EEG Classification by Implementing CNN, RNN and ViT

Yinglu Deng
SID: 305496193
ceciliadeng12@g.ucla.edu

Zhi Zuo
SID: 305346349
joannazz@g.ucla.edu

Abstract

The goal of this report is to optimize the classification of electroencephalography (EEG) data, which is provided by the Brain-Computer Interaction (BCI) Competition[2]. To achieve this, deep learning models such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Vision Transformer (ViT) Model are employed and compared in terms of their performance. The report describes the methodology used in this project, including data preprocessing, model architecture, and evaluation metrics. By analyzing the results, this study provides insights into the effectiveness of deep learning models in improving the accuracy of EEG data classification, which is relevant to future research in the field of BCI.

1. Introduction

This part details the methodology and examines the impact of hyper-parameters and data processing on the performance of various network architectures.

1.1. Data Inspection & Pre-processing

Initially, we attempted to visualize the raw data, which comprised 22 electrodes across 1,000 time bins. However, this approach did not yield discernable details or valuable information.

To gain further insights, we delved deeper into the data by analyzing and visualizing the time series of a specific EEG channel during each mental imagery task. The graph reveals that the initial portion of the signal contains pronounced fluctuations, encompassing both global maximum and minimum points that could serve as crucial features for identification. In contrast, the latter half of the signals tends to flatten, rendering them less distinguishable. So, we applied preprocessing techniques to the raw data by trimming it in half for extracting valuable information and enhancing model performance. This refined data was subsequently utilized in both the ConvLSTM and ViT models as input data.

1.2. Pure CNN Model Architecture

The Convolutional Neural Network (CNN) is a deep learning model specifically tailored for processing grid-like data structures, such as images. In our research, biomedical signal graphs can be regarded as image-like data, making the CNN model highly suitable for training and analysis in this context.

In designing the CNN architecture, the first convolutional layer employs a single filter with a kernel size of (1, 5), stride of (1, 3), and padding of 1 to effectively filter the input signal and remove extraneous information. Following this, the signal is processed through residual blocks, which used skip connection and serve to double the channel count from 22 to 44 while preserving the original signal length. [4] As per conventional CNN methodologies, the architecture progressively increases the number of channels and simultaneously reduces the size of the feature map. The feature map, in this case, is a one-dimensional vector representing the signal length, which transitions from (332 x 1) to (166 x 1) and finally to (83 x 1) throughout the processing stages. [5]

1.3. RNN (CNN-LSTM Hybrid) Architecture

We create a hybrid model that combines a Convolutional Neural Network (CNN) with a Long Short-Term Memory (LSTM) network, which is a specialized type of Recurrent Neural Network (RNN). Our main idea for this architecture is: First, the signal of the EGG data is filtered by the first convolution layer, and some of the less needed signals are processed, then the signals after filtering are fed into CNN and LSTM respectively to extract image features and timing features, and then they are flattened and concatenated together and fed into a fully connected layer. The model is expected to learn both timing and image features.

In the hybrid architecture, the LSTM layer captures long-range dependencies and temporal patterns in the EEG data. The LSTM layer in our architecture has four high-level structures: 1. Input: The LSTM layer takes input from the output of the CNN layers after a series of convolutional,

pooling, and batch normalization operations. 2. LSTM Configuration: The LSTM layer is configured with 256 hidden sizes, and it is set to have a single layer. 3. LSTM Operations: The LSTM processes the input data, capturing temporal dependencies and generating an output for each time step. The final output at the last time step is then passed to the next layer in the model. 4. Concatenation: The outputs from the CNN layers and the LSTM layer are concatenated to form a combined feature representation of the input EEG data. [3]

1.4. Vision Transformer (ViT) Architecture

The Vision Transformer (ViT) model has shown promising results in EEG (electroencephalography) modeling because it can capture long-range dependencies and modeling spatial relationships between different parts of the input signal. EEG signals contain information that is spread across different electrodes and frequencies, and the ViT model's ability to learn these complex relationships makes it a suitable choice for EEG modeling.

