

---

## Unitatea de învățare 8.

### MODELE NEURONALE

---

#### Cuprins

8.1. Modele neuronale.....	
8.2. Arhitecturi de rețele neuronale .....	
8.3. Instruirea rețelelor neuronale.....	



#### Introducere

Încă de la începutul cercetărilor în inteligența artificială s-au conturat două direcții rivale ce constituie și acum modelele dominante:

- *Paradigma logico-simbolică* care presupune mecanisme de reprezentare simbolică a cunoștințelor și utilizarea unor modele logice pentru a deduce noi cunoștințe din faptele memorate în baza de cunoștințe a sistemului.
- *Paradigma conexionistă*, bazată pe calculul neuronal, care a generat realizări concrete cunoscute sub numele de rețele neuronale artificiale (pe scurt rețele neuronale sau rețele neurale).

Un aspect important, care este probabil principalul argument în dezvoltarea cu succes a modelelor conexioniste, respectiv a rețelelor neuronale, este capacitatea acestora de a învăța din exemple. În mod tradițional pentru a rezolva o problemă trebuie să elaborăm mai întâi un model (matematic, logic, lingvistic, etc) al acesteia pe baza căruia se stabilește sub forma unei succesiuni de operații algoritmul de rezolvare a problemei.

Există, însă, probleme practice de mare complexitate pentru care chiar construcția modelului prezintă dificultăți mari și stabilirea unui algoritm, fie el și unul aproximativ, este dificilă sau chiar imposibilă.

Caracteristic rețelelor neuronale este faptul că, pornind de la o mulțime de exemple, ele sunt capabile să sintetizeze în mod explicit un anumit model al problemei. Se poate spune că o rețea neuronală își construiește singură algoritmul pentru rezolvarea unei probleme, dacă are la dispoziție o mulțime reprezentativă de cazuri particulare (exemple de instruire). Așadar rețeaua neuronală extrage de fapt informația prezentă în mulțimea de instruire. Această etapă (fază) de funcționare a unei rețele neuronale se numește de instruire sau antrenare. În faza de lucru – sau

de referință – rețeaua va folosi informația achiziționată în etapa de instruire pentru a trata situații de aceeași natură cu cele conținute în mulțimea de instruire.

Capacitatea rețelei neuronale de a rezolva probleme practice complexe, utilizând o mulțime de exemple de instruire conferă acestora un potențial de aplicabilitate extrem de larg. Rețelele neuronale sunt foarte potrivite pentru a rezolva aceleași tipuri de probleme ca și creierul.

Rețelele neuronale artificiale (RN) reprezintă modele simplificate ale sistemului nervos central. Ele au abilitatea de a răspunde la stimuli de intrare și de a se adapta la mediu. Modelele RN oferă cea mai promițătoare procedură pentru construcția sistemelor de calcul inteligente. Procesarea paralelă și distribuită într-o rețea neuronală artificială este cea mai bună cale pentru implementarea acestor sisteme.



#### Competențele unității de învățare

- *Operarea cu noțiuni fundamentale din domeniul rețelelor neuronale*
- *Cunoașterea principiilor de bază ale algoritmilor de instruire ai rețelelor neuronale*



**Durata medie de parcurgere a unității de învățare este de 2 ore.**

## 8.1. Modele neuronale

Un neuron artificial are  $N$  intrări pe care le vom nota  $x_1, x_2, \dots, x_N$  (numere reale) și o ieșire  $y$ . Intrările sunt conectate la neuron prin intermediul *sinapselor*, care modifică valoarea semnalului de intrare cu un factor  $w_j (\in \mathbb{R})$ , numit *pondere sinaptică*. Prin însumarea semnalelor de intrare ponderate se obține *activarea (excitația) totală* sau *starea internă* a neuronului notată în continuare  $s$ , care se determină cu relația:

$$s = \sum_{j=1}^N w_j x_j. \quad (8.1)$$

În modelul cel mai simplu al neuronului (McCulloch-Pitts), mărimea de ieșire  $y$  se modifică dacă activarea totală depășește o anumită valoare  $P$  numită *prag de excitare (deplasare, eng. bias)*. Astfel avem:

$$y = \begin{cases} 1, & \text{dacă } s \geq P \\ 0, & \text{dacă } s < P \end{cases} \quad (8.2)$$

