# Gated Recurrent Neural Networks for EMG-Based Hand Gesture Classification: A Comparative Study

Ali Samadani

***Abstract*— Electromyographic activities (EMG) generated during contraction of upper limb muscles can be mapped to distinct hand gestures and movements, posing them as a promising modality for prosthetic and cybernetic applications. This paper presents a comparative analysis between different recurrent neural network (RNN) configurations for EMG-based hand gesture classification. In particular, RNNs with recurrent units of long short-term memory (LSTM) and gated recurrent unit (GRU) are evaluated. Furthermore, the effects of an attention mechanism and varying learning rates are evaluated. Results show a classifier 1) with a bidirectional recurrent layer composed of LSTM units, 2) that applies the attention mechanism, and 3) trained with step-wise learning rate outperforms all other tested RNN classifiers.**

## I. Introduction

Hand is the most dexterous of the human limbs and is central to execution of physical, functional, communicative, expressive activities. Conventional hand-gesture classification models are mainly based on modalities captured either via cameras, inertial measurement units (IMUs), or data gloves. Cameras require an unobstructed light-of-sight, and IMUs and datagloves need to be directly attached to the fingers for accurate tracking of hand joints, all of which, hamper their performance in naturalistic interactions. Furthermore, these technologies while capturing the motion, do not provide any information about the kinetic dimension of the hand motions, which might be important for discriminating between different gestures especially their expressivity (e.g., a forceful hand clenching vs a weak fist).

Electromyographic signals (EMG) captured at arm and forearm provide a wealth of information about corresponding hand gestures [1], [2]. EMG-based classification of hand gestures has applications ranging from prostheses to cybernetics applications for remote control and commanding of personal devices. EMG sensors do not require a line-of-sight or a direct attachment to the hand, allowing natural finger movements. However, EMG suffers from some key difficulties: 1) high amount of noise and contamination with motion artifacts, 2) significant spatiotemporal variability resulting in within-class variable-length and unaligned signals, 3) the signal quality and strength are affected by subcutaneous fat, muscle fiber composition, and the amount of hair. Therefore, an EMG-based classifier should account for the time-series and class-specific multi-modal nature of EMG signals.

There are two main approaches to representing time-series observations for classification: 1) temporal encoding, and 2) meta-feature representation. The temporal approaches include functional time-series representation [3], stochastic feature transformaiton and abstraction [4], and recurrent neural networks [5]. The temporal approaches characterize the dynamics and temporal progression of observations, whereas approaches based on meta-features use a set of user-defined scalar features that 1) are usually chosen in an ad-hoc manner, 2) do not necessary capture the temporal variations salient to the classification task, 3) do not explicitly account for stochastic and interpersonal variabilities, and 4) are limited to same-length and landmark-aligned observations in order to produce homogeneous features across the observations. Previous studies have reported on the performance of the feature-based approaches for EMG classification [6].

This paper presents a comparative analysis of RNN classifiers with LSTM and GRU units for hand gesture classification based on EMG signals. Furthermore, the paper evaluates the efficacy of an attention mechanism and a step-wise learning rate to improve the classification performance.

A. Samadani is with Philips Research North America, Cambridge, MA, USA.

## II. Background

Recurrent neural networks (RNNs) are an extension of feed-forward networks especially designed to handle sequential data. RNNs incorporate a hidden recurrent state whose value depends on the previous step in time. For an input sequence $\mathbf{x} = \left[\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_t\right]$ fed to a recurrent unit, the output at every time step $t$, $\mathbf{h}_t$ is computed as:

$$\mathbf{h}_t = \begin{cases} 0, & t = 0 \\ \phi(\mathbf{h}_{t-1}, \mathbf{x}_t), & \text{otherwise,} \end{cases}$$

