Non-Convex Optimization

CS6787 Lecture 7 — Fall 2018

Review — We've covered many methods

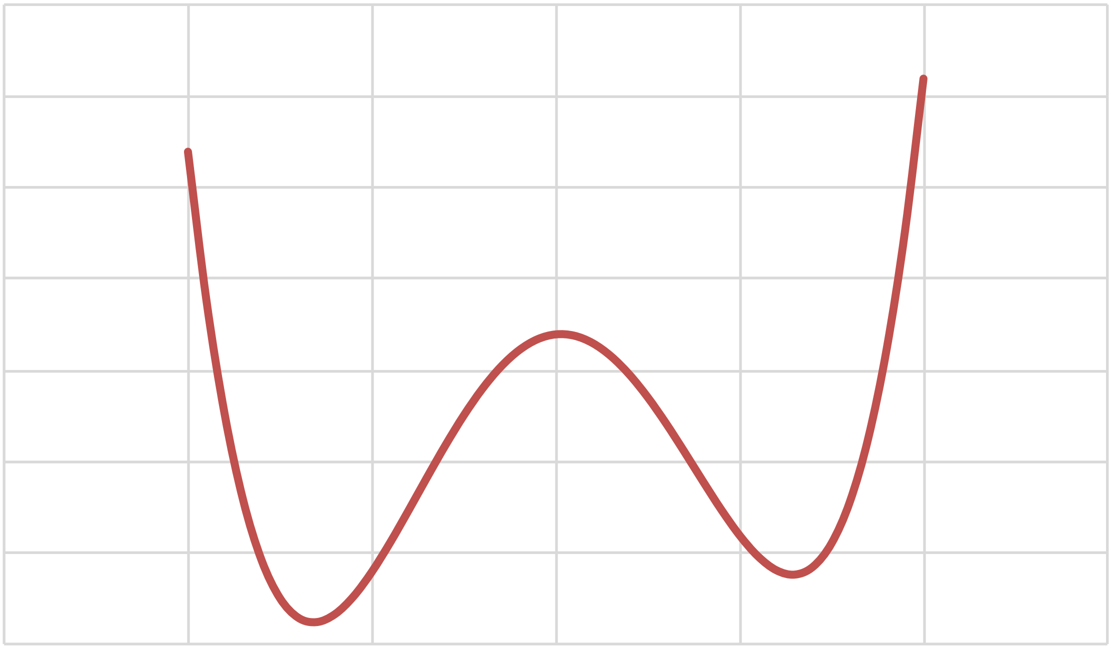
- Stochastic gradient descent
- Mini-batching
- Momentum
- Variance reduction

