

```
!pip install unsloth transformers trl
```

```
Requirement already satisfied: unsloth in  
/usr/local/lib/python3.12/dist-packages (2025.9.11)  
Requirement already satisfied: transformers in  
/usr/local/lib/python3.12/dist-packages (4.56.1)  
Requirement already satisfied: trl in /usr/local/lib/python3.12/dist-  
packages (0.23.0)  
Requirement already satisfied: unsloth_zoo>=2025.9.13 in  
/usr/local/lib/python3.12/dist-packages (from unsloth) (2025.9.14)  
Requirement already satisfied: torch>=2.4.0 in  
/usr/local/lib/python3.12/dist-packages (from unsloth) (2.8.0+cu126)  
Requirement already satisfied: xformers>=0.0.27.post2 in  
/usr/local/lib/python3.12/dist-packages (from unsloth) (0.0.32.post2)  
Requirement already satisfied: bitsandbytes in  
/usr/local/lib/python3.12/dist-packages (from unsloth) (0.48.0)  
Requirement already satisfied: triton>=3.0.0 in  
/usr/local/lib/python3.12/dist-packages (from unsloth) (3.4.0)  
Requirement already satisfied: packaging in  
/usr/local/lib/python3.12/dist-packages (from unsloth) (25.0)  
Requirement already satisfied: tyro in /usr/local/lib/python3.12/dist-  
packages (from unsloth) (0.9.32)  
Requirement already satisfied: datasets!=4.0.*,!=4.1.0,>=3.4.1 in  
/usr/local/lib/python3.12/dist-packages (from unsloth) (4.1.1)  
Requirement already satisfied: sentencepiece>=0.2.0 in  
/usr/local/lib/python3.12/dist-packages (from unsloth) (0.2.1)  
Requirement already satisfied: tqdm in /usr/local/lib/python3.12/dist-  
packages (from unsloth) (4.67.1)  
Requirement already satisfied: psutil in  
/usr/local/lib/python3.12/dist-packages (from unsloth) (5.9.5)  
Requirement already satisfied: wheel>=0.42.0 in  
/usr/local/lib/python3.12/dist-packages (from unsloth) (0.45.1)  
Requirement already satisfied: numpy in  
/usr/local/lib/python3.12/dist-packages (from unsloth) (2.0.2)  
Requirement already satisfied: accelerate>=0.34.1 in  
/usr/local/lib/python3.12/dist-packages (from unsloth) (1.10.1)  
Requirement already satisfied: peft!=0.11.0,>=0.7.1 in  
/usr/local/lib/python3.12/dist-packages (from unsloth) (0.17.1)  
Requirement already satisfied: protobuf in  
/usr/local/lib/python3.12/dist-packages (from unsloth) (5.29.5)  
Requirement already satisfied: huggingface_hub>=0.34.0 in  
/usr/local/lib/python3.12/dist-packages (from unsloth) (0.35.0)  
Requirement already satisfied: hf_transfer in  
/usr/local/lib/python3.12/dist-packages (from unsloth) (0.1.9)  
Requirement already satisfied: diffusers in  
/usr/local/lib/python3.12/dist-packages (from unsloth) (0.35.1)  
Requirement already satisfied: torchvision in  
/usr/local/lib/python3.12/dist-packages (from unsloth) (0.23.0+cu126)  
Requirement already satisfied: filelock in  
/usr/local/lib/python3.12/dist-packages (from transformers) (3.19.1)
```

Requirement already satisfied: pyyaml<=5.1 in
/usr/local/lib/python3.12/dist-packages (from transformers) (6.0.2)

Requirement already satisfied: regex!=2019.12.17 in
/usr/local/lib/python3.12/dist-packages (from transformers)
(2024.11.6)

Requirement already satisfied: requests in
/usr/local/lib/python3.12/dist-packages (from transformers) (2.32.4)

Requirement already satisfied: tokenizers<=0.23.0,>=0.22.0 in
/usr/local/lib/python3.12/dist-packages (from transformers) (0.22.0)

Requirement already satisfied: safetensors<=0.4.3 in
/usr/local/lib/python3.12/dist-packages (from transformers) (0.6.2)

Requirement already satisfied: pyarrow<=21.0.0 in
/usr/local/lib/python3.12/dist-packages (from datasets!=4.0.*,!
=4.1.0,>=3.4.1->unsloth) (21.0.0)