where $\phi$ is an activation function (e.g., logistic sigmoid). The mapping, conventionally, is of the form $\mathbf{h}_t = \phi(\mathbf{W}\mathbf{x}_t + \mathbf{U}\mathbf{h}_{t-1})$, where $\mathbf{W}$ and $\mathbf{U}$ are weight matrices. As such, the output of a recurrent unit at time $t$ is a probability distribution over the next element of the sequence. Training RNNs using back-propagation is difficult due to issues related to gradient computation (mainly, vanishing, but sometimes, exploding gradients) [7] that obscure long-term dependencies potentially salient to the classification task. Two main variations of recurrent units were introduced: 1) long-short-term-memory units (LSTMs) [8], and 2) gated recurrent units (GRUs) [9] to address the problem of vanishing gradient incurred during back-propagation through time. These more sophisticated units modify their activation function to include an affine transformation followed by an element-wise non-linearity using a set of gates. Due to their ability to encode temporal progression in variable-length observations and capture long-term dependencies, RNNs

978-1-5386-3646-6/18/$31.00 ©2018 IEEE

with LSTM or GRU units have been shown to perform well in tasks such as speech recognition e.g., [10].

In the following, $\odot$ is used to refer to Hadamard product, $\mathbf{x}_t$ and $\mathbf{h}_t$ denote input and output to a layer, and $\mathbf{W}$, $\mathbf{U}$, and $\mathbf{V}$ denote weight matrices associated with input, output, and memory variables, respectively. Sigmoid and hyperbolic tangent activations are denoted as $\sigma$ and $\tanh$, respectively. The subscripts $i, f$, and $o$ denote parameters associated with input, forget, and output gates. In this paper, the LSTM and GRU implementations as used in [5] are used.

### A. Long Short Term Memory

An LSTM unit maintains an internal memory cell for which the write, reset, and read actions are governed by three separate multiplicative gates of input ($i$), forget ($f$), and output ($o$) gates. The LSTM output as time $t$ is computed as

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t), \tag{1}$$

where $\mathbf{o}_t$ is the output gate variable at time $t$ modulating the amount of exposure to the memory $\mathbf{c}_t$ and is computed as

$$\mathbf{o}_t = \sigma\big(\mathbf{W}_o\mathbf{x}_t + \mathbf{U}_o\mathbf{h}_{t-1} + \mathbf{V}_o\mathbf{c}_t\big). \tag{2}$$

The memory gate variable at time $t$ (denoted as $\mathbf{c}_t$) is computed as

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \widetilde{\mathbf{c}}_t, \tag{3}$$

where $\mathbf{f}_t$ is the forget gate variable that controls the amount of existing memory carried forward and $\widetilde{\mathbf{c}}_t$ is the new memory content modulated by the input gate variable $\mathbf{i}_t$

$$\widetilde{\mathbf{c}}_t = \tanh(\mathbf{W}_c\mathbf{x}_t + \mathbf{U}_c\mathbf{h}_{t-1}). \tag{4}$$

As can be seen, an LSTM unit can carry an important feature from earlier steps in an input sequence forward; hence, capturing long-distance dependencies. The input and forget gate variables are computed as

$$\mathbf{i}_t = \sigma(\mathbf{W}_i\mathbf{x}_t + \mathbf{U}_i\mathbf{h}_{t-1} + \mathbf{V}_i\mathbf{c}_{t-1}) \tag{5}$$

$$\mathbf{f}_t = \sigma(\mathbf{W}_f\mathbf{x}_t + \mathbf{U}_f\mathbf{h}_{t-1} + \mathbf{V}_f\mathbf{c}_{t-1}). \tag{6}$$

In Equations 2 and 5, the $\mathbf{V}$ matrices are diagonal.

