

5. Rețele neuronale. Reprezentare

Definiții
Reprezentarea modelului
Aplicații

În acest curs vom construi o rețea neurală artificială, urmând ca în cursul următor să o antrenăm folosind metoda de minimizare stochastică a gradientului și metoda propagării înapoi a erorii.

Rețelele neuronale artificiale sunt algoritmi moderni utilizați în clasificarea datelor. Acestea sunt folosite în multe domenii, dintre care remarcăm conducerea autonomă a autovehiculelor, unde rețele neuronale sunt folosite pentru percepția mediului înconjurător, în recunoașterea comenzilor vocale (de exemplu, algoritmi de inteligență artificială care pot înțelege comenzi vocale, precum cei prezenți pe anumite telefoane de ultimă generație), în procesarea de imagini (algoritmi de detecție a fețelor), etc.

5.1 Ipoteze $h_{\theta}(x)$ neliniare

Vom începe prin a reprezenta o rețea neuronală ca și o funcție ipoteză, având un număr ridicat de parametri:

$$h_{\theta}(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1 x_2 + \theta_4 x_1^2 x_2 + \theta_5 x_1^3 x_2 + \dots), \quad (5.1)$$

unde $g(\cdot)$ este funcția sigmoid.

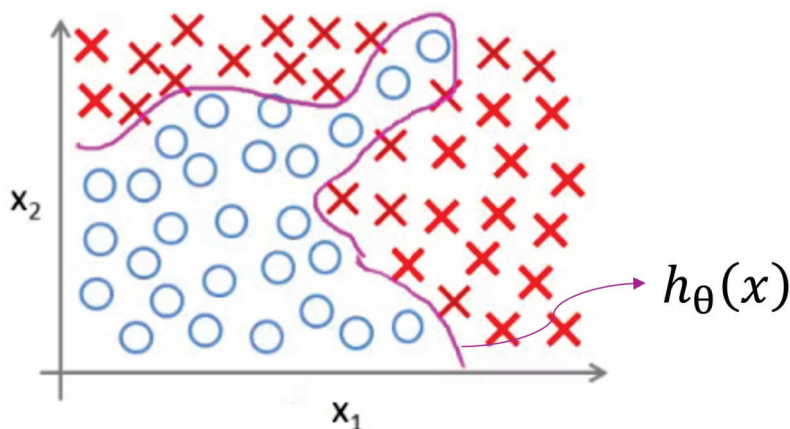


Fig. 5.1 Ipoteza unui aproximator matematic universal.

Figura 5.1 ilustrează forma unei ipoteze neliniare complexe aplicată pe un spațiu de caracteristici (features space) de 2 dimensiuni $x \in \mathbb{R}^2$. Chiar dacă o astfel de ipoteză poate fi construită relativ ușor pe un număr redus de caracteristici, vom întâlni probleme de overfitting atunci când numărul de caracteristici este mare. În Figura 5.2 este redat numărul de caracteristici pentru problema de recunoaștere a obiectelor, în cazul acesta de detecție a unei mașini.

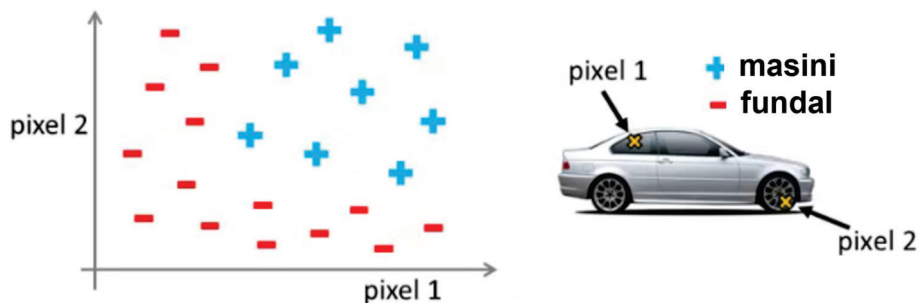


Fig. 5.2 Numărul de caracteristici de intrare în problema de recunoaștere a obiectelor este egal cu numărul de pixeli din imagine.

