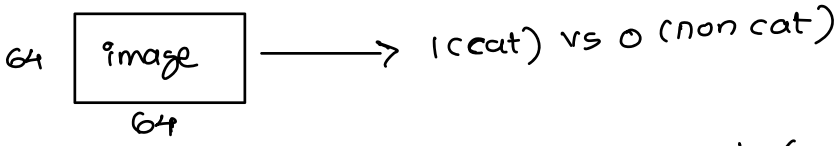
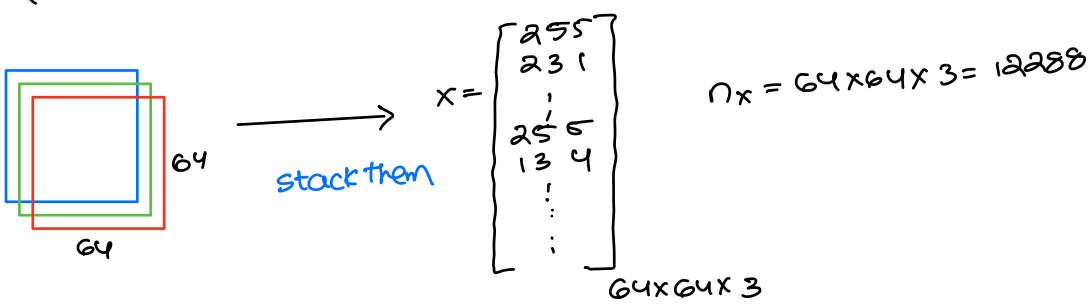
Lec 1. Binary Classification:



RGB (64 x 64 x 3) ⟶ feature vector



$n_x = 64 \times 64 \times 3 = 12288$

$n_x = n$ = dimensions of input feature vector (x)

Binary classification

$$x \rightarrow y$$

image (0,1)

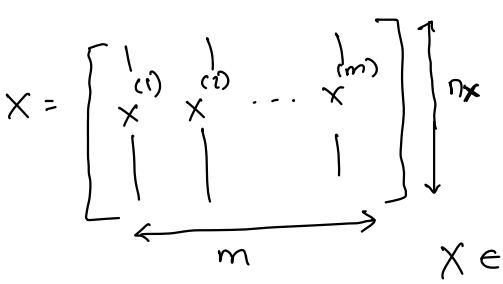
Notation:

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

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

training set m = M train $\quad$ n_x = dimension of feature vector.

test set m = M test



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

X.shape (n_x, m)

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

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

Y.shape $(1, m)$

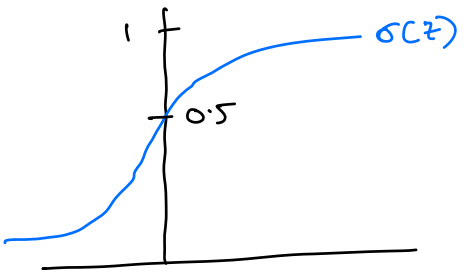
Lec2:- Logistic Regression:-

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

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

parameters of NN: $w \in \mathbb{R}^{n_x}$, $b \in \mathbb{R}$

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



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

if z large $\sigma(z) \simeq \frac{1}{1+0} = 1$

if z large negtive $\sigma(z) \simeq \frac{1}{1+Big} \simeq 0$

if $z = 0$ $\sigma(z) = \frac{1}{1+1} = 0.5$

Lec 3:- Logistic Regression Cost function:

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

Given $\{(x^{(1)}, y^{(1)}), \cdots (x^{(m)}, y^{(m)})\}$, want $\hat{y}^{(i)} \approx y^{(i)}$ (pred, ground truth)

$z^{(i)} = w^T x^{(i)} + b$; i^{th} example.

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

GD doesn't work well with squared error

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

$\mathcal{L}(\hat{y}, y) = -\left[y \log \hat{y} + (1-y) \log(1-\hat{y})\right]$ for i^{th} example.

min

if $y=1$ $\mathcal{L}(\hat{y}, 1) = -\log \hat{y} \Rightarrow \log \hat{y}$ as big as possible

$\Rightarrow \hat{y}$ has be large $\simeq 1$

if $y=0$ $\mathcal{L}(\hat{y}, 0) = -\log(1-\hat{y}) \Rightarrow \log(1-\hat{y})$ as big as possible

$\Rightarrow 1-\hat{y}$ has be lagre $\simeq 1$

$\Rightarrow \hat{y}$ has to be min.

