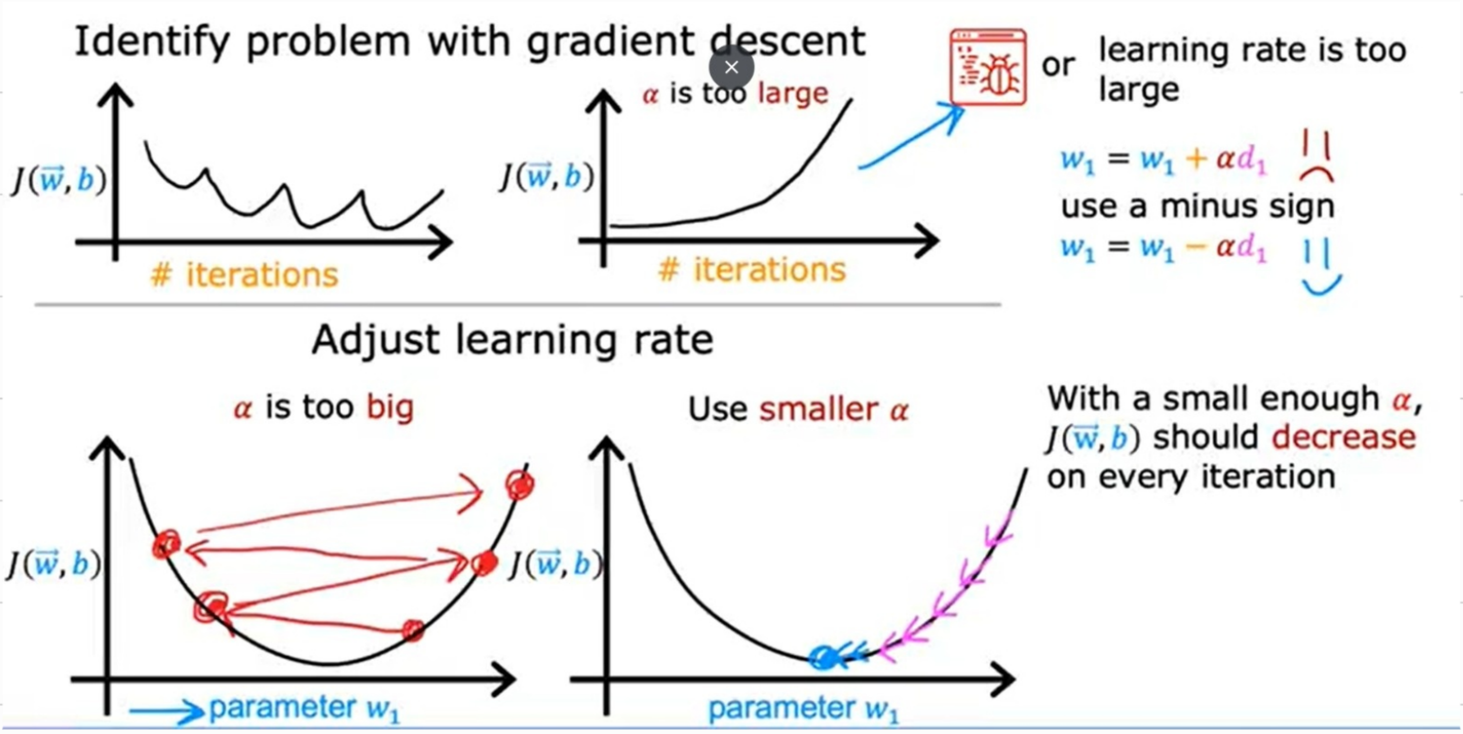
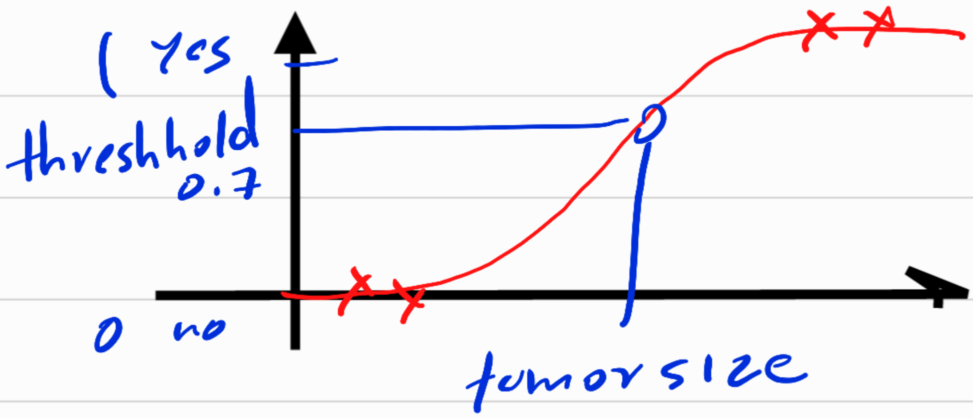
Classification

