

RETELE NEURONALE 2020

MATEMATICA-INFORMATICA

Retele neuronale

Elemente RNA

- Rezolvarea problemelor folosind RNA
- Algoritmi de clasificare RNA

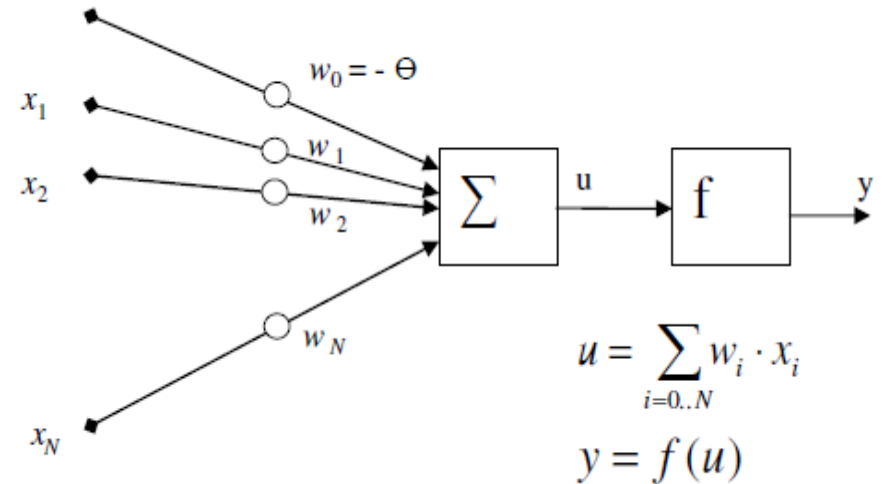
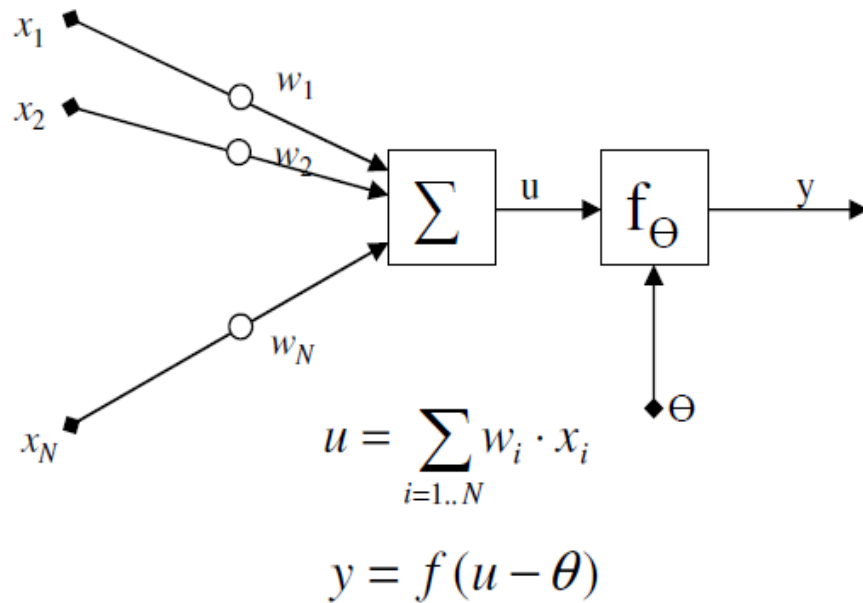
RNA este complet determinată prin:

- tipul unităților funcționale (elemente de procesare numite neuroni)
- arhitectură (amplasare unități funcționale)
- algoritm de funcționare (transformare semnal intrare în semnal ieșire)
- algoritm de învățare (cum achiziționează rețeaua noi cunoștințe pe bază de exemple)

Neuronul artificial

Modelul de bază McCulloch-Pitts (1943)

