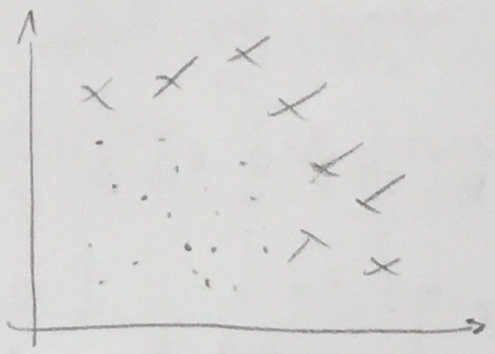
COURSERA ML WEEK 4 - NEURAL NETWORKS

LEO ROBINOVITCH

Non-Linear Hypothesis:



could use $g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1 x_2 + \dots)$ (Non-Linear)

x_1 = size

x_2 = # bedrooms

x_3 = # floors

x_4 = age

⋮

x_{100}

$\Rightarrow$ $x_1^2, x_1 x_2, x_1 x_3, x_1 x_4 \dots x_1 x_{100}$

$x_2^2, x_2 x_3, x_2 x_4 \dots x_2 x_{100}$

$\Downarrow$

$\simeq$ 5000 features!

- # features $O(n^2) \simeq \frac{n^2}{2}$

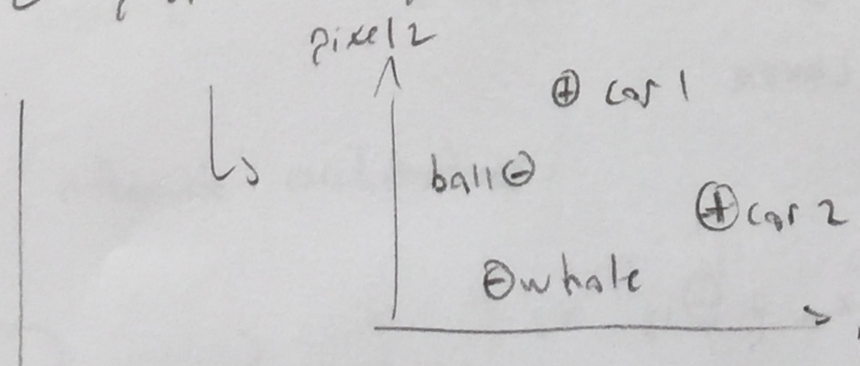
↳ computationally ✗. Could just include $x_1^2, x_2^2 \dots x_{100}^2$, but doesn't cover all decision boundary forms.

↳ additionally, including $x_1^2 x_2 \dots$ => # feat. $O(n^3)$!

=> for many problems, n is large.

eg: picture of car → matrix of pixel intensity values

2 pics, 2 pixel locations (same locations on each pic)



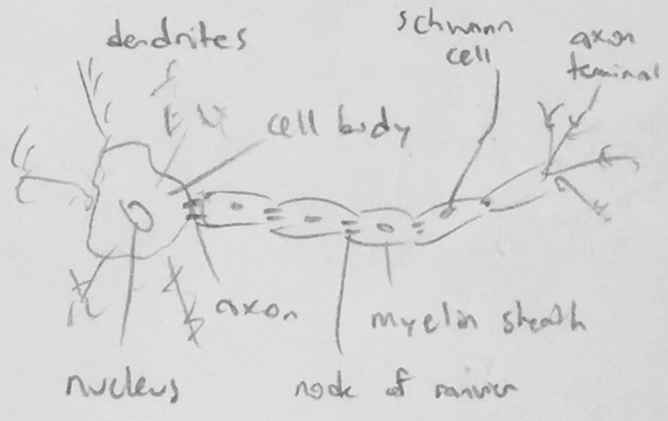
↳ 50×50 pixel imges → $n = 2500$! $x = \begin{bmatrix} \text{pixel 1} \\ \text{pixel 2} \\ \vdots \\ \text{pixel 2500} \end{bmatrix}$

↳ quadratic features: $x_i \times x_j \simeq 3\text{MM}$ features! $\left(\frac{n^2}{2}\right)$

=> Neural networks → better way to learn complex non-linear hypothesis than using massive # of features!