Cost Function:

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

$$= -\frac{1}{m} \sum_{i=1}^{m} \left[y^{(i)} \log \hat{y}^{(i)} + (1-y^{(i)}) \log(1-\hat{y}^{(i)})\right]$$

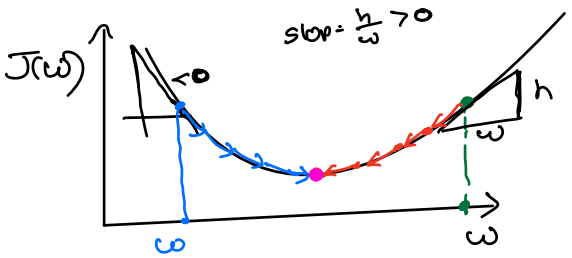
$\Rightarrow$ average loss function for entire training set.

dec 4: Gradient Descent :

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

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

want to find (w,b) that minimizes $J(w,b)$



Repeat {

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

}

α - learning rate

$dw = \frac{dJ(w)}{dw}$ - gradient

$J(w,b)$

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

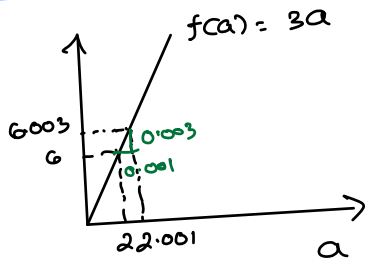
$$b := b - \alpha \frac{\partial J(w,b)}{\partial b}$$

$J(w,b)$

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

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

dec 5: Derivatives :-



$a = 2 \quad f(a) = 6$

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

derivate of $f(a)$ at $a = 2 = 3$

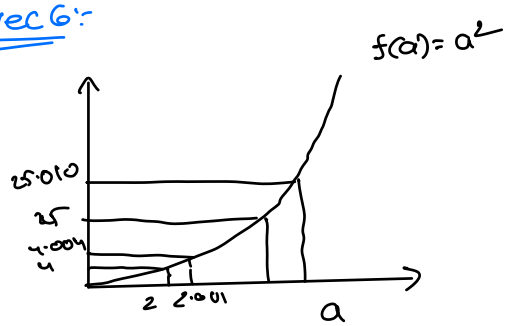
$$f'(2) = \frac{\Delta f(a)}{\Delta a} = \frac{0.003}{0.001} = 3$$

