Homework 3

Multilayer Perceptron

Deadline: 2021/12/22 23:30

There is no handwriting part in homework 3.

Grading Policy

In the implementation, you have to complete the blocks enclosed by "PLEASE WRITE HERE" comments. If your submitted python file cannot be successfully compiled due to the mis-modification outside of the blocks mentioned above, some points will be deducted. In the report, answers are expected to be stated with your observations followed by explanations. Writing in English or Chinese is acceptable.
Please upload your python file and report named **hw3_StudentID.py** and **hw3_StudentID.pdf** to the eeclass. No late submission will be accepted.

Implementation

1. (5%) Write the information about this script in the comments.

2. (5%) The given file “data.csv” consists of 20621 samples, each with a 24-dimensional feature set and a categorical label. Please complete the function *load_data()* to load “data.csv” and return the corresponding x and y.

3. Please complete the function *preprocessing()* and return *x_train, x_val, x_test, y_train, y_val, y_test*.
    a. (5%) [Split Dataset] In order to perform “model selection” in later problems, we will first split the dataset into training set, validation set, and testing set with the portions of [0.8, 0.1, 0.1], respectively. Note that in order to observe certain phenomena, please assign the random state with 1 in the function.
    b. (5%) [Feature Scaling] Please perform feature scaling on each dataset.

4. Please complete the function *train()* and return *clf, train_loss, train_acc, val_loss, val_acc*. Note that in order to observe certain phenomena, please do not modify the random state assigned in *MLPClassifier()*.
    a. (5%) [Training] Please train the MLP classifier with the given number of the hidden layers and epochs.
    b. (15%) [Evaluation at each epoch] Please calculate and store the loss and accuracy of the training set and the validation set, respectively, at each epoch. Note that you have to calculate the cross-entropy loss for *train_loss* and *val_loss*.
    Hint: *sklearn.metrics.log_loss*.

5. (10%) Please complete the function *evaluation()* and return *recalls, acc, uar, cf_matrix*. In order to calculate scores, the prediction must be generated in the first place. Note that *recalls* should be a 5-element array, referring to the recall of each class.
    Hint: *sklearn.metrics.recall_score, sklearn.metrics.confusion_matrix*.

Report

1. Please train a MLP classifier with *hidden_layer_sizes=(16, 16, 16)* and *n_epochs=100*.
    a. (5%) Report the plot of log loss curves. How do the curves behave when the number of epochs increases? What is the difference between both curves and why?
    b. (5%) Report the plot of accuracy curves. How do the curves behave when the number of epochs increases? What is the difference between both curves and why?
    c. (5%) Report the scores of recalls, accuracy, and UAR. What do you observe and why? Please analyze it with the number of samples in each class. Hint: Imbalanced data.
    d. (5%) Report the confusion matrix. What do you observe and why?

2. Please train 9 MLP classifiers according to the combination of the following hyperparameters.
    - *hidden_layer_sizes*
        - *(16, 16, 16)*
        - *(16, 16, 16, 16)*
        - *(16, 16, 16, 16, 16)*
    - *n_epochs*
        - *100*
        - *200*
        - *400*

    a. (10%) Report the process times of training, as well as losses, accuracies, and UAR on the validation set, of these 9 classifiers, respectively. Please choose a best performing classifier and a worst one. On which criteria do you choose them? What and why result in these two best and worst classifiers?
    b. (10%) Do any classifiers overfit during the training? If yes, please list the classifiers and report their plots of log loss curves and plots of accuracy curves. On which criteria do you assume these models are overfitted?

3. (10%) According to problem 2. (a), please evaluate the best performing classifier on the testing set. Report the corresponding recalls, accuracy, and UAR. Are these scores as competitive as those evaluated on the validation set? Why do we have to do such comparisons to find the best performing classifier?