UBC CPSC 440/550 (25W2) Tutorial #1

2026-01-14 Wednesday (DMP-201 at 11 am)

Yuwei Yin

yuweiyin@cs.ubc.ca
https://www.yuweiyin.com/

.

CPSC 440/550: Advanced Machine Learning
2025-26 Winter Term 2 (Jan–Apr 2026)
https://www.cs.ubc.ca/~dsuth/440/25w2/
University of British Columbia, on unceded Musqueam land

THE UNIVERSITY OF BRITISH COLUMBIA

About Me

- Hi, I'm Yuwei
- 3rd year PhD student with Giuseppe in the UBC NLP Group
  - Research: mindful thinking and logical reasoning in AI systems
- Homepage: `https://www.yuweiyin.com/`

About Tutorials

- Tutorials are optional (Tue $\times$ 1 + Wed $\times$ 2)
    - Tutorials are supplemental to lectures
    - Lectures: canonical and taught by the professor
    - Tutorials: supplementary and useful plugins
- The content of tutorials depends on TAs
    - usually, related to lecture materials and assignments

About My Tutorials

- **Lecture materials**: review on key concepts
- **Ongoing assignments**: reinforce background knowledge
- **Past assignments/quizzes**: review after grading
- Rest of Time: **QA** (like Office Hours)
  For more QA sessions:
  - Post your questions on Piazza
  - Office hours by Danica on Thursdays

Assignment 1

- Do NOT miss the deadline (due this Friday: Jan 16 at 11:59 pm)
  Do NOT wait until the last couple of hours to work on it.
  Do Assignment 1 alone. Follow the instructions.
- Questions in Assignment 1 are (mostly) covered in CPSC 340.
  Review 340 if there are any basic concepts unclear.
  `https://www.students.cs.ubc.ca/~cs-340/`
- CPSC 440/550 materials are on the course website.
  Review the course-related materials frequently.
  `https://www.cs.ubc.ca/~dsuth/440/25w2/`
- Preview the course content via last year's course website
  `https://www.cs.ubc.ca/~dsuth/440/24w2/`

Matrix Notation

- Scalar (1 by 1): $a \in \mathbb{R}$
- Column vector (d by 1): $x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix} \in \mathbb{R}^{d \times 1}$
- Row vector (1 by d): $x^\top = \begin{bmatrix} x_1 & x_2 & \cdots & x_d \end{bmatrix} \in \mathbb{R}^{1 \times d}$
- Generally, we assume all vectors to be **column vectors**
- Matrix (n by d): $A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1d} \\ a_{21} & a_{22} & \cdots & a_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nd} \end{bmatrix} \in \mathbb{R}^{n \times d}$
- Matrix transpose: $(A^\top)_{ij} = (A)_{ji}$
  - Symmetric matrix: $A^\top = A$

Matrix Algebra

Let vectors $x, y \in \mathbb{R}^d$ and matrices $A, B \in \mathbb{R}^{n\times d}$.

- Basis: determinant, rank, inverse, inner product, cross product, eigenvalue, eigenvector, gaussian elimination, change of basis ...
- Inner product: $\langle x, y\rangle = y^\top x = x^\top y = \sum_{i=1}^d x_i y_i$
- Euclidean norm (L2 norm): $\|x\|_2 = \sqrt{\langle x, x\rangle} = \sqrt{\sum_{i=1}^d x_i^2}$
- Matrix-vector product:
  - $Ax = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1d} \\ a_{21} & a_{22} & \cdots & a_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nd} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix} = \begin{bmatrix} \sum_{j=1}^d a_{1j}x_j \\ \sum_{j=1}^d a_{2j}x_j \\ \vdots \\ \sum_{j=1}^d a_{nj}x_j \end{bmatrix}$
  - $(Ax)_i = \sum_{j=1}^d a_{ij}x_j$
  - $y^\top Ax = y^\top(Ax) = \sum_{i=1}^d \sum_{j=1}^d y_i a_{ij} x_j = (Ax)^\top y = x^\top A^\top y$

Matrix Algebra

Let vectors $x, y \in \mathbb{R}^d$ and matrices $A, B \in \mathbb{R}^{n \times d}$.

