Biomedical Signal Processing and Control 58 (2020) 101837

Contents lists available at ScienceDirect

Biomedical Signal Processing and Control

journal homepage: www.elsevier.com/locate/bspc

Evaluation of convolutional neural networks using a large multi-subject P300 dataset

Lukáš Vařeka

NTIS – New Technologies for the Information Society, Faculty of Applied Sciences, University of West Bohemia, Univerzitni 8, 306 14 Pilsen, Czech Republic

ARTICLE INFO

Article history:
Received 6 August 2019
Received in revised form 4 December 2019
Accepted 31 December 2019

Keywords:
Convolutional neural networks
Event-related potentials
P300
BCI
LDA
Machine learning

ABSTRACT

Deep neural networks (DNN) have been studied in various machine learning areas. For example, event-related potential (ERP) signal classification is a highly complex task potentially suitable for DNN as signal-to-noise ratio is low, and underlying spatial and temporal patterns display a large intra- and inter-subject variability. Convolutional neural networks (CNN) have been compared with baseline traditional models, i.e. linear discriminant analysis (LDA) and support vector machines (SVM) for single trial classification using a large multi-subject publicly available P300 dataset of school-age children (138 males and 112 females). For single trial classification, classification accuracy stayed between 62% and 64% for all tested classification models. When applying the trained classification models to averaged trials, accuracy increased to 76–79% without significant differences among classification models. CNN did not prove superior to baseline for the tested dataset. Comparison with related literature, limitations and future directions are discussed.

1. Introduction

In recent years, both fundamental and applied research in deep learning has rapidly developed. In image processing and natural language processing, it has led to significantly better classification rates than previous state-of-the-art algorithms [7]. Therefore, there has been a growing interest in applying deep neural networks (DNNs) to various fields of applied research. Such an effort can also be seen in electroencephalographic (EEG) data processing and classification. A well-known application of EEG classification is a brain-computer interface (BCI) [18] which allows immobile persons to operate devices only by decoding their intent from EEG signal without any need for muscle involvement. A significant challenge in BCI systems is to recognize the intention of the user correctly since the brain components of interest often have a significantly lower amplitude than random EEG signal [18].

DNNs often do not require costly feature engineering, and thus could lead to more universal and reliable EEG classification. However, recent review of the field reached a conclusion that so far, these benefits have not been convincingly presented in the literature [14]. Many studies did not compare the studied DNN to state-of-the-art BCI methods or performed biased comparisons, with either suboptimal parameters for the state-of-the-art competitors or with unjustified choices of parameters for the DNN [14]. Similar conclusion has been reached in another review of DNN and EEG [22]. Many related papers suffer from poor reproducibility: a majority of papers would be hard or impossible to reproduce given the unavailability of their data and code [22]. Moreover, one of the drawbacks of DNNs is having to collect a large training dataset. Typical BCI datasets have very small numbers of training examples, since BCI users cannot be asked to perform thousands of mental operations before actually using the BCI. To overcome this problem, it has been proposed to obtain BCI applications with very large training data bases, e.g. for multi-subject classification. Multi-subject classification has one more advantage — it solves the problem of DNN long training times. Instead, a universal BCI system can be trained only once and then just applied to a new dataset from a new user without any additional training [14].

Guess the number (GTN) is a simple P300 event-related potential (ERP) BCI experiment. Its aim is to ask the measured participant to pick a number between 1 and 9. Then, he or she is exposed to corresponding visual stimuli. The P300 waveform is expected following the selected (target) number. During the measurement, experimenters try to guess the selected number based on manual evaluation of average ERPs associated with each number. Finally, both the numbers thought and guesses of the experimenters are recorded as metadata. 250 school-age children participated in the experiments that were carried out in elementary and secondary schools in the Czech Republic. Only three EEG channels (Fz, Cz,

E-mail address: lvareka@ntis.zcu.cz

https://doi.org/10.1016/j.bspc.2019.101837
1746-8094/© 2020 Elsevier Ltd. All rights reserved.

Pz) were recorded to decrease preparation time. Nevertheless, to the author's best knowledge, this is the largest P300 BCI dataset available so far [19].

The main aim of this paper is to evaluate one of the deep learning models, convolutional neural networks (CNN) for classification of P300 BCI data. Unlike most related studies, multi-subject classification was performed with the future goal of developing a universal BCI. Two state-of-the art BCI classifiers were used as baseline to minimize the risk of biased comparison. To avoid overtraining, cross-validation and final testing using a previously unused part of the dataset were performed. Another aim of this manuscript is to evaluate some CNN parameters in this application.

