ECE 8101: Nonconvex Optimization for Machine Learning

Lecture Note 1: Course Info & Introduction

Jia (Kevin) Liu

Associate Professor
Department of Electrical and Computer Engineering
The Ohio State University, Columbus, OH, USA

Autumn 2024

Course Info (1)

- Instructor: Jia (Kevin) Liu, Associate Professor
- Office: 620 Dreese Labs
- Email: `liu@ece.osu.edu`
- Time: TTh 11:10AM – 12:30PM
- Location: Baker Systems 140
- Office Hour: Wed 5–6pm or by appointment
- Office Hour Zoom Link:
  `https://osu.zoom.us/j/98827464068?`
  `pwd=p2pWLylQGR3v1OMjmJXn4EpIdoVzMW.1`
- Websites: Carmen: announcements, grade management, course materials)
  Schedule: `https://kevinliu-osu.github.io/teaching/ECE8101_A24/`
- Prerequisite:
  - Working knowledge of Linear Algebra and Probability
  - Exposure to optimization and machine learning is a plus but not required

Course Info (2)

Grading Policy:

- Class Participation (10%): Top Hat (please install on your phone/tablet)
- Paper Reading Assignment (60%)
  - Assigned after each major topic set (approximately)
  - May involve open-ended questions
  - Must be typeset using LaTeX in ICML format
- Final Project (30%)
  - Finished by a team of 2, but solo project is OK. ~~Project proposal due soon after spring break~~
  - Project report due in the final exam week. Follow ICML format (Could become a publication of yours! "Automatic A" if determined publishable by instructor ☺)
  - 20 (?)-minute in-class presentation at the end of the semester. Final report due by the *end* of final exam week (Dec. 11)
  - Potential ideas of project topics (should contain something new & useful):
    - Nontrivial extension of the results introduced in class
    - Novel applications in your own research area
    - New theoretical analysis/insights of an existing/new algorithm
    - It is important that you justify its novelty!

Course Info (3)

Course Materials:

- No required textbook
- Lecture notes are developed based on:
  - Important & trending papers in the field
  - [BV] S. Boyd and L. Vandenberghe, "*Convex Optimization*," Cambridge University Press, 2004 (available online)
  - [NW] J. Nocedal and S. Wright, "*Numerical Optimization*," Ed. 2, Springer,
  - [BSS] M. Bazarra, H.D. Sherali, and C.M. Shetty, "*Nonlinear Programming: Theory and Algorithms*," John Wiley & Sons, 2006
  - [Nesterov] Y. Nesterov, "*Introductory Lectures on Convex Optimization: A Basic Course*," Springer, 2004

Tentative Topics

- Stochastic First-Order Nonconvex Optimization
  - Fundamental of SGD; variance-reduced algorithms (SVRG, SAGA, SPIDER); accelerated algorithms (STORM, Hybrid)
- Federated and Decentralized Optimization
  - Decentralized (stochastic) gradient descent, FedAvg, and variants
- Complex-Structured Optimization
  - Minimax optimization, bilevel optimization, multi-objective optimization ...
- Zeroth-order Optimization
  - One-point and two-point gradient estimator; zeroth-order SGD; zeroth-order variance-reduced optimization methods ...
- Geometry of Nonconvex Optimization
  - Landscape of learning models, PL conditions, NTK ...

Special Notes

- Advanced, research-oriented
  - There will be paper reading assignments and a term project
- Goal: Prepare & train students for theoretical ML research
- But will (briefly) mention relevant applications in ML:
  - Deep Learning
  - Big data analytics
  - LLM, GenAI
  - ...
- Caveat: Focus on theory & proofs, rather than "coding/programming"
  - No "one book fits all" $\Rightarrow$ Many readings required
  - Will try to cover a wide range of major topics
  - Background materials will be introduced but at very fast pace
  - So, mathematical maturity is essential!

How to Best Prepare for the Lectures?

Read, read, read!

