

AI



自然語言處理

第 9 章 大語言模型

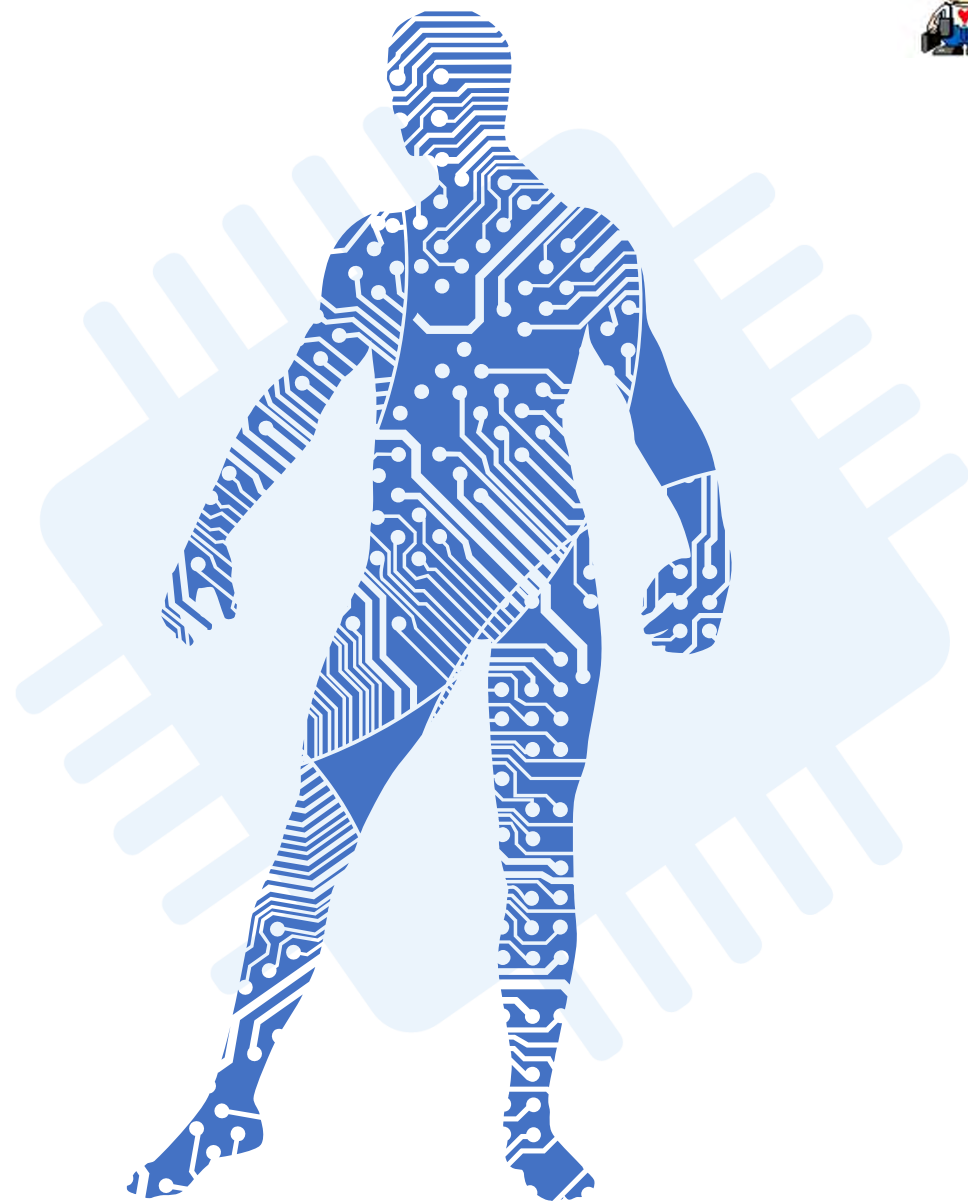
Large Language Models (LLM)

