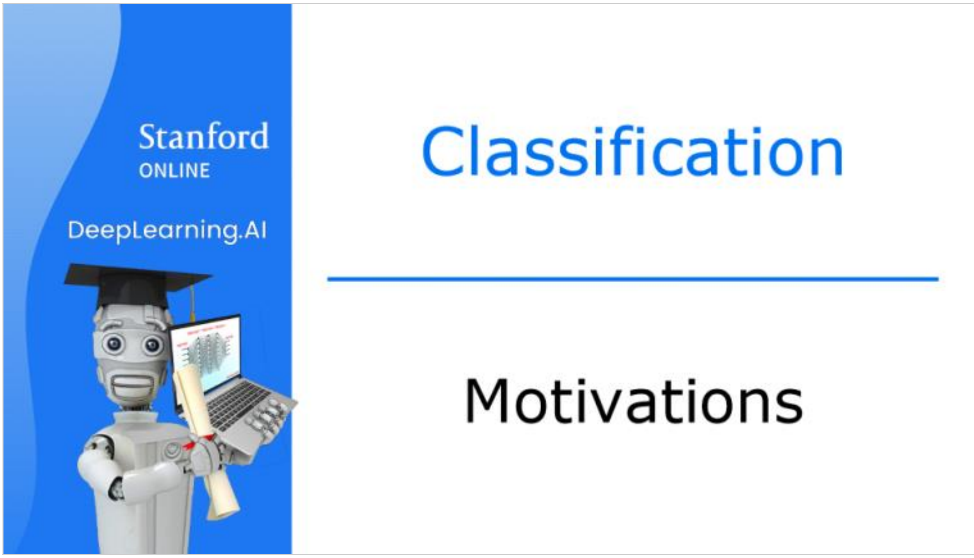
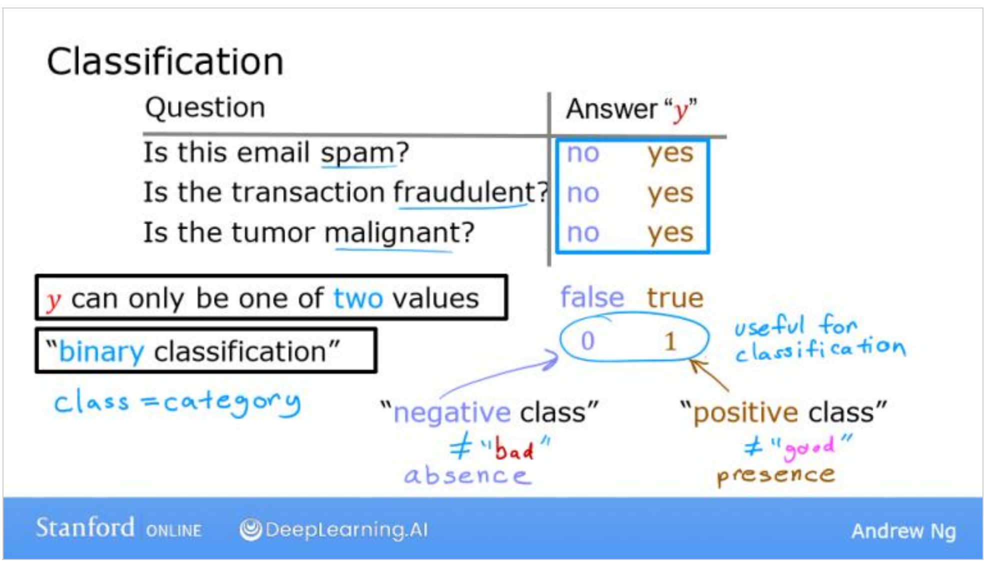
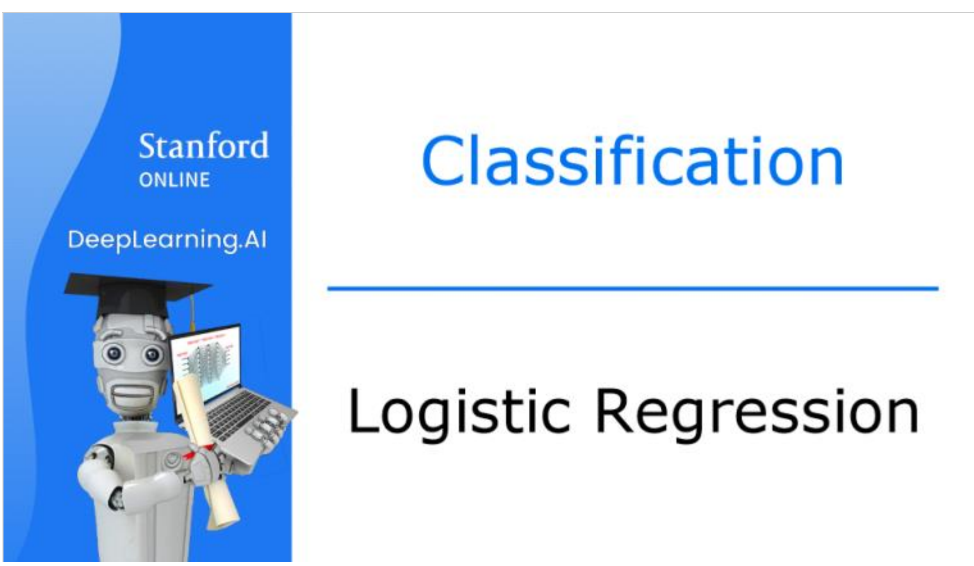
# Logistic Regression