$a = 5 \quad f(a) = 15$

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

$$f'(5) = \frac{\Delta f(a)}{\Delta a} = \frac{0.003}{0.001} = 3$$

dec 6 :-



$a = 2 \quad f(a) = 4$

$a = 2.001 \quad f(a) = 4.004$

$$f'(2) = \frac{0.004}{0.001} = 4$$

$a = 5 \quad f(a) = 25$

$a = 5001 \quad f(a) = 25.010$

$$f'(5) = \frac{0.01}{0.001} = 10$$

$f(a) = a^2$

$f'(a) = 2a$

$f'(2) = 4 \quad f'(5) = 10$

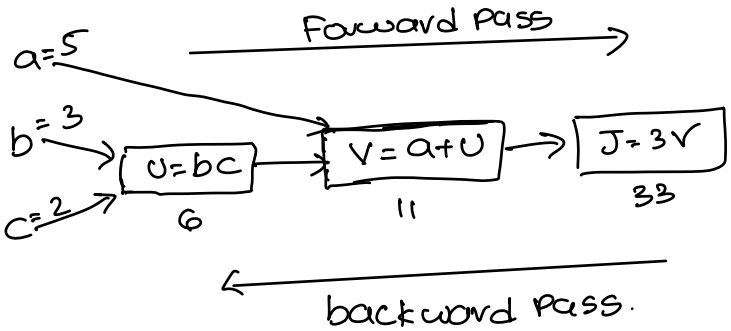
Lec 7 :- Computation Graph:-

$J(a,b,c) = 3(a+bc)$

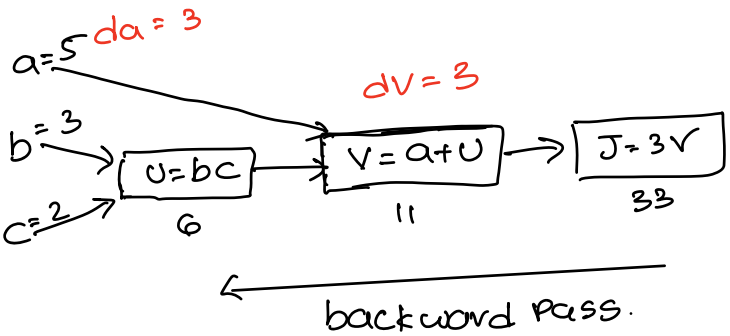
$u = bc$

$v = a+u$

$J = 3v$



Lec 8 :- Derivatives with a computation graph.



$\frac{dJ}{dv} = ?$

$J = 3v$

$dv = \frac{dJ}{dv} = 3$

$\frac{dJ}{da} = ?$

$a = 5 \rightarrow 5.001$

$v = 11 \rightarrow 11.001$

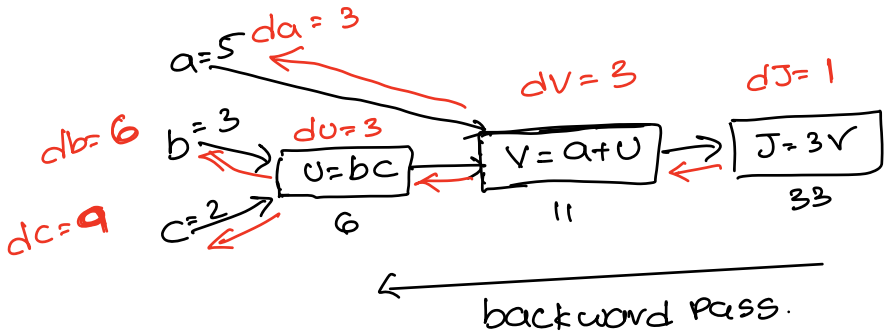
$J = 33 \rightarrow 33.003$

$\frac{dJ}{da} = \frac{0.003}{0.001} = 3$

$\frac{dJ}{da} = \frac{dJ}{dv}\frac{dv}{da} = \frac{d(3v)}{dv}\frac{d(a+u)}{da} = (3)(1) = 3$

$\therefore \frac{dJ}{da} = 3$

$\frac{d\,\text{Final output variable}}{d\,\text{var}} = d\text{var}$



$$du = \frac{dJ}{du} = \frac{dJ}{dV}\frac{dV}{du} = \frac{d(3V)}{dV}\frac{d(a+U)}{du} = (3)(1) = 3$$

$$db = \frac{dJ}{db} = \frac{dJ}{du}\frac{du}{db} = (3)\frac{d(bc)}{db} = (3)(c) = 3\times 2 = 6$$

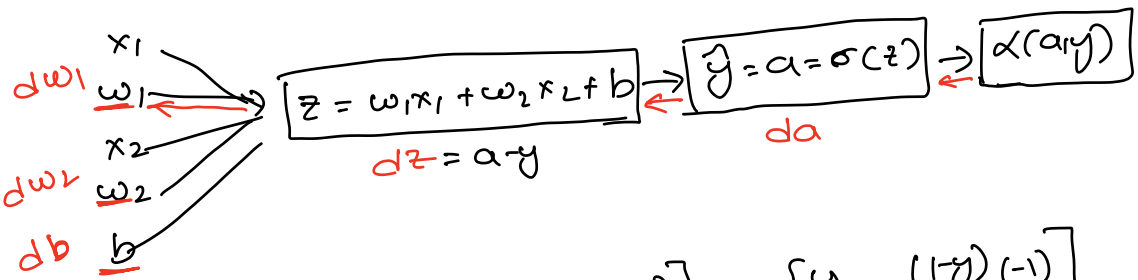
$$dc = \frac{dJ}{dc} = \frac{dJ}{du}\frac{du}{dc} = (3)\frac{d(bc)}{db} = (3)(b) = 3\times 3 = 9$$

Lec 9:- Logistic regression (GD with single training example).

