Different Networks on EEG classification dataset

PinHan Chen

August 3, 2023

A Abstract

In this project, we implemented different machine learning models that help us classify the electroencephalography data. The data is provided by the Brain-Computer Interaction (BCI) Competition in 2008. The target value of it is to classify four different tasks (Cue onset left, Cue onset right, Cue onset foot, and Cue onset tongue). Here, we will implement **Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), Convolutional Recurrent Neural Network (CRNN), and Convolutional Variational Autoencoder (ConvVAE)** to analyze this data. We will discuss the performance between each model, the possible reason behinds the comparison, and also the effects of time versus the classification accuracy.

B Introduction

Here, we will introduce the neural network architecture we use, and their corresponding performance we obtained from the data.

B.1 Convolutional Neural Network (CNN)

Figure 1. is the architechture of the Convolutional Neural Network. I choose the similar architecture TA provides in the duscussion 9. It consists of four convolution layers and each one has:

- an activation layer using ELU (Exponential Linear Unit): Helping us improve learning performance and alleviate vanish gradient problem.

- a 2-dimensional max pooling layer: Helping to reduce spatial dimensionality and increase computational efficiency.

- a batch normalization layer: Improve the stability of the model.

- a dropout layer: Prevent overfitting and improve the generalization performance.

Finally, we build a fully-connected layer to calculate the one-hot vector of classification.

B.2 Recurrent Neural Network (RNN)

Here, we implement two Recurrent Neural Network models, LSTM (Long Short-Term Memory) and GRU (Gated Recurrent Unit), to measure the performance. We use the built-in keras package to determine the performance of RNN model on this dataset.

To compare these two models quickly, LSTM has a more complex architecture compared to GRU models. It has three gates: input, output, and forget gates to control its ability to learn informations. On the other hand, GRU only has two gates: update and reset gates. In which makes it more easier to train and performs better in certain tasks in terms of efficiency.

B.3 Convolutional Recurrent Neural Network (CRNN)

Combining the structure of CNN and RNN, we obtain the structure of Convolutional Recurrent Neural Network. Figure 2 is its structure (using LSTM).

The first four layers are same as our convolutional neural network. After the last dropout layer, we add the layers of RNN model into the structure. We also test two different RNN model (LSTM vs. GRU) here.

By combining these two, the CRNN model can capture both spatial and temporal information of the training data. It can have the pros of CNN and RNN model, but it also has a larger complexity which we should notice.

B.4 Convolutional Variational Autoencoder (ConvVAE)

Lastly, we want to see if we can generate our own artificial EEG data to increase dataset size. If so, it can help to improve the performance of previous classification models. Figure 3. is its structure. We built two module 'encoder' and 'decoder' as the discriminator and generator works in the model.

The 'encoder' has two convolution layers to catch the information of input data, then follows by a Flatten layer and Dense layer to generate the latent representation of input data.

In 'decoder', the size of layer should correspond to the layers in 'encoder', so that we can transform the latent representation back to the input data. It has Dense Layer, Reshape Layer, and 2 transpose convolution layers respectively.

C Results

Here, we first show the data that we need to answer the questions in discussion:

C.1 The classification accuracy for subject 1

After optimization, Table 1 is the accuracy table of classifiers that are only trained with "subject 1's" data.

Table 2 is the accuracy table of classifiers that trained with the whole data and test on subject 1's data.

C.2 The classification accuracy across all subjects

Table 3 is the accuracy table of classification accuracy across all subjects.

C.3 The possible relationship between classification accuracy and time bin

Figure 5 shows the relationship between classification accuracy and time bin. We select numbers range from 100 to 1000 per 200, and choose the cloest time bin (len: 500) to filter data.

C.4 The performance of ConvVAE

Figure 4 is the latent space projection graph of ConvVAE, we will discuss it in next section.

D Discussion

In this section, we will analyze the performance of different models, and other results we obtained.

D.1 classification on subject 1

From table 1 and 2, we can see that performance of complex models (CNN-LSTM, CNN-GRU) are better when it's both trained and tested on subject 1 data. CNN performs always the best among other models.

It might because the complex models is overfitting to the subject 1 data so that they can performs better in this scenario. On the ther hand, CNN performs best in both case since it's a suitable model for this dataset considering to its parameters with others.