- Especially if you're unfamiliar with the background (e.g., linear algebra, probability, ...)
  - Will quickly go over some related background in class
- Appendices in [BV] and [BSS] provide some math background
- You are welcome to ask questions in office hours
- But careful self-studies may still be needed

Mathematical Optimization

Mathematical optimization problem:

w.l.o.g.

$$\begin{array}{ll} \text{Minimize} & f_0(\mathbf{x}) \\ \text{subject to} & f_i(\mathbf{x}) \leq 0, \quad i = 1, \ldots, m \end{array}$$

- $\mathbf{x} = [x_1, \ldots, x_N]^\top \in \mathbb{R}^N$: decision variables
- $f_0 : \mathbb{R}^N \to \mathbb{R}$: objective function
- $f_i : \mathbb{R}^N \to \mathbb{R}, i = 1, \ldots, m$: constraint fucntions

Solution or **optimal point** $\mathbf{x}^*$ has the smallest value of f_0 among all vectors that satisfy the constraints

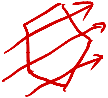
Brief History of Optimization

Theory:

- Early foundations laid by many all-time great mathematicians (e.g., Newton, Gauss, Lagrange, Euler, Fermat, ...)
- Convex analysis 1900–1970 (Duality by von Neumann, KKT conditions...)

Algorithms

- 1947: simplex algorithm for linear programming (Dantzig)
- 1970s: ellipsoid method [Khachiyan 1979], 1st polynomial-time alg. for LP
- 1980s & 90s: polynomial-time interior-point methods for convex optimization [Karmarkar 1984, Nesterov & Nemirovski 1994]
- since 2000s: many methods for large-scale convex optimization

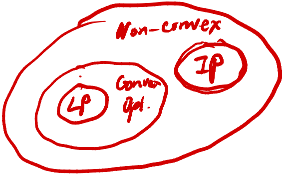


Applications

- before 1990: mostly in operations research, a few in engineering
- since 1990: many applications in engineering (control, signal processing, networking and communications, circuit design,...)
- since 2000s: machine learning

Solving Optimization Problems

- General optimization problems
  - Very difficult to solve (NP-hard in general)
  - Often involve trade-offs: long computation time, may not find an optimal solution (approximation may be acceptable in practice)
- Exceptions: Problems with special structures
  - Linear programming problems
  - Convex optimization problems
  - Some non-convex optimization problems with strong-duality



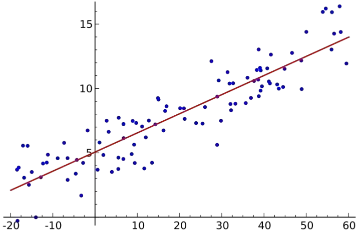
- Watershed between Problem Hardness: Convexity
  - This course focuses on nonconvex problems arising from ML context

Applying Optimization Tools in Machine Learning

- Linear Regression
- Variable Selection & Compressed Sensing
- Support Vector Machine
- Logistic Regression (+ Regularization)
- Matrix Completion
- Deep Neural Network Training
- Reinforcement Learning
- Distributed/Federated/Decentralized Learning
- LLM Pretraining and Finetuning
- ...

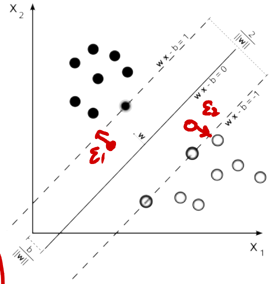
Example 1: Linear Regression (Convex)

$$\text{Minimize}_{\beta} \quad \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2$$



- Given data samples: $\{(\mathbf{x}_i, y_i), i = 1, \ldots, m\}$, where $\mathbf{x}_i \in \mathbb{R}^n$, $\forall i$
- Find a linear estimator: $y = \boldsymbol{\beta}^\top \mathbf{x}$, so that "error" is small in some sense
- Let $\mathbf{X} \triangleq [\mathbf{x}_1, \ldots, \mathbf{x}_m]^\top \in \mathbb{R}^{m \times n}$, $\mathbf{y} \triangleq [y_1, \ldots, y_m]^\top \in \mathbb{R}^m$
- Linear algebra for $\|\cdot\|_2$: $\boldsymbol{\beta}^* = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}$ (analytical solution)
- Computation time proportional to $n^2 m$ (less if structured)
- Stochastic gradient if m, n are large

Example 2: Support Vector Machine (Convex)



- Given data samples: $\{(\mathbf{x}_i, y_i), i = 1, \dots, m\}$
  - $\mathbf{x}_i \in \mathbb{R}^n$ called "feature vectors", $\forall i$
  - $y_i \in \{-1, +1\}$ are "labels"
- Linear classifier: $f(\mathbf{x}) = \mathrm{sgn}(\mathbf{w}^\top \mathbf{x} + b)$:
  - $\mathbf{w} \in \mathbb{R}^n$: weight vector for features
  - $b \in \mathbb{R}$: Some "bias"

