

Lab 9: Hyperparameter Grid Search

Stuart Harley

Introduction: In this lab I will be using the command line to execute my model training on the self-driving car model. First, I will use the anaconda environment, and then I will get experience running my training within a singularity container. I will use this opportunity to be able to easily run simultaneous experiments with different models and hyperparameters to find an “optimal” hyperparameter configuration and model structure.

Learning Outcomes:

- Running a model from the command line
- Running a model within a container
- Writing scripts to launch multiple containers and training instances
- Creation and execution of a grid search of models and hyperparameters

Part 1:

Screenshot of command line running the notebook from lab 8 and creating a pdf. (Ran the provided dataset for 5 epochs).

```
[NbConvertApp] Support files will be in self-driving-car_files/
[NbConvertApp] Making directory ./self-driving-car_files
[NbConvertApp] Writing 54672 bytes to ./notebook.tex
[NbConvertApp] Building PDF
[NbConvertApp] Running xelatex 3 times: ['xelatex', './notebook.tex', '-quiet']
[NbConvertApp] Running bibtex 1 time: ['bibtex', './notebook']
[NbConvertApp] WARNING | bibtex had problems, most likely because there were no citations
[NbConvertApp] PDF successfully created
[NbConvertApp] Writing 73578 bytes to self-driving-car.pdf
(base) harleys@dh-node9:~/CS-2300/lab8$
```

Part 2:

Screenshot of command line starting and ending running model.py (run 1: 32bs 3e)

```
Namespace(batch_size='32', data='/data/cs2300/L8/', epochs='3', h5modeloutput='mAIPrime_bs32_e3.h5', model='commaAiModelPrime')
Total Images: 24108
Train samples: 19286
Validation samples: 4822
Epoch 1/3
1/602 [.....] - ETA: 1:00:37 - loss: 0.5519
2/602 [.....] - ETA: 31:12 - loss: 0.5461
3/602 [.....] - ETA: 21:12 - loss: 0.5415
4/602 [.....] - ETA: 16:09 - loss: 0.5269
5/602 [.....] - ETA: 13:01 - loss: 0.5243
601/602 [=====>.] - ETA: 0s - loss: 0.1370
602/602 [=====>.] - ETA: 0s - loss: 0.1369
603/602 [=====] - 77s 127ms/step - loss: 0.1369 - val_loss: 0.1130
dict_keys(['val_loss', 'loss'])
Loss
[0.37780289493024466, 0.22533524005656344, 0.13689242962666298]
Validation Loss
[0.2782703638076782, 0.17272837460041046, 0.11304745078086853]
harleys@dh-mgmt1:~$
```

Screenshot of GPU utilization view from the command line (run 1: 32bs, 3e)

```
harleys@dh-node13:~$ nvidia-smi
Wed Feb 12 10:57:29 2020
```

NVIDIA-SMI 440.33.01				Driver Version: 440.33.01		CUDA Version: 10.2	
GPU	Name	Persistence-M	Bus-Id	Disp.A	Volatile	Uncorr. ECC	
Fan	Temp	Perf	Pwr:Usage/Cap	Memory-Usage	GPU-Util	Compute M.	
0	Tesla T4	On	00000000:60:00:0	Off		0	
N/A	51C	P0	29W / 70W	14622MiB / 15109MiB	12%	Default	
1	Tesla T4	On	00000000:61:00:0	Off		0	
N/A	42C	P8	11W / 70W	0MiB / 15109MiB	0%	Default	
2	Tesla T4	On	00000000:DA:00:0	Off		0	
N/A	32C	P8	10W / 70W	0MiB / 15109MiB	0%	Default	
3	Tesla T4	On	00000000:DB:00:0	Off		0	
N/A	29C	P8	11W / 70W	0MiB / 15109MiB	0%	Default	
Processes:							GPU Memory
GPU	PID	Type	Process name	Usage			
0	60094	C	python	14611MiB			

Screenshot of command line starting and ending running model.py (run 2: 128bs, 1e)

```
Namespace(batch_size='128', data='/data/cs2300/L8/', epochs='1', h5modeloutput='mAiPrime_bs128_e1.h5', model='commaAiModelPrime')
Total Images: 24108
Train samples: 19286
Validation samples: 4822
Epoch 1/1
1/150 [.....] - ETA: 18:37 - loss: 0.5144
2/150 [.....] - ETA: 9:36 - loss: 0.5101
3/150 [.....] - ETA: 6:32 - loss: 0.5053
4/150 [.....] - ETA: 5:00 - loss: 0.5014
5/150 [.....] - ETA: 4:04 - loss: 0.4978
148/150 [=====>.] - ETA: 1s - loss: 0.4322
149/150 [=====>.] - ETA: 0s - loss: 0.4318
150/150 [=====>.] - ETA: 0s - loss: 0.4315
151/150 [=====] - 94s 621ms/step - loss: 0.4312 - val_loss: 0.3818
dict_keys(['val_loss', 'loss'])
loss
[0.43126338478639437]
Validation loss
[0.3817683458328247]
```