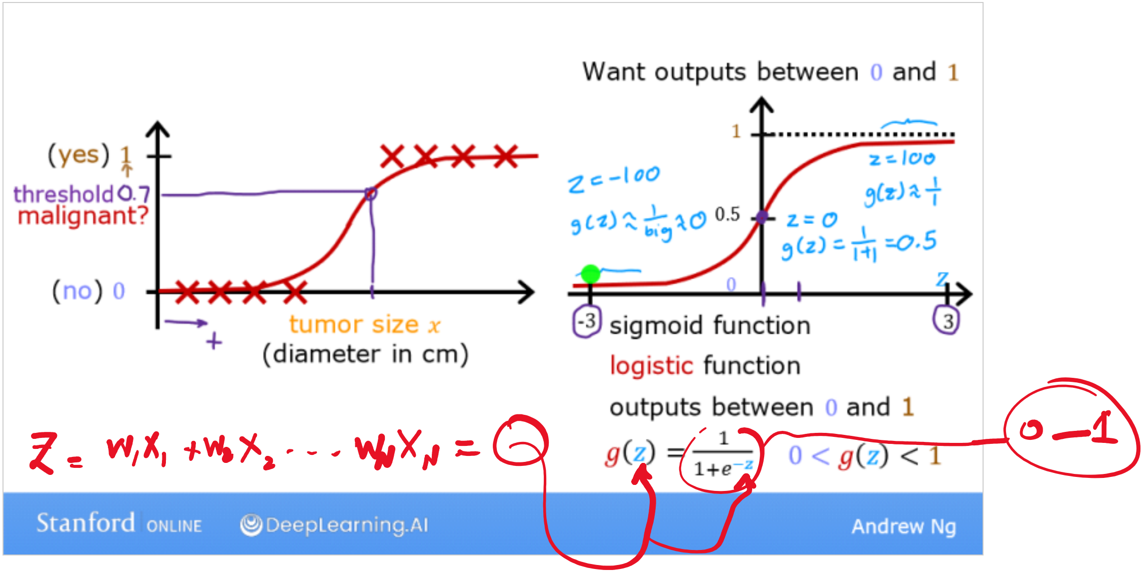
Sunday, March 23, 2025 11:59 AM

C1_W33






$X = X_1, X_2 \cdots X_N$

$W_1X_1 + W_2X_2 \cdots + W_NX_N$

$g(Z)$

| =1 | =0 |
|---|---|

A Set of "good" W i.e $W_1, W_2, W_3 \cdots W_N, b$ (bias)

