Brain Invaders: P300-based Brain-Computer Interface

Comparison of the classification models

Group members:

- Aung Zar Lin
- Lin Tun Naing
- Min Khant Soe
- Win Win Phyo

Introduction & Motivation

- P300 is the largest event-related potential (ERP) which occurs with a positive deflection in voltage at a latency about 300ms after a rare stimulus.
- Dataset: Multi-User P300-based Brain-Computer Interface Dataset
- Construct 3 deep learning models.
- Compared these deep learning and machine learning classification models based on subject specific, train on 1 subject and test with another subject and regional channels.

Objective

- To find the best deep learning model based on the dataset we use

- To improve the accuracy of deep learning models and beat that of the machine learning model as possible as we can.

- To find a channel region where we can get highest accuracy from the models

Related Work

Deep learning based on Batch Normalization for P300 signal detection

- They develop a novel CNN, termed BN3, for detecting P300 signals, where Batch Normalization is introduced in the input and convolutional layers to alleviate over-fitting, and the rectified linear unit (ReLU) is employed in the convolutional layers to accelerate training.

- Their results show that BN3 both achieves the state-of-the-art character recognition performance and that it outperforms existing detection approaches with small flashing epoch numbers.

Related Work

A Novel P300 Classification Algorithm Based on a Principal Component Analysis-Convolutional Neural Network
(https://www.mdpi.com/2076-3417/10/4/1546/htm?fbclid=IwAR0R2ePTkrlOSsPMVAb4n2-hPbtr0RxVZLfQO29hqDICgKMG3wHOXqtHEhM)

- The proposed P300 classification algorithm employed the parallel convolution method to improve the traditional convolutional neural network framework, which can increase the network depth and improve the network's ability to classify P300 electroencephalogram signals.
- The parallel convolution layer could increase the data capacity of the network, and may overcome the lack of features caused by improper selection of the convolution kernel size
- Their proposed P300 classification algorithm can get accuracy rates higher than 90%,

Data Introduction

- The dataset is taken from HAL Paper Id: hal-02173958 (https://hal.archives-ouvertes.fr/hal-02173958)

- Our data is collected at 512 Hz across 38 subjects.

- Each subject participated in 5 sessions of Brain Invaders, a visual P300-based BCI game. The full dataset is available at (https://zenodo.org/record/3267302#.YY_Yl71By3I)

- A repetition is composed of 12 flashes of pseudo-random groups of six symbols chosen and each symbol has flashed exactly two times.

Data Introduction

- Red : Target flash
- White : Non-Target flash
- Grey : Non-Target no flash

Project Flow

Data Preprocessing →

- ML
- BN3
- Conv2D
- CNN + LSTM

→

- Train on one sub, Test another sub
- Subject Specific
- Regional Channel
- Subject pool

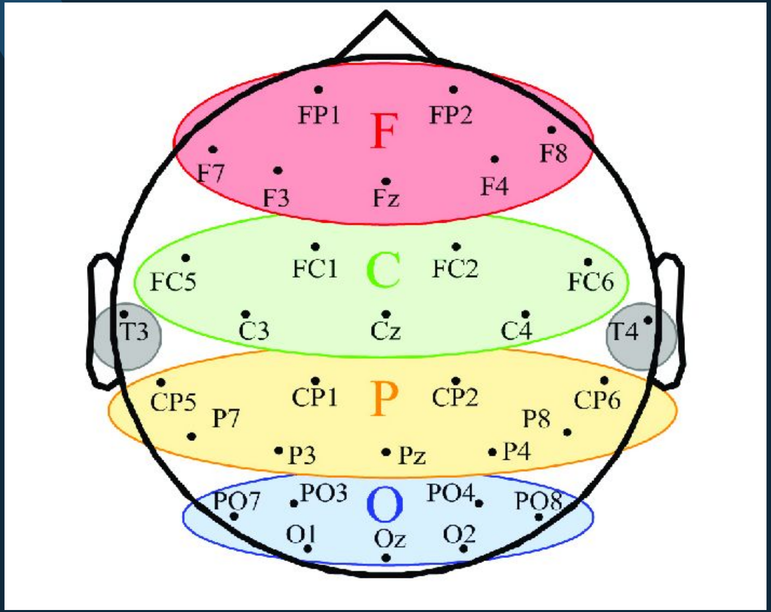
Subject Specific

	Train				Test
Subject	Session 1	Session 2	Session 3	Session 4	Session 5
S1					
S2					
. . .					
S38					

Train on one subject and test with other subjects

	Train				Validation	Test
Subject	Session 1	Session 2	Session 3	Session 4	Session 5	
S1						S2 S3 . . . S38

Channel Regions



F – Frontal Region

C – Central Region

P – Parietal Region

O – Occipital Region

T – Temporal Region

Models Implementation

BN3 Model

Input & BN	32(channel) * 410(TimeSample)

