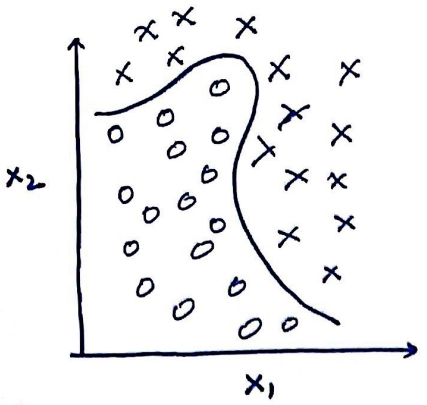
WEEK-4

Neural Networks.



$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \dots \theta_n x_1^d x_2^d)$

↙ Non-Linear hypotheris

→ Linear/Logistic regression works well when we have less features (i.e 2 or 3)

x_1
x_2
x_3
x_4
$\vdots$
x_n

$O(n^2) \approx \frac{n^2}{2}$

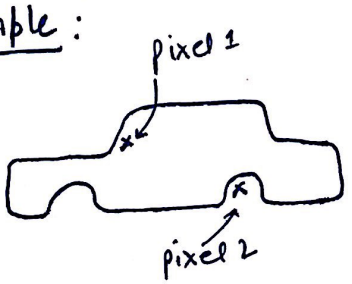
↓

$x_i^d x_j^d$

computationally expensive when ~~n~~ n → large

$x_i\ x_j\ x_k \Rightarrow O(n^3)$

Example :



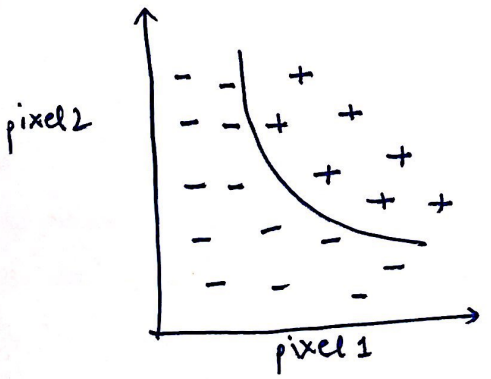
→ learning Algorithm.

50 × 50 pixel images → 2500 pixels

$n = 2500$ (7500 if RGB)

Gray scale ←

$$x = \begin{bmatrix} \text{pixel 1 intensity} \\ \vdots \\ \text{pixel 2500} \end{bmatrix}$$



+ → Car

− → Not Car.

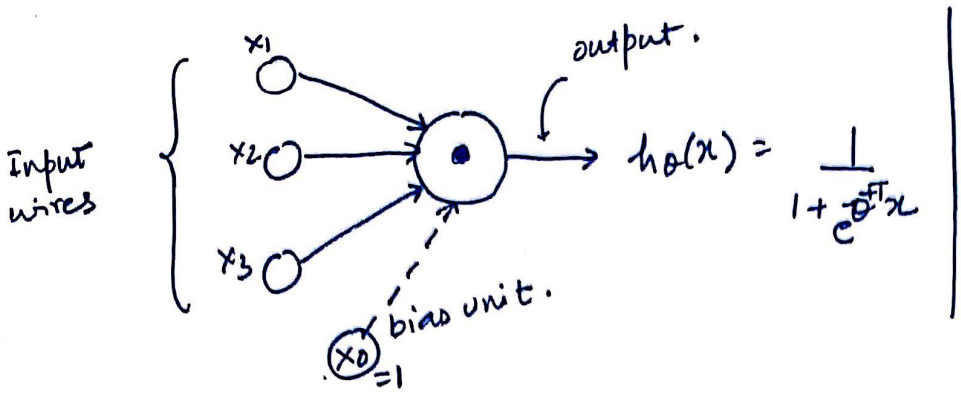
Quadratic features ($x_i \times x_j$)

≈ 3 million features.

~~1000~~ $^{2500}C_2$.

