Zadání příkladů pro 6.cvičení z předmětu MUIN

Sestavte Kohonenovu síť (mapu).
Jako tréninkové vzory použijte obrázky číslic 0 – 4 (uloženy ve formátu csv). Pro aktivaci použijte zašuměné číslice a vhodně vykreslete výstupu ze sítě (mapy).

Odevzdání
Vykreslení:

- Původního vzoru
- Zašuměného vzoru (použitého pro aktivaci)
- Výstup ze sítě (výstup všech neuronů) a vítězný neuron pro daný vzor

Neuron

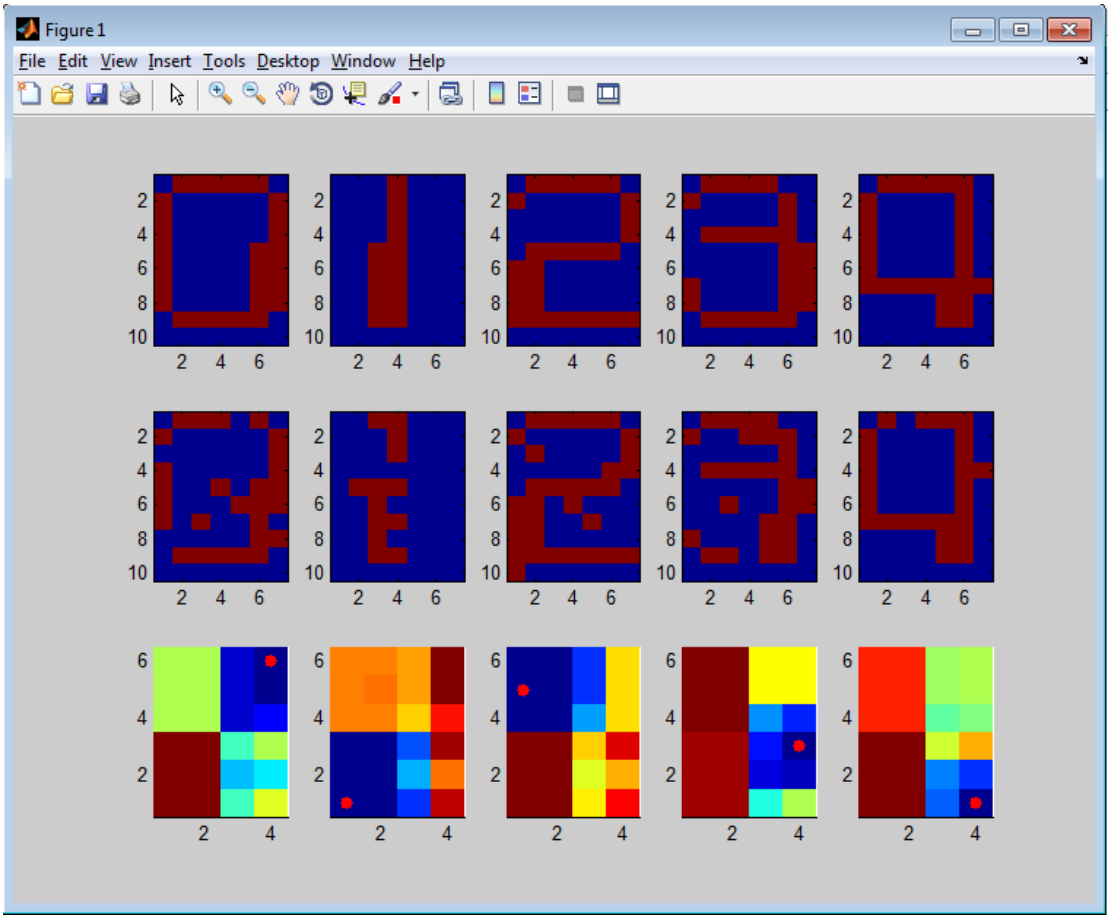
- Bez přenosové funkce a bez prahu
- Provádí výpočet vzdálenosti předloženého vzoru

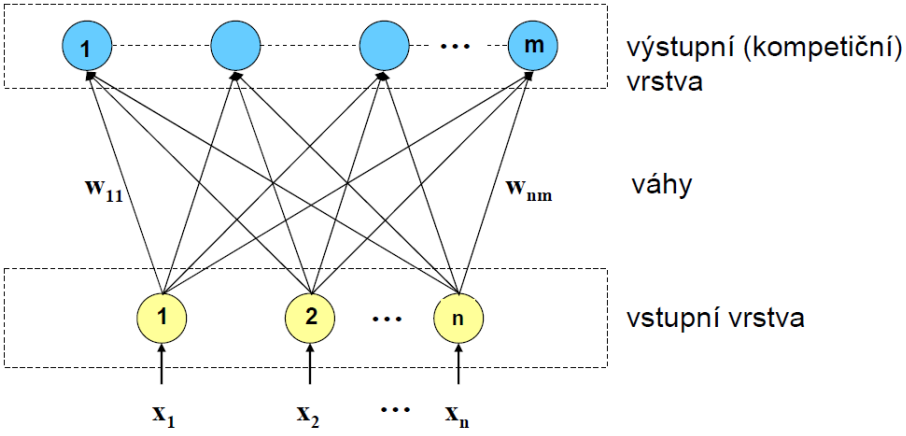
Topologie sítě

- Dvouvrstvá síť:
    - Vstupní vrstva – počet vstupů
    - výstupní vrstva 10 x 10 neuronů
- Topologická struktura určuje sousedství neuronů
- Čtvercové okolí (N = 2), nemění se

