

REȚELE NEURONALE 2021

SPECIALIZAREA INFORMATICĂ

ANUL II - Semestrul al II-lea

NOTE de Curs Rețele Neuronale

Elemente

- Principii
- Bazele biologice
- Arhitectura RNA (retea neuronală artificială)
- Instruire
- Tipuri de probleme rezolvabile cu RNA

Principiile RNA

- Calcul Natural și Calcul Neural
- Retele Neuronale Artificiale (RNA)
- Componentele care determină o rețea neuronală

Încadrarea în Inteligența Computațională - IC

Inteligentă
artificială

Logică
fuzzy

Rețele
neuronale

Calcul
evoluționist

Hard
computing
(calcul convențional)

pur simbolic

inteligentă computațională (soft computing)

pur numeric

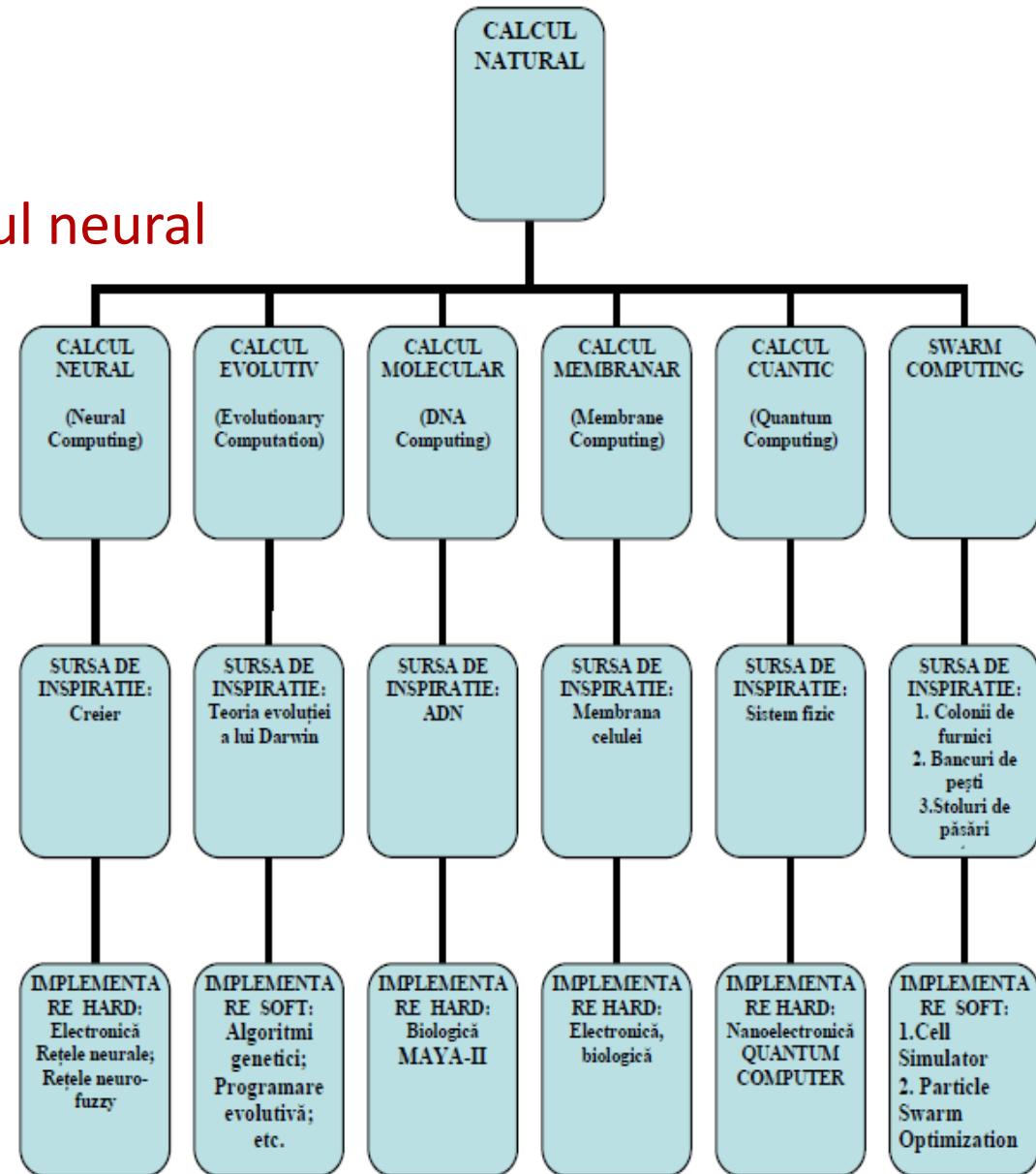
IC = modelarea numerică a inteligenței biologice