1.1. State-of-the-art

Although various BCI algorithms have been evaluated and published in recent decades, there is still no feature extraction or machine learning algorithm clearly established as state-of-the-art. However, several studies have focused on reviews and comparisons with partly consistent results. In [12], a comparison of several classifiers (Pearson's correlation method, Fisher's linear discriminant analysis (LDA), stepwise linear discriminant analysis (SWLDA), linear support-vector machine (SVM), and Gaussian kernel support vector machine (nSVM)) was performed on 8 healthy subjects. It was shown that SWLDA and LDA achieved the best overall performance. As originally proposed by Blankertz et al. [3] and also confirmed in a recent review [14], shrinkage LDA is another useful tool for BCI, particularly with small training datasets. In [16], the authors demonstrated that LDA and Bayesian linear discriminant analysis (BLDA) were able to beat other classification algorithms.

Efforts to develop a universal multi-subject P300 BCI machine learning have been relatively rare in the literature. In [21], the authors developed a generic shrinkage LDA classifier using the training data of 18 subjects. The performance was evaluated with the data of 7 subjects. It was concluded that generic classifier achieved comparable results regarding the effectiveness and efficiency as personalized classifiers.

2. Methods

2.1. Data acquisition

The data described in detail and accessible in [19] were used in subsequent experiments. The measurements were taken between 8 am and 3 pm. Unfortunately, the environment was usually quite noisy since many children and also many electrical devices were present in the room at the same time. However, in any case there were no people standing or moving behind the monitor or in the close proximity of the measured participant.

The participants were stimulated with numbers between 1 and 9 flashing on the monitor in random order. The numbers were white on the black background. The inter-stimulus interval was set to 1500 ms. The following hardware devices were used: the BrainVision standard V-Amp amplifier, standard small or medium 10/20 EEG cap, monitor for presenting the numbers, and two notebooks necessary to run stimulation and recording software applications. The reference electrode was placed at the root of the nose and the ground electrode was placed on the ear. To speed up the guessing task, only three electrodes, Fz, Cz and Pz, were active. The stimulation protocol was developed and run using the Presentation software tool produced by Neurobehavioral Systems, Inc. The BrainVision Recorder was used for recording raw EEG data.

The participants were school-age children and teenagers (aged between 7 and 17; average age 12.9), 138 males and 112 females. All participants and their parents were informed about the programme of the day and the experiments carried out. All participants took part in the experiment voluntarily. The gender, age, and laterality of the participants were collected. No personal or sensitive data were recorded.

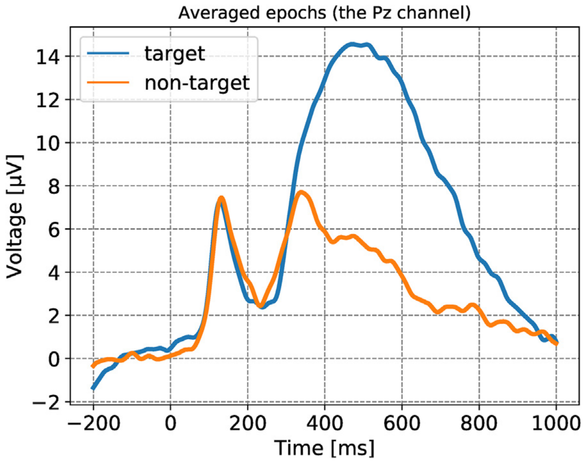

Fig. 1. Comparison of target and non-target epoch grand averages. As expected, there is a large P300 component following the target stimuli. Note that the P300 average latency is somewhat delayed compared to what is commonly reported in the literature [15].

2.2. Preprocessing and feature extraction

The data were preprocessed as follows:

1. From each participant of the experiments, short parts of the signal (i.e. ERP trials, epochs) associated with two numbers displayed were extracted. One of them was the target (thought) number. Another one was randomly selected number out of the remaining stimuli between 1 and 9. Consequently, similar number of training examples for both classification classes (target, non-target) was extracted. The extracted epochs were stored into a file (available in [20]).
2. For epoch extraction, intervals between 200 ms prestimulus and 1000 ms poststimulus were used. The prestimulus interval between −200 and 0 ms was used for baseline correction, i.e. computing average of this period and subtracting it from the data. Thus given the sampling frequency of 1 kHz, $11,532 \times 3 \times 1200$ (number of epochs × number of EEG channels x number of samples) data matrix was produced.
3. To skip severely damaged epochs, especially caused by eye blinks or bad channels, amplitude threshold was set to 100 μV according to common guidelines (such as in [15]). Any epoch $x[c, t]$ with c being the channel index and t time was rejected if:

$$\max_{c,t} |x[c, t]| > 100 \tag{1}$$

With this procedure, 30.3% of epochs were rejected. In Fig. 1, grand averages of accepted epochs (across all participants) are depicted.

