video analysis [6,7], and are rapidly transforming machine learning more generally. More recently convolutional neural nets and recurrent nets have been used in the realm of EEG signal classification. Cecotti et al.[8] used convolutional neural networks for P300 EEG classification. Mirowski et al.[9] used convolutional networks to predict epileptic seizures before they happen. Bashivan et al.[10] used a deep recurrent-convolutional network to learn representations from EEG, and demonstrated its advantages in the context of a mental load classification task. They successfully preserved the spectral, spatial, and temporal structure of the data during classification. There are multiple reasons to believe that deep learning could transform EEG processing: a) convolutional neural networks provide an intuitive and well understood way to deal with natural spatial relationships [5], b) neural networks easily allow filtering and classification to be combined in one discriminative framework, and c) recent advances in recurrent neural network (RNN) structures such as Long Short Term Memory (LSTM) [11] provide an intuitive and well understood way to deal with natural temporal relationships.

Our goal is to develop various deep learning architectures for classifying the P300 EEG signals. The proposed classifiers respect the spatial and temporal nature of the EEG signals, optimally combine their information, and naturally permit the sharing of sub-structure between tasks and between subjects. We propose a three-dimensional convolutional neural network (3D-CNN) [6,12] in conjunction with a two-dimensional convolutional neural network (2D-CNN) and LSTM to capture spatio-temporal patterns in the EEG signals We also explore the use of transfer learning, where the information can be shared between different subjects [13].

Transfer learning is important in EEG analysis as due to cortical folding and other differences between people, EEG classifiers trained on one subject do not generalize as well to other subjects. However, it is time-consuming to collect training data from each new subject, so a desirable strategy is to train a "proto-classifier" with many previous subjects and then refine it with a small amount of training data from the new subject.

Our proposed method for P300 EEG signal classification is closely related to the one that is proposed by Bashivan et al.[10]. This method preserved the spatial structure of the data by transforming EEG into 2D image frames, and combined 2D-CNN and LSTM for the classification. In contrast, we propose the use of a 3D CNN in order to preserve spatio-temporal features, and also employ transfer learning to further increase classification performance.

MATERIALS AND METHODS

EEG DATASET: Data were collected from a P300 segment speller paradigm where letter segments were flashed and subjects had to mentally note which stimuli were segments from their target letter[19]. That is, targets were colored segments that form part of the desired letter and non-targets were differently colored segments that are not part of the letter. There were 10 segments total and each segment has a unique color. Subjects were cued with the colored segments at the beginning of each trial. Responses to target segments give similar P300 responses to target letters in a more common P300 BCI paradigm.

We performed experiments using two training datasets. EEG dataset 1 was recorded from 4 subjects using a 64-channel active electrode EEG system (BioSemi Active II) with a sampling rate of 512 Hz and bilaterally referenced to the average of the two mastoids. Later, the data were segmented and temporally downsampled to 128 Hz. EEG Dataset 2 was recorded from 5 subjects using the BrainVision BrainAmp 64-channel EEG system. The native sampling rate was 5 kHz, downsampled to 100 Hz for classification. Data were re-referenced to common mean (montage average reference). On both the datasets, the EEG signal is bandpass filtered with an FIR filter of length 68-taps, with passband between 2 and 35 Hz and stopband cutoffs at 0.1 and 40 Hz. Also, the signal from 150 ms to 800 ms after segment flash onset was segmented and downsampled to ten frames.

The classification task is to classify an EEG signal signal into a target class or a non target class. The target class refers to the signals collected when a target segment is flashed, while the non target class refers to the signals collected when a non target segment is flashed. The dimension of each input signal is 64 x 10 (Channels in 3D space x Time Points).

