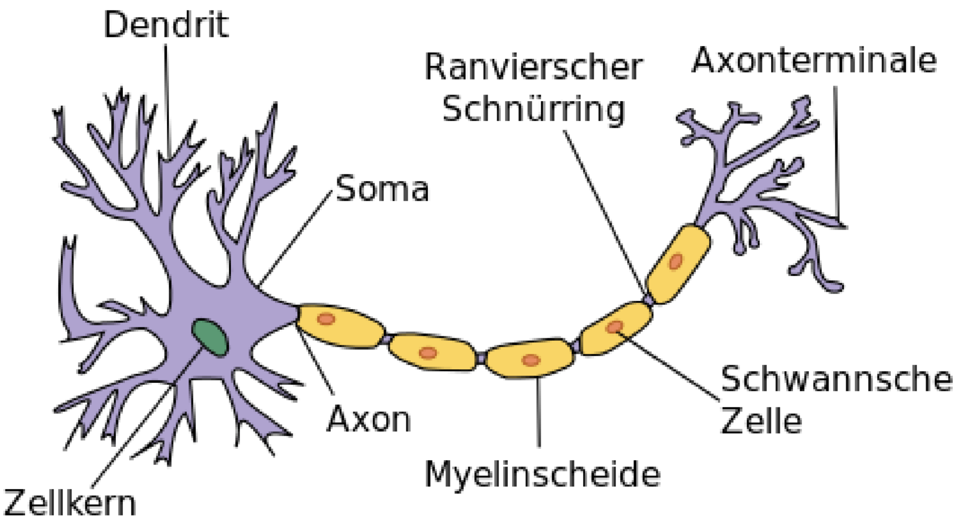
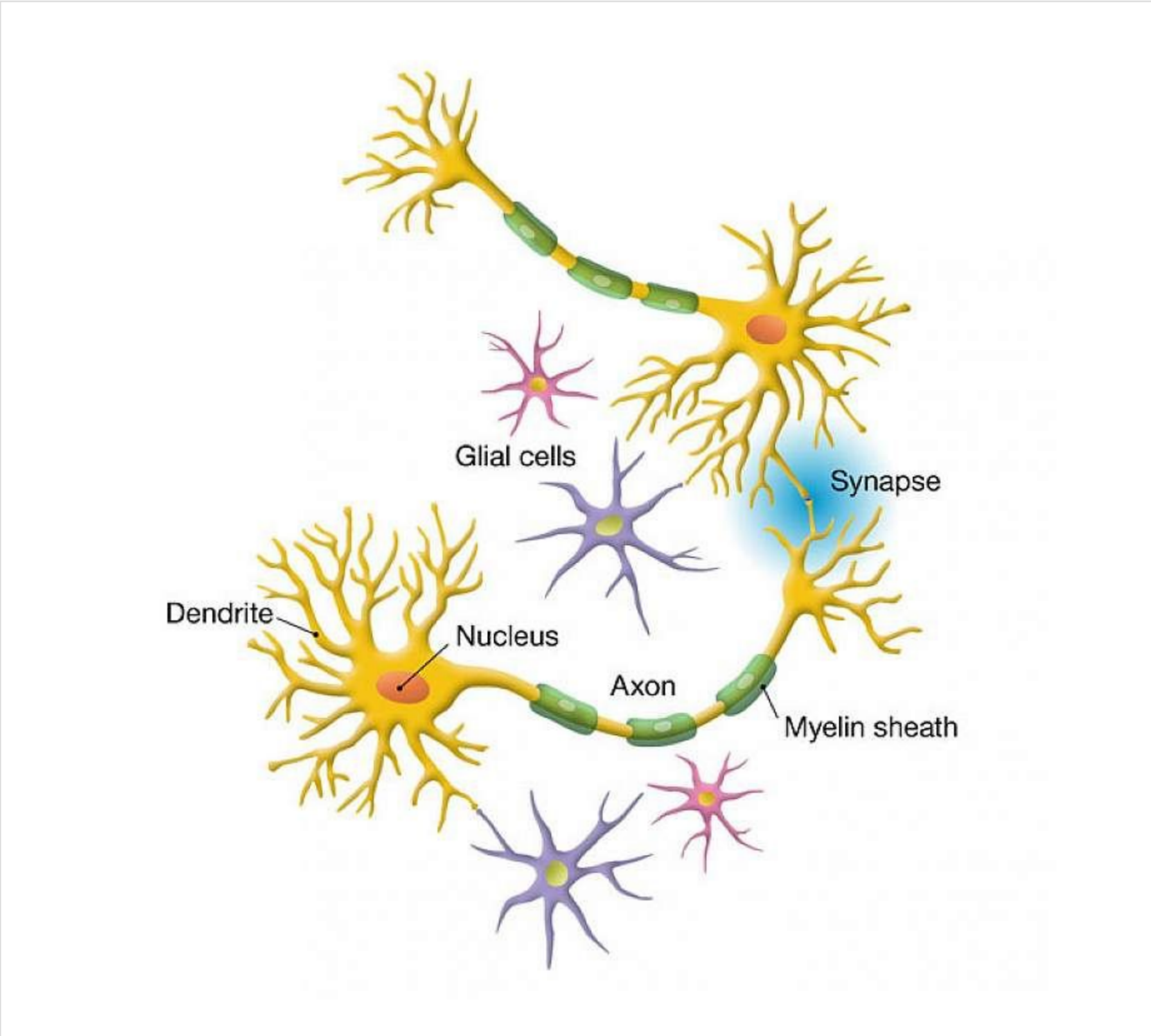
Neural Networks - retele neuronale