Feature extraction. Many deep learning methods such as CNN are designed to avoid significant feature engineering [2,28]. On the other hand, linear classifiers usually perform better when the dimensionality of the original data matrix is reduced, and only the most significant features are extracted [3]. In the parameter optimization phase, state-of-the-art classifiers were used either with original data dimension, or after feature selection proposed in [3] to compare the performance. The feature extraction method was

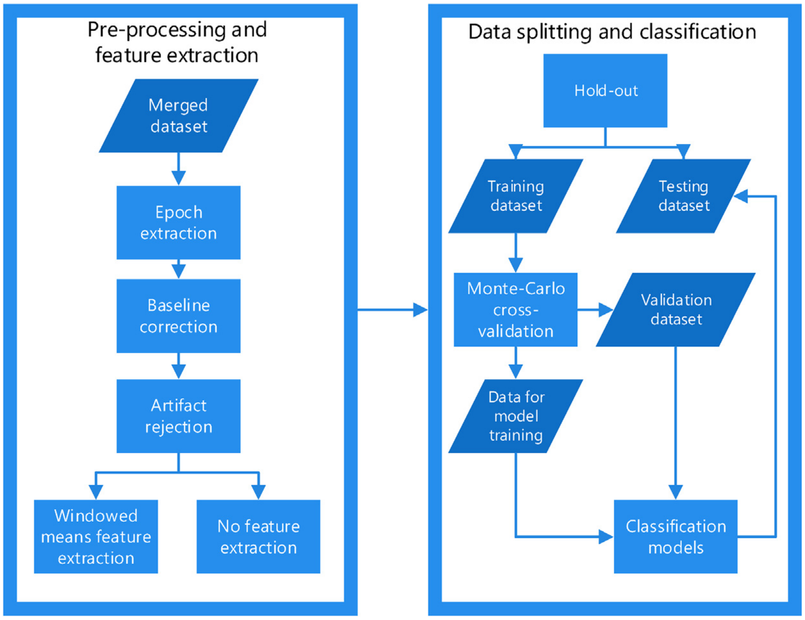

Fig. 2. Flowchart of preprocessing, feature extraction and data splitting applied.

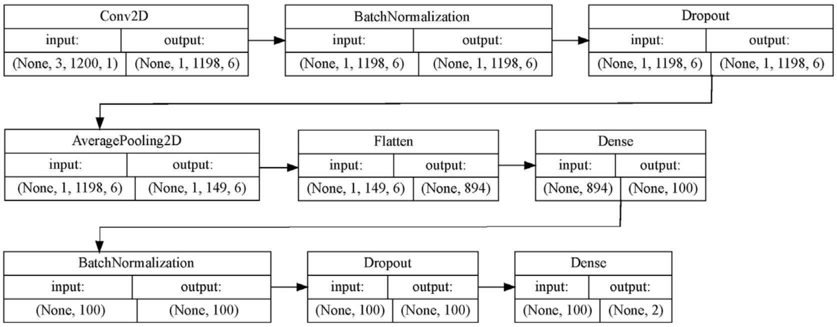

Fig. 3. Architecture of the convolutional neural network. There was one convolutional layer, one dense layer, and finally a softmax layer for binary classification (target/non-target). Batch normalization and dropout followed both the convolutional and dense layers.

based on averaging time intervals of interest and merging these averages across all relevant EEG channels to get reduced spatio-temporal feature vectors (Windowed means feature extraction, WM). In line with recommendations for P300 BCIs, a priori time window was initially set between 300 and 500 ms after stimuli [25]. This time window was further divided into 20 equal-sized time intervals in which amplitude averages were computed. Therefore, with three EEG channels, the dimensionality of feature vectors was reduced to 60. Finally, these feature vectors were scaled to zero mean and unit variance.

2.3. Classification

Fig. 2 depicts procedures used to extract features and split the data for classification.

Data splitting. Before classification, the data were randomly split into training (75%) and testing (25%) sets. Using the training set, 30 iterations of Monte-Carlo cross-validation (again 75:25 from the subset) were performed to optimize parameters. Results using the holdout testing set were computed in each cross-validation iteration and averaged at the end of the processing. No parameter decision was based on the holdout set.

LDA. State-of-the-art [3] LDA with eigenvalue decomposition used as the solver, and automatic shrinkage using the Ledoit–Wolf lemma [13] was applied.

SVM. The implementation was based on libsvm [4]. Both recommendations in the literature [8] and validation subsets were used to find the optimal parameters. Finally, penalty parameter C was set to 1, the kernel cache was 500 MB, and degree of the polynomial kernel function was set to 3. One-vs-rest decision function of shape with the RBF kernel type and shrinking heuristics were used.

