Installing packages

```
%%capture
!pip install unsloth
# Also get the latest nightly Unsloth!
!pip uninstall unsloth -y && pip install --upgrade --no-cache-dir --
no-deps git+https://github.com/unslothai/unsloth.git
```

In this step, we are initializing our Qwen2.5-7B-Instruct model using Unsloth's FastLanguageModel.

```
from unsloth import FastLanguageModel
import torch
max_seq_length = 2048 # Choose any! We auto support RoPE Scaling
internally!
dtype = None # None for auto detection. Float16 for Tesla T4, V100,
Bfloat16 for Ampere+
load_in_4bit = True # Use 4bit quantization to reduce memory usage.
Can be False.

model, tokenizer = FastLanguageModel.from_pretrained(
    model_name = "unsloth/Qwen2.5-7B-Instruct",
    max_seq_length = max_seq_length,
    dtype = dtype,
    load_in_4bit = load_in_4bit,
)
```

```
🦥 Unsloth: Will patch your computer to enable 2x faster free
finetuning.
🦥 Unsloth Zoo will now patch everything to make training faster!
==((====))==  Unsloth 2025.3.9: Fast Qwen2 patching. Transformers:
4.48.3.
   \\   /|    Tesla T4. Num GPUs = 1. Max memory: 14.741 GB. Platform:
Linux.
O^O/ \_/ \    Torch: 2.6.0+cu124. CUDA: 7.5. CUDA Toolkit: 12.4.
Triton: 3.2.0
\        /    Bfloat16 = FALSE. FA [Xformers = 0.0.29.post3. FA2 =
False]
 "-____-"     Free license: http://github.com/unslothai/unsloth
Unsloth: Fast downloading is enabled - ignore downloading bars which
are red colored!
```


```
{"model_id":"5577c77e3ec74d8989b280b2e9a0e3ea","version_major":2,"vers
ion_minor":0}
```

```
{"model_id":"7cd4d3351b6a4a3884e122acf2ecd761","version_major":2,"vers
ion_minor":0}
```

{"model_id":"ab4a8f29a7444190ade76ac9c09b4799","version_major":2,"version_minor":0}

{"model_id":"bd037123761645dca2fe175c7ef192f0","version_major":2,"version_minor":0}

{"model_id":"35eaad6ff5624f62805b35e40b77e654","version_major":2,"version_minor":0}

{"model_id":"541e39da255047f0be79507e27c35573","version_major":2,"version_minor":0}

{"model_id":"8285060135854d1eab33a7dbf90a48c3","version_major":2,"version_minor":0}

{"model_id":"3c0b8e5fbddd4880a98da56d4bc0dcc0","version_major":2,"version_minor":0}

{"model_id":"be29c5adcca94315b528f99b7d61fcbe","version_major":2,"version_minor":0}

{"model_id":"ce80a3a309a44c488d5bb38e30913c7c","version_major":2,"version_minor":0}

{"model_id":"7eab5619a1de4aecbfa4d7267606d4cf","version_major":2,"version_minor":0}

{"model_id":"b50ab5312f6243ee8f58b7e33b676821","version_major":2,"version_minor":0}

In this step, we are applying Parameter-Efficient Fine-Tuning (PEFT) using LoRA (Low-Rank Adaptation) to our model. Instead of modifying all the model parameters, LoRA injects trainable low-rank matrices into specific layers, making fine-tuning much more memory-efficient.

```
model = FastLanguageModel.get_peft_model(
    model,
    r = 16,
    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 = "unsloth",
    random_state = 3407,
    use_rslora = False,
    loftq_config = None,
)
```

```
Unsloth 2025.3.9 patched 28 layers with 28 QKV layers, 28 O layers and 28 MLP layers.
```

In this step, we are loading a CSV file containing user interactions and converting it into a Hugging Face Dataset format for fine-tuning.

This transforms the formatted data into a Hugging Face Dataset, making it efficient for fine-tuning large models.

```
import pandas as pd

df = pd.read_csv("/content/merged_df.csv")

from datasets import Dataset

data_examples = df.apply(lambda row: {
    "conversations": [
        {"from": "human", "value": row["user_input"]},
        {"from": "gpt", "value": row["reference"]}
    ]
}, axis=1).tolist()

dataset = Dataset.from_list(data_examples)

dataset
```

```
Dataset({
    features: ['conversations'],
    num_rows: 548
})
```

This step formats the dataset using the Qwen-2.5 chat template, ensuring that conversations follow the expected structure for the model.

```
from unsloth.chat_templates import get_chat_template

tokenizer = get_chat_template(
    tokenizer,
    chat_template = "qwen-2.5",
)

def formatting_prompts_func(examples):
    convos = examples["conversations"]
    texts = [tokenizer.apply_chat_template(convo, tokenize = False,
add_generation_prompt = False) for convo in convos]
    return { "text" : texts, }
pass
```

Instead of raw conversations, the dataset now contains properly formatted prompts, improving the model's ability to learn conversational nuances.

```
dataset[5]["conversations"]
```

```
[{'from': 'human',
  'value': 'What advancements does DeepSeek-R1-Zero bring to reasoning
capabilities in language models?'},
 {'from': 'gpt',
  'value': 'DeepSeek-R1-Zero exhibits super performance on reasoning
benchmarks, with a pass@1 score on AIME 2024 increasing from 15.6% to
71.0%, and further improving to 86.7% with majority voting, matching
the performance of OpenAI-o1-0912. However, it also faces challenges
such as poor readability and language mixing.'}]
```

his step standardizes and formats the dataset for training using ShareGPT-style conversation formatting and applies the previously defined chat template.

```
from unsloth.chat_templates import standardize_sharegpt
dataset = standardize_sharegpt(dataset)
dataset = dataset.map(formatting_prompts_func, batched = True,)
```


```
{"model_id":"cc0a6644a12549549c66498ed84cf247","version_major":2,"vers
ion_minor":0}
```

```
{"model_id":"45d1cf4e801548e68ec54e9ba69d0854","version_major":2,"vers
ion_minor":0}
```


```
dataset[5]["conversations"]
```

```
[{'content': 'What advancements does DeepSeek-R1-Zero bring to
reasoning capabilities in language models?',
  'role': 'user'},
 {'content': 'DeepSeek-R1-Zero exhibits super performance on reasoning
benchmarks, with a pass@1 score on AIME 2024 increasing from 15.6% to
71.0%, and further improving to 86.7% with majority voting, matching
the performance of OpenAI-o1-0912. However, it also faces challenges
such as poor readability and language mixing.',
  'role': 'assistant'}]
```

```
dataset
```

```
Dataset({
    features: ['conversations', 'text'],
    num_rows: 548
})
```

Hyperparameter in training

This step splits the dataset into training and validation sets and initializes the Supervised Fine-Tuning (SFT) Trainer for optimizing the model.

```
from datasets import Dataset
from trl import SFTTrainer
from transformers import TrainingArguments, DataCollatorForSeq2Seq
from unsloth import is_bfloat16_supported

dataset_split = dataset.train_test_split(test_size=0.2, seed=42)
train_dataset = dataset_split["train"]
val_dataset = dataset_split["test"]

trainer = SFTTrainer(
    model=model,
    tokenizer=tokenizer,
    train_dataset=train_dataset,
    eval_dataset=val_dataset,
    dataset_text_field="text",
    max_seq_length=max_seq_length,
    data_collator=DataCollatorForSeq2Seq(tokenizer=tokenizer),
    dataset_num_proc=8,
    packing=False,
    args=TrainingArguments(
        per_device_train_batch_size=2,
        gradient_accumulation_steps=4,
        warmup_steps=5,
        num_train_epochs=4,
        learning_rate=1e-4,
        fp16=not is_bfloat16_supported(),
        bf16=is_bfloat16_supported(),
        logging_steps=1,
        optim="adamw_8bit",
        weight_decay=0.01,
        lr_scheduler_type="cosine",
        seed=3407,
        output_dir="outputs",
        report_to="none",
        evaluation_strategy="epoch",
    ),
)
```

```
/usr/local/lib/python3.11/dist-packages/transformers/
training_args.py:1575: FutureWarning: `evaluation_strategy` is
deprecated and will be removed in version 4.46 of  Transformers. Use
`eval_strategy` instead
  warnings.warn(
/usr/local/lib/python3.11/dist-packages/transformers/training_args.py:
1575: FutureWarning: `evaluation_strategy` is deprecated and will be
removed in version 4.46 of  Transformers. Use `eval_strategy` instead
  warnings.warn(
```


```
{"model_id":"0d061ec4eb1d44c39d25e0a34339a271","version_major":2,"vers
ion_minor":0}
```

{"model_id":"8dd55329105f493cbc32baa9f141c59a","version_major":2,"version_minor":0}

```
train_dataset
```

```
Dataset({
    features: ['conversations', 'text'],
    num_rows: 438
})
```

We also use Unsloth's train_on_completions method to only train on the assistant outputs and ignore the loss on the user's inputs.

```
from unsloth.chat_templates import train_on_responses_only
trainer = train_on_responses_only(
    trainer,
    instruction_part = "<|im_start|>user\n",
    response_part = "<|im_start|>assistant\n",
)
```

{"model_id":"25e3803abbbd4acbb60d923d36a62cba","version_major":2,"version_minor":0}

{"model_id":"9b63825fffd54061a02e2a2bcc5a8eba","version_major":2,"version_minor":0}

Verify masking is actually done:

```
tokenizer.decode(trainer.train_dataset[5]["input_ids"])
```

{"type":"string"}

```
space = tokenizer(" ", add_special_tokens = False).input_ids[0]
tokenizer.decode([space if x == -100 else x for x in
trainer.train_dataset[5]["labels"]])
```

{"type":"string"}

We can see the System and Instruction prompts are successfully masked!

```
#@title Show current memory stats
gpu_stats = torch.cuda.get_device_properties(0)
start_gpu_memory = round(torch.cuda.max_memory_reserved() / 1024 /
1024 / 1024, 3)
max_memory = round(gpu_stats.total_memory / 1024 / 1024 / 1024, 3)
print(f"GPU = {gpu_stats.name}. Max memory = {max_memory} GB.")
print(f"{start_gpu_memory} GB of memory reserved.")
```

```
GPU = Tesla T4. Max memory = 14.741 GB.
8.76 GB of memory reserved.
```

Inference before fine tuning

```
df['user_input'][530]
```

```
{"type":"string"}
```

```
df['reference'][530]
```

```
{"type":"string"}
```

We give the above Question to the model below and as we can see from the output it is not the correct as we have in Answers set

```
FastLanguageModel.for_inference(model) # Enable native 2x faster inference

messages = [
    {"role": "user", "content": "Can you elaborate on the features and significance of FlashMLA in the context of open-source AI development?"},
]
inputs = tokenizer.apply_chat_template(
    messages,
    tokenize = True,
    add_generation_prompt = True, # Must add for generation
    return_tensors = "pt",
).to("cuda")

from transformers import TextStreamer
text_streamer = TextStreamer(tokenizer, skip_prompt = True)
_ = model.generate(input_ids = inputs, streamer = text_streamer, max_new_tokens = 2048,
                   use_cache = True, temperature = 1.5, min_p = 0.1)
```

```
The attention mask is not set and cannot be inferred from input because pad token is same as eos token. As a consequence, you may observe unexpected behavior. Please pass your input's `attention_mask` to obtain reliable results.

FlashMLA, which stands for Fast and Scalable Machine Learning Application, is an open-source framework that facilitates rapid and efficient machine learning workflows for data scientists and engineers. It is designed to bridge the gap between complex machine learning tasks and practical, scalable implementations. Here are some key features and aspects of significance in the context of open-source AI development:

1. **Integration with Open-Source Ecosystem**: FlashMLA leverages popular Python libraries like TensorFlow, PyTorch, and Keras for building machine learning models. This integration allows developers
```

to work within their preferred environment and seamlessly transition between different technologies based on the project requirements.

2. **Automatic Tuning and Hyperparameter Optimization**: One of the significant advantages of FlashMLA is its capability to automatically tune hyperparameters using Bayesian optimization or evolutionary algorithms. This automated tuning can significantly enhance model performance without the need for manual trial and error.

3. **Parallel Processing Support**: The framework is optimized for parallel processing, which can drastically reduce the training time for complex models. By supporting distributed training across multiple machines, it addresses one of the major challenges in scaling machine learning pipelines.

4. **Unified Interface**: FlashMLA provides a unified interface to handle both supervised and unsupervised learning tasks, as well as deep learning applications. This consistency in API design makes it easier for users to adopt and use the tool without extensive re-learning.

5. **Community and Contribution**: Being open-source, FlashMLA benefits from community contributions. Developers can contribute code, share improvements, and report issues. This collaborative model accelerates innovation and ensures that best practices are integrated into the framework.

6. **Performance and Scalability**: FlashMLA is designed with scalability in mind, which is crucial for handling large datasets and real-time applications. Its architecture allows for efficient memory management and optimized computations, ensuring that it performs well even with growing dataset sizes.

7. **Support for Real-Time Applications**: The framework supports real-time prediction capabilities, making it suitable for applications where quick decision-making based on new data points is critical. This feature is particularly valuable in industries such as finance, healthcare, and IoT.

8. **Documentation and Resources**: Open-source projects like FlashMLA typically come with extensive documentation, tutorials, and resources. These resources help newcomers understand the framework and get started quickly.

9. **Customizability**: FlashMLA is customizable to meet the specific needs of different projects. Users can extend its functionality by writing custom modules or integrating additional algorithms, enhancing its versatility.

10. **Community Collaboration**: By being part of the open-source

```
community, FlashMLA can benefit from collaborative research and
development. Innovations in other projects often lead to improvements
in FlashMLA, creating a virtuous cycle of advancement.

In the broader context of open-source AI development, frameworks like
FlashMLA play a critical role by providing robust, user-friendly tools
that democratize access to powerful machine learning technologies.
They not only help individual developers but also accelerate
innovation in AI by fostering collaboration and sharing of knowledge
among the global developer community.<|im_end|>
```

```
df['user_input'][123]
```

```
{"type":"string"}
```

```
df['reference'][123]
```

```
{"type":"string"}
```

Here also the answer generated from the model is not correct

```
FastLanguageModel.for_inference(model) # Enable native 2x faster
inference

messages = [
    {"role": "user", "content": "Who developed the DualPipe
algorithm?"},
]
inputs = tokenizer.apply_chat_template(
    messages,
    tokenize = True,
    add_generation_prompt = True, # Must add for generation
    return_tensors = "pt",
).to("cuda")

from transformers import TextStreamer
text_streamer = TextStreamer(tokenizer, skip_prompt = True)
_ = model.generate(input_ids = inputs, streamer = text_streamer,
max_new_tokens = 2048,
                   use_cache = True, temperature = 1.5, min_p = 0.1)

The DualPipe algorithm was developed by researchers from Google AI and
University of California, Berkeley. Specifically, the team involved in
this development included individuals like Chris Albert, Kaimi Yao,
Yujia (Jerry) Li, and Pieter Abbeel. The algorithm was designed to
improve reinforcement learning in environments with delayed rewards,
which is a common challenge in training agents for tasks like robot
control or game playing. It was introduced in a research paper
published in 2020 and has been influential in the field of
reinforcement learning.<|im_end|>
```

Let's start training

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

```
==((====))==  Unsloth - 2x faster free finetuning | Num GPUs used = 1
   \\   /|    Num examples = 438 | Num Epochs = 4 | Total steps = 216
O^O/ \_/ \    Batch size per device = 2 | Gradient accumulation steps
= 4
\        /    Data Parallel GPUs = 1 | Total batch size (2 x 4 x 1) =
8
 "-____-"     Trainable parameters = 40,370,176/4,931,917,312 (0.82%
trained)

<IPython.core.display.HTML object>

Unsloth: Not an error, but Qwen2ForCausalLM does not accept
`num_items_in_batch`.
Using gradient accumulation will be very slightly less accurate.
Read more on gradient accumulation issues here:
https://unsloth.ai/blog/gradient
```

Let's take the training logs in to a dataset to analyze

```
logs = pd.DataFrame(trainer.state.log_history)
logs
```


```
{"summary":"{\n  \"name\": \"logs\",\n  \"rows\": 221,\n  \"fields\":
[\n    {\n      \"column\": \"loss\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 0.1518706545632991,\n
\"min\": 0.015,\n        \"max\": 0.6504,\n
\"num_unique_values\": 211,\n        \"samples\": [\n
0.4659,\n          0.0212,\n          0.0515\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"grad_norm\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
0.6063032070241274,\n        \"min\": 0.61030113697052,\n
\"max\": 3.568993330001831,\n        \"num_unique_values\": 216,\n
\"samples\": [\n          1.0988860130310059,\n
1.2493927478790283,\n          1.4881526231765747\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"learning_rate\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
3.532243506051998e-05,\n        \"min\": 0.0,\n        \"max\":
0.0001,\n        \"num_unique_values\": 216,\n        \"samples\": [\n
1.0823253562649793e-06,\n          2.2166786708976983e-08,\n
2.9414105126205497e-05\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"epoch\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 1.1624975286719443,\n        \"min\": 0.0182648401826484,\n
\"max\": 3.9863013698630136,\n        \"num_unique_values\": 216,\n
```

```
\"samples\": [\n          3.730593607305936,\n
3.949771689497717,\n          2.5662100456621006\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"step\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 62,\n        \"min\": 1,\n
\"max\": 216,\n        \"num_unique_values\": 216,\n
\"samples\": [\n          202,\n          214,\n          139\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"eval_loss\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
0.06509710070334868,\n        \"min\": 0.4827331602573395,\n
\"max\": 0.6288233399391174,\n        \"num_unique_values\": 4,\n
\"samples\": [\n          0.5109438896179199,\n
0.6288233399391174,\n          0.4827331602573395\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"eval_runtime\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
0.12141620910460617,\n        \"min\": 29.6331,\n        \"max\":
29.9009,\n        \"num_unique_values\": 4,\n        \"samples\": [\n
29.6331,\n          29.9009,\n          29.6723\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"eval_samples_per_second\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
0.014899664425751448,\n        \"min\": 3.679,\n        \"max\":
3.712,\n        \"num_unique_values\": 4,\n        \"samples\": [\n
3.712,\n          3.679,\n          3.707\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"eval_steps_per_second\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
0.007767453465154079,\n        \"min\": 1.839,\n        \"max\":
1.856,\n        \"num_unique_values\": 4,\n        \"samples\": [\n
1.856,\n          1.839,\n          1.854\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"train_runtime\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
null,\n        \"min\": 1469.5101,\n        \"max\": 1469.5101,\n
\"num_unique_values\": 1,\n        \"samples\": [\n
1469.5101\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"train_samples_per_second\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": null,\n        \"min\":
1.192,\n        \"max\": 1.192,\n        \"num_unique_values\": 1,\n
\"samples\": [\n          1.192\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"train_steps_per_second\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
null,\n        \"min\": 0.147,\n        \"max\": 0.147,\n
\"num_unique_values\": 1,\n        \"samples\": [\n          0.147\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
```

