LSTM Network for hand movement prediction

Gianmarco Guarnier, Davide Lazzarin, Virna Marliani, Giulia Nava, Lorenzo Sterzi

Abstract

In this study we have analysed the code proposed by *Kirill Yashuk* for hand gesture predictions with the purpose of controlling a robotic prosthesis by using a LSTM network. In particular, four different hand gestures were recorded with a Myo armband in order to train the network. LSTM NNs are one of the most used approaches to extract time-domain features.

The aim of our work is to understand the limitations of the original project as well as its strengths. To achieve this objective we analysed the dataset, modified the training process and evaluated the network performances on a different set of data, downloaded from Ninapro databases.

The sEMG signal has been firstly analysed through an Exploratory Data Analysis (EDA) in order to interpret predictive capability of the net and to explain criticalities in relation to the activation of different muscles in different movements. The hyperparameters have been adjusted to evaluate if some combinations can increase performances, but the results confirm the goodness of the original net. We used the cross-test in order to have a better evaluation of the network performances.

Then, different combinations of movements from Ninapro, similar to “rock” - “paper” - “scissors” - “ok”, have been compared. We observed that the accuracy is lower than in the original dataset, this is probably due to the fact that the Ninapro dataset has a lower training size. After applying smoothing techniques (linear envelope) on rectified signals we obtained comparable results to the original dataset. We have been adding more and more hand gestures produced by a single subject from Ninapro to the net, and we observed that by increasing the number of the hand gestures added to the net, the accuracy decreases.

Finally, the net has been tested on several other combinations of movements in order to evaluate the capability of distinguishing little muscles’ activations differences. This is limited by the small number of sensors, their position and the lack of sensors able to distinguish the activation of single muscles and therefore the movement of one finger from another.

Introduction

Electromyography (EMG) is a valuable technique for studying human movement, evaluating mechanisms involving neuromuscular physiology and diagnosing neuromuscular disorders.
The EMG signal is detected by electrodes. There are two basic types of electrodes: surface and indwelling.
The signals are sent to an amplifier that increases the magnitude of the signal so that it can be digitized with high fidelity by the analog-to-digital conversion board residing in a computer.

However, there are many potential pitfalls in the use of EMG as a tool. The researcher could err in the selection of recording electrodes, the recording site, or the data acquisition specifications. Furthermore, the interpretation of the EMG signal requires a thorough knowledge of the origin of the signal. The acquisition of EMG signal has several limitations:

- A repeated identical movement does not give a unique representation: this means that every time the movement is repeated, we will have a different signal.
- It cannot be compared between different subjects, due to a lack of pattern univocity, different muscle mass and different thickness of adjacent tissues that perform low pass signal filtering. For instance, thicker adipose tissue will reduce the signal amplitude. An electrode placed farther away from active fibers has a greater high-frequency attenuation [1].
- The salient anatomical features that affect the EMG signal include: variations in muscle fiber length, fiber type composition, muscle partitioning. These anatomical and architectural muscle features differ among muscles and even within and among individual subjects. Thus, they need to be considered to ensure proper EMG recording and interpretation.

- It is difficult to isolate the activation of a single muscle because of physiological cross talk, that is an interference due to concurrent activation of other muscles.
- ECG contamination. Fortunately, the impact of ECG contamination decreases as the intensity of the contraction increases.

It is therefore necessary to identify the distinctive patterns that give information about the relevant signal and not consider the less significant patterns. In the case of a stationary signal, such as isometric contraction, the amplitude of interference rises as the force increases.

Some of the most commonly used techniques for processing the sEMG signal include: Linear Envelope Detection, Moving Average, Average Rectified Value, Root-Mean-Square. Linear Envelope Detection is the most applied demodulation technique used to extract information from the observed EMG waveform. This technique consists of two steps: the first step is full-wave rectification, the second step is low pass filtering. The slowly changing EMG waveform associated with linear envelope detection is often preferred to facilitate the extraction of: area, slope, onset, shape characteristics of the muscle activity profile.

In literature, several methods and Neural Network structures to recognize EMG sequences are reported, dynamic methods (CNN, RNN) are preferred to static (FFNN) thanks to lower parameters and training time. The LSTM is a special structure of the RNN, which can model the long-term dependencies by alleviating the vanishing gradient problem of the RNN. There are controversial debates in literature regarding the use of CNN versus RNN for sequences modelling tasks, sometimes the use of a hybrid network using CNN and LSTM in parallel has been proposed, in general there is a similar accuracy. LSTM has been efficiently employed to extract global temporal features of motion profile sequences and recognize hand gestures with high accuracy. On the other hand, the LSTM encoder is able to effectively extract features even with variable-length input sequences. It's particularly useful: in fact, length depends on gestures and the people performing it, therefore LSTM allows to maintain accuracy for new participants and executions [2].

