Homework 0528

1. For different GPUs, figure out what’s the biggest model you can run.

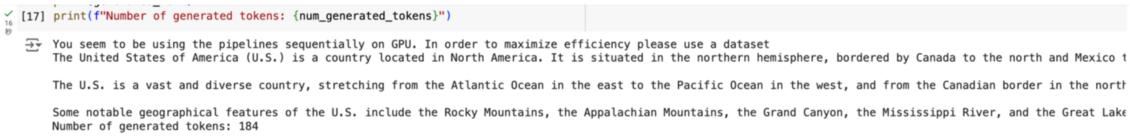
My GPU is A100 and, so far, I can run the following models:

- *"stabilityai/stablelm-3b-4e1t",*
- *"stabilityai/stablelm-base-alpha-3b",*
- *"llama-7b", and "llama-3-8b".*

I haven’t figured out how to run models such as, "llama-2-70b" or "llama-2-13b", but I’ll try to figure it out and hope to successfully run larger models soon.

2. For different GPUs for the biggest model that you can run, measure the token per second speed.

I’ve run the llama-3-8b model to generate text with the prompt 'Where is the U.S.?'. It took 16 seconds to generate 184 tokens (11.5 tokens/sec).



```
[17] print(f"Number of generated tokens: {num_generated_tokens}")
```

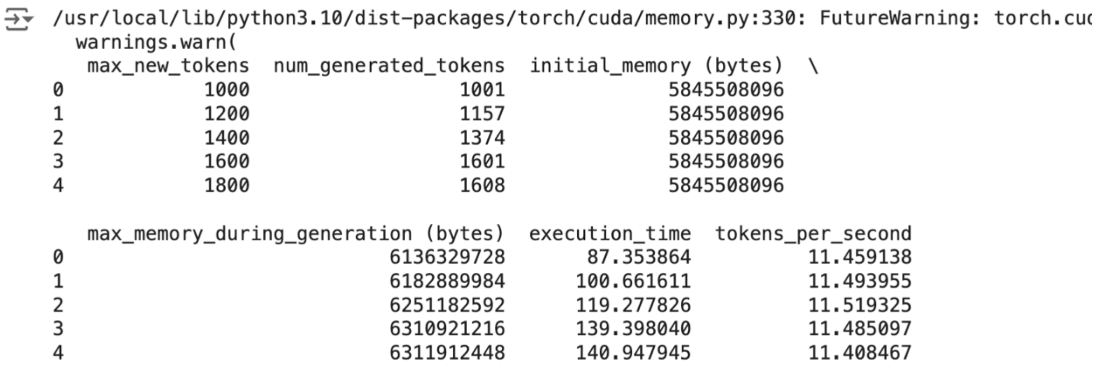
```
You seem to be using the pipelines sequentially on GPU. In order to maximize efficiency please use a dataset
The United States of America (U.S.) is a country located in North America. It is situated in the northern hemisphere, bordered by Canada to the north and Mexico t

The U.S. is a vast and diverse country, stretching from the Atlantic Ocean in the east to the Pacific Ocean in the west, and from the Canadian border in the north

Some notable geographical features of the U.S. include the Rocky Mountains, the Appalachian Mountains, the Grand Canyon, the Mississippi River, and the Great Lake
Number of generated tokens: 184
```

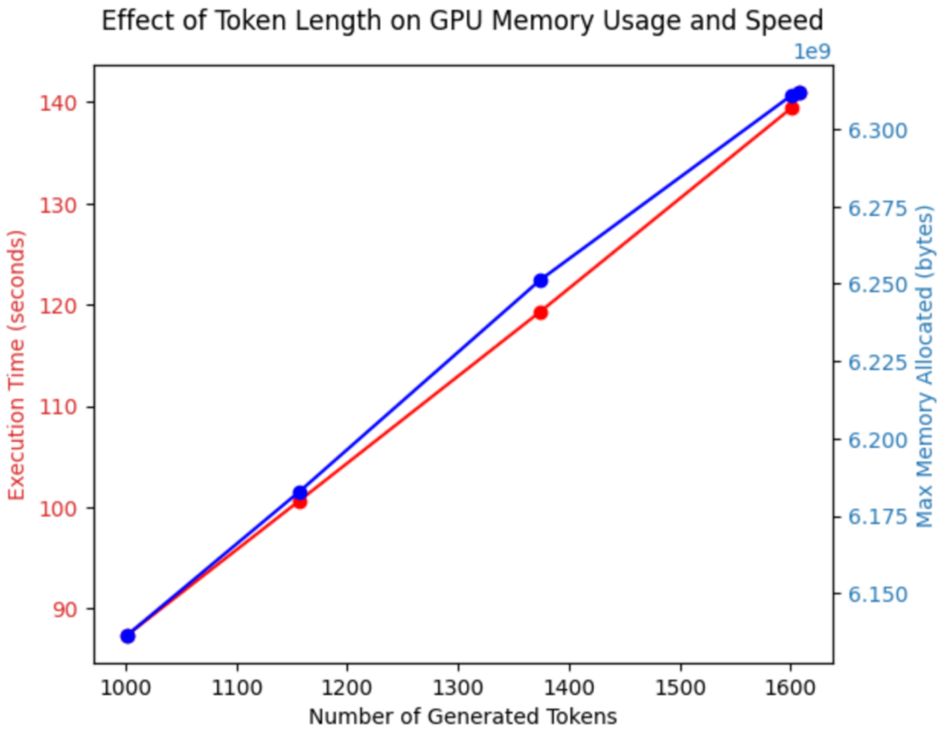
3. Measure the effect of prompt & output token length on GPU memory & speed.