講師：紀俊男



本章大綱

- 大語言模型簡介
- Transformer 演算法
- 開源大語言模型：使用篇
- 開源大語言模型：程式篇
- 本章總結

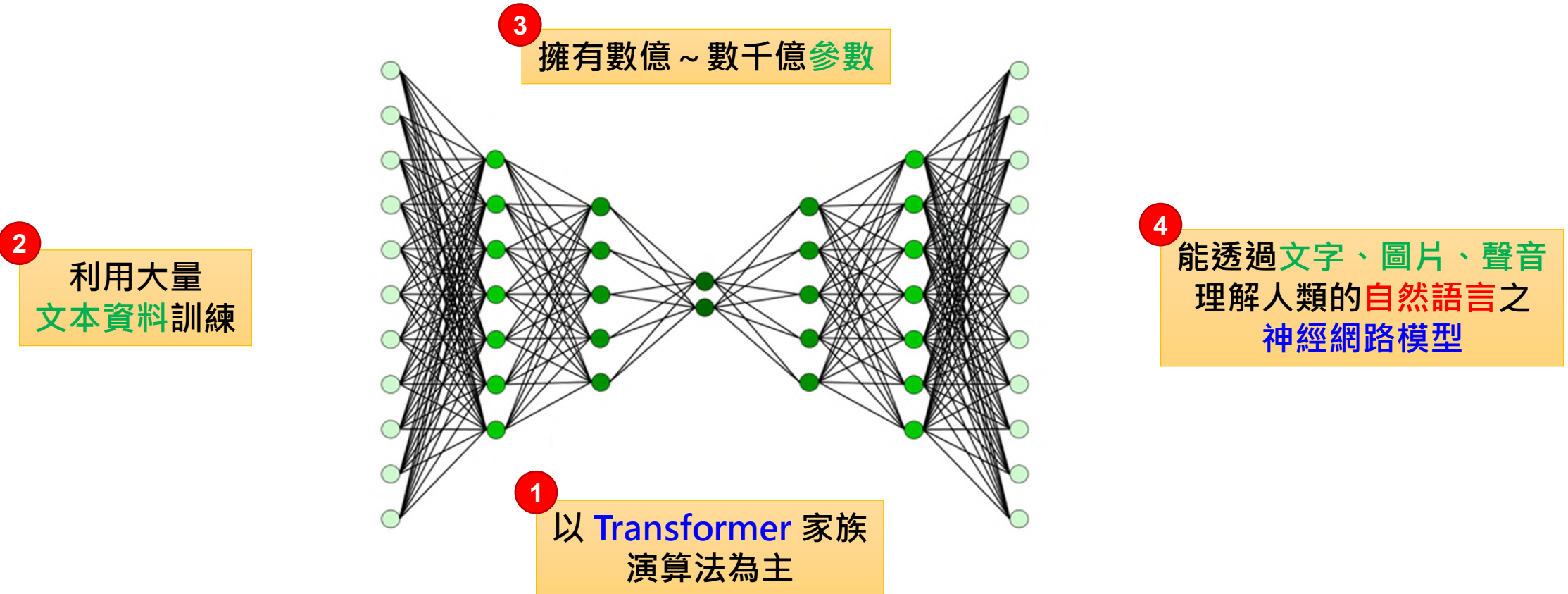




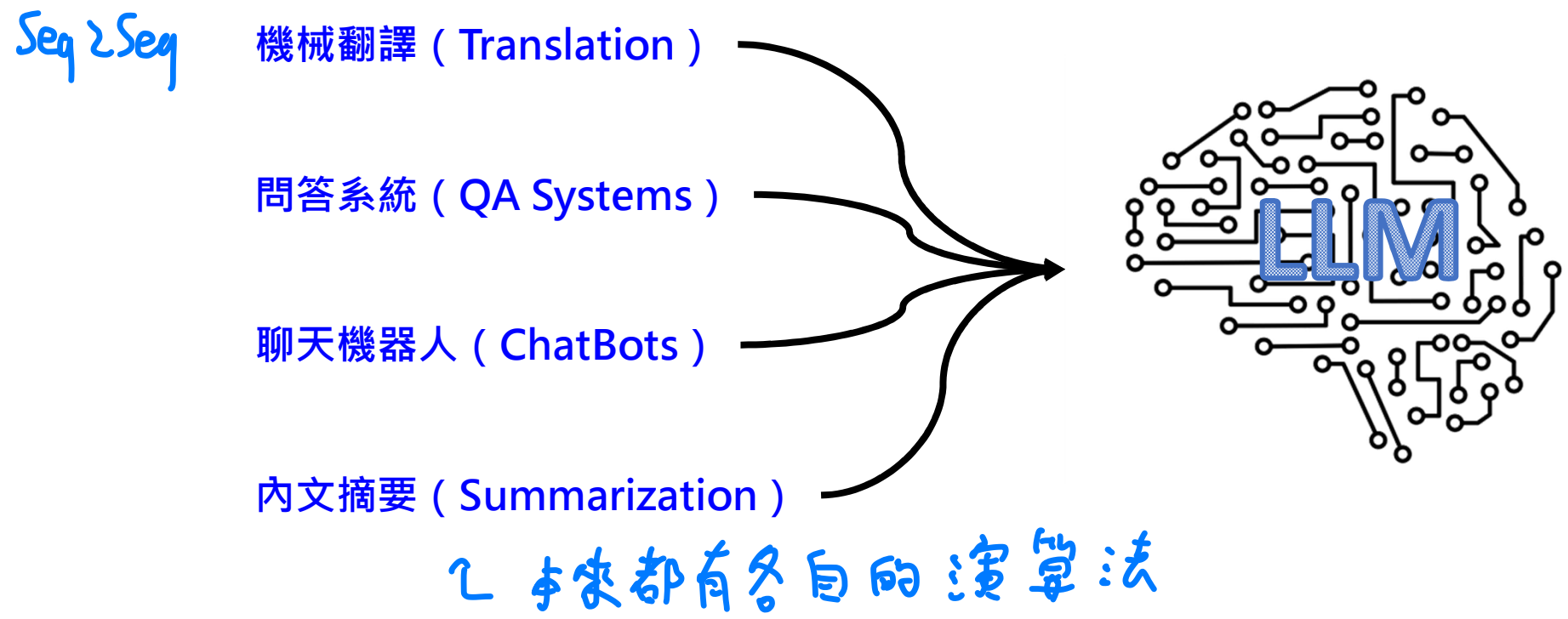
AI

大語言模型簡介

- Large Language Models (LLM)



• NLP 常見問題，一次解決！





Transformer 演算法

2017 Google

- 使用 RNN 做自然語言理解的難處
 - RNN 雖然可以參考 N 個時步內的文字，但沒辦法做到「長距離參考」

time step

0
「羅潔梅茵大人還沒有退燒嗎？」布麗姬娣問道。
斐迪南搖搖頭，先幫忙安排了讓侍從們返回神殿。
到了傍晚總算退燒，斐迪南摸了摸我的頭，說道：「嗯，現在應該沒問題了。」
59
於是我乘坐斐迪南的騎獸，再由達穆爾和布麗姬娣護在兩側，返回神殿。

時步 = 60
「我」是誰？

時步無法太長的原因

權重 $W = 1.1$ (時步 $n = 120$)
 $1.1^{120} = 92709.0688 \dots$

梯度爆炸！！

權重 $W = 0.9$ (時步 $n = 120$)
 $0.9^{120} = 0.00000323 \dots$

梯度消失～～

「長距離參考」解決方法之曙光



• 注意力機制 (Attention)

