

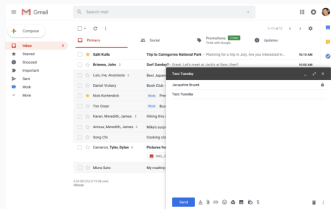
CMPT 409/981: Optimization for Machine Learning

Lecture 1

Sharan Vaswani

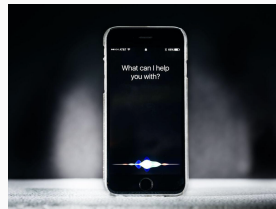
September 8, 2022

Successes of Machine Learning



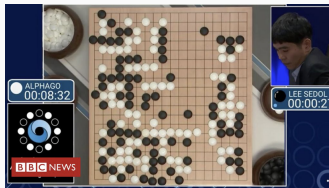
<https://www.blog.google/products/gmail/subject-write-emails-faster-smart-compose-gmail/>

(a) Natural language processing



<https://www.cnet.com/news/what-is-siri/>

(b) Speech recognition



<https://www.bbc.com/news/technology-35785875>

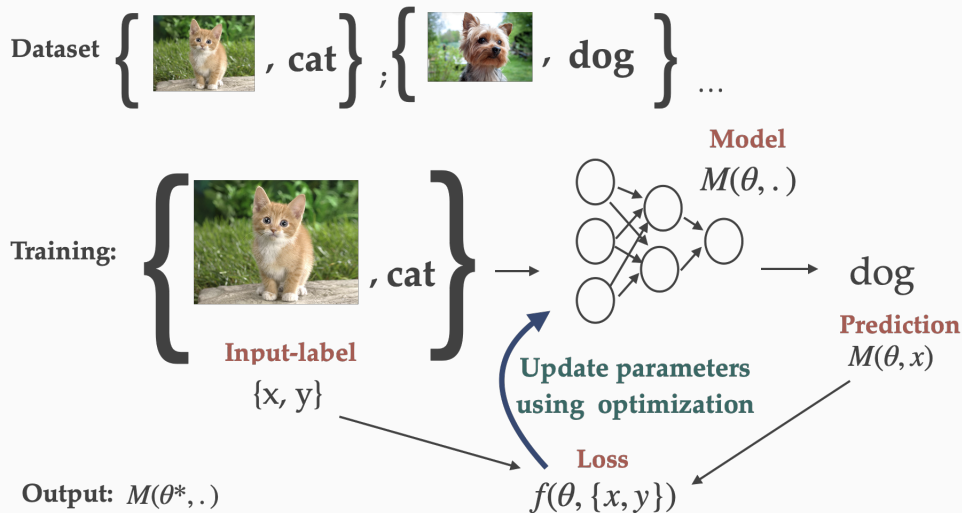
(c) Reinforcement learning



<https://www.pbs.org/newshour/science/in-a-crash-should-self-driving-cars-save-passengers-or-pedestrians-2-million-people-weigh-in>

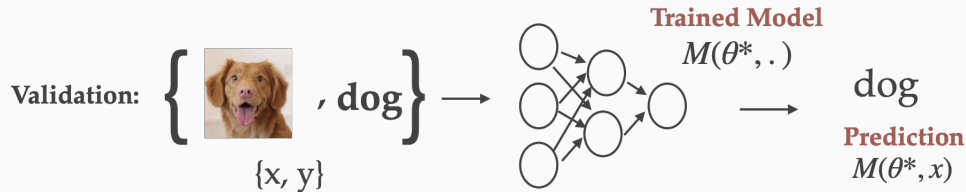
(d) Self-driving cars

Machine Learning 101



Machine Learning 101

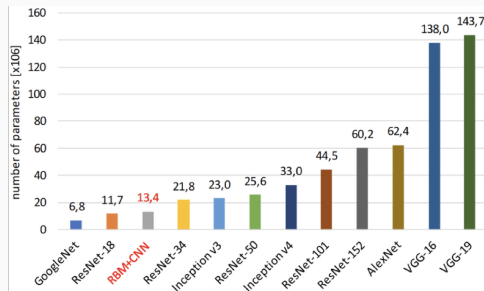
Validation Dataset: $\left\{ \text{cat image}, \text{cat} \right\}; \left\{ \text{dog image}, \text{dog} \right\} \dots$



