

# Podstawy sztucznej inteligencji

Kinga Synowiec, Inżynieria obliczeniowa, grupa 2

## Sprawozdanie nr 5

- **Temat ćwiczenia:**

Budowa i działanie sieci Kohonena dla WTA.

- **Cel ćwiczenia:**

Poznanie budowy i działania sieci Kohonena przy wykorzystaniu reguły WTA do odwzorowywania istotnych cech kwiatów.

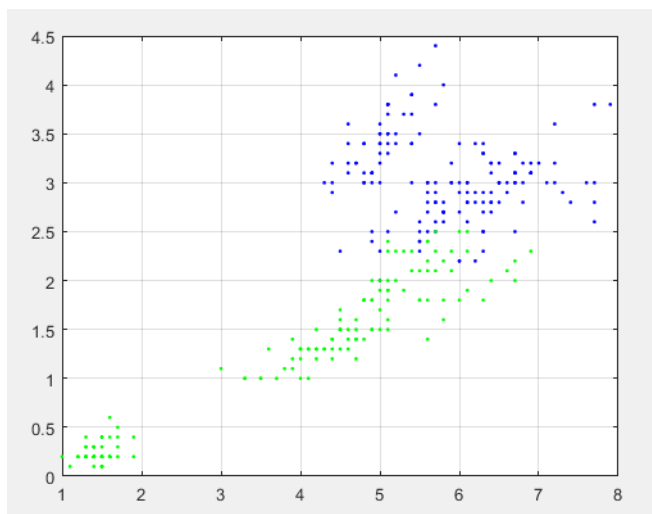
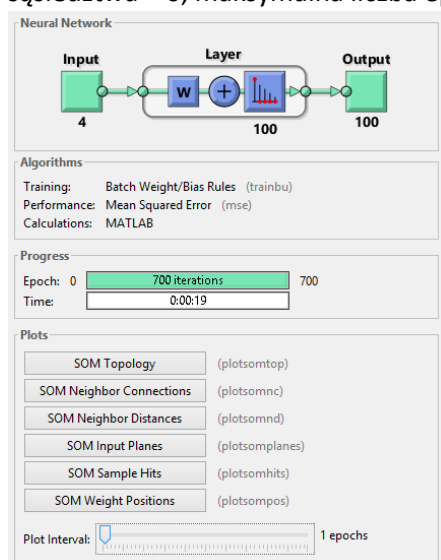
- **Wstęp teoretyczny:**

