COMP7180: Quantitative Methods for DAAI

(Credits from Prof. Andrew Ng)

(Credits from HKBU)

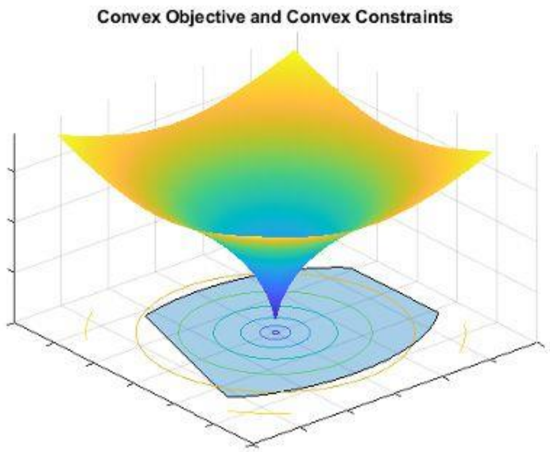
Course Instructors: Dr. Yang Liu and Dr. Bo Han

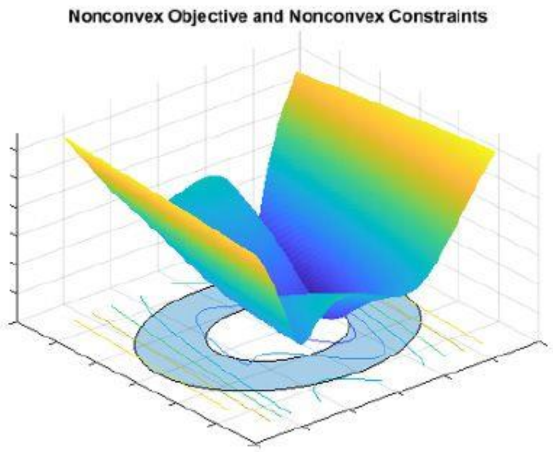
Teaching Assistant: Mr. Minghao Li

Course Contents

- Continuous and Discrete Random Variables (Week 7)
- Conditional Probability and Independence (Week 8)
- Maximum Likelihood Estimation (Week 9)
- Mathematical Optimization (Week 10)
- Convex and Non-Convex Optimization (Week 11) ← Our Focus
- Quiz and Course Review (Week 12)

Convex Optimization





General optimization problem

- very difficult to solve
- methods involve some compromise, e.g., very long computation time, or not always finding the solution (which may not matter in practice)

Exceptions: certain problem classes can be solved efficiently and reliably:

- linear optimization problems
- convex optimization problems

Credits from Boyd & Vandenberghe

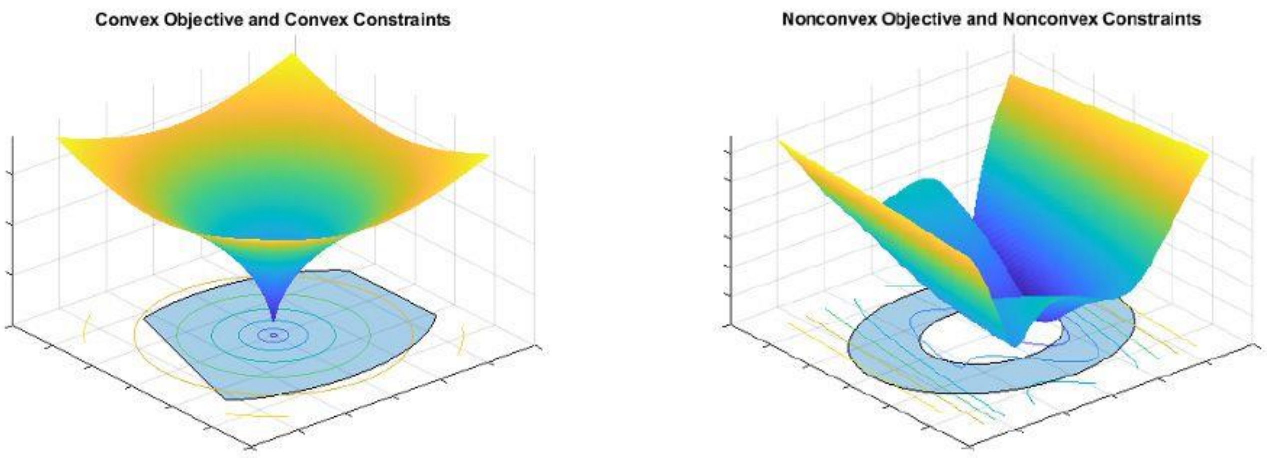
Convex Optimization

- Convex optimization problems are far more general than linear optimization (LO) problems

  Linear optimization is a special case of convex optimization (Page 28).

- Convex optimization share the desirable properties of LO problems:

  They can be solved quickly and reliably up to very large size -- hundreds of thousands of variables and constraints.



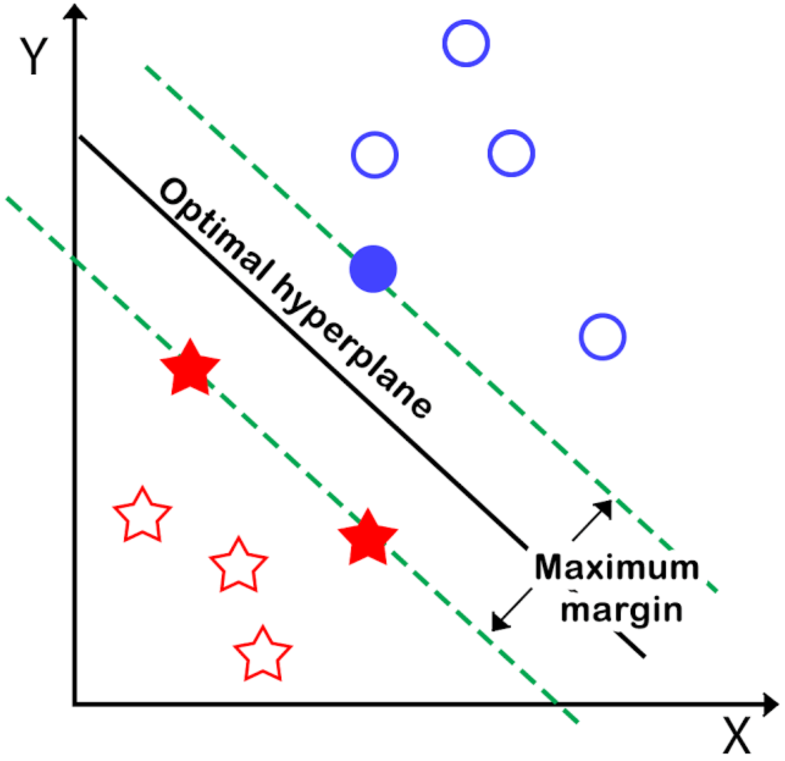
Convex Optimization

- Why Convex Optimization matters in Machine Learning?

Answer: Many machine learning problems can be summarized into Convex Optimation problems.

Examples:

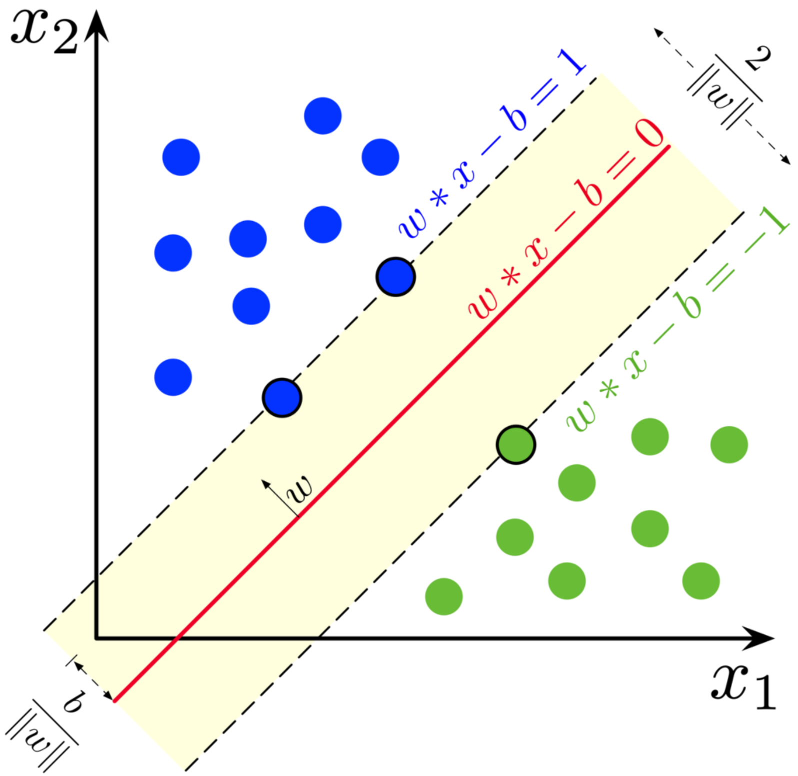
- Support vector machine (SVM)
- Kernel ridge regression (KRR)



Convex Optimization

- Support vector machine (SVM)

In machine learning, SVM are supervised learning models with associated learning algorithms that analyze data for classification and regression analysis.



More formally, a SVM constructs a hyperplane or set of hyperplanes in a high or infinite-dimensional space, which can be used for classification, regression, or other tasks like outliers detection.

Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the nearest training-data point of any class (so-called functional margin)

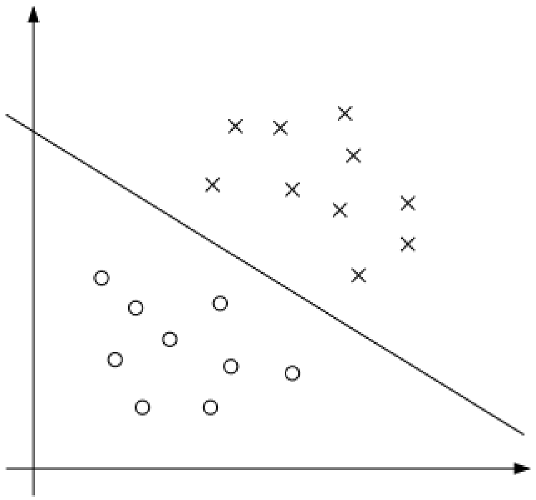
Credits from Wikipedia

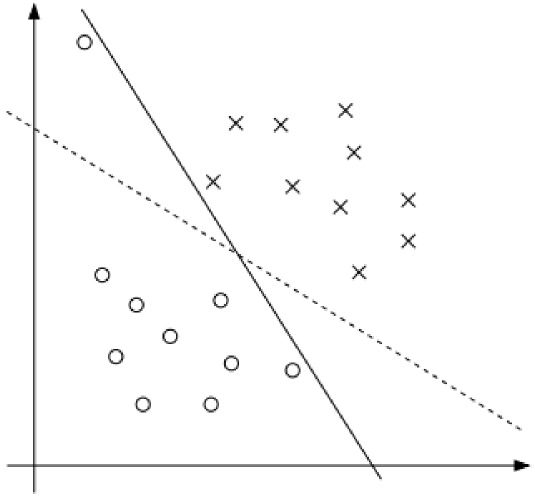
SVM: An Example of Convex Optimization

$$\min_{w,b} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n} \varepsilon_i$$

$$\text{s.t. } y^{(i)}\left(w^\top x^{(i)} + b\right) \geq 1 - \varepsilon_i,\ i = 1, \cdots, n$$

$$\varepsilon_i \geq 0,\ i = 1, \cdots, n$$





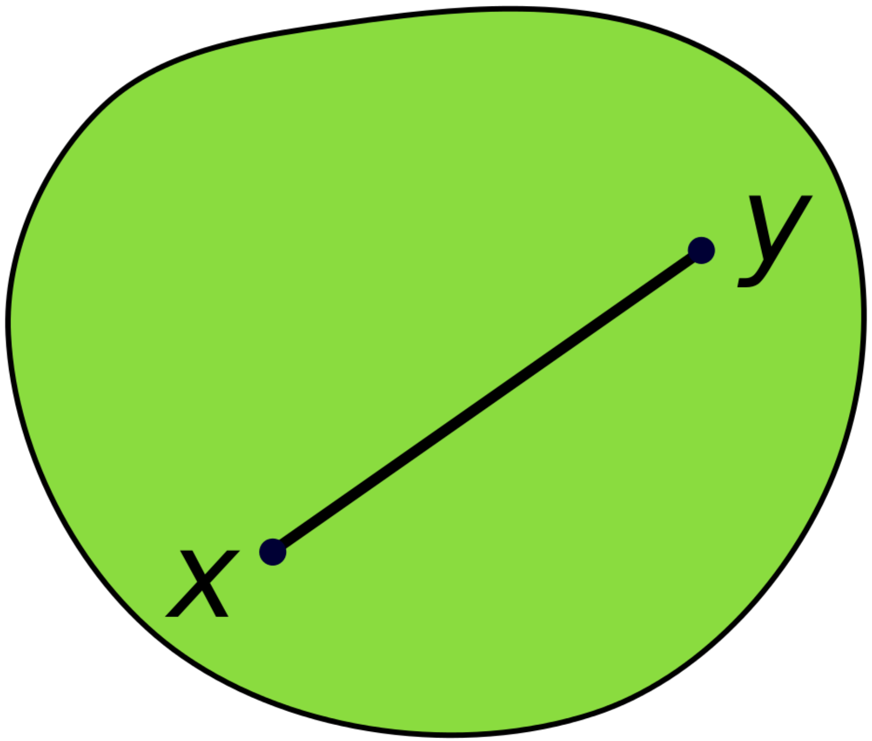
(Credits from CS229)

Convex Optimization

- What is convex optimization?

- Before introducing convex optimization, we need to introduce two concepts: convex set and convex function.

- ***Convex Set*** is a set that the line segment between any two points in the set lies in the set: C is a convex set, if for any two points x, y in C, then $tx + (1-t)y \in C$, where $0 \leq t \leq 1$.



Credits from Wikipedia

Convex Optimization

Example of Convex Set: the solution set of linear equation $A\mathbf{x}=b$ is a convex set.

Proof. Let x and y be the solutions of Ax=b, then we need to show that tx+(1-t)y is also a solution:

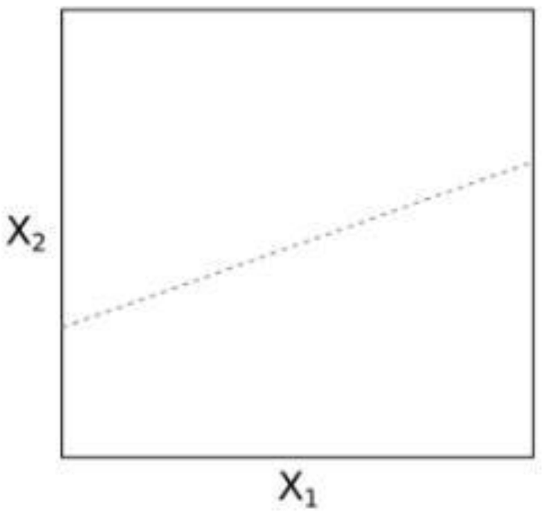
$$A(t\mathbf{x}+(1-t)\mathbf{y}) = tA\mathbf{x}+(1-t)A\mathbf{y}$$
$$= tb+(1-t)b = b$$

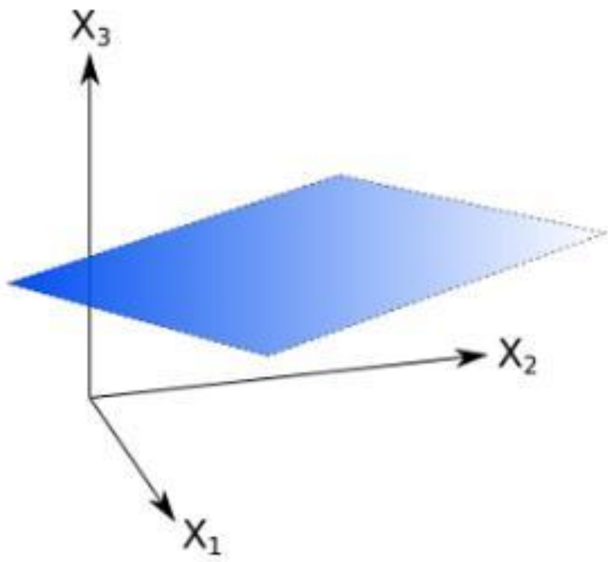
Therefore, the solution set of linear equation is a convex set.

