

## Scenariusz 6

### Temat ćwiczenia:

Budowa i działanie sieci Kohonena dla WTM.

### Cel ćwiczenia:

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

### Zadania do wykonania

- a) Wygenerowanie danych uczących i testujących, zawierających 20 dużych liter dowolnie wybranego alfabetu w postaci dwuwymiarowej tablicy
- b) Przygotowanie (implementacja lub wykorzystanie gotowych narzędzi) sieci Kohonena i algorytmu uczenia opartego o regułę Winner Takes Most (WTM).
- c) Uczenie sieci dla różnych współczynników uczenia.
- d) Testowanie sieci.

Zasada **WTM** - neuron zwycięski oraz neurony sąsiadujące z neuronem zwycięskim, czyli należące do sąsiedztwa  $N_c(k)$ , aktualizują swoje wagi według zasady:

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

Funkcja  $G$  oznacza tu wpływ sąsiedztwa.

Sieci samorganizujące się (SOMy), opracowane przez prof. Tuevo Kohonena, wykorzystywane są często do wizualizacji danych wielowymiarowych. Pozwalają na zredukowanie wymiaru analizowanych danych poprzez samouczenie się sieci.

**SOMy** stanowią narzędzie ułatwiające zrozumienie (analizę) danych wielowymiarowych (nasze zmysły odczytują jedynie wizualizację max. trzech wymiarów).

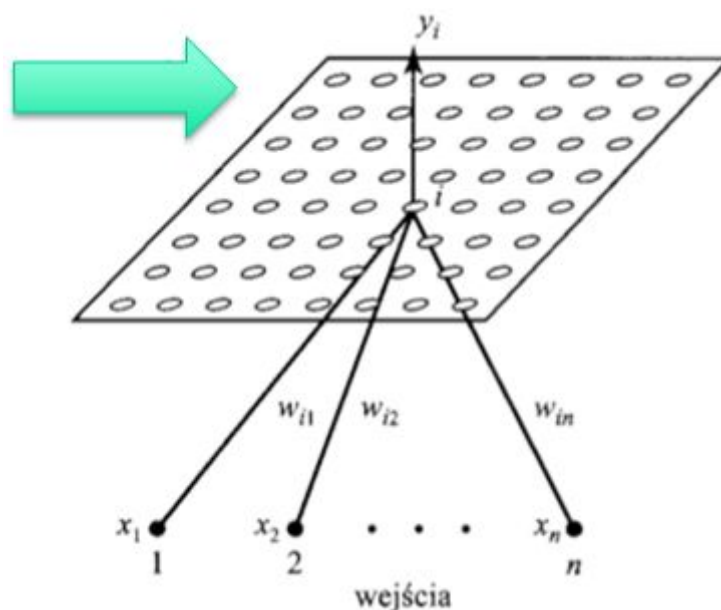
**SOMy** dokonują redukcji wymiarów danych poprzez tworzenie map, zazwyczaj jedno- lub dwu- wymiarowych, które przedstawiają graficznie podobieństwo analizowanych danych grupując podobne dane razem.

**SOMy** realizują więc dwa zadania: redukują wymiar danych i ilustrują graficzne podobieństwo między nimi.

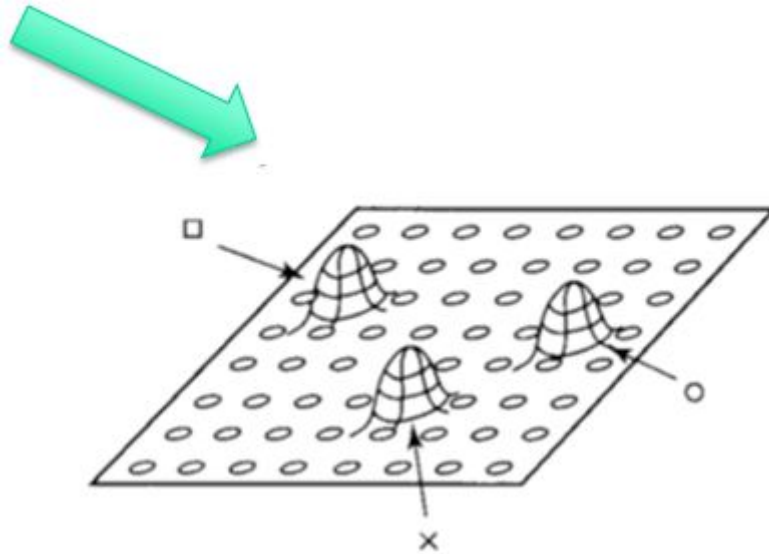
Sieci Kohonena, to sieci samoorganizujące się, które uczą się przy pomocy algorytmu Kohonena.

