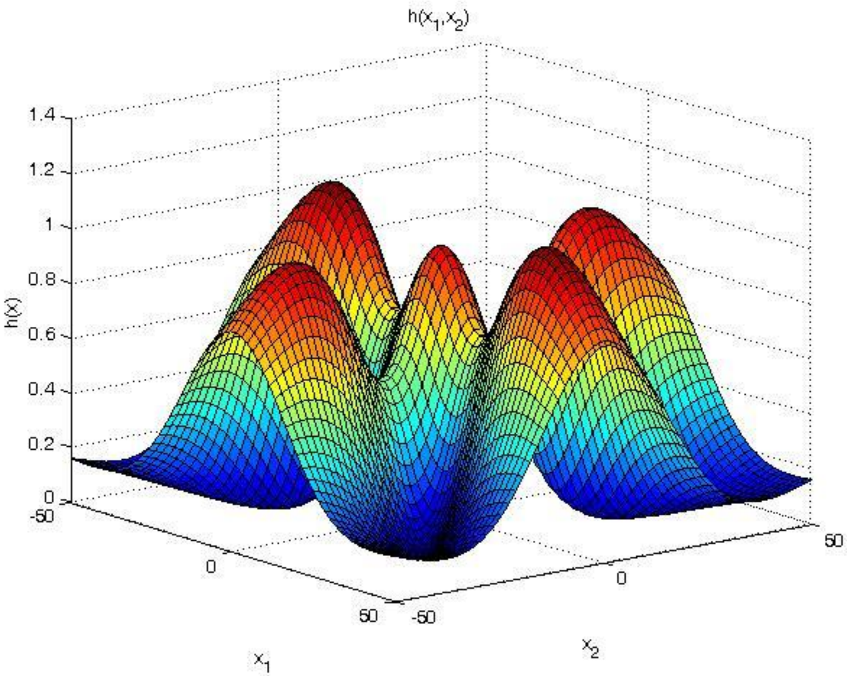
CS 395T
Numerical Optimization for Graphics and AI



Qixing Huang

August 31th 2017

Course Setup

- Welcome to the course on Numerical Optimization, with a focus on its applications to AI and Graphics

- Basic administrative details:
  - Instructor: Qixing Huang
  - Teaching assistant: Xiangru Huang
  - Course website: http://www.cs.utexas.edu/~huangqx/CS395T_Numerical_Optimization.html

  - We will use canvas for announcements

Course Setup

- Prerequisites: no formal ones, but we assume some knowledge of
  - Linear algebra, Probability, Calculus, Geometry
  - Programming (Matlab, Python…)
  - Core problems in AI or ML or Graphics
    - The course material is application driven
  - Formal mathematical thinking
    - If you just like tuning neural networks, this class will be hard for you

Course Setup

- Evaluation
  – 7 homeworks (70%)
  – 1 final project (30%)

Homework: A mixture of theory and coding

Project: solving a real problem in AI/ML/Graphics with modern optimization techniques. Groups of 2 or 3.

Most important: work hard and have fun!

What is mathematical optimization?

- Optimization models the goal of solving a problem in the "optimal way"

- Examples
  - Running a business: to maximize profit, minimize risk
  - Design: maximize the strength, within the design constraints
  - Planning: select a route from Austin and Yellowstone to minimize the fuel consumption

- Formal definition: to minimize (or maximize) a real function by deciding the values of free variables from within a feasible set.

Optimization problems are ubiquitous

We will see time and time again:

Translate  into $P: \quad \min_{x \in D} \; f(x)$

Real problem | Optimization problem

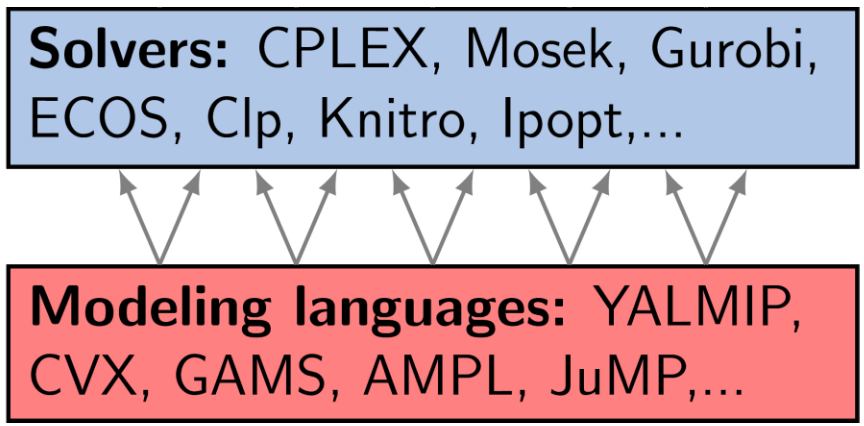
Examples in Vision/Robotics/NLP /ML/Graphics/

Example of the contrary?

This course: how to formulate P, how to solve P, and what are the guarantees

Why bother how to solve P and what are the guarantees

- There are plenty of optimization softwares



- Almost all algorithms are data-dependent and can perform better or worse on different problems and data sets
- In many cases, studying P leads to new algorithms, and this is where research papers come from