Note:

In this class x and y represent real values in R

x represents a vector $\begin{pmatrix} x_1 \\ \vdots \\ x_d \end{pmatrix}$

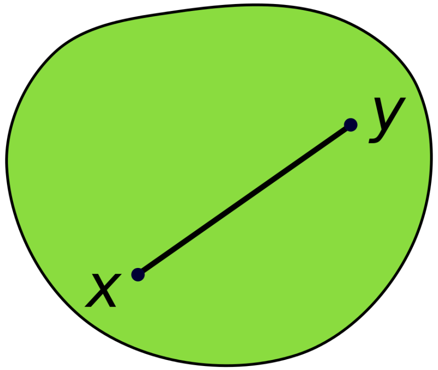


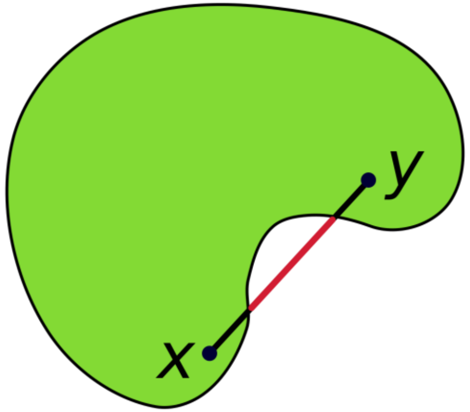


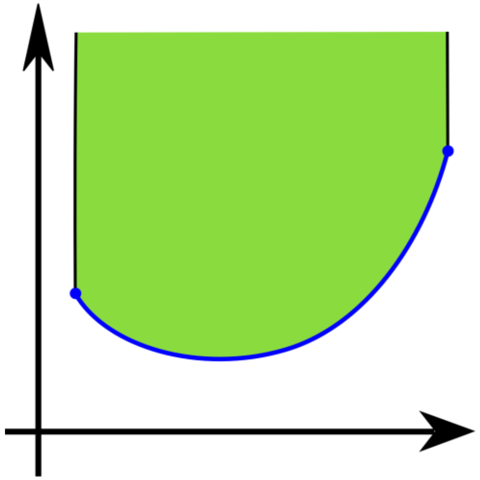
Credits from Dr. Yang Liu

Convex Optimization: Exercises

- Please answer whether the green area is a convex set or not.

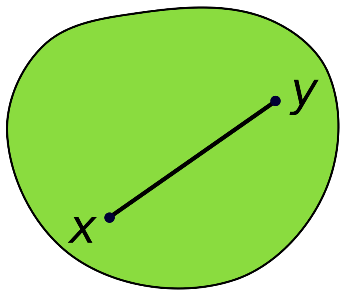




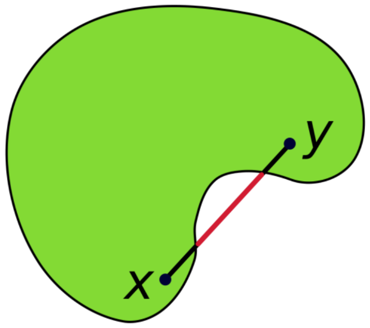


Credits from Wikipedia

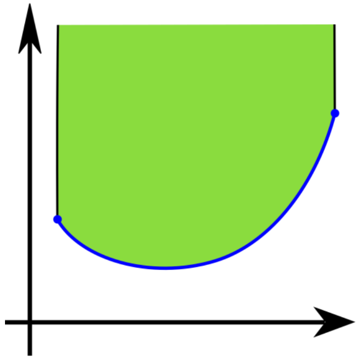
Convex Optimization: Exercises



- It is a convex set, because the line segment between any two points in the set lies in the set



- It is not a convex set, because the line segment between x and y in the set does not lie in the set



- It is a convex set, because the line segment between any two points in the set lies in the set

Credits from Wikipedia

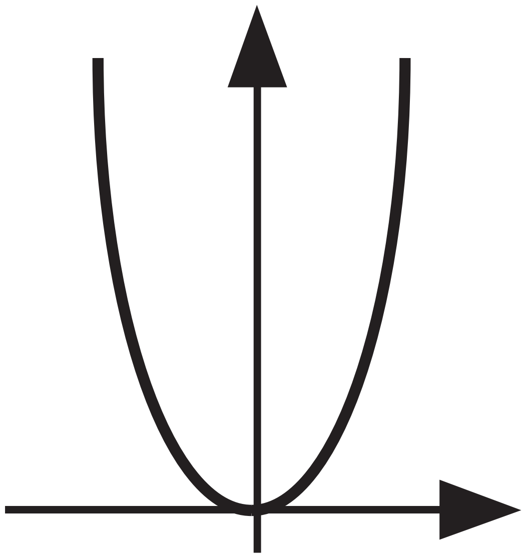
Convex Optimization

- Convex function: a function f is a convex function, if f meets the following two conditions:

1. the domain of f is a convex set C, that is f: C $\rightarrow$R

2. for any points **x**, **y**$\in$C and $0 \leq t \leq 1$,

$$f(t\mathbf{x}+(1-t)\mathbf{y}) \leq tf(\mathbf{x})+(1-t)f(\mathbf{y})$$



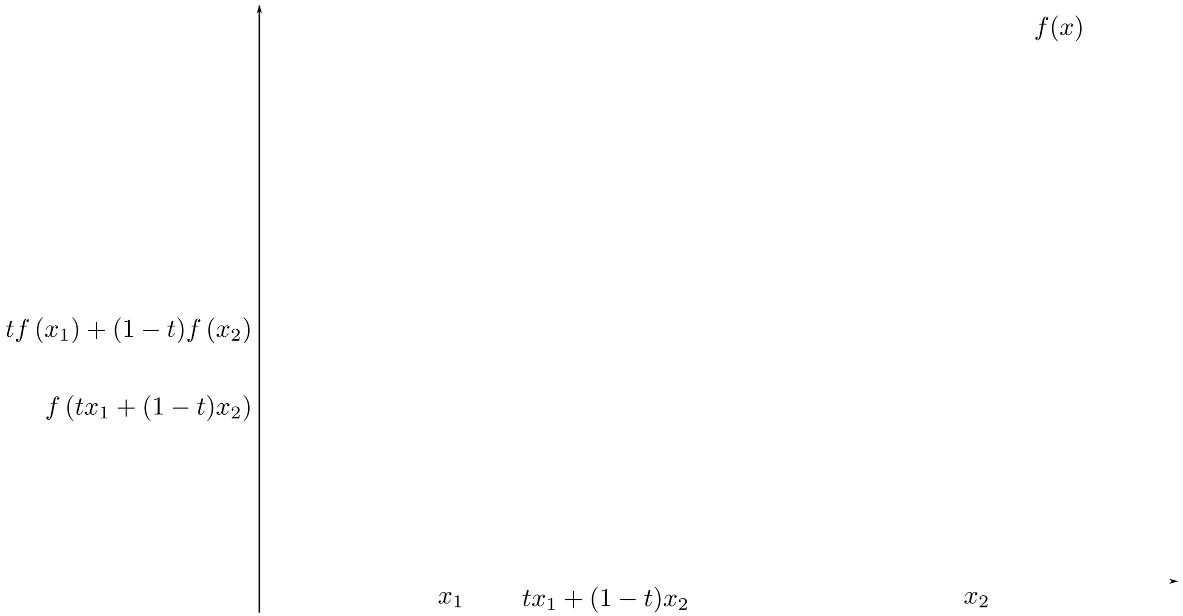
convex function

The graph of a convex function is similar to a bowl.

Convex Optimization

What is meaning of the following inequality?

$$f(t\mathbf{x}+(1-t)\mathbf{y}) \leq tf(\mathbf{x})+ (1-t)f(\mathbf{y})$$



The line segment between any two points ($\mathbf{x}$, f($\mathbf{x}$)) and ($\mathbf{y}$, f($\mathbf{y}$)) is above the convex function. Please see the purple line and black curve.

Convex Optimization: Examples

- Linear functions are convex fucntions:

Linear function is f(**x**) = **Ax+b**, where **A** is a matrix and the domain is R^{d}

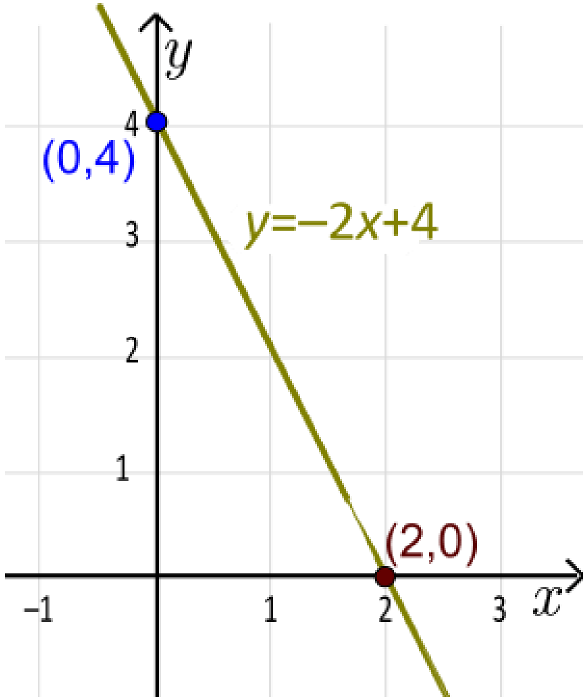
Firstly, the domain is a convex set, because R^{d} is a convex set.

Secondly, for any $0 \leq t \leq 1$, and any **x**, **y**, then

f(t**x**+(1-t)**y**)= **A**(t**x**+(1-t)**y**)+**b** = **A**(t**x**+(1-t)**y**)+t**b**+(1-t)**b**
=t**Ax**+t**b**+(1-t)**Ay**+(1-t)**b** = tf(**x**)+(1-t)f(**y**)

Therefore, f(t**x**+(1-t)**y**) = tf(**x**)+(1-t)f(**y**).

So f(**x**) is a convex function.



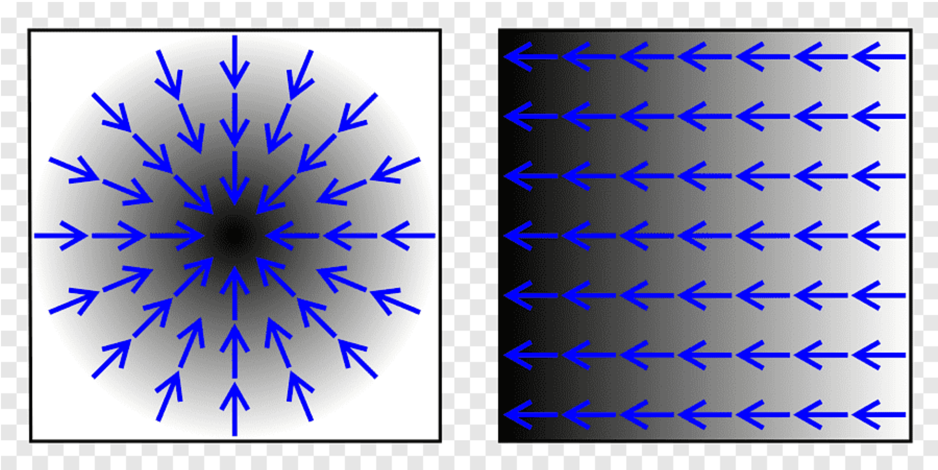
Convex Optimization

When f(**x**) is differential, can we check whether f(**x**) is convex by the gradient?

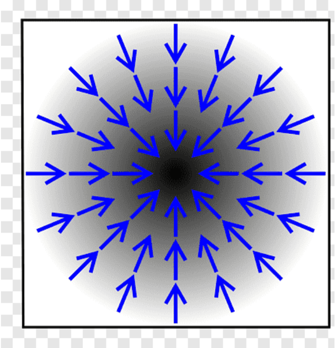
Answer: Yes

Recall what is gradient? Let **x** = $(x_1, x_2, \ldots, x_d)^T$, then the gredient of f(**x**) is

$$\nabla f(\mathbf{x}) = \begin{bmatrix} \frac{\partial f(\mathbf{x})}{\partial x_1} \\ \vdots \\ \vdots \\ \frac{\partial f(\mathbf{x})}{\partial x_d} \end{bmatrix}$$



Convex Optimization



What is meaning of gradient?

The gradient can be interpreted as the direction and rate of fastest increase.

Example: $f(x_1, x_2) = -({x_1}^2 + {x_2}^2)$. Then,

$$\nabla f(x_1, x_2) = \begin{bmatrix} -2x_1 \\ -2x_2 \end{bmatrix}$$

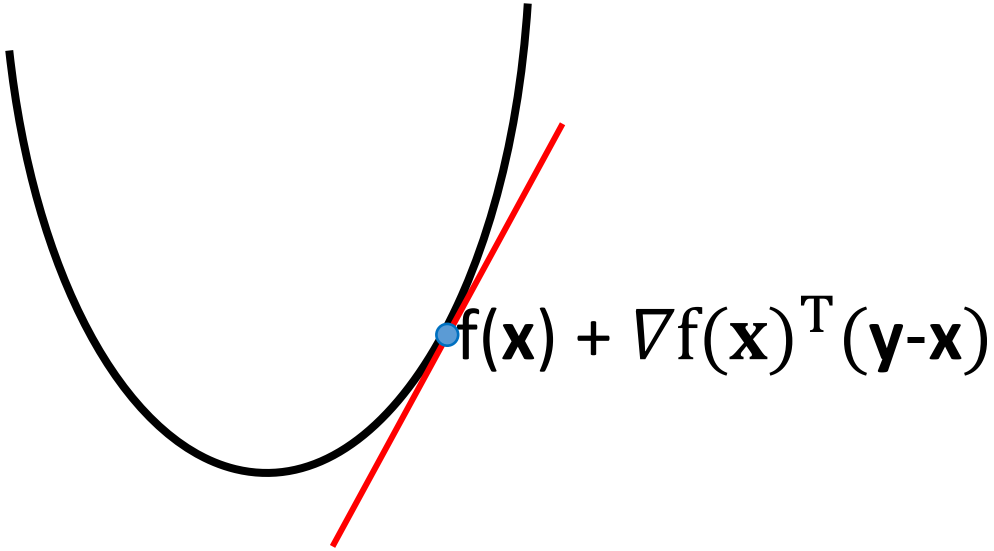
Convex Optimization

When f($\mathbf{x}$) is differential, can we check whether f($\mathbf{x}$) is convex by the gradient?
Following theorem gives the answer:

Assume that f($\mathbf{x}$) is differential, then f($\mathbf{x}$) is convex

if and only if

the domain C is convex and $f(\mathbf{y}) \geq f(\mathbf{x}) + \nabla f(\mathbf{x})^T(\mathbf{y}-\mathbf{x})$.



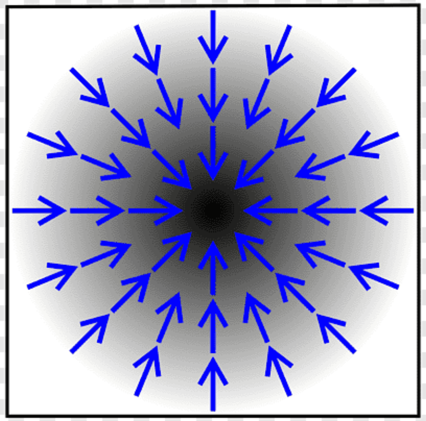
Convex Optimization

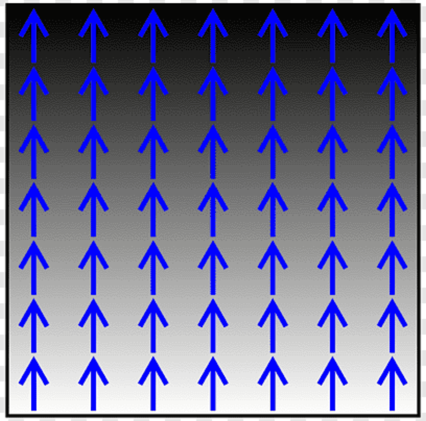
Theorem 1. Assume that f(**x**) is differentiable, then f(**x**) is convex

if and only if

the domain C is convex and $f(\mathbf{y}) \geq f(\mathbf{x}) + \nabla f(\mathbf{x})^T(\mathbf{y}-\mathbf{x})$.

The proof can be found in Proposition 4 in
https://wiki.math.ntnu.no/_media/tma4180/2016v/note2.pdf





We still use the linear function to check the result.

