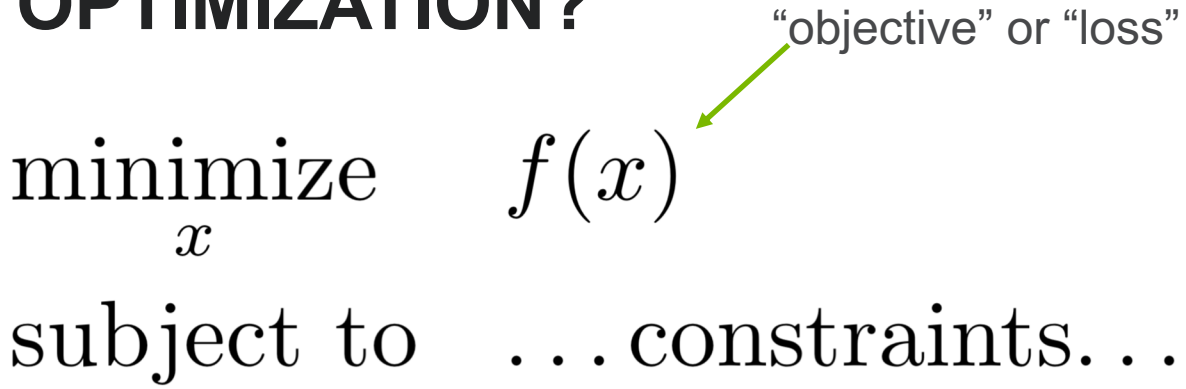
Argonne NATIONAL LABORATORY

OPTIMIZATION METHODS FOR MACHINE LEARNING

BETHANY LUSCH
Asst. Computer Scientist
Argonne National Lab
Leadership Computing Facility
blusch@anl.gov

August 9, 2019
ATPESC

WHAT IS OPTIMIZATION?



- “best route” (Minimize cost of delivery subject to all mail is delivered)
- “best product” (Minimize -profit subject to safety)
- “best prediction model” (Minimize prediction error)

MACHINE LEARNING

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f(x) \\ \text{subject to} \quad & \ldots \text{constraints} \ldots \end{aligned}$$

- “best prediction model” (Minimize prediction error)
- “best recommendation” (Minimize # who don’t buy anything)
- “best clusters” (Minimize distance within clusters while maximizing distance between clusters)

Underneath most ML problems is an optimization problem

TYPES OF OPTIMIZATION

- Linear
- Quadratic
- Convex
- Have 2nd derivs
- Have gradients
- General

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f_0(x) \\ \text{subject to} \quad & f_i(x) \leq b_i, \; i = 1, \ldots, m. \end{aligned}$$

TYPES OF OPTIMIZATION

- Linear
- Quadratic
- Convex
- Have 2nd derivs
- Have gradients
- General

Roughly…

More time on formulating problem to fit these categories

More time on optimization algorithm

LINEAR PROGRAMMING

Desperation

- Linear
- Quadratic
- Convex
- Have 2nd derivs
- Have gradients
- General

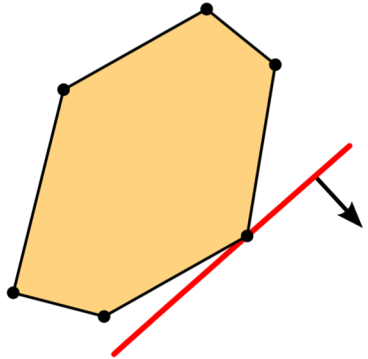
linear

$$c^T x$$

Convex polytope

$$Ax \leq b$$
$$x \geq 0.$$

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f_0(x) \\ \text{subject to} \quad & f_i(x) \leq b_i, \; i = 1, \ldots, m. \end{aligned}$$