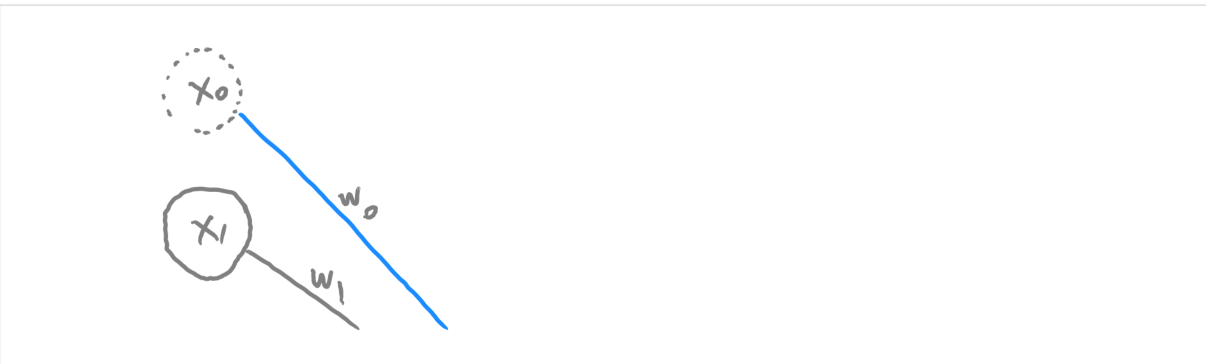
Retelele neuronale in invatarea automata sunt inspirate de structura neuronilor. Sa ne amintim cum arata un neuron.

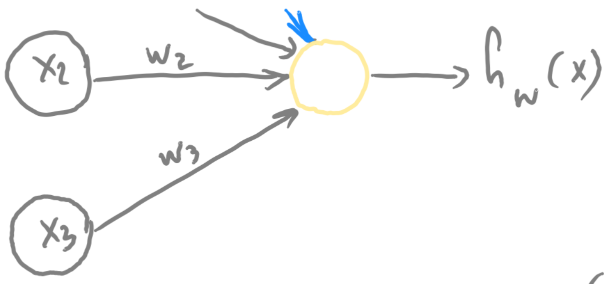


Apropo, comunicarea intre neuroni are loc pe baza de impuls electronic.



Intr-o retea neuronala artificiala noi vom face ceva foarte similar cu ceea ce face o retea neuronala adevarata, insa un model mult mai simplificat.



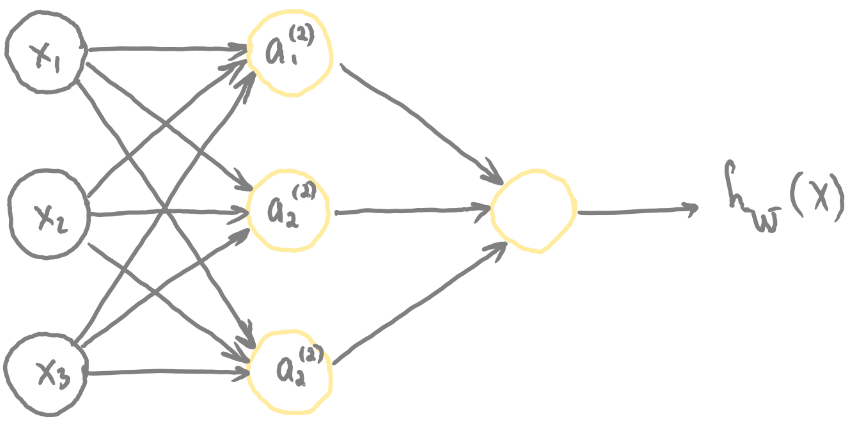


Pentru logistic regression (clasificare liniară)

$$h_w(x) = \frac{1}{1+e^{-w^T x}}$$

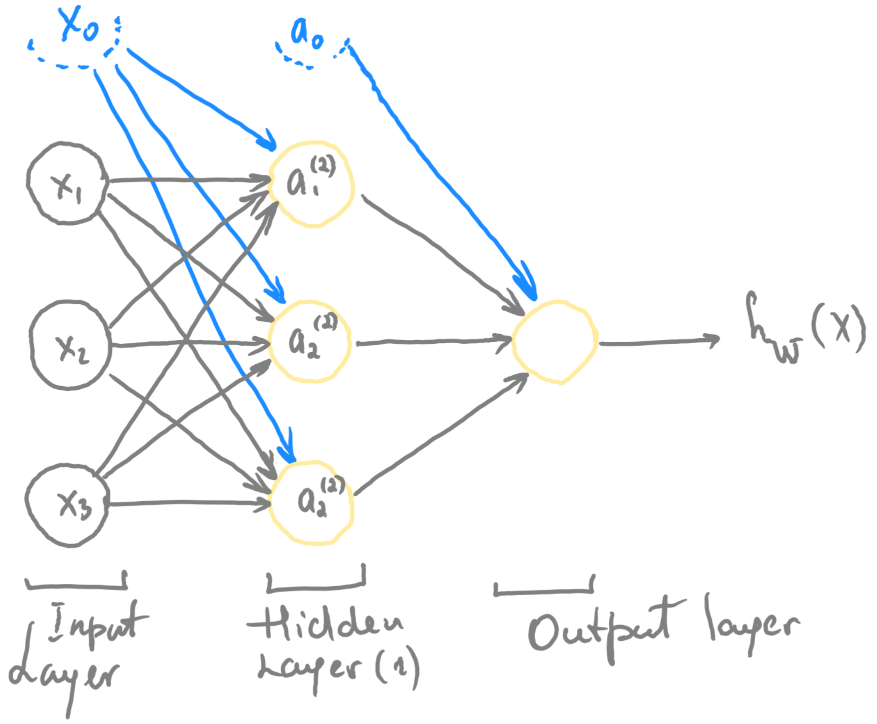
Aceasta este o reprezentare a unui neuron, dar cum arata reprezentarea unei retele neuronale?

