Demonstrating the Viability of Mapping Deep Learning Based EEG Decoders to Spiking Networks on Low-powered Neuromorphic Chips

Conference Paper · November 2021

DOI: 10.1109/EMBC46164.2021.9629621

6 authors, including:

Nicolas Berberich
Technische Universität München

Stefan K. Ehrlich
Harvard Medical School

John Nassour
Technische Universität München

Gordon Cheng
Technische Universität München

Demonstrating the Viability of Mapping Deep Learning Based EEG Decoders to Spiking Networks on Low-powered Neuromorphic Chips

Matthijs Pals[1], Rafael J. Pérez Belizón[1], Nicolas Berberich[2], Stefan K. Ehrlich, John Nassour, Gordon Cheng

***Abstract*— Accurate and low-power decoding of brain signals such as electroencephalography (EEG) is key to constructing brain-computer interface (BCI) based wearable devices. While deep learning approaches have progressed substantially in terms of decoding accuracy, their power consumption is relatively high for mobile applications. Neuromorphic hardware arises as a promising solution to tackle this problem since it can run massive spiking neural networks with energy consumption orders of magnitude lower than traditional hardware. Herein, we show the viability of directly mapping a continuous-valued convolutional neural network for motor imagery EEG classification to a spiking neural network. The converted network, able to run on the SpiNNaker neuromorphic chip, only shows a 1.91% decrease in accuracy after conversion. Thus, we take full advantage of the benefits of both deep learning accuracies and low-power neuro-inspired hardware, properties that are key for the development of wearable BCI devices.**

I. INTRODUCTION

Brain-computer interfaces (BCIs) are devices that record and classify users' neural activity for communication and control without requiring muscle movement. Thus, they offer high potential as assistive technology for disabled individuals, e.g., in the form of BCI-controlled exoskeletons and other forms of BCI-based wearable devices [1]–[3].

Constructing BCIs can be mediated in a non-invasive manner by using electroencephalography (EEG), where a set of electrodes are placed on the scalp. While offering a relatively cheap and lightweight solution, fast and efficient decoding of the (often noisy) EEG data is generally regarded as a challenging problem, with decoding accuracies using traditional machine learning methods stagnating [4]. Yet, fast and accurate decoding of EEG is key to constructing successful non-invasive BCIs.

Deep learning has recently broken barriers in many research domains. In the field of BCIs, promising results are now starting to flourish. In particular, convolutional neural networks (CNNs) have achieved good decoding accuracies on motor imagery based EEG tasks, where subjects image the execution of motor movements [5]–[8].

While these studies have indeed shown high accuracies, it is furthermore crucial for real-world applicable assistive devices to minimize power consumption. A possible strategy to both utilize the increasing progress in the field of deep learning and to overcome the limits with respect to energy consumption comes from the field of neuromorphic computing. This class of hardware can run massive, parallel simulations of spiking neural networks on small devices with energy consumption orders of magnitudes lower than traditional hardware [9], [10]. Previous studies have shown that it is possible to directly map various trained continuous-valued deep learning networks to their spiking analogue [11], [12]. Although an EEG classifier running on neuromorphic hardware has been previously constructed [13], [14], therein the potential and rich framework coming along with deep learning-based classifiers has not yet been exploited.

In this contribution, we demonstrate the viability of mapping a CNN for EEG classification to a spiking network running on the SpiNNaker neuromorphic chip. With this, we aim to provide a scalable guideline for creating well-performing EEG decoders deployable on lightweight and power-efficient hardware, setting the appropriate starting point for the creation of wearable BCI-based devices.

