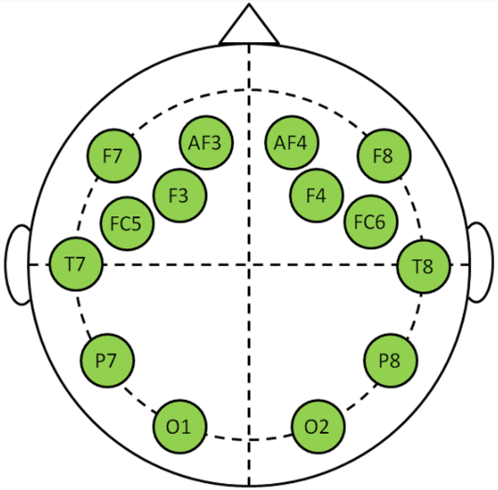
EEG-Based BCI Cursor Control

An application with Convolutional Neural Networks and Recurrent Neural Networks

Haoqi WANG, Jing WU
Mentor: Dr. Xiaopeng ZHAO, Soheil BORHANI

Introduction



BCI

Brain–computer interface (BCI) systems are allowing humans and non-human primates to drive prosthetic devices such as computer cursors and artificial arms with just their thoughts.
Invasive BCI systems acquire neural signals with intracranial or subdural electrodes, while **noninvasive** BCI systems typically acquire neural signals with scalp electroencephalography (EEG)

EEG

EEG refers to the recording of the brain's spontaneous electrical activity over a period of time, as recorded from multiple electrodes placed on the scalp.

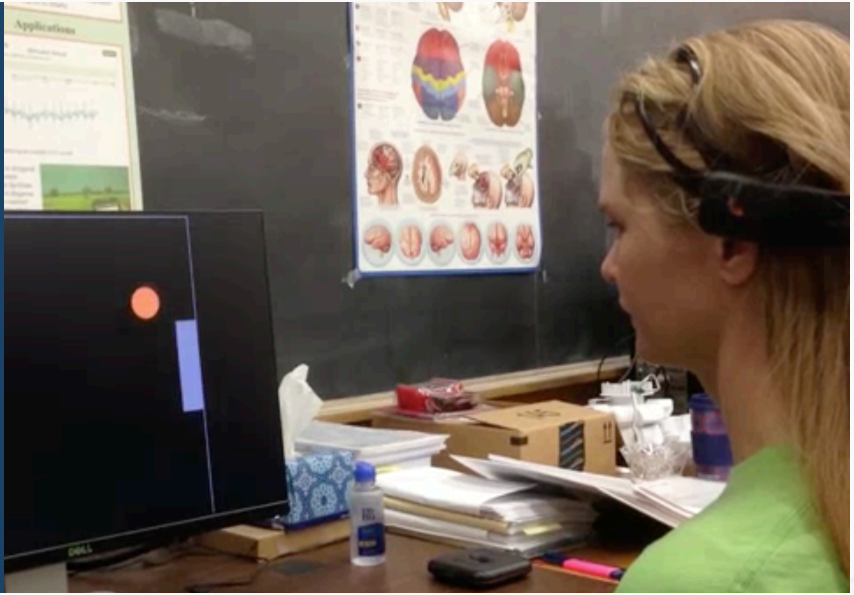
Related Study

In previous study, a decoder model of Multiple Linear Regression was used to predict the velocity of the computer cursor from EEG.

Experiment Setup

The computer cursor was programmed to move in one dimension(Horizontal/Vertical). The subjects were asked to track the moving cursor on the screen by imaging that they were using their dominant hand to control the cursor moving the same way as it was on the screen.

In the mean time, the EEG signal is recorded wirelessly with Emotiv EPOC headset of 14 channels. The headset with hydrated electrodes was put on the scalp of the subject. Meanwhile, TestBench, a Emotiv software was used to ensure the signal quality during the recording process. 14 channel EEG data and cursor movement were recorded simultaneously at a sample rate of 128Hz.



Data

34 subjects; 2 orientation/subject; 5 trials/orientation;

Cursor Movement

Measured by a vector

Magnitude: RNN regression

Orientation: CNN classification

TASK A
REGRESSION

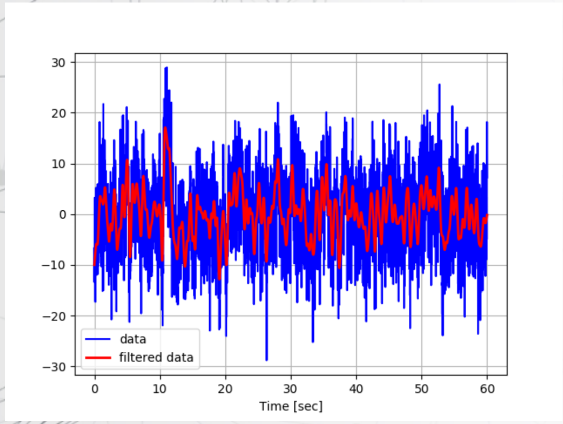
PIPELINE

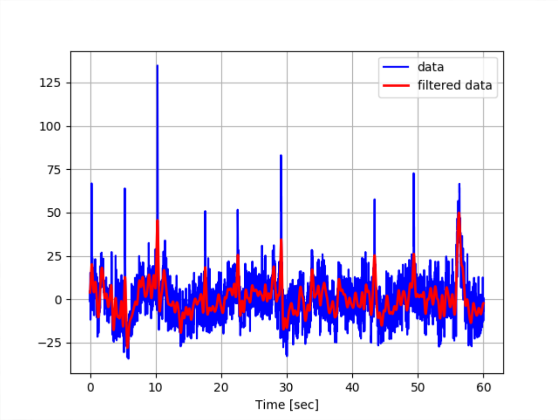
- FILTERING → Butterworth Low Pass Filter
- RNN → LSTM & GRU
- EVALUATION → Correlation & R squared Goodness of Fit

