Classification of Electroencephalogram (EEG) Data

Haoyu Luo
haoyu0730@g.ucla.edu

Shihe Pan
shihepan@ucla.edu

Xiaoman Cui
xiaomancui05@g.ucla.edu

Yiyang Wu
yiyangwu@ucla.edu

Abstract

Deep learning techniques allow people to extract information from electroencephalogram (EEG) data of brain activities. Given a specific EEG dataset, we implemented various CNN, RNN, hybrid CNN-RNN, and some unconventional architectures such as transformer and ResNet to do a 4-class classification task. Over 70% classification accuracy was achieved by using shallow CNN, hybrid CCN-LSTM, and hybrid CNN-Transformer architectures separately, among which the ***highest accuracy 76%*** *was reached. We also studied and gave insights into the impact of different subjects, sampling time windows, and dataset augmentation on the model performance.*

1. Introduction

The electroencephalogram (EEG) data comprises the synchronized activity of neurons near scalp electrodes. In the BCI Competition 2008, a set of EEG data was captured from 9 participants while they were imagining performing 4 different actions. The original dataset encompasses 2115 trials, each containing electrical signals from 22 electrodes across a 1000-bin temporal period. In this study, we implemented various CNN, RNN, hybrid CNN-RNN, and some unconventional architectures (e.g., transformer and ResNet) to do classification on this dataset.

1.1. CNN (shallow & deep)

We instantiated two convolutional neural network (CNN) architectures, with one having a greater depth than the other. CNN is believed to have the ability to automatically extract temporal features from the time-series data (e.g. EEG), which can then be utilized by subsequent layers in the network for higher-level abstraction. Therefore, our CNN architecture was designed with $[\text{conv} - \text{pool}] \times \text{N}$, followed by $[\text{affine}] \times \text{M}$, where N and M represent the number of repeated blocks. Dropout and batch normalization were inserted in each block to prevent overfitting and accelerate convergence.

By defining the height of the input data as the number of time bins and the width as 1, the CNN operates essentially in one-dimension (1D), even though a two-dimensional (2D) convolution function is employed. The depth of channels is then determined by the number of electrodes, which is 22 in our dataset. For this data input configuration, since the convolution operates only along the time variable, the network can mainly extract temporal but not spatial features.

1.2. RNN (vanilla & LSTM)

Besides CNN, recurrent neural network (RNN) is particularly well-suited for processing time-series data due to its inherent ability to capture temporal dependencies in sequential data. Here, we implemented a vanilla RNN model and a Long Short-Term Memory (LSTM) model to do classification of EEG data. The latter one is expected to address the vanishing gradient problem in the vanilla RNN and capture long-term dependencies in sequential data.

Along with RNNs, multiple affine layers with corresponding batch normalization and dropout are added beforehand to help the network extract high-level features, improve regularization, and facilitate better training dynamics. After the affine layers, the data is flattened and reshaped before being substituted into the RNN structure. Overall, our RNN architecture can be represented by $[\text{affine}] \times \text{N} - [\text{flatten}] - [\text{affine}] - [\text{vanilla RNN}/\text{LSTM}]$ with dropout and batch normalization in between.

1.3. CNN + RNN (vanilla & LSTM & GRU)

Research has shown that the accuracy of EEG analysis may be enhanced by combining the strong modeling power of CNN in temporal feature extraction and the advantage of RNN in processing sequential information. Therefore, we implemented a hybrid CNN-RNN architecture. The architecture of the total network is $[\text{conv} - \text{pool}] \times \text{N} - [\text{flatten}] - [\text{affine}] - [\text{RNN}]$. To find out the best performance of such architecture, we tried vanilla RNN, LSTM, and Gated Recurrent Unit (GRU) with other components remaining the same. The architectures using LSTM and GRU are expected to have better performance than vanilla RNN because they can capture long-term dependencies in the EEG data, but which of them performs better usually needs to be determined experimentally.

1.4. CNN + LSTM: spatiotemporal 2D convolution

All convolutional operations mentioned in 1.1-1.3 were performed in 1D along the time variable, which meant the spatial information in the EEG data (i.e., the potential connection between electrodes) was neglected. Assuming the data obtained from different electrodes at the same time have underlying connection, a 2D convolution version of hybrid CNN-RNN architecture was implemented. Defining the number of electrodes as the width and the number of time bins as the height, the input data was reshaped into a 3D tensor with a channel depth of 1, as shown in Figure 1. In this case, the convolution with 2D filters was performed in both time and space, which is expected to extract temporal and spatial features simultaneously. Different sizes and shapes of the filter were designed to adjust the emphasis on either time or space variable.

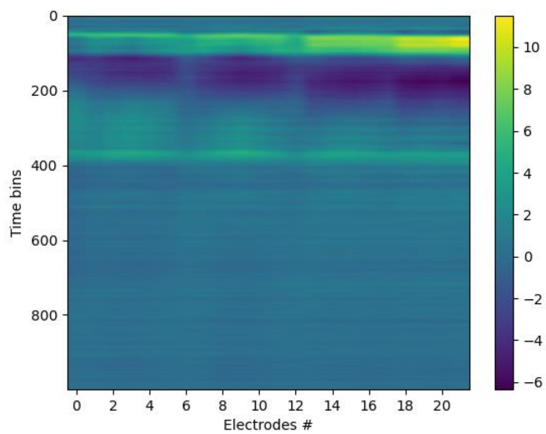

Figure 1. Input data format of spatiotemporal 2D convolution. The height represents time, the width represents electrodes, and the color represents the signal magnitude.

