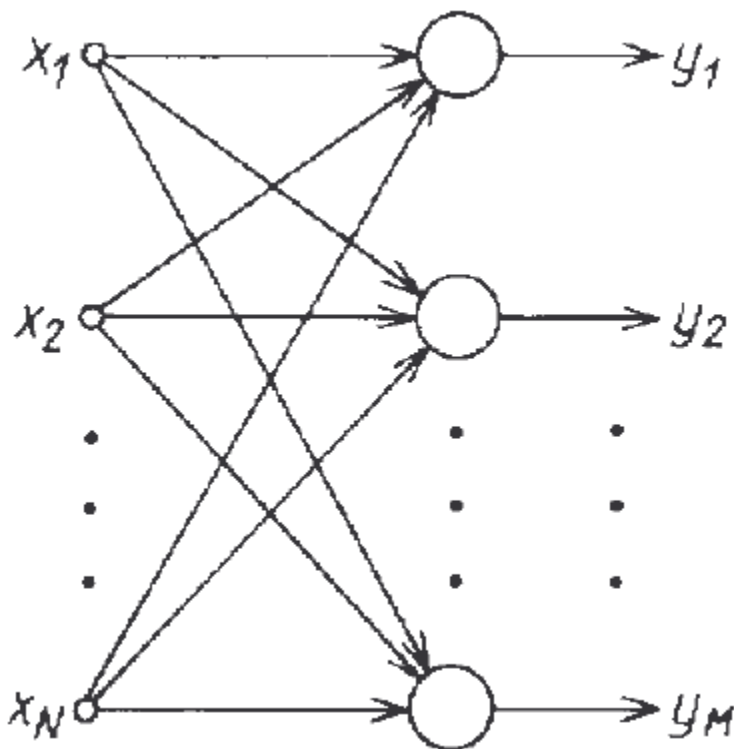


1. Temat:

Budowa i działanie sieci jednowarstwowej.

2. Wstęp teoretyczny:

Sieć neuronowa (sztuczna sieć neuronowa) – ogólna nazwa struktur matematycznych i ich programowych lub sprzętowych modeli, realizujących obliczenia lub przetwarzanie sygnałów poprzez rzędy elementów, zwanych sztucznymi neuronami, wykonujących pewną podstawową operację na swoim wejściu. Oryginalną inspiracją takiej struktury była budowa naturalnych neuronów, łączących je synaps, oraz układów nerwowych, w szczególności mózgu. Czasem nazwą sztuczne sieci neuronowe określa się interdyscyplinarną dziedzinę wiedzy zajmującą się konstrukcją, trenowaniem i badaniem możliwości tego rodzaju sieci.



3. Użyte funkcje:

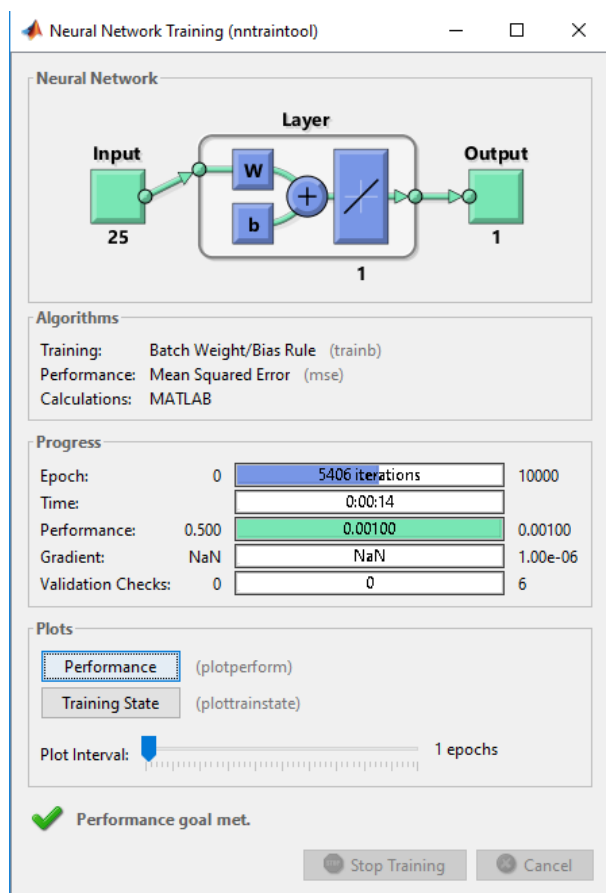
- net – zmienna, do której przypisywana jest nowa sieć neuronowa.
- newlin() – funkcja tworzy jednowarstwową sieć neuronową. Jako pierwszy argument przyjmuje wartości minimalne i maksymalne dla wejściowych elementów sieci dla funkcji tworzących sieć neuronową. Argument ten składa się z 25 par oznaczających odpowiednio 1, czyli wartość maksymalną, lub 0 dla wartości minimalnej. Drugim argumentem jest liczba elementów wektora

