6. Reţele neurale. Reprezentare

Definiţii
Reprezentarea modelului
Aplicații

În acestă lucrare de laborator vom construi o rețea neurală artificială, urmând ca în lucrarea următorare să o antrenăm folosind metoda de minimizare stochastică a gradientului și metoda propagării înapoi a erorii [1].

Rețelele neurale artificiale sunt algoritmi moderni utilizaţi în clasificarea datelor. Acestea sunt folosite în multe domenii, dintre care remarcăm domeniul medical, unde rețele neurale sunt folosite pentru a depista anumite afecțiuni, în recunoașterea comenzilor vocale (de exemplu, algoritmi de inteligență artificială care pot înţelege comenzi vocale, precum cei prezenți pe anumite telefoane de ultimă generație), în procesarea de imagini (algoritmi de detecție a fețelor), etc.

6.1 Reprezentarea modelului

Vom începe prin a reprezenta o funcție ipoteză folosind rețele neurale. 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 funcţia de activare se va folosi aceeași funcție logistică ca şi în problemele de clasificare, denumită şi *funcția de activare sigmoid*: ${}^{1}/_{1+e^{-\theta \cdot T_x}}$. În acest caz, θ poartă denumirea de *ponderi* (eng. weights). O reprezentare simplistă poate arăta în felul următor:

$$\begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix} \rightarrow \begin{bmatrix} \ \end{bmatrix} \rightarrow h_\theta(x). \tag{6.1}$$

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 şi un strat ascuns în reţea, ecuația 6.1 devine:

$$\begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} \rightarrow \begin{bmatrix} a_1^{(2)} \\ a_2^{(2)} \\ a_3^{(2)} \end{bmatrix} \rightarrow h_\theta(x). \tag{6.2}$$

Structura unei astfel de reţele neurale este redată în figura 6.1. Valorile fiecărui nod activare se obțin în felul următor:

$$a_1^{(2)} = g(\Theta_{10}^{(1)} x_0 + \Theta_{11}^{(1)} x_1 + \Theta_{12}^{(1)} x_2 + \Theta_{13}^{(1)} x_3), \tag{6.3}$$

$$a_2^{(2)} = g(\Theta_{20}^{(1)} x_0 + \Theta_{21}^{(1)} x_1 + \Theta_{22}^{(1)} x_2 + \Theta_{23}^{(1)} x_3), \tag{6.4}$$

$$a_3^{(2)} = g(\Theta_{30}^{(1)} x_0 + \Theta_{31}^{(1)} x_1 + \Theta_{32}^{(1)} x_2 + \Theta_{33}^{(1)} x_3), \tag{6.5}$$

$$h_\theta(x) = a_1^{(3)} = g(\Theta_{10}^{(2)} a_0^{(2)} + \Theta_{11}^{(2)} a_1^{(2)} + \Theta_{12}^{(2)} a_2^{(2)} + \Theta_{13}^{(2)} a_3^{(2)}). \tag{6.6}$$

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 are s_j neuroni în stratul j și s_{j+1} neuroni în stratul $j+1$, atunci Θ^j va avea dimensiunea $s_{j+1} \times (s_j + 1)$.

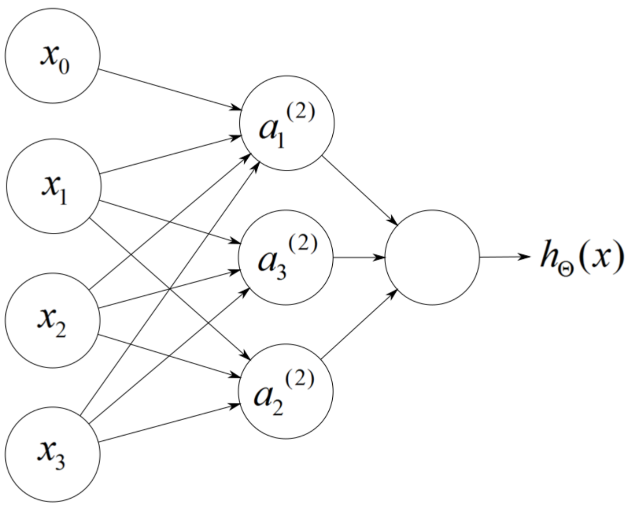

Fig. 6.1 Structura unei reţele neurale cu un singur strat ascuns.

