|  |  |  |
| --- | --- | --- |
| Grupa ćwicz. **2** | Data wykonania 11.10.2017 | Nr. Scenariusza  **1** |
| **Temat ćwiczenia:** Budowa i działanie perceptronu | | |
| Imię i nazwisko  **Kamil Szczurkowski** | | Ocena i Uwagi |

**Cel ćwiczenia:**

Celem ćwiczenia jest poznanie budowy i działanie perceptronu poprzez implementację oraz

uczenie perceptronu realizującego wybraną funkcję logiczną dwóch zmiennych.

**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 wartość na wyjściu przybierała 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.



Rysunek - Perceptron złożony z jednego neurona McCullocha-Pittsa

**Algorytm uczenia perceptronu**Jest to podstawowy algorytm uczenia.

1. Przypisujemy wagom i progowi małe losowe wartości wokół 0

2. Przebiegamy kolejno przykłady (albo jeden po drugim, albo (zalecane) losując z listy),

3. Dla wybranego przykładu E sprawdzamy, czy E jest dobrze klasyfikowany,

->Jeśli tak, to wracamy do 2

->Jeśli nie, to korygujemy wartości wag

Oraz progu

4. Wracamy do kroku 2

Iteracje kończymy, jeśli wszystkie przykłady są dobrze sklasyfikowane (w przypadku wyborów zgodnych z kolejnością odpowiada to przebiegnięciu całej listy bez wprowadzania zmian, w przypadku doboru losowego — odpowiednio długiej serii iteracji bez zmian wag).

**Zadania do wykonania**

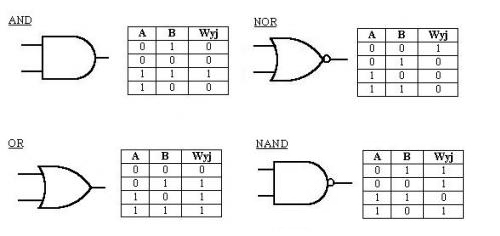
a) Implementacja sztucznego neuronu wg algorytmu podanego na wykładzie lub

dowolnego innego (z podaniem źródła).

Do zaimplementowania sztucznego neuronu użyje MATLABA oraz wykorzystam Neural Network jest to jeden z pakietów który się tam znajduje.

b) Wygenerowanie danych uczących i testujących wybranej funkcji logicznej dwóch

zmiennych.



Rysunek - Dane uczące

c) Uczenie perceptronu dla różnej liczby danych uczących, różnych współczynników

uczenia.

**1. bramka AND**

Wprowadzane dane do Matlaba:

net = newp([0 1;-2 2], 1);

A = [0 0 1 1];

B = [1 0 1 0];

P = [A;B];

T = [0 0 1 0 ];

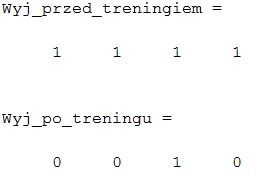
net = init(net);

Wyj\_przed\_treningiem = sim(net,P)

net = train(net,P,T);

Wyj\_po\_treningu = sim(net,P)

Linia komend:



**2. bramka NOR**

Wprowadzane dane do Matlaba:

net = newp([0 1;-2 2], 1);

A = [0 0 1 1];

B = [1 0 1 0];

P = [A;B];

T = [1 0 0 0 ];

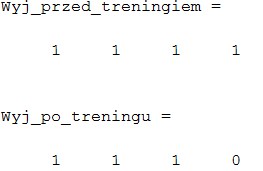
net = init(net);

Wyj\_przed\_treningiem = sim(net,P)

net = train(net,P,T);

Wyj\_po\_treningu = sim(net,P)

Linia komend:



**3. bramka OR**

Wprowadzane dane do Matlaba:

net = newp([0 1;-2 2], 1);

A = [0 0 1 1];

B = [1 0 1 0];

P = [A;B];

