Convexity, Strong Convexity, and Smoothness in Optimization

Presented by Zhanyu Wang

February 3, 2021

Outline

- Basics of convexity, strong convexity, and smoothness
- Smoothness: GD, convergence rate w/ or w/o convexity
- Strong convexity: regularization, convergence w/ smoothness
- Gradient descent in practice: choosing step-size
- General descent: lower bounds, accelerated gradient, stochastic gradient, more about SGD

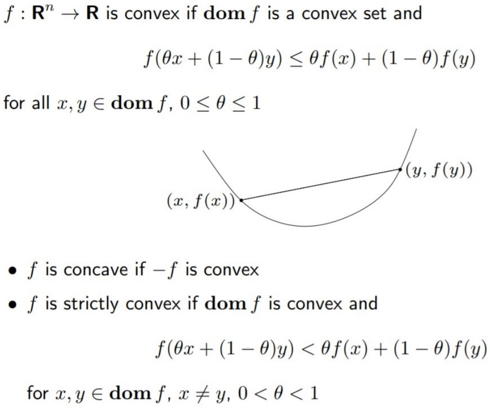
Take-home message: convergence rates and assumptions

Some rules of thumb on convergence rates of $f(x_t) - f(x^*)$ (not comprehensive, and there are other ways).

- $O(1/t)$ is often a result by smoothness.
- $O(1/\sqrt{t})$ uses Lipschitz (thus $\|\nabla f(x)\| = O(1)$) in place of smoothness upper bound on $\|\nabla f(x)\|$. (optimal under Lipschitzness + convexity)
- $O(1/t^2)$ uses "acceleration," which is a fancy momentum inside the gradient. (optimal under smoothness + convexity)
- $exp(-O(t))$ (aka linear convergence) uses strong convexity (or other fine structure on f, e.g., local strong convexity, regularity condition, Polyak-Lojasiewicz condition). (optimal under smoothness + strong convexity)
- Stochasticity changes some rates and what is possible, but there are multiple settings and inconsistent terminology.

Convexity: Intuition and Definition

- First intuition: the second order derivative $f''(x)$ is non-negative.
- For high dimension: the Hessian matrix is positive semi-definite.
- Important property: Jensen's inequality



$f : \mathbf{R}^n \to \mathbf{R}$ is convex if $\mathbf{dom}\, f$ is a convex set and

$$f(\theta x + (1-\theta)y) \le \theta f(x) + (1-\theta) f(y)$$

for all $x, y \in \mathbf{dom}\, f$, $0 \le \theta \le 1$

- f is concave if $-f$ is convex
- f is strictly convex if $\mathbf{dom}\, f$ is convex and

$$f(\theta x + (1-\theta)y) < \theta f(x) + (1-\theta) f(y)$$

for $x, y \in \mathbf{dom}\, f$, $x \ne y$, $0 < \theta < 1$

Convexity: First-order condition

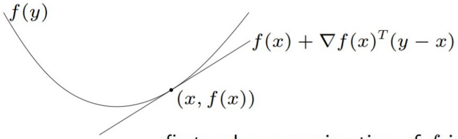
f is **differentiable** if $\mathbf{dom}\, f$ is open and the gradient

$$\nabla f(x) = \left(\frac{\partial f(x)}{\partial x_1}, \frac{\partial f(x)}{\partial x_2}, \ldots, \frac{\partial f(x)}{\partial x_n} \right)$$

exists at each $x \in \mathbf{dom}\, f$

1st-order condition: differentiable f with convex domain is convex iff

$$f(y) \geq f(x) + \nabla f(x)^T (y - x) \quad \text{for all } x, y \in \mathbf{dom}\, f$$



first-order approximation of f is global underestimator

Alternative definition: $\langle \nabla f(x) - \nabla f(y), x - y \rangle \geq 0$

Strong convexity (SC)

- First intuition: the second order derivative $f''(x)$ is positive.
- For high dimension: the Hessian matrix is positive definite.

