



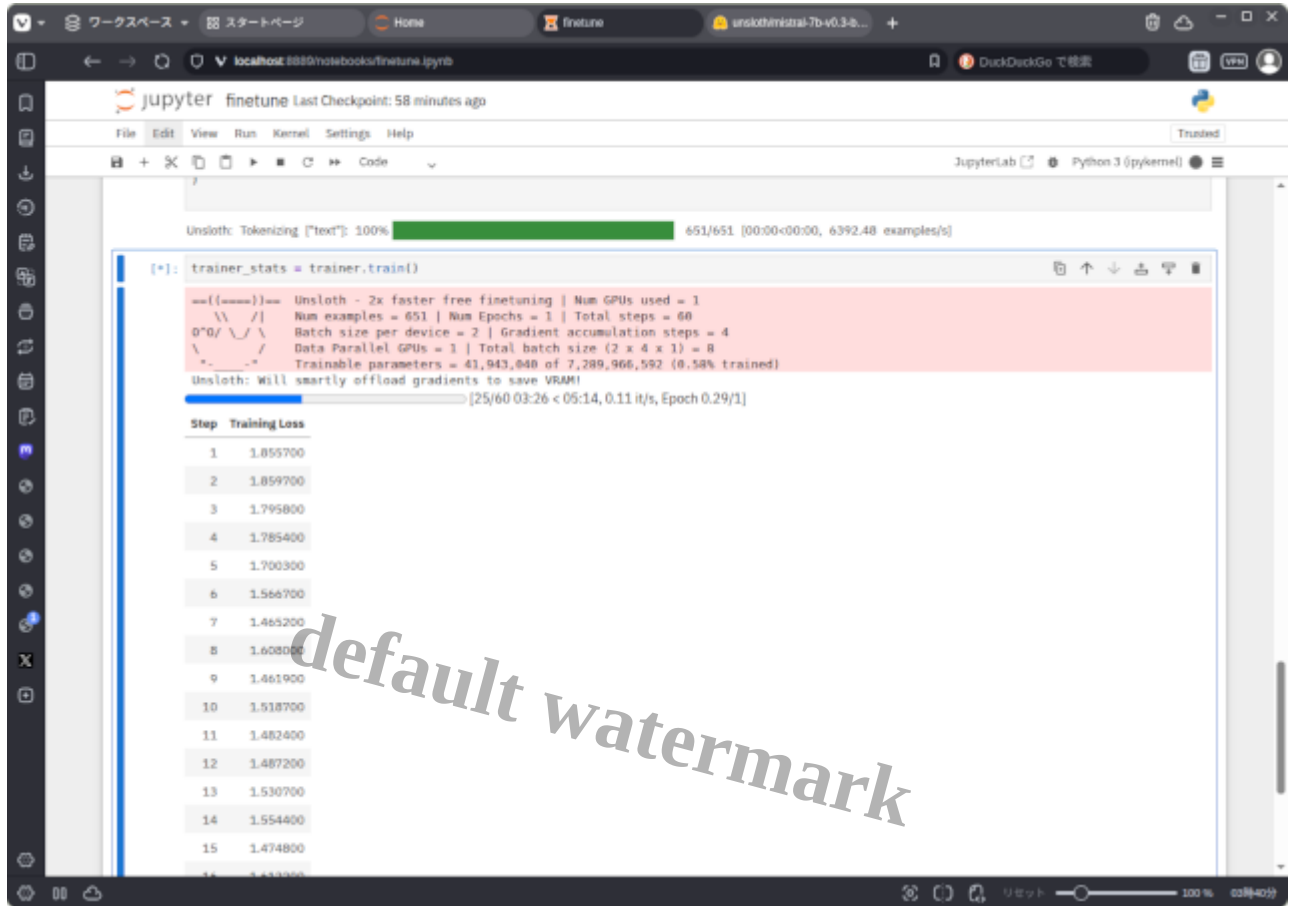
ç?¥è?è??ç©•ã?»å ±æ??ã•®ç ?ç©¶¶ã??ã•ã•®2 Unslothã??ã½¿ã•£ã•
?ã??ã?;ã?ã?ã?³ã?•ã?¥ã?¼ã??ã?³ã?°ã•®è©|ã•¿

Description

æ?¼é??ã•¯æ??ã•ã•|ã½?ã??ã•?ã??æ°?ã•«ã•ã??ã•¾ã•?ã??ã??æ??è¿?ã•¯ã?å®?å "ã•
«æ?¼å?é??è»¤ã•?ã•?æ?¥ã?ã??é?•ã•£ã•|ã•?ã•¾ã•?ã??æ?"å?ã?ã??ã?»?æ?•ã•«ã•?ã•?ã•
|ã?³ã?³ã??ã?¥ã?¼ã?¿ã?¼ã•®å?•ã•«å°§ã•£ã•|LLMã•®ã?•ã?¥ã?¼ã??ã?³ã?°ã??è©|ã•¿ã•|ã•?ã•¾ã•?ã•
?ã??æ?•æ?¥ã?ç?©ã•?ã•?ã•§ã•?ã??

