

Maciej Krzywda,
Inżynieria Obliczeniowa, IMiP
Podstawy Sztucznej Inteligencji
nr albumu: 293102

Sprawozdanie 5

Tytuł projektu:

Budowa i działanie sieci Kohonena dla WTA

- **Cel projektu:**

Celem ćwiczenia jest poznanie budowy i działania sieci Kohonena przy wykorzystaniu reguły WTA do odwzorowania istotnych cech kwiatów

- **Przebieg ćwiczenia**

1. Przygotowanie danych uczących zawierających numeryczny opis cech kwiatów. przykładowy zestaw:
https://en.wikipedia.org/wiki/Iris_flower_data_set b)
2. Przygotowanie (implementacja lub wykorzystanie gotowych narzędzi) sieci Kohonena i algorytmu uczenia opartego o regułę Winner Takes All (WTA).
3. Uczenie sieci dla różnych współczynników uczenia i zapominania. współczynnika zapominania.
4. Uczenie sieci dla różnych współczynników uczenia i zapominania.
5. Testowanie sieci.

- **Część Teoretyczna**

Sieć neuronowa - Nazwa struktur matematycznych i ich programowych lub sprzętowych modeli. Wykonuje obliczenia za pomocą powiązanych ze sobą elementów (neuronów). Sieć neuronowa została zaprojektowana w taki sposób by przypominać działanie ludzkiego mózgu złożonego z naturalnych neuronów i łączących je synaps.

Sieć jednokierunkowa - Jest to sieć neuronowa, w której nie występuje zjawisko sprzężenia zwrotnego oznacza to, iż każdy sygnał przechodzi przez każdy neuron dokładnie raz w swoim cyklu.

Sieć Kohonena - Jest podstawowym typem sieci samoorganizującym się. Metodyka nauki sieci samoorganizującej różni się tym, iż nie podajemy Jej wzorca do jakiego dążymy, a suche dane, które musi interpretować sama. Uczenia polega na wybieraniu w drodze konkurencji zwycięskiego neuronu, który najlepiej

odpowiada wektorowi wejściowemu. System ten w zbliżeniu bardzo przypomina uczenie się przez ludzi.

Etapy działania takiej sieci to.:

- Konstrukcja
- Uczenie
- Rozpoznawanie

Algorytm uczenia WTA:

Przyjmujemy losowe, znormalizowane wartości wag poszczególnych neuronów

Po podaniu pierwszego wektora wejściowego x wyłaniany jest zwycięzca o numerze v .

$$w_v^T x = \max_{i=1,2,\dots,k} (w_i^T x)$$

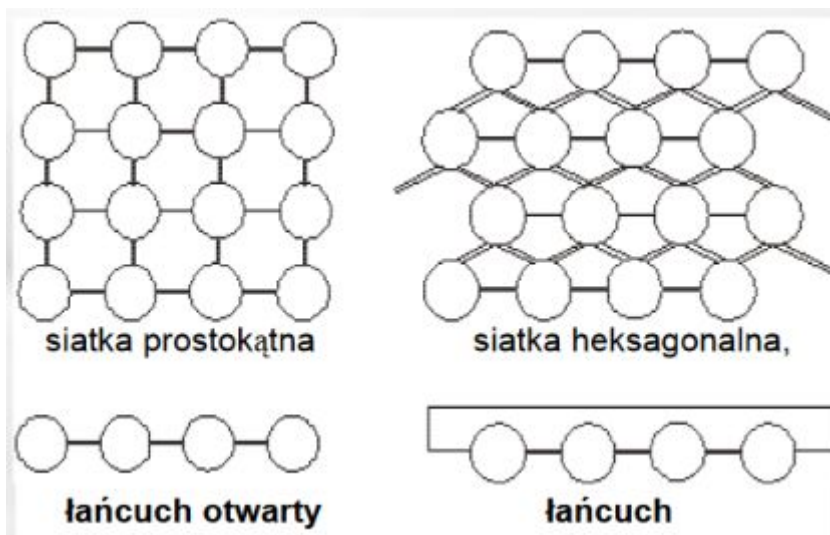
Aktualizacja wag neuronu zwycięzcy (neurony przegrywające mają na wyjściu stan 0, co blokuje proces aktualizacji ich wag). Aktualizacja ta odbywa się według reguły Kohenena:

$$w_v(t+1) = w_v(t) + \eta[x - w_v(t)]$$

gdzie η - wsp. uczenia ($0 < \eta < 1$) i maleje w miarę postępu nauki.

