EEG Classification: Attention, CNN, GRU, CNN+LSTM, and CNN+GRU

Yang-Shan Chen
UID: 505851728
cys0621@g.ucla.edu

Chih-En Lin
UID: 005627440
chihenl72@g.ucla.edu

Yu-Hsiang Liu
UID: 205626487
shawnliu@g.ucla.edu

Sanae Amani Geshnigani
UID: 705666305
samani@ucla.edu

Abstract

*In this project, we study EEG classification using 1) mixed CNN and Conformer (Attention); 2) CNN; 3) mixed CNN and LSTM; 4) GRU; and 5) mixed CNN and GRU architectures. We implemented these architectures using Keras and Pytorch packages. Our results show that the best testing accuracy for Attention is **72.06%**, for CNN is **67.21%**, for CNN+LSTM is **70.75%**, for GRU is **58.24%**, and for CNN+GRU is **71.90%**. Our codes can be found at https://drive.google.com/drive/folders/1SnKZL5lrDd24tHGffi6Sejf9i89VkJG8?usp=sharing.*

1. Introduction

EEG reflects the coordinated activity of millions of neurons near a non-invasive scalp electrode. Because these are scalp potentials, necessarily, they have relatively poor spatiotemporal resolution compared to other neural recording techniques. EEG is believed to be recording dipoles that are transmitted through the scalp [2].

This dataset consists of EEG data from 9 subjects. In the dataset, there are 2115 trials; each trial has corresponding EEG data from 22 electrodes over 1000 time bins. In particular, the shape of data we used are as follows:

- Training/Valid data: $(2115, 22, 1000)$
- Test data: $(443, 22, 1000)$
- Training/Valid target: $(2115,)$
- Test target: $(443,)$

The brain computer interfaces (BCIs) are made of four different motor imagery tasks, namely the imagination of movement of the left hand (class 1), right hand (class 2), both feet (class 3), and tongue (class 4) [1].

The goal of this project is to optimize the performance of decoding on the task of EEG multiclass classification using different neural network architectures.

2. Results

2.1. CNN+Conformer (Attention)

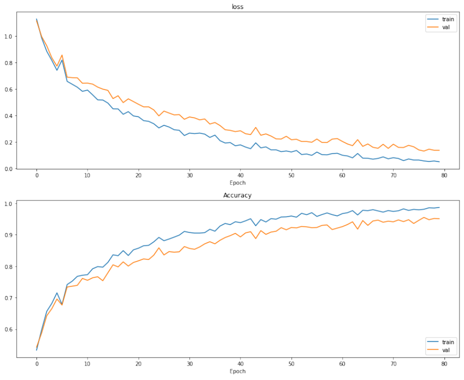

Figure 1: Results for the Attention model.

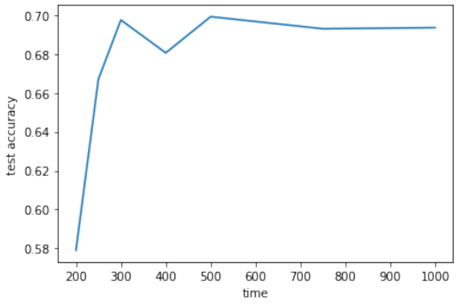

Figure 2: Testing accuracy of Attention as a function of number of time bins.

2.2. CNN

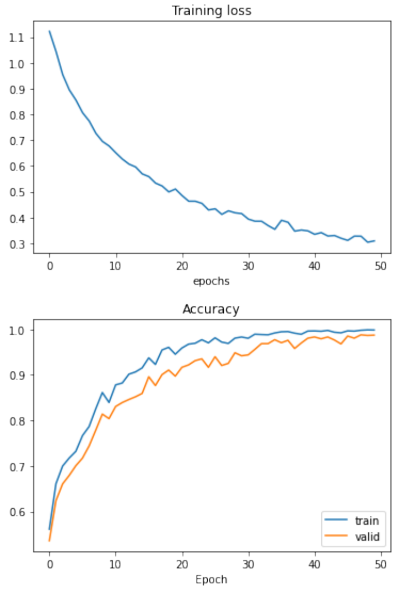

Figure 3: Results for the CNN model.

2.3. CNN+LSTM

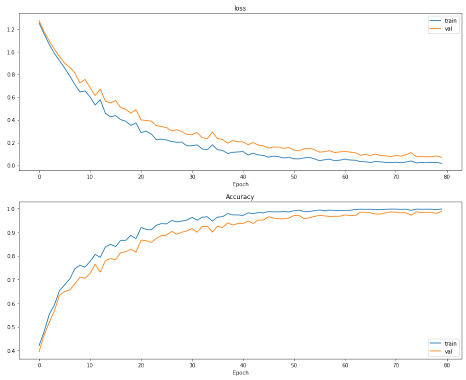

Figure 4: Results for the CNN+LSTM model.

