Machine Learning Engineer Nanodegree

Capstone Proposal

Jonatan Dellagostin
March 15th, 2018

Domain Background

Surface electromyography (sEMG) is the temporal and spatial superposition of faint bioelectrical signals generated by the muscle nerve cells during muscle contraction [1]. It is collected and recorded through skin surface electrodes. Compared to conventional EMG signal acquisition that requires inserting a needle electrode into muscle tissue, the sEMG signal has the advantages of being noninvasive and providing the convenience of collection [2]. Besides, it is known [3] that information extracted from intramuscular MES (myoelectric signal) results in the same classification accuracy compared to information extracted from surface MES.

Knowing the daily life of hand amputees can be extremely difficult compared to what it was before the amputation, the research involving hand prosthetics is of great importance. The development of open source state-of-the-art algorithms in hand movement classification is crucial in this field, parallel with the development of open-source easy-to-print 3d robotic hands, both contributing to the development of cheaper and more accessible prosthetics to amputees, since commercial devices are too expensive and unaffordable for most of them.

All articles that guided the present work are properly cited in the References section.

Problem Statement

Amputees that lack some part of the arm, specially the hand, suffer from not being able to perform simple and crucial task in everyday life. A robotic hand is a proposed solution in the field of prosthetics. In a real hand, electrical impulses sent by the brain travel through the nerves until reaching the arm muscles innervations, which are then activated and produces the muscular activity responsible for the hand movements.

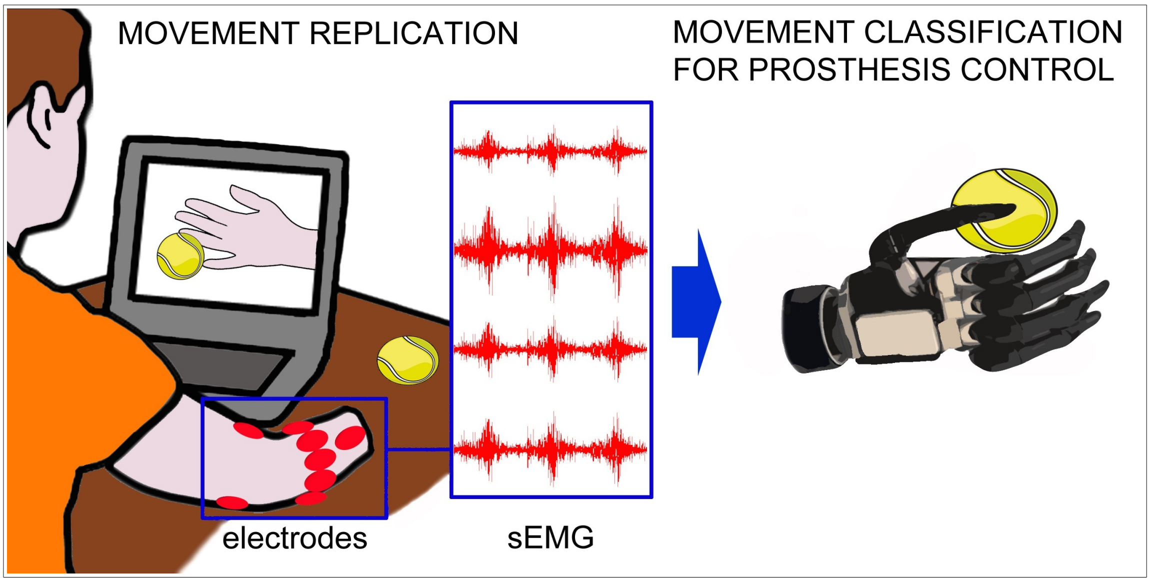


Figure 1: Illustration of prosthesis training and operation (source: http://ninapro.hevs.ch)

A common and established approach in the literature to mimic this behavior is to capture arm muscle's electric signals through sEMG techniques, preprocess them in a computer, apply some classification algorithm to map this signals into one class from a pre-determined finite set of hand motions, and send to the robotic hand the appropriate electrical signals in order to perform the required movement. Our problem then, in the present work, comprises of finding and training an appropriate classifier for the task of predicting the output class (hand movement) given the input data collected (preprocessed sequences of electrical signals converted to the digital world).