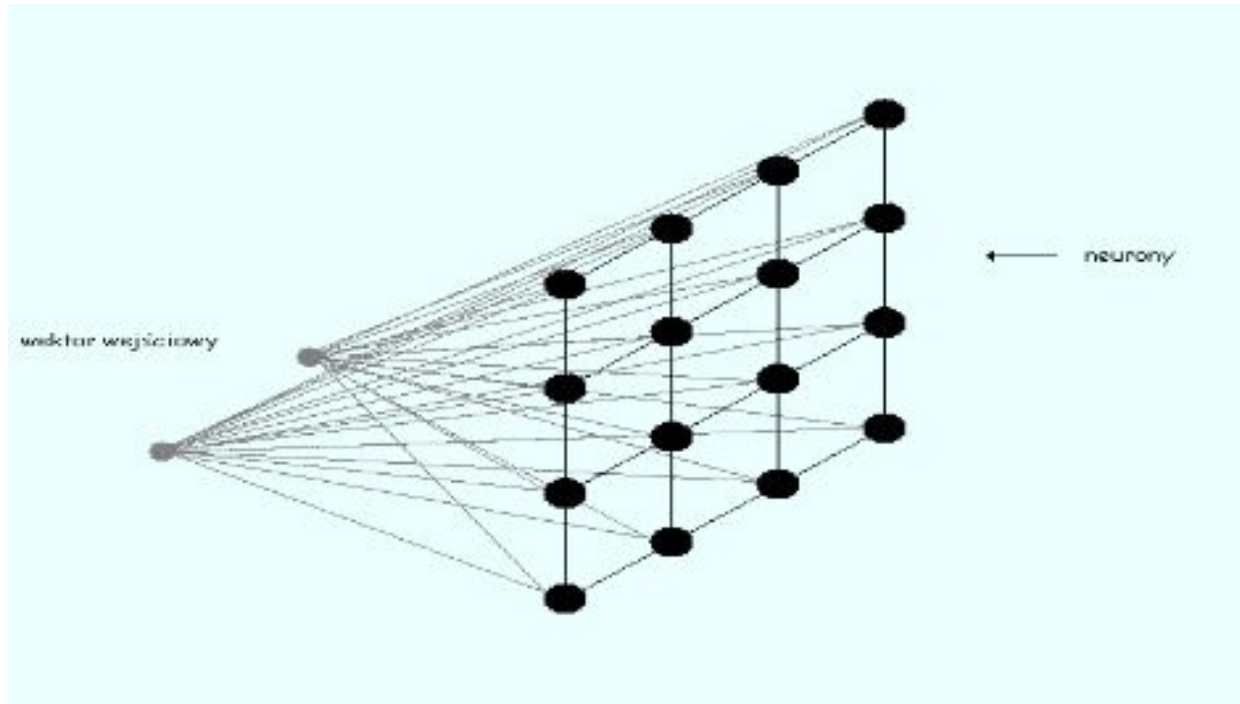
Wektor wag neuronu zwycięzcy jest zwiększany o ułamek różnicy $x-w$, w wyniku czego w następnych krokach lepiej odtwarza rozpatrywany wektor wejściowy.

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.



Ważnym parametrem jest określenie ile neuronów obok ma podlegać uczeniu w przypadku zwycięstwa danego neuronu

Powstawanie prostokątnej dwuwymiarowej sieci Kohonena



Wykorzystaliśmy funkcję do tworzenia map samoorganizacji *selforgmap*. W ramach wykonywania programu otrzymaliśmy 6 różnych wykresów, które mają swoje znaczenie:

- *SOM Topology* - jeden kształt symbolizuje jeden neuron. Są one rozmieszczone w siatce o określonych wymiarach. Ułożenie jest nieprzypadkowe - ich sąsiedztwo może wskazywać na ich podobieństwo.
- *SOM Neighbor Distances* - Na tej mapie otrzymamy możliwość sprawdzenia, jak bardzo połączone są ze sobą poszczególne neurony - czyli jak silne podobieństwo one wykazują. Im jaśniejsze połączenie, tym bardziej te dane są do siebie podobne. Ciemne linie mogą zatem oznaczać granice klas.
- *SOM Neighbor Connections* - na tym wykresie umieszczone są połączenia między sąsiadami. Sąsiadami są zwykle próbki do siebie podobne.
- *SOM Weight Planes* - jest to zestaw wykresów, który wskazuje na rozkład wag poszczególnych neuronów w zależności od cechy. Im ciemniejszy kolor, tym większą wagę dany neuron skupia. Im więcej neuronów o podobnych kolorach w danym sąsiedztwie, tym te neurony są bardziej ze sobą skorelowane

- *SOM Sample Hits* - ten wykres pokazuje nam, ile podobnych wyników otrzymaliśmy dla danej klasy- im wyższa liczba, tym więcej obiektów o podobnych cechach można wykazać
- *SOM Weight Positions* - zielone kropki oznaczają dane wejściowe, a linie je łączące wskazują na korelacje pomiędzy poszczególnymi neuronami.