$f(\mathbf{x}) = \mathbf{Ax+b}$, then $\nabla f(\mathbf{x})^T = \mathbf{A}$.

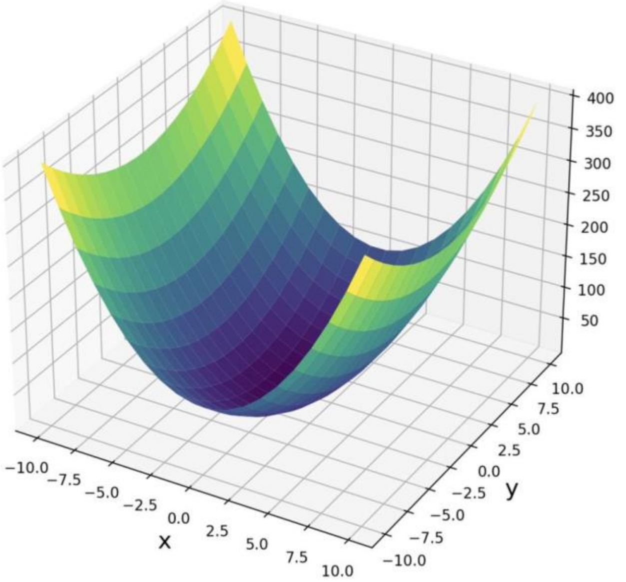
$f(\mathbf{x}) + \nabla f(\mathbf{x})^T(\mathbf{y}-\mathbf{x}) = \mathbf{Ax+b+A(y-x)} = \mathbf{Ay+b}$

So $f(\mathbf{y}) = f(\mathbf{x}) + \nabla f(\mathbf{x})^T(\mathbf{y}-\mathbf{x})$

Convex Optimization

When f(**x**) is twice differentiable, can we check whether f(**x**) is convex by Hessian Matrix? Answer: Yes

Recall what is Hessian Matrix? Let **x** = $(x_1, x_2, \ldots, x_d)^T$, then the Hessian Matrix f(**x**) is the Hessian Matrix is a dxd matrix, for the ij-th element in the matrix is the second-order partial derivatives of f: $\frac{\partial^2 f(\mathbf{x})}{\partial x_i \partial x_j}$



$$\mathbf{H}(\mathbf{x}) = \begin{bmatrix} \frac{\partial^2 f(\mathbf{x})}{\partial x_1 \partial x_1} & \cdots & \frac{\partial^2 f(\mathbf{x})}{\partial x_1 \partial x_d} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f(\mathbf{x})}{\partial x_d \partial x_1} & \cdots & \frac{\partial^2 f(\mathbf{x})}{\partial x_d \partial x_d} \end{bmatrix}$$

Convex Optimization

- If the second-order partial derivatives are continuous functions, then the Hessian Matrix is symmetric, i.e., $\mathbf{H}(\mathbf{x}) = \mathbf{H}(\mathbf{x})^{\mathrm{T}}$

Example: $f(x_1, x_2) = x_1^2 + x_2^2$

$$\frac{\partial^2 f(\mathbf{x})}{\partial x_1 \partial x_1} = 2, \quad \frac{\partial^2 f(\mathbf{x})}{\partial x_2 \partial x_2} = 2, \quad \frac{\partial^2 f(\mathbf{x})}{\partial x_1 \partial x_2} = 0, \quad \frac{\partial^2 f(\mathbf{x})}{\partial x_2 \partial x_1} = 0$$

So

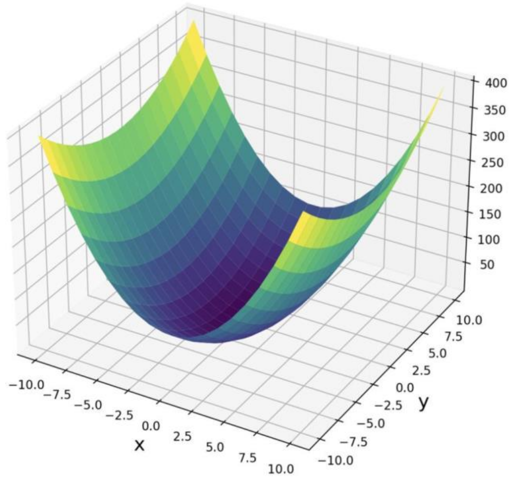
$$\mathbf{H}(\mathbf{x}) = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$

Convex Optimization: Exercises

Please Compute:

- the Hessian Matrix of $f(x_1, x_2) = x_1^{2} + 2x_1x_2 + x_2^{2}$

- the Hessian Matrix of $f(x_1, x_2) = 2x_1^{3} + 6x_1x_2 + x_2^{2} + x_2^{3}$



Convex Optimization: Exercises

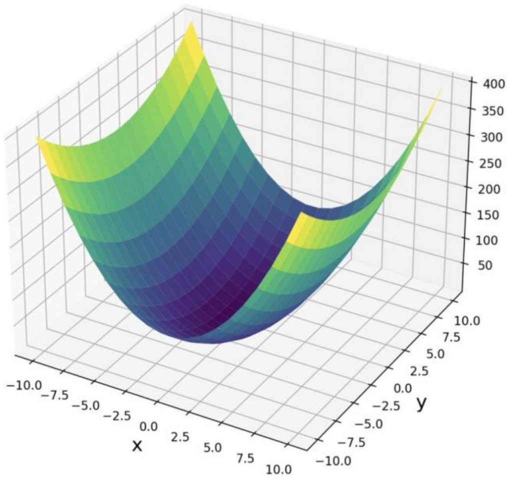
Solution:

- the Hessian Matrix of $f(x_1, x_2) = x_1^2 + 2x_1x_2 + x_2^2$

$$\frac{\partial^2 f(\mathbf{x})}{\partial x_1 \partial x_1} = 2, \frac{\partial^2 f(\mathbf{x})}{\partial x_2 \partial x_2} = 2, \frac{\partial^2 f(\mathbf{x})}{\partial x_1 \partial x_2} = 2, \frac{\partial^2 f(\mathbf{x})}{\partial x_2 \partial x_1} = 2$$

So

$$\mathbf{H}(\mathbf{x}) = \begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix}$$

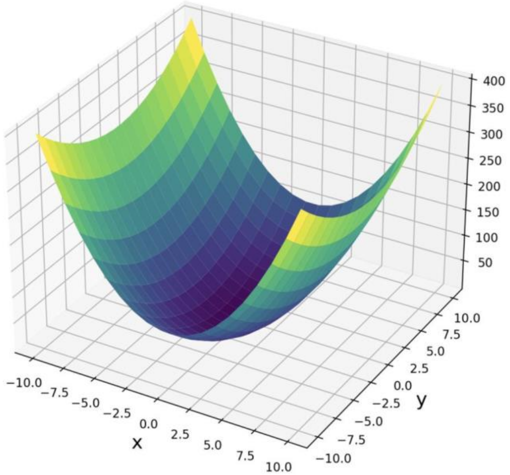


Convex Optimization: Exercises

Solution:

the Hessian Matrix of $f(x_1, x_2) = 2x_1^{\,3} + 6x_1x_2 + x_2^{\,2} + x_2^{\,3}$

$$\frac{\partial^2 f(\mathbf{x})}{\partial x_1 \partial x_1} = 12x_1, \quad \frac{\partial^2 f(\mathbf{x})}{\partial x_2 \partial x_2} = 2+6x_2, \quad \frac{\partial^2 f(\mathbf{x})}{\partial x_1 \partial x_2} = 6, \quad \frac{\partial^2 f(\mathbf{x})}{\partial x_2 \partial x_1} = 6$$



So

$$\mathbf{H}(\mathbf{x}) = \begin{bmatrix} 12x_1 & 6 \\ 6 & 2 + 6x_2 \end{bmatrix}$$

Convex Optimization

Theorem 2. Assume that f($\mathbf{x}$) is twice differential, then f($\mathbf{x}$) is convex

if and only if

the domain C is convex and the Hessian Matrix $\mathbf{H}(\mathbf{x})$ is positive semi-definite.

The proof can be found in Proposition 7 in https://wiki.math.ntnu.no/_media/tma4180/2016v/note2.pd

- What is positive semi-definite matrix **M**?

Positive semi-definite matrix **M** is a nxn sysmetric matrix **M** = $\mathbf{M}^{\mathrm{T}}$ and for any real n-dimensional vector **z,** $\mathbf{z}^{\mathrm{T}}\mathbf{M}\mathbf{z} \geq 0$.

Convex Optimization

Examples of positive semi-definite matrix:

- **M** = $\begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix}$ is a positive semi-definite matrix, if a$\geq$0 and b$\geq$0

Because for any (x,y), (x,y)$\mathbf{M}(x, y)^{T}$ = ax^2+by^2 $\geq$0

- Please check that **M** = $\begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{pmatrix}$ is a positive semi-definite matrix

Convex Optimization

- Please check that **M** = $\begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{pmatrix}$ is a positive semi-definite matrix

For any (x,y,z), we can check that

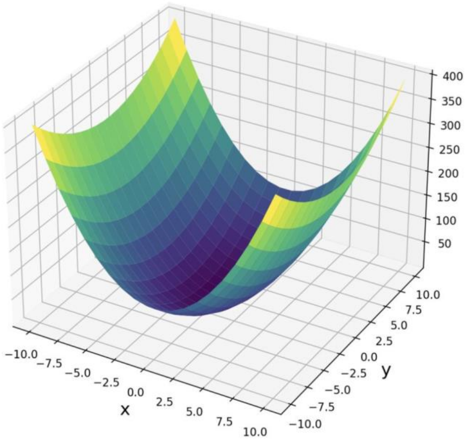
$$(x,y,z)\begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{pmatrix}(x, y, z)^T = 2x^2-2xy+2y^2-2yz+2z^2$$

$$= x^2 + (x-y)^2 + (z-y)^2 + z^2 \geq 0$$

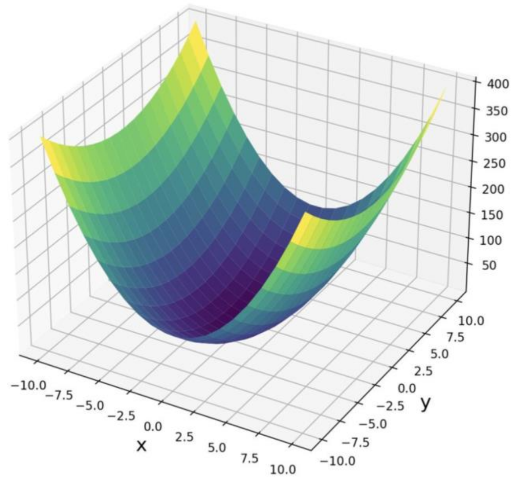
Convex Optimization

Theorem 3. f(**x**) = $\mathbf{x}^T$**Mx**+**bx** is a convex function, if **M** is a positive semi-definite matrix.

***Proof*.** The Hessian Matrix of f(**x**) is 2**M**
(The details about how to compute the matrix derivatives can be found in *https://cloud.tencent.com/developer/article/1551901*).
Because **M** is a positive semi-definite matrix, 2**M** is positive semi-definite



Convex Optimization



Definition of Convex Optimization Problem:

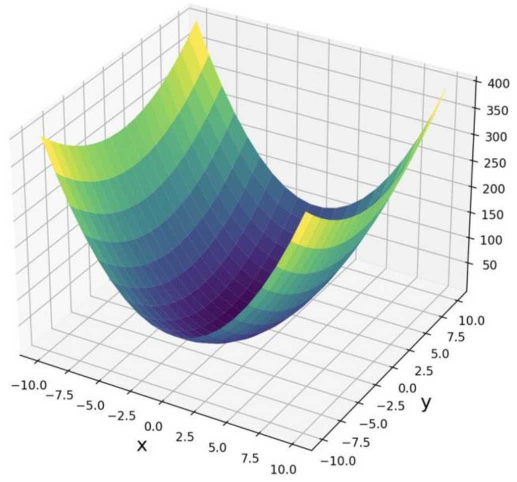
Minimize $f(\mathbf{x})$

Subject to $g_i(\mathbf{x}) \leq 0$, $i=1,\ldots,m$,

$h_j(\mathbf{x}) = 0$, $j=1,\ldots,n$.

- $g_i(\mathbf{x})$ is convex function, $i=1,\ldots,m$
- $h_j(\mathbf{x})$ is linear function $\mathbf{A}_j\mathbf{x}+\mathbf{b}_j$, $j=1,\ldots,n$
- $f(\mathbf{x})$ is a convex function

Convex Optimization



Examples of Convex Optimization:

- Linear optimization belongs to Convex Optimization. Because linear functions are also convex function.

- Minimize f(**x**) = $\mathbf{x}^{T}\mathbf{M}\mathbf{x}+\mathbf{b}\mathbf{x}$ is a Convex Optimization problem without constraints, if **M** is a positive semi-definite matrix.

Convex Optimization without Constraints

We first introduce how to address convex optimization without constraints: that is

Minimize f($\mathbf{x}$), where f($\mathbf{x}$) is convex

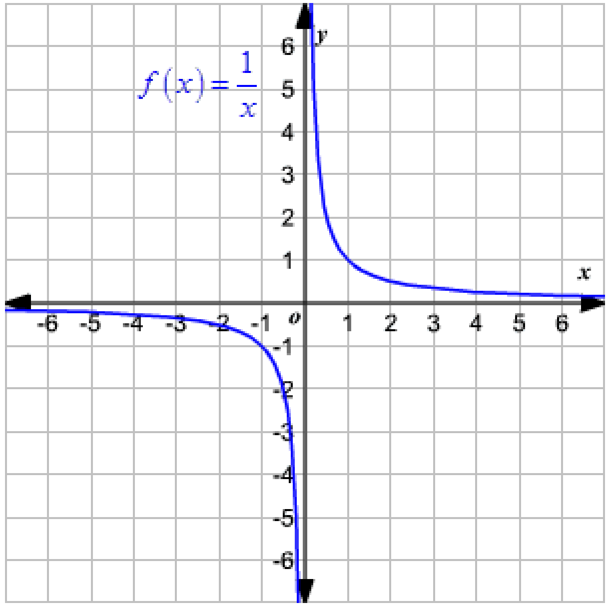
We want to ask some issues:

- Issue 1. Whether we can find a solution to this issue?

- Issue 2. Whether the solution is unique.

Convex Optimization without Constraints

- Issue 1. Whether we can find a solution to convex optimization without constraints? (解的存在性)



Not all Minimize f(**x**) has a solution such as

f(x) = 1/x, where the domain is (0,+∞)

Because f(x) gets close to 0, when x gets close to +∞. So the optimal solution should be +∞, but we cannot obtain it.

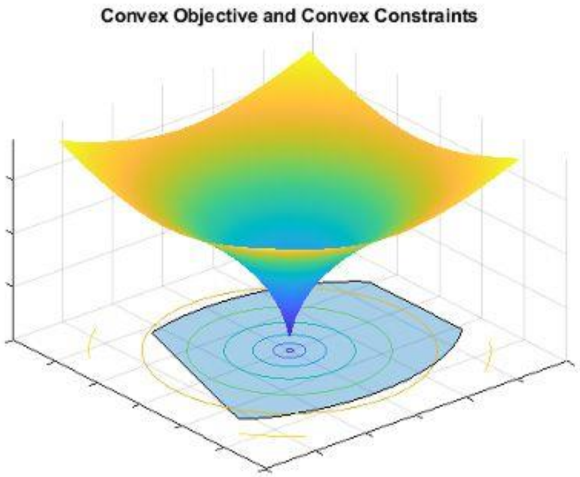
Convex Optimization without Constraints

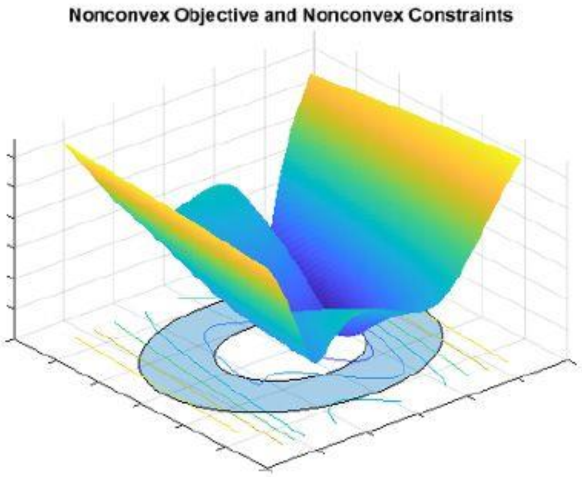
Theorem 4. Assume that f($\mathbf{x}$) is differential, then $\mathbf{x}_0$ is the optimal solution of Convex optimization problem Minimize f($\mathbf{x}$) if and only if $\nabla \mathrm{f}(\mathbf{x}_0)$ = $\mathbf{0}$.