Data analysis

For this project, data was made available to the public. To achieve a better understanding of the strengths and weaknesses, the data was processed offline inside MATLAB 2020®. All the data comes from a Myo armband, this device consists of 8 medical grade stainless steel sEMG single differential electrodes and a 9 axis inertial measurement unit. The Myo armband samples at a 200 Hz frequency and saturates at ±137 mV. The device itself presents a notch filter at 50 Hz to cancel power line interferences so no filtering was required for these sensors. All data was recorded from non-amputated subjects, it is important to remark that amputees cannot, in general, produce any reliable ground truth due to the inability to operate any sensor with the missing limb. The Exploratory Data Analysis can be divided in 2 sections: the original dataset [4] and the additional data taken from the NinaPro DB5 [5]. We will proceed discussing the Exploratory Data Analysis of the original dataset in the first section of this paragraph, then further considerations will be made regarding additional data taken from the NinaPro DB5. Finally, the main differences between the two sets of data will be highlighted in the last section of this paragraph.

The Original dataset by *Kirill Yashuk* [4][6] contains the sEMG recordings of four isometric hand positions coming from a single Myo armband positioned on the right forearm. Data was recorded at 200 Hz. According to the author, each gesture was recorded six times for 20 s, when the gesture was already held in position; each recording terminated when the position was still being held. No information was given neither about rest time in between the acquisitions nor about post processing of the acquisitions.

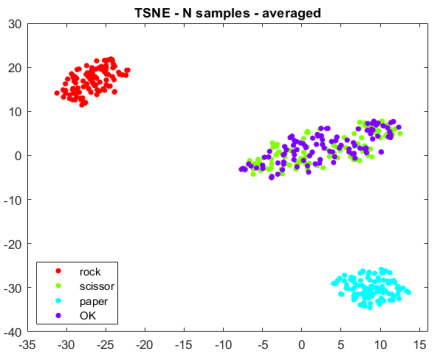

Figure 1. *t-SNE plot*

After a first plot of the data, immediate conclusions could be drawn: the recordings were not rectified or smoothed; the total number of samples for each gesture was inferior to 24000, suggesting that some cleaning was made before making the database available to the public; for most of the recordings, the absolute magnitude of the acquisitions is consistent through the repetitions and coherent with the hypothesis of isometric contractions.

To assess the diversity of the classes we adopted a t-distributed Stochastic Neighbor Embedding (t-SNE) technique. Given the huge

number of samples, the best result was obtained by considering a small number (N=100) of averages applied to a portion of samples picked randomly for each category. The graphic highlights a strong diversity between two of the positions ("Rock" and "Paper") while conveying similarity between the other two gestures ("OK" and "Scissors"). An anatomical analysis of the muscles involved, confirms the conclusions drawn.

The NinaPro database 5 is thoroughly described in the paper *"Comparison of six electromyography acquisition setups on hand movement classification tasks"* [7]. Basically, the datasets from Database 5 are acquisitions of several hand gestures and movements recorded from 10 subjects. Subjects are chosen according to several parameters that may affect the performance to avoid biases. The relevant exercises are divided in three categories: Exercise A - 12 basic movements of the fingers; Exercise B - 8 isometric and isotonic hand configurations and 9 basic movements of the wrist; Exercise C - 23 grasping and functional movements. Each movement repetition lasted 5 seconds and it was alternated with a rest posture lasting 3 seconds. Due to human reaction time, the movements performed may not perfectly match the ground truth labels so a movement detection algorithm was used to correct imperfect labelling. After a plot of the data, the fundamental assumptions regarding the acquisitions protocol and setup were confirmed. With the same algorithm proposed in section 2, the discriminability between each gesture was assessed.

The two sets of data are different under many aspects:

- The acquisition protocol and the post processing of the first set of data are not defined under many aspects, making the ground truth identification uncertain; the opposite can be said regarding the second set of data where automatic labelling was used, and every manipulation was declared.
- While original dataset provides few contextual information, it does provide a significant amount of data for a single subject (116.97 seconds effective acquisition for each gesture on average) compared to the samples from the Nina Pro DB5 (subject 2: 25.82 seconds of effective acquisition for each movement on average).
- The data itself is not comparable between the two datasets because the acquisition protocol is different: in the original dataset, the electromyographic data is acquired only while the gestures are being held, in the second dataset, instead, several exercises performed from a resting position are recorded in a single acquisition.

