Podstawy sztucznej inteligencji

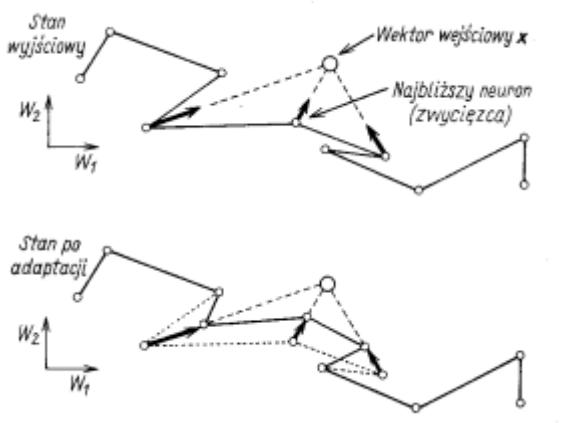
Kinga Synowiec, Inżynieria obliczeniowa, grupa 2

Sprawozdanie nr 6

- **Temat ćwiczenia:**
  Budowa i działanie sieci Kohonena dla WTM.

- **Cel ćwiczenia:**
  Poznanie budowy i działania sieci Kohonena przy wykorzystaniu reguły WTM do odwzorowywania istotnych cech liter alfabetu.

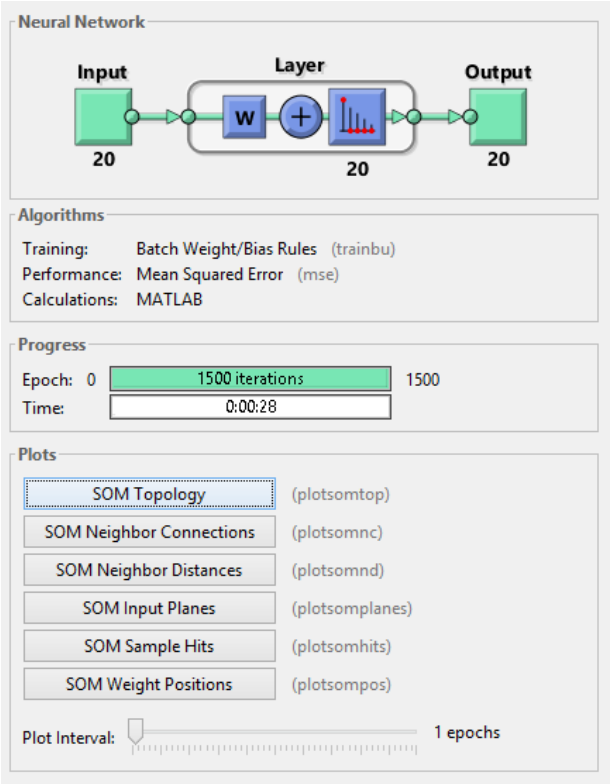
- **Wstęp teoretyczny:**
  - WTM (Winner Takes Most) – rodzaj algorytmu, w którym oprócz zwycięzcy swoje wagi uaktualniają także neurony z jego sąsiedztwa. Przykładowe zobrazowanie adaptacji wag wg WTM przedstawia poniższa ilustracja:



