

ML perceptron (P1)

v0.1

Generated by Doxygen 1.13.2

1 ML-perceptron	1
1.1 Student	1
1.2 Introduction	1
1.3 Installing	1
2 Class Index	3
2.1 Class List	3
3 File Index	5
3.1 File List	5
4 Class Documentation	7
4.1 halfAdder Class Reference	7
4.2 halfAdderOutput Struct Reference	7
4.3 Perceptron Class Reference	7
4.3.1 Detailed Description	8
4.3.2 Constructor & Destructor Documentation	8
4.3.2.1 Perceptron()	8
4.3.3 Member Function Documentation	8
4.3.3.1 __str__()	8
4.3.3.2 predict()	8
4.3.3.3 train()	9
4.4 PerceptronLayer Class Reference	9
4.4.1 Detailed Description	9
4.4.2 Constructor & Destructor Documentation	9
4.4.2.1 PerceptronLayer()	9
4.4.3 Member Function Documentation	10
4.4.3.1 __str__()	10
4.4.3.2 feedForward()	10
4.5 PerceptronNetwork Class Reference	10
4.5.1 Detailed Description	11
4.5.2 Constructor & Destructor Documentation	11
4.5.2.1 PerceptronNetwork()	11
4.5.3 Member Function Documentation	11
4.5.3.1 __str__()	11
4.5.3.2 feedForward()	11
5 File Documentation	13
5.1 halfAdder.hpp	13
5.2 /Users/stanislav/Github/MachineLearning/ML-Perceptron/src/header/perceptron.hpp File Reference	14
5.2.1 Detailed Description	14
5.3 perceptron.hpp	14
5.4 /Users/stanislav/Github/MachineLearning/ML-Perceptron/src/header/perceptronLayer.hpp File Reference	15

5.4.1 Detailed Description	15
5.5 perceptronLayer.hpp	15
5.6 /Users/stanislav/Github/MachineLearning/ML-Perceptron/src/header/perceptronNetwork.hpp File Reference	16
5.6.1 Detailed Description	16
5.7 perceptronNetwork.hpp	16
5.8 /Users/stanislav/Github/MachineLearning/ML-Perceptron/src/test/test.cpp File Reference	17
5.8.1 Detailed Description	18
5.8.2 Function Documentation	18
5.8.2.1 TEST_CASE()	18
Index	19

Chapter 1

ML-perceptron

1.1 Student

Name: Stan Merlijn

Student nummer: 1863967

1.2 Introduction

In this repo we are going to implement and test perceptrons, perceptron layers and a perceptron networks(neural network). Theset are going to be tested by creating AND, OR, INVERT, NAND, XOR and half adder logic gates. the reader can be found [here](#)

1.3 Installing

Enter the test dir then

Generate build files:

```
cmake -S . -B build
```

Build the project:

```
cmake --build build
```

Run the executable:

```
./build/MLPerceptronTest
```


Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

halfAdder	7
halfAdderOutput	7
Perceptron	
A simple perceptron model for binary classification	7
PerceptronLayer	
Represents a layer of perceptrons in a neural network	9
PerceptronNetwork	
Represents a multi-layer perceptron network	10

Chapter 3

File Index

3.1 File List

Here is a list of all documented files with brief descriptions:

/Users/stanislav/Github/MachineLearning/ML-Perceptron/src/header/ halfAdder.hpp	13
/Users/stanislav/Github/MachineLearning/ML-Perceptron/src/header/ perceptron.hpp In this file the Perceptron class is defined	14
/Users/stanislav/Github/MachineLearning/ML-Perceptron/src/header/ perceptronLayer.hpp In this file the PerceptronLayer class is defined	15
/Users/stanislav/Github/MachineLearning/ML-Perceptron/src/header/ perceptronNetwork.hpp In this file the PerceptronNetwork class is defined	16
/Users/stanislav/Github/MachineLearning/ML-Perceptron/src/test/ test.cpp In this file the test cases for the Perceptron , PerceptronLayer and PerceptronNetwork classes are defined	17

Chapter 4

Class Documentation

4.1 halfAdder Class Reference

Public Member Functions

- [halfAdderOutput](#) **predict** (const std::vector< int > &x) const
- void **__str__** (int verbose) const

The documentation for this class was generated from the following file:

- /Users/stanislav/Github/MachineLearning/ML-Perceptron/src/header/halfAdder.hpp

4.2 halfAdderOutput Struct Reference

Public Attributes

- int **sum**
- int **carry**

The documentation for this struct was generated from the following file:

- /Users/stanislav/Github/MachineLearning/ML-Perceptron/src/header/halfAdder.hpp

4.3 Perceptron Class Reference

A simple perceptron model for binary classification.

```
#include <perceptron.hpp>
```

Public Member Functions

- [Perceptron](#) (std::vector< double > weights, double bias, double learningRate)
Constructs a [Perceptron](#) with given weights, bias, and learning rate.
- int [predict](#) (const std::vector< int > &inputs) const
Predicts the output for a given input vector.
- void [train](#) (const std::vector< std::vector< int > > &inputs, const std::vector< int > &targets, int epochs)
Trains the perceptron using the given dataset. Using th learning rule to update the weights.
- void [__str__](#) (int verbose) const
Prints perceptron details.

4.3.1 Detailed Description

A simple perceptron model for binary classification.

4.3.2 Constructor & Destructor Documentation

4.3.2.1 Perceptron()

```
Perceptron::Perceptron (
    std::vector< double > weights,
    double bias,
    double learningRate)
```

Constructs a [Perceptron](#) with given weights, bias, and learning rate.

Parameters

<i>weights</i>	Initial weights.
<i>bias</i>	Initial bias.
<i>learningRate</i>	Learning rate for training.

4.3.3 Member Function Documentation

4.3.3.1 __str__()

```
void Perceptron::__str__ (
    int verbose) const
```

Prints perceptron details.

Parameters

<i>verbose</i>	Verbosity level.
----------------	------------------

4.3.3.2 predict()

```
int Perceptron::predict (
    const std::vector< int > & inputs) const
```

Predicts the output for a given input vector.

Parameters

<i>inputs</i>	Input vector.
---------------	---------------

Returns

1 if activated, otherwise 0.

4.3.3.3 train()

```
void Perceptron::train (
    const std::vector< std::vector< int > > & inputs,
    const std::vector< int > & targets,
    int epochs)
```

Trains the perceptron using the given dataset. Using th learning rule to update the weights.

Parameters

<i>inputs</i>	Input samples.
<i>targets</i>	Target outputs.
<i>epochs</i>	Number of training iterations.

The documentation for this class was generated from the following files:

- /Users/stanislav/Github/MachineLearning/ML-Perceptron/src/header/[perceptron.hpp](#)
- /Users/stanislav/Github/MachineLearning/ML-Perceptron/src/perceptron.cpp

4.4 PerceptronLayer Class Reference

Represents a layer of perceptrons in a neural network.

```
#include <perceptronLayer.hpp>
```

Public Member Functions

- [PerceptronLayer](#) (const std::vector< [Perceptron](#) > &neurons)
Constructs a perceptron layer.
- std::vector< int > [feedForward](#) (const std::vector< int > &input) const
Feeds input forward through the layer.
- void [__str__](#) (int verbose) const
Prints layer details.

4.4.1 Detailed Description

Represents a layer of perceptrons in a neural network.

4.4.2 Constructor & Destructor Documentation

4.4.2.1 PerceptronLayer()

```
PerceptronLayer::PerceptronLayer (
    const std::vector< Perceptron > & neurons)
```

Constructs a perceptron layer.

Parameters

<i>neurons</i>	List of perceptrons.
----------------	----------------------

4.4.3 Member Function Documentation

4.4.3.1 `__str__()`

```
void PerceptronLayer::__str__ (
    int verbose) const
```

Prints layer details.

Parameters

<i>verbose</i>	Verbosity level.
----------------	------------------

4.4.3.2 `feedForward()`

```
std::vector< int > PerceptronLayer::feedForward (
    const std::vector< int > & input) const
```

Feeds input forward through the layer.

Parameters

<i>input</i>	Input vector.
--------------	---------------

Returns

Output vector after applying all perceptrons.

The documentation for this class was generated from the following files:

- /Users/stanislav/Github/MachineLearning/ML-Perceptron/src/header/[perceptronLayer.hpp](#)
- /Users/stanislav/Github/MachineLearning/ML-Perceptron/src/perceptronLayer.cpp

4.5 PerceptronNetwork Class Reference

Represents a multi-layer perceptron network.

```
#include <perceptronNetwork.hpp>
```

Public Member Functions

- [PerceptronNetwork](#) (std::vector< [PerceptronLayer](#) > layers)
Constructs a perceptron network.
- std::vector< int > [feedForward](#) (const std::vector< int > &input) const
Feeds input forward through the network.
- void [__str__](#) (int verbose) const
Prints network details.

