DV2618/DV2619 – Assignment 2 - Genetic Algorithm, Data Science and Implementations

This is individual assessed assignment. You are allowed to discuss this assessment with other students, but you should not copy their code, and you should not share your own code with other students. Note that we will carry out plagiarism checks on all submissions.

This assignment has **4 different tasks** and requires you to apply your knowledge of Genetic Algorithms, Graph Search, Data science and implementation (**preferably in Python)**. These topics are covered in the learning materials in Lectures 4, 5, 6 and 7 and it is strongly recommended that you work through the presentations from these weeks.

The **deadline for completing this assignment is the 9th October 2024, 23:59 pm**. You should submit your work (**e.g., zip file including your report, implementation etc.**) on Canvas course webpage. Note that, the zip file should be named with your name and surname (e.g., name_surname.zip).

Genetic Algorithms

Goal: To get acquainted with the idea and implementation of Genetic Algorithm (GA). To understand the effects of parameters in the GA on an optimization problem. To make necessary changes and modifications on the given implementation to resolve optimization problem.

Reading advice: Russell & Norvig: Artificial Intelligence - A Modern Approach, Section 4.3 in the 2nd edition.

Tasks

Tasks 1 (15 Points):

Problem Definition: You are going to solve a Travelling Salesman Problem (TSP). The TSP is defined as finding the shortest route between a set of locations or cities that must be visited by a salesman. The salesman's goal is to keep both the travel costs and the distance traveled as low as possible. Therefore, it is necessary to design and implement a computer-based algorithm to find these set of locations that must be visited by the salesman with a minimum cost. In this task, you will use a GA to resolve this optimization problem and the implementation of the GA is provided on the Canvas.

You will test the performance of the GA using different combinations of population sizes and mutation probabilities to understand and analyze the effects of parameters on the GA and you are going to change the following parameter values to solve the corresponding problem:

a) *Population size*: 10, 20, 50, 100
b) *Mutation Rate*: 0.9, 0.6, 0.3, 0.1
c) Do **not** change the other parameters in the GA.

Note that, you will run your GA implementation one time for each parameter value to obtain the final result at the end of the iteration of the algorithm. Use the provided Genetic Algorithm (GA) code available on Canvas (**Task1.ipynb**).

*** In the report, you must include, summarize, analyze, and discuss the results.**

Tasks 2 (20 Points):

Problem Definition: You are tasked with solving a single-objective optimization problem using a Genetic Algorithm (GA). Your primary objective is to implement a fitness function based on the following mathematical equation:

$$f(x, y, z) = 2 \cdot x \cdot z \cdot e^{-x} - 2 \cdot y^3 + y^2 - 3 \cdot z^3 + \frac{\cos(x \cdot z)}{1 + e^{-(x+y)}}$$

Instructions:

1. **Implement the Fitness Function:**
   - You need to write a Python function that computes the value of the above equation given inputs *x*, *y*, and *z*.
2. **Integration with GA Code:**
   - Use the provided Genetic Algorithm (GA) code available on Canvas (**Task2.ipynb**). You will integrate your fitness function into this existing GA framework.
   - The integrated GA framework should run properly, optimize the problem and find a final solution for the optimization problem.
3. **Report Requirements:**
   - In your report, include the implementation of the fitness function only. Ensure that your function is well-documented and correctly implemented.
   - Submit both your complete Python code and the report.
4. **Code Submission:**
   - Your Python code must include the fitness function as described and demonstrate that it correctly computes the values for optimization.
   - Ensure that the GA implementation you use operates properly with the fitness function and produces correct results.

Evaluation Criteria:

- Correctness and accuracy of the fitness function implementation.
- Integration with the GA code.
- The implementation should run properly without any issue and optimize the problem.
- Clarity and completeness of the report.

Tasks 3 (35 Points):

Problem Definition: You are going to solve a maze problem using GA. The aim of this task is to find a path in a maze, shown in figure 1, for a mouse to reach the destination or food with a shortest and acceptable path. You are going to implement a Genetic Algorithm that will find a solution to find a path by optimizing the problem automatically.