If $y_i = 1$, $\underline{w}^T \underline{x} + b \geq 1$

$y_i = -1$ $\underline{w}^T \underline{x} + b \leq -1$

- Goal: To find a pair $(\mathbf{w}, b)$ to minimize a weighted sum such that
  - Minimize classification error on training samples
  - Robust to random noise in the training samples

$\to y_i(\underline{w}^T \underline{x} + b) \geq 1$

$$\underset{\mathbf{w}, b, \boldsymbol{\epsilon}}{\text{Minimize}} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C \sum_{i=1}^{m} \epsilon_i$$

$$\text{subject to} \quad y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \epsilon_i, \quad \epsilon_i \geq 0, \quad i = 1, \dots, m$$

Optimization Algorithms for SVM

- Coordinate Descent [Platt, 1999; Chang and Lin, 2011]
- Stochastic gradient [Bottou and LeCun, 2004; Shalev-Shwartz et al., 2007]
- Higher-order methods (interior-point) [Ferris and Munson, 2002; Fine and Scheinberg, 2001]; (on reduced space) [Joachims, 1999]
- Shrink Algorithms [Duchi and Singer, 2009; Xiao, 2010]
- Stochastic gradient + shrink + higher-order [Lee and Wright, 2012]

Nonconvex Optimization Problems in ML

- Lower complexity bound for solving general nonconvex problems
  - Consider, w.l.o.g., $\min_{\mathbf{x}\in[0,1]^d} f(\mathbf{x})$
  - f is nonconvex and L-Lipschitz-continuous, with global optimal $f^* > -\infty$
  - To find an ϵ-approximate solution $\hat{\mathbf{x}}$ (i.e., $f(\hat{\mathbf{x}}) - f^* \leq \epsilon$), number of iterations required: $\Omega(L^d\epsilon^{-d})$ (exponential)

Nonconvex Optimization Problems in ML

- Lower complexity bound for solving general nonconvex problems
  - Consider, w.l.o.g., $\min_{\mathbf{x}\in[0,1]^d} f(\mathbf{x})$
  - f is nonconvex and L-Lipschitz-continuous, with global optimal $f^* > -\infty$
  - To find an ϵ-approximate solution $\hat{\mathbf{x}}$ (i.e., $f(\hat{\mathbf{x}}) - f^* \leq \epsilon$), number of iterations required: $\Omega(L^d\epsilon^{-d})$ (exponential)
- Several ways to relax this challenging goal: Geomettic Prog.
  - Finding hidden convexity or reformulate into an equivalent convex problem $x_1^{d_1}\cdot x_2^{d_2}\cdots x_n^{d_n}$
    - Need to exploit special problem structure as much as possible
    - However, solution approaches cannot be generalized
  - Change the goal to finding a stationary point or a local extremum
    - Often possible to obtain FO methods with polynomial dependence of the complexity on the dimension of the problem and desired accuracy
  - Identify a class of problems:
    - General enough to characterize a wide range of applications (in ML)
    - Allow one to obtain global performance guarantees of an algorithm
    - E.g., Polyak-Lojasiewicz condition (linear convergence), α-weakly-quasi-convexity (sublinear convergence), etc.

Nonconvex Optimization Problems in ML

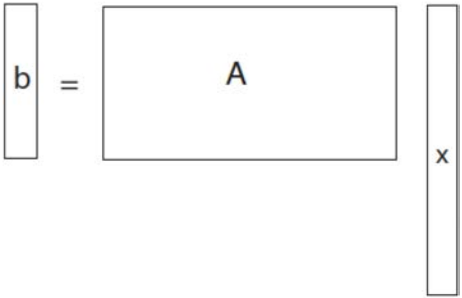
- Lower complexity bound for solving general nonconvex problems
  - Consider, w.l.o.g., $\min_{\mathbf{x}\in[0,1]^d} f(\mathbf{x})$
  - f is nonconvex and L-Lipschitz-continuous, with global optimal $f^* > -\infty$
  - To find an ϵ-approximate solution $\hat{\mathbf{x}}$ (i.e., $f(\hat{\mathbf{x}}) - f^* \le \epsilon$), number of iterations required: $\Omega(L^d \epsilon^{-d})$ (exponential)
- Several ways to relax this challenging goal:
  - Finding hidden convexity or reformulate into an equivalent convex problem
    - Need to exploit special problem structure as much as possible
    - However, solution approaches cannot be generalized
  - Change the goal to finding a stationary point or a local extremum
    - Often possible to obtain FO methods with polynomial dependence of the complexity on the dimension of the problem and desired accuracy
  - Identify a class of problems:
    - General enough to characterize a wide range of applications (in ML)
    - Allow one to obtain global performance guarantees of an algorithm
    - E.g., Polyak-Lojasiewicz condition (linear convergence), α-weakly-quasi-convexity (sublinear convergence), etc.
  - But what if gradients are hard to obtain?
    - E.g., reinforcement learning, blackbox adversarial attacks on DNN?
    - Zeroth-order or derivative-free methods

