**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:

Cost(*hθ*(*x*),*y*)=−*y*log(*hθ*(*x*))−(1−*y*)log(1−*hθ*(*x*))

Notice that when y is equal to 1, then the second term (1-y)\log(1-h\_\theta(x))(1−*y*)log(1−*hθ*​(*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))−*y*log(*hθ*​(*x*)) will be zero and will not affect the result.

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

|  |
| --- |
| J(\theta) = - \frac{1}{m} \displaystyle \sum\_{i=1}^m [y^{(i)}\log (h\_\theta (x^{(i)})) + (1 - y^{(i)})\log (1 - h\_\theta(x^{(i)}))]*J*(*θ*)=−*m*1​*i*=1∑*m*​[*y*(*i*)log(*hθ*​(*x*(*i*)))+(1−*y*(*i*))log(1−*hθ*​(*x*(*i*)))] |

A vectorized implementation is:

|  |
| --- |
| *h*=*g*(*Xθ*)*J*(*θ*)=1*m*⋅(−*yT*log(*h*)−(1−*y*)*T*log(1−*h*)) |

**Gradient Descent**

Remember that the general form of gradient descent is:

|  |
| --- |
| *Repeat*{*θj*:=*θj*−*α*∂∂*θjJ*(*θ*)} |

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

|  |
| --- |
| *Repeat*{*θj*:=*θj*−*αm*∑*i*=1*m*(*hθ*(*x*(*i*))−*y*(*i*))*x*(*i*)*j*} |

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

A vectorized implementation is:

\theta := \theta - \frac{\alpha}{m} X^{T} (g(X \theta ) - \vec{y})*θ*:=*θ*−*mα*​*XT*(*g*(*Xθ*)−*y*​)