CSE 446
Gradient Descent Theoretical Analysis

Natasha Jaques

UNIVERSITY of WASHINGTON

Lecture plan

- Gradient descent algorithm + examples

- **Theoretical analysis**
  - When does it work?
    - Key idea: Convexity
  - **How quickly does it converge?** ← **we are (finally) here**
  - **How do we choose a step size?**

Convergence analysis steps

1. Study single iteration: $f(w_{t+1})$ vs. $f(w_t)$

   # Want to show that loss goes down each step: $f(w_{t+1}) <= f(w_t)$

2. Piece iterations together to study how they converge

   # After enough iterations, do we converge to w*? Or at least $\left\| \nabla_w f(w) \right\| \approx 0$

1. Single-iteration progress bound

Don't need convexity (yet).

Assume:

- f is C^2 # Twice differentiable
- gradient of f is **Lipschitz continuous**:

  There exists L such that:

  For all u, v, $\|\nabla f(u) - \nabla f(v)\| \leq L\|u - v\|$. # L is how fast your gradient can change

  For all w, $\nabla^2 f(w) \preccurlyeq LI$. # The slope is bounded everywhere

  For any $u,\ v$, $v^T \nabla^2 f(u) v \leq L\|v\|^2$. # Smooth.

  # "L-Smooth"

1. Single-iteration progress bound

Don't need convexity (yet).

Assume:

- f is C^2 # Twice differentiable
- gradient of f is **Lipschitz continuous**:

  There exists L such that:

  For all u, v, $\|\nabla f(u) - \nabla f(v)\| \leq L\|u - v\|$. # L is how fast your gradient can change

  For all w, $\nabla^2 f(w) \preccurlyeq LI$. # The slope is bounded everywhere

  For any $u,\ v$, $v^T \nabla^2 f(u) v \leq L\|v\|^2$. # Smooth.

  # "L-Smooth"

Spoiler alert: how would knowing L for your function help us guarantee convergence of gradient descent?

- When does gradient descent diverge?
- **Use L to pick step size!**
  - Small L -> larger η
  - Large L -> smaller η

Single-iteration progress bound

Want to show that loss goes down each step: $f(w_{t+1}) <= f(w_t)$

For any $u,\ v,$ $\quad v^T \nabla^2 f(u) v \leq L\|v\|^2.$ # By Lipschitz assumption

What quantities do we even have in gradient descent? $w_t,\ f(w_t),\ \nabla f(w_t),\ f(w_{t+1})$

Take Taylor expansion:

$$f(w_{t+1}) = f(w_t) + (w_{t+1} - w_t)^T \nabla f(w_t) + \frac{1}{2}(w_{t+1} - w_t)^T \nabla^2 f(u)(w_{t+1} - w_t)$$

where $u \in [w_{t+1}, w_t]$

$u \in \alpha w_t + (1-\alpha) w_{t+1}$

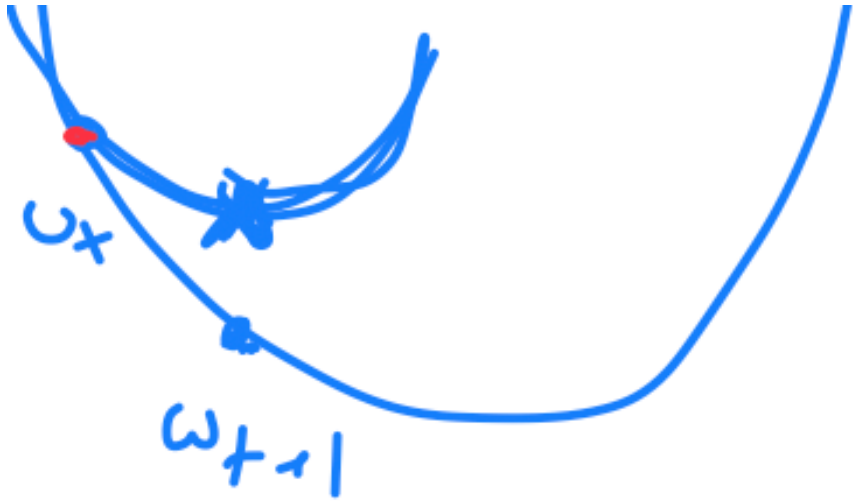
But by the Lipschitz assumption above, what can we do to the second derivative term?