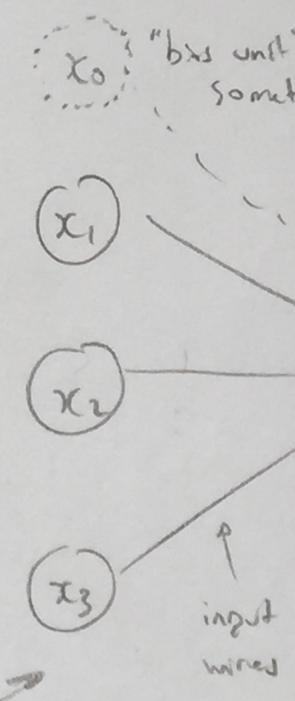
↳ plug any sensor into the brain → brain will learn how to interpret this data!

Model Representation 1:



↑

SINGLE NEURON



$$x = \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{bmatrix} \quad (x_0 = 1) \qquad \theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_n \end{bmatrix}$$

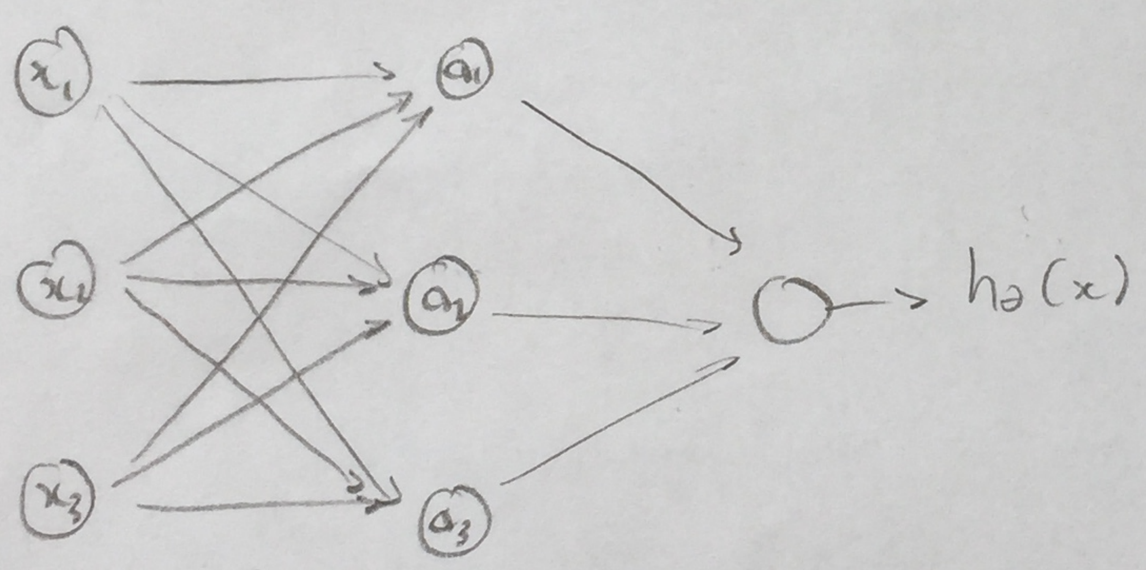
$n = 3$ here

parameters = "weights"

$$\left(h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}} \right)$$

$g(z)$: sigmoid (logistical) activation function

Neural Network:



Layer 1 — INPUT LAYER — x

Layer 2 — HIDDEN LAYER

Layer 3 — OUTPUT LAYER — y

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

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

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

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

$$a_3^{(2)} = \dots$$

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

If network has s_j units in layer j, s_{j+1} units in layer $j+1$ → $\Theta^j \in \mathbb{R}^{s_{j+1} \times (s_j + 1)}$

Model Representation II

layer 2 (hidden layer)

