features

Now, it is my turn: for the implementation of deep neural network, we manually extracted some features from the filtered signals using them as input for the network. We chose time-domain features recommended by some relevant articles, that you can find in the references of the report. And they are the features mostly used for this type of classification. So, they are Mean Absolute value, Waveform length, Willison amplitude, auto-regressive model coefficients and root mean square. (For example, Willison amplitude is defined as the number of times that the change in EMG signal amplitude exceeds a threshold that we fixed to 0.5 mV, and auto-regressive model is a predicted model which describes each sample of the EMG signal as a linear combination of the previous samples plus a white noise error term. And we took from this feature the coefficients of the linear combination.)  
As we will see better for the LSTM implementation, we divided the whole signal in segments of equal length and extracted the feature values from each sensor putting them together in a row vector. So, we got a matrix of N rows which are all the segments with the corresponding values of features.

Dnn structure

For what concerns the structure we put a featureInputLayer at the beginning of the network that takes as input the features and applies a normalization and then 3 series of fully connected layer with its own activation function, for which we chose the sigmoid, a batch normalization and dropout layer concluding with a SoftMax and the classification layer.   
For the training options both networks, the DNN and the LSTM, use the same ones. Between them shuffle every epoch is surely one of the most important: it does a data scramble at each new epoch, needed to avoid bias during training and to make the training representative of the overall set of data. We tried different values for the size of batches and number of epochs, changing them depending on the simulation.   
The dataset has been divided randomly in 80% for training and 20% for testing.

Dnn accuracy

At the end we reached a very good accuracy of almost 93%, as you can see also from the confusion matrix. It shows that few movements have been predicted wrong.

((((Mean Absolute value is an average of absolute value of the EMG signal amplitude in a segment.

Rms is the square root of the mean square (the arithmetic mean of the squares of a set of numbers)

WL: is defined as cumulative length of the EMG waveforms over the time segment.

WAMP: This feature is defined as the amount of times that the change in EMG signal amplitude exceeds a threshold

Auto-regressive (AR) model is a prediction model that describes each sample of the EMG signal as a linear combination of the previous samples xi plus a white noise error term wi))))

Riccardo’s part

For the LSTM, as well as the DNN, we used a sliding window of time to divide the signals into segments. Since the LSTM cell is able to extract itself features from a provided time series, we don’t need to extract features, but we the maximum length of the time series the lstm can handle before being affected by the vanishing gradient problem is of maximum 100. Since one second is made of 4000 values, we decided to use a method similar to max pooling, where the segment is divided into 100 sub-segments and the maximum value is taken from each sub-segment.

The structure of the network is made of an lstm cell followed by a DNN structure, with the same layers seen before.

As for the DNN, the dataset was divided in 80% training and 20% validation. The LSTM is able to reach a final accuracy of around 92.6% for the validation.

To evaluate the utility of the 2 networks for a real application, we need to consider their robustness. By that we mean how well the networks perform when presented with a new subject or a new repetition of the same motion.   
To evaluate this, we trained the networks again, this time excluding from the training set either 1 subject or 1 of the 3 repetitions, and using them for the validation.  
The results are shown in the tables. We can see that both struggle when given a new subject, and the LSTM accuracy drops significantly when presented a new repetition of a movement, while the DNN only suffers a moderate decrease in the accuracy, which is also to be expected with the reduction of the size of the Training Set

To try to understand why this happens, we compared the same motion performed by the subjects. From a qualitative pov we can see that both the time domain and the frequency domain signals are noticeably different. This would explain the struggle with new patients. We are instead unsure about why the LSTM performs so poorly with new repetitions, while the DNN is still able to handle them.

In the future we believe that the robustness of the networks could be improved by adding extra features for the DNN, and by increasing and augmenting the dataset, from which both would benefit