$$f(w_{t+1}) \leq f(w_t) + (w_{t+1} - w_t)^T \nabla f(w_t) + \frac{L}{2}\|w_{t+1} - w_t\|_2^2$$

Can I get a visual?

$$f(w_{t+1}) \leq f(w_t) + (w_{t+1} - w_t)^T \nabla f(w_t) + \frac{L}{2}\|w_{t+1} - w_t\|_2^2$$

The 2nd order Taylor expansion around w_t + Lipschitz assumption gives us a quadratic upper bound on the function



If I want to pick w_{t+1} to make the most progress in minimizing my function, what should I do?

Picking η to guarantee convergence

$$f(w_{t+1}) \leq f(w_t) + (w_{t+1} - w_t)^T \nabla f(w_t) + \frac{L}{2}\|w_{t+1} - w_t\|_2^2$$

Let $\Delta = w_{t+1} - w_t$ # Recall gradient descent: $w_{t+1} = w_t - \eta \nabla f(w_t)$

$$f(w_{t+1}) \leq f(w_t) + \nabla f(w_t)^T \Delta + \frac{L}{2}\|\Delta\|_2^2$$

Goal: pick Δ to minimize $\nabla f(w_t)^T \Delta + \frac{L}{2}\|\Delta\|_2^2$

Why? Make the most progress in minimizing loss, while still guaranteeing $f(w_{t+1}) <= f(w_t)$

How can I minimize something? **Take the derivative, set it to zero!**

Picking η to guarantee convergence

$$f(w_{t+1}) \leq f(w_t) + (w_{t+1} - w_t)^T \nabla f(w_t) + \frac{L}{2}\|w_{t+1} - w_t\|_2^2$$

Let $\Delta = w_{t+1} - w_t$ # Recall gradient descent: $w_{t+1} = w_t - \eta \nabla f(w_t)$

$$\rightarrow \Delta = -\eta \nabla f(w_t)$$

$$f(w_{t+1}) \leq f(w_t) + \nabla f(w_t)^T \Delta + \frac{L}{2}\|\Delta\|_2^2$$

Goal: pick Δ to minimize $\nabla f(w_t)^T \Delta + \frac{L}{2}\|\Delta\|_2^2$ **Take the derivative, set it to zero!**

$$\nabla f(w_t) + L\Delta = 0$$

$$\Delta = \frac{-1}{L}\nabla f(w_t) \quad = -\eta \nabla f(w_t) \rightarrow \boxed{\eta = \frac{1}{L}}$$

$$w_{t+1} = w_t - \frac{1}{L}\nabla f(w_t)$$

Single-iteration progress bound

Still trying to show that loss goes down each step: $f(w_{t+1}) <= f(w_t)$

$$w_{t+1} = w_t - \frac{1}{\mathrm{L}} \nabla f(w_t)$$

Now we know how to pick the right η = 1/L

$$f(w_{t+1}) \leq f(w_t) + \nabla f(w_t)^\top (w_{t+1} - w_t) + \frac{L}{2} \|w_{t+1} - w_t\|_2^2$$

Bound from before

$$\leq f(w_t) - \eta \|\nabla f(w_t)\|_2^2 + \frac{L}{2} \eta^2 \|\nabla f(w_t)\|_2^2$$

By $\Delta = -\eta \nabla f(w_t)$

$$f(w_{t+1}) \leq f(w_t) - \frac{1}{2L} \|\nabla f(w_t)\|_2^2$$

Final one step progress bound!

Amount of progress guaranteed every step

So when do we make more progress?

- Large gradients
- Smooth function (small L)

Single-iteration progress bound

When $\eta = \frac{1}{L}$,

$$f(w_{t+1}) \leq f(w_t) - \frac{1}{2L}\left\| \nabla f(w_t) \right\|_2^2$$

Same argument shows any $\eta < \frac{2}{L}$ will decrease f.

