Optimization for Deep Learning

Lecture 4-1: Projected Gradient Descent

Kun Yuan

Peking University

Accelerated GD review

Method	Convexity	Rate	Complexity
GD	Non-convex	$O(L/k)$	$O(L/\epsilon)$
	Convex	$O(L/k)$	$O(L/\epsilon)$
	Strongly convex	$O((1-\frac{\mu}{L})^k)$	$O(\frac{L}{\mu}\log(1/\epsilon))$
NAG	Non-convex	$O(L/k)$	$O(L/\epsilon)$
	Convex	$O(L/k^2)$	$O(L/\sqrt{\epsilon})$
	Strongly convex	$O((1-\sqrt{\frac{\mu}{L}})^k)$	$O(\sqrt{\frac{L}{\mu}}\log(1/\epsilon))$
Lower bound	Non-convex	$\Omega(L/k)$	$\Omega(L/\epsilon)$
	Convex	$\Omega(L/k^2)$	$\Omega(L/\sqrt{\epsilon})$
	Strongly convex	$\Omega((1-\sqrt{\frac{\mu}{L}})^k)$	$\Omega(\sqrt{\frac{L}{\mu}}\log(1/\epsilon))$

Main contents in this lecture

- Projection
- Projected gradient descent
- Convergence properties

Projection onto closed convex sets

- Given a closed convex set $\mathcal{C} \subseteq \mathbb{R}^d$, for any $z \in \mathbb{R}^d$, we define

$$\mathcal{P}_{\mathcal{C}}[z] := \arg\min_{x \in \mathcal{C}} \{\|z - x\|\}$$

as the projection onto set $\mathcal{C}$.

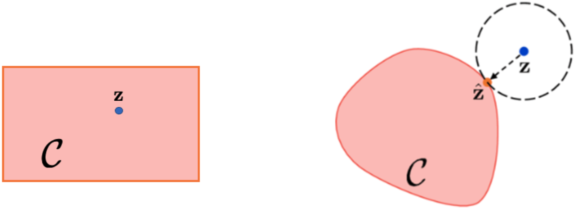

Figure: Projection onto convex sets $\mathcal{C}$[1]

[1] The right plot is from (Jain et al., 2017)

Projection onto closed convex sets

Lemma 1

Given a closed convex set $\mathcal{C} \subseteq \mathbb{R}^d$, we let $\hat{z} = \mathcal{P}_{\mathcal{C}}[z]$ for any $z \in \mathbb{R}^d$. It holds that $\hat{z}$ exists and is unique, i.e., there exists a unique $\hat{z} \in \mathcal{C}$ such that

$$\|z - \hat{z}\| \leq \|z - x\| \quad \forall x \in \mathcal{C}.$$

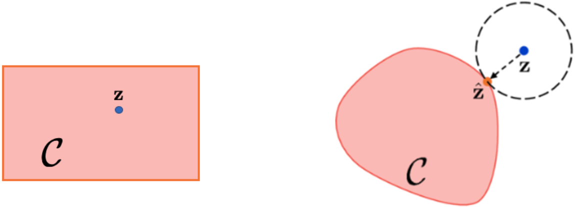

Figure: Projection onto convex sets $\mathcal{C}$

Projection onto closed convex sets

Lemma 2

Let $\mathcal{C} \subseteq \mathbb{R}^d$ be a closed convex set, then for any $z \in \mathbb{R}^d$, we have $\hat{z} = \mathcal{P}_C[z]$ if and only if $\langle z - \hat{z}, x - \hat{z} \rangle \leq 0$ for any $x \in \mathcal{C}$.

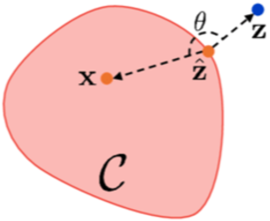

Figure: Illustration of Lemma 2[2]

[2] This plot is from (Jain et al., 2017)

Projection onto closed convex sets

Lemma 3

Let $\mathcal{C} \subseteq \mathbb{R}^d$ be a closed convex set. For any $x, y \in \mathbb{R}^d$, it holds that

$$\|\mathcal{P}_{\mathcal{C}}[x] - \mathcal{P}_{\mathcal{C}}[y]\| \leq \|x - y\|.$$

It implies that the projection operator is non-expansive.