Datasets and Inputs

Our research into the field led us to the first successful initiative to establish an open sEMG database, the NinaPro - Non-Invasive Adaptive Hand Prosthetics. The goal of this project is to develop a family of algorithms able to significantly augment the dexterity, and reduce the training time, for sEMG controlled prosthesis. The project is thoroughly described in [4].

By testing researchers findings on a very large collection of data, this project pave the way for a new generation of prosthetic hands. The Data Acquisition is done with the goal of developing a reproducible protocol to acquire large data sets for healthy patients performing certain movements and amputated patients also making complex movements, while analyzing and assessing the data as they become available. The data acquisition includes the acquisition of signal data and the calibration of the sensors to limit the noise in the data. Relevant clinical data is acquired at the same time such as age, gender, height, weight and for amputated patients also the exact place of the amputation and the time between amputation and tests performed.

From its beginning (2014) until the present day, the NinaPro has collected seven datasets, which one comprising of data collected in a uniform way (Ninapro protocol) from several healthy and disabled patients, while performing a variety of hand movements. Each dataset collection and posterior analysis was performed by a group of researchers and is thoroughly described in academic papers. All of the datasets are available online in a dedicated website: http://ninapro.hevs.ch/.

The present work will focus on the dataset number 1 to 3 (http://ninapro.hevs.ch/data1, http://ninapro.hevs.ch/data2 and http://ninapro.hevs.ch/data3)

Solution Statement

Usually, the researchers use classifiers renowned for a long time for being successful in the task of accurately predicting hand movements from sEMG, mostly Random Forests and Support Vector Machines, using a very few number of engineered features.

However, it’s well known that in the last years deep learning based models are surpassing in several fields the performance of other algorithms. These approaches, like in Convolutional Neural Networks, induce some bias into the models about the nature of the inputs, and overcome the burden of feature engineering by learning the features itself.

Since our inputs are basically time-series events, we propose the use of a Recurrent Neural Network, more particularly a Long-Short Term Memory Neural Network (LSTM). This model is already known for its overwhelming capacity of dealing with time-series sequences, and we hope it will improve the classification performance when compared to other already used models.

Benchmark Model

We will benchmark our proposed model against the models used and described in [4]. In this work, the datasets 1, 2 and 3 from NinaPro database were used as input, and a Convolutional Neural Network architecture was chosen as a model for the classifier. This paper compares its own results with earlier works too.

For 50 classes (hand movements), the classifier obtained an average classification accuracy of 66.59 ± 6.40% on dataset 1, 60.27 ± 7.7% on dataset 1 and 38.09 ± 14.29% on dataset 3 (composed only by data collected from amputees – classification accuracy is expected to be lower than the previous datasets, composed only by able subjects).

Figure 2: The 50 hand movements used as output classes for the classifier (source: http://ninapro.hevs.ch)

Earlier works cited in the paper got better performance, though; the best classical classification methods were: on dataset 1, Random Forests with all features, with an average classification accuracy of 75.32 ± 5.69%; on dataset 2, Random Forests with all features with an average classification accuracy of 75.27% ± 7.89%; and in dataset 3, a Support Vector Machine with all features, with an average classification accuracy of 46.27% ± 7.89%.

Evaluation Metrics

Since the classes are balanced by definition (the acquisition phase took the same number of examples for each class), the classification accuracy (number of correct classifications/ number of trials) seems a reasonable metric to be used when comparing the performance of different algorithms (in fact, this is the usual metric found in the literature).

Nevertheless, we pretend to report some other metrics along, like the chance level (classification accuracy / number of classes) and the training and classification average times (given a certain CPU or GPU used). Since the model training could be done offline, but the classification couldn’t, the classification average time is of great importance: a delay of more than 300 ms between the muscular activation an the prosthesis response can be noted by the patient and degrade the usability of the device.