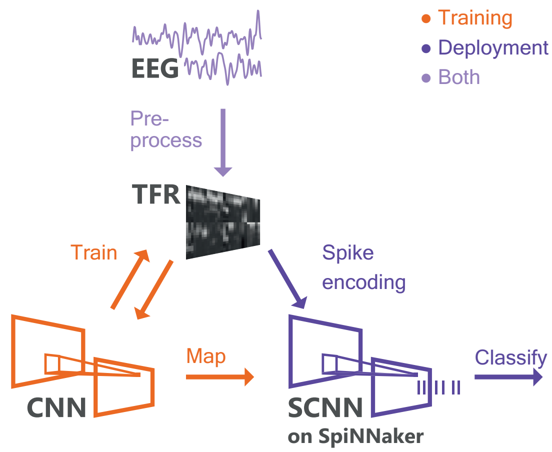

Fig. 1: Overview of the pipeline. Raw EEG data is preprocessed and transformed into a time-frequency representation (TFR). A continuous-valued convolutional neural network (CNN) is trained to classify the TFR. The CNN is subsequently mapped to a spiking convolutional neural network (SCNN) and deployed on the SpiNNaker neuromorphic chip.

II. MODEL AND METHODS

A. Pipeline Overview

An overview of our work is pictured in Figure 1. First, the raw EEG signal was filtered and a time-frequency representation (TFR) of each trial was created. Next, convolution

[1]The first two authors contributed equally to this work

[2]Corresponding author (n.berberich@tum.de)

*Authors are with the Institute for Cognitive Systems, Technical University of Munich, Germany

*This work was supported by the Elite Master Program in Neuroengineering at the Technical University of Munich, funded by the Elite Network Bavaria (ENB)

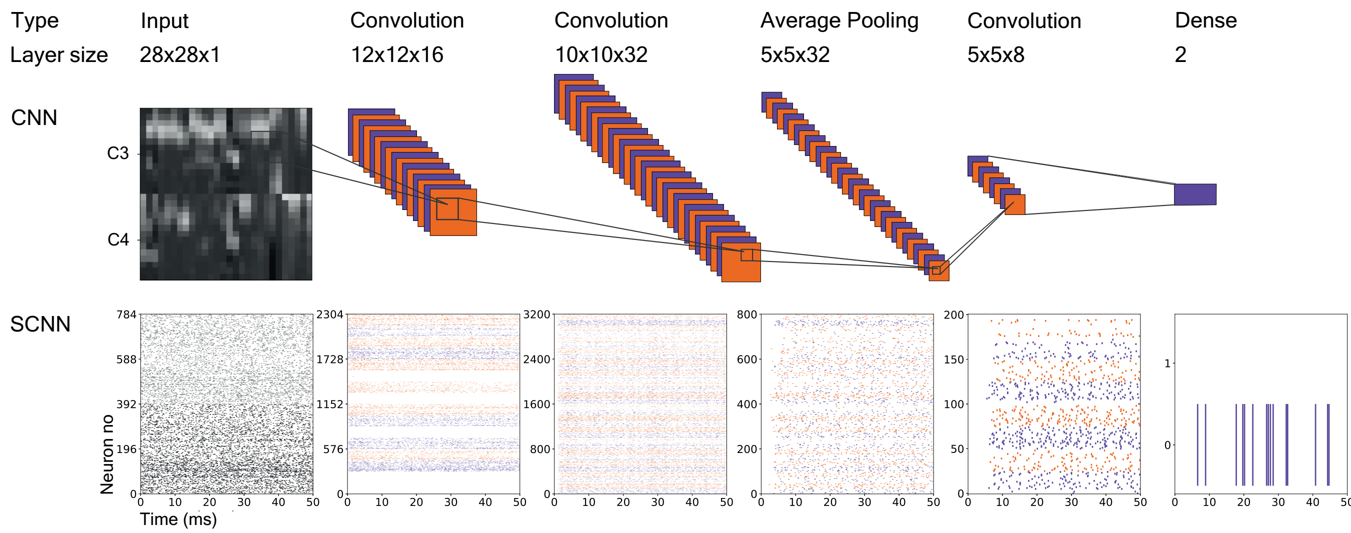

Fig. 2: Overview of the presented architecture for the CNN (top) and spike raster plots of the corresponding layers in the SCNN (bottom). The models consist of 2 convolutional layers followed by an average pooling layer, one more convolutional layer, and finally, one dense layer. Note that the input to the SCNN (bottom-left) is generated by using neurons guided by a Poisson process with rates proportional to pixel values in the TFR (top-left). By counting the spikes of the last layer (bottom-right) the classifier decides between ”0”, i.e. motor imagery of the right hand and ”1”, i.e., motor imagery of the left hand.

neural networks (CNNs) were trained for each subject. Subsequently, the CNNs' weights were mapped to analogue spiking convolutional neural networks (SCNNs), running on the SpiNNaker neuromorphic chip. Code to reproduce our results is available here: `https://github.com/Matthijspals/neuromorphic-EEG`.

