ECE247 - Winter 2023: Multi-Class EEG Motor Imagery Classification Using Deep Learning Architectures

Ananya Deepak Deoghare
UID: 005627628

Kimaya Kulkarni
UID: 805528337

Sreya Banerjee
UID: 005429893

Abstract

This project explores deep learning techniques for multi-class classification of EEG signals for motor imagery tasks. Various neural network architectures such as CNNs, LSTMs, RNNs, VAEs, Transformers, and attention are implemented and tuned. Effects of pre-processing and time duration on accuracy are analyzed. Results from training on every subject and the entire dataset are compared. The project shows the potential of deep learning techniques for EEG signal analysis and classification, highlighting the accuracy of the models and their ability to learn underlying patterns in EEG data.
Please note that the accuracies that we got for CNN and CNN+LSTM are different due to architectural changes made by us, and the same is represented using asterisk.

1. Introduction

In this project, we explore the use of neural networks to classify electroencephalography (EEG) data for motor imagery tasks[2] [1]. We begin by visualizing the data and performing preprocessing to obtain more meaningful data. Then, we train and compare multiple neural network architectures to predict the imagery task performed by each subject. The details of the project are discussed in the following sections.

1.1. Preprocessing

Pre-processing of the EEG dataset for motor imagery classification included trimming the data to the first 500 time steps and applying various techniques for data augmentation such as max pooling, average pooling with Gaussian noise, and subsampling with Gaussian noise. Concatenating the outputs of these techniques over the sample axis provided a diverse dataset, improving the accuracy of the deep-learning models. These pre-processing steps prepared a robust dataset for the models to learn the underlying patterns in the EEG data.

1.2. Architectures

1.2.1 Convolutional Neural Network(CNN*)

A CNN model was developed using Keras Sequential class to classify EEG data based on imagery tasks. The model included four convolutional blocks with increasing filters, followed by a MaxPooling2D layer, a BatchNormalization layer, and a Dropout layer. The highest accuracy achieved was **72.12%**, with a relu activation function, a learning rate of 1e-5, 150 epochs, and a batch size of 64. Modifications, such as adding an extra convolution block, softmax layer and changing the activation function, did not significantly improve the accuracy. These results suggest that the proposed CNN architecture can effectively classify EEG data based on imagery tasks.

1.2.2 Recurrent Neural Network (CNN + LSTM*)

The model utilized a combination of Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) layers to extract significant features and model temporal dependencies, achieving an accuracy of **72.23%** on the test set. The LSTM layers processed the output of the CNN layers, which was a sequence of feature maps, to produce the final output. The LSTM layer was regularized using dropout to prevent overfitting. Increasing the number of filters caused a decline in performance, which suggests that increasing filter size may result in overfitting and higher memory consumption. Additionally, incorporating multiple LSTM layers with more units did not improve performance, but it did increase memory usage. The architecture was implemented by removing the fully connected network and instead utilizing multiple LSTM layers to capture the temporal dependencies in the Electroencephalogram (EEG) signals and reduced computation time.

1.2.3 Recurrent Neural Network (CNN + GRU)

In this experiment, a CNN+GRU combination was used, as RNNs had shown good performance in previous experiments. We chose GRU as GRU uses less training parameters and therefore uses less memory and executes faster than

LSTM. It also prevents the vanishing gradient problem. In the architecture, CNN layers learned spatial features, while GRU layers captured temporal patterns in sequential data. The model utilized multiple GRU layers following the CNN architecture. The highest accuracy of **74.09%** was obtained with ReLU activation and multilayer GRU using CNN filters. Increasing filter or GRU units did not improve accuracy, while smaller learning rates achieved comparable accuracy. Higher number of filters or GRU units can lead to overfitting and higher memory usage.

1.2.4 VAE

Variational autoencoders (VAEs) consist of an encoder and decoder network, used for unsupervised learning, but the output of the encoder can be given to a classifier for classification tasks. VAEs can learn the underlying distribution of input data and improve accuracy for classification tasks. The encoder network can learn an informative representation of the input data, which is then used for classification. The architecture of combining a CNN with a VAE achieved accuracies ranging from 45% to 68.34% depending on the number of epochs, batch size, and learning rate. The addition of a GRU or LSTM network in combination with the CNN and VAE significantly improved performance. The VAE with CNN+GRU achieved an accuracy of **71.04%** and the VAE with CNN+LSTM achieved an accuracy of **73.36%**. Overall, these results suggest that the combination of CNN and VAE may have potential for certain classification tasks, and the choice of classifier network can also affect performance.