- **Nice convergence proofs that give us a rate**

- But **only for convex problems!**

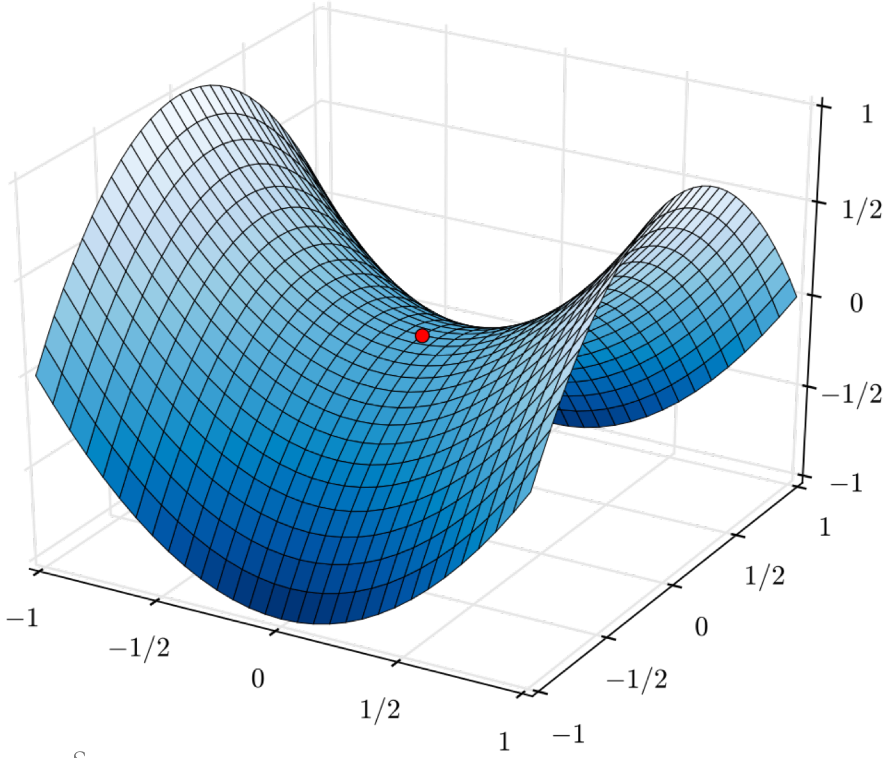
Non-Convex Problems

- Anything that's **not convex**



What makes non-convex optimization hard?

- Potentially many local minima
- Saddle points
- Very flat regions
- Widely varying curvature



Source:
https://commons.wikimedia.org/wiki/File:Saddle_point.svg

But is it actually that hard?

- Yes, many classes of non-convex optimization are at least **NP-hard**
  - Can encode most problems as non-convex optimization problems

- Example: subset sum problem
  - **Given a set of integers, is there a non-empty subset whose sum is zero?**
  - Known to be NP-complete.

- How do we encode this as an optimization problem?

Subset sum as non-convex optimization

- Let $\mathbf{a_1}$, $\mathbf{a_2}$, …, $\mathbf{a_n}$ be the input integers
- Let $\mathbf{x_1}$, $\mathbf{x_2}$, …, $\mathbf{x_n}$ be 1 if $\mathbf{a_i}$ is in the subset, and 0 otherwise
- Objective:

$$\text{minimize} \;\; \left(a^T x\right)^2 + \sum_{i=1}^{n} x_i^2 (1 - x_i)^2$$

- **What is the optimum if subset sum returns true? What if it's false?**

Non-convex problems can be harder than NP

- We can use the same trick to encode any **Diophantine equation**
  - Let **p** be an integer polynomial in **n** variables, then solve

$$\underset{x_1, x_2, \ldots, x_n \in \mathbb{R}}{\text{minimize}} \quad p^2(x_1, x_2, \ldots, x_n) + \sum_{i=1}^{n} \sin^2(\pi x_i)$$

- **What is the optimum if the Diophantine equation has a solution? What if it doesn't have a solution?**

- Matiyasevich theorem: Diophantine equations **undecidable in general**

So non-convex optimization is pretty hard

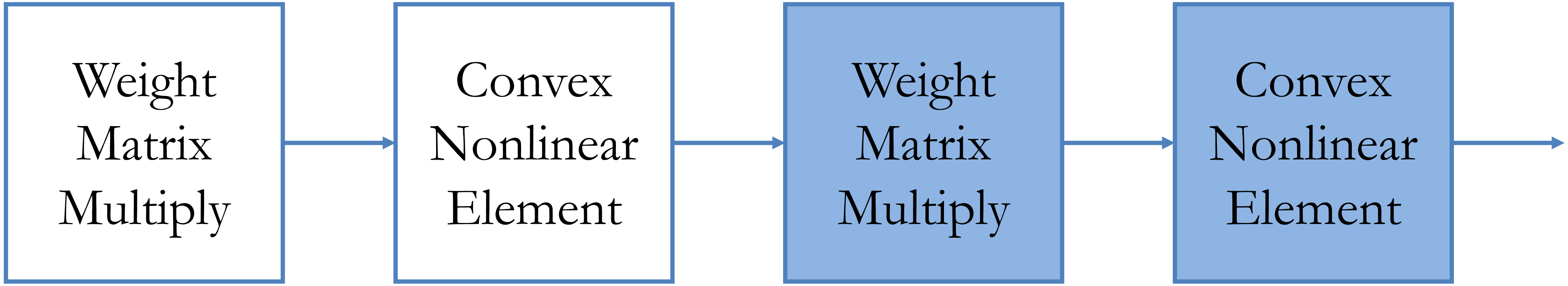
- There can't be a general algorithm to solve it efficiently in all cases
  - Or at all!
- Downsides: **theoretical guarantees are weak** or nonexistent
  - Depending on the application
  - There's usually no theoretical recipe for setting hyperparameters
- Upside: an endless array of **problems to try to solve better**
  - And gain theoretical insight about
  - And improve the performance of implementations

