

Budowa i działanie sieci jednowarstwowej

1. Cel projektu

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

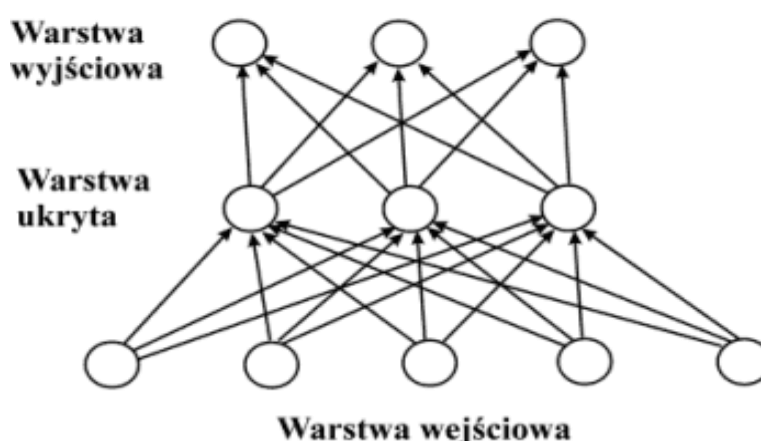
2. Podstawowe pojęcia

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.

Jest to zbiór neuronów realizujących różne cele. W przypadku sztucznych sieci neuronowych jest to sztuczna struktura zaprojektowana i zbudowana w taki sposób, aby modelowała działanie naturalnego układu nerwowego, a w szczególności mózgu.

Sieci jednokierunkowe - sieci, w których nie występują żadne sprzężenia zwrotne. W sieciach jednokierunkowych sygnały są przesyłane od warstwy wejściowej poprzez warstwy ukryte (jeśli występują) do warstwy wyjściowej. Sposób działania tego rodzaju sieci określa jednocześnie ich nazwę.

Sieć jednokierunkowa



3. Opis działania

Newp- funkcja tworząca jednowarstwową sieć neuronową złożoną z określonej ilości neuronów.

Parametry tej funkcji;

- Macierz określająca liczbę wejść do sieci
- Liczba neuronów w sieci
- Funkcja aktywacji

Init – służy do inicjowania sieci neuronowej, wartości wag i progów wykorzystując tą funkcję zostają przyjęte losowo.

Sim – symulacja wytworzonej sieci neuronowej. Argumentami funkcji jest nazwa sieci oraz tablica elementów, których sieć ma się nauczyć, czyli w tym przypadku małe_litery i duże_litery.

Train – funkcja uczenia sieci neuronowej, argumenty jak w sim.

4. Wykonanie zadania i kod

- a) Tworzenie tablicy dużych i oddzielnie małych liter.
- b) Tworzenie wektora danych wejściowych, potrzebnego do określenia warunku, że 0 to mała litera, 1 duża.
- c) Stworzenie i zainicjowanie sieci neuronowej.
- d) Symulacja sieci.
- e) Trening sieci.
- f) Ponowna symulacja sieci.
- g) Stworzenie wektora litery i symulacja z jego użyciem
- h) Warunek, który sprawdza, czy litera jest mała czy duża. Stosowny komunikat.

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

```
%Litera która sprawdza nasza sieć neuronowa
```

```
A = [0 1 1 1 0 1 0 0 0 1 1 0 0 0 1 1 1 1 1 1 0 0 0 1 1 0 0 0 1 1 0 0 0  
1];  
A=A';
```

```
B = [1 1 1 1 0 1 0 0 0 1 1 0 0 0 1 1 1 1 1 0 1 0 0 0 1 1 0 0 0 1 1 1 1 1  
0];  
B = B';
```

```
a_mal=[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 1 0 0 1 0 1 0 0 1 0 1 1 1 1  
1 ];  
a_mal = a_mal';
```

```
%Zbiór naszych małych liter
```

