Algorithm Stability

of (Stochastic) Gradient Descent

Rongzhen Wang

12/13/2023

What is stability?

Intuition

What is stability?

Intuition

- Stability of a system

- Be disturbed + Changes little

What is stability?

Intuition

- Stability of a system

- Be disturbed + Changes little

- Stability of an algorithm
- An algorithm $\mathcal{A}$ takes in some input x, e. g., a training data set, and returns an output, e.g., models/parameters/…

a small perturbation

$$x \to x + \Delta x$$

What is stability?

Intuition

- Stability of a system

- Be disturbed + Changes little

- Stability of an algorithm
- An algorithm $\mathcal{A}$ takes in some input x, e. g., a training data set, and returns an output, e.g., models/parameters/…

a small perturbation

$$x \to x + \Delta x$$

$$\mathcal{A}(x) \to \mathcal{A}(x + \Delta x)$$

Will it change a lot?

What is stability?

Intuition

- Stability of a system

- Be disturbed + Changes little

- Stability of an algorithm
- An algorithm $\mathcal{A}$ takes in some input x, e. g., a training data set, and returns an output, e.g., models/parameters/…

a small perturbation

$$x \to x + \Delta x$$

$$\mathcal{A}(x) \to \mathcal{A}(x + \Delta x)$$

Will it change a lot?

This is what stability concerns about!

What is stability?

Formal definition

- **Definition 1.1 (Uniform Stability,** Bousquet and Elisseeff [2002]**):** Let S, S' be two training sets differ by a single point. An algorithm $\mathcal{A}$ is **ϵ-uniformly stable** if for all such samples S, S', we have

$$\forall z \in Z, |\ell(\mathcal{A}(S); z) - \ell(\mathcal{A}(S'); z)| \leq \epsilon.$$

What is stability?

Formal definition

- **Definition 1.1 (Uniform Stability,** Bousquet and Elisseeff [2002]**):** Let S, S' be two training sets differ by a single point. An algorithm $\mathcal{A}$ is **ϵ-uniformly stable** if for all such samples S, S', we have

$$\forall z \in Z, |\ell(\mathcal{A}(S); z) - \ell(\mathcal{A}(S'); z)| \leq \epsilon.$$

- Explain in terms of stability:
- Input: Training set of size n
- Output: Model parameter

What is stability?

Formal definition

- **Definition 1.1 (Uniform Stability,** Bousquet and Elisseeff [2002]**):** Let S, S' be two training sets differ by a single point. An algorithm $\mathcal{A}$ is **ϵ-uniformly stable** if for all such samples S, S', we have

$$\forall z \in Z, |\ell(\mathcal{A}(S); z) - \ell(\mathcal{A}(S'); z)| \leq \epsilon.$$

- Explain in terms of stability:
- Input: Training set of size n

  Perturbation: replace a sample point $S = \{z_1, \cdots, z_{n-1}, z_n\} \rightarrow S' = \{z_1, \cdots, z_{n-1}, z_n{}'\}$

- Output: Model parameter

What is stability?

Formal definition

- **Definition 1.1 (Uniform Stability,** Bousquet and Elisseeff [2002]**):** Let S, S' be two training sets differ by a single point. An algorithm $\mathcal{A}$ is **ϵ-uniformly stable** if for all such samples S, S', we have

$$\forall z \in Z, |\ell(\mathcal{A}(S); z) - \ell(\mathcal{A}(S'); z)| \leq \epsilon.$$

- Explain in terms of stability:
- Input: Training set of size n

  Perturbation: replace a sample point $S = \{z_1, \cdots, z_{n-1}, z_n\} \to S' = \{z_1, \cdots, z_{n-1}, z_n'\}$

- Output: Model parameter

  Response: change measured by losses $|\ell(\mathcal{A}(S); z) - \ell(\mathcal{A}(S'); z)|$

What is stability?

Formal definition

- **Definition 1.1 (Uniform Stability,** Bousquet and Elisseeff [2002]**):** Let S, S' be two training sets differ by a single point. An algorithm $\mathcal{A}$ is **ϵ-uniformly stable** if for all such samples S, S', we have

$$\forall z \in Z, |\ell(\mathcal{A}(S); z) - \ell(\mathcal{A}(S'); z)| \leq \epsilon.$$

the maximum change of loss, measured on any example

- Explain in terms of stability:
- Input: Training set of size n

  Perturbation: replace a sample point $S = \{z_1, \cdots, z_{n-1}, z_n\} \to S' = \{z_1, \cdots, z_{n-1}, z_n'\}$
- Output: Model parameter

  Response: change measured by losses $|\ell(\mathcal{A}(S); z) - \ell(\mathcal{A}(S'); z)|$

What is stability?

Extended definition for stochastic algorithms

- For a stochastic algorithm like Stochastic Gradient Descent (SGD), its output is not determined, then how can we measure a random variable

$$|\ell(\mathcal{A}(S); z) - \ell(\mathcal{A}(S'); z)| \text{ ???}$$

What is stability?

Extended definition for stochastic algorithms

- For a stochastic algorithm like Stochastic Gradient Descent (SGD), its output is not determined, then how can we measure a random variable

$$|\ell(\mathcal{A}(S); z) - \ell(\mathcal{A}(S'); z)| \text{ ???}$$

- Taking expectation!

- **Definition 1.2 (Uniform Stability in Expectation,** Hardt et al. [2016]**):** Let S, S' be two training sets differ by a single point. A stochastic algorithm $\mathcal{A}$ is **ϵ-uniformly stable in expectation** if for all such samples S, S', we have

$$\forall z \in Z, \mathbb{E}_{\mathcal{A}}|\ell(\mathcal{A}(S); z) - \ell(\mathcal{A}(S'); z)| \leq \epsilon.$$

And the smallest such ϵ is called the stability coefficient of $\mathcal{A}$, denoted as ϵ_{stab}.

Today's topics

- Stability of (Stochastic) Gradient Descent
  - shows different properties on different loss functions
  - can be controlled by adjusting the learning rate

- Techniques to induce stability of SGD

- Some core problems

(Stochastic) Gradient Descent

- GD

Algorithm 1 Gradient Descent

Input: Initialization θ_0; learning rate scheme α_t;
for $t = 1$ **to** T **do**
  $\theta_t \leftarrow \theta_{t-1} - \alpha_t \frac{1}{n} \sum_{i=1}^{n} \nabla \ell(\theta_{t-1}; z_i)$
end for
Output: θ_T

- SGD

Algorithm 2 Stochastic Gradient Descent

Input: Initialization θ_0; learning rate scheme α_t;
for $t = 1$ **to** T **do**
  $\boxed{\text{Sample } i_t \text{ uniformly from } 1, \ldots, n}$
  $\theta_t \leftarrow \theta_{t-1} - \alpha_t \boxed{\nabla \ell(\theta_{t-1}; z_{i_t})}$
end for
Output: θ_T

(Stochastic) Gradient Descent

- GD

Algorithm 1 Gradient Descent

Input: Initialization θ_0; learning rate scheme α_t;
for $t = 1$ **to** T **do**
  $\theta_t \leftarrow \theta_{t-1} - \alpha_t \frac{1}{n} \sum_{i=1}^{n} \nabla \ell(\theta_{t-1}; z_i)$
end for
Output: θ_T

- SGD

Algorithm 2 Stochastic Gradient Descent

Input: Initialization θ_0; learning rate scheme α_t;
for $t = 1$ **to** T **do**
  $\boxed{\text{Sample } i_t \text{ uniformly from } 1, \ldots, n}$
  $\theta_t \leftarrow \theta_{t-1} - \alpha_t \boxed{\nabla \ell(\theta_{t-1}; z_{i_t})}$
end for
Output: θ_T

mini-batch SGD is somewhere between

(Stochastic) Gradient Descent

From *loss* to *parameters*

- Denote the corresponding outputs of (Stochastic) Gradient Descent trained on S and S' by $\{\theta_0, \theta_1, \cdots, \theta_T\}$ and $\{\theta_0, {\theta_1}', \cdots, {\theta_T}'\}$.
- The algorithm outputs $\theta_T / {\theta_T}'$.
- Recalling our goal is to bound $|\ell(\mathcal{A}(S); z) - \ell(\mathcal{A}(S'); z)|$, which is $|\ell(\theta_T; z) - \ell({\theta_T}'; z)|$ in this case.

(Stochastic) Gradient Descent

From *loss* to *parameters*

- Denote the corresponding outputs of (Stochastic) Gradient Descent trained on S and S' by $\{\theta_0, \theta_1, \cdots, \theta_T\}$ and $\{\theta_0, \theta_1{}', \cdots, \theta_T{}'\}$.
- The algorithm outputs $\theta_T/\theta_T{}'$.
- Recalling our goal is to bound $|\ell(\mathcal{A}(S); z) - \ell(\mathcal{A}(S'); z)|$, which is $|\ell(\theta_T; z) - \ell(\theta_T{}'; z)|$ in this case.
- The analyses of the gap of losses is not straightforward, and is relevant to particular form of the loss function. 😟

(Stochastic) Gradient Descent

From *loss* to *parameters*

- Denote the corresponding outputs of (Stochastic) Gradient Descent trained on S and S' by $\{\theta_0, \theta_1, \cdots, \theta_T\}$ and $\{\theta_0, \theta_1{}', \cdots, \theta_T{}'\}$.
- The algorithm outputs $\theta_T / \theta_T{}'$.
- Recalling our goal is to bound $|\ell(\mathcal{A}(S); z) - \ell(\mathcal{A}(S'); z)|$, which is $|\ell(\theta_T; z) - \ell(\theta_T{}'; z)|$ in this case.
- The analyses of the gap of losses is not straightforward, and is relevant to particular form of the loss function. 😟
- Can we convert the analysis of loss into the analysis of parameters?
- Yes!

