EEG 4-class Motor Classification with Deep Learning Architectures

Diplav
UID: 605627748
UCLA
diplav@g.ucla.edu

Gaurav Singh
UID: 305353434
UCLA
grvsingh@g.ucla.edu

Abstract

Using Electroencephalogram (EEG) data to classify intended human activity is an important task for building Human Computer interfaces. This project aims to evaluate different Deep Learning architectures - CNN, CRNN (LSTM and GRU units), UNet - on the Brain Computer Interface (BCI) dataset [1] to classify an EEG signal corresponding to 4 human activity imaginations (i.e. left hand, right hand, both feet and tongue motion). A comparison of different time series data preprocessing techniques - Subsampling, Pooling, Windowing, Frequency Analysis - is reported on all and per subject basis.

1. Introduction

Motor imagery classification is a key topic in brain-computer interface (BCI) research as it facilitates the recognition of a subject`s intent of performing different tasks. Electroencephalography (EEG) is used to measure the brain dynamics of motor imagery as a nonstationary time series. In this project we have explored different methods of preprocessing the EEG data and applied various deep learning techniques for motor imagery classification.

1.1. Preprocessing Techniques

Trimming: We see that by taking mean across different users for a given activity EEG signal showed max. Variance in the first half of the time series signal as shown in figure 2. There we have considered trimming the first half of the signal (500 samples) data as one of the preprocessing techniques.

Windowing: This approach divides the entire signal into windows of fixed length and then concatenates them in vertical orientation as shown in figure 1. The idea behind this is to increase the receptive field and to easily get correlation between distant samples of signal.

Wavelet transformation: Wavelet transformation is used to provide us with the frequency of the signals and the time associated to those frequency. Thus converts the time series signal into a 2d representation. Morlet window is used for this transformation.

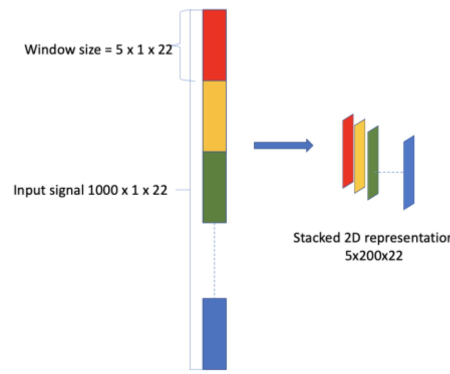

Figure 1: Windowing Preprocessing

1.2. Data augmentation strategies

EEG data is generally quite noisy and given the BCI dataset [1] is quite small having only 2115 data points, data was augmented using following strategy:
A single time series EEG signal is decomposed into 3 different kinds of representations. First is obtained by taking a max pool of window size 2. Second set of representation is obtained by taking the average pool of window size 2 and lastly by using random sub sampling and adding zero mean gaussian noise to it. By the above strategy we get three times the original training size and noise addition has the regularization effect on the training

1.3. Methods

We explored a total of 4 architecture.

CNN using 1D conv: We started with architecture shown in figure 6 given that the data in each channel is a 1D sampled signal. 1D convolution generates features captured along 22 channels for a given unit sample. It consists of 4 convolution blocks. Each convolution block consists of a 1D convolution filter of size 10 followed by elu activation function and batch normalization. A dropout of 0.5 is used for all 4 convolution blocks. The output is passed to a dense layer to get a one-hot vector for classification. The reason for using CNN is that it can extract differential patterns in activity windows regardless of precise location and enables scale invariance while reducing the number of parameters via weight sharing, making the architecture suitable as a high-dimensional feature extractor and classifier for motor activity recognition. ELU was used as the activation unit over ReLU and PReLU as it was performing experimentally better. One of the possible reasons for this is that it becomes smooth slowly until its output equals to negative α. Dropout is used to avoid overfitting given the capacity of the network was quite huge and training data is quite limited. Three convolution layers are used to ensure that the network has sufficient capacity to extract the relevant features from the noisy EEG dataset.

CNN using 2D conv: The architecture is shown in figure . It consists of 4 convolution blocks. A single convolution block consists of (3X7) 2D convolution filters followed by ELU activation function, batch normalization. A dropout of 0.5 is used for all 4 convolution blocks. The output is passed to a dense layer to get a one-hot vector for classification. Data is preprocessed using windowing method as shown in figure 1. for this architecture to work.