1.2.5 CNN + Attention or CNNA

The CNN model utilizes convolutional blocks, batch normalization, dropout, and self-attention[3] to extract features from the input data. The transformer layer applies self-attention to the feature maps, and a dense layer with a softmax activation function produces the classification output. We experimented with different hyperparameters, including attention, activation function, number of epochs, batch size, learning rate, and dropout rate. The elu activation function with attention and a learning rate of 1e-3 performed the best, and the dropout rate was more effective at 0.6. We also tested different optimizers such as RMSProp and Adagrad. The highest accuracy achieved was **72.68%**. We created another CNN model with attention and squeeze-and-excitation (SE) blocks to improve the representation of input data. The SE blocks recalibrate the feature maps based on channel-wise information by reducing the dimensions of the feature maps and scaling them based on channel importance. However, this model showed lower accuracy compared to the model with only attention. Overall, the results suggest that the proposed CNN architecture with attention and optimized hyperparameters can effectively classify the input data.

2. Results

2.1. Classification for single subject

In part a, the classifier was trained using only the data from subject 1, and then tested on subject 1's data, similarly trained using only data from subject 2 and then tested on subject 2's data, and so on. The results can be found in Table 1, which shows the classification accuracy for three different models.

In part b, the classifier was trained using data from all subjects and then tested on subject 1's data. We did this experiment for all subjects to see if we could observe any pattern. The results can be found in Table 2, which shows the classification accuracy for all models.

Comparing the results in Table 1 and Table 2, we can see that in general, training across all subjects leads to better classification accuracy on subject 1's data. However, the extent of the improvement varies depending on the model. For example, the CNN model sees a relatively small improvement from training across all subjects, while the CNN+GRU model sees a larger improvement.

Another interesting trend to note is that the performance of the models generally varies significantly across different subjects. For example, in Table 1, the CNN model performs well on Subject 4, achieving an accuracy of 0.723, while its performance on Subject 1 is much lower, with an accuracy of only 0.375. Similarly, the CNN with LSTM model performs well on Subject 6, achieving an accuracy of 0.759, but its performance on Subject 1 is only 0.384.

This indicates that the EEG data collected from different subjects may have different characteristics and require different modeling approaches to achieve optimal classification performance. Therefore, it may be necessary to perform subject-specific modeling and optimization to achieve the best results. This may also be indicative of possible presence of noise in some data.

2.2. Classification across all subjects

Looking at the Table 3, we can see that the CNN + GRU model has the highest accuracy among all the models with a value of 0.7409. This indicates that the addition of the GRU layer has helped improve the accuracy of the model. On the other hand, the VAE (CNN) model has the lowest accuracy, which suggests that the VAE technique might not be suitable for this particular classification problem.

Another interesting trend to notice is that models that incorporate LSTM or GRU layers perform better than the basic CNN model. This is expected since LSTM and GRU are designed to work well with sequence data, and this dataset consists of time-series EEG signals. Additionally, the CNN + Attention model has a similar accuracy as the basic CNN model, which might suggest that adding an attention mechanism doesn't necessarily improve the performance of the model in this case.

Overall, the trends suggest that incorporating recurrent layers such as LSTM and GRU can improve classification accuracy, while the VAE technique might not be suitable for this problem.

2.3. Classification accuracy as a function of time

The Table 4 presents the classification accuracy of three models - CNN, CNN+LSTM, and CNN+GRU - trained on EEG data over time. The classification accuracy of all models improves over time, but with fluctuations across time intervals. The CNN+GRU model consistently outperforms the other two models, indicating that the GRU's ability to capture long-term dependencies in sequences may be particularly relevant for EEG data.

The required training time to achieve a reasonable classification accuracy depends on the specific application's requirements. For instance, the CNN+GRU model achieves an accuracy of 0.692 at 500 time units, which may suffice for some applications. However, higher accuracy requirements may necessitate longer training times or more advanced models.

The insights gained from the table align with our expectations, as the model's generalization improves with increasing training data. Moreover, the CNN+GRU model's superiority can be attributed to its ability to capture longer-term dependencies in EEG sequences.

3. Discussion

The results section shows that CNN+GRU had the highest accuracy in most experiments. The best-performing model was chosen based on validation accuracies. The model weights of the epoch just before the validation accuracy started increasing(after decreasing), indicating overfitting, were selected assuming training accuracy was decreasing(i.e. model is learning).

We will discuss our results and the reasoning behind it in this section. These are the contributions made by our work.
Contribution 1: Achieved 74.09% classification accuracy for the action-based task on the entire subject population.
Contribution 2: We identify which subjects have the noisiest data and therefore are most challenging to classify for the action-based task.
Contribution 3: We showed the importance of hyperparameter tuning and choosing correct model weights by using validation accuracy data to achieve optimum accuracy.