(Stochastic) Gradient Descent

From *loss* to *parameters*

- **Definition 2 (L-Lipschitz):** A function f is L-Lipschitz, if for all u, v in its domain Ω we have

$$\|f(u) - f(v)\| \leq L\|u - v\|.$$

- And for a continuously differentiable f, this is equivalent with

$$\sup_{x \in \Omega} \|\nabla f(x)\| \leq L.$$

(Stochastic) Gradient Descent

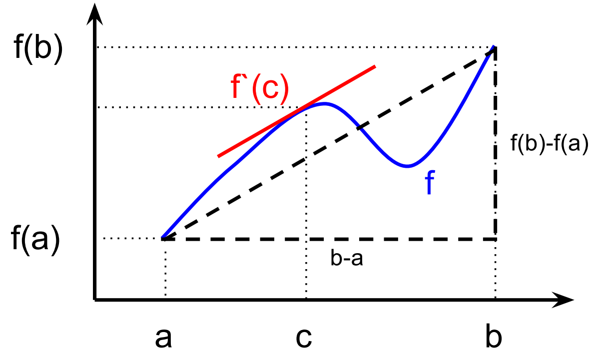
From *loss* to *parameters*

- **Definition 2.1 (L-Lipschitz):** A function f is L-Lipschitz, if for all u, v in its domain Ω we have

$$\|f(u) - f(v)\| \le L\|u - v\|.$$

- And for a continuously differentiable f, this is equivalent with

$$\forall x \in \Omega, \|\nabla f(x)\| \le L.$$



Intuition from Mean Value Theorem:

f(b)-f(a) = (some gradient)*(b-a)

upper bounded by L

Image Credit: https://commons.wikimedia.org/wiki/File:Mean_value_theorem_(Lagrange%27s_theorem).svg

(Stochastic) Gradient Descent

From *loss* to *parameters*

- **Lemma 1 (Lipschitz Bounded Loss):** Suppose the loss function ℓ is L-Lipschitz on all example z in data space Z, then

$$\forall z \in Z, \mathbb{E}_{\mathcal{A}} |\ell(\theta_T; z) - \ell(\theta_T{}'; z)| \leq L\|\theta_T - \theta_T{}'\|.$$

(Stochastic) Gradient Descent

From *loss* to *parameters*

- **Lemma 1 (Lipschitz Bounded Loss):** Suppose the loss function ℓ is L-Lipschitz on all example z in data space Z, then

$$\forall z \in Z, \mathbb{E}_{\mathcal{A}}|\ell(\theta_T; z) - \ell(\theta_T{}'; z)| \leq L\|\theta_T - \theta_T{}'\|.$$

- Now we can get into the analyses of the stability of (Stochastic) Gradient Descent!
- Let's look at GD first!

Stability of GD

- Note that GD is an iterative method. Let's think step by step.

$$\theta_t = \theta_{t-1} - \alpha_t \frac{1}{n} \sum_{i=1}^{n} \nabla \ell(\theta_{t-1}; z_i)$$

Stability of GD

- Note that GD is an iterative method. Let's think step by step.

$$\theta_t = \theta_{t-1} - \alpha_t \frac{1}{n} \sum_{i=1}^{n} \nabla \ell(\theta_{t-1}; z_i) = \frac{1}{n} \sum_{i=1}^{n} [\theta_{t-1} - \alpha_t \nabla \ell(\theta_{t-1}; z_i)]$$

Decomposed into updates derived from each example

Stability of GD

- Note that GD is an iterative method. Let's think step by step.

$$\theta_t = \theta_{t-1} - \alpha_t \frac{1}{n} \sum_{i=1}^{n} \nabla \ell(\theta_{t-1}; z_i) = \frac{1}{n} \sum_{i=1}^{n} [\theta_{t-1} - \nabla \ell(\theta_{t-1}; z_i)]$$

Decomposed into updates derived from each example

- Similarly, for S' we have

$$\theta_t{}' = \frac{1}{n} \sum_{i=1}^{n} [\theta_{t-1}{}' - \alpha_t \nabla \ell(\theta_{t-1}{}'; \ z_i{}')] \, .$$

Stability of GD

- Note that GD is an iterative method. Let's think step by step.

$$\theta_t = \theta_{t-1} - \alpha_t \frac{1}{n}\sum_{i=1}^{n} \nabla \ell(\theta_{t-1}; z_i) = \frac{1}{n}\sum_{i=1}^{n} [\theta_{t-1} - \nabla \ell(\theta_{t-1}; z_i)]$$

Decomposed into updates derived from each example

- Similarly, for S' we have

$$\theta_t{}' = \frac{1}{n}\sum_{i=1}^{n} [\theta_{t-1}{}' - \alpha_t \nabla \ell(\theta_{t-1}{}'; z_i{}')] .$$

- Then we get the divergence of parameter as

$$\theta_t - \theta_t{}' = \frac{1}{n}\sum_{i=1}^{n} [\theta_{t-1} - \alpha_t \nabla \ell(\theta_{t-1}; z_i)] - \frac{1}{n}\sum_{i=1}^{n} [\theta_{t-1}{}' - \alpha_t \nabla \ell(\theta_{t-1}{}'; z_i{}')]$$

Stability of GD

$$\theta_t - \theta_t{}' = \frac{1}{n}\sum_{i=1}^{n}[\theta_{t-1} - \alpha_t \nabla\ell(\theta_{t-1}; z_i)] - \frac{1}{n}\sum_{i=1}^{n}[\theta_{t-1}{}' - \alpha_t \nabla\ell(\theta_{t-1}{}';\ z_i{}')]$$
$$= \frac{1}{n}\sum_{i=1}^{n}\left[\theta_{t-1} - \theta_{t-1}{}' - \alpha_t\left(\nabla\ell(\theta_{t-1}; z_i) - \nabla\ell(\theta_{t-1}{}';\ z_i{}')\right)\right]$$

Stability of GD

$$\theta_t - \theta_t{}' = \frac{1}{n}\sum_{i=1}^{n}[\theta_{t-1} - \alpha_t \nabla \ell(\theta_{t-1}; z_i)] - \frac{1}{n}\sum_{i=1}^{n}[\theta_{t-1}{}' - \alpha_t \nabla \ell(\theta_{t-1}{}';\ z_i{}')]$$

$$= \frac{1}{n}\sum_{i=1}^{n}\left[\theta_{t-1} - \theta_{t-1}{}' - \alpha_t\left(\nabla \ell(\theta_{t-1}; z_i) - \nabla \ell(\theta_{t-1}{}';\ z_i{}')\right)\right] |$$

$$= \frac{1}{n}\left\{\underbrace{\sum_{i=1}^{n-1}\left[\theta_{t-1} - \theta_{t-1}{}' - \alpha_t\left(\nabla \ell(\theta_{t-1}; z_i) - \nabla \ell(\theta_{t-1}{}'; z_i)\right)\right]}_{\text{(I): update on the same examples}}\right.$$

$$S = \{z_1, \cdots, z_{n-1}, z_n\}$$
$$S' = \{z_1, \cdots, z_{n-1}, z_n{}'\}$$

$$\left.+\underbrace{\left[\theta_{t-1} - \theta_{t-1}{}' - \alpha_t\left(\nabla \ell(\theta_{t-1}; z_n) - \nabla \ell(\theta_{t-1}{}'; z_n{}')\right)\right]}_{\text{(II): update on the different examples}}\right\}$$

(I): update on the *same* examples

(II): update on the *different* examples

Stability of GD

Bound for (II)

- First bound the part of (II). It is simpler.
- (II): $\theta_{t-1} - \theta_{t-1}' - \alpha_t\left(\nabla\ell(\theta_{t-1}; z_n) - \nabla\ell(\theta_{t-1}'; z_n')\right)$

Stability of GD

Bound for (II)

- First bound the part of (II). It is simpler.
- (II): $\theta_{t-1} - \theta_{t-1}' - \alpha_t\left(\nabla\ell(\theta_{t-1}; z_n) - \nabla\ell(\theta_{t-1}'; z_n')\right)$
- There is no internal connection between $\nabla\ell(\theta_{t-1}; z_n) - \nabla\ell(\theta_{t-1}'; z_n')$ (they are two different functions), so we use the most ordinary tools to bound it.
- Triangle inequality + Lipschitzness

Stability of GD

Bound for (II)

- First bound the part of (II). It is simpler.
- (II): $$\theta_{t-1} - \theta_{t-1}{}' - \alpha_t\Big(\nabla\ell(\theta_{t-1}; z_n) - \nabla\ell(\theta_{t-1}{}'; z_n{}')\Big)$$
- There is no internal connection between $\nabla\ell(\theta_{t-1}; z_n) - \nabla\ell(\theta_{t-1}{}'; z_n{}')$ (they are two different functions), so we use the most ordinary tools to bound it.
- Triangle inequality + Lipschitzness
- $$\begin{aligned} &\Big\|\theta_{t-1} - \theta_{t-1}{}' - \alpha_t\Big(\nabla\ell(\theta_{t-1}; z_n) - \nabla\ell(\theta_{t-1}{}'; z_n)\Big)\Big\| \\ &\le \|\theta_{t-1} - \theta_{t-1}{}'\| + \alpha_t\|\nabla\ell(\theta_{t-1}; z_n)\| + \alpha_t\|\nabla\ell(\theta_{t-1}{}'; z_n)\| \\ &\le \|\theta_{t-1} - \theta_{t-1}{}'\| + 2\alpha_t \sup\|\nabla\ell(\theta; z)\| \\ &= \|\theta_{t-1} - \theta_{t-1}{}'\| + 2\alpha_t L \end{aligned}$$

Triangle inequality

L-Lipschitz of loss: bounded gradient

Stability of GD

Bound for (I)

- Then bound the part of (I). It is more complex but vital.
- (I): $\sum_{i=1}^{n-1} \left[\theta_{t-1} - \theta_{t-1}{}' - \alpha_t\left(\nabla\ell(\theta_{t-1}; z_i) - \nabla\ell(\theta_{t-1}{}'; z_i)\right)\right]$

Stability of GD

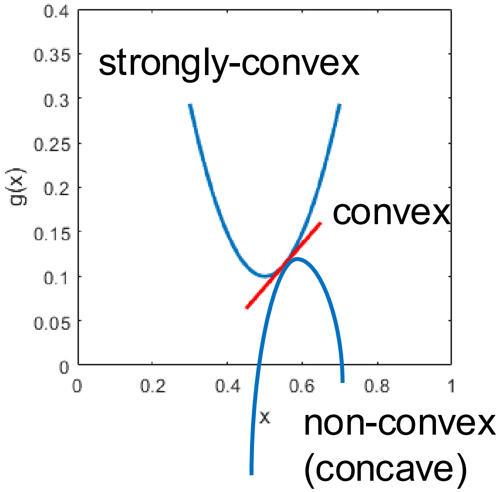
Bound for (I)

- Then bound the part of (I). It is more complex but vital.
- (I): $\sum_{i=1}^{n-1}\left[\theta_{t-1} - \theta_{t-1}' - \alpha_t\left(\nabla\ell(\theta_{t-1}; z_i) - \nabla\ell(\theta_{t-1}'; z_i)\right)\right]$
- Element term: $\theta - \theta' - \alpha\left(\nabla\ell(\theta; z) - \nabla\ell(\theta'; z)\right)$

Stability of GD

Bound for (I)

