REPORT MISURAZIONI

Indice

Introduzione

Nel seguente report vengono descritti hardware, modelli, librerie e prompt utilizzati al fine di produrre un dataset di misurazioni relative ai consumi di energia durante l'inferenza dei modelli LLM di seguito presentati.

1 Hardware

I cicli di inferenza sono stati eseguiti su hardware vario, nello specifico:

- Un PC (HP Pavillion Gaming) con le seguenti caratteristiche:
  - CPU Intel i5-10300H, 2.5GHz;
  - GPU NVIDIA GeForce 1650Ti;
  - 4GB di RAM GPU;
  - 16GB di TAM CPU;
- Macchina virtuale Google Colab T4:
  - CPU Intel Xeon R, 2.2GHz;
  - GPU NVIDIA Tesla T4;
  - 15GB RAM GPU;
  - 12GB RAM CPU;
- Macchina virtuale su HPC Leonardo:
  - TODO
- Eventuale altro hardware che utilizzerò

2 Modelli

Le diverse combinazioni di iperparametri e prompt che saranno descritte in seguito sono state utilizzate per misurare i consumi relativi all'inferenza nei seguenti modelli LLM:

2.1 Llama2

LLama2 è una famiglia di LLM open source. Nello specifico, per le misurazioni è stata utilizzata la versione 7B-chat caratterizzata da 7 miliardi di parametri (la versione più leggera) e un'ottimizzazione per i casi d'uso basati sul dialogo. E' stato poi necessario operare una quantizzazione, per le esecuzioni su hardware meno prestante, per cercare un giusto compromesso fra prestazioni e output.

2.2 Mistral

Mistral è un modello LLM disponibile nella sola versione 7B con 7.3 miliardi di parametri e completamente open-source. Per le misurazioni è stata utilizzata la versione 0.1. Anche in questo caso è stata operata una quantizzazione del modello.

2.3 Eventuali altri modelli

TODO

3 Prompt

Per le misurazioni sono stati individuati 3 casi d'uso:

1. Descrizione di un mondo fantastico: al modello è stata fornita la direttiva di descrivere un mondo di fantasia con 10 prompt sempre più ricchi di indicazioni;

2. Storytelling creativo: al modello è stato chiesto di raccontare una breve storia. Anche in questo caso sono stati forniti 10 input diversi, simili dal punto di vista delle indicazioni;

3. Coreografie di danza: al modello è stato chiesto di generare delle coreografie di danza come successioni di movimenti;

Di seguito i prompt selezionati.

3.1 Descrizione mondo fantasy

1. "Imagine you are in an imaginary world. Describe the fantastic setting and the adventures that can be experienced."

2. "Imagine being in a fictional world inhabited by both humans and mythical creatures. Describe the fantastic setting and the adventures that can be experienced."

3. "Imagine being in a fictional world inhabited by both humans and mythical creatures. Describe the fantastical setting, including enchanted landscapes, ancestral forests, and snow-capped mountains, and the adventures that can be experienced."

4. "Imagine being in a fictional world inhabited by both humans and mythical creatures. Describe the fantastical setting, including enchanted landscapes and mysterious places, and the adventures that can be experienced."

5. "Imagine being in a fictional world inhabited by both humans and mythical creatures. Describe the fantastical setting, including enchanted landscapes, ancestral forests populated by talking trees and magical creatures, and the adventures that can be experienced."

6. "Imagine being in a fictional world inhabited by both humans and mythical creatures. Describe the fantastical setting, including enchanted landscapes, ancestral forests populated by talking trees and magical creatures, and the adventures that can be experienced, such as searching for ancient hidden treasures or fighting against dark masters of evil."

7. "Imagine being in a fictional world inhabited by both humans and mythical creatures. Describe the fantastical setting, including enchanted landscapes, ancestral forests populated by talking trees and magical creatures, mysterious cities and haunted castles, and the adventures that can be experienced, such as exploring underground worlds or fighting giant dragons."

