The following list is based on my memory, it doesn't include everything to be expected but only what I remember

- Linear Algebra

  Matrix operations:

  - Element-wise addition: $\begin{bmatrix} a & b \\ c & d \end{bmatrix} \pm \begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} a \pm e & b \pm f \\ c \pm g & d \pm h \end{bmatrix}$
  - Multiplication: $\begin{bmatrix} a & b \\ c & d \end{bmatrix} \cdot \begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} ae + bg & af + bh \\ ce + dg & cf + dh \end{bmatrix}$
  - Is AB=BA? Try with A= $\begin{bmatrix} 1 & 2 \\ 7 & 4 \end{bmatrix}$ and B= $\begin{bmatrix} 4 & 1 \\ -2 & 0 \end{bmatrix}$
  - $\begin{bmatrix} \alpha & 0 \\ 0 & \alpha \end{bmatrix}$ can be used to scale a vector by α: $\begin{bmatrix} \alpha & 0 \\ 0 & \alpha \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \alpha x \\ \alpha y \end{bmatrix} = \alpha \begin{bmatrix} x \\ y \end{bmatrix}$

- Complex numbers (learn how to find the sign based on its location in the cartesian system)

  ### 1) Complex Numbers

  Example Question

  > **Given a complex number $z = -3 + 4i$, find:**
  > 1. **Its magnitude $|z|$.**
  > 2. **Its argument θ (the angle in radians).**
  > 3. **Determine which quadrant it lies in.**

  How to Solve

  1. **Magnitude:**

  $$|z| = \sqrt{(-3)^2 + (4)^2} = \sqrt{9 + 16} = \sqrt{25} = 5.$$

  2. **Argument:**

  $$\theta = \operatorname{atan2}(y, x) = \operatorname{atan2}(4, -3).$$

  Numerically, $\operatorname{atan2}(4, -3)$ is approximately 2.214 radians (about $126.87°$).

  3. **Quadrant:**
     - $x = -3 < 0, y = 4 > 0 \implies$ Quadrant II.

  Hence, $|z| = 5, \theta \approx 2.214$ radians, QII.

- matrix multiplication

  ### 2) Matrix Multiplication

  Example Question

  > **Let**
  > $$A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}, \quad B = \begin{pmatrix} 2 & 0 \\ 1 & -1 \end{pmatrix}.$$
  > **Compute $C = A \times B$.**

  How to Solve

  Use the standard formula

  $$C_{ij} = \sum_k A_{ik} B_{kj}.$$

  So, C is a 2×2 matrix:

  $$\begin{aligned} C_{11} &= A_{11}B_{11} + A_{12}B_{21} = (1)(2) + (2)(1) = 2 + 2 = 4, \\ C_{12} &= A_{11}B_{12} + A_{12}B_{22} = (1)(0) + (2)(-1) = 0 - 2 = -2, \\ C_{21} &= A_{21}B_{11} + A_{22}B_{21} = (3)(2) + (4)(1) = 6 + 4 = 10, \\ C_{22} &= A_{21}B_{12} + A_{22}B_{22} = (3)(0) + (4)(-1) = 0 - 4 = -4. \end{aligned}$$

  Hence,

  $$C = \begin{pmatrix} 4 & -2 \\ 10 & -4 \end{pmatrix}.$$

- Matrix Factorization

Factorization:

- $x^2 - y^2 = (x - y)(x + y)$
- $x^2 + y^2 + 2xy = (x + y)^2$

- recommendation system (given the three matrices, find the original)

17 秒の間、Thought about matrix factorization

以下給出一個**「已知三個矩陣，還原(重建)原始矩陣」**的範例，常見於推薦系統的奇異值分解(SVD)或其他矩陣分解場景。這裡示範的是 SVD 形式：

$$R = U\Sigma V^T$$

其中 U、Σ 和 V 三個矩陣已經事先求出，我們要根據它們來「找回」或「重建」原始的矩陣 R。

範例：SVD 重建

假設某個原始評分矩陣 R 在降維或分解後，得到以下三個矩陣 (U、Σ、V^T)，尺寸都為 2×2 以便示範：

1. U

$$U = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix}$$

2. Σ（奇異值對角矩陣）

$$\Sigma = \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix}$$

3. V^T

$$V^T = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix}$$

- **步驟 1：先計算 $M = U \times \Sigma$**

  我們先把 U 與 Σ 做矩陣相乘，得到一個中間矩陣 M。以 (2×2) 的情況為例，乘法公式是

  $$M_{ij} = \sum_{k=1}^{2} U_{ik}\, \Sigma_{kj}.$$

  用具體數值計算：

  $$U = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix}, \quad \Sigma = \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix}.$$

  - $M_{11} = \frac{1}{\sqrt{2}} \times 2 + \frac{1}{\sqrt{2}} \times 0 = \frac{2}{\sqrt{2}} = \sqrt{2}.$
  - $M_{12} = \frac{1}{\sqrt{2}} \times 0 + \frac{1}{\sqrt{2}} \times 1 = \frac{1}{\sqrt{2}}.$
  - $M_{21} = \frac{1}{\sqrt{2}} \times 2 + \left(-\frac{1}{\sqrt{2}}\right) \times 0 = \frac{2}{\sqrt{2}} = \sqrt{2}.$
  - $M_{22} = \frac{1}{\sqrt{2}} \times 0 + \left(-\frac{1}{\sqrt{2}}\right) \times 1 = -\frac{1}{\sqrt{2}}.$

