Pattern Recognition

Lecture 02. Review of Linear Algebra

Dr. Shanshan ZHAO & Dr. Hong Seng Gan

School of AI and Advanced Computing
Xi'an Jiaotong-Liverpool University

Academic Year 2023-2024

Table of Contents

What is a Linear Algebra?

Vector

A vector $x \in R_n$ is a list of n numbers, usually written as **column vector**.

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \tag{1}$$

The **unit vector** e_i is a vector of all 0's, except entry i, which has value 1: $e_i = (0, ...0, 1, 0, ...0)$, This is also called **one-hot vector**.

Properties

$$\mathrm{x} = (x_1, x_2) \quad \mathrm{y} = (y_1, y_2)$$
$$\mathrm{x} + \mathrm{y} = (x_1, x_2) + (y_1, y_2) = (x_1 + y_1, x_2 + y_2)$$
$$\mathrm{x} - \mathrm{y} = (x_1, x_2) - (y_1, y_2) = (x_1 - y_1, x_2 - y_2)$$
$$a\mathrm{x} = a(x_1, x_2) = (ax_1, ax_2)$$

Vector

create a vector with python

```
import numpy as np
# method with list
v1 = [1,2,3]
v1 = np.array([v1]).T
print(v1)
# method with numpy
v2 = np.array([[1],[2],[3]])
print(v2)
v = v1+v2
print(v)
print(3*v)
```

code/vector1.py

Matrices

A matrix $A \in \mathbb{R}^{m\times n}$ with m rows and n columns is a $2d$ array of numbers, arranged as follows: $\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{12} \\ a_{21} & a_{22} & \cdots & a_{12} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$

If $m = n$, the matrix is said to be **square**.

(a) Lena grayscale (b) Lena standard image

Basic operations-Create a matrix

2. create a matrix: $v = \begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix}$

```
# with the method of list
# TODO
M
print(M)
```

```
# with the method of array
# TODO
M =
print(M)
```

Basic operations-Create a matrix

```
# with the method of list
# TODO
M = [[1,4,7],[2,5,8],[3,6,9]]
M = np.array(M)
print(M)
```

```
# with the method of array
# TODO
M = np.array([[1,4,7],[2,5,8],[3,6,9]])
print(M)
```

Basic operations-matrix indexing

Task: get the result by indexing M:

$$v = \begin{bmatrix} 4 & 7 \\ 5 & 8 \end{bmatrix}$$

```
# TODO
```

Basic operations-matrix indexing

Task: get the result by indexing M:

$$v = \begin{bmatrix} 4 & 7 \\ 5 & 8 \end{bmatrix}$$

```
# TODO
print(M[0:2,1:3])
```

```
[[4 7]
 [5 8]]
```

Basic operations-other ways to create matrices

Many other functions for creating matrices/vectors provided Numpy

```
a = np.zeros((2,3)) # create an array with all zeros
print(a)
```

```
[[0. 0. 0.]
 [0. 0. 0.]]
```

```
b = np.ones((1,2)) # create an array with all ones
print(b)
```

```
c = np.full((2,4), 7) # create an array with constant values
print(c)
```

```
[[7 7 7 7]
 [7 7 7 7]]
```

```
d = np.eye(3) # create a 3 by 3 identity matrix
print(d)
```

```
[[1. 0. 0.]
 [0. 1. 0.]
 [0. 0. 1.]]
```

Vector-Vector Products

Given two vectors $x, y \in \mathscr{R}^n$, the quantity $x^T y$, sometimes called the ***inner product*** or ***dot product*** of the vectors, is a real number given by

$$x^T y \in \mathscr{R} = \begin{bmatrix} x_1, x_2, \cdots, x_n \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \sum_{i=1}^{n} x_i y_i$$

