BDDL 2018 HW 02 Parallax

Jiho Choi
(jihochoi@snu.ac.kr)

Overview

The main goal of this project is to get familiar with distributed deep learning training with Vanilla TensorFlow, Horovod, and Parallax.

Environment

- Ubuntu 16.04 on AWS (4 cores, 16G RAM, Without GPU **)
- OpenMPI 3.0.0, Bazel 0.18.1
- TensorFlow v1.11, Horovod 0.11.2, Parallax (cpu_enabled)

  - [setup_env.sh] has all the details of the setting.

Execution (Same as default commands)

```
>> cd BD18F-JihoChoi/hw2_parallax/hw2_rnn
```

TensorFlow

```
>> python rnn_tf.py --ps_hosts=localhost:12345 --worker_hosts=localhost:12346,localhost:12347,localhost:12348 --job_name=ps --task_index=0 --max_steps=500
>> python rnn_tf.py --ps_hosts=localhost:12345 --worker_hosts=localhost:12346,localhost:12347,localhost:12348 --job_name=worker --task_index=0 --max_steps=500
>> python rnn_tf.py --ps_hosts=localhost:12345 --worker_hosts=localhost:12346,localhost:12347,localhost:12348 --job_name=worker --task_index=1 --max_steps=500
>> python rnn_tf.py --ps_hosts=localhost:12345 --worker_hosts=localhost:12346,localhost:12347,localhost:12348 --job_name=worker --task_index=2 --max_steps=500
```

Horovod

```
>> mpirun --mca btl_vader_single_copy_mechanism none --allow-run-as-root -bind-to none -map-by slot -mca orte_base_help_aggregate 0 -x NCCL_DEBUG=INFO -np 2 -H localhost:2 python rnn_horovod.py --max_steps=500
```

Parallax

```
>> python rnn_parallax.py --max_steps=200
```

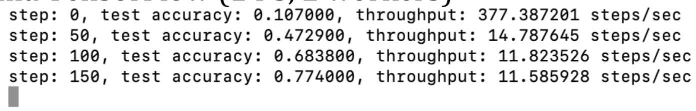
Models

I been implementing the distributed deep learning model with GAN (Generative Adversarial Networks) which generates image by learning the dataset. Unfortunately, I wasn't able to fully debug the GAN model with distributed setting. However, to do the performance evaluation, I switched to RNN model which was not the suited dataset for this project. I understand that this is not quite correct however I wish there is some partial points for the not full correct implementation.

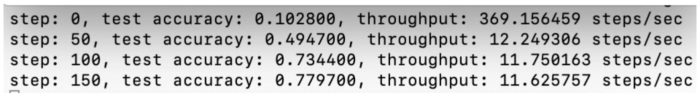
Performances

The performance evaluation is done in Ubuntu 16.04 on AWS (4 virtual cores, 16G RAM) with RNN model **without GPU.** Throughputs of training 300 steps were checked. All experiