**自主學習報告 – LLM(Large Language Model)初探**

|  |  |  |
| --- | --- | --- |
| 參與者:吳庭佑 | 地點:北大高中 | 時間:2025/3/15~2025/4/1 |
| 操作方式:實驗操作 | | |
| 簡述:透過對LLM的研究，我重新審視了自己整理資料、設計實驗、找到解決方法的過程。無論是接下來大學的專題、論文的撰寫，這次的過程都將成為我的經驗和養分，而不用在大學端重新學習。 | | |

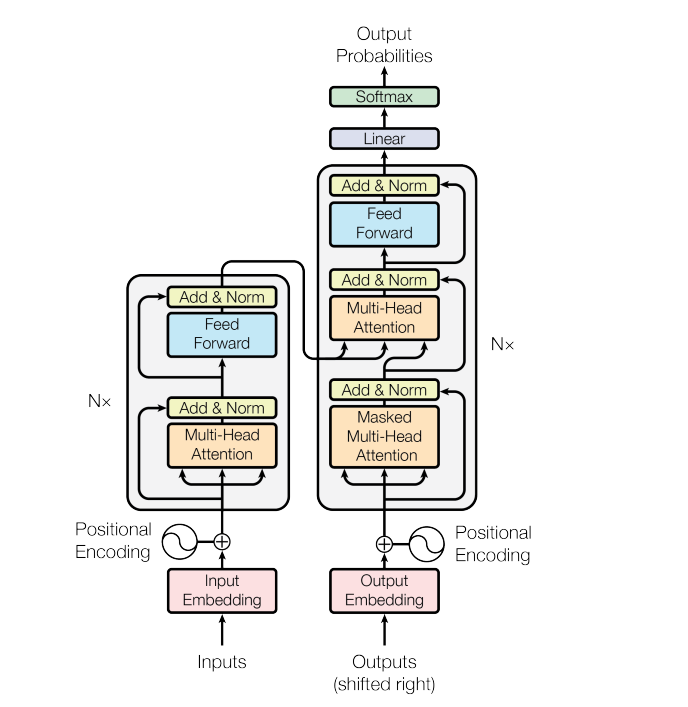
1. **研究動機**

生成式AI正逐漸滲透進每一個人的生活中，對業界、學界皆帶來一定的衝擊。所以為了和AI可以做到更好的協作，我想研究生成式AI的架構和實作生成式AI，並最後希望能轉化成自己可以使用的AI工具。

1. **背景知識**

**Transformer**

由<<Attention Is All You Need>>[[1]](#footnote-1)這篇論文所提出，其使用Encoder-decoder結構[[2]](#footnote-2):encoder將輸入(input)的序列(標註為連續的序列，而decoder根據序列生成輸出(output)序列，而模型的這些步驟皆擁有自回歸(auto-regressive)的特性，也就是將前個時間步的輸出作為下個時間步的輸入。



The Transformer – 模型結構

**Encoder**

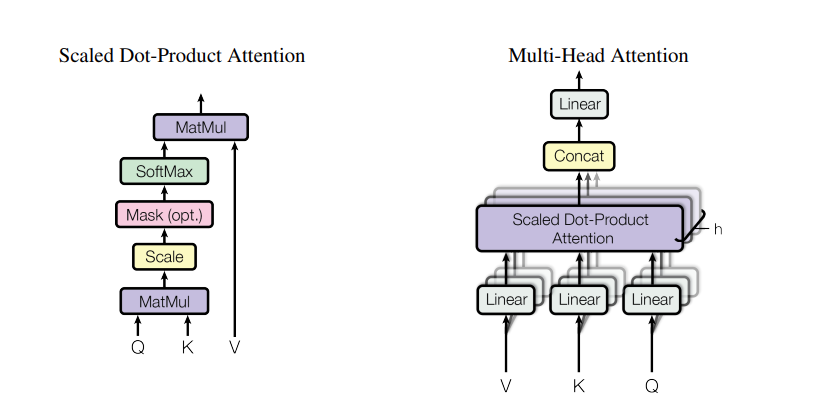
在<<Attention Is All You Need>>這篇論文中，Transformer的Encoder使用層相同的編碼器(identical layers)組成，每層都有兩個子層(sub-layers)組成。第一子層是一個Multi-Head Self-Attention，第二子層是一個簡單的Position-wise Fully Connected Feed-forward Network，子層和子層間使用Residual Connection後接著Layer Normalization，意味著每個子層的輸出為。為了使Residual Connection可行，所有層的輸出維度必須相同，因此固定。

