

Podstawy sztucznej inteligencji Sprawozdanie do scenariusza 5

Zadanie:

Przygotowanie sieci Kohonena przy wykorzystaniu reguły WTA, która będzie odwzorowywać istotne części kwiatów.

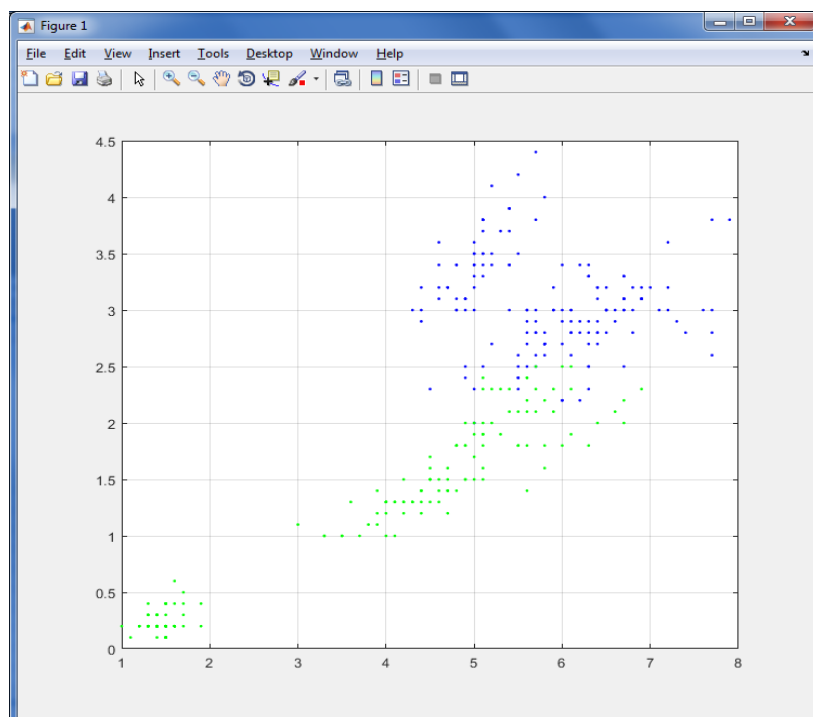
Podczas realizacji scenariusza będę korzystał z oprogramowania MATLAB 2016a oraz biblioteki Neural Network Toolbox.

Dane uczące:

Danymi uczącymi jest zestaw zaimplementowany w oprogramowaniu MATLAB o nazwie iris_dataset. Zawiera on numeryczny opis 4 cech kwiatów irysa (długość i szerokość płatków oraz długość i szerokość działki kielicha) na przykładzie 150 kwiatów zapisany w tablicy o wymiarach 4x150.

| 75 | 76 | 77 | 78 | 79 | 80 | 81 | 82 | 83 |
|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| 6.4000 | 6.6000 | 6.8000 | 6.7000 | 6 | 5.7000 | 5.5000 | 5.5000 | 5.8000 |
| 2.9000 | 3 | 2.8000 | 3 | 2.9000 | 2.6000 | 2.4000 | 2.4000 | 2.7000 |
| 4.3000 | 4.4000 | 4.8000 | 5 | 4.5000 | 3.5000 | 3.8000 | 3.7000 | 3.9000 |
| 1.3000 | 1.4000 | 1.4000 | 1.7000 | 1.5000 | 1 | 1.1000 | 1 | 1.2000 |

Przykładowy fragment danych uczących, każda kolumna opisuje kolejno: długość i szerokość płatków oraz długość i szerokość kielicha dla konkretnego kwiatu.



Wykres przedstawia długość (oś X) i szerokość (oś Y) płatków (niebieskie kropki) oraz długość (oś X) i szerokość (oś Y) kielicha (zielone kropki) wszystkich 150 kwiatów znajdujących się w danych uczących.

