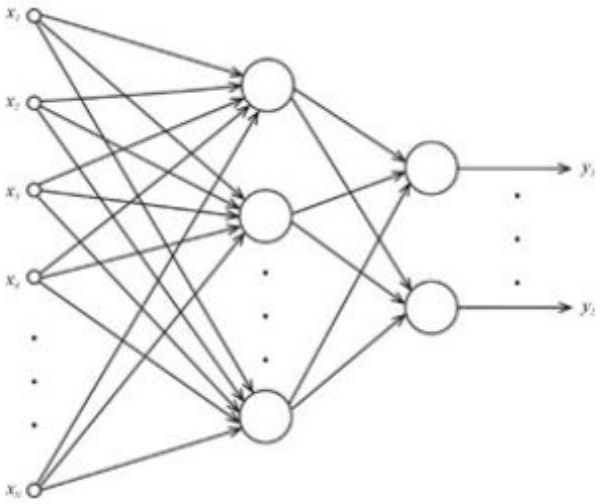
Maciej Stawarz Nr albumu 284310	Inżynieria Obliczeniowa Podstawy Sztucznej Inteligencji
Projekt 2 - Budowa i działanie jednowarstwowej sieci - Sprawozdanie	

1) Cel ćwiczenia:
   Celem poniższego zadania było poznanie sposobu posługiwania się obliczeniami wykonywanymi przez sieci neuronowe, na podstawie nauki rozpoznawania znaków.

2) Opis użytej struktury:
   Sztuczna sieć neuronowa - struktura matematyczno-programowa, przetwarzająca sygnały poprzez rzędy sztucznych neuronów, każdy wykonujący pewną podstawową operację. Sieć może realizować różne funkcje, najczęściej wzorowane na działaniu biologicznego mózgu.

   W tym przypadku używane są sieci jednokierunkowe, to znaczy takie w których nie istnieją połączenia pozwalające na sprzężenie zwrotne. Informacje są przesyłane od warstwy wejściowej, poprzez (jeśli sieć jest > 2 warstwowa) warstwy ukryte, do warstwy wyjściowej.



3) Opis użytych poleceń:
   newp(P,T,TF,LF) - funkcja tworząca perceptron
   Parametry:
   P - macierz o rozmiarze r x q wektorów wejściowych
   T - macierz o rozmiarze s x q wektorów oczekiwanych
   TF - funkcja aktywacji
   LF - funkcja trenowania
   Zwraca perceptron.

newlin(P,S,ID,LR) - funkcja tworząca sieć linową

Parametry:

P - macierz o rozmiarze r x q wektorów wejściowych

S - liczba elementów wektora wyjściowego

ID - wektor opóźnienia wejścia

LR - współczynnik uczenia

Zwraca nową warstwę liniową.

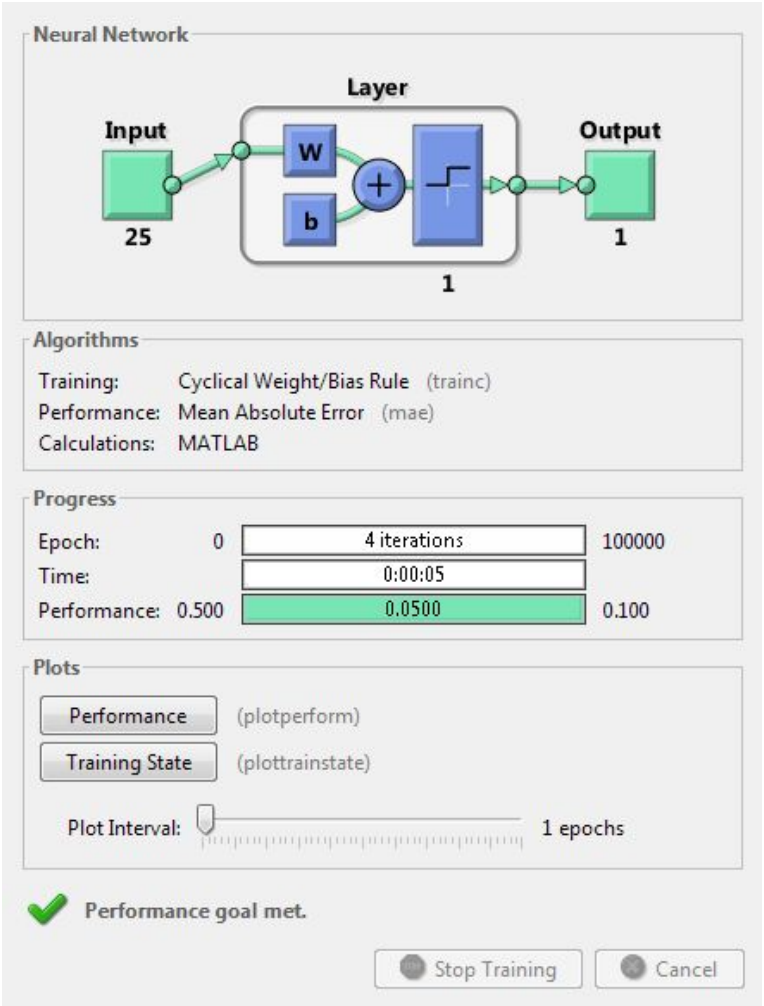
Litery są reprezentowane przez zbiór zer i jedynek w wektorze, przykładowo:

1	1	1	1	0
1	0	0	0	1
1	1	1	1	0
1	0	0	0	1
1	1	1	1	0

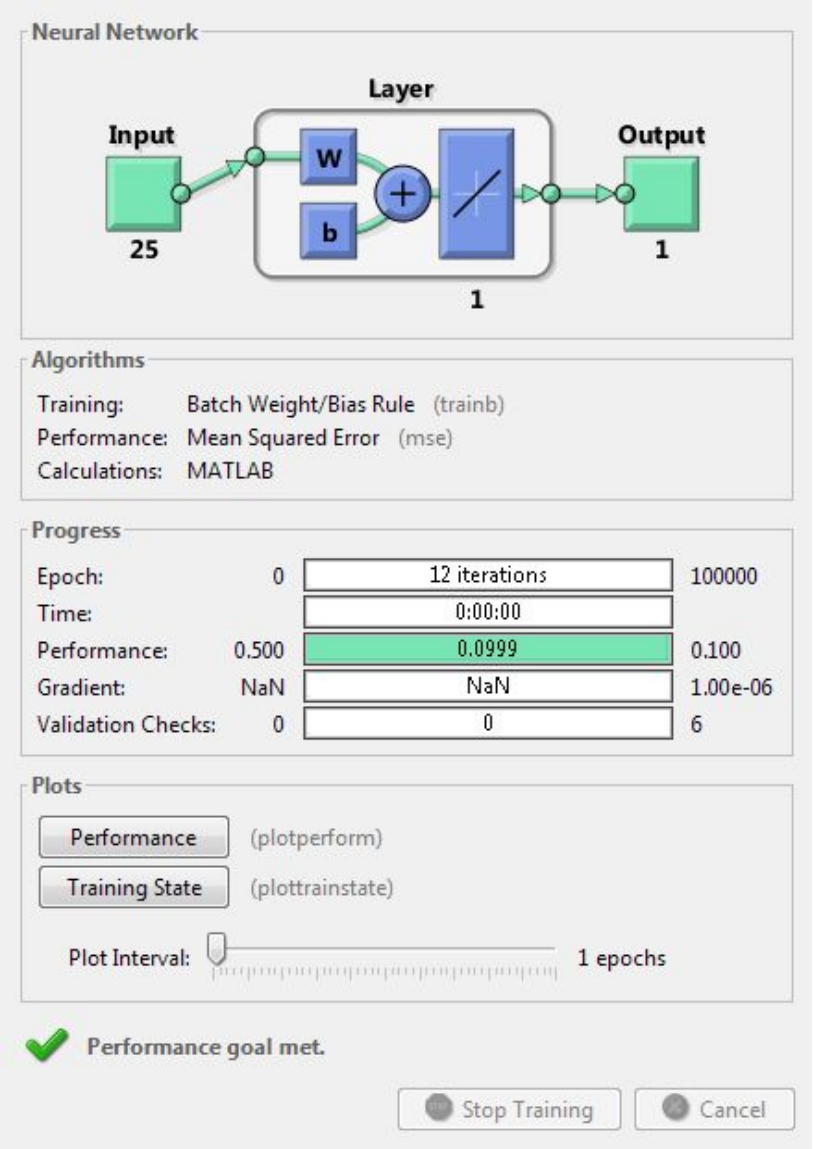
Dane wyjściowe to informacja o wielkości litery - 1 czyli wielka lub 0 czyli mała.

