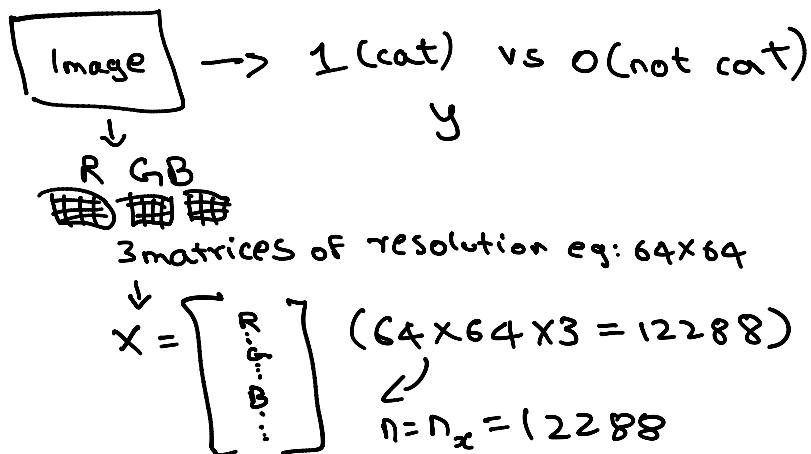


# Week 2a - Logistic regression as a Neural Network



Thursday, July 30, 2020 10:23 PM

## ① Binary classification



→ Notation

Single training set represented by  $(x, y)$  where  $x \in \mathbb{R}^{n_x}$ ,  $y \in \{0, 1\}$

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

$m = m_{\text{train}}$        $m_{\text{test}}$   
↑  
training set      test set

$$X = \left[ \begin{array}{c|c|c|c} 1 & x^{(1)} & x^{(2)} & \cdots & x^{(m)} \\ \hline x^{(1)} & | & | & \cdots & | \\ \hline \vdots & \vdots & \vdots & \ddots & \vdots \\ \hline \end{array} \right] \quad n_x$$

$$X \in \mathbb{R}^{n_x \times m} \quad X\text{-shape} = (n_x, m)$$

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

$$Y \in \mathbb{R}^{1 \times m}$$

$$Y\text{-shape} = (1, m)$$

Stacking as columns is easier to implement than rows

## ② Logistic Regression

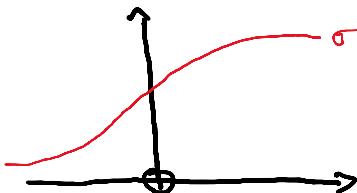
Given  $x$ , we want  $\hat{y} = P(y=1|x)$

$x \in \mathbb{R}^{n_x}$   $\hat{y}$  is the probability that  $y$  is 1 when  $x$  is given

Parameters:  $w \in \mathbb{R}^{n_x}$ ,  $b \in \mathbb{R}$

Output  $\hat{y} = w^T x + b$  (in case of linear regression)

$$\therefore \hat{y} = \sigma(w^T x + b) \quad \begin{array}{l} \text{sigmoid of} \\ w^T x + b \end{array} \quad \text{(Logistic regression)}$$



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

If  $z$  is large,

If  $z$  is small,  
 $\sigma(z) \approx 0$

Alt. convention  $x_0 = 1$ ,  $x \in \mathbb{R}^{n_x+1}$

$$\hat{y} = \sigma(\theta^T x)$$

$$\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_{n_x} \end{bmatrix} \rightarrow b$$

## ③ Logistic Regression Cost Function

→ Given  $\{(x^{(i)}, y^{(i)})\}, \dots, (x^{(m)}, y^{(m)})\}$ , we want  $\hat{y}^{(i)} \approx y^{(i)}$

