Stat 212b: Topics in Deep Learning
Lecture 21

Joan Bruna
UC Berkeley

Iterative Optimization Algorithms

- [Bottou & Bousquet '08] study four main iterative algorithms in the large-scale learning regime:
  - Gradient Descent
  - Second Order Gradient Descent (i.e. Newton method)
  - Stochastic Gradient Descent (SGD)
  - Second Order Gradient Descent.

- Assumptions:
  - Signal class $\mathcal{F}$ is fixed,
  - linearly parametrized by $w \in \mathbb{R}^d$: $\Phi_w(x) = \langle \Phi(x), w \rangle$.

    loss functions $w \mapsto \ell(\Phi_w(x), y)$ convex and twice differentiable.

Iterative Optimization

- Let H and G be respectively the Hessian and gradient covariance matrices at the empirical optimum $w_n = \arg\min_w F_n(\Phi_w)$:

$$H = \frac{\partial^2 F_n}{\partial w^2}(\Phi_{w_n}) = \frac{1}{n}\sum_i \frac{\partial^2 \ell(\Phi_w(x_i), y_i)}{\partial w^2}$$

$$G = \frac{1}{n}\sum_i \left(\frac{\partial \ell(\Phi_w(x_i), y_i)}{\partial w}\right)\left(\frac{\partial \ell(\Phi_w(x_i), y_i)}{\partial w}\right)^T .$$

Iterative Optimization

- Let H and G be respectively the Hessian and gradient covariance matrices at the empirical optimum $w_n = \arg\min_w F_n(\Phi_w)$:

$$H = \frac{\partial^2 F_n}{\partial w^2}(\Phi_{w_n}) = \frac{1}{n}\sum_i \frac{\partial^2 \ell(\Phi_w(x_i), y_i)}{\partial w^2}$$

$$G = \frac{1}{n}\sum_i \left(\frac{\partial \ell(\Phi_w(x_i), y_i)}{\partial w}\right)\left(\frac{\partial \ell(\Phi_w(x_i), y_i)}{\partial w}\right)^T .$$

- Suppose that

$$\lambda(H) \subset [\lambda_{min}, \lambda_{max}] \ , \ \text{with } \lambda_{min} > 0$$

$$tr(GH^{-1}) \leq \nu \ .$$

- Condition number: $\kappa = \lambda_{max}/\lambda_{min}$.

Gradient Descent (GD)

$$w_{t+1} = w_t - \eta \nabla_w F_n(\Phi_{w_t}) \ .$$

- When step size $\eta = \lambda_{max}^{-1}$, $O(\kappa \log(\rho^{-1}))$ iterations to reach accuracy ρ (linear convergence).

	Cost per iteration	Iterations to reach ρ	Time to reach accuracy ρ	Time to reach $F(\tilde{\Phi}_n) - F(\Phi^*_{\mathcal{F}} < \epsilon$
GD	$O(nd)$	$O(\kappa \log \rho^{-1})$	$O(nd\kappa \log \rho^{-1})$	$O\left(d^2 \kappa \epsilon^{-1/\alpha} \log^2(\epsilon^{-1})\right)$

Second Order Gradient Descent

$$w_{t+1} = w_t - H^{-1}\nabla_w F_n(\Phi_{w_t}) \ , \ H^{-1} \text{ known in advance.}$$

	Cost per iteration	Iterations to reach ρ	Time to reach accuracy ρ	Time to reach $F(\tilde{\Phi}_n) - F(\Phi^*_{\mathcal{F}} < \epsilon$
GD	$O(nd)$	$O(\kappa \log \rho^{-1})$	$O(nd\kappa \log \rho^{-1})$	$O\left(d^2\kappa\epsilon^{-1/\alpha}\log^2(\epsilon^{-1})\right)$
2GD	$O((n+d)d)$	$O(\log\log \rho^{-1})$	$O((n+d)d \log\log \rho^{-1})$	$O\left(d^2\epsilon^{-1/\alpha}\log\log(\epsilon^{-1})\log(\epsilon^{-1})\right)$

- Optimization speed is much faster
- The problem does not depend on condition number.

Stochastic Gradient Descent (SGD)

- At each t, we draw random z_t from training set.

$$w_{t+1} = w_t - \frac{\eta}{t} \nabla_w f(\Phi_w(z_t)) \ .$$

- With $\eta = \lambda_{min}^{-1}$, we have $\|w_t - w_n\| = O(1/\sqrt{t})$.