binary classification → two values

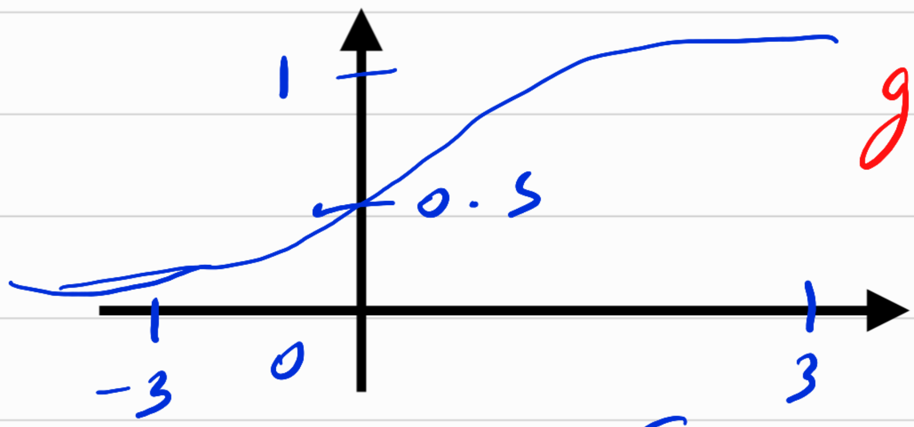
Problem with gradient descent



Logistic Regression



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



$0 < g(z) < 1$

Sigmoid Function

logistic function

$$f_{\vec{w},b}(\vec{x}) - \quad z = \vec{w}\cdot\vec{x}+b$$

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

$$\therefore f_{\vec{w},b}(\vec{x}) = g(\underbrace{\vec{w}\cdot\vec{x}+b}_{z})$$

