Predicting Keystrokes from Electromyography Signals

Nick J. C. Smith
Undergraduate Electrical Engineering Student
University of California, Los Angeles
nickjcsmith@g.ucla.edu

Abstract

Electromyography (EMG)-based keystroke prediction has the potential to allow for cutting-edge technologies regarding human-computer interaction. In this study, we explore deep learning-based approaches for predicting keystrokes from EMG signals, focusing on the primary objective of minimizing Character Error Eate (CER). We implemented and evaluated three model architectures: (1) a baseline convolutional model using a Time-Depth Seperable Encoder (TDSConv), (2) recurrent models based on Gated Recurrent Units (GRU) and Long Short-Term Memory (LSTM) networks, and (3) transformer-based architectures. All models were trained using Connectionist Temporal Classification (CTC) loss, allowing for keystroke sequence prediction without explicit frame alignment. The data used for the training-validation-testing split was collected from a single user. Experimental results show that the GRU model achieved the lowest CER, with a score of 14.977 on the test set, compared to 15.085 for the LSTM. Both of the models outscored the baseline test CER score of 21.828 set by TDSConv under the same training parameters. While LSTM scored lower in terms of deletion rates, GRU obtained lower substitution and insertion error rates, which indicates better overall sequence prediction accuracy. Transformer-based models did not yield improvements in the reduction of CER, suggesting that for training and prediction situations conducted on a single user, recurrent networks are more suited. These results provide insights into optimizing deep learning models for keystroke prediction from EMG signals.

1 Introduction

In this study, we focus on minimizing Character Error Rate (CER) within a single subject for EMG-based keystroke prediction. While prior work in surface electromyography (sEMG) decoding has explored generalization across multiple users, achieving low CER within an individual subject has the potential to be very beneficial in the development of highly accurate, personalized text entry systems. The ability for personalization in sEMG applications is important due to the high variance in muscle activation patterns across multiple subjects, which can lead to poor model generalization. In addressing this problem, we implement and evaluate deep learning architectures optimized for single-subject performance, comparing convolutional and recurrent based models. It is our hypothesis that the ability of these recurrent networks to model long-term dependencies in EMG signals will result in better performance than the convolutional encoders. Additionally, we implement and test transformer-based models to assess their effectiveness in this context. The models will be evaluated using five different metrics: (1) Character Error Rate (CER), (2) Deletion Error Rate (DER), (3) Insertion Error Rate (IER), (4) Substitution Error Rate, and (5) Loss. These metrics will be compared across different architectures with the goal of identifying the most effective model for minimizing CER in a single subject scenario. The findings of this study provide insights into the trade-offs between various deep learnig approaches for EMG-based keystroke prediction and contribute to the development of reliable EMG decoding modules.

2 Methods

2.1 Data preprocessing

2.1.1 Spectrogram normalization

EMG signals are transformed into spectrogram representations and normalized using **SpectrogramNorm**, which applies a 2D batch normalization to each electrode channel per band. This ensures that inputs maintain a standardized scale across different sessions.

2.1.2 Multi-band feature extraction

The **MultiBandRotationInvariantMLP** processes EMG signals across two frequency bands, applying rotation-invariant multi-layer perceptrons to extract robust features from each band independently.

2.1.3 Dataset handling

The the preprocessed EMG data is organized using **WindowedEMGDataset**, which provides training and validation data with fixed segment lengths and optional padding, while test data is provided as full sessions.

2.2 Baseline convolutional model

The repository for the emg2qwerty dataset included a baseline model to serve as a reference for function and to set goals for metric improvement with more advanced architectures. This model employs a stack of two-dimensional convolutional blocks, each composed of temporal depth-separable convolution to capture local temporal dependencies, layer normalization to stabilize training, a fully connected block to refine the learned feature representations, and an output layer to map the features to keystroke logits. The model is trained using Connectionist Temporal Classification (CTC) loss.

2.3 Recurrent neural network models

Both of the following GRU and LSTM models are implemented using the **RecurrentEncoder** module, which consists of a stacked recurrent neural network (RNN) with bidirectional processing. The models are evaluated on the CTC loss framework, allowing keystroke sequences to be inferred from EMG inputs without requiring explicit frame-to-character alignment.

