|  |  |  |
| --- | --- | --- |
| Grupa ćwicz. **2** | Data wykonania 19.10.2017 | Nr. Scenariusza  **1** |
| **Temat ćwiczenia:** Budowa i działanie perceptronu | | |
| Imię i nazwisko  **Andrzej Pawlikowski** | | Ocena i Uwagi |

# POJĘCIA

**Perceptronem -** nazywamy prosty element obliczeniowy, który sumuje ważone sygnały wejściowe i porównuje tę sumę z progiem aktywacji - w zależności od wyniku perceptron może być albo wzbudzony (wynik 1), albo nie (wynik 0).

**Algorytmu uczenia perceptronu** – polega na automatycznym doborze wag. Metoda uczenia perceptronu należy do grupy algorytmów zwanych uczeniem z nauczycielem lub uczeniem nadzorowanym. Uczenie tego typu polega, że na wyjściu perceptronu podaje się sygnały wejściowe dla których znamy prawidłowe wartości sygnałów wyjściowych, zwanych sygnałami wzorcowymi. Zbiór takich próbek wejściowych wraz z odpowiadającymi im wartościami sygnałów wzorcowych nazywamy ciągiem uczącym.

Podstawowym **algorytmem uczenia** perceptronu jest algorytm perceptronowy. Przedstawia się on następująco:

1. Przyjmij zerowy wektor wag [W] = [0];
2. Wybierz parę trenującą [Ek] i Ck; wybór ten może być kolejny ze wzrostem k od 1 do N, lub losowy, z uwzględnieniem wszystkich par trenujących
3. Jeżeli dla bieżącego wektora wag [W] następuje poprawa klasyfikacyjna, tzn. s>0, dla Ck = 1 lub s<0 dla Ck=-1, wówczas nie rób nic (wektor wag [W] pozostaje bez zmian) czyli [W\*]=[W];

inaczej: zmodyfikuj wektor wag [W] przez dodanie lub odjęcie od niego wektora trenującego [Ek] w zależności od tego czy: Ck=1 czy Ck=-1, tzn.:



gdzie funkcja signum sgn(s) opisana jest zależnością:



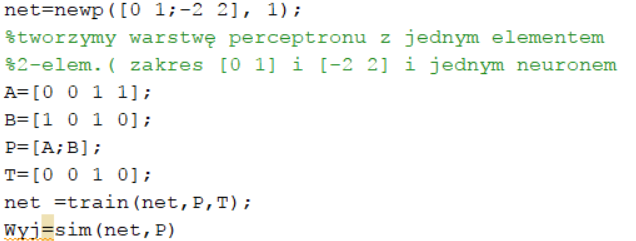
4. Dla s=0, zmień wagi według zależności:

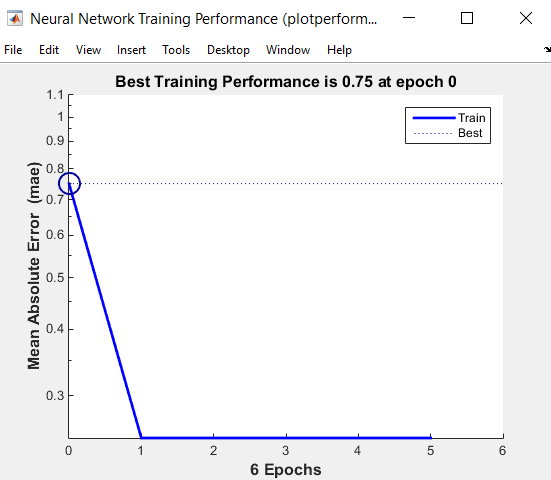
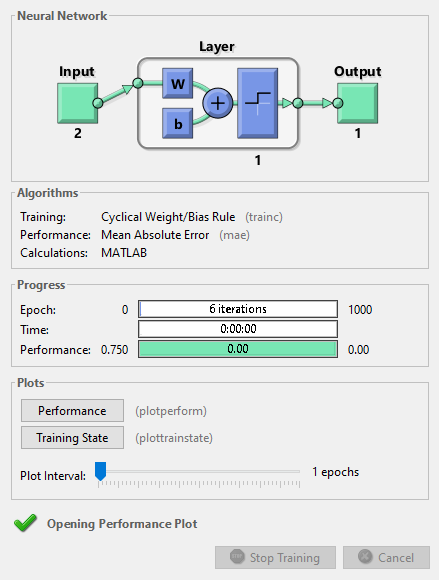


