```
!pip install unsloth
!pip install --force-reinstall --no-cache-dir --no-deps git+https://github.com/unslothai/unsloth.git
```

Show hidden output

```
from google.colab import userdata
hf_token = userdata.get('Hugging_face_new')
```

```
from unsloth import FastLanguageModel

max_seq_length = 512
dtype = None
load_in_4bit = True
```

```
🦥 Unsloth: Will patch your computer to enable 2x faster free finetuning.
Unsloth: Failed to patch Gemma3ForConditionalGeneration.
🦥 Unsloth Zoo will now patch everything to make training faster!
```

```
model, tokenizer = FastLanguageModel.from_pretrained(
    model_name="unsloth/Phi-3-mini-4k-instruct",
    max_seq_length=max_seq_length,
    dtype=dtype,
    load_in_4bit=load_in_4bit
)
```

```
==((====))==  Unsloth 2025.3.19: Fast Mistral patching. Transformers: 4.51.1.
   \\   /|    Tesla T4. Num GPUs = 1. Max memory: 14.741 GB. Platform: Linux.
O^O/ \_/ \    Torch: 2.6.0+cu124. CUDA: 7.5. CUDA Toolkit: 12.4. Triton: 3.2.0
\        /    Bfloat16 = FALSE. FA [Xformers = 0.0.29.post3. FA2 = False]
 "-____-"     Free license: http://github.com/unslothai/unsloth
Unsloth: Fast downloading is enabled - ignore downloading bars which are red colored!
model.safetensors: 100%  2.26G/2.26G [00:06<00:00, 500MB/s]
generation_config.json: 100%  194/194 [00:00<00:00, 28.2kB/s]
tokenizer_config.json: 100%  3.34k/3.34k [00:00<00:00, 473kB/s]
tokenizer.model: 100%  500k/500k [00:00<00:00, 8.76MB/s]
added_tokens.json: 100%  293/293 [00:00<00:00, 41.1kB/s]
special_tokens_map.json: 100%  458/458 [00:00<00:00, 67.6kB/s]
tokenizer.json: 100%  1.84M/1.84M [00:00<00:00, 7.57MB/s]
```

```
from datasets import load_dataset
```

```
dataset = load_dataset("json", data_files="/content/MTU Data Science Instruct Dataset.json")
```

```
Generating train split: 117/0 [00:00<00:00, 2071.75 examples/s]
```

```
dataset
```

```
DatasetDict({
    train: Dataset({
        features: ['instruct', 'answer'],
        num_rows: 117
    })
})
```

```
def format_phi3(example):
    return {
        "text": f"<|user|>\n{example['instruct']}<|end|>\n<|assistant|>\n{example['answer']}<|end|>\n<|endoftext|>"
    }
```

```
formatted_dataset = dataset.map(format_phi3, remove_columns=dataset["train"].column_names)
```

```
Map: 100%  117/117 [00:00<00:00, 6277.61 examples/s]
```

```
formatted_dataset
```

```
DatasetDict({
    train: Dataset({
        features: ['text'],
        num_rows: 117
    })
})
```

```
formatted_dataset['train'][45]
```

```
{'text': '<|user|>\nWhat does MA 5770 - Bayesian Statistics cover?<|end|>\n<|assistant|>\nIt covers Bayesian inference methods for statistical analysis.<|end|>\n<|endoftext|>'}
```

```
split_dataset = formatted_dataset['train']
```

```
split_dataset = split_dataset.train_test_split(test_size = 0.1, seed = 42)
```

```
train_dataset = split_dataset["train"]
val_dataset = split_dataset["test"]
```

```
formatted_dataset['train']
```

```
Dataset({
    features: ['text'],
    num_rows: 117
})
```

```
train_dataset
```

```
Dataset({
    features: ['text'],
    num_rows: 105
})
```

```
val_dataset
```

```
Dataset({
    features: ['text'],
    num_rows: 12
})
```

```
model = FastLanguageModel.get_peft_model(
    model,
    r=16,
    target_modules= ["q_proj", "k_proj", "v_proj", "o_proj"],
    lora_alpha=16,
    lora_dropout=0,
    bias="none",
    use_gradient_checkpointing=True,
    random_state=3407,
)
```

```
Not an error, but Unsloth cannot patch MLP layers with our manual autograd engine since either LoRA adapters
are not enabled or a bias term (like in Qwen) is used.
Unsloth 2025.3.19 patched 32 layers with 32 QKV layers, 32 O layers and 0 MLP layers.
```

```
model.print_trainable_parameters()
```

```
trainable params: 12,582,912 || all params: 3,833,662,464 || trainable%: 0.3282
```

```
from transformers import TrainingArguments
from unsloth import is_bfloat16_supported
```

```
training_args = TrainingArguments(
    output_dir = "/content/logs",
    per_device_train_batch_size=2,
    gradient_accumulation_steps=4,
    num_train_epochs=3,
    learning_rate=2e-4,
    logging_steps=10,
```

```
    save_steps=50,
    fp16=not is_bfloat16_supported(),
    bf16=is_bfloat16_supported(),

    save_total_limit=1,
    gradient_checkpointing=True,
    optim="paged_adamw_8bit",
    lr_scheduler_type="cosine",
    warmup_steps=10,
    max_steps=100,
    report_to="none"
)
```

```
from trl import SFTTrainer
```

```
trainer = SFTTrainer(
    model=model,
    tokenizer=tokenizer,
    train_dataset=formatted_dataset['train'],
    dataset_text_field="text",
    max_seq_length=2000,
    dataset_num_proc=2,
    args = training_args
)
```

Unsloth: Tokenizing ["text"] (num_proc=2): 100% 117/117 [00:00<00:00, 199.21 examples/s]

```
MTU_Model = trainer.train()
```

Show hidden output

```
question = "What are the core course for Datascience Students?"
```

```
FastLanguageModel.for_inference(model)

prompt = tokenizer.apply_chat_template(
    [{"role": "user", "content": question}],
    tokenize=False,
    add_generation_prompt=True
)
```

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

outputs = model.generate(
    input_ids=inputs.input_ids,
    attention_mask=inputs.attention_mask,
    max_new_tokens=512,
    temperature=0,
    top_p=0.9,
    eos_token_id=tokenizer.eos_token_id
)

response = tokenizer.decode(outputs[0], skip_special_tokens=True)
print(response)
```

Start coding or generate with AI.