For any η in 0 < η < 2/L, you will still make progress

Convergence analysis steps

1. Study single iteration: $f(w_{t+1})$ vs. $f(w_t)$
2. Piece iterations together to study how they converge

2. Convergence rate of gradient descent

First, we will show you converge to a 0 gradient

But this could be a local minima or saddle point

But if you assume convexity…

$$\left\| \nabla f(w_t) \right\| \to 0 \longrightarrow f(w_t) - f(w^*) \to 0$$

Then the point you converge to is the global minimum

Convergence rate of gradient descent

For some ϵ, how many iterations before $\left\| \nabla f(w_t) \right\|^2 \leq \epsilon$?

Assumptions:

- Gradient is Lipschitz continuous (as before)
- Step size is small enough (assume $1/L$) $\eta = \frac{1}{L}$
- f is bounded below by $f(w^*)$ $f(w*) \geq c$

Why is that needed / realistic?

e.g. MSE. Otherwise ∞ steps

Proof sketch:

- Each iteration decreases f by at least $\frac{1}{2L}\left\| \nabla f(w_t) \right\|^2$ # Already showed
- Can't decrease below $f(w^*)$ # By assumption
- So $\left\| \nabla f(w_t) \right\|^2$ must be decaying fast enough

Convergence rate of gradient descent

$$f(w_{t+1}) \le f(w_t) - \frac{1}{2L}\left\|\nabla f(w_t)\right\|_2^2$$

Already showed

$$\left\|\nabla f(w_t)\right\|_2^2 \le 2L(f(w_t) - f(w_{t+1}))$$

$$\sum_{t=0}^{T}\left\|\nabla f(w_t)\right\|_2^2 \le 2L\sum_{t=0}^{T}[f(w_t) - f(w_{t+1})]$$

What happens over T iterations?

Telescoping sum

Deal with LHS later

$$= 2L[f(w_0) - \cancel{f(w_1) + f(w_1)} - \cancel{f(w_2) - f(w_2)} \cdots + f(w_T)]$$

$$= 2L[f(w_0) - f(w_T)]$$

If you have convexity…

$$\le 2L[f(w_0) - f(w^*)]$$

How many steps this will take depends on **initial gap** (choice of w_0)

Convergence rate of gradient descent

Deal with LHS later

$$\sum_{t=0}^{T} \left\| \nabla f(w_t) \right\|_2^2 \leq 2L[f(w_0) - f(w^*)]$$

$$T \cdot \min_{0 \leq t \leq T} \left\| \nabla f(w_t) \right\|_2^2 \leq 2L[f(w_0) - f(w^*)]$$

$$\min_t \left\| \nabla f(w_t) \right\|_2^2 \leq \frac{2L}{T}[f(w_0) - f(w^*)] \quad \leq \varepsilon$$

How many steps T do I need for min gradient to reach epsilon size?

Convergence rate of gradient descent

Large L → function not smooth → more gradient steps required

$$T \geq \frac{2L\left(f(w_0) - f(w^*)\right)}{\epsilon}$$

Large initial gap → more gradient steps required

Closer to f(w*) → more gradient steps required

Gradient descent requires

$T = O(1/\epsilon)$ iterations

to achieve $\left\| \nabla f(w_t) \right\|^2 \leq \epsilon$

Big Oh: $f(n) = O(g(n)) \iff \exists C, n_0 > 0$ s.t. $\forall n \geq n_0$: $|f(n)| \leq C\,|g(n)|$

Convergence analysis steps

1. Study single iteration: $f(w_{t+1})$ vs. $f(w_t)$
2. Piece iterations together to study how they converge

We have now, finally, **proven that gradient descent converges to a gradient of ~0 in a finite number of steps**, given the assumptions

Convergence rate of gradient descent

But if you assume convexity…

$$\left\| \nabla f(w_t) \right\| \leq \epsilon \longrightarrow f(w_t) - f(w^*) \leq \epsilon$$

Gradient descent converges in $T_\varepsilon = O(\frac{1}{\varepsilon})$ iterations

Gradient descent still converges in $T_\varepsilon = O(\frac{1}{\varepsilon})$ iterations… to?

The global minimum!