	Cost per iteration	Iterations to reach ρ	Time to reach accuracy ρ	Time to reach $F(\tilde{\Phi}_n) - F(\Phi^*_{\mathcal{F}}) < \epsilon$
GD	$O(nd)$	$O(\kappa \log \rho^{-1})$	$O(nd\kappa \log \rho^{-1})$	$O\left(d^2 \kappa \epsilon^{-1/\alpha} \log^2(\epsilon^{-1})\right)$
2GD	$O((n+d)d)$	$O(\log \log \rho^{-1})$	$O((n+d)d \log \log \rho^{-1})$	$O\left(d^2 \epsilon^{-1/\alpha} \log \log(\epsilon^{-1}) \log(\epsilon^{-1})\right)$
SGD	$O(d)$	$\nu\kappa^2\rho^{-1} + o(\rho^{-1})$	$O(\frac{d\nu\kappa^2}{\rho})$	$O(\frac{d\nu\kappa^2}{\epsilon})$

- Optimization speed is much worse than GD.
- However, learning speed is better.

Second Order Stochastic Gradient Descent (2SGD)

- At each t, we draw random z_t from training set.

$$w_{t+1} = w_t - \frac{H^{-1}}{t} \nabla_w f(\Phi_w(z_t)) \ .$$

	Cost per iteration	Iterations to reach ρ	Time to reach accuracy ρ	Time to reach $F(\tilde{\Phi}_n) - F(\Phi^*_{\mathcal{F}}) < \epsilon$
GD	$O(nd)$	$O(\kappa \log \rho^{-1})$	$O(nd\kappa \log \rho^{-1})$	$O\left(d^2 \kappa \epsilon^{-1/\alpha} \log^2(\epsilon^{-1})\right)$
2GD	$O((n+d)d)$	$O(\log \log \rho^{-1})$	$O((n+d)d \log \log \rho^{-1})$	$O\left(d^2 \epsilon^{-1/\alpha} \log \log(\epsilon^{-1}) \log(\epsilon^{-1})\right)$
SGD	$O(d)$	$\nu\kappa^2\rho^{-1} + o(\rho^{-1})$	$O(\frac{d\nu\kappa^2}{\rho})$	$O(\frac{d\nu\kappa^2}{\epsilon})$
2SGD	$O(d^2)$	$\nu\rho^{-1} + o(\rho^{-1})$	$O(\frac{d^2\nu}{\rho})$	$O(\frac{d^2\nu}{\epsilon})$

- Iteration is more expensive, but less iterations.
- Constants are affected.

Objectives

- Accelerated Gradient Descent

- Regularization
  - Weight Decay
  - Dropout

Accelerated Gradient Descent

- We saw that Gradient Descent, when applied to smooth convex functions, has a rate of convergence $1/T$ after T steps:

Theorem: If f is convex on $\mathbb{R}^n$ and
for any $x, y \in \mathbb{R}^n$ one has $\|\nabla f(x) - \nabla f(y)\| \leq \beta \|x - y\|$,
then the gradient descent with step $\eta = \beta^{-1}$ satisfies

$$f(x_t) - \min_x f(x) \leq \frac{2\beta \|x_1 - \arg\min_x f(x)\|^2}{t+3} \ .$$

Accelerated Gradient Descent

- We saw that Gradient Descent, when applied to smooth convex functions, has a rate of convergence $1/T$ after T steps:

Theorem: If f is convex on $\mathbb{R}^n$ and for any $x, y \in \mathbb{R}^n$ one has $\|\nabla f(x) - \nabla f(y)\| \leq \beta \|x - y\|$, then the gradient descent with step $\eta = \beta^{-1}$ satisfies

$$f(x_t) - \min_x f(x) \leq \frac{2\beta \|x_1 - \arg\min_x f(x)\|^2}{t+3} \ .$$

- Q: Can we improve this rate using only first order information?

Accelerated Gradient Descent

- Use a momentum term (Nesterov,'83):

$$\lambda_0 = 0 \ , \ \lambda_t = \frac{1 + \sqrt{1 + 4\lambda_{t-1}^2}}{2} \ , \ \gamma_t = \frac{1 - \lambda_t}{\lambda_{t+1}} \ .$$

$$y_{t+1} = x_t - \frac{1}{\beta} \nabla f(x_t) \ ,$$

$$x_{t+1} = (1 - \gamma_t) y_{t+1} + \gamma_t y_t \ .$$

- Same complexity as Gradient descent.

