See launch_cpu.s and launch_gpu.s for the commands used to produce the information in C2 - C5.

C1

```
Usage:

positional arguments:
  csv_path              Path to the csv file.
  dataset_directory     Path to the dataset directory.

optional arguments:
  -h, --help            show this help message and exit
  --n_workers N_WORKERS
                        The number of workers to use for the
dataloader.
  --enable-cuda         Enable CUDA.
  --optimiser {sgd,sgdwithnesterov,adagrad,adadelta,adam}
                        The optimiser to use.
```

C2

```
Torch version 0.4.1.post2
Running on CPU.
Number of workers: 1; Optimiser: sgd
*Batch computation time includes data loading and nn forward/backward
```

Epoch	**Avg Batch Load Time(s)**	**Avg Batch Computation Time(s)**
1	3.9560	4.2445
2	3.8928	4.1808
3	3.6979	3.9969
4	3.5340	3.8283
5	3.0444	3.3335

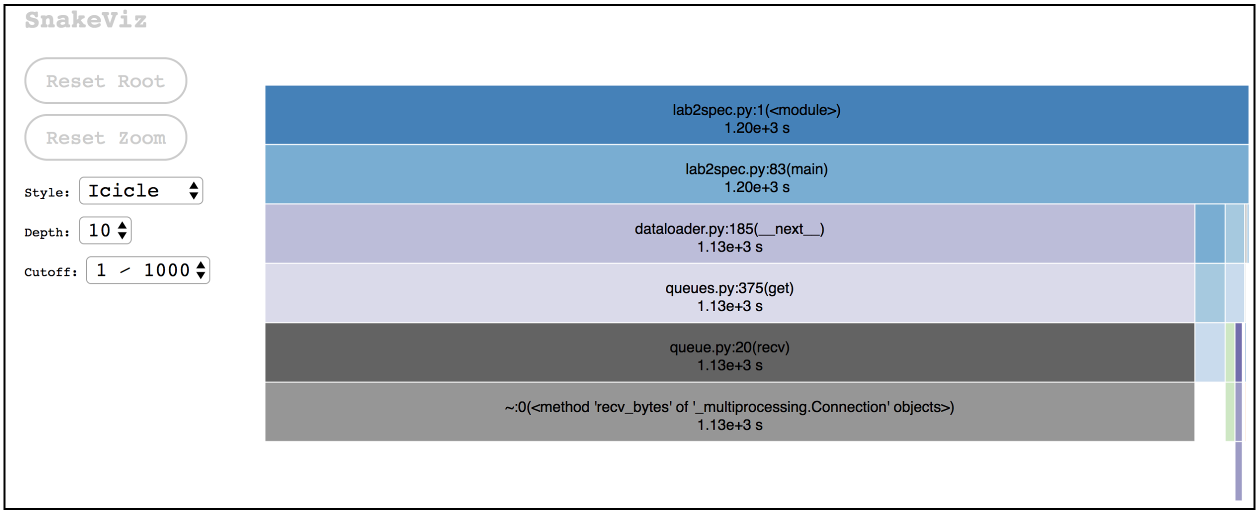
Average time per epoch: 470.0135s

C3

```
Torch version 0.4.1.post2
Running on CPU.
Optimiser: sgd
```

Number of workers	Total Load Time (s)
0	2037.2154
1	2175.0208
2	764.8240
4	262.6946
8	33.7689
12 (best performance)	**21.3648**
16	22.6730
20	29.9539
24	29.0790
28	40.9763
32	43.9927