2.3.1 Gated recurrent unit (GRU) model

The GRU model captures sequential dependencies in the EMG signal while maintaining computational efficiency. Processed features from **MultiBandRotationInvariantMLP** are flattened and passed into the GRU encoder. The **RecurrentEncoder** module contains bidirectional GRU layers with a hidden layer size specified by the hyperparameters. The final recurrent outputs are mapped to logits through a fully connected block and a linear transformation to match the number of keystroke classes. The output is then transformed into a probability distribution over keystroke sequences using the Softmax function. Finally, training is performed using the CTC loss function, comparing predicted keystrokes with the true values.

2.3.2 Long-short term memory (LSTM) model

The LSTM model is almost identical in architecture to the GRU model, except the GRU layers are replaced with LSTM layers. The potential benefit of this change would be to retain long-term dependencies more effectively. A stacked bidirectional LSTM network processes the preprocessed input sequences, capturing both past and future dependencies in the EMG signal. Similar to the GRU model, the outputs of the LSTM are sent through a fully connected block and a linear transformation. Lastly, CTC loss is used to effectively train the model.

2.4 Hyperparameters

Both the GRU and LSTM models were trained with the same hyperparameter values, listed in the table below.

Table 1: GRU & LSTM hyperparameters

Name	Value
epochs	100
input_size	528
hidden_size	256
num_layers	2
optimizer	adam
learning_rate	10^{-3}
weight_decay	10^{-4}
dropout	0.3

3 Results

3.1 Baseline model

The values in the table below serve as a reference point for the error optimization achieved in this study. The most important values to note are the CER and loss values for both training and validation as comparison to these results will be present in the following sections.

Table 2: TDSConv metrics for 100 epochs

Name	Validation	Testing
Character Error Rate (CER)	18.742	21.828
Deletion Error Rate (DER)	1.3514	1.6856
Insertion Error Rate (IER)	4.8294	4.6034
Substitution Error Rate (SER)	12.561	15.539
Loss	0.8637	0.9792

3.2 Recurrent neural network models

3.2.1 Gated recurrent unit (GRU) model

Table 3: GRU metrics

Name	Validation	Testing
Character Error Rate (CER)	14.510	14.977
Deletion Error Rate (DER)	1.4400	2.1396
Insertion Error Rate (IER)	3.5224	1.9451
Substitution Error Rate (SER)	9.5481	10.893
Loss	0.5956	0.5911

CER The GRU-based model showed considerable improvements in CER compared to the baseline model using the TDSConv encoder. As displayed in the table below, the GRU model achieved a test CER of 14.977, a relative reduction of 31.39% in keystroke prediction error. This result is likely due to the ability of GR to capture sequential dependencies in EMG signals more than the convolution based baseline model.

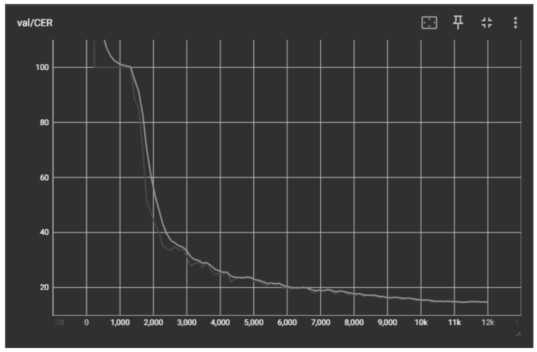

Figure 1: GRU validation CER vs epochs

Loss The GRU model showed a significant decrease in both validation and test loss compared to the baseline model, with a relative reduction of 31.04% and 39.63% respectively. This suggests that the recurrent GRU model resulted in better convergence and generalization.

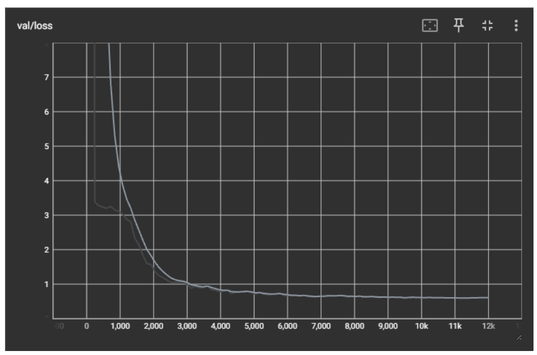

Figure 2: GRU validation loss vs epochs

Other error metrics The SER decreased from 15.539 to 10.893, indicating that the recurrent architecture led to more accurate character predictions. The IER was significantly lower in the GRU model (1.9451) compared to the baseline model (4.6034), which suggests the occurrence of fewer extraneous keystroke predictions. However, the DER increased slightly from 1.6856 to 2.1396, meaning that the GRU model resulted in a tradeoff where some valid keystrokes were omitted.