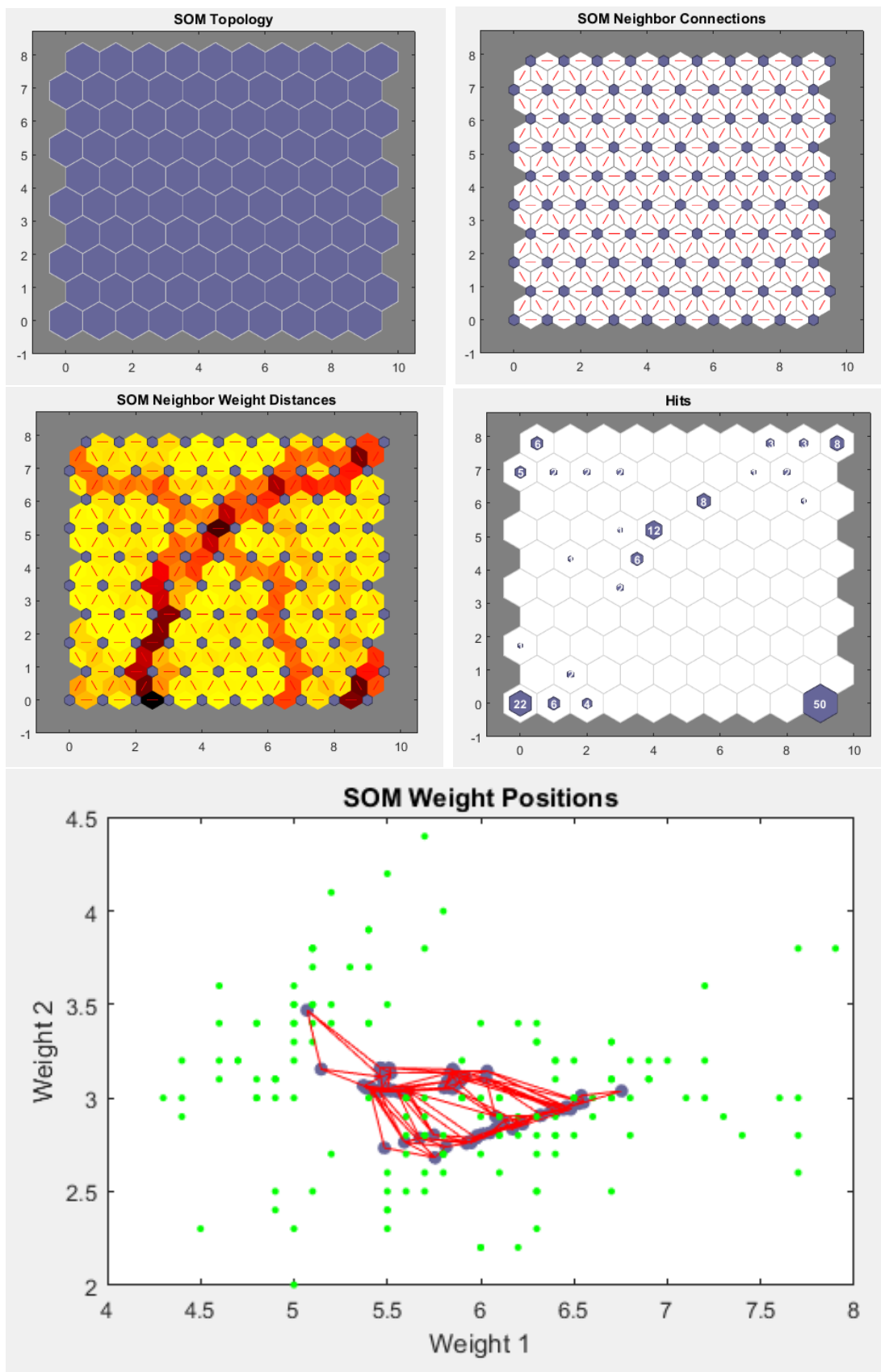
- WTA (Winner Takes All) – jedna z konkurencyjnych metod uczenia sieci, w której tylko jeden element wyjściowy może znajdować się w stanie aktywnym. Nazywany jest on zwycięzcą. Każda komórka w takiej warstwie jest połączona ze wszystkimi elementami wzorca wejściowego za pomocą połączeń wagowych. Komórka zwycięska to neuron otrzymujący najsilniejszy sygnał pobudzenia.