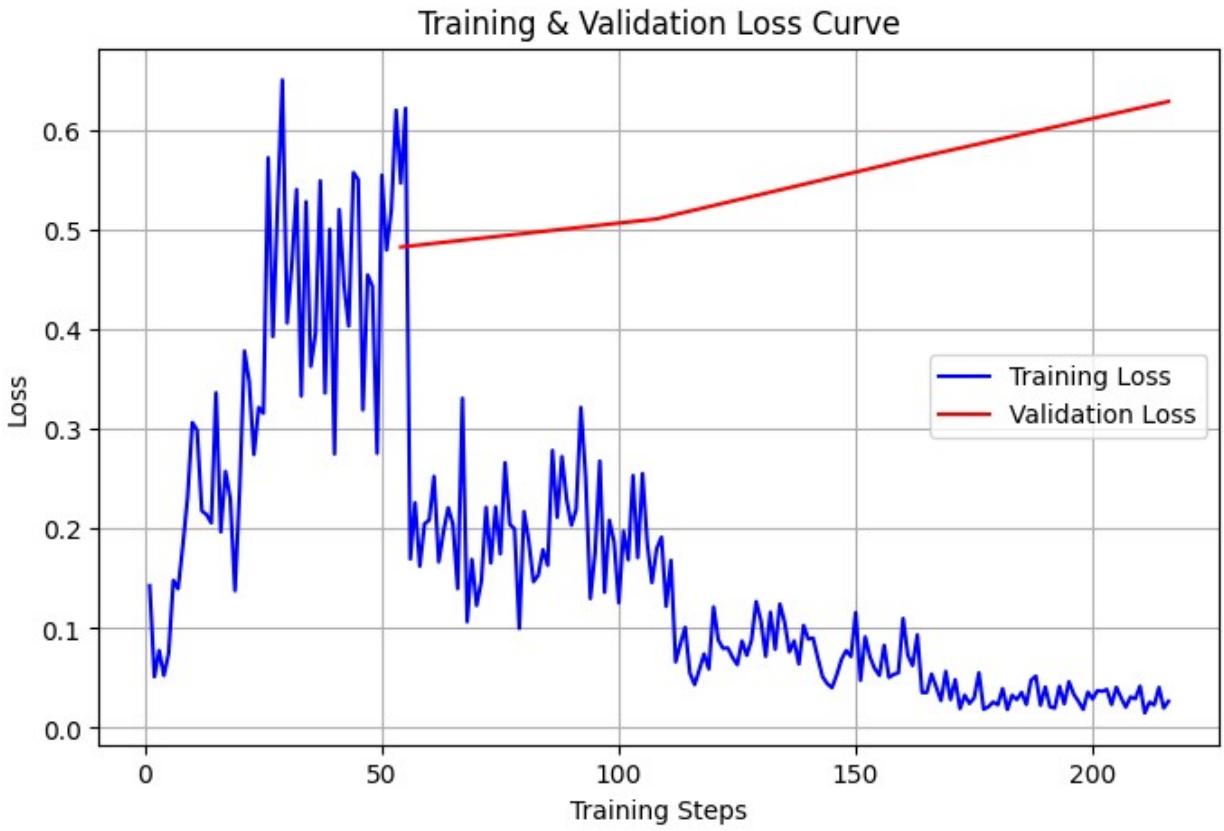
```
}\n    },\n    {\n      \"column\": \"total_flos\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
null,\n        \"min\": 1.503248918578176e+16,\n        \"max\":
1.503248918578176e+16,\n        \"num_unique_values\": 1,\n
\"samples\": [\n          1.503248918578176e+16\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"train_loss\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
null,\n        \"min\": 0.16739999026025612,\n        \"max\":
0.16739999026025612,\n        \"num_unique_values\": 1,\n
\"samples\": [\n          0.16739999026025612\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    }\n  ]\n}","type":"dataframe","variable_name":"logs"}
```


```
import matplotlib.pyplot as plt

train_loss = logs[logs["loss"].notna()]
eval_loss = logs[logs["eval_loss"].notna()]

plt.figure(figsize=(8, 5))
plt.plot(train_loss["step"], train_loss["loss"], label="Training
Loss", color="blue")
plt.plot(eval_loss["step"], eval_loss["eval_loss"], label="Validation
Loss", color="red")

plt.xlabel("Training Steps")
plt.ylabel("Loss")
plt.title("Training & Validation Loss Curve")
plt.legend()
plt.grid()
plt.show()
```



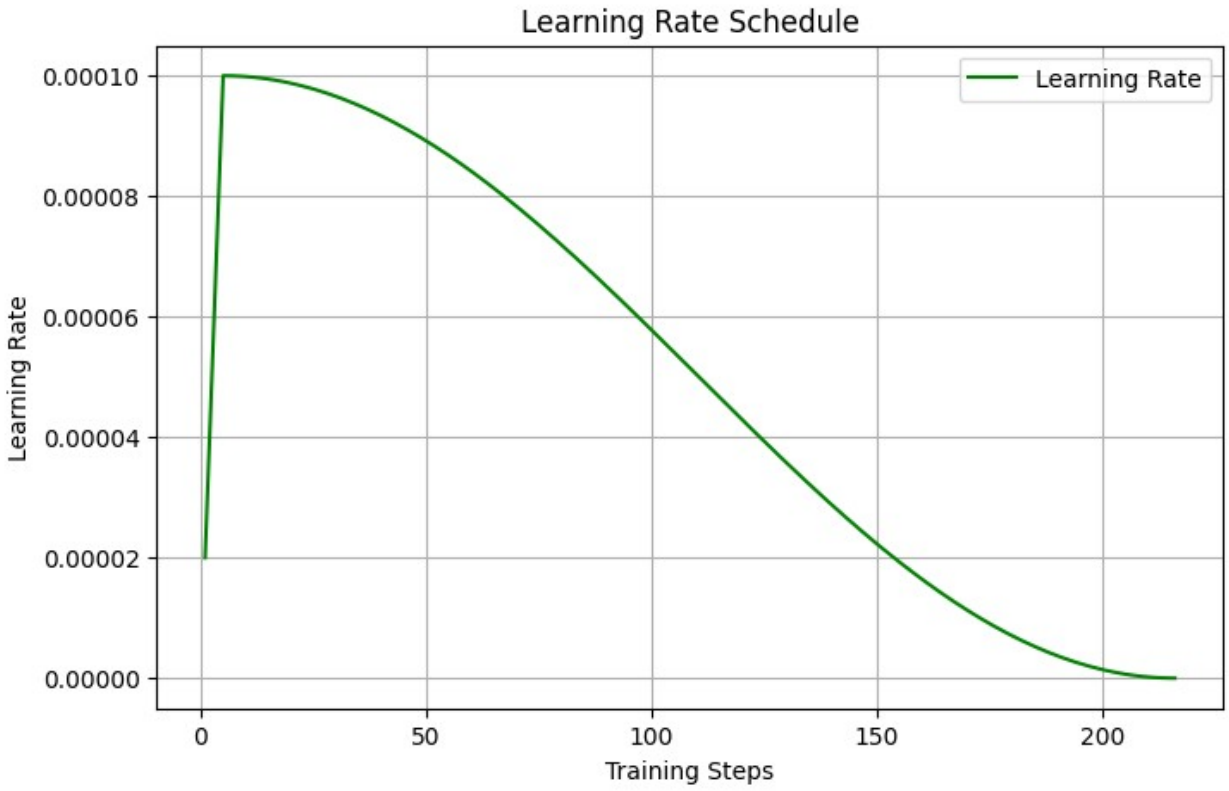
Here training loss is decreasing, but validation loss is increasing, which could suggest slight overfitting.

The presence of spikes in the training loss early on suggests that the optimizer is making large updates, which later stabilize as the learning rate decreases.

```
lr_logs = logs[logs["learning_rate"].notna()]

plt.figure(figsize=(8, 5))
plt.plot(lr_logs["step"], lr_logs["learning_rate"], label="Learning
Rate", color="green")

plt.xlabel("Training Steps")
plt.ylabel("Learning Rate")
plt.title("Learning Rate Schedule")
plt.legend()
plt.grid()
plt.show()
```



Further improvements could involve early stopping or regularization techniques (e.g., dropout, weight decay, data augmentation) to mitigate overfitting.

For time concerns we are going forward with this now

```
#@title Show final memory and time stats
used_memory = round(torch.cuda.max_memory_reserved() / 1024 / 1024 /
1024, 3)
used_memory_for_lora = round(used_memory - start_gpu_memory, 3)
used_percentage = round(used_memory         /max_memory*100, 3)
lora_percentage = round(used_memory_for_lora/max_memory*100, 3)
print(f"{trainer_stats.metrics['train_runtime']} seconds used for
training.")
print(f"{round(trainer_stats.metrics['train_runtime']/60, 2)} minutes
used for training.")
print(f"Peak reserved memory = {used_memory} GB.")
print(f"Peak reserved memory for training = {used_memory_for_lora}
GB.")
print(f"Peak reserved memory % of max memory = {used_percentage} %.")
print(f"Peak reserved memory for training % of max memory =
{lora_percentage} %.")
```

```
1469.5101 seconds used for training.
24.49 minutes used for training.
Peak reserved memory = 8.832 GB.
Peak reserved memory for training = 0.072 GB.
```

```
Peak reserved memory % of max memory = 59.915 %.
Peak reserved memory for training % of max memory = 0.488 %.
```

Inference after fine tuning

```
from unsloth.chat_templates import get_chat_template

tokenizer = get_chat_template(
    tokenizer,
    chat_template = "qwen-2.5",
)

FastLanguageModel.for_inference(model) # Enable native 2x faster
inference

messages = [
    {"role": "user", "content": "Can you elaborate on the features and
significance of FlashMLA in the context of open-source AI
development?"},
]
inputs = tokenizer.apply_chat_template(
    messages,
    tokenize = True,
    add_generation_prompt = True, # Must add for generation
    return_tensors = "pt",
).to("cuda")

from transformers import TextStreamer
text_streamer = TextStreamer(tokenizer, skip_prompt = True)
_ = model.generate(input_ids = inputs, streamer = text_streamer,
max_new_tokens = 2048,
                   use_cache = True, temperature = 1.5, min_p = 0.1)

FlashMLA is an efficient MLA decoding kernel specifically optimized
for Hopper GPUs, designed to handle variable-length sequences and has
been battle-tested in production. It boasts several impressive
features, including BF16 support, a paged KV cache with a block size
of 64, and impressive performance metrics, achieving 3000 GB/s memory-
bound and 580 TFLOPS compute-bound on H800 GPUs. The introduction of
FlashMLA as part of the open-source initiative reflects a commitment
to transparency and community-driven innovation, allowing developers
to share their progress and contribute to the collective momentum in
AI exploration.<|im_end|>

FastLanguageModel.for_inference(model) # Enable native 2x faster
inference

messages = [
    {"role": "user", "content": "Who developed the DualPipe
```

```
algorithm?"},
]
inputs = tokenizer.apply_chat_template(
    messages,
    tokenize = True,
    add_generation_prompt = True, # Must add for generation
    return_tensors = "pt",
).to("cuda")

from transformers import TextStreamer
text_streamer = TextStreamer(tokenizer, skip_prompt = True)
_ = model.generate(input_ids = inputs, streamer = text_streamer,
max_new_tokens = 2048,
                   use_cache = True, temperature = 1.5, min_p = 0.1)
```

```
DualPipe was created and developed by Jiashi Li and Chengqi Deng and
Wenfeng Liang.<|im_end|>
```

Now we can see that the model correctly answers the same question we asked before fine-tuning. This implies that our training worked well in some contexts. Let's evaluate it further on unseen data in the evaluation phase.