2024 Mamba $O(n \log n)$

Google, 2017

Attention Is All You Need

Ashish Vaswani*
Google Brain
avaswani@google.com

Noam Shazeer*
Google Brain
noam@google.com

Niki Parmar*
Google Research
nikip@google.com

Jakob Uszkoreit*
Google Research
usz@google.com

Llion Jones*
Google Research
llion@google.com

Aidan N. Gomez*[†]
University of Toronto
aidan@cs.toronto.edu

Lukasz Kaiser*
Google Brain
lukaszkaizer@google.com

Illia Polosukhin*[‡]
illia.polosukhin@gmail.com

指揮艙
從
登月艇
拿到
樣本
後
它
即將
返航



0.84

0.11

0.05

指揮艙
從
登月艇
拿到
樣本
後
它
即將
返航

更新「注意力分數」

$O(n^2)$

$n = \text{上下文窗口}$

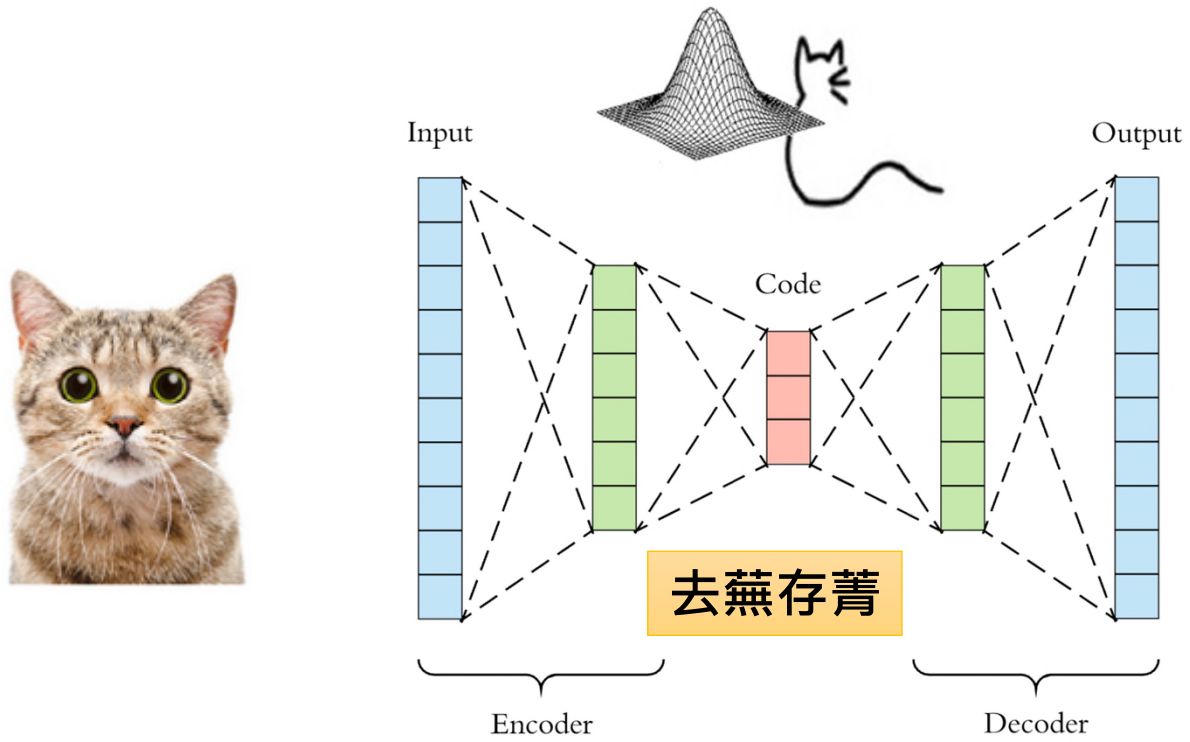
它 = 指揮艙

少量文字 → 理解意義



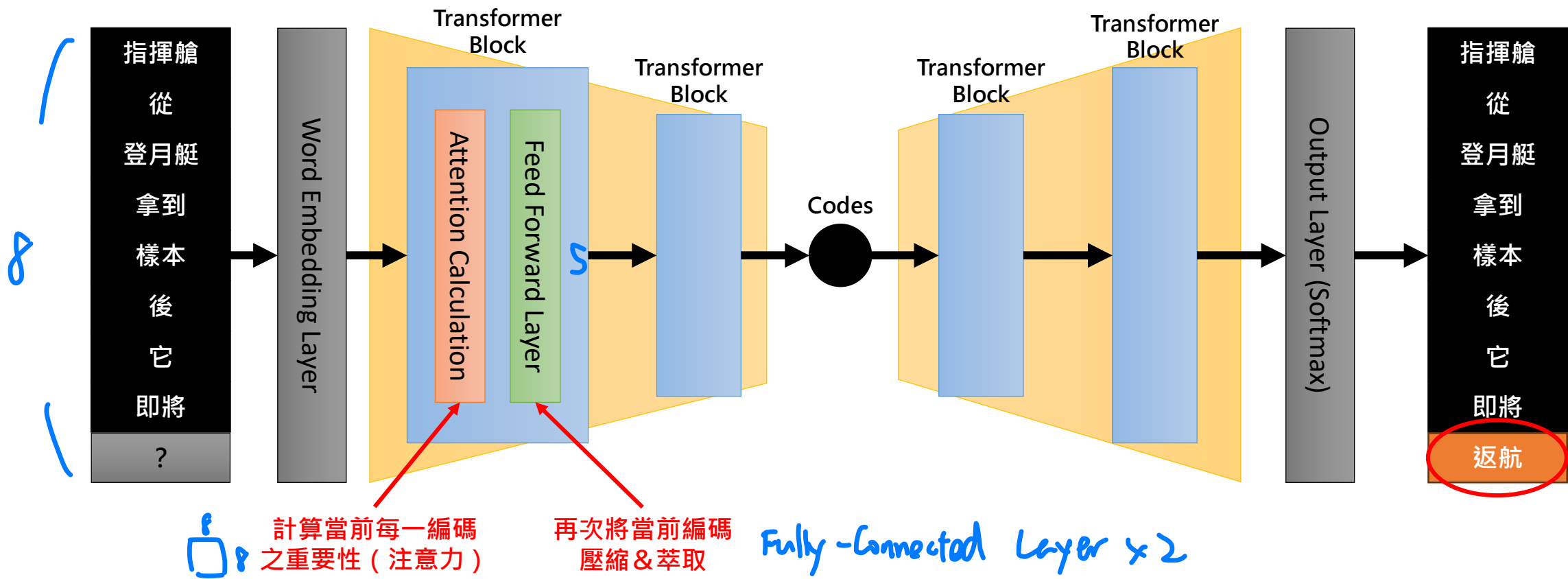
理解「注意力機制」最佳影片：<https://ishort.ink/7U3Z>

- 理解文字之後，能否「說」出相應的句子呢？
 - Autoencoder：濃縮語意，產生文字的神經網路

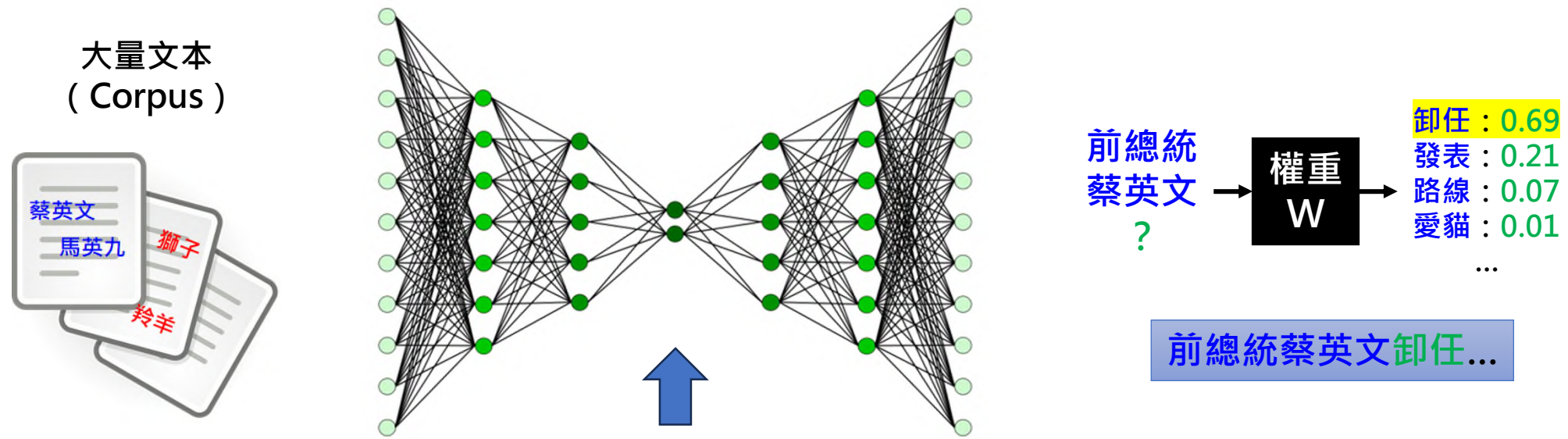


與原本不完全相同

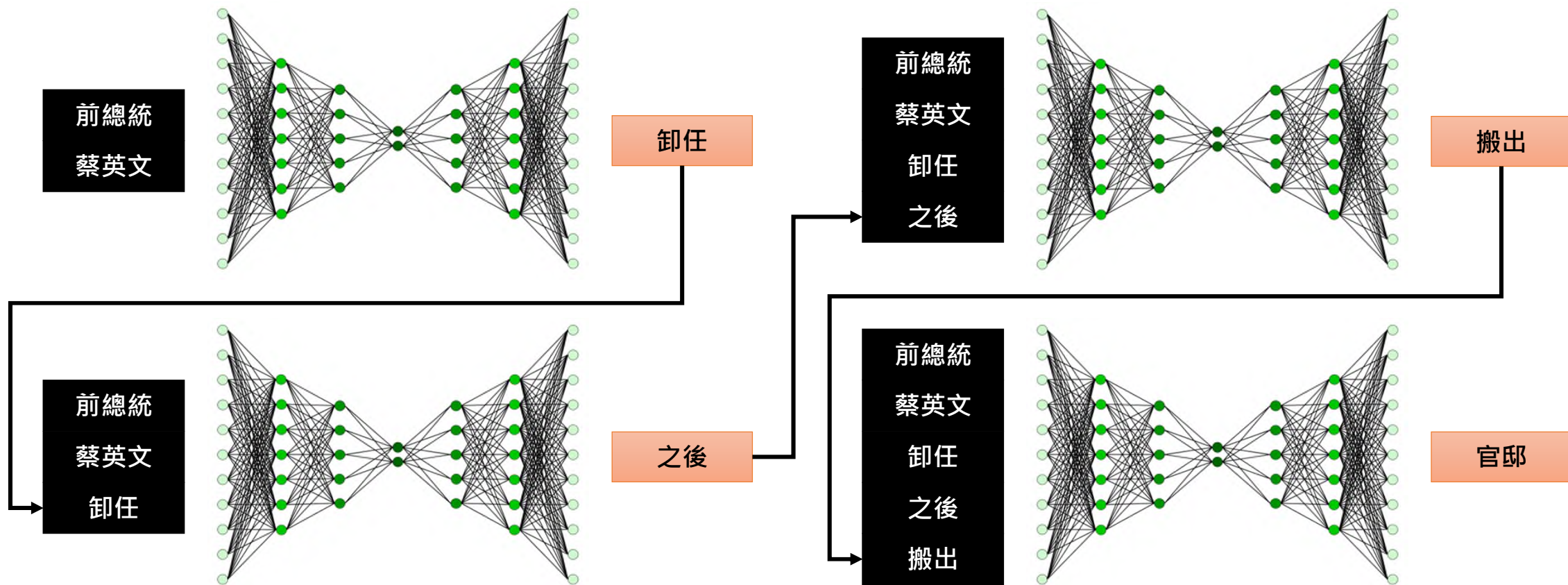
Transformer = 多層次 Transformer Blocks 之 Autoencoder



Transformer 模型如何訓練？



蔡英文卸任後，新北小英之友會續辦公益捐血
蔡英文發表國慶談話
總統賴清德520就職演說，未脫離前總統蔡英文路線



前總統蔡英文卸任之後搬出官邸...

常見的 Transformer 家族演算法



修改 BERT

模型	年份	單位	優點	缺點
BERT	2018	Google	雙向上下文、預訓練和微調、廣泛應用	模型龐大、序列長度限制
GPT	2018	OpenAI	生成能力強、單向上下文、靈活性高	單向上下文限制、資源消耗大
T5	2019	Google	統一框架、靈活性高、性能優越	訓練複雜、資源需求高
RoBERTa	2019	Meta	性能提升、簡化模型	資源消耗大、序列長度限制
XLNet	2019	Google & CMU	雙向上下文、性能優越	訓練複雜、模型龐大

值得參考的影音教學：

- 李宏毅：[淺談 Transformer](#)
- 李宏毅：[淺談大語言模型的可解釋性](#)
- The AI Hacker: [Illustrated Guide to Transformers Neural Network](#)
- 3Blue1Brown: [But what is a GPT? Visual intro to transformers](#)
- Datafuse Analytics: [Confused which Transformer Architecture to use? BERT, GPT-3, T5, Chat GPT? Encoder Decoder Explained](#) (Transformer 家族介紹)