Mature polynomial-time algorithms
Local minima are global optima

Picture source: Ylloh at wikipedia.org/wiki/Linear_programming

QUADRATIC PROGRAMMING

Desperation

- Linear
- Quadratic
- Convex
- Have 2nd derivs
- Have gradients
- General

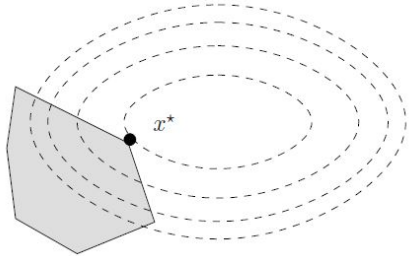
quadratic

$$\frac{1}{2}x^T Qx + c^T x$$

Convex polytope

$$Ax \leq b$$
$$x \geq 0.$$

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f_0(x) \\ \text{subject to} \quad & f_i(x) \leq b_i, \; i = 1, \ldots, m. \end{aligned}$$



If Q is positive definite:
Weakly polynomial-time algorithms
Local minima are global optima

CONVEX OPTIMIZATION

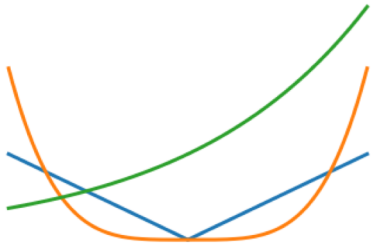
Desperation

- Linear
- Quadratic
- Convex
- Have 2nd derivs
- Have gradients
- General

Convex function

Convex set

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f_0(x) \\ \text{subject to} \quad & f_i(x) \leq b_i, \; i = 1, \ldots, m. \end{aligned}$$



In some cases: poly-time
Local minima are global optima
Reliable methods

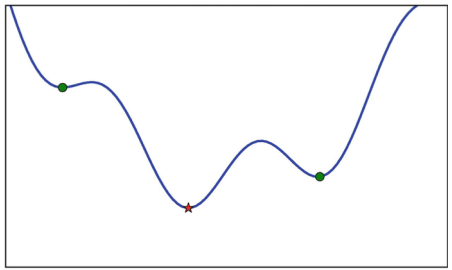
DIFFERENTIABLE OPTIMIZATION

Desperation

- Linear
- Quadratic
- Convex
- Have 2nd derivs
- Have gradients
- General

Differentiable function

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f_0(x) \\ \text{subject to} \quad & f_i(x) \leq b_i, \; i = 1, \ldots, m. \end{aligned}$$



Generally NP-hard
Local minima problematic
Can use gradients and ideally Hessians in algorithm

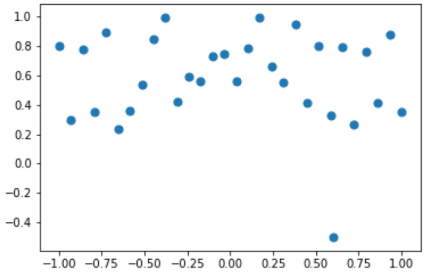
Picture source: Robert Johansson in “Optimization”

GENERAL OPTIMIZATION

Desperation

- Linear
- Quadratic
- Convex
- Have 2nd derivs
- Have gradients
- General

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f_0(x) \\ \text{subject to} \quad & f_i(x) \leq b_i, \; i = 1, \ldots, m. \end{aligned}$$

