

# OPTIMIZATION OF NEURAL NETWORK ARCHITECTURE FOR BIOMECHANIC CLASSIFICATION TASKS WITH ELECTROMYOGRAM INPUTS

Alayna Kennedy<sup>1</sup> and Rory Lewis<sup>2</sup>

<sup>1</sup>The Pennsylvania State University, University Park, PA, USA

<sup>2</sup>Department of Computer Science, University of Colorado at Colorado Springs, USA

## ABSTRACT

*Electromyogram signals (EMGs) contain valuable information that can be used in man-machine interfacing between human users and myoelectric prosthetic devices. However, EMG signals are complicated and prove difficult to analyze due to physiological noise and other issues. Computational intelligence and machine learning techniques, such as artificial neural networks (ANNs), serve as powerful tools for analyzing EMG signals and creating optimal myoelectric control schemes for prostheses. This research examines the performance of four different neural network architectures (feedforward, recurrent, counter propagation, and self organizing map) that were tasked with classifying walking speed when given EMG inputs from 14 different leg muscles. Experiments conducted on the data set suggest that self organizing map neural networks are capable of classifying walking speed with greater than 99% accuracy.*

## KEYWORDS

*Electromyogram, Prostheses, Neural Networks, Biomechanical Analysis, Machine Learning, Myoelectric Control Schemes, Self Organizing Maps, Pattern Recognition Algorithms*

## 1. INTRODUCTION

Assistive prosthetic devices commonly use a myoelectric control scheme, which adjusts the function of a prosthetic device given electromyogram (EMG) inputs, and which has been proven an effective method of man-machine interfacing between user and prosthetic [1]-[4]. However, despite the significant development of the prosthetic industry over the past decade, high-accuracy commercial prostheses remain too expensive for the average middle-class amputee to afford [5], [6]. Even the most accurate devices possess multiple issues, including difficulty actuating multiple degrees of freedom and low accuracy without a high number of electrodes [7]. Within academia, computational intelligence pattern recognition techniques of EMG analysis provide accurate results, but are often computationally expensive [8]. Therefore, academic models result in limited real-world improvements in the control of prosthetic devices, since available cheap

prostheses do not possess the computation power necessary to run complex pattern recognition algorithms. An ideal prosthetic control scheme would be able to achieve accuracy in classification tasks while remaining simple enough to be implemented in inexpensive devices [6], [8].

Among the common electrophysiological signals, EMG recordings are used most extensively in man-machine interfacing because of their non-invasiveness, relatively easy application, and richness of neural information [3], [8]. While other biosignals like electroencephalograms (EEGs) and electrooculograms (EOGs) can be used to predict human movement, EMG signals have become the standard biosignal for myoelectric prostheses because they directly transmit electrical signals from the muscle during periods of contraction or relaxation [3], [9], [10]. In addition, EMG signals have been shown to precede muscle kinematics by 100ms; therefore, they can be used to predict human movements to create a prosthetic device that would analyze EMG signals and transmit them to a prosthetic, which would then move as if it were the amputee's biological limb [1], [2].

The EMG signal is the sum of the electrical activity of the muscle fibers, as triggered by the impulses of activation of the innervating motor neurons [3], [8], [10]. Surface EMGs are obtained by convolution of each motor neuron spike train by the motor unit action potential and have mathematically the same expression as the neural efferent signal [10].

Equation 1 shows a simple model of the EMG signal:

$$x(n) = \sum_{r=0}^N h(r)e(n-r) + w(n)$$

Equation 1

where  $x(n)$  is the modeled EMG signal,  $e(n)$  represents the firing impulse,  $h(r)$  represents the motor unit action potential,  $w(n)$  is the zero mean additive white Gaussian noise, and  $N$  is the number of motor unit firings [8].

Despite the rather direct relation between a motion and the expressed EMG signal, however, there remain several open issues before the control of prostheses by EMG will reach the ideal characteristics needed for widespread acceptance by patients [4], [8]. Raw EMG data alone provides very valuable information about human biomechanics in a fairly useless form, since the amount of noise in the data require it to be processed, quantified, cleared of noise, and decomposed into individuals MUAPs (Fig 1.) before it can be used in any significant way [7], [10], [11]. While difficult, the effective analysis of EMG signals can be achieved through wavelet analysis, artificial neural networks (ANNs), and different clustering algorithms [10].

