

深入淺出 Google Gemma LLM 模型

Simon Liu

Google GDG Activity - Build With AI
2024/05/30



About me: 劉育維 / Simon Liu

- AI, ML, DL, LLM Architect and Engineer / Technical Writer / Speaker
- Try to use AI to do the application and help to solve the pain point by AI methods.
- My Personal Information:
 - Link: <https://simonliuyuwei-4ndgcf4.gamma.site/>



Today's Topic

1 Gemma 模型介紹

2 如何使用 Gemma 模型

Colab 1: 如何透過 KerasNLP, Ollama 來啟動 Google Gemma LLM 模型

3 透過 sentence transformer 和 Google Gemma 模型, 進行 RAG 應用

Colab 2: 透過 sentence transformer 和 Google Gemma 模型, 進行 RAG 應用

4 如何透過 LoRA 在 Keras Fine-Tune 中文資料 (Optional)

Colab 3: 如何透過 LoRA Fine-Tune 方法, 來Fine-Tune CKIP 來啟動 Google Gemma LLM 模型

5 結語

演講專用警語



技術隨時在變化

請依照工具提供的官方資訊為主，我已盡力做好所有查核處理工作。

Part 1

Gemma 模型介紹

先問問看，大家是否玩過 **Gemma LLM** 模型呢？



Gemma LLM 模型的自我介紹

- 第一次釋出日期: Feb 21, 2024
- 模型名稱: Gemma 系列模型
 - 此模型由 Google 官方所開源出來的 SOTA AI 模型
- 模型開源狀況 / License:
 - Gemma 目前採用 Google 所撰寫的 License 授權方式 – Gemma Terms of Use, 統整相關內容並理解後, 是一個可商用的模型。

目前 Gemma 種類

Gemma v1

- 高效能且輕量化的大型語言模型
- 主要可做對話式大型語言模型
- 各項評估上表現優良

RecurrentGemma

- 支援研究人員進行大批次的高效推理，採用循環神經網路和局部注意力機制提升記憶效率。
- 基準測試上成績與Gemma 2B模型相當，但是RecurrentGemma使用的記憶體量更少

CodeGemma

- 針對開發人員和企業的程式碼完成、生成和聊天工具使用情境
- 多程式語言能力，主要以Python、JavaScript、Java等各種熱門程式語言的程式碼撰寫建議

PaliGemma

- 視覺語言開放模型，能夠針對影像字幕、視覺問答、理解圖像內文字、物件偵測、物件切割的應用案例提供最佳化

都已經有了 Gemini 模型，為何要釋出 Gemma 模型？

- 社群回饋機制
- 易於使用和獲得
- 可供研究人員和開發人員用於各種目的，包括：
 - a. 探索新的 LLM 應用
 - b. 開發新的 LLM 技術
 - c. 使 LLM 更易於獲取

Gemma 目前的模型能力比較

	Meta Llama 3 8B	Meta Llama 3 70B	Meta Llama 3 400B+	Meta Llama 2 7B	Meta Llama 2 70B	Anthropic Claude 3 Opus	Anthropic Claude 3 Sonnet	Anthropic Claude 3 Haiku	OpenAI GPT-4	OpenAI GPT-3.5	Google Gemma 7B-it	Google Gemini 1.0 Ultra	Google Gemini 1.0 Pro	Google Gemini 1.5 Pro	MistralAI Mistral 7B Instruct	MistralAI Mistral large
Open Source / Close Source	Open Source	Open Source	-	Open Source	Open Source	Close Source	Close Source	Close Source	Close Source	Close Source	Open Source	Close Source	Close Source	Close Source	Open Source	Close Source
MMLU (5-shot)	68.4	82.0	86.1	34.1	52.9	86.8	79.0	75.2	86.4	70.0	53.3	83.7	71.8	81.9	58.4	81.2
GPQA (0-shot)	34.2	39.5	48.0	21.7	21.0	50.4 (0-shot CoT)	40.4 (0-shot CoT)	33.3 (0-shot CoT)	35.7 (0-shot CoT)	28.1 (0-shot CoT)	21.4	-	-	41.5 (0-shot CoT)	26.3	
HumanEval (0-shot)	62.2	81.7	84.1	7.9	25.6	84.9	73.0	75.9	67.0	48.1	30.5	74.4	67.7	71.9	36.6	45.1
GSM-8K (8-shot, CoT)	79.6	93.0	94.1	25.7	57.5	95.0	92.3	88.9	92.0 (5-shot CoT)	57.1 (5-shot)	30.6	94.4 (Maj1@32)	86.5 (Maj1@32)	91.7	39.9	
MATH (4-shot, CoT)	30.0	50.4	57.8	3.8	11.6	60.1 (0-shot CoT)	43.1 (0-shot CoT)	38.9 (0-shot CoT)	52.9	34.1	12.2	53.2	32.6	58.5	11.0	