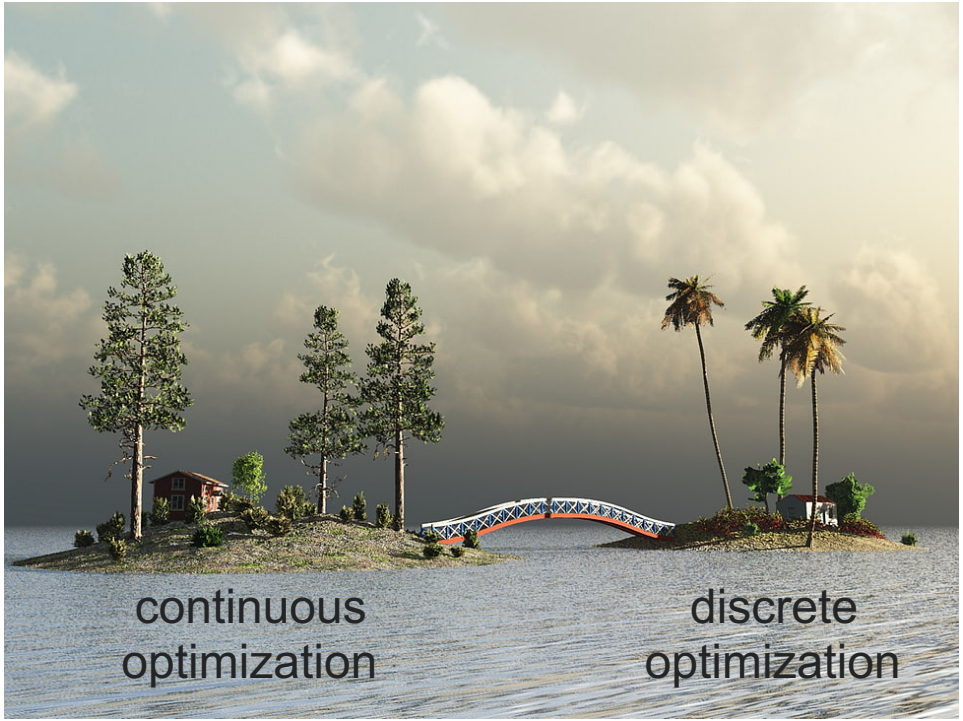


Generally NP-hard
Hopefully know *some* structure!

DISCRETE OPTIMIZATION

Desperation

- Linear
- Quadratic
- Convex
- Have 2nd derivs
- Have gradients
- General



Picture source: wallpaperflare.com

CLASSIFICATION EXAMPLE

- Problem: label each document x as related to politics or not (1 or -1).
- Hard to come up with rules by hand, so ML helps: learn function $h(x)$
- Really want to minimize expected risk of misclassification:

$$\underset{h}{\text{minimize}} \quad R(h) = \mathbb{P}[h(x) \neq y] = \mathbb{E}[\mathbb{1}[h(x) \neq y]]$$

- How do we pick family of functions to optimize over?
- How do we know which one is optimal?

REALITIES

- Really want to minimize expected risk of misclassification:

$$\underset{h}{\text{minimize}} \quad R(h) = \mathbb{P}[h(x) \neq y] = \mathbb{E}[\mathbb{1}[h(x) \neq y]]$$

- Don't know probability distribution, so minimize empirical risk:

$$\underset{h}{\text{minimize}} \quad R_n(h) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}[h(x_i) \neq y_i]$$

- Easier if smooth loss and parameterized h:

$$\underset{w}{\text{minimize}} \quad R_n(h) = \frac{1}{n} \sum_{i=1}^{n} \ell(h(x_i; w), y_i)$$

CHOOSING FUNCTION FAMILY

- Possibility of low empirical risk on training data
- Expected risk and empirical risk don't have large gap

} Bias vs. variance

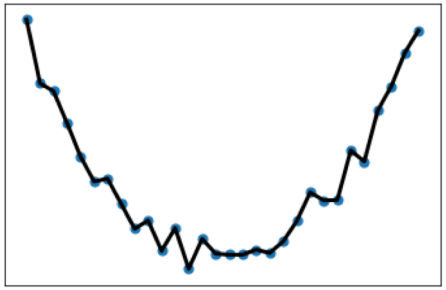
} Overfitting vs. underfitting

- Can efficiently solve optimization
- (convenient representation, smoothness,…)

Major themes of machine learning!

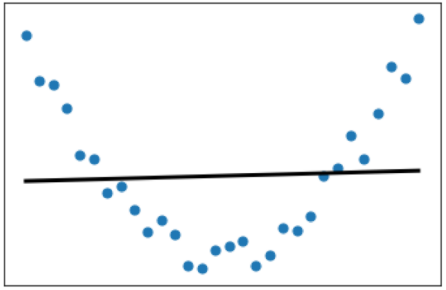
BIAS VS. VARIANCE

High variance



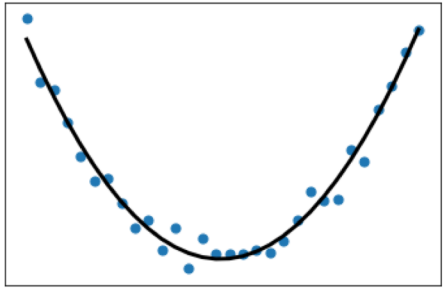
overfitting

High bias



underfitting

Low bias, low variance



balanced

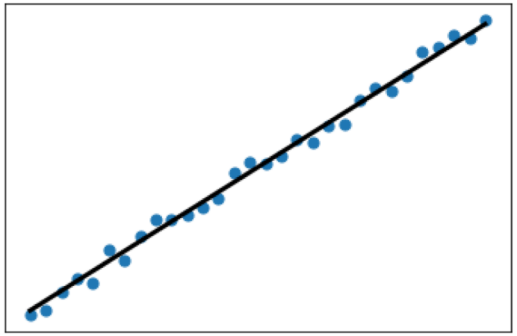
Picture similar to: Seema Singh, “Understanding the Bias-Variance Tradeoff”