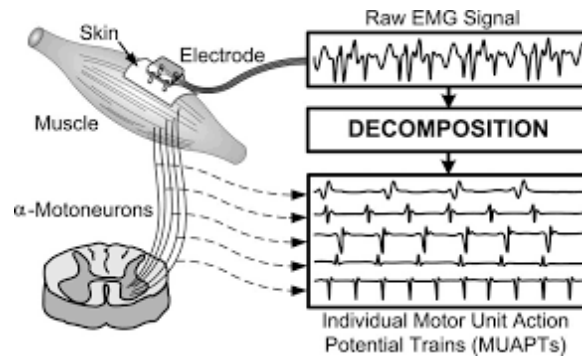


Fig. 1: Decomposition of EMG signals into individual MUAPs

While many studies have been done on the continuous analysis of EMG data using neural networks and complex pattern recognition algorithms, little work has been done on less computationally expensive classification tasks [4]. While regression tasks provide detailed information in the form of bodily spatial coordinates, they also require more electrodes and achieve a lower degree of accuracy than many classification tasks [7]. Although classification tasks only output a discrete category, they can also provide a great deal of information about future movement. For example, walking speed can determine multiple factors about human gait [12]. Other applications of classification tasks to prosthetic technology can be found in Gandolla and Gehani's studies, where trained ANNs classified distinct hand movements, resulting in a hand prosthetic performing the movement [1], [3].

This study aims to find a network architecture that can accurately classify kinematic information from EMG signals while remaining simple enough for application in affordable prostheses, since one of the most significant applications of EMG analysis technology is within the field of orthotics and prosthetics [5].

This study aims to find the best way to use ANNs to classify human movement from EMG data, by measuring the performance of multiple neural network architectures when tasked with classification tasks. The different ANN architectures will attempt to classify walking speed into one of five categories given surface EMG inputs from 14 different leg muscles [13].

## 2. PREVIOUS WORK

The parallel computation power and nonlinear operations performed by the human brain inspired the original ANNs, so when creating biomimetic control schemes for prosthetics, many researchers turned to these networks to create efficient myoelectric devices that worked with the human body [2]. While some of the earliest attempts to create myoelectric prostheses date back to the 1970s, advanced pattern recognition techniques did not emerge until the 1990s with the rise of more accurate machine learning techniques like neural network backpropagation and time-series analysis of inputs [3], [14]. One of these developments in myoelectric control was a dynamic

recurrent neural network which accurately predicted spatial coordinates of arm trajectory [15]. The development of recurrent networks, in which all neurons are interconnected, resulted in a marked improvement in the accuracy of spatial coordinate prediction of the body given EMG inputs. Networks like Draye's and Cheron's also implemented time-series backpropagation, which took into account the previous values of EMG input instead of just the immediate input channel [14], [15].

Other networks modeled EMG inputs with an auto-regressive (AR) model, and then passed them through an ANN to control the movements of a virtual prosthetic [3]. The results from these studies have shown that controlling prostheses through an ANN recognition of EMG patterns can optimize the number of electrodes, provide greater degrees of freedom for the device, and accurately predict user intent before muscle movements [1], [7], [20]. Two important network architectures for classification of EMG data are the Kohonen Network, and a cascade architecture network with a preprocessing step involving Kohonen maps.

## 2.1. Kohonen Map

Also referred to as a self-organizing map (SOM), this network implements unsupervised learning, which maps points in the input space to points in the output space while preserving the topology [16]. Normally, the input space is of high dimension while the output is usually two dimensional. The network identifies the spatial concentration of the network activity that is best tuned to the present input [3], [16].

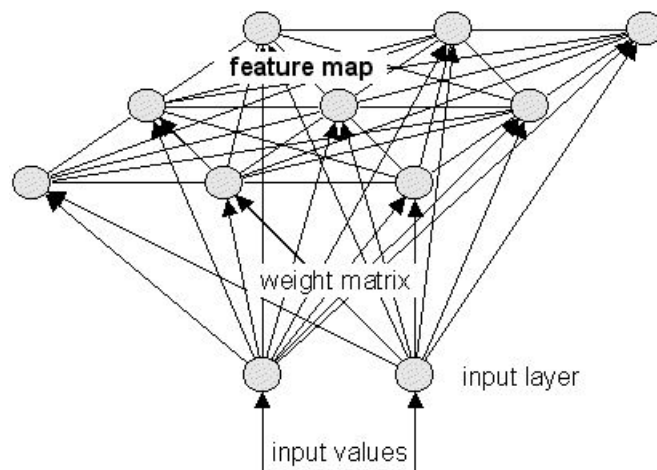


Fig. 2: Detailed view of a Kohonen self organizing map network

The following six steps explain the algorithm for producing Kohonen maps:

**Step one: Select network topology:** The arrangement of the clusters for the network can be square, circular, etc.

**Step Two: Initialize weights to random values:** The weight matrix represents the connections between the neurons of the network, and it is randomized to initial values which are subsequently modified during the learning phase

**Step Three: Select a pattern:** Chooses an input pattern, “x”, from the input examples.

**Step Four: Find best matching unit:** The node with the weight vector most similar to the input vector, defined as the node with the smallest Euclidean distance to the input weights, is selected as the matching node.

