A Suvery of Deep Learning EEG Decoders

Ryan Wang
UID: 405271956
ryanwang25@g.ucla.edu

Eric Liu
UID: 305735283
eliu0120@gmail.com

Ryan Chaiyakul
UID: 606028764
ryanchaiyakul@gmail.com

Abstract

Deep learning has taken the world by storm by becoming the de facto end-to-end solution for complex tasks across many fields. One niche that has benefited from modern advancements is EEG analysis. Here, we postulate potential improvements through architecture choices and data augmentation and compare them to baseline models. In particular, we combined domain specific data augmentation with new LSTM layers in hopes of a better model. While our optimized CNN model performed the best, our exploratory work has suggested guidelines backed by empirical results and laid new avenues to study.

1. Introduction

In 2018, Tonio Ball et al. [8] applied various deep learning networks consisting of standard convolution blocks and residual blocks from ResNet [2] in hopes to match or exceed the performance of FBCSP based designs, the winner of the BCI competition IV dataset 2a [1]. While they failed to overtly outperform FBCSP, their ability to match the performance with the deep ConvNet using alternative features suggested that deep learning models could leverage novel features for possibly better accuracy.

By leveraging newer advancements such as the LSTM layer and various data augmentation techniques, we tried to create a model which could outperform the FBCSP algorithm: the baseline.

1.1. Architectures

1.1.1 CNN

Since ILSVRC2012 [7] where AlexNet dropped the top-5 error by 10.4 percent, CNNs have made a resurgence as the starting model for many deep learning tasks. Following the steps of Tonio Ball et al. [8], we decided to implement our own CNN model that crucially forgoes their discrete temporal and spatial convolution for a simple single convolution layer allowing our model to learn its own mapping without restraint.

Inspired VGGnet, we chose small (3x3) filters for the convolution layers. Corresponding to these smaller filters, we expanded the depth to 100 filters thus leveraging the advantages of a bigger receptive field as well as adding more non-linearity. Furthermore, with insight that more layers have empircally been shown to offer advantages, after trying variable number of convolution blocks, we settled on 4 as it offered the best performance.

1.1.2 CNN + LSTM

Given the time-based nature of the EEG dataset which consists of voltages from 22 electrode taken over 1000 time steps, we explored hybrid architectures which combined CNN and LSTM layers. Based on the insight that hybrid models combining CNNs and RNNs have been competitive in the realm of speech recognition and EEG, similar to speech is a time-based waveform [8], we believe this architecture could yield promising results. We created two CNN + LSTM architectures, one based on a cropped input and another optimized for the data augmentation pathway described later.

1.1.3 LSTM / CSP + LSTM / CSP + LSTM + LDA

While the ResNet tested in Tonio Ball et al. [8] performed considerably worse than their ConvNets, we wanted to test a strictly network made of strictly LSTMs, a modern iteration of residual layers, and a softmax output layer. Inspired by Kumar et al. [4] who improved upon the BCI competition IV dataset 1 [1] with a CSP + LSTM + LDA model, we tested their preprocessing techniques on a traditional LSTM network to reduce the dimensions of the feature vector for faster training speed and hopefully better accuracy.

Architecture Breakdown

The final architecture for CSP + LSTM starts with a custom time segmentation layer which split the input from N x C x t to N x W x C x t' where W was the # of windows and t' was the new time steps. This design was inspired by Optical [4] and Tonio Ball et al. [8] success with using cropped slices to attempt to be resistant to time shifting. These windows were passed into a CSP layer trained on the augmented training set to become N x W x F. This is passed into the LSTM layers as W is a a time step of a windowed view of the dataset. Finally, a Dense layer outputs the results as a softmax.

Inspired by Optical's [4] success with an alternate pathway, we tried a non sequential model to leverage various views of the data. An additional path passed the input into a different CSP and LDA reduced the dimensionality further. This was concatenated with the output of the last LSTM to be fed into the dense layer.

By adding an alternative pathway, we allow the model to consider both windowed and full views of the data which is important as EEG is assumed to be a superposition of global voltage patterns and windows may lack the full picture.

1.1.4 FC

For a baseline model, we implemented a FC network with only dense layers and a softmax output. Due to dense connections within the model, we expected the model to overfit the training dataset and take a considerably amount of time to train due to the number of parameters. By using the data augmentation pathway and PCA covered later in this paper, we reduced the training speed to a competitive time span to get empirical results.

