**大型語言模型 (LLM) 教學簡介**

**1. LLM 的定義與原理**

* **基本定義：大型語言模型（Large Language Model, LLM）是一種具有龐大參數量**的機器學習模型，專門用於自然語言處理任務，能夠理解與生成文字​

[en.wikipedia.org](https://en.wikipedia.org/wiki/Large_language_model#:~:text=A%20large%20language%20model%20,a%20vast%20amount%20of%20text)

。LLM 本質上是高階的語言模型，透過在海量文本語料上訓練，學習語言統計模式以預測後續文字。

* **運作原理：LLM 使用機率預測**的方式產生文字，透過給定前文來預測下一個最可能的詞彙或「tokens」。例如，對於輸入句子“Mary had a little”，模型會預測下一個字很可能是“lamb”​

[cset.georgetown.edu](https://cset.georgetown.edu/article/the-surprising-power-of-next-word-prediction-large-language-models-explained-part-1/#:~:text=If%20you%20are%20given%20the,1)

。模型基於訓練數據中學到的模式，為每個可能的下一字計算機率，選擇最高機率者做為輸出​

[cset.georgetown.edu](https://cset.georgetown.edu/article/the-surprising-power-of-next-word-prediction-large-language-models-explained-part-1/#:~:text=When%20an%20LLM%20receives%20a,was%20generated%20by%20the%20model)

。如此逐字預測，LLM 得以生成通順的句子和段落。

* \*\*上下文學習：\*\*LLM 能根據提示（prompt）提供連續的回答，因為它可以將之前生成的內容視為新的上下文，再次預測下一步文字，連續迭代下去​

[cset.georgetown.edu](https://cset.georgetown.edu/article/the-surprising-power-of-next-word-prediction-large-language-models-explained-part-1/#:~:text=These%20examples%20only%20involve%20short,distant%20from%20the%20initial%20input)

。這讓模型能夠完成長篇幅的生成任務，例如整篇文章撰寫或長對話。

* \*\*「大型」的意義：\*\*LLM 被稱為大型，是因為其參數規模遠超傳統模型，往往達到億億級別（數十億以上參數）​

[en.wikipedia.org](https://en.wikipedia.org/wiki/Large_language_model#:~:text=A%20large%20language%20model%20,a%20vast%20amount%20of%20text)

。龐大的參數量使模型擁有更強的表達能力，可以從海量數據中學習細緻的語言規律。然而，大規模也意味著需要複雜的訓練過程和高運算資源支援（詳見第5點）。

**2. LLM 常見架構：Transformer 架構**

* \*\*Transformer 簡介：**主流的 LLM 幾乎都基於 Transformer 神經網路架構​**

[**elastic.co**](https://www.elastic.co/what-is/large-language-models#:~:text=So%2C%20what%20is%20a%20transformer,model)

**。Transformer 由 Google 團隊在 2017 年提出（論文〈Attention Is All You Need〉），其創新在於摒棄了傳統 RNN 的順序處理，轉而採用**自注意力機制（Self-Attention）\*\*來並行處理序列資料​

[kolena.com](https://www.kolena.com/guides/transformer-vs-rnn-4-key-differences-and-how-to-choose/#:~:text=1)

。Transformer 原始架構包含編碼器（Encoder）和解碼器（Decoder）兩部分：編碼器將輸入句子轉換為內部表示，解碼器根據該表示產生輸出句子​

[elastic.co](https://www.elastic.co/what-is/large-language-models#:~:text=So%2C%20what%20is%20a%20transformer,model)

。例如在機器翻譯中，編碼器讀入英文句子表示其語義，解碼器再生成對應的中文翻譯。

* **Self-Attention 自注意力：Transformer 的核心是自注意力機制，它允許模型在處理每個詞時，都能關注同一句子中其他相關的詞**。簡而言之，模型為輸入序列中的每對詞計算一個「注意力權重」，以衡量一個詞對理解另一個詞的重要程度​

[elastic.co](https://www.elastic.co/what-is/large-language-models#:~:text=Transformer%20models%20work%20with%20self,a%20sentence%2C%20to%20generate%20predictions)

。例如在句子中，主語和謂語可能會互相賦予較高權重。透過自注意力，模型可以在每一層動態調整對不同上下文詞的關注，捕捉長程依賴關係​

[kolena.com](https://www.kolena.com/guides/transformer-vs-rnn-4-key-differences-and-how-to-choose/#:~:text=Transformers%20handle%20dependencies%20and%20memory,position%20in%20the%20input%20sequence)

。此外，Transformer 使用**多頭注意力（Multi-Head Attention）**，即同時執行多組不同的自注意力計算，讓模型從多種角度學習語意關聯，大幅提升了表達能力。

* **並行計算與長距依賴：相較於 RNN 逐步處理序列的方式，Transformer 可以並行處理整個序列，因而訓練和推理速度更快​**

[**kolena.com**](https://www.kolena.com/guides/transformer-vs-rnn-4-key-differences-and-how-to-choose/#:~:text=Transformers%20use%20parallel%20processing%20to,scale%20parallel%20computational%20power)

**。這種並行化得益於自注意力機制和位置編碼（Positional Encoding）的配合：儘管 Transformer 沒有內建順序迴路，但透過加入位置編碼向量，模型能知曉詞語在序列中的相對位置。由於自注意力可以直接連結序列中任意兩個位置，Transformer 尤其擅長捕捉長距離的依存關係**而不會像 RNN 那樣出現梯度消失問題​

[kolena.com](https://www.kolena.com/guides/transformer-vs-rnn-4-key-differences-and-how-to-choose/#:~:text=Transformers%20handle%20dependencies%20and%20memory,position%20in%20the%20input%20sequence)

。實踐證明，Transformer 在很長的文本上依然能保持良好性能，這對於長文摘要、長對話上下文等任務十分關鍵。

* \*\*Encoder-Decoder 與變體：\*\*原始 Transformer 同時包含編碼器和解碼器，但在實際應用中有所變體。例如，**GPT 系列**模型採用僅有解碼器的 Transformer（自迴歸語言模型），專注於生成任務；**BERT** 等模型則採用僅有編碼器的架構，用於理解類任務（如填空、分類）​

