Suppose you are facing a supervised learning problem and have a very large dataset (m = 100,000,000). How can you tell if using all of the data is likely to perform much better than using a small subset of the data (say m = 1,000)

There is no need to verify this; using a larger dataset always gives much better performance.

Plot $J_{\text{train}}(\theta)$ as a function of the number of iterations of the optimization algorithm (such as gradient descent).

Plot a learning curve ($J_{\text{train}}(\theta)$ and $J_{\text{CV}}(\theta)$, plotted as a function of m) for some range of values of m (say up to m = 1,000) and verify that the algorithm has bias when m is small.

CORRECT: Plot a learning curve for a range of values of m and verify that the algorithm has high variance when m is small.

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

When the training set size m is very large, stochastic gradient descent can be much faster than gradient descent.

The cost function $J_{\text{train}}(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$ should go down with every iteration of batch gradient descent (assuming a well-tuned learning rate α) but not necessarily with stochastic gradient descent.

Stochastic gradient descent is applicable only to linear regression but not to other models (such as logistic regression or neural networks).

Before beginning the main loop of stochastic gradient descent, it is a good idea to "shuffle" your training data into a random order.

ABD

Suppose you use mini-batch gradient descent on a training set of size m, and you use a mini-batch size of b. The algorithm becomes the same as batch gradient descent if:

b = 1

b = m / 2

b = m

None of the above

C

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

Picking a learning rate α that is very small has no disadvantage and can only speed up learning.

If we reduce the learning rate α (and run stochastic gradient descent long enough), it's possible that we may find a set of better parameters than with larger \alphaα.

If we want stochastic gradient descent to converge to a (local) minimum rather than wander of "oscillate" around it, we should slowly increase \alphaα over time.

If we plot $\mathrm{cost}(\theta, (x^{(i)}, y^{(i)}))$ (averaged over the last 1000 examples) and stochastic gradient descent does not seem to be reducing the cost, one possible problem may be that the learning rate \alphaα is poorly tuned.

BD

Some of the advantages of using an online learning algorithm are:

It can adapt to changing user tastes (i.e., if p(y\vert x;\theta)p(y|x;θ) changes over time).

Correct

There is no need to pick a learning rate \alphaα.

It allows us to learn from a continuous stream of data, since we use each example once then no longer need to process it again.

Correct

It does not require that good features be chosen for the learning task.

Suppose you apply the map-reduce method to train a neural network on ten machines. In each iteration, what will each of the machines do?

Compute either forward propagation or back propagation on 1/5 of the data.

Compute forward propagation and back propagation on 1/10 of the data to compute the derivative with respect to that 1/10 of the data.

Correct

Compute only forward propagation on 1/10 of the data. (The centralized machine then performs back propagation on all the data).

Compute back propagation on 1/10 of the data (after the centralized machine has computed forward propagation on all of the data).