- suprapunere semnificativă cu soft computing
- exploatează **toleranță la imprecizie, nesiguranță și adevăr parțial**, pentru a obține **flexibilitate, robustețe și costuri scăzute**
- diferă **fundamental** de calculul convențional (hard), caracterizat de lipsa impreciziei și a adevărurilor parțiale

Calcul neural (neuronal)

RN = suport hard pentru calcul neural

- rezolvarea problemelor de asociere
- se bazează pe extragerea unui model pe bază de exemple
- învățare pe bază de suficient de multe exemple



Alte definiții

RNA = calculator **distribuit, masiv paralel**, care achiziționează **noi** cunoștințe pe baza experienței anteroioare și le face disponibile pentru utilizarea **ulterioară** (S.Haykin, 1994)

Asemănarea cu creierul uman

- cunoștințele sunt achiziționate de rețea neurală printr-un proces de **învățare**
- cunoștințele sunt depozitate în **conexiunile inter-neuronale** (ponderi sinaptice)

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)

Denumiri alternative pentru RNA

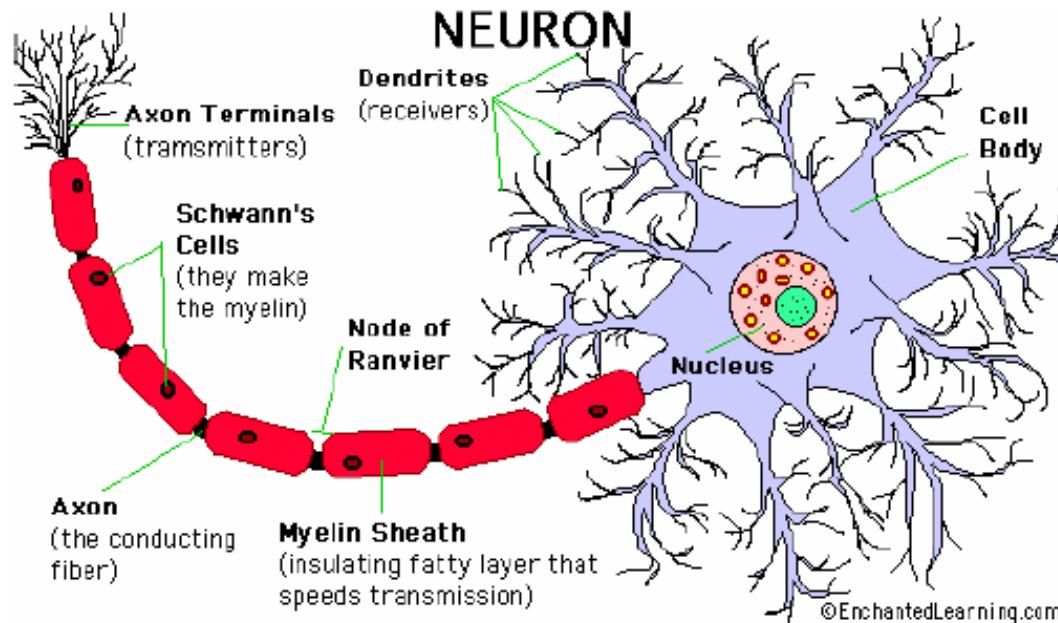
- neurocalcul
- conexionism
- procesare paralelă distribuită
- sisteme adaptive
- rețele cu auto-organizare

Bazele biologice ale RNA

- Descrierea neuronului natural
- Descrierea modelului de neuron artificial
- Exemple de funcții de activare

