

Deep Learning for Real-Time Neural Decoding of Grasp

Original

Deep Learning for Real-Time Neural Decoding of Grasp / Viviani, P., Gesmundo, I., Ghinato, E., Agudelo-Toro, A., Vercellino, C., Vitali, G., Bergamasco, L., Scionti, A., Ghislieri, M., Agostini, V., Terzo, O., Scherberger, H.. - ELETTRONICO. - 14174:(2023), pp. 379-393. (European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in DatabasesT (ECML PKDD 2023) Turin (Italy) September 18–22, 2023) [10.1007/978-3-031-43427-3_23].

Availability:

This version is available at: 11583/2982387 since: 2023-11-03T16:28:38Z

Publisher:

Springer

Published

DOI:10.1007/978-3-031-43427-3_23

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

Springer postprint/Author's Accepted Manuscript

This version of the article has been accepted for publication, after peer review (when applicable) and is subject to Springer Nature's AM terms of use, but is not the Version of Record and does not reflect post-acceptance improvements, or any corrections. The Version of Record is available online at: http://dx.doi.org/10.1007/978-3-031-43427-3_23

(Article begins on next page)

Deep Learning for real-time neural decoding of grasp

Paolo Viviani
LINKS Foundation
Torino, Italy
paolo.viviani@linksfoundation.com

Iaria Gesmundo
Politecnico di Torino
Torino, Italy

Elios Ghinato
Politecnico di Torino
Torino, Italy

Andres Agudelo-Toro
DPZ, Deutsches Primatenzentrum GmbH
Göttingen, Germany
aagudelo-toro@dpz.eu

Chiara Vercellino
LINKS Foundation
Politecnico di Torino
Torino, Italy

Giacomo Vitali
LINKS Foundation
Politecnico di Torino
Torino, Italy

Letizia Bergamasco
LINKS Foundation
Politecnico di Torino
Torino, Italy

Alberto Scionti
LINKS Foundation
Torino, Italy

Marco Ghislieri
Politecnico di Torino
Torino, Italy

Valentina Agostini
Politecnico di Torino
Torino, Italy

Olivier Terzo
LINKS Foundation
Torino, Italy

Hansjörg Scherberger
DPZ, Deutsches Primatenzentrum GmbH
Göttingen, Germany

Abstract

Neural decoding involves correlating signals acquired from the brain to variables in the physical world like limb movement or robot control in Brain Machine Interfaces. In this context, this work starts from a specific pre-existing dataset of neural recordings from monkey motor cortex and presents a Deep Learning-based approach to the decoding of neural signals for grasp type classification. Specifically, we propose here an approach that exploits LSTM networks to classify time series containing neural data (i.e., spike trains) into classes representing the object being grasped.

The main goal of the presented approach is to improve over state-of-the-art decoding accuracy without relying on any prior neuroscience knowledge, and leveraging only the capability of deep learning models to extract correlations from data. The paper presents the results achieved for the considered dataset and compares them with previous works on the same dataset, showing a significant improvement in classification accuracy, even if considering simulated real-time decoding.

1 Introduction

Neural decoding refers to the task of correlating signals recorded from the brain to variables in the outside world such as limb movement. This task is relevant for neuroscientists trying to understand the information contained in neural signal to improve our models of the brain, as well as for researchers developing Brain-Machine Interfaces (BMIs) to control physical and virtual objects (e.g., robotic prostheses, mouse cursors) [1]. While the role of Machine Learning (ML) in neural decoding is known [2, 3], improvements in this field can positively affect the quality of life of patients relying on BMIs to control their prosthesis (i.e., by providing a more natural control or by reducing the need of frequent re-training).

This work was developed in the context of the B-Cratos EU project, which deals with the real-time translation of intra-cranial brain signals, with the final goal of controlling a robotic prosthesis. In particular, this paper reports the results obtained in the simulated real-time decoding of the grasp type, based on a specific dataset previously recorded from the motor cortex of two Non-Human Primates (NHP) [4], investigated as a propaedeutic activity to the final prosthesis controller model deployment based on experimental data acquisition carried out within the project's scope. The main contributions proposed are:

- an LSTM model to detect the grasping phase from time series of neural data;
- an LSTM model to classify the object being grasped by the monkey that provides higher accuracy than the current state of the art for the same dataset.

Both models work on data provided as they would be in a real-time application: the grasping phase detection is representative of the capability to identify the beginning and the end of a grasp movement, while the classification of the object is used as a proxy for the grip type. Importantly, in the presented approach, the authors only focused on improving the predictive performance for practical applications without introducing any prior neuroscience knowledge in the model, and without the intent to provide any deeper understanding of neural activity and its correlation to the outside world.

The paper is structured as follows: section 2 discusses the most relevant prior works that define the state of the art for neural decoding in general and this specific task in particular. Section 3 and 4 describe in detail the dataset used for this work and the ML approach for detecting movement and classifying the grasp type. Section 5 reports and discusses the results obtained with the proposed approach, along with some considerations about the real-life application of these results. Finally, section 6 provides an overview about the relevance of these results and introduces ongoing research in the same direction.