Note: The information is for reference only. If there are any errors in the information, please refer to the official information provided.

Reference :

- Llama benchmark: <https://llama.meta.com/llama3/>
- Anthropic Claude 3 benchmark: <https://www.anthropic.com/news/claude-3-family>
- Introducing Meta Llama 3: <https://ai.meta.com/blog/meta-llama-3/>

Gemma 目前的模型能力比較 (擷取重點)

	Meta Llama 3 8B	Meta Llama 2 7B	OpenAI GPT-4	Google Gemma 7B-it	Google Gemini 1.5 Pro	MistralAI Mistral 7B Instruct
Open Source / Close Source	Open Source	Open Source	Close Source	Open Source	Close Source	Open Source
MMLU (5-shot)	68.4	34.1	86.4	53.3	81.9	58.4
GPQA (0-shot)	34.2	21.7	35.7 (0-shot CoT)	21.4	41.5 (0-shot CoT)	26.3
HumanEval (0-shot)	62.2	7.9	67.0	30.5	71.9	36.6
GSM-8K (8-shot, CoT)	79.6	25.7	92.0 (5-shot CoT)	30.6	91.7	39.9
MATH (4-shot, CoT)	30.0	3.8	52.9	12.2	58.5	11.0

Part 2

如何使用 Gemma 模型

你有幾種方式使用 Gemma

套件程式碼撰寫



寫程式碼，讓模型按照套件方式啟動

相關工具啟用



透過工具啟動服務，透過 API 來使用

KerasNLP

- 專門用於自然語言處理(NLP)的 Keras 擴展庫
- 提供了一系列的工具和模組來簡化和加速NLP 任務的開發。

Quickstart

Fine-tune BERT on a small sentiment analysis task using the `keras_nlp.models` API:

```
import os
os.environ["KERAS_BACKEND"] = "tensorflow" # Or "jax" or "torch"!

import keras_nlp
import tensorflow_datasets as tfds

imdb_train, imdb_test = tfds.load(
    "imdb_reviews",
    split=["train", "test"],
    as_supervised=True,
    batch_size=16,
)

# Load a BERT model.
classifier = keras_nlp.models.BertClassifier.from_preset(
    "bert_base_en_uncased",
    num_classes=2,
)

# Fine-tune on IMDB movie reviews.
classifier.fit(imdb_train, validation_data=imdb_test)

# Predict two new examples.
classifier.predict(["What an amazing movie!", "A total waste of my time."])
```

Gemma Model in KerasNLP

Google AI for Developers

Gemini API

Gemma

Google AI 邊緣

工具

社群

搜尋結果

中文 - 繁體

登入

總覽

文件

篩選器

Gemma 模型系列

- CodeGemma
- PaliGemma
- RecurrentGemma

版本

指南

- Gemma 設定
- 透過 Keras 開始使用 Gemma
- 使用 Keras 使用 Gemma 進行基本調整
- 使用 Keras 使用 Gemma 進行分散式調整
- 透過 PyTorch 開始使用 Gemma
- 與 Gemma 聊天
- 格式和系統操作說明
- Gemma C++ 教學課程
- 使用 JAX 和 Flax 推論
- 使用 JAX 和 Flax 微調

首頁 > Gemma > 文件

這對你有幫助嗎?