O retea neuronala este pur si simplu un grup de astfel de "neuroni" uniti.

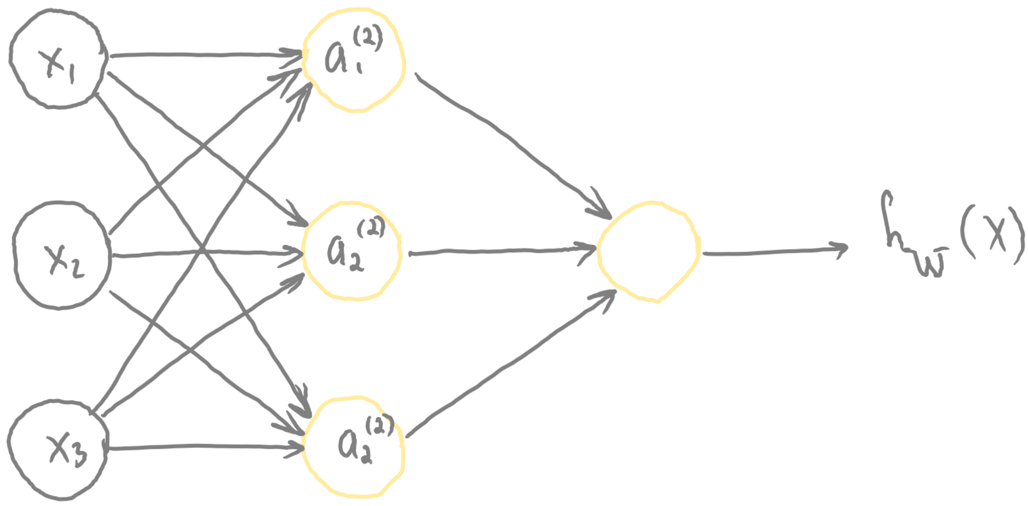


Adesea o să reprezint rețeaua neuronală cu sau fără termenul bias:

x_0 $\qquad$ $a_0^{(2)}$



Pentru a explica calculele din reteaua neuronala voi folosi urmatoarea notatie:



$a_i^{(j)}$ - "activarea" unității i în layerul j

$W^{(j)}$ - matricea de parametri (weights) care controlează map-area din care controlează map-area din layerul j în layerul j+1

Prin urmare, calculele pentru reteau neuronala de mai sus sunt date de:

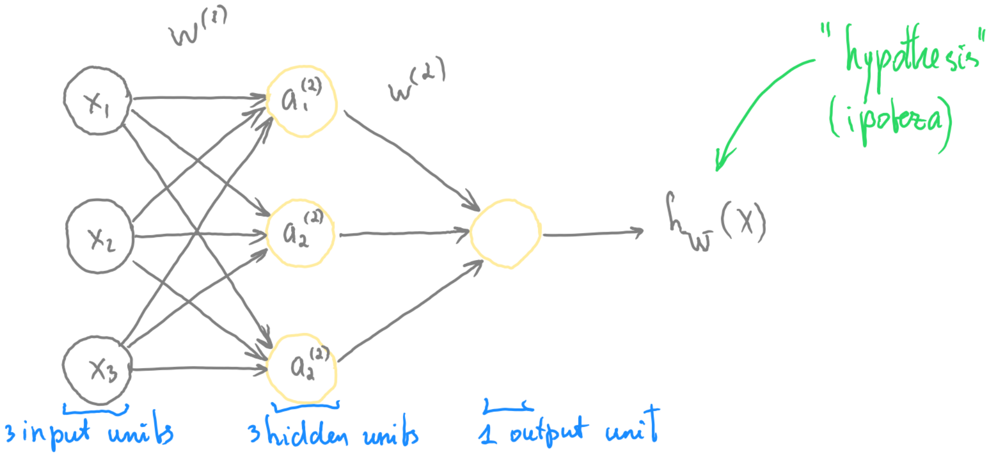
$$a_1^{(2)} = g(W_{10}^{(1)} x_0 + W_{11}^{(1)} x_1 + W_{12}^{(1)} x_2 + W_{13}^{(1)} x_3)$$

$$a_2^{(2)} = g(W_{20}^{(1)} x_0 + W_{21}^{(1)} x_1 + W_{22}^{(1)} x_2 + W_{23}^{(1)} x_3)$$

$$a_3^{(2)} = g(W_{30}^{(1)} x_0 + W_{31}^{(1)} x_1 + W_{32}^{(1)} x_2 + W_{33}^{(1)} x_3)$$

$$h_W(x) = a_1^{(3)} = g\left(W_{10}^{(2)} a_0^{(2)} + W_{11}^{(2)} a_1^{(2)} + W_{12}^{(2)} a_2^{(2)} + W_{13}^{(2)} a_3^{(2)}\right)$$

Care ar fi dimensiunile vectorilor W la fiecare layer?