2 Related work

Glaser et al. [2] and Livezey et al. [3] recently provided comprehensive reviews of the state of the art for ML and Deep Learning (DL) for neural decoding, observing how most of the applications still rely on very simple models (i.e., linear regression, Kalman filters) [1, 5]. These reviews also highlighted several works that have shown how LSTM can be effective to model the time evolution of neural signals [6, 7]. Another interesting aspect is the different purposes that models for neural decoding can serve: if the goal is to understand the information contained in neural activity from a neuroscientist point of view, simple, explainable models are to be preferred over black-box machine learning algorithms, which are conversely more suited to provide strong predictive performance for BCI application. It should also be considered that advancements in explainable ML/DL may change this scenario.

With respect to the decoding performance of ML models, since the purpose of this work is strictly related to the predictive performance on a specific dataset, comparing this approach to prior works that dealt with completely different datasets is not trivial, nor particularly interesting for this work, as the task performed by the NHP, the recording conditions, and the pre-processing applied by each researcher can be completely different. As a consequence, the main reference benchmark for this work is a paper from Schaffelhofer et al. [4], which first tried to apply neural grasp decoding to this specific dataset. The approach was based on a naive Bayesian model used to classify the type of object being grasped by the NHP by looking at an entire region of interest in the neural recording (i.e., the phase when the NHP was holding the object). The accuracy obtained by Schaffelhofer et al. was of 62%, on 50 classes (i.e., objects), evaluated using a leave-one-out cross validation technique. A recent thesis from F. Fabiani [8] discussed the results obtained by applying a sliding window to a longer region of the recordings, to simulate real-time operation. An LSTM network with a convolutional layer on top was used in this case, and the decoding results are reported not on the individual 50 objects, but grouping them by their shape (6 groups, variable size within groups) and size (6 groups variable shape within group): while the shape of the object was successfully classified with an accuracy of 92% over 6 classes, the size decoding accuracy was around 25% over 6 classes. Section 5 will discuss how the presented approach differs from the previous works and how the results compare.

3 Dataset structure

3.1 Acquisition and original pre-processing

The dataset used for this work was acquired by researchers at Deutsches Primatenzentrum¹ (DPZ) and involved recording neural signals from electrode arrays implanted in the AIP, F5, M1 regions of the brain cortex of two purpose-bred macaque monkeys (*Macaca mulatta*), animal M and Z. The specific task performed by the NHPs is described in figure 1 and involved grasping a set of objects at specific times, while recording the signal from 192 electrodes in the motor cortex. The original recording

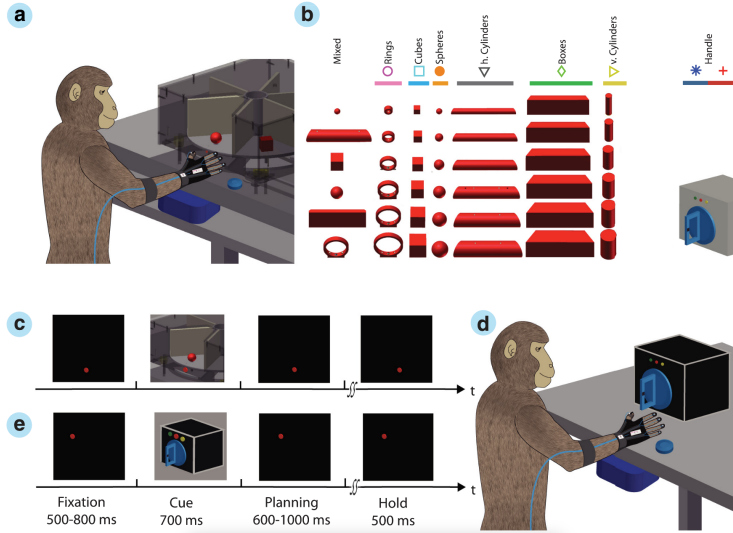


Figure 1: Experimental set up, adapted from [4]. a) Two macaque monkeys were trained to grasp objects presented on computer-controlled turntable. b) The grasping objects are grouped in 50 classes. c) Objects were presented in pseudorandom order and the grasping task followed pre-defined steps: Fixation; Cue; Planning; Movement and Hold. d,e) Adding two supplementary LEDs the monkey is prompted to perform a precision (yellow led) or a power (green led) grip on a special handle.

sampled at $24kHz$ has been processed through a band pass filter ($0.3 - 7kHz$) [9], then a *spike sorting* algorithm has been applied offline to each recording session independently [4]. Spike sorting algorithms recognise neurons activation patterns in the raw signal and can also identify multiple patterns to isolate multiple neurons from the same physical channel. The output of the spike sorting algorithm is defined as a multi-channel *spike train*. A binary time series contains the information about activation time of each neuron (as identified by the spike sorting) during the experiment. For each animal involved in the experiment, three recording sessions acquired in during different days were made available, each one with a different number of spike channels (i.e., neurons) recorded due to the different outcome of the spike sorting algorithm for each session. The details about the datasets used are provided in section 3.2.