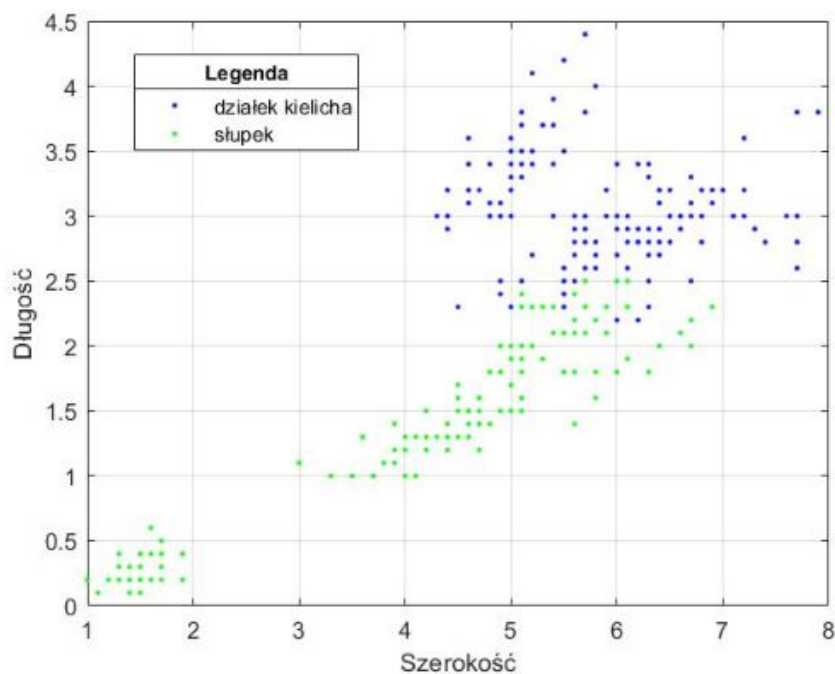
● Przebieg ćwiczenia

- Zaimportowanie z biblioteki programu MatLab „iris_dataset”, zawierającej dane na temat kwiatów typu Iris.

<https://se.mathworks.com/help/deeplearning/examples/iris-clustering.html>

Dane te odpowiednio przedstawiają.:

- Szerokość działki kielicha kwiatu
- Szerokość płatków kwiatowych - Wysokość działki kielicha kwiatu
- Wysokość płatków kwiatowych
- Dla trzech rodzajów kwiatów.:
- Setosa
- Virginica
- Versicolor
- Zostało wykonane po 50 pomiarów, co oznacza iż mamy 4 rodzaje danych x 3 rodzaje kwiatów x 50 pomiarów, co daje nam dane wejściowe jako 4 x 150, które przechowywane są w postaci macierzy
- Dane z biblioteki są przechowywane w zmiennej „Data” - Do przeprowadzenia ćwiczenia posłużyłem się funkcją Kohonena przy wykorzystaniu reguły WTA opisanej powyżej
- Sieć została stworzona z domyślnymi wymiarami 8 x 8 w kształcie sześciokątów, natomiast początkowy rozmiar sąsiedztwa ustawiony jest na 0, liczba epok = 1000

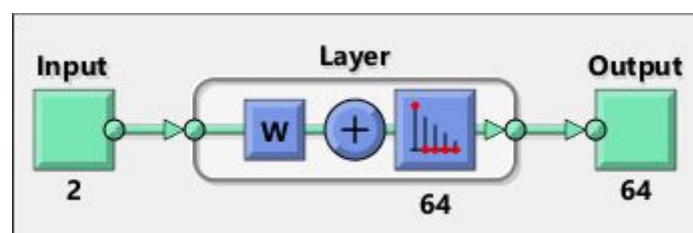


selforgmap(dimensions,coverSteps,initNeighbor,topologyFcn,distanceFcn)

Samoorganizujące się mapy uczą się klastra danych na podstawie podobieństwa, topologii, z preferencją (ale bez gwarancji) przypisywania tej samej liczby wystąpień do każdej klasy.