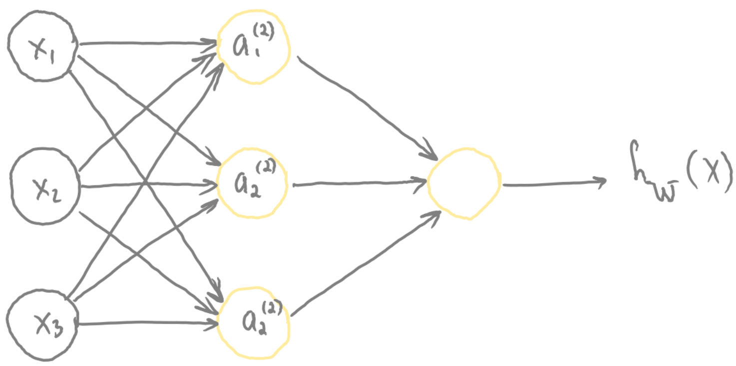
Care este dimensiunea $W^{(1)}$, $W^{(2)}$?

$W^{(1)} \in \mathbb{R}^{\_\_\_}$ $\quad$ $W^{(2)} \in \mathbb{R}^{\_\_\_}$

$\in \mathbb{R}$

Dacă o rețea neuronală are s_j unități în layer-ul j, s_{j+1} unități în layerul $j+1$, atunci $w^{(j)}$ va avea dimensiunea $s_{j+1} \times (s_j + 1)$

Avand ipoteza noastra definita, sa ne uitam cum are loc "trecerea" datelor prin retea (forward pass):



Având:

$$a_1^{(2)} = g(\underbrace{w_{10}^{(1)} x_0 + w_{11}^{(1)} x_1 + w_{12}^{(1)} x_2 + w_{13}^{(1)} x_3}_{z_1^{(2)}})$$

$$a_2^{(2)} = g(\underbrace{w_{20}^{(1)} x_0 + w_{21}^{(1)} x_1 + w_{22}^{(1)} x_2 + w_{23}^{(1)} x_3}_{z_2^{(2)}})$$

$$a_3^{(2)} = g(\underbrace{w_{30}^{(1)} x_0 + w_{31}^{(1)} x_1 + w_{32}^{(1)} x_2 + w_{33}^{(1)} x_3}_{z_3^{(2)}})$$

$$h_W(x) = a_1^{(3)} = g\left(w_{10}^{(2)} a_0^{(2)} + w_{11}^{(2)} a_1^{(2)} + w_{12}^{(2)} a_2^{(2)} + w_{13}^{(2)} a_3^{(2)}\right)$$

$z_1^{(2)}$, $z_2^{(2)}$, $z_3^{(3)}$ – combinații liniare parametrizate $(w^T x)$

Dacă e să ne uităm mai atent la

Dacă e să ne uităm mai atent la

observăm că acesta poate fi scris ca:

$$\underbrace{\begin{bmatrix} w_{10}^{(1)} & w_{11}^{(1)} & w_{12}^{(1)} & w_{13}^{(1)} \\ w_{20}^{(1)} & w_{21}^{(1)} & w_{22}^{(1)} & w_{23}^{(1)} \\ w_{30}^{(1)} & w_{31}^{(1)} & w_{32}^{(1)} & w_{33}^{(1)} \end{bmatrix}}_{\bar{W}^{(1)}} \underbrace{\begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix}}_{X} = \bar{W}^{(1)} \cdot X$$

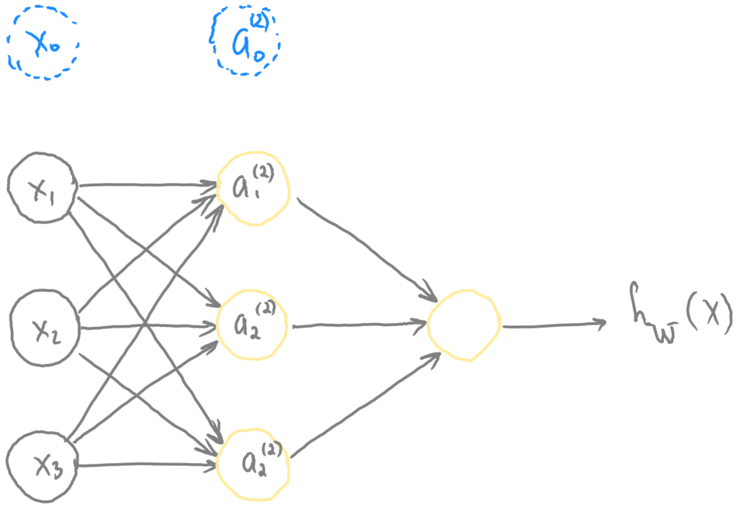
Având notațiile de mai sus $z_1^{(2)}$, $z_2^{(2)}$, $z_3^{(3)}$
putem scrie:

$$z^{(2)} = \begin{bmatrix} z_1^{(2)} \\ z_2^{(2)} \\ z_3^{(2)} \end{bmatrix} = \bar{W}^{(1)} \underbrace{X}_{= a^{(1)}}$$

$$\underbrace{a^{(2)}}_{\in \mathbb{R}^3} = g(\underbrace{z^{(2)}}_{\in \mathbb{R}^3})$$

Aceste formule ne ajută să calculăm $a_1^{(2)}, a_2^{(2)}, a_3^{(2)}$.