[en.wikipedia.org](https://en.wikipedia.org/wiki/Large_language_model#:~:text=At%20the%202017%20NeurIPS%20,improvements%20in%20the%20abilities%20of)

。無論編碼器或解碼器，內部都堆疊多層自注意力和前饋神經網路，以逐步提取高層次特徵。總之，Transformer 架構奠定了 LLM 的基石，使模型能有效學習語言中的複雜模式。

**3. LLM 的訓練方式與資料來源**

* **大規模語料預訓練：LLM 的訓練首先在巨量文本語料上進行預訓練（pre-training）**，通常採用**自我監督學習**方式，即利用未標註的純文本來學習。常用語料來源包括開放網頁（如 **Common Crawl** 的網際網路爬蟲資料）、維基百科、書籍、新聞文章、論壇帖子等​

[hackthefuture.com](https://www.hackthefuture.com/news/how-llms-are-trained-and-how-red-teaming-helps#:~:text=The%20first%20step%20is%20to,vast%20datasets%2C%20like%20common%20crawl)

​

[hackthefuture.com](https://www.hackthefuture.com/news/how-llms-are-trained-and-how-red-teaming-helps#:~:text=So%2C%20before%20we%20can%20train,has%20tens%20of%20trillions%20of)

。由於資料極為龐大且品質不一，訓練前需要**資料清理**：去除重複內容、低品質文本和不當言論等​

[hackthefuture.com](https://www.hackthefuture.com/news/how-llms-are-trained-and-how-red-teaming-helps#:~:text=The%20raw%20datasets%20are%20repetitive,if%20it%20has%20been%20uncovered)

。例如，OpenAI 為訓練 GPT-3 準備的語料最初來自約45 TB的壓縮網路文本，經過過濾和去重後縮減至僅0.6 TB，以確保多樣性和降低有害內容​

[hackthefuture.com](https://www.hackthefuture.com/news/how-llms-are-trained-and-how-red-teaming-helps#:~:text=So%2C%20before%20we%20can%20train,has%20tens%20of%20trillions%20of)

。最終 GPT-3 使用了約4990億個標記（tokens）的訓練語料，其中包含大量來自 Common Crawl 的資料​

[hackthefuture.com](https://www.hackthefuture.com/news/how-llms-are-trained-and-how-red-teaming-helps#:~:text=So%2C%20before%20we%20can%20train,has%20tens%20of%20trillions%20of)

。

* **自我監督目標：預訓練時最常用的目標是語言模型目標**，即讓模型預測給定文本片段中下一個字詞是哪一個（這屬於*自我監督*，因為答案由文本本身提供）。例如，給模型一段文字“今天天氣很好，我想去”，讓它預測下一個可能是“散步”。模型不斷通過降低預測錯誤來調整內部權重。這種方式也稱為**自迴歸語言模型**訓練（如 GPT 系列），或在某些模型中使用**遮掩語言模型**目標（如 BERT，隨機遮掩部分詞讓模型預測）。總之，模型學會在廣泛語境下**預測文字的能力**​

[cset.georgetown.edu](https://cset.georgetown.edu/article/the-surprising-power-of-next-word-prediction-large-language-models-explained-part-1/#:~:text=If%20you%20are%20given%20the,1)

​

[cset.georgetown.edu](https://cset.georgetown.edu/article/the-surprising-power-of-next-word-prediction-large-language-models-explained-part-1/#:~:text=When%20an%20LLM%20receives%20a,was%20generated%20by%20the%20model)

，這賦予了 LLM 對語言的基本理解和生成能力。

* **監督微調 (Fine-tuning)：在預訓練完成後，開發者常使用較小且有標註**的資料對模型進行微調，以適應特定任務或提高模型對人類指令的響應能力​

[hackthefuture.com](https://www.hackthefuture.com/news/how-llms-are-trained-and-how-red-teaming-helps#:~:text=To%20give%20you%20a%20solid,responses%20to%20make%20them%20better)

。微調可以是**監督學習**形式，例如提供范例對話或問答對，訓練模型按照給定輸入產生期望輸出。透過微調，LLM 可以從通用的「預訓練知識」轉化為專精某類任務的模型。例如，使用問答資料微調可使模型在問答系統中表現更佳；使用對話資料微調並適當懲戒模型的語氣，可得到一個禮貌且遵循指令的聊天助理模型。

* **人類反饋強化學習 (RLHF)：近期先進的聊天模型（如 ChatGPT）在監督微調後，還引入從人類反饋中強化學習**的階段​

[hackthefuture.com](https://www.hackthefuture.com/news/how-llms-are-trained-and-how-red-teaming-helps#:~:text=To%20give%20you%20a%20solid,responses%20to%20make%20them%20better)

。具體作法是讓人類標註者對模型的回應進行質量評分，然後用這些反饋訓練一個獎勵模型，再通過強化學習（例如策略梯度法）調整 LLM，使其傾向產生獲得高分的回應。RLHF 有助於模型更好地對齊人類偏好，產生**更有用且無害**的回答。例如，透過 RLHF，模型學會迴避不當提問、遵守內容政策並給出詳細有幫助的答覆。這一步驟顯著提升了聊天機器人的實用性和安全性。

* **遷移學習與微調效益：LLM 經過預訓練已掌握通用的語言表示，因此開發者只需少量特定資料做微調，就能遷移**模型到新任務上，這遠較從零訓練模型高效。這種遷移學習能力是 LLM 的重要特點之一：例如，一個醫學領域的問答系統，可從通用 LLM 出發，透過在醫療問答資料上微調得到，無需蒐集海量醫療語料重新訓練整個模型。總結而言，LLM 的訓練通常分三階段：先大規模**自我監督預訓練**，再針對任務的**監督微調**，最後透過**人類反饋調教**模型行為，逐步打造出在實際應用中表現優異且符合人類期望的模型。

