Week 4 - Notes

Neural - networks representation: Non-linear hypothesis

Consider a super-vised learning classification problem with training set as follows:

Non-linear classification

possibly apply logistic regression with heaps of non-linear features such as:

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

sigmoid function.

Logistic Regression is o.k for say ~ two features. But what happens if we have more?

x_1 = size
x_2 = # bedrooms
x_3 = # floors
x_4 = age.
...
100

$h = 100$

$x_1^2, x_1x_2, x_1x_3, x_1x_4 \dots x_1x_{100}$
x_2^2, x_2x_3

"called second order terms"

collecting all second order terms for the case when $h=100$ results in ≈ 5000 features. $\left(\approx \frac{n^2}{2}\right) = \mathcal{O}(n^2)$

- could overfit training set
- and be computationally expensive.

one thing you could do is say include:

→ $x_1^2, x_2^2, x_3^2, \dots, x_{100}^2$ so only ≈ 100

If you were to include 3rd order polynomial features

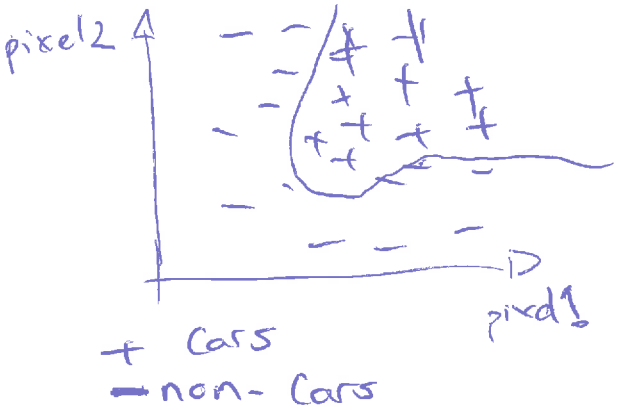
e.g $x_1x_2x_3, x_1^2x_2, x_{10}, x_{11}, x_{17} \dots$

$\mathcal{O}(n^3) \approx 170,000$.

Week 4 - notes

neural - networks representation : non-linear hypothesis

For too many machine learning problems n will be large.



lets say:

50 x 50 pixel images → 2500 pixels

n = 2500 (7500 in RGB)

$$x = \begin{bmatrix} \text{pixel 1 intensity} \\ \text{pixel 2 intensity} \\ \vdots \\ \text{pixel}_{2500} \text{ intensity} \end{bmatrix} \quad 0-255$$

All quadratic features $(x_i \times x_j) \approx$ 3 million features

- Logistic regression is not a good way to extract information when n is large.

Week 4 - notes

Neurons and the brain

Origins: Algorithms that try to mimic the brain.

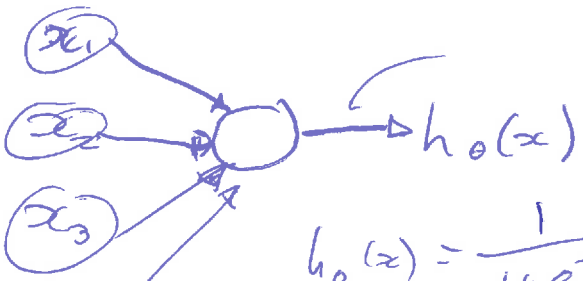
Neural Network Representation I

biological neuron

Dendrites = "input wires"

Axon = "output wires"

Neuron model: Logistic unit.



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

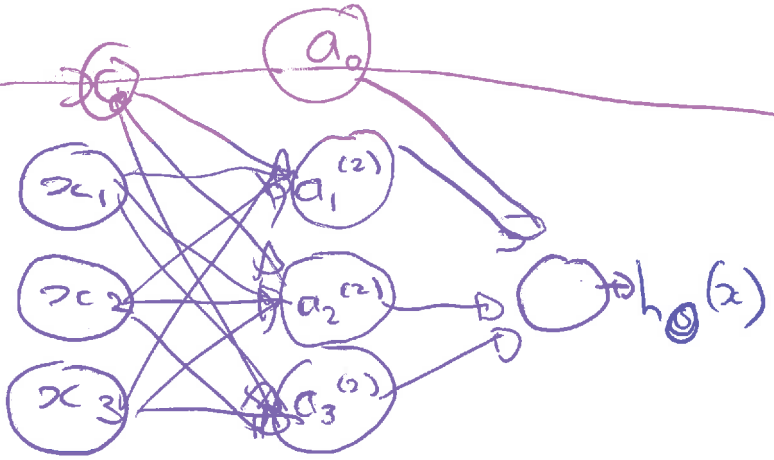
bias unit. x_0

Always ($x_0 = 1$)

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

Sigmoid (logistic) activation function

Also known as "weights" or parameters.



