Deep Learning Srihari

Numerical Computation

Sargur N. Srihari srihari@cedar.buffalo.edu

This is part of lecture slides on Deep Learning: http://www.cedar.buffalo.edu/~srihari/CSE676

Topics

- Numerical Computation
 - Overflow and Underflow
 - Poor Conditioning
- Gradient-based Optimization

Deep Learning Srihari

Overview

- ML algorithms usually require a high amount of numerical computation
 - To update estimate of solutions iteratively
 - not analytically derive formula providing expression
- Common operations:
 - Optimization
 - Determine maximum or minimum of a function
 - Solving system of linear equations
- Just evaluating a mathematical function of real numbers with finite memory can be difficult

Overflow and Underflow

- Problem caused by representing real numbers with finite bit patterns
 - For almost all real numbers we encounter approximations
- Although a rounding error it compounds across many operations and algorithm will fail
 - Numerical errors
 - Underflow: when nos close to zero are rounded to zero
 - $-\log 0$ is $-\infty$ (which becomes not-number for further operations)
 - Overflow: when nos with large magnitude are approximated as $-\infty$ or $+\infty$ (Again become not-no.)

Deep Learning Srihari

Function needing stabilization for Over/Underflow

Softmax probabilities in multinoulli

$$\operatorname{softmax}(x)_{i} = \frac{\exp(x_{i})}{\sum_{j=1}^{n} \exp(x_{j})}$$

- Consider when all x_i are equal to some c. Then all probabilities must equal 1/n. This may not happen
 - When c is a large negative; denominator = 0, result undefined underflow
 - When c is large positive, $\exp(c)$ will overflow
- Circumvented using softmax(z) where $z=x-\max_i x_i$
- Another problem: underflow in numerator can cause log softmax (x) to be -∞
 - Same trick can be used as for softmax

Dealing with numerical consderations

- Developers of low-level libraries should take this into consideration
- ML libraries should be able to provide such stabilization
 - Theano for Deep Learning detects and provides this

Poor Conditioning

- Conditioning refers to how rapidly a function changes with a small change in input
- Rounding errors can rapidly change the output
- Consider $f(x) = A^{-1}x$
 - $-A \varepsilon R^{n \times n}$ has a eigendecomposition
 - Its condition no. is $\max_{i,j} \left| \frac{\lambda_i}{\lambda_j} \right|$, i.e. ratio of largest to smallest eigenvalue
 - When this large, the output is very sensitive to input error
 - Poorly conditioned matrices amplify pre-existing errors when we multiply by its inverse