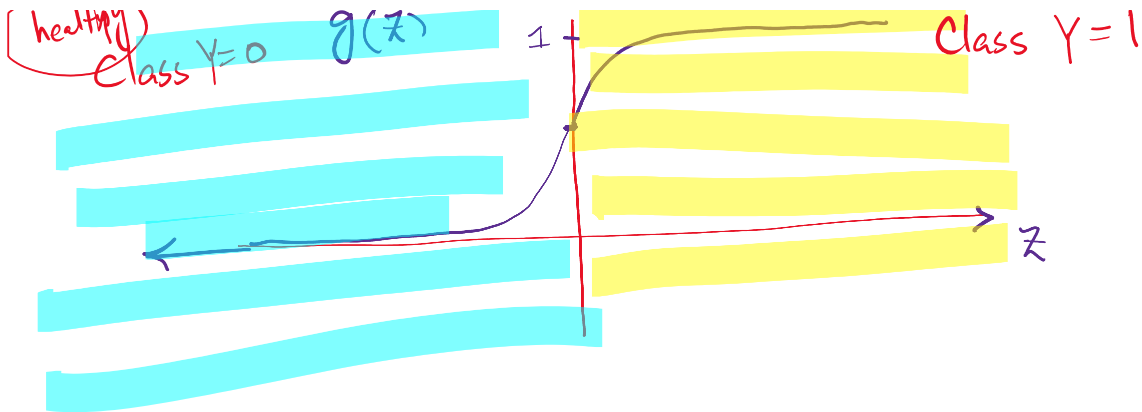
will generate a $Z = W_1X_1 + W_2X_2 \cdots W_NX_N + b$, SUCH THAT

$Z$ will be -ve (Preferably BIG -ve)
**For Class Y=0**
So that $g(Z) \longrightarrow 0$
giving correct Classification

$Z$ will be +ve (Preferably BIG +ve)
**For class Y=1**
So that $g(Z) \longrightarrow 1$
giving correct classification

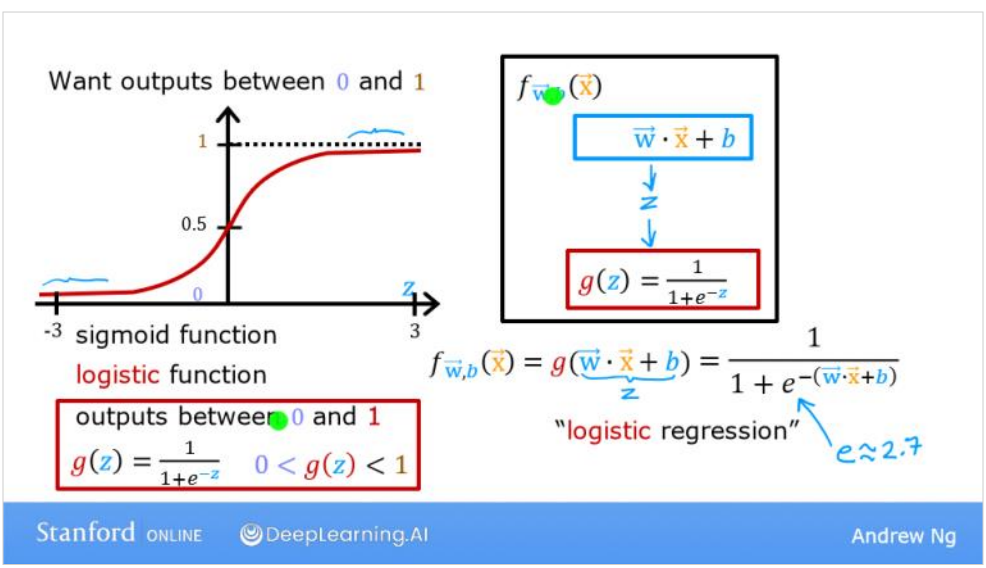
healthy Class Y=0 $g(Z)$

Class Y=1




# Define Loss function

| z | −ve | 0 | +ve | |
|---|---|---|---|---|
| g(z) | 0 | 0.5 | 1 | |
| Loss for Y=1 | ∞ high | | 0 min | $-\log[g(z)]$ |
| Loss for Y=0 | 0 Low | | ∞ high | $-\log[1-g(z)]$ |