Examples of non-convex problems

- Matrix completion, principle component analysis
- Low-rank models and tensor decomposition
- Maximum likelihood estimation with hidden variables
  - Usually non-convex
- The big one: **deep neural networks**

Why are neural networks non-convex?

- They're often made of convex parts!
  - **This by itself would be convex.**



- **Composition of convex functions is not convex**
  - So deep neural networks also aren't convex

Why do neural nets need to be non-convex?

- Neural networks are **universal function approximators**
  - With enough neurons, they can learn to approximate any function arbitrarily well
- To do this, they need to be able to approximate non-convex functions
  - **Convex functions can't approximate non-convex** ones well.
- Neural nets also have many symmetric configurations
  - For example, exchanging intermediate neurons
  - This symmetry means they can't be convex. **Why?**

How to solve non-convex problems?

- Can use many of the **same techniques as before**
  - Stochastic gradient descent
  - Mini-batching
  - SVRG
  - Momentum

- There are also specialized methods for solving non-convex problems
  - Alternating minimization methods
  - Branch-and-bound methods
  - These generally **aren't very popular for machine learning** problems

Varieties of theoretical convergence results

- Convergence **to a stationary point**

- Convergence **to a local minimum**

- **Local convergence** to the global minimum

- **Global convergence** to the global minimum

Non-convex
Stochastic Gradient Descent

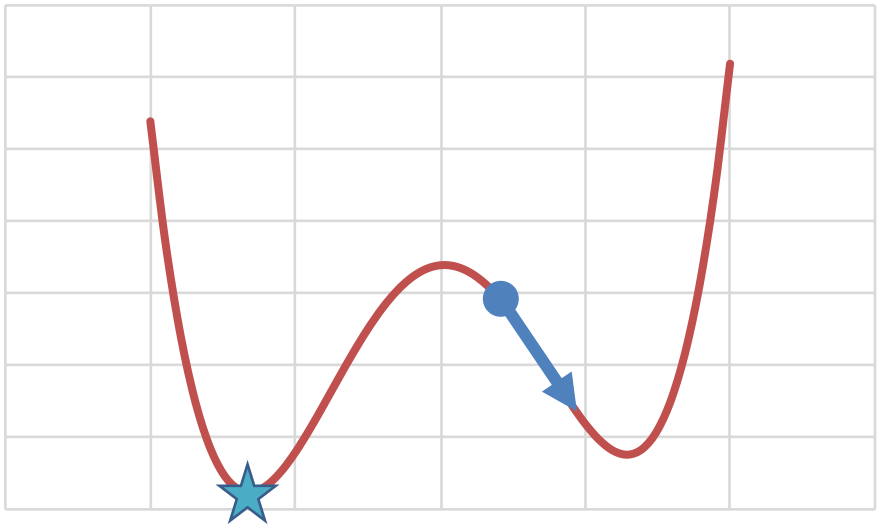
Stochastic Gradient Descent

- The update rule is the same for non-convex functions

$$w_{t+1} = w_t - \alpha_t \nabla \tilde{f}_t(w_t)$$

- Same intuition of **moving in a direction that lowers objective**

- **Doesn't necessarily go towards optimum**
  - Even in expectation



Non-convex SGD: A Systems Perspective

- It's **exactly the same as the convex case**!
- The **hardware doesn't care** whether our gradients are from a convex function or not
- This means that all our intuition about computational efficiency from the convex case directly applies to the non-convex case
- But **does our intuition about statistical efficiency also apply?**

When can we say SGD converges?

