

Scenariusz 2

Temat ćwiczenia: Budowa i działanie sieci jednowarstwowej

Celem ćwiczenia jest poznanie budowy i działania jednowarstwowych sieci neuronowych oraz uczenie rozpoznawania wielkości liter.

Sieć neuronowa (sztuczna sieć neuronowa) – jest to 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 synapsy, oraz układów nerwowych, w szczególności mózgu.

Aby wypełnić zadanie utworzyłem dwie sieci neuronowe, które miały być poddane procesowi nauczania rozpoznawania różnic między małymi oraz dużymi literami. W tym celu wprowadziłem dane uczące w postaci pierwszych dziesięciu liter angielskiego alfabetu zarówno w dużej jak i małej postaci oraz dane testowe. Każdą z liter przedstawiłem w postaci dwuwymiarowej tablicy o wymiarach 5x5.

Kolejnym krokiem było wykorzystanie biblioteki Neural Network Tool zawartej w Matlabie, by wygenerować dwie jednowarstwowe sieci neuronowe a następnie je nauczyć i przetestować.

Każda z liter w tablicy została zinterpretowana przez sieć w sposób następujący:

- Tablica składa się z 25 pól
- Jeżeli pole jest czarne (ma wartość 1) oznacza to, że w danym polu występuje fragment litery
- Jeżeli pole jest białe (ma wartość 0) oznacza to, że w danym polu nie występuje fragment litery

Wykorzystane metody zmiennych:

1) Funkcja **newp(start, wyjścia_sieci, TF, LF)** - tworzy prostą sieć neuronową. Funkcja ta pobiera dane wejściowe, a następnie zwraca perceptron. Jej argumentami są:

- **start** - wartości minimalne i maksymalne dla elementów wejściowych sieci funkcji tworzących sieć neuronową, składa się z 25 par, które odpowiednio oznaczają wartość maksymalną- 1 oraz minimalną- 0
- **wyjścia_sieci** - liczba elementów wektora wyjściowego sieci, jego wartość jest równa 1
- **TF** - funkcja transferu, gdy nie zostanie podana domyślnie przyjmuje 'hardlim'
- **LF** - funkcja uczenia, gdy nie zostanie podana domyślnie przyjmuje 'learnp'

2) Funkcja **newlin(start, wyjścia_sieci)** - tworzy jednowarstwową sieć neuronową. Argumenty jak w newp()

3) Inne użyte funkcje i zmienne:

-**sim** - symuluje działanie perceptronu

-**disp** - wyświetla informacje

-**train** - służy do nauki sieci

Kod w Matlabie:

```
close all; clear all; clc;
```

```
%wejścia do sieci oraz minimalne oraz maksymalne wartości wejść
start = [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;];
```

```
%wyjścia sieci (maksymalna ilość wyjść)
wyjścia_sieci = 1;
```

```
%net = newlin(start, wyjścia_sieci); %metoda 1
net = newp(start, wyjścia_sieci);    %metoda 2
```

```
%kolumnowa reprezentacja binarna pierwszych 10 liter alfabetu dla tablicy o
rozmiarach 5x5
```

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

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

```

```

%dane do testów

```

```

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];

%treningu sieci
net.trainParam.epochs = 10000;
net.trainParam.goal = 0.01;
net.trainParam.mu = 0.001;

%uczenie
net = train(net, WEJSCIE, WYJSCIE);

%testowanie
wynik = sim(net, B)