Performance on all subjects : CNN+GRU has the highest accuracy of 0.7409, followed by CNN+LSTM* with 0.722. RNN architectures, including LSTM and GRU, excel at processing sequential data due to their ability to capture temporal dependencies and retain information over time. These factors contribute to RNNs' superior performance in EEG classification. VAE is a generative model that learns the distribution of data and generates new data by sampling from this learned distribution. VAEs work well on low noise levels and clear structured data, but struggle with noisy data. VAEs may not be suitable for EEG classification due to the noisy nature of EEG data. Attention mechanism assigns weights to different parts of the input data for models to focus on relevant information. In EEG classification, relevant information may be scattered across the signal, so attention may not improve performance compared to models that process the entire signal.

Subject-wise performance: The accuracies for EEG classification vary greatly across subjects, indicating varying levels of noise in signals. Architecture performance trends differ across subjects, suggesting that optimal architecture varies depending on signal characteristics. We also trained models on all subjects and tested them on subject-wise data. These accuracies were better than the ones observed in the previous experiment, maybe because the model could generalize well for different data. It can also be seen that accuracy values vary a lot and this may be due to noise present in the data itself.

Performance based on time: The CNN+GRU model outperforms the CNN* and CNN+LSTM* models in later stages in terms of time, indicating the effectiveness of GRU in handling temporal information. GRU has a gating mechanism that can better model temporal dependencies in EEG signals, leading to better accuracy with fewer parameters and reduced risk of overfitting.

To avoid overfitting of data we observed validation accuracy decrease and increase, and chose the model just before an increase in val accuracy. We also tuned the learning rate, activation function, optimization technique, and other hyperparameters to ensure maximum test accuracy value.

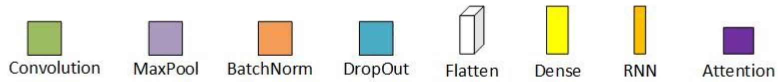
References

[1] Bci competition iv. www.bbci.de/competition/iv/. 1

[2] Clemens Brunner, Robert Leeb, Gernot Müller-Putz, Alois Schlögl, and Gert 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. 1