Accelerated Gradient Descent

- But better provable convergence rate:

Theorem: (Nesterov 83) If f is convex on $\mathbb{R}^n$ and for any $x, y \in \mathbb{R}^n$ one has $\|\nabla f(x) - \nabla f(y)\| \leq \beta \|x - y\|$, then the accelerated gradient descent satisfies

$$f(y_t) - \min_x f(x) \leq \frac{2\beta \|x_1 - \arg\min_x f(x)\|^2}{t^2} .$$

Accelerated Gradient Descent

- But better provable convergence rate:

Theorem: (Nesterov 83) If f is convex on $\mathbb{R}^n$ and for any $x, y \in \mathbb{R}^n$ one has $\|\nabla f(x) - \nabla f(y)\| \leq \beta \|x - y\|$, then the accelerated gradient descent satisfies

$$f(y_t) - \min_x f(x) \leq \frac{2\beta \|x_1 - \arg\min_x f(x)\|^2}{t^2} .$$

- Q: Can we do better with a first order method?

Accelerated Gradient Descent

- Not in general:

Theorem: For any black-box optimization algorithm such that $x_{s+1} \in x_1 + \text{span}\{\nabla f(x_1), \ldots, \nabla f(x_s)\}$
and for any $t \leq (n-1)/2$, there exists f convex and β-smooth such that
$\min_{s \leq t} f(x_s) - \min_x f(x) \geq C\beta \frac{\|x_1 - \arg\min_x f(x)\|^2}{(t+1)^2}$.

- Second order methods (e.g. Newton) have access to more information: not concerned with this result.

Interlude: Interpretation of Accelerated G.

- Nesterov's method is typically associated to a *momentum* term:

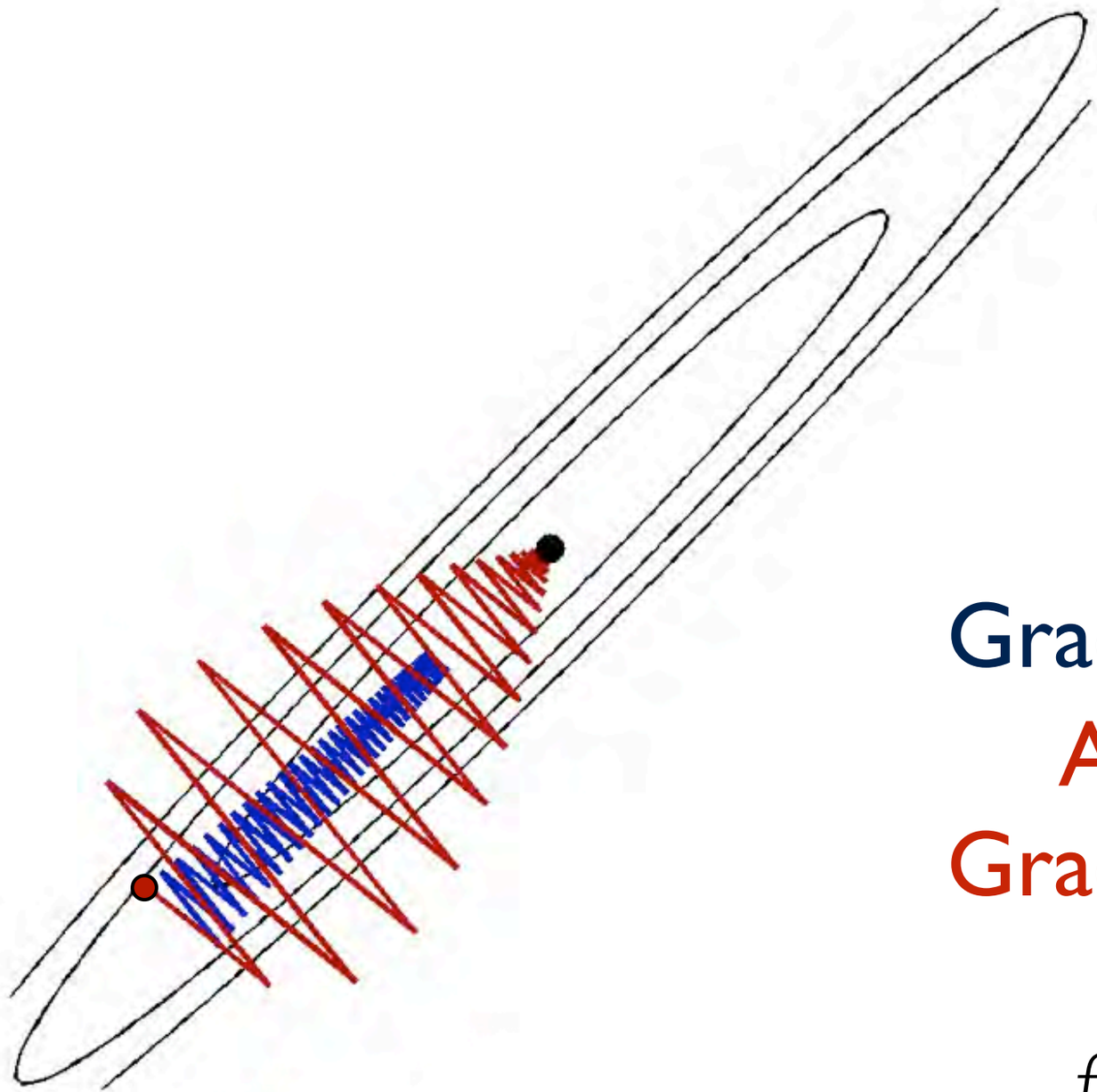
$$y_{t+1} = x_t - \frac{1}{\beta}\nabla f(x_t) \ ,$$

