DLR Internship Report

Thank you for this opportunity!

F-RELACS

- This project aims at creating an autonomous model to predict the frustration levels of drivers and/or passengers using the following data points:
  - Facial Video
  - Pose Video
  - EEG Data
  - ECG Data
  - Skin conductance levels
  - Eye tracking

- An extensive dataset is obtained using a dynamic driving simulator at DLR Braunschweig along with a subjective frustration rating from the subjects.

Project 1: Subjective Rating System

- Build an application to allow participants to efficiently rate their frustration levels on a scale of 1-10 (3 decimal digit accuracy) at a frequency of 50 Hz with a joystick.

- Build an application to automate the process of iterating (changing paths, names, saving, etc) through all cases for the participants to make the study easier for all parties.

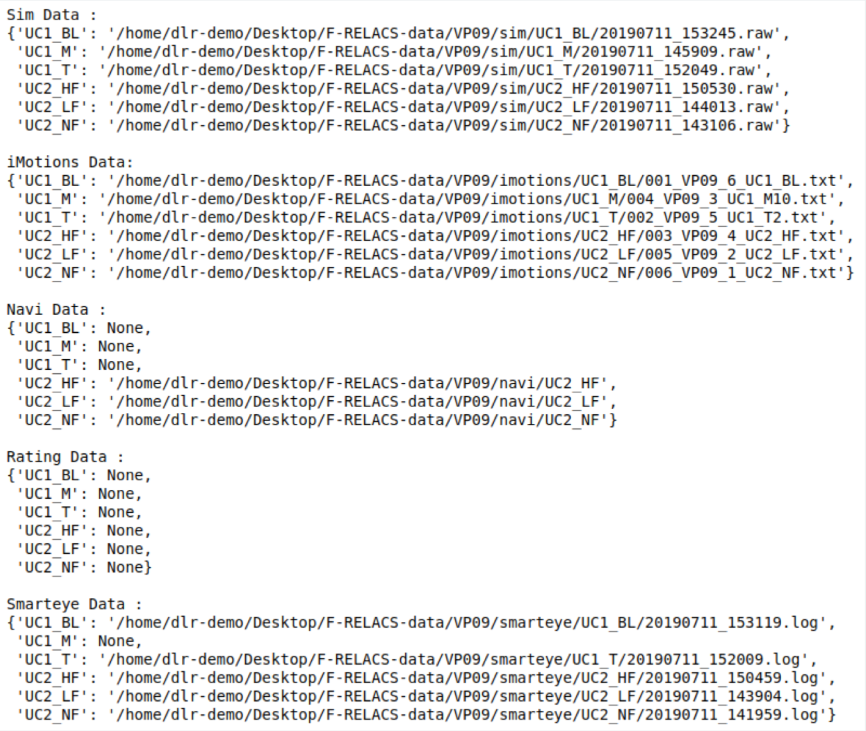
- Documented all code and instructions for future use.

Project 2: Sorting the data files

- Wrote a script to iterate through all files in the a given folder and give an easy to read report on missing and extra files to automate human checking.

- Wrote a script to iterate through all files (.csv, .raw, .txt, etc) and move them into their respective study folders to automate human time consuming grunt work.

- Documented all code and instructions for future use.



