CNN, RNN (LSTM) architectures for EEG data

Shijin Koshy John
UCLA Samueli School Of Engineering
johnskj@g.ucla.edu

Abstract

This project involved the study of different neural network architectures for classification problems using the EEG data (https://www.bbci.de/competition)[1]. The focus of the project was on CNN and CNN+LSTM architectures for classifying the subject data and its performance as a function of time. Data science concepts related to data preprocessing, data augmentation, Batch normalization, dropouts, and regularization techniques were explored to improve the accuracy of the classifications and to identify the best model. During the tests, the CNN architecture predicted with a test accuracy above 70% for all models and above 60% for each subjects in the best model.

1. Introduction

For this classification problem, both convolutional neural networks (CNN) and long short-term memory (LSTM) a particular recurrent neural network (RNN) architecture with CNN combination was studied for the accuracy of classifying the EEG data into 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) for every subject. In addition to the classification problem, the models were also evaluated for accuracy as a function of time.

1.1. CNN

Convolutional Neural network (CNN) architecture was explored to extract the spatial features [2] related to EEG data. The CNN architecture is detailed in Model_1. There are a total of 4 layers where the first 3 are convolution blocks which includes a 2-dimensional convolution layer with L2 regularization followed by a batch normalization layer, an ELU activation layer, a max-pooling layer, and a dropout layer. The output of final convolution block is fed to the final fully connected layer that includes a flattening layer, a dropout layer, and a dense layer with L2 regularization and SoftMax activation. The input is a 4-dimensional tensor with batch-size, time-bins, and 22 EEG channels in the last dimension.

1.2. CRNN (CNN + LSTM)

Long short-term memory (LSTM) with CNN combination was explored to extract both the spatial and temporal features [2] related to EEG data. The CNN+LSTM architecture is detailed in Model_2. There are a total of 5 layers where the first 3 are convolution blocks which includes a 2-dimensional convolution layer with L2 regularization followed by a batch normalization layer, an ELU activation layer, a max-pooling layer, and a dropout layer. The output of final convolution block is fed to the first fully connected LSTM layer that includes a flattening layer, a dense layer with L2 regularization and Relu activation, a reshape layer, a LSTM layer, and a dropout layer. Finally there is a fully connected layer with a flattening layer, a dropout layer, and a dense layer with L2 regularization and SoftMax activation. The input is a 4-dimensional tensor with batch-size, time-bins, and 22 EEG channels in the last dimension.

2. Results

Project studied the optimization classification accuracy for subject-1 model, all-subject model and the test accuracies of each model on all subjects. Also the accuracies were evaluated as a function of time for each model. This section details the results of the experiments.

2.1. Classification accuracy of Subject-1 model

Table_1 displays the model, architecture (*CNN, CNN+LSTM*) and its parameters along with test scores and run time. The test accuracies for each subject using this model is detailed in Table_2.

2.2. Classification accuracy of All-Subject model

Table_1 displays the model, architecture (*CNN, CNN+LSTM*) and its parameters along with test scores and run time. The test accuracies for each subject using this model is detailed in Table_2.

