

Sprawozdanie 6:

Budowa i działanie sieci Kohonena dla WTA. cz.2

1. Cel projektu:

Celem ćwiczenia było poznanie budowy i działania sieci Kohonena przy wykorzystaniu reguły WTM do odwzorowywania istotnych cech liter alfabetu.

2. Pojęcia niezbędne do realizacji projektu:

→ WTM(Zasada WTM):

[*Winner Takes Most*]

Najprościej mówiąc zasada **WTM** realizuje sieć, w której zwycięzca bierze większość.

Neuron zwycięski oraz neurony sąsiadujące z neuronem zwycięskim, czyli należące do sąsiedztwa $N_c(k)$ aktualizują swoje wagi według zasady:

$$w_i(k+1) = w_i(k) + \eta(k)G(i,c)[x(k) - w_i(k)], \quad i \in N_c(k)$$

$G(i,c)$ - funkcja określająca wpływ sąsiedztwa.

→SOM:

Sieć samoorganizująca się.

Poprzez samouczenie się sieci samoorganizujące się pozwalają na redukcję wymiaru analizowanych sieci.

SOM-y:

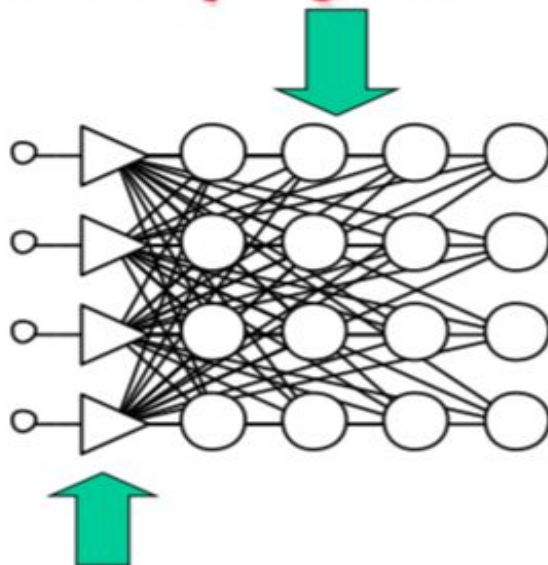
- dokonują redukcji wymiarów danych poprzez tworzenie map (jedno- lub dwu- wymiarowych), które przedstawiają graficznie podobieństwo analizowanych danych grupując podobne dane razem.
- redukują wymiar danych i ilustrują graficzne podobieństwo między nimi.
- ułatwiają zrozumienie (analizę) danych wielowymiarowych.

Sieci Kohonena:

- sieci samoorganizujące się, które uczą się przy pomocy algorytmu Kohonena.
- Sieć jednokierunkowa, złożona z jednej warstwy w której neurony ułożone są w pewnym porządku topologicznym.
- Mapa jednowymiarowa to po prostu jeden wiersz neuronów.

Struktura sieci Kohonena:

Warstwa topologiczna



Warstwa
wejściowa

3. Kod programu i wyjaśnienia:

%liczby zapisane w postaci zer i jedynek

```
A_ =A';
B_ =B';
C_ =C';
D_ =D';
E_ =E';
F_ =F';
G_ =G';
H_ =H';
I_ =I';
J_ =J';
K_ =K';
L_ =L';
M_ =M';
N_ =N';
O_ =O';
P_ =P';
Q_ =Q';
R_ =R';
S_ =S';
T_ =T';

input=[ A_(:) B_(:) C_(:) D_(:) E_(:) F_(:) G_(:) H_(:) I_(:) J_(:) K_(:) L_(:) M_(:) N_(:) O_(:) P_(:) Q_(:) R_(:) S_(:) T_(:)];

%wymiar naszej samoorganizującej się sieci
dimensions = [10 10];
coverSteps = 100;
%sąsiedztwo - stopień
initNeighbor = 1;
%tworzymy mapę sześciokątów
topologyFcn = 'hextop';
distanceFcn = 'linkdist';

net = selforgmap(dimensions,coverSteps,initNeighbor,topologyFcn,distanceFcn);

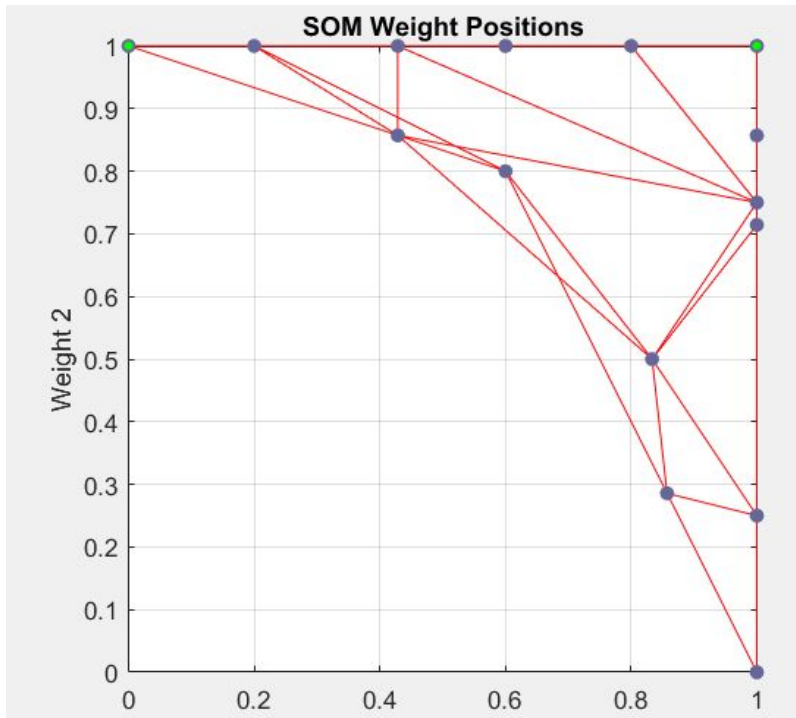
net.trainParam.epochs = 200;
net.trainFcn = 'trainbu';
net = train(net,input);
y = net(input);
classes = vec2ind(y);

plotsompos(net,input);
grid on
```

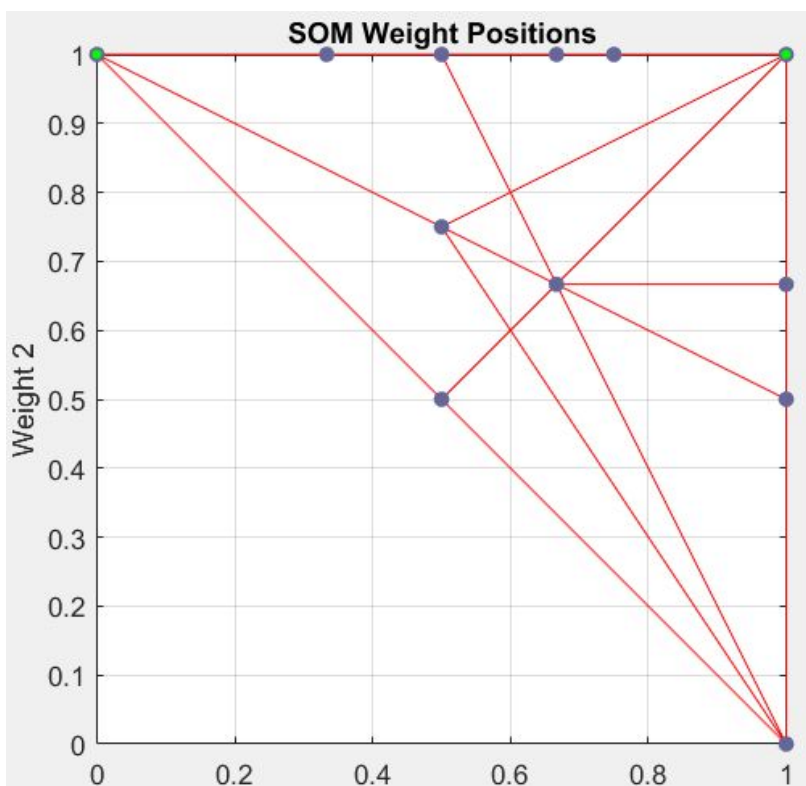
4. Wyjście programu i wnioski:

Macierz SOM:

➔ 4x4:



➔ 10x10:



W kolejnych iteracjach po znalezieniu neuronu zwycięzcy, wyznaczani są jego sąsiedzi. Co krok uaktualniane są wagi węzłów promienia otaczającego neuron zwycięzcy, który jest zdefiniowany przez nas (wielkość promienia).

Węzły muszą mieć na wstępie ustalone wagi, w przeciwnym razie może się zdarzyć, że wszystkie neurony będą zwycięskie.