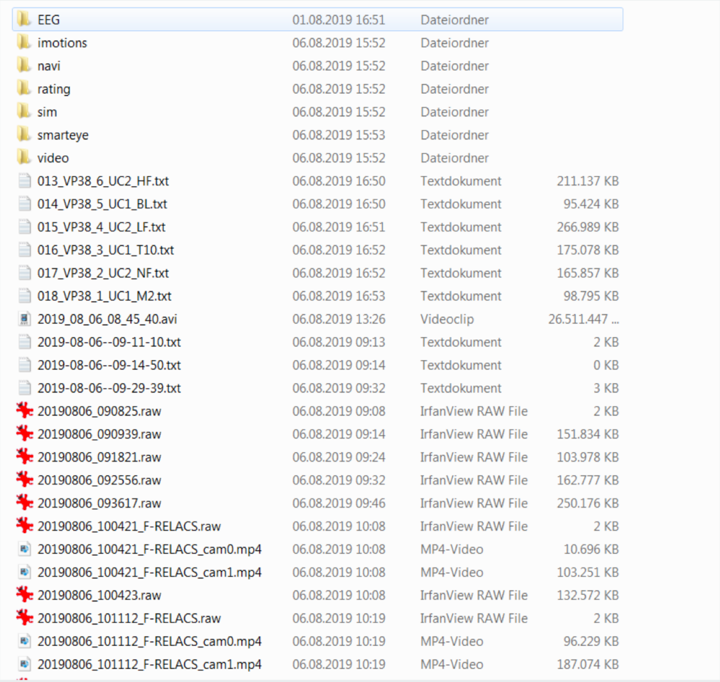
```
Sim Data :
{'UC1_BL': '/home/dlr-demo/Desktop/F-RELACS-data/VP09/sim/UC1_BL/20190711_153245.raw',
 'UC1_M': '/home/dlr-demo/Desktop/F-RELACS-data/VP09/sim/UC1_M/20190711_145909.raw',
 'UC1_T': '/home/dlr-demo/Desktop/F-RELACS-data/VP09/sim/UC1_T/20190711_152049.raw',
 'UC2_HF': '/home/dlr-demo/Desktop/F-RELACS-data/VP09/sim/UC2_HF/20190711_150530.raw',
 'UC2_LF': '/home/dlr-demo/Desktop/F-RELACS-data/VP09/sim/UC2_LF/20190711_144013.raw',
 'UC2_NF': '/home/dlr-demo/Desktop/F-RELACS-data/VP09/sim/UC2_NF/20190711_143106.raw'}

iMotions Data:
{'UC1_BL': '/home/dlr-demo/Desktop/F-RELACS-data/VP09/imotions/UC1_BL/001_VP09_6_UC1_BL.txt',
 'UC1_M': '/home/dlr-demo/Desktop/F-RELACS-data/VP09/imotions/UC1_M/004_VP09_3_UC1_M10.txt',
 'UC1_T': '/home/dlr-demo/Desktop/F-RELACS-data/VP09/imotions/UC1_T/002_VP09_5_UC1_T2.txt',
 'UC2_HF': '/home/dlr-demo/Desktop/F-RELACS-data/VP09/imotions/UC2_HF/003_VP09_4_UC2_HF.txt',
 'UC2_LF': '/home/dlr-demo/Desktop/F-RELACS-data/VP09/imotions/UC2_LF/005_VP09_2_UC2_LF.txt',
 'UC2_NF': '/home/dlr-demo/Desktop/F-RELACS-data/VP09/imotions/UC2_NF/006_VP09_1_UC2_NF.txt'}

Navi Data :
{'UC1_BL': None,
 'UC1_M': None,
 'UC1_T': None,
 'UC2_HF': '/home/dlr-demo/Desktop/F-RELACS-data/VP09/navi/UC2_HF',
 'UC2_LF': '/home/dlr-demo/Desktop/F-RELACS-data/VP09/navi/UC2_LF',
 'UC2_NF': '/home/dlr-demo/Desktop/F-RELACS-data/VP09/navi/UC2_NF'}

Rating Data :
{'UC1_BL': None,
 'UC1_M': None,
 'UC1_T': None,
 'UC2_HF': None,
 'UC2_LF': None,
 'UC2_NF': None}

Smarteye Data :
{'UC1_BL': '/home/dlr-demo/Desktop/F-RELACS-data/VP09/smarteye/UC1_BL/20190711_153119.log',
 'UC1_M': None,
 'UC1_T': '/home/dlr-demo/Desktop/F-RELACS-data/VP09/smarteye/UC1_T/20190711_152009.log',
 'UC2_HF': '/home/dlr-demo/Desktop/F-RELACS-data/VP09/smarteye/UC2_HF/20190711_150459.log',
 'UC2_LF': '/home/dlr-demo/Desktop/F-RELACS-data/VP09/smarteye/UC2_LF/20190711_143904.log',
 'UC2_NF': '/home/dlr-demo/Desktop/F-RELACS-data/VP09/smarteye/UC2_NF/20190711_141959.log'}
```

Project 2: Sorting the data files

- Wrote a script to iterate through all files in the a given folder and give an easy to read report on missing and extra files to automate human checking.