if wynik == 0
    disp('Mala litera');
else
    disp('Duza litera');
end

```

Przykład:

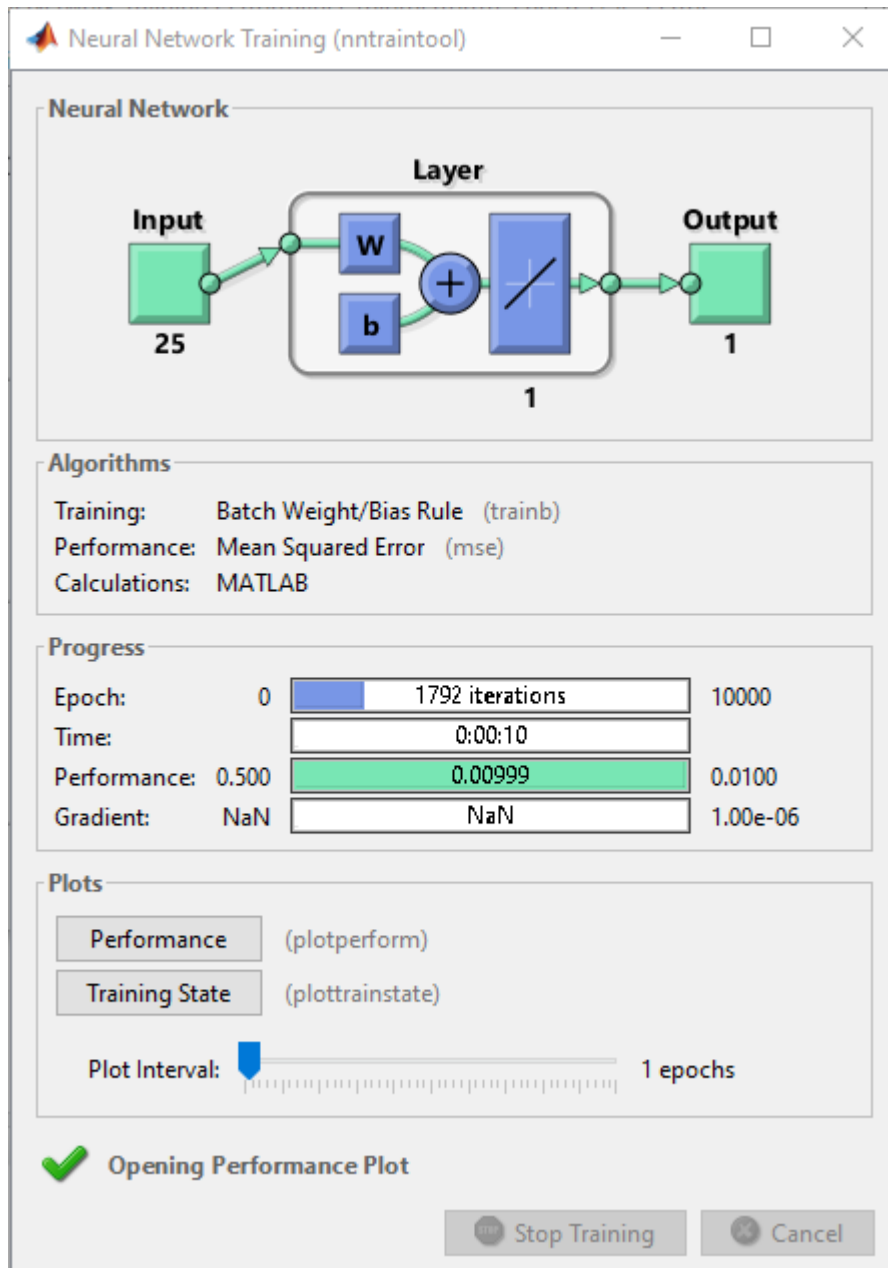
```

