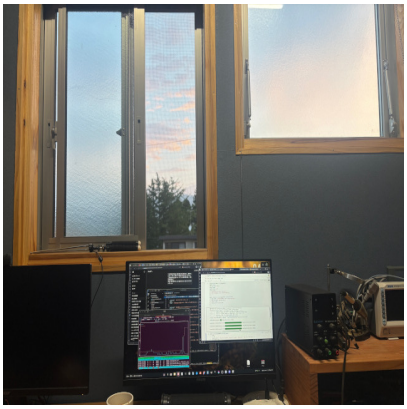


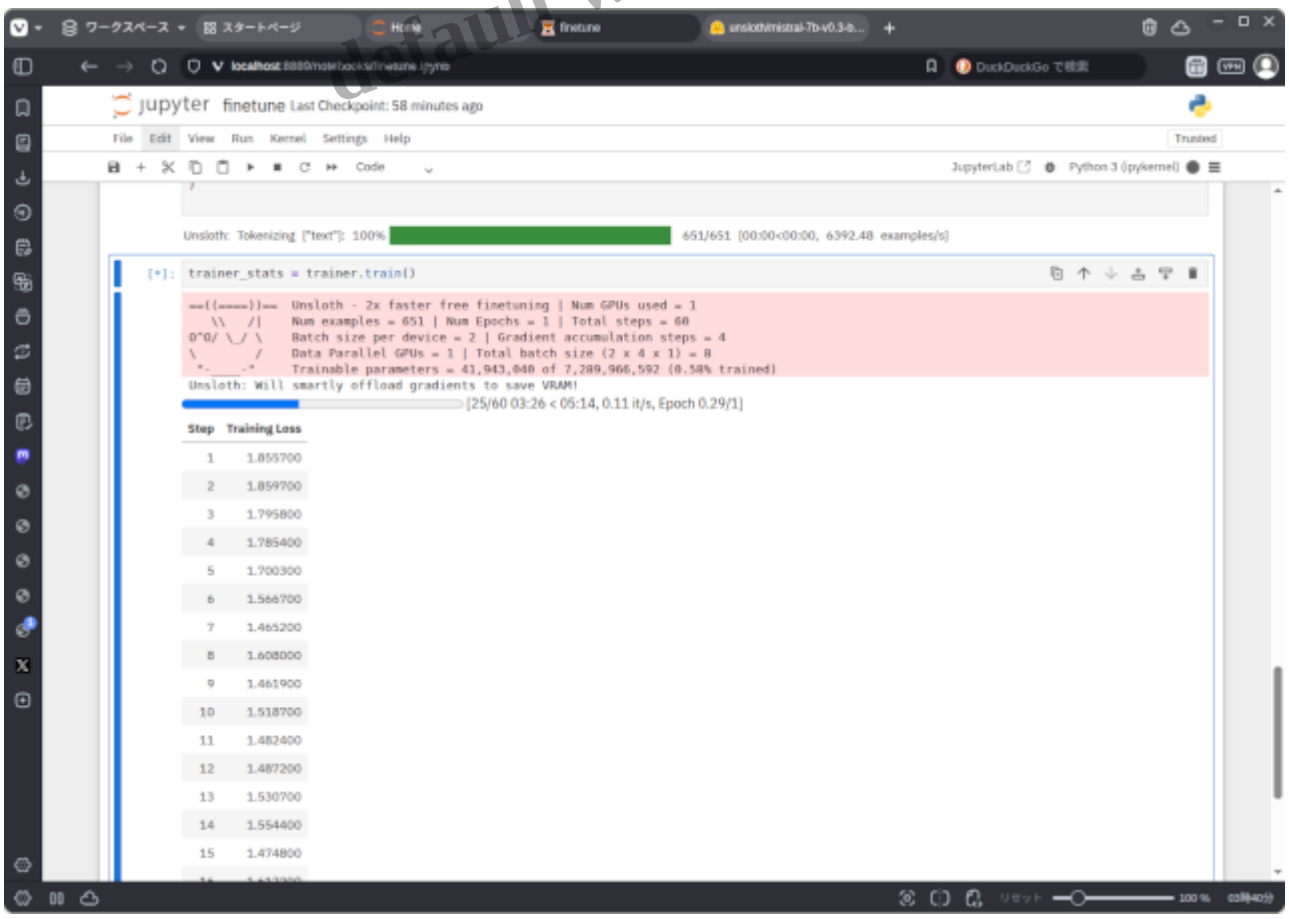
?????????????2 Unsloth???????????????