Transformer
BERT
GPT

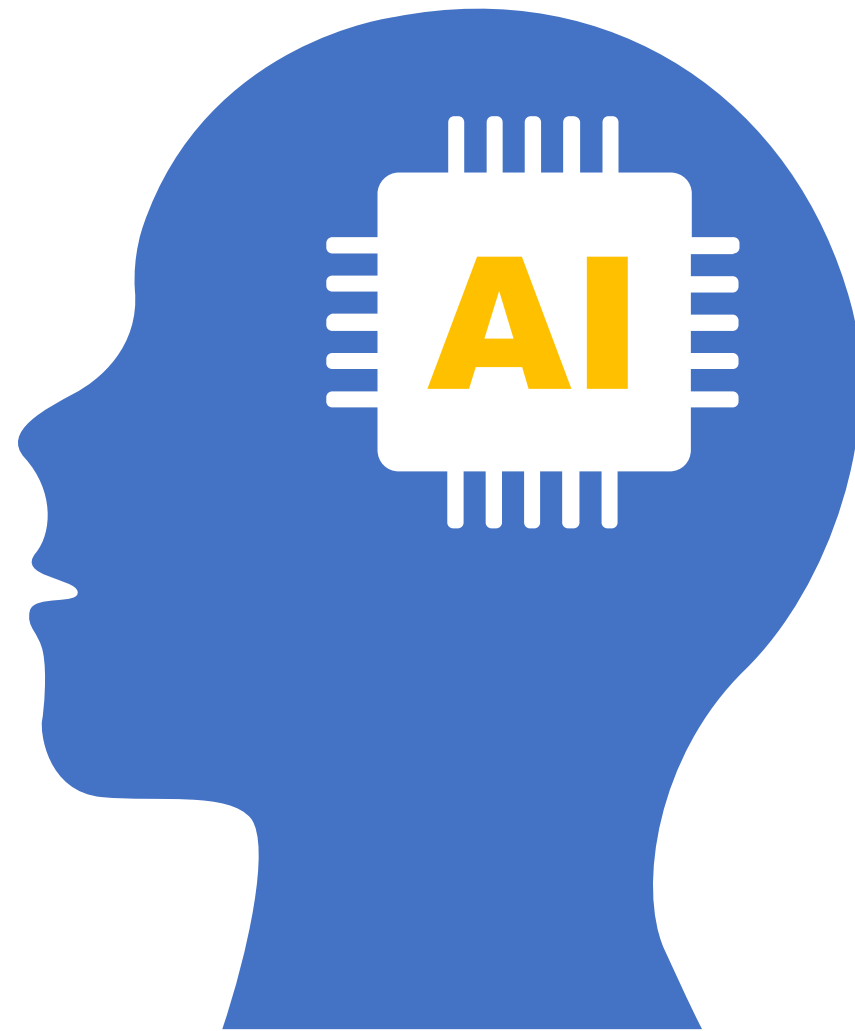


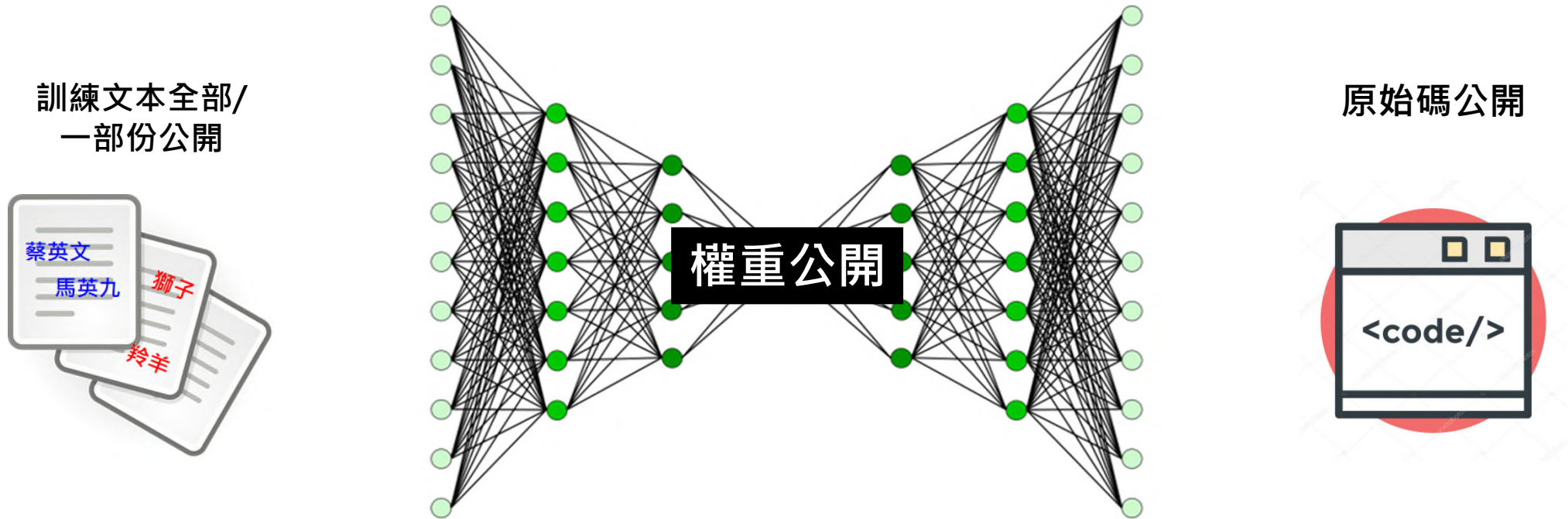
開源大語言模型： 使用篇



使用開源大語言模型

- 簡介
- 使用 LM Studio
- 使用 Ollama





- 優點

- 可本地端執行
 - 於本地端執行大語言模型。
 - 免費享用先進技術。
- 透明可學習
 - 一切權重、程式碼公開。
 - 可以學習 LLM 是如何做出來的。
- 快速迭代與創新
 - 全球人員共同開發。
 - 可能碰撞出大公司沒有的功能。
- 客製化 (re-trainable)
 - 可以投入自己的文本。
 - 將 LLM 客製化成特定領域的專家。


- 缺點

- 需要強悍的硬體
 - 推薦 SSD 硬碟、32GB 記憶體。
 - 顯卡推薦 8 ~ 16GB VRAM。
- 安全隱私風險
 - 程式碼龐大。
 - 無從得知是否藏了有害內容。
- 版權與商用問題
 - 某些 LLM 盜用他人資源。
 - 下載使用可能有版權疑慮。
 - 某些 LLM 會註明不可商用。

模型	參數	單位	優點	缺點
Gemma	2B / 7B	Google	<ul style="list-style-type: none">輕量高效，筆電也能跑。	<ul style="list-style-type: none">有時會答非所問。
Llama 3	8B / 70B	Meta AI	<ul style="list-style-type: none">性能強大，適合客製化。	<ul style="list-style-type: none">使用許可有限制。
Phi	1.3B / 2.7B	Microsoft	<ul style="list-style-type: none">緊湊高效，就算單板機電腦也能跑。	<ul style="list-style-type: none">較難客製化，有時會答非所問。
Mistral	7B / 8x22B	Mistral AI	<ul style="list-style-type: none">參數大小與回應品質最佳平衡。支援高達 16000 個 Token 上下文。	<ul style="list-style-type: none">與其它相比，需要較高硬體能耗。對於中文支援比較沒那麼好。
特殊的大語言模型				
LLaVa	7B / 34B	Xtuner	<ul style="list-style-type: none">有「視覺」能力的大語言模型。丟照片給它，會描述照片內容。	<ul style="list-style-type: none">描述照片的精準度普通。與 GPT4 差不多，但不如 GPT4o
TAIDE	7B / 8B	台灣國科會	<ul style="list-style-type: none">以 Llama 2 7B 為基礎重新訓練。多繁體中文支援良好。	<ul style="list-style-type: none">用 Llama 2 模型改造而來。某些回應品質還是沒有很好。
Taiwan LLM	8B / 70B	Yen-Ting Lin	<ul style="list-style-type: none">台灣網友以 Llama 3 為基礎訓練。硬體性能好者，可以直上 70B。	<ul style="list-style-type: none">70B 性能雖好，但須頂規硬體。
NSFW_13B	13B	中國網友	<ul style="list-style-type: none">移除大語言模型道德限制的版本。	<ul style="list-style-type: none">若擔心資安風險，請勿下載。



Hugging Face
<https://huggingface.co/>

 **Hugging Face**

Models

Datasets


Spaces

Posts

Docs

Pricing

⌵



Tasks

Libraries

Datasets

Languages

Licenses

Other

Multimodal

Image-Text-to-Text

Visual Question Answering

Document Question Answering

Computer Vision

Depth Estimation

Image Classification

Object Detection

Image Segmentation

Text-to-Image

Image-to-Text

Image-to-Image

Image-to-Video

Unconditional Image Generation

Video Classification

Text-to-Video

Zero-Shot Image Classification

Mask Generation

Zero-Shot Object Detection

Text-to-3D

Image-to-3D

Image Feature Extraction

Models 675,835

Full-text search

Sort: Trending

openbmb/MiniCPM-Llama3-V-2_5

Visual Question Answering • Updated about 1 hour ago • 15.5k • 569

microsoft/Phi-3-vision-128k-instruct

Text Generation • Updated 2 days ago • 14.1k • 483

mistralai/Mistral-7B-Instruct-v0.3

Text Generation • Updated 4 days ago • 21.7k • 399

meta-llama/Meta-Llama-3-8B

Text Generation • Updated 13 days ago • 956k • 4.2k

microsoft/Phi-3-medium-128k-instruct

Text Generation • Updated 4 days ago • 20.3k • 235

CohereForAI/aya-23-8B

Text Generation • Updated about 16 hours ago • 7.23k • 156

mistralai/Mistral-7B-v0.3

Text Generation • Updated 4 days ago • 21.8k • 151

模型分類

模型列表



開源大語言模型命名習慣



Fine-Tuning

公司、個人
名稱

模型名稱

基礎模型
名稱

參數規模

參數量

客製化方式
(微調方式)

參數量化方法
(壓縮模型之法)