Let's save the model on hugging face

```
from huggingface_hub import login
login(token="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")

model.push_to_hub_gguf(
    "AkinduH/Qwen2.5-3B-Instruct-Fine-Tuned-on-Deepseek-Research-
Papers",
    tokenizer,
    quantization_method = "q4_k_m",
    token="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"
)
```

```
Unsloth: Merging 4bit and LoRA weights to 16bit...
Unsloth: Will use up to 5.31 out of 12.67 RAM for saving.
Unsloth: Saving model... This might take 5 minutes ...

100%|██████████| 28/28 [04:40<00:00, 10.01s/it]

Unsloth: Saving tokenizer... Done.
Unsloth: Saving AkinduH/Qwen2.5-3B-Instruct-Fine-Tuned-on-Deepseek-
Research-Papers/pytorch_model-00001-of-00004.bin...
Unsloth: Saving AkinduH/Qwen2.5-3B-Instruct-Fine-Tuned-on-Deepseek-
Research-Papers/pytorch_model-00002-of-00004.bin...
Unsloth: Saving AkinduH/Qwen2.5-3B-Instruct-Fine-Tuned-on-Deepseek-
Research-Papers/pytorch_model-00003-of-00004.bin...
Unsloth: Saving AkinduH/Qwen2.5-3B-Instruct-Fine-Tuned-on-Deepseek-
```

```
Research-Papers/pytorch_model-00004-of-00004.bin...
Done.
==((====))==  Unsloth: Conversion from QLoRA to GGUF information
   \\   /|    [0] Installing llama.cpp might take 3 minutes.
O^O/ \_/ \    [1] Converting HF to GGUF 16bits might take 3 minutes.
\        /    [2] Converting GGUF 16bits to ['q4_k_m'] might take 10
minutes each.
 "-____-"     In total, you will have to wait at least 16 minutes.

Unsloth: Installing llama.cpp. This might take 3 minutes...
Unsloth: [1] Converting model at AkinduH/Qwen2.5-3B-Instruct-Fine-
Tuned-on-Deepseek-Research-Papers into f16 GGUF format.
The output location will be /content/AkinduH/Qwen2.5-3B-Instruct-Fine-
Tuned-on-Deepseek-Research-Papers/unsloth.F16.gguf
This might take 3 minutes...
INFO:hf-to-gguf:Loading model: Qwen2.5-3B-Instruct-Fine-Tuned-on-
Deepseek-Research-Papers
INFO:gguf.gguf_writer:gguf: This GGUF file is for Little Endian only
INFO:hf-to-gguf:Exporting model...
INFO:hf-to-gguf:gguf: loading model weight map from
'pytorch_model.bin.index.json'
INFO:hf-to-gguf:gguf: loading model part 'pytorch_model-00001-of-
00004.bin'
INFO:hf-to-gguf:token_embd.weight,         torch.float16 --> F16,
shape = {3584, 152064}
INFO:hf-to-gguf:blk.0.attn_q.bias,         torch.float16 --> F32,
shape = {3584}
INFO:hf-to-gguf:blk.0.attn_q.weight,       torch.float16 --> F16,
shape = {3584, 3584}
INFO:hf-to-gguf:blk.0.attn_k.bias,         torch.float16 --> F32,
shape = {512}
INFO:hf-to-gguf:blk.0.attn_k.weight,       torch.float16 --> F16,
shape = {3584, 512}
INFO:hf-to-gguf:blk.0.attn_v.bias,         torch.float16 --> F32,
shape = {512}
INFO:hf-to-gguf:blk.0.attn_v.weight,       torch.float16 --> F16,
shape = {3584, 512}
INFO:hf-to-gguf:blk.0.attn_output.weight,  torch.float16 --> F16,
shape = {3584, 3584}
INFO:hf-to-gguf:blk.0.ffn_gate.weight,     torch.float16 --> F16,
shape = {3584, 18944}
INFO:hf-to-gguf:blk.0.ffn_up.weight,       torch.float16 --> F16,
shape = {3584, 18944}
INFO:hf-to-gguf:blk.0.ffn_down.weight,     torch.float16 --> F16,
shape = {18944, 3584}
INFO:hf-to-gguf:blk.0.attn_norm.weight,    torch.float16 --> F32,
shape = {3584}
INFO:hf-to-gguf:blk.0.ffn_norm.weight,     torch.float16 --> F32,
shape = {3584}
```

```
INFO:hf-to-gguf:blk.1.attn_q.bias,          torch.float16 --> F32,
shape = {3584}
INFO:hf-to-gguf:blk.1.attn_q.weight,        torch.float16 --> F16,
shape = {3584, 3584}
INFO:hf-to-gguf:blk.1.attn_k.bias,          torch.float16 --> F32,
shape = {512}
INFO:hf-to-gguf:blk.1.attn_k.weight,        torch.float16 --> F16,
shape = {3584, 512}
INFO:hf-to-gguf:blk.1.attn_v.bias,          torch.float16 --> F32,
shape = {512}
INFO:hf-to-gguf:blk.1.attn_v.weight,        torch.float16 --> F16,
shape = {3584, 512}
INFO:hf-to-gguf:blk.1.attn_output.weight,   torch.float16 --> F16,
shape = {3584, 3584}
INFO:hf-to-gguf:blk.1.ffn_gate.weight,      torch.float16 --> F16,
shape = {3584, 18944}
INFO:hf-to-gguf:blk.1.ffn_up.weight,        torch.float16 --> F16,
shape = {3584, 18944}
INFO:hf-to-gguf:blk.1.ffn_down.weight,      torch.float16 --> F16,
shape = {18944, 3584}
INFO:hf-to-gguf:blk.1.attn_norm.weight,     torch.float16 --> F32,
shape = {3584}
INFO:hf-to-gguf:blk.1.ffn_norm.weight,      torch.float16 --> F32,
shape = {3584}
INFO:hf-to-gguf:blk.2.attn_q.bias,          torch.float16 --> F32,
shape = {3584}
INFO:hf-to-gguf:blk.2.attn_q.weight,        torch.float16 --> F16,
shape = {3584, 3584}
INFO:hf-to-gguf:blk.2.attn_k.bias,          torch.float16 --> F32,
shape = {512}
INFO:hf-to-gguf:blk.2.attn_k.weight,        torch.float16 --> F16,
shape = {3584, 512}
INFO:hf-to-gguf:blk.2.attn_v.bias,          torch.float16 --> F32,
shape = {512}
INFO:hf-to-gguf:blk.2.attn_v.weight,        torch.float16 --> F16,
shape = {3584, 512}
INFO:hf-to-gguf:blk.2.attn_output.weight,   torch.float16 --> F16,
shape = {3584, 3584}
INFO:hf-to-gguf:blk.2.ffn_gate.weight,      torch.float16 --> F16,
shape = {3584, 18944}
INFO:hf-to-gguf:blk.2.ffn_up.weight,        torch.float16 --> F16,
shape = {3584, 18944}
INFO:hf-to-gguf:blk.2.ffn_down.weight,      torch.float16 --> F16,
shape = {18944, 3584}
INFO:hf-to-gguf:blk.2.attn_norm.weight,     torch.float16 --> F32,
shape = {3584}
INFO:hf-to-gguf:blk.2.ffn_norm.weight,      torch.float16 --> F32,
shape = {3584}
INFO:hf-to-gguf:blk.3.attn_q.bias,          torch.float16 --> F32,
```

```
shape = {3584}
INFO:hf-to-gguf:blk.3.attn_q.weight,       torch.float16 --> F16,
shape = {3584, 3584}
INFO:hf-to-gguf:blk.3.attn_k.bias,         torch.float16 --> F32,
shape = {512}
INFO:hf-to-gguf:blk.3.attn_k.weight,       torch.float16 --> F16,
shape = {3584, 512}
INFO:hf-to-gguf:blk.3.attn_v.bias,         torch.float16 --> F32,
shape = {512}
INFO:hf-to-gguf:blk.3.attn_v.weight,       torch.float16 --> F16,
shape = {3584, 512}
INFO:hf-to-gguf:blk.3.attn_output.weight,  torch.float16 --> F16,
shape = {3584, 3584}
INFO:hf-to-gguf:blk.3.ffn_gate.weight,     torch.float16 --> F16,
shape = {3584, 18944}
INFO:hf-to-gguf:blk.3.ffn_up.weight,       torch.float16 --> F16,
shape = {3584, 18944}
INFO:hf-to-gguf:blk.3.ffn_down.weight,     torch.float16 --> F16,
shape = {18944, 3584}
INFO:hf-to-gguf:blk.3.attn_norm.weight,    torch.float16 --> F32,
shape = {3584}
INFO:hf-to-gguf:blk.3.ffn_norm.weight,     torch.float16 --> F32,
shape = {3584}
INFO:hf-to-gguf:blk.4.attn_q.bias,         torch.float16 --> F32,
shape = {3584}
INFO:hf-to-gguf:blk.4.attn_q.weight,       torch.float16 --> F16,
shape = {3584, 3584}
INFO:hf-to-gguf:blk.4.attn_k.bias,         torch.float16 --> F32,
shape = {512}
INFO:hf-to-gguf:blk.4.attn_k.weight,       torch.float16 --> F16,
shape = {3584, 512}
INFO:hf-to-gguf:blk.4.attn_v.bias,         torch.float16 --> F32,
shape = {512}
INFO:hf-to-gguf:blk.4.attn_v.weight,       torch.float16 --> F16,
shape = {3584, 512}
INFO:hf-to-gguf:blk.4.attn_output.weight,  torch.float16 --> F16,
shape = {3584, 3584}
INFO:hf-to-gguf:blk.4.ffn_gate.weight,     torch.float16 --> F16,
shape = {3584, 18944}
INFO:hf-to-gguf:blk.4.ffn_up.weight,       torch.float16 --> F16,
shape = {3584, 18944}
INFO:hf-to-gguf:blk.4.ffn_down.weight,     torch.float16 --> F16,
shape = {18944, 3584}
INFO:hf-to-gguf:blk.4.attn_norm.weight,    torch.float16 --> F32,
shape = {3584}
INFO:hf-to-gguf:blk.4.ffn_norm.weight,     torch.float16 --> F32,
shape = {3584}
INFO:hf-to-gguf:blk.5.attn_q.bias,         torch.float16 --> F32,
shape = {3584}
```

```
INFO:hf-to-gguf:blk.5.attn_q.weight,       torch.float16 --> F16,
shape = {3584, 3584}
INFO:hf-to-gguf:blk.5.attn_k.bias,         torch.float16 --> F32,
shape = {512}
INFO:hf-to-gguf:blk.5.attn_k.weight,       torch.float16 --> F16,
shape = {3584, 512}
INFO:hf-to-gguf:blk.5.attn_v.bias,         torch.float16 --> F32,
shape = {512}
INFO:hf-to-gguf:blk.5.attn_v.weight,       torch.float16 --> F16,
shape = {3584, 512}
INFO:hf-to-gguf:blk.5.attn_output.weight,  torch.float16 --> F16,
shape = {3584, 3584}
INFO:hf-to-gguf:blk.5.ffn_gate.weight,     torch.float16 --> F16,
shape = {3584, 18944}
INFO:hf-to-gguf:blk.5.ffn_up.weight,       torch.float16 --> F16,
shape = {3584, 18944}
INFO:hf-to-gguf:blk.5.ffn_down.weight,     torch.float16 --> F16,
shape = {18944, 3584}
INFO:hf-to-gguf:blk.5.attn_norm.weight,    torch.float16 --> F32,
shape = {3584}
INFO:hf-to-gguf:blk.5.ffn_norm.weight,     torch.float16 --> F32,
shape = {3584}
INFO:hf-to-gguf:blk.6.attn_q.bias,         torch.float16 --> F32,
shape = {3584}
INFO:hf-to-gguf:blk.6.attn_q.weight,       torch.float16 --> F16,
shape = {3584, 3584}
INFO:hf-to-gguf:blk.6.attn_k.bias,         torch.float16 --> F32,
shape = {512}
INFO:hf-to-gguf:blk.6.attn_k.weight,       torch.float16 --> F16,
shape = {3584, 512}
INFO:hf-to-gguf:blk.6.attn_v.bias,         torch.float16 --> F32,
shape = {512}
INFO:hf-to-gguf:blk.6.attn_v.weight,       torch.float16 --> F16,
shape = {3584, 512}
INFO:hf-to-gguf:blk.6.attn_output.weight,  torch.float16 --> F16,
shape = {3584, 3584}
INFO:hf-to-gguf:blk.6.ffn_gate.weight,     torch.float16 --> F16,
shape = {3584, 18944}
INFO:hf-to-gguf:blk.6.ffn_up.weight,       torch.float16 --> F16,
shape = {3584, 18944}
INFO:hf-to-gguf:blk.6.ffn_down.weight,     torch.float16 --> F16,
shape = {18944, 3584}
INFO:hf-to-gguf:blk.6.attn_norm.weight,    torch.float16 --> F32,
shape = {3584}
INFO:hf-to-gguf:blk.6.ffn_norm.weight,     torch.float16 --> F32,
shape = {3584}
INFO:hf-to-gguf:blk.7.attn_q.bias,         torch.float16 --> F32,
shape = {3584}
INFO:hf-to-gguf:blk.7.attn_q.weight,       torch.float16 --> F16,
```

```
shape = {3584, 3584}
INFO:hf-to-gguf:blk.7.attn_k.bias,          torch.float16 --> F32,
shape = {512}
INFO:hf-to-gguf:blk.7.attn_k.weight,        torch.float16 --> F16,
shape = {3584, 512}
INFO:hf-to-gguf:blk.7.attn_v.bias,          torch.float16 --> F32,
shape = {512}
INFO:hf-to-gguf:blk.7.attn_v.weight,        torch.float16 --> F16,
shape = {3584, 512}
INFO:hf-to-gguf:blk.7.attn_output.weight,   torch.float16 --> F16,
shape = {3584, 3584}
INFO:hf-to-gguf:blk.7.ffn_gate.weight,      torch.float16 --> F16,
shape = {3584, 18944}
INFO:hf-to-gguf:blk.7.ffn_up.weight,        torch.float16 --> F16,
shape = {3584, 18944}
INFO:hf-to-gguf:blk.7.ffn_down.weight,      torch.float16 --> F16,
shape = {18944, 3584}
INFO:hf-to-gguf:blk.7.attn_norm.weight,     torch.float16 --> F32,
shape = {3584}
INFO:hf-to-gguf:blk.7.ffn_norm.weight,      torch.float16 --> F32,
shape = {3584}
INFO:hf-to-gguf:blk.8.attn_q.bias,          torch.float16 --> F32,
shape = {3584}
INFO:hf-to-gguf:blk.8.attn_q.weight,        torch.float16 --> F16,
shape = {3584, 3584}
INFO:hf-to-gguf:blk.8.attn_k.bias,          torch.float16 --> F32,
shape = {512}
INFO:hf-to-gguf:blk.8.attn_k.weight,        torch.float16 --> F16,
shape = {3584, 512}
INFO:hf-to-gguf:blk.8.attn_v.bias,          torch.float16 --> F32,
shape = {512}
INFO:hf-to-gguf:blk.8.attn_v.weight,        torch.float16 --> F16,
shape = {3584, 512}
INFO:hf-to-gguf:blk.8.attn_output.weight,   torch.float16 --> F16,
shape = {3584, 3584}
INFO:hf-to-gguf:gguf: loading model part 'pytorch_model-00002-of-
00004.bin'
INFO:hf-to-gguf:blk.8.ffn_gate.weight,      torch.float16 --> F16,
shape = {3584, 18944}
INFO:hf-to-gguf:blk.8.ffn_up.weight,        torch.float16 --> F16,
shape = {3584, 18944}
INFO:hf-to-gguf:blk.8.ffn_down.weight,      torch.float16 --> F16,
shape = {18944, 3584}
INFO:hf-to-gguf:blk.8.attn_norm.weight,     torch.float16 --> F32,
shape = {3584}
INFO:hf-to-gguf:blk.8.ffn_norm.weight,      torch.float16 --> F32,
shape = {3584}
INFO:hf-to-gguf:blk.9.attn_q.bias,          torch.float16 --> F32,
shape = {3584}
```

```
INFO:hf-to-gguf:blk.9.attn_q.weight,       torch.float16 --> F16,
shape = {3584, 3584}
INFO:hf-to-gguf:blk.9.attn_k.bias,         torch.float16 --> F32,
shape = {512}
INFO:hf-to-gguf:blk.9.attn_k.weight,       torch.float16 --> F16,
shape = {3584, 512}
INFO:hf-to-gguf:blk.9.attn_v.bias,         torch.float16 --> F32,
shape = {512}
INFO:hf-to-gguf:blk.9.attn_v.weight,       torch.float16 --> F16,
shape = {3584, 512}
INFO:hf-to-gguf:blk.9.attn_output.weight,  torch.float16 --> F16,
shape = {3584, 3584}
INFO:hf-to-gguf:blk.9.ffn_gate.weight,     torch.float16 --> F16,
shape = {3584, 18944}
INFO:hf-to-gguf:blk.9.ffn_up.weight,       torch.float16 --> F16,
shape = {3584, 18944}
INFO:hf-to-gguf:blk.9.ffn_down.weight,     torch.float16 --> F16,
shape = {18944, 3584}
INFO:hf-to-gguf:blk.9.attn_norm.weight,    torch.float16 --> F32,
shape = {3584}
INFO:hf-to-gguf:blk.9.ffn_norm.weight,     torch.float16 --> F32,
shape = {3584}
INFO:hf-to-gguf:blk.10.attn_q.bias,        torch.float16 --> F32,
shape = {3584}
INFO:hf-to-gguf:blk.10.attn_q.weight,      torch.float16 --> F16,
shape = {3584, 3584}
INFO:hf-to-gguf:blk.10.attn_k.bias,        torch.float16 --> F32,
shape = {512}
INFO:hf-to-gguf:blk.10.attn_k.weight,      torch.float16 --> F16,
shape = {3584, 512}
INFO:hf-to-gguf:blk.10.attn_v.bias,        torch.float16 --> F32,
shape = {512}
INFO:hf-to-gguf:blk.10.attn_v.weight,      torch.float16 --> F16,
shape = {3584, 512}
INFO:hf-to-gguf:blk.10.attn_output.weight, torch.float16 --> F16,
shape = {3584, 3584}
INFO:hf-to-gguf:blk.10.ffn_gate.weight,    torch.float16 --> F16,
shape = {3584, 18944}
INFO:hf-to-gguf:blk.10.ffn_up.weight,      torch.float16 --> F16,
shape = {3584, 18944}
INFO:hf-to-gguf:blk.10.ffn_down.weight,    torch.float16 --> F16,
shape = {18944, 3584}
INFO:hf-to-gguf:blk.10.attn_norm.weight,   torch.float16 --> F32,
shape = {3584}
INFO:hf-to-gguf:blk.10.ffn_norm.weight,    torch.float16 --> F32,
shape = {3584}
INFO:hf-to-gguf:blk.11.attn_q.bias,        torch.float16 --> F32,
shape = {3584}
INFO:hf-to-gguf:blk.11.attn_q.weight,      torch.float16 --> F16,
```

```
shape = {3584, 3584}
INFO:hf-to-gguf:blk.11.attn_k.bias,        torch.float16 --> F32,
shape = {512}
INFO:hf-to-gguf:blk.11.attn_k.weight,      torch.float16 --> F16,
shape = {3584, 512}
INFO:hf-to-gguf:blk.11.attn_v.bias,        torch.float16 --> F32,
shape = {512}
INFO:hf-to-gguf:blk.11.attn_v.weight,      torch.float16 --> F16,
shape = {3584, 512}
INFO:hf-to-gguf:blk.11.attn_output.weight, torch.float16 --> F16,
