Understanding the "Unstable Convergence" of Gradient Descent

Kwangjun Ahn
MIT EECS http://kjahn.mit.edu/
Joint w/ Jingzhao Zhang (MIT EECS → Tsinghua, IIIS) and Prof. Suvrit Sra (MIT EECS)

ICML 2022, Baltimore, MD

July 10, 2022

Motivation

- Gradient descent (GD) runs the iteration

$$\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^{t} - \eta \nabla f(\boldsymbol{\theta}^{t}),$$

to optimize a cost function f.

Motivation

- Gradient descent (GD) runs the iteration

$$\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^{t} - \eta \nabla f(\boldsymbol{\theta}^{t}),$$

to optimize a cost function f.

- (As we all know) it's conceptual building block for SGD

Motivation

- Gradient descent (GD) runs the iteration

$$\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^{t} - \eta \nabla f(\boldsymbol{\theta}^{t}),$$

to optimize a cost function f.
- (As we all know) it's conceptual building block for SGD
- most analyses of (S)GD relies on **"descent lemma"**: if f is $\boldsymbol{L}$-smooth, i.e., $\nabla^2 f \preceq \boldsymbol{L}$ then

Motivation

- Gradient descent (GD) runs the iteration

$$\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^t - \eta \nabla f(\boldsymbol{\theta}^t)\,,$$

to optimize a cost function f.
- (As we all know) it's conceptual building block for SGD
- most analyses of (S)GD relies on **"descent lemma"**: if f is $\boldsymbol{L}$-smooth, i.e., $\nabla^2 f \preceq \boldsymbol{L}$ then

$$f(\boldsymbol{\theta}^{t+1}) \leq f(\boldsymbol{\theta}^t) - \eta\big(1 - \boldsymbol{L}\frac{\eta}{2}\big)\|\nabla f(\boldsymbol{\theta}^t)\|^2\,. \tag{1}$$

Motivation

- Gradient descent (GD) runs the iteration

$$\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^t - \eta \nabla f(\boldsymbol{\theta}^t),$$

to optimize a cost function f.
- (As we all know) it's conceptual building block for SGD
- most analyses of (S)GD relies on **"descent lemma"**: if f is $\boldsymbol{L}$-smooth, i.e., $\nabla^2 f \preceq \boldsymbol{L}$ then

$$f(\boldsymbol{\theta}^{t+1}) \leq f(\boldsymbol{\theta}^t) - \eta\big(1 - \boldsymbol{L}\frac{\eta}{2}\big)\|\nabla f(\boldsymbol{\theta}^t)\|^2. \tag{1}$$

To ensure descent via inequality (1), most analyses impose the condition:

$$\boxed{\boldsymbol{L} < \frac{2}{\eta}.} \qquad \text{(Stable Regime)}$$

Motivation

- Gradient descent (GD) runs the iteration

$$\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^t - \eta \nabla f(\boldsymbol{\theta}^t)\,,$$

to optimize a cost function f.
- (As we all know) it's conceptual building block for SGD
- most analyses of (S)GD relies on **"descent lemma"**: if f is $\boldsymbol{L}$-smooth, i.e., $\nabla^2 f \preceq \boldsymbol{L}$ then

$$f(\boldsymbol{\theta}^{t+1}) \leq f(\boldsymbol{\theta}^t) - \eta\big(1 - \boldsymbol{L}\frac{\eta}{2}\big)\|\nabla f(\boldsymbol{\theta}^t)\|^2\,. \tag{1}$$

To ensure descent via inequality (1), most analyses impose the condition:

$$\boxed{\boldsymbol{L} < \frac{2}{\eta}.} \tag{Stable Regime}$$

- When cost is quadratic&convex, condition (Stable Regime) is in fact necessary for convergence: if $\eta > \frac{2}{\boldsymbol{L}}$, then GD diverges.

Is this true for nonconvex deep learning optimization?