The process described in this section was performed out of the scope of the present work: only the final spike trains were available to the authors and are used here.

3.2 Structure

The datasets corresponding to the six recording sessions were provided as Neo [10] objects, with multi-channel spike trains segmented in *trials*, each one corresponding to an instance of the grasp/release process described in figure 1. Figure 2 represents the recording of the spike trains for a single trial from animal Z, plotting both the spike trains (top) and the same data discretised in $40ms$ time bins (bottom), where each bin contains the number of spikes counted in that time interval for that channel. As also shown in figure 2, timestamps for each experiment phase, corresponding to subsequent stages of the monkey task (e.g., visual cue, planning, grasping, etc.) are reported in the data, as well as the

¹<https://www.dpz.eu/en/home.html>

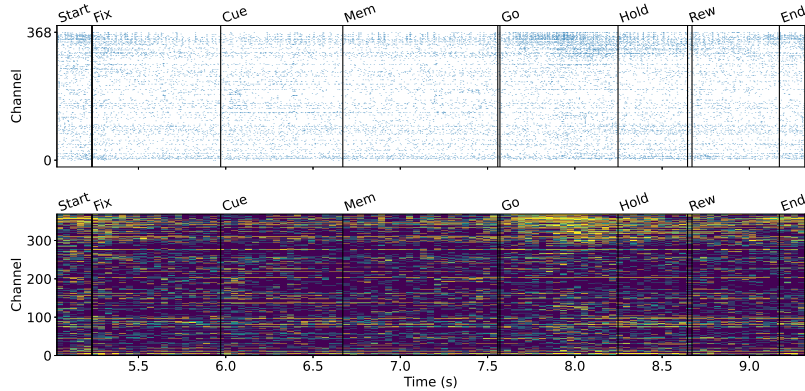


Figure 2: Example of recording for an experimental trial: the top plot reports the firing times for individual spikes for each channel, while the bottom plot reports the intensity map after a $40ms$ time discretisation. The color of each bin is correlated to the number of spikes counted in that time interval and in that channel (increasing from purple to yellow). Time on x axis is reported from the beginning of the recording session.

Table 1: Number of channels and number of trials for each experimental recording session.

NHP identifier	Dataset identifier	# Channels	# Trials
M	MRec40	552	745
	MRec41	568	757
	MRec42	554	653
Z	ZRec32	391	687
	ZRec35	388	724
	ZRec50	369	610

information about the object being grasped (*object id*). It should be noted that not all the phases are reported in figure 1 and that time intervals between phases can change between trials due to subject behaviour.

Table 1 reports the number of channels identified by the spike sorting for each recording session, along with the dataset identifier and the number of trials for that session. It should be also noted that the number of times each object appears in the dataset is not constant among objects and recording sessions, resulting in a class imbalance when training a classifier based on grasped objects.

4 Decoding approach

4.1 Pre-processing

The proposed approach is aimed at simulating on-line decoding of grasped object (used as a proxy for the grasp type), hence the data should be prepared to fit this purpose. The pre-processing steps are split as follows, to separate the different steps and support the development of models without the need of re-processing the whole dataset every time, as intermediate data structures are always stored. These steps are performed separately for each recording session of each animal.

1) Time discretisation The spike trains are discretised in bins of $40ms$ of length. Also shorter time binning has been evaluated to understand if longer, more detailed sequences could benefit the classification performance, but due the lack of evidence of a positive impact, shorter sequences and larger time bins have been selected to lower the computational cost. Each trial was converted to a dense matrix with an ID and stored to disk, and the related metadata was stored in a dedicated data structure (progressive bin number associated to experiment phases; object id; data matrix id).

2) Training, validation and test split The split of each dataset in training, validation and test sets is critical: in fact, since the sequences used to train the model are partially overlapping, splitting data at sequence level would lead to an artificially high accuracy as highly overlapped sequences can appear in all sets. To avoid this effect and provide realistic validation of the approach, the split was performed at the entire trial level (i.e., entire trials were assigned to either train, validation or test set), setting apart 80% of the trials for training, 16% for validation, and 20% for testing. Moreover, class-based stratification² was performed to ensure sufficient representation of all classes in all the datasets.

In this phase, under-represented classes (i.e., objects appearing in less than 3 trials for each recording session) were removed from the dataset to ensure the presence of at least one representative of the class in each dataset partition. Moreover, object of identical shape and size are present in multiple groups (c.f. columns of figure 1b) and were mapped to the same class.

3) Sequence creation After splitting the trials in the three datasets, a *sliding window* was applied to each trial. The sub-sequences extracted by the sliding window represent the data a real-time decoder would see at any given moment, and were tagged either with a label indicating that the monkey was in rest position, or with the grasped object id if the last bin of the sequence was falling in the *region of interest* that has been defined as the *hold* phase of the experiment. This choice was made to be coherent with the approach from Schaffelhofer et al.