**Step Five: Update weights to all nodes:** The “winning” node and its topological neighborhood are updated by the SOM algorithm according to the equation:

$$m_i(t+1) = \begin{cases} m_i(t) + \alpha(t)[x(t) - m_i(t)] & \text{if } i \in Nc(t) \\ m_i(t) & \text{if } i \notin Nc(t) \end{cases}$$

Equation 2

Where  $m_i(t+1)$  is the new weight, and  $m_i(t)$  is the old weight,  $\alpha(t)$  is the learning rate factor ( $0 < \alpha(t) < 1$ ) and  $t = 0, 1, 2, \dots$  is an integer representing the discrete time coordinate.

**Step Six: Iteration:** Repeat steps 1 to 5 for all input patterns and the repeat for a pre-determined number of iterations [3], [16].

SOMs have been used widely in many applications, including in EMG clustering classification. This network can be used as a preprocessing stage for other ANN architectures such as a Cascade Architecture with Feature Maps (CANFM).

## 2.2. Cascade Architecture with Feature Maps (CANFMs):

The CANFM network implements a cascaded architecture of neural networks with feature maps (CANFM) [17]. This network first passes the data through an unsupervised Kohonen self-organizing map, outputting 2D coordinates onto the x and y axes of the 2D topological net, which both reduces the input dimensions of the EMG data channels and removes some noisy data from the original inputs [16], [17].

This network first randomizes the initial weights of the network, then passes the values through a self-organizing map (SOM) where the EMG values are clustered. The unsupervised SOM can find a winning neuron on the 2-D topology map to represent the original pattern, and the x and y coordinate of this winning neuron  $y_{\text{texts subscript}\{c\}}$  become the input values for a back-propagation neural network (BPNN) [17], [18]. After reduction of the input space using Kohonen's SOM, the three sets of 2-D coordinates (six newly condensed features) are fed into the

BPNN for further classification. Huang chose the BPNN as the post-classifier of CANFM because of its learning ability and fast recall speed. In the case of Huang's study, there are eight postures to be classified, so the BPNN has eight output nodes (Fig. 3) [17].

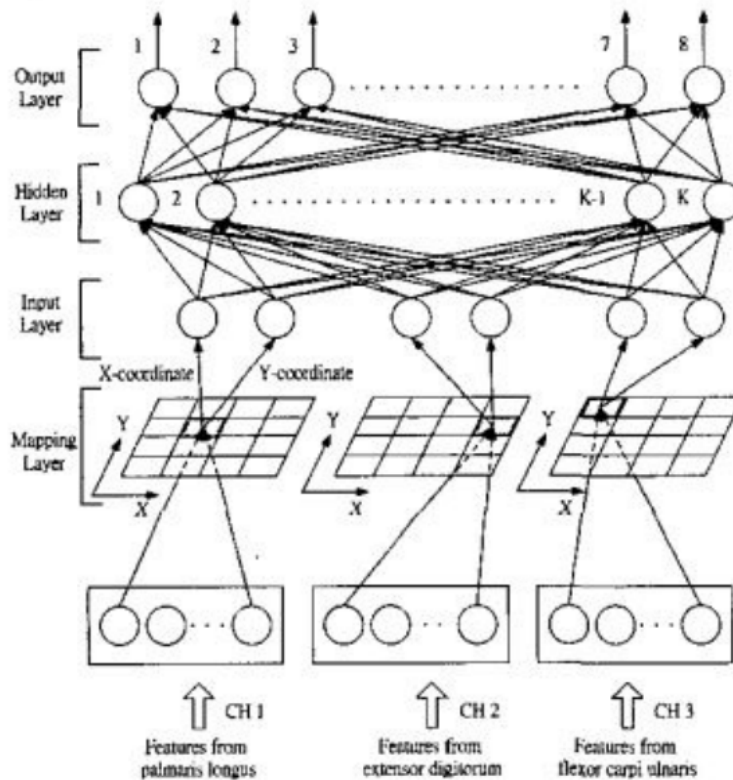


Fig. 3: Detailed information of CANFM

### 3. ACQUISITION OF EMG DATA

The EMG data for this study was collected by Hof et. al and made available to the public via the Clinical Gait Analysis Database [13].

#### 3.1 Collecting Data

Surface EMGs of 14 leg muscles were recorded from two homogeneous groups (n=9 and 11, respectively) of young healthy male subjects (mean age 22 years (S.D. 1.5), stature 1.85 m (S.D. 0.05), leg length 0.98 m (S.D. 0.04), body mass 73 kg S.D. 8). The average personal data of both groups was matched to compensate for the division in the two groups. Subjects walked barefoot on a 10 m indoor walkway at speeds of 0.75, 1.00, 1.25, 1.50, and 1.75 m s<sup>-1</sup>.

### 3.2 Pre-processing: Filtering and Blocking

Compared with other biosignals, EMG signals are difficult to analyze due to their small amplitude, which makes them highly subject to both internal and external electrical noise. Internal noise occurs when an electrode picks up signals from more than one MUAP and overlaps their signals, while external noise can result from equipment noise, electromagnetic radiation, or motion artefacts. Therefore, pre-processing and filtering is essential to obtain reliable raw EMG data [3], [10], [13].