- First, we need to decide what type of convergence we want to show
  - Here I'll just show **convergence to a stationary point**, the weakest type
- Assumptions:
  - Second-differentiable objective

$$-LI \preceq \nabla^2 f(x) \preceq LI$$

  - Lipschitz-continuous gradients
  - Noise has bounded variance

$$\mathbf{E}\left[\left\|\nabla \tilde{f}_t(x) - f(x)\right\|^2\right] \leq \sigma^2$$

  - But **no convexity assumption**!

Convergence of Non-Convex SGD

- Start with the update rule:

$$w_{t+1} = w_t - \alpha_t \nabla \tilde{f}_t(w_t)$$

- At the next time step, by Taylor's theorem, the objective will be

$$\begin{aligned} f(w_{t+1}) &= f(w_t - \alpha_t \nabla \tilde{f}_t(w_t)) \\ &= f(w_t) - \alpha_t \nabla \tilde{f}_t(w_t)^T \nabla f(w_t) + \frac{\alpha_t^2}{2} \nabla \tilde{f}_t(w_t)^T \nabla^2 f(y_t) \nabla \tilde{f}_t(w_t) \\ &\leq f(w_t) - \alpha_t \nabla \tilde{f}_t(w_t)^T \nabla f(w_t) + \frac{\alpha_t^2 L}{2} \left\| \nabla \tilde{f}_t(w_t) \right\|^2 \end{aligned}$$

Convergence (continued)

- Taking the expected value

$$\mathbf{E}\left[f(w_{t+1}) | w_t\right] \le f(w_t) - \alpha_t \mathbf{E}\left[\nabla \tilde{f}_t(w_t)^T \nabla f(w_t) \middle| w_t\right] + \frac{\alpha_t^2 L}{2} \mathbf{E}\left[\left\| \nabla \tilde{f}_t(w_t) \right\|^2 \middle| w_t\right]$$

$$= f(w_t) - \alpha_t \left\| \nabla f(w_t) \right\|^2 + \frac{\alpha_t^2 L}{2} \mathbf{E}\left[\left\| \nabla \tilde{f}_t(w_t) \right\|^2 \middle| w_t\right]$$

$$= f(w_t) - \alpha_t \left\| \nabla f(w_t) \right\|^2 + \frac{\alpha_t^2 L}{2} \left\| \nabla f(w_t) \right\|^2$$

$$+ \frac{\alpha_t^2 L}{2} \mathbf{E}\left[\left\| \nabla \tilde{f}_t(w_t) - \nabla f(w_t) \right\|^2 \middle| w_t\right]$$

$$\le f(w_t) - \left(\alpha_t - \frac{\alpha_t^2 L}{2}\right) \left\| \nabla f(w_t) \right\|^2 + \frac{\alpha_t^2 \sigma^2 L}{2}.$$

Convergence (continued)

- So now we know how the expected value of the objective evolves.

$$\mathbf{E}\left[f(w_{t+1}) | w_t\right] \le f(w_t) - \left(\alpha_t - \frac{\alpha_t^2 L}{2}\right) \left\| \nabla f(w_t) \right\|^2 + \frac{\alpha_t^2 \sigma^2 L}{2}.$$

- If we set α small enough that 1 - αL/2 > 1/2, then

$$\mathbf{E}\left[f(w_{t+1}) | w_t\right] \le f(w_t) - \frac{\alpha_t}{2} \left\| \nabla f(w_t) \right\|^2 + \frac{\alpha_t^2 \sigma^2 L}{2}.$$

Convergence (continued)

- Now taking the full expectation,

$$\mathbf{E}\left[f(w_{t+1})\right] \le \mathbf{E}\left[f(w_t)\right] - \frac{\alpha_t}{2}\mathbf{E}\left[\left\|\nabla f(w_t)\right\|^2\right] + \frac{\alpha_t^2 \sigma^2 L}{2}.$$

