DLSG – Week 2
Neural Network Basic

Do Thi Ngan
July 2019

Binary Classification

- Pair(x,y) – Single training example
  - Input x: dimensional feature vector
  - Output y: label value 0 or 1

28

35

Input x

1 (dog) or 0 (not dog)

Feature vector
X

Ouput y

- nx = 28*35*3 = 2940 dimensions

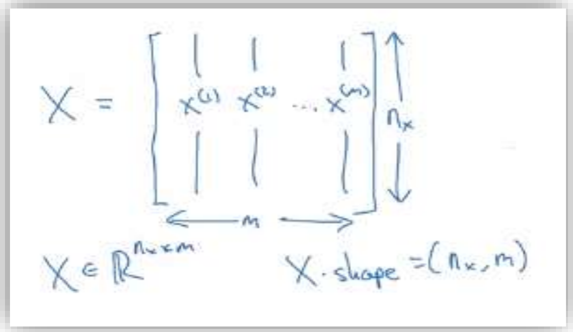
Binary Classification

- M pair (x,y) – Multiple training example
  - M = 2

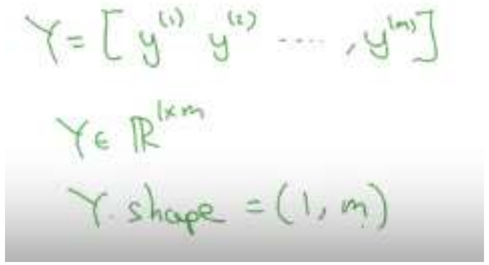
1 (dog) or 0 (not dog)
1 (cat) or 0 (not cat)

⇒M train = {(x1,y1), (x2,y2), ..., (xm,ym)}

- Testing set



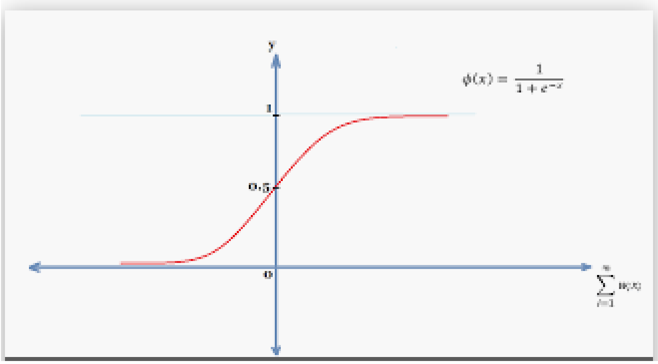
Input X



Output Y

Logistic Regression

- Input x $\rightarrow$ y predict ($\hat{y}$) = P(y=1|x)
  x € R^{nx}
- Output $\hat{y}$ = $w^T x + b$
  Đặt $w^T x + b$ = z



Do $\hat{y}$ € [0,1] => $\hat{y}$ tương đương hàm sigmoid g(z)

$\Rightarrow$ $\hat{y}$ = g(z) = $\frac{1}{1+ e^{-z}}$

- If z >> 0 $\rightarrow$ g(z) = $\frac{1}{1+0}$ = 1
- If z << 0 $\rightarrow$ g(z) = $\frac{1}{1+larger\ numer}$ = 0

$\Rightarrow$ Tìm w, b sao cho $\hat{y}$ ~ y

Logistic Regress Cost Function

- Lost function: error for a single training example

$$£(\hat{y},y) = \frac{1}{2}(\hat{y} - y)^2$$

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

- Cost function: average of the lost function of m training examples

$$J(w,b) = -\frac{1}{m}\sum_{i=1}^{m} £(\hat{y}i, yi)$$

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

Gradient Descent

Recap:

- Lost function: error for a single training example

$$£(\hat{y}, y) = \frac{1}{2}(\hat{y} - y)^2$$

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

- Cost function: average of the lost function of m training examples

$$J(w,b) = -\frac{1}{m}\sum_{i=1}^{m} £(\hat{y}i, yi)$$

$$= -\frac{1}{m}\sum_{i=1}^{m}[yi\log\widehat{yi} + (1-yi)\log(1-\widehat{yi})]$$

Gradient Descent

- Gradient descent is do moving point (w,b) for algorithm converges to minimum value rate

