

Rețele neuronale artificiale

-AI notes-

Curs 4: Rețele neuronale artificiale (RNA)

- O structură similară unei rețele neuronale biologice
- O mulțime de noduri (unități, neuroni), dispuse ca într-un graf pe mai multe straturi (layere)

- Unele sarcini pot fi efectuate foarte ușor de către oameni, însă sunt greu de codificat sub forma unor algoritmi

- Recunoașterea formelor
 - vechi prieteni
 - caractere scrise de mână
 - vocea
- Diferite raționamente
 - conducerea autovehiculelor
 - cântatul la pian
 - jucarea baschetului
 - înotul

Definire → Cum învață rețeaua?

- Plecând de la un set de n date de antrenament de forma
$$((x_{p1}, x_{p2}, \dots, x_{pm}, y_{p1}, y_{p2}, \dots, y_{pr}))$$
cu $p = 1, 2, \dots, n$, m – nr atributelor, r – nr ieșirilor
- se formează o RNA cu m noduri de intrare, r noduri de ieșire și o anumită structură internă
 - un anumit nr de nivele ascunse, fiecare nivel cu un anumit nr de neuroni
 - cu legături ponderate între oricare 2 noduri
- se caută valorile optime ale ponderilor între oricare 2 noduri ale rețelei prin minimizarea erorii
 - diferența între rezultatul real y și cel calculat de către rețea

□ Proiectare

■ Construirea RNA pentru rezolvarea unei probleme P

- pp. o problemă de clasificare în care avem un set de date de forma:

- (x^d, t^d) , cu:
- $x^d \in \mathbf{R}^m \rightarrow x^d = (x^d_1, x^d_2, \dots, x^d_m)$
- $t^d \in \mathbf{R}^R \rightarrow t^d = (t^d_1, t^d_2, \dots, t^d_R)$,
- cu $d = 1, 2, \dots, n, n+1, n+2, \dots, N$

- primele n date vor fi folosite drept bază de antrenament a RNA
- ultimele $N-n$ date vor fi folosite drept bază de testare a RNA

- se construiește o RNA astfel:

- stratul de intrare conține exact m noduri (fiecare nod va citi una dintre proprietățile de intrare ale unei instanțe a problemei – $x^d_1, x^d_2, \dots, x^d_m$)
- stratul de ieșire poate conține R noduri (fiecare nod va furniza una dintre proprietățile de ieșire ale unei instanțe a problemei $t^d_1, t^d_2, \dots, t^d_R$)
- unul sau mai multe straturi ascunse cu unul sau mai mulți neuroni pe fiecare strat

□ Inițializarea parametrilor RNA

- Inițializarea ponderile între oricare 2 noduri de pe straturi diferite
- Stabilirea funcției de activare corespunzătoare fiecărui neuron (de pe straturile ascunse)

■ Antrenarea (învățarea) RNA

- Scop:
 - stabilirea valorii optime a ponderilor dintre 2 noduri
- Algoritm
 - Se caută valorile optime ale ponderilor între oricare 2 noduri ale rețelei prin minimizarea erorii (diferența între rezultatul real y și cel calculat de către rețea)
- Cum învață rețeaua?
 - Rețeaua = mulțime de unități primitive de calcul interconectate între ele →
 - Învățarea rețelei = \cup învățarea unităților primitive
 - Unități primitive de calcul
 - Perceptron
 - Unitate liniară
 - Unitate sigmoidală

Funcții:

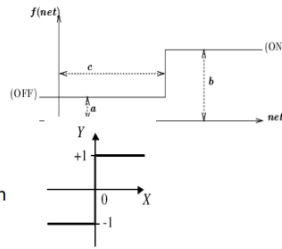
- constantă
- prag
- rampă
- liniară
- sigmoidală
- Gaussiană
- ReLu

■ Funcția de activare a unui neuron

- Funcția constantă $f(net) = \text{const}$
- Funcția prag (c - pragul)

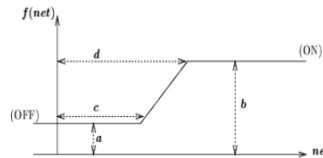
$$f(net) = \begin{cases} a, & \text{dacă } net < c \\ b, & \text{dacă } net > c \end{cases}$$

- Pentru $a=+1$, $b=-1$ și $c=0 \rightarrow$ funcția semn
- Funcție discontinuă



□ Funcția rampă

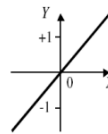
$$f(net) = \begin{cases} a, & \text{dacă } net \leq c \\ b, & \text{dacă } net \geq d \\ a + \frac{(net-c)(b-a)}{d-c}, & \text{altfel} \end{cases}$$



□ Funcția liniară

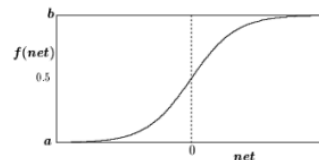
$$f(net) = a \cdot net + b$$

- Pentru $a=1$ și $b=0 \rightarrow$ funcția identitate $f(net)=net$
- Funcție continuă