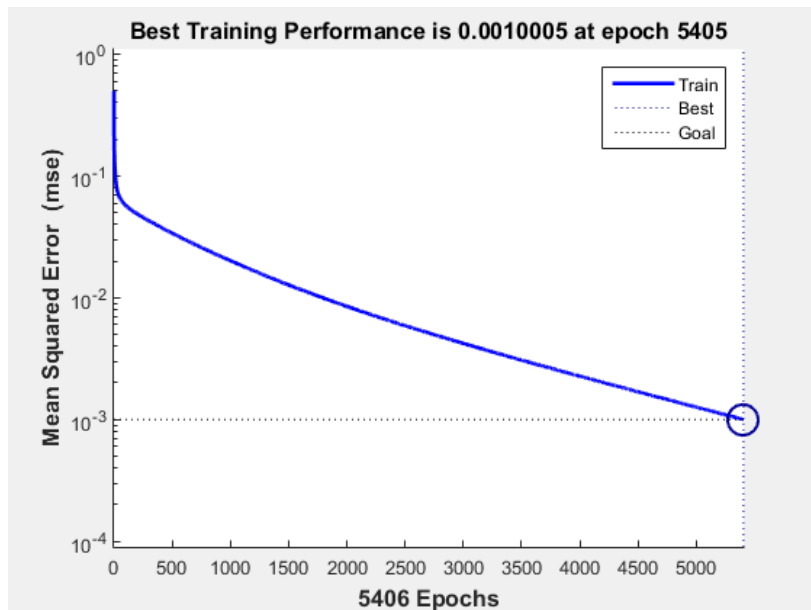
wyjściowego sieci, równa 1. Jego zadaniem jest przechowanie informacji, czy litera jest mała czy duża.

- newp() – funkcja tworzy prostą sieć neuronową, pobiera dane wejściowe i zwraca perceptron. Przyjmuje te same argumenty co newlin().
- WEJŚCIE/WYJŚCIE – zmienne uczące oraz zmienne przechowujące dane wyjściowe odpowiadające danym uczącym.
- net.trainParam.x – funkcja umożliwiająca określenie parametrów treningu sieci.
- train() – uczenie sieci z wykorzystaniem danych wejściowych i wyjściowych.
- x_test – zmienne dla danych sterujących
- sim() – symulacja sieci, parametry to sieć oraz badana litera

4. Wnioski i wyniki

Newlin():	0.001	0.01	0.1
A	0.9356	0.9356	0.9356
a	0.0896	0.0896	0.0896
C	0.9629	0.9629	0.9629
c	0.0036	0.0036	0.0036
E	1.0372	1.0372	1.0372
e	0.0885	0.0885	0.0885
H	1.0269	1.0269	1.0269
h	0.0194	0.0194	0.0194





Newp niezależnie od argumentów przyjmował wartość 1 – gdy duża litera, 0 – gdy mała litera. Wartość współczynnika uczenia, mimo zmian jego wartości od 0.001 do 0.1 nie ma żadnego wpływu na otrzymane wyniki. Czas uczenia neuronu również nie zależy od współczynnika. Na podstawie zaobserwowanych wyników można stwierdzić, że nie istnieje metoda pozwalająca na jednoznaczne i pewne zidentyfikowanie właściwego wyniku, dlatego zawsze należy dobrą metodę zgodnie z potrzebami i przeprowadzić różne próby, a najlepiej wykorzystać różne metody.

5. Kod źródłowy:

Napisany program:

```
close all; clear all; clc;

poczatek = [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; 0
1;];
wyjscia = 1;
%net = newlin(poczatek, wyjscia);
net = newp(poczatek, wyjscia);

WEJSCIE = [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 0 1 0 0 0;
            0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0;
            1 0 1 1 1 0 1 0 1 1 1 1 1 1 1 1 0 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 1 0 0 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 0 0 1 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 1 0 0 1 0 0 0;
            1 1 1 1 0 1 0 1 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 0 1 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 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 1 1 1 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0;
        1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0;
        1 0 1 1 0 0 1 0 1 0 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 0 1;
        0 1 1 1 1 1 1 1 1 1 0 0 0 1 0 0 0 1 1;
        0 1 0 0 1 0 0 1 1 0 0 0 0 0 0 0 1 0 1;
        1 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0];

WYJSCIE = [1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0];

net.trainParam.epochs = 10000;
net.trainParam.goal = 0.001;
net.trainParam.mu = 0.01;

net = train(net, WEJSCIE, WYJSCIE);

A_test = [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_test = [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_test = [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_test = [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_test = [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_test = [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_test = [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_test = [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_test = [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_test = [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_test = [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_test = [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_test = [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_test = [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_test = [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_test = [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_test = [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_test = [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_test = [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_test = [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];
testuj = sim(net, L_test)
if round(testuj) == 0
disp('Mala litera');
else
disp('Duza litera');
end

```