檔案格式

microsoft/cogvlm2-llama3-large-70b-instruct-Q4_K_M_S-gguf

- 公司、個人名稱：就是 Hugging Face 的帳號名稱。
- 模型名稱：可自由命名的模型名稱。想取什麼名字都可以。
- 基礎模型名稱：若此模型是從其它模型修改而來，就會附上基礎模型名稱。
- 參數規模：mini - 百 ~ 千萬、small - 千萬 ~ 億、medium - 十億、large - 百億。
- 參數量：2B = 20 億、70B = 七百亿。
- 微調方式：以新資料集對基礎模型再訓練。此處標籤很多很雜。
 - -instruct, -it：用各種指令（如：「幫我摘要這篇文章」）再訓練、特化過。 *e.g. QA*
 - -chat：餵過很多「閒聊」之文本再訓練過。 *e.g. PPT, Ocard*
 - -uncensored：移除倫理限制，且用暴力、情色...文本再訓練過。
- 量化方法：Quantization。一種縮小模型的方法。
 - Q4 = 將 32 bits 浮點數壓縮到 4 bits。另有 Q3, Q8...等。
 - K = 使用 K-Means 集群演算法 (Clustering) 將權重分堆，並用中心值取代同群所有權重。
 - M = Mixture。混合了多種浮點數壓縮法。如 Q4 + Q8。
 - S = Sparse。稀疏矩陣之意。將接近零的權重，全都當成零。
- 檔案格式：
 - gguf = General Graphical UI Format。自帶圖形使用者介面的模型，而非僅含一堆函數。
 - hf = Hugging Face。符合 Hugging Face 網頁介面標準，可以在 Hugging Face Live Demo 的模型。



隨堂練習：尋找開源大語言模型



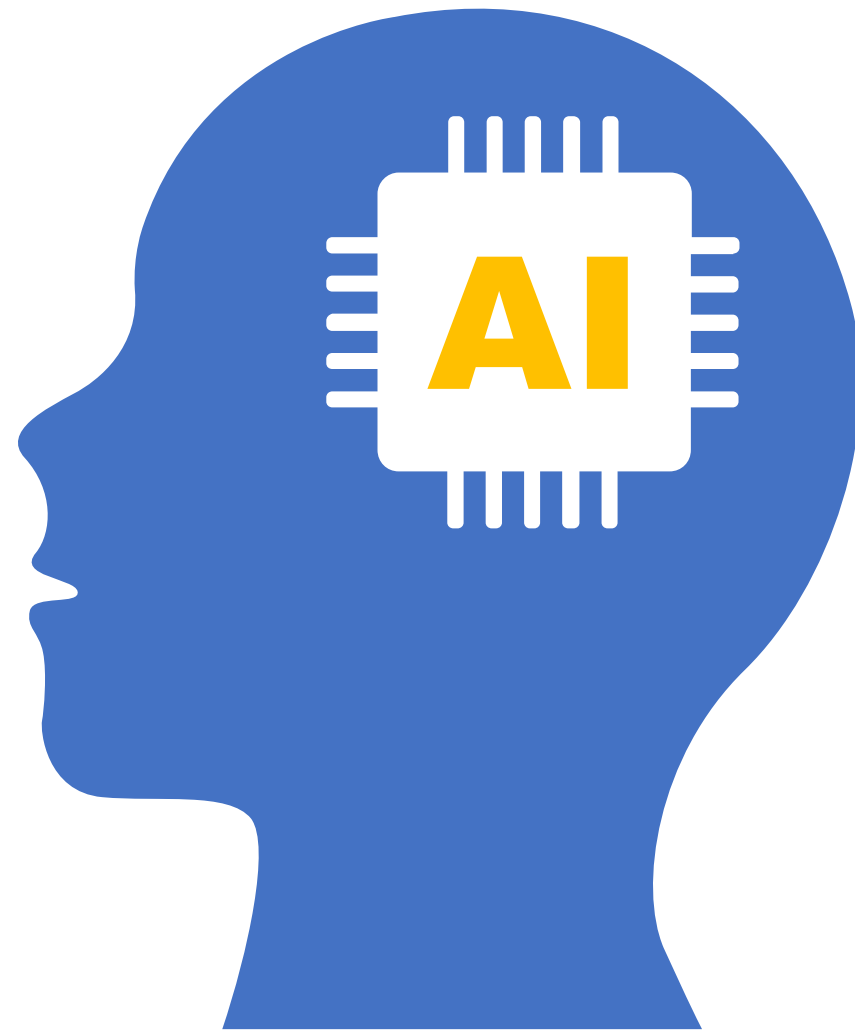
- 請打開 Hugging Face 的 Models 頁面：
 - <https://huggingface.co/models>
- 假設你想找一款能夠辨識「圖+文」的大語言模型，請點擊：
 - 左側欄：Multimodal → Image-Text-to-Text。
 - 右上角：依照「Most Downloads」排序。
- 說說看，最多人下載的「圖文識別」大語言模型叫什麼名字？
- 如果你下載的模型，想自帶「使用者圖形介面（GUI）」之定義，應該下載帶有哪個字樣的模型？