### B. Gated Recurrent Unit

Gated recurrent units (GRUs) capture long-term dependencies in sequential observations $\mathbf{x}$ using an update gate $z$. The output of a GRU unit at time $t$ is

$$\mathbf{h}_t = (1 - \mathbf{z}_t) \odot \mathbf{h}_{t-1} + \mathbf{z}_t \odot \widetilde{\mathbf{h}}_t, \tag{7}$$

where $\mathbf{z}_t$ is the update gate variable controlling the extent of update for the GRU output at each time step $t$.

$$\mathbf{z}_t = \sigma(\mathbf{W}_z\mathbf{x}_t + \mathbf{U}_z\mathbf{h}_{t-1}) \tag{8}$$

The candidate output $\widetilde{\mathbf{h}}_t$ is computed as

$$\widetilde{\mathbf{h}}_t = \tanh(\mathbf{W}_o\mathbf{x}_t + \mathbf{U}_o(\mathbf{r}_t \odot \mathbf{h}_{t-1})), \tag{9}$$

where $\mathbf{r}_t$ is the reset gate variable at time $t$; an $\mathbf{r}_t$ close to zero erases the contribution of the previously computed state in $\widetilde{\mathbf{h}}_t$. The reset gate variable is computed as

$$\mathbf{r}_t = \sigma(\mathbf{W}_r\mathbf{x}_t + \mathbf{U}_r\mathbf{h}_{t-1}). \tag{10}$$

Unlike LSTMs, GRUs completely expose the memory in each state update. Both the GRU's update gate and the LSTM's forget gate allow for carrying the older values of a feature forward and preventing it from being overwritten if found important. Furthermore, they provide a balanced mix of new and old states of the feature (Equations 1 and 7). Finally, the gating operation helps addressing the problem of vanishing gradient via bypassing unimportant time steps.

### C. Attention Mechanism

It is often the case that we are dealing with variable-length sequential datasets for which not all the sequential samples are of equal importance to the classification task. The attention mechanism can be used to identify segments of sequentional observations most salient to the classification task [11]. In this work, a simplified version of attention mechanism based on the feed-forward network is used [12]. Given a recurrent unit whose output ($\mathbf{h}_t$) is updated at every time step, the attention mechanism uses attention over recurrent unit's encoder as follows

$$n = \Sigma_{t=1}^{T}\alpha_t\mathbf{h}_t, \quad \text{where} \tag{11}$$

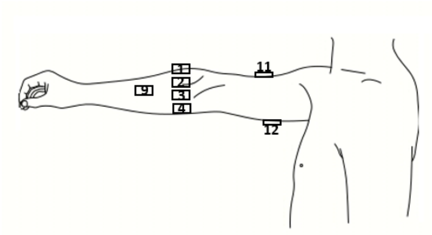
$$\alpha_t = \text{softmax } (e_t), \text{ and } e_t = a(\mathbf{h}_t), \tag{12}$$

where $n$ is the context variable for the entire sequence of input and $a$ is a learnable function (e.g., feed forward function). In this configuration, the attention mechanism can be regarded as mapping the state sequence $\mathbf{h}$ into $n$. Consequently, information from different time steps are weighted based on their importance and aggregated into $n$.

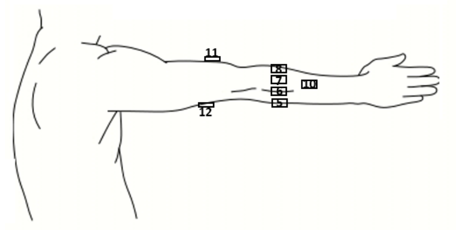
## III. Experimental Design

In this work, the publicly-available NinaPro hand gesture dataset (NinaPro2) was used [13]. The NinaPro2 dataset includes surface EMG signals from 40 healthy intact subjects (28 males, 12 females; 35 right-handed, 5 left-handed; age 29.9±3.9 years) performing 49 hand gestures plus rest. Figure 1.a-b shows the placement of 12 EMG sensors on the arm and forearm. There are eight equidistant EMG electrodes wrapped around the forearm at the height of the radio humeral joint (electrodes 1 to 8), two electrodes are used to capture finger flexor/extensor contractions at flexor digitorum superficialis and of the extensor digitorum superficialis (electrodes 9 and 10), and the remaining two electrodes are placed on the belly of biceps brachii and the triceps brachii muscles(electrodes 11 and 12) [6]. The subjects were instructed via a visual cue (movies depicting the movements) on the screen to perform the set of gestures with their right hand. Each gesture was repeated 6 times with the repetitions being separated by a rest interval of 3 seconds. In this work, the basic, isometric, and isotonic gestures plus rest (a total of 18 gestures Figure 1.c-d) from right-handed subjects were used. Further details about the dataset and acquisition protocol can be found in [13].

