



11.12.2017r.

Akademia Górniczo-Hutnicza
im. Stanisława Staszica w Krakowie

Podstawy sztucznej inteligencji

Sprawozdanie numer 5

Budowa i działanie sieci Kohonena dla WTA

Inżynieria Obliczeniowa
Urszula Ślusarz

nr. indeksu: 286132

1. Cel ćwiczenia

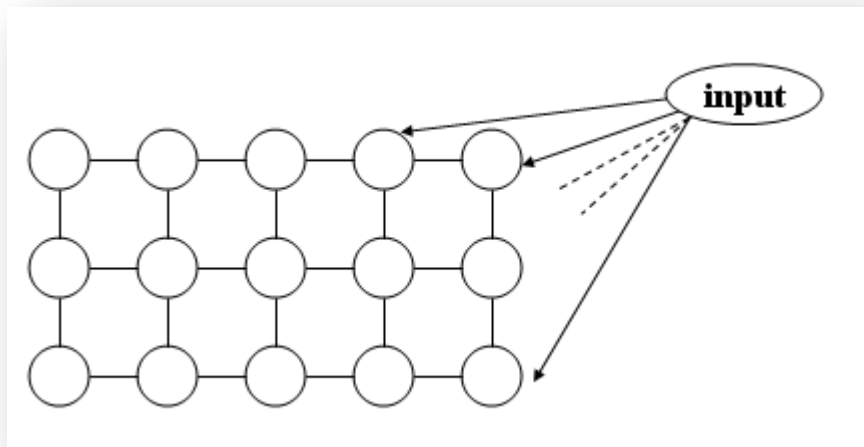
Celem ćwiczenia jest poznanie budowy i działania sieci Kohonena przy wykorzystaniu reguły WTA do odwzorowywania istotnych cech kwiatów.

2. Samoorganizująca się sieć Kohonena

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 (SOM)**.

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:

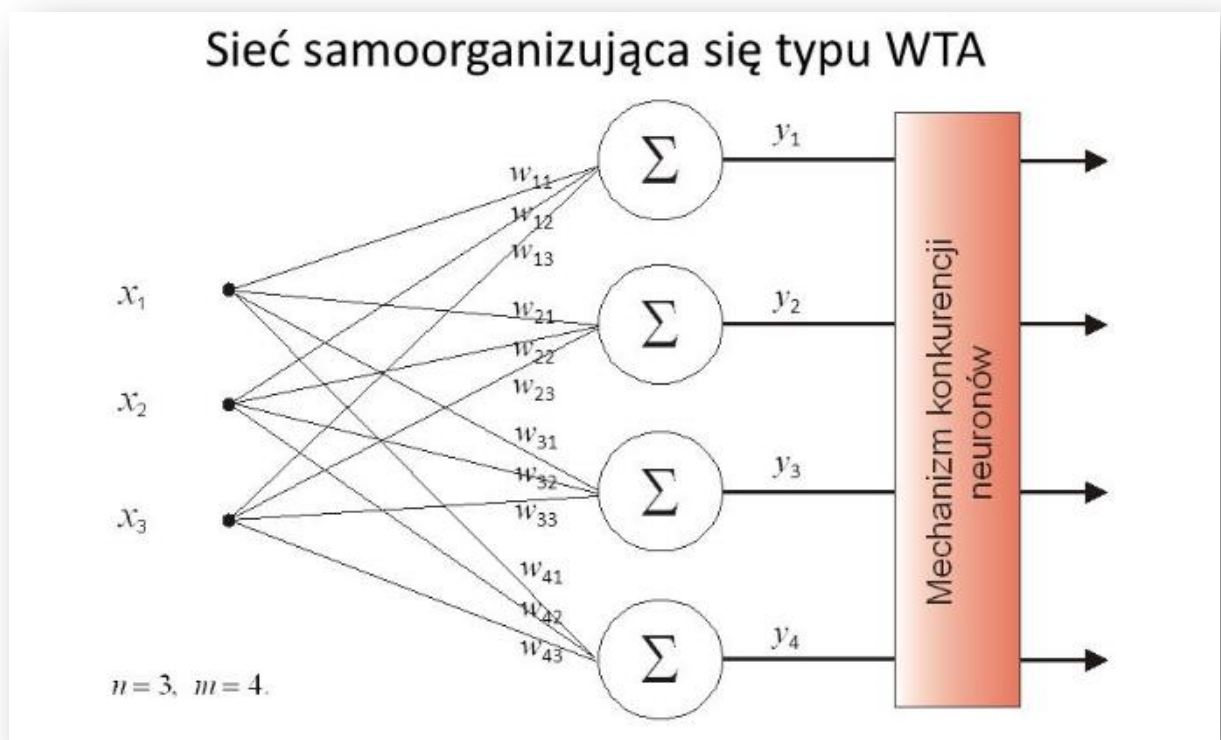
- Wejścia (tyle, iloma parametrami opisano obiekty) połączone są ze wszystkimi węzłami sieci
- Każdy węzeł przechowuje wektor wag o wymiarze identycznym z wektorami wejściowymi
- Każdy węzeł oblicza swój poziom aktywacji jako iloczyn skalarny wektora wag i wektora wejściowego (podobnie jak w zwykłym 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.