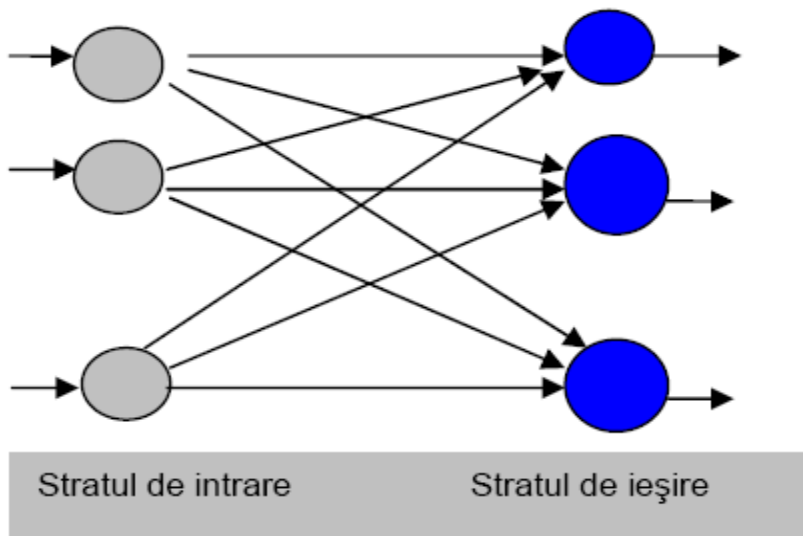
Modelul derivat



w – ponderi sinaptice
 f – funcție de integrare/agregare/activare
 Θ – prag (b- polarizare)

Arhitectura RNA

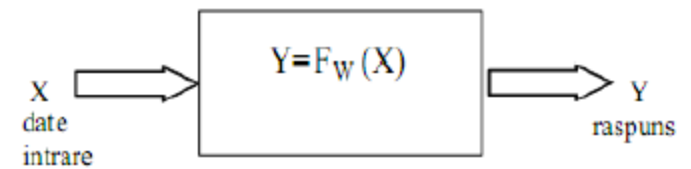
➤ rețele feed-forward (unidirecționale)



unistrat

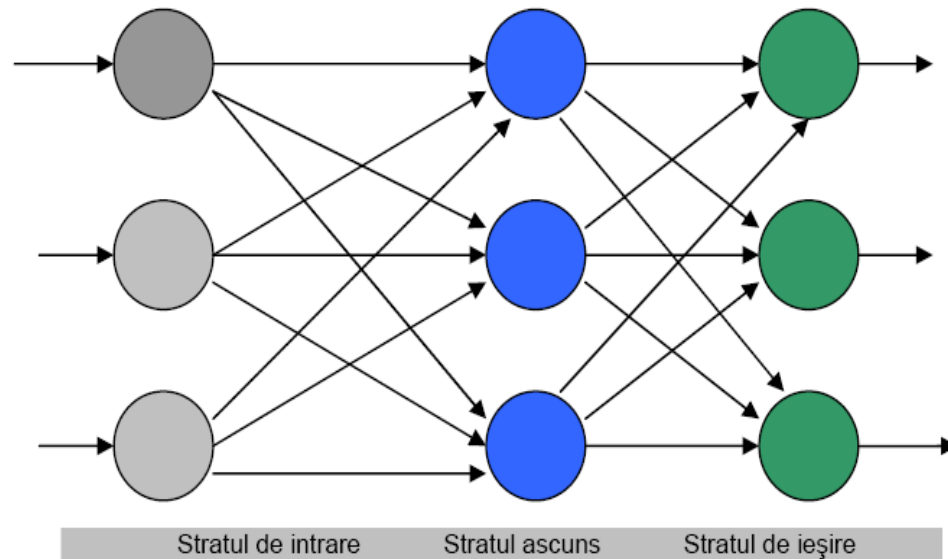
$$X = \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_N \end{bmatrix} \quad W = \begin{bmatrix} w_{11} \dots w_{1M} \\ w_{21} \\ \dots \\ w_{N1} \dots w_{NM} \end{bmatrix} \quad B = \begin{bmatrix} b_1 \\ b_2 \\ \dots \\ b_M \end{bmatrix}$$

$$Y = f(W^T \cdot X + B)$$



Arhitectura RNA

➤ rețele feed-forward (unidirecționale)



multistrat

- mai lente decât cele unistrat
- pot implementa funcții mai complexe

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_N \end{bmatrix} \quad W = \begin{bmatrix} w_{11} \dots w_{1M} \\ w_{21} \\ \dots \\ w_{N1} \dots w_{NM} \end{bmatrix} \quad B = \begin{bmatrix} b_1 \\ b_2 \\ \dots \\ b_M \end{bmatrix}$$

$$Y = f(W^T \cdot X + B)$$

A block diagram showing the function $Y = F_W(X)$. An input vector X (labeled 'date intrare') enters a rectangular box from the left. The box contains the text $Y = F_W(X)$. An output vector Y (labeled 'raspuns') exits the box to the right.