Neuronul natural

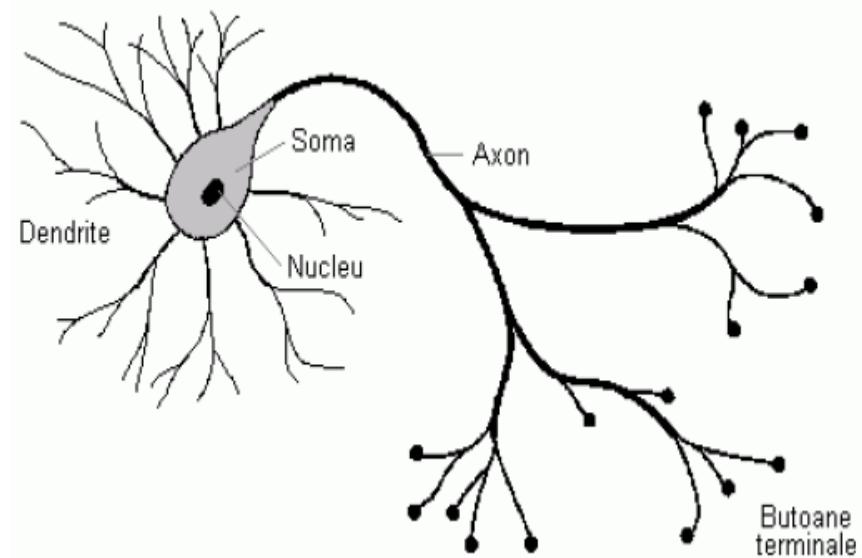
- unitate morfo-funcțională a sistemului nervos



Număr aproximativ de 100 mld. neuroni în creier și măduva spinării
Estimare în jur de 1000-10.000 sinapse/neuron

Tipuri de neuroni naturali

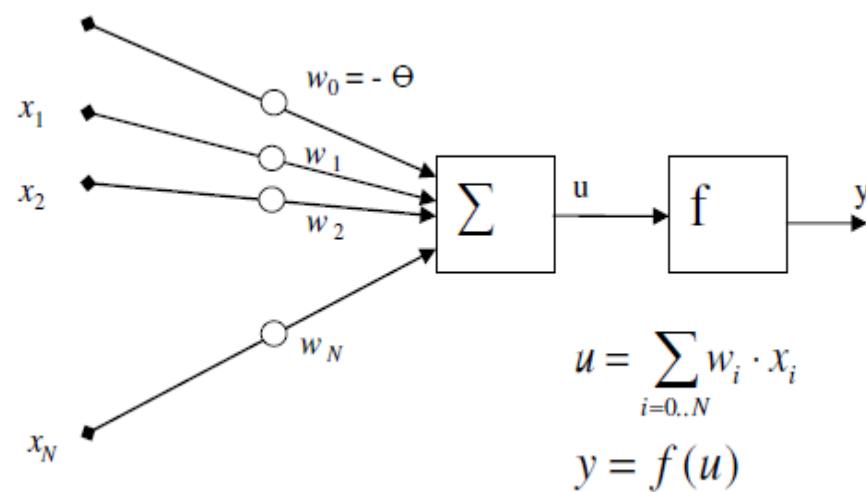
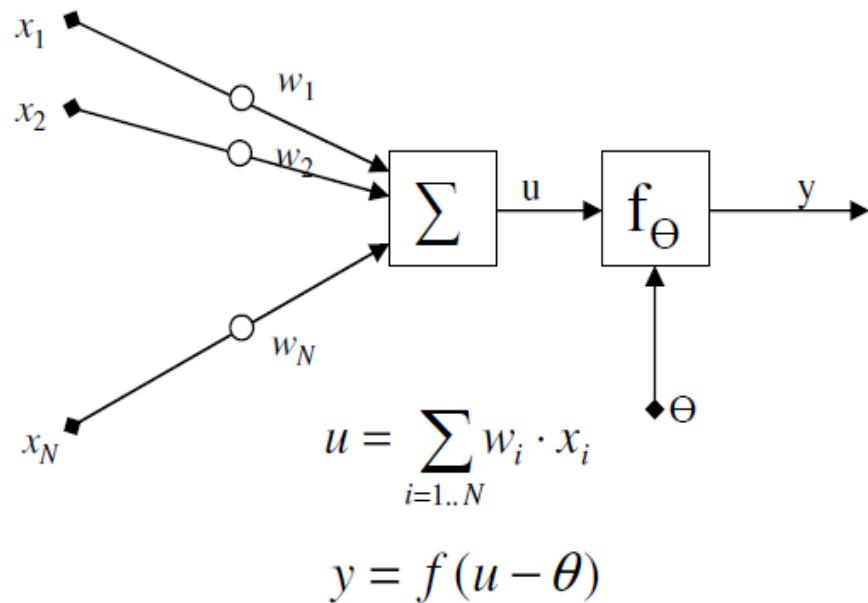
- senzoriali (bipolari) (0.9%)
- moto-neuron (multipolari) (9%)
- inter-neuroni (pseudopolari)