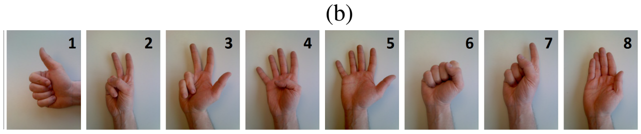
As for the experimental setup, RNNs with LSTM and GRU units were evaluated. Furthermore, the effects of attention mechanism, unidirectional versus bidirectional recurrent layers, and constant versus step-wise learning rates



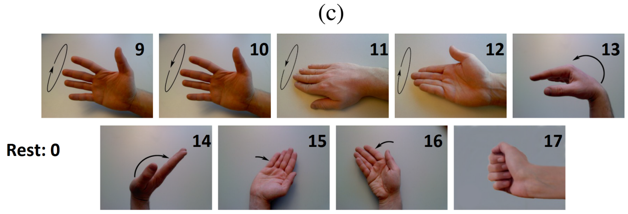
(a)



(b)




(c)




(d)

Fig. 1: (a) and (b) EMG sensor placement, (c) Isometric, isotonic gestures,(d) Gestures involving basic movement of the wrist (adduction/abduction, flexion/extension, and pronation/supination). Gesture 0 is the rest gesture.

(Figure 2) on the classification performance were examined. To establish a fair comparison between different classifiers, except the components mentioned above, all other network configurations were kept the same across the classifiers (Table I). The size of the models were kept small to avoid over-fitting as previously done in [5]. There were a total number of 16 models tested: 8 with LSTM and 8 with GRU units (Table I).

Raw EMG signals were preprocessed to extract signal envelopes by: 1) centering EMG signals to remove any DC offset in the signals, 2) rectifying the centered signals (taking the absolute value of the signals), 3) low-pass filtering of the centered and rectified signals [1]. To evaluate the performance of the classifiers, samples of each gesture were randomly divided into train (50%), validation (16.7%), and test(33.3%) sets for each subject, resulting in a total of 1890, 630, 1260 train, validation, and test samples, respectively. At the end of the training phase, the best performing model was selected using the validation set and the selected model was then tested using the left-out test samples. The models were trained using the Adam optimizer [14]. The models all used a recurrent layer with 128 recurrent unit (LSTM or GRU).

TABLE I: RNN models evaluated in this work. The last two columns show the number of parameters for each classifier. LR: learning rate

| Network configuration | | | Recurrent unit | |
|---|---|---|---|---|
| Direction | Attention | LR | LSTM | GRU |
| Unidirectional | 0 | 0.01 | 82,386 | 64,338 |
| Unidirectional | 1 | 0.01 | 82,781 | 64,733 |
| Unidirectional | 0 | Stepwise | 82,386 | 64,338 |
| Unidirectional | 1 | Stepwise | 82,781 | 64,733 |
| Bidirectional | 0 | 0.01 | 163,282 | 127,186 |
| Bidirectional | 1 | 0.01 | 163,805 | 127,709 |
| Bidirectional | 0 | Stepwise | 163,282 | 127,186 |
| Bidirectional | 1 | Stepwise | 163,805 | 127,709 |

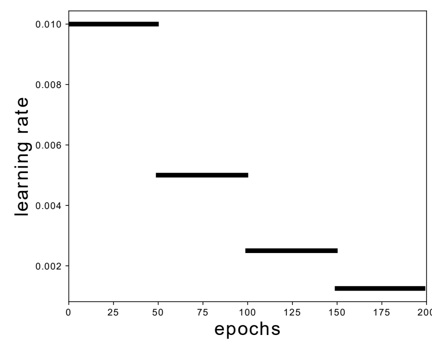



Fig. 2: The step-wise learning rate.

