

# GPU nodes in Tetralith

## HARDWARE:

- 170 retrofitted thin nodes
- 96 GiB of primary memory (RAM)
- One NVIDIA Tesla T4 GPU (Turing)
- One NVMe SSD scratch disk of ~2TiB

Further reading: [www.nsc.liu.se/systems/tetralith/](http://www.nsc.liu.se/systems/tetralith/)

**TESLA T4 SPECIFICATIONS:**

|                              |            |
|------------------------------|------------|
| compute capability           | 7.5        |
| tensor cores                 | 320        |
| CUDA cores                   | 2560       |
| single precision performance | 8.1 TFLOPS |
| memory                       | 16 GB      |
| power                        | 70 W       |

**PRIMARILY SUITABLE FOR:**

- Machine learning
- Single precision FP (*e.g.* MD)
- Hardware accelerated graphics

Available to all projects with allocations on Tetralith!

# ALLOCATING A GPU NODE

Using `interactive`:

```
1 [x_torra@tetralith2]$ interactive -n 1 -c 32 --gpus-per-task=1 -t 60 -A naiss2024-5-11 --reservation=now
2 salloc: Pending job allocation 22767410
3 salloc: job 22767410 queued and waiting for resources
4 salloc: job 22767410 has been allocated resources
5 salloc: Granted job allocation 22767410
6 salloc: Waiting for resource configuration
7 salloc: Nodes n86 are ready for job
8 [x_torra@n86]$
```

`-n 1 -c 32` or `(-N 1)`

Allocate a complete compute node

`--gpus-per-task=1` or `(--gpus=1)`

Allocates the GPU

`-A "slurm account"`

Only needed if you are included in several projects

`--reservation=now`

Is for short (*max* 60 min.) jobs

Don't use for longer jobs!

*Always allocate a full compute node when using a GPU on Tetralith!*

# ALLOCATING A GPU NODE

Batch script header:

```
#!/bin/bash

#SBATCH --ntasks=1
#SBATCH --cpus-per-task=32
#SBATCH --gpus-per-task=1
#SBATCH --time=24:00:00
#SBATCH --account=naiss2024-5-11
:
```

Here I've used long options (e.g. `--ntasks=1`), but short options (e.g. `-n 1`) also work!

Further reading: [www.nsc.liu.se/support/systems/tetralith-GPU-user-guide/](http://www.nsc.liu.se/support/systems/tetralith-GPU-user-guide/)

```

1 [x_torra@tetralith2]$ interactive -n 1 -c 32 --gpus-per-task=1 -t 60 --reservation=now
2 salloc: Pending job allocation 33885382
3 salloc: job 33885382 queued and waiting for resources
4 salloc: job 33885382 has been allocated resources
5 salloc: Granted job allocation 33885382
6 salloc: Waiting for resource configuration
7 salloc: Nodes n74 are ready for job
8 [x_torra@n74]$ nvidia-smi
9 Mon Nov 11 15:01:49 2024

```

```

10 +-----+
11 | NVIDIA-SMI 560.35.03                Driver Version: 560.35.03          CUDA Version: 12.6          |
12 |-----+-----+-----+-----+-----+-----+
13 | GPU   Name                               Persistence-M | Bus-Id        Disp.A | Volatile Uncorr. ECC |
14 | Fan   Temp   Perf             Pwr:Usage/Cap |      Memory-Usage | GPU-Util  Compute M. |
15 |                                           |              MIG M. |                         |
16 |-----+-----+-----+-----+-----+-----+
17 |    0  Tesla T4                          On          | 00000000:3B:00.0 Off  |          0          |
18 | N/A   29C   P8               10W / 70W | 1MiB / 15360MiB |    0%      Default  |
19 |                                           |                      |                      | N/A
20 +-----+-----+-----+-----+-----+-----+
21
22 +-----+
23 | Processes:
24 | GPU   GI   CI          PID    Type   Process name          GPU Memory
25 |      ID   ID                 |    Type              |                 |      Usage
26 |-----+-----+-----+-----+-----+-----+
27 | No running processes found
28 +-----+

