CSE 354 - LAB 4

Arnav Jain - 220002018

Question 1

Code:

q1.cpp

```cpp
#include <iostream>
#include <vector>
#include <cstdlib>
#include <ctime>
using namespace std;

class MPNeuron {
public:
    MPNeuron(int numInputs, int threshold) {
        this->threshold = threshold;
        // Generate random weights for the inputs
        weights.resize(numInputs);
        for (int i = 0; i < numInputs; ++i) {
            weights[i] = rand() % 2;
        }
    }

    int computeOutput(const vector<int>& inputs) {
        if (inputs.size() != weights.size()) {
            cout << "Input and weight vectors must have the same size!" << endl;
            return -1;
        }

        int sum = 0;
        for (size_t i = 0; i < inputs.size(); ++i) {
            sum += inputs[i] * weights[i];
        }
        return sum >= threshold ? 1 : 0;
    }

    void displayWeights() {
        cout << "Randomly assigned weights: ";
        for (int weight : weights) {
            cout << weight << " ";
        }
        cout << endl;
    }

private:
    vector<int> weights;
    int threshold;
};
```

q1.cpp

```cpp
class MPNeuron {
    void displayWeights() {
    }

private:
    vector<int> weights;
    int threshold;
};

int main() {
    srand(time(0));

    int numInputs, threshold;

    cout << "Enter the number of input nodes: ";
    cin >> numInputs;
    cout << "Enter the threshold: ";
    cin >> threshold;
    vector<int> inputs(numInputs);
    cout << "Enter the " << numInputs << " inputs (0 or 1): ";
    for (int i = 0; i < numInputs; ++i) {
        cin >> inputs[i];
    }

    // Create the MPNeuron object with random weights and the provided threshold
    MPNeuron neuron(numInputs, threshold);
    neuron.displayWeights();

    int output = neuron.computeOutput(inputs);
    cout << "Net output: " << output << endl;

    return 0;
}
```

Result:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  POLYGLOT NOTEBOOK  GITLENS  SPELL CHECKER
arnav@arnav-IdeaPad-Gaming-3-15ACH6:~/Desktop/CI LAB/LAB 4$ ./q1
Enter the number of input nodes: 3
Enter the threshold: 2
Enter the 3 inputs (0 or 1): 1 1 1
Randomly assigned weights: 0 1 1
Net output: 1
arnav@arnav-IdeaPad-Gaming-3-15ACH6:~/Desktop/CI LAB/LAB 4$
```

Question 2

Code:

q2.cpp

q2.cpp > MPNeuron > computeOutput(const vector<int>&)

```cpp
#include <iostream>
#include <vector>
using namespace std;

class MPNeuron {
public:
    MPNeuron(int numInputs) {
        weights.resize(numInputs, 1);  // Setting all weights to 1
        threshold = numInputs; // AND gate requires all 1s to output 1 , hence threshold is set to numInputs
    }

    int computeOutput(const vector<int>& inputs) {
        if (inputs.size() != weights.size()) {
            cout << "Input and weight vectors must have the same size!" << endl;
            return -1;
        }

        int sum = 0;
        for (size_t i = 0; i < inputs.size(); ++i) {
            sum += inputs[i] * weights[i];
        }

        return sum >= threshold ? 1 : 0;
    }

private:
    vector<int> weights;
    int threshold;
};
```

q2.cpp

```
q2.cpp > MPNeuron > computeOutput(const vector<int>&)
class MPNeuron {
private:
    vector<int> weights;
    int threshold;
};

int main() {
    int numInputs;
    cout << "Enter the number of input nodes (e.g., 3): ";
    cin >> numInputs;

    cout << "AND Gate Implementation" << endl;
    vector<int> inputs(numInputs);
    cout << "Enter " << numInputs << " inputs (0 or 1): ";
    for (int i = 0; i < numInputs; ++i) {
        cin >> inputs[i];
    }