shape = {3584, 3584}
INFO:hf-to-gguf:blk.11.ffn_gate.weight,    torch.float16 --> F16,
shape = {3584, 18944}
INFO:hf-to-gguf:blk.11.ffn_up.weight,      torch.float16 --> F16,
shape = {3584, 18944}
INFO:hf-to-gguf:blk.11.ffn_down.weight,    torch.float16 --> F16,
shape = {18944, 3584}
INFO:hf-to-gguf:blk.11.attn_norm.weight,   torch.float16 --> F32,
shape = {3584}
INFO:hf-to-gguf:blk.11.ffn_norm.weight,    torch.float16 --> F32,
shape = {3584}
INFO:hf-to-gguf:blk.12.attn_q.bias,        torch.float16 --> F32,
shape = {3584}
INFO:hf-to-gguf:blk.12.attn_q.weight,      torch.float16 --> F16,
shape = {3584, 3584}
INFO:hf-to-gguf:blk.12.attn_k.bias,        torch.float16 --> F32,
shape = {512}
INFO:hf-to-gguf:blk.12.attn_k.weight,      torch.float16 --> F16,
shape = {3584, 512}
INFO:hf-to-gguf:blk.12.attn_v.bias,        torch.float16 --> F32,
shape = {512}
INFO:hf-to-gguf:blk.12.attn_v.weight,      torch.float16 --> F16,
shape = {3584, 512}
INFO:hf-to-gguf:blk.12.attn_output.weight, torch.float16 --> F16,
shape = {3584, 3584}
INFO:hf-to-gguf:blk.12.ffn_gate.weight,    torch.float16 --> F16,
shape = {3584, 18944}
INFO:hf-to-gguf:blk.12.ffn_up.weight,      torch.float16 --> F16,
shape = {3584, 18944}
INFO:hf-to-gguf:blk.12.ffn_down.weight,    torch.float16 --> F16,
shape = {18944, 3584}
INFO:hf-to-gguf:blk.12.attn_norm.weight,   torch.float16 --> F32,
shape = {3584}
INFO:hf-to-gguf:blk.12.ffn_norm.weight,    torch.float16 --> F32,
shape = {3584}
INFO:hf-to-gguf:blk.13.attn_q.bias,        torch.float16 --> F32,
shape = {3584}
INFO:hf-to-gguf:blk.13.attn_q.weight,      torch.float16 --> F16,
shape = {3584, 3584}
```

```
INFO:hf-to-gguf:blk.13.attn_k.bias,         torch.float16 --> F32,
shape = {512}
INFO:hf-to-gguf:blk.13.attn_k.weight,       torch.float16 --> F16,
shape = {3584, 512}
INFO:hf-to-gguf:blk.13.attn_v.bias,         torch.float16 --> F32,
shape = {512}
INFO:hf-to-gguf:blk.13.attn_v.weight,       torch.float16 --> F16,
shape = {3584, 512}
INFO:hf-to-gguf:blk.13.attn_output.weight, torch.float16 --> F16,
shape = {3584, 3584}
INFO:hf-to-gguf:blk.13.ffn_gate.weight,     torch.float16 --> F16,
shape = {3584, 18944}
INFO:hf-to-gguf:blk.13.ffn_up.weight,       torch.float16 --> F16,
shape = {3584, 18944}
INFO:hf-to-gguf:blk.13.ffn_down.weight,     torch.float16 --> F16,
shape = {18944, 3584}
INFO:hf-to-gguf:blk.13.attn_norm.weight,    torch.float16 --> F32,
shape = {3584}
INFO:hf-to-gguf:blk.13.ffn_norm.weight,     torch.float16 --> F32,
shape = {3584}
INFO:hf-to-gguf:blk.14.attn_q.bias,         torch.float16 --> F32,
shape = {3584}
INFO:hf-to-gguf:blk.14.attn_q.weight,       torch.float16 --> F16,
shape = {3584, 3584}
INFO:hf-to-gguf:blk.14.attn_k.bias,         torch.float16 --> F32,
shape = {512}
INFO:hf-to-gguf:blk.14.attn_k.weight,       torch.float16 --> F16,
shape = {3584, 512}
INFO:hf-to-gguf:blk.14.attn_v.bias,         torch.float16 --> F32,
shape = {512}
INFO:hf-to-gguf:blk.14.attn_v.weight,       torch.float16 --> F16,
shape = {3584, 512}
INFO:hf-to-gguf:blk.14.attn_output.weight, torch.float16 --> F16,
shape = {3584, 3584}
INFO:hf-to-gguf:blk.14.ffn_gate.weight,     torch.float16 --> F16,
shape = {3584, 18944}
INFO:hf-to-gguf:blk.14.ffn_up.weight,       torch.float16 --> F16,
shape = {3584, 18944}
INFO:hf-to-gguf:blk.14.ffn_down.weight,     torch.float16 --> F16,
shape = {18944, 3584}
INFO:hf-to-gguf:blk.14.attn_norm.weight,    torch.float16 --> F32,
shape = {3584}
INFO:hf-to-gguf:blk.14.ffn_norm.weight,     torch.float16 --> F32,
shape = {3584}
INFO:hf-to-gguf:blk.15.attn_q.bias,         torch.float16 --> F32,
shape = {3584}
INFO:hf-to-gguf:blk.15.attn_q.weight,       torch.float16 --> F16,
shape = {3584, 3584}
INFO:hf-to-gguf:blk.15.attn_k.bias,         torch.float16 --> F32,
```

```
shape = {512}
INFO:hf-to-gguf:blk.15.attn_k.weight,      torch.float16 --> F16,
shape = {3584, 512}
INFO:hf-to-gguf:blk.15.attn_v.bias,        torch.float16 --> F32,
shape = {512}
INFO:hf-to-gguf:blk.15.attn_v.weight,      torch.float16 --> F16,
shape = {3584, 512}
INFO:hf-to-gguf:blk.15.attn_output.weight, torch.float16 --> F16,
shape = {3584, 3584}
INFO:hf-to-gguf:blk.15.ffn_gate.weight,    torch.float16 --> F16,
shape = {3584, 18944}
INFO:hf-to-gguf:blk.15.ffn_up.weight,      torch.float16 --> F16,
shape = {3584, 18944}
INFO:hf-to-gguf:blk.15.ffn_down.weight,    torch.float16 --> F16,
shape = {18944, 3584}
INFO:hf-to-gguf:blk.15.attn_norm.weight,   torch.float16 --> F32,
shape = {3584}
INFO:hf-to-gguf:blk.15.ffn_norm.weight,    torch.float16 --> F32,
shape = {3584}
INFO:hf-to-gguf:blk.16.attn_q.bias,        torch.float16 --> F32,
shape = {3584}
INFO:hf-to-gguf:blk.16.attn_q.weight,      torch.float16 --> F16,
shape = {3584, 3584}
INFO:hf-to-gguf:blk.16.attn_k.bias,        torch.float16 --> F32,
shape = {512}
INFO:hf-to-gguf:blk.16.attn_k.weight,      torch.float16 --> F16,
shape = {3584, 512}
INFO:hf-to-gguf:blk.16.attn_v.bias,        torch.float16 --> F32,
shape = {512}
INFO:hf-to-gguf:blk.16.attn_v.weight,      torch.float16 --> F16,
shape = {3584, 512}
INFO:hf-to-gguf:blk.16.attn_output.weight, torch.float16 --> F16,
shape = {3584, 3584}
INFO:hf-to-gguf:blk.16.ffn_gate.weight,    torch.float16 --> F16,
shape = {3584, 18944}
INFO:hf-to-gguf:blk.16.ffn_up.weight,      torch.float16 --> F16,
shape = {3584, 18944}
INFO:hf-to-gguf:blk.16.ffn_down.weight,    torch.float16 --> F16,
shape = {18944, 3584}
INFO:hf-to-gguf:blk.16.attn_norm.weight,   torch.float16 --> F32,
shape = {3584}
INFO:hf-to-gguf:blk.16.ffn_norm.weight,    torch.float16 --> F32,
shape = {3584}
INFO:hf-to-gguf:blk.17.attn_q.bias,        torch.float16 --> F32,
shape = {3584}
INFO:hf-to-gguf:blk.17.attn_q.weight,      torch.float16 --> F16,
shape = {3584, 3584}
INFO:hf-to-gguf:blk.17.attn_k.bias,        torch.float16 --> F32,
shape = {512}
```

```
INFO:hf-to-gguf:blk.17.attn_k.weight,      torch.float16 --> F16,
shape = {3584, 512}
INFO:hf-to-gguf:blk.17.attn_v.bias,        torch.float16 --> F32,
shape = {512}
INFO:hf-to-gguf:blk.17.attn_v.weight,      torch.float16 --> F16,
shape = {3584, 512}
INFO:hf-to-gguf:blk.17.attn_output.weight, torch.float16 --> F16,
shape = {3584, 3584}
INFO:hf-to-gguf:blk.17.ffn_gate.weight,    torch.float16 --> F16,
shape = {3584, 18944}
INFO:hf-to-gguf:blk.17.ffn_up.weight,      torch.float16 --> F16,
shape = {3584, 18944}
INFO:hf-to-gguf:blk.17.ffn_down.weight,    torch.float16 --> F16,
shape = {18944, 3584}
INFO:hf-to-gguf:blk.17.attn_norm.weight,   torch.float16 --> F32,
shape = {3584}
INFO:hf-to-gguf:blk.17.ffn_norm.weight,    torch.float16 --> F32,
shape = {3584}
INFO:hf-to-gguf:blk.18.attn_q.bias,        torch.float16 --> F32,
shape = {3584}
INFO:hf-to-gguf:blk.18.attn_q.weight,      torch.float16 --> F16,
shape = {3584, 3584}
INFO:hf-to-gguf:blk.18.attn_k.bias,        torch.float16 --> F32,
shape = {512}
INFO:hf-to-gguf:blk.18.attn_k.weight,      torch.float16 --> F16,
shape = {3584, 512}
INFO:hf-to-gguf:blk.18.attn_v.bias,        torch.float16 --> F32,
shape = {512}
INFO:hf-to-gguf:blk.18.attn_v.weight,      torch.float16 --> F16,
shape = {3584, 512}
INFO:hf-to-gguf:blk.18.attn_output.weight, torch.float16 --> F16,
shape = {3584, 3584}
INFO:hf-to-gguf:blk.18.ffn_gate.weight,    torch.float16 --> F16,
shape = {3584, 18944}
INFO:hf-to-gguf:blk.18.ffn_up.weight,      torch.float16 --> F16,
shape = {3584, 18944}
INFO:hf-to-gguf:gguf: loading model part 'pytorch_model-00003-of-
00004.bin'
INFO:hf-to-gguf:blk.18.ffn_down.weight,    torch.float16 --> F16,
shape = {18944, 3584}
INFO:hf-to-gguf:blk.18.attn_norm.weight,   torch.float16 --> F32,
shape = {3584}
INFO:hf-to-gguf:blk.18.ffn_norm.weight,    torch.float16 --> F32,
shape = {3584}
INFO:hf-to-gguf:blk.19.attn_q.bias,        torch.float16 --> F32,
shape = {3584}
INFO:hf-to-gguf:blk.19.attn_q.weight,      torch.float16 --> F16,
shape = {3584, 3584}
INFO:hf-to-gguf:blk.19.attn_k.bias,        torch.float16 --> F32,
shape = {512}
```

```
INFO:hf-to-gguf:blk.19.attn_k.weight,      torch.float16 --> F16,
shape = {3584, 512}
INFO:hf-to-gguf:blk.19.attn_v.bias,        torch.float16 --> F32,
shape = {512}
INFO:hf-to-gguf:blk.19.attn_v.weight,      torch.float16 --> F16,
shape = {3584, 512}
INFO:hf-to-gguf:blk.19.attn_output.weight, torch.float16 --> F16,
shape = {3584, 3584}
INFO:hf-to-gguf:blk.19.ffn_gate.weight,    torch.float16 --> F16,
shape = {3584, 18944}
INFO:hf-to-gguf:blk.19.ffn_up.weight,      torch.float16 --> F16,
shape = {3584, 18944}
INFO:hf-to-gguf:blk.19.ffn_down.weight,    torch.float16 --> F16,
shape = {18944, 3584}
INFO:hf-to-gguf:blk.19.attn_norm.weight,   torch.float16 --> F32,
shape = {3584}
INFO:hf-to-gguf:blk.19.ffn_norm.weight,    torch.float16 --> F32,
shape = {3584}
INFO:hf-to-gguf:blk.20.attn_q.bias,        torch.float16 --> F32,
shape = {3584}
INFO:hf-to-gguf:blk.20.attn_q.weight,      torch.float16 --> F16,
shape = {3584, 3584}
INFO:hf-to-gguf:blk.20.attn_k.bias,        torch.float16 --> F32,
shape = {512}
INFO:hf-to-gguf:blk.20.attn_k.weight,      torch.float16 --> F16,
shape = {3584, 512}
INFO:hf-to-gguf:blk.20.attn_v.bias,        torch.float16 --> F32,
shape = {512}
INFO:hf-to-gguf:blk.20.attn_v.weight,      torch.float16 --> F16,
shape = {3584, 512}
INFO:hf-to-gguf:blk.20.attn_output.weight, torch.float16 --> F16,
shape = {3584, 3584}
INFO:hf-to-gguf:blk.20.ffn_gate.weight,    torch.float16 --> F16,
shape = {3584, 18944}
INFO:hf-to-gguf:blk.20.ffn_up.weight,      torch.float16 --> F16,
shape = {3584, 18944}
INFO:hf-to-gguf:blk.20.ffn_down.weight,    torch.float16 --> F16,
shape = {18944, 3584}
INFO:hf-to-gguf:blk.20.attn_norm.weight,   torch.float16 --> F32,
shape = {3584}
INFO:hf-to-gguf:blk.20.ffn_norm.weight,    torch.float16 --> F32,
shape = {3584}
INFO:hf-to-gguf:blk.21.attn_q.bias,        torch.float16 --> F32,
shape = {3584}
INFO:hf-to-gguf:blk.21.attn_q.weight,      torch.float16 --> F16,
shape = {3584, 3584}
INFO:hf-to-gguf:blk.21.attn_k.bias,        torch.float16 --> F32,
shape = {512}
INFO:hf-to-gguf:blk.21.attn_k.weight,      torch.float16 --> F16,
```

```
shape = {3584, 512}
INFO:hf-to-gguf:blk.21.attn_v.bias,        torch.float16 --> F32,
shape = {512}
INFO:hf-to-gguf:blk.21.attn_v.weight,      torch.float16 --> F16,
shape = {3584, 512}
INFO:hf-to-gguf:blk.21.attn_output.weight, torch.float16 --> F16,
shape = {3584, 3584}
INFO:hf-to-gguf:blk.21.ffn_gate.weight,    torch.float16 --> F16,
shape = {3584, 18944}
INFO:hf-to-gguf:blk.21.ffn_up.weight,      torch.float16 --> F16,
shape = {3584, 18944}
INFO:hf-to-gguf:blk.21.ffn_down.weight,    torch.float16 --> F16,
shape = {18944, 3584}
INFO:hf-to-gguf:blk.21.attn_norm.weight,   torch.float16 --> F32,
shape = {3584}
INFO:hf-to-gguf:blk.21.ffn_norm.weight,    torch.float16 --> F32,
shape = {3584}
INFO:hf-to-gguf:blk.22.attn_q.bias,        torch.float16 --> F32,
shape = {3584}
INFO:hf-to-gguf:blk.22.attn_q.weight,      torch.float16 --> F16,
shape = {3584, 3584}
INFO:hf-to-gguf:blk.22.attn_k.bias,        torch.float16 --> F32,
shape = {512}
INFO:hf-to-gguf:blk.22.attn_k.weight,      torch.float16 --> F16,
shape = {3584, 512}
INFO:hf-to-gguf:blk.22.attn_v.bias,        torch.float16 --> F32,
shape = {512}
INFO:hf-to-gguf:blk.22.attn_v.weight,      torch.float16 --> F16,
shape = {3584, 512}
INFO:hf-to-gguf:blk.22.attn_output.weight, torch.float16 --> F16,
shape = {3584, 3584}
INFO:hf-to-gguf:blk.22.ffn_gate.weight,    torch.float16 --> F16,
shape = {3584, 18944}
INFO:hf-to-gguf:blk.22.ffn_up.weight,      torch.float16 --> F16,
shape = {3584, 18944}
INFO:hf-to-gguf:blk.22.ffn_down.weight,    torch.float16 --> F16,
shape = {18944, 3584}
INFO:hf-to-gguf:blk.22.attn_norm.weight,   torch.float16 --> F32,
shape = {3584}
INFO:hf-to-gguf:blk.22.ffn_norm.weight,    torch.float16 --> F32,
shape = {3584}
INFO:hf-to-gguf:blk.23.attn_q.bias,        torch.float16 --> F32,
shape = {3584}
INFO:hf-to-gguf:blk.23.attn_q.weight,      torch.float16 --> F16,
shape = {3584, 3584}
INFO:hf-to-gguf:blk.23.attn_k.bias,        torch.float16 --> F32,
shape = {512}
INFO:hf-to-gguf:blk.23.attn_k.weight,      torch.float16 --> F16,
shape = {3584, 512}
```

```
INFO:hf-to-gguf:blk.23.attn_v.bias,         torch.float16 --> F32,
shape = {512}
INFO:hf-to-gguf:blk.23.attn_v.weight,       torch.float16 --> F16,
shape = {3584, 512}
INFO:hf-to-gguf:blk.23.attn_output.weight, torch.float16 --> F16,
shape = {3584, 3584}
INFO:hf-to-gguf:blk.23.ffn_gate.weight,     torch.float16 --> F16,
shape = {3584, 18944}
INFO:hf-to-gguf:blk.23.ffn_up.weight,       torch.float16 --> F16,
shape = {3584, 18944}
INFO:hf-to-gguf:blk.23.ffn_down.weight,     torch.float16 --> F16,
shape = {18944, 3584}
INFO:hf-to-gguf:blk.23.attn_norm.weight,    torch.float16 --> F32,
shape = {3584}
INFO:hf-to-gguf:blk.23.ffn_norm.weight,     torch.float16 --> F32,
shape = {3584}
INFO:hf-to-gguf:blk.24.attn_q.bias,         torch.float16 --> F32,
shape = {3584}
INFO:hf-to-gguf:blk.24.attn_q.weight,       torch.float16 --> F16,
shape = {3584, 3584}
INFO:hf-to-gguf:blk.24.attn_k.bias,         torch.float16 --> F32,
shape = {512}
INFO:hf-to-gguf:blk.24.attn_k.weight,       torch.float16 --> F16,
shape = {3584, 512}
INFO:hf-to-gguf:blk.24.attn_v.bias,         torch.float16 --> F32,
shape = {512}
INFO:hf-to-gguf:blk.24.attn_v.weight,       torch.float16 --> F16,
shape = {3584, 512}
INFO:hf-to-gguf:blk.24.attn_output.weight, torch.float16 --> F16,
shape = {3584, 3584}
INFO:hf-to-gguf:blk.24.ffn_gate.weight,     torch.float16 --> F16,
shape = {3584, 18944}
INFO:hf-to-gguf:blk.24.ffn_up.weight,       torch.float16 --> F16,
shape = {3584, 18944}
INFO:hf-to-gguf:blk.24.ffn_down.weight,     torch.float16 --> F16,
shape = {18944, 3584}
INFO:hf-to-gguf:blk.24.attn_norm.weight,    torch.float16 --> F32,
shape = {3584}
INFO:hf-to-gguf:blk.24.ffn_norm.weight,     torch.float16 --> F32,
shape = {3584}
INFO:hf-to-gguf:blk.25.attn_q.bias,         torch.float16 --> F32,
shape = {3584}
INFO:hf-to-gguf:blk.25.attn_q.weight,       torch.float16 --> F16,
shape = {3584, 3584}
INFO:hf-to-gguf:blk.25.attn_k.bias,         torch.float16 --> F32,
shape = {512}
INFO:hf-to-gguf:blk.25.attn_k.weight,       torch.float16 --> F16,
shape = {3584, 512}