$$x_{t+1} = (1-\gamma_t)y_{t+1} + \gamma_t y_t \ .$$

$$x_{t+1} - x_t = -\gamma_t[x_t - x_{t-1}] + \frac{\gamma_t}{\beta}[\nabla f(x_t) - \nabla f(x_{t-1})] \ .$$

$$\dot{x}_{t+1} = -\gamma_t \dot{x}_t + \frac{\gamma_t}{\beta}\dot{\nabla} f(x_t) \ .$$

Interlude: Interpretation of Accelerated G.

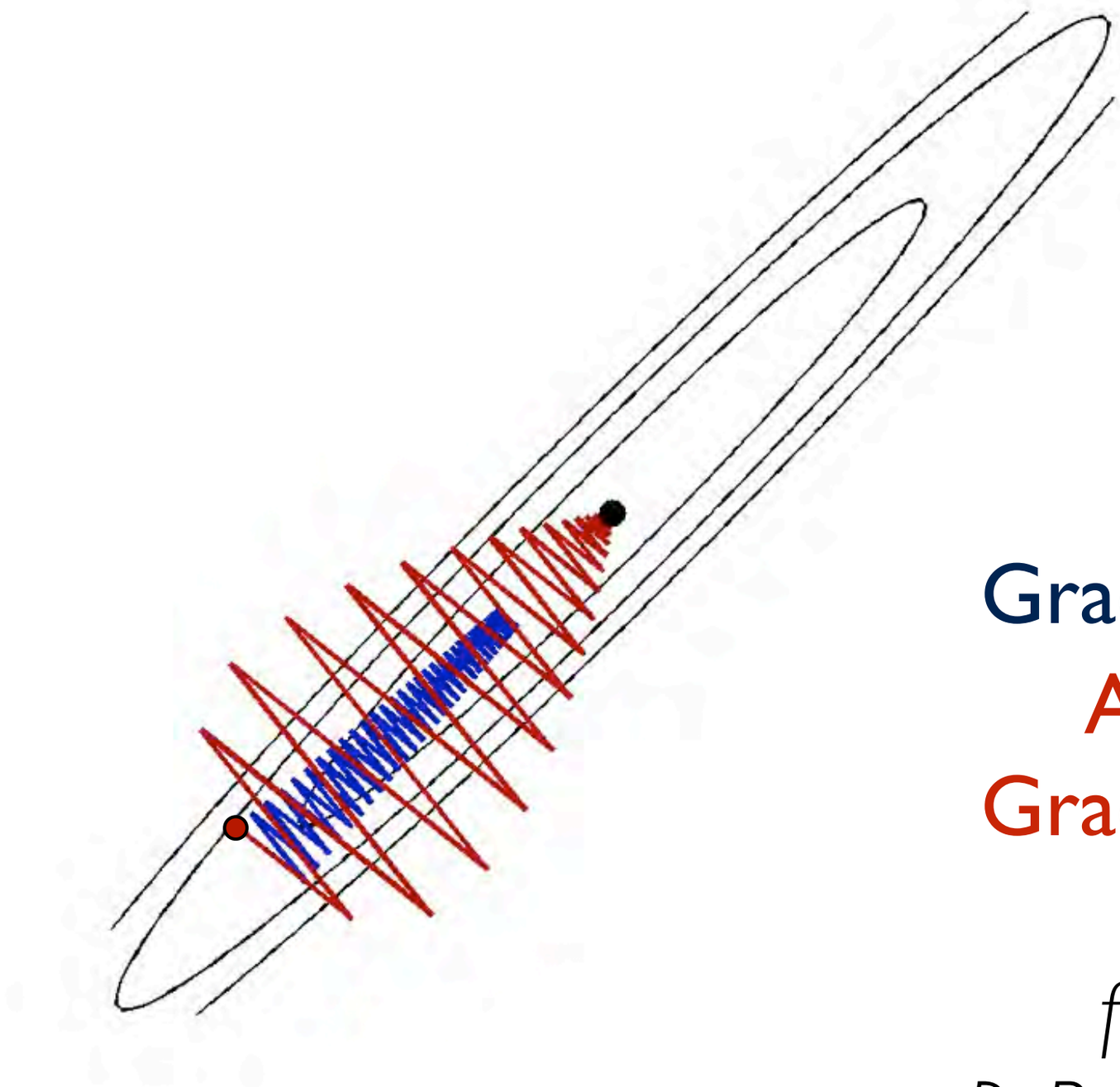


Gradient Descent

Accelerated Gradient Descent

figure credit: B. Recht [Simons'13]

Interlude: Interpretation of Accelerated G.



Gradient Descent

Accelerated Gradient Descent

figure credit: B. Recht [simons'13]

- Q: Why does it work?

Interlude: Interpretation of Accelerated G.

(from M. Hardt)

- Suppose we want to minimize

$$f(x) = \frac{1}{2} x^T A x - b^T x \ .$$

$$A \in \mathbb{R}^{n \times n} \text{ positive definite.}$$

- Its unique minimum is at $x^* = A^{-1} b$:

$$\nabla f(x) = 0 \Leftrightarrow Ax = b \ .$$

- We run Gradient Descent starting at $x_0 = 0$ with step t:

$$x_{k+1} = x_k - t \nabla f(x_k) = [I - tA] x_k + tb \ .$$

- At iteration k we thus have

$$x_{k+1} = \left(\sum_{j \leq k} (I - tA)^j \right) (tb) \ .$$

Interlude: Interpretation of Accelerated G.

(from M. Hardt)

- Let $0 < l \leq L < \infty$ be the spectral bounds of A:

$$\forall\ x\ ,\ l\|x\| \leq \|Ax\| \leq L\|x\|\ .$$