3.2.2 Long-short term memory (LSTM) model

Table 4: LSTM metrics

Name	Validation	Testing
Character Error Rate (CER)	14.555	15.085
Deletion Error Rate (DER)	1.8387	1.2967
Insertion Error Rate (IER)	2.8135	2.2693
Substitution Error Rate (SER)	9.9025	11.519
Loss	0.5513	0.5681

CER The LSTM model lowered test CER by 30.89% relative to the baseline model, which highlights its ability to learn long-term dependencies in EMG signals more effectively than the convolutional encoder.

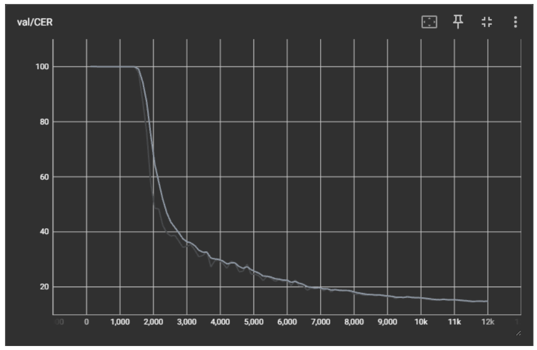

Figure 3: LSTM validation CER vs epochs

Loss The LSTM model achieved lower validation and test loss values (0.5513 and 0.5681, respectively) compared to the baseline (0.8637 and 0.9792, respectively), indicating better training convergence and generalization.

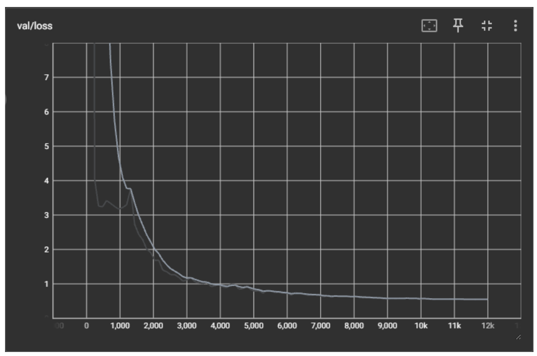

Figure 4: LSTM validation loss vs epochs

Other error metrics In terms of other error metrics, the LSTM model reduced the values of deletion, substitution, and insertion error rates. This suggests that the model was able to avoid omitting fewer valid keystrokes, substituting incorrect character predictions, and adding extraneous keystrokes.

4 Discussion

The results of this study show that recurrent neural networks, specifically GRU and LSTM architectures, outperform convolutional encoders in EMG-based keystroke prediction. This finding aligns with prior research into the ability of RNNs to effectively capture long-term dependencies in sequential data, while convolutional models such as TDSConv struggle due to their limited temporal receptive fields.[1] Both GRU and LSTM reduced Character Error Rate (CER) compared to the baseline model, with GRU achieving a slightly lower test CER (14.977) than LSTM (15.085). One idea supporting the success of the GRU model is that its simpler gating mechanism allows for faster convergence and improved generalization.[2] The LSTM resulted in a lower deletion error rate (DER), highlighting the ability of its separate memory cell structure to better retain temporal features.[3]

While transformer models have achieved great success in speech and NLP tasks, the models implemented for this study did not effectively reduce CER on the EMG dataset. One major factor in this result may be that the model was trained on an insufficient amount of data to enable

[1]Graves et al., 2006
[2]Chung et al., 2014
[3]Greff et al., 2017

the discovery of meaningful positional representations, highlighted by previous studies on the requirement of large datasets for transformer architecture.[4] Not only was this study conducted on data collected from a single user, EMG data is inherently lower in volume when compared to text or speech datasets and may have resulted in failure to generalize well.

Future directions The results of this study suggest that GRU and LSTM are the most effective architectures for EMG-based keystroke prediction, but there is still opportunity for improvement. Ideas for future work include hybrid CNN+RNN models with the goal of combining the convolutional spatial feature extraction with recurrent sequential learning. Also, increasing the size of the dataset by incorporating more than one user may allow for further exploration into the effectiveness of transformers in reducing CER for EMG keystroke prediction.

5 References

[1] Graves, A., Fernández, S., Gomez, F., & Schmidhuber, J. (2006). Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd International Conference on Machine Learning (ICML)*, pp. 369–376.

[2] Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R., & Schmidhuber, J. (2017). LSTM: A search space odyssey. *IEEE Transactions on Neural Networks and Learning Systems, 28*(10), 2222–2232.

[3] Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

[4] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30.

[4] Vaswani et al., 2017