

Podstawy Sztucznej Inteligencji

Sprawozdanie z Projektu 5

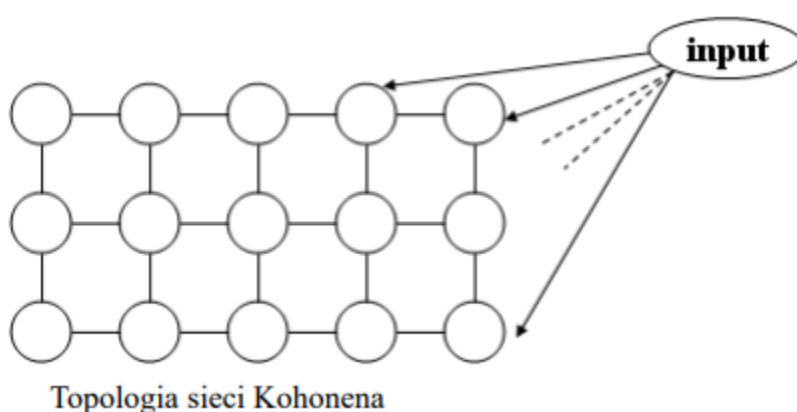
Budowa i działanie sieci Kohonena dla WTA

1. Cel ćwiczenia:

Celem projektu było poznanie budowy i działania sieci Kohonena przy wykorzystaniu reguły WTA do odwzorowania istotnych cech gatunków kwiatów.

2. Część teoretyczna:

Sieć Kohonena - jest szczególnym przypadkiem algorytmu realizującego uczenie się bez nadzoru. Jej 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. Sieć Kohonena znana jest też pod nazwami Self-Organizing Maps, Competitive Filters. Topologia sieci Kohonena odpowiada topologii docelowej przestrzeni. Jeśli np. chcemy prezentować wynik na ekranie, rozsądnym modelem jest prostokątna siatka węzłów (im więcej, tym wyższą rozdzielczość będzie miała mapa).



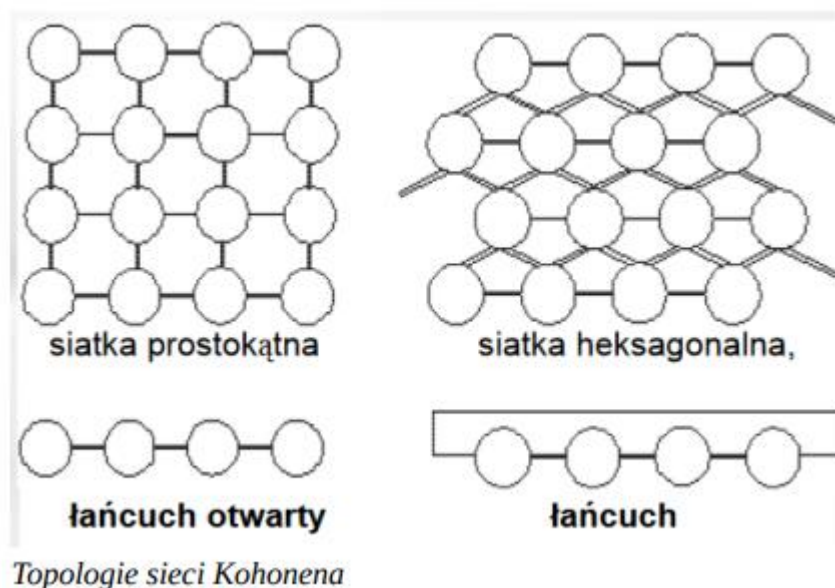
Zasady działania sieci Kohonena:

- 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. Wektory wejściowe stanowią próbę uczącą, podobnie jak w przypadku zwykłych sieci rozpatrywaną w pętli podczas budowy mapy. Wykorzystanie utworzonej w ten sposób mapy polega na tym, że zbiór obiektów umieszczamy na wejściu sieci i obserwujemy, które węzły sieci się uaktywniają. Obiekty podobne powinny trafiać w podobne miejsca mapy.

Winner Takes All (WTA) - zwycięzca bierze wszystko. Po przedstawieniu sieci wektora wejściowego, neuron najbardziej podobny do elementu prezentowanego (którego wagi są najbardziej podobne składowym wektora wejściowego) zostaje zmodyfikowany zgodnie z funkcją f tak, aby jego wagi były jak najbardziej zbliżone do wektora wejściowego. W programie Matlab, za pomocą biblioteki Neural Network Toolbox, zaimplementowałam sztuczną sieć neuronową. Danymi wejściowymi jest zestaw zaimplementowany w oprogramowaniu MATLAB o nazwie `iris_dataset`. Zawiera on opis 4 cech kwiatów irysa, tj. długość i szerokość płatków oraz długość i szerokość działki kielicha.

