Previous notation

Parameters: $w_1, \ldots, w_n$

Model: $f_{\vec{w},b}(\vec{x}) = w_1 x_1 + \ldots + w_n x_n + b$

Cost Function: $J(w_1, \ldots, w_n, b)$

Gradient Descent:

repeat {

$$w_j = w_j - \alpha \frac{\partial}{\partial w_j} J(w_1, \ldots, w_n, b)$$

$$b = b - \alpha \frac{\partial}{\partial b} J(w_1, \ldots, w_n, b)$$

}

Vector Notation

$\vec{w} = [w_1 \ \ldots \ w_n]$

b is still a number

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

$J(\vec{w}, b)$

repeat {

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

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

Gradient Descent

One feature

repeat {

$$w = w - \alpha \boxed{\frac{1}{m} \sum_{i=1}^{m} \left(f_{w,b}(x^{(i)}) - y^{(i)} \right) x^{(i)}}$$

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

$$b = b - \alpha \boxed{\frac{1}{m} \sum_{i=1}^{m} \left(f_{w,b}(x^{(i)}) - y^{(i)} \right)}$$

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

simultaneously update w and b

}

Multiple features

repeat {

$$w_1 = w_1 - \alpha \boxed{\frac{1}{m} \sum_{i=1}^{m} (f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)}) x_1^{(i)}}$$

$\frac{\partial J(\vec{w},b)}{\partial w_1}$

$\vdots$

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

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

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

simultaneously update w (for $j = 1, \ldots, n$) and b

}

An alternative to gradient descent

→ normal equation

- only for linear regression
- solves for w, b without iterations
  - uses an advanced linear algebra library to do so

It maybe used in machine learning libraries that implement linear regression

but for most, gradient descent is better

Disadvantages

- Doesn't generalize to other learning algorithms (like logistical regression, neural networks)

- Slow when number of features is large ($> 10,000$)