CNN. Convolutional neural networks were implemented in Keras [5]. They were configured to maximize classification performance on the validation subsets. Its structure is depicted in Fig. 3. Initially, after empirical parameter tuning based on cross-validation, the parameters were selected as follows:

- The first convolutional layers had six 3 × 3 filters. The filter size was set to cover all three EEG channels. Both the second filter dimension and number of filters were tuned experimentally.
- In both cases, dropout was set to 0.5.
- The output of the convolutional layer was further downsampled by a factor of 8 using the average pooling layer.
- ELU activation function [6] was used for both convolutional and dense layers as recommended in related literature [23]. Compared to sigmoid function, ELU mitigates the vanishing gradient problem using the identity for positive values. Moreover, in contrast to rectified linear units (ReLU), ELUs have negative values which allow them to push mean unit activations closer to zero

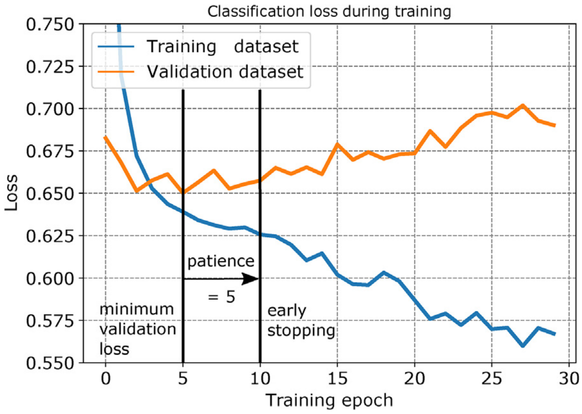

Fig. 4. Decrease of classification loss based on the baseline CNN architecture is shown. Although training loss kept declining throughout all 30 epochs, validation loss reached the minimum after only five epochs. Because the patience parameter was set to five, in this case, the training was stopped after 10 epochs. As seen from the growing difference between training and validation loss, further training would lead to substantial overtraining.

while ensuring a noise-robust deactivation state [6]. The parameter $\alpha > 0$ was set to 1:

$$f(x) = \begin{cases} x & \text{if} \quad x > 0 \\ \alpha(e^x - 1) & \text{if} \quad x \le 0 \end{cases}$$

- Batch size was set to 16.
- Cross-entropy was used as the loss function.
- Adam [11] optimizer was used for training because it is computationally efficient, has little memory requirements and is frequently used in the field [22].
- The number of training epochs was set to 30.
- Early stopping with the patience parameter of 5 was used.

3. Results

As mentioned above, cross-validation for hyperparameter estimation was followed by testing on a holdout set. Accuracy, precision, recall and AUC (area under the ROC curve) have been computed [10]. In the validation phase, the aim was to reach the configuration yielding the highest accuracy while ensuring it is not at the expense of precision and recall. In Fig. 4, an example of searching for an optimal configuration of CNN weights and biases based on the training and validation sets is shown.

3.1. Effect of parameter modifications on validation performance

Feature extraction for LDA and SVM. Parameter optimization of the classifiers themselves has been discussed above. Additionally, different feature extraction settings were compared regarding the average classification results achieved during cross-validation. Results of the comparisons are depicted in Table 1. Accuracy had an increasing trend when the time window got prolonged to 800 and 1000 ms. It can be speculated that the standard apriopri time window is not enough for capturing target to non-target differences when classifying children data that display a large variety in their P300 components. As expected, classification performance with WM features was slightly higher than for preprocessed epochs without feature extraction. Based on the results, both LDA and SVM configured as described above with the time window between 300 and 1000 ms were used in the testing phase.

CNN. The neural network architecture described above was used as the starting point. However, some parameter modifications were explored regarding their effect on the validation classification results. The results are shown in Table 2. Performance mostly displayed only small and insignificant changes with these parameter modifications. Consistently with [23], batch normalization led to slightly better accuracy. Moreover, the absence of batch normalization made the results less predictable and more fluctuating as can be seen in standard deviation of recall. Another clear decrease in performance was observed without dropout regularization. Finally, average pooling was better than max pooling for the validation data. Consequently, the initial configuration described in Section 2.3 was used for testing.

3.2. Testing results

Based on the results in Section 3.1, both feature extraction method for LDA and SVM, and CNN configuration achieving the best average accuracy during cross-validation were selected for the testing phase. Fig. 5 shows the achieved results. All tested models achieved comparable classification results. LDA had the highest classification recall (around 67%). Single trial classification accuracy stayed within the range between 62% and 64%.

Averaging of epochs associated with the same markers is a standard ERP technique for increasing signal-to-noise ratio [15]. When averaging, repeated ERPs including the P300 are amplified while continuous random EEG noise is suppressed. Because even in P300 BCIs, repeated stimulation is usually used to achieve good performance [17], it is worth exploring how once trained classifiers can generalize to averaged epochs. Therefore, consecutive groups of one to six neighboring epochs from the testing set were used instead of single trials. Fig. 6 depicts the results achieved. With averaging, classification accuracy increased from original 61–64% up to 76–79%. There were no significant differences among classifiers, although CNN displayed slightly higher standard deviations.

4. Discussion

Single trial classification accuracy was between 62% and 64% for all tested classification models without significant differences. Similar results have been commonly reported in the literature. For example, in [9], 65% single trial accuracy was achieved (using one to three EEG channels and personalized training data). In [24], 40–66% classification accuracy was reported, highly dependent on the tested subject. Comparably, this manuscript achieved similar performance for a large multi-subject dataset of school-age children.