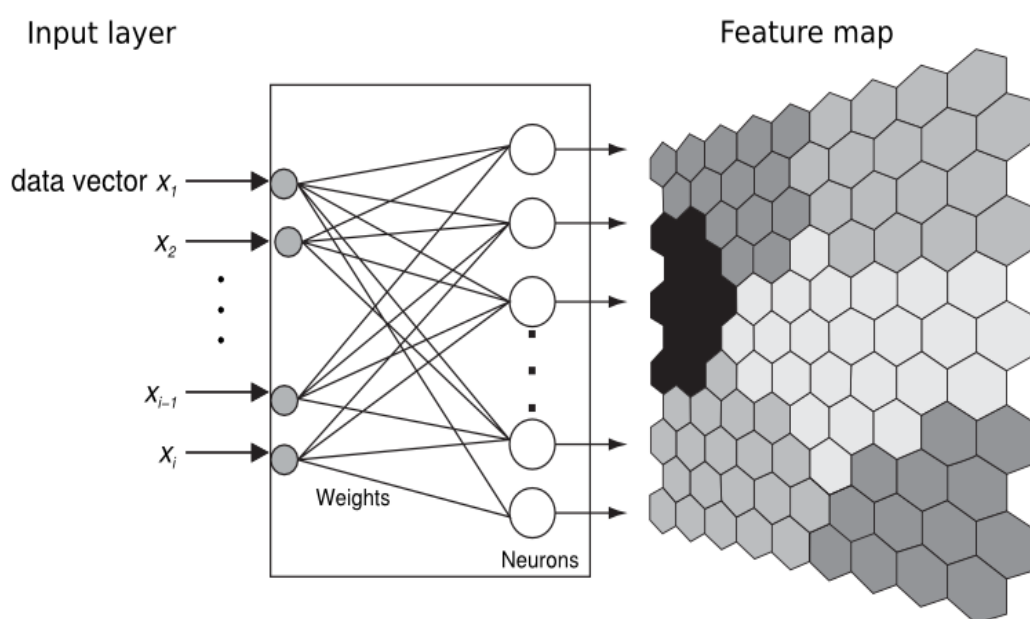
Sieć Kohonena:

Sieć Kohonena jest szczególnym przypadkiem algorytmu realizującego uczenie bez nauczyciela. Jej głównym zadaniem jest organizacja wielowymiarowej informacji (np. obiektów opisanych 20 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, Competitive Filters.

Funkcjonowanie sieci Kohonena odbywa się w trzech etapach:

- konstrukcja
- uczenie
- rozpoznawanie

Sieci Kohonena opierają się na uczeniu metodą samoorganizacyjną typu konkurencyjnego – oznacza to że neurony współzawodniczą między sobą o to, by zwiększyć wartość swojej wagi. Istnieje wiele różnych metod współzawodnictwa, które w zróżnicowany sposób wybierają zwycięskiego neuron. Ponieważ sieć Kohonena opiera się na uczeniu bez nauczyciela dane wyjściowe są opracowywane samodzielnie przez sieć na podstawie obserwacji danych wejściowych i wyławianiu zależności pomiędzy nimi. W sieciach SOM każdy neuron warstwy wejściowej komunikuje się z neuronami warstwy topologicznej ale neurony z tej samej warstwy nie komunikują się między sobą.



Zasada działania sieci Kohonena – dane wejściowe trafiają do neuronów a następnie są odwzorowywane na warstwę topologiczną, co daje siatkę neuronów z efektem

działania sieci. Neurony mogą być ułożone w siatkę hexagonalną lub prostokątną.

Reguła Winner Takes All:

Jest to reguła aktywacji neuronów w sieci polegająca na tym że w danej chwili tylko jeden neuron może znajdować się w stanie aktywnym. W wyniku tego naraz zostaje zmodyfikowana waga tylko jednego neuronu. Aby uniknąć tego, że jeden neuron będzie zawsze wygrywał często stosuje się także mechanizm zmęczenia który polega na tym, że jeśli jakiś neuron wygrywa zbyt często to na pewien czas przestaje być brany pod uwagę w rywalizacji.