Jest to sieć jednokierunkowa, złożona z jednej warstwy (matrycy, mapy) w której neurony ułożone są w pewnym porządku topologicznym (np. wiersze i kolumny). Mapa jednowymiarowa to po prostu jeden wiersz (kolumna) neuronów.

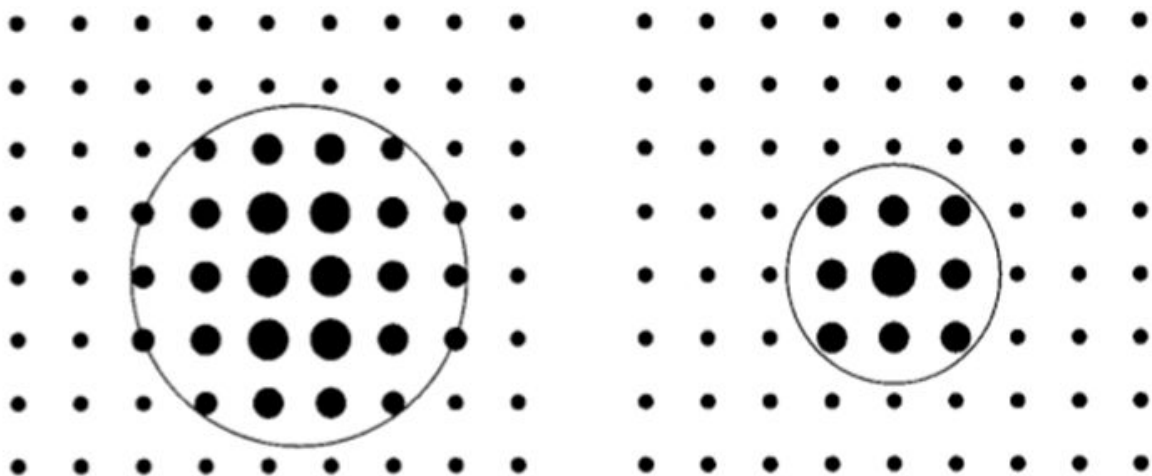
Wektor wejść odwzorowujemy na mapę neuronów. Zwycięża dany neuron, który ma największą wartość wyjścia.



W zależności od ustalonej liczby sąsiednich neuronów, dana funkcja sąsiedztwa przyjmuje różne wartości.



Przykładowo, jeśli damy stopień sąsiedztwa 2 to wszystkie neurony oddalone o promień 2, zostają im odpowiednio skorygowane wagi. Jeśli damy 1 to tylko najbliższe.



## 1. Listing kodu

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

```
AA=A'; BB=B'; CC=C'; DD=D'; EE=E'; FF=F'; GG=G'; HH=H'; II=I'; JJ=J';  
KK=K'; LL=L'; MM=M'; NN=N'; OO=O'; PP=P'; QQ=Q'; RR=R'; SS=S'; TT=T';
```

**Litery zapisane w postaci 0 i 1**

```
wejscie=[ AA(:) BB(:) CC(:) DD(:) EE(:) FF(:) GG(:) HH(:) II(:) JJ(:) KK(:) LL(:) MM(:) NN(:)  
OO(:) PP(:) QQ(:) RR(:) SS(:) TT(:)];
```

```
dimensions = [4 4]; wymiar SOM-u
```

```
coverSteps = 100;
```

```
initNeighbor = 1; stopień sąsiedztwa
```

```
topologyFcn = 'hextop'; mapa sześciokątów
```

```
distanceFcn = 'linkdist';
```

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

```
net.trainParam.epochs = 200;
```

```
net.trainFcn = 'trainbu';
```

```
net = train(net,wejscie);
```

```
y = net(wejscie);
```

