# Using 🤗 PEFT & bitsandbytes to finetune a LoRa checkpoint

!pip install -q bitsandbytes datasets accelerate loralib  
!pip install -q git+https://github.com/huggingface/transformers.git@main git+https://github.com/huggingface/peft.git

━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 76.3/76.3 MB 20.0 MB/s eta 0:00:00  
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 469.0/469.0 KB 11.8 MB/s eta 0:00:00  
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 212.8/212.8 KB 14.7 MB/s eta 0:00:00  
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.0/1.0 MB 8.5 MB/s eta 0:00:00  
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 212.2/212.2 KB 20.3 MB/s eta 0:00:00  
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 110.5/110.5 KB 12.1 MB/s eta 0:00:00  
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 132.9/132.9 KB 16.9 MB/s eta 0:00:00  
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 199.2/199.2 KB 20.7 MB/s eta 0:00:00  
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 158.8/158.8 KB 16.8 MB/s eta 0:00:00  
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 114.2/114.2 KB 13.9 MB/s eta 0:00:00  
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 264.6/264.6 KB 26.7 MB/s eta 0:00:00  
ents to build wheel ... etadata (pyproject.toml) ... ents to build wheel ... etadata (pyproject.toml) ... ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 7.6/7.6 MB 78.8 MB/s eta 0:00:00  
ers (pyproject.toml) ... l) ...

from huggingface\_hub import notebook\_login  
  
notebook\_login()

{"model\_id":"e7927778c808426f8e7d571ff637af93","version\_major":2,"version\_minor":0}

!nvidia-smi -L

GPU 0: NVIDIA A100-SXM4-40GB (UUID: GPU-e422d711-e39a-eaa9-2988-783d144f55c1)

### Setup the model

import os  
os.environ["CUDA\_VISIBLE\_DEVICES"]="0"  
import torch  
import torch.nn as nn  
import bitsandbytes as bnb  
from transformers import AutoTokenizer, AutoConfig, AutoModelForCausalLM  
  
model = AutoModelForCausalLM.from\_pretrained(  
 "bigscience/bloom-7b1",  
 load\_in\_8bit=True,  
 device\_map='auto',  
)  
  
tokenizer = AutoTokenizer.from\_pretrained("bigscience/bloom-7b1")

===================================BUG REPORT===================================  
Welcome to bitsandbytes. For bug reports, please submit your error trace to: https://github.com/TimDettmers/bitsandbytes/issues  
================================================================================  
CUDA\_SETUP: WARNING! libcudart.so not found in any environmental path. Searching /usr/local/cuda/lib64...  
CUDA SETUP: CUDA runtime path found: /usr/local/cuda/lib64/libcudart.so  
CUDA SETUP: Highest compute capability among GPUs detected: 8.0  
CUDA SETUP: Detected CUDA version 118  
CUDA SETUP: Loading binary /usr/local/lib/python3.9/dist-packages/bitsandbytes/libbitsandbytes\_cuda118.so...

/usr/local/lib/python3.9/dist-packages/bitsandbytes/cuda\_setup/main.py:136: UserWarning: /usr/lib64-nvidia did not contain libcudart.so as expected! Searching further paths...  
 warn(msg)  
/usr/local/lib/python3.9/dist-packages/bitsandbytes/cuda\_setup/main.py:136: UserWarning: WARNING: The following directories listed in your path were found to be non-existent: {PosixPath('/sys/fs/cgroup/memory.events /var/colab/cgroup/jupyter-children/memory.events')}  
 warn(msg)  
/usr/local/lib/python3.9/dist-packages/bitsandbytes/cuda\_setup/main.py:136: UserWarning: WARNING: The following directories listed in your path were found to be non-existent: {PosixPath('//colab.research.google.com/tun/m/cc48301118ce562b961b3c22d803539adc1e0c19/gpu-a100-s-2gmbmgolbub5k --tunnel\_background\_save\_delay=10s --tunnel\_periodic\_background\_save\_frequency=30m0s --enable\_output\_coalescing=true --output\_coalescing\_required=true'), PosixPath('--listen\_host=172.28.0.12 --target\_host=172.28.0.12 --tunnel\_background\_save\_url=https')}  
 warn(msg)  
/usr/local/lib/python3.9/dist-packages/bitsandbytes/cuda\_setup/main.py:136: UserWarning: WARNING: The following directories listed in your path were found to be non-existent: {PosixPath('/env/python')}  
 warn(msg)  
/usr/local/lib/python3.9/dist-packages/bitsandbytes/cuda\_setup/main.py:136: UserWarning: WARNING: The following directories listed in your path were found to be non-existent: {PosixPath('module'), PosixPath('//ipykernel.pylab.backend\_inline')}  
 warn(msg)

{"model\_id":"a4aca3caf668475ea0393f4b30759dce","version\_major":2,"version\_minor":0}

Overriding torch\_dtype=None with `torch\_dtype=torch.float16` due to requirements of `bitsandbytes` to enable model loading in mixed int8. Either pass torch\_dtype=torch.float16 or don't pass this argument at all to remove this warning.