↓

Conv1d(1,1) and ReLU	32 * 410
	16 * 410

↓

Conv1d(20,20) and ReLU	16 * 410
	16 * 20

↓ Flatten

Linear, Tanh and Dropout(0.8)	320
	128

↓

Linear, Tanh and Dropout(0.8)	128
	128

↓

Linear and Sigmoid	128
	1

BN3 + LSTM

Input & BN	32(channel) * 410(TimeSample)

Conv1d(1,1) and ReLU	32 * 410
	16 * 410

Conv1d(20,20) and ReLU	16 * 410
	16 * 20

LSTM(16, 128, 2, True, 0.2) Tanh	16 * 20
	128 * 2

Flatten

Linear, Tanh, Dropout(0.8) and Sigmoid	128 * 2
	1

Conv2d Layers Setting

No of Layers	Input Size	Kernel and Operation	Output Size	Activation Function
Layer1	1, 32, 410		1, 32, 410	BatchNorm
Layer2	1, 32, 410	(32, 1) * 128	128, 1, 410	Tanh
Layer3	128, 1, 410	(1, 5) * 64	64, 1, 82	BN, Tanh, Dropout
	128, 1, 410	(1, 10) * 64	64, 1, 41	BN, Tanh, Dropout
	128, 1, 410	(1, 15) * 64	64, 1, 27	BN, Tanh, Dropout
Layer4		Concat, Dropout	64, 1, 150	
Layer5	64, 1, 150	(1, 2) * 32	32, 1, 75	BN, Tanh, Dropout
	64, 1, 150	(1, 4) * 32	32, 1, 37	BN, Tanh, Dropout
	64, 1, 150	(1, 11) * 32	32, 1, 13	BN, Tanh, Dropout
Layer6		Concat, Dropout	32, 1, 125	
Layer7	32, 1, 125	(1, 2)	32, 1, 62	Maxpool, Tanh, Dropout
Layer8	32, 1, 60	FC: 1984	128	Flatten, Linear, Tanh, Dropout
Layer9	128		1	Linear, Sigmoid

Models Implementation

Number of parameters in different models		
BN3 Conv1D	BN3+LSTM	CNN Conv2D
63,053	550,849	539,875

Results and Discussion

Machine Learning Accuracy

Support vector machine(SVM)

	precision	recall	f1-score	support
-1.0	0.84	1.00	0.91	200
1.0	1.00	0.05	0.10	40
accuracy			0.84	240
macro avg	0.92	0.53	0.50	240
weighted avg	0.87	0.84	0.78	240

Linear Discriminant Analysis (LDA)

	precision	recall	f1-score	support
1.0	0.97	0.93	0.95	200
2.0	0.69	0.85	0.76	40
accuracy			0.91	240
macro avg	0.83	0.89	0.86	240
weighted avg	0.92	0.91	0.92	240

