# Using Recurrent Neural Networks to Classify EEG Data

Jonathan Bunton
405218799
j.bunton@ucla.edu

Eden Haney
005221845
ehaney@ucla.edu

Joshua Hannan
805221851
jhannan@g.ucla.edu

Yiming Zhou
205257089
yimingz0416@g.ucla.edu

## Abstract

*In this work, we analyze the classification of EEG data from the BCI Competition using several Recurrent Neural Network (RNN) architectures. In particular, we study simple Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) structures, and investigate equipping these networks with initial convolutional layers for increased accuracy. In addition to comparing network structures, we analyze several choices of data pre-processing, ranging from simple data windowing to wavelet transforms. We show that data windowing combined with an architecture composed of stacked CNN and RNN layers achieves 62% classification accuracy for Subject1 and 61% classification accuracy accross all subjects on the test data.*

## 1. Introduction

This work seeks to classify electroencephalogram (EEG) data of nine subjects while they perform four different tasks. In particular, we wish to use neural networks to classify which task a subject is performing based on this data.

### 1.1. Dataset

We analyze a portion of the BCI Competition data, which has been published for public use [2]. The portion of the data examined in this work consists of measurements taken from nine subjects, with each containing 1000 samples of EEG signal time-series (at 250 Hz) from 22 electrodes placed across the subject's head. Each set of time series signals corresponds to one of four actions that the subject completed during the measurement window.

### 1.2. Network Architecture

Each EEG trial in the dataset consists of 22 correlated time-series, naturally suggesting a network architecture capable of encoding and utilizing time-dependent information to perform classification. Recurrent Neural Networks (RNNs) are well suited for this task as they allow previous signals to repeatedly influence network activations.

We implemented two commonly used RNN architectures: Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU) and investigated the effects of modifying these architectures, for example, by using different hidden layers and varying dropout. In addition, we tested several new configurations, combining convolutional neural networks (CNNs) and RNNs. Detailed explanations of these architectures are found in Section 3.1.

### 1.3. Augmentation and Pre-processing

We found that training on unprocessed EEG data with a simple RNN performed poorly with a testing accuracy around (25%). We sought to improve this, and uncover underlying latent features with preprocessing.

To uncover the underlying latent features in the EEG data, we implemented several data augmentation and preprocessing techniques, including windowing the data, performing the Short-time Fourier transform (STFT), Continuous wavelet transform (CWT), and using a k-nearest neighbor voting scheme.

#### 1.3.1 Windowing

To increase amount of data available for training, we preprocessed the data using windowing. We selected a fixed window size and subsampled each trial across a fixed stride in time. We then used each of these windows as a new, independent trial. Windowing the data allowed us to evaluate the classification accuracy as a function of time.

#### 1.3.2 Short-Time Fourier Transform

One way to analyze EEF data is to examine its frequency content through the Fourier Transform. While providing excellent frequency domain detail, the Fourier Transform sacrifices all time-dependent information carried by the signal.

One typical method to counteract this issue is to apply the Short-Time Fourier Transform (STFT). The STFT process begins with a simple Fourier Transform of the entire time-series, then transforms progressively shorter segments. The resulting output data is a set of frequency con-

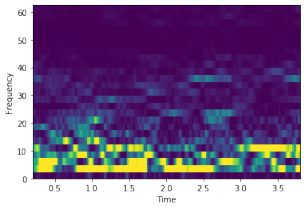


Figure 1. STFT of Trial 1, Electrode 1.

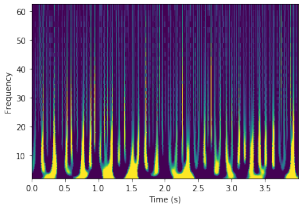


Figure 2. CWT of Trial 1, Electrode 1 with the "mexican hat" wavelet.

tents at various points in time, across the input data, converting a one-dimensional time-series into a combination of times and frequencies, shown in Fig. 1. This new frequency data can be interpreted as a new set of useful latent features, which are then fed into the RNN.