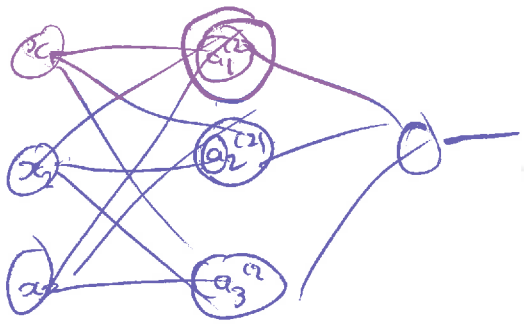
Layer 1 "input layer"

Layer 2 "hidden layer"

layer 3 "output layer"

Week 4 - notes

Neural Network Representation I



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

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

Sigmoid

$$a_1^{(2)} = g(\Theta_{10}^{(1)}x_0 + \Theta_{11}^{(1)}x_1 + \Theta_{12}^{(1)}a_2 + \Theta_{13}^{(1)}x_3)$$

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

$$a_3^{(2)} = g(\text{~~~~~~~~~~~~~~~~})$$

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

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

s_{j+1} ← subscript...

eg

0
0 0
0 0
0 0
input layer 1 — layer 2

$s_1 = 2$ $s_2 = 4$ ∴ $\Theta = 4 \times 3$

4 = number of

Week 4 – Notes

Neural Networks representation – Model representation I

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

$a_2^{(2)} = g(\ldots) \rightarrow z_2^{(2)}$ $\quad \therefore a_2^{(2)} = g(z_{\text{Ex2}}^{(2)})$

$a_3^{(2)} = g(\ldots) \rightarrow z_3^{(2)}$

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

$z^{(2)} = \Theta^{(1)} x$ | step 1 (*)

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

$a^{(2)} \in \mathbb{R}^3$ $\quad z^{(2)} \in \mathbb{R}^3$ (3 dimensional vector)

previous output from layer 1.

add bias step 3

then to Add the bias neuron Add $a_0^{(2)} = 1 \rightarrow a^{(2)} \in \mathbb{R}^4$

repeat of step 1 e 2 (from previous *)

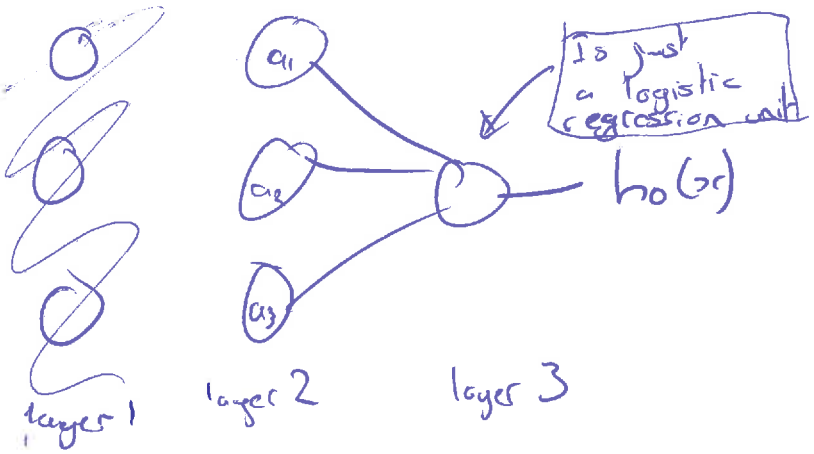
$z^{(3)} = \Theta^{(2)} a^{(2)}$

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

week 4 notes

Neural Network representation II

- Neural network learning its own features.



$$h_\theta(x) = g(\Theta_0^{(2)} a_0^{(2)} + \Theta_1^{(2)} a_1^{(2)} + \Theta_2^{(2)} a_2^{(2)} + \Theta_3^{(2)} a_3^{(2)})$$

* just the same as logistical regression but instead of using the features x_1, x_2, x_3 we are ~~so~~ now using $a_1^{(2)}, a_2^{(2)}, a_3^{(2)}$

---

the 2 new instead of feeding the features $x_1, x_2 x_3$ into logistical regression, the network gets to learn its own parameters a_1, a_2, a_3 to do it.