In this study, the EMGs were high-pass filtered at 20 Hz, rectified and smoothed with a 25 Hz third order Butterworth low-pass filter. Smoothed rectified EMGs were, after A/D conversion with a sample frequency of 100 Hz, linearly interpolated to 100 points per stride, triggered by heel contact of the leg of interest. The recorded steps were screened to exclude those with obvious artefacts or incorrect foot contacts. In this way for every individual  $i$ , normalized speed  $\hat{v}$ , and muscle  $m$ , average individual profiles  $e(p, m, v, i)$  were determined from at least 10 steps over  $p=1-100\%$  of the gait cycle [13].

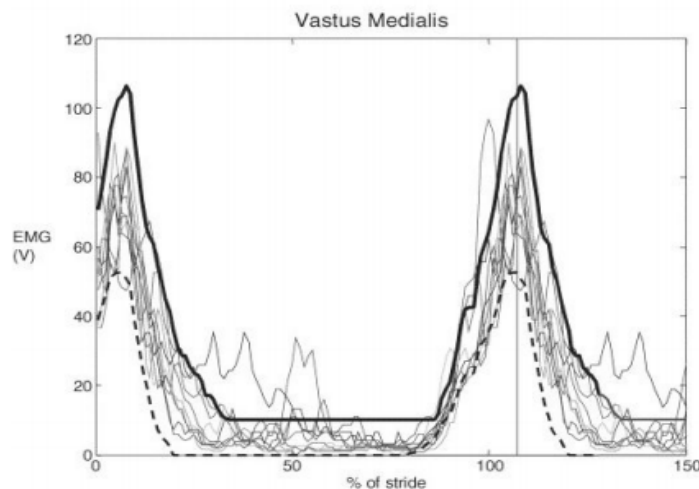


Fig. 4: Average EMG profiles of vastus medialis muscle after normalizing. Also shows low and high limits. Time is given as a percentage of stride, starting at heel contact, but in a scale running 0-100-50%, in order to represent better the activity around heel strike.

## 4. NEURAL NETWORK CREATION

The EMG data taken from the Hof study was split into training, validation, and testing data in a 2-1-1 ratio, as is considered best practice when training neural networks [19]. The data was then normalized to a range of values between 0 and 1 using the following equation [12].

$$f(x) = \frac{(x - d_L)(n_H - n_L)}{(d_H - d_L)} + n_L$$

Equation 3

where  $x$  is the value to be normalized,  $d$  represents the high and low values of the data, and  $n$  represents the high and low normalization range desired [20].

Four different neural networks architectures were created, a multilayer feedforward perceptron network, a recurrent neural network, a self organizing map (SOM), and a counter propagation neural network (CPN). Each network performed classification tasks on the EMG data, and their performance was recorded. In the first type of architecture, feedforward perceptrons, the networks all had a single hidden layer and were tested with four different combinations of activation functions: linear, radial basis function (RBF), sigmoid (SIG), and hyperbolic tangent function (TANH) [21]. The TANH activation function uses the hyperbolic tangent function

$$f(x) = \frac{e^{2x} - 1}{e^{2x} + 1}$$

Equation 4

and has proved a powerful tool for classification tasks [20]. However, in the analysis of EMG data with neural networks, the SIG function

$$f(x) = \frac{1}{1 + e^{-x}}$$

Equation 5

is more typically used [15], [19]. The feedforward networks were also tested with varying numbers of hidden layer neurons to ascertain the optimal number for both speed and accuracy of classification. All of the feedforward networks implemented a resilient backpropagation (RPROP) learning algorithm, which is a gradient-based optimization technique similar to the more common regular backpropagation. RPROP is often faster than training with backpropagation and does not require any free parameter values to be specified. RPROP works similarly to traditional backpropagation, except an individual delta value is calculated for each connection. These delta values are gradually changed until the neural network weight matrix converges on a potentially ideal weight matrix [19], [20], [22].

In each iteration of RPROP, the new weights are given by [22]:



$$w_{ij}^{(t+1)} = w_{ij}^{(t)} + \Delta w_{ij}^{(t)}$$

Equation 6

The second type of architecture tested was a recurrent neural network (RNN) architecture. RNNs are a subset of ANNs where connections between units form a directed cycle. This allows the internal state of the network to exhibit dynamic temporal behavior, using their internal memory to process sequences of inputs. The two types of RNNs tested were a three-layer Elman network and a Jordan network. In the Elman simple recurrent network, context units connected to the hidden layer maintain a copy of the previous values of the hidden units, allowing the network to maintain a memory of the previous time step [23]. Jordan networks are similar, but the context units are fed from the output layer instead of the hidden layer [24].