The proof can be found in Section 4.2.3 in *https://web.stanford.edu/~boyd/cvxbook/bv_cvxbook.pdf*

This theorem implies an important fact that:

if we want to address Minimize f($\mathbf{x}$), we only need to check its gradient.





Convex Optimization without Constraints

Theorem 4. Assume that f(**x**) is differential and the domain is R^{d}, then $\mathbf{x}_0$ is the optimal solution of Convex optimization problem Minimize f(**x**)

if and only if $\nabla \mathrm{f}(\mathbf{x}_0) = \mathbf{0}$.

The proof can be found in Section 4.2.3 in https://web.stanford.edu/~boyd/cvxbook/bv_cvxbook.pdf

Application of Theorem 4:

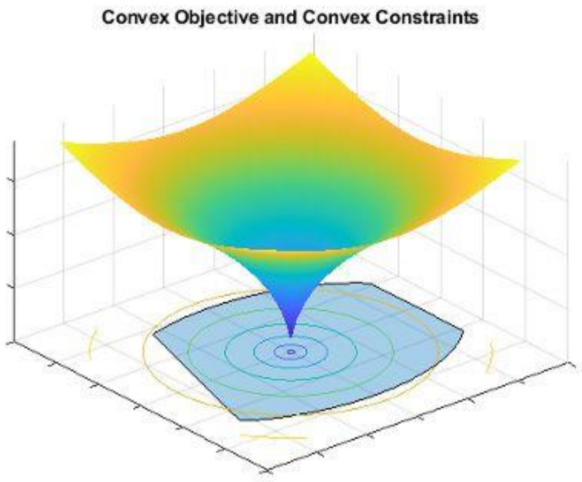
$\mathrm{f}(x,y) = ax^2+by^2$, where a>0, b>0. Then

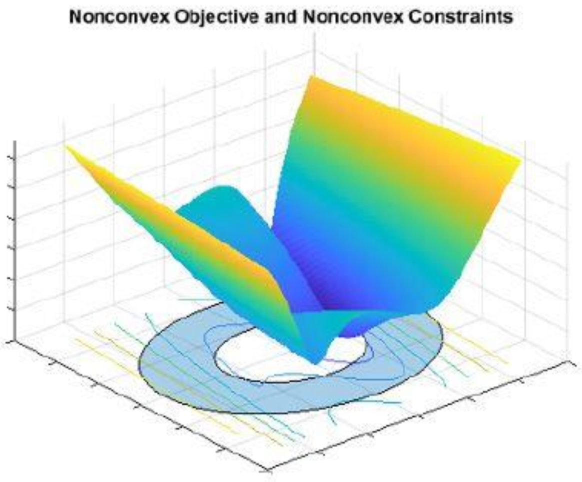
$$\nabla \mathrm{f}(\mathbf{x}_0) = \begin{bmatrix} 2a\mathrm{x}_0 \\ 2\mathrm{b}\mathrm{y}_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \text{where } \mathbf{x}_0 = \begin{bmatrix} \mathrm{x}_0 \\ \mathrm{y}_0 \end{bmatrix}, \text{so the optimal solution is } \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

Convex Optimization without Constraints

Exercise: $\mathbf{M} = \begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{pmatrix}, \mathbf{b} = (1,0,0)$

what is the solution of Minimize f($\mathbf{x}$) $= \mathbf{x}^T \mathbf{M}\mathbf{x} + \mathbf{b}\mathbf{x}$**?**



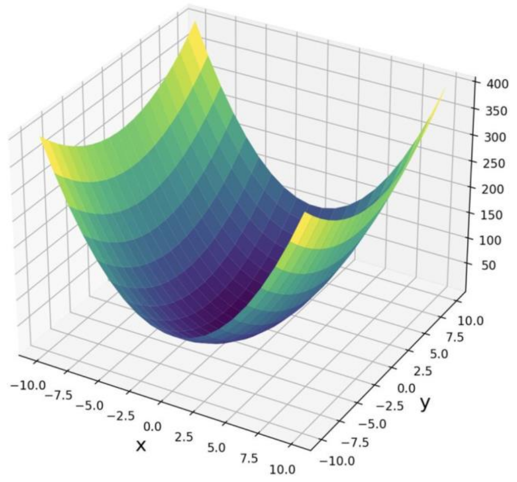


Convex Optimization without Constraints

Solution: Firstly, we note that **M** is an inverse matrix, and the inverse is

$$\mathbf{M}^{-1} = \begin{pmatrix} 0.75 & 0.5 & 0.25 \\ 0.5 & 1 & 0.5 \\ 0.25 & 0.5 & 0.75 \end{pmatrix}$$

Secondly, we need to compute the gradient $\nabla f(\mathbf{x})$



How to compute the gradient?

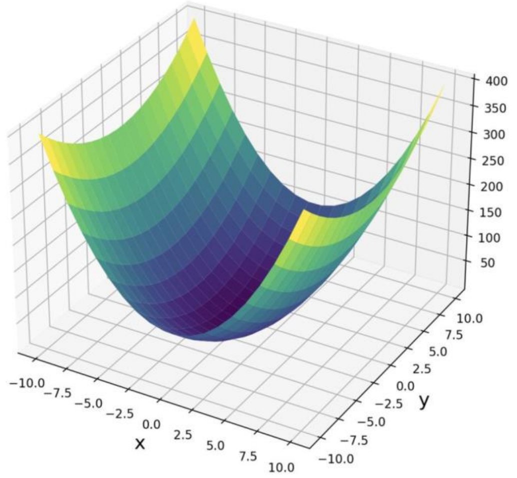
Convex Optimization without Constraints

- How to compute the gredient?

We introduce the Matrix derivatives:

$$\frac{\partial \mathbf{x}^{\mathrm{T}}\mathbf{M}\mathbf{x}}{\partial \mathbf{x}} = (\mathbf{M} + \mathbf{M}^{\mathrm{T}})\mathbf{x},$$

$$\frac{\partial \mathbf{b}\mathbf{x}}{\partial \mathbf{x}} = \mathbf{b}^{\mathrm{T}},$$

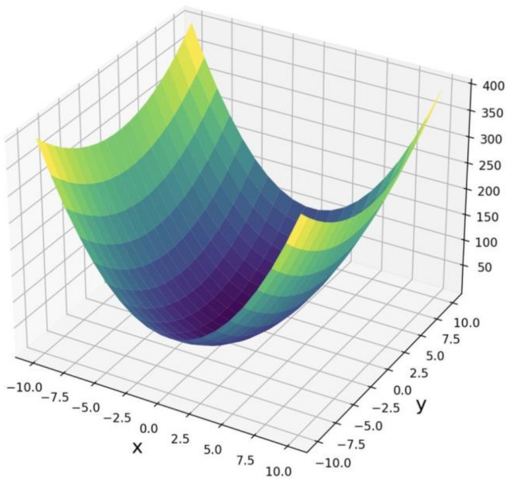


More details can be found in https://cloud.tencent.com/developer/article/1551901

Convex Optimization without Constraints

Because **M** is sysmetric, then

$$\frac{\partial \mathbf{x}^{\mathrm{T}}\mathbf{M}\mathbf{x}}{\partial \mathbf{x}} = (\mathbf{M} + \mathbf{M}^{\mathrm{T}})\mathbf{x} \; = \; \mathbf{2Mx}.$$



So we obtain that

$$\nabla \mathrm{f}(\mathbf{x}) = \mathbf{2Mx} + \mathbf{b}^{\mathrm{T}}$$

Then according to Theorem 4, the optimal solution sholud satisfy that

$\mathbf{2Mx_0} + \mathbf{b}^{\mathrm{T}} = \mathbf{0}$

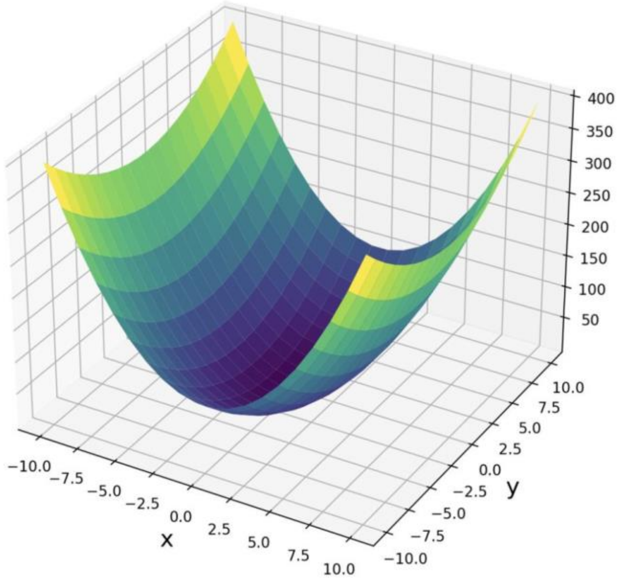
Convex Optimization without Constraints

Then according to Theorem 4, the optimal solution sholud satisfy that

$\mathbf{2Mx_0}+\mathbf{b}^{\mathrm{T}} = \mathbf{0}$

The solution of $\mathbf{2Mx_0}+\mathbf{b}^{\mathrm{T}} = \mathbf{0}$ is $\frac{-\mathbf{M}^{-1}\mathbf{b}^{\mathrm{T}}}{2} = \frac{-1}{2}\begin{pmatrix} 0.75 & 0.5 & 0.25 \\ 0.5 & 1 & 0.5 \\ 0.25 & 0.5 & 0.75 \end{pmatrix}\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$

So the optimal solution is $\begin{pmatrix} -3/8 \\ -1/4 \\ -1/8 \end{pmatrix}$



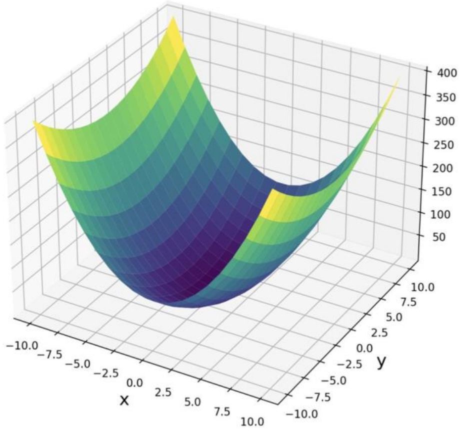
Convex Optimization without Constraints

- Issue 2. Whether the solution is unique.

Theorem 5. Considering Minimize f($\mathbf{x}$), if f($\mathbf{x}$) is **strictly convex**, the optimal solution is unique.

What is strictly convex?

$$f(t\mathbf{x}+(1-t)\mathbf{y}) < tf(\mathbf{x}) + (1-t)f(\mathbf{y})$$



- x^2 is strictly convex. Why? we will show it in page 40
- Linear function is not strictly convex but convex. Why? Because

$$f(t\mathbf{x}+(1-t)\mathbf{y}) = tf(\mathbf{x}) + (1-t)f(\mathbf{y})$$

Convex Optimization without Constraints

The methods to check whether a function is strictly convex.

Theorem 6. Assume that f(**x**) is differential, then f(**x**) is strictly convex if and only if the domain C is convex and f(**y**) > f(**x**) + $\nabla f(\mathbf{x})^{T}(\mathbf{y}-\mathbf{x})$.

See Section 3.1 in *https://www.princeton.edu/~aaa/Public/Teaching/ORF523/S16/ORF523_S16_Lec7_gh.pdf*

Theorem 7. Assume that f(**x**) is twice differential, then if the domain C is convex and the Hessian matrix of f(**x**) is positive definite, then f(**x**) is strictly convex.

See Section 3.1 in *https://www.princeton.edu/~aaa/Public/Teaching/ORF523/S16/ORF523_S16_Lec7_gh.pdf*

What is positive definite?

Convex Optimization without Constraints

What is a positive definite matrix?

Positive definite matrix **M** is a nxn symmetric matrix **M** = $\mathbf{M}^{\mathrm{T}}$ and for any real n-dimensional vector **z** (**z**$\neq \mathbf{0}$), $\mathbf{z}^{\mathrm{T}}\mathbf{M}\,\mathbf{z} > 0$.

Compare to Positive semi-definite:

- Positive semi-definite: $\mathbf{z}^{\mathrm{T}}\mathbf{M}\mathbf{z} \geq 0$.
- Positive definite: $\mathbf{z}^{\mathrm{T}}\mathbf{M}\mathbf{z} > 0$, if **z**$\neq \mathbf{0}$.

M = $\begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{pmatrix}$ is *positive definite* ,

because (x,y,z) $\begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{pmatrix} (x, y, z)^{\mathrm{T}}$

= $x^2 + (x-y)^2 + (z-y)^2 + z^2 > 0$

if (x,y,z) is not **0.**

Convex Optimization without Constraints

How to check whether a matrix is a positive definite matrix?

- First, the matrix should be symmetric **M** = $\mathbf{M}^{\mathrm{T}}$.
- Second, the eigenvalues are larger than 0.

For example,

M = $\begin{pmatrix} 5 & 1 \\ 1 & 5 \end{pmatrix}$ is symmeric and the eigenvalues are 4 and 6.

So **M** is a positive definite matrix

Convex Optimization without Constraints

According to Theorems 5 and 7, we know that f($\mathbf{x}$) =$\mathbf{x}^{\mathrm{T}}\mathbf{M}\mathbf{x}+\mathbf{b}\mathbf{x}$ has anunique solution, if **M** is positive definite.

Proof. The Hessian Matrix of f is 2**M** (The details about how to compute the matrix derivatives can be found in *https://cloud.tencent.com/developer/article/1551901*).

So if **M** is positive definite, then 2**M** is positive definite.

By Theorem 7, f($\mathbf{x}$) is a strictly convex function.

By Theorem 5, f($\mathbf{x}$) =$\mathbf{x}^{\mathrm{T}}\mathbf{M}\mathbf{x}+\mathbf{b}\mathbf{x}$ has an unique solution

Convex Optimization with Constraints

We next introduce convex optimization with constraints: that is

Minimize $f(\mathbf{x})$,

Subject to $g_i(\mathbf{x}) \leq 0$, $i=1,\ldots,m$,

$h_j(\mathbf{x}) = 0$, $j=1,\ldots,n$.

We want to ask an issue:

- Issue. Whether we can find a solution to this issue? (解的存在性)

Convex Optimization with Constraints

- Issue. Whether we can find a solution to this issue? (解的存在性)

Following theorem gives the answer:

Theorem 8. Assume that f(**x**) is differential, then $\mathbf{x}_0$ is the optimal solution of the Convex optimization problem with constraints if and only if

$$\nabla f(\mathbf{x}_0)^T(\mathbf{y}-\mathbf{x}_0) \geq 0,$$

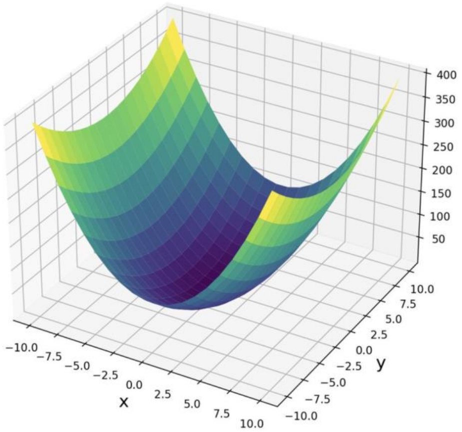
for all **y** satisfy the constraints.

