

## Ćwiczenie Koh

# Sieć Kohonena

### Część teoretyczna

Wykład 7: Sztuczne sieci neuronowe – sieć Kohonena.

### Zadania pomocnicze

- Zapoznaj się z programami wymienionymi poniżej (uruchomienie programów następuje poprzez wpisanie ich nazw w oknie Matlab).
  - program **democ1** obrazuje uczenie konkurencyjne
  - program **demom1** demonstruje uczenie jednowymiarowej samoorganizującej się tablicy neuronów
  - program **demom2** demonstruje uczenie dwuwymiarowej samoorganizującej się tablicy neuronów
- Zapoznaj się z funkcjami `newsom1` i `train` (help lub dokumentacja Matlab).

### Zadania do wykonania

Dane są punkty  $(x_1, x_2)$  rozmieszczone w kilku grupach. Za pomocą samoorganizującej się sieci Kohonena odwzoruj te punkty na tablice neuronów.

- Generowanie zbioru danych.  
Tworzymy funkcję generującą punkty uczące i zapisujemy ją pod nazwą `gener_grup.m`:

```
function v = gener_grup(nr_gr,x,c,n,d,rk)

if nargin < 3, error('Not enough arguments. '), end
if nargin == 3, d = 1; end

[r,q] = size(x);
minv = min(x')';
maxv = max(x')';
rand('state',nr_gr*rk);
v = rand(r,c) .* ((maxv-minv) * ones(1,c)) + (minv * ones(1,c));
t = c*n;
randn('state',nr_gr*rk*2);
v = nncopy(v,1,n) + randn(r,t)*d;
```

- Tworzymy i uczymy sieć neuronową Kohonena:

Uwaga - polecenia zamieszczone poniżej należy umieścić w jednym skrypcie.

```
nr_gr =
rk =
r = 0; %końcowy promień sąsiedztwa
t_neur = [1, 9]; %rozmiary tablicy neuronów
l_epok = 1; %liczba epok uczenia
top='hextop'; %topologia neuronów w tablicy (hextop, gridtop, randtop) -
zobacz w dokumentacji Matlab

%zbiór uczący
z = [-1 1; -1 1]; % zakres danych wejściowych
l_gr= 9; % liczba grup danych
l_p = 10; % liczba punktów w grupie
odch = 0.1; % odchylenie standardowe punktów w grupie
x = gener_grup(nr_gr,z,l_gr,l_p,odch,rk); %funkcja generująca wektory uczące

%projekt sieci
lrl = 0.9; %początkowa wartość współczynnika uczenia
```

```

lr2 = 0.1; %końcowa wartość współczynnika uczenia
l_po = l_epok/2; %liczba epok fazy porządkowania
net = newsum(minmax(x), t_neur, top, 'linkdist', lr1, l_po, lr2, r); % utworzenie
sieci Kohonena
plotsom(net.IW{1}, net.layers{1}.distances); %narysowanie wektorów wagowych
title('Początkowe wartości wag i wektory wejściowe');
hold on;
plot(x(1,:), x(2,:), '+k');
xlabel('x1, w1');
ylabel('x2, w2');
hold off;
'wciśnij ENTER'
pause

net.trainParam.epochs = l_epok;
net.trainParam.show = 5;

net = train(net, x); %trening sieci

plot(x(1,:), x(2,:), '+k');
hold on;
plotsom(net.iw{1,1}, net.layers{1}.distances); %narysowanie wektorów wagowych
title('Końcowe wartości wag i wektory wejściowe');
hold on;
plot(x(1,:), x(2,:), '+k');
xlabel('x_1, w_1');
ylabel('x_2, w_2');
hold off;

```

Do `nr_gr` przypisz numer swojej sekcji, a do `rk` aktualny rok kalendarzowy.

3. Zaobserwuj jak lokują się neurony (wektory wagowe neuronów) dla różnej liczby epok uczenia (1, 20, 40, 60, 80 i 100) i średniego stopnia skupienia grup: `odch = 0.1`. Eksperymenty wykonaj dla dziewięciu grup (`l_gr = 9`), dziesięciu punktów w każdej grupie (`l_p = 10`), końcowego promienia sąsiedztwa `r = 0` i tablicy neuronów  $3 \times 3$  (`t_neur = [3, 3]`) oraz  $1 \times 9$  (`t_neur = [1, 9]`). Pokaż i skomentuj wykresy dla każdego wariantu.
4. Wykonaj eksperymenty dla trzech stopni skupienia grup `odch = 0.01, 0.1, 0.5` i najlepszego twoim zdaniem wariantu z p. 1. Pokaż i skomentuj wykresy dla każdego wariantu.
5. Jakie znaczenie ma promień sąsiedztwa? Wykonaj eksperymenty dla `r = 0, 1, 2, 3 i 4`, `l_epok = 100`, `t_neur=[5, 5]`, `odch = 0.1` i liczby punktów w grupie `l_p = 20`. Pokaż i skomentuj wykresy dla każdego wariantu.
6. Czym różnią się topologie neuronów? Wykonaj eksperymenty z topologią `gridtop` i `hextop` oraz różnymi rozmiarami dwuwymiarowych tablic neuronów `t_neur` (przynajmniej 5 różnych rozmiarów).
7. Program zamieszczony poniżej demonstruje jak zmieniają się wagi sieci w kolejnych epokach uczenia. Sieć posiada `l_neur = 6` neuronów rozmieszczonych liniowo. Punkty uczące generowane są losowo z zakresu od 0 do 1 (zbiór punktów A) lub zagęszczają się w pobliżu zera (zbiór punktów B). Punkty uczące oznaczono kółkami na wykresach utworzonych w programie. Przeprowadź kilka eksperymentów dla obu zbiorów punktów A i B oraz dla różnej długości fazy porządkowania (`l_po`) w stosunku do liczby epok uczenia (`l_epok = 100`). Zamieść w sprawozdaniu i zinterpretuj wykresy zmian wag w trakcie treningu. Co można powiedzieć o samoorganizacji sieci?

```

r = 0; %końcowy promień sąsiedztwa
l_neur = 6; %liczba neuronów w warstwie liniowej
l_epok = 100; %liczba epok uczenia
l_po = 50; %liczba epok fazy porządkowania

%zbiór uczący
l_p = 100; % liczba punktów uczących
x = rand(1, l_p); %punkty uczące rozłożone losowo - zbiór A