Samoorganizujące się mapy są używane zarówno do grupowania danych, jak i do zmniejszania wymiarów danych. Inspirują się sensorycznymi i motorycznymi mapowaniami w mózgu ssaka, które również wydają się automatycznie organizować informacje topologicznie.

dimensions	Row vector of dimension sizes (default = [8 8])
coverSteps	Number of training steps for initial covering of the input space (default = 100)
initNeighbor	Initial neighborhood size (default = 3)
topologyFcn	Layer topology function (default = 'hextop')
distanceFcn	Neuron distance function (default = 'linkdist')



Schemat działania sieci w programie Matlab

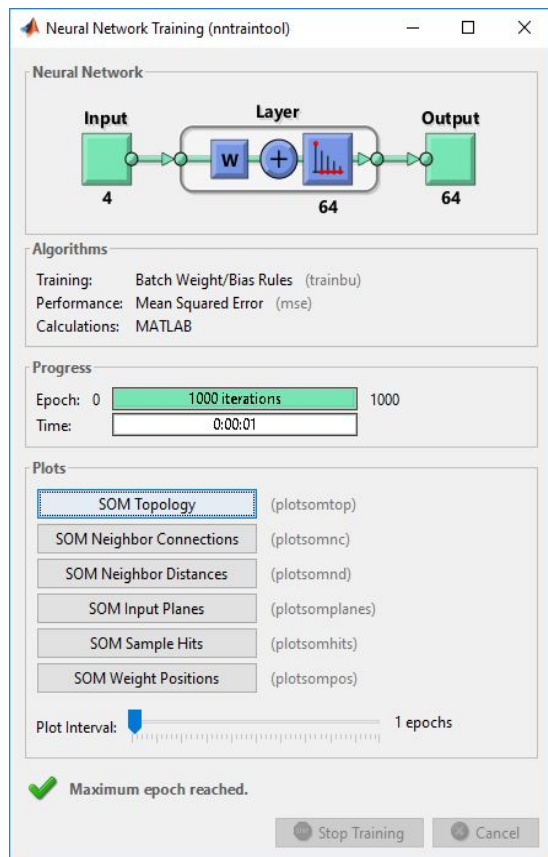
- Listing programu

```
close all; clear all; clc;
Data = iris_dataset;
size(Data);
plot(Data(1,:),Data(2,:), 'g. ');
hold on; grid on;
dimensions = [8 8];
coverSteps = 100;
initNeighbor = 0;

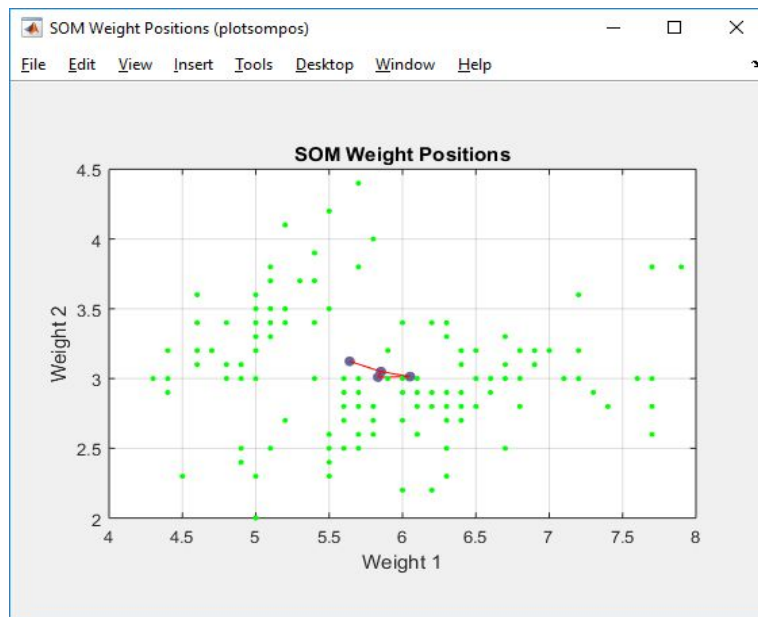
%topologyFcn = 'gridtop';
topologyFcn = 'hextop';
distanceFcn = 'dist';
net =
selforgmap(dimensions,coverSteps,initNeighbor,topologyFcn,distanceFcn);
net.trainFcn = 'trainbu';
net.trainParam.epochs = 1000;

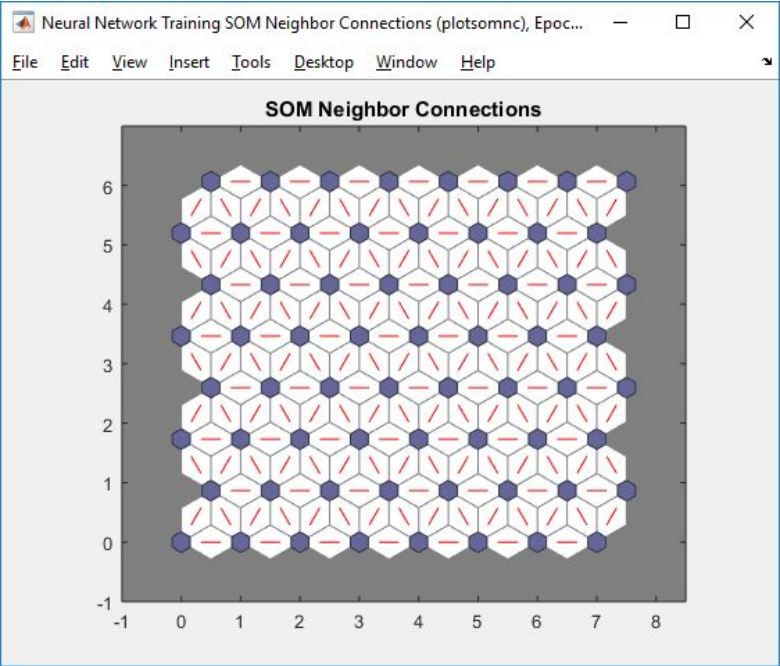
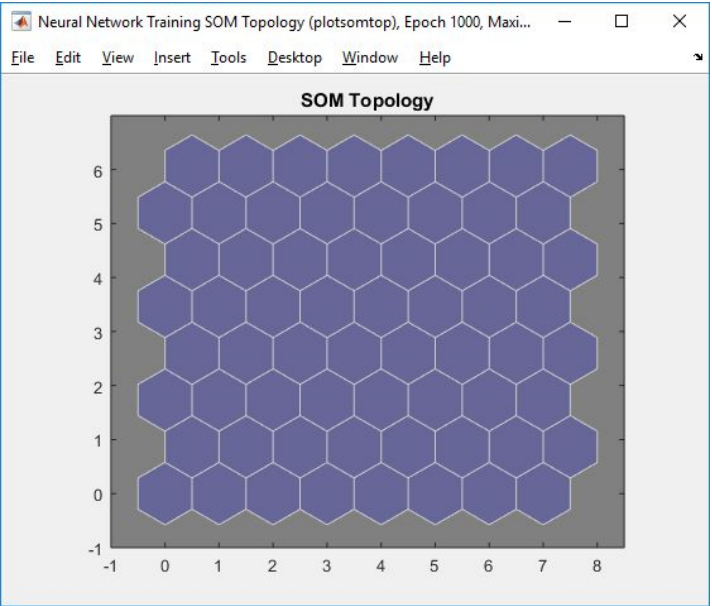
[net,tr] = train(net,Data);
y = net(Data);
plotsompos(net,Data);
grid on
```

