壓縮：
複雜系統子模組間的運作

- PeterWolf

- 壓縮
- Kolmogorov Complexity 的起源與定義
- 核心概念與數學基礎
- 語言學研究裡的近似概念
- 實際應用與範例
- 在 NLP 與機器學習中的應用
- 理論限制與實務近似方法
- 未來發展方向

Put the universe into computation...

But our computer is too small!

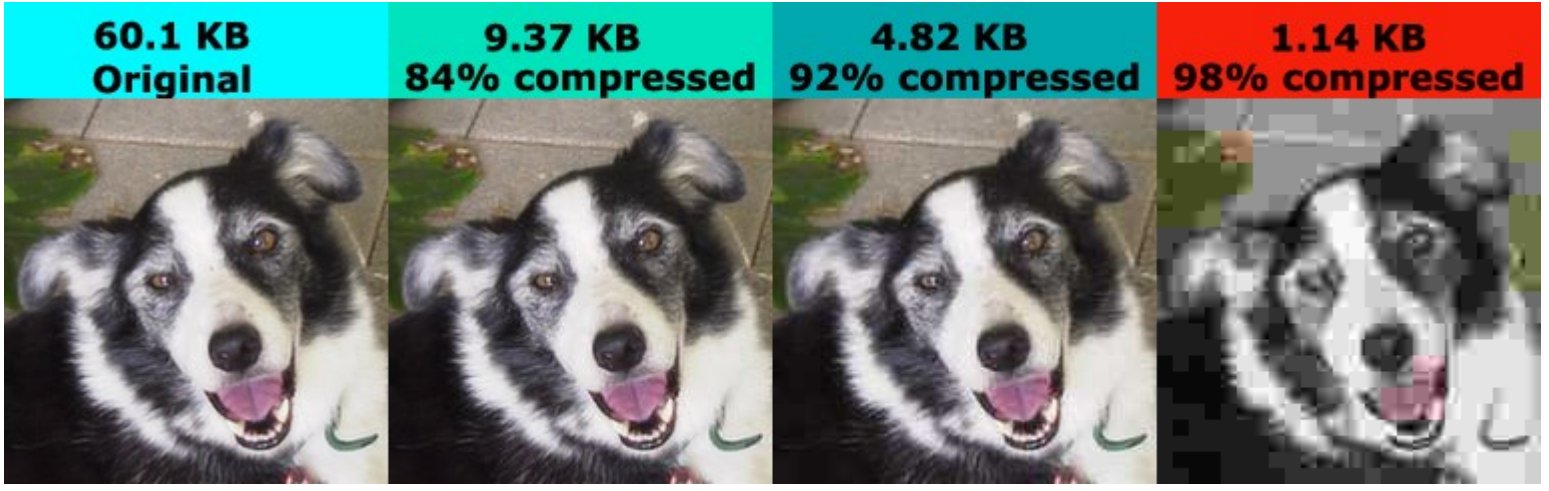
Binary compression

- 二元碼壓縮：二元碼縮短成　一個二元碼。

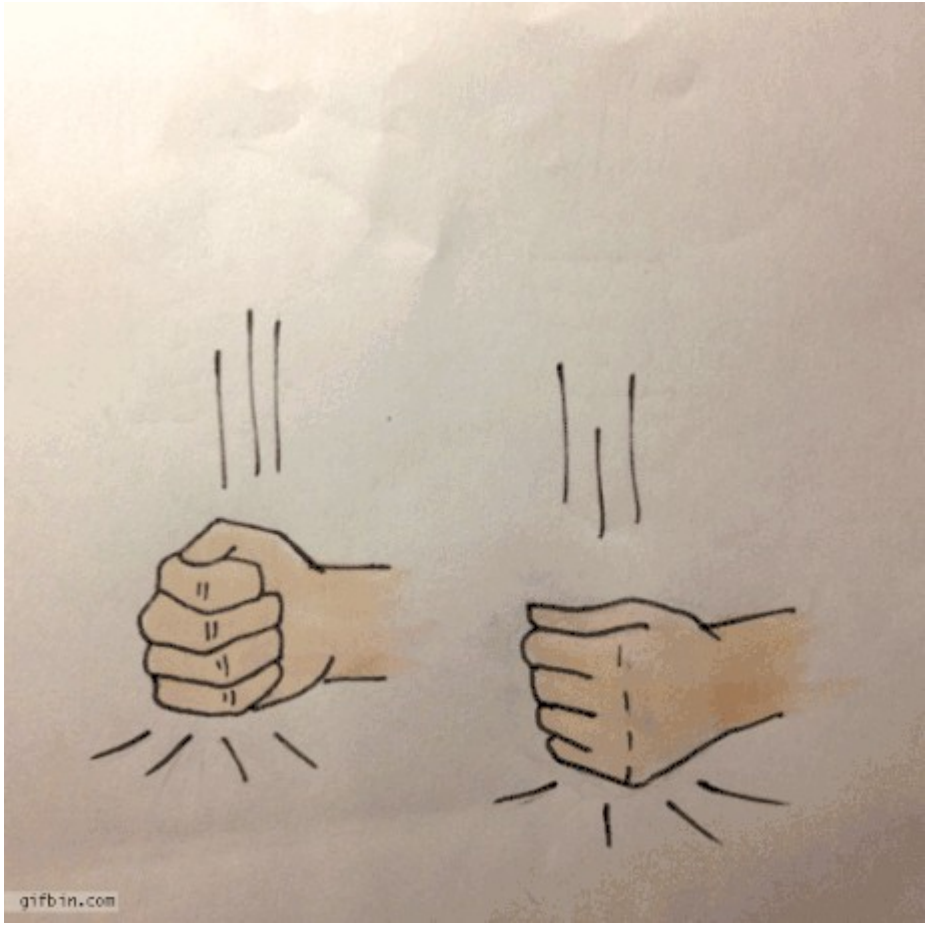
```
001001100101001 => [00: 000,  => 0101101101
                    01: 001,
                    10: 100,
                    11: 101]
```

