

# Incep-EEGNet: A ConvNet for Motor Imagery Decoding

Mouad Riyad<sup>(⊠)</sup>, Mohammed Khalil, and Abdellah Adib

Networks, Telecoms and Multimedia Team, LIM@II-FSTM, B.P. 146, 20650 Mohammedia, Morocco riyadmouad10gmail.com, mohammed.khalil@univh2c.ma, adib@fstm.ac.ma

Abstract. The brain-computer interface consists of connecting the brain with machines using the brainwaves as a mean of communication for several applications that help to improve human life. Unfortunately, Electroencephalography that is mainly used to measure brain activities produces noisy, non-linear and non-stationary signals that weaken the performances of Common Spatial Pattern (CSP) techniques. As a solution, deep learning waives the drawbacks of the traditional techniques, but it still not used properly. In this paper, we propose a new approach based on Convolutional Neural Networks (ConvNets) that decodes the raw signal to achieve state-of-the-art performances using an architecture based on Inception. The obtained results show that our method outperforms state-of-the-art filter bank common spatial patterns (FBCSP) and ShallowConvNet on based on the dataset IIa of the BCI Competition IV.

**Keywords:** Deep learning  $\cdot$  Electroencephalography  $\cdot$  Convolutional neural network  $\cdot$  Brain-computer interfaces

## 1 Introduction

Brain-computer interfaces (BCI) link machines and human brains with the brainwaves as mean of communication for several purposes [1]. The necessity of such a link is crucial to automatize several tasks such as the prediction of epilepsy seizure, or the detection of neurological pathologies. Also, it commonly uses brain signals as a control signal for devices such as keyboards or joysticks, which can improve the quality of life of severely disabled patients, or many non-medical applications such as video games, controlling a robot or authentication [13]. The most used sensor is electroencephalography (EEG) that relies on electrodes placed in the scalp to detect the variation of electrical activity. It processes the collected data with signal processing techniques to keep important features. Then, machine learning take a decision depending on the use case.

The most well-known applications are related to Motor Imagery (MI) [15]. It is a neural response that is produced when a person performs a movement or just imagine it. Unfortunately, the signals are intrinsically non-stationary, non-linear, and noisy [13]. Overcoming those problems requires the use of sophisticated © Springer Nature Switzerland AG 2020

A. El Moataz et al. (Eds.): ICISP 2020, LNCS 12119, pp. 103–111, 2020. https://doi.org/10.1007/978-3-030-51935-3\_11 algorithms that requires human intervention (e.g. the eye blink elimination) and computational power that can be constraining. Deep Learning permits to waive a solution to all the previously cited obstacles [9]. It extracts the features automatically without human-engineered features and classifies in the same process which enables end-to-end approaches. Several other advances in new activation function, regularization, training strategies, and data augmentation yielded to state-of-the-art performances in several fields [3,7,10]. Also, it is possible to explain the decision of deep classifiers by advance visualization methods such as weight visualization to discover the learned features.

In this paper, we propose a new convolutional neural network (Convnet) architecture based on Inception for motor imagery classification. It allows to process the data with parallel process In our approach, we use the multivariate raw signal as input with a bandpass filter as preprocessing. Therefore, we use the same first block of [12] but with higher complexity which increases the capacity of the network. Then, an Inception block will extract temporal features more efficiency which improves the performance and speeds up the learning despite the depth to reduce the degradation problem [18]. To test our approach, we use dataset IIa from the BCI Competition IV [19]. As a baseline, we compare with FBCSP and ShallowConvNet which are the state-of-the-art techniques [2]. We investigate some visualization techniques to examine the ability of our networks to extract relevant features.

The rest of the paper is organize as follows: We presents some related works in Sect. 2. We introduce our method in Sect. 3. In Sect. 4, we evaluate the performances and visualize the learned features. Section 5 discuss the result and conclude the paper.

