# Gradient descent

$$\min f(x)$$

- GD
$$x_{k+1} = x_k - s_k \nabla f(x_k)$$
- If the eigenvalues $\lambda_i$ of the Hessian $H$ are $m \leq \lambda_i \leq M$ for all $i$, and in particular if the Hessian is strongly convex: $m > 0$
$$f(x_{k+1}) - f(x^*) \leq (1 - \frac{m}{M})(f(x_k) - f(x^*))$$
each step gets us closer by a constant factor: $(1 - \frac{m}{M})$, i.e.
- i.e., the convergence $O((1 - \frac{m}{M})^k)$ for $0 < c < 1$.
- This means that a bound of
$$f(x_k) - f(x^*) \leq \epsilon$$
can be achieved using only $O(\log(1/\epsilon))$ iterations.
- This rate is called "linear convergence"
- If the Hessian is not strongly convex, the convergence is $O(k)$.

# GD and learning

- Let's look at our linear regression objective (in machine learning we typically scale the loss by the number of data points, but that shouldn't affect the minimization

$$R(x) = \frac{1}{n}\|Ax - b\|^2 = \frac{1}{n}\sum_{i=1}^{n}(x^T a_i - b_i)^2$$

- To learn $x$, we can follow the GD descent algorithm

$$x_{k+1} = x_k - s_k \nabla R(x)$$

where

$$\nabla R(x) = \frac{1}{n}(2x^\top A^\top A - 2b^\top A) = \frac{1}{n}\sum_{i=1}^{n} 2(x^T a_i - b_i)a_i$$

# GD and learning

- Our linear regression objective is an example of a loss, which is a function of the parameter $x$ and the data $a_i, b_i$

$$\ell_i(x) = (x^T a_i - b_i)^2$$

- You can generalize this to other loss functions and learning algorithms

$$\ell_i(x) = \ell(F(x, a_i) - b_i)$$

- E.g., $F$ can represent a neural network, which outputs a label $F(x, a)$ given features $a$ and parameters $x$.
- The training, i.e., learning $x$ given the data, can be done by minimizing

$$L(x) = \frac{1}{n} \sum_{i=1}^{n} \ell_i(x)$$

- Sometimes the above term is called is called empirical risk, and the training process is called empirical risk minimization

# GD and large data

- However computing

$$\nabla L(x) = \frac{1}{n} \sum_{i=1}^{n} \nabla \ell_i(x)$$

  is $O(n)$
- At each step the descent method needs to touch $n$ points
- Can we make progress without looking at all the data?

# Stochastic gradient descent

- Previously we've randomly sampled large matrices, s.t the expectation of the smaller matrix was equal to the larger matrix.
- We can do the same here
- Intuition
  - GD is an iterative process
  - At every step, we have a chance to recover from previous missteps
- Turns out even terrible estimates work as long as they are unbiased

# minibatch/SGD

- In the GD we take the gradient of the full batch (all of the $n$ samples

$$\nabla L(x) = \frac{1}{n} \sum_{i=1}^{n} \ell_i(x)$$

- An algorithm can choose a batch of size of $B$ uniformly at random $i(1,k), ..., i(B,k)$ (*minibatch GD*)
- When $B = 1$, the sampling chooses a single $i(k)$ at step $k$ uniformly at random (*SGD*)
- Alternatively a random ordering of the data can be used (this is used in practice).

# SGD and OLS

- Let say our features are 1D, i.e. $a_i$ and $b_i$ are both scalars.
- Then

$$dl_i(x)/dx = 2a_i(x^T a_i - b_i)$$

and solving

$$\nabla L(x) = \frac{1}{n}\sum_{i=1}^{n} \nabla \ell_i(x) = \frac{1}{n}\sum_{i=1}^{n} 2a_i(x^T a_i - b_i) = 0$$

gives the OLS estimate

$$x^* = \frac{\sum_{i=1}^{n} a_i b_i}{\sum_{i=1}^{n} a_i^2}$$

# SGD and OLS

- If for all $i$,

$$\frac{b_i}{a_i} \leq \frac{B}{A} \Rightarrow Aa_i b_i \leq Ba_i^2 \Rightarrow A\sum a_i b_i \leq B\sum a_i^2 \Rightarrow x^* \leq \frac{B}{A}$$

- Similarly if for all $i$,

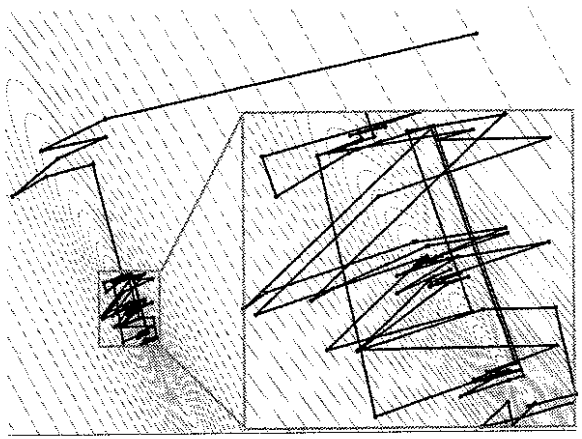
$$\frac{\beta}{\alpha} \leq \frac{b_i}{a_i} \Rightarrow \frac{\beta}{\alpha} \leq x^*$$

# SGD and OLS

- If for all $i$,
$$\frac{\beta}{\alpha} \leq \frac{b_i}{a_i} \leq \frac{B}{A}$$
- Therefore if $x_k$ is outside $I = [\frac{\beta}{\alpha}, ..., \frac{B}{A}]$, then GD will move towards that interval.
- Expect SGD to do the same since
$$x^*_{i(k)} = a_{i(k)} b_{i(k)} / a^2_{i(k)}$$