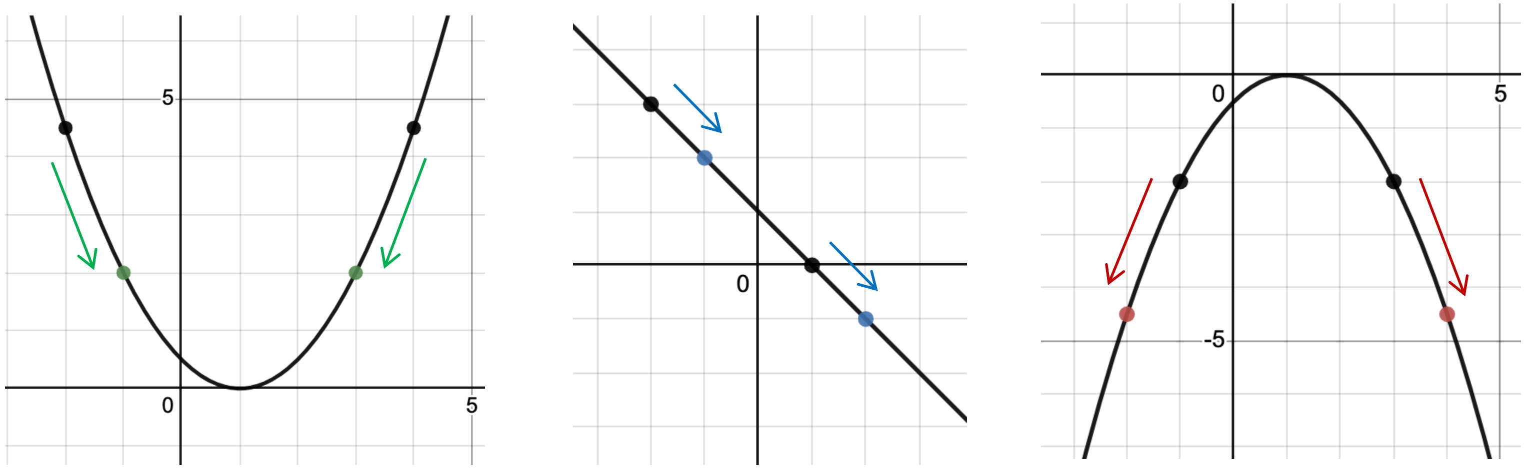
- Then bound the part of (I). It is more complex but vital.
- (I): $\sum_{i=1}^{n-1}\left[\theta_{t-1}-\theta_{t-1}{}'-\alpha_t\left(\nabla\ell(\theta_{t-1};z_i)-\nabla\ell(\theta_{t-1}{}';z_i)\right)\right]$
- Element term: $G(\theta)-G(\theta')\triangleq\theta-\theta'-\alpha\left(\nabla\ell(\theta;z)-\nabla\ell(\theta';z)\right)$
- (Forecast) Property of update rules:
  - non-convex loss: $\|G(\theta)-G(\theta')\|\leq(1+\kappa_1)\|\theta-\theta'\|$
  - convex loss: $\|G(\theta)-G(\theta')\|\leq\|\theta-\theta'\|$
  - γ-strongly convex loss: $\|G(\theta)-G(\theta')\|\leq(1-\kappa_2)\|\theta-\theta'\|$



Stability of GD

Bound for (I)

- (Forecast) Property of update rules:
  - non-convex loss: $\|G(\theta) - G(\theta')\| \leq (1 + \kappa_1)\|\theta - \theta'\|$
  - convex loss: $\|G(\theta) - G(\theta')\| \leq 1\|\theta - \theta'\|$
  - γ-strongly convex loss: $\|G(\theta) - G(\theta')\| \leq (1 - \kappa_2)\|\theta - \theta'\|$

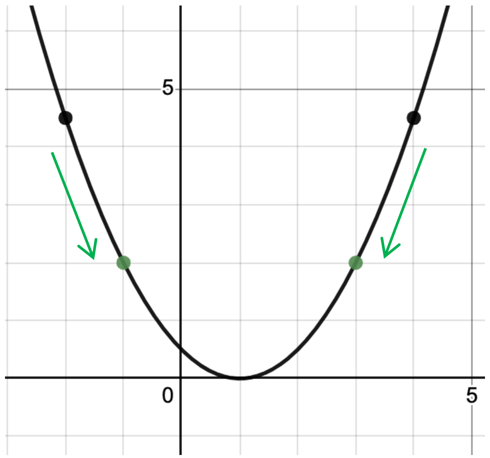


strongly convex | convex | non-convex

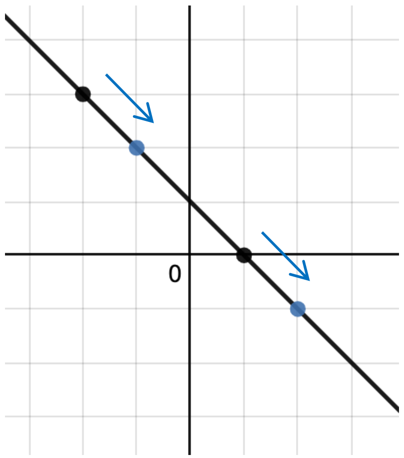
Stability of GD

Bound for (I)

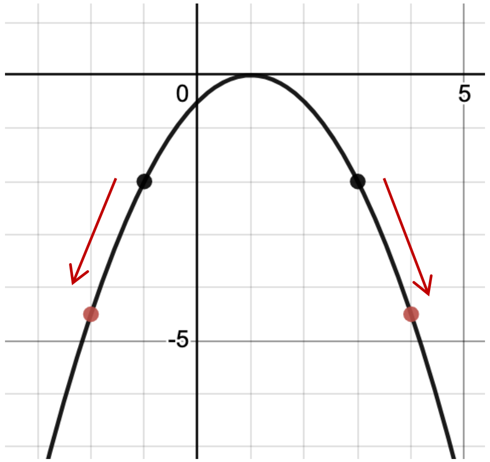
- (Forecast) Property of update rules:
  - non-convex loss: $\|G(\theta) - G(\theta')\| \leq (1 + \kappa_1)\|\theta - \theta'\|$
  - convex loss: $\|G(\theta) - G(\theta')\| \leq 1\|\theta - \theta'\|$
  - γ-strongly convex loss: $\|G(\theta) - G(\theta')\| \leq (1 - \kappa_2)\|\theta - \theta'\|$



strongly convex



convex



non-convex

Differ in scaling index!

And it matters!!

Property of update rules

Non-convex case

- Check again, the element term: $G(\theta) - G(\theta') \triangleq \theta - \theta' - \alpha\big(\nabla\ell(\theta; z) - \nabla\ell(\theta'; z)\big)$,
- and what we want: $\|G(\theta) - G(\theta')\| \leq (1 + \kappa_1)\|\theta - \theta'\|$.
- Consider some property like Lipschitz for the gradient $\nabla\ell(\theta; z)$?

Property of update rules

Non-convex case

- Check again, the element term: $G(\theta) - G(\theta') \triangleq \theta - \theta' - \alpha\big(\nabla \ell(\theta; z) - \nabla \ell(\theta'; z)\big)$,
- and what we want: $\|G(\theta) - G(\theta')\| \leq (1 + \kappa_1)\|\theta - \theta'\|$.
- Consider some property like Lipschitz for the gradient $\nabla \ell(\theta; z)$?

- **Definition 2.2 (β-smooth):** A differentiable function f is β-smooth, if for all u, v in its domain Ω we have

$$\|\nabla f(u) - \nabla f(v)\| \leq \beta \|u - v\|.$$

Property of update rules

Non-convex case

- Check again, the element term: $G(\theta) - G(\theta') \triangleq \theta - \theta' - \alpha_t\big(\nabla \ell(\theta; z) - \nabla \ell(\theta'; z)\big)$,
- and what we want: $\|G(\theta) - G(\theta')\| \leq (1 + \kappa_1)\|\theta - \theta'\|$.
- Consider some property like Lipschitz for the gradient $\nabla \ell(\theta; z)$?

- **Definition 2.2 (β-smooth):** A differentiable function f is β-smooth, if for all u, v in its domain Ω we have

$$\|\nabla f(u) - \nabla f(v)\| \leq \beta \|u - v\|.$$

- **Definition 2.1 (L-Lipschitz):** A function f is L-Lipschitz, if for all u, v in its domain Ω we have

$$\|f(u) - f(v)\| \leq L\|u - v\|.$$

Property of update rules

Non-convex case

$$
\begin{aligned}
\|G(\theta) - G(\theta')\| &= \left\|\theta - \theta' - \alpha\big(\nabla\ell(\theta; z) - \nabla\ell(\theta'; z)\big)\right\| \\
&\leq \|\theta - \theta'\| + \alpha\|\nabla\ell(\theta; z) - \nabla\ell(\theta'; z)\| \\
&\leq \|\theta - \theta'\| + \alpha\beta\|\theta - \theta'\| \\
&= (1 + \alpha\beta)\|\theta - \theta'\|
\end{aligned}
$$

Property of update rules

Non-convex case

$$\begin{aligned}\|G(\theta) - G(\theta')\| &= \left\|\theta - \theta' - \alpha\left(\nabla\ell(\theta; z) - \nabla\ell(\theta'; z)\right)\right\| \\ &\le \|\theta - \theta'\| + \alpha\|\nabla\ell(\theta; z) - \nabla\ell(\theta'; z)\| \\ &\le \|\theta - \theta'\| + \alpha\beta\|\theta - \theta'\| \\ &= (1 + \alpha\beta)\|\theta - \theta'\|\end{aligned}$$

- **Lemma 2.1 (Property of Update in the Non-convex Case):** Suppose the loss function ℓ is β-smooth on any example $z \in Z$, then

$$\forall\theta, \theta' \in \Theta, \|G(\theta) - G(\theta')\| \le (1 + \alpha\beta)\|\theta - \theta'\|.$$

Property of update rules

Convex and strongly-convex case

- We will give the following conclusions without proof. The results come mainly from some basic properties of convex and smooth functions.