Projection examples

- Box: $\mathcal{C} = [\eta_1, \eta_2]^d$. For any $x \in \mathbb{R}^d$, we have

$$(\mathcal{P}_{\mathcal{C}}[x])_i = (\max\{\eta_1, \min\{x_i, \eta_2\}\})$$

- Hyperplane: $\mathcal{C} = \{x \mid u^\top x = \eta, u \in \mathbb{R}^d, \eta \in \mathbb{R}\}$. For any $x \in \mathbb{R}^d$, we have

$$\mathcal{P}_{\mathcal{C}}[x] = x + \frac{\eta - u^\top x}{\|u\|_2^2} u$$

- Probability simplex: $\mathcal{C} = \{x \mid 1^\top x = 1, x \geq 0\}$. There are $O(d)$ algorithms to achieve projection onto this set (Duchi et al., 2008).

- Norm-ball constraints: $\mathcal{C} = \{\|x\|_1 \leq \tau\}$. There are fast algorithms to achieve projection onto this set (Duchi et al., 2008).

Constrained optimization

- Consider the following constrained minimization problem

$$\min_{x \in \mathbb{R}^d} \quad f(x) \quad \text{subject to} \quad x \in \mathcal{X}$$

- We assume $\mathcal{X}$ is a closed convex set

- We assume $f(x)$ is differentible so that $\nabla f(x)$ exists for any $x \in \mathbb{R}^d$

Application: Anderson acceleration subproblem

- Let $G^k = [\nabla f(x_k), \cdots, \nabla f(x_{k-m})] \in \mathbb{R}^{d\times(m+1)}$, Anderson acceleration is

$$\boldsymbol{\alpha}^k = \operatorname*{arg\,min}_{\boldsymbol{\alpha}\geq 0 : \mathbf{1}^\top \boldsymbol{\alpha}=1} \{\|G^k \boldsymbol{\alpha}\|^2\} \tag{1}$$

$$x_{k+1} = \sum_{i=0}^{m} \alpha_i^k x_{k-i}$$

- Problem (1) is a constrained minimization problem.

Application: Adversarial attacks on deep neural network

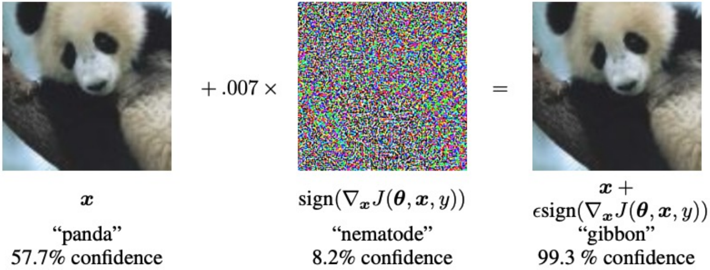

Figure: A demonstration of the adversarial example [Goodfellow et.al., 2015].

Application: Adversarial attacks on deep neural network

- An adversarial example is a perturbation η to maximize misclassification
- Given an input pair (ξ, y), its adversarial example $\eta \in \mathbb{R}^d$ is defined as

$$\eta = \underset{\eta : \|\eta\| \le \epsilon}{\arg\max}\, L\Big(h(x^\star, \xi + \eta), y\Big)$$

  where $x^\star$ is the optimal DNN model.

- The above problem is a constrained maximization problem.

Projected gradient descent

- Given any initialization $x_0 \in \mathcal{X}$, projected gradient descent iterates as

$$y_{k+1} = x_k - \gamma \nabla f(x_k),$$
$$x_{k+1} = \mathcal{P}_{\mathcal{X}}[y_{k+1}].$$

- How fast does it converge? Is it slower than gradient descent for unconstrained optimization?

