Check for updates

Neural Network Including Alternative Pre-processing for Electroencephalogram by Transposed Convolution

Kenshi Machida(✉), Isao Nambu, and Yasuhiro Wada

Nagaoka University of Technology,
1603-1 Kamitomioka, Nagaoka, Niigata 940-2188, Japan
kmachida@stn.nagaokaut.ac.jp

Abstract. In the classification of electroencephalograms for a brain-computer interface (BCI), two steps are generally applied: preprocessing for feature extraction and classification using a classifier. As a result, combinations of a myriad of preprocessing and a myriad of classification method have disordered for each classification target and data. Conversely, neural networks can be applied to any classification problem because they can transform an arbitrary form of input into an arbitrary form of output. We considered a transposed convolution as a preprocessor that can set the window width and number of output features and classified it using a convolutional neural network (CNN). Using a simple CNN with a transposed convolution in the first layer, we classified the data of the motor imagery tasks of the BCI competition IV 2 dataset. The results showed that, despite not being among the best conventional methods available, we were still able to obtain a high degree of accuracy.

Keywords: BCI competition IV · Electroencephalogram · Neural network · CNN · Transposed convolution

1 Introduction

Most studies on electroencephalography (EEG) in this field have focused on the brain-computer interface (BCI). Further, numerous studies have applied EEG classification techniques to achieve such an interface. A typical example is a method using a dataset from a BCI competition, and the latest dataset is from the BCI Competition IV held in 2008, although methods with a higher classification accuracy have since been reported. For example, Ang et al. [1] applied a Filter Bank Common Spatial Pattern and conducted a classification using a naÏve Bayesian Parzen window classifier. In addition, Gaur et al. [2] applied subject-specific multivariate empirical mode decomposition-based filtering and conducted a classification using the minimum distance to Riemannian mean method. As described above, when analyzing an EEG, a two-step process, namely, preprocessing followed by a classification, is generally applied. Hence,

© Springer Nature Switzerland AG 2020
H. Yang et al. (Eds.): ICONIP 2020, CCIS 1333, pp. 139–146, 2020.
https://doi.org/10.1007/978-3-030-63823-8_17

various combinations of pre-processing and classification methods have resulted in inconsistencies for each classification target. However, it remains unclear whether such methods have been applied to a BCI.

The most effective technique to achieve a BCI through modern technology is the use of a neural network. Neural networks convert any input into any output and can thus be applied to any classification problem. In addition, a processor specialized for neural network processing was developed [4] and has recently been actively incorporated into small devices, such as smartphones. The environment in which this approach can be implemented is becoming more popular now.

To date, EEG classification has been conducted in two stages: feature extraction through a preprocessing and classification using a classifier. A neural network may convert an input into any output, that is, a model obtaining a classification label should be used by entering an EEG directly. In this study, we demonstrate that a previously applied window analysis of the signals can be primitively reproduced on a neural network and that a high classification accuracy can be obtained.

2 Method

A neural network must have the flexibility to cope with various types of data, including images and time signals, and be able to arbitrarily adjust the degree of freedom of the calculation. Conversely, an overlearning is likely to occur, and depending on the network configuration, numerous hyperparameters must be adjusted. In addition, when learning, many techniques such as adjusting the learning rate and early stopping must be considered. However, such situations can be avoided by applying only a few techniques.

2.1 Batch Normalization

Batch normalization [5] is a method used to normalize the input to each layer of a neural network such that a mean of zero and a variance of 1 are achieved.

According to [6], the higher the learning rate that is set, the more regularized the regularization effect becomes because the amount of noise is increased, owing to the mini-batch selection. When not normalized, the mean and variance increase exponentially as the layer deepens, but they can be kept constant by applying batch normalization, thereby making the gradient highly dependent on the input. Furthermore, the error does not diverge even when a high learning rate is set.

2.2 Convolution Layer

A convolution layer is used for inputs with one or more dimensions of spatial features. A convolution is applied to the spatial feature directions using a kernel. In a convolution layer, the number of weighting parameters is smaller than the number of input–output features and, thus, is regularized. Because an EEG is a time signal, the necessity of considering the regularization parameter is reduced by using a convolution layer that applies a convolution in the time direction.

2.3 Transposed Convolution

A transposed convolution is a conversion applied opposite a convolution. If a convolution is an encoder, a transposed convolution behaves similar to a decoder. An up-sampling can be conducted in the spatial feature direction, and the feature map can be projected onto a higher dimensional space.