**4. LLM 的應用場景**

* \*\*對話聊天 (ChatGPT 等)：\*\*LLM 最廣為人知的應用是智能對話助理，例如 OpenAI 的 ChatGPT。這類應用中，模型充當聊天機器人，能以貼近人類的方式與使用者交流，回答一般知識問答、提供建議、協助寫作等。由於 LLM 擁有大規模知識和語言生成能力，用戶可以就各種主題與之對談並獲得有意義的回應，從日常資訊詢問到專業領域知識皆涵蓋其中。
* \*\*機器翻譯：\*\*早期的機器翻譯依賴統計或 RNN 方法，如今大型語言模型（特別是 Transformer 架構）已成為機器翻譯的核心技術。LLM 可以在不使用傳統詞典規則的情況下，學習兩種語言之間的對應關係，實現高品質的自動翻譯。例如，Google 翻譯自 2016 年起全面採用神經網路翻譯 (Neural Machine Translation)，提升了翻譯流暢度​

[en.wikipedia.org](https://en.wikipedia.org/wiki/Large_language_model#:~:text=After%20neural%20networks%20became%20dominant,deep%20%20286%20networks)

。現代 LLM（如 Meta 的 NLLB-200 模型）甚至支持數百種語言的翻譯。在日常應用中，透過 LLM 驅動的翻譯工具，使用者可以即時將一段文字從一種語言轉換為另一種語言，極大地方便了跨語言交流。

* \*\*問答系統：\*\*LLM 驅動的問答系統能夠根據自然語言問題從知識庫中提取答案。例如，FAQ客服機器人、知識圖譜問答、醫學問答平台等都可利用 LLM 理解用戶的提問並給出相關答案。因為預訓練語料中包含廣泛的常識和專業知識，LLM 在很多情況下能直接回答事實性問題。同時，結合資料庫或搜索引擎（檢索增強型生成，RAG），LLM 可先檢索資料再作答以提高準確性。應用實例包括：維基百科問答助手、法律諮詢AI、校園問答系統等，使得複雜問題的答案獲取變得快捷而直觀。
* **內容生成：內容創作是 LLM 的強項之一。給定一些提示，模型可以自動生成各類文本，例如：撰寫新聞稿、產品描述、劇本對話、小說片段，甚至學生作文。LLM 已被用於文章摘要**（從長文檔中提取重點）、**電子郵件或報告撰寫**（自動生成初稿）、**社群媒體內容**（例如自動產生貼文或回覆）等場景​

[elastic.co](https://www.elastic.co/what-is/large-language-models#:~:text=tuned%20so%20that%20they%20can,AI%20assistants%2C%20and%20so%20on)

​

[elastic.co](https://www.elastic.co/what-is/large-language-models#:~:text=can%20be%20applied%20to%20fields,AI%20assistants%2C%20and%20so%20on)

。由於生成內容往往語法通順且風格多樣，許多內容創作者將 LLM 作為輔助工具，加速文案製作。然而在實務中，生成內容仍需人員審核，以避免模型潛在的錯誤或不當輸出。

* **代碼輔助：近年來，LLM 在程式碼領域的應用也取得了驚人成果。大型語言模型（如 OpenAI Codex、GPT-4、Meta 的 Code Llama）已能理解程式語言並生成代碼片段。開發人員可以以自然語言描述他們需要的功能，模型即可輸出相應的程式碼實現。此外，LLM 可用於程式碼自動補全**（如 GitHub Copilot 根據上下文補全函式）、**代碼解說與除錯**（將難懂的程式碼翻譯成人類語言或找出可能的 bug）、**單元測試生成**等。有了代碼輔助工具，程式設計師能提高開發效率，初學者也能藉此獲得學習支援。不過模型生成的程式碼有時會存在錯誤或不最佳的地方，因此通常需由開發者進行測試和修改。
* **客服與輔助服務：許多企業將 LLM 應用在自動客服聊天機器人**上，以提供7×24小時的線上諮詢服務。這類系統能即時回答用戶關於產品、業務的常見問題，或引導用戶完成某些操作（如密碼重設指引）。由於 LLM 擁有上下文理解能力，客服機器人可以處理多輪對話，記住用戶先前提供的信息，提供連貫且個性化的服務體驗​

[elastic.co](https://www.elastic.co/what-is/large-language-models#:~:text=tuned%20so%20that%20they%20can,AI%20assistants%2C%20and%20so%20on)

。除了客服外，LLM 還可作為**個人助理**應用於手機或智慧喇叭上，執行語音指令、日程管理、電子郵件撰寫等任務。例如，Apple Siri、Google Assistant 等也在引入更強大的語言模型以提升對話智能和任務處理能力。

* \*\*其他應用：\*\*LLM 的應用幾乎遍及所有涉及語言的領域。例如：文本分類（如垃圾郵件偵測、情感分析）、資訊擷取（從文件中抓取關鍵訊息）、教育輔助（自動批改作業、解答學生提問）、醫療健康（分析醫療記錄、提供初步診斷建議）等。甚至在非語言領域，LLM 也有間接應用，如理解蛋白質序列（將其視為“語言”）​

[elastic.co](https://www.elastic.co/what-is/large-language-models#:~:text=In%20addition%20to%20teaching%20human,AI%20assistants%2C%20and%20so%20on)

。總之，LLM 作為通用的語言智能技術，正在各行各業催生新的應用場景，改變我們與技術互動的方式。

**5. LLM 的參數規模與硬體需求**

* **參數規模（Billion級）：LLM 動輒具有數十億到上千億不等的參數（權重）。這裡的「參數」指模型經訓練後學到的權重值數量，是衡量模型容量的關鍵指標。以 GPT-3 為例，其擁有 1750 億 個參數​**

[**developer.nvidia.com**](https://developer.nvidia.com/blog/openai-presents-gpt-3-a-175-billion-parameters-language-model/#:~:text=OpenAI%20researchers%20recently%20released%20a,up%20of%20175%20billion%20parameters)

**（而上一代 GPT-2 為15億）。再如 Google 的 PaLM 2 模型約有 5400 億參數，Meta 的最大 LLaMA 2 模型有 700 億參數等。如此龐大的參數量使模型具備記憶海量語言模式的能力，從而在廣泛任務上表現出色。然而也要注意，參數數量並非唯一決定因素，模型性能還取決於架構創新和訓練數據品質等，但總體而言更大的參數規模帶來了更強的表現潛力**（遵循所謂的「Scaling Laws」)​

[hackthefuture.com](https://www.hackthefuture.com/news/how-llms-are-trained-and-how-red-teaming-helps#:~:text=The%20first%20step%20is%20to,vast%20datasets%2C%20like%20common%20crawl)