```

# ALLOCATING A GPU NODE FOR GRAPHICS

1. Login with ThinLinc!
2. Allocate a GPU node using `interactive.vgl`
3. Launch GUI with `vglrun gui_name`

```
[x_torra@tetralith2]$ interactive.vgl -t 60 --reservation=now
Enabling VirtualGL mode.
Adding --exclusive option. Note: your project will be charged for full nodes!
Adding --constraint=virtualgl to enable VirtualGL.
Adding --gres=gpu to allocate GPU to job.
Allocating one GPU for the interactive shell to allow accelerated graphics. Note: GPU will not be available from e.g job st
Remember to use "vglrun <application>" to enable accelerated graphics for <application>.
salloc: Pending job allocation 33885503
salloc: job 33885503 queued and waiting for resources
salloc: job 33885503 has been allocated resources
salloc: Granted job allocation 33885503
salloc: Waiting for resource configuration
salloc: Nodes n73 are ready for job
[x_torra@n73]$
```

Further reading: [www.nsc.liu.se/support/graphics/](http://www.nsc.liu.se/support/graphics/)



torbenr@tetralith2.nsc.liu.se - ThinLinc Client

MATLAB R2020b - academic use

19:51 Torben Rasmussen

HOME PLOTS APPS LIVE EDITOR INSERT VIEW

Current Folder: /proj/nsc/users/torbenr/jobs/matlab/CardiacMRI-master/Part02\_Modeling.mlx

### Semantic Segmentation Transfer Learning

We are now ready to actually train our network. Let's set up some training options and get to work. Our network was trained on four NVIDIA® V100 Tensor Core GPUs on the cloud, taking approximately one hour of training time.

```

64 if doTraining
65     options = trainingOptions('sgdm', ...
66         'Momentum', 0.9, ...
67         'InitialLearnRate', 0.0002, ...
68         'L2Regularization', 0.0005, ...
69         'MaxEpochs', 100, ...
70         'MiniBatchSize', 4, ...
71         'Shuffle', 'every-epoch', ...
72         'VerboseFrequency', 100, ...
73         'ValidationData', valds, ...
74         'ValidationPatience', 5, ...
75         'Plots', 'training-progress', ...
76         'ExecutionEnvironment', 'gpu');
77     tic
78     [net, info] = trainNetwork(trains, lgraph, options);
79     toc
80 else
81     imshow(fullfile(prj.RootFolder, "HelperFunctions", "Images", "SegnetTrainingProgressPlot.png"));
82 end

```

Initializing input data normalization.

| Epoch | Iteration | Time Elapsed (hh:mm:ss) | Mini-batch Accuracy | Validation Accuracy | Mini-batch Loss | Validation Loss | Base Learning Rate |
|-------|-----------|-------------------------|---------------------|---------------------|-----------------|-----------------|--------------------|
| 1     | 1         | 00:00:28                | 63.13%              | 63.11%              | 0.9535          | 0.9476          | 0.0002             |
| 1     | 50        | 00:01:02                | 63.59%              | 63.27%              | 0.9687          | 0.9473          | 0.0002             |
| 1     | 100       | 00:01:34                | 63.25%              | 63.47%              | 0.9245          | 0.9468          | 0.0002             |
| 1     | 150       | 00:02:07                | 63.98%              | 63.62%              | 0.9672          | 0.9466          | 0.0002             |
| 2     | 200       | 00:02:38                | 64.04%              | 63.79%              | 0.9513          | 0.9463          | 0.0002             |
| 2     | 250       | 00:03:11                | 63.65%              | 63.94%              | 0.9282          | 0.9461          | 0.0002             |
| 2     | 300       | 00:03:43                | 63.92%              | 64.08%              | 0.9314          | 0.9459          | 0.0002             |
| 3     | 350       | 00:04:15                | 64.06%              | 64.19%              | 0.9385          | 0.9458          | 0.0002             |
| 3     | 400       | 00:04:48                | 64.23%              | 64.32%              | 0.9356          | 0.9456          | 0.0002             |
| 3     | 450       | 00:05:20                | 64.34%              | 64.45%              | 0.9456          | 0.9455          | 0.0002             |
| 4     | 500       | 00:05:52                | 64.27%              | 64.56%              | 0.9270          | 0.9454          | 0.0002             |

Workspace: ans 1x1 struct

Command Window: New to MATLAB? See resources for [Getting Started](#).