- **Lemma 2.2 (Property of Update in the Convex Case):** Suppose the loss function ℓ is convex and β-smooth on any example $z \in Z$, if the learning rate $\alpha \leq \frac{2}{\beta}$, then

$$\forall \theta, \theta' \in \Theta, \|G(\theta) - G(\theta')\| \leq \|\theta - \theta'\|.$$

- **Lemma 2.3 (Property of Update in the Strongly-convex Case):** Suppose the loss function ℓ is γ-strongly convex and β-smooth any example $z \in Z$, if the learning rate $\alpha \leq \frac{2}{\beta+\gamma}$, then

$$\forall \theta, \theta' \in \Theta, \|G(\theta) - G(\theta')\| \leq (1 - \alpha \frac{\beta\gamma}{\beta + \gamma})\|\theta - \theta'\|.$$

Stability of GD

One-step dynamic

- Recall that: $\theta_t - \theta_t{}' = \frac{1}{n}\Big\{\sum_{i=1}^{n-1}\Big[\theta_{t-1} - \theta_{t-1}{}' - \alpha_t\Big(\nabla\ell(\theta_{t-1}; z_i) - \nabla\ell(\theta_{t-1}{}'; z_i)\Big)\Big]$

$$+\Big[\theta_{t-1} - \theta_{t-1}{}' - \alpha_t\Big(\nabla\ell(\theta_{t-1}; z_n) - \nabla\ell(\theta_{t-1}{}'; z_n{}')\Big)\Big]\Big\}.$$

- Now we have bounded both (I) and (II).
- Plugging in the previous results, we have
- $\|\theta_t - \theta_t{}'\| \le \frac{1}{n}[(n-1)\,(1+\alpha_t\beta)\|\theta_{t-1} - \theta_{t-1}{}'\| + \|\theta_{t-1} - \theta_{t-1}{}'\| + 2\alpha_t L]$

$$\le (1+\alpha_t\beta)\|\theta_{t-1} - \theta_{t-1}{}'\| + \frac{2\alpha_t L}{n}.$$

Stability of GD

One-step dynamic

- $\|\theta_t - \theta_t{}'\| \le (1 + \alpha_t\beta)\|\theta_{t-1} - \theta_{t-1}{}'\| + \frac{2\alpha_t L}{n}$.
- Consider a constant learning rate, with $\theta_0 = \theta_0{}' = 0$, we recursively get
- $\|\theta_T - \theta_T{}'\| \le \sum_{t=1}^{T}(1 + \alpha\beta)^t \frac{2\alpha L}{n} = \frac{2L}{n\beta}[(1 + \alpha\beta)^T - 1]$.

Stability of GD

One-step dynamic

- $\|\theta_t - \theta_t{}'\| \le (1+\alpha_t\beta)\|\theta_{t-1} - \theta_{t-1}{}'\| + \frac{2\alpha_t L}{n}$.
- Consider a constant learning rate, with $\theta_0 = \theta_0{}' = 0$, we recursively get
- $\|\theta_t - \theta_t{}'\| \le \sum_{t=1}^{T}(1+\alpha\beta)^t \frac{2\alpha L}{n} = \frac{2L}{n\beta}[(1+\alpha\beta)^T - 1]$.
- For Lemma 1, $\forall z \in Z, \mathbb{E}_{\mathcal{A}}|\ell(\theta_T; z) - \ell(\theta_T{}'; z)| \le L\|\theta_T - \theta_T{}'\|$, then we have

- **Theorem 1.1 (Stability of GD in the Non-convex Case):** Suppose the loss function ℓ is β-smooth on any example $z \in Z$, then implementing T-step GD with constant learning rate α leads to a stability coefficient less than

$$\epsilon_{\text{stab}} \le \frac{2L^2}{n\beta}[(1+\alpha\beta)^T - 1] = \mathcal{O}(\frac{(1+\alpha\beta)^T}{n}).$$

Stability of GD

One-step dynamic

- The convex and strongly-convex case only differ in the scaling index, which leads to similar results.

- **Theorem 1.2 (Stability of GD in the Convex Case):** Suppose the loss function ℓ is convex and β-smooth on any example $z \in Z$, then implementing T-step GD with constant learning rate $\alpha \leq \frac{2}{\beta}$ leads to a stability coefficient less than

$$\epsilon_{\text{stab}} \leq \frac{2\alpha L^2}{n\beta} T = \mathcal{O}\left(\frac{T}{n}\right).$$

Stability of GD

One-step dynamic

- The convex and strongly-convex case only differ in the scaling index, which leads to similar results.

- **Theorem 1.3 (Stability of GD in the Strongly-convex Case):** Suppose the loss function ℓ is γ-strongly convex and β-smooth on any example $z \in Z$, then implementing T-step GD with constant learning rate $\alpha \leq \frac{2}{\beta+\gamma}$ leads to a stability coefficient less than

$$\epsilon_{\text{stab}} \leq \frac{2L^2}{n\frac{\beta\gamma}{\beta+\gamma}}\left[1-\left(1-\alpha\frac{\beta\gamma}{\beta+\gamma}\right)^T\right] = \mathcal{O}(\frac{1}{n}).$$

Summary

- Until now, we have discussed…

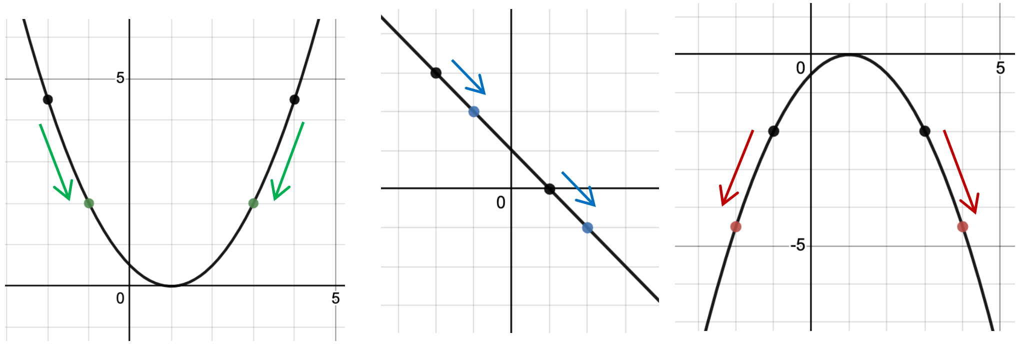
Summary

- Until now, we have discussed…
- Definition of stability:
  - Uniform stability (in expectation) for (stochastic) algorithms.

Summary

- Until now, we have discussed…
- Definition of stability:
  - Uniform stability (in expectation) for (stochastic) algorithms.

- Stability of Gradient Descent: with constant learning rate,
  - Non-convex case: $\epsilon_{\text{stab}} = \mathcal{O}(\frac{(1+\alpha\beta)^T}{n})$,
  - Convex case: $\epsilon_{\text{stab}} = \mathcal{O}\left(\frac{T}{n}\right)$,
  - γ-Strongly convex case: $\epsilon_{\text{stab}} = \mathcal{O}(\frac{1}{n})$.



Tightness of the upper bound

- Wait…Is our bound tight?

Image Credit: https://www.sciencedirect.com/topics/engineering/upper-bound-method

Tightness of the upper bound

- Wait…Is our bound tight?
- Why do we care about the tightness? ?

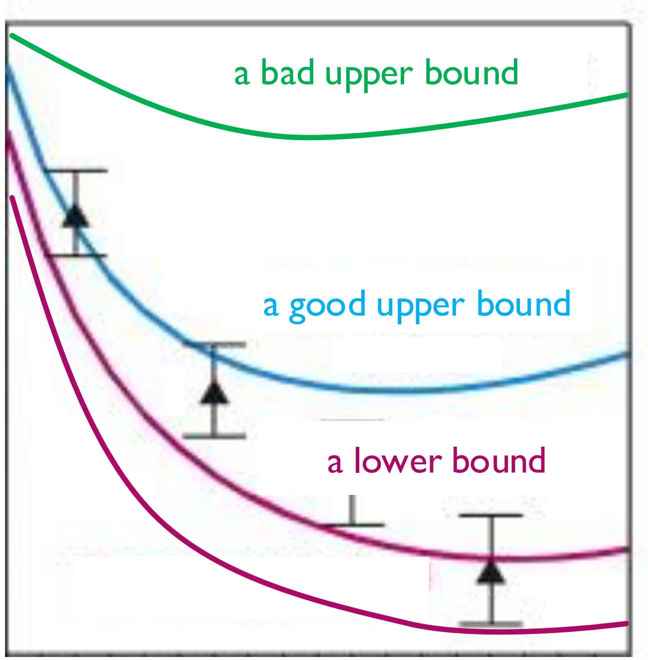
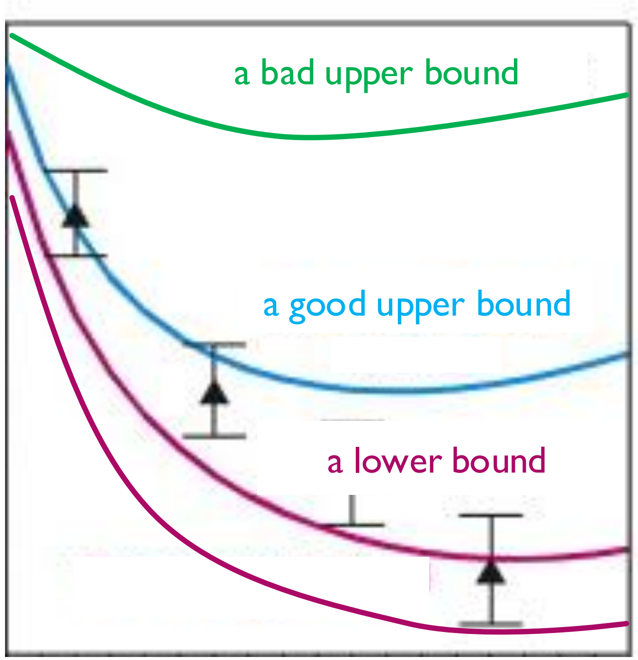


Image Credit: https://www.sciencedirect.com/topics/engineering/upper-bound-method

Tightness of the upper bound

- Wait…Is our bound tight?
- Why do we care about the tightness? ?



Exploring the lower bound for the upper bound matters!

Image Credit: https://www.sciencedirect.com/topics/engineering/upper-bound-method

Tightness of the upper bound

A simple case

- Consider a simple setting:
- Only 1 example in the training set,

$$S = \{z\}, S' = \{z'\}.$$

Tightness of the upper bound

A simple case

- Consider a simple setting:
- Only 1 example in the training set,

$$S = \{z\}, S' = \{z'\}.$$

- Only 1 dimension for feature and label,

$$z = (x, y) = (1,1),\ z' = (x', y') = (-1,1).$$

Tightness of the upper bound

A simple case

- Consider a simple setting:
- Only 1 example in the training set,

$$S = \{z\}, S' = \{z'\}.$$

- Only 1 dimension for feature and label,

$$z = (x, y) = (1,1),\ z' = (x', y') = (-1,1).$$

- Linear relation $y \sim \theta x$, so that θ is also a scalar. Initialize it with $\theta_0 = 0$.