D.2 Model Comparison across all subjects and one subject

Table 3 and 5 is the comparison between different models in all subjects and one subject respectively. As you can see, the performance of CNN is better than others. It might because that the complex models all get overfitting to the dataset. There are no

Subject	model	accuracy
1	CNN	0.63
1	LSTM	0.34
1	GRU	0.42
1	CNN-LSTM	0.58
1	CNN-GRU	0.56

Table 1: The classification accuracy [subject 1 vs. subject 1]

Subject	model	accuracy
1	CNN	0.66
1	LSTM	0.36
1	GRU	0.30
1	CNN-LSTM	0.28
1	CNN-GRU	0.57

Table 2: The classification accuracy [all vs. subject 1]

Subject	model	accuracy
all	CNN	0.73
all	LSTM	0.38
all	GRU	0.36
all	CNN-LSTM	0.68
all	CNN-GRU	0.66

Table 3: The classification accuracy across all subjects

Subject	model	accuracy
1	CNN	0.63
2	CNN-GRU	0.55
3	CNN	0.69
4	CNN	0.67
5	CNN	0.73
6	CNN-GRU	0.58
7	CNN-GRU	0.67
8	CNN	0.64
9	CNN-LSTM	**0.79**

Table 4: The best classification accuracy for all single subject

Subject	**CNN**	LSTM	GRU	CNN-LSTM	CNN-GRU
1	0.63	0.34	0.42	0.58	0.56
2	0.46	0.27	0.41	0.51	0.55
3	0.69	0.30	0.29	0.53	0.50
4	0.67	0.39	0.31	0.43	0.66
5	0.73	0.34	0.26	0.62	0.70
6	0.57	0.37	0.35	0.54	0.58
7	0.65	0.43	0.38	0.63	0.67
8	0.64	0.34	0.21	0.61	0.57
9	0.73	0.28	0.26	0.79	0.74
Average	**0.64**	0.34	0.32	0.58	0.56

Table 5: The performance of each model

sufficient data for these complex models to learn so that a simple model (CNN) performs better here.

On the other hand, you can tell that the pure RNN models here all have worse performance compares to others. It might suggest that this dataset is not suitable for RNN. RNN models need sufficient data to generalize its performance. This dataset is too small so that CNN performs better here.

D.3 classification accuracy vs. time bin

From figure 4, it shows that the classification accuracy doesn't improve as we have data over longer periods of time. From our result, we can get the best accuracy when time = 300. Therefore, a time duration of 300 unit should be enough for us get a reasonable classfication accuracy.

D.4 Failure of ConvVAE

From figure 4, it's the latent space projection graph of ConvVAE. As you can see, we can not tell a significant clustering or separation between the data. It's not a good signal when we are determine the performance of a VAE model using its latent projection graph. To solve this problem, we might need to collect more relevant data to train our generative model if we can. The size of current dataset is too small for VAE so that it's overfitting and can not generate an artificial EEG data for now.

