LZSCC.361 – Coursework Part II: AI Project

Due date: 15 December 2023, 3pm German Time (submit through MOODLE)

The maximum grade for the assessment is of **50 marks**

This assessment is worth **20% of your overall mark**

Introduction

In this coursework you are provided with a dataset of 900 labelled input data. The dataset is divided into a training dataset composed of 360 labelled data, and a validation dataset composed of 540 labelled data.

Each data point represents a point in a three dimensional space (i.e., each point is expressed as a triplet (x, y, z) where $2 \leq x \leq 8$, $-8 \leq y \leq -2$, and $-3 \leq z \leq 3$). Each data point is labelled/classified as belonging to either the class -1 or to the class 1.

There are four files for the dataset:

- `training_set` – containing the three coordinates of data points in the training set. Each point is represented as three comma separated numbers on a single line;
- `training_labels` – containing lines with either 1 or -1 for each data point in the training set. The order matters! – the first line represents the class of the first point in the `training_set`, the second line represents the class of the second point in the `training_set`, and so on;
- `validation_set` – containing the three coordinates of data points in the validation set. Each point is represented as three comma separated numbers on a single line; and
- `validation_labels` – containing lines with either 1 or -1 for each data point in the validation set following the same idea as for the training dataset.

You are asked to

1. implement a k-nearest neighbour classifier and compute the classification errors on both datasets for different values of k;
2. implement and experiment with a genetic algorithm to perform a regression based on the training data; and
3. report your approach/design choices, experiments, and comparison between classifier and genetic algorithm.

All tasks are described in details in the following sections.

Submission

This coursework requires you to submit two documents

1. a zip archive named "⟨studentID⟩_ai.zip". The archive must contain your implementations of the classifier and genetic algorithm;
2. a report as a pdf file named "⟨studentID⟩_ai.pdf"

IMPORTANT: do not put your name or ways to identify you anywhere within the code and/or report!

Restrictions

In your implementation, you are only allowed to import/use the following Python libraries.

- numpy – required for the genetic algorithm
- matplotlib – in case you decide to plot some of your experiments/results
- time – in case you decide to time your code within python
- random – required for the genetic algorithm

IMPORTANT: very heavy penalties (which could result in a mark of 0) are applied for the use of any other library!

Task 1 – Classifier

In this task you are asked to implement a k-nearest neighbour classifier. Your implementation should be saved in a file called `classifier.py` to be included in your zip file submission.

In order to use the classifier you need to use a distance measure among points. The input data is such that the standard Euclidean distance should do the trick, but feel free to use other distance measures (but you will have to justify why in your report).

Once the classifier has been implemented, run it for the three values of $k = 7$, $k = 19$, and $k = 31$ on both the training data and the validation data. For the three k values and two datasets, compute the classification error (i.e., the ratio between misclassified points over the dataset). Remember the training data points are the "memory" of the classifier and distances need to be computed only with respect to those!

Your implementation should be such that it outputs the six classification errors. That is, your output should look similar to the following lines (the numbers are not the correct ones!).

```
k=7
Classification error on training set:  0.0234
Classification error on validation set:  0.1234
k=19
Classification error on training set:  0.0567
Classification error on validation set:  0.1001
k=31
Classification error on training set:  0.1234
Classification error on validation set:  0.0098
```

Task 2 – Regression with Genetic Algorithm

In this task you are asked to use a genetic algorithm to find the values of a_0, a_1, and a_2 of the following function[1].

$$f(x,y) = a_o\sqrt[3]{x-5} + a_1\sqrt[3]{y+5} + a_2 \quad \text{with} \quad a_o \in [0,2], a_1 \in [-2,0], \text{ and } a_2 \in [-1,1].$$

This is an example of regression. That is, we want to minimise the distance between the surface defined by $f(x,y)$ and the points in the dataset in such a way that points classified as -1 are below the surface, and points classified as 1 are above the surface (or, in other words, you could see this second task also as a classification task where you are trying to define the boundary between the two classes).