Numărul de caracteristici este în acest caz mult mai mare. De exemplu, pentru o imagine relativ mică de $50px \times 50px$, avem un număr de $n = 2500$ de caracteristici:

$$x = \begin{bmatrix} \text{intensitatea pixelului 1} \\ \text{intensitatea pixelului 2} \\ \vdots \\ \text{intensitatea pixelului 2500} \end{bmatrix} \quad (5.2)$$

5.2 Structura unui neuron

Structura biologică a unui neuron este redată în Figura 5.3. Semnalele de intrare sunt preluate de la dendrite, urmând ca celula să calculeze valoare de activare a neuronului ce va fi transmisă pe axon. La randul lor, terminațiile axonului vor fi conectate la dendritele altor neuroni.

Simplificând, putem spune că neuronii sunt unități computaționale ce au la intrare semnale, ce sunt ulterior procesate și trimise către ieșire. În modelul nostru, intrările sunt caracteristici de forma $x_1 \dots x_n$, iar ieșirile sunt rezultatul dat de o funcție de activare. Pentru funcția de activare se va folosi aceeași funcție logistică ca și în problemele de clasificare, denumită și *funcția de activare sigmoid*: $1/(1+e^{-\theta x})$. În acest caz, θ poartă denumirea de

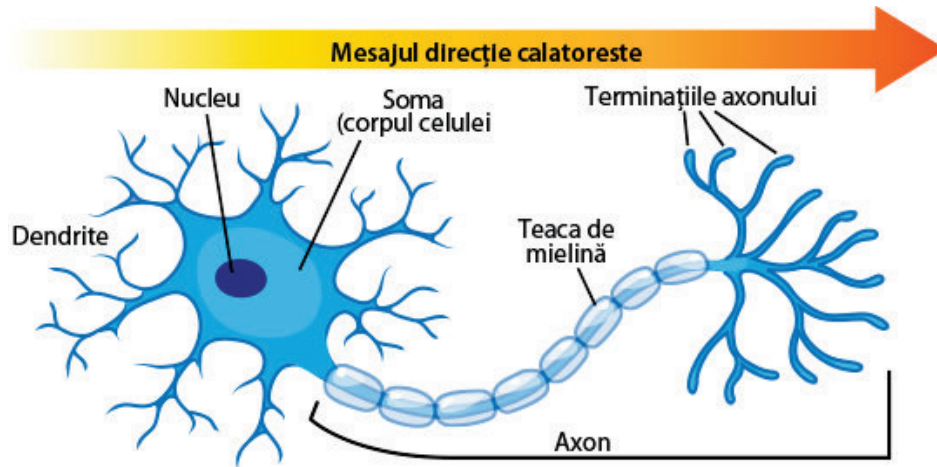


Fig. 5.3 Structura biologică a unui neuron.

ponderi (eng. weights). O reprezentare simplistă poate arăta în felul următor:

$$\begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix} \rightarrow \left[\quad \right] \rightarrow h_{\theta}(x). \quad (5.3)$$

Modelul simplificat al unui neuron este prezentat în Figura 5.4. Se poate observa că modelul unui neuron artificial aproximează funcția de regresie logistică. Valorile de intrare $x = [x_0, x_1, x_2, x_3]$ sunt multiplicare cu parametrii $\theta = [\theta_0, \theta_1, \theta_2, \theta_3]$, formând astfel polinomul $\theta^T x$ al ipotezei descrise în capitolul de regresie liniară. Ieșirea neuronului din Figura 5.4 este dată de funcția sigmoid $g(\cdot)$, aplicată lui $\theta^T x$.