Requirement already satisfied: dill<0.4.1,>=0.3.0 in
/usr/local/lib/python3.12/dist-packages (from datasets!=4.0.*,!
=4.1.0,>=3.4.1->unsloth) (0.3.8)

Requirement already satisfied: pandas in
/usr/local/lib/python3.12/dist-packages (from datasets!=4.0.*,!
=4.1.0,>=3.4.1->unsloth) (2.2.2)

Requirement already satisfied: xxhash in
/usr/local/lib/python3.12/dist-packages (from datasets!=4.0.*,!
=4.1.0,>=3.4.1->unsloth) (3.5.0)

Requirement already satisfied: multiprocessing<0.70.17 in
/usr/local/lib/python3.12/dist-packages (from datasets!=4.0.*,!
=4.1.0,>=3.4.1->unsloth) (0.70.16)

Requirement already satisfied: fsspec<=2025.9.0,>=2023.1.0 in
/usr/local/lib/python3.12/dist-packages (from
fsspec[http]<=2025.9.0,>=2023.1.0->datasets!=4.0.*,!=4.1.0,>=3.4.1-
>unsloth) (2025.3.0)

Requirement already satisfied: typing-extensions<=3.7.4.3 in
/usr/local/lib/python3.12/dist-packages (from huggingface_hub>=0.34.0-
>unsloth) (4.15.0)

Requirement already satisfied: hf-xet<2.0.0,>=1.1.3 in
/usr/local/lib/python3.12/dist-packages (from huggingface_hub>=0.34.0-
>unsloth) (1.1.10)

Requirement already satisfied: charset_normalizer<4,>=2 in
/usr/local/lib/python3.12/dist-packages (from requests->transformers)
(3.4.3)

Requirement already satisfied: idna<4,>=2.5 in
/usr/local/lib/python3.12/dist-packages (from requests->transformers)
(3.10)

Requirement already satisfied: urllib3<3,>=1.21.1 in
/usr/local/lib/python3.12/dist-packages (from requests->transformers)
(2.5.0)

Requirement already satisfied: certifi<=2017.4.17 in
/usr/local/lib/python3.12/dist-packages (from requests->transformers)
(2025.8.3)

Requirement already satisfied: setuptools in

```
/usr/local/lib/python3.12/dist-packages (from torch>=2.4.0->unsloth)
(75.2.0)
Requirement already satisfied: sympy>=1.13.3 in
/usr/local/lib/python3.12/dist-packages (from torch>=2.4.0->unsloth)
(1.13.3)
Requirement already satisfied: networkx in
/usr/local/lib/python3.12/dist-packages (from torch>=2.4.0->unsloth)
(3.5)
Requirement already satisfied: jinja2 in
/usr/local/lib/python3.12/dist-packages (from torch>=2.4.0->unsloth)
(3.1.6)
Requirement already satisfied: nvidia-cuda-nvrtc-cu12==12.6.77 in
/usr/local/lib/python3.12/dist-packages (from torch>=2.4.0->unsloth)
(12.6.77)
Requirement already satisfied: nvidia-cuda-runtime-cu12==12.6.77 in
/usr/local/lib/python3.12/dist-packages (from torch>=2.4.0->unsloth)
(12.6.77)
Requirement already satisfied: nvidia-cuda-cupti-cu12==12.6.80 in
/usr/local/lib/python3.12/dist-packages (from torch>=2.4.0->unsloth)
(12.6.80)
Requirement already satisfied: nvidia-cudnn-cu12==9.10.2.21 in
/usr/local/lib/python3.12/dist-packages (from torch>=2.4.0->unsloth)
(9.10.2.21)
Requirement already satisfied: nvidia-cublas-cu12==12.6.4.1 in
/usr/local/lib/python3.12/dist-packages (from torch>=2.4.0->unsloth)
(12.6.4.1)
Requirement already satisfied: nvidia-cufft-cu12==11.3.0.4 in
/usr/local/lib/python3.12/dist-packages (from torch>=2.4.0->unsloth)
(11.3.0.4)
Requirement already satisfied: nvidia-curand-cu12==10.3.7.77 in
/usr/local/lib/python3.12/dist-packages (from torch>=2.4.0->unsloth)
(10.3.7.77)
Requirement already satisfied: nvidia-cusolver-cu12==11.7.1.2 in
/usr/local/lib/python3.12/dist-packages (from torch>=2.4.0->unsloth)
(11.7.1.2)
Requirement already satisfied: nvidia-cusparse-cu12==12.5.4.2 in
/usr/local/lib/python3.12/dist-packages (from torch>=2.4.0->unsloth)
(12.5.4.2)
Requirement already satisfied: nvidia-cusparselt-cu12==0.7.1 in
/usr/local/lib/python3.12/dist-packages (from torch>=2.4.0->unsloth)
(0.7.1)
Requirement already satisfied: nvidia-nccl-cu12==2.27.3 in
/usr/local/lib/python3.12/dist-packages (from torch>=2.4.0->unsloth)
(2.27.3)
Requirement already satisfied: nvidia-nvtx-cu12==12.6.77 in
/usr/local/lib/python3.12/dist-packages (from torch>=2.4.0->unsloth)
(12.6.77)
Requirement already satisfied: nvidia-nvjitlink-cu12==12.6.85 in
/usr/local/lib/python3.12/dist-packages (from torch>=2.4.0->unsloth)
```

