

Real-time Detection of Evoked Potentials by Deep Learning: a Case Study

Leonardo Amato¹, Marta Maschietto², Alessandro Leparulo², Mattia Tambaro²,
Stefano Vassanelli², Alessandro Sperduti¹

1- Department of Mathematics “Tullio Levi-Civita”

2- Department of Biomedical Sciences

University of Padua - Italy

Abstract. In Local Field Potential (LFP) recordings it is hard to distinguish Evoked Potentials (EPs) from spontaneous activity. Automatic real-time detection of all EPs in a recording would enable the deployment of neuromorphic prostheses. In this paper, we present a case study involving EPs induced by stimulation of a whisker in rats. We compare the detection performance of three deep learning models: a Temporal Convolutional Network, a Recurrent Neural Network, and a Mixed model. A data augmentation technique for LFP data and a technique to learn the delay of causal models are proposed. Experimental results show that the three deep learning models are capable of detecting most EPs with few false positives, a delay of less than 100ms, and for a pruned TCN, using only 1,282 parameters.

1 Introduction

Controlling a machine via external recordings of brain activity is an active research area that achieved significant results thanks to Machine Learning approaches (see [1]). These results motivated research in a more challenging and invasive task: the automatic real-time detection of evoked potentials (EPs), i.e. the change in potential generated in response to an external stimulation, in Local Field Potential (LFP) signals recorded via a probe inserted in the brain. Solving this important task would enable, for example, the deployment of neuromorphic prostheses. Up to now, automatic analysis of brain signals through the use of deep learning mainly focused on the classification of already detected EPs [2, 3, 4]. In these works the time-stamps of the EPs are known and the models only classify the type and the quality of the response. Other studies focused on identifying events or decoding brain signals in real-time through the use of sequence-to-sequence models [5, 6]. However, deployment of neuromorphic prosthesis requires solving a different task, i.e. the real-time detection of any single occurrence of EPs. With detection, we refer to the ability of a model to identify when an EP has happened, i.e. to classify each sample of the signal as either “EP” or “non-EP”. The aim of this study is to assess suitable deep learning architectures in the real-time detection of EPs in LFP signals recorded from a probe inserted into the barrel cortex of rats and evoked by the stimulation of a whisker. Specifically, we are interested in models that can learn to distinguish EPs from spontaneous activity taking into account both *efficacy* and *efficiency*.

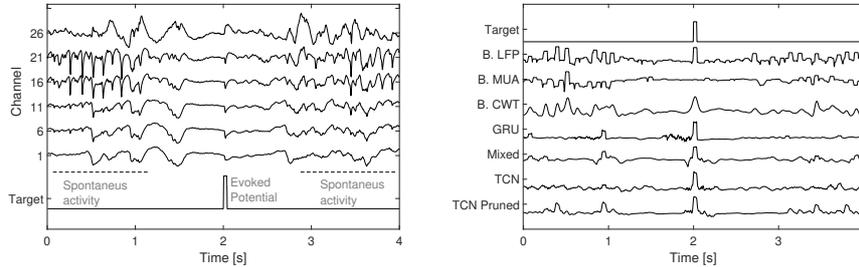
The studied models are Temporal Convolutional Networks [7], Recurrent Neural Networks [8], and a Mixed model that combines them. They are compared to three baseline models that exploit traditional signal processing techniques.

2 Methods

The developed deep learning models all use 6 layers, with convolutional, recurrent, dropout, normalization, and relu activation layers. The baselines use a single convolutional layer with a single filter of size 100ms. This filter can detect the evoked potential by learning the shape of the response, but is unable to discriminate it from spontaneous activity. Using these baselines, we can measure the gain in performance achieved using deep learning techniques. While each deep learning model is trained only on LFP data, which measures the activity of multiple neuron populations, the baselines are trained, alternatively, on LFP data, or Multi-Unit Activity (MUA), which measures the activity of a small set of neuron, or the Continuous Wavelet Transform (CWT) of the LFP. The metrics for efficacy are the Area Under the precision-recall Curve (AUC), and the maximum F1 score computed on the precision-recall curve, while for efficiency we consider the number of parameters and delay introduced by the models. All the models use a sigmoid as output function and a weighted binary cross-entropy as loss.