1.2. Data Augmentation

1.2.1 CNN + LSTM and FC Pathway

To reduce the dimensions of the feature vector and increase the trial count, a series of steps were performed.

1. The last 200 time steps of each trial are trimmed.
2. The remaining 800 time steps are sub sampled into 400 time steps by maxpool.
3. *n* additional trials are added as the mean of the dataset + gaussian noise.
4. The initial dateset is subsampled with noise another 2 times with a step size of 2 and a starting index of 0 and 1.

This results in a final dataset of 4*n x 22 x 400. During testing, only the first two techniques were implemented resulting in a n x 22 x 400 dataset.

1.2.2 PCA

Principal Component Analysis is a linear dimensionality reduction technique used to transform data onto a new coordinate system with lower dimensions. We utilized this with the fully connected neural network to significantly increase the training speed of the model. We reduced the number of features from 8800 (the channels multiplied by the number of time steps) to 566, while capturing 95% of the variance of the original data.

1.2.3 CSP

Common Spatial Pattern is an algorithm which separates multivariate signals into sub components with maximum variance. Both the FBCSP and Optical models use this technique on the frequency bands and windows respectively to reduce their dimensional. This was used in our LSTM model to extract features from the time segmented windows while reducing dimensionality.

1.2.4 LDA

Linear Discriminant Analysis is an alternative method of linear dimensionality reduction and was used in the CSP + LSTM + LDA model to reduce the output of the alternative CSP.

1.3. Hyper-parameter Search

We used Keras Tuner [6] with hyper-band tuning to find optimal hyper parameters for our models. We first tuned the learning rate of the model. Next we tuned for optimization of number of filters, kernel width, and kernel length of the convolution layers as well as dropout. Given limited computational power, we believe using the hyper-band tuner was the right choice given the speed-up orders of magnitude over other methods such as Bayesian optimization [5]. However, even with this, limited computational power prevented us from searching the entire hyper-parameter space in a reasonable time, as the model with optimal hyperparameters from the hyper-band tuner after 5.5 hours of runtime performed worse than the model we found through empirical rationale.

2. Results

We evaluated results based on best test-accuracy achieved by a given model. These models are trained on the entire dataset and the results for the model trained on subject one can be found in Table 1.

2.1. CNN

The unoptimized CNN model, when trained on all, achieved a 50% testing accuracy for subject 1 and 60.7% testing accuracy on the entire dataset. Our optimized CNN model, trained on all and tested on both subject 1 and all, achieved a testing accuracy of 58.0% and 71.1%.

2.2. CNN + LSTM

Without the data augmentation pathway and optimizations, the CNN + LSTM model achieved a 60.9% testing accuracy for the entire test dataset and 46.0% testing accuracy for subject 1 when trained on the entire dataset. With the data augmentation pathway and optimizations, the architecture achieved a 69.1% testing accuracy for the entire test dataset and 54.0% testing accuracy for subject 1 when trained on the entire dataset.

2.3. LSTM / CSP + LSTM

The LSTM model, when trained on all, achieved a 34.0% testing accuracy for subject 1 and 30.9% testing accuracy on the entire dataset. The CSP + LSTM model, when trained on all, achieved a 25.9% testing accuracy for subject 1 and 50.3% testing accuracy on the entire dataset.

2.4. FC

The FC model achieved a 42.0% testing accuracy on subject 1's testing data and 45.1% testing accuracy for the entire dataset.

3. Discussion

3.1. Best Performance

Our best model, the CNN, achieved a test accuracy of 71.1%, achieving good performance relative to the FBCSP algorithm (68% from Tonio Ball et al. [8] implementation). Compared to other algorithms we tested, the CNN performed much better, with the closest model being the CNN+LSTM achieving a test accuracy of 69.1%. Compared to state of the art models from literature which have achieved 92% [3] on the same dataset, our model performs relatively subpar.

We trained our CNN model for 94 epochs on Google Colab with their CPU which took 18 minutes. This was an adequate amount of time as our validation error started to taper off at this point. Due to the simplistic nature of a standard CNN model, a GPU was not necessary to get our optimal model.

3.2. Other Takeaways

Takeaway 1: Complex neural networks struggle to avoid overfitting on the small EEG dataset.

When training on subject one's training examples, the model can only see 236 of the 2115 total training dataset. With a feature space of 22,000 (Channels x Time steps) prior to data augmentation, the small dataset causes all of the models to over fit the training dataset which is seen clearly in all models performing worse when trained on subject one's training dataset and tested on the entire test dataset compared to trained on entire training set and tested on entire test set.