4) <u>Wyniki:</u>

newp



newlin



5) <u>Wnioski:</u>

Sieć złożona z perceptronów (newp) radzi sobie z problemem około 3 razy szybciej niż sieć stworzona funkcją newlin.

W pierwszym wypadku 4 epoki okazały się wystarczające, natomiast w przypadku sieci liniowej potrzebne było ich aż 12.

6) Listing kodu:

```
%% - CZĘŚĆ PIERWSZA
close all; clear all; clc;

%Dane wejściowe - wektory jednokolumnowe
%Litery G, J, oraz M do Z zostały pominięte dla uproszczenia

%       A a B b C c D d E e F f H h I i K k L l
iData = [ 0 0 1 1 0 0 1 0 1 0 1 0 1 1 1 1 1 1 1 1;
          1 1 1 0 1 0 1 0 1 1 1 1 0 0 0 0 0 0 0 0;
          1 1 1 0 1 0 1 0 1 1 1 1 0 0 0 0 0 0 0 0;
          1 0 0 0 1 0 0 1 1 0 1 0 0 0 0 0 1 0 0 0;
          0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0;
          1 0 1 1 1 0 1 0 1 1 1 1 1 1 1 0 1 1 1 1;
          0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
          0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0;
          0 1 1 0 0 0 1 1 0 1 0 0 0 0 0 0 0 0 0 0;
          1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0;
          1 0 1 1 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1;
          1 1 1 1 0 1 0 1 1 1 1 1 1 1 0 0 1 0 0 0;
          1 1 1 1 0 1 0 1 1 1 1 1 1 1 0 0 0 1 0 0;
          1 1 0 0 0 0 1 1 0 0 0 0 1 0 0 0 0 0 0 0;
          1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0;
          1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1;
          0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0;
          0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0;
          0 1 1 1 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0;
          1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0;
          1 0 1 1 0 0 1 0 1 0 1 1 1 1 1 1 1 1 1 1;
          0 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 1 1 1;
          0 1 1 1 1 1 1 1 1 1 0 0 0 1 0 0 0 1 1 1;
          0 1 0 0 1 0 0 1 1 0 0 0 0 0 0 0 1 0 1 0;
          1 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0;];

%Dane testowe:
A = [ 0; 1; 1; 1; 0; 1; 0; 0; 0; 1; 1; 1; 1; 1; 1; 1; 0; 0; 0; 1; 1; 0; 0; 0; 1 ];
a = [ 0; 1; 1; 0; 0; 0; 0; 0; 1; 0; 0; 1; 1; 1; 0; 1; 0; 0; 1; 0; 0; 1; 1; 1; 1 ];
B = [ 1; 1; 1; 0; 0; 1; 0; 0; 1; 0; 1; 1; 1; 0; 0; 1; 0; 0; 1; 0; 1; 1; 1; 0; 0 ];
b = [ 1; 0; 0; 0; 0; 1; 0; 0; 0; 0; 1; 1; 1; 0; 0; 1; 0; 0; 1; 0; 1; 1; 1; 0; 0 ];
C = [ 0; 1; 1; 1; 0; 1; 0; 0; 0; 0; 1; 0; 0; 0; 0; 1; 0; 0; 0; 0; 0; 1; 1; 1; 0 ];