{"model\_id":"864806aa8db544a685e7645afd1cc6fd","version\_major":2,"version\_minor":0}

{"model\_id":"5d357d1ee5434d55a8d2ad84d67e7fde","version\_major":2,"version\_minor":0}

{"model\_id":"a54af21bebad43919d441f7439c6f3ea","version\_major":2,"version\_minor":0}

{"model\_id":"1dc45ac9a1304a4cac48dea1ed62198d","version\_major":2,"version\_minor":0}

{"model\_id":"a9d8f269a95045f097a1affced473508","version\_major":2,"version\_minor":0}

{"model\_id":"b74befb90b5a42f4a30e3fb55fc665e4","version\_major":2,"version\_minor":0}

{"model\_id":"811e1ab385be434a8a9e0076d0f5a959","version\_major":2,"version\_minor":0}

{"model\_id":"6bd93937c3a0434d87ba82e97054c7f4","version\_major":2,"version\_minor":0}

### Freezing the original weights

for param in model.parameters():  
 param.requires\_grad = False # freeze the model - train adapters later  
 if param.ndim == 1:  
 # cast the small parameters (e.g. layernorm) to fp32 for stability  
 param.data = param.data.to(torch.float32)  
  
model.gradient\_checkpointing\_enable() # reduce number of stored activations  
model.enable\_input\_require\_grads()  
  
class CastOutputToFloat(nn.Sequential):  
 def forward(self, x): return super().forward(x).to(torch.float32)  
model.lm\_head = CastOutputToFloat(model.lm\_head)

### Setting up the LoRa Adapters

def print\_trainable\_parameters(model):  
 """  
 Prints the number of trainable parameters in the model.  
 """  
 trainable\_params = 0  
 all\_param = 0  
 for \_, param in model.named\_parameters():  
 all\_param += param.numel()  
 if param.requires\_grad:  
 trainable\_params += param.numel()  
 print(  
 f"trainable params: {trainable\_params} || all params: {all\_param} || trainable%: {100 \* trainable\_params / all\_param}"  
 )

from peft import LoraConfig, get\_peft\_model  
  
config = LoraConfig(  
 r=16, #attention heads  
 lora\_alpha=32, #alpha scaling  
 # target\_modules=["q\_proj", "v\_proj"], #if you know the  
 lora\_dropout=0.05,  
 bias="none",  
 task\_type="CAUSAL\_LM" # set this for CLM or Seq2Seq  
)  
  
model = get\_peft\_model(model, config)  
print\_trainable\_parameters(model)

trainable params: 7864320 || all params: 7076880384 || trainable%: 0.11112693126452029

## Data

import transformers  
from datasets import load\_dataset  
data = load\_dataset("Abirate/english\_quotes")

{"model\_id":"ceb7dff5fa364f4689688fd85e7ed4d0","version\_major":2,"version\_minor":0}

Downloading and preparing dataset json/Abirate--english\_quotes to /root/.cache/huggingface/datasets/Abirate\_\_\_json/Abirate--english\_quotes-6e72855d06356857/0.0.0/0f7e3662623656454fcd2b650f34e886a7db4b9104504885bd462096cc7a9f51...

{"model\_id":"64f3155f66d84ed08d945abcb7e9a9df","version\_major":2,"version\_minor":0}

{"model\_id":"9518c8aa7e3d4affbbf2dc3a0e7eb221","version\_major":2,"version\_minor":0}

{"model\_id":"f0969e2407164dd89c5f3c157c5c4398","version\_major":2,"version\_minor":0}

{"model\_id":"00e70866c6614ec2aadcd6c725676f51","version\_major":2,"version\_minor":0}

Dataset json downloaded and prepared to /root/.cache/huggingface/datasets/Abirate\_\_\_json/Abirate--english\_quotes-6e72855d06356857/0.0.0/0f7e3662623656454fcd2b650f34e886a7db4b9104504885bd462096cc7a9f51. Subsequent calls will reuse this data.

