Linear Systems Theoretic Approach to Interpretation of Spatial and Temporal Weights in Compact CNNs: Monte-Carlo Study

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. Interpretation of the neural networks architectures for decoding the signals of the brain usually reduced to the analysis of spatial and temporal weights. We propose a theoretically justified method of their interpretation within the simple architecture based on a priori knowledge of the subject area. This architecture is comparable in decoding quality to the winner of the BCI IV competition and allows for automatic engineering of physiologically meaningful features. To demonstrate the operation of the algorithm, we performed Monte Carlo simulations and received a significant improvement in the restoration of patterns for different noise levels and also investigated the relation between the decoding quality and patterns reconstruction fidelity.

Keywords: ECoG · Weights interpretation · Limb kinematics decoding · Deep learning · Machine learning · Monte Carlo

1 Introduction

A step towards improving the performance of neurointerfaces is the use of advanced machine learning methods - Deep Neural Networks (DNN). DNNs learn a complete signal processing pipeline and do not require hand-crafted features. Interpretation of DNN solution plays a crucial role to 1) identify optimal spectral and temproral patterns that appear pivotal in providing the decoding quality (knowledge discovery) 2) ensure that the decoding relies on the neural activity and not on the unrelated physiological or external artefacts.

Recently, a range of compact neural architectures has been suggested for EEG, ECoG and MEG data analysis: EEGNet [4], DeepConvNet [10], VAR-CNN and LF-CNN [11]. The weights of these architectures are readily interpretable using the standard linear estimation theoretic approaches [3]. However, a special attention is needed to make the correct weights interpretation in the architectures with simultaneously adaptable temporal and spatial weights.

A. V. Samsonovich et al. (Eds.): BICA 2020, AISC 1310, pp. 365–370, 2021.
https://doi.org/10.1007/978-3-030-65596-9_44

2 Generating Data Model

The data generative model is illustrated in Fig. 1. Neural populations $G_1 - G_I$, which are responsible for movement, generate activity $\mathbf{e}(t) = [e_1(t), \dots, e_I(t)]^T$ that is further translated into a trajectory of movements with some non-linear transform H, i.e. $z(t) = H(\mathbf{e}(t))$. We assume that there are populations $A_1 - A_J$ with unrelated movement activity. Its activity is mixed into the sensors as well. At each time step t, we observe K-dimensional $\mathbf{x}(t)$ vector of sensor signals instead of the intensity of firing $\mathbf{e}(t)$ of individual populations. Vector $\mathbf{x}(t)$ is traditionally modelled as a linear mixture with $\mathbf{A}$ and $\mathbf{G}$ matrices, reflecting local field potentials $\mathbf{f}(t)$ and $\mathbf{s}(t)$ formed around both populations:

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

The local field potentials (LFPs) result from the nearby populations' activity and their characteristic frequency is typically related to the size [1] of each population. The firing intensity of the proximal neuronal population is approximated by the envelope of LPF. To counter the volume conduction effect, we will seek to obtain the estimates of LFPs as $\hat{\mathbf{s}}(t) = \mathbf{W}^T\mathbf{X}(t)$ and columns of $\mathbf{W} = [\mathbf{w}_1, \dots, \mathbf{w}_M]$ are referred to as spatial filters.

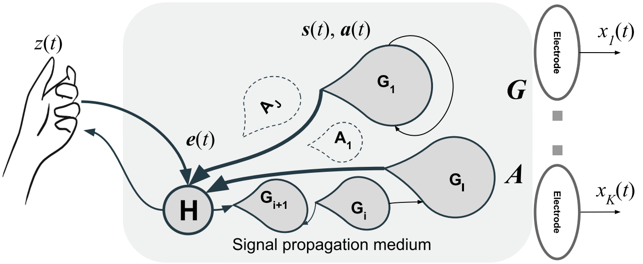

Fig. 1. Phenomenological model

Our regression task is to decode the kinematics $z(t)$ from simultaneously recorded neural populations activity $\mathbf{x}(t)$. Generally, we do not have any true knowledge on $\mathbf{G}$ and other parameters of the forward mapping and transform H, therefore we need to parameterize and learn the entire mapping $z(t) = \mathcal{F}(\mathbf{x}(t))$.

3 Network Architecture

Figure 2 demonstrates our adaptable and compact CNN architecture based on the idea of (1). It consists of spatial filtering, adaptive envelope extractor, and a fully-connected layer. Spatial filtering is done via a pointwise convolution layer

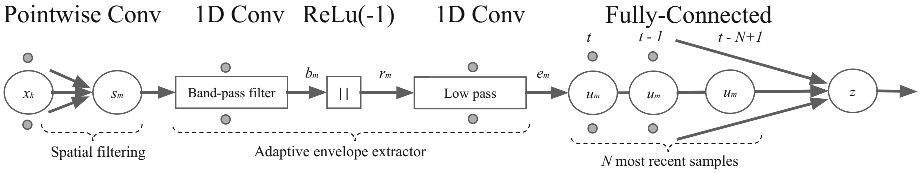

Fig. 2. Proposed compact DNN

used to unmix the sources. Adaptive envelope extractor is in a form of two depthwise convolutions for bandpass and lowpass filtering, used with non-trainable batch normalization (for explicit power extraction) and absolute value nonlinearity in-between. The fully-connected layer is used in order to model envelope to kinematics $z(t)$ transformation with H as a function of lagged envelopes of layers signals extracted previously. This architecture is implemented using the standard DNN layers. In principle, the temporal filtering layers can be replaced by a sinc-layer [8].