B. Dataset Description

The well-established motor imagery EEG database ”Graz data set IV 2B” was used in this study [15], [16]. 9 participants took part in 5 sessions, each consisting of 120 to 160 trials in which left or right hand motor imagery had to be imagined for a period of 4 seconds. During the last three sessions, online visual feedback was given. EEG signal was recorded with electrodes at C3, C4, and Cz with a sampling frequency of 250 Hz.

C. EEG Data Preprocessing

Increasing the signal-to-noise ratio is a crucial first step in any EEG classification. A type 1 FIR band-pass filter was implemented using a hamming window. The cut-off frequencies were set to 8 and 30 Hz, as we assumed based on neurophysiological insight on event-related synchronization and desynchronization during motor imagery, that the information is mainly encoded in the alpha and beta bands [17].

As previous studies showed state-of-the-art accuracy for motor imagery classification by using just the signal from the C3 and C4 electrodes, which are positioned over the left and right motor cortex, we decided to follow suit [18], [19]. In order to create a two-dimensional representation of the EEG signal that can be classified with regular CNNs, a family of complex Morlet wavelets (CMWs) was used, since they have outperformed other TFRs, see e.g. [19]. The EEG signal was convolved with a family of CMWs ranging from 8 to 30 Hz to include the alpha and beta frequency bands, for reasons stated below. A logarithmic scale with a base of 10 was used to assign more points to the lower frequencies, which is appropriate for EEG motor imagery classification. The number of cycles was set to be equal to half of the frequency, ensuring a good trade-off between frequency and temporal resolution. Afterwards, the TFR for each of the two channels were stacked on top of each other and the resulting image was normalized. Preprocessing was performed using the open-source MNE package [20].

D. Artificial Neural Network Model

Since it is not within the scope of this work to design a new architecture, we used a small example CNN available with the SNNtoolbox [11]. Given that we are not interested in the exact timing of motor imagery within a trial (and thus not in the exact location of the resulting pattern within the TFR of a trial) we expected CNNs to be appropriate, as they can learn translational invariant patterns.

The CNN consisted of five different layers: two convolutional layers, one average pooling layer, one more convolutional layer and finally one dense layer, as shown in Figure 2. The convolutional layers have 16, 32, and 8 filters with kernel sizes of (5,5), (3,3), and (3,3), respectively. The first convolutional layer has a stride of (2,2), while the other convolutional layers have one of (1,1). The average pooling layer uses both a kernel of size (2,2) and stride of (2,2). All layers in our network use ReLU as an activation function, except for the output layer which is using softmax. Spatial dropout with a dropout rate of 0.5 was added after each convolutional layer to avoid overfitting [21].

E. Mapping Spiking Neural Networks

Rueckauer et al. developed a toolbox that maps the parameters of pre-trained deep continuous-valued networks to (neuromorphic) spiking networks with comparable error rates between the two networks [11], [12]. The SNNtoolbox makes use of proportionality between the input-output behaviour of a spiking neuron and rectified linear unit (ReLU) and thus the possibility of doing such a mapping [11]. It reaches comparable accuracies between the continuous and spiking networks for complicated structures by furthermore allowing max-pooling layers, softmax activation, neuron biases, and batch normalization layers to be converted.