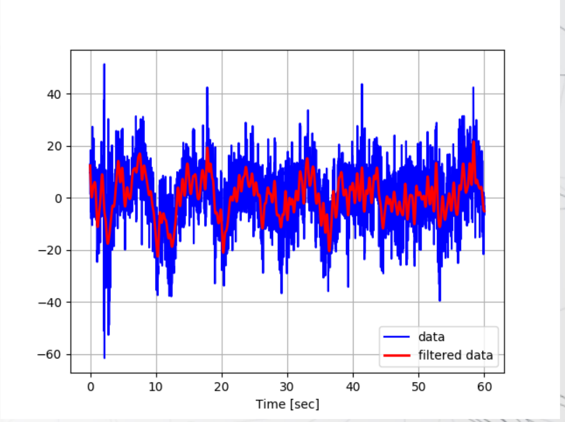
FILTERING

The Butterworth filter is a type of signal processing filter designed to have a frequency response as flat as possible in the passband. The main purpose of applying a low pass filter is to reduce the high frequency fluctuations for more smooth input signal of the neural network. According to some literature, the EEG signal under 2Hz has a close relation with the cursor control.

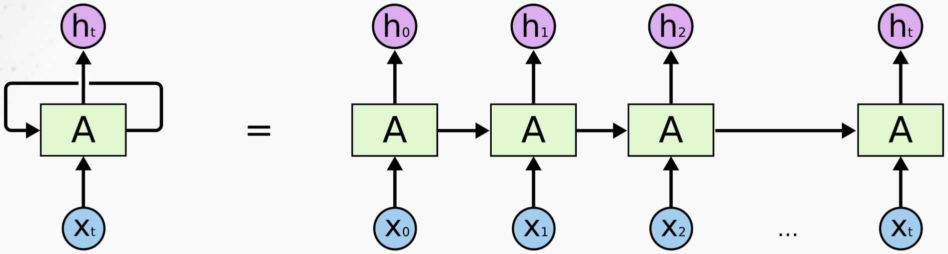
```
# Filter arguments.
order = 4
fs = 128        # sample rate, Hz
cutoff = 2  # cutoff frequency of the filter, Hz
T = 60.0         # seconds
n = int(T * fs) # total number of samples
```

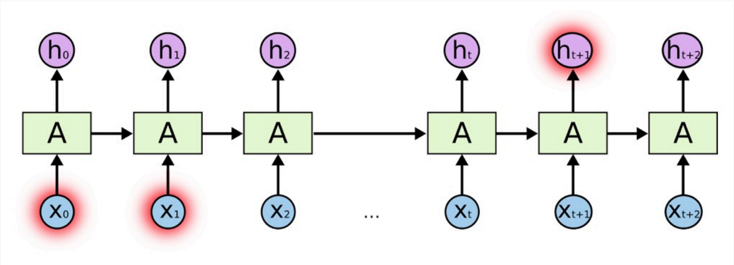






RNN

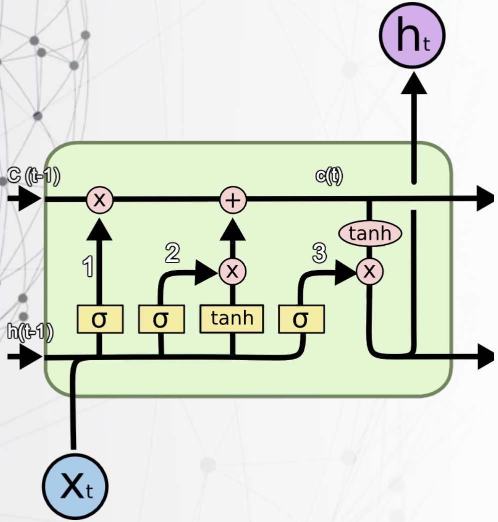


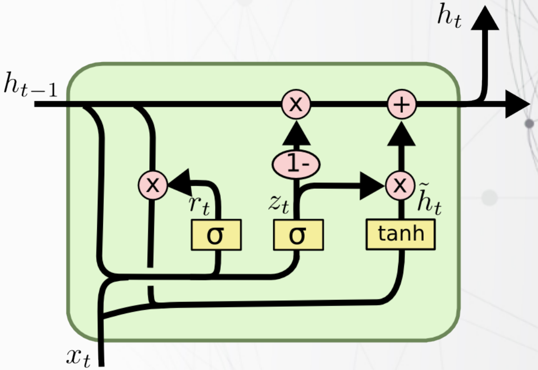


A **recurrent neural network** (RNN) is a class of artificial neural network where connections between nodes form a directed graph along a sequence. This allows it to exhibit dynamic temporal behavior for a time sequence. Unlike feedforward neural networks, RNNs can use their internal state (memory) to process sequences of inputs. This makes them applicable to tasks in this experiment since our input data is a sequence of time.

LSTM vs GRU

The LSTM model is composed of LSTM units which have cells, input gates, output gates and forget gates. The GRU consists of update gates and reset gates. In this project, they are of similar performance result.



