ECE C147/C247 - Winter 2022 - EEG Classificiation Project

Ashish Kumar Singh
UID: 105479019

Madhav Sankar
UID: 405692669

Suyash Prakash
UID: 905728171

Vivek Arora
UID: 505526269

Abstract

The project aims to perform multi-class classification of electroencephalography (EEG) data provided by the BCI Competition. EEG reflects the coordinated activity of neurons near a non-invasive scalp electrode. The four classes are for different motor imaginary tasks - movement of left hand, right hand, feet and tongue. We compare performance of multiple deep learning architectures, including CNNs, CRNN and Autoencoder for multiple experiments in our report including training on single subject vs all subjects and varying the time duration.

1. Introduction

First of all, we visualize the data and based on the insights perform data preprocessing. Then we train multiple neural network architectures and compare their accuracies. Finally we discuss the results obtained. Details of these steps are provided in following subsections.

1.1. Preprocessing

We are provided with EEG data of 2115 trials from 8 different subjects. Each trial has data from 22 electrodes for 1000 time intervals. We visualize the dataset and figure out that the first 500 time intervals are more meaningful. After trimming, we perform maxpooling, average pooling with gaussian noise and concatenate them over the samples axis. We also perform subsampling with gaussian noise as a data augmentation technique.

1.2. Architectures

We trained 3 different types of architectures - Convolutional Neural Networks, Convolutional Recurrent Neural Networks and Convolutional Recurrent Neural Networks with Autoencoders on this dataset. Fig. 1 shows the diagram of architectures. Details of each of them are as follows:

1.2.1 Convolutional Neural Networks (CNN)

The architecture of our convolutional neural network (CNN) is shown in Figure 1. It consists of 4 convolutional blocks followed by a fully connected layer with softmax activation for getting class probabilities. Each convolutional block consists of a Convolutional layer, followed by Max Pooling. We use Batch Normalization in each convolutional block and use dropout for regularization. We used Adam optimizer for training of this network.

1.2.2 Convolutional Recurrent Neural Networks (CRNN)

The architecture of our convolutional recurrent neural network (CRNN) is similar to our CNN architecture and is shown in Figure 1. It consists of the same 4 convolutional blocks. It is then followed by 4 GRU layers. We try out with vanilla RNN and LSTM as well but obtain the best classification accuracy with GRUs. It is then followed by a fully connected layer with softmax activation for getting class probabilities as before. We use the same hyperparameters in the convoltional blocks as in out CNN architecture. We used Adam optimizer for training of this network.

1.2.3 Convolutional Recurrent Neural Networks with Autoencoder (CRNNA)

We now add an Autoencoder layer to our base CRNN architecture as shown in Figure 1. The autoencoder part is added before the last fully connected layer where the encoding dimension in our model is 20 units which is equivalent to a compression factor of 10. We also try out sparse autoencoder by providing a sparsity regularizer value in the encoding layer. We used Adam optimizer for training of this network.

2. Results

We perform the three different experiments during training of our model. We compare the accuracies of all the three models on these experiments. The experiments are classification for single subject, classification for across all subjects an classification accuracy as function of time.

2.1. Classification for single subject

We first trained on complete dataset and tested on subject 1. Results are shown in Table 1. Then we used dataset of subject 1 to train the model and tested on. Results are shown in Table 2. The aim of this experiment is to check whether training on a single subject is better than training on all subjects if we are considering test on that particular subject only. Atleast for Subject 1, we can see from the results that training on all subjects give better accuracy for all the models. The reason could be that training only on one subject would have small dataset that might lead to overfitting. Also smaller dataset could be noisy and harder to train on.

2.2. Classification across all subjects

For this experiment, we train our models on all subjects and tested across all subjects. Results are shown in Table 3. We also check the performance of our CRNN model for different test subjects and shown the result in Table 4. The aim of this experiment is to check how the classification accuracy varies for different subjects. We can see that few of the subjects give better accuracy compared to others.

2.3. Classification accuracy as a function of time

We now use our best performing model - CRNN and see classification accuracy as a function of time steps. Results are shown in table 5. We see that the test accuracy increases till time steps 700 then decreases, which shows that most of the relevant information for classification is present in first 700 time-stamps.

3. Discussion

We perform the three different experiments during training of our model. We will discuss the results and their possible reasons in following subsections:

3.1. Performance of different architectures

We evaluate our model performance on Convolutional Neural Networks, Convolutional Recurrent Neural Networks and Convolutional Neural Networks with Autoencoder. We observe best performance on CRNN model which was able to perform around 1.5% better than CNN and CRNNA models. The reason for this could be that RNN layers are better at capturing information in time-slice data. In CRNN, we use multiple CNN layers before the RNN layers, which will increase the receptive field but still there would be time dependent information present for RNN layers. But in case of CRNNA, the use of two dense layers before RNN would decrease the time dependent information given to RNN layers, thus leading to less accuracy.

3.2. Training on one subject vs all

We first train and test our model on Subject 1 data only. We then train our models on all subjects and test on subject 1. We observe that the models trained on all subjects performs better. We conclude that the batch effects introduced by different subject is minimal and training on all subject helps during testing (atleast for subject 1).

We now train the model on all subjects and find test accuracies for all the subjects. We observe that accuracies vary a lot on subjects. Subject 3, Subject 5 and Subject 8, Subject 9 have higher accuracy range than Subject 1, Subject 2, Subject 4, Subject 6 and Subject 7. One reason for this could be that different level of noise are present in training data for different subjects, which might lead the model to give better results on subject who have data with less noise.