- Matrix product: $(AB^\top)_{ij} = \sum_{k=1}^{d} (A)_{ik}(B)_{kj}$
  - Associative: $A(BC) = (AB)C$
  - Distributive over addition: $A(B + C) = AB + AC$
  - NOT commutative: generally, $AB \neq BA$
  - Transpose: $(AB)^\top = B^\top A^\top$
- Matrix powers keep the order: $(AB)^2 = ABAB$
- Identity matrix: $I_n = \underbrace{\begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}}_{n \text{ columns}}$, $I_n A = A = A I_n$
- Elementary vectors e_i: the i-th column of the identity matrix

Matrix Calculus - Jacobian

- We often want to apply m functions ($\mathbb{R} \to \mathbb{R}$) on n variables $\in \mathbb{R}$ $\Rightarrow$ we apply a vector-valued function $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^m$, which consists of m $\mathbb{R} \to \mathbb{R}$ functions, to a vector of length n.
- Question: How do we compute the derivative of $\mathbf{f}$?
- **Jacobian matrix** (the first-order partial derivative) of a vector-valued function $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^m$ is defined as

$$\mathbf{J_f} = \begin{bmatrix} \frac{\partial \mathbf{f}}{\partial x_1} & \cdots & \frac{\partial \mathbf{f}}{\partial x_n} \end{bmatrix} = \begin{bmatrix} \nabla^\top f_1 \\ \vdots \\ \nabla^\top f_n \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix} \in \mathbb{R}^{m \times n},$$

  where $\nabla^\top f_i$ is the transpose (row vector) of the gradient of the i-th component.
- The Jacobian matrix can also be denoted as $\nabla \mathbf{f}$

Matrix Calculus - Jacobian

Compute Jacobian matrix:

- *Example 1*: Let $\mathbf{f}(x) = A^\top x$, where $A \in \mathbb{R}^{n\times m}$ and $x \in \mathbb{R}^n$.
  - The ith component of $\mathbf{f}$ is $f_i = \sum_{k=1}^n a_{ik}x_k$.
  - $(\mathbf{J_f})_{ij} = \frac{\partial f_i}{\partial x_j} = \frac{\partial \sum_{k=1}^n a_{ik}x_k}{\partial x_j} = a_{ij}$.
  - $\mathbf{J_f} = A$
  - $\nabla^\top f_i = \begin{bmatrix} a_{i1} & \cdots & a_{in} \end{bmatrix}$
- In real-value cases, when $f(x) = ax$, where $f : \mathbb{R} \to \mathbb{R}$ and $a, x \in \mathbb{R}$, we have $\frac{df}{dx} = a$

Matrix Calculus - Jacobian

Compute Jacobian matrix:

- *Example 2*: Let $f(x) = x^\top Ax$, where $A \in \mathbb{R}^{n\times n}$ and $x \in \mathbb{R}^n$.
    - $f(x) = \sum_{i=1}^n \sum_{k=1}^n a_{ik}x_ix_k = \sum_{i\neq k} a_{ik}x_ix_k + \sum_{i=1}^n a_{ii}x_i^2$

$$\frac{\partial f}{\partial x_j} = (\underbrace{\sum_{k\neq j} a_{jk}x_k}_{\text{when } i=j} + \underbrace{\sum_{i\neq j} a_{ij}x_i}_{\text{when } k=j}) + \underbrace{2a_{jj}x_j}_{\text{when } i=j}$$

$$= \sum_{k=1}^n a_{jk}x_k + \sum_{i=1}^n a_{ij}x_i = (Ax)_j + (A^\top x)_j$$

-
    - $\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x_1} & \cdots & \frac{\partial f}{\partial x_n} \end{bmatrix}^\top = (A + A^\top)x$
    - If A is symmetric, meaning $A = A^\top$, we have $\nabla f = 2Ax$
- In real-value cases, when $f(x) = ax^2$, where $f : \mathbb{R} \to \mathbb{R}$ and $a, x \in \mathbb{R}$, we have $\frac{df}{dx} = 2ax$

Matrix Calculus - Hessian

- How about the second-order derivatives?
- **Hessian matrix** (the second-order partial derivatives) of a scalar-valued function $f : \mathbb{R}^n \to \mathbb{R}$ is defined as

$$\mathbf{H_f} = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f_1}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f_1}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f_1}{\partial^2 x_n} \end{bmatrix} \in \mathbb{R}^{n \times n}.$$

- $(\mathbf{H_f})_{ij} = \frac{\partial^2 f}{\partial x_i \partial x_j}$
- Hessian matrix can also be computed using $\mathbf{H}_f = \mathbf{J}(\nabla f(x))^\top$, where $\mathbf{J}(\nabla f(x))$ is the Jacobian matrix of the gradient $\nabla f(x)$.