□ Funcția sigmoidală

- În formă de S
- Continuă și diferențiabilă în orice punct
- Simetrică rotațional față de un anumit punct ($net = c$)
- Atinge asimptotic puncte de saturație



$$\lim_{net \rightarrow -\infty} f(net) = a \quad \lim_{net \rightarrow \infty} f(net) = b$$

- Exemple de funcții sigmoidale:

$$f(net) = z + \frac{1}{1 + \exp(-x \cdot net + y)}$$

$$f(net) = \tanh(x \cdot net - y) + z$$

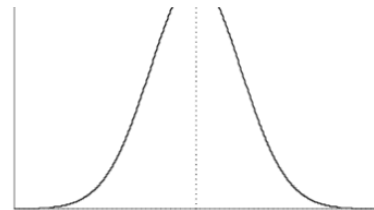
$$\text{unde } \tanh(u) = \frac{e^u - e^{-u}}{e^u + e^{-u}}$$

- Pentru $y=0$ și $z=0 \rightarrow a=0$, $b=1$, $c=0$
- Pentru $y=0$ și $z=-0.5 \rightarrow a=-0.5$, $b=0.5$, $c=0$
- Cu cât x este mai mare, cu atât curba este mai abruptă

□ Funcția Gaussiană

- În formă de clopot
- Continuă
- Atinge asimptotic un punct de saturație

$$\lim_{net \rightarrow \infty} f(net) = a$$



- Are un singur punct de optim (maxim) – atins când $net = \mu$
- Exemplu

$$f(net) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{1}{2}\left(\frac{net-\mu}{\sigma}\right)^2\right]$$

□ Funcția ReLU

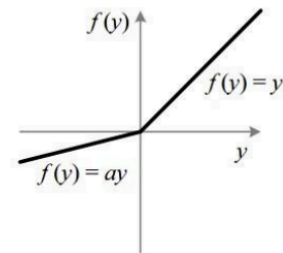
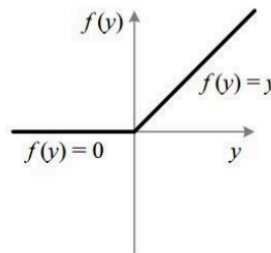
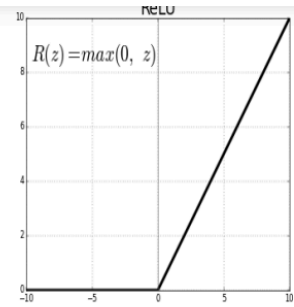
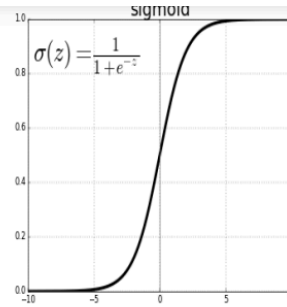
- În formă de rampă
- Continuă, monotonă
- Derivata ei este monotonă
- Codomeniu pozitiv $[0, \infty)$

$$f(\text{net}) = \max(0, \text{net})$$

$$f(\text{net}) = \begin{cases} 0, & \text{dacă } \text{net} < 0 \\ \text{net}, & \text{dacă } \text{net} \geq 0 \end{cases}$$

- Variantă: Leaky ReLU
 - Compensează problemele cu argumentele negative din ReLU

$$f(\text{net}) = \begin{cases} a \cdot \text{net}, & \text{dacă } \text{net} < 0 \\ \text{net}, & \text{dacă } \text{net} \geq 0 \end{cases}$$



Învățarea neuronului:

- Regula perceptronului -> algoritmul perceptronului

1. Se porneste cu un set de ponderi oarecare
2. Se stabilește calitatea modelului creat pe baza acestor ponderi pentru **UNA** dintre datele de intrare
3. Se ajustează ponderile în funcție de calitatea modelului
4. Se reia algoritmul de la pasul 2 până când se ajunge la calitate maximă

- Regula Delta -> algoritmul scăderii după gradient

1. Se porneste cu un set de ponderi oarecare
2. Se stabilește calitatea modelului creat pe baza acestor ponderi pentru **TOATE** dintre datele de intrare
3. Se ajustează ponderile în funcție de calitatea modelului
4. Se reia algoritmul de la pasul 2 până când se ajunge la calitate maximă

● RNA feed-forward:

- Informația se procesează și circulă de pe un strat pe altul
- Conexiunile între noduri nu formează cicluri
- Se folosesc, în special, pentru învățarea supervizată
- Funcțiile de activare a nodurilor -> liniare, sigmoide, gaussiene

● RNA recurente (cu feedback):

- Pot conține conexiuni între noduri de pe același strat

- Conexiunile între noduri pot forma cicluri
- RNA de tip Jordan
- RNA de tip Elman
- RNA de tip Hopfield
- RNA auto-organizate -> pentru învățarea nesupervizată
 - De tip Hebbian
 - De tip Kohonen (Self organised maps)