LINEAR REGRESSION (LEAST-SQUARES)

Desperation

- Linear
- Quadratic
- Convex
- Have 2nd derivs
- Have gradients
- General

$$\underset{x}{\text{minimize}} \quad \|Ax - b\|_2^2$$



if multiply out:

$$\underset{x}{\text{minimize}} \quad x^t A^T A x - 2b^T A x + b^T b$$

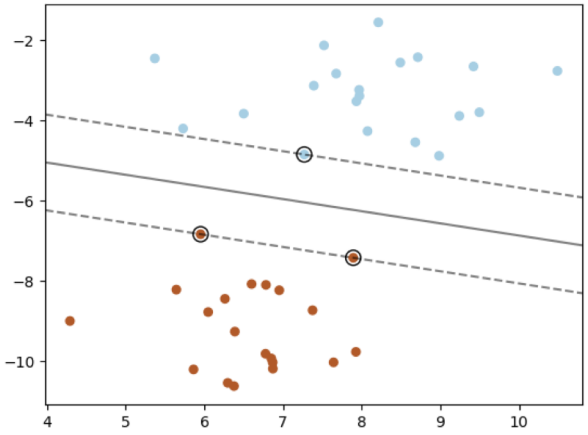
… quadratic program!

SUPPORT VECTOR MACHINE

Desperation

- Linear
- Quadratic
- Convex
- Have 2nd derivs
- Have gradients
- General

Find linear classifier with maximum margin



$$\begin{aligned} \underset{a,b,u,v}{\text{minimize}} \quad & \|a\|_2 + \gamma(\mathbf{1}^T u + \mathbf{1}^T v) \\ \text{subject to} \quad & a^T x_i - b \geq 1 - u_i, \ i = 1, \ldots, N \\ & a^T y_i - b \leq -(1 - v_i), \ i = 1, \ldots, M \\ & u \geq 0, v \geq 0 \end{aligned}$$

… quadratic program!

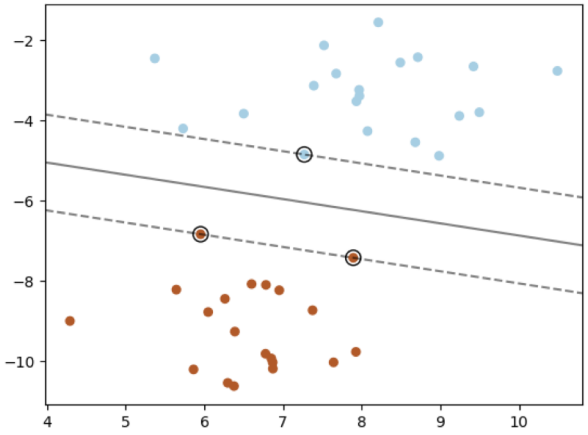
Picture source: https://scikit-learn.org/stable/modules/svm.html