Neuronul artificial

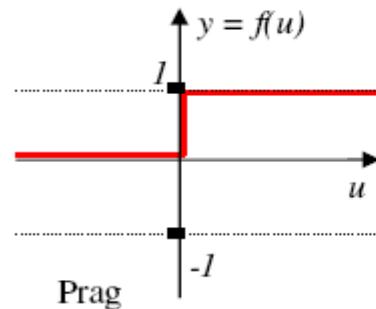
Modelul de bază McCulloch-Pitts (1943)

Modelul derivat

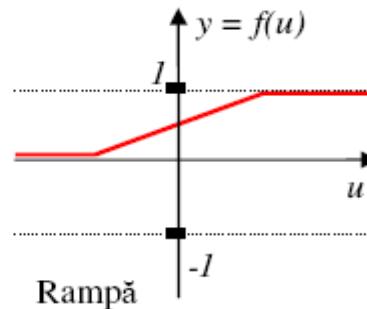


w – ponderi sinaptice
f – funcție de integrare/agregare/activare
Θ – valoare prag (b-polarizare)

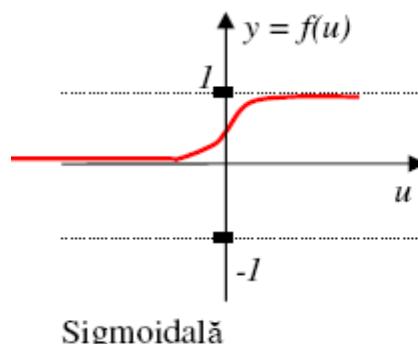
Funcții de activare



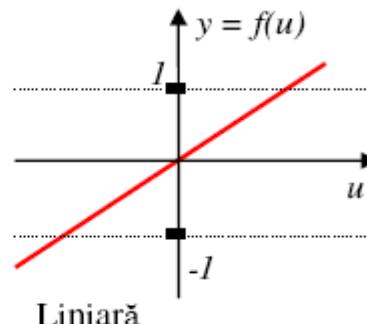
$$f(u) = \begin{cases} 0, u \leq 0 \\ 1, u > 0 \end{cases}$$



$$f(u) = \begin{cases} 0, u \leq 0 \\ u/k, 0 < u \leq k \\ 1, k < u \end{cases}$$