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

logistic regression

eg:-

$f_{\vec{w},b}(\vec{x}) = 0.7$

$\hookrightarrow$ 70% chance that y is 1

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

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

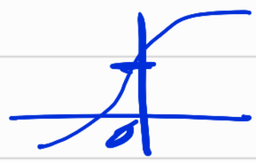
Decision Boundary

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

when

$$f_{\vec{w},b}(\vec{x}) \geq 0.5 \ ?$$

$$g(z) \geq 0.5$$



$$z \geq 0$$

$$\vec{w}\vec{x}+b \geq 0$$

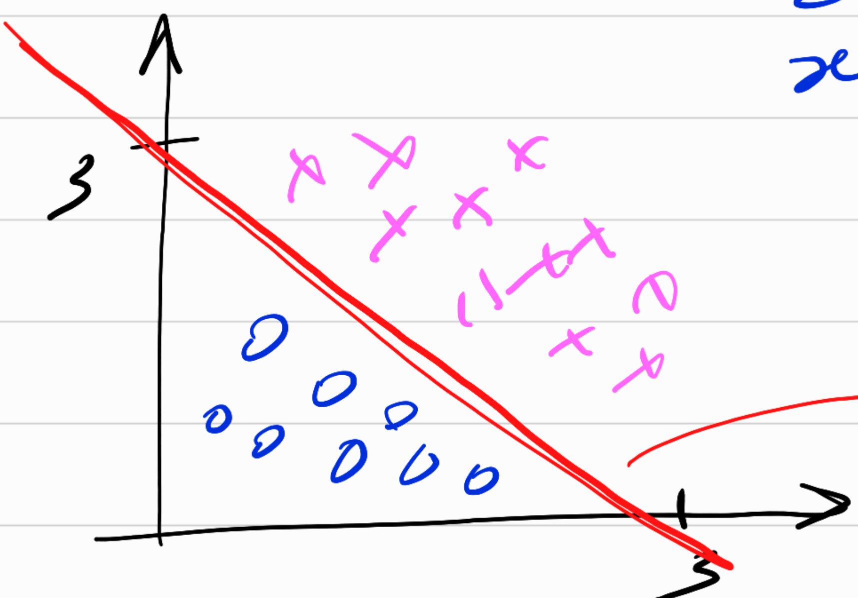
Decision boundary $\vec{w}\cdot\vec{x}+b=0$

eg:-

$$g(z) = g(\underset{1}{w_1} x_1 + \underset{1}{w_2} x_2 + \underset{-3}{b})$$

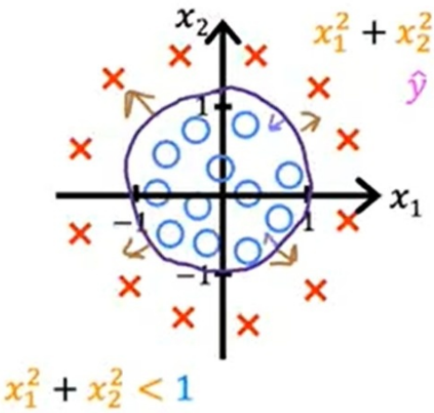
$$\Rightarrow x_1 + x_2 - 3 = 0$$

$$x_1 + x_2 = 3$$



dec. boundary

Non-linear decision boundaries



$$f_{\vec{w},b}(\vec{x}) = g(z) = g(\overbrace{\underset{1}{w_1}x_1^2 + \underset{1}{w_2}x_2^2 + \underset{-1}{b}}^{z})$$

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

$$x_1^2 + x_2^2 = 1$$

* threshhold doesn't need to be 0.5 $\longrightarrow$ can lower to flag early

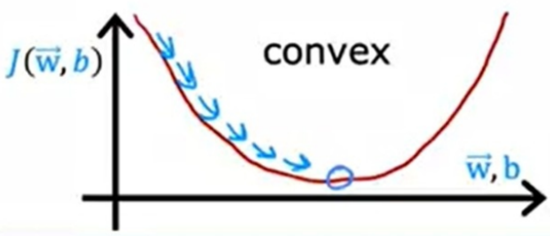
Cost Function

Squared error cost - doesn't work ↓

$$J(\vec{w}, b) = \frac{1}{m}\sum_{i=1}^{m}\frac{1}{2}(f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)})^2$$