INFO:hf-to-gguf:blk.25.attn_v.bias,         torch.float16 --> F32,
```

```
shape = {512}
INFO:hf-to-gguf:blk.25.attn_v.weight,      torch.float16 --> F16,
shape = {3584, 512}
INFO:hf-to-gguf:blk.25.attn_output.weight, torch.float16 --> F16,
shape = {3584, 3584}
INFO:hf-to-gguf:blk.25.ffn_gate.weight,    torch.float16 --> F16,
shape = {3584, 18944}
INFO:hf-to-gguf:blk.25.ffn_up.weight,      torch.float16 --> F16,
shape = {3584, 18944}
INFO:hf-to-gguf:blk.25.ffn_down.weight,    torch.float16 --> F16,
shape = {18944, 3584}
INFO:hf-to-gguf:blk.25.attn_norm.weight,   torch.float16 --> F32,
shape = {3584}
INFO:hf-to-gguf:blk.25.ffn_norm.weight,    torch.float16 --> F32,
shape = {3584}
INFO:hf-to-gguf:blk.26.attn_q.bias,        torch.float16 --> F32,
shape = {3584}
INFO:hf-to-gguf:blk.26.attn_q.weight,      torch.float16 --> F16,
shape = {3584, 3584}
INFO:hf-to-gguf:blk.26.attn_k.bias,        torch.float16 --> F32,
shape = {512}
INFO:hf-to-gguf:blk.26.attn_k.weight,      torch.float16 --> F16,
shape = {3584, 512}
INFO:hf-to-gguf:blk.26.attn_v.bias,        torch.float16 --> F32,
shape = {512}
INFO:hf-to-gguf:blk.26.attn_v.weight,      torch.float16 --> F16,
shape = {3584, 512}
INFO:hf-to-gguf:blk.26.attn_output.weight, torch.float16 --> F16,
shape = {3584, 3584}
INFO:hf-to-gguf:blk.26.ffn_gate.weight,    torch.float16 --> F16,
shape = {3584, 18944}
INFO:hf-to-gguf:blk.26.ffn_up.weight,      torch.float16 --> F16,
shape = {3584, 18944}
INFO:hf-to-gguf:blk.26.ffn_down.weight,    torch.float16 --> F16,
shape = {18944, 3584}
INFO:hf-to-gguf:blk.26.attn_norm.weight,   torch.float16 --> F32,
shape = {3584}
INFO:hf-to-gguf:blk.26.ffn_norm.weight,    torch.float16 --> F32,
shape = {3584}
INFO:hf-to-gguf:blk.27.attn_q.bias,        torch.float16 --> F32,
shape = {3584}
INFO:hf-to-gguf:blk.27.attn_q.weight,      torch.float16 --> F16,
shape = {3584, 3584}
INFO:hf-to-gguf:blk.27.attn_k.bias,        torch.float16 --> F32,
shape = {512}
INFO:hf-to-gguf:blk.27.attn_k.weight,      torch.float16 --> F16,
shape = {3584, 512}
INFO:hf-to-gguf:blk.27.attn_v.bias,        torch.float16 --> F32,
shape = {512}
```

```
INFO:hf-to-gguf:blk.27.attn_v.weight,      torch.float16 --> F16,
shape = {3584, 512}
INFO:hf-to-gguf:blk.27.attn_output.weight, torch.float16 --> F16,
shape = {3584, 3584}
INFO:hf-to-gguf:blk.27.ffn_gate.weight,    torch.float16 --> F16,
shape = {3584, 18944}
INFO:hf-to-gguf:blk.27.ffn_up.weight,      torch.float16 --> F16,
shape = {3584, 18944}
INFO:hf-to-gguf:blk.27.ffn_down.weight,    torch.float16 --> F16,
shape = {18944, 3584}
INFO:hf-to-gguf:blk.27.attn_norm.weight,   torch.float16 --> F32,
shape = {3584}
INFO:hf-to-gguf:blk.27.ffn_norm.weight,    torch.float16 --> F32,
shape = {3584}
INFO:hf-to-gguf:output_norm.weight,        torch.float16 --> F32,
shape = {3584}
INFO:hf-to-gguf:gguf: loading model part 'pytorch_model-00004-of-
00004.bin'
INFO:hf-to-gguf:output.weight,             torch.float16 --> F16,
shape = {3584, 152064}
INFO:hf-to-gguf:Set meta model
INFO:hf-to-gguf:Set model parameters
INFO:hf-to-gguf:gguf: context length = 32768
INFO:hf-to-gguf:gguf: embedding length = 3584
INFO:hf-to-gguf:gguf: feed forward length = 18944
INFO:hf-to-gguf:gguf: head count = 28
INFO:hf-to-gguf:gguf: key-value head count = 4
INFO:hf-to-gguf:gguf: rope theta = 1000000.0
INFO:hf-to-gguf:gguf: rms norm epsilon = 1e-06
INFO:hf-to-gguf:gguf: file type = 1
INFO:hf-to-gguf:Set model tokenizer
INFO:numexpr.utils:NumExpr defaulting to 2 threads.
2025-03-09 18:34:44.813580: E
external/local_xla/xla/stream_executor/cuda/cuda_fft.cc:477] Unable to
register cuFFT factory: Attempting to register factory for plugin
cuFFT when one has already been registered
WARNING: All log messages before absl::InitializeLog() is called are
written to STDERR
E0000 00:00:1741545284.849072   52624 cuda_dnn.cc:8310] Unable to
register cuDNN factory: Attempting to register factory for plugin
cuDNN when one has already been registered
E0000 00:00:1741545284.862584   52624 cuda_blas.cc:1418] Unable to
register cuBLAS factory: Attempting to register factory for plugin
cuBLAS when one has already been registered
INFO:gguf.vocab:Adding 151387 merge(s).
INFO:gguf.vocab:Setting special token type eos to 151645
INFO:gguf.vocab:Setting special token type pad to 151654
INFO:gguf.vocab:Setting add_bos_token to False
INFO:gguf.vocab:Setting chat_template to {%- if tools %}
```

```
    {{- '<|im_start|>system\n' }}
    {%- if messages[0]['role'] == 'system' %}
        {{- messages[0]['content'] }}
    {%- else %}
        {{- 'You are Qwen, created by Alibaba Cloud. You are a helpful
assistant.' }}
    {%- endif %}
    {{- "\n\n# Tools\n\nYou may call one or more functions to assist
with the user query.\n\nYou are provided with function signatures
within <tools></tools> XML tags:\n<tools>" }}
    {%- for tool in tools %}
        {{- "\n" }}
        {{- tool | tojson }}
    {%- endfor %}
    {{- "\n</tools>\n\nFor each function call, return a json object
with function name and arguments within <tool_call></tool_call> XML
tags:\n<tool_call>\n{\"name\": <function-name>, \"arguments\": <args-
json-object>}\n</tool_call><|im_end|>\n" }}
{%- else %}
    {%- if messages[0]['role'] == 'system' %}
        {{- '<|im_start|>system\n' + messages[0]['content'] + '<|
im_end|>\n' }}
    {%- else %}
        {{- '<|im_start|>system\nYou are Qwen, created by Alibaba
Cloud. You are a helpful assistant.<|im_end|>\n' }}
    {%- endif %}
{%- endif %}
{%- for message in messages %}
    {%- if (message.role == "user") or (message.role == "system" and
not loop.first) or (message.role == "assistant" and not
message.tool_calls) %}
        {{- '<|im_start|>' + message.role + '\n' + message.content +
'<|im_end|>' + '\n' }}
    {%- elif message.role == "assistant" %}
        {{- '<|im_start|>' + message.role }}
        {%- if message.content %}
            {{- '\n' + message.content }}
        {%- endif %}
        {%- for tool_call in message.tool_calls %}
            {%- if tool_call.function is defined %}
                {%- set tool_call = tool_call.function %}
            {%- endif %}
            {{- '\n<tool_call>\n{"name": "' }}
            {{- tool_call.name }}
            {{- '", "arguments": ' }}
            {{- tool_call.arguments | tojson }}
            {{- '}\n</tool_call>' }}
        {%- endfor %}
        {{- '<|im_end|>\n' }}
```

```
    {%- elif message.role == "tool" %}
        {%- if (loop.index0 == 0) or (messages[loop.index0 - 1].role !
= "tool") %}            {{- '<|im_start|>user' }}
        {%- endif %}
        {{- '\n<tool_response>\n' }}
        {{- message.content }}
        {{- '\n</tool_response>' }}
        {%- if loop.last or (messages[loop.index0 + 1].role != "tool")
%}
            {{- '<|im_end|>\n' }}
        {%- endif %}
    {%- endif %}
{%- endfor %}
{%- if add_generation_prompt %}
    {{- '<|im_start|>assistant\n' }}
{%- endif %}

INFO:hf-to-gguf:Set model quantization version
INFO:gguf.gguf_writer:Writing the following files:
INFO:gguf.gguf_writer:/content/AkinduH/Qwen2.5-3B-Instruct-Fine-Tuned-
on-Deepseek-Research-Papers/unsloth.F16.gguf: n_tensors = 339,
total_size = 15.2G
Writing: 100%|██████████| 15.2G/15.2G [03:58<00:00, 64.0Mbyte/s]
INFO:hf-to-gguf:Model successfully exported to
/content/AkinduH/Qwen2.5-3B-Instruct-Fine-Tuned-on-Deepseek-Research-
Papers/unsloth.F16.gguf
Unsloth: Conversion completed! Output location:
/content/AkinduH/Qwen2.5-3B-Instruct-Fine-Tuned-on-Deepseek-Research-
Papers/unsloth.F16.gguf
Unsloth: [2] Converting GGUF 16bit into q4_k_m. This might take 20
minutes...
main: build = 4857 (0fd7ca7a)
main: built with cc (Ubuntu 11.4.0-1ubuntu1~22.04) 11.4.0 for x86_64-
linux-gnu
main: quantizing '/content/AkinduH/Qwen2.5-3B-Instruct-Fine-Tuned-on-
Deepseek-Research-Papers/unsloth.F16.gguf' to
'/content/AkinduH/Qwen2.5-3B-Instruct-Fine-Tuned-on-Deepseek-Research-
Papers/unsloth.Q4_K_M.gguf' as Q4_K_M using 4 threads
llama_model_loader: loaded meta data with 26 key-value pairs and 339
tensors from /content/AkinduH/Qwen2.5-3B-Instruct-Fine-Tuned-on-
Deepseek-Research-Papers/unsloth.F16.gguf (version GGUF V3 (latest))
llama_model_loader: Dumping metadata keys/values. Note: KV overrides
do not apply in this output.
llama_model_loader: - kv   0:
general.architecture str              = qwen2
llama_model_loader: - kv   1:
general.type str              = model
llama_model_loader: - kv   2:
general.name str              = Qwen2.5 7b Instruct Unsloth Bnb 4bit
```

```
llama_model_loader: - kv   3:
general.organization str              = Unsloth
llama_model_loader: - kv   4:
general.finetune str              = instruct-unsloth-bnb-4bit
llama_model_loader: - kv   5:
general.basename str              = qwen2.5
llama_model_loader: - kv   6:
general.size_label str              = 7B
llama_model_loader: - kv   7:
qwen2.block_count u32              = 28
llama_model_loader: - kv   8:
qwen2.context_length u32              = 32768
llama_model_loader: - kv   9:
qwen2.embedding_length u32              = 3584
llama_model_loader: - kv  10:
qwen2.feed_forward_length u32              = 18944
llama_model_loader: - kv  11:
qwen2.attention.head_count u32              = 28
llama_model_loader: - kv  12:
qwen2.attention.head_count_kv u32              = 4
llama_model_loader: - kv  13:
qwen2.rope.freq_base f32              = 1000000.000000
llama_model_loader: - kv  14:
qwen2.attention.layer_norm_rms_epsilon f32              = 0.000001
llama_model_loader: - kv  15:
general.file_type u32              = 1
llama_model_loader: - kv  16:
tokenizer.ggml.model str              = gpt2
llama_model_loader: - kv  17:
tokenizer.ggml.pre str              = qwen2
llama_model_loader: - kv  18:
tokenizer.ggml.tokens arr[str,152064]  = ["!", "\"", "#", "$", "%",
"&", "'", ...
llama_model_loader: - kv  19:
tokenizer.ggml.token_type arr[i32,152064]  = [1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, ...
llama_model_loader: - kv  20:
tokenizer.ggml.merges arr[str,151387]  = ["Ġ Ġ", "ĠĠ ĠĠ", "i n", "Ġ
t",...
llama_model_loader: - kv  21:
tokenizer.ggml.eos_token_id u32              = 151645
llama_model_loader: - kv  22:
tokenizer.ggml.padding_token_id u32              = 151654
llama_model_loader: - kv  23:
tokenizer.ggml.add_bos_token bool             = false
llama_model_loader: - kv  24:
tokenizer.chat_template str              = {%- if tools %}\n    {{-
'<|im_start|>...
llama_model_loader: - kv  25:
```

```
general.quantization_version u32              = 2
llama_model_loader: - type  f32:  141 tensors
llama_model_loader: - type  f16:  198 tensors
[   1/ 339]                         output.weight - [ 3584, 152064,
1,     1], type =    f16, converting to q6_K .. size =  1039.50 MiB ->
426.36 MiB
[   2/ 339]                   output_norm.weight - [ 3584,     1,
1,     1], type =    f32, size =    0.014 MB
[   3/ 339]                    token_embd.weight - [ 3584, 152064,
1,     1], type =    f16, converting to q4_K .. size =  1039.50 MiB ->
292.36 MiB
[   4/ 339]                     blk.0.attn_k.bias - [  512,     1,
1,     1], type =    f32, size =    0.002 MB
[   5/ 339]                   blk.0.attn_k.weight - [ 3584,   512,
1,     1], type =    f16, converting to q4_K .. size =     3.50 MiB ->
0.98 MiB
[   6/ 339]                blk.0.attn_norm.weight - [ 3584,     1,
1,     1], type =    f32, size =    0.014 MB
[   7/ 339]              blk.0.attn_output.weight - [ 3584,  3584,
1,     1], type =    f16, converting to q4_K .. size =    24.50 MiB ->
6.89 MiB
[   8/ 339]                     blk.0.attn_q.bias - [ 3584,     1,
1,     1], type =    f32, size =    0.014 MB
[   9/ 339]                   blk.0.attn_q.weight - [ 3584,  3584,
1,     1], type =    f16, converting to q4_K .. size =    24.50 MiB ->
6.89 MiB
[  10/ 339]                     blk.0.attn_v.bias - [  512,     1,
1,     1], type =    f32, size =    0.002 MB
[  11/ 339]                   blk.0.attn_v.weight - [ 3584,   512,
1,     1], type =    f16, converting to q6_K .. size =     3.50 MiB ->
1.44 MiB
[  12/ 339]                 blk.0.ffn_down.weight - [18944,  3584,
1,     1], type =    f16, converting to q6_K .. size =   129.50 MiB ->
53.12 MiB
[  13/ 339]                 blk.0.ffn_gate.weight - [ 3584, 18944,
1,     1], type =    f16, converting to q4_K .. size =   129.50 MiB ->
36.42 MiB
[  14/ 339]                 blk.0.ffn_norm.weight - [ 3584,     1,
1,     1], type =    f32, size =    0.014 MB
[  15/ 339]                   blk.0.ffn_up.weight - [ 3584, 18944,
1,     1], type =    f16, converting to q4_K .. size =   129.50 MiB ->
36.42 MiB
[  16/ 339]                     blk.1.attn_k.bias - [  512,     1,
1,     1], type =    f32, size =    0.002 MB
[  17/ 339]                   blk.1.attn_k.weight - [ 3584,   512,
1,     1], type =    f16, converting to q4_K .. size =     3.50 MiB ->
0.98 MiB
[  18/ 339]                blk.1.attn_norm.weight - [ 3584,     1,
1,     1], type =    f32, size =    0.014 MB
```

```
[  19/ 339]             blk.1.attn_output.weight - [ 3584,  3584,
1,     1], type =    f16, converting to q4_K .. size =    24.50 MiB ->
6.89 MiB
[  20/ 339]                   blk.1.attn_q.bias - [ 3584,     1,
1,     1], type =    f32, size =    0.014 MB
[  21/ 339]                blk.1.attn_q.weight - [ 3584,  3584,
1,     1], type =    f16, converting to q4_K .. size =    24.50 MiB ->
6.89 MiB
[  22/ 339]                   blk.1.attn_v.bias - [  512,     1,
1,     1], type =    f32, size =    0.002 MB
[  23/ 339]                blk.1.attn_v.weight - [ 3584,   512,
1,     1], type =    f16, converting to q6_K .. size =     3.50 MiB ->
1.44 MiB
[  24/ 339]                 blk.1.ffn_down.weight - [18944,  3584,
1,     1], type =    f16, converting to q6_K .. size =   129.50 MiB ->
53.12 MiB
[  25/ 339]                 blk.1.ffn_gate.weight - [ 3584, 18944,
1,     1], type =    f16, converting to q4_K .. size =   129.50 MiB ->
36.42 MiB
[  26/ 339]                 blk.1.ffn_norm.weight - [ 3584,     1,
1,     1], type =    f32, size =    0.014 MB
[  27/ 339]                   blk.1.ffn_up.weight - [ 3584, 18944,
1,     1], type =    f16, converting to q4_K .. size =   129.50 MiB ->
