COMP2712 Heuristic Optimisation: Practical02

The overall task of this practical is to explore evaluation methods and some basic properties of a multi-layered perceptron (MLP) and to get you started in Heuristic Optimisation!

There is a starting Google Colaboratory Notebook here:

https://colab.research.google.com/drive/15Iw3m05wSe2yJvYaMSBoZW6nZYuUTS2R?usp=sharing

You should open this notebook in your Colab and save a copy to work on for the practical.

Checkpoint 1: Discover the "optimal" settings for the MLP

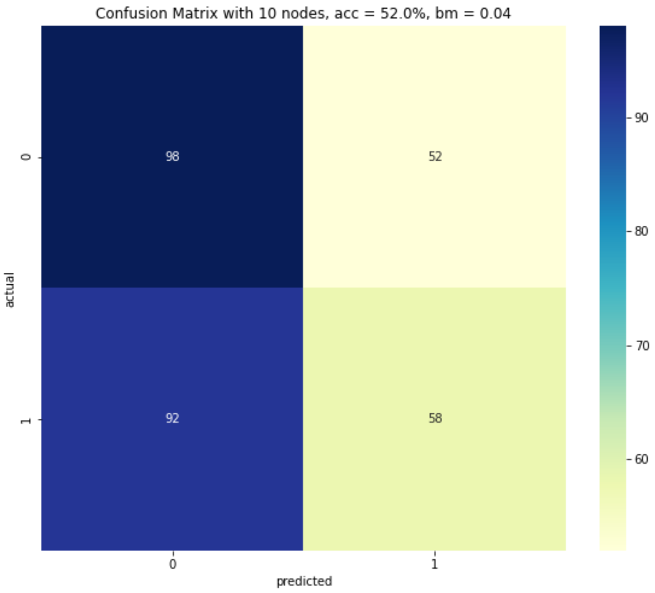
The current MLP definition results in poor performance. Adjust the hidden layer nodes (`hidden_layer_size`) and the number of training iterations (`max_iter`) and discover which combination gives an "optimal" performance. By "optimal" we want to find a network that performs well, but with the minimum number of hidden layer nodes and training iterations.

To gain this checkpoint you should present the results from at least 5 settings of `hidden_layer_size` with a fixed `max_iter` and five settings of number of training iterations `max_iter` with a fixed `hidden_layer_size` (this should give 10 results). You can then explore different combinations of parameters. You should set out your results in a table, for example

hidden_layer_size	max_iter	accuracy	precision	recall	f-score	bookmaker
1	50	50%	0.25	0.5	0.33	0
10	50	52%	0.52	0.52	0.51	0.04

Note that the results in this table are barely above guessing (e.g., an accuracy of 50% for a two-class problem, a bookmaker of 0) and you should be aiming to be closer to an accuracy of 100% (1) and a bookmaker of 1.

For at least your best result you should also capture the confusion matrix, for example



Show the modified notebook and explain to the demonstrator your results.

Checkpoint 2: Convert to Stratified K-Fold Evaluation (5%)

The current implementation in the notebook uses the simple train-test split. However, the methodology does not utilise the full data for testing and can therefore either under- or over-estimate the generalisation error. A better approach that uses all the data is the stratified k-fold method. You second task is to convert the current notebook to use stratified k-fold evaluation.

An example is shown in the notebook COMP2712 Evaluating Machine Learning:

https://colab.research.google.com/drive/1tbbjAMc9QetoYQRsB19KagXBz6d9Pqwl?usp=sharing

To gain this checkpoint you should again produce a table collating your results as was done for Checkpoint 1 and show the combined confusion matrix for the best result.

Show the modified notebook and explain to the demonstrator your results.

Checkpoint 3: Explore other parameter settings of the MLP

The MLP has a number of other settings that can be explored. Using the help documentation located here:

https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html

explore other parameter settings that lead to "more" optimal performance. You could explore what impact a different `solver` has, what about `momentum` or the `learning_rate`?

For this checkpoint expand the table to include the additional settings that you have explored

Show the modified notebook and explain to the demonstrator your results.