sEMG Gesture Recognition With a Simple Model of Attention

David Josephs[1], Carson Drake[1], Che' Cobb[1], John Santerre[1], and Jennifer Graves

Master of Science in Data Science, Southern Methodist University, Dallas TX 75275 USA {josephsd, drakec, cobbc}@smu.edu

Abstract. Idk man

1 Introduction

Electrophysiological studies of the nervous system are the core area of research in clinical neurophysiology, where scientists attempt to link electrical signals from the body to real world effects. These studies include measuring brain waves (electroencephalography), comparison of sensory stimuli to electrical signals in the central nervous system (evoked potential), and the measure of electrical signals in skeletal muscles (electromyography). Electromyography is of particular interest to this paper. The nervous system uses electrical signals to communicate with the rest of the body. When a signal from the nervous system reaches a skeletal muscle, the myocites (muscle cells) contract, causing a physical motion. By measuring these electrical signals in a supervised manner, we can develop a link between signal and physical action. This connection yields many powerful uses, ranging from quantifying physical veracity to diagnosing neurodegenerative diseases. An example of the latter can be found in Akhmadeev et al. [1], where electromyographic (EMG) signals were used to classify Multiple Sclerosis patients from healthy control subjects with 82% accuracy.

Deep learning can be used to further improve the power and utility of the EMG analysis. A deep neural network is, in essence, a composition of neurons (regressors) that learns a functional mapping between two sets of data. By learning a mapping between EMG signal and physical effect, we can develop more sensitive and accurate models of what connects the two. This also allows us to use less intrusive measurement devices in studies and in the real world. The applications of this range from clinical trials and prognostication of neuromuscular diseases to gesture prediction in "brain-controlled" prosthetic limbs.

There are several factors which need to be considered in order to develop practically useful deep learning models for real time gesture prediction (myoelectric control)[1]. Arguably the most important of these is the amount of time

[1] Myoelectric control in this context means classifying a gesture well before it is completed, using myoelectrographic signals. These predictions need to be done fast enough for the user to feel that the gesture is being made as they decide to complete it.

before the model makes its first prediction (referred to as "prediction latency"). The absolute largest prediction latency for a model to be considered useful for myoelectric control lies between 250 and 300 milliseconds [4], [8]. Many recent papers claim excellent gesture prediction results, however they require time samples (windows) between 0.5 and 1.5 seconds long [7], [10]. In this paper, all windows are 260 milliseconds in length, following the precedent set by Allard et al. [3]. This leaves 40 milliseconds for any transformations done on the window, as well as the amount of time it takes for the model to make a prediction.

The next factor to consider when designing a model for myoelectric control is the number of gestures the model is actually capable of predicting. In many recent papers, models are evaluated using between six and eight gestures. This presents an interesting theoretical result, but does not hold much practical value. Thus, we evaluate our model on 53 gestures within 3 subclasses: 12 fine finger movements, 17 wrist movements, 23 functional movements (such as grasping), and rest (present in all subclasses).

It is also important to account for data collection. As the intention of this research is to provide a practical myoelectric control tool for mass use in real hardware (prosthetic limbs, robotic arms, etc.), we elected to use a cheap sEMG sensor, the Thalmic MYO armband (referred to as "MYO armband") [9]. The MYO armband collects data at a relatively low frequency, however it costs about $100 and has several other qualities which make it an ideal tool for this research. The most important of these qualities is standardization. In clinical sEMG research, sensors are placed on a per patient basis in exact locations, determined by medical professionals. This is not feasible in the case of prosthetics, as requiring a doctor be present to put your hand on every day would be far from convenient. In contrast, the MYO armband automatically contracts to more or less the same points on every arm. This makes it far more practical, and it far simpler to imply the model will generalize to more subjects.

The final two factors to consider with respect to model and experimental design are related to the evaluation and generalization of the model. A myoelectric control model needs to bear two qualities: robustness over time and generalizability from subject to subject. Over time, the armband will shift, the user will become fatigued, and actions will change. For a model to be useful for mass production in prosthetics and robotic arms, it will also need to work on new people. While it is possible to train a model on a new person, even on amputees, it is preferable for the model to work immediately [2].