#### 1.3.3 CWT

Kiymik *et al.* [4] compared the STFT with a continuous wavelet transform (CWT) on EEG data by transforming the data and the reconstructing it. They found that while the STFT had a shorter process time, the CWT gave more accurate results. In addition, prior work [5] has seen some benefit to using this transform in conjunction with a CNN, as the resulting data appears as an "image" (see figure 2) .

Rather simple Fourier Transforms as in the STFT, the CWT computes how much of a localized *mother wavelet* lies within the data as it is translated across the time-series. The mother wavelet is scaled to various widths to capture a range of local frequency behaviors, creating joint time-frequency data (see Fig. 2).

#### 1.3.4 ICA

Independent component analysis (ICA) is a preprocessing method used by experts in processing EEG data [1]. Described by Hyvärinen [3], it is a decomposition method that involves a linear change of basis from data collected by individual electrodes to one that results in maximally temporally independent signals.

#### 1.3.5 K-Nearest Neighbor Voting

Similarly to what we have done in the augmentation of training data, we can also window the data to get several sub-sequences from a single sequence during testing. Then passing the $k$ sub-sequences to the network gives us $k$ results. Finally we do a voting among the $k$ results and get the class label with the highest score.

## 2. Results

We performed extensive testing to investigate the tradeoffs between our different architectures and various data augmentation techniques. We achieved 62% classification accuracy for subject 1 using our Model13 architecture (see Table 4) training on all subjects and 61% classification accuracy for all subjects using our Model13 architecture. We found that our Model13 architecture consistently demonstrated the best accuracy across different data augmentation techniques. We present details of these results in section 5.

## 3. Discussion

### 3.1. Architectures

We found that networks that combined CNNs with RNNs consistently yielded the highest accuracy, as shown in Table 2. Adding CNNs before RNNs helps by effectively pre-processing the data and extracting the salient features as the input of the RNN. Model13 had the highest accuracy across all subjects and subject1 alone. The main reason Model13 outperforms others is that it did convolution on both temporal and spatial dimensions. The convolution on spatial dimension is crucial here since it also extracted spatial features of the 22 electrodes while other models neglected this dimension.

GRU-based RNNs performed better than their LSTM-based RNNs counter-parts in all of our tests (Table 2). GRUs have two gates (instead of three like LSTMs) and consequently have fewer parameters to train and require less data to learn compared to LSTMs. As a result, GRUs are more accurate on our EEG dataset, which has relatively few samples (particularly when trained on just Subject1's data).

Adding more than two layers to an RNN generally increased its classification accuracy. This may be due to deeper models having a higher capacity for capturing local nonlinearities while learning long term features. Determining the depth of stacked RNNs is an active field of research and good source of future work for EEG classification [6].

We found that incorporating relatively high dropout rates (50 - 60) and mild $L_2$ regularization was necessary to prevent overfitting. We chose a relatively low number of hidden units for our RNNs to improve its generalizability to unseen EEG data.

### 3.2. Data Augmentation

#### 3.2.1 Windowing

Data windowing (or subsampling) proved to be a useful tool in producing more effective classifiers. We found windows allowed the RNN to actually see the data as localized time segments rather than as somewhat opaque, self-contained time-series. Another reason for this improvement is its ability to increase the number of training samples, since networks with more training data often produce better classification accuracy.

One major downfall of this augmentation, however, is that the resulting windowed samples are highly correlated,

leading to possible overfitting, which we often saw in training. By examining various window parameters (i.e. width and stride), we attempted to mitigate these issues.

Adjusting the size of this window also changes the length of the time-series used for classification, so we may use it as an indicator of the ideal length of EEG data for our network architecture. Across various architectures, smaller windows with small strides produced the highest all-subject test accuracy (see Figs. 3, 4, and 5).

