```
!pip install bitsandbytes>=0.45.3

from huggingface_hub import login

# Hugging face token
from google.colab import userdata
HF_TOKEN = userdata.get('HF_TOKEN')

# Log in to Hugging Face Hub
login(token=HF_TOKEN)

from huggingface_hub import snapshot_download

sql_lora_path =
snapshot_download(repo_id="google-cloud-partnership/gemma-2-2b-it-
lora-sql")

{"model_id":"2653543333c14e5980061bb195a3dce6","version_major":2,"vers
ion_minor":0}

import torch
from vllm import LLM, SamplingParams
from vllm.lora.request import LoRARequest

prompts = [
    "The future of AI is",
]
sampling_params = SamplingParams(temperature=0.3, top_p=0.95,
max_tokens=256)

model_id = "google/gemma-2-2b"
llm = LLM(model=model_id,dtype=torch.float16,enable_lora=True,
trust_remote_code=True,
          kv_cache_dtype="fp8",calculate_kv_scales=True , \
          quantization="bitsandbytes", load_format="bitsandbytes")

outputs = llm.generate(prompts,sampling_params,
                       lora_request=LoRARequest("sql_adapter", 1,
sql_lora_path))

for output in outputs:
    prompt = output.prompt
    generated_text = output.outputs[0].text
    print(f"Prompt: {prompt!r}, Generated text: {generated_text!r}")

INFO 05-03 13:07:29 [__init__.py:239] Automatically detected platform
cuda.
INFO 05-03 13:07:39 [config.py:2968] Downcasting torch.float32 to
torch.float16.
INFO 05-03 13:07:54 [config.py:717] This model supports multiple
tasks: {'classify', 'reward', 'embed', 'score', 'generate'}.
```

```
Defaulting to 'generate'.
WARNING 05-03 13:07:54 [config.py:830] bitsandbytes quantization is
not fully optimized yet. The speed can be slower than non-quantized
models.
WARNING 05-03 13:07:54 [arg_utils.py:1658] Compute Capability < 8.0 is
not supported by the V1 Engine. Falling back to V0.
INFO 05-03 13:07:54 [config.py:1403] Using fp8 data type to store kv
cache. It reduces the GPU memory footprint and boosts the performance.
Meanwhile, it may cause accuracy drop without a proper scaling factor
INFO 05-03 13:07:56 [llm_engine.py:240] Initializing a V0 LLM engine
(v0.8.5.post1) with config: model='google/gemma-2-2b',
speculative_config=None, tokenizer='google/gemma-2-2b',
skip_tokenizer_init=False, tokenizer_mode=auto, revision=None,
override_neuron_config=None, tokenizer_revision=None,
trust_remote_code=True, dtype=torch.float16, max_seq_len=8192,
download_dir=None, load_format=LoadFormat.BITSANDBYTES,
tensor_parallel_size=1, pipeline_parallel_size=1,
disable_custom_all_reduce=False, quantization=bitsandbytes,
enforce_eager=False, kv_cache_dtype=fp8,  device_config=cuda,
decoding_config=DecodingConfig(guided_decoding_backend='auto',
reasoning_backend=None),
observability_config=ObservabilityConfig(show_hidden_metrics=False,
otlp_traces_endpoint=None, collect_model_forward_time=False,
collect_model_execute_time=False), seed=None,
served_model_name=google/gemma-2-2b, num_scheduler_steps=1,
multi_step_stream_outputs=True, enable_prefix_caching=None,
chunked_prefill_enabled=False, use_async_output_proc=True,
disable_mm_preprocessor_cache=False, mm_processor_kwargs=None,
pooler_config=None, compilation_config={"splitting_ops":
[],"compile_sizes":[],"cudagraph_capture_sizes":
[256,248,240,232,224,216,208,200,192,184,176,168,160,152,144,136,128,1
20,112,104,96,88,80,72,64,56,48,40,32,24,16,8,4,2,1],"max_capture_size
":256}, use_cached_outputs=False,
INFO 05-03 13:07:58 [cuda.py:240] Cannot use FlashAttention-2 backend
for Volta and Turing GPUs.
INFO 05-03 13:07:58 [cuda.py:289] Using XFormers backend.
INFO 05-03 13:07:59 [parallel_state.py:1004] rank 0 in world size 1 is
assigned as DP rank 0, PP rank 0, TP rank 0
INFO 05-03 13:07:59 [model_runner.py:1108] Starting to load model
google/gemma-2-2b...