$$f(u) = \frac{1}{1 + e^{-ku}}$$

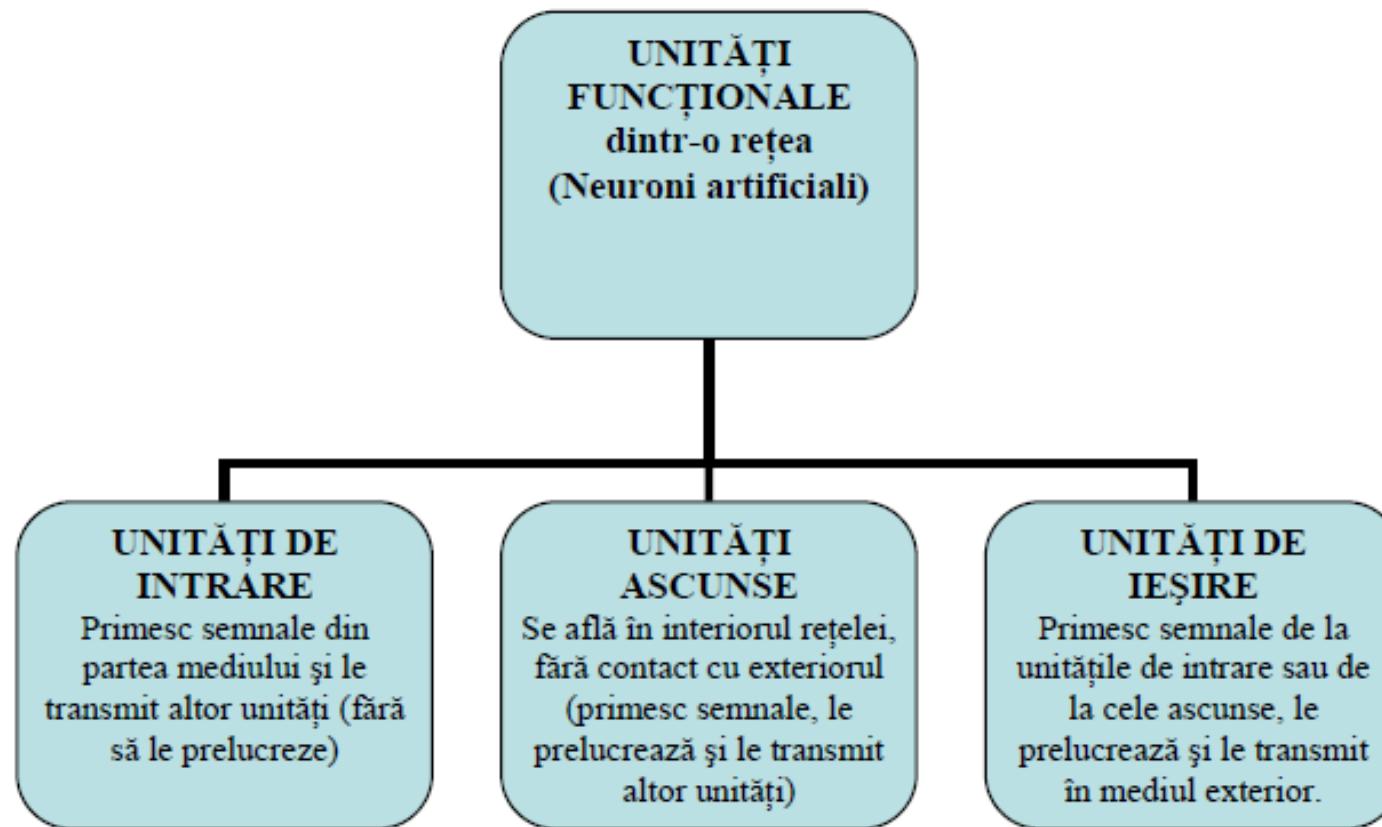


$$f(u) = ku$$

Arhitectura RNA

- Unitățile funcționale - Straturile de neuroni ale RNA
- Tipuri de arhitecturi RNA în funcție de instruire
- Caracteristici preluate de RNA de la creier

Unități funcționale/ Straturi de neuroni ale RNA



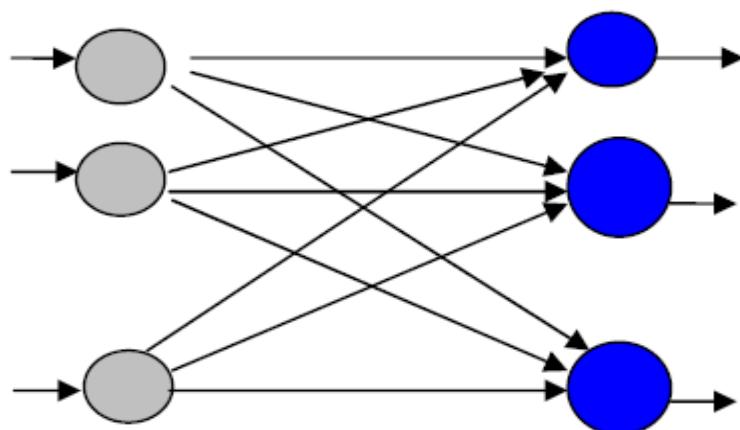
Input Layer RNA

Hidden Layer RNA

Output Layer RNA

Arhitectura RNA

- rețele feed-forward (unidirecționale)