- 15 碼縮成了 10 碼
  - 無損壓縮：沒有資訊無法還原
  - 表達力不足：有些可能原碼無法壓縮

https://bitmovin.com/wp-content/uploads/2020/03/LossyCompressionDoggo.jpeg



https://seanfitzpatrick1.wordpress.com/wp-content/uploads/2011/12/compression-dog-examples.jpg



因為人類會「自行腦補」，
因此對「失真」有一定程度的
容錯能力。

在「有損壓縮」（省記性）
和「無損壓縮」（存細節）
之間，演化的方向是…

Kolmogorov Complexity 的起源與定義

- 三位獨立發現者（1960年代）
  - Andrey Kolmogorov（蘇聯）
    - 莫斯科大學數學家
    - 提出「演算法複雜度」概念
  - Ray Solomonoff（美國）
    - 演算法機率理論：對 Occam's razor 敘述的數學化描述。
      - 在所有能夠完全描述的已觀測的可計算類中，較短的可計算理論在估計下一次觀測結果的機率時具有較大的權重。
    - 通用人工智慧先驅
  - Gregory Chaitin（美國-阿根廷，1966)
    - 演算法資訊理論
    - 提出 Chaitin's Omega：停機問題 (a.k.a.「算得完算不完」的問題)
    - cf. 上週的 c-command 實作程式

什麼是 Kolmogorov Complexity

- 核心定義
  - 一個字串的 Kolmogorov Complexity 是能夠生成該字串描述的最短程序描述。
  - 或可呈現為 [最短壓縮結果的長度和原長度的比值]
- 直觀理解
  - 簡單規律的字串 → 低複雜度
  - 隨機無序的字串 → 高複雜度
- 範例
  - "000000000000" → 低複雜度（可用「印 12 個 0」描述）
  - "a8f3k9m2x7q1" → 高複雜度（沒有簡短描述）

其實就是一種「壓縮比」！

Kolmogorov Complexity 的數學定義

- 對字串 X，其 Kolmogorov Complexity 為：
- K(x) = min { |p| : U(p) = x }
- 其中：
  - K(x) = 字串 x 的 Kolmogorov Complexity
  - p= 程序（program/process）
  - |p| = 程序 p 的長度（位元數）
  - U = 通用圖靈機（Universal Turing Machine）
  - U(p) = x = 程序 p 在 U 上執行後輸出 x

Kolmogorov Complexity 的簡單範例

- "AAAAAAAAAA" (10 個 A)
  - 直接表示： 10 個字符 （請用 Python 撰寫並計算程式碼長度）
  - 壓縮表示： "print('A' * 10)"（請用 Python 計算程式碼長度）
  - K("AAAAAAAAAA") ≒ ?