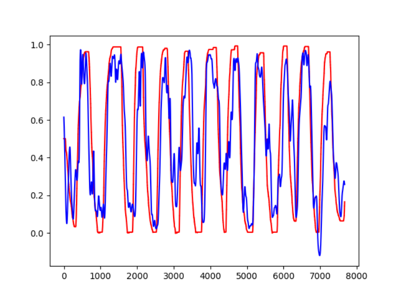


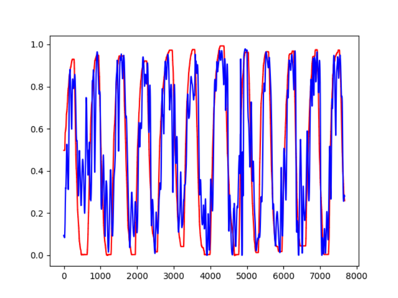
Cross Validation

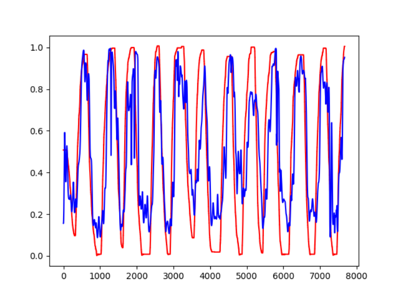
Leave One Out cross validation method is used to prevent the model over fitting problem. Since there are five trails for each regression model, one trail was left out for validation while the other four trails were for training. Therefore, five models are generated according to five different training and testing combinations.

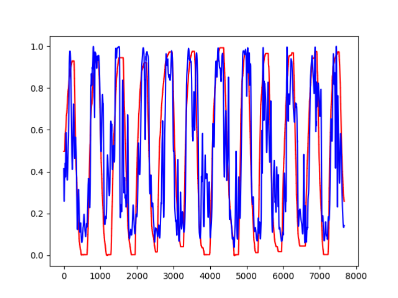
Result

Horizontal

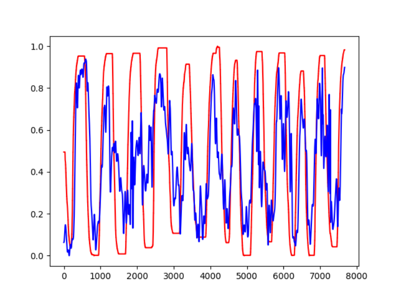


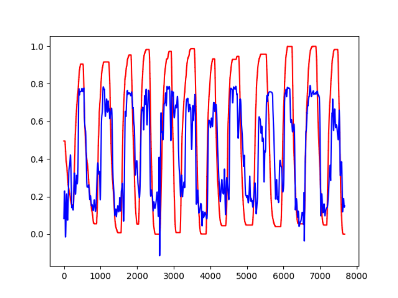


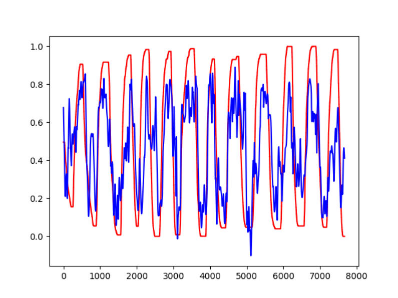


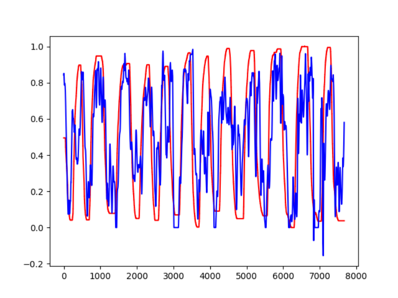


Vertical









Evaluation

The models were evaluated using two types of score called Goodness-of-Fit (GoF). This scoring technique separated the trial into segments of 5 seconds. The first type is to average the Pearson correla- tion scores between the predicted and ac- tual cursor velocities. Then, the averaged value of the Pearson correlation scores over each trial was defined as the GoF. The second type is to average the R squared value scores between the predicted and actual cursor velocities. These methods can provide a better representation of fit by not allowing one improperly fit window to reduce the overall models score.

Correlation Goodness of Fit Score:

$$GoF_1 = \frac{1}{M}\sum_{i=1}^{M} Corr(V^i_{decoded}, V^i_{observed})$$

R squared Goodness of Fit Score:

$$GoF_2 = \frac{1}{M}\sum_{i=1}^{M} R^2(V^i_{decoded}, V^i_{observed})$$

	Run1	Run2	Run3	Run4	Run5
subject1	0.50766366	0.64512496	0.61166563	0.58728453	0.69603025
subject2	0.08507769	0.2638802	-0.113182	-0.3842724	-0.1433992
subject3	0.34572155	0.2148576	0.07895965	0.20209551	0.40207659
subject4	0.49666897	0.52799297	0.45608423	0.49052988	0.50824087
subject5	0.46289468	0.18884032	0.71573085	0.1566184	0.27271921
subject6	0.66520578	0.71182724	0.73367876	0.69218537	0.84254325
subject7	0.75455727	0.64478183	0.78563624	0.79321395	0.81588211
subject8	0.60086354	0.56205967	0.6747443	0.72968948	0.63696507
subject9	0.58435528	0.58255071	0.62942517	0.47261191	0.7585664
subject10	0.35248094	0.19871911	0.17434969	0.35792352	0.49773695
subject11	0.1389093	0.00234993	0.15135268	0.5617493	0.62065569
subject12	0.41188501	0.45120455	0.481525	0.50031873	0.70758665
subject13	0.66366788	0.69452078	0.7439891	0.70459441	0.67387979
subject14	0.57211102	0.62093893	0.52665561	0.64229876	0.58034485
subject15	0.24171167	-0.0415362	0.20312596	0.03093479	0.23648607
subject16	0.03098635	-0.0717127	-0.0081039	-0.0329113	-0.1108109
subject17	0.45581892	0.71802318	0.69034734	0.72862916	0.76213866
subject18	0.02220374	-0.0812381	-0.1121656	0.27003718	-0.1072684
subject19	0.46220968	0.41256495	0.48282179	0.23070115	0.04608949
subject20	-0.0173909	-0.0549314	-0.0475847	0.05991706	0.0274714
subject21	0.4797889	0.55080388	0.59067602	0.73404319	0.82445002
subject22	0.56237856	0.59571952	0.5692891	0.78121702	0.46910826
subject23	0.11787857	-0.0586053	-0.1314836	0.48557939	0.48398752
subject24	0.23005666	0.02109972	-0.1570638	-0.0517793	0.04306263
subject25	0.80454375	0.79007072	0.71276072	0.79860442	0.74019998
subject26	-0.1266298	-0.0247938	0.01070082	-0.118654	-0.0262573
subject27	0.49360704	0.31858663	0.54608756	0.71307273	0.53715279
subject28	0.78764647	0.48650974	0.71381918	0.67882497	0.63465288
subject29	0.13819644	0.37453564	0.69017207	0.08402667	0.46011937
subject30	0.41744113	0.43064542	0.65938577	0.74423582	0.57482018
subject31	0.65296312	0.70525362	0.67656091	0.73426094	0.73166589
subject32	0.47027693	0.09430701	0.55261385	0.69652091	0.71460414
subject33	0.58854287	0.55842894	0.52464844	0.0096614	0.22805586
subject34	0.60114025	0.37164907	0.25439529	0.43048829	0.61056891

LSTM Horizontal Correlation

Best score: 0.8425

LSTM Vertical Correlation

Best score: 0.7695

	Run1	Run2	Run3	Run4	Run5
subject1	0.18505074	-0.0423576	0.11134406	0.05963649	0.02652203
subject2	-0.0863451	0.04469344	0.17962893	-0.1504774	0.23380085
subject3	0.40883314	0.39351921	0.35512852	0.38725501	0.28276712
subject4	0.0824757	0.10637813	0.2466645	0.18794468	0.30051072
subject5	0.11117085	0.19377716	0.19437758	0.41560192	-0.0388267
subject6	0.31159573	0.22728567	0.35365928	0.40096705	0.34667411
subject7	0.45856755	0.51663731	0.69208078	0.59077011	0.35682059
subject8	0.02370695	0.02453002	0.02646202	0.10996054	0.13682753
subject9	0.41999514	0.34272215	0.26813713	0.21966507	0.3946416
subject10	-0.1598008	0.07971421	0.18060727	0.13038102	0.08624144
subject11	0.04286716	0.15530748	0.08196272	-0.031257	0.00376222
subject12	0.13029792	0.10851208	0.16139014	0.27754349	0.01737465
subject13	0.62464297	0.38673639	0.40891341	0.1076267	0.20106618
subject14	0.50368758	0.63718989	0.64013858	0.60374182	0.67260799
subject15	0.3716616	0.35619257	0.38961436	0.07886109	0.12769529
subject16	0.21852468	-0.1131208	-0.0773546	-0.0416004	-0.2473735
subject17	0.21774177	0.29169703	0.35853269	0.42912166	0.33976902
subject18	0.18576636	0.15763833	-0.0371485	-0.0288132	-0.1512133
subject19	0.70626452	0.41059004	0.43050469	0.52986638	0.54269504
subject20	0.24868696	0.08952156	0.25309592	0.15282968	-0.1211243
subject21	-0.0878669	-0.256299	0.1619595	0.20086652	0.3014165
subject22	0.45106672	0.16181494	0.52729463	0.48268159	0.64979381
subject23	-0.0200612	0.1212714	0.30441041	0.20990383	0.08568964
subject24	0.04583481	-0.0585906	0.09529967	0.12624308	-0.0009648
subject25	0.76948602	0.47686288	0.45013436	0.54320723	0.5878762
subject26	-0.0382265	0.1504996	-0.0821204	0.00381755	0.26143044
subject27	-0.1844682	-0.0469997	-0.1632362	-0.1063766	-0.0584439
subject28	0.43586497	0.52899981	0.44617159	0.50989444	0.57094737
subject29	0.15343968	-0.02846	0.00147637	0.07742407	0.14606397
subject30	0.40526485	0.15547495	0.21024939	0.21415538	0.15030492
subject31	0.35205935	0.50132093	0.1961242	0.27880546	0.48500748
subject32	0.17641475	0.16695224	0.01392484	0.16836725	0.08872442
subject33	0.31778521	0.08344425	0.18525871	0.52450593	0.44422351
subject34	0.37239137	0.2581854	0.41498174	0.23412601	0.34493018

Conclusion

From the tables in the appendix, it is shown that the LSTM and the GRU model have very similar performance and accuracy. In a random seeded machine learning process, the highest correlation mark of the LSTM horizontal regression is 0.8425. The correlation mark of the GRU horizontal regression can reach up to 0.8203. The best vertical regression of LSTM score is 0.7695. And the vertical regression of GRU model goes up to 0.7113. The table also reveals that the horizontal cursor movement prediction is generally better than the vertical cursor movement prediction for the same subject. It is worth noting that the prediction accuracy varies from subject to subject. For example, subject 25 performs very well in both horizontal and vertical LSTM trails with an average score of 0.7692 for horizontal trails and an average score of 0.5655 for vertical trails. Subject 21 performs well in the horizontal tests but poorly in the vertical tests. Subject 14 can perform equally well results on both types. However, there are subjects who have no significant patterns such as subject 16.

