Day 16 - Gradient Descent and Stochastic Gradient Descent

Outline:

Gradient Descent (GD)
Stochastic Gradient Descent (SGD)
Convex Optimization
Convergence of GD

Optimization and machine learning

Data $\{(x_i, y_i)\}_{i=1\cdots n}$

Consider a model $\hat{y}_\theta(x_i)$

$$\min_{\theta} \sum_{i=1}^{n} \ell(\hat{y}_\theta(x_i), y_i)$$

Optimization in general

$$\min_{x} f(x)$$

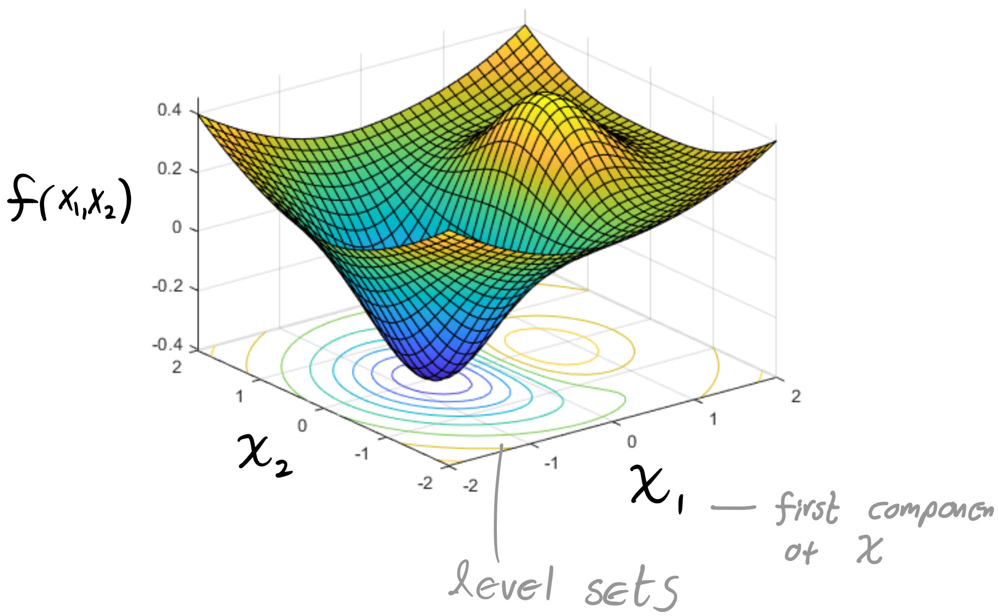
Gradient descent: Take successive steps "downhill"

$$x^{(i+1)} = x^{(i)} - \alpha \nabla f(x^{(i)})$$

iteration index

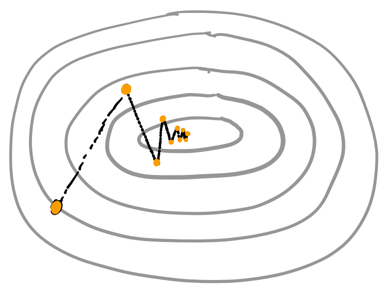
step size, learning rate

$-\nabla f$ points in direction of steepest descent



Depiction of gradient descent

Top down view:



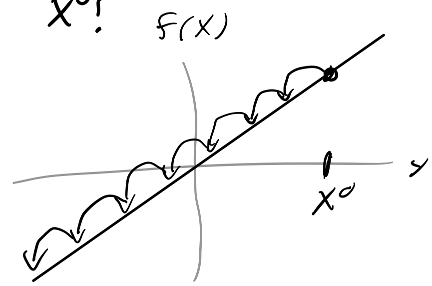
Recall: gradient is orthogonal to level sets

Example: Suppose $f: \mathbb{R} \to \mathbb{R}$, $f(x) = x$.

What is sequence of points given by GD if starting from x^0?

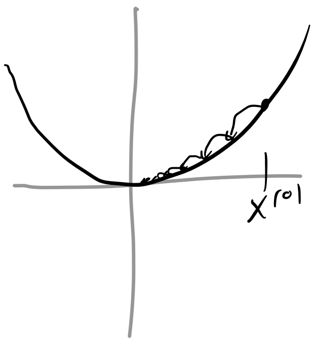
$\nabla f(x) = 1$

$x^{(i+1)} = x^{(i)} - 1$



$f: \mathbb{R} \to \mathbb{R}$

Example: $f(x) = \frac{1}{2} L x^2$. If GD is initialized at $x^{(0)}$, what is value of $x^{(n)}$?



$$x^{(i+1)} = x^{(i)} - \alpha L x^{(i)} = (1-\alpha L) x^{(i)}$$

$$\boxed{x^{(i)} = (1-\alpha L)^i x^{(0)}}$$

When does $x^{(i)}$ converge to minimizer of f as $i \to \infty$?

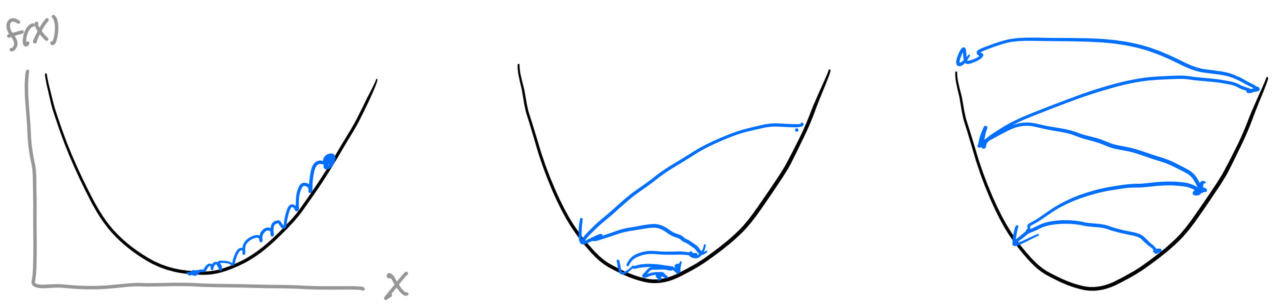
recall: $\lim_{n \to \infty} r^n = 0$ if $|r| < 1$

If $|1-\alpha L| < 1$ then $x^{(i)} \to 0$

If $-1 < \underline{1-\alpha L < 1}$ . . . .

$1 - \alpha L > -1 \Rightarrow 2 > \alpha L \Rightarrow \boxed{\alpha < \frac{2}{L}}$

Picture:



Small learning rate

medium learning rate

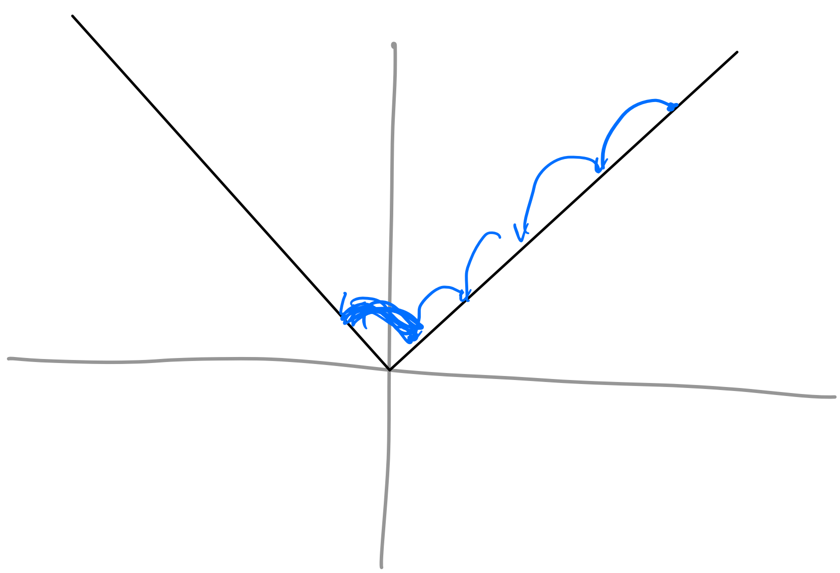
high learning rate

GD (for a quadratic function) converges if the step size is small enough

Step size should scale like 1 / curvature …. Where curvature is the second derivative

$f: \mathbb{R} \to \mathbb{R}$

Example: $f(x) = |x|$. If GD is initialized at $x^{(0)}$, Describe what GD will do.



