Setup of Jetson Nano 4GB

With Unofficially Compiled PyTorch

Document Authored by Arif Iqbal

Document Authored by Arif Iqbal

Host vs. Device

The PC under use for interaction with Jetson Nano in Headless Mode is referred to as the Host in this documentation. Please note that some commands must be executed on the Host PC during the setup process. You will find it mentioned alongside the commands. The Jetson Nano may be referred to as the device.

Host Operating System

Debian 12 (bookworm) has been utilised as the operating system for Host PC for the purpose of this documentation. You may use any other linux OS such as Ubuntu but it might result in some minor changes for the process.

Jetpack

This documentation assumes that the Jetpack Version 4.6.1 is deployed on Jetson Nano. You can acquire the image for the required Jetpack version from the Nvidia website provided below:

Jetpack Version 4.6.1 Image Download

Pytorch & Torch Vision Wheels

You can download the compiled Pytorch and Torch vision in the form of Wheel files from the following link:

Pytorch & Torch Vision

Preparing MicroSD Card

1) Open the Terminal application.
2) Insert your microSD card, then use a command like this to show which disk device was assigned to it:

```
$ sudo dmesg | tail
```

Confirm by matching the specifications of your microSD card to the detected device.
Note: If you have previously burned some image onto your microSD card with dd command, then it may show up in somewhat anomalous form.

```
$ lsblk
```

Document Authored by Arif Iqbal

```
jet@rocket:~$ sudo dmesg | tail
[sudo] password for jet:
[194551.097423] scsi 0:0:0:0: Direct-Access     Generic  Mass-Storage     1.11 PQ: 0 ANSI: 2
[194551.097866] sd 0:0:0:0: Attached scsi generic sg0 type 0
[194551.893757] sd 0:0:0:0: [sda] 124735488 512-byte logical blocks: (63.9 GB/59.5 GiB)
[194551.894074] sd 0:0:0:0: [sda] Write Protect is off
[194551.894082] sd 0:0:0:0: [sda] Mode Sense: 03 00 00 00
[194551.894403] sd 0:0:0:0: [sda] No Caching mode page found
[194551.894405] sd 0:0:0:0: [sda] Assuming drive cache: write through
[194551.901720]  sda: sda1 sda2 sda3 sda4 sda5 sda6 sda7 sda8 sda9 sda10 sda11 sda12 sda13 sda14
[194551.902869] sd 0:0:0:0: [sda] Attached SCSI removable disk
[194552.573954] EXT4-fs (sda1): mounted filesystem with ordered data mode. Quota mode: none.
jet@rocket:~$ lsblk
NAME        MAJ:MIN RM   SIZE RO TYPE MOUNTPOINTS
sda           8:0    1  59.5G  0 disk
├─sda1        8:1    1  59.5G  0 part /media/jet/66181078-3393-4d65-bae1-8e21a434c5b5
├─sda2        8:2    1   128K  0 part
├─sda3        8:3    1   448K  0 part
├─sda4        8:4    1   576K  0 part
├─sda5        8:5    1    64K  0 part
├─sda6        8:6    1   192K  0 part
├─sda7        8:7    1   384K  0 part
├─sda8        8:8    1    64K  0 part
├─sda9        8:9    1   448K  0 part
├─sda10       8:10   1   448K  0 part
├─sda11       8:11   1   768K  0 part
├─sda12       8:12   1    64K  0 part
├─sda13       8:13   1   192K  0 part
└─sda14       8:14   1   128K  0 part
nvme0n1     259:0    0 476.9G  0 disk
├─nvme0n1p1 259:1    0   260M  0 part /boot/efi
├─nvme0n1p2 259:2    0    16M  0 part
├─nvme0n1p3 259:3    0 313.8G  0 part /media/jet/OS
├─nvme0n1p4 259:4    0   9.3G  0 part [SWAP]
└─nvme0n1p5 259:5    0 153.5G  0 part /
jet@rocket:~$
```

3) Use this command to write the zipped SD card image to the microSD card:

```
$ /usr/bin/unzip -p ~/Downloads/jetson_nano_devkit_sd_card.zip |
        sudo /bin/dd of=/dev/sd<x> bs=1M status=progress
```

Note: The downloaded zip file of the jetpack has been placed in the home directory of the user in this example.

```
jet@rocket:~$ /usr/bin/unzip -p ~/jetson-nano-jp461-sd-card-image.zip | sudo /bin/dd of=/dev/sda bs=1M status=progress
13798539264 bytes (14 GB, 13 GiB) copied, 699 s, 19.7 MB/s
0+210816 records in
0+210816 records out
13816037376 bytes (14 GB, 13 GiB) copied, 699.761 s, 19.7 MB/s
jet@rocket:~$
```

4) To ensure that there is no data pending in the buffer for writing to microSD card, run the following command:

```
$ sync
```

