**Running a LLM on Raspberry Pi using conda-forge mini, torch and huggingface libraries such as transformers, accelerate, bitsandbytes**

|  |
| --- |
| LLM: SmolLM-135M-Instruct  Python Tool: Conda-Forge Mini  Libraries: pytorch, huggingface\_hub, transformers, accelerate, bitsandbytes |

**Finding Information about Raspberry Pi**

1. **Check the OS of the Raspberry Pi**

$ cat /etc/os-release

**Output should be somewhat as below:**

PRETTY\_NAME="Debian GNU/Linux 12 (bookworm)"

NAME="Debian GNU/Linux"

VERSION\_ID="12"

VERSION="12 (bookworm)"

VERSION\_CODENAME=bookworm

ID=debian

HOME\_URL="https://www.debian.org/"

SUPPORT\_URL="https://www.debian.org/support"

BUG\_REPORT\_URL="https://bugs.debian.org/"

**This means the Raspberry Pi OS is “bookworm”**

1. **Next, check the architecture of Raspberry Pi**

$ uname -a

**Output should be somewhat as below:**

Linux raspberrypi 6.6.28+rpt-rpi-v8 #1 SMP PREEMPT Debian 1:6.6.28-1+rpt1 (2024-04-22) aarch64 GNU/Linux

**This means the Raspberry Pi architecture is “aarch64”**

1. **Then, check the kernel bit size of Raspberry Pi**

$ getconf LONG\_BIT

**Output should be somewhat as below:**

64

**This means the Raspberry Pi kernel bit size is 64 bit.**

**Install Conda-Forge Mini**

1. **Install *conda-forge mini* as per** [**https://github.com/conda-forge/miniforge?tab=readme-ov-file**](https://github.com/conda-forge/miniforge?tab=readme-ov-file)

Run below commands:

$ wget "https://github.com/conda-forge/miniforge/releases/latest/download/Miniforge3-$(uname)-$(uname -m).sh"

**Accept all**

$ bash Miniforge3-$(uname)-$(uname -m).sh

**Accept all**

After this step, the conda-forge mini should be successfully installed on Raspberry Pi

**[Optional] If you want to prevent Conda from automatically activating the (base) environment every time you open a new terminal, you can run:**

$ conda config --set auto\_activate\_base false

**Directory Formation for Project**

1. **Create a directory under which the project should be developed and then *cd* into it**

$ mkdir smolm

$ cd smolm

**Conda Virtual Environment Formation**

1. **Create the conda virtual environment**

$ conda create -n smolm python=3.10 pip

This creates a conda virtual environment ‘smolm’ for python 3.10 and pip is also installed for the same python 3.10

1. **Then activate the conda environment**

$ conda activate smolm

**To deactivate** the virtual environment for conda do follow:

$ conda deactivate

**PyTorch Installation via Conda-Forge Channel**

1. **Install pytorch as per the pytorch web https://pytorch.org/get-started/locally/**

$ conda install pytorch torchvision torchaudio cpuonly -c pytorch

If ‘torchaudio’ is **NOT available** then just write below:

$ conda install pytorch torchvision cpuonly -c pytorch

1. **Check pytorch version whether it is successfully installed**

**Option 1.**

$ python -c "import torch; print(torch.\_\_version\_\_)"

OR

**Option 2.**

$ python3

$ import torch

$ torch.\_\_version\_\_

**Huggingface Related Libraries Installation via Conda-Forge Channel**

1. **Install ‘huggingface\_hub’ from Conda-Forge**

Use the conda-forge channel to install the huggingface\_hub package:

$ conda install -c conda-forge huggingface\_hub

**[Optional] After installation, you can verify that huggingface\_hub is installed correctly by running:**

$ python -c "from huggingface\_hub import model\_info; print(model\_info('gpt2'))"

This command fetches information about the ‘GPT-2’ model from Hugging Face Hub, ensuring that the package is working as expected.

1. **Install ‘transformers’ via Conda-Forge**

$ conda install -c conda-forge transformers

**[Optional] Verify the Installation:** To verify that the installation was successful, you can try importing the library in Python:

$ python -c "import transformers; print(transformers.\_\_version\_\_)"

1. **Install ‘accelerate’ via Conda-Forge**

$ conda install -c conda-forge accelerate

**[Optional]** **Verify the Installation:** To verify that the installation was successful, you can try importing the library in Python:

$ python -c "import accelerate; print(accelerate.\_\_version\_\_)"

1. **Install other necessary files via Conda-Forge or pip**

|  |
| --- |
| **Install ‘bitsandbytes’ via pip**  As **bitsandbytes is not** available **for aarch64 platform via Conda-Forge, so use pip3 to install it**  $ pip3 install bitsandbytes  **[Optional]** **Verify the Installation:** To verify that the installation was successful, you can try importing the library in Python:  $ python3 -c "import bitsandbytes as bnb; print(bnb.\_\_version\_\_)" |

**Load the LLM from Huggingface and Running**

1. **Talk to LLM ‘SmolLM-135M-Instruct’**

|  |
| --- |
| from transformers import AutoModelForCausalLM, AutoTokenizer  checkpoint = "HuggingFaceTB/SmolLM-135M-Instruct"  device = "cpu" # for GPU usage or "cpu" for CPU usage  tokenizer = AutoTokenizer.from\_pretrained(checkpoint)  # for multiple GPUs install accelerate and do `model = AutoModelForCausalLM.from\_pretrained(checkpoint, device\_map="auto")`  model = AutoModelForCausalLM.from\_pretrained(checkpoint).to(device)  messages = [{"role": "user", "content": "What is the capital of France."}]  input\_text=tokenizer.apply\_chat\_template(messages, tokenize=False)  print(input\_text)  inputs = tokenizer.encode(input\_text, return\_tensors="pt").to(device)  outputs = model.generate(inputs, max\_new\_tokens=50, temperature=0.2, top\_p=0.9, do\_sample=True)  print(tokenizer.decode(outputs[0])) |

**Where about Local Loaded LLMs**

1. **Where is Model Loaded in Disk?**

Normally it should be loaded below:

/home/pi/.cache/huggingface/hub

**[Optional] To find where the ‘SmolLM-135M-Instruct’ model is loaded on Raspberry Pi run below command:**

$ find / -name "\* SmolLM-135M-Instruct \*" 2>/dev/null

1. **How to Remove the ‘SmolLM-135M-Instruct’ From the Disk?**

Based on the above location do follows:

**Delete Model Directory:**

$ rm -rf /home/pi/.cache/huggingface/hub/models--HuggingFaceTB--SmolLM-135M-Instruct

**Delete Lock File Directory:**

$ rm -rf /home/pi/.cache/huggingface/hub/.locks/models--HuggingFaceTB--SmolLM-135M-Instruct

**Finally, Verify Deletion:**

$ ls /home/pi/.cache/huggingface/hub/models--HuggingFaceTB--SmolLM-135M-Instruct

**It should show below:**

ls: cannot access '/home/pi/.cache/huggingface/hub/models--HuggingFaceTB-- SmolLM-135M-Instruct': No such file or directory

$ ls /home/pi/.cache/huggingface/hub/.locks/models--HuggingFaceTB--SmolLM-135M-Instruct

**It should show below:**

ls: cannot access '/home/pi/.cache/huggingface/hub/.locks/models--HuggingFaceTB-- SmolLM-135M-Instruct ': No such file or directory