UTF-8 script Ln 83 Col 1

torbenr@tetralith2.nsc.liu.se - ThinLinc-klient

2024-04-22 15:47 Torben Rasmussen

Applications PyMOL Terminal - torbenr@n73:...

Terminal - torbenr@n73: ~

```
File Edit View Terminal Tabs Help
PyMOL/2.3.0-1-PReSTO      PyMOL/2.5.0-PReSTO-9.0  PyMOL/2.5.0-1-PReSTO    PyMOL/2.5.0-5-PReSTO
PyMOL/2.3.0-2-PReSTO      PyMOL/2.5.0-PReSTO-9.4  PyMOL/2.5.0-2-PReSTO    PyMOL/2.5.0-7-PReSTO
PyMOL/2.5.0-PReSTO-toolchain-foss-2021a  PyMOL/2.5.0-PReSTO-9.6  PyMOL/2.5.0-3-PReSTO

----- /software/sse2/tetralith_el9/modules -----
PyMOL/recommendation (D)  PyMOL/2.5.0-hpc1-gcc-2022a-eb

Where:
D: Default Module
[torbenr@n73 ~]$ module load PyMOL/2.5.0-hpc1-gcc-2022a-eb
[torbenr@n73 ~]$ pymol
Executing 'vglrun pymol.real'
QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/r...'
PyMOL(TM) Molecular Graphics System, Version 2.5.0.
Copyright (c) Schrodinger, LLC.
All Rights Reserved.