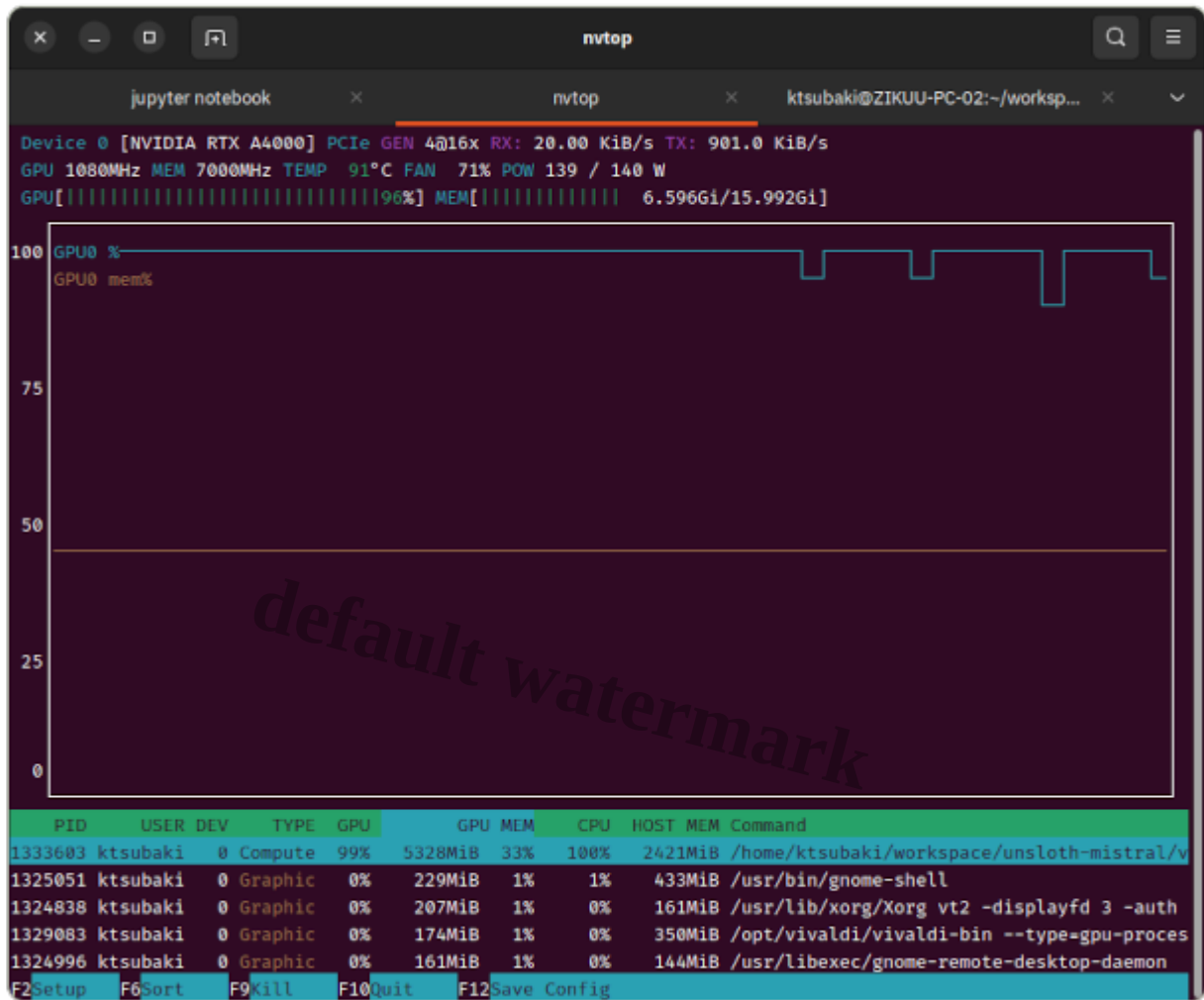
ä»?»å??ã•®ã??ã?¼ã??ã•¯ã??Mistral 7Bã??Unslothã??ã½¿ã•£ã•|QLoRAã?•ã?¥ã?¼ã??ã?³ã?°ã??è¿?ã•
?ã?•ã•§ã•?ã??

