CS229M/STATS214: Machine Learning Theory

Lecturer: Tengyu Ma
Scribe: Kevin Han and Han Wu

Lecture # 10
Feb 17th, 2021

10.1 Review and overview

Last lecture, we outlined conceptual topics in deep learning theory and how the situation was different from classical machine learning theory. In particular, we described *approximation theory*, *statistical generalization* and *optimization*. In this lecture, we will focus on optimization theory in deep learning. We will introduce some basics about optimization, discuss how we can make the notion “all local minimums are global minimums” rigorous, and walk through two examples where this is the case.

10.2 Optimization landscape

The big question that we have in mind is the following: many existing optimizers are designed for optimizing convex functions. **Why do they still work well for non-convex functions?** We note that it is not true that these optimizers always work well with non-convex functions: there are still some very hard cases that give trouble (e.g. very deep feed-forward networks are still hard to fit because of issues like vanishing and exploding gradients). One possible reason is that the non-convex functions that we are minimizing in deep learning usually have some nice properties: see Figure 10.1 for an illustration.

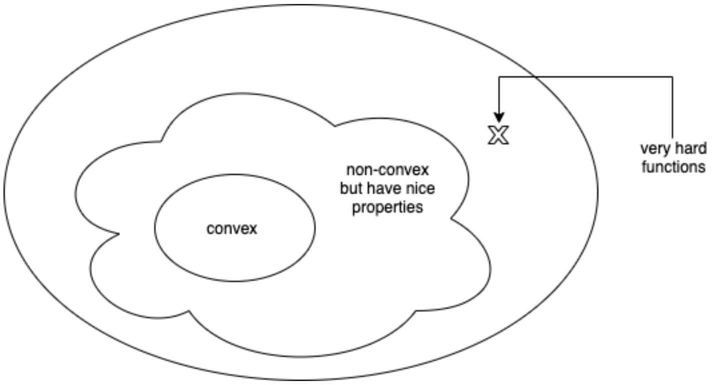

Figure 10.1: Classification of different functions for optimization. The functions we optimize in deep learning seem to fall mostly within the middle cloud.

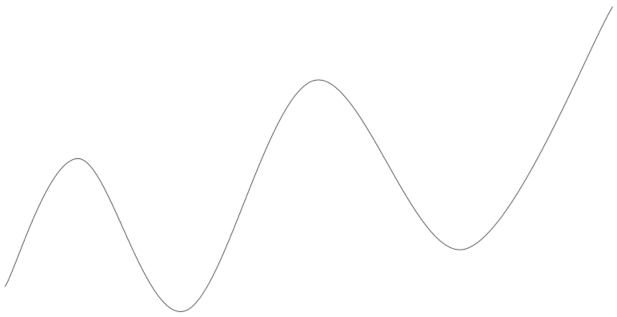

Figure 10.2: Illustration of a general non-convex function. A useful plot to keep in mind when thinking about non-convex functions.

Before diving into details, we first highlight some observations that will be important to keep in mind when discussing optimization in deep learning. Suppose $g(\theta)$ is the loss function. Recall that the *gradient descent (GD)* algorithm would do the following:

1. $\theta_0 :=$ initialization

2. $\theta_{t+1} = \theta_t - \eta \nabla g(\theta_t)$, where η is the step size.

Here are some observations to :

> *Observation 1*: Gradient descent cannot always find the global minimum.
>
> *Observation 2*: Finding the global minimum of general non-convex functions is NP-hard.
>
> *Observation 3*: Gradient descent can find the global minimum for convex functions.
>
> *Observation 4*: The objective function in deep learning is non-convex.
>
> *Observation 5*: Gradient descent/stochastic gradient descent finds an approximate global minimum of loss function in deep learning.

These observations motivate the following two-step plan:

1. Identify a large set of functions that stochastic gradient descent/gradient descent can solve.

2. Prove that some of the loss functions in machine learning problems belong to this set. (Most of the effort will be spent here.)

Basic idea: Gradient descent can find local minimum + all local minimums of f are also global $\Rightarrow$ Gradient descent can find global minimums.

10.3 Convergence to local minimum

Let f be a twice-differentiable function. We start with the following definition:

Definition 10.1 (Local minimum of a function)**.** We say that x is a *local minimum* of a function f if there exists an open neighborhood N around x such that in N, the function values are at least $f(x)$.