  所以

  $$M = U\,\Sigma = \begin{pmatrix} \sqrt{2} & \frac{1}{\sqrt{2}} \\ \sqrt{2} & -\frac{1}{\sqrt{2}} \end{pmatrix}.$$

  $$\frac{\sqrt{2}}{2} \approx \frac{1.414}{2} \approx 0.717$$

  - $\frac{\sqrt{2}}{2}$ 已经是简化形式无需化解
  - $$\frac{2}{\sqrt{2}} = \frac{\sqrt{2}\times\sqrt{2}}{\sqrt{2}} = \frac{\sqrt{2}\times\cancel{\sqrt{2}}}{\cancel{\sqrt{2}}} = \sqrt{2}$$

步驟 2：再計算 $R = M \times V^T = U\,\Sigma\,V^T$

接著，把 M 和 V^T 相乘，就能得到重建後的 R。

$$V^T = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix}.$$

令 $R = M \times V^T$。同樣使用矩陣乘法公式：

- $R_{11} = M_{11}\,V^T_{11} + M_{12}\,V^T_{21}$
  $= \sqrt{2} \times \frac{1}{\sqrt{2}} + \frac{1}{\sqrt{2}} \times \frac{1}{\sqrt{2}} = 1 + \frac{1}{2} = 1.5.$
- $R_{12} = M_{11}\,V^T_{12} + M_{12}\,V^T_{22}$
  $= \sqrt{2} \times \frac{1}{\sqrt{2}} + \frac{1}{\sqrt{2}} \times \left(-\frac{1}{\sqrt{2}}\right) = 1 + \left(-\frac{1}{2}\right) = 0.5.$
- $R_{21} = M_{21}\,V^T_{11} + M_{22}\,V^T_{21}$
  $= \sqrt{2} \times \frac{1}{\sqrt{2}} + \left(-\frac{1}{\sqrt{2}}\right) \times \frac{1}{\sqrt{2}} = 1 + \left(-\frac{1}{2}\right) = 0.5.$
- $R_{22} = M_{21}\,V^T_{12} + M_{22}\,V^T_{22}$
  $= \sqrt{2} \times \frac{1}{\sqrt{2}} + \left(-\frac{1}{\sqrt{2}}\right) \times \left(-\frac{1}{\sqrt{2}}\right) = 1 + \frac{1}{2} = 1.5.$

於是重建出

$$R = \begin{pmatrix} 1.5 & 0.5 \\ 0.5 & 1.5 \end{pmatrix}.$$

這個 R 就是由三個已知矩陣 U、Σ、V^T 所重建（近似）回來的原矩陣。

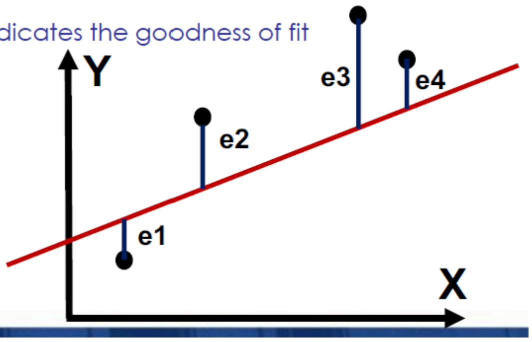
- R2

We will adopt a least square *loss function*

- A metric to minimize the overall error (sum of squared residuals)
  - $LSLF = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$ where y_i and $\hat{y}_i$ are the actual and estimated y-values at x_i, respectively.
- R-squared (Coef. of determination) indicates the goodness of fit

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y}_i)^2}$$

- R-squared lies between 0 and 1
  - $R^2 = 0$ means a very bad fit
  - $R^2 = 1$ means an excellent fit
- Other metrics: RMSE, MAE, MAPE



- ### 公式中的參數與符號含義

  1. y_i：真實值

  - 這是數據集中第 i 筆觀測值的真實結果。
  - 例子：在房價預測中，y_i 可能代表實際房價。

  ---

  2. $\hat{y}_i$：預測值

  - 模型對第 i 筆觀測值的預測結果。
  - 例子：在房價預測中，$\hat{y}_i$ 是模型輸出的預測房價。

- **3. $\bar{y}$：真實值的平均值**

  $$\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i,$$

  - n 是觀測值的總數。
  - $\bar{y}$ 表示數據集中所有真實值的平均值，代表「用單一值（平均值）進行預測時的基準」。

  ---

  4. SS_{res}：殘差平方和 (Sum of Squared Residuals)

  $$SS_{\text{res}} = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2.$$

  - 這是模型預測與真實值之間的「誤差平方和」。
  - $y_i - \hat{y}_i$ 是第 i 筆的預測誤差（或殘差），平方後加總表示整體的誤差大小。
  - 值越小，表示模型預測越接近真實值，擬合度越好。

- **5. SS_{tot}：總平方和 (Total Sum of Squares)**

  $$SS_{\text{tot}} = \sum_{i=1}^{n}(y_i - \bar{y})^2.$$

  - 這是所有真實值相對於平均值的變異總和。
  - 它表示「若我們只用平均值 $\bar{y}$ 來預測每個 y_i 時，總共會產生多少平方誤差」。
  - SS_{tot} 是衡量資料集中「總變異」的一個基準。

