Feedback — IX. Neural Networks: Learning

You submitted this quiz on **Wed 22 May 2013 11:13 AM PDT (UTC -0700)**. You got a score of **5.00** out of **5.00**.

Question 1

You are training a three layer neural network and would like to use backpropagation to compute the gradient of the cost function. In the backpropagation algorithm, one of the steps is to update $\Delta^{(2)}_{ij} := \Delta^{(2)}_{ij} + \delta^{(3)}_i * (a^{(2)})_j$ for every i, j. Which of the following is a correct vectorization of this step?

Your Answer		Score	Explanation
$\Delta^{(2)} := \Delta^{(2)} + \delta^{(2)} * (a^{(3)})^T$			
$\Delta^{(2)} := \Delta^{(2)} + (a^{(3)})^T * \delta^{(3)}$			
$\Delta^{(2)} := \Delta^{(2)} + \delta^{(3)} * (a^{(2)})^T$	✔	1.00	This version is correct, as it takes the "outer product" of the two vectors $\delta^{(3)}$ and $a^{(2)}$ which is a matrix such that the (i, j)-th entry is $\delta^{(3)}_i * (a^{(2)})_j$ as desired.
$\Delta^{(2)} := \Delta^{(2)} + \delta^{(2)} * (a^{(2)})^T$			
Total		1.00 / 1.00	

Question 2

Suppose `Theta1` is a 6x2 matrix, and `Theta2` is a 2x7 matrix. You set `thetaVec = [Theta1(:) ; Theta2(:)]`. Which of the following correctly recovers `Theta2`?

Your Answer		Score	Explanation
`reshape(thetaVec(12:26), 2, 7)`			
`reshape(thetaVec(11:24), 2, 7)`			
`reshape(thetaVec(13:26), 2, 7)`	✔	1.00	This choice is correct, since `Theta1` has 12 elements, so `Theta2` begins at index 13 and ends at index 13 + 14 - 1 = 26.
`reshape(thetaVec(13:24), 2, 7)`			
Total		1.00 / 1.00	

Question 3

Let $J(\theta) = 3\theta^3 + 2$. Let $\theta = 1$, and $\epsilon = 0.01$. Use the formula $\frac{J(\theta+\epsilon)-J(\theta-\epsilon)}{2\epsilon}$ to numerically compute an approximation to the derivative at $\theta = 1$. What value do you get? (When $\theta = 1$, the true/exact derivati ve is $\frac{dJ(\theta)}{d\theta} = 9$.)

Your Answer	Score	Explanation
11		

◉ 9.0003	✔	1.00	We compute $\frac{(3(1.01)^3+2)-(3(0.99)^3+2)}{2(0.01)} = 9.0003$.
○ 9			
○ 8.9997			
Total		1.00 / 1.00	

Question 4

Which of the following statements are true? Check all that apply.

Your Answer		Score	Explanation
☑ If our neural network overfits the training set, one reasonable step to take is to increase the regularization parameter λ.	✔	0.25	Just as with logistic regression, a large value of λ will penalize large parameter values, thereby reducing the changes of overfitting the training set.
☐ Computing the gradient of the cost function in a neural network has the same efficiency when we use backpropagation or when we numerically compute it using the method of gradient checking.	✔	0.25	Numerical gradient checking is much slower, as you must perform forward propagation twice for every parameter in the newtork.
☐ Gradient checking is useful if we are using one of the advanced optimization methods (such as in `fminunc`) as our optimization algorithm. However, it serves little purpose if we are using gradient descent.	✔	0.25	Gradient descent depends on the computation of correct gradient values at different parameter settings. Gradient checking ensures the computed values are correct.
☑ For computational efficiency, after we have performed gradient checking to verify that our	✔	0.25	Checking the gradient numerically is a debugging

backpropagation code is correct, we usually disable gradient checking before using backpropagation to train the network.			tool: it helps ensure a corre ct implementation, but it is too slow to use as a method for actually computing gradients.
Total		1.00 / 1.00	

Question 5

Which of the following statements are true? Check all that apply.

Your Answer		Score	Explanation
☑ Suppose we have a correct implementation of backpropagation, and are training a neural network using gradient descent. Suppose we plot $J(\Theta)$ as a function of the number of iterations, and find that it is **increasing** rather than decreasing. One possible cause of this is that the learning rate α is too large.	✔	0.25	If the learning rate is too large, the cost function can diverge during gradient descent. Thus, you should select a smaller value of α.
☐ If we initialize all the parameters of a neural network to ones instead of zeros, this will suffice for the purpose of "symmetry breaking" because the parameters are no longer symmetrically equal to zero.	✔	0.25	The trouble with initializing the parameters to all zeros is not the specific value of zero but instead that every unit in the network will get the same update after backpropagation. Initializing the parameters to all ones has the same difficulty.

☐ Suppose you have a three layer network with parameters $\Theta^{(1)}$ (controlling the function mapping from the inputs to the hidden units) and $\Theta^{(2)}$ (controlling the mapping from the hidden units to the outputs). If we set all the elements of $\Theta^{(1)}$ to be 0, and all the elements of $\Theta^{(2)}$ to be 1, then this suffices for symmetry breaking, since the neurons are no longer all computing the same function of the input.	✔	0.25	Since the parameters are the same within layers, every unit in each layer will receive the same update during backpropagation. The result is that such an initialization does not break symmetry.
☑ Suppose you are training a neural network using gradient descent. Depending on your random initialization, your algorithm may converge to different local optima (i.e., if you run the algorithm twice with different random initializations, gradient descent may converge to two different solutions).	✔	0.25	The cost function for a neural network is non-convex, so it may have multiple minima. Which minimum you find with gradient descent depends on the initialization.
Total		1.00 / 1.00	