# Classification

## Decision Boundary

## Interpretation of logistic regression output

$$f_{\vec{w},b}(\vec{x}) = \frac{1}{1+e^{-(\vec{w}\cdot\vec{x}+b)}}$$

"probability" that class is 1

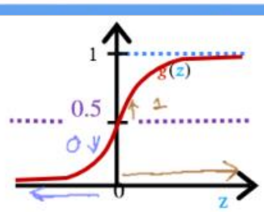
$$\overset{P(1)}{f_{\vec{w},b}(\vec{x})} = P(y=1|\vec{x};\vec{w},b)$$

Probability that $y$ is 1,
given input $\vec{x}$, parameters $\vec{w}$,$b$

Example:
$x$ is "tumor size"
$y$ is 0 (not malignant)
or 1 (malignant)

$$P(y=0)+P(y=1)=1$$

$f_{\vec{w},b}(\vec{x}) = 0.7$
70% chance that $y$ is 1




$$f_{\vec{w},b}\left(\vec{x}\right) = g(\underbrace{\vec{w}\cdot\vec{x}+b}_{z}) = \frac{1}{1+e^{-\left(\vec{w}\cdot\vec{x}+b\right)}}$$

$$= P\left(y=1\,\middle|\,x;\vec{w},b\right) \quad 0.7 \quad 0.3$$

0 or 1? threshold

Is $f_{\vec{w},b}\left(\vec{x}\right) \geq 0.5$?
Yes: $\hat{y}=1$ No: $\hat{y}=0$

When is
$f_{\vec{w},b}\left(\vec{x}\right) \geq 0.5$? $g(z) \geq 0.5$

| | |
|---|---|
| $z \geq 0$ | $z < 0$ |
| $\vec{w}\cdot\vec{x}+b \geq 0$ | $\vec{w}\cdot\vec{x}+b < 0$ |
| $\hat{y}=1$ | $\hat{y}=0$ |

$f_{\vec{w},b}\left(\vec{x}\right)$

$\boxed{z = \vec{w}\cdot\vec{x}+b}$

$\downarrow$ $z$ $\downarrow$

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

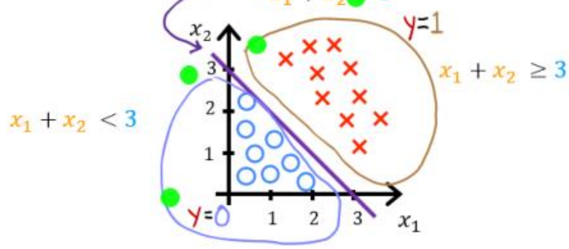
# Decision boundary

$$f_{\vec{w},b}(\vec{x}) = g(z) = g(\overbrace{\underset{1}{w_1}x_1 + \underset{1}{w_2}x_2 + \underset{-3}{b}})$$

Decision boundary $z = \vec{w} \cdot \vec{x} + b = 0$

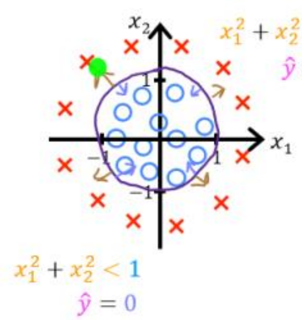
$z = x_1 + x_2 - 3 = 0$

$x_1 + x_2 = 3$






# Non-linear decision boundaries



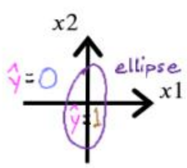

$$\overbrace{\underset{1}{w_1}x_1^2 + \underset{1}{w_2}x_2^2 + \underset{-1}{b}}^{z}$$

decision boundary $z = x_1^2 + x_2^2 - 1 = 0$

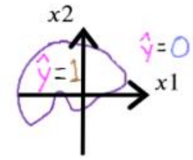
$x_1^2 + x_2^2 = 1$

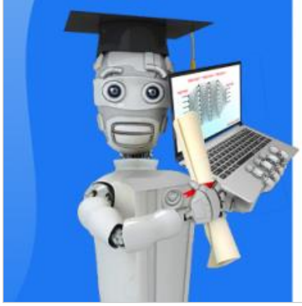


# Non-linear decision boundaries




$$f_{\vec{w},b}(\vec{x}) = g(z) = g(w_1x_1 + w_2x_2 + w_3x_1^2 + w_4x_1x_2 + w_5x_2^2 + w_6x_1^3 + \cdots + b)$$