提供意見

透過 KerasNLP 開始使用 Gemma

在 Google Colab 中執行

在 Vertex AI 中開啟

在 GitHub 上查看原始碼

本教學課程說明如何透過 [KerasNLP](#) 開始使用 Gemma。Gemma 是一系列先進的輕量開放模型，採用與建立 Gemini 模型相同的研究和技術打造而成。KerasNLP 是一組在 [Keras](#) 中實作的自然語言處理 (NLP) 模型，可在 JAX、PyTorch 和 TensorFlow 上執行。

在這個教學課程中，您將使用 Gemma 產生多則提示的文字回應。如果您是 Keras 新手，建議您先閱讀「[開始使用 Keras](#)」一文，然後再進行後續步驟。逐步進行本教學課程時，將會進一步瞭解 Keras。

設定

Gemma 設定

如要完成本教學課程，您必須先在 [Gemma 設定](#) 中完成設定。Gemma 設定操作說明會說明如何執行下列操作：

- 在 [kaggle.com](#) 上存取 Gemma。
- 請選取具備足夠資源的 Colab 執行階段，才能執行 Gemma 2B 模型。
- 產生並設定 Kaggle 使用者名稱與 API 金鑰。

這個頁面中的內容

設定

- Gemma 設定
- 設定環境變數
- 安裝依附元件
- 選取後端
- 匯入套件

建立模型

- 生成文字
- 選用：改用其他取樣器

後續步驟

https://ai.google.dev/gemma/docs/get_started?hl=zh-tw

Ollama

- Ollama 是一個開源軟體，讓使用者可以在自己的硬體上運行、創建和分享大型語言模型服務。
- 這個平台適合在地端運行模型，因為它不僅可以保護隱私，還允許使用者透過命令行介面輕鬆地設置和互動。
- Ollama 支援非常多種模型，並提供彈性的客製化選項，例如從其他格式導入模型並設置參數。



Gemma Model in Ollama

[Blog](#)[Discord](#)[GitHub](#)

Gemma is available in both **2b** and **7b** parameter sizes:

- `ollama run gemma:2b`
- `ollama run gemma:7b` (default)

gemma

Gemma is a family of lightweight, state-of-the-art open models built by Google DeepMind. Updated to version 1.1

