```
#!pip install datasets bitsandbytes trl
# bitsandbytes had to be bumped to 0.45.2 to avoid errors in Colab env
!pip install transformers==4.46.2 peft==0.13.2 accelerate==1.1.1 trl==0.12.1 bitsandby
```

```
Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.11/dist-packages (from transformers==4.46
Requirement already satisfied: requests in /usr/local/lib/python3.11/dist-packages (from transformers==4.46.2) (2.32
Requirement already satisfied: tokenizers<0.21,>=0.20 in /usr/local/lib/python3.11/dist-packages (from transformers=
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.11/dist-packages (from transformers==4.46.2) (4.
Requirement already satisfied: psutil in /usr/local/lib/python3.11/dist-packages (from peft==0.13.2) (5.9.5)
Requirement already satisfied: torch>=1.13.0 in /usr/local/lib/python3.11/dist-packages (from peft==0.13.2) (2.6.0+c
Requirement already satisfied: rich in /usr/local/lib/python3.11/dist-packages (from trl==0.12.1) (13.9.4)
Requirement already satisfied: pyarrow>=15.0.0 in /usr/local/lib/python3.11/dist-packages (from datasets==3.1.0) (18
Requirement already satisfied: dill<0.3.9,>=0.3.0 in /usr/local/lib/python3.11/dist-packages (from datasets==3.1.0)
Requirement already satisfied: xxhash in /usr/local/lib/python3.11/dist-packages (from datasets==3.1.0) (3.5.0)
Requirement already satisfied: multiprocess<0.70.17 in /usr/local/lib/python3.11/dist-packages (from datasets==3.1.0
Requirement already satisfied: fsspec<=2024.9.0,>=2023.1.0 in /usr/local/lib/python3.11/dist-packages (from fsspec[h
Requirement already satisfied: aiohttp in /usr/local/lib/python3.11/dist-packages (from datasets==3.1.0) (3.11.15)
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python3.11/dist-packages (from huggingfa
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.11/dist-packages (from pandas==2.2.2
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/dist-packages (from pandas==2.2.2) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/dist-packages (from pandas==2.2.2) (2025.
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.11/dist-packages (from matplotlib==3.8.0)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.11/dist-packages (from matplotlib==3.8.0) (0.1
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.11/dist-packages (from matplotlib==3.8.0)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.11/dist-packages (from matplotlib==3.8.0)
Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.11/dist-packages (from matplotlib==3.8.0) (11
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.11/dist-packages (from matplotlib==3.8.0)
Requirement already satisfied: aiohappyeyeballs>=2.3.0 in /usr/local/lib/python3.11/dist-packages (from aiohttp->dat
Requirement already satisfied: aiosignal>=1.1.2 in /usr/local/lib/python3.11/dist-packages (from aiohttp->datasets==
Requirement already satisfied: attrs>=17.3.0 in /usr/local/lib/python3.11/dist-packages (from aiohttp->datasets==3.1
Requirement already satisfied: frozenlist>=1.1.1 in /usr/local/lib/python3.11/dist-packages (from aiohttp->datasets=
Requirement already satisfied: multidict<7.0,>=4.5 in /usr/local/lib/python3.11/dist-packages (from aiohttp->dataset
Requirement already satisfied: propcache>=0.2.0 in /usr/local/lib/python3.11/dist-packages (from aiohttp->datasets==
Requirement already satisfied: yarl<2.0,>=1.17.0 in /usr/local/lib/python3.11/dist-packages (from aiohttp->datasets=
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-packages (from python-dateutil>=2.8.2->par
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.11/dist-packages (from requests->t
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/dist-packages (from requests->transformers=
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.11/dist-packages (from requests->transfo
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.11/dist-packages (from requests->transfo
Requirement already satisfied: networkx in /usr/local/lib/python3.11/dist-packages (from torch>=1.13.0->peft==0.13.2
Requirement already satisfied: jinja2 in /usr/local/lib/python3.11/dist-packages (from torch>=1.13.0->peft==0.13.2)
Requirement already satisfied: nvidia-cuda-nvrtc-cu12==12.4.127 in /usr/local/lib/python3.11/dist-packages (from tor
Requirement already satisfied: nvidia-cuda-runtime-cu12==12.4.127 in /usr/local/lib/python3.11/dist-packages (from t
Requirement already satisfied: nvidia-cuda-cupti-cu12==12.4.127 in /usr/local/lib/python3.11/dist-packages (from tor
Requirement already satisfied: nvidia-cudnn-cu12==9.1.0.70 in /usr/local/lib/python3.11/dist-packages (from torch>=1
Requirement already satisfied: nvidia-cublas-cu12==12.4.5.8 in /usr/local/lib/python3.11/dist-packages (from torch>=
Requirement already satisfied: nvidia-cufft-cu12==11.2.1.3 in /usr/local/lib/python3.11/dist-packages (from torch>=1
Requirement already satisfied: nvidia-curand-cu12==10.3.5.147 in /usr/local/lib/python3.11/dist-packages (from torch
Requirement already satisfied: nvidia-cusolver-cu12==11.6.1.9 in /usr/local/lib/python3.11/dist-packages (from torch
Requirement already satisfied: nvidia-cusparse-cu12==12.3.1.170 in /usr/local/lib/python3.11/dist-packages (from tor
Requirement already satisfied: nvidia-cusparselt-cu12==0.6.2 in /usr/local/lib/python3.11/dist-packages (from torch>
Requirement already satisfied: nvidia-nccl-cu12==2.21.5 in /usr/local/lib/python3.11/dist-packages (from torch>=1.13
Requirement already satisfied: nvidia-nvtx-cu12==12.4.127 in /usr/local/lib/python3.11/dist-packages (from torch>=1.