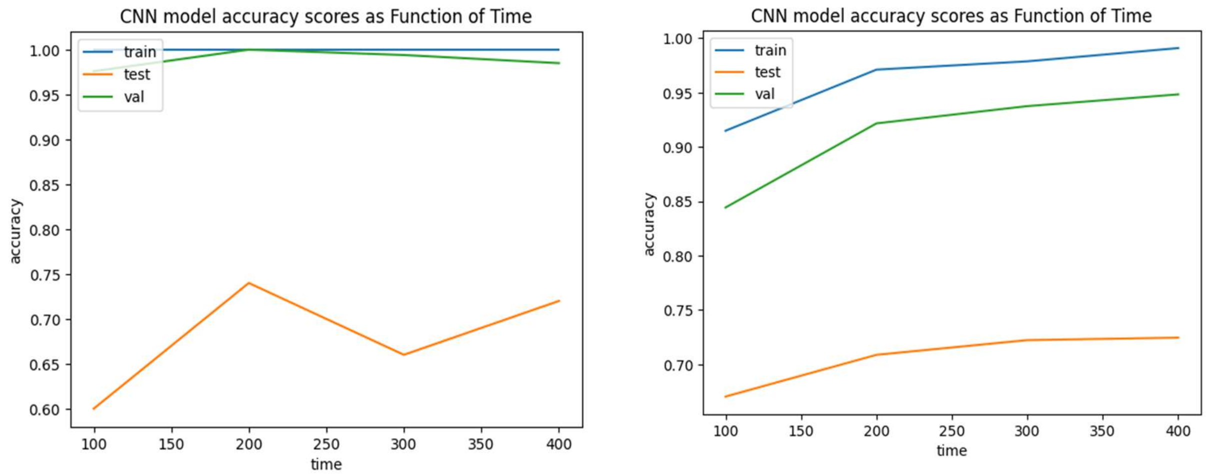


Figure 1: Classification accuracy as a function of time for CNN (Subject_1 model and All-Subject model)

2.3. Classification accuracy as a function of Time

Figure 1 shows the classification accuracy as a function of time for the CNN model and Figure 2 shows the classification accuracy as a function of time for the CNN+LSTM model. The first plot is for Subject_1 model and the second plot is for all-Subject model.

3. Discussion

This section discusses the insights gained from this project related to the architectures used for model, and the model performance based on parameters, pre-processing, and function of time.

3.1. Architecture with highest accuracy

This project tested with CNN and CNN+LSTM architectures. On comparing the resulting accuracies for each model, the CNN model classified the EEG data with highest test accuracy of above 70% for each individual models and above 60% test accuracy for each subjects. As CNN explores the spatial features, the resulting high accuracies mean that these features are more prominent in given EEG data and results in better classification.

3.2. Accuracies of Subject_1 vs All-Subject model

On evaluating the performance of Subject_1 model and all-subject model for every subject, the results in Table_2 confirm that all-subject model performs the best in accuracy terms with more than 60% classification accuracy for CNN model and above 45% accuracy for CNN+LSTM model for every subject from 1-9. The disadvantage in the all-subject model will be the training time involved which is over 850% increase in training time than a Subject_1 model but helps in better generalization across subjects.

Ultimately the choice of model depends on the task at hand.

3.3. Pre-processing and Data augmentation

In this project, various preprocessing, and data augmentation techniques like trimming, max-pooling, averaging, adding noise, and concatenation to increase sample size was implemented and these steps resulted in improving the test accuracy scores. So preprocessing, and data augmentation techniques are helpful in smoothing out and increasing the sample size of EEG signals which helps with improving the test accuracy scores for every model.

But when these techniques where applied to Fourier transformed EEG signals, the resulting test accuracy for Subject_1 model was 66% and all-subject model was 40% in CNN architecture. So, over-applying these steps can have the opposite effect and result in model performance being worse.

3.4. Factors affecting the model performance

In this project, the choice of hyperparameters like learning rate, weight decay, epochs and batch size had direct impact on the accuracies. The best model had a learning rate of .001, batch size of 256, and 100 epochs. The weight decay was set in the range of 0 to 0.05 for best performing model. Additional details on hyperparameter settings for each model are available in Table_1.

Additionally, the system installed program versions and the package versions also had a direct impact on the performance of these models and resulting test accuracies.

