Week 7, SW ML individual component test, Wang Zihao, Group 16, A0204706M

1. Human activities chosen for classification (8-class classification)
    a. Walking, jogging, typing, brushing, drinking soup, eating chips, writing, clapping.
    b. Two activities involving legs while six activities involving upper body movement only, which is similar to our actual dance classification.
2. Dataset (online)
    a. https://archive.ics.uci.edu/ml/datasets/WISDM+Smartphone+and+Smartwatch+Activity+and+Biometrics+Dataset+#
    b. One accelerometer and one gyroscope on the wrist, one accelerometer and one gyroscope on the waist. Each sensor will record data in three axes (X, Y and Z). In total 4 sensors (2 accelerometers + 2 gyroscopes) are used, which is similar to our design.
    c. 51 participants and 18 different activities
    d. Sampling rate at 20 Hz
3. Machine learning models selected
    a. Multilayer perceptron (MLP)
    b. Support Vector Machine with linear kernel (SVM)
    c. K-nearest Neighbor (kNN)
    d. All three models are compatible with the FPGA. Train and save the model, and save the weights, bias and the output of each hidden layer respectively and build the multilayer perceptron and kNN on FPGA using C++. Similar accuracy and halve the runtime.
4. Data segmentation
    a. 3s sliding window size with no overlapping
    b. I have compared the classification accuracy by varying sliding window size and it appears that 3s is the most appropriate as it yields the highest classification accuracy for the same model.
5. Feature selection
    a. Time domain feature: Absolute mean, Standard Deviation, Variance

    b. Frequency domain feature: Subbandpower, Peak power spectral density, Peak frequency, Spectral energy (Not used as it increases overfitting to a significant extent)
    c. 6 features × 4 sensors × 3 axis = 72 dimensions
6. Hyperparameters
    a. For MLP, 1 input layer of 72 nodes + 3 hidden layers of 70 nodes, 35 nodes and 10 nodes respectively + 1 output layer of 8 nodes. Initial learning rate is 0.01. Activation function for hidden layers is ReLU. Activation function for the output layer is Softmax.
    b. For SVM, the kernel used is linear kernel
    c. For kNN, k chosen is 3 (i.e., 3 nearest neighbour)
    d. Considerations: If the model is too simple, it leads to high bias error and low variance error, thus causing underfitting. If the model is too complex, it leads to high variance error and low bias error, thus causing overfitting.
7. Libraries and software:
    a. Python on Google Colab
    b. Numpy, Math, Itertools, Matplotlib, Seaborn, Scipy, Pandas, Time, Sklearn
8. Training classifiers:
    a. Splitting data to training set and test set (80-20) and normalise training set and test set separately to prevent data snooping.
    b. 5-fold cross validation
9. Model evaluation:
    a. Classification accuracy
    b. Confusion matrix (i.e., x-axis = predicted label and y-axis = actual label → try to get predicted label = actual label → diagonal matrix)
    c. Classification report: Precision (i.e., when it predicts yes, how often is it correct), Recall (i.e., true positive rate), F1-score (weighted average of the precision and recall)

       Precision = $\frac{tp}{tp + fp}$

       Recall = $\frac{tp}{tp + fn}$

F1 = $\frac{2}{recall^{-1} + precision^{-1}} = \frac{tp}{tp + \frac{1}{2}(fp+fn)}$

d. MLP and SVM are better than kNN. The runtime for kNN is expected to be long as it takes O(n) for testing, n = number of training data points. MLP is the best according to the performance matrix.

10. Position detection
    a. Not using ML but using the raw gyroscope data
    b. Gyroscope → direction of the turn
    c. Accelerometer → distance of the movement