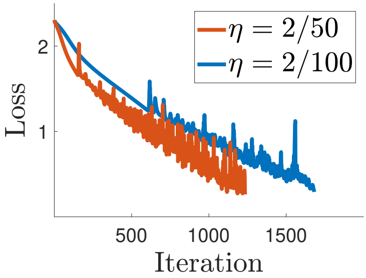
- We use (full-batch) GD to train a neural network on $5,000$ examples from CIFAR-10 with the CrossEntropy loss.
- A fully-connected architecture with two hidden layers of width 200 with ReLU activations.

Is this true for nonconvex deep learning optimization?

- We use (full-batch) GD to train a neural network on $5,000$ examples from CIFAR-10 with the CrossEntropy loss.
- A fully-connected architecture with two hidden layers of width 200 with ReLU activations.
- Throughout the talk, **sharpness** means the maximum eigenvalue of the loss Hessian, i.e., $\lambda_{\max}(\nabla^2 f(\boldsymbol{\theta}^t))$.

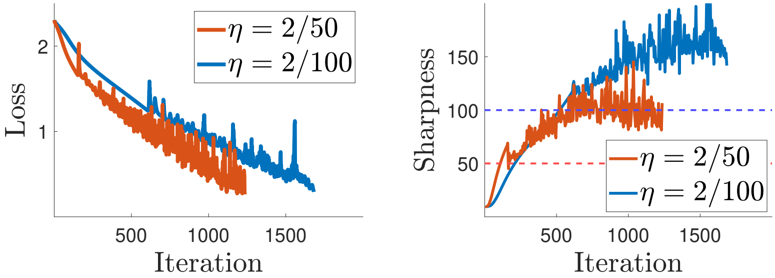
Is this true for nonconvex deep learning optimization?

- We use (full-batch) GD to train a neural network on $5,000$ examples from CIFAR-10 with the CrossEntropy loss.
- A fully-connected architecture with two hidden layers of width 200 with ReLU activations.
- Throughout the talk, **sharpness** means the maximum eigenvalue of the loss Hessian, i.e., $\lambda_{\max}(\nabla^2 f(\boldsymbol{\theta}^t))$.



Is this true for nonconvex deep learning optimization?

- We use (full-batch) GD to train a neural network on $5,000$ examples from CIFAR-10 with the CrossEntropy loss.
- A fully-connected architecture with two hidden layers of width 200 with ReLU activations.
- Throughout the talk, **sharpness** means the maximum eigenvalue of the loss Hessian, i.e., $\lambda_{\text{max}}(\nabla^2 f(\boldsymbol{\theta}^t))$.



Unstable convergence

- Recently, it has been observed that GD on neural networks often violates condition (Stable Regime). (Cohen et al. 2021) observe that when we run GD to train a neural network, the condition (Stable Regime) fails, [1]

[1] Cohen, Kaur, Li, Kolter, Talwalkar. "*Gradient descent on neural networks typically occurs at the edge of stability*." ICLR, 2021

Unstable convergence

- Recently, it has been observed that GD on neural networks often violates condition (Stable Regime). (Cohen et al. 2021) observe that when we run GD to train a neural network, the condition (Stable Regime) fails, [1]

- but contrary to the common wisdom from convex optimization, the training loss still (non-monotonically) decreases in the long run.

[1] Cohen, Kaur, Li, Kolter, Talwalkar. "*Gradient descent on neural networks typically occurs at the edge of stability*." ICLR, 2021

Unstable convergence

- Recently, it has been observed that GD on neural networks often violates condition (Stable Regime). (Cohen et al. 2021) observe that when we run GD to train a neural network, the condition (Stable Regime) fails, [1]

- but contrary to the common wisdom from convex optimization, the training loss still (non-monotonically) decreases in the long run.

- We call this phenomenon **unstable convergence**.

[1] Cohen, Kaur, Li, Kolter, Talwalkar. "*Gradient descent on neural networks typically occurs at the edge of stability*." ICLR, 2021

What is this work about?

- **Discuss the main causes** driving the unstable convergence phenomenon.

What is this work about?

- **Discuss the main causes** driving the unstable convergence phenomenon.
- **Identify the main features** that characterize unstable convergence (in terms of loss, iterates, and sharpness behaviors).

What is this work about?

- **Discuss the main causes** driving the unstable convergence phenomenon.
- **Identify the main features** that characterize unstable convergence (in terms of loss, iterates, and sharpness behaviors).
- Investigate and clarify the **relations between them**.