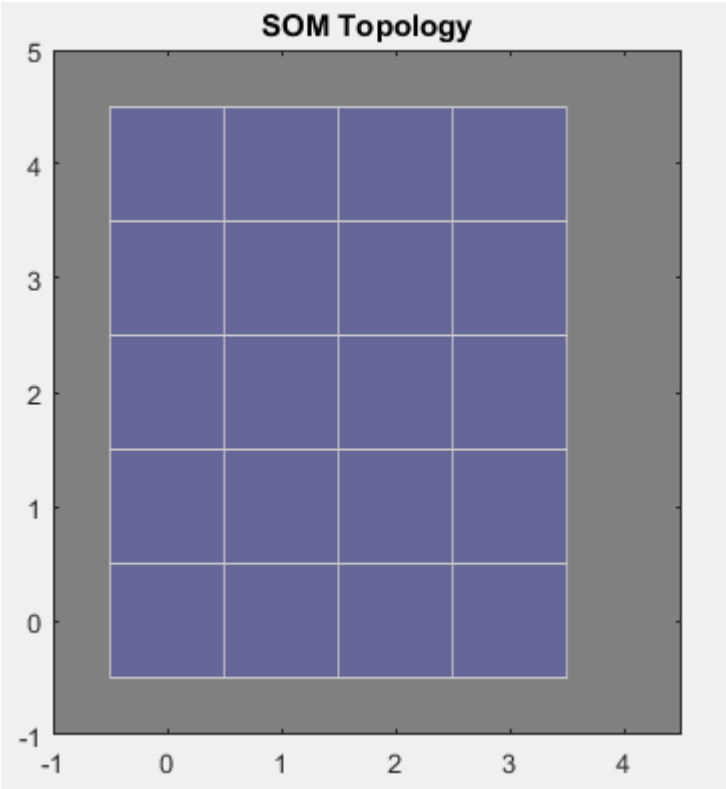
  - Sieć Kohonena - sieć neuronowa uczona w trybie bez nauczyciela w celu wytworzenia niskowymiarowej (przeważnie dwuwymiarowej) zdyskretyzowanej reprezentacji przestrzeni wejściowej. Sieć Kohonena wyróżnia się tym od innych sieci, że zachowuje odwzorowanie sąsiedztwa przestrzeni wejściowej. Wynikiem działania 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

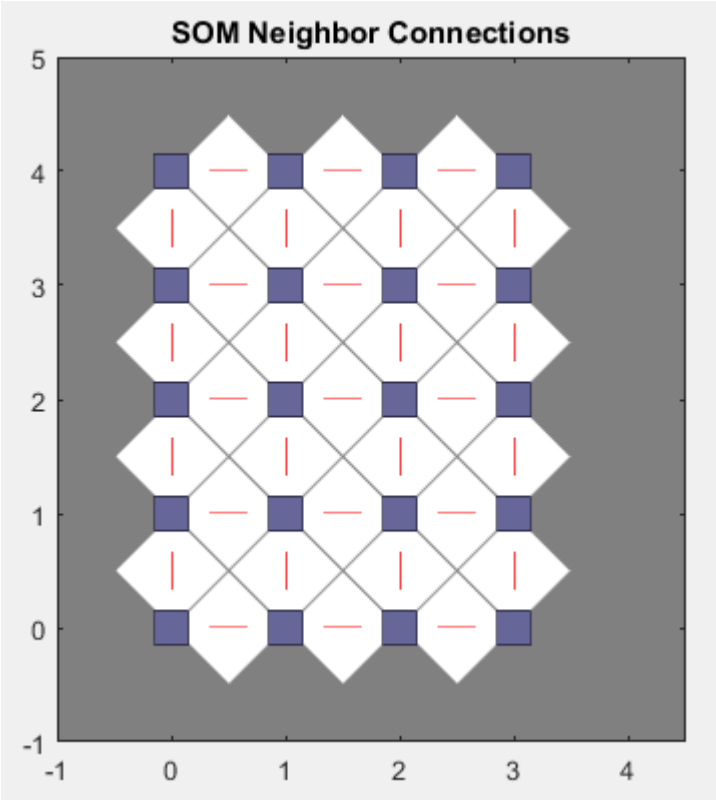
- **Zrzuty ekranu przedstawiające rezultaty po uruchomieniu programu wraz z opisem przedstawionych wykresów:**



