



deeplearning.ai

# Basics of Neural Network Programming

---

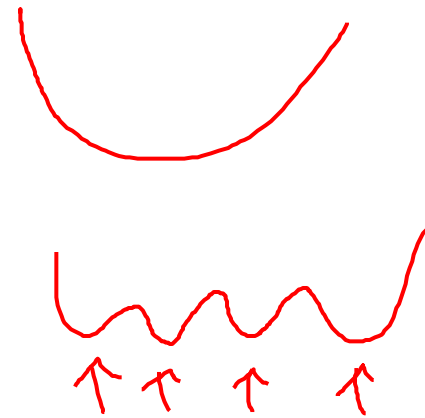
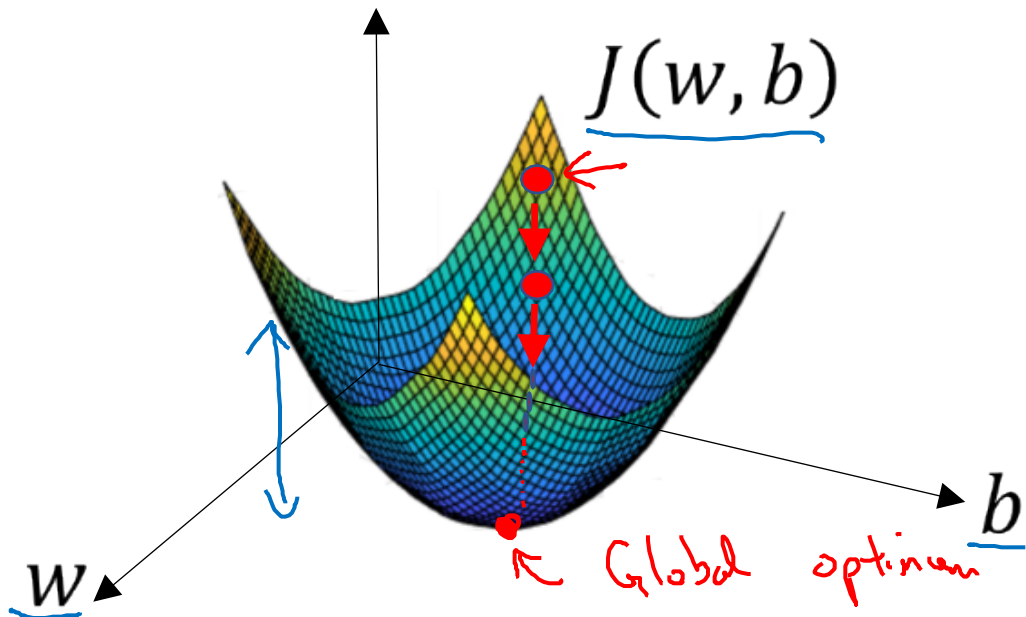
## Gradient Descent

# Gradient Descent

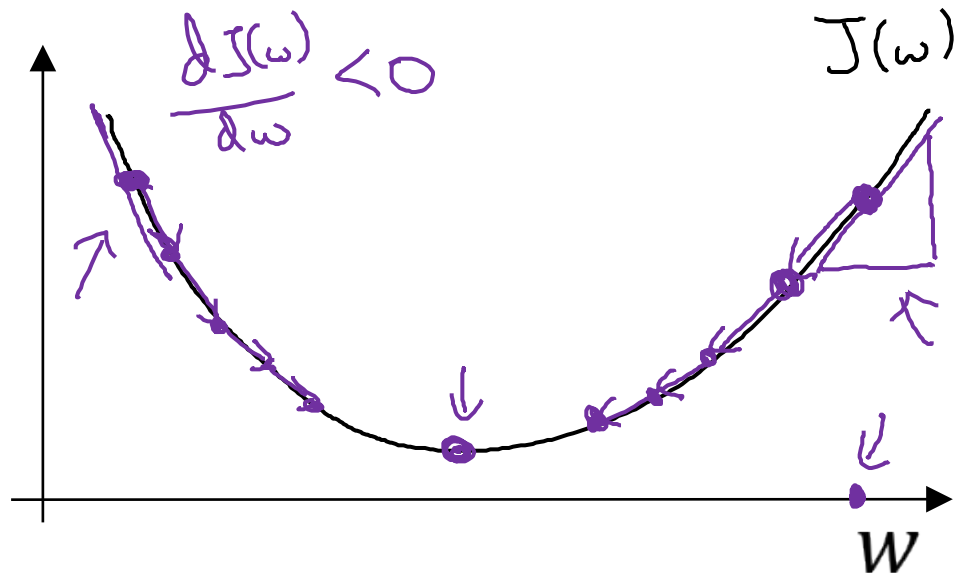
Recap:  $\hat{y} = \sigma(w^T x + b)$ ,  $\sigma(z) = \frac{1}{1+e^{-z}}$   $\leftarrow$

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

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



# Gradient Descent



Repeat {

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

learning rate

}

$$w := w - \alpha \underbrace{dw}_{\text{"dw"}}$$

$$\frac{dJ(w)}{dw} = ?$$

$$J(w, b)$$

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

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

$$\frac{\partial J(w, b)}{\partial w}$$

$$\frac{\partial J(w, b)}{\partial b}$$

$$\frac{\partial}{\partial}$$

$$\frac{\partial}{\partial}$$

"partial  
derivative"  
J

$dw$

$db$