Created by Warren L. DeLano, Ph.D.

PyMOL is user-supported open-source software. Although source
code and binaries are freely available, PyMOL is not in the public domain.

If PyMOL is helpful in your work or study, then please vote
for our ongoing efforts to create open and affordable
software by purchasing a PyMOL Maintenance and/or Support
contract.

More information can be found at "http://www.pymol.org".

Enter "help" for a list of commands.
Enter "help <command-name>" for information on a specific
command.

Hit ESC anytime to toggle between text and graphics.

Detected OpenGL version 4.6. Shaders available.
Detected GLSL version 4.60.
OpenGL graphics engine:
GL_VENDOR: NVIDIA Corporation
GL_RENDERER: Tesla T4/PCIe/SSE2
GL_VERSION: 4.6.0 NVIDIA 550.54.14
Detected 32 CPU cores. Enabled multithreaded rendering.

CmdLoad: "" loaded as "apo1".

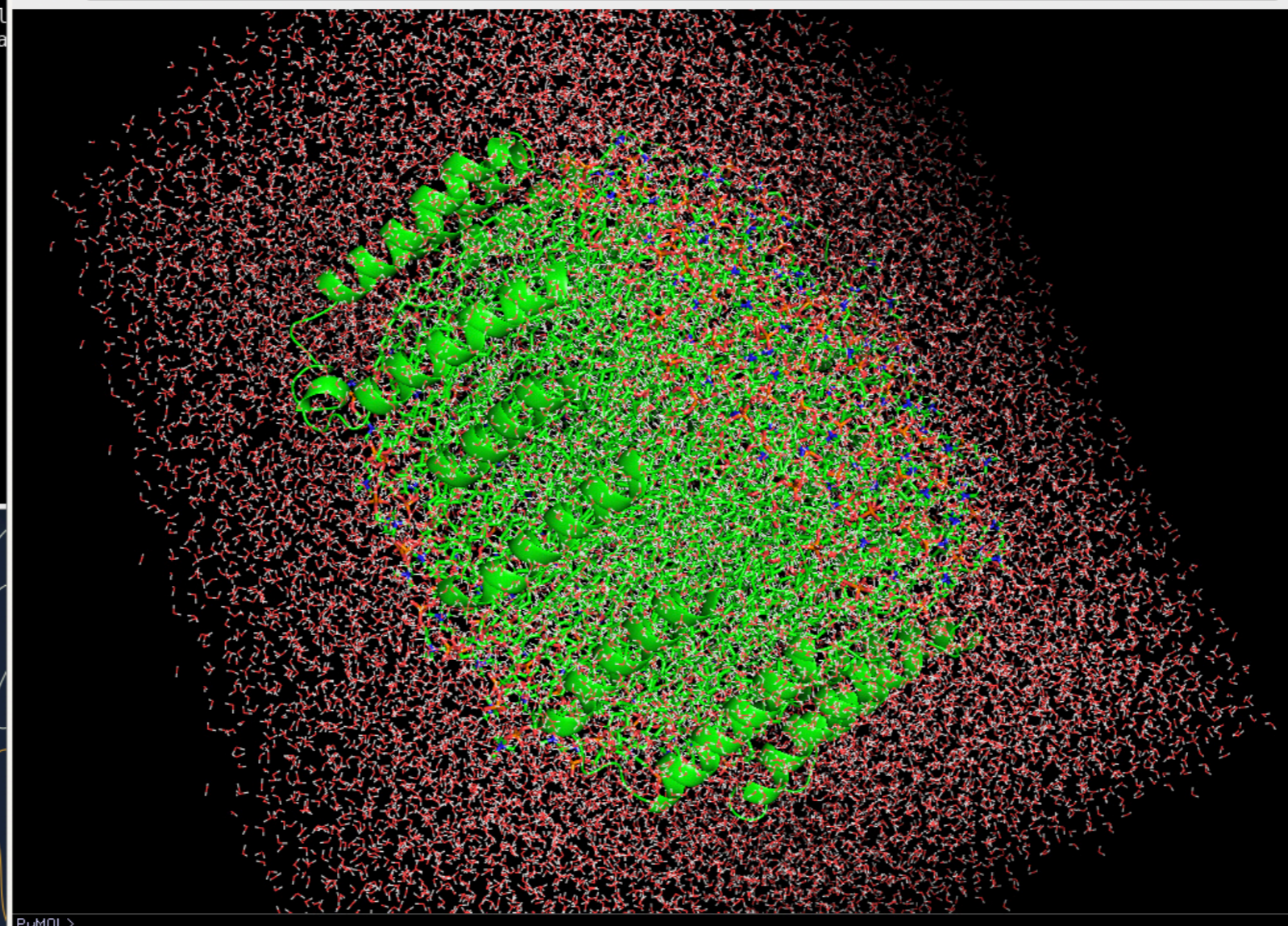
PyMOL>
```