Variația mărimii de ieșire a unui neuron se poate caracteriza prin intermediul unei funcții  $f: \mathbb{R} \rightarrow \mathbb{R}$ , numită în *funcție de ieșire*, *funcție neuronală* sau *funcție de activare*, definită pentru modelul McCulloch-Pitts conform relației:

$$f(x) = \begin{cases} 1 & \text{pentru } x \geq 0 \\ 0 & \text{pentru } x < 0 \end{cases} \quad (8.3)$$

În acest caz ieșirea neuronului se poate scrie:

$$y = f\left(\sum_{j=1}^N w_j x_j - P\right) \quad (8.4)$$

Valoarea prag  $P$  poate fi eliminată din argumentul funcției  $f$ , dacă se adaugă neuronului un semnal de intrare suplimentar cu valoarea -1 și ponderea  $P$ , adică

$$x_{N+1} = -1, \quad w_{N+1} = P. \quad (8.5a)$$

sau

$$x_0 = -1, \quad w_0 = P \quad (8.5b)$$

În acest caz activarea totală va fi

$$s' = \sum_{j=0}^N w_j x_j, \quad (8.6)$$

iar ieșirea se va scrie acum

$$y = f(s') \text{ cu } f(s') = \begin{cases} 1 & \text{pentru } s' \geq 0 \\ 0 & \text{pentru } s' < 0 \end{cases}. \quad (8.7)$$

Avantajul acestei abordări este acela că pragul poate fi ajustat împreună cu celelalte ponderi în decursul fazei de instruire.

Pentru dezvoltarea algoritmilor de instruire este indicat, ca mărimile de intrare și ponderile sinaptice să fie scrise vectorial. Vom avea în acest caz vectorul mărimilor de intrare:

$$\mathbf{x} = [x_0 \quad x_1 \quad \dots \quad x_N]^T, \quad (8.8)$$

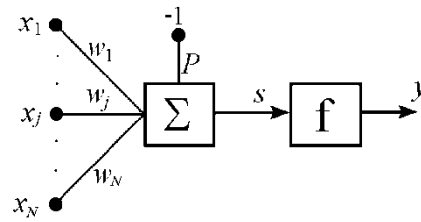
cu  $x_0 = -1$  și vectorul ponderilor sinaptice ale neuronului:

$$\mathbf{w} = [w_0 \quad w_1 \quad \dots \quad w_N]^T. \quad (8.9)$$

Cu aceste notații ieșirea se determină cu relația

$$y = f(\mathbf{w}^T \mathbf{x}) \quad (8.10)$$

Pentru simplificarea expunerii în continuare vom folosi aceeași notație  $s$  pentru activare și în cazul în care valoarea pragului este considerată o pondere a unei mărimi suplimentare de intrare.



**Figura 8.1.** Reprezentarea grafică ale modelului neuronal.

Modelul neuronal este exclusiv determinat de funcția de activare folosită. Funcțiile de activare utilizate cu precădere sunt:

- funcție de activare neliniară cu prag asimetric,

$$f(x) = \begin{cases} 1 & \text{pentru } x \geq 0 \\ 0 & \text{pentru } x < 0 \end{cases}; \quad (8.11)$$

- funcție de activare neliniară cu prag simetric (signum),

$$f(x) = \begin{cases} 1 & \text{pentru } x \geq 0 \\ -1 & \text{pentru } x < 0 \end{cases}; \quad (8.12)$$

- funcție de activare liniară,

$$f(x) = x; \quad (8.13)$$

- funcție de activare liniară cu limitare asimetrică,

$$f(x) = \begin{cases} 0 & \text{pentru } x \leq 0 \\ x & \text{pentru } x \in (0, 1] \\ 1 & \text{pentru } x > 1 \end{cases}; \quad (8.14)$$

