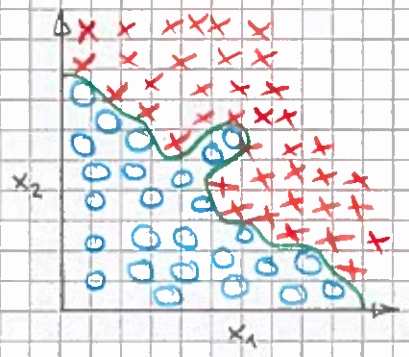
8. Neural Networks
- Non Linear Hypothesis



$$g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1 x_2 + \theta_4 x_1^2 x_2 + \theta_5 x_1^3 x_2 + \theta_6 x_1 x_2^2 + \dots)$$

by using only second order terms
$x_1^2, x_1 x_2 \dots x_{100}$

x_1 = size
x_2 = # bedrooms
x_3 = # floors
x_4 = # age
$\vdots$
x_{100}
} $n=100$

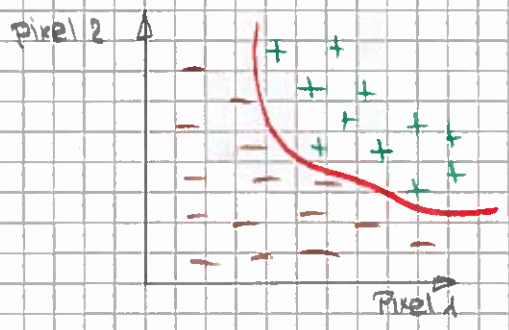
Including all the quadratic features would mean to have ≈ 5000 features
$\Rightarrow$ grows $O(n^2) \Rightarrow \frac{n^2}{2}$ $\Rightarrow$ could lead to over-fitting

just using the $x_1^2, x_2^2, x_3^2 \dots x_{100}^2$ does not allow to create a function as more complex functions, would not fit the complex dataset

while using cubic features it would $O(n^3)$ be about 170'000 features for $n = 100$ features

Why is non linear hypothesis relevant?

$\Rightarrow$ Learning Algorithm



+ Cars
- Non Cars

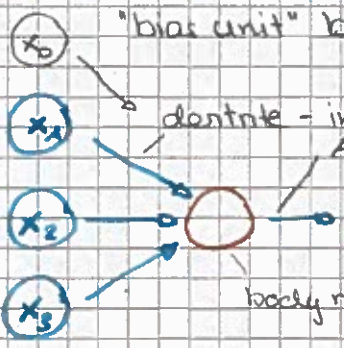
$\Rightarrow$ 50 x 50 pixels images $\Rightarrow$ 2500 pixels
$n = 2500$ (7500 RGB)

$$x = \begin{bmatrix} \text{pixel 1 intensity} \\ \text{pixel 2} \\ \text{pixel 3} \\ \vdots \\ \text{pixel n intensity} \end{bmatrix}$$ — Value 0-255

2500

by using all quadratic features $(x_i \times x_j)$
$\rightarrow$ 3 million features per training example
is computational expensive
Not good for algorith. with lot of features

Neuron model: Logistic unit



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

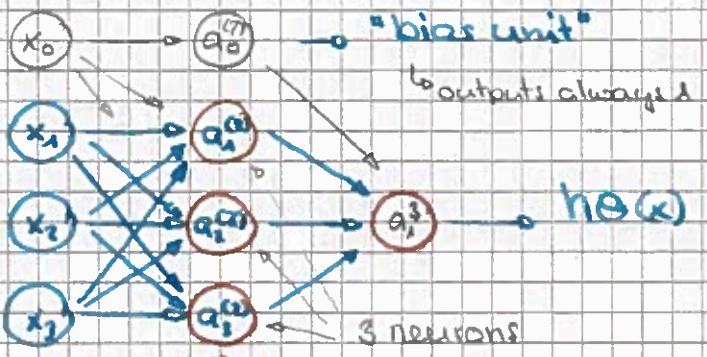
$$x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad \theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix}$$

↑ Sometimes called "weights" - parameters of model

Sigmoid (logistic) activation function.