[3] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 2

4. Model Architecture & Performance



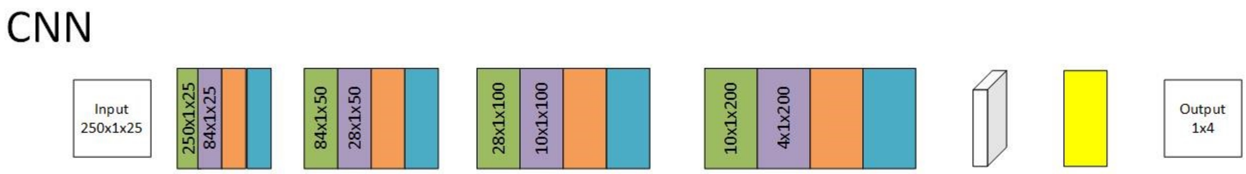


Figure 1

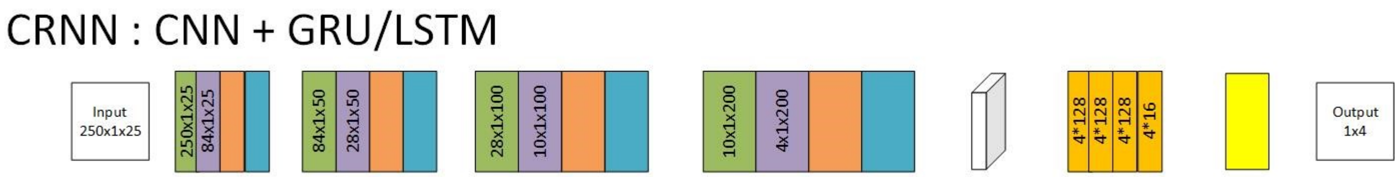


Figure 2

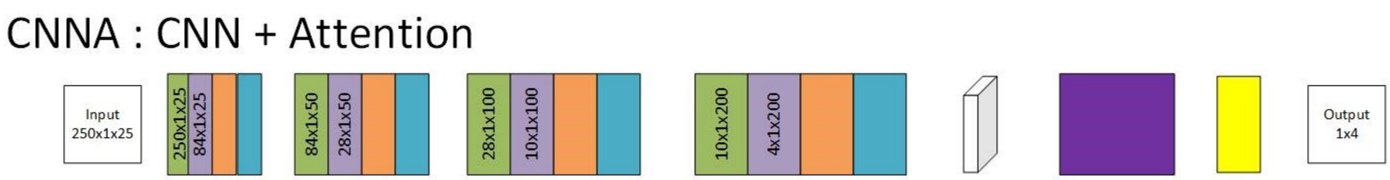


Figure 3

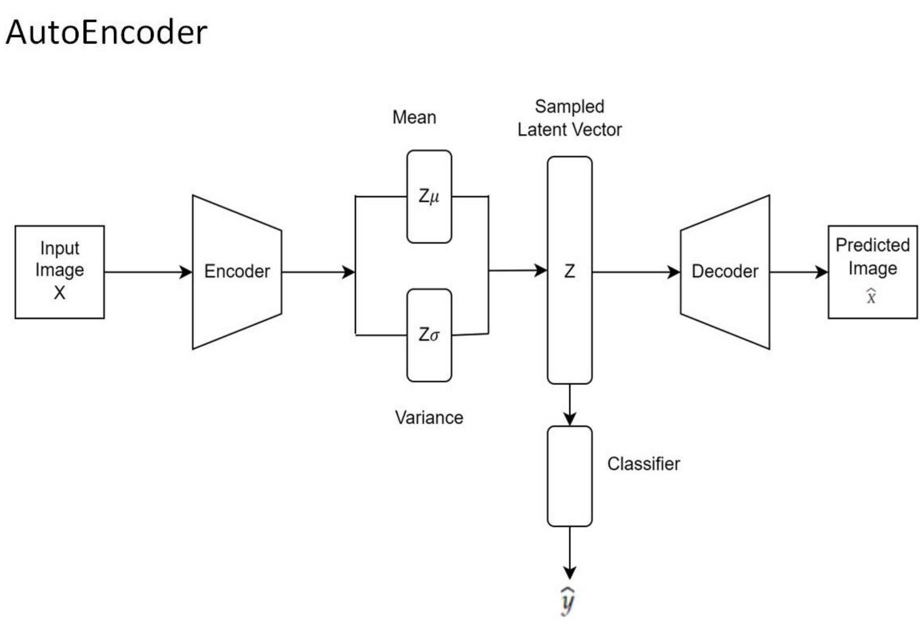


Figure 4

Fig 1: Base CNN Architecture, Fig 2: CNN + RNN Architecture, Fig 3: CNN with Attention block, Fig 4: Variational AutoEncoder Decoder Architecture with classifier

Subject	CNN*	CNN with LSTM*	CNN+GRU
0	0.439	0.449	0.474
1	0.375	0.384	0.264
2	0.540	0.519	0.425
3	0.479	0.405	0.389
4	0.723	0.409	0.473
5	0.367	0.428	0.459
6	0.625	0.759	0.535
7	0.605	0.435	0.395
8	0.670	0.515	0.569

Table 1: Training and testing subject-wise data

Subject	Model								
	CNN*	CNNA with relu	CNNA with elu	CNN with relu	CNN + LSTM*	CNN + GRU	VAE + CNN	VAE with CNN+GRU	VAE with CNN+LSTM
0	0.62	0.439	0.64	0.62	0.600	0.639	0.560	0.639	0.560
1	0.579	0.479	0.64	0.62	0.460	0.600	0.540	0.479	0.579
2	0.82	0.72	0.82	0.759	0.779	0.800	0.740	0.759	0.720
3	0.699	0.759	0.66	0.62	0.680	0.680	0.600	0.759	0.660
4	0.787	0.723	0.787	0.702	0.787	0.787	0.680	0.787	0.765
5	0.673	0.714	0.775	0.693	0.673	0.673	0.632	0.653	0.734
6	0.759	0.74	0.759	0.74	0.720	0.759	0.639	0.740	0.720
7	0.74	0.759	0.759	0.639	0.699	0.699	0.680	0.720	0.720
8	0.723	0.787	0.765	0.787	0.723	0.787	0.787	0.765	0.829

Table 2: Accuracy values for different models on each subject

Model	Accuracy
CNN*	0.710
CNN + LSTM*	0.722
CNN + GRU	0.7409
CNN + Attention	0.729
VAE(CNN + LSTM)	0.733
VAE (CNN + GRU)	0.7104
VAE (CNN)	0.683

Table 3: Classification accuracy across all subjects

Time	CNN*	CNN + LSTM*	CNN + GRU
100	0.543	0.494	0.514
200	0.636	0.557	0.593
300	0.692	0.664	0.686
400	0.721	0.655	0.686
500	0.686	0.675	0.692
600	0.689	0.683	0.681
700	0.679	0.678	0.703
800	0.677	0.664	0.683
900	0.682	0.709	0.694
1000	0.664	0.665	0.687

Table 4: Classification accuracy as a function of time