    MPNeuron neuron(numInputs);
    int output = neuron.computeOutput(inputs);

    cout << "Output: " << output << endl;

    return 0;
}
```

Result:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  POLYGLOT NOTEBOOK  GITLENS  SPELL CHECKER
arnav@arnav-IdeaPad-Gaming-3-15ACH6:~/Desktop/CI LAB/LAB 4$ ./q2
Enter the number of input nodes (e.g., 3): 3
AND Gate Implementation
Enter 3 inputs (0 or 1): 1 0 1
Output: 0
arnav@arnav-IdeaPad-Gaming-3-15ACH6:~/Desktop/CI LAB/LAB 4$ ./q2
Enter the number of input nodes (e.g., 3): 3
AND Gate Implementation
Enter 3 inputs (0 or 1): 1 1 1
Output: 1
arnav@arnav-IdeaPad-Gaming-3-15ACH6:~/Desktop/CI LAB/LAB 4$
```

Question 3

Code:

q3.cpp

q3.cpp > MPNeuron > computeOutput(const vector<int>&)

```cpp
#include <iostream>
#include <vector>
using namespace std;

class MPNeuron {
public:
    MPNeuron(int numInputs) {
        weights.resize(numInputs, 1);  // Set all weights to 1
        threshold = numInputs; // NAND gate requires all 1s to output 0
    }

    int computeOutput(const vector<int>& inputs) {
        if (inputs.size() != weights.size()) {
            cout << "Input and weight vectors must have the same size!" << endl;
            return -1;
        }

        int sum = 0;
        for (size_t i = 0; i < inputs.size(); ++i) {
            sum += inputs[i] * weights[i];
        }

        return sum >= threshold ? 0 : 1;
    }

private:
    vector<int> weights;
    int threshold;
};
```

q3.cpp

q3.cpp > MPNeuron > computeOutput(const vector<int>&)

```
 5  class MPNeuron {
26  private:
27      vector<int> weights;
28      int threshold;
29  };
30
31  int main() {
32      int numInputs;
33      cout << "Enter the number of input nodes (e.g., 3): ";
34      cin >> numInputs;
35
36      cout << "NAND Gate Implementation" << endl;
37      vector<int> inputs(numInputs);
38      cout << "Enter " << numInputs << " inputs (0 or 1): ";
39      for (int i = 0; i < numInputs; ++i) {
40          cin >> inputs[i];
41      }
42
43      MPNeuron neuron(numInputs);
44      int output = neuron.computeOutput(inputs);
45
46      cout << "Output: " << output << endl;
47
48      return 0;
49  }
```

Result:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  POLYGLOT NOTEBOOK  GITLENS  SPELL CHECKER
arnav@arnav-IdeaPad-Gaming-3-15ACH6:~/Desktop/CI LAB/LAB 4$ g++ q3.cpp -o q3
arnav@arnav-IdeaPad-Gaming-3-15ACH6:~/Desktop/CI LAB/LAB 4$ ./q3
Enter the number of input nodes (e.g., 3): 3
NAND Gate Implementation
Enter 3 inputs (0 or 1): 1 0 1
Output: 1
arnav@arnav-IdeaPad-Gaming-3-15ACH6:~/Desktop/CI LAB/LAB 4$ ./q3
Enter the number of input nodes (e.g., 3): 3
NAND Gate Implementation
Enter 3 inputs (0 or 1): 1 1 1
Output: 0
arnav@arnav-IdeaPad-Gaming-3-15ACH6:~/Desktop/CI LAB/LAB 4$
```

Question 4

Code:

q4.cpp

q4.cpp > MPNeuron > computeOutput(const vector<int>&)

```cpp
#include <iostream>
#include <vector>
using namespace std;

class MPNeuron {
public:
    MPNeuron(int numInputs) {
        weights.resize(numInputs, 1);  // Set all weights to 1
        threshold = 1; // OR gate requires at least one input to be 1
    }

    int computeOutput(const vector<int>& inputs) {
        if (inputs.size() != weights.size()) {
            cout << "Input and weight vectors must have the same size!" << endl;
            return -1;
        }

        int sum = 0;
        for (size_t i = 0; i < inputs.size(); ++i) {
            sum += inputs[i] * weights[i];
        }

        return sum >= threshold ? 1 : 0;
    }

private:
    vector<int> weights;
    int threshold;
};

```

q4.cpp

q4.cpp > MPNeuron > computeOutput(const vector<int>&)

