

4 p of

- I. a) Specificați două proceduri de estimare a lui $R_{real}(\phi)$. Ce proprietăți are estimatorul furnizat de procedură?
 b) Specificați două reguli de învățare robuste (pt. perceptra). Justificați

II. Construiți o RPM standard care să implementeze forma normal des-junctivă a funcției $f: \{0,1\}^2 \rightarrow \{0,1\}$ cu $f(x,y) = [(2x+y) \bmod 2]$.

III. a) Scrieți ecuațiile de actualizare a ponderilor din alg. backprop pt. o rețea ff cu funcții de transfer tanh, funcția de eroare criterial corelației folosind metoda gradientului descendent

b) Scrieți în MATLAB subrutina
 function [w,b] = algRb (w,b,S)

ce implementează alg Rosenblatt pt. un perceptra cu funcția de transfer sigmoid, antrenat 'off-line', pe o mulțime de antrenare $S = [p; t]$ cu $p \in \mu_{d \times n}$ și $t \in \mu_{1 \times n}$ cu rata de învățare 0.5. Intrările corespund lui $w^{(t)}$, $b^{(t)}$ iar ieșirile lui $w^{(t+1)}$, $b^{(t+1)}$.

10 pte

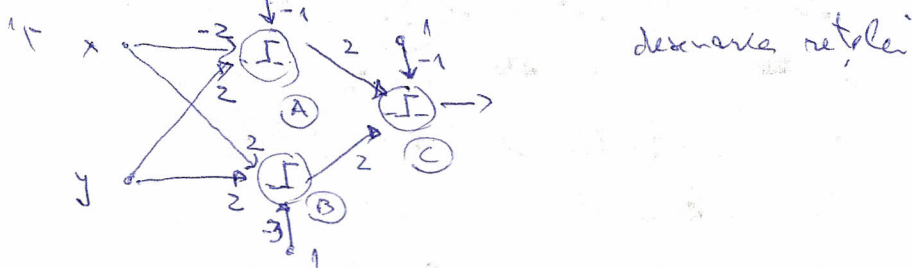
Țimp de lucru 1:30 h

- Soluții -

I. (2p) 1p a) Estimatic lui $R_{real}(h)$ este notată $\hat{R}_{real}(h)$ și calculată prin variante ale metodei validării încrucișate (cross-validation). Variante ale acestei metode sunt: k -fold cross-validation

• leave one out method
1p b). Regula lui May care nu permite "apropierea de zero"
• Regula lui Bute care evită "oscilațiile" în fața lui w
• sau regulile AAK I-III.

II (3p) 1p $f(x,y) = [(2x+y) \bmod 2] = K_1 \vee K_2 = (1x \wedge y) \vee (x \wedge 1y)$ forma normal disjunctivă



1p deus credibilitate

x	y	$f(x,y)$	K_1	K_2	$K_1 \vee K_2$	A		B		C	
						G	bad	G	bad	G	bad
0	0	0	0	0	0	-1	0	-3	0	-1	0
0	1	1	1	0	1	1	1	-1	0	1	1
1	0	0	0	0	0	-3	0	-1	0	-1	0
1	1	1	0	1	1	-1	0	1	1	1	1

III (4p) a) $J(\underline{w}^{(1)}, \underline{w}^{(2)}) = - \sum_{e=1}^L d_e f_0(wet_e)$ M. Actes din fig 3.1.

2p $net_e = \sum_{j=0}^J w_{ej}^{(1)} z_j$, $z_j = f_h(net_j)$, $wet_e = \sum_{i=0}^n w_{ji}^{(2)} x_i$, $f_0(x) = f_h(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$

1p $\underline{w}^{(1)} = \begin{pmatrix} w_{10}^{(1)} & \dots & w_{1J}^{(1)} \\ \vdots & & \vdots \\ w_{L0}^{(1)} & \dots & w_{LJ}^{(1)} \end{pmatrix}$, $\underline{w}^{(2)} = \begin{pmatrix} w_{10}^{(2)} & \dots & w_{1n}^{(2)} \\ \vdots & & \vdots \\ w_{J0}^{(2)} & \dots & w_{Jn}^{(2)} \end{pmatrix}$

multimea de antrenare $S = \{(\underline{x}, \underline{d}) \mid \underline{x} = (x_1, \dots, x_n)'$, $\underline{d} = (d_1, \dots, d_L)'\}$
Actualizarea ponderilor pe stratul de ieșire

$$\Delta w_{ej}^{(1)} = -\eta \frac{\partial J(\underline{w}^{(1)}, \underline{w}^{(2)})}{\partial w_{ej}^{(1)}} = \eta d_e f'_0(wet_e) z_j \quad \begin{matrix} e=1, L \\ j=0, J \end{matrix}$$

cu $\eta > 0$ rată de învățare și $f'_0(x) = 1 - f_0^2(x)$

→ (Estimare (pag 1))

III a) Actualizarea ponderilor pe stratul ascuns

$$\Delta w_{ji}^{(2)} = -\beta_h \frac{\partial J}{\partial w_{ji}^{(2)}}$$

$$\bar{j} = \overline{[j]} \\ i = 0, n$$

$$\frac{\partial J}{\partial w_{ji}^{(2)}} = \frac{\partial J}{\partial z_j} \cdot \frac{\partial z_j}{\partial \text{net}_j} \cdot \frac{\partial \text{net}_j}{\partial w_{ji}^{(2)}}$$

cu $\frac{\partial \text{net}_j}{\partial w_{ji}^{(2)}} = x_i$; $\frac{\partial z_j}{\partial \text{net}_j} = f'_h(\text{net}_j)$

$$\frac{\partial J}{\partial z_j} = \frac{\partial}{\partial z_j} \left[- \sum_{l=1}^L d_l f_0(\text{net}_l) \right] = - \sum_{l=1}^L d_l \frac{\partial f_0}{\partial z_j} = \sum_{l=1}^L d_l f'_0(\text{net}_l) w_{lj}^{(1)}$$

$$\Rightarrow \Delta w_{ji}^{(2)} = \beta_h \left[\sum_{l=1}^L d_l f'_0(\text{net}_l) w_{lj}^{(1)} \right] f'_h(\text{net}_j) x_i$$

cu $\beta_h > 0$ rate de învățare în $f'_h(x) = 1 - f^2(x)$

b) funcția $[w, b] = \text{algRo}(w, b, s)$

$[d1, n] = \text{size}(s);$

$d1 = d1 - 1;$

$p = s[1:d1, :];$

$t = s[d1, :];$

$a = \text{handlin}(w * p, b)$

$e = (t - a) / 2;$

$dw = e * p';$

$db = e * \text{ones}(n, 1);$

$w = w + dw;$

$b = b + db;$