4.5.1 Detailed Description

Represents a multi-layer perceptron network.

4.5.2 Constructor & Destructor Documentation

4.5.2.1 PerceptronNetwork()

```
PerceptronNetwork::PerceptronNetwork (
    std::vector< PerceptronLayer > layers)
```

Constructs a perceptron network.

Parameters

<i>layers</i>	List of perceptron layers.
---------------	----------------------------

4.5.3 Member Function Documentation

4.5.3.1 __str__()

```
void PerceptronNetwork::__str__ (
    int verbose) const
```

Prints network details.

Parameters

<i>verbose</i>	Verbosity level.
----------------	------------------

4.5.3.2 feedForward()

```
std::vector< int > PerceptronNetwork::feedForward (
    const std::vector< int > & input) const
```

Feeds input forward through the network.

Parameters

<i>input</i>	Input vector.
--------------	---------------

Returns

Output vector after processing through all layers.

The documentation for this class was generated from the following files:

- /Users/stanislav/Github/MachineLearning/ML-Perceptron/src/header/[perceptronNetwork.hpp](#)
- /Users/stanislav/Github/MachineLearning/ML-Perceptron/src/perceptronNetwork.cpp

Chapter 5

File Documentation

5.1 halfAdder.hpp

```
00001 #include "perceptron.hpp"
00002 #include <iostream>
00003 #include <vector>
00004
00005 #define EPOCHS 100
00006
00007 struct halfAdderOutput
00008 {
00009     int sum;
00010     int carry;
00011 };
00012
00013 class halfAdder
00014 {
00015 private:
00016     // Perceptrons for the half adder
00017     Perceptron andGate;
00018     Perceptron orGate;
00019     Perceptron nanGate;
00020
00021 public:
00022     halfAdder()
00023         : andGate({0.1, 0.1}, 1, 0.1),
00024           orGate({0.1, 0.1}, 1, 0.1),
00025           nanGate({0.1, 0.1}, 1, 0.1)
00026     {
00027         // Training data for the half adder
00028         std::vector<std::vector<int>> inputs = {{0, 0}, {0, 1}, {1, 0}, {1, 1}};
00029         std::vector<int> targetsAnd = {0, 0, 0, 1};
00030         std::vector<int> targetsOr = {0, 1, 1, 1};
00031         std::vector<int> targetsNand = {1, 1, 1, 0};
00032
00033         // Train the perceptrons
00034         andGate.train(inputs, targetsAnd, EPOCHS);
00035         orGate.train(inputs, targetsOr, EPOCHS);
00036         nanGate.train(inputs, targetsNand, EPOCHS);
00037     }
00038
00039     halfAdderOutput predict(const std::vector<int>& x) const
00040     {
00041         halfAdderOutput output;
00042         output.sum = andGate.predict({orGate.predict(x), nanGate.predict(x)});
00043         output.carry = andGate.predict(x);
00044         return output;
00045     }
00046
00047     void __str__(int verbose) const
00048     {
00049         std::cout << "Half Adder Structure:" << std::endl;
00050         std::cout << "AND Gate: ";
00051         andGate.__str__(verbose);
00052         std::cout << "OR Gate: ";
00053         orGate.__str__(verbose);
00054         std::cout << "NAND Gate: ";
00055         nanGate.__str__(verbose);
00056     }
00057 };
00058
00059 };
```

5.2 /Users/stanislav/Github/MachineLearning/ML- Perceptron/src/header/perceptron.hpp File Reference

In this file the [Perceptron](#) class is defined.

```
#include <iostream>
#include <vector>
```

Classes

- class [Perceptron](#)
A simple perceptron model for binary classification.

5.2.1 Detailed Description

In this file the [Perceptron](#) class is defined.