Our Vision Transformer model was developed based on the inspiration from the paper *Transformers for Image Recognition at Scale*[1]. To extract information from the signal image of size 22 x 500, we divided it into small patches with size 22 x 5 and performed patch embedding with embedding dimension 22. And by combining the patch embedding with the first layer of conv2d, we were able to obtain information about each patch. We then separately inputted the patch embeddings into a multi-head attention layer with n_head = 4 and a CNN layer, then concatenated the results. This approach allowed us to effectively extract meaningful features from the signal image and improve the overall performance of our model.

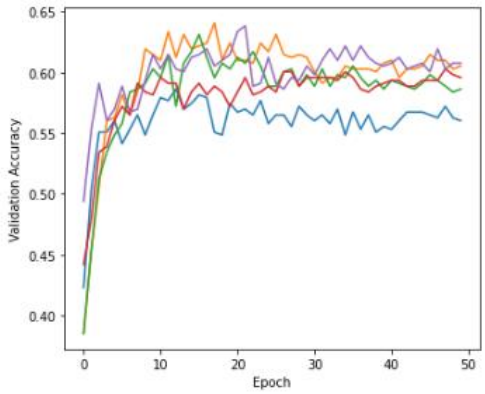
2. Results

2.1. CNN Training

The optimal test accuracy of the CNN model is 0.73584. The best model was trained over 100 epochs with a learning rate of 0.005 and a dropout rate of 0.5. We evaluated the performance of the CNN model using 5-fold cross-validation and got the validation accuracy between 0.65 to 0.75. Moreover, for tuning the hyperparameter, we set the dropout within a range of 0.5 to 0.8. However, model performance declines when the dropout exceeds 0.5. The model performs particularly poorly when the dropout is at 0.8, and the test acc is only 0.54.

2.2. RNN (CNN-LSTM Hybrid)

Based on the results of the CNN-LSTM hybrid model, the test accuracy was found to be **0.62528**. To train the model, we utilized 5-fold cross-validation, which involves partitioning the dataset into 5 parts and performing 5 separate training and validation runs. The validation accuracy for each of the 5 runs was found to be within the range of 0.55 to 0.65, indicating that the model's performance is consistent across different folds.



The confusion matrix (appendix2.3-CM2) revealed that the left-hand class (class0) had the highest probability of being misidentified as the right-hand class (class1). The right-hand class (class1) had the highest probability of being misidentified as the tongue class (class3). Additionally, both feet classes (class2) had the highest probability of being misidentified as the right-hand class (class1), as well as the tongue class (class3).

2.3. Vision Transformer (ViT) Training

The ViT achieves a peak test accuracy of 0.72686. The optimal model configuration included training for 100 epochs, employing a learning rate of 0.003 and a dropout rate of 0.5. Through 5-fold cross-validation, the model demonstrated consistent performance with accuracies ranging between 0.63 and 0.7.

To fine-tune the learning rate, we conducted a series of experiments within a range of 0.001 to 0.01. The selected learning rate of 0.003 was the optimal value for our model. It facilitated faster convergence during training, striking a balance between the speed of learning and the stability of the model. The bigger learning rate may cause the optimization algorithm to overshoot the optimal solution, resulting in oscillations and instability in the training loss.

3. Discussion

3.1. CNN & ViT Performance

The CNN model and ViT model both get great performance in the EEG task with 0.7358 test accuracy and 0.7268 test accuracy. CNN performs best in our training.

For CNN, the CNN model can effectively extract spatio-temporal features through local convolution operations to capture local patterns in EEG signals. This is very useful for analyzing EEG signals, as local features are often associated with biological sources (e.g. specific brain regions). Meanwhile, the convolutional and pooling layers of CNN help to integrate information from multiple moments and spatial locations. And in our model design, we set 4 convolutional layers to CNN which is very important when dealing with spatiotemporal correlations in EEG tasks. [6]

While for ViT, the ViT model can effectively capture the long-distance dependencies between signals in EEG tasks through the self-attention mechanism, which enables it to handle complex spatio-temporal patterns.