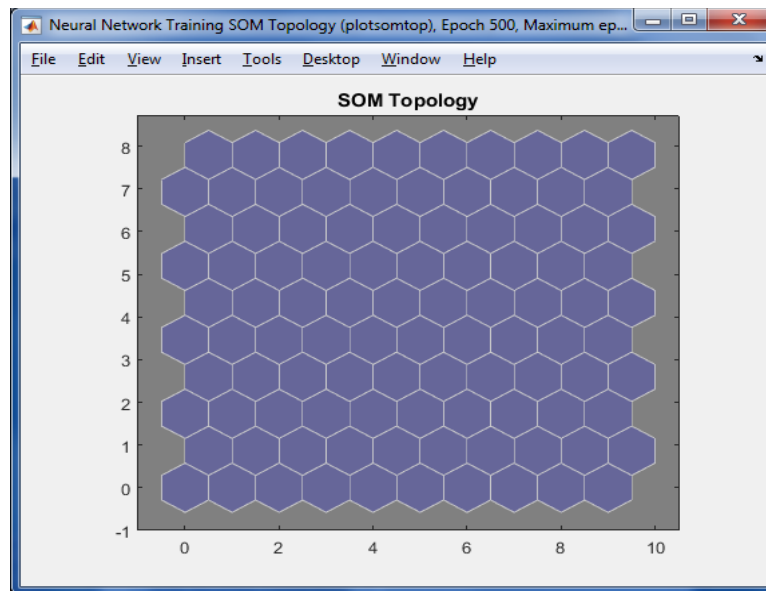
Do uczenia sieci używamy następującego algorytmu:

- generujemy losowo znormalizowane wektory wag
- losujemy wektor x oraz liczymy dla niego aktywację y dla wszystkich neuronów
- szukamy neuronu zwycięzcy
- modyfikujemy wektor wag zwycięzcy a następnie go normalizujemy (sprawdzamy czy nie wygrywa zbyt często, jeśli tak to jest na chwilę usypiany)
- zatrzymujemy algorytm po odpowiednio dużej ilości iteracji

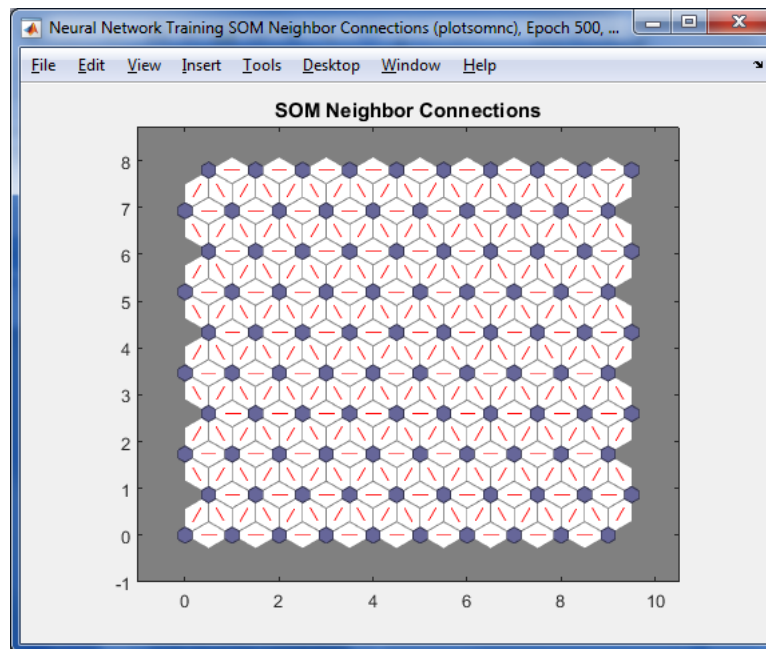
Stosowanie WTA prowadzi to podzielenia mapy neuronów na tzw. „strefy wpływów”
- są to obszary które znajdują się pod dominacją konkretnego silnego neuronu, który uniemożliwia modyfikację wag neuronów z jego otoczenia. Brak normalizacji może doprowadzić do sytuacji w której na niewielkim obszarze znajduje się kilka silnych neuronów lub kilka niewielkich stref wpływów natomiast pozostały obszar nie posiada żadnego silnego neuronu.