Challenges of gradient descent in machine learning & minibatches

$$\underbrace{\min_{\theta} \frac{1}{n}\sum_{i=1}^{n} \ell(\hat{y}_\theta(x_i), y_i)}_{f(\theta)}$$

$$\theta^{k+1} = \theta^k - \alpha \nabla f(\theta) = \theta^k - \alpha \frac{1}{n}\sum_{i=1}^{n} \nabla_\theta \ell(\hat{y}_\theta(x_i), y_i)$$

To evaluate $\nabla f(\theta)$, one needs to loop through all data (batch gradient descent)

- expensive
- not possible in some contexts

Idea: use minibatches

Select a minibatch $B \subset \{1, 2, \cdots, n\}$

$$\theta^{k+1} = \theta^k - \alpha \underbrace{\frac{1}{|B|}\sum_{i \in B} \nabla_\theta \ell(\hat{y}_\theta(x_i), y_i)}_{\text{use as approximation of } \nabla_\theta f(\theta)}$$

If you try to generate a minibatch by selecting a random subset of B data points uniformly, what practical challenges arise?

Issue is disk access

Idea: store the data randomly so you can draw a minibatch sequentially

What considerations would affect the minibatch size you should use?

Your GPU has finite RAM, so choose a minibatch that maxes out your RAM

If the minibatch is chosen randomly, on average, the gradient of a minibatch is the full gradient

$\Rightarrow$ Stochastic gradient descent

Stochastic Gradient Descent

Want to solve $\min_x f(x)$

Instead of having access to $\nabla f(x)$, suppose only have $G(x)$ w/ $\mathbb{E}[G(x)] = \nabla f(x)$.

Stochastic estimate of ∇f

Write SGD as

$$x^{(k+1)} = x^{(k)} - \alpha_k G(x^{(k)})$$

- on average, move in direction of steepest descent
- may move further from minimizer

Simple model: additive noise

$$G(x) = \nabla f(x) + w, \qquad w \sim \mathcal{N}(0, \sigma^2 I)$$

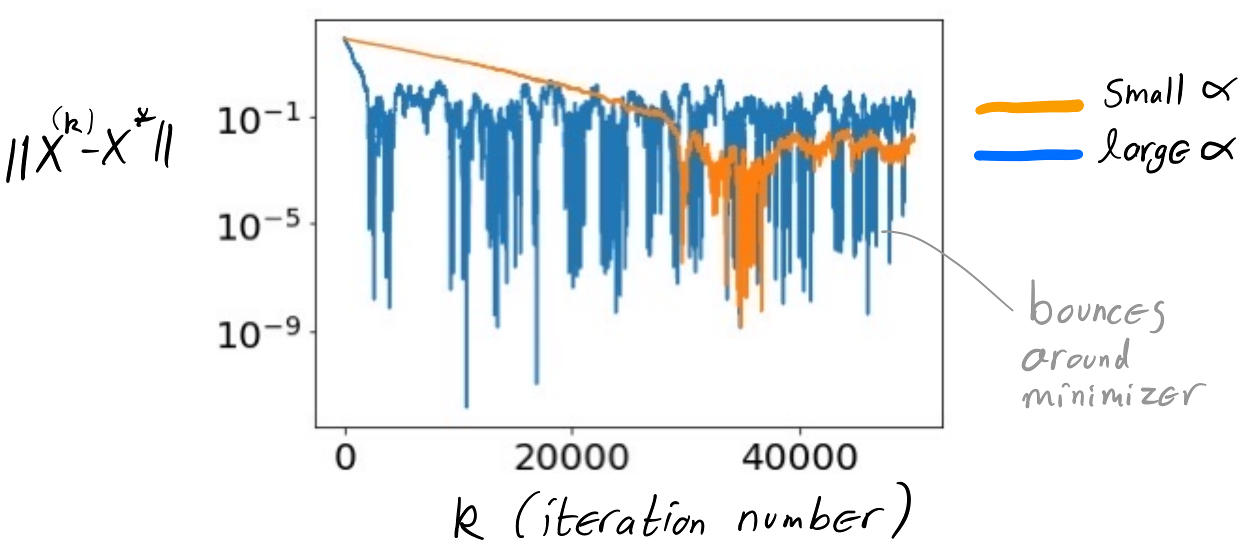
Use in ML: minibatches

$$f(\theta) = \frac{1}{n}\sum_{i=1}^{n} \ell(\hat{y}_\theta(x_i), y_i)$$

$$G(\theta) = \frac{1}{|B|}\sum_{i \in B} \nabla_\theta \ell(\hat{y}_\theta(x_i), y_i) \quad \text{for random subset } B$$