Tightness of the upper bound

A simple case

- Consider a simple setting:
- Only 1 example in the training set,

$$S = \{z\}, S' = \{z'\}.$$

- Only 1 dimension for feature and label,

$$z = (x, y) = (1,1),\ z' = (x', y') = (-1,1).$$

- Linear relation $y \sim \theta x$, so that θ is also a scalar. Initialize it with $\theta_0 = 0$.
- Three kinds of loss functions,

$$\ell_1 = y - \theta x, \qquad \ell_2 = \frac{1}{2}(y - \theta x)^2, \qquad \ell_3 = -\frac{1}{2}(y - \theta x)^2.$$

convex   strongly-convex   non-convex

Tightness of the upper bound

For convex loss ℓ_1

- $\ell_1 = y - \theta x$, $\nabla \ell_1 = -x$.
- Trained on $S = \{(1,1)\}$
- $\theta_1 = \theta_0 - \alpha(-x) = \alpha$

- Trained on $S' = \{(-1,1)\}$
- $\theta_1{}' = \theta_0 - \alpha(-x') = -\alpha$

Tightness of the upper bound

For convex loss ℓ_1

- $\ell_1 = y - \theta x$, $\nabla \ell_1 = -x$.
- Trained on $S = \{(1,1)\}$
- $\theta_1 = \theta_0 - \alpha(-x) = \alpha$
- $\theta_2 = \theta_1 - \alpha(-x) = 2\alpha$

- Trained on $S' = \{(-1,1)\}$
- $\theta_1{}' = \theta_0 - \alpha(-x') = -\alpha$
- $\theta_2{}' = \theta_1{}' - \alpha(-x') = -2\alpha$

Tightness of the upper bound

For convex loss ℓ_1

- $\ell_1 = y - \theta x$, $\nabla \ell_1 = -x$.
- Trained on $S = \{(1,1)\}$
- $\theta_1 = \theta_0 - \alpha(-x) = \alpha$
- $\theta_2 = \theta_1 - \alpha(-x) = 2\alpha$
- …
- $\theta_T = \theta_{T-1} - \alpha(-x) = T\alpha$

- Trained on $S' = \{(-1,1)\}$
- $\theta_1{}' = \theta_0 - \alpha(-x') = -\alpha$
- $\theta_2{}' = \theta_1{}' - \alpha(-x') = -2\alpha$
- …
- $\theta_T{}' = \theta_{T-1}{}' - \alpha(-x') = -T\alpha$

Tightness of the upper bound

For convex loss ℓ_1

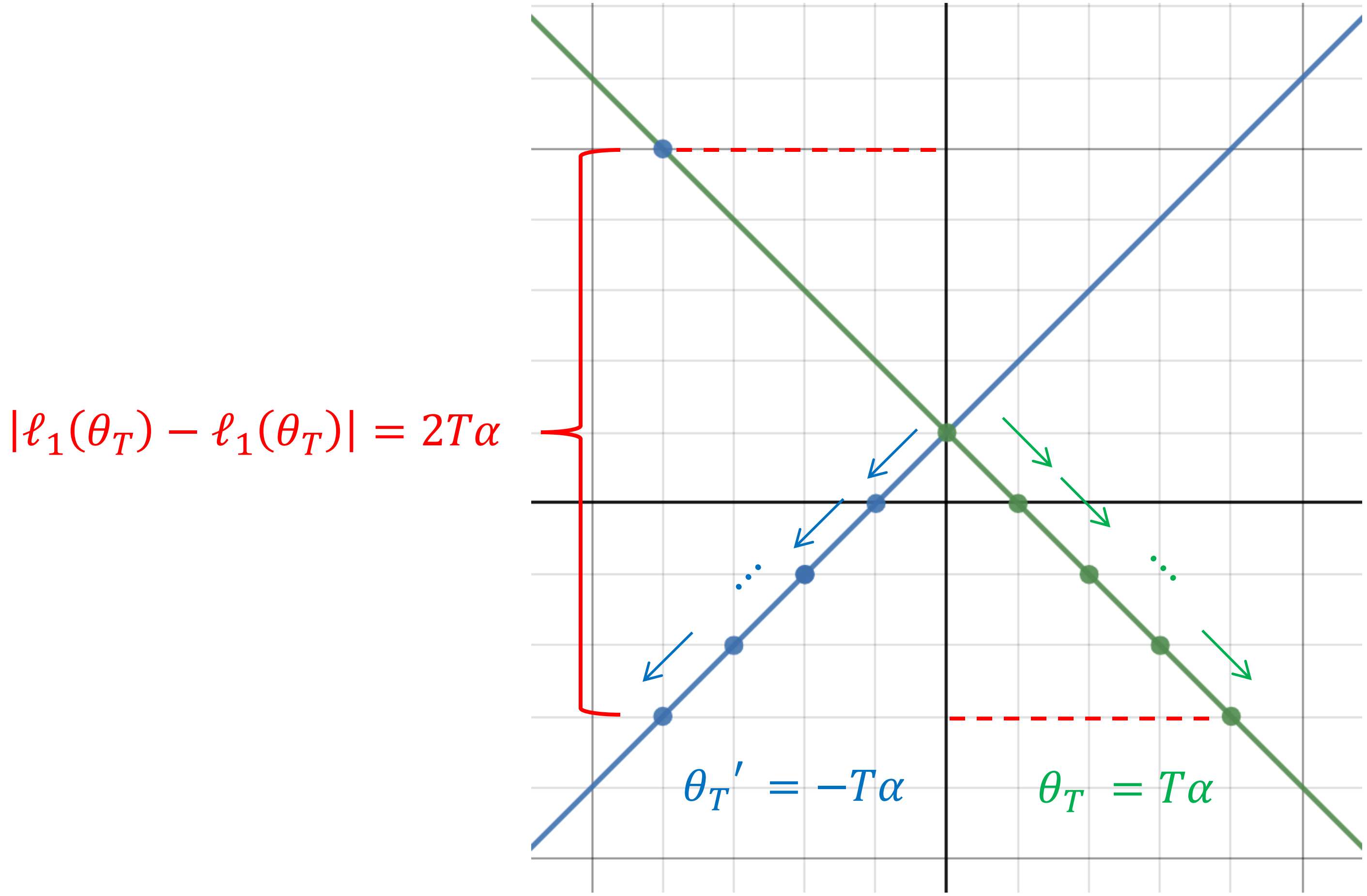
- $\ell_1 = y - \theta x$, $\nabla \ell_1 = -x$.
- Trained on $S = \{(1,1)\}$
- $\theta_1 = \theta_0 - \alpha(-x) = \alpha$
- $\theta_2 = \theta_1 - \alpha(-x) = 2\alpha$
- …
- $\theta_T = \theta_{T-1} - \alpha(-x) = T\alpha$

- Trained on $S' = \{(-1,1)\}$
- $\theta_1{}' = \theta_0 - \alpha(-x') = -\alpha$
- $\theta_2{}' = \theta_1{}' - \alpha(-x') = -2\alpha$
- …
- $\theta_T{}' = \theta_{T-1}{}' - \alpha(-x') = -T\alpha$

- Tested on $z = (1,1)$, $|\ell_1(\theta_T; z) - \ell_1(\theta_T; z)| = |1 - T\alpha - (1 + T\alpha)| = 2T\alpha$.

Tightness of the upper bound

For convex loss ℓ_1



- $\epsilon_{\text{stab}} = \Theta\left(\frac{T}{n}\right)$

Tightness of the upper bound

For strongly convex loss ℓ_2

- $\ell_2 = \frac{1}{2}(y - \theta x)^2, \nabla \ell_2 = (y - \theta x)(-x)$.
- Trained on $S = \{(1,1)\}$
- $\theta_1 = \theta_0 - \alpha(1 - \theta_0)(-1) = \alpha$
- $\theta_2 = \theta_1 - \alpha(1 - \theta_1)(-1)$
  $= (1 - \alpha)\alpha + \alpha$
- …
- $\theta_T = \alpha \sum_{t=1}^{T} (1 - \alpha)^{t-1}$
  $= 1 - (1 - \alpha)^T$

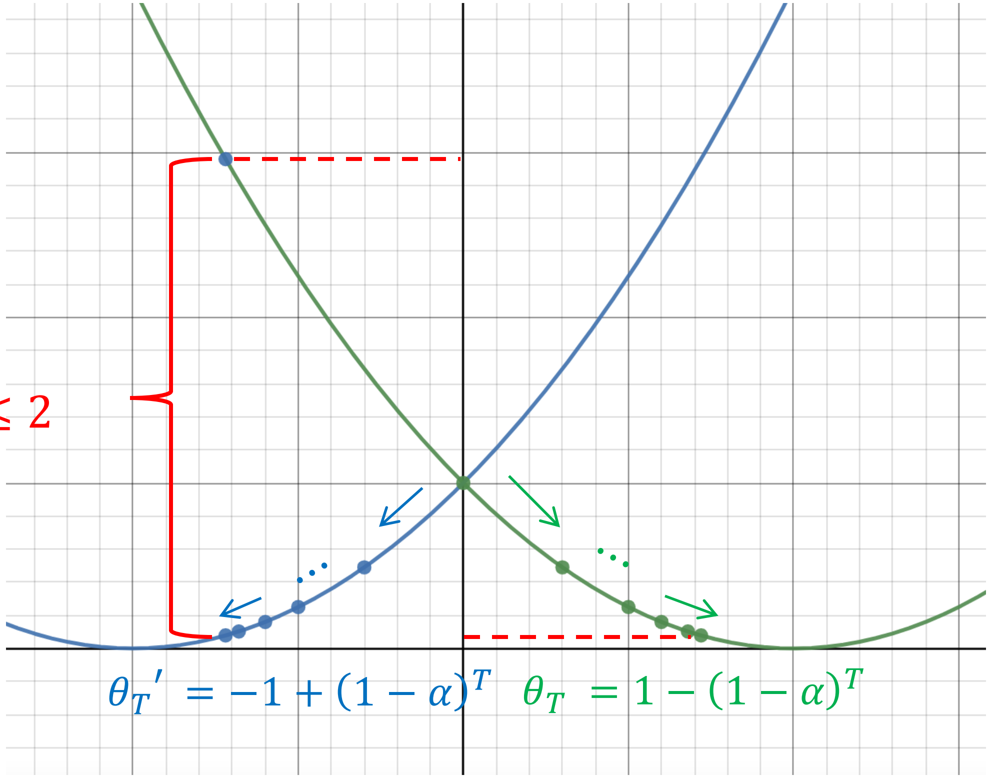
- Trained on $S' = \{(-1,1)\}$
- $\theta_1{}' = \theta_0 - \alpha(1 + \theta_0)(1) = -\alpha$
- $\theta_2 = \theta_1 - \alpha(1 + \theta_1)(1)$
  $= -(1 - \alpha)\alpha - \alpha$
- …
- $\theta_T{}' = -\alpha \sum_{t=1}^{T} (1 - \alpha)^{t-1}$
  $= -1 + (1 - \alpha)^T$