- funcție de activare liniară cu limitare simetrică,

$$f(x) = \begin{cases} -1 & \text{pentru } x < -1 \\ x & \text{pentru } x \in [-1, 1] \\ 1 & \text{pentru } x > 1 \end{cases}; \quad (8.15)$$

- funcție de activare sigmoidală,

$$f(x) = \frac{1}{1 + e^{-kx}} \text{ cu } k > 0; \quad (8.16)$$

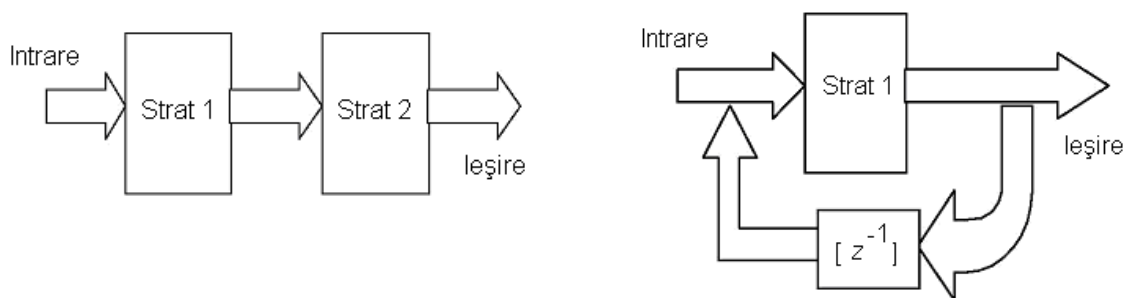
- funcție de activare tangentă hiperbolică

$$f(x) = \frac{e^{kx} - e^{-kx}}{e^{kx} + e^{-kx}} \text{ cu } k > 0. \quad (8.17)$$

## 8.2. Arhitecturi de rețele

Din punct de vedere al arhitecturii, rețelele neuronale se împart în două categorii:

- *Rețele feed-forward*, unde semnalele se transmit în interiorul rețelei într-o singură direcție: de la intrare spre ieșire. Nu există bucle, ieșirea fiecărui neuron neafectând neuronul respectiv. De obicei, sunt utilizate în recunoașterea de imagini;
- *Rețele feed-back*, în care impulsurile se pot transmite în ambele direcții, introducând conexiuni de reacție în arhitectura rețelei. Aceste tipuri de rețele sunt foarte puternice și pot fi extrem de complicate. Sunt dinamice, starea lor schimbându-se permanent, până când rețeaua ajunge la un punct de echilibru. Căutarea unui nou echilibru are loc la fiecare schimbare a intrării. Aceste arhitecturi sunt deseori numite *rețele recurente*.



**Figura 8.2.** Rețele feed-forward și rețele feed-back.

Există câteva structuri de rețele neuronale artificiale consacrate și foarte des utilizate, deoarece au oferit rezultate încurajatoare în diverse aplicații:

- Rețele cu învățare supervizată:
  - feed-forward: perceptron, adaline;
  - feed-back: mașina Boltzmann, Learning Vector Quantization (LVQ);
- Rețele cu învățare nesupervizată:
  - feed-forward: memorii asociative liniare, distribuite, memorii asociative fuzzy;
  - feed-back: rețele Grossberg, rețele Hopfield, memorii asociative (bidirecționale și temporale), rețele Kohonen cu auto-organizare.

Rețelele neuronale cu transmitere înainte (feed-forward) formate dintr-un strat sau mai multe se pot implementa cu funcțiile de activare introduse anterior, respectiv prin utilizarea repetată a acestora.