In this paper, we propose a novel attentional architecture for sEMG recognition and myoelectric control. We demonstrate the model's validity on the 53-class NinaPro DB5 [5]. We also compare different techniques for dealing with the inherent class imbalance in sEMG, a synthetic data based approach (augmentation), an undersampling based approach, and ,w

We also present and discuss several methods for dealing with the class imWith this, we also present a procedure for dealing with class imbalance in myoelectric data, a novel data augmentation technique, and a variety of preprocessing techniques. We also propose the usage of the 5th NinaPro MYO database as a standard benchmark dataset for myoelectric control [5]. A common benchmark dataset will help current results be easily comparable, which is a large issue facing myoelectric control today. Our evaluation procedures will also help evaluate the issues brought up in [11], in which practical results do not match theoretical results due to overfitting.

2 EMG and sEMG signals

* Discuss EMG signals, converge to sEMG

2.1 Challenges

2.2 Preprocessing

* Butterworth filters
  * moving average and other fast transforms on rectified data

2.3 Augmentation

* electrode shifting
  * SNR spectrum sampling

3 Deep Learning for sEMG Signal

All discussed in context of sEMG and EMG (obviously discuss generalization between the two)

3.1 Recurrent Neural Networks

In this research we utilize a special class of neural network known as a *recurrent neural network* (RNN). Before we discuss a recurrent neural network, let us first formalize our definition of a standard, feedforward deep neural network (DNN). A DNN is a network of neurons (small learners which learn simple, nonlinear functions), which are composed and trained in a way such that they learn a function which maps one set of data (for example, an image or a time series) to another set of data (for example, a classification label). Formally, this can be expressed as:

$$y(x) = g\left(\sum_{i=0}^{d} W_{k_i} x_{k_i}\right) \tag{1}$$

This can be read as follows: given data with d dimensions, the neural network sums up the product of a set of *linear weights* (W) and the data at dimension d. It then multiplies this by a nonlinear function, g. This nonlinearity allows the neural network to build incredibly powerful functional representations of the relationship between x and y.

Although a DNN can learn powerful relationships, it is somewhat unintuitve for order-dependent data, such as sequences or time series. This is where an RNN becomes incredibly useful. An RNN not only learns an observation-by-observation functional mapping, but can also map sets of observations (for example, windows of 8-channel electromyographic signal) to labels. They can also map sets of observations to new sets of observations, which is especially important in Natural Language Generation and time series forecasting, however that is beyond the scope of this paper. An RNN accomplishes this in the following manner. First, instead of inputting a single observation, observations are fed in in a while loop. This is not quite enough to learn long term dependencies within the time windows however. In order to do this, during the while loop of n timesteps, the RNN must also record *state*, that is the RNN must have some sort of memory of previous observations. To do this, the RNN learns a functional mapping which is dependent on its own previous states. This can be expressed mathematically as follows. First, let us define the weights which the RNN uses to update its state, W_s:

$$W_s = (I - (\mathit{\Delta} T)\, A)^{-1} \tag{2}$$

Where I is the identity, $\mathit{\Delta} T$ is is the discrete change in time between observations in the time window, and A is a linear, block-circulant matrix. We can then perform a linear transformation on Equation 2 to get the weights for the input signal, W_x, and bias θ:

$$W_x = (\mathit{\Delta} T)\, W_s C \tag{3}$$

$$\theta = (\mathit{\Delta} T)\, W_s \phi \tag{4}$$

Where C is another block-circulant, linear transformation matrix, and *phi* is a learned vector of biases for the RNN (similar to the intercept in regression). This allows us to define the state of the function at timestep n as:

$$\vec{s}\,[n] = W_s \vec{s}\,[n-1] + W_x \vec{x}\,[n] + \vec{\theta_s} \tag{5}$$

However, this definition of an RNN has an issue: the state term $(W_s \vec{s}\,[n-1])$ will grow or shrink without bound, leading to a highly unstable model which in all likelihood fails to converge. To bound it while preserving all of the information, we perform a nonlinear transformation: the hyperbolic tangent. We define a new term, $\vec{r}$, which represents the recurrent portion of an RNN:

$$\vec{r}\,[n] = \tanh\left(\vec{s}\,[n]\right) \tag{6}$$

$$W_r = (\mathit{\Delta} T)\, W_s B \tag{7}$$

$$\tag{8}$$

Where W_r represents the recurrent weights of the model, and B is a block circulant linear transformation matrix. We can then rewrite Equation 5 as the classic RNN equation:

$$\vec{s}\,[n] = W_r \vec{r}\,[n-1] + W_x \vec{x}\,[n] + \vec{\theta_s} \tag{9}$$

While the simple recurrent neural network can learn powerful functional mappings of time-dependent data, it suffers from difficulties due to training issues. If a single eigenvalue of W_r lies outside of of the range $(0, 1)$, the gradient used to train the RNN will either explode or decay exponentially, causing the RNN to stop learning [6].