$z^{(i)}$  can also be defined  $\rightarrow z^{(i)} = w^T x^{(i)} + b$

→ Loss (error) function:  $L(\hat{y}, y) = \frac{1}{2} (\hat{y} - y)^2$

↓ Mean squared error loss fn.  
results in non-convex graph

~~~~ → multiple minima for optimization problems

$$\rightarrow L(\hat{y}, y) = -(y \log \hat{y} + (1-y) \log(1-\hat{y}))$$

→ If  $y=1$ ,  $L(\hat{y}, y) = -\log \hat{y}$  as  $\hat{y} > 0$



$\rightarrow$  If  $y=1$ ,  $\mathcal{L}(\hat{y}, y) = -\log \hat{y}$   $\curvearrowleft$   
 ↳ as small as possible  $\Rightarrow \log \hat{y}$  is large  
 $\Rightarrow \hat{y}$  is large  
 $\rightarrow \hat{y} = 1$  (max. of  $\hat{y}$ )  
 $\rightarrow$  If  $y=0$ ,  $\mathcal{L}(\hat{y}, y) = -\log(1-\hat{y})$   $\Rightarrow \hat{y}=0$  (min. of  $\hat{y}$ )  
 ↳ as small as possible  $\Rightarrow \log(1-\hat{y})$  large  
 $\Rightarrow 1-\hat{y}$  is large  
 $\Rightarrow \hat{y}$  is small

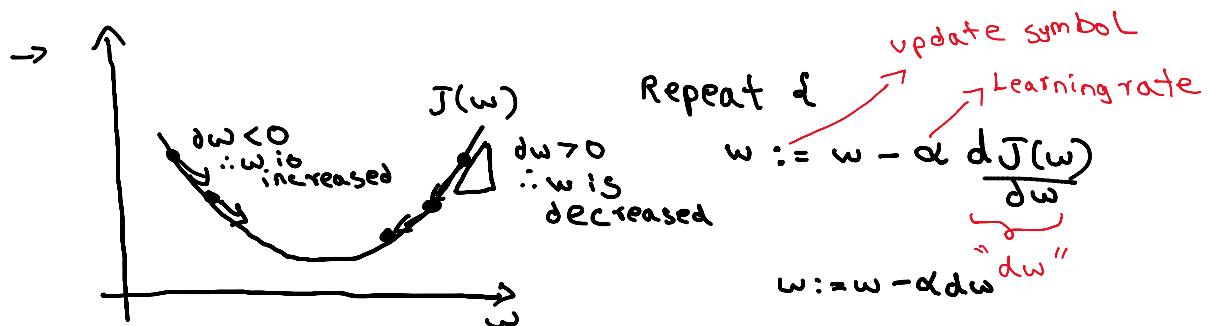
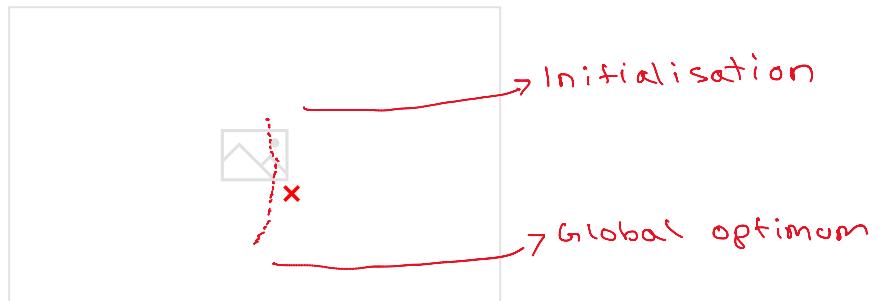
$\rightarrow$  Cost function:  $J(w, b) = \frac{1}{m} \sum_{i=1}^m \mathcal{L}(\hat{y}^{(i)}, y^{(i)})$

↳ average loss across all sets

$$J(w, b) = -\frac{1}{m} \sum_{i=1}^m [y^{(i)} \log \hat{y}^{(i)} + (1-y^{(i)}) \log(1-\hat{y}^{(i)})]$$

## ④ Gradient Descent

→ Want to find  $w, b$  that minimize  $(w, b)$



$$J(w, b) \quad w := w - \alpha \frac{\partial J(w, b)}{\partial w}$$

$$b := b - \alpha \frac{\partial J(w, b)}{\partial b}$$

$$\frac{\partial J(w, b)}{\partial w} \rightarrow \partial w$$

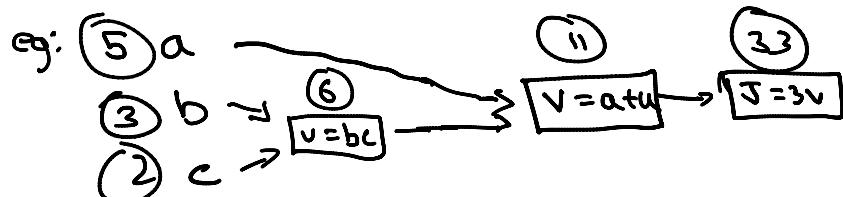
$$\frac{\partial J(w, b)}{\partial b} \rightarrow \partial b$$

## ⑤ Computation graph

Let's say we need to compute  
 $J(a, b, c) = 3(a + bc)$

Steps:

- i. Compute  $v = bc$
- ii. Compute  $v = a + v$
- iii. Compute  $J = 3v$



## ⑥ Derivatives with a Computation graph

$$\frac{\partial J}{\partial v} = ? \quad \left| \begin{array}{l} J = 3v \\ v = 11 \rightarrow 11.001 \\ J = 33 \rightarrow 33.003 \end{array} \right.$$

$$J = 3v \quad \frac{\partial J}{\partial v} = 3$$