Author

Stan Merlijn

Version

0.1

Date

2025-02-12

Copyright

Copyright (c) 2025

5.3 perceptron.hpp

[Go to the documentation of this file.](#)

```
00001
00011
00012 #pragma once
00013 #include <iostream>
00014 #include <vector>
00015
00020 class Perceptron
00021 {
00022 public:
00029     Perceptron(std::vector<double> weights, double bias, double learningRate);
00030
00036     int predict(const std::vector<int>& inputs) const;
00037
00044     void train(const std::vector<std::vector<int>& inputs, const std::vector<int>& targets, int
epochs);
00045
00050     void __str__(int verbose) const;
00051
00052 private:
00053     std::vector<double> weights;
00054     double bias;
00055     double learningRate;
00056 };
```

5.4 /Users/stanislav/Github/MachineLearning/ML-Perceptron/src/header/perceptronLayer.hpp File Reference

In this file the [PerceptronLayer](#) class is defined.

```
#include "perceptron.hpp"
#include <iostream>
#include <vector>
```

Classes

- class [PerceptronLayer](#)
Represents a layer of perceptrons in a neural network.

5.4.1 Detailed Description

In this file the [PerceptronLayer](#) class is defined.

Author

Stan Merlijn

Version

0.1

Date

2025-02-12

Copyright

Copyright (c) 2025

5.5 perceptronLayer.hpp

[Go to the documentation of this file.](#)

```
00001
00011 #pragma once
00012 #include "perceptron.hpp"
00013 #include <iostream>
00014 #include <vector>
00015
00020 class PerceptronLayer
00021 {
00022 public:
00027     PerceptronLayer(const std::vector<Perceptron>& neurons);
00028
00034     std::vector<int> feedForward(const std::vector<int>& input) const;
00035
00040     void __str__(int verbose) const;
00041
00042 private:
00043     std::vector<Perceptron> neurons;
00044 };
```

5.6 /Users/stanislav/Github/MachineLearning/ML-[↩](#) Perceptron/src/header/perceptronNetwork.hpp File Reference

In this file the [PerceptronNetwork](#) class is defined.

```
#include "perceptronLayer.hpp"
#include <iostream>
#include <vector>
```

Classes

- class [PerceptronNetwork](#)
Represents a multi-layer perceptron network.

5.6.1 Detailed Description

In this file the [PerceptronNetwork](#) class is defined.

Author

Stan Merlijn

Version

0.1

Date

2025-02-12

Copyright

Copyright (c) 2025

5.7 perceptronNetwork.hpp

[Go to the documentation of this file.](#)

```
00001
00011 #pragma once
00012 #include "perceptronLayer.hpp"
00013 #include <iostream>
00014 #include <vector>
00015
00020 class PerceptronNetwork
00021 {
00022 public:
00027     PerceptronNetwork(std::vector<PerceptronLayer> layers);
00028
00034     std::vector<int> feedForward(const std::vector<int>& input) const;
00035
00040     void __str__(int verbose) const;
00041
00042 private:
00043     std::vector<PerceptronLayer> layers;
00044
00045 };
```

5.8 /Users/stanislaw/Github/MachineLearning/ML-Perceptron/src/test/test.cpp File Reference

In this file the test cases for the [Perceptron](#), [PerceptronLayer](#) and [PerceptronNetwork](#) classes are defined.

```
#include "catch.hpp"
#include "../header/perceptron.hpp"
#include "../header/perceptronLayer.hpp"
#include "../header/perceptronNetwork.hpp"
#include "../header/halfAdder.hpp"
#include <iostream>
```

Macros

- `#define CATCH_CONFIG_MAIN`
- `#define EPOCHS 100`

Functions

- **TEST_CASE** ("Perceptron for INVERT Gate", "[perceptron]")
Perceptron for INVERT Gate: Tests the perceptron's ability to learn the INVERT gate.
- **TEST_CASE** ("Perceptron for AND Gate", "[perceptron]")
Perceptron for AND Gate: Tests the perceptron's ability to learn the AND gate.
- **TEST_CASE** ("Perceptron for OR Gate", "[perceptron]")
Perceptron for OR Gate: Tests the perceptron's ability to learn the OR gate.
- **TEST_CASE** ("Perceptron for NOR Gate (3 inputs)", "[perceptron]")
Perceptron for NOR Gate (3 inputs): Tests the perceptron's ability to learn the NOR gate with 3 inputs. The NOR gate is a digital logic gate that implements logical NOR
- **TEST_CASE** ("Perceptron for 3-input Majority Gate", "[perceptron]")
Perceptron for 3-input Majority Gate: Tests the perceptron's ability to learn the 3-input Majority gate.
- **TEST_CASE** ("PerceptronLayer for AND and OR Gates", "[perceptronLayer]")
PerceptronLayer for AND and OR Gates: Tests the PerceptronLayer's ability to learn the AND and OR gates. It contains two perceptrons: one for the AND gate and one for the OR gate.
- **TEST_CASE** ("PerceptronNetwork for the XOR gate with 2 inputs", "[perceptronNetwork]")
PerceptronNetwork for the XOR gate with 2 inputs. This network contains two layers: inputLayer for the AND gate and one for the OR gate. outputLayer for the AND gate.
- **TEST_CASE** ("PerceptronNetwork for half adder", "[perceptronNetwork]")
PerceptronNetwork for a half adder. This network contains two layers: hiddenLayer for the OR and AND gates. outputLayer for the XOR gate(sum) and the carry.

