**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。 | | |

目錄

[一、 研究動機 2](#_Toc194439117)

[二、 背景知識 2](#_Toc194439118)

[三、 實驗操作 8](#_Toc194439119)

[四、 實驗結果 12](#_Toc194439120)

[五、 實驗結果檢討 15](#_Toc194439121)

[六、 未來展望 16](#_Toc194439122)

[七、 心得與反思 16](#_Toc194439123)

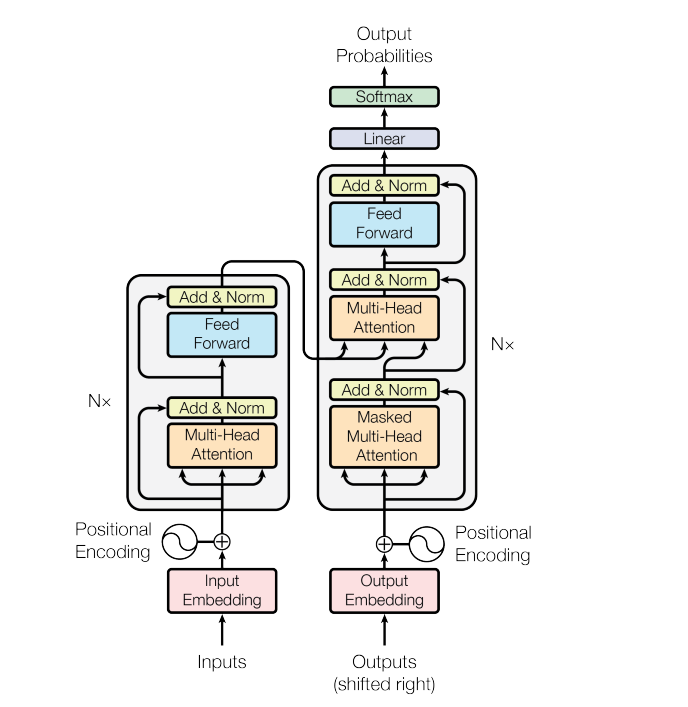
1. 研究動機

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**

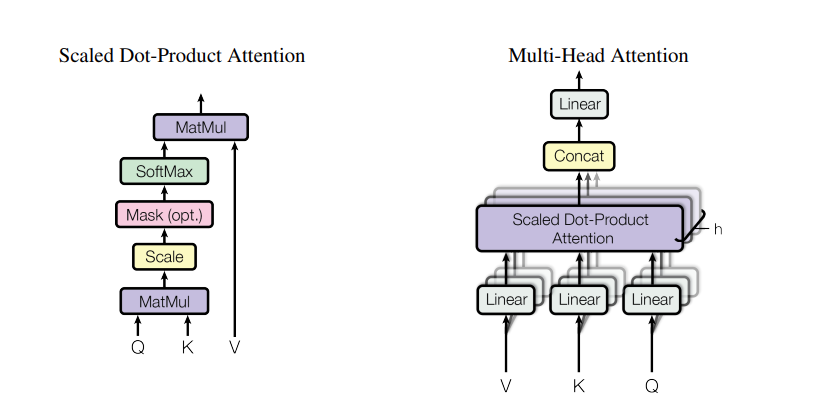
在<<Attention Is All You Need>>這篇論文中，Transformer的Encoder使用層相同的編碼器(identical layers)組成，每層都有兩個子層(sub-layers)組成。第一子層是一個Multi-Head Self-Attention，第二子層是一個簡單的Position-wise Fully Connected Feed-forward Network，子層和子層間使用Residual Connection後接著Layer Normalization，意味著每個子層的輸出為。為了使Residual Connection可行，所有層的輸出維度必須相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

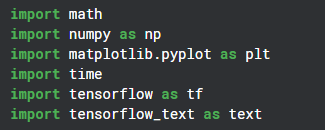
此資料集是一個經過精心整理的資料集，包含約200GB的數學文字。這個資料集來自不同平台，包含各種網站、arXiv和GitHub(OpenWebMath、RedPajama、Algebraic Stack)等。這個豐富的資料庫是由最先進的開放原始碼語言模型 Qwen-72B 所自主選擇 (標籤)。資料集中的每篇內容都會在 [0, 1] 的範圍內分配一個分數 lm\_q1q2\_score，以反映其在數學智慧方面的相關性、品質與教育價值。

