2021 43rd Annual International Conference of the
IEEE Engineering in Medicine & Biology Society (EMBC)
Oct 31 - Nov 4, 2021. Virtual Conference

Performance Evaluation of Compressed Deep CNN for Motor Imagery Classification using EEG

Vishnupriya R, Neethu Robinson, Ramasubba Reddy M, and Cuntai Guan

Abstract— **Recently, deep learning and convolutional neural networks (CNNs) have reported several promising results in the classification of Motor Imagery (MI) using Electroencephalography (EEG). With the gaining popularity of CNN-based BCI, the challenges in deploying it in a real-world mobile and embedded device with limited computational and memory resources need to be explored. Towards this objective, we investigate the impact of the magnitude-based weight pruning technique to reduce the number of parameters of the pre-trained CNN-based classifier while maintaining its performance. We evaluated the proposed method on an open-source Korea University dataset which consists of 54 healthy subjects' EEG, recorded while performing right-and left-hand MI. Experimental results demonstrate that the subject-independent model can be maximumly pruned to 90% sparsity, with a compression ratio of 4.77× while retaining classification accuracy at 84.44% with minimal loss of 0.02% when compared to the baseline model's performance. Therefore, the proposed method can be used to design more compact deep CNN- based BCIs without compromising on their performance.**

I. Introduction

In recent years, deep learning (DL) has significantly elevated the classification performance of Brain-Computer Interface (BCI) systems [1]. By operating directly on raw EEG signals to learn distinguishable feature representations, deep learning avoids time-consuming pre-processing and feature engineering steps [2]. In BCI studies, out of many DL models, convolutional neural networks (CNN) are the most used model [3][4]. To achieve high classification accuracy, CNN models typically need an enormous number of computations and thus high-performance servers are essential [5]. It is difficult to deploy CNN on end devices with limited resources, such as mobile phones or embedded devices. These issues must be resolved in order to use mobile-based BCI systems or BCI systems outside of the laboratory [6].

Recently, many DL methods for EEG-based BCI systems have been proposed. Lawhern *et* al. introduced EEGNet, a compact network that utilizes depth-wise and separable convolutions to build an EEG-specific model integrating well-known EEG feature extraction techniques [7]. Schirrmeister *et* al. proposed a deep CNN model that has four convolutional-max-pooling blocks, out of which the first one is built specially to handle EEG input signal, followed by three convolutional-max-pooling blocks and a fully connected layer [8]. However, models based on CNN are complex with many trainable parameters. Furthermore, recent BCI studies have shown the benefit of fine-tuning a pre-trained model for better performance in individual subjects [9]. Reducing the number of parameters in the network may facilitate faster model fine-tuning.

One of the most common optimization algorithms used for reducing network complexity is pruning. It is possible to delete many redundant weights from a trained network with a marginal loss in accuracy by pruning. This results in a more compressed DL model. Pruning methods are divided into two categories: weight pruning and filter pruning. Weight pruning removes redundant weights from the weight tensor, whereas filter pruning removes redundant convolutional filters [10][11].

More recently, by applying group sparsity regularization to the loss function, Lebedev *et* al. [12] discovered that certain whole groups of weights can be reduced to zero and excluded. Structured Sparsity Learning (SSL) was proposed by Wen *et* al. [13] to regularize the structure of deep neural networks (DNN). The structured compactness of DNN allows us to accomplish higher speedups for the DNN evaluation. Hu *et* al. [14] proposed an algorithm based on the concept that in a large network, the output of many neurons is often zero. It is reasonable to assume that these zero activations are redundant and can be removed. ThiNet was proposed by Luo *et* al. [15] which greedily prunes the convolutional filter that has the least impact on the next layer's output. Chin *et* al. [16] introduced a layer compensated pruning algorithm, which improves performance across a range of heuristic metrics.

