**a) Team Members:**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Email | Contribution | Tasks |
| John Lee | hyunmail94@csu.fullerton.edu | 50% | Brainstorm  Coding  Testing/Build  Troubleshoot  Documentation |
| Qiyuan Liu | qiyuan.liu@csu.fullerton.edu | 50% | Brainstorm  Coding  Testing/Build  Troubleshoot  Documentation |

\*Steps to run at end of document)\

**Problem 1:**

**Fitness function:** (sin(10\*PI\*value)\*value + 1)

How to calculate value (x value):

double bitValue = 0.0;  
 for (int i = 0; i < 22; i++) {  
 bitValue += (bitString[21 - i] - '0') \* pow(2.0, i);

// convert bit to its base 2 value  
 }  
 double value = bitValue \* 3 / (pow(2.0, 22) - 1) - 1.0;

// map base 2 value to value on [-1,2]

Fitness value = (sin(10\*PI\*value)\*value + 1)

**Size of population:** 500

**Crossover rate:** 0.8

**Mutation rate:** 0.01

**Termination criteria:** terminate the loop if one of the following condition is met:

1, number of generations exceeded the upper limit 3000

2, the maximum fitness value among one generation minus the maximum fitness value of its previous generation is less than the threshold 0.000001 and their corresponding x values are not too far away (> 0.1)

**Number of generations:** number of generations should not exceed 3000

**Problem 2:**

**Output:**

TRAINING DATA FROM CSV:

| ID | x1 | x2 | t |

| r1 | 1 | 1 | 1 |

| r2 | 9.4 | 6.4 | -1 |

| r3 | 2.5 | 2.1 | 1 |

| r4 | 8 | 7.7 | -1 |

| r5 | 0.5 | 2.2 | 1 |

| r6 | 7.9 | 8.4 | -1 |

| r7 | 7 | 7 | -1 |

| r8 | 2.8 | 0.8 | 1 |

| r9 | 1.2 | 3 | 1 |

| r10 | 7.8 | 6.1 | -1 |

Activation Function: f(net) = return 1.0 if net >=0, otherwise return -1.0

Learning Rate: 0.294161 //Random Value between [0.1,0.4]

Initial Weights: (-0.923584,-0.449129,0.175556) //Random Value between [-1,1] for (x1,x2,bias)

Epochs: 500

//Example output

r1 Final Updated Weights: (-0.923584,0.139193,2.52885) | A==T?: 1 == 1 (True)

r2 Final Updated Weights: (-0.923584,0.139193,2.52885) | A==T?: -1 == -1 (True)

r3 Final Updated Weights: (-0.923584,0.139193,2.52885) | A==T?: 1 == 1 (True)

r4 Final Updated Weights: (-0.923584,0.139193,2.52885) | A==T?: -1 == -1 (True)

r5 Final Updated Weights: (-0.923584,0.139193,2.52885) | A==T?: 1 == 1 (True)

r6 Final Updated Weights: (-0.923584,0.139193,2.52885) | A==T?: -1 == -1 (True)

r7 Final Updated Weights: (-0.923584,0.139193,2.52885) | A==T?: -1 == -1 (True)

r8 Final Updated Weights: (-0.923584,0.139193,2.52885) | A==T?: 1 == 1 (True)

r9 Final Updated Weights: (-0.923584,0.139193,2.52885) | A==T?: 1 == 1 (True)

r10 Final Updated Weights: (-0.923584,0.139193,2.52885) | A==T?: -1 == -1 (True)

Boundary Function = -0.923584x1 + 0.139193x2 + 2.52885

**How to Run:**

1. Download the project folder and cd into it from linux.
2. There are two folders : GeneticAlgorithm and PerceptronLearning
3. cd into the appropriate folder.
4. Type : “make clean” in case executable already exists.
5. Type : “make” into the command line.
6. The program should run on its own.
7. Repeat from step 2 for the other problem solution.