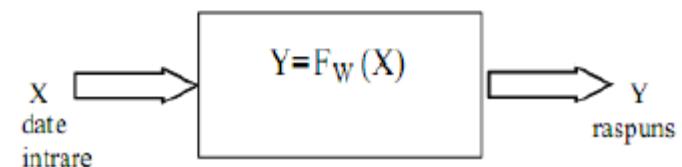
Stratul de intrare

Stratul de ieșire

unistrat

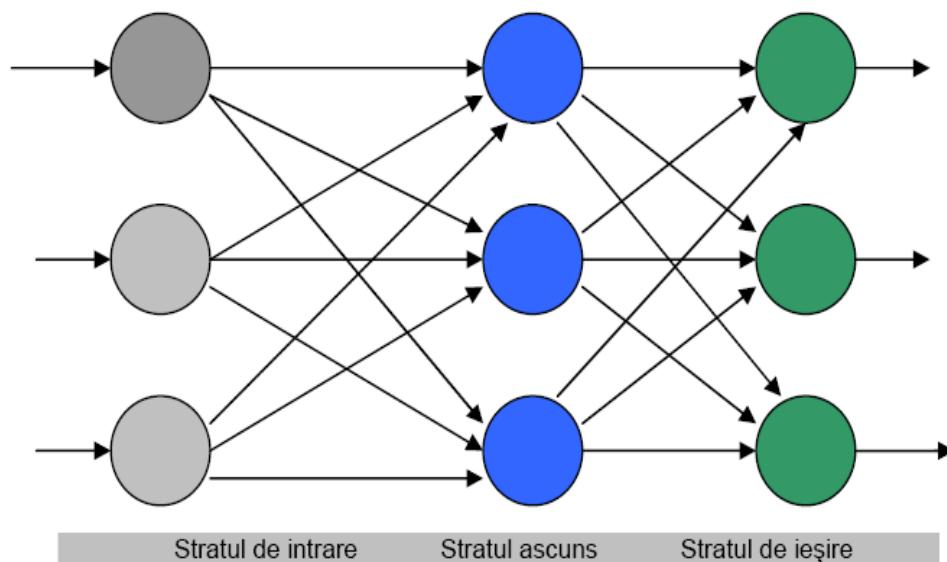
$$X = \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_N \end{bmatrix} \quad W = \begin{bmatrix} w_{11} \dots w_{1M} \\ w_{21} \\ \dots \\ w_{N1} \dots w_{NM} \end{bmatrix} \quad B = \begin{bmatrix} b_1 \\ b_2 \\ \dots \\ b_M \end{bmatrix}$$

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



Arhitectura RNA

- rețele feed-forward (unidirecționale)



multistrat

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

$$\begin{aligned}
 X &= \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} & W &= \begin{bmatrix} w_{11} \dots w_{1M} \\ w_{21} \\ \vdots \\ w_{N1} \dots w_{NM} \end{bmatrix} & B &= \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_M \end{bmatrix} \\
 Y &= f(W^T \cdot X + B)
 \end{aligned}$$

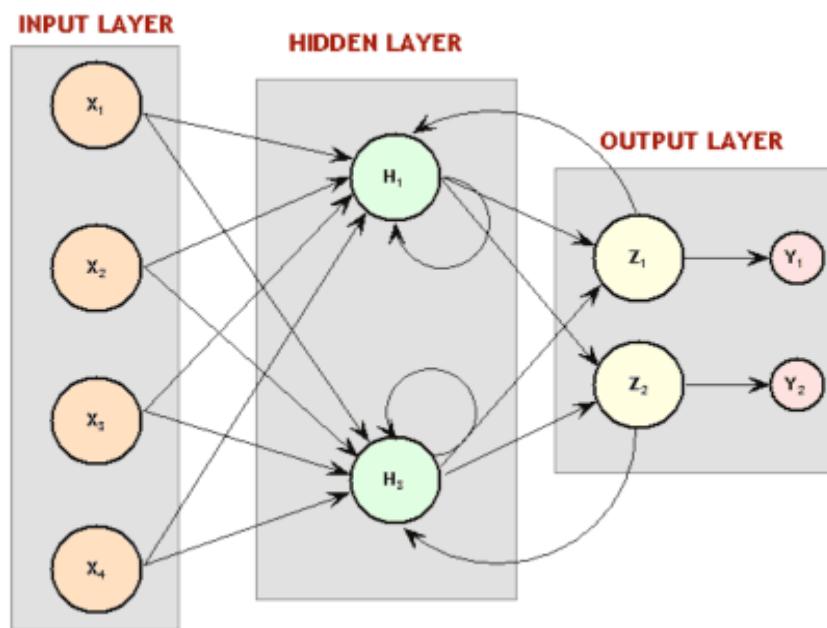
Y = F_W(X)

X date intrare → Y raspuns

Arhitectura RNA

➤ rețele feed-back (bidirectionale/recurente)

cele mai complexe dinamice
pentru fiecare stare de intrare, se caută starea de echilibru



$$X(0)=X \quad \xrightarrow{\text{starea initiala}} \boxed{X(t+1)= F_w(X(t))} \quad \xrightarrow{\text{(sistem dinamic)}} \quad Y=X(T) \quad \xrightarrow{\text{starea stationara}}$$

Caracteristici ale RNA preluate de la creier

➤ capacitatea de a învăța

- învățare din exemple
- ajustarea ponderilor pe baza unor modele
- antrenare cu seturi mari de date

➤ capacitatea de a generaliza

- pot da răspunsuri corecte pentru intrări ușor diferite de cele cu care au fost antrenate

➤ capacitatea de a sintetiza

- pot da răspunsuri corecte pentru intrări afectate de zgomot/imprecise/partiale

RNA - tipuri de arhitecturi în funcție de instruire

➤ Rețele cu învățare supervizată

feed-back: mașina Boltzmann, Learning Vector Quantization (LVQ)

feed-forward: perceptron, Adaline, Madaline

➤ Rețele cu învățare nesupervizată

feed-back: rețele Grossberg, rețele Hopfield (discrete/continue), memorii asociative (bidirectionale/temporale), rețele Kohonen cu auto-organizare

feed-forward: memorii asociative liniare/distribuite, memorii asociative fuzzy

Principii de instruire RNA

- Instruirea RNA
- Tehnicile de instruire ale RNA
- Elemente privind regulile și algoritmii de instruire
- Elemente descriptive pentru reguli/algoritmi de instruire

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.