The extraction of sequences is depicted in figure 3. From the sub-sequences, two different datasets are created that will be used for different learning tasks:

- an *grasping phase* dataset, that contains all the sequences labelled as *grasp* (if they fall in the region of interest) and *rest* (elsewhere). Class imbalance is roughly 10 to 1 for rest vs. grasp;
- a *classification* dataset, containing only the sequences with the last bin falling in the region of interest, labelled with the id of the grasped object.

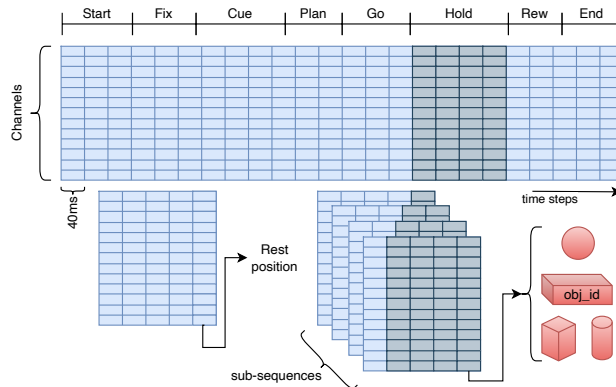


Figure 3: Illustration of the sequence creation process. Each discretised trial is scanned with a sliding window, advancing one time step at a time and extracting fixed-length sequences (2D arrays of dimension $channels \times length$). Each sequence is stored along with a label that can be *rest*, if the last bin falls outside of the *hold* phase, or the object id if it falls within.

These two datasets (for each recording session) are then passed on to the model building step, discussed in the next section.

4.2 Decoding Model

Among the two tasks identified, the classification one has proven to be significantly more challenging than the grasping phase one. All the effort to identify a suitable architecture was then focused on the classification task, and the best performing model was re-used for the grasping phase detection task.

²https://scikit-learn.org/stable/modules/cross_validation.html#stratification

4.2.1 Architecture

Based on previous literature, it has been decided to start the search for a suitable architecture from LSTM networks. Additional experiments were performed within the scope of this work with Bidirectional LSTM networks [11, 12], that demonstrated significantly better performance than simple LSTM networks.

These experiments included a preliminary architecture and hyperparameter search that led to identify the following subset of hyperparameters, which have been used to run a final optimisation campaign for each dataset with KerasTuner [13] bayesian optimiser; the search space was defined as in table 2. An example of model architecture is shown in figure 4.

Table 2: Final hyperparameter search space. One model identified for each animal.

Hyperparameter	Values (L1=0.01, L2=0.01)	Selected	
		M	Z
LSTM layers	{ 1, 2, 3, 4 }	2	1
Hidden units	{ 16, 32, 40, 64 }	40	40
Dropout	{ 0, 0.2, 0.4, 0.6, 0.7, 0.8 }	0.8	0.7
Kernel regularisation	{ None, L1, L2, L1 + L2 }	L2	L2
Recurrent regularization	{ None, L1, L2, L1 + L2 }	L2	L1+L2
Initial learning rate	{ 10^{-3} , $2 \cdot 10^{-4}$, 10^{-4} }	10^{-3}	10^{-3}

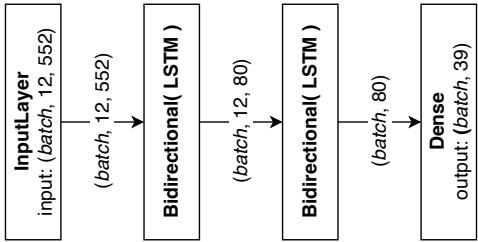


Figure 4: Final architecture for classification network (animal M). Activation functions for the LSTM layers are tanh and sigmoid (recurrent activation). Output layer activation is softmax. Loss function is categorical crossentropy.

4.2.2 Training

The model was implemented with Keras [14], to facilitate its future deployment with Tensrflow lite on a dedicated embedded board³. Training used a mini-batch size of 256 and typically converged within 100 epochs, which are almost never reached due to an early stopping policy based on the evolution of validation accuracy. A learning rate (LR) scaling policy was implemented to halve the LR when loss is plateauing for more than 10 epochs.

5 Results and discussion

5.1 Grasping phase detection

This is a binary classification task with a significant class imbalance towards the *rest* class. The LSTM model reaches an accuracy of at least 98% for all datasets, the F1 score is always greater than 0.95. Table 3 presents both the confusion matrix for one dataset (MRec40) and the accuracy figures for all datasets: if considering this model in the context of a finite-state prostheses control scenario, unwanted movements (rest classified as grasp) would amount to 1% of the time steps considered, while unresponsiveness (grasp classified as rest) is around the 0.1% of time steps. The relevance of these results for a real-life prostheses control is discussed in section 6.

³<https://coral.ai/docs/dev-board-mini/datasheet/>

Table 3: Grasping phase detection results.

(a) Confusion matrix for grasping phase detection (MRec40).

True	Predicted	
	rest	grasp
rest	12824	173
grasp	25	1347

(b) Grasping phase detection accuracy metrics.