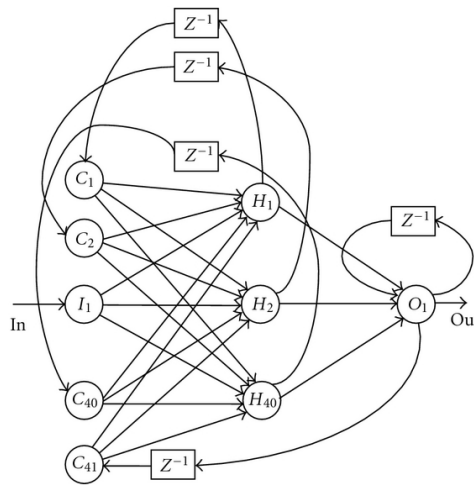


Fig. 5: A structure of the trained Elman-Jordan neural network

The third type of architecture tested were SOM neural networks. This network, unlike the previous two, was trained using unsupervised learning to produce a two-dimensional discretized representation of the input space of the training samples, called a topology map. In addition, the SOM applied competitive learning techniques to improve the network. The SOM architecture was tested with two different training algorithms: K-means nearest neighbor training and cluster copy training.

The fourth network type, counter propagation networks (CPNs) are similar to CANFM networks, consisting of an outstar network and a competitive filter network. Each neuron in the input layer is processed through a Kohonen network which categorizes the input pattern, serving as the hidden layer for the network [25]. The outputs of the Kohonen map are then filtered through an outstar array which reproduces the correct output pattern for the category. Training is done in two

stages; first the hidden layer learns to categorize the patterns and then the weights for that layer become fixed. Then the output layer is trained. One of the advantages of CPN and Kohonen networks is that the training phase requires a relatively small number of epochs, usually several hundred, which is considered a tiny number compared to other ANNs, such as convolutional and deep neural networks which require thousands of iterations of training [3].

For each of the four network architectures chosen, all of the networks were trained, tested, and evaluated for error four times, then the error results over each of the four runs were averaged. Each network was trained using the training and validation, then the method was evaluated using the testing data. When training the network, a maximum error percentage of 1%, max step of 50 and initial update of 0.1 were used [26].

## **5. RESULTS FROM THE COMPARISON OF NETWORK ARCHITECTURES**

Initial results on optimal classification for multiple neural network architectures corroborated results from previous studies, showing that in feedforward networks, the number of neurons in the hidden layer does not significantly affect accuracy of classification, but a higher number of hidden neurons allows the network to train faster. The feedforward networks tested all had 14 inputs, corresponding to the 14 EMG channels, and used a TANH function for both the hidden and output layer activation.

First, the network was tested with 7 hidden layer neurons, half the number of input neurons, which resulted in a 4.782% error in 102 seconds of training time. The number of hidden layers was increased by two during each subsequent experiment, and both the error and training time recorded, with the final test consisting of a network with 30 hidden layer neurons. While error decreased slightly with a greater number of hidden neurons, the more significant change was the decrease in training time with more hidden layer nodes. Both the error and the training time leveled off at around 28 hidden layer neurons, with an error value of 4.4% and a training time of 0.35 seconds. Therefore, we can state generally that the optimal number of hidden neurons to minimize both training time and error for this experiment is about twice the number of input neurons, a result which has been produced in several previous studies [25].

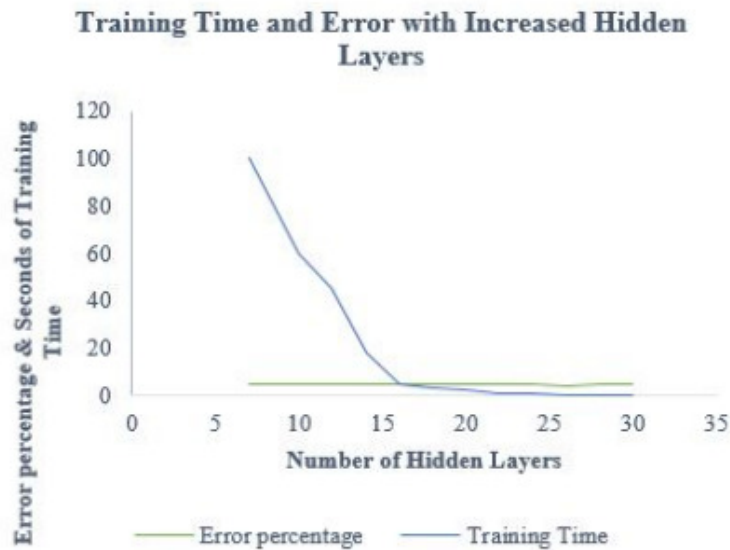


Fig. 6: Graph of training time and percent error of feedforward network vs the number of hidden neurons in the hidden layer of the perceptron.

In analyzing the performance of the feedforward neural network architecture, we can see that using the TANH activation function for both the hidden layer and the output layer activation provides the most accurate results for classification tasks, with an average error of 0.99%. Using the SIG activation function in the hidden layer and output layer activation resulted in an error of 12%, a result which is surprising since most neural networks with EMG inputs use the sigmoid activation function with relative success. However, most studies using the SIG function utilize networks which perform regression tasks to output spatial coordinates, while our network classified the EMG data into 5 discrete walking speeds.

