

CSE 354 - LAB 5

Arnav Jain - 220002018

Question 1

Code:

```
q1.py x q2.py
q1.py > NOR_DATA
1  import random
2
3  # Dataset for the gates
4  # format [ x1 , x2 , x3 , y1] where x1,x2,x3 are inputs and y1 is the output
5  AND_DATA = [
6      [0, 0, 0, 0],
7      [1, 0, 0, 0],
8      [0, 1, 0, 0],
9      [0, 0, 1, 0],
10     [1, 1, 0, 0],
11     [1, 0, 1, 0],
12     [0, 1, 1, 0],
13     [1, 1, 1, 1],
14 ]
15
16 OR_DATA = [
17     [0, 0, 0, 0],
18     [1, 0, 0, 1],
19     [0, 1, 0, 1],
20     [0, 0, 1, 1],
21     [1, 1, 0, 1],
22     [1, 0, 1, 1],
23     [0, 1, 1, 1],
24     [1, 1, 1, 1],
25 ]
26
27 NAND_DATA = [
28     [0, 0, 0, 1],
29     [1, 0, 0, 1],
30     [0, 1, 0, 1],
31     [0, 0, 1, 1],
32     [1, 1, 0, 1],
33     [1, 0, 1, 1],
34     [0, 1, 1, 1],
35     [1, 1, 1, 0],
36 ]
37
38 NOR_DATA = []
39     [0, 0, 0, 1],
40     [1, 0, 0, 0],
41     [0, 1, 0, 0],
42     [0, 0, 1, 0],
43     [1, 1, 0, 0],
44     [1, 0, 1, 0],
45     [0, 1, 1, 0],
46     [1, 1, 1, 0],
47 ]
```

```
q1.py x q2.py
q1.py > NOR_DATA
49 def perceptron(weights, data, learning_rate, num_epochs, threshold=0, bias=0, verbose = False):
50     input_size = len(weights)
51     if num_epochs == -1:
52         num_epochs = 1000
53     for epoch in range(num_epochs):
54         done = True
55         if verbose:
56             print(f"\nEpoch: {epoch}")
57         for d in data:
58             output = bias # adding bias as its default weight is always 1 .so its contribution to the output will be 1*bias
59             for weight, input in zip(weights, d[:input_size]):
60                 output += weight * input
61             if output > threshold:
62                 output = 1
63             else:
64                 output = 0
65             if d[-1] - output:
66                 for i in range(len(weights)):
67                     # print(learning_rate, (d[-1]-output), input)
68                     weight = weights[i]
69                     input = d[i]
70                     weight += learning_rate * (d[-1] - output) * input
71                     weights[i] = weight
72                     if verbose:
73                         print(f"Weight {i} : {weight}")
74                 bias += learning_rate * (d[-1] - output)
75                 if verbose:
76                     print(f"Bias: {bias}")
77                 done = False
78                 # break
79         if done:
80             print("All y_target = y_true predicted correctly")
81             print(f"Converged after {epoch} epochs")
82             break
83     return weights, bias
```

```
q1.py x q2.py
q1.py > NOR_DATA
85
86 def forward(weights, data, threshold, bias):
87     input_size = len(weights)
88     output = 0
89     for weight, input in zip(weights, data[:input_size]):
90         output += weight * input
91     output += bias
92     if output > threshold:
93         output = 1
94     else:
95         output = 0
96     return output
97
98
99 GATES = {"AND": AND_DATA, "OR": OR_DATA, "NAND": NAND_DATA, "NOR": NOR_DATA}
100
```

```
q1.py  x  q2.py
q1.py > [e] NOR_DATA
98
99  GATES = {"AND": AND_DATA, "OR": OR_DATA, "NAND": NAND_DATA, "NOR": NOR_DATA}
100
101  for gate_name, data in GATES.items():
102
103      print(f"Gate: {gate_name}")
104      print("Initialising Hyperparameters...")
105      # generating random hyperparameters
106      theta = random.uniform(-1, 1)
107      init_weights = [random.uniform(-1, 1) for _ in range(3)]
108      learning_rate = random.uniform(0.01, 0.5)
109      num_epochs = random.randint(100, 1000)
110      init_bias = random.uniform(-1, 1)
111
112      print(f"Initial Weights: {init_weights}")
113      print(f"Learning Rate: {learning_rate}")
114      print(f"Threshold: {theta}")
115      print(f"Bias: {init_bias}")
116
117      verbose = True
118      # running the perceptron algorithm
119      # start training
120      weights, bias = perceptron(
121          init_weights, data, learning_rate, num_epochs, theta, init_bias, verbose
122      )
123
124      # printing the final weights and bias after convergence
125      print("\nFinal Weights:")
126      for i, weight in enumerate(weights):
127          print(f"Weight {i}: {weight}")
128      print(f"Final Bias: {bias}")
129
130      # running validation
131      print("\nRunning Validation...")
132      for d in data:
133          print(
134              f"Input: {d[:-1]} \nOutput: {forward(weights, d, theta, bias)}\nActual Value: {d[-1]}"
135          )
136      print("\n\n\n")
```

