Optimization for Training Deep Models

Sargur N. Srihari
srihari@cedar.buffalo.edu

Topics in Optimization

- Overview of Optimization in Deep Learning
- How learning differs from optimization
  - Risk, empirical risk and surrogate loss
  - Batch, minibatch, data shuffling
- Challenges in neural network optimization
- Basic Algorithms
- Parameter initialization strategies
- Algorithms with adaptive learning rates
- Approximate second-order methods
- Optimization strategies and meta-algorithms

Optimization in Deep Learning

- Optimization is encountered often in ML
  1. Inference with PCA requires optimization
     - Encoding: $f(\boldsymbol{x})=\boldsymbol{c}$, Decoding: $\boldsymbol{x} \approx g(f(\boldsymbol{x}))$, $g(\boldsymbol{c})=D\boldsymbol{c}$
     - Optimal $\boldsymbol{c}^*=\mathrm{argmin}_{\boldsymbol{c}}||\boldsymbol{x}\text{-}g(\boldsymbol{c})||_2$, Reconstruction: $g(f(\boldsymbol{x})=DD^{\mathrm{T}}\boldsymbol{x}$
  2. Optimization to write proofs or design algorithms
     - In linear regression: sum-of-squared errors objective is same as obtained using maximum likelihood with Gaussian noise
  3. Neural network training
     - Most difficult optimization is neural network training
     - Weight decay minimization:

$$\boxed{J(\boldsymbol{w}) = \lambda \left\|\boldsymbol{w}\right\|_2^2 - E_{x,y\sim \hat{p}_{data}} \log p_{\mathrm{model}}(y \mid \boldsymbol{x})}$$

Neural network optimization is difficult

- Commonly months of time on 100s of machines to solve a single instance of neural network training
- So specialized optimization techniques developed

Our focus on particular case of optimization

- Find parameters $\boldsymbol{\theta}$ of a neural network that significantly reduce a cost function $J(\boldsymbol{\theta})$
  - Which typically includes:
    - a performance measure evaluated on training set, e.g.,
      - $\boxed{J(\boldsymbol{w}) = E_{x,y\sim\hat{p}_{data}} \log p_{\text{model}}(y \mid \boldsymbol{x})}$ where $p_{\text{model}}(y|\boldsymbol{x})$ is a likelihood function
      - For linear regression $\boxed{p_{\text{model}}(y \mid \boldsymbol{x}) = N(y; \boldsymbol{x}^T \boldsymbol{w} + \boldsymbol{b}, 1)}$ and $J(\boldsymbol{w})$ is same as sum-of-squared errors
    - additional regularization terms, e.g., $\boxed{\lambda \|\boldsymbol{w}\|_2^2}$

Plan of Discussion of Optimization

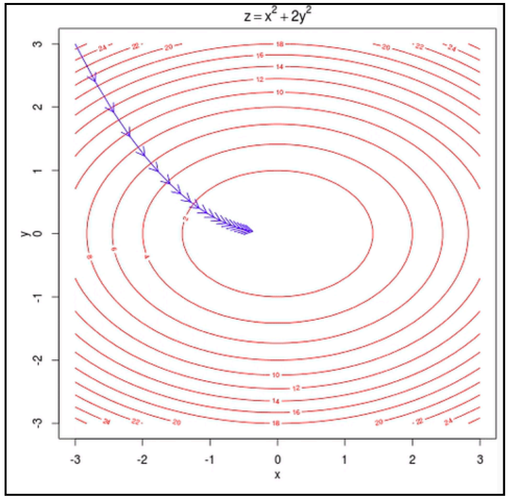
1. How training optimization differs from pure optimization
2. Challenges that make optimization of neural networks difficult
3. Several practical algorithms including
   1. Optimization algorithms
   2. Strategies for initializing parameters
      - Most advanced algorithms
        - adapt learning rates or
        - leverage *second derivatives* of cost function
4. Combine simple optimization algorithms into higher-level procedures

Summary of Optimization Methdos

- Movies:

http://hduongtrong.github.io/2015/11/23/coordinate-descent/

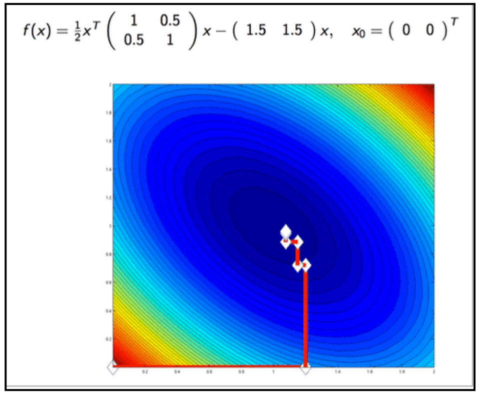
Gradient Descent



$$\boldsymbol{g} = \frac{1}{M}\nabla_\theta \sum_{i=1}^{M} L\left(\boldsymbol{x}^{(i)}, y^{(i)}, \boldsymbol{\theta}\right)$$

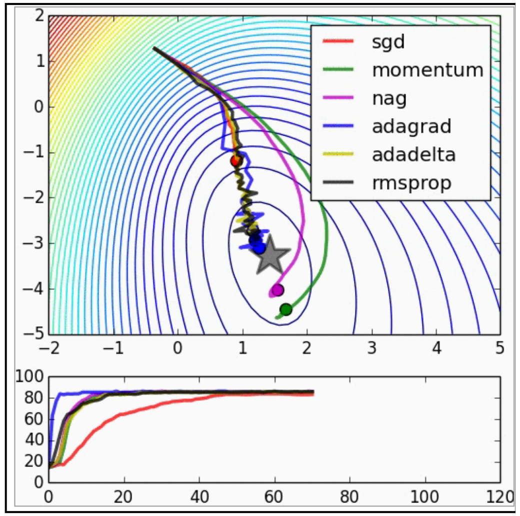
$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \varepsilon \boldsymbol{g}$$

Coordinate Descent



Minimize $f(\boldsymbol{x})$ wrt a single variable, x_i, then wrt x_j etc

SGD



$$\boldsymbol{g} = \frac{1}{m'}\nabla_\theta \sum_{i=1}^{m'} L\left(\boldsymbol{x}^{(i)}, y^{(i)}, \boldsymbol{\theta}\right)$$

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \varepsilon \boldsymbol{g}$$