- Wrote a script to iterate through all files (.csv, .raw, .txt, etc) and move them into their respective study folders to automate human time consuming grunt work.

- Documented all code and instructions for future use.



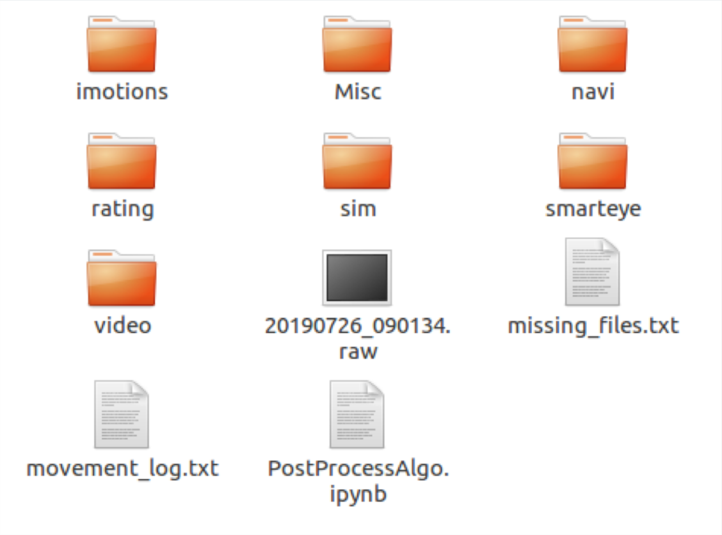
EEG	01.08.2019 16:51	Dateiordner	
imotions	06.08.2019 15:52	Dateiordner	
navi	06.08.2019 15:52	Dateiordner	
rating	06.08.2019 15:52	Dateiordner	
sim	06.08.2019 15:52	Dateiordner	
smarteye	06.08.2019 15:53	Dateiordner	
video	06.08.2019 15:52	Dateiordner	
013_VP38_6_UC2_HF.txt	06.08.2019 16:50	Textdokument	211.137 KB
014_VP38_5_UC1_BL.txt	06.08.2019 16:50	Textdokument	95.424 KB
015_VP38_4_UC2_LF.txt	06.08.2019 16:51	Textdokument	266.989 KB
016_VP38_3_UC1_T10.txt	06.08.2019 16:52	Textdokument	175.078 KB
017_VP38_2_UC2_NF.txt	06.08.2019 16:52	Textdokument	165.857 KB
018_VP38_1_UC1_M2.txt	06.08.2019 16:53	Textdokument	98.795 KB
2019_08_06_08_45_40.avi	06.08.2019 13:26	Videoclip	26.511.447 ...
2019-08-06--09-11-10.txt	06.08.2019 09:13	Textdokument	2 KB
2019-08-06--09-14-50.txt	06.08.2019 09:14	Textdokument	0 KB
2019-08-06--09-29-39.txt	06.08.2019 09:32	Textdokument	3 KB
20190806_090825.raw	06.08.2019 09:08	IrfanView RAW File	2 KB
20190806_090939.raw	06.08.2019 09:14	IrfanView RAW File	151.834 KB
20190806_091821.raw	06.08.2019 09:24	IrfanView RAW File	103.978 KB
20190806_092556.raw	06.08.2019 09:32	IrfanView RAW File	162.777 KB
20190806_093617.raw	06.08.2019 09:46	IrfanView RAW File	250.176 KB
20190806_100421_F-RELACS.raw	06.08.2019 10:08	IrfanView RAW File	2 KB
20190806_100421_F-RELACS_cam0.mp4	06.08.2019 10:08	MP4-Video	10.696 KB
20190806_100421_F-RELACS_cam1.mp4	06.08.2019 10:08	MP4-Video	103.251 KB
20190806_100423.raw	06.08.2019 10:08	IrfanView RAW File	132.572 KB
20190806_101112_F-RELACS.raw	06.08.2019 10:19	IrfanView RAW File	2 KB
20190806_101112_F-RELACS_cam0.mp4	06.08.2019 10:19	MP4-Video	96.229 KB
20190806_101112_F-RELACS_cam1.mp4	06.08.2019 10:19	MP4-Video	187.074 KB

Project 2: Sorting the data files

- Wrote a script to iterate through all files in the a given folder and give an easy to read report on missing and extra files to automate human checking.

