

Klaudia Wrona Inżynieria Obliczeniowa Numer indeksu: 293128	Podstawy Sztucznej Inteligencji Sprawozdanie nr 6
Budowa i działanie sieci Kohonena dla WTM	

1. Cel projektu

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

2. Zadania do wykonania

- a) Wygenerowanie danych uczących i testujących, zawierających 20 dużych liter w postaci tablic.
- 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.

3. Podstawowe pojęcia

Sieci Kohonena są szczególnym przypadkiem algorytmu realizującego uczenie się bez nadzoru. Ich głównym zadaniem jest organizacja wielowymiarowej informacji (np. obiektów opisanych 50 parametrami w taki sposób, żeby można ją było prezentować i analizować w przestrzeni o znacznie mniejszej liczbie wymiarów, czyli mapie (np. na dwuwymiarowym ekranie). Warunek: rzuty "podobnych" danych wejściowych powinny być bliskie również na mapie. Sieci Kohonena znane są też pod nazwami Self-Organizing Maps, Competitive Filters.

Zasady działania sieci Kohonena:

- Wejścia (tyle, iloma parametrami opisano obiekty) połączone są ze wszystkimi węzłami sieci
- Każdy węzeł przechowuje wektor wag o wymiarze identycznym z wektorami wejściowymi
- Każdy węzeł oblicza swój poziom aktywacji jako iloczyn skalarny wektora wag i wektora wejściowego (podobnie jak w zwykłym neuronie)
- Ten węzeł, który dla danego wektora wejściowego ma najwyższy poziom aktywacji, zostaje zwycięzcą i jest uaktywniony
- Wzmacniamy podobieństwo węzła-zwycięzcy do aktualnych danych wejściowych poprzez dodanie do wektora wag wektora wejściowego (z pewnym współczynnikiem uczenia)
- Każdy węzeł może być stowarzyszony z pewnymi innymi, sąsiednimi węzłami - wówczas te węzły również zostają zmodyfikowane, jednak w mniejszym stopniu.
- Inicjalizacja wag sieci Kohonena jest losowa.

Reguła WTM - reguła aktywacji neuronów w sieci neuronowej, która jest oparta na zasadzie działania WTA z tą różnicą, że oprócz zwycięzcy wagi modyfikują również neurony z jego sąsiedztwa, przy czym im dalsza odległość od zwycięzcy, tym mniejsza jest zmiana wartości wag neuronu. Metoda WTA jest metodą słabo zbieżną - w szczególności dla dużej liczby neuronów. Sąsiedztwo jest pojęciem umownym - można definiować sąsiadów bliższych i dalszych, sąsiedztwo nie oznacza również, że neurony muszą być bezpośrednio połączone ze zwycięzcą.

4. Wykonanie zadania i kod

`selforgmap(dimensions,coverSteps,initNeighbor,topologyFcn,distanceFcn)`

Dimensions	Rozmiar wektora
coverSteps	Liczba kroków szkoleniowych dla początkowego pokrycia przestrzeni wejściowej
initNeighbor	Początkowy rozmiar sąsiedztwa
topologyFcn	Funkcja topologii warstw
distanceFcn	Odległość neuronowa

