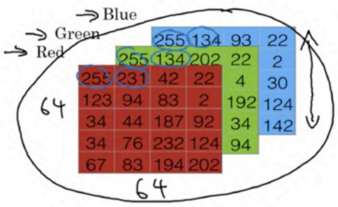
Binary Classification

64

64

$\underline{1}$ (cat) vs $\underline{0}$ (non cat)

y



$$x = \begin{bmatrix} 255 \\ 231 \\ \vdots \\ 255 \\ 134 \\ \vdots \\ \\ \end{bmatrix}$$

$64 \times 64 \times 3 = 12288$

$n = n_x = 12288$ Dimension of input features

$x \longrightarrow y$

Andrew Ng

Notation

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

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

$M = M_{train}$

$M_{test} = \#\text{test examples.}$

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

$\longleftarrow m \longrightarrow$

Stack training set inputs into columns

$$\cancel{\begin{bmatrix} - x^{(1)T} - \\ \vdots \\ - x^{(m)T} - \end{bmatrix}}$$

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

Stack Y in columns also

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

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

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

Andrew Ng

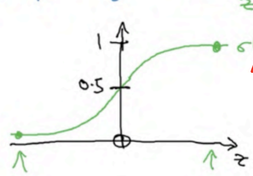
Logistic Regression

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

$x \in \mathbb{R}^{n_x}$ $\quad 0 \leq \hat{y} \leq 1$

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

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



Want value between 0 & 1, so sigmoid

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

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

$$\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \vdots \\ \theta_{n_x} \end{bmatrix} \begin{matrix} \} b \leftarrow \\ \\ \} w \leftarrow \\ \\ \end{matrix}$$

Don't use this notation in this course

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

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

If z large negative number

$\sigma(z) = \dfrac{1}{1 + e^{-z}} \approx \dfrac{1}{1 + \text{Bignum}} \approx 0$

Andrew Ng

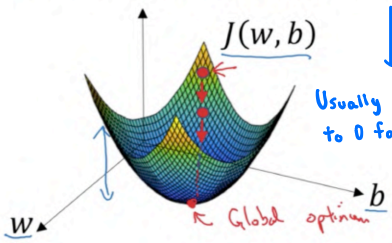
Gradient Descent

Cost Function

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

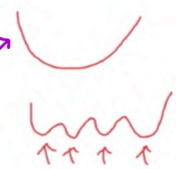
$$J(w,b) = \frac{1}{m}\sum_{i=1}^{m} \mathcal{L}(\boxed{\hat{y}^{(i)}}, \boxed{y^{(i)}}) = -\frac{1}{m}\sum_{i=1}^{m} y^{(i)} \log \hat{y}^{(i)} + (1 - y^{(i)}) \log(1 - \hat{y}^{(i)})$$

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



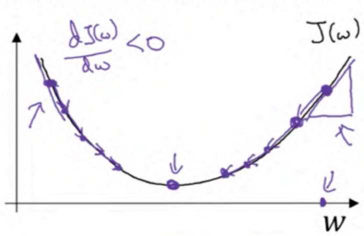
Usually initialize w, b to 0 for logistic regressions

Convex



Andrew Ng

Gradient Descent



learning rate

Repeat {

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

}

"dw"

$w := w - \alpha dw$

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

$J(w,b)$

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

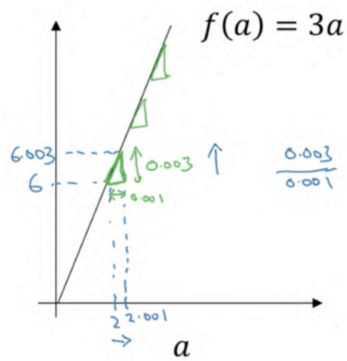
$b := b - \alpha \boxed{\frac{dJ(w,b)}{db}}$ $\boxed{\frac{\partial J(w,b)}{\partial b}} \rightarrow db$