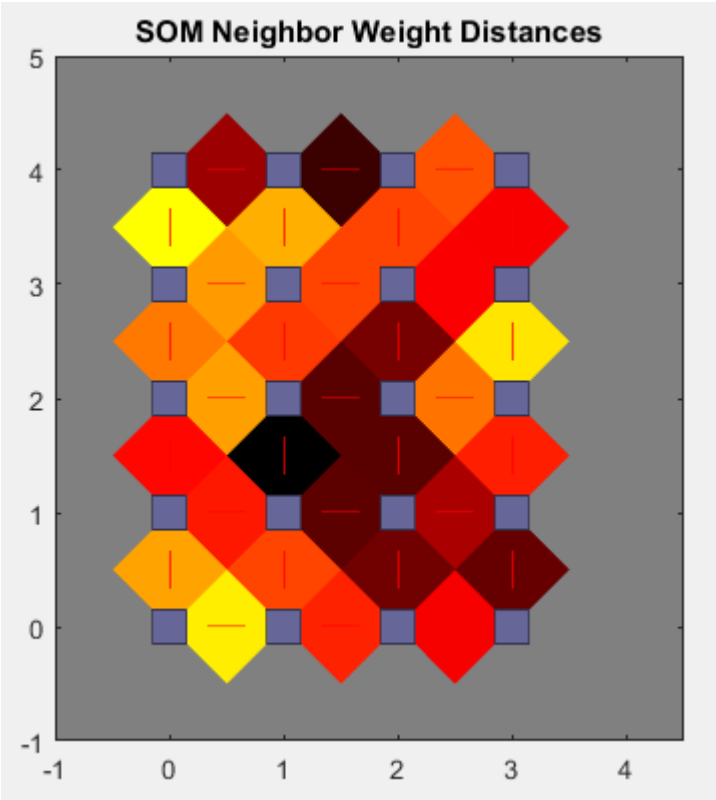
- SOM Topology – topologia sieci Kohonena dla wcześniej określonej maksymalnej liczby epok treningowych. W tym przypadku jest to prostokątna siatka neuronów 4x5.



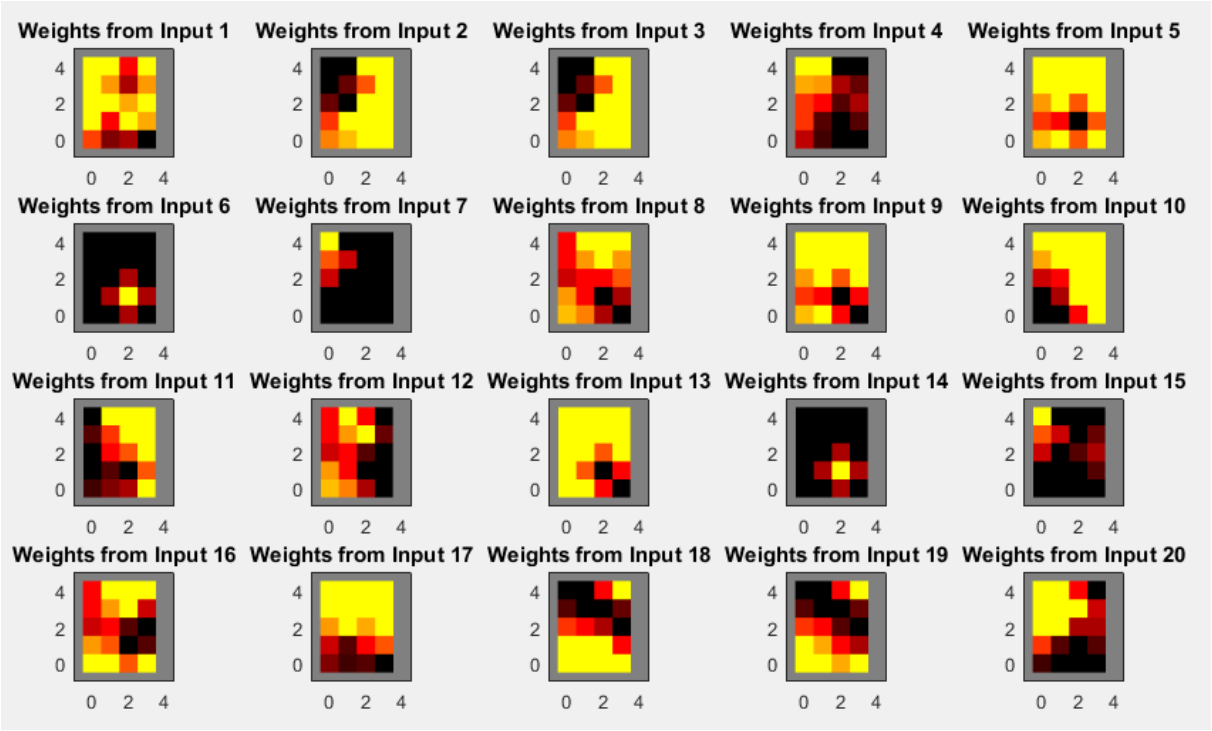
- SOM Neighbor Connections – ilustracja przedstawiająca połączenia pomiędzy poszczególnymi neuronami w utworzonej sieci.