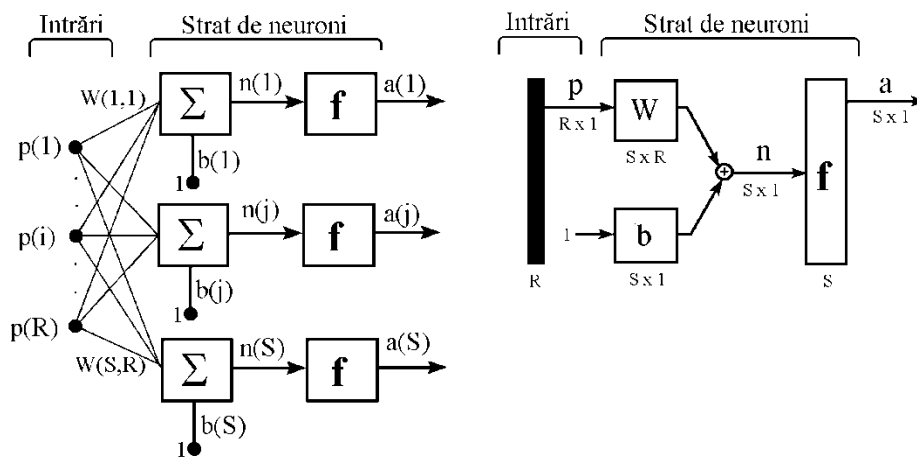
### Rețea neuronală cu un strat de neuroni

Se va considera un strat format din  $S$  neuroni la intrarea căruia se aplică  $R$  mărimi de intrare (fig. 8.3). Pentru modelarea stratului de neuroni se utilizează tot funcția de activare (transfer) introdusă în paragraful anterior. Sintaxa funcției **hardlim** rămâne neschimbată, dar în acest caz  $n$  va fi un vector  $S$ -dimensional.

Matricea ponderilor conexiunilor de la cele  $R$ -intrări la stratul de  $S$  neuroni este de genul:

$$\mathbf{W} = \begin{bmatrix} W(1,1) & W(1,2) & \dots & W(1,R) \\ W(2,1) & W(2,2) & & W(2,R) \\ \vdots & \vdots & \ddots & \vdots \\ W(S,1) & W(S,2) & & W(S,R) \end{bmatrix}.$$

Subliniem faptul că ponderea  $\mathbf{W}(j,i)$  caracterizează conexiunea neuronului  $j$  la intrarea  $i$ .



**Figura 8.3.** Un strat de neuroni. a) Structura; b) Schema bloc de calcul în MATLAB.

Cu aceste notații, dacă se consideră un singur vector de instruire, stratul de neuroni se va implementa cu instrucțiunile:

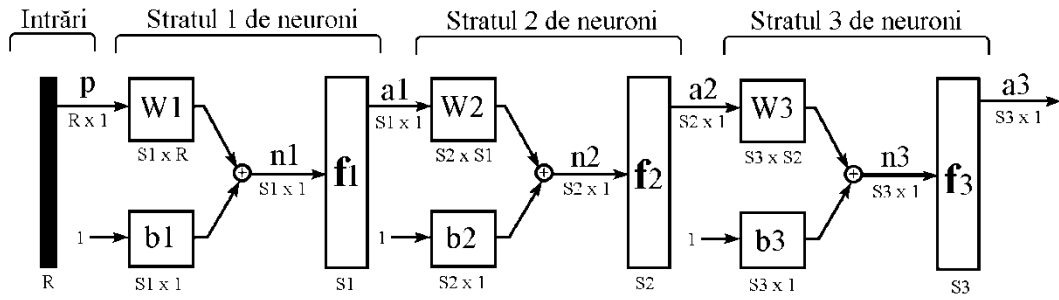
$$y = \text{hardlim}(\mathbf{W} * \mathbf{p} + \mathbf{b})$$

sau

$$y = \text{hardlim}(\mathbf{W} * \mathbf{p}, \mathbf{b}).$$

### Rețea neuronală multistrat cu transmitere înainte

Pentru concretizarea expunerii vom considera o rețea unidirecțională cu trei straturi. Arhitectura rețelei este prezentată în figura 8.4, sub forma unei scheme bloc, care permite calculul în MATLAB.