This trend is likely due to these parameters greatly increasing the amount of training data, while too small of a window reduces the information quality of each sample drastically. However, larger window sizes produced the best accuracy for Model13 on Subject1. This is likely due to the natural windowing in Model13's layers, meaning the data may be over-windowed (see fig 5).

#### 3.2.2 STFT

Despite the joint time-frequency content of data preprocessed with the STFT, we found that resulting classification accuracy was fairly low regardless of the choice of window or strides. This is shown in Table 3, where varying window and stride parameters produced generally low results, with few fairing better than a random guess.

One major shortcoming of the STFT as opposed to other frequency-domain approaches is its relatively low time resolution for lower frequencies. This low amount of temporal information causes data processed with STFT seems to be a poor match for RNN architectures.

#### 3.2.3 CWT

As seen in Fig. 2 and discussed in Section 1.3.3, given a times series input the CWT outputs a 2-D frequency and time series output. The number of frequencies examined, or number of levels, was restricted by the computational complexity of handling the processed data. We tried to incorporate these extra frequency dimensions into our data two different ways.

First, we added the new frequency axes from each trial as extra features, creating input data of size (num Samples x (num Electrodes x num CWT Levels) x Time). This increased the complexity of our data, leading to poor performance and overfitting due to the small number of samples.

In our second attempt, we incorporated each frequency's data as a new trial. This improved performance on the basic GRU architecture (Model7), potentially because it artificially increased the number of trials. Moreover, the CWT acts similarly to a "well-informed convolution" which may assist in identifying latent features in the data. Performing simple data windowing on the transformed data improved accuracy further by again reducing sample complexity and increasing the number of training samples.

Prior work [5] noted that the region of interest for motor movement lies in the 7-30 Hz range, so we focused on this range when implementing the CWT. Indeed, with this second method we found that the performance was greatly affected by the frequency band. Table 6 shows the CWT's testing accuracy over a variety of frequencies when implemented on a basic GRU. We see that the frequencies that performed best were between 31Hz and 50Hz, resulting in a testing accuracy of 53%. We believe this could be increased if we consolidated these split trials with voting.

Interestingly, these results did not transfer to the other architectures we tested and 53% was the best accuracy we were able to achieve with this pre-processing method.

#### 3.2.4 ICA

We were unable to get any meaningful results using ICA preprocessed data. There are many possible reasons ICA was not able to decompose the data in a meaningful way including using too many/too few components and not bandpassing the data properly. Unfortunately, processing the data with ICA was cumbersome, limiting the number of experiments with it.

### 3.3. Subject 1 Accuracy

We ran several tests to evaluate the effect of training on all subjects compared to training on just Subject1 for classification on Subject1's EEG data, as shown in Figure 4. We found that in almost all tests, training on all subjects achieved better accuracy compared to training on just Subject1's data. Training on all subjects to classify Subject1's data functions as ensemble learning. By learning a more diverse set of features, models trained on all subjects exhibit better generalizability and thus, better Subject1 test accuracy. Model13 exhibited the best classification accuracy for Subject1, as shown in Figure 5.

### 3.4. All Subjects Accuracy

We found that Model13 achieved the best classification accuracy for all subjects on windowed data, as shown in Figure 5. As is explained in 3.1, the addition of several CNN layers in front of the RNN improved the model's ability to extract the salient features from the EEG data in order to successfully generalize to all subjects, which is necessary for the larger augmented dataset. Determining the optimal combination of CNN and RNN layers is a good source of future work.