(12.6.85)
Requirement already satisfied: nvidia-cufile-cu12==1.11.1.6 in
/usr/local/lib/python3.12/dist-packages (from torch>=2.4.0->unsloth)
(1.11.1.6)
Requirement already satisfied: torchao in
/usr/local/lib/python3.12/dist-packages (from unsloth_zoo>=2025.9.13->unsloth) (0.10.0)
Requirement already satisfied: cut_cross_entropy in
/usr/local/lib/python3.12/dist-packages (from unsloth_zoo>=2025.9.13->unsloth) (25.1.1)
Requirement already satisfied: pillow in
/usr/local/lib/python3.12/dist-packages (from unsloth_zoo>=2025.9.13->unsloth) (11.3.0)
Requirement already satisfied: msgspec in
/usr/local/lib/python3.12/dist-packages (from unsloth_zoo>=2025.9.13->unsloth) (0.19.0)
Requirement already satisfied: importlib_metadata in
/usr/local/lib/python3.12/dist-packages (from diffusers->unsloth)
(8.7.0)
Requirement already satisfied: docstring-parser>=0.15 in
/usr/local/lib/python3.12/dist-packages (from tyro->unsloth) (0.17.0)
Requirement already satisfied: rich>=11.1.0 in
/usr/local/lib/python3.12/dist-packages (from tyro->unsloth) (13.9.4)
Requirement already satisfied: shtab>=1.5.6 in
/usr/local/lib/python3.12/dist-packages (from tyro->unsloth) (1.7.2)
Requirement already satisfied: typeguard>=4.0.0 in
/usr/local/lib/python3.12/dist-packages (from tyro->unsloth) (4.4.4)
Requirement already satisfied: aiohttp!=4.0.0a0,!4.0.0a1 in
/usr/local/lib/python3.12/dist-packages (from
fsspec[http]<=2025.9.0,>=2023.1.0->datasets!=4.0.*,!=4.1.0,>=3.4.1->unsloth) (3.12.15)
Requirement already satisfied: markdown-it-py>=2.2.0 in
/usr/local/lib/python3.12/dist-packages (from rich>=11.1.0->tyro->unsloth) (4.0.0)
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in
/usr/local/lib/python3.12/dist-packages (from rich>=11.1.0->tyro->unsloth) (2.19.2)
Requirement already satisfied: mpmath<1.4,>=1.1.0 in
/usr/local/lib/python3.12/dist-packages (from sympy>=1.13.3->torch>=2.4.0->unsloth) (1.3.0)
Requirement already satisfied: zipp>=3.20 in
/usr/local/lib/python3.12/dist-packages (from importlib_metadata->diffusers->unsloth) (3.23.0)
Requirement already satisfied: MarkupSafe>=2.0 in
/usr/local/lib/python3.12/dist-packages (from jinja2->torch>=2.4.0->unsloth) (3.0.2)
Requirement already satisfied: python-dateutil>=2.8.2 in
/usr/local/lib/python3.12/dist-packages (from pandas->datasets!=4.0.*,!=4.1.0,>=3.4.1->unsloth) (2.9.0.post0)

```
Requirement already satisfied: pytz>=2020.1 in
/usr/local/lib/python3.12/dist-packages (from pandas->datasets!
=4.0.*,!=4.1.0,>=3.4.1->unsloth) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in
/usr/local/lib/python3.12/dist-packages (from pandas->datasets!
=4.0.*,!=4.1.0,>=3.4.1->unsloth) (2025.2)
Requirement already satisfied: aiohappyeyeballs>=2.5.0 in
/usr/local/lib/python3.12/dist-packages (from aiohttp!=4.0.0a0,!
=4.0.0a1->fsspec[http]<=2025.9.0,>=2023.1.0->datasets!=4.0.*,!
=4.1.0,>=3.4.1->unsloth) (2.6.1)
Requirement already satisfied: aiosignal>=1.4.0 in
/usr/local/lib/python3.12/dist-packages (from aiohttp!=4.0.0a0,!
=4.0.0a1->fsspec[http]<=2025.9.0,>=2023.1.0->datasets!=4.0.*,!
=4.1.0,>=3.4.1->unsloth) (1.4.0)
Requirement already satisfied: attrs>=17.3.0 in
/usr/local/lib/python3.12/dist-packages (from aiohttp!=4.0.0a0,!
=4.0.0a1->fsspec[http]<=2025.9.0,>=2023.1.0->datasets!=4.0.*,!