Result:

```
PROBLEMS 5 OUTPUT DEBUG CONSOLE TERMINAL PORTS POLYGLOT NOTEBOOK GITLENS SPELL CHECKER 5 COMMENTS
• arnav@arnav-IdeaPad-Gaming-3-15ACH6:~/Desktop/Computational-Intelligence-Lab-CS354N/LAB 5$ python3 q1.py
Gate: AND
Initialising Hyperparameters...
Initial Weights: [0.4020370528296111, 0.9484220653821174, 0.8136972810746796]
Learning Rate: 0.32328281292060995
Threshold: -0.6353266388228132
Bias: -0.13517517691696246
All y_target = y_true predicted correctly
Converged after 3 epochs

Final Weights:
Weight 0: 0.4020370528296111
Weight 1: 0.6251392524615074
Weight 2: 0.4904144681540697
Final Bias: -1.7515892415200123

Running Validation...
Input: [0, 0, 0]
Output: 0
Actual Value: 0
Input: [1, 0, 0]
Output: 0
Actual Value: 0
Input: [0, 1, 0]
Output: 0
Actual Value: 0
Input: [0, 0, 1]
Output: 0
Actual Value: 0
Input: [1, 1, 0]
Output: 0
Actual Value: 0
Input: [1, 0, 1]
Output: 0
Actual Value: 0
Input: [0, 1, 1]
Output: 0
Actual Value: 0
Input: [1, 1, 1]
Output: 1
Actual Value: 1
```

```
Gate: OR
Initialising Hyperparameters...
Initial Weights: [0.6691092908238241, 0.14713774403541424, 0.6825577636364508]
Learning Rate: 0.018954019652259638
Threshold: -0.17822943795032153
Bias: 0.505446161967259
All y target = y true predicted correctly
Converged after 37 epochs
```

```
Final Weights:
Weight 0: 0.6691092908238241
Weight 1: 0.14713774403541424
Weight 2: 0.6825577636364508
Final Bias: -0.1958525651663473
```

```
Running Validation...
```

```
Input: [0, 0, 0]
Output: 0
Actual Value: 0
Input: [1, 0, 0]
Output: 1
Actual Value: 1
Input: [0, 1, 0]
Output: 1
Actual Value: 1
Input: [0, 0, 1]
Output: 1
Actual Value: 1
Input: [1, 1, 0]
Output: 1
Actual Value: 1
Input: [1, 0, 1]
Output: 1
Actual Value: 1
Input: [0, 1, 1]
Output: 1
Actual Value: 1
Input: [1, 1, 1]
Output: 1
Actual Value: 1
```

```
Gate: NAND
Initialising Hyperparameters...
Initial Weights: [0.7748892258552131, -0.09417410836742057, 0.14996056626676446]
Learning Rate: 0.4336010147079483
Threshold: 0.8779736612730189
Bias: 0.07325663920490566
All y_target = y_true predicted correctly
Converged after 10 epochs
```

```
Final Weights:
Weight 0: -0.5259138182686318
Weight 1: -0.5277751230753689
Weight 2: -1.5844434925650288
Final Bias: 3.108463742160543
```

```
Running Validation...
```

```
Input: [0, 0, 0]
Output: 1
Actual Value: 1
Input: [1, 0, 0]
Output: 1
Actual Value: 1
Input: [0, 1, 0]
Output: 1
Actual Value: 1
Input: [0, 0, 1]
Output: 1
Actual Value: 1
Input: [1, 1, 0]
Output: 1
Actual Value: 1
Input: [1, 0, 1]
Output: 1
Actual Value: 1
Input: [0, 1, 1]
Output: 1
Actual Value: 1
Input: [1, 1, 1]
Output: 0
Actual Value: 0
```