We define $f(x)$ is λ-strongly-convex (λ-SC) when

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle + \frac{\lambda}{2}\|y - x\|^2, \quad \lambda > 0$$

Some alternative definitions

1. $f(x) - \lambda\|x - x_0\|_2^2/2$ is convex. (lower bounded by a quadratic function)
2. $\nabla^2 f(x) \succeq \lambda I$.
3. $\langle \nabla f(y) - \nabla f(x), y - x \rangle \geq \lambda\|y - x\|^2$
4. $f(\alpha x + (1 - \alpha)y) \leq \alpha f(x) + (1 - \alpha)f(y) - \frac{\lambda\alpha(1-\alpha)}{2}\|y - x\|^2$

Implications of SC

$f(y) \leq f(x) + \nabla f(x)^T(y - x) + \frac{\|\nabla f(y) - \nabla f(x)\|^2}{2\lambda}$; $\quad \|\nabla f(y) - \nabla f(x)\| \geq \mu\|y - x\|$

$\frac{1}{2}\|\nabla f(x)\|^2 \geq \lambda(f(x) - f^*)$; $\quad (\nabla f(y) - \nabla f(x))^T(y - x) \leq \frac{\|\nabla f(y) - \nabla f(x)\|^2}{\lambda}$

Smoothness

- It is NOT the smoothness in Mathematics ($\mathcal{C}^\infty$)
- Lipschitzness controls the changes in function value, while smoothness controls the changes in gradients.

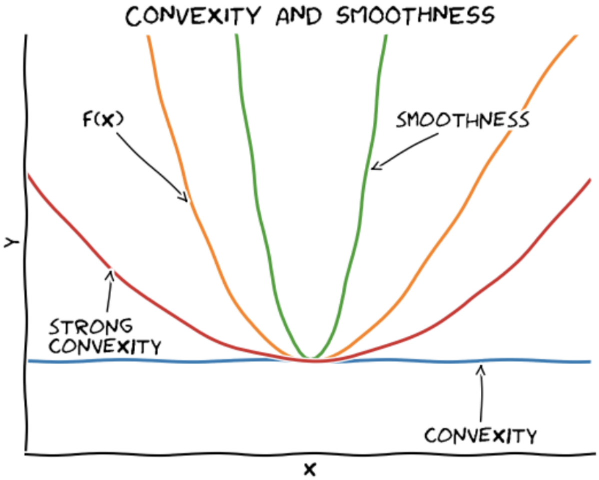
We say $f(x)$ is β-smooth when

$$f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{\beta}{2}\|y - x\|^2, \quad \lambda > 0$$

Some alternative definitions

1. $\beta\|x - x_0\|_2^2/2 - f(x)$ is convex. (upper bounded by a quadratic function)
2. $-\beta I \preceq \nabla^2 f(x) \preceq \beta I$.
3. $\langle \nabla f(y) - \nabla f(x), y - x \rangle \leq \beta\|y - x\|^2$ (weaker when not require convexity)

Illustration of Strong Convexity and Smoothness



Smoothness: guaranteed gradient descent

How to use one definition get another?

$$\langle \nabla f(y) - \nabla f(x), y - x \rangle \le \beta \|y - x\|^2 \Rightarrow f(y) \le f(x) + \langle \nabla f(x), y - x \rangle + \frac{\beta}{2}\|y - x\|^2$$

$$f(y) - f(x) - \langle \nabla f(x), y - x \rangle = \int_0^1 \langle \nabla f(x + t(y - x)), y - x \rangle dt - \langle \nabla f(x), y - x \rangle$$

$$= \int_0^1 \langle \nabla f(x + t(y - x)) - \nabla f(x), y - x \rangle dt \le \int_0^1 t\beta \|y - x\|^2 dt = \beta \|y - x\|^2 / 2$$

- We can guarantee gradient descent does not increase the objective.
- Intuition: the gradient is not changing much, so a small descent step following the gradient would go down.