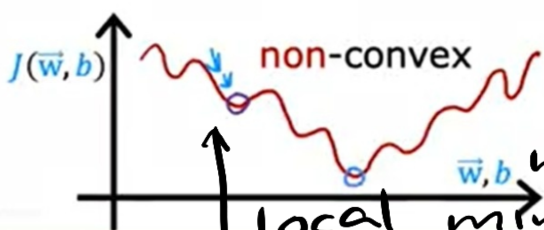
linear regression

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



logistic regression

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

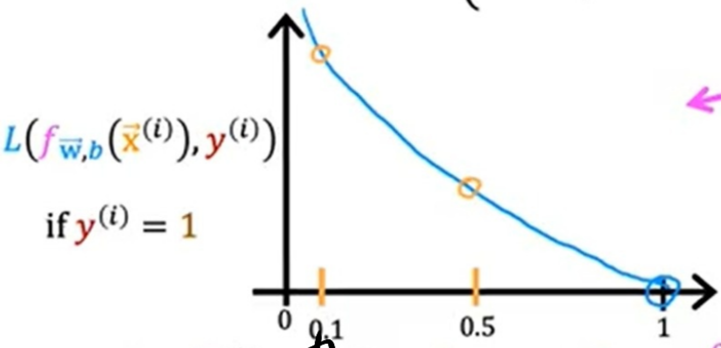


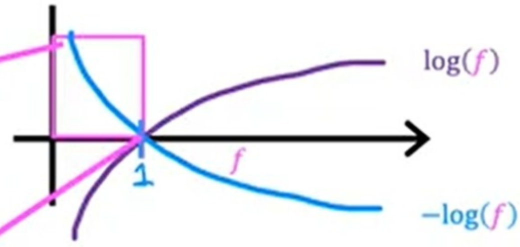
local minimas when g.d.

If $y = 1$

Logistic loss function

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





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

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

As $f_{\vec{w},b}(\vec{x}^{(i)}) \to 0$ then loss → [redacted]

predicts not to say model a bigger loss

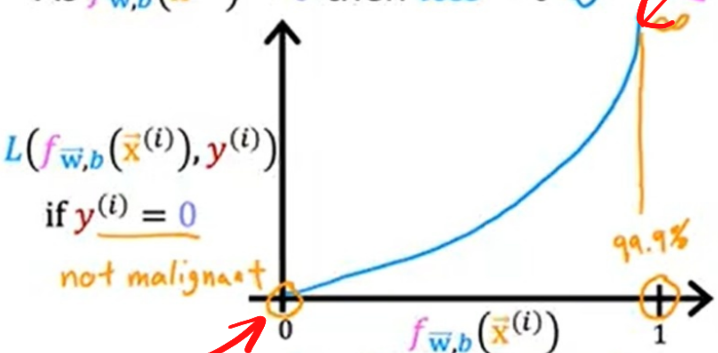
$y = 0$ if close to 1 → ∞ loss

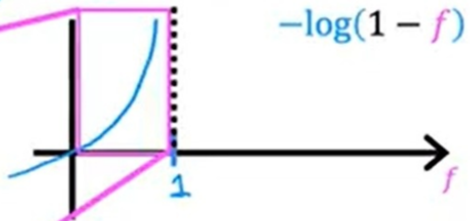
Logistic loss function

prediction differs from target

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

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





matches target

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

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

The loss function above can be rewritten to be easier to implement.

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

This is a rather formidable-looking equation. It is less daunting when you consider $y^{(i)}$ can have only two values, 0 and 1. One can then consider the equation in two pieces:
when $y^{(i)} = 0$, the left-hand term is eliminated:

$$\begin{aligned} loss(f_{\mathbf{w},b}(\mathbf{x}^{(i)}), 0) &= (-(0)\log\left(f_{\mathbf{w},b}\left(\mathbf{x}^{(i)}\right)\right) - (1-0)\log\left(1 - f_{\mathbf{w},b}\left(\mathbf{x}^{(i)}\right)\right) \\ &= -\log\left(1 - f_{\mathbf{w},b}\left(\mathbf{x}^{(i)}\right)\right) \end{aligned}$$

and when $y^{(i)} = 1$, the right-hand term is eliminated:

$$\begin{aligned} loss(f_{\mathbf{w},b}(\mathbf{x}^{(i)}), 1) &= (-(1)\log\left(f_{\mathbf{w},b}\left(\mathbf{x}^{(i)}\right)\right) - (1-1)\log\left(1 - f_{\mathbf{w},b}\left(\mathbf{x}^{(i)}\right)\right) \\ &= -\log\left(f_{\mathbf{w},b}\left(\mathbf{x}^{(i)}\right)\right) \end{aligned}$$

simplified loss func

$$loss = -y^{(i)}\log(f_{\vec{w},b}(x^{i})) - (1-y^{(i)})\log(1-f_{\vec{w},b}(x^{i}))$$

Cost func

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

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

(maximum likelihood estimation)

also func is convex

Gradient Descent Implementation

$$J(\vec{w},b) = -\frac{1}{m}\sum_{i=1}^{m} - - - - -$$

repeat

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

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

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

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

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