∂ "partial derivative"

d J

Andrew Ng

Intuition about derivatives a.k.a. slope



$\frac{0.003}{0.001}$ $\frac{\text{height}}{\text{width}}$

→ $a = 2$ $f(a) = 6$

$a = 2.001$ $f(a) = 6.003$

slope (derivative) of $f(a)$ at $a = 2$ is 3

→ $a = 5$ $f(a) = 15$

$a = 5.001$ $f(a) = 15.003$

slope at $a = 5$ is also 3

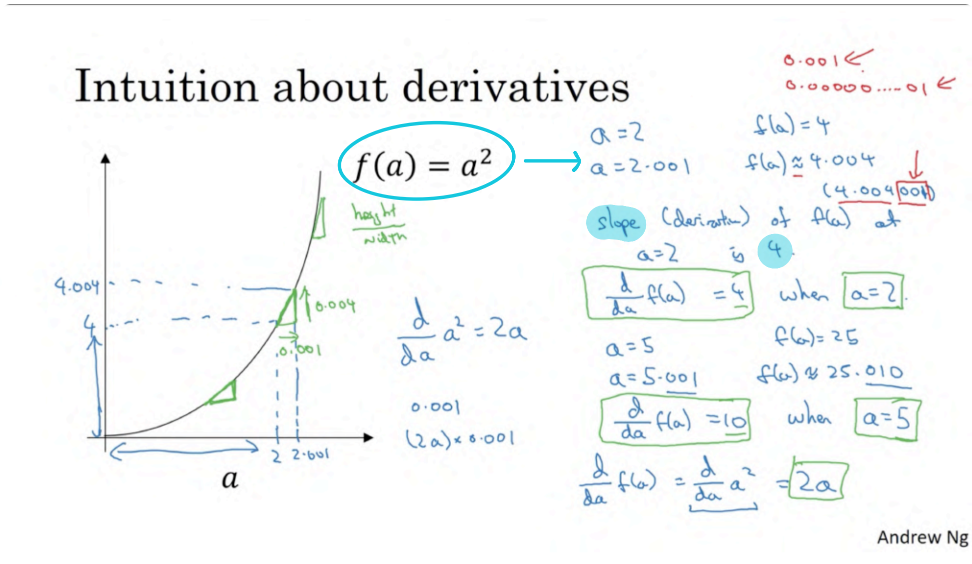
$\frac{df(a)}{da} = 3 = \frac{d}{da} f(a)$

