Identifying Hand Gestures through Myographic Signals via ANN

Eduardo Coronado
Duke University

Sebastian Knigge
Duke University

Sam Voisin
Duke University

Yuan Zheng
Duke University

Abstract

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

1 Introduction

Recent advances in surface electromyographic signal (sEMG) recordingsystems and analytics methods have encouraged the use of sEMGs in human-machine interfaces to control exoskeletons and protheses; however, challenges remain.[1] Accurate classification of user movements is highly variable given the inherent noise of sEMG recording systems and per-user variability. In turn, this leads to problems downstream when attempting to convert these classifications into spatial directions (e.g. up, down, right, and left). Here, we aim to address the former challenge specifically. Our goal is to design and implement a light-weight unsupervised learning model via a multi-layered neural network to accurately identify six distinct hand gestures from sEMG data.

2 Methods

Our analysis workflow was comprised of 4 main phases: preprocessing, dimension reduction via PCA, modeling, and performance comparison as shown in Figure 1.

2.1 Preprocessing

We implemented a root means squared (RMS) envelope of 200 ms overlapping time windows at $100ms$ steps via the *biosignalEMG* R package[2] to remove some of the noise generated during sEMG signal collection. As shown in Figure 2 the preprocessing does help to get clearer signals.

[1] Sergey Lobov & Makarov (2018)
[2] J.A. Guerrero (2018)

32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, Canada.

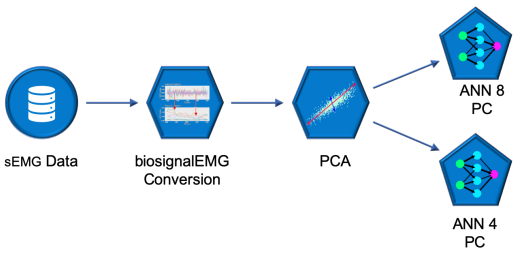
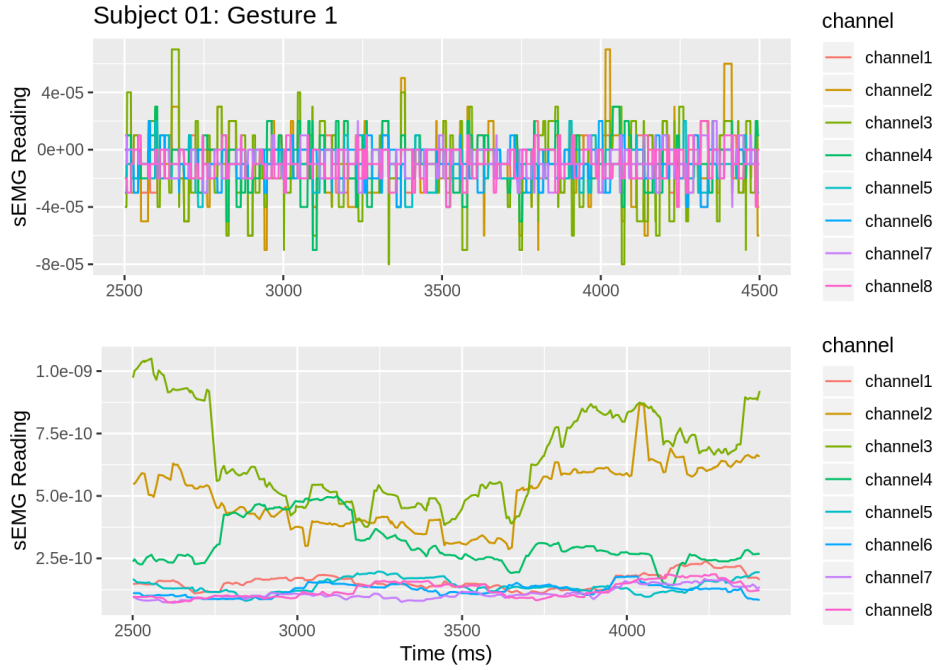


Figure 1: Schematic of overall workflow stages: preprocessing, PCA, modeling and performance comparison

Figure 2: Schematic of overall workflow stages: preprocessing, PCA, modeling and performance comparison



2.2 Dimension Reduction

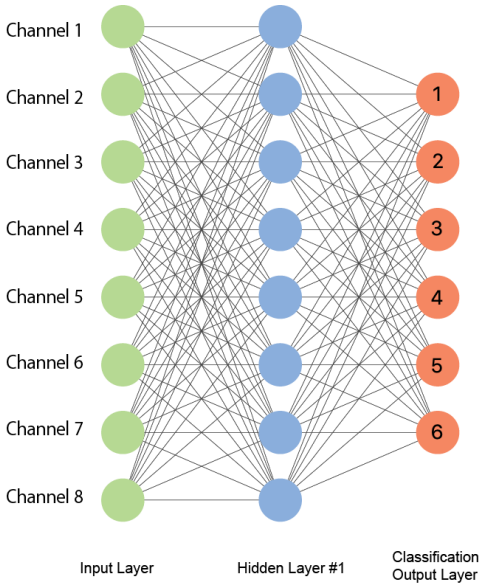
Dimension reduction was done via a principal component analysis (PCA) of the 8 distinct channels used to record sEMG data in order to reduce the training time of the neuronal network. We achieved this via the R *princomp* function that performs a spectral decomposition of the gesture design matrix.