=4.1.0,>=3.4.1->unsloth) (25.3.0)
Requirement already satisfied: frozenlist>=1.1.1 in
/usr/local/lib/python3.12/dist-packages (from aiohttp!=4.0.0a0,!
=4.0.0a1->fsspec[http]<=2025.9.0,>=2023.1.0->datasets!=4.0.*,!
=4.1.0,>=3.4.1->unsloth) (1.7.0)
Requirement already satisfied: multidict<7.0,>=4.5 in
/usr/local/lib/python3.12/dist-packages (from aiohttp!=4.0.0a0,!
=4.0.0a1->fsspec[http]<=2025.9.0,>=2023.1.0->datasets!=4.0.*,!
=4.1.0,>=3.4.1->unsloth) (6.6.4)
Requirement already satisfied: propcache>=0.2.0 in
/usr/local/lib/python3.12/dist-packages (from aiohttp!=4.0.0a0,!
=4.0.0a1->fsspec[http]<=2025.9.0,>=2023.1.0->datasets!=4.0.*,!
=4.1.0,>=3.4.1->unsloth) (0.3.2)
Requirement already satisfied: yarl<2.0,>=1.17.0 in
/usr/local/lib/python3.12/dist-packages (from aiohttp!=4.0.0a0,!
=4.0.0a1->fsspec[http]<=2025.9.0,>=2023.1.0->datasets!=4.0.*,!
=4.1.0,>=3.4.1->unsloth) (1.20.1)
Requirement already satisfied: mdurl~=0.1 in
/usr/local/lib/python3.12/dist-packages (from markdown-it-py>=2.2.0-
>rich>=11.1.0->tyro->unsloth) (0.1.2)
Requirement already satisfied: six>=1.5 in
/usr/local/lib/python3.12/dist-packages (from python-dateutil>=2.8.2-
>pandas->datasets!=4.0.*,!=4.1.0,>=3.4.1->unsloth) (1.17.0)
```

```
import torch
from unsloth import FastLanguageModel
from datasets import load_dataset
from trl import SFTTrainer
from transformers import TrainingArguments
from unsloth.chat_templates import get_chat_template,
standardize_sharegpt
```

```

model, tokenizer = FastLanguageModel.from_pretrained(
    model_name = "unsloth/Llama-3.2-1B-Instruct",
    max_seq_length=2048,
    load_in_4bit=True
)

==(====)== Unsloth 2025.9.11: Fast Llama patching. Transformers:
4.56.1.
  \ \ / | Tesla T4. Num GPUs = 1. Max memory: 14.741 GB. Platform:
Linux.
0^0/ \_/ \ Torch: 2.8.0+cu126. CUDA: 7.5. CUDA Toolkit: 12.6.
Triton: 3.4.0
\ / Bfloat16 = FALSE. FA [Xformers = 0.0.32.post2. FA2 =
False]
"-_____"- Free license: http://github.com/unslothai/unsloth
Unsloth: Fast downloading is enabled - ignore downloading bars which
are red colored!

model = FastLanguageModel.get_peft_model(
    model, r=16,
    target_modules = ["q_proj", "k_proj", "v_proj", "o_proj",
"gate_proj", "up_proj", "down_proj"]
)

tokenizer = get_chat_template(tokenizer, chat_template="llama-3.1")
dataset = load_dataset("mlabonne/FineTome-100k", split="train")
dataset = standardize_sharegpt(dataset)
dataset

Dataset({
  features: ['conversations', 'source', 'score'],
  num_rows: 100000
})

dataset[1]

{'conversations': [{'content': 'Explain how recursion works and
provide a recursive function in Python that calculates the factorial
of a given number.',
  'role': 'user'},
  {'content': "Recursion is a programming technique where a function
calls itself to solve a problem. It breaks down a complex problem into
smaller, more manageable subproblems until a base case is reached. The
base case is a condition where the function does not call itself, but
instead returns a specific value or performs a specific action.\n\nIn
the case of calculating the factorial of a number, recursion can be
used to break down the problem into simpler subproblems. The factorial
of a non-negative integer n is the product of all positive integers