2B **7B**

↓ 1.6M Pulls ⌚ Updated 6 weeks ago

7b

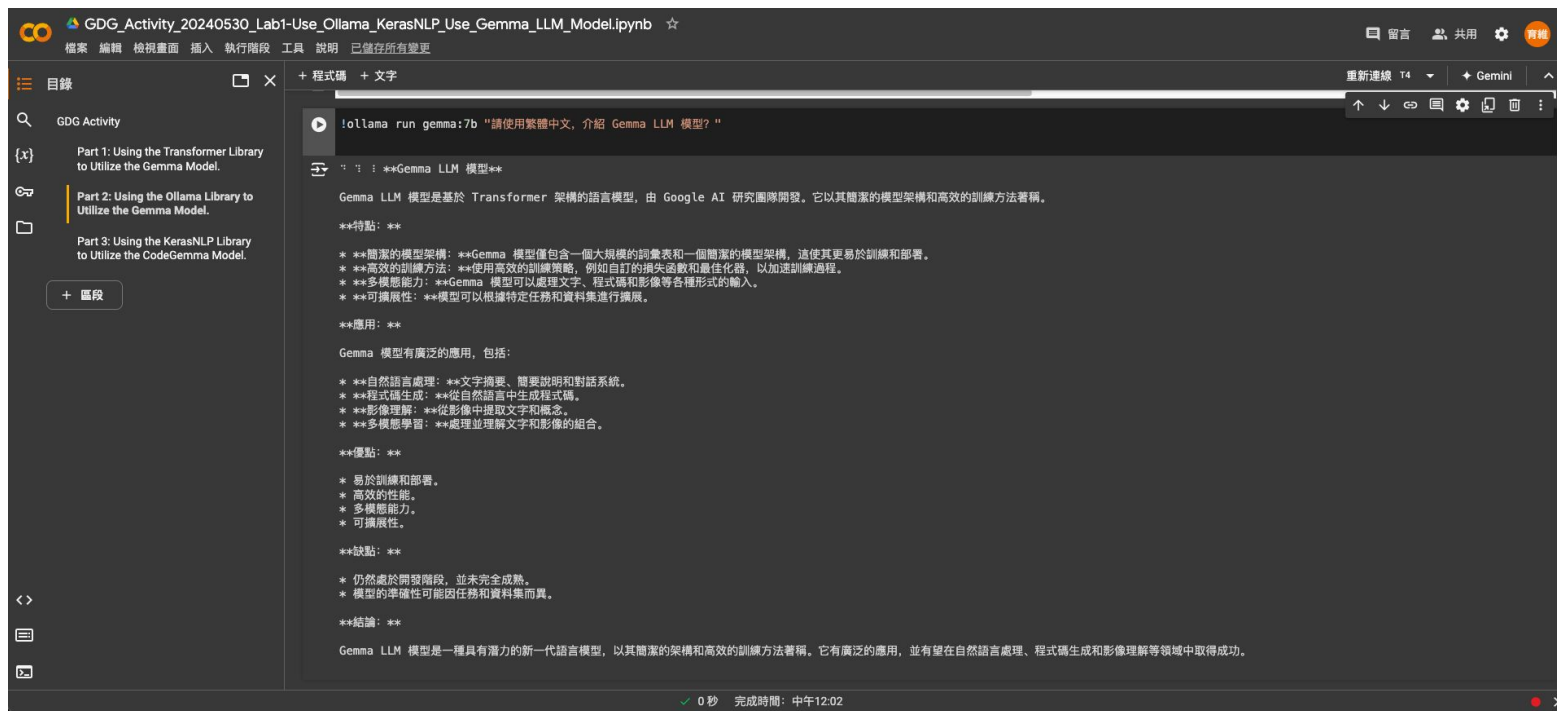
102 Tags

ollama run gemma

Updated 6 weeks ago		a72c7f4d0a15 · 5.0GB
model	arch gemma · parameters 8.5B · quantization Q4_0	5.0GB
license	Gemma Terms of Use Last modified: February 21, 2024 By usi_	8.4kB
template	<start_of_turn>user {{ if .System }}{{ .System }} {{ end }}_	136B
params	{"penalize_newline":false,"repeat_penalty":1,"stop":["<sta_	109B

<https://ollama.com/library/gemma>

Colab 1: Use KerasNLP and Ollama to run Gemma Model



Colab 連結 :https://colab.research.google.com/github/LiuYuWei/GDG-Activity-Gemma-20240530/blob/main/GDG_Activity_20240530_Lab1-Use_Ollama_KerasNLP_Use_Gemma_LLM_Model.ipynb

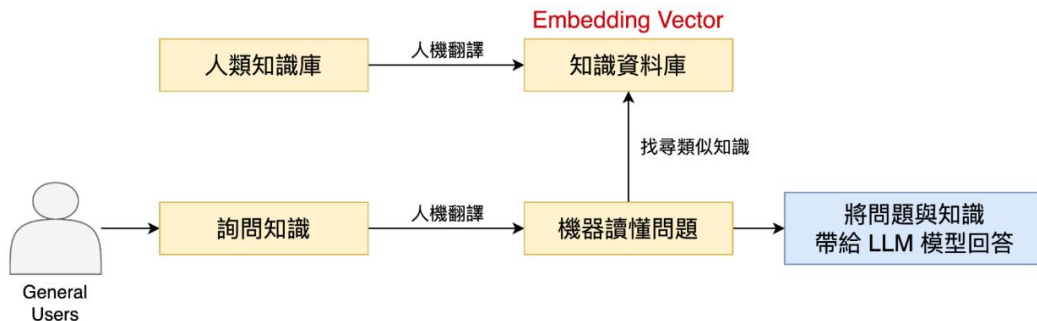
Part 3

透過 sentence transformer 和 Google Gemma 模型, 進行 RAG 應用

RAG / Fine-Tune

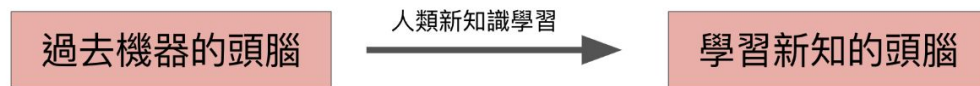
RAG (Embedding)

