SMU Data Science Review

Volume 3 | Number 1 Article 9

2020

sEMG Gesture Recognition With a Simple Model of Attention

John D. Josephs Jr
Southern Methodist University, josephsd@smu.edu

Carson Drake
Southern Methodist University, drakec@smu.edu

Che Cobb
Southern Methodist University, chec@smu.edu

John Santerre
SMU, jsanterre@smu.edu

sEMG Gesture Recognition with a Simple Model of Attention

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

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

Abstract. This paper presents a novel method for fast classification of surface electromyography(sEMG) signals, using a simple model of attention. The brain transmits electrical signals throughout the body to contract and relax muscles. sEMG measures these signals by recording muscle activity from the surface above the muscle on the skin. By classifying these signals with low latency, they can be used to control a prosthetic limb using an amputee's brain power. On a difficult, industry benchmark sEMG dataset, the proposed attentional architecture yields excellent results, classifying 36 more gestures (53 in total) with about 20% higher accuracy (87% overall) than the current standards in the field. These results have direct and immediate application in the fields of robotics, myoelectric control, and prosthetics.

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 myocytes (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. [2], 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. This paper focuses on the latter application. Current state of the art myoelectric prosthetic limbs are capable of detecting and performing between 7 and 18 motions [16] [25]. While this represents a significant increase in the quality of life of an amputee, there remains a large amount of room for improvement. Most of the potential for improvement does not lie in the robotics themselves, which are fairly robust, but within the device which maps brain signal to motion of the hand. The aim of this paper is therefore to build a highly accurate gesture classifier using EMG signal, capable of classifying a broad range of gestures, improving the quality of life of amputees as much as possible.

Extending from this initial goal, there are certain metrics which must be met for the classifier to be deemed "useful" for an amputee. First, it must be fast. If there is a large degree of latency from thought to hand motion, the user will simply not use the device. Thus, it is important to determine the time window in which a prediction must be made. The absolute largest prediction latency for a model to be considered useful for myoelectric control lies between 250 and 300 milliseconds [18], [33]. For this paper, the precedent set by [12] was followed, with a 260 millisecond prediction window. Another key consideration for the classifier is generalizability. Within the field of myoelectric control and gesture recognition, there are two important types of generalizability, intra-subject and inter-subject [31]. Intra-subject generalizability indicates that the model is robust to personal elements, such as muscle fatigue. In contrast, inter-subject generalizability refers to the ability of the model to generalize to new people. This paper focuses on intra-subject generalizability, as it is important that the device keep working for extended periods of time, while models can be fit to and personalized for amputees and non-amputees alike [4].

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 [27]. 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 a loss-based approach. All methods are evaluated on an intra-subject basis using the same holdout samples, yielding promising results. This architecture represents our main contribution to the sub-fields of myoelectric control, time series classification, and prosthetics, however the inclusion of ideas and techniques from other domains, such as focal loss for class imbalance, layer normalization, and the novel data augmentation technique used in this paper represent further contributions.

The structure of this paper is as follows. In Section 2, an overview of sEMG and the dataset is provided for the reader's convenience. In Section 3, a brief overview of sEMG classification with machine learning, and then with deep learning is given. Section 4 highlights the preprocessing techniques used on the data, as well as a novel data augmentation technique, and Section 5 outlines modeling techniques and procedures, including the attentional architecture, novel and

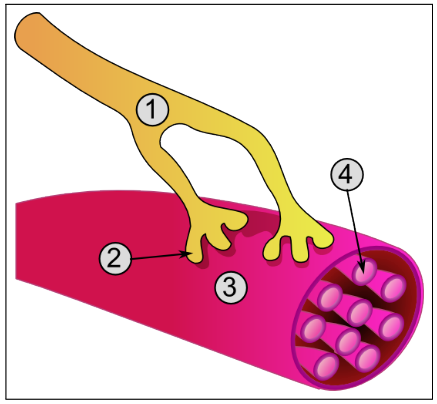


Fig. 1: A motor unit, consisting of: *1* an Axon, *2* a junction site, *3* a Myocyte, and *4* Myofibrils. The Axon sends electrical signal through the junction cite, causing the all of the myofibrils (and thus the entire myocyte) to contract.

powerful training techniques, and the computer vision inspired loss function. In Section 6, the results are analyzed extensively, and Section 7 concludes the paper.

2 Surface Electromyography

Before discussing the mechanics of the proposed model, it is important to first discuss electromyography more closely. As mentioned in the Introduction, electromyography is the measure of electrical signals from the brain in the rest of the body. To be more specific, the body controls the forces in the muscles using electrical signals, via units called *motor control units*, as seen in Figure 1. By measuring the amplitude and frequency of these signals, the strength or veracity of the motion and the type of motion can be determined. These signals are recorded in units of *Motor Unit Action Potential* (MUAP), either through internal electrodes or electrodes on the skin. This paper is particularly focused on the measure of signals across the skin, or surface electromyography (sEMG). While sEMG is not quite as powerful as true EMG (e.g., the signals are weaker), it is far more practical and scalable, and far less disruptive, invasive, or sensitive to change. In this paper, data are collected using two MYO armbands [37], on the upper arm of the subject (just above the elbow), as seen in Figure 2. Intuitively, the idea is to collect the signal where the sensors are, and classify them with enough speed that it feels as if the hand were being moved on its own.

2.1 The NinaPro Database

The data in this paper comes from the NinaPro Database, specifically database number 5 [27]. Within this database, there are 52 unique motions measured,

Fig. 2: The MYO armbands are placed just above the elbow [10].

as well as rest, collected over 10 subjects. Each subject does each exercise 6 times, with 3 seconds of rest between each repetition. The signal from these are collected at the frequency of the MYO armband, 200 Hertz. Figure 3 displays the gestures performed by each of the subjects.

This is one of the big challenges and benefits of the MYO armband. Low frequency sEMG is very sparse in information, and thus difficult to classify (especially relative to expensive HD sEMG sensors), however this also means that the MYO armband is cheap and accessible to the average consumer. The other benefit of MYO's affordability and sparsity can be inferred from Figure 2. The inexpensive sensor constricts the arm and stays in place. Because the signals are weaker, slight shifts in the electrode do not have as much of an impact on the underlying signal (as the sensor is not incredibly sensitive). This means that the MYO armband is simpler and more robust for use by the general public.