- $\Rightarrow$ Eigenvalues of $(I - tA) \in (0, 1)$ if $t < L^{-1}$.
- Thus

$$(tA)^{-1} = [I - (I - tA)]^{-1} = \sum_{j=0}^{\infty}(I - tA)^j$$

$$\|(tA)^{-1} - \sum_{j=0}^{k}(I - tA)^j\| = O(\|(I - tA)^k\|) = O((1 - \frac{l}{L})^k)\ .$$

- This corresponds to the rate of standard gradient descent for strongly convex functions:

$$\|x_k - x^*\| \leq (1 - 2(\kappa + 1)^{-1})^k\|x_0 - x^*\|\ .$$

Interlude: Interpretation of Accelerated G.

(from M. Hardt)

- We are thus approximating $(tA)^{-1}$ with a polynomial $q_k(A)$ of degree k.
- Q: What is the best polynomial approximation in our setting?

$$\text{for each } k \ , \ \min_{q_k} \|I - Aq_k(A)\| \ .$$

-

Interlude: Interpretation of Accelerated G.

(from M. Hardt)

- We are thus approximating $(tA)^{-1}$ with a polynomial $q_k(A)$ of degree k.
- Q: What is the best polynomial approximation in our setting?

$$\text{for each } k \ , \ \min_{q_k} \|I - Aq_k(A)\| \ .$$

- A: Since A has eigenvalues in $[l, L]$, and we need $q_k(0) = 1$, Chebyshev polynomials are optimal:

Lemma: There is a polynomial p_k of degree $O(\sqrt{(L/l)\log(\epsilon^{-1})})$ such that $p_k(0) = 1$ and $|p_k(x)| \leq \epsilon$ for all $x \in [l, L]$.

Interlude: Interpretation of Accelerated G.