CRNN with LSTM unit: The output of the above architecture (CNN with 2D) is passed through 3 different LSTM layers. The dropout for each LSTM unit is kept as 0.6. The output is then passed to a dense layer to get a one-hot vector for classification. Although LSTM ideally should have worked better for the given time series classification given, they keep track of data over time, this is not observed with EEG data one of the reasons is that the EEG signals are noisy and the data per channel alone is not enough expressive of the underlying features, this is supported by the experimental result that a simple LSTM network gave a meagre ~32% test accuracy. CNN helps in extracting detailed underlying features in the signals and thus a combination of RNN fed with CNN features worked better giving 64% accuracy compared to pure LSTM networks.

CRNN with GRU unit: The LSTM units in above (CRNN with LSTM unit) architecture is replaced with GRU units.

UNet: We wanted to improve the receptive field of the network, with a fewer number of learnable parameters. As per current norms, UNet is best for this purpose. Therefore, we tried exploring the UNet using 1D convolution filters.

1.4. Results

From table 1 shows the classification performance on validation and test set and the type of preprocessing used in the same. Validation set was taken to be 0.20 % of the test set. 2D CNN with widowing pre- processing on the dataset achieves the highest test accuracy of 72.01 % for Motor imagery classification task.

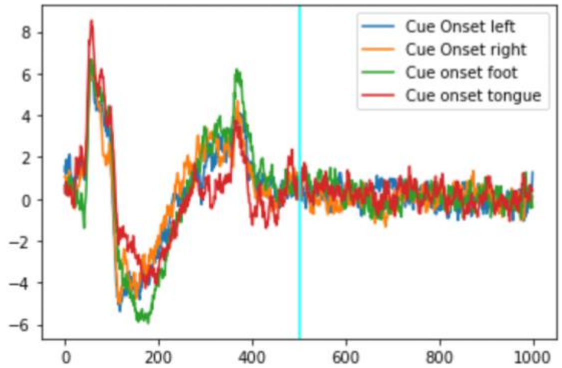

Figure 2: Avg. signal per activity for one channel

1.5. Discussion

Effect of trimming: Through our experiments on applying trimming on the training dataset the test accuracy of 1D CNN is increased by 3% from 65% to 68%. This is in congruence to the analysis on dataset which can be seen in figure 2 that the data beyond 500 samples is mostly noise. One possible reason we can think of is that humans have a reaction time of ~250ms [6] thus most of the information is found in first part of the signal.

Effect of Windowing: Trimming generally works better for architectures containing 1D convolution filters but in case when architecture has 2D convolutions it is better to have an input layer with high 2D dimensions. Thus, we used windowing as shown in figure 1. with window size of 5 found experimentally to convert 1D EEG signal to a 2D representation. On applying windowing method and using 2D CNN instead of 1D CNN increased our model test accuracy from 68.22 % to 72.01%. The best window size was found to be 5. Windowing signifies the impact of increase in receptive field thus validating our intuition.

Effect of Wavelet: The idea behind this preprocessing method was to convert 1D signals to an image like input and to use it with 2D CNN. The computational complexity for this method is very memory intensive and thus, due to limited computation resources we restricted the number of channels to 5 which are evenly chosen from the 22-channel range. The evaluation was done on only one subject with 5-fold cross validation given per subject data is very low. This method is not able to give good results compared to rest of the preprocessing methods as the test accuracy is ~0.32. Given more computation resources we would like to the same experiment with higher number of channels.

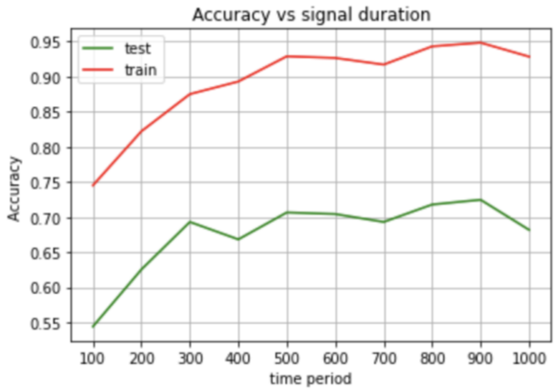

Figure 3: Train Test Accuracy as function of signal length. Range is from 100 to 1000 samples with time step 100

Effect of signal length: We considered signal length of 100 – 1000 with gaps of 100 to find the optimal time duration required for training our best architecture (2D CNN with windowing). The results can be seen in figure 3. The best signal length is found to be one with 900 samples which gives the accuracy of 72.46%. This justifies our intuition that for architectures with 2D convolution filter, it is better to have an input layer with high 2D dimensions and it also allows to create a deeper network which increases the receptive field.