Variables

- `std::vector< std::vector< int > > inputs = {{0, 0}, {0, 1}, {1, 0}, {1, 1}}`

5.8.1 Detailed Description

In this file the test cases for the [Perceptron](#), [PerceptronLayer](#) and [PerceptronNetwork](#) classes are defined.

Unit tests for the [Perceptron](#), [PerceptronLayer](#) and [PerceptronNetwork](#) classes.

Author

Stan Merlijn

Version

0.1

Date

2025-02-12

Copyright

Copyright (c) 2025

This file contains a series of test cases to verify the functionality of the [Perceptron](#) and [PerceptronLayer](#) classes. The tests include training and prediction for various logic gates.

Test Cases:

- [Perceptron](#) for INVERT Gate: Tests the perceptron's ability to learn the INVERT gate.
- [Perceptron](#) for AND Gate: Tests the perceptron's ability to learn the AND gate.
- [Perceptron](#) for OR Gate: Tests the perceptron's ability to learn the OR gate.
- [Perceptron](#) for NOR Gate (3 inputs): Tests the perceptron's ability to learn the NOR gate with 3 inputs.
- [Perceptron](#) for 3-input Majority Gate: Tests the perceptron's ability to learn the 3-input Majority gate.
- [PerceptronLayer](#) for AND and OR Gates: Tests the [PerceptronLayer](#)'s ability to learn the AND and OR gates.
- [PerceptronNetwork](#) for the XOR gate with 2 inputs.
- [PerceptronNetwork](#) for a half adder.

Note

The tests use the Catch2 framework for unit testing.

5.8.2 Function Documentation

5.8.2.1 TEST_CASE()

```
TEST_CASE (
    "Perceptron for NOR Gate (3 inputs)" ,
    "" [perceptron])
```

[Perceptron](#) for NOR Gate (3 inputs): Tests the perceptron's ability to learn the NOR gate with 3 inputs. The NOR gate is a digital logic gate that implements logical NOR

0, 0, 0, 0, 0, 0, 0, 1

Index

- [/Users/stanislav/Github/MachineLearning/ML-Perceptron/src/header/halfAdder.hpp,](#)
[13](#)
- [/Users/stanislav/Github/MachineLearning/ML-Perceptron/src/header/perceptron.hpp,](#)
[14](#)
- [/Users/stanislav/Github/MachineLearning/ML-Perceptron/src/header/perceptronLayer.hpp,](#)
[15](#)
- [/Users/stanislav/Github/MachineLearning/ML-Perceptron/src/header/perceptronNetwork.hpp,](#)
[16](#)
- [/Users/stanislav/Github/MachineLearning/ML-Perceptron/src/test/test.cpp,](#)
[17](#)
- [__str__](#)
 - [Perceptron, 8](#)
 - [PerceptronLayer, 10](#)
 - [PerceptronNetwork, 11](#)
- [feedForward](#)
 - [PerceptronLayer, 10](#)
 - [PerceptronNetwork, 11](#)
- [halfAdder, 7](#)
- [halfAdderOutput, 7](#)
- [ML-perceptron, 1](#)
- [Perceptron, 7](#)
 - [__str__, 8](#)
 - [Perceptron, 8](#)
 - [predict, 8](#)
 - [train, 9](#)
- [PerceptronLayer, 9](#)
 - [__str__, 10](#)
 - [feedForward, 10](#)
 - [PerceptronLayer, 9](#)
- [PerceptronNetwork, 10](#)
 - [__str__, 11](#)
 - [feedForward, 11](#)
 - [PerceptronNetwork, 11](#)
- [predict](#)
 - [Perceptron, 8](#)
- [test.cpp](#)
 - [TEST_CASE, 18](#)
- [TEST_CASE](#)
 - [test.cpp, 18](#)
- [train](#)
 - [Perceptron, 9](#)