Project Design

The workflow of our work will consist in two steps, data extraction an model implementation.

1- Data extraction

Our input data will be downloaded from Ninapro database and separated by dataset, specifically datasets one to three. Although there are other variables included in the datasets, we will focus only in the sEMG signals included. The datasets are described as follows:
1 - Database 1 contains data from 27 intact subjects, sEMG captured by 10 electrodes.
2 - Database 2 contains data from 40 intact subjects, sEMG captured by 12 electrodes.
3 - Database 3 contains data from 11 transradial amputees, sEMG captured by 12 electrodes.
Each example in the datasets corresponds to a hand movement performed by a subject during a interval of 5 seconds, and the raw sEMG captured during this interval. This raw sEMG, segmented in time windows, will be used as input and the corresponding label (hand movement) will be used as the output class, in order to train the classifier.

2 - Model Implementation

Convolutional neural networks (CNNs) are a type of DNN (deep neural network) with the ability to act as feature extractors, stacking several convolutional operators to create a hierarchy of progressively more abstract features. Such models are able to learn multiple layers of feature hierarchies automatically (also called “representation learning”). Long-short-term memory recurrent (LSTMs) neural networks are recurrent networks that include a memory to model temporal dependencies in time series problems.

The combination of CNNs and LSTMs in a unified framework has already offered state-of-the-art results in the speech recognition domain, where modeling temporal information is required. This kind of architecture is able to capture time dependencies on features extracted by convolutional operations.

Our research pointed to a very interesting Deep Neural Network Architecture, described extensively in [4]. The model, called DeepConvLSTM, is a deep learning framework composed of convolutional and LSTM recurrent layers, and is perfectly suitable for our needs, since it is capable of automatically learning feature representations along the Convolutional layers (automating feature extraction from raw sensor data instead of using a time consuming heuristic process of engineering the features) and modeling the temporal dependencies between their activation (using the LSTM units in the recurrent layers). Finally, the model has a softmax layer that outputs the predicted class probability.

The input to the network consists of a data sequence. The sequence is a short time series extracted from the sensor data (sEMG) using a sliding window approach composed of several sensor channels. The number of sensor channels is denoted as D. Within that sequence, all channels have the same number of samples S^l . The length of feature maps S^l varies in different convolutional layers. The convolution is only computed where the input and the kernel fully overlap. Thus, the length of a feature map is defined by:

$$S^{(l+1)} = S^l - P^l + 1$$

where P^l is the length of kernels in layer l. The length of the kernels is the same for every convolutional layer, being defined as $P^l = 5,\ \forall l = 2, \ldots, 5$.