Conclusion

Conclusion

1. RNN is an effective method for EEG decoding.
2. LSTM and the GRU model have very similar performance and accuracy.
3. Horizontal velocity regression is generally better than vertical velocity regression
4. The prediction accuracy varies from subject to subject.

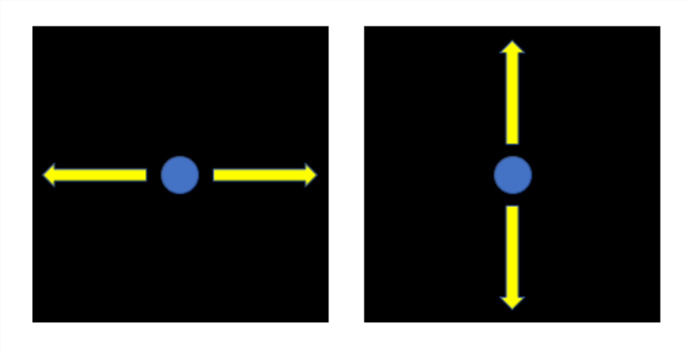
Task B
A Binary Classifier with CNN

OUTLINE

01 INTRODUCTION

- **Input data**:
  34 subjects, each subject practiced 10 trials. 5 trials of horizontal cursor movement, 5 trials of vertical cursor movement. Each trial last for 60s with a sampling rate at 128Hz.

- **Practical application**:
  Need to specify the orientation of the cursor movement in order to apply the corresponding regression model.



- **Binary classifier:**
  Act as a gate in front of the regression model to indicate if the user want to move the cursor horizontally or vertically

Why DNN ?

- **Deep Neural Networks(DNN)** has recently achieved remarkable accomplishment in several kinds of recognition tasks, including image, video, speech and text.

- Relatively **unexplored** in **neuroimaging** domain
  - DNN typically takes advantage of large data sets
  - Sample set of neuroimaging data is quite limited

- Some **previous work** in learning EEG representations using **CNN** and **RNN** with a **moderate data set** achieved remarkable result.
  - Piotr Mirowski et al. "Classification of patterns of EEG synchronization for seizure prediction" (2009)
  - Hubert Cecotti and Axel Graser. "Convolutional neural networks for P300 detection with application to brain-computer interfaces" (2011)
  - Nihal Fatma Güler, Elif Derya Übeyli, and Inan Güler. "Recurrent neural networks employing Lyapunov exponents for EEG signals classification" (2005)

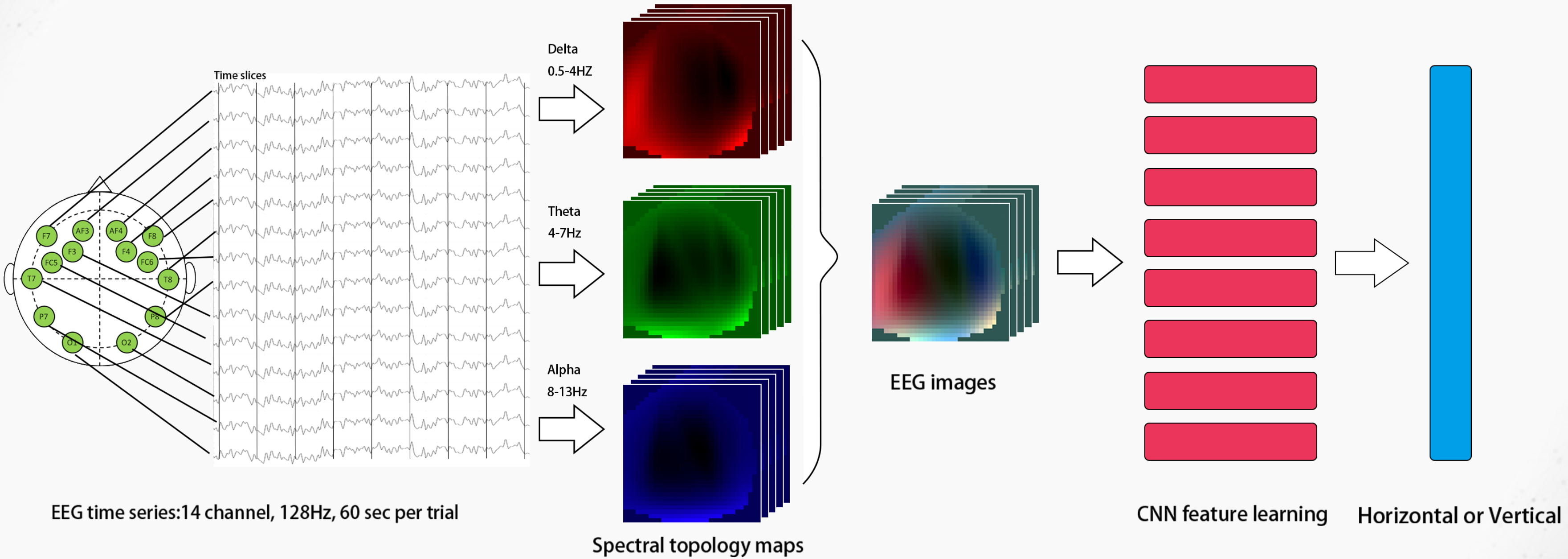
OBJECTIVES

- Build a classifier for recognizing users ’ intend of cursor movement orientation, and achieve a satisfying prediction accuracy with an acceptable model training time