PyMOL (on n73)

```
File Edit Build Movie Display Setting Scene Mouse Wizard Plugin Help
Detected GLSL version 4.60.
OpenGL graphics engine:
GL_VENDOR: NVIDIA Corporation
GL_RENDERER: Tesla T4/PCIe/SSE2
GL_VERSION: 4.6.0 NVIDIA 550.54.14
Detected 32 CPU cores. Enabled multithreaded rendering.

CmdLoad: "" loaded as "apo1".

PyMOL>
```



all A S H L C  
apo1 A S H L C

Mouse Mode 3-Button Viewing  
Buttons L M R Wheel  
& Keys Rota Move MovZ Slab  
Shft +Box -Box Clip MovS  
Ctrl Move PkAt Pk1 MovS2  
CtrlSh Sele Orig Clip MovZ  
SingleClick +/- Cent Menu  
DoubleClick Menu = PkAt  
Selecting Residues  
State 1/1

# USING SINGULARITY/APPTAINER AND NGC

**NGC Catalog:** Software Hub with containers with a range of GPU-accelerated software for NVIDIA GPUs

Building an Apptainer container from an NGC Docker image:

```
[torbenr@tetralith1]$ interactive -n 1 -c 32 --gpus-per-task=1 -t 60 --reservation=now
:
[torbenr@n1112]$ cat ngc_sourceme.txt
export SINGULARITY_DOCKER_USERNAME='$oauthtoken'
export SINGULARITY_DOCKER_PASSWORD="long-pw-private-string"
export APPTAINER_BIND="/proj,/scratch/local,/software:/software:ro"
[torbenr@n1112]$ . ./ngc_sourceme.txt
[torbenr@n1112]$ apptainer build tf20.09_py3.sif docker://nvcr.io/nvidia/tensorflow:20.09-tf2-py3
:
Copying blob 1fa632a1a9de done |
Copying config eb8a6351c4 done |
Writing manifest to image destination
2024/04/22 17:49:24 info unpack layer: sha256:f08d8e2a3ba11bea23cf5c17e8e1c620057412ed05c32d1114640e18d6dd0a43
2024/04/22 17:49:25 info unpack layer: sha256:3baa9cb2483bd9c5329a44d9c2fe72535625bbd4308bca95785dd58e72c06365
:
2024/04/22 17:50:55 info unpack layer: sha256:1fa632a1a9de2caa795450e3871396931851e848d11a227f18e10592a409ffb0
INFO: Creating SIF file...
INFO: Build complete: tf20.09_py3.v4.sif
[torbenr@n1112]$ apptainer run --nv tf20.09_py3.sif
Singularity>
```

# USING SINGULARITY/APPTAINER AND NGC

## Running an Apptainer container:

```

1 [torbenr@tetralith1]$ interactive -n 1 -c 32 --gpus-per-task=1 -t 60 --reservation=now
2 salloc: Pending job allocation 22784449
3 salloc: job 22784449 queued and waiting for resources
4 salloc: job 22784449 has been allocated resources
5 salloc: Granted job allocation 22784449
6 salloc: Waiting for resource configuration
7 salloc: Nodes n1130 are ready for job
8 [torbenr@n1130]$ export APPTAINER_BIND="/proj,/scratch/local,/software:/software:ro"
9 [torbenr@n1130]$ apptainer run --nv tf20.09_py3.v3.sif
10 Usage example: change_mofed_version.sh 4.5-1.0.1
11
12 =====
13 == TensorFlow ==
14 =====
15
16 NVIDIA Release 20.09-tf2 (build 16003717)
17 TensorFlow Version 2.3.0
18
19 Container image Copyright (c) 2020, NVIDIA CORPORATION. All rights reserved.
20 Copyright 2017-2020 The TensorFlow Authors. All rights reserved.
21
22 Various files include modifications (c) NVIDIA CORPORATION. All rights reserved.
23 NVIDIA modifications are covered by the license terms that apply to the underlying project or file.
24
25 Detected MOFED .
26
27 Singularity>

```

## Use the python environment in the container:

```
Singularity> jupyter-notebook --no-browser --ip=n1130
2022-11-15 14:37:00.421476: I tensorflow/stream_executor/platform/default/dso_loader.cc:49] Successfully opened dynamic lib
[I 14:37:03.099 NotebookApp] jupyter_tensorboard extension loaded.
[I 14:37:03.494 NotebookApp] JupyterLab extension loaded from /usr/local/lib/python3.6/dist-packages/jupyterlab
[I 14:37:03.494 NotebookApp] JupyterLab application directory is /usr/local/share/jupyter/lab
[I 14:37:03.712 NotebookApp] [JupyterText Server Extension] NotebookApp.contents_manager_class is (a subclass of) jupyter_text.Text
[I 14:37:03.714 NotebookApp] Serving notebooks from local directory: /proj/nsc/users/torbenr/jobs/ngc
[I 14:37:03.714 NotebookApp] The Jupyter Notebook is running at:
[I 14:37:03.714 NotebookApp] http://n1130:8900/?token=a39669f6c2e66a0e02e98be791ef9ce042346d15e755ac3d
[I 14:37:03.714 NotebookApp] or http://127.0.0.1:8900/?token=a39669f6c2e66a0e02e98be791ef9ce042346d15e755ac3d
[I 14:37:03.714 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 14:37:03.801 NotebookApp]
```