The goal of this paper is to develop an accurate mapping between MUAP in the upper arm and the motion of the hands, in order to help amputees have access to reliable smart prosthetics.

3 sEMG Classification

In order to justify the need for deep sEMG classifiers, it is important to briefly discuss why shallow classifiers are inappropriate. Shallow classifiers (and really all of classical machine learning) relies on *feature engineering*, or the creation of robust features which represent the data in a precise manner [26]. This is very powerful, but it also comes with the assumption that the same fundamental process is generating all data points, that is the same feature set that works at time a for person x works at time b for person y. However, in the case of

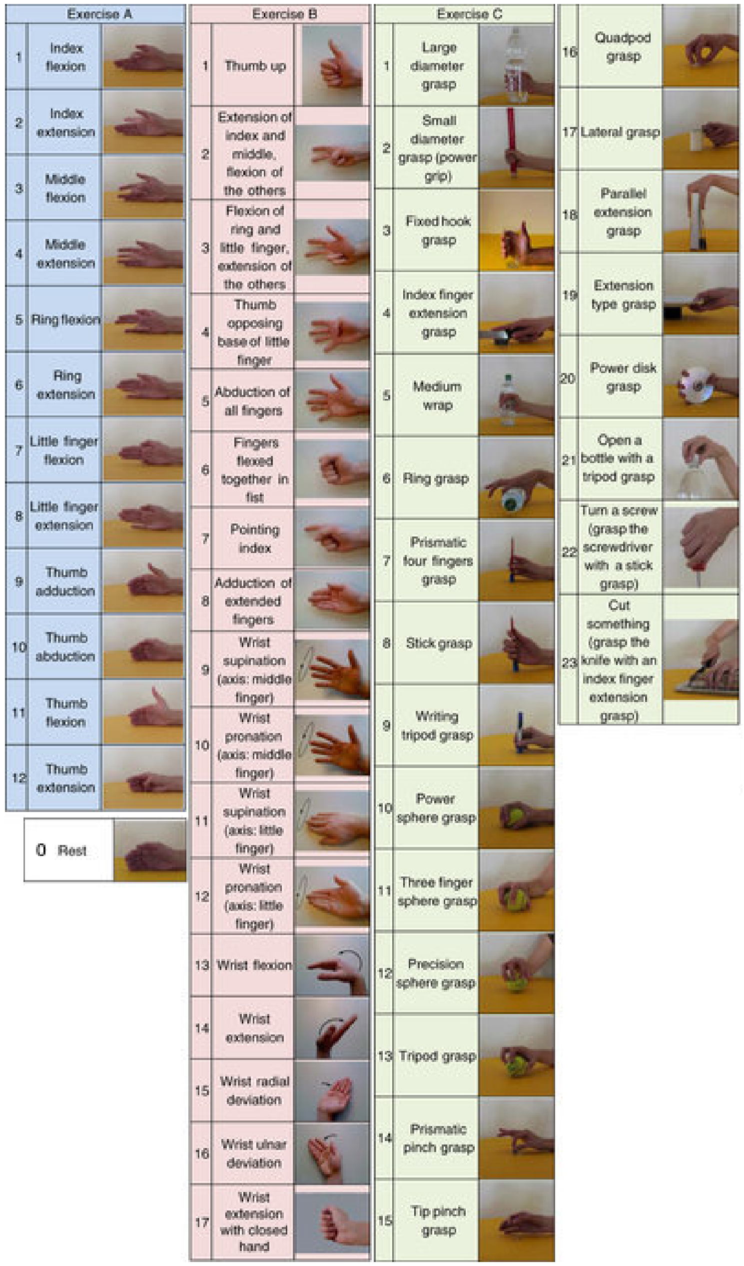


Fig. 3: The gestures contained in the NinaPro 5 dataset [5]. This paper focuses on Rest, Exercise A, Exercise B, and Exercise C

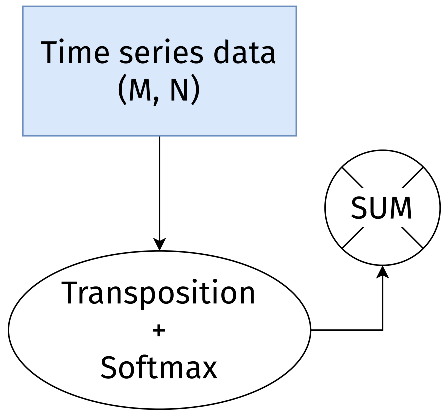


Fig. 4: A simple model of temporal attention.

surface electromyography, all brains and bodies are completely unique, which means that a classical, engineered feature set is neither robust between subjects or between repetitions (over time) [30]. In order to build robust models, intricate ensembling techniques with deep analysis and feature engineering are employed [32]. Therefore, the focus of this paper is to build a simple, powerful, model, which is able to *learn features from the data*, which are more robust over time and over subjects, and easily implemented in embedded hardware.

3.1 Deep Learning for sEMG Classification

Much of the previous work in sEMG classification (and time series classification in general [13]) with deep learning revolves around convolutional neural networks. In this subsection, notable and influential recent work will be highlighted. [3] very clearly demonstrated the power of convolutional neural networks with respect to sEMG classification, namely their flexibility and ease of use (due to feature learning vs feature engineering). This work is expanded in [12], in which the authors use convolutional neural networks, wavelet transformations, and deep transfer learning to slightly improve on the results highlighted in [3]. In [38], the authors proposed a multi-stream convolutional model, yielding 85% accuracy on the NinaPro DB1 [6]. [39] explores beyond the world of CNNs, testing a convLSTM architecture, which yielded poor results on NinaPro DB5, however this does not mean long term memory is a bad idea, and is a promising alternative to pure convolutions. The idea of both recurrent neural networks/long term memory and transfer learning is investigated in [20], in which the authors use a multistage training technique in order to adapt a recurrent neural network to dynamic sEMG data.