Output: Validation Accuracy

Measures how good the trained model is

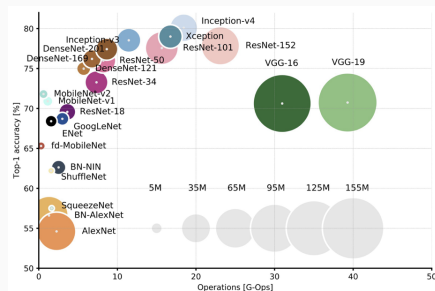
Modern Machine Learning



Sobczak, Szymon, et al. "Restricted Boltzmann machine as an aggregation technique for binary descriptors.", 2019.

Model size

(a)



Canziani et al, "An Analysis of Deep Neural Network Models for Practical Applications", 2016.

Number of operations for computing the loss

(b)

Figure 1: Models for multi-class classification on Image-Net. Number of examples = 1.2 M

Faster optimization methods can have a big practical impact!

- **(Non)-Convex minimization:** Supervised learning (classification/regression), Matrix factorization for recommender systems, Image denoising.
- **Online optimization:** Learning how to play Go/Atari games, Imitating an expert and learning from demonstrations, Regulating control systems like industrial plants.
- **Min-Max optimization:** Generative Adversarial Networks, Adversarial Learning, Multi-agent RL.

Objective: Introduce foundational optimization concepts with applications to machine learning.

Syllabus:

- **(Non)-Convex minimization:** Gradient Descent, Momentum/Acceleration, Mirror Descent, Newton/Quasi-Newton methods, Stochastic gradient descent (SGD), Variance reduction
- **Online optimization:** Follow the (regularized) leader, Adaptive methods (AdaGrad, Adam)
- **Min-Max optimization:** (Stochastic) Gradient Descent-Ascent, (Stochastic) Extragradient

What we won't get time to cover in detail: Non-smooth optimization, Convex analysis, Global optimization.

What we won't get time to cover: Constrained optimization, Distributed optimization, Multi-objective optimization.

- **Instructor:** Sharan Vaswani (TASC-1 8221) Email: sharan_vaswani@sfu.ca
- **Office Hours:** Monday 4 pm - 5 pm (TASC-1 8221), TBD
- **Teaching Assistant:** Zahra MiriKharaji Email: zmirikha@sfu.ca
- **Course Webpage:** https://vaswanis.github.io/409_981-F22.html
- **Piazza:** <https://piazza.com/sfu.ca/fall2022/cmpt409981/home>
- **Prerequisites:** Linear Algebra, Multivariable calculus, (Undergraduate) Machine Learning

Assignments $[4 \times 12.5\% = 50\%]$

- Assignments to be submitted online, typed up in Latex with accompanying code submitted as a zip file.
- Each assignment will be due in 10 days (at 11.59 pm PST).
- For some flexibility, each student is allowed 1 late-submission and can submit in the next class (no late submissions beyond that).
- If you use up your late-submission and submit late again, you will lose 50% of the mark.

Final Project [50%]

- Aim is to give you a taste of research in Optimization.
- Projects to be done in groups of 3-4 (more details will be on Piazza)
- Will maintain a list on Piazza on possible project topics. You are free to choose from the list or propose a topic that combines Optimization with your own research area.
- Project Proposal [10%] – Discussion (before 20 October) + Report (due 24 October)
- Project Milestone [5%] – Update (before 20 November)
- Project Presentation [10%] (6 December)
- Project Report [25%] (15 December)

Questions?

