COMP 307/420 — *Introduction to* AI

Assignment 2: Neural and Evolutionary Learning

30% of Final Mark — Due: 23:59 Monday 25 May 2020

1 Objectives

The goal of this assignment is to help you understand the basic concepts and algorithms of neural and evolutionary learning, use these algorithms to perform regression and classification tasks, and analyse the results to draw some conclusions. In particular, the following topics should be reviewed:

- Perceptron learning for binary classification,
- Multilayer feed forward neural network architectures and applications,
- Back (error) propagation algorithm and its variations,
- Tackling a problem with an existing neural network package,
- Evolutionary computing and learning paradigms,
- Genetic programming for solving real world applications particularly for regression and binary classification problems, and
- Tackling a problem with an existing genetic programming package.

These topics are (to be) covered in lectures 07–12. The online materials can also be checked.

In this assignment, neural networks refer to the standard multilayer feed forward neural networks trained by the back propagation algorithm. Genetic programming refers to the standard genetic programming approach with the tree-like structure for the evolved programs.

2 Question Description

Part 1: Perceptron for Binary Classification [20 marks]

In this part, you are required to implement and apply the simple Perceptron Learning Algorithm to a binary classification task.

Data Set

Instance No.	Feature 1	Feature 2	Feature 3	Class/Target
1	0	0	1	0
2	0	1	0	1
3	1	0	1	1
4	1	1	0	0
5	1	1	1	0
6	1	0	0	1
7	0	1	1	1
8	0	0	0	0

Perceptron Algorithm

The algorithm for learning the weights of the perceptron was given in our lecture is:

```
Until the perceptron is always right (or some limit):
    Present an example (+ve or -ve)
    If perceptron is correct, do nothing
    If -ve example and wrong
        (ie, weights on active features are too high)
        Subtract feature vector from weight vector
    If +ve example and wrong
        (ie, weights on active features are too low)
        Add feature vector to weight vector
```

Requirements

You can write your program in Java, C/C++, Python, or any other programming language. Note that **you need to write your own code from scratch** in this question.

In this part of the assignment, you should submit:

- Program code for your perceptron classifier (both source code and executable program runnable on the ECS School machines).
- Make a training set file with the appropriate format used in your program.
- `readme.txt` describing how to run your program, and
- A report in PDF, text or DOC format. The report should:
  1. Report the classification accuracy your perceptron learning algorithm after running 200 epochs.
  2. Analyse and describe reasons that your algorithm could not achieve better results.

Part 2: Neural Networks for Classification [30 marks]

In this part, you are required to use an existing neural network package to perform classification on the *wine* data set described below. Please note that you already used this data set in the previous assignment.

Problem Description

The *wine* data set is taken from the UCI Machine Learning Repository (`http://archive.ics.uci.edu/ml/datasets.php`). The data set contains 178 instances in 3 classes, having 59, 71 and 48 instances, respectively. Each instance has 13 attributes: *Alcohol, Malic_acid, Ash, Alcalinity_of_ash, Magnesium, Total_phenols, Flavanoids, Non-flavanoid_phenols, Proanthocyanins, Color_intensity, Hue, OD280%2FOD315_of_diluted_wines, Proline.* The data set is split into two subsets, one for training and one for testing.

Requirements

You should define a neural network architecture for this problem, determine related network learning parameters, use the training set `wine-training` to learn/train your neural network, then apply the learned/trained neural network to the test set `wine-test`.

While it is good to write your own program packages to implement the back (error) propagation algorithm for training the multilayer feed forward networks, it might take you quite a while to do so. So we recommend the following neural network simulators (packages) for doing this assignment. One is BPNN (in ANSI C), and one is scikit-learn (or Keras, in Python). More information/instructions for using these packages could be found in the course assignment page.

You can choose any of these existing packages/simulators to perform the task. You can also choose any other package, but please describe your choice in your report.

You should submit the following files electronically.