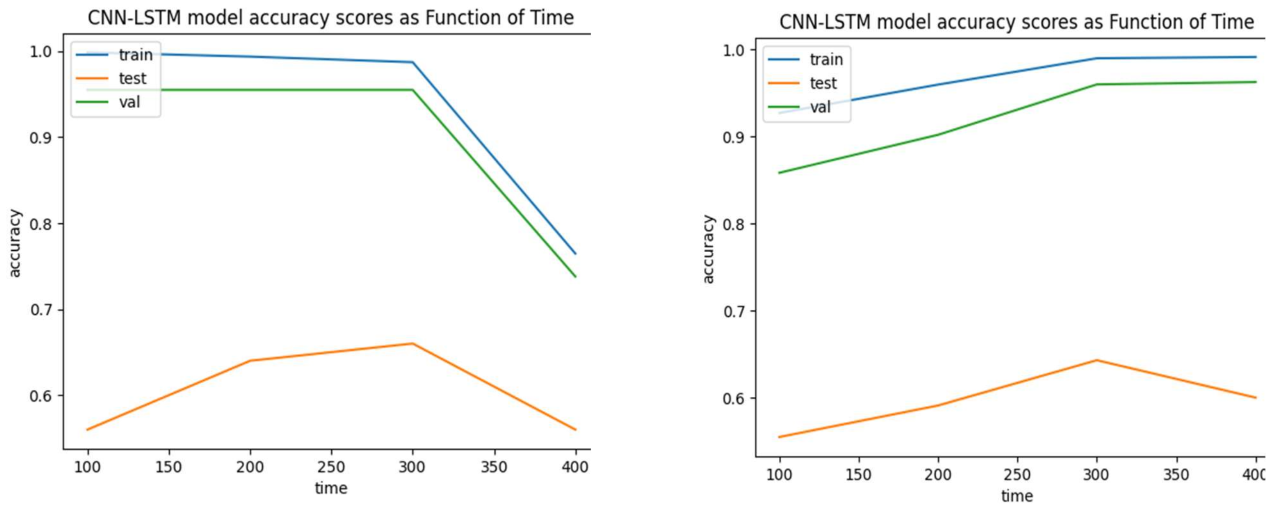


Figure 2: Classification accuracy as a function of time for CNN-LSTM (Subject_1 model and All-Subject model)

3.5. Accuracies as a function of time

Figure 1 details the classification accuracy as a function of time for CNN model and Figure 2 details the classification accuracy as a function of time for CNN-LSTM model.

For CNN, the test accuracy for all-subject model increases as a function of time and results in better classification as time bins increase, but for subject_1 model lower time bins result in higher accuracies and fluctuations are noticed as the time bins increase. So, for all-subject CNN model, the test accuracy can increase for higher values of time-bins but the same is not true for Subject_1 model.

For CNN-LSTM model, the test accuracy increases till time bin 300, but as the value increases, the test accuracy drops. So increasing the time-bins after a particular value does not improve the test accuracy.

References

[1] BCI Competitions - https://www.bbci.de/competition
[2] Motor imagery EEG classification algorithm based on CNN-LSTM feature fusion network - https://www.sciencedirect.com/science/article/abs/pii/S1746809421009393?via%3Dihub
[3] https://arxiv.org/pdf/1703.05051.pdf
[4] https://keras.io/api/
[5] UCLA EC ENGR-C247 Lectures