On single trial level, CNN achieved comparable performance to both LDA and SVM. Similar performance was also achieved when applying averaged testing epochs. However, CNN seemed slightly less stable and more dependent on training/validation split as can be seen in standard deviations.

Consistently with related deep learning literature [23], a combination of ELUs, dropout and batched normalization were beneficial for classification performance. Unlike many image classification applications, average pooling was better than max pooling, perhaps because it is not associated with data loss. Even less prominent features may contribute to classifier discriminative abilities. To further verify how the CNN was able to classify between targets and non-targets, the network was exposed to all target, or all non-target patterns. Average hidden layer outputs (the 4th average pooling layer used as an example) across these conditions were calculated and shown in Fig. 7. There is a clear difference between some CNN outputs although the most remain stable across both conditions.

In our previous work [27], we applied stacked autoencoders (SAE) to the same GTN dataset. In contrast with the current work, manual feature extraction using discrete wavelet transform was

Table 1
Average cross-validation classification results based on the feature extraction method with the LDA classifier configured as described in Section 2.3. Averages from 30 repetitions and related sample standard deviations (in brackets) are reported. WM – windowed means (time intervals relative to stimuli onsets in square brackets).

Feature extraction	AUC	Accuracy	Precision	Recall
WM [300–500 ms]	59.56% (1.04)	59.54% (1.04)	59.48% (1.83)	61.69% (2.08)
WM [300–800 ms]	60.94% (1.04)	60.93% (1.05)	60.75% (1.9)	63.38% (1.85)
WM [300–1000 ms]	61.77% (0.9)	**61.76**% (0.91)	61.45% (1.9)	64.64% (1.48)
None	61.09% (1.13)	61.08% (1.13)	61.68% (1.67)	59.90% (1.35)

The bold values denote the configuration that yielded the highest accuracy.

Table 2
Average cross-validation classification results based on the CNN parameter settings. Averages from 30 repetitions and related sample standard deviations (in brackets) are reported. CNN configuration described in Section 2.3 was used as the baseline model.

Changed parameter	AUC	Accuracy	Precision	Recall
None	66.12% (0.68)	**62.18**% (0.94)	62.76% (1.95)	61.34% (2.63)
RELUs instead of ELUs	66.36% (0.62)	61.85% (1.15)	62.7% (2.19)	60.1% (3.04)
Filter size (3, 30)	65.84% (0.49)	61.95% (1.18)	62.7% (2.1)	60.5% (3.91)
12 conv. filters	66.31% (0.51)	61.83% (1.1)	62.3% (2.21)	61.6% (3.08)
No batch normalization	65.99% (0.77)	60.55% (1.52)	61.02% (3.16)	61.5% (7.21)
Dropout 0.2	67.67% (0.65)	60.8% (1.49)	61.33% (2.31)	60.33% (4.0)
No dropout	68.63% (1.11)	59.49% (1.2)	59.61% (1.93)	60.7% (4.44)
Dense (150)	66.07% (0.8)	61.81% (0.95)	62.33% (1.83)	61.18% (2.49)
Two dense l. (120-60)	65.72% (0.77)	62.11% (0.9)	63.14% (2.03)	59.5% (2.55)
Max- instead of AvgPool	64.23% (1.15)	58.94% (1.94)	60.22% (4.18)	59.24% (13.76)

The bold values denote the configuration that yielded the highest accuracy.

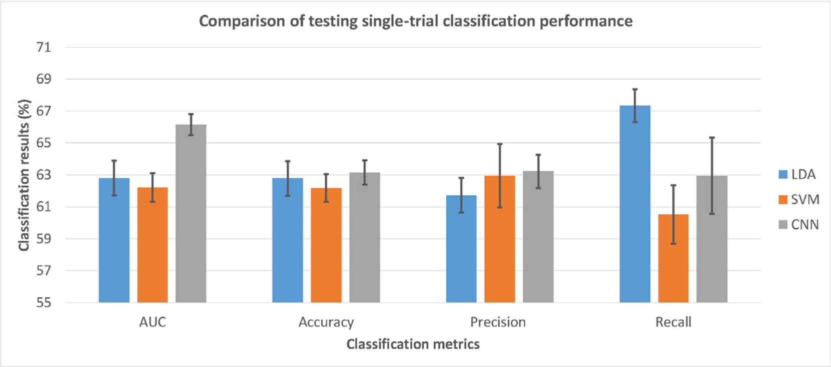

Fig. 5. Testing results for single trial classification (error bars show standard deviations).