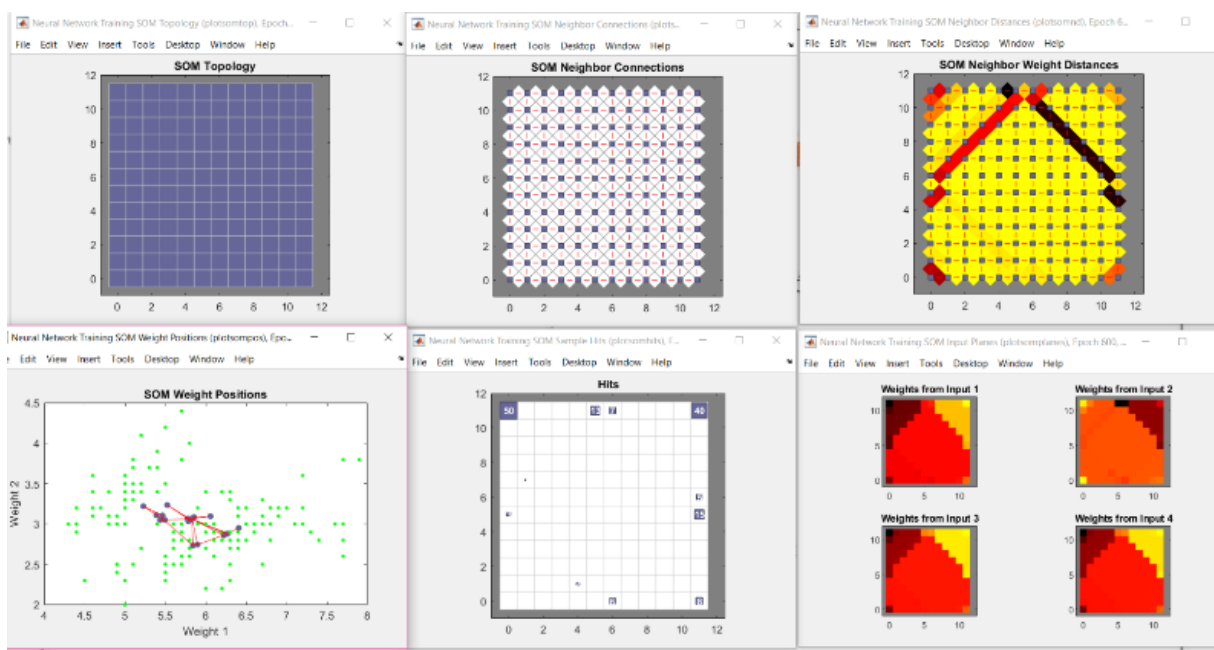
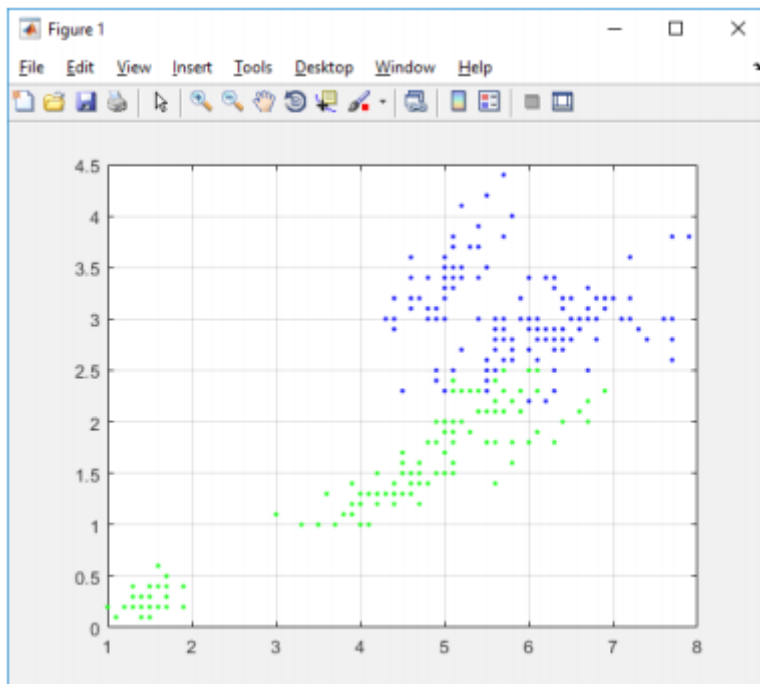
Topologię sieci Kohonena można określić poprzez zdefiniowanie sąsiadów dla każdego neuronu. Jednostka, której odpowiedź na nasze pobudzenie jest maksymalna nazywany obrazem pobudzenia. Sieć jest uporządkowana, jeśli topologiczne relacje między sygnałami wejściowymi i ich obrazami są takie same.



