The Coders' Club

Machine Learning: G1

Week 10: Assignment

Topic:
Large Scale Machine Learning

Some Additional Courses:

- Machine Learning Onramp (MathWorks)
  https://www.mathworks.com/learn/tutorials/machine-learning-onramp.html

- Deep Learning Onramp (MathWorks)
  https://www.mathworks.com/learn/tutorials/deep-learning-onramp.html

- AI From the Data Center to the Edge – An Optimized Path Using Intel® Architecture (Intel AI)
  https://software.intel.com/en-us/ai/courses/data-center-to-edge

- Machine Learning (Intel)
  https://software.intel.com/en-us/ai/courses/machine-learning

- Deep Learning (Intel)
  https://software.intel.com/en-us/ai/courses/deep-learning

Large Scale Machine Learning

Q.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?

- Use fewer examples from your training set.
- Try averaging the cost over a smaller number of examples (say 250 examples instead of 500) in the plot.
- 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.
- This is not possible with stochastic gradient descent, as it is guaranteed to converge to the optimal parameters θ.

Q.2. Which of the following statements about stochastic gradient descent are true? Check all that apply.

- One of the advantages of stochastic gradient descent is that it uses parallelization and thus runs 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 $J_{train}(\theta)$ after each iteration (and plot it) in order to make sure that the cost function is generally decreasing.
- If you have a huge training set, then stochastic gradient descent may be much faster than batch gradient descent.

Q.3. Which of the following statements about online learning are true? Check all that apply.

- One of the advantages of online learning is that if the function we're modelling changes over time (such as if we are modelling 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.
- When using online learning, you must save every new training example you get, as you will need to reuse past examples to re-train the model even after you get new training examples in the future.
- Online learning algorithms are usually best suited to the problems where we have a continuous/non-stop stream of data that we want to learn from.
- Online learning algorithms are most appropriate when we have a fixed training set of size m that we want to train on.

Q.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.

- An online learning setting, where you repeatedly get a single example (x, y) and want to learn from that single example before moving on.
- Logistic regression trained using stochastic gradient descent.
- A neural network trained using batch gradient descent.
- Linear regression trained using batch gradient descent.

Q.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.
- 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.
- If we run map-reduce using N computers, then we will always get at least an N-fold speedup compared to using 1 computer.