```

```

%x=1./(1:l_p); %punkty uczące rozłożone gęściej w pobliżu zera - zbiór B

%projekt sieci
lp.order_lr = 0.9; %początkowa wartość współczynnika uczenia
lp.order_steps = l_po*l_p;
lp.tune_lr = 0.02; %końcowa wartość współczynnika uczenia
lp.tune_nd = r;
w = rand(l_neur,1); %wagi startowe
ls = [];
pos = hextop(l,l_neur); %topologia tablicy neuronów
d = linkdist(pos); %odległości pomiędzy neuronami

ww=zeros(l_neur,l_epok);
for i=1:l_epok
    i
    ww(:,i)=w;
    for j=1:l_p
        p=x(j);
        a=compet(-dist(w,p)); %detekcja neuronu zwycięskiego
        [dW,ls] = learnsom(w,p,[],[],a,[],[],[],[],d,lp,ls); %jeden cykl
    uczenia
        w=w+dW; %korekta wag
    end;
end;

figure(1);
plot(ww');
hold on;
plot(l_epok,x,'ok');
xlabel('nr epoki');
ylabel('w');
hold off;
legend(num2str((1:l_neur)'));

```

## Zawartość sprawozdania

Sprawozdania powinny być sporządzone według wzoru zamieszczonego na stronie i zawierać:

- A) Cel ćwiczenia.
- B) Treść zadania.
- C) Opis sieci neuronowej używanej w zadaniu (nie kopiuj treści wykładu, poszukaj w literaturze i Internecie).
- D) Metodyka rozwiązania zadania.
- E) Zestawienie wyników (wykresy, tabele z **komentarzem**).
- F) Wnioski końcowe.
- G) Wydruk programu.

## Zadania dodatkowe dla ambitnych

Wybrane zadanie student wykonuje indywidualnie, po uzgodnieniu z prowadzącym. Zadania nie są obowiązkowe. Z zadania sporządzamy sprawozdanie.

1. Dla wszystkich wariantów uczenia z p. 3 wyznacz średnie odległości punktów uczących od najbliższych neuronów. Jak zmieni się ta średnia odległość, gdy liczba epok uczenia wzrośnie do 1000?
2. Przeprowadź uczenie sieci Kohonena punktami z zadania aproksymacji (ćwicz. WPA). Pokaż na wykresie punkty uczące i końcowe wagi sieci.
3. Przeprowadź uczenie sieci Kohonena punktami 4-ro wymiarowymi. W układach współrzędnych  $x_1$ - $x_2$ ,  $x_1$ - $x_3$ ,  $x_1$ - $x_4$ , ...,  $x_3$ - $x_4$  pokaż te punkty i wagi końcowe sieci.
4. Oprogramuj przykład samoorganizującej się mapy cech odwzorowującej pięciowymiarowe wektory z rys. 7.15 [Żurada96].
5. Przeprowadź uczenie sieci Kohonena na punktach uczących rozłożonych jednorodnie wewnątrz trójkąta i koła (patrz rys. 8.10 [Osowski96]). Zastosuj tablicę liniową oraz prostokątną, w tym drugim przypadku zastosuj topologię prostokątną (gridtop) i heksagonalną (hextop). Powtórz uczenie dla danych

rozmieszczonych w dwóch trójkątach, przy czym liczba punktów w jednym trójkącie jest dwukrotnie większa niż w drugim.

6. Wykonaj podobne ćwiczenie w innym środowisku, np. R, Python, Statistica, C#, ...

## Przykładowe zagadnienia i pytania zaliczeniowe

1. Cel i przebieg ćwiczenia.
2. Narysuj model sieci Kohonena.
3. Na czym polega grupowanie danych.
4. Narysuj i objaśnij wykres ze sprawozdania.
5. Metody uczenia "zwycięzca bierze wszystko" i "zwycięzca bierze większość".
6. Znaczenie promienia sąsiedztwa i funkcji sąsiedztwa.
7. Topologie neuronów.
8. Fazy uczenia sieci.

## Do przygotowania na następne zajęcia

1. Zapoznać się z instrukcją do kolejnego ćwiczenia.
2. Zapoznać się z częścią teoretyczną do kolejnego ćwiczenia.
3. Wykonać zadania pomocnicze do kolejnego ćwiczenia.