Coding Project (Refactored collaboration)

--

Issam H. Laradji - 81491145
Matthew Dirks - 71992119

1) Assigned Tasks

- **Issam Laradji:** ***matlearn_classification_CV.m***
- **Matthew Dirks:** ***matlearn_regression_CV.m***

2) Refactored Collaboration

This is a refactoring collaboration. Matt Dirks and I combined the cross-validation for classification and regression into ***matlearn_CV.m.***
It has a parameter option **'loss'** which defines the loss function to be used,

- **'square error'** and **'absolute error'** for regression problems
- **'zero-one loss'** for classification problems

3) Four added values (First two are major, second two are minor) for the cross-validation algorithm:

• **Leave-one-out:** Boolean option, if true each sample is used once as a validation set (singleton) while the remaining samples make up the training set. This is equivalent to setting the number of folds to the number of samples in the dataset.

• **Early-stop:** Boolean option, if true: stops the grid search when the error starts increasing after it has decreased at least once.

• **Shuffle data:** Boolean option, if true CV will shuffle the dataset randomly before using it.

• **k-Fold cross-validation:** Integer option 'k' which sets the number of folds to perform.

4) Additional added value by Issam Laradji for Extreme Learning Machines (ELM):

- **Regularized ELM**: added a regularization term to the ELM model, implemented in ***matLearn_classification_ELM_Issam.m***, to achieve bias-variance tradeoff.

Running the demos

---

1) Issam Laradji individual demo

Run ***demo_CV_classification_ELM_issam.m*** in the zip file to execute cross-validation for choosing the best number of hidden neurons for ELM and display their results in a plot. The demo will also illustrate the effect of varying the regularization term for ELM on learning the decision model.

2) Matthew Dirks individual demo

Open ***html_demo_CV_regression_KNN_Dirks/demo.html*** which shows the demo code and explanations including screenshots (created via MATLAB publishing).
Run ***demo_CV_regression_KNN_Dirks.m*** to execute cross-validation for choosing the best value of ‘k’ for regression via KNN. The resulting error over the various values of k will be shown, along with a visualization of the predictions.