Algorytm WTA

Algorytm WTA polega na obliczaniu aktywacji każdego neuronu, a następnie wyborze zwycięzcy o największym sygnale wyjściowym.

Sieć WTA jest siecią jednowarstwową należącą do grupy sieci samoorganizujących się z konkurencją. Ciąg uczący się składa się jedynie z wektorów wejściowych (sieć bez nauczyciela). Istotą działania sieci jest współzawodnictwo neuronów ze sobą – jeden z nich wygrywa przy prezentacji pojedynczego wektora – stąd akronim WTA (ang. Winner Takes All – „zwycięzca bierze wszystko”). Reguła „wygrywający bierze wszystko” przyznaje więc prawo do korekcji wag tylko temu neuronowi, który jest najbardziej wrażliwy na aktualny wektor wejściowy. W efekcie współzawodnictwa następuje samoorganizacja procesu uczenia. Neurony dopasowują swoje wagi w taki sposób, że dla grupy wektorów bliskich sobie (wg pewnej miary odległości – szczegóły dalej) zwycięża zawsze ten sam neuron. Zatem układy tego typu mogą być stosowane do klasyfikacji wektorów.



Po podaniu na wyjściu wektora uczącego $x=(x_1, x_2, x_3)$ mechanizm konkurencji wybierze tylko jeden z czterech pokazanych neuronów (zwycięzcę). Tylko jego wagi będą modyfikowane (w tym cyklu, gdyż dla następnego wektora zwycięzca może być inny).

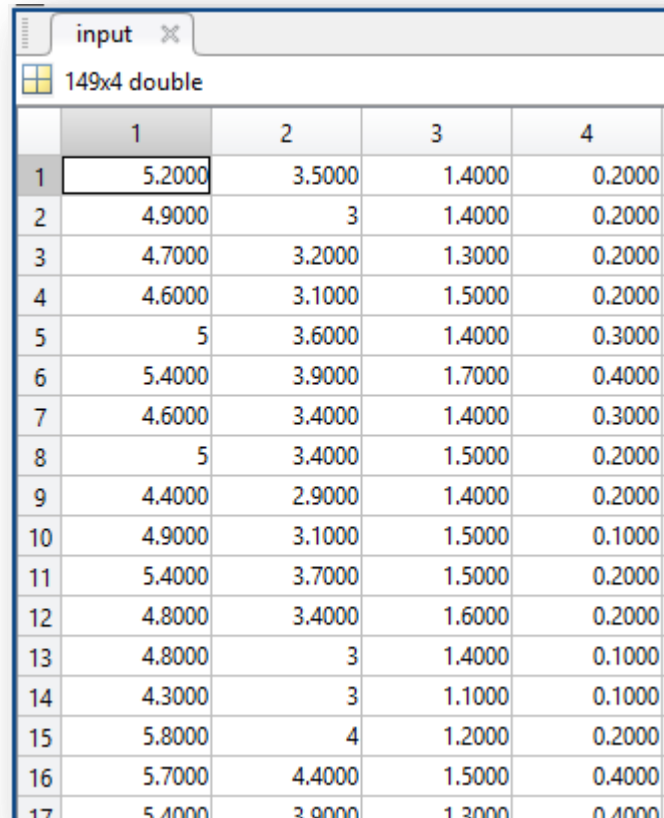
Dlaczego WTA nie wystarcza?

Taka reguła adaptacji powoduje, że algorytm słabo zbiega. Reguła, w której modyfikuje się nie tylko wagi zwycięzcy, ale również jego sąsiadów, jest zwana regułą WTM (Winner Takes Most). W praktyce reguła WTA została zastąpiona regułą WTM.