36.42 MiB
[  28/ 339]                   blk.2.attn_k.bias - [  512,     1,
1,     1], type =    f32, size =    0.002 MB
[  29/ 339]                blk.2.attn_k.weight - [ 3584,   512,
1,     1], type =    f16, converting to q4_K .. size =     3.50 MiB ->
0.98 MiB
[  30/ 339]               blk.2.attn_norm.weight - [ 3584,     1,
1,     1], type =    f32, size =    0.014 MB
[  31/ 339]             blk.2.attn_output.weight - [ 3584,  3584,
1,     1], type =    f16, converting to q4_K .. size =    24.50 MiB ->
6.89 MiB
[  32/ 339]                   blk.2.attn_q.bias - [ 3584,     1,
1,     1], type =    f32, size =    0.014 MB
[  33/ 339]                blk.2.attn_q.weight - [ 3584,  3584,
1,     1], type =    f16, converting to q4_K .. size =    24.50 MiB ->
6.89 MiB
[  34/ 339]                   blk.2.attn_v.bias - [  512,     1,
1,     1], type =    f32, size =    0.002 MB
[  35/ 339]                blk.2.attn_v.weight - [ 3584,   512,
1,     1], type =    f16, converting to q6_K .. size =     3.50 MiB ->
1.44 MiB
[  36/ 339]                 blk.2.ffn_down.weight - [18944,  3584,
1,     1], type =    f16, converting to q6_K .. size =   129.50 MiB ->
53.12 MiB
[  37/ 339]                 blk.2.ffn_gate.weight - [ 3584, 18944,
1,     1], type =    f16, converting to q4_K .. size =   129.50 MiB ->
```

```
36.42 MiB
[  38/ 339]                 blk.2.ffn_norm.weight - [ 3584,     1,
1,     1], type =    f32, size =    0.014 MB
[  39/ 339]                   blk.2.ffn_up.weight - [ 3584, 18944,
1,     1], type =    f16, converting to q4_K .. size =   129.50 MiB ->
36.42 MiB
[  40/ 339]                   blk.3.attn_k.bias - [  512,     1,
1,     1], type =    f32, size =    0.002 MB
[  41/ 339]                   blk.3.attn_k.weight - [ 3584,   512,
1,     1], type =    f16, converting to q4_K .. size =     3.50 MiB ->
0.98 MiB
[  42/ 339]                blk.3.attn_norm.weight - [ 3584,     1,
1,     1], type =    f32, size =    0.014 MB
[  43/ 339]              blk.3.attn_output.weight - [ 3584,  3584,
1,     1], type =    f16, converting to q4_K .. size =    24.50 MiB ->
6.89 MiB
[  44/ 339]                   blk.3.attn_q.bias - [ 3584,     1,
1,     1], type =    f32, size =    0.014 MB
[  45/ 339]                   blk.3.attn_q.weight - [ 3584,  3584,
1,     1], type =    f16, converting to q4_K .. size =    24.50 MiB ->
6.89 MiB
[  46/ 339]                   blk.3.attn_v.bias - [  512,     1,
1,     1], type =    f32, size =    0.002 MB
[  47/ 339]                   blk.3.attn_v.weight - [ 3584,   512,
1,     1], type =    f16, converting to q4_K .. size =     3.50 MiB ->
0.98 MiB
[  48/ 339]                 blk.3.ffn_down.weight - [18944,  3584,
1,     1], type =    f16, converting to q4_K .. size =   129.50 MiB ->
36.42 MiB
[  49/ 339]                 blk.3.ffn_gate.weight - [ 3584, 18944,
1,     1], type =    f16, converting to q4_K .. size =   129.50 MiB ->
36.42 MiB
[  50/ 339]                 blk.3.ffn_norm.weight - [ 3584,     1,
1,     1], type =    f32, size =    0.014 MB
[  51/ 339]                   blk.3.ffn_up.weight - [ 3584, 18944,
1,     1], type =    f16, converting to q4_K .. size =   129.50 MiB ->
36.42 MiB
[  52/ 339]                   blk.4.attn_k.bias - [  512,     1,
1,     1], type =    f32, size =    0.002 MB
[  53/ 339]                   blk.4.attn_k.weight - [ 3584,   512,
1,     1], type =    f16, converting to q4_K .. size =     3.50 MiB ->
0.98 MiB
[  54/ 339]                blk.4.attn_norm.weight - [ 3584,     1,
1,     1], type =    f32, size =    0.014 MB
[  55/ 339]              blk.4.attn_output.weight - [ 3584,  3584,
1,     1], type =    f16, converting to q4_K .. size =    24.50 MiB ->
6.89 MiB
[  56/ 339]                   blk.4.attn_q.bias - [ 3584,     1,
1,     1], type =    f32, size =    0.014 MB
[  57/ 339]                   blk.4.attn_q.weight - [ 3584,  3584,
```

```
1,     1], type =    f16, converting to q4_K .. size =    24.50 MiB ->
6.89 MiB
[  58/ 339]                    blk.4.attn_v.bias - [  512,     1,
1,     1], type =    f32, size =    0.002 MB
[  59/ 339]                   blk.4.attn_v.weight - [ 3584,   512,
1,     1], type =    f16, converting to q4_K .. size =     3.50 MiB ->
0.98 MiB
[  60/ 339]                 blk.4.ffn_down.weight - [18944,  3584,
1,     1], type =    f16, converting to q4_K .. size =   129.50 MiB ->
36.42 MiB
[  61/ 339]                 blk.4.ffn_gate.weight - [ 3584, 18944,
1,     1], type =    f16, converting to q4_K .. size =   129.50 MiB ->
36.42 MiB
[  62/ 339]                 blk.4.ffn_norm.weight - [ 3584,     1,
1,     1], type =    f32, size =    0.014 MB
[  63/ 339]                   blk.4.ffn_up.weight - [ 3584, 18944,
1,     1], type =    f16, converting to q4_K .. size =   129.50 MiB ->
36.42 MiB
[  64/ 339]                    blk.5.attn_k.bias - [  512,     1,
1,     1], type =    f32, size =    0.002 MB
[  65/ 339]                   blk.5.attn_k.weight - [ 3584,   512,
1,     1], type =    f16, converting to q4_K .. size =     3.50 MiB ->
0.98 MiB
[  66/ 339]                blk.5.attn_norm.weight - [ 3584,     1,
1,     1], type =    f32, size =    0.014 MB
[  67/ 339]              blk.5.attn_output.weight - [ 3584,  3584,
1,     1], type =    f16, converting to q4_K .. size =    24.50 MiB ->
6.89 MiB
[  68/ 339]                    blk.5.attn_q.bias - [ 3584,     1,
1,     1], type =    f32, size =    0.014 MB
[  69/ 339]                   blk.5.attn_q.weight - [ 3584,  3584,
1,     1], type =    f16, converting to q4_K .. size =    24.50 MiB ->
6.89 MiB
[  70/ 339]                    blk.5.attn_v.bias - [  512,     1,
1,     1], type =    f32, size =    0.002 MB
[  71/ 339]                   blk.5.attn_v.weight - [ 3584,   512,
1,     1], type =    f16, converting to q6_K .. size =     3.50 MiB ->
1.44 MiB
[  72/ 339]                 blk.5.ffn_down.weight - [18944,  3584,
1,     1], type =    f16, converting to q6_K .. size =   129.50 MiB ->
53.12 MiB
[  73/ 339]                 blk.5.ffn_gate.weight - [ 3584, 18944,
1,     1], type =    f16, converting to q4_K .. size =   129.50 MiB ->
36.42 MiB
[  74/ 339]                 blk.5.ffn_norm.weight - [ 3584,     1,
1,     1], type =    f32, size =    0.014 MB
[  75/ 339]                   blk.5.ffn_up.weight - [ 3584, 18944,
1,     1], type =    f16, converting to q4_K .. size =   129.50 MiB ->
36.42 MiB
```

```
[  76/ 339]                    blk.6.attn_k.bias - [  512,     1,
1,     1], type =    f32, size =    0.002 MB
[  77/ 339]                      blk.6.attn_k.weight - [ 3584,   512,
1,     1], type =    f16, converting to q4_K .. size =     3.50 MiB ->
0.98 MiB
[  78/ 339]                 blk.6.attn_norm.weight - [ 3584,     1,
1,     1], type =    f32, size =    0.014 MB
[  79/ 339]               blk.6.attn_output.weight - [ 3584,  3584,
1,     1], type =    f16, converting to q4_K .. size =    24.50 MiB ->
6.89 MiB
[  80/ 339]                    blk.6.attn_q.bias - [ 3584,     1,
1,     1], type =    f32, size =    0.014 MB
[  81/ 339]                      blk.6.attn_q.weight - [ 3584,  3584,
1,     1], type =    f16, converting to q4_K .. size =    24.50 MiB ->
6.89 MiB
[  82/ 339]                    blk.6.attn_v.bias - [  512,     1,
1,     1], type =    f32, size =    0.002 MB
[  83/ 339]                      blk.6.attn_v.weight - [ 3584,   512,
1,     1], type =    f16, converting to q4_K .. size =     3.50 MiB ->
0.98 MiB
[  84/ 339]                  blk.6.ffn_down.weight - [18944,  3584,
1,     1], type =    f16, converting to q4_K .. size =   129.50 MiB ->
36.42 MiB
[  85/ 339]                  blk.6.ffn_gate.weight - [ 3584, 18944,
1,     1], type =    f16, converting to q4_K .. size =   129.50 MiB ->
36.42 MiB
[  86/ 339]                  blk.6.ffn_norm.weight - [ 3584,     1,
1,     1], type =    f32, size =    0.014 MB
[  87/ 339]                    blk.6.ffn_up.weight - [ 3584, 18944,
1,     1], type =    f16, converting to q4_K .. size =   129.50 MiB ->
36.42 MiB
[  88/ 339]                    blk.7.attn_k.bias - [  512,     1,
1,     1], type =    f32, size =    0.002 MB
[  89/ 339]                      blk.7.attn_k.weight - [ 3584,   512,
1,     1], type =    f16, converting to q4_K .. size =     3.50 MiB ->
0.98 MiB
[  90/ 339]                 blk.7.attn_norm.weight - [ 3584,     1,
1,     1], type =    f32, size =    0.014 MB
[  91/ 339]               blk.7.attn_output.weight - [ 3584,  3584,
1,     1], type =    f16, converting to q4_K .. size =    24.50 MiB ->
6.89 MiB
[  92/ 339]                    blk.7.attn_q.bias - [ 3584,     1,
1,     1], type =    f32, size =    0.014 MB
[  93/ 339]                      blk.7.attn_q.weight - [ 3584,  3584,
1,     1], type =    f16, converting to q4_K .. size =    24.50 MiB ->
6.89 MiB
[  94/ 339]                    blk.7.attn_v.bias - [  512,     1,
1,     1], type =    f32, size =    0.002 MB
[  95/ 339]                      blk.7.attn_v.weight - [ 3584,   512,
```

```
1,     1], type =    f16, converting to q4_K .. size =     3.50 MiB ->
0.98 MiB
[  96/ 339]                 blk.7.ffn_down.weight - [18944,  3584,
1,     1], type =    f16, converting to q4_K .. size =   129.50 MiB ->
36.42 MiB
[  97/ 339]                 blk.7.ffn_gate.weight - [ 3584, 18944,
1,     1], type =    f16, converting to q4_K .. size =   129.50 MiB ->
36.42 MiB
[  98/ 339]                 blk.7.ffn_norm.weight - [ 3584,     1,
1,     1], type =    f32, size =    0.014 MB
[  99/ 339]                   blk.7.ffn_up.weight - [ 3584, 18944,
1,     1], type =    f16, converting to q4_K .. size =   129.50 MiB ->
36.42 MiB
[ 100/ 339]                   blk.8.attn_k.bias - [  512,     1,
1,     1], type =    f32, size =    0.002 MB
[ 101/ 339]                   blk.8.attn_k.weight - [ 3584,   512,
1,     1], type =    f16, converting to q4_K .. size =     3.50 MiB ->
0.98 MiB
[ 102/ 339]                 blk.8.attn_norm.weight - [ 3584,     1,
1,     1], type =    f32, size =    0.014 MB
[ 103/ 339]              blk.8.attn_output.weight - [ 3584,  3584,
1,     1], type =    f16, converting to q4_K .. size =    24.50 MiB ->
6.89 MiB
[ 104/ 339]                   blk.8.attn_q.bias - [ 3584,     1,
1,     1], type =    f32, size =    0.014 MB
[ 105/ 339]                   blk.8.attn_q.weight - [ 3584,  3584,
1,     1], type =    f16, converting to q4_K .. size =    24.50 MiB ->
6.89 MiB
[ 106/ 339]                   blk.8.attn_v.bias - [  512,     1,
1,     1], type =    f32, size =    0.002 MB
[ 107/ 339]                   blk.8.attn_v.weight - [ 3584,   512,
1,     1], type =    f16, converting to q6_K .. size =     3.50 MiB ->
1.44 MiB
[ 108/ 339]                 blk.8.ffn_down.weight - [18944,  3584,
1,     1], type =    f16, converting to q6_K .. size =   129.50 MiB ->
53.12 MiB
[ 109/ 339]                 blk.8.ffn_gate.weight - [ 3584, 18944,
1,     1], type =    f16, converting to q4_K .. size =   129.50 MiB ->
36.42 MiB
[ 110/ 339]                 blk.8.ffn_norm.weight - [ 3584,     1,
1,     1], type =    f32, size =    0.014 MB
[ 111/ 339]                   blk.8.ffn_up.weight - [ 3584, 18944,
1,     1], type =    f16, converting to q4_K .. size =   129.50 MiB ->
36.42 MiB
[ 112/ 339]                   blk.9.attn_k.bias - [  512,     1,
1,     1], type =    f32, size =    0.002 MB
[ 113/ 339]                   blk.9.attn_k.weight - [ 3584,   512,
1,     1], type =    f16, converting to q4_K .. size =     3.50 MiB ->
0.98 MiB
```

```
[ 114/ 339]                blk.9.attn_norm.weight - [ 3584,     1,
1,     1], type =    f32, size =    0.014 MB
[ 115/ 339]             blk.9.attn_output.weight - [ 3584,  3584,
1,     1], type =    f16, converting to q4_K .. size =    24.50 MiB ->
6.89 MiB
[ 116/ 339]                  blk.9.attn_q.bias - [ 3584,     1,
1,     1], type =    f32, size =    0.014 MB
[ 117/ 339]                 blk.9.attn_q.weight - [ 3584,  3584,
1,     1], type =    f16, converting to q4_K .. size =    24.50 MiB ->
6.89 MiB
[ 118/ 339]                  blk.9.attn_v.bias - [  512,     1,
1,     1], type =    f32, size =    0.002 MB
[ 119/ 339]                 blk.9.attn_v.weight - [ 3584,   512,
1,     1], type =    f16, converting to q4_K .. size =     3.50 MiB ->
0.98 MiB
[ 120/ 339]                blk.9.ffn_down.weight - [18944,  3584,
1,     1], type =    f16, converting to q4_K .. size =   129.50 MiB ->
36.42 MiB
[ 121/ 339]                blk.9.ffn_gate.weight - [ 3584, 18944,
1,     1], type =    f16, converting to q4_K .. size =   129.50 MiB ->
36.42 MiB
[ 122/ 339]                blk.9.ffn_norm.weight - [ 3584,     1,
1,     1], type =    f32, size =    0.014 MB
[ 123/ 339]                  blk.9.ffn_up.weight - [ 3584, 18944,
1,     1], type =    f16, converting to q4_K .. size =   129.50 MiB ->
36.42 MiB
[ 124/ 339]                 blk.10.attn_k.bias - [  512,     1,
1,     1], type =    f32, size =    0.002 MB
[ 125/ 339]                blk.10.attn_k.weight - [ 3584,   512,
1,     1], type =    f16, converting to q4_K .. size =     3.50 MiB ->
0.98 MiB
[ 126/ 339]               blk.10.attn_norm.weight - [ 3584,     1,
1,     1], type =    f32, size =    0.014 MB
[ 127/ 339]             blk.10.attn_output.weight - [ 3584,  3584,
1,     1], type =    f16, converting to q4_K .. size =    24.50 MiB ->
6.89 MiB
[ 128/ 339]                 blk.10.attn_q.bias - [ 3584,     1,
1,     1], type =    f32, size =    0.014 MB
[ 129/ 339]                blk.10.attn_q.weight - [ 3584,  3584,
1,     1], type =    f16, converting to q4_K .. size =    24.50 MiB ->
6.89 MiB
[ 130/ 339]                 blk.10.attn_v.bias - [  512,     1,
1,     1], type =    f32, size =    0.002 MB
[ 131/ 339]                blk.10.attn_v.weight - [ 3584,   512,
1,     1], type =    f16, converting to q4_K .. size =     3.50 MiB ->
0.98 MiB
[ 132/ 339]                blk.10.ffn_down.weight - [18944,  3584,
1,     1], type =    f16, converting to q4_K .. size =   129.50 MiB ->
36.42 MiB
```

```
[ 133/ 339]                 blk.10.ffn_gate.weight - [ 3584, 18944,
1,     1], type =    f16, converting to q4_K .. size =   129.50 MiB ->
36.42 MiB
[ 134/ 339]                 blk.10.ffn_norm.weight - [ 3584,     1,