Matrix Calculus - Hessian

Compute Hessian matrix:

- *Example*: Let $f(x) = x^\top A x$, where $A \in \mathbb{R}^{n\times n}$ and $x \in \mathbb{R}^n$.
  $f(x) = \sum_{i=1}^{d} \sum_{j=1}^{d} a_{ij}x_i x_j = \sum_{i\neq j} a_{ij}x_i x_j + \sum_{i=1}^{n} a_{ii}x_i^2$
- *Approach 1*: compute H directly from the definition
  - $(\mathbf{H}_f)_{ij} = \frac{\partial^2 f}{\partial x_i \partial x_j} = \begin{cases} a_{ij} + a_{ji}, & i \neq j \\ 2a_{ii}, & i = j \end{cases}.$
  - Thus, $\mathbf{H}_f = A + A^\top$.
- *Approach 2*: compute $\mathbf{H}_f = \mathbf{J}(\nabla f(x))^\top$
  - We already know $\nabla f(x) = (A + A^\top)x$ and $\mathbf{J}(A^\top x) = A$
  - Thus, $\mathbf{H}_f = \mathbf{J}((A + A^\top)x)^\top = ((A + A^\top)^\top)^\top = A + A^\top$.
- If A is symmetric, meaning $A = A^\top$, we have $\mathbf{H}_f = 2A$
- In real-value cases, when $f(x) = ax^2$, where $f : \mathbb{R} \to \mathbb{R}$ and $a, x \in \mathbb{R}$, we have $\frac{d^2 f}{dx^2} = 2a$

Matrix Calculus - Jacobian & Hessian

- Why do Jacobian (1st derivative) and Hessian (2nd derivative) matter?
- Critical points
  - stationary points: 1st derivative = 0 $\Rightarrow$ local minima/maxima
  - inflection points: 2nd derivative = 0 $\Rightarrow$ changes of convexity

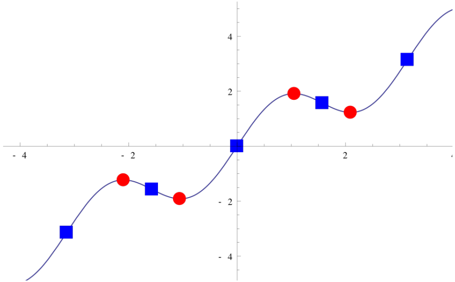

Figure 1: Stationary (red) and inflection (blue) points. (Source)

Linear Regression - Least Squares

- Assume our linear model is given by:

$$\hat{y} = Xw,$$

where $X \in \mathbb{R}^{n\times d}$, $w \in \mathbb{R}^d$, $\hat{y} \in \mathbb{R}^n$

- The least squares objective is

$$f(w) = \frac{1}{2}\|Xw - y\|^2 = \frac{1}{2}w^\top \underbrace{X^\top X}_{A} w - w^\top \underbrace{X^\top y}_{b} + \underbrace{\frac{1}{2}y^\top y}_{c}$$

- Next Steps:
  - Compute gradient and Hessian of the objective:
    - $\mathbf{J}_f = \nabla f(w) = \frac{1}{2}(A + A^\top)w + b + 0 = X^\top Xw + X^\top y$
    - $\mathbf{H}_f = \frac{1}{2}(A + A^\top) = X^\top X$
  - Determine the solution by setting Jacobian to 0 (stationary points) and verifying if Hessian is positive definite (local minima).