2.1 Raw data, preprocessing, and labeling

The raw data was extracted using a 32-electrode probe from 25 different rats anesthetized with urethane. In total, 48 recordings were extracted. Each recording is 5 minutes long, has 32 channels, and has a sampling rate of 25,000Hz. To be able to use these data in the training and testing phases, a preprocessing pipeline was used to extract useful features, decrease the size of the data, remove false positives, and remove artifacts. First, the LFP was extracted. Each recording was filtered in the range 1-100Hz with an order 4 Butter-worth filter and down-sampled to 250Hz. Usually, a range of 1-300Hz is used to extract LFP, but in this case, we are mainly interested in the evoked potential, which is found below 100 Hz. To extract the MUA, we used a band-pass filter in the 300-5,000Hz range and selected all points that were below the threshold $-4 \cdot RMS$ to detect each action potential. A moving sum window of size 100 (the downsampling factor) was used on the MUA data before down-sampling to 250Hz to avoid loss of information. This processed MUA contains the number of spikes detected in a 4ms time window. The CWT data is computed in the forward phase of the models with a dedicated layer. During the experiment, all the stimulation time-stamps were stored in a vector. This was used to identify the EPs in the recordings. We labeled the 40 ms after each stimulation as “evoked potential” (class 1), while the rest of the data are labeled as “not evoked potential” (class 0). Data cleaning was performed by first detecting all faulty electrodes and by replacing their relative channels using the average of the two adjacent channels. Of the 32 channels, the top 6 were removed since they mostly contained noise.



(a) Segment of 4 seconds of input's channels 1, 6, 11, 16, 20, 26, and output target. A single evoked potential occurs at second 2, surrounded by significant spontaneous activity at seconds 0-1 and 3-4. (b) Pre-Sigmoid output of the investigated models for the signal reported in Fig. 1a compared versus the ground truth (top signal). Baseline models are reported in rows 2-4. Deep models follow.

Fig. 1: (a) Example of an input signal containing an EP. (b) Target and outputs of the models to the input signal (a).

Second, all recordings where many points labeled as Evoked Potential did not correspond to a detectable change in potential were removed. This situation may occur for multiple reasons, such as incorrect positioning of the probe or faulty whisker stimulation. To speed up this process, we ranked each rat by the amplitude of the averaged evoked potential. Following common practice in this case, only the highest-ranking rat data were selected. The final data set contains 30 recordings extracted from 14 different rats, for a total of 9,000 seconds. An example of a portion of the recording and the target signal is shown in Fig. 1a.

The recordings of one rat, presenting both strong spontaneous activity and strong evoked potentials, were used as test set. The remaining data from the other 13 rats were then segmented and divided into training and validation sets. To partition the data, we extracted a segment of 12 seconds centered on each stimulation. 85% of the segments were used for training, while 15% for validation. This segmentation process has two main advantages: more heterogeneous mini-batches and validation set; faster training. Although this segmentation is not problematic for convolutional networks, which use temporal sliding windows and symmetric padding, recurrent networks could use the center position as evidence of EP. For this reason, we randomly shifted the signal (in the range [-1 sec, +1 sec]) in the segment to prevent the model from learning the center position of the evoked potential. No partitioning was applied to the test set, which is composed of two 5 minute long recordings. This lets us verify that we have not introduced any biases with this partitioning.

2.2 Data Augmentation and Output Delay

The obtained dataset presents three issues: i) it contains only 840 evoked responses for training and 60 for testing; ii) the position of the probe or the size

of the layer of the barrel cortex may vary between different rats; iii) undetected artifacts generated by the stimulation process may be present in some channels, and the model may experience artifact over-fitting. To solve these issues, we introduced a data augmentation layer for LFP data. This layer randomly shifts up or down by 1 channel the segments in the forward phase. This way, we simulate different positions of the probe and expand the training dataset.

