You can pick up where you left off. Just join a new session and we'll reset your deadlines.

Join a session

∢ Back to Week 3

X Lessons

This Course: Machine Learning

Prev

Next

Simplified Cost Function and Gradient Descent

Note: [6:53 - the gradient descent equation should have a 1/m factor]

We can compress our cost function's two conditional cases into one case:

$$\operatorname{Cost}(h_{ heta}(x),y) = -y \ \log(h_{ heta}(x)) - (1-y) \log(1-h_{ heta}(x))$$

Notice that when y is equal to 1, then the second term $(1-y)\log(1-h_{ heta}(x))$ will be zero and will not affect the result. If y is equal to 0, then the first term $-y \log(h_{\theta}(x))$ will be zero and will not affect the result.

We can fully write out our entire cost function as follows:

$$J(heta) = -rac{1}{m} \sum_{i=1}^m [y^{(i)} \log(h_{ heta}(x^{(i)})) + (1-y^{(i)}) \log(1-h_{ heta}(x^{(i)}))]$$

A vectorized implementation is:

$$h = g(X\theta)$$

$$J(\theta) = \frac{1}{m} \cdot \left(-y^T \log(h) - (1 - y)^T \log(1 - h)\right)$$

Gradient Descent

Remember that the general form of gradient descent is:

Repeat {
$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

We can work out the derivative part using calculus to get:

Repeat {
$$\theta_{j} := \theta_{j} - \frac{\alpha}{m} \sum_{i=1}^{m} (h_{\theta}(x^{(i)}) - y^{(i)}) x_{j}^{(i)}$$
}

Notice that this algorithm is identical to the one we used in linear regression. We still have to simultaneously update all values in

A vectorized implementation is:

$$heta := heta - rac{lpha}{m} X^T (g(X heta) - ec{y})$$

✓ Complete



