**Podstawy sztucznej inteligencji**

Kinga Synowiec, Inżynieria obliczeniowa, grupa 2

Sprawozdanie nr 1

1. **Temat ćwiczenia:**

Budowa i działanie perceptronu

1. **Cel ćwiczenia:**

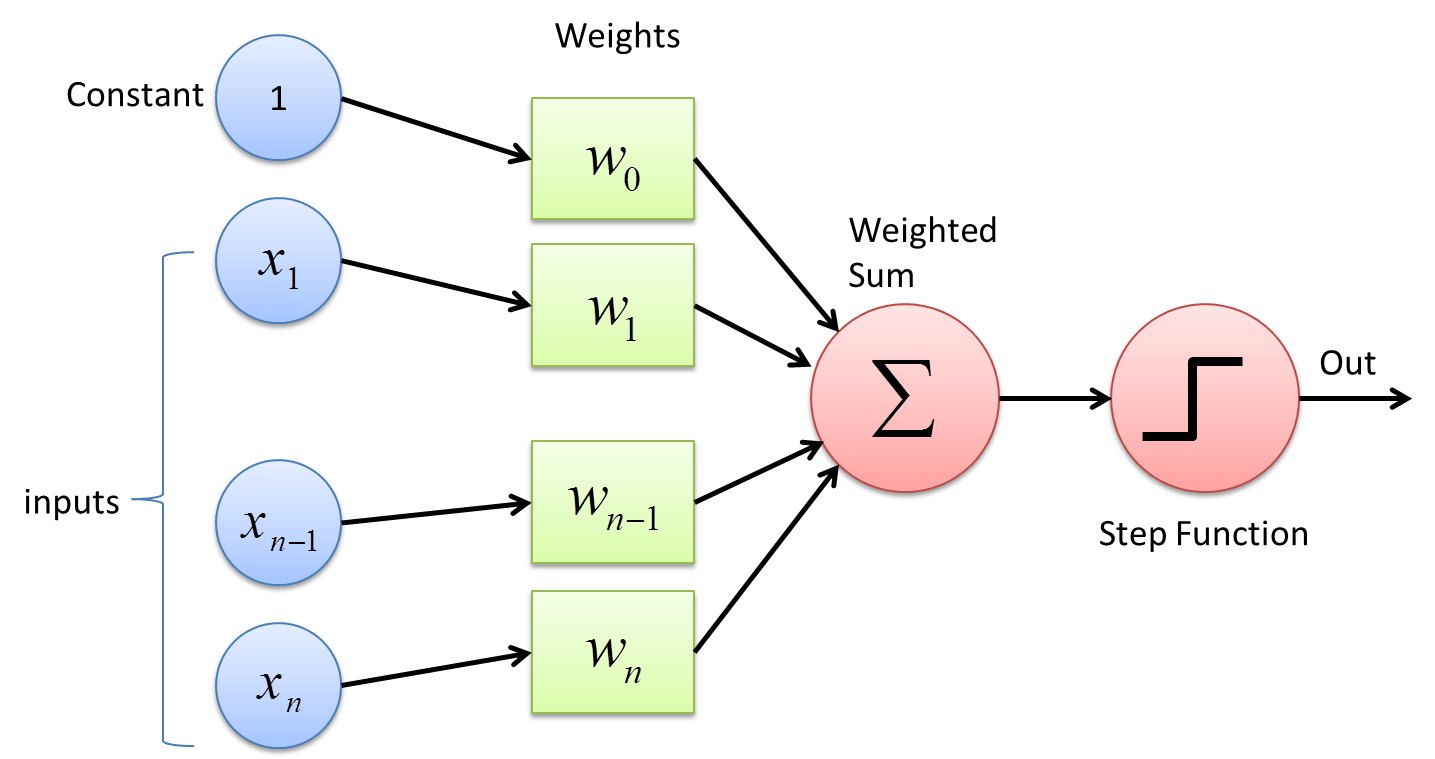
Poznanie budowy i działania perceptronu poprzez implementację oraz uczenie perceptronu realizującego wybraną funkcję logiczną dwóch zmiennych.

1. **Wstęp teoretyczny:**

*Perceptron* - najprostsza sieć neuronowa, składająca się z jednego bądź wielu niezależnych neuronów McCullocha-Pittsa, implementująca algorytm uczenia nadzorowanego klasyfikatorów binarnych. Perceptron jest funkcją, która potrafi określić przynależność parametrów wejściowych do jednej z dwóch klas. Może być wykorzystywany tylko do klasyfikowania zbiorów liniowo separowalnych.

Działanie perceptronu polega na klasyfikowaniu danych pojawiających się na wejściu i ustawianiu stosownie do tego wartości wyjścia. Przed używaniem perceptron należy wytrenować, podając mu przykładowe dane na wejście i modyfikując w odpowiedni sposób wagi wejść i połączeń między warstwami neuronów, tak aby wynik na wyjściu przybierał pożądane wartości. Perceptrony mogą klasyfikować dane na zbiory, które są liniowo separowalne. Własność ta uniemożliwia na przykład wytrenowanie złożonego z jednego neuronu perceptronu, który wykonywałby logiczną operację XOR na wartościach wejść.