- Tested on $z = (1,1), |\ell_2(\theta_T; z) - \ell_2(\theta_T; z)| = |2\theta_T| = 2 - 2(1 - \alpha)^T$.
- $\epsilon_{\text{stab}} = \Omega(1) \Longleftrightarrow \epsilon_{\text{stab}} = \mathcal{O}\left(\frac{1}{n}\right)$

Tightness of the upper bound

For strongly convex loss ℓ_2

$|\ell_1(\theta_T) - \ell_1(\theta_T)|$
$= 2 - 2(1-\alpha)^T \leq 2$



- $\epsilon_{\text{stab}} = \Theta\left(\frac{1}{n}\right)$

Tightness of the upper bound

For non-convex loss ℓ_3

- $\ell_3 = -\frac{1}{2}(y - \theta x)^2, \nabla \ell_2 = (y - \theta x)(x)$.
- Trained on $S = \{(1,1)\}$
- $\theta_1 = \theta_0 - \alpha(1 - \theta_0)(1) = -\alpha$
- $\theta_2 = \theta_1 - \alpha(1 - \theta_1)(1)$
  $= -(1 + \alpha)\alpha - \alpha$
- …
- $\theta_T = -\alpha \sum_{t=1}^{T} (1 + \alpha)^{t-1}$
  $= -(1 + \alpha)^T + 1$

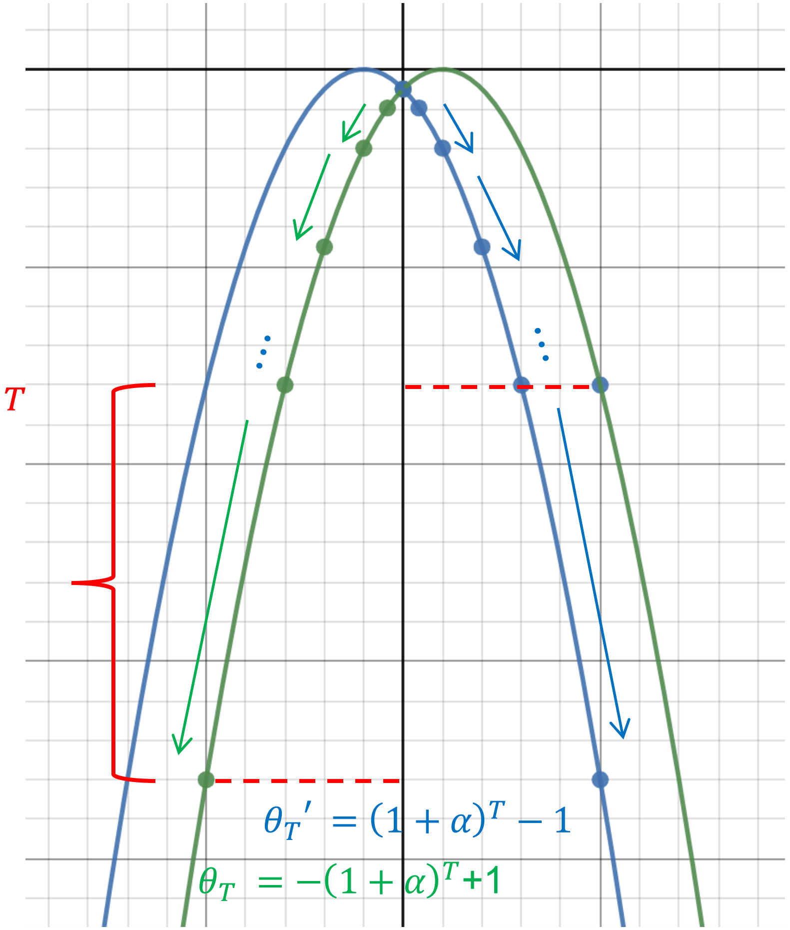
- Trained on $S' = \{(-1,1)\}$
- $\theta_1{}' = \theta_0 - \alpha(1 + \theta_0)(-1) = \alpha$
- $\theta_2 = \theta_1 - \alpha(1 + \theta_1)(-1)$
  $= (1 + \alpha)\alpha + \alpha$
- …
- $\theta_T{}' = \alpha \sum_{t=1}^{T} (1 + \alpha)^{t-1}$
  $= (1 + \alpha)^T - 1$

- Tested on $z = (1,1), |\ell_3(\theta_T; z) - \ell_3(\theta_T; z)| = |-2\theta_T| = 2(1 + \alpha)^T - 2$.
- $\epsilon_{\text{stab}} = \Omega((1 + \alpha)^T)$ $\Longleftrightarrow$ $\epsilon_{\text{stab}} = \mathcal{O}(\frac{(1+\alpha\beta)^T}{n})$

Tightness of the upper bound

For non-convex loss ℓ_3

$|\ell_3(\theta_T) - \ell_3(\theta_T)| = 2(1+\alpha)^T$



- $\epsilon_{\text{stab}} = \Theta\left(\frac{(1+\alpha\beta)^T}{n}\right)$

Stability of SGD

- Totally the same if we *take the expectation* and use the concept of uniform stability for *stochastic* algorithms.
- Property of update rules for SGD:
  - non-convex loss: $\mathbb{E}_{\mathcal{A}} \|G(\theta) - G(\theta')\| \leq (1 + \kappa_1)\mathbb{E}_{\mathcal{A}}\|\theta - \theta'\|$
  - convex loss: $\mathbb{E}_{\mathcal{A}} \|G(\theta) - G(\theta')\| \leq \mathbb{E}_{\mathcal{A}}\|\theta - \theta'\|$
  - γ-strongly convex loss: $\mathbb{E}_{\mathcal{A}} \|G(\theta) - G(\theta')\| \leq (1 - \kappa_2)\mathbb{E}_{\mathcal{A}}\|\theta - \theta'\|$

Stability of SGD

- Totally the same if we *take the expectation* and use the concept of uniform stability for *stochastic* algorithms.
- Property of update rules for SGD:
  - non-convex loss: $\mathbb{E}_{\mathcal{A}} \|G(\theta) - G(\theta')\| \leq (1 + \kappa_1)\mathbb{E}_{\mathcal{A}}\|\theta - \theta'\|$
  - convex loss: $\mathbb{E}_{\mathcal{A}} \|G(\theta) - G(\theta')\| \leq \mathbb{E}_{\mathcal{A}}\|\theta - \theta'\|$
  - γ-strongly convex loss: $\mathbb{E}_{\mathcal{A}} \|G(\theta) - G(\theta')\| \leq (1 - \kappa_2)\mathbb{E}_{\mathcal{A}}\|\theta - \theta'\|$
- Stability of SGD: with constant learning rate,
  - Non-convex case: $\epsilon_{\text{stab}} = \mathcal{O}(\frac{(1+\alpha\beta)^T}{n})$,
  - Convex case: $\epsilon_{\text{stab}} = \mathcal{O}\left(\frac{T}{n}\right)$,
  - γ-Strongly convex case: $\epsilon_{\text{stab}} = \mathcal{O}(\frac{1}{n})$.

Stability of SGD

Diminishing learning rate

- But things may become different if we tune the learning rate…
- Recall that in the non-convex case, we have $\epsilon_{\text{stab}} = \mathcal{O}\left(\frac{(1+\alpha\beta)^T}{n}\right)$ with constant learning rate. It explodes quickly!
- Can we control this error with a diminishing learning rate?
- Yes!

Stability of SGD

Diminishing learning rate

- But things may become different if we tune the learning rate…
- Recall that in the non-convex case, we have $\epsilon_{\text{stab}} = \mathcal{O}\left(\frac{(1+\alpha\beta)^T}{n}\right)$ with constant learning rate. It explodes quickly!
- Can we control this error with a diminishing learning rate?
- Yes!

- **Theorem 2.1 (Stability of GD with Diminishing Learning Rate):** Suppose the loss function ℓ is β-smooth, implement T-step GD with $\alpha_t \leq \frac{c}{t}$, where c is a constant. Then, the stability coefficient is less than

$$\epsilon_{\text{stab}} \leq \frac{2L^2}{(n-1)\beta} T^{c\beta} = \mathcal{O}(\frac{T^{c\beta}}{n}).$$

Stability of SGD

Diminishing learning rate

- And SGD performs better than GD in this case!

- **Theorem 2.1 (Stability of GD with Diminishing Learning Rate):** Suppose the loss function ℓ is β-smooth, implement T-step GD with $\alpha_t \le \frac{c}{t}$, where c is a constant. Then, the stability coefficient is less than

$$\epsilon_{\text{stab}} \le \frac{2L^2}{(n-1)\beta} T^{c\beta} = \mathcal{O}\left(\frac{T^{c\beta}}{n}\right).$$

- **Theorem 2.2 (Stability of SGD with Diminishing Learning Rate):** Suppose the loss function ℓ is β-smooth, implement T-step GD with $\alpha_t \le \frac{c}{t}$, where c is a constant. Then, the stability coefficient is less than

$$\epsilon_{\text{stab}} \le \frac{2L^{2\frac{1}{1+c\beta}}}{(n-1)(1+\frac{1}{c\beta})} T^{\frac{c\beta}{1+c\beta}} = \mathcal{O}\left(\frac{T^{\frac{c\beta}{1+c\beta}}}{n}\right).$$

Stability of SGD

Diminishing learning rate

- GD: $\epsilon_{\text{stab}} = \mathcal{O}\left(\frac{T^{c\beta}}{n}\right)$, SGD: $\epsilon_{\text{stab}} = \mathcal{O}\left(\frac{T^{\frac{c\beta}{1+c\beta}}}{n}\right)$.
- Why? Stochastic in the random sampling for each batch.
  - SGD *delays* encountering different samples.
  - When encountering different samples, the learning rate has decayed to be not too large.