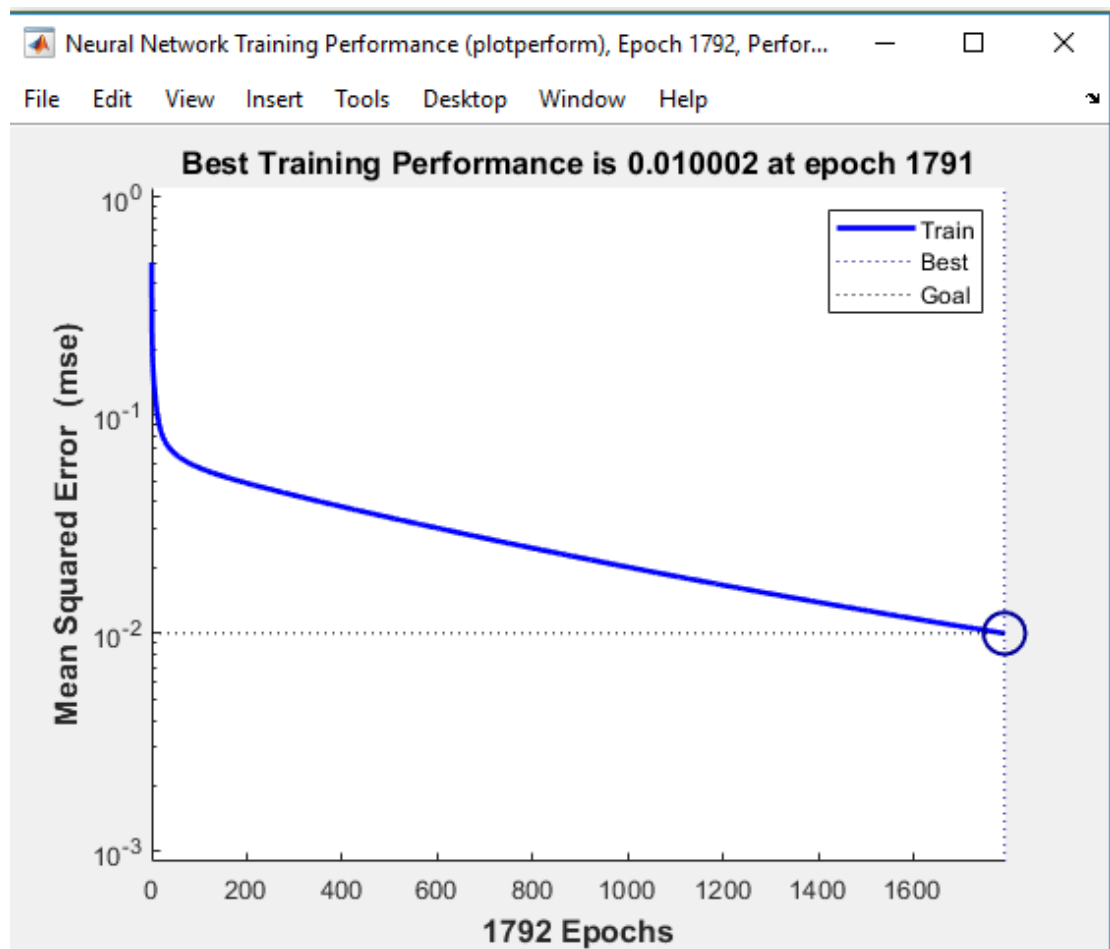
wynik =
    1
Duza litera
>>

wynik = sim(net, B)
|
if wynik == 0
    disp('Mala litera');
else
    disp('Duza litera');
end