- And summing up over an epoch of length **T**

$$\mathbf{E}\left[f(w_T)\right] \le f(w_0) - \sum_{t=0}^{T-1} \frac{\alpha_t}{2}\mathbf{E}\left[\left\|\nabla f(w_t)\right\|^2\right] + \sum_{t=0}^{T-1} \frac{\alpha_t^2 \sigma^2 L}{2}.$$

Convergence (continued)

- Now we need to decide how to set the step size $\boldsymbol{\alpha_t}$
  - Let's just **set it to be decreasing** like we did in the convex setting

$$\alpha_t = \frac{\alpha_0}{t+1}$$

- So our bound on the objective becomes

$$\mathbf{E}\left[f(w_T)\right] \le f(w_0) - \sum_{t=0}^{T-1} \frac{\alpha_0}{2(t+1)} \mathbf{E}\left[\|\nabla f(w_t)\|^2\right] + \sum_{t=0}^{T-1} \frac{\alpha_0^2 \sigma^2 L}{2(t+1)^2}.$$

Convergence (continued)

- Rearranging the terms,

$$\sum_{t=0}^{T-1} \frac{\alpha_0}{2(t+1)} \mathbf{E}\left[\|\nabla f(w_t)\|^2\right] \le f(w_0) - \mathbf{E}\left[f(w_T)\right] + \sum_{t=0}^{T-1} \frac{\alpha_0^2 \sigma^2 L}{2(t+1)^2}$$

$$\le f(w_0) - f(w^*) + \frac{\alpha_0^2 \sigma^2 L}{2} \sum_{t=0}^{\infty} \frac{1}{(t+1)^2}$$

$$\le f(w_0) - f(w^*) + \frac{\alpha_0^2 \sigma^2 L}{2} \cdot \frac{\pi^2}{6}.$$

Now, we're kinda stuck

- How do we use the bound on this term to say something useful?

$$\sum_{t=0}^{T-1} \frac{\alpha_0}{2(t+1)} \mathbf{E}\left[\|\nabla f(w_t)\|^2 \right]$$

- Idea: rather than outputting $\mathbf{w_T}$, instead output some randomly chosen $\mathbf{w_i}$ from the history.
  - You might recall this trick from the proof in the SVRG paper.

Let $z_T = w_t$ with probability $\frac{1}{H_T(t+1)}$, where $H_t = \sum_{t=0}^{T-1} \frac{1}{t+1}$

Using our randomly chosen output

Let $z_T = w_t$ with probability $\frac{1}{H_T(t+1)}$, where $H_t = \sum_{t=0}^{T-1} \frac{1}{t+1}$

- So the expected value of the gradient at this point is

$$\mathbf{E}\left[\|\nabla f(z_T)\|^2\right] = \sum_{t=0}^{T-1} \mathbf{P}\left(z_T = w_t\right) \cdot \mathbf{E}\left[\|\nabla f(w_t)\|^2\right]$$

$$= \sum_{t=0}^{T-1} \frac{1}{H_T(t+1)} \mathbf{E}\left[\|\nabla f(w_t)\|^2\right].$$

Convergence (continued)

- Now we can continue to bound this with

$$\mathbf{E}\left[\|\nabla f(z_T)\|^2\right] = \frac{2}{\alpha_0 H_T} \sum_{t=0}^{T-1} \frac{\alpha_0}{2(t+1)} \mathbf{E}\left[\|\nabla f(w_t)\|^2\right]$$

$$\leq \frac{2}{\alpha_0 H_T} \left(f(w_0) - f(w^*) + \frac{\pi^2 \alpha_0^2 \sigma^2 L}{12} \right)$$

$$\leq \frac{2}{\alpha_0 \log T} \left(f(w_0) - f(w^*) + \frac{\pi^2 \alpha_0^2 \sigma^2 L}{12} \right)$$

Convergence (continued)