În continuare, vom realiza o implementare vectorizată a funcției 6.6. Se va defini o nouă variabilă $z_k^{(j)}$ care cuprinde parametrii din funcția de activare. Dacă înlocuim în ecuația 6.6 parametrii cu variabila z, vom obține:

$$a_1^{(2)} = g(z_1^{(2)})a_2^{(2)} = g(z_2^{(2)})a_3^{(2)} = g(z_3^{(2)}).$$

Pentru stratul $j = 2$ și nodul k, variabila z va fi:

$$z_k^{(2)} = \Theta_{k,0}^{(1)}x_0 + \Theta_{k,1}^{(1)}x_1 + \cdots + \Theta_{k,n}^{(1)}x_n. \tag{6.7}$$

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

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

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:

$$h_\Theta(x) = a^{(j+1)} = g(z^{(j+1)}). \tag{6.8}$$

6.2 Cerinţe

Având acum cunoștințe despre straturile şi conexiunile dintre neuroni, cât și activările acestora, putem construi o rețea neurală ce poate simula o funcţie logică. Folosind rețeaua, se va simula următorul circuit ce realizează suma a doi biți, folosind porți logice ȘI-NU.

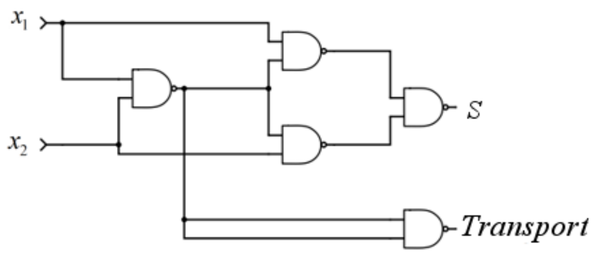

Fig. 6.2 Circuit de însumare biți.

Circuitul din figura 6.2 primește la intrare valorile a doi biți, x_1 și x_2, livrând la ieșire suma acestora. Comportamentul acestui circuit pentru toate valorile biților de intrare este prezentat în următorul tabel de adevăr:

x_1	x_2	S	Transport
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

În continuare, vom proiecta o rețea neurală având structura din figura 6.3:

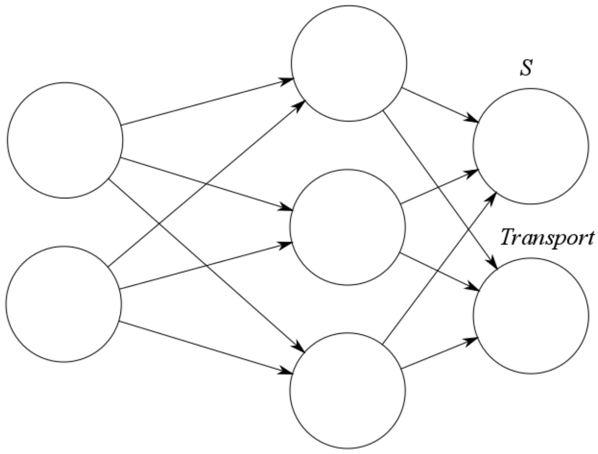

Fig. 6.3 Rețea neurală destinată simulării circuitul de însumare a biților.

În funcția `__init__` a clasei NeuronLayer, se vor inițializa activările stratului, vectorul bias și ponderile. Avem nevoie să știm dacă stratul curent este de intrare sau de ieșire, iar acest lucru este dat de către parametrii is_input și is_output. Funcția `feedforward` este cea care calculează ieșirea stratului de neuroni bazat pe ecuaţia 6.8. Clasa NeuralNetwork conține o listă de obiecte de tip NeuronLayer și are rolul de a conecta mai multe straturi de neuroni și de a apela funcția `feedforward` a acestora, pentru a calcula ieșirile rețelei neurale.

Bibliografie

[1] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 2nd Ed. Pearson Education, 2003.