Furthermore, both the feedforward network with TANH activation for the hidden layer and linear outputs and the feedforward network with an RBF activation function performed poorly when classifying EMG data, with average errors of 15%. While the TANH & linear network was included primarily to serve as a control for the TANH & TANH network, the RBF function has been shown to perform well in networks that classify numerical data. However, these results suggest that RBF networks do not perform well on classification tasks with small-amplitude EMG data.

TABLE 1  
ERROR PERCENTAGES OF NEURAL NETWORK ARCHITECTURES

Machine Learning Method	Average Accuracy
Feedforward 1: Activation TANH & Linear	15%
Feedforward 2: Activation SIG & SIG	11%
Feedforward 3: Activation TANH & TANH	0.99%
Feedforward 4: RBF Activation	15%
Recurrent 1: Elman Network	13.7%
Recurrent 2: Jordan Network	15.3%
Counter Propagation Network	15.4%
Self Organizing Map 1: K-Means nearest neighbor	2.1%
Self Organizing Map 2: Cluster copy training	0.27%

Fig. 7: Average error percentages of four feedforward networks with varying activation functions and two SOM with different training techniques.

The performance of recurrent neural networks on classification tasks was comparable to the TANH and linear feedforward neural network, with the Jordan recurrent network producing an error of 15.3% and the Elman recurrent network producing an error of 13.7%. While dynamic recurrent neural networks have previously performed accurate classifications of EMG data when diagnosing neuromuscular diseases or prehensile human postures, they do not appear to perform well when given normalized EMG inputs to classify walking speed [17], [24]. This result could be due to the generalized normalization of the data to values between 0 and 1, which has been shown to improve results in feedforward networks, but is often not used with recurrent systems [23].

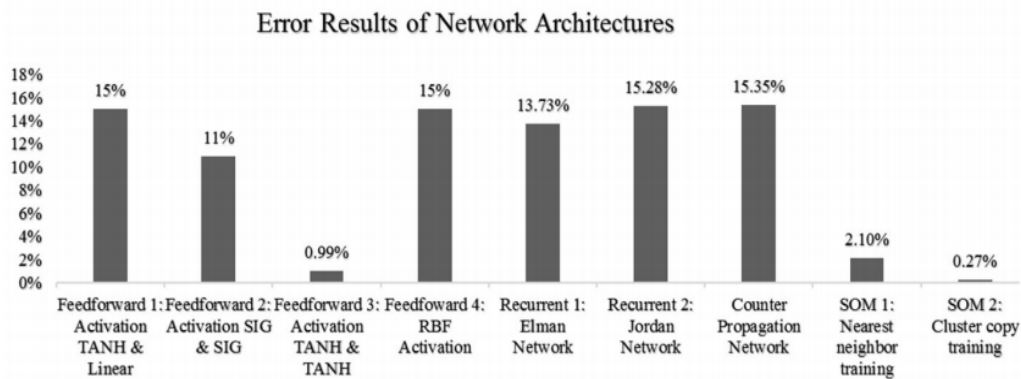


Fig. 8: Average error percentages of four feedforward networks with varying activation functions and two SOM with different training techniques.

Although the TANH feedforward network classified walking speed with an error of less than 1%, the most effective networks for classifying EMG data were SOM neural networks. Both SOM networks had an average error rate of under 2.5%, and the SOM network trained with a cluster copy training algorithm had an error rate of 0.27%. These results give the cluster copy trained SOM in this study an average accuracy percentage of 99.73%, a result which improves upon many machine learning techniques of EMG classification.

**TABLE 2**  
**COMPARISON OF CLASSIFICATION RESULTS**

Machine Learning Method	Average Accuracy
Adaptive neuro-fuzzy inference system	88.9%
K-NN Clustering	91.25%
Fuzzy K-NN Clustering	92.5%
Discrete Wavelet Transform	96.67%
Dynamic CPN	97.22%
Backpropagation neural net	97.5%
CANFM	98.75
CCT SOM	99.73%

Fig. 9: Comparison of the accuracy percentages obtained from multiple machine learning techniques, including the cluster copy training SOM created in this study.

Table 2 compares the results obtained in this study from the cluster copy training (CCT) SOM with results from different machine learning techniques of EMG classification [6][17][27]. Most of the studies cited in this table classified gestures or movements, not walking speed. For example, the adaptive neuro-fuzzy inference system (ANFIS), classified input EMG data into one of four different hand movements [6]. Although the classification tasks differ slightly from those in this study, the techniques are similar enough for a comparison to be made between the results. Each study utilized various machine learning methods implemented in software packages that can be found in the respective papers.

The results displayed in Table 2 show that the CCT SOM outperforms the other machine learning methods by as much as 10.83% ( $10.83\% = 99.73\% - 88.9\%$ ). We can conclude that in creating an algorithm for the classification of walking speed given EMG inputs, a CCT SOM provides the most accurate results, with an equal amount of computational expense as most other machine learning pattern classification algorithms.

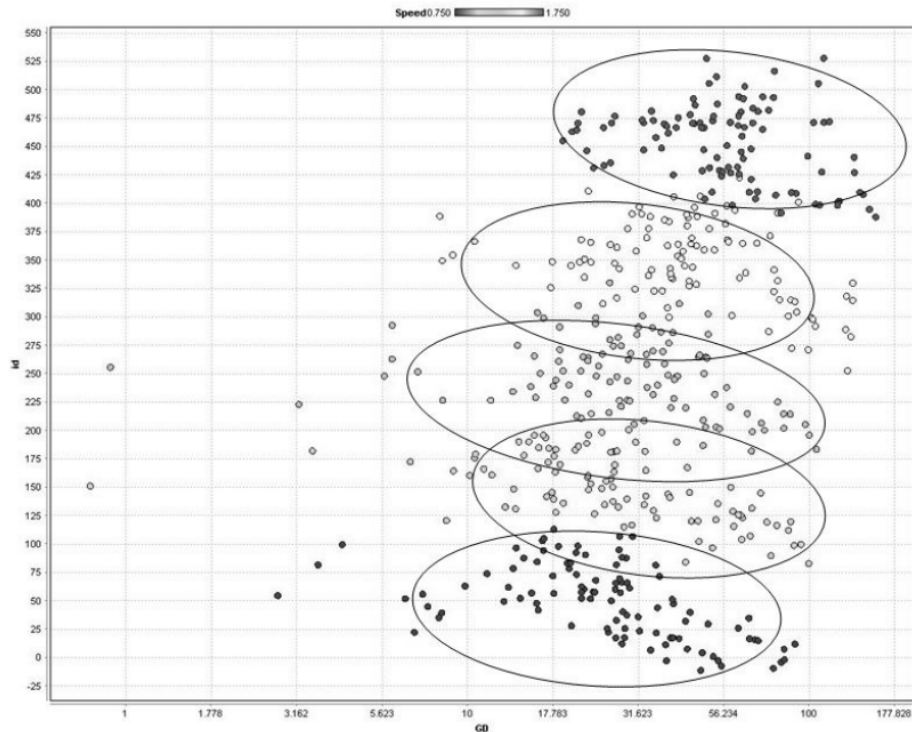


Fig. 10: Visualization of the two-dimensional topology map outputted by the SOM. This visualization does not reflect the exact CCT SOM created in this study, but simply serves for visualization purposes.

## 5. CONCLUSION

The final goal of this research project is to further research on a more optimal myoelectric prosthetic control scheme that would be able to achieve accuracy in classification tasks while remaining simple enough to be implemented in inexpensive devices. Results from this preliminary study suggest that self organizing maps can be used to accurately classify human movement given EMG input data. Given their promising accuracy in walking speed classification, low computational expense compared to networks that output spatial coordinates, and fast training time, SOMs could potentially form the basis for a more optimal myoelectric control scheme for prosthetics. However, this study utilized a limited data set and did not assess every method of EMG analysis for prosthetics.

Therefore; further work should include a more comprehensive study to cover a wider range of movements with high performance data acquisition and better statistical significance. In addition, the possibility of testing EMG data with different neuroclustering techniques should be considered, with the ultimate goal of creating a myoelectric control scheme that can successfully integrate machine learning techniques to create a computationally inexpensive and accurate prosthetic device.