{"model\_id":"23e584a31f1146cfbfc230cf1f869227","version\_major":2,"version\_minor":0}

def merge\_columns(example):  
 example["prediction"] = example["quote"] + " ->: " + str(example["tags"])  
 return example  
  
data['train'] = data['train'].map(merge\_columns)  
data['train']["prediction"][:5]

{"model\_id":"93dcef37faa141a3b6588686f0b209c1","version\_major":2,"version\_minor":0}

["“Be yourself; everyone else is already taken.” ->: ['be-yourself', 'gilbert-perreira', 'honesty', 'inspirational', 'misattributed-oscar-wilde', 'quote-investigator']",  
 "“I'm selfish, impatient and a little insecure. I make mistakes, I am out of control and at times hard to handle. But if you can't handle me at my worst, then you sure as hell don't deserve me at my best.” ->: ['best', 'life', 'love', 'mistakes', 'out-of-control', 'truth', 'worst']",  
 "“Two things are infinite: the universe and human stupidity; and I'm not sure about the universe.” ->: ['human-nature', 'humor', 'infinity', 'philosophy', 'science', 'stupidity', 'universe']",  
 "“So many books, so little time.” ->: ['books', 'humor']",  
 "“A room without books is like a body without a soul.” ->: ['books', 'simile', 'soul']"]

data['train'][0]

{'quote': '“Be yourself; everyone else is already taken.”',  
 'author': 'Oscar Wilde',  
 'tags': ['be-yourself',  
 'gilbert-perreira',  
 'honesty',  
 'inspirational',  
 'misattributed-oscar-wilde',  
 'quote-investigator'],  
 'prediction': "“Be yourself; everyone else is already taken.” ->: ['be-yourself', 'gilbert-perreira', 'honesty', 'inspirational', 'misattributed-oscar-wilde', 'quote-investigator']"}

data = data.map(lambda samples: tokenizer(samples['prediction']), batched=True)

{"model\_id":"77e9cc01e6bb4170936bc7592902ee20","version\_major":2,"version\_minor":0}

data

DatasetDict({  
 train: Dataset({  
 features: ['quote', 'author', 'tags', 'prediction', 'input\_ids', 'attention\_mask'],  
 num\_rows: 2508  
 })  
})

### Training

trainer = transformers.Trainer(  
 model=model,  
 train\_dataset=data['train'],  
 args=transformers.TrainingArguments(  
 per\_device\_train\_batch\_size=4,  
 gradient\_accumulation\_steps=4,  
 warmup\_steps=100,  
 max\_steps=200,  
 learning\_rate=2e-4,  
 fp16=True,  
 logging\_steps=1,  
 output\_dir='outputs'  
 ),  
 data\_collator=transformers.DataCollatorForLanguageModeling(tokenizer, mlm=False)  
)  
model.config.use\_cache = False # silence the warnings. Please re-enable for inference!  
trainer.train()

/usr/local/lib/python3.9/dist-packages/transformers/optimization.py:391: FutureWarning: This implementation of AdamW is deprecated and will be removed in a future version. Use the PyTorch implementation torch.optim.AdamW instead, or set `no\_deprecation\_warning=True` to disable this warning  
 warnings.warn(  
You're using a BloomTokenizerFast tokenizer. Please note that with a fast tokenizer, using the `\_\_call\_\_` method is faster than using a method to encode the text followed by a call to the `pad` method to get a padded encoding.  
/usr/local/lib/python3.9/dist-packages/bitsandbytes/autograd/\_functions.py:298: UserWarning: MatMul8bitLt: inputs will be cast from torch.float32 to float16 during quantization  
 warnings.warn(f"MatMul8bitLt: inputs will be cast from {A.dtype} to float16 during quantization")

<IPython.core.display.HTML object>

TrainOutput(global\_step=200, training\_loss=2.327150729894638, metrics={'train\_runtime': 771.2822, 'train\_samples\_per\_second': 4.149, 'train\_steps\_per\_second': 0.259, 'total\_flos': 1.3172999996964864e+16, 'train\_loss': 2.327150729894638, 'epoch': 1.28})

## Share adapters on the 🤗 Hub

model.push\_to\_hub("samwit/bloom-7b1-lora-tagger",  
 use\_auth\_token=True,  
 commit\_message="basic training",  
 private=True)

{"model\_id":"0ec66e82bc8247e98e98686c051f89e9","version\_major":2,"version\_minor":0}

{"model\_id":"95f2c69bcde84296b9f505149a5bf23e","version\_major":2,"version\_minor":0}

CommitInfo(commit\_url='https://huggingface.co/samwit/bloom-7b1-lora-tagger/commit/62cfae6c87a7d657b2bd3e6e2abac2d5a7d07caf', commit\_message='basic training', commit\_description='', oid='62cfae6c87a7d657b2bd3e6e2abac2d5a7d07caf', pr\_url=None, pr\_revision=None, pr\_num=None)

## Load adapters from the Hub

import torch  
from peft import PeftModel, PeftConfig  
from transformers import AutoModelForCausalLM, AutoTokenizer  
  
peft\_model\_id = "samwit/bloom-7b1-lora-tagger"  
config = PeftConfig.from\_pretrained(peft\_model\_id)  
model = AutoModelForCausalLM.from\_pretrained(config.base\_model\_name\_or\_path, return\_dict=True, load\_in\_8bit=True, device\_map='auto')  
tokenizer = AutoTokenizer.from\_pretrained(config.base\_model\_name\_or\_path)  
  