Projected gradient descent: an illustration

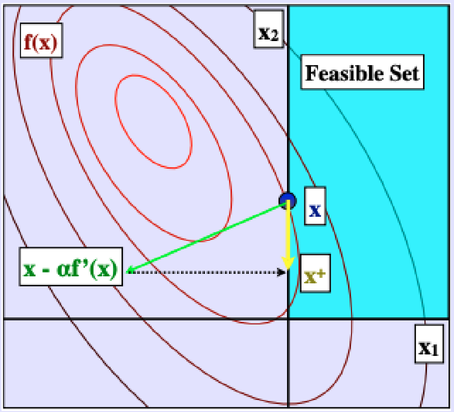

Figure: Projected GD progresses to the optimal solution[3].

[3] This figure is from Prof. Mark Schmidt's lecture on projected gradient descent.

Smooth and convex scenario

Lemma 4

Suppose $f(x)$ is L-smooth. If $\gamma = \frac{1}{L}$, then the sequence generated by projected gradient descent with arbitrary $x_0 \in \mathcal{X}$ satisfies

$$f(x_{k+1}) \leq f(x_k) - \frac{L}{2}\|x_{k+1} - x_k\|^2, \quad k = 0, 1, 2, \cdots$$

The sequence is monotonically decreasing

Smooth and convex scenario

Theorem 1

Suppose $f(x) : \mathbb{R}^d \to \mathbb{R}$ is L-smooth. If $\gamma = \frac{1}{L}$, then the sequence generated by projected gradient descent with arbitrary $x_0 \in \mathcal{X}$ satisfies

$$f(x_K) - f(x^\star) \leq \frac{L}{2K} \|x_0 - x^\star\|^2.$$

Projected GD has a rate $O(L/K)$, which amounts to complexity $O(L/\epsilon)$

It has the same order in rate and complexity as gradient descent

Smooth and strongly-convex scenario

Theorem 2

Let $f(x) : \mathbb{R}^d \to \mathbb{R}$ be differentiable, L-smooth and μ-strongly convex. If $\gamma = \frac{1}{L}$, projected gradient descent 2 with arbitrary $x_0 \in \mathcal{X}$ satisfies

$$\|x_K - x^\star\| \leq (1 - \frac{\mu}{L})^K \|x_0 - x^\star\|.$$

Projected GD has a rate $O((1 - \mu/L)^K)$ and a complexity $O(L/\mu \log(1/\epsilon))$

It has the same order in rate and complexity as gradient descent

Comparison between GD and projected GD

Method	Convexity	Rate	Complexity
GD	Non-convex	$O(L/k)$	$O(L/\epsilon)$
	Convex	$O(L/k)$	$O(L/\epsilon)$
	Strongly convex	$O((1-\frac{\mu}{L})^k)$	$O(\frac{L}{\mu}\log(1/\epsilon))$
Projected GD	Non-convex	$O(L/k)$	$O(L/\epsilon)$
	Convex	$O(L/k)$	$O(L/\epsilon)$
	Strongly convex	$O((1-\frac{\mu}{L})^k)$	$O(\frac{L}{\mu}\log(1/\epsilon))$

Projected GD converges as fast as GD even with the projection step. It makes sense since GD is a special algorithm of projected GD if $\mathcal{C}$ is $\mathbb{R}^d$.

Summary

- Optimizaiton with simple closed sets are common in applications, especially in deep learning.
- Projected GD is very useful when projection operation is cheap.
- Projected GD has the same convergence rate and complexity as GD.

References I

P. Jain, P. Kar *et al.*, "Non-convex optimization for machine learning," *Foundations and Trends® in Machine Learning*, vol. 10, no. 3-4, pp. 142–363, 2017.

J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra, "Efficient projections onto the l1-ball for learning in high dimensions," in *Proceedings of the 25th international conference on Machine learning*, 2008, pp. 272–279.