I created a folder /nlp_project and inside it:

```
sudo apt install python3-pip
sudo apt install python3-venv

cd /boot/nlp_project/
$/nlp_project$ python3 -m venv env
source env/bin/activate
$/nlp_project$ pip install torch torchvision torchaudio transformers


python3 -m venv env
source env/bin/activate

python test_cuda.py
True
1
NVIDIA GeForce RTX 4060
```

indicative that torch is enabled and using o cuda on GPU

to enable, install and run a deepseek model:

```
 pip install deepseek
```

I tryed the

```
 model_name = "deepseek-ai/DeepSeek-R1"
```

at default but got

```
python deep_seek_verify.py

Unknown quantization type, got fp8 - supported types are: ['awq',
'bitsandbytes_4bit', 'bitsandbytes_8bit', 'gptq', 'aqlm', 'quanto', 'eetq',
'hqq', 'compressed-tensors', 'fbgemm_fp8', 'torchao', 'bitnet']
```

```
pip install --pre torch --index-url
https://download.pytorch.org/whl/nightly/cu126/
pip install --pre fbgemm-gpu --index-url
https://download.pytorch.org/whl/nightly/cu126/
pip install --upgrade accelerate

python deep_seek_verify.py

ValueError: FP8 quantized models is only supported on GPUs with compute
capability >= 9.0 (e.g H100)
```

Meaning I can´t run FP8 with my 4060 (nor a 4090) then I tryed forcing 'bitsandbytes_8bit'

```
pip install -U bitsandbytes
```

```
if not hasattr(config, 'quantization_config'):
    config.quantization_config = {}
config.quantization_config['quantization_type'] = 'bitsandbytes_8bit'
config.quantization_config['load_in_8bit'] = True
```

And it was running, but the overall file size would be more than 800GB, got this at first run:

```
python deep_seek_verify.py trust_remote_code=True
The repository for deepseek-ai/DeepSeek-R1 contains custom code which must
be executed to correctly load the model. You can inspect the repository
content at https://hf.co/deepseek-ai/DeepSeek-R1.
You can avoid this prompt in future by passing the argument
`trust_remote_code=True`.

Do you wish to run the custom code? [y/N] y
The repository for deepseek-ai/DeepSeek-R1 contains custom code which must
be executed to correctly load the model. You can inspect the repository
content at https://hf.co/deepseek-ai/DeepSeek-R1.
You can avoid this prompt in future by passing the argument
`trust_remote_code=True`.

Do you wish to run the custom code? [y/N] y
Unused kwargs: ['activation_scheme', 'fmt', 'quant_method',
'weight_block_size', 'quantization_type']. These kwargs are not used in
<class 'transformers.utils.quantization_config.BitsAndBytesConfig'>.
`low_cpu_mem_usage` was None, now default to True since model is quantized.
model.safetensors.index.json: 100%|
███████████████████████████████████████████████████████████████████████████
███████████████████████| 8.90M/8.90M [00:00<00:00, 9.29MB/s]
model-00001-of-000163.safetensors: 100%|
██████████████████████████████████████████████████████████████████████████
██████████████████| 5.23G/5.23G [02:34<00:00, 33.9MB/s]
```

It would take too long and maybe not run. so I went for the pretrained models, the larger one with 70B considering what is in https://huggingface.co/deepseek-ai/DeepSeek-R1/blob/main/README.md , that is a Llama3.3 model fine-tuned with samples generated by DeepSeek-R1 .

That means running:

```
 import os
import torch
from transformers import AutoModelForCausalLM, AutoTokenizer, AutoConfig

# Set the cache directory
os.environ['TRANSFORMERS_CACHE'] = '/usr/local/huggingface_cache'

model_name = "deepseek-ai/DeepSeek-R1-Distill-Llama-70B"
tokenizer = AutoTokenizer.from_pretrained(model_name)

# Load the model configuration
config = AutoConfig.from_pretrained(model_name)

# Force the quantization type to 'bitsandbytes_8bit'
if not hasattr(config, 'quantization_config'):
    config.quantization_config = {}
config.quantization_config['quantization_type'] = 'bitsandbytes_8bit'
config.quantization_config['load_in_8bit'] = True

# Load the model with the updated configuration
model = AutoModelForCausalLM.from_pretrained(model_name, config=config)

input_text = "Your input text here"
input_ids = tokenizer.encode(input_text, return_tensors="pt")

with torch.no_grad():
    output = model.generate(
        input_ids,
        max_length=50,  # Maximum number of tokens to generate
        num_return_sequences=1,  # Number of sequences to generate
        no_repeat_ngram_size=2,  # Prevent repetition of n-grams
        top_k=50,  # Top-k sampling
        top_p=0.95,  # Nucleus sampling
        temperature=0.7  # Sampling temperature
    )

# Decode the output tokens to text
output_text = tokenizer.decode(output[0], skip_special_tokens=True)
print("Generated Text:")
print(output_text)
```

and got this on the terminal:

```
$ python deep_seek_verify.py --trust_remote_code=True
Unused kwargs: ['quantization_type']. These kwargs are not used in <class
'transformers.utils.quantization_config.BitsAndBytesConfig'>.
`low_cpu_mem_usage` was None, now default to True since model is quantized.
model-00002-of-000017.safetensors: 100%|
████████████████████████████████████████████████████████████████████████████
██████████████████| 8.69G/8.69G [04:06<00:00, 33.9MB/s]
model-00003-of-000017.safetensors: 100%|
████████████████████████████████████████████████████████████████████████████
██████████████████| 1.58G/1.58G [00:46<00:00, 34.0MB/s]
```

That is much smaller, around 150GB from https://huggingface.co/deepseek-ai/DeepSeek-R1-Distill-Llama-70B