Convolutional Neural Network for Hand Gesture Recognition Using Myo

Ian de Medeiros Esper M.Sc in Defence Engineering
Email: imesper@gmail.com

Abstract—

1 Introduction

Myo, an armband released by Thalmic labs, has revolutionized the arm electromyography (EMG) market, by offering a extremely low cost solution to read arm muscles EMG signals. One of the many usage possibilities of the armband is to make a human-computer interaction by hand gestures [1]. There are a widely gesture recognition input tools, such as cameras or stereoscopic cameras, data gloves, controllers, depth aware cameras [2] and EMG from arm muscles.

Several classification method has been published for hand gesture, as hidden markiov model (HMM), artificial neural network(ANN), Kalman filter, spectral collaborative representation based classification(SCRC) and support vector machine (SVM) are some of them as cited by [3], [4] and [5]. Being SCRC research [4] the only specific Myo based research on hand gesture recognition.

2 Hand Gestures and Myo

In this work there are a total of 6 possible hand gestures. The figure 1, shows a possible set of 6 hand gesture, just for illustrative purposes.

Fig. 1: Illustration of 6 hand gestures.

2.1 Signals

The Myo armband is connected to the computer through bluetooth technology. There are 8 EMG channels being sampled at a frequency of 200Hz. Each channel delivers an integer value to the computer.

2.2 Data

The data files was provided by Thalmic Labs in CSV format. Each of the 6 possible gestures had 2000 files with the following naming standard:

$$GestureM\_ExampleN.txt \quad (1)$$

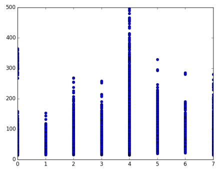
Where M is the gesture number and N is the example number. A random number of files from each gesture was plotted, so the respective distribution of the data could be observed as shown on 2.

The task to load this files everytime we execute the program is not trivial, so a python pickle file was adopted to data in order to speedup the execution of the function. The pickle files was saved with the following names:

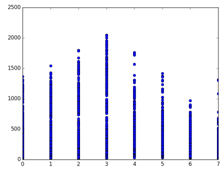
- *train.p* - Data ready for CNN
- *ytrain.p* - Y axis data for CNN
- *test.p* - Test data for CNN
- *ytest.p* - Y axis data for test data
- *data1.p* - Data from all gesture 1 files
- *data2.p* - Data from all gesture 2 files
- *data3.p* - Data from all gesture 3 files
- *data4.p* - Data from all gesture 4 files
- *data5.p* - Data from all gesture 5 files
- *data6.p* - Data from all gesture 6 files

3 Artificial Neural Network

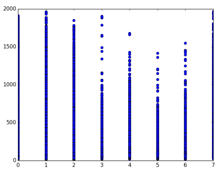
Inspired by the neuronal structure, illustrated in figure 3, the artificial neural network (ANN) are formed by interconnected simple and similar structures to process information [6] e [7]. Figure 4 shows a computational neuron.



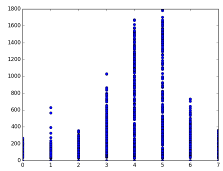
(a) Gesture 1



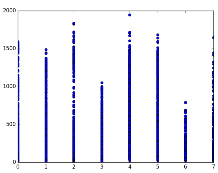
(b) Gesture 2



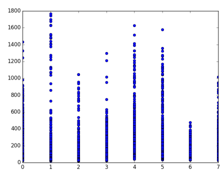
(c) Gesture 3



(d) Gesture 4



(e) Gesture 5



(f) Gesture 6

Fig. 2: Illustration of various images

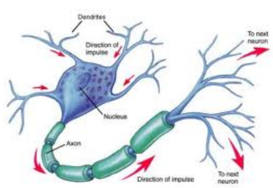


Fig. 3: Biological Neuron

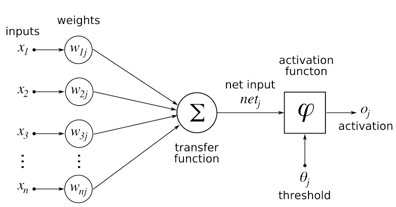


Fig. 4: Computational Neuron

Where x_j are inputs to neuron k. These inputs are multiplied by the synaptic weights w_{kj}, the sum of all products is the linear combiner output u_k, as seen in equation 2.

The overall output of the neuron y_k is the result of the activation function $\varphi(.)$ with parameters being the sum of the bias (b_k) and the output of equation 2, *i.e.* u_k, as shown in equations 3 [7].

$$u_k = \sum_{j=1}^{m} w_{jk} x_j \tag{2}$$

$$y_k = \varphi(u_k + b_k) \tag{3}$$

The ANN is a catenation of several neuron in layers as shown in figure 5. The main advantages of a neural networks are the capability to learn from data, the lack of need of a specialist, it can complete the gaps even with incomplete data, it is robust to noise and does not require a mathematical model.