3.2. RNN (CNN-LSTM Hybrid) Performance

In the analysis of the CNN-LSTM hybrid model, we observed that the validation accuracy showed an increasing trend over time, and after epoch 30, the accuracy level remained relatively constant. This indicates that the model was able to learn and generalize well on the task of interest.

Moreover, the training loss and validation loss showed a decreasing trend as the model was trained with 50 epochs. (appendix2.4-figure2) However, after epoch 30, the loss values remained relatively constant, indicating that the model had reached its optimal performance and was not overfitting to the training data.

Overall, the results suggest that the CNN-LSTM hybrid model is able to perform reasonably well on the task of classifying motor imagery EEG signals, with a test accuracy of 0.62528. However, the misclassification patterns revealed by the confusion matrix indicate that the model may have difficulty distinguishing between certain classes, particularly the tongue and feet classes.

3.3. Compare Between Different Models

The ConvLSTM has only 0.6252 test accuracy and does not perform as well as other models, one of the reasons may be the EEG raw data sequence is too long, the gradients can become very small and vanish as they are propagated backwards through time. This can make it difficult for the model to learn long-term dependencies in the sequence.

In our experiment, the CNN model works best with the EEG task with 0.7358 test accuracy. It may be because we add a residual block, and it makes CNNs more robust to variations in input. The advantage of training with CNN is it can be trained end-to-end, the model can learn the feature extraction and classification tasks simultaneously, potentially leading to improved performance.

In the future, there are several things that can be done to potentially improve the CNN model's performance like exploring alternative architectures such as Inception networks may lead to better performance. Or incorporate additional data sources, combining EEG data with other physiological signals or behavioral data may provide additional information. Also, we can use transfer learning, pre-training the CNN model on a large, diverse dataset such as ImageNet and then fine-tuning it on the EEG dataset.

3.4. Compare Between All vs One (subject 0)

In this analysis, we aimed to investigate if training across all subjects will be better for improving the test accuracy for subject 0. To do so, we trained two CNN models: one using all subjects' trials to train, and the other only using subject 0's trials that we extract from X_train by using the tag. The accuracy of the models was then tested on the subject 0's test dataset.

Our results indicated that training across all subjects improved the test accuracy for subject0. Specifically, when the model was trained on a dataset that contained trials from all subjects, the accuracy on subject0's test dataset was 0.66. In contrast, when the model was trained on a dataset that only contained subject 0's trials, the accuracy on the same test dataset was 0.54.

This suggests that training on a single-subject dataset may not be sufficient for the CNN model to generalize well to new, unseen data. This is because the variance in a single-subject dataset is small and may lead to overfitting. On the other hand, using data from multiple subjects increases the size and diversity of the training dataset, which may help the model to capture more of the underlying patterns and relationships in the input data, resulting in a better generalization ability.

References

[1] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," arXiv.org, 03-Jun-2021. [Online]. Available: https://arxiv.org/abs/2010.11929. [Accessed: 20-Mar-2023].

[2] C. Brunner, R. Leeb, G. R. M¨uller-Putz, A. Schl¨ogl, and G. Pfurtscheller. BCI Competition 2008 – Graz data set A.

