

# Strongly convex functions. Optimality conditions.

## Seminar

Optimization for ML. Faculty of Computer Science. HSE University

# Convex Function

The function  $f(x)$ , which is defined on the convex set  $S \subseteq \mathbb{R}^n$ , is called **convex** on  $S$ , if:

$$f(\lambda x_1 + (1 - \lambda)x_2) \leq \lambda f(x_1) + (1 - \lambda)f(x_2)$$

for any  $x_1, x_2 \in S$  and  $0 \leq \lambda \leq 1$ .

If the above inequality holds as strict inequality  $x_1 \neq x_2$  and  $0 < \lambda < 1$ , then the function is called **strictly convex** on  $S$ .

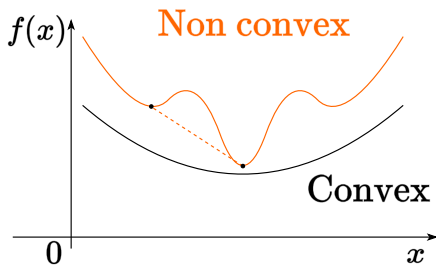


Figure 1: Difference between convex and non-convex function

# Strong Convexity

$f(x)$ , defined on the convex set  $S \subseteq \mathbb{R}^n$ , is called  $\mu$ -strongly convex (strongly convex) on  $S$ , if:

$$f(\lambda x_1 + (1 - \lambda)x_2) \leq \lambda f(x_1) + (1 - \lambda)f(x_2) - \frac{\mu}{2}\lambda(1 - \lambda)\|x_1 - x_2\|^2$$

for any  $x_1, x_2 \in S$  and  $0 \leq \lambda \leq 1$  for some  $\mu > 0$ .

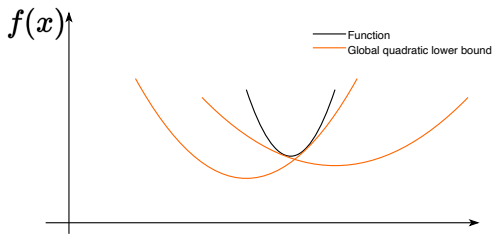


Figure 2: Strongly convex function is greater or equal than global quadratic lower bound at any point

## First-order differential criterion of convexity

The differentiable function  $f(x)$  defined on the convex set  $S \subseteq \mathbb{R}^n$  is convex if and only if  $\forall x, y \in S$ :

$$f(y) \geq f(x) + \nabla f^T(x)(y - x)$$

Let  $y = x + \Delta x$ , then the criterion will become more tractable:

$$f(x + \Delta x) \geq f(x) + \nabla f^T(x)\Delta x$$



## Second-order differential criterion of strong convexity

Twice differentiable function  $f(x)$  defined on the convex set  $S \subseteq \mathbb{R}^n$  is  $\mu$ -strongly convex if and only if  $\forall x \in \text{int}(S) \neq \emptyset$ :

$$\nabla^2 f(x) \succeq \mu I$$

In other words:

$$\langle y, \nabla^2 f(x)y \rangle \geq \mu \|y\|^2$$

# Motivational Experiment with JAX

Why convexity and strong convexity is important? Check the simple code snippet.

# Problem 1

## Question

Show, that  $f(x) = \|x\|$  is convex on  $\mathbb{R}^n$ .

## Question

Show, that  $f(x) = x^\top Ax$ , where  $A \succeq 0$  - is convex on  $\mathbb{R}^n$ .

## Problem 2

### Question

Show, that if  $f(x)$  is convex on  $\mathbb{R}^n$ , then  $\exp(f(x))$  is convex on  $\mathbb{R}^n$ .



## Problem 3

### Question

If  $f(x)$  is convex nonnegative function and  $p \geq 1$ . Show that  $g(x) = f(x)^p$  is convex.

