使用Hugging Face在本地端執行Florence-2-large

WZX
2 min read · Just now

microsoft/Florence-2-large · Hugging Face

We're on a journey to advance and democratize artificial intelligence through open source and open science.

huggingface.co

Module:

- `transformers` ：Hugging Face Transformers，提供 `AutoProcessor` 和 `AutoModelForCausalLM`
- `torch` ：PyTorch，深度學習框架，用於模型推理（`AutoModelForCausalLM`）
- `pillow` ：PIL（Python Imaging Library），用於處理圖片（`Image.open()`）
- `requests` ：用於從網路載入圖片（`requests.get()`）

```
from transformers import AutoProcessor, AutoModelForCausalLM
import torch
from PIL import Image
import requests
```

額外需要:

```
pip install einops timm
```

`einops`（高效Tensor操作）

`timm`（高级 PyTorch 視覺）

初始化模型

- 載入 Causal Language Model（CLM）自回歸語言模型
- 允許執行遠端 Python 代碼
- 根據模型權重自動選擇最佳的數據類型（如 `float16` 或 `float32`）
- 切換到推理模式
- 將模型放入 GPU 記憶體
- 載入 AutoProcessor (適用多模態模型)

```
def init_model():
    try:
        model_id = 'microsoft/Florence-2-large'
        model = AutoModelForCausalLM.from_pretrained(
            model_id,
            trust_remote_code=True,
            torch_dtype="auto"
        ).eval().to("cuda")
        processor = AutoProcessor.from_pretrained(
            model_id,
            trust_remote_code=True
        )
        return model, processor
    except Exception as e:
        print(f"模型初始化失敗: {e}")
        return None, None
```

載入照片

- 發送 HTTP 請求(以串流方式讀取，減少記憶體佔用)

- 檢查請求是否成功(如果 HTTP 狀態碼不是 `200 OK` ，則會拋出 `HTTPError` 異常)
- 確保圖片能正確解碼(`response.raw.decode_content = True` 會讓 `requests` 自動解壓縮，確保讀取到的內容是解壓縮後的原始數據)
- 打開圖片(`response.raw` 是請求的二進制數據流， `PIL.Image.open(response.raw)` 直接讀取)

```
def load_image(url):
    try:
        response = requests.get(url, stream=True)
        response.raise_for_status()
        response.raw.decode_content = True
        return Image.open(response.raw)
    except Exception as e:
        print(f"載入圖片失敗: {e}")
        return None
```

模型生成

- 組合提示詞
- 轉換輸入格式(使用 `processor` 將圖片和文字轉為 PyTorch tensor)
- 讓模型生成輸出

num_beams :

1貪婪搜索（Greedy Search），速度快但品質可能較差

3Beam Search（3 條路徑），平衡品質與速度

5更高品質的 Beam Search，但運算較慢

- 解碼生成的文本將 `generated_ids` 轉回人類可讀的文字，不跳過特殊 token (可改為 `True`)
- 後處理輸出

```
def run_example(model, processor, image, task_prompt, text_input=None):
    try:
        prompt = task_prompt if text_input is None else task_prompt + text_inpu
        inputs = processor(
            text=prompt,
            images=image,
            return_tensors="pt"
        ).to('cuda', torch.float16)

        generated_ids = model.generate(
            input_ids=inputs["input_ids"].cuda(), # 文字輸入
            pixel_values=inputs["pixel_values"].cuda(), # 圖片輸入
            max_new_tokens=1024, # 限制最大可生成的 token 數量
            early_stopping=False,  # 是否提前結束
            do_sample=False,  # 是否使用隨機取樣來產生不同的結果
            num_beams=3,
        )

        generated_text = processor.batch_decode(
            generated_ids, skip_special_tokens=False)[0]
        return processor.post_process_generation(
            generated_text,
            task=task_prompt,
            image_size=(image.width, image.height)
        )
    except Exception as e:
        print(f"執行過程發生錯誤: {e}")
        return None
```

Example :

task_prompt = '<MORE_DETAILED_CAPTION>'

url = "https://th.bing.com/th/id/R.87f818548092f71ec17f03286bdc5b3a?rik=PnqHpUnvuZPliw&pid=ImgRaw&r=0"

result:

{‘<MORE_DETAILED_CAPTION>’: ‘The image is a panoramic view of a city with a tall skyscraper in the center. The skyscraper is the Taipei 101, a famous landmark in Taipei, Taiwan. It is a tall, modern building with a unique design that stands out against the other buildings in the city. The building is made up of multiple levels, with the tallest building at the top and the smaller ones at the bottom. The sky is blue and clear, and the city is densely populated with other buildings and skyscrapers. In the background, there are mountains and a clear blue sky. The image is taken from a high vantage point, looking down on the city below.’}

Florence Hugging Face Python Deep Learning Pytorch

Edit profile

Written by WZX