`gridtop` - oblicza pozycje neuronów w warstwach

`dist` – odległość euklidesowa

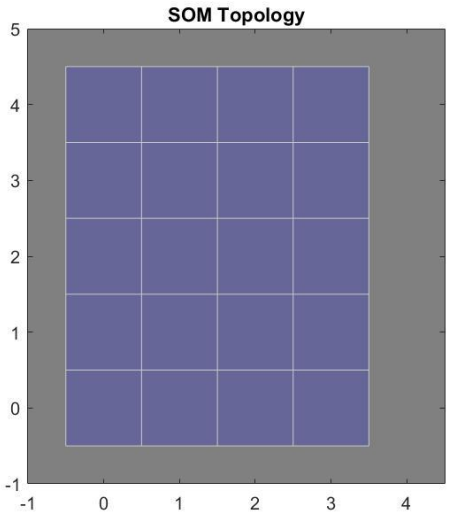
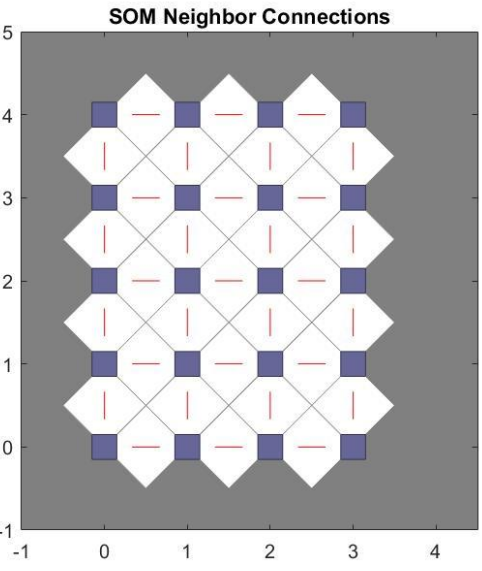
```

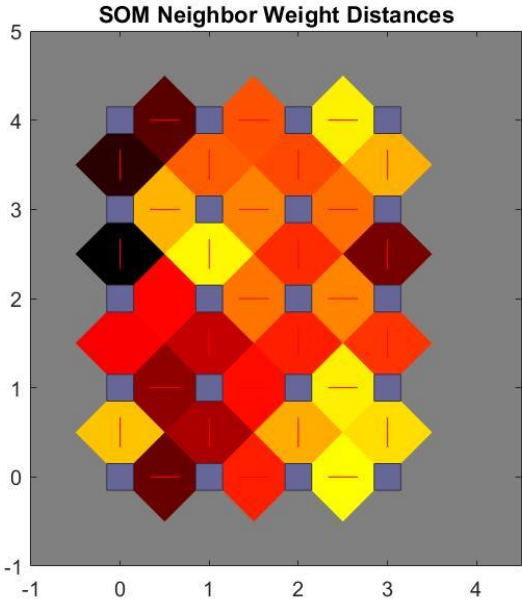
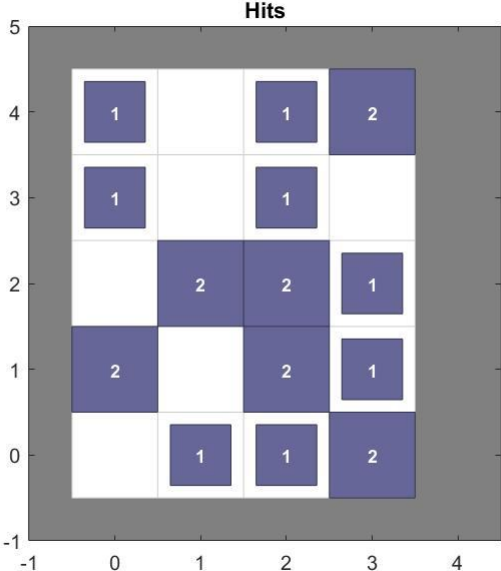
%kolumnowa reprezentacja binarna pierwszych 20 duzych liter
alfabetu dla tablicy 4x5
%dane wejsciowe:
%A B C D E F G H I J K L M N O P R S T U
WEJSCIE = [0 1 0 1 1 1 0 1 1 1 1 1 1 1 1 0 1 1 0 1 1;
1 1 1 1 1 1 1 0 1 1 0 0 0 0 1 1 1 1 1 0;
1 1 1 1 1 1 1 0 1 1 0 0 0 0 1 1 1 1 1 0;
0 0 0 0 1 1 0 1 0 1 1 0 1 1 0 0 0 0 0 1;
1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 1 1 1 0 1;
0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 1 0;
0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0;
1 1 1 1 0 0 0 1 0 1 0 0 1 1 1 1 1 0 0 1;
1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 1 1 0 0 1;
1 1 0 0 1 1 0 1 1 0 1 0 0 1 0 1 1 1 1 0;
1 1 0 0 1 1 1 1 0 0 0 0 1 0 0 1 1 1 0 0;
1 0 0 1 0 0 1 1 0 1 0 0 1 1 1 0 0 0 0 1;
1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 0 0 1;
0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0;
0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 1 0 0 0;
1 1 1 1 0 0 1 1 0 1 0 0 1 1 1 0 0 1 0 1;
1 1 0 1 1 1 0 1 1 0 1 1 1 1 0 1 1 0 0 0;
0 1 1 1 1 0 1 0 1 1 0 1 0 0 1 0 0 1 1 1;
0 1 1 1 1 0 1 0 1 1 0 1 0 0 1 0 0 1 0 1;
1 0 0 0 1 0 0 1 0 1 1 1 1 1 0 0 1 0 0 0];

% PARAMETRY SIECI KOHONENA
dimensions = [4 5]; %wymiar wektora
coverSteps = 100; %liczba kroków szkoleniowych dla początkowego
pokrycia przestrzeni wejściowej
initNeighbor = 1; %początkowy rozmiar sąsiedztwa
topologyFcn = 'gridtop'; %funkcja topologii warstw
distanceFcn = 'dist'; %funkcja odległości neuronowej
% TWORZENIE SIECI KOHONENA (SOM)
net = selforgmap(dimensions, coverSteps,
initNeighbor, topologyFcn, distanceFcn);
net.trainParam.epochs = 1500; %ustalenie maksymalnej liczby epok
treningowych utworzonej sieci
% TRENING SIECI
[net, tr] = train(net, WEJSCIE); %uczenie sieci
y = net(WEJSCIE); %przypisanie sieci do wyjścia Y
classes = vec2ind(y); %konwertowanie wektorów uczonej sieci na
indeksy

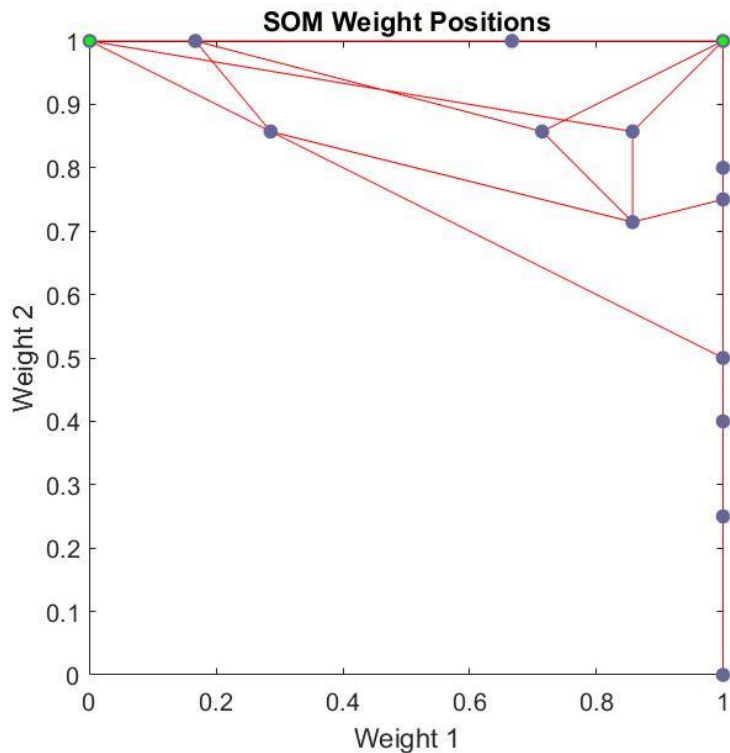
```