Dar cum rămâne cu termenul bias?



Pentru termenul bias adăugăm $a_0^{(2)} = 1 > a^{(2)} \in \mathbb{R}^4$

Pentru termenul bias adăugăm $a_0^{(2)} = 1 \rightarrow a^{(2)} \in \mathbb{R}$

$$z^{(3)} = W^{(2)} a^{(2)}$$

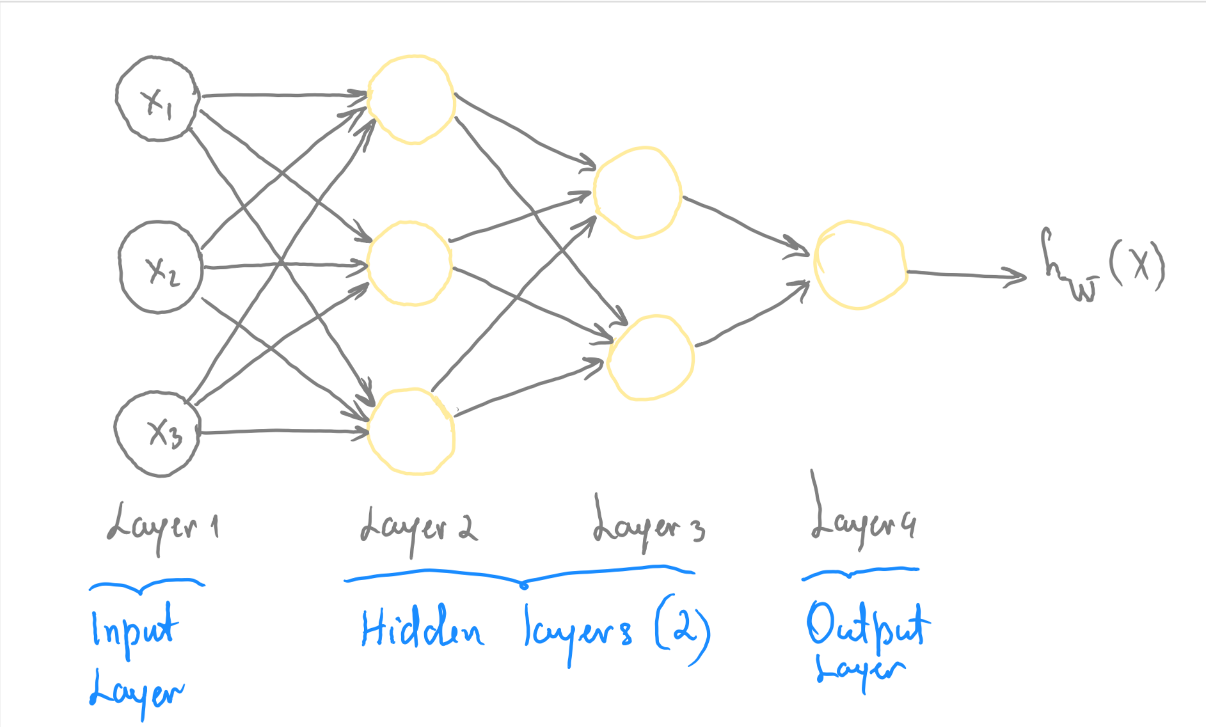
todo: de ilustrat $z^{(3)}$ mai sus

$$h_W(x) = a^{(3)} = g(z^{(3)})$$

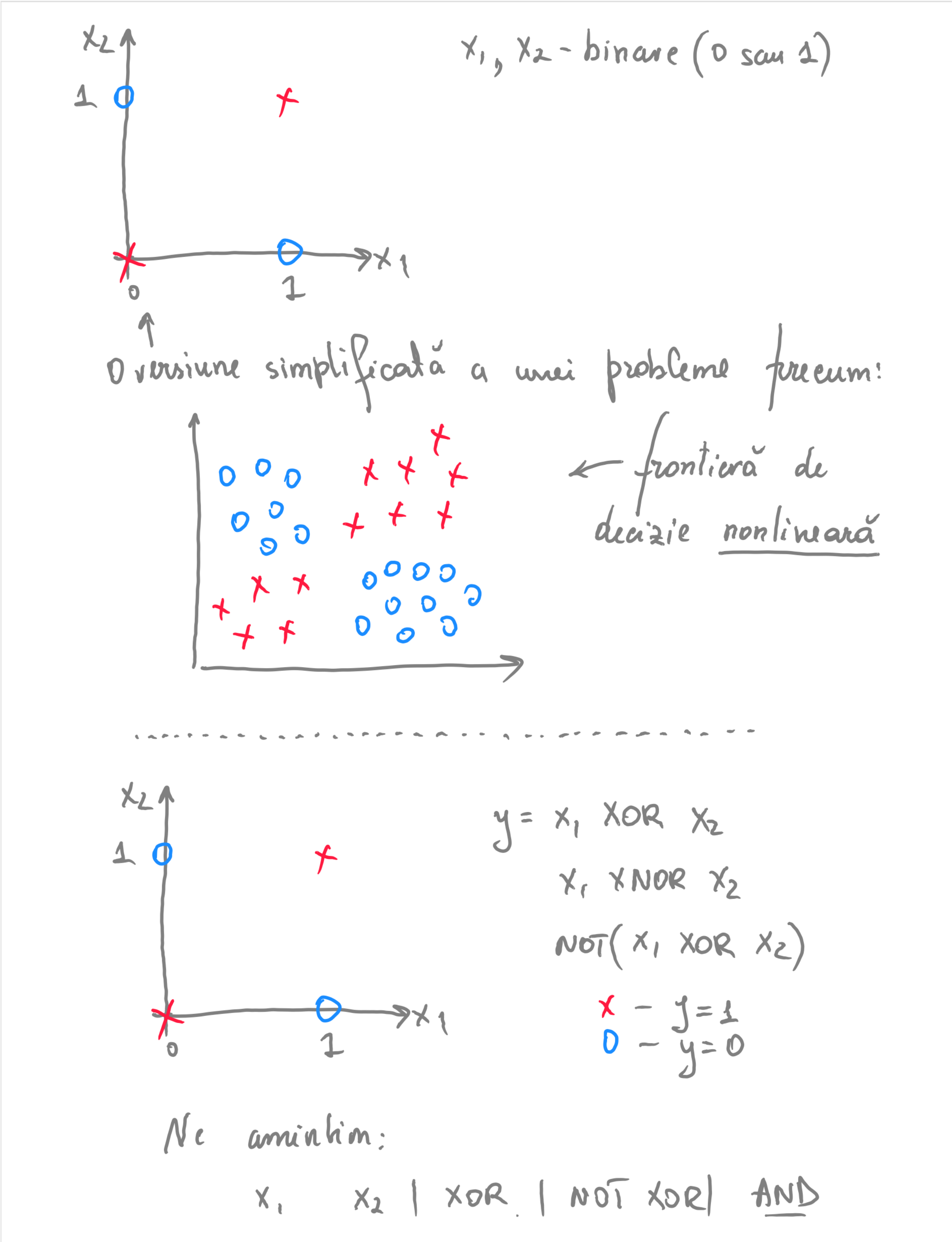
Acest proces de propagare a datelor X de la stânga la dreapta se numește "forward propagation".

Idee: explicatie de ce asta se aseamana mult cu logistic regression, si cum NN invata features complexe.

Apropo, diferitw configuratii a felului cum se conecteaza neuronii formeaza diferite "arhitecturi":