1. 實驗操作

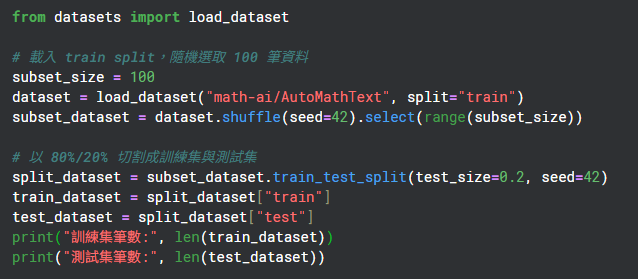
為了方便訓練模型，本次研究選擇在Colab平台上訓練模型，Colab平台提供了免費的GPU資源，亦提供安裝好的虛擬環境，意味著使用者不需要自行設定虛擬環境便可使用。

而訓練的模型和數據則從Hugging Face上取得，Hugging Face整合了大量的模型的API和訓練數據，以方便使用者操作。

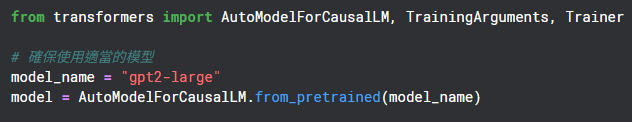
本次實驗使用GPT2模型進行操作。首先將要用的模組引入:



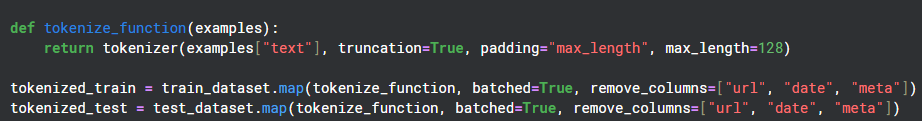
取得數據集:



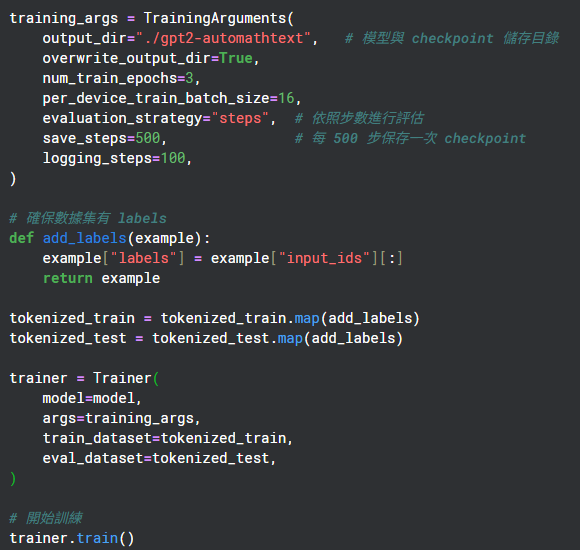
引入模型:



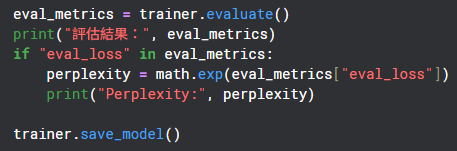
設定tokenizer:



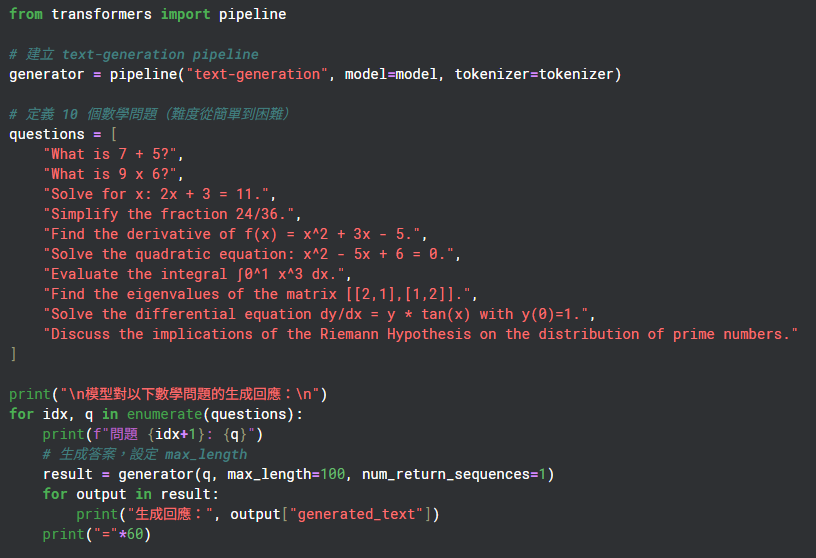
訓練參數設置:



評估和儲存模型:



用模型生成數學題目的回覆:



1. 實驗結果

本次研究總共設定了10個數學問題來測試訓練過後的GPT2模型，難度從簡易到難都有，但除了第一個問題有實際回答出來以外，其他的回覆皆沒有得到想要的結果:

模型對以下數學問題的生成回應：

問題 1: What is 7 + 5?