Qualitatively,

With fixed step size α, $x^{(k)}$ will move close to x^* but will bounce around due to stochasticity



large $\alpha \Rightarrow$ fast initial convergence
large error

Small $\alpha \Rightarrow$ slow initial convergence
smaller error

Can formalize these observations w/ theory

How to choose step sizes/learning rates?

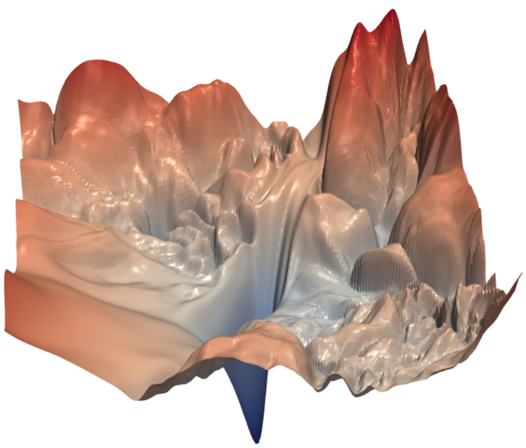
$\left\{\begin{array}{l}\text{Run at a large value for } \alpha \text{ while} \\ \text{Shrink learning rate} \\ \text{Repeat}\end{array}\right.$

$\{$ Have schedule of α_k decaying in k

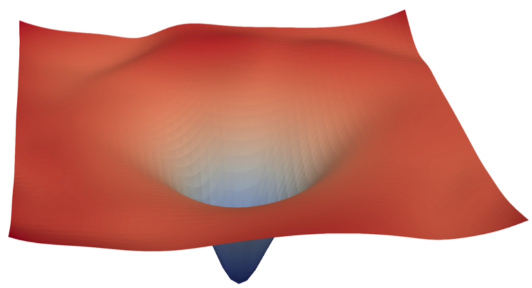
In these cases can hope for convergence

Challenges w/ GD and SGD in Deep Learning

Nonconvexity and nonsmoothness



(a) without skip connections



(b) with skip connections

Figure 1: The loss surfaces of ResNet-56 with/without skip connections. The proposed filter normalization scheme is used to enable comparisons of sharpness/flatness between the two figures.

(Li et al. 2018)

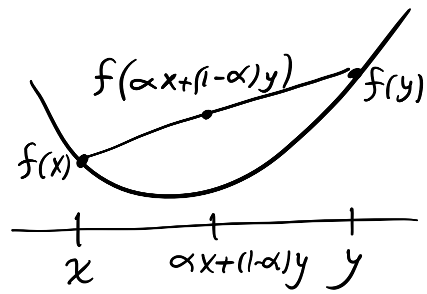
may be stuck in a local minimum,
so may want to temporarily increase
learning rate to get unstuck.

Convex Optimization

We say $f: \mathbb{R}^d \to \mathbb{R}$ is convex if

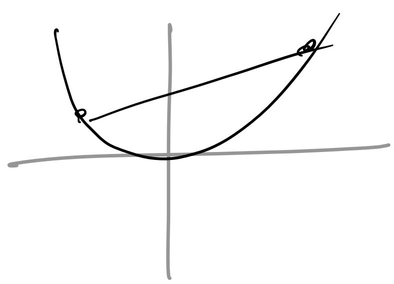
$$f(\underbrace{\alpha x+(1-\alpha)y}_{\text{Convex combo}}) \leq \alpha f(x) + (1-\alpha) f(y)$$

for all $0 \leq \alpha \leq 1$, x, y.



