

Join GitHub today

[Dismiss](#)

GitHub is home to over 40 million developers working together to host and review code, manage projects, and build software together.

[Sign up](#)

Branch: master ▼

[Find file](#)[Copy path](#)

[deep-learning-specialization-coursera](#) / [01-Neural-Networks-and-Deep-Learning](#) / [week2](#) / [01-Logistic-Regression-as-a-Neural-Network.ipynb](#)



andersy005 Update Logistic Regression notebook

d137c4b on Aug 12, 2017

[1 contributor](#)

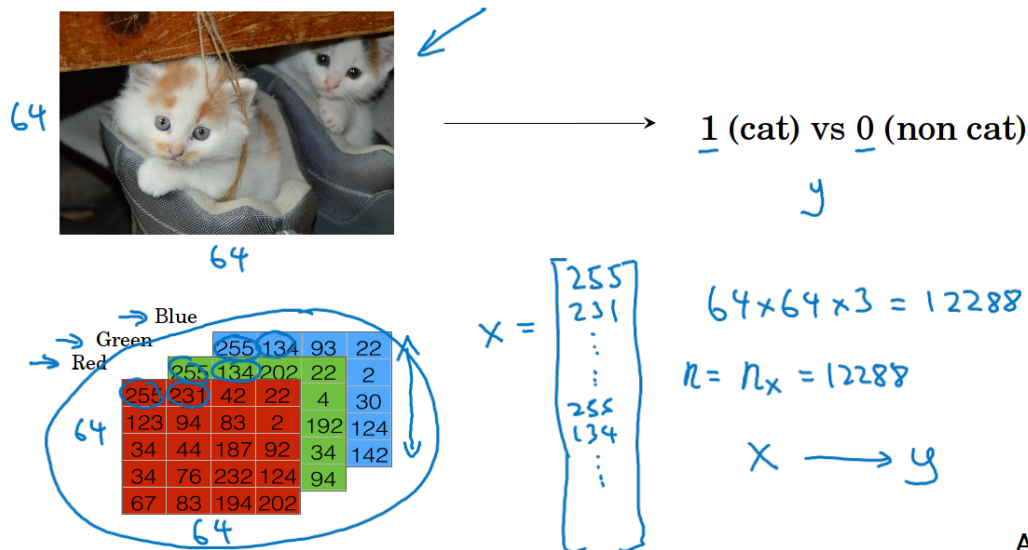
[Raw](#)[Blame](#)[History](#)

218 lines (218 sloc) 5.37 KB

Table of Contents

- 1 Binary Classification
- 2 Logistic Regression
- 3 Logistic Regression Cost Function
- 4 Gradient Descent
- 5 Computational Graph
- 6 Logistic Regression Gradient Descent
- 7 Logistic Regression on m examples

Binary Classification



Andrew Ng

Notation

$(x, y) \quad x \in \mathbb{R}^{n_x}, y \in \{0, 1\}$

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

$M = M_{\text{train}} \quad M_{\text{test}} = \# \text{test examples.}$

$X = \begin{bmatrix} | & | & | & | \\ x^{(1)} & x^{(2)} & \dots & x^{(m)} \\ | & | & | & | \end{bmatrix}$

$X \in \mathbb{R}^{n_x \times m}$

$X \cdot \text{shape} = (n_x, m)$

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

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

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

Andrew Ng

```
In [2]: from IPython.display import IFrame
        IFrame("standard-notation.pdf", width=700, height=500)
```

Out[2]:

Logistic Regression

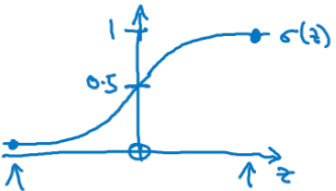
Logistic Regression

Given x , want $\hat{y} = \frac{P(y=1|x)}{0 \leq \hat{y} \leq 1}$

$x \in \mathbb{R}^{n_x}$

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

Output $\hat{y} = \sigma(\underbrace{w^T x + b}_z)$



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

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

$\Theta = \begin{bmatrix} \Theta_0 \\ \Theta_1 \\ \Theta_2 \\ \vdots \\ \Theta_{n_x} \end{bmatrix}$ $\left\{ \begin{array}{l} b \leftarrow \Theta_0 \\ w \leftarrow \begin{bmatrix} \Theta_1 \\ \Theta_2 \\ \vdots \\ \Theta_{n_x} \end{bmatrix} \end{array} \right.$

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

If z large $\sigma(z) \approx \frac{1}{1+0} = 1$

If z large negative number

$\sigma(z) = \frac{1}{1+e^{-z}} \approx \frac{1}{1+\text{Big number}} \approx 0$

Andrew Ng

Logistic Regression Cost Function

Logistic Regression cost function

$\rightarrow \hat{y}^{(i)} = \sigma(w^T x^{(i)} + b)$, where $\sigma(z^{(i)}) = \frac{1}{1+e^{-z^{(i)}}}$ $z^{(i)} = w^T x^{(i)} + b$

Given $\{(x^{(1)}, y^{(1)}), \dots, (x^{(m)}, y^{(m)})\}$, want $\hat{y}^{(i)} \approx y^{(i)}$. $\begin{matrix} x^{(i)} \\ y^{(i)} \\ z^{(i)} \end{matrix}$ i -th example.