生成回應： What is 7 + 5?

7 + 5 = 14

============================================================

問題 2: What is 9 x 6?

生成回應： What is 9 x 6?

The first question you should ask is, "what is 9 x 6?"

8 x 9 = 60

10 x 9 = 60

11 x 9 = 60

So, " 9 x 6 " is the ratio of a number to the number of digits.

9 x 6 = 60 x 10 = 60

9 × 6 = 60

============================================================

問題 3: Solve for x: 2x + 3 = 11.

生成回應： Solve for x: 2x + 3 = 11.

============================================================

問題 4: Simplify the fraction 24/36.

生成回應： Simplify the fraction 24/36.

============================================================

問題 5: Find the derivative of f(x) = x^2 + 3x - 5.

生成回應： Find the derivative of f(x) = x^2 + 3x - 5.

============================================================

問題 6: Solve the quadratic equation: x^2 - 5x + 6 = 0.

生成回應： Solve the quadratic equation: x^2 - 5x + 6 = 0.

To find the solution of the quadratic equation x^2 - 5x + 6 = 0, substitute the value of at into the equation.

============================================================

問題 7: Evaluate the integral ∫0^1 x^3 dx.

生成回應： Evaluate the integral ∫0^1 x^3 dx. I think it is a polynomial. Any help is appreciated.

============================================================

問題 8: Find the eigenvalues of the matrix [[2,1],[1,2]].

生成回應： Find the eigenvalues of the matrix [[2,1],[1,2]].

The eigenvalues are the diagonal matrix elements.

============================================================

問題 9: Solve the differential equation dy/dx = y \* tan(x) with y(0)=1.

生成回應： Solve the differential equation dy/dx = y \* tan(x) with y(0)=1.

Solution 1.

Take the gradient of the function y(0) by setting dx = 0.

============================================================

問題 10: Discuss the implications of the Riemann Hypothesis on the distribution of prime numbers.

生成回應： Discuss the implications of the Riemann Hypothesis on the distribution of prime numbers.

============================================================

經過整理，我們可以發現訓練過後的GPT2有以下兩種情況:

1. 生成錯誤的回覆(可能是答案錯誤，或者解答是亂碼)
2. 重複問題

種種跡象都表明，GPT2並沒有在這次訓練中學習到回答以上數學題的能力，或者它似乎沒有真正理解這是一個「問題」，以至於生成出的內容並不符合我們的期待，亦顯示這絕對是一次糟糕的訓練。

1. 實驗結果檢討

根據實驗結果顯示:這是一次糟糕的訓練。我認為可以統整出以下三個問題。

第一個問題是，這次實驗的時間並不夠長，我沒有足夠的時間可以調整參數，才剛看完資料就必須開始實驗了，也來不及測試其他的模型是否可以勝任這些任務，這導致無法控制實驗的變量，亦沒有明顯的對照組，或許我可以在時間相對充裕的情況下進行實驗。

第二個問題是，餵給模型的資料過少，導致模型還沒學到足夠的資訊就被迫中止訓練，或許我可以在下次實驗的時候調大餵給模型的數據量。

第三個問題是，經費不足以提升設備效能，也導致這次實際訓練中時常模型還沒訓練完就被平台中止了，未來如果有機會完整實驗的話，我想經費會是一個很大的問題。

1. 未來展望

我還是希望可以訓練一個自己的模型出來，無論是未來工作上或大學學業上的使用，都有助於我完成被指定的任務，但首先我要解決以下問題:

* 訓練時間過短
* 訓練資料過少
* 經費不足

1. 心得與反思

雖然這次研究LLM的時間很短，但我還是學到了不少，以下是我學到的能力及其摘要:

|  |  |
| --- | --- |
| 能力 | 摘要 |
| 資料的整理能力 | 從原文論文中提取重要資料並用自己的言語描述 |
| 發現問題和解決問題 | 對程式debug、解決實驗設計中隱含的問題 |
| 程式設計的能力 | 調用API和訓練模型 |
| 文書處理工具 | 利用不同的工具如Latex來描述實驗資訊 |

透過這次研究的經驗，讓我發現我還有很多需要學習的地方，從實驗設計的精確度到對資源的掌握度，這些都是需要靠我自己去完善的目標，而我也勢必要在進入大學前解決這些我固有的問題，才有助於我的未來發展。

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