## Problem 4

### Question

Show that, if  $f(x)$  is concave positive function over convex  $S$ , then  $g(x) = \frac{1}{f(x)}$  is convex.

### Question

Show, that the following function is convex on the set of all positive denominators

$$f(x) = \frac{1}{x_1 - \frac{1}{x_2 - \frac{1}{x_3 - \frac{1}{\dots}}}}, x \in \mathbb{R}^n$$

## Problem 5

### Question

Let  $S = \{x \in \mathbb{R}^n \mid x \succ 0, \|x\|_\infty \leq M\}$ . Show that  $f(x) = \sum_{i=1}^n x_i \log x_i$  is  $\frac{1}{M}$ -strongly convex.

# Polyak-Lojasiewicz (PL) Condition

PL inequality holds if the following condition is satisfied for some  $\mu > 0$ ,

$$\|\nabla f(x)\|^2 \geq \mu(f(x) - f^*) \forall x$$

The example of a function, that satisfies the PL-condition, but is not convex.

$$f(x, y) = \frac{(y - \sin x)^2}{2}$$

Example of PL non-convex function  Open in Colab.

# Optimality Conditions. Important notions recap

$$f(x) \rightarrow \min_{x \in S}$$

A set  $S$  is usually called a budget set.

- A point  $x^*$  is a global minimizer if  $f(x^*) \leq f(x)$  for all  $x$ .
- A point  $x^*$  is a local minimizer if there exists a neighborhood  $N$  of  $x^*$  such that  $f(x^*) \leq f(x)$  for all  $x \in N$ .
- A point  $x^*$  is a strict local minimizer (also called a strong local minimizer) if there exists a neighborhood  $N$  of  $x^*$  such that  $f(x^*) < f(x)$  for all  $x \in N$  with  $x \neq x^*$ .
- We call  $x^*$  a stationary point (or critical) if  $\nabla f(x^*) = 0$ . Any local minimizer must be a stationary point.

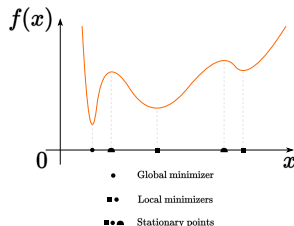


Figure 4: Illustration of different stationary (critical) points

# Unconstrained optimization recap

## 💡 First-Order Necessary Conditions

If  $x^*$  is a local minimizer and  $f$  is continuously differentiable in an open neighborhood, then

$$\nabla f(x^*) = 0 \quad (1)$$

## 💡 Second-Order Sufficient Conditions

Suppose that  $\nabla^2 f$  is continuous in an open neighborhood of  $x^*$  and that

$$\nabla f(x^*) = 0 \quad \nabla^2 f(x^*) \succ 0. \quad (2)$$

Then  $x^*$  is a strict local minimizer of  $f$ .

# Lagrange multipliers recap

Consider simple yet practical case of equality constraints:

$$\begin{aligned} f(x) &\rightarrow \min_{x \in \mathbb{R}^n} \\ \text{s.t. } h_i(x) &= 0, i = 1, \dots, p \end{aligned}$$

The basic idea of Lagrange method implies the switch from conditional to unconditional optimization through increasing the dimensionality of the problem:

$$L(x, \nu) = f(x) + \sum_{i=1}^p \nu_i h_i(x) \rightarrow \min_{x \in \mathbb{R}^n, \nu \in \mathbb{R}^p}$$

# Problem 1

## Question

Function  $f : E \rightarrow \mathbb{R}$  is defined as

$$f(x) = \ln(-Q(x))$$

where  $E = \{x \in \mathbb{R}^n : Q(x) < 0\}$  and

$$Q(x) = \frac{1}{2}x^\top Ax + b^\top x + c$$

with  $A \in \mathbb{S}_{++}^n$ ,  $b \in \mathbb{R}^n$ ,  $c \in \mathbb{R}$ .

Find the maximizer  $x^*$  of the function  $f$ .