5. Id do kroku 2.

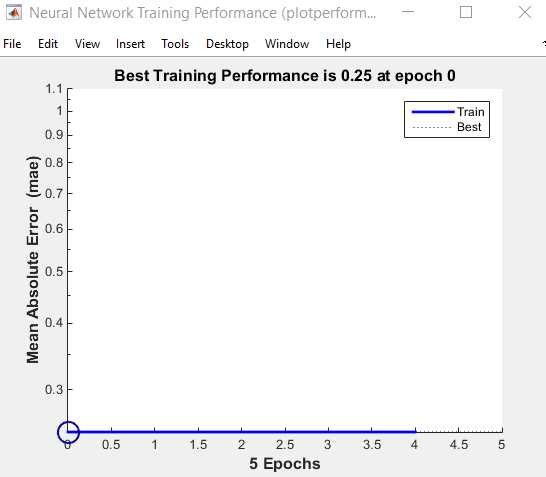
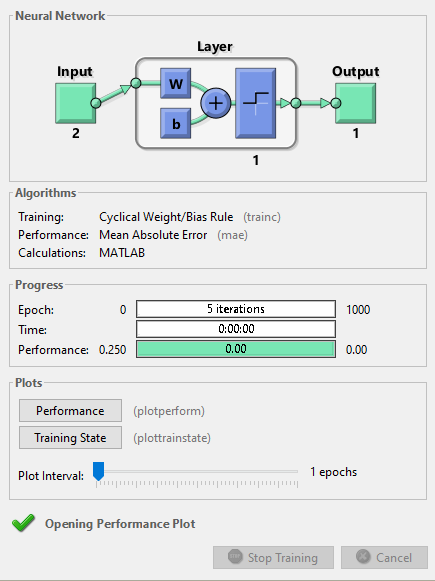
Algorytm z punktu 2 kończy pracę, gdy po podaniu kolejno wszystkich wektorów trenujących odpowiedź perceptronu jest prawidłow

**1.** Wykorzystanie algorytmu do uczenia bramki logicznej „and”

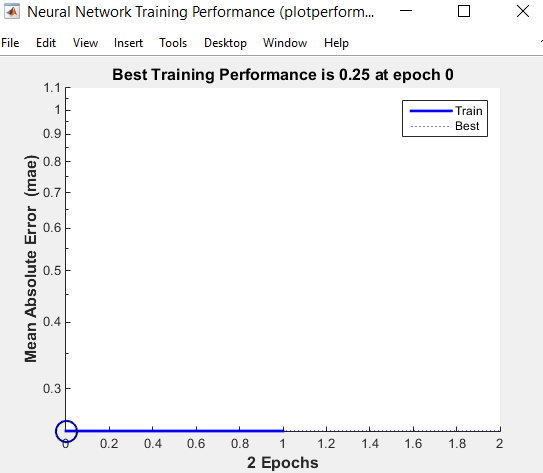
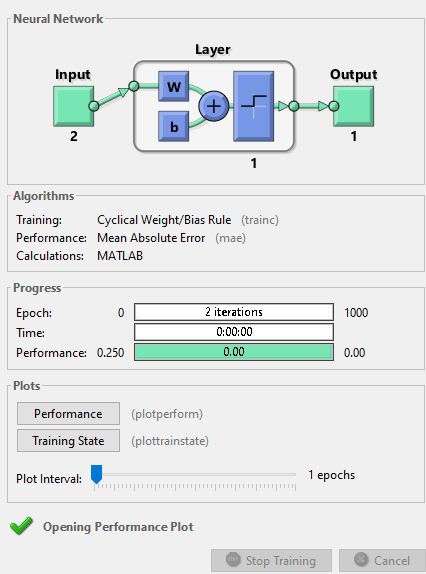




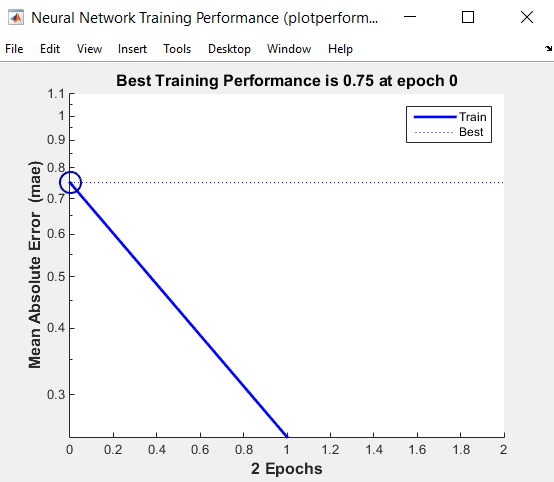
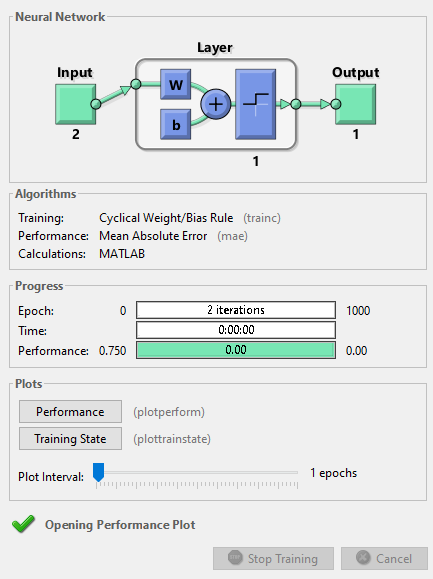
**2.** Wykorzystanie algorytmu do uczenia bramki logicznej „nand”