3. Zrealizowane kroki scenariusza

Przygotowałam dane uczące zawierające numeryczny opis cech kwiatów, skorzystałam z przykładowego zestawu umieszczone na stronie pod linkiem:

https://en.wikipedia.org/wiki/Iris_flower_data_set.



	1	2	3	4
1	5.2000	3.5000	1.4000	0.2000
2	4.9000	3	1.4000	0.2000
3	4.7000	3.2000	1.3000	0.2000
4	4.6000	3.1000	1.5000	0.2000
5	5	3.6000	1.4000	0.3000
6	5.4000	3.9000	1.7000	0.4000
7	4.6000	3.4000	1.4000	0.3000
8	5	3.4000	1.5000	0.2000
9	4.4000	2.9000	1.4000	0.2000
10	4.9000	3.1000	1.5000	0.1000
11	5.4000	3.7000	1.5000	0.2000
12	4.8000	3.4000	1.6000	0.2000
13	4.8000	3	1.4000	0.1000
14	4.3000	3	1.1000	0.1000
15	5.8000	4	1.2000	0.2000
16	5.7000	4.4000	1.5000	0.4000
17	5.4000	3.9000	1.3000	0.4000

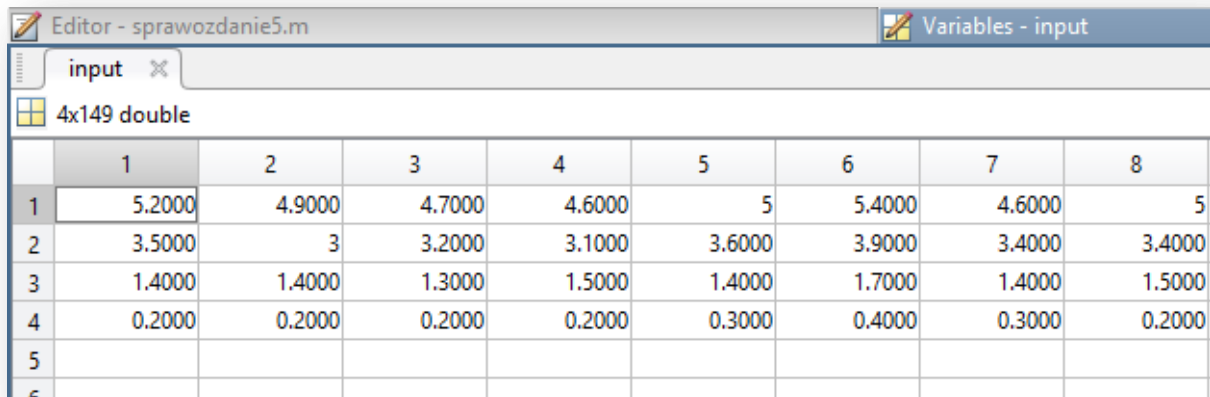
W pierwszej kolejności przesłałam do „input” dane uczące. Zbiór danych zawiera zestaw 50 rekordów pod 5 atrybutami - **długość płatka, szerokość płatka, długość sepii, szerokość sepa i klasa**. Pełna liczba danych uczących dołączona została do repozytorium.

W kolejnym kroku przygotowałam za pomocą gotowych narzędzi sieć Kohonena i algorytm uczenia oparty o regułę „Winner Takes All” (WTA).

Dokonałam transponowania macierzy oraz za pomocą funkcji „**nntool**” otworzyłam Network/Data Manager, które umożliwia importowanie, tworzenie, używanie i eksportowanie sieci neuronowych i danych:

```
Trial>> input=input';  
Trial>> nntool  
fx Trial>> |
```

Dzięki transponowaniu macierzy nasze dane wejściowe zostały przedstawione jak na zdjęciu poniżej:



	1	2	3	4	5	6	7	8
1	5.2000	4.9000	4.7000	4.6000	5	5.4000	4.6000	5
2	3.5000	3	3.2000	3.1000	3.6000	3.9000	3.4000	3.4000
3	1.4000	1.4000	1.3000	1.5000	1.4000	1.7000	1.4000	1.5000
4	0.2000	0.2000	0.2000	0.2000	0.3000	0.4000	0.3000	0.2000
5								
6								

W „Create Network or Data” dokonujemy wyboru takich parametrów jak:

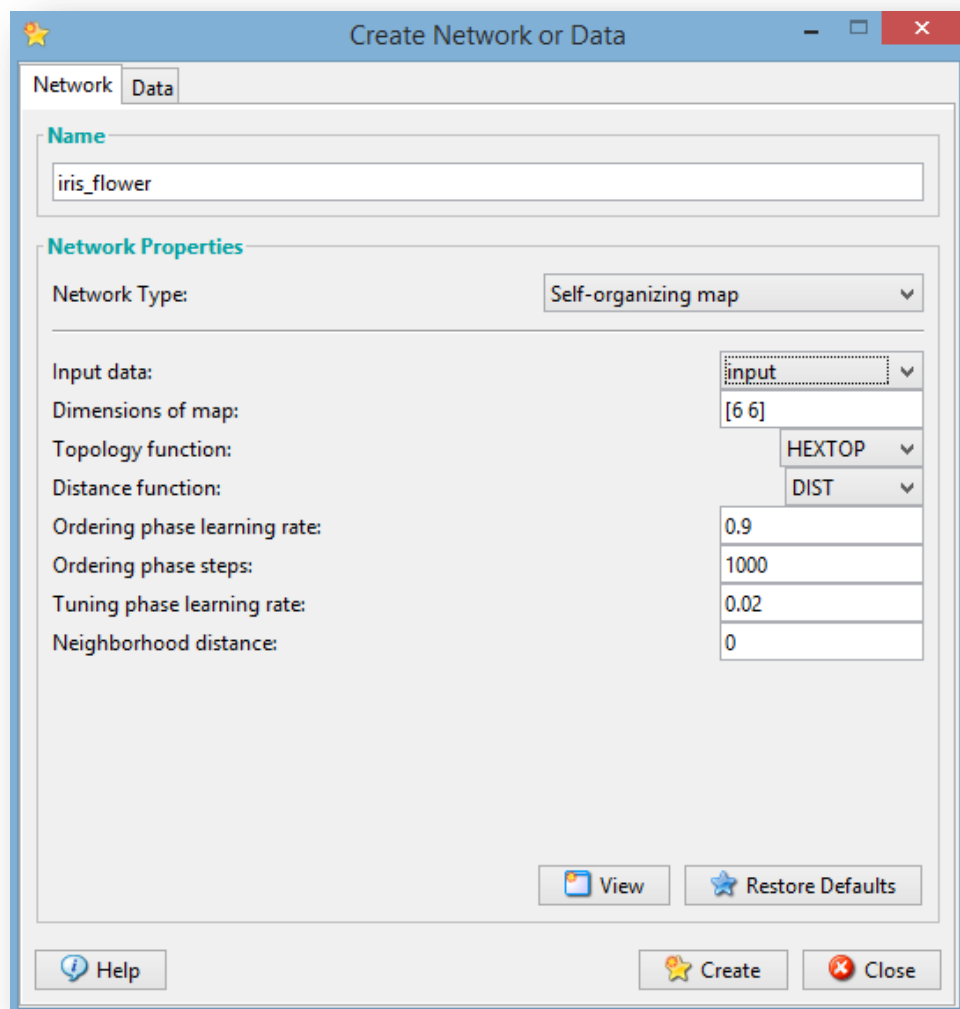
Typ sieci: Self-organizing map (SOM), czyli nasza sieć **Kohonena**

Rozkład na mapie : 6x6

Funkcja topologii warstw: HEXTOP -hextop oblicza położenia neuronów dla warstw, których neurony są ułożone w N sześciokątny wzór wymiarowy.

Funkcja odległości: DIST- funkcja wagi odległości euklidesowej

Rozmiar sąsiedztwa: 0 –ponieważ korzystamy z reguły WTA



```

11 % Create a Self-Organizing Map
12 dimension1 = 6;
13 dimension2 = 6;
14 net = selforgmap([dimension1 dimension2]);