  公式中的意義與關係

  - $\frac{SS_{\text{res}}}{SS_{\text{tot}}}$：模型預測的誤差佔總變異的比例。
    - 如果這個比例接近 1，表示模型預測很差，幾乎沒有解釋數據的能力。
    - 如果這個比例接近 0，表示模型的預測與真實值非常接近，解釋了幾乎所有的變異。
  - $R^2 = 1 - \frac{SS_{\text{res}}}{SS_{\text{tot}}}$：模型解釋的變異佔總變異的比例。
    - 如果 $R^2 = 1$，表示模型完美擬合（預測值等於真實值，$SS_{\text{res}} = 0$）。
    - 如果 $R^2 = 0$，表示模型完全無法解釋數據（預測值等於 $\bar{y}$，$SS_{\text{res}} = SS_{\text{tot}}$）。

- Accuracy Rate

-
  - 橫軸：假陽性率 (FPR)，公式為：
    $$FPR = \frac{FP}{FP + TN}$$
  - 縱軸：真陽性率 (TPR)，公式為：
    $$TPR = \frac{TP}{TP + FN}$$
  - True Positive (TP)、False Positive (FP)、False Negative (FN) 和 True Negative (TN)
  - An example of a confusion matrix for disease diagnosis for 200 patients:

n = 200 patients		Actual Class	
		P	N
Predicted Class	P	**40**	15
	N	5	**140**

    A couple of definitions:
    - Accuracy: (TP+TN)/TOTAL = 180/200 = 90%
    - Misclassification error: (FP+FN)/TOTAL = 20/200 = 10%
- tf-idf
  - TF
    TF = The number of times a word appears in a document divided by the total number of words in the document
    - e.g.: when a 30-word document contains the term "student" 5 times, the TF for the word 'student' is 5/30=1/6.
  - IDF
    IDF = $log\left(\frac{N}{df_x}\right)$

    df_x: number of documents containing x

    N: total number of documents
    - e.g.: Let's assume the size of the corpus is 100 documents. If there are 20 documents that contain the term "student", then the IDF is:
      $log\left(\frac{100}{20}\right) = log5 = 0.70$

    Mathematically, TF-IDF ($W_{x,y}$) of a word x in a document y is obtained from:
    $$W_{x,y} = TF_{x,y} \times IDF_x$$

4) TF–IDF

Example Question

> Suppose we have 3 documents, and the term "cat" appears:
>
> - "2 times in Document 1 (which has 10 total words)"
> - "5 times in Document 2 (which has 20 total words)"
> - "0 times in Document 3 (which has 15 total words)"
>
> 1. Compute the term frequency (TF) for "cat" in Document 2 (use raw count / total words).
> 2. Compute the IDF for "cat" across these 3 documents (assume log base 10).
> 3. Compute TF–IDF for "cat" in Document 2.

How to Solve

1. TF for "cat" in Doc 2:

$$\mathrm{tf}(\text{"cat"}, \mathrm{Doc2}) = \frac{f(\text{"cat"}, \mathrm{Doc2})}{\text{total words in Doc2}} = \frac{5}{20} = 0.25.$$

2. IDF for "cat":
   - Number of documents, $N = 3$.
   - Documents containing "cat": Doc1 and Doc2 (2 documents).

$$\mathrm{idf}(\text{"cat"}) = \log_{10}\left(\frac{N}{\mathrm{df}(\text{"cat"})}\right) = \log_{10}\left(\frac{3}{2}\right) \approx \log_{10}(1.5) \approx 0.1761.$$

3. TF–IDF for "cat" in Doc 2:

$$\mathrm{tfidf}(\text{"cat"}, \mathrm{Doc2}) = \mathrm{tf}(\text{"cat"}, \mathrm{Doc2}) \times \mathrm{idf}(\text{"cat"}) = 0.25 \times 0.1761 \approx 0.0440.$$

- entropy

Decision Tree algorithms (Cont'd)

GEORGE BROWN COLLEGE

Which attribute at the root level would minimize the entropy of the data in the second level (remaining data belonging to one class as much as possible)?

- In simple terms, which attribute makes the remaining data belong to one class as much as possible?
  - Splitting over A: YY ZZ and YYY ZZ
  - Splitting over B: YYYYY and ZZZZ
  - Splitting over C: Y ZZZ and YYYY Z

Split on A		Split on B		Split on C	
A = 0 YY ZZ	A = 1 YYY ZZ	B = 0 YYYYY	B = 1 ZZZZ	C = 0 Y ZZZ	C = 1 YYYY Z

5) Entropy (Shannon)

Example Question

> A discrete random variable X can take values $\{a, b, c\}$ with probabilities
> $p(a) = 0.5,\ p(b) = 0.3,\ p(c) = 0.2.$
> Compute the Shannon entropy $H(X)$ in bits (log base 2).

How to Solve

Use the definition

$$H(X) = -\sum_i p(x_i)\,\log_2(p(x_i)).$$

So,

$$H(X) = -\Big[0.5\log_2(0.5)\ +\ 0.3\log_2(0.3)\ +\ 0.2\log_2(0.2)\Big].$$

One can compute each term numerically:

- $0.5\log_2(0.5) = 0.5 \times (-1) = -0.5$
- $0.3\log_2(0.3) \approx 0.3 \times (-1.737) \approx -0.5211$
- $0.2\log_2(0.2) \approx 0.2 \times (-2.322) \approx -0.4644$

Sum inside bracket: $-0.5 - 0.5211 - 0.4644 \approx -1.4855.$

Hence,

$$H(X) \approx -(-1.4855) = 1.4855\ \text{bits}.$$

- discrete equations manipulation (translation, shrink/expansion, flip)

Affine Transformation

An affine transformation is any transformation that preserves collinearity (all points lying on a line before the transformation will lie on a line after it) and ratios of distances (the midpoint of a line segment remains the midpoint after transformation).