Matrix Definiteness

- **Definiteness** of symmetric matrix $A \in \mathbb{R}^{d \times d}$:
  - Positive semi-definite $A \succcurlyeq 0$: $\forall x \neq 0, x^\top A x \geq 0$
  - Strictly positive definite $A \succ 0$: $\forall x \neq 0, x^\top A x > 0$
- In real-value cases, we say a number a is non-negative, i.e., $a \geq 0$ $\Leftrightarrow \forall x \neq 0$, we have $ax^2 \geq 0$;
  Similarly, we say a number a is positive, i.e., $a > 0 \Leftrightarrow \forall x \neq 0$, we have $ax^2 > 0$
- For any real matrix A, the product $A^\top A$ is a positive semi-definite matrix (and also a symmetric matrix)

$$x^\top A^\top A x = (Ax)^\top (Ax) = \|Ax\|^2 \geq 0$$

- Further reading: Positive Semi-Definite Matrices

Convexity - Convex Set

- **Convex set**
  - Definition: A set C is convex if the line between any two points stays also in the set:
  - For all $w \in C$ and $v \in C$, we have $\alpha w + (1 - \alpha)v \in C$ for any $0 \leq \alpha \leq 1$
- *Example*: norm ball $\{w : \|w\|_p \leq r\}$ is a convex set
- Useful property: the intersection of convex sets is convex.

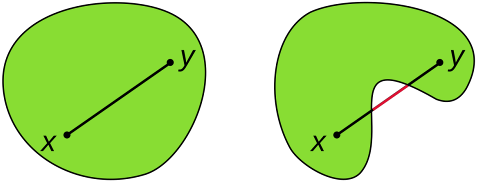

Figure 2: Convex set vs non-convex set. (Source)

Convexity - Convex Function

- **Convex function**. Equivalent definitions of convex functions:
  - For any continuous function $f : \text{dom}(f) \to \mathbb{R}$, it is convex if the area above the function is a convex set.
  - For any continuous function $f : \text{dom}(f) \to \mathbb{R}$, f is convex iff $\forall x, y \in \text{dom}(f), \forall \alpha \in [0, 1]$, we have $f(\alpha x + (1 - \alpha)y) \le \alpha f(x) + (1 - \alpha) f(y)$.
  - For any continuous function f with continuous first- and second-derivatives, it is convex iff its Hessian matrix $\nabla f^2(x) \succcurlyeq 0$

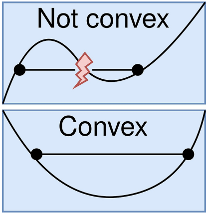

Figure 3: Convex function vs non-convex function. (Source)

Convexity

- Common convex functions:
  - linear functions, norms, and squared norms.
- Let $f(x)$ and $g(x)$ be convex; Convexity-preserving operations:
  - Non-negative scaling: $h(x) = \alpha f(x)$ is convex, where $\alpha \geq 0$.
  - Sum: $h(x) = f(x) + g(x)$ is convex.
  - Max: $h(x) = \max\{f(x), g(x)\}$ is convex.
  - Composition of a convex function and a linear function is convex.
    E.g., $h(x) = f(Ax)$, where A is a matrix (linear operator).
- However,
  - Multiplication of convex functions may NOT be convex.
    E.g., $f(x) = x$ and $g(x) = x^2$ are both convex,
    but $f(x)g(x) = x^3$ is not convex.
  - Composition of convex with convex may NOT be convex.
    E.g, $f(x) = x^2$ and $g(x) = (x-1)^2$ are both convex,
    but $g(f(x)) = (x^2 - 1)^2$ is not convex.

Matrix Computation with Python & Numpy

Let vectors $x, y \in \mathbb{R}^d$ and matrices $A, B \in \mathbb{R}^{n \times d}$.

- Matrix Transpose: `A.T`
- Element-wise product: `A * B`
- Element-wise square: `A ** 2`
- Matrix product: `A.T @ B`
- Solve linear equations $Ax = y$: `numpy.linalg.solve(A, y)`
- Indexing
  - Row vector: `A[0, :]` or `A[0]`
  - Column vector: `A[:, 0]`
  - Scalar: `A[0,1]`
- Identity matrix: I_n = `np.eye(n)`
- Matrix of zeros of shape `(2,3)`: `np.zeros((2,3))`
  Matrix of ones of shape `(2,3)`: `np.ones((2,3))`

Q&A

Thanks[1]

[1] Part of the content is credited to Zheng He (`zhhe@cs.ubc.ca`).