Tractable Nonconvex Optimization Problems in ML

- Problems with hidden convexity or analytic solutions
  - Eigen-problems (e.g., PCA, multi-dimensional scaling, ...)
  - Non-convex proximal operators (e.g., Hard-thresholding, Potts minimization)
  - Some discrete problems (binary graph segmentation, discrete Potts minimization, nearly optimal K-means)
  - Infinite-dimensional problems (smoothing splines, locally adaptive regression splines, reproducing kernel Hilbert spaces)
  - Non-negative matrix factorization (NMF)
  - Compressive sensing with ℓ_1 regularization
- Problems with (global) convergence results
  - Phase retrieval problem
  - Low-rank matrix completion
  - Deep learning
- Problems with certain properties of symmetry
  - Rotational symmetry, discrete symmetry, etc.

Example 3: Compressive Sensing (Nonconvex)

Interested in solving undetermined systems of linear equations:



- Estimate $\mathbf{x} \in \mathbb{R}^n$ from linear measurements $\mathbf{b} = \mathbf{A}\mathbf{x} \in \mathbb{R}^m$, where $m \ll n$.
- Seems to be hopelessly ill-posed, since more unknowns than equations...
- Or does it?

A Little History of Compressive Sensing (CS)

- Name coined by David Donoho
- Pioneered by Donoho and Candès, Tao and Romberg in 2004

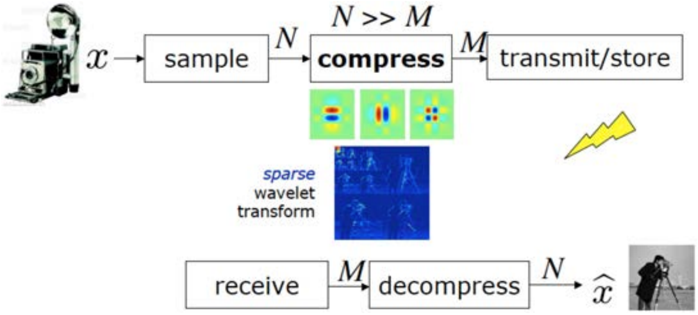
   

Compressed sensing
DL Donoho - Information Theory, IEEE Transactions on, 2006 - ieeexplore.ieee.org
Abstract—Suppose is an unknown vector in(a digital image or signal); we plan to measure general linear functionals of and then reconstruct. If is known to be compressible by transform coding with a known transform, and we reconstruct via the nonlinear procedure ...
Cited by 9878 Related articles All 31 versions Cite Save More

Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information
EJ Candès, J Romberg, T Tao - Information Theory, IEEE ..., 2006 - ieeexplore.ieee.org
Abstract—This paper considers the model problem of recon-structing an object from incomplete frequency samples. Consider a discrete-time signal and a randomly chosen set of frequencies. Is it possible to reconstruct from the partial knowledge of its Fourier ...
Cited by 6692 Related articles All 38 versions Cite Save

Sensing and Signal Recovery

Conventional paradigm of data acquisition: Acquire then compress



Q: Why compression works?

A: Quite often, there's only marginal loss in "quality" between the raw data and its compression form.

Q: But still, why marginal loss?

Sparse Representation

- **Sparsity:** Many real world data admit sparse representation. The signal $\mathbf{s} \in \mathbb{C}^n$ is sparse in a basis $\mathbf{\Phi} \in \mathbb{C}^{n \times n}$ if

$$\mathbf{s} = \mathbf{\Phi}\mathbf{x} \quad \text{and} \quad \mathbf{x} \in \mathbb{R}^n \text{ only has very few non-zero elements}$$

- For example, images are sparse in the wavelet domain

- The $\#$ of large coefficients in the wavelet domain is small $\Rightarrow$ compression

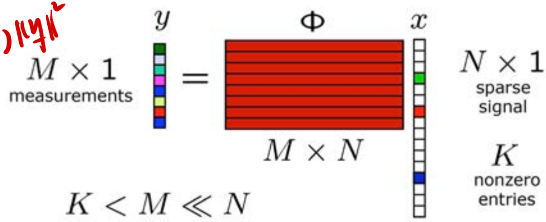
Compressed Sensing: Compression on the Fly!