Screenshot of GPU utilization view from the command line (run 2: 128 bs, 1e)

```
harleys@dh-node13:~$ nvidia-smi
Wed Feb 12 11:22:38 2020
```

NVIDIA-SMI 440.33.01				Driver Version: 440.33.01		CUDA Version: 10.2	
GPU	Name	Persistence-M	Bus-Id	Disp.A	Volatile	Uncorr. ECC	
Fan	Temp	Perf	Pwr:Usage/Cap	Memory-Usage	GPU-Util	Compute M.	
0	Tesla T4	On	00000000:60:00:0	Off		0	
N/A	31C	P8	9W / 70W	0MiB / 15109MiB	0%	Default	
1	Tesla T4	On	00000000:61:00:0	Off		0	
N/A	47C	P0	28W / 70W	14622MiB / 15109MiB	14%	Default	
2	Tesla T4	On	00000000:DA:00:0	Off		0	
N/A	32C	P8	9W / 70W	0MiB / 15109MiB	0%	Default	
3	Tesla T4	On	00000000:DB:00:0	Off		0	
N/A	29C	P8	9W / 70W	0MiB / 15109MiB	0%	Default	
Processes:							GPU Memory
GPU	PID	Type	Process name	Usage			
1	27135	C	python	14611MiB			

Part 3: Shown: Command line running of 4 jobs running simultaneously. (running in 4 different shells). The jobs are creating models of varying epochs and batch sizes.

```
harleys@dh-mgmt1:~/CS-2300/lab8$ sbatch self-drive.sh
Submitted batch job 3960
harleys@dh-mgmt1:~/CS-2300/lab8$ tail -f slurm-3960.out
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_backend.py:422: The name
tf.global_variables is deprecated. Please use tf.compat.v1.global_variables instead.

2020-02-15 12:43:14.275472: I tensorflow/stream_executor/platform/default/dso_loader.cc:44] Successfully opened
dynamic library libcublas.so.10
2020-02-15 12:43:15.598569: I tensorflow/stream_executor/platform/default/dso_loader.cc:44] Successfully opened
dynamic library libcudnn.so.7
Namespace(batch_size='32', data='/data/cs2300/L8/', epochs='5', h5modeloutput='mAiPrime_bs32_e5.h5', model='comm
aAiModelPrime')
Total Images: 24108
Train samples: 19286
Validation samples: 4822
Epoch 1/5
- 93s - loss: 0.3761 - val_loss: 0.2762
Epoch 2/5
- 79s - loss: 0.2167 - val_loss: 0.1598
Epoch 3/5
- 80s - loss: 0.1260 - val_loss: 0.0926
Epoch 4/5
- 79s - loss: 0.0756 - val_loss: 0.0543
Epoch 5/5
- 83s - loss: 0.0489 - val_loss: 0.0416
dict_keys(['val_loss', 'loss'])
Loss
[0.3761177527191208, 0.2167035766231016, 0.12602665055982432, 0.07561780960773959, 0.04894972893806472]
Validation Loss
[0.2761610150337219, 0.15975023806095123, 0.09260711073875427, 0.05430377274751663, 0.04160543531179428]
```

```
harleys@dh-mgmt1:~/CS-2300/lab8$ sbatch self-drive128-3.sh
Submitted batch job 3962
harleys@dh-mgmt1:~/CS-2300/lab8$ tail -f slurm-2962.out
tail: cannot open 'slurm-2962.out' for reading: No such file or directory
tail: no files remaining
harleys@dh-mgmt1:~/CS-2300/lab8$ tail -f slurm-3962.out
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_backend.py:422: The nam
tf.global_variables is deprecated. Please use tf.compat.v1.global_variables instead.

2020-02-15 12:44:40.938037: I tensorflow/stream_executor/platform/default/dso_loader.cc:44] Successfully opened
