#### coursera

### Q

# Cost Function and Backpropagation

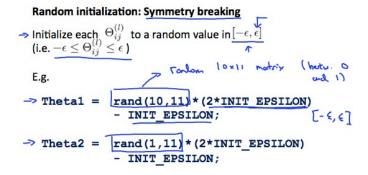
- Video: Cost Function 6 min
- Reading: Cost Function 4 min
- Video: Backpropagation
  Algorithm
  11 min
- Reading: Backpropagation
  Algorithm
  10 min
- Video: Backpropagation
  Intuition
  12 min
- Reading: Backpropagation
  Intuition
  4 min

## Backpropagation in Practice

- Video: Implementation
  Note: Unrolling Parameters
  7 min
- Reading: Implementation
  Note: Unrolling Parameters
  3 min
- Video: Gradient Checking
  11 min
- Reading: Gradient Checking 3 min
- Video: Random Initialization 6 min
- Reading: Random Initialization 3 min

### Random Initialization

Initializing all theta weights to zero does not work with neural networks. When we backpropagate, all nodes will update to the same value repeatedly. Instead we can randomly initialize our weights for our  $\Theta$  matrices using the following method:



Hence, we initialize each  $\Theta_{ij}^{(l)}$  to a random value between  $[-\epsilon,\epsilon]$ . Using the above formula guarantees that we get the desired bound. The same procedure applies to all the  $\Theta$ 's. Below is some working code you could use to experiment.

```
1  If the dimensions of Theta1 is 10x11, Theta2 is 10x11
2
3  Theta1 = rand(10,11) * (2 * INIT_EPSILON) - INIT_EPSI
4  Theta2 = rand(10,11) * (2 * INIT_EPSILON) - INIT_EPSI
5  Theta3 = rand(1,11) * (2 * INIT_EPSILON) - INIT_EPSIL6
```

rand(x,y) is just a function in octave that will initialize a matrix of random real numbers between 0 and 1.

(Note: the epsilon used above is unrelated to the epsilon from Gradient Checking)

✓ Complete

Go to next item

1 of 1