WARNING 05-03 13:07:59 [xformers.py:398] XFormers does not support
logits soft cap. Outputs may be slightly off.
INFO 05-03 13:07:59 [loader.py:1187] Loading weights with BitsAndBytes
quantization. May take a while ...
INFO 05-03 13:08:00 [weight_utils.py:265] Using model weights format
['*.safetensors']

{"model_id":"a208cabe8ba14fb28dac335f08dab005","version_major":2,"vers
ion_minor":0}
```

```
INFO 05-03 13:08:47 [punica_selector.py:18] Using PunicaWrapperGPU.
INFO 05-03 13:08:47 [model_runner.py:1140] Model loading took 2.1314
GiB and 47.851604 seconds
INFO 05-03 13:09:07 [worker.py:287] Memory profiling takes 19.33
seconds
INFO 05-03 13:09:07 [worker.py:287] the current vLLM instance can use
total_gpu_memory (14.74GiB) x gpu_memory_utilization (0.90) = 13.27GiB
INFO 05-03 13:09:07 [worker.py:287] model weights take 2.13GiB;
non_torch_memory takes 0.05GiB; PyTorch activation peak memory takes
2.36GiB; the rest of the memory reserved for KV Cache is 8.72GiB.
INFO 05-03 13:09:07 [executor_base.py:112] # cuda blocks: 10995, # CPU
blocks: 5041
INFO 05-03 13:09:07 [executor_base.py:117] Maximum concurrency for
8192 tokens per request: 21.47x
INFO 05-03 13:09:12 [model_runner.py:1450] Capturing cudagraphs for
decoding. This may lead to unexpected consequences if the model is not
static. To run the model in eager mode, set 'enforce_eager=True' or
use '--enforce-eager' in the CLI. If out-of-memory error occurs during
cudagraph capture, consider decreasing `gpu_memory_utilization` or
switching to eager mode. You can also reduce the `max_num_seqs` as
needed to decrease memory usage.

{"model_id":"0f13278872114e5f9aaf95a86882e596","version_major":2,"vers
ion_minor":0}

INFO 05-03 13:11:19 [model_runner.py:1592] Graph capturing finished in
127 secs, took 0.78 GiB
INFO 05-03 13:11:19 [llm_engine.py:437] init engine (profile, create
kv cache, warmup model) took 151.34 seconds

{"model_id":"48edd1930cbc4f51ad336602e2fa85e6","version_major":2,"vers
ion_minor":0}

Prompt: 'The future of AI is', Generated text: ' here. The future of
AI is now. The future of AI is here. The future of AI is now. The
future of AI is here. The future of AI is now. The future of AI is
here. The future of AI is now. The future of AI is here. The future of
AI is now. The future of AI is here. The future of AI is now. The
future of AI is here. The future of AI is now. The future of AI is
here. The future of AI is now. The future of AI is here. The future of
AI is now. The future of AI is here. The future of AI is now. The
future of AI is here. The future of AI is now. The future of AI is
here. The future of AI is now. The future of AI is here. The future of
AI is now. The future of AI is here. The future of AI is now. The
future of AI is here. The future of AI is now. The future of AI is
here. The future of AI is now. The future of AI is here. The future of
AI is now. The future of AI is here. The future of AI is now. The
future of AI is here. The future'
```

```python
from huggingface_hub import HfApi, login
import os

login(token=HF_TOKEN)  # Replace with your actual token

# Define the repository name on Hugging Face
repo_name = "Tharun013/gemma2-2b-finetuned-sql"

# Ensure the LoRA adapter directory exists
if not os.path.exists(sql_lora_path):
    raise ValueError(f"LoRA adapter path {sql_lora_path} does not
exist. Please save the adapter first.")

# Initialize the HfApi client
api = HfApi()

# Create the repository if it doesn't exist (optional)
api.create_repo(repo_id=repo_name, exist_ok=True)

# Upload the LoRA adapter directory to the Hugging Face Hub
api.upload_folder(
    folder_path=sql_lora_path,
    repo_id=repo_name,
    commit_message="Pushed gemma2-2b finetuned LoRA adapter - v0",
    repo_type="model"
)

print(f"Successfully pushed LoRA adapter to {repo_name}")
```

```
No files have been modified since last commit. Skipping to prevent
empty commit.
WARNING:huggingface_hub.hf_api:No files have been modified since last
commit. Skipping to prevent empty commit.

Successfully pushed LoRA adapter to Tharun013/gemma2-2b-finetuned-sql
```