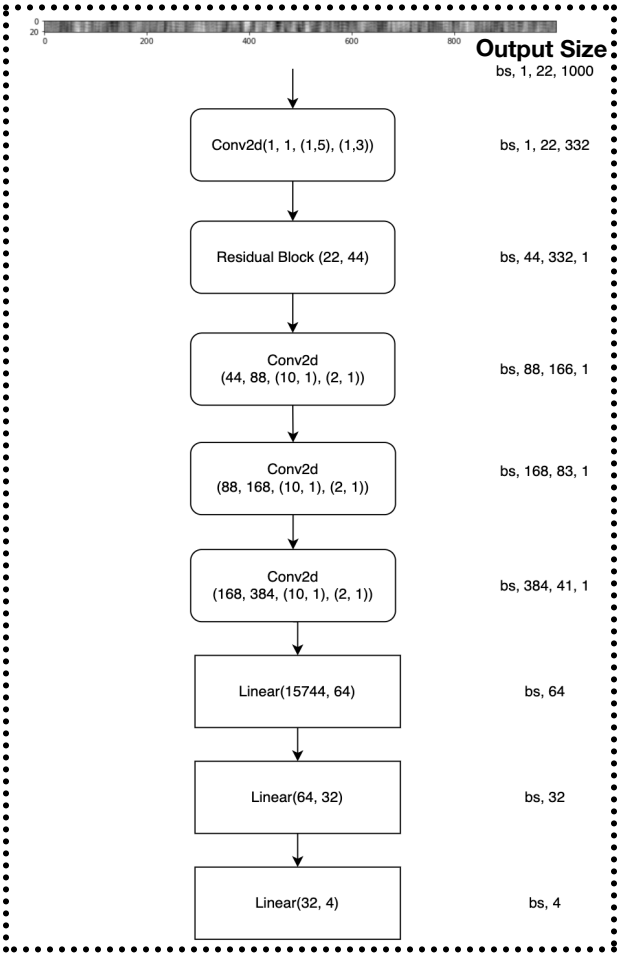
[3] Has¸im Sak, Andrew Senior, Franc¸oise Beaufays. "Long short-term memory recurrent neural network architectures for large Scale Acoustic Modeling". Available: https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/43905.pdf . [online]

[4] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. arXiv 1512:03385 [cs] arXiv:1512.03385. 2015.

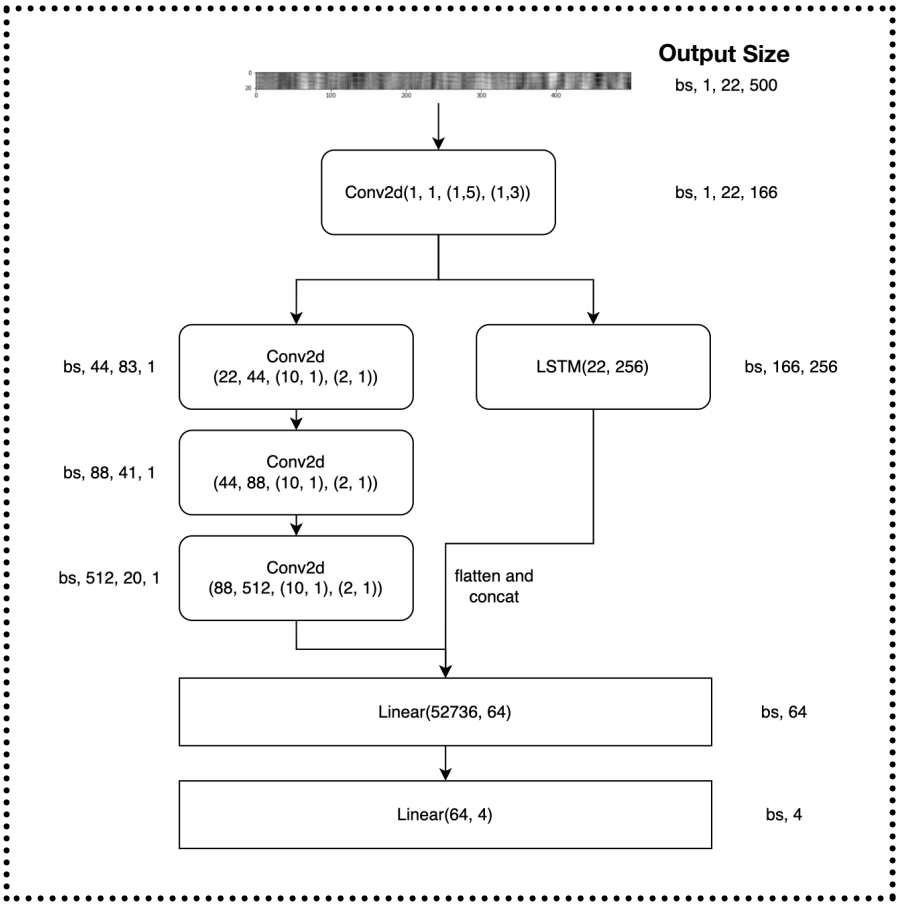
[5] R. T. Schirrmeister, J. T. Springenberg, L. D. J. Fiederer, M. Glasstetter, K. Eggensperger, M. Tangermann, F. Hutter, W. Burgard, and T. Ball, "Deep learning with convolutional neural networks for EEG decoding and visualization," *arXiv.org*, 08-Jun-2018.

