Decoding Neural Signals with a Compact and Interpretable Convolutional Neural Network

Artur Petrosyan(✉), Mikhail Lebedev, and Alexey Ossadtchi

National Research University Higher School of Economics, Moscow, Russia
petrosuanartur@gmail.com, aossadtchi@hse.ru
https://bioelectric.hse.ru/en/

Abstract. In this work, we motivate and present a novel compact CNN. For the architectures that combine the adaptation in both space and time, we describen a theoretically justified approach to interpreting the temporal and spatial weights. We apply the proposed architecture to Berlin BCI IV competition and our own datasets to decode electrocorticogram into finger kinematics. Without feature engineering our architecture delivers similar or better decoding accuracy as compared to the BCI competition winner. After training the network, we interpret the solution (spatial and temporal convolution weights) and extract physiologically meaningful patterns.

Keywords: Limb kinematics decoding · Ecog · Machine learning · Convolutional neural network

1 Introduction

The algorithms used to extract relevant neural modulations are a key component of the brain-computer interface (BCI) system. Most often, they implement signal conditioning, feature extraction, and decoding steps. Modern machine learning prescribes performing the two last steps simultaneously with the Deep Neural Networks (DNN) [5]. DNNs automatically derive features in the context of assigned regression or classification tasks. Interpretation of the computations performed by a DNN is an important step to ensure the decoding is based on brain activity and not artifacts only indirectly related to the neural phenomena at hand. A proper features interpretation obtained from the first several layers of a DNN can also benefit the automated knowledge discovery process. In case of BCI development, one way to enable this is to use specific DNN architectures that reflect prior knowledge about the neural substrate of the specific neuromodulation used in a particular BCI.

Several promising and compact neural architectures have been developed in the context of EEG, MEG and ECoG data analysis over recent years: EEGNet

B. Kryzhanovsky et al. (Eds.): NEUROINFORMATICS 2020, SCI 925, pp. 420–428, 2021.
https://doi.org/10.1007/978-3-030-60577-3_50

[4], DeepConvNet [8], LF-CNN and VAR-CNN [9]. By design the weights of these DNNs are readily interpretable with the use of well-known approaches for understanding the linear model weights [3]. However, to make such interpretations correct, extra care is needed.

Here we present another compact architecture, technically very similar to LF-CNN, but motivated by somewhat different arguments than those in [9]. We also provide a theoretically based approach to the interpretation of the temporal and spatial convolution weights and illustrate it using a realistically simulated and real data.

2 Methods

We assume the phenomenological setting presented in Fig. 1. The activity $\mathbf{e}(t)$ of a complex set of neural populations $G_1 - G_I$, responsible for performing a movement act, gets translated into a movement trajectory by means of some most likely non-linear transformation H, i.e. $z(t) = H(\mathbf{e}(t))$. There are also populations $A_1 - A_J$ whose activity is not related to movement but impinges onto the sensors. We do not have a direct access to the intensity of firing $\mathbf{e}(t)$ of individual populations. Instead, we observe a K-dimensional vector of sensor signals $\mathbf{x}(t)$, which is traditionally modeled as a linear mixture of local field potentials (LFPs) $\mathbf{s}(t)$ formed around task relevant populations and taks-irrelevant LFPs $\mathbf{f}(t)$. The task-relevant and task-irrelevant LFPs impinge onto the sensors with forward model matrices $\mathbf{G}$ and $\mathbf{A}$ correspondingly, i.e.

$$\mathbf{x}(t) = \mathbf{G}\mathbf{s}(t) + \mathbf{A}\mathbf{f}(t) = \sum_{i=1}^{I} \mathbf{g}_i s_i(t) + \sum_{j=1}^{J} \mathbf{a}_j f_j(t) \tag{1}$$

We will refer to the task-irrelevant term recorded by our K sensors as $\eta(t) = \sum_{j=1}^{J} \mathbf{a}_j f_j(t)$.