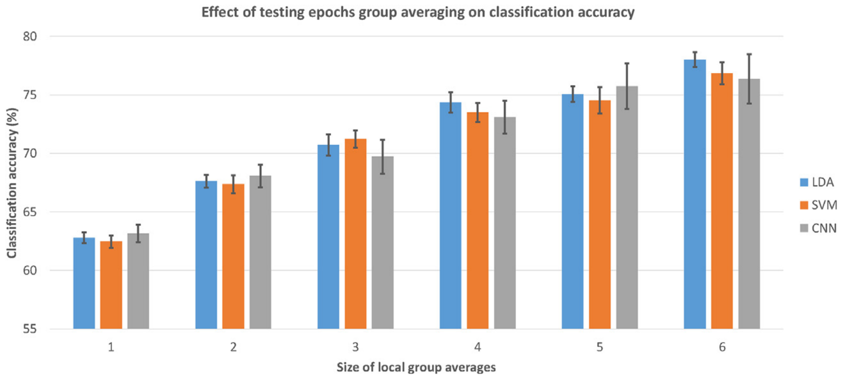

Fig. 6. Testing results when averaging neighboring epochs (error bars show standard deviations).

performed. Instead of single trial classification, success rate of detecting the number thought based on multiple single trial classification results was computed. Maximum success rate on the testing dataset was 79.4% for SAE, 75.6% for LDA and 73.7% for SVM. It seems that while SAE combined with traditional feature engineering and involving multiple trials per marker can outperform linear classifiers, the same benefits cannot be repeated when applying CNN to single trial classification of raw EEG data.

Computational efficiency is another important factor to consider when applying the methods in online BCI systems. Experimental comparison was performed with Intel Core i7-7700K, four cores, 4.2 GHz, 64 GB RAM and NVIDIA GeForce GTX 1050 Ti GPU. CNN took 46 s to train on CPU and 26 s to train on GPU. Both LDA and SVM were much faster to train, with 300 and 1600 ms, respectively. However, training times were not critical in the presented experiment since any universal classifier needs to be trained just once and

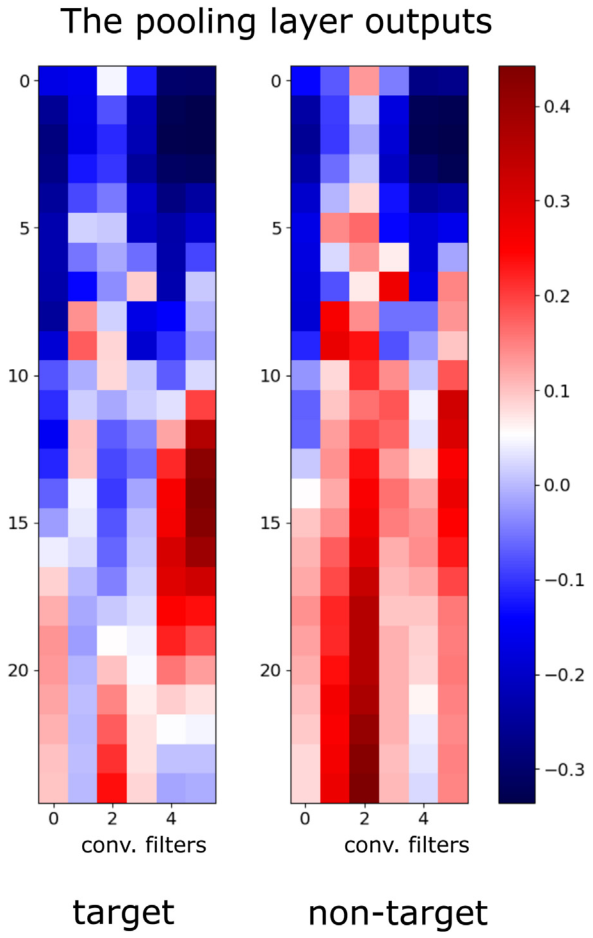

Fig. 7. Average outputs of the 4th (pooling) layer are depicted after the CNN was exposed to all target/non-target patterns. *X*-axis corresponds to indices of convolutional filters (six in total). *Y*-axis is the output of convolution originally corresponding to time information, after average pooling further downsampled by a factor of 6. There is a clear difference in outputs, mainly in the bottom part of the maps. However, many outputs seem independent of classification labels, poorly contributing to CNN discrimination abilities.

not with every new BCI user. Testing times were calculated relative to one processed feature vector and were low enough for all classifiers (CNN took 0.3 ms to classify one pattern on CPU and 0.1 ms on GPU, LDA took 0.1 and SVM 0.2 ms). It can be concluded that all tested algorithms can be used in online BCIs. Neural networks are slower to train and this could be a problem for personalized BCIs, retrained with each new user.