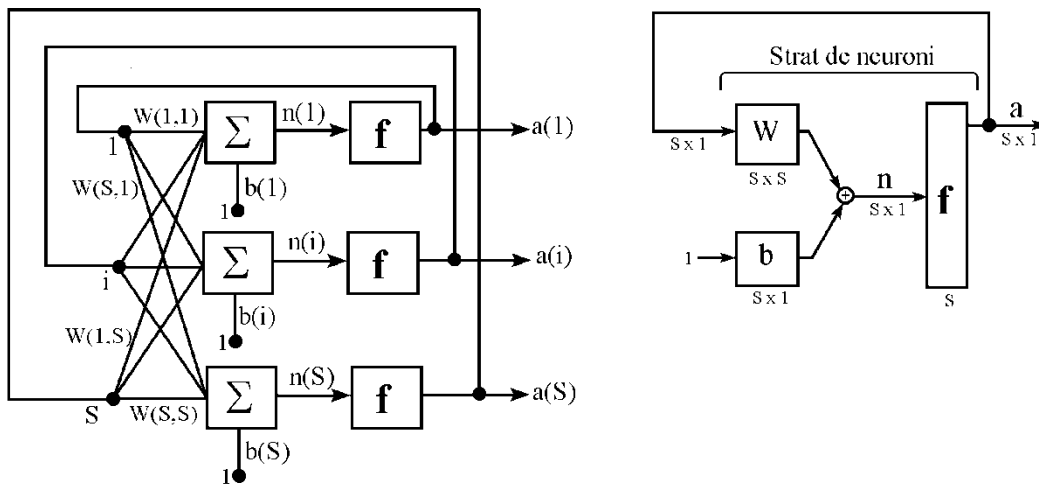
**Figura 8.4.** Schema bloc de calcul în MATLAB pentru o rețea unidirecțională cu trei straturi.

Funcțiile de activare  $f_1$ ,  $f_2$ ,  $f_3$  pot fi de tipuri diferite. Spre exemplificare vom presupune că  $f_2$  și  $f_3$  sunt de tip sigmoidal iar  $f_1$  este funcție liniară. Presupunând că matricile  $W_1$ ,  $W_2$ ,  $W_3$  și vectorii  $b_1$ ,  $b_2$ ,  $b_3$  au fost deja introduse, rețeaua considerată se va modela cu instrucțiunea:

$$a3 = \text{logsig}(W3 * \text{logsig}(W2 * \text{purelin}(W1 * p + b1) + b2) + b3).$$

### Rețeaua Hopfield

Rețeaua Hopfield, cu structura prezentată în fig. 1.5a, se implementează în MATLAB folosind schema bloc de calcul prezentată în fig. 1.5b.



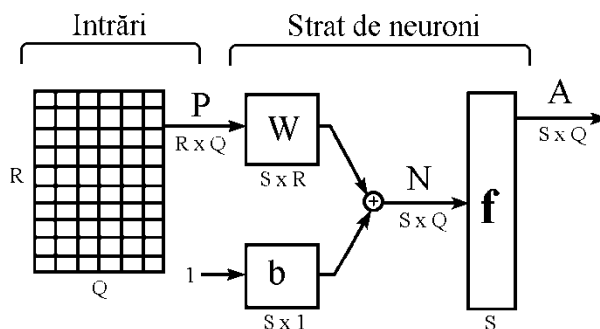
**Figura 8.5.** Arhitectura unei rețele Hopfield. a) Structura; b) Schema bloc de calcul în MATLAB.

Matricea ponderilor  $W$  se introduce în modul prezentat mai înainte, considerând că  $W(j,i)$  este intensitatea conexiunii de la intrarea  $i$  la neuronul  $j$ . Calculul ieșirii se realizează recurent cu instrucțiunea:

$$a = \text{hardlim}(W * a + b).$$

Modele neuronale și arhitecturi de rețele instruite cu o mulțime de  $Q$  vectori de intrare

Modelele prezentate mai înainte s-au introdus considerându-se că, la intrare, se aplică un singur vector  $R$ -dimensional de instruire.



**Figura 8.6.** Schema bloc de calcul în MATLAB pentru un neuron cu  $R$  intrări instruite cu o mulțime de  $Q$  vectori.