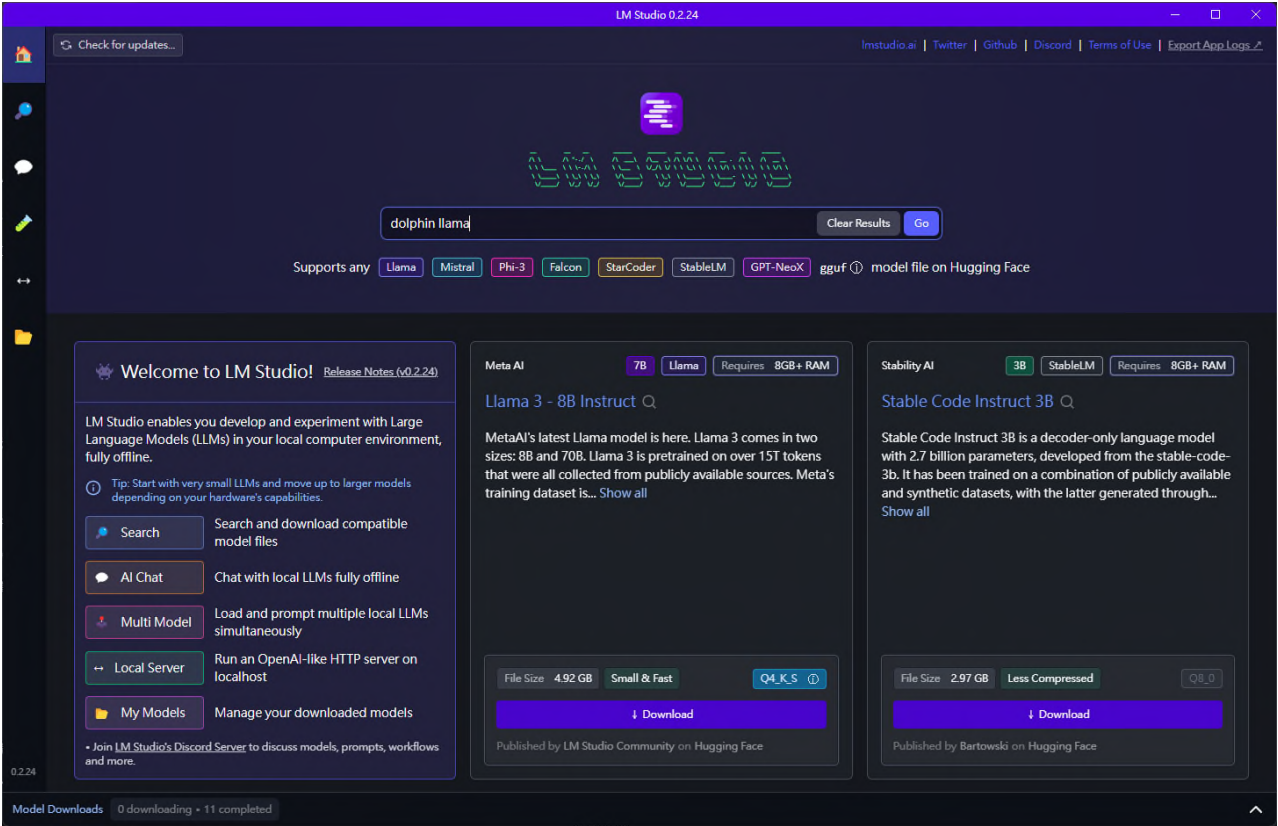


使用開源大語言模型

- 簡介
- 使用 LM Studio
- 使用 Ollama



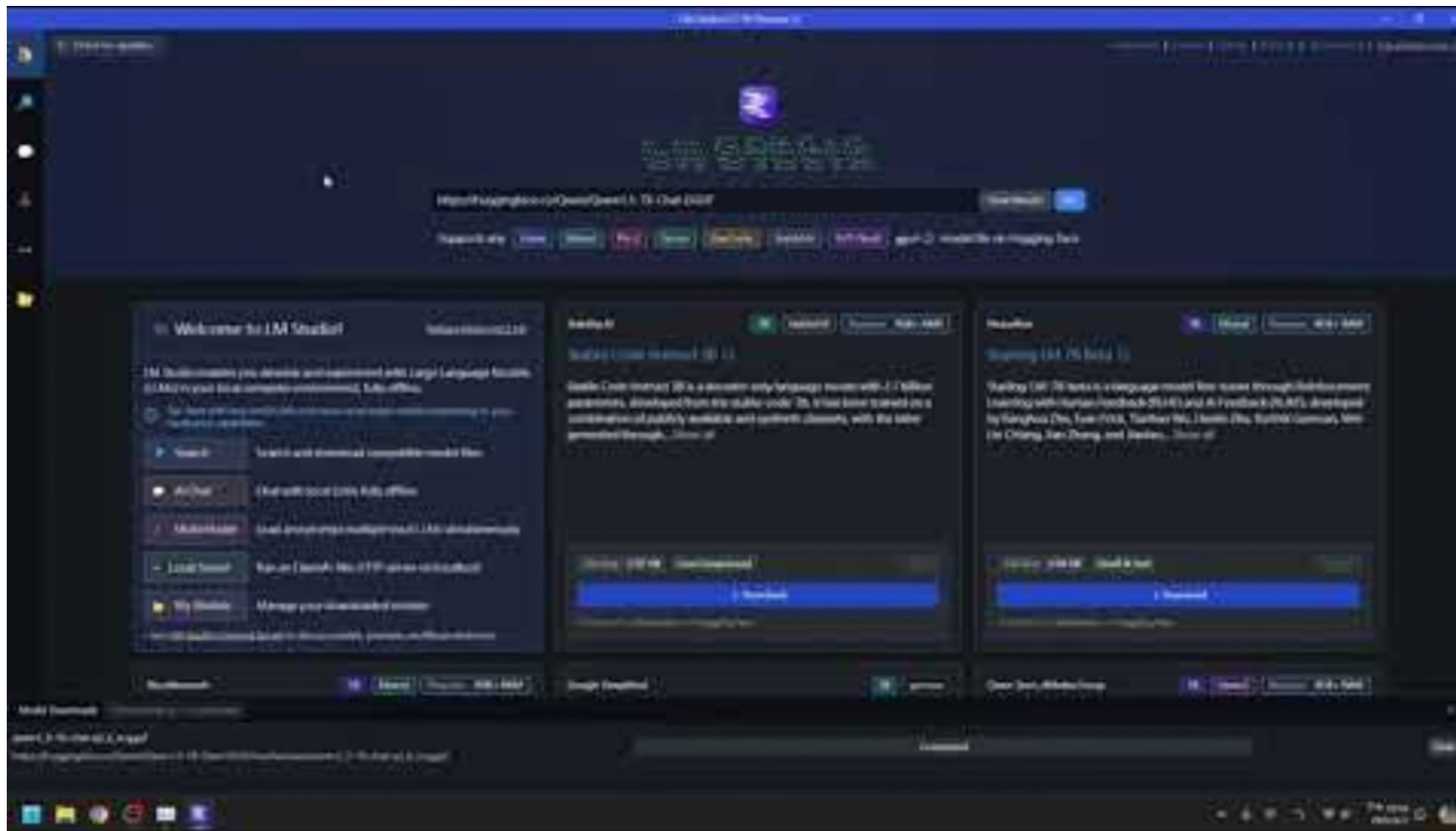
- 能下載、運行 Hugging Face 上任何 GGUF 模型檔的軟體



安裝、搜尋、下載、使用



示範影片





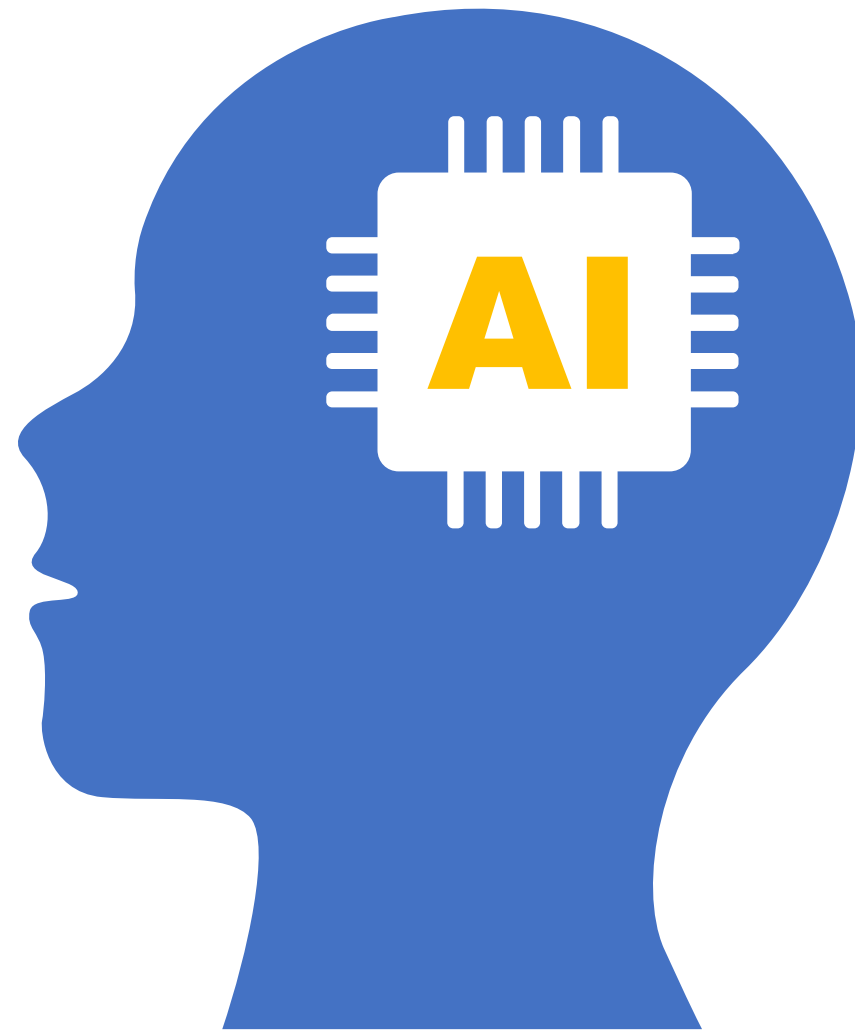
- 請先下載 & 安裝 LM Studio。
- 搜尋 Microsoft 推出的 Phi 3 大語言模型，並下載之。
- 試著在 Chat 面板，掛載 Microsoft / Phi 3，並且與之對話。
- 看老師講解下列兩個左側欄功能：
 - Local Server：將 LLM 以伺服器模式執行，供程式呼叫。
 - My Models：對已下載的 LLM，進行刪除、修改之管理。





使用開源大語言模型

- 簡介
- 使用 LM Studio
- 使用 Ollama



- 支援下載、運行大語言模型於本地端的命令列工具（程式適用）

```
命令提示字元
Microsoft Windows [版本 10.0.22631.3593]
(c) Microsoft Corporation. 著作權所有，並保留一切權利。

C:\Users\cnchi>ollama
Usage:
  ollama [flags]
  ollama [command]

Available Commands:
  serve      Start ollama
  create     Create a model from a Modelfile
  show       Show information for a model
  run        Run a model
  pull       Pull a model from a registry
  push       Push a model to a registry
  list       List models
  ps         List running models
  cp         Copy a model
  rm         Remove a model
  help       Help about any command

Flags:
  -h, --help      help for ollama
  -v, --version    Show version information

Use "ollama [command] --help" for more information about a command.
C:\Users\cnchi>
```

Ollama 操作界面

Model	Parameters	Size	Download
Llama 3	8B	4.7GB	<code>ollama run llama3</code>
Llama 3	70B	40GB	<code>ollama run llama3:70b</code>
Phi 3 Mini	3.8B	2.3GB	<code>ollama run phi3</code>
Phi 3 Medium	14B	7.9GB	<code>ollama run phi3:medium</code>
Gemma	2B	1.4GB	<code>ollama run gemma:2b</code>
Gemma	7B	4.8GB	<code>ollama run gemma:7b</code>
Mistral	7B	4.1GB	<code>ollama run mistral</code>
Moondream 2	1.4B	829MB	<code>ollama run moondream</code>
Neural Chat	7B	4.1GB	<code>ollama run neural-chat</code>
Starling	7B	4.1GB	<code>ollama run starling-lm</code>
Code Llama	7B	3.8GB	<code>ollama run codellama</code>
Llama 2 Uncensored	7B	3.8GB	<code>ollama run llama2-uncensored</code>
LLaVA	7B	4.5GB	<code>ollama run llava</code>
Solar	10.7B	6.1GB	<code>ollama run solar</code>

Ollama 常用模型（完整列表）

• 安裝

• <https://ollama.com/>



Get up and running with large language models.