There are several limitations of the reported experiments. As a noise suppression procedure, severely damaged epochs (with amplitude exceeding ±100 μV when compared to baseline) were rejected before further processing. While epoch rejection is beneficial for classification accuracy, on the other hand, it would also lead to lower bit-rates when used in on-line P300 BCI systems because of data loss. Artifact correction methods based on Independent Component Analysis were not feasible because of the low number of EEG channels (three). Moreover, the low number of EEG channels could have a detrimental effect on classification performance because of limited spatial information provided on the input. Another possible limitation was that there might be an architecture of CNN that would lead to better classification performance and had not been discovered by the author. However, several manipulations of CNN parameters were tested using cross-validation, including adding a new dense layer, with only very modest changes in validation classification accuracy.

Recent review of EEG and DNNs [22] studies reported the median gain in accuracy of DNNs over traditional baselines to be 5.4%. It also revealed significant challenges in the field. Low number of training examples is a common complaint especially for event-related data that contain the relevant information in time domain. In this case, only a small fraction of continuous EEG measurement near the onset of trials can be used and strategies such as overlapping time windows to obtain more examples in frequency domain are not feasible. In the current study, 11,532 epochs were used which is below mean number of examples (251,532) and medium number of examples (14,000) in the reviewed papers [22]. Strategies such as data augmentation can be considered to increase the number of training examples to be sufficient for DNNs. Moreover, half of the studies [22] used between 8 and 62 EEG channels. Adding more channels to Fz, Cz and Pz could increase spatial resolution and accuracy but would also increase preparation time and the participant's discomfort. In future work, more on the effect of number of EEG channels on the P300 classification accuracy can be investigated. Furthermore, soft or hard thresholding based on discrete wavelet transform can be considered for noise cancellation [1]. Another line of research would be to propose different deep learning models for the same classification task, with extensive parameter grid search, or genetic algorithms. Based on the recent review of the field [29], frequently cited and promising networks include Recurrent Neural Networks, especially Long short-term memory (LSTM). Moreover, a CNN layer to capture spatial patterns can be followed by a LSTM layer for temporal feature extraction [29].

5. Conclusion

The aims of the presented experiments were to compare CNN with baseline classifiers (LDA, SVM) using a large multi-subject P300 dataset. CNN was applied to raw ERP epochs (with the dimensionality of 3 × 1200). Baseline classifiers were applied to windowed means features (with the dimensionality of 60). Empirical parameter optimization was performed using cross-validation and classifiers were tested on a holdout set. Various CNN parameters are discussed. Single trial classification accuracy was between 62% and 64% for all tested models with CNN able to match but not outperform its competitors. When the trained models were applied to averaged trials in the testing phase, accuracy increased up to 76–79%. Achieved accuracy is comparable with state-of-the-art despite using a multi-subject dataset from 250 children. Potential explanation of the results are discussed. Based on the results, LDA and SVM with state-of-the-art feature extraction still seem to be a good choice for P300 classification, especially with relatively small training datasets. CNN might need more spatial information in the data (by means of more channels) to better understand the patterns. Alternatively, the dataset was not large enough for CNN to prove its benefits and, e.g. data augmentation techniques could help to overcome this obstacle. Both the preprocessed data [20] and Python codes [26] are available to ensure reproducibility of the experiments.

Authors' contribution

LV designed and performed the machine learning workflow. LV wrote the manuscript.

Acknowledgement

This publication was supported by the project LO1506 of the Czech Ministry of Education, Youth and Sports under the program NPU I.

Conflict of interest

None declared.

References

[1] M. Ahmadi, R. Quian Quiroga, Automatic denoising of single-trial evoked potentials, NeuroImage 66 (2013 Feb) 672–680.

[2] Y. Bengio, A.C. Courville, P. Vincent, Unsupervised Feature Learning and Deep Learning: A Review and New Perspectives, 2012 arXiv:1206.5538.

[3] B. Blankertz, S. Lemm, M. Treder, S. Haufe, K. Muller, Single-trial analysis and classification of ERP components - a tutorial, NeuroImage 56 (2) (2011) 814–825, http://dx.doi.org/10.1016/j.neuroimage.2010.06.048.

[4] C.C. Chang, C.J. Lin, LIBSVM: a library for support vector machines, ACM Trans. Intell. Syst. Technol. 2 (2011), 27:1–27:27, Software Available at http://www.csie.ntu.edu.tw/cjlin/libsvm.

[5] F. Chollet, et al., Keras, 2015 https://keras.io.

[6] D.A. Clevert, T. Unterthiner, S. Hochreiter, Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs), 2016 arXiv:1511.07289.

[7] L. Deng, D. Yu, Deep learning: Methods and applications, Found. Trends(r) Signal Process. 7 (3-4) (2014) 197–387, http://dx.doi.org/10.1561/2000000039.

[8] R.E. Fan, K.W. Chang, C.J. Hsieh, X.R. Wang, C.J. Lin, Liblinear: a library for large linear classification, J. Mach. Learn. Res. 9 (2008 Jun) 1871–1874 http://dl.acm.org/citation.cfm?id=1390681.1442794.