The original project aims to train a subject-tailored algorithm so it is important to remark that EMG signals coming from different subjects can't be mixed for the training of a classification algorithm. On the other hand, they can be used to validate the model and to assess the robustness of the algorithm.

Code Explanation

The object of this study is a LSTM Network created using Tensorflow.Keras Python3™ libraries whose aim is to classify different hand movements from EMG signals. The related notebook with the Python code is available on Kaggle as "Hand Prediction" and it was uploaded by Yasin Soylu [3].

The EMG signals recorded using a Myo armband with 8 sensors and related to 4 classes of motion have been used to train the LSTM Network. The gesture classes are rock, scissors, paper and ok and they are respectively identified with numbers 0, 1, 2, 3. Rock, paper and scissors gestures are the same as the homonymous game, while to make the OK sign the index finger touches the thumb and the rest of the fingers are spread. The gestures were selected pretty much randomly by the team that created this network. In total there are 120 seconds for each gesture being held in a fixed position, all of them recorded from the same right forearm in a short time span. Every recording of a certain gesture category was concatenated into a .csv file.

The code is divided into three main parts: the data processing, the network building, the training and the performance evaluation.

1)**The data processing**: EMG signals related to each gesture are loaded from the .csv files and saved as 4 different data frames. So, they are concatenated into a single data frame of size 11678 x 65. Each row is composed of 8 samples of the 8 sensors with a sampling period of T = 0,05 s. The last number of each row is a value between 0 and 3 and it stands for the class target. The samples are saved in a matrix 11678 x 64 called X, while the last column is saved in an array called Y. Afterwards X is reshaped into a three-dimensional array of size 11678 x 8 x 8 made as shown in *figure 2*. Moreover, starting from Y, at each row of the new X is associated an array 1 x 4 with the value 1 in the position corresponding to the value of its class movement and 0 elsewhere. Finally, the data frame X with the associated labels Y are splitted up into the training set and test set maintaining a good balance between samples from different movements using stratification. The test size is set to be 25% of the whole dataset.

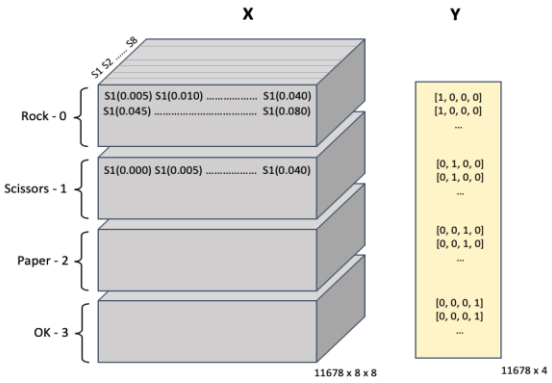

Figure 2. *Data structure*

2)**The network building:** the neural network is built layer by layer. A Long Short-Term Memory (LSTM) recurrent neural network was used. The use of LSTM-NN in this type of application is suggested and confirmed by Miguel Simao et al. and Marco Ghislieri et al. in their recent publications [8][9].
The architecture is the following:

- An input layer with the 8 channels;
- Four LSTM layers used to learn the time-dependencies within the sequential data;
- Each one of the LSTM layers is followed by a dropout layer set to 0.2 to reduce overfitting;
- Three fully connected layer used to convert the output size of the previous layers, into the number of classes to be recognized;
- A SoftMax layer is used to compute the belonging probability to each class.

Moreover, some of the training parameters are set. The team that created this code chose the ADAptive Moment estimation (ADAM) as optimization algorithm and the cross-entropy loss as cost function.

3)**The training and the performance evaluation**: the model is trained upon the training set by using an early stopping on the loss with patience set to 5 and the parameter *validation_split* set to 0.2 so that the 20% of the training set is used for the validation of the model. So, accuracy and loss per epoch are plotted both in the training set and validation set.

Finally, the model is validated on the test set, which was previously created, and its outputs are compared with the targets. This is visualized using the confusion matrix.
The last lines of code generate a summary of the model performance.

Improvements and Results

First of all, to provide a better evaluation of the model performance we decided to perform a cross test. To do that we paid attention to not use samples from the same acquisition both for training and test. For convenience we used six folds, each one corresponding to the i-th acquisitions of the 4 signs concatenated. Thus, we maintained a good balance between signs in each fold. Comparing cross-test results with those obtained by running the original code 5 times and taking the average of runs, we didn't observe any significant differences. In the subsequent code modifications and improvements, the cross test was used to evaluate the performances.