Instruirea RNA = procesul adaptării ponderilor, printr-o stimulare din partea unui expert sau a mediului, sau nesupervizat, prin analiza statistică a vectorilor de intrare

Algoritm de instruire = modul în care se modifică ponderile

$$W_{kj}(n+1) = W_{kj}(n) + \Delta W_{kj}(n)$$

unde:

- k este indicele stratului de neuroni,
- j este indicele neuronului pe stratul k ,
- n este momentul de timp.

$$\Delta W_{kj}(n) = \text{funcție (algoritm de instruire)}$$

Utilizarea RNA in rezolvarea de probleme

- Etape de lucru pentru rezolvarea problemelor folosind RNA
- Aspecte de lucru

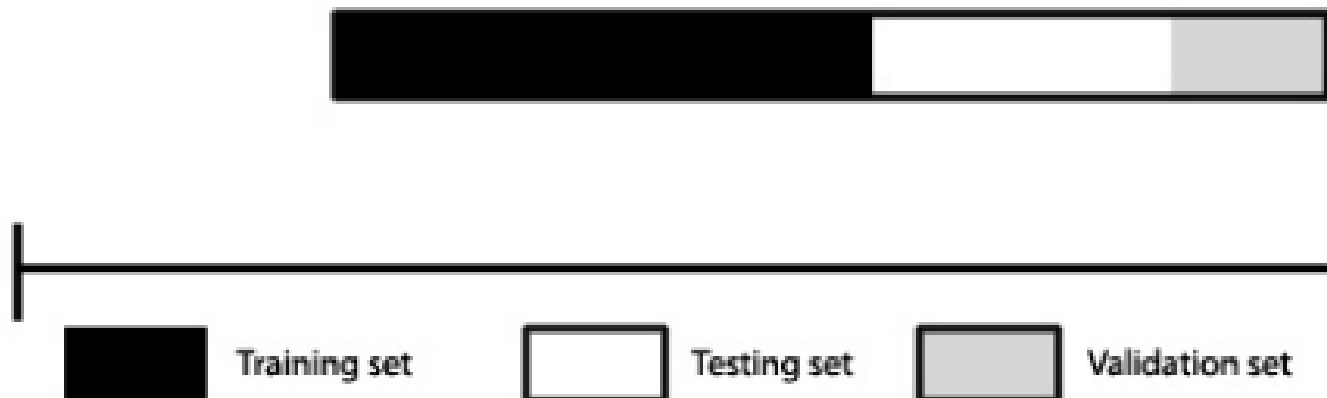
Etapele rezolvării problemelor folosind RNA

- stabilirea arhitecturii inițiale
- alegerea unui algoritm de instruire
- antrenarea rețelei
- testare (validare)
- utilizarea propriu-zisă

Aspecte de proiectare

Alegerea setului de date de intrare

- antrenare – 50% din numarul total de inregistrari/date
- validare – 20% - 40% din numarul total de date
- testare – 10% - 30% din numarul total de date



Aspecte de proiectare

Număr de neuroni

Rețea cu 3 straturi (alegere posibilă):

$$p_{hidden} = \sqrt{n_{in} * m_{out}}$$

$m_{out} > 1$ - rezultate inferioare față de $m_{out} = 1$

Observație: pentru aplicații de clasificare, $m_{out} = nr. \text{ clase}$

Aspecte de proiectare

Inițializare ponderi

Valori aleatoare - random, într-un interval simetric față de 0

Exemplu: [-0.5; 0.5]

$$\left[-\frac{1}{\# ponderi}; \frac{1}{\# ponderi} \right]$$

#ponderi – numărul de conexiuni (ponderi) care intră într-un nod

Valori mai mici pentru nodurile în care intră mai multe conexiuni

Aspecte de proiectare

Pregătirea setului de date (data preparation)