Categories of optimization models

- Linear vs. Nonlinear

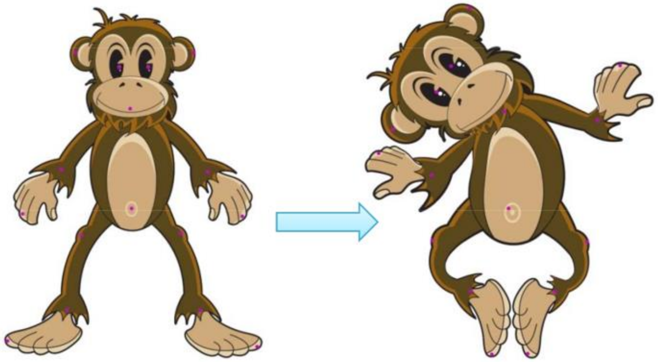
- Convex vs. Nonconvex

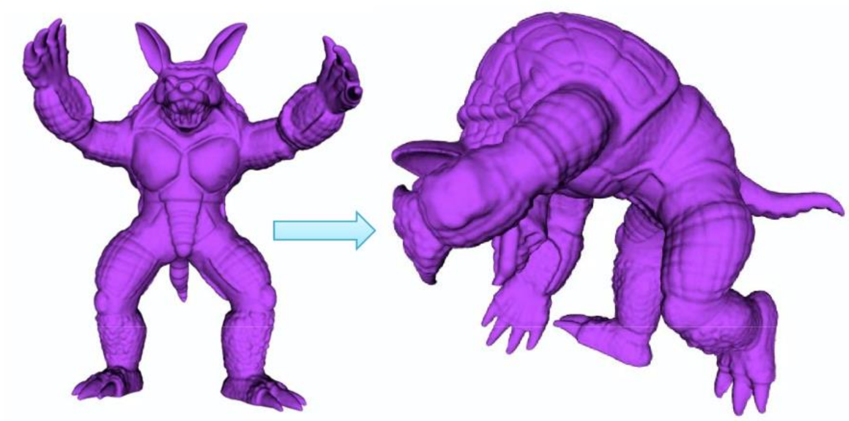
- Continuous vs. Discrete

- Deterministic vs. Stochastic

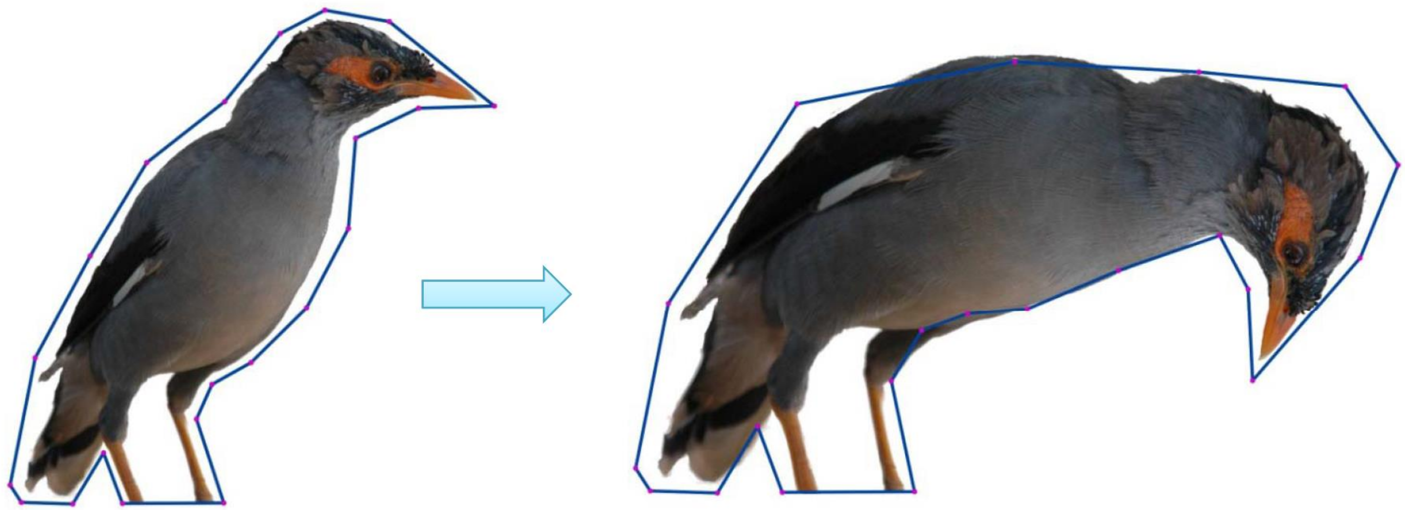
Some Examples

Shape Deformation



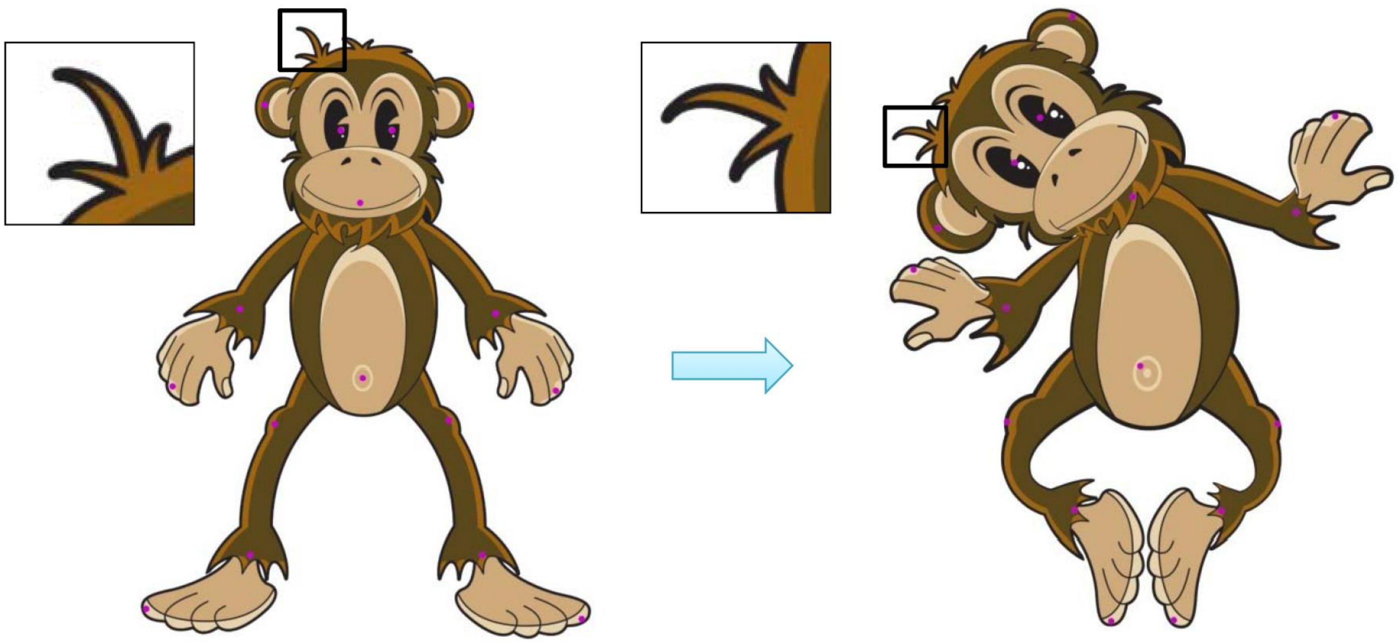


Shape Deformation



Goals

User constraints + global change + local detail preservation



Typical formulation

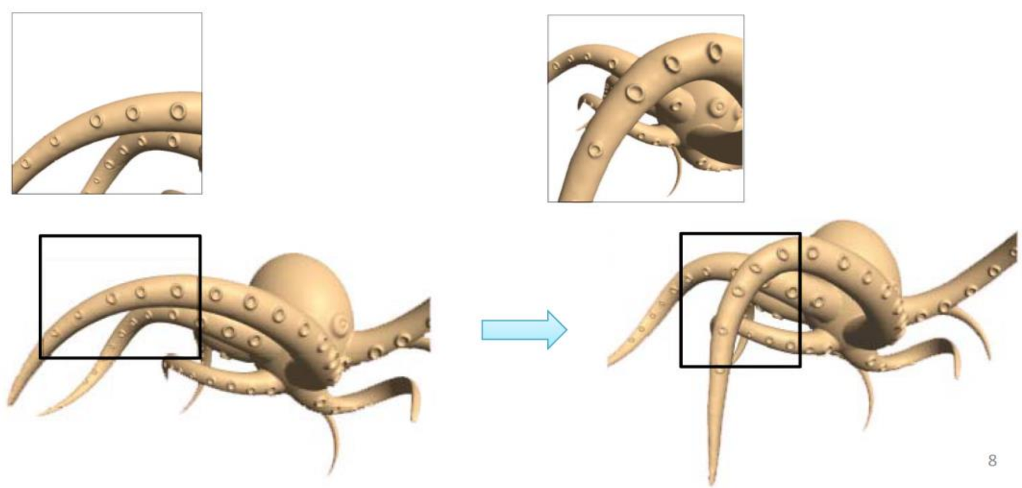
- Solve for local transformation and deformed surface simultaneously

$$V' = \underset{V'}{\operatorname{argmin}} \left(\sum_{i=1}^{n} \| L(\boldsymbol{v}'_i) - T_i(\delta_i) \|^2 + \sum_{j \in \mathcal{C}} \| \boldsymbol{v}'_j - \boldsymbol{u}_j \|^2 \right)$$

Laplacian coordinates of the deformed mesh

Laplacian coordinates of the original mesh

User constraints



[Sorkine et al. 04]

Typical formulation

- Solve for local transformation and deformed surface simultaneously

$$V' = \underset{V'}{\operatorname{argmin}} \left(\sum_{i=1}^{n} \|L(\boldsymbol{v}_i') - T_i(\delta_i)\|^2 + \sum_{j \in \mathcal{C}} \|\boldsymbol{v}_j' - \boldsymbol{u}_j\|^2 \right)$$

Laplacian coordinates of the deformed mesh

Laplacian coordinates of the original mesh

User constraints

Non-convex optimization, and people have tried:

Newton method

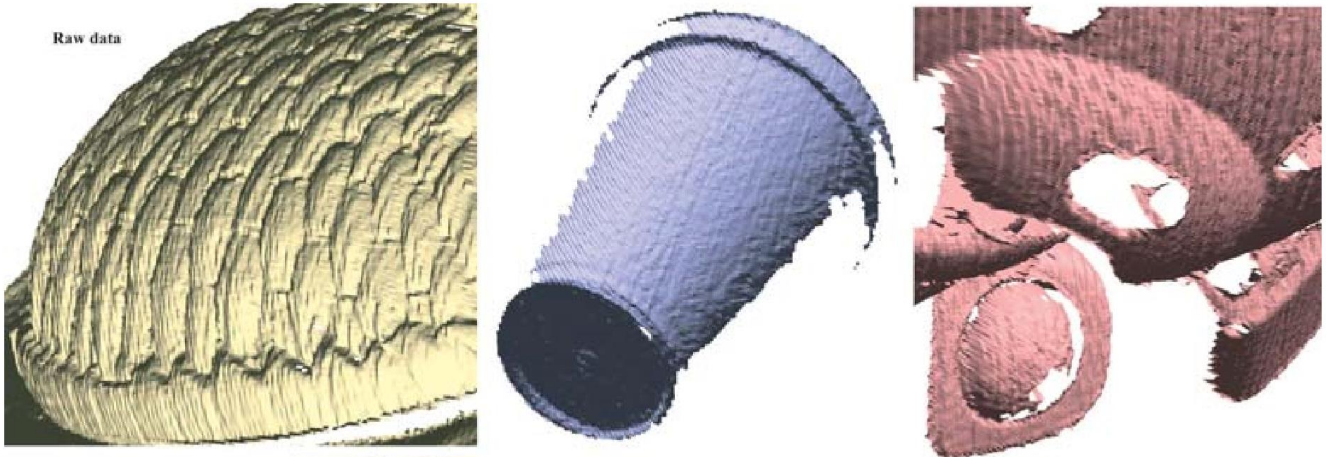
Alternating minimization

Gradient-based method

Open question: when can we find the global optimal solution?

Scan smoothing

[Diebel et al. 06]



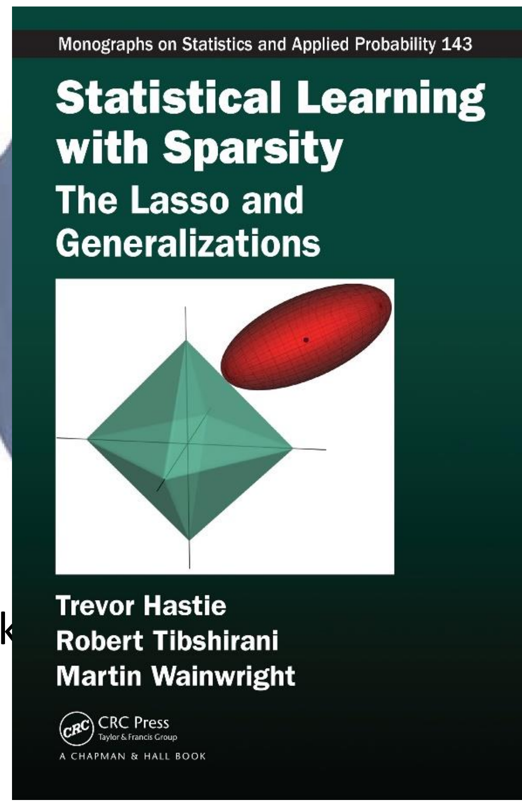
Scan smoothing

[Diebel et al. 06]

$$\hat{x} = \operatorname*{argmin}_x \left[\sum_i (z_i - x_i)^T \, \Omega_i \, (z_i - x_i) \right] + \left[\sum_{k,j} f_{\text{sqrt}}(n_k - n_j) \right]$$

Data term

Prior term



How this work ... atistics (Lasso)