```cpp
class MPNeuron {
private:
    vector<int> weights;
    int threshold;
};

int main() {
    int numInputs;
    cout << "Enter the number of input nodes (e.g., 3): ";
    cin >> numInputs;

    cout << "OR Gate Implementation" << endl;
    vector<int> inputs(numInputs);
    cout << "Enter " << numInputs << " inputs (0 or 1): ";
    for (int i = 0; i < numInputs; ++i) {
        cin >> inputs[i];
    }

    MPNeuron neuron(numInputs);
    int output = neuron.computeOutput(inputs);

    cout << "Output: " << output << endl;

    return 0;
}
```

Result:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  POLYGLOT NOTEBOOK  GITLENS  SPELL CHECKER
arnav@arnav-IdeaPad-Gaming-3-15ACH6:~/Desktop/CI LAB/LAB 4$ g++ q4.cpp -o q4
arnav@arnav-IdeaPad-Gaming-3-15ACH6:~/Desktop/CI LAB/LAB 4$ ./q4
Enter the number of input nodes (e.g., 3): 3
OR Gate Implementation
Enter 3 inputs (0 or 1): 1 0 1
Output: 1
arnav@arnav-IdeaPad-Gaming-3-15ACH6:~/Desktop/CI LAB/LAB 4$ ./q4
Enter the number of input nodes (e.g., 3): 3
OR Gate Implementation
Enter 3 inputs (0 or 1): 0 0 0
Output: 0
arnav@arnav-IdeaPad-Gaming-3-15ACH6:~/Desktop/CI LAB/LAB 4$
```

Question 5

Code:

q5.cpp

q5.cpp > MPNeuron > computeOutput(const vector<int>&)

```cpp
#include <iostream>
#include <vector>
using namespace std;

class MPNeuron {
public:
    MPNeuron(int numInputs) {
        weights.resize(numInputs, 1);  // Set all weights to 1
        threshold = 1; // NOR gate requires all inputs to be 0
    }

    int computeOutput(const vector<int>& inputs) {
        if (inputs.size() != weights.size()) {
            cout << "Input and weight vectors must have the same size!" << endl;
            return -1;
        }

        int sum = 0;
        for (size_t i = 0; i < inputs.size(); ++i) {
            sum += inputs[i] * weights[i];
        }

        return sum >= threshold ? 0 : 1;
    }

private:
    vector<int> weights;
    int threshold;
};
```

```cpp
q5.cpp > MPNeuron > computeOutput(const vector<int>&)
class MPNeuron {
private:
    vector<int> weights;
    int threshold;
};

int main() {
    int numInputs;
    cout << "Enter the number of input nodes (e.g., 3): ";
    cin >> numInputs;

    cout << "NOR Gate Implementation" << endl;
    vector<int> inputs(numInputs);
    cout << "Enter " << numInputs << " inputs (0 or 1): ";
    for (int i = 0; i < numInputs; ++i) {
        cin >> inputs[i];
    }

    MPNeuron neuron(numInputs);
    int output = neuron.computeOutput(inputs);

    cout << "Output: " << output << endl;

    return 0;
}
```

Result:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  POLYGLOT NOTEBOOK  GITLENS  SPELL CHECKER
arnav@arnav-IdeaPad-Gaming-3-15ACH6:~/Desktop/CI LAB/LAB 4$ g++ q5.cpp -o q5
arnav@arnav-IdeaPad-Gaming-3-15ACH6:~/Desktop/CI LAB/LAB 4$ ./q5
Enter the number of input nodes (e.g., 3): 3
NOR Gate Implementation
Enter 3 inputs (0 or 1): 1 1 1
Output: 0
arnav@arnav-IdeaPad-Gaming-3-15ACH6:~/Desktop/CI LAB/LAB 4$ ./q5
Enter the number of input nodes (e.g., 3): 3
NOR Gate Implementation
Enter 3 inputs (0 or 1): 1 0 1
Output: 0
arnav@arnav-IdeaPad-Gaming-3-15ACH6:~/Desktop/CI LAB/LAB 4$
```

For Code , refer to GitHub:

https://github.com/arnavjain2710/Computational-Intelligence-Lab-CS354N/tree/main/LAB%204