- We are thus approximating $(tA)^{-1}$ with a polynomial $q_k(A)$ of degree k.
- Q: What is the best polynomial approximation in our setting?

$$\text{for each } k \ , \ \min_{q_k} \|I - Aq_k(A)\| \ .$$

- A: Since A has eigenvalues in $[l, L]$, and we need $q_k(0) = 1$, Chebyshev polynomials are optimal:

Lemma: There is a polynomial p_k of degree $O(\sqrt{(L/l)\log(\epsilon^{-1})})$ such that $p_k(0) = 1$ and $|p_k(x)| \leq \epsilon$ for all $x \in [l, L]$.

Moreover, p_k can be computed recursively from previous **two** polynomials. It results that

$x_{k+1} = x_k + \alpha_k \nabla f(x_k) + \beta_k \nabla f(x_{k-1})$ for suitable α_k, β_k .

gives a convergence rate $\|x_k - x^*\| = O(\beta^k)$ with $\beta = 1 - 2(\sqrt{\kappa} + 1)^{-1}$ *("minimax" optimal)*

Other existing analysis

- Geometric optimization from [Bubeck et al. ,'15]
  - Adaptation of the Ellipsoid Method

- ODE analysis from [Su, Boyd, Candes,'14]
  - Show that Nesterov accelerated gradient is the discretization of a first-order ODE.

Many other results

- Stochastic Gradient improvements to recover better convergence rate across many settings:
  - Stochastic Average Gradient [Le Roux et al, 2012]
  - Stochastic Dual Coordinate Ascent [Shalev-Shwarth et al,'12]
  - Stochastic Variance Reduced Gradient Descent [Johnson et al '13]
  - etc…
- See [Bubeck,'14] for an extensive treatment of convex optimization.

Generalization Error

- Recall

$$\Phi^* = \arg\min_{\Phi} F(\Phi) \ , \ \text{optimal model} \ ,$$

$$\Phi^*_{\mathcal{F}} = \arg\min_{\Phi \in \mathcal{F}} F(\Phi) \ , \ \text{optimal achievable model in } \mathcal{F} \ ,$$

$$\Phi_{\mathcal{F},n} = \arg\min_{\Phi \in \mathcal{F}} \hat{F}_n(\Phi) \ , \ \text{optimal empirical model in } \mathcal{F} \ ,$$

$$\widetilde{\Phi}_{\mathcal{F},n} = \ \text{solution of our optimization of } \min_{\Phi \in \mathcal{F}} \hat{F}_n(\Phi) \ ,$$

- With

$$F(\Phi) = \mathbb{E}_{z \sim \pi} f(z; \Phi) \ . \qquad \hat{F}_n(\Phi) = \frac{1}{n} \sum_{i \leq n} f(z_i; \Phi) \ .$$

- Q: How to modify our optimization in order to improve generalization error?

Popular Regularization Strategies

- Tikhonov regularization [Tikhonov'43]
  - aka ridge regression [Hoerl'62]
  - aka Weight decay [krogh, hertz'91].
- Dropout [Hinton et al'12]
- Lasso [Efron], L1 regularization [Ng]
- Model averaging (ensemble methods)
  - Bagging
  - Boosting
  - Bayesian ensembles
- "Computational" Regularization
  - See [Bach'13], [Less is More: Nystrom Computational Regularization, Rudi et al'15].

Tikhonov Regularization

- Suppose we have the following inverse linear problem

$$\min_x \|y - Ax\|^2 \ , \ A \in \mathbb{R}^{n \times p} \ .$$

- When $p \leq \text{rank}(A)$, the system has unique solution

$$A^T(y - Ax) = 0 \Rightarrow x^* = (A^T A)^{-1} A^T y = A^\dagger y \ .$$

$A^\dagger = (A^T A)^{-1} A^T$: Moore-Penrose pseudoinverse of A.

- When $p > \text{rank}(A)$, under-determined system.
  Which solution to select?

  Tikhonov proposed selecting the solution x^* having smallest norm $\|\Gamma^{1/2} x\|$:

$$\min_{Ax=y} \langle x, \Gamma x \rangle \ , \ \Gamma : \text{Tikhonov psd kernel.}$$

Tikhonov Regularization Examples

- $\Gamma = \lambda I$: Ridge regression.

  Let $K = \ker A$ and $K^{\perp}$ its orthogonal complement.

  Then $\|x\|^2 = \|P_K x\|^2 + \|P_{K^{\perp}} x\|^2$, and $Ax = AP_{K^{\perp}} x$.

  Thus we project the solution onto the space $K^{\perp}$.