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

Neural Network



Neuron

Layer 1 — Layer 2 — Layer 3

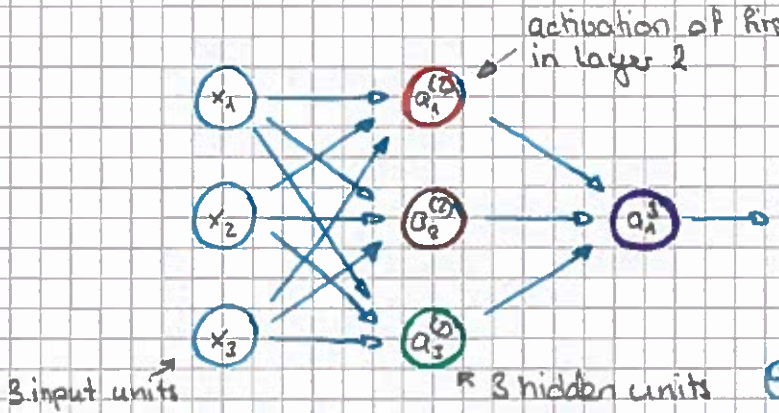
x — a — y

input lay. — hidden lay. — output lay.

*) you don't observe the values processed in the hidden layer

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

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



$\Theta^{(1)} \in \mathbb{R}^{3\times 4}$ dim Matrix

a "activation" is the value which is computed and output by the node

sigmoid

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

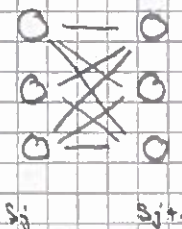
$$a_2^{(2)} = g(\Theta_{20}^{(1)} x_0 + \Theta_{21}^{(1)} x_1 + \Theta_{22}^{(1)} x_2 + \Theta_{13}^{(1)} x_3)$$

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

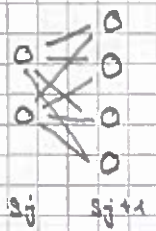
$$h_\Theta(x) = a_1^{(3)} = g(\Theta_{10}^{(2)} a_0^{(2)} + \Theta_{11}^{(2)} a_1^{(2)} + \Theta_{12}^{(2)} a_2^{(2)} + \Theta_{13}^{(2)} a_3^{(2)})$$

If network has s_j units in layer j, s_{j+1} in unit j+1, then $\Theta^{(j)}$ will be of dimension $(s_{j+1} \times s_j + 1)$

Example =>

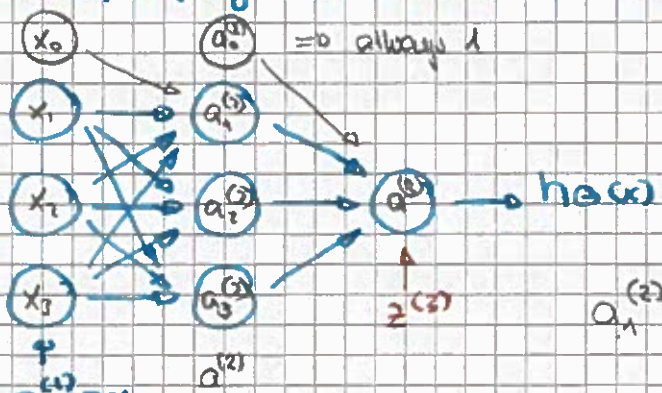


=> $\mathbb{R}$ 3 x 4



=> 4 x 3

Forward propagation: vectorized implementation



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

$a^{(1)} = x$ $\qquad a^{(2)}$

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

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

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

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

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

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

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

Vectorized implementation:

$z^{(2)} = \Theta^{(1)} x \Rightarrow \Theta^{(1)} \cdot a^{(1)}$

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

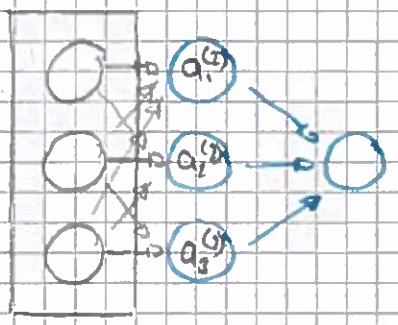
$a^{(2)} \in \mathbb{R}^3 \qquad z^{(2)} \in \mathbb{R}^3 \Rightarrow$ 3 dym. vector

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

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

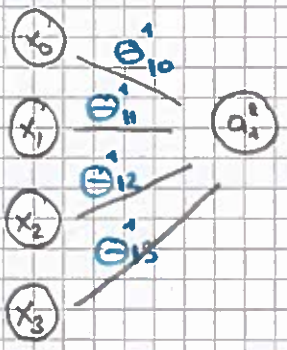
$h_\Theta(x) = a^{(3)} = g(z^{(3)})$

Neural Network learning its own features.



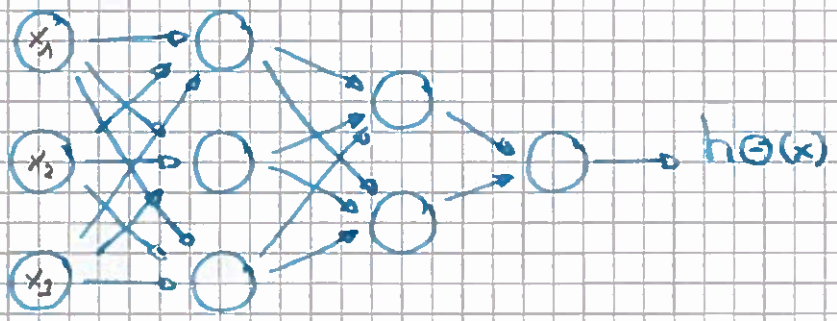
$$h_\Theta(x) = g(\Theta_{10}^{(2)} a_0^{(2)} + \Theta_{11}^{(2)} a_1^{(1)} + \Theta_{12}^{(2)} a_2^{(2)} + \Theta_{13}^{(2)} a_3^{(2)})$$

it's just logistic regression except using the new features $a_1^{(2)}, a_2^{(2)}, a_3^{(2)}$ instead of x_1, x_2, x_3

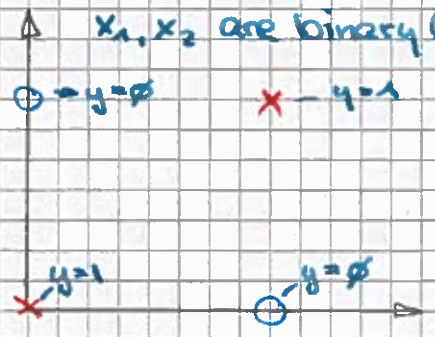


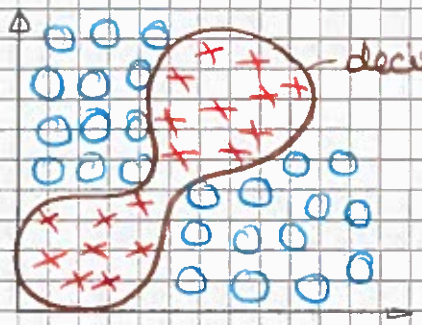
$$\hookrightarrow g(\Theta_{10}^1 x_0 + \Theta_{11}^1 x_1 + \Theta_{12}^1 x_2 + \Theta_{13}^1 x_3)$$

Other network architectures



Non linear classification example: XOR \ XNOR





$y = x_1$ XOR x_2

x_1 XNOR x_2

NOT (x_1 XOR x_2)