1,     1], type =    f32, size =    0.014 MB
[ 135/ 339]                   blk.10.ffn_up.weight - [ 3584, 18944,
1,     1], type =    f16, converting to q4_K .. size =   129.50 MiB ->
36.42 MiB
[ 136/ 339]                    blk.11.attn_k.bias - [  512,     1,
1,     1], type =    f32, size =    0.002 MB
[ 137/ 339]                   blk.11.attn_k.weight - [ 3584,   512,
1,     1], type =    f16, converting to q4_K .. size =     3.50 MiB ->
0.98 MiB
[ 138/ 339]                blk.11.attn_norm.weight - [ 3584,     1,
1,     1], type =    f32, size =    0.014 MB
[ 139/ 339]              blk.11.attn_output.weight - [ 3584,  3584,
1,     1], type =    f16, converting to q4_K .. size =    24.50 MiB ->
6.89 MiB
[ 140/ 339]                    blk.11.attn_q.bias - [ 3584,     1,
1,     1], type =    f32, size =    0.014 MB
[ 141/ 339]                   blk.11.attn_q.weight - [ 3584,  3584,
1,     1], type =    f16, converting to q4_K .. size =    24.50 MiB ->
6.89 MiB
[ 142/ 339]                    blk.11.attn_v.bias - [  512,     1,
1,     1], type =    f32, size =    0.002 MB
[ 143/ 339]                   blk.11.attn_v.weight - [ 3584,   512,
1,     1], type =    f16, converting to q6_K .. size =     3.50 MiB ->
1.44 MiB
[ 144/ 339]                 blk.11.ffn_down.weight - [18944,  3584,
1,     1], type =    f16, converting to q6_K .. size =   129.50 MiB ->
53.12 MiB
[ 145/ 339]                 blk.11.ffn_gate.weight - [ 3584, 18944,
1,     1], type =    f16, converting to q4_K .. size =   129.50 MiB ->
36.42 MiB
[ 146/ 339]                 blk.11.ffn_norm.weight - [ 3584,     1,
1,     1], type =    f32, size =    0.014 MB
[ 147/ 339]                   blk.11.ffn_up.weight - [ 3584, 18944,
1,     1], type =    f16, converting to q4_K .. size =   129.50 MiB ->
36.42 MiB
[ 148/ 339]                    blk.12.attn_k.bias - [  512,     1,
1,     1], type =    f32, size =    0.002 MB
[ 149/ 339]                   blk.12.attn_k.weight - [ 3584,   512,
1,     1], type =    f16, converting to q4_K .. size =     3.50 MiB ->
0.98 MiB
[ 150/ 339]                blk.12.attn_norm.weight - [ 3584,     1,
1,     1], type =    f32, size =    0.014 MB
[ 151/ 339]              blk.12.attn_output.weight - [ 3584,  3584,
1,     1], type =    f16, converting to q4_K .. size =    24.50 MiB ->
6.89 MiB
```

```
[ 152/ 339]                    blk.12.attn_q.bias - [ 3584,     1,
1,     1], type =    f32, size =    0.014 MB
[ 153/ 339]                  blk.12.attn_q.weight - [ 3584,  3584,
1,     1], type =    f16, converting to q4_K .. size =    24.50 MiB ->
6.89 MiB
[ 154/ 339]                    blk.12.attn_v.bias - [  512,     1,
1,     1], type =    f32, size =    0.002 MB
[ 155/ 339]                  blk.12.attn_v.weight - [ 3584,   512,
1,     1], type =    f16, converting to q4_K .. size =     3.50 MiB ->
0.98 MiB
[ 156/ 339]                blk.12.ffn_down.weight - [18944,  3584,
1,     1], type =    f16, converting to q4_K .. size =   129.50 MiB ->
36.42 MiB
[ 157/ 339]                blk.12.ffn_gate.weight - [ 3584, 18944,
1,     1], type =    f16, converting to q4_K .. size =   129.50 MiB ->
36.42 MiB
[ 158/ 339]                blk.12.ffn_norm.weight - [ 3584,     1,
1,     1], type =    f32, size =    0.014 MB
[ 159/ 339]                  blk.12.ffn_up.weight - [ 3584, 18944,
1,     1], type =    f16, converting to q4_K .. size =   129.50 MiB ->
36.42 MiB
[ 160/ 339]                    blk.13.attn_k.bias - [  512,     1,
1,     1], type =    f32, size =    0.002 MB
[ 161/ 339]                  blk.13.attn_k.weight - [ 3584,   512,
1,     1], type =    f16, converting to q4_K .. size =     3.50 MiB ->
0.98 MiB
[ 162/ 339]               blk.13.attn_norm.weight - [ 3584,     1,
1,     1], type =    f32, size =    0.014 MB
[ 163/ 339]             blk.13.attn_output.weight - [ 3584,  3584,
1,     1], type =    f16, converting to q4_K .. size =    24.50 MiB ->
6.89 MiB
[ 164/ 339]                    blk.13.attn_q.bias - [ 3584,     1,
1,     1], type =    f32, size =    0.014 MB
[ 165/ 339]                  blk.13.attn_q.weight - [ 3584,  3584,
1,     1], type =    f16, converting to q4_K .. size =    24.50 MiB ->
6.89 MiB
[ 166/ 339]                    blk.13.attn_v.bias - [  512,     1,
1,     1], type =    f32, size =    0.002 MB
[ 167/ 339]                  blk.13.attn_v.weight - [ 3584,   512,
1,     1], type =    f16, converting to q4_K .. size =     3.50 MiB ->
0.98 MiB
[ 168/ 339]                blk.13.ffn_down.weight - [18944,  3584,
1,     1], type =    f16, converting to q4_K .. size =   129.50 MiB ->
36.42 MiB
[ 169/ 339]                blk.13.ffn_gate.weight - [ 3584, 18944,
1,     1], type =    f16, converting to q4_K .. size =   129.50 MiB ->
36.42 MiB
[ 170/ 339]                blk.13.ffn_norm.weight - [ 3584,     1,
1,     1], type =    f32, size =    0.014 MB
```

```
[ 171/ 339]                 blk.13.ffn_up.weight - [ 3584, 18944,
1,     1], type =    f16, converting to q4_K .. size =   129.50 MiB ->
36.42 MiB
[ 172/ 339]                   blk.14.attn_k.bias - [  512,     1,
1,     1], type =    f32, size =    0.002 MB
[ 173/ 339]                 blk.14.attn_k.weight - [ 3584,   512,
1,     1], type =    f16, converting to q4_K .. size =     3.50 MiB ->
0.98 MiB
[ 174/ 339]               blk.14.attn_norm.weight - [ 3584,     1,
1,     1], type =    f32, size =    0.014 MB
[ 175/ 339]             blk.14.attn_output.weight - [ 3584,  3584,
1,     1], type =    f16, converting to q4_K .. size =    24.50 MiB ->
6.89 MiB
[ 176/ 339]                   blk.14.attn_q.bias - [ 3584,     1,
1,     1], type =    f32, size =    0.014 MB
[ 177/ 339]                 blk.14.attn_q.weight - [ 3584,  3584,
1,     1], type =    f16, converting to q4_K .. size =    24.50 MiB ->
6.89 MiB
[ 178/ 339]                   blk.14.attn_v.bias - [  512,     1,
1,     1], type =    f32, size =    0.002 MB
[ 179/ 339]                 blk.14.attn_v.weight - [ 3584,   512,
1,     1], type =    f16, converting to q6_K .. size =     3.50 MiB ->
1.44 MiB
[ 180/ 339]               blk.14.ffn_down.weight - [18944,  3584,
1,     1], type =    f16, converting to q6_K .. size =   129.50 MiB ->
53.12 MiB
[ 181/ 339]               blk.14.ffn_gate.weight - [ 3584, 18944,
1,     1], type =    f16, converting to q4_K .. size =   129.50 MiB ->
36.42 MiB
[ 182/ 339]               blk.14.ffn_norm.weight - [ 3584,     1,
1,     1], type =    f32, size =    0.014 MB
[ 183/ 339]                 blk.14.ffn_up.weight - [ 3584, 18944,
1,     1], type =    f16, converting to q4_K .. size =   129.50 MiB ->
36.42 MiB
[ 184/ 339]                   blk.15.attn_k.bias - [  512,     1,
1,     1], type =    f32, size =    0.002 MB
[ 185/ 339]                 blk.15.attn_k.weight - [ 3584,   512,
1,     1], type =    f16, converting to q4_K .. size =     3.50 MiB ->
0.98 MiB
[ 186/ 339]               blk.15.attn_norm.weight - [ 3584,     1,
1,     1], type =    f32, size =    0.014 MB
[ 187/ 339]             blk.15.attn_output.weight - [ 3584,  3584,
1,     1], type =    f16, converting to q4_K .. size =    24.50 MiB ->
6.89 MiB
[ 188/ 339]                   blk.15.attn_q.bias - [ 3584,     1,
1,     1], type =    f32, size =    0.014 MB
[ 189/ 339]                 blk.15.attn_q.weight - [ 3584,  3584,
1,     1], type =    f16, converting to q4_K .. size =    24.50 MiB ->
6.89 MiB
```

```
[ 190/ 339]                    blk.15.attn_v.bias - [  512,     1,
1,     1], type =    f32, size =    0.002 MB
[ 191/ 339]                  blk.15.attn_v.weight - [ 3584,   512,
1,     1], type =    f16, converting to q4_K .. size =     3.50 MiB ->
0.98 MiB
[ 192/ 339]                blk.15.ffn_down.weight - [18944,  3584,
1,     1], type =    f16, converting to q4_K .. size =   129.50 MiB ->
36.42 MiB
[ 193/ 339]                blk.15.ffn_gate.weight - [ 3584, 18944,
1,     1], type =    f16, converting to q4_K .. size =   129.50 MiB ->
36.42 MiB
[ 194/ 339]                blk.15.ffn_norm.weight - [ 3584,     1,
1,     1], type =    f32, size =    0.014 MB
[ 195/ 339]                  blk.15.ffn_up.weight - [ 3584, 18944,
1,     1], type =    f16, converting to q4_K .. size =   129.50 MiB ->
36.42 MiB
[ 196/ 339]                    blk.16.attn_k.bias - [  512,     1,
1,     1], type =    f32, size =    0.002 MB
[ 197/ 339]                  blk.16.attn_k.weight - [ 3584,   512,
1,     1], type =    f16, converting to q4_K .. size =     3.50 MiB ->
0.98 MiB
[ 198/ 339]                blk.16.attn_norm.weight - [ 3584,     1,
1,     1], type =    f32, size =    0.014 MB
[ 199/ 339]              blk.16.attn_output.weight - [ 3584,  3584,
1,     1], type =    f16, converting to q4_K .. size =    24.50 MiB ->
6.89 MiB
[ 200/ 339]                    blk.16.attn_q.bias - [ 3584,     1,
1,     1], type =    f32, size =    0.014 MB
[ 201/ 339]                  blk.16.attn_q.weight - [ 3584,  3584,
1,     1], type =    f16, converting to q4_K .. size =    24.50 MiB ->
6.89 MiB
[ 202/ 339]                    blk.16.attn_v.bias - [  512,     1,
1,     1], type =    f32, size =    0.002 MB
[ 203/ 339]                  blk.16.attn_v.weight - [ 3584,   512,
1,     1], type =    f16, converting to q4_K .. size =     3.50 MiB ->
0.98 MiB
[ 204/ 339]                blk.16.ffn_down.weight - [18944,  3584,
1,     1], type =    f16, converting to q4_K .. size =   129.50 MiB ->
36.42 MiB
[ 205/ 339]                blk.16.ffn_gate.weight - [ 3584, 18944,
1,     1], type =    f16, converting to q4_K .. size =   129.50 MiB ->
36.42 MiB
[ 206/ 339]                blk.16.ffn_norm.weight - [ 3584,     1,
1,     1], type =    f32, size =    0.014 MB
[ 207/ 339]                  blk.16.ffn_up.weight - [ 3584, 18944,
1,     1], type =    f16, converting to q4_K .. size =   129.50 MiB ->
36.42 MiB
[ 208/ 339]                    blk.17.attn_k.bias - [  512,     1,
1,     1], type =    f32, size =    0.002 MB
```

```
[ 209/ 339]                 blk.17.attn_k.weight - [ 3584,   512,
1,     1], type =    f16, converting to q4_K .. size =     3.50 MiB ->
0.98 MiB
[ 210/ 339]              blk.17.attn_norm.weight - [ 3584,     1,
1,     1], type =    f32, size =    0.014 MB
[ 211/ 339]           blk.17.attn_output.weight - [ 3584,  3584,
1,     1], type =    f16, converting to q4_K .. size =    24.50 MiB ->
6.89 MiB
[ 212/ 339]                   blk.17.attn_q.bias - [ 3584,     1,
1,     1], type =    f32, size =    0.014 MB
[ 213/ 339]                 blk.17.attn_q.weight - [ 3584,  3584,
1,     1], type =    f16, converting to q4_K .. size =    24.50 MiB ->
6.89 MiB
[ 214/ 339]                   blk.17.attn_v.bias - [  512,     1,
1,     1], type =    f32, size =    0.002 MB
[ 215/ 339]                 blk.17.attn_v.weight - [ 3584,   512,
1,     1], type =    f16, converting to q6_K .. size =     3.50 MiB ->
1.44 MiB
[ 216/ 339]               blk.17.ffn_down.weight - [18944,  3584,
1,     1], type =    f16, converting to q6_K .. size =   129.50 MiB ->
53.12 MiB
[ 217/ 339]               blk.17.ffn_gate.weight - [ 3584, 18944,
1,     1], type =    f16, converting to q4_K .. size =   129.50 MiB ->
36.42 MiB
[ 218/ 339]               blk.17.ffn_norm.weight - [ 3584,     1,
1,     1], type =    f32, size =    0.014 MB
[ 219/ 339]                 blk.17.ffn_up.weight - [ 3584, 18944,
1,     1], type =    f16, converting to q4_K .. size =   129.50 MiB ->
36.42 MiB
[ 220/ 339]                   blk.18.attn_k.bias - [  512,     1,
1,     1], type =    f32, size =    0.002 MB
[ 221/ 339]                 blk.18.attn_k.weight - [ 3584,   512,
1,     1], type =    f16, converting to q4_K .. size =     3.50 MiB ->
0.98 MiB
[ 222/ 339]              blk.18.attn_norm.weight - [ 3584,     1,
1,     1], type =    f32, size =    0.014 MB
[ 223/ 339]           blk.18.attn_output.weight - [ 3584,  3584,
1,     1], type =    f16, converting to q4_K .. size =    24.50 MiB ->
6.89 MiB
[ 224/ 339]                   blk.18.attn_q.bias - [ 3584,     1,
1,     1], type =    f32, size =    0.014 MB
[ 225/ 339]                 blk.18.attn_q.weight - [ 3584,  3584,
1,     1], type =    f16, converting to q4_K .. size =    24.50 MiB ->
6.89 MiB
[ 226/ 339]                   blk.18.attn_v.bias - [  512,     1,
1,     1], type =    f32, size =    0.002 MB
[ 227/ 339]                 blk.18.attn_v.weight - [ 3584,   512,
1,     1], type =    f16, converting to q4_K .. size =     3.50 MiB ->
0.98 MiB
```

```
[ 228/ 339]                 blk.18.ffn_down.weight - [18944,  3584,
1,     1], type =    f16, converting to q4_K .. size =   129.50 MiB ->
36.42 MiB
[ 229/ 339]                 blk.18.ffn_gate.weight - [ 3584, 18944,
1,     1], type =    f16, converting to q4_K .. size =   129.50 MiB ->
36.42 MiB
[ 230/ 339]                 blk.18.ffn_norm.weight - [ 3584,     1,
1,     1], type =    f32, size =    0.014 MB
[ 231/ 339]                   blk.18.ffn_up.weight - [ 3584, 18944,
1,     1], type =    f16, converting to q4_K .. size =   129.50 MiB ->
36.42 MiB
[ 232/ 339]                     blk.19.attn_k.bias - [  512,     1,
1,     1], type =    f32, size =    0.002 MB
[ 233/ 339]                   blk.19.attn_k.weight - [ 3584,   512,
1,     1], type =    f16, converting to q4_K .. size =     3.50 MiB ->
0.98 MiB
[ 234/ 339]                blk.19.attn_norm.weight - [ 3584,     1,
1,     1], type =    f32, size =    0.014 MB
[ 235/ 339]              blk.19.attn_output.weight - [ 3584,  3584,
1,     1], type =    f16, converting to q4_K .. size =    24.50 MiB ->
6.89 MiB
[ 236/ 339]                     blk.19.attn_q.bias - [ 3584,     1,
1,     1], type =    f32, size =    0.014 MB
[ 237/ 339]                   blk.19.attn_q.weight - [ 3584,  3584,
1,     1], type =    f16, converting to q4_K .. size =    24.50 MiB ->
6.89 MiB
[ 238/ 339]                     blk.19.attn_v.bias - [  512,     1,
1,     1], type =    f32, size =    0.002 MB
[ 239/ 339]                   blk.19.attn_v.weight - [ 3584,   512,
1,     1], type =    f16, converting to q4_K .. size =     3.50 MiB ->
0.98 MiB
[ 240/ 339]                 blk.19.ffn_down.weight - [18944,  3584,
1,     1], type =    f16, converting to q4_K .. size =   129.50 MiB ->