Abordăm acum cazul modelării în MATLAB a modelelor neuronale și a arhitecturilor de rețele în care, la intrare, se aplică o mulțime de instruire formată din  $Q$  vectori. Schema bloc de calcul pentru un neuron alimentat la intrare cu o mulțime de  $Q$  vectori  $R$ -dimensionali se prezintă în fig. 1.6. În mod asemănător se pot modela și rețelele cu un strat, multistrat și recurente. Matricea ponderi se definește conform relației prin particularizare în funcție de tipul rețelei. Vectorii de intrare vor fi colectați în matricea  $P$  de dimensiune  $R \times Q$ , fiecare vector constituind o coloană a acestei matrice.

### 8.3. Instruirea rețelelor neuronale

Inteligența artificială în general și calculul neuronal în particular implică două aspecte fundamentale: *învățarea* și *reprezentarea cunoașterii*. Caracteristica esențială a calculului neuronal rezultă din posibilitatea învățării bazate pe informații parțiale sau pe date ce conțin erori (care sunt afectate de zgomote). Rețelele neuronale achiziționează cunoașterea (învăță) prin *instruire*. *O rețea neuronală este instruită dacă la aplicarea unei mulțimi de vectori de intrare va produce ieșirile dorite*. Cunoașterea pe care o dobândește rețeaua neuronală este *memorată* de sinapsele neuronale, mai exact în ponderile conexiunilor dintre mărimile de intrare și neuroni respectiv dintre neuroni. Vectorii de instruire se prezintă rețelei în mod secvențial, iar ponderile rețelei sunt ajustate pentru a capta cunoașterea pe care o reprezintă acești vectori. Ajustarea ponderilor se face în conformitate cu o procedură prestabilită numită *lege de învățare* sau *algoritm de instruire*. De obicei fiecare vector din mulțimea de instruire se prezintă rețelei de mai multe ori. Algoritm de instruire determină convergența ponderilor spre valori care să producă ieșirea dorită pentru toți vectorii de intrare prezentați rețelei. După instruire rețeaua neuronală este utilizată în faza de lucru sau de referință pentru a se valorifica cunoașterea dobândită.

Există două tipuri fundamentale de învățare utilizate în rețelele neuronale: *învățarea supervizată* și *învățarea nesupervizată*.



În cazul instruirii *supervizate* se prezintă rețelei o mulțime de *exemple de instruire*. Un exemplu de instruire este o pereche formată dintr-un vector de intrare și ieșirea dorită. Când se aplică rețelei un vector de intrare se calculează ieșirea rețelei și se compară cu vectorul de ieșire dorit. Diferența între ieșirea dorită și cea obținută reprezintă eroarea. Ponderile rețelei sunt modificate pe baza unui algoritm care tinde să minimizeze această mărime de eroare. Vectorii din mulțimea de instruire sunt aplicați secvențial și ciclic până când prin modificarea ponderilor se atinge o valoare minim acceptabilă a erorii totale.

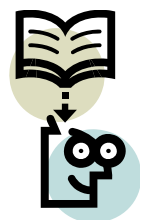
La unele metode de instruire supervizată se modifică ponderile conexiunilor sinaptice numai dacă ieșirea reală diferă de cea dorită. Avem de-a face în acest caz cu un proces de învățare prin corectarea erorilor. Există de asemenea alte metode de învățare supervizată care determină schimbări ale valorilor ponderilor pentru orice vector de intrare.

În ciuda numeroaselor aplicații, instruirea supervizată a fost criticată ca fiind neplauzibilă din punct de vedere biologic. Aceste critici pornesc de la observația că este greu de conceput existența în creier a unui mecanism de instruire care ar compara ieșirile dorite cu cele reale. Se pune de asemenea întrebarea de unde vin formele dorite etc

Instruirea nesupervizată nu necesită cunoașterea a priori a mărimilor de ieșire ale rețelei, sau altfel spus procesul de învățare se desfășoară fără o cunoaștere specifică a ceea ce ar putea să fie un răspuns corect. În decursul instruirii, prin ajustarea ponderilor se realizează o clasificare a vectorilor de intrare în grupe de forme similare sau foarte apropiate (această proprietate poartă denumirea de auto-organizare). Răspunsul pe care îl va produce o clasă de vectori de intrare nu poate fi însă determinat înainte de încheierea procesului de instruire. Prin urmare, ieșirile unei rețele instruite cu o metodă nesupervizată trebuie, în general, să fie interpretate, adică să primească o formă comprehensibilă după încheierea procesului de învățare.