---

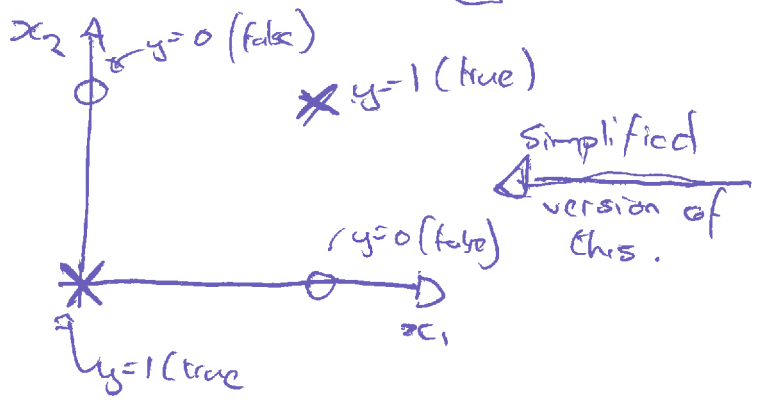
other network architecture

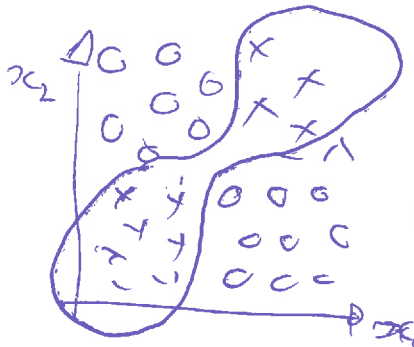
Week 4 - Notes.

Neural Networks representation - examples e intuitions I

Non-linear classification example: XOR/XNOR.

→ x_1, x_2 are binary (0 or 1)



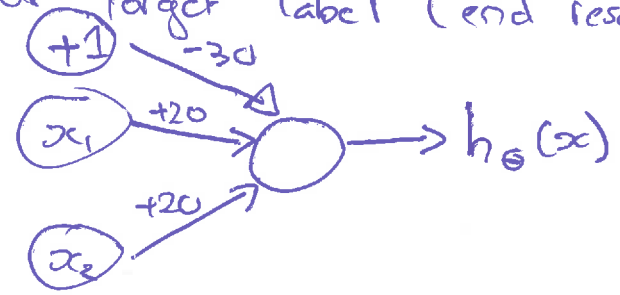


What we would like to do is learn a non-linear boundary, that seperates the positive and the negative examples.

Simple example "AND"

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

our target label (end result) is: $y = x_1$ AND x_2



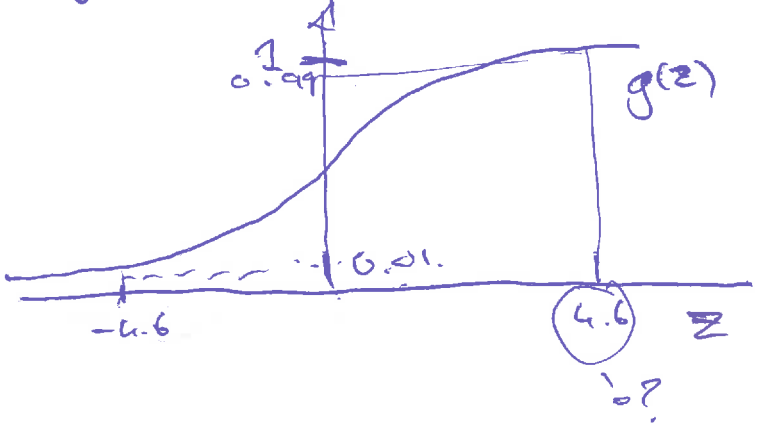
$$\begin{bmatrix} -10 & +20 & +20 \\ \text{FOR} & \downarrow & \end{bmatrix} \quad x_1 \quad x_2$$

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

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

$\therefore h_\Theta(x) \simeq x_1$ AND x_2. Logical AND function

Sigmoid function:



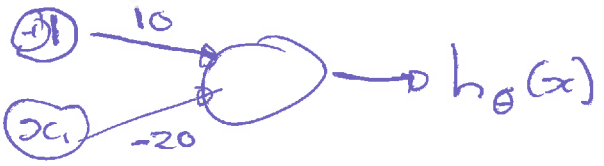
AND function

x_1	x_2	$h_\Theta(x)$
0	0	$g(-30) \simeq 0$
0	1	$g(-10) \simeq 0$ (very close to 0)
1	0	$g(-10) \simeq 0$
1	1	$g(10) \simeq 1$.