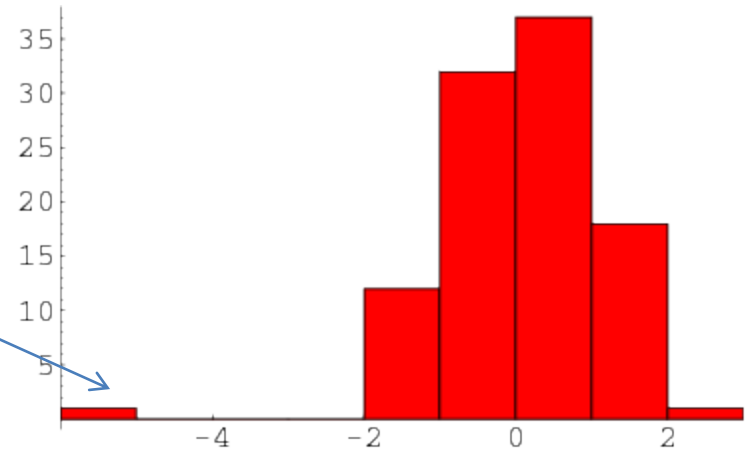
Tipuri de date:

Simbolice: A, da/nu, mic/mare/mediu

Numerice discrete: -2, 5, 34

Numerice continue: -3.421, 98.02

- eliminare excepții vizibile (outliers)
- eliminare date redundante
- transformare și codare



Perceptronul - Clasificator RNA

- Etapele proiectării unui clasificator RNA
- Clasificare liniară
- Perceptronul - algoritm de clasificare liniară
- Limite ale clasificării cu perceptronul

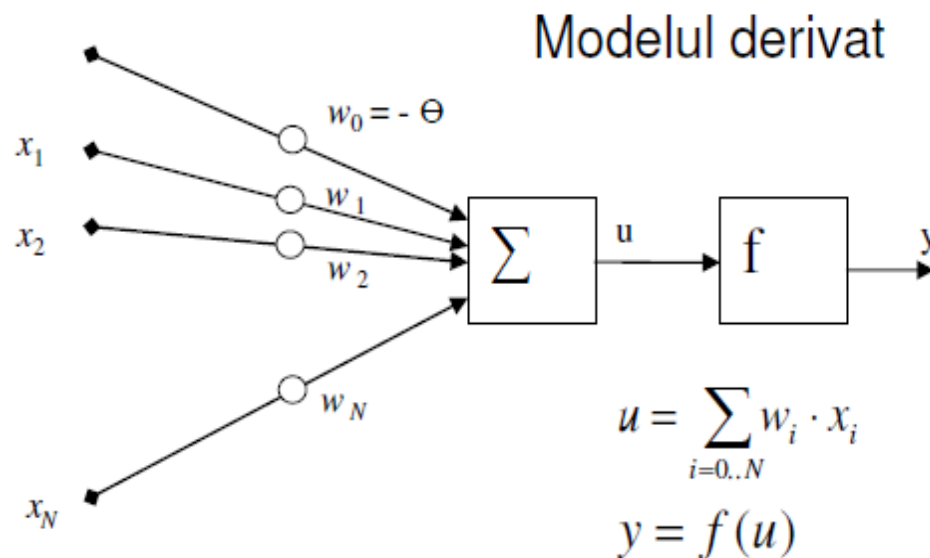
Etapele proiectării unui clasificator

- stabilirea claselor de obiecte din mulțimea datelor de instruire
- stabilirea prototipurilor claselor
- stabilirea unei reguli pentru alocarea unui obiect necunoscut la una din clasele stabilite anterior

Clasificare liniară

Perceptronul

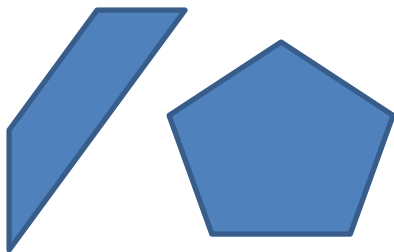
- clasificator binar (2 clase)
- clase liniar separabile
- hiperplan de separație