In BCI so far, pruning has been used for choosing the most relevant features/electrode channels for the P300 based BCI [17]. The accuracy was 87% when using the best 8 relevant electrodes and 87.5% when using the 8 most salient and fixed electrodes: F_Z, C_Z, P_Z, P_3, P_4, PO_7, PO_8, and O_Z. Arvaneh *et* al. proposed a decision tree-based approach for EEG channel selection. Irrelevant channels are eliminated using the decision tree. Following that, a pruning process was used to rank the remaining channels [18]. In this study, EEG signals were recorded using 22 electrodes per subject. The proposed method reduces the average number of electrodes from 22 to 8.44, whereas 3.63% of classification accuracy is reduced. It is worth noting that the impact of pruning of deep CNN used in a subject-independent MI BCI [19] has yet to be investigated.

The aim of this paper is to reduce the complexity of the subject-independent MI based BCI model without reducing

Vishnupriya R, Ramasubba Reddy M are with the Indian Institute of Technology Madras, Chennai 600036, India (e-mail: vishnupriyaece94@gmail.com, rsreddy@iitm.ac.in).

Neethu Robinson, Cuntai Guan are with the Nanyang Technological University, 50 Nanyang Avenue, Singapore (e-mail: nrobinson@ntu.edu.sg, ctguan@ntu.edu.sg).

Corresponding author: Cuntai Guan

the performance using a magnitude-based weight pruning algorithm. In order to achieve this purpose, we train end-to-end subject-independent models on the MI dataset. A deep CNN model [8] is used in this study, which has reported the highest number of trainable parameters among the state-of-the-art methods in CNN-BCIs [9]. Then we perform pruning for different sparsity levels on the pre-trained models to reduce their complexity without affecting the performance of the model. The results suggest that even after pruning the model to 90% sparsity, the model compressed 4.77× with minimal loss in the performance when compared to the baseline model.

The paper is organized as follows. Section II describes the dataset. The proposed methodology is presented in section III. Following that, the results and discussion are reported in section IV. Finally, in section V, the conclusion and future work are discussed.

II. DATA

The proposed method is evaluated on the motor imagery (MI) dataset reported in Lee *et* al [20]. This dataset consists of 54 subject's EEG while performing two-class MI tasks (left-and right-hand imagined movement). EEG signals were recorded with 62 Ag/AgCl electrodes and at a sampling rate of 1000 Hz. Each subject underwent two data recording sessions on different days, with training and test phases in each session. Each phase had 100 trials per class, totaling 400 trials. The experiment started with a 3s resting time to prepare subjects for performing the MI task. Then the subject performed the corresponding MI task for 4s by following the visual cue. The screen remained blank for 6s after completing each task. For this study, all 62 channels are used and for each trial, 0 to 4 second MI tasks are segmented from the continuous EEG signals and further downsampled to 250 Hz.

III. METHODOLOGY

This section describes the EEG signal representation, the baseline model's architecture, training strategy, and the proposed methodology for applying magnitude-based weight pruning on subject-independent MI based BCI models.

A. EEG Representation

Each subject i has only one EEG dataset and it is divided into labeled trials. A single trial for a subject i is denoted as (X^j, y^j),where the pre-processed signal is represented by the input matrix $X^j \in \mathbb{R}^{N_e \times N_t}$,where N_e is the number of EEG electrodes and N_t is the discretized time samples for a trial and y^jstands for the corresponding class label for trial j. Based on the performance of the imagined or executed hand movement during the MI paradigm experiment class label $y^j \in L = \{0: \text{" Right hand "}, 1: \text{"Left hand"}\}$.

B. Network Architecture

We used the deep CNN model proposed by Schirrmeister *et* al. as our baseline, which is denoted as d. The model d is trained on the input trials so that it can correctly classify unlabeled trials using the output of the classifier $d: \mathbb{R}^{N_e \cdot N_t} \rightarrow L$. The deep CNN architecture consists of four convolution-max-pooling blocks followed by a fully connected *softmax* classification layer. In particular, the first block alone contains temporal and spatial filters to handle EEG signals. Batch normalization and dropout are added for each convolutional-max-pooling block.

C. Training Strategy

Each subject-independent model is trained with training and validation data. Adam [21] is used as the optimizer. The model training is carried out with early stopping criteria, in which validation set accuracy is monitored. The model is trained for a maximum of 200 epochs and the epoch with the highest validation accuracy is selected. The resulting best model is further retrained with the entire training and validation data. The model with maximum validation accuracy is saved as W.