Document Authored by Arif Iqbal

```
jet@rocket:~$ sync
jet@rocket:~$
```

5) When the dd command finishes, eject the disk device from the command line:

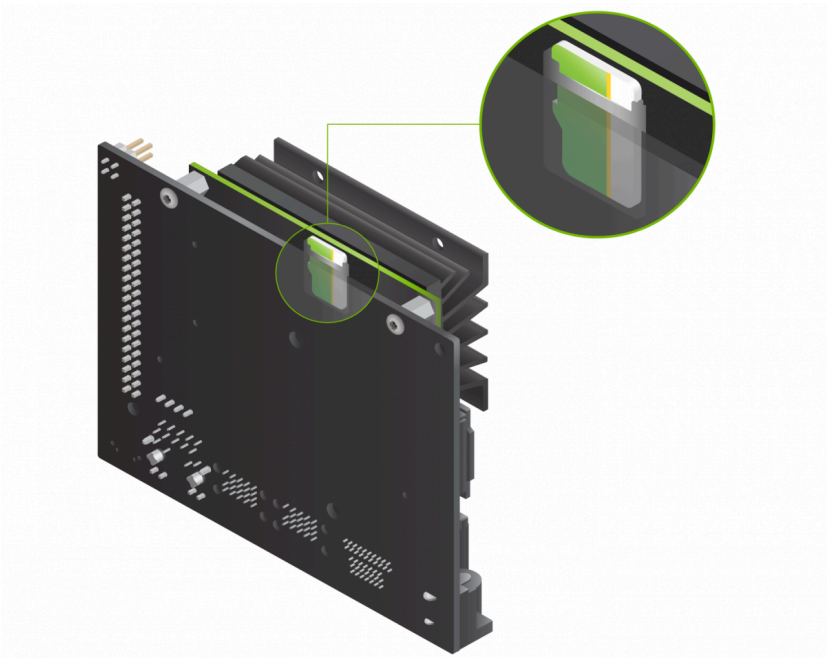
```
$ sudo eject /dev/sd<x>
```

```
jet@rocket:~$ sudo eject /dev/sda
jet@rocket:~$
```

6) Physically remove the microSD card from the computer.

Initial Setup in Headless Mode

1) Unfold the paper stand and place it inside the developer kit box.
2) Insert the microSD card (with system image already written to it) into the slot on the underside of the Jetson Nano module.



3) Set the developer kit on top of the paper stand.
4) Check the Jetson Nano Developer Kit User Guide for location of J48 Power Select Header and J25 Power Jack.
5) Jumper the J48 Power Select Header pins.
6) Connect a DC power supply to the J25 Power Jack. The developer kit will power on automatically.
7) Allow 1 minute for the developer kit to boot.
8) Before connecting to your Jetson developer kit for initial setup, check to see what Serial devices are already shown on your Linux computer.

```
$ sudo dmesg | grep --color 'tty'
```

Document Authored by Arif Iqbal

```
jet@rocket:~$ sudo dmesg | grep --color 'tty'
[    3.510176] systemd[1]: Created slice system-getty.slice - Slice /system/getty.
[    4.166563] cdc_acm 3-1:1.2: ttyACM0: USB ACM device
[11769.666617] cdc_acm 3-1:1.2: ttyACM0: USB ACM device
[90114.116438] cdc_acm 3-1:1.2: ttyACM0: USB ACM device
[95064.016784] cdc_acm 3-1:1.2: ttyACM0: USB ACM device
jet@rocket:~$
```

9) Connect your host computer to the developer kit's Micro-USB port. Run the same command again to find what's newly added.

```
$ sudo dmesg | grep --color 'tty'
```

```
jet@rocket:~$ sudo dmesg | grep --color 'tty'
[    3.510176] systemd[1]: Created slice system-getty.slice - Slice /system/getty.
[    4.166563] cdc_acm 3-1:1.2: ttyACM0: USB ACM device
[11769.666617] cdc_acm 3-1:1.2: ttyACM0: USB ACM device
[90114.116438] cdc_acm 3-1:1.2: ttyACM0: USB ACM device
[95064.016784] cdc_acm 3-1:1.2: ttyACM0: USB ACM device
[200401.272160] cdc_acm 3-2:1.2: ttyACM0: USB ACM device
jet@rocket:~$
```

10) The new serial device is for your Jetson developer kit.
11) Install the Screen program on your Linux computer if it is now already available. For example, use this command to install Screen if you are running Ubuntu.

```
$ sudo apt-get install -y screen
```

12) Use the device name discovered previously as a command line option for the `screen` command.

```
$ sudo screen /dev/ttyACM0 115200
```

13) Once connected to the developer kit, hit the SPACE key on the host PC if the initial setup screen does not appear automatically.
14) Follow the instructions shown on the screen. If you are unsure about any given configuration, leaving it in default settings is recommended.
15) Once, system is configured, it will restart automatically. So, you will have to reconnect with the screen command.
16) To terminate your screen session, Ctrl + a, then k, finally press y on confirmation.