2.4 Network Structure

Two different network structures were applied in this study. One is a 1D CNN model that convolves the EEG only in the time-axis direction. In this case, global average pooling was conducted after the four convolution layers, and classification was applied using all connected layers. The other is a 2D CNN model that applies a convolution layer after transforming it into a form with 2D spatial features using a transposed convolution. There are several benefits to converting from 1D to 2D. For example, when a convolution layer is used, the ratio of the number of weight parameters to the number of inputs and outputs decreases; thus, the regularization effect is enhanced. In addition, because a 2D CNN model has many useful learned models, some of which have achieved high results in the ImageNet Large-Scale Visual Recognition Challenge, the model can be transferred. In general, during the process of transforming 1D spatial features into 2D spatial features through a transformed convolution, a window analysis is applied on the time signal, and the features are extended in the new axial direction. Thus, the calculation corresponding to the preprocessing performed thus far can be primitively reproduced by the neural network.

Although the input EEG has spatial features in two directions, namely, the time direction and the channel direction, the channel direction is arranged in order of the channel number of the EEG at the time of measurement and is not spatially a significant feature. The EEG channel direction is placed the unconstrained channel direction of the convolution layer.

Tables 1 and 2 show the specific configurations of the above two models:

Table 1. 1D CNN model structure

Layer	1D CNN model	
	Structure	Output shape
Input	–	(250, 1, 22)
1	Conv [3, 1] 22ch Batch Normalization Leaky ReLU(α = 0.2)	(248, 1, 22)
2	Conv [3, 1] 22ch Batch Normalization Leaky ReLU(α = 0.2)	(246, 1, 22)
3	Conv [3, 1] 32ch Batch Normalization Leaky ReLU(α = 0.2)	(244, 1, 32)
4	Conv [3, 1] 32ch Batch Normalization Leaky ReLU(α = 0.2)	(242, 1, 32)
5	Global Average Pooling	(32)
6	Dense [4] Softmax	(4)

Table 2. 2D CNN model structure

Layer	2D CNN model	
	Structure	Output shape
Input	–	(250, 1, 22)
1	Transposed Conv [8, 150] 8ch Batch Normalization	(257, 150, 8)
2	Conv [3, 3] 8ch Batch Normalization Leaky ReLU(α = 0.2)	(255, 148, 8)
3	Conv [3, 3] 16ch Batch Normalization Leaky ReLU(α = 0.2)	(253, 146, 16)
4	Conv [3, 3] 32ch Batch Normalization Leaky ReLU(α = 0.2)	(251, 144, 32)
5	Conv [3, 3] 32ch Batch Normalization Leaky ReLU(α = 0.2)	(249, 142, 32)
6	Global Average Pooling	(32)
7	Dense [4] Softmax	(4)

2.5 Dataset

The BCI competition IV 2a dataset [3] was used for the network evaluation. This dataset includes EEG signals recorded from nine subjects on four types of motor image tasks: right hand, left hand, tongue, and foot. A total of 22 EEG channels is applied, and the sampling frequency 250 Hz. A total of 288 training and test data trials are recorded for each subject, including missing parts and some excluded trials.

In this case, after replacing the missing values with zeros and applying a low-pass filter, the sampling frequency was down-sampled 63 Hz, which was approximately one-fourth of the original rate, and all trials including the excluded trials were applied. As the input, a 4s signal was used, for which a motor image was shown during the trial. The signal was normalized to achieve an average value of zero and a variance of 1 for each trial, as well as for each channel before being input into the neural network.

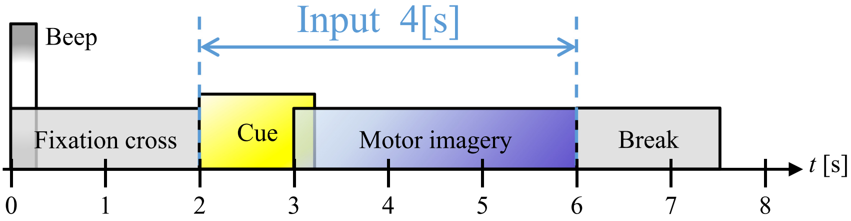


Fig. 1. Timing scheme of the paradigm

2.6 Training and Evaluation

The cross-entropy is used as the cost function. In addition, Adam [7] was used as the optimizer, the mini-batch size was 58, and the learning rate was 0.0001. The parameters were updated 1,000 times without changing the learning rate or stopping the learning early.

5-fold cross-validation was conducted for each subject 6 times using the training data, and the classification accuracy of the test data was observed 30 times.