Spre deosebire de modelel anterioare (linear/logistic regression) retelele neuronale se descurca foarte bine la invatarea ipotezelor complexe non-lineare. Sa analizam urmatorul exemplu:

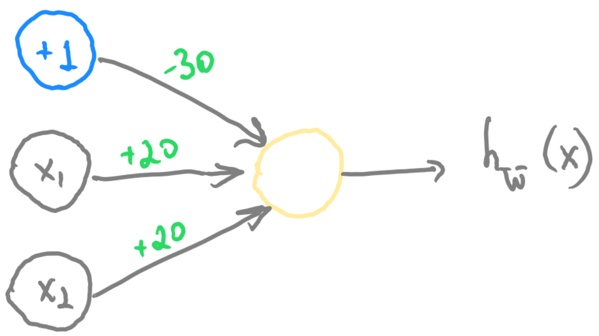


0	0	0	1	0
1	0	1	0	0
0	1	1	0	0
1	1	0	1	1

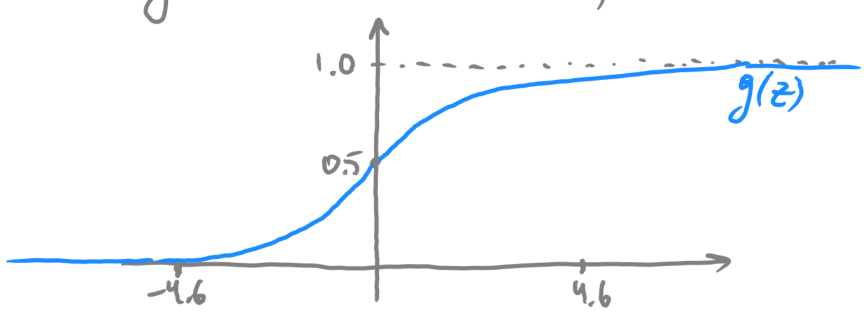
Sa analizam intai un exemplu pentru functia AND:

$x_1, x_2 \in \{0, 1\}$

$y = x_1 \text{ AND } x_2$



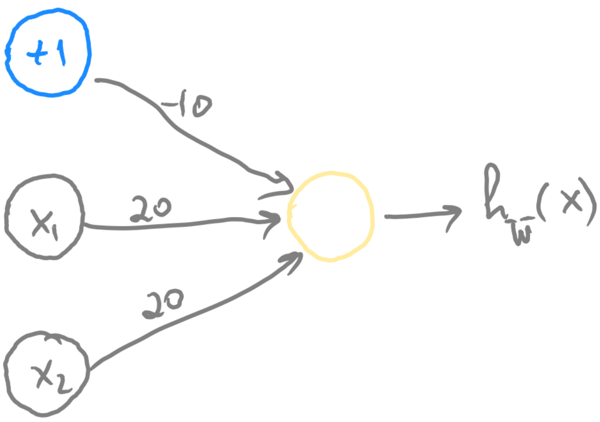
$$h_{\bar{w}}(x) = g(-30 \cdot 1 + 20 \cdot x_1 + 20 \cdot x_2)$$



x_1	x_2	$h_w(x)$

0	0	
0	1	
1	0	
1	1	

Dar cum arata acelasi exemplu pentru functua OR?