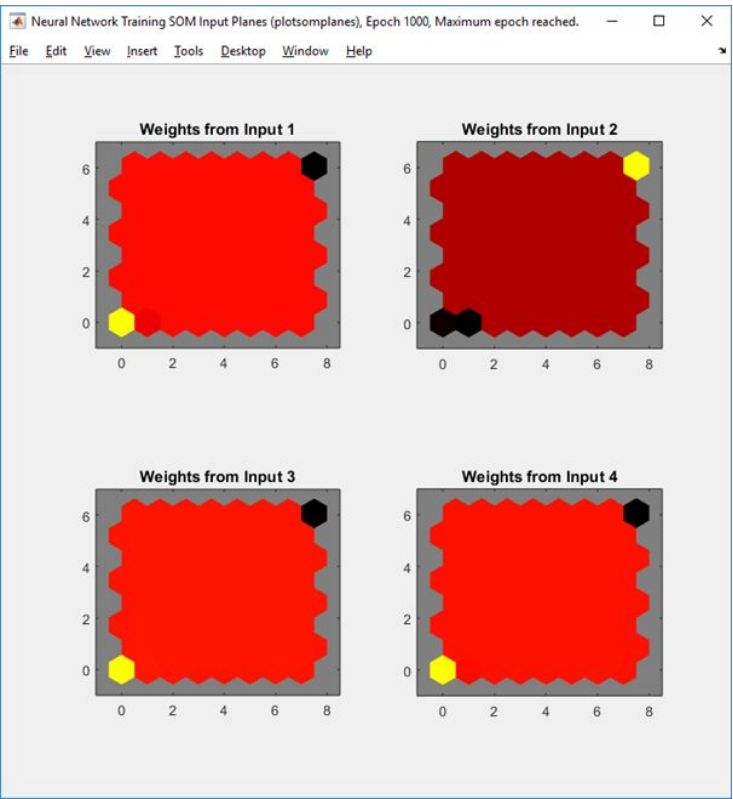
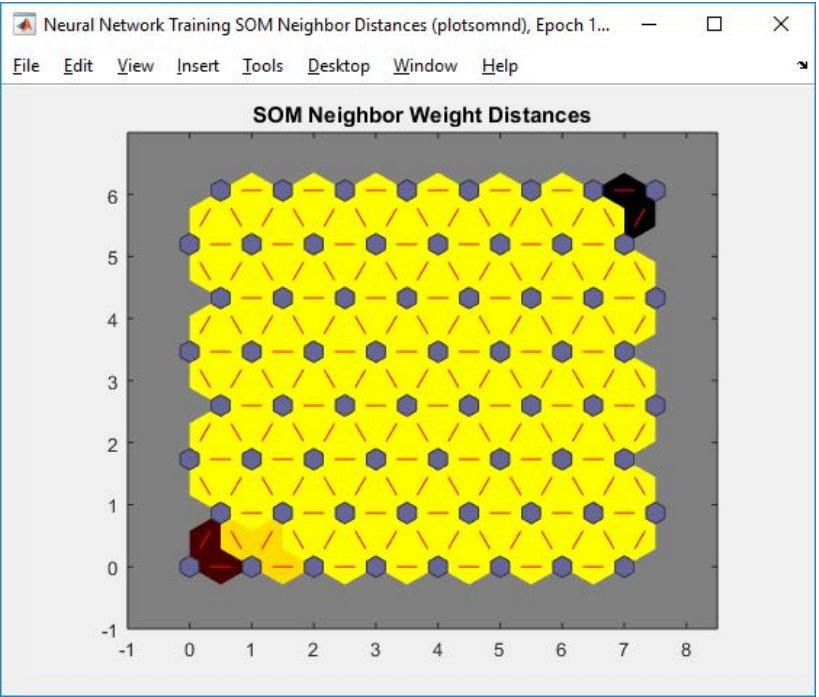
- Analiza Wyników

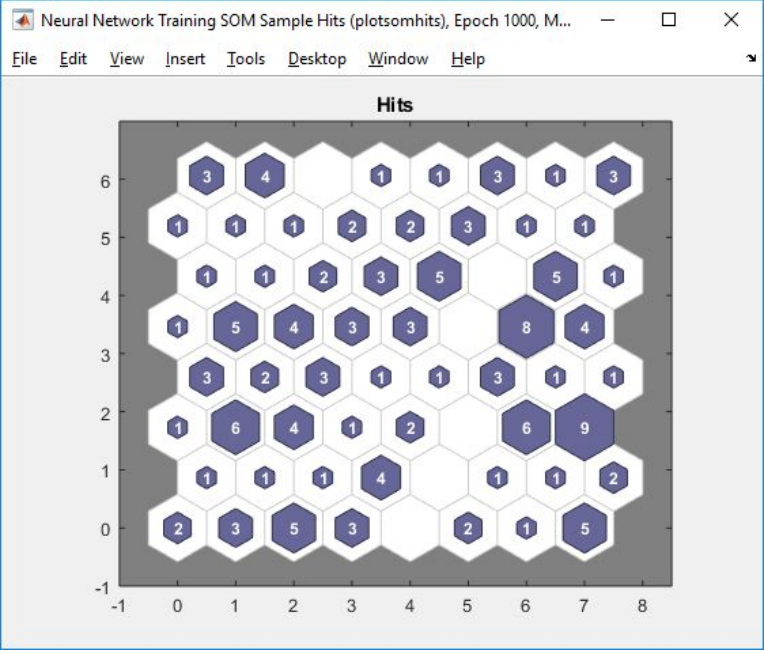


1.dla wersji Hexagonalnej

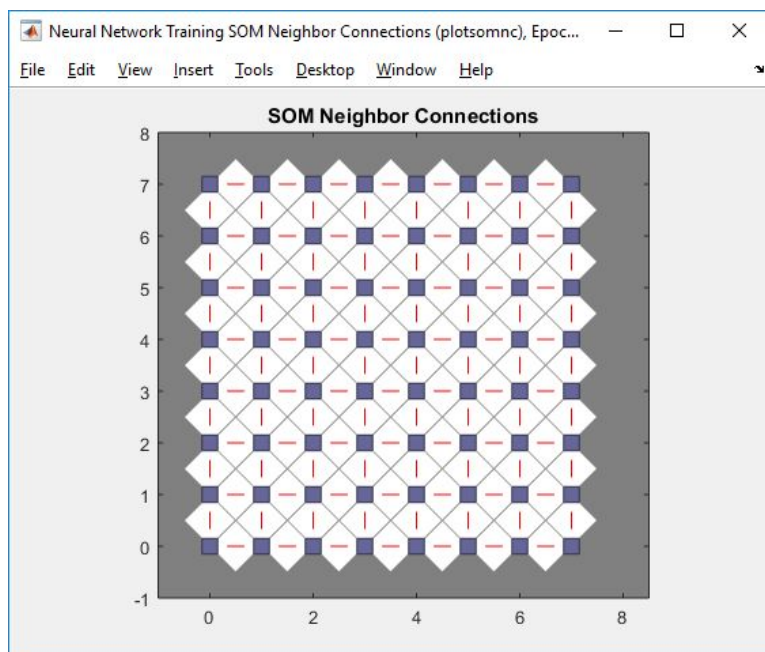