```

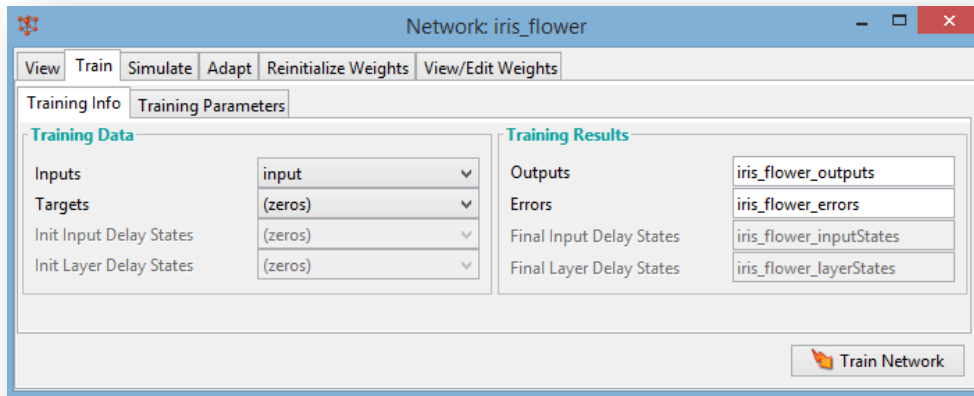
Dimension 1 jak i dimension2 oznacza, że rozkład mapy będzie wynosił 6x6.

Automatyczne tworzenie sieci jak uczyniłam to wyżej robi to za pomocą funkcji „selforgmap”, która jest siecią samoorganizującą się.

Skłania:

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

Uczenie sieci:



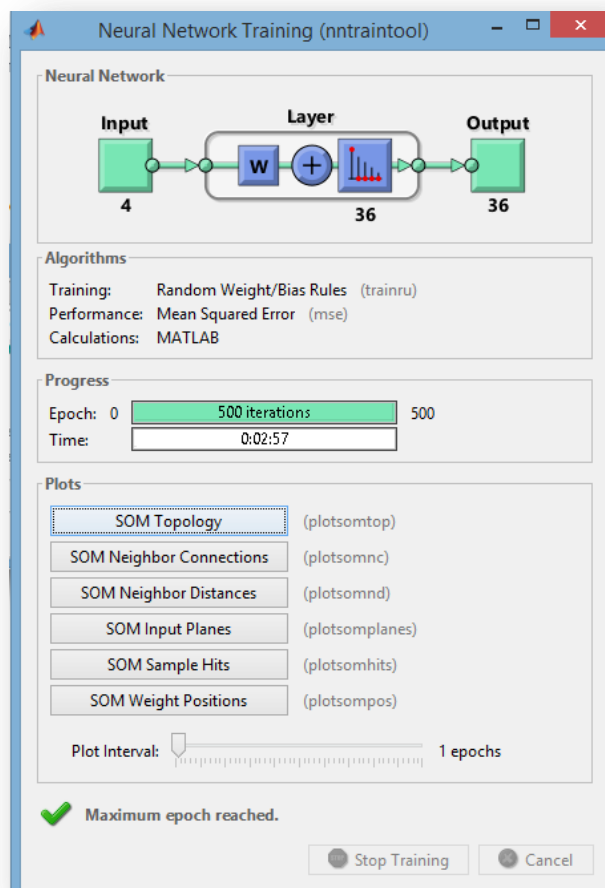
```

16 % Train the Network
17 [net,tr] = train(net,x);
18
19 % Test the Network
20 y = net(x);

```

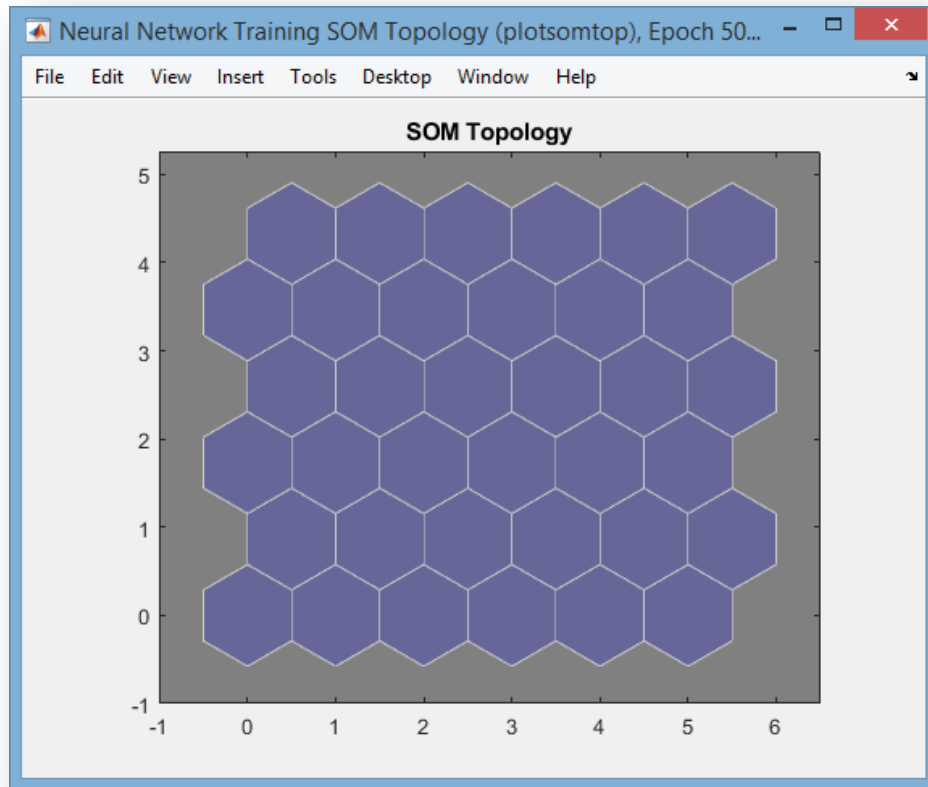
Wyniki działania sieci:

Szkolenie obejmuje maksymalną liczbę epok, która wynosi 500, czas wykonania zadania to 0:02:57.



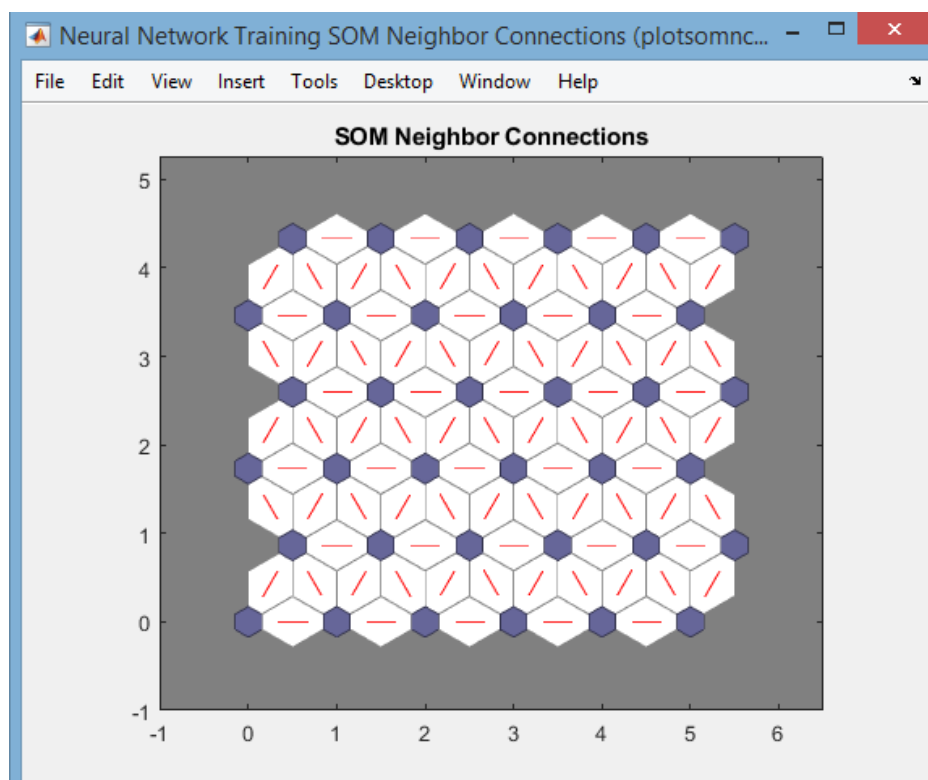
SOM Topology:

Na tej figurze każdy z sześciokątów reprezentuje neuron. Siatka to 6 na 6, więc w tej sieci jest łącznie 36 neuronów. W każdym wektorze wejściowym znajdują się cztery elementy, więc przestrzeń wejściowa jest czterowymiarowa. Wektory masy (centra skupień) mieszczą się w tej przestrzeni.



SOM Neighbor Connections:

Niżej umieszczona mapa przedstawia powiązanie sąsiedzkie.

