DS-GA 1006 Capstone

Lab Session 3

Instructors: Julia Kempe
Mark Ho
Najoung Kim
Wenda Zhou

TA: Wenxin Zhang (wz2164@nyu.edu)

Lab Session 3

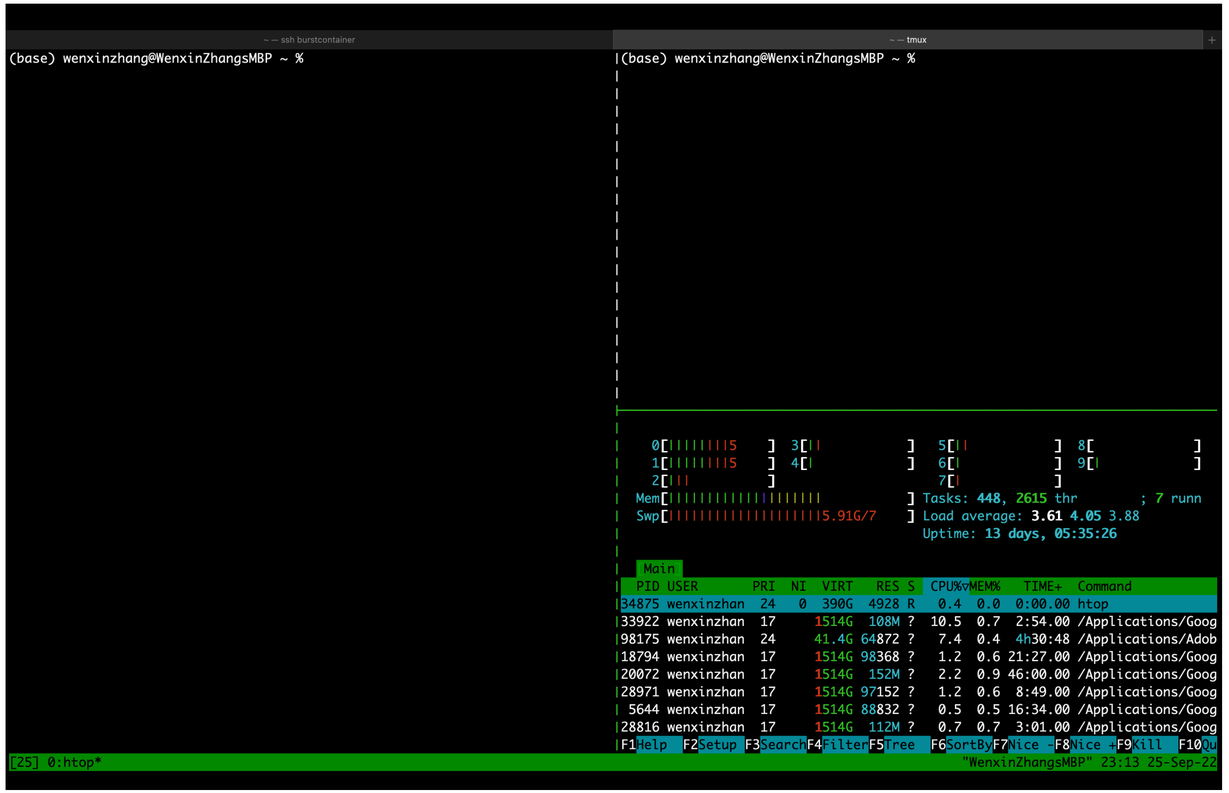
- HW2 - NLP-based problem
  - Overview
  - Tmux
  - Setup
    - Keep your burst instance alive using Slurm
    - Build the overlays & Add the packages
    - Start singularity instance & Connect to instance from VSCode
    - Using Notebooks from VSCode
  - Train the provided network
    - 20 epochs | mixed precision training | 2 GPUs
    - Nvidia-smi
    - Tensorboard
  - Testing
    - Smoke test

Overview

- Huggingface: fine-tune a pretrained model
- Datasets: Yelp review datasets
  - Mainly used for text classification: given the text, predict the sentiment
- Load the dataset & Create a smaller subset of the full dataset to fine-tune on
- Text Preprocess: Tokenize the text using the pretrained BERT base model
- Train the provided network with PyTorch Trainer
  - 20 Epochs
  - Mixed Precision
  - 2 GPUs