dynamic library libcublas.so.10
2020-02-15 12:44:42.258121: I tensorflow/stream_executor/platform/default/dso_loader.cc:44] Successfully opened
dynamic library libcudnn.so.7
Namespace(batch_size='128', data='/data/cs2300/L8/', epochs='3', h5modeloutput='mAiPrime_bs128_e3.h5', model='c
mmaAiModelPrime')
Total Images: 24108
Train samples: 19286
Validation samples: 4822
Epoch 1/3
- 100s - loss: 0.4419 - val_loss: 0.4018
Epoch 2/3
- 83s - loss: 0.3534 - val_loss: 0.3170
Epoch 3/3
- 86s - loss: 0.2850 - val_loss: 0.2582
dict_keys(['val_loss', 'loss'])
Loss
[0.44202426775021264, 0.35346625820551747, 0.2850229186220781]
Validation Loss
[0.4017925560474396, 0.31698259711265564, 0.2581871747970581]
```

```
harleys@dh-mgmt1:~/CS-2300/lab8$ sbatch self-drive8-1.sh
```

```
Submitted batch job 3961
```

```
harleys@dh-mgmt1:~/CS-2300/lab8$ tail -f slurm-3961.out
```

```
Total Images: 24108
```

```
Train samples: 19286
```

```
Validation samples: 4822
```

```
Epoch 1/1
```

```
- 97s - loss: 0.2423 - val_loss: 0.1255
```

```
dict_keys(['val_loss', 'loss'])
```

```
Loss
```

```
[0.2422843946384083]
```

```
Validation Loss
```

```
[0.1255030333995819]
```

```
harleys@dh-mgmt1:~/CS-2300/lab8$ sbatch self-drive1024-10.sh
```

```
Submitted batch job 3963
```

```
harleys@dh-mgmt1:~/CS-2300/lab8$ tail -f slurm-3963.out
```

```
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_backend.py:422: The name tf.global_variables is deprecated. Please use tf.compat.v1.global_variables instead.
```

```
2020-02-15 12:45:13.734644: I tensorflow/stream_executor/platform/default/dso_loader.cc:44] Successfully opened dynamic library libcublas.so.10
```

```
2020-02-15 12:45:15.201267: I tensorflow/stream_executor/platform/default/dso_loader.cc:44] Successfully opened dynamic library libcudnn.so.7
```

```
Namespace(batch_size='1024', data='/data/cs2300/L8/', epochs='10', h5modeloutput='mAiPrime_bs1024_e3.h5', model='commaAiModelPrime')
```

```
Total Images: 24108
```

```
Train samples: 19286
```

```
Validation samples: 4822
```

```
Epoch 1/10
```

```
- 89s - loss: 0.4819 - val_loss: 0.4672
```

```
Epoch 2/10
```

```
- 71s - loss: 0.4538 - val_loss: 0.4422
```

```
Epoch 3/10
```

```
- 73s - loss: 0.4307 - val_loss: 0.4185
```

```
Epoch 4/10
```

```
- 77s - loss: 0.4091 - val_loss: 0.3998
```

```
Epoch 5/10
```

```
- 76s - loss: 0.3889 - val_loss: 0.3776
```

```
Epoch 6/10
```

```
- 78s - loss: 0.3699 - val_loss: 0.3601
```

```
Epoch 7/10
```

```
- 73s - loss: 0.3522 - val_loss: 0.3448
```

```
Epoch 8/10
```

```
- 73s - loss: 0.3356 - val_loss: 0.3262
```

```
Epoch 9/10
```

```
- 73s - loss: 0.3200 - val_loss: 0.3112
```

```
Epoch 10/10
```

```
- 77s - loss: 0.3053 - val_loss: 0.2978
```

```
dict_keys(['val_loss', 'loss'])
```

