ML & Optimization Lecture 14

- Stochastic Gradient Descent (SGD)
- Tricks & Tips for SGD a la Bottou
- Adaptive Gradient Descent Algorithms:
  per coordinate stepsizes & exponential averaging
  - AdaGrad (2011)
  - RMSProp (2012)
  - AdaDelta (2012)
  - ADAM (2015)
- SGD w/ momentum (Polyak heavy-ball method) (?)
- Nesterov's Accelerated Gradient Descent (NAG) (?)

GD for solving a linear system

$$f(w) = \frac{1}{2}\|Xw - y\|_2^2 \qquad X = \begin{bmatrix} x_1^T \\ \vdots \\ x_n^T \end{bmatrix} \in \mathbb{R}^{n \times d}$$

$$\Rightarrow \nabla f(w) = X^T(Xw - y)$$

$$= X^T X w - X^T y$$

Two cases:

1) can store $X^T X \in \mathbb{R}^{d \times d}$, then cost of computing $\nabla f(w)$ is $O(d^2)$

2) can't store $X^T X$.
cost of computing $\nabla f(w)$:
   i) compute residual $Xw - y$ in $O(nd)$
   ii) compute $X^T \cdot$ residual $O(nd)$ again
total cost is $O(nd)$

Usually $n \gg d$, so both costs are $\Omega(d^2)$

Problem: GD/subgradient descent is expensive if n or d are large because cost of gradient or subgradient oracle is $\mathcal{O}(nd)$.

We want a cheaper algorithm, ideally $\mathcal{O}(d)$ per iteration

Idea: notice that $\nabla f(w_t)$ is already an approximate search direction

$$f(w) \approx f(w_t) + \langle \nabla f(w_t), w - w_t \rangle + \frac{1}{2\alpha_t} \|w - w_t\|_2^2$$

if $f(w) = \frac{1}{n} \sum_{i=1}^{n} \ell(w^T x_i, y_i)$

$$\Rightarrow \nabla f(w) = \frac{1}{n} \sum_{i=1}^{n} x_i \ell'(w^T x_i, y_i)$$

Idea: $\nabla f(w)$ is a sample average that approximates a population average

Treat $\nabla f(w)$ itself as a population average, and approximate it with a sample average

- select a random subset $I \subset [n] = \{1, \dots, n\}$ of size k and choose an approximation

$$g = \frac{1}{k} \sum_{i \in I} x_i \ell'(w^T x_i, y_i)$$

$$\Rightarrow \mathbb{E} g = \nabla f(w)$$

The set I is called a <u>minibatch</u>

Stochastic Gradient Descent (SGD)

$$x_* = \operatorname*{argmin}_x f(x)$$

Choose a batch size K

for $t = 0, \dots, T-1$

- sample $I \subseteq [n]$ of size K
- $g_t = \frac{1}{K} \sum_{i \in I} \nabla l_i(x_t)$
- $x_{t+1} = x_t - \alpha_t g_t$

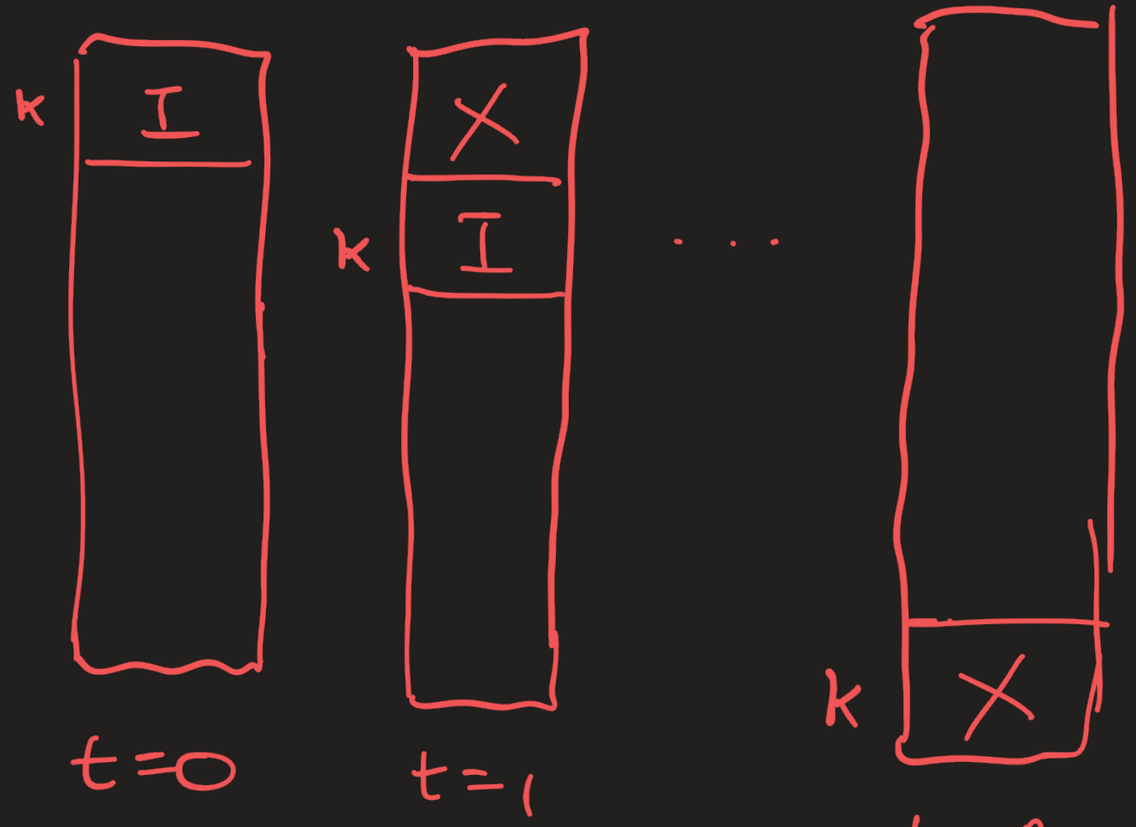
return $\hat{x} = \frac{1}{T} \sum_{i=1}^{T} x_i$