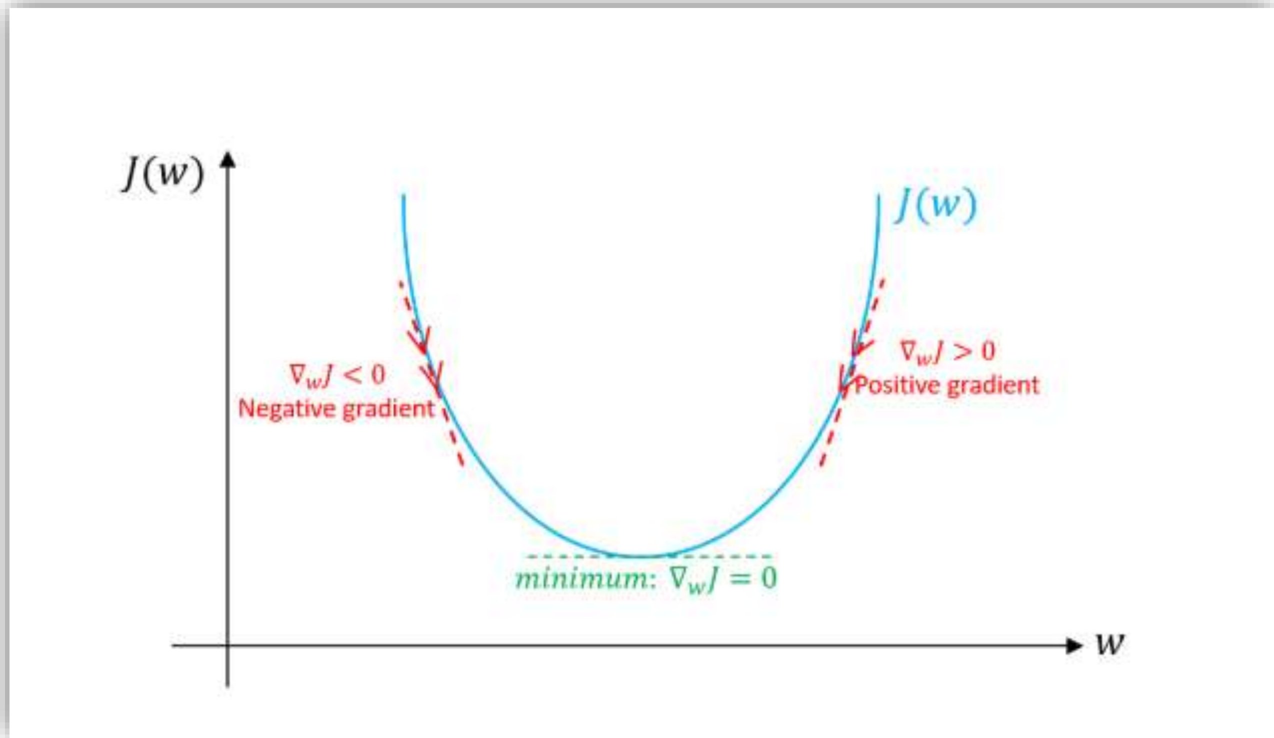
$$dw = \frac{dJ(w,b)}{dw}$$

- Algorithm implement repeat like graph beside:

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

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

α is learning rate



Derivative

- Derivative is the sl o to ope of fuction
- Different point on function → different slope
- Examples:

$\boxed{f(a) = a^2}$

- $f(2) = 2^2 = 4$
- $f(2.001) = 4.0004$
- Slope $= \frac{f(2.001)-f(2)}{2.001-2} = 4$
- $f'(2) = 2a = 4$

=> Slope of f(a) = 4 at a = 2

$\boxed{f(a) = \ln(a)}$

- $f(2) = \ln(2) = 0.69315$
- $f(2.001) = 0.69365$
- Slope $= \frac{f(2.001)-f(2)}{2.001-2} = 0.5$
- $f'(2) = {}^{1}/_{a} = 0.5$