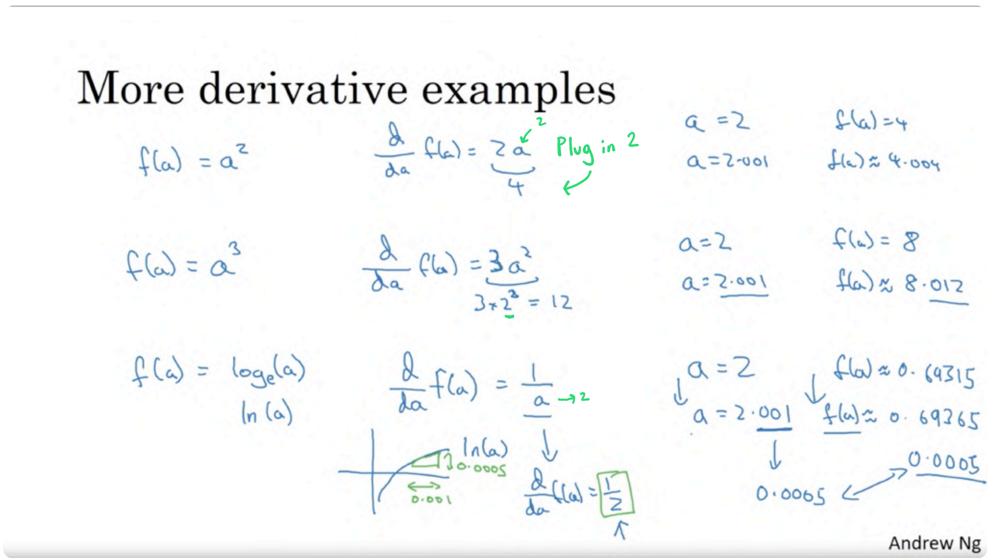
0.001

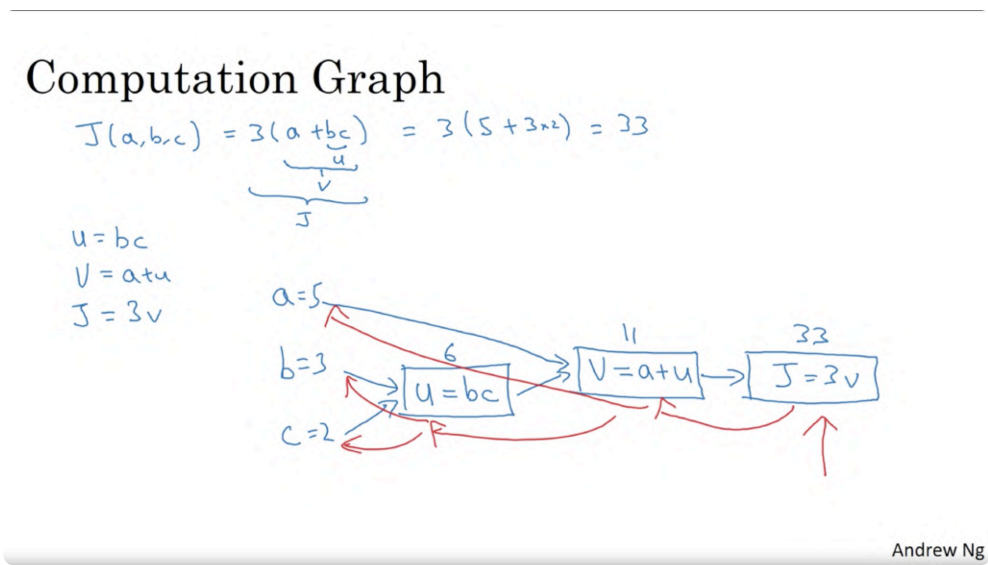
0.00000001

0.000000001

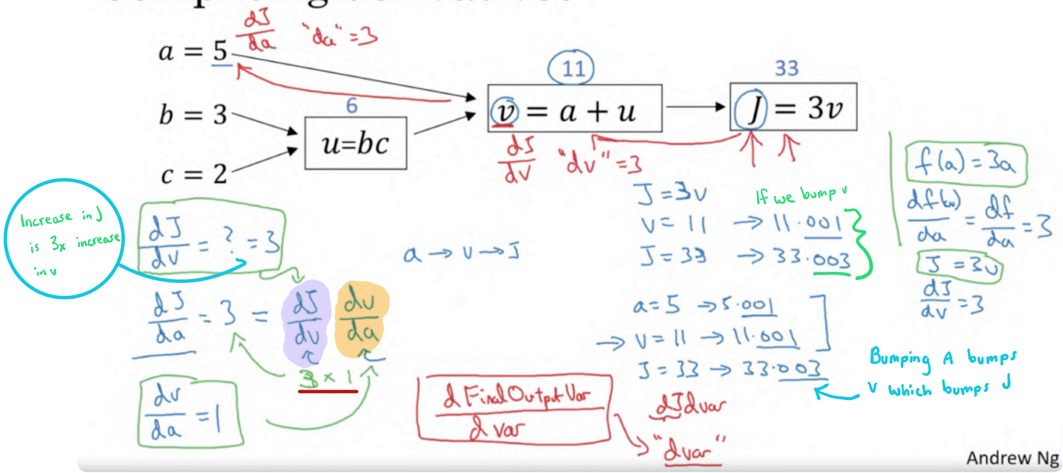
Andrew Ng


Intuition about derivatives
f(a) = a²
a = 2   f(a) = 4
a = 2.001   f(a) ≈ 4.004
slope (derivative) of f(a) at a = 2 is 4
d/da f(a) = 4 when a = 2
a = 5   f(a) = 25
a = 5.001   f(a) ≈ 25.010
d/da f(a) = 10 when a = 5
d/da f(a) = d/da a² = 2a
d/da a² = 2a
0.001
(2a) × 0.001
height / width
4.004
0.004
0.001
2 2.001
a
Andrew Ng