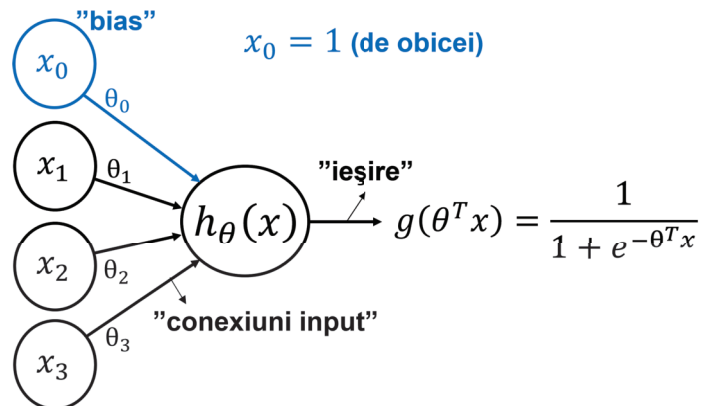


Fig. 5.4 Aproximarea matematică a unui neuron.

5.3 Structura unui rețele neuronale pentru clasificarea binară

O rețea neuronală reprezintă o mulțime de neuroni structurați în așa fel încât fiecare neuron să contribuie la valoarea finală de ieșire a rețelei. În continuare, vom denumi un singur neuron ca și nod unic într-o rețea neuronală.

Nodurile de intrare, care poartă denumirea de **strat de intrare**, sunt legate de alte noduri, ce returnează la ieșire funcția ipoteză și poartă denumirea de **strat de ieșire**. Pot exista și straturi intermediare de neuroni, denumite **straturi ascunse**. Straturile intermediare, denumite **unități de activare**, se vor nota $a_0^2 \dots a_n^2$. Dacă includem și un strat ascuns în rețea, ecuația 5.3 devine:

$$\begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} \rightarrow \begin{bmatrix} a_1^{(2)} \\ a_2^{(2)} \\ a_3^{(2)} \end{bmatrix} \rightarrow h_{\theta}(x). \quad (5.4)$$

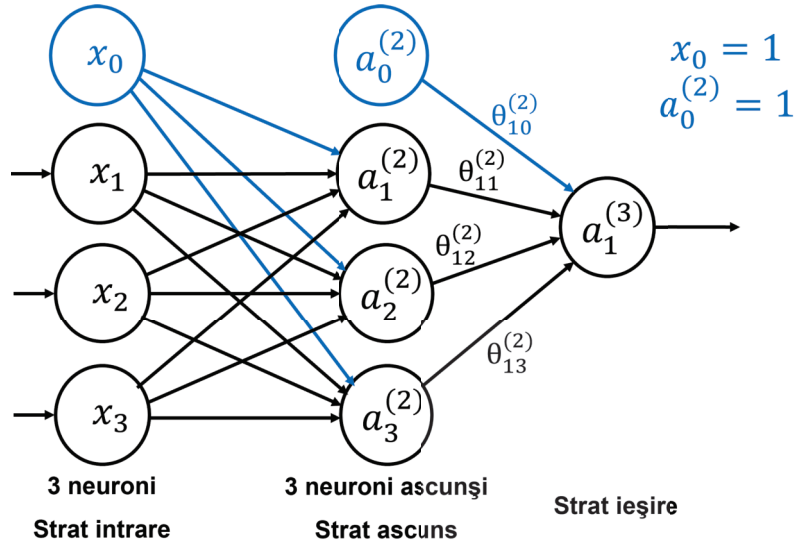


Fig. 5.5 Structura unei rețele neuronale cu un singur strat ascuns.

Structura unei astfel de rețele neuronale este redată în Figura 5.5. Notății:

- $a_i^{(j)}$ - "activarea" neuronului i din stratul j ;
- $\theta^{(j)}$ - matricea ponderilor ce controlează funcția de mapare din stratul j în stratul $j+1$.

