Assignment 3: Sensing and Machine Learning

Assigned: Mar 13 2024 **Due: 11:59 pm EST Mar 27 2024**

Last Updated: *03/12/2024, 5:00PM EST*

In this assignment, you will use the Qwiic accelerometer connected to your Particle Argon to detect gestures. The aim of this assignment is to learn: (1) how to do feature engineering by analyzing your sensor data, (2) tuning the parameters of classification algorithms, and (3) training, validating and testing your machine learning models. You will use the Argon to send data to a Python server, where you will featurize it, train it through a ML pipeline and then perform live inference. Starter code for the same is provided.

Starter code: HW3GestureClassifier.zip

Part 1: Gesture Detection using Accelerometer

In this part, you will build a gesture detector using the Qwiic accelerometer, and a Machine Learning model trained on a Python backend that we provide. Building on top of your Assignment A2 for gestures, you are free to choose any four of the alphanumeric characters (lower or uppercase).

There are two parts of the assignment:

- Make changes to the script *particle_server.py* adding your featurization code to the `featurize` function and trying different ML classifiers in `train_ml_classifier`. This will be used to showcase the demo of your system working.
- Collect 10 trials per gesture using server.py and save the data for analysis using *visualization.py* (or Jupyter - *visualize.ipynb*). Here you can test out your features and use the featurization code directly in server.py. Ideal goal is to make a pipeline that gets average of 90%+ accuracy in a Leave-One-Trial Out Cross Validation across gestures across 3 different classifiers (eg. SVC, KNeighborsClassifier, RandomForestClassifier). The reason we look at the average accuracy is to reduce overfitting. We have provided start code for calculating Accuracy and visualizing signals. The starter code flattens the raw values and uses them as features. In your final code, do not use the raw values(raw values will be prone to per-session overfitting) by themselves but rather extract features from them. The ideal goal is to use no more than 5 features.
- Which features work the best? Make a plot showcasing the feature ranking of the different features you used. Also, note how you determined which feature is helpful and calculated the feature ranking.

Part 2: Generalizability of your model.

How generalizable is your model? Apart from training/testing on your chosen four gestures in Part 1, we will ask you to train your model on one more alphabet gesture of our choice. You will not be informed of what gesture will be tested beforehand, until the time you do the demo. We evaluate if the features and algorithms you chose are able to handle a new class of gestures. During the demo, we will ask you to collect 10 training examples of this new gesture. You can just call this new gesture "bonus" in your data collection server.

Notes

1) Make sure to start your particle_server.py **before** the Argon runs setup(). You can do this by clicking the reset button on the Argon after the server is listening on the TCP socket.
2) The starter code includes a README.md. Make sure to follow the steps in order to set up your environment. It may be useful to set up a virtual environment to isolate your environment from existing package conflicts (see README.md > Notes).

Deliverables

- Python source code
- Particle source code
- Video capturing the phone screen while performing the gestures activity
- Write-up
    - Features used and your feature selection process
    - Feature importance plot of your final selected features
    - With your final selection of features: average accuracy across at least 3 ML model, accuracy of the ML model used in demo app
- Demo of the whole system during the office hours

Grade distribution

```
Part 1
   -  Accuracy (0.25% per accuracy point,
      (i.e., 70% accuracy = 35% grade): 40%
Part 2
   -  Instructor-selected gesture: 10%
   -  Particle implementation (since it's mostly starter code): 10%
   -  Python implementation and Feature Selection (-1% for each additional
      feature over 5, up to a maximum of -20%, i.e., if you have more than 25
      features, you only lose 20%): 25%
   -  Writeup (5% per sub-bullet point): 15%
=====================================================
Total Pts: 100%
=====================================================
```