Let P be the sets of points in your training set, then the genetic algorithm should look for values of a_0, a_1, and a_2 such that

[1] While implementing the algorithm, use the numpy function `np.cbrt` to compute the cubic root! Using the intuitive `**(1/3)` will not work!

$$min_{a_0,a_1,a_2} \sum_{(x,y,z)\in P} (z - f(x,y))^2$$

This is a standard approach to the problem, but you can maximise/minimise different objective functions if they result in a better genetic algorithm. For example, if we look at the problem as a classification problem, then we could change the aim of the genetic algorithm as follows.

$$max_{a_0,a_1,a_2} \sum_{(x,y,z)\in P} g(x,y,z) \quad \text{where } g(x,y,z) = \begin{cases} 1 & \text{if } z \geq f(x,y) \text{ and class } 1 \\ 1 & \text{if } z < f(x,y) \text{ and class } -1 \\ -1 & \text{otherwise} \end{cases}$$

In this task you are asked to implement a genetic algorithm to solve the problem based on the training dataset. The resulting implementation should be saved in a file called `regression.py` to be included in your zip file submission.

Remember that many decisions need to be made while designing a genetic algorithm, such as

- what is your objective function?
- what are the parameters to be encoded in the individuals' chromosomes?
- what is the mapping between genotype and phenotype? (i.e., how do you move between the actual values of your parameters and their representation in the chromosome?)
- how do you initialise your population? How big is your population?
- how do you assign fitness values to individuals?
- how do you perform selection?
- what are your genetic operators (i.e., crossover and mutation operators)? What is their probability of happening?
- how do you transition to the next generation? (e.g., elitism, new population, ...)
- what is your termination strategy?
- ...

You do not need to provide an explanation to the above points in your code, but you need to have initial answers in order to implement the algorithm (and you will have to explain all the above decisions in your report).

There is no way to ensure you have a good genetic algorithm at the first attempt, you need to experiment[2] with different strategies and parameters' values!

Once you have an implementation of the genetic algorithm and you have run your experiments to fine-tune the algorithm parameters, then find the values of a_0, a_1, and a_2 using only the training dataset. The output of the algorithm should look similar to the following lines (the numbers are the wrong ones!).

```
a0 = 0.1234
a1 = -1.2345
a2 = 0
```

Once you have the values, create another file called `boundary.py` to compute the classification error on both the training dataset and the validation dataset based on the values you have found. The file `boundary.py` should be included in your zip file submission.

[2] "To experiment" means running the algorithm multiple times, collecting and analysing the results! Do not forget to do this as it is asked in your report!

Task 3 – Report

Your last task is to write a report describing your design/implementation decisions, experiments you have run, an evaluation of your experiments, and a comparison between the classifier and the genetic algorithm outcomes.

The report should be at most 7 pages long (font sizes below 11pt are not allowed!).

The expected report structure is as follows.

- classifier (max 1.5 pages) – brief description of your classifier implementation including the distance measure used and a table with the six classification errors
- genetic algorithm design (max. 2 pages)– a description/justification of all design choices you have made (see the list on task 2)
- genetic algorithm experiments (max. 2 pages) – a description of the setting of your experiments, the results you got, and how those experiments guided you in your final decision for the parameters
- comparison and conclusions (max 1.5 pages) – compare at least the accuracy of the two approaches (i.e., classification errors), draw and justify your conclusions. You are encouraged to collect other statistics which could make the comparison more interesting (training time, validation time, memory usage, . . .)

In case you are using strategies found in the literature (NOT code or libraries), any reference is allowed to go beyond the 7 pages limit.

Marks Breakdown

Marks for the different parts of the coursework are distributed as follows.

- Classifier
  - code **[5 Marks]** – quality/legibility of code, e.g., comments, variable names, . . .
  - correct errors **[3 Marks]** – given a distance measure, the classifier is deterministic on the given datasets
- Genetic Algorithm
  - GA code **[10 Marks]** – quality/legibility of code, e.g., comments, variable names, . . .
  - GA accuracy **[6 Marks]** – 2 marks based on the classification errors; 4 marks based on the distance of the returned values with respect to the optimal ones. To be based on the average distance over 10 runs of the algorithm.
- Report
  - classifier report **[3 Marks]** – good description of the classifier implementation
  - GA design **[10 Marks]** – good explanation and justification of all the design choices
  - GA experiments **[10 Marks]** – good description of the experimental settings, experimental results, and how those results guided your fine-tuning of parameters
  - Comparison and conclusion **[3 Marks]** – good comparison and result evaluation of the two approaches