Requirement already satisfied: nvidia-nvjitlink-cu12==12.4.127 in /usr/local/lib/python3.11/dist-packages (from torc
Requirement already satisfied: triton==3.2.0 in /usr/local/lib/python3.11/dist-packages (from torch>=1.13.0->peft==0
Requirement already satisfied: sympy==1.13.1 in /usr/local/lib/python3.11/dist-packages (from torch>=1.13.0->peft==0
Requirement already satisfied: mpmath<1.4,>=1.1.0 in /usr/local/lib/python3.11/dist-packages (from sympy==1.13.1->to
Requirement already satisfied: markdown-it-py>=2.2.0 in /usr/local/lib/python3.11/dist-packages (from rich->trl==0.1
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in /usr/local/lib/python3.11/dist-packages (from rich->trl==0
Requirement already satisfied: mdurl~=0.1 in /usr/local/lib/python3.11/dist-packages (from markdown-it-py>=2.2.0->ri
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.11/dist-packages (from jinja2->torch>=1.13.
```

```
import os
import torch
from datasets import load_dataset
from peft import get_peft_model, LoraConfig, prepare_model_for_kbit_training
from transformers import AutoModelForCausalLM, AutoTokenizer, BitsAndBytesConfig
from trl import SFTConfig, SFTTrainer



bnb_config = BitsAndBytesConfig(
    load_in_4bit=True,
    bnb_4bit_quant_type="nf4",
    bnb_4bit_use_double_quant=True,
    bnb_4bit_compute_dtype=torch.float32
)
repo_id = 'microsoft/Phi-3-mini-4k-instruct'
model = AutoModelForCausalLM.from_pretrained(
    repo_id, device_map="cuda:0", quantization_config=bnb_config
)
```

```
/usr/local/lib/python3.11/dist-packages/huggingface_hub/utils/_auth.py:94: UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab (https://huggingface.co/settings/token
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access public models or datasets.
  warnings.warn(
config.json:  100%                                          967/967  [00:00<00:00,  17.1kB/s]

model.safetensors.index.json:  100%                                 16.5k/16.5k  [00:00<00:00,  547kB/s]

Downloading  shards:  100%                                        2/2  [01:06<00:00,  32.81s/it]

model-00001-of-00002.safetensors:  100%                            4.97G/4.97G  [00:34<00:00,  242MB/s]

model-00002-of-00002.safetensors:  100%                            2.67G/2.67G  [00:31<00:00,  57.0MB/s]

Loading  checkpoint  shards:  100%                                   2/2  [00:35<00:00,  16.93s/it]

generation_config.json:  100%                                     181/181  [00:00<00:00,  14.7kB/s]
```

```
print(model.get_memory_footprint()/1e6)
```

```
2206.347264
```

```
model
```

```
Phi3ForCausalLM(
  (model): Phi3Model(
    (embed_tokens): Embedding(32064, 3072, padding_idx=32000)
    (embed_dropout): Dropout(p=0.0, inplace=False)
    (layers): ModuleList(
      (0-31): 32 x Phi3DecoderLayer(
        (self_attn): Phi3SdpaAttention(
          (o_proj): Linear4bit(in_features=3072, out_features=3072, bias=False)
          (qkv_proj): Linear4bit(in_features=3072, out_features=9216, bias=False)
          (rotary_emb): Phi3RotaryEmbedding()
        )
        (mlp): Phi3MLP(
          (gate_up_proj): Linear4bit(in_features=3072, out_features=16384, bias=False)
          (down_proj): Linear4bit(in_features=8192, out_features=3072, bias=False)
          (activation_fn): SiLU()
        )
        (input_layernorm): Phi3RMSNorm((3072,), eps=1e-05)
        (resid_attn_dropout): Dropout(p=0.0, inplace=False)
        (resid_mlp_dropout): Dropout(p=0.0, inplace=False)
        (post_attention_layernorm): Phi3RMSNorm((3072,), eps=1e-05)
      )
    )
    (norm): Phi3RMSNorm((3072,), eps=1e-05)
  )
  (lm_head): Linear(in_features=3072, out_features=32064, bias=False)
)
```

```
model = prepare_model_for_kbit_training(model)

config = LoraConfig(
    # the rank of the adapter, the lower the fewer parameters you'll need to train
    r=8,
    lora_alpha=16, # multiplier, usually 2*r
    bias="none",
    lora_dropout=0.05,
    task_type="CAUSAL_LM",
    # Newer models, such as Phi-3 at time of writing, may require
    # manually setting target modules
    target_modules=['o_proj', 'qkv_proj', 'gate_up_proj', 'down_proj'],
)
model = get_peft_model(model, config)
model
```

```
PeftModelForCausalLM(
  (base_model): LoraModel(
    (model): Phi3ForCausalLM(
      (model): Phi3Model(