Wyniki działania programu:

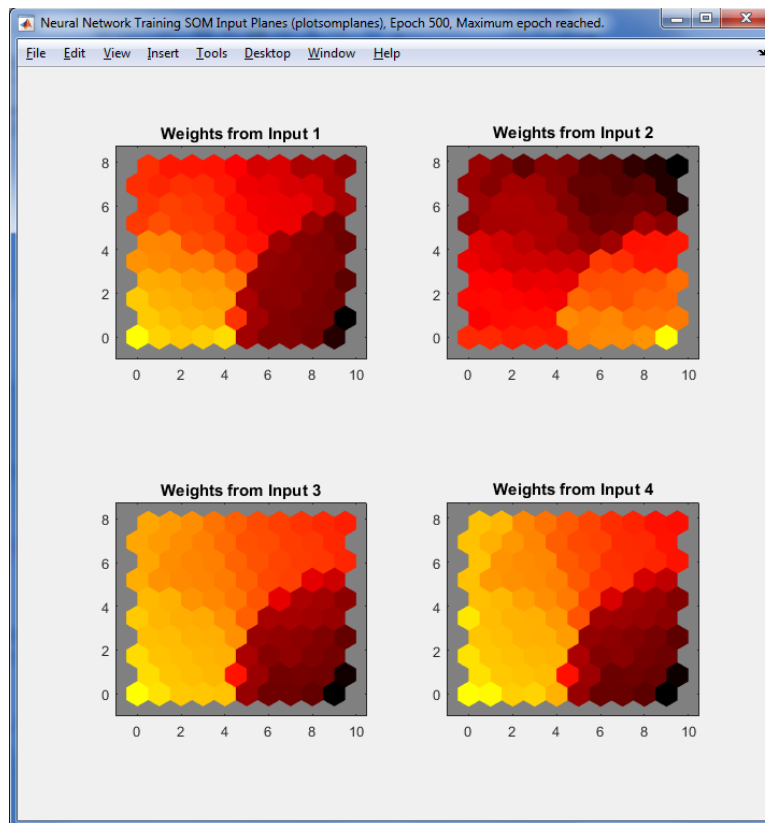
W programie wykorzystałem heksagonalną siatkę neuronów, uczenie wg reguły Kohonena i WTA. Program w efekcie powinien odwzorować istotne cechy kwiatów irysa na podstawie otrzymanych danych.



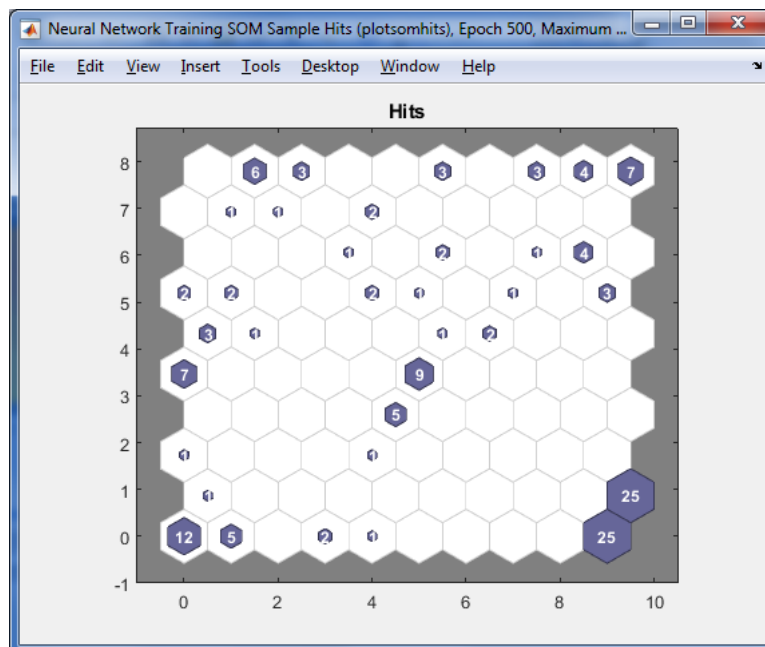
Topologia sieci



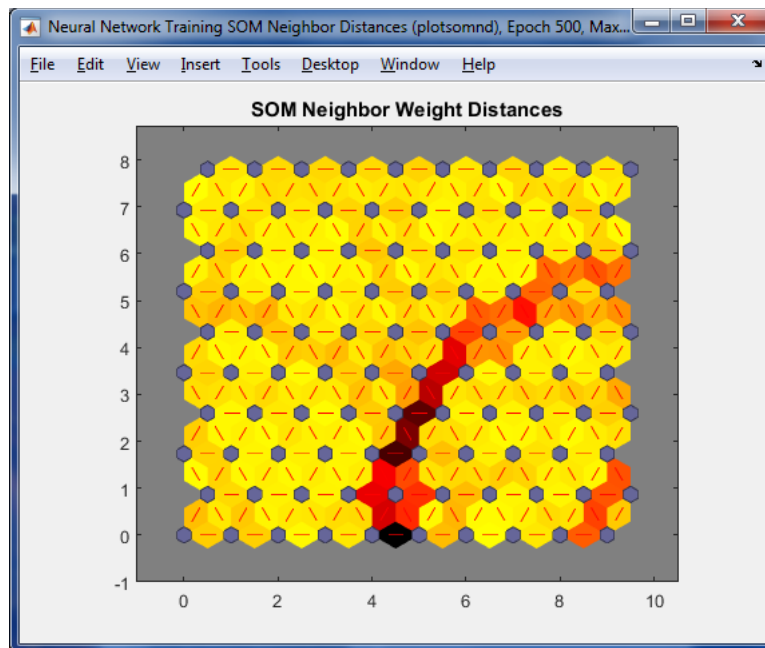
Połączenia pomiędzy poszczególnymi neuronami



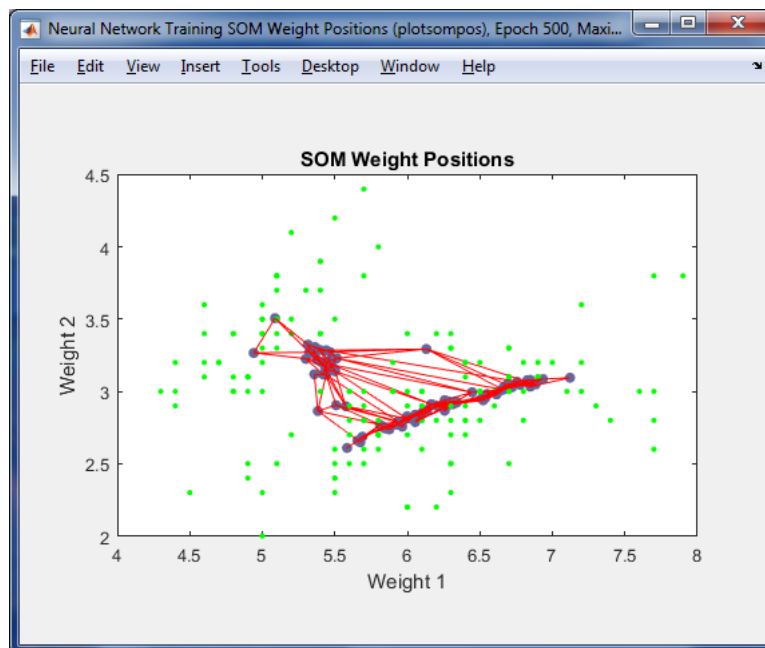
Rozkład wag dla każdego z wejść, ciemniejszy kolor oznacza wyższe wagi. Wejścia to kolejno: długość płatka, szerokość płatka, długość kielicha i szerokość kielicha.



Grafika obrazuje ile razy poszczególne neurony zwyciężyły podczas rywalizacji w ramach Winner Takes All



Odległości pomiędzy wagami



Efekt końcowy: zielone kropki przedstawiają poszczególne kwiaty, niebieskie to te kwiaty, które zawierają najbardziej typowe cechy irysa.

Analiza oraz wnioski:

Analizując ostatni obraz można stwierdzić że sieć w poprawny sposób wyznaczyła typowe cechy irysa – sieć wytypowała jako te najbardziej typowe kwiaty, które znajdowały się na środku wykresu co oznacza, że miały przeważnie parametry oscylujące w okolicach ich średnich wartości co sugeruje, że były bardzo zbliżone do „statystycznego” kwiatu irysa. Możliwe że wynika to z tego że zastosowałem heksagonalną siatkę neuronów dzięki czemu sieć miała znacznie więcej połączeń niż w przypadku siatki prostokątnej w wyniku czego miała większe możliwości doboru odpowiednich wag.