2.3 Modeling

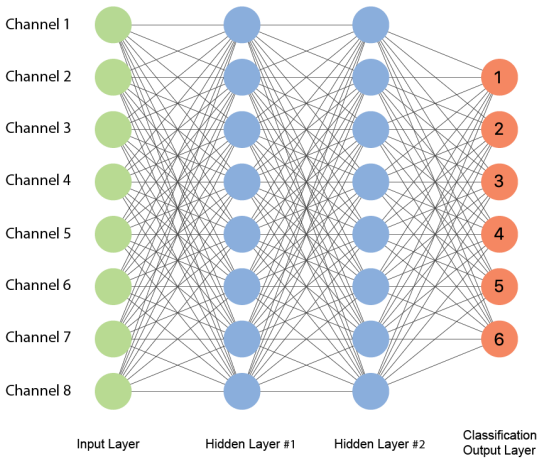
The final step of preprocessing requires adding "padding" to the end of each gesture's data matrix. This is done so that the vectors passed into the models are of equal dimension. We then developed two Artificial Neural Networks (ANN), one with all components and a second with a subset selected during the dimension reduction stage (Figure 3a). Each of these has a single hidden layer with eight hidden nodes having linear activation. These eight nodes feed into six output nodes with a soft-max activation function. This model was inspired by Lobov et al (2018).

To create the computation graphs for the ANN we used the R implementation of Keras[3] which relies on Google's TensorFlow engine for backpropogation. This allows us to add, edit, and remove layers from our models as needed during the development process. The flexibility of this modular design means we are able to prototype new models quickly.

[3]Daniel Falbel (2019)



(a) Schematic of input hidden and classification layers of ANN with all 8 channels as input



(b) Schematic of input hidden and classification layers of ANN with 4 principle components as input

Figure 3: Neuronal Net structures

Comment 1 *To assess the performance of each ANN, we used an* $80/20$ *split our data to generate a gesture-specific training and a testing sets. These were done automatically in the Keras package via random sampling. We then assessed the classifcation accuracy of each ANN using a categorical cross entropy metric which measures the Kullback-Leibler (KL) divergence between the true distribution of the response variable and the distribution of the predictions.*

3 Results

3.1 Principal Component Analysis

Using the spectral decomposition we found that only 4 of the 8 principal components in our dataset contributed meaningfully to variance in our response. From figure 3, we can see that these 4 components account for approximately 80% of the variability we are seeking to model.
We might be interested in compareable results, or a decision rule which is applicable to different data sets. We introduce a relative measure κ which can be used for the PCA of data sets with any dimension.

$$\kappa_i = \frac{proportional\ explained\ variance_i}{\frac{1}{\#\ of\ dimensions}}$$

One can define a decision rule, which is dependent on a threshold δ: *Include* $PC_i\ \forall\ \kappa_i > \delta$

For our data set we choose $\delta := 0.6$.

Figure 4: PCA Screeplot and cumulutive variance plot

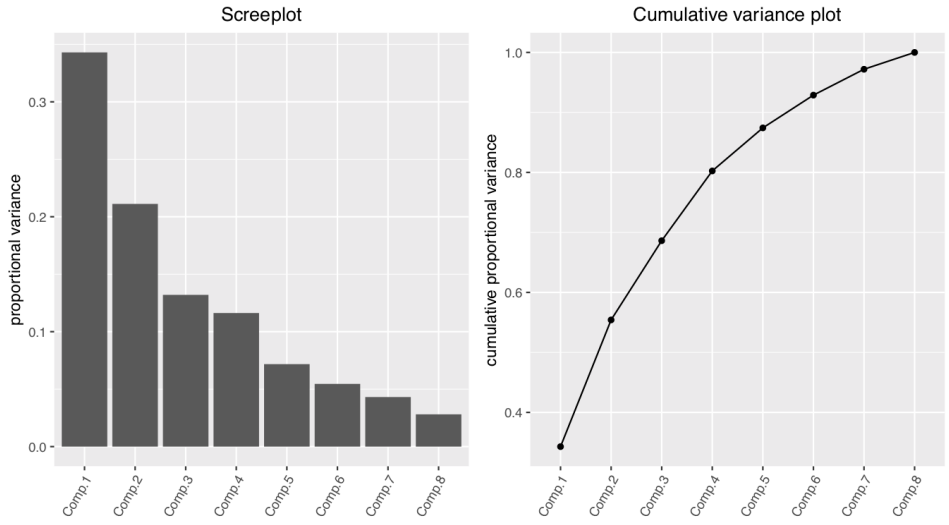


Table 1: PCA measures

	Comp. 1	Comp. 2	Comp. 3	Comp. 4	Comp. 5	Comp. 6	Comp. 7	Comp. 8
proportion of explained variance	0.343	0.211	0.132	0.116	0.072	0.055	0.043	0.028
cummulative explained variance	0.343	0.554	0.686	0.802	0.874	0.929	0.972	1
κ	2.744	1.689	1.056	0.930	0.574	0.437	0.345	0.225

By reducing the number of inputs from 8 smoothed signals to only 4 principle components, we were able to reduce overfitting in our model prior to ultizing any dropout layers. This reduction in overfitting decreased prediction accuracy in the validation data set only slightly see the *ANN Performance Comparison* section.

3.2 ANN Performance Comparison

– Predictive Accuracy Using 8 Smoothed Channels –
– Predictive Accuracy Using 4 Principle Components –

References

References

Daniel Falbel, e. a. (2019). Package ‘keras’ [Computer software manual].

J.A. Guerrero, J. M.-D. (2018). Package ‘biosignalemg’ [Computer software manual].

Sergey Lobov, I. K. V. K., Nadia Krilova, & Makarov, V. A. (2018). Latent factors limiting the performance of semg-interfaces. *Sensors*, *18*(1122).