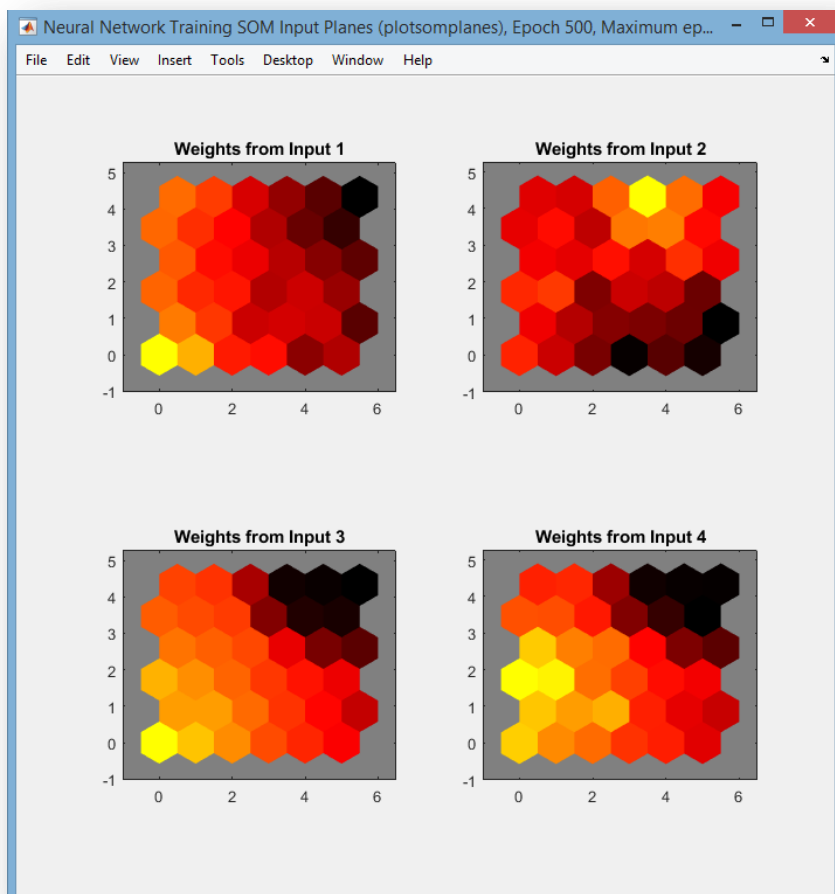


SOM Neighbor Distance:

Na tej figurze niebieskie sześciokąty reprezentują neurony. Czerwone linie łączą sąsiednie neurony. Kolory w obszarach zawierających czerwone linie wskazują odległości między neuronami. Ciemniejsze kolory reprezentują większe odległości, a jaśniejsze kolory oznaczają mniejsze odległości. Pasma ciemnych segmentów przechodzi od górnego środkowego regionu do środkowego prawego regionu.



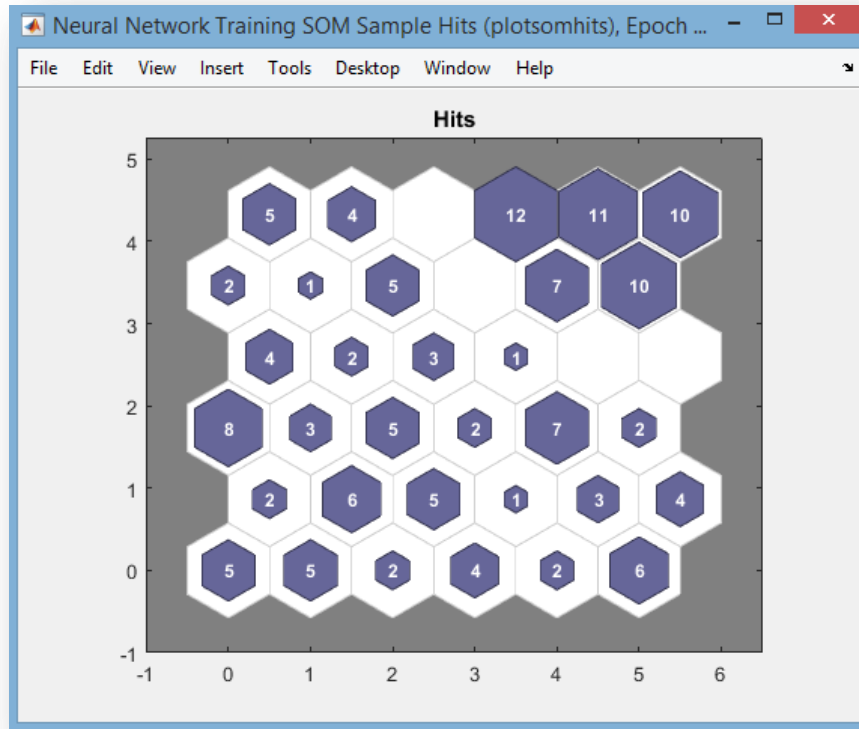
SOM Input Planes:



Ta figura przedstawia płaszczyznę ciężkości dla każdego elementu wektora wejściowego (w tym przypadku cztery). Są to wizualizacje wag, które łączą każde wejście z każdym z neuronów. **(Ciemniejsze kolory oznaczają większe wagi.)** Jeśli schematy połączeń dwóch wejść były bardzo podobne, można założyć, że dane wejściowe są wysoce skorelowane.