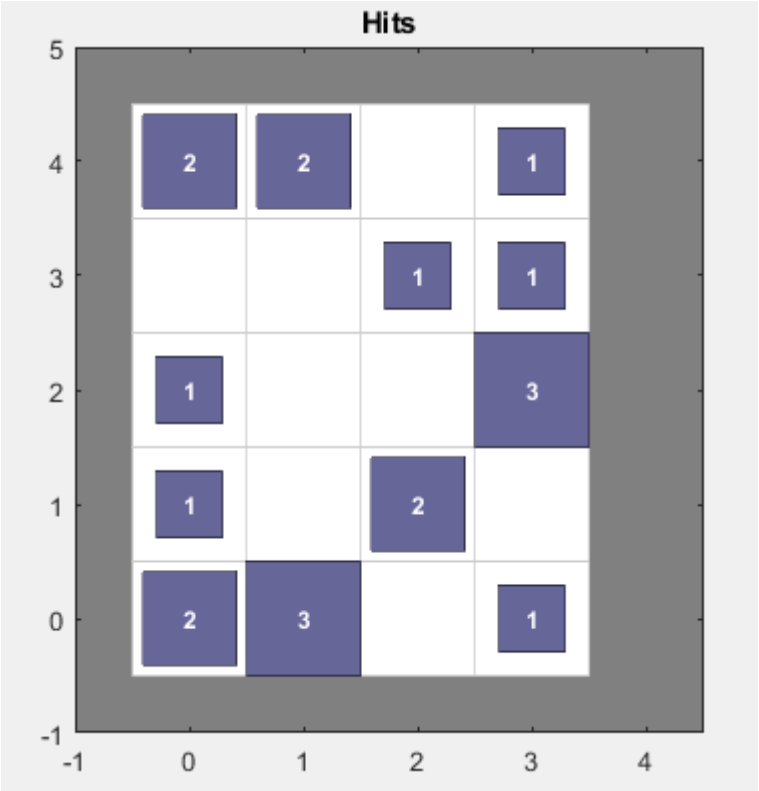
- SOM Neighbor Distance – ilustracja obrazująca odległości pomiędzy wagami poszczególnych neuronów (im ciemniejszy kolor, tym większa wartość wag).