Note that it is always the case that $x^T y = y^T x$

Given vectors $x \in \mathscr{R}^m$,$y \in \mathscr{R}^n$, $xy^T \in \mathscr{R}^{m \times n}$ is called the ***outer product*** or ***cross product***. It is a matrix :

$$xy^T \in \mathscr{R}^{m \times n} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix} \begin{bmatrix} y_1, y_2, \cdots, y_n \end{bmatrix} = \begin{bmatrix} x_1 y_1 & x_1 y_2 & \cdots & x_1 y_n \\ x_2 y_1 & x_2 y_2 & \cdots & x_2 y_n \\ \vdots & \vdots & \ddots & \vdots \\ x_m y_1 & x_m y_2 & \cdots & x_m y_n \end{bmatrix}$$

Matrix-Vector Products

Given a matrix $A \in \mathscr{R}^n$ and a vector $x \in \mathscr{R}^n$, their product is a vector $y = Ax \in \mathscr{R}^m$. There are a couple ways of looking at matrix-vector multiplication.

If we write A by rows,then we can express Ax as,

$$y = Ax = \begin{bmatrix} - & a_1^T & - \\ - & a_2^T & - \\ & \vdots & \\ - & a_m^T & - \end{bmatrix} x = \begin{bmatrix} a_1^T x \\ a_2^T x \\ \vdots \\ a_m^T x \end{bmatrix}$$

If we write A by columns,then we can express Ax as,

$$y = Ax = \begin{bmatrix} | & | & & | \\ a_1 & a_2 & \cdots & a_n \\ | & | & & | \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} a_1 \end{bmatrix} x_1 + \begin{bmatrix} a_2 \end{bmatrix} x_2 + \cdots + \begin{bmatrix} a_n \end{bmatrix} x_n$$

y is a ***linear combination*** of the columns of A, where the coefficients of the linear combination are given by the entries of x.

Matrix-Matrix Products

1. We can view matrix-matrix multiplication as a set of vector-vector products. The most obvious viewpoint, which follows immediately from the definition, is that the (i, j)th entry of C is equal to the inner product of the ith row of A and the jth column of B.

$$C = AB = \begin{bmatrix} - & a_1^T & - \\ - & a_2^T & - \\ & \vdots & \\ - & a_m^T & - \end{bmatrix} \begin{bmatrix} | & | & & | \\ b_1 & b_2 & \cdots & b_n \\ | & | & & | \end{bmatrix}$$

$$= \begin{bmatrix} a_1^T b_1 & a_1^T b_2 & \cdots & a_1^T b_n \\ a_2^T b_1 & a_2^T b_2 & \cdots & a_2^T b_n \\ \vdots & \vdots & \ddots & \vdots \\ a_m^T b_1 & a_m^T b_2 & \cdots & a_m^T b_n \end{bmatrix}$$

Matrix-Matrix Products

2. We can represent A by columns, and B by rows. This representation leads to a much trickier interpretation of AB as a sum of outer products.

$$C = AB = \begin{bmatrix} | & | & & | \\ a_1 & a_2 & \cdots & a_n \\ | & | & & | \end{bmatrix} \begin{bmatrix} - & b_1^T & - \\ - & b_2^T & - \\ & \vdots & \\ - & b_n^T & - \end{bmatrix} = \sum_{i=1}^{n} a_i b_i^T$$

Basic operations: matrix multiplications

practise

2.4 Multiplications

2.4.1 Dot Multiplication

[ ]:
```
np.dot(3, 4)
```

$$M = \begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix} v = \begin{bmatrix} 2 \\ 4 \\ 6 \end{bmatrix}$$

Figure 1: Guess before running the cell

Properties of Matrices

- Matrix multiplication is associative: (AB)C = A(BC).
- Matrix multiplication is distributive: A(B+C)=AB+AC.
- Matrix multiplication is *not* commutative: $AB \neq BA$.
- Identity matrix, denoted $I \in \mathscr{R}^{n \times n}$:$AI = A = IA$
- $(A^T)^T = A$
- $(AB)^T = B^T A^T$
- $(A + B)^T = A^T + B^T$