## Problem 2

### Question

Give an explicit solution of the following task.

$$\begin{aligned} \langle c, x \rangle + \sum_{i=1}^n x_i \log x_i &\rightarrow \min_{x \in \mathbb{R}^n} \\ \text{s.t. } \sum_{i=1}^n x_i &= 1, \end{aligned}$$

where  $x \in \mathbb{R}_{++}^n, c \neq 0$ .

# Adversarial Attacks as Constrained Optimization



Figure 5: Any neural network can be fooled with invisible perturbation

- Targetted Adversarial Attack:

$$\begin{aligned} \rho(x, x + r) &\rightarrow \min_{r \in \mathbb{R}^n} \\ \text{s.t. } y(x + r) &= \text{target\_class}, \end{aligned}$$

# Adversarial Attacks as Constrained Optimization

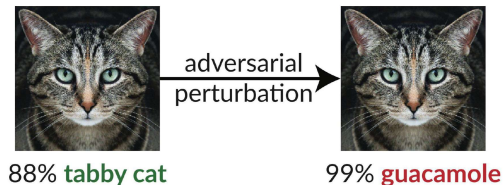


Figure 5: Any neural network can be fooled with invisible perturbation

- Targetted Adversarial Attack:

$$\begin{aligned} \rho(x, x + r) &\rightarrow \min_{r \in \mathbb{R}^n} \\ \text{s.t. } y(x + r) &= \text{target\_class}, \end{aligned}$$

- Non-targetted Adversarial Attack:

$$\begin{aligned} \rho(x, x + r) &\rightarrow \min_{r \in \mathbb{R}^n} \\ \text{s.t. } y(x + r) &= y(x), \end{aligned}$$

## Solution from Szegedy et al, “Intriguing properties of neural networks”

- Targetted Adversarial Attack Task:

$$\begin{aligned} \rho(x, x + r) &\rightarrow \min_{r \in \mathbb{R}^n} \\ \text{s.t. } y(x + r) &= \text{target\_class}, \end{aligned}$$

## Solution from Szegedy et al, “Intriguing properties of neural networks”

- Targetted Adversarial Attack Task:

$$\begin{aligned} \rho(x, x + r) &\rightarrow \min_{r \in \mathbb{R}^n} \\ \text{s.t. } y(x + r) &= \text{target\_class}, \end{aligned}$$

- Targetted Lagrange function  $L(r, c | x)$ :

$$\|r\|^2 - c \log p(y = \text{target\_class} | x + r) \rightarrow \min_{r \in \mathbb{R}^n}$$

## Solution from Szegedy et al, “Intriguing properties of neural networks”

- Targetted Adversarial Attack Task:

$$\begin{aligned} \rho(x, x + r) &\rightarrow \min_{r \in \mathbb{R}^n} \\ \text{s.t. } y(x + r) &= \text{target\_class}, \end{aligned}$$

- Targetted Lagrange function  $L(r, c | x)$ :

$$\|r\|^2 - c \log p(y = \text{target\_class} | x + r) \rightarrow \min_{r \in \mathbb{R}^n}$$

- Non-targetted Adversarial Attack Task:

$$\begin{aligned} \rho(x, x + r) &\rightarrow \min_{r \in \mathbb{R}^n} \\ \text{s.t. } y(x + r) &= y(x), \end{aligned}$$

## Solution from Szegedy et al, “Intriguing properties of neural networks”

- Targetted Adversarial Attack Task:

$$\begin{aligned} \rho(x, x + r) &\rightarrow \min_{r \in \mathbb{R}^n} \\ \text{s.t. } y(x + r) &= \text{target\_class}, \end{aligned}$$

- Targetted Lagrange function  $L(r, c | x)$ :

$$\|r\|^2 - c \log p(y = \text{target\_class} | x + r) \rightarrow \min_{r \in \mathbb{R}^n}$$