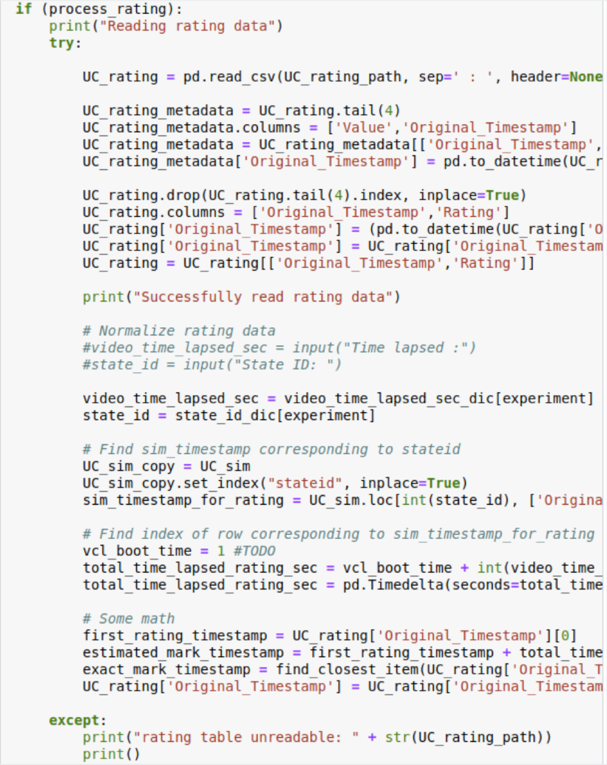
- Wrote a script to iterate through all files (.csv, .raw, .txt, etc) and move them into their respective study folders to automate human time consuming grunt work.

- Documented all code and instructions for future use.



Project 3: Post processing data

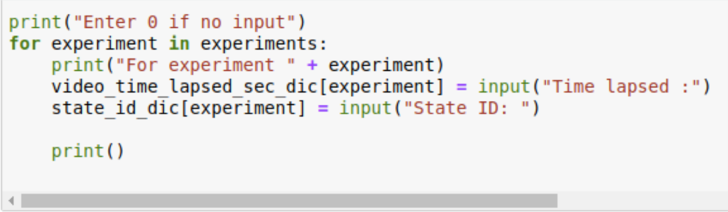
- Wrote a script to autonomously read all different file formats based on sorted folder and normalize different timestamps.
- Wrote a script to efficiently input state ID and time lapsed for each Rating-Video pair to correlate rating timestamps to all other data.
- Wrote and optimized (based on sampling frequency) a script to find closest data point for iMotions, Navigation, Rating and Smarteye based on timestamp for every master Sim timestamp.



```
if (process_rating):
    print("Reading rating data")
    try:

        UC_rating = pd.read_csv(UC_rating_path, sep=' : ', header=None

        UC_rating_metadata = UC_rating.tail(4)
        UC_rating_metadata.columns = ['Value','Original_Timestamp']
        UC_rating_metadata = UC_rating_metadata[['Original_Timestamp',
        UC_rating_metadata['Original_Timestamp'] = pd.to_datetime(UC_r

        UC_rating.drop(UC_rating.tail(4).index, inplace=True)
        UC_rating.columns = ['Original_Timestamp','Rating']
        UC_rating['Original_Timestamp'] = (pd.to_datetime(UC_rating['O
        UC_rating['Original_Timestamp'] = UC_rating['Original_Timestam
        UC_rating = UC_rating[['Original_Timestamp','Rating']]

        print("Successfully read rating data")

        # Normalize rating data
        #video_time_lapsed_sec = input("Time lapsed :")
        #state_id = input("State ID: ")

        video_time_lapsed_sec = video_time_lapsed_sec_dic[experiment]
        state_id = state_id_dic[experiment]

        # Find sim_timestamp corresponding to stateid
        UC_sim_copy = UC_sim
        UC_sim_copy.set_index("stateid", inplace=True)
        sim_timestamp_for_rating = UC_sim.loc[int(state_id), ['Origina

        # Find index of row corresponding to sim_timestamp_for_rating
        vcl_boot_time = 1 #TODO
        total_time_lapsed_rating_sec = vcl_boot_time + int(video_time_
        total_time_lapsed_rating_sec = pd.Timedelta(seconds=total_time

        # Some math
        first_rating_timestamp = UC_rating['Original_Timestamp'][0]
        estimated_mark_timestamp = first_rating_timestamp + total_time
        exact_mark_timestamp = find_closest_item(UC_rating['Original_T
        UC_rating['Original_Timestamp'] = UC_rating['Original_Timestam

    except:
        print("rating table unreadable: " + str(UC_rating_path))
        print()
```