- Investigate the potential influence of the input **window size**

- Investigate the variance between different experiment subjects

02 PIPELINE

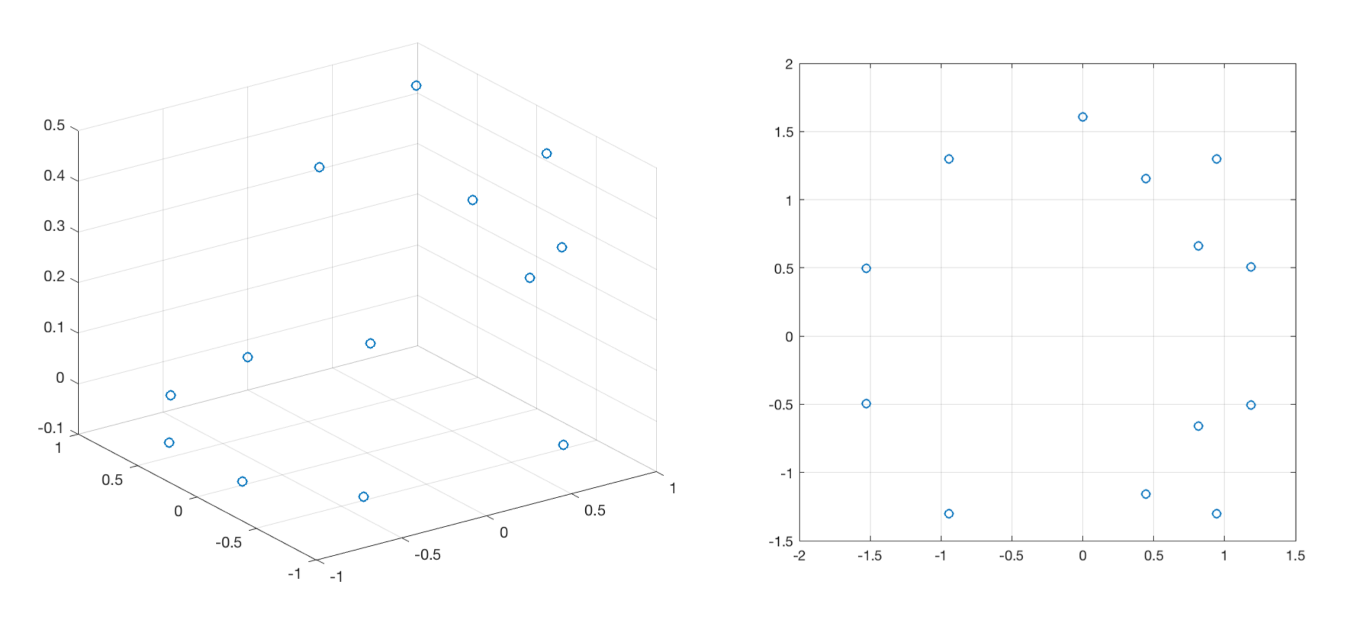


Time slices
F7
AF3
AF4
F8
F3
F4
FC5
FC6
T7
T8
P7
P8
O1
O2
Delta
0.5-4HZ
Theta
4-7Hz
Alpha
8-13Hz
EEG time series:14 channel, 128Hz, 60 sec per trial
Spectral topology maps
EEG images
CNN feature learning
Horizontal or Vertical

03 PREPROCESSING

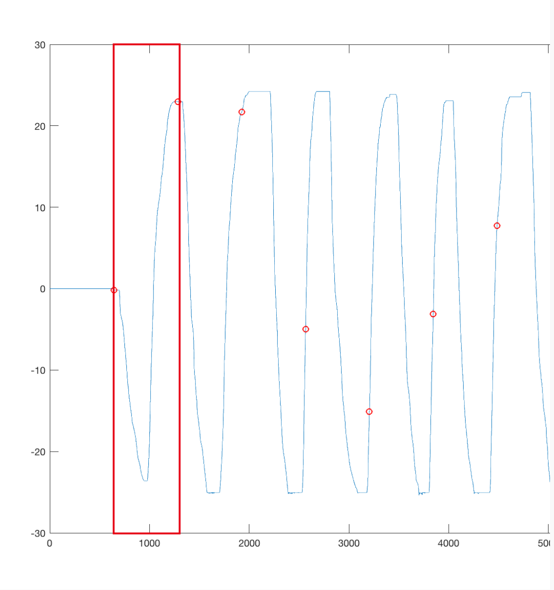
PROJECTION

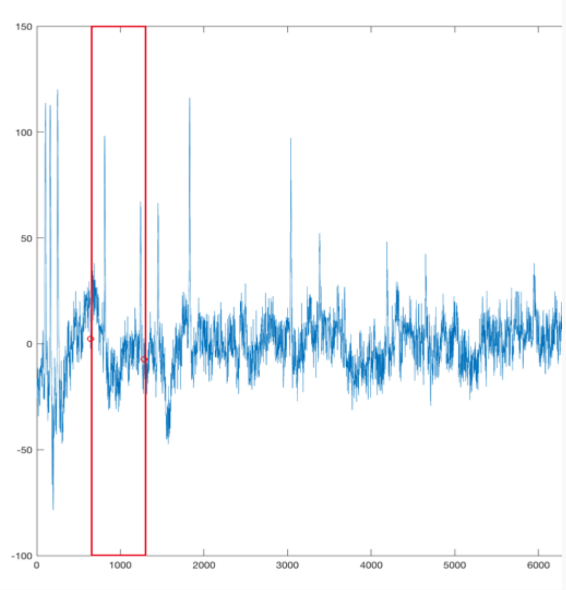
Project the 3-D location of the 14 electrodes to 2-D location using **Azimuthal Equidistant Projection**