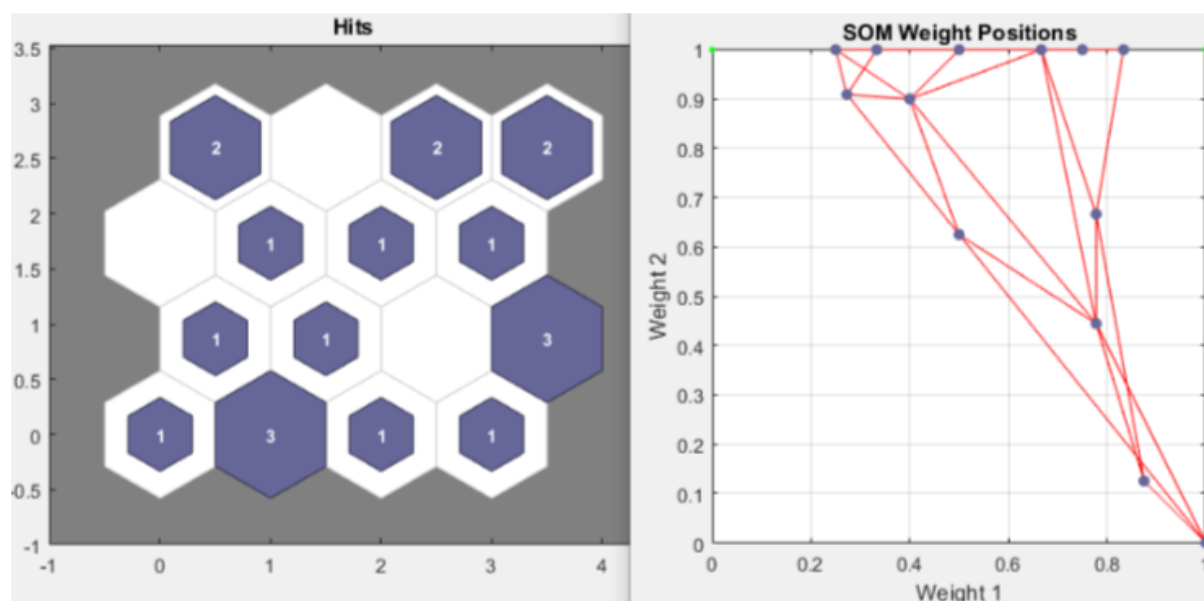
```
classes = vec2ind(y);
```

```
plotsompos(net,wejscie);
```