T = [0 1 1 1 ];

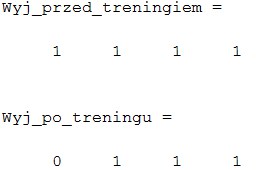
net = init(net);

Wyj\_przed\_treningiem = sim(net,P)

net = train(net,P,T);

Wyj\_po\_treningu = sim(net,P)

Linia komend:



**4. bramka NAND**

Wprowadzane dane do Matlaba:

net = newp([0 1;-2 2], 1);

A = [0 1 0 1];

B = [0 0 1 1];

P = [A;B];

T = [1 1 1 0 ];

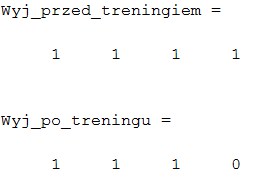
net = init(net);

Wyj = sim(net,P)

net = train(net,P,T);

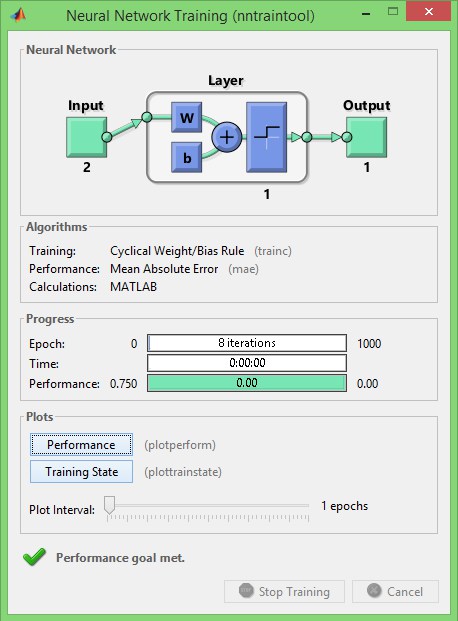
Wyj = sim(net,P)

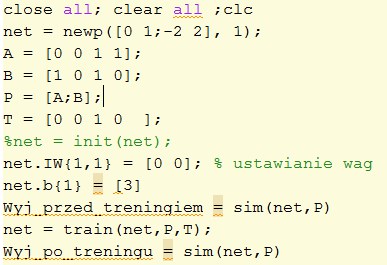
Linia komend:



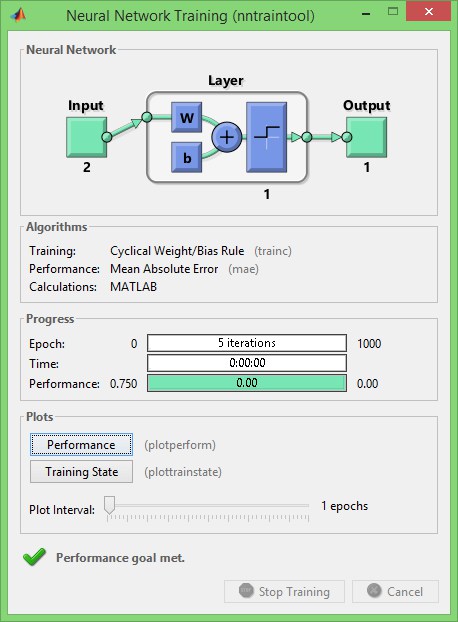
**Uczenie dla różnych współczynników uczenia:**

1.Zmodyfikowana bramka and, kilka przykładów:



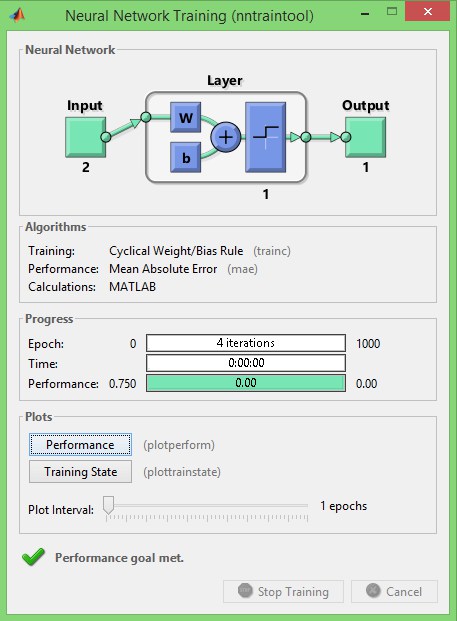


Rysunek - Kod z wagami Rysunek - Otrzymany wynik

****



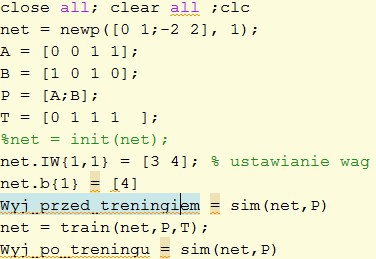
Rysunek - Wagi Rysunek - otrzymane wyniki

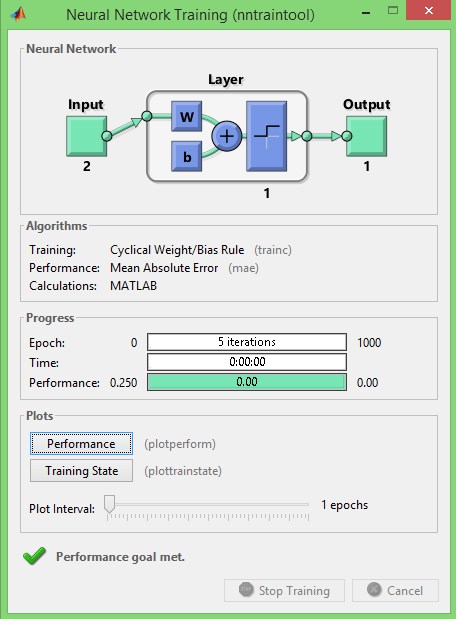




Rysunek - Wagi Rysunek - Otrzymane wyniki

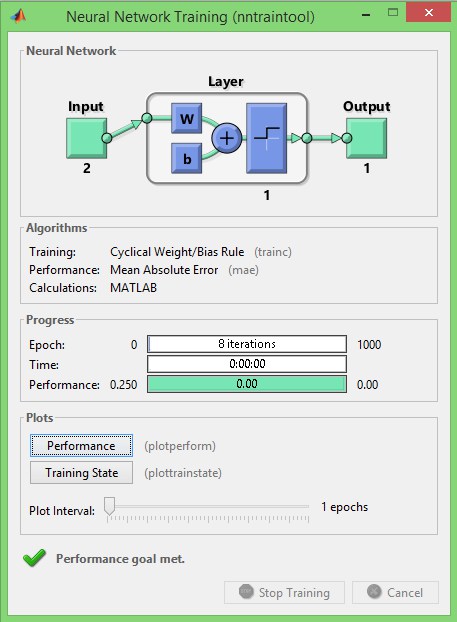
1.Zmodyfikowana bramka or, kilka przykładów:





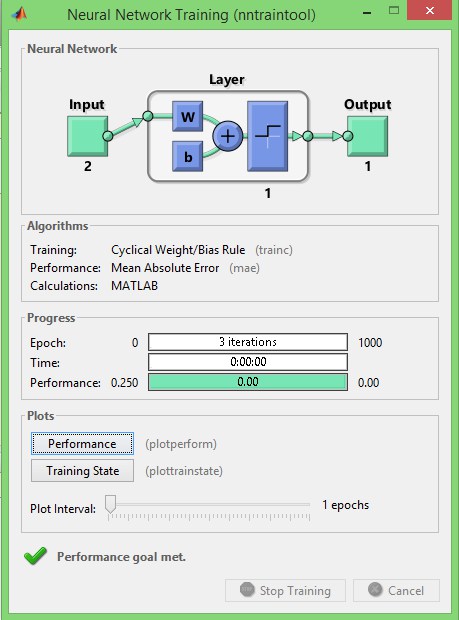
Rysunek - Otrzymane wyniki Rysunek Wagi

Screenshot 2017-10-16 22-20-23



Rysunek Otrzymane wyniki Rysunek Wagi

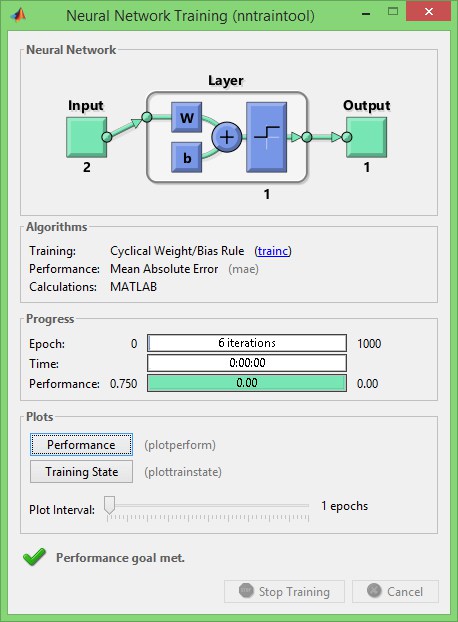
Screenshot 2017-10-16 22-23-24



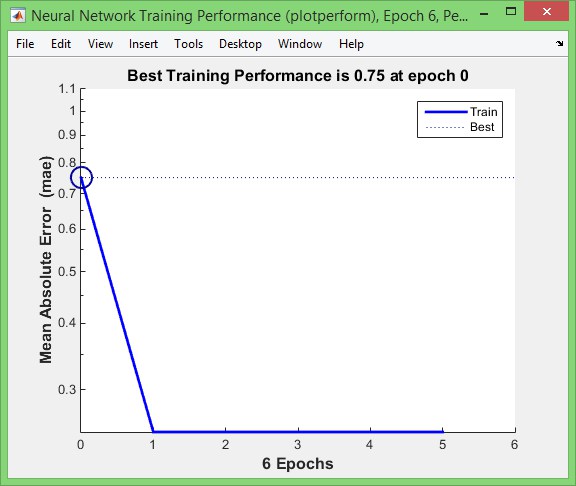
Rysunek Otrzymane wyniki Rysunek Wagi

d) Testowanie perceptronu.

