Basics of Neural Network Programming

Gradient Descent

deeplearning.ai

Gradient Descent

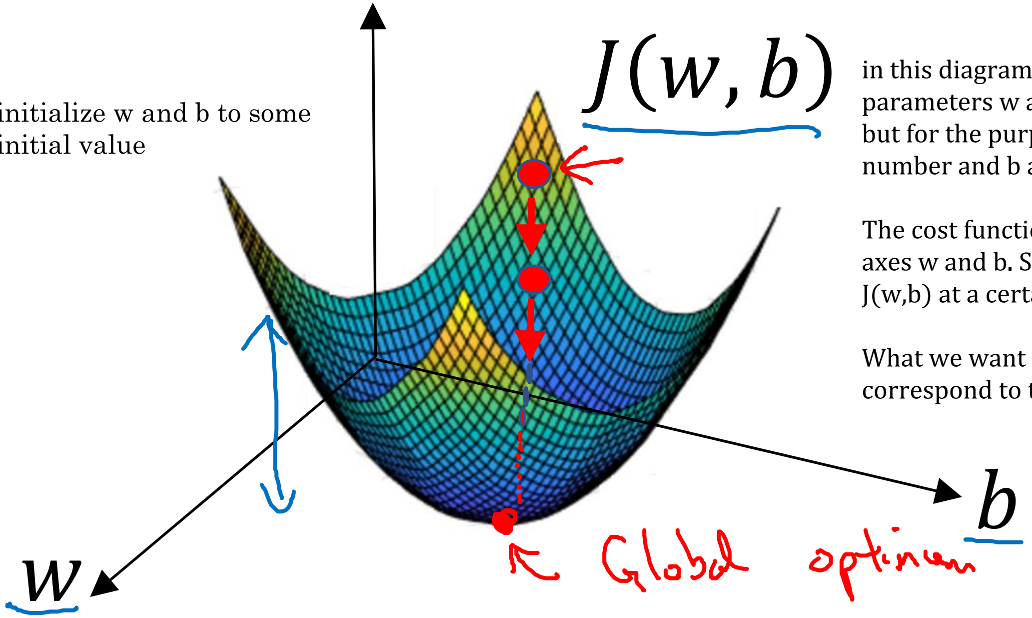
Recap: $\hat{y} = \sigma(w^T x + b), \ \sigma(z) = \frac{1}{1+e^{-z}}$

So the cost function measures how well your parameters w and b are doing on the training set

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

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

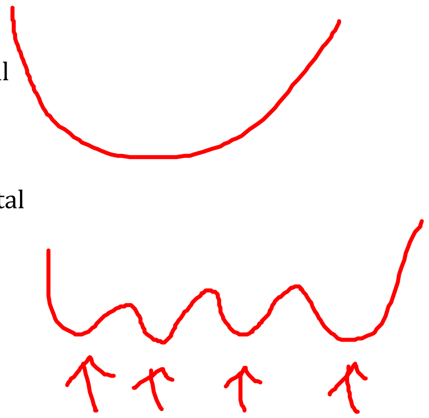
initialize w and b to some initial value



in this diagram the horizontal axes represent your spatial parameters w and b. In practice w can be much higher dimensional but for the purposes of plotting, let's illustrate w as a single real number and b as a single real number.

The cost function J(w,b) is then some surface above these horizontal axes w and b. So the height of the surface represent the value of J(w,b) at a certain point.

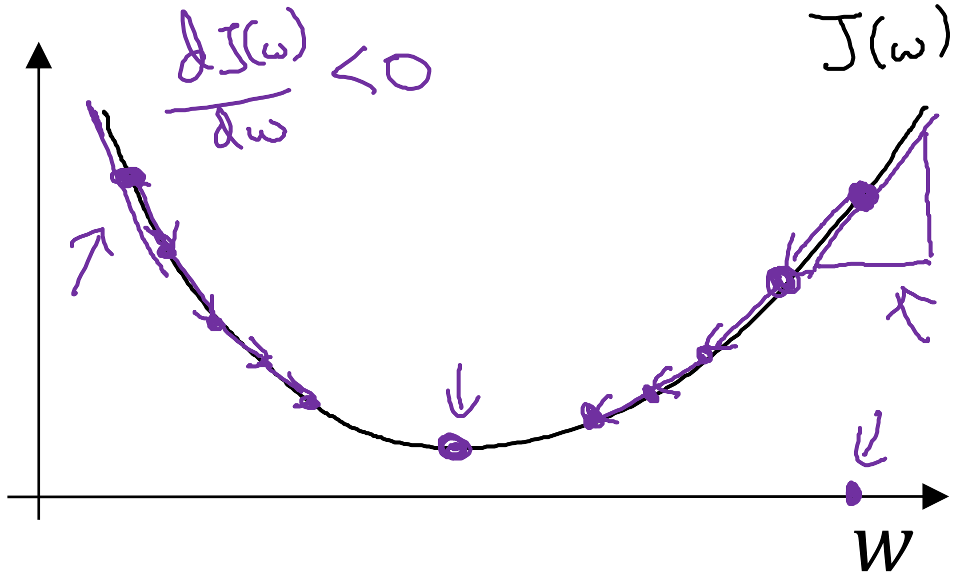
What we want to do is really find the value of w and b that correspond to the minimum of the cost function J.



for logistic regression almost any initialization method works, initialize the value to zero, but people don't usually do that for logistic regression.

Andrew Ng

Gradient Descent



controls how big a step we take on each iteration or gradient descent

learning rate

Repeat {

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

}

"dw" derivative term

$$w := w - \alpha \, dw$$

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

$J(w,b)$

$$w := w - \alpha \boxed{\frac{dJ(w,b)}{dw}} \quad \boxed{\frac{\partial J(w,b)}{\partial w}} \rightarrow dw$$

$$b := b - \alpha \boxed{\frac{dJ(w,b)}{db}} \quad \boxed{\frac{\partial J(w,b)}{\partial b}} \rightarrow db$$

∂ "partial derivative"

d ← J

implement in code

Andrew Ng