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

$a_2^{(2)} = g(z_2^{(2)})$

$a_3^{(2)} = g(z_3^{(2)})$

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

$a^{(2)} = g(z^{(2)})$

$\mathbb{R}^3$ $\mathbb{R}^3$

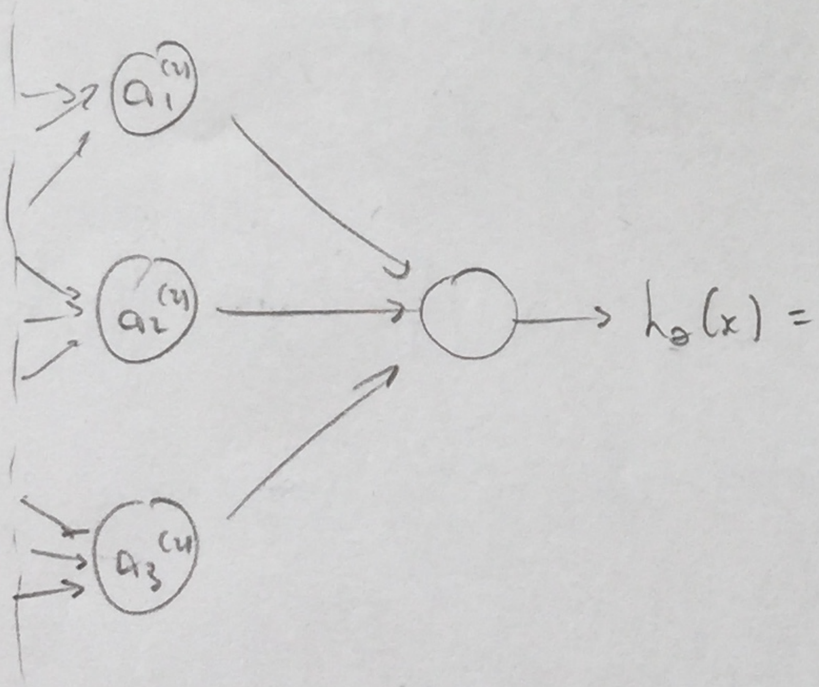
$\Rightarrow$ add $a_0^{(2)} = 1$ $(x_0) \longrightarrow a^{(2)} \in \mathbb{R}^4$

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

$h_\Theta(x) = a^{(3)} = g(z^{(3)})$

$\Leftarrow$ Vectorized implementation of forward propagation

Note:



↳ just like logistic regression, but replaces $\underline{x}$ w/ $a^{(2)}$ → "learns its own features!"

Could have alternate architectures:

Vector Summary:

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

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

$$a_3^{(2)} = g(\dots)$$

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

$\Rightarrow$ $a_1^{(2)} = g(z_1^{(2)})$

$a_2^{(2)} = g(z_2^{(2)})$ $\Rightarrow$ $z_k^{(2)} = \Theta_{k0}^{(1)} x_0 + \Theta_{k1}^{(1)} x_1 + \dots + \Theta_{kn}^{(1)} x_n$

$a_3^{(2)} = g(z_3^{(2)})$

$\Rightarrow$ $x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$ $z^{(j)} = \begin{bmatrix} z_1^{(j)} \\ z_2^{(j)} \\ \vdots \\ z_n^{(j)} \end{bmatrix}$

$\Rightarrow$ if $x = a^{(1)}$ (first layer) $\rightarrow$ $z^{(j)} = \Theta^{(j-1)} a^{(j-1)}$ $\quad a^{(j-1)} \rightarrow [(n+1) \times 1]$

$[s_j \times 1]$ $\quad$ $[s_j \times (n+1)]$ where s_j # activation nodes

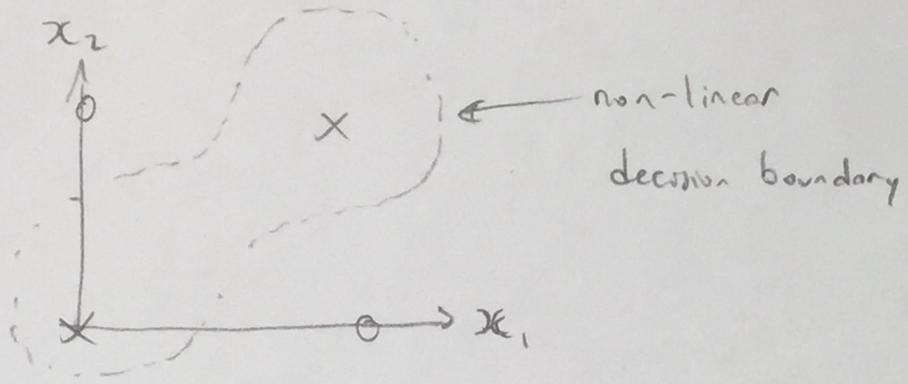
$\Rightarrow$ add bias unit $\rightarrow$ $a_0^{(j)} = 1$ added to top of $a^{(j-1)}$ to make $[(n+1) \times 1]$