Valorile fiecărui nod activare se obțin în felul următor:

$$a_1^{(2)} = g(\theta_{10}^{(1)}x_0 + \theta_{11}^{(1)}x_1 + \theta_{12}^{(1)}x_2 + \theta_{13}^{(1)}x_3), \quad (5.5)$$

$$a_2^{(2)} = g(\theta_{20}^{(1)}x_0 + \theta_{21}^{(1)}x_1 + \theta_{22}^{(1)}x_2 + \theta_{23}^{(1)}x_3), \quad (5.6)$$

$$a_3^{(2)} = g(\theta_{30}^{(1)}x_0 + \theta_{31}^{(1)}x_1 + \theta_{32}^{(1)}x_2 + \theta_{33}^{(1)}x_3), \quad (5.7)$$

$$h_{\theta}(x) = a_1^{(3)} = g(\theta_{10}^{(2)}a_0^{(2)} + \theta_{11}^{(2)}a_1^{(2)} + \theta_{12}^{(2)}a_2^{(2)} + \theta_{13}^{(2)}a_3^{(2)}), \quad (5.8)$$

$$\theta^{(1)} \in \mathfrak{R}^{3 \times 4} \quad (5.9)$$

Așadar, putem calcula activările din noduri folosind o matrice 3×4 , denumită *matrice de ponderi*. Fiecare strat de neuroni are propria matrice de ponderi.

Dacă rețeaua neuronală are s_j neuroni în stratul j și s_{j+1} neuroni în stratul $j + 1$, atunci Θ^j va avea dimensiunea:

$$s_{j+1} \times (s_j + 1). \quad (5.10)$$

Variind parametrii Θ ai funcției $y = h_\theta(x)$, care în acest caz reprezintă ieșirea rețelei neuronale, obținem diferite funcții de mapare între variabilele de intrare x și ieșirea y .

5.4 Reprezentarea vectorială

În continuare, vom realiza o implementare vectorizată a funcției 5.8. Se va defini o nouă variabilă $z_k^{(j)}$ care cuprinde parametrii din funcția de activare. Dacă înlocuim în ecuația 5.8 parametrii cu variabila z , vom obține:

$$\begin{aligned} z^{(2)} &= \Theta^{(1)}x \\ a_1^{(2)} &= 1 \\ a_1^{(2)} &= g(z_1^{(2)}) \\ a_2^{(2)} &= g(z_2^{(2)}) \\ a_3^{(2)} &= g(z_3^{(2)}) \\ z^{(3)} &= \Theta^{(2)}a^{(3)} \\ h_\Theta(x) &= a^{(3)} = g(z^{(3)}) \end{aligned} \quad (5.11)$$

Matricile de ponderi ale rețelei neuronale din Figura 5.5 sunt evidențiate în Figura 5.6. Pentru stratul $j = 2$ și nodul k , variabila z va fi:

$$z_k^{(2)} = \theta_{k,0}^{(1)}x_0 + \theta_{k,1}^{(1)}x_1 + \cdots + \theta_{k,n}^{(1)}x_n. \quad (5.12)$$

Reprezentarea vectorizată ale lui x și $z^{(j)}$ este:

$$x = \begin{bmatrix} x_0 \\ x_1 \\ \cdots \\ x_n \end{bmatrix} \quad z^{(j)} = \begin{bmatrix} z_1^{(j)} \\ z_2^{(j)} \\ \cdots \\ z_n^{(j)} \end{bmatrix}.$$

$$\begin{aligned}
 a_1^{(2)} &= g \left(\theta_{10}^{(1)} x_0 + \theta_{11}^{(1)} x_1 + \theta_{12}^{(1)} x_2 + \theta_{13}^{(1)} x_3 \right) \\
 a_2^{(2)} &= g \left(\theta_{20}^{(1)} x_0 + \theta_{21}^{(1)} x_1 + \theta_{22}^{(1)} x_2 + \theta_{23}^{(1)} x_3 \right) \\
 a_3^{(2)} &= g \left(\theta_{30}^{(1)} x_0 + \theta_{31}^{(1)} x_1 + \theta_{32}^{(1)} x_2 + \theta_{33}^{(1)} x_3 \right) \\
 h_{\theta}(x) &= a_1^{(3)} = g \left(\theta_{10}^{(2)} a_0^{(2)} + \theta_{11}^{(2)} a_1^{(2)} + \theta_{12}^{(2)} a_2^{(2)} + \theta_{13}^{(2)} a_3^{(2)} \right)
 \end{aligned}$$

Fig. 5.6 Matricile de ponderi pentru rețeaua neuronală din Figura 5.5.