```

less than or equal to n.\n\nHere is a recursive function in Python that calculates the factorial of a given number:\n\n`python\ndef factorial(n):\n # Base case: factorial of 0 or 1 is 1\n if n == 0 or n == 1:\n return 1\n # Recursive case: factorial of n is n multiplied by factorial of (n-1)\n else:\n return n * factorial(n - 1)\n`\n\nIn this function, the base case is when n equals 0 or 1, as the factorial of these numbers is defined as 1. For any other positive integer, the function calls itself with a smaller value (n-1) and multiplies the result by n.\n\nFor example, let's calculate the factorial of 5 using this function:\n\n`python\nprint(factorial(5))\n`\n\nThe function will execute as follows:\n1. factorial(5) calls factorial(4) and multiplies the result by 5.\n2. factorial(4) calls factorial(3) and multiplies the result by 4.\n3. factorial(3) calls factorial(2) and multiplies the result by 3.\n4. factorial(2) calls factorial(1) and multiplies the result by 2.\n5. factorial(1) is a base case and returns 1.\n6. The final result is calculated as: 5 * 4 * 3 * 2 * 1 = 120.\n\n`role`: 'assistant'}],\n`source`: 'infini-instruct-top-500k',\n`score`: 5.157649040222168}

```
dataset = dataset.map(
    lambda examples:{
        "text": [
            tokenizer.apply_chat_template(convo, tokenize=False)
            for convo in examples["conversations"]
        ]
    },
    batched=True
)
```

dataset

```
Dataset({
  features: ['conversations', 'source', 'score'],
  num_rows: 100000
})
```

dataset[0]

```
{'conversations': [{'content': 'Explain what boolean operators are, what they do, and provide examples of how they can be used in programming. Additionally, describe the concept of operator precedence and provide examples of how it affects the evaluation of boolean expressions. Discuss the difference between short-circuit evaluation and normal evaluation in boolean expressions and demonstrate their usage in code. \n\nFurthermore, add the requirement that the code must be written in a language that does not support short-circuit evaluation natively, forcing the test taker to implement their own logic for short-circuit evaluation.\n\nFinally, delve into the concept
```

of truthiness and falsiness in programming languages, explaining how it affects the evaluation of boolean expressions. Add the constraint that the test taker must write code that handles cases where truthiness and falsiness are implemented differently across different programming languages.',