In general, an affine transformation is a composition of rotations, translations, magnifications, and shears.

$$u = \boxed{c_{11}}x + c_{12}y + \boxed{c_{13}}$$
$$v = c_{21}x + \boxed{c_{22}}y + \boxed{c_{23}}$$

Magnifications

Translations (change in location)

The combination of all these non-zero parameters creates rotations and shears.

What is the difference between linear transformation and affine transformation?

A linear transformation fixes the origin, whereas an affine transformation need not do so. An affine transformation is the composition of a linear transformation with a translation, so while the linear part fixes the origin, the translation can map it somewhere else.

- 共線性：變換前在一條直線上的所有點，變換後仍會在同一條直線上。
- 比例性：線段的中點在變換後仍然是該線段的中點。

2. 仿射變換的公式

仿射變換的公式可寫成如下形式：

$$u = c_{11}x + c_{12}y + c_{13},$$

$$v = c_{21}x + c_{22}y + c_{23}.$$

其中：

- x, y：原始坐標。
- u, v：變換後的坐標。
- $c_{11}, c_{12}, c_{21}, c_{22}$：與旋轉、縮放（放大或縮小）、錯切有關的參數。
- c_{13}, c_{23}：與平移（Translation，改變位置）有關的參數。

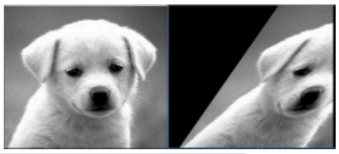
Affine Transformation

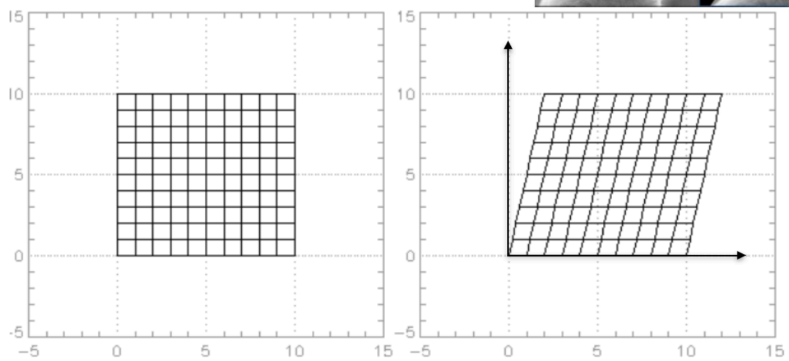
BROWN COLLEGE

A shear in the x direction shown in the below graph is produced by

$$u = x + 0.2y$$

$$v = y$$

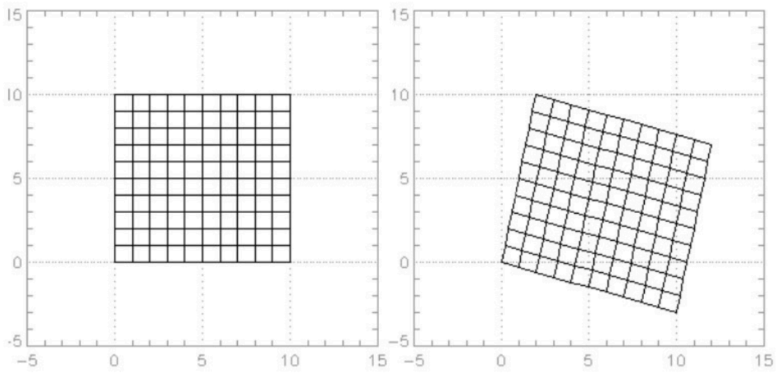




Affine transformation

This transformation produces both, a shear and a rotation.

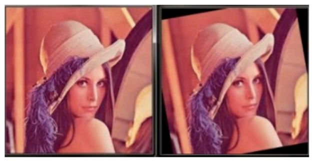
$$u = x + 0.2y$$
$$v = -0.3x + y$$

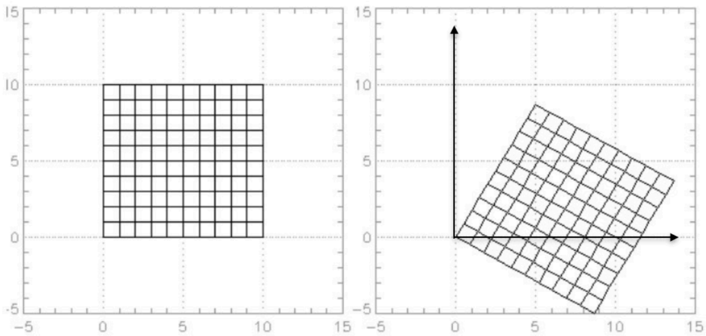


Affine Transformation

A rotation by θ is produced by

$$u = x\cos\theta + y\sin\theta$$
$$v = -x\sin\theta + y\cos\theta$$





Affine transformation

The transformation matrices can be used as building blocks.

How do transformation matrices for translation and scaling look like?

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & x_0 \\ 0 & 1 & y_0 \\ 0 & 0 & 1 \end{bmatrix} \quad \mathbf{T} = \begin{bmatrix} s_1 & 0 & 0 \\ 0 & s_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \mathbf{T} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Translation by (x_0, y_0) | Scaling by s_1 and s_2 | Rotating by θ

You will usually want to translate the center of the image to the origin of the coordinate system, do any rotations and scalings, and then translate it back.