Let $x' = x - \eta \nabla f(x), \eta \le 2/\beta$

$$f(x') \le f(x) - \langle \nabla f(x), \eta \nabla f(x) \rangle + \frac{\beta \|\eta \nabla f(x)\|^2}{2} = f(x) - (\eta - \eta^2 \beta / 2) \|\nabla f(x)\|^2 \le f(x)$$

Smoothness: convergence to stationary points

Let $w_{i+1} = w_i - \eta_i \nabla f(x), \eta_i \leq 2/\beta$

$$f(w_{i+1}) \leq f(w_i) - (\eta_i - \eta_i^2 \beta/2) \|\nabla f(w_i)\|^2$$

$$f(w_T) \leq f(w_0) - \sum_{i=0}^{T-1} (\eta_i - \eta_i^2 \beta/2) \|\nabla f(w_i)\|^2$$

If we set $\eta_i = \eta$ as a constant

$$\min_{i<T} \|\nabla f(w_i)\|^2 \leq \frac{1}{T(\eta - \eta^2 \beta/2)} (f(w_0) - f(w_T))$$

Here the best choice is $\eta = 1/\beta$. We have no guarantee about the last iterate $\|\nabla f(w_T)\|$: we may get near a flat region at some $i < t$, but thereafter bounce out. For gradient flow, we have similar result: $\inf_{s\in[0,t]} \|\nabla f(w(s))\|^2 \leq \frac{f(w(0))-f(w(t))}{t}$.

Smoothness: view from approximation

- We are using gradient descent because of quadratic approximation

$$w - \nabla f(w)/\beta = \underset{x}{\operatorname{argmin}}(f(w) + \langle \nabla f(w), x - w \rangle + \beta \|x - w\|^2/2)$$

- This is also the key component to proximal gradient descent (approximate the differentiable part and keep the others).
- There are also other approaches: e.g., Nesterov-Polyak cubic regularization guarantees convergence w.r.t. gradient norm $O(t^{-2/3})$ and the minimum eigenvalue of Hessian $-O(t^{-1/3})$ (still could be negative but goes to 0).

Smoothness with convexity: convergence rate

If we choose $\eta = 1/\beta$ and $f(x)$ is convex. Then for any z and sequence $(w_i)_{i \in [0,T]}$ by gradient descent, we have

$$f(w_t) - f(z) \leq \frac{\beta}{2t}(\|w_0 - z\|^2 - \|w_t - z\|^2)$$

Proof:

$$\begin{aligned}
\|w_{t+1} - z\|^2 &= \|w_t - z\|^2 - \frac{2}{\beta}\langle \nabla f(w_t), w_t - z\rangle + \frac{1}{\beta^2}\|\nabla f(w_t)\|^2 \\
&\leq \|w_t - z\|^2 + \frac{2}{\beta}(f(z) - f(w_t)) + \frac{2}{\beta}(f(w_t) - f(w_{t+1})) \\
&= \|w_t - z\|^2 + \frac{2}{\beta}(f(z) - f(w_{t+1}))
\end{aligned}$$

Since $f(w_{t+1})$ is decreasing w.r.t. t, we take the sum and get the result.
For gradient flow, similarly: $\frac{1}{2}(\|w(t) - z\|^2 - \|w(0) - z\|^2) \geq \int_0^t (f(w(s)) - f(z))ds$.

Smoothness with convexity: convergence rate v2

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

Since $f(x)$ is convex, $f(w_k) - f(w^*) \leq -\nabla f(w_k)^T(w^* - w_k) \leq ||\nabla f(w_k)|| \cdot ||w_k - w^*||$

$$\delta_k \leq ||\nabla f(w_k)|| \cdot ||w_k - w^*|| \Rightarrow ||\nabla f(w_k)||^2 \geq \frac{\delta_k^2}{||w_k - w^*||^2} \Rightarrow \delta_{k+1} \leq \delta_k - \frac{\delta_k^2}{2L||w_k - w^*||^2}$$

Since we have shown that $||w_k - w^*||$ is monotonically decreasing,

$$\delta_{k+1} \leq \delta_k - \frac{\delta_k^2}{2L||w_0 - w^*||^2} = \delta_k(1 - \frac{\delta_k}{2L||w_0 - w^*||^2})$$

$$\frac{1}{\delta_k} \leq \frac{1}{\delta_{k+1}} - \frac{1}{2L}\frac{1}{||w_0 - w^*||^2}\frac{\delta_k}{\delta_{k+1}} \Rightarrow \frac{1}{\delta_{k+1}} - \frac{1}{\delta_k} \geq \frac{1}{2L}\frac{1}{||w_0 - w^*||^2}\frac{\delta_k}{\delta_{k+1}} \geq \frac{1}{2L}\frac{1}{||w_0 - w^*||^2}$$

$$\Rightarrow \delta_k \leq \frac{1}{\frac{1}{\delta_0} + k(\frac{1}{2L}\frac{1}{||w_0 - w^*||^2})} = O\left(\frac{1}{k}\right)$$