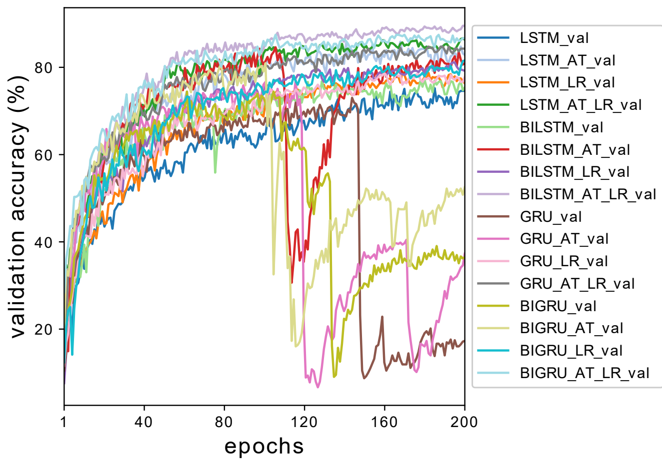
The recurrent layer was either unidirectional or bidirectional, followed by an attention layer in some cases (Table I). Batch normalization was applied on the output of the recurrent layer at each batch. This was then followed by a dense layer with the softmax activation.

## IV. Results

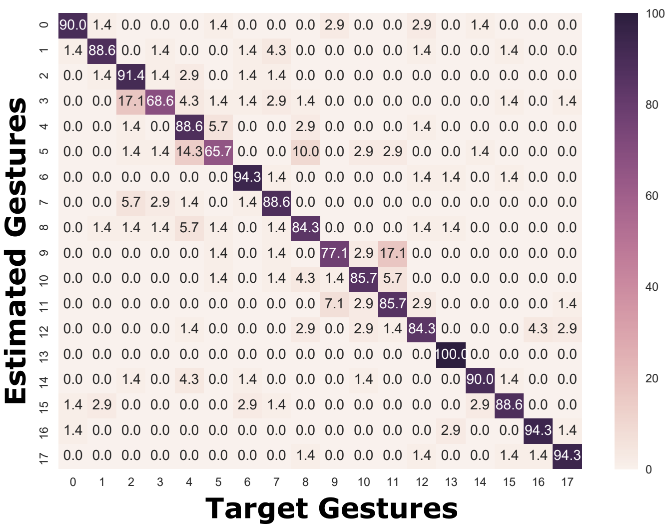
Figure 3.a shows the learning curves for the validation set. The network with bidirectional LSTM units, attention mechanism, and the step-wise learning rate (BILSTM-AT-LR) outperformed other configurations with a multi-class validation classification accuracy of 89.5%. Therefore, the BILSTM-AT-LR model was selected as the best performing model and its performance was evaluated using the left-out test set. The BILSTM-AT-LR model achieved a multi-class test classification accuracy of 86.7%. Figure 3.b shows the BILSTM-AT-LR confusion matrix for the 18 gestures in the test set. Higher gesture-specific classification rates can be achieved using a deeper network with multiple recurrent and dense layers. Since, the main purpose of this paper was to compare different RNN configurations, size of the networks were intentionally kept small. Note that the RNN configuration allows encoding variable-length sequential observations.

## V. Discussion

Among the RNN classifiers, the one with a bidirectional recurrent layer composed of LSTM units, attention mechanism, and the step-wise learning rate outperforms other




(a)




(b)

Fig. 3: a) Learning curves based on the validation set (_val) for different RNN classifiers, b) Confusion matrix for the test set (%) using the BILSTM-AT-LR classifier.

configurations. Looking at the learning curves (Figure 3.a), it is clear that the addition of attention improved the performance almost in all the cases as compared with the models with no attention. Furthermore, the step-wise learning rate helped parameter optimization resulting in more accurate models. Finally, bidirectional recurrent layers resulted in more accurate classifiers indicating the discriminative value of exploiting information in both directions in encoding the hand gestures. As such, the combination of the three, bidirectional recurrent layer with LSTM units, attention mechanism, and the step-wise learning rate resulted in the best classifier as evaluated using the validation set. Using this classifier, an 18-class classification accuracy of 86.7% on the left-out test set was achieved.

