Topics in Deep Learning Theory (Spring 2025)

Lecture 11: How GD/SGD Converges to Flat Minima

Instructor: Lei Wu Scribe: Zilin Wang

1 Introduction

So far we have discussed the stability condition for SGD, e.g. $\mathrm{Tr} \leq 2/\eta$. But it remains unclear how SGD evolves to find these stable global minima. This question is about dynamics.

Generally there are two ways of studying dynamics. One is on toy models, e.g. linear regression, random feature model, linear network. In these settings the analysis can be thorough but hard to extend to more complex settings. What we want to talk about here is some "realistic" argument that can also apply in real settings when there is not so strong assumptions.

In realistic settings it is nearly impossible to study the global dynamics. So we turn to local dynamics, such as

- Near initialization, NTK and Kaiming initialization, the effect of learning rate warm up, etc.
- Near convergence, what is SGD's behavior when it is close to global minima.

2 Initialization

Consider the two-layer network

$$f(x) = \sum_{j=1}^{m} a_j \sigma(\mathbf{w}_j^{\mathrm{T}} \mathbf{x}),$$

initialized as

$$a_j \sim \mathcal{N}(0,1), \mathbf{w}_j \sim \mathcal{N}(0, I_d/d).$$

Is this initialization good?

Note that

$$\|G(\theta)\|_2^2 \sim \sum_{j=1}^{m} (a_j^2 + \|\mathbf{w}_j\|^2) \sim 2m$$

for $x \sim \mathrm{Unif}(\sqrt{d}\mathbb{S}^{d-1})$. In order to prevent the training from blowing up, we need to set $\eta = O(\frac{1}{\sqrt{m}})$. This is not scalable since the learning rate tuned well on small nets cannot be applied to larger nets.

To fix this problem, we can choose to

- Use a different initialization, $a_j \sim \mathcal{N}(0, 1/m), \mathbf{w}_j \sim \mathcal{N}(0, I_d/(md))$, such that $\|G(\theta)\|_2 = O(1)$.

- Keep the initialization, but now the model is

$$f(x) = \frac{1}{m}\sum_{j=1}^{m} a_j \sigma(\mathbf{w}_j^{\mathrm{T}}\mathbf{x}),$$

  The limit is $\int a\sigma(\mathbf{w}^{\mathrm{T}}\mathbf{x})\,\mathrm{d}\pi(a,\mathbf{w})$ as $m \to \infty$, thus scalable.

An example from ResNet Consider the ResNet

$$\begin{cases} x^{\ell+1} = x^{\ell} + \dfrac{1}{m}\displaystyle\sum_{j=1}^{m} a_j \sigma(\mathbf{w}_j^{\mathrm{T}}\mathbf{x}^{\ell}) \\ x^0 = x \end{cases}$$

and $f(x;\theta) = \phi^{\mathrm{T}} x^L$. What happens when $L \to \infty$?

An intuitive way is to think

$$x^{\ell+1} = (1+\alpha)x^{\ell} = \cdots = (1+\alpha)^{\ell+1} x$$

where α is a constant. So, the model will blow up as $L \to \infty$.

In practice, batch normalization solves it. But without BN, it is important to modify the structure to make it consistent for L.

One approach:

$$\begin{cases} x^{\ell+1} = x^{\ell} + \dfrac{1}{mL}\displaystyle\sum_{j=1}^{m} a_j \sigma(\mathbf{w}_j^{\mathrm{T}}\mathbf{x}^{\ell}) \\ x^0 = x \end{cases}$$

so that

$$x^L = \left(1 + \frac{\alpha}{L}\right) x^{L-1} = \cdots = \left(1 + \frac{\alpha}{L}\right)^L x$$

is finite.

3 Local dynamics near global minima

Now, we start the discussion on local dynamics near global minima.

First let's see a toy example to gain some intuition. Consider the optimization with objective function $f(x,y) = (xy-1)^2$. Figure 1 shows the GD trajectory. Although initialized near global minima, GD does not converge to a minimizer closest to the initial point, but a flatter one which satisfies the stability condition. In this process,

- In the direction vertical to the global minima manifold, the trajectory oscillates.
- In the direction parallel to the global minima manifold, the trajectory moves slowly (relative to the oscillation) towards a flatter region.

The movement towards flatter regions are slow because

- The gradient is almost in the normal space (vertical to global minima). Let $\mathbf{g} = \mathbf{g}^{\perp} = \mathbf{g}^{\parallel}$, we have $\|\mathbf{g}^{\perp}\| \gg \|\mathbf{g}^{\parallel}\|$. The learning rate η cannot be too large otherwise $\eta\mathbf{g}^{\perp}$ blows up. So, $\eta\mathbf{g}^{\parallel}$ is small.