Tmux

- Cheatsheet
- Enables you to switch easily between several problem in one terminal; keeps your session alive
- Install tmux through conda
  - conda install -c conda-forge tmux
- Start a new session
  - tmux | tmux new -s mysession
- Kill session
  - tmux kill-session -t mysession
- Split pane
  - (Horizontally) Ctrl + b + “
  - (Vertically) Ctrl + b + %
  - (Switch to pane to the direction) Ctrl + b + ↑, ↓ , →, ←
  - (Change the size of the panes) Ctrl + b + esc + ↑, ↓ , →, ←



3.1 Setup

Following the instructions in `homework/nlp/README.md`, create the overlays containing the required packages for running the homework. Start a singularity instance with the container, and connect VSCode to your instance. Open the `homework/nlp/data.ipynb` notebook and run the existing code to display samples from the dataset. How many reviews are there in total in the dataset? Write code in a new cell to compute this quantity. Take a screenshot of the notebook with the review example and the total number of reviews displayed.

Setup

Request GCP instance using Slurm

Slurm

- A cluster management and job scheduling system for Linux clusters, through which we interact with the Greene clusters and the GCP
  - Allocates access to compute nodes to users
  - Provides framework for starting, executing, and monitoring work (parallel job)
  - Arbitrates contention for resources by managing a queue of pending work
- Commands
  - srun: run jobs interactively
  - sbatch *.sh: queue jobs using a bash scripts
  - squeue -u $USER: reports the state of jobs
  - scancel $jobid: cancel pending or running jobs

```
#!/bin/bash
#
#SBATCH --job-name=request_burst_instance
#SBATCH --account=ds_ga_1006_001-2022fa
#SBATCH --partition=n1c16m96-v100-2
#SBATCH --gres=gpu:v100:2
#SBATCH --time=8:00:00


sleep 8h
```

- Account: ds_ga_1006_001-2022fa
- partition: n1c16m96-v100-2
  - Machine: 16 CPU, 96 GB memory, 2 V100 GPU
- GPUs: --gres=gpu:v100:2
  - --gres=gpu:type:count
- Time: 8:00:00
  - Run time of the machine: 8 hours
  - Maximum: 24 hours
- Remember to cancel the running jobs after work

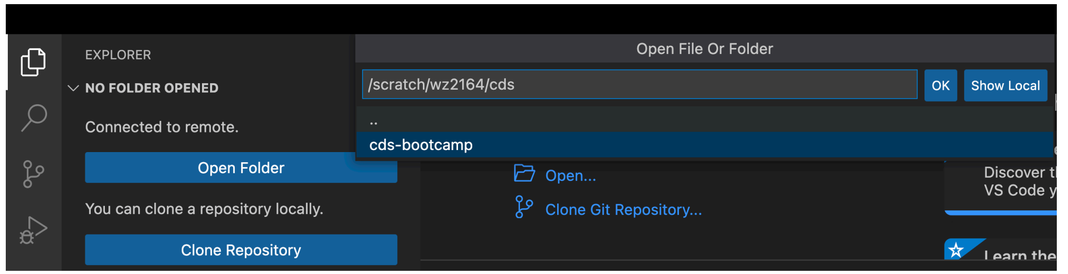
```
(base) [wz2164@log-burst ~]$ sbatch sleep.sh
Submitted batch job 89887
(base) [wz2164@log-burst ~]$ squeue -u wz2164
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
             89887 n1c16m96- request_   wz2164 CF       0:03      1 b-17-1
```