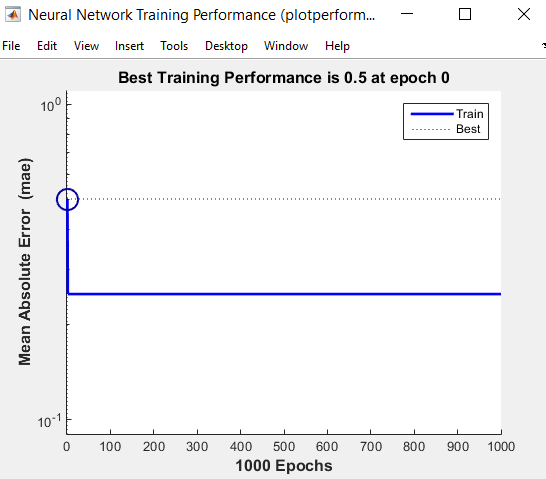
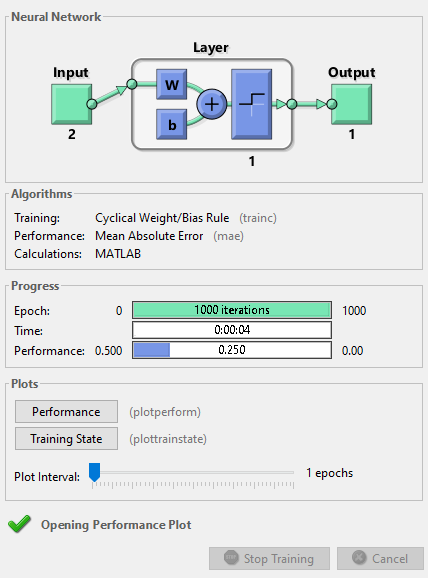
**3.** Wykorzystanie algorytmu do uczenia bramki logicznej „or”



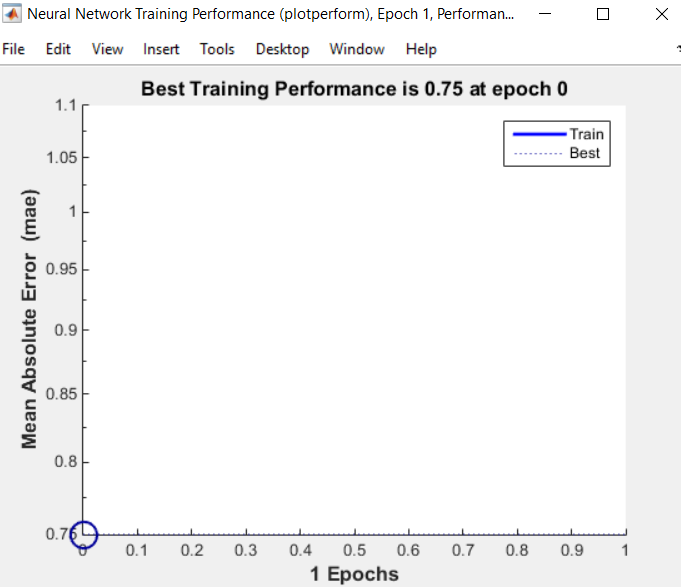
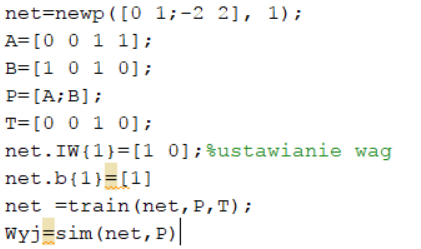
**4.** Wykorzystanie algorytmu do uczenia bramki logicznej „nor”