Note that if x is a local minimum of f, then $\nabla f(x) = 0$ and $\nabla^2 f(x) \succeq 0$. However, as the next example shows, the reverse is not true. When $\nabla f(x) = 0$ and $\nabla^2 f(x)$ vanishes in some direction (i.e. merely positive semi-definite instead of being strictly positive definite), higher-order derivatives start to matter.

Example 10.2. Consider the function $f(x_1, x_2) = x_1^2 + x_2^3$. $(x_1, x_2) = (0, 0)$ satisfies $\nabla f(x) = 0$ and $\nabla^2 f(x)|_{(x_1,x_2)=(0,0)} = \begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix} \succeq 0$. However, if we move in the negative direction of x_2, we can decrease the function value. Hence, this example shows why $\nabla f(x) = 0$ and $\nabla^2 f(x) \succeq 0$ does not imply that x is a local minimum.

It is generally not easy to verify if a point is a local minimum. In fact, we have the following theorem regarding the computational tractability:

Theorem 10.3. *Verifying if x is a local minimum of f is NP-hard. Moreover, finding a local minimum is NP-hard.*

10.3.1 Strict-saddle condition

Theorem 10.3 forces us to consider more specific types of functions to be able to obtain computational tractability. To this end, we define the following *strict-saddle condition*:

Definition 10.4 (Strict-saddle condition ([LSJR16]; [GHJY15]))**.** For positive α, β, γ, we say that $f : \mathbb{R}^d \mapsto \mathbb{R}$ is (α, β, γ)-*strict-saddle* if every $x \in \mathbb{R}^d$ satisfies one of the following:

1. $\|\nabla f(x)\|_2 \geq \alpha$.
2. $\lambda_{\min}(\nabla^2 f(x)) \leq -\beta$.
3. x is γ-close to a local minimum x^* in Euclidean distance, i.e. $\|x - x^*\|_2 \leq \gamma$.

Intuitively speaking, this definition is saying if a point has zero gradient and positive semi-definite Hessian, it must be close to a local minimum, i.e. there is no pathological case like Example 10.2.

We have the following theorem for functions that satisfy strict-saddle condition:

Theorem 10.5 (Informally stated)**.** *If f is (α, β, γ)-strict-saddle for some positive α, β, γ, then many optimizers (e.g. gradient descent, stochastic gradient descent, cubic regularization) can converge to a local minimum with ϵ-error in Euclidean distance in time poly $\left(d, \frac{1}{\alpha}, \frac{1}{\beta}, \frac{1}{\gamma}, \frac{1}{\epsilon}\right)$.*

Therefore, if all local minimums are global minimums and the function satisfies the strict-saddle condition, then optimizers can converge to a global minimum with ϵ-error in polynomial time. (See Figure 10.3 for an example of a function whose local minimums are all global minimums.) The next theorem expresses this concretely by being explicit about the strict-saddle condition:

Theorem 10.6. *Suppose f is a function that satisfies the following condition: $\exists\ \epsilon_0, \tau_0, c > 0$ such that if $x \in \mathbb{R}^d$ satisfies $\|\nabla f(x)\|_2 \le \epsilon < \epsilon_0$ and $\nabla^2 f(x) \succeq -\tau_0 I$, then x is ϵ^c-close to a global minimum of f. Then many optimizers can converge to a global minimum of f up to δ-error in Euclidean distance in time poly $\left(\frac{1}{\delta}, \frac{1}{\tau_0}, d\right)$.*

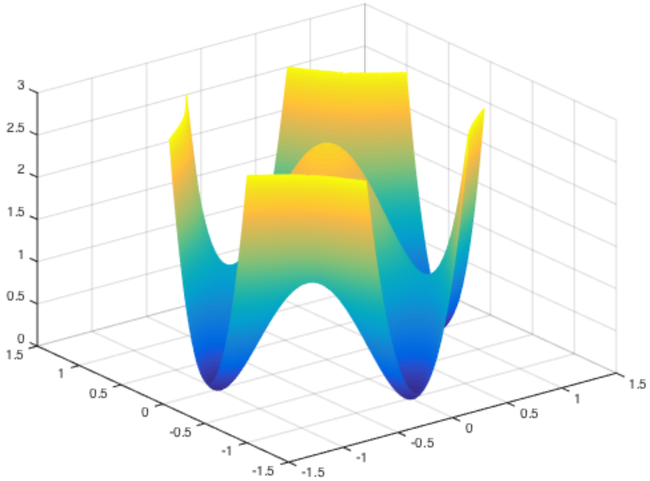

Figure 10.3: A two-dimensional function with the property that all local minimums are global minimums. It also satisfies the strict-saddle condition because all the saddle points have a strictly negative curvature in some direction.

