CSE 354 - LAB 4

Arnav Jain - 220002018

Question 1

Code:

q1.cpp

```cpp
#include <iostream>
#include <vector>
#include <cstdlib>
#include <ctime>
#include <fstream>
#include <sstream>
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 {

int main(int argc, char* argv[]) {
    if (argc != 2) {
        cerr << "Usage: " << argv[0] << " <filename>" << endl;
        return 1;
    }

    string filename = argv[1];
    ifstream file(filename);
    if (!file) {
        cerr << "Error opening file!" << endl;
        return 1;
    }

    int numIterations;
    file >> numIterations; // number of iterations
    file.ignore();

    for (int i = 0; i < numIterations; ++i) {
        cout<<"Iteration "<<i+1<<endl;
        int numInputs, threshold;
        file >> numInputs;  // number of inputs
        file.ignore();
        file >> threshold;  // threshold
        file.ignore();

        vector<int> inputs(numInputs);
        for (int j = 0; j < numInputs; ++j) {
            file >> inputs[j];  // inputs
        }

    MPNeuron neuron(numInputs, threshold);
    neuron.displayWeights();

    int output = neuron.computeOutput(inputs);
    cout << "For input ";
    for (int j = 0; j < numInputs; ++j) {
            cout << inputs[j] << " ";
        }
    cout << ", the Net output is: " << output << endl;
    cout<<endl;
    }

    return 0;
}
```

Input File:

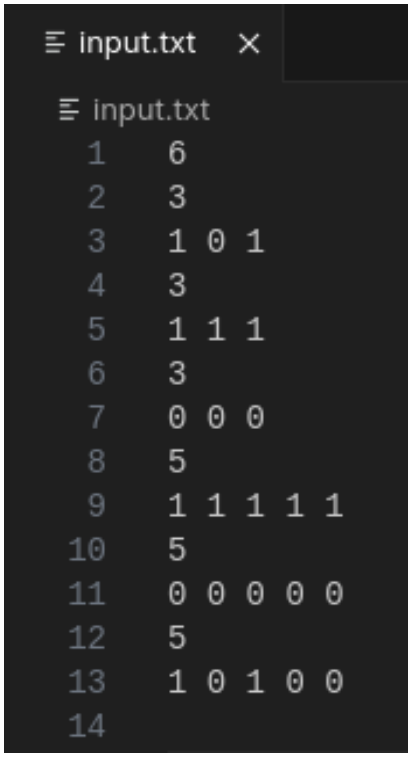
q1_input.txt

```
6
3
0
1 0 1
3
1
1 1 1
3
2
0 0 0
5
0
1 1 1 1 1
5
2
0 0 0 0 0
5
4
1 0 1 0 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 q1_input.txt
Iteration 1
Randomly assigned weights: 1 0 1
For input 1 0 1 , the Net output is: 1

Iteration 2
Randomly assigned weights: 1 1 1
For input 1 1 1 , the Net output is: 1

Iteration 3
Randomly assigned weights: 0 0 1
For input 0 0 0 , the Net output is: 0

Iteration 4
Randomly assigned weights: 1 0 1 0 1
For input 1 1 1 1 1 , the Net output is: 1

Iteration 5
Randomly assigned weights: 1 0 0 0 0
For input 0 0 0 0 0 , the Net output is: 0

Iteration 6
Randomly assigned weights: 0 1 0 1 1
For input 1 0 1 0 0 , the Net output is: 0

arnav@arnav-IdeaPad-Gaming-3-15ACH6:~/Desktop/CI LAB/LAB 4$
```

For next questions , the input file used is:



Question 2

Code:

q2.cpp

q2.cpp > main(int, char * [])

You, 6 minutes ago | 1 author (You)