Determinant

Useful value computed from the elements of a square matrix A

$$det\,[a_{11}] = a_{11}$$

$$det \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = a_{11}a_{22} - a_{12}a_{21}$$

$$det \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = a_{11}a_{22}a_{33} + a_{12}a_{23}a_{31} + a_{13}a_{21}a_{32}$$

$$- a_{13}a_{22}a_{31} - a_{23}a_{32}a_{11} - a_{33}a_{12}a_{21}$$

further reading:
`https://mathinsight.org/determinant_linear_transformation`

Matrix Inverse

Does not exist for all matrices, necessary (but not sufficient) that the matrix is square

$$AA^{-1} = A^{-1}A = I$$

And,

$$A^{-1} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}^{-1} = \frac{1}{detA} \begin{bmatrix} a_{22} & -a_{12} \\ -a_{21} & a_{11} \end{bmatrix}, detA \neq 0$$

If $detA = 0$, A does not have an inverse.

2.5.3 Matrix Inverse

```
from numpy.linalg import inv

a = np.array([[1., 2.], [3., 4.]])
ainv = inv(a)
```

```
print('a:',a)
print('inv:',ainv)
```

Eigenvectors and Eigenvalues

An eigenvalue λ and eigenvector x satisfies

$$Mx = \lambda x$$

Where A is a square matrix.
for scalar λ, which can be rewritten $(M - \lambda I)x = 0$ Which has a soluton if and only if $det(M - \lambda I) = 0$

- The eigenvalues are the roots of this determinant which is polynomial in λ.
- Substitute the resulting eigenvalues back into $Mx = \lambda x$ and solve to obtain the corresponding eigenvetor.

2.5.4 Eigenvalues and Eigenvectors

```
from numpy import linalg as LA
w, v = LA.eig(np.diag((1, 2, 3)))
```

The Gradient

Suppose that $f : \mathbb{R}^{m \times n} \to \mathbb{R}$ is a function that takes as input a matrix A of size $m \times n$ and returns a real value. Then the ***gradient*** of f (with respect to $A \in \mathbb{R}^{m \times n}$) is the matrix of partial derivatives, defined as:

$$\nabla_A f(A) \in \mathbb{R}^{m \times n} = \begin{bmatrix} \frac{\partial f(A)}{\partial A_{11}} & \frac{\partial f(A)}{\partial A_{12}} & \cdots & \frac{\partial f(A)}{\partial A_{1n}} \\ \frac{\partial f(A)}{\partial A_{21}} & \frac{\partial f(A)}{\partial A_{22}} & \cdots & \frac{\partial f(A)}{\partial A_{2n}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f(A)}{\partial A_{m1}} & \frac{\partial f(A)}{\partial A_{m2}} & \cdots & \frac{\partial f(A)}{\partial A_{mn}} \end{bmatrix}$$

i.e., an $m \times n$ matrix with $(\nabla_A f(A))_{ij} = \frac{\partial f(A)}{\partial A_{ij}}$

Note that the size of $\nabla_A f(A)$ is always the same as the size of A.

The Gradient

So if, in particular, A is just a vector $x \in \mathbb{R}^n$,

$$\nabla_x f(x) = \begin{bmatrix} \frac{\partial f(x)}{\partial x_1} \\ \frac{\partial f(x)}{\partial x_2} \\ \vdots \\ \frac{\partial f(x)}{\partial x_n} \end{bmatrix}$$

It is very important to remember that the gradient of a function is only defined if the function is real-valued, that is, if it returns a scalar value.