**Decoder**

在<<Attention Is All You Need>>這篇論文中，Transformer的Decoder同樣使用層相同的編碼器組成，每層除了原本的兩個子層，更加入了第三個子層Multi-performs Attention，加入這層的原因是為了防止Decoder的子層提前注意到了還未解讀的內容，子層和子層間同樣使用Residual Connection後接著Layer Normalization。

**Attention**

這是一種函數，它會根據一個query vector和一組key-value pairs，產生一個output vector。Output vector是通過values的加權總合得到，而每個值的加權由compatibility function得到。



**Scaled Dot-Product Attention**

是Key向量的維度，是以query和所有keys做dot-product得到的結果，矩陣Q是矩陣包裝的注意力函數，K為keys vector，V為values vector。

**Multi-Head Attention**

是計算注意力的次數，意味著它有個head，使這個模型可以關注到不同表現的資訊。每個head都有自己的線性投影矩陣，將原始維的query、key和value映射到更小的維度，則以上過程可以用以下矩陣表示:

**Position-wise Feed-Forward Networks with ReLU**

FNN的輸入和輸出維度，而內部隱藏層的維度。

**Embeddings and Softmax**

Embedding是將Input tokens和Output tokens轉換成向量的技術，而Softmax則是將Output vector轉換成機率的函數。Transformer採用了一樣的參數矩陣在Embedding Matrix和Pre-Softmax Linear Transformation中，並將Embedding的權重乘上。

**Positional Encoding**

在<<Attention Is All You Need>>這篇論文中，因為Transformer模型沒有RNN或CNN結構，所以需要PE來讓模型理解序列的順序。於是Transformer模型在Input embeddings的最後加入了Positional encodings，來讓模型知道位置的關係。在Transformer模型中使用sine和cosine函數達成這個目的:

pos是位置，i是維度。

**BERT**

引用資料:

<https://haren.medium.com/paper-notes-bert-bert-%E6%9E%B6%E6%A7%8B%E7%90%86%E8%A7%A3-31c014d7dd63>

<https://arxiv.org/pdf/1810.04805>

全名:Bidirectional Encoder Representations from Transformers

使用Masked LM(MLM)和Next Sentence Prediction(NSP)捕捉token-level和sentence-level的特徵。訓練方式為透過遮住某個詞語來讓模型預測被遮住的詞語。

**GPT**

引用資料:

<https://reurl.cc/mxGN0A>

全名:Generative Pre-trained Transformer

GPT僅採用Transformer decoder的部分，透過前面的文字序列來預測下一個單詞，或透過周圍的語境來推斷缺失單詞的意思。

**PaLM**

引用資料:

<https://arxiv.org/pdf/2204.02311>

全名:Pathways Language Model

PaLM採用完整的Transformer架構，並使用Pathways架構訓練，使其可以同時學習多種不同的任務。

**LLaMA**

引用資料:

<https://ithelp.ithome.com.tw/m/articles/10366983>

全名:Large Language Model Meta AI

LLaMA僅採用Transformer decoder的部分，相比於Transformer的LayerNorm，它使用RMSNorm，且使用更加平滑的SwiGLU函數作為激活函數。LLaMA是開源的，每個人都可以訓練自己的模型。

**AutoMathText**

引用資料: <https://arxiv.org/pdf/2402.07625>

此資料集是一個經過精心整理的資料集，包含約200GB的數學文字。這個資料集來自不同平台，包含各種網站、arXiv和GitHub(OpenWebMath、RedPajama、Algebraic Stack)等。這個豐富的資料庫是由最先進的開放原始碼語言模型 Qwen-72B 所自主選擇 (標籤)。資料集中的每篇內容都會在 [0, 1] 的範圍內分配一個分數 lm\_q1q2\_score，以反映其在數學智慧方面的相關性、品質與教育價值。

1. **實驗操作**

為了方便訓練模型，我選擇在Kaggle平台上訓練模型，Kaggle平台提供了免費的GPU資源，且不需要自行安裝虛擬環境來訓練模型，並提供簡便的接口來接入輸入(Input)和輸出(Output)，

1. https://doi.org/10.48550/arXiv.1706.03762 [↑](#footnote-ref-1)
2. https://doi.org/10.48550/arXiv.1406.1078 [↑](#footnote-ref-2)