The LFPs are thought to be the result of activity of the nearby populations and the characteristic frequency of LFPs is related to the population size [1]. The envelope of LFP then approximates the firing intensity of the proximal neuronal population. The inverse mapping is also most commonly sought in the linear form so that the estimates of LFPs are obtained as a linear combination of the sensor signals, i.e. $\hat{\mathbf{s}}(t) = \mathbf{W}^T\mathbf{X}(t)$ where columns of $\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_M]$ are the spatial filters that aim to counteract the volume conduction effect and tune away from the activity of interference sources.

Our goal is to approximate the kinematics $z(t)$ using concurrently obtained indirect records $\mathbf{x}(t)$ of activity of neural populations. In general, we do not know $\mathbf{G}$ and the most straightforward approach is to learn the direct mapping $z(t) = \mathcal{F}(\mathbf{x}(t))$.

3 Network Architecture

Based on the above considerations, we have developed a compact adaptable architecture shown in Fig. 2. The key component of this architecture is an

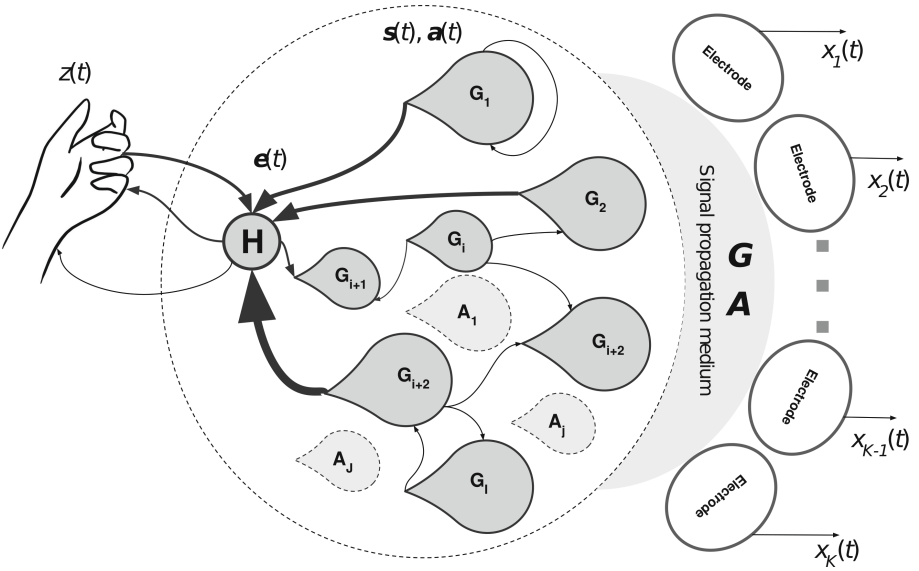

Fig. 1. Phenomenological model

adaptive envelope extractor. Interestingly, the envelope extractor, a typical module widely used in signal processing, can be readily implemented using deep learning primitives. It comprises several convolutions used for band-pass and low-pass filtering and computing the absolute value. We also use non-trainable batch-norm before activation and standardize input signals.

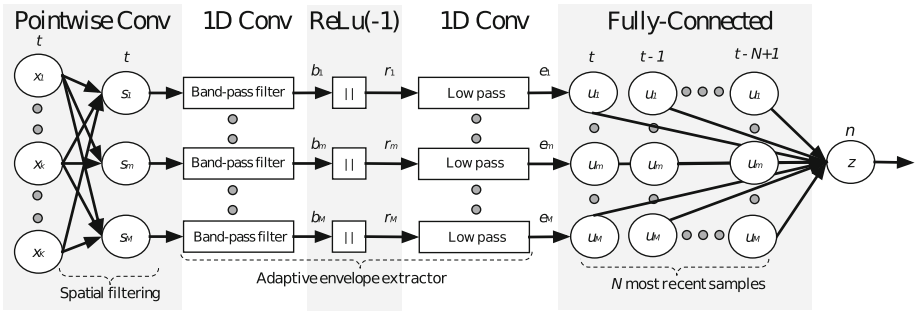

Fig. 2. The proposed compact DNN architecture

The envelope detectors receive spatially filtered sensor signals $\mathbf{s}_m$ obtained by the pointwise convolution layer, which counteracts the volume conduction processes modeled by the forward model matrix $\mathbf{G}$, see Fig. 1. Then, as mentioned earlier, we approximate operator H as some function of the lagged power