D. Proposed methodology: pruning pre-trained deep CNN model

Magnitude-based weight pruning [22][23] is performed on the pre-trained models to remove redundant values in tensor weights W. We use the TensorFlow framework to prune the network's connections. Pruning is performed on the entire model. A binary mask variable that is of the same size and shape as the weight tensor is introduced. The binary mask variable determines which weights participate in the graph's forward execution. The model weights are sorted by their absolute values and the smallest magnitude weights are masked to zero until a desired sparsity level X % is reached. Sparsity implies that X % of the tensor weight will be lost. The range of the sparsity (k) varies from 10 to 90% in steps of 10%. The pruned model is retrained for a maximum of 200 epochs to regain the lost performance. The number of iterations is decided by an early stopping method based on validation data accuracy. During retraining of the pruned model, the sparse structure is maintained, and the remaining weights are trained to produce the final sparse model weight $\widetilde{W}$. Both W and $\widetilde{W}$ are then evaluated on test data from the test subject. The block diagram for the proposed method is shown in Fig.1. A file compression algorithm zip is applied to the baseline model weights W and the pruned model weights $\widetilde{W_k}$. This helps to reduce the size of the sparse pruned model, whereas the baseline model remains the same.

E. Experiment

We use the leave-one-subject-out (LOSO) method to evaluate subject-independent classification. The model is trained with all data except the target subject. Based on the previous studies, data from all 53 subjects are split randomly into 85% training and 15% validation data, and the subject-independent model is trained [9][24]. The model with the highest validation accuracy is saved and pruned for different sparsity levels (10%, 20 % till 90%). The pruned models are evaluated on the last 100 trials (session 2 of Day 2) of the target subject's data. The model training was carried out on an NVIDIA Tesla V100 GPU with 32 GB GPU of memory. To evaluate the impact of pruning, we have calculated compression ratio, number of network parameters, and classification accuracy as metrics for each k. Compression ratio is defined as the original model size W divided by the compressed model size W_k [25].

IV. RESULTS AND DISCUSSION

In this section, the results of the pruned model are compared

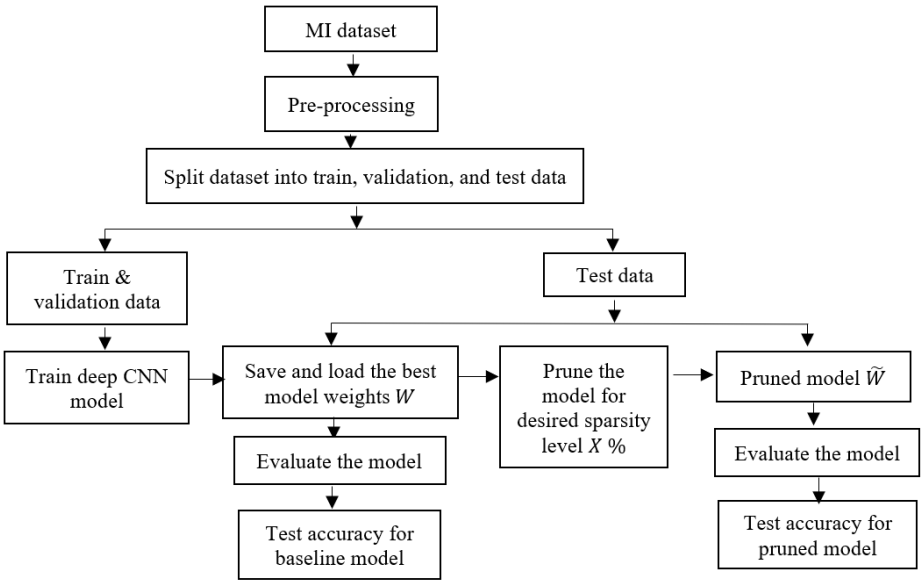

Figure 1. Flow diagram for the proposed method

with the baseline model. The metrics such as the average classification accuracy for all subjects, compression ratio, and the number of network parameters for the pruned model at different sparsity levels are reported.