Another issue concerns the nature of the task: for this dataset we labeled as evoked potential only a 40ms window that starts when the stimulation to the whisker is applied (as seen in Fig. 1a), but we have no means to measure if this is the optimal "EP" window to use. Due to this the training could fail, since the causal model could not see enough past information in order to detect the EPs. To overcome this problem, we added a non-causal convolutional layer before the sigmoid output function. This solves the problem since we let the model learn the optimal window to use, which can be delayed with respect to the one we have chosen, and then we let the final convolutional layer shift back the output by the same delay. In other words, we introduce "non-causality" in the training phase, which can then be removed by deleting this last layer. The causal model obtained can now detect EPs with a certain delay that has been learned during training.

2.3 Baseline and Deep Learning models

All baseline models follow a general structure: after the data augmentation layer, a single convolutional filter is used. Since we were only interested in the efficacy of the baselines, not the efficiency, we did not put a limit on the number of parameters, and disregarded the detection delay issue. A max pooling of size 40ms has been used for the LFP and MUA input, to approximate an upper envelope of the peak of the EPs, which would otherwise only last for 1 – 4 ms.

As for the Deep Learning models, we selected three suitable architectures: 1.) a Temporal Convolutional Network (TCN), i.e. a causal isomorphic sequence-to-sequence convolutional model that uses dilation and skip connections; 2.) a Recurrent Neural Network using either Gated Recurrent Unit (GRU) or Long Short-Term Memory (LSTM) units; 3.) a mixed model using recurrent layers followed by convolutional layers. After an initial analysis of the models, a decision was made to use a dropout layer with 20% drop chance after each layer, layer normalization after recurrent layers, instance normalization after convolutional layers, gradient clipping, and ADAM optimization. The three models were generated through a model selection process using a random search involving as hyperparameters: the number of layers (from 1 to 6), the number of units/filters for each layer (from 2 to 64), the type of recurrent unit (GRU or LSTM), the size of the filters (from 40 to 300 ms), the type of activation function (none, ReLU or ELU), the type of pooling (none, Max, Average), and the size of pooling (from 2 to 40 ms). In the model selection, the use of an additional dense layer (with a number of nodes from 2 to 12) was also tested. This layer with 6 nodes was tested to increase both efficiency, since it reduces the number of input channels from 26 to 6, and validation efficacy. For all the three models we achieved the

best validation AUC using 6 layers, ReLU activation and max pooling. Gated Recurrent Units were selected as the best units for recurrent layers. Since the TCN model is the most expensive in terms of number of parameters, we used a gradient magnitude pruning approach [9] to optimize it. More specifically, we used an iterative approach by removing the least important filter at each step with a 5% validation error threshold as the stop condition.

3 Results

In Table 1a the efficacy of all models is reported. As expected, the baseline models return many false positives when handling spontaneous activity. In contrast, all deep models are able to filter out spontaneous activity in most of the test recordings. The TCN achieved the best AUC and F1. This model can predict up to 87% of the evoked potentials with 0.86 precision compared to the best baseline that can detect 49% of EPs with 0.34 precision. The GRU model and the mixed model also achieved high precision and recall, but with more false negatives and false positives than the TCN model. Table 1b reports the efficiency of the deep learning models in terms of the number of parameters and detection delay. In terms of detection delay required to detect the evoked potential, the GRU model is the best, having a delay of only 32ms. In terms of number of parameters, the pruned TCN model is the best, using only 1,282 parameters. From the various model selection training trials and the final tests, the learned delay is consistent for each model: a small delay of 30–40 ms for recurrent models; a larger delay of 70–100ms for convolutional and mixed models. Causal convolutional layers can shift the output in the *future* with no loss of efficacy in the same way as the final non-causal convolutional layer can shift the output in the *past*. This explains why we see a larger delay in the convolutional and mixed models compared to the recurrent models.