Experiment	Mini-batching	Epochs	Steps	Modifications	Val. Acc.%
Baseline SGD	✓	300	117,000	-	95.70($\pm$0.11)
SGD regularized	✓	300	117’000	reg	95.81($\pm$0.18)
Baseline FB	✗	300	300	-	75.42($\pm$0.13)
FB train longer	✗	3000	3000	-	87.36($\pm$1.23)
FB clipped	✗	3000	3000	clip	93.85($\pm$0.10)
FB regularized	✗	3000	3000	clip+reg	95.54($\pm$0.09)
FB strong reg.	✗	3000	3000	clip+reg+bs32	95.68($\pm$0.09)
FB in practice	✗	3000	3000	clip+reg+bs32+shuffle	95.91($\pm$0.14)

Table 1: Validation accuracies on the CIFAR-10 validation set for each experiment with data augmentations considered in Section 3. All validation accuracies are averaged over 5 runs.

Geiping, Jonas, et al. "Stochastic training is not necessary for generalization." *arXiv preprint arXiv:2109.14119* (2021).

Takeaway

Stability

- Stability of GD/SGD with constant learning rate:
  - Non-convex case: $\epsilon_{\mathrm{stab}} = \mathcal{O}(\frac{(1+\alpha\beta)^T}{n})$,
  - Convex case: $\epsilon_{\mathrm{stab}} = \mathcal{O}\left(\frac{T}{n}\right)$,
  - γ-Strongly convex case: $\epsilon_{\mathrm{stab}} = \mathcal{O}(\frac{1}{n})$.

Takeaway

Stability

- Stability of GD/SGD with constant learning rate:
  - Non-convex case: $\epsilon_{\text{stab}} = \mathcal{O}(\frac{(1+\alpha\beta)^T}{n})$,
  - Convex case: $\epsilon_{\text{stab}} = \mathcal{O}\left(\frac{T}{n}\right)$,
  - γ-Strongly convex case: $\epsilon_{\text{stab}} = \mathcal{O}(\frac{1}{n})$.
- Stability with linearly diminishing learning rate:
  - GD: $\epsilon_{\text{stab}} = \mathcal{O}\left(\frac{T^{c\beta}}{n}\right)$,
  - SGD: $\epsilon_{\text{stab}} = \mathcal{O}(\frac{T^{\frac{c\beta}{1+c\beta}}}{n})$.

SGD may generalize better than GD in some cases.

Takeaway

Some techniques with theoretical guarantee

- Stability-inducing operations
  - Weight decay/Regularization:

    good for the update rules

    $$\|G(\theta) - G(\theta')\| \leq (1 + \alpha(\beta - \mu))\|\theta - \theta'\|$$

  - Dropout/Projection:

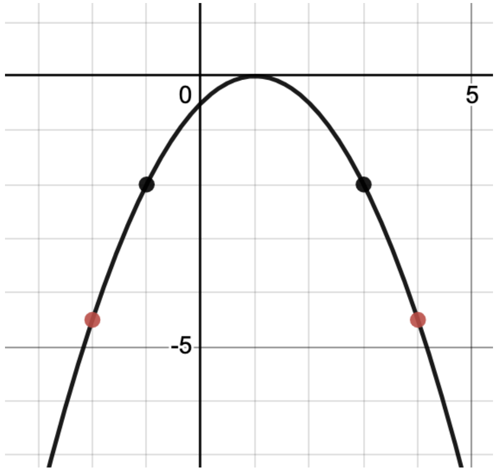
    reduce the update

    $$\epsilon_{\mathrm{stab}} \leq \frac{2sL^2}{n\beta}[(1 + \alpha\beta)^T - 1], 0 < s < 1$$

Takeaway

Open problems

- The main concern lies in the non-convex case.
  - Constant learning rate: $\epsilon_{\text{stab}} = \mathcal{O}(\frac{(1+\alpha\beta)^T}{n})$,
  - Linearly diminishing learning rate: $\epsilon_{\text{stab}} = \mathcal{O}(\frac{T^{\frac{c\beta}{1+c\beta}}}{n})$.
- Some problems:
  - The assumption for loss function is too general.
  - Need to consider convergence to local minimum?



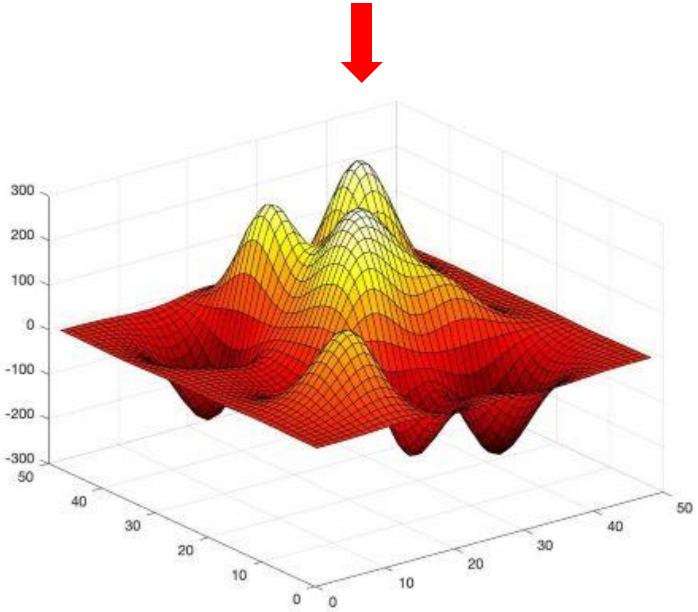


Image Credit: https://blogs.mathworks.com/deep-learning/2020/12/10/using-deep-learning-for-complex-physical-processes/

Takeaway

Open problems

- The main concern lies in the non-convex case.
  - Constant learning rate: $\epsilon_{\text{stab}} = \mathcal{O}(\frac{(1+\alpha\beta)^T}{n})$,
  - Linearly diminishing learning rate: $\epsilon_{\text{stab}} = \mathcal{O}(\frac{T^{\frac{c\beta}{1+c\beta}}}{n})$.
- Some problems:
  - The assumption for loss function is too general.
  - Need to consider convergence to local minimum?
- Some thoughts…
  - Consider more informative assumptions for the loss.
  - A better bound with sub-linear diminishing learning rate.

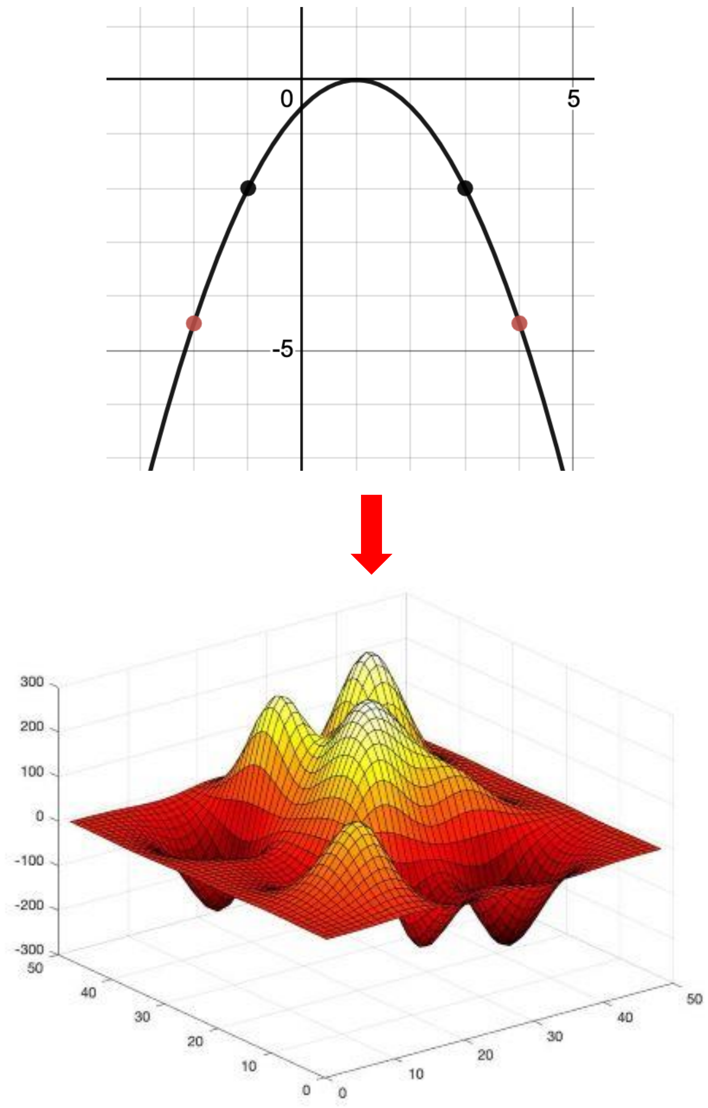
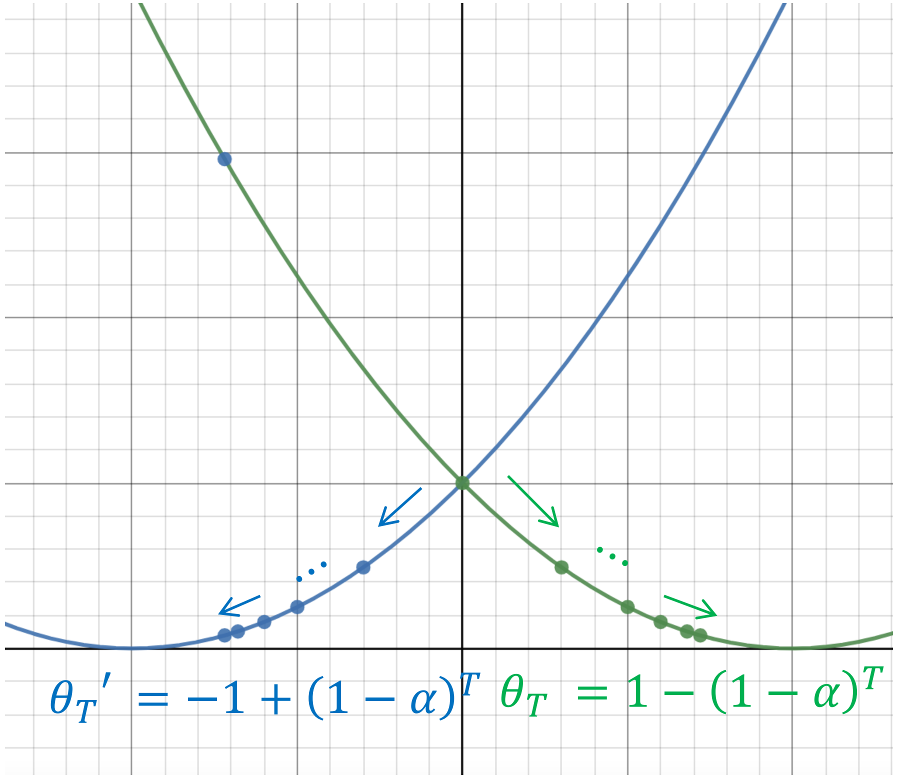
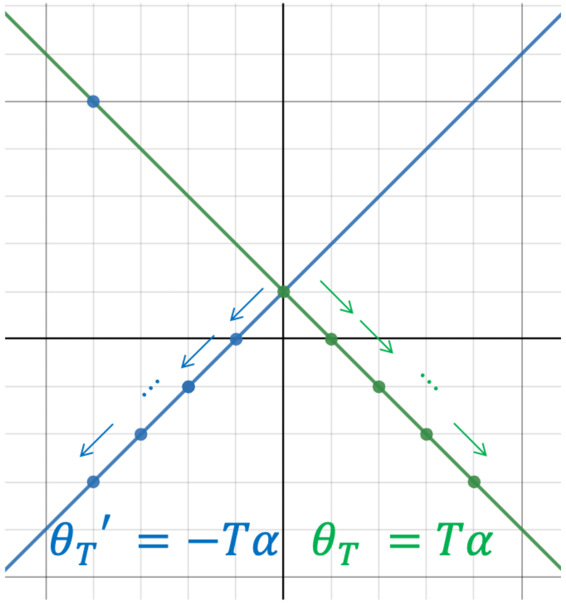