F. SpiNNaker Neuromorphic Chip Specification

The Neuromorphic chip employed in this study is the SpiNNaker chip. It is a neuro-inspired chip able to simulate spiking neurons with a timestep of 10 μs [22]. One chip contains 18 cores, each able to simulate 1000 biologically plausibly spiking neurons in real-time, and has a power consumption of only 1 W at 1.2 V when all 18 cores operate at 180 MHz [10]. The actual power consumption can be expected to fall even below this, as SpiNNaker uses parallel asynchronous event-based processing: neurons are activated only when some event happens and are otherwise maintained in a low-power idle state. The amount of possible connections (synapses) per neuron depends on the model topology but is, in general, around 1000. Although all of the computations are currently run on SpiNNaker through server access, for future online testing, one single chip would be sufficient (20 mm x 20 mm), as it has enough memory capacity to simulate the number of neurons in our current network, while still being of a size small enough to readily be used for real-world brain-computer interfaces, such as those developed for stroke rehabilitation [2].

G. Experiment

To evaluate our CNN, a 5-fold cross-validation scheme for each participant was adopted in the following way. First, the participant's data was split into 5 disjoint test folds, containing equal proportions of each class and session. Every fold thus consisted of a test set entailing 20% of the data. The remaining 80% was then split up in a training (64%) and validation (16%) set. Similar to Behrenbeck et al. [14], this scheme was chosen such that we do not touch the test fold during optimization. For every fold, a model was trained on its training set, while the validation set was used for early stopping; we kept the model that performed best on the validation set, before finally evaluating it on the test fold.

For every trained model, we calculated the factor by which we needed to scale the weights for the SCNN mapping using the SNNtoolbox as described previously. Next, we recreated our CNN architecture in the python spiking neural networks simulator PyNN [23], using the scaled weights. For the input layer, we assigned one neuron per spectrogram pixel (normalized to be between 0 and 1) and encoded its value

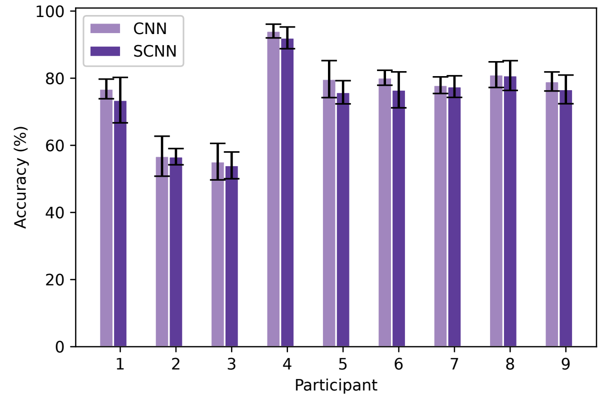

Fig. 3: Mean 5-fold cross-validation accuracy per participant. After converting the CNN to the low energy consuming SCNN, the accuracy decreases by only 1.91%. Error bars denote standard deviation (SD).

as a Poisson process, such that the highest pixel intensity corresponds to an average firing rate of 1000 Hz. This value was set such that the input was high enough to have spikes in the output layer of our model for most trials. After the input layer, we stacked five different populations of neurons with 2304, 3200, 800, 200 and 2 neurons, respectively, to mimic the layers in the CNN. We set the neurons to behave as a leaky integrate-and-fire model. The cell parameters were set to the SNNtoolbox's default such that the SCNN's neurons' input-output behaviour matches that of those in the CNN [11]. The posterior probability for each class after running the spiking network can be calculated by applying a softmax over the average firing rate per output neuron. In case of a tie or no spiking output within the simulation time, a random decision was made.

III. RESULTS

We evaluated the results of our models using an intra-subject 5-fold cross-validation scheme. The convolutional Neural Network (CNN) classified the trials with a mean test accuracy of $75.63\%(\pm 12.25$, standard deviation) over all participants as can be seen in Figure 3 (light bars). While there is a low standard deviation within participants, there is a relatively high standard deviation between subjects, which can likely be explained by the bad performance of our classifier on participant 2 and 3.

Following the pipeline presented in the previous sections, we mapped the trained models for every participant, for every fold, to their spiking equivalents. Then, we deployed the SCNNs on SpiNNaker and measured their performance on their corresponding test fold, resulting in a mean accuracy of $73.72\%(\pm 11.73)$ over all participants, as can be seen in 3 (dark bars). We show that there is only a 1.91% decrease in accuracy after conversion.