To access the notebook, open this file in a browser:

```
file:///home/torbenr/.local/share/jupyter/runtime/nbserver-195427-open.html
```

Or copy and paste one of these URLs:

```
http://n1130:8900/?token=a39669f6c2e66a0e02e98be791ef9ce042346d15e755ac3d
```

```
or http://127.0.0.1:8900/?token=a39669f6c2e66a0e02e98be791ef9ce042346d15e755ac3d
```

```

Container image Copyright (c) 2020, NVIDIA CORPORATION. All rights reserved.
Copyright 2017-2020 The TensorFlow Authors. All rights reserved.

Various files include modifications (c) NVIDIA CORPORATION. All rights reserved.
NVIDIA modifications are covered by the license terms that apply to the underlying project or file.

Detected MOFED .

Singularity> jupyter-notebook --no-browser --ip=n1130
2022-11-15 14:37:00.421476: I tensorflow/stream_executor/platform/default/dso_loader.cc:49] Successfully opened dynamic library libcudart.so.11.0
[I 14:37:03.099 NotebookApp] jupyter_tensorboard extension loaded.
[I 14:37:03.494 NotebookApp] JupyterLab extension loaded from /usr/local/lib/python3.6/dist-packages/jupyterlab
[I 14:37:03.494 NotebookApp] JupyterLab application directory is /usr/local/share/jupyter/lab
[I 14:37:03.712 NotebookApp] [Jupyter Server Extension] NotebookApp.contents_manager_class is (a subclass of)
jupyterlab.TextFileContentsManager already - OK
[I 14:37:03.714 NotebookApp] Serving notebooks from local directory: /proj/nsc/users/torbenr/jobs/ngc
[I 14:37:03.714 NotebookApp] The Jupyter Notebook is running at:
[I 14:37:03.714 NotebookApp] http://n1130:8900/?token=a39669f6c2e66a0e02e98be791ef9ce042346d15e755ac3d
[I 14:37:03.714 NotebookApp] or http://127.0.0.1:8900/?token=a39669f6c2e66a0e02e98be791ef9ce042346d15e755ac3d
[I 14:37:03.714 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 14:37:03.801 NotebookApp]

To access the notebook, open this file in a browser:
file:///home/torbenr/.local/share/jupyter/runtime/nbserver-195427-open.html
Or copy and paste one of these URLs:
http://n1130:8900/?token=a39669f6c2e66a0e02e98be791ef9ce042346d15e755ac3d
or http://127.0.0.1:8900/?token=a39669f6c2e66a0e02e98be791ef9ce042346d15e755ac3d
[I 14:38:34.347 NotebookApp] 302 GET /?token=a39669f6c2e66a0e02e98be791ef9ce042346d15e755ac3d (10.24.254.11) 2.62ms
[I 14:38:47.394 NotebookApp] 302 GET /notebooks/Lab1/images/DLI_Header.png (10.24.254.11) 3.58ms
[I 14:38:48.059 NotebookApp] 302 GET /notebooks/Lab1/images/SGD1.png (10.24.254.11) 4.14ms
[I 14:38:48.074 NotebookApp] 302 GET /notebooks/Lab1/images/SGD2.png (10.24.254.11) 3.94ms
[I 14:38:48.091 NotebookApp] 302 GET /notebooks/Lab1/images/SGD3.png (10.24.254.11) 4.02ms
[I 14:38:48.681 NotebookApp] Kernel started: 59f51f23-a5a7-4ca4-8eb5-5ae1a1197c30
2022-11-15 14:39:10.966216: I tensorflow/stream_executor/platform/default/dso_loader.cc:49] Successfully opened dynamic library libcudart.so.11.0