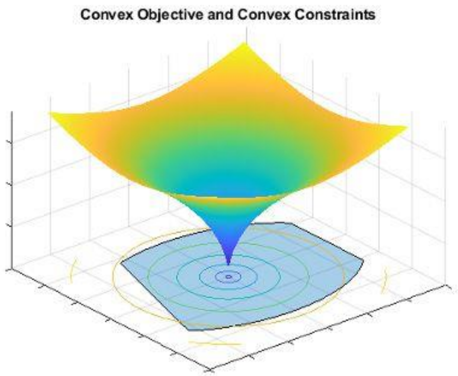
The proof can be found in Section 4.2.3 in https://web.stanford.edu/~boyd/cvxbook/bv_cvxbook.pdf

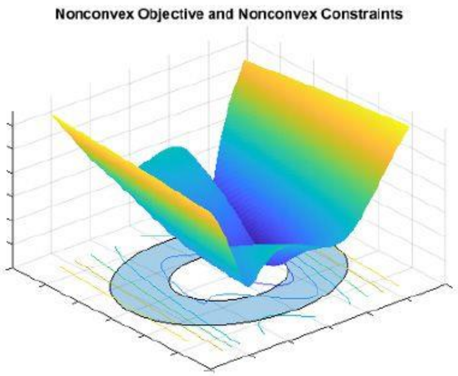
Convex Optimization with Constraints

- Due to the complexity of constraints, it is complex and difficult to introduce the common solutions of convex optimization with constraints.

- In this class, we only address a popular examples to help us understand Theorem 8. If you are interested in deeper theory with respect to Convex Optimization with Constraints, you can read Chapter 4 and Chapter 5 in the book https://web.stanford.edu/~boyd/cvxbook/bv_cvxbook.pdf







Convex Optimization with Constraints

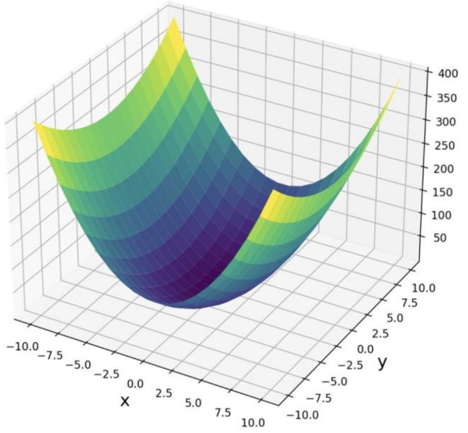
- We consider a conic form problem

Minimize **bx**

Subject to $\mathbf{x}^{\mathrm{T}}\mathbf{M}\mathbf{x} - \mathrm{c} \leq 0,$

where

$$\mathbf{M} = \begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{pmatrix}, \mathbf{b} = (1,0,0)\,, \mathrm{c} = 1$$



Convex Optimization with Constraints

- We firstly transform this problem to a simple form

Note that **M** is positive definite. According to the property of positive definite matrix, **M** can be decomposed as folloing form:

$$\mathbf{M} = \mathrm{U}^{\mathrm{T}}\mathrm{D}\mathrm{U},$$

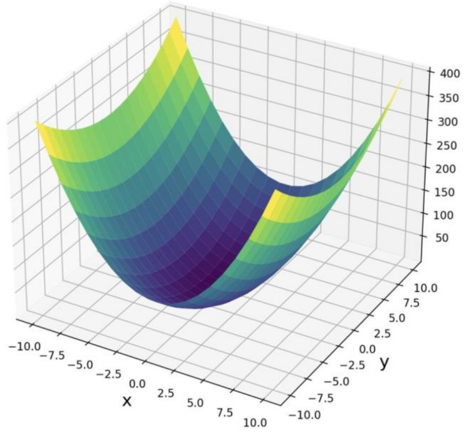
where U is the orthogonal matrix and D is the diagonal matrix whose elements in the diagonal elements are M's eigenvalues.

This decomposition is called singular value decomposition, see *https://en.wikipedia.org/wiki/Singular_value_decomposition*

Convex Optimization with Constraints

- D can be written as follows:

$$\begin{pmatrix} a_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & a_d \end{pmatrix},$$



where a_i >0 (i=1,...,d), because **M** is positive definite.

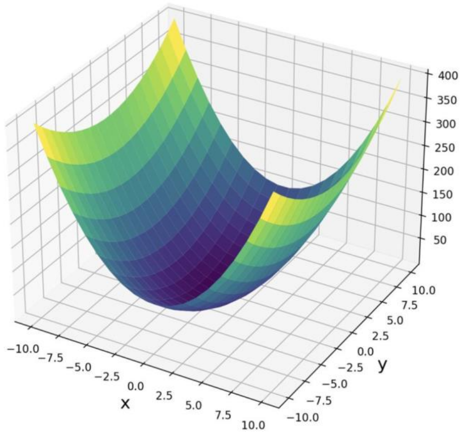
Convex Optimization with Constraints

- D can be written as follows:

$$\sqrt{\mathrm{D}}^{\mathrm{T}}\sqrt{\mathrm{D}}\,,$$

where $\sqrt{\mathrm{D}}$ is the diagonal matrix whose elements in the square roots of diagonal elements

$$\sqrt{\mathrm{D}} = \begin{pmatrix} \sqrt{a_1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sqrt{a_d} \end{pmatrix}$$

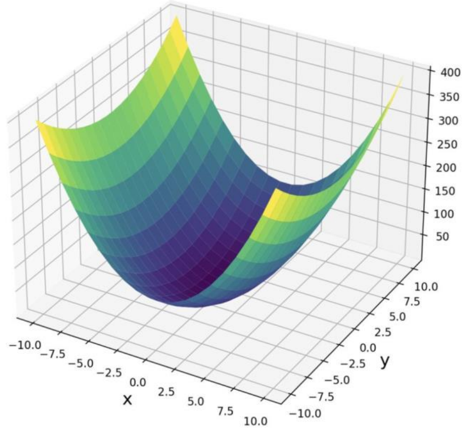


Convex Optimization with Constraints

Then if we set (We transform this problem to a simple form as follows)

$$\mathbf{y} = \sqrt{\mathrm{D}}\,\mathrm{U}\,\mathbf{x}, \text{ so } \mathrm{U}^{\mathrm{T}}\sqrt{\mathrm{D}}^{-1}\,\mathbf{y} = \mathbf{x} \text{ and } \mathbf{x}^{\mathrm{T}}\mathbf{M}\mathbf{x} = \mathbf{y}^{\mathrm{T}}\mathbf{y}$$

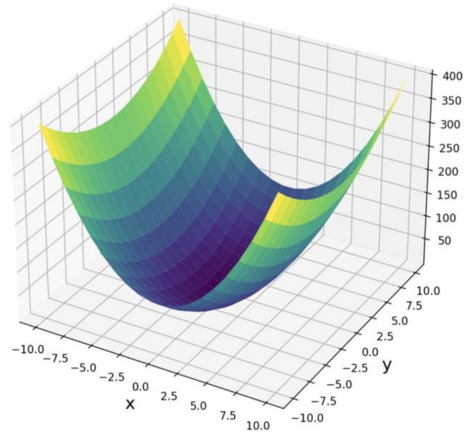
Then, the problem will be transformed into



Minimize $\mathbf{b}\,\mathrm{U}^{\mathrm{T}}\sqrt{\mathrm{D}}^{-1}\,\mathbf{y}$

Subject to $\mathbf{y}^{\mathrm{T}}\mathbf{y} - \mathrm{c} \leq 0.$

Convex Optimization with Constraints



Now we address this simpler issue

$$\text{Minimize } \mathbf{b}\, \mathrm{U}^{\mathrm{T}} \sqrt{\mathrm{D}}^{-1} \mathbf{y}$$

$$\text{Subject to } \mathbf{y}^{\mathbf{T}}\mathbf{y} - \mathrm{c} \leq 0.$$

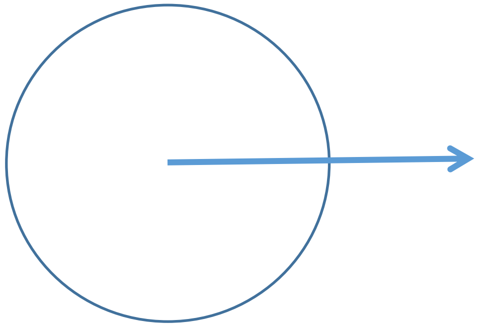
Using **Theorem 8,** we obtain that: if $\mathbf{y_0}$ is the optimal solution, then

$$\nabla \mathrm{f}(\mathbf{y}_0)^{\mathrm{T}}(\mathbf{y} - \mathbf{y}_0) \geq 0, \text{ for all } \mathbf{y} \text{ satisfy } \mathbf{y}^{\mathbf{T}}\mathbf{y} - \mathrm{c} \leq 0.$$

which is equal to $\mathbf{b}\, \mathrm{U}^{\mathrm{T}} \sqrt{\mathrm{D}}^{-1} \mathbf{y} \geq \mathbf{b}\, \mathrm{U}^{\mathrm{T}} \sqrt{\mathrm{D}}^{-1} \mathbf{y}_0$

Convex Optimization with Constraints

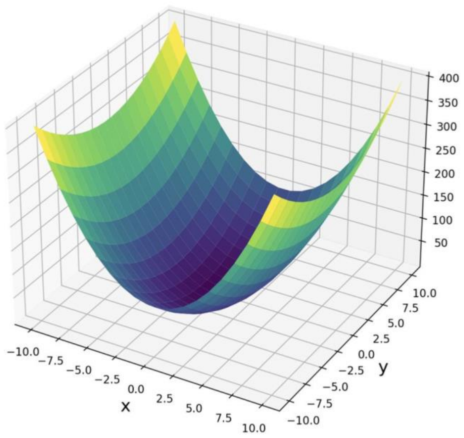
Note that $\mathbf{y}^{\mathbf{T}}\mathbf{y} - \mathrm{c} \leq 0$ means a ball with radius $\sqrt{\mathrm{c}}$.



$\sqrt{\mathrm{D}}^{-1}\mathrm{U}\mathbf{b}^{\mathbf{T}}$

$$\mathbf{b}\,\mathrm{U}^{\mathrm{T}}\sqrt{\mathrm{D}}^{-1}\,\mathbf{y} \geq \mathbf{b}\,\mathrm{U}^{\mathrm{T}}\sqrt{\mathrm{D}}^{-1}\,\mathbf{y}_0$$

means that inner product between the optimal solution $\mathbf{y}_0$ and $\mathbf{b}\,\mathrm{U}^{\mathrm{T}}\sqrt{\mathrm{D}}^{-1}$ should be smallest in the ball.



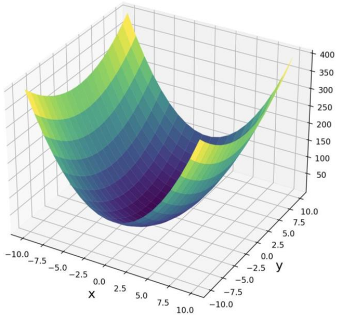
Convex Optimization with Constraints

Cauchy inequality: *https://en.wikipedia.org/wiki/Cauchy%E2%80%93Schwarz_inequality*

$$\mathbf{x}^{\mathrm{T}}\mathbf{y} \geq -\|\mathbf{x}\|\|\mathbf{y}\| \quad (\|*\| \text{ is L2 norm})$$

and $\mathbf{x}^{\mathrm{T}}\mathbf{y} = -\|\mathbf{x}\|\|\mathbf{y}\|$ if and only if $\mathbf{x} = -k\mathbf{y}$, where k is any positive constant.

Using Cauchy inequality

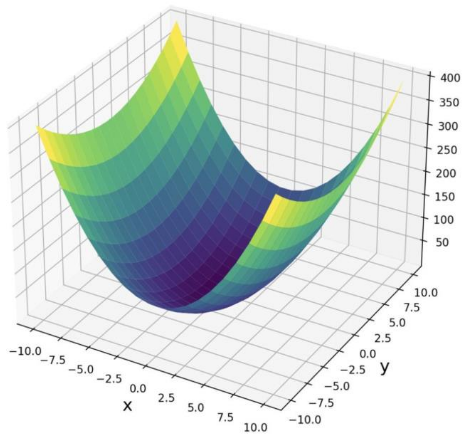


$$\mathbf{b}\,\mathrm{U}^{\mathrm{T}}\sqrt{\mathrm{D}}^{-1}\mathbf{y} \geq \mathbf{b}\,\mathrm{U}^{\mathrm{T}}\sqrt{\mathrm{D}}^{-1}\mathbf{y}_0 \geq -\left\|\mathbf{b}\,\mathrm{U}^{\mathrm{T}}\sqrt{\mathrm{D}}^{-1}\right\|\|\mathbf{y}_0\| \geq -\sqrt{\mathrm{c}}\left\|\mathbf{b}\,\mathrm{U}^{\mathrm{T}}\sqrt{\mathrm{D}}^{-1}\right\|$$

Convex Optimization with Constraints

So

$$\mathbf{b}\,\mathrm{U}^{\mathrm{T}}\sqrt{\mathrm{D}}^{-1}\mathbf{y} \geq -\sqrt{\mathrm{c}}\,\left\|\mathbf{b}\,\mathrm{U}^{\mathrm{T}}\sqrt{\mathrm{D}}^{-1}\right\|$$



If we take $\mathbf{y}_0 = -\sqrt{\mathrm{c}}\sqrt{\mathrm{D}}^{-1}\mathrm{U}\mathbf{b}^{\mathbf{T}} / \left\|\mathbf{b}\,\mathrm{U}^{\mathrm{T}}\sqrt{\mathrm{D}}^{-1}\right\|$,

Then, it is clear that $\mathbf{b}\,\mathrm{U}^{\mathrm{T}}\sqrt{\mathrm{D}}^{-1}\mathbf{y}_0 = -\sqrt{\mathrm{c}}\,\left\|\mathbf{b}\,\mathrm{U}^{\mathrm{T}}\sqrt{\mathrm{D}}^{-1}\right\|$.

So $\mathbf{y}_0 = -\sqrt{\mathrm{c}}\sqrt{\mathrm{D}}^{-1}U\mathbf{b}^{\mathbf{T}} / \left\|\mathbf{b}\,\mathrm{U}^{\mathrm{T}}\sqrt{\mathrm{D}}^{-1}\right\|$ is the optimal solution.

Convex Optimization with Constraints

So $\mathbf{x}_0 = \mathrm{U}^{\mathrm{T}}\sqrt{\mathrm{D}}^{-1}\,\mathbf{y}_0 = -\sqrt{\mathrm{c}}\,\mathrm{U}^{\mathrm{T}}\sqrt{\mathrm{D}}^{-1}\sqrt{\mathrm{D}}^{-1}\mathrm{U}\mathbf{b}^{\mathbf{T}} / \left\|\mathbf{b}\,\mathrm{U}^{\mathrm{T}}\sqrt{\mathrm{D}}^{-1}\right\|$

$$= -\ \sqrt{\mathrm{c}}\,\mathbf{M}^{-1}\mathbf{b}^{\mathbf{T}} / \left\|\mathbf{b}\,\mathrm{U}^{\mathrm{T}}\sqrt{\mathrm{D}}^{-1}\right\|$$

$$\left\|\mathbf{b}\,\sqrt{\mathrm{D}}^{-1}\mathrm{U}^{\mathrm{T}}\right\| = \sqrt{\mathbf{b}\,\mathrm{U}^{\mathrm{T}}\sqrt{\mathrm{D}}^{-1}\,\sqrt{\mathrm{D}}^{-1}\mathrm{U}\mathbf{b}^{\mathrm{T}}} = \sqrt{\mathbf{b}\mathbf{M}^{-1}\mathbf{b}^{\mathbf{T}}}$$

So, the optimal solution should be $-\ \sqrt{\mathrm{c}}\,\mathbf{M}^{-1}\mathbf{b}^{\mathbf{T}} / \sqrt{\mathbf{b}\mathbf{M}^{-1}\mathbf{b}^{\mathbf{T}}}$

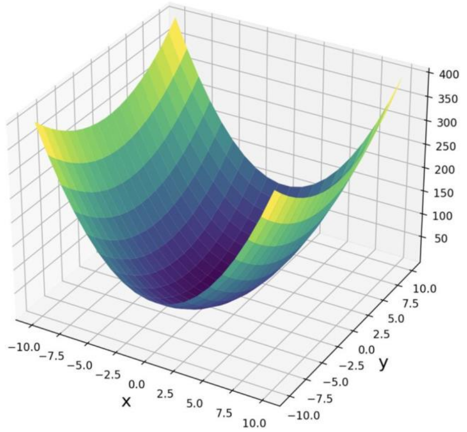
Convex Optimization with Constraints

$$\mathbf{M}^{-\mathbf{1}} = \begin{pmatrix} 0.75 & 0.5 & 0.25 \\ 0.5 & 1 & 0.5 \\ 0.25 & 0.5 & 0.75 \end{pmatrix}, \mathbf{b} \; = \; (1,0,0), \mathrm{c} = 1$$

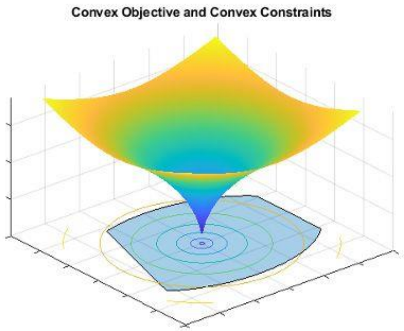
So the solution

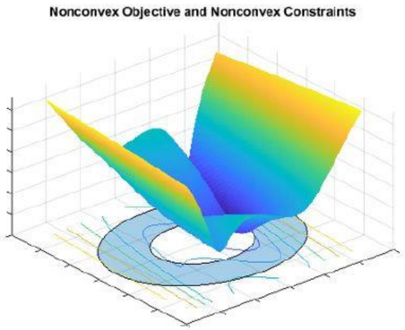
$$-\ \sqrt{\mathrm{c}}\, \mathbf{M}^{-1}\mathbf{b}^{\mathbf{T}} / \sqrt{\mathbf{b}\mathbf{M}^{-1}\mathbf{b}^{\mathbf{T}}}$$

is equal to $\begin{pmatrix} -\dfrac{\sqrt{3}}{2} \\ -\dfrac{\sqrt{3}}{3} \\ -\dfrac{\sqrt{3}}{6} \end{pmatrix}$



Non-Convex Optimization





General optimization problem

- very difficult to solve
- methods involve some compromise, e.g., very long computation time, or not always finding the solution (which may not matter in practice)

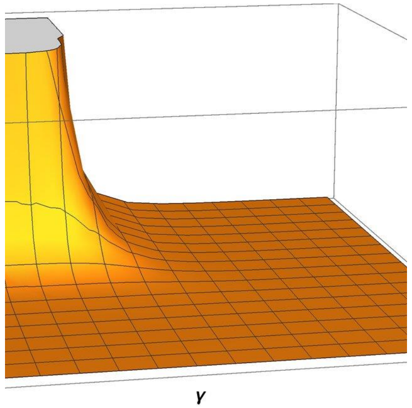
- The non-convex optimization is difficult to address.