$l_i(x) \equiv l(x^T z_i, y_i)$

- datapoint i
- loss on data point (z_i, y_i)

In practice, to efficiently use all your data and avoid cycling degeneracies, we use minibatches formed in the following way:

- randomize the order of the training data
- everytime you sample a minibatch, use the first k points you have not already used



Here are $\frac{n}{k}$ iterates in one epoch

- once you've touched all of your data (this is called an epoch) then reshuffle your data and start repeating

Minibatch SGD

Inputs: ∇f oracle, α_t stepsizes, K minibatch size, E epochs

Alg $z_0 = x_0$

for $e = 1, \dots, E$

$(X, y) \leftarrow$ random permutation of itself

$x_0 = z_{e-1}$

for $t = 0, \dots, \frac{n}{K} - 1$

$I \leftarrow$ first K untouched samples in this epoch

$$g_t \leftarrow \frac{1}{K} \sum_{i \in I} \nabla l_i(x_t)$$

$$x_{t+1} = x_t - \alpha_t g_t$$

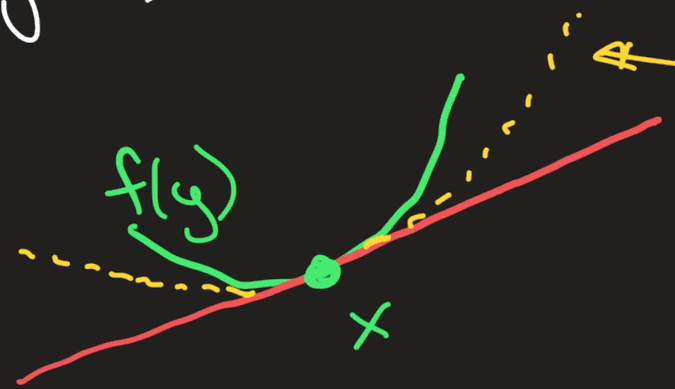
$z_e = x_{\frac{n}{K}}$

return z_E

Analysis of SGD convergence

1) f is strongly cvx (μ-strongly convex)

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle + \frac{\mu}{2} \|y - x\|_2^2$$



$$f(x) + \langle \nabla f(x), y - x \rangle + \frac{\mu}{2} \|y - x\|_2^2$$

2) the estimate of the gradient $\nabla f(x_t)$ given by g_t is unbiased and has bounded variance:

1) $\mathbb{E}[g_t \mid x_0, \dots, x_t] = \nabla f(x_t)$ "unbiased"

2) $\mathbb{E}[\|g_t\|_2^2] \leq B^2$ "finite variance" condition

Convergence rate for SGD

Outline: first show

$$\mathbb{E}\left[f(x_{t-1}) - f(x_*)\right] \leq \frac{B^2}{2\mu t} + \left[\frac{\mu(t-1)}{2}\mathbb{E}\|x_{t-1} - x^*\|_2^2 - \frac{\mu t}{2}\mathbb{E}\|x_t - x^*\|_2^2\right]$$

using strong convexity and $\alpha_t = \frac{1}{\mu t}$

$$c_t \equiv \frac{\mu t}{2}\mathbb{E}\|x_t - x^*\|_2^2$$

Use telescoping sums to show

$$\sum_{i=1}^{T}\mathbb{E}\left[f(x_{t-1}) - f(x_*)\right] \leq \frac{B^2}{2\mu}\sum_{t=1}^{T}\frac{1}{t} + \sum_{t=1}^{T}\left[c_{t-1} - c_t\right]$$

$$\leq \frac{B^2}{2\mu}\log(T) + \left(0 - \frac{\mu T}{2}\mathbb{E}\|x_T - x^*\|_2^2\right)$$

$$\leq \frac{B^2}{2\mu}\log(T)$$

Now divide both sides by T:

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\{f(x_{t-1}) - f(x_*)\} \leq \frac{B^2}{2\mu T}\log(T)$$

$$\Rightarrow \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}f(x_{t-1}) \leq f(x_*) + \frac{B^2}{2\mu T}\log(T)$$

by Jensen's ineq & the fact f is convex

$$\mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T}f(x_{t-1})\right] \geqslant \mathbb{E}\left[f\left(\frac{1}{T}\sum_{t=1}^{T}x_{t-1}\right)\right]$$

so $\hat{x}_T = \frac{1}{T}\sum_{t=1}^{T}x_{t-1}$ be the Polyak average

we have established that

$$\mathbb{E} f(\hat{x}_T) \leq f(x^*) + \frac{B^2}{2\mu T} \log(T)$$

Tips & Trick for SGD (Bottou 2012)

- How to choose stepsizes, in practice
  - use a small subset of your data to test convergence on (look at model accuracy, not loss)
  - use a stepsize schedule like $\alpha_t = \frac{\beta_0}{(\beta_0 + \beta_1 t)}$

    choose β_0 and β_1 on your small dataset

- Check the correctness of your gradient code using finite differences:

$$f(x+h) - f(x) = \langle \nabla f(x), h \rangle + o(\|h\|_2)$$

$$\Rightarrow \quad \frac{|f(x+h) - f(x) - \langle \nabla f(x), h \rangle|}{\|h\|_2} \to 0 \quad \text{as} \quad h \to 0$$