Na trzecim obrazie można zauważyć zjawisko tworzenia się stref wpływów. Analiza rozkładu barw na wykresach pozwala stwierdzić jak wygląda typowy kwiat irysa według sieci neuronowej. Ciemne kolory oznaczają istotniejsze wagi tak więc dana cecha jest tym bardziej typowa dla irysa im ciemniejszym kolorem jest oznaczona. Na podstawie danych informacji wydobytych przez sieć można się domyślać że kwiat irysa posiada długie i szerokie listki a także dość długi i szeroki kielich. W przybliżeniu te informacje pokrywają się z typowym wyglądem kwiatu irysa. Czwarty obraz pokazuje, jak wiele razy poszczególne neurony zostały zwycięzcami w trakcie rywalizacji. Jak wyraźnie widać sieć zadziałała bezbłędnie, nie modyfikowała wag wyłącznie jednego neuronu. Zapewne wynika to z zastosowania normalizacji, która wyłączała z rywalizacji te neurony, które wygrywały zbyt często. Gdyby jej nie było to prawdopodobnie wszystkie zwycięstwa zgarnęłyby neurony znajdujące się w prawym dolnym rogu ponieważ widać jak dużą przewagę mają nad pozostałymi neuronami. Prawdopodobnie zostały one wykluczone już na samym początku działania sieci i dopiero wtedy rywalizację zaczęły wygrywać pozostałe neurony.

Nakładając na siebie czwarty i piąty obraz można zauważyć, że odległości wag neuronów są powiązane z tym, jak wiele razy neuron wygrał rywalizację – szczególnie widoczne jest to na pasie, który przebiega od środka dolnej części wykresu do środka jego prawej strony: pas ten idealnie pokrywa się z występowaniem neuronów zwyciężających.

Ilość epok treningu którą wybrałem (500) jest wystarczająca dla poprawnego działania sieci – zwiększenie jej nie poprawiało w znacznym stopniu efektywności programu a przy wyższych wartościach znacząco wydłużało czas treningu.

Podsumowując, sieć Kohoneta wraz z regułą Winner Takes All bardzo dobrze poradziła sobie z zadaniem które otrzymała: odwzorowała typowe cechy dla kwiatów irysa bardzo efektywnie przy niewielkiej ilości epok jak na uczenie bez nauczyciela. Bardzo zaskakujące było to jak skutecznie sieć wykryła podobieństwa pomiędzy kwiatami i na ich podstawie była w stanie sklasyfikować je jako istotne cechy irysów. Dla poprawnego działania bardzo ważna jest ilość neuronów – zbyt mała spowoduje niedokładne wyniki natomiast zbyt duża będzie skutkowała bardzo długim czasem nauki. Sieć prawdopodobnie nie działała by w tak dobry sposób gdyby nie normalizacja – pozwoliła ona na wyłączanie z rywalizacji neuronów wygrywających zbyt często przez co umożliwiła rywalizację innym neuronom, które również mogły mieć bardzo istotne znaczenie w działaniu sieci. Bez normalizacji prawdopodobnie tylko kilka neuronów wygrałoby we wszystkich rywalizacjach powodując to, że cała uwaga sieci zostałaby skupiona na bardzo małej ilości neuronów na której oparte byłoby całe jej działanie co z pewnością nie skutkowałoby poprawnymi wynikami gdyż wtedy sieć mogłaby działać tak, jak sieć złożona z małej ilości neuronów, ponieważ pozostałe i tak byłyby ignorowane ze względu na zbyt małe wagi. Prowadzi to wszystko do wniosku że konkurencja pomimo swojej skuteczności niesie ryzyko niedokładności, która będzie wynikała z przesadnego skupienia się na pojedynczym, wykrywającym neuronie.

Listing programu:

```
close all; clear all; clc;

WE = iris_dataset;
size(WE);
plot(WE(1,:),WE(2,:), 'b.',WE(3,:),WE(4,:), 'g. ');
hold on; grid on;

%parametry sieci
dimensions = [10 10];
coverSteps = 100;
initNeighbor = 0;
topologyFcn = 'hextop';
distanceFcn = 'dist';

% tworzenie SOM
net =
selforgmap(dimensions,coverSteps,initNeighbor,topologyFcn,distanceFcn);
net.trainParam.epochs = 500;

% trenowanie sieci
[net,tr] = train(net,WE);
y = net(WE);

grid on
```