This paper instead explores a novel direction for both time series classification in general and for sEMG classification in particular: attention. In the last few years of language modeling, attentional models have proven more and more utilitarian, managing to solve problems of long term memory (which are commonplace in time series and language) while also managing to avoid the training and gradient issues of recurrent neural networks. Attention is a flexible, intuitive mechanism with lots of untapped application. In this paper, the simple, feed-forward model of attention described in [29] is expanded upon. The attention mechanism used in this paper is shown in Figure 4.

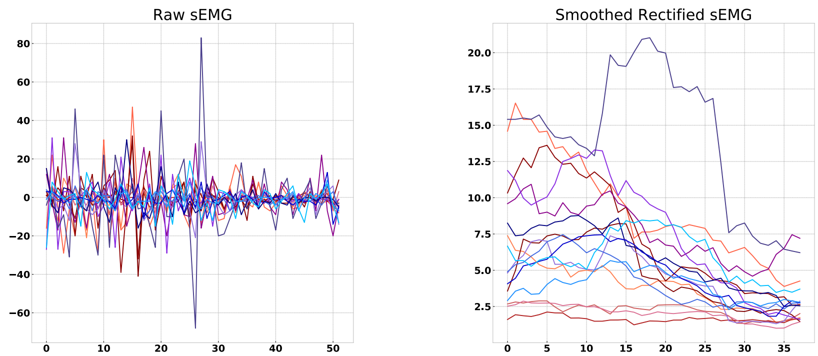
The mechanism works as follows: First time series data, an (M, N) matrix (where M represents timesteps and N represents features or channels). This matrix is transposed into an N by M matrix. In this matrix, an observation at a single row represents all timesteps of that feature or channel. For example, row x represents the time series produced by the feature $N = x$. This matrix is then fed in row by row to a standard, feedforward layer, using the softmax activation function. Thus, for each timestep of each feature, an importance (referred to as an "attention score") is calculated. Next, these are summed across time, producing a single number per feature, which represents the overall importance of the feature within that sample. As per [29], this simple mechanism, akin to a parametric time weighted average, actually successfully solves long term memory problems where order is not very important (or in this case, where memorization is to be avoided). This represents a promising new direction for time series classification as a whole and for sEMG recognition and myoelectric control in particular, as the order of observations does not necessarily matter, but the pattern does.

4 Data Processing

There are two considerations for preprocessing sEMG data for use in a prosthetic limb or other smart robotics. First, the preprocessing method must be fast and simple, as it needs to run with near zero latency on edge hardware (for example in a prosthetic hand). Second, the goal of using a deep neural network is to *learn* robust features. Therefore, preprocessing methods must be simple, computationally efficient, and maximize information density. The data (collected at 200 Hz), is sampled in time windows of 260 milliseconds (or 52 time steps), following the precedent set in [12]. One window of raw data can be seen in Figure 4a.

This raw data represents what is collected by the two MYO armbands. Each armband collects 8-channel signal across the arm at 200 Hz. To reduce powerline interference, the armband filters the data with a notch filter, minimizing the effects of large nearby electronics (such as power lines). Next to further refine and extract useful information from the signal data, the sEMG channels are rectified (in layman's terms, take the absolute value of). In [24], it was shown that rectification significantly increased the availability of information pertaining to the firing rate (temporal activity and pattern) of the motor units producing the signal.

After increasing the information density, the next important preprocessing step is to refine and distill the information. First, the packet of rectified signal uses a 20 Hz high-pass Butterworth filter to remove low frequency artifacts



a: Raw sEMG

b: Processed sEMG

Fig. 5: Raw and processed sEMG. the x and y axes represent time and motor unit potential respectively.

(aliases), typically caused by the sensor shifting. Artifacts can obscure and interfere with the relevant signal. Finally, the data is processed using a moving average transform. The smoothed average signal more clearly correlates with muscle activity [34]. The results of this preprocessing can be seen in Figure 4b.

One of the great hurdles of utilizing deep learning is its reliance on large, robust sets of train data, and its tendency to overfit to the said data. In order to learn more robust mappings, without collecting a significantly larger amount of data, synthetic data can be generated, in order to allow for a more robust and balanced training set. This helps prevent the deep learning algorithm from simply memorizing the training data. In this study, a novel data augmentation technique was employed, as seen in Figure 6. The methodology for the data augmentation is a simple random sample of a spectrum of different signal-to-noise ratios (SNR). A processed series is fed into the augmenter, and then random noise, at a given signal to noise ratio in magnitude is added. The signal to noise ratio for a realization is calculated at random as well, with linearly increasing likelihood as SNR increases. This allows the model to be trained for an extended period, without ever seeing the same observation twice.

5 Modeling

In this section, the overall architecture of the model is discussed, and then the novel training techniques used are outline, for the sake of reproducibility. The proposed model consists of three distinct phases: a convolutional preprocessor, an attention layer (as discussed in Section 3), and a classifier. The the various phases data processing, model design, training, and evaluation are implemented in Python, leveraging Keras [11], Tensorflow 2.0 [1], and Scikit-Learn[9].

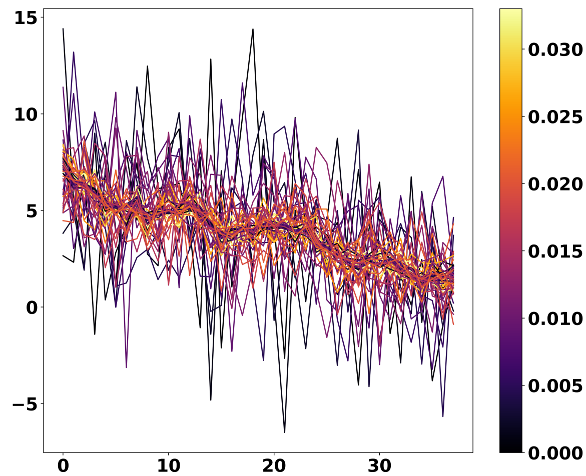


Fig. 6: Data augmentation scheme, highlighted on a single channel of a single sample. Brighter realizations are more likely than dim ones.