IV. DISCUSSION

We report only a small (1.91%) decrease in accuracy after conversion of the continuous-valued deep decoder for EEG data to its spiking equivalent, demonstrating the viability of this approach. The decrease in accuracy is comparable to what is reported in previous studies that map CNNs to SCNNs [11], [12]. We noted that part of the SCNN's errors arose from the models not spiking within the simulation time, yet spiking when given longer or when lowering the neuron's voltage threshold. In fact, during future real-time classification, we expect there to be some trade-off between latency and accuracy.

Regarding future real-time use, we currently use spectrograms as input - a preprocessing step not straightforwardly executed by a spiking network. We aim to explore deep learning-based networks that are able to effectively classify raw data in the future [18]. Recent deep learning approaches like these and others [4], [6], [7] outperform our current proof of concept by using, e.g., data augmentation and larger networks. We note that more complex architectures might be more effectively implemented in neuromorphic chips that allow for weight sharing (e.g., [24]). Since convolutional kernels repeatedly apply the same set of weights, we would need about an order of magnitude less memory to store the synaptic weights per kernel.

V. CONCLUSION

Low-powered and low-weight computing substrates are key for the real-world applicability of wearable brain-computer interfaces. We have demonstrated the viability of transferring accurate deep learning-based EEG classifiers to SCNNs with marginal decreases in accuracy, yet able to run on neuromorphic hardware coming with large gains in energy efficiency.

We show an average accuracy for the SCNN running on SpiNNaker similar to previous reports deploying a biologically-inspired EEG classifier on the same chip [14]. Yet, our approach leads the way towards running deeper and higher performing deep learning architectures on energy-efficient neuromorphic hardware.

ACKNOWLEDGMENTS

Authors thank the Human Brain Project for providing server access to the SpiNNaker neuromorphic chip.

REFERENCES

[1] J. R. Wolpaw, N. Birbaumer, D. J. McFarland, G. Pfurtscheller, and T. M. Vaughan, "Brain–computer interfaces for communication and control," *Clinical neurophysiology*, vol. 113, no. 6, pp. 767–791, 2002.

[2] Z. Qin, Y. Xu, X. Shu, L. Hua, X. Sheng, and X. Zhu, "econhand: A wearable brain-computer interface system for stroke rehabilitation," in *2019 9th International IEEE/EMBS Conference on Neural Engineering (NER)*, 2019, pp. 734–737.

[3] G. Cheng, S. K. Ehrlich, M. Lebedev, and M. A. Nicolelis, "Neuroengineering challenges of fusing robotics and neuroscience," *Science Robotics*, vol. 5, no. 49, pp. 7–10, 2020.

[4] Y. R. Tabar and U. Halici, "A novel deep learning approach for classification of EEG motor imagery signals," *Journal of Neural Engineering*, vol. 14, no. 1, p. 016003, 2016.

[5] 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," *Human brain mapping*, vol. 38, no. 11, pp. 5391–5420, 2017.

[6] Z. Tayeb, J. Fedjaev, N. Ghaboosi, C. Richter, L. Everding, X. Qu, Y. Wu, G. Cheng, and J. Conradt, "Validating deep neural networks for online decoding of motor imagery movements from eeg signals," *Sensors*, vol. 19, no. 1, p. 210, 2019.

[7] W. Huang, L. Wang, Z. Yan, and Y. Liu, "Classify motor imagery by a novel cnn with data augmentation*," in *2020 42nd Annual International Conference of the IEEE Engineering in Medicine Biology Society (EMBC)*, 2020, pp. 192–195.

[8] A. M. Anwar and A. M. Eldeib, "Eeg signal classification using convolutional neural networks on combined spatial and temporal dimensions for bci systems," in *2020 42nd Annual International Conference of the IEEE Engineering in Medicine Biology Society (EMBC)*, 2020, pp. 434–437.

[9] E. Stromatias, D. Neil, F. Galluppi, M. Pfeiffer, S.-C. Liu, and S. Furber, "Scalable energy-efficient, low-latency implementations of trained spiking deep belief networks on spinnaker," in *2015 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2015, pp. 1–8.

