```
!pip install -q -U transformers peft accelerate bitsandbytes trl
```

显示隐藏的输出项

```
import torch
import transformers
from transformers import (
        AutoModelForCausalLM,
        AutoTokenizer,
        BitsAndBytesConfig,
        TrainingArguments
)
from peft import LoraConfig
from trl import SFTTrainer,SFTConfig
from datasets import load_dataset
import huggingface_hub
```

```
2025-10-14 22:35:39.552411: E external/local_xla/xla/stream_executor/cuda/cuda_fft.cc:477] Unable to register cuFFT factory: Attempting to re
WARNING: All log messages before absl::InitializeLog() is called are written to STDERR
E0000 00:00:1760481339.911362      37 cuda_dnn.cc:8310] Unable to register cuDNN factory: Attempting to register factory for plugin cuDNN whe
E0000 00:00:1760481340.029342      37 cuda_blas.cc:1418] Unable to register cuBLAS factory: Attempting to register factory for plugin cuBLAS
/usr/local/lib/python3.11/dist-packages/pydantic/_internal/_generate_schema.py:2225: UnsupportedFieldAttributeWarning: The 'repr' attribute w
  warnings.warn(
/usr/local/lib/python3.11/dist-packages/pydantic/_internal/_generate_schema.py:2225: UnsupportedFieldAttributeWarning: The 'frozen' attribute
  warnings.warn(
```

```
from huggingface_hub import login, HfApi
from kaggle_secrets import UserSecretsClient

# 获取 Token
user_secrets = UserSecretsClient()
token = user_secrets.get_secret("huggingface")

# 登录
login(token)

# ✅ 新版方式查看当前账户信息
api = HfApi()
print(api.whoami())
```

```
{'type': 'user', 'id': '68def30d0f9c4d6971a915fc', 'name': 'pine-cone', 'fullname': 'Yanchao Wang', 'canPay': False, 'periodEnd': None, 'isPr
```

```
# QLoRA configuration
nf4_config = BitsAndBytesConfig(
    load_in_4bit=True,
    bnb_4bit_quant_type="nf4", # Use nf4 for weights initialized from a normal distribution
    bnb_4bit_use_double_quant=True, # Use a second quantization after the first one
    bnb_4bit_compute_dtype=torch.bfloat16 # Use bfloat16 for faster computation
)

# Load the model
model_id = "mistralai/Mistral-7B-v0.3"
model = AutoModelForCausalLM.from_pretrained(
        model_id,
        quantization_config=nf4_config,
        device_map="auto" # Automatically map layers to available devices (GPU/CPU)
)

# Load the tokenizer
tokenizer = AutoTokenizer.from_pretrained(model_id)
# A pad token is required for batching, but Mistral doesn't have one. We can use the EOS token.
tokenizer.pad_token = tokenizer.eos_token
```

```
config.json:   0%|          | 0.00/601 [00:00<?, ?B/s]
model.safetensors.index.json: 0.00B [00:00, ?B/s]
Fetching 3 files:   0%|          | 0/3 [00:00<?, ?it/s]
model-00003-of-00003.safetensors:   0%|          | 0.00/4.55G [00:00<?, ?B/s]
model-00002-of-00003.safetensors:   0%|          | 0.00/5.00G [00:00<?, ?B/s]
model-00001-of-00003.safetensors:   0%|          | 0.00/4.95G [00:00<?, ?B/s]
Loading checkpoint shards:   0%|          | 0/3 [00:00<?, ?it/s]
generation_config.json:   0%|          | 0.00/116 [00:00<?, ?B/s]
tokenizer_config.json: 0.00B [00:00, ?B/s]
tokenizer.model:   0%|          | 0.00/587k [00:00<?, ?B/s]
tokenizer.json: 0.00B [00:00, ?B/s]
special_tokens_map.json:   0%|          | 0.00/414 [00:00<?, ?B/s]
```

```
# Load the dataset
dataset = load_dataset("timdettmers/openassistant-guanaco", split="train")
```

```
# For faster training, you can select a subset
train_subset = dataset.select(range(1000))
eval_subset = dataset.select(range(1000, 1200))


# --- CHOOSE YOUR EXPERIMENT HERE ---
# Example for Experiment 1 (Baseline)
peft_config = LoraConfig(
    r=16,
    lora_alpha=32,
    lora_dropout=0.05,
    task_type="CAUSAL_LM",
)

sft_config = SFTConfig(
    output_dir="mistral-guanaco-baseline-2",
    push_to_hub=True, # Set to your desired repo name
    num_train_epochs=1, # One epoch is often sufficient for fine-tuning
    per_device_train_batch_size=4,
    learning_rate=2e-4,
    lr_scheduler_type="cosine",
    logging_steps=10,
    gradient_checkpointing=True,
    packing=True,
    dataset_text_field="text",
    max_length=512, # Use max_length as per lecture note
    eval_strategy="steps",
    eval_steps=50,
    report_to="none", # Set to "wandb" if you use Weights & Biases
)
```

```
Repo card metadata block was not found. Setting CardData to empty.
```

```
# Initialize the trainer
trainer = SFTTrainer(
    model=model,
    args=sft_config,
    train_dataset=train_subset,
    eval_dataset=eval_subset,
    peft_config=peft_config,
)

# Start training
trainer.train()
metrics = trainer.evaluate()
print(metrics)
trainer.push_to_hub()
```

```
Adding EOS to train dataset:   0%|          | 0/1000 [00:00<?, ? examples/s]
Tokenizing train dataset:   0%|          | 0/1000 [00:00<?, ? examples/s]
Packing train dataset:   0%|          | 0/1000 [00:00<?, ? examples/s]
Adding EOS to eval dataset:   0%|          | 0/200 [00:00<?, ? examples/s]
Tokenizing eval dataset:   0%|          | 0/200 [00:00<?, ? examples/s]
Packing eval dataset:   0%|          | 0/200 [00:00<?, ? examples/s]
/usr/local/lib/python3.11/dist-packages/torch/_dynamo/eval_frame.py:745: UserWarning: torch.utils.checkpoint: the use_reentrant parameter sho
  return fn(*args, **kwargs)
```