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

You, 2 days ago | 1 author (You)
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

```cpp
class MPNeuron {

int main(int argc, char* argv[]) {
    if (argc != 2) {
        cerr << "Usage: " << argv[0] << " <filename>" << endl;
        return 1;
    }

    string filename = argv[1];
    ifstream file(filename);
    if (!file) {
        cerr << "Error opening file!" << endl;
        return 1;
    }

    int numIterations;
    file >> numIterations; // Read the number of iterations
    file.ignore();

    for (int i = 0; i < numIterations; ++i) {
        cout<<"Iteration "<<i+1<<endl;
        int numInputs;
        file >> numInputs;  // Read the number of inputs for this iteration
        file.ignore();

        vector<int> inputs(numInputs);
        for (int j = 0; j < numInputs; ++j) {
            file >> inputs[j];  // inputs
        }

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

        cout << "For input ";
        for (int j = 0; j < numInputs; ++j) {
            cout << inputs[j] << " ";
        }
        cout << ", the output of AND gate is: " << output << endl;
        cout<< endl;      You, 6 minutes ago • Uncommitted changes
    }

    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 input.txt
Iteration 1
For input 1 0 1 , the output of AND gate is: 0

Iteration 2
For input 1 1 1 , the output of AND gate is: 1

Iteration 3
For input 0 0 0 , the output of AND gate is: 0

Iteration 4
For input 1 1 1 1 1 , the output of AND gate is: 1

Iteration 5
For input 0 0 0 0 0 , the output of AND gate is: 0

Iteration 6
For input 1 0 1 0 0 , the output of AND gate is: 0

arnav@arnav-IdeaPad-Gaming-3-15ACH6:~/Desktop/CI LAB/LAB 4$
```

Question 3

Code:

q3.cpp

q3.cpp > main(int, char * [])

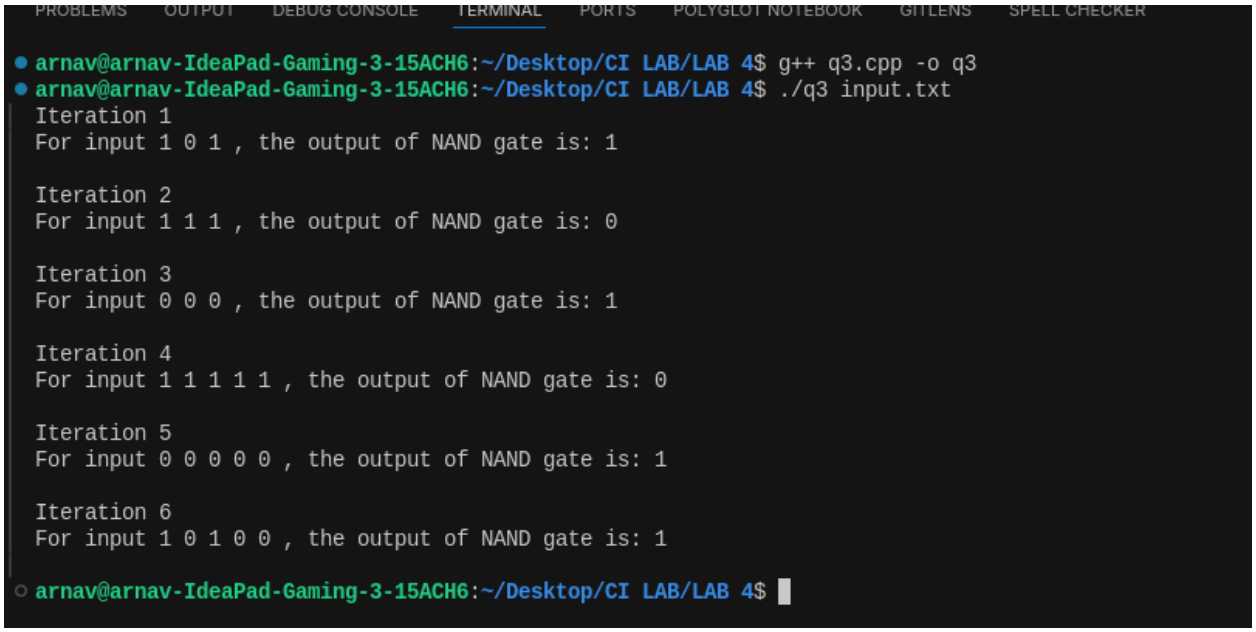
You, 8 minutes ago | 1 author (You)

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

You, 8 minutes ago | 1 author (You)
class MPNeuron {
public:
    MPNeuron(int numInputs) {
        weights.resize(numInputs, -1);  // Set all weights to -1
        threshold = -1*numInputs + 1; // 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 ? 1 : 0;
    }

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

q3.cpp

```cpp
class MPNeuron {

int main(int argc, char* argv[]) {
    if (argc != 2) {
        cerr << "Usage: " << argv[0] << " <filename>" << endl;
        return 1;
    }

    string filename = argv[1];
    ifstream file(filename);
    if (!file) {
        cerr << "Error opening file!" << endl;
        return 1;
    }

    int numIterations;
    file >> numIterations; // Read the number of iterations
    file.ignore();

    for (int i = 0; i < numIterations; ++i) {
        cout<<"Iteration "<<i+1<<endl;
        int numInputs;
        file >> numInputs;  // Read the number of inputs for this iteration
        file.ignore();

        vector<int> inputs(numInputs);
        for (int j = 0; j < numInputs; ++j) {
            file >> inputs[j];  // inputs
        }

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

        cout << "For input ";
        for (int j = 0; j < numInputs; ++j) {
            cout << inputs[j] << " ";
        }
        cout << ", the output of NAND gate is: " << output << endl;
        cout<< 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++ q3.cpp -o q3
arnav@arnav-IdeaPad-Gaming-3-15ACH6:~/Desktop/CI LAB/LAB 4$ ./q3 input.txt
Iteration 1
For input 1 0 1 , the output of NAND gate is: 1
Iteration 2
For input 1 1 1 , the output of NAND gate is: 0
Iteration 3
For input 0 0 0 , the output of NAND gate is: 1
Iteration 4
For input 1 1 1 1 1 , the output of NAND gate is: 0
Iteration 5
For input 0 0 0 0 0 , the output of NAND gate is: 1
Iteration 6
For input 1 0 1 0 0 , the output of NAND gate is: 1
arnav@arnav-IdeaPad-Gaming-3-15ACH6:~/Desktop/CI LAB/LAB 4$

Question 4

Code:

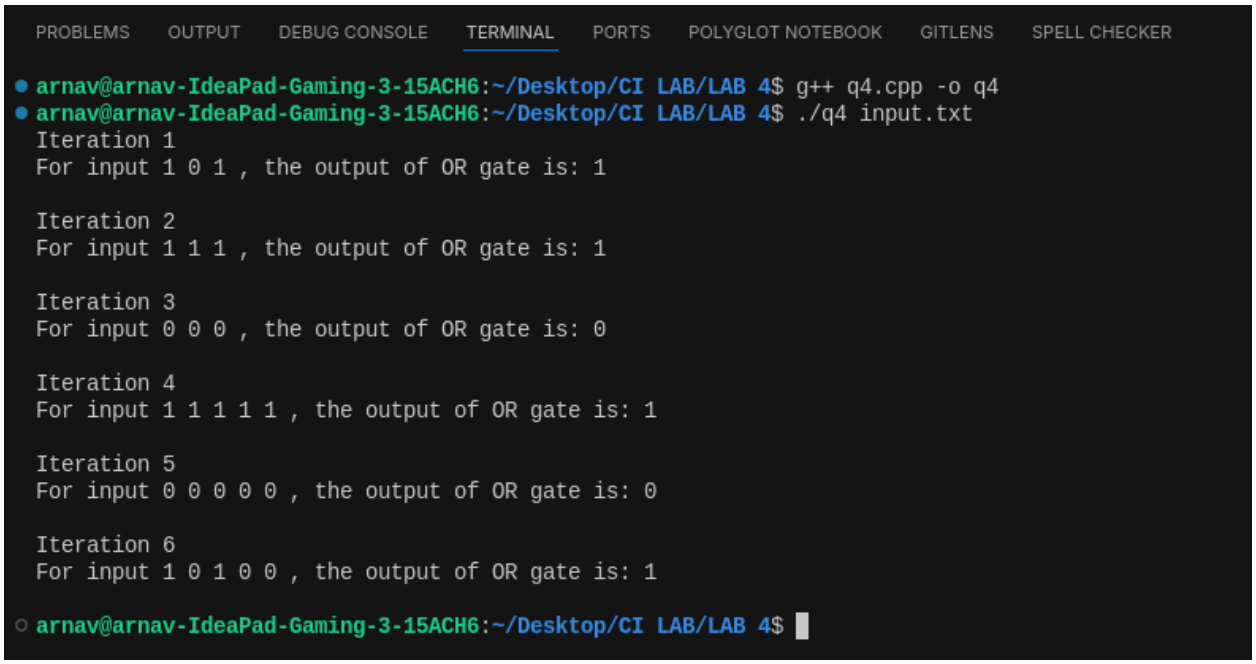
q4.cpp