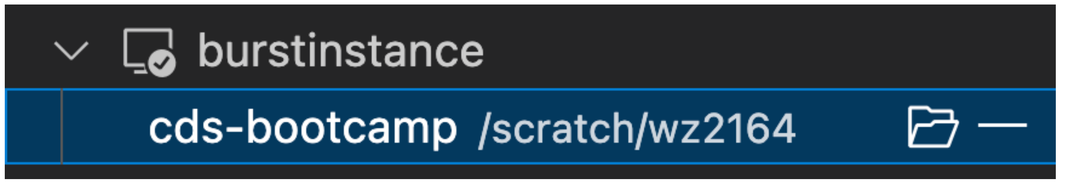
Setup

Build the Overlays & Add the packages

- Set the directories that store overlays, container images on GCP
- Copy the original overlay to the current working directory; unzip & rename the overlay for later use
- Run singularity container
  - Deactivate binding of $HOME directory; bind $HOME/.ssh to make the directory appear inside the singularity container
  - Create base overlay: Obtain a minimal new conda environment at the location /ext3/conda/bootcamp
  - Create package overlay: Package up additional packages that we need (e.g. datasets, transformers, pytorch-lightning, hydra-core, omegaconf)

```
bash-4.4$ git clone https://github.com/wendazhou/cds-bootcamp.git
Cloning into 'cds-bootcamp'...
remote: Enumerating objects: 348, done.
remote: Counting objects: 100% (104/104), done.
remote: Compressing objects: 100% (85/85), done.
remote: Total 348 (delta 37), reused 59 (delta 14), pack-reused 244
Receiving objects: 100% (348/348), 83.50 KiB | 1.21 MiB/s, done.
Resolving deltas: 100% (136/136), done.
```






```
bash-4.4$ pwd
/scratch/wz2164/cds-bootcamp
bash-4.4$ ls
doc  homework  lecture1  lecture2  lecture3  lecture4  LICENSE  README.md
bash-4.4$ cd homework/nlp/
bash-4.4$ ls
bootcamp  data.ipynb  README.md  scripts  setup.py  start_singularity_instance.sh  tests
bash-4.4$ ./scripts/create_base_overlay.sh
Extracting base package overlay
Cloning base packages into overlay
Source:      /opt/conda
Destination: /ext3/conda/bootcamp
The following packages cannot be cloned out of the root environment:
 - conda-forge/linux-64::conda-4.13.0-py38h578d9bd_1
 - conda-forge/linux-64::conda-build-3.21.9-py38h578d9bd_1
Packages: 138
Files: 44425
```

```
bash-4.4$ ls
bootcamp  data.ipynb  overlay-base.ext3  README.md  scripts  setup.py  start_singularity_instance.sh  tests
bash-4.4$ ./scripts/create_package_overlay.sh
Extracting additional package overlay
Installing additional packages
```

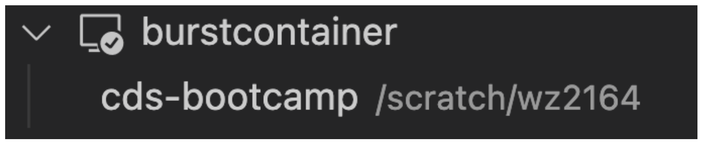
Setup

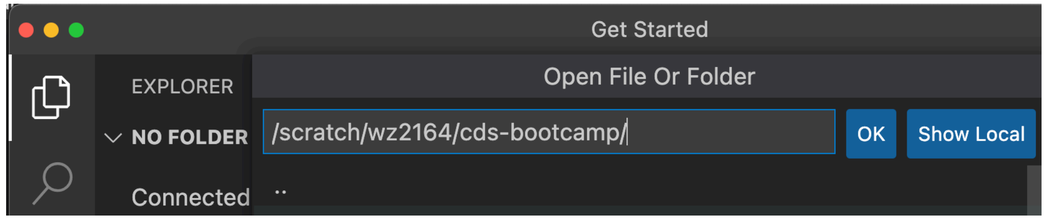
Start singularity instance

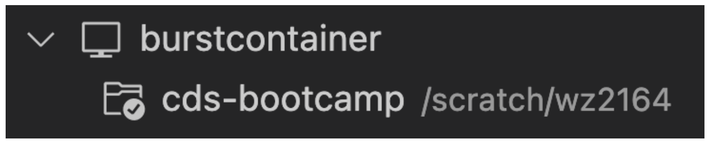
```
bash-4.4$ ls
bootcamp  data.ipynb  overlay-base.ext3  overlay-packages.ext3  README.md  scripts  setup.py  start_singularity_instance.sh  tests
bash-4.4$ ./start_singularity_instance.sh
Temporary overlay not found, automatically creating a new one.
INFO:    instance started successfully
bash-4.4$ singularity instance list
INSTANCE NAME    PID       IP    IMAGE
mycontainer      249340          /scratch/wz2247/singularity/images/pytorch_22.08-py3.sif
bash-4.4$
```

```
Host burstinstance burstcontainer
    User wz2164
    HostName b-17-1
    ForwardAgent yes
    ProxyJump greeneburst
    PasswordAuthentication yes
    ChallengeResponseAuthentication no
    StrictHostKeyChecking=No
    # UserHostsKnownFile=/dev/null


Host burstcontainer
    RemoteCommand singularity shell --containall --shell='/bin/bash' instance://mycontainer
    RequestTTY true
```