Deployment with Docker

The recommended method for deployment of AI solutions on Jetson Nano is by using the Docker containers. However, Nvidia has stopped releasing the newer versions of Jetpack for Jetson Nano 4GB. So, the officially supported versions of python and pytorch have now been frozen for the last few years. It might not hinder your development process if you are developing all the models yourself as you can cater to the version restriction, but the ability to deploy pretrained latest models (such as YOLOv8) is impacted.

Document Authored by Arif Iqbal

The pytorch and torch vision provided here have been compiled manually as a workaround to this problem. Another benefit for some might be that it allows you to develop AI solutions for the Jetson Nano outside of the docker environment. However, as is the case with any workaround, you might encounter unexpected problems.

Therefore, it is recommended that you stick with the official docker images provided by Nvidia unless strictly necessary. Do keep in mind that when working with these images, you might need to build your own custom images on top of these. You will have to do this at the Jetson Nano itself. Because it has a different architecture. So, images built on your PC will not run as is on the device. Unfortunately, the limited processing power of Jetson Nano makes this process extremely tedious.

Setup of Swapfile

1) The Jetson Nano has limited memory. While deploying large models on it, you may run out of memory. It will result in unexpected behaviours such as a frozen system of the device. To avoid such issues, you should use a swapfile.
2) Create a file that will be used as a swap.

```
$ sudo fallocate -l 4G /swapfile
```

Here, we have allocated 4GB for this purpose.

```
nano@jetson:~$ sudo fallocate -l 4G /swapfile
[sudo] password for nano:
nano@jetson:~$
```

3) Write all zeros to this file

```
$ sudo dd if=/dev/zero of=/swapfile bs=1024 count=4194304
                    status=progress
```

Note: count is calculated like so - $(File\ size\ in\ GBs) \times 2^{30} \div (Block\ size,\ bs)$

4) Set the file permissions to 600 to prevent regular users from writing and reading the file.

```
$ sudo chmod 600 /swapfile
```

5) Create a Linux swap area on the file.

```
$ sudo mkswap /swapfile
```

6) Activate the swap file by running the following command.

```
$ sudo swapon /swapfile
```

7) To make the change permanent open the /etc/fstab file.

```
$ sudo vi /etc/fstab
```

8) Paste the following line inside it.

```
/swapfile swap swap defaults 0 0
```

Note: Press ‘i’ to enter insert mode. Paste the line and press ‘ESC’ to exit the insert mode. Write ‘:wq’ on your keyboard followed by ‘ENTER’ to save & quit.

9) Verify that the swap is active by using the free command.

```
$ sudo free -h
```

Internet through micro USB Connection

Internet connectivity on the Jetson Nano is required to install multiple packages during the setup process.

1) Run the following command on Host OS

```
$ sudo apt install usbutils ifupdown
```

2) Enable IP Forwarding and NAT on the host OS using following commands:

```
$ sudo sysctl net.ipv4.ip_forward=1
```

```
$ echo "net.ipv4.ip_forward = 1" | sudo tee -a /etc/sysctl.conf
```

```
$ sudo iptables -t nat -A POSTROUTING -o wlo1 -j MASQUERADE
```

Note: the last command assumes that wlo1 is your WiFi interface. If yours is named differently, please update the command accordingly.

3) Connect Jetson Nano to the Host
4) Execute the following command on the device to ensure internet connectivity:

```
$ ping google.com
```

5) If it does not work, then try the following command:

```
$ ping 8.8.8.8
```

6) If this works, then there is likely a DNS configuration issue on the device.

Document Authored by Arif Iqbal

7) Edit the configuration file for DNS

```
$ sudo vi /etc/resolv.conf
```

8) Insert the following line

```
nameserver 8.8.8.8
```

9) Update the configuration using the following command:

```
$ sudo resolvconf -u
```

10) Again, execute the following command on the device to ensure internet connectivity:

```
$ ping google.com
```