## 2 Related Works

The first interesting approach was a ConvNet that uses raw EEG data for P300 speller application [6]. It uses convolutional layers that extract temporal and spatial features. It is inspired from Filter Banks Common Spatial Pattern (FBCSP) [2]. A convolution is performed with a kernel of size  $(1, n_t)$ , then an other convolution with a kernel with a size (C, 1) where C is the number of the channels. Then, it use a softmax layer to classifies the features extracted. [17] introduced similar architectures for MI. ShallowConvNet is a shallow convnet that is composed with the two convolutional layers then the classification layers. DeepConvNet is a deep architecture that includes more aggregation layer after the convolutional layer. ShallowConvNet outperforms state-of-the-art FBCSP. [12] proposed EEGNet as a compact version of the existing methods. It relies on Depthwise convolutional and separable convolution which permitted to reduce the number of the parameter using 796 parameters only for the EEGNet 4, 2. EEGNet performs lower than ShallowConvNet since it was not trained with the same data augmentation (cropped training) suggested by |17|. Also, cropped training requires a huge time to train which can be problematic in that cas of a takes a huge time to train, for one subjects compared with EEGNet.

## 3 Method

#### 3.1 EEG Proprieties and Data Representation

MI yields on the apparition of fluctuation of the amplitude of the neuro-signals generated in the primary sensorimotor cortex [14]. It appears as an increase and a decrease of amplitude that target specific frequency bands that are related to motor activities. They are called Event-Related Synchronization (ERS) and Event-Related Desynchronization (ERD). The  $\mu$  and  $\beta$  bands are present respectively in [8, 13] Hz and the beta band [13, 30] Hz are the targeted pattern. As input, each trial is turned into a matrix of  $\mathbb{R}^{C \times T}$  where C represents the number of electrodes and T represents the number of time samples. We sample our data at 128 Hz and we use the segment [0.5–2.5] s after cue.

#### 3.2 Incep-EEGNet

We propose Incep-EEGNet as it is illustrated in Fig. 1. It is a multistage ConvNet that is based on Inception [18]. It is composed as follows:

The first part is the same as EEGNet from [12]. They base it on two convolutional layers that act as temporal and spatial filter as act similarly to FBCSP, which is a widely used approach. We use a temporal convolutional layer with F kernel of size (1, tx) with padding. This layer will learn to extract relevant temporal features as it act as a FIR filter. We choose a size of 32 which correspond to a duration of 0.25 s of a signal sampled at 128 Hz. A second convolution is used to extract the spatial feature. It relies on Depthwise convolution that produces the number of feature maps per input which reduces considerably the computational cost. It is a convolution with a size of (C, 1) where C represents the number of channels. Also, we use batch normalization after each convolution and activation after the second one. This layer will allow only the important electrodes to contribute to the decision and learning frequency-specific spatial filter with Depthwise convolution where it controls the number of connections by the depth parameter D.

In the second part, we introduce the novelty of this architecture which is an inception based block. This block comes as a solution to the inconvenience of EEGNET that is too shallow and too compact, which restricts the capacity of the networks leading to overfitting in most cases. Even with a deeper network, the performance still low because of a degradation problem for DeepConvNet. Hence, we suggest to use an inception stage based That will learn features from several branches:

- A convolutional branch with a convolution with a kernel size of (1,7).
- A convolutional branch with a convolution with a kernel size of (1,9).
- A branch with a pointwise convolution with a kernel size of 1, 1 with a stride of (1, 2)
- A branch with an average pooling with a kernel size of

We merge the output of the different branches by stacking them along with the feature map dimension. We apply batch normalization and an activation. The use of dropout restricted only after final the activation cause we observed no improvement. Each convolutional branch include a pointwise convolution that reduces the number of feature map to 64 and an average pooling layer with a size of (1, 2).

In the final part, we use an additional convolutional layer with a F \* D kernel with a size of (1, 5) along with batchnormalization, activation, and dropout. We use an Global AveragePooling layer to reduce the number of parameters to 2\*F. Then, we use Softmax classification with 4 units that represent the 4 classes of the dataset.