I’m using the same method that I used in Question 2 with the prompt, ‘Tell me a long story’, to generate text. I set different maximum new tokens: [1000, 1200, 1400, 1600, 1800].



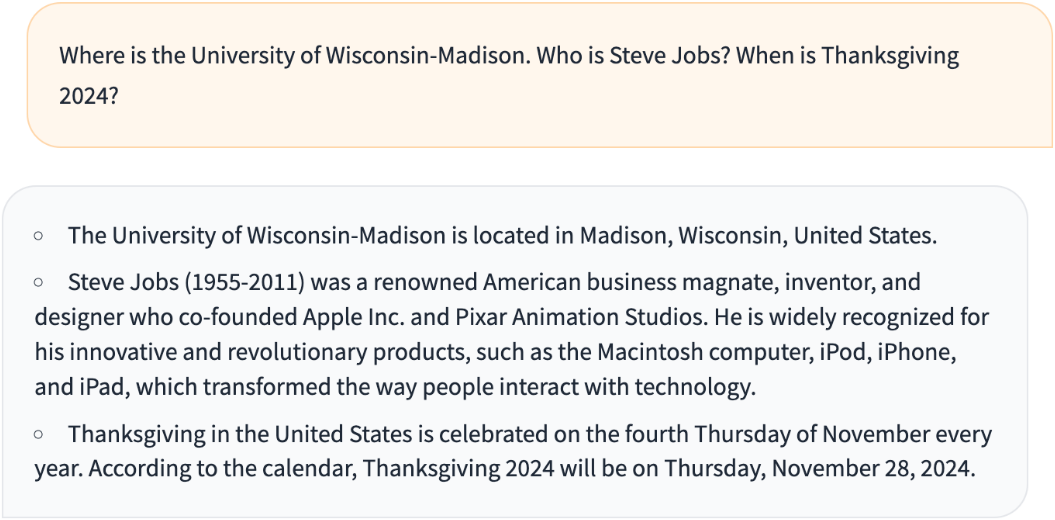
```
/usr/local/lib/python3.10/dist-packages/torch/cuda/memory.py:330: FutureWarning: torch.cu
  warnings.warn(
   max_new_tokens  num_generated_tokens  initial_memory (bytes)  \
0            1000                  1001              5845508096   
1            1200                  1157              5845508096   
2            1400                  1374              5845508096   
3            1600                  1601              5845508096   
4            1800                  1608              5845508096   

   max_memory_during_generation (bytes)  execution_time  tokens_per_second  
0                            6136329728       87.353864          11.459138  
1                            6182889984      100.661611          11.493955  
2                            6251182592      119.277826          11.519325  
3                            6310921216      139.398040          11.485097  
4                            6311912448      140.947945          11.408467  
```



4. Advanced: enable the inference code to do batching (i.e. compute multiple inference request at the same time)

I've asked three questions, which are not related at the same time. This is the result.



Where is the University of Wisconsin-Madison. Who is Steve Jobs? When is Thanksgiving 2024?

- ○ The University of Wisconsin-Madison is located in Madison, Wisconsin, United States.
- ○ Steve Jobs (1955-2011) was a renowned American business magnate, inventor, and designer who co-founded Apple Inc. and Pixar Animation Studios. He is widely recognized for his innovative and revolutionary products, such as the Macintosh computer, iPod, iPhone, and iPad, which transformed the way people interact with technology.
- ○ Thanksgiving in the United States is celebrated on the fourth Thursday of November every year. According to the calendar, Thanksgiving 2024 will be on Thursday, November 28, 2024.

5. Advanced: look into libraries that does quantization on the model.

One of the libraries I found that does quantization is "Hugging Face-quanto"

Quanto provides several unique features such as:

- *weights quantization (float8,int8,int4,int2)*
- *activation quantization (float8,int8)*
- *modality agnostic (e.g CV,LLM)*
- *device agnostic (e.g CUDA,MPS,CPU)*
- *compatibility with torch.compile*
- *easy to add custom kernel for specific device*
- *supports quantization aware training*

I will continue to explore more libraries that support quantization.