# Load the Lora model  
model = PeftModel.from\_pretrained(model, peft\_model\_id)

===================================BUG REPORT===================================  
Welcome to bitsandbytes. For bug reports, please submit your error trace to: https://github.com/TimDettmers/bitsandbytes/issues  
================================================================================  
CUDA\_SETUP: WARNING! libcudart.so not found in any environmental path. Searching /usr/local/cuda/lib64...  
CUDA SETUP: CUDA runtime path found: /usr/local/cuda/lib64/libcudart.so  
CUDA SETUP: Highest compute capability among GPUs detected: 8.0  
CUDA SETUP: Detected CUDA version 118  
CUDA SETUP: Loading binary /usr/local/lib/python3.9/dist-packages/bitsandbytes/libbitsandbytes\_cuda118.so...

/usr/local/lib/python3.9/dist-packages/bitsandbytes/cuda\_setup/main.py:136: UserWarning: /usr/lib64-nvidia did not contain libcudart.so as expected! Searching further paths...  
 warn(msg)  
/usr/local/lib/python3.9/dist-packages/bitsandbytes/cuda\_setup/main.py:136: UserWarning: WARNING: The following directories listed in your path were found to be non-existent: {PosixPath('/sys/fs/cgroup/memory.events /var/colab/cgroup/jupyter-children/memory.events')}  
 warn(msg)  
/usr/local/lib/python3.9/dist-packages/bitsandbytes/cuda\_setup/main.py:136: UserWarning: WARNING: The following directories listed in your path were found to be non-existent: {PosixPath('--listen\_host=172.28.0.12 --target\_host=172.28.0.12 --tunnel\_background\_save\_url=https'), PosixPath('//colab.research.google.com/tun/m/cc48301118ce562b961b3c22d803539adc1e0c19/gpu-a100-s-c1xr75l0apxr --tunnel\_background\_save\_delay=10s --tunnel\_periodic\_background\_save\_frequency=30m0s --enable\_output\_coalescing=true --output\_coalescing\_required=true')}  
 warn(msg)  
/usr/local/lib/python3.9/dist-packages/bitsandbytes/cuda\_setup/main.py:136: UserWarning: WARNING: The following directories listed in your path were found to be non-existent: {PosixPath('/env/python')}  
 warn(msg)  
/usr/local/lib/python3.9/dist-packages/bitsandbytes/cuda\_setup/main.py:136: UserWarning: WARNING: The following directories listed in your path were found to be non-existent: {PosixPath('module'), PosixPath('//ipykernel.pylab.backend\_inline')}  
 warn(msg)

{"model\_id":"9853102a87bb48e4b5e44659344cf8c6","version\_major":2,"version\_minor":0}

{"model\_id":"bf2681d51529472594962ff2479e093e","version\_major":2,"version\_minor":0}

Overriding torch\_dtype=None with `torch\_dtype=torch.float16` due to requirements of `bitsandbytes` to enable model loading in mixed int8. Either pass torch\_dtype=torch.float16 or don't pass this argument at all to remove this warning.

{"model\_id":"c2737f2d82f74387a79973ef24c12361","version\_major":2,"version\_minor":0}

{"model\_id":"441b9419c77b4382a698d8321e38873c","version\_major":2,"version\_minor":0}

{"model\_id":"dbef3e1fcdd2416fa91daca4498de075","version\_major":2,"version\_minor":0}

{"model\_id":"6ae9608924f84327bc21291a01317599","version\_major":2,"version\_minor":0}

{"model\_id":"35734133f5364871a179a222d64363d0","version\_major":2,"version\_minor":0}

{"model\_id":"a80f36f36cd84d1fa1ac2b8ed7397b86","version\_major":2,"version\_minor":0}

{"model\_id":"bb266c20cd5d44028229c34b91af259d","version\_major":2,"version\_minor":0}

{"model\_id":"2a12754ffc2044c7abf5e16b26e6a289","version\_major":2,"version\_minor":0}

{"model\_id":"2744a3554c4744fba09b6183ba03572d","version\_major":2,"version\_minor":0}

## Inference

batch = tokenizer("“Training models with PEFT and LoRa is cool” ->: ", return\_tensors='pt')  
  
with torch.cuda.amp.autocast():  
 output\_tokens = model.generate(\*\*batch, max\_new\_tokens=50)  
  
print('\n\n', tokenizer.decode(output\_tokens[0], skip\_special\_tokens=True))

“Training models with PEFT and LoRa is cool” ->: ['training', 'teaching']  
  
A:  
  
I think the best way to describe it is to say that it is a combination of the two. The idea is that you can train a model on a dataset, and then you can use the