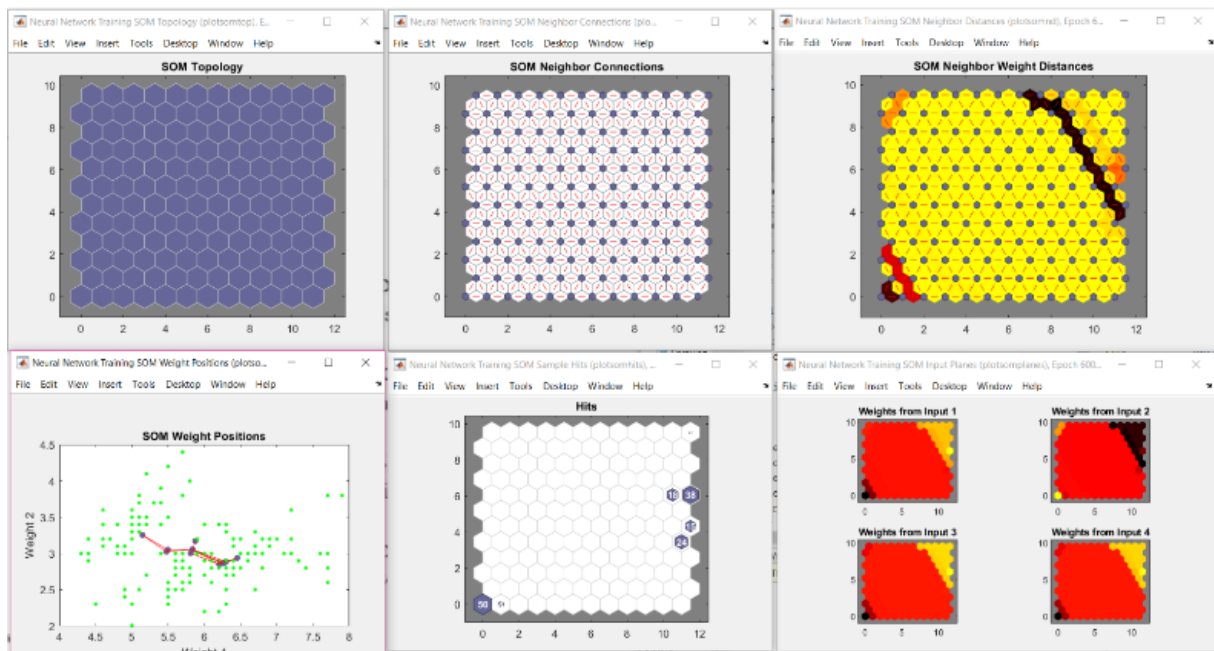
3. Opis wykonanego zadania:

Do stworzenia naszej sieci wykorzystaliśmy pakiet MATLAB, narzędzie Neural Networking Training Tool. Naszymi danymi wejściowymi był wbudowany w pakiecie zbiór danych