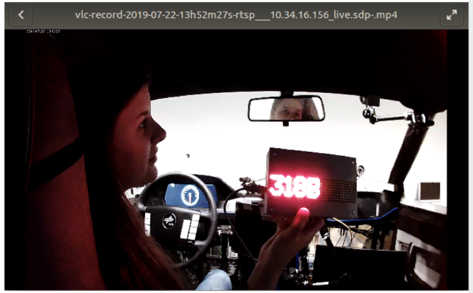
Project 3: Post processing data

- Wrote a script to autonomously read all different file formats based on sorted folder and normalize different timestamps.
- Wrote a script to efficiently input state ID and time lapsed for each Rating-Video pair to correlate rating timestamps to all other data.
- Wrote and optimized (based on sampling frequency) a script to find closest data point for iMotions, Navigation, Rating and Smarteye based on timestamp for every master Sim timestamp.



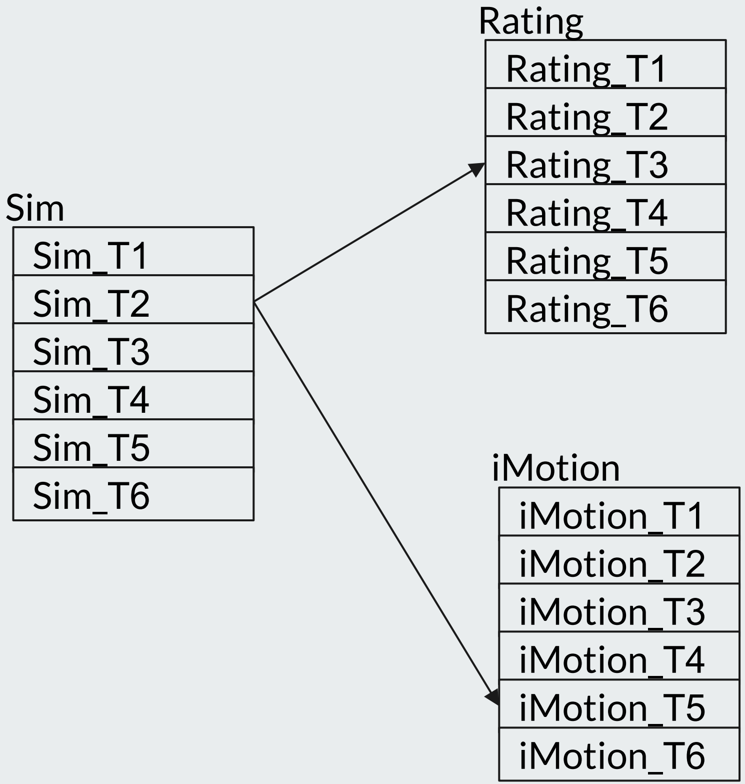
```python
print("Enter 0 if no input")
for experiment in experiments:
    print("For experiment " + experiment)
    video_time_lapsed_sec_dic[experiment] = input("Time lapsed :")
    state_id_dic[experiment] = input("State ID: ")

    print()
```

```
Enter 0 if no input
For experiment UC1_BL
Time lapsed :22
State ID: 1197

For experiment UC1_M
Time lapsed :19
State ID: 1578

For experiment UC1_T
Time lapsed :17
State ID: 768

For experiment UC2_HF
Time lapsed :3
State ID: 2385

For experiment UC2_LF
Time lapsed :17
State ID: 100

For experiment UC2_NF
Time lapsed :17
State ID: 683
```