FEATURE EXTRACTION

- For each trial of data, chop them into several **equal-sized** time slices
- Window size(length): **5sec**, **2.5sec**, **1.5sec**
- Calculate the **band-power** for each time slice under certain frequency ranges
- Delta(0.5~4Hz), Theta(4~7Hz), Alpha(8~13Hz)



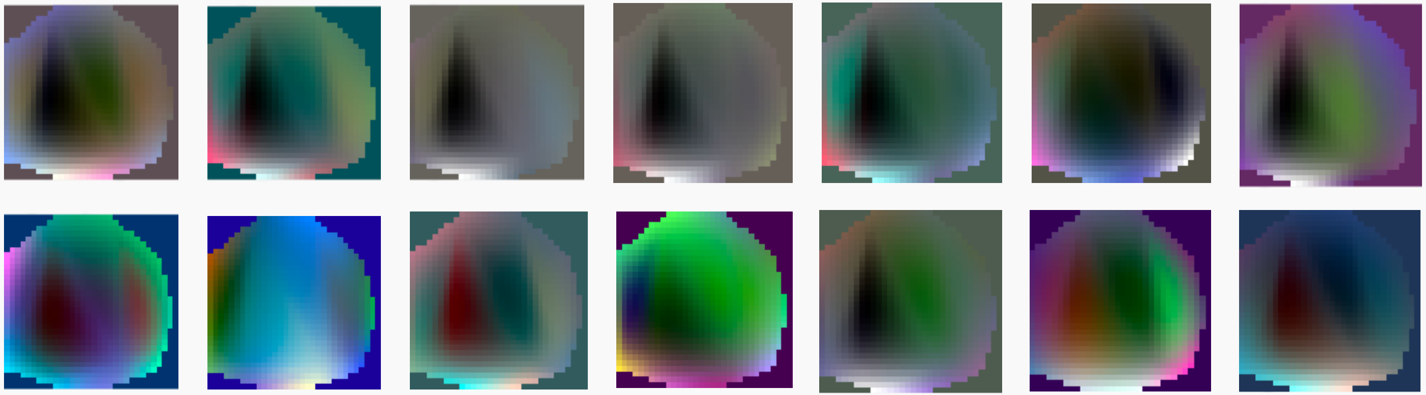


INTERPOLATION

- For each time window, apply **Clough-Tocher Scheme** to interpolate the scattered power with the 2-D location into three 32x32 spatial maps in R,G,B channels corresponding to three frequency ranges.
- **R:** Delta **G:** Theta **B:** Alpha

Horizontal

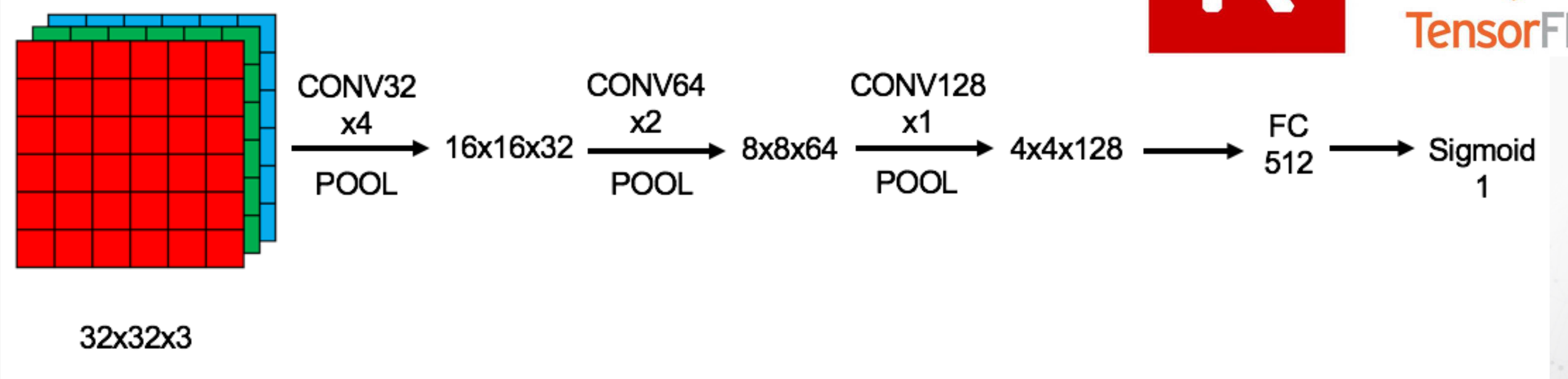
Vertical



First 40sec of subject 1

04 MODELLING

- **CONV**: 3x3 filter, stride=1 Activation Function: ReLU
  - "same" padding for the first two convolutional layer, "valid" padding for the rest of the convolutional layer.
  - **Batch Normalization**: accelerate the training by reducing internal covariance shift
- **MAX-POOLING**: 2x2 filter, stride=2
  - **Dropout** at a rate of 0.5: reduce overfitting by preventing complex co-adaptations on training data
- **Optimizer**: Adam



