We use gradient descent for minimizing cost function in logistics regression also.

$$J(\vec{w}, b) = \frac{1}{m}\sum_{i=1}^{m} -y^{(i)} \log\left(\underbrace{f_{\vec{w},b}(\vec{x}^{(i)})}_{\frac{1}{1+e^{-(w\vec{x}+b)}}}\right) - (1-y^{(i)}) \log\left(f_{\vec{w},b}(\vec{x}^{(i)})\right)$$

repeat {

$j = 1 \ldots n$

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

particular x from i^{th} training example multiple features

$$\frac{\partial}{\partial w_j} = \frac{1}{m}\sum_{i=1}^{m}\left(f_{\vec{w},b}(x^{(i)}) - y^{(i)}\right) \underline{x_j^{(i)}}$$

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

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

} simultaneous updates

It looks like linear regression but is not

Same concepts :-

- Monitor gradient descent (learning curve)
- Vectorized implementation
- Feature scaling