We tested several hyperparameters, in order to check for differences in network performance and evaluate the network behaviour. Our approach is based on the use of two databases: the default Kaggle database, and the NinaPro DB5. The main hyperparameters tested are: *test_size* in the sklearn.model_selection.train_test_split function, *dropout* and *activation* in the layers from tensorflow.keras.layers, *optimizer* and *loss* in tensorflow.keras.model.compile and *patience* in EarlyStopping callbacks. EarlyStopping stops training when a monitored metric (loss) stops improving. Patience is the number of epochs with no improvement after which training will be stopped.

The original network, using Kaggle database, has an accuracy of 96.8% and we observed that none of these edits improve the overall performance, nevertheless we achieved similar results, with an average accuracy of 96.9%.
In the second approach, we evaluated the same classifier using the NinaPro DB5. The high number of possible movements allowed us to evaluate the capability of the network to classify gradually more similar gestures and a greater number of hand exercises. The results were correlated with differences in the activation of the muscles whose EMG signal is collected.
We first selected the most different exercises of this category, on the basis of the results obtained with EDA; therefore we have used the four most widely out spaced exercises in the t-SNE plot. The classifier was tested on exercises 1,5,7 and 17 of category B, performed by subject 2 (a 28-year-old right-handed male). The results showed, with default parameters, that the accuracy is 82,6%. However, overfitting phenomenon and higher variability appear during tests.
Then, the network was tested on the most similar movements to the original dataset ("rock"- "paper" - "scissors" - "ok") as more movements corresponding to these classes, various combinations were tested. ("rock": cat. B 6-17; "scissor", cat. B 2; "paper" cat. B 5-8; "OK" cat. C: 14-15-6), each combination was cross-tested. The best results were obtained with 3 combinations of gestures: 17-2-5-6 with an accuracy of 85%, (performances are worse for "ok" precision 83.8% compared to 90% and more in the other combinations); 17-2-5-15 with an accuracy of 86.8% (highest precision for "scissors" and below average for "paper"); 17-2-8-15 with an accuracy of 86.2% and with good results for each category overall. We supposed that the different performance of the classifier, over Kaggle and Ninapro databases, is due to a limited amount of data in the NinaPro DB5 and to a different acquisition protocol. To improve the accuracy using NinaPro DB5, we needed to process raw data. By rectifying the data we could observe an overall 5% improvement across all gesture combinations. By considering the linear envelope of as well, the performances of the last combination mentioned above improved from 86% to 94%. The network was tested on 2 subjects with comparable results, so even if the EMG is not a comparable signal between different subjects the network is able to detect the characteristic patterns of different movements.

Subsequently, more similar movements were tested. Hand to fist (1 "thumb up", 6 "fingers flexed together in fist", 7 "pointing index", 17 "wrist extension with closed hand"), the accuracy achieved was 82.2% and this is due to the fact that although the hand has a similar position, the thumb up is recognisable thanks to a sensor that records the activity of the radial extensor, in the case 17 we will have an activity of the extensors greater than the sole extension of the index picked up by a different sensor than the radial extensor.
Then the different cylindrical grips were evaluated (1 "large grasp", 2 "small diameter grasp" or power grip, 3 "fixed hook grasp", 5 "medium wrap", 6 "ring grasp") the combination 1,2,3,5 obtained only 58,4% of accuracy, while the combination 1 2 3 6 obtained 67, 4% with the highest accuracy for movement 6 in which in addition to the activity of the flexors there is also that of the extensors, in particular, from the confusion matrix it is observed that movements 2 and 3 are confused a lot, to a lesser extent also 1-2 and 3 with 0 and 1. The objective of this test was to identify if the network was able to distinguish the intensity of the grip but this is complex. With regard to the spherical grasps, 10 "power sphere grasps", 11 "three finger sphere grasps", 12 "precision sphere grasps" and 13 "tripod grasps" were evaluated. The worst accuracy recorded was 50.8%, this is probably due to the fact that in each movement there is the predominant activation of the flexors and there are no substantial differences that help the network to classify the gestures.
Afterwards, the comparison between the "extension and flexion of a single finger with open hand" (1, 2 cat. A), "extension and flexion of the wrist with open hand" was tested (13, 14 cat. B), the accuracy was 66.8% this shows that the network is not able to distinguish the degree of activation of the flexor-extensors, the only one that the network distinguishes with greater ease is the flexion of the wrist that, unlike the other movements that present a preponderant extensor activity, has an important activity of the flexors. Slightly higher accuracy 72.6% was obtained by comparing different degrees of finger extension from the fist to the extended hand (6 "fingers fixed together in fist", 2 "scissors", 3 "flexion of ring and little finger and extension of the others", 5 "abduction of all fingers").
Since the results on similar movements were poor, we tried to combine different types of grips which however obtained unsatisfactory accuracy values (from 61.6% to 73.6%), the only tests that obtained better results were the combinations of a cylindrical grip (1 cat. C), a spherical grip (10 or 11 cat. C) a pincer (15 "tip pinch grasp") and "wrist extension" (14 cat. B) with accuracy of 79.2% and 82.8%. "Wrist extension" is recognisable for the exclusive activity of the extensors, in 15 (cat. C) there is an activation of the brachioradialis and a mix