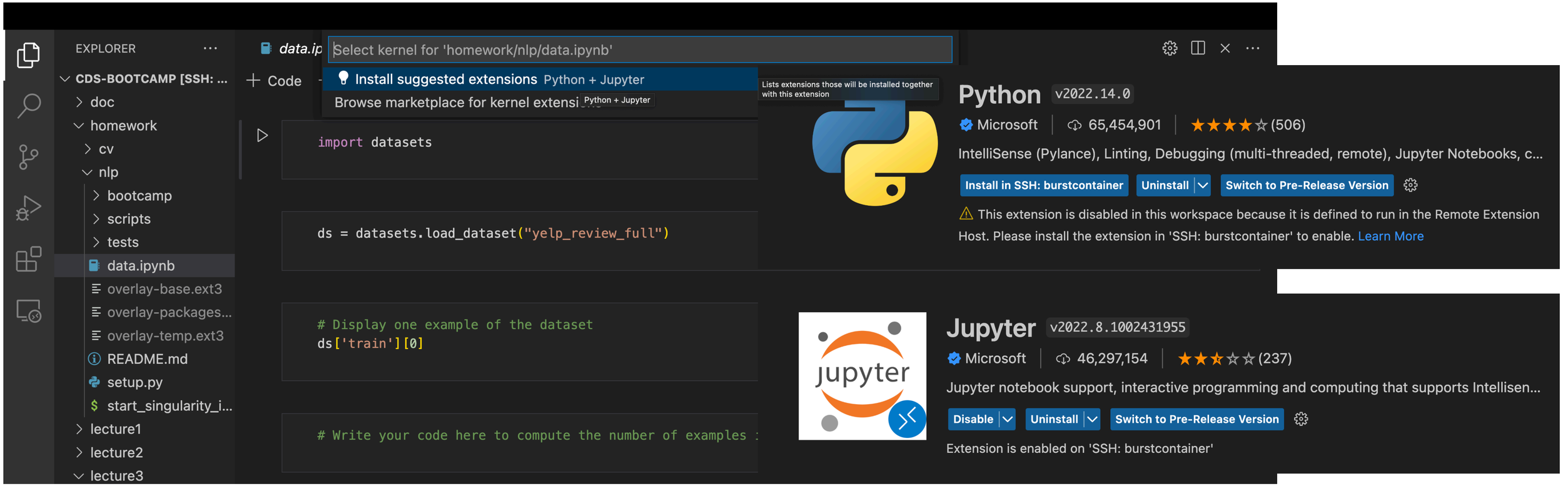


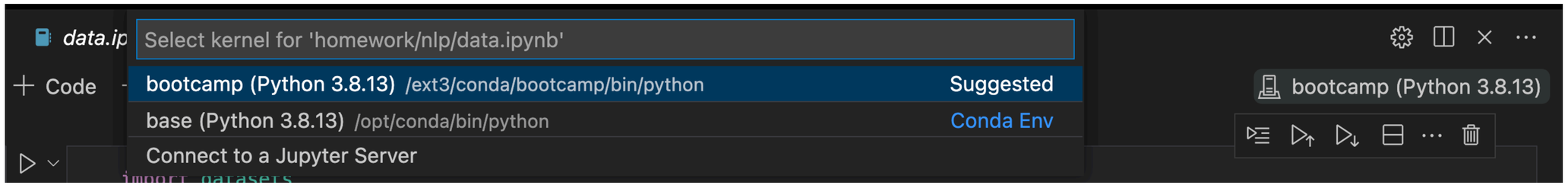


- After creating the required overlays, start singularity instance with the expected binds and overlays
  - Define the directory containing the downloaded datasets and models from the huggingface hub
  - Set the directory that stores container images on GCP; define name of the singularity instance; set the temporary writable overlay
  - Bind the /scratch filesystem, $HOME/.ssh, $PWD; deactivate binding the /home filesystem
  - Overlay with the base packages and additionally installed pacakges

Setup

Using Notebook on VSCode





3.2 Training

We will fine-tune a pre-trained BERT model for the task at hand. Train the provided network on the yelp review dataset for 20 epochs, using mixed precision training and two GPUs. Ensure that your GPU utilization is close to optimal: run `nvidia-smi` and include a screenshot. How many reviews are you processing per second (say how you computed this number, and provide a screenshot with your source information)? Note: this should take less than one hour.

Open tensorboard (by either port forwarding, or within VSCode), and take a screenshot of the accuracy and loss for both training and validation.