Minimizing functions

Consider minimizing a function over a unit hyper-cube:

$$\min_{w \in [0,1]^d} f(w).$$

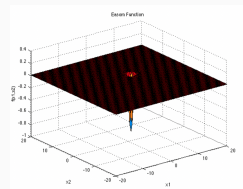
Objective: For a target accuracy of $\epsilon > 0$, if w^* is the minimizer of f , what is the the number of oracle calls required to return a point \hat{w} s.t. $f(\hat{w}) - f(w^*) \leq \epsilon$?

Setting: Have access to a **zero-order oracle** – querying the oracle at $w \in \mathcal{D}$ returns $f(w)$.

Example 1: Minimize a one-dimensional function s.t. $f(w) = 0$ for all $x \neq w^*$, and $f(w^*) = -\epsilon$.

Example 2: Easom function:

$$f(x_1, x_2) = -\cos(x_1) - \cos(x_2) \exp(-(x_1 - \pi)^2 - (x_2 - \pi)^2).$$



Minimizing generic functions is hard! We need to make assumptions on the structure.

Lipschitz continuous functions

Consider minimizing a function over a unit hyper-cube:

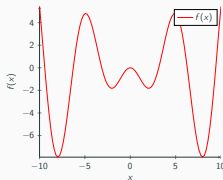
$$\min_{w \in [0,1]^d} f(w).$$

Assumption: f is *Lipschitz continuous* meaning that f can not change arbitrarily fast as x changes. Formally, for any $x, y \in [0,1]^d$,

$$|f(x) - f(y)| \leq G \|x - y\|$$

where G is the Lipschitz constant.

Example: $f(x) := -x \sin(x)$ in the $x \in [-10, 10]$ interval.



Lipschitz continuity of the function immediately implies that the gradients are *bounded* i.e. for all $x \in [0,1]^d$, $\|\nabla f(x)\| \leq G$.

Global Minimization

Consider minimizing a G -Lipschitz continuous function over a unit hyper-cube:

$$\min_{w \in [0,1]^d} f(w).$$

Access to a **zero-order oracle** – we can query the oracle at any point w and it returns $f(w)$.

Objective: For a target accuracy of $\epsilon > 0$, if w^* is the minimizer of f , what is the the number of oracle calls required to return a point \hat{w} s.t. $f(\hat{w}) - f(w^*) \leq \epsilon$?

Naive algorithm: Divide the hyper-cube into cubes with length of each side equal to $\epsilon' > 0$ (to be determined). Call the zero-order oracle on the centers of these $\frac{1}{\epsilon'^d}$ cubes and return the point \hat{w} with the minimum function value.

Analysis: The minimizer lies in/at the boundary of one of these cubes, and hence by returning the minimum \hat{w} , we guarantee that \hat{w} is at most $\sqrt{d}\epsilon'$ -away from w^* i.e. $\|\hat{w} - w^*\| \leq \sqrt{d}\epsilon'$. By G -Lipschitz continuity, $f(\hat{w}) - f(w^*) \leq G \|\hat{w} - w^*\| \leq G\sqrt{d}\epsilon'$. For a target accuracy of ϵ , we can set $\epsilon' = \frac{\epsilon}{G\sqrt{d}}$. Hence, for this naive algorithm, total number of oracle calls = $\left(\frac{G\sqrt{d}}{\epsilon}\right)^d$.

Consider minimizing a differentiable, G -Lipschitz continuous function over a unit hyper-cube:

$$\min_{w \in [0,1]^d} f(w).$$

Q: Suppose we do a random search over the cubes? What is the expected number of function evaluations?

Is our naive algorithm good? Can we do better?

Lower-Bound: For minimizing a continuous, G -Lipschitz continuous function over a unit hyper-cube, any algorithm requires $\Omega\left(\left(\frac{G}{\epsilon}\right)^d\right)$ calls to the zero-order oracle.