vlc-record-2019-07-22-13h52m27s-rtsp___10.34.16.156_live.sdp-.mp4
3188

Project 3: Post processing data

- Wrote a script to autonomously read all different file formats based on sorted folder and normalize different timestamps.
- Wrote a script to efficiently input state ID and time lapsed for each Rating-Video pair to correlate rating timestamps to all other data.
- Wrote and optimized (based on sampling frequency) a script to find closest data point for iMotions, Navigation, Rating and Smarteye based on timestamp for every master Sim timestamp.



Sim

Sim_T1
Sim_T2
Sim_T3
Sim_T4
Sim_T5
Sim_T6

Rating

Rating_T1
Rating_T2
Rating_T3
Rating_T4
Rating_T5
Rating_T6

iMotion

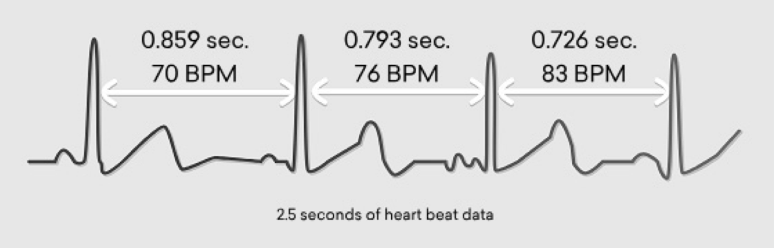
iMotion_T1
iMotion_T2
iMotion_T3
iMotion_T4
iMotion_T5
iMotion_T6

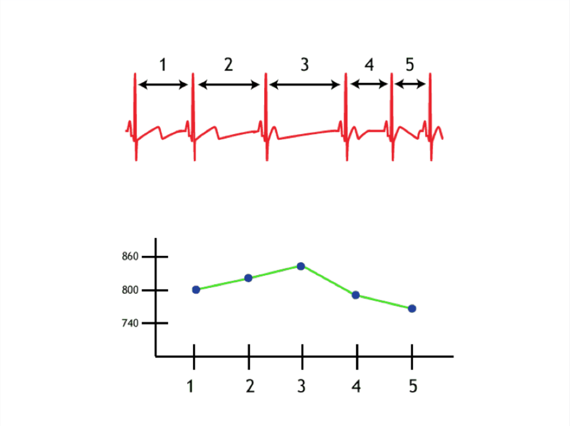
Project 4: Cleaning and Visualizing

- Wrote a script to clean Heart Rate data based on difference between two consecutive data points.

- Wrote a script to populate IBI data to ensure no Null values are picked up due to the nature of the data recording system.

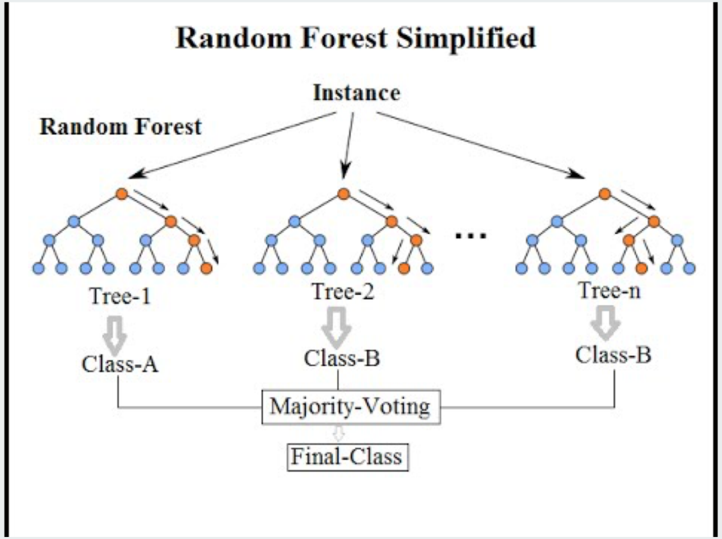
- Wrote a script to visualize any data point against frustration to get preliminary understanding of correlation.