of the source time series by means of a fully connected layer that mixes lagged samples of envelopes $[e_m(n), \ldots, e_m(n-N+1)]$ from all branches into a single prediction of the kinematic $z(n)$.

4 Two Regression Problems and DNN Weights Interpretation

The proposed architecture processes data in chunks $\mathbf{X}(t) = [\mathbf{x}(t), \mathbf{x}(t-1), \ldots \mathbf{x}(t-N+1)]$ of some prespecified duration of N samples. In the case when chunk size N equals to the length of the first convolution layer weight vector $\mathbf{h}_m$, the processing of $\mathbf{X}(t)$ by the first two layers applying spatial and temporal filtering can be simply presented as

$$b_m(n) = \mathbf{w}_m^T \mathbf{X}(t) \mathbf{h}_m \tag{2}$$

By design $ReLu(-1)$ non-linearity followed by the low-pass filtering performed by the second convolution layer extracts envelopes of the estimates of the underlying rhythmic LFPs.

Given the one-to-one mapping between the analytic signal and its envelope [2] we can mentally replace the task of optimizing the parameters of the first three layers of the architecture in Fig. 2 to predict envelopes $e_m(t)$ with a simple regression task of adjusting the spatial and temporal filter weights to obtain envelope's generating analytic signal $b_m(t)$, see Fig. 2. Fixing temporal weights to their optimal value $\mathbf{h}_m^*$, the optimal spatial weights can be received as a solution to the following convex optimization problem:

$$\mathbf{w}_m^* = argmin_{\mathbf{w}_m} \{ \parallel b_m(n) - \mathbf{w}_m^T \mathbf{X}(t) \mathbf{h}_m^* \parallel_2^2 \} \tag{3}$$

and similarly for the temporal convolution weights:

$$\mathbf{h}_m^* = argmin_{\mathbf{h}_m} \{ \parallel b_m(t) - \mathbf{w}_m^{*T} \mathbf{X}(t) \mathbf{h}_m \parallel_2^2 \} \tag{4}$$

If we assume statistical independence of neural sources $s_m(t),\ m = 1, \ldots, M$, then (given the regression problem (3) and forward model (1)) their topographies can be assessed as:

$$\mathbf{g}_m = E\{\mathbf{Y}(t)\mathbf{Y}(t)^T\}\mathbf{w}_m^* = \mathbf{R}_m^Y \mathbf{w}_m^*, \tag{5}$$

where $\mathbf{R}_m^Y = E\{\mathbf{Y}(t)\mathbf{Y}(t)^T\}$ is a $K \times K$ covariance matrix of $\mathbf{Y}(t) = \mathbf{X}(t)\mathbf{h}_m$ temporally filtered multi-channel data under the assumption that $x_k(t),\ k = 1, ..., K$ are all zero-mean processes [3].

Then, we observe the exactly symmetric recipe for interpreting the temporal weights. The temporal pattern can be found as:

$$\mathbf{q}_m = E\{\mathbf{V}(t)\mathbf{V}(t)^T\}\mathbf{h}_m^* = \mathbf{R}_m^V \mathbf{h}_m^* \tag{6}$$

where $\mathbf{V}(t) = \mathbf{X}(t)^T \mathbf{w}_m^*$ is a chunk of input signal passed through the spatial filter and $\mathbf{R}_m^V = E\{\mathbf{V}(t)\mathbf{V}(t)^T\}$ is a branch specific $N \times N$ covariance matrix

of spatially filtered data. Here we again assume that $x_k(t)$, $k = 1, ..., K$ are zero-mean processes. Commonly, we explore the frequency domain of temporal pattern to get the sense of it, i.e. $Q_m(f) = \sum_{t=0}^{t=N-1} q_m(t)e^{-j2\pi ft}$, where $q_m(t)$ is the t-th element of $\mathbf{q}_m$ temporal pattern vector.