E Appendix

References

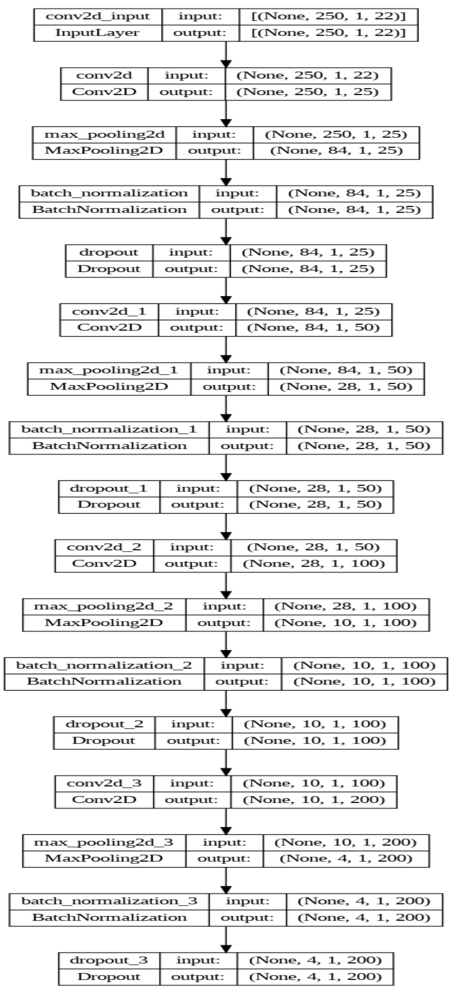

Figure 1: The structure of CNN.cnn_layers

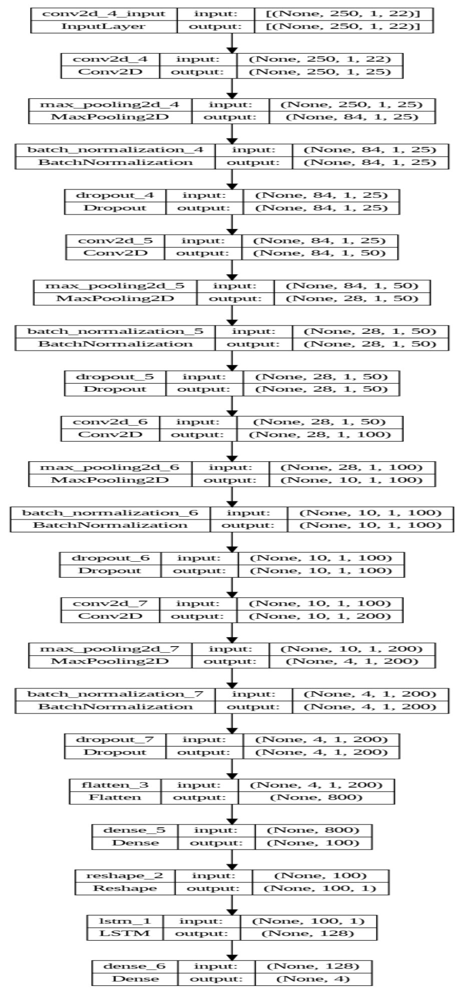

Figure 2: The structure of CRNN (using LSTM)

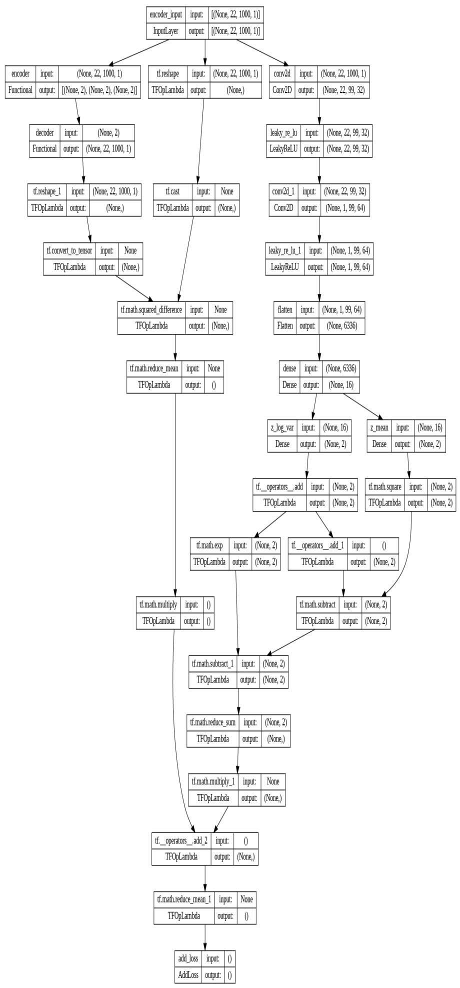

Figure 3: The structure of ConvVAE

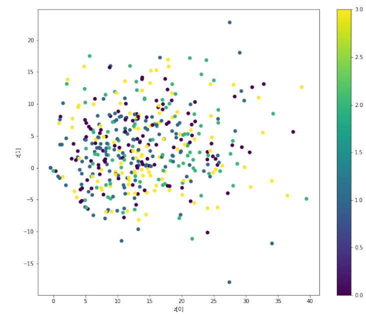

Figure 4: ConvVAE latent space projection graph

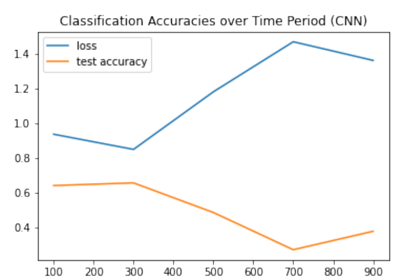

Figure 5: Classification accuracy vs. time bin