```

01\_Notebook\_Neural\_Network\_Optimization - Jupyter Notebook — Mozilla Firefox

n1130:8900/notebooks/Lab1/01\_Notebook\_Neural\_Network\_Optimization.ipynb

jupyter 01\_Notebook\_Neural\_Network\_Optimization Last Checkpoint: 12/02/2020 (unsaved changes) Logout

Python 3

```

In [3]: # This section generates the training dataset as defined by the variables in the section above.
x = np.random.uniform(0, 10, n_samples)
y = np.array([w_gen * (x + np.random.normal(loc=mean_gen, scale=std_gen, size=None)) + b_gen for x in x])

In [4]: # Plot our randomly generated dataset
plt.close()
plt.plot(x, y, 'go')
plt.xlabel("x", size=24)
plt.ylabel("y", size=24)
plt.tick_params(axis='both', labels=16)
plt.tight_layout()
plt.show()

```

Figure 1

Defining the model

Regardless of the complexity of the machine learning problem the process of code development consists of:

- Creating a definition of the model
- Defining the loss (cost) function that will guide our training process. The loss function is effectively a definition of success that informs our optimization

## Monitoring GPU usage:

```

1 [torbenr@tetralith1]$ interactive -N 1 --gpus=1 -t 60 --reservation=now
2 :
3 [torbenr@n75]$ tmux
4 # ^b c
5 [torbenr@n75]$ aptainer run --nv -B $PWD:/host_pwd --pwd /host_pwd relion_3.1.3.sif ./ngc_run_relion.sh
6 # ^b 0
7 [torbenr@n75]$ nvidia-smi
8 Mon Apr 24 15:17:39 2023
9 +-----+
10 | NVIDIA-SMI 515.105.01    Driver Version: 515.105.01    CUDA Version: 11.7    |
11 |-----+-----+-----+
12 | GPU   Name               Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
13 | Fan  Temp  Perf    Pwr:Usage/Cap|      Memory-Usage | GPU-Util  Compute M. |
14 |                                           |              MIG M. |
15 |-----+-----+-----+
16 |    0   Tesla T4              On          | 00000000:3B:00.0 Off  |
17 | N/A   40C    P0     34W / 70W | 233MiB / 15360MiB |      0%      Default |
18 |                                           |                      N/A |
19 |-----+-----+-----+
20
21 +-----+
22 | Processes:
23 | GPU   GI    CI          PID    Type    Process name          GPU Memory
24 |      ID    ID              |                 |           Usage      |
25 |-----+-----+-----+
26 |    0   N/A  N/A       66383     C   relion_refine_mpi     103MiB
27 |    0   N/A  N/A       66384     C   relion_refine_mpi     103MiB
28 |    0   N/A  N/A       66392     C   nvidia-cuda-mps-server 23MiB
29 |-----+-----+-----+

```

## Monitoring GPU usage: (load)

```
[torbenr@n75]$ nvidia-smi dmon
# gpu      pwr gtemp mtemp   sm   mem   enc   dec   mclk  pclk
# Idx      W    C     C     %    %    %    %    MHz  MHz
  0        68  67   -    94   1    0    0  5000 1155
  0        71  67   -    83   1    0    0  5000 1230
  0        67  67   -    87   1    0    0  5000 1170
  0        69  67   -    94   1    0    0  5000 1170
  0        68  67   -    93   1    0    0  5000 1155
  0        65  67   -    94   1    0    0  5000 1155
  0        69  67   -    81   1    0    0  5000 1200
  0        71  67   -    95   1    0    0  5000 1140
  0        72  67   -    89   1    0    0  5000 1140
  0        65  67   -    94   1    0    0  5000 1140
  0        70  67   -    93   1    0    0  5000 1170
  0        69  67   -    93   1    0    0  5000 1185
  :
```