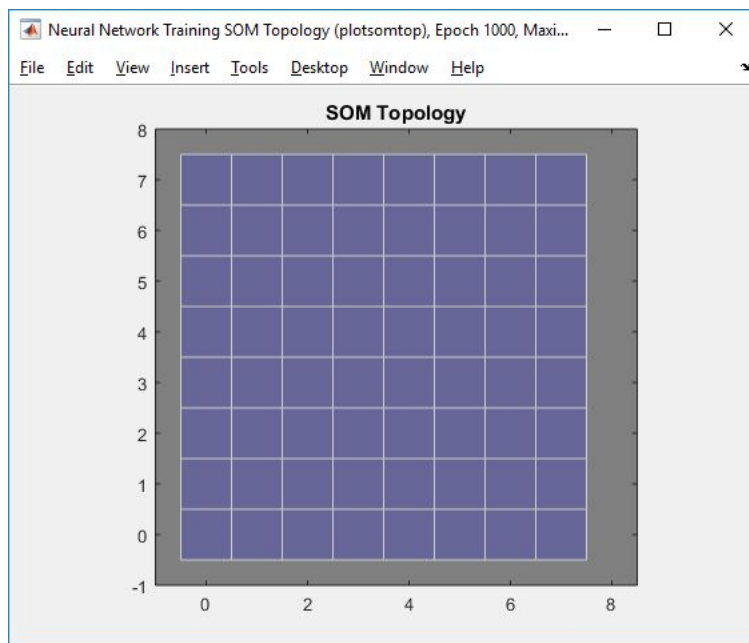


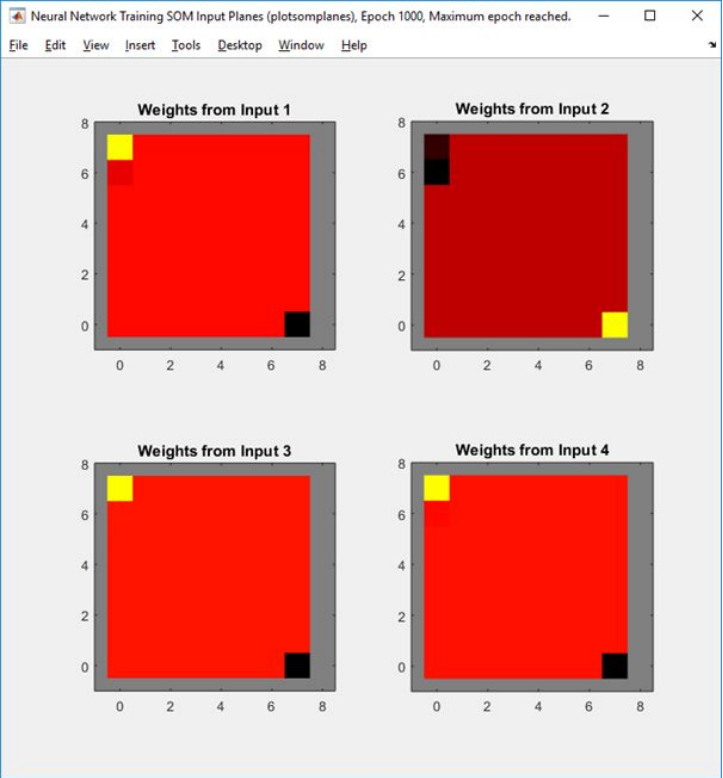
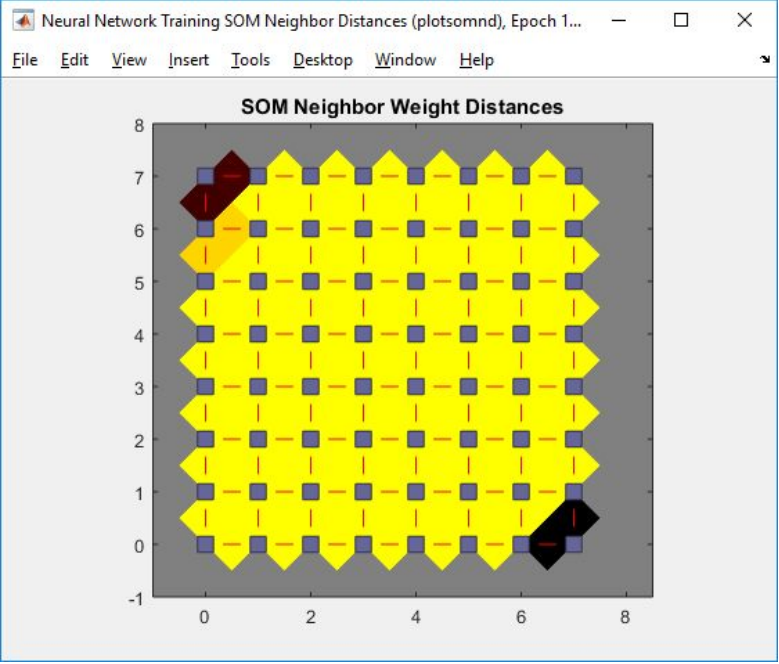