When the chunk of data is longer than the filter length, the equation (2) has to be written with the convolution operation and will result not into a scalar, but a vector. In this case using the standard Wiener filtering arguments we can arrive at

$$Q_m^*(f) = P_m^{yy}(f)H_m^*(f) \tag{7}$$

as the expression for the Fourier domain representation of the LFP activity pattern in the m-th branch. $H_m^*(f)$ in equation (7) is simply the Fourier transform of the temporal convolution weights vector $\mathbf{h}_m^*$.

5 Simulated and Real Data

In order to generate the simulated data, we precisely followed the setup described in our phenomenological diagram in Fig. 1 with the following parameters. We generated four task-related sources with rhythmic LFPs $s_i(t)$ as narrow-band processes that resulted from filtering the Gaussian pseudo-random sequences in 30–80 Hz, 80–120 Hz, 120–170 Hz and 170–220 Hz bands using FIR filters. We add 10 task-unrelated sources per band with activation time series located in four bands: 40–70 Hz, 90–110 Hz, 130–160 Hz and 180–210 Hz. Kinematics $z(t)$ was generated as a linear combination of the four envelopes. To simulate volume conduction effect we simply randomly generated 4×5 dimensional forward matrix $\mathbf{G}$ and 40×5 dimensional forward matrix $\mathbf{A}$. We simulated 15 min of the synthetic data sampled at 1000 Hz and then split it into equal contiguous train and test parts.

We used open source ECoG + kinematics data set from the BCI Competition IV collected by Kubanek et al to compare our compact DNN's decoding quality to linear models with pre-engineered features. The winning solution provided by Liang and Bougrain [6] have chosen as a baseline in this comparison. Another data set is our own ECoG data CBI (the Center for Bioelectric Interfaces) recorded with a 64-channel microgrid during self paced flexion of each individual finger over 1 min. The ethics research committee of the National Research University, The Higher School of Economics approved the experimental protocol of this study.

6 Simulated Data Results

We have trained the algorithm on simulated data to decode the kinematic $z(t)$ and then to recover the patterns of sources that were found to be important for this task. Figure 3 shows that the only good match with the simulated topographies based on the true underlying sources is performed by *Patterns* using specific to branch temporal filters. The characteristic dips in the bands that correspond to the interference sources activity are demonstrated by the spectral

characteristics of the trained temporal filtering weights. Using the estimation theoretical approach (7), we acquire spectral patterns that closely match the simulated ones and have dips compensation.