Results and Discussion

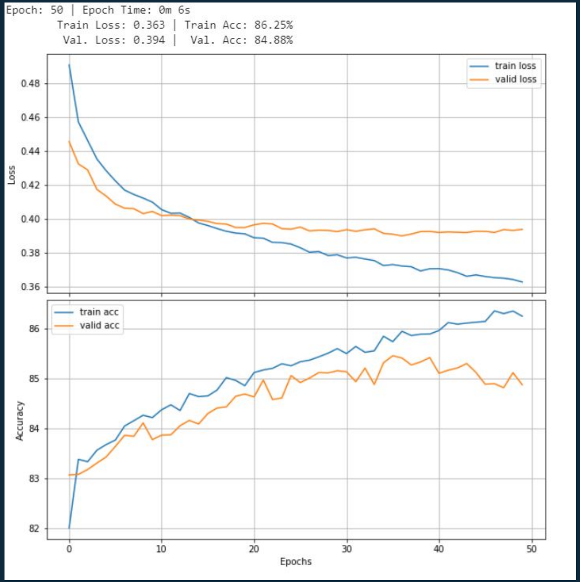
Subject Pool Accuracy and Loss

BN3	BN3 + LSTM	CNN Conv2D
84.54%	84.88	84.39

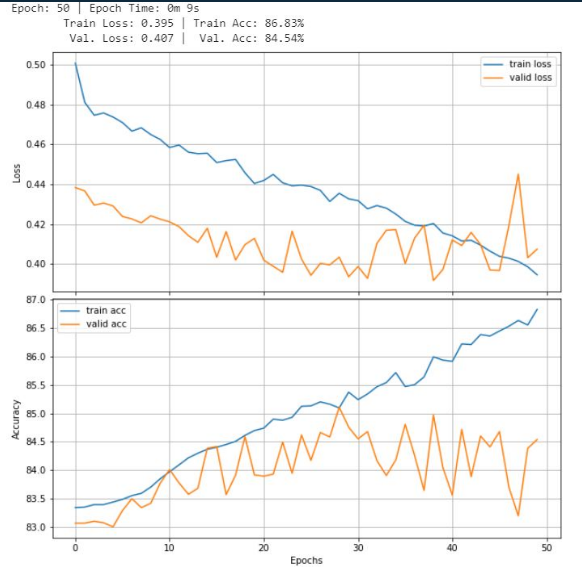
Results and Discussion

Loss and Accuracy Pooled Subjects

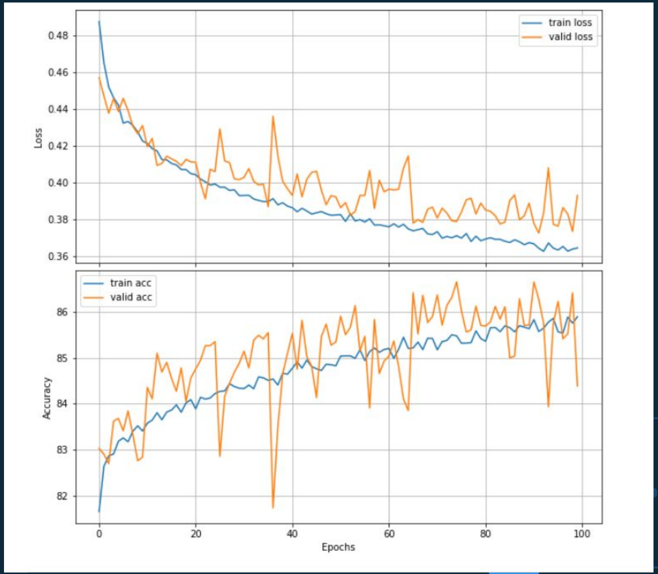
BN3 Conv1D



BN3+LSTM



CNN Conv2D



Results and Discussion

Deep Learning Accuracy

Subject Specific

- https://docs.google.com/spreadsheets/d/1RJUCVi3ZyuCSZcflDVqt_k3ZLTS_SmOdjE6z8Xe7_f8/edit#gid=1557508083

Results and Discussion

Deep Learning Accuracy

Test with another Subject

- https://docs.google.com/spreadsheets/d/1RJUCVi3ZyuCSZcflDVqt_k3ZLTS_SmOdjE6z8Xe7_f8/edit#gid=693610900

Results and Discussion

Deep Learning Accuracy

Regional Channel

https://docs.google.com/spreadsheets/d/1RJUCVi3ZyuCSZcflDVqt_k3ZLTS_SmOdjE6z8Xe7_f8/edit#gid=626629935

Conclusions

Subject Specific:

- CNN (Conv2D) generates the highest accuracy which is 92.46%, followed by BN3 with 89.99%, and BN3+LSTM model give the least accuracy at 86.2% accuracy.

- As for average accuracy of 38 subjects, BN3, BN3+LSTM and CNN (Conv2D) have 83.05, 81.31 and 80.80 respectively.

Conclusions

Test with another subject:

- CNN (Conv2D) generates the highest accuracy which is 91.35%, followed by BN3 with 89.93%, and BN3+LSTM model give the least accuracy at 84.22% accuracy.

- As for average accuracy of 38 subjects, BN3, BN3+LSTM and CNN (Conv2D) have 83.89, 82.89 and 82.07 respectively.

Conclusions

Channel Region:

- Compared to other channel regions' accuracy of every model, P region has the highest accuracy.

- BN3 generates the highest accuracy among 3 models.

- Highest Accuracy Region: Region P

Conclusions

We can conclude that for generalization purpose, we will choose BN3 as the best model among three of our models. Because, although all three models perform almost the same accuracy, BN3 is computationally efficient. It has total of 63,053 parameters to train which is nearly 10 times less than other two models

For subject specific model, CNN Conv2D outperforms other two models. Its accuracy reaches to 92.46 %

Files/ Folders Checklist

0. Visualize ERP
1. Load_data_pool.ipynb
2. Load_data
3. BN3 subject specific
4. BN3 train on one subject and test on another subject
5. BN3 regional channel
6. BN3+LSTM subject specific
7. BN3+LSTM train on one subject and test on another subject
8. BN3+LSTM regional channel
9. CNN subject specific
10. CNN train on one subject and test on another subject
11. CNN regional channel
12. BN3 regional channel P Region Test
13. BN3+LSTM regional channel P Region Test
14. CNN regional channel P Region Test
15. BN3 subject pool
16. BN3+LSTM subject pool
17. CNN subject pool
18. LDA train on one subject and test on another subject
19. SVM train on one subject and test on another subject

-- 'data' Folder contains '_epo.fif' files

Github Repository

https://github.com/aungzarlin1/EEG_Project

Future Work

- Use different preprocessing steps including ICA and check if we get higher accuracy from our models.
- Improve our 3 deep learning models using different and advance techniques.
- Construct a great classification model using deep learning knowledge that can contribute in BCI applications.

Thank You