Description

??LLM??????????????????

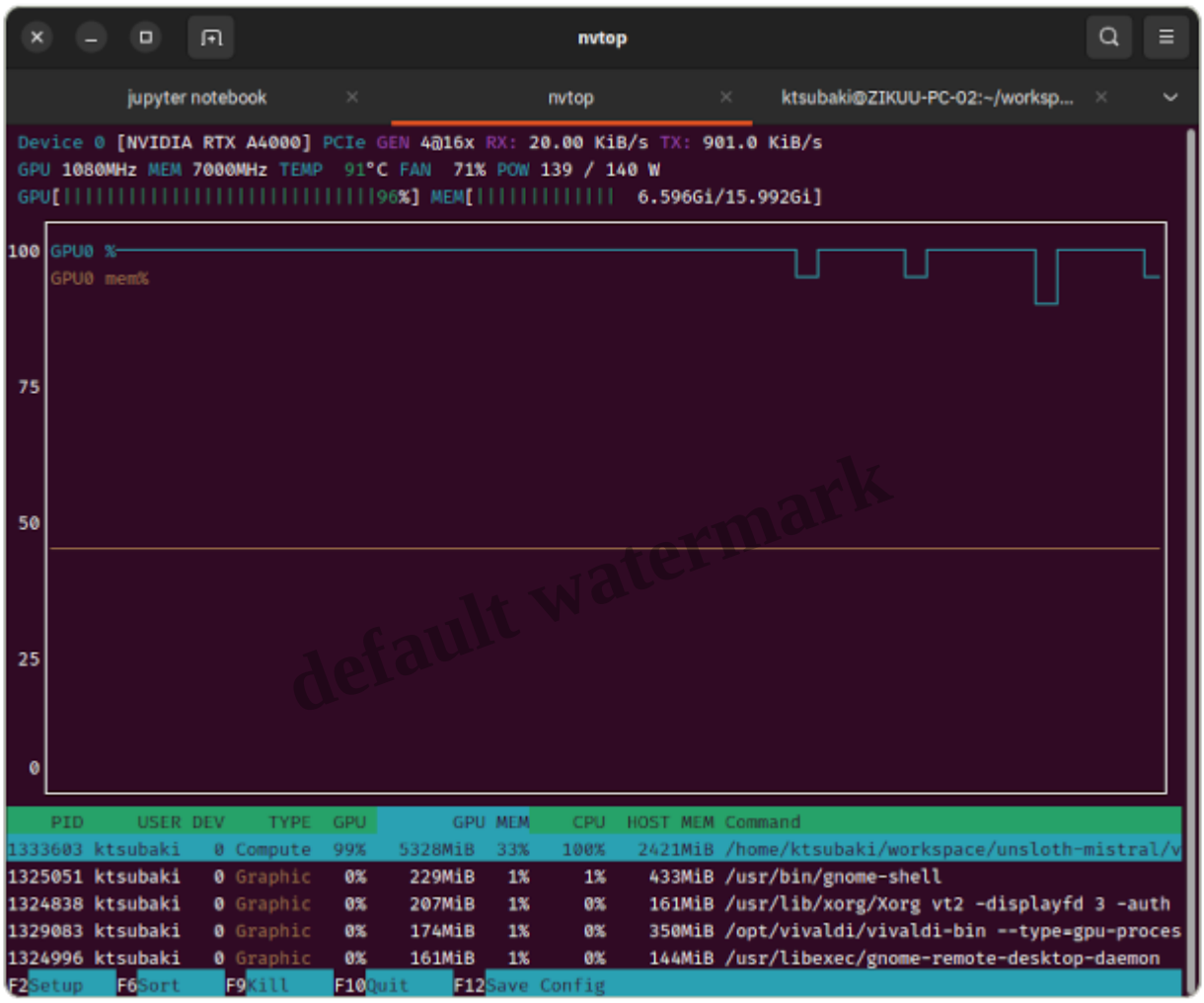
????????Mistral 7B?Unsloth????QLoRA?????????????

Unsloth?VRAM?????????LLM??????????????????