。

* **B 的含義：B 是 Billion（十億）的縮寫。例如“百億參數”表示模型有100億個可調權重。模型參數在計算中以浮點數表示，常用 FP16(半精度) 格式儲存，每個參數約佔2位元組記憶體​**

[**unfoldai.com**](https://unfoldai.com/gpu-memory-requirements-for-llms/#:~:text=,2%20bytes)

**。因此可以粗略估算模型所需記憶體：例如 1750億參數 \* 2 bytes ≈ 350 GB 顯示記憶體（VRAM）​**

[**unfoldai.com**](https://unfoldai.com/gpu-memory-requirements-for-llms/#:~:text=Examples%3A)

**。如此龐大的記憶體需求遠非單一 GPU 所能滿足（一般單張高階 GPU 有數十GB VRAM）。實務中，大型模型的權重會分散存放在多張 GPU** 上，透過分布式並行計算來共同承載。此外也常使用更低精度的參數（如 INT8 量化，每參數1 byte）以減少記憶體佔用。

* **訓練硬體需求：訓練一個 B 級參數的 LLM 是極其耗費資源的工程。需要大量 GPU 或 TPU 協同運算、多週甚至數月的時間以及專門的分布式訓練框架。以 OpenAI GPT-3 為例，據報導其訓練使用了一個包含 1萬顆 GPU 的超級計算機集群​**

[**developer.nvidia.com**](https://developer.nvidia.com/blog/openai-presents-gpt-3-a-175-billion-parameters-language-model/#:~:text=%E2%80%9CThe%20supercomputer%20developed%20for%20OpenAI,companies%20stated%20in%20a%20blog)

**（由微軟Azure提供），總計處理了約 3000 億個訓練標記​**

[**developer.nvidia.com**](https://developer.nvidia.com/blog/openai-presents-gpt-3-a-175-billion-parameters-language-model/#:~:text=ImageFigure%201,Source%3A%20OpenAI)

**。如此龐大的計算，預估花費數百萬美元的電力和雲資源。為了高效訓練，研究人員採用資料並行**（將大語料庫分片給不同GPU各自計算梯度）和**模型並行/張量並行**（將模型不同部分分配給不同GPU同步運算）等技術。同時需要高速網路互連（例如 GPU 間 400Gb/s 連接）來減少通信瓶頸​

[developer.nvidia.com](https://developer.nvidia.com/blog/openai-presents-gpt-3-a-175-billion-parameters-language-model/#:~:text=%E2%80%9CThe%20supercomputer%20developed%20for%20OpenAI,companies%20stated%20in%20a%20blog)

。硬體上，NVIDIA A100、H100 或 Google TPU v4 等是常用的加速器。總之，**沒有分布式計算，就無法在合理時間內訓練出當今規模的 LLM**。

* **推理與部署需求：即使在模型訓練完畢後，要部署LLM進行推理（Inference）依然對硬體有較高要求。首先是記憶體：模型需要載入到 GPU 的 VRAM 中才能運算。以 FP16 為例，一個 13B 參數的模型約需 26GB VRAM​**

[**unfoldai.com**](https://unfoldai.com/gpu-memory-requirements-for-llms/#:~:text=Examples%3A)

**；175B 模型理論上需350GB，通常必須使用多張 GPU 分擔。​**

[**unfoldai.com**](https://unfoldai.com/gpu-memory-requirements-for-llms/#:~:text=Examples%3A)

**此外在處理長序列輸入時，還需額外的快取記憶體儲存注意力機制的 Key-Value 值，使得記憶體佔用隨上下文長度線性增加​**

[**unfoldai.com**](https://unfoldai.com/gpu-memory-requirements-for-llms/#:~:text=Key)

**。這意味著長文本推理可能進一步吃緊顯存。為了降低部署門檻，工程上常用 模型壓縮和優化 技術：如權重量化（將參數壓縮為8-bit甚至4-bit表示，大幅節省記憶體）、蒸餾精簡模型（訓練一個較小模型模仿大模型行為）等。同時也可採用分批推理**或非同步請求等手段，提高硬體的利用率。需要指出的是，**計算速度**也是考量，越大的模型單步推理越慢，為達到實時應用的要求，往往需要集群併發或更快的硬體（如新一代GPU、專用AI加速晶片）。綜上，LLM 在訓練和推理階段都對硬體有極高需求，大規模的參數讓它們擁有強大能力的同時，也帶來計算與部署成本的挑戰。

**6. LLM 與 CNN、RNN、傳統 ANN 的差異比較**

大型語言模型（LLM，以 Transformer 架構為主）與傳統的卷積神經網路 (CNN)、遞歸神經網路 (RNN)、以及傳統前饋神經網路（ANN）在設計理念和應用領域上有明顯差異：

* **應用領域：LLM (Transformer) 主要應用於自然語言處理** (NLP) 任務，例如文本生成、翻譯、對話、問答等，近年也拓展到多模態領域（如結合圖像文本的模型）。CNN 則主要應用在**計算機視覺**領域，如圖像分類、物體檢測、影像分割等任務，因其擅長處理圖像這類具有空間結構的資料。RNN 傳統上用於**序列數據**處理，如語音識別、機器翻譯（早期系統）、時間序列預測等，需要考慮資料的時間順序關係。傳統 ANN（如多層感知機 MLP）一般用於**結構化數據**或簡單任務，例如表格資料分析、模式分類等，缺乏對時序或空間結構的特殊處理能力。
* **資料類型：LLM 專門針對文字序列**資料，在長文本上表現優異，也可經過擴充處理程式碼、甚至DNA序列等其他“語言化”序列資料。CNN 以處理**格狀網格資料**（典型為影像像素矩陣）見長，它能從局部圖像區域提取特徵，因而也被應用在視頻（連續幀視作多通道圖像）或語音的時頻譜圖表示上。RNN 處理**時間序列**數據，例如文字序列（在 Transformer 出現前的主流方法）、連續的聲學特徵序列等。RNN 也被用於解析手寫筆跡、一系列金融數據等序列訊號。傳統 ANN 則多處理**固定長度向量輸入**（比如若干手工提取的特徵），不直接適用可變長的序列或大尺寸影像。
* **架構設計：LLM/Transformer 採用自注意力機制**和完全並行的架構。每一層透過計算序列中各詞對其他詞的注意力分數來聚合資訊，並通過多頭注意力擴展表達力​

