**Cost Function and Backpropagation**

- **Video:** Cost Function
  6 min
- **Reading:** Cost Function
  4 min
- **Video:** Backpropagation Algorithm
  11 min
- **Reading:** Backpropagation Algorithm
  10 min
- **Video:** Backpropagation Intuition
  12 min
- **Reading:** Backpropagation Intuition
  4 min

**Backpropagation in Practice**

**Application of Neural Networks**

**Review**

# Backpropagation Algorithm

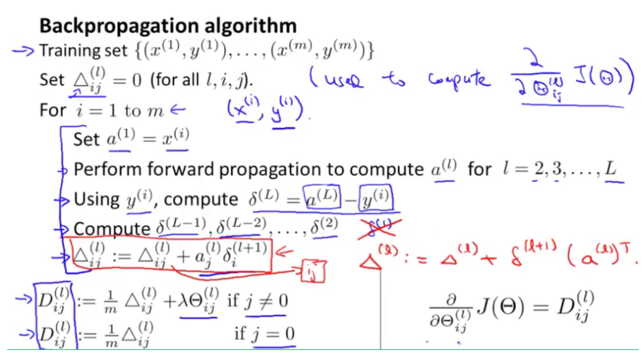
"Backpropagation" is neural-network terminology for minimizing our cost function, just like what we were doing with gradient descent in logistic and linear regression. Our goal is to compute:

$$\min_\Theta J(\Theta)$$

That is, we want to minimize our cost function J using an optimal set of parameters in theta. In this section we'll look at the equations we use to compute the partial derivative of J(Θ):

$$\frac{\partial}{\partial \Theta_{i,j}^{(l)}} J(\Theta)$$

To do so, we use the following algorithm:




### Back propagation Algorithm

Given training set $\{(x^{(1)}, y^{(1)}) \cdots (x^{(m)}, y^{(m)})\}$

- Set $\Delta_{i,j}^{(l)}$ := 0 for all (l,i,j), (hence you end up having a matrix full of zeros)

For training example t =1 to m:

1. Set $a^{(1)} := x^{(t)}$