[9] N. Haghighatpanah, R. Amirfattahi, V. Abootalebi, B. Nazari, A single channel-single trial P300 detection algorithm., 2013 21st Iranian Conference on Electrical Engineering (ICEE) (2013) 1–5, http://dx.doi.org/10.1109/IranianCEE.2013.6599576, May.

[10] M. Hossin, M. Sulaiman, A review on evaluation metrics for data classification evaluations, Int. J. Data Mining Knowl. Manag. Process 5 (2) (2015) 1.

[11] D.P. Kingma, J. Ba, Adam: A Method for Stochastic Optimization, 2014, Comment: Published as a Conference Paper at the 3rd International Conference for Learning Representations, San Diego, 2015 arxiv:1412.6980.

[12] D.J. Krusienski, E.W. Sellers, F. Cabestaing, S. Bayoudh, D.J. McFarland, T.M. Vaughan, J.R. Wolpaw, A comparison of classification techniques for the P300 speller, J. Neural Eng. 3 (4) (2006) 299–305, http://dx.doi.org/10.1088/1741-2560/3/4/007.

[13] O. Ledoit, M. Wolf, Honey, i shrunk the sample covariance matrix, J. Portf. Manag. 30 (4) (2004) 110–119, http://dx.doi.org/10.3905/jpm.2004.110.

[14] F. Lotte, L. Bougrain, A. Cichocki, M. Clerc, M. Congedo, A. Rakotomamonjy, F. Yger, A review of classification algorithms for EEG-based brain-computer interfaces: a 10 year update, J. Neural Eng. 15 (3) (2018) 031005, http://dx.doi.org/10.1088/1741-2552/aab2f2.

[15] S.J. Luck, An Introduction to the Event-Related Potential Technique, MIT Press, Cambridge, MA, 2005.

[16] N.V. Manyakov, N. Chumerin, A. Combaz, M.M. Van Hulle, Comparison of classification methods for P300 brain-computer interface on disabled subjects, Intell. Neurosci. (2011), http://dx.doi.org/10.1155/2011/519868, 2:1–2:12.

[17] D.J. McFarland, W.A. Sarnacki, G. Townsend, T. Vaughan, J.R. Wolpaw, The P300-based brain-computer interface (BCI): effects of stimulus rate, Clin. Neurophysiol. 122 (4) (2011) 731–737.

[18] D.J. McFarland, J.R. Wolpaw, Brain-computer interfaces for communication and control, Commun. ACM 54 (5) (2011) 60–66, http://dx.doi.org/10.1145/1941487.1941506, May.

[19] R. Mouček, L. Vařeka, T. Prokop, J. Štěbeták, P. Brůha, Event-related potential data from a guess the number brain-computer interface experiment on school children, Sci. Data 4 (2017).

[20] R. Mouček, L. Vařeka, T. Prokop, J. Štěbeták, P. Brůha, Replication Data for: Evaluation of Convolutional Neural Networks Using A Large Multi-Subject P300 Dataset, 2019, http://dx.doi.org/10.7910/DVN/G9RRLN.

[21] A. Pinegger, G. Müller-Putz, No training, same performance!? – a generic P300 classifier approach, Proceedings of the 7th International BCI Conference Graz 2017 (2017), http://dx.doi.org/10.3217/978-3-85125-533-1-77, 9.

[22] Y. Roy, H.J. Banville, I. Albuquerque, A. Gramfort, T.H. Falk, J. Faubert, Deep Learning-Based Electroencephalography Analysis: A Systematic Review, 2019 arXiv:1901.05498.

[23] R.T. Schirrmeister, J.T. Springenberg, L.D.J. Fiederer, M. Glasstetter, K. Eggensperger, M. Tangermann, F. Hutter, W. Burgard, T. Ball, Deep Learning With Convolutional Neural Networks for Brain Mapping and Decoding of Movement-Related Information from the Human EEG, 2017 arXiv:1703.05051.

[24] N. Sharma, Single-Trial P300 Classification Using PCA With LDA, QDA and Neural Networks, 2017 arXiv:1712.01977.

[25] D.S. Tan, A. Nijholt, Brain-Computer Interfaces: Applying Our Minds to Human-Computer Interaction, 1st ed., Springer Publishing Company, Incorporated, 2010.

[26] L. Vařeka, CNNforGTN, 2019 https://bitbucket.org/lvareka/cnnforgtn/src/master/.

[27] L. Vařeka, T. Prokop, P. Mautner, R. Mouček, J. Štěbeták, Application of stacked autoencoders to P300 experimental data, Proceedings of the 16th International Conference on Artificial Intelligence and Soft Computing, ICAISC 2017 (2017).

[28] M.D. Zeiler, R. Fergus, Visualizing and understanding convolutional networks, 2013 arXiv:1311.2901.

[29] X. Zhang, L. Yao, X. Wang, J. Monaghan, D. McAlpine, A Survey on Deep Learning Based Brain Computer Interface: Recent Advances and New Frontiers, 2019 arXiv:1905.04149.