  If $A = USV^T$ is the SVD of A, then

$$A^{\dagger} = V \bar{S} U^T \ , \ \bar{s}_{ii} = \begin{cases} \frac{1}{s_{ii}} & \text{if } s_{ii} > 0 \ , \\ 0 & \text{otherwise.} \end{cases} \ .$$

- $\langle x, \Gamma x \rangle = \int |\xi|^2 |\hat{x}(\xi)|^2 d\xi$: Sobolev Norm

Tikhonov Regularization

- Examples:

$$\Gamma = \lambda I \ : \ \text{Ridge regression.}$$

$$\langle x, \Gamma x \rangle = \int |\xi|^2 |\hat{x}(\xi)|^2 d\xi \ : \ \text{Sobolev Norm}$$

- Lagrangian formulation:

$$\min_x \frac{1}{2} \|y - Ax\|^2 + \lambda \langle x, \Gamma x \rangle$$

$$-A^T(y - Ax^*) + \lambda \Gamma x^* = 0 \Leftrightarrow (A^T A + \lambda \Gamma) x^* = A^T y$$

$$\Rightarrow x^* = (A^T A + \lambda \Gamma)^{-1} A^T y$$

Tikhonov Regularization

- When $\Gamma = I$, using again the SVD, the predictions become

$$\hat{y} = Ax^* = A(A^TA + \lambda I)^{-1}A^Ty = \sum_{k=1}^{p} \frac{s_{kk}^2}{\lambda + s_{kk}^2} u_k u_k^T$$

  - Shrinkage affects smaller empirical singular values than larger ones.
  - Sample small eigenvectors/eigenvalues are more unreliable than larger ones.
  - In rank degenerate cases, the ridge kills the terms in the null space.

Tikhonov Regularization

- In a simple linear learning setup, suppose

$$y = \langle x, \beta \rangle + \epsilon \ , \ \epsilon \text{ zero mean. variance } \sigma^2$$

- Given training data $\{x_i, y_i\}_{i \leq N}$ we optimize the loss

$$E(w) = \|Y - Xw\|^2 + \lambda \|w\|^2 \ , \ X = (x_i) \in \mathbb{R}^{N \times d} \ , Y = (y_i) \in \mathbb{R}^N$$

$$w^* = (X^T X + \lambda I)^{-1} X^T Y \ .$$

- The generalization error is given by

$$\mathbb{E}_{x,\epsilon} |\langle x, w^* \rangle - y|^2 = \mathbb{E}_x |\langle x, \beta - w^* \rangle|^2 + \mathbb{E}_\epsilon |\epsilon|^2$$
$$= v^T \Sigma_x v + \sigma^2 \ . \qquad v = \beta - w^*$$

Tikhonov Regularization

- By expressing $X = USV^T, \beta = U\alpha$ we have

$$w^* = X^\dagger Y = V\bar{S}(S\alpha + \epsilon) \ . \qquad \bar{s}_{kk} = \frac{1}{s_{kk}^2 + \lambda}$$

- So the generalization error becomes

$$\mathbb{E}|\langle x, w^* \rangle - y|^2 = \sum_k \frac{\lambda^2 \alpha_k^2 + s_{kk}^2 \sigma^2}{(\lambda + s_{kk}^2)^2} \ .$$

- this has an optimum at $\lambda = \frac{\sigma^2}{\mathbb{E}|x|^2}$

- Wiener filtering

Tikhonov Regularization

- Limitations?
  - Minimizing the L2 norm tends to spread out the weights. Lack of sparsity in our predictions.
  - In image applications, this tends to produce blurred estimates.
  - We can regularize using different priors that favor sparsity (e.g. Lasso).
  - In machine learning, some models work better with L1 regularization (e.g. Logistic Regression, [Ng,'04]).

Stability vs Generalization

[Bousquet, Eliseff], [Hardt, Recht, Singer]

- We can interpret generalization as a form of stability of our learning protocol.
- Expected Generalization error:

$$\epsilon_{gen} = \mathbb{E}_{S,A}[F_n(\Phi(A,S)) - F(\Phi(A,S))] \ ,$$

A : (randomized) algorithm

S : (random) sample

- Stability of a randomized algorithm:

A randomized algorithm A is ϵ-uniformly stable if for all datasets S,S' differing in at most one sample we have

$$\sup_z \mathbb{E}_A[f(\Phi(A(S)); z) - f(\Phi(A(S')); z)] \leq \epsilon \ .$$

Stability vs Generalization

Theorem [HBS'15] If algorithm A is ϵ-uniformly stable then

$$\epsilon_{gen} \leq \epsilon \ .$$

- When is Stochastic Gradient Descent uniformly stable?