```cpp
#include <iostream>
#include <fstream>
#include <vector>
#include <sstream>
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

```
 7  class MPNeuron {
32
33  int main(int argc, char* argv[]) {
34      if (argc != 2) {
35          cerr << "Usage: " << argv[0] << " <filename>" << endl;
36          return 1;
37      }
38
39      string filename = argv[1];
40      ifstream file(filename);
41      if (!file) {
42          cerr << "Error opening file!" << endl;
43          return 1;
44      }
45
46      int numIterations;
47      file >> numIterations; // Read the number of iterations
48      file.ignore();
49
50      for (int i = 0; i < numIterations; ++i) {
51          cout<<"Iteration "<<i+1<<endl;
52          int numInputs;
53          file >> numInputs;  // Read the number of inputs for this iteration
54          file.ignore();
55
56          vector<int> inputs(numInputs);
57          for (int j = 0; j < numInputs; ++j) {
58              file >> inputs[j];  // inputs
59          }
60
61          MPNeuron neuron(numInputs);
62          int output = neuron.computeOutput(inputs);
63
64          cout << "For input ";
65          for (int j = 0; j < numInputs; ++j) {
66              cout << inputs[j] << " ";
67          }
68          cout << ", the output of OR gate is: " << output << endl;
69          cout<< endl;
70      }
71
72      return 0;
73  }
74
```

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 input.txt
Iteration 1
For input 1 0 1 , the output of OR gate is: 1
Iteration 2
For input 1 1 1 , the output of OR gate is: 1
Iteration 3
For input 0 0 0 , the output of OR gate is: 0
Iteration 4
For input 1 1 1 1 1 , the output of OR gate is: 1
Iteration 5
For input 0 0 0 0 0 , the output of OR gate is: 0
Iteration 6
For input 1 0 1 0 0 , the output of OR gate is: 1
arnav@arnav-IdeaPad-Gaming-3-15ACH6:~/Desktop/CI LAB/LAB 4$

Question 5

Code:

q5.cpp

```cpp
You, 1 second ago | 1 author (You)
#include <iostream>
#include <fstream>
#include <vector>
#include <sstream>
using namespace std;

You, 1 second ago | 1 author (You)
class MPNeuron {
public:
    MPNeuron(int numInputs) {
        weights.resize(numInputs, -1);  // Set all weights to -1
        threshold = 0; // 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 ? 1 : 0;
    }

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

q5.cpp

```
class MPNeuron {

int main(int argc, char* argv[]) {
    if (argc != 2) {
        cerr << "Usage: " << argv[0] << " <filename>" << endl;
        return 1;
    }

    string filename = argv[1];
    ifstream file(filename);
    if (!file) {
        cerr << "Error opening file!" << endl;
        return 1;
    }

    int numIterations;
    file >> numIterations; // Read the number of iterations
    file.ignore();

    for (int i = 0; i < numIterations; ++i) {
        cout<<"Iteration "<<i+1<<endl;
        int numInputs;
        file >> numInputs;  // Read the number of inputs for this iteration
        file.ignore();

        vector<int> inputs(numInputs);
        for (int j = 0; j < numInputs; ++j) {
            file >> inputs[j];  // inputs
        }

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

        cout << "For input ";
        for (int j = 0; j < numInputs; ++j) {
            cout << inputs[j] << " ";
        }
        cout << ", the output of NOR gate is: " << output << endl;
        cout<<endl;
    }

    return 0;
}
```

Result:

```
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 input.txt
 Iteration 1
 For input 1 0 1 , the output of NOR gate is: 0

 Iteration 2
 For input 1 1 1 , the output of NOR gate is: 0

 Iteration 3
 For input 0 0 0 , the output of NOR gate is: 1

 Iteration 4
 For input 1 1 1 1 1 , the output of NOR gate is: 0

 Iteration 5
 For input 0 0 0 0 0 , the output of NOR gate is: 1

 Iteration 6
 For input 1 0 1 0 0 , the output of NOR gate is: 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