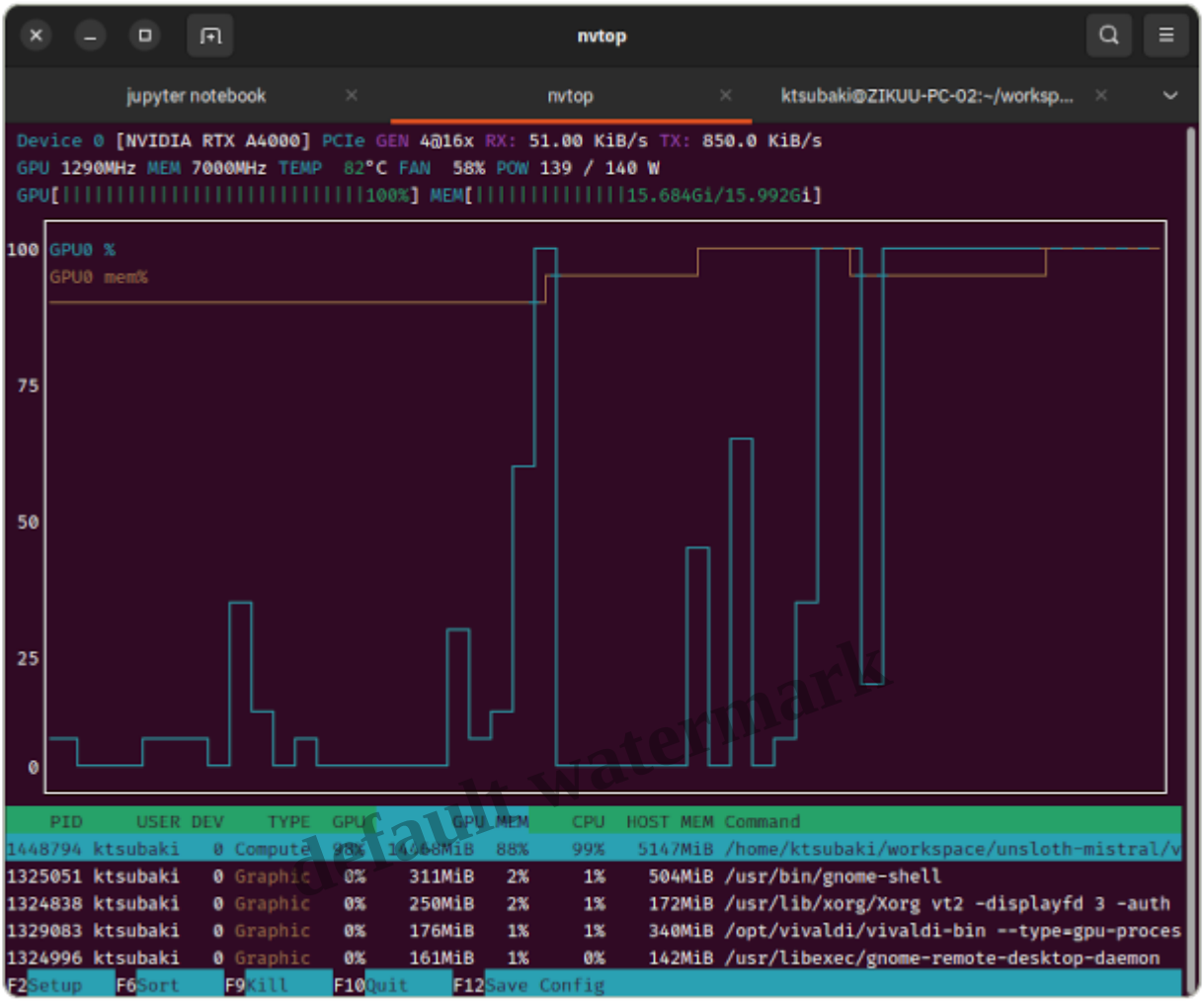
???VRAM??690????????Alpaca?????

???Mistral 7B?4??????????????????????VRAM??????VRAM????8GB????????????????????????????????Core i5/Ryzen 5 ? GeForce RTX 3060 12GB??????????????????PC??????



?????Gemma 3 12B?4??????????????????????VRAM????????????????????????

??PC?Core i5 13500 + RTX A4000?????????16GB?VRAM?????????????A4000????????GPU????????????????????????????????



??

??????????????????????????????????????

```
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
    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
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(
            "??????????????????????",
            "??????ZIKUU??????????",
            "",
        )
    ], return_tensors = "pt").to("cuda")
text_streamer = TextStreamer(tokenizer)
_ = model.generate(**inputs, streamer = text_streamer, max_new_tokens = 128)
```


Category

1. ???

Tags

1. AI
2. Linux
3. LLM
4. Mistral
5. QLoRA
6. Unsloth
7. ??????????
8. ????
9. ????????
10. ???

Date Created
2025?8?6?
Author
kazuo-tsubaki