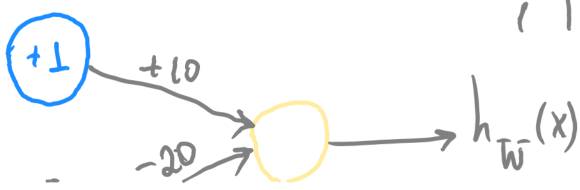
x_1	x_2	$h_{\bar{w}}(x)$
0	0	$g(-10) \approx 0$
0	1	$g(10) \approx 1$
1	0	$g(10) \approx 1$
1	1	$g(30) \approx 1$

$$g(-10 \cdot 1 + 20 \cdot x_1 + 20 \cdot x_2)$$

Cum ar lucra acelasi exemplu pentru functia de "Negatie" - NOT:

Negația :

X	NOT X
0	1
1	0



x_1	$h_{\bar{w}}(x)$
0	$g(10) \approx 1$

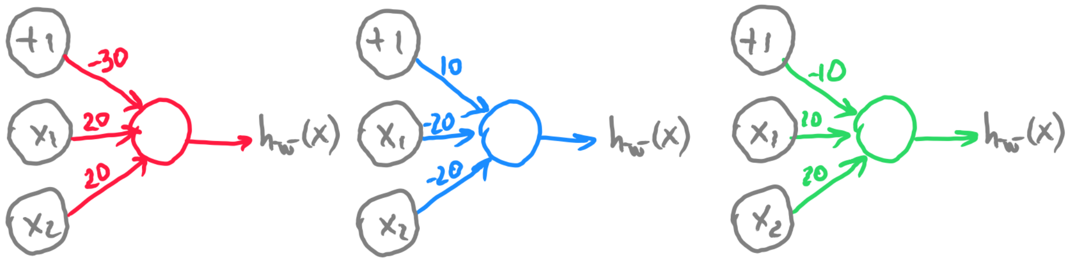
$g(10) \approx 1$

1 | $g(-10) \approx 0$

$h_{\bar{w}}(x) = g(10 - 20x_1)$

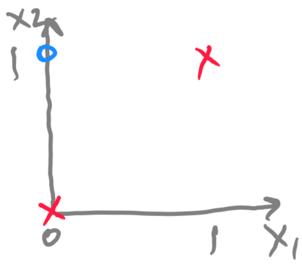
! Idee: încercați să reprezentați rețeaua neuronală și găsiți valorile parametrilor pentru $(\text{NOT } x_1) \text{ AND } (\text{NOT } x_2)$

Avand exemplele de mai sus, sa unim "puzzleul":



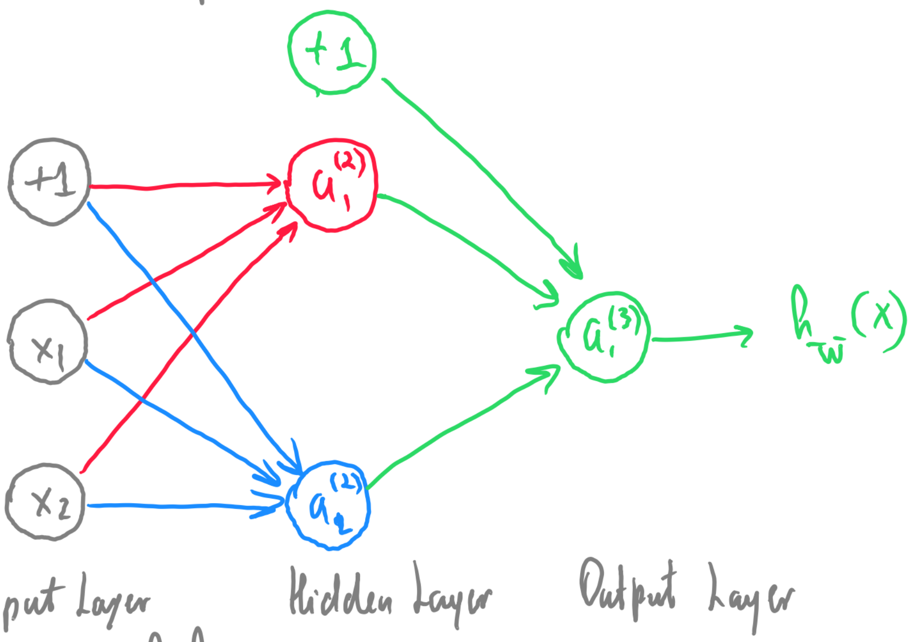
→ x_1 AND x_2 $\quad$ $(\text{NOT } x_1) \text{ AND } (\text{NOT } x_2)$ $\quad$ x_1 OR x_2