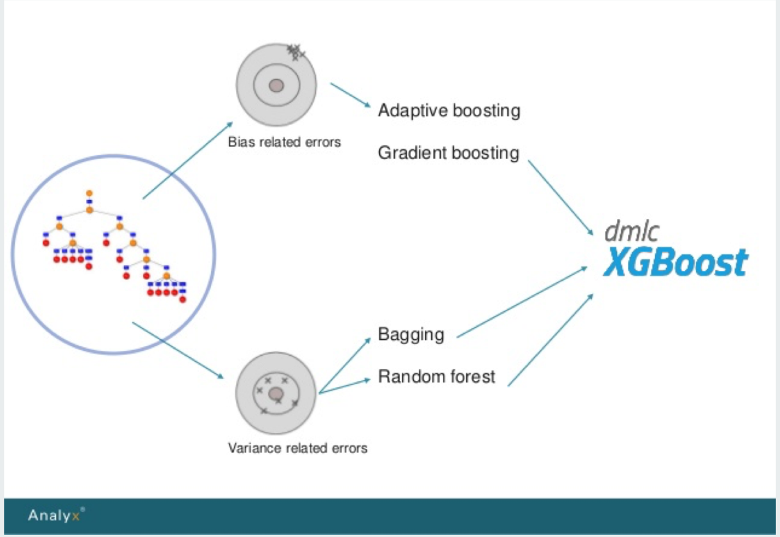
Project 5: Random Forest Classifier

- Wrote a script to train a basic RF model to predict the Rating on a 10 point scale based on the heart rate and GSR values at a particular timestamp.
- Wrote a script to randomly try a combination of parameters for the RF model and return the best combination.
- Wrote a script to brute force a number of RF models with parameters close to those found above to find best classifier
- Currently, due to not enough data the classifier is overfitting and resulting in 0.06 MSE.



Project 6: XGBoost Classifier

- Wrote a script to train a basic XGBoost model to predict the Rating on a 10 point scale based on the heart rate and GSR values at a particular timestamp.
- Wrote a script to randomly try a combination of parameters for the XGB model and return the best combination.
- Wrote a script to brute force a number of RF models with parameters close to those found above to find best classifier
- Currently, due to not enough data the classifier is overfitting and resulting in 0.04 MSE.

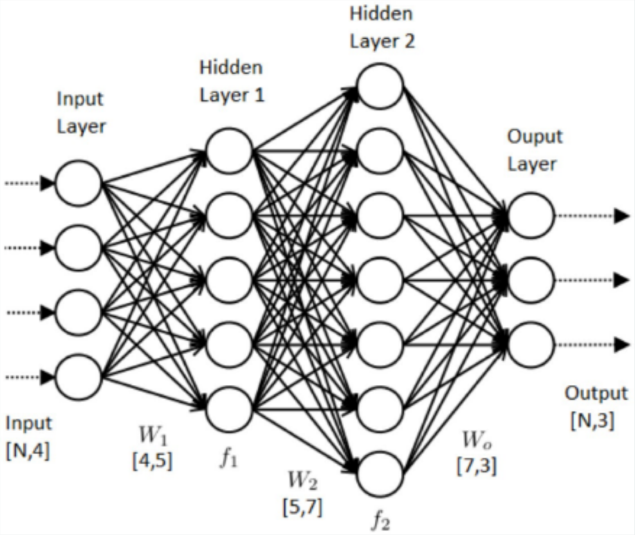


Project 7: Neural Net Model

- Wrote a script to train a basic Neural Net model to predict the Rating on a 10 point scale based on Action Units values at a particular timestamp.