Neural Networks & Brain

Model Representation

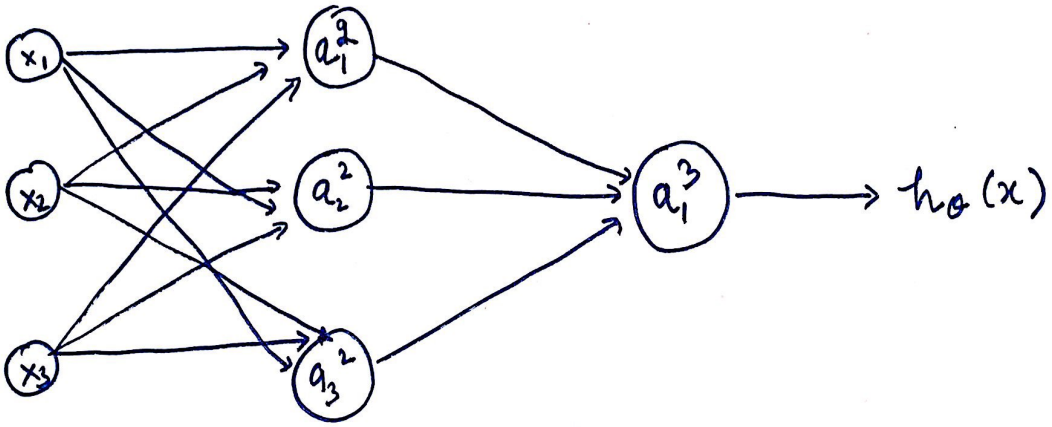


single Neuron model: logistic unit.

→ sigmoid (logistic) activation fn. : Neuron

→ parameters ≈ weights (θ)

Neural Network:



Layer 1 — Input Layer

Layer 2 — Hidden Layer

Layer 3 — Output Layer

$a_i^{(j)}$ = "activation" of unit i in layer j

$\Theta^{(j)}$ = matrix of weights controlling fn. mapping from layer j to layer $j+1$.

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

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

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

$$\underline{h_\Theta(x)} = a_1^{(3)} = g(\Theta_{10}^{(2)} a_0^{(2)} + \Theta_{11}^{(2)} a_1^{(2)} + \Theta_{12}^{(2)} a_2^{(2)} + \Theta_{13}^{(2)} a_3^{(2)})$$

final o/p.

$$= g(z_1^{(3)})$$

→ if network has s_j units in layer j, s_{j+1} units in layer $j+1$

then $\Theta^{(j)}$ will be of dimension ⇒ $\left[s_{j+1} \times (s_j + 1)\right]$

$\therefore \Theta^{(1)} \Rightarrow 3 \times 4$ matrix.

→ Vectorized implementation

$$a^{(1)} = x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad z^{(2)} = \begin{bmatrix} z_1^{(2)} \\ z_2^{(2)} \\ z_3^{(2)} \end{bmatrix}$$

$$\left\{ \begin{array}{l} z^{(2)} = \Theta^{(1)} x \quad (x = a^{(1)}) \\ a^{(2)} = g(z^{(2)}) \end{array} \right\}$$

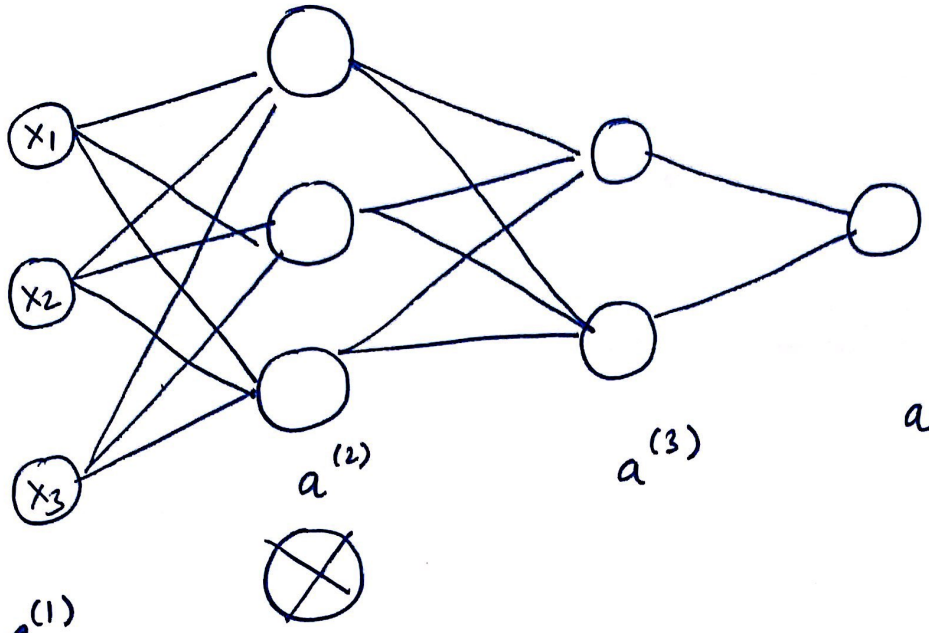
$a^{(2)} \in \mathbb{R}^{3\times 1}$, $z^{(2)} \in \mathbb{R}^{3\times 1}$

→ Add $a_0^{(2)} = 1$

Forward propagation →

$$\left\{ \begin{array}{l} z^{(3)} = \Theta^{(2)} a^{(2)} \\ h_\Theta(x) = a^{(3)} = g(z^{(3)}) \end{array} \right\}$$

example: different neural architecture



$$a^{(1)} = \begin{bmatrix} a_0^{(1)} \\ a_1^{(1)} \\ a_2^{(1)} \\ a_3^{(1)} \end{bmatrix}$$