The Hessian

Suppose that $f : \mathbb{R}^n \to \mathbb{R}$ is a function that takes input a vector in $\mathbb{R}^n$ and returns a real value. Then the ***Hessian*** matrix with respect to x, written $\nabla_x^2 f(x)$ or simply as H is the $n \times n$ matrix of partial derivatives,

$$\nabla_x^2 f(x) \in \mathbb{R}^{n\times n} = \begin{bmatrix} \frac{\partial^2 f(x)}{\partial x_1^2} & \frac{\partial^2 f(x)}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f(x)}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f(x)}{\partial x_2 \partial x_1} & \frac{\partial^2 f(x)}{\partial x_2^2} & \cdots & \frac{\partial^2 f(x)}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f(x)}{\partial x_n \partial x_1} & \frac{\partial^2 f(x)}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f(x)}{\partial x_n^2} \end{bmatrix}$$

In other words, $\nabla_x^2 f(x) \in \mathbb{R}^{n\times n}$, with

$$(\nabla_x^2 f(x) \in \mathbb{R}^{n\times n})_{ij} = \frac{\partial^2 f(x)}{\partial x_i \partial x_j}$$

Note that the Hessian is always symmetric, since

$$\frac{\partial^2 f(x)}{\partial x_i \partial x_j} = \frac{\partial^2 f(x)}{\partial x_j \partial x_i}$$

Exercise 1

We have $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$, and $f(x) = x_1 x_2 + x_1^2 cos x_2$, compute $\nabla_x f(x)$.

$$\begin{bmatrix} \frac{\partial f(x)}{\partial x_1} \\ \frac{\partial f(x)}{\partial x_2} \end{bmatrix} = \begin{bmatrix} x_2 + 2x_1 \cos x_2 \\ x_1 - x_1^2 \sin x_2 \end{bmatrix}$$

Exercise 1 - Solution

$$\nabla f(x) = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \end{bmatrix} = \begin{bmatrix} x_2 + 2x_1 cos x_2 \\ x_1 - x_1^2 sin x_2 \end{bmatrix}$$

Exercise 2

$$(\nabla_x^2 f(x) \in \mathbb{R}^{n\times n})_{ij} = \frac{\partial^2 f(x)}{\partial x_i \partial x_j}$$

We have $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$, and $g(x) = x_1^2 + x_2^2$, compute $\nabla_x^2 g(x)$.

$$\nabla_x^2 g(x) = \begin{bmatrix} \frac{d^2 g(x)}{d x_1^2} & \frac{d^2 g(x)}{d x_1 \, d x_2} \\ \frac{d^2 g(x)}{d x_2 \, d x_1} & \frac{d^2 g(x)}{d x_2^2} \end{bmatrix}$$

$$\frac{d^2 g(x)}{d^2 x_1}$$

Exercise 2 - Solution

$$\nabla^2 g(x) = \begin{bmatrix} \frac{\partial^2 g}{\partial x_1^2} & \frac{\partial^2 g}{\partial x_1 \partial x_2} \\ \frac{\partial^2 g}{\partial x_2 \partial x_1} & \frac{\partial^2 g}{\partial x_2^2} \end{bmatrix} = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$

Since,

$$\frac{\partial g}{\partial x_1} = 2x_1, \frac{\partial g}{\partial x_2} = 2x_2 \checkmark$$

We have,

$$\frac{\partial^2 g}{\partial x_1^2} = \frac{\partial}{\partial x_1}\left(\frac{\partial g}{\partial x_1}\right) = \frac{\partial}{\partial x_1}(2x_1) = 2$$

$$\frac{\partial^2 g}{\partial x_2^2} = \frac{\partial}{\partial x_2}\left(\frac{\partial g}{\partial x_2}\right) = \frac{\partial}{\partial x_2}(2x_2) = 2$$

$$\frac{\partial^2 g}{\partial x_1 \partial x_2} = \frac{\partial}{\partial x_1}\left(\frac{\partial g}{\partial x_2}\right) = \frac{\partial}{\partial x_1}(2x_2) = 0$$

$$\frac{\partial^2 g}{\partial x_2 \partial x_1} = \frac{\partial}{\partial x_2}\left(\frac{\partial g}{\partial x_1}\right) = \frac{\partial}{\partial x_2}(2x_1) = 0$$

偏导数的运算法则：

1. 常数法则：如果 $f(x)$ 是常数，则其对任何自变量 x_i 的偏导数都是 0，即：$\frac{\partial f}{\partial x_i}=0$