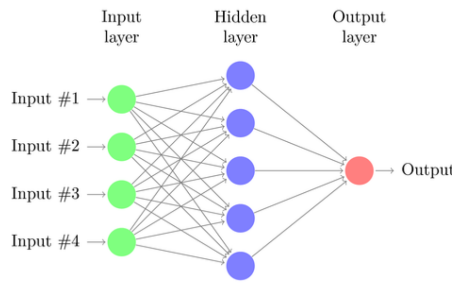


Fig. 5: Architecture of a Neural Network

The ANN are capable of executing many tasks as function approximation, learning, generalization, classification and others.

3.1 Convolutional Neural Network - CNN

A convolutional neural network is an artificial neural network, much like the one explained on previous section with convolutional and subsamplings layers connected to dense or fully connected layers, as shown in figure 6.

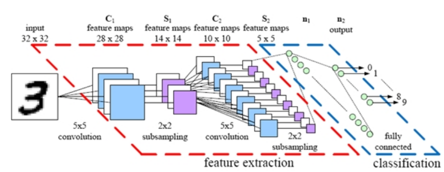


Fig. 6: Architecture of a Convolutional Neural Network

For classification of multi class system, is normally used a softmax activation function in the last layer with the total number of classes as the depth parameter.

4 Proposal

The objective of this work is to create and test a CNN for classification of hand gestures, where the input signals are acquired by Myo.

5 IMPLEMENTATION

The code was implemented in Python using keras and Theano framework. The structure of the cnn implemented is shown in 4.

$$
\begin{aligned}
InputData = & \\
& \Rightarrow Convolution2D(activation = tanh) \\
& \Rightarrow Convolution2D(activation = tanh) \\
& \Rightarrow Dropout \\
& \Rightarrow Convolution2D(activation = tanh) \\
& \Rightarrow Convolution2D(activation = tanh) \\
& \Rightarrow Dropout \\
& \Rightarrow Convolution2D(activation = tanh) \\
& \Rightarrow Flatten \\
& \Rightarrow Dense \\
& \Rightarrow Dropout \\
& \Rightarrow Dense \\
& \Rightarrow Output
\end{aligned} \tag{4}
$$

6 TESTS

There were some tests under the CNN classification method. Many more tests can be done by varying the following parameters:

- CNN kernel size;
- Dense layers depth;
- Batch size;
- Optimizer
- Learning rate
- Activation function

Epochs	Loss	Accuracy	Validation Loss	Validation Accuracy
1	0.9623	0.6551	4.4500	0.1867
25	0.2315	0.922	3.6577	0.31

TABLE 1: Results of CNN for Hand Gesture)

Results after some changes in configuration of the network. The convolutional layers was reduced, because it was probably overfitting the data.

Epochs	Loss	Accuracy	Validation Loss	Validation Accuracy
1	0.9444	0.6739	3.19	0.2267
25	0.3966	0.8779	2.2065	0.6067

TABLE 2: Results of CNN for Hand Gesture after making CNN smaller)

7 CONCLUSION

The convolutional neural network shows to be a very interesting deep neural network for hand gesture recognition. It is observable that it easily over fits the data, making a good result in training data a but a terrible results in testing data.

One testing that can be done in the future is to make the data unidimensional by stacking time domain rows in channel column and train the network with one dimensional convolutional layer. There are many others possibilities to test.

A lot research has to be done in order to decide if CNN has really a good fit for hand gestures recognition, but it sure seems like a possibility.

REFERENCES

[1] S. Mitra and T. Acharya, "Gesture recognition: A survey," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 37, no. 3, pp. 311–324, May 2007.

[2] B. Bansal, "Gesture recognition: A survey," *International Journal of Computer Applications*, vol. 139, no. 2, 2016.

[3] A. R. Sarkar, G. Sanyal, and S. Majumder, "Hand gesture recognition systems: A survey," *International Journal of Computer Applications*, vol. 71, no. 15, 2013.

[4] A. BOYALI.

[5] M. Rossi, S. Benatti, E. Farella, and L. Benini.

[6] D. Nguyen and B. Widrow, "Neural networks for self-learning control systems," *Control Systems Magazine, IEEE*, vol. 10, no. 3, pp. 18–23, April 1990.

[7] S. Haykin, *Redes Neurais: Princpios e Prtica*. Bookman, 2001.