4 Spatial and Temporal Weights Interpretation

Assume that the data are processed in chunks of size N equal to the length of the temporal convolutional layer weights $\mathbf{h}_m$ $\mathbf{X}(t) = [\mathbf{x}(t), \mathbf{x}(t-1), \dots \mathbf{x}(t-N+1)]$. Since the set of envelopes maps isomorphically onto a set of analytic signals [2], perhaps with the accuracy to a sign, the task of tuning the weights of the first three layers of our architecture to predict envelopes $e_m(t)$ can be replaced with a regression problem of learning and correcting spatial and temporal weights to get the analytic signal $b_m(t)$ giving rise to the envelope. Assume that the temporal weights are fixed to their optimum value $\mathbf{h}^*_m$, then the optimal spatial filter weights can be obtained as:

$$\mathbf{w}^*_m = argmin_{\mathbf{w}_m}\{\| \; b_m(n) - \mathbf{w}^T_m \mathbf{X}(t)\mathbf{h}^*_m \; \|^2_2\} \tag{2}$$

and therefore assuming statistical independence of the rhythmic LFPs $\{s_m(t)\}$, $m = 1, \dots, M$ the spatial pattern of the underlying neuronal population is [3]

$$\mathbf{g}_m = E\{\mathbf{Y}(t)\mathbf{Y}^T(t)\}\mathbf{w}^*_m = \mathbf{R}^Y_m \mathbf{w}^*_m, \tag{3}$$

where $\mathbf{Y}(t) = \mathbf{X}(t)\mathbf{h}_m$ is a temporally filtered chunk of multichannel data and $\mathbf{R}^Y_m = E\{\mathbf{Y}(t)\mathbf{Y}^T(t)\}$ is a branch-specific $K \times K$ covariance matrix of temporally filtered data, assuming that $x_k(t)$, $k = 1, ..., K$ are zero-mean processes.

Symmetrically we can write an expression for the temporal weights interpretation as

$$\mathbf{q}_m = E\{\mathbf{V}(t)\mathbf{V}^T(t)\}\mathbf{h}^*_m = \mathbf{R}^V_m \mathbf{h}^*_m, \tag{4}$$

where $\mathbf{V}(t) = \mathbf{X}^T(t)\mathbf{w}^*_m$ is a piece of spatially filtered data and $\mathbf{R}^V_m = E\{\mathbf{V}(t)\mathbf{V}^T(t)\}$ is a branch specific $N \times N$ covariance matrix of spatially filtered data, assuming that $x_k(t)$, $k = 1, ..., K$ are all zero-mean processes. To make sense out of the temporal pattern we explore it in the frequency domain, i.e. $Q_m[f] = \sum_{k=0}^{k=N-1} q_m[k]e^{-j2\pi fk}$, where $q_m[k]$ if the k-th element of $\mathbf{q}_m$.

Importantly, as it is the case with spatial pattern, that the obtained vectors $\mathbf{g}_m$ can be usually used to fit dipolar models [6] and locate the corresponding source [3], the temporal patterns $\mathbf{h}_m$ found according to (4) can be used to fit dynamical models such as those, for example, implemented in [7].

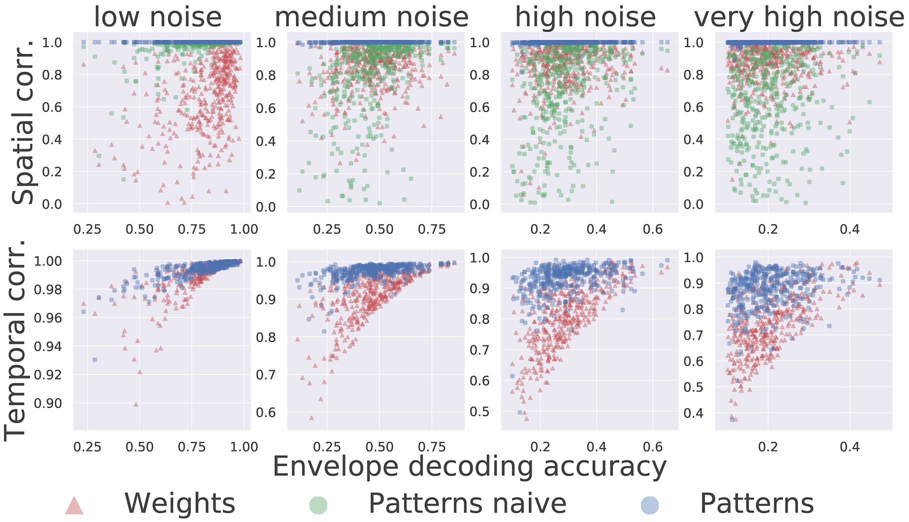

Fig. 3. Monte Carlo simulations. Point coordinates reflect the achieved at each Monte Carlo trial envelope decoding accuracy (x-axis) and correlation coefficient with the true pattern (y-axis). Each point of a specific color corresponds to a single Monte Carlo trial and codes for a method used to compute patterns. *Weights* direct weights interpretation. *Patterns naive* Spatial patterns interpretation without taking branch specific temporal filters into account, *Patterns* - the proposed method