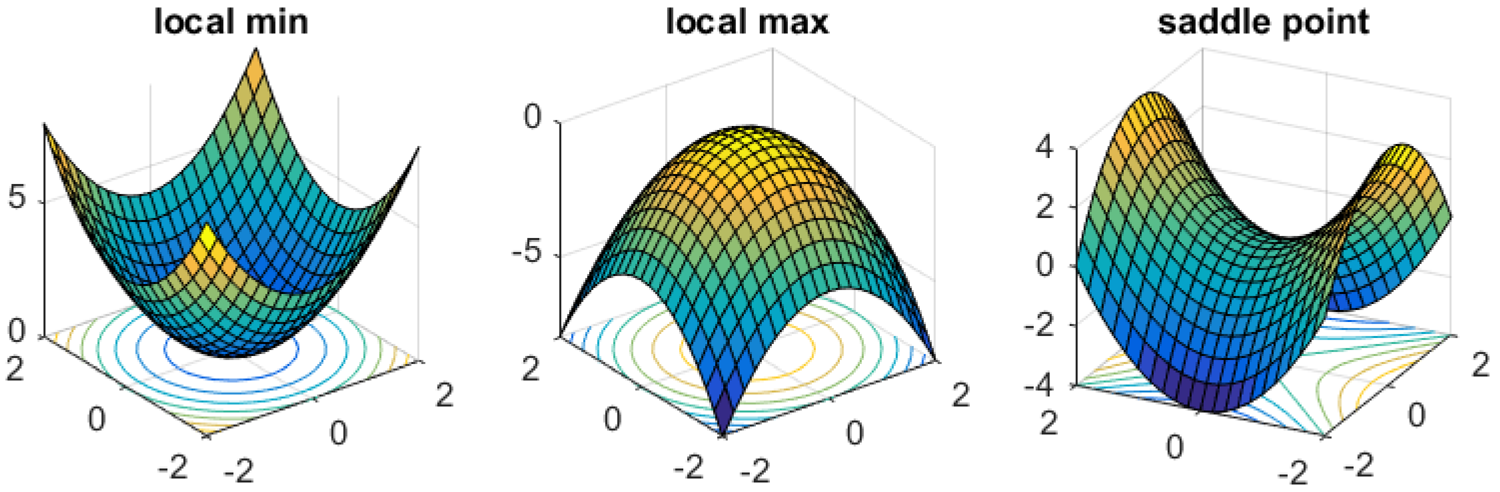
- But it is very impratant in machine learning, because of the deep neural networks.

Non-Convex Optimization

Why is non-convex optimization hard?

- Potentially many local minimal points
- Saddle points
- Very flat regions





Credits from Dr. Christopher De Sa

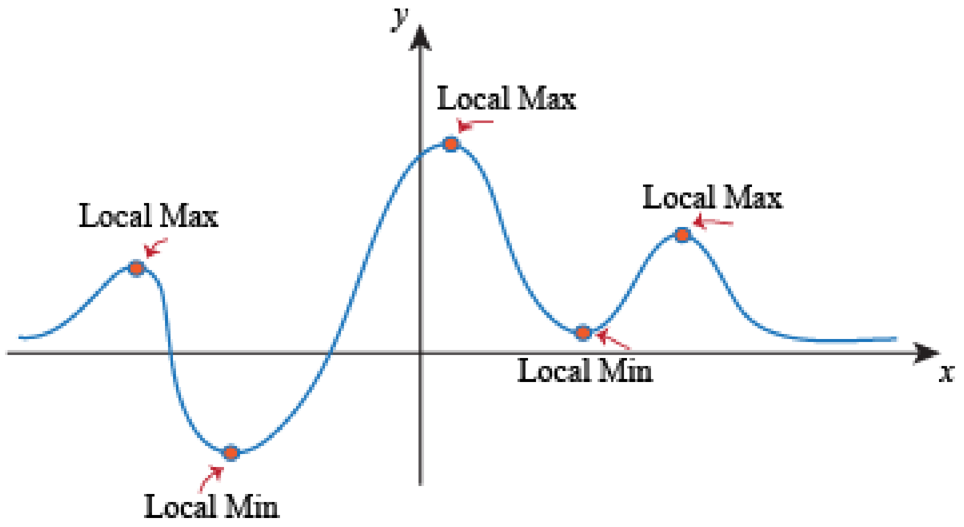
Non-Convex Optimization

- **Global minimum point:**

A real-valued function f defined on a domain Ω has a global maximum point at x∗, if f(x∗) ≤ f(x) for all x in Ω.

- **Local minimum point:**

A real-valued function f defined on a domain Ω has a local maximum point at x∗, if there exists some ε > 0 such that f(x∗) ≤ f(x) for all x in Ω within distance ε of x∗



Credits from Wikipedia

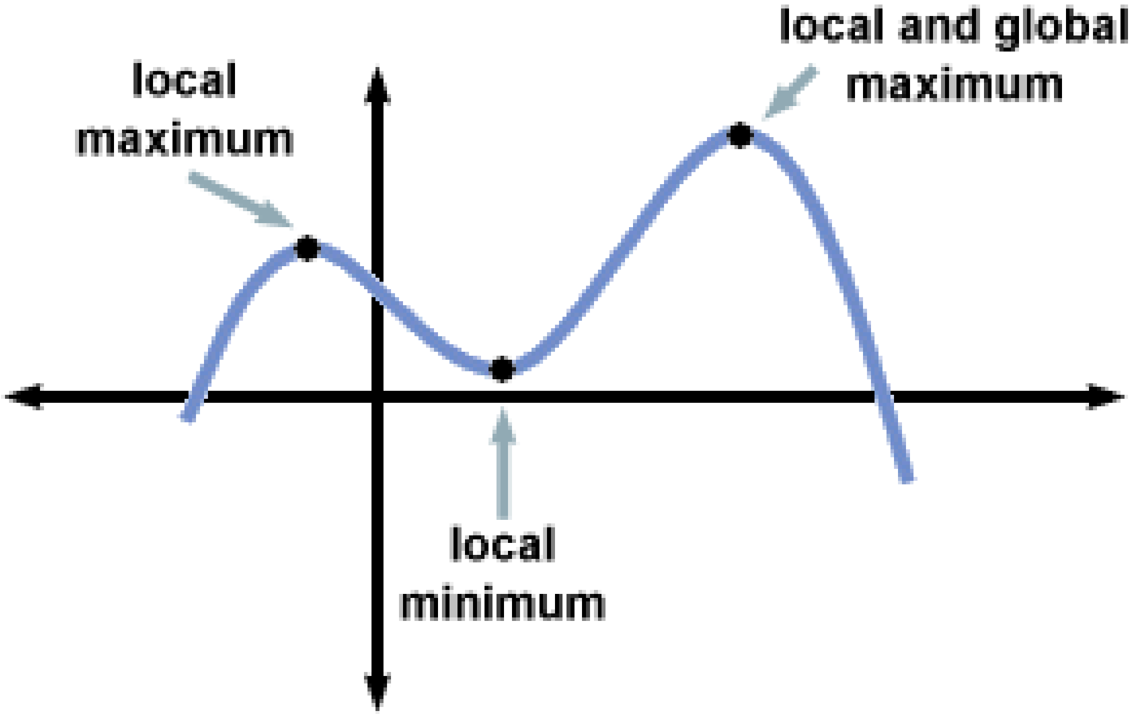
Non-Convex Optimization

- **Global maximum point:**

A real-valued function f defined on a domain Ω has a global maximum point at x∗, if f(x∗) ≥ f(x) for all x in Ω.

- **Local maximum point:**

A real-valued function f defined on a domain Ω has a local maximum point at x∗, if there exists some ε > 0 such that f(x∗) ≥ f(x) for all x in Ω within distance ε of x∗

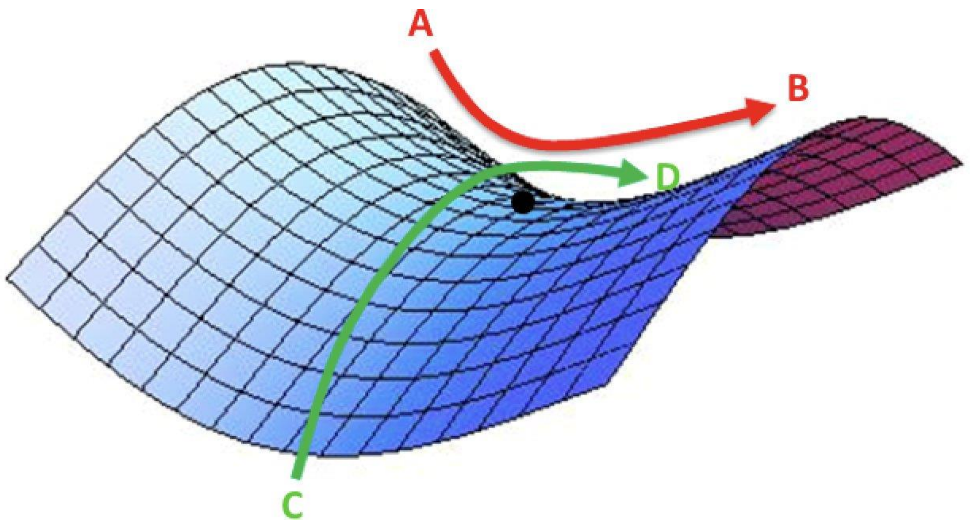


Credits from Wikipedia

Non-Convex Optimization

- **Saddle Point:**

A saddle point or minimax point is a point on the surface of the graph of a function where the slopes (derivatives) in orthogonal directions are all zero (a critical point), but which is not a local extremum (local minimal point or local maximal point) of the function.



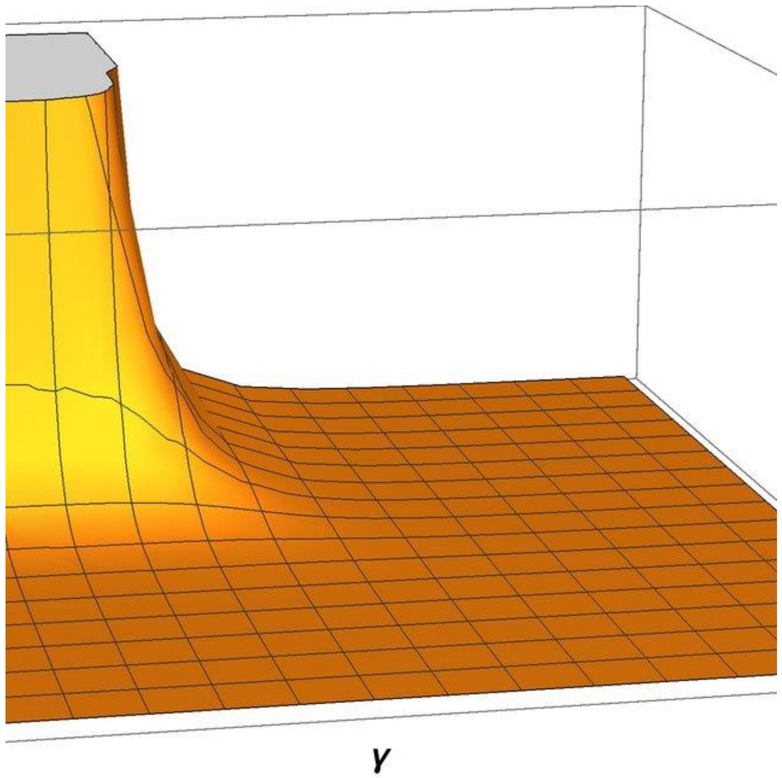
For example, x^2-y^2 (see the figure). (0,0) is a saddle point, because the gradient at (0,0) is zero, but it is not the local extremum

Credits from Wikipedia

Non-Convex Optimization

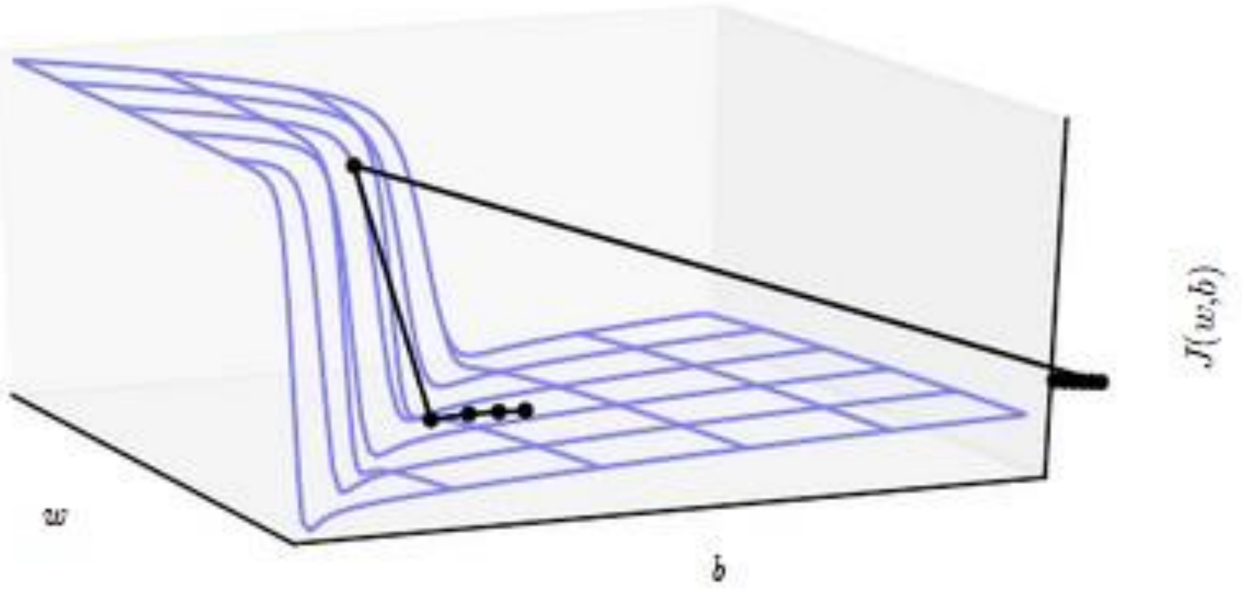
- **Very Flat Regions:**

Very Flat Regions is an area on the surface of the graph of a function where the gredients (derivatives) in orthogonal directions are very close to zero, but which is not a local extremum of the function.