[elastic.co](https://www.elastic.co/what-is/large-language-models#:~:text=Transformer%20models%20work%20with%20self,a%20sentence%2C%20to%20generate%20predictions)

。這讓 Transformer 能在**全局範圍**建模詞與詞之間的關係，而且計算可以向量化並行處理​

[kolena.com](https://www.kolena.com/guides/transformer-vs-rnn-4-key-differences-and-how-to-choose/#:~:text=Transformers%20use%20parallel%20processing%20to,scale%20parallel%20computational%20power)

。相較之下，**CNN** 的架構基於**卷積層**：使用幾何鄰域的卷積核提取局部特徵，再經過**池化層**降維抽象，逐層擴大感受野以捕獲全局模式​

[aws.amazon.com](https://aws.amazon.com/tw/compare/the-difference-between-deep-learning-and-neural-networks/#:~:text=CNN%20%E6%9C%89%E4%B8%89%E5%80%8B%E5%B1%A4%E7%BE%A4%E7%B5%84%EF%BC%9A)

。CNN 的參數透過權值共享大幅減少，適合擷取平移不變的特徵，這在影像處理上非常有效。**RNN** 的架構由**遞迴單元**組成，序列中的每個時間步都接收上一步的隱藏狀態作為記憶，通過循環連接來保留歷史資訊​

[aws.amazon.com](https://aws.amazon.com/tw/compare/the-difference-between-deep-learning-and-neural-networks/#:~:text=RNN%20%E7%9A%84%E6%9E%B6%E6%A7%8B%E5%8F%AF%E4%BB%A5%E8%A6%96%E8%A6%BA%E5%8C%96%E7%82%BA%E4%B8%80%E7%B3%BB%E5%88%97%E9%81%9E%E6%AD%B8%E5%96%AE%E5%85%83%E3%80%82)

。典型的 RNN（如 LSTM、GRU）有機制解決長序列的梯度消失，能在一定範圍內記住先前輸入。然而 RNN **順序處理**的天性使其難以並行，長序列下效率較低​

[kolena.com](https://www.kolena.com/guides/transformer-vs-rnn-4-key-differences-and-how-to-choose/#:~:text=Transformers%20use%20parallel%20processing%20to,scale%20parallel%20computational%20power)

。傳統 **ANN**（前饋神經網路）則由多層全連接線性變換和非線性激活組成，相鄰層神經元全互連​

[aws.amazon.com](https://aws.amazon.com/tw/compare/the-difference-between-deep-learning-and-neural-networks/#:~:text=%E5%9C%A8%E7%B0%A1%E5%96%AE%E7%A5%9E%E7%B6%93%E7%B6%B2%E8%B7%AF%E4%B8%AD%EF%BC%8C%E4%B8%80%E5%B1%A4%E4%B8%AD%E7%9A%84%E6%AF%8F%E5%80%8B%E7%AF%80%E9%BB%9E%E6%9C%83%E9%80%A3%E7%B7%9A%E8%87%B3%E4%B8%8B%E4%B8%80%E5%B1%A4%E4%B8%AD%E7%9A%84%E6%AF%8F%E5%80%8B%E7%AF%80%E9%BB%9E%E3%80%82%E5%8F%AA%E6%9C%89%E4%B8%80%E5%80%8B%E9%9A%B1%E8%97%8F%E5%B1%A4%E3%80%82)

。它沒有顯式的機制處理序列順序或空間位置，因而對於語言或圖像這類結構化輸入往往效果不佳，但用於簡單分類回歸問題時架構更簡潔、易於訓練。

* **模型規模與訓練難度：LLM 往往是參數超大**的模型（數十億以上參數），依賴於超大規模資料訓練，訓練成本非常高。然而大規模帶來了前所未有的表現（例如 GPT-3/4 展現出跨任務的能力）。CNN 和 RNN 的模型規模相對中等，一般從幾百萬到幾億參數（例如 ResNet 圖像網路約數千萬參數，典型語音辨識 RNN 模型幾千萬參數）。它們的訓練可在數顆到數十顆 GPU 上完成。傳統 ANN 規模通常更小，可在單GPU甚至CPU上訓練。**擴展性**方面，Transformer 顯示出更好的可擴展性——隨著資料和參數增加，性能持續提升​

[kolena.com](https://www.kolena.com/guides/transformer-vs-rnn-4-key-differences-and-how-to-choose/#:~:text=Transformers%20scale%20well%20with%20increased,training%20corpora%20and%20model%20capacity)

——這點是 RNN 隨規模增大後逐漸力不從心的。CNN 在增加深度和寬度後也能提升效果（深度學習浪潮證明了非常深的CNN在視覺領域的成功），但它主要受限於任務資料量。ANN 則因缺乏針對複雜結構的設計，一味增大規模收效有限。

* **性能與現狀：目前在 NLP 領域，LLM (Transformer) 已全面取代 RNN** 成為主流方案。Transformer 由於可以同時考慮整個句子的所有詞而非依序處理，對長文本理解和生成遠勝於RNN​

[kolena.com](https://www.kolena.com/guides/transformer-vs-rnn-4-key-differences-and-how-to-choose/#:~:text=Transformers%20handle%20dependencies%20and%20memory,position%20in%20the%20input%20sequence)

。而在計算機視覺領域，CNN 佔據主導，不過最近也出現**Vision Transformer (ViT)** 等將 Transformer 引入影像分析的趨勢，並取得與CNN相當甚至更好的效果，說明Transformer的通用性。在一些情況下，Transformer、CNN、RNN 也會混合使用，例如視頻分析可能結合 CNN 提取每幀特徵，再用 Transformer 建模跨幀關係。傳統 ANN 現今多用於小型任務或作為其他模型的基礎組件（如 MLP 層），獨立在複雜AI任務中已較少見。總的來說，LLM/Transformer 以其靈活的注意力機制和強大表達能力，在處理**大規模語言任務**上有明顯優勢，而 CNN、RNN 則在各自主攻的領域繼續發揮作用，但受到 Transformer 不斷擴展應用範圍的挑戰。