As seen in Figure 7, the first stage of the model consists of a single, 1-dimensional convolutional preprocessing layer. This layer takes in the processed time series data, and filters them with a kernel size of three. That is, every three timesteps is filtered down into a single value (for each channel). This is done for each timestep, with 128 filters, resulting in a $(M, 128)$ matrix (where M is the length of the time series). The output of this layer, and the output of all subsequent layers, are normalized across features (layer normalization) [7]. Layer normalization differs from batch normalization in that it normalizes across features, instead of across batches. Mathematically, consider a batch with dimensions $\mathbf{(samples, timesteps, features)}$. When batch normalization is applied, the weights are normalized using summary statistics across the *samples*, while with layer normalization, the summary statistics used to normalize the weights are calculated across the *features*. This holds true for both lower and higher dimensional data as well.

This matrix is then fed into the feed-forward attention mechanism described in Figure 4, resulting in a feature vector of 128 values. These are fed into the classification network, which consists of a set of standard, feed-forward layers. The architecture of the classifier is based off of the architecture proposed in [17], and was chosen due to its ability to effectively represent complex data. The output of this network is softmaxed in order to produce the final classifications. Each layer of the classification network is regularized using dropout.

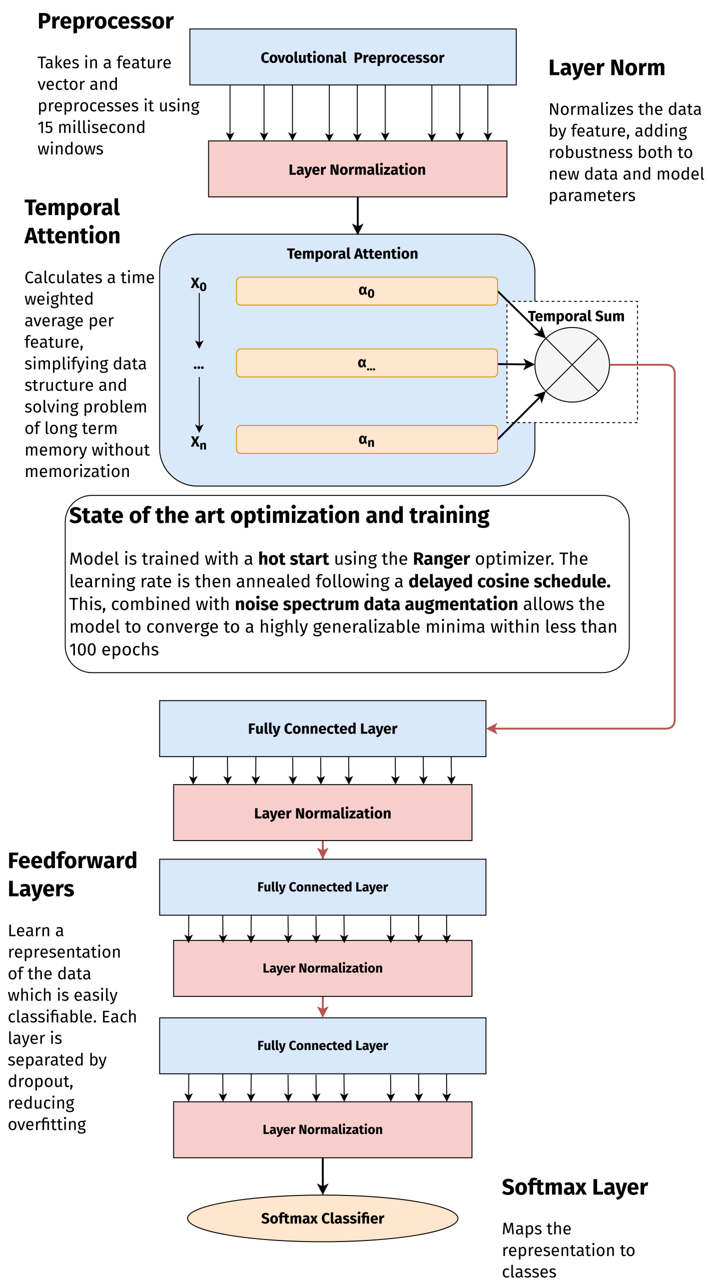


Fig. 7: The proposed architecture. Red arrows represent dropout connections.

5.1 Training

Several state-of-the-art training techniques were employed in this model. First, instead of using the Rectified Linear Unit(ReLU) activation function, all layers use the "Mish" activation function [23]. The Mish activation function is slightly less regularized then ReLU, thus yielding results slightly faster (and avoiding the "dead neuron" problem). Instead of optimizing the network with Adam or SGD, the model was optimized using the Ranger optimizer. The Ranger optimizer consists of two components: Rectified Adam (RAdam) and Lookahead. The RAdam algorithm represents an improvement over the Adam optimizer in that it does not require a "warm up period", which Adam is notorious for needing. This allows the model to be trained to an optima much faster [22]. The Lookahead algorithm works in conjunction with a primary optimizer. The primary optimizer calculates weights as it normally does, and then the Lookahead optimizer explores the loss landscape near the calculated weights. This allows for even faster convergence to an optima [40]. The combination of Lookahead and RAdam is the Ranger optimizer used in this paper [35]. Due to the very hot start of this combination (Mish and Ranger), the model learns very quickly. To further speed up training, the learning rate followed a delayed cosine annealing schedule. For the first 5 epochs, the model trains at a very high learning rate, and is subsequently annealed over the course of 50 epochs to a very low learning rate. This allows the model to quickly propel itself to a flat optima, and then work towards the bottom. This "hot start" methodology allows the model to quickly converge to a robust local minima [19].

A significant issue in sEMG classification is the large class imbalance present in the data. Most of the time, the hand is at rest. In the NinaPro 5 dataset, this means that there was over 30 times rest as any of the other 53 classes (and this also means that a non movable prosthetic hand would be about 60% accurate). Three methods were tested to combat this imbalance. First, undersampling the majority class (rest), so that it would have the same likelihood of occurring in the training dataset as the other classes yielded decent results. A similar approach, augmenting all the minority classes to match the majority class, was tested, however the slight benefit over undersampling was far outweighed by the prohibitive training cost of this method. Finally, a loss-based method was tested, using focal loss [21]. Focal loss is calculated in almost exactly the same manner as categorical cross entropy, except it gives a lower importance to well aligned (easy to classify) samples, and raises the importance of difficult samples. This yielded by far the best results.

6 Results

All models followed a training-validation-test split consisting of all 53 gesture classes, but across different repetitions. The repetition split schedule consisted of repetitions first through third for training set, fourth and fifth for validation set, and the final, sixth, repetition for the test set. This section introduces the evaluation metrics used in this paper for evaluating model performance, provides

Table 1: Results compared to other influential papers

Model	Window Size	Accuracy	Number of Gestures
Attentional Network, Focal Loss	260 ms	87%	53
Shen et al, 2019	200 ms	75%	53
Allard et al, 2019	289 ms	68%	17
Simao, 2019	1.5 s	90%	7
Atzori, 2016	150 ms	66%	17

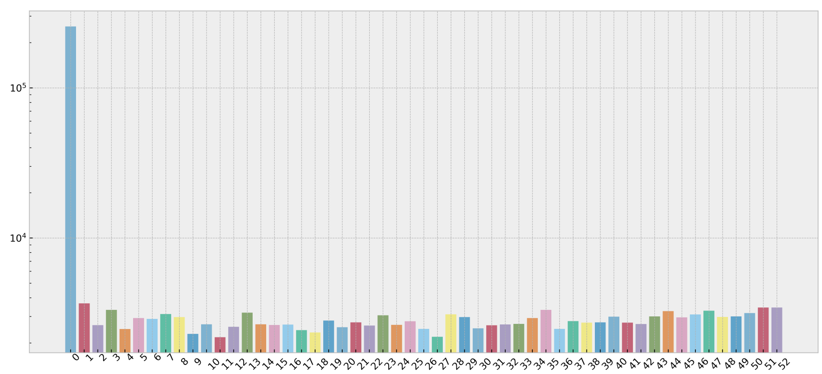


Fig. 8: Count of each class, it is apparent there is a large class imbalance

context to communicate the utility of each model, and finally compares and contrasts the results of this paper with both typical baseline models as well as other results relevant to the current state of sEMG gesture classification.

6.1 Empirical Analysis of Results Considering Class Imbalances

As each class is equally important, regardless of count, we would like to use a metric that is less reliant on counts. For this, we will use the Matthews Correlation Coefficient (MCC)[8]. The MCC calculates, in essence, the correlation coefficient between two sets of classes, as seen in Equation 3. This means that a value of 1 represents perfect prediction, a value of 0 represents completely random predictions, and a value of -1 represents inverse predictions. Because it relies on correlations between the groupings, and accounts for true positives, false positives, true negatives, and false negatives, it is completely robust to class imbalances. The overall MCC for the model on the full dataset, including rest is 0.78. This indicates that the model is a strong predictive model, and is generally correct, other than some bad samples, as it is relatively close to 1.

Table 2 shows more detailed diagnostics of the classification model. It shows 4 metrics, precision, recall, f1-score, and support. Precision is the ratio of times a class was predicted correctly to the total number of times that class was predicted. Recall represents the number of times a class was predicted correctly to the number of occurrences of that class, and F1-score is the harmonic average of precision and recall. Support is simply the number of occurrences of that class. It is apparent that the model performed very well on rest, however it struggled with some of the other motions. In order to make the model more useful to amputees, it may be helpful to remove these problem classes. The power of this technique can be observed using the balanced accuracy metric. The calculation of accuracy, given a confusion matrix C can be seen in Equation 1, while Equation 2 shows the calculation of balanced accuracy.

$$acc = \frac{\sum \operatorname{diag}(C)}{\sum_{i=0}^{m} \sum_{j=0}^{m} C} \tag{1}$$

$$acc_{\text{balanced}} = \sum \frac{\sum \operatorname{diag}(C)}{\sum_{j=0}^{m} C} \tag{2}$$

$$\text{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{3}$$

Intuitively, in balanced accuracy, scores are calculated given the support of that class, while normal accuracy calculates scores given the total number of observations. The balanced accuracy of the model predicting all of the classes is 70.4%. This can be seen with other references in Table 3.

In order to provide simple yet meaningful context to the model evaluation metrics, the upper half of Table 3 demonstrates the statistical performance of some generic “Dummy”, baseline, classifiers. In addition to providing useful baseline to compare and contrast the model's metrics, these baseline classifiers help illustrate relevant concepts specific to evaluating sEMG gesture classification models. This upper half also powerfully demonstrates the use of the MCC metric, as it is far more conservative and yields near-zero scores for bad results. The lower half of the table shows the results grouped by exercise type, where there is clear inflation of the results from the class imbalance in the accuracy score, while balanced accuracy is more robust to this. As mentioned above, one way to further improve model utility is to remove problematic classes, as they lower the ability of the model to help those missing limbs, while any improvement over current standards (7-15 classes) represents a big improvement in the quality of life for an amputee. This can be accomplished by removing classes with balanced accuracy a certain number of standard deviations away from the average balanced accuracy, as seen in Table 4.

Simply removing 5 problematic classes greatly increases the balanced accuracy, and slightly increases accuracy. By removing 15 slightly problematic classes, the accuracy even eclipses 90%, representing significantly higher utility to the end user.

