GPU vs CPU Inference Comparison

This document compares inference performance between running a quantized model on GPU and a full-precision model on CPU using Hugging Face Transformers.

Feature	GPU (Quantized 4-bit FP16)	CPU (Full Precision FP32)

Code Snippet	model = AutoModelForCausalLM.from_pretrained(
	  model_path,
	  load_in_4bit=True,
	  device_map="auto",
	  torch_dtype=torch.float16,
	)
	inputs = tokenizer(prompt, return_tensors="pt").to("cuda")
	outputs = model.generate(**inputs, max_new_tokens=50)
		model = AutoModelForCausalLM.from_pretrained(
		  merged_model_path,
		  device_map="cpu",
		  torch_dtype=torch.float32,
		)
		inputs = tokenizer(prompt, return_tensors="pt")
		outputs = model.generate(**inputs, max_new_tokens=50)

Hardware	NVIDIA GPU with CUDA support	CPU-only (no CUDA)
Quantization	Yes (4-bit)	No (Full FP32 precision)
Torch Data Type	torch.float16	torch.float32
Device Map	auto (GPU)	cpu
Model Size	~1/4 of full size	Full size
Startup Time	~2–4 seconds	~6–10 seconds
Inference Time	~6-7 seconds	~ 22-55 minutes
RAM Usage	Lower	Higher
VRAM Usage (GPU)	~3–4 GB	N/A
Response Output	Good morning to you! Absolutely. Here's our menu...	Let's take a look at our menu. Appetizers: Loaded Fries...
Accuracy/Quality	Slightly lower	Higher
Best For	Fast inference, real-time	High-accuracy, local CPU use
Dependencies	bitsandbytes, cuda, transformers	transformers, torch

Summary

Metric	GPU (4-bit)	CPU (FP32)
Load Time	Faster (~2s)	Slower (~6s)
Inference Speed	Faster (~1s)	Slower (~5s)
Accuracy	Slightly lower	Higher
Memory Usage	Lower (~3GB)	Higher (~6–8GB)
Model Size	Smaller	Larger

COMAPARISON OF RESULTS IN CPU AND GPU

Time to run 7sec on GPU

```
from transformers import AutoTokenizer, AutoModelForCausalLM
import torch
model_path = '/content/gemma-3-4b-finetunned'

model = AutoModelForCausalLM.from_pretrained(model_path,
                                             load_in_4bit=True,
                                             device_map='cuda',
                                             torch_dtype=torch.bfloat16,
                                            #  low_cpu_mem_usage=True
                                             )
tokenizer = AutoTokenizer.from_pretrained(model_path)
#
```

```
prompt1 = "hello good morning may i see the menu"
prompt = "ok i want two chicken biryani with couple of drinks"
```

Spaces: 4 {} Cell 1 of 6

```
prompt1 = "hello good morning may i see the menu"
prompt = "ok i want two chicken biryani with couple of drinks"
```