3.3. Classification as a function of time

We use our best performing model - CRNN and observe classification accuracy as a function of time steps. We start from 100 and go till 1000 in steps of 100. We observe that most of the information in the data is contained in the first half and thus we're getting good accuracy with only the first 500 - 700 time steps. Thus we conclude with this that the second half of the time steps are not very relevant to this classification problem. We confirm our observations by plotting the EEG signals and observe that the second half is almost the same for all 4 tasks.

3.4. Choice of hyperparameters

We optimize our models on a lot of hyperparameters. We try out 3-5 convolutional blocks, 1-4 RNN layers where we try out all RNN, LSTM and GRU units, number of hidden dimensions in RNN units. We also modify the convolution filter size to 15 to test our models.
We also tried multiple values dropout probabilities and observe that low dropout probabilities lead to overfitting.

References

[1] BCI Competition IV. BCI Competition IV, www.bbci.de/competition/iv/.

[2] Brunner, C, et al. BCI Competition - 2008 Graz Data Set A.

4. Model performance report

All results on Table 1, Table 2, Table 3, Table 4 are for hyperparameter t = 500

Model	Test Accuracy
CNN	62.0%
CRNN	61.5%
CRNNA	62.9%

Table 1. Train on all subjects, test on subject 1

Model	Test Accuracy
CNN	56.0%
CRNN	58.5%
CRNNA	61.8%

Table 2. Train on subject 1, test on subject 1

Model	Test Accuracy
CNN	70.1%
CRNN	71.6%*
CRNNA	70.0%

Table 3. Train on all subjects, test on all subjects

*After hyperparameter tuning with respect to time steps, we observed the best test accuracy to be **74.3%** for **t = 700**.

Model	Test Accuracy
Subject 1	58.5%
Subject 2	59.5%
Subject 3	78.5%
Subject 4	62.5%
Subject 5	85.6%
Subject 6	71.2%
Subject 7	73.0%
Subject 8	79.5%
Subject 9	78.7%

Table 4. Classification across all subjects

Time Steps	Test Accuracy
100	54.7%
200	60.9%
300	71.4%
400	69.4%
500	71.0%
600	72.5%
700	74.3%
800	72.5%
900	70.8%
1000	70.3%

Table 5. Classification accuracy as a function of time

5. Model architecture diagrams

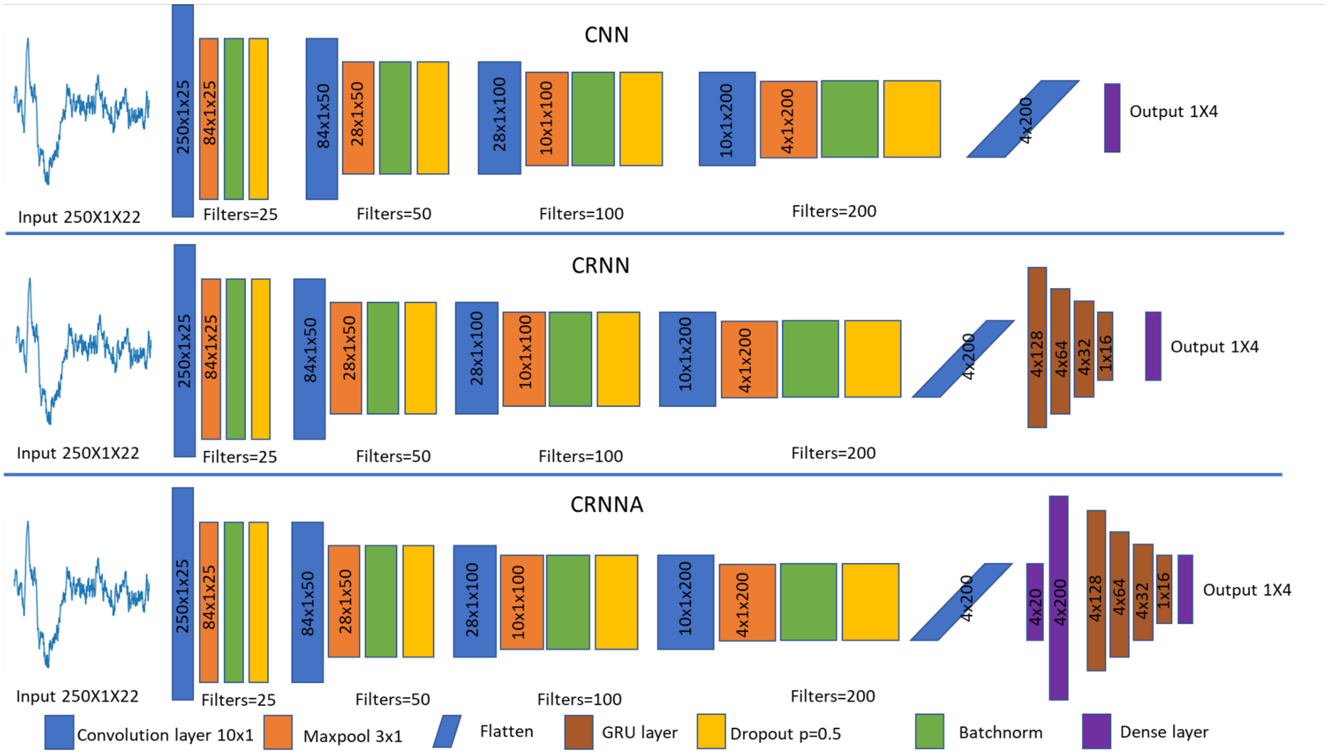

Figure 1. Architectures diagram