In this task, the implementation should be done in Python, and you should submit the *implementation in .py or ipynb*.

Hint: To solve this problem, you need to create a 2D - matrix where the possible passes should be 1 and obstacles must be a very large value (e.g., 1000). The matrix should be of the same size as the maze matrix.

More information and details about the problem can be found on the following webpages:
https://www.educative.io/answers/what-is-the-maze-problem

https://tonytruong.net/solving-a-2d-maze-game-using-a-genetic-algorithm-and-a-search-part-2/

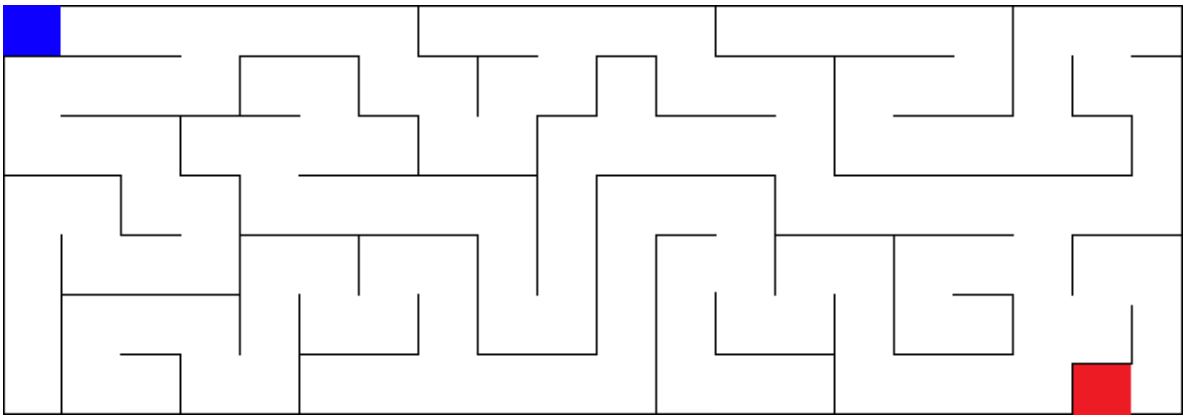

Figure 1. Maze.

Tasks 4 (30 Points):

Use Python software to complete this assignment. You can use sklearn, keras and/or tensorlow to complete the assignment.

1. Regression for Data

1.1 Generate data points in Python using:

x_data = np.linspace(−0.5, 0.5, 200)[:, np.newaxis]
noise = np.random.normal(0, 0.02, x_data.shape)
y_data = np.square(x_data) + noise

a. Establish a linear regression model to predict y data (you can use sklearn).
b. Establish a polynomial model with highest power of 2 to predict y data (you can use sklearn).
c. Establish a three-layer (including input, 1 hidden (6 nodes) and output layer) neural network to predict y data (keras or tensorflow is recommended). Split the dataset into training 80% and testing data 20% using the following code in Python:

```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x_data, y_data, test_size = 0.2, random_state = 0)
```

d. Calculate and compare mean squared errors of three models in 1.1 (a), (b) and (c).
e. Plot data points and curve of predictions in 1.1 (a) and 1.1 (b).

*** In the report, briefly explain what you have done in (a), (b), (c), (d), and (e). Also, you must submit your whole python code with the report.**

Evaluation

The assessment criteria document for the Assignment 2 is provided below.

Total Mark Range	Grade
>85	A
75-84	B
65-74	C
55-64	D
40-54	E

0-39	F

Important Note: Please ensure that your code is fully functional and does not require any debugging or modifications. Teachers will not spend additional effort troubleshooting or running your code. For full consideration and a successful evaluation of your assignment, your code must work seamlessly as provided. Additionally, you are allowed to use ChatGPT as a helper tool, but you are not allowed to use it for plagiarism. Ensure that all work is original and properly credited.