How big are the dataset and pre-trained weights (in (giga)bytes)? Include a screenshot of the output of the appropriate command. Note: you may wish to look into the `du` command.

Train the provided Network

_config.py homework/nlp/bootcamp/_config.py

```python
import dataclasses

@dataclasses.dataclass
class BertFineTuningConfig:
    precision: int = 16
    max_epochs: int = 20
    batch_size: int = 16
    gpus: int = 2
```

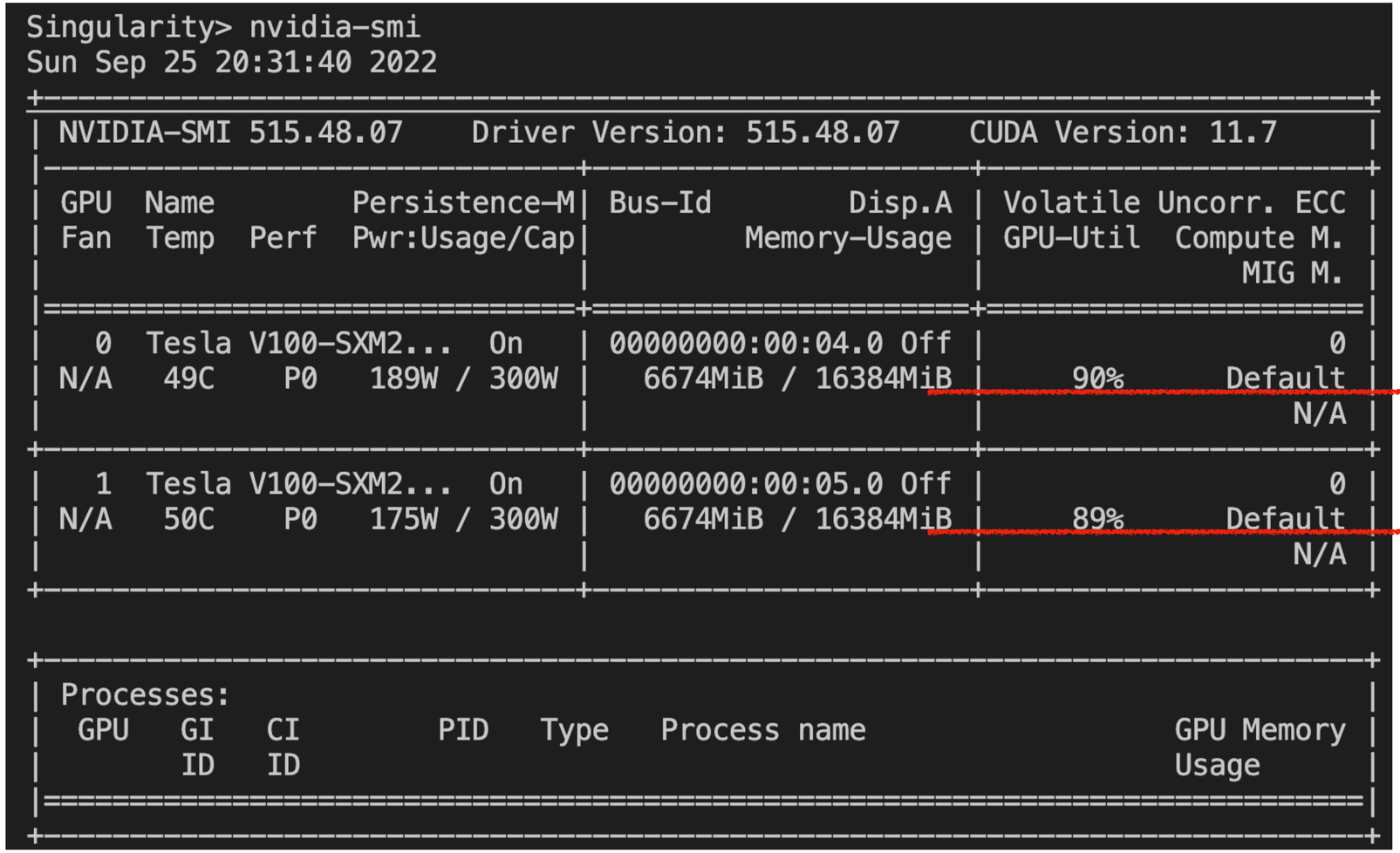
```
(base) wenxinzhang@WenxinZhangsMBP ~ % ssh burstcontainer
Singularity> conda activate /ext3/conda/bootcamp/
(/ext3/conda/bootcamp) Singularity> cd /scratch/wz2164/cds-bootcamp/homework/nlp
(/ext3/conda/bootcamp) Singularity> python -m bootcamp.train
/scratch/wz2164/cds-bootcamp/homework/nlp/bootcamp/train.py:9: UserWarning:
The version_base parameter is not specified.
Please specify a compatability version level, or None.
Will assume defaults for version 1.1
  @hydra.main(config_name='conf', config_path=None)
/ext3/conda/bootcamp/lib/python3.8/site-packages/hydra/_internal/hydra.py:119: Use
 job runtime by default.
See https://hydra.cc/docs/next/upgrades/1.1_to_1.2/changes_to_job_working_dir/ for
  ret = run_job(
Using 16bit native Automatic Mixed Precision (AMP)
GPU available: True (cuda), used: True
TPU available: False, using: 0 TPU cores
IPU available: False, using: 0 IPUs
HPU available: False, using: 0 HPUs
```