of extensors and flexors, the cylindrical grip is distinguishable from the spherical one for the activation of the brachioradialis even if they have flexor activity in common.

In order to conclude our evaluation of the network behaviour, we tested the original code to train the network on an increasing number of signs. Since the highest scores were obtained on the combination rock-scissors-paper-ok we evaluated the ability of the network to distinguish more movements (raw signal) from this combination. In particular, with 5 movements adding "open a bottle with a tripod grasp" or "wrist extension" the accuracy is reduced to 81%, in the second case the network has more difficulty in distinguishing paper and wrist extension this is due to the common activation of the extensors only. They were also tested 6 movements (6-1-2-3-4-5) starting from the fist to the extended hand and the accuracy was 70.4% so slightly lower than the same test with 4 movements, from the confusion matrix and the precision is observed that the position 2 is exchanged with the 3 and the 5 with the 4, confirming the greater difficulty in distinguishing movements that differ for the extension of a finger. Then with a rectified and smoothed signal we tested from 5 up to 20 movements. We used this approach to verify if the network could classify more than 4 signs. As shown, increasing the number of signs reduces the validation accuracy. Finally, training more signs increases the number of epochs. In *figure 3* we also reported the epochs.

Number of signs	5	6	7	8	9	12	16	20
Accuracy (%)	95,70	94,98	94,90	94,31	94,78	93,82	92,27	91,16
Validation accuracy (%)	93,32	91,67	93,59	90,61	92,41	88,64	88,10	84,65
Epochs	45	59	71	65	70	84	85	96

Figure 3. *Performance with increasing number of signs*

Conclusions

The original network presents a good accuracy and it's robust against the hyperparameters variation. Thus, there was no need to change to the original hyperparameters in order to have better performances.
Comparing the original network with the NinaPro dataset, the latter showed lower accuracy. This probably happens because of the reduced acquisition time of each movement, so due to the inferior amount of data in this second database.
Moreover, the accuracy is quite reduced when similar movements, which cause similar muscle activation, have to be classified, such as spherical and cylindrical grasps. The main limitation comes from the fact that most of the everyday life activities involve a flexed posture of the hand, and the available sensors are not able to differentiate the activity of the deep flexors from that of the superficial ones. They're also not able to detect the activity of e.g. the flexor of the 3rd finger from the activity of the 2nd finger, or the activity of all of them at the same time.
A direct comparison and further evaluations on the best signal acquisition protocol to training the network is not possible with the currently available data, we can't exclude that a larger Ninapro dataset might produce equal or better results; in a future study it would be ideal to obtain a new dataset of the Ninapro gestures using the same amount of data of the original in order to have more comparable results between the two datasets.

Bibliography:

[1] Kamen and Gabriel, "Essentials of Electromyography", Human Kinetics, 2010
[2] J. -W. Choi, S. -J. Ryu and J. -H. Kim, "Short-Range Radar Based Real-Time Hand Gesture Recognition Using LSTM Encoder," in IEEE Access, vol. 7, pp. 33610-33618, 2019, doi: 10.1109/ACCESS.2019.2903586.
[3] https://www.kaggle.com/yasinsoylu123/hand-prediction
[4] https://www.kaggle.com/kyr7plus/emg-4
[5] http://ninapro.hevs.ch/DB5_DoubleMyo
[6] https://www.kaggle.com/kyr7plus
[7] Pizzolato S, Tagliapietra L, Cognolato M, Reggiani M, Müller H, et al. (2017) Comparison of six electromyography acquisition setups on hand movement classification tasks. PLOS ONE 12(10): e0186132.
[8] Miguel Simão, Pedro Neto, Olivier Gibaru, EMG-based online classification of gestures with recurrent neural networks, *Pattern Recognition Letters*,Volume 128
[9] Ghislieri, M., Cerone, G.L., Knaflitz, M. *et al.* Long short-term memory (LSTM) recurrent neural network for muscle activity detection. *J NeuroEngineering Rehabil* 18**,** 153 (2021).