Our naive-algorithm is *sub-optimal* by a factor of $O\left((\sqrt{d})^d\right)$.

Questions?

Smooth functions

Recall that Lipschitz continuous functions have bounded gradients i.e. $\|\nabla f(w)\| \leq G$ and can still include *non-smooth* (not differentiable everywhere) functions.

For example, $f(x) = |x|$ is 1-Lipschitz continuous but not differentiable at $x = 0$ and the gradient changes from -1 at 0^- to $+1$ at 0^+ .

An alternative assumption we can make is that f is *smooth* – it is differentiable everywhere and its gradient is Lipschitz-continuous i.e. it can not change arbitrarily fast.

Formally, the gradient ∇f is L -Lipschitz continuous if for all x, y in the domain \mathcal{D} ,

$$\|\nabla f(x) - \nabla f(y)\| \leq L \|x - y\|$$

where L is the Lipschitz constant of the gradient.

Q: Does Lipschitz-continuity of the gradient imply Lipschitz-continuity of the function?

Smooth functions – Examples

If f is twice-differentiable and smooth, then for all $x \in \mathcal{D}$, $\nabla^2 f(x) \preceq L I_d$ i.e. $\sigma_{\max}[\nabla^2 f(x)] \leq L$ where σ_{\max} is the maximum singular value.

Q: Does $f(x) = x^3$ have a Lipschitz-continuous gradient over \mathbb{R} ?

Q: Does $f(x) = x^3$ have a Lipschitz-continuous gradient over $[0, 1]$?

Q: The *negative entropy function* is given by $f(x) = x \log(x)$. Does it have a Lipschitz-continuous gradient over $[0, 1]$?

Smooth functions – Examples

Linear Regression on n points with d features. Feature matrix: $X \in \mathbb{R}^{n \times d}$, vector of measurements: $y \in \mathbb{R}^n$ and parameters $w \in \mathbb{R}^d$.

$$\min_{w \in \mathbb{R}^d} f(w) := \frac{1}{2} \|Xw - y\|^2$$

$$f(w) = \frac{1}{2} [w^\top (X^\top X) w - 2w^\top X^\top y + y^\top y] ; \nabla f(w) = X^\top X w - X^\top y ; \nabla^2 f(w) = X^\top X$$

If f is L -smooth, then, $\sigma_{\max}[\nabla^2 f(w)] \leq L$ for all w . Hence, for linear regression $L = \lambda_{\max}[X^\top X]$.

Q: Is the linear regression loss-function Lipschitz continuous?

Q: Compute L for *ridge regression* – ℓ_2 -regularized linear regression where $f(w) := \frac{1}{2} \|Xw - y\|^2 + \frac{\lambda}{2} \|w\|^2$.

Smooth functions

Claim: For an L -smooth function, $f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{L}{2} \|y - x\|^2$ for all $x, y \in \mathcal{D}$.

Proof:

$$f(y) = f(x) + \int_{t=0}^1 [\nabla f(x + t(y - x))] (y - x)^\top dt \quad (\text{Fundamental theorem of calculus})$$

$$= f(x) + \langle \nabla f(x), y - x \rangle + \int_{t=0}^1 [\nabla f(x + t(y - x))] (y - x)^\top dt - [\nabla f(x)] (y - x)^\top$$

$$= f(x) + \langle \nabla f(x), y - x \rangle + \int_{t=0}^1 [\nabla f(x + t(y - x)) - \nabla f(x)] (y - x)^\top dt$$

$$\leq f(x) + \langle \nabla f(x), y - x \rangle + \int_{t=0}^1 \|\nabla f(x + t(y - x)) - \nabla f(x)\| \|y - x\| dt$$

(Cauchy-Schwarz)

$$\leq f(x) + \langle \nabla f(x), y - x \rangle + L \int_{t=0}^1 \|x + t(y - x) - x\| \|y - x\| dt \quad (\text{Lipschitz continuity})$$

$$= f(x) + \langle \nabla f(x), y - x \rangle + L \|y - x\|^2 \int_{t=0}^1 t dt = f(x) + \langle \nabla f(x), y - x \rangle + \frac{L}{2} \|y - x\|^2$$

The inequality $f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{L}{2} \|y - x\|^2$ can be interpreted as a *global* quadratic upper-bound on f at point x i.e. the bound holds for all $y \in \mathcal{D}$.