DATA PRE-PROCESSING: Our goal is to improve the detection/classification of P300 responses by learning representations from the EEG data. We preserve spatial information, by projecting the EEG data into a 2D grid as in [10]. The EEG electrodes are located in a three dimensional space over the scalp. Transforming the EEG measurements from 3D locations on the head into a 2D grid is accomplished through spatial interpolation. An azimuthal equidistant projection [14] is used to project the position of electrodes in a 2D surface. Subsequently, cubic spline interpolation [15] is applied on the resultant 2D mapping of the electrodes to obtain a 2D grid of size n x n, where n is the number of points in each row. A square with roughly 8 or 12 interpolated points on each side seems sufficient for capturing the spatial variation. For this experiment, we use 8 interpolated points on each row and column resulting in an 8 x 8 2D grid, analogous to a 2D image with pixel dimension of 8 x 8. 2D images are constructed for every time window for each trial, and are given as an input to the deep convolutional neural network. The input dimension for each signal is 10 x 8 x 8 x 1 (Time Samples x height x width x Depth).

PROPOSED MODEL ARCHITECTURES: In order to learn the inherent spatial and temporal features from an EEG signal, we use a model which combines a deep hierarchical feature extractor with the one that can learn to recognize and synthesize the temporal features.

General multilayer perceptrons have not been widely successful in EEG as the massive number of unconstrained interdependent parameters can lead to overfitting. The convolutional framework allows for successfully learning complex relationships in images without overfitting for at least two reasons: (1) Each filter is only applied to a few local inputs, and (2) each filter is learned based on multiple windows (replicated throughout the training pattern) in each labeled example. This effectively increases the amount of training data available for learning the parameters.

Moreover, human brain activity is a temporally dynamic process. Variations of the signals between time points may actually contain additional information about the underlying P300 response. Hence, Long Short Term Memory (LSTM) is adopted on top of the CNN, to learn temporal patterns as has been done for action recognition in videos [7].

Generally, convolutions are applied on 2D feature maps to compute spatial features, and later recurrent layers are used to compute temporal features. However, the 2D ConvNets do not take the temporal information into account while performing the spatial convolutions on each frame. Hence, we propose an approach based on 3D ConvNets, which initially perform spatio-temporal convolutions, to consider both spatial information in each frame and temporal information encoded in multiple contiguous frames, preserving the temporal information of the input signal.

Here are the proposed architectures to extract the spatial and temporal information from the EEG signal.

- $2D^3 - L$: Layers of 2D-CNN are stacked on top of each other, and used on each frame to extract the spatial information, while LSTM was used to extract the temporal information, on the sequence of frames.

- $3D-2D^3-L$: A 3D-CNN was used initially to extract spatiotemporal features, and then a 2D-CNN and LSTM are applied on top of the 3D CNN.

- $S(3D)-2D^3-L$: A transfer learning approach is used on $3D-2D^3-L$ architecture, where we pretain the network on a different dataset, freeze the 3D CNN layers, and train the rest of the network on the current dataset. $S(3D)$ in $S(3D)-2D^3-L$ denotes that we are sharing the weights of 3D-CNN across the subjects.

We implemented the architectures using the Keras library and Theano framework. As described in the previous section, the EEG electrode positions are projected and interpolated into an 8x8 2D square grid and a sequence of 8x8 images are extracted over the successive time windows.

$2D^3-L$ **Architecture:** Here, we combine a 2D-CNN and LSTM, as a result separately utilizing both the spatial and temporal information for the classification.

As described in the previous section, in order to account for the temporal activity, we extract 10 frames from each trial ie., EEG signal, by dividing each trial into 10 time windows, and averaging over each time window. The sequence of these images are given as input to the CNN. The input data to the CNN is of the following dimension:

Total Number of Trials x *10 frames per each trial* x *8 width* x *8 height* x *1 depth*

The outputs of the 2D-CNN are fed into a recurrent network, where we investigate the temporal activity in the EEG signals.