Week 4 - Notes

Neural networks representation - examples and intuitions II

negation / NOT x_1



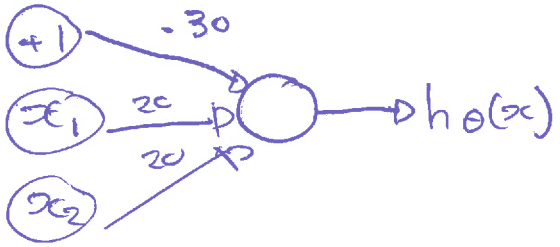
x_1	$h_\theta(x)$
0	$g(10) \approx 1$
1	$g(-10) \approx 0$

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

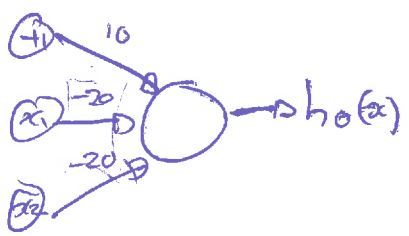
(NOT x_1) AND (NOT x_2)

(=1 if and only if $x_1 = x_2 = 0$)

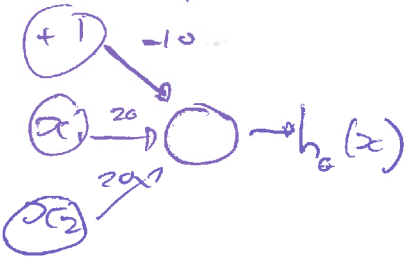
Putting it together: x_1 XNOR x_2 → looks like



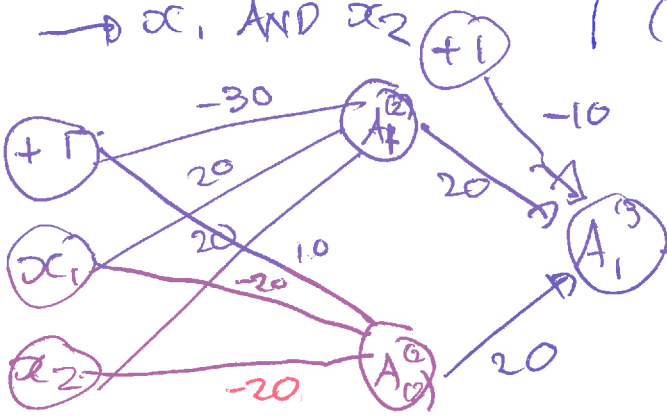
→ x_1 AND x_2



(Not x_1) AND (NOT x_2)



→ x_1 OR x_2



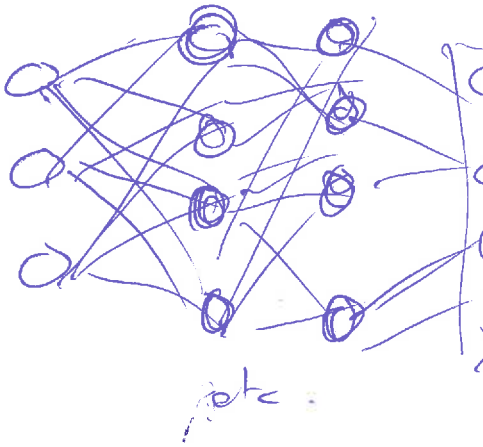
x_1	x_2	$a_1^{(2)}$	$a_2^{(2)}$	$h_\theta(x)$
0	0	0	1	1
0	1	0	0	0
1	0	0	0	0
1	1	1	0	1

Week 4 - Notes

Multiclass classification

Neural networks representation - ~~Model representation~~

example: images of
- pedestrian
- car
- motorcycle
- truck



choose 4 outputs

∴ we want

$h_\theta(x) = \begin{bmatrix}1\\0\\0\\0\end{bmatrix}$ → pedestrian

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

$h_\theta(x) = \begin{bmatrix}0\\0\\1\\0\end{bmatrix}$ → motorcycle

etc

---

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

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

pedestrian, car, motorcycle, truck

previously

$y = \{1, 2, 3, 4\}$

now representing binarily...