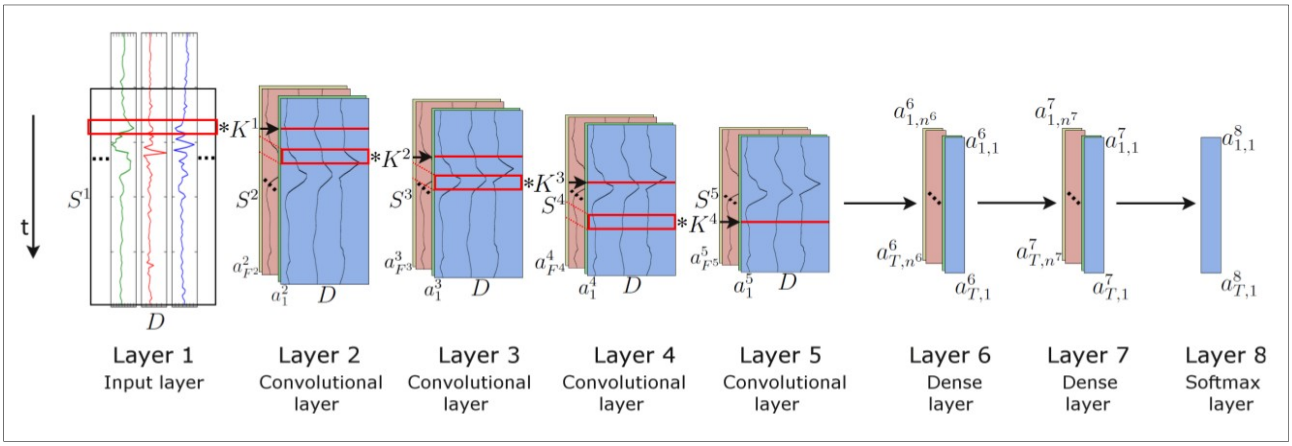


Figure 3. Architecture of the **DeepConvLSTM** (Conv, convolutional) framework for activity recognition. From the left, the signals coming from the sEMG sensors are processed by four convolutional layers, which allow learning features from the data. Two dense layers then perform a non-linear transformation, which yields the classification outcome with a softmax logistic regression output layer on the right. Input at Layer 1 corresponds to sensor data of size $\boldsymbol{D \times S^1}$, where $\boldsymbol{D}$ denotes the number of sensor channels and $\boldsymbol{S^l}$ the length of features maps in layer $\boldsymbol{l}$. Layers 2–5 are convolutional layers. $\boldsymbol{S^l}$ denotes the kernels in layer $\boldsymbol{l}$ (depicted as red squares). $\boldsymbol{F^l}$ denotes the number of feature maps in layer $\boldsymbol{l}$. In convolutional layers, $\boldsymbol{a^l_i}$ denotes the activation that defines the feature map $\boldsymbol{i}$ in layer $\boldsymbol{l}$. Layers 6 and 7 are dense layers. In dense layers, $\boldsymbol{a^l_{t,i}}$ denotes the activation of the unit $\boldsymbol{i}$ in hidden layer $\boldsymbol{l}$ at time $\boldsymbol{t}$. The time axis is vertical.(source: [5])

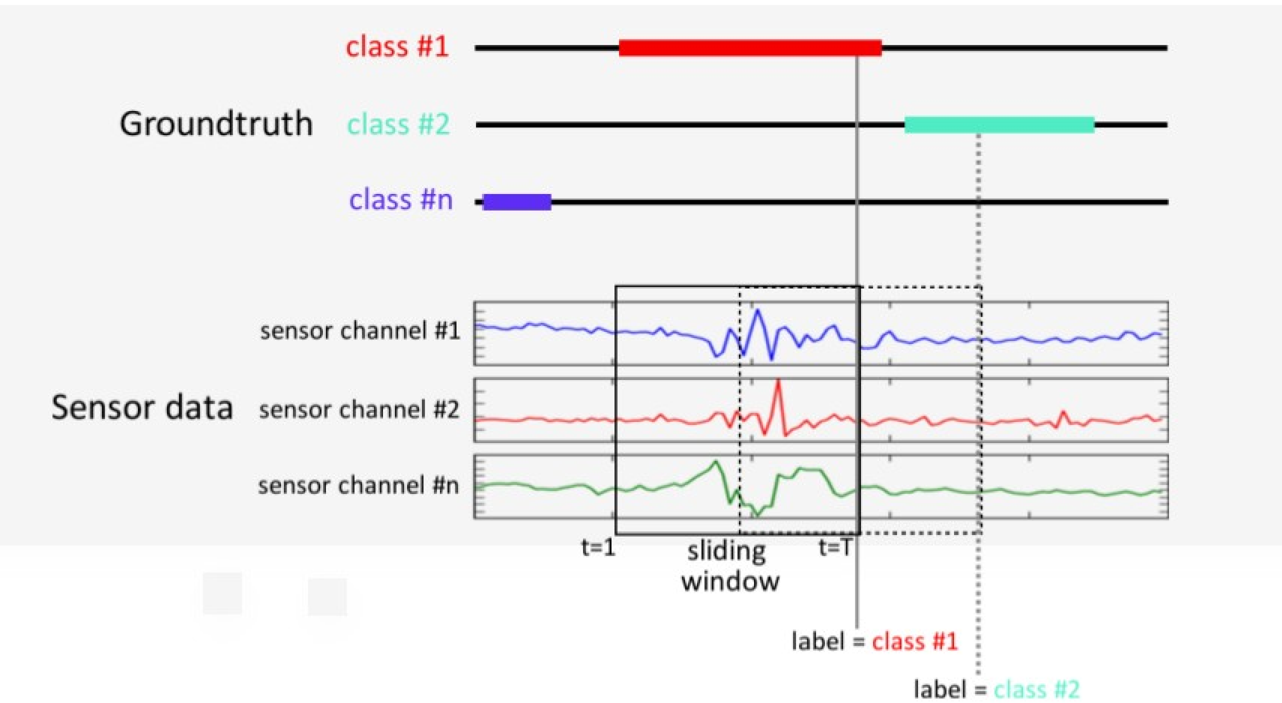
Convolutional layers process the input only along the axis representing time. The number of sensor channels is the same for every feature map in all layers. These convolutional layers employ rectified linear units (ReLUs) to compute the feature maps. Layers 6 and 7 are recurrent dense layers. Recurrent dense layers adapt their internal state after each time step. The activation of the recurrent units is computed using the hyperbolic tangent function. The output of the model is obtained from a softmax layer (a dense layer with a softmax activation function), yielding a class probability distribution for every single time step t.