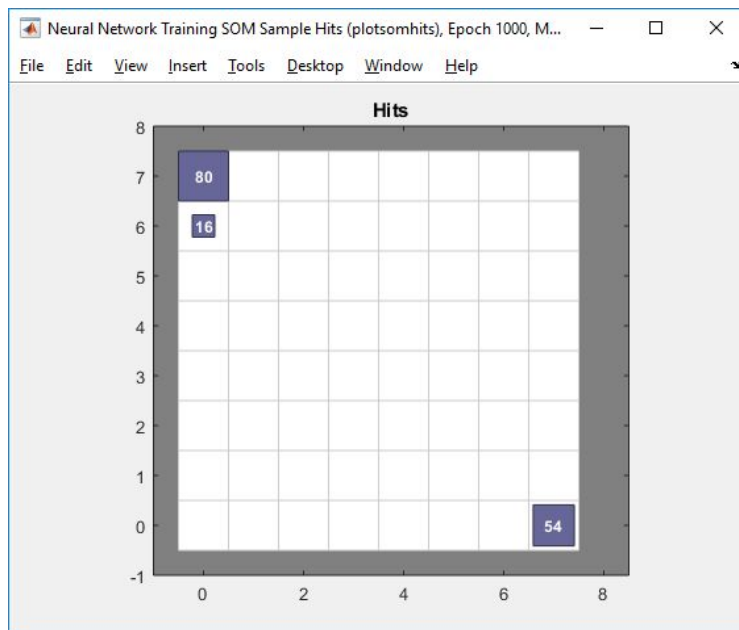




2.dla wersji GridTop







W wyniku analizy otrzymanych wyników możemy zdecydowanie stwierdzić iż topologia Hexagonalna posiada więcej połączeń pomiędzy neuronami niż topologia typu GridTop, co przy większym rozkładzie może wpłynąć na dobór odpowiednich wag dla poszczególnych neuronów. Analizując rozłożenie wag wyznaczonych przez algorytm możemy założyć iż wynikami są najmocniejsze neurony wyselekcjonowane przez WTA, niemniej jednak jeżeli zrezygnujemy z ograniczeń do zwycięskiego neuronu (WTA) i skorzystamy z samej sieci Kohonena ustalając początkowy rozmiar sąsiedztwa jako 3 otrzymamy znacznie bogatszy wysyp danych, przynajmniej dla zadanego tematu.

● Wnioski

Algorytmy uczenia bez nauczyciela można wykorzystać tam, gdzie istnieje potrzeba grupowania elementów pod względem pewnych cech

Algorytm WTA nie zwraca uwagi na sąsiedztwo – dla niego najważniejsza jest aktualizacja wag.

Istotnym czynnikiem może okazać się współczynnik uczenia. Jednak za niski, bądź za wysoki powoduje wytworzenie się zbyt szczegółowego podziału na grupy. Im wyższy był współczynnik uczenia, tym czas uczenia był wyższy.

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

W pewnych przedziałach rozkład wag dla poszczególnych klas jest podobny, co pozwala domniemywać, że w tych neuronach kumulują się pewne dominujące cechy.

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

Wynik działania topologii jest podobny, jednak w naszym przypadku topologia heksagonalna dol