Setând $x = a^{(1)}$, vom obține: $z^j = \Theta^{(j-1)} \cdot a^{j-1}$.

Pentru a calcula ipoteza finală, trebuie mai întâi să calculăm un alt vector z : $z^{(j+1)} = \Theta^{(j)} a^{(j)}$. Obținem vectorul final z multiplicând matricea Θ cu valorile activărilor nodurilor:

$$h_{\theta}(x) = a^{(j+1)} = g(z^{(j+1)}). \quad (5.13)$$

După cum se poate vedea din Figura 5.5, în locul utilizării caracteristicilor originale (x_1, x_2, x_3) , o rețea neuronală folosește noile caracteristici învățate $(a_1^{(2)}, a_2^{(2)}, a_3^{(2)})$.

5.5 Clasificarea multclasă

Spre deosebire de clasificarea binară, unde metoda de predicție furnizează doar valorile 0 și 1, un sistem de clasificare multclasă dispune de un număr de minim 3 ieșiri. Figura 5.7 ilustrează o rețea neuronală ce prezice patru clase de obiecte, și anume pieton, mașină, motocicletă și camion. Astfel, $h_{\theta}(x) \in \mathbb{R}^4$.

Într-un sistem de clasificare multclasă, valoarea de ieșire este un vector, unde fiecare element reprezintă probabilitatea de apariție a fiecărei clase. În cazul ideal, pentru cele patru clase considerate mai sus, vectorul de ieșire al rețelei neuronale este descris în Figura 5.8.

Clasa $y^{(i)}$ a datelor de antrenare $\{(x^{(1)}, y^{(1)}), \dots, (x^{(m)}, y^{(m)})\}$ are una din formele vectorilor din Figura 5.8, în cazul ideal rețeaua aproximând vectorii indicați în datele de intrare:

$$h_{\theta}(x^{(i)}) \sim y^{(i)} \quad (5.14)$$

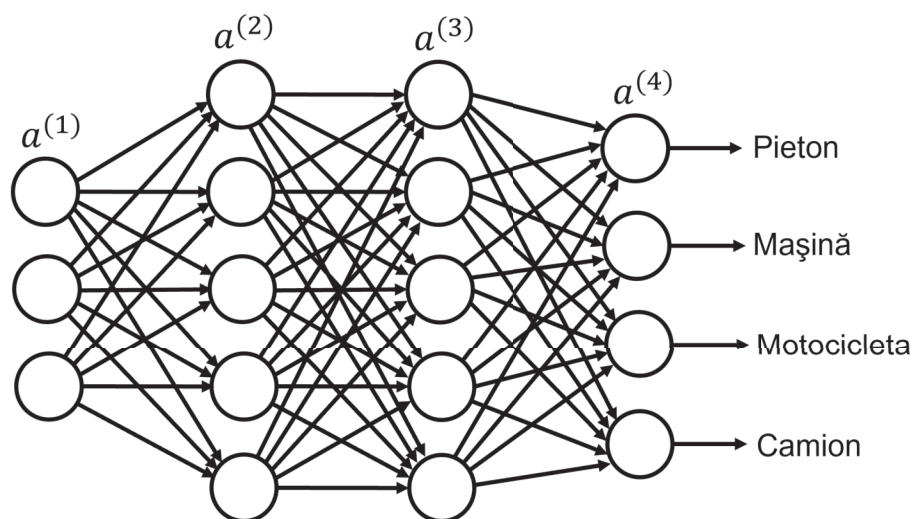


Fig. 5.7 Exemplu de rețea neuronală utilizată în clasificarea multclasă.

$$\begin{array}{cccc}
 h_{\theta}(x) \sim \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} & h_{\theta}(x) \sim \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} & h_{\theta}(x) \sim \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} & h_{\theta}(x) \sim \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \\
 \text{Pieton} & \text{Mașină} & \text{Motocicleta} & \text{Camion}
 \end{array}$$

Fig. 5.8 Vectorul de ieșire pentru cazul ideal al rețelei neuronale din Figura 5.7.