- $z^{(2)} = \Theta^1 \cdot a^1$
  $a^{(2)} = g(z^{(2)})$
  } Layer 2

  Add $a_0^{(2)} = 1$

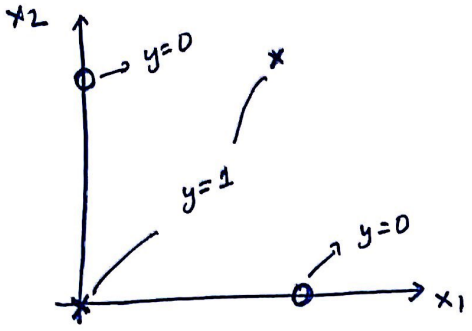
$z^{(3)} = \Theta^2 \cdot a^{(2)}$
$a^{(3)} = g(z^{(3)})$
} Layer 3

Add $a_0^{(3)} = 1$

Examples & Intuitions 1 : Non-linear classification example

XOR/NXOR

NOT $(x_1$ XOR $x_2) \rightarrow$ x_1 NXOR x_2 XOR.



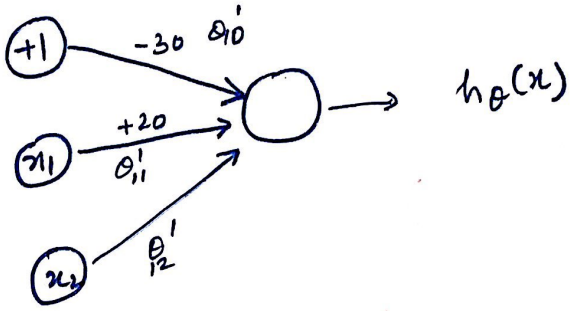
x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	0

x_1 XOR x_2 = $x_1 \cdot \bar{x}_2 + x_2 \cdot \bar{x}_1$

Example: AND

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

$y = x_1$ AND x_2

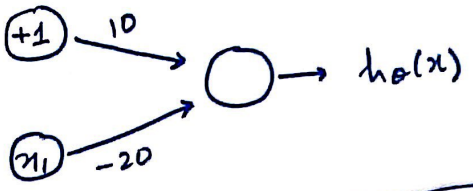


$h_\theta(x) = g(-30 \cdot +20x_1 + 20x_2) = x_1$ AND x_2

θ_{10}' θ_{11}' θ_{12}'

x_1	x_2	$h_\theta(x)$
0	0	$g(-30) \approx 0$
0	1	$g(-10) \approx 0$
1	0	$g(-10) \approx 0$
1	1	$g(10) \approx 1$

example : Negation :



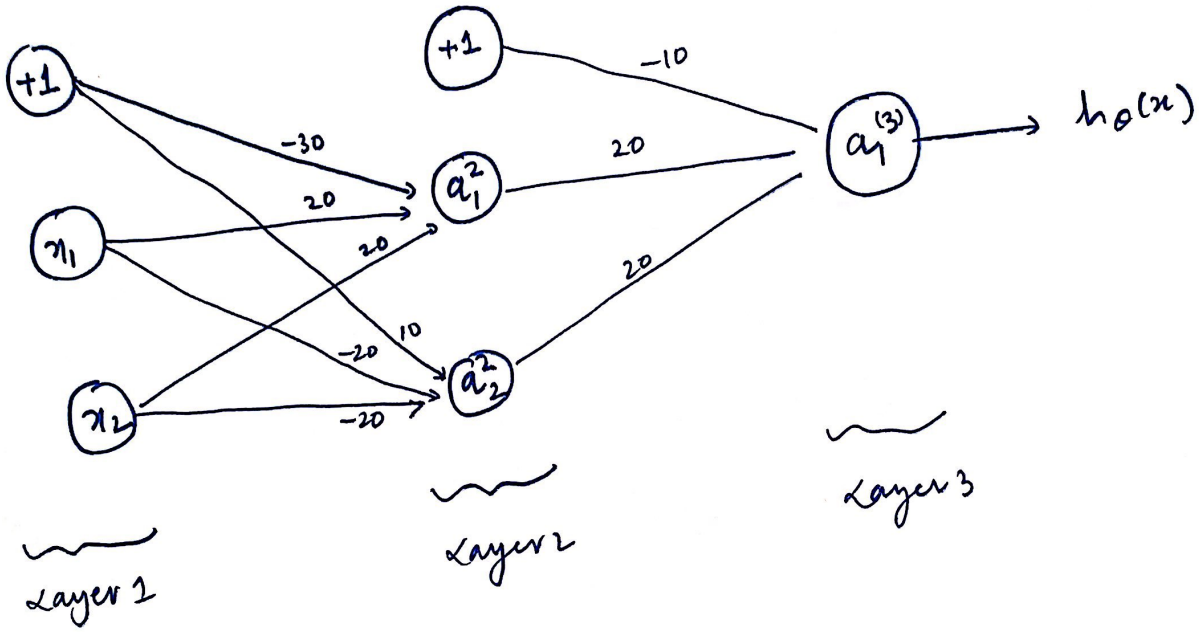
x_1	$h_\theta(x)$
0	1
1	0

$h_\theta(x) = g(10 - 20x_1)$

intuition: put large negative weight in front of the variable you want to negate.