Ne amintim că ceea ce vrem să obținem este o separare nonliniară:



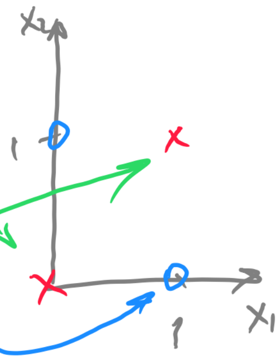
Să reprezentăm rețeaua

Să reprezentăm rețeaua:

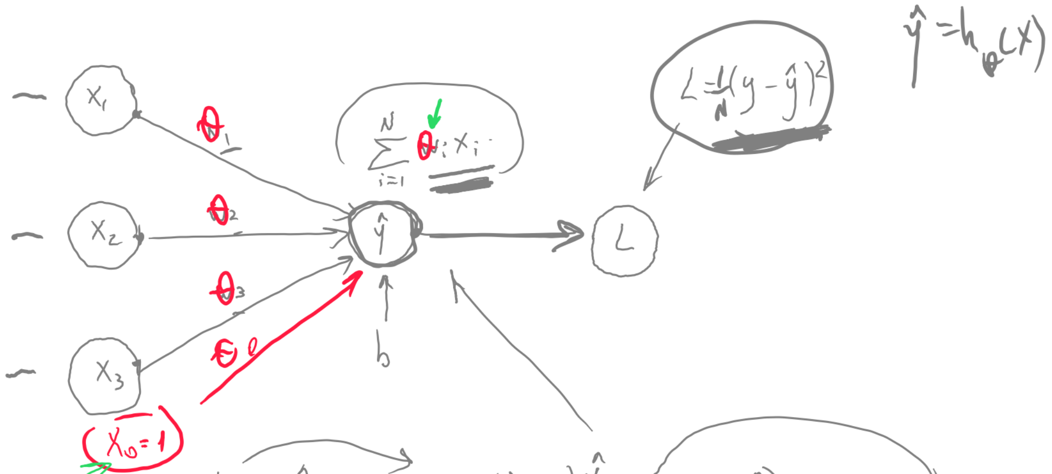


Tabelul de adevăr

x_1	x_2	$a_1^{(2)}$	$a_2^{(2)}$	$h_{\bar{w}}(x)$
0	0	0	1	1
0	1	0	0	0
1	0	0	0	0
1	1	1	0	1



Acum avem parametrii w "optimali". Dar cum aflam acesti parametri in alte probleme pe care incercam sa le rezolvam? Sa ne amintim de lectia de linear regression:



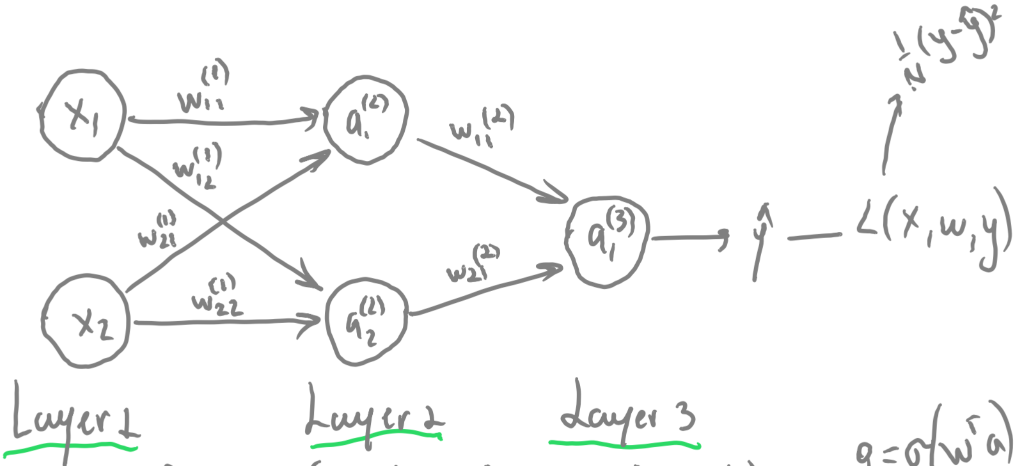
$$\frac{\partial L}{\partial \theta_i} = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial \theta_i} = 2(y-\hat{y}) \cdot \frac{\partial \hat{y}}{\partial \theta_i} = \underbrace{2(y-\hat{y})}_{\text{Loss gradien}} \cdot \underbrace{x_i}_{\text{Linear operation gradient}}$$

$$\sum_{i=0}^{N} \theta_i x_i = \overbrace{\theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3}^{\alpha}$$

$$\frac{\partial \alpha}{\partial \theta_1} = x_1$$

todo: definirea linear regression cu notația de mai sus. Rescrierea derivatei.

Cum se aplica asta la retele "mai complicate"?



$g = \sigma(w^T a)$

$$a_1^{(3)} = g\left(w_1^{(2)} a_1^{(2)} + w_2^{(2)} a_2^{(2)}\right)$$

$$w_1^{(2)} = \frac{\partial L}{\partial w_1^{(2)}} = \frac{\partial L}{a_1^{(3)}} \cdot \frac{\partial a_1^{(3)}}{\partial g} \cdot \frac{\partial g}{\partial w_1^{(2)}}$$

loss.backward()

linear2.backward()

activation.backward()