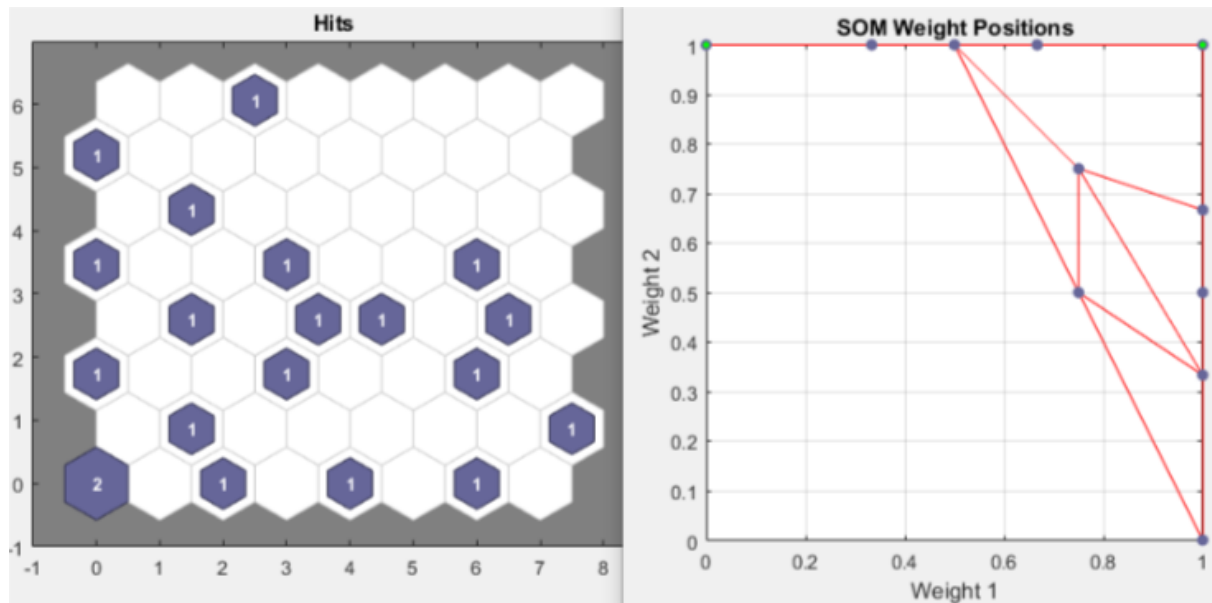
```
grid on
```

## 2. Wynik działania programu

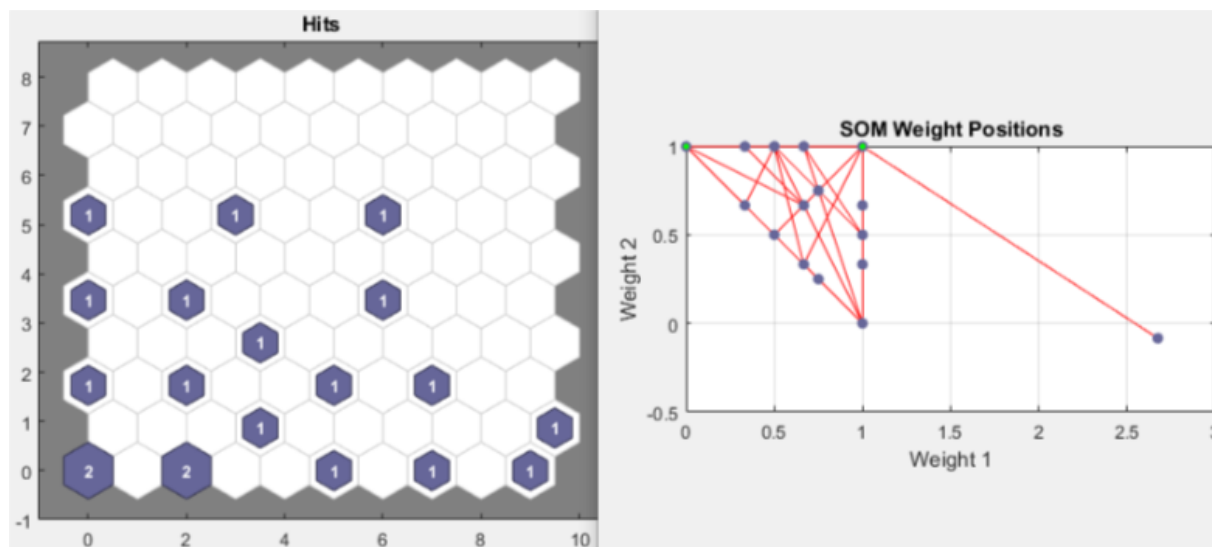
**Macierz SOM 4x4**



## Macierz SOM 8x8



## Macierz SOM 10x10



Przy mapie 4x4 liczba trafień na jeden neuron jest różna i to pozwala stwierdzić, że w jednym neuronie jest wiele podobnych do siebie liter. Np. O jest podobne do D i Q i wszystkie te litery zostały sklasyfikowane do jednego neuronu.

Przy większych mapach już takie ryzyko nie występuje. Z reguły mapa w większości jest pusta, a tylko raz na całe dane wejściowe zdarzyło się, że sieć sklasyfikowała dwie litery do jednego neuronu pod względem tych samych cech istotnych.

### 3. Spostrzeżenia i wnioski

Warunki samoorganizacji kohonenowskiej mapy cech są następujące:

- każdy z neuronów jest pobudzany wystarczającą liczbę razy,
- korekcji podlegają tylko wagi w pobudzonym sąsiedztwie,
- korekcja jest proporcjonalna do pobudzenia neuronu.

Jeżeli w rzeczywistych sytuacjach występuje jakaś naturalna kolejność pojawiania się obrazów, to należy to uwzględnić przy konstrukcji ciągu uczącego.

Na wstępie procesu uczenia, wszystkim wagom każdego z węzłów muszą być przypisane pewne wartości początkowe.

Zazwyczaj, wartości te są wybierane w sposób losowy w zakresie  $0 < w < 1$ . Gdy tego nie zrobimy, wszystkie neurony będą zwycięskie.

W każdej iteracji, po znalezieniu neuronu zwycięzcy BMU, wyznaczane jest jego sąsiedztwo.

Wagi wszystkich węzłów znajdujących się wewnątrz tego promienia są uaktualniane w kolejnym kroku.

Definiujemy wartość tego promienia, a następnie sprawdzamy, który z węzłów znajduje się w jego wnętrzu.

SamoOrganizujące się Mapy Kohonena, pozwalają łatwo sklasyfikować dane. SOM mogą być stosowane do wizualizacji zależności pomiędzy danymi wielowymiarowymi. Przykład - Mapa biedy (bogactwa) świata. Polega na sklasyfikowaniu (pogrupowaniu w klastry) danych statystycznych opisujących różne parametry jakości życia – zdrowie, wyżywienie, dostęp do edukacji, itp.