deeplearning.ai

Basics of Neural Network Programming

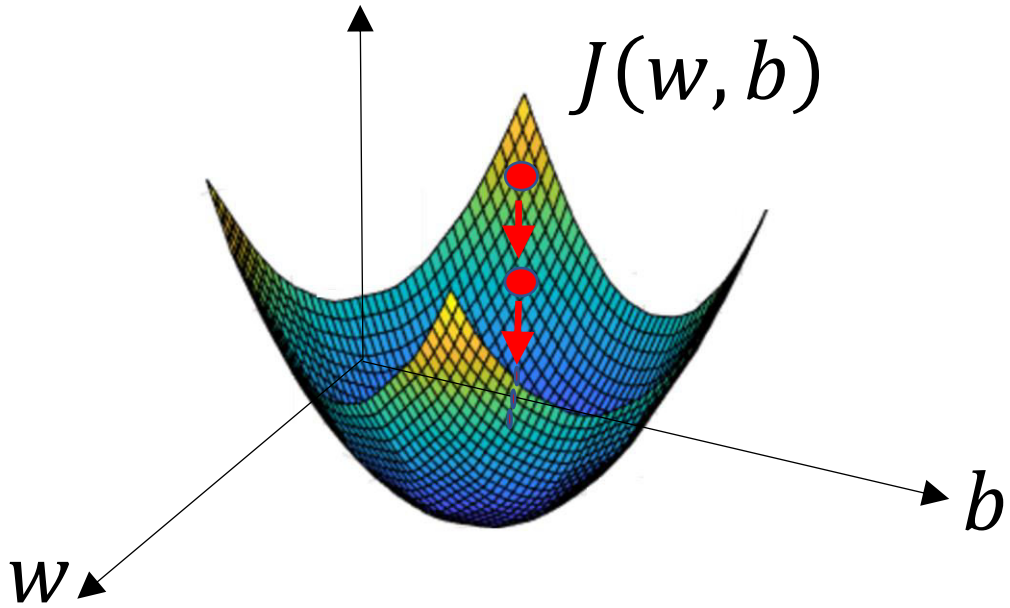
Gradient Descent

Gradient Descent

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

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

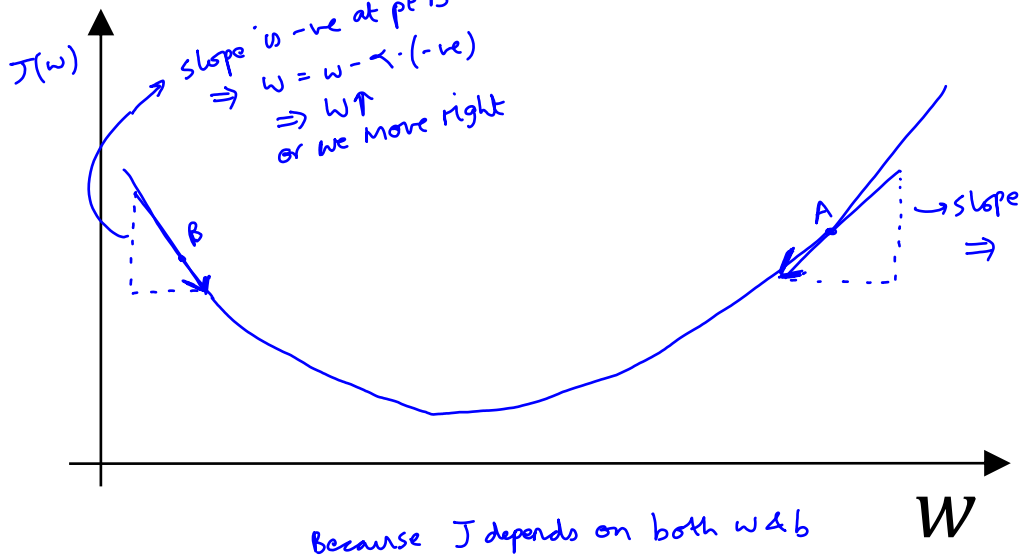
Want to find w, b that minimize $J(w,b)$



Andrew Ng

Gradient Descent

Repeat { $w := w - \alpha \cdot \frac{\partial J(w)}{\partial w}$ } (learning rate)



slope is -ve at pt B
$\Rightarrow w = w - \alpha \cdot (-ve)$
$\Rightarrow w \uparrow$
or we move right

slope is +ve at pt A
$\Rightarrow w = w - \alpha \cdot (+ve)$
$\Rightarrow w \downarrow$
or we move left

Because J depends on both w & b
$\Rightarrow$ There are 2 updates that need to happen, for w & b

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

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

Basically $\frac{\partial J(w,b)}{\partial w}$, means how much does this func $J(w,b)$ slope in the direction of w

convention
If J is a function of 2 or more params, say w & b we use "partial derivatives" or "∂"
$\frac{\partial J(w,b)}{}$
=
If J was a func only of w you use "full derivative" or "d"
$\frac{dJ(w)}{}$