Stochastic gradient descent (SGD)

$$\hat{w} = \operatorname*{argmin}_w \frac{1}{n} \sum_{i=1}^{n} \ell_i(w)$$

Gradient descent: $w_{t+1} = w_t - \eta \nabla_w \left(\frac{1}{n} \sum_{i=1}^{n} \ell_i(w) \right) \Big|_{w=w_t}$ # Entire training dataset

Stochastic gradient descent: $w_{t+1} = w_t - \eta \nabla_w \ell_{I_t}(w) \big|_{w=w_t}$

I_t drawn uniformly at random from $\{1, \ldots, n\}$

n times faster per iteration!

And can even be better minimizer.

If you were actually doing 1 sample at a time, but we usually use a batch…

Minibatch stochastic gradient descent

- Instead of one iterate, average B stochastic gradients together

  # Sample a batch of data $D_B = \{x_i \sim D : i = 1, \ldots, B\}, \quad B \ll N$

- Advantages:
  - Smaller variance (by $1/B$)
  - Parallelization: Each gradient in the minibatch can be computed in parallel

- This is very widely used! # All of deep learning

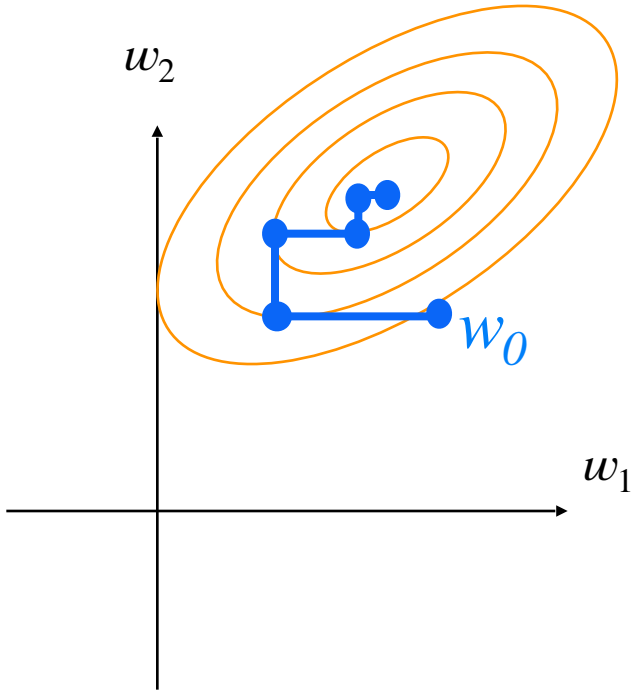
Summary

- Closed form -> iterative methods
- (Minibatch stochastic) gradient descent as a general-purpose optimizer
- Key theoretical tool: Convexity
- Many many variants. Highly active research area!
  - Schedulers
  - Adaptive step sizes
  - Momentum
  - Higher-order methods
  - Non-convex analysis
  - …

Bonus: Coordinate descent

Optimize 1 weight (coordinate) at a time

Each time you only have to solve a 1d optimization problem



Given w_2 = C, how to pick w_1 to minimize loss

Can eventually reach w* if loss is convex and differentiable

Coordinate descent for lasso

$$\hat{w} = \underset{w}{\operatorname{argmin}} \ \frac{1}{2}\sum_{i=1}^{n}\left(y_i - x_i^\top w\right)^2 + \lambda\|w\|_1$$

$$\frac{d}{dw_{k_n}} f(w)$$
$$= \sum_{i=1}\left(x_i^\top w - y_i\right)x_{ik} + \lambda\operatorname{sign}\left(w_k\right)$$
$$= \sum_{i=1}^{n}\left(\sum_{j\neq k} x_{ij}w_j + x_{ik}w_k - y_i\right)x_{ik} + \lambda\operatorname{sign}\left(w_k\right)$$
$$= \sum_{i=1}^{n}\left(\sum_{j\neq k} x_{ij}w_j - y_i\right)x_{ik} + w_k\sum_{i=1}^{n}x_{ik} + \lambda\operatorname{sign}\left(w_k\right) \ni 0$$
$$\dots$$

Further reading

- Example gradient descent code on class website
- Boyd and Vandenberghe, Convex Optimization https://stanford.edu/~boyd/cvxbook/
- Mark Schmidt's CPSC 540 notes: https://www.cs.ubc.ca/~schmidtm/Courses/540-W18/L4.pdf
- 3Blue1Brown