[6] Vernon J Lawhern, Amelia J Solon, Nicholas R Waytowich, Stephen M Gordon, Chou P Hung and Brent J Lance. EEGNet: a compact convolutional neural network for EEG-based brain–computer interfaces, 2018.

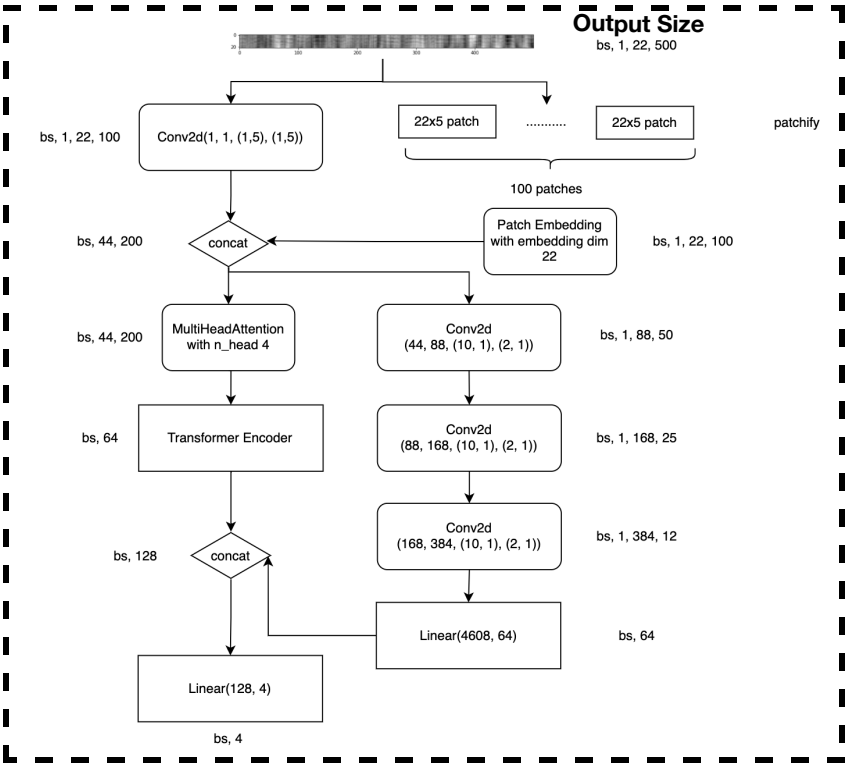
Appendix 1: Three model's architecture



Model 1: Pure CNN Architectures



Model 2: Recurrent Neural Network Architectures



Model 3: Vision Transformer Architectures

Appendix 2: Tables and Important Graph

1. Table 1:

Model	Parameters	Test Accuracy
CNN	epochs: 100 drop out: 0.5	0.7358
RNN	epochs: 50 hidden sizes: 256 num layers: 1 drop out: 0.5	0.62528
Vision Transformer	epochs: 100 drop out: 0.5	0.72686

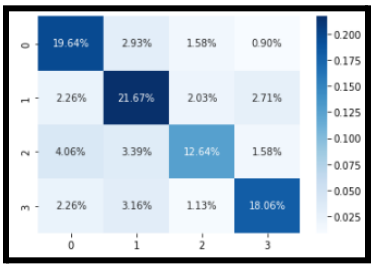
Table 1. The Final Models' Testing Accuracy

2. Table 2:

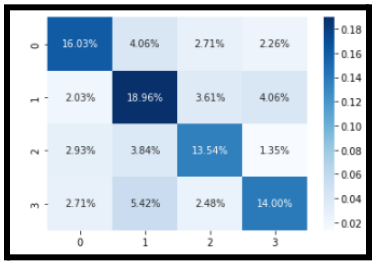
Model	Train set	Test Accuracy
CNN	All subjects	0.66
CNN	Subject 0	0.54