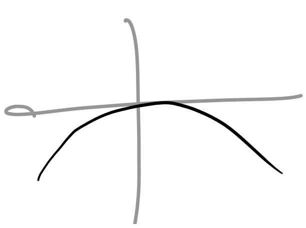
"always curves up"

Examples: $f: \mathbb{R} \to \mathbb{R}$
$f(x) = x^2$ is or is not convex



Fix a $c \in \mathbb{R}$.
$f: \mathbb{R} \to \mathbb{R}$
$f(x) = cx^2$ is or is not convex

If $c \geq 0$, yes
$c < 0$, no



$f: \mathbb{R} \to \mathbb{R}$
$f(x) = x$

is or is not convex

$f: \mathbb{R} \to \mathbb{R}$
$f(x) = |x|$

is or is not convex

$f: \mathbb{R}^2 \to \mathbb{R}$
$f(x) = \|x\|^2 = x_1^2 + x_2^2$

is or is not convex

$f: \mathbb{R}^2 \to \mathbb{R}$
$f(x) = x_1^2$

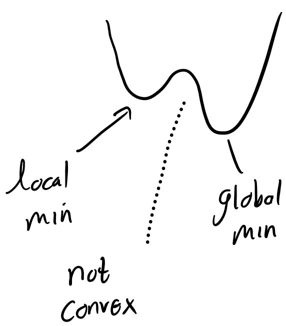
is or is not convex

We will study the minimization of convex functions.

Does every convex function f have a minimal value?

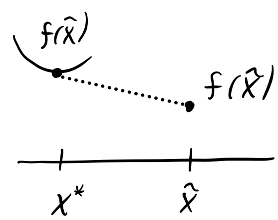
$$\min_{x} f(x)$$

All local minima of convex functions are global minima.



Suppose X_* is a local min of f.
If $X \approx X_*$ the $f(X) \geq f(X_*)$.

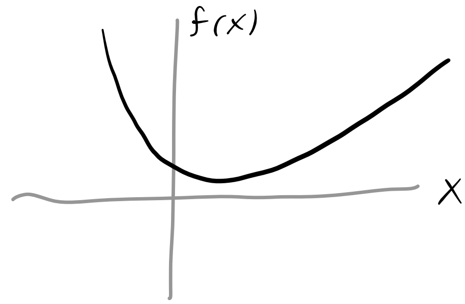
Suppose $f(\hat{X}) < f(X^*)$.
By convexity, $f(X)$ lies below dotted line between X^* and $\hat{X}$. So X^* not a local min.



Convexity and Second derivatives

Functions of one variable

If $f: \mathbb{R} \to \mathbb{R}$ is twice differentiable everywhere, f is convex if and only if $f''(x) \geq 0$ for all x.



Functions of multiple variables

Let $f: \mathbb{R}^n \to \mathbb{R}$

f is convex if $D^2 f = Hf$ is positive semidefinite everywhere

Hessian matrix

$$D^2 f = Hf(x) = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \cdots & \frac{\partial^2 f}{\partial x_n \partial x_1} \\ \vdots & & \\ \frac{\partial^2 f}{\partial x_1 \partial x_n} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{pmatrix}$$

H is positive definite if all eigenvalues are positive

H is positive semidefinite if all eigenvalues are nonnegative

Theorem: H is positive semidefinite if and only if

$$z^t H z \geq 0 \quad \text{for all } z \in \mathbb{R}^n$$

Recall, because H is symmetric $(H = H^t)$, H has an orthonormal basis of eigenvectors with real eigenvalues. So

$$H = U \Lambda U^t$$ where U has orthonormal columns

and Λ is diagonal

We say U_i is an eigenvector of H with eigenvalue λ_i if $H U_i = \lambda_i$

$$H = \begin{pmatrix} | & | & & | \\ U_1 & U_2 & \cdots & U_n \\ | & | & & | \end{pmatrix} \begin{pmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_n \end{pmatrix} \begin{pmatrix} - & U_1^t & - \\ - & U_2^t & - \\ & \vdots & \\ - & U_n^t & - \end{pmatrix}$$

$$U \cdot \Lambda \cdot U^t$$

Columns are
unit length eigenvectors
that are orthogonal to
each other

$$U_i \cdot U_j = \begin{cases} 1 & \text{if } i=j \\ 0 & \text{if } i \neq j \end{cases}$$

Proof of Theorem: • PSD $\Rightarrow$ $z^t H z \geq 0$ for all z

As H is PSD, Λ has nonneg. diagonal entries. So $z^t H z = z^t U \Lambda U^t z$

$$= \sum_{i=1}^{n} \Lambda_{ii} (U^t z)_i^2$$

$$\geq 0.$$

• $z^t H z \geq 0$ for all $z \Rightarrow H$ is PSD

Suppose H is not PSD. At least one eigenvalue is negative.

