Close

Large Scale Machine Learning

5 questions

1  
point

1.

Suppose you are training a logistic regression classifier using stochastic gradient descent. You find that the cost (say, *cost*(*θ*,(*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?



Try averaging the cost over a smaller number of examples (say 250 examples instead of 500) in the plot.



Use fewer examples from your training set.



This is not possible with stochastic gradient descent, as it is guaranteed to converge to the optimal parameters *θ*.



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  
point

2.

Which of the following statements about stochastic gradient

descent are true? Check all that apply.



If you have a huge training set, then stochastic gradient descent may be much faster than batch gradient descent.



Before running stochastic gradient descent, you should randomly shuffle (reorder) the training set.



In order to make sure stochastic gradient descent is converging, we typically compute *J*train(*θ*) after each iteration (and plot it) in order to make sure that the cost function is generally decreasing.



One of the advantages of stochastic gradient descent is that it uses parallelization and thus runs much faster than batch gradient descent.

1  
point

3.

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



When using online learning, in each step we get a new example (*x*,*y*), perform one step of (essentially stochastic gradient descent) learning on that example, and then discard that example and move on to the next.



One of the advantages of online learning is that there is no need to pick a learning rate *α*.



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.



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.

1  
point

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.



Linear regression trained using stochastic gradient descent.



Computing the average of all the features in your training set *μ*=1*m*∑*mi*=1*x*(*i*)(say in order to perform mean normalization).



Logistic regression trained using batch gradient descent.



Logistic regression trained using stochastic gradient descent.

1  
point

5.

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



If you have only 1 computer with 1 computing core, then map-reduce is unlikely to help.



When using map-reduce with gradient descent, we usually use a single machine that accumulates the gradients from each of the map-reduce machines, in order to compute the parameter update for that iteration.



Linear regression and logistic regression can be parallelized using map-reduce, but not neural network training.



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.

1 question unanswered

Submit Quiz