1.5. CNN + Unconventional architectures (transformer & ResNet)

Transformer, renowned for its attention mechanism, may enhance model's understanding of EEG patterns by focusing on relevant temporal information, while ResNet tackles vanishing gradients during the training which leads to better performance. By combining CNN with these unconventional architectures, we are expected to further enhance the performance our models.

2. Results

2.1. Model comparison

The classification accuracy of all the models is illustrated in Table 1. The hybrid CNN-LSTM achieves the highest accuracy of 0.76, followed by the shallow CNN and deep CNN, reaching the accuracy of 0.73 and 0.68, respectively. The hybrid CNN。-transformer model also reaches the accuracy over 0.7. The hybrid CNN-RNN models adopting the similar architectures (i.e., CNN+ Vanilla RNN/GRU) has a relatively large gap with the optical value. The models using the RNN architectures only, such as vanilla RNN and LSTM, have the lowest accuracy that is even below 0.5. The hybrid CNN-LSTM with 2D convolution has accuracy of 0.57, which is similar to that of other hybrid CNN-RNN models, but lower than its 1D convolution version.

2.2. Individual vs. collective training

A comparison of individual and collective training results is shown in Table 2. Both CNN and hybrid CNN-LSTM architectures were used for this analysis. By limiting the training data to only one subject (Subject 1), the testing accuracy for all subjects and that subject become far below the optimal value. This trend is observed for both models, but the accuracy drop seems to be more severe in the hybrid CNN-LSTM compared to the CNN, which might indicate that the hybrid CNN-LSTM model is more sensitive to the training data size.

2.3. Classification accuracy vs. time

We evaluated the classification accuracy with respect to time (i.e., the number of time bins in the training dataset) on our most accurate model (hybrid CNN-LSTM) and the second most accurate model (shallow CNN), to analyze the time that is needed to obtain a reasonable accuracy and a robust model. Since the original EEG data has around 1000-time bins, we conducted tests using 10 distinct time clips from 100 to 1000.

As shown in Figure 2, at the beginning with only 200 time bins, the model has a relatively low accuracy. With the increase of data being used for training, the accuracy is dramatically enhanced and exceeds 0.7. Although some drops occur at 600 and 700 time clips, the accuracy shows an overall upward trend with time. It is also worth noting that the accuracy can reach around 0.7 using only one-third amount of the EEG data, which means that our model is robust to the changing of data with respect to time.

A similar trend is observed in the shallow CNN model (Figure 3), where the accuracy exceeds 0.7 after the 500 time clips, and gradually reaches its plateau at 700.

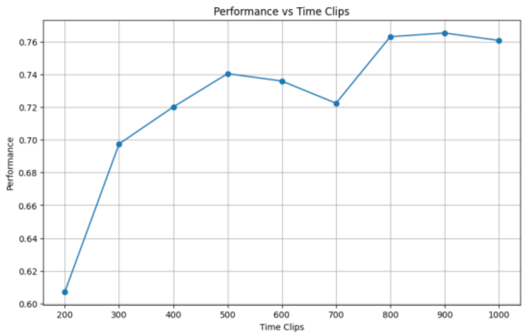

Figure 2. Accuracy vs. time using hybrid CNN-LSTM model

Figure 3. Accuracy vs. time using shallow CNN model

2.4. Dataset augmentation

Dataset augmentation can be considered as an efficient regularization technique. In the original EEG dataset, there are only 2115 trails, which may lead to a high risk of overfitting. Therefore, two dataset augmentation methods were developed and applied to the original dataset before

3. Discussion

3.1. Model analysis

3.2. Individual vs. collective training analysis

The decrease in accuracy can be attributed to the limited size of the training dataset (approximately 200 instances) and the constrained testing dataset (50 instances) associated with Subject 1. Consequently, the model's performance might exhibit a higher susceptibility to biases and variability due to the reduced sample size, leading to a lack of stability that is typically afforded by larger datasets. Furthermore, from the perspective of data acquisition, the training dataset for Subject 1 is inherently more tailored to the idiosyncratic characteristics of that individual. Generalizing this subject-specific model to encompass the broader spectrum of channel activities across different individuals may result in substantial errors. For instance, neural activity patterns captured across 22 channels in one individual, indicative of foot movement, might closely resemble patterns in another individual that correspond to tongue movement, thereby highlighting the potential for significant misclassification in a generalized model.

3.3. Classification accuracy vs. time analysis

References

Appendix

Architecture	**Accuracy**
Shallow CNN	**0.73**
Deep CNN	0.68
Vanilla RNN	0.56
LSTM	0.49
CNN + Vanilla RNN	0.52
CNN + LSTM	**0.76**
CNN + GRU	0.58
CNN + LSTM (2D convolution)	0.57
CNN + Transformer	**0.72**
CNN + ResNet	0.54

Table 1. A comparison of testing accuracy between models

Training	**Testing**	**Architecture**	**Accuracy**
Subject 1	Subject 1	CNN	0.56
Subject 1	All Data	CNN	0.43
All Data	All Data	CNN	**0.73**
Subject 1	All Data	CNN + LSTM	0.42
Subject 1	All Data	CNN + LSTM	0.33
All Data	All Data	CNN + LSTM	**0.76**

Table 2. Individual and collective training results based on CNN and hybrid CNN-LSTM architectures