**7. LLM 的優點與限制**

**LLM 的優點：**

* **強大的生成和理解能力：經過海量語料學習，LLM 具備生成流暢連貫文本**的能力，可以模仿各種文體風格進行創作，從寫小說劇本到生成程式碼無所不精​

[en.wikipedia.org](https://en.wikipedia.org/wiki/Large_language_model#:~:text=A%20large%20language%20model%20,a%20vast%20amount%20of%20text)

。同時，它對語義和上下文有深入的掌握，能理解複雜的問題並給出合適的回答。例如 GPT-4 等模型在許多標準化測試中達到接近人類專家的水準​

[en.wikipedia.org](https://en.wikipedia.org/wiki/GPT-4#:~:text=match%20at%20L482%20GPT,In%20contrast%2C%20OpenAI%20claims%20that)

。這種強大的NLP能力使其在對話、寫作、問答等任務上表現遠勝以往的模型。

* **遷移學習與通用性：LLM 在預訓練時已學到通用的語言表示**，因此可以通過少量範例或指示完成新任務（即**零樣本/少樣本學習**）。用戶只需透過 prompt 描述任務或給出幾個例子，模型便能舉一反三地解題，而不需要針對每個任務專門訓練新模型​

[elastic.co](https://www.elastic.co/what-is/large-language-models#:~:text=tuned%20so%20that%20they%20can,AI%20assistants%2C%20and%20so%20on)

。此外，LLM 經過適當的微調可以快速**遷移**到特定領域，這較傳統模型方便許多。它們也往往是**多才多藝**的，同一模型可勝任翻譯、總結、寫詩、答題等多種應用，不像過去必須為每個NLP任務訓練獨立模型，這在實用上大大提高了效率。

* **大規模知識儲備：由於訓練資料涵蓋維基百科、百科全書、新聞等，LLM 累積了廣博的知識**。它能回答各領域的常見問題，擁有相當於百科知識庫的內容廣度。因此在無額外資料庫支援下，LLM 也可以直接回覆許多一般知識問答，對使用者而言就像一個懂得大量資訊的智能助理。一些研究還指出，隨著模型參數和訓練數據規模增長，LLM 會出現**涌現能力**（Emergent Abilities），能解決只有更大模型才展現的新型任務​

[hackthefuture.com](https://www.hackthefuture.com/news/how-llms-are-trained-and-how-red-teaming-helps#:~:text=The%20first%20step%20is%20to,vast%20datasets%2C%20like%20common%20crawl)

。這意味著大模型可以自行「萌生」出一些小模型所不具備的技能，拓展了AI的應用邊界。

* **方便的微調和個性化：對產業和研究者來說，LLM 提供了一個強大的基礎，可在其上進行微調**以打造客製化應用。由於預訓練成本高昂，大家通常使用開源或商用已有的大模型，然後在小規模專門數據上微調，使模型適應特定領域（如法律、醫療）或特定風格（如客服禮貌用語）。這種基於 LLM 的開發范式降低了AI應用的門檻。另外，一些 LLM 支援**可控生成**，透過提示或特殊標記控制輸出風格內容，讓創作更加定制化。總體而言，LLM 作為通用 AI 引擎具有靈活性和可塑性，可滿足不同用戶需求。

**LLM 的限制：**

* **幻覺現象 (Hallucination)：LLM 著名的缺陷之一是可能產生看似合理但事實上錯誤的內容**，這種情況被稱為模型的“幻覺”​

[en.wikipedia.org](https://en.wikipedia.org/wiki/Large_language_model#:~:text=phenomenon%20which%20has%20been%20termed,117)

。例如模型可能自信地捏造一個不存在的引用、虛構不真實的事件或給出錯誤的數據。這是因為 LLM 生成文字只根據語言模式，並不直接連結真實世界的事實來源。當模型不確定答案時，往往仍會編造一個聽起來很 plausibe 的回應而非坦誠地回答“不知道”。幻覺現象在關鍵場合可能造成誤導或錯誤決策，限制了 LLM 在需要高可靠性的應用中的直接使用。為減輕此問題，常採取結合資料庫查證、加入不確定性估計或人類審核等機制。

* **偏見與歧視 (Bias)：LLM 從龐大的網路文本中學習，不可避免地繼承了數據中的偏見與不精確**​

[en.wikipedia.org](https://en.wikipedia.org/wiki/Large_language_model#:~:text=The%20largest%20and%20most%20capable,3)

。如果訓練語料對某些群體存在刻板印象或歧視性言論，模型生成內容時可能會反映出這些偏見。同樣，模型對政治、宗教、性別等議題的表達也可能帶有潛在傾向。這在應用中可能導致不公平或冒犯性的結果。例如先前出現過模型對不同種族給出迥異風格回答的爭議事件。為了減緩偏見問題，模型開發者在資料過濾和後處理上投入許多精力，但完全消除偏見非常困難。此外，LLM 往往**缺乏常識約束**，可能在道德倫理判斷上出現反常行為，需要透過人類偏好微調（如 RLHF）來引導模型遵循正確價值觀。