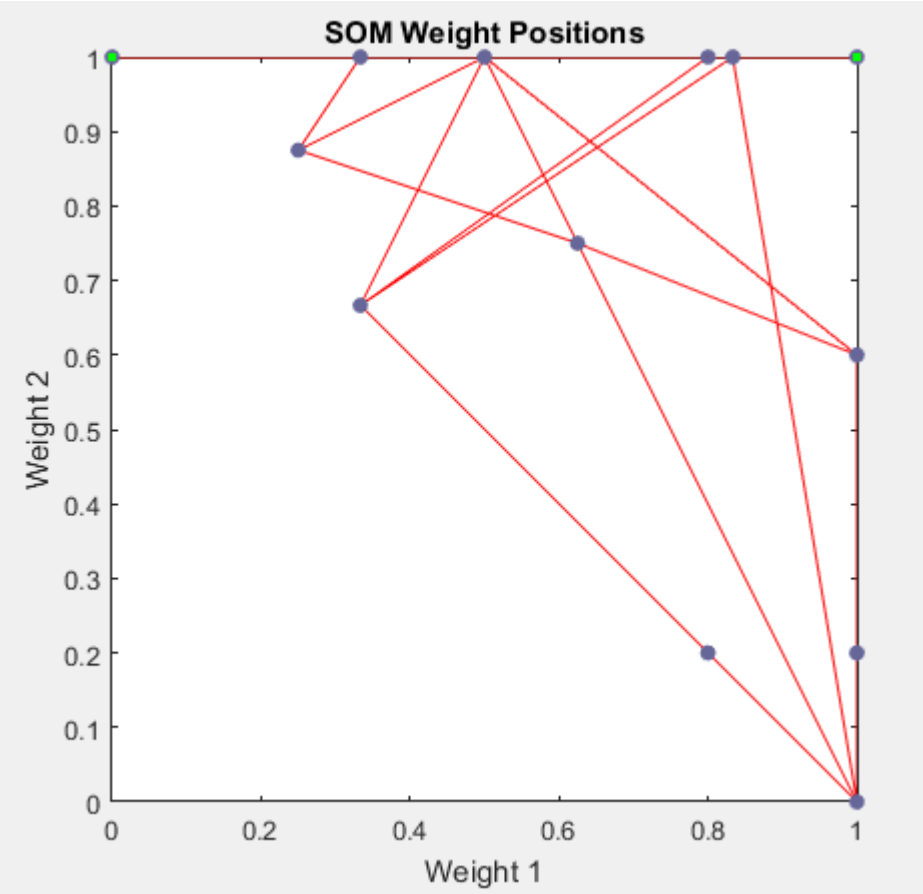
- SOM Input Planes – kolejne wejścia, im ciemniejszy kolor, tym wyższa waga.



- SOM Sample Hits – rozkład sił neuronów, a dokładniej ilość zwycięstw neuronów w konkurencji do WTM.



- SOM Weight Positions – efekt końcowy nauki, gdzie niebieskie kropki to neurony, a linie to połączenia między nimi.



- **Wnioski:**

  - Prostokątna siatka neuronowa pozwala na stworzenie bezpośrednich powiązań między neuronami w najbliższym sąsiedztwie.
  - Dzięki zastosowaniu algorytmu WTM rozłożenie zwycięstw na całej sieci było prawie równomierne.
  - Zwiększenie rozmiarów sąsiedztwa powoduje nieprawidłowości w działaniu sieci neuronowej – program nie generuje błędów jedynie, gdy rozmiar sąsiedztwa zmienia się wraz ze wzrostem sieci.
  - Ilość neuronów w sieci wpływa na rozłożenie wag poszczególnych neuronów. Im więcej neuronów w sieci, tym dłuższy był czas obliczeń.
  - Analizując rozkład neuronów zwycięskich można stwierdzić, że sieć korzystała z algorytmu WTM, gdyż rozkład ten jest zdecydowanie bardziej równomierny niż w przypadku WTA z poprzedniego zadania.
  - Ilość epok w programie została ustalona w taki sposób, aby w miarę możliwości nie przedłużać zbytnio czasu nauczania, ale aby wyniki były klarowne i dokładne.
  - Algorytm WTM sprawdza się lepiej w przypadku sieci z dużą ilością neuronów niż WTA.

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

```
close all; clear all; clc;
%kolumnowa reprezentacja binarna pierwszych 20 dużych liter alfabetu dla
tablicy 4x5
%dane wejściowe – dane uczące, wartości implementowane przez oprogramowanie
%A B C D E F G H I J K L M N O P R S T U, litery zapisane binarnie i
kolumnowo
WEJSCIE =
[0 1 0 1 1 1 0 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]; %wektor rzędów wymiarów
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; dist – funkcja wagi
odległości euklidesowej
% TWORZENIE SIECI KOHONENA (SOM)
net = selforgmap(dimensions, coverSteps, initNeighbor, topologyFcn,
distanceFcn); %zmienna, do której przypisujemy nowo tworzoną sieć neuronową
z wykorzystaniem w.w. parametrów
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
```