A shorthand description of this model would be: ***C(64) => C(64) => C(64) => C(64) => R(128) => R(64) => Sm***, where $\boldsymbol{C\ (F^l)}$ q denotes a convolutional layer $\boldsymbol{l}$ with $\boldsymbol{F^l}$ feature maps, $\boldsymbol{R(n^l)}$ a recurrent LSTM layer with $\boldsymbol{n^l}$ cells and ***Sm*** a softmax classifier.

The model is trained in a fully-supervised way, backpropagating the gradients from the softmax layer through to the convolutional layers. The network parameters are optimized by minimizing the cross-entropy loss function using mini-batch gradient descent with the RMSProp update rule.

For the sake of efficiency, when training and testing, data are segmented on mini-batches of a size of 100 data segments. Using this configuration, an accumulated gradient for the parameters is computed after every mini-batch. Both models are trained with a learning rate of 0.001 and a decay factor of $\rho = 0.9$. Weights are randomly orthogonally initialized. We introduce a drop-out operator on the inputs of every dense layer, as a form of regularization. This operator sets the activation of randomly-selected units during training to zero with probability $p = 0.5$.

The length of the time window is 200 ms, with a step size of 100 ms. The number of instances (segments) obtained after using this sliding window configuration will be aproximatelly 5s/100ms = 50. Each segment is composed by synchronized sEMG signals captured from D sensors, at a rate of 2 KHz, resulting in an input matrix of dimensions D x 400. The class associated with each segment corresponds to the hand movement that has been observed during that interval (0.2 seconds).



DeepConvLSTM outputs a class probability distribution for every single time step t in the sequence (i.e., the 200-ms window of the sEMG sensor signal). However, we are interested in the class probability distribution once DeepConvLSTM has observed the entire 200-ms sequence.

Since the memory of LSTM units tends to become progressively more informed as a function of the number of samples they have seen, DeepConvLSTM returns the class probability distribution only at the last time step T, when the full sequence has been observed. Thus, at the time of each sample of the original sensor signal, DeepConvLSTM provides a class probability distribution inferred from processing a 200-ms extract of the sEMG sensor signal prior to that time, as illustrated in Figure 6.

This approach of only using the last time step T is encouraged by the results of a previous work [5]. Their investigation demonstrates that misclassifications occur primarily during the movement onset (initial phase) and offset (final phase). The explanation for this phenomenon is that the sEMG signals are not yet (or not anymore) sufficiently discriminative in these transitory phases between movement and rest. Thus, we hope LSTM units will overcome this issue by captioning these temporal dynamics of the sEMG signals.