3 Results

Table 3 shows the mean and standard deviation (std) of accuracy when the validation and test data were classified in the model after the parameter update. The classification accuracy of the 2D CNN model was higher than that of the 1D CNN model.

Table 4 shows the accuracy when converted into the kappa value and compared with the value from a previous study. The p-value has been computed using the Wilcoxon signed rank test. The neural network containing the Transposed Convolution achieved an mean kappa of 0.62, which is superior to all previously reported results.

Table 3. Accuracy validation and evaluation

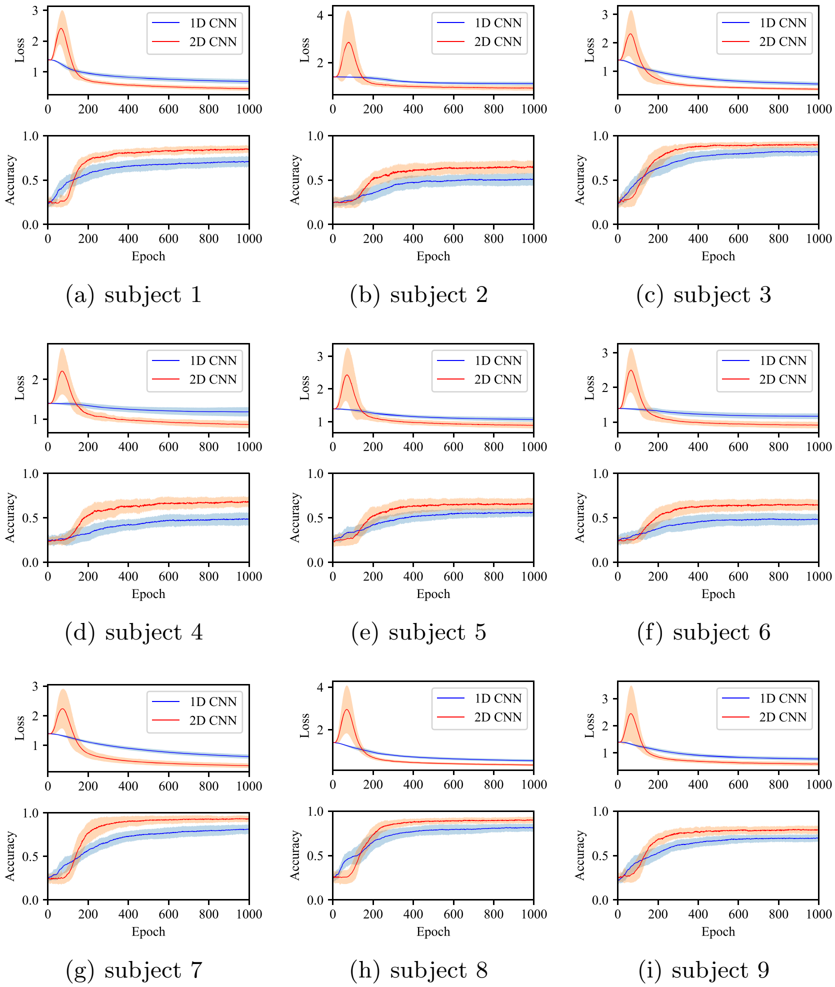
Subject	5-fold cross-validation					Evaluation				
	1D CNN		2D CNN		2D-1D	1D CNN		2D CNN		2D-1D
	mean	std	mean	std		mean	std	mean	std	
1	70.41	± 5.97	84.91	± 4.24	14.50	75.61	± 2.40	86.79	± 2.10	11.18
2	51.17	± 7.13	64.09	± 7.34	12.92	39.44	± 2.85	46.84	± 3.10	7.40
3	81.99	± 5.08	89.71	± 3.67	7.72	81.79	± 1.86	90.76	± 1.04	8.97
4	48.83	± 7.32	68.83	± 6.34	20.00	48.78	± 2.05	66.33	± 4.70	17.55
5	56.49	± 4.97	65.15	± 6.79	8.65	49.68	± 2.73	55.68	± 3.37	6.01
6	48.54	± 6.18	64.91	± 6.49	16.37	46.01	± 2.79	49.88	± 3.80	3.88
7	81.05	± 5.37	93.16	± 3.34	12.11	73.67	± 3.00	88.09	± 2.01	14.42
8	81.40	± 4.12	89.94	± 3.26	8.54	74.48	± 2.09	81.83	± 1.55	7.35
9	69.42	± 3.91	79.77	± 3.97	10.35	73.61	± 2.24	78.10	± 2.75	4.49
Average	65.48%		77.83%		12.35	62.56%		71.59%		9.03