- Precision: Mixed precision combines the use of both 32 and 16-bit floating points to reduce memory footprint during model training, resulting in improved performance
- Max_epochs: Setting max_epochs=20 will ensure that training won't happen after 20 epochs

```
  @hydra.main(config_name='conf', config_path=None)
Epoch 0:  43%|                                                  | 1140/2625 [01:56<02:31,  9.82it/s, loss=0.923, v_num=0]
```

```
Epoch 19: 100%|                                                 | 1313/1313 [02:59<00:00,  7.30it/s, los
s=0.124, v_num=0]
```

Nvidia-smi


Singularity> nvidia-smi
Sun Sep 25 20:31:40 2022
NVIDIA-SMI 515.48.07 Driver Version: 515.48.07 CUDA Version: 11.7
GPU Name Persistence-M Bus-Id Disp.A Volatile Uncorr. ECC
Fan Temp Perf Pwr:Usage/Cap Memory-Usage GPU-Util Compute M.
MIG M.
0 Tesla V100-SXM2... On 00000000:00:04.0 Off 0
N/A 49C P0 189W / 300W 6674MiB / 16384MiB 90% Default
N/A
1 Tesla V100-SXM2... On 00000000:00:05.0 Off 0
N/A 50C P0 175W / 300W 6674MiB / 16384MiB 89% Default
N/A
Processes:
GPU GI CI PID Type Process name GPU Memory
ID ID Usage

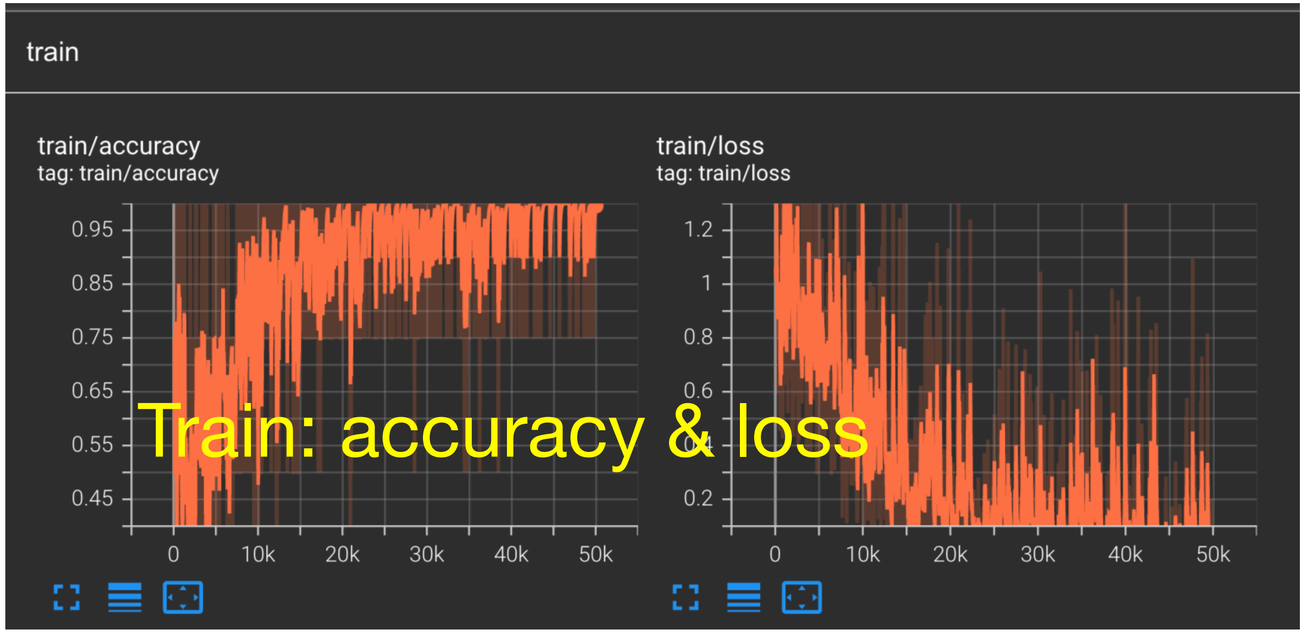
Size of the dataset and pre-trained weights

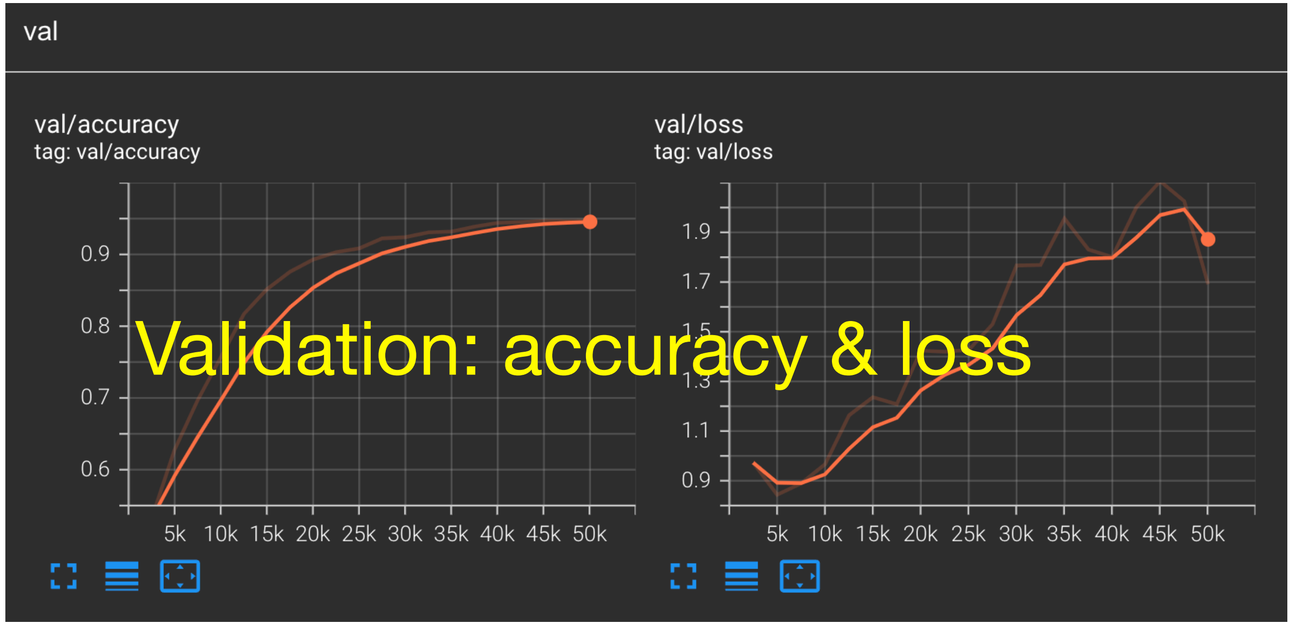
du -h

```
(/ext3/conda/bootcamp) Singularity> pwd
/scratch/wz2164
(/ext3/conda/bootcamp) Singularity> du -h cache/
188M    cache/huggingface/datasets/downloads
3.1G    cache/huggingface/datasets/yelp_review_full/yelp_review_full/1.0.0/e8e18e19d7be9e75642fc66b198abadb116f73599ec89a69ba5dd8d1e57ba0bf
3.1G    cache/huggingface/datasets/yelp_review_full/yelp_review_full/1.0.0
3.1G    cache/huggingface/datasets/yelp_review_full/yelp_review_full
3.1G    cache/huggingface/datasets/yelp_review_full
3.3G    cache/huggingface/datasets
4.0K    cache/huggingface/hub/models--bert-base-cased/refs
417M    cache/huggingface/hub/models--bert-base-cased/blobs
20K     cache/huggingface/hub/models--bert-base-cased/snapshots/a8d257ba9925ef39f3036bfc338acf5283c512d9
20K     cache/huggingface/hub/models--bert-base-cased/snapshots
0       cache/huggingface/hub/models--bert-base-cased/.no_exist/a8d257ba9925ef39f3036bfc338acf5283c512d9
0       cache/huggingface/hub/models--bert-base-cased/.no_exist
417M    cache/huggingface/hub/models--bert-base-cased
417M    cache/huggingface/hub
4.0K    cache/huggingface/modules/datasets_modules/datasets/__pycache__
4.0K    cache/huggingface/modules/datasets_modules/datasets/yelp_review_full/__pycache__
8.0K    cache/huggingface/modules/datasets_modules/datasets/yelp_review_full/e8e18e19d7be9e75642fc66b198abadb116f73599ec89a69ba5dd8d1e57ba0bf/__pycache__
24K     cache/huggingface/modules/datasets_modules/datasets/yelp_review_full/e8e18e19d7be9e75642fc66b198abadb116f73599ec89a69ba5dd8d1e57ba0bf
28K     cache/huggingface/modules/datasets_modules/datasets/yelp_review_full
32K     cache/huggingface/modules/datasets_modules/datasets
4.0K    cache/huggingface/modules/datasets_modules/__pycache__
36K     cache/huggingface/modules/datasets_modules
36K     cache/huggingface/modules
3.7G    cache/huggingface
3.7G    cache/
(/ext3/conda/bootcamp) Singularity>
```

You see the pre-trained weights at cache/huggingface/hub/models--bert-base-cased/ (the directory blobs contains the actual weights). For example, here it shows about ~420MB of weights, which is what we would expect for a Bert model.

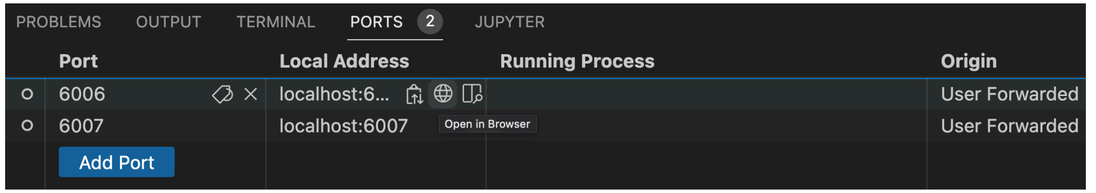
Tensorboard



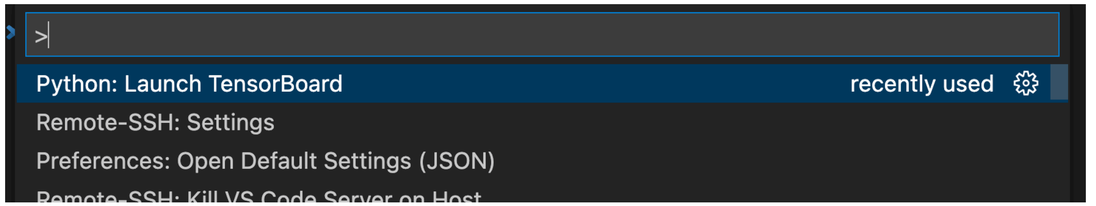


- Tensorboard
  - Provide measurements and visualizations needed during the machine learning workflow
  - Help track experiment metrics like loss and accuracy
  - /output/2022-09-25: called event file into which Tensorboard saves the summary data
- Port Forwarding
  - To launch the visualization server
  - Run tensorboard —logdir=$EVENTS_FOLDER
  - View your visualization in a web browser

```
Singularity> pwd
/scratch/wz2164/cds-bootcamp/homework/nlp
Singularity> tensorboard --logdir /output/2022-09-25
```



- VSCode



Smoke Test

```
# smoke test
self.ds_train = ds["train"].shuffle(seed=42).select(range(8))
self.ds_test = ds["test"].shuffle(seed=42).select(range(8))
```

- Select a small sample of data (e.g. 8 review/rating pairs) and runs it through the model
- Use the save / load functionality to save the subset you made to a small file on disk.
- On a new computer, you would only need that file instead of having to download the entire dataset (~500MB of data)
  - encoded_dataset.save_to_disk(“path/of/my/dataset/directory”)
  - from datasets import load_from_disk
  - reloaded_dataset = load_from_disk(“path/of/my/dataset/directory")
  - reloaded_dataset.set_format(‘torch’): change the format of a column to be compatible with the common data format (type=‘torch’)