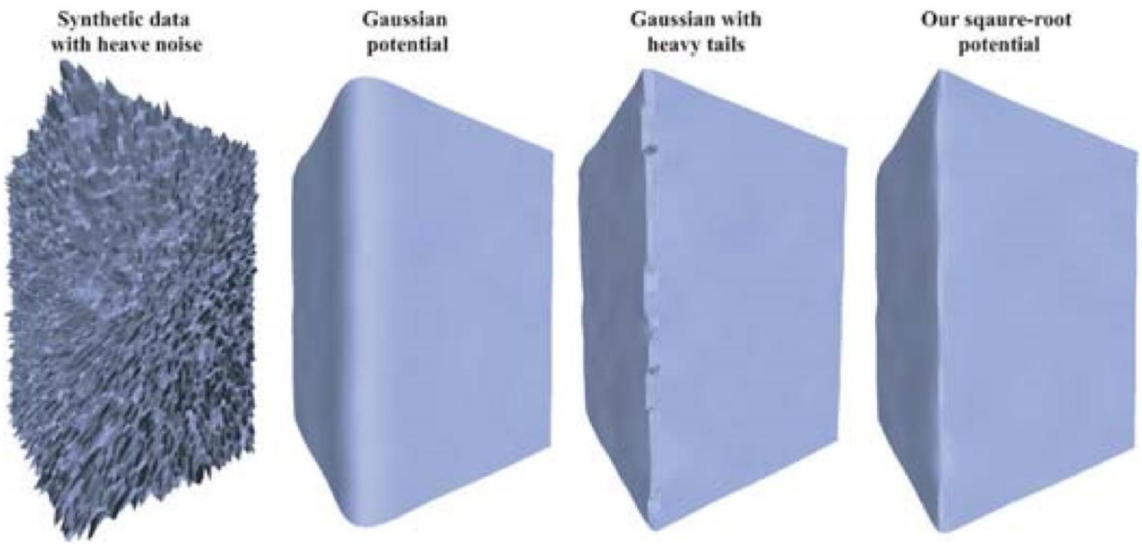
Scan smoothing

[Diebel et al. 06]

$$\hat{x} = \operatorname*{argmin}_x \left[\sum_i (z_i - x_i)^T \, \Omega_i \, (z_i - x_i) \right] + \left[\sum_{k,j} f_{\text{sqrt}}(n_k - n_j) \right]$$