10.4 Two examples where local minimums are global minimums

So far, we have focused on general results. Next, we give two concrete examples that satisfies all local minimums are global minimums:(i) principal components analysis (PCA)/matrix factorization/linearized neural nets, and (ii) matrix completion.

10.4.1 Principal components analysis (PCA)

Let matrix $M \in \mathbb{R}^{d\times d}$ be symmetric and positive semi-definite. Consider the problem of finding the best rank-1 approximation of the matrix M. The objective function here is non-convex:

$$\min_{x\in\mathbb{R}^d} g(x) \triangleq \frac{1}{2}\|M - xx^T\|_F^2. \tag{10.1}$$

Theorem 10.7. *All local minimums of g are global minimums (even though g is non-convex).*

Remark 10.8. For $d = 1$, $g(x) = (m - x^2)^2$ for some constant m. Figure 10.4 below shows such an example. We can see that all local minimums are indeed global minimums.

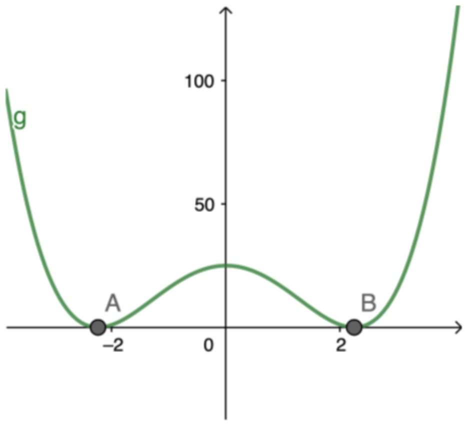

Figure 10.4: Objective function for principal components analysis (PCA) when $d = 1$.

Proof. Step 1: Show that all stationary points must be eigenvectors. From HW0, we know that $\nabla g(x) = -(M - xx^T)x$, hence

$$\nabla g(x) = 0 \implies Mx = \|x\|_2^2 \cdot x, \tag{10.2}$$

which implies that x is an eigenvector of M with eigenvalue $\|x\|_2^2$. From the Eckart–Young–Mirsky theorem we know the global minimum (i.e. the best rank-1 approximation) is the eigenvector with the largest eigenvalue.

Step 2: Show that all local minimums must be eigenvectors of the largest eigenvalue. We use the second order condition for this. For x to be a local minimum we need $\nabla^2 g(x) \succeq 0$, which means for any $v \in \mathbb{R}^d$,

$$\langle v, \nabla^2 g(x) v \rangle \geq 0. \tag{10.3}$$

To compute $\langle v, \nabla^2 g(x) v \rangle$, we use the following trick: expand $g(x+v)$ into $g(x) +$ linear term in $v +$ quadratic term in v, then the quadratic term will be $\langle v, \nabla^2 g(x) v \rangle$ (see HW0 Problem 2d for an example). Using this trick, we get

$$\langle v, \nabla^2 g(x) v \rangle = 2\langle x, v \rangle^2 - v^T M v + \|x\|_2^2 \|v\|_2^2. \tag{10.4}$$

Picking $v = v_1$, the unit eigenvector with the largest eigenvalue (denoted λ_1), for x to be a local minimum it must satisfy

$$\langle v_1, \nabla^2 g(x) v_1 \rangle = 2\langle x, v_1 \rangle^2 - v_1^T M v_1 + \|x\|_2^2 \geq 0. \tag{10.5}$$

Note that by (10.2), all our candidates for local minimums are eigenvectors of M so naturally we have two cases:

- *Case 1: x has eigenvalue* λ_1. Then x is the global minimum (by the Eckart–Young–Mirsky theorem).