During training, the input to the CNN is a fixed-size 8x8 image. The image is passed through a stack of convolutional (conv.) layers, where we use filters with a very small receptive field: 3 x 3 (the smallest size to capture the notion of left/right, up/down, center). The convolution stride is fixed to 1 pixel with rectified linear (ReLu) activation functions. In order to preserve the spatial resolution of the image, spatial padding of 1 pixel is used in each convolutional layer. Multiple convolution layers are stacked together and followed by a Max-pooling layer. Max-pooling is performed over a 2 x 2 pixel window, with stride 2. Using dataset D1, Best results were obtained by stacking 3 convolutional layers, with kernels respectively 32, 48 and 64, together, followed by a Max-Pooling layer. The dimension of the input image for each time step by the end of the 3rd Conv Layer is 8 x 8 x 64. After applying pooling, the dimension reduces to 4 x 4 x 64.

Fig. 1 illustrates the optimal CNN configuration.

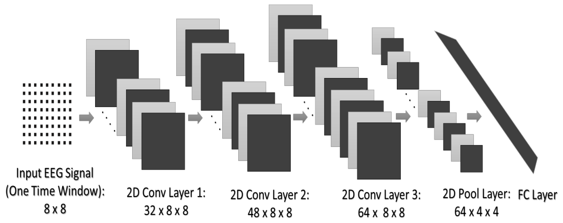

Figure 1: 2D Convolutional Neural Network for processing each temporal frame

This 2D-CNN architecture was adopted for each frame. The number of parameters in the network were reduced by sharing the parameters of the CNN over all the frames. A recurrent layer is applied on top of 2D-CNN. In order to use recurrent layers, the 3D output of 4 x 4 x 64 for each time step is converted into a 1D output of size 1024. The input dimension when training the recurrent neural network is 10 x 1024 (Time Steps x No. of Features).

RNN's provide an elegant way of dealing with sequential data that embodies correlations between data points that are close in the sequence. Though RNN's are successful in classifying many tasks such as speech recognition and text generation, they have difficulty with long term dependencies, due to the vanishing and exploding gradient problem [11], which results from propagating the gradients back through many layers. LSTMs are capable of handling the long term dependency problem; they learn when to forget previous hidden states and when to update the hidden states. We experimented (using subject D1) with various number of LSTM layers and memory cells in each layer. The best results were obtained when using a single LSTM layer with 32 memory cells.

For our implementation, we fed the extracted CNN outputs of 10 frames to the LSTM layer. The prediction of the LSTM layer at each time step was propagated up to the fully connected layer. The classification is done by averaging scores across all the frames. The outputs of the LSTM layer are fed into a fully connected layer, followed by a sigmoid layer.

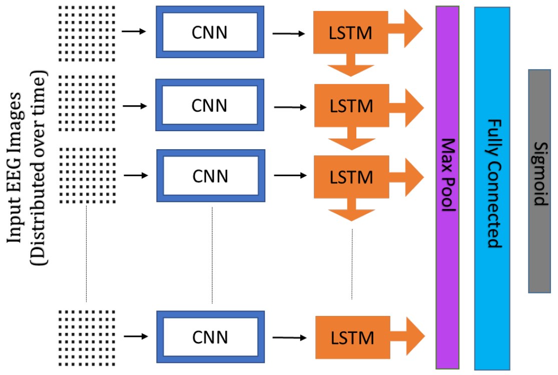

Figure 2: Overall architecture which combines 2D CNN with LSTM.

We also compared using temporal convolutions instead of LSTM, in order to evaluate the performance of LSTM in extracting temporal information from a sequence of EEG images ($2D^3 - 1D$). In this model, the 2D CNN outputs across time frames are fed into a 1D CNN with 32 kernels of size 3 and a stride of 1 frame. These kernels capture different temporal patterns across multiple time frames. Fig. 2 shows the proposed $2D^3 - L$ architecture.

$3D - 2D^3 - L$ Architecture:

The 3D ConvNets are well-suited for spatio-temporal feature learning. The 3D Convolution is achieved by convolving a 3D kernel to the cube formed by stacking multiple contiguous frames together[12]. Therefore the feature maps in the convolutional layer are connected to multiple contiguous frames in the previous layer, thereby capturing the temporal information.