**1. bramka AND**

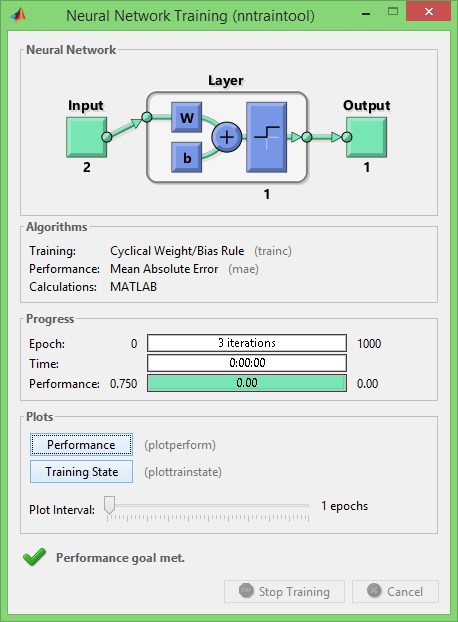


Rysunek - Trening - and

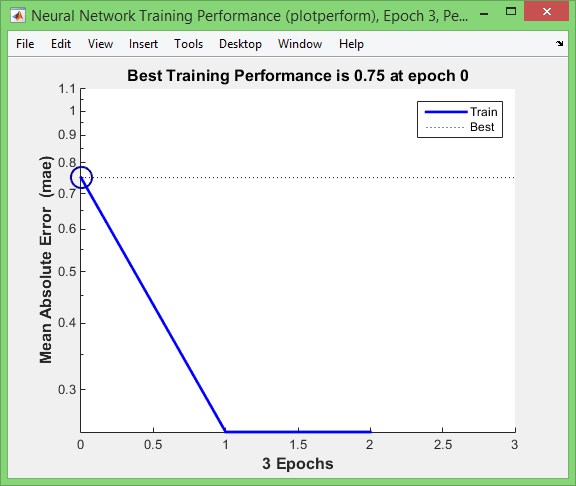


Rysunek - wykres uczenia - and

**2. bramka NOR**

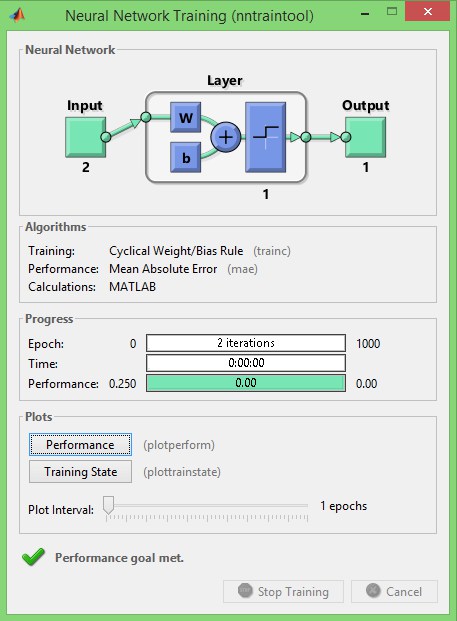


Rysunek - trening - nor

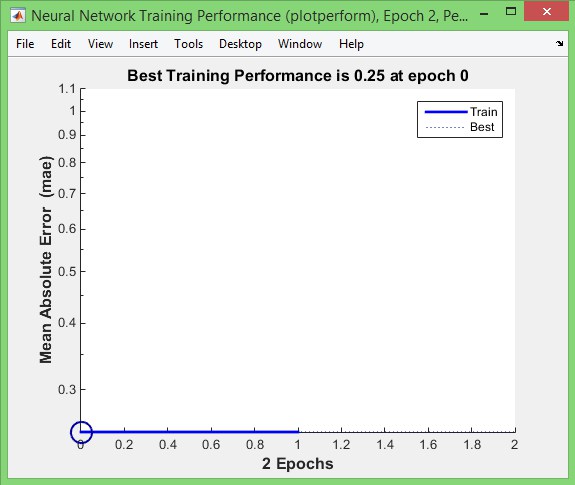


Rysunek - wykres uczenia - nor

**3. bramka OR**

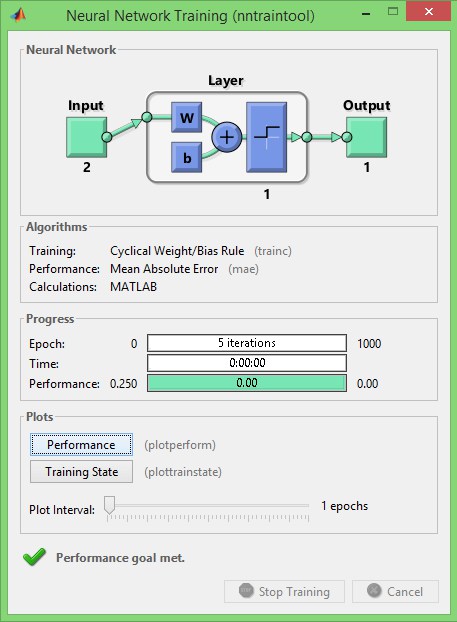


Rysunek - trening - or

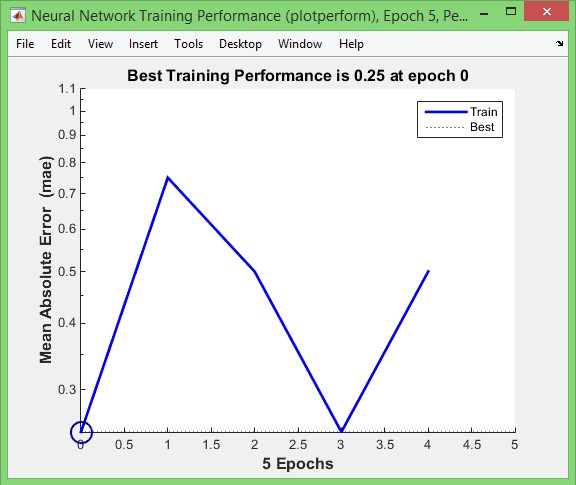


Rysunek - wykres uczenia - or

**4. bramka NAND**

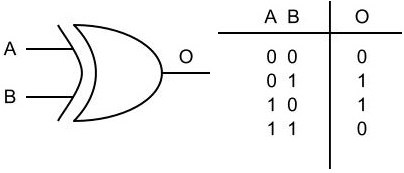


Rysunek - trening – nand



Rysunek - wykres uczenia – nand

e) Problem bramki xor



Wprowadzane dane do Matlaba:

net = newp([0 1;-2 2], 1);

P2 = [0 0 1 1;0 1 0 1];

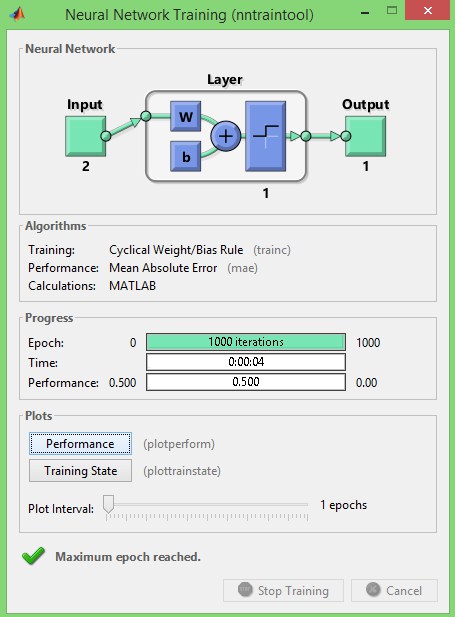
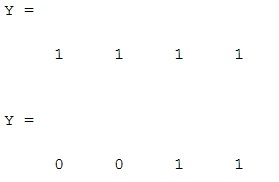
T2 = [1 0 0 1 ];

net = init(net);

Y = sim(net,P2)

net = train(net,P2,T2);

Y = sim(net,P2)



Uczenie zakończyło się niepowodzeniem.

Wnioski:  
Celem naszego ćwiczenia było poznanie budowy i działania perceptronu poprzez implementację oraz

uczenie perceptronu realizacji wybranej funkcji logicznej dwóch zmiennych. Badanie zostało przeprowadzone na bramkach and, or, nor, nand oraz xor. Patrząc na otrzymane wykresy w Matlabie można stwierdzić ze bardzo sprawnie poszła nauka gdyż po kilku iteracjach(epokach) otrzymaliśmy wynik. Podczas zmiany wag można było zaobserwować jak zmienia się stosunek nauki przy różnych wagach. Już na pierwszy zut oka można powiedzieć ze bardzo ciężko jest dobrać optymalne wagi do większych procesów. Jeśli chciałem przeprowadzić podobny zabieg na bramce xor to badania kończyły się niepowodzeniem ponieważ za pomocą jednego perceptronu nie jest możliwe nauczenie.

Bibliografia:

<http://www-users.mat.umk.pl/~piersaj/www/contents/teaching/wsn2013/wsn-notatki.pdf>  
<https://pl.wikipedia.org>