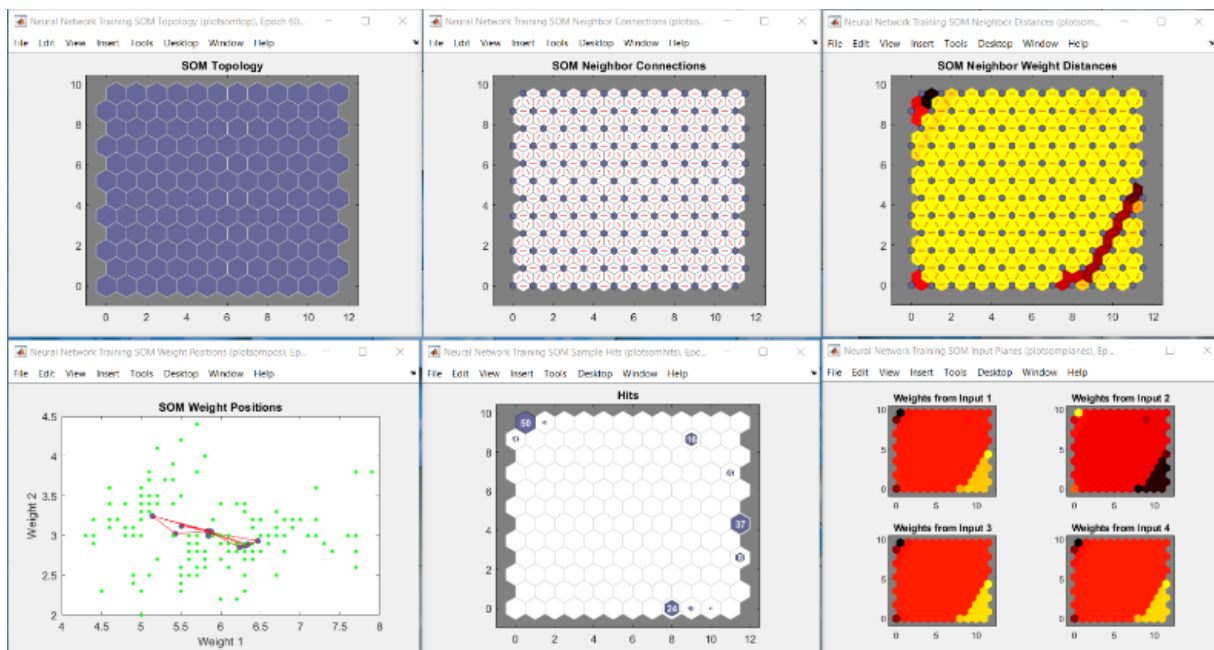
iris_dataset. Udało się również wygenerować dwuwymiarowy wykres dla parametrów płatk i działki kielicha.



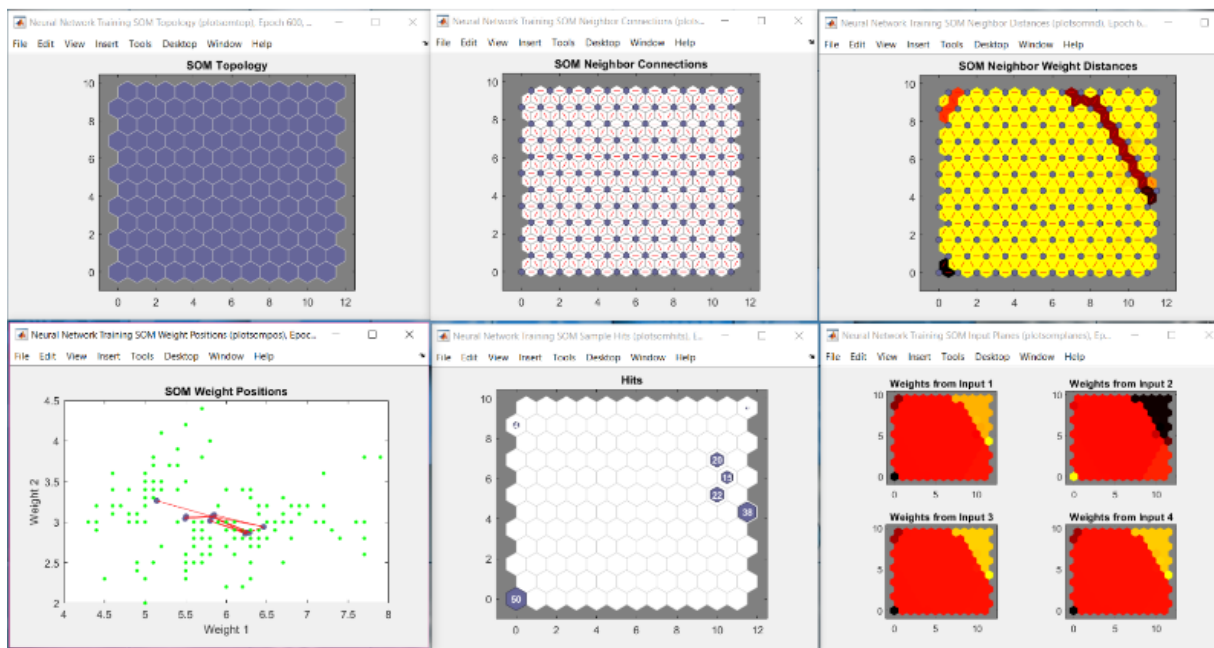
Wykresy dla topologii kwadratowej (wsp. uczenia = 0,5)



