Project: Robustness of Newton Methods and Running Time Analysis

Last updated: May 25, 2020

Outline

1. Robustness of Newton methods
2. Running time of two implementations of Gauss-Newton matrix-matrix products

Outline

Goal I

- From past projects we have known that stochastic gradient is sensitive to the learning rate
- We would like to check the situation of Newton methods

Settings I

- All settings are the same as before
- However, to avoid lengthy training time, let's consider a 5000-instance subset at this directory
- We still use the full test data

Parameters Considered (Newton Method)

- The Newton implementation is available in simpleNN
- You can use either Python or MATLAB implementations
- Check README for their use
- We check the following parameters
  - Percentage of data for subsampled Hessian: 5%, 10%
  - With/without

    Levenberg-Marquardt method

Parameters Considered (Stochastic Gradient) I

- Everything is the same as before:

  simple stochastic gradient + momentum
- We check the following initial learning rates

$$10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}$$

Checking the Convergence I

- Try to see the relation between training time and test accuracy
- You now have

  4 settings for Newton

  and

  4 settings for stochastic gradient
- You need to design maybe figures for the comparison
- The comparison can be, for example,

  training time versus test accuracy

Checking the Convergence II

- Visualization is always a concern. For example, you may not want to draw eight curves in the same figure

Outline

Gauss-Newton Matrix-vector Products I

- We talked about two ways
- One is solely by back propagation. The complexity is the lowest, but the memory consumption is huge
- The other is

$$\text{forward} + \text{backward}$$

- We want to analyze their running time by using the MATLAB code.
- An option `-Jacobian` decides which one is used.
- You want to git pull the latest code

Gauss-Newton Matrix-vector Products II

- For other parameters let's use the default ones (e.g., 5% subsampled Gauss Newton)
- For this part we should use full data. Otherwise each iteration takes too little time and the results may not be accurate
- No need to run many iterations.
- By profiling we may check if for the matrix-matrix products, the result is consistent with our complexity analysis

Gauss-Newton Matrix-vector Products III

- How about the situation of the total time? Any operations with lower complexity but take more time? Why did that happen?

Presentations and Reports I

- Presentations for projects 5 and 6

```
proj ID
5 ntust_f10802006
6 b05201015
5 b05201024
6 b05201037
5 t08303135
5 b06502060
5 r08521508
6 d08525008
6 b05701231
```

Presentations and Reports II

6 b06901143
5 t08902130
5 b06902124
6 b05902035
5 b05902050
5 b05902105
5 d08921024
6 a08922103
5 a08922119
6 a08922203
6 d08922029

Presentations and Reports III

5 d08922034
5 p08922005
6 r08922019
6 r08922082
5 r08922163
5 r07922100
6 r07922154
6 r08922a07
5 d04941016
6 r08942062
6 a08946101

Presentations and Reports IV

please do a 10-minute presentation (9-minute the contents and 1-minute Q&A)