### Să ne reamintim ...

O rețea neuronală constă dintr-o mulțime de elemente de prelucrare (neuroni sau unități cognitive) înalt conectate, fiecare neuron din rețea fiind conectat cu ceilalți neuroni din rețea. Tipul unei rețele neuronale este determinat de modelul neuronal, de arhitectura rețelei și de algoritmul de instruire folosit.



Din punct de vedere al arhitecturii, rețelele neuronale se împart în două categorii:

- *Rețele feed-forward*, unde semnalele se transmit în interiorul rețelei într-o singură direcție: de la intrare spre ieșire. Nu există bucle, ieșirea fiecărui neuron neafectând neuronul respectiv. De obicei, sunt utilizate în recunoașterea de imagini;
- *Rețele feed-back*, în care impulsurile se pot transmite în ambele direcții, introducând conexiuni de reacție în arhitectura rețelei. Aceste tipuri de rețele sunt foarte puternice și pot fi extrem de complicate. Sunt dinamice, starea lor schimbându-se permanent, până când rețeaua ajunge la un punct de echilibru. Căutarea unui nou echilibru are loc la fiecare schimbare a intrării. Aceste arhitecturi sunt deseori numite *rețele recurente*.



### Test de evaluare a cunoștințelor

1. Descrieți modelul neuronului artificial. Enumerați funcțiile de activare uzuale.
2. Explicați diferența dintre învățarea supervizată și învățarea nesupervizată.



### Test de autoevaluare a cunoștințelor

1. În modelul neuronal, activarea totală este:
  - a) suma ponderată a intrărilor
  - b) suma ponderată a intrărilor la care se adună valoarea de prag
  - c) valoarea funcției de activare
2. Funcția de activare:
  - a) calculează valoarea ieșirii pe baza activării totale
  - b) calculează activarea neuronului
  - c) calculează valoarea de prag
3. Între funcțiile de activare uzuale se pot aminti:
  - a) funcția de tip triunghiular
  - b) funcția de tip prag
  - c) funcția liniară
4. Intrările neuronului artificial sunt scalate prin intermediul:
  - a) ponderilor sinaptice
  - b) factorii de scalare
  - c) valorile de prag
5. La rețelele neuronale cu transmitere înainte:

- a) mărimile / semnalele se transmit doar într-o singură direcție, către ieșirile rețelei
- b) neuronii artificiali pot fi organizați pe mai multe straturi
- c) funcțiile de activare sunt calculate înaintea activării totale

6. Instruirea rețelei neuronale se referă la:

- a) ajustarea tuturor ponderilor sinaptice în baza unui algoritm specific
- b) ajustarea funcțiilor de activare pentru determinarea celor optime
- c) ajustarea selectivă a ponderilor sinaptice sau a funcțiilor de activare

7. Un algoritm de instruire este din categoria instruirii supervizate dacă:

- a) datele de instruire conține perechi formate din valori ale semnalelor de intrare și valori corespunzătoare dorite pentru semnalele de ieșire
- b) este necesar ca, pe parcursul execuției implementării software, un supervisor să verifice la fiecare iterație funcționarea algoritmului
- c) funcțiile de activare sunt corespunzător alese în faza de implementare

8. Rețeaua Hopfield este:

- a) o rețea unidirecțională
- b) o rețea bidirecțională
- c) o rețea fără semnale de prag

9. Referitor la algoritmii de instruire:

- a) pentru un anumt tip de rețea neuronală, aceștia completează definirea modelului neuronal
- b) nu au nici o legătură cu modelul neuronal, fiind stabiliți ulterior
- c) definesc modelul neuronal ulterior etepeii de instruire

10. Algoritmii de instruire nesupervizată:

- a) nu necesită date de instruire
- b) nu necesită specificarea valorilor dorite ale semnalelor de ieșire corespunzătoare semnalelor de intrare din datele de antrenare
- c) realizează o clasificare a datelor de instruire