$\Rightarrow$ $z^{(j+1)} = \Theta^{(j)} a^{(j)}$

$\hookrightarrow$ final $\Theta^{(j)}$ $[1 \times (n+1)]$ times $a^{(j)}$ $[(n+1) \times 1] \rightarrow [1 \times 1]$

$\Rightarrow$ $h_\Theta(x) = a^{(j+1)} = g(z^{(j+1)})$ $\leftarrow$ same as logistical regression!

Non-Linear Classification:



x_1, x_2 binary (1 or 0)

$y = x_1$ XOR x_2

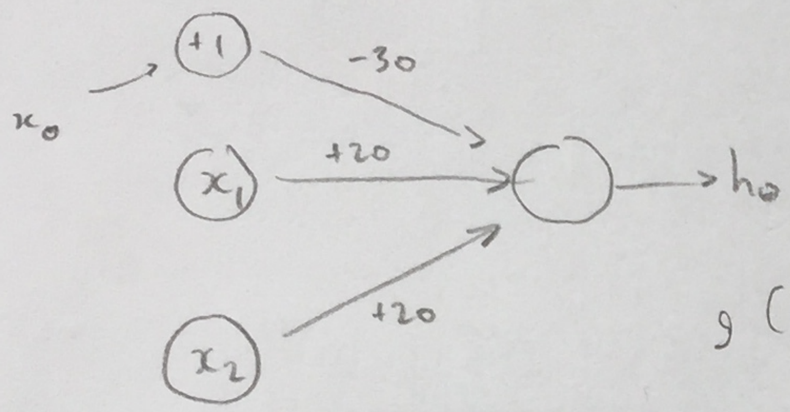
$y = x_1$ XNOR x_2 (same)

$y =$ NOT $(x_1$ XOR $x_2)$

] both true or both false

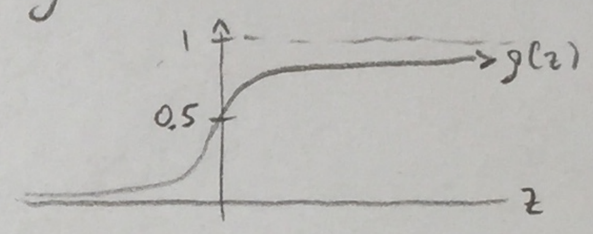
⇒ Neural Example: AND:

$x_1, x_2 \in [0,1]$, $y = x_1$ AND x_2:



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

$g(z) = \frac{1}{1+e^{-z}} \Rightarrow$



if $x_1 = x_2 = 0 \longrightarrow g(-30) \simeq 0$

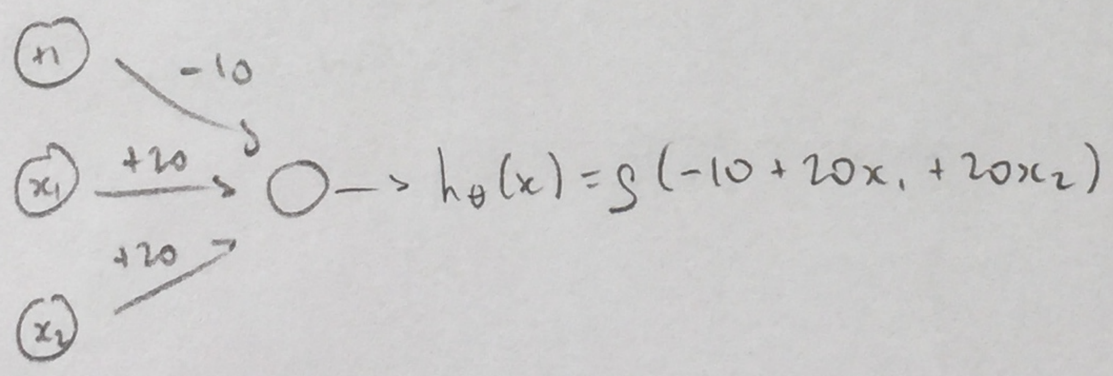
if $x_1 = 0$ & $x_2 = 1 \longrightarrow g(-10) \simeq 0$