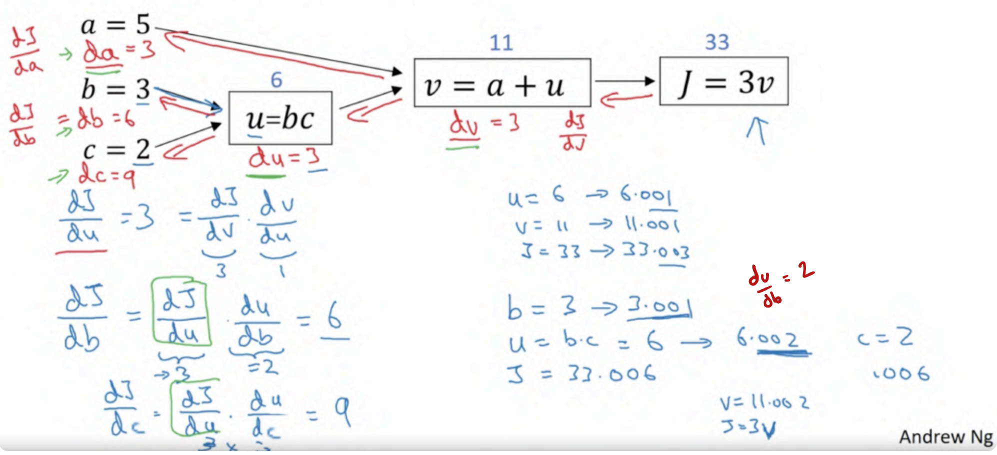

More derivative examples
f(a) = a²   d/da f(a) = 2a   Plug in 2
a = 2   f(a) = 4
a = 2.001   f(a) ≈ 4.004
f(a) = a³   d/da f(a) = 3a²   3×2² = 12
a = 2   f(a) = 8
a = 2.001   f(a) ≈ 8.012
f(a) = logₑ(a)   ln(a)   d/da f(a) = 1/a
a = 2   f(a) ≈ 0.69315
a = 2.001   f(a) ≈ 0.69365
0.0005
d/da f(a) = 1/2
Andrew Ng


Computation Graph
J(a,b,c) = 3(a + bc) = 3(5 + 3×2) = 33
u = bc
v = a+u
J = 3v
a = 5
b = 3
c = 2
u = bc
v = a+u
J = 3v
6
11
33
Andrew Ng

Computing derivatives



Computing derivatives

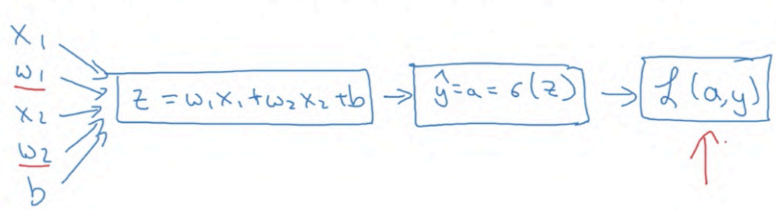


Logistic regression recap

$$\rightarrow z = w^T x + b$$

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

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



Andrew Ng

Logistic regression derivatives

$$x_1,\ w_1,\ x_2,\ w_2,\ b \rightarrow \boxed{z = w_1x_1 + w_2x_2 + b} \rightarrow \boxed{a = \sigma(z)} \rightarrow \boxed{\mathcal{L}(a, y)}$$

derivative of sigmoid

$$dz = \frac{d\mathcal{L}}{dz} = \frac{d\mathcal{L}(a,y)}{dz}$$

$$= a - y$$

$$= \frac{d\mathcal{L}}{da} \cdot \frac{da}{dz}$$

$$a(1-a)$$

$$\text{"}da\text{"} = \frac{d\mathcal{L}(a,y)}{da} = -\frac{y}{a} + \frac{1-y}{1-a}$$

$$\frac{\partial \mathcal{L}}{\partial w_1} = \text{"}dw_1\text{"} = x_1 \cdot dz \qquad dw_2 = x_2 \cdot dz \qquad db = dz$$

$$w_1 := w_1 - \alpha\, dw_1$$
$$w_2 := w_2 - \alpha\, dw_2$$
$$b := b - \alpha\, db$$

Andrew Ng

Logistic regression on m examples

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

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

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

$$dw_1^{(i)},\ dw_2^{(i)},\ db^{(i)}$$

$$\frac{\partial}{\partial w_1} J(w,b) = \frac{1}{m}\sum_{i=1}^{m} \underbrace{\frac{\partial}{\partial w_1} \mathcal{L}(a^{(i)}, y^{(i)})}_{dw_1^{(i)}\ -\ (x^{(i)},\, y^{(i)})}$$

Andrew Ng

Logistic regression on m examples

$J = 0;\ dw_1 = 0;\ dw_2 = 0;\ db = 0$

For $i = 1$ to m

$$z^{(i)} = w^T x^{(i)} + b$$
$$a^{(i)} = \sigma(z^{(i)})$$
$$J \mathrel{+}= -[y^{(i)} \log a^{(i)} + (1-y^{(i)}) \log(1-a^{(i)})]$$
$$dz^{(i)} = a^{(i)} - y^{(i)}$$
$$dw_1 \mathrel{+}= x_1^{(i)} dz^{(i)}$$
$$dw_2 \mathrel{+}= x_2^{(i)} dz^{(i)}$$
$$db \mathrel{+}= dz^{(i)}$$

$n = 2$

dw_3 ⋮ dw_n

$J /= m$ ←

$dw_1 /= m;\ dw_2 /= m;\ db /= m.$ ←

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

$$w_1 := w_1 - \alpha\, dw_1$$
$$w_2 := w_2 - \alpha\, dw_2$$
$$b := b - \alpha\, db.$$

Vectorization

Andrew Ng

Vectorizing Logistic Regression

$dz^{(1)} = a^{(1)} - y^{(1)}$ $\quad dz^{(2)} = a^{(2)} - y^{(2)}$ $\quad \dots$

$$\boxed{dZ = [dz^{(1)}\ dz^{(2)} \dots dz^{(m)}]} \leftarrow$$

$1 \times m$

$A = [a^{(1)} \dots a^{(m)}]$ $\quad Y = [y^{(1)} \dots y^{(m)}]$

$\rightarrow dZ = A - Y = [a^{(1)} - y^{(1)} \quad a^{(2)} - y^{(2)} \quad \dots]$

$\rightarrow dw = 0$

$dw \mathrel{+}= x^{(1)} dz^{(1)}$

$dw \mathrel{+}= x^{(2)} dz^{(2)}$

$\vdots$

$dw \mathrel{/}= m$

$\cancel{dw_1}$, $\cancel{dw_2}$

$db = 0$

$db \mathrel{+}= dz^{(1)}$

$db \mathrel{+}= dz^{(2)}$

$\vdots$

$db \mathrel{+}= dz^{(m)}$

$db \mathrel{/}= m.$

$$\boxed{db = \frac{1}{m} \sum_{i=1}^{m} dz^{(i)} = \frac{1}{m}\, np.sum(dZ)}$$

$$\boxed{dw = \frac{1}{m} X\, dZ^T}$$

$$= \frac{1}{m} \begin{bmatrix} x^{(1)} & \dots & x^{(m)} \end{bmatrix} \begin{bmatrix} dz^{(1)} \\ \vdots \\ dz^{(m)} \end{bmatrix}$$

$$= \frac{1}{m} \left[x^{(1)} dz^{(1)} + \dots + x^{(m)} dz^{(m)} \right]$$

$n \times 1$

Andrew Ng

Implementing Logistic Regression

```
J = 0, dw1 = 0, dw2 = 0, db = 0
for i = 1 to m:
    z(i) = w^T x(i) + b  ←
    a(i) = σ(z(i))  ←
    J += -[y(i) log a(i) + (1 - y(i)) log(1 - a(i))]
    dz(i) = a(i) - y(i)  ←
    dw1 += x1(i) dz(i)
    dw2 += x2(i) dz(i)
    db += dz(i)
J = J/m, dw1 = dw1/m, dw2 = dw2/m
db = db/m
```