The model will be splitted in a 1:3 proportion of test:training set and individually trained using each one of the 3 Ninapro datasets previously cited. After properly trained, the model will be tested and benchmarked, providing the already discussed evaluation metrics for the sake of comparison.

References

1. G Staude, W Wolf, Objective motor response onset detection in surface myoelectric signals. Med. Eng. Phys. 21, 449–467 (1999)
2. Yansheng Wu, Shili Liang , Ling Zhang, Zongqian Chai, Chunlei Cao and Shuangwei Wang. Gesture recognition method based on a single-channel sEMG envelope signal, https://link.springer.com/content/pdf/10.1186/s13638-018-1046-0.pdf
3. Hargrove, Levi & Englehart, Kevin & Hudgins, Bernard. (2007). A Comparison of Surface and Intramuscular Myoelectric Signal Classification. IEEE transactions on bio-medical engineering. 54. 847-53. 10.1109/TBME.2006.889192.

4. Atzori M, Gijsberts A, Castellini C, Caputo B, Mittaz Hager A, Elsig S, Giatsidis G, Bassetto F, Müller H (2014) Electromyography data for non-invasive naturally controlled robotic hand prostheses. Scientific Data 1:140053. https://doi.org/10.1038/sdata.2014.53
5. Ordóñez F.J., Roggen, D., 2016. Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. Sensors 16, 115
6. Allard, Ulysse Côté et al. "Deep Learning for Electromyographic Hand Gesture Signal Classification by Leveraging Transfer Learning."*CoRR* abs/1801.07756 (2018)
7. Boyali, Ali. (2015). Spectral Collaborative Representation based Classification for Hand Gestures recognition on Electromyography Signals. Biomedical Signal Processing and Control. 24. 10.1016/j.bspc.2015.09.001.
8. M. Romaszewski, P. Glomb, and P. Gawron. Natural hand gestures for human identification in a human-computer interface. In Image Processing Theory, Tools and Applications (IPTA), 2014 4th International Conference on, pages 1–6, Oct 2014.
9. Cisotto, Giulia & Michieli, Umberto & Badia, Leonardo. (2017). A coherence study on EEG and EMG signals.
10. Francesca Palermo, Matteo Cognolato, Arjan Gijsberts, Henning Müller, Barbara Caputo, Manfredo Atzori:Repeatability of grasp recognition for robotic hand prosthesis control based on sEMG data. ICORR 2017: 1154-1159
11. Khushaba, Rami & Al-Timemy, Ali & Kodagoda, Sarath & Nazarpour, Kianoush. (2016). Combined Influence of Forearm Orientation and Muscular Contraction on EMG Pattern Recognition. Expert Systems with Applications. 61. 10.1016/j.eswa.2016.05.031.
12. F. Palermo, M. Cognolato, A. Gijsberts, H. Muller, B. Caputo, and ¨ M. Atzori, "Repeatability of grasp recognition for robotic hand prosthesis control based on sEMG data," in Rehabilitation Robotics (ICORR), 2017 International Conference on. IEEE, 2017, pp. 1154–1159.
13. Pizzolato S. Comparison of six electromyography acquisition setups on hand movement classification tasks. PLoS One. 2017; 12(10): e0186132.
14. Cognolato, Matteo & Atzori, Manfredo & Caputo, Barbara & Brugger, Peter & Müller, Henning. (2016). From NinaPro to MeganePro towards the natural control of myoelectric prosthetic hands,.
15. Atzori, M., and Müller, H. (2015). Control capabilities of myoelectric robotic prostheses by hand amputees: a scientific research and market overview. Front. Syst. Neurosci. 9:162. doi: 10.3389/fnsys.2015.00162
16. Manfredo Atzori, Matteo Cognolato, Henning Müller: Deep Learning with Convolutional Neural Networks Applied to Electromyography Data: A Resource for the Classification of Movements for Prosthetic Hands. Front. Neurorobot. 2016 (2016)