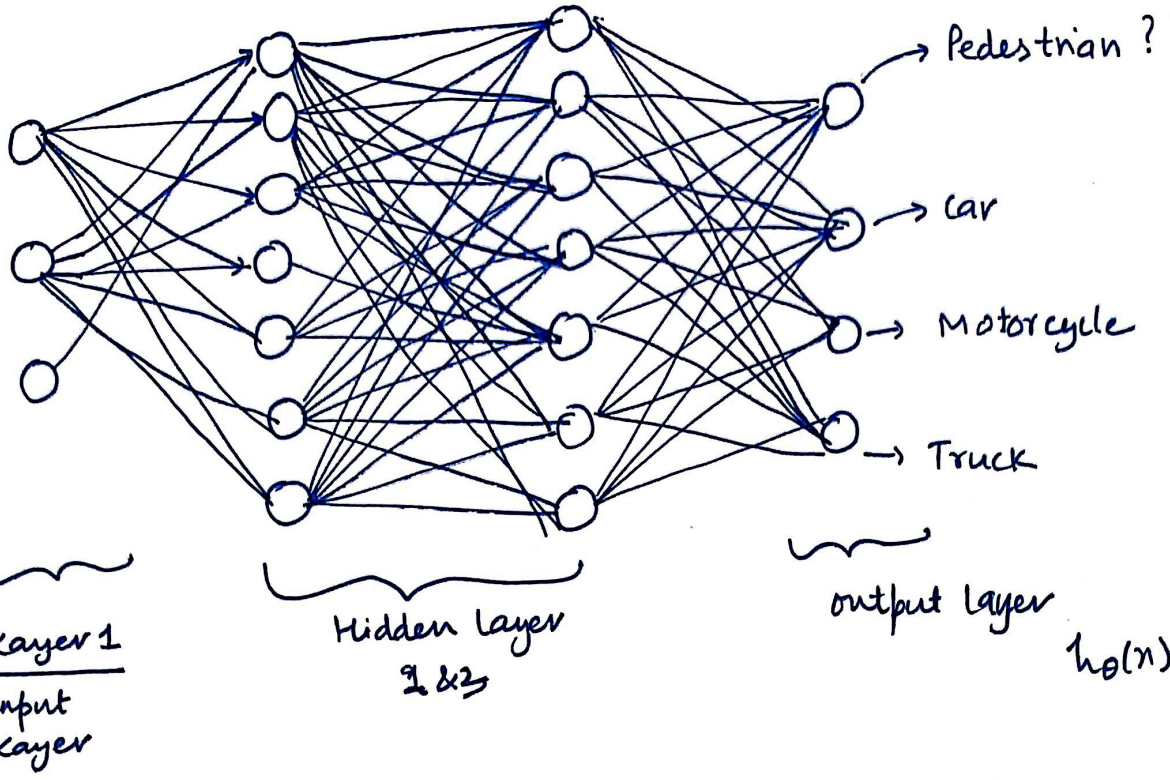
---

x_1 XNOR x_2



x_1	x_2	$a_1^{(2)}$	$a_2^{(2)}$	$h_\theta(x) = a_1^{(3)} = g(z_1^{(3)})$
0	0	0	1	1
0	1	0	0	0
1	0	0	0	0
1	1	1	0	1

Multi-class Classification : One. Vs. All



$h_\theta(x) = \mathbb{R}^4$

$h_\theta(x) = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$ when pedestrian $\quad h_\theta(x) = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$ etc.

when Motorcycle

$h_\theta(x) = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$ when car

---

Training set : $(x^1, y^1), (x^2, y^2) \ldots \quad (x^m, y^m)$

(y^i) will be one of $\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$

$h_\theta(x^i) \approx y^{(i)}$

$\mathbb{R}^4$

predict : Neural network.

```
function p = predict (theta1, theta2, X)
%
  m = size(X,1);
  p = zeros(size(X,1), 1);          % 5000 x 1
  X = [ones(size(X,1),1) X]         % 5000 x 401

  out_1 = sigmoid(x * theta1')

  Output
  out_1 = [ones(size(outp_1),1) output 1];

  out2 = sigmoid(out_1 * theta2')

  p = max(out2, [], 2)

end
```