There are other related (not necessarily equivalent) ways to state the L -smoothness of f (you will need to prove these in Assignment 1).

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle + \frac{1}{2L} \|\nabla f(y) - \nabla f(x)\|^2$$
$$\langle \nabla f(x) - \nabla f(y), x - y \rangle \leq L \|x - y\|^2$$

Questions?

Local Minimization

Even though f is smooth, it still includes functions with multiple local/global minimum and stationary points. Eg: $f(x) = -x \sin(x)$.

Consider minimizing a smooth function over \mathbb{R}^d

$$\min_{w \in \mathbb{R}^d} f(w).$$

Since we have seen that global minimization can be impossible (without Lipschitz assumption on f) or the number of oracle calls can be exponential in d , let us aim to solve an easier problem.

Access to a **first-order oracle** – query the oracle at point w and it returns $f(w)$ and $\nabla f(w)$.

Objective: For a target accuracy of $\epsilon > 0$, what is the the number of oracle calls required to return a point \hat{w} s.t. $\|\nabla f(\hat{w})\|^2 \leq \epsilon$?

We only care about making the gradient small and finding an approximate stationary point.

Local Minimization

Recall that L -smoothness of f implies that $f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{L}{2} \|y - x\|^2$.

Idea: Since the RHS is a global upper-bound on the true function, instead of minimizing the function directly, let us minimize the upper-bound at x w.r.t y .

Setting the gradient of the RHS w.r.t y to zero, we obtain \hat{y} as:

$$\nabla f(x) + L [\hat{y} - x] = 0 \implies \hat{y} = x - \frac{1}{L} \nabla f(x)$$

This is exactly the gradient descent update at x !

We can do this iteratively i.e. starting at x_0 , form the upper-bound at x_0 , minimize it by setting $x_1 = x_0 - \frac{1}{L} \nabla f(x_0)$, then form the quadratic upper-bound at x_1 and repeat. Continue to do this until we find a point \hat{x} s.t. $\|\nabla f(x)\|^2 \leq \epsilon$ and terminate.

This is exactly the gradient descent procedure – move in the direction of the negative gradient (“downhill”) with a *step-size* equal to η . Formally, at iteration k ,

$$w_{k+1} = w_k - \eta \nabla f(w_k).$$

Gradient Descent

But does this algorithm terminate and return \hat{w} such that $\|\nabla f(w)\|^2 \leq \epsilon$?

Claim: For L -smooth functions, gradient descent with $\eta = \frac{1}{L}$ returns a point \hat{w} such that $\|\nabla f(w)\|^2 \leq \epsilon$ and requires $T = \frac{2L[f(w_0) - \min_w f(w)]}{\epsilon}$ oracle calls or iterations.

Proof:

At iteration k , the algorithm calls the oracle to compute $\nabla f(w_k)$ and does a gradient descent update: $w_{k+1} = w_k - \frac{1}{L} \nabla f(w_k)$. Using the L -smoothness of f ,

$$\begin{aligned} f(w_{k+1}) &\leq f(w_k) + \langle \nabla f(w_k), -\frac{1}{L} \nabla f(w_k) \rangle + \frac{L}{2} \left\| \frac{1}{L} \nabla f(w_k) \right\|^2 \\ &\quad \text{(Substitute } x = w_k \text{ and } y = w_k - \frac{1}{L} \nabla f(w_k) \text{ in the quadratic bound)} \\ \implies f(w_{k+1}) &\leq f(w_k) - \frac{1}{2L} \|\nabla f(w_k)\|^2 \end{aligned}$$

By moving from w_k to w_{k+1} , we have decreased the value of f since $f(w_{k+1}) \leq f(w_k)$.