Run [Llama 3](#), [Phi 3](#), [Mistral](#), [Gemma](#), and other models. Customize and create your own.

Download ↓

Available for macOS, Linux, and Windows (preview)

Download Ollama



Download for Windows (Preview)

Requires Windows 10 or later

• 下載使用

• ollama run <模型名稱>

```
命令提示字元 - ollama run gemma

C:\Users\cnchi>ollama run gemma
>>> Send a message (!? for help)
```

```
命令提示字元 - ollama run gemma

C:\Users\cnchi>ollama run gemma
>>> 請問「自然語言處理」是什麼東西？
**自然語言處理 (NLP)** 是一種用於電腦處理人類語言的技術。它包括用於理解、生成和轉換語言的各種演算法和技術。

**自然語言處理的目標：**
* 讓電腦理解人類語言的語義和語法。
* 將人類語言轉換為電腦可理解的表示。
* 將電腦生成的語言轉換為人類可理解的表示。

**自然語言處理的應用：**
* **語義解析：** 理解文字的含義。
* **語法分析：** 分析文字的語法結構。
* **詞彙建置：** 建立詞彙庫並識別詞根和詞彙。
* **情感分析：** 識別文字的情感極性。
* **機器翻譯：** 將文字從一種語言轉換為另一種語言。
* **語音處理：** 處理和理解音訊。
```

想離開：輸入 /bye



隨堂練習：使用 Ollama



- 請先下載 & 安裝 Ollama 。
- 到 Ollama [模型清單頁面](#)，找尋 Microsoft Phi 3 。
- 在命令列輸入 **ollama run phi3**，下載並執行之。
- 試著與之**對話**。不想對話請輸入 **/bye**。





小節整理



- 了解何謂「大語言模型 (LLM) 」
- 知道常用的 LLM 有哪些？去哪裡下載？適用於哪些場景？
- 能用 LM Studio 下載大語言模型，並使用之。
- 能用 Ollama 下載大語言模型，並使用之。
- 結論
 - 想讓 LLM 與真人互動 → 推薦使用 LM Studio。
 - 想讓 LLM 與程式互動 → 推薦使用 Ollama。





開源大語言模型： 程式篇

安裝 GPU 驅動相關套件



1 指定 Colab 使用 GPU

筆記本設定

執行階段類型

Python 3

硬體加速器 ?

☐ CPU ☐ A100 GPU ☐ L4 GPU ☒ T4 GPU ☐ TPU (deprecated)

☐ TPU v2

大量 RAM ☐

☐ 執行時，一律自動執行第一個儲存格或區段

☐ 儲存這個筆記本時，忽略程式碼儲存格輸出內容

取消 儲存

2

```
1 # 更新 Linux 內的套件清單至最新版
2 !apt-get update
3
```

3

```
4 # 安裝 PCI 匯流排工具 (PCI Utility) 與 lshw (LiSt HardWare), 以便能偵測到 GPU
5 # -y : 遇到詢問是否安裝, 一律自動回答 yes
6 !apt-get install -y pciutils lshw
7
```

4

```
8 # 用 nVidia 的 System Management Interface (SMI) 確認 GPU 的確抓得到
9 !nvidia-smi
```

NVIDIA-SMI 535.104.05			Driver Version: 535.104.05		CUDA Version: 12.2		
GPU	Name	Persistence-M	Bus-Id	Disp.A	Volatile Uncorr. ECC		
Fan	Temp	Perf	Pwr:Usage/Cap	Memory-Usage	GPU-Util	Compute M.	MIG M.
0	Tesla T4	Off	00000000:00:04:0	Off	0		
N/A	34C	F8	9W / 70W	0MiB / 15360MiB	0%	Default	N/A
Processes:							
GPU	GI	CI	PID	Type	Process name	GPU Memory	
	ID	ID				Usage	
No running processes found							



- 請先建立一個新的 **Colab 頁面**。
- 在「**編輯** > **筆記本設定**」中，選擇 **T4 GPU**。
- 撰寫下列程式碼，並執行之：

```
1 # 更新 Linux 內的套件清單至最新版
2 !apt-get update
3
4 # 安裝 PCI 匯流排工具 (PCI Utility) 與 lshw (LiSt HardWare), 以便能偵測到 GPU
5 # -y : 遇到詢問是否安裝, 一律自動回答 yes
6 !apt-get install -y pciutils lshw
7
8 # 用 nVidia 的 System Management Interface (SMI) 確認 GPU 的確抓得到
9 !nvidia-smi
```





安裝 Ollama 與大語言模型



```
1 # 至 https://ollama.com/download/linux
2 # 直接將安裝 Ollama 於 Linux 的指令貼上
3 !curl -fsSL https://ollama.com/install.sh | sh
4
5 # 啟動 Ollama, 讓它執行於背景中
6 # ollama serve: 用 Server 模式、而非互動模式執行 Ollama
7 # > server.log: 將本應顯示於螢幕的訊息, 轉向輸出至 server.log 這個檔備查
8 # 2>&1: 2 為 stderr。將所有錯誤訊息, 轉向 &1 (stdout, 螢幕) 輸出。
9 # &: 將程式啟動之後, 馬上返回, 不要等該程式執行完成
10 !ollama serve > server.log 2>&1 &
11
12 # 將 Llama-3 模型下載, 並做為此次的大語言模型
13 # ollama run <模型名稱>: 下載並執行特定 LLM
14 # > model.log: 將本應顯示於螢幕的訊息, 轉向輸出至 model.log 這個檔備查
15 !ollama run llama3 > model.log 2>&1 &
16
17 # 注意: 上述兩指令皆以「&」後綴, 告知 Colab「不用等 Linux 執行完」。
18 # 但事實上, 不論啟動為 Server, 或下載 LLM, 皆須 1~5 分鐘不等的時間。
19 # 可以查看 server.log、model.log 兩檔案內容, 得知當前執行狀況。
```



- 請撰寫下列程式碼，並執行之：

```
1 # 至 https://ollama.com/download/linux
2 # 直接將安裝 Ollama 於 Linux 的指令貼上
3 !curl -fsSL https://ollama.com/install.sh | sh
4
5 # 啟動 Ollama, 讓它執行於背景中
6 # ollama serve: 用 Server 模式、而非互動模式執行 Ollama
7 # > server.log: 將本應顯示於螢幕的訊息, 轉向輸出至 server.log 這個檔備查
8 # 2>&1: 2 為 stderr。將所有錯誤訊息, 轉向 &1 (stdout, 螢幕) 輸出。
9 # &: 將程式啟動之後, 馬上返回, 不要等該程式執行完成
10 !ollama serve > server.log 2>&1 &
11
12 # 將 Llama-3 模型下載, 並做為此次的大語言模型
13 # ollama run <模型名稱>: 下載並執行特定 LLM
14 # > model.log: 將本應顯示於螢幕的訊息, 轉向輸出至 model.log 這個檔備查
15 !ollama run llama3 > model.log 2>&1 &
16
17 # 注意: 上述兩指令皆以「&」後綴, 告知 Colab「不用等 Linux 執行完」。
18 # 但事實上, 不論啟動為 Server, 或下載 LLM, 皆須 1~5 分鐘不等的時間。
19 # 可以查看 server.log、model.log 兩檔案內容, 得知當前執行狀況。
```