* **知識時效性：LLM 的知識庫停留在訓練數據的時間點**。也就是說，除非特別設計，模型不會自動更新對世界的認識。例如 GPT-3 的訓練數據截至 2021 年，因此對於2022年後發生的事件它一無所知。如果直接詢問，它可能會給出過時或錯誤的資訊。這對於需要處理**最新資訊**的任務（如實時新聞問答）是一大限制。為解決此問題，一些應用結合了**檢索模組**（讓模型在網路/資料庫搜尋最新內容）或採用**持續微調**（定期用新資料更新模型）。但即使如此，由於 LLM 沒有連接外部實時信息源的內建機制，用戶在使用時必須意識到它回答的時效局限，對涉及最新動態的問題保持警惕並交叉驗證。
* **推理和數學能力有限：雖然 LLM 在表面語言任務上表現卓越，但在嚴格邏輯推理、數學計算**等方面仍有不足。模型可能在涉及多步推理的問題上前後矛盾，或者在簡單算術計算上出錯（例如對較大的數字進行加減乘除常出錯）。這是因為傳統 LLM 並非專門為精確推理設計，欠缺類似程式執行的機制。不過新的研究正嘗試改進，例如引入**鏈式思考 (Chain-of-Thought)** 提示讓模型分步推理，或將模型與符號計算模組結合。即便如此，目前遇到複雜推理任務時，LLM 的表現仍不如專用的推理系統或人類專家。使用時需要小心核對推理過程，重要場合可能需輔以外部驗算。
* **計算資源與成本：如前所述，LLM 的龐大規模意味著部署與調用成本高**。在雲端調用大型模型的API服務費用不菲，而自行部署需要昂貴的硬體。對於資源有限的終端裝置，如手機或邊緣設備，大模型難以直接運行。不僅如此，LLM 推理速度相對較慢，生成長篇文本可能有明顯延遲。這些都限制了 LLM 在一些即時或本地場景的應用。不過隨著硬體進步和壓縮技術的發展，這一限制在逐步緩解，例如出現了一些輕量化的7B參數模型可以在筆記本電腦上運行。但總體而言，**高成本**仍是LLM的一大現實限制，用戶在考慮應用時需要權衡利弊。
* **黑箱與可解釋性：LLM 屬於深度學習模型，內部決策機制對人類而言如同黑箱**。它為何生成某句話、某個答案，很難給出具備直觀意義的解釋。缺乏可解釋性可能降低用戶對模型的信任度，特別是在醫療、法律等高風險領域，人們希望知道AI為何做出某結論。目前一些研究致力於探究 LLM 的內部表示，或者從輸出追溯重要的提示詞，但距離完全可解釋仍有很大距離。在實踐中，通常透過**大量測試**來增進對模型行為的理解，用策略約束模型輸出，以避免災難性後果。隨著模型應用愈廣，提升 AI 決策透明度將變得愈發重要。

**8. 現今主流 LLM 模型介紹**

當前（2023-2025）已有多款具代表性的 LLM 問世，以下介紹幾個主流模型及其特點與開發者：

* **GPT-4 (OpenAI)：由 OpenAI 開發的 GPT-4 是 GPT 系列的第4代模型，於2023年發布。GPT-4 是一個封閉商業模型，精確的參數量未公布（據傳數千億以上），但性能遠超前代 GPT-3。GPT-4 最大的亮點之一是其多模態**能力：它不僅能處理文本輸入，還能理解圖像，對圖片內容進行描述和分析​

[openai.com](https://openai.com/index/gpt-4-research/#:~:text=GPT%E2%80%914%20is%20a%20large%20multimodal,)

（例如，解讀一張諷刺漫畫或根據照片回答問題）。在語言任務上，GPT-4 展現出接近專家水準的實力：OpenAI 提供的測試顯示，它在法律律師資格考（Bar Exam）中名列前10百分位，在研究生入學考(GRE)寫作部分取得99百分位的成績，在許多知識問答基準上刷新紀錄​

[en.wikipedia.org](https://en.wikipedia.org/wiki/GPT-4#:~:text=match%20at%20L482%20GPT,In%20contrast%2C%20OpenAI%20claims%20that)

。GPT-4 支援更長的上下文（OpenAI 提供長達 8K 和 32K token 兩個版本），能處理超長文檔。開發者主要透過 OpenAI 的 API 或 ChatGPT 服務使用 GPT-4，其應用包括高級問答、創意寫作、代碼生成、語言翻譯等。需要注意的是，GPT-4 作為封閉模型，其內部架構細節和訓練數據並未公開，但已知採用了類似 GPT-3 的 Transformer 解碼器架構並結合了 RLHF 等調教手段以提升對齊人類意圖的程度。

* **Claude (Anthropic)：Claude 是由新創公司 Anthropic 開發的系列大模型，致力於成為一個安全、可控的 AI 助手**。Claude 的開發團隊成員多來自 OpenAI，Anthropic 強調透過「憲法式 AI (Constitutional AI)」來引導模型行為，以減少有害輸出。最新版本 **Claude 2** 發布於2023年7月，據推測其參數規模超過1300億​

[textcortex.com](https://textcortex.com/post/claude-2-parameters#:~:text=TL%3BDR)

。Claude 2 的一大特色是**超長上下文**能力：支援高達 100K tokens 的輸入輸出長度​

[textcortex.com](https://textcortex.com/post/claude-2-parameters#:~:text=,of%20documents%20with%20one%20click)

（約相當於75,000字），這意味著它可以一口氣讀入一本小說或長報告並進行分析摘要。Anthropic 的測試顯示 Claude 善於處理長文檔的問答和總結​

[anthropic.com](https://www.anthropic.com/news/100k-context-windows#:~:text=We%E2%80%99ve%20expanded%20Claude%E2%80%99s%20context%20window,for%20hours%20or%20even%20days)

。Claude 在編碼、數學推理等方面也有增強，比前代更擅長寫程式和解題​

[textcortex.com](https://textcortex.com/post/claude-2-parameters#:~:text=Claude%202%20is%20a%20large,3)

。Anthropic 提供 Claude 的API服務，並在網站 claude.ai 上開放試用。Claude 定位是 ChatGPT 的有力競爭者，在一些基準測試上表現接近 GPT-4 水準，同時在避免不當內容上更為謹慎。其**開發理念**強調 AI 系統的可靠性與倫理邊界，例如Anthropic透過內部規則（“AI憲法”）自動調教模型，使其回答更符合人類價值觀。Claude 目前主要以英文能力見長，已被整合到一些商業客戶服務與文檔分析應用中。

* **LLaMA (Meta)：LLaMA 是 Meta（臉書）AI 研究部門於2023年推出的開放大型語言模型**系列。名字 “LLaMA” 代表 Large Language Model Meta AI。第一版 LLaMA 提供了 7B、13B、33B、65B 等不同規模的模型，僅供研究用途；隨後 Meta 在2023年7月發布了 **LLaMA 2**，參數規模擴展至 7B、13B 和 **70B**​

[ai.meta.com](https://ai.meta.com/research/publications/llama-2-open-foundation-and-fine-tuned-chat-models/#:~:text=We%20develop%20and%20release%20Llama,billion%20to%2070%20billion%20parameters)