GD on Convex function without Smoothness

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

$$\begin{aligned} f(w_k) - f(w^*) &\leq \nabla f(w_k)^T (w_k - w^*) = \frac{1}{\eta}(w_k - w_{k+1})^T (w_k - w^*) \\ &\leq \frac{1}{2\eta}\left(\|w_k - w_{k+1}\|^2 + \|w_k - w^*\|^2 - \|w_{k+1} - w^*\|^2\right) \\ &\leq \frac{1}{2\eta}\left(\eta^2 L^2 + \|w_k - w^*\|^2 - \|w_{k+1} - w^*\|^2\right) \end{aligned}$$

$$\sum_{i=0}^{k} f(w_i) - f(w^*) \leq \frac{(k+1)\eta L^2}{2} + \frac{1}{2\eta}(\|w_0 - w^*\|^2 - \|w_{k+1} - w^*\|^2)$$

$$f\left(\frac{1}{k+1}\sum_{i=0}^{k} w_i\right) - f(w^*) \leq \frac{1}{k+1}\sum_{i=0}^{k} f(w_i) - f(w^*) \leq \frac{\eta L^2}{2} + \frac{\|w_0 - w^*\|^2}{2(k+1)\eta} = O\left(\frac{1}{\sqrt{k}}\right)$$

Strong convexity

Suppose $f(x)$ is λ-SC, then

$$\forall w,\ \ f(w) - \inf_v f(v) \leq \frac{1}{2\lambda}\|\nabla f(w)\|^2$$

Proof: Use minimizer of quadradit approximation at w

$$Q_w(v) := f(w) + \langle \nabla f(w), v - w\rangle + \frac{\lambda}{2}\|v - w\|^2.$$

$$\inf_v f(v) \geq \inf_v Q_w(v) = f(w) - \frac{1}{2\lambda}\|\nabla f(w)\|^2$$

We can use this as a stopping criterion: stop when $\|\nabla f(x)\| \leq \sqrt{2\lambda\epsilon}$.
Many software packages use heuristics. Some people just run their methods as long as possible. In convex cases, sometimes we can compute duality gaps.

Strong convexity: regularization and boundedness

- If we manually enhance strong convexity: $f_\lambda(w) = f(w) + \lambda\|w\|^2/2$, the new optimal solution w^* satisfies

$$\frac{\lambda}{2}\|w^*\|_2^2 \leq f_\lambda(w^*) \leq f_\lambda(0) = f(0)$$

- It is now in bounded region and can be used in generalization bounds.
- In deep learning people use weight decay, but it isn't necessary for generalization (although it helps a lot), and is much smaller than what many generalization analyses suggest (the new function is still not convex), and thus its overall role is unclear.
- Time Matters in Regularizing Deep Networks
- Three Mechanisms of Weight Decay Regularization
  1. increasing the effective learning rate,
  2. approximately regularizing the input-output Jacobian norm, and
  3. reducing the effective damping coefficient for second-order optimization.

Strong convexity with Smoothness: convergence rate

Suppose f is λ-SC and β-smooth, and GD is run with step size $1/\beta$. Then a minimum w^* exists, and

$$f(w_t) - f(w^*) \leq (f(w_0) - f(w^*)) exp(-t\lambda/\beta),$$

$$\|w_t - w^*\|^2 \leq \|w_0 - w^*\|^2 exp(-t\lambda\beta).$$

Proof: (consider β/λ as condition number; use the fact that $\forall x \geq 0,\ e^{-x} \geq 1 - x$.)