Table 1. Correlation between true and predicted kinematics of the winning solution for BCI competition IV dataset (Winner) and proposed architecture (NET)

Subject 1—2—3	Thumb	Index	Middle	Ring	Little
Winner	.58—.51—.69	.71—.37—.46	.14—.24—.58	.53—.47—.58	.29—.35—.63
NET	.54—.50—.71	.70—.36—.48	.20—.22—.50	.58—.40—.52	.25—.23—.61

5 Comparative Decoding Accuracy

In the context of the electrophysiological data processing, the main benefit of deep learning solutions is their end-to-end learning method which does not require task-specific features preparation [9]. To make sure that our implementation of a simple CNN is capable of learning the needed mapping, we applied it to the collected by Kubanek et al. publicly available data from the BCI Competition IV and compared its performance to that of the winning solution [5].

The results of both algorithms are listed in Table 1. Our simple neural network has comparable decoding quality as the linear model [5] but does not require upfront feature engineering but rather learns the features itself.

6 Monte-Carlo Simulations

We followed the setting described in Fig. 1 to generate the data. We simulated $I = 4$ sources related to the task with rhythmic LFPs $s_i(t)$, occupying the different ranges: 170–220 Hz, 120–170 Hz, 80–120 Hz and 30–80 Hz bands. The target kinematics $z(t)$ was simulated as a linear combination of 4 envelopes of rhythmic LFPs with a vector of random coefficients. We used $J = 40$ unrelated to the task rhythmic LFP sources in the bands of 180–210 Hz, 130–160 Hz, 90–110 Hz and 40–70 Hz. Each band contained ten sources. Matrices $\mathbf{G}$ and $\mathbf{A}$ which model the volume conduction effects at each Monte Carlo trial were randomly generated according to $\mathcal{N}(0, 1)$ distribution. We created 20 min worth of data sampled 1000 Hz.

For neural network training we use Adam optimiser. We made about 15k steps. At 5k and 10k step we halved the learning rate to get more accurate patterns. In total, we have performed more then 3k simulations.

We performed Monte-Carlo study with different spatial configuration of sources at each trial. For each realisation of the generated data we have trained the DNN to predict the kinematic variable $z(t)$ and then computed the patterns of sources the individual branches of our architecture got "connected" to as a result of training.

Figure 3 shows that only the spatial *Patterns* interpreted using branch-specific temporal filters match well the simulated topographies of the true underlying sources. Moreover *Patterns naive* and *Weights* correlation decreasing with noise raises, while *Patterns* is almost perfectly recover true patterns for all noise level settings.

The spectral patterns recovered using the proposed approach also appear to match well with the true spectral profiles of the underlying sources, while directly considering the Fourier coefficients of the temporal convolution layer weights results into erroneous spectral profiles. Using the proper spectral patterns of the underlying neuronal population it is now possible to fit biologically plausible models, e.g. [7], and recover true neurophysiological mechanisms underlying the decoded process.

7 Conclusion

We proposed a theoretically justified method for the interpretation of spatial and temporal weights of the CNN architecture composed of simple envelope extractors. This result extends already existing approaches [3] to weights interpretation. With Monte-Carlo simulations we were able to demonstrate that the proposed approach accurately recovers both spatial and temporal patterns of the underlying phenomenological model for a broad range of signal to noise ratio values.

Acknowledgments. 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. Sig. 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. 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
6. Mosher, J., Leahy, R., Lewis, P.: EEG and MEG: forward solutions for inverse methods. NeuroImage **46**, 245–259 (1999). https://doi.org/10.1109/10.748978
7. Neymotin, S.A., Daniels, D.S., Caldwell, B., McDougal, R.A., Carnevale, N.T., Jas, M., Moore, C.I., Hines, M.L., Hämäläinen, M., Jones, S.R.: Human Neocortical Neurosolver (HNN), a new software tool for interpreting the cellular and network origin of human MEG/EEG data. eLife **9**, e51214 (2020). https://doi.org/10.7554/eLife.51214
8. Ravanelli, M., Bengio, Y.: Speaker recognition from raw waveform with SincNet. In: 2018 IEEE Spoken Language Technology Workshop (SLT), IEEE, pp. 1021–1028 (2018)
9. 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)
10. 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)
11. Zubarev, I., Zetter, R., Halme, H.L., Parkkonen, L.: Adaptive neural network classifier for decoding MEG signals. NeuroImage **197**, 425–434 (2019)