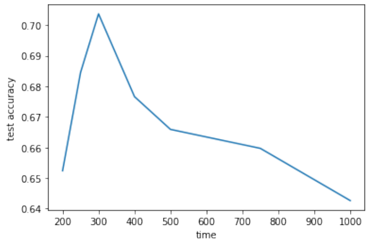

Figure 5: Testing accuracy of CNN+LSTM as a function of number of time bins.

2.4. GRU

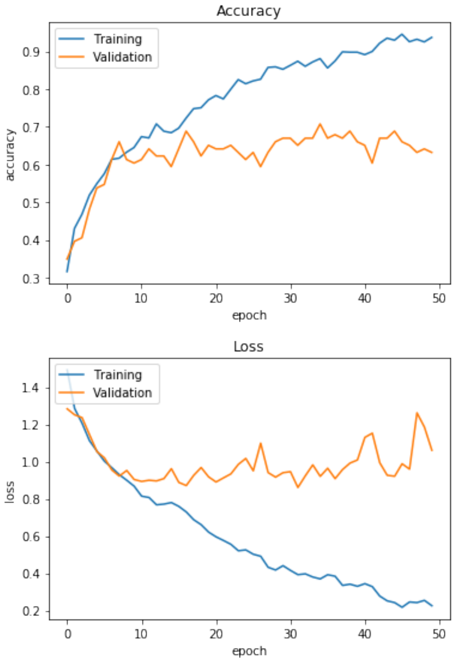

Figure 6: Results for the GRU model.

2.5. CNN+GRU

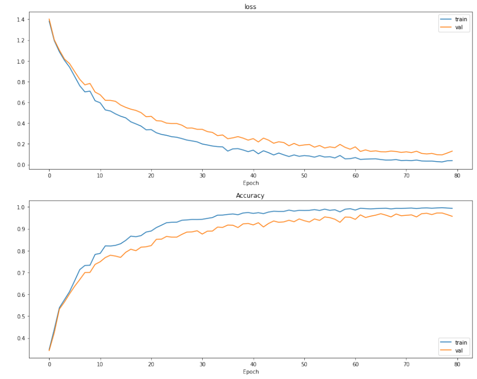

Figure 7: Results for the CNN+GRU model.

3. Discussion

In this project, we developed five neural network architectures: 1) mixed CNN and Conformer (Attention); 2) CNN; 3) mixed CNN and LSTM; 4) GRU; and 5) CNN+GRU. Test accuracy of the 1), 3), 5) models are higher than **70%**, 2) model are close to **70%**, and that of the forth model is about **60%**. In particular, mixed CNN and LSTM/GRU uses CNN layers that play the role of decoder and takes advantage of the strengths of both CNN and LSTM/GRU.

3.1. CNN+Conformer (Attention)

The model is constructed by 2 CNN layers and 1 Conformer block, which includes feed forward modules, attention module and convolution module. Also, we do the data preprocessing as TA did in discussion 9. That is to say, we only take data with the previous 500 time bins.

From Figure 1, due to preprocessing, the training data and validation data would be highly correlated, so it's unlikely to face overfitting. Therefore, the training loss and validation loss are close all the time.

3.2. CNN

In this section, we use 4 CNN layers and use the preprocessed data that TA mentioned in discussion 9. Because of the effect of preprocessing the training and validating data are high correlated, the validation accuracy is as high as training accuracy as shown in Figure 3. The best validation accuracy among the 50 epochs is 98.73%, and the accuracy on test data is **67.21%**.

3.3. CNN+LSTM

The model is constructed by 4 CNN layers and 1 LSTM with the data preprocessing. The training and validation datasets are highly correlated, and therefore, the validation accuracy is as high as training accuracy as shown in Figure 4

Our best test accuracy of CNN+LSTM model is **70.75%** when we choose 300 time bins.

3.4. GRU

In the preprocessing step, we downsampled the data through FFT in an attempt to remove high-frequency noises.

Figure 6 shows that the best validation accuracy of our GRU model with a downsampling size of 25 points and 50 number of epochs is **71.7%** when the model is trained over all subjects. Also, our accuracy on test data is **58.24%**.

3.5. CNN+GRU

The model is constructed by 4 CNN layers and 1 GRU with data preprocessing. The training and validation datasets are highly correlated, and therefore, the validation accuracy is as high as training accuracy as shown in Figure 7.

Our best test accuracy of CNN+GRU model is **71.90%** when we choose 300 time bins.

3.6. Comparison between networks

Compared to CNN, CNN+LSTM and CNN+GRU increases about 3%-4% in the test accuracy. It shows that LSTM and GRU indeed works in time series data.

Compared to only GRU, CNN+GRU increases about 12% in the test accuracy. It shows that mixed CNN and GRU takes advantage of the strengths of both CNN and GRU. Moreover, it takes more training time on both CNN+LSTM and CNN+GRU models due to the complexity of LSTM and GRU. Three layers in both LSTM and GRU let the model need more time to find the optimal status.