Q: Could we directly compress data and then reconstruct?

$$y_i = \langle \mathbf{a}_i, \mathbf{x} \rangle, \quad i = 1, \ldots, m$$

RIP : $\underline{A}$, $\exists\, \delta > 0$

$(1-\delta) \cdot \|y\|^2 \leq \|\underline{A}y\|^2 \leq (1+\delta)\|y\|^2$



- Goal: To learn (recover) $\mathbf{x}$'s value through some given (noisy) samples y_i?
- Mathematically, this gives rise to an underdetermined system of equations, where the signal of interests is *sparse*

Sparse Recovery

In optimization, CS can be written in the form of:

$$\underset{\mathbf{x}\in\mathbb{R}^n}{\text{Minimize}}\ \phi_\gamma(\underbrace{\mathbf{x}}_{\text{estimate}}) \triangleq \underbrace{f(\mathbf{y},\overset{\text{iid } N(0,1)}{\mathbf{\Phi}};\mathbf{x})}_{\text{error}} + \overset{\text{reg. para.}}{\gamma}\|\mathbf{x}\|_1$$

$\min \|\underline{z}\|_0 \overset{\text{e.g.}}{\equiv} \min \|\underline{x}\|_1,$

In machine learning context, questions of interests include:

- How to design the measurement/sampling matrix $\mathbf{\Phi}$?
- What are the efficient algorithms to search for $\mathbf{x}$?
- Are they stable under noisy inputs?
- How many measurements/samples are necessary/sufficient (i.e., size of $\mathbf{y}$)?

Insight: Turns out $m = \Omega(\log(n))$ random samples will suffice

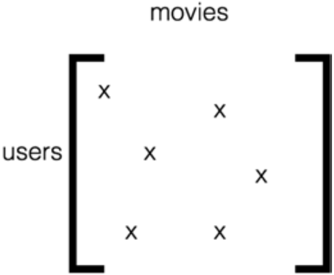
Some Optimization Algorithms for Compressed Sensing

- Shrink algorithms (for l_1 term) [Wright et al., 2009]
- Accelerated gradient [Beck and Teboulle, 2009b]
- ADMM [Zhang et al., 2010]
- Higher-order: Reduced inexact Newton [Wen et al., 2010]; Interior-point [Fountoulakis and Gondzio, 2013]

Example 4: Matrix Completion (Nonconvex)

In 2006, Netflix offered $1 million prize to improve movie rating prediction

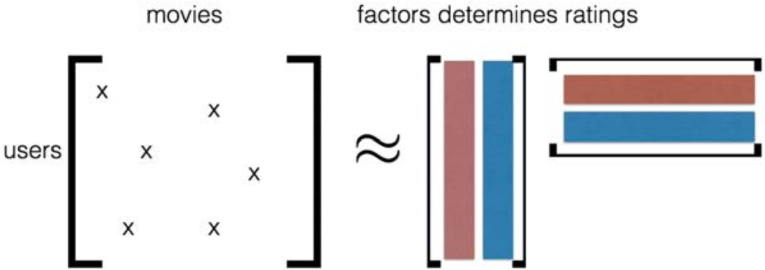
- How to estimate the missing ratings?



- About a million users, and 25,000 movies, with sparsely sampled ratings
- In essence, a low-rank matrix completion problem

Low-Rank Matrix Completion

- Completion Problem: Consider $\mathbf{M} \in \mathbb{R}^{n_1 \times n_2}$ to represent Netflix data, we may model it through factorization:



- In other words, the rank r of $\mathbf{M}$ is much smaller than its dimension $r \ll \min\{n_1, n_2\}$

Low-Rank Matrix Completion

In optimization, the low-rank matrix completion problem can be written as:

$$\begin{array}{lll} \underset{\mathbf{X}}{\text{Minimize}} & \underline{\operatorname{rank}(\mathbf{X})} & \\ \text{subject to} & (\mathbf{X})_{ij} = (\mathbf{M})_{ij}, & \forall i,j \in \text{observed entries} \end{array}$$

$\|\underline{X}\|_*$: nuclear norm $= \sum_i \sigma_i(\underline{X})$

In machine learning context, questions of interests include:

- What are the efficient algorithms to search for $\mathbf{X}$?
- Are they stable under noisy inputs and outliers?
- How many samples are necessary/sufficient (i.e., size of $(\mathbf{M})_{i,j}$)?