PROBLEMS 5 OUTPUT DEBUG CONSOLE TERMINAL PORTS POLYGLOT NOTEBOOK GITLENS SPELL CHECKER 5 COMMENTS

```
Output: 1
Actual Value: 1
Input: [1, 1, 1]
Output: 0
Actual Value: 0
```

```
Gate: NOR
Initialising Hyperparameters...
Initial Weights: [-0.19042455645316547, 0.9478164991132072, -0.2824019589837279]
Learning Rate: 0.25473895118504
Threshold: 0.40694025929801025
Bias: -0.7536715898288984
All y_target = y_true predicted correctly
Converged after 10 epochs
```

```
Final Weights:
Weight 0: -0.19042455645316547
Weight 1: -0.3258782568119927
Weight 2: -0.2824019589837279
Final Bias: 0.5200231660963015
```

```
Running Validation...
```

```
Input: [0, 0, 0]
Output: 1
Actual Value: 1
Input: [1, 0, 0]
Output: 0
Actual Value: 0
Input: [0, 1, 0]
Output: 0
Actual Value: 0
Input: [0, 0, 1]
Output: 0
Actual Value: 0
Input: [1, 1, 0]
Output: 0
Actual Value: 0
Input: [1, 0, 1]
Output: 0
Actual Value: 0
Input: [0, 1, 1]
Output: 0
Actual Value: 0
Input: [1, 1, 1]
Output: 0
Actual Value: 0
```

Question 2

Code:

```
q1.py  q2.py  X
q2.py > perceptron
1  import random
2  # Dataset for the task
3  # format [ x1 , x2 , y1] where x1,x2 are inputs and y1 is the output
4  DATASET = [
5      [1, 4, 0],
6      [1, 5, 0],
7      [2, 4, 0],
8      [2, 5, 0],
9      [3, 1, 1],
10     [3, 2, 1],
11     [4, 1, 1],
12     [4, 2, 1],
13 ]
```

```
q1.py  q2.py  X
q2.py > perceptron
14
15 def perceptron(weights, data, learning_rate, num_epochs, threshold=0, bias=0, verbose = False):
16     input_size = len(weights)
17     if num_epochs == -1:
18         num_epochs = 1000
19     for epoch in range(num_epochs):
20         done = True
21         if verbose:
22             print(f"\nEpoch: {epoch}")
23         for d in data:
24             output = bias # adding bias as its default weight is always 1 .so its contribution to the output will be 1*bias
25             for weight, input in zip(weights, d[:input_size]):
26                 output += weight * input
27             if output > threshold:
28                 output = 1
29             else:
30                 output = 0
31             if d[-1] - output:
32                 for i in range(len(weights)):
33                     # print(learning_rate, (d[-1]-output), input)
34                     weight = weights[i]
35                     input = d[i]
36                     weight += learning_rate * (d[-1] - output) * input
37                     weights[i] = weight
38                     if verbose:
39                         print(f"Weight {i} : {weight}")
40                 bias += learning_rate * (d[-1] - output)
41                 if verbose:
42                     print(f"Bias: {bias}")
43                 done = False
44                 # break
45         if done:
46             print("All y_target = y_true predicted correctly")
47             print(f"Converged after {epoch} epochs")
48             break
49     return weights, bias
```



```
q1.py  q2.py  X
q2.py > perceptron
51
52 def forward(weights, data, threshold, bias):
53     input_size = len(weights)
54     output = 0
55     for weight, input in zip(weights, data[:input_size]):
56         output += weight * input
57     output += bias
58     if output > threshold:
59         output = 1
60     else:
61         output = 0
62     return output
63
64
65 dataset = {"Factory Classification": DATASET}
```

```
q1.py  X  q2.py  X
q2.py > perceptron
64
65 dataset = {"Factory Classification": DATASET}
66
67 for task, data in dataset.items():
68
69     print(f"Task: {task}")
70     print("Initialising Hyperparameters...")
71     # generating random hyperparameters
72     theta = random.uniform(-1, 1)
73     init_weights = [random.uniform(-1, 1) for _ in range(3)]
74     learning_rate = random.uniform(0.01, 0.5)
75     num_epochs = random.randint(100, 1000)
76     init_bias = random.uniform(-1, 1)
77
78     print(f"Initial Weights: {init_weights}")
79     print(f"Learning Rate: {learning_rate}")
80     print(f"Threshold: {theta}")
81     print(f"Bias: {init_bias}")
82
83     verbose = True
84     # running the perceptron algorithm
85     # start training
86     weights, bias = perceptron(
87         init_weights, data, learning_rate, num_epochs, theta, init_bias, verbose
88     )
89
90     # printing the final weights and bias after convergence
91     print("\nFinal Weights:")
92     for i, weight in enumerate(weights):
93         print(f"Weight {i}: {weight}")
94     print(f"Final Bias: {bias}")
95
96     # running validation
97     print("\nRunning Validation...")
98     for d in data:
99         print(
100             f"Input: {d[:-1]} \nOutput: {forward(weights, d, theta, bias)}\nActual Value: {d[-1]}"
101         )
102     print("\n\n")
```