Credits from Wikipedia

Cliffs and Exploding Gradients



Neural networks with many layers will have cliffs and exploding gradients. Therefore, gradient clipping is useful

(Credits from Deep Learning Book)

Non-Convex Optimization

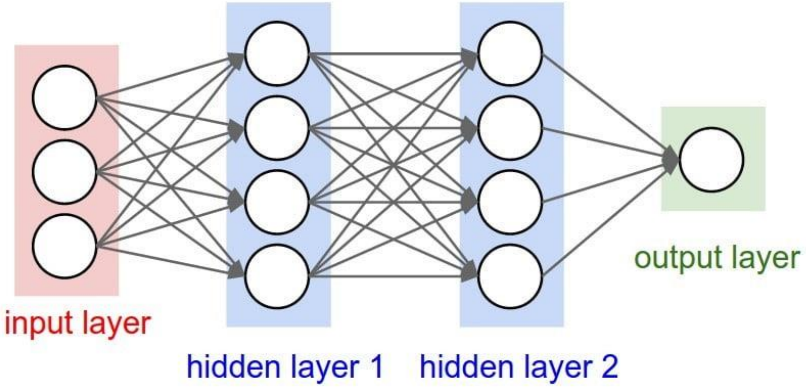
Two Sum Problem

[2, 4, 7, 8, 9] 11

Why is non-convex optimization hard?

- Non-convex optimization is at least NP-hard
- Example: subset sum problem
  Given a set of integers, is there a non-empty subset whose sum is zero?
  Known to be NP-complete.
  https://en.wikipedia.org/wiki/Subset_sum_problem

Non-Convex Optimization

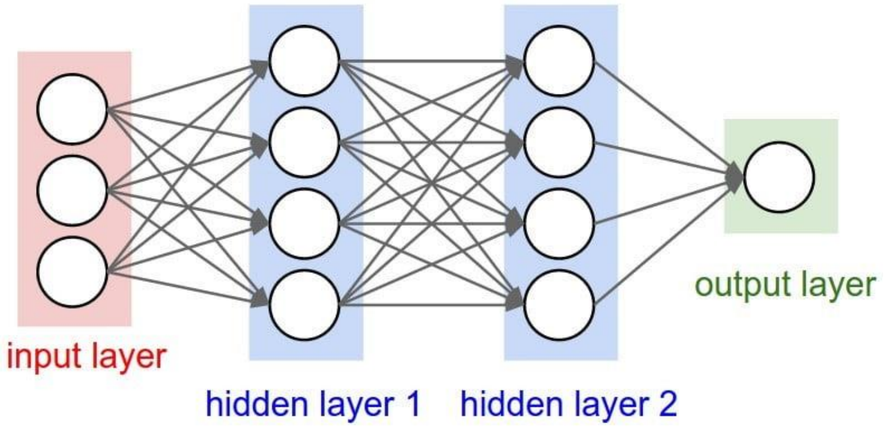


Examples of non-convex problems

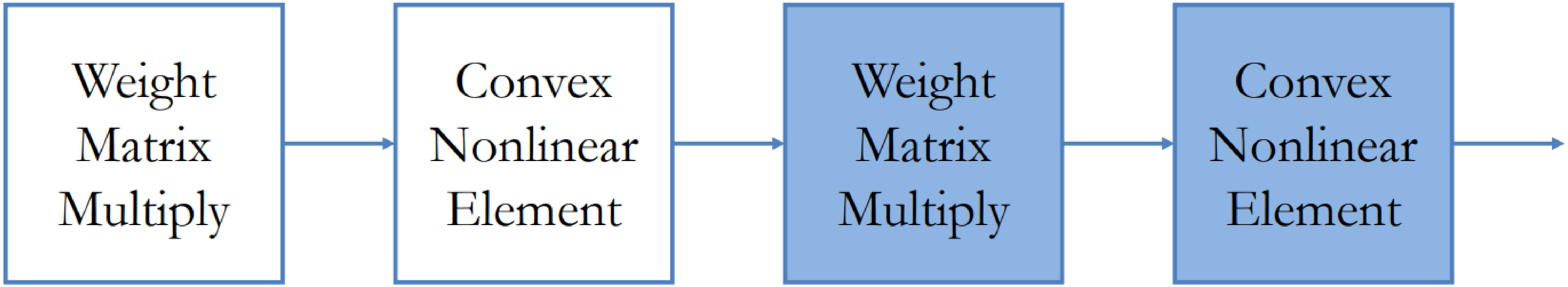
- Matrix completion, principle component analysis
- Low-rank models and tensor decomposition
- Maximum likelihood estimation with hidden variables (Usually non-convex)
- The big one: deep neural networks

Credits from Dr. Christopher De Sa

Non-Convex Optimization



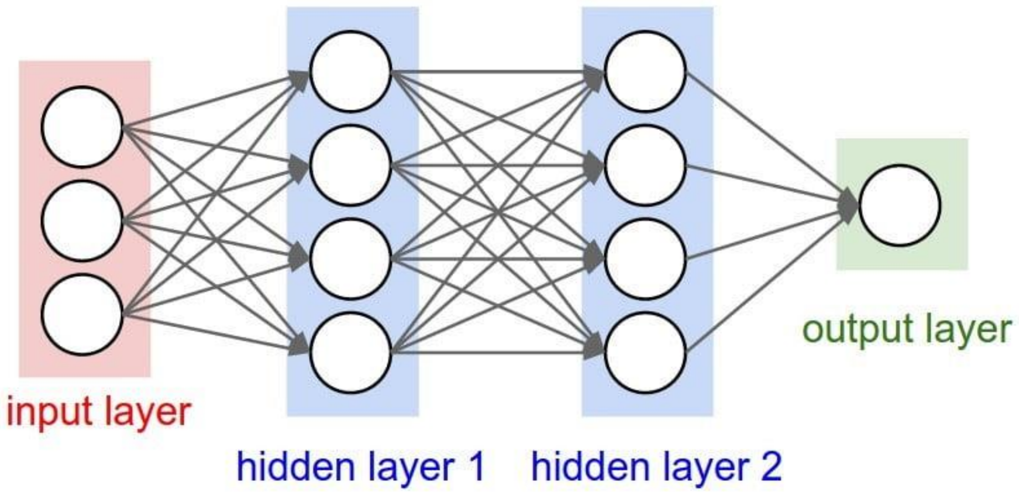
Why are neural networks non-convex?



- Composition of convex functions is not convex
- So deep neural networks also aren't convex

Credits from Dr. Christopher De Sa

Non-Convex Optimization



Why do neural nets need to be non-convex?

- Neural networks are universal function approximators

That is with enough neurons, they can learn to approximate any function arbitrarily well

- To do this, they need to be able to approximate non-convex functions

Convex functions can't approximate non-convex ones well.

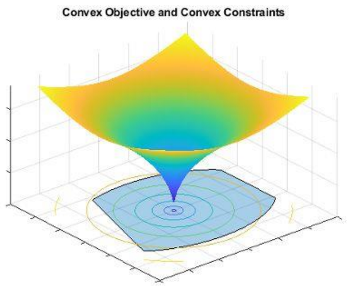
Credits from Dr. Christopher De Sa

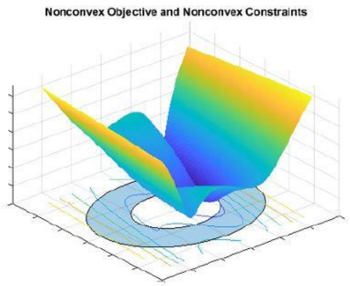
Non-Convex Optimization

How to solve non-convex problems?

- Gradient descent
- Stochastic gradient descent https://en.wikipedia.org/wiki/Stochastic_gradient_descent
- Adaptive gradient algorithm https://conferences.mpi-inf.mpg.de/adfocs/material/alina/adaptive-L1.pdf
- RMSprop https://optimization.cbe.cornell.edu/index.php?title=RMSProp
- Momentum https://en.wikipedia.org/wiki/Momentum

Non-Convex Optimization





How to solve non-convex problems?

There are also specialized methods for solving non-convex problems

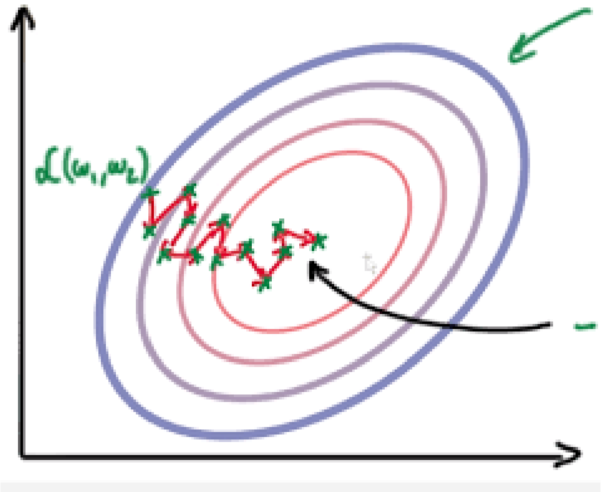
- Alternating minimization methods https://web.eecs.umich.edu/~fessler/course/598/l/n-06-alt.pdf
- Branch-and-bound methods http://web.tecnico.ulisboa.pt/mcasquilho/compute/_linpro/TaylorB_module_c.pdf
- These generally aren't very popular for machine learning problems

Credits from Dr. Christopher De Sa

Non-Convex Optimization

How to solve non-convex problems?

- If you are interested in different optimization strategies related to deep learning and machine learning, you can watch the following video as a begining. *https://mofanpy.com/tutorials/machine-learning/torch/optimizer*



In this class, we mainly introduce
Gradient descent

Gradient Descent

- Consider the following non-convex optimization problem:

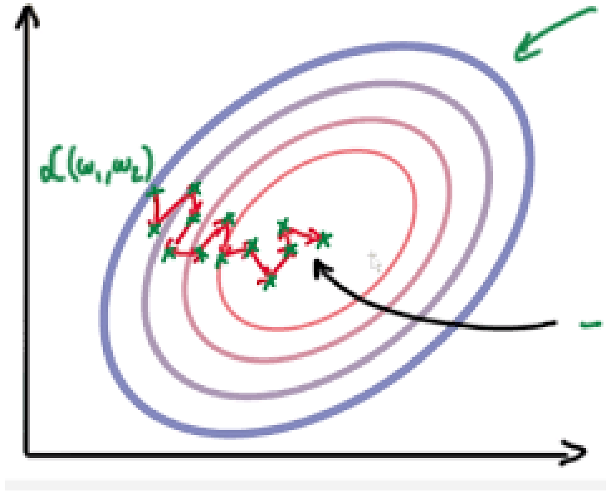
Minimize f(**x**),

where f is non-convex and is defined in R^{d}.

Given an inital point $\mathbf{x_0}$ (which is also called initial weight)

Then the next updated point should be

$$\mathbf{x_1} = \mathbf{x_0} - t\nabla \mathrm{f}(\mathbf{x_0})$$

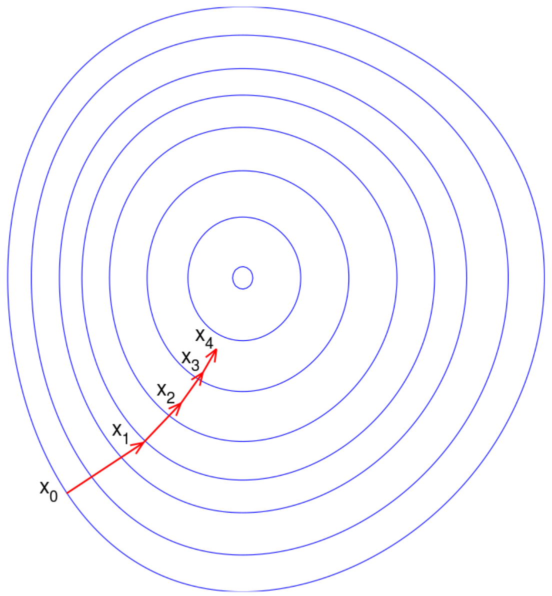


where t (t>0) is called learning rate.

Gradient Descent

After we obtain the n-th weight $\mathbf{x_n}$, then the (n+1)-th weight $\mathbf{x_{n+1}}$ is

$$\mathbf{x_{n+1}} = \mathbf{x_n} - t\nabla f(\mathbf{x_n})$$

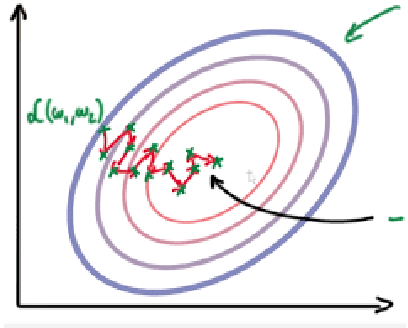


Motivation 1. We hope the final point $\mathbf{x}$ will get close to the a critical point $\nabla f(\mathbf{x})=0$.

Motivation 2. to take repeated steps in the opposite direction of the gradient (or approximate gradient) of the function at the current point, because this is the direction of steepest descent (下降最快的方向).

- Doesn’t necessarily go towards optimal point.

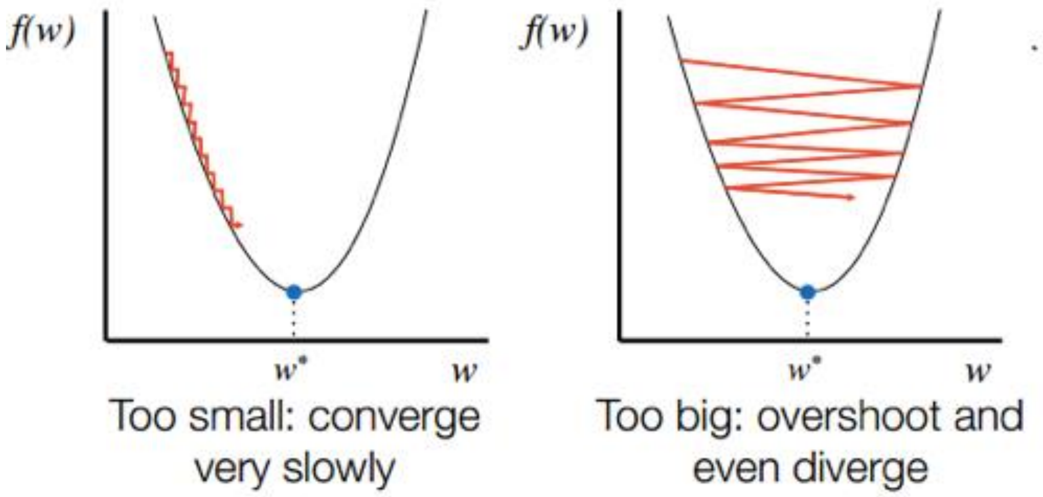
Gradient Descent



- The convergent point is the optimal point if the function is convex!
- The hardware doesn't care whether our gradients are from a convex function or not.
- This means that all our intuition about computational efficiency from the convex case directly applies to the non-convex case.

Gradient Descent

The influence of Step Size (Learning Rate t)



- Choosing a good Learning Rate is important in gradient descent

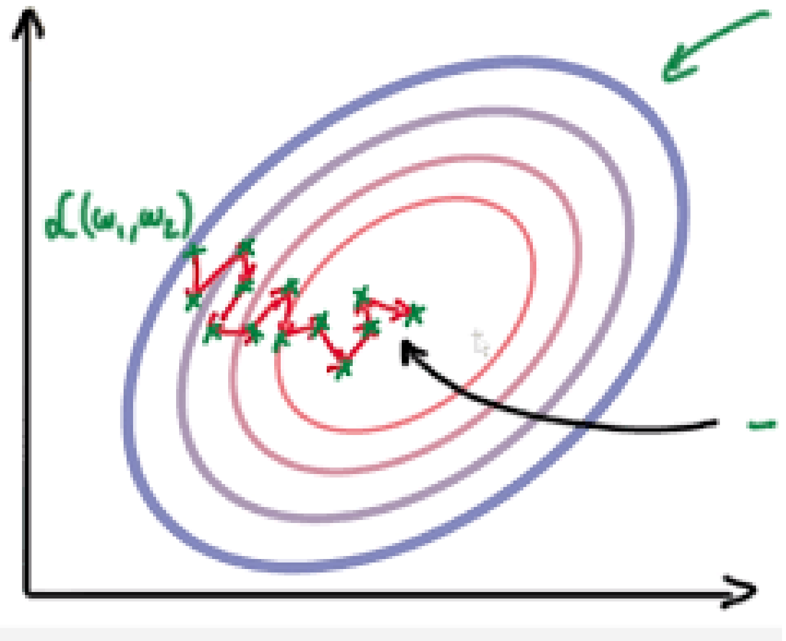
- Heuristics
  - When the function value increases after a gradient step, the step-size was too large. Undo the step and decrease the step-size.
  - When the function value decreases the step could have been larger. Try to increase the step-size.

