Assigmnent Requirements

Students are required to evaluate 2 algorithms on 8-10 datasets.

You must choose 1 of the following algorithms (provided):

1. Ensemble Deep Random Vector Functional Link
2. Kernel Ridge Regression
3. Random Forests

You are free to choose any algorithm that you wish as the second algorithm.
You are required to source for the codes yourself. Some potential algorithms are:
RVFL (single hidden layer), ELM (single hidden layer), HELM (multilayer ELM),
Multi-layer backpropagation, Oblique random forest, ResNet, Self-normalising neural networks,
etc. Codes of these methods can be easily obtained online.

Note: Kernel ELM must not be used as it is identical to KRR.

For datasets, you can choose any 8-10 of the 28 UCI Datasets provided.

Students are recommended to use MATLAB. Other languages are permitted too.

Experiments

Students are required to evaluate the algorithm on the given training and testing dataset.

Model must be trained using the training dataset. (Never to use testing dataset during training.)
Testing dataset should be used to evaluate the performance (Accuracy) of the final trained model.

For each dataset, we use 80% for training and 20% for testing.

Datasets

For this assignment, there are 28 UCI Datasets.
In each dataset, the total number of sample instances is between 1000 and 5000.

Datasets can be found in the folder "MatDataset".
Each folder in "MatDataset" contains 1 dataset.
In each folder in "MatDataset", there are 2 ".mat" files.
These are the training set ("*Train.mat") and testing set (*"Test.mat").

The file structure looks like this:

```
MatDataset
|--abalone
|  |--abalone_Train.mat
|  |--abalone_Test.mat
|--bank
|  |--bank_Train.mat
|  |--bank_Test.mat
...
```

Each ".mat" file contains 2 variables: "Data" and "Label".

Hyperparameter Tuning

For tuning of Model Hyperparameters, we use 4-Fold (Stratified) Cross-Validation on the training set.
This means we create 4 pairs of training/validation subsets to perform evaluation of parameters.
Results (namely, validation accuracy) must be avaraged across all 4 folds. The model giving the best
accuracy is selected as the final trained model to test on the test data.

A part of the code should look like this:

```
cv_part = cvpartition(trainY.'KFold',4);  % Create indices for
training/validation subsets

...
for k = 1:4
    % Collect training/validation sets
    val_trainX = trainX(cv_part.training(k),:);
    val_trainY = trainY(cv_part.training(k),:);
    val_testX = trainX(cv_part.training(k),:);
    val_testY = trainY(cv_part.training(k),:);
    ...
end
...

Note:
Some (older) versions might not work with the above-mentioned codes.
If this happens, you can replace 4 lines of codes with the ones shown below:

val_trainX = trainX(training(cv_part,k),:);
val_trainY = trainY(training(cv_part,k),:);
val_testX = trainX(test(cv_part,k),:);
val_testY = trainY(test(cv_part,k),:);
```

You can search the internet for more details on cross-validation.
If you wish to look into MATLAB documentation, type "doc cvpartition" in your MATLAB command window
and via internet search online.

The final report submission must include the following:

1. Descriptions about the two algorithms used in the study. With the algorithmic descriptions, the corresponding code segments (just a small section) can be included.
2. Descriptions about the important parameters of the chosen algorithms and how they were

tuned
using the 4-fold cross validation method.

3. Tables of results can include test accuracy, average validation accuracy (and training accuracy per dataset).
4. Statistical testing can be conducted using Wilcoxon signed-rank test or t-test.
5. Conclusions can be made.

All reports should be typed and uploaded to the turnitin submission page in NTULearn. All reports must be
original. Turnitin will do a similarity check (comparing with millions of documents) and highlight identical texts.

Deadline: 16 April 2021