## References


[1] https://sccn.ucsd.edu/wiki/EEGLAB#The_EEGLAB_Tutorial_Outline. 2

[2] C. Brunner, R. Leeb, G. Müller-Putz, A. Schlögl, and G. Pfurtscheller. Bci competition 2008–graz data set a. *Insti-*

*tute for Knowledge Discovery (Laboratory of Brain-Computer Interfaces), Graz University of Technology*, 16, 2008. 1

[3] A. Hyvärinen and E. Oja. Independent component analysis: algorithms and applications. *Neural Networks*, 13(4–5):411–430, Jun 2000. 2

[4] M. Kıymık, Güler, A. Dizibüyük, and M. Akın. Comparison of stft and wavelet transform methods in determining epileptic seizure activity in eeg signals for real-time application. *Computers in Biology and Medicine*, 35(7):603–616, Oct 2005. 2

[5] N. Yahya, H. Musa, Z. Y. Ong, and I. Elamvazuthi. Classification of motor functions from electroencephalogram (eeg) signals based on an integrated method comprised of common spatial pattern and wavelet transform framework. *Sensors*, 19(22):4878, 2019. 2, 3

[6] S. Zhang, Y. Wu, T. Che, Z. Lin, R. Memisevic, R. Salakhutdinov, and Y. Bengio. Architectural complexity measures of recurrent neural networks, 2016. 2

## 4. Architectures

| Network | Network Architecture |
|---|---|
| **Model1** | 2x LSTM (50 units)<br>FC Layer (4 units) |
| **Model2** | 3x LSTM (50 units)<br>FC Layer (4 units) |
| **Model3** | 5x LSTM (50 units)<br>FC Layer (4 units) |
| **Model4** | LSTM (50 units)<br>LSTM (25 units)<br>FC Layer (4 units) |
| **Model5** | Conv1d (40, 10)<br>2x LSTM (50 units)<br>FC Layer (4 units) |
| **Model6** | Conv1d (40, 10)<br>2x LSTM (50 units)<br>Conv1d (25, 10)<br>2x LSTM (20 units)<br>FC Layer (4 units)<br>*Dropout=0.6, $L_2$ reg=1e-03 |

| Network | Network Architecture |
|---|---|
| **Model7** | 2x GRU (50 units)<br>FC Layer (4 units) |
| **Model8** | 3x GRU (50 units)<br>FC Layer (4 units) |
| **Model9** | 5x GRU (50 units)<br>FC Layer (4 units) |
| **Model10** | GRU (50 units)<br>GRU (25 units)<br>FC Layer (4 units) |
| **Model11** | Conv1d (40, 10)<br>2x GRU (40 units)<br>FC Layer (4 units) |
| **Model12** | GRU (50 units)<br>2x GRU (40 units)<br>Conv1d (25, 10)<br>2x GRU (20 units)<br>FC Layer (4 units) |
| **Model13** | Conv2d (16, 1, 10)<br>Conv2d (32, 22, 1)<br>Conv2d (64, 1, 10)<br>Conv2d(128, 1, 10)<br>GRU(64 units)<br>GRU(32 units)<br>FC Layer (4 units)<br>*Dropout=0.4, $L_2$ reg=1e-02 |

Table 1. Architectures used to classify EEG Data. The order of layers for each model is given in the table top to bottom. FC = Fully Connected Layer. All models were trained using Adam optimizer with a learning rate of 1e-03. All RNNs used $\tanh$ activation functions. We used cross entropy loss for all models. Conv1d and Conv2d describe 1d-convolution and 2d-convolution respectively. Each convolution layer is described by the filter size, for Conv1d (channels, length) and for Conv2d (channels, height, width). Convolution layers are followed by batch normalization and ReLU activations (omitted in the table). We trained LSTM models with $L_2$ regularization=$10^{-5}$ and dropout=0.5 and trained GRU models with $L_2$ regularization=$10^{-4}$ and dropout=0.6 (unless noted otherwise in the table). We used different data augmentation techniques and test voting as described in Section 1.

## 5. Detailed Results

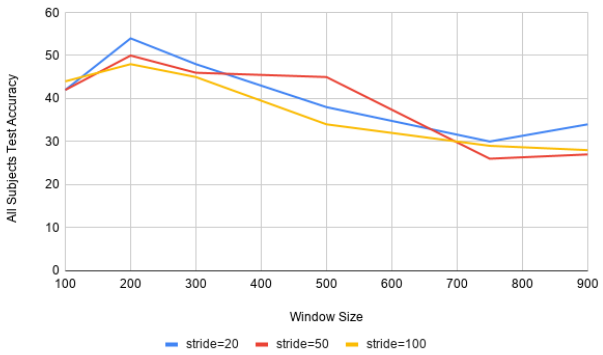


Figure 3. EEG classification test accuracy for all subjects, plotted as a function of window size and strides. Trained on the whole dataset using Model 11 (see Table 4.)