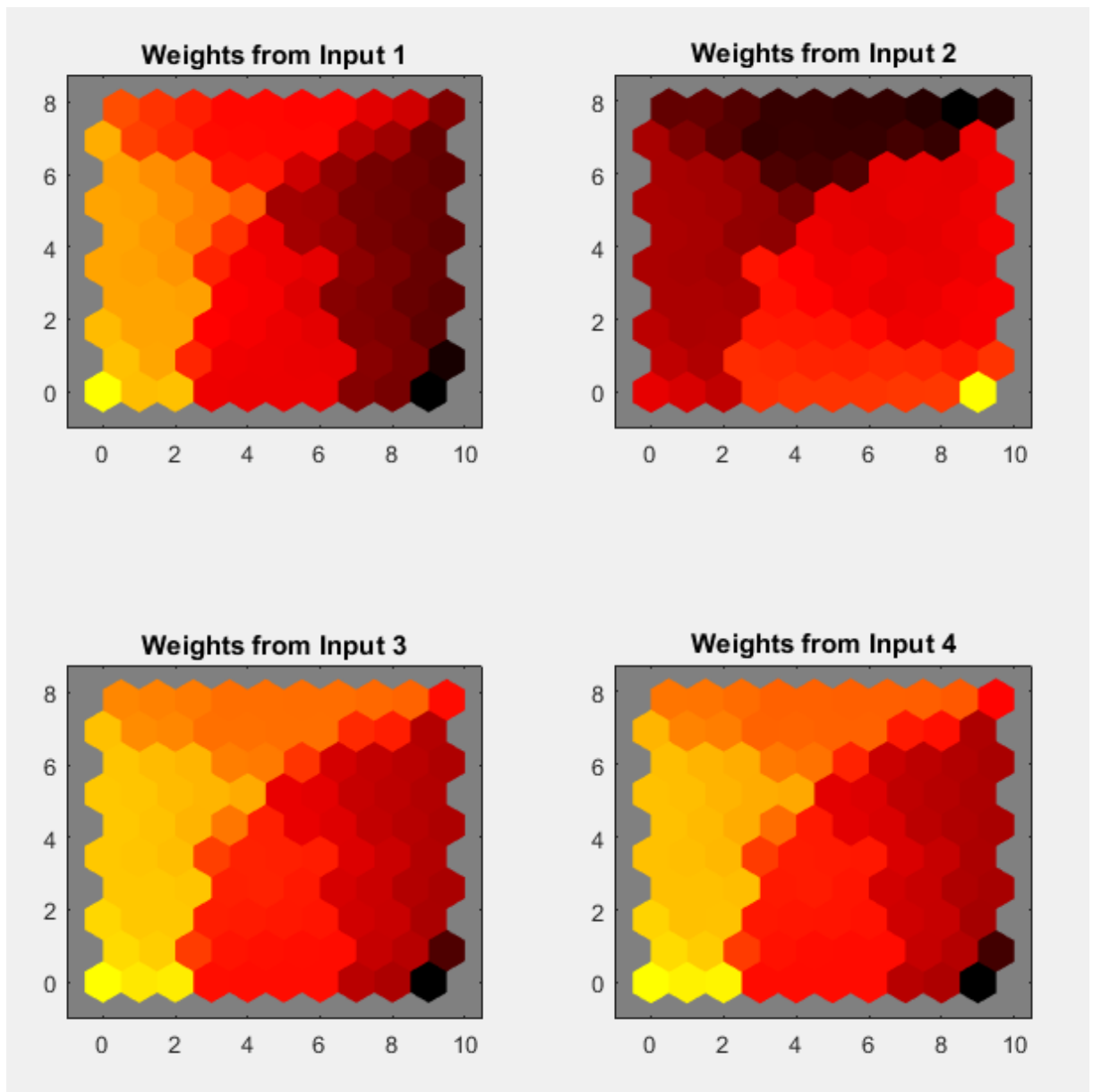
- Sieć Kohonena - sieć neuronowa uczona w trybie bez nauczyciela w celu wytworzenia niskowymiarowej (przeważnie dwuwymiarowej) zdyskretyzowanej reprezentacji przestrzeni wejściowej. Sieć Kohonena wyróżnia się tym od innych sieci, że zachowuje odwzorowanie sąsiedztwa przestrzeni wejściowej. Wynikiem działania sieci jest klasyfikacja przestrzeni w sposób grupujący zarówno przypadki ze zbioru uczącego, jak i wszystkie inne wprowadzenia po procesie uczenia.

//źródło : [https://pl.wikipedia.org/wiki/Sie%C4%87\\_Kohonena](https://pl.wikipedia.org/wiki/Sie%C4%87_Kohonena)

- **Zrzuty ekranu przedstawiające rezultaty po uruchomieniu programu przy ustawionych parametrach:** liczba kroków szkoleniowych = 100, początkowy rozmiar sąsiedztwa = 0, maksymalna liczba epok = 700:







- **Opis przedstawionych wykresów:**

- SOM Topology – sześciokąty to reprezentacje neuronów. W każdym wektorze wejściowym znajdują się cztery elementy, dlatego cała przestrzeń jest czterowymiarowa.
- SOM Neighbor Connections – mapa przedstawiająca powiązania sąsiedzkie.
- SOM Neighbor Distance – neurony oznaczone niebieskimi sześciokątami, sąsiednie neurony połączone są czerwonymi liniami. Kolory opisują odległości pomiędzy nimi – ciemniejsze to odległości większe, a jaśniejsze mniejsze.
- SOM Sample Hits – mapa przedstawiająca, ile razy dany neuron zwyciężył podczas rywalizacji.
- SOM Input Planes – płaszczyzna ciężkości dla każdego elementu wektora wejściowego. Mapa przedstawia wizualizację wag, które łączą każde wejście z każdym z neuronów. Im ciemniejszy kolor, tym większa waga.

- **Przebieg działania programu:**

- Wygenerowanie losowo znormalizowanych wektorów wag.
- Wylosowanie wektora X oraz dla wszystkich neuronów obliczenie dla niego aktywacji Y.
- Szukanie neuronu zwycięzcy.
- Zmodyfikowanie wektora wag neuronu zwycięzcy i jego normalizacja poprzez sprawdzenie warunków WTA.

- **Wnioski:**

- Dzięki analizie rozkładu barw na wykresie można określić wygląd irysa według szkolonej sieci neuronowej. Należy skoncentrować się na kolorach ciemniejszych, gdyż to one najlepiej oddają cechy irysa.
- Na podstawie wykresu przedstawiającego rozkład sił neuronów można stwierdzić, że sieć korzystała z mechanizmu WTA, ponieważ gdyby nie ten fakt, prawdopodobnie najmocniejsze neurony za każdym razem by zwyciężały, nie dopuszczając innych neuronów do udziału.
- Analizując wykresy kolorów można zauważyć, że sieć zdefiniowała jako irysy te punkty, których parametry były bliskie lub równe wartościom średnim. Wartości te określone były jako typowe dla irysów.
- Porównując do siebie wykresy można stwierdzić, że występują pomiędzy nimi powiązania. Rozkład ciemnych barw na wykresie Neighbor Weight Distances pokrywa się z miejscami, gdzie na wykresie Hits występują neurony zwycięskie.
- Podczas tworzenia sieci ważne było dobranie odpowiedniej liczby neuronów tak, aby nie była za mała, gdyż wtedy rośnie ryzyko błędów, ani za duża, co wydłużyłoby czas uczenia.

- **Listing całego kodu programu wykonanego w MATLABie wraz z opisem użytych funkcji:**

```
%%
close all; clear all; clc;
WEJSCIE = iris_dataset; %dane uczące, sa to wartości implementowane przez
oprogramowanie
size(WEJSCIE); %okreslenie rozmiaru tablicy przez przypisanie danych
wejsciovych
plot(WEJSCIE(1, :), WEJSCIE(2, :), 'b.', WEJSCIE(3, :), WEJSCIE(4, :),
'g. ');
hold on; grid on; %do wyświetlenia (trzymanie wykresu i siatka)
% PARAMETRY SIECI KOHONENA
dimensions = [10 10]; %wektor rzędów wymiarów
coverSteps = 100; %liczba kroków szkoleniowych dla początkowego pokrycia
przestrzeni wejściowej
initNeighbor = 0; %początkowy rozmiar sąsiedztwa
topologyFcn = 'hextop'; %funkcja topologii warstw, hextop - funkcja
topologii szesciokatnej
distanceFcn = 'dist'; %funkcja odległości neuronowej, dist - funkcja wagi
odleglosci euklidesowej
% TWORZENIE SIECI KOHONENA
net = selforgmap(dimensions, coverSteps, initNeighbor, topologyFcn,
distanceFcn); %zmienna, do ktorej przypisywana jest nowo tworzona siec
neuronowa za pomocą algorytmu Kohonena z wykorzystaniem wymienionych
parametrow
net.trainParam.epochs = 1000; % ustalenie maksymalnej liczby epok
treningowych utworzonej sieci
```

```
% TRENING SIECI
[net, tr] = train(net, WEJSCIE);
y = net(WEJSCIE);
grid on
```