$$\frac{\partial J}{\partial a} = ? \quad \left| \begin{array}{l} a = 5 \rightarrow 5.001 \\ v = 11 \rightarrow 11.001 \quad (\frac{\partial J}{\partial a} = 1) \\ J = 33 \rightarrow 33.003 \end{array} \right.$$

$$\frac{\partial J}{\partial a} = \frac{\partial J}{\partial v} \cdot \frac{\partial v}{\partial a}$$

$$\frac{\partial J}{\partial a} = 3$$

$$\frac{\partial J}{\partial u} \quad \left| \begin{array}{l} u = 6 \rightarrow 6.001 \\ v = 11 \rightarrow 11.001 \\ J = 33 \rightarrow 33.003 \end{array} \right.$$

$$\frac{\partial J}{\partial u} = \frac{\partial J}{\partial v} \cdot \frac{\partial v}{\partial u}$$

$$\frac{\partial J}{\partial u} = 3$$

$$\frac{\partial J}{\partial b} \quad \left| \begin{array}{l} b = 3 \rightarrow 3.001 \\ v = bc = 6 \rightarrow 6.002 \quad (\because c=2) \end{array} \right.$$

$$\frac{\partial J}{\partial b} = \frac{\partial J}{\partial v} \cdot \frac{\partial v}{\partial b} \quad \frac{\partial v}{\partial b} = 2 \Rightarrow \frac{\partial J}{\partial b} = 3 \cdot 2 = 6$$

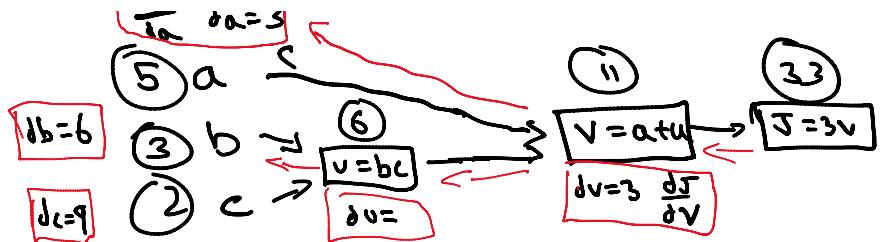
$$\frac{\partial J}{\partial c} \quad \left| \begin{array}{l} \frac{\partial J}{\partial c} = 0 \end{array} \right.$$

$$\frac{d(\text{Final Output Var})}{d(\text{var})}$$

→ represented as "dvar" in code

$$\frac{\partial J}{\partial a} \quad \frac{\partial a}{\partial a} = 3$$





⑦ Logistic regression gradient descent  
(for 1 training example)

$$Z = w^T x + b$$

$$\hat{y} = a = \sigma(z)$$

$$L(a, y) = -(y \log a + (1-y) \log(1-a))$$

eg:

Diagram illustrating the backpropagation of error for a linear regression model:

- Inputs:  $x_1, x_2$  (red arrows)
- Weights:  $w_1, w_2$
- Bias:  $b$
- Model:  $z = w_1 x_1 + w_2 x_2 + b$
- Activation:  $\hat{y} = a = \sigma(z)$
- Loss:  $l(a, y)$

Handwritten annotations:

$$\frac{\partial z}{\partial z} = \frac{\partial l}{\partial z} = \frac{\partial l(a, y)}{\partial z}$$

$$\frac{\partial z}{\partial a} = a(1-a)$$

$$\frac{\partial a}{\partial z} = \sigma'(z)$$

$$\frac{\partial l}{\partial w_1} = \frac{\partial l}{\partial a} \cdot \frac{\partial a}{\partial z} \cdot \frac{\partial z}{\partial w_1} = x_1 \delta z$$