當人類詢問問題時，找尋說明書了解知識後，再回覆問題。



Fine-Tune Model

類似小孩子學習新知的概念，經過學習，就能夠得到新知。



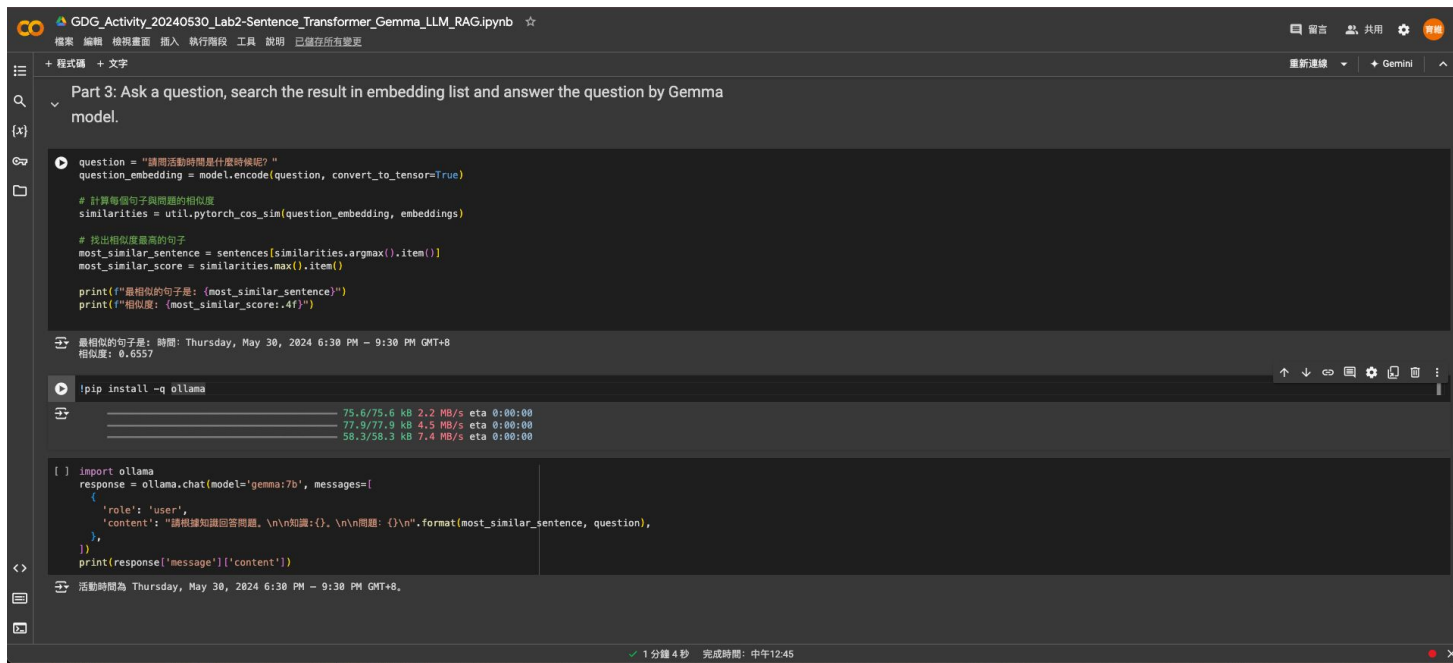
RAG 和 Fine-Tune 比較

	Fine-Tune 微調模型	RAG (Embedding)
比喻	就像考試前認真讀書，考試 closed book 去回答考試題目。	就像考試 open book，帶筆記去考試，若筆記上有寫可以回答的很好
缺點	訓練模型需要花時間和計算成本，不可能隨時訓練更新資料	仍有 Token 長度限制 要用工具抓資料因此處理時間較長
優點	品質可能更好，這需要機器學習專業知識	不用擔心新資料更新

Sentence Transformer

1. Sentence Transformer 是一種專為句子和段落 Embedding 而設計的模型，它能夠以高效的方式計算句子的向量表示。這使得它在多種 NLP 任務（例如語意相似度計算和文本檢索）中表現出色。
2. RAG + Sentence Transformer 所帶來的好處
結合 RAG 和 Sentence Transformer 可以實現一個高效且準確的自然語言處理框架。具體來說，可以使用 Sentence Transformer 進行高效的知識庫搜尋檢索，並將檢索到的知識庫作為 RAG 模型的輸入，讓大型語言模型可以生成具有高度相關性和準確性的回答或文本。這種結合方式使得在大規模數據集上的文本檢索和生成更為快速和準確。