if $x_1 = 1$ & $x_2 = 0$

if $x_1 = 1$ & $x_2 = 1 \longrightarrow g(10) \simeq 1$

] Logical AND function!

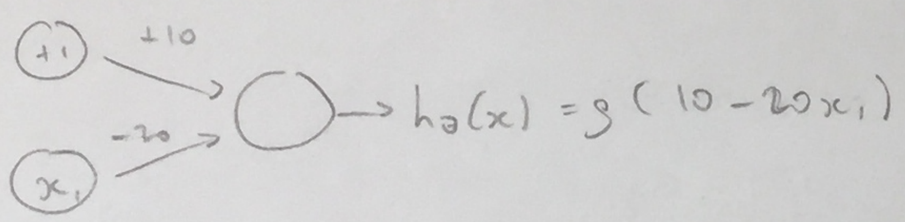
ex 2]



x_1	x_2	$h_\theta(x)$
0	0	0
0	1	1
1	0	1
1	1	1

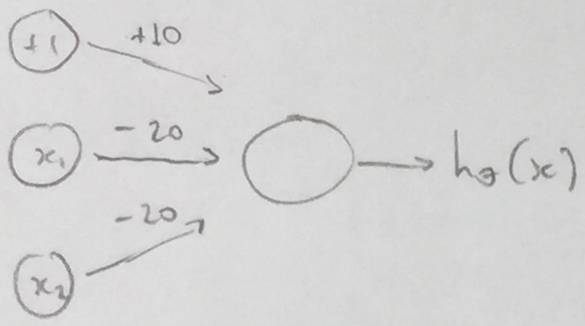
x_1 OR x_2 logic

⇒ Negation (NOT x_1):



