**Large Scale Machine Learning**

* Suppose you’re training a logistic regression classifier using stochastic GD. 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 halving (decreasing) the learning rate α, + see if that causes the cost to now consistently go down; and if not, keep halving it until it does.**
* **Try using a smaller learning rate α.**
* Which of the following statements about stochastic gradient descent are true? Check all that apply.
* **1 advantage of stochastic GD is it can start progress in improving parameters θ after looking at just a single training example; in contrast, batch GD needs to take a pass over the entire training set before it starts to make progress in improving the parameters' values.**
* **In each iteration of stochastic GD, the algorithm needs to examine/use only 1 training example**
* **If you have a huge training set, then GD descent may be much faster than batch GD**
* **Before running stochastic GD, you should randomly shuffle (reorder) the training set.**
* **In order to make sure stochastic GD is converging, we typically compute cost of the parameters ϴ w/ respect to a single training example, ({x(i), y(i}) *(NOT  Jtrain(θ)*) after each iteration (+ plot it) in order to make sure the cost function is generally decreasing.**
* Which of the following statements about online learning are true? Check all that apply.
* **When using online learning, you can discard training example you get, as you will not need to reuse past examples to re-train the model**
* **1 advantage of online learning is if the function we're modeling changes over time (such as if modeling the probability of users clicking on different URLs + user tastes/preferences are changing over time), the online learning algorithm will automatically adapt to these changes.**
* **Online learning algorithms are usually best suited to problems were we have a continuous/non-stop stream of data that we want to learn from.**
* **When using online learning, in each step we get a new example (x, y), perform 1 step of (essentially stochastic GD) learning on that example, + then discard it + move on to the next.**
* **Repeatedly get a single training example, take 1 step of GD descent using that example, and then move on to the next example.**
* Assuming that you have a very large training set, which of the following algorithms do you think can be parallelized using MapReduce + splitting the training set across different machines?
* An online learning setting, where you repeatedly get a single example (x, y), and want to learn from that single example before moving on.
* A neural network trained using batch gradient descent.
* Logistic regression trained using stochastic gradient descent.
* Linear regression trained using batch gradient descent.
* Which of the following statements about MapReduce are true?
* **Because of network latency + other overhead associated w/ MapReduce, if we run MapReduce using N CPUs, we might get less than an N-fold speedup compared to using 1 CPU.**
* **When using MapReduce with gradient descent, we usually use a single machine that accumulates the gradients from each of the MapReduce machines, in order to compute the parameter update for that iteration.**
* **If you have only 1 CPU with 1 computing core, then map-reduce is likely to help.**