Gradient Descent

Rearranging the inequality from the previous slide, for every iteration k ,

$$\frac{1}{2L} \|\nabla f(w_k)\|^2 \leq f(w_k) - f(w_{k+1})$$

Adding up $k = 0$ to $T - 1$,

$$\begin{aligned} \frac{1}{2L} \sum_{k=0}^{T-1} \|\nabla f(w_k)\|^2 &\leq \sum_{k=0}^{T-1} [f(w_k) - f(w_{k+1})] = f(w_0) - f(w_T) \leq [f(w_0) - \min_w f(w)] \\ \implies \frac{\sum_{k=0}^{T-1} \|\nabla f(w_k)\|^2}{T} &\leq \frac{2L [f(w_0) - \min_w f(w)]}{T} \end{aligned}$$

The LHS is the average of the gradient norms over the T iterates. Let

$\hat{w} := \arg \min_{k \in \{0, 1, \dots, T-1\}} \|\nabla f(w_k)\|^2$. Since the minimum is smaller than the average,

$$\|\nabla f(\hat{w})\|^2 \leq \frac{2L [f(w_0) - \min_w f(w)]}{T}$$

Since, $\|\nabla f(\hat{w})\|^2 \leq \frac{2L[f(w_0) - \min_w f(w)]}{T}$, the *rate of convergence* is $O(1/T)$.

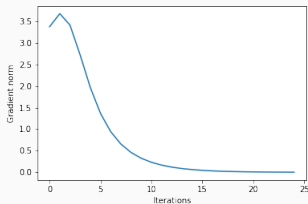
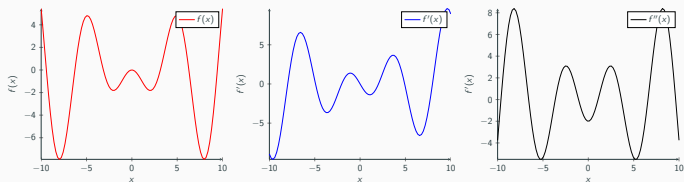
Since $\|\nabla f(\hat{w})\|^2 \leq \frac{2L[f(w_0) - \min_w f(w)]}{T}$ for all T , if the RHS $\frac{2L[f(w_0) - \min_w f(w)]}{T} \leq \epsilon$, this would guarantee that $\|\nabla f(\hat{w})\|^2 \leq \epsilon$ and we would achieve our objective.

Hence, we need to run the algorithm for $T \geq \frac{2L[f(w_0) - \min_w f(w)]}{\epsilon}$ iterations.

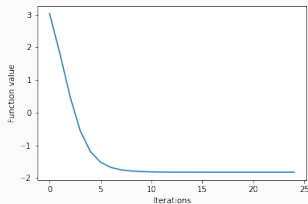
Lower-Bound: When minimizing a smooth function (without additional assumptions), any algorithm requires $\Omega\left(\frac{1}{\epsilon}\right)$ oracle calls to return a point \hat{w} such that $\|\nabla f(\hat{w})\|^2 \leq \epsilon$. Hence, gradient descent is optimal for minimizing smooth functions!

Gradient Descent – Example

$\min_{x \in [-10, 10]} f(x) := -x \sin(x)$. Run GD with $\eta = 1/L \approx 0.1$ and $x_0 = 4$.



(a) Gradient norm



(b) Function value

Questions?

We have seen that we can reach a stationary point of a smooth function in $O\left(\frac{1}{\epsilon}\right)$ iterations of GD with a step-size $\eta = \frac{1}{L}$.