6.2 Estimation of Uncertainty

It is highly important, especially when dealing with black box models such as neural networks, to quantify the uncertainty of a prediction. The de-facto stan-

Table 2: Diagnostics of each class

Gesture	Precision	Recall	f1-score	Support
0	0.961	0.971	0.966	256527
1	0.665	0.676	0.670	3672
2	0.785	0.776	0.780	2633
3	0.692	0.744	0.717	3327
4	0.781	0.824	0.802	2477
5	0.751	0.739	0.745	2933
6	0.818	0.754	0.785	2896
7	0.663	0.707	0.684	3121
8	0.840	0.819	0.829	2977
9	0.670	0.671	0.670	2301
10	0.619	0.611	0.615	2664
11	0.704	0.738	0.721	2186
12	0.754	0.710	0.732	2562
13	0.813	0.811	0.812	3183
14	0.794	0.811	0.803	2663
15	0.861	0.877	0.869	2636
16	0.674	0.714	0.693	2654
17	0.701	0.639	0.668	2439
18	0.855	0.829	0.842	2349
19	0.787	0.751	0.768	2819
20	0.645	0.629	0.637	2549
21	0.694	0.711	0.702	2743
22	0.662	0.663	0.662	2614
23	0.625	0.556	0.589	3057
24	0.533	0.562	0.547	2644
25	0.752	0.655	0.700	2790
26	0.737	0.683	0.709	2482
27	0.752	0.767	0.759	2199
28	0.671	0.662	0.667	3102
29	0.702	0.633	0.666	2977
30	0.607	0.637	0.621	2503
31	0.554	0.583	0.568	2625
32	0.565	0.505	0.533	2657
33	0.762	0.713	0.736	2679
34	0.638	0.645	0.642	2930
35	0.753	0.751	0.752	3323
36	0.709	0.687	0.698	2486
37	0.703	0.723	0.713	2795
38	0.699	0.647	0.672	2734
39	0.686	0.714	0.700	2742
40	0.683	0.698	0.690	2992
41	0.708	0.696	0.702	2734
42	0.644	0.560	0.599	2675
43	0.634	0.613	0.623	3006
44	0.608	0.692	0.647	3261
45	0.623	0.548	0.583	2966
46	0.751	0.749	0.750	3097
47	0.685	0.721	0.702	3279
48	0.782	0.715	0.747	2980
49	0.810	0.792	0.801	3009
50	0.729	0.660	0.693	3168
51	0.850	0.779	0.813	3446
52	0.848	0.805	0.826	3449
macro avg	0.717	0.704	0.710	403712
weighted avg	0.871	0.872	0.871	403712

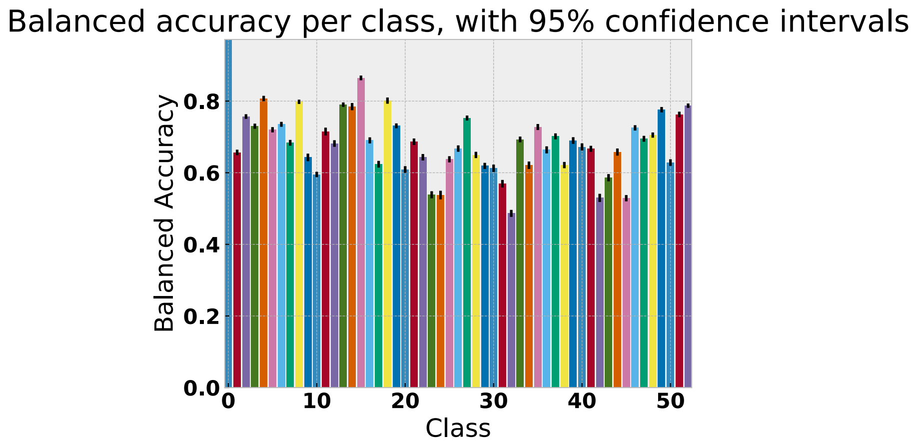


Fig. 9: Balanced accuracy by class, with 95% confidence intervals. We see that there is fairly low uncertainty with the classifications

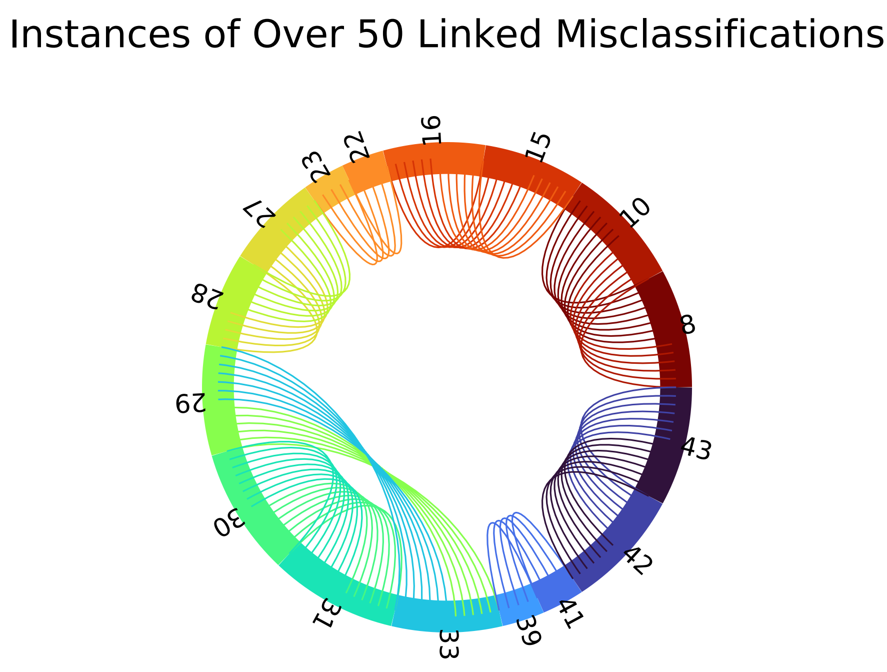


Fig. 10: Chord plot of misclassifications, showing strong links. If a line connects 0 to 12, this means that the model predicted 0 when the true class was 12, 50 times

Table 3: Scores stratified by exercise group and reference results

	Weighted Random	Unweighted Random	All Zeros	All Ones
Acc	0.4070	0.4060	0.635	0.009
Balanced Acc	0.0190	0.0190	0.019	0.019
MCC	0.0006	-0.0002	0.000	0.000
Precision	0.4070	0.4060	0.404	0.000
Recall	0.4070	0.4060	0.635	0.009
f1-Score	0.4070	0.4060	0.494	0.000
	All	Finger	Wrist	Functional
Acc	0.8720	0.7300	0.7010	0.6840
Balanced Acc	0.7040	0.7310	0.7030	0.6800
MCC	0.7827	0.7142	0.6869	0.6719
Precision	0.8710	0.8600	0.8040	0.7570
Recall	0.8720	0.7300	0.7010	0.6840
f1-Score	0.8710	0.7880	0.7480	0.7180