Colab 2: 透過 Sentence Transformer 和 Google Gemma 模型，進行 RAG 應用



The screenshot shows a Google Colab notebook titled "GDG_Activity_20240530_Lab2-Sentence_Transformer_Gemma_LLM_RAG.ipynb". The notebook is in the "Part 3: Ask a question, search the result in embedding list and answer the question by Gemma model." section. The code defines a question, encodes it, calculates similarities with a list of sentences, finds the most similar sentence, and then uses the Gemma LLM to generate an answer based on the retrieved sentence and the original question. The output shows the most similar sentence and its similarity score, followed by the installation of the llama package and the final LLM response.

```
question = "請問活動時間是什麼時候?"
question_embedding = model.encode(question, convert_to_tensor=True)

# 計算每個句子與問題的相似度
similarities = util.pytorch_cos_sim(question_embedding, embeddings)

# 找出相似度最高的句子
most_similar_sentence = sentences[similarities.argmax().item()]
most_similar_score = similarities.max().item()

print(f"最相似的句子是: {most_similar_sentence}")
print(f"相似度: {most_similar_score:.4f}")
```

最相似的句子是: 時間: Thursday, May 30, 2024 6:30 PM - 9:30 PM GMT+8
相似度: 0.6557

```
!pip install -q ollama
```

```
[ ] import ollama
response = ollama.chat(model='gemma:7b', messages=[
    {
        'role': 'user',
        'content': "請根據知識回答問題。\\n\\n知識: {}。\\n\\n問題: {}\\n".format(most_similar_sentence, question),
    },
])
print(response['message']['content'])
```

活動時間為 Thursday, May 30, 2024 6:30 PM - 9:30 PM GMT+8.

Colab 連結: https://colab.research.google.com/github/LiuYuWei/GDG-Activity-Gemma-20240530/blob/main/GDG_Activity_20240530_Lab2-Sentence_Transformer_Gemma_LLM_RAG.ipynb

Google Cloud 也支援 RAG 相關服務

- Google Cloud Vertex AI Vector Search
 - Google Official Doc: [Google Cloud Vertex AI Vector Search](#)
- Pgvector:
 - Google blog: [Building AI-powered apps on Google Cloud databases using pgvector, LLMs and LangChain](#)