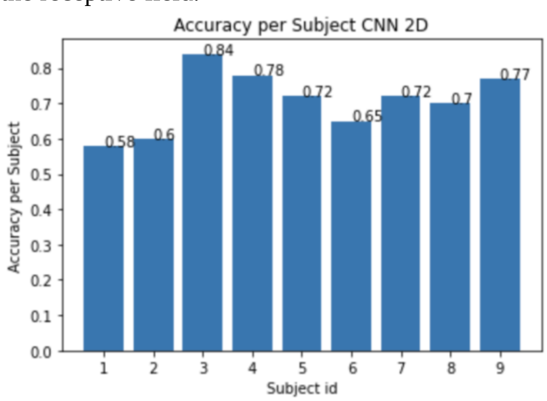

Figure 4: 2D CNN performance on different subjects

Performance on different subjects: To measure the generalization of our best achieved model we evaluate it on different subjects and find that on average it is achieving accuracy of 70.66% with standard deviation of 8.04%. From figure 4. we can infer that it's generalizing quite well but for some subjects [1, 2], the performance is low which can be linked to noisy data.

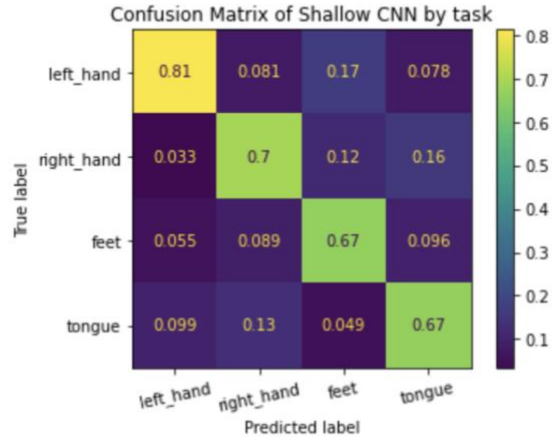

Figure 5: Confusion Matrix for 2D CNN results to analyze its efficacy for different motor activities.

Performance on different Motor Activities: We evaluated the best performing architecture on different motor activities using confusion matrix as shown in figure 5. We observe that left hand and right-hand activities are classified well with comparatively high true positives as compared to feet and tongue activities which are confused with the hand activities.

Effect of Ensembling during testing: As the data per activity is quite low, we augment it with random cropping i.e. randomly cropping the signal of fixed length at different time. We take into consideration two variations of this augmentation method. First, we took 20 crops of widow size 250, from the first 500 samples. The validation error with this approach is 60.46 % and the overall test accuracy we achieve is 60.48%. Second, we took 20 crops of window size 600, from the full 1000 samples. The validation error with this approach turns out to be 62% with overall testing accuracy of 61.11%. For both above methods Windowing preprocessing with window size of 5 is applied and 2D CNN with best found hyperparameters are used. But the above testing accuracies are not true representatives of actual test error since the testing dataset is also augmented in the process. Therefore, we took maximum voting among all the 20 crops for each test sample and find the true accuracies to be 65.19% for first method and 69.07% for the second method. Thus, we can conclude that cropping with max Voting boosts the performance by ~8% and signal length taken into consideration is an important aspect for this classification task. The results from figure 3 supports the above analysis.

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.

[3] Dey, Rahul, and Fathi M. Salem. "Gate-variants of gated recurrent unit (GRU) neural networks." 2017 IEEE 60th international midwest symposium on circuits and systems (MWSCAS). IEEE, 2017

[4] Greff, Klaus, et al. "LSTM: A search space odyssey." IEEE transactions on neural networks and learning systems 28.10 (2016): 2222-2232.

[5] Li, Xiaomeng, et al. "H-DenseUNet: hybrid densely connected UNet for liver and tumor segmentation from CT volumes." IEEE transactions on medical imaging 37.12 (2018): 2663-2674.