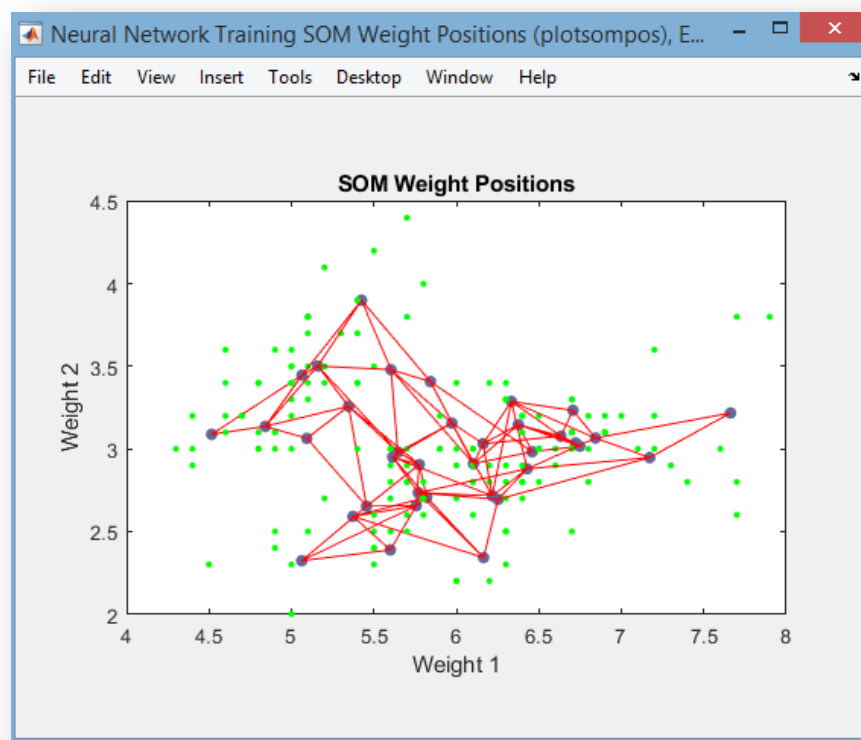
SOM Sample Hits:

W przypadku SOM training wektor wagowy związany z każdym neuronem przesuwa się, aby stać się centrum skupienia wektorów wejściowych. Ponadto, neurony, które sąsiadują ze sobą w topologii, powinny również poruszać się blisko siebie w przestrzeni wejściowej, dlatego możliwe jest zwizualizowanie wielowymiarowej przestrzeni wejściowej w dwóch wymiarach topologii sieci. Zdjęcie to przedstawia ile razy dany neuron zwyciężył podczas rywalizacji.



Domyślna topologia SOM jest sześciokątna. Ta figura pokazuje lokalizacje neuronów w topologii i wskazuje, ile danych treningowych jest powiązanych z każdym z neuronów (centrami klastra). Topologia to 6 na 6, więc istnieje 36 neuronów.

SOM Weight Positions:



Podsumowanie:

Celem sieci SOM jest klasyfikowanie wielowymiarowych danych wejściowych - obiektów opisanych dużą ilością parametrów – w taki sposób, by przedstawić reprezentację tych danych w mniejszej ilości wymiarów, przeważnie dwóch, przy możliwie jak najwierniejszym odwzorowaniu struktury wewnętrznej wektora wejściowego. Dzięki temu sieci te przydatne są przy wizualizacjach skomplikowanych struktur, a przetworzone przez nie dane mogą stanowić podstawę do diagramów wyświetlanych na ekranie, jak również znajdują zastosowanie wszędzie tam, gdzie istotna jest redukcja rozmiarów danych wejściowych, ze względu na zdolność kompresji samoorganizującej sieci Kohonena.

Należy zwrócić szczególną uwagę przy doborze ilości neuronów, ponieważ WTA powoduje konkurencję między neuronami. Jeśli chodzi o topologię, w tym przypadku używałam złożoną z sześciątów.