[10] E. Painkras, L. A. Plana, J. Garside, S. Temple, F. Galluppi, C. Patterson, D. R. Lester, A. D. Brown, and S. B. Furber, "Spinnaker: A 1-w 18-core system-on-chip for massively-parallel neural network simulation," *IEEE Journal of Solid-State Circuits*, vol. 48, no. 8, pp. 1943–1953, 2013.

[11] B. Rueckauer, I.-A. Lungu, Y. Hu, M. Pfeiffer, and S.-C. Liu, "Conversion of continuous-valued deep networks to efficient event-driven networks for image classification," *Frontiers in Neuroscience*, vol. 11, p. 682, 2017.

[12] B. Rueckauer and S. Liu, "Conversion of analog to spiking neural networks using sparse temporal coding," in *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2018, pp. 1–5.

[13] Z. Tayeb, E. Erelik, and J. Conradt, "Decoding of motor imagery movements from eeg signals using spinnaker neuromorphic hardware," in *2017 8th International IEEE/EMBS Conference on Neural Engineering (NER)*, 2017, pp. 263–266.

[14] J. Behrenbeck, Z. Tayeb, C. Bhiri, C. Richter, O. Rhodes, N. Kasabov, J. I. Espinosa-Ramos, S. Furber, G. Cheng, and J. Conradt, "Classification and regression of spatio-temporal signals using NeuCube and its realization on SpiNNaker neuromorphic hardware," *Journal of Neural Engineering*, vol. 16, no. 2, p. 026014, feb 2019.

[15] R. Leeb, C. Brunner, S. A. Mueller-Putg, and G. Pfurtscheller, "Bci competition 2008 - graz data set b," *Graz University of Technology*, 2008.

[16] V. Jayaram and A. Barachant, "Moabb: Trustworthy algorithm benchmarking for bcis," *Journal of Neural Engineering*, vol. 15, no. 6, p. 066011, 2018.

[17] G. Pfurtscheller and F. Lopes da Silva, "Event-related eeg/meg synchronization and desynchronization: basic principles," *Clinical Neurophysiology*, vol. 110, no. 11, pp. 1842–1857, 1999.

[18] X. Lun, Z. Yu, T. Chen, F. Wang, and Y. Hou, "A simplified cnn classification method for mi-eeg via the electrode pairs signals," *Frontiers in Human Neuroscience*, vol. 14, p. 338, 2020.

[19] B. Xu, L. Zhang, A. Song, C. Wu, W. Li, D. Zhang, G. Xu, H. Li, and H. Zeng, "Wavelet transform time-frequency image and convolutional network-based motor imagery eeg classification," *IEEE Access*, vol. 7, pp. 6084–6093, 2019.

[20] A. Gramfort, M. Luessi, E. Larson, D. Engemann, D. Strohmeier, C. Brodbeck, R. Goj, M. Jas, T. Brooks, L. Parkkonen, and M. Hämäläinen, "MEG and EEG data analysis with MNE-Python," *Frontiers in Neuroscience*, vol. 7, p. 267, 2013.

[21] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler, "Efficient object localization using convolutional networks," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 648–656.

[22] S. B. Furber, F. Galluppi, S. Temple, and L. A. Plana, "The spinnaker project," *Proceedings of the IEEE*, vol. 102, no. 5, pp. 652–665, 2014.

[23] A. Davison, D. Brderle, J. Eppler, J. Kremkow, E. Muller, D. Pecevski, L. Perrinet, and P. Yger, "Pynn: a common interface for neuronal network simulators," *Frontiers in Neuroinformatics*, vol. 2, p. 11, 2009.

[24] M. Davies, N. Srinivasa, T. H. Lin, G. Chinya, Y. Cao, S. H. Choday, G. Dimou, P. Joshi, N. Imam, S. Jain, Y. Liao, C. K. Lin, A. Lines, R. Liu, D. Mathaikutty, S. McCoy, A. Paul, J. Tse, G. Venkataramanan, Y. H. Weng, A. Wild, Y. Yang, and H. Wang, "Loihi: A Neuromorphic Manycore Processor with On-Chip Learning," *IEEE Micro*, vol. 38, no. 1, pp. 82–99, 2018.