三種，並採用較寬鬆的開源許可（允許商業使用）。LLaMA 系列模型在相同規模下達到了當時最先進性能，例如 LLaMA 13B 的表現可比肩 GPT-3 175B 在許多基準上，凸顯出資料和訓練技巧的重要性。LLaMA 開源後，引發了社群的極大熱情，許多開發者基於它進行微調，衍生出各種應用模型（對話、編碼、角色扮演等）。LLaMA 的架構本質上是 Transformer 解碼器（類GPT架構），訓練使用大量網路文本以及 Meta 自有資料集。**開發者**方面，LLaMA 由 Meta 团队開發，目的在於促進學術和產業對大型模型的研究，降低研究門檻。LLaMA 2 公開了模型權重，並提供了與人類和AI對話資料微調的 Chat 版。在性能上，LLaMA 2 70B 接近當時最頂尖的封閉模型能力，一些開源社群的測試顯示其問答和推理能力僅略遜於 GPT-4。LLaMA 的出現標誌著**開源社群掌握了大型模型**，許多後續的開源模型（如 Alpaca、Vicuna 等對話模型）都是以它為基礎。對企業而言，LLaMA 提供了一條不依賴外部API、可私有部署大模型的道路，在隱私和定制性上具有吸引力。

* **Mistral (Mistral AI)：Mistral 是一家歐洲初創公司 Mistral AI（成立於2023年）推出的開源大型模型。Mistral 7B 是其首個公開模型，在2023年9月發布，具有 73億 個參數。雖然參數遠小於動輒百億的巨模型，但 Mistral 7B 的性能令人矚目：根據官方聲明，它在多項基準測試上超越了 LLaMA 2 的 13B 模型**，甚至接近某些 30B 級模型的表現​

[mistral.ai](https://mistral.ai/news/announcing-mistral-7b#:~:text=Mistral%207B%20is%20a%207,parameter%20model%20that)

。這得益於優化的訓練工藝和高品質的訓練數據，以及一些架構改進（如引入 Grouped-Query Attention 等技術以提升推理效率​

[mistral.ai](https://mistral.ai/news/announcing-mistral-7b#:~:text=,longer%20sequences%20at%20smaller%20cost)

）。Mistral 7B 發布時採用 **Apache 2.0 開源許可**，允許任何人無限制使用​

[mistral.ai](https://mistral.ai/news/announcing-mistral-7b#:~:text=We%27re%20releasing%20Mistral%207B%20under,can%20be%20used%20without%20restrictions)

。這使其迅速被社群接受並運用在各類應用中，包括對話聊天、開源助手等。Mistral AI 團隊計劃後續推出更大規模的模型（如 30B 等）以延續性能提升。同時，他們也提供了經微調的聊天版模型，在對話任務上表現突出​

[mistral.ai](https://mistral.ai/news/announcing-mistral-7b#:~:text=Mistral%207B%20is%20easy%20to,outperforms%20Llama%202%2013B%20chat)

。Mistral 的出現代表新創團隊也能在大型模型領域與科技巨頭競爭，通過創新達到**小模型大性能**的效果。在硬體需求上，7B 模型相對輕量，很適合部署在單機 GPU 上，這進一步推動了它的普及。總而言之，Mistral 7B 強調了優化和資料的重要性，證明經精心訓練的小型 LLM 也能滿足許多應用需求，是開源 LLM 生態的一顆新星。

* **Gemini (Google DeepMind)：Gemini 是 Google DeepMind 在2023年宣布研發的下一代大型模型。《Google I/O 2023》上，CEO Sundar Pichai 將 Gemini 定位為 PaLM 2 的強力升級後繼者​**

[**en.wikipedia.org**](https://en.wikipedia.org/wiki/Gemini_(language_model)#:~:text=Google%20announced%20Gemini%2C%20a%20large,CEO%20Sundar%20Pichai%20stating%20that)

**。Gemini 的獨特之處在於其從設計之初就是多模態**的，不僅訓練於文本，同時結合了圖像、視訊、音訊甚至程式碼等多種資料​

[en.wikipedia.org](https://en.wikipedia.org/wiki/Gemini_(language_model)#:~:text=Gemini%20was%20still%20in%20its,OpenAI%20%27s%20%2099%2C%20which)

。另外，DeepMind CEO Demis Hassabis 提到，Gemini 將結合 DeepMind 在 AlphaGo 等系統中的長處，把類AlphaGo的規劃推理能力融入 LLM​

[en.wikipedia.org](https://en.wikipedia.org/wiki/Gemini_(language_model)#:~:text=believed%20would%20allow%20the%20algorithm,5)

，目標是在**推理能力**上超越 OpenAI 的 GPT-4。同年12月，Google 正式發布 **Gemini 1.0**，包括不同尺寸的模型：如 “Gemini Ultra”（超大型，用於最複雜任務）、“Gemini Pro”（中型，用於廣泛任務）和“小型的 Gemini Nano”​

[en.wikipedia.org](https://en.wikipedia.org/wiki/Gemini_(language_model)#:~:text=On%20December%206%2C%202023%2C%20Pichai,116%2C%20Duet%20AI)

。Gemini Pro 已整合進 Google 的 Bard 聊天服務，Nano 則用於 Pixel 8 手機上的本地AI功能，而最大的 Ultra 計劃提供給企業與開發者使用​

[en.wikipedia.org](https://en.wikipedia.org/wiki/Gemini_(language_model)#:~:text=On%20December%206%2C%202023%2C%20Pichai,116%2C%20Duet%20AI)

。Gemini 展現了強大的能力：在代碼、數學、推理等評測上均取得領先，其多模態 API 可處理實時的語音、影像輸入。在後續發展中，Google 持續改進 Gemini，例如在2024年推出 Gemini 2 系列，引入“Thinking”模型概念，讓 AI 有明確的中間推理步驟，並據報支援多達百萬 token 級別的上下文窗口。**開發者**為 Google DeepMind（由原 Google Brain 和 DeepMind 團隊合併），Gemini 的推出是 Google 對ChatGPT熱潮的強力回應，旨在重新取得AI技術領先地位。由於 Google 擁有龐大的產品生態，未來我們將會在搜尋引擎、廣告、工作空間等各種服務中看到 Gemini 的身影。作為使用者，可以期待更加**多才多藝且智能**的 Google 助手式應用問世。