Copying Files from Host PC to Jetson Nano

1) On the Host OS, execute the following command.

```
$ scp <File Source at Host> <Jetson Nano
Username>@192.168.55.1:<File Destination at Nano>
```

```
jet@rocket:~$ scp ./torch-1.12.0a0+git67ece03-cp38-cp38-linux_aarch64.whl nano@192.168.55.1:/home/nano/torch-1.12.0a0+git67ece03-cp38-cp38-linux_aarch64.whl
The authenticity of host '192.168.55.1 (192.168.55.1)' can't be established.
ED25519 key fingerprint is SHA256:p5B8dPZuvLGnr6pyiKke+RkfDnH1qNPcEwrYZgxC1as.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.55.1' (ED25519) to the list of known hosts.
nano@192.168.55.1's password:
torch-1.12.0a0+git67ece03-cp38-cp38-linux_aarch64.whl                                                                100%  170MB  34.6MB/s   00:04
jet@rocket:~$ scp ./torchvision-0.13.0a0+da3794e-cp38-cp38-linux_aarch64.whl nano@192.168.55.1:/home/nano/torchvision-0.13.0a0+da3794e-cp38-cp38-linux_aarch64.whl
nano@192.168.55.1's password:
torchvision-0.13.0a0+da3794e-cp38-cp38-linux_aarch64.whl                                                             100%   15MB  33.3MB/s   00:00
jet@rocket:~$
```

Note: the availability of these files in the home directory is required in the later sections.

Copying Files from Jetson Nano to Host PC

1) Install OpenSSH Server on the Host OS

```
$ sudo apt update && sudo apt install openssh-server
```

2) Check SSH Server Status

```
$ sudo systemctl status ssh
```

3) If it is not running, then use the following command

```
$ sudo systemctl start ssh
```

Document Authored by Arif Iqbal

4) To make sure that is starts automatically on boot

```
$ sudo systemctl enable ssh
```

5) If you are using firewall on the host, then allow SSH

```
$ sudo ufw allow ssh
```

Note: Above commands are required only for initial setup.

6) Obtain IP Address of the Host PC

```
$ ip addr show eth0
```

7) Use SCP to transfer file from Jetson Nano

```
$ scp /Jetson/Nano/path/to/file
username@HOST-PC-IP:/destination/path
```

Note: '-r' flag must be used if you wish to transfer directory recursively

Update sources for apt

1) To ensure that you have access to the latest version of libraries available on the repositories, you must run the following command:

```
$ sudo apt update
```

Install Python3.8

1) The pytorch and torch vision used in this documentation is the latest that is supported by the CUDA version of Jetson Nano and it has been compiled with python3.8.
2) By default, the installed jetpack only has python3.6
3) You must install the required version of the python by executing the following command.

```
$ sudo apt install python3.8 python3.8-venv python3-pip python3-dev
```

Setup Virtual Environment

1) It is recommended that the libraries required for each project must be kept separate. This allows us to manage and utilise different versions of the same libraries at the same system without any unnecessary interference and clashes.
2) To implement this separation, the virtual environment is one of the simplest options.
3) To setup a virtual environment with python3.8 as default:

```
$ python3.8 -m venv ~/pytorch-env
```

4) To activate the environment

```
$ source pytorch-env/bin/activate
```

Note: the virtual environment created in this section is used in later sections.

Insert CUDA libraries to the path

1) Availability of the CUDA libraries on the path is paramount. Otherwise, full computing power of the device cannot be put to work, and the installation of subsequent libraries may also suffer from failures.
2) Edit the following configuration file using vi

```
$ vi ~/.bashrc
```

3) Insert the following lines to the end of the file

```
export PATH=/usr/local/cuda/bin:$PATH

export LD_LIBRARY_PATH=/usr/local/cuda/lib64:$LD_LIBRARY_PATH
```

Note: Ensure that the CUDA is installed at the location specified, ‘/usr/local/cuda’. Otherwise, make appropriate changes.

4) To apply change

```
$ source ~/.bashrc
```

5) To verify that the CUDA libraries are accessible now

```
$ nvcc --version
```

Install prerequisite libraries for Pytorch

1) Use apt to install the following libraries

```
$ sudo apt install build-essential cmake

$ sudo apt install libopenblas-base libopenmpi-dev
```

2) Activate the pytorch environment

```
$ source pytorch-env/bin/activate
```

3) Use pip to install the following

```
$ pip3 install cython
```

```
$ pip3 install --upgrade pip setuptools
```

```
$ pip3 install numpy
```

Installation of Pytorch

1) The following instructions assume that the wheel file for pytorch (`torch-1.12.0a0+git67ece03-cp38-cp38-linux_aarch64.whl`) is placed in the home directory of the user at Jetson Nano.

2) Install pytorch using the following command

```
$ pip3 install
~/torch-1.12.0a0+git67ece03-cp38-cp38-linux_aarch64.whl
```

3) Confirm that the installation was successful

```
$ python3 -c "import torch; print(torch.__version__)"
```

Installation of Torch Vision

1) The following instructions assume that the wheel file for torch vision (`torchvision-0.13.0a0+da3794e-cp38-cp38-linux_aarch64.whl`) is placed in the home directory of the user at Jetson Nano.

2) Install torch vision using the following command

```
$ pip3 install
~/torchvision-0.13.0a0+da3794e-cp38-cp38-linux_aarch64.whl
```

3) Confirm that the installation was successful

```
$ python3 -c "import torchvision; print(torchvision.__version__)"
```