Credits from Dr. Yang Liu

Gradient Descent

What points can Gradient Descent converge?

- Local minimal points
- Saddle points
- Miminal points
- Points in Very Flat Regions

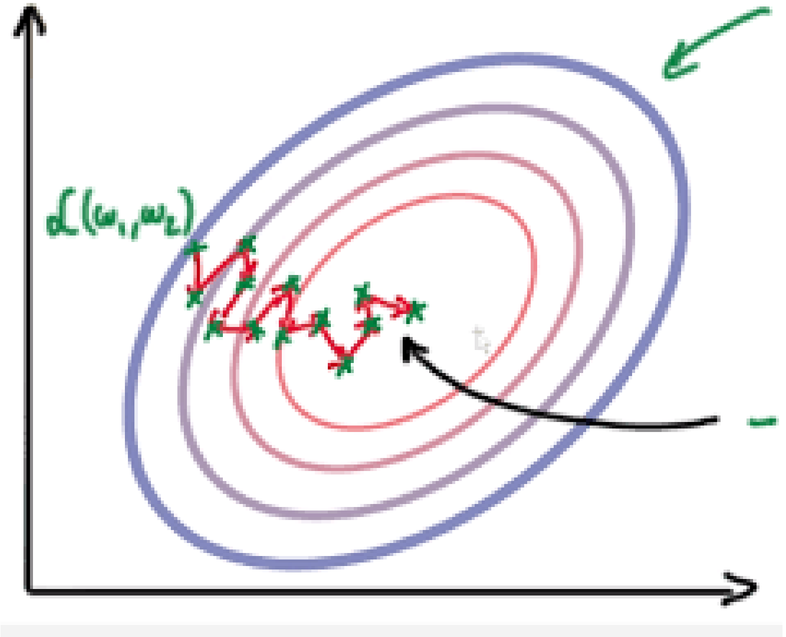


Gradient Descent

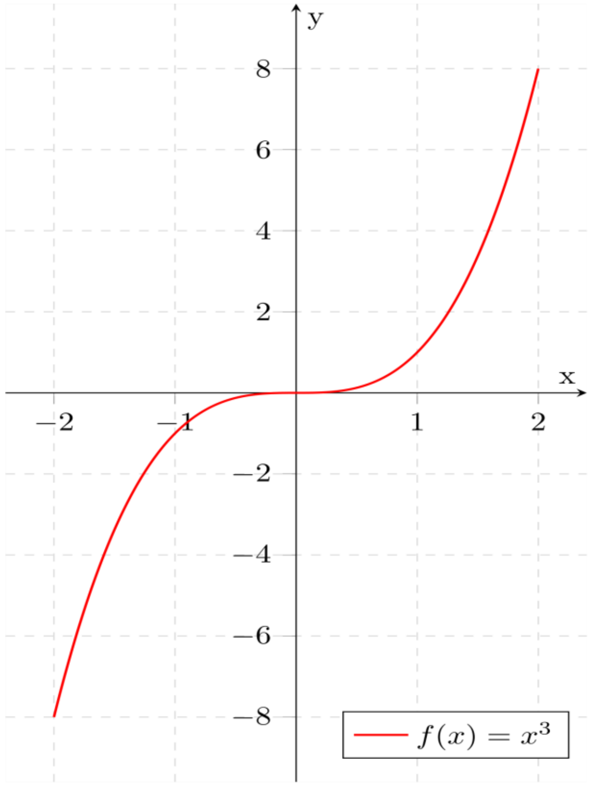
Exercises

- Minimize f(x), where f(x) = x^3.

Initial weight x_0 = 1, learning rate t = 1/3. Then what is the convergent point?



Gradient Descent



Solution

- Minimize f(x), where f(x) = x^3.

Then the grendent descent formula is

$$x_{n+1} = x_n - 3tx_n^{\ 2}$$

Because x_0= 1, t= 1/3, then

$$x_1=0$$

Note that $\nabla f(\mathbf{x_1})$ = 0, so x_2=x_1, which implies $x_{n+1} = x_1 = 0$.

So 0 is the convergent point. But 0 is a saddle point of f(x).

Stochastic Gradient Descent

Algorithm 8.1 Stochastic gradient descent (SGD) update

Require: Learning rate schedule $\epsilon_1, \epsilon_2, \ldots$

Require: Initial parameter $\boldsymbol{\theta}$

$k \leftarrow 1$

while stopping criterion not met **do**

Sample a minibatch of m examples from the training set $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ with corresponding targets $\boldsymbol{y}^{(i)}$.

Compute gradient estimate: $\hat{\boldsymbol{g}} \leftarrow \frac{1}{m} \nabla_{\boldsymbol{\theta}} \sum_i L(f(\boldsymbol{x}^{(i)}; \boldsymbol{\theta}), \boldsymbol{y}^{(i)})$

Apply update: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \epsilon_k \hat{\boldsymbol{g}}$

$k \leftarrow k + 1$

end while

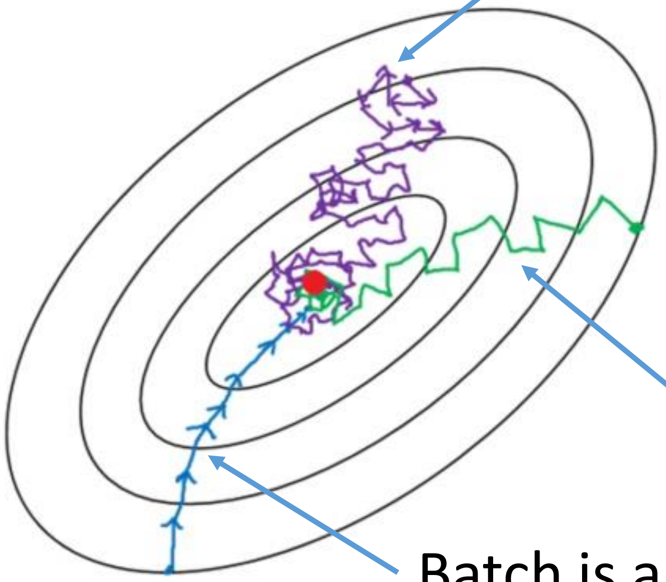
Sample i uniformly from $\{1, \cdots, n\}$, and update θ by

$$\theta = \theta - \epsilon \nabla_{\theta} J^{(i)}(\theta)$$

Mini-batch SGD

- When $B = 1$, we conduct stochastic gradient descent
- When $B = n$, we conduct full-batch gradient desecent

Stochastic is fast but unstable

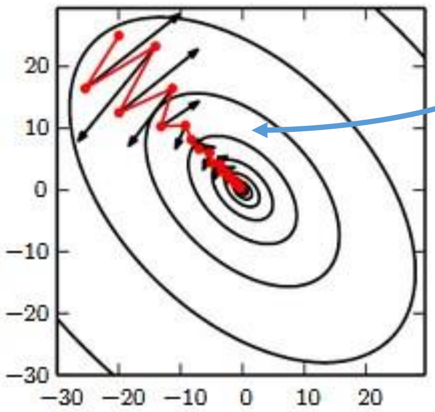


Mini-batch is a compromise between efficiency and efficacy

Batch is a stable but slow

(Credits from Deep Learning Book)

Momentum



$$v \leftarrow \alpha v - \epsilon \nabla_\theta J^{(i)}(\theta)$$
$$\theta \leftarrow \theta + v$$

Algorithm 8.2 Stochastic gradient descent (SGD) with momentum

Require: Learning rate ϵ, momentum parameter α

Require: Initial parameter $\boldsymbol{\theta}$, initial velocity $\boldsymbol{v}$

while stopping criterion not met **do**

Sample a minibatch of m examples from the training set $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ with corresponding targets $\boldsymbol{y}^{(i)}$.

Compute gradient estimate: $\boldsymbol{g} \leftarrow \frac{1}{m} \nabla_{\boldsymbol{\theta}} \sum_i L(f(\boldsymbol{x}^{(i)}; \boldsymbol{\theta}), \boldsymbol{y}^{(i)})$.

Compute velocity update: $\boldsymbol{v} \leftarrow \alpha \boldsymbol{v} - \epsilon \boldsymbol{g}$.

Apply update: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \boldsymbol{v}$.

end while

(Credits from Deep Learning Book)

Nesterov Momentum

$$v \leftarrow \alpha v - \epsilon \nabla_\theta J^{(i)}(\theta + \alpha v)$$
$$\theta \leftarrow \theta + v$$

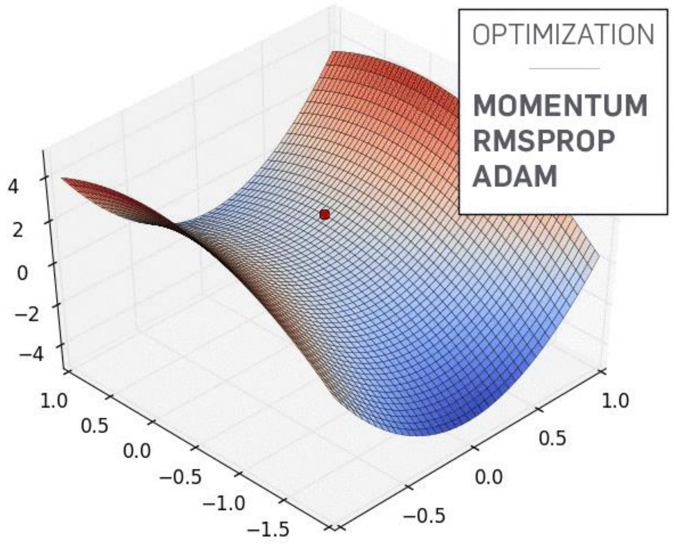
Algorithm 8.3 Stochastic gradient descent (SGD) with Nesterov momentum

Require: Learning rate ϵ, momentum parameter α
Require: Initial parameter $\boldsymbol{\theta}$, initial velocity $\boldsymbol{v}$
while stopping criterion not met **do**
Sample a minibatch of m examples from the training set $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ with corresponding labels $\boldsymbol{y}^{(i)}$.
Apply interim update: $\tilde{\boldsymbol{\theta}} \leftarrow \boldsymbol{\theta} + \alpha \boldsymbol{v}$.
Compute gradient (at interim point): $\boldsymbol{g} \leftarrow \frac{1}{m} \nabla_{\tilde{\boldsymbol{\theta}}} \sum_i L(f(\boldsymbol{x}^{(i)}; \tilde{\boldsymbol{\theta}}), \boldsymbol{y}^{(i)})$.
Compute velocity update: $\boldsymbol{v} \leftarrow \alpha \boldsymbol{v} - \epsilon \boldsymbol{g}$.
Apply update: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \boldsymbol{v}$.
end while

(Credits from Deep Learning Book)

Adaptive Algorithms

- AdaGrad
- RMSProp
- Adam



AdaGrad

Algorithm 8.4 The AdaGrad algorithm

Require: Global learning rate ϵ
Require: Initial parameter $\boldsymbol{\theta}$
Require: Small constant δ, perhaps 10^{-7}, for numerical stability
  Initialize gradient accumulation variable $\boldsymbol{r} = \mathbf{0}$
  while stopping criterion not met **do**
    Sample a minibatch of m examples from the training set $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ with corresponding targets $\boldsymbol{y}^{(i)}$.
    Compute gradient: $\boldsymbol{g} \leftarrow \frac{1}{m} \nabla_{\boldsymbol{\theta}} \sum_i L(f(\boldsymbol{x}^{(i)}; \boldsymbol{\theta}), \boldsymbol{y}^{(i)})$.
    Accumulate squared gradient: $\boldsymbol{r} \leftarrow \boldsymbol{r} + \boldsymbol{g} \odot \boldsymbol{g}$.
    Compute update: $\Delta\boldsymbol{\theta} \leftarrow -\frac{\epsilon}{\delta + \sqrt{\boldsymbol{r}}} \odot \boldsymbol{g}$. (Division and square root applied element-wise)
    Apply update: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \Delta\boldsymbol{\theta}$.
  end while

$$g_t = \nabla_{\theta_t} J(\theta_t)$$

AdaGrad: $\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{G_t + \epsilon}} \odot g_t$

(Credits from Deep Learning Book)

RMSProp

Algorithm 8.5 The RMSProp algorithm

Require: Global learning rate ϵ, decay rate ρ

Require: Initial parameter $\boldsymbol{\theta}$

Require: Small constant δ, usually 10^{-6}, used to stabilize division by small numbers

Initialize accumulation variables $\boldsymbol{r} = 0$

while stopping criterion not met **do**

Sample a minibatch of m examples from the training set $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ with corresponding targets $\boldsymbol{y}^{(i)}$.

Compute gradient: $\boldsymbol{g} \leftarrow \frac{1}{m} \nabla_{\boldsymbol{\theta}} \sum_i L(f(\boldsymbol{x}^{(i)}; \boldsymbol{\theta}), \boldsymbol{y}^{(i)})$.

Accumulate squared gradient: $\boldsymbol{r} \leftarrow \rho \boldsymbol{r} + (1 - \rho) \boldsymbol{g} \odot \boldsymbol{g}$.

Compute parameter update: $\Delta\boldsymbol{\theta} = -\frac{\epsilon}{\sqrt{\delta + \boldsymbol{r}}} \odot \boldsymbol{g}$. ($\frac{1}{\sqrt{\delta + \boldsymbol{r}}}$ applied element-wise)

Apply update: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \Delta\boldsymbol{\theta}$.

end while

Changing the gradient accumulation into an exponentially weighted moving average

(Credits from Deep Learning Book)

Adam

Algorithm 8.7 The Adam algorithm

Require: Step size ϵ (Suggested default: 0.001)
Require: Exponential decay rates for moment estimates, ρ_1 and ρ_2 in $[0, 1)$. (Suggested defaults: 0.9 and 0.999 respectively)
Require: Small constant δ used for numerical stabilization (Suggested default: 10^{-8})
Require: Initial parameters $\boldsymbol{\theta}$
  Initialize 1st and 2nd moment variables $\boldsymbol{s} = \boldsymbol{0}$, $\boldsymbol{r} = \boldsymbol{0}$
  Initialize time step $t = 0$
  while stopping criterion not met **do**
    Sample a minibatch of m examples from the training set $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ with corresponding targets $\boldsymbol{y}^{(i)}$.
    Compute gradient: $\boldsymbol{g} \leftarrow \frac{1}{m} \nabla_{\boldsymbol{\theta}} \sum_i L(f(\boldsymbol{x}^{(i)}; \boldsymbol{\theta}), \boldsymbol{y}^{(i)})$
    $t \leftarrow t + 1$
    Update biased first moment estimate: $\boldsymbol{s} \leftarrow \rho_1 \boldsymbol{s} + (1 - \rho_1)\boldsymbol{g}$
    Update biased second moment estimate: $\boldsymbol{r} \leftarrow \rho_2 \boldsymbol{r} + (1 - \rho_2)\boldsymbol{g} \odot \boldsymbol{g}$
    Correct bias in first moment: $\hat{\boldsymbol{s}} \leftarrow \frac{\boldsymbol{s}}{1-\rho_1^t}$
    Correct bias in second moment: $\hat{\boldsymbol{r}} \leftarrow \frac{\boldsymbol{r}}{1-\rho_2^t}$
    Compute update: $\Delta\boldsymbol{\theta} = -\epsilon \frac{\hat{\boldsymbol{s}}}{\sqrt{\hat{\boldsymbol{r}}}+\delta}$ (operations applied element-wise)
    Apply update: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \Delta\boldsymbol{\theta}$
  end while

$$g_t = \nabla_{\theta_t} J(\theta_t)$$

Adam: $\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t + \epsilon}} \hat{m}_t$,

where $\hat{m}_t = \frac{m_t}{1-\beta_1^t}$

and $\hat{v}_t = \frac{v_t}{1-\beta_2^t}$

$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$

and $v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$

Using Adam when you are unfamiliar about optimization; while using momentum SGD when you are familiar about optimization

(Credits from Deep Learning Book)

Thank You!