Unslothã•VRAMã•®ã½¿ç?´é?•ã??ç?´?ç?ã?ã•|LLMã??ã?•ã?¥ã?¼ã??ã?³ã?°ã•§ã••
ã??ã?©ã?ã??ã?©ã?ã?¼ã•§ã•?ã??



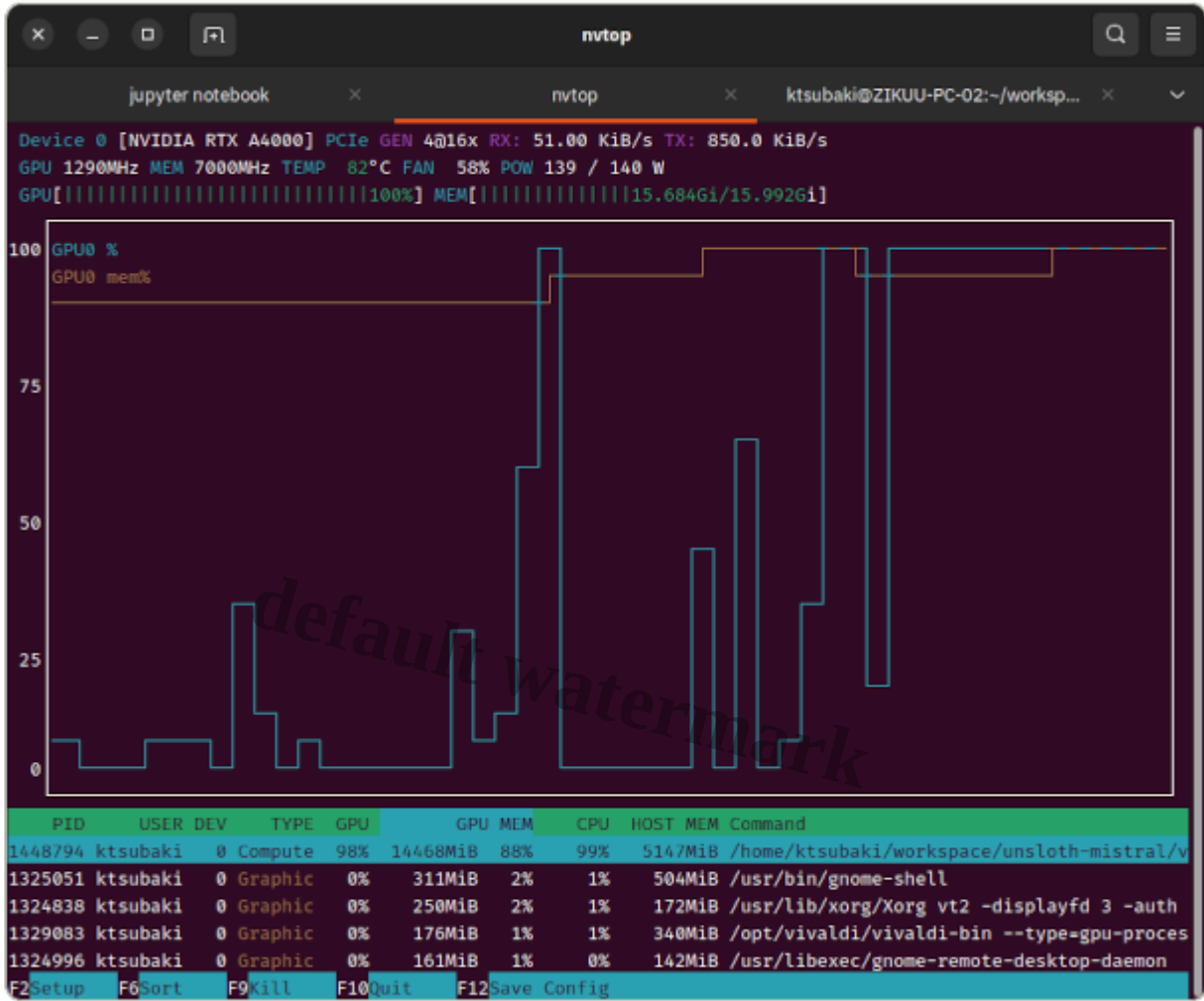
à»?â??ã·VRAMã®ä½¿ç?´é?·ã·«ç?ç?®ã·ã·ä½¿æ¥ã·ã·¾ã·ã·ã??ä½¿ç?´ã·ã·
 ?ã??ã·¾ã·¿ã?»ã??ã??ã·ã·ã·®ã??ã??ã·ã·ã??æ?½ã·ã·ã·ã??ã??ã·ã·ã°ã·
 ?ã??ã·¾ã·¿ã??è¿½ã·ã·ã·ã·690ä»¶ã·ã·©ã·®ã??ã·®ã·Sã·?Alpacã½¿ã½ã·ã·Sã·ã??