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

$\mathcal{L}(\hat{y}, y) = -[y \log \hat{y} + (1-y) \log(1-\hat{y})]$

If $y=1$: $\mathcal{L}(\hat{y}, y) = -\log \hat{y} \leftarrow$ Want $\log \hat{y}$ large, want \hat{y} large.

If $y=0$: $\mathcal{L}(\hat{y}, y) = -\log(1-\hat{y}) \leftarrow$ Want $\log(1-\hat{y})$ large \dots want \hat{y} small

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

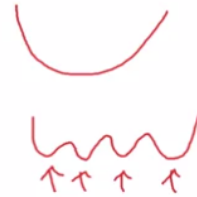
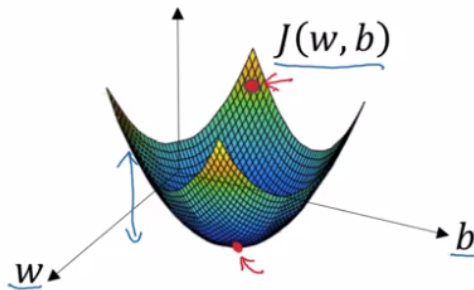
Gradient Descent

Gradient Descent

Recap: $\hat{y} = \sigma(w^T x + b)$, $\sigma(z) = \frac{1}{1+e^{-z}}$ ←

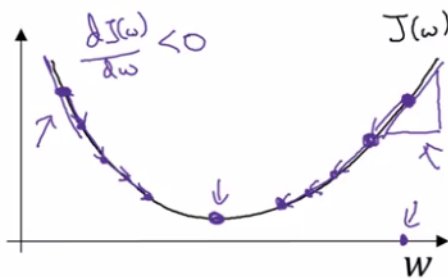
$$J(w, b) = \frac{1}{m} \sum_{i=1}^m \mathcal{L}(\hat{y}^{(i)}, y^{(i)}) = -\frac{1}{m} \sum_{i=1}^m y^{(i)} \log \hat{y}^{(i)} + (1 - y^{(i)}) \log(1 - \hat{y}^{(i)})$$

