***NNs – Learning - QUIZ***

* You are training a 3 layer NN + would like to use backprop to compute the gradient of the cost function J(ϴ) In the backprop algorithm, 1 of the steps is to update:



for every (i,j). Which of the following is a correct vectorization of this step?

* 
* Suppose Theta1 is a 5x3 matrix, and Theta2 is a 4x6 matrix. You set thetaVec = [Theta1(:);Theta2(:)]. Which of the following correctly recovers Theta2?
* **reshape(thetaVec(16:39),4,6)**
* Let J(θ) = 3θ^4 + 4. Let θ = 1, and ϵ = 0.01. Use the formula (J(θ + ϵ) − J(θ − ϵ)) / 2ϵ to numerically compute an approximation to the derivative at θ = 1. What value do you get? (When θ = 1, the true/exact derivative is dJ(θ)/dθ = 12)
* **12.0012**
* Which of the following statements are true? Check all that apply.
* **Using gradient checking can help verify if an implementation of backprop is bug-free.**
* **Computing the gradient of the cost function in a NN does NOT have the same efficiency when we use backprop vs. when we numerically compute it using the method of gradient checking.**
* **Gradient checking is useful if we are using EITHER gradient descent or one of the advanced optimization methods (such as in fminunc) as our optimization algorithm.**
* **For computational efficiency, after performing gradient checking to verify that our backprop code is correct, we disable gradient checking before using backprop to train the network.**
* Which of the following statements are true? Check all that apply.
* **If training a NN using gradient descent, one reasonable "debugging" step to make sure it is working is to plot J(Θ) as a function of the number of iterations, and make sure it is decreasing (or at least non-increasing) after each iteration.**
* **Suppose we have a correct implementation of backprop + are training a NN using gradient descent. Suppose we plot J(Θ) as a function of the number of iterations + find it is increasing rather than decreasing. One possible cause of this is that the learning rate α is too large.**
* **Suppose the parameter Θ(1) is a square matrix (number of rows = number of columns). If we replace Θ(1) w/ its transpose (Θ(1))T, we have changed the function the network is computing**