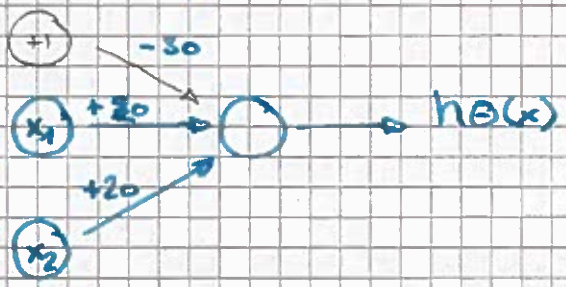
XOR

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

Simple example AND:

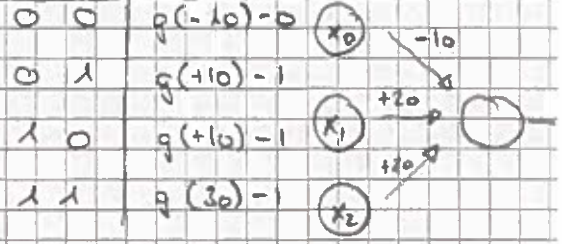
$x_1, x_2 \in \{0, 1\}$

$y = x_1$ AND x_2



OR:

x_1	x_2	h(x)
0	0	g(-10) ≈ 0
0	1	g(+10) ≈ 1
1	0	g(+10) ≈ 1
1	1	g(30) ≈ 1

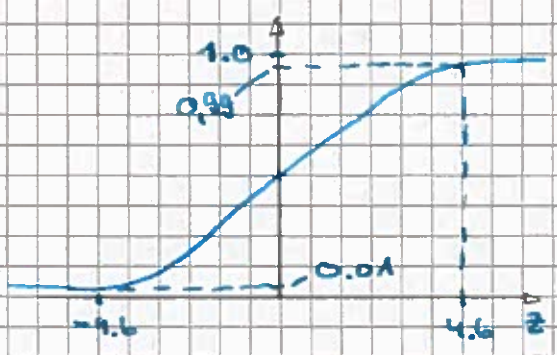


$g(-10 \cdot 1 + 20 \cdot 0 + 20 \cdot 0) = g(-10$

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

$\Theta_{10}^{(1)}$ $\Theta_{11}^{(1)}$ $\Theta_{12}^{(1)}$

$(-30 + 20 \cdot 1 + 20 \cdot 0)$

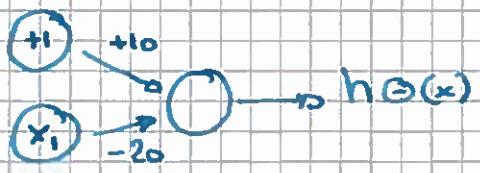


AND $\Rightarrow$

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	$g(10) \approx 1$

x_1 AND x_2

Negation: NOT x_1

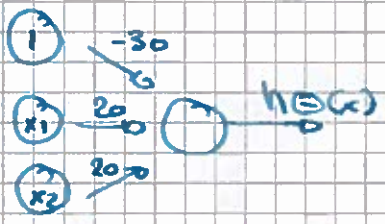


x_1	$h_\Theta(x)$
0	$g(10) \approx 1$
1	$g(-10) \approx 0$

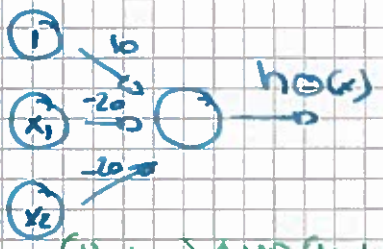
$$h_\Theta(x) = g(\Theta_{10} + \Theta_{11} x_1)$$