- `new training set/test set file(s) with a correct format` for the neural network package you have chosen, and
- `A report` in the PDF, text or DOC formats. The report should:
    1. Determine and report the network architecture, including the number of input nodes, the number of output nodes, the number of hidden nodes (assume only one hidden layer is used here). Describe the rationale of your choice.
    2. Determine the learning parameters, including the learning rate, momentum, initial weight ranges, and any other parameters you used. Describe the rationale of your choice.
    3. Determine your network training termination criteria. Describe the rationale of your decision.
    4. Report your results (average results of 10 independent experiment runs with different random seeds) on both the training set and the test set. Analyse your results and make your conclusions.
    5. (optional/bonus 5 marks) Compare the performance of this method (neural networks) and the nearest neighbour methods.

Part 3: Genetic Programming for Symbolic Regression [30 marks]

In this part, you are required to use genetic programming to evolve a mathematical function (which is represented as an individual program in the population) for a simple symbolic regression task. In real world applications, a test set of data is needed in many situations but not needed in other scenarios, depending on the nature of the problems to be solved. In this assignment, to make the question simple, it is not required to have a test set — you are just required to evolve a mathematical model to reveal the relationship between the output variable and the input variable(s) from a (training) set of instances.

Problem Description

The task involves mapping a single input variable x to the (single) output variable y. In a 2D (two-dimensional) space (x-y coordinates), there are 20 points (x-y pairs). The task is to find a mathematical model to describe the relationship between the output variable y and input variable x. The 20 points are as follows. They are also saved in the file `regression` in plain text format.

x	-4.4	-4.2	-3.3	-3.0	-2.2	-2.4	-1.2	-1.6	-1.0	-0.5
y	5.721	4.882	1.685	1.0	0.04	0.119	0.64	0.16	0.956	2.25
x	0.3	0.6	1.7	2.0	2.4	2.9	3.9	3.6	4.4	5.0
y	5.27	6.76	13.69	16.0	19.391	24.01	34.81	31.36	40.96	49.0

Requirements

It is very time consuming to write your own GP package from scratch. As many GP packages are available, it is actually not necessary to write GP from scratch, but you can use the existing GP packages.. In this assignment, we recommend the following GP packages for the GP questions. They are JGAP (a Java GP library), ECJ (in Java), and DEAP (in Python). More information/instructions for using these packages could be found in the course assignment page.

Your job is to use any of the genetic programming packages with necessary changes of the terminal set, function set, fitness function, parameters and termination criteria to solve the task described above. Please describe your choice in your report.

You should submit the following files electronically.

- `Program code` written by yourself (both the source code and the executable program running on ECS School machines),
- `readme.txt` describing how to run your program, and
- `A report` in PDF, text or DOC format. The report should:
    1. Determine a good terminal set for this task.
    2. Determine a good function set for this task.

3. Construct a good fitness function and describe it using plain language, and mathematical formula (or any other format that can describe the fitness function as accurately as mathematical formula, such as pseudo code).
4. Describe the relevant parameter values and the stopping criteria you used.
5. Report the mean squared error for each of 10 independent runs with different random seeds, and their average value.
6. List three different best programs and their fitness values.
7. (optional, bonus, 5 marks) Analyse one of the best programs and explain why it can solve the problem in the task.

Part 4: Genetic Programming for Classification [40 marks]

In this part, you are required to use GP to classify the `Statlog (Landsat Satellite)` data set instances into two classes. This data set was obtained from the *UCI machine learning repository*

(`http://archive.ics.uci.edu/ml/datasets.php`).

Problem Description

The original data set was modified for this assignment. The new version of this data set has 195 instances of anomaly classification. This data set has 75 anomaly instances and 120 normal instances, where the task is to classify these instances into the *anomaly* and *normal* classes. Each instances has 36 attributes/features that are extracted from the satellite images. The file `satellite` contains all the 195 instances.

The 36 attributes are integer valued between 25 and 175. The attributes are denoted as V1, V2, ..., V36 in the file. The final column of the file shows the class label ("target") of each instance. The file satellite gives detailed description of the data.

Requirements

Your job is to use any of the genetic programming packages with necessary changes of the terminal set, function set, fitness function, parameters and termination criteria to solve the simple task described above.

You should submit the following files electronically.