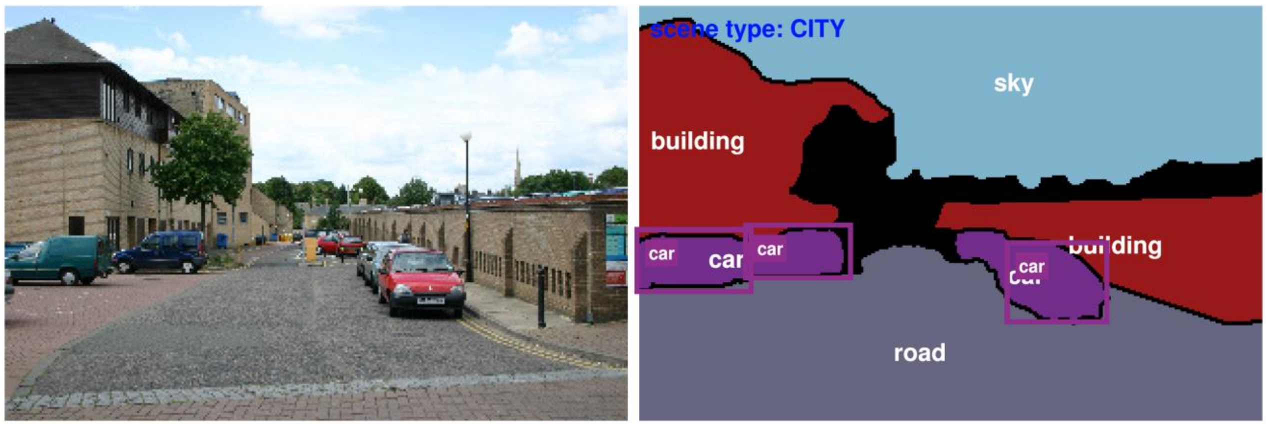
Data term

Prior term

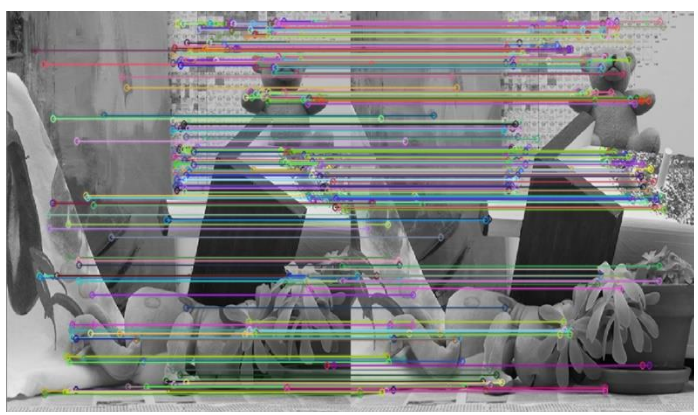


We will also study optimization algorithms:
ADMM, Coordinate descent….

Computer Vision Applications



Semantic Segmentation



Stereo Matching

MRF Inference

$$\arg\min_{w_{1\ldots N}} \sum_{n=1}^{N} U_n(w_n) + \sum_{(m,n)\in\mathcal{C}} P_{mn}(w_m, w_n),$$

Unary terms
(compatability of data with label w)

Pairwise terms
(compatability of neighboring labels)

The literature reflects almost all advances in optimization during the past decade:

Graphcut

Linear programming relaxation

Quadratic programming relaxation

Semidefintie programming relaxation

Coordinate descent

Dual coordinate descent

Stochastic gradient descent

Block-coordinate descent

Deep neural networks versus Deep residual networks

Deep Neural Networks

- Linear model for simplicity

$$\min_{W_i, 1 \le i \le l} E_{(\boldsymbol{x},\boldsymbol{y})\sim p} \, \|\boldsymbol{y} - (\prod_{i=1}^{l} W_i)\boldsymbol{x}\|^2$$

Deep neural network training

[He et al. 16]

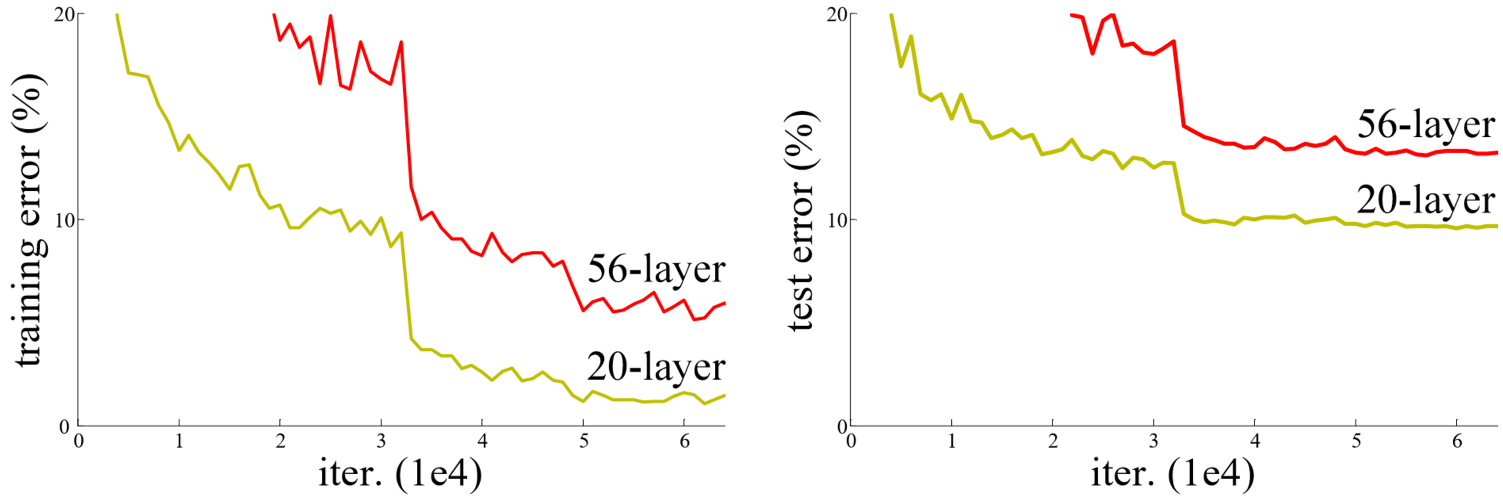


Figure 1. Training error (left) and test error (right) on CIFAR-10 with 20-layer and 56-layer "plain" networks. The deeper network has higher training error, and thus test error. Similar phenomena on ImageNet is presented in Fig. 4.