The RNN classifiers with GRUs exhibited unstable learning curves starting around epoch 100, whereas same networks trained with the step-wise learning rate did not demonstrate the unstable behavior. This indicates the importance of proper choice of optimizer and optimization parameter for GRU training. To the best of our knowledge, this work presents the first attempt to RNN classification of hand gestures in NinaPro dataset and examining different gated RNN configurations for EMG-based hand gesture classification.

## VI. Conclusion

In this paper, different configurations of recurrent neural networks with the gated units of LSTM and GRU were evaluated for the task of EMG-based hand-gesture classification. Results show that networks with LSTM units surpassed those with GRUs. Furthermore, the addition of a backward recurrent layer (bidirectional LSTM), attention mechanism, and step-wise learning rate all proved to enhance the hand-gesture classification performance. Additionally, our experiments demonstrated the sensitivity of the RNNs with GRU units to constant learning rates, which merits further exploration in the future to identify the optimizer and optimization parameter settings that best serve the classification performance.

## References

[1] A. Samadani and D. Kulic, "Hand gesture recognition based on surface electromyography," in *EMBS*, 2014, pp. 4196–4199.

[2] M. Atzori, A. Gijsberts, C. Castellini, B. Caputo, A.-G. M. Hager, S. Elsig, G. Giatsidis, F. Bassetto, and H. Müller, "Electromyography data for non-invasive naturally-controlled robotic hand prostheses," *Scientific data*, vol. 1, p. 140053, 2014.

[3] A. Samadani, A. Ghodsi, and D. Kulic, "Discriminative functional analysis of human movements," *Pattern Recognition Letters*, vol. 34, no. 15, pp. 1829 – 1839, 2013, smart Approaches for Human Action Recognition. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167865512004242

[4] A. Samadani, R. Gorbet, and D. Kulic, "Affective movement recognition based on generative and discriminative stochastic dynamic models," *IEEE Trans. HumanMach. Syst*, vol. 44, no. 4, pp. 454–467, 2014.

[5] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling," *ArXiv e-prints*, Dec. 2014.

[6] M. Atzori, A. Gijsberts, I. Kuzborskij, S. Elsig, A. G. M. Hager, O. Deriaz, C. Castellini, H. Mller, and B. Caputo, "Characterization of a benchmark database for myoelectric movement classification," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 23, no. 1, pp. 73–83, 2015.

[7] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Trans. Neural Netw.*, vol. 5, no. 2, pp. 157–166, Mar 1994.

[8] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[9] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," *CoRR*, vol. abs/1409.1259, 2014. [Online]. Available: http://arxiv.org/abs/1409.1259

[10] A. Graves, A. r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *ICASSP*, 2013, pp. 6645–6649.

[11] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *CoRR*, vol. abs/1409.0473, 2014. [Online]. Available: http://arxiv.org/abs/1409.0473

[12] C. Raffel and D. P. W. Ellis, "Feed-forward networks with attention can solve some long-term memory problems," *CoRR*, vol. abs/1512.08756, 2015. [Online]. Available: http://arxiv.org/abs/1512.08756

[13] M. Atzori, A. Gijsberts, S. Heynen, A. G. M. Hager, O. Deriaz, P. van der Smagt, C. Castellini, B. Caputo, and H. Mller, "Building the ninapro database: A resource for the biorobotics community," in *BIOROB*, June 2012, pp. 1258–1265.

[14] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014. [Online]. Available: http://arxiv.org/abs/1412.6980