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

Tehnici de instruire

După momentul de timp în care se face instruirea

- Instruire Offline
(instruire + utilizare)

Exemplu: recunoașterea comenziilor vocale, recunoașterea scrisului de mâna

- Instruire Online
(instruire în timpul funcționării)

Exemplu: compensare ecou, filtre adaptive

Tehnici de instruire

După modul în care se face instruirea se disting următoarele categorii:

- Instruire Supervizată
 - Set de instruire este format din perechi (intrare, ieșire dorită)
 - Epocă de instruire = prezentarea întregului set de instruire
- Instruire cu Întărrire
 - online
 - semnal de întărrire extern (poate fi binar)
- Instruire Nesupervizată
 - online/offline
 - cea mai rapidă
 - clasificarea statistică a intrărilor (distanță euclidiană)

Reguli și algoritmi de instruire

Regulă de instruire = formulă matematică care specifică cum se modifică parametrii RNA, pentru a atinge obiectivul dorit

- regula de corecție a erorii
- regula hebbiană
- regula de instruire competitivă
- regula de tip Boltzmann

Algoritm de instruire = proces iterativ bazat pe o regulă de instruire

Etape:

1. inițializare ponderi
2. calcul ieșire neuroni
3. calcul cantitate Δw cu care se modifică fiecare pondere
4. modificare ponderi
5. salt la etapa 2 – recalcularea ieșirilor neuronilor

Instruire prin minimizarea erorii

unde:

$$E(n) = \frac{1}{M} \sum_{k=1..M} e_k^2(n)$$

n este momentul de timp,

M este numărul de neuroni pe stratul de ieșire,

k este indicele neuronului de pe strat

$e_k(n)$ este eroarea neuronului k la momentul n .

Pentru neuronul k , eroarea la momentul n se calculează după formula:

$$e_k(n) = d_k(n) - y_k(n)$$

unde:

$d_k(n)$ este ieșirea dorită pentru neuronul k , la momentul n

$y_k(n)$ este ieșirea neuronului k calculată de către rețeaua neuronală, la momentul n .

Instruire bazată pe gradient

Regula de instruire

$$\Delta w_k(n) = -\eta \nabla E(n) = -\eta \frac{\partial E(n)}{\partial w_k(n)}$$

η – rată de instruire (valoare subunitară)

Regula Widrow-Hoff (regula Delta)

$$\Delta w_k(n) = -\eta * x_k * e_k$$

Algoritmul bazat pe gradient

1. inițializare ponderi w_{kj} și prag cu valori mici - subunitare
2. calcul ieșire neuroni
3. calcul eroare e_k
4. calcul cantitate Δw_{kj} cu care se modifică fiecare pondere
5. modificare ponderi $w_{kj}(n+1) = w_{kj}(n) + \Delta w_{kj}(n)$
6. salt la etapa 2 – recalcularea ieșirilor neuronilor $n := n + 1$

Instruire hebbiană

Postulatul lui Donald Hebb (1949)

„În cazul în care un axon al unui neuron A transmite impulsuri unui neuron B și în mod repetat și persistent A duce la activarea lui B, atunci apare un proces metabolic, la nivelul uneia sau ambelor celule, astfel încât această tendință de activare a lui B de către A este întărită”.

Dacă doi neuroni sunt activați sincron, ponderea conexiunii dintre ei crește.