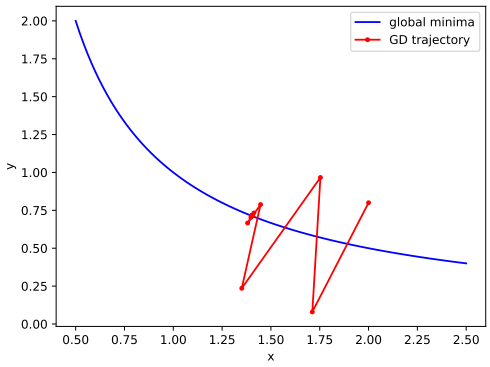

Figure 1: GD trajectory of minimizing $(xy-1)^2$.

- The distance it needs to go through is $O(1)$.

Then we introduce a model to formulate this intuition.

Let $f(x,y) = \frac{1}{2}y^{\mathrm{T}}H(x)y$ where $x \in \mathbb{R}^m, y \in \mathbb{R}^{p-m}$. Let $n = p-m$. Assume $\lambda_{\min}(H(x)) > 0$. It is easy to see the global minima $M = \{(x,y) : y = 0\}$.

The GD update gives

$$\begin{cases} x_{t+1} = x_t - \dfrac{\eta}{2}\displaystyle\sum_{i,j=1}^{n} \nabla H_{ij}(x) y_t(i) y_t(j) \\ y_{t+1} = y_t - \eta H(x) y_t \end{cases}$$

The Hessian

$$\nabla^2 f = \begin{pmatrix} \frac{1}{2}\nabla^2 H[y,y] & \nabla H(x)y \\ y^{\mathrm{T}}\nabla H(x) & H(x) \end{pmatrix} \approx \begin{pmatrix} 0 & 0 \\ 0 & H(x) \end{pmatrix}$$

near global minima.

Assume further that

$$H(x) = \mathrm{diag}(\lambda_1(x), \cdots, \lambda_n(x)),$$

where $\lambda_1 > \cdots > \lambda_n$, then

$$\begin{cases} x_{t+1} = x_t - \dfrac{\eta}{2}\displaystyle\sum_{i=1}^{n} \nabla \lambda_i(x) y_t^2(i) \\ y_{t+1}(i) = y_t(i) - \eta\lambda_i(x_t) y_t(i) \quad \Rightarrow \quad y_{t+1}^2(i) = (1-\eta\lambda_i(x_t))^2 y_t^2(i) \end{cases}$$

Stable regime: $\lambda_1(x_t) < 2/\eta$ For smaller λ_i, $y_t(i)$ decays slower. So the weight of $\nabla\lambda_i$ keeps large for a longer time $\Rightarrow$ λ_i decays faster.

Smaller eigenvalues decays faster. Therefore, the spectrum concentrates to large eigenvalues.

Unstable Regime: $\lambda_1(x_t) > 2/\eta > \lambda_2(x_t) > \cdots$ For $i = 2, \cdots, n$, we have $y_t^2(i) = \max(1-\eta\lambda_2, 1-\eta\lambda_n)^{2t} y_0^2(i)$, which is exponential decay.

There exists T s.t. for $t > T$,

$$x_{t+1} = x_t - \frac{\eta y_t^2(1)}{2} \nabla \lambda_1(x_t)$$

But how large is $y_t(1)$? This cannot be estimated in this model since in this quadratic landscape, unstable means blowing up. We can introduce an asymmetric landscape to tackle this problem. Suppose the loss landscape is $\ell(y) = \frac{1}{2}ay^2 + \frac{1}{6}by^3$. The cubic term prevents it from blowing up even if $\eta > 2/a$. The dynamics bounce between two sides and in most iterations it nearly holds that $\ell'' = a + by = 2/\eta$, which yields

$$y = \frac{a - 2/\eta}{b}$$

where b is a constant. In a word, in most of the time $y_t(1)$ is close to the value where sharpness $= 2/\eta$. Therefore,

$$x_{t+1} = x_t - \eta(\lambda_1(x_t) - 2/\eta)^2 \nabla \lambda_1(x_t)$$

So $\lambda_1(x_t) \approx 2/\eta$.

The implicit bias of SAM Now let's take a look at

$$x_{t+1} = x_t - \frac{\eta}{2} \sum_{i=1}^{n} \nabla \lambda_i(x) y_t^2(i)$$

Since $y_t(1)$ decays fast and the spectrum concentrates to λ_1, how can we continue to decrease λ_1?

One way is to add perturbation to $y_t(1)$ to keep it big. This is exactly the implicit bias of Sharpness-aware Minimization (SAM).

SAM:

$$\begin{aligned}
\theta_{t+1} &= \theta_t - \eta \nabla f\left(\theta_t + \rho \frac{\nabla f(\theta_t)}{\|\nabla f(\theta_t)\|}\right) \\
&= \theta_t - \eta\left(\nabla f(\theta_t) + \rho \nabla^2 f(\theta_t) \frac{\nabla f(\theta_t)}{\|\nabla f(\theta_t)\|} + O(\rho^2)\right)
\end{aligned}$$

We have

$$\nabla^2 f(\theta_t) \approx \nabla^2 f(\theta^*) = \begin{pmatrix} 0 & 0 \\ 0 & H \end{pmatrix}$$

Then,

$$\begin{aligned}
y_{t+1} &= y_t - \eta H y_t - \eta\rho H \frac{Hy}{\|Hy\|} \\
&= \left(I - \eta H - \frac{\eta\rho H^2}{\|Hy_t\|}\right) y_t \\
&\approx -\frac{\eta\rho H^2}{\|Hy_t\|} y_t
\end{aligned}$$

as y_t is small.