In order to combat this, the Long Short Term Memory (LSTM) is introduced. The LSTM

Both of these derivations are based off of the comprehensive work done in [6]

3.2 Attention

3.3 Deep Transfer Learning

3.4 General evaluation

4 Training Procedures

5 Novel RNN based model

6 results

6.1 Novel RNN

7 Conclusion

References

1. Akhmadeev, K., Houssein, A., Moussaoui, S., Høgestøl, E.A., Tutturen, I., Harbo, H.F., Bos-Haugen, S.D., Graves, J., Laplaud, D.A., Gourraud, P.A.: Svm-based tool to detect patients with multiple sclerosis using a commercial emg sensor. 2018 IEEE 10th Sensor Array and Multichannel Signal Processing Workshop (SAM) pp. 376–379 (2018)
2. Atzori, M., Gijsberts, A., Castellini, C., Caputo, B., Hager, A.G.M., Elsig, S., Giatsidis, G., Bassetto, F., Müller, H.: Electromyography data for non-invasive naturally-controlled robotic hand prostheses. Scientific data **1**(1), 1–13 (2014)
3. Côté-Allard, U., Fall, C.L., Drouin, A., Campeau-Lecours, A., Gosselin, C., Glette, K., Laviolette, F., Gosselin, B.: Deep learning for electromyographic hand gesture signal classification using transfer learning. IEEE Transactions on Neural Systems and Rehabilitation Engineering **27**(4), 760–771 (April 2019). https://doi.org/10.1109/TNSRE.2019.2896269

4. Hudgins, B., Parker, P., Scott, R.N.: A new strategy for multifunction myoelectric control. IEEE Transactions on Biomedical Engineering **40**(1), 82–94 (Jan 1993). https://doi.org/10.1109/10.204774
5. Pizzolato, S., Tagliapietra, L., Cognolato, M., Reggiani, M., Müller, H., Atzori, M.: Comparison of six electromyography acquisition setups on hand movement classification tasks. PLOS ONE **12**, e0186132 (10 2017). https://doi.org/10.1371/journal.pone.0186132
6. Sherstinsky, A.: Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. CoRR **abs/1808.03314** (2018), `http://arxiv.org/abs/1808.03314`
7. Simão, M., Neto, P., Gibaru, O.: Emg-based online classification of gestures with recurrent neural networks. Pattern Recognition Letters **128** (08 2019). https://doi.org/10.1016/j.patrec.2019.07.021
8. Smith, L.H., Hargrove, L.J., Lock, B.A., Kuiken, T.A.: Determining the optimal window length for pattern recognition-based myoelectric control: Balancing the competing effects of classification error and controller delay. IEEE Transactions on Neural Systems and Rehabilitation Engineering **19**(2), 186–192 (Apr 2011). https://doi.org/10.1109/tnsre.2010.2100828, `http://dx.doi.org/10.1109/TNSRE.2010.2100828`
9. Visconti, P., Gaetani, F., Zappatore, G., Primiceri, P.: Technical features and functionalities of myo armband: An overview on related literature and advanced applications of myoelectric armbands mainly focused on arm prostheses. International Journal on Smart Sensing and Intelligent Systems **11**, 1–25 (06 2018). https://doi.org/10.21307/ijssis-2018-005
10. Xie, B., Li, B., Harland, A.: Movement and gesture recognition using deep learning and wearable-sensor technology. In: Proceedings of the 2018 International Conference on Artificial Intelligence and Pattern Recognition. pp. 26–31 (2018)
11. Yasen, M., Jusoh, S.: A systematic review on hand gesture recognition techniques, challenges and applications. PeerJ Computer Science **5**, e218 (2019)