c = [ 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 1; 1; 0; 0; 1; 0; 0; 0; 0; 0; 1; 1; 0; 0 ];
D = [ 1; 1; 1; 0; 0; 1; 0; 0; 1; 0; 1; 0; 0; 1; 0; 1; 0; 0; 1; 0; 1; 1; 1; 0; 0 ];
d = [ 0; 0; 0; 1; 0; 0; 0; 0; 1; 0; 0; 1; 1; 1; 0; 1; 0; 0; 1; 0; 0; 1; 1; 1; 0 ];
E = [ 1; 1; 1; 1; 0; 1; 0; 0; 0; 0; 1; 1; 1; 0; 0; 1; 0; 0; 0; 0; 1; 1; 1; 1; 0 ];
```

```
e = [ 0; 1; 1; 0; 0; 1; 0; 0; 1; 0; 1; 1; 1; 0; 0; 1; 0; 0; 0; 0; 0; 1; 1; 0; 0 ];
F = [ 1; 1; 1; 1; 0; 1; 0; 0; 0; 0; 1; 1; 1; 0; 0; 1; 0; 0; 0; 0; 1; 0; 0; 0; 0 ];
f = [ 0; 1; 1; 0; 0; 1; 0; 0; 0; 0; 1; 1; 1; 0; 0; 1; 0; 0; 0; 0; 1; 0; 0; 0; 0 ];
H = [ 1; 0; 0; 0; 1; 1; 0; 0; 0; 1; 1; 1; 1; 1; 1; 1; 0; 0; 0; 1; 1; 0; 0; 0; 1 ];
h = [ 1; 0; 0; 0; 0; 1; 0; 0; 0; 0; 1; 1; 1; 0; 0; 1; 0; 1; 0; 0; 1; 0; 1; 0; 0 ];
I = [ 1; 0; 0; 0; 0; 1; 0; 0; 0; 0; 1; 0; 0; 0; 0; 1; 0; 0; 0; 0; 1; 0; 0; 0; 0 ];
i = [ 1; 0; 0; 0; 0; 0; 0; 0; 0; 0; 1; 0; 0; 0; 0; 1; 0; 0; 0; 0; 1; 0; 0; 0; 0 ];
K = [ 1; 0; 0; 1; 0; 1; 0; 1; 0; 0; 1; 1; 0; 0; 0; 1; 0; 1; 0; 0; 1; 0; 0; 1; 0 ];
k = [ 1; 0; 0; 0; 0; 1; 0; 0; 0; 0; 1; 0; 1; 0; 0; 1; 1; 0; 0; 0; 1; 0; 1; 0; 0 ];
L = [ 1; 0; 0; 0; 0; 1; 0; 0; 0; 0; 1; 0; 0; 0; 0; 1; 0; 0; 0; 0; 1; 1; 1; 1; 0 ];
l = [ 1; 0; 0; 0; 0; 1; 0; 0; 0; 0; 1; 0; 0; 0; 0; 1; 0; 0; 0; 0; 1; 1; 1; 0; 0 ];

%Dane wyjściowe (1 - wielka litera, 0 - mała litera)

%        A a B b C c D d E e F f H h I i K k L l
oData = [ 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 ];

%PR - zakres wartości -> matryca 5x5 = 25 wejść
PR=[01;01;01;01;01;01;01;01;01;01;01;01;01;01;01;01;01;01;01;01;01;01;01;01;01;];
PR2= [0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1;
          0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; ];
S = 1; %liczba neuronów w sieci

%net - struktura zawierająca parametry sieci
%net=newp(PR2,S);
net=newlin(PR,S);

%newp - Tworzenie jednowarstwowej sieci z twardymi perceptronami
%newlin - Tworzenie jednowarstwowej sieci z neuronami linowymi

% Parametry:
% Maksymalna liczba epok trwania treningu
net.trainParam.epochs = 100000;

% Błąd średniokwadratowy
net.trainParam.goal = 0.1;
% Współczynnik uczenia
net.trainParam.mu = 0.1;
%Przed treningiem
Przed_treningiem_dane=sim(net,iData);
%Trening
net=train(net,iData,oData);

%% - CZĘŚĆ DRUGA
%Po treningu - sieć gotowa do kategoryzowania liter
Po_treningu_dane=sim(net,A);
```

```
%Dane wyjściowe nie są całkowite - żeby podać wynik potrzebne będzie przybliżenie
if round(Po_treningu_dane)==0
        disp('mala litera')
else
        disp('Duza litera')
end;

disp(Po_treningu_dane)
```