- If loss $f(\cdot; z)$ is convex, has smooth gradients and is Lipschitz, then T steps of SGD with step sizes γ_t satisfies

$$\epsilon \leq C \frac{\sum_{t \leq T} \alpha_t}{n} \ .$$

If loss $f(\cdot; z)$ is strongly convex, has smooth gradients
and is Lipschitz, then scaled SGD with constant step size satisfies

$$\epsilon \leq \frac{C}{n} \ .$$

Stability vs Generalization

- Stability increases mildly with iterations
- Optimization error decreases with iterations
- Optimal tradeoffs can be studied in convex settings
- Results also extend to non-convex settings
  - Partially explain why multiple epochs over the training have better generalization.

- Stability-inducing operations/regularization
  - Ridge regression improves stability constants.
  - Dropout also improves the stability constants.
- Question: sharpness of results with respect to step size.

Dropout [Hinton'12]

- The ridge regression replaced the empirical data covariance $X^T X$ by $X^T X + \lambda I$.
  - This is equivalent as replacing data x_i by

$$\tilde{x}_i = x_i + \epsilon \ , \ \mathbb{E}\epsilon = 0, \mathrm{cov}(\epsilon) = \lambda I \ .$$

- Q: to what extent one can regularize by adding noise to the input? what noise distributions are appropriate?

Dropout [Hinton et al.'12]

- Given a deep model

$$\Phi(x;\Theta) = \phi_K(\phi_{K-1}(\dots\phi_1(X;\Theta_1);\Theta_2)\dots;\Theta_K)$$

- we consider the following noise distribution

$$\tilde{\Phi}(x;\Theta) = \phi_K(b_{K-1}\cdot\phi_{K-1}(\dots(b_1\cdot\phi_1(b_0\cdot X;\Theta_1);\Theta_2)\dots;\Theta_K)\ ,$$

$$b_0,\dots,b_{K-1} \text{ Bernoulli } p\ .$$

- At test time, we approximate $\mathbb{E}_b\tilde{\Phi}(x;\Theta)$ with $\Phi(x;p\Theta)$.
- Typically, we choose $p = 0.5$.
- Very robust, very efficient.
- Not clear why (yet).

Dropout and Ensemble Methods

- Dropout performs a form of exponential ensemble of tiny networks.
  - Let $M = \sum_{k=1}^{K} \dim(\Theta_k)$ be the total number of weights.
  - For each given training sample, on average we have pM active weights. Number of different configurations is $\sim \binom{M}{pM}$
  - At test time, we approximate the committee of these smaller networks.
  - Hinton argues that this fights feature "co-adaptation": relying on spurious, unreliable high-order dependencies within the data.

Dropout and Adaptive Regularization

- [Wager et al'13] performed the first rigorous analysis of Dropout in the context of Generalized Linear Models:

$$\text{Suppose response } y \text{ given input features } x \in \mathbb{R}^d$$

$$p(y|x,\beta) = p_0(y)\exp\left(y\langle x,\beta\rangle - A(x,\beta)\right) \ , \ \ell(\beta) = -\log p(y|x,\beta) \ .$$

$$\text{Standard MLE } \hat{\beta}\text{:} \qquad \hat{\beta} = \arg\min_{\beta} \sum_i \ell_{x_i,y_i}(\beta) \ .$$

$$\text{Noisy features: } \tilde{x_i} = \nu(x_i,\xi_i).$$

$$\text{Regularized MLE estimation:}$$

$$\hat{\beta} = \arg\min_{\beta} \sum_i \mathbb{E}_{\xi} \ell_{\nu(x_i,\xi),y_i}(\beta) \ .$$

- The latter can be rewritten as

$$\sum_i \ell_{x_i,y_i}(\beta) + R(\beta) \ , \ \text{with } R(\beta) = \sum_i \mathbb{E}_{\xi} A(\tilde{x_i},\beta) - A(x_i,\beta)$$

Dropout and Adaptive Regularization

- By doing a Taylor approximation of R, the authors show that dropout noise performs *adaptive regularization:*

$$R(\beta) \approx \beta^T \mathrm{diag}(X^T V(\beta) X)\beta \ ,$$

$$X^T V(\beta) X \ : \ \text{Fisher information}$$