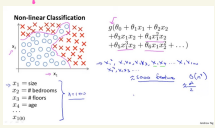
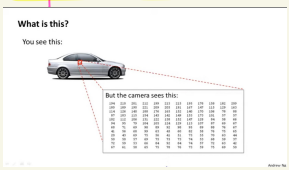
Need of NN:

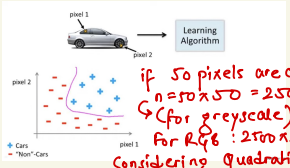
- Consider a classification problem (binary) that is not linear.
- We use polynomial logistic regression for drawing a decision boundary.
- From the fig. it can be seen that a poly. decision boundary can be found.
- ★ But what if $p > 2$, like $p = 100$
- The no. of parameters will be more than 100 (4950 considering interaction with degree = 2)
- This will cause overfitting.



+ NN is majorly used in "Computer Vision"
+ An object is viewed as a matrix of 'pixel intensity values' and trained using these matrices.

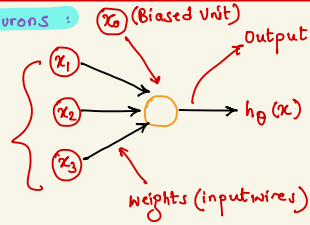


+ Learning Algorithm (CV):
- Pixels with same intensity are fed to the model. (Positive for cars, negative for other objects)
- Then the separation (decision) boundary is found for classification.



if 50 pixels are consi.
$n = 50 \times 50 = 2500$ features
↳ (for greyscale)
for RGB : $2500 \times 3 = 7500$
Considering Quadratic features: 3 million!!

Neurons:



★★★

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

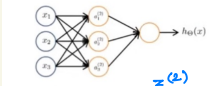
Sigmoid / Activation function:

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

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

→ weights (parameters) like $\beta_0, \beta_1, \ldots \beta_p$

Forward propagation: Vectorized implementation



$$a_1^{(2)} = g(\Theta_{10}^{(1)} x_0 + \Theta_{11}^{(1)} x_1 + \Theta_{12}^{(1)} x_2 + \Theta_{13}^{(1)} x_3)$$
$$a_2^{(2)} = g(\Theta_{20}^{(1)} x_0 + \Theta_{21}^{(1)} x_1 + \Theta_{22}^{(1)} x_2 + \Theta_{23}^{(1)} x_3)$$
$$a_3^{(2)} = g(\Theta_{30}^{(1)} x_0 + \Theta_{31}^{(1)} x_1 + \Theta_{32}^{(1)} x_2 + \Theta_{33}^{(1)} x_3)$$
$$h_\Theta(x) = g(\Theta_{10}^{(2)} a_0^{(2)} + \Theta_{11}^{(2)} a_1^{(2)} + \Theta_{12}^{(2)} a_2^{(2)} + \Theta_{13}^{(2)} a_3^{(2)})$$

$z_1^{(2)}$, $z_2^{(2)}$, $z_3^{(2)}$

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

$z^{(2)} = \Theta^{(1)} x$ (2D) ★

and $a^{(2)} = g(z^{(2)})$ --- (3D)

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

$\Rightarrow a^{(2)}$ = 4D ★

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

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

Andrew Ng

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

$\Rightarrow$ Similarly, $a_2^{(2)} = g(z_2^{(2)})$

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

"Forward Propagation"