Neural Networks: Learning

Total points 5

1. You are training a three layer neural network and would like to use backpropagation to compute the gradient   1 point
    of the cost function. In the backpropagation algorithm, one of the steps is to update

    $$\Delta_{ij}^{(2)} := \Delta_{ij}^{(2)} + \delta_i^{(3)} * (a^{(2)})_j$$

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

    ⦿ $\Delta^{(2)} := \Delta^{(2)} + \delta^{(3)} * (a^{(2)})^T$

    ○ $\Delta^{(2)} := \Delta^{(2)} + (a^{(2)})^T * \delta^{(3)}$

    ○ $\Delta^{(2)} := \Delta^{(2)} + (a^{(3)})^T * \delta^{(2)}$

    ○ $\Delta^{(2)} := \Delta^{(2)} + \delta^{(3)} * (a^{(3)})^T$

2. Suppose `Theta1` is a 5x3 matrix, and `Theta2` is a 4x6 matrix. You set `thetaVec` = $[\text{Theta1}(:); \text{Theta2}(:)]$.   1 point
    Which of the following correctly recovers `Theta2`?

    ⦿ $\text{reshape}(\text{thetaVec}(16:39), 4, 6)$

    ○ $\text{reshape}(\text{thetaVec}(15:38), 4, 6)$

    ○ $\text{reshape}(\text{thetaVec}(16:24), 4, 6)$

    ○ $\text{reshape}(\text{thetaVec}(15:39), 4, 6)$

    ○ $\text{reshape}(\text{thetaVec}(16:39), 6, 4)$

3. Let $J(\theta) = 3\theta^4 + 4$. Let $\theta = 1$, and $\epsilon = 0.01$. Use the formula $\frac{J(\theta+\epsilon) - J(\theta-\epsilon)}{2\epsilon}$ to numerically compute an   1 point
    approximation to the derivative at $\theta = 1$. What value do you get? (When $\theta = 1$, the true/exact derivative is $\frac{dJ(\theta)}{d\theta} = 12$.)

    ○ 12

    ○ 6

    ○ 11.9988

    ⦿ 12.0012

4. Which of the following statements are true? Check all that apply.   1 point

    ☐ 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.

    ☑ Using gradient checking can help verify if one's implementation of backpropagation is bug-free.

    ☑ For computational efficiency, after we have performed gradient checking to

    verify that our backpropagation code is correct, we usually disable gradient checking before using
    backpropagation to train the network.

    ☐ 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.

5. Which of the following statements are true? Check all that apply.   1 point

    ☐ 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.

    ☐ 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.

    ☑ If we are training a neural network using gradient descent, one reasonable "debugging" step to make sure it is
    working is to plot $J(\Theta)$ as a function of the number of iterations, and make sure it is decreasing (or at least
    non-increasing) after each iteration.

    ☑ 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).

Coursera Honor Code Learn more

[Submit]   [⚙ Saving...]

👍 **Like**   👎 **Dislike**   ⚐ **Report an issue**