Problems with this approach:

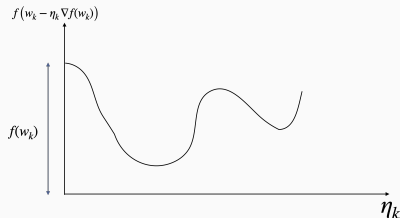
- Computing L in closed-form can be difficult as the functions get complicated.
- Theoretically computed L is global (the “local” L might be much smaller) and often loose in practice (typically we tend to overestimate L resulting in a smaller step-size).

Gradient Descent with Line-search

Instead of setting η according to L , we can “search” for a good step-size η_k in each iteration k .

Exact line-search: At iteration k , solve the following sub-problem:

$$\eta_k = \arg \min_{\eta} f(w_k - \eta \nabla f(w_k)).$$



After computing η_k , do the usual GD update: $w_{k+1} = w_k - \eta_k \nabla f(w_k)$.

- (i) Can solve the sub-problem approximately by doing gradient descent w.r.t η (expensive),
- (ii) Compute η_k analytically (can be done only in special cases).

Gradient Descent with Line-search – Example

Recall linear regression: $\min_{w \in \mathbb{R}^d} f(w) := \frac{1}{2} \|Xw - y\|^2 = \frac{1}{2} [w^\top (X^\top X)w - 2w^\top X^\top y + y^\top y]$.

For the exact line-search, we need to $\min_{\eta} h(\eta) := f(w_k - \eta \nabla f(w_k))$.

Since f is a quadratic, we can directly use the second-order Taylor series expansion.

$$h(\eta) = f(w_k - \eta \nabla f(w_k)) = f(w_k) + \langle \nabla f(w_k), -\eta \nabla f(w_k) \rangle + \frac{1}{2} [-\eta \nabla f(w_k)]^\top \nabla^2 f(w_k) [-\eta \nabla f(w_k)]$$

$$\nabla h(\eta) = -\|\nabla f(w_k)\|^2 + \eta [\nabla f(w_k)]^\top \nabla^2 f(w_k) [\nabla f(w_k)] = 0 \implies \eta = \frac{\|\nabla f(w_k)\|^2}{\|\nabla f(w_k)\|_{\nabla^2 f(w_k)}^2}$$

For linear regression, $\nabla^2 f(w_k) = X^\top X$ and $\nabla f(w_k) = X^\top (Xw_k - y)$. With exact line-search, the GD update for linear regression is:

$$w_{k+1} = w_k - \frac{\|X^\top (Xw_k - y)\|^2}{\|X^\top (Xw_k - y)\|_{X^\top X}^2} [X^\top (Xw_k - y)]$$

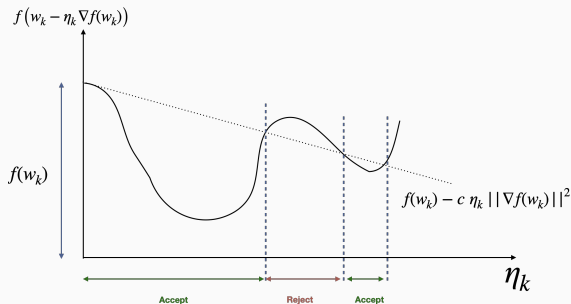
Gradient Descent with Line-search

Usually, the cost of doing an exact line-search is not worth the computational effort.

Armijo condition for a prospective step-size $\tilde{\eta}_k$ as:

$$f(w_k - \tilde{\eta}_k \nabla f(w_k)) \leq f(w_k) - c \tilde{\eta}_k \|\nabla f(w_k)\|^2$$

where $c \in (0, 1)$ is a hyper-parameter.



Backtracking line-search: At iteration k , starting with an initial “guess” of the step-size η_{\max} , check the Armijo condition for a step-sizes $\tilde{\eta}_k$.

- If $\tilde{\eta}_k$ satisfies that the Armijo condition, set $\eta_k = \tilde{\eta}_k$ and do the usual GD update.
- Else, decrease $\tilde{\eta}_k$ by a multiplicative factor $\beta \in (0, 1)$ and check the Armijo condition again for a new prospective step-size $\tilde{\eta}_k \beta$.