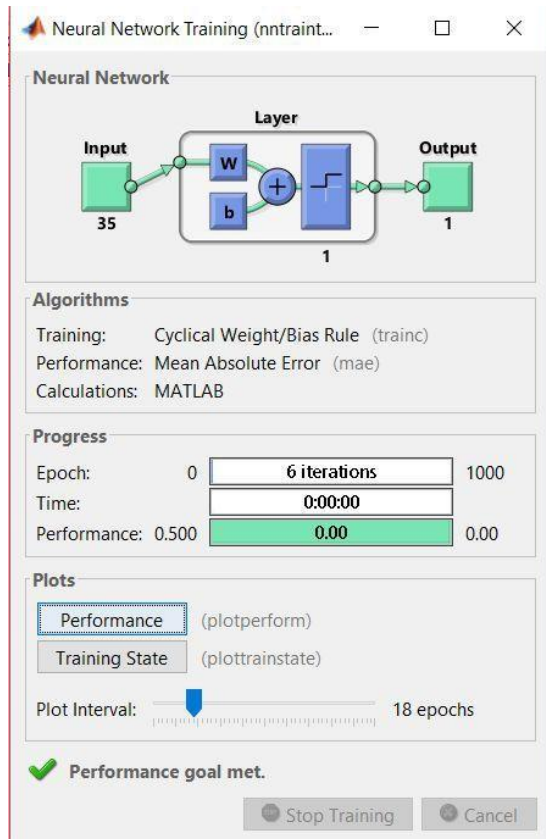
```
a=[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 1 0 0 1 0 1 0 0 1 0 1 1 1 1 ];  
b=[1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 1 1 1 0 1 0 0 0 1 1 0 0 0 1 1 1 1 0 ];  
c=[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 1 0 0 0 0 1 0 0 0 0 1 1 1 1 0 ];  
d=[0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 1 1 1 1 1 0 0 0 1 1 0 0 0 1 0 1 1 1 1 ];  
e=[0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 1 0 0 0 1 1 1 1 1 1 1 1 0 0 0 0 0 1 1 1 1 ];  
f=[0 0 0 0 0 0 1 1 1 0 0 1 0 0 0 1 1 1 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 ];  
g=[0 1 1 1 0 0 1 0 1 0 0 1 1 1 1 0 0 0 1 0 0 0 1 1 0 0 1 0 1 0 0 1 1 1 0 ];  
h=[1 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 1 1 1 0 1 0 0 1 0 0 1 0 0 1 0 0 1 0 ];  
i=[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 1 0 0 0 0 1 0 0 0 ];  
j=[0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 1 0 0 0 ];
```

```
%Warunek sieci neuronowej:
if Siec_litery==1 disp('Litera jest duza');
else disp('Litera jest mala');
end
```

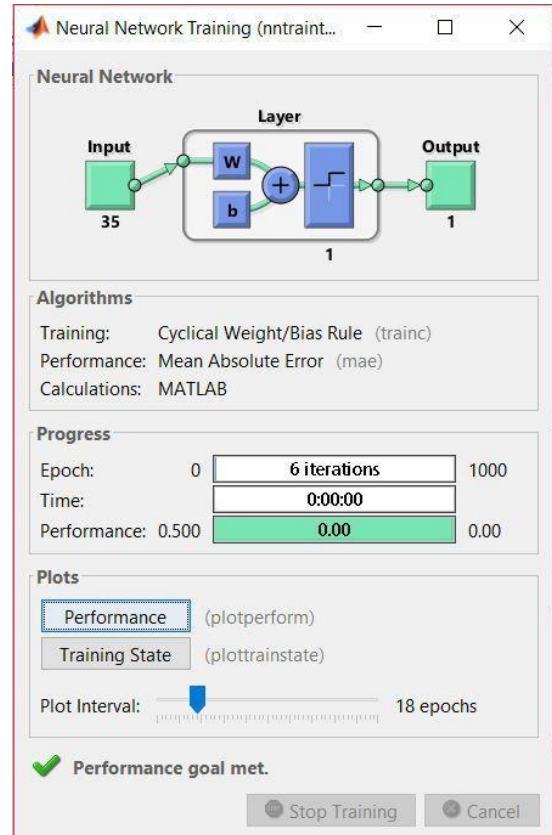
0	1	1	1	0
1	0	0	0	1
1	1	1	1	1
1	0	0	0	1
1	0	0	0	1

Dane wyjściowe to informacja, czy litera jest mała czy duża. 0 – mała, 1- duża.

Dla litery A:



Dla litery a:



Ilość iteracji (epoch) w obu przypadkach wynosi 6.

Na podstawie tego można uznać, że program jest szybki.

Funkcja newp daje dużo korzystniejsze wyniki niż funkcja newlin (tu ilość iteracji potrafi wzrosnąć aż do tysięcy).