- `Program code` written by yourself (both the source code and the executable program running on ECS School machines),
- `training.txt` and `test.txt` with appropriate formats,
- `readme.txt` describing how to run your program, and
- `A report` in PDF, text or DOC format. The report should:
  1. Determine a good terminal set for this task.
  2. Determine a good function set for this task.
  3. Construct a good fitness function and describe it using plain language, and mathematical formula (or any other format that can describe the fitness function as accurately as mathematical formula, such as pseudo code).
  4. Describe the relevant parameter values and the stopping criteria you used.
  5. Describe your main considerations in splitting the original data set into a training set `training.txt` and a test set `test.txt`.
  6. Report the classification accuracy (average accuracy over 10 independent experiment runs with different random seeds) on both the training set and the test set.
  7. List three best programs evolved by GP and the fitness value of them.
  8. (optional, bonus, 5 marks) Analyse one of best programs to identify patterns you can find in the evolved/learned program and why it can solve the problem well (or badly).

Part 5: Genetic Programming for Classification/Regression [30 marks]

(This part is only for COMP420 students. COMP307 students do not need to do this part.)

In this part, you are required to use GP to perform classification or regression on the data set (same as Part 1). The data set has eight instances with three features and a target ourput value/a class label (target value: 0 or 1). This problem can be solved as a regression task or a classification task.

Instance No.	Feature 1	Feature 2	Feature 3	Class/Target
1	0	0	1	0
2	0	1	0	1
3	1	0	1	1
4	1	1	0	0
5	1	1	1	0
6	1	0	0	1
7	0	1	1	1
8	0	0	0	0

Requirements

Your job is to use any of the genetic programming packages with necessary changes of the terminal set, function set, fitness function, parameters and termination criteria to solve the simple task described above.

You should submit the following files electronically.

- `Program code` written by yourself (both the source code and the executable program running on ECS School machines),
- `readme.txt` describing how to run your program, and
- `A report` in PDF, text or DOC format. The report should:
  1. List the function set (only arithmetic operators, no logic), the terminal set, and the fitness function of GP to solve this problem.
  2. Run the experiment five times and report the averaged results (either mean squared error or classification accuracy).
  3. Report three best (different) programs/trees evolved by GP from the five runs.
  4. Analyse the results and programs. Make your conclusions.
  5. Compare the performance of GP in this part and that of Perceptron that you obtained in Part 1 and make your own discussions.

3 Relevant Data Files and Program Files

The relevant data files, information files about the data sets, and some utility programs files can be found from the following directory:

`/vol/comp307/assignment2/`

Under this directory, there are three/four subdirectories `part1, part2`, `part3`, `part4`, and `part5 (COMP420 students only)` corresponding to the four/five parts of the assignment, respectively.

The data files can also be downloaded from the course website as a ZIP file.

4 Notes

As an assignment in a 300 level course, you can make your own assumptions if necessary.

During the time between the assignment handout and submission, the tutor(s) will run a number of helpdesks to provide assistance.

5 Submission Guidelines

5.1 Submission Requirements

1. Program codes you wrote and training (and test) set files that are used for all individual parts. To avoid confusion, all the individual parts should use directories `part1/, part2/, ...` and all pieces of programs should be stored in their corresponding directories. Within each directory, please provide a `readme file` that specifies how to compile and run your program on the ECS school machines. If possible, a script file called `sampleoutput.txt` should also be provided to show how your program run properly. If you programs cannot run properly, you can provide a `buglist` file. You need to submit these codes and files in *soft copy* only.

2. A document consisting of the report of all the individual parts. The document should mark each part clearly. The document can be written in PDF, text or the DOC format. You need to submit this document in *soft copy* only.

5.2 Submission Method

The programs and the PDF/Text/DOC version of the document should be submitted through the web submission system from the COMP307 course web site **by the due time**.

5.3 Late Penalties

All assignments must be submitted on time unless you have made a prior arrangement with the lecturer or have a valid medical excuse (for minor illnesses it is sufficient to discuss this with the lecturer.) The penalty for assignments that are handed in late without prior arrangement is one grade reduction per day. Assignments that are more than one week late will not be marked.