Keep “backtracking” on $\tilde{\eta}_k$ until the Armijo condition is satisfied.

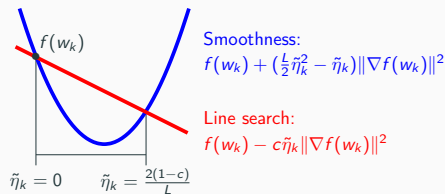
Gradient Descent with Line-search

Claim: The (exact) backtracking procedure terminates and returns $\eta_k \geq \min \left\{ \frac{2(1-c)}{L}, \eta_{\max} \right\}$.

Proof:

$$f(w_k - \tilde{\eta}_k \nabla f(w_k)) \leq f(w_k) - \|\nabla f(w_k)\|^2 \left(\eta_k - \frac{L\tilde{\eta}_k^2}{2} \right) \quad (\text{Quadratic bound using smoothness})$$

$$f(w_k - \tilde{\eta}_k \nabla f(w_k)) \leq f(w_k) - \|\nabla f(w_k)\|^2 (c\tilde{\eta}_k) \quad (\text{Armijo condition})$$



(i) If $c\tilde{\eta}_k \geq \tilde{\eta}_k - \frac{L\tilde{\eta}_k^2}{2}$, then the Armijo condition is satisfied, and the back-tracking procedure terminates $\implies \tilde{\eta}_k \geq \frac{2(1-c)}{L}$. (ii) If $\eta_{\max} \leq \frac{2(1-c)}{L}$, the line-search terminates in the first iteration.

Gradient Descent with Line-search

Claim: Gradient Descent with (exact) backtracking Armijo line-search (with $c = 1/2$) returns point \hat{w} such that $\|\nabla f(w)\|^2 \leq \epsilon$ and requires $T = \frac{2L[f(w_0) - \min_w f(w)]}{\epsilon}$ oracle calls or iterations.

Proof: Since η_k satisfies the Armijo condition and $w_{k+1} = w_k - \eta_k \nabla f(w_k)$,

$$\begin{aligned} f(w_{k+1}) &\leq f(w_k) - c \eta_k \|\nabla f(w_k)\|^2 \\ &\leq f(w_k) - \left(\min \left\{ \frac{1}{2L}, \eta_{\max} \right\} \right) \|\nabla f(w_k)\|^2 \\ &\quad \text{(Result from previous slide with } c = 1/2) \end{aligned}$$

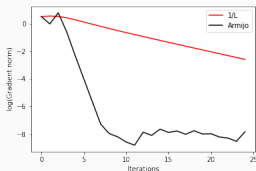
Continuing the proof similar to Slide 22,

$$\implies \|\nabla f(\hat{w})\|^2 \leq \frac{\max\{2L, 1/\eta_{\max}\} [f(w_0) - \min_w f(w)]}{T}$$

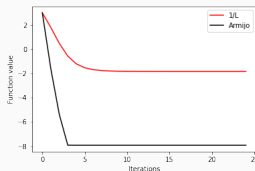
The claim is proved by reasoning similar to Slide 22.

Gradient Descent with Line-search – Examples

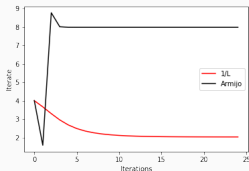
$\min_{x \in [-10, 10]} f(x) := -x \sin(x)$. Compare GD (with $x_0 = 4$) with (i) $\eta = 1/L \approx 0.1$ and (ii) Armijo line-search with $\eta_{\max} = 10, c = 1/2, \beta = 0.9$.



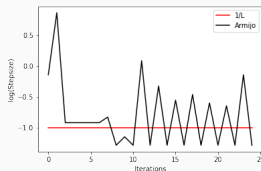
(a) Gradient norm



(b) Function value



(c) Iterate



(d) Stepsize

Questions?