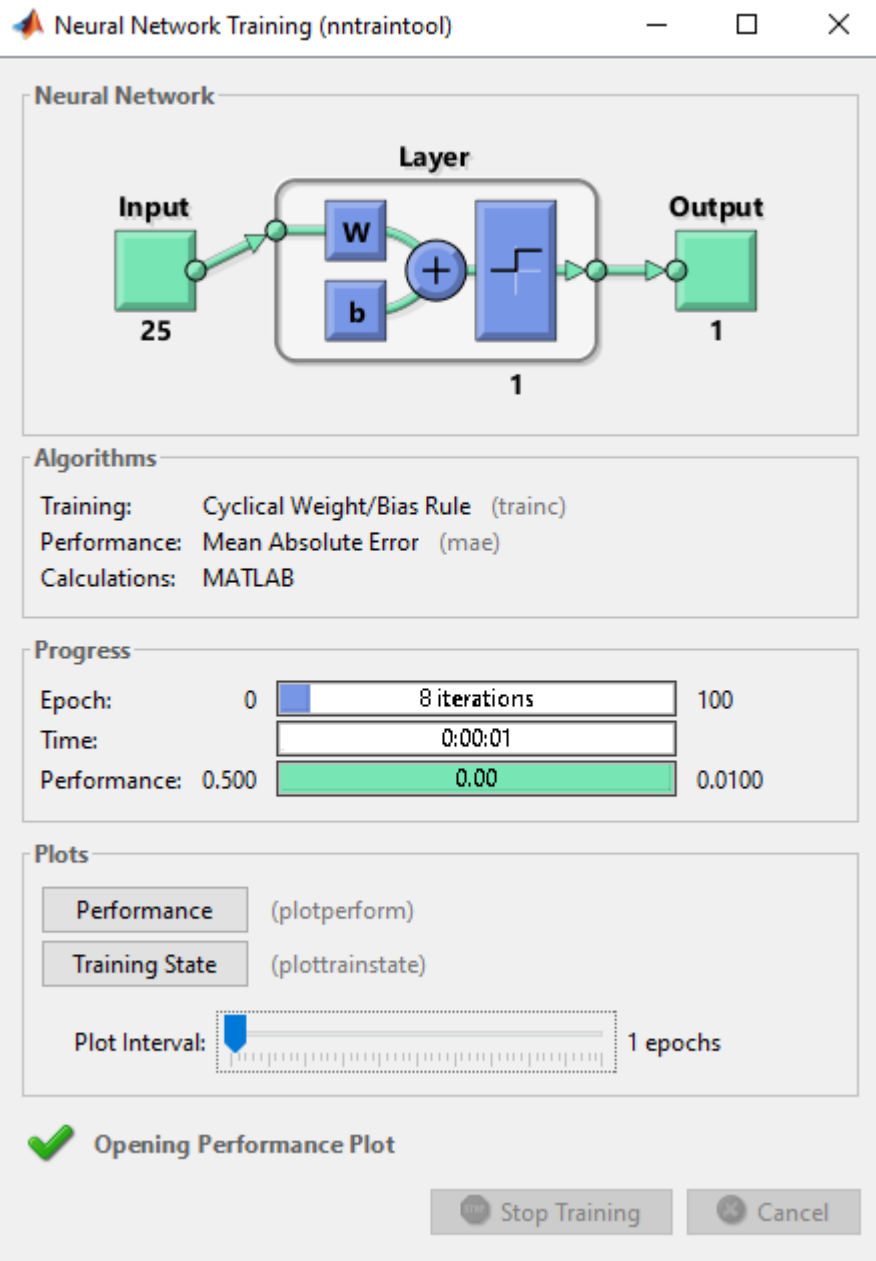
```

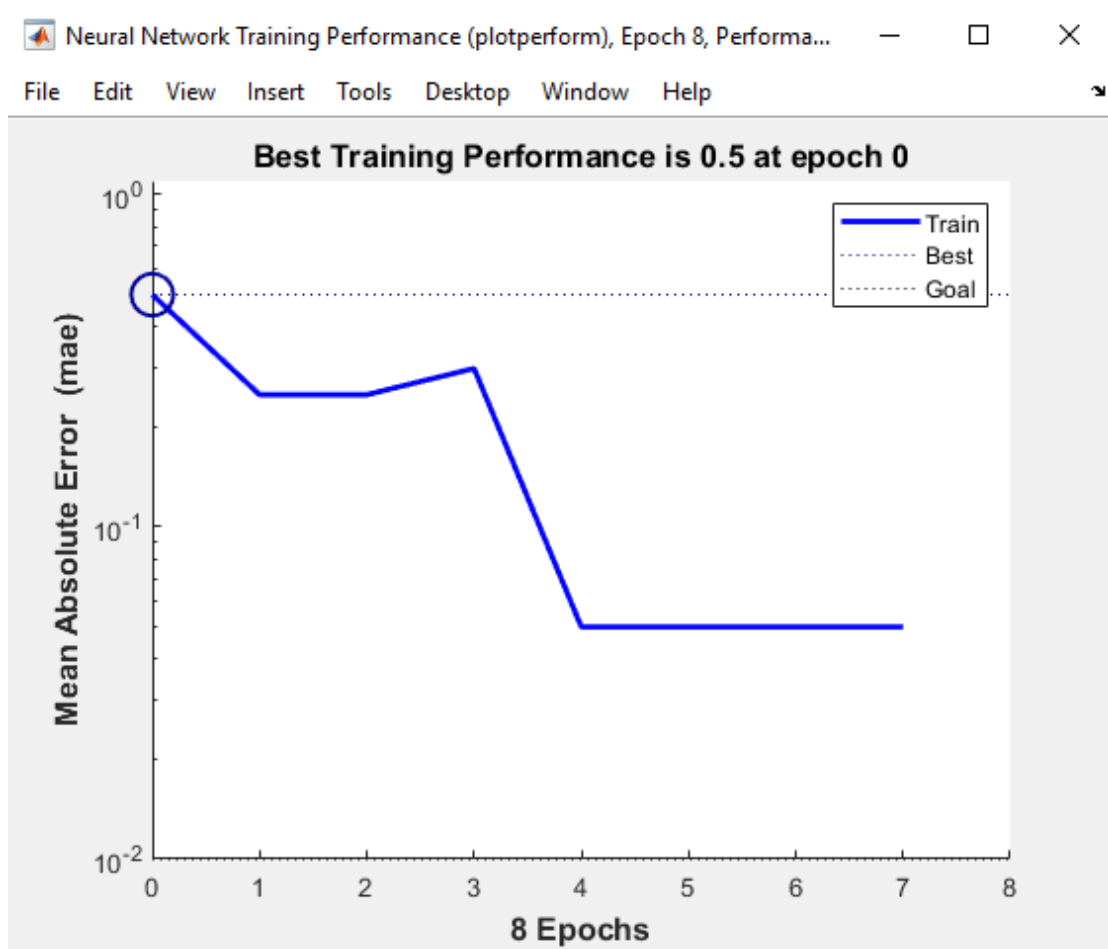
Dla funkcji newlin:





Dla funkcji newp:





Wnioski:

- newp jest szybsze niż newlin
- Dane podane na wejściu były kompletne, co znacząco wpłynęło na szybkość nauki
- W przypadku użyc obu funkcji wszystkie testy zakończyły się poprawnie
- Dla funkcji **newp** zmiana parametrów uczenia sieci nie ma wpływu na wyniki
- Dla funkcji **newlin** zwiększenie błędu średniokwadratowego oraz współczynnika przyspiesza czas nauki sieci