4 Discussion

A high-level accuracy was obtained by extending an EEG into a 2D map using a transposed convolution. Using this method, a transposed convolution was applied to extend a signal with a fixed window width in a new axis direction. The best-known method for extending a signal to a 2D map during a conventional

Table 4. Evaluated kappa value comparison with previous studies

Subject	Kappa value			
	1D CNN	2D CNN	Method-[2]	Method-[1]
1	0.675	0.824	0.86	0.75
2	0.193	0.291	0.24	0.37
3	0.757	0.877	0.70	0.66
4	0.317	0.551	0.68	0.53
5	0.329	0.409	0.36	0.29
6	0.280	0.332	0.34	0.27
7	0.649	0.841	0.66	0.56
8	0.660	0.758	0.75	0.58
9	0.648	0.708	0.82	0.68
Average	0.501	0.621	0.60	0.52
p value	0.0039	–	0.5898	0.0391



(a) subject 1 (b) subject 2 (c) subject 3

(d) subject 4 (e) subject 5 (f) subject 6

(g) subject 7 (h) subject 8 (i) subject 9

Fig. 2. Loss and accuracy curve of the validation data

analysis is a short-term Fourier transform (STFT). For example, when an STFT is used, the frequency axis is extended by the same number as the window width. However, only frequency information at equal intervals is extracted, and the validity of information is not guaranteed. Therefore, by incorporating the preprocessing into a neural network, the window width and number of expansions can be set arbitrarily, and learning is applied to extract features effective for classification. The idea here is that, by visualizing the parameters and the output of each layer, understanding of the EEG can be improved by analyzing the components of the source necessary for the classification that are included in the signal.

In this neural network, the learning rate can be an important parameter because batch normalization is used in each layer. Although the weight parameter does not diverge owing to the effect of such normalization, if the learning rate is too large, the regularization effect becomes stronger and the accuracy does not increase. Figure 2 shows the mean and std of learning curve for 6 times 5-fold cross-validation. When observing the transition of the validation loss, the loss continues to decrease near the end of learning. That is, the accuracy might be improved by adjusting the learning rate for each validation or according to the learning epoch.

5 Conclusion

We attempted to learn the transformation equivalent of a preprocessing with arbitrary window widths and spatial sizes by applying a transposed convolution to EEG signals, followed by using a network of simple structures to which a 2D CNN can be naturally applied. We showed that high accuracy can be achieved by replacing the complex preprocessing of EEGs with a neural network.

Acknowledgements. This work was partly supported by JSPS KAKENHI Grant Numbers 18K19807 and 18H04109, KDDI foundation, and Nagaoka University of Technology Presidential Research Grant. We would like to thank Editage (www.editage.com) for English language editing.

References

1. 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. Frontiers Neurosci. **6**, 39 (2012). https://doi.org/10.3389/fnins.2012.00039. https://www.frontiersin.org/article/10.3389/fnins.2012.00039
2. Gaur, P., Pachori, R.B., Wang, H., Prasad, G.: A multi-class EEG-based BCI classification using multivariate empirical mode decomposition based filtering and Riemannian geometry. Expert Syst. Appl. **95**, 201-211 (2018). https://doi.org/https://doi.org/10.1016/j.eswa.2017.11.007. http://www.sciencedirect.com/science/article/pii/S0957417417307492
3. Brunner, C., Leeb, R., Muller-Putz, G., Schlogl, A., Pfurtscheller, G.: BCI competition 2008-graz data set a. Institute for Knowledge Discovery (Laboratory of Brain–Computer Interfaces), Graz University of Technology (2008). http://bbci.de/competition/iv/desc_2a.pdf
4. Jouppi, N.P., et al.: In-datacenter performance analysis of a tensor processing unit. SIGARCH Comput. Archit. News **45**(2), 1-12 (2017). https://doi.org/10.1145/3140659.3080246
5. 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, Lille, France, 07–09 July 2015, vol. 37, pp. 448-456. PMLR (2015). http://proceedings.mlr.press/v37/ioffe15.html

6. Bjorck, N., Gomes, C.P., Selman, B., Weinberger, K.Q.: Understanding batch normalization. In: Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R. (eds.) Advances in Neural Information Processing Systems, vol. 31, pp. 7694–7705. Curran Associates, Inc. (2018). http://papers.nips.cc/paper/7996-understanding-batch-normalization.pdf
7. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. In: Bengio, Y., LeCun, Y. (eds.) 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, 7–9 May 2015, Conference Track Proceedings (2015). http://arxiv.org/abs/1412.6980