Suppose U_i is eigenvector w/ e-val $\lambda_i < 0$. Then let $z = U_i$.

$$z^t H z = U_i^t H U_i = \lambda_i U_i^t U_i < 0.$$

Many but not all ML optimization problems are convex.

How fast does gradient descent converge?

$$\min_{x} f(x), \qquad x^{(i+1)} = x^{(i)} - \alpha \nabla f(x^{(i)})$$

Suppose $x^{(i)} \to x^*$ as $i \to \infty$.

How long do you need to wait to get a certain accuracy ε?

Can gain understanding in some convex cases.

Convergence of GD for quadratic functions

Let $f(x) = \frac{1}{2}x^t Q x - b^t x$

where $x \in \mathbb{R}^d$, $b \in \mathbb{R}^d$, $Q \in \mathbb{R}^{d \times d}$ is positive definite

Let $m = \lambda_{min}(Q)$, $M = \lambda_{max}(Q)$, $K = \frac{M}{m}$

condition number of Q

Consider GD w/ fixed step size α

$$x^{k+1} = x^k - \alpha \nabla f(x^k)$$

Note: $x^* = Q^{-1}b$ is the unique global min of f

Analytically show that this is the solution to the problem

Theorem: If $\alpha = \frac{2}{M+m}$, then GD for $f(x) = \frac{1}{2}x^t Q x - b^t x$ satisfies

$$\|x^k - x^*\| \leq \left(\frac{1-1/K}{1+1/K}\right)^k \|x^0 - x^*\|$$

"first-order convergence"
Error decays exponentially
To get error ε, need $O(\log(\varepsilon^{-1}))$ iterations

Proof: Note $\nabla f(x) = Qx - b$.
The global minimizer solves $Qx^* = b \Rightarrow x^* = Q^{-1}b$

$$\begin{aligned}
x^{k+1} - x^* &= x^k - \alpha \nabla f(x^k) - x^* \\
&= x^k - \alpha(Qx^k - b) - x^* \\
&= x^k - \alpha(Qx^k - Qx^*) - x^* \\
&= (I - \alpha Q)(x^k - x^*)
\end{aligned}$$

So,

$$\|x^{k+1} - x^*\| \leq \underbrace{\|I - \alpha Q\|}_{\max(\alpha M - 1,\, 1 - \alpha m)} \|x^k - x^*\|$$

We choose $\alpha = \dfrac{2}{M+m}$.

So $\|I - \alpha Q\| = \dfrac{M-m}{M+m} = \dfrac{1-1/k}{1+1/k} < 1$

$$\Rightarrow \|x^{k+1} - x^*\| \leq \left(\frac{1-1/k}{1+1/k}\right) \|x^k - x^*\|$$

$$\Rightarrow \|x^k - x^*\| \leq \left(\frac{1-1/k}{1+1/k}\right)^k \|x^0 - x^*\| \quad \blacksquare$$

Interpretation:

If f doesnt curve up too much and doesnt curve up too little, then GD with fixed step size can exhibit first order convergence to the global minimizer

Should we think of GD as converging "quickly"?

Theorem: Let f be convex and $\lambda_{max}(Hf(x)) \leq M$ for all x. If $\alpha \leq \frac{1}{M}$, then GD satisfies

$$f(x^{(i)}) - f(x^*) \leq \frac{1}{2i\alpha} \|x^{(0)} - x^*\|^2$$

where x^* is a minimizer of f.

- Error decays slowly

- To get error ε from optimal value, need $O(\varepsilon^{-1})$ iterations

Summary:

- Too large learning rate can lead to divergence
- In convex case, to get convergence α should be small relative to curvature of f
- Too small learning rate can lead to slow convergence
- For convex quadratic functions, convergence of GD can be first order (fast)
- For more general convex functions, convergence can be slow
- SGD w/ fixed step size is not expected to converge
- SGD with decaying step sizes may converge