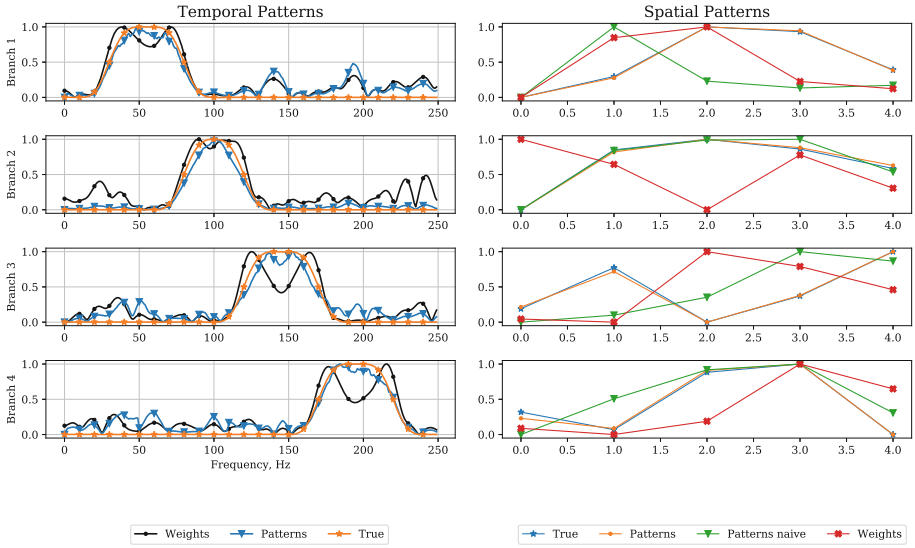

Fig. 3. Temporal and spatial patterns acquired for a noisy case, SNR = 1.5. See the main text for the more detailed description.

7 Real Data Results: BCI Competition IV

In the context of processing electrophysiological data, the main advantage of deep learning based architectures is their ability to perform automatic feature selection in regression or classification tasks [7]. We have found that the architecture with the adaptive envelope detectors applied to Berlin BCI Competition IV data set performs on par or better compared to the winning solution [6], see Table 1.

8 Real Data Results: CBI Data

The following table shows the achieved accuracy for the four fingers of the two patients achieved with the proposed architecture.

In Fig. 4 we have applied the interpretation of the obtained spatial and temporal weights similarly to the way we analysed realistically simulated data. Below we show the interpretation plots for Patient 1 index finger.

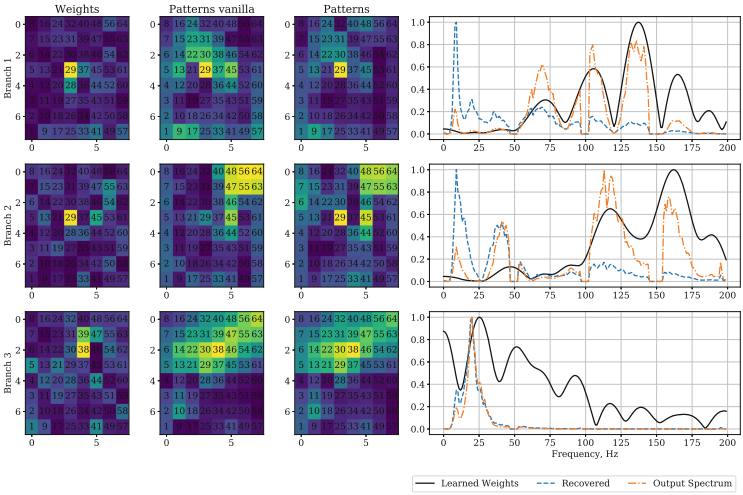

Fig. 4. The interpretation of network weights for the index finger decoder for patient 1 from CBI data set. Each plot line corresponds to one out of three trained decoder's branches. The leftmost column shows the spatial filter weights mapped into colours, while the second and the third columns correspond to vanilla spatial patterns and properly recovered ones. The line graphs interpret the temporal filter weights in the Fourier domain. The filter weights are presented by the solid line, the power spectral density (PSD) pattern of the underlying LFP is marked by the blue dash line. The orange dash line, which is more similar to the filter weights Fourier coefficients, is the PSD of the signal at the output of the temporal convolution block.

Table 1. Comparative performance of our model architecture (NET) and the winning solution (Winner) of BCI IV competition Data set 4: «finger movements in ECoG ».

Subject 1					
	Thumb	Index	Middle	Ring	Little
Winner	0.58	0.71	0.14	0.53	0.29
NET	0.53	0.69	0.19	0.57	0.24
Subject 2					
	Thumb	Index	Middle	Ring	Little
Winner	0.51	0.37	0.24	0.47	0.35
NET	0.49	0.35	0.23	0.39	0.22
Subject 3					
	Thumb	Index	Middle	Ring	Little
Winner	0.69	0.46	0.58	0.58	0.63
NET	0.72	0.49	0.49	0.53	0.6

Table 2. Decoding performance obtained in two CBI patients. The results show the correlation coefficients between the actual and decoded finger trajectories for four fingers in two patients.