ā·?ā??ā·─Mistral 7Bā®4ā??ā??ā??é?·ā·ā??ā?ā??ā?«ā??ā?·ā?¥ā?¼ā??ā?³ā?°ā·?ā·?ā··ā·ā·
 ®VRAMā½½ç?·é?·ā·šā·?ā??VRAMā½½ç?·é?·ā·─8GBā»¥ā? ā«ā·?ā·¾ā·ā·ā·?ā·¾ā·?ā??ā·?ā??ā·
 āā??ā?·ā·?ā·?é?·ā·ā·ā·é· ā·?ā·®ā??æ??æ ā·?ā??ā·ā·?·Core i5/Ryzen 5 ā·─ GeForce RTX 3060
 12GBā??ç·ā·ç·ā·?ā·?ā·?ā·æ─è¼ç??ā·®?ā¾¼ā·ç·ā??ā·?ā·PCā·šā??ā??ā·¾ā·?ā??



ã•jã•ã•¿ã•«ã?•Gemma 3 12Bã•@4ã??ã??ã??é?•ã•ã??ã?¢ã??ã?«ã•Sã??ã?-ã?¼ã??ã?³ã?°ã•?ã•?ã•ã•ã•
ã•@VRAMã½¿ç?•é?•ã??ç?°èª•ã•?ã•?ã•@ã•?ã•@ã•¹ã?-ã?ªã?¼ã?³ã?•ã?Sã??ã??ã?Sã•?ã??

åj¼ã•@PCã•Core i5 13500 + RTX A4000ã•@çµ?ã•¿ã•?ã•?ã•?ã•Sã•?ã??16GBã•@VRAMã•
Sã?@ã?ªã?@ã?ªã??ã??ã??æ??ã•?ã•Sã•?ã??A4000ã•æ??æ??°ã•@ã?²ã?¼ã? ç?“GPUã•«æ-?ã¹ã??ã•
“ã?|ç•?é??ã°|ã•Sã?£ã??ã•¾ã•?ã??ã?¿ã?¼ã?³ã?¢ã?©ã?|ã?³ã??ã??é??ã•ã•?ã??ã•«ã•å?ã¾jæ ¼ã½-ã
@ã?²ã?¼ã? ç?“GPUã??ã½¿ã•£ã•?æ?¹ã•?è-ã?ã•æ?•ã•?ã•¾ã•?ã??ã•?ã•?ã??ã•?ã•£ã•é«?é??ã•
ªGPUã??æ??ã•«å ¥ã??ã•|ã?•A4000ã•-ã?µã?¼ã?ã?¼ã•«ç§S»è-ã•?ã•|æ?“è«?ã°?ç?“ã•«ã•?ã•?ã•?ã•Sã•
?ã??



çµ•æ??ã·-æ??å¼¾ ã?ã·|ã·?ã·?ã??ã·®ã·«ã·-ç'?'é ã·?ã??ã·®ã·«ã·ã??ã·¾ã·?ã·
 ?ã??ã»?å¾¾?ã??è©|èj?é?-èªã??ç¶¶?ã?ã·|ã·?ã·ã·¾ã·?ã??æ¬¡ã·®ã??ã??¼ã??ã·
 -ã??ã??¼ã??¿ã?»ã??ã??ã??ç¶¶°éº?ã·«ã·?ã??ã·?ã·ã·ã·?ã·?ã·¥ã??¼ã??ã??ã??ã??ã??ã??©ã??ã??¼ã??¿ã??¼ã·
 ®èª¿æ??ã·Sã·?ã??ã??ã·?ã??ã·-ã·?ã·?ã·?ã·?ã»?çµ¶ã·¿ã·«ã·ã·£ã·|ã·?ã??ã·?ã·?ã·?ã·?ã·?ã·?ã·?
 ·ã·?ã·?ã·ã·?ã·?ã·ã·ã·£ã·?ã·?ã·ã·?ã·?ç·?è§£ã·Sã·ã·?ã·¾ã·Sã·-ã·æ??é??ã·?ã·?ã·?ã·£ã·|ã??è?-ã·?ã·
 ®ã·Sã·?ã·¶ã·ã·ã·?ã??çµ¶ã·¿ã·¾ã·?ã??