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

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

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



$$da = \frac{d\mathcal{L}}{da} = \frac{d}{da}\left[-[y \log a + (1-y)\log(1-a)]\right] = -\left[\frac{y}{a} + \frac{(1-y)(-1)}{(1-a)}\right]$$

$$\boxed{da = -\frac{y}{a} + \frac{(1-y)}{(1-a)}}$$

$$dz = \frac{d\mathcal{L}}{dz} = \frac{d\mathcal{L}}{da}\frac{da}{dz} = (da)\frac{d\sigma(z)}{dz} =$$

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

$$\frac{d\sigma(z)}{dz} = \frac{-(e^{-z})}{(1+e^{-z})^2} = \frac{e^{-z}}{(1+e^{-z})}\left(\frac{1}{1+e^{-z}}\right)$$

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

$$= (1 - \sigma(z))(\sigma(z))$$

$$= (1-a)(a)$$

$$dz = (da)(1-a)(a)$$

$$= \left[\frac{-y}{a} + \frac{(1-y)}{(1-a)}\right](1-a)\,a = -y(1-a) + (1-y)a$$

$$= -y + ya + a - ya$$

$$= a - y$$

$$dz = a - y$$

$$dw_1 = \frac{dJ}{dw_1} = \frac{dJ}{dz}\frac{dz}{dw_1} = (dz)\frac{d(x_1w_1+x_2w_2+b)}{dw_1} = x_1dz$$

$$dw_2 = \frac{dJ}{dw_2} = \frac{dJ}{dz}\frac{dz}{dw_2} = (dz)\frac{d(x_1w_1+x_2w_2+b)}{dw_2} = x_2dz$$

$$db = \frac{dJ}{db} = \frac{dJ}{dz}\frac{dz}{db} = (dz)\frac{d(x_1w_1+x_2w_2+b)}{db} = dz$$

$dz = a - y$

$dw_1 = x_1dz$

$dw_2 = x_2dz$

$db = dz$

$w_1 := w_1 - \alpha dw_1$

$w_2 := w_2 - \alpha dw_2$

$b := b - \alpha db$

dec10:- Logistic Regression with m examples:-

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

$a^{(i)} = \hat{y}^{(i)} = \sigma(z^{(i)}) = \sigma(w^T x^{(i)} + b)$ $dw_1^{(i)}, dw_2^{(i)}, db^{(i)}$

$$\frac{\partial J(w,b)}{\partial w_1} = \frac{1}{m}\sum_{i=1}^{m} \frac{\partial L(a^{(i)}, y^{(i)})}{\partial w_1^{(i)}}$$

Initialize $J=0,\ dw_1=0,\ dw_2=0,\ db=0$

For $i = 1$ to m

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

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

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

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

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

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

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

$J \mathrel{/}= m$

$dw_1 \mathrel{/}= m$

$dw_2 \mathrel{/}= m$

$db \mathrel{/}= m$

$w_1 := w_1 - \alpha\, dw_1$

$w_2 := w_2 - \alpha\, dw_2$

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

Single Step of GD

Vectorization: dec11

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

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

Non vector:-

```
z = 0
for i in range(len(nx)):
    z += w[i] * x[i]
z += b
```

400ms

vector:-

```
z = np.dot(w, x) + b
```

$w^T x$

1.5ms

dec12:- More examples (avoid for loops, whenever possible)

$$u = Av$$

$$u = np.dot(A, v)$$

$$v = \begin{bmatrix} v_1 \\ \vdots \\ v_n \end{bmatrix} \qquad u = \begin{bmatrix} e^{v_1} \\ \vdots \\ e^{v_n} \end{bmatrix}$$

$$u = np.exp(v)$$

Vectorized inside loop :-

$J=0,\ dw = np.zeros((n_x, 1)),\ db = 0$

for i in range(len(m)):

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

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

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

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

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

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

$J /= m,\ dw /= m,\ db /= m$

dec 13:- Vectorizing logistic Regression :- (predictions)

$$z^{(1)} = w^T x^{(1)} + b \qquad z^{(2)} = w^T x^{(2)} + b$$

$$a^{(1)} = \sigma(z^{(1)}) \qquad a^{(2)} = \sigma(z^{(2)})$$

$$X = \begin{bmatrix} | & | & & | \\ x^{(1)} & x^{(2)} & \cdots & x^{(m)} \\ | & | & & | \end{bmatrix} \in \mathbb{R}^{(n_x, m)}$$

$$Z = \underset{(1,m)}{\left[z^{(1)}\ z^{(2)}\ \cdots\ z^{(m)}\right]} = \underset{(1,n_x)}{w^T} \underset{(n_x,m)}{X} + \underset{(1,m)}{\left[b\ b\ \cdots\ b\right]}$$

$$Z = np.dot(w.T, X) + b_{(1,1)}$$

$(1,m) + (1,1) \rightarrow$ broadcasting

$$A = \left[a^{(1)}\ a^{(2)}\ \cdots\ a^{(m)}\right] = \sigma(Z)$$

Lec 14:- Vectorizing backward pass:

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

$$dz = \begin{bmatrix} dz^{(1)} & dz^{(2)} & \cdots & dz^{(m)} \end{bmatrix}_{(1,m)}$$

$$A = \begin{bmatrix} a^{(1)} & a^{(2)} & \cdots & a^{(m)} \end{bmatrix}$$

$$Y = \begin{bmatrix} y^{(1)} & y^{(2)} & \cdots & y^{(m)} \end{bmatrix}$$

$$dz = A - Y = \begin{bmatrix} a^{(1)} - y^{(1)} & a^{(2)} - y^{(2)} & \cdots & a^{(m)} - y^{(m)} \end{bmatrix}$$

Initialize $dw = 0$

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

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

$\vdots$

$dw \mathrel{/}= m$

$db = 0$

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

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

$\vdots$

$db \mathrel{/}= m$

looping through m examples

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

$$dw = \frac{1}{m} X dz^{T} = \frac{1}{m} \begin{bmatrix} x^{(1)} & x^{(2)} & \cdots & x^{(m)} \\ | & | & & | \end{bmatrix} \begin{bmatrix} dz^{(1)} \\ dz^{(2)} \\ \vdots \\ dz^{(m)} \end{bmatrix}$$

$$dw = \frac{1}{m} \left[\underset{(n_x,1)}{x^{(1)} dz^{(1)}} + \underset{(n_x,1)}{x^{(2)} dz^{(2)}} + \cdots + \underset{(n_x,1)}{x^{(m)} dz^{(m)}} \right] \quad (n_x, 1)$$

Final Implementation of Logistic Regression : Lec 15

For i in range(iter):

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

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

$$A = \sigma(z)$$

$$dz = A - Y$$

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

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

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

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

dec 16: Broadcasting example cal in 100g of food

	Apples	Beef	Eggs	Potatoes
carb	56.0	0.0	4.4	68.0
Protein	1.2	104.0	52.0	8.0
Fat	1.8	135.0	99.0	0.9

A =

% of cal from carbs, protein, Fat.

cal = A.sum(axis=0) = [59.0 239.0 155.4 76.9]

per = [A/cal] (broadcasting)

A(3,4) cal(1,4)

$$\begin{bmatrix}1\\2\\3\\4\end{bmatrix}_{4\times1} + 100_{(1,1)} = \begin{bmatrix}1\\2\\3\\4\end{bmatrix} + \begin{bmatrix}100\\100\\100\\100\end{bmatrix} = \begin{bmatrix}101\\102\\103\\104\end{bmatrix}_{(4,1)}$$

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

General principle:-

$(m,n) \;\underset{*\,/}{\pm}\; (1,n) \rightsquigarrow (m,n)$

$(m,n) \;\underset{*\,/}{\pm}\; (m,1) \rightsquigarrow (m,n)$

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

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

Lec17: Numpy Vectors:-

assert (a.shape == (5,1))

don't use rank1 array (n,)

use row [1,n] vector

use column [n,1] vector.

use veshape operation.

Lec18:- Cost Function:-

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

interpret $\hat{y} = P(y=1|x)$

if $y=1$: $P(y|x) = \hat{y}$

$y=0$: $P(y|x) = 1-\hat{y}$

$$P(y|x) = \hat{y}^y (1-\hat{y})^{(1-y)}$$

if $y=1$ $P(y|x) = \hat{y}$

if $y=0$ $P(y|x) = (1-\hat{y})$

$$\log P(y|x) = y \log \hat{y} + (1-y) \log(1-\hat{y}) = -\mathcal{L}(y, \hat{y})$$

min loss $\Rightarrow$ max $P(y|x)$

(iid) (identical independent dist)

cost on m examples:-

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

$$\log P(\quad) = \sum_{i=1}^{m} \log P(y^{(i)}|x^{(i)})$$

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

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

minimize