SUPPORT VECTOR MACHINE

Desperation

- Linear
- Quadratic
- Convex
- Have 2nd derivs
- Have gradients
- General

Find linear classifier with maximum margin



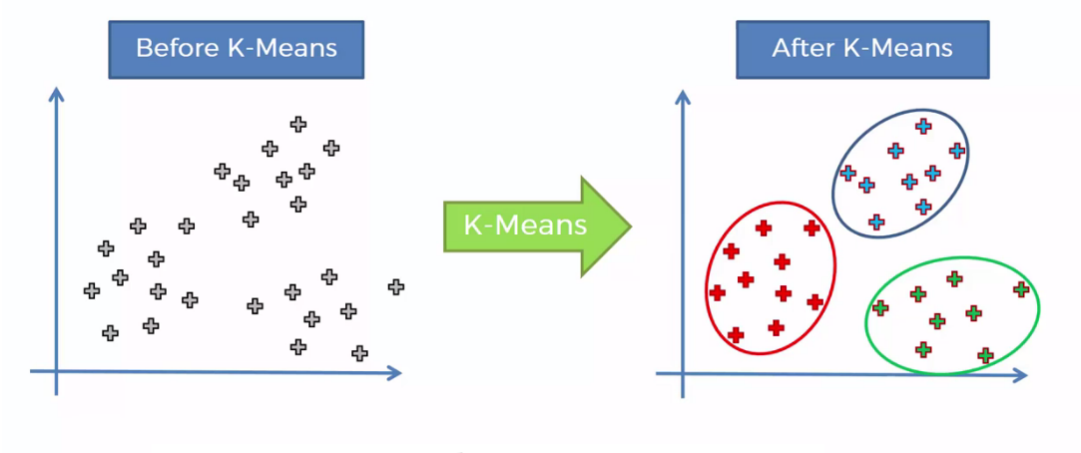
Kernel SVM can do nonlinear classification while remaining a quadratic program

K-MEANS CLUSTERING

Desperation

- Linear
- Quadratic
- Convex
- Have 2nd derivs
- Have gradients
- General

Find clustering that minimizes distances within clusters

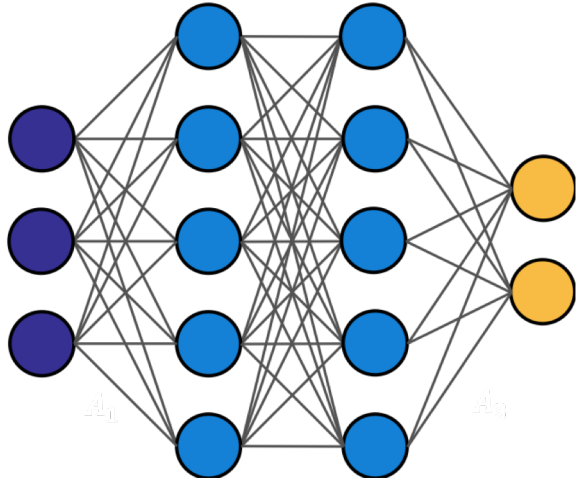


$$\underset{S}{\text{minimize}} \sum_{i=1}^{k} \sum_{x \in S_i} \|x - \mu\|_2^2$$

NP-hard discrete problem, so use approx. algorithm

Picture source: Prasad Patil, “K Means Clustering : Identifying F.R.I.E.N.D.S in the World of Strangers”

DEEP LEARNING



Desperation

- Linear
- Quadratic
- Convex
- Have 2nd derivs
- Have gradients
- General

Too expensive

$$\underset{\theta}{\text{minimize}} \quad \|x - f(x)\|_2^2$$

Getting desperate, plus want to be scale well, so use gradient descent

RECALL: TYPES OF OPTIMIZATION

- Linear
- Quadratic
- Convex
- Have 2nd derivs
- Have gradients
- General

Roughly…

More time on formulating problem to fit these categories

More time on optimization algorithm

ANALOGOUSLY…

- Linear
- Quadratic
- Convex
- Have 2nd derivs
- Have gradients
- General

Roughly…

More time on formulating problem (choosing features) so that these (biased) methods are suitable

More time on optimization algorithm

REMINDER

- Possibility of low empirical risk on training data
- Expected risk and empirical risk don't have large gap