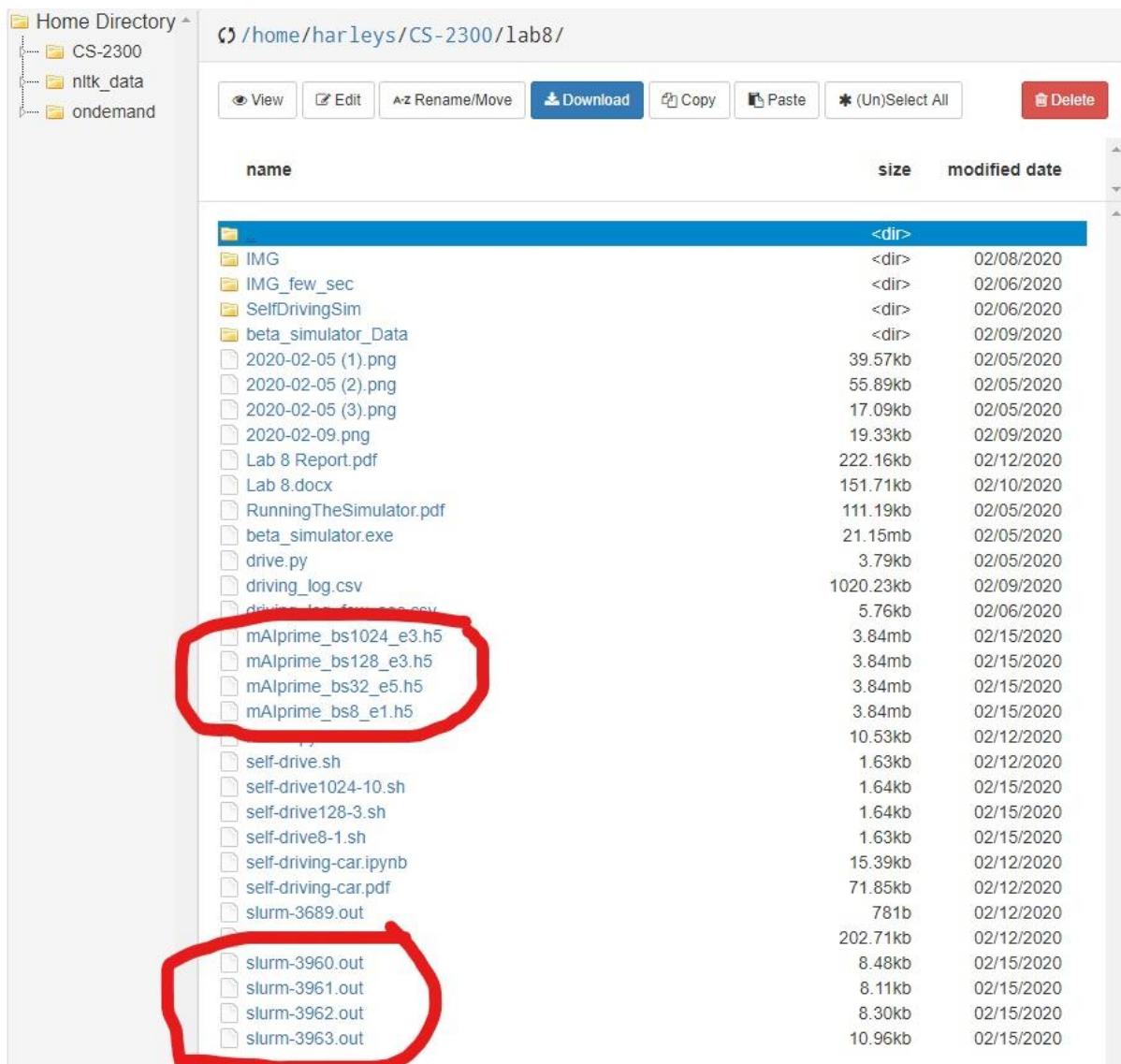
```
Loss
```

```
[0.48204542919769333, 0.4538452469355633, 0.4307852323391768, 0.40920349758071645, 0.38900466064416334, 0.36998696598499303, 0.35224966971406696, 0.3356678389290192, 0.3200207199964313, 0.3053855828189088]
```

```
Validation Loss
```

```
[0.4672061800956726, 0.44219350814819336, 0.4184579849243164, 0.39984458684921265, 0.3775944709777832, 0.3601183593273163, 0.34478843212127686, 0.32615259289741516, 0.31123486161231995, 0.297841876745224]
```

Shown: Output .h5 files and slurm-<jobid>.out files created in file explorer



Part 4: For this part I am comparing 3 different types of models in regard to their performance in the driving simulator. The three models are the CommaAIModelPrime model, the basic model, and the nVidia model. In order to standardize the comparison of the performances of these different models, I used the same hyperparameters to train each one. The hyperparameters I used were a batch size of 32 running for 5 epochs. The loss of the CommaAIModelPrime model was about .045. The loss of the basic model was about 3. The loss of the nVidia model was about .01.

CommaAIModelPrime: Lake Track – The car drove around most of the track mostly in the center of the track. However, on the right turn of the track, the car went off the road and into the lake.

Jungle Track – The car immediately veered left and ran into one of the separators between the two roads.

Basic Model: Lake Track – The car immediately turned to the left and did a full circle before getting stuck on the outside of the track trying to drive back on the track. It is trying to drive in a counterclockwise circle.

Jungle Track – The car immediately turned to the right and got stuck on the mountain to the right of the start of the track. It is trying to drive in a clockwise circle.

nVidia Model: Lake Track – The car drove around the entire track without any issues. For a split second it looked like it might go off onto the dirt part of the track, but then it turned back towards the main track.

Jungle Track – The car veered off to the right and went off the road and got stuck,

Conclusion: It was very obvious that the basic model (that had the worst loss) did the worst as all it tried to do were circles. The next best was the CommaAIModelPrime that made it most of the way around the lake track. And the best model was the nVidia (which had the best loss) which was able to navigate the entire lake track. None of the models were able to navigate the jungle track but with more epochs it would probably be possible, at least for the CommonAI and nVidia models. The basic model may not be a large enough network to be able to perform this task. However, in summary, the higher the loss of the model, the worse the driving is.