- This means that for some fixed constant **C**

$$\mathbf{E}\left[\|\nabla f(z_T)\|^2\right] \leq \frac{C}{\log T}$$

- And so in the limit

$$\lim_{T \to \infty} \mathbf{E}\left[\|\nabla f(z_T)\|^2\right] = 0.$$

Convergence Takeaways

- So even **non-convex SGD converges!**
  - In the sense of getting to points where the **gradient is arbitrarily small**
- But this **doesn't mean it goes to a local minimum**!
  - Doesn't rule out that it goes to a saddle point, or a local maximum.
  - Doesn't rule out that it goes to a region of very flat but nonzero gradients.
- Certainly **doesn't mean that it finds the global optimum**
- And the theoretical **rate here was really slow**

Strengthening these theoretical results

Convergence to a local minimum

- Under stronger conditions, can prove that SGD **converges to a local minimum**
  - For example using the **strict saddle property** (Ge et al 2015)

- Using even stronger properties, can prove that **SGD converges to a local minimum with an explicit convergence rate** of **1/T**

- **But, it's unclear whether common classes of non-convex problems, such as neural nets, actually satisfy these stronger conditions.**

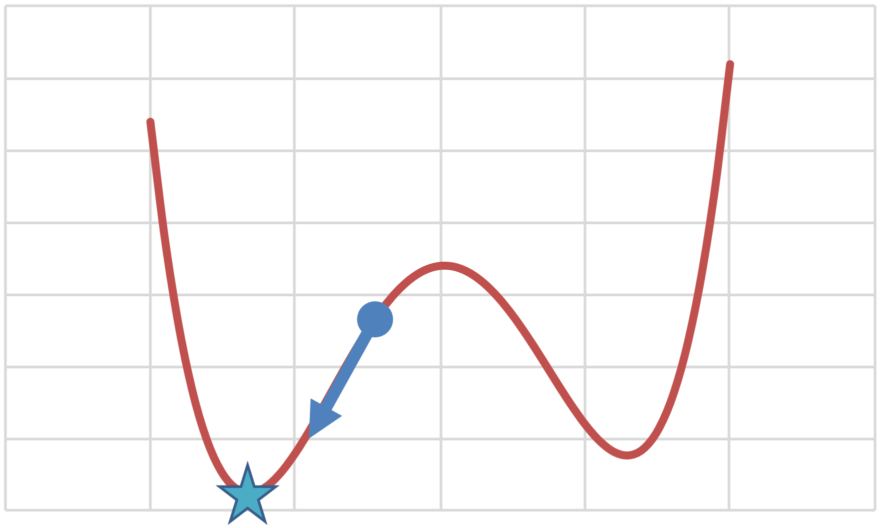
Strengthening these theoretical results

Local convergence to the global minimum

- Another type of result you'll see are **local convergence results**

- Main idea: if we **start close enough to the global optimum**, we will converge there with high probability

- Results often give **explicit initialization scheme** that is guaranteed to be close
  - But it's often **expensive to run**
  - And **limited to specific problems**



Strengthening these theoretical results

Global convergence to a global minimum

- The strongest result is **convergence no matter where we initialize**
  - Like in the convex case
- To prove this, we need a **global understanding of the objective**
  - So it can only apply to a **limited class of problems**
- For many problems, we **know empirically that this doesn't happen**
  - Deep neural networks are an example of this

Other types of results

- Bounds on generalization error
  - Roughly: we can't say it'll converge, but we can say that it won't overfit
- Ruling out "spurious local minima"
  - Minima that **exist in the training loss, but not in the true/test loss**.
- Results that use the Hessian to escape from saddle points
  - By using it to find a descent direction, but rarely enough that it doesn't damage the computational efficiency

Questions?

- Upcoming things
  - Paper review #5a or #5b **due Today**
  - Final Project Proposal due **on Monday**

Project Idea Discussion

Things to discuss and give feedback on

- What is interesting about the proposed project?
  - What other interesting things could be done connected with the project?
- What technique(s) discussed in the course does the project explore?
- What experiments are planned?
  - What other experiments would make sense to run?
  - Is the project feasible to complete in the available time?