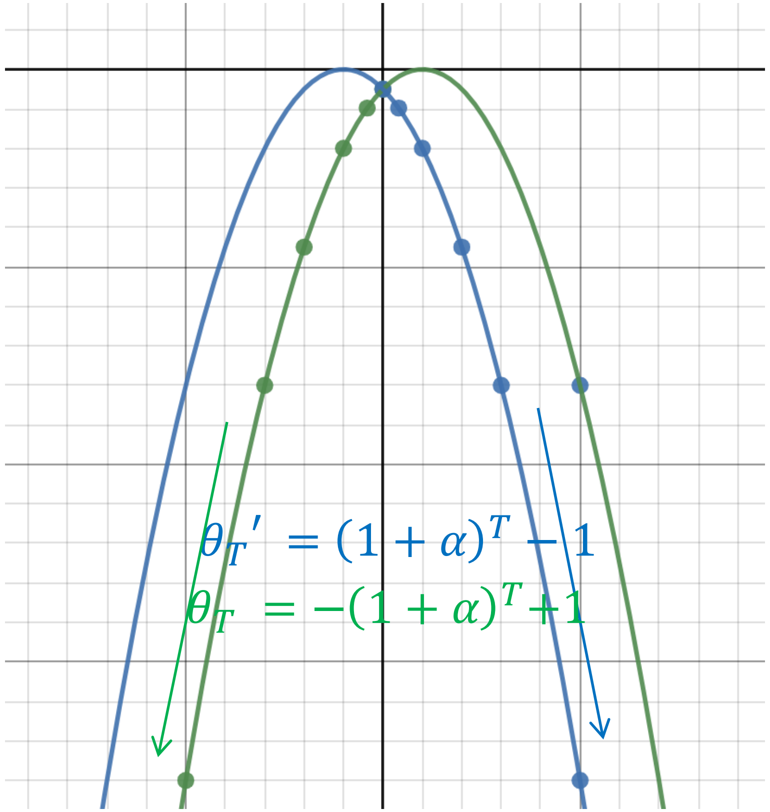
Image Credit: https://blogs.mathworks.com/deep-learning/2020/12/10/using-deep-learning-for-complex-physical-processes/



$$\epsilon_{\text{stab}} = \Theta\left(\frac{1}{n}\right)$$



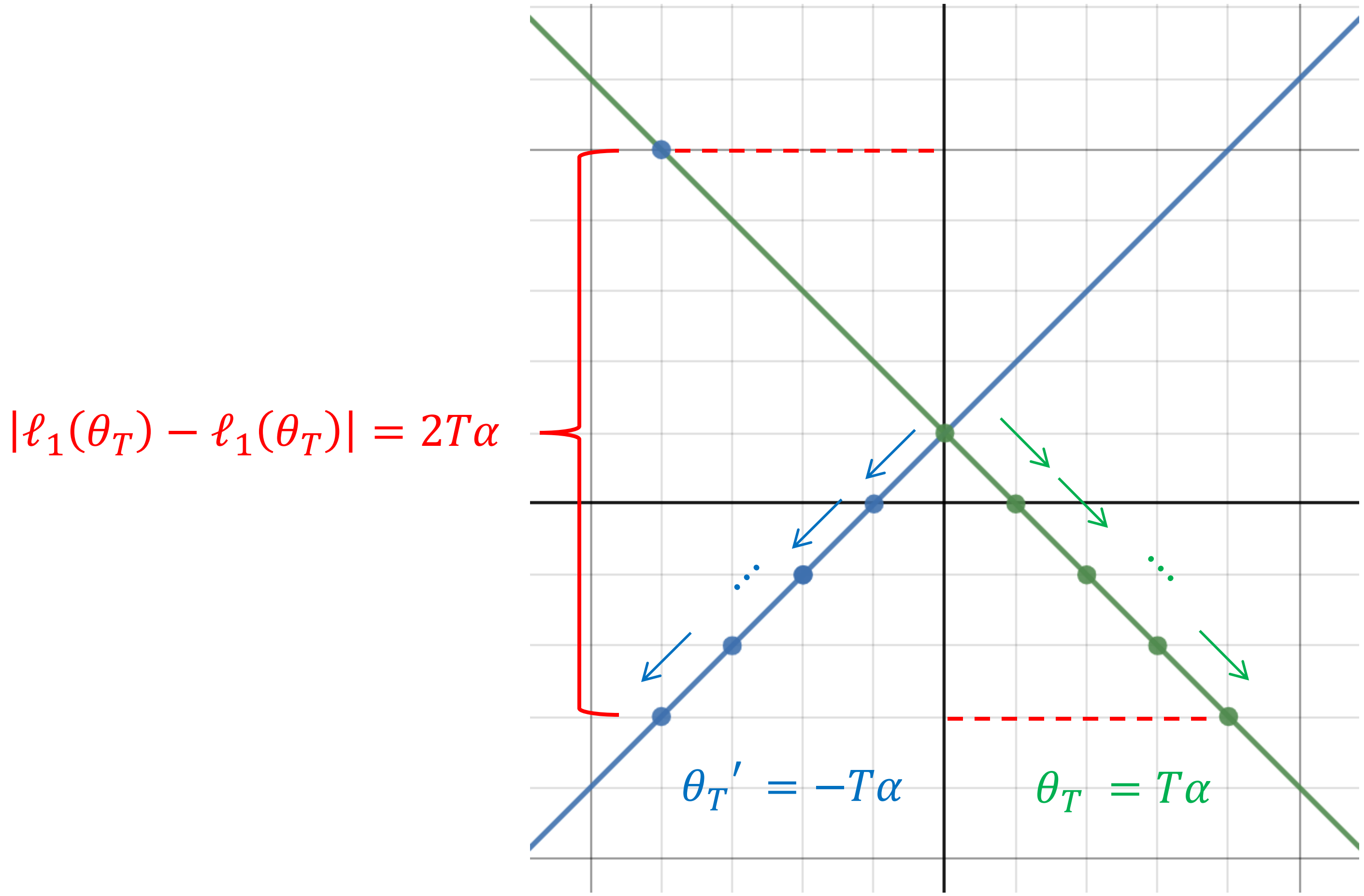
$$\epsilon_{\text{stab}} = \Theta\left(\frac{T}{n}\right)$$



$$\epsilon_{\text{stab}} = \Theta\left(\frac{(1+\alpha\beta)^T}{n}\right)$$

Tightness of the upper bound

For convex loss ℓ_1

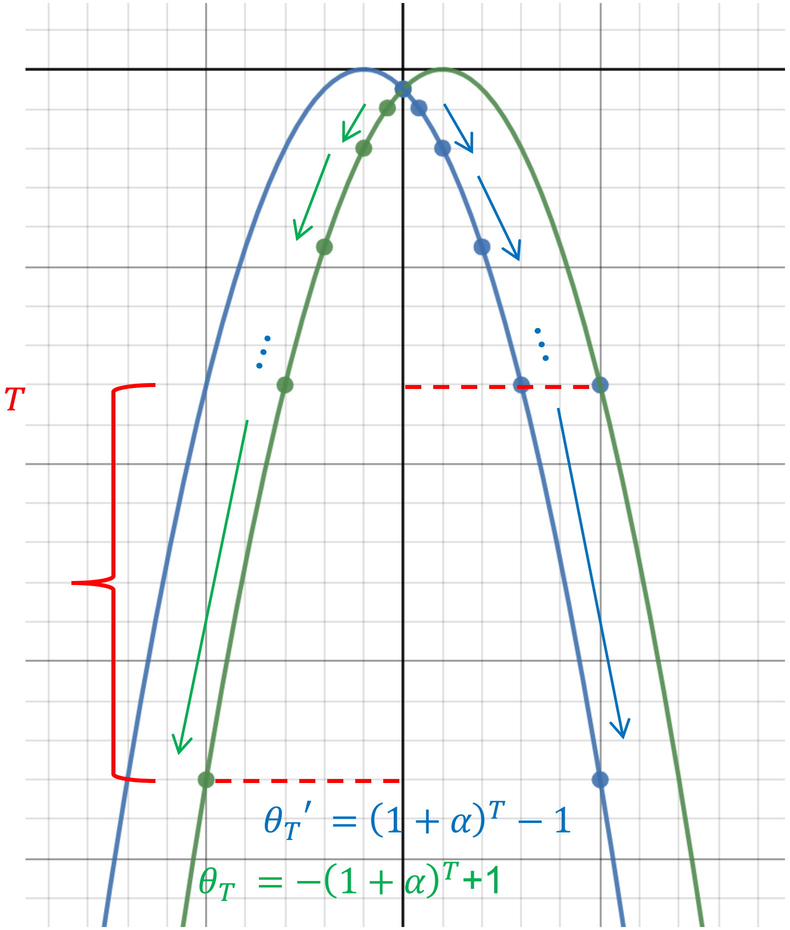


- $\epsilon_{\text{stab}} = \Theta\left(\frac{T}{n}\right)$

Tightness of the upper bound

For non-convex loss ℓ_3

$|\ell_3(\theta_T) - \ell_3(\theta_T)| = 2(1+\alpha)^T$



- $\epsilon_{\mathrm{stab}} = \Theta\left(\frac{(1+\alpha\beta)^T}{n}\right)$