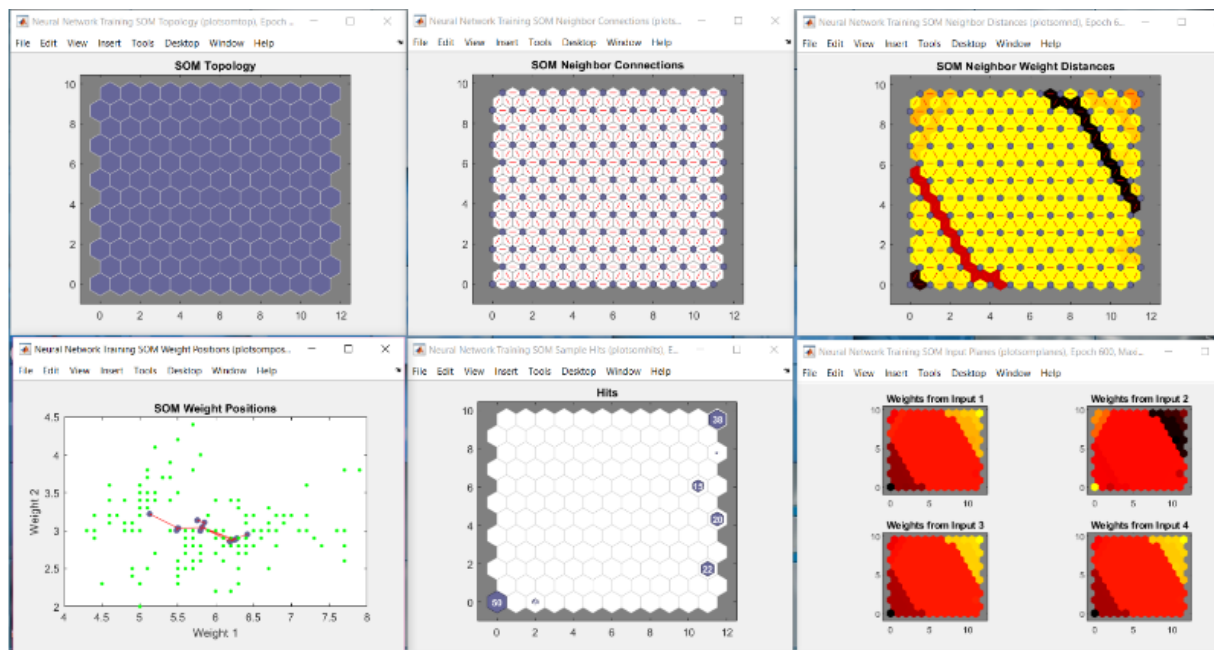
Wykresy dla topologii heksagonalnej (wsp. uczenia = 0,5)



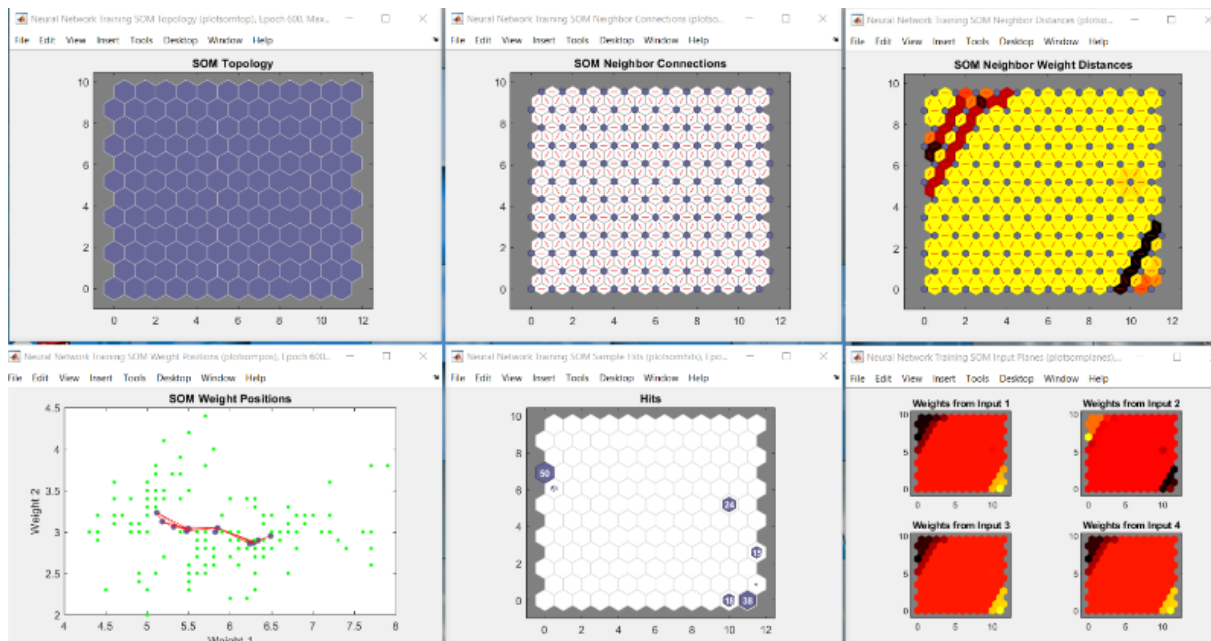
Wykresy dla topologii heksagonalnej (wsp. uczenia = 0)