- instruire nesupervizată
- rețea "cu memorie"

Instruire hebbiană

Regula de instruire

$$\Delta w_{ij} = \eta * x_j * y_i$$

Algoritmul de instruire hebbiană

1. inițializare ponderi w_{kj} și bias cu valori mici, subunitare
2. calcul ieșire neuronii

$$y_j(n) = \sum_{i=1}^N x_i(n) w_{ij}(n) - \theta_j$$

3. modificare ponderi $w_{kj}(n+1) = w_{kj}(n) + \Delta w_{kj}(n)$
4. salt la etapa 2 – recalcularea ieșirilor neuronilor $n := n + 1$

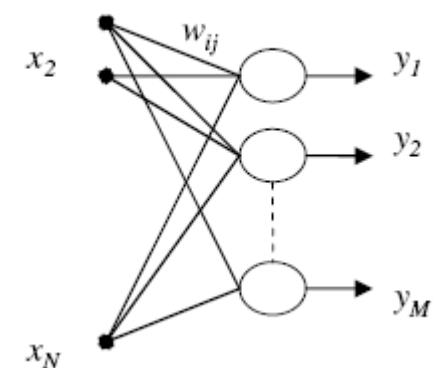
Instruire hebbiană

Avantaje: - simplitatea arhitecturii rețelei
- simplitatea algoritmului de instruire
- rezultate satisfăcătoare dacă intrarea și ieșirea sunt ortogonale

Dezvantaje: - necesitate ortogonalitate între intrare și ieșire
- creștere nelimitată a ponderilor

$$\Delta w_{ij} = \eta * y_j * x_i - \varphi * y_j * w_{ij}$$

φ – factor de uitare, subunitar, tipic între 0.01 și 0.1



Instruire competitivă

- unul sau mai mulți neuroni concurează pentru a fi activați
- "winner takes all" – criteriul activării maxime
- instruire nesupervizată

Exemplu: Self Organizing Maps (Theuvo Kohonen, anii '80)

Instruire competitivă

Regula de instruire

$$\Delta w_{ij} = \begin{cases} \alpha(x_j - w_{ij}), & \text{neuron } j \text{ câștigător} \\ 0 & \end{cases}$$

Dat fiind un vector de intrare x_i , câștigă neuronul al cărui vector de ponderi w este cel mai apropiat (ca distanță euclidiană) față de x_i .

Algoritmul de instruire competitivă

1. inițializare ponderi w_{kj} și prag cu valori mici, subunitare
2. căutare neuron câștigător (distanță euclidiană minimă)
3. calcul și modificare ponderi $w_{kj}(n+1) = w_{kj}(n) + \Delta w_{kj}(n)$
4. salt la etapa 2 – recalcularea ieșirilor neuronilor $n := n + 1$

Instruirea se oprește când nu mai apar modificări semnificative ale ponderilor.

Instruirea Boltzmann

- *simulated annealing (călire simulată)*
- inspirată din termodinamică
- analogie cu procesul restructurării cristaline a metalelor
- nivel cât mai scăzut de energie -> minim al erorii

Se ridică temperatura T a unui metal la o valoare mare. Din această cauză, atomii devin foarte mobili și au mișcari aleatorii, cu atât mai ample cu cât este mai ridicată temperatura T .

Se face o răcire lentă, progresivă, reducând valoarea lui T . Atomii se reorganizează și ajung la stări de energie inferioare energiei lor inițiale.

În final, metalul devine mai dur și energia sa este minimă pentru temperatura dată.

Tipuri de probleme rezolvabile cu RNA

- Tipuri de probleme se pot rezolva cu RNA
- Avantajele RNA
- Dezavantajele RNA

Tipuri de probleme rezolvabile cu RNA

➤ clasificare

- gruparea vectorilor de intrare în clase
- domeniu de ieșire discret

Ex: recunoașterea formelor, operații de telecom (decodare, demodulare, regenerare semnale), decizie

➤ aproximare de funcții

- interpolare perechi intrare-ieșire
- domeniu de ieșire continuu

Ex: modelarea directă/inversă a unui sistem necunoscut, predicție

➤ optimizare

- găsirea punctului de minim/maxim al unei funcții
- set de ponderi pentru care diferența dintre ieșirea dorită și ieșirea rețelei este minimă, în sens statistic

Ex: problema comis-voiajorului

Avantaje ale RNA

- instrumente robuste de calcul
- capabile de generalizare
- pot modela funcții liniare/neliniare
- realizează mapare intrări/ieșiri fără a beneficia de un model cunoscut apriori (free estimation, blind processing)
- adaptabile online/offline
- toleranță la erori/date imprecise
- pot rezolva sarcini complexe

Dezavantaje ale RNA

- problema scalabilității
 - numărul mare de conexiuni reduce viteza de procesare; se preferă implementare hardware
- abordare de tip *black-box*
 - nu pot explica rezultatul furnizat
- instrumente sub-optimale
- pentru rețelele recurente este în cercetare problema cu stabilitatea

Rețele neuronale

- Principii
- Bazele biologice ale RNA
- Instruire
- Arhitectura RNA
- Tipuri de probleme rezolvabile cu RNA