## REFERENCES

- [1] Gandolla, S. Ferrante, D. Baldassini, M. Cotti Cottini, C. Seneci and A. Pedrocchi, Mediterranean Conference on Medical and Biological Engineering and Computing, 14th ed. Milano, Italy: NearLab, Department of Electronics, Information, and Bioengineering, 2016, pp. 634-637.
- [2] H. Herr, G. Whiteley and D. Childress, *Cyborg Technology– Biomimetic Orthotic and Prosthetic Technology*. Bellingham, Washington: SPIE Press, 2003, pp. 2-35.
- [3] A. El Gehani, S. Mohamed, H. Busedra and A. Gawedar, “EMG Pattern Recognition System Based on Self-Organizing Map for Myo-Electric Prosthesis”, ICECE, 2013.
- [4] C. Pattichis, C. Schizas and L. Middleton, “Neural network models in EMG diagnosis”, IEEE Transactions on Biomedical Engineering, vol. 42, no. 5, pp. 486-496, 1995.
- [5] G. McGimpsey and T. Bradford, *Limb prosthetics services and devices*, 1st ed. Worcester, Massachusetts: Bioengineering Institute Center for Neuroprosthetics, 2008.
- [6] H. Jahani Fariman, S. Ahmad, M. Hamiruce Marhaban, M. Ali Jan Ghasab and P. Chappell, “Simple and Computationally Efficient Movement Classification Approach for EMG-controlled Prosthetic Hand: ANFIS vs. Artificial Neural Network”, Intelligent Automation & Soft Computing, vol. 21, no. 4, pp. 559-573, 2015.
- [7] A. Balbinot, A. Junior and G. Favieiro, “Decoding Arm Movements by Myoelectric Signal and Artificial Neural Networks”, ICA, vol. 04, no. 01, pp. 87-93, 2013.
- [8] D. Farina, Ning Jiang, H. Rehbaum, A. Holobar, B. Graimann, H. Dietl and O. Aszmann, “The Extraction of Neural Information from the Surface EMG for the Control of Upper-Limb Prostheses: Emerging Avenues and Challenges”, IEEE Trans. Neural Syst. Rehabil. Eng., vol. 22, no. 4, pp. 797-809, 2014.
- [9] M. Seyedali, J. Czerniecki, D. Morgenroth and M. Hahn, “Co-contraction patterns of trans-tibial amputee ankle and knee musculature during gait”, Journal of NeuroEngineering and Rehabilitation, vol. 9, no. 1, p. 29, 2012.
- [10] M. Reaz, M. Hussain and F. Mohd-Yasin, “Techniques of EMG signal analysis: detection, processing, classification and applications”, Biol. Proced. Online, vol. 8, no. 1, pp. 11-35, 2006.
- [11] A. Au and R. Kirsch, “EMG-based prediction of shoulder and elbow kinematics in able-bodied and spinal cord injured individuals”, IEEE Transactions on Rehabilitation Engineering, vol. 8, no. 4, pp. 471-480, 2000.
- [12] C. Vaughan, B. Davis and J. O’Connor, *Dynamics of human gait*. Champaign, Ill.: Human Kinetics Publishers, 1992.
- [13] A. Hof, H. Elzinga, W. Grimmius and J. Halbertsma, “Detection of non-standard EMG profiles in walking”, Gait and Posture, vol. 21, no. 2, pp. 171-177, 2005.
- [14] J. Draye, D. Pavisic, G. Cheron and G. Libert, “Dynamic recurrent neural networks: a dynamical analysis”, IEEE Trans. Syst., Man, Cybern. B, vol. 26, no. 5, pp. 692-706, 1996.
- [15] G. Cheron, J. Draye, M. Bourgeois and G. Libert, “A dynamic neural network identification of electromyography and arm trajectory relationship during complex movements”, IEEE Transactions on Biomedical Engineering, vol. 43, no. 5, pp. 552-558, 1996.
- [16] T. Kohonen, “The self-organizing map”, Neurocomputing, vol. 21, no. 1-3, pp. 1-6, 1998.
- [17] H. Huang, Y. Liu, L. Liu and C. Wong, “EMG Classification for Prehensile Postures Using Cascaded Architecture of Neural Networks with Self-Organizing Maps”, 2003 IEEE International Conference on Robotics & Automation, 2003.
- [18] C. Christodoulou and C. Pattichis, “Unsupervised pattern recognition for the classification of EMG signals”, IEEE Transactions on Biomedical Engineering, vol. 46, no. 2, pp. 169-178, 1999.
- [19] Gunther and S. Fritsch, “neuralnet: Training of Neural Networks”, The R journal, vol. 2, no. 1, pp. 30-38, 2010.

- [20] J. Heaton, Programming neural networks with Encog3 in Java.
- [21] L. Fausett, Fundamentals of neural networks. Englewood Cliffs, NJ: Prentice-Hall, 1994
- [22] C. Igel and M. Hsken, "Empirical evaluation of the improved Rprop learning algorithms", Neurocomputing, vol. 50, pp. 105-123, 2003.
- [23] A. Nicholson, "A Beginner's Guide to Recurrent Networks and LSTMs - Deeplearning4j: Open-source, distributed deep learning for the JVM", Deeplearning4j.org, 2016. [Online]. Available: <http://deeplearning4j.org/lstm.html>. [Accessed: 01- Aug- 2016].
- [24] J. Szkola, K. Pancarz and J. Warchol, "Recurrent Neural Networks in Computer-Based Clinical Decision Support for Laryngopathies: An Experimental Study", Computational Intelligence and Neuroscience, vol. 2011, pp. 1-8, 2011.
- [25] M. Dweib, "Optimal Architecture Setting of Learning Parameters Based on Counter Back-Propagation Neural Networks", ARPN Journal of Science and Technology, vol. 5, no. 12, pp. 616-619, 2015.
- [26] J. Heaton, "Encog Library of Interchangeable Machine Learning Models for Java and C#", Journal of Machine Learning Research, vol. 16, pp. 1243-47, 2015.
- [27] E. Gokgoz and A. Subasi, "Comparison of decision tree algorithms for EMG signal classification using DWT", Biomedical Signal Processing and Control, vol. 18, pp. 138-144, 2015.