

## Algoritmi metaeuristici.

### Lab 7:

**Rețele neuronale cu învățare nesupervizată. Aplicații în gruparea datelor, maparea datelor și optimizare combinatorială.**

**Antrenarea evolutivă a rețelelor neuronale**

---

### 1. Specificul învățării nesupervizate

Setul de antrenare conține doar date de intrare, rolul algoritmului de antrenare fiind de a identifica corelații între acestea care să permită extragerea de informații privind structura lor (de exemplu identificarea de grupuri de date similare).

Arhitectura suport pentru algoritmi de antrenare supervizată este de regulă constituită dintr-un nivel de unități de intrare și un nivel de unități funcționale. Cele mai cunoscute modele de rețele cu antrenare nesupervizată sunt:

- Rețelele bazate pe procese de competiție între unitățile funcționale (de tip WTA și ART) – pot fi utilizate în rezolvarea problemelor de grupare a datelor.
- Rețelele de tip Kohonen (utilizate pentru mapare topologică).

### 2. Gruparea datelor.

Gruparea datelor (clustering) are ca scop identificarea într-un set de date a unor subseturi de date similare între ele. Algoritmii partiționali de grupare permit identificarea câte unui prototip pentru fiecare dintre clase iar datele sunt asignate unei clase pe criteriul distanței minime față de prototip. Procesul de grupare poate fi ilustrat utilizând o rețea cu un nivel de unități liniare antrenată cu un algoritm de tipul WTA.

- Arhitectura.* Pentru gruparea unui set de date N-dimensionale în K clustere se consideră o rețea cu N unități de intrare și K unități de ieșire. Matricea W a ponderilor va avea K linii și N coloane. Fiecare dintre liniile matricii poate fi interpretată ca un vector prototip corespunzător cluster-ului identificat de unitatea funcțională corespunzătoare.
- Funcționare.* Pentru un vector de intrare X se determină unitatea funcțională  $k^*$  (numită unitate învingătoare) caracterizată prin faptul că  $d(X, W^{k^*}) < d(X, W^k)$  pentru orice  $k=1..K$ .
- Antrenare.* Pentru fiecare exemplu X din setul de antrenare se determină unitatea învingătoare  $k^*$  și se ajustează ponderile corespunzătoare acesteia folosind o regulă de forma:  $W^{k^*} = W^{k^*} + \eta(X - W^{k^*})$ .

**Aplicație 1.** Să se implementeze o rețea WTA pentru gruparea unor date bidimensionale.

**Indicație.** Vezi fișierul WTA.sci

### 3. Rețele Kohonen.

#### 3.1. Particularități ale rețelelor Kohonen

Principalele particularități ale rețelelor Kohonen (în raport cu rețelele de tip WTA) sunt:

- unitățile de ieșire sunt plasate în nodurile unei grile uni, bi sau tri-dimensionale și fiecare dintre unitățile de ieșire are asociată o mulțime de unități vecini

(vecinătatea este determinată de tipul de distanță utilizat și de parametrul care controlează dimensiunea vecinătății).

- pentru fiecare element din setul de antrenare se ajustează atât ponderile unității învingătoare cât și cele ale unităților vecine.

Rețelele Kohonen pot fi utilizate pentru maparea topologică a unor date multi-dimensionale într-un spațiu de dimensiune mai mică.

**Aplicație 2.** Implementarea unei rețele Kohonen cu grilă pătratică și maparea acesteia pe un set de date generate uniform aleator în  $[0,1] \times [0,1]$ .

Indicație: fișier SOM.sci

**Exercițiu.** Reantrenați și testați rețeaua în cazul unor date generate în interiorul unei alte figuri geometrice (cerc, coroana circulară etc).

### 3.2. Rezolvarea problemei comis voiajorului folosind rețele Kohonen

Principalele caracteristici ale acestei abordări:

- Rețelele de tip Kohonen au o arhitectură constituită din două unități de intrare și  $m$  unități de ieșire plasate în nodurile unei topologii de tip inel: vecinătatea de dimensiune  $2s$  a unui neuron  $i$  conține mulțimea de neuroni  $\{i-s, i-(s-1), \dots, i-1, i, i+1, \dots, i+(s-1), i+s\}$ . Numărul de unități funcționale trebuie să fie mai mare decât numărul de orașe ale problemei (de exemplu  $m=5n$ )
- Fiecare unitate funcțională are asociate două ponderi inițializate aleator cu valori din domeniul coordonatelor orașelor.
- Procesul de antrenare constă în parcurgerea de mai multe ori a listei cu coordonate ale orașelor și ajustarea ponderilor folosind regula specifică. Pentru fiecare vector de intrare  $x$  (pereche de două coordonate) se efectuează următoarele prelucrări:
  - Se determină unitatea învingătoare,  $i^*$ , (cea având vectorul de ponderi cel mai apropiat de  $x$ )
  - Se ajustează ponderile tuturor unităților folosind regula:
    - $$W^i(k+1) = W^i(k) + \eta(X - W^i(k))\Lambda(i, i^*)$$
unde funcția de vecinătate este  $\Lambda(i, i^*) = \exp(-(i - i^*)^2 / (2s^2))$

**Aplicație 3.** Să se implementeze rețeaua Kohonen descrisă mai sus.

Indicație. Vezi fișierul Kohonen\_TSP.sci

**Exercițiu:** Scrieți o funcție care să calculeze costul traseului generat.

## 4. Proiectarea evolutivă a rețelelor neuronale

Algoritmii evolutivi pot fi utilizați în contextul proiectării rețelelor neuronale pentru:

- 4.1. Determinarea ponderilor (în locul algoritmilor de antrenare bazați pe metode de optimizare locală se folosește o strategie evolutivă)
- 4.2. Stabilirea arhitecturii

#### **Aplicatie 4.** Determinare evolutivă a ponderilor unei rețele neuronale

Considerăm problema antrenării unei rețele neuronale pentru reprezentarea disjuncției exclusive. Se consideră o rețea constituită din:

- 3 unități de intrare (două efective și una corespunzătoare pragului)
- $K$  unități ascunse ( $K$  este un parametru de intrare) cu funcție de transfer de tip Heaviside
- o unitate de ieșire cu funcție de transfer de tip Heaviside

Fiecare element din populație conține  $4 \cdot K + 1$  elemente corespunzătoare tuturor ponderilor și valorilor prag. Criteriul de optimizat este eroarea medie pătratică pe setul de antrenare iar algoritmul utilizat poate fi o strategie evolutivă.

*Indicație.* Se adaugă la aplicația de la strategii evolutive o funcție pentru evaluarea erorii medii pătratice în cazul unei rețele cu arhitectură specificată (vezi [SEreteea.sci](#)).

#### **Temă.**

1. Să se modifice strategia evolutivă din [SE.sci](#) prin includerea auto-adaptării (varianta din cursul 8)
2. Extindeți aplicația cu antrenarea evolutivă a unei rețele neuronale (pentru reprezentare XOR) pentru cazul în care numărul de unități ascunse este variabil (elementele populației vor conține atât numărul de unități ascunse cât și valorile ponderilor – prin urmare elementele populației vor avea număr variabil de componente). Observație: e necesară adaptarea operatorilor de recombinare și mutație
3. Implementați o rețea de tip ART (detalii în curs 4) pentru identificarea de clustere într-un set de date bidimensionale. *Indicație.* Se modifică algoritmul de antrenare din implementarea [WTA.sci](#)