Initially, the multidimensional input signal, with dimension 8 x 8 x 1 x 10 (Height X Width x Depth x Time Steps) is fed into a 3D-CNN. We tried various configurations of 3D-CNN, involving different number of kernels and kernel size. Our findings indicate that using one 3D Conv layer with 24 kernels of size 3 x 3 x 3 (kernel Depth x Kernel Height x Kernel Width) is the best option. The dimensions of the signal by the end of this layer would be 8 x 8 x 24 x 10 (Height x Width x Depth x Time Steps). We used the $2D^3-L$ architecture (Fig. 2) on top of the 3D CNN. The output of the 3D Conv layer in Figure Fig. 3 is fed into the $2D^3 - L$ network. The input for the $2D^3 - L$ network would be 10 frames of size 8 x 8 x 24. One layer of 3D-CNN with 24 kernels is followed by 3 layers of 2D-CNN with kernels 32, 48, 64 respectively for each time step.

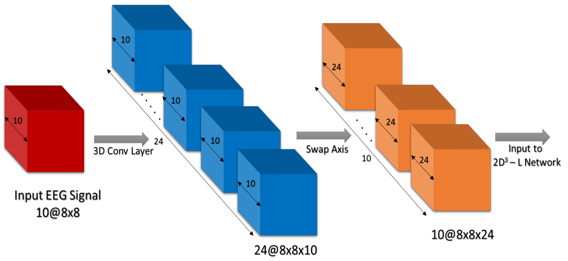

Figure 3: 3D convolutional neural network on EEG signal. Input signal (Red Cube) is of dimension 8 x 8 x 1 x 10 (Height x Weight x Depth x Time Points). The Red cube is basically formed by stacking up the 10 2D frames of size 8 x 8. A 3D -Conv layer of 24 filters with 3 x 3 x 3 size is applied (Blue cubes). The resultant output of size 8 x 8 x 10 x 24 is converted to 8 x 8 x 24 x 10 (Orange cubes). This would be the input to $2D^3 - L$ network, where 2D CNN is applied on each block of size 8 x 8 x 24, and the corresponding output of the blocks from 10 time steps is given to LSTM

Overall, the 3D-CNN helps us to extract the broad spatio-temporal features, and the 2D-CNN and LSTM find the hidden spatial and temporal features respectively.

$S(3D) - 2D^3 - L$ Architecture:

One advantage of the neural network approach is the ease in applying transfer learning where lower layer weights can be taken from a network trained on other subjects and then frozen for training of the whole network on the new subject [13].

The base network, which uses the $3D - 2D^3 - L$ architecture was initially trained on multiple subjects, and the corresponding parameters used in training the target network, which involves a fresh subject. Once the weights corresponding to the base network are copied to the target network, the 3D conv layer in the target network are frozen and do not change during the training. Only the weights corresponding to the higher layers (2D-CNN, LSTM, FC) of the target network change. We chose to freeze the CNN layers, instead of fine-tuning them, as the dataset is small and the number of parameters is large.

RESULTS

The network was trained and tested on two datasets. Dataset 1 comprises 4 subjects, while Dataset 2 comprises 5 subjects. The goal is to classify a signal as a P300 Target Signal or a Non-target signal. A split of 80% - 10% - 10% was used while dividing each fold into training, validation and testing respectively. Training is performed by optimizing the cross-entropy loss function. The network is trained using Adadelta, a variant of gradient descent which adapts over time using only first order information and has minimal computational overhead. In order to counteract possible overfitting due to the large number of weights, we used L2 Regularization penalty of 0.001 and Dropout of 20%. The architecture was the same for all subjects, where model tuning was done only on subject D1.