	Thumb	Index	Ring	Little
Subject 1	0.47	**0.80**	0.62	0.33
Subject 2	0.74	0.54	0.77	**0.80**

The DNN architecture for the CBI data had three branches, which were tuned to specific spatial-temporal pattern. We demonstrate the spatial filter weights, vanilla and proper patterns, which were interpreted by the expressions described in the Methods section. As you can in Fig. 4, the temporal filter weights (marked by solid line) clearly emphasize the frequency range above 100 Hz in the first two branches and the actual spectral pattern of the source (marked by dash line) in addition to the gamma-band content has peaks at around 11 Hz (in the first and second branches) and in the 25–50 Hz range (the second branch). It may correspond to the sensory-motor rhythm and lower components of the gamma rhythm correspondingly. The third branch appears to be focused on a lower frequency range. Its spatial pattern is notably more diffused than pattern, focused on the higher frequency components in the first two branches. It is consistent with the phenomenon that the activation frequency and size of neural populations are mutually proportional.

9 Conclusion

We introduced a novel compact and interpretable architecture motivated by the knowledge present in the field. We have also extended the weights interpretation approach described earlier in [3] to the interpretation of the temporal convolution weights. We performed experiments with the proposed approach using both simulated and real data. In simulated data set the proposed architecture was able to almost exactly recover the underlying neuronal substrate that contributes to the kinematic time series that it was trained to decode.

We applied the proposed architecture to the real data set of BCI IV competition. Our neural network performed the decoding accuracy similar to the winning solution of the BCI competition [6]. Unlike the traditional approach, our DNN model does not require any feature engineering. On the contrary, after training the structure to decode the finger kinematics, we are able to interpret the weights as well as the extracted physiologically meaningful patterns, which correspond to the both temporal and spatial convolution weights.

Acknowledgement. This work is supported by the Center for Bioelectric Interfaces NRU HSE, RF Government grant, ag. No.14.641.31.0003.

References

1. Buzsaki, G.: Rhythms of the Brain. Oxford University Press, New York (2006)
2. Hahn, S.L.: On the uniqueness of the definition of the amplitude and phase of the analytic signal. Signal Process. **83**(8), 1815–1820 (2003)
3. Haufe, S., Meinecke, F., Görgen, K., Dähne, S., Haynes, J.D., Blankertz, B., Bießmann, F.: On the interpretation of weight vectors of linear models in multivariate neuroimaging. Neuroimage **87**, 96–110 (2014)
4. Lawhern, V.J., Solon, A.J., Waytowich, N.R., Gordon, S.M., Hung, C.P., Lance, B.J.: EEGnet: a compact convolutional network for EEG-based brain-computer interfaces. arXiv preprint arXiv:161108024 (2016)
5. Lemm, S., Blankertz, B., Dickhaus, T., Müller, K.R.: Introduction to machine learning for brain imaging. Neuroimage **56**(2), 387–399 (2011)
6. Liang, N., Bougrain, L.: Decoding finger flexion from band-specific ECoG signals in humans. Front. Neurosci. **6**, 91 (2012). https://doi.org/10.3389/fnins.2012.00091
7. Roy, Y., Banville, H., Albuquerque, I., Gramfort, A., Falk, T.H., Faubert, J.: Deep learning-based electroencephalography analysis: a systematic review. J. Neural Eng. **16**(5), 051001 (2019)
8. Schirrmeister, R.T., Springenberg, J.T., Fiederer, L.D.J., Glasstetter, M., Eggensperger, K., Tangermann, M., Hutter, F., Burgard, W., Ball, T.: Deep learning with convolutional neural networks for brain mapping and decoding of movement-related information from the human EEG. arXiv preprint arXiv:170305051 (2017)
9. Zubarev, I., Zetter, R., Halme, H.L., Parkkonen, L.: Adaptive neural network classifier for decoding meg signals. NeuroImage **197**, 425–434 (2019)