Stanford ONLINE

DeepLearning.AI

# Cost Function

## Cost Function for Logistic Regression

---

## Training set

| | tumor size (cm) $x_1$ | ... | patient's age $x_n$ | malignant? $y$ |
|---|---|---|---|---|
| $i=1$ | 10 | | 52 | 1 |
| $\vdots$ | 2 | | 73 | 0 |
| | 5 | | 55 | 0 |
| | 12 | | 49 | 1 |
| $i=m$ | ... | | ... | ... |

$i = 1, \ldots, m$ ← training examples

$j = 1, \ldots, n$ ← features

$\boxed{\text{target } y \text{ is } 0 \text{ or } 1}$

$$f_{\vec{w},b}(\vec{x}) = \frac{1}{1 + e^{-(\vec{w}\cdot\vec{x}+b)}}$$

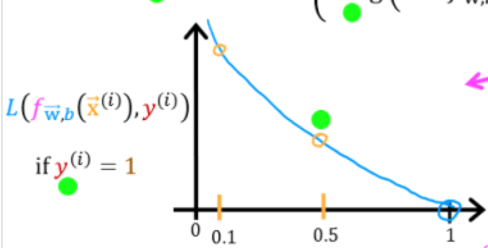
How to choose $\vec{w} = [w_1 \quad w_2 \quad \cdots \quad w_n]$ and $b$?

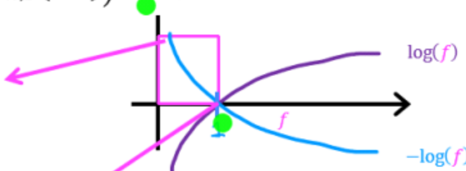


---

## Logistic loss function

$$L\left(f_{\vec{w},b}\left(\vec{x}^{(i)}\right), y^{(i)}\right) = \begin{cases} -\log\left(f_{\vec{w},b}\left(\vec{x}^{(i)}\right)\right) & \text{if } y^{(i)} = 1 \\ -\log\left(1 - f_{\vec{w},b}\left(\vec{x}^{(i)}\right)\right) & \text{if } y^{(i)} = 0 \end{cases}$$



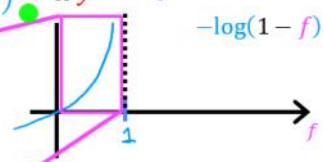

As $f_{\vec{w},b}\left(\vec{x}^{(i)}\right) \to 1$ then $\text{loss} \to 0$

As $f_{\vec{w},b}\left(\vec{x}^{(i)}\right) \to 0$ then $\text{loss} \to \infty$

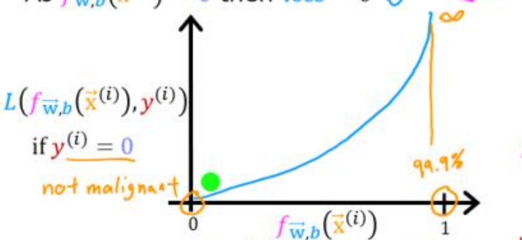
Loss is lowest when $f_{\vec{w},b}\left(\vec{x}^{(i)}\right)$ predicts close to true label $y^{(i)}$.

## Logistic loss function

$$L\left(f_{\vec{w},b}\left(\vec{x}^{(i)}\right), y^{(i)}\right) = \begin{cases} -\log\left(f_{\vec{w},b}\left(\vec{x}^{(i)}\right)\right) & \text{if } y^{(i)} = 1 \\ -\log\left(1 - f_{\vec{w},b}\left(\vec{x}^{(i)}\right)\right) & \text{if } y^{(i)} = 0 \end{cases}$$




As $f_{\vec{w},b}\left(\vec{x}^{(i)}\right) \to 0$ then $\text{loss} \to 0$




As $f_{\vec{w},b}\left(\vec{x}^{(i)}\right) \to 1$ then $\text{loss} \to \infty$

The further prediction $f_{\vec{w},b}\left(\vec{x}^{(i)}\right)$ is from target $y^{(i)}$, the higher the loss.