An example of output of all models for the same input is reported in Fig. 1b.

Model	F1-Score	AUC
Baseline LFP	0.388	0.424
Baseline MUA	0.203	0.334
Baseline CWT	0.221	0.341
GRU	0.776	0.740
Mixed	0.778	0.738
TCN	0.872	0.862
TCN Pruned	0.842	0.803

(a) Efficacy comparison of all tested models using the F1-Score and AUC.

Model	#Param.	Delay
GRU	5,002	32 ms
Mixed	8,090	90 ms
TCN	11,640	80 ms
TCN Pruned	1,282	80 ms

(b) Comparison of the efficiency of deep models in terms of number of parameters and detection delay.

Table 1: Comparison of efficacy (a) and efficiency (b) of the models. The best values for each score are highlighted in bold.

4 Conclusion

With deep neural networks, we can predict up to 87% of the evoked potentials with high precision and low energy cost. The recurrent model was shown to be the fastest in terms of response time, with a detection delay of only 32ms. The TCN has shown to be the most effective model and the one that requires the least operations per second when using pruning. For these reasons, it can be used for both online and offline analysis, opening the door to new experimental settings, where the subject is awake and subject to multiple stimulations. It is also a good candidate for close-loop systems: its low energy cost makes it a suitable component in neural prosthesis for the detection of evoked responses which can be then used to control electrical or magnetic stimulations. A possible explanation on why deeper models perform better in this task is the ability to detect more complex patterns in spontaneous activity. As can be seen in Fig. 1a, spontaneous activity comes as long bursts of negative peaks. Deeper models can detect and discriminate these burst patterns in order to separate them from the evoked potentials.

References

- [1] Ziyang Huang and Mei Wang. A review of electroencephalogram signal processing methods for brain-controlled robots. *Cognitive Robotics*, 1:111–124, 2021.
- [2] No-Sang Kwak, Klaus-Robert Müller, and Seong-Whan Lee. A convolutional neural network for steady state visual evoked potential classification under ambulatory environment. *PLOS ONE*, 12(2):1–20, 02 2017.
- [3] Mohamed Attia, Imali Hettiarachchi, Mohammed Hossny, and Saeid Nahavandi. A time domain classification of steady-state visual evoked potentials using deep recurrent-convolutional neural networks. In *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*, pages 766–769, 2018.
- [4] Nidan Qiao, Mengju Song, Zhao Ye, Wenqiang He, Zengyi Ma, Yongfei Wang, Yuyan Zhang, and Xuefei Shou. Deep learning for automatically visual evoked potential classification during surgical decompression of sellar region tumors. *Translational Vision Science & Technology*, 8(6):21, 2019.
- [5] Gaëlle Milon-Harnois, Nisrine Jrad, Daniel Schang, Patrick van Bogaert, and Pierre Chauvet. 1D vs 2D convolutional neural networks for scalp high frequency oscillations identification. In *2022 30th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, page 211, Bruges, Belgium, October 2022.
- [6] Nur Ahmadi, Timothy G. Constandinou, and Christos-Savvas Bouganis. End-to-end hand kinematic decoding from lfps using temporal convolutional network. In *2019 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, pages 1–4, 2019.
- [7] Colin Lea, René Vidal, Austin Reiter, and Gregory D. Hager. Temporal convolutional networks: A unified approach to action segmentation. In Gang Hua and Hervé Jégou, editors, *Computer Vision – ECCV 2016 Workshops*, pages 47–54, Cham, 2016. Springer International Publishing.
- [8] Yong Yu, Xiaosheng Si, Changhua Hu, and Jianxun Zhang. A review of recurrent neural networks: Lstm cells and network architectures. *Neural Computation*, 31(7):1235–1270, 2019.
- [9] Pavlo Molchanov, Arun Mallya, Stephen Tyree, Iuri Frosio, and Jan Kautz. Importance estimation for neural network pruning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.