Fig. 1. Architecture of the proposed system with layers hyperparameters

#### 3.3 Hyperparameters and Training

Our implementation uses publicly available codes of preprocessing based on *braindecode* [17]. We trained deep learning methods on a NVIDIA P100 1.12.0. We train our method by optimizing the categorical cross-entropy using ADAM Optimizer [11] with Nesterov. Dropout probability is 0.5 as advised by [3]. We use a batch size of 64 as for EEGNET [12]. We fix the network parameter to F = 64 and D = 4. Exponential Linear Unit (ELU) is chosen as the activation [7]. We train our ConvNets as follows: We train for 100 epochs with a learning

rate (Lr) of  $5 \times 10^{-4}$ . At the end of the training, we retrain it for 50 epochs and Lr set to  $1 \times 10^{-4}$  with the merged training and validation set. Once again, we do the same operation for 30 epochs and a Lr set to  $2 \times 10^{-5}$ . Similar training was done for ShallowConvNet [17].

### 4 Experiment

#### 4.1 Dataset

As a dataset, we use the dataset IIa from the BCI competition IV [19]. It contains EEG data of four MI tasks (right hand, left hand, foot, and tongue imagined movements) from nine subjects. It uses a set of 22 electrodes placed on the scalp. The recording was on two different sessions where the first was defined as a training set and the second one as a testing set. The subjects are asked to performs 288 MI tasks per session (72 trials for each class) after a cue that was. The original data is sampled at 240 Hz and filtered with a bandpass filter between 0.1 Hz and 100 Hz. We add additional preprocessing to the data as described in [17]. We resample the signals at 128 Hz and filter with a bandpass filter between 1 Hz and 32 Hz. We use 20% of the training set as a validation set. We use a cropping data augmentation by extracting the segments [0.3, 2.3] s, [0.4, 2.4] s, [0.5, 2.5] s, [0.6, 2.6] s, [0.7, 2.7] s post cue only on the training set (1152 trials). The validation and testing set contain only [0.5, 2.5] s segment to prevent leaking (for validation set) that can compromise the training. Therefor, the input will have a shape of  $22 \times 256$ .

#### 4.2 Results

To assert the performances of our method, we compare with FBCSP, Riemannian geometry [4], Bayesian optimization [5], and ShallowNet [17]. Table 1 shows the results of the classification of our method and the baselines in terms of accuracy. It shows that the proposed method outperforms the baselines for several subjects (S2, S3, S5, S6, S7, S9). However, BO got better results for S1 and S8, when ShallowNet performs better for S4. On the other hand, FBCSP2 and RG did not achieve higher results. For an advanced evaluation, we conduct statistical testing with the Wilcoxon test. To evaluate the significance of the results on the mean value. It shows that our method has a statistically significant difference compared with BO with p < 0.05. Comparing with FBCSP2 and RG, the difference is highly significant with p < 0.01.

Table 2 shows the results of the classification of our method and the baselines in terms of kappa. The result shows that our method outperforms for most of the subjects. It only failed to outperform FBCSP1 for S2 and ShallowNet for S4. Once Again, FBCSP2 and RG got bad results. Statistical testing shows that the increase in mean kappa is statistically significant with p < 0.05 for FBCSP1, MDRM, and ShallowNet. For the other methods, the difference is highly significant at p < 0.01.