We compared the classification results of the proposed architectures with standards in the field - Stepwise LDA and shrinkage LDA. The shrinkage LDA algorithm uses automated shrinkage computation using the formula developed by Schaefer and Strimmer [18] based on the work of Ledoit and Wolf[17]. All tests were run with identical training/valid/test splits so that sensitive pairwise comparisons of accuracy could be made between the different algorithms. 50 different accuracy measurements were made for each network as follows: We made 5 random shuffles of the dataset and for each of these obtained 10 test sets by using an 80% train/10% validation/10% test division where each 10% of the data is used as the test set once.

Table 1: AUC measures for the proposed models on Dataset 1

	Sub. A1	Sub. B1	Sub. C1	Sub. D1
Stepwise LDA	0.67	0.65	0.69	0.65
Shrinkage LDA	0.69	0.64	0.71	0.65
$2D^3 - 1D$	0.65	0.65	0.67	0.64
$2D^3 - L$	**0.66**	**0.66**	**0.70**	**0.67**
$3D - 2D^3 - L$	**0.68**	**0.67**	**0.72**	**0.68**
$S(3D) - 2D^3 - L$	**0.69**	**0.70**	**0.72**	**0.69**

DOI: 10.3217/978-3-85125-533-1-54

Table 2: AUC measures for the proposed models on Dataset 2

	Sub. A2	Sub. B2	Sub. C2	Sub. D2	Sub. E2
Stepwise LDA	0.60	0.69	0.68	0.72	0.77
Shrinkage LDA	0.62	0.71	0.69	0.75	0.78
$2D^3 - 1D$	0.58	0.66	0.59	0.62	0.74
$2D^3 - L$	**0.61**	**0.68**	**0.62**	**0.70**	**0.76**
$3D - 2D^3 - L$	**0.64**	**0.70**	**0.67**	**0.75**	**0.76**
$S(3D) - 2D^3 - L$	**0.65**	**0.72**	**0.67**	**0.75**	**0.76**

Tab. 1 and Tab. 2 reports the Area under the ROC curve (AUC) for the baseline models and the proposed models on Dataset 1 (4 subjects) and Dataset 2 (5 subjects) respectively.

For the transfer learning in Dataset 1, we pretrain the 3D CNN network on all the subjects other than the current subject, and finally freeze the 3D convolutional layer and train the network on the current subject. For transfer learning in Dataset 2, due to the discovery of one subject (D2) with a very different "P300" signal (possibly more of an error-related potential signal), using all other subjects for transfer learning was not very successful, so one subject (E2) was used as transfer for the other subjects (A2-D2). For subject E2, all the other subjects (A2-D2) were used for transfer learning. To provide an alternate perspective, we replicated positive samples as done in [8] instead of subsampling. The first 70% was used for training, and the next 15% for validation and the last 15% for testing. The results comparing the performance of LDA with shrinkage and $S(3D) - 2D^3 - L$ are presented in Tables 3 and 4.

Table 3: Comparing AUC on Dataset 1

	Sub. A1	Sub. B1	Sub. C1	Sub. D1
Shrinkage LDA	0.67	0.58	0.7	0.59
$S(3D) - 2D^3 - L$	**0.68**	**0.69**	**0.7**	**0.7**

Table 4: Comparing AUC on Dataset 2

	Sub. A1	Sub. B1	Sub. C1	Sub. D1	Sub. E1
Shrinkage LDA	0.58	0.64	0.68	0.78	0.7
$S(3D) - 2D^3 - L$	**0.66**	**0.7**	**0.65**	**0.74**	**0.78**

DISCUSSION

We measured and compared the performance of the proposed and baseline models using a paired t-test.

On Dataset 1, Our $S(3D) - 2D^3 - L$ performed significantly better than the stepwise LDA on all the subjects, and performed significantly better (pairwise t-test) than Shrinkage LDA on subjects B1 ($p = 2.32 \times 10^{-7}$) and D1 ($p = 2.31 \times 10^{-4}$). The results from the $S(3D) - 2D^3 - L$ net did not differ significantly from the shrinkage LDA results on subjects A1 and C1.

