Basics of Neural Network
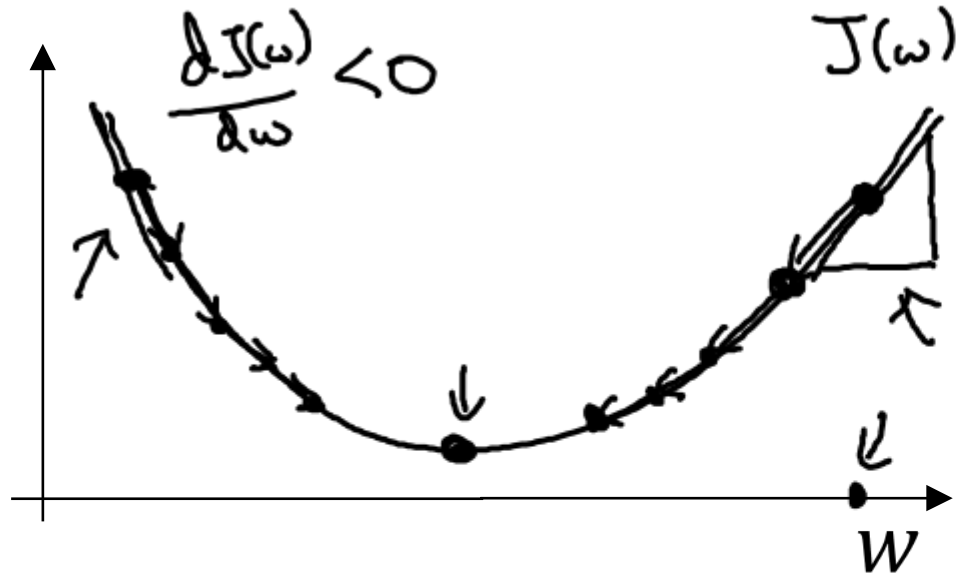
Programming

Gradient Descent

deeplearning.ai

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 $J(w,b)$

$J(w,b)$

← Global optimum

b

w

Andrew Ng

Gradient Descent

$$\frac{dJ(w)}{dw} < 0 \qquad J(w)$$

Repeat {

$$w := w - \alpha \boxed{\frac{dJ(w)}{dw}} \quad \leftarrow \text{learning rate}$$

$$\uparrow \uparrow$$

}
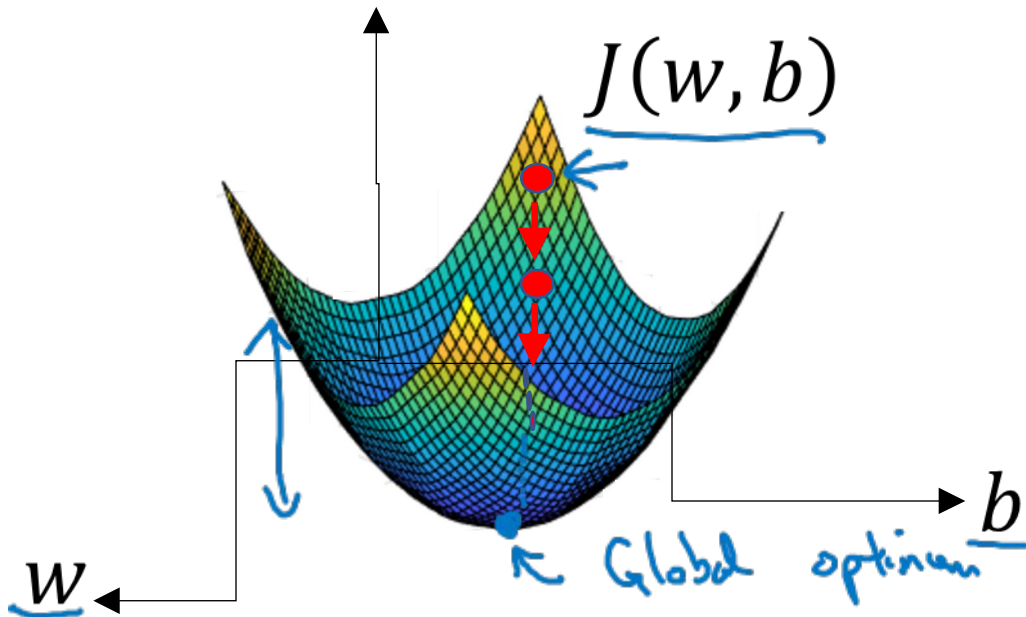
$$w := w - \alpha \, dw \quad "dw"$$

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

$$J(w,b)$$

$$w := w - \alpha \boxed{\frac{\partial J(w,b)}{\partial w}} \qquad \boxed{\frac{\partial J(w,b)}{\partial w}} \qquad \textcircled{\partial} \quad "partial \; derivative"$$

$$b := b - \alpha \boxed{\frac{\partial J(w,b)}{\partial b}} \qquad \boxed{\frac{\partial J(w,b)}{\partial b}} \qquad \textcircled{\partial} \leftarrow J$$

$$dw$$

$$db$$

Andrew Ng