funcție de activare binară

Clasificare liniară

Perceptronul



clase convexe

2 straturi de neuroni

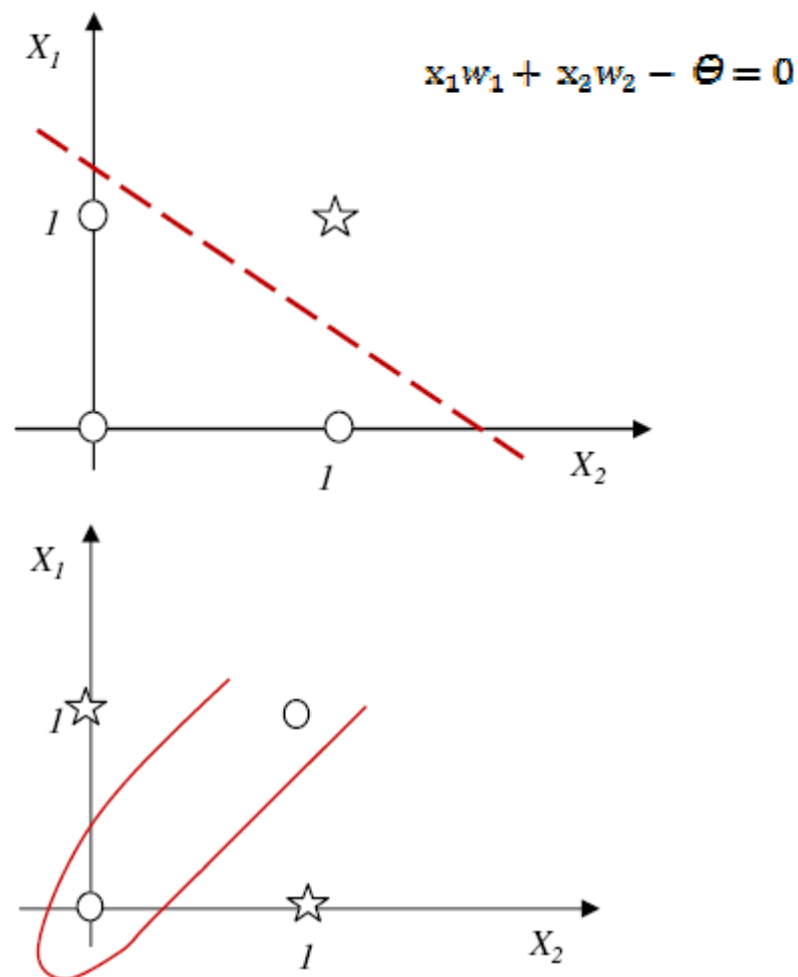


clase concave

3 straturi de neuroni

Clasificare liniară Separabilitate

X1	X2	AND	OR	XOR
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0



XOR nu se poate implementa cu perceptron!

Alte structuri

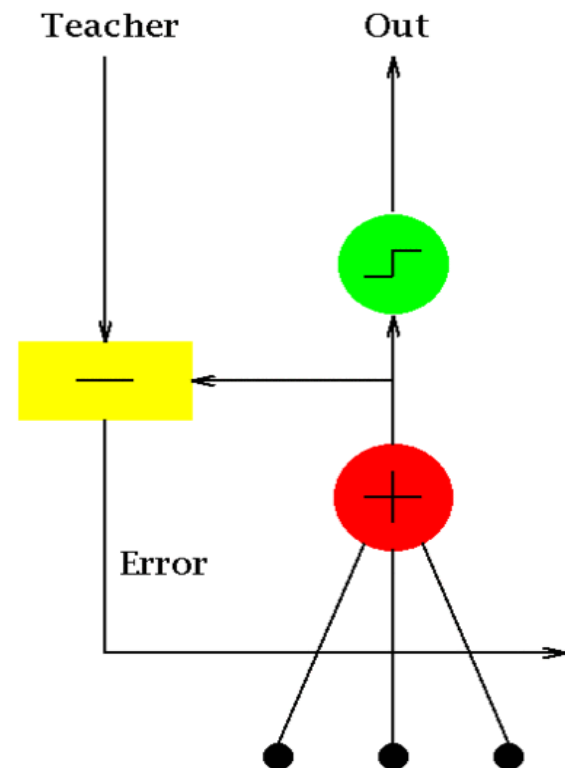
ADALINE (Widrow, 1960)

Adaptive Linear Neuron/Element

- RNA cu un singur strat
- adaptare supervizată

Adaptarea ponderilor se face pe baza sumei ponderate a intrărilor.

La perceptron, adaptarea se face pe baza ieșirii funcției de activare.



MADALINE – mai multe elemente de tip ADALINE