# SGD and OLS

- Also if $x_k$ is inside $I$, then so will be $x_{k+1}$ for both GD and SGD
- However,
  - GD will converge to $x^*$
  - While SGD will bounce around
- This is OK because in the overparametrized regime, like NN, you don't need (or want) to fit the training data perfectly
- Need to avoid overfitting
- Also this justifies early stopping of SGD



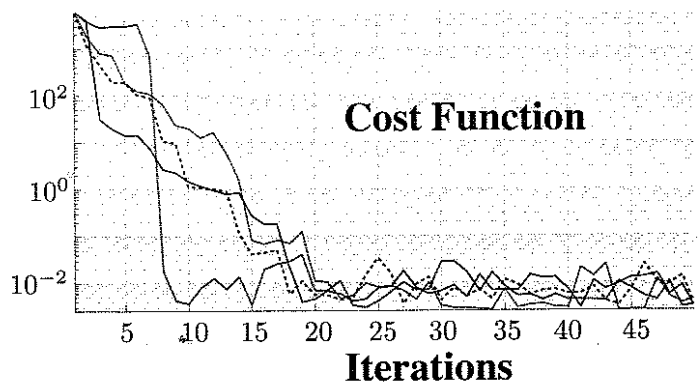



Figure: Fig from p 362 in [1]

# minibatch/SGD in expectation

- For SGD,

$$\mathbb{E}\nabla \ell_{i(k)}(x) = \sum_{j=1}^{n} P(i(k) = j)\nabla \ell_{i(k)}(x)$$
$$= \frac{1}{n}\sum_{j=1}^{n} \nabla \ell_j(x) = \nabla L(x)$$

- So SGD uses an unbiased esimator of gradient.
- A similar argument shows that *minibatch GD* is also unbiased.

# SGD convergence

- However, you can achieve convergence in expectation if the step size is

$$s = \text{constant}/\sqrt{T}$$

where $T$ is the number of steps, and

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

- If $f$ is convex and twice differentiable, the first assumption implies that the largest eigenvalue of the Hessian is bounded by $L$
- This clear in 1D by mean value theorem, and also follows in $\mathbb{R}^m$ by multivariate mean value theorem
- Therefore, with a step $-s\nabla f_{i(k)}$, by Taylor's theorem

$$f(x_{k+1}) \leq f(x_k) + \langle \nabla f(x_k), -s\nabla f_{i(k)} \rangle + \frac{1}{2} L s^2 \|\nabla f_{i(k)}\|^2$$

# SGD convergence

- From the previous slide

$$f(x_{k+1}) \leq f(x_k) + \langle \nabla f(x_k), -s \nabla f_{i(k)} \rangle + \frac{1}{2} L s^2 \|\nabla f_{i(k)}\|^2$$

- Taking the expectation of this expression gives

$$\mathbb{E} f(x_{k+1}) \leq \mathbb{E} f(x_k) - s \mathbb{E}[\|\nabla f(x_k)\|^2] + \frac{1}{2} L s^2 G^2$$

and rearranging the above

$$\mathbb{E} \|\nabla f(x_k)\|^2 \leq \frac{1}{s} \mathbb{E}[f(x_k) - f(x_{k+1})] + \frac{1}{2} L s G^2$$

# SGD convergence

- From the previous slide

$$\mathbb{E}\|\nabla f(x_k)\|^2 \leq \frac{1}{s}\mathbb{E}\left[f(x_k) - f(x_{k+1})\right] + \frac{1}{2}LsG^2$$

- Choosing the step size $s = c/\sqrt{T}$ and summing the above expression from 1 to $T$, the sum on the RHS telescopes

$$\frac{1}{T}\sum_{k=1}^{T}\mathbb{E}\|\nabla f(x_k)\|^2 \leq \frac{1}{\sqrt{T}}\left(\mathbb{E}\left[\frac{f(x_1) - f(x_{k+1})}{c}\right] + \frac{1}{2}LcG^2\right)$$

- Now we use the fact that $\mathbb{E}f(x_1) = f(x_1)$ and $\mathbb{E}f(x_k) \geq f(x^*)$

$$\frac{1}{T}\sum_{k=1}^{T}\mathbb{E}\|\nabla f(x_k)\|^2 \leq \frac{1}{\sqrt{T}}\left(\frac{f(x_1) - f(x^*)}{c} + \frac{1}{2}LcG^2\right) = \frac{C}{\sqrt{T}}$$

- Here we picked a fixed step size $s$.
- However, optimizing for $s_k$ reveals that decreasing the step sizes give you a better rate. Intuitively, you want to dampen noise in the step direction as you get closer to the solutions

# SGD convergence

- From the previous slide

$$\frac{1}{T}\sum_{k=1}^{T}\mathbb{E}\|\nabla f(x_k)\|^2 \leq \frac{1}{\sqrt{T}}\left(\frac{f(x_1)-f(x^*)}{c}+\frac{1}{2}LcG^2\right)=\frac{C}{\sqrt{T}}$$

- Since the smallest term is below average

$$\min_{1\leq k\leq T}\mathbb{E}\|\nabla f(x_k)\|^2 \leq \frac{C}{\sqrt{T}}$$

# Next steps

- ADAGRAD, ADAM
- Construction of deep neural networks (Sec. VII.1)
- Convolutional neural nets (Sec. VII.2)