Dataset id	Accuracy	F1 score
MRec40	99%	0.96
MRec41	99%	0.96
MRec42	99%	0.97
ZRec32	99%	0.96
ZRec35	98%	0.96
ZRec50	98%	0.95

5.2 Grasped object classification

The results for the classification of the grasped object represent the major contribution of this work and figure 5 and 6 report the confusion matrices and accuracy figures for each recording session available to the authors. To make a sensible comparison with prior art, we should carefully consider the differences between the presented approach and the previous works mentioned in section 2.

In the original 2015 paper [4], Schaffelhofer et al. report an average accuracy for the *hold* phase of 62% over a total of 10 recording sessions (against the six available for this work). This was obtained with an off-line naive bayesian classifier applied to the whole *hold* phase and validated with a leave-one-out (LOO) approach, hence the dataset fraction used for training was significantly higher than what is used in this work. On the other hand, 50 classes were considered while in this case under-represented classes (i.e., less than 3 samples) were removed, and identical objects with different IDs were collapsed onto the same class. In general it is possible to say that the presented approach outperforms the naive bayes classifier (significantly for animal M, and slightly for animal Z) in a harder set-up (sliding window vs. fixed region; smaller training partition). Additionally, as classes are ordered as the objects in figure 1, neighboring classes represent the same object with a slightly different size, hence a very similar grip type; in this sense it has been computed also the *relaxed accuracy*, counting 1-class-away misclassification as correct. Relaxed accuracy results are also reported in table 4 and figure 5, 6, showing a better performance than previous works.

With respect to the results reported by Fabiani [8] for the online decoding, the approach is very similar (i.e., sliding window), although a larger region of interest was considered, including also the *go* phase. In this case results are not reported by individual objects, but by shapes and sizes (respectively rows and columns in figure 1b): to compare the accuracy with the numbers presented in this paper, shape and size accuracy figures are multiplied. The results obtained here are even outperforming the offline decoding algorithm also proposed by Fabiani, which considers a fixed, larger window of each trial for the classification. The comparison of table 4 shows that also in this case, the presented LSTM model outperforms significantly the previous approach, which can be surprising given that LSTM models are used in both cases; it is the author’s opinion that the adoption of Bidirectional LSTM layers provide the most significant improvement in the decoding performance, along with careful usage of regularisation strategies (i.e., a high dropout fraction proved to be particularly effective).

5.3 Real-life usage and current deployment

5.3.1 Reduced training set

When testing real-time neural decoders in the context of real NHP experiments, one of the requirements is the capability to quickly start decoding signals and shortening the training data acquisition as much as possible. Moreover, neural signal evolves through time (i.e., due to mechanical movement of the implant, inflammatory processes, brain plasticity, etc.), making re-training models daily mostly unavoidable.

Here are reported the results of an investigation aimed at understanding how the size of the training set w.r.t. the entire dataset is affecting the accuracy of the trained model. This is particularly relevant as gathering samples is a time consuming operation in both neuroscience research with non-human primates and BMI clinical practice with patients. In this sense, the model capability to retain a good

Table 4: Comparison of results with previous works. Metrics are averaged over the available recording session for the present work (standard deviation is referred to multiple recording sessions, not to multiple training shots) and for Schaffelhofer et al [4]. The work by F. Fabiani only reports results grouped by object type and object size, the individual object accuracy is calculated as the product between the two accuracies [8]. Relaxed accuracy also considers as correctly classified objects belonging to the first neighboring classes.

Animal	Metric	Present work	Schaffelhofer, 2015	Fabiani, 2021
M	Accuracy	69.7 ± 4%	62.9 ± 3.6%	n/a
Z	Accuracy	62.3 ± 1.2%	61.4 ± 4.1%	n/a
Global	Accuracy	65.9 ± 4.9%	62%	22%
	Relaxed accuracy	94.4 ± 3.1%	86.5%	59%

accuracy is critical to ensure its usability in the real world. Table 5 reports the accuracy numbers for an increasing test set partition (and an equally decreasing training+validation set). Qualitatively the accuracy remains quite close to the previous state of the art up to a 30% training partition; moreover, the relaxed accuracy is also better than the state of the art down to 30% of training data.

Table 5: Accuracy and relaxed accuracy for a progressively reduced training set.

Training + validation set	Accuracy	Relaxed accuracy
80%	74.1%	98.3%
70%	70.1%	97.1%
60%	69%	93.7%
50%	62.8%	94%
40%	63.8%	92.8%
30%*	59.2%	87.5%
20%**	51%	81%

*validation is 30% of training set to ensure at least one representative per class