Table 2. All vs One (subject0) Testing Accuracy

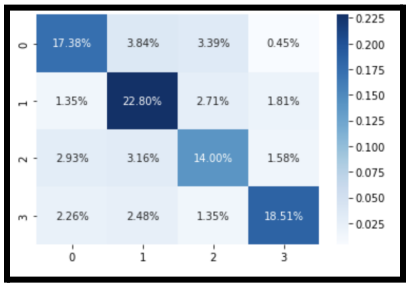
3. Confusion Matrix:



CM 1. CM for CNN



CM 2. CM for CNN-LSTM



CM 3. CM for Vision transformation

4. Training loss and validation loss:

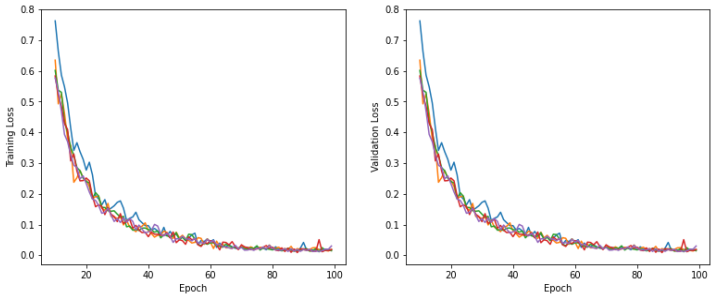


figure 1. Train & Validation loss for CNN

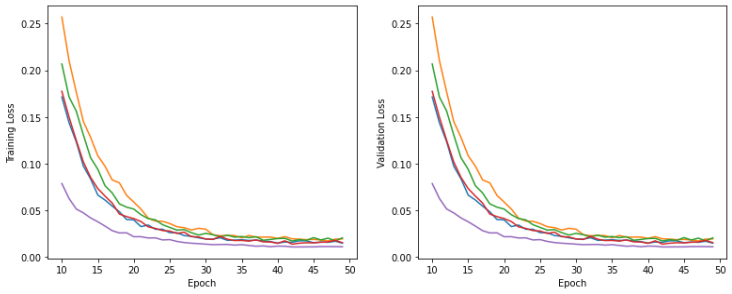


figure 2. Train & Validation loss for CNN-LSTM

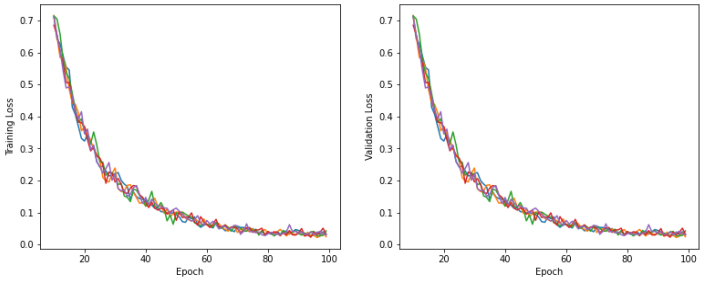


figure 3. Train & Validation loss for ViT

5. Validation accuracy

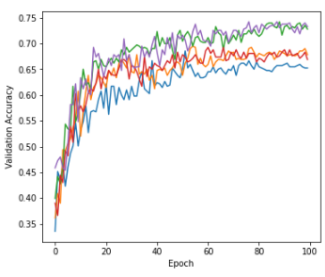


figure 1. CNN

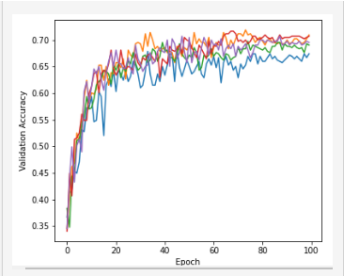


figure 2. ViT

6. Table 3: Different drop out to train CNN

Drop out	0.5	0.6	0.7	0.8
Test acc.	0.74	0.72	0.65	0.54