Table 4: Balanced accuracy after removing classes n standard deviations below the mean

Number of SDs	Number of Classes	Balanced Accuracy	Accuracy
∞	53	70.4%	87%
2	52	70.8%	87.5%
1.5	48	72%	88%
1	45	73%	89%
0.5	38	74%	90%
0	28	77%	92%

dard uncertainty estimation for deep learning models is Monte Carlo Dropout [14]. At the cost of a slight decrease in accuracy, dropout is applied at inference time in order to induce some non-determinism in the model, which allows the neural network to act as an approximation of a fully probabilistic Bayesian Network.

Figure 9 displays the results of this Bayesian approximation using dropout, as well as the general results of the classification. In general, the uncertainty is small relative to the prediction, and thus the predictions are fairly confident. The model also seems to struggle with a few motions of the wrist and a few functional motions. This is in line with the observations from Table 3.

Monte Carlo Dropout can also be used to identify classes which are closely linked. By repeating predictions, if there is a link between classes, the model will tend to err towards the links. Therefore, making many Monte Carlo predictions and then counting misclassifications can help identify common linked errors. Combined with the actual motions of the classes (as described in [5] and Figure 3), certain motions can be combined in order to improve model power with little effect on the end user. Nina 5 is often criticized for having a small subset of motions which are closely linked and very difficult to differentiate[36], and thus identifying them is appropriate.

Figure 10 shows these misclassifications in detail. Each line represents 50 misclassifications over, meaning these represent which motions the model is most uncertain about. Referring to Figure 10 and [5], classes which are similar can be identified. Gestures 42 and 43 represent a tripod grasp and prismatic pinch respectively, which are highly similar. 43 is used slightly more often in the real world, therefore they could be combined into gesture 43 every time on the prosthetic. 30 and 31 are both cylindrical grasps, and can be comfortably combined, just as 39 and 41, sphere grasps, can be. By changing these and similar classes as reported in Figure 10, the accuracy increases to 88%, MCC to 0.79, and balanced accuracy to 71%, at a minimal cost in utility.

7 Conclusions

This paper presents a novel and powerful method for classifying sEMG signal for use in smart prosthetics. First, processed data is fed into a simple, feed-forward attention mechanism which assigns a numerical score to each series. This is then fed into a simple classifier network with dropout connections and feature-wise normalization, achieving breakthrough accuracy with a simple model. The model was trained with several recent advancements in deep learning and computer vision, including focal loss and Ranger optimization, as well as a novel data augmentation scheme. The results of these techniques yielded a significant gain in both accuracy and number of gestures classifiable using low-frequency sEMG data.

The resultant boost in accuracy and breadth of available motions could be used to directly impact the lives of prosthetics users. The MYO armband used to collect the data is a low-frequency, consumer-grade sensor, readily available for general use, meaning the results of this study are not only powerful, but affordable. By having a significantly wider range of motions, and not being limited to just functional motions, amputees will have a significantly higher quality of life. Moreover, using post-hoc analysis of where the misclassifications occur, the gesture classifier can be made more useful by removing problem classes, boosting the accuracy anywhere from 1% (using linked misclassifications) and 5% (using mean-based removal). This may limit the range of motions of the device, but as discussed in subsection 6.2, many of these motions are closely related and somewhat redundant. Therefore, they can be removed with little impact to the end user. This implies that this model can be made more useful with a more robust training set, such as that proposed in [36]. Similarly, a simple feed-forward model can easily be embedded in lightweight hardware for long term, low latency, low power usage. Simple DNNs such as this can be assisted by cheap accelerators, such as those mentioned in [28], which make them more efficient and consume less power, meaning they can act as an amputee's hand for a longer period of time.

One general flaw with the methods discussed in this paper is the use of labels at all. The human brain is not a supervised algorithm, and can learn and adapt quickly to new situations, while a classifier simply cannot. This research should be expanded into the fields of reinforcement and imitation learning. In [15], imitation learning is used in a prosthetic leg, and it is proven that it can adapt to walking on new, never before seen terrains, and capable of achieving superhuman optimization in walking. This is the next big step for smart prosthetics, and should be heavily researched in the future. If combined with computer vision and other machine learning sub-fields, sEMG analysis could be the basis for robust models with an infinite range of gestures, allowing amputees to regain complete functionality of their limbs.

References

1. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D.G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., Wicke, M., Yu, Y., Zheng, X.: Tensorflow: A system for large-scale machine learning. In: 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16). pp. 265–283 (2016), `https://www.usenix.org/system/files/conference/osdi16/osdi16-abadi.pdf`
2. 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)
3. Atzori, M., Cognolato, M., Müller, H.: Deep learning with convolutional neural networks applied to electromyography data: A resource for the classification of movements for prosthetic hands. Frontiers in Neurorobotics **10** (Jul 2016). https://doi.org/10.3389/fnbot.2016.00009
4. 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)
5. 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)
6. Atzori, M., Gijsberts, A., Kuzborskij, I., Elsig, S., Hager, A.G.M., Deriaz, O., Castellini, C., Muller, H., Caputo, B.: Characterization of a benchmark database for myoelectric movement classification. IEEE Transactions on Neural Systems and Rehabilitation Engineering **23**(1), 73–83 (2015). https://doi.org/10.1109/tnsre.2014.2328495
7. Ba, J.L., Kiros, J.R., Hinton, G.E.: Layer normalization (2016)
8. Baldi, P., Brunak, S., Chauvin, Y., Andersen, C.A.F., Nielsen, H.: Assessing the accuracy of prediction algorithms for classification: an overview . Bioinformatics **16**(5), 412–424 (05 2000). https://doi.org/10.1093/bioinformatics/16.5.412, `https://doi.org/10.1093/bioinformatics/16.5.412`
9. Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J., Layton, R., VanderPlas, J.,