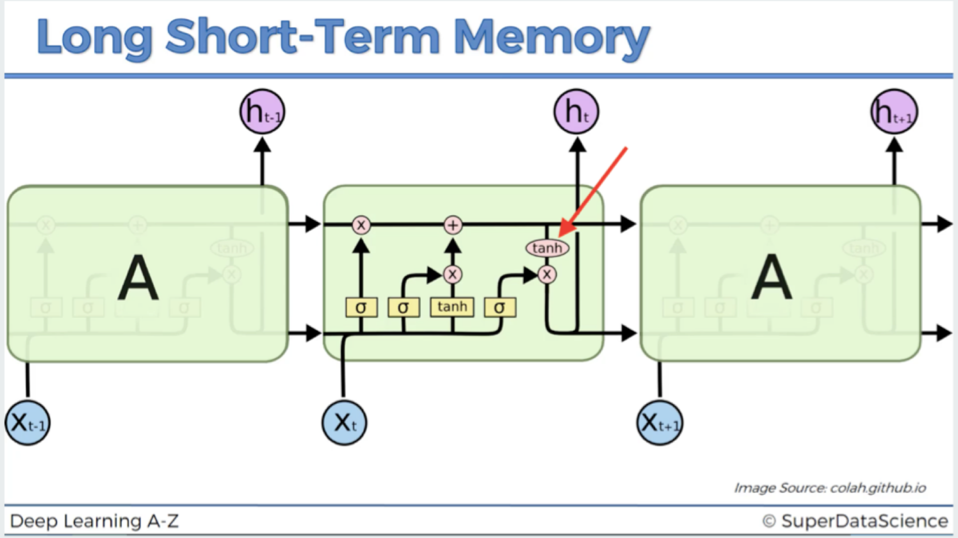
- Wrote a script to test following NN models variations randomly and return best parameters:
  - Hidden Layers: 1-4
  - Activation functions:
    ReLU, Linear, Gaussian, SoftPlus
  - Loss functions: MSE, MSLE, MAE

- Currently running.



Project 8: Long Short Term Memory RNN

- Wrote a script to train a basic LSTM model to predict the Rating on a 10 point scale based on Action Units values at a particular timestamp.

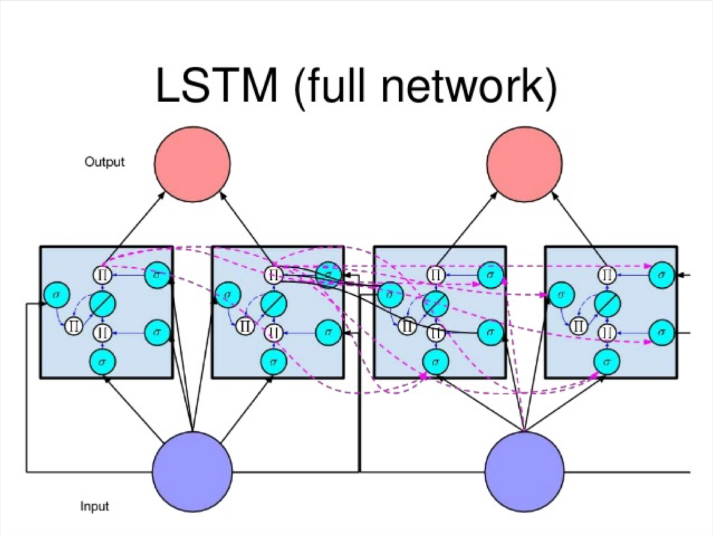


Long Short-Term Memory

Image Source: colah.github.io

Deep Learning A-Z

© SuperDataScience



LSTM (full network)

Project 8: Long Short Term Memory RNN

- Wrote a script to test following RNN models variations and return best parameters:
    - Hidden Layers : 1 or 2
    - Dropout Layer : 0.1-0.3
    - Activation functions : ReLU, Linear, TanH
    - Loss functions : MSE
    - Optimizer : adaptive moment estimation, short _Adam_

- Currently running.

Thank you

For the great time!