[6] https://spectrum.ieee.org/enabling-superhuman-reflexes-without-feeling-like-a-robot

A. Performance Summary

Architecture	Preprocessing Method	Validation Accuracy (%)	Test Accuracy (%)
1D CNN	Trimming + DA	67.12	68.22
2D CNN	Windowing	**70.12**	**72.01**
1D CNN + LSTM	Trimming + DA	62.03	64.00
1D CNN + GRU	Trimming + DA	61.82	61.90
UNet	Trimming + DA	47.00	44.00

Table 1: Motor Activity Classification Validation and Test accuracy

B. Network Architectures

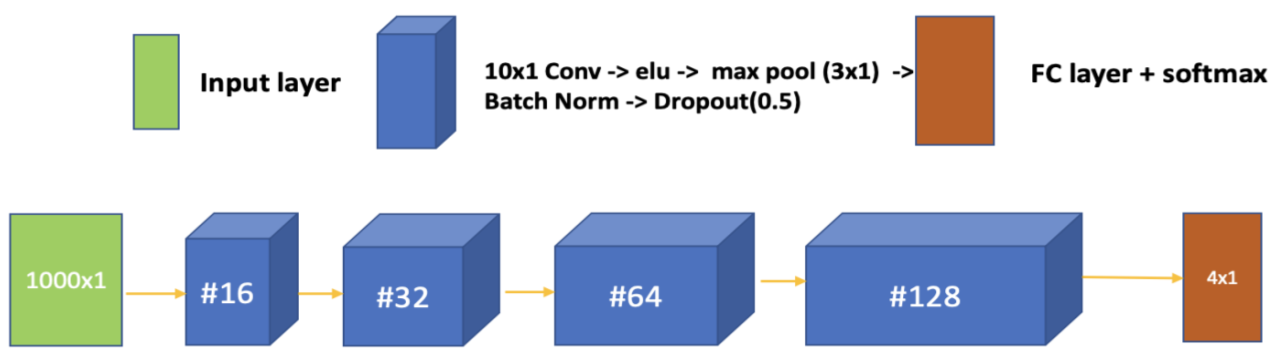

Figure 6: 1D CNN

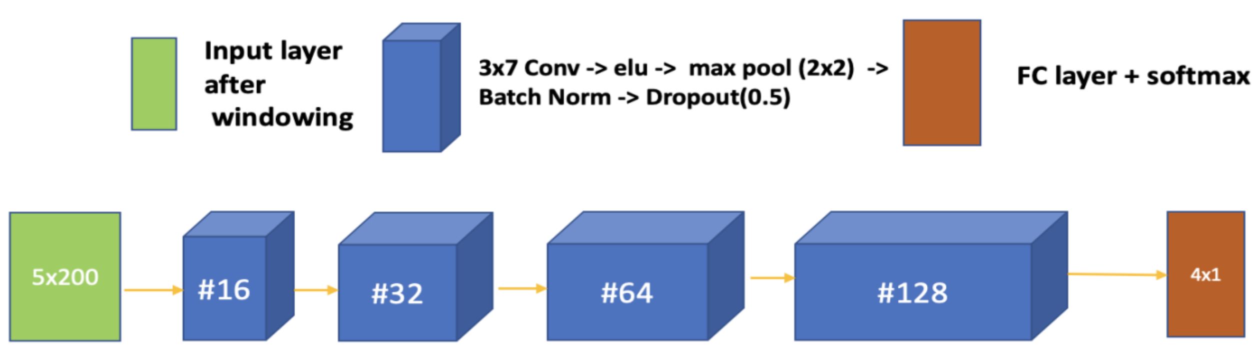

Figure 7: 2D CNN

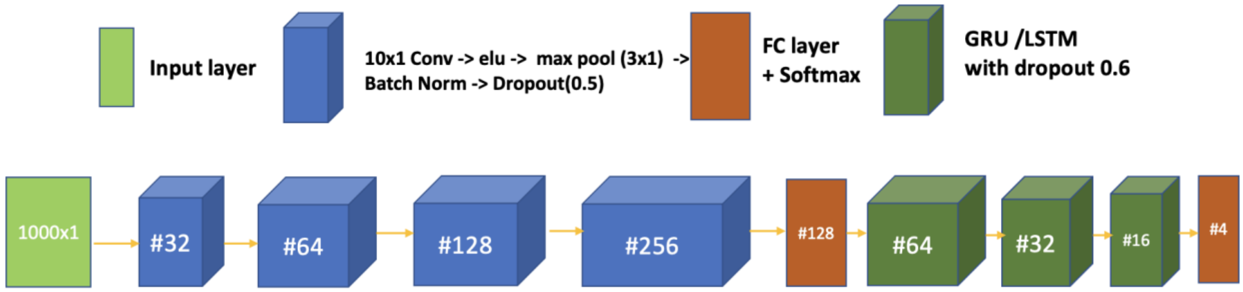

Figure 8: 2D CNN