$$g(+10 - 20x_1)$$

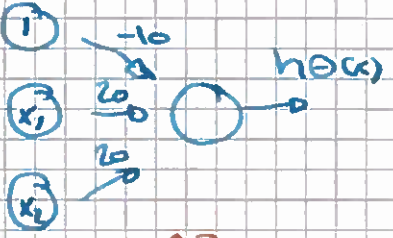
Putting it together: x_1 XNOR x_2



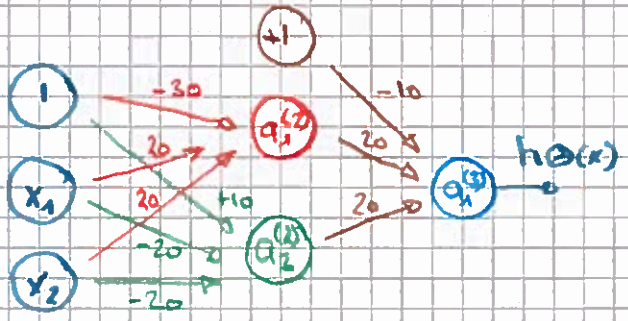
x_1 AND x_2



$(\text{Not } x_1)$ AND $(\text{Not } x_2)$



x_1 OR x_2

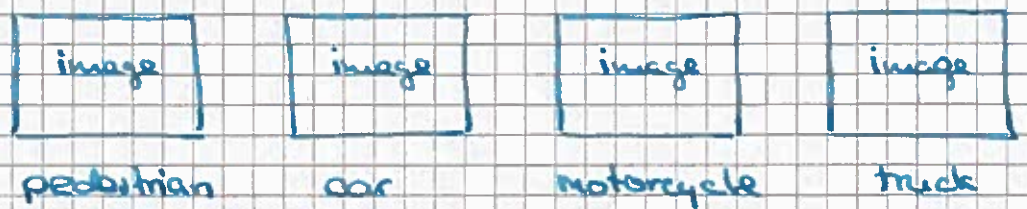


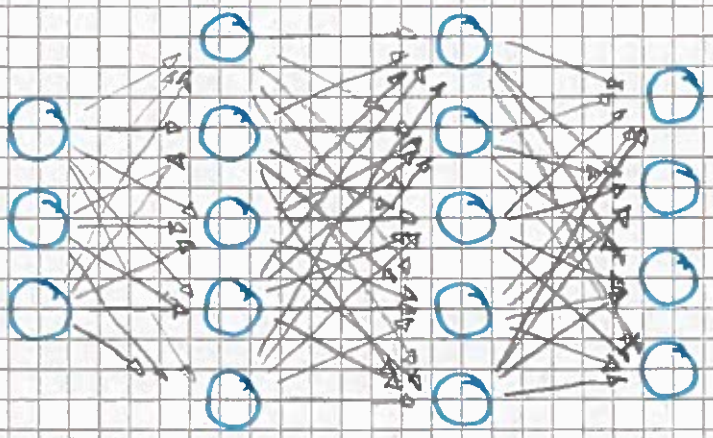
$\overline{x_1}$ AND $\overline{x_2}$

AND OR

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

Multiple output units: One vs. all





output would be a vector

$h_\Theta(x) \in \mathbb{R}^4$

want $h_\Theta(x)$ $\begin{bmatrix}1\\0\\0\\0\end{bmatrix}$ $h_\Theta(x)$ $\begin{bmatrix}0\\1\\0\\0\end{bmatrix}$ $h_\Theta(x)$ $\begin{bmatrix}0\\0\\1\\0\end{bmatrix}$ $h_\Theta(x)$ $\begin{bmatrix}0\\0\\0\\1\end{bmatrix}$

pedestrian car motorcycle truck

Training set $(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots (x^{(m)}, y^{(m)})$

$y^{(i)}$ one of $\begin{bmatrix}1\\0\\0\\0\end{bmatrix}, \begin{bmatrix}0\\1\\0\\0\end{bmatrix}, \begin{bmatrix}0\\0\\1\\0\end{bmatrix}, \begin{bmatrix}0\\0\\0\\1\end{bmatrix}$ $\Rightarrow (x^{(i)}, y^{(i)})$ ← image

pedestrian car motor-cycle truck

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

$y^{(i)} \in \mathbb{R}^4$ 4 dim. vectors