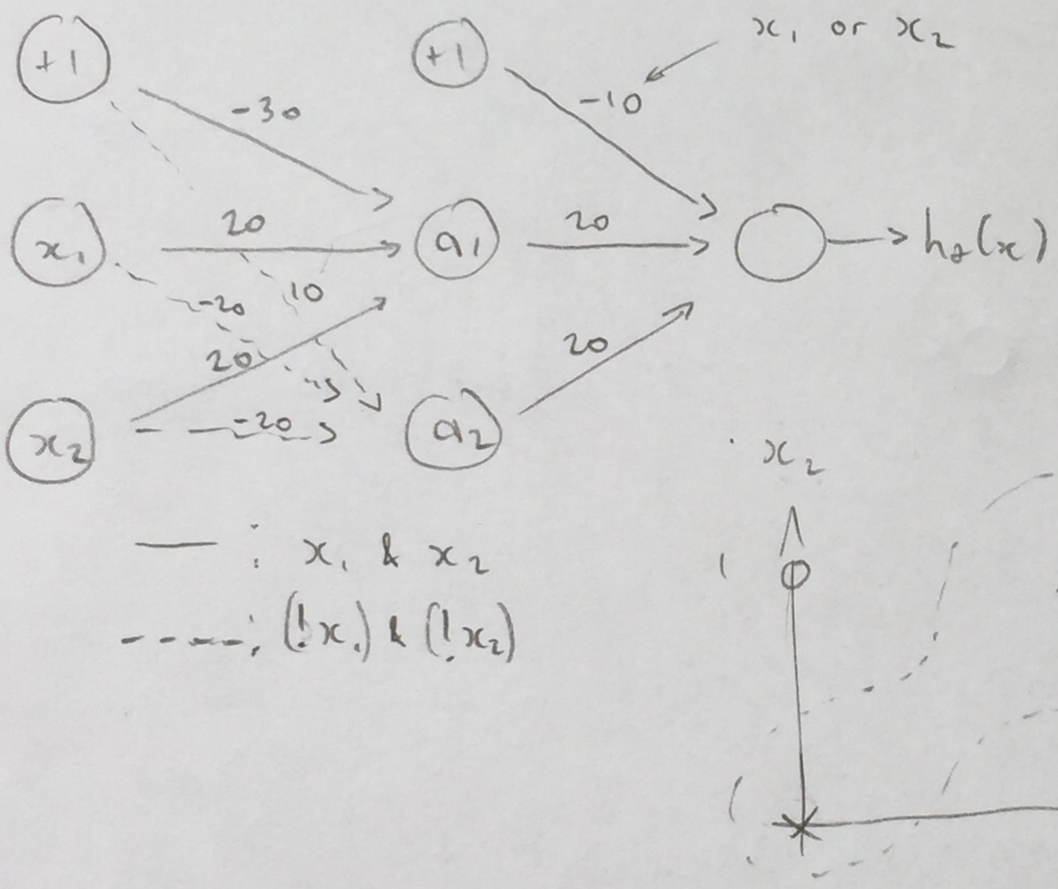
x_1	$h_\theta(x)$
0	1
1	0

⇒ (NOT x_1) AND (NOT x_2):



x_1	x_2	$h_\theta(x)$
0	0	1
0	1	0
1	0	0
1	1	0

To Create x_1 XNOR x_2 (both 0 or both 1):

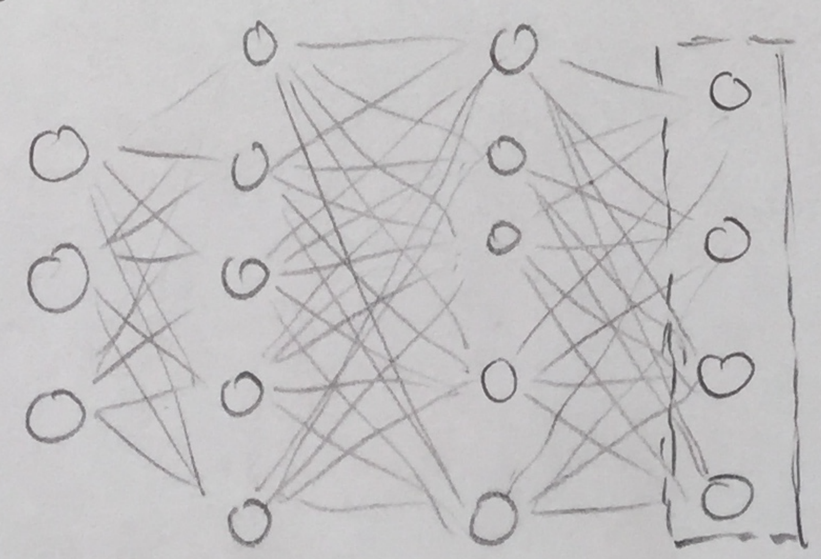


x_1	x_2	$a_1^{(2)}$	$a_2^{(2)}$	$h_\theta(x)$
0	0	0	1	1
0	1	0	0	0
1	0	0	0	0
1	1	1	0	1

Multiclass Classification:

eg] rather than y/n, 0/1 → digits from 0-9

eg: recognize car, pedestrian, motorcycle or truck:



4 logistical regression classifiers:

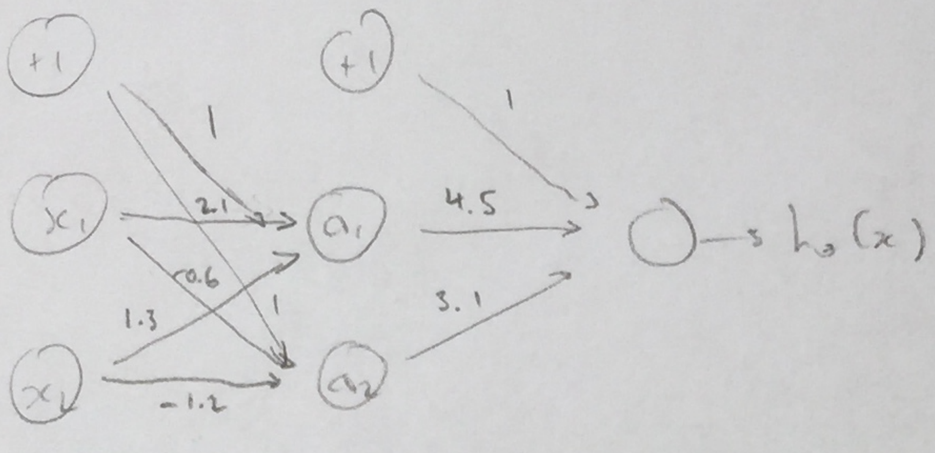
outputs $\begin{bmatrix}1\\0\\0\\0\end{bmatrix}, \begin{bmatrix}0\\1\\0\\0\end{bmatrix}$, etc.

- → Represent training set: $(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}) \dots (x^{(m)}, y^{(m)})$

  ↳ $y^{(i)}$ one of $\begin{bmatrix}1\\0\\0\\0\end{bmatrix}, \begin{bmatrix}0\\1\\0\\0\end{bmatrix}, \begin{bmatrix}0\\0\\1\\0\end{bmatrix}, \begin{bmatrix}0\\0\\0\\1\end{bmatrix}$ (4 classes)

  ↳ $h_\Theta(x^{(i)}) \simeq y^{(i)}$ (hopefully)

QUIZ.

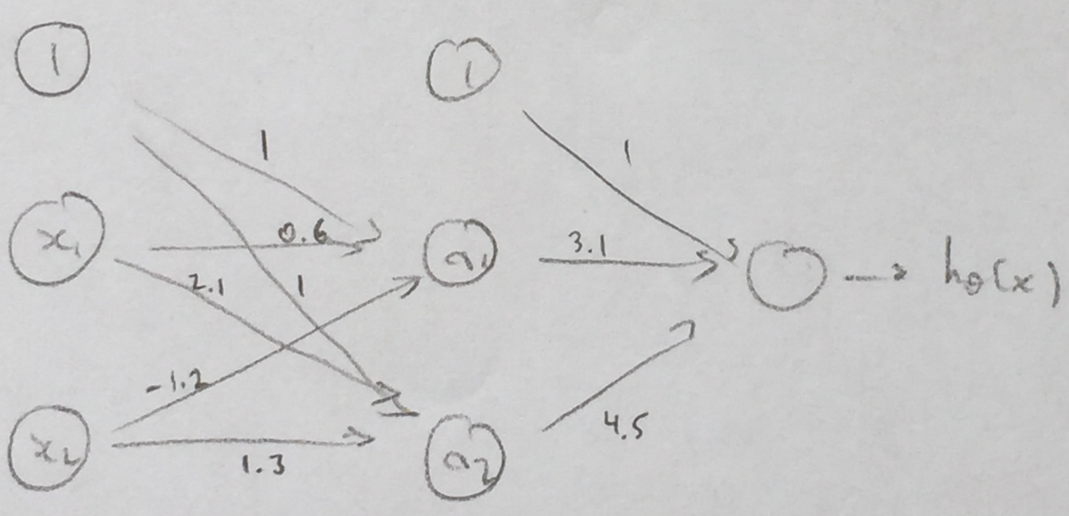


↳ $a_1 = g(1 + 2.1x_1 + 1.3x_2)$

$a_2 = g(1 + 0.6x_1 - 1.2x_2)$

→ $h_\Theta(x) = g(1 + 4.5a_1 + 3.1a_2)$

VS.



↳ $a_1 = g(1 + 0.6x_1 - 1.2x_2)$

$a_2 = g(1 + 2.1x_1 + 1.3x_2)$

→ $h_\Theta(x) = g(1 + 3.1a_1 + 4.5a_2)$

SAME