deeplearning.ai

Multi-class classification

Trying a softmax classifier

Understanding softmax

$z^{[L]}$ (4,1) $= \begin{bmatrix} 5 \\ 2 \\ -1 \\ 3 \end{bmatrix}$ $t = \begin{bmatrix} e^5 \\ e^2 \\ e^{-1} \\ e^3 \end{bmatrix}$

"soft max"

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

$C = 4$ $g^{[L]}(\cdot)$

"hard max"

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

Softmax regression generalizes logistic regresson to C classes.

If C = 2, softmax reduces to logistic regression.

Andrew Ng

Loss function

$y^{(1)} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$ $\hat{y}^{(1)} = \begin{bmatrix} 0.3 \\ 0.2 \\ 0.1 \\ 0.4 \end{bmatrix}$ $C = 4$

$$\mathcal{L}(\hat{y}, y) = -\sum_{i=1}^{4} y_i \log \hat{y}_i$$

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

make $\hat{y}_2$ bigger, $\mathcal{L}\downarrow$

$$J(W^{[1]}, b^{[1]}, \ldots) = \frac{1}{m} \sum_{i=1}^{m} \mathcal{L}(\hat{y}^{(i)}, y^{(i)})$$

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

$(4, m)$

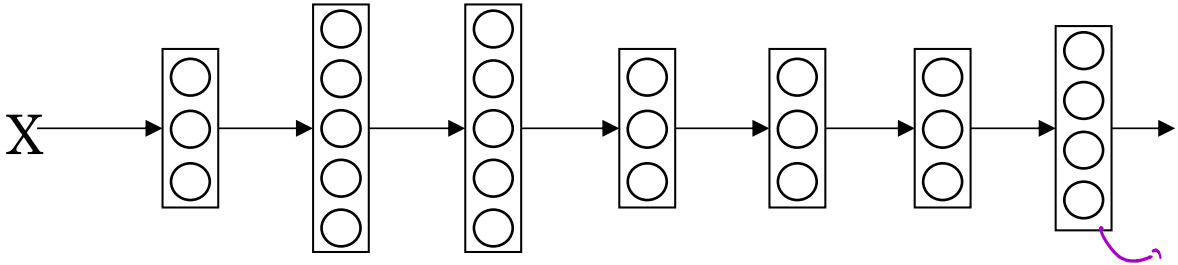
$= \begin{bmatrix} 0 & 1 & \ldots \\ 1 & 0 & \\ 0 & 0 & \\ 0 & 0 & \end{bmatrix}$

$\hat{Y} = [\hat{y}^{(1)}, \hat{y}^{(2)}, \ldots \hat{y}^{(m)}]$

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

Andrew Ng

Summary of softmax classifier



$z^{[L]}$ $\rightarrow$ $a^{[L]} = \hat{y}$ $\rightarrow$ $L(\hat{y}, y)$

$(4,1)$

Backprop:

$dz^{[L]} = \hat{y} - y$

$\frac{\partial J}{\partial z^{[L]}}$