Kolmogorov Complexity 的不簡單範例

- "8k3mQ9xL2p"（10 個 A）
  - 直接表示： 10 個字符 （請用 Python 撰寫並計算程式碼長度）
  - 壓縮表示： "_____"（請用 Python 撰寫並計算程式碼長度）
  - K("8k3mQ9xL2p") ≒ ？

Kolmogorov Complexity 的關鍵性質

- 不可壓縮性：
  - 存在不可壓縮的字串
    - 對於長度為 n 的字串，至少有一個字串 x 使得 $K(x) \geq n$
    - 這樣的字串被稱為「隨機字串」
- 不變性定理：
  - 與程式語言選擇無關（有一個常數 C 的小差異而已）
  - $K\_Python(x) \approx K\_Java(x) \pm C$
  - 前式中的 C 是與語言相關的常數
- 上界性質
  - $K(x) \leq |x| + c$
  - 任何字串的複雜度不會遠超過其長度

Kolmogorov Complexity 為什麼重要

- 理論意義：
  - 資訊理論基礎
    - 定義「隨機性」的精確數學概念
    - 連結資訊論與計算理論
- 哲學意義：
  - 什麼是「簡單」？什麼是「複雜」？
  - 奧坎剃刀原理的形式化
- 實務影響
  - 數據壓縮
    - 理論上的壓縮極限
    - 評估壓縮演算法的效能

Kolmogorov Complexity 實務近似方法

- 使用壓縮演算法
  - 常用壓縮演算法
  - gzip / zlib
    - 基於 LZ77 + Huffman 編碼
    - 快速，適合一般用途
  - bzip2
    - 基於 Burrows-Wheeler 轉換
    - 壓縮率較高
  - LZMA / 7z
    - 更高的壓縮率
    - 計算成本較高

- 「近似」是什麼意思？ $K(x) \approx |\text{compressed}(x)|$

壓縮完以後，可以代表什麼？

- **標準化壓縮距離 (NCD)**
  - **測量兩個物件的相似度**
  - **定義：NCD(x, y) = [C(xy) - min(C(x), C(y))] / max(C(x), C(y))**
  - **其中：**
    - **C(x) = x 的壓縮長度**
    - **C(xy) = x 和 y 連接後的壓縮長度**
  - **特性**
    - **0 ≤ NCD ≤ 1**
    - **NCD = 0 → x 和 y 完全相同**
    - **NCD = 1 → x 和 y 完全不相關**
    - **對稱性： NCD(x,y) = NCD(y,x)**

應用範例

- 文本相似度分析
- 文本 1：＂機器學習是人工智慧的一個分支＂
- 文本 2：＂機器學習屬於人工智慧領域＂
- 文本 3：＂今天天氣很好＂

- 結果：
  - NCD(文本1，文本2) ≈ 0.3 ← 相較於3，文本1和2較相似
  - NCD(文本1，文本3) ≈ 0.9 ← 相較於2，文本1和3較不同

優弱勢分析

優點：

- **無需訓練：不需要預訓練模型**
- **語言無關：適用於任何語言**
- **通用性：可用於文本、圖像、音頻等**

缺點：

- **和語意無關，只是字串符號的比對**

在 NLP 中的應用

1. 文本分類
   - 使用 NCD 計算文本與各類別代表文本的距離
   - 將文本分配給距離最小的類別
   - 無需特徵工程

2. 語言檢測
   - 比較文本與不同語言語料庫的 NCD
   - 自動識別未知文本的語言

在 NLP 中的應用

3. 抄襲檢測
   - 計算文檔之間的 NCD
   - 高相似度可能表示抄襲

4. 文本摘要評估
   - 評估摘要保留了多少原文資訊
   - K(摘要 | 原文) 應該很小

在 NLP 中的應用

5. 異常檢測
   - 基於複雜度的異常檢測
     - 正常數據：低複雜度（和原有的常模一致，有規律）
     - 異常數據：高複雜度（隨機、不規律，和常模不同）
   - 方法
     - 計算數據集中　個樣本的 K(x)
     - 計算平均複雜度和標準差
     - 標記複雜度異常高的樣本
   - 應用場景
     - 網路入侵檢測 / 認知戰檢測
     - 工業設備異常監控
     - 醫療診斷輔助 (某個醫療意見和大家不一致)

生物資訊學和語言類型學的應用

6. DNA 序列分析 OR 句法相似度 / 構詞相似度分析
   - 基因序列複雜度
     - K(序列) 反映基因的資訊內容
     - 識別編碼區 vs 非編碼區 OR Complement vs. Adjunct
   - 物種分類 OR 語言類型分類
     - 基於基因組序列的壓縮距離比較
     - 無需序列逐段比對

限制與挑戰

理論限制

- **不可計算性**
  - **只能使用近似方法比較**
  - **不同壓縮器因壓縮演算法不同，而給出不同結果**

實務挑戰

- **短字串問題**
- **灰盒性質：我們知道它怎麼算，但還是很難解釋為什麼兩個東西被歸為「近似」**

作業

請回家操作本週課堂間講解的程式碼，利用各種「同語言但不同類型文本」、「不同語言但同類型」、「同語言也同類型」…等等文本進行操作與比較。

並在頻道裡和同學分享你的初步觀察。

本次作業是為了下一週的作業做準備，請務必動手操作！