Part 4

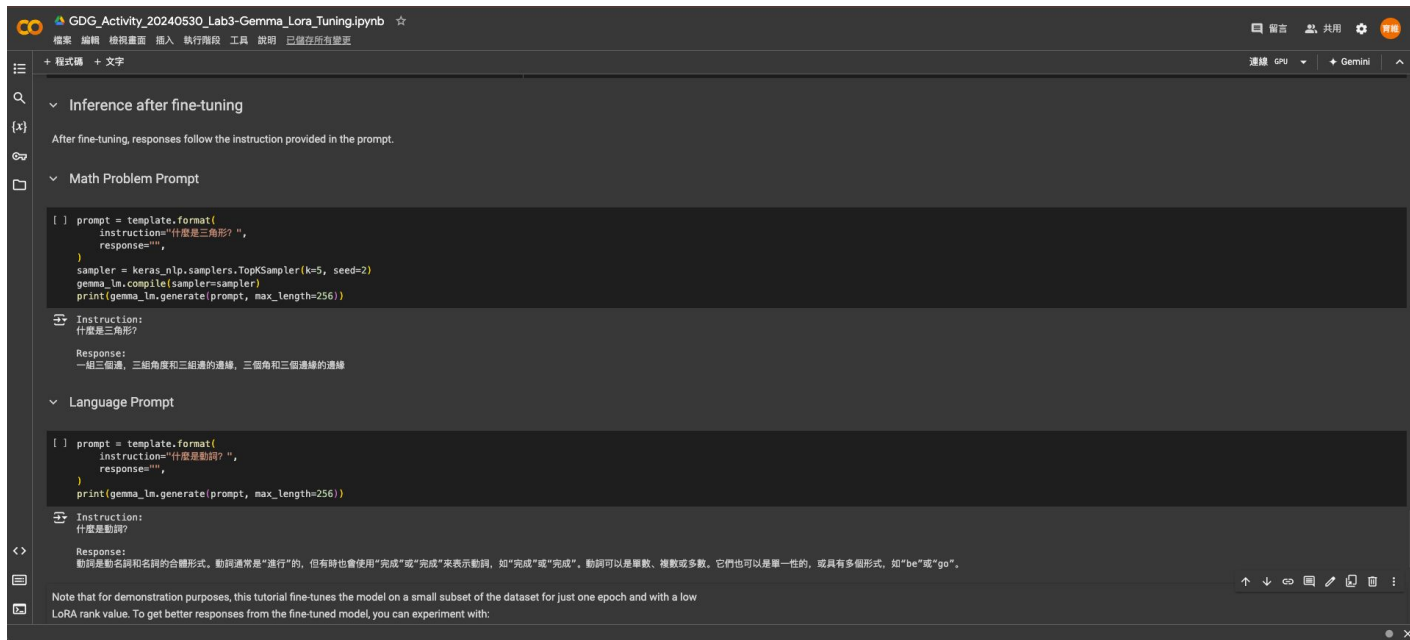
如何透過 LoRA 在 Keras Fine-Tune 中文資料

或許，你(可能)不需要 Fine-tune LLM

- 目前對於一般企業應用來說, Fine-tune 的商業價值還不夠高：
 - 大多數情境下其實用 Prompt Engineering (再加上 RAG) 就可以解決。
 - 微調需要準備訓練資料(至少100筆以上資料), 花費計算資源, 比起用 Prompting 成本跟難度要高
 - i. 用 prompting 迭代開發比用微調快很多, 因為後者需要建立數據和訓練時間
 - 建議先從 Prompting 開始看看能做到什麼程度
- 什麼時候需要微調?
 - a. 當用 Prompting 太花 tokens 數時, 微調後可以減少 few-shot 或指示文字
 - b. 當用 Prompting 需要的 inference time 太久
 - c. 當用 Prompting 對特定任務的結果不夠穩定可靠時, 例如輸出特定格式、風格語氣
 - d. 你有收集到有品質且輕易拿來用的資料集

Colab 3: 如何透過 LoRA 在 Keras Fine-Tune 中文資料

-> 此範例將不帶著大家做，大家可以活動結束後，運行此程式碼，如有任何問題，歡迎與我再交流！



```
GDG_Activity_20240530_Lab3-Gemma_Lora_Tuning.ipynb ☆
檔案 編輯 檢視畫面 插入 執行階段 工具 說明 已儲存所有變更

+ 程式碼 + 文字

Inference after fine-tuning
After fine-tuning, responses follow the instruction provided in the prompt.

Math Problem Prompt

[ ] prompt = template.format(
    instruction="什麼是三角形？",
    response="",
)
sampler = keras_nlp.samplers.TopKSampler(k=5, seed=2)
gemma_lm.compile(sampler=sampler)
print(gemma_lm.generate(prompt, max_length=256))

Instruction:
什麼是三角形？
Response:
一組三個邊，三組角度和三組邊的連線，三個角和三個連線的連線

Language Prompt

[ ] prompt = template.format(
    instruction="什麼是動詞？",
    response="",
)
print(gemma_lm.generate(prompt, max_length=256))

Instruction:
什麼是動詞？
Response:
動詞是動名詞和名詞的合體形式。動詞通常是“進行”的，但有時也會使用“完成”或“完成”來表示動詞，如“完成”或“完成”。動詞可以是單數、複數或多數。它們也可以是單一性的，或具有多個形式，如“be”或“go”。

Note that for demonstration purposes, this tutorial fine-tunes the model on a small subset of the dataset for just one epoch and with a low LoRA rank value. To get better responses from the fine-tuned model, you can experiment with:
```

Colab 連結: https://colab.research.google.com/github/LiuYuWei/GDG-Activity-Gemma-20240530/blob/main/GDG_Activity_20240530_Lab3-Gemma_Lora_Tuning.ipynb

Part 5

結語

結論

- Gemma 在六月份會迎來一次更新, 相信在能力上能夠與其他開源模型一拼
- 透過工具來啟動 Gemma 服務
 - Ollama / Llama-cpp / VLLM 等
 - KerasNLP / Transformer
- 透過 RAG 等方式, 讓產品化能夠做的更好
 - Sentence Transformer
 - 第三方工具
- Fine-Tune?
 - 成本高, 特殊條件下再思考是否做 Fine-Tune 處理。

Thanks for listening!

Simon Liu
Google GDG Activity
2024/05/30