- 平移 (Translation)

2. 三種基本變換矩陣

(a) 平移矩陣 (Translation Matrix)

公式：

$$T = \begin{pmatrix} 1 & 0 & x_0 \\ 0 & 1 & y_0 \\ 0 & 0 & 1 \end{pmatrix}.$$

- **作用**：將圖形在平面上移動。
- x_0：水平方向的平移量。
- y_0：垂直方向的平移量。
- **解釋**：這個矩陣中的 x_0 和 y_0 表示在 x 和 y 軸上的偏移量。當一個點 (x, y) 乘以這個矩陣時，點會被移動到新位置 $(x + x_0, y + y_0)$。

- 縮放 (Scaling)

(b) 縮放矩陣 (Scaling Matrix)

公式：

$$T = \begin{pmatrix} s_1 & 0 & 0 \\ 0 & s_2 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

- **作用**：將圖形在水平方向和垂直方向分別進行縮放（或放大）。
- s_1：水平縮放因子（大於 1 為放大，小於 1 為縮小）。
- s_2：垂直縮放因子。
- **解釋**：這個矩陣會將點 (x, y) 映射到 $(s_1 x, s_2 y)$，即按比例拉伸或壓縮。

- 旋轉 (Rotation)

(c) 旋轉矩陣 (Rotation Matrix)

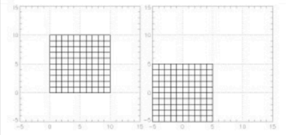
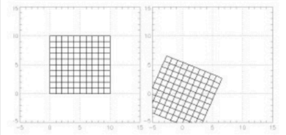
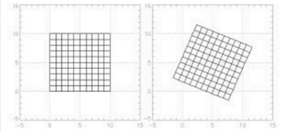
公式：

$$T = \begin{pmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

- **作用**：將圖形圍繞原點旋轉一個角度 θ（以弧度為單位）。
- $\cos\theta$、$\sin\theta$：決定旋轉後新位置的方向和距離。
- **解釋**：這個矩陣會將點 (x, y) 映射到一個繞原點旋轉 θ 度後的新位置。

- 仿射變換的使用順序
  - 圖片中的文字提到：
    - 通常，仿射變換需要按照一定的順序進行：
    - 將圖像的中心點移動到原點（用平移矩陣實現）。
    - 進行旋轉或縮放（用對應矩陣實現）。
    - 將圖像移回到原始位置（再次使用平移矩陣）。

Affine transformations

Operation	Expression	Result
Translate to Origin	$\mathbf{T}_1 = \begin{bmatrix} 1.00 & 0.00 & -5.00 \\ 0.00 & 1.00 & -5.00 \\ 0.00 & 0.00 & 1.00 \end{bmatrix}$	
Rotate by 23 degrees	$\mathbf{T}_2 = \begin{bmatrix} 0.92 & 0.39 & 0.00 \\ -0.39 & 0.92 & 0.00 \\ 0.00 & 0.00 & 1.00 \end{bmatrix}$	
Translate to original location	$\mathbf{T}_3 = \begin{bmatrix} 1.00 & 0.00 & 5.00 \\ 0.00 & 1.00 & 5.00 \\ 0.00 & 0.00 & 1.00 \end{bmatrix}$	

Affine transformation

Matrices for two-dimensional transformation in homogeneous coordinate:

1. Translation $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_x & t_y & 1 \end{bmatrix}$ or $\begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}$
2. Scaling $\begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$
3. Rotation (counter-clockwise) $\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$
4. Rotation (clockwise) $\begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$
5. Reflection against X axis $\begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
6. Reflection against Y axis $\begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
7. Reflection against origin $\begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
8. Reflection against line Y=X $\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
9. Reflection against Y= −X $\begin{bmatrix} 0 & -1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
10. Shearing in X direction $\begin{bmatrix} 1 & 0 & 0 \\ Sh_x & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
11. Shearing in Y direction $\begin{bmatrix} 1 & Sh_y & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
12. Shearing in both x and y direction $\begin{bmatrix} 1 & Sh_y & 0 \\ Sh_x & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

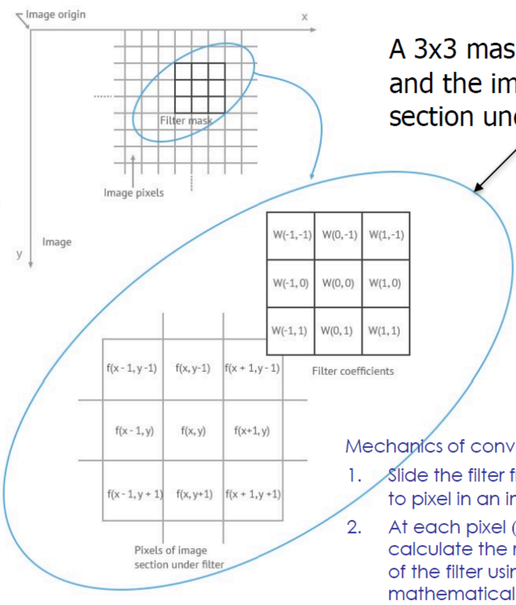
Convolution/spatial filtering

In image processing, a kernel, convolution matrix, filter, or mask is a small matrix used for blurring, sharpening, embossing, edge detection, and more. This is accomplished by doing a convolution between the kernel and an image.

Convolving a filter with an image results in extraction of features that help the computer detect an object in an image. Otherwise, by simply collapsing an image into a vector of its pixel intensities, these features would not be captured.

Convolution operations on images are widely used for image enhancement. Before we state the mathematical definition of convolution in the context of images, let's form its conceptual understanding.

Convolution is a misnomer, it actually is cross-correlation since the kernel is not flipped (reversed).