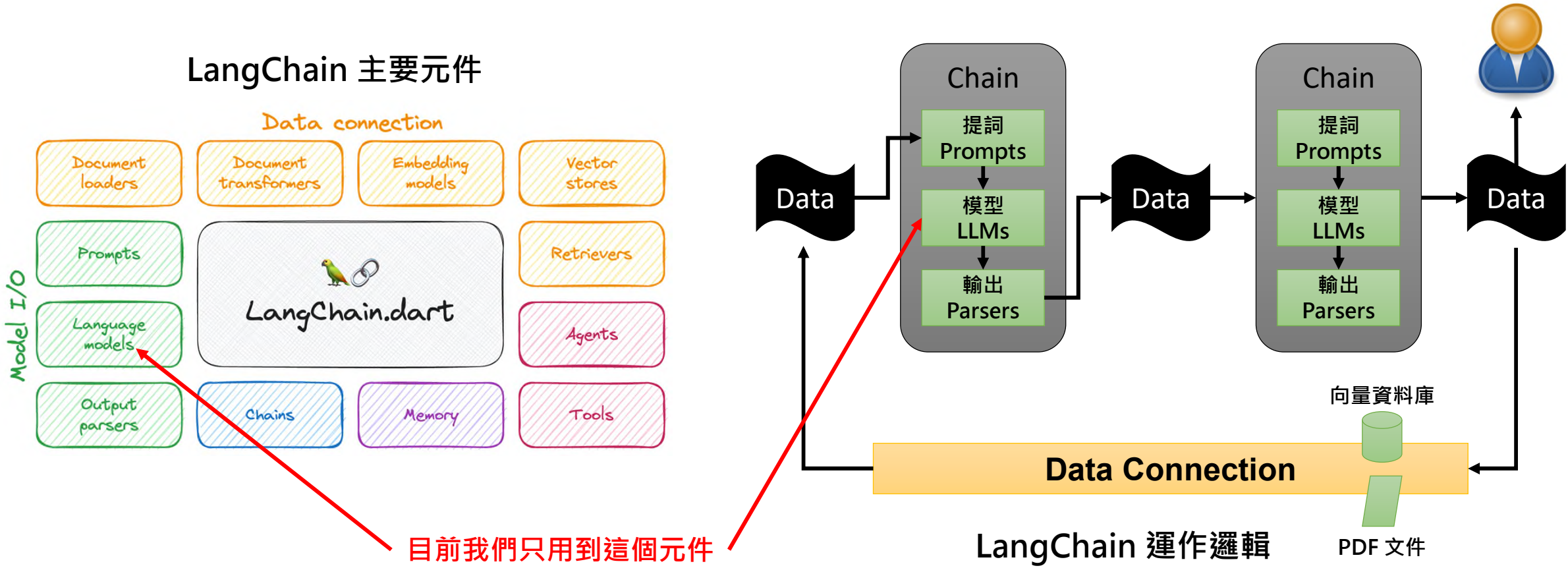
```
1 # 下載 LangChain, 一套專門連上各種大語言模型的 Python 套件
2 !pip install langchain # LangChain 核心元件
3 !pip install langchain-community # 各種開源大語言模型連接函數套件
4
5 # 載入 Ollama 以便連上後端 LLM
6 from langchain_community.llms import Ollama
7
8 # LLM 名稱需與 ollama run 後方名稱相同
9 llm = Ollama(model="llama3")
10
11 # 對 LLM 送出提詞, 並且印出回應
12 msg = llm.invoke("什麼是「自然語言處理」呢? 可以用繁體中文回答我嗎?")
13 print(msg)
```

「自然語言處理」(Natural Language Processing · 簡稱 NLP) 是一個跨學科領域的研究和應用領域，旨在研究和發展使用 computer 和人工智慧 (Artificial Intelligence) 來分析、理解、生成和控制自然語言 (Human Language) 的能力。

NLP 的主要目的是：

1. 分析：對於文本或音訊進行分析，例如，斷句、命名實體識別 (Named Entity Recognition) 、 Dependency Parsing 等。
2. 理解：了解語言的語義和意義，例如，情感分析 (Sentiment Analysis) 、意圖識別 (Intent Identification) 等。
3. 生成：產生新的文本或音訊，例如，自動寫作、對話系統 (Chatbot) 等。
4. 控制：控制語言的處理和應用，例如，語音助手 (Voice Assistant) 、自然語言基於的智慧家居設備等。

- 僅用幾個指令，就能與 LLM 交談之程式套件（2022/10/25）





隨堂練習：與大語言模型對話



- 請撰寫下列程式碼，並執行之：

```
1 # 下載 LangChain, 一套專門連上各種大語言模型的 Python 套件
2 !pip install langchain # LangChain 核心元件
3 !pip install langchain-community # 各種開源大語言模型連接函數套件
4
5 # 載入 Ollama 以便連上後端 LLM
6 from langchain_community.llms import Ollama
7
8 # LLM 名稱需與 ollama run 後方名稱相同
9 llm = Ollama(model="llama3")
10
11 # 對 LLM 送出提詞，並且印出回應
12 msg = llm.invoke("什麼是「自然語言處理」呢？可以用繁體中文回答我嗎？")
13 print(msg)
```





• 問題 & 需求說明

- 不少 YouTube 並沒有附上字幕，當然也無法把字幕交給 LLM 做摘要濃縮。
- 製作一款程式，可以讓使用者輸入 YouTube URL。如下：
 - <https://www.youtube.com/watch?v=zKndCikg3R0>
- 你的程式要能將該影片的逐字稿辨識出來。如：
 - "英國研發的AI人型機器人Amika一亮相就警告人類最可怕的情況我可以想像的是...TVBS新聞 政府報導"
 - "Humans need language to communicate, so it makes sense that ...like and subscribe."
- 連上你所使用的 LLM，設計適當的提詞 (Prompts)，使 LLM 能將逐字稿標點符號補上，並翻譯成你所指定的語言 (如：繁體中文) 後印出。
- 設計適當的提詞 (Prompts)，再次要求 LLM，請它針對逐字稿，將內容摘要 (Summarize) 後，以條列 (Bullet Points) 的方式印出來。

• 評分標準

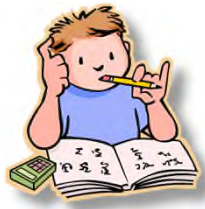
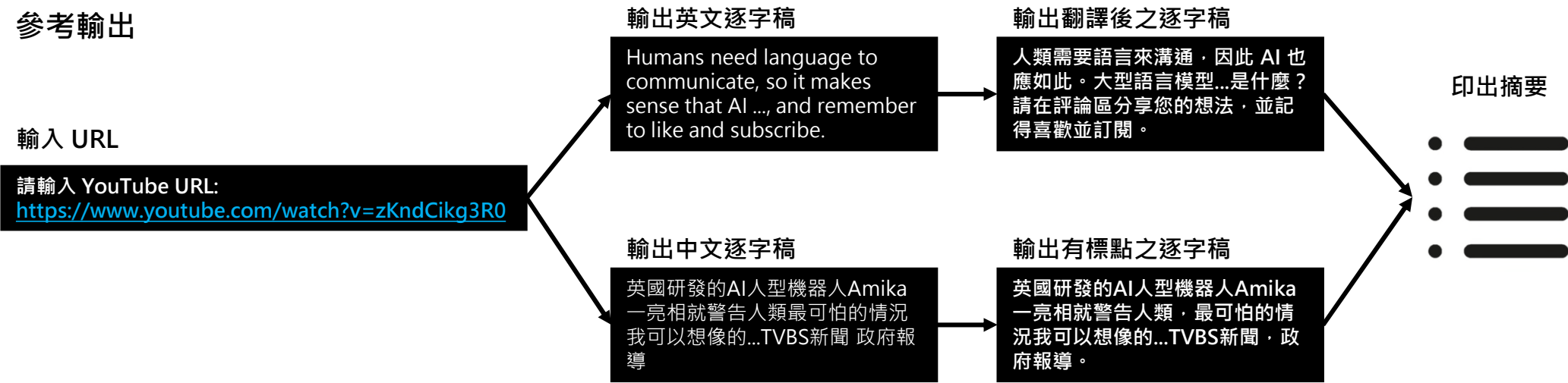
- 要能夠讓使用者輸入 YouTube URL (一分)。
- 要能夠印出逐字稿，且逐字稿需有標點符號，並以慣用語言 (如：繁體中文) 輸出 (四分)。
- 要能將逐字稿摘要整理成條列的形式，印在螢幕上 (二分)。



• 提示

- 大語言模型的安裝、設定、操作，請參考本章範例。
- 由於 Llama-3 對中文調校不良，常常「問中文，答英文」。有此現象，可改用 Gemma 7B 模型。需要修改的指令如下：
 - `!ollama run gemma:7b > model.log 2>&1 &`
 - `llm = Ollama(model="gemma:7b")`
- 影片逐字稿的取得，可以參考本課程「[資料取得](#)」一章之範例。
- Colab 記憶體有限，測試用影片時長請盡量維持在 4~5 分鐘以內。
- LLM 的輸出內容如果不滿意，請重複調整提詞 (Prompts)，讓它能夠輸出你要求的結果。

• 參考輸出





- **大語言模型**
 - Large Language Models (LLM)
- **Transformer 演算法**
 - RNN 遇到長距離參考，容易梯度爆炸或消失。
 - Transformer = 擁有注意力機制的 Autoencoder。
 - 常見 Transformer = BERT, GPT, T5, ...
- **開源大語言模型之使用**
 - 常見開源大語言模型 = Gemma, Llama-3, Phi, Mistral ...。
 - 開源大語言模型集散地：Hugging Face。
 - 互動使用：LM Studio
 - 程式使用：Ollama
- **開源大語言模型之程式設計**
 - 用 Ollama 做為 LLM 的管理軟體。
 - 用 LangChain 負責串起 Python 與 LLM 兩端。