'role': 'user'},

{'content': 'Boolean operators are logical operators used in programming to manipulate boolean values. They operate on one or more boolean operands and return a boolean result. The three main boolean operators are "AND" (&&), "OR" (||), and "NOT" (!).\n\nThe "AND" operator returns true if both of its operands are true, and false otherwise. For example:\n\n```\npython\nx = 5\ny = 10\nresult = (x > 0) and (y < 20) # This expression evaluates to True\n```\n\nThe "OR" operator returns true if at least one of its operands is true, and false otherwise. For example:\n\n```\npython\nx = 5\ny = 10\nresult = (x > 0) or (y < 20) # This expression evaluates to True\n```\n\nThe "NOT" operator negates the boolean value of its operand. It returns true if the operand is false, and false if the operand is true. For example:\n\n```\npython\nx = 5\nresult = not (x > 10) # This expression evaluates to True\n```\n\nOperator precedence refers to the order in which operators are evaluated in an expression. It ensures that expressions are evaluated correctly. In most programming languages, logical AND has higher precedence than logical OR. For example:\n\n```\npython\nresult = True or False and False # This expression is evaluated as (True or (False and False)), which is True\n```\n\nShort-circuit evaluation is a behavior where the second operand of a logical operator is not evaluated if the result can be determined based on the value of the first operand. In short-circuit evaluation, if the first operand of an "AND" operator is false, the second operand is not evaluated because the result will always be false. Similarly, if the first operand of an "OR" operator is true, the second operand is not evaluated because the result will always be true.\n\nIn programming languages that support short-circuit evaluation natively, you can use it to improve performance or avoid errors. For example:\n\n```\npython\nif x != 0 and (y / x) > 10:\n # Perform some operation\n```\n\nIn languages without native short-circuit evaluation, you can implement your own logic to achieve the same behavior. Here's an example in pseudocode:\n\n```\nif x != 0 {\n if (y / x) > 10 {\n // Perform some operation\n }\n}\n```\n\nTruthiness and falsiness refer to how non-boolean values are evaluated in boolean contexts. In many programming languages, non-zero numbers and non-empty strings are considered truthy, while zero, empty strings, and null/None values are considered falsy.\n\nWhen evaluating boolean expressions, truthiness and falsiness come into play. For example:\n\n```\npython\nx = 5\nresult = x # The value of x is truthy, so result is also truthy\n```\n\nTo handle cases where truthiness and falsiness are implemented differently across programming languages, you can explicitly check the desired condition. For example:\n\n```\npython\nx = 5\nresult = bool(x)\n# Explicitly converting x to a boolean value\n```\n\nThis ensures that


```
the result is always a boolean value, regardless of the language\'s
truthiness and falsiness rules.',
    'role': 'assistant'}]],
    'source': 'infini-instruct-top-500k',
    'score': 5.212620735168457}
```

```
from transformers import AutoTokenizer
```

```
tokenizer = AutoTokenizer.from_pretrained("unsloth/Llama-3.2-1B-
Instruct")
```

```
from transformers import AutoTokenizer
```

```
tokenizer = AutoTokenizer.from_pretrained("unsloth/Llama-3.2-1B-
Instruct")
```

```
dataset = dataset.map(
    lambda examples:{
        "text": [
            tokenizer.apply_chat_template(convo, tokenize=False)
            for convo in examples["conversations"]
        ]
    },
    batched=True
)
```

```
def formatting_func(examples):
    return examples["text"]
```

```
trainer = SFTTrainer(
    model=model,
    tokenizer=tokenizer,
    train_dataset=dataset,
    dataset_text_field="text",
    max_seq_length=2048,
    args=TrainingArguments(
        per_device_train_batch_size=2,
        gradient_accumulation_steps=4,
        warmup_steps=5,
        max_steps=60,
        learning_rate=2e-4,
        fp16=not torch.cuda.is_bf16_supported(),
        bf16=torch.cuda.is_bf16_supported(),
        logging_steps=1,
        output_dir="outputs",
    ),
    formatting_func=formatting_func,
)
```

```
{"model_id": "d1565260eee54ec8b075f1ec2cfd06c4", "version_major": 2, "version_minor": 0}
```

```
{"model_id": "9c021e6f39c34f84af0d55379af2bc3c", "version_major": 2, "version_minor": 0}
```

```
trainer.train()
```

```
==(=====)== Unsloth - 2x faster free finetuning | Num GPUs used = 1
  \ \      / |   Num examples = 100,000 | Num Epochs = 1 | Total steps =
60
0^0/ \_/ \     Batch size per device = 2 | Gradient accumulation steps
= 4
\         /     Data Parallel GPUs = 1 | Total batch size (2 x 4 x 1) =
8
"-_____"-      Trainable parameters = 11,272,192 of 1,247,086,592
(0.90% trained)
```

```
<IPython.core.display.Javascript object>
```

```
wandb: Logging into wandb.ai. (Learn how to deploy a W&B server
locally: https://wandb.me/wandb-server)
```

```
wandb: You can find your API key in your browser here:
```

```
https://wandb.ai/authorize?ref=models
```

```
wandb: Paste an API key from your profile and hit enter:
```

```
.....
```

```
wandb: WARNING If you're specifying your api key in code, ensure this
code is not shared publicly.
```

```
wandb: WARNING Consider setting the WANDB_API_KEY environment
variable, or running `wandb login` from the command line.
```

```
wandb: No netrc file found, creating one.
```

```
wandb: Appending key for api.wandb.ai to your netrc file: /root/.netrc
```

```
wandb: Currently logged in as: ayanpanja348 (ayanpanja348-indian-
institute-of-technology-patna) to https://api.wandb.ai. Use `wandb
login --relogin` to force relogin
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
wandb: Detected [huggingface_hub.inference, openai] in use.
```

```
wandb: Use W&B Weave for improved LLM call tracing. Install Weave with
```