On Dataset 2, $S(3D) - 2D^3 - L$ performed significantly better than $3D - 2D^3 - L$ on a few subjects and performed equally well on the other subjects. $S(3D) - 2D^3 - L$ performed better than Shrinkage ($p = 6.98 \times 10^{-4}$) and Stepwise LDA ($p = 5.9 \times 10^{-3}$) on subject A2. Also, it performed significantly better than stepwise LDA, (but not Shrinkage LDA) on subjects B2 ($p = 2.73 \times 10^{-5}$ vs Stepwise LDA) and D2($p = 6.27 \times 10^{-5}$ vs Stepwise LDA). However, the shrinkage LDA performed significantly better than the $S(3D) - 2D^3 - L$ on subjects C2 ($p = 8.74 \times 10^{-4}$) and E2 ($p = 0.0324$).

Overall, the proposed models work better than stepwise LDA and work better than the best baseline model (shrinkage LDA) on a few subjects and relatively lower on some other subjects.

The $2D^3 - L$ architecture performed numerically better than $2D^3 - 1D$ for all subjects and the difference reached statistical significance on 7 out of the 9 subjects (p- values between 3.42×10^{-11} and 0.0498). Thus it appears that LSTM did better in dealing with temporal patterns in the EEG signals compared to temporal convolution (1D-CNN) at least among the architectures we tried. The LSTM has richer temporal dynamics and can look at temporal patterns arbitrarily far back in time. 1D-CNN just looks for specific patterns in time of length up to the kernel length, while the LSTM understands and keeps track on the previous patterns and perform backpropagation through time [16].

From Tab. 1 and Tab. 2, we notice that the $3D - 2D^3 - L$ architecture performs better than the $2D^3 - L$ architecture. Moreover, the difference reaches statistical significance on 6 out of the 9 subjects (p-values between 1.17×10^{-6} and 0.0016). Even though, the performance gap is very small, it is a consistent difference. The 3D-CNN model effectively learns spatio-temporal patterns of the EEG signal.

Moreover, the transfer learning approach managed to perform better than the 3D-CNN, which states that 3D conv layer potentially learns spatio-temporal representations that are subject-independent, and the 2D-CNN and LSTM are able to deal with the intra-subject spatial and temporal patterns. The 3DConv layer captures the underlying spatial and temporal information, which are subject-independent.

Overall, the results suggest that deep learning may be used as an alternative compared to traditional machine learning techniques and that a 3D-CNN architecture is an effective model for learning the nature of P300 EEG signals.

CONCLUSION

In this work, a new approach for P300-EEG signal classification is demonstrated. As opposed to the traditional techniques, the proposed classifiers respect the inherent spatial and temporal nature of the EEG signals. This is accomplished by representing the multi-channel EEG time series as a sequence of 2D image frames. Inspired by the state-of-the-art video classification techniques, we train a deep convolutional and recurrent neural network on these sequence of 2D images. We proposed three different architectures. We discovered that using a 3D-CNN in conjunction with 2D-CNN and LSTM performed better than using 2D-CNN and LSTM. The 3D-CNN appears to most effectively model spatial and temporal information.

The best performance is obtained when using a transfer learning approach, where we pretrain a network compris-

ing 3D Conv, 2D Conv and LSTM layers on different subjects, freeze the 3D Conv Layers, and perform classification training and then testing on a fresh subject. The classification performance of the proposed models are compared with common best performing classifiers in this field - Stepwise LDA and shrinkage LDA. The proposed models perform relatively better than the base line models on a few subjects. However, there is plenty of scope for improvement.
One of the benefits of the neural network approach is that spatio-temporal generalizations arise naturally. Combinations of 3D and 2D kernels could be used to optimally extract all the spatio-temporal structure that distinguishes the signals in P300 BCIs (or other temporal ERP signals). As a future direction, we will work on further increasing the performance, as improving the classification rates in BCI systems would make many more applications feasible and could improve the quality of life for those that are not able to communicate in other ways.