} Bias vs. variance

} Overfitting vs. underfitting

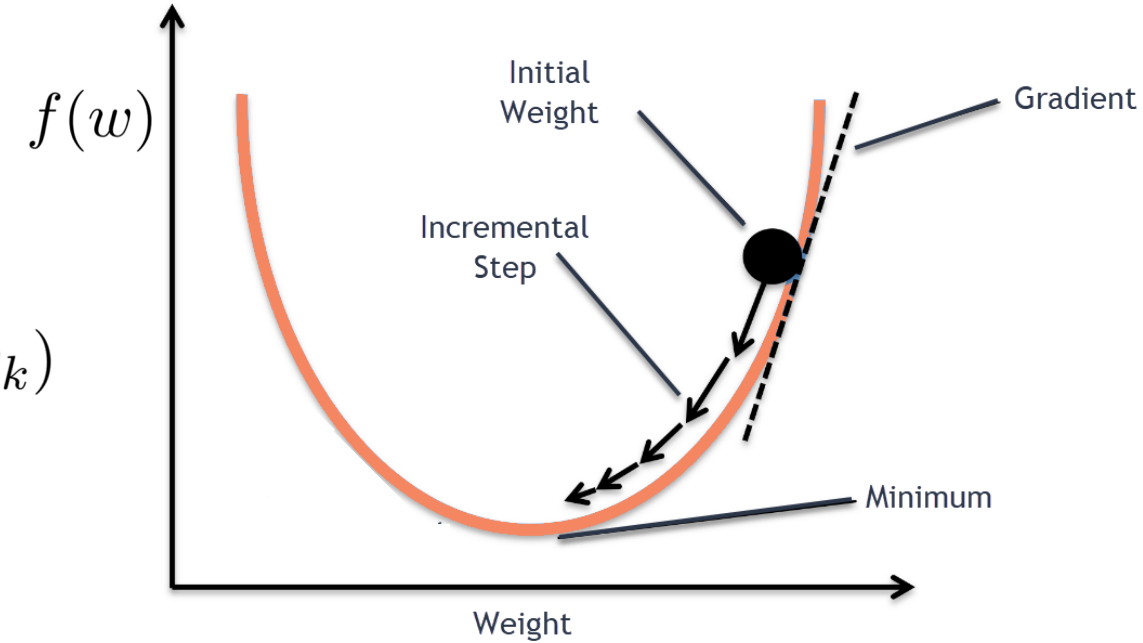
- Can efficiently solve optimization

- Big neural networks can be very expressive (low bias)
- So don't need to be as clever about input features
- But then easy to overfit…
- Optimization is tricky: optimization stalls, plus local minima or saddle points

GRADIENT DESCENT

$$\underset{w}{\text{minimize}} \quad f(w)$$

$$w_{k+1} \leftarrow w_k - \alpha_k \nabla f(w_k)$$



Picture source: Divakar Kapil in “Stochastic vs Batch Gradient Descent”

TYPES OF GRADIENT DESCENT

i.e. for empirical risk, explicitly summing over data points

$$R_n(w) = \frac{1}{n} \sum_{i=1}^{n} f_i(w)$$

$$w_{k+1} \leftarrow w_k - \frac{\alpha_k}{n} \sum_{i=1}^{n} \nabla f_{ik}(w_k)$$

Batch GD: use all points every step

$$w_{k+1} \leftarrow w_k - \alpha_k \nabla f_{ik}(w_k)$$

Stochastic GD: use one point per step

$$w_{k+1} \leftarrow w_k - \frac{\alpha_k}{|S_k|} \sum_{i \in S_k} \nabla f_{ik}(w_k)$$

Mini-batch GD: use a subset each step

TYPES OF GRADIENT DESCENT

Batch GD: use all points every step — Each step is accurate but expensive

Stochastic GD: use one point per step — Each step is noisy but fast

Mini-batch GD: use a subset each step — Happy medium?

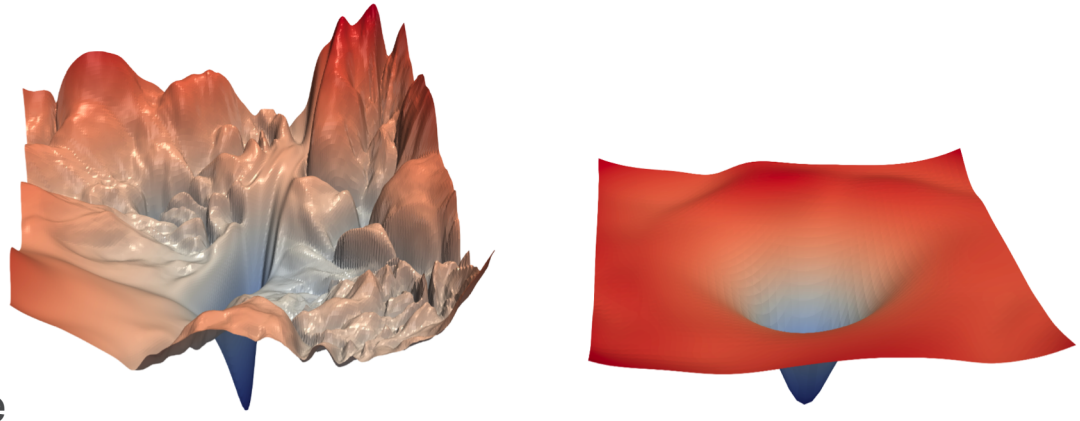
Very common in deep learning, but often call it SGD

