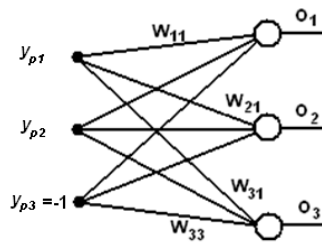


# Inteligență computațională - laborator

## Regula delta

*Regula delta* este o metodă de instruire supervizată care se aplică unei rețele monostrat de perceptroni de tip continuu. În figura de mai jos avem o rețea care folosește la intrare pattern-uri cu două componente. Întotdeauna se adaugă o intrare suplimentară cu valoarea constantă -1. Între intrări și ieșiri se realizează toate legăturile posibile. În exemplul nostru se obțin 3x3 conexiuni notate prin  $w_{ji}$ , unde  $j$  este indicele perceptronului, iar  $i$  este indicele intrării. De exemplu, conexiunea care leagă intrarea 1 de perceptronul 2 este  $w_{21}$ , conexiunea care leagă intrarea suplimentară cu perceptronul 3 este  $w_{33}$ .

Funcția de activare a perceptronilor este continuă. Vom folosi varianta continuă bipolară care are formula  $f(net) = \frac{2}{1+e^{-net}} - 1$ .



Algoritmul de mai jos implementează *regula delta*:

1. Se citesc pattern-urile de instruire  $\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_N$  și ieșirile dorite  $\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_N$ . Se scalează toate valorile din setul de date. Se stabilește numărul  $M$  de intrări și numărul  $K$  de perceptroni. Se inițializează ponderile  $w_{ji}$  cu valori aleatoare din intervalul  $[-1;1]$ . Se inițializează constanta de instruire  $c$ . Se stabilește eroarea maximă  $E_{max}$ . Se inițializează  $E=0$ .
2. Având un pattern  $\mathbf{y}_p = [y_{p1}, \dots, y_{pi}, \dots, y_{pM}]$  dintre cele  $n$ , se calculează ieșirile fiecărui perceptron:

$$o_j = f(w_{j1}y_{p1} + w_{j2}y_{p2} + \dots + w_{jM}y_{pM}), \quad j=1 \dots K,$$

unde  $K$  este numărul de perceptroni,  $M$  este numărul de intrări, iar  $f$  este funcția de activare bipolară.

3. Se actualizează toate ponderile rețelei conform regulii:

$$w_{ji}^{(t+1)} = w_{ji}^{(t)} + c(d_{pj} - o_j)(1 - o_j^2)y_{pi},$$

unde  $\mathbf{d}_p = [d_{p1}, \dots, d_{pj}, \dots, d_{pK}]$  reprezintă ieșirile dorite pentru pattern-ul  $\mathbf{y}_p$ .

4. Se calculează eroarea cumulată:

$$E^{(t+1)} = E^{(t)} + (d_{p1} - o_1)^2 + (d_{p2} - o_2)^2 + \dots + (d_{pK} - o_K)^2.$$

5. Se revine la pasul 2 cu un nou vector de instruire, până la epuizarea setului de  $N$  pattern-uri.
6. Dacă  $E > E_{max}$ , se revine la pasul 2 pentru o nouă epocă de instruire în care se reiau toate pattern-urile de instruire și se reinițializează  $E=0$ .

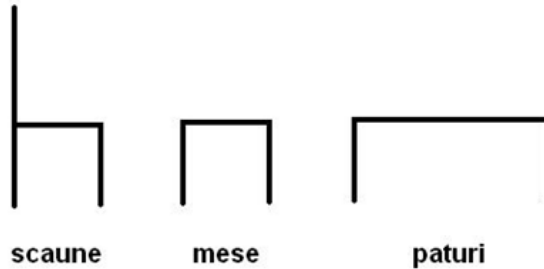
Implementați *regula delta* pentru o rețea de  $K=3$  perceptroni care folosește pattern-uri bidimensionale cărora li se adaugă a treia intrare -1,  $\mathbf{y}_1, \mathbf{y}_2 \dots \mathbf{y}_9$  cu ieșirile dorite  $\mathbf{d}_1, \mathbf{d}_2 \dots \mathbf{d}_9$ :

## Inteligență computațională - laborator

45	85	1	-1	-1
50	43	-1	1	-1
40	80	1	-1	-1
55	42	-1	1	-1
200	43	-1	-1	1
48	40	-1	1	-1
195	41	-1	-1	1
43	87	1	-1	-1
190	40	-1	-1	1

Aceste pattern-uri reprezintă lățimea și înălțimea unor obiecte de mobilier asemănătoare celor din imaginea de mai jos, iar cele trei clase sunt codificate astfel:

- scaunele prin (1, -1, -1)
- mesele prin (-1, 1, -1)
- paturile prin (-1, -1, 1).



La final afișați pattern-urile de instruire și răspunsurile celor trei perceptroni pentru fiecare pattern.