Insight: Turns out $m = \Omega(r \max\{n_1, n_2\} \log^2(\max\{n_1, n_2\}))$ samples will suffice

Some Optimization Algorithms for Matrix Completion

- (Block) Coordinate Descent [Wen et al., 2012]
- Shrink [Cai et al., 2010a; Lee et al., 2010]
- Stochastic Gradient [Lee et al., 2010]

Example 5: Phase Retrieval (Nonconvex)

- A classical topic from at least 1980s:
  - Recovery of a function given magnitude of its Fourier transform
  - Applications: optimal imaging, electron microscopy, crystallography, etc.

- Recover an $\mathbf{x}^* \in \mathbb{C}^d$ from a phase-less measurements:

$$y_k = |\langle \mathbf{a}_k, \mathbf{x} \rangle|^2, \quad k = 1, \ldots, M,$$

where $\mathbf{a}_k$ denotes some measurement vectors. The phase-retrieval problem can be formulated as an empirical risk minimization (ERM): problem

$$\min_{\mathbf{x}} \sum_{k=1}^{M} (y_k - |\langle \mathbf{a}_k, \mathbf{x} \rangle|^2)^2.$$

- Phase retrieval is nonconvex and unclear how to find a global minimum
  - Provable convergence result: [Candes et al. '15], [Yang et al. '19], [Wu and Rebeschini, '20], [Tan and Vershynin, '16], [Chen et al. '19]

Writinger flow (WF) alg.

Example 6: Deep Learning (Nonconvex)

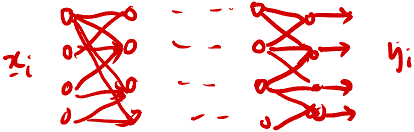
- Example: Train an L-layer fully-connected NN for supervised learning:

$$\min_{\mathbf{W}} \left\{ F(\mathbf{W}) \triangleq \frac{1}{m} \sum_{i=1}^{m} \ell(\mathbf{y}_i, f(\mathbf{x}_i, \mathbf{W})) \right\},$$

  - $\mathbf{W} = \{\mathbf{W}_1, \ldots, \mathbf{W}_L\}$, with $\mathbf{W}_i \in \mathbb{R}^{n_i \times n_{i-1}}$, are weights of NN model
  - $\{(\mathbf{x}_i, \mathbf{y}_i)_{i=1}^m\}$, $\mathbf{x}_i \in \mathbb{R}^{n_0}$, are training samples
  - $\ell(\cdot, \cdot)$ is a loss function (e.g., quadratic or logistic loss)
  - NN model can be written as:

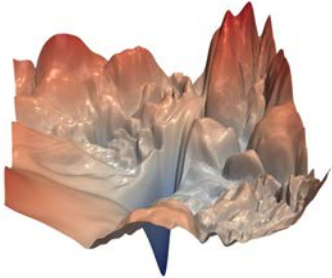
$$f(\mathbf{x}_i, \mathbf{W}) = \mathbf{W}_L \sigma(\mathbf{W}_{L-1} \sigma(\ldots, \sigma(\mathbf{W}_2 \sigma(\mathbf{W}_1 \mathbf{x}_i)) \ldots)),$$

where $\sigma(\cdot)$ is scalar-valued and called activation function.

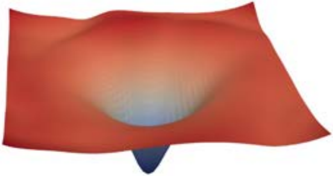


Example 6: Deep Learning (Nonconvex)

- Landscape of deep neural networks
  - Loss surfaces of ResNet-56 with/without skip connections [Li et al. '18]



(a) without skip connections



(b) with skip connections

- Training NN is NP-complete in general [Blum and Rivest, '89], but:
  - All local minima are global for 1-layer NN: [Soltanolkotabi et al. '18], [Haeffele and Vidal, '17], [Feizi et al. '17]
  - GD/SGD converge to global min for linear networks [Arora et al. '18], [Ji and Telgarsky, '19], [Shin, '19], wide over-parameterized networks [Allen-Zhu et al., '19], and pyramid networks [Nguyen and Mondelli, '19]

Next Class...

We will start from some related math background.