Result:

```
PROBLEMS 5 OUTPUT DEBUG CONSOLE TERMINAL PORTS POLYGLOT NOTEBOOK GITLENS SPELL CHECKER 5 COMMENTS
● arnav@arnav-IdeaPad-Gaming-3-15ACH6:~/Desktop/Computational-Intelligence-Lab-CS354N/LAB 5$ python3 q2.py
Task: Factory Classification
Initialising Hyperparameters...
Initial Weights: [-0.26649295281925567, 0.6592749845250931, -0.3958168759172176]
Learning Rate: 0.2466893393969161
Threshold: 0.41824648212016347
Bias: -0.7332648908009964

Epoch: 0
Weight 0 : -0.5131822922161717
Weight 1 : -0.3274823730625712
Weight 2 : -0.3958168759172176
Bias: -0.9799542301979125
Weight 0 : 0.22688572597457657
Weight 1 : -0.08079303366565513
Weight 2 : -0.1491275365203015
Bias: -0.7332648908009964
Weight 0 : 0.9669537441653249
Weight 1 : 0.4125856451281771
Weight 2 : 0.09756180287661459
Bias: -0.4865755514040804

Epoch: 1
Weight 0 : 0.7202644047684088
Weight 1 : -0.5741717124594873
Weight 2 : 0.09756180287661459
Bias: -0.7332648908009964
Weight 0 : 1.4603324229591572
Weight 1 : -0.0807930336656551
Weight 2 : 0.3442511422735307
Bias: -0.4865755514040804

Epoch: 2
Weight 0 : 1.2136430835622412
Weight 1 : -1.0675503912533195
Weight 2 : 0.3442511422735307
Bias: -0.7332648908009964

Epoch: 3
All y_target = y_true predicted correctly
Converged after 3 epochs

Final Weights:
Weight 0: 1.2136430835622412
Weight 1: -1.0675503912533195
Weight 2: 0.3442511422735307
Final Bias: -0.7332648908009964

Running Validation...
Input: [1, 4]
Output: 0
Actual Value: 0
Input: [1, 5]
Output: 0
```

```
PROBLEMS 5 OUTPUT DEBUG CONSOLE TERMINAL PORTS POLYGLOT NOTEBOOK GITLENS SPELL CHECKER 5 COMMENTS

Weight 1 : -0.5741717124594873
Weight 2 : 0.09756180287661459
Bias: -0.7332648908009964
Weight 0 : 1.4603324229591572
Weight 1 : -0.0807930336656551
Weight 2 : 0.3442511422735307
Bias: -0.4865755514040804

Epoch: 2
Weight 0 : 1.2136430835622412
Weight 1 : -1.0675503912533195
Weight 2 : 0.3442511422735307
Bias: -0.7332648908009964

Epoch: 3
All y target = y true predicted correctly
Converged after 3 epochs

Final Weights:
Weight 0: 1.2136430835622412
Weight 1: -1.0675503912533195
Weight 2: 0.3442511422735307
Final Bias: -0.7332648908009964

Running Validation...
Input: [1, 4]
Output: 0
Actual Value: 0
Input: [1, 5]
Output: 0
Actual Value: 0
Input: [2, 4]
Output: 0
Actual Value: 0
Input: [2, 5]
Output: 0
Actual Value: 0
Input: [3, 1]
Output: 1
Actual Value: 1
Input: [3, 2]
Output: 1
Actual Value: 1
Input: [4, 1]
Output: 1
Actual Value: 1
Input: [4, 2]
Output: 1
Actual Value: 1

arnav@arnav-IdeaPad-Gaming-3-15ACH6:~/Desktop/Computational-Intelligence-Lab-CS354N/LAB 5$
```

For code refer here:

<https://github.com/arnavjain2710/Computational-Intelligence-Lab-CS354N/tree/main/LAB%205>