A. Classification Accuracy

The subject-independent baseline model has an average classification accuracy of 84.46% (±11.39%). The baseline accuracy is in line with the reported results in the literature [9]. The classification accuracy of the pruned model varies with different sparsity levels, as shown in Fig.2. The accuracy is affected by the number of redundant values removed from the weight tensor. When the model is pruned to lower sparsity levels, the model's redundancy is high, which affects the classification performance of the model. For example, at k = 10%, (statistically insignificant, p=0.143) 0.86% drop in performance is observed. The maximum classification accuracy of 85.53% is achieved at 70% sparsity. When the model is pruned more than 70% sparsity, important weights of the model are discarded. As a result, the classification accuracy drops to 84.44%. Even after pruning 90% of the weights, the classification accuracy is still 84.44% (±11.39%) which remains close to the baseline model's accuracy of 84.46%. Since the difference in the performance is low (statistically insignificant, p=0.976), which again helps with the research goal to compress the model without impacting the performance.

B. Compression Ratio

Fig.3 depicts the compression ratio trend for different sparsity levels. The model is compressed by removing the redundant weights based on the sparsity level. When the sparsity is higher than 80%, our proposed method produces a maximum compression ratio of 4. 77×. This proves that the efficient deep CNN model can be reduced to 4.77× without affecting the model performance.

C. Number of Parameters

The baseline deep CNN model has 305K parameters and a model size of ~703KBs. When the model is pruned for maximum sparsity of 90% with a compression ratio of 4.77×, the parameters of the model are reduced to 63K with only a 0.02% (p=0.976) loss in the model performance. This demonstrates that the pruned model performs similarly to the baseline model, which has a significantly larger number of parameters. The performance metrics for the baseline model and the pruned models are shown in Table I and Table II, respectively.

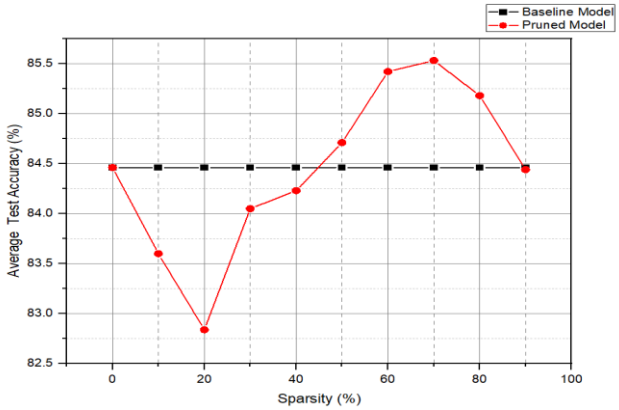

Figure 2. Average classification accuracy (N=54) for different sparsity levels and the baseline model

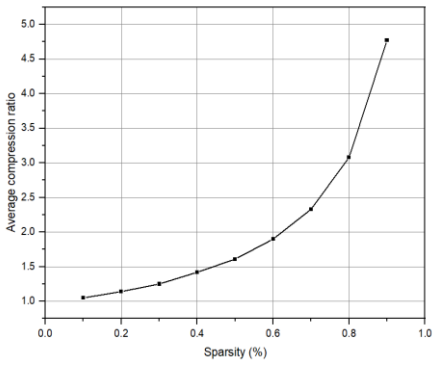

Figure 3. Sparsity vs Average compression ratio

TABLE I. RESULTS FOR THE BASELINE MODEL

Model Size (bytes)	Parameters	Average classification accuracy ± SD
720687	305K	84.46 ± 11.39%