Stanford ONLINE DeepLearning.AI Andrew Ng

## Cost

cost

$$J(\vec{w}, b) = \frac{1}{m}\sum_{i=1}^{m} \underbrace{L\left(f_{\vec{w},b}\left(\vec{x}^{(i)}\right), y^{(i)}\right)}_{\text{loss}}$$

$$= \begin{cases} -\log\left(f_{\vec{w},b}\left(\vec{x}^{(i)}\right)\right) & \text{if } y^{(i)} = 1 \\ -\log\left(1 - f_{\vec{w},b}\left(\vec{x}^{(i)}\right)\right) & \text{if } y^{(i)} = 0 \end{cases}$$

convex
↳ can reach a global minimum

find w, b that minimize cost J

Stanford ONLINE DeepLearning.AI Andrew Ng

Stanford ONLINE

DeepLearning.AI

# Cost Function

## Simplified Cost Function for Logistic Regression

## Simplified loss function

$$L\left(f_{\vec{w},b}(\vec{x}^{(i)}), y^{(i)}\right) = \begin{cases} \boxed{-\log\left(f_{\vec{w},b}(\vec{x}^{(i)})\right)} & \text{if } y^{(i)} = 1 \\ -\log\left(1 - f_{\vec{w},b}(\vec{x}^{(i)})\right) & \text{if } y^{(i)} = 0 \end{cases}$$

$$\boxed{L\left(f_{\vec{w},b}(\vec{x}^{(i)}), y^{(i)}\right) = -\underset{1}{y^{(i)}}\log\left(f_{\vec{w},b}(\vec{x}^{(i)})\right) \cancel{- \underbrace{(1 - y^{(i)})}_{\underbrace{(1-1)}_{0}}\log\left(1 - f_{\vec{w},b}(\vec{x}^{(i)})\right)}}$$

if $y^{(i)} = 1$:

$$L\left(f_{\vec{w},b}(\vec{x}^{(i)}), y^{(i)}\right) = \underline{-1 \log(f(\vec{x}))}$$

Stanford ONLINE DeepLearning.AI Andrew Ng

## Simplified loss function

$$L\left(f_{\vec{w},b}(\vec{x}^{(i)}), y^{(i)}\right) = \begin{cases} -\log\left(f_{\vec{w},b}(\vec{x}^{(i)})\right) & \text{if } y^{(i)} = 1 \\ \boxed{-\log\left(1 - f_{\vec{w},b}(\vec{x}^{(i)})\right)} & \text{if } y^{(i)} = 0 \end{cases}$$

$$\boxed{L\left(f_{\vec{w},b}(\vec{x}^{(i)}), y^{(i)}\right) = \cancel{-\underset{0}{y^{(i)}}\log\left(f_{\vec{w},b}(\vec{x}^{(i)})\right)} - \underset{(1-0)}{(1 - y^{(i)})}\log\left(1 - f_{\vec{w},b}(\vec{x}^{(i)})\right)}$$

if $y^{(i)} = 1$:

$$L\left(f_{\vec{w},b}(\vec{x}^{(i)}), y^{(i)}\right) = -1 \log(f(\vec{x}))$$

if $y^{(i)} = 0$:

$$L\left(f_{\vec{w},b}(\vec{x}^{(i)}), y^{(i)}\right) = \underline{-(1-0)\log(1-f(\vec{x}))}$$

Stanford ONLINE DeepLearning.AI Andrew Ng

## Simplified cost function

loss

$$L\left(f_{\vec{w},b}(\vec{x}^{(i)}), y^{(i)}\right) = -\underbrace{y^{(i)}\log\left(f_{\vec{w},b}(\vec{x}^{(i)})\right) - (1 - y^{(i)})\log\left(1 - f_{\vec{w},b}(\vec{x}^{(i)})\right)}$$

cost

$$J(\vec{w}, b) = \frac{1}{m}\sum_{i=1}^{m}\left[\underline{L\left(f_{\vec{w},b}(\vec{x}^{(i)}), y^{(i)}\right)}\right]$$

convex
(single global minimum)

$$= -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\log\left(f_{\vec{w},b}(\vec{x}^{(i)})\right) + (1 - y^{(i)})\log\left(1 - f_{\vec{w},b}(\vec{x}^{(i)})\right)\right]$$

maximum likelihood
(don't worry about it!)

