Feedback — XVII. Large Scale Machine Learning

You submitted this quiz on **Thu 20 Jun 2013 1:16 PM PDT (UTC -0700)**. You got a score of **5.00** out of **5.00**.

Question 1

Suppose you are training a logistic regression classifier using stochastic gradient descent. You find that the cost (say, $cost(\theta, (x^{(i)}, y^{(i)}))$, averaged over the last 500 examples), plotted as a function of the number of iterations, is slowly increasing over time. Which of the following changes are likely to help?

Your Answer		**Score**	**Explanation**
○ Try averaging the cost over a larger number of examples (say 1000 examples instead of 500) in the plot.			
○ Try using a larger learning rate α.			
● Try halving (decreasing) the learning rate α, and see if that causes the cost to now consistently go down; and if not, keep halving it until it does.	✓	1.00	Such a plot indicates that the algorithm is diverging. Decreasing the learning rate α means that each iteration of stochastic gradient descent will take a smaller step, thus it will likely converge instead of diverging.
○ This is not possible with stochastic gradient descent, as it is guaranteed to converge to the optimal parameters θ.			
Total		1.00 / 1.00	

Question 2

Which of the following statements about stochastic gradient descent are true? Check all that apply.

Your Answer		Score	Explanation
☑ If you have a huge training set, then stochastic gradient descent may be much faster than batch gradient descent.	✓	0.25	Because stochastic gradient descent can make progress after only a few examples, it can converge much more quickly than batch gradient descent.
□ In order to make sure stochastic gradient descent is converging, we typically compute $J_{\text{train}}(\theta)$ after each iteration (and plot it) in order to make sure that the cost function is generally decreasing.	✓	0.25	We want to plot $cost(\theta, (x^{(i)}, y^{(i)}))$ at each iteration, as computing the full summation $J_{\text{train}}(\theta)$ is too expensive.
☑ You can use the method of numerical gradient checking to verify that your stochastic gradient descent implementation is bug-free. (One step of stochastic gradient descent computes the partial derivative $\frac{\partial}{\partial\theta_j} cost(\theta, (x^{(i)}, y^{(i)}))$.)	✓	0.25	Just as with batch gradient descent, you can compute the derivative numerically and compare it to your computed value to check for correctness.
□ Suppose you are using stochastic gradient descent to train a linear regression classifier. The cost function $J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$ is guaranteed to decrease after every iteration of the stochastic gradient descent algorithm.	✓	0.25	Since each iteration of stochastic gradient descent takes into account only one training example, it is not guaranteed that every update lowers the cost function over the entire training set.
Total		1.00 / 1.00	

Question 3

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

Your Answer		Score	Explanation
☑ In the approach to online learning discussed in the lecture video, we repeatedly get a single training example, take one step of stochastic gradient descent using that example, and then move on to the next example.	✓	0.25	This is one good approach to online learning discussed in the lecture video.
□ One of the disadvantages of online learning is that it requires a large amount of computer memory/disk space to store all the training examples we have seen.	✓	0.25	Since online learning algorithms do not save old examples, they can be very efficent in terms of computer memory and disk space.
☑ One of the advantages of online learning is that if the function we're modeling changes over time (such as if we are modeling the probability of users clicking on different URLs, and user tastes/preferences are changing over time), the online learning algorithm will automatically adapt to these changes.	✓	0.25	Online learning algorithms move toward correctly classifying the most recent examples, so as user tastes change and we receive new, different data, the algorithm will automatically take those into account.
□ Online learning algorithms are most appropriate when we have a fixed training set of size m that we want to train on.	✓	0.25	It is the opposite: they are most appropriate when we have a stream of training data of unbounded size.
Total		1.00 / 1.00	

Question 4

Assuming that you have a very large training set, which of the following algorithms do you think can be parallelized using map-reduce and splitting the training set across different machines? Check all that apply.

Your Answer		Score	Explanation
☑ Computing the average of all the features in your training set $\mu = \frac{1}{m} \sum_{i=1}^{m} x^{(i)}$ (say in order to perform mean normalization).	✓	0.25	You can split the dataset into N smaller batches, compute the feature average of each smaller batch on one of N separate computers, and then average those results on a central computer to get the final result.
☐ Linear regression trained using stochastic gradient descent.	✓	0.25	Since stochastic gradient descent processes one example at a time and updates the parameter values after each, it cannot be easily parallelized.
☑ Logistic regression trained using batch gradient descent.	✓	0.25	You can split the dataset into N smaller batches, compute the gradient for each smaller batch on one of N separate computers, and then average those gradients on a central computer to use for the gradient update.
☐ Logistic regression trained using stochastic gradient descent.	✓	0.25	Since stochastic gradient descent processes one example at a time and updates the parameter values after each, it cannot be easily parallelized.
Total		1.00 / 1.00	

Question 5

Which of the following statements about map-reduce are true? Check all that apply.

Your Answer		Score	Explanation
☑ If you are have just 1 computer, but your computer has multiple CPUs or multiple cores, then map-reduce might be a viable way to parallelize your	✓	0.25	Treating each core as a separate computer makes map-reduce just

learning algorithm.			as useful with multiple cores as with multiple computers.
☑ Because of network latency and other overhead associated with map-reduce, if we run map-reduce using N computers, we might get less than an N-fold speedup compared to using 1 computer.	✓	0.25	The maximum speedup possible is N-fold, and it is unlikely you will get an N-fold speedup because of the overhead.
☑ If you have only 1 computer with 1 computing core, then map-reduce is unlikely to help.	✓	0.25	Map-reduce is a useful model for parallel computation.
□ Running map-reduce over N computers requires that we split the training set into N^2 pieces.	✓	0.25	Usually, you will split the data into N pieces, but map-reduce does not require a specific division of the data.
Total		1.00 / 1.00	