3.7. Comparison between one vs. all subjects

For CNN and GRU models, we compare the accuracy between the model trained over all subjects and the model only trained over subject 1. As a result, the accuracy of the CNN model only trained over subject 1 is 32.22%; while, it is 67.21% when trained over all the 9 subjects. Moreover, when we train the GRU model over subject 1 our validation accuracy and testing accuracy are 66.67% and 46%, respectively, which are worse than those of trained model over all the subjects, 71.7% and 58.24%.

One of the reasons is the different number of training data. The more training data, the more chance to avoid overfitting. Also, the diversity of different subjects is a beneficial as it makes the model more robust.

3.8. Compare between Different Time Bins

From Figure 2 of Attention model, we observe that the test accuracy decreases rapidly when time is less than 300, which means the data before that is important and we cannot discard this portion of data. However, after time 500, the accuracy is almost the same. We can say the data after that does not help to our model. Our best test accuracy of attention model is **72.06%** when we choose 500 time bins.

Figure 2 showcases that the first half of the data is more useful than the second half. We guess the reason for this result is when a person imagines performing an action in the brain, the first moment of the image is the most clear. And then a person may be distracted or lose focus on the action, leading to some noises in the second half.

Figure 5 is testing accuracy of CNN+LSTM model as a function of number of time bins. We can also see that the test accuracy decreases rapidly when time is less than 300. We conclude that our CNN+LSTM model will fit the data which is not important.

References

[1] C. Brunner, R. Leeb, G. Müller-Putz, A. Schlögl, and G. Pfurtscheller. Bci competition 2008–graz data set a. *Institute for Knowledge Discovery (Laboratory of Brain-Computer Interfaces), Graz University of Technology*, 16:1–6, 2008.

[2] E. Niedermeyer and F. L. da Silva. *Electroencephalography: basic principles, clinical applications, and related fields*. Lippincott Williams & Wilkins, 2005.

4. Summary of all Algorithm Performances

Our results show that the best testing accuracy for Attention is **72.06%**, for CNN is **67.21%**, for CNN+LSTM is **70.75%**, and for GRU is **58.24%**. For all these architectures, we trained over all subjects.

Model	Epoch	Time Bins	Test Accuracy
Attention	80	500	72.06%
CNN	50	500	67.21%
CNN+LSTM	80	300	70.75%
GRU	50	1000	58.24%
CNN+GRU	80	300	71.90%

The following table is the test accuracy of each subject with CNN model. The data is augmented by the method we mentioned previously.

No. of Subject	number of test data	Test Accuracy
0	200	60.50%
1	200	58.50%
2	200	80.50%
3	200	69.50%
4	188	81.91%
5	196	66.33%
6	200	73.50%
7	200	76.00%
8	188	80.32%

5. Architectures

Data preprocessing: We do it as TA did in discussion 9 in all the models except for the GRU model.

5.1. CNN+Conformer (Attention)

The first 2 layers are CNN, and each layer consists of one convolution, activation function of ELU, maxpool, batachnorm and dropout. To be specific, in the first layer, the input channel size of 1-D convolution is 22, and the output channel size is 44. In the second layer, the input channel size is 44, and the output channel size is 88. The size of both kernels are 10. The 1-D maxpool are the same through the whole model, the filter's size is 3, and stride is equal to 3. The probability of dropout is 60%. And then we pass through a conformer block with kernel size 10 and 1 head. The dropout is 30% in attention, feed forward and convolution modules. At the end, we give an avgpool layer with size 7 and a fully connected layer.

Loss Function: Cross-entropy; Optimizer: Adam

5.2. CNN

This model is composed of 4 CNN layers, each of which consists of one convolution, activation function of ELU, maxpool, batchnorm and dropout. To be specific, in the first layer, the input channel size of 1-D convolution is 22, and the output channel size is 25. In the second layer, the input channel size is 25, and the output channel size is 50. In the third layer, the input channel size is 50, and the output channel size is 100. In the forth layer, the input channel size is 100, and the output channel size is 200. The 1-D maxpool are the same through the whole model, the filter size is 3, and stride is equal to 3. The probability of dropout is 50%. At the end, we use a fully connected layer to make the output into four classes.

Loss Function: Cross-entropy; Optimizer: Adam

5.3. CNN+LSTM

This model is composed of the above CNN model, in addition to one LSTM. In LSTM, there are 3 layers, the input size is 200, and the units are 400. The probability of dropout is 50%, and we set batch_first equal to true. At the end, we use a fully connected layer to return an output which is one of the four classes.

Loss Function: Cross-entropy; Optimizer: Adam

5.4. GRU

- Three GRU recurrent layers, in which dropout is applied.

- One fully connected (FC) layer, in which dropout, batch normalization, and ReLU activation are applied.

- One output layer, in which cross-entropy loss function and softmax activation function are used.

Loss Function: Cross-entropy; Optimizer: Adam

5.5. CNN+GRU

This model is composed of the above CNN model, in addition to one GRU. In GRU, there are 3 layers, the input size is 200, and the units are 400. The probability of dropout is 50%, and we set batch_first equal to true.At the end, we use a fully connected layer to return an output which is one of the four classes.

Loss Function: Cross-entropy; Optimizer: Adam