ǎ•?è??ǎ•¾ǎ•šǎ•«ǎ•?ǎ»?ǎ??ǎ®?è¡?ǎ•?ǎ•?ǎ?³ǎ?¼ǎ??ǎ•šǎ•?ǎ??ǎ•?ǎ•?ǎ•?ǎ•ǎ•ǎ•?ǎ•?ǎ•?ǎ•?ǎ•ǎ•?ǎ•?ǎ•?ǎ•
 ?æ??æ??ǎ•ǎ•ǎ•?ǎ•?ǎ??

```
from unsloth import FastLanguageModel
import torch
from trl import SFTTrainer
from transformers import TrainingArguments, TextStreamer
```

```
max_seq_length = 2048
dtype = None
load_in_4bit = True
fourbit_models = [
    "unsloth/mistral-7b-v0.3-bnb-4bit",
```

```
]

model, tokenizer = FastLanguageModel.from_pretrained(
    model_name = "unsloth/mistral-7b-v0.3-bnb-4bit",
    max_seq_length = max_seq_length,
    dtype = dtype,
    load_in_4bit = load_in_4bit,
)

model = FastLanguageModel.get_peft_model(
    model,
    r = 16,
    target_modules = ["q_proj", "k_proj", "v_proj", "o_proj", "gate_proj", "up_proj", "down_proj"],
    lora_alpha = 16,
    lora_dropout = 0,
    bias = "none",
    use_gradient_checkpointing = True,
    random_state = 3407,
    use_rslora = False,
    loftq_config = None,
)

alpaca_prompt = """"Below is an instruction that describes a task, paired with
### Instruction:
{}

### Input:
{}

### Response:
{}""""
EOS_TOKEN = tokenizer.eos_token

def formatting_prompts_func(examples):
    instructions = examples["instruction"]
    inputs = examples["input"]
    outputs = examples["output"]
    texts = []
    for instruction, input, output in zip(instructions, inputs, outputs):
        text = alpaca_prompt.format(instruction, input, output) + EOS_TOKEN
        texts.append(text)
    return { "text": texts, }

pass

from datasets import load_dataset
dataset = load_dataset("json", data_files="datasets/qlora_dataset.json", split="train")
dataset = dataset.map(formatting_prompts_func, batched = True)

training_args = TrainingArguments(
    per_device_train_batch_size = 2,
    gradient_accumulation_steps = 4,
    warmup_steps = 5,
    max_steps = 60,
    learning_rate = 2e-4,
```

```
fp16 = not torch.cuda.is_bf16_supported(),
lr_scheduler_type = "linear",
seed = 3407,
bf16 = torch.cuda.is_bf16_supported(),
logging_steps = 1,
optim = "adamw_8bit",
weight_decay = 0.01,
output_dir = "outputs"
)

trainer = SFTTrainer(
    model = model,
    tokenizer = tokenizer,
    train_dataset = dataset,
    dataset_text_field = "text",
    max_seq_length = max_seq_length,
    dataset_num_proc = 2,
    packing = False,
    args = training_args
)

trainer_stats = trainer.train()
print(trainer_stats)
trainer.save_model("./trained")
tokenizer.save_pretrained("./trained")

FastLanguageModel.for_inference(model)
inputs = tokenizer(
    [
        alpaca_prompt.format(
            "è³ª?•ã•«ã¼ã?ã•|è©³ç´°ã?ã•ªä, •ã¸ã•«ã??ç?ã?ã•|ã••ã•ã•?ã•
?ã??",
            "ã?¢ã??ã•¥ã••ã??ã¼ZIKUUã•®ã¼¼é?•ã•®ã??ã?ã??ã?£ã?¼ã?«",
            ""
        )
    ], return_tensors = "pt").to("cuda")
text_streamer = TextStreamer(tokenizer)
_ = model.generate(**inputs, streamer = text_streamer, max_new_tokens = 128)
```

Category

- 1. æ?¥ã?ã•®æ´»ã??

Tags

- 1. AI
- 2. Linux
- 3. LLM
- 4. Mistral
- 5. QLoRA
- 6. Unsloth
- 7. ã??ã?jã?ªã?³ã?•ã?¥ã?¼ã??ã?³ã?°
- 8. ä,?é??å??å,?

9. åð§è!•æ¨;è¨?èª?ã?¢ã??ã?«
10. å±±±æ¢¨ç??

Date Created

2025å¹8æ??6æ?¥

Author

kazuo-tsubaki

default watermark