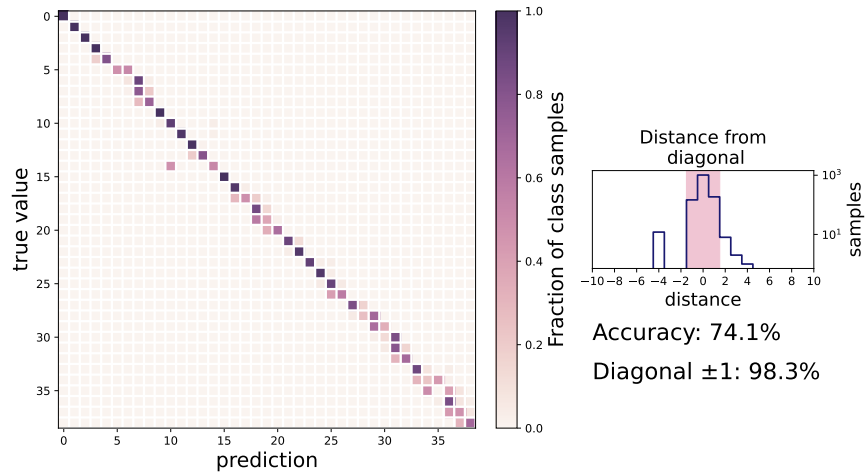
** validation is 40% of training set

5.3.2 Computational cost

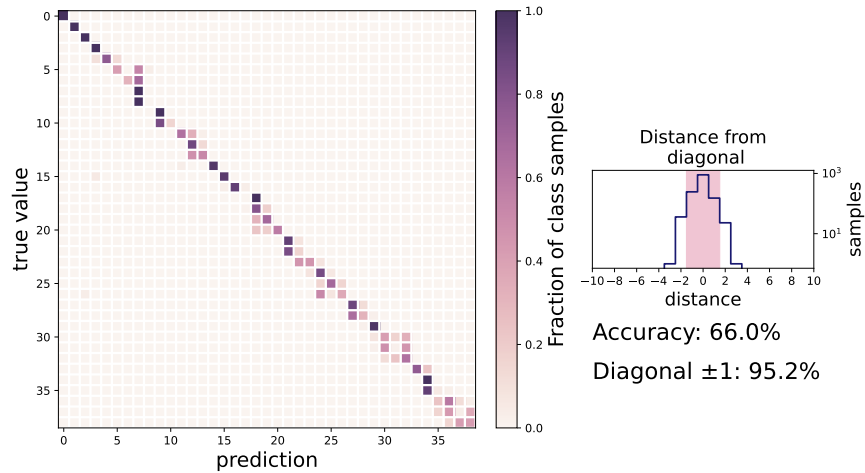
The training time for all the grasp classification cases is lower than one minute on an Nvidia V100, suitable for on-the-fly re-training when new data becomes available during experiments. The larger number of sequences used to train the grasping phase detection model makes each epoch longer, but the number of epochs required to reach a satisfactory accuracy is smaller and contributes to keep the training time at around one minute.

5.3.3 Current deployment of the model

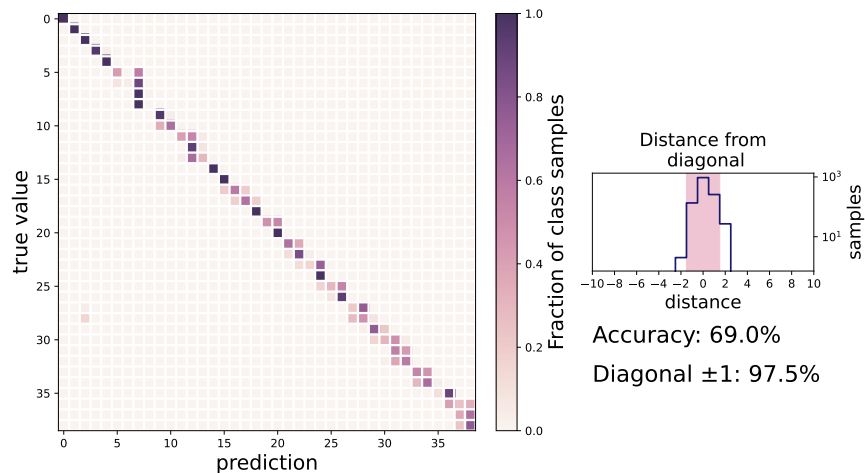
This work represents a preparatory step toward an effective model for real-time continuous control of a prosthesis: while this is a classification task, the data recorded from the brain is fully representative of the kind of data that will be acquired within the scope of the project during 2023, and the network architecture that has been identified will be the starting point for the deployment of a real-time decoder on a dedicated device. Since the final model will predict a limited number of real-valued degrees of freedom, it has been built a synthetic dataset reproducing a trajectory of a single degree of freedom based on the recordings used in this work and the same model presented in figure 4 has been trained to predict this real value with good accuracy. It is in fact currently being used to control the opening and closing of a robotic hand for the first integration tests.