A 3x3 mask/filter and the image section under it

Mechanics of convolution:

1. Slide the filter from pixel to pixel in an image,
2. At each pixel (x, y), calculate the response of the filter using a mathematical formula.

6) Discrete Equations Manipulation (Translation, Shrink/Expansion, Flip)

Example Question

> **Consider a discrete‑time signal $x[n]$. Let**
>
> $$y_1[n] = x[n-2], \quad y_2[n] = x[2n], \quad y_3[n] = x[-n].$$
>
> **Describe the effect of each on the original signal.**

How to Solve

- **Translation (shift) $y_1[n] = x[n-2]$:**
  This shifts the signal in time by 2 units to the right.
- **Time scaling (shrink/expansion) $y_2[n] = x[2n]$:**
  This compresses the signal in time by a factor of 2 (i.e., it "plays faster").
- **Flip (time reversal) $y_3[n] = x[-n]$:**
  This mirrors the signal around $n = 0$ (a reversal in the time axis).

6) 離散方程操作 (平移、壓縮/擴展、翻轉)

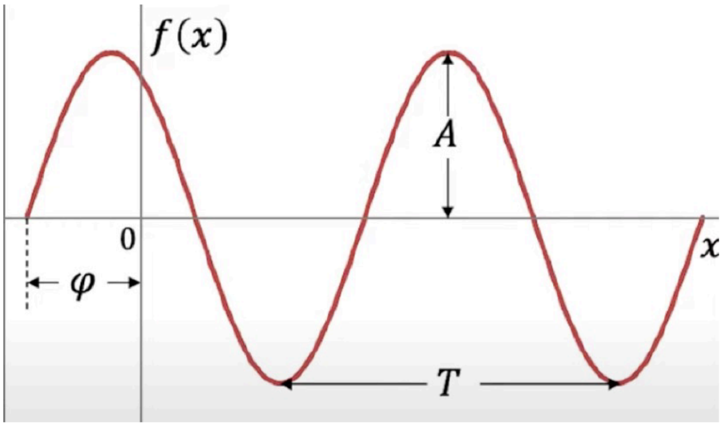
範例問題

> **考慮一個離散時間訊號 $x[n]$。定義：**
>
> $$y_1[n] = x[n-2], \quad y_2[n] = x[2n], \quad y_3[n] = x[-n].$$
>
> **說明各函數與原訊號之間的關係。**

解題步驟

- **平移 (Translation) $y_1[n] = x[n-2]$**：向右平移 2 單位。
- **時間縮放 (Shrink/Expansion) $y_2[n] = x[2n]$**：時間軸壓縮 2 倍（訊號被"加速"播放）。
- **翻轉 (Flip) $y_3[n] = x[-n]$**：以 $n = 0$ 為軸鏡射反轉。

$f(x) = A\sin(2\pi u x + \varphi)$



A: amplitude
u: ordinary frequency, the number of cycles per unit of time (Hertz) or distance
$\omega = 2\pi u$: angular frequency, radians per unit of time/distance
$T = 2\pi/(2\pi u) = 1/u$: period
φ: phase

- image transformation

7) Image Transformation (2D Geometric)

Example Question

> You have a point $(x, y) = (2, 3)$. Apply the following affine transform matrix:
>
> $$T = \begin{pmatrix} 1 & 0 & 4 \\ 0 & 1 & -2 \\ 0 & 0 & 1 \end{pmatrix}$$
>
> to find the new coordinates (x', y').

How to Solve

Write the point in homogeneous form $\begin{pmatrix} 2 \\ 3 \\ 1 \end{pmatrix}$, then multiply:

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 4 \\ 0 & 1 & -2 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 2 \\ 3 \\ 1 \end{pmatrix} = \begin{pmatrix} 1\cdot 2 + 0\cdot 3 + 4\cdot 1 \\ 0\cdot 2 + 1\cdot 3 + (-2)\cdot 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 2+4 \\ 3-2 \\ 1 \end{pmatrix} = \begin{pmatrix} 6 \\ 1 \\ 1 \end{pmatrix}.$$

So $(x', y') = (6, 1)$. This is effectively a translation by $(4, -2)$.

7) 影像幾何轉換 (Image Transformation)

範例問題

> **有一個平面座標點 $(x, y) = (2, 3)$，給定以下仿射轉換矩陣 T：**
>
> $$T = \begin{pmatrix} 1 & 0 & 4 \\ 0 & 1 & -2 \\ 0 & 0 & 1 \end{pmatrix}.$$
>
> **試求變換後的新座標 (x', y')。**

解題步驟

先將 $(2, 3)$ 寫成齊次座標 $\begin{pmatrix} 2 \\ 3 \\ 1 \end{pmatrix}$，然後相乘：

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 4 \\ 0 & 1 & -2 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 2 \\ 3 \\ 1 \end{pmatrix} = \begin{pmatrix} 2+4 \\ 3-2 \\ 1 \end{pmatrix} = \begin{pmatrix} 6 \\ 1 \\ 1 \end{pmatrix}.$$

故 $(x', y') = (6, 1)$。這代表在平面上做了 $(+4, -2)$ 的平移。

展開計算

具體點坐標為 $(x, y) = (2, 3)$。將其代入公式：

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 4 \\ 0 & 1 & -2 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 2 \\ 3 \\ 1 \end{pmatrix}.$$

按照矩陣乘法規則，逐項計算：

1. 第一行：

$$x' = 1 \cdot 2 + 0 \cdot 3 + 4 \cdot 1 = 2 + 4 = 6.$$

2. 第二行：

$$y' = 0 \cdot 2 + 1 \cdot 3 + (-2) \cdot 1 = 3 - 2 = 1.$$

3. 第三行：

$$1 = 0 \cdot 2 + 0 \cdot 3 + 1 \cdot 1 = 1.$$

- kernel filter calculation

8) Kernel Filter Calculation (Convolution)

Example Question

> Suppose an image patch I is:
>
> $$I = \begin{pmatrix} 2 & 3 & 1 \\ 0 & 1 & 5 \\ 2 & 2 & 2 \end{pmatrix},$$
>
> and you have a 3×3 kernel:
>
> $$K = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{pmatrix}.$$
>
> Calculate the convolution result at the center pixel (i.e., the output for the pixel at coordinates $(1, 1)$ within this 3×3 patch).