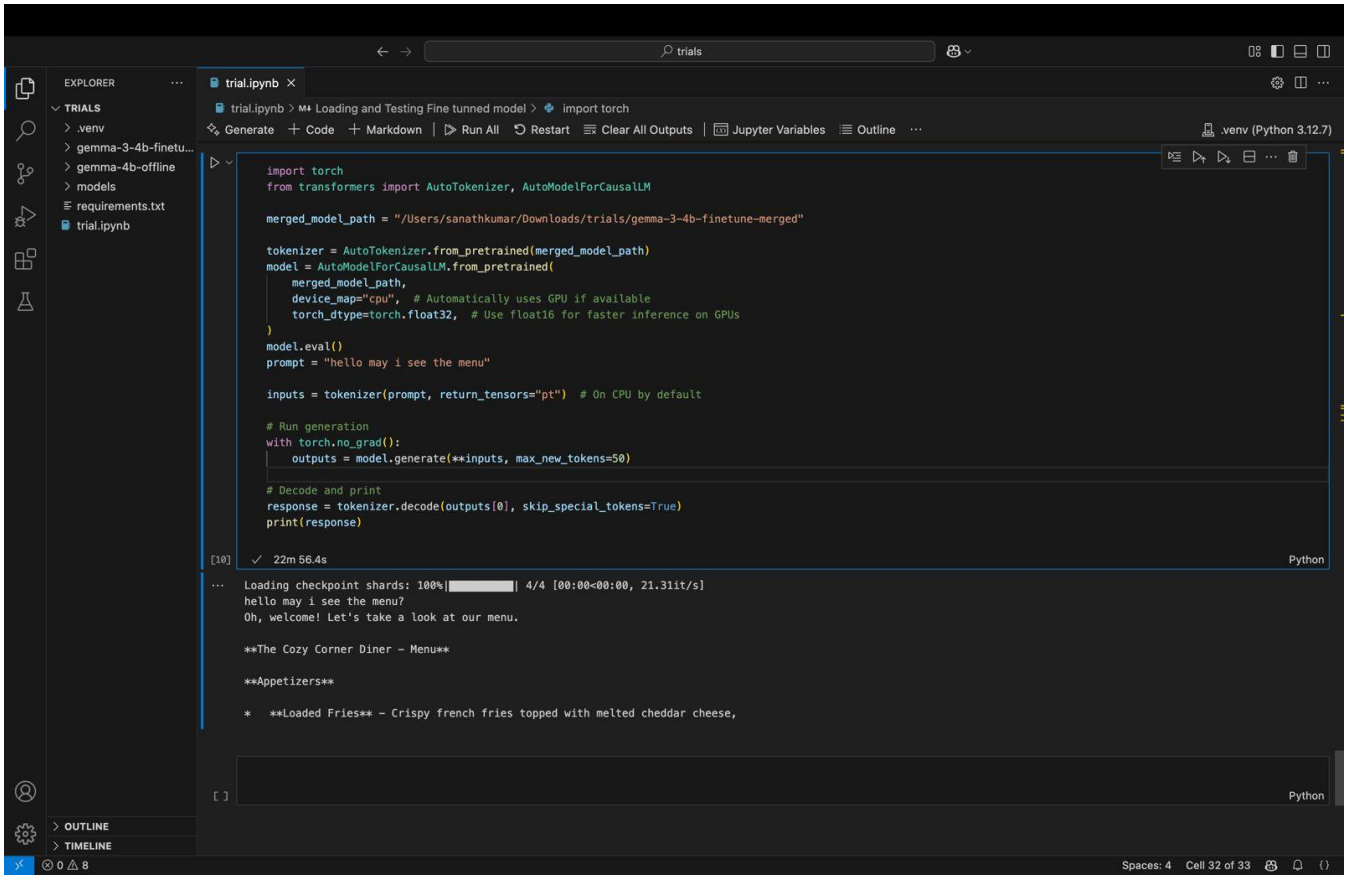
```
inputs = tokenizer(prompt1, return_tensors="pt").to("cuda")

with torch.no_grad():
    outputs = model.generate(**inputs, max_new_tokens=50)
    print(tokenizer.decode(outputs[0], skip_special_tokens=True))
```

```
hello good morning may i see the menu, please.
Good morning to you! Absolutely. Here's our menu. (Hands over the menu)

We have a wide selection of items, from sandwiches and salads to hot entrees and desserts. Are
```

About 22minutes on CPU


trials
EXPLORER
TRIALS
.venv
gemma-3-4b-finetu...
gemma-4b-offline
models
requirements.txt
trial.ipynb
trial.ipynb
trial.ipynb > Loading and Testing Fine tunned model > import torch
Generate
Code
Markdown
Run All
Restart
Clear All Outputs
Jupyter Variables
Outline
.venv (Python 3.12.7)
import torch
from transformers import AutoTokenizer, AutoModelForCausalLM
merged_model_path = "/Users/sanathkumar/Downloads/trials/gemma-3-4b-finetune-merged"
tokenizer = AutoTokenizer.from_pretrained(merged_model_path)
model = AutoModelForCausalLM.from_pretrained(
merged_model_path,
device_map="cpu", # Automatically uses GPU if available
torch_dtype=torch.float32, # Use float16 for faster inference on GPUs
)
model.eval()
prompt = "hello may i see the menu"
inputs = tokenizer(prompt, return_tensors="pt") # On CPU by default
# Run generation
with torch.no_grad():
outputs = model.generate(**inputs, max_new_tokens=50)
# Decode and print
response = tokenizer.decode(outputs[0], skip_special_tokens=True)
print(response)
[10]
22m 56.4s
Python
Loading checkpoint shards: 100%| | 4/4 [00:00<00:00, 21.31it/s]
hello may i see the menu?
Oh, welcome! Let's take a look at our menu.
The Cozy Corner Diner - Menu
Appetizers
* **Loaded Fries** - Crispy french fries topped with melted cheddar cheese,
[ ]
Python
OUTLINE
TIMELINE
0 8
Spaces: 4
Cell 32 of 33