$dw \mathrel{+}= x^{(i)} * dz^{(i)}$

for iter in range(1000): $\leftarrow$

$Z = w^T X + b$

$= np.dot(w.T, X) + b$

$A = \sigma(Z)$

$dZ = A - Y$

$dw = \frac{1}{m} X dZ^T$

$db = \frac{1}{m} np.sum(dZ)$

$w := w - \alpha\, dw$

$b := b - \alpha\, db$

Andrew Ng

Broadcasting example

$$\begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix} + \begin{bmatrix} 100 \\ 100 \\ 100 \\ 100 \end{bmatrix} \cancel{100} = \begin{bmatrix} 101 \\ 102 \\ 103 \\ 104 \end{bmatrix}$$

$$\underset{(m,n)\ (2,3)}{\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}} + \underset{(1,n) \rightsquigarrow (m,n)\quad (2,3)}{\begin{bmatrix} 100 & 200 & 300 \\ 100 & 200 & 300 \end{bmatrix}} = \begin{bmatrix} 101 & 202 & 303 \\ 104 & 205 & 306 \end{bmatrix}$$

$$\underset{(m,n)}{\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}} + \underset{(m,1) \to (m,n)}{\begin{bmatrix} 100 & 100 & 100 \\ 200 & 200 & 200 \end{bmatrix}} = \begin{bmatrix} 101 & 102 & 103 \\ 204 & 205 & 206 \end{bmatrix} \begin{matrix} \leftarrow \\ \leftarrow \end{matrix}$$

General Principle

(m,n) matrix $\quad + \; - \; * \; / \quad (1,n) \rightsquigarrow (m,n)$

$(m,1) \rightsquigarrow (m,n)$

$(m,1) \quad + \quad \mathbb{R}$

$\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} + 100 = \begin{bmatrix} 101 \\ 102 \\ 103 \end{bmatrix}$

$[1 \;\; 2 \;\; 3] + 100 = [101 \;\; 102 \;\; 103]$

Matlab/Octave: bsxfun

Python/numpy vectors

```
a = np.random.randn(5)
```

a.shape = (5,)

"rank 1 array"

Don't use

```
a = np.random.randn(5,1)
```

→ a.shape = (5,1) column vector ✓

```
a = np.random.randn(1,5)
```

→ a.shape = (1,5) row vector. ✓

```
assert(a.shape == (5,1)) ←
```

a = a.reshape((5,1))

Andrew Ng

Logistic regression cost function

→ $\boxed{\text{If} \quad y = 1: \quad p(y|x) = \hat{y}}$

→ $\boxed{\text{If} \quad y = 0: \quad p(y|x) = 1 - \hat{y}}$

$p(y|x)$

$\boxed{p(y|x) = \hat{y}^{y} \, (1-\hat{y})^{(1-y)}}$ ←

If $y=1$: $p(y|x) = \hat{y} \quad \underbrace{(1-\hat{y})^{0}}_{=1}$

If $y=0$: $p(y|x) = \underbrace{\hat{y}^{0}}_{=1} \quad (1-\hat{y})^{(1-y)} = 1 \times (1-\hat{y}) = 1-\hat{y}$

$\uparrow \log p(y|x) = \log \hat{y}^{y} (1-\hat{y})^{(1-y)} = y \log \hat{y} + (1-y) \log(1-\hat{y})$

$= -\mathcal{L}(\hat{y}, y) \downarrow$

Andrew Ng

Cost on m examples

$\log p(\text{labels in training set}) = \log \prod_{i=1}^{m} p(y^{(i)} | x^{(i)})$

$\log p(\ldots\ldots) = \sum_{i=1}^{m} \underbrace{\log p(y^{(i)} | x^{(i)})}_{-\mathcal{L}(\hat{y}^{(i)},\, y^{(i)})}$

Maximum likelihood estimation

$= -\sum_{i=1}^{m} \mathcal{L}(\hat{y}^{(i)}, y^{(i)})$

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

(minimize)

Andrew Ng