Stanford ONLINE DeepLearning.AI Andrew Ng

Stanford ONLINE

DeepLearning.AI

# Gradient Descent

## Gradient Descent Implementation

### Training logistic regression

Find $\vec{w}, b$

Given new $\vec{x}$, output $f_{\vec{w},b}(\vec{x}) = \frac{1}{1+e^{-(\vec{w}\cdot\vec{x}+b)}}$

$$P(y = 1|\vec{x}; \vec{w}, b)$$

Stanford ONLINE DeepLearning.AI Andrew Ng

derivative of sigmoid

$$\begin{aligned}
g'(z) &= \frac{d}{dz}\frac{1}{1+e^{-z}} \\
&= \frac{1}{(1+e^{-z})^2}\left(e^{-z}\right) \\
&= \frac{1}{(1+e^{-z})}\cdot\left(1-\frac{1}{(1+e^{-z})}\right) \\
&= g(z)(1-g(z)).
\end{aligned}$$

### Gradient descent

cost

$m$

## Gradient descent

cost

$$J(\vec{w},b) = -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\log\left(f_{\vec{w},b}(\vec{x}^{(i)})\right) + (1-y^{(i)})\log\left(1-f_{\vec{w},b}(\vec{x}^{(i)})\right)\right]$$

repeat {

j=1...n

$$w_j = w_j - \alpha\frac{\partial}{\partial w_j}J(\vec{w},b) \qquad \frac{\partial}{\partial w_j}J(\vec{w},b) = \frac{1}{m}\sum_{i=1}^{m}\left(f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)}\right)x_j^{(i)}$$

$$b = b - \alpha\frac{\partial}{\partial b}J(\vec{w},b) \qquad \frac{\partial}{\partial b}J(\vec{w},b) = \frac{1}{m}\sum_{i=1}^{m}\left(f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)}\right)$$

} simultaneous updates

Stanford ONLINE DeepLearning.AI Andrew Ng

chain Rule chain Rule

$$\frac{\partial}{\partial w_j}J(w,b) = \frac{\partial J(w,b)}{\partial[g(z)]}\frac{\partial[g(z)]}{\partial w_j} = \frac{\partial J(w,b)}{\partial[g(z)]} \times g(z)(1-g(z))\frac{\partial z}{\partial w}$$

$$= \frac{\partial J(w,b)}{\partial[g(z)]}\, g(z)[1-g(z)]\, x_j$$

$$\frac{\partial J(w,b)}{\partial g(z))} = \frac{-y^i\log[g(z)] - (1-y^i)\log[1-g(z)]}{\partial g(z)}$$

$$= -y^i\frac{1}{g(z)} - (1-y^i)\frac{-1}{1-g(z)}$$

$$= -\frac{y^i}{g(z)} - \frac{-(1-y^i)}{1-g(z)}$$

$$= \left[\frac{-y^i}{g(z)} - \frac{-(1-y^i)}{1-g(z)}\right] g(z)[1-g(z)]\, x_j$$

Many steps .....

$$= -[y^i - g(z)]\,x_j$$

$$\frac{\partial J(w,b)}{\partial w_j} = [g(z) - y^i]\,x_j$$

- This is derivative of Loss function w.r.t $w_j$
- Grad-descent will use this derivative to minimize loss function $J(w,b)$
- Final Goal is to find best $w_j$ $[w_1, w_2, w_3 ---- w_N]$

# Gradient descent for logistic regression

repeat {    looks like linear regression!

$$w_j = w_j - \alpha \left[ \frac{1}{m} \sum_{i=1}^{m} \left( f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)} \right) x_j^{(i)} \right]$$

$$b = b - \alpha \left[ \frac{1}{m} \sum_{i=1}^{m} \left( f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)} \right) \right]$$

} simultaneous updates

Same concepts:
- Monitor gradient descent (learning curve)
- Vectorized implementation
- Feature scaling

Linear regression $f_{\vec{w},b}(\vec{x}) = \vec{w} \cdot \vec{x} + b$

Logistic regression $f_{\vec{w},b}(\vec{x}) = \frac{1}{1 + e^{(-\vec{w} \cdot \vec{x} + b)}}$

Stanford ONLINE DeepLearning.AI Andrew Ng