How to Solve

By definition of 2D convolution (or correlation if no flipping—this is a typical "mask multiply and sum" for filtering), at the central location $(x, y) = (1, 1)$:

$$O(1,1) \ = \sum_{i=-1}^{1} \sum_{j=-1}^{1} I(1+i,\ 1+j)\, K(i,j).$$

But for a small example, you can visually align the kernel with the image:

mathematica コピーする

```mathematica
Image I      Kernel K
 2  3  1      0  1  0
 0  1  5   *  1  1  1
 2  2  2      0  1  0
```

We multiply corresponding entries and sum:

$$\begin{aligned} &(2 \times 0) + (3 \times 1) + (1 \times 0) \\ &+ (0 \times 1) + (1 \times 1) + (5 \times 1) \\ &+ (2 \times 0) + (2 \times 1) + (2 \times 0). \end{aligned}$$

This becomes:

$$0 + 3 + 0 + 0 + 1 + 5 + 0 + 2 + 0 = 11.$$

Hence, the filtered (convolved) value at the center pixel is 11.

○

8) 核心濾波運算 (Kernel Filter Calculation)

範例問題

> **假設有一個 3×3 的影像區域（小視窗） I：**
>
> $$I = \begin{pmatrix} 2 & 3 & 1 \\ 0 & 1 & 5 \\ 2 & 2 & 2 \end{pmatrix},$$
>
> **以及一個 3×3 的濾波器（卷積核） K：**
>
> $$K = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{pmatrix}.$$
>
> **請計算該卷積核作用在此影像「中心像素」的卷積結果（亦可視為該中心位置的濾波輸出**值）**。**

解題步驟

在 2D 卷積（或相關操作）中，中心像素 $(x, y) = (1, 1)$ 的輸出可透過對應位置的對位乘積後相加：

對齊如下：(左邊是影像，右邊是濾波器)

mathematica　コピーする

```
I:          K:
 2  3  1     0  1  0
 0  1  5  *  1  1  1
 2  2  2     0  1  0
```

○

對應相乘再加總：

$$\begin{aligned} &(2 \times 0) + (3 \times 1) + (1 \times 0) \\ &+ (0 \times 1) + (1 \times 1) + (5 \times 1) \\ &+ (2 \times 0) + (2 \times 1) + (2 \times 0). \end{aligned}$$

計算得：

$$0 + 3 + 0 + 0 + 1 + 5 + 0 + 2 + 0 = 11.$$

因此該位置的輸出值為 11。

○

Convolution

Stride =1 pixel

1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

6×6 image

Dot product → 3   -1

1	-1	-1
-1	1	-1
-1	-1	1

Filter 1

- first one is 3, second one is -1

Smoothing filters

Given below is a general form of a 3×3 smoothing filter.

$$W = \begin{bmatrix} w(-1,-1) & w(-1,0) & w(-1,1) \\ w(0,-1) & w(0,0) & w(0,1) \\ w(1,-1) & w(1,0) & w(1,1) \end{bmatrix}$$

In general, a smoothing filter is a weighted averaging filter of size $m \times n$ (m and n are odd). The formula for filtering an $M \times N$ image with the weighted averaging filter is given by the expression

$$g(x,y) = \frac{\sum_{s=-a}^{s=a} \sum_{t=-b}^{t=b} f(x+s, y+t) w(s,t)}{\sum_{s=-a}^{s=a} \sum_{t=-b}^{t=b} w(s,t)} = f(x,y) * W. \quad (1)$$

The complete filtered image is obtained by applying equation (1) for $x = 0,1,2,\ldots,M\text{-}1$ and $y = 0,1,2,\ldots,N\text{-}1$. * denotes the operation of convolution.

What is the denominator equal to? Is it a constant?

2. 濾波器模板（Kernel 或 Mask）

圖中展示的是一個 3×3 濾波器：

$$W = \begin{pmatrix} w(-1,-1) & w(-1,0) & w(-1,1) \\ w(0,-1) & w(0,0) & w(0,1) \\ w(1,-1) & w(1,0) & w(1,1) \end{pmatrix}.$$

- 每個 $w(s,t)$ 是濾波器中的權重。
- 當濾波器作用於圖像時，將圖像中每個像素的鄰域與濾波器進行逐項相乘，然後求和得到新的像素值。

3. 加權平均濾波公式

圖中給出了加權平均濾波的公式：

$$g(x,y) = \frac{\sum_{s=-a}^{a} \sum_{t=-b}^{b} f(x+s,y+t)w(s,t)}{\sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t)},$$

其中：

- $g(x,y)$：濾波後的圖像在位置 (x,y) 的像素值。
- $f(x+s,y+t)$：原始圖像在位置 $(x+s,y+t)$ 的像素值。
- $w(s,t)$：濾波器對應位置的權重。
- a、b：分別是濾波器在水平方向和垂直方向的半徑，對於 3×3 濾波器，$a=b=1$。

Logarithmic transformations

Logarithmic transformation can be used to brighten the intensities in an image. It is used to increase the detail(contrast) of lower intensity values. They are especially useful for bringing out detail in Fourier transforms. The logarithmic transform of image $f(x, y)$ is:

$$g(x, y) = c \log(1 + f(x, y))$$

The constant c is typically used to scale the range of the log function to match the intensity range of the original image.

$$c = 255/\log(1 + 255)$$

It can also be used to further increase contrast. The higher the c, the brighter the image will appear.