        (embed_tokens): Embedding(32064, 3072, padding_idx=32000)
        (embed_dropout): Dropout(p=0.0, inplace=False)
        (layers): ModuleList(
          (0-31): 32 x Phi3DecoderLayer(
            (self_attn): Phi3SdpaAttention(
              (o_proj): lora.Linear4bit(
                (base_layer): Linear4bit(in_features=3072, out_features=3072, bias=False)
                (lora_dropout): ModuleDict(
                  (default): Dropout(p=0.05, inplace=False)
                )
                (lora_A): ModuleDict(
                  (default): Linear(in_features=3072, out_features=8, bias=False)
                )
                (lora_B): ModuleDict(
                  (default): Linear(in_features=8, out_features=3072, bias=False)
                )
                (lora_embedding_A): ParameterDict()
                (lora_embedding_B): ParameterDict()
                (lora_magnitude_vector): ModuleDict()
              )
              (qkv_proj): lora.Linear4bit(
                (base_layer): Linear4bit(in_features=3072, out_features=9216, bias=False)
                (lora_dropout): ModuleDict(
                  (default): Dropout(p=0.05, inplace=False)
                )
                (lora_A): ModuleDict(
                  (default): Linear(in_features=3072, out_features=8, bias=False)
                )
                (lora_B): ModuleDict(
                  (default): Linear(in_features=8, out_features=9216, bias=False)
                )
                (lora_embedding_A): ParameterDict()
                (lora_embedding_B): ParameterDict()
                (lora_magnitude_vector): ModuleDict()
              )
              (rotary_emb): Phi3RotaryEmbedding()
            )
            (mlp): Phi3MLP(
              (gate_up_proj): lora.Linear4bit(
                (base_layer): Linear4bit(in_features=3072, out_features=16384, bias=False)
                (lora_dropout): ModuleDict(
                  (default): Dropout(p=0.05, inplace=False)
                )
                (lora_A): ModuleDict(
                  (default): Linear(in_features=3072, out_features=8, bias=False)
                )
                (lora_B): ModuleDict(
                  (default): Linear(in_features=8, out_features=16384, bias=False)
                )
                (lora_embedding_A): ParameterDict()
                (lora_embedding_B): ParameterDict()
                (lora_magnitude_vector): ModuleDict()
              )
              (down_proj): lora.Linear4bit(
```

```
print(model.get_memory_footprint()/1e6)
```

```
2651.080704
```

```
train_p, tot_p = model.get_nb_trainable_parameters()
print(f'Trainable parameters:      {train_p/1e6:.2f}M')
print(f'Total parameters:          {tot_p/1e6:.2f}M')
print(f'% of trainable parameters: {100*train_p/tot_p:.2f}%')
```

```
Trainable parameters:      12.58M
Total parameters:          3833.66M
% of trainable parameters: 0.33%
```

```
dataset = load_dataset("dvgodoy/yoda_sentences", split="train")
dataset
```

```
README.md: 100%                                    531/531 [00:00<00:00, 40.3kB/s]

sentences.csv: 100%                                98.4k/98.4k [00:00<00:00, 9.54MB/s]

Generating train split: 100%                       720/720 [00:00<00:00, 5283.26 examples/s]

Dataset({
    features: ['sentence', 'translation', 'translation_extra'],
    num_rows: 720
})
```

```
dataset[0]
```

```
{'sentence': 'The birch canoe slid on the smooth planks.',
 'translation': 'On the smooth planks, the birch canoe slid.',
 'translation_extra': 'On the smooth planks, the birch canoe slid. Yes, hrrrm.'}
```

```
dataset = dataset.rename_column("sentence", "prompt")
dataset = dataset.rename_column("translation_extra", "completion")
dataset = dataset.remove_columns(["translation"])
dataset
```

```
Dataset({
    features: ['prompt', 'completion'],
    num_rows: 720
})
```

```
dataset[0]
```

```
{'prompt': 'The birch canoe slid on the smooth planks.',
 'completion': 'On the smooth planks, the birch canoe slid. Yes, hrrrm.'}
```

```
type(dataset)
```

```
datasets.arrow_dataset.Dataset
def __init__(arrow_table: Table, info: Optional[DatasetInfo]=None, split: Optional[NamedSplit]=None,
indices_table: Optional[Table]=None, fingerprint: Optional[str]=None)

A Dataset backed by an Arrow table.