What is this work about?

- **Discuss the main causes** driving the unstable convergence phenomenon.
- **Identify the main features** that characterize unstable convergence (in terms of loss, iterates, and sharpness behaviors).
- Investigate and clarify the **relations between them**.
- Our characterizations demonstrate that the features of unstable convergence are in stark contrast with those of traditional stable convergence.

What is this work about?

- **Discuss the main causes** driving the unstable convergence phenomenon.
- **Identify the main features** that characterize unstable convergence (in terms of loss, iterates, and sharpness behaviors).
- Investigate and clarify the **relations between them**.
- Our characterizations demonstrate that the features of unstable convergence are in stark contrast with those of traditional stable convergence.
- In particular, our main features provide **alternative ways to identify unstable convergence** in practice.

Illustration of our main results (ReLU network)

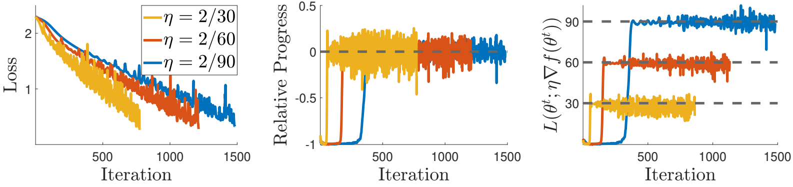
Object	Quantity	Behavior
Loss	$\mathrm{RP}(\boldsymbol{\theta}^t)$	oscillates near 0
Iterates	$L(\boldsymbol{\theta}^t; \eta \nabla f(\boldsymbol{\theta}^t))$	oscillates near $2/\eta$
Sharpness	$\lambda_{\max}(\nabla^2 f(\boldsymbol{\theta}^t))$	oscillates $\frac{\text{near}}{\text{above}}$ $2/\eta$

- **Relative Progress:** $\mathrm{RP}(\boldsymbol{\theta}) := \frac{f(\boldsymbol{\theta} - \eta \nabla f(\boldsymbol{\theta})) - f(\boldsymbol{\theta})}{\eta \|\nabla f(\boldsymbol{\theta})\|^2}$
- **Directional smoothness:** $L(\boldsymbol{\theta}; \mathbf{v}) := \frac{\langle \mathbf{v}, \nabla f(\boldsymbol{\theta}) - \nabla f(\boldsymbol{\theta} - \mathbf{v}) \rangle}{\|\mathbf{v}\|^2}$

Illustration of our main results (ReLU network)

Object	Quantity	Behavior
Loss	$\mathrm{RP}(\boldsymbol{\theta}^t)$	oscillates near 0
Iterates	$L(\boldsymbol{\theta}^t; \eta\nabla f(\boldsymbol{\theta}^t))$	oscillates near $2/\eta$
Sharpness	$\lambda_{\max}(\nabla^2 f(\boldsymbol{\theta}^t))$	oscillates $\frac{\text{near}}{\text{above}}$ $2/\eta$

- **Relative Progress:** $\mathrm{RP}(\boldsymbol{\theta}) := \frac{f(\boldsymbol{\theta} - \eta\nabla f(\boldsymbol{\theta})) - f(\boldsymbol{\theta})}{\eta\|\nabla f(\boldsymbol{\theta})\|^2}$
- **Directional smoothness:** $L(\boldsymbol{\theta}; \mathbf{v}) := \frac{\langle \mathbf{v}, \nabla f(\boldsymbol{\theta}) - \nabla f(\boldsymbol{\theta} - \mathbf{v})\rangle}{\|\mathbf{v}\|^2}$



It's actively studied in the literature!!

- Cohen, Kaur, Li, Kolter, Talwalkar. "*Gradient descent on neural networks typically occurs at the edge of stability*." ICLR, 2021
- Arora, Li, Panigrahi, "*Understanding gradient descent on edge of stability in deep learning.*" ICML 2022.
- Ma, Kunin, Wu, Ying. "*The multiscale structure of neural network loss functions: The effect on optimization and origin.*" 2022.
- and more!!

Thank you for listening

If you have any questions, shoot me an email! `kjahn@mit.edu`