	во	FBCSP2	RG	ShallowNet	Incep-EEGNet
S1	82.120	75.694	77.778	75.347	78.472
S2	44.860	44.792	43.750	43.056	52.778
S3	86.600	85.069	83.681	80.208	89.931
S4	66.280	63.542	56.597	68.056	66.667
S5	48.720	59.028	47.917	58.681	61.111
S6	53.300	36.458	47.569	49.306	60.417
S7	72.640	86.111	78.472	85.417	90.625
S8	82.330	79.167	79.861	77.778	82.292
S9	76.350	82.639	81.250	80.556	84.375
Average	68.133	68.056	66.319	68.711	74.074
p-values	0.038	0.008	0.008	0.011	1.000

Table 1. Classification accuracy (%) comparaison of our methods and the baselines,

Table 3 and Table 4 show the confusion matrix of Incep-EEGNet and FBCSP2 respectively. They show that both methods have difficulties to classify foot classes. Also, they confuse between right-hand and left-hand classes. Performances of our method are better than the reference.

	FBCSP1	2nd	MDRM	FBCSP2	RG	ShallowNet	Incep-EEGNet
S1	0.680	0.690	0.750	0.676	0.704	0.671	0.713
S2	0.420	0.340	0.370	0.264	0.250	0.241	0.370
S3	0.750	0.710	0.660	0.801	0.782	0.736	0.866
S4	0.480	0.440	0.530	0.514	0.421	0.574	0.556
S5	0.400	0.160	0.290	0.454	0.306	0.449	0.481
S6	0.270	0.210	0.270	0.153	0.301	0.324	0.472
S7	0.770	0.660	0.560	0.815	0.713	0.806	0.875
S8	0.750	0.730	0.580	0.722	0.731	0.704	0.764
S9	0.610	0.690	0.680	0.769	0.750	0.741	0.792
Average	0.570	0.514	0.521	0.574	0.551	0.583	0.654
p-values	0.021	0.008	0.021	0.008	0.008	0.011	1.000

Table 2. Kappa values comparison of our methods and the baselines

Figure 2a represents the Fourier transform of a temporal filter learned in the first convolution. It was designed to extract the temporal features of the EEG signals. As it was expected, Incep-EEGNet learned exactly the frequencies that are involved in the MI neural response. Also, we observe that there is a peak at 55 Hz, which can indicate that MI may be also characterized by this band as was

		Predicted				
		L	R	F	Т	
Actual	L	80.40	8.80	5.09	5.71	
	R	15.59	74.38	4.48	5.56	
	F	9.88	7.56	65.43	17.13	
	Т	10.80	5.56	7.56	76.08	

Table 3. Confusion matrix of Incep-

EEGNet

 Table 4. Confusion matrix of FBCSP

		Predicted				
		L	R	F	Т	
Actual	L	73.30	15.28	4.78	6.64	
	R	14.97	73.92	5.71	5.40	
	F	8.64	13.43	56.02	21.91	
	Т	11.57	11.27	8.18	68.98	



Fig. 2. Sample of relevant convolutional weights.

reported by [8]. Figure 2b shows a spatial filter reconstructed by interpolation of the weights. The scale in the right is from 1 to -1. It shows that Incep-EEGNet extracts the signals from the electrodes C3, CZ, and C4. It happens that those electrodes cover the part of the brain that is responsible for the movement of the hands and the feet.

#### 5 Discussion and Conclusion

Designing ConvNets for BCI applications may be problematic. The existing approaches need an intensive data augmentation, and to be Shallow. Deep ConvNets are defective and lacks performances. Therefore, we built the Incep-EEGnet which is a modified EEGNET with a greater number of feature map that increases the complexity of the model where it outperforms state-of-the-art methods. To diminish any problem of degradation, we use an inception block that has several branches that offer an efficient feature extraction layer. The pointwise convolution works as a residual connection that prevents from vanishing gradient problems. Incep-EEGNet outperforms FBCSP, RG, and several ConvNets. Indeed, CSP techniques are considered state-of-the-art techniques for their efficiency, but as drawbacks, they are sensitive to noises, artifacts, and need larger datasets [16]. RG relies on and representation of the data that does not take into account the frequential features as its authors praise. But, it lowers its performances compared with FBCSP and ConvNets. ConvNet methods perform better and faster in the same conditions if we wisely use them. The overall performances are still low for several subjects highlighting a strong incompatibility between some subjects.