ACKNOWLEDGMENTS

Supported by NSF grants SMA 1041755 and IIS 1528214

REFERENCES

[1] Kutas M, McCarthy G, Donchin E. Augmenting mental chronometry: the P300 as a measure of stimulus evaluation time. Science. 1977 Aug 19;197(4305):792-5.

[2] Krusienski DJ, Sellers EW, Cabestaing F, Bayoudh S, McFarland DJ, Vaughan TM, et al. A comparison of classification techniques for the P300 Speller. Journal of neural engineering. 2006 Oct 26;3(4):299.

[3] Mirghasemi H, Fazel-Rezai R, Shamsollahi MB. Analysis of P300 classifiers in brain computer interface speller. InEngineering in Medicine and Biology Society, 2006. EMBS'06. 28th Annual International Conference of the IEEE 2006 Aug 30 (pp. 6205-6208). IEEE.

[4] Hinton G, Deng L, Yu D, Dahl GE, Mohamed AR, Jaitly N, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. IEEE Signal Processing Magazine. 2012 Nov;29(6):82-97.

[5] Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. InAdvances in neural information processing systems 2012 (pp. 1097-1105).

[6] Karpathy A, Toderici G, Shetty S, Leung T, Sukthankar R, Fei-Fei L. Large-scale video classification with convolutional neural networks. InProceedings of the IEEE conference on Computer Vision and Pattern Recognition 2014 (pp. 1725-1732).

[7] Donahue J, Anne Hendricks L, Guadarrama S, Rohrbach M, Venugopalan S, Saenko K, et al. Long-term recurrent convolutional networks for visual recognition and description. InProceedings of the IEEE conference on computer vision and pattern recognition 2015 (pp. 2625-2634).

[8] Cecotti H, Graser A. Convolutional neural networks for P300 detection with application to brain-computer interfaces. IEEE transactions on pattern analysis and machine intelligence. 2011 Mar;33(3):433-45.

[9] Mirowski PW, LeCun Y, Madhavan D, Kuzniecky R. Comparing SVM and convolutional networks for epileptic seizure prediction from intracranial EEG. InMachine Learning for Signal Processing, 2008. MLSP 2008. IEEE Workshop on 2008 Oct 16 (pp. 244-249). IEEE.

[10] Bashivan P, Rish I, Yeasin M, Codella N. Learning representations from EEG with deep recurrent-convolutional neural networks. arXiv preprint arXiv:1511.06448. 2015 Nov 19.

[11] Hochreiter S, Schmidhuber J. Long short-term memory. Neural computation. 1997 Nov 15;9(8):1735-80.

[12] Ji S, Xu W, Yang M, Yu K. 3D convolutional neural networks for human action recognition. IEEE transactions on pattern analysis and machine intelligence. 2013 Jan;35(1):221-31.

[13] Yosinski J, Clune J, Bengio Y, Lipson H. How transferable are features in deep neural networks?. In Advances in neural information processing systems 2014 (pp. 3320-3328).

[14] Snyder JP. Map projections–A working manual. US Government Printing Office; 1987.

[15] Alfeld P. A trivariate clough—tocher scheme for tetrahedral data. Computer Aided Geometric Design. 1984 Nov 1;1(2):169-81.

[16] Werbos PJ. Backpropagation through time: what it does and how to do it. Proceedings of the IEEE. 1990 Oct;78(10):1550-60.

[17] Ledoit, O. and Wolf, M. (2004). A well-conditioned estimator for large-dimensional covariance matrices. *Journal of Multivariate Analysis*, 88:365–411.

[18] Schaefer, J. and Strimmer, K. (2005). A shrinkage approach to large-scale covariance matrix estimation and implications for function genomics. *Stat. Appl. Genet. Mol. Biol.*, 4(1).

[19] Stivers, J.M. and de Sa, V.R. (2017). Spelling in parallel: towards a rapid, spatially independent BCI. Proceedings of the 7th Graz Brain-Computer Interface Conference 2017.