`pip install weave` then add `import weave` to the top of your script.
wandb: For more information, check out the docs at: <https://weave-docs.wandb.ai/>

<IPython.core.display.HTML object>

```
TrainOutput(global_step=60, training_loss=1.1610590428113938,
metrics={'train_runtime': 206.1845, 'train_samples_per_second': 2.328,
'train_steps_per_second': 0.291, 'total_flos': 1902805765840896.0,
'train_loss': 1.1610590428113938, 'epoch': 0.0048})
```

```
model.save_pretrained("finetuned_model")
```

```
inference_model, inference_tokenizer =
FastLanguageModel.from_pretrained(
    model_name="finetuned_model",
    max_seq_length=2048,
    load_in_4bit=True
)
```

```
==(====)== Unsloth 2025.9.11: Fast Llama patching. Transformers:
4.56.1.
```

```
\\    /|    Tesla T4. Num GPUs = 1. Max memory: 14.741 GB. Platform:
Linux.
```

```
0^0/ \_/ \    Torch: 2.8.0+cu126. CUDA: 7.5. CUDA Toolkit: 12.6.
```

```
Triton: 3.4.0
```

```
\    /    Bfloat16 = FALSE. FA [Xformers = 0.0.32.post2. FA2 =
False]
```

```
"-_____"      Free license: http://github.com/unslothai/unsloth
Unsloth: Fast downloading is enabled - ignore downloading bars which
are red colored!
```

```
text_prompts = [
    "what are the key principles of investement?"
]
```

```
for prompt in text_prompts:
    formatted_prompts = inference_tokenizer.apply_chat_template([
        "role": "user",
        "content": prompt
    ], tokenize=False)
```

```
model_input = inference_tokenizer(formatted_prompts,
return_tensors="pt").to("cuda")
```

```
generated_ids = inference_model.generate(
    **model_input,
    max_new_tokens=512,
    temperature=0.7,
    do_sample=True,
    pad_token_id=inference_tokenizer.pad_token_id
)
```

```
response = inference_tokenizer.batch_decode(generated_ids,
```

```
skip_special_tokens=True)[0]  
print(response)
```

system

Cutting Knowledge Date: December 2023

Today Date: 02 Oct 2025

user

what are the key principles of investment?assistant

Key principles of investment include:

1. Diversification: Diversifying your investment portfolio by spreading investments across different asset classes, sectors, and geographic regions can help manage risk and increase potential returns.
2. Long-term perspective: Holding onto investments for a long time can help ride out market fluctuations and potentially lead to higher returns.
3. Risk management: Identifying and managing risk can help minimize losses and maximize gains.
4. Diversification of income streams: Having multiple income streams can help reduce dependence on a single source of income and provide a safety net.
5. Low-cost investing: Using low-cost index funds or ETFs can help reduce investment costs and increase potential returns.
6. Tax optimization: Understanding tax implications and optimizing tax strategies can help minimize tax liabilities and maximize after-tax returns.
7. Rebalancing: Regularly reviewing and rebalancing your portfolio can help ensure it remains aligned with your investment goals and risk tolerance.
8. Education and research: Continuously learning about investing and staying up-to-date with market trends can help make informed decisions and stay ahead of the curve.
9. Risk tolerance: Understanding your risk tolerance and adjusting your investment strategy accordingly can help manage risk and maximize potential returns.
10. Long-term focus: Prioritizing long-term goals and avoiding short-term pressure to make quick profits can help create a stable investment foundation.