- Non-targetted Adversarial Attack Task:

$$\begin{aligned} \rho(x, x + r) &\rightarrow \min_{r \in \mathbb{R}^n} \\ \text{s.t. } y(x + r) &= y(x), \end{aligned}$$

- Non-targetted Lagrange function  $L(r, c | x)$ :

$$\|r\|^2 + c \log p(y = y_{\text{origin}} | x + r) \rightarrow \min_{r \in \mathbb{R}^n}$$

## Solution from Szegedy et al, “Intriguing properties of neural networks”

- Targetted Adversarial Attack Task:

$$\begin{aligned} \rho(x, x + r) &\rightarrow \min_{r \in \mathbb{R}^n} \\ \text{s.t. } y(x + r) &= \text{target\_class}, \end{aligned}$$

- Targetted Lagrange function  $L(r, c | x)$ :

$$\|r\|^2 - c \log p(y = \text{target\_class} | x + r) \rightarrow \min_{r \in \mathbb{R}^n}$$

- Non-targetted Adversarial Attack Task:

$$\begin{aligned} \rho(x, x + r) &\rightarrow \min_{r \in \mathbb{R}^n} \\ \text{s.t. } y(x + r) &= y(x), \end{aligned}$$

- Non-targetted Lagrange function  $L(r, c | x)$ :

$$\|r\|^2 + c \log p(y = y_{\text{origin}} | x + r) \rightarrow \min_{r \in \mathbb{R}^n}$$



## Solution from Szegedy et al, “Intriguing properties of neural networks”

- Targetted Adversarial Attack Task:

$$\begin{aligned} \rho(x, x + r) &\rightarrow \min_{r \in \mathbb{R}^n} \\ \text{s.t. } y(x + r) &= \text{target\_class}, \end{aligned}$$

- Targetted Lagrange function  $L(r, c | x)$ :

$$\|r\|^2 - c \log p(y = \text{target\_class} | x + r) \rightarrow \min_{r \in \mathbb{R}^n}$$

- Non-targetted Adversarial Attack Task:

$$\begin{aligned} \rho(x, x + r) &\rightarrow \min_{r \in \mathbb{R}^n} \\ \text{s.t. } y(x + r) &= y(x), \end{aligned}$$

- Non-targetted Lagrange function  $L(r, c | x)$ :

$$\|r\|^2 + c \log p(y = y_{\text{origin}} | x + r) \rightarrow \min_{r \in \mathbb{R}^n}$$

### ! Method Problems

1. *Attack success or not* – there is no guarantee the method will work;
2. Simple optimizers may not work due to nonconvexity of Neural Networks (authors use L-BFGS);

## Solution from Szegedy et al, “Intriguing properties of neural networks”

- Targetted Adversarial Attack Task:

$$\begin{aligned} \rho(x, x + r) &\rightarrow \min_{r \in \mathbb{R}^n} \\ \text{s.t. } y(x + r) &= \text{target\_class}, \end{aligned}$$

- Targetted Lagrange function  $L(r, c | x)$ :

$$\|r\|^2 - c \log p(y = \text{target\_class} | x + r) \rightarrow \min_{r \in \mathbb{R}^n}$$

- Non-targetted Adversarial Attack Task:

$$\begin{aligned} \rho(x, x + r) &\rightarrow \min_{r \in \mathbb{R}^n} \\ \text{s.t. } y(x + r) &= y(x), \end{aligned}$$

- Non-targetted Lagrange function  $L(r, c | x)$ :

$$\|r\|^2 + c \log p(y = y_{\text{origin}} | x + r) \rightarrow \min_{r \in \mathbb{R}^n}$$

### ! Method Problems

1. *Attack success or not* – there is no guarantee the method will work;
2. Simple optimizers may not work due to nonconvexity of Neural Networks (authors use L-BFGS);

### i More sophisticated methods

- Fast Gradient Sign Method (FGSM)
- Deep Fool