$$\frac{\partial l}{\partial w_2} = \frac{\partial l}{\partial a} \cdot \frac{\partial a}{\partial z} \cdot \frac{\partial z}{\partial w_2} = x_2 \delta z$$

$$\frac{\partial l}{\partial b} = \delta z$$

$$\delta a = -\frac{y}{a} + \frac{1-y}{a}$$

$$\delta z = a(1-a)$$

$$\Rightarrow \delta w_1 = \frac{\partial l}{\partial w_1} = \alpha x_1 \delta z$$

$$\Rightarrow \delta w_2 = \frac{\partial l}{\partial w_2} = \alpha x_2 \delta z$$

$$\Rightarrow \delta b = \delta z = \alpha \delta z$$

$$\Rightarrow w_1 := w_1 - \alpha \delta w_1$$

$$\Rightarrow w_2 := w_2 - \alpha \delta w_2$$

$$\Rightarrow b := b - \alpha \delta b$$

⑧ Gradient descent on 'm' examples

$$J(\omega, b) = \frac{1}{m} \sum_{i=1}^m L(a^{(i)}, y)$$

$$a^{(i)} = \hat{y}^{(i)} = \sigma(z^{(i)}) = \sigma(\omega^T x^{(i)} + b)$$

$$(x^{(i)}, y^{(i)})$$

$\xrightarrow{\delta w_1^{(i)} \quad \delta w_2^{(i)} \quad \delta b^{(i)}}$

$\overleftarrow{\overrightarrow{n=2}}$

$$\frac{\partial}{\partial w_i} J(\omega, b) = \frac{1}{3} \sum_{i=1}^3 \underbrace{\frac{\partial}{\partial w_i} L(a^{(i)}, y^{(i)})}_{J(w_i^{(i)}) - (x^{(i)}, y^{(i)})}$$

- Algorithm

$\rightarrow$  Initialise  $\bar{J} = 0$ ;  $\delta w_1 = 0$ ;  $\delta w_2 = 0$ ;  $\delta b = 0$

$\rightarrow$  for  $i=1$  to  $m$        $\xrightarrow{\text{accumulated } \Delta w_1^{(i)} \text{ & } \Delta w_2^{(i)}}$

$\rightarrow$  for  $i=1$  to  $m$

$$z^{(i)} = w^T x^{(i)} + b$$

$$a^{(i)} = \sigma(z^{(i)})$$

$$J+ = -[y^{(i)} \log a^{(i)} + (1-y^{(i)}) \log(1-a^{(i)})]$$

$$\delta z^{(i)} = a^{(i)} - y^{(i)}$$

$$\delta w_1+ = x_1^{(i)} \delta z^{(i)}$$

$$\delta w_2+ = x_2^{(i)} \delta z^{(i)}$$

$$\delta b+ = \delta z^{(i)}$$

→ accumulated  $\delta w_1^{(i)}$  &  $\delta w_2^{(i)}$

↑ assuming  $n=2$  for this eg.  
else has to be looped  
 $n$  times

$\rightarrow J/m;$

$\delta w_1/m;$

$\delta w_2/m;$

$\delta b/m;$

$\rightarrow w_1 := w_1 - \alpha \delta w_1$

$w_2 := w_2 - \alpha \delta w_2$

$b := b - \alpha \delta b$

- vectorization can be done to eliminate multiple loops.