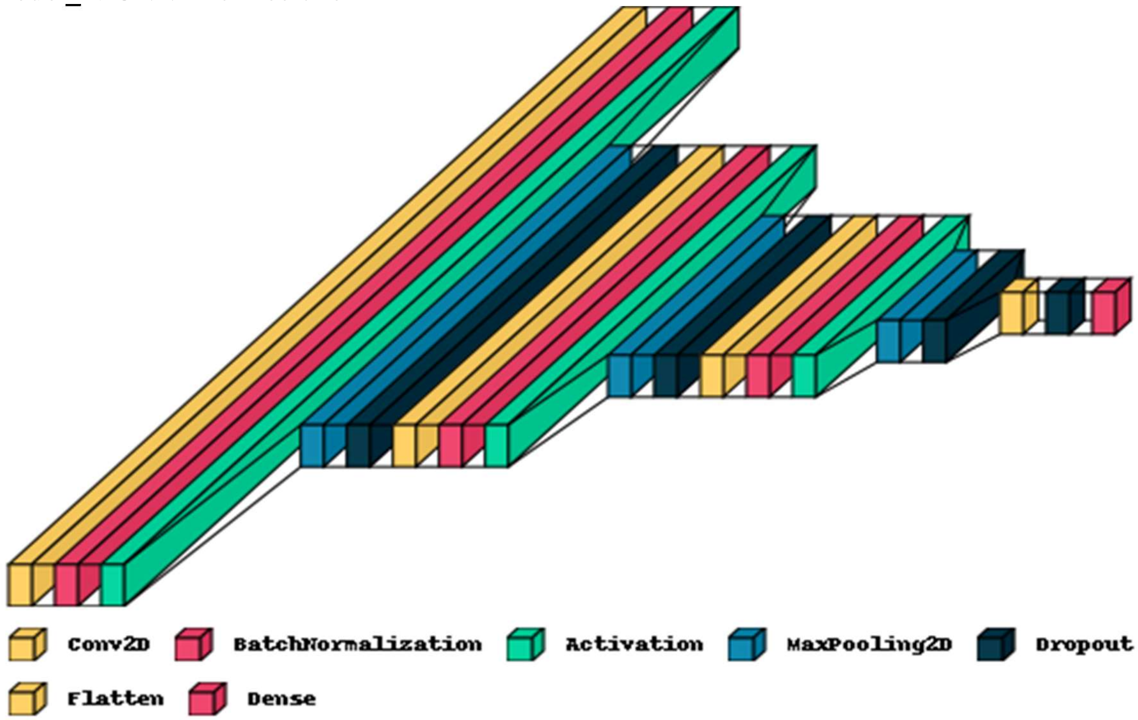
Table_1: Test scores and run times

Model	Data Augmentation	Parameters	Model details	Run Time (sec approx.)	Scores
CNN Subject_1 model	Trimming, Maxpooling, Averaging, Adding noise, concatenating to increase sample size.	Optimizer: Adam learning_rate = 1e-3 weight_decay = 0.00 epochs = 100 batch_size = 256	3 CNN layers with L2 Regularizers, batch_normalization, ELU activation, and dropout. 1 FC layer with softmax activation and dropout.	230s	72% - 74%
CNN All Subject model	Trimming, Maxpooling, Averaging, Adding noise, concatenating to increase sample size.	Optimizer: Adam learning_rate = 1e-3 weight_decay = 0.00 epochs = 100 batch_size = 256	3 CNN layers with L2 Regularizers, batch_normalization, ELU activation, and dropout. 1 FC layer with softmax activation and dropout.	1990s	71% - 72%
CNN-LSTM Subject_1 model	Trimming, Maxpooling, Averaging, Adding noise, concatenating to increase sample size.	Optimizer: Adam learning_rate = 1e-3 weight_decay = 0.00 epochs = 200 batch_size = 256	3 CNN layers with L2 Regularizers, batch_normalization, ELU activation, and dropout layers. 1 FC + LSTM layer with L2 Regularizers, RELU activation, reshape and dropout. 1 FC layer with softmax activation and L2 Regularizers.	310s	70.00%
CNN-LSTM All Subject model	Trimming, Maxpooling, Averaging, Adding noise, concatenating to increase sample size.	Optimizer: Adam learning_rate = 1e-3 weight_decay = 0.00 epochs = 200 batch_size = 256	3 CNN layers with L2 Regularizers, batch_normalization, ELU activation, and dropout layers. 1 FC + LSTM layer with L2 Regularizers, RELU activation, reshape and dropout. 1 FC layer with softmax activation and L2 Regularizers.	2230s	64.78%

Table_2: Subject scores

	Scores			
Subject	CNN Subject_1 model	CNN All Subject model	CNN-LSTM Subject_1 model	CNN-LSTM All Subject model
1	*72% - 74%*	*66% - 72%*	*70%*	*58%*
2	*36% - 38%*	*60% - 62%*	*26%*	*54%*
3	*60% - 62%*	*64% - 80%*	*40%*	*72%*
4	*40%*	*68% - 80%*	*30%*	*66%*
5	*27% - 29%*	*76% - 78%*	*25%*	*76%*
6	*32% - 36%*	*65% - 71%*	*20%*	*45%*
7	*36%*	*76% - 78%*	*34%*	*74%*
8	*38%*	*76%*	*30%*	*64%*
9	*36% - 38%*	*72% - 76%*	*32%*	*74%*

Model_1: CNN Architecture



Model_2: CNN-LSTM Architecture