HSD5230 High Performance Computing
Homework 12

Miao Cai[*]

3/15/2019

1) How does optimization in machine learning differ from pure optimization?

Machine learning ususally acts indirectly: we intend to optimize the objective function P, which is defined based on the test set. However, we actually we optimize a different objective function $J(\theta)$ that is defined on the training set. In contrast, pure optimization has the goal of minimizing the objective function J alone.

2) How does 'on-line learning' differ from 'deterministic' (also called 'batch) approaches to machine learning? Can you describe this in a way a non-statistician would understand?

- **Batch gradient methods** use the entire training set to calculate the gradient at each step.
- **Stochastic (online) gradient methods** only use a single row at a time to estimate the gradient.

3) How do batch gradient descent, stochastic gradient descent, and mini-batch gradient descent differ?

- **Batch gradient descent**: it uses the entire training set to calculate the gradient at each step,
- **Stochastic (online) gradient descent**: it only uses a single example (row) at a time to estimate the gradient,
- **Mini-batch gradient method**: it uses more than one, but less than the full training dataset to estimate the gradient.

4) There is a new source of variation introduced when we implement stochastic gradient descent (compared with batch gradient descent). What is it that new source of variation? Does it decrease as the algorithm converges?

Sampling error. Yes, it decreases as the algorithm converges.

5) What does it mean for a function to be 'ill-conditioned'? How does this manifest in ML optimization (think of Jacobian and Hessian)?

The output value of the function changes a lot for a small change in the input value.

6) What is the advantage of using a learning rate schedule (dynamic learning rate) instead of a fixed learning rate? When do we need to use a dynamic learning rate instead of a fixed learning rate? Why?

[*]Department of Epidemiology and Biostatistics, College for Public Health and Social Justice, Saint Louis University. Email miao.cai@slu.edu

It enforces smaller steps as the algorithm proceeds, which helps the algorithm to reach the minimum. We need to use a dynamic learning rate if we want to reach the minumum. The reason is that if we keep a constant learning rate and a fixed step size, the algorithm will jump around the the minumum and never reach it.