This result can be extrapolated to the full training dataset as the LSTM derivative models overfit the entire dataset in 40-50 epochs as shown in Figure 1 [Add figure]. To alleviate this issue, future designs should attempt to do more data augmentation techniques to expand the training dataset or transfer preexisting networks. In fact, Khademi et al. [3] achieved a 92% accuracy on the BCI IV 2a Dataset [1] by transferring Google's InceptionV3.

Additionally, an early CNN + LSTM model we attempted tried to remove the dense layer in front of the LSTM layer, as well as removing multiple maxpooling layer so as to keep the data through the network more time-based for the LSTM layer. This however also resulted in the model overfitting heavily with an increased training accuracy compared to significantly lower testing error compared to our current CNN + LSTM model, which uses l3 regularizers to attempt to combat overfitting.

Takeaway 2: EEG data may have patient specific features

When comparing the testing accuracies of all the models trained on the entire dataset, the test accuracy of the entire dataset was significantly higher than the test accuracy of subject 1. As this trend was observed with all models, it raises the question whether the models have learned features that do not exist within subject 1 to determine the class of the input data. This is further supported as the testing accuracies of all the models trained on subject one's training dataset is highest for testing on subject one for most models: the reverse of training on all.

Our finding is supported by literature which has found that the responsive frequency range varies from subject to subject [4].

Takeaway 3: Data augmentation and PCA helps the models train data better and improves testing accuracy.

PCA reduction was necessary to train the fully connected model effectively. Despite this, the model still performed poorly on both individual testing data as well as the entire testing dataset. Due to the dense connections between all neurons, the FC-model likely over fit the training data, resulting in worse validation and testing accuracy.

Additionally, data augmentation had a clear impact on the CNN + LSTM model, increasing testing accuracy on subject 1 and the entire dataset by 8% and 8.2% respectively. This arises due to both the increased standardization and simplicty of the data after augmentation which enables the model to trainer quicker and evaluate novel test cases more accurately.

By adding the time segmenting and CSP layers to the LSTM model, the testing accuracy on entire dataset increased by 19.4%. While the training accuracy on subject one decreased, this could be explained by the LSTM's tendency to overfit the data and the uniqueness of EEG data. Unfortunately, the LDA layer did not increase the accuracy of the LSTM model as it was already overfitting the data. Overall, data augmentation was successful in improving the LSTM model.

References

[1] Bci competition iv. 1, 3

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. 1

[3] Zahra Khademi, Farideh Ebrahimi, and Hussain Montazery Kordy. A transfer learning-based cnn and lstm hybrid deep learning model to classify motor imagery eeg signals. *Computers in Biology and Medicine*, 143:105288, 2022. 3

[4] Shiu Kumar, Alok Sharma, and Tatsuhiko Tsunoda. Brain wave classification using long short-term memory network based optical predictor. *Scientific Reports*, 9(1):9153, Jun 2019. 1, 2, 3

[5] Lisha Li, Kevin Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar. Hyperband: A novel bandit-based approach to hyperparameter optimization, 2018. 2

[6] Tom O'Malley, Elie Bursztein, James Long, François Chollet, Haifeng Jin, Luca Invernizzi, et al. Kerastuner. https://github.com/keras-team/keras-tuner, 2019. 2

[7] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. 1

[8] Robin Tibor Schirrmeister, Jost Tobias Springenberg, Lukas Dominique Josef Fiederer, Martin Glasstetter, Katharina Eggensperger, Michael Tangermann, Frank Hutter, Wolfram Burgard, and Tonio Ball. Deep learning with convolutional neural networks for eeg decoding and visualization. *Human Brain Mapping*, 38(11):5391–5420, 2017. 1, 3

Table 1. Table of Model Test Accuracy Results

	Trained on 1		Trained on All	
Model	Tested on 1	Tested on All	Tested on 1	Tested on All
BasicCNN	X	X	0.500	0.607
Optimized CNN	0.540	0.404	0.580	0.711
BasicLSTM	0.259	0.300	0.340	0.309
CSP + LSTM	0.500	0.293	0.259	0.503
CSP+LSTM+LDA	0.500	0.278	0.200	0.479
Basic CNN + LSTM	X	X	0.440	0.508
Optimized CNN+LSTM	0.340	0.368	0.540	0.691
PCA+FC	X	X	0.420	0.451