$$f(w_{i+1}) - f(w^*) \leq f(w_i) - f(w^*) - \frac{\|\nabla f(w_i)\|^2}{2\beta} \leq f(w_i) - f(w^*) - \frac{2\lambda}{2\beta}(f(w_i) - f(w^*))$$

$$\begin{aligned} \|w_{i+1} - w^*\|^2 &= \|w_i - w^*\|^2 + \frac{2\nabla f(w_i)^T(w^* - w_i)}{\beta} + \frac{\|\nabla f(w_i)\|^2}{\beta^2} \\ &\leq \|w_i - w^*\|^2 + \frac{2(f(w^*) - f(w_i) - \lambda\|w^* - w_i\|^2/2)}{\beta} + \frac{2\beta(f(w_i) - f(w_{i+1}))}{\beta^2} \end{aligned}$$

Use Gradient descent in Practice

The worst optimization method in the world. – Aaron Defazio

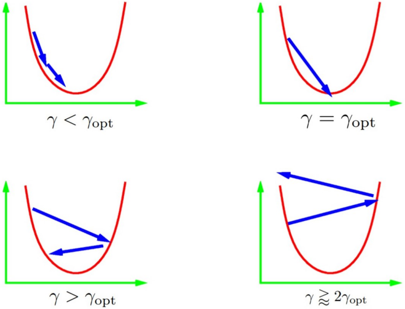
The condition number only formally makes sense on simple problems (“strongly convex”). But we often talk about “poorly conditioned” and “well conditioned” problems in machine learning informally.

Typically we don’t have a good estimate of the learning rate!

Standard practice is to try a bunch of values on a log scale and use the one that gave the best final result

Learning rates that are too large cause “divergence”, where the function value (loss) explodes

The optimal learning rate can change during optimization! Often decreasing it over time is necessary


$\gamma < \gamma_{\text{opt}}$
$\gamma = \gamma_{\text{opt}}$
$\gamma > \gamma_{\text{opt}}$
$\gamma \gtrapprox 2\gamma_{\text{opt}}$

Choosing the Step-Size in Practice

- In practice, you should never use $\eta = 1/\beta$: hard to compute β and this choice of η is usually too small (small for worst case).
- To approximate β:
  - start with a small guess $\widehat{\beta}$, e.g., $\beta_0 = 1$
  - Double $\widehat{\beta}$ if below is not satisfied

$$f(w_k - \nabla f(w_k)/\widehat{\beta}) \leq f(w_k) - \|\nabla f(w_k)\|^2/(2\widehat{\beta}).$$

  - Worst case: $\beta \leq \widehat{\beta} \leq 2\beta$; Good case: $\widehat{\beta} << \beta$ and you get great speedup.
- Another way: backtracking line-search for step-size α:
  - start with a large guess α
  - Decrease α until if Armijo condition is satisfied (often choose $\gamma = 10^{-4}$)

$$f(w_k - \alpha \nabla f(w_k)) \leq f(w_k) - \alpha\gamma\|\nabla f(w_k)\|^2 \text{ for } \gamma \in (0, 1/2].$$

  - Good codes usually only try 1 value per iteration.
- Even more fancy line-search: Wolfe conditions. Check Nocedal and Wright's Numerical Optimization book.

Lower bounds for first-order methods

- Black Box Model: $\forall i, x_i \in x_0 + \text{span}\{\nabla f(x_0), \dots, \nabla f(x_{i-1})\}$
- Let $k \leq d$ (dimension) then there exists a convex L-Lipschitz function f such that

$$f(x_k) - f(x^*) \geq \frac{RL}{2(1+\sqrt{k+1})}$$

dimension dependent version

- (Nemirovski, Yudin '83) There exists an L-smooth convex function f such that

$$f(x_k) - f(x^*) \geq \frac{3L}{32} \frac{\|x_0 - x^*\|^2}{(k+1)^2}$$

- Let $\kappa = L/\mu > 1$. Then there exists an L-smooth μ-strongly convex function f

$$f(x_k) - f(x^*) \geq \frac{\mu}{2} \left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1} \right)^{2(k-1)} \|x_k - x^*\|^2$$

Accelerated gradient methods

- Heavy-ball method (Boris Polyak): $w_{t+1} = w_t - \eta_t \nabla f(w_t) + \theta_t (w_t - w_{t-1})$.
- Yurii Nesterov's accelerated gradient methods ('83)

$$x_{t+1} = y_t - \eta_t \nabla f(y_t),$$
$$y_{t+1} = x_{t+1} + \frac{t-1}{t+2}(x_{t+1} - x_t).$$

