Double-click (or enter) to edit

```
%%capture
# Installs Unsloth, Xformers (Flash Attention) and all other packages
!pip install "unsloth[colab-new] @ git+https://github.com/unslothai/unsloth.git"
!pip install --no-deps "xformers<0.0.26" trl peft accelerate bitsandbytes
```

Double-click (or enter) to edit

```
from unsloth import FastLanguageModel
import torch
max_seq_length = 2048
dtype = None
load_in_4bit = True

model, tokenizer = FastLanguageModel.from_pretrained(
    model_name = "unsloth/llama-3-8b-bnb-4bit",
    max_seq_length = max_seq_length,
    dtype = dtype,
    load_in_4bit = load_in_4bit,
)
```

```
/usr/local/lib/python3.10/dist-packages/huggingface_hub/file_download.py:1132: FutureWarning: `resume_download` is deprecated and will b
  warnings.warn(
config.json: 100%                                  1.12k/1.12k [00:00<00:00, 68.7kB/s]
==((====))==  Unsloth: Fast Llama patching release 2024.4
   \\   /|    GPU: NVIDIA A100-SXM4-40GB. Max memory: 39.564 GB. Platform = Linux.
O^O/ \_/ \    Pytorch: 2.2.1+cu121. CUDA = 8.0. CUDA Toolkit = 12.1.
\        /    Bfloat16 = TRUE. Xformers = 0.0.25.post1. FA = False.
 "-____-"     Free Apache license: http://github.com/unslothai/unsloth
Unused kwargs: ['quant_method']. These kwargs are not used in <class 'transformers.utils.quantization_config.BitsAndBytesConfig'>.
model.safetensors: 100%                                  5.70G/5.70G [00:27<00:00, 189MB/s]
generation_config.json: 100%                                  143/143 [00:00<00:00, 11.0kB/s]
tokenizer_config.json: 100%                                  50.6k/50.6k [00:00<00:00, 2.50MB/s]
tokenizer.json: 100%                                  9.09M/9.09M [00:00<00:00, 25.1MB/s]
special_tokens_map.json: 100%                                  464/464 [00:00<00:00, 42.1kB/s]
Special tokens have been added in the vocabulary, make sure the associated word embeddings are fine-tuned or trained.
Special tokens have been added in the vocabulary, make sure the associated word embeddings are fine-tuned or trained.
```

Show current memory stats

Show code

```
GPU = NVIDIA A100-SXM4-40GB. Max memory = 39.564 GB.
8.305 GB of memory reserved.
```

We now add LoRA adapters so we only need to update 1 to 10% of all parameters!

```
model = FastLanguageModel.get_peft_model(
    model,
    r = 64,
    target_modules = ["q_proj", "k_proj", "v_proj", "o_proj",
                      "gate_proj", "up_proj", "down_proj",],
    lora_alpha = 16,
    lora_dropout = 0,
    bias = "none",
    use_gradient_checkpointing = True ,
    random_state = 3407,
    use_rslora = False,
)
```

```
Unsloth 2024.4 patched 32 layers with 32 QKV layers, 32 O layers and 32 MLP layers.
```

Data Prep

```
alpaca_prompt = """Below is Questions, paired with an context that provides further context. Write a response that appropriately completes t

### question:
{}

### context:
{}

### final_decision:
{}

### long_answer:
{}"""

EOS_TOKEN = tokenizer.eos_token
def formatting_prompts_func(examples):
    question = examples["question"]
    context  = examples["context"]
    final_decision = examples["final_decision"]
    long_answer = examples["long_answer"]
    texts = []
    for question, context, final_decision, long_answer in zip(question, context, final_decision, long_answer):
        text = alpaca_prompt.format(question, context, final_decision, long_answer) + EOS_TOKEN
        texts.append(text)
    return { "text" : texts, }
pass

from datasets import load_dataset
from datasets import DatasetDict

dataset = load_dataset("qiaojin/PubMedQA", "pqa_artificial")

# Assuming you want a standard 80/20 train/test split


# Access and split the 'train' dataset
train_test_dataset = dataset['train'].train_test_split(test_size=0.002)
dataset['train'] = train_test_dataset['train']
dataset['test'] = train_test_dataset['test']

# Now you can access the splits:
train_dataset = dataset['train']
test_dataset = dataset['test']


dataset = dataset.map(formatting_prompts_func, batched = True,)
```

```
Downloading readme: 100%                                  5.19k/5.19k [00:00<00:00, 416kB/s]
Downloading data: 100%                                    233M/233M [00:01<00:00, 198MB/s]
Generating train split: 100%              211269/211269 [00:01<00:00, 113937.69 examples/s]
Map: 100%                                 210846/210846 [00:20<00:00, 10270.13 examples/s]
Map: 100%                                          423/423 [00:00<00:00, 6910.08 examples/s]
```

Train the model

```
from trl import SFTTrainer
from transformers import TrainingArguments

trainer = SFTTrainer(
    model = model,
    tokenizer = tokenizer,
    train_dataset=dataset['train'],
    eval_dataset=dataset['test'],
    dataset_text_field = "text",
    max_seq_length = max_seq_length,
    dataset_num_proc = 2,
    args = TrainingArguments(
        per_device_train_batch_size = 2,
        gradient_accumulation_steps = 4,
        warmup_steps = 5,
        num_train_epochs=1,
        max_steps = 1000,
        learning_rate = 2e-4,
        fp16 = not torch.cuda.is_bf16_supported(),
        bf16 = torch.cuda.is_bf16_supported(),
        logging_steps = 1,
        optim = "adamw_8bit",
        weight_decay = 0.01,
        lr_scheduler_type = "linear",
        seed = 3407,
        output_dir = "outputs",
    ),
)
```

```
/usr/local/lib/python3.10/dist-packages/multiprocess/popen_fork.py:66: RuntimeWarning: c
  self.pid = os.fork()
Map (num_proc=2): 100%                    210846/210846 [02:41<00:00, 855.15 examples/s]
Map (num_proc=2): 100%                          423/423 [00:01<00:00, 371.52 examples/s]
max_steps is given, it will override any value given in num_train_epochs
```

Double-click (or enter) to edit

```
trainer_stats = trainer.train()
```

92	1.183400
93	1.199600
94	1.084300
95	1.240400
96	1.228000
97	1.257700
98	1.087000
99	1.235900
100	1.254700
101	1.129500
102	1.227100
103	1.367400
104	1.237700
105	1.184500
106	1.055700
107	1.212900
108	1.212600
109	1.410100
110	1.057500
111	1.097000
112	1.101600
113	1.132000
114	1.003400
115	1.351800
116	1.176400
117	1.259900
118	1.267300
119	1.205800
120	1.324700
121	1.144200
122	1.198100
123	1.137300