GRADIENT DESCENT CONSIDERATIONS

$$\underset{w}{\text{minimize}} \quad f(w)$$

$$w_{k+1} \leftarrow w_k - \alpha_k \nabla f(w_k)$$

- Step size α_k
  - Too big: overshoot
  - Too small: very slow
  - (But can be good to escape local minima)
- Initialization
- Can you make the problem easier?



(a) without skip connections

(b) with skip connections

Li, et al. “Visualizing the Loss Landscape of Neural Nets” NeurIPS 2018

VARIANT: ADAM

Popular improvement on GD: Adam optimizer

- Separate learning rate for each weight
- Momentum: uses moving average of the gradient
- Also incorporates squared gradients

Cool exploration/visualization of momentum: https://distill.pub/2017/momentum/

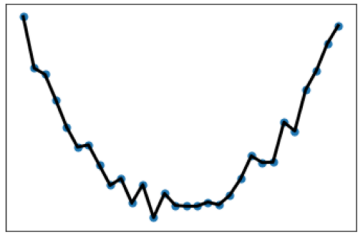
(For those familiar: combines the best properties of AdaGrad, momentum, and RMSProp)

REGULARIZATION

- Common way to avoid overfitting: regularization
- Most common: L2 regularization

balance

$$\underset{w}{\text{minimize}} \quad \underbrace{\frac{1}{n}\sum_{i=1}^{n}(h(x_i;w)-y_i)^2}_{\text{error}} + \underbrace{\lambda\|w\|_2^2}_{\text{regularization}}$$



Roughly: big coefficients/weights correspond to large variation

OVERFITTING CAUTION

- Can you generalize outside of your training set to validation/testing set?
- What about interpolating to data you haven't collected?
- Extrapolation extra unlikely to work

SUMMARY

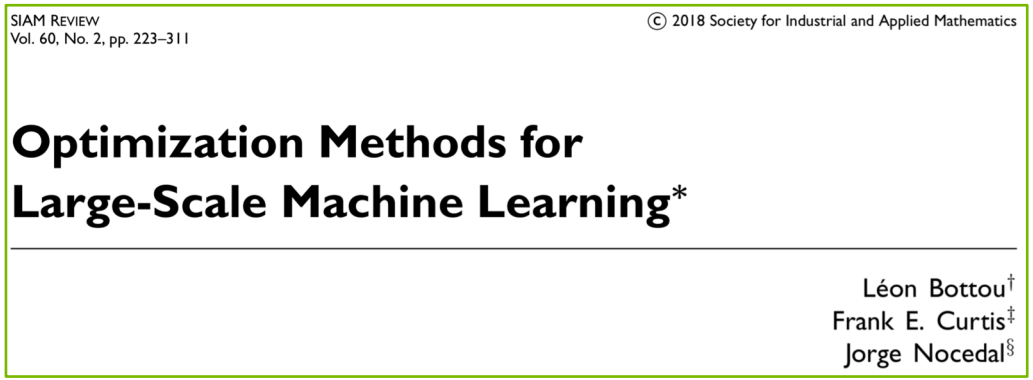
Desperation

- Linear
- Quadratic
- Convex
- Have 2nd derivs
- Have gradients
- General

Major themes in machine learning:

- Overfitting vs. underfitting
- Ability to efficiently solve optimization problem

For more, see:



SIAM REVIEW
Vol. 60, No. 2, pp. 223–311

Optimization Methods for Large-Scale Machine Learning*

Léon Bottou†
Frank E. Curtis‡
Jorge Nocedal§

ANY QUESTIONS?

Thinking ahead to next talk: how would you parallelize gradient descent?