- *Case 2: x has eigenvalue $\lambda < \lambda_1$.* Then we know x and v_1 are orthogonal (eigenvectors with different eigenvalues are always orthogonal), hence

$$2\langle x, v_1\rangle^2 - v_1^T M v_1 + \|x\|_2^2 = 0 - \lambda_1 + \lambda \geq 0, \tag{10.6}$$

  which implies $\lambda \geq \lambda_1$, a contradiction.

In summary, if x is a stationary point and x is not a global minimum, then moving in the direction of v_1 would lead to second-order improvement and x cannot be a local minimum. □

10.4.2 Matrix Completion [GLM16]

We consider rank-1 matrix completion for simplicity. Let $M = zz^T$ be a rank-1 symmetric and positive semi-definite matrix for some $z \in \mathbb{R}^d$. Given random entries of M, our goal is to recover the rest of entries. Formally, we have the following definitions:

Definition 10.9. Suppose $M \in \mathbb{R}^{d\times d}$ and $\Omega \subseteq [d] \times [d]$, we define $P_\Omega(M)$ to be the matrix obtained by zeroing out every entry outside Ω.

Definition 10.10 (Matrix Completion)**.** Suppose $M \in \mathbb{R}^{d\times d}$ and every entry of M is included in Ω with probability p. The *matrix completion task* is to recover M (with respect to some loss functions) given the observation $P_\Omega(M)$.

A nice real world example of matrix completion is when we have a matrix describing the user ratings for each item. We only observe a small portion of the entries as each customer only buys a small subset of the items. A good matrix completion algorithm is indispensable for a recommendation engine.

Remark 10.11. We need d parameters to describe a rank-1 matrix M and the number of observations is roughly pd^2. Thus, for identifiability we need to work in the regime where $pd^2 > d$, i.e. $p \gg \frac{1}{d}$.

We define our non-convex loss functions to be

$$\min_{x\in\mathbb{R}^d} g(x) \triangleq \frac{1}{2} \sum_{(i,j)\in\Omega} (M_{ij} - x_i x_j)^2 \tag{10.7}$$

$$= \frac{1}{2}\|P_\Omega(M - xx^T)\|_F^2. \tag{10.8}$$

To really solve our problem we need some regularity condition on the ground truth vector z (recall $M = zz^T$). *Incoherence* is one such condition:

Definition 10.12 (Incoherence)**.** Without loss of generality, assume the ground truth vector $z \in \mathbb{R}^d$ satisfies $\|z\|_2 = 1$. z satisfies the *incoherence condition* if $\|z\|_\infty \leq \frac{\mu}{\sqrt{d}}$, where μ is considered to be a constant or log in dimension d.

Remark 10.13. A nice counterexample to think about why such condition is necessary is when $z = \mathbf{e_1}$ and $M = \mathbf{e_1}\mathbf{e_1}^T$. All entries of M are 0 except for a 1 in the top-left corner. There is no way to recover M without observing the top-left corner.

We conclude with a theorem which we will prove next lecture:

Theorem 10.14. *Suppose $p = \frac{poly(\mu, \log d)}{d\epsilon}$, $\epsilon > 0$ and assume z is incoherent. Then all local minimums of f are $O(\sqrt{\epsilon})$-close to either z or $-z$ (i.e. the global minimums), and we also have the strict saddle condition.*

Bibliography

[GHJY15] Rong Ge, Furong Huang, Chi Jin, and Yang Yuan, *Escaping from saddle points — online stochastic gradient for tensor decomposition*, Proceedings of The 28th Conference on Learning Theory (Paris, France) (Peter Grünwald, Elad Hazan, and Satyen Kale, eds.), Proceedings of Machine Learning Research, vol. 40, PMLR, 03–06 Jul 2015, pp. 797–842.

[GLM16] Rong Ge, Jason D Lee, and Tengyu Ma, *Matrix completion has no spurious local minimum*, Advances in Neural Information Processing Systems (D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, eds.), vol. 29, Curran Associates, Inc., 2016.

[LSJR16] Jason D. Lee, Max Simchowitz, Michael I. Jordan, and Benjamin Recht, *Gradient descent only converges to minimizers*, 29th Annual Conference on Learning Theory (Columbia University, New York, New York, USA) (Vitaly Feldman, Alexander Rakhlin, and Ohad Shamir, eds.), Proceedings of Machine Learning Research, vol. 49, PMLR, 23–26 Jun 2016, pp. 1246–1257.