  Convergence rate: suppose f is convex and L-smooth, $\eta_t = 1/L$,

$$f(w_t) - f^* \leq \frac{2L\|w_t - w^*\|^2}{(t+1)^2}.$$

  The insight is from a second-order ODE and corresponding Lyapunov function

$$\ddot{X}(\tau) + \frac{\alpha}{\tau}\dot{X}(\tau) + \nabla f(X(\tau)) = 0,\ \alpha \geq 3 \Rightarrow f(X(\tau)) - f^* \leq O(\frac{1}{\tau^2}).$$

- Fast iterative shrinkage-thresholding algorithm FISTA (Beck, Teboulle '09)
- When a certain criterion is met, restart running FISTA (O'Donoghue, Candes '12)

General descent

We first consider a generalization of gradient descent (e.g., SGD or coordinate descent)

$$w_{i+1} := w_i - \eta g_i$$

Assume $f(x)$ is convex. We have

$$\begin{aligned}\|w_{i+1} - z\|^2 &= \|w_i - z\|^2 + 2\eta\langle g_i - \nabla f(w_i) + \nabla f(w_i), w_i - z\rangle + \eta^2\|g_i\|^2 \\ &\leq \|w_i - z\|^2 + 2\eta(f(z) - f(w_i) + \langle g_i - \nabla f(w_i), w_i - z\rangle) + \eta^2\|g_i\|^2\end{aligned}$$

Let $\epsilon_i := \langle g_i - \nabla f(w_i), w_i - z\rangle$. $\eta = \frac{c}{\sqrt{t}}$, $G := \max_i \|g_i\|_2$, $D := \max_i \|w_i - z\|$. We have

$$f\left(\frac{1}{t}\sum_{i=0}^{t-1} w_i\right) \leq \frac{1}{t}\sum_{i=0}^{t-1} f(w_i) \leq f(z) + \frac{\|w_0 - z\|^2}{2c\sqrt{t}} + \frac{cG^2}{2\sqrt{t}} + \frac{1}{t}\sum_{i=0}^{t-1}\epsilon_i.$$

- Averaged iterates are often suggested in theory, but rare in applied classification (loss is not convex). If t not fixed in advance, we can use $\eta_i = c/\sqrt{1+i}$.
- $|\epsilon_i| \leq 2\|\nabla f(w_i)\|\|w_i - z\| \leq 2GD$. We need concentration of the sum.

Stochastic gradients

Define the standard stochastic gradient oracle:

$$\mathbb{E}[g_i | w_{\leq i}] = \nabla f(w_i).$$

- One way is to use one sample (x, y) from population to calculate the loss and gradient. If each time the sample is new, then it is SGD on population risk; If the sample is always from training data, then it is on empirical risk.
- Stochastic minibatch gradient descent is standard for deep networks. However, there is a delicate interplay between step size, minibatch size, and number of training epochs (Shallue et al. '18).
- Matus Telgarsky said: There are many deep learning papers that claim SGD does miraculous things to the optimization process. Unfortunately, none of these seem to come with a compelling and general theoretical analysis.

Stochastic gradients

Azuma-Hoeffding inequality: suppose Z_i is a martingale difference sequence $\mathbb{E}[Z_i|Z_{<i}] = 0$ and $\mathbb{E}|Z_i| \le R$. W.p. at least $1-\delta$, we have $\sum_{i=1}^t Z_i \le R\sqrt{2t\log(1/\delta)}$.

We know $\mathbb{E}[\epsilon_i|w_{\le i}] = \langle \mathbb{E}[g_i - \nabla f(w_i)|w_{\le i}], z - w_i\rangle = 0$ and $\mathbb{E}|\epsilon_i| \le 2GD$. Therefore, we let $c = D/G$, and w.p. at least $1-\delta$, we have

$$f\left(\frac{1}{t}\sum_{i=0}^{t-1} w_i\right) \le \frac{1}{t}\sum_{i=0}^{t-1} f(w_i) \le f(z) + \frac{DG}{2\sqrt{t}} + \frac{DG}{2\sqrt{t}} + \frac{2DG\sqrt{2\log(1/\delta)}}{\sqrt{t}}.$$