Joly, A., Holt, B., Varoquaux, G.: API design for machine learning software: experiences from the scikit-learn project. In: ECML PKDD Workshop: Languages for Data Mining and Machine Learning. pp. 108–122 (2013)
10. Campbell, P.: Apl's modular prosthetic limb reaches new levels of operability. JHU/APL Brand (Jan 2016), https://www.jhuapl.edu/PressRelease/160112
11. Chollet, F.: keras. GitHub repository (2015), https://github.com/fchollet/keras
12. 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
13. Fawaz, H.I., Forestier, G., Weber, J., Idoumghar, L., Muller, P.A.: Deep learning for time series classification: a review. Data Mining and Knowledge Discovery **33**(4), 917–963 (Feb 2019). https://doi.org/10.1007/s10618-019-00619-1
14. Gal, Y., Ghahramani, Z.: Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In: international conference on machine learning. pp. 1050–1059 (2016)
15. Gao, C., Gehlhar, R., Ames, A.D., Liu, S.C., Delbruck, T.: Recurrent neural network control of a hybrid dynamic transfemoral prosthesis with edgedrnn accelerator. arXiv preprint arXiv:2002.03197 (2020)
16. Geethanjali, P.: Myoelectric control of prosthetic hands: state-of-the-art review. Medical Devices: Evidence and Research **Volume 9**, 247–255 (2016). https://doi.org/10.2147/mder.s91102
17. Hinton, G.E., Osindero, S., Teh, Y.W.: A fast learning algorithm for deep belief nets. Neural Computation **18**(7), 1527–1554 (2006). https://doi.org/10.1162/neco.2006.18.7.1527
18. 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
19. Jastrzebski, S., Szymczak, M., Fort, S., Arpit, D., Tabor, J., Cho, K., Geras, K.: The break-even point on optimization trajectories of deep neural networks. arXiv preprint arXiv:2002.09572 (2020)
20. Ketyko, I., Kovacs, F., Varga, K.Z.: Domain adaptation for semg-based gesture recognition with recurrent neural networks. 2019 International Joint Conference on Neural Networks (IJCNN) (2019). https://doi.org/10.1109/ijcnn.2019.8852018
21. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision. pp. 2980–2988 (2017)
22. Liu, L., Jiang, H., He, P., Chen, W., Liu, X., Gao, J., Han, J.: On the variance of the adaptive learning rate and beyond. arXiv preprint arXiv:1908.03265 (2019)
23. Misra, D.: Mish: A self regularized non-monotonic neural activation function. arXiv preprint arXiv:1908.08681 (2019)
24. Myers, L., Lowery, M., Omalley, M., Vaughan, C., Heneghan, C., Gibson, A.S.C., Harley, Y., Sreenivasan, R.: Rectification and non-linear pre-processing of emg signals for cortico-muscular analysis. Journal of Neuroscience Methods **124**(2), 157–165 (2003). https://doi.org/10.1016/s0165-0270(03)00004-9
25. ottobock: Myoelectric prosthetics. ottobock. https://www.ottobockus.com/prosthetics/upper-limb-prosthetics/solution-overview/myoelectric-prosthetics/

26. Phinyomark, A., Scheme, E.: Emg pattern recognition in the era of big data and deep learning. Big Data and Cognitive Computing **2**(3) (2018). https://doi.org/10.3390/bdcc2030021, https://www.mdpi.com/2504-2289/2/3/21
27. 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
28. Putic, M., Buyuktosunoglu, A., Venkataramani, S., Bose, P., Eldridge, S., Stan, M.: Dyhard-dnn: Even more dnn acceleration with dynamic hardware reconfiguration. In: 2018 55th ACM/ESDA/IEEE Design Automation Conference (DAC). pp. 1–6 (2018)
29. Raffel, C., Ellis, D.P.: Feed-forward networks with attention can solve some long-term memory problems. arXiv preprint arXiv:1512.08756 (2015)
30. Rehman, M.Z.U., Waris, A., Gilani, S., Jochumsen, M., Niazi, I., Jamil, M., Farina, D., Kamavuako, E.: Multiday emg-based classification of hand motions with deep learning techniques. Sensors **18**(8), 2497 (Jan 2018). https://doi.org/10.3390/s18082497
31. RING, F.: Learning Approaches in Signal Processing. PAN STANFORD Publishing (2018)
32. Shen, S., Gu, K., Chen, X., Yang, M., Wang, R.: Movements classification of multi-channel semg based on cnn and stacking ensemble learning. IEEE Access **7**, 137489–137500 (2019)
33. 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
34. Tamura, H., Yan, M., Sakurai, K., Tanno, K.: Eog-semg human interface for communication. Computational Intelligence and Neuroscience **2016**, 1–10 (2016). https://doi.org/10.1155/2016/7354082
35. Tong, Q., Liang, G., Bi, J.: Calibrating the adaptive learning rate to improve convergence of adam. arXiv preprint arXiv:1908.00700 (2019)
36. Tsagkas, N., Tsinganos, P., Skodras, A.: On the use of deeper cnns in hand gesture recognition based on semg signals. 2019 10th International Conference on Information, Intelligence, Systems and Applications (IISA) (2019). https://doi.org/10.1109/iisa.2019.8900709
37. 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
38. Wei, W., Wong, Y., Du, Y., Hu, Y., Kankanhalli, M., Geng, W.: A multi-stream convolutional neural network for semg-based gesture recognition in muscle-computer interface. Pattern Recognition Letters **119**, 131–138 (2019). https://doi.org/10.1016/j.patrec.2017.12.005
39. Wu, Y., Zheng, B., Zhao, Y.: Dynamic gesture recognition based on lstm-cnn. 2018 Chinese Automation Congress (CAC) (2018). https://doi.org/10.1109/cac.2018.8623035

40. Zhang, M., Lucas, J., Ba, J., Hinton, G.E.: Lookahead optimizer: k steps forward, 1 step back. In: Advances in Neural Information Processing Systems. pp. 9593–9604 (2019)