05 TRAINING & CROSS VALIDATION

TRAINING

- Input: $\frac{10\times 60}{window\ size}$ images. E.g. window size=5 sec, input size=120x3x32x32
- Train the model with different input window size to find out the optimal one among 5 sec, 2.5 sec, and 1.5 sec.
- Train the model with normalized input and unnormalized input.
- Train the model with all subject data using the optimal parameter setting

CROSS VALIDATION

- Leave-one-group-out cross-validation(LOGOCV)
- every trial was set to be one group
- Eliminate the internal connection the training set and validation set
- For each split, one group is chosen to be the validation set, and the rest groups are used as the training set.
- Totally C_{10}^{1} = 10 splits

06 RESULT & DISCUSSION

RESULT

	Window size		
Test set	5sec	2.5sec	1.5sec
Trial1H	58.33	62.50	40.00
Trial2H	83.33	79.17	52.50
Trial3H	75.00	70.83	77.50
Trial4H	66.67	83.33	45.00
Trial5H	83.33	79.17	75.00
Trial6V	100.00	100.00	100.00
Trial7V	100.00	100.00	100.00
Trial8V	100.00	4.17	70.00
Trial9V	91.67	100.00	87.50
Trial10V	25.00	25.00	40.00
MEAN	**78.33**	**70.42**	**68.75**
STD	0.2364	0.3242	0.2334

Subject2 unnormalized

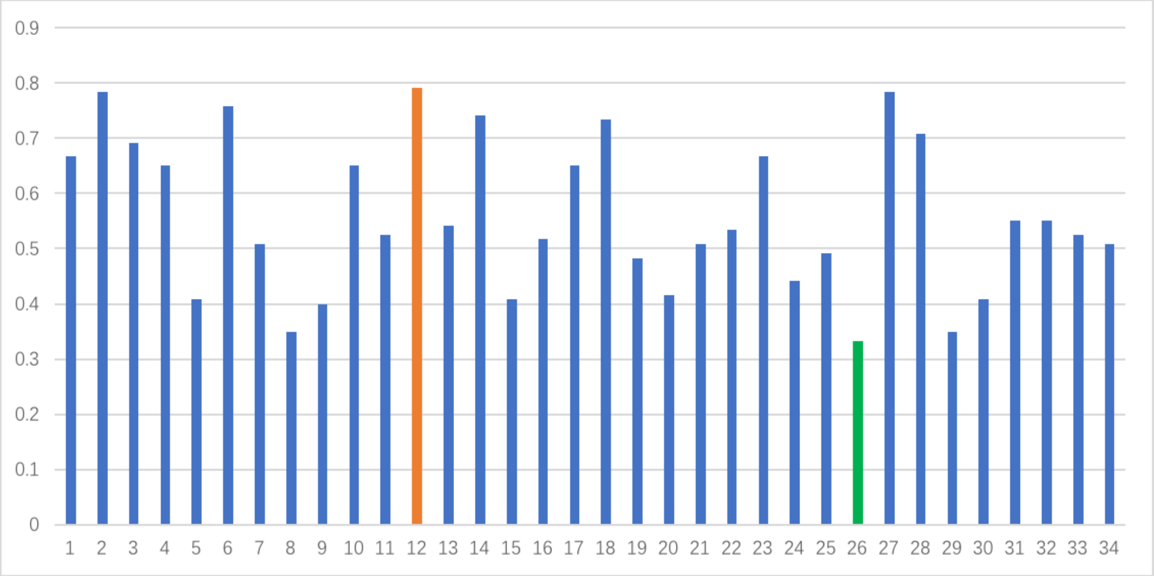
	Window size		
Test set	5sec	2.5sec	1.5sec
Trial1H	75.00	83.33	50.00
Trial2H	91.67	75.00	65.00
Trial3H	75.00	70.83	75.00
Trial4H	83.33	66.67	72.50
Trial5H	91.67	75.00	65.00
Trial6V	100.00	79.17	82.50
Trial7V	66.67	79.17	77.50
Trial8V	66.67	58.33	65.00
Trial9V	58.33	70.83	70.00
Trial10V	0.00	50.00	35.00
MEAN	**70.83**	**70.83**	**65.75**
STD	0.2812	0.1021	0.1400

Subject2 normalized

- Optimal window size: 5sec
- The model performs better without normalization

- Training time:
  - Multi-thread: 4'10" for each training set
  - Single thread: 10'50" for each training set
  - 2.8 GHz 4-core Intel i7 processor

- The best average prediction accuracy is **79.17%** given by subject12’ s data, with the maximum prediction accuracy at 100% using the second vertical trial as the test set.
- The worst performance is **33.3%** of average prediction accuracy given by subject26, although the maxi- mum prediction accuracy is also 100%.



S12	S26
66.67	25.00
75.00	0.00
75.00	91.67
33.33	100.00
75.00	0.00
91.67	0.00
100.00	41.67
91.67	75.00
91.67	0.00
91.67	0.00
79.17	**33.33**
0.1934	0.4120

DISCUSSION

- While the model has achieved a satisfying prediction accuracy at 79.17% with a acceptable training time, it is not robust to all subject.

- Possible reason
  - The parameter setting that is optimal for one subject is not that suitable for all other subject
  - The temporal property of EEG signals was not utilized

FUTURE WORK

- For every subject, try to find the optimal parameter setting, then find the optimal parameter setting for all subjects such that our model could be generalized for all kind of person.
- Try to implement RNN after each time-distributed CNN output, try to use algorithms like RCNN, LRCN and other video classification techniques.
- Implement the model on GPU to accelerate the training.

THANKS