## References

- Abdulkader, S.N., Atia, A., Mostafa, M.S.M.: Brain computer interfacing: applications and challenges. Egypt. Inform. J. 16(2), 213–230 (2015). https://doi.org/ 10.1016/j.eij.2015.06.002
- Ang, K.K., Chin, Z.Y., Wang, C., Guan, C., Zhang, H.: Filter bank common spatial pattern algorithm on BCI competition IV datasets 2a and 2b. Front. Neurosci. 6, 39 (2012)
- 3. Baldi, P., Sadowski, P.J.: Understanding dropout, p. 9
- Barachant, A., Bonnet, S., Congedo, M., Jutten, C.: Multiclass brain-computer interface classification by Riemannian geometry. IEEE Trans. Biomed. Eng. 59(4), 920–928 (2012)
- Bashashati, H., Ward, R.K., Bashashati, A.: User-customized brain computer interfaces using Bayesian optimization. J. Neural Eng. 13(2), 026001 (2016). https:// doi.org/10.1088/1741-2560/13/2/026001. 00007
- Cecotti, H., Graser, A.: Convolutional neural networks for P300 detection with application to brain-computer interfaces. IEEE Trans. Pattern Anal. Mach. Intell. 33(3), 433–445 (2011). https://doi.org/10.1109/TPAMI.2010.125
- Clevert, D.A., Unterthiner, T., Hochreiter, S.: Fast and accurate deep network learning by exponential linear units (ELUs). In: International Conference on Learning Representations (ICLR) (2016)
- Dose, H., Møller, J.S., Iversen, H.K., Puthusserypady, S.: An end-to-end deep learning approach to MI-EEG signal classification for BCIs. Expert Syst. Appl. 114, 532–542 (2018). https://doi.org/10.1016/j.eswa.2018.08.031.00015
- 9. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning: Adaptive Computation and Machine Learning. The MIT Press, Cambridge (2016)
- Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. In: Bach, F., Blei, D. (eds.) Proceedings of the 32nd International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 37, pp. 448–456. PMLR, Lille, July 2015. 16886
- 11. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. In: Proceedings of the 3rd International Conference on Learning Representations (ICLR), December 2014
- Lawhern, V.J., Solon, A.J., Waytowich, N.R., Gordon, S.M., Hung, C.P., Lance, B.J.: EEGNet: a compact convolutional neural network for EEG-based braincomputer interfaces. J. Neural Eng. 15(5), 056013 (2018). https://doi.org/10.1088/ 1741-2552/aace8c
- Ortiz-Rosario, A., Adeli, H.: Brain-computer interface technologies: from signal to action. Rev. Neurosci. 24(5) (2013). https://doi.org/10.1515/revneuro-2013-0032
- Pfurtscheller, G., Neuper, C.: Motor imagery and direct brain-computer communication. Proc. IEEE 89(7), 1123–1134 (2001). https://doi.org/10.1109/5.939829
- Pfurtscheller, G., Neuper, C.: Movement and ERD/ERS. In: Jahanshahi, M., Hallett, M. (eds.) The Bereitschaftspotential: Movement-Related Cortical Potentials, pp. 191–206. Springer, Boston (2003). https://doi.org/10.1007/978-1-4615-0189-3\_12.00054

- 16. Reuderink, B., Poel, M.: Robustness of the common spatial patterns algorithm in the BCI-pipeline. Technical report, University of Twente (2008). 00042
- Schirrmeister, R.T., et al.: Deep learning with convolutional neural networks for EEG decoding and visualization. Hum. Brain Mapp. 38(11), 5391–5420 (2017). https://doi.org/10.1002/hbm.23730. Convolutional Neural Networks in EEG Analysis
- Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2818–2826. IEEE (2016). 01916
- Tangermann, M., et al.: Review of the BCI competition IV. Front. Neurosci. 6 (2012). https://doi.org/10.3389/fnins.2012.00055