[164/164 2:39:28, Epoch 1/1]

Step	Training Loss	Validation Loss	Entropy	Num Tokens	Mean Token Accuracy
50	1.149900	1.017614	1.196431	101268.000000	0.717799
100	0.996600	1.011508	1.185752	202745.000000	0.717799
150	1.017900	1.007504	1.179635	304360.000000	0.719476

[17/17 08:06]

```
{'eval_loss': 1.0074366331100464, 'eval_runtime': 520.3571, 'eval_samples_per_second': 0.248, 'eval_steps_per_second': 0.033, 'eval_entropy':
No files have been modified since last commit. Skipping to prevent empty commit.
CommitInfo(commit_url='https://huggingface.co/pine-cone/mistral-guanaco-baseline-2/commit/56dd2e821d1736aee5e4d270f77d22b31d3d6fda',
commit_message='End of training', commit_description='', oid='56dd2e821d1736aee5e4d270f77d22b31d3d6fda', pr_url=None,
repo_url=RepoUrl('https://huggingface.co/pine-cone/mistral-guanaco-baseline-2', endpoint='https://huggingface.co', repo_type='model',
repo_id='pine-cone/mistral-guanaco-baseline-2'), pr_revision=None, pr_num=None)
```

```
from transformers import pipeline

torch.cuda.empty_cache()
# Load fine-tuned model from the Hub
pipe = pipeline("text-generation", model="pine-cone/mistral-guanaco-baseline-2")

# --- Prompt 1 ---
prompt1 = "What is the capital of Germany? Explain why that's the case and if it was different in the past?"
```

```
result1 = pipe(f"### Human: {prompt1} ### Assistant:", max_new_tokens=350)
print(result1[0]['generated_text'])

# --- Prompt 2 ---
prompt2 = "Write a Python function to calculate the factorial of a number."
result2 = pipe(f"### Human: {prompt2} ### Assistant:", max_new_tokens=150)
print(result2[0]['generated_text'])

# --- Prompt 3 ---
prompt3 = "A rectangular garden has a length of 25 feet and a width of 15 feet. If you want to build a fence around
result3 = pipe(f"### Human: {prompt3} ### Assistant:", max_new_tokens=150)
print(result3[0]['generated_text'])

# --- Prompt 4 ---
prompt4 = "What is the difference between a fruit and a vegetable? Give examples of each."
result4 = pipe(f"### Human: {prompt4} ### Assistant:", max_new_tokens=250)
print(result4[0]['generated_text'])
```

```
adapter_config.json:   0%|          | 0.00/860 [00:00<?, ?B/s]
config.json:   0%|          | 0.00/601 [00:00<?, ?B/s]
model.safetensors.index.json: 0.00B [00:00, ?B/s]
Fetching 3 files:   0%|          | 0/3 [00:00<?, ?it/s]
model-00002-of-00003.safetensors:   0%|          | 0.00/5.00G [00:00<?, ?B/s]
model-00003-of-00003.safetensors:   0%|          | 0.00/4.55G [00:00<?, ?B/s]
model-00001-of-00003.safetensors:   0%|          | 0.00/4.95G [00:00<?, ?B/s]
Loading checkpoint shards:   0%|          | 0/3 [00:00<?, ?it/s]
generation_config.json:   0%|          | 0.00/116 [00:00<?, ?B/s]
adapter_model.safetensors:   0%|          | 0.00/27.3M [00:00<?, ?B/s]
tokenizer_config.json: 0.00B [00:00, ?B/s]
tokenizer.model:   0%|          | 0.00/587k [00:00<?, ?B/s]
tokenizer.json: 0.00B [00:00, ?B/s]
special_tokens_map.json:   0%|          | 0.00/414 [00:00<?, ?B/s]
Device set to use cuda:0
Setting `pad_token_id` to `eos_token_id`:2 for open-end generation.
Setting `pad_token_id` to `eos_token_id`:2 for open-end generation.
### Human: What is the capital of Germany? Explain why that's the case and if it was different in the past? ### Assistant: The capital of Ger

In the past, Berlin was not the capital of Germany. The first German capital was Frankfurt, which served as the capital from 1871 to 1945. Du

In conclusion, the capital of Germany is Berlin because it is the largest city in the country and serves as a center for politics, culture, a
Setting `pad_token_id` to `eos_token_id`:2 for open-end generation.
### Human: Write a Python function to calculate the factorial of a number. ### Assistant: Here's a Python function to calculate the factorial

def factorial(n):
    if n == 0:
        return 1
    else:
        return n * factorial(n-1)

For example, to calculate the factorial of 5 (5!), you would call the function like this:

print(factorial(5))

This will print the result 120.
Setting `pad_token_id` to `eos_token_id`:2 for open-end generation.
### Human: A rectangular garden has a length of 25 feet and a width of 15 feet. If you want to build a fence around the entire garden, how ma
### Human: What is the difference between a fruit and a vegetable? Give examples of each. ### Assistant: A fruit is a fleshy part of a flower

1. Fruit: a fleshy part of a plant that contains seeds and is sweet or sour in taste. Examples include apples, oranges, bananas, strawberries

2. Vegetable: an edible part of a plant that is not a fruit or a seed. Examples include carrots, potatoes, broccoli, spinach, and cauliflower

It's important to note that some fruits are not considered vegetables, and vice versa. For example, bell peppers and cucumbers are technicall
```