// *źródło :* [*https://pl.wikipedia.org/wiki/Perceptron*](https://pl.wikipedia.org/wiki/Perceptron)



1. **Kroki algorytmu z kodem wykonanym w programie MATLAB:**

* Zadeklarowanie parametrów, których używać będziemy przy uczeniu:

x = [1 0 1 1; 0 1 0 1];

t = [0 0 0 1];

* Utworzenie perceptronu poprzez funkcję dostępną w pakiecie MATLAB:

net = perceptron;

* Ustawienie parametrów oraz ilości iteracji wykorzystując kolejną funkcję dostępną w programie:

net.trainParam.epochs=15;

* Przeprowadzenie algorytmu uczenia perceptronu (w screenach zamieszczonych poniżej przykładowe wykresy powstałe przy zastosowaniu różnych parametrów opisanych w pierwszym podpunkcie):

net = train(net,x,t);

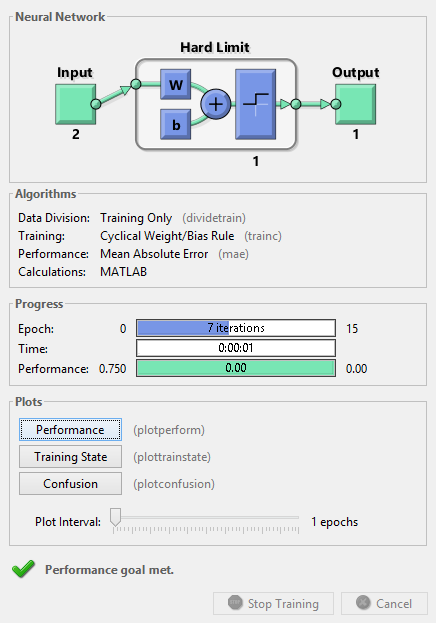
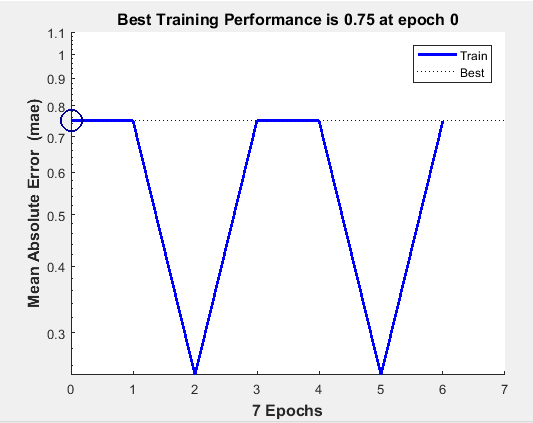
* Wykorzystanie narzędzia do przeprowadzania symulacji aby uzyskać wykres oraz wynikowe informacje o przeprowadzonym procesie (do funkcji w pierwszym argumencie przesyłamy model, w drugim parametry):

sym=sim(net,x)

*//źródła:* [*www.mathworks.com*](http://www.mathworks.com)*, dokumentacja programu MATLAB*

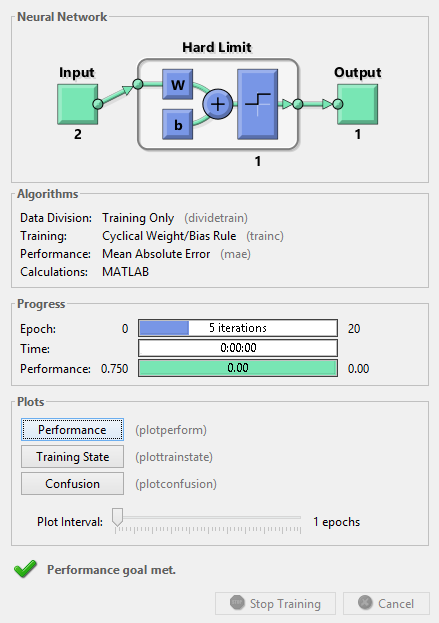
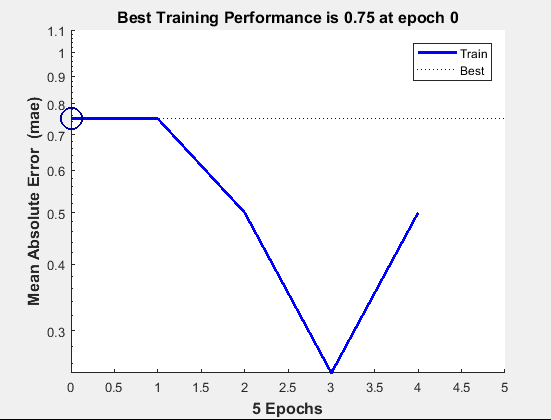
1. **Wyniki w postaci tabel i wykresów dla wybranych parametrów:**

* parametry jak w punkcie 6:

** **

* x=[0 0 1 1; 0 1 0 1]

t=[0 0 0 1];

1. **Listing całego kodu programu:**

%%

close all; clear all; clc;

x = [1 0 1 1; 0 1 0 1];

t = [0 0 0 1];

net = perceptron;

net.trainParam.epochs=15;

net = train(net,x,t);

sym=sim(net,x)

1. **Wnioski:**

* Przy użytych próbach dwukrotnie użyłam bramki logicznej AND.
* W pierwszej próbie użyłam 10 iteracji, w próbie drugiej 20.
* Wykresy, które zamieściłam we wcześniejszej części sprawozdania, pokazują wynik działania funkcji Mean Absolute Error(MAE).
* W pierwszym przypadku na wykresie można zauważyć cykliczność procesu uczenia, punkty zetknięcia wykresu z osią poziomą to 2 i 5.
* Na drugim wykresie różnica wartości nie jest schematyczna, błąd maleje stopniowo aż do trzeciej epoki, a następnie zaczyna rosnąć.
* Ilość wykonanych iteracji zależy od dobranych parametrów.