5. Wnioski

SOM Topology	Każdy neuron reprezentowany jest przez kwadrat.	 <p>The plot titled "SOM Topology" displays a 4x4 grid of blue squares. The x-axis is labeled from -1 to 4, and the y-axis is labeled from 0 to 5. The grid is centered within the plot area, with squares spanning from x=0 to x=4 and y=0 to y=4.</p>
SOM Neighbor Connections	Powiązania sąsiedzkie	 <p>The plot titled "SOM Neighbor Connections" shows the same 4x4 grid of blue squares. Red lines connect each square to its four immediate neighbors (up, down, left, and right), forming a mesh of diamond shapes. The axes are identical to the first plot, with x from -1 to 4 and y from 0 to 5.</p>

<p>SOM Neighbor Distance -</p>	<p>Niebieskie sześciokąty reprezentują neurony. Czerwone linie łączą sąsiednie neurony. Kolory w obszarach zawierających czerwone linie wskazują odległości między neuronami. Ciemniejsze kolory = większe odległości. Jaśniejsze kolory = mniejsze odległości.</p>	<p>SOM Neighbor Weight Distances</p> 
<p>SOM Sample Hits</p>	<p>Przedstawia ile razy dany neuron zwyciężył podczas rywalizacji</p>	<p>Hits</p> 

WAGI



- ❖ Prostokątna siatka neuronów umożliwia utworzonej sieci stworzenie bezpośrednich powiązań pomiędzy najbliższymi neuronami.
- ❖ Algorytm Winner Takes Most pokazywał prawie równomierne rozłożenie zwycięstw na całej wygenerowanej wcześniej sieci. W poprzednim scenariuszu metoda Winner Takes All koncentrowała wygrane neurony głównie w jednym miejscu sieci heksagonalnej. Rozkład zwycięstw metody WTM zwiększała poprawność działania sieci ze względu na równomierny rozkład zwycięstw.
- ❖ Wagi poszczególnych neuronów są rozłożone w zależności od ilości neuronów w sieci. Zwiększanie liczby neuronów wpływało na czas obliczeń znacznie go wydłużając.
- ❖ Zwiększanie sąsiedztwa powodowało błędy w działaniu sieci neuronowej. Wielkość sąsiedztwa jest uzależniona od wielkości sieci - jeśli oba te parametry rosną jednocześnie wtedy działanie programu jest poprawne.
- ❖ Na podstawie wykresu pokazującego rozkład sił neuronów można zauważyć, że sieć korzystała z mechanizmu WTM (wykres pokazuje prawie równomierny rozkład zwycięstw neuronów).