



deeplearning.ai

# Multi-class classification

---

## Trying a softmax classifier

# Understanding softmax

(4,1)

$$z^{[L]} = \begin{bmatrix} 5 \\ 2 \\ -1 \\ 3 \end{bmatrix} \quad t = \begin{bmatrix} e^5 \\ e^2 \\ e^{-1} \\ e^3 \end{bmatrix}$$

$C=4$

$g^{[L]}(\cdot)$

"soft max"

$$a^{[L]} = g^{[L]}(z^{[L]}) = \begin{bmatrix} e^5 / (e^5 + e^2 + e^{-1} + e^3) \\ e^2 / (e^5 + e^2 + e^{-1} + e^3) \\ e^{-1} / (e^5 + e^2 + e^{-1} + e^3) \\ e^3 / (e^5 + e^2 + e^{-1} + e^3) \end{bmatrix} = \begin{bmatrix} 0.842 \\ 0.042 \\ 0.002 \\ 0.114 \end{bmatrix}$$

"hard max"

$$\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Softmax regression generalizes logistic regression to  $C$  classes.

If  $C=2$ , softmax reduces to logistic regression.  $a^{[L]} = \begin{bmatrix} 0.842 \\ 0.158 \end{bmatrix}$

# Loss function

## Loss function

云课堂

$C=4$

$(4,1)$   
 $y^{(1)} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$  - cat  
 $y_2 = 1$   
 $y_1 = y_3 = y_4 = 0$

$(4,1)$   
 $\hat{y}^{(1)} = \begin{bmatrix} 0.3 \\ 0.2 \\ 0.1 \\ 0.4 \end{bmatrix}$

$\mathcal{L}(\hat{y}, y) = - \sum_{j=1}^C y_j \log \hat{y}_j$   
small

$\mathcal{J}(w^{(1)}, b^{(1)}, \dots) = \frac{1}{m} \sum_{i=1}^m \mathcal{L}(\hat{y}^{(i)}, y^{(i)})$

$- y_2 \log \hat{y}_2 = - \log \hat{y}_2$

Make  $\hat{y}_2$  big.

$Y = [y^{(1)} \ y^{(2)} \ \dots \ y^{(m)}]$

$= \begin{bmatrix} 0 & 0 & 1 & \dots \\ 1 & 0 & 0 & \dots \\ 0 & 1 & 0 & \dots \\ 0 & 0 & 0 & \dots \end{bmatrix}$

$(4, m)$

$\hat{Y} = [\hat{y}^{(1)} \ \dots \ \hat{y}^{(m)}]$

$= \begin{bmatrix} 0.3 & \dots \\ 0.2 & \dots \\ 0.1 & \dots \\ 0.4 & \dots \end{bmatrix}$

$(4, m)$

Andrew Ng

# Summary of softmax classifier