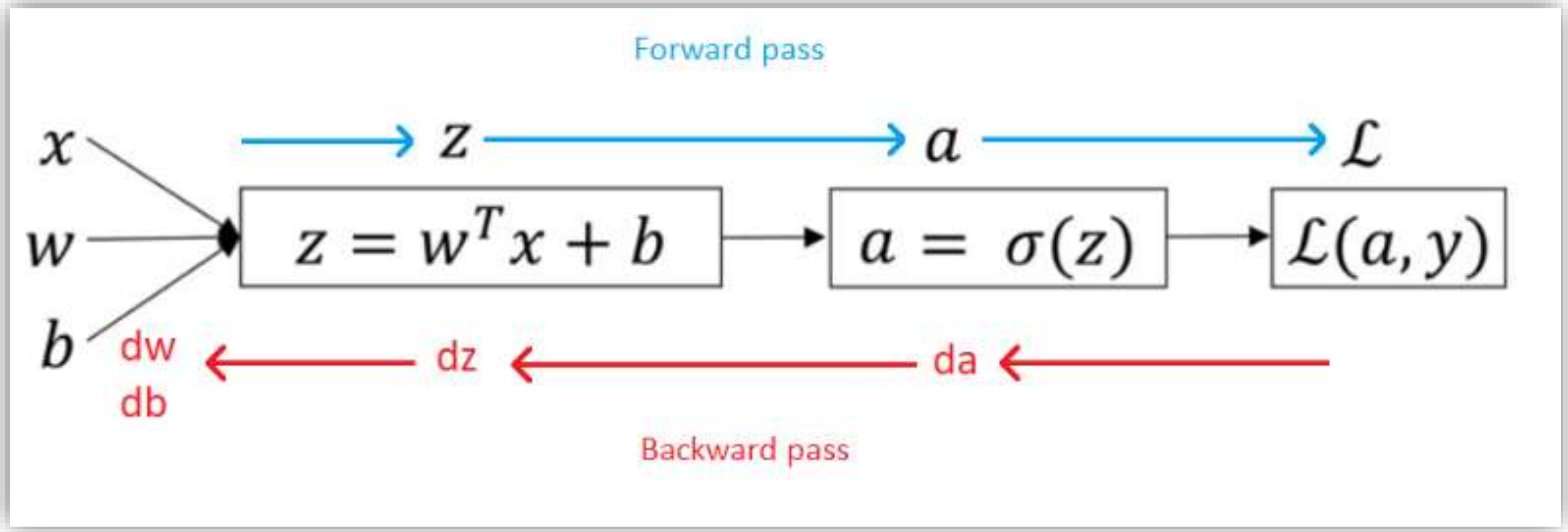
=> Slope of f(a) = 0.5 at a = 2

??? What is slope applied in Neural Network and Deep learning

Derivative with a Computation Graph

- Computation graph is flow to implement an calculation express
- Base on computation graph, we can analysis follow 2 way:
  - Forward: right to left → calculate derivative
  - Backward: left to right → determine affection of each variable to derivative

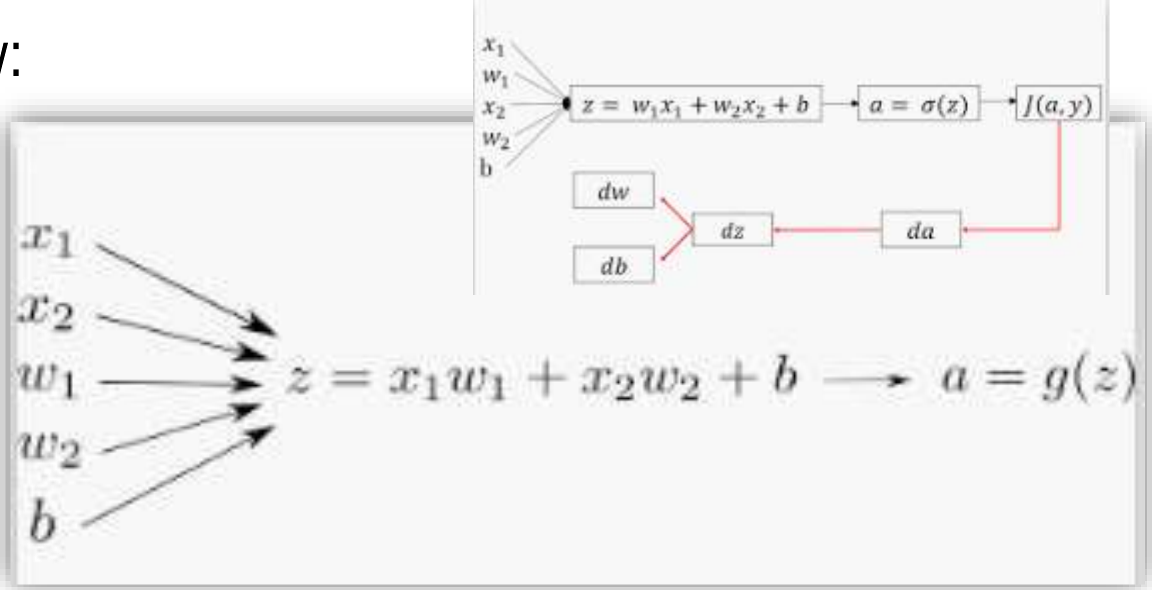
- Examples:



Logistic Regression Gradient Descent

- Logistic Regression recap:
  - z = $w^T x + b$
  - $\hat{y}$ = a = sigmoid(z)
  - £($\hat{y}$,y) = = - (yloga$+(1-y)$log(1-a)

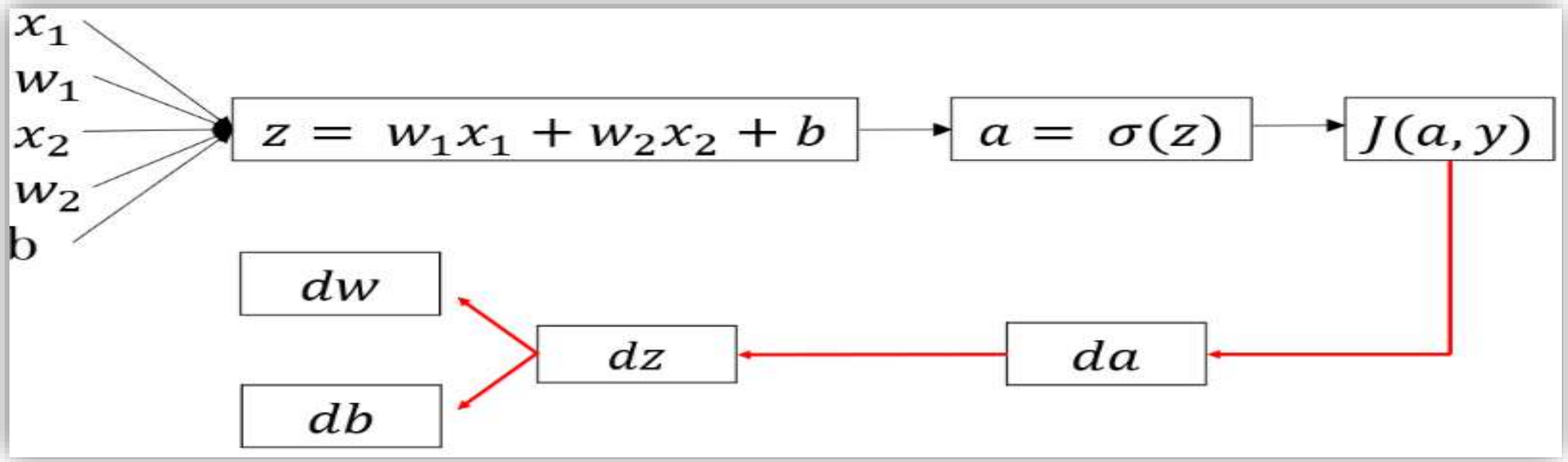
- Forward flow:



→ ??? w, b to minimize £($\hat{y}$,y)

Logistic Regression Gradient Descent

- Forward flow:



$$da = \frac{d£(a,y)}{da} = \frac{y}{a} + \frac{1-y}{1-a}$$

$$dz = \frac{d£(a,y)}{dz} = a - y$$

$$= \frac{df}{da} * \frac{da}{dz} = da * \frac{da}{dz}$$

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

$$\frac{\partial £}{\partial w1} = dw1 = x1dz$$

$$\frac{\partial £}{\partial w2} = dw2 = x2dz$$

$$db = dz$$