- Without Azuma-Hoeffding we still can get a bound on the expected average error:

$$\mathbb{E}\left[\frac{1}{t}\sum_{i=0}^{t-1} f(w_i)\right] \le f(z) + \frac{\|w_0 - z\|^2 + G^2}{2\sqrt{t}}.$$

A more careful analysis lets us use the last iterate (Shamir and Zhang '13, Thm 2)

Stochastic gradients: convergence v2

$$x_{k+1} = x_k - \eta_k(\nabla f(x_k) + \epsilon_{k+1})$$

Assume $\mathbb{E}[\epsilon_{k+1}|\mathcal{F}_k] = 0, \mathbb{E}[\|\epsilon_{k+1}\|^2|\mathcal{F}_k] = \sigma^2$, f is μ-SC and L-smooth, $\eta_k \leq 1/L$

$$\mathbb{E}[f(x_{k+1}) - f(x^*)|\mathcal{F}_k] \leq f(x_k) - f(x^*) - \eta_k\|\nabla f(x_k)\|^2(1 - \frac{L}{2}\eta_k) + \frac{L}{2}\eta_k^2\sigma^2$$

$$\mathbb{E}[\delta_{k+1}|\mathcal{F}_k] \leq \delta_k(1 - \eta_k\frac{\mu^2}{L}) + \frac{L}{2}\eta_k^2\sigma^2$$

Simplified Robbins-Siegmind Lemma: Suppose $V_k, k \geq 1$ is a sequence of random variables and $\alpha_k, k \geq 1$, $\beta_k, k \geq 1$ are positive-valued deterministic sequences,
$\mathbb{E}[V_{k+1}|\mathcal{F}_k] \leq V_k(1 - \alpha_k) + \beta_k$
If $\sum \alpha_k = \infty, \sum \beta_k < \infty, \beta_k/\alpha_k \to 0$, then $V_k \to 0$ *a.s.*, $\mathbb{E}[V_k] \to 0$

$$\sum \alpha_k = \infty \Rightarrow \sum \eta_k = \infty$$

$$\sum \beta_k < \infty \Rightarrow \sum \eta_k^2 < \infty$$

Why stochastic gradients?

- Why SGD in ML? In statistical problems, we shouldn't expect test error better than $1/\sqrt{n}$ or $1/n$ anyway, so we shouldn't optimize to crazy accuracy. With SGD, the periteration cost is low. Meanwhile, heavyweight solvers like Newton methods require a massive per-iteration complexity, with the promise of crazy accuracy; but, again we don't need that crazy accuracy here.

More about SGD

- SGD + Armijo
- Aaron Defazio Lecture in NYU (About using SGD in deep learning)
  - SGD vs GD: (19:18) At early stages, the correlation is remarkable. The stochastic gradient can be correlated up to a coefficient of 0.999 to the true gradient.
  - Minibatch: (23:09) Yann recommended we used mini batches equal to the size of the number of classes in our data set. (36:20) Change your learning rate when you change your batch size, rather than changing the momentum.
  - Momentum: (40:35) This acceleration is hard to realize when you have stochastic gradients. Probably why momentum helps is noise smoothing: it averages gradients.
  - Initialization: (63:45) For large neural network, as long as you use the same variance scaling initialization, you'll end up practically same quality solutions.
  - BatchNorm: (71:07) it is no longer SGD if you use BatchNorm. (72:45) Group Norm works better sometimes. (74:19) In terms of a practical consideration, this normalization makes the weight initialization that you use a lot less important.

References

The content in this slide comes from

- Deep learning theory lecture notes – Section 9-12
- Princeton ELE 522: Large-Scale Optimization for Data Science
- UBC CPSC 540: Machine Learning (Mark Schmidt) - Lec 4: Convergence of GD
- Convex Optimization (Boyd and Vandenberghe). Chapter 3.
- Fenchel Duality between Strong Convexity and Lipschitz Continuous Gradient
  - Check this if you would like to know the relationship between different definitions

Further reading:

- Optimization Methods for Large-Scale Machine Learning. 2018
- Optimization for deep learning: theory and algorithms. 2019