TABLE II. RESULTS FOR THE PRUNED MODEL

Sparsity level	Model Size (bytes)	Average compression Ratio	Parameters	Average classification accuracy ± SD (%)
10 %	681396	1.05 ×	290K	83.6 ± 11.36
20 %	628807	1.14 ×	267K	82.84 ± 13.11
30 %	573475	1.25 ×	244K	84.05 ± 12.24
40 %	507381	1.42 ×	214K	84.23 ± 12.21
50 %	446941	1.61 ×	189K	84.71 ± 11.39
60 %	377444	1.9 ×	160K	85.42 ± 11.23
70 %	308858	2.33 ×	130K	85.53 ± 11.33
80 %	233314	3.08 ×	99K	85.18 ± 11.74
90 %	150968	4.77 ×	63K	84.44 ± 11.39

D. Discussion

In this paper, we investigated the effectiveness of magnitude-based weight pruning on the subject-independent deep CNN model. The proposed method reduces the number of parameters, which in turn reduces the model size while maintaining the classification accuracy. The results suggest that the baseline subject-independent models are likely to be severely over-parameterized. Throughout the study, we examined the effectiveness of pruning as a method of model compression. In the future, frameworks that support sparse computations will be evaluated to investigate potential advantages in computational speed, power, etc. Interpreting the network representation before and after pruning is added to learn the most relevant EEG parameters.

V. CONCLUSION

In this study, we evaluated the impact of pruning on subject-independent MI based BCI models. The state-of-the-art deep CNN model with a large number of parameters reported a classification accuracy of 84.46% in subject-independent MI classification. The deep CNN models have many parameters and large model size. As a result, such complex models consume a significant amount of storage and computational resources. To address this limitation, the proposed method compresses the model by 4.77× for 90% sparsity while maintaining the classification accuracy at 84.44%. The results suggest that the complexity of the subject-independent MI based model can be reduced while retaining the model performance and thus it is easier to deploy the model into resource-constrained end devices. Thus, the proposed method can be used to create more compact deep CNN-based BCIs while maintaining the performance of the model.

ACKNOWLEDGMENT

This work was partially supported by the RIE2020 AME Programmatic Fund, Singapore (No. A20G8b0102). We would like to thank Ding Yi for the assistance in reviewing the codes.

REFERENCES

[1] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015, doi: 10.1038/nature14539.

[2] X. Zhang, L. Yao, X. Wang, J. J. M. Monaghan, D. Mcalpine, and Y. Zhang, "A survey on deep learning-based non-invasive brain signals: recent advances and new frontiers," *J. Neural Eng.*, 2020, doi: 10.1088/1741-2552/abc902.

[3] A. Craik, Y. He, and J. L. Contreras-Vidal, "Deep learning for electroencephalogram (EEG) classification tasks: A review," *J. Neural Eng.*, vol. 16, no. 3, 2019, doi: 10.1088/1741-2552/ab0ab5.

[4] S. Sakhavi, C. Guan, and S. Yan, "Learning Temporal Information for Brain-Computer Interface Using Convolutional Neural Networks," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 29, no. 11, pp. 5619–5629, 2018, doi: 10.1109/TNNLS.2018.2789927.

[5] S. Moon, Y. Byun, J. Park, S. Lee, and Y. Lee, "Memory-Reduced Network Stacking for Edge-Level CNN Architecture With Structured Weight Pruning," *IEEE J. Emerg. Sel. Top. Circuits Syst.*, vol. 9, no. 4, pp. 735–746, 2019, doi: 10.1109/JETCAS.2019.2952137.

[6] Y.T. Wang, Y. Wang, and T.P. Jung, "A Cell-phone based Brain Computer Interface for Communication in Daily Life," *J. Neural Eng.*, pp. 233–240, 2011, [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.704.8338.

[7] V. J. Lawhern, A. J. Solon, N. R. Waytowich, S. M. Gordon, C. P. Hung, and B. J. Lance, "EEGNet: A compact convolutional neural network for EEG-based brain-computer interfaces," *J. Neural Eng.*, vol. 15, no. 5, pp. 1–30, 2018, doi: 10.1088/1741-2552/aace8c.

[8] R. T. Schirrmeister, J.T. Springenberg, L.D.J. Fiederer, M. Glasstetter, K. Eggensperger, M.Tangermann, F. Hutter, W. Burgard and T.Ball, "Deep learning with convolutional neural networks for EEG decoding and visualization," *Hum. Brain Mapp.*, vol. 38, no. 11, pp. 5391–5420, 2017, doi: 10.1002/hbm.23730.