Log transformation compresses the dynamic range of images with large variations in pixel values.

1. 對數變換的定義

對數變換是一種非線性操作，用於**增強低強度像素的細節**並壓縮圖像中高強度像素的動態範圍。這在某些圖像處理應用中非常有用，特別是在需要增強低亮度細節的情況下。

公式為：

$$g(x,y) = c \cdot \log(1 + f(x,y)),$$

其中：

- $f(x,y)$：原始圖像在像素位置 (x,y) 的灰度值（像素強度）。
- $g(x,y)$：對數變換後的圖像在像素位置 (x,y) 的灰度值。
- c：比例常數，用於調整變換後的圖像亮度範圍。

○

■ Logarithmic Transformation
- Log 的作用
- 壓縮動態範圍:
  - 當像素強度值範圍很大時(例如 0 到 255 或更高)，直接處理可能導致亮部過曝、暗部細節損失。
  - 對數函數的增長速度隨輸入值增大而變慢，因此可以壓縮高亮區域的值，讓圖像中的亮部和暗部同時清晰可見。
- 增強低亮度區域的細節:
  - 對數函數對小值的變化比較敏感，能更有效地突出暗部區域的差異，從而增強細節。

數字越大，Log 變換的結果越小

1. 對數函數的特性：
   - 對於 $y = \log(1+x)$，當 x 很大時，$\log(1+x)$ 的變化趨於平緩。
   - 例如：
     $$\log(1+10) = 2.397, \quad \log(1+100) = 4.615, \quad \log(1+1000) = 6.908.$$
     可以看到，輸入值增大 10 倍，但輸出值的增量逐漸減小。
   - 這種特性使得對數變換能壓縮亮部區域的動態範圍，避免過度曝光。

■

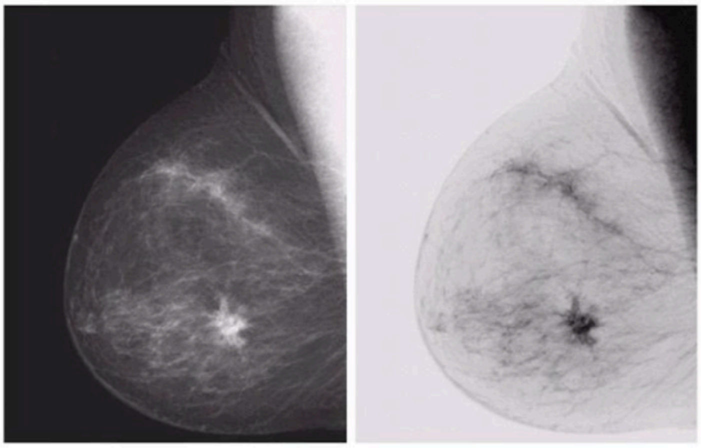
數字越小，Log 變換的步態（變化率）越大

1. 對數函數對小數值的響應：
   - 當 x 很小時，對數函數的增長速度較快。
   - 例如：
     $$\log(1+0.1) = 0.095, \quad \log(1+1) = 0.693, \quad \log(1+10) = 2.397.$$
     x 從 0.1 到 1 的增量 0.9 引起輸出值增加 0.598；而從 1 到 10 的增量 9 只引起輸出值增加 1.704。這表明對數變換在低強度區域更加敏感，能顯著增強暗部細節。

■

Photographic negative

Assume input pixel intensities are in range [0, L], where L=255.



(a) (b)

(a) Original digital mammogram $f(x, y)$,

(b) Negative image obtained using the negative transformation

$$g(x, y) = 255 - f(x, y)$$

1. 負片轉換的基本概念

負片轉換是一種將圖像的亮度值（像素強度）進行反轉的處理方法，用於創建圖像的「負片」。這種轉換的目的是突出細節或改變對比度，使圖像中的結構更加清晰。

公式如下：

$$g(x, y) = L - 1 - f(x, y),$$

或者當 $L = 256$ 時（對於 8 位灰度圖像）：

$$g(x, y) = 255 - f(x, y),$$

其中：

- $f(x, y)$：原始圖像在像素位置 (x, y) 的強度值。
- $g(x, y)$：負片圖像在像素位置 (x, y) 的強度值。
- L：像素值範圍的最大值（對於 8 位圖像，$L = 256$）。

2. 公式中的含義

- **強度值的反轉**：
  - 如果像素值 $f(x, y)$ 越大（越亮），則經過負片轉換後的像素值 $g(x, y)$ 越小（越暗）。
  - 例如：
    - $f(x, y) = 0$（全黑），$g(x, y) = 255$（全白）。
    - $f(x, y) = 255$（全白），$g(x, y) = 0$（全黑）。
- **目的**：
  - 反轉後的圖像可以讓人眼更容易觀察到暗區域的細節。
  - 常用於醫學影像處理（如 X 光片）或藝術效果。

○