SGD and noise structure SGD update

$$\theta_{t+1} = \theta_t - \eta(\nabla f(\theta_t) + \xi_t)$$

We assume the noise is constant level with a Hessian geometric structure

$$\mathbb{E}[\xi_t \xi_t^{\mathrm{T}}] = \sigma^2 \nabla^2 f(\theta_t) \approx \sigma^2 \begin{pmatrix} 0 & 0 \\ 0 & H(x_t) \end{pmatrix}$$

Similar to the above argument, the update is fast for y but slow for x. The dynamics of x actually sees the average $\mathbb{E}[y^2]$.

We have

$$\mathbb{E}\, y_{t+1}^2(i) = (1 - \eta\lambda_i)^2\, \mathbb{E}\, y_t^2(i) + \eta^2\sigma^2\lambda_i(x_t)$$

To solve the stationary value, let $q(i) = \mathbb{E}\, y_{t+1}^2(i) = \mathbb{E}\, y_t^2(i)$, then

$$q(i) = \frac{\eta^2\sigma^2\lambda_i}{2\eta\lambda_i - (\eta\lambda_i)^2} \approx \frac{\eta\sigma^2}{2}$$

Therefore,

$$\begin{aligned} x_{t+1} &= x_t - \eta \sum_{i=1}^{n} \nabla\lambda_i(x_t) \cdot \frac{\eta\sigma^2}{2} \\ &= x_t - \frac{\eta^2}{2B}\nabla(\mathrm{Tr}(H(x_t))) \end{aligned}$$

As a comparison, we investigate the case where the noise is isotropic, namely

$$\mathbb{E}[\xi_t \xi_t^{\mathrm{T}}] = \sigma^2 I$$

Following the same derivation we obtain

$$q_i = \frac{\eta\sigma^2}{2\lambda_i}$$

and

$$\begin{aligned} x_{t+1} &= x_t - \frac{\eta^2\sigma^2}{2} \sum_{i=1}^{n} \nabla\lambda_i(x_t) \cdot \frac{1}{\lambda_i(x_t)} \\ &= x_t - \frac{\eta^2}{2B}\nabla(\log\det H(x_t))) \end{aligned}$$

The first noise structure is good because it keeps a finite energy while minimizing sharpness efficiently. Note that, in order for the sharpness to decay in the same rate, the moment of noise with Hessian geometry is

$$\mathbb{E}\,\|\xi_t\|^2 = \mathrm{Tr}(H) = O(1),$$

but for isotropic noise it is

$$\mathbb{E}\,\|\xi_t\|^2 = O(n)$$

Another noise structure is that the noise level is proportion to the loss and the geometry is Hessian, namely

$$\mathbb{E}[\xi_t \xi_t^{\mathrm{T}}] = 2L(x_t, y_t) \begin{pmatrix} 0 & 0 \\ 0 & H(x_t) \end{pmatrix}$$

Therefore,

$$y_{t+1}(i) = (1 - \eta\lambda_i)y_t(i) + \eta\sqrt{\lambda_i \cdot 2L(x_t, y_t)}\tilde{\xi}_{t,i}$$

where $\tilde{\xi}_{t,i}$ is a random variable with zero mean and unit variance. Let $q_t := \mathbf{E}y_t^2$, we have

$$q_{t+1}(i) = (1 - \eta\lambda_i)^2 q_t(i) + \eta^2\lambda_i \left(\sum_{k=1}^{d} \lambda_k q_t(k)\right)$$

which is a linear dynamics. We van rewrite it as

$$q_{t+1} = (I - 2\eta H + \eta^2 H^2 + \eta\lambda\lambda^{\mathrm{T}})q_t$$

where $\lambda = (\lambda_1, \cdots, \lambda_m)^{\mathrm{T}}$. This is still a quadratic case where q_t either converges or blows up. In practice, it remains an open question what is SGD's EoS. We may need a model to characterize the phenomenon that the loss does not decay during the EoS process.

Discussion: Large η in practice From above discussion we know that large learning rate leads to EoS. This makes the convergence slow in sharp directions. So a question is, why do not use infinitely-small learning rate to accelerate the convergence corresponding to large eigenvalues in practice?

One guess: Although $\eta < 2/\lambda_1$ speeds up convergence in sharp directions, progressive sharpening will increase the gap between eigenvalues, i.e. increase the condition number of the problem. Using large η controls λ_1, thus controls the condition number. EoS helps convergence in this sense. In another word, with a large learning rate, the optimizer adaptively explores well-behaved regions.

References