[9] K. Zhang, N. Robinson, S. W. Lee, and C. Guan, "Adaptive transfer learning for EEG motor imagery classification with deep Convolutional Neural Network," *Neural Networks*, vol. 136, pp. 1–10, 2021, doi: 10.1016/j.neunet.2020.12.013.

[10] H. Li, A. Kadav, I. Durdanovic, H. Samet, H.P. Graf, "Pruning filters for efficient ConvNets," no. 2016, pp. 1–13, 2017.

[11] Z. Liu, J. Li, Z. Shen, G. Huang, S. Yan, and C. Zhang, "Learning Efficient Convolutional Networks through Network Slimming," *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2017-Octob, pp. 2755–2763, 2017, doi: 10.1109/ICCV.2017.298.

[12] V. Lebedev and V. Lempitsky, "Fast ConvNets Using Group-Wise Brain Damage," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-Decem, pp. 2554–2564, 2016, doi: 10.1109/CVPR.2016.280.

[13] W.Wen, C.Wu,Y. Wang, Y. Chen, H.Li ,"Learning Structured Sparsity in Deep Neural Networks," no. Nips, 2016.

[14] H. Hu, R. Peng, Y.-W. Tai, and C.-K. Tang, "Network Trimming: A Data-Driven Neuron Pruning Approach towards Efficient Deep Architectures," 2016, [Online]. Available: http://arxiv.org/abs/1607.03250.

[15] J. H. Luo, H. Zhang, H. Y. Zhou, C. W. Xie, J. Wu, and W. Lin, "ThiNet: Pruning CNN Filters for a Thinner Net," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 10, pp. 2525–2538, 2019, doi: 10.1109/TPAMI.2018.2858232.

[16] T. W. Chin, C. Zhang, and D. Marculescu, "Layer-compensated pruning for resource-constrained convolutional neural networks," *arXiv*, 2018.

[17] H. Cecotti and A. Gräser, "Neural network pruning for feature selection application to a P300 brain-computer interface," *ESANN 2009 Proceedings, 17th Eur. Symp. Artif. Neural Networks - Adv. Comput. Intell. Learn.*, no. January, pp. 473–478, 2009.

[18] M. Arvaneh, C. Guan, K. K. Ang, and H. C. Quek, "EEG channel selection using decision tree in brain-computer interface," *APSIPA ASC 2010 - Asia-Pacific Signal Inf. Process. Assoc. Annu. Summit Conf.*, no. December, pp. 225–230, 2010.

[19] O. Y. Kwon, M. H. Lee, C. Guan, and S. W. Lee, "Subject-Independent Brain-Computer Interfaces Based on Deep Convolutional Neural Networks," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 31, no. 10, pp. 3839–3852, 2020, doi:

10.1109/TNNLS.2019.2946869.
[20] M.H. Lee, O.Y. Kwon, Y.J. Kim, H.K. Kim, Y.E. Lee, J. williamson, S. Fazli and S.W. Lee, "EEG dataset and OpenBMI toolbox for three BCI paradigms: An investigation into BCI illiteracy," *Gigascience*, vol. 8, no. 5, pp. 1–16, 2019, doi: 10.1093/gigascience/giz002.
[21] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, pp. 1–15, 2015.
[22] Z. Liu, M. Sun, T. Zhou, G. Huang, and T. Darrell, "Rethinking the value of network pruning," pp. 1–21, 2019.
[23] M. H. Zhu and S. Gupta, "To prune, or not to prune: Exploring the efficacy of pruning for model compression," *arXiv*, 2017.
[24] N. Robinson, S.W. Lee, C.Guan, "EEG representation in deep convolutional neural networks for classification of motor imagery," *Neural Networks*, vol. 136, pp. 1322–1326, 2021, doi: 10.1109/SMC.2019.8914184.
[25] D. Blalock, J. J. G. Ortiz, J. Frankle, and J. Guttag, "What is the state of neural network pruning?," *arXiv*, 2020.