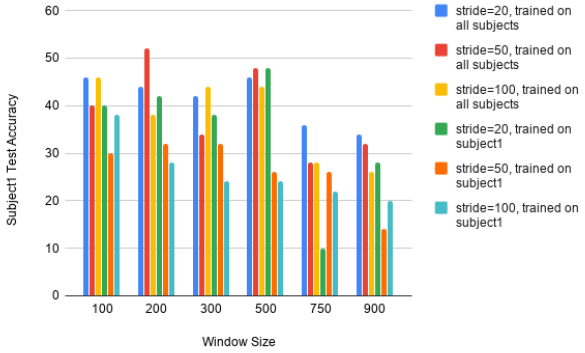


Figure 4. EEG classification test accuracy for Subject1, plotted as a function of window size and strides. Results from Model 11 (see Table 4) trained on all subjects and just Subject1, as indicated.

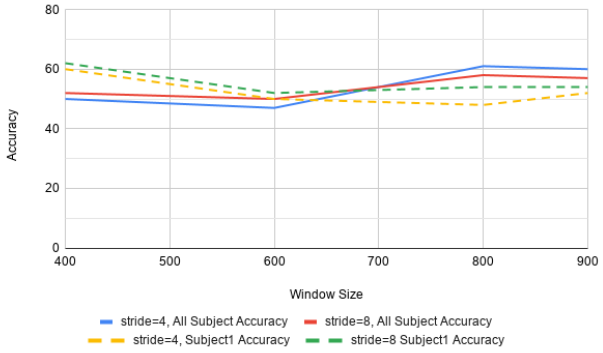


Figure 5. EEG classification test accuracy across all subjects and for subject1, plotted as a function of window size and strides. Trained on the whole dataset using Model 13 (see Table 4.)

| Model | Test Acc. |
|---|---|
| Model1 | 42% |
| Model2 | 43% |
| Model3 | 32% |
| Model4 | 42% |
| Model5 | 49% |
| Model6 | 52% |

| Model | Test Acc. |
|---|---|
| Model7 | 44% |
| Model8 | 47% |
| Model9 | 47% |
| Model10 | 44% |
| Model11 | 54% |
| Model12 | 46% |
| **Model13** | **61%** |

Table 2. EEG Classification Accuracy for all subjects for each architecture (trained with the parameters given in Table 1). All tests were run on the windowed data set, with a window size = 200 and stride = 20 (Model13: window size=800 and stride=4). Each architecture was trained for 20 epochs.

| Low Freq. | High Freq. | Test Acc. |
|---|---|---|
| 25Hz | 62.5Hz | 52% |
| 31Hz | 62.5Hz | 52% |
| 25Hz | 50Hz | 52% |
| **31Hz** | **50Hz** | **53%** |
| 40Hz | 50Hz | 45% |
| 25Hz | 40Hz | 49% |
| 31Hz | 40Hz | 48% |

Figure 6. CWT test accuracy against all subjects, divided by frequency range. Tested on Model7 with windowing (window size=200 and stride=20).

| Str/Win | 100 | 110 | 120 | 130 | 140 | 150 |
|---|---|---|---|---|---|---|
| **4** | 31% | 27% | 26% | 30% | 27% | 29% |
| **6** | 27% | 31% | 32% | 29% | 28% | 28% |
| **8** | 28% | 31% | 31% | 31% | 28% | 29% |

Table 3. Test accuracy for data pre-processed with the STFT using Model7 (see Table 1). The network was trained and tested on all subjects' data using the reported STFT stride (vertical) and window (horizontal).