2. 线性法则：如果 $f(x)$ 和 $g(x)$ 都是关于 x 的函数，且 a 和 b 是常数，则对于 $f(x)+g(x)$ 或者 $af(x)+bg(x)$，它们的偏导数分别等于 $f'(x)+g'(x)$ 和 $af'(x)+bg'(x)$，其中 $f'(x)$ 表示 $f(x)$ 对 x 的偏导数，$g'(x)$ 对 x 的偏导数。

3. 求积法则：如果有两个函数 $u(x)$ 和 $v(x)$，它们分别是关于 x 的函数，那么它们的乘积 $u(x)v(x)$ 对 x 的偏导为：

$$\frac{\partial[u(x)v(x)]}{\partial x}=\frac{u(x)\partial v(x)}{\partial x}+\frac{v(x)\partial u(x)}{\partial x}$$

4. 商法则：如果有两个函数 $u(x)$ 和 $v(x)$，它们分别是关于 x 的函数，那么它们的商 $\frac{u(x)}{v(x)}$ 对 x 的偏导为：

$$\frac{\partial\left[\frac{u(x)}{v(x)}\right]}{\partial x}=\frac{\frac{v(x)\partial u(x)}{\partial x}-\frac{u(x)\partial v(x)}{\partial x}}{[v(x)]^2}$$

5. 链式法则：链式法则用于计算复合函数的偏导数，如果有两个函数，$u(y)$ 和 $v(x)$，其中 y 是 x 的函数，即 $y=g(x)$，那么复合函数 $F(x)=u(g(x))$ 对 x 的偏导为：

$$\frac{\partial F}{\partial x}=\frac{\partial u}{\partial y}\cdot\frac{\partial y}{\partial x}$$

Thank You !

$\mathscr{Q}$ & $\mathscr{A}$

eg、设 $z = x^3y^2 - 3xy^3 - xy + 1$，求 $\frac{\partial^2 z}{\partial x^2}$、$\frac{\partial^2 z}{\partial y \partial x}$、$\frac{\partial^2 z}{\partial x \partial y}$、$\frac{\partial^2 z}{\partial y^2}$ 及 $\frac{\partial^3 z}{\partial x^3}$

解：$\frac{\partial z}{\partial x} = 3x^2y^2 - 3y^3 - y$ $\quad \frac{\partial z}{\partial y} = 2x^3y - 9xy^2 - x$

$$\frac{\partial^2 z}{\partial x^2} = \frac{\partial}{\partial x}\left(\frac{\partial z}{\partial x}\right) = \frac{\partial(3x^2y^2 - 3y^3 - y)}{\partial x} = 6xy^2 \checkmark$$

$$\frac{\partial^2 z}{\partial y \partial x} = \frac{\partial}{\partial y}\left(\frac{\partial z}{\partial x}\right) = \frac{\partial(3x^2y^2 - 3y^3 - y)}{\partial y} = 6x^2y - 9y^2 - 1 \checkmark$$

$$\frac{\partial^2 z}{\partial x \partial y} = \frac{\partial}{\partial x}\left(\frac{\partial z}{\partial y}\right) = \frac{\partial(2x^3y - 9xy^2 - x)}{\partial x} = 6x^2y - 9y^2 - 1 \checkmark$$

$$\frac{\partial^2 z}{\partial y^2} = \frac{\partial}{\partial y}\left(\frac{\partial z}{\partial y}\right) = \frac{\partial(2x^3y - 9xy^2 - x)}{\partial y} = 2x^3 - 18xy \checkmark$$

$$\frac{\partial^3 z}{\partial x^3} = \frac{\partial^2}{\partial x^2}\left(\frac{\partial z}{\partial x}\right) = \frac{\partial^2(3x^2y^2 - 3y^3 - y)}{\partial x^2} = \frac{\partial}{\partial x}\left[\frac{\partial(3x^2y^2 - 3y^3 - y)}{\partial x}\right] = \frac{\partial(6xy^2)}{\partial x} = 6y^2 \checkmark$$