Deep Residual Networks

- Linear model for simplicity

$$\min_{W_i, 1 \le i \le l} E_{(\boldsymbol{x},\boldsymbol{y})\sim p} \;\|\boldsymbol{y} - \Big(\prod_{i=1}^{l}(I + W_i)\Big)\boldsymbol{x}\|^2 + \lambda \sum_{i=1}^{l} \|W_i\|_{\mathcal{F}}^2$$

Deep Residual Networks

[He et al. 16]

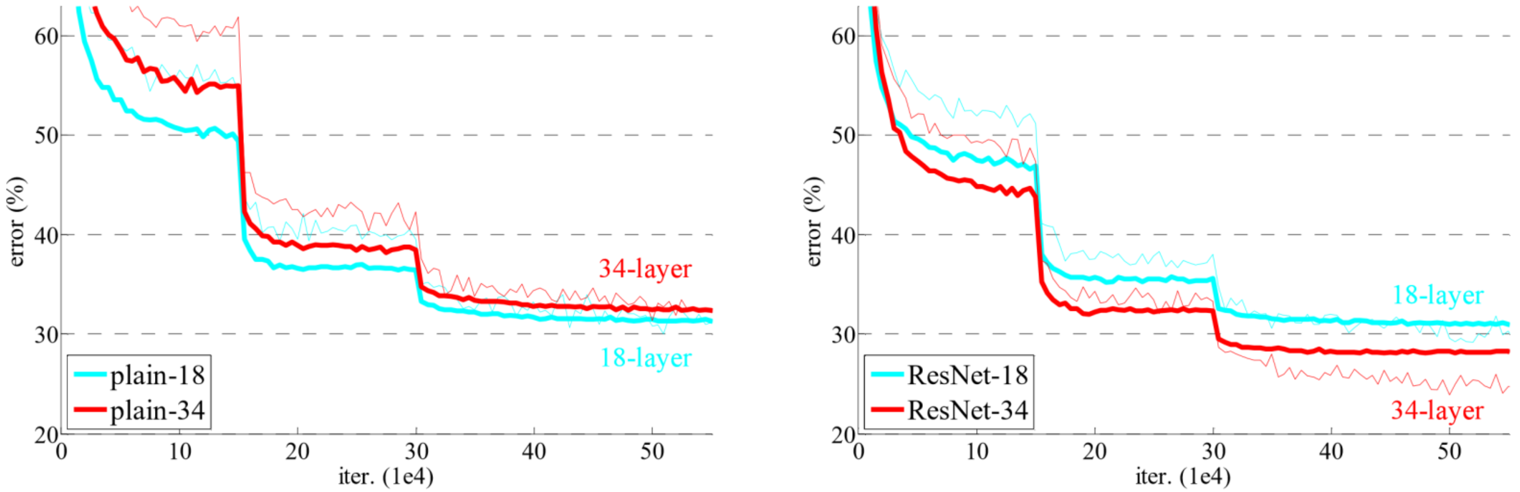


Figure 4. Training on **ImageNet**. Thin curves denote training error, and bold curves denote validation error of the center crops. Left: plain networks of 18 and 34 layers. Right: ResNets of 18 and 34 layers. In this plot, the residual networks have no extra parameter compared to their plain counterparts.

Other topics to be covered

- Iterative closest point method for geometry registration
- Policy gradient descent
- Simultaneous localization and mapping

- Compressive sensing
- Low-rank matrix recovery
- Phase retrieval

Trends in optimization

Non-linear optimization

Convex optimization

Non-convex optimization

Trends in optimization

Second-order methods

First-order methods

Distributed optimization