(a) Results for MRec40

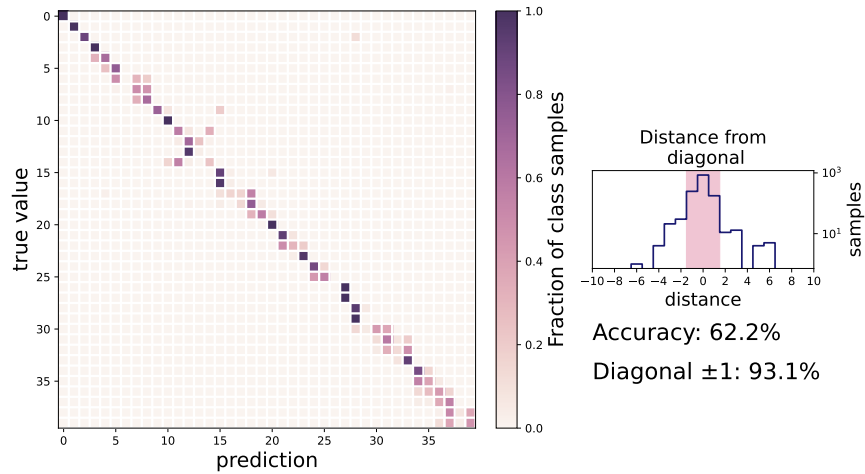


(b) Results for MRec41

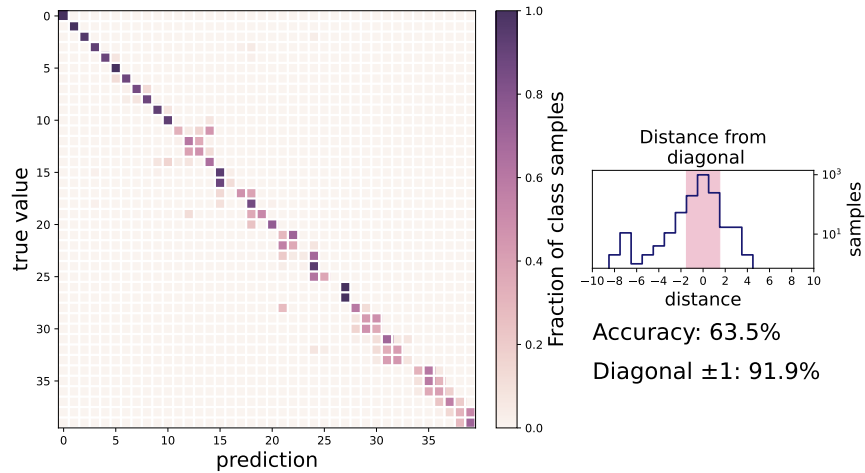


(c) Results for MRec42

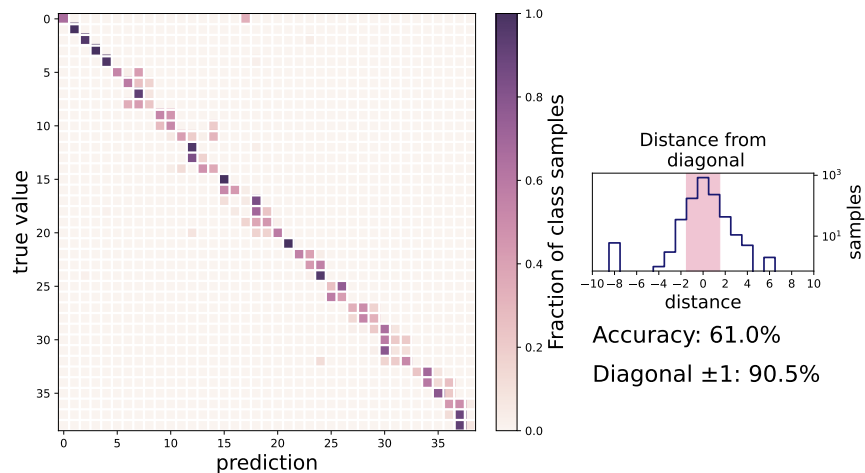
Figure 5: Confusion matrices and accuracy results for grasped object classification for animal M. The models used are the best performing on the validation set among 10 training runs. It can be noted how most of the misclassified samples lie very close to the diagonal.



(a) Results for ZRec32



(b) Results for ZRec35



(c) Results for ZRec50

Figure 6: Confusion matrices and accuracy results for grasped object classification for animal Z. The models used are the best performing on the validation set among 10 training runs. It can be noted how most of the misclassified samples lie very close to the diagonal.

5.3.4 Reproducibility note

The code used to achieve the presented results is available in a public repository⁴. Instructions to access a subset of the data (one recording session) has been made available to reviewers as additional material through the submission portal, and such dataset can be tested directly with the code just mentioned. The entire dataset has been provided by DPZ out of courtesy as part of an ongoing collaboration in the B-Cratos project, but the ownership remains to the original researchers and its complete publication is beyond the scope of this work.

6 Conclusion and future work

The experimental results presented in the previous section demonstrated the capability of bidirectional LSTM networks to match and outperform state-of-the-art methodologies on known data. While this dataset, being used for a categorical classification task, is not immediately relevant for the continuous control of a robotic prosthesis, it validates the suitability of LSTM models for neural decoding of hand grasp actions. In particular, the very good results obtained for the relaxed accuracy are more representative of the performance to be expected when controlling a limited number of degrees of freedom or less nuanced categories (i.e., 5/6 grasp types, opening and closing). It is also worth to highlight how these results were obtained without involving any prior neuroscience knowledge.