36.42 MiB
[ 241/ 339]                 blk.19.ffn_gate.weight - [ 3584, 18944,
1,     1], type =    f16, converting to q4_K .. size =   129.50 MiB ->
36.42 MiB
[ 242/ 339]                 blk.19.ffn_norm.weight - [ 3584,     1,
1,     1], type =    f32, size =    0.014 MB
[ 243/ 339]                   blk.19.ffn_up.weight - [ 3584, 18944,
1,     1], type =    f16, converting to q4_K .. size =   129.50 MiB ->
36.42 MiB
[ 244/ 339]                     blk.20.attn_k.bias - [  512,     1,
1,     1], type =    f32, size =    0.002 MB
[ 245/ 339]                   blk.20.attn_k.weight - [ 3584,   512,
1,     1], type =    f16, converting to q4_K .. size =     3.50 MiB ->
0.98 MiB
[ 246/ 339]                blk.20.attn_norm.weight - [ 3584,     1,
1,     1], type =    f32, size =    0.014 MB
```

```
[ 247/ 339]            blk.20.attn_output.weight - [ 3584,  3584,
1,     1], type =    f16, converting to q4_K .. size =    24.50 MiB ->
6.89 MiB
[ 248/ 339]                   blk.20.attn_q.bias - [ 3584,     1,
1,     1], type =    f32, size =    0.014 MB
[ 249/ 339]                 blk.20.attn_q.weight - [ 3584,  3584,
1,     1], type =    f16, converting to q4_K .. size =    24.50 MiB ->
6.89 MiB
[ 250/ 339]                   blk.20.attn_v.bias - [  512,     1,
1,     1], type =    f32, size =    0.002 MB
[ 251/ 339]                 blk.20.attn_v.weight - [ 3584,   512,
1,     1], type =    f16, converting to q6_K .. size =     3.50 MiB ->
1.44 MiB
[ 252/ 339]                blk.20.ffn_down.weight - [18944,  3584,
1,     1], type =    f16, converting to q6_K .. size =   129.50 MiB ->
53.12 MiB
[ 253/ 339]                blk.20.ffn_gate.weight - [ 3584, 18944,
1,     1], type =    f16, converting to q4_K .. size =   129.50 MiB ->
36.42 MiB
[ 254/ 339]                blk.20.ffn_norm.weight - [ 3584,     1,
1,     1], type =    f32, size =    0.014 MB
[ 255/ 339]                  blk.20.ffn_up.weight - [ 3584, 18944,
1,     1], type =    f16, converting to q4_K .. size =   129.50 MiB ->
36.42 MiB
[ 256/ 339]                   blk.21.attn_k.bias - [  512,     1,
1,     1], type =    f32, size =    0.002 MB
[ 257/ 339]                 blk.21.attn_k.weight - [ 3584,   512,
1,     1], type =    f16, converting to q4_K .. size =     3.50 MiB ->
0.98 MiB
[ 258/ 339]               blk.21.attn_norm.weight - [ 3584,     1,
1,     1], type =    f32, size =    0.014 MB
[ 259/ 339]            blk.21.attn_output.weight - [ 3584,  3584,
1,     1], type =    f16, converting to q4_K .. size =    24.50 MiB ->
6.89 MiB
[ 260/ 339]                   blk.21.attn_q.bias - [ 3584,     1,
1,     1], type =    f32, size =    0.014 MB
[ 261/ 339]                 blk.21.attn_q.weight - [ 3584,  3584,
1,     1], type =    f16, converting to q4_K .. size =    24.50 MiB ->
6.89 MiB
[ 262/ 339]                   blk.21.attn_v.bias - [  512,     1,
1,     1], type =    f32, size =    0.002 MB
[ 263/ 339]                 blk.21.attn_v.weight - [ 3584,   512,
1,     1], type =    f16, converting to q4_K .. size =     3.50 MiB ->
0.98 MiB
[ 264/ 339]                blk.21.ffn_down.weight - [18944,  3584,
1,     1], type =    f16, converting to q4_K .. size =   129.50 MiB ->
36.42 MiB
[ 265/ 339]                blk.21.ffn_gate.weight - [ 3584, 18944,
1,     1], type =    f16, converting to q4_K .. size =   129.50 MiB ->
36.42 MiB
```

```
[ 266/ 339]                blk.21.ffn_norm.weight - [ 3584,     1,
1,     1], type =    f32, size =    0.014 MB
[ 267/ 339]                  blk.21.ffn_up.weight - [ 3584, 18944,
1,     1], type =    f16, converting to q4_K .. size =   129.50 MiB ->
36.42 MiB
[ 268/ 339]                    blk.22.attn_k.bias - [  512,     1,
1,     1], type =    f32, size =    0.002 MB
[ 269/ 339]                  blk.22.attn_k.weight - [ 3584,   512,
1,     1], type =    f16, converting to q4_K .. size =     3.50 MiB ->
0.98 MiB
[ 270/ 339]               blk.22.attn_norm.weight - [ 3584,     1,
1,     1], type =    f32, size =    0.014 MB
[ 271/ 339]             blk.22.attn_output.weight - [ 3584,  3584,
1,     1], type =    f16, converting to q4_K .. size =    24.50 MiB ->
6.89 MiB
[ 272/ 339]                    blk.22.attn_q.bias - [ 3584,     1,
1,     1], type =    f32, size =    0.014 MB
[ 273/ 339]                  blk.22.attn_q.weight - [ 3584,  3584,
1,     1], type =    f16, converting to q4_K .. size =    24.50 MiB ->
6.89 MiB
[ 274/ 339]                    blk.22.attn_v.bias - [  512,     1,
1,     1], type =    f32, size =    0.002 MB
[ 275/ 339]                  blk.22.attn_v.weight - [ 3584,   512,
1,     1], type =    f16, converting to q4_K .. size =     3.50 MiB ->
0.98 MiB
[ 276/ 339]                blk.22.ffn_down.weight - [18944,  3584,
1,     1], type =    f16, converting to q4_K .. size =   129.50 MiB ->
36.42 MiB
[ 277/ 339]                blk.22.ffn_gate.weight - [ 3584, 18944,
1,     1], type =    f16, converting to q4_K .. size =   129.50 MiB ->
36.42 MiB
[ 278/ 339]                blk.22.ffn_norm.weight - [ 3584,     1,
1,     1], type =    f32, size =    0.014 MB
[ 279/ 339]                  blk.22.ffn_up.weight - [ 3584, 18944,
1,     1], type =    f16, converting to q4_K .. size =   129.50 MiB ->
36.42 MiB
[ 280/ 339]                    blk.23.attn_k.bias - [  512,     1,
1,     1], type =    f32, size =    0.002 MB
[ 281/ 339]                  blk.23.attn_k.weight - [ 3584,   512,
1,     1], type =    f16, converting to q4_K .. size =     3.50 MiB ->
0.98 MiB
[ 282/ 339]               blk.23.attn_norm.weight - [ 3584,     1,
1,     1], type =    f32, size =    0.014 MB
[ 283/ 339]             blk.23.attn_output.weight - [ 3584,  3584,
1,     1], type =    f16, converting to q4_K .. size =    24.50 MiB ->
6.89 MiB
[ 284/ 339]                    blk.23.attn_q.bias - [ 3584,     1,
1,     1], type =    f32, size =    0.014 MB
[ 285/ 339]                  blk.23.attn_q.weight - [ 3584,  3584,
```

```
1,     1], type =    f16, converting to q4_K .. size =    24.50 MiB ->
6.89 MiB
[ 286/ 339]                   blk.23.attn_v.bias - [  512,     1,
1,     1], type =    f32, size =    0.002 MB
[ 287/ 339]                 blk.23.attn_v.weight - [ 3584,   512,
1,     1], type =    f16, converting to q6_K .. size =     3.50 MiB ->
1.44 MiB
[ 288/ 339]               blk.23.ffn_down.weight - [18944,  3584,
1,     1], type =    f16, converting to q6_K .. size =   129.50 MiB ->
53.12 MiB
[ 289/ 339]               blk.23.ffn_gate.weight - [ 3584, 18944,
1,     1], type =    f16, converting to q4_K .. size =   129.50 MiB ->
36.42 MiB
[ 290/ 339]               blk.23.ffn_norm.weight - [ 3584,     1,
1,     1], type =    f32, size =    0.014 MB
[ 291/ 339]                 blk.23.ffn_up.weight - [ 3584, 18944,
1,     1], type =    f16, converting to q4_K .. size =   129.50 MiB ->
36.42 MiB
[ 292/ 339]                   blk.24.attn_k.bias - [  512,     1,
1,     1], type =    f32, size =    0.002 MB
[ 293/ 339]                 blk.24.attn_k.weight - [ 3584,   512,
1,     1], type =    f16, converting to q4_K .. size =     3.50 MiB ->
0.98 MiB
[ 294/ 339]              blk.24.attn_norm.weight - [ 3584,     1,
1,     1], type =    f32, size =    0.014 MB
[ 295/ 339]            blk.24.attn_output.weight - [ 3584,  3584,
1,     1], type =    f16, converting to q4_K .. size =    24.50 MiB ->
6.89 MiB
[ 296/ 339]                   blk.24.attn_q.bias - [ 3584,     1,
1,     1], type =    f32, size =    0.014 MB
[ 297/ 339]                 blk.24.attn_q.weight - [ 3584,  3584,
1,     1], type =    f16, converting to q4_K .. size =    24.50 MiB ->
6.89 MiB
[ 298/ 339]                   blk.24.attn_v.bias - [  512,     1,
1,     1], type =    f32, size =    0.002 MB
[ 299/ 339]                 blk.24.attn_v.weight - [ 3584,   512,
1,     1], type =    f16, converting to q6_K .. size =     3.50 MiB ->
1.44 MiB
[ 300/ 339]               blk.24.ffn_down.weight - [18944,  3584,
1,     1], type =    f16, converting to q6_K .. size =   129.50 MiB ->
53.12 MiB
[ 301/ 339]               blk.24.ffn_gate.weight - [ 3584, 18944,
1,     1], type =    f16, converting to q4_K .. size =   129.50 MiB ->
36.42 MiB
[ 302/ 339]               blk.24.ffn_norm.weight - [ 3584,     1,
1,     1], type =    f32, size =    0.014 MB
[ 303/ 339]                 blk.24.ffn_up.weight - [ 3584, 18944,
1,     1], type =    f16, converting to q4_K .. size =   129.50 MiB ->
36.42 MiB
```

```
[ 304/ 339]                    blk.25.attn_k.bias - [  512,     1,
1,     1], type =    f32, size =    0.002 MB
[ 305/ 339]                  blk.25.attn_k.weight - [ 3584,   512,
1,     1], type =    f16, converting to q4_K .. size =     3.50 MiB ->
0.98 MiB
[ 306/ 339]                blk.25.attn_norm.weight - [ 3584,     1,
1,     1], type =    f32, size =    0.014 MB
[ 307/ 339]             blk.25.attn_output.weight - [ 3584,  3584,
1,     1], type =    f16, converting to q4_K .. size =    24.50 MiB ->
6.89 MiB
[ 308/ 339]                    blk.25.attn_q.bias - [ 3584,     1,
1,     1], type =    f32, size =    0.014 MB
[ 309/ 339]                  blk.25.attn_q.weight - [ 3584,  3584,
1,     1], type =    f16, converting to q4_K .. size =    24.50 MiB ->
6.89 MiB
[ 310/ 339]                    blk.25.attn_v.bias - [  512,     1,
1,     1], type =    f32, size =    0.002 MB
[ 311/ 339]                  blk.25.attn_v.weight - [ 3584,   512,
1,     1], type =    f16, converting to q6_K .. size =     3.50 MiB ->
1.44 MiB
[ 312/ 339]                blk.25.ffn_down.weight - [18944,  3584,
1,     1], type =    f16, converting to q6_K .. size =   129.50 MiB ->
53.12 MiB
[ 313/ 339]                blk.25.ffn_gate.weight - [ 3584, 18944,
1,     1], type =    f16, converting to q4_K .. size =   129.50 MiB ->
36.42 MiB
[ 314/ 339]                blk.25.ffn_norm.weight - [ 3584,     1,
1,     1], type =    f32, size =    0.014 MB
[ 315/ 339]                  blk.25.ffn_up.weight - [ 3584, 18944,
1,     1], type =    f16, converting to q4_K .. size =   129.50 MiB ->
36.42 MiB
[ 316/ 339]                    blk.26.attn_k.bias - [  512,     1,
1,     1], type =    f32, size =    0.002 MB
[ 317/ 339]                  blk.26.attn_k.weight - [ 3584,   512,
1,     1], type =    f16, converting to q4_K .. size =     3.50 MiB ->
0.98 MiB
[ 318/ 339]                blk.26.attn_norm.weight - [ 3584,     1,
1,     1], type =    f32, size =    0.014 MB
[ 319/ 339]             blk.26.attn_output.weight - [ 3584,  3584,
1,     1], type =    f16, converting to q4_K .. size =    24.50 MiB ->
6.89 MiB
[ 320/ 339]                    blk.26.attn_q.bias - [ 3584,     1,
1,     1], type =    f32, size =    0.014 MB
[ 321/ 339]                  blk.26.attn_q.weight - [ 3584,  3584,
1,     1], type =    f16, converting to q4_K .. size =    24.50 MiB ->
6.89 MiB
[ 322/ 339]                    blk.26.attn_v.bias - [  512,     1,
1,     1], type =    f32, size =    0.002 MB
[ 323/ 339]                  blk.26.attn_v.weight - [ 3584,   512,
```

```
1,     1], type =    f16, converting to q6_K .. size =     3.50 MiB ->
1.44 MiB
[ 324/ 339]               blk.26.ffn_down.weight - [18944,  3584,
1,     1], type =    f16, converting to q6_K .. size =   129.50 MiB ->
53.12 MiB
[ 325/ 339]               blk.26.ffn_gate.weight - [ 3584, 18944,
1,     1], type =    f16, converting to q4_K .. size =   129.50 MiB ->
36.42 MiB
[ 326/ 339]               blk.26.ffn_norm.weight - [ 3584,     1,
1,     1], type =    f32, size =    0.014 MB
[ 327/ 339]                 blk.26.ffn_up.weight - [ 3584, 18944,
1,     1], type =    f16, converting to q4_K .. size =   129.50 MiB ->
36.42 MiB
[ 328/ 339]                   blk.27.attn_k.bias - [  512,     1,
1,     1], type =    f32, size =    0.002 MB
[ 329/ 339]                 blk.27.attn_k.weight - [ 3584,   512,
1,     1], type =    f16, converting to q4_K .. size =     3.50 MiB ->
0.98 MiB
[ 330/ 339]              blk.27.attn_norm.weight - [ 3584,     1,
1,     1], type =    f32, size =    0.014 MB
[ 331/ 339]            blk.27.attn_output.weight - [ 3584,  3584,
1,     1], type =    f16, converting to q4_K .. size =    24.50 MiB ->
6.89 MiB
[ 332/ 339]                   blk.27.attn_q.bias - [ 3584,     1,
1,     1], type =    f32, size =    0.014 MB
[ 333/ 339]                 blk.27.attn_q.weight - [ 3584,  3584,
1,     1], type =    f16, converting to q4_K .. size =    24.50 MiB ->
6.89 MiB
[ 334/ 339]                   blk.27.attn_v.bias - [  512,     1,
1,     1], type =    f32, size =    0.002 MB
[ 335/ 339]                 blk.27.attn_v.weight - [ 3584,   512,
1,     1], type =    f16, converting to q6_K .. size =     3.50 MiB ->
1.44 MiB
[ 336/ 339]               blk.27.ffn_down.weight - [18944,  3584,
1,     1], type =    f16, converting to q6_K .. size =   129.50 MiB ->
53.12 MiB
[ 337/ 339]               blk.27.ffn_gate.weight - [ 3584, 18944,
1,     1], type =    f16, converting to q4_K .. size =   129.50 MiB ->
36.42 MiB
[ 338/ 339]               blk.27.ffn_norm.weight - [ 3584,     1,
1,     1], type =    f32, size =    0.014 MB
[ 339/ 339]                 blk.27.ffn_up.weight - [ 3584, 18944,
1,     1], type =    f16, converting to q4_K .. size =   129.50 MiB ->
36.42 MiB
llama_model_quantize_impl: model size  = 14526.27 MB
llama_model_quantize_impl: quant size  =  4460.45 MB

main: quantize time = 805645.67 ms
main:    total time = 805645.67 ms
```

```
Unsloth: Conversion completed! Output location:
/content/AkinduH/Qwen2.5-3B-Instruct-Fine-Tuned-on-Deepseek-Research-
Papers/unsloth.Q4_K_M.gguf
Unsloth: Saved Ollama Modelfile to AkinduH/Qwen2.5-3B-Instruct-Fine-
Tuned-on-Deepseek-Research-Papers/Modelfile
Unsloth: Uploading GGUF to Huggingface Hub...

{"model_id":"e54195385daf459db85f07a70e26cb3a","version_major":2,"vers
ion_minor":0}

{"model_id":"5bd0af5659c14235a36313bb7d539cd0","version_major":2,"vers
ion_minor":0}

Saved GGUF to https://huggingface.co/AkinduH/Qwen2.5-3B-Instruct-Fine-
Tuned-on-Deepseek-Research-Papers

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.

Saved Ollama Modelfile to https://huggingface.co/AkinduH/Qwen2.5-3B-
Instruct-Fine-Tuned-on-Deepseek-Research-Papers
```