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



Andrew Ng

Gradient Descent



Repeat {
 $w := w - \alpha \frac{dJ(w)}{dw}$
 }
 $w := w - \alpha dw$

learning rate
 α
 $\frac{dJ(w)}{dw}$
 dw

$$\frac{dJ(w)}{dw} = ?$$

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

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

partial derivative
 ∂

Computational Graph

Computation Graph

$$J(a, b, c) = 3(a + bc) = 3(5 + 3 \times 2) = 33$$

$$u = bc$$

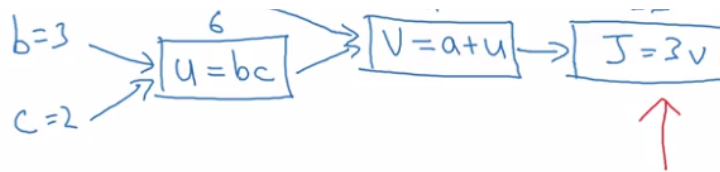
$$v = a + u$$

$$J = 3v$$

$$a = 5$$

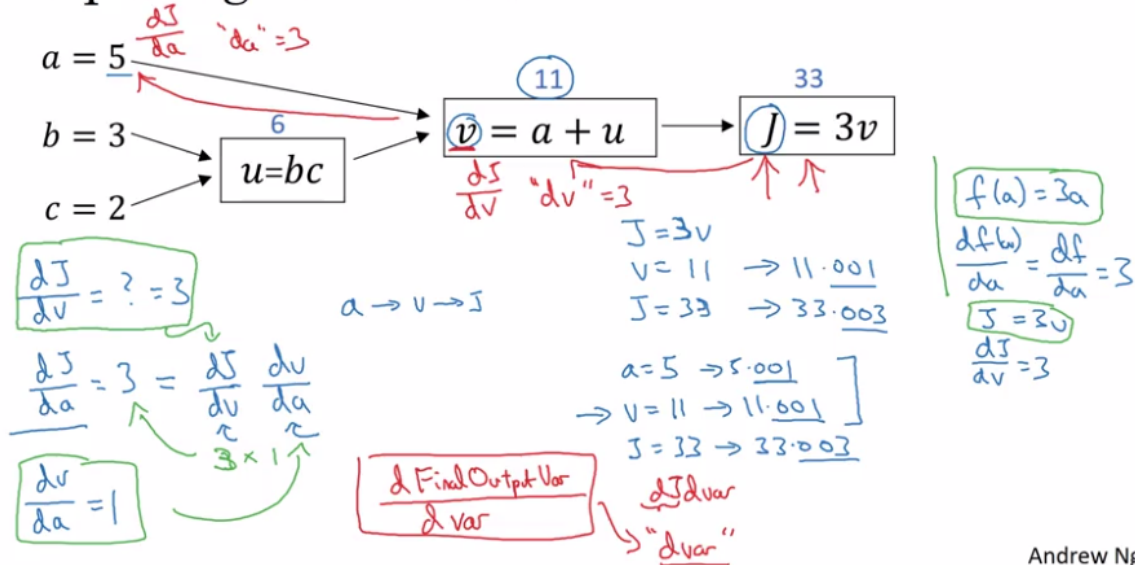
11

33



Andrew Ng

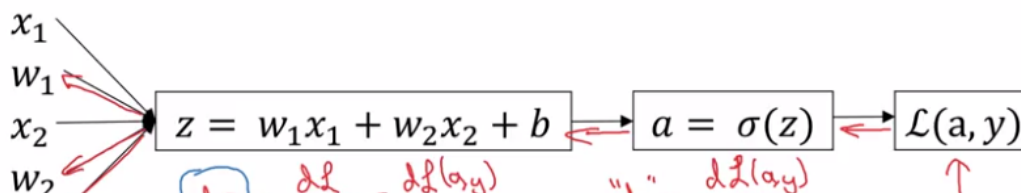
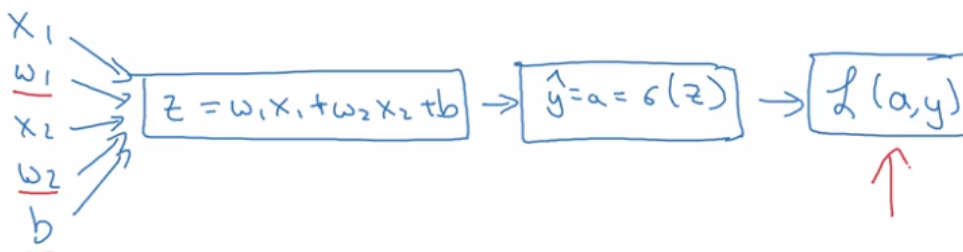
Computing derivatives



Andrew Ng

Logistic Regression Gradient Descent

- $\rightarrow z = w^T x + b$
- $\rightarrow \hat{y} = a = \sigma(z)$
- $\rightarrow \mathcal{L}(a, y) = -(y \log(a) + (1 - y) \log(1 - a))$



b

$$\frac{\partial \mathcal{L}}{\partial z} = \frac{\partial \mathcal{L}}{\partial a} \cdot \frac{da}{dz}$$

$$= (a - y) \cdot a(1-a)$$

$$= \frac{\partial \mathcal{L}}{\partial a} \cdot \frac{da}{dz}$$

$$\frac{da}{dz} = \frac{da}{da} \cdot \frac{da}{dz} = 1 \cdot a(1-a) = a(1-a)$$

$$\frac{\partial \mathcal{L}}{\partial w_1} = \frac{d\mathcal{L}}{dw_1} = x_1 \cdot dz$$

$$dw_2 = x_2 \cdot dz$$

$$db = dz$$

$$w_1 := w_1 - \alpha \frac{d\mathcal{L}}{dw_1}$$

$$w_2 := w_2 - \alpha \frac{d\mathcal{L}}{dw_2}$$

$$b := b - \alpha \frac{d\mathcal{L}}{db}$$

Logistic Regression on m examples

$J=0; \frac{dw_1}{dz}=0; \frac{dw_2}{dz}=0; \frac{db}{dz}=0$

→ For $i=1$ to m

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

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

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

$$\frac{dz^{(i)}}{dz} = a^{(i)} - y^{(i)}$$

$$\frac{dw_1}{dz} += x_1^{(i)} \frac{dz^{(i)}}{dz}$$

$$\frac{dw_2}{dz} += x_2^{(i)} \frac{dz^{(i)}}{dz}$$

$$\frac{db}{dz} += \frac{dz^{(i)}}{dz}$$

$n=2$

$J/=m$

$\frac{dw_1}{dz} /= m; \frac{dw_2}{dz} /= m; \frac{db}{dz} /= m.$

$$\frac{dw_1}{dz} = \frac{\partial J}{\partial w_1}$$

$$w_1 := w_1 - \alpha \frac{dw_1}{dz}$$

$$w_2 := w_2 - \alpha \frac{dw_2}{dz}$$

$$b := b - \alpha \frac{db}{dz}$$

Vectorization