C4



Style: Icicle

Depth: 10

Cutoff: 1 ⁄ 1000

lab2spec.py:1(<module>)
1.20e+3 s

lab2spec.py:83(main)
1.20e+3 s

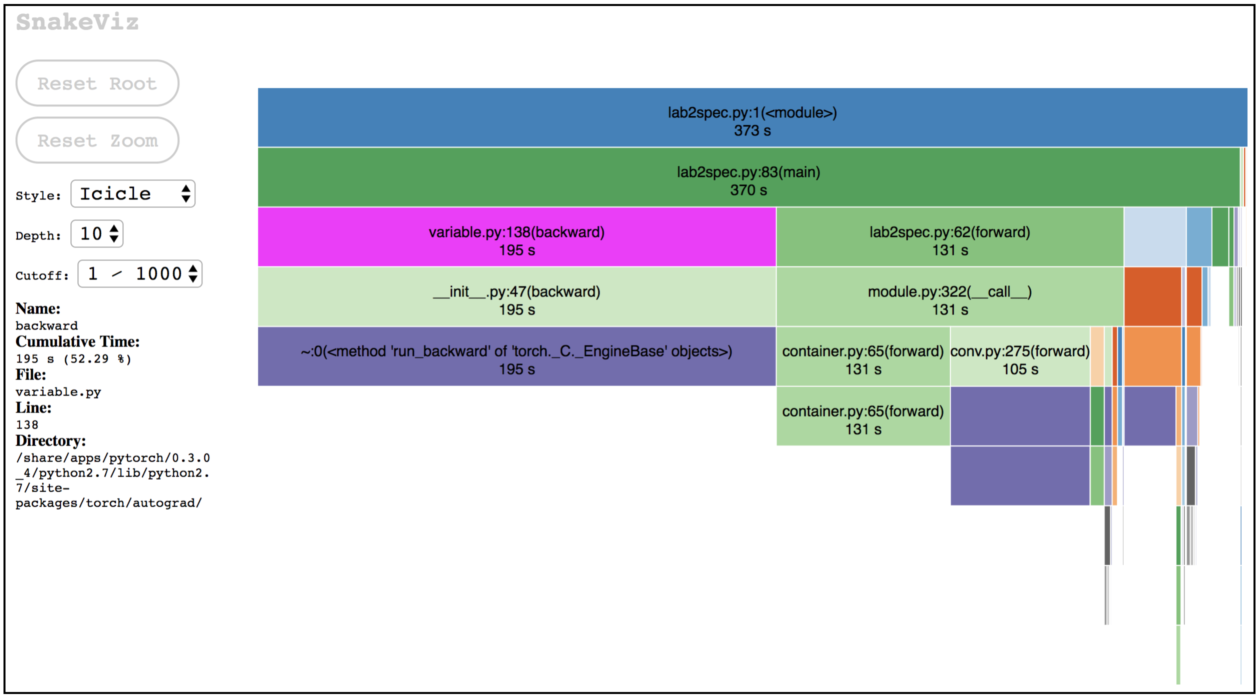
dataloader.py:185(__next__)
1.13e+3 s

queues.py:375(get)
1.13e+3 s

queue.py:20(recv)
1.13e+3 s

~:0(<method 'recv_bytes' of '_multiprocessing.Connection' objects>)
1.13e+3 s

Fig. 1. Snakeviz Icicle plot, 1 worker



Style: Icicle

Depth: 10

Cutoff: 1 / 1000

Name:
backward
Cumulative Time:
195 s (52.29 %)
File:
variable.py
Line:
138
Directory:
/share/apps/pytorch/0.3.0_4/python2.7/lib/python2.7/site-packages/torch/autograd/

Fig. 2. Snakeviz Icicle plot, 12 workers

When using one worker, execution time is dominated by data loading. On the other hand, using 12 workers, execution time is dominated by forward/backward computations.

C5

(1)

Running on CPU.
Number of workers: 1; Optimiser: sgd
Average time per epoch: 470.0135s

Running on GPU
Number of workers: 1; Optimiser: sgd
Average time per epoch: 442.0619s

(2)

```
Torch version 0.4.1.post2
Python version 3.6.4 |Anaconda, Inc.| (default, Jan 16 2018, 18:10:19)
[GCC 7.2.0]
Running on GPU
Number of GPUs: 1

Number of workers: 12; Optimiser: sgd
Epoch   Avg Loss   Avg p@1   Avg p@3
1       0.4430     0.8324    0.5758
2       0.2758     0.9258    0.6380
3       0.2555     0.9258    0.6423
4       0.2493     0.9258    0.6433
5       0.2456     0.9258    0.6437
Average time per epoch: 150.3721s


Number of workers: 12; Optimiser: sgdwithnesterov
Epoch   Avg Loss   Avg p@1   Avg p@3
1       0.4030     0.8653    0.5843
2       0.2681     0.9256    0.6389
3       0.2491     0.9258    0.6417
4       0.2436     0.9258    0.6423
5       0.2398     0.9259    0.6429
Average time per epoch: 118.0847s
```

Number of workers: 12; **Optimiser**: adagrad

Epoch	**Avg Loss**	**Avg p@1**	**Avg p@3**
1	0.2493	0.9205	0.6357
2	0.2210	0.9360	0.6576
3	0.2130	0.9388	0.6663
4	0.2071	0.9398	0.6708
5	0.2033	0.9421	0.6744

Average time per epoch: 136.8236s

Number of workers: 12; **Optimiser**: adadelta

Epoch	**Avg Loss**	**Avg p@1**	**Avg p@3**
1	0.2765	0.9111	0.6285
2	0.2371	0.9261	0.6447
3	0.2318	0.9292	0.6480
4	0.2283	0.9326	0.6504
5	0.2245	0.9350	0.6534

Average time per epoch: 137.5252s

Number of workers: 12; Optimiser: adam

Epoch	**Avg Loss**	**Avg p@1**	**Avg p@3**
1	0.2714	0.9115	0.6230
2	0.2255	0.9353	0.6508
3	0.2175	0.9374	0.6579
4	0.2117	0.9409	0.6638
5	0.2063	0.9406	0.6686

Average time per epoch: 138.8749s