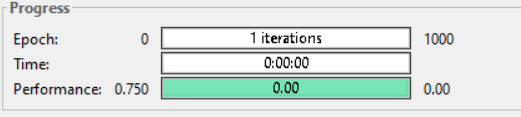


**5.** Wykorzystanie algorytmu do uczenia bramki logicznej „xor”

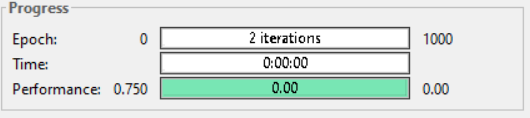


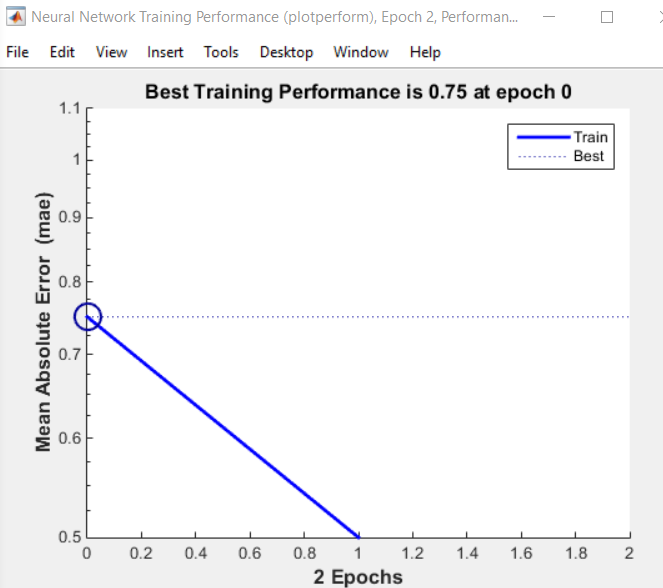
**7.** Algorytmu uczenia dla różnych współczynników wagowych





**8.** Zwiększenie wagi na:





**9.** Zwiekszenie wagi na:

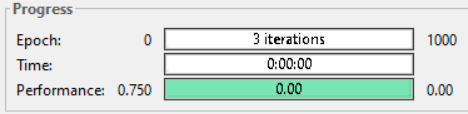
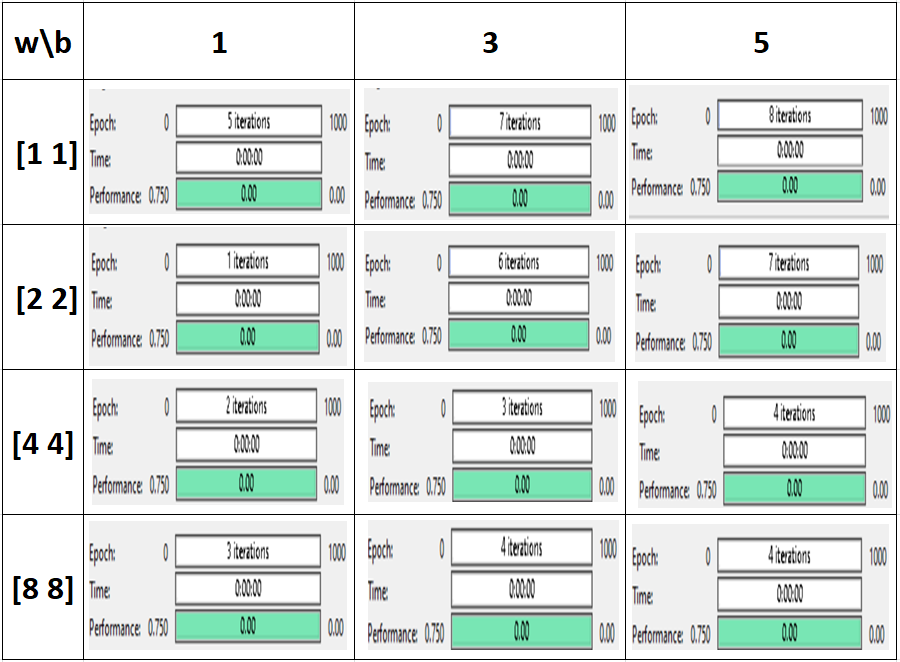




Tabela przedstawiająca wyniki pomiarów z rozróżnieniem parametrów wagowych i współczynnika b dla x wejściowego.



**Wnioski:**

Nauka bramek and, or, nor i nand zakończyła się powodzeniem już po kilku iteracjach(epokach). Problem z na bramce xor wynika z użycia w danym ćwiczeniu tylko jednego perceptronu.

Fundamentalne znaczenie w procesie trenowania sieci ma taki dobór

współczynników wagowych poszczególnych zmiennych wejściowych, aby błędy na wyjściu były jak najmniejsze.

Jeśli chodzi o błędy testowania to podczas żadnej z prób nie stwierdzono ani jednego błędu. Wynika to głównie z niezbyt skomplikowanej natury ćwiczenia.

**Literatura:**

<http://cens.ioc.ee/local/man/matlab/toolbox/nnet/>

<http://galera.ii.pw.edu.pl/~zsz/sieci_neuronowe/realizacja_sieci_neuronowych.pdf>

<http://www.neurosoft.edu.pl/media/pdf/jbartman/sztuczna_inteligencja/NTI%20Cwiczenie_2.pdf>

<https://www.mathworks.com>