Wykresy dla topologii heksagonalnej (wsp. uczenia = 0,25)



Wykresy dla topologii heksagonalnej (wsp. uczenia = 0,75)



Wykresy dla topologii heksagonalnej (wsp. uczenia = 0,9)

4. Wnioski:

Bardziej szczegółową informację o możliwych korelacjach i podobieństwach daje nam topologia heksagonalna. Topologia kwadratowa pozwala na podgląd pewnych tendencji, jednak pewne informacje na temat zależności są zbyt uogólnione.

Wynik działania topologii jest podobny, jednak w naszym przypadku topologia heksagonalna doliczyła się mniejszej ilości podobieństw, a co za tym idzie mniej klas danego obiektu.

Algorytmy uczenia bez nauczyciela można wykorzystać tam, gdzie istnieje potrzeba grupowania elementów pod względem pewnych cech. Wykresy dla topologii heksagonalnej (wsp. uczenia = 0,75) Wykresy dla topologii heksagonalnej (wsp. uczenia = 0,9)

Algorytm WTA nie zwraca uwagi na sąsiedztwo – dla niego najważniejsza jest aktualizacja wag. Istotnym czynnikiem może okazać się współczynnik uczenia – w pewnym zakresie kształtują się pewne tendencje – jednak za niski, bądź za wysoki powoduje wytworzenie się zbyt szczegółowego podziału na grupy, co nie zawsze jest naszym celem.

W zależności od współczynnika uczenia otrzymujemy też różne połączenia korelacyjne – im był on wyższy, tym te korelacje zachodziły rzadziej.


5 . Listing kodu wraz z komentarzami:

```
Close
all;
clear
all;
clc;

in_value = iris_dataset; %danymi wejściowymi jest
% zbiór danych, gdzie w I i II kolumnie mamy dane o długości
```

```

% i szerokości działki kielicha, a w III i IV kolumnie
% o długości i szerokości płatka (wszystkie wymiary podane są w cm)
plot(in_value(1, :) ,in_value(2, :), 'b.', in_value(3, :), in_value(4, :), 'g.');
```



```

% narysowanie dwuwymiarowego wykresu zależności
% długości od szerokości
hold on; grid on; %hold - zapamiętanie aktualnego wykresu,
% kiedy tworzą się inne wykresy
% grid - wyświetlanie linii siatek wykresu
dimensions = [12 12]; % wymiary wektora
coverstep = 50; %etapy szkolenia w celu pokrycia przestrzeni wejściowej
initNeighbor = 0; % wejściowy rozmiar sąsiedztwa
topologyFcn = 'hextop'; %funkcja topologiczna -> kształt, jaki będą przyjmować nasze dane
% mogą przyjmować kształt trójkąta, siatek kwadratowych, sześciokątów, itp.
distanceFcn = 'dist'; %funkcja dystansu neuronów - miara euklidesowa (znormalizowana)
% domyślnym parametrem jest odległość między neuronami warstwy z
% uwzględnieniem ich położenia
net = selforgmap(dimensions, coverstep, initNeighbor, topologyFcn, distanceFcn); %tworzenie mapy
samoorganizacji
net.trainFcn = 'trainbu'; %uczenie bez nauczyciela
net.trainParam.epochs = 600;
net.trainParam.lr = 0.9; %współczynnik uczenia
[net, tr] = train(net, in_value); %trening sieci
y = net(in_value); %testowanie i zapis wyników osiągniętych przez sieć
```