8. "Imagine being in a fictional world inhabited by both humans and mythical creatures. Describe the fantastical setting, including enchanted landscapes, ancestral forests populated by talking trees and magical creatures, mysterious cities and haunted castles, stormy seas and remote islands, and the adventures that can be experienced, such as finding the source of eternal life or defending the kingdom against an army of the undead."

9. "Imagine being in a fictional world inhabited by both humans and mythical creatures. Describe the fantastical setting, including enchanted landscapes, ancestral forests populated by talking trees and magical creatures, mysterious cities and haunted castles, stormy seas and remote islands, fiery deserts and frozen lands, and the adventures that can be experienced, such as the search for the divine artifact or the liberation of an entire enslaved race."

10. "Imagine being in a fictional world inhabited by both humans and mythical creatures. Describe the fantastical setting, including enchanted landscapes, ancestral forests populated by talking trees and magical creatures, mysterious cities and haunted castles, stormy seas and remote islands, fiery deserts and frozen lands, cloud-touching mountains and endless plains, and the adventures that can be experienced, such as the quest for the book of forbidden spells or the final battle against the ultimate evil."

3.2 Storytelling creativo

TODO

3.3 Coreografie di Danza

TODO

4 CodeCarbon

Per misurare i consumi è stata utilizzata la libreria codecarbon disponibile per python. In sostanza, dopo aver instanziato un oggetto di tipo EmissionsTracker, lo si avvia con il metodo start() e questo ogni 15 secondi eseguirà una piccola misura del consumo di energia (le metodologie di misura dei consumi le descriverò poi in altro documento). Nel pratico, si avvia un tracker in questo modo:

```
from codecarbon import EmissionsTracker
tracker=EmissionsTracker()

...
tracker.start()
try:
    outputs = model.generate(**inputs, **config)
finally:
    tracker.stop()
...
```

Listato 1: Misurazione con codecarbon

Il risultato delle misurazioni sarà poi salvato all'interno di un file .csv con i seguenti campi:

Campo	Descrizione
timestamp	Time stamp dell'esperimento in formato "%Y-%m-%dT%H:%M:%S"
project_name	Nome del progetto
run-id	id dell'esecuzione
duration	Durata dell'esecuzione, in secondi
emissions	Emissioni espresse in CO_2equivalente, in Kg
emissions_rate	emissioni/durata, in Kg/s
cpu_power	Potenza della CPU, in Watt
gpu_power	Potenza della GPU, in Watt
ram_power	Potenza della RAM, in Watt
cpu_energy	Energia consumata dalla CPU, in KWh
gpu_energy	Energia consumata dalla GPU, in KWh
ram_energy	Energia consumata dalla RAM, in KWh
energy_consumed	Somma dell'energia consumata da CPU, GPU e RAM
country_name	Nome della nazione in cui si sta eseguendo
country_iso_code	ISO code corrispondente alla nazione
region	Provincia/Stato/Città in cui si sta eseguendo
on_cloud	Y se si sta eseguento su cloud, N altrimenti
cloud_provider	Provider su cui si sta eseguendo (se on_cloud = Y)
cloud_region	Regione geografica del servizio di cloud computing
os	Sistema operativo in esecuzione sulla macchina
python_version	Versione di python installata sulla macchina
cpu_count	Numero di CPU
cpu_model	Nome del modello della CPU
gpu_count	Numero di GPU
gpu_model	Nome del modello della GPU
longitude	Longitudine (con precisione ridotta per ragioni di privacy)
latitude	Latitudine (con precisione ridotta per ragioni di privacy)
ram_total_size	Dimensione totale della RAM
Tracking_mode	Indica se si sta tracciando un processo o l'intera macchina

Tabella 1: Campi del file .csv

5 Struttura del Dataset

Come prima fase si è scelto di misurare i consumi relativi all'inferenza dei modelli così come sono stati preaddestrati dalle organizzazioni creatrici di questi ultimi. In fasi successive magari saranno valutati consumi relativi ad altri casi d'uso come ad esempio il fine-tuning dei parametri per adattare i modelli a domini applicativi più specifici. Il processo di raccolta e organizzazione del dataset si è articolato quindi nelle seguenti fasi:

- Selezione degli opportuni iperparametri che impattano in modo significativo su performance e tipo di output del modello;
- Scelta dei valori degli iperparametri (strategie di sampling);
- Esecuzione degli script;
- Valutazione degli output;

5.1 Iperparametri scelti

La scelta degli iperparametri, in seguito ad una revisione della letteratura sull'argomento (inserire le fonti), è ricaduta su 4 iperparametri che impattano sia sulle performance che sul livello di "creatività" e in generale di determinismo dell'output e cioè:

- **max_length**: impone al modello una lunghezza massima per l'output definita come lunghezza massima della stringa finale decodificata;
- **top_p**: calcola la probabilità cumulativa di una serie di token successivi selezionando la successione di token con probabilità più alta fra le successioni con probabilità cumulativa maggiori o uguali a p;
- **top_k**: Limita la scelta del prossimo token alle prime k parole con probabilità più alta;
- **temperature**: valore usato per modulare la probabilità del next token; aggiunge computazione ma un valore più alto di questo parametro rende il modello più deterministico nelle scelte e in un certo senso più creativo nelle risposte;

5.2 Scelta dei valori degli iperparametri

Per prima cosa si è definito un range di valori in cui far variare ciascun iperparametro selezionato (inserire fonti). Di seguito i range:

- **max_length** : $\{x \mid x \in \mathbb{Z}, 100 \leq x \leq 10000\}$

- **top_p** : $\{x \mid x \in \mathbb{R}, 0 \leq x \leq 1\}$
- **top_k** : $\{x \mid x \in \mathbb{Z}, 0 \leq x \leq 100\}$
- **temperature** : $\{x \mid x \in \mathbb{R}, 0 \leq x \leq 2\}$

Per settare i valori dei parametri e procedere alle misure si sono adottate diverse strategie di sampling:

- Uniform Sampling: i parametri sono stati fatti variare uniformemente all'interno dei loro range;
- Latin Hypercube Sampling: una ricerca a griglia (come quella uniforme) in cui però l'algoritmo che genera le configurazioni esamina più valori per ogni iperparametro e garantisce che ogni valore venga considerato una sola volta in combinazioni casuali;
- Eventuali altre strategie di sampling;

5.3 Esecuzione degli script

Gli script sono stati creati in modo da eseguire le seguenti operazioni per ogni esecuzione:

1. Si scarica e si quantizza il modello;
2. In un dizionario viene definita la configurazione dei parametri;
3. Si caricano i 10 prompt in una struttura dati;
4. Si instanzia un oggetto di tipo EmissionsTracker;
5. Si avviano in sequenza il tracker e la generazione dell'output;
6. Consumi e output vengono salvati, rispettivamente, nei file emissions.csv e output.txt;

5.4 Valutazione degli output

In seguito si è proceduto ad una valutazione di quanto gli output risultavano coerenti a quello che ci si aspettava e sono state scartate le misurazioni relative ad output poco o per nulla sensati.

5.5 Struttura delle directory che compongono il dataset

Il lavoro di misurazione ha prodotto un ricco dataset di misurazioni organizzato secondo la seguente struttura gerarchica.

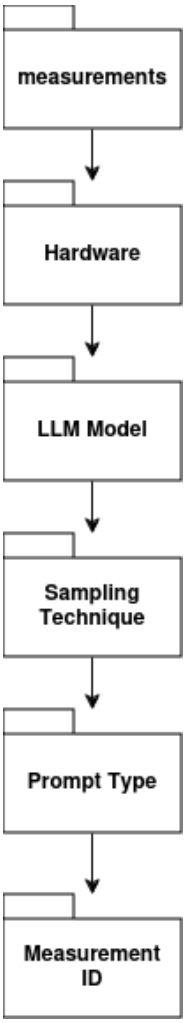

Figura 1: Struttura delle directories

Ad esempio, se volessi accedere alle misurazioni relative alla generazione di coreografie di danza, eseguite sulla macchina virtuale di Google con Latin Hypercube Sampling su Mistral, dovrei seguire il percorso *measurements* > *colab* > *mistral* > *latinHypercubeSampling* > *danceCoreography*.