T1: XGBoost classifier

Design a XGBoost classifier for the EMG signal classification. An optimized implementation of Gradient Boosting is available in the popular Python Library XGBoost, which stands for Extreme Gradient Boosting. XGBoost' API is quite similar to Scikit-Learn's.

Useful Links:

1. https://xgboost.readthedocs.io/en/latest/
2. https://github.com/dmlc/xgboost
3. https://github.com/sumansamui/EMG_Signal_Classification

Steps to followed:

1. Load the Train and Test data from the repo. Don't need to use dev data for this task. Path → ../data/train and ../data/test
2. Scale the data
3. Apply XGBoost classifier
4. Fine tune the model using CV (k=5)
5. Generate the Results (6 sets) on the test data (set1 and set2)

Deadline for T1: 4th Sept, 2020

T2: Stacking Ensemble and Meta Learner

Run the individual classifiers (saved in the . . . /models/ folder) to make predictions on the dev set, and create a new training set with the resulting predictions. Each training instance is a vector containing the set of predictions from all the four classifiers for an instance in the dev set, and target is the ground truth label of the instance. Train a Random Forrest or MLP classifier on this new training set. This way, you have to create a Stacking ensemble or Meta Learner.

Steps to be followed:

1. Load the data from data folder of repository.

2. Scale the data

3. Make predictions of the dev data using the four models (LR, SVM, RF and ET) and create a new train set

4. Train a Random Forrest and MLP classifier individually on the new created train set

5. Evaluate the fine-tuned ensemble model on the two test sets

6. Compare the results with the original baseline models.

Deadline for T2: 8^{th} Sept, 2020