With respect to the next steps, the main research activity ongoing is directed towards making LSTM models robust to the evolution of the neural signal through time. In this case this is reflected by the capability to retain information learned during, for instance, the MRec40 session and re-use that information, in the form of a pre-trained model, for decoding MRec41. The goal is to reduce even more the need for time consuming data acquisition and re-training of the models, hence, in a long term perspective, lowering the daily effort for BMI patients and clinicians. In this specific case, the spike sorting applied separately for each session is not helping, as it changes the number of channels and mixes them up preventing the straightforward re-use of a model. In this sense, an effort of applying this model to datasets from literature that are not spike-sorted per session is ongoing. Relying on public datasets will also provide an opportunity to validate this approach against a much wider prior literature.

Acknowledgements

This work was supported by the European Union Horizon 2020 research and innovation program under Grant Agreement No. 965044 and by CINECA, through the Italian SuperComputing Resource Allocation - ISCRA-C grant. We thank Stefan Schaffelhofer for data collection and Federico Fabiani for sharing the useful experience gained during his Master's Thesis work.

Ethics statement

A comprehensive discussion of the ethical aspect of BMIs is out of the scope of this paper: the B-Cratos project has a dedicated ethics advisory board and provides an ethics report to the European Commission yearly. With respect to animal experimentation, the data discussed here has been acquired long before the start of this work and they were subject to regulations and ethics assessment.

On the other hand, ethics topics are particularly relevant when dealing with ML algorithms for BMI use: in this work, for instance, different hyperparameters configurations were needed by animal Z (female) and animal M (male). For a future clinical use, assuming that a single model architecture would fit all users may introduce unwanted biases and negatively affect the user experience of patients.

Finally, several privacy and security aspects have been discussed in relation to this application: from anonymised data storage that can prevent the association between patients brain recordings and their identity, up to the possibility of adversarial actors affecting the prosthesis movement by exploiting weaknesses specifically related to the ML model (i.e., by injecting malicious training data) and the potential mitigation actions. These discussions are also reported in the ethics deliverable submitted to the EC and not yet publicly accessible.

⁴<https://github.com/LINKS-Foundation-CPE/Neural-decoding-paper-2023>

References

- [1] C. Pandarinath and S. J. Bensmaia, “The science and engineering behind sensitized brain-controlled bionic hands,” *Physiological Reviews*, physrev.00034.2020, Sep. 2021.
- [2] J. I. Glaser, A. S. Benjamin, R. H. Chowdhury, *et al.*, “Machine Learning for Neural Decoding,” *eneuro*, vol. 7, no. 4, ENEURO.0506–19.2020, Jul. 2020.
- [3] J. A. Livezey and J. I. Glaser, “Deep learning approaches for neural decoding across architectures and recording modalities,” *Briefings in Bioinformatics*, vol. 22, no. 2, pp. 1577–1591, Mar. 2021.
- [4] S. Schaffelhofer, A. Agudelo-Toro, and H. Scherberger, “Decoding a Wide Range of Hand Configurations from Macaque Motor, Premotor, and Parietal Cortices,” *Journal of Neuroscience*, vol. 35, no. 3, pp. 1068–1081, Jan. 2015.
- [5] D. M. Brandman, T. Hosman, J. Saab, *et al.*, “Rapid calibration of an intracortical brain–computer interface for people with tetraplegia,” *Journal of Neural Engineering*, vol. 15, no. 2, p. 026 007, Apr. 2018.
- [6] S.-H. Yoo, S.-W. Woo, and Z. Amad, “Classification of three categories from prefrontal cortex using LSTM networks: fNIRS study,” in *2018 18th International Conference on Control, Automation and Systems (ICCAS)*, 2018, pp. 1141–1146.
- [7] N. Ahmadi, T. G. Constandinou, and C.-S. Bouganis, “Decoding Hand Kinematics from Local Field Potentials Using Long Short-Term Memory (LSTM) Network,” in *2019 9th International IEEE/EMBS Conference on Neural Engineering (NER)*, San Francisco, CA, USA: IEEE, Mar. 2019, pp. 415–419.
- [8] Fabiani, F., “Brain-computer interface for bionic prosthetic arm actuation,” M.S. thesis, Politecnico di Torino, Torino, Italy.
- [9] V. K. Menz, S. Schaffelhofer, and H. Scherberger, “Representation of continuous hand and arm movements in macaque areas M1, F5, and AIP: A comparative decoding study,” *Journal of Neural Engineering*, vol. 12, no. 5, p. 056 016, Oct. 2015.
- [10] S. Garcia, D. Guarino, F. Jaillet, *et al.*, “Neo: An object model for handling electrophysiology data in multiple formats,” *Frontiers in Neuroinformatics*, vol. 8, Feb. 2014.
- [11] M. Schuster and K. Paliwal, “Bidirectional recurrent neural networks,” *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.
- [12] J. P. Chiu and E. Nichols, “Named Entity Recognition with Bidirectional LSTM-CNNs,” *Transactions of the Association for Computational Linguistics*, vol. 4, pp. 357–370, Jul. 2016.
- [13] T. O’Malley, E. Bursztein, J. Long, *et al.*, *Keras Tuner*, <https://github.com/keras-team/keras-tuner>, 2019.
- [14] F. Chollet *et al.*, *Keras*, <https://keras.io>, 2015.