Doporučení

- Matice vah – 3 rozměrná (řádek, sloupec, váha), pozor na indexování, použití pomocné matice pro konkrétní neuron
- Počáteční nastavení vah 0,1 až 0,9
- Dvourozměrná matice pro výstup sítě
- Použití funkce *imagesc()* pro vykreslení výstupu sítě
- Použití funkce *scatter()* pro vykreslení vítězného neuronu do výstupu sítě
- Vzory načíst do dvourozměrné matice
- Počáteční nastavení koeficientu **alfa 0,6 až 1**, **beta 0,3 až 0,5**
- Nová hodnota alfy a bety 0,95 až 0,99 z předešlé hodnoty
- Počet iterací kolem 500





Obr. 1: Kohonenova síť

Algoritmus učení Kohonenovy sítě

Krok 1. Inicializace

Nastavíme váhy w_{ij}, pro všechny spoje z N vstupů do M výstupních neuronů
Nastavení parametrů *alfa* a *beta*
Nastavit úroveň změny těchto parametrů
Nastavení okolí (N)

Krok 2. Předložení vzoru

Předložíme nový trénovací vzor na vstupy neuronové sítě.

$$X(t) = \{x_0(t), x_1(t), \ldots, x_{N-1}(t)\}$$

Krok 3. Výpočet vzdálenosti vzorů

Vypočteme vzdálenosti (podobnosti) d_j mezi předloženým vzorem a všemi výstupními neurony *j* podle vztahu

$$d_j = \sum_{i=0}^{N-1} \left(x_i(t) - w_{ij}(t)\right)^2 ,$$

kde $x_i(t)$ jsou jednotlivé elementy vstupního vzoru $X(t)$ a $w_{ij}(t)$ jsou váhy mezi *i*-tým vstupem a *j*-tým výstupním neuronem, které představují zakódované vzory.

Krok 4. Výběr nejbližšího (vítězného) neuronu

Vybereme výstupní neuron j^*, který splňuje následující podmínku a odpovídá tak nejpodobnějšímu neuronu:

$$d_{j^*} = \min_j \left(d_j\right)$$

Krok 5. Přizpůsobení vah

Přizpůsobíme všechny váhy pro neuron j^* a jeho okolí $N_{j^*}(t)$, tj. pro všechny neurony ležící uvnitř tohoto okolí podle následujících vztahů:
Pro vítězný neuron:

$$w_{ij}(t+1) = w_{ij}(t) + \alpha(t)\left(x_i(t) - w_{ij}(t)\right)$$

Pro okolí:

$$w_{ij}(t+1) = w_{ij}(t) + \beta(t)\left(x_i(t) - w_{ij}(t)\right)$$

Krok 6. Pokračování učícího procesu

Kroky 2 až 5 opakovat pro všechny vzory

Krok 7. Aktualizace parametrů

Zmenšení parametrů alfa a beta o předem zvolenou změnu, např:

$$\alpha(nova) = \alpha(stara) * 0.98$$
$$\beta(nova) = \beta(stara) * 0.98$$

Krok 8. Test ukončení

Opakovat kroky 2 až 7 do vyčerpání počtu iterací.

Algoritmus aktivace Kohonenovy sítě

Krok 1. Předložení testovacího vzoru

Na vstupy sítě předložíme nový neznámý vzor

$$X(t) = \{x_0, x_1, \ldots, x_{N-1}\}$$

Krok 2. Výpočet nejbližšího vzoru

Nejprve spočítáme všechny vzdálenosti d_j daného předloženého výstupu od všech váhových vektorů výstupních neuronů podle známého vztahu:

$$d_j = \sum_{i=0}^{N-1} \left(x_i(t) - w_{ij}(t)\right)^2$$

a vybereme ten neuron j^*, který splňuje následující podmínku:

$$d_{j^*} = \min_j \left(d_j\right)$$

Tento j^*-tý neuron je tedy odpovědí sítě na předložený vzor. Udává nám vlastně třídu, do které náš nový vzor patří.

Krok 3. Opakování procesu

Pokud chceme pokračovat v klasifikaci dalších nových vzorů, pak přejdeme ke Kroku 1. V opačném případě skončíme.

Užitečné příkazy pro Matlab
Načtení bitmapy do matice:

A = imread(filename)
(nezapomeňte, že matice se dá indexovat pomocí (řádek, sloupec) nebo (pořadí prvku))

Načtení souboru *:csv do matice:
A = csvread('.csv')*

Vykreslení obrázku uloženého v matici, pro správné vykreslení je potřeba vynásobit 255:

image(A)

Zapnutí černobílého vykreslování:

colormap(gray)

Přidání šumu do obrazu:

Y = imnoise(vstup, ´typ sumu´, ...)

Změnění rozměrů matice:

out = reshape(in, řádky, sloupce)

Vykreslení výstupů ze sítě:

imagesc(A)

Zakreslení bodu do obrázku:

scatter(x, y, size, ´filled´)