```

```
messages = [
    {"role": "user", "content": dataset[0]['prompt']},
    {"role": "assistant", "content": dataset[0]['completion']}
]
messages
```

```
[{'role': 'user', 'content': 'The birch canoe slid on the smooth planks.'},
 {'role': 'assistant',
  'content': 'On the smooth planks, the birch canoe slid. Yes, hrrrm.'}]
```

```
# Adapted from trl.extras.dataset_formatting.instructions_formatting_function
# Converts dataset from prompt/completion format (not supported anymore)
# to the conversational format
def format_dataset(examples):
    if isinstance(examples["prompt"], list):
        output_texts = []
        for i in range(len(examples["prompt"])):
            converted_sample = [
                {"role": "user", "content": examples["prompt"][i]},
                {"role": "assistant", "content": examples["completion"][i]},
            ]
            output_texts.append(converted_sample)
        return {'messages': output_texts}
    else:
        converted_sample = [
            {"role": "user", "content": examples["prompt"]},
            {"role": "assistant", "content": examples["completion"]},
        ]
        return {'messages': converted_sample}

dataset = dataset.map(format_dataset).remove_columns(['prompt', 'completion'])
```

```
Map: 100%                                   720/720 [00:00<00:00, 11304.13 examples/s]
```

```
tokenizer = AutoTokenizer.from_pretrained(repo_id)
tokenizer.chat_template
```

```
tokenizer_config.json: 100%                                 3.44k/3.44k [00:00<00:00, 323kB/s]

tokenizer.model: 100%                                   500k/500k [00:00<00:00, 40.5MB/s]

tokenizer.json: 100%                                  1.94M/1.94M [00:00<00:00, 3.82MB/s]

added_tokens.json: 100%                                     306/306 [00:00<00:00, 29.9kB/s]

special_tokens_map.json: 100%                                   599/599 [00:00<00:00, 53.0kB/s]

'{% for message in messages %}{% if message['role'] == 'system' %}{{'<|system|>\n' + message['content'] + '<|end|>
\n'}}{% elif message['role'] == 'user' %}{{'<|user|>\n' + message['content'] + '<|end|>\n'}}{% elif message['role'] =
= 'assistant' %}{{'<|assistant|>\n' + message['content'] + '<|end|>\n'}}{% endif %}{% endfor %}{% if add_generation_p
rompt %}{{ '<|assistant|>\n' }}{% else %}{{ eos_token }}{% endif %}'
```

```
print(tokenizer.apply_chat_template(messages, tokenize=False))
```

```
<|user|>
The birch canoe slid on the smooth planks.<|end|>
<|assistant|>
On the smooth planks, the birch canoe slid. Yes, hrrrm.<|end|>
<|endoftext|>
```

```
tokenizer.pad_token = tokenizer.unk_token
tokenizer.pad_token_id = tokenizer.unk_token_id


sft_config = SFTConfig(
    ## GROUP 1: Memory usage
    # These arguments will squeeze the most out of your GPU's RAM
    # Checkpointing
    gradient_checkpointing=True,
    # this saves a LOT of memory
    # Set this to avoid exceptions in newer versions of PyTorch
    gradient_checkpointing_kwargs={'use_reentrant': False},
    # Gradient Accumulation / Batch size
    # Actual batch (for updating) is same (1x) as micro-batch size
    gradient_accumulation_steps=1,
    # The initial (micro) batch size to start off with
    per_device_train_batch_size=16,
    # If batch size would cause OOM, halves its size until it works
    auto_find_batch_size=True,

    ## GROUP 2: Dataset-related
    max_seq_length=64,
    # Dataset
    # packing a dataset means no padding is needed
    packing=True,

    ## GROUP 3: These are typical training parameters
    num_train_epochs=3,
    learning_rate=3e-4,
    # Optimizer
    # 8-bit Adam optimizer - doesn't help much if you're using LoRA!
    optim='paged_adamw_8bit',

    ## GROUP 4: Logging parameters
    logging_steps=10,
    logging_dir='/content/sample_data/logs',
    output_dir='/content/sample_data/phi3-mini-yoda-adapter',
    report_to='none'
)

trainer = SFTTrainer(
    model=model,
    processing_class=tokenizer,
    args=sft_config,
    train_dataset=dataset,
)
```

```
Generating train split:    351/0 [00:00<00:00,   8.05 examples/s]
/usr/local/lib/python3.11/dist-packages/trl/trainer/sft_trainer.py:403: UserWarning: You passed a processing_class wit
  warnings.warn(
```

```
dl = trainer.get_train_dataloader()
batch = next(iter(dl))
```

```
batch['input_ids'][0], batch['labels'][0]
```

```
(tensor([29892,   278,   270,  6472,   310,   278, 17251,  1258, 29889,   379,
          1758,  4317, 29889, 32007, 32000, 32000, 32010,   450,   696,   412,
           674,  7868,   278,  9881,  8277,   472,  2748, 29889, 32007, 32001,
         29672,   278,  9881,  8277,   472,  2748, 29892,   278,   696,   412,
           674, 29889, 32007, 32000, 32000, 32010,  7753,   263,   925,  4556,
          4225,  3081,   304,  5401, 29889, 32007, 32001,  9206,   304,  5401,
         29892,  1584,   263,   925], device='cuda:0'),
 tensor([29892,   278,   270,  6472,   310,   278, 17251,  1258, 29889,   379,
          1758,  4317, 29889, 32007, 32000, 32000, 32010,   450,   696,   412,
           674,  7868,   278,  9881,  8277,   472,  2748, 29889, 32007, 32001,
         29672,   278,  9881,  8277,   472,  2748, 29892,   278,   696,   412,
           674, 29889, 32007, 32000, 32000, 32010,  7753,   263,   925,  4556,
          4225,  3081,   304,  5401, 29889, 32007, 32001,  9206,   304,  5401,
         29892,  1584,   263,   925], device='cuda:0'))
```

```
trainer.train()
```

```
`use_cache=True` is incompatible with gradient checkpointing. Setting `use_cache=False`...
```

[66/66 07:49, Epoch 3/3]

Step	Training Loss
10	3.033500
20	1.812400
30	1.586200
40	1.497400
50	1.453600
60	1.382900

```
TrainOutput(global_step=66, training_loss=1.7526665167375044, metrics={'train_runtime': 476.6257,
'train_samples_per_second': 2.209, 'train_steps_per_second': 0.138, 'total_flos': 1510320166797312.0, 'train_loss':
```

```python
def gen_prompt(tokenizer, sentence):
    converted_sample = [
        {"role": "user", "content": sentence},
    ]
    prompt = tokenizer.apply_chat_template(converted_sample,
                                           tokenize=False,
                                           add_generation_prompt=True)
    return prompt

sentence = 'The Force is strong in you!'
prompt = gen_prompt(tokenizer, sentence)
print(prompt)
```

```
<|user|>
The Force is strong in you!<|end|>
<|assistant|>
```

```python
def generate(model, tokenizer, prompt, max_new_tokens=64, skip_special_tokens=False):
    tokenized_input = tokenizer(prompt, add_special_tokens=False, return_tensors="pt"

    model.eval()
    generation_output = model.generate(**tokenized_input,
                                       eos_token_id=tokenizer.eos_token_id,
                                       max_new_tokens=max_new_tokens)

    output = tokenizer.batch_decode(generation_output,
                                    skip_special_tokens=skip_special_tokens)
    return output[0]

print(generate(model, tokenizer, prompt))
```

```
<|user|> The Force is strong in you!<|end|><|assistant|> Strong in you, the Force is.<|end|><|endoftext|>
```

```python
trainer.save_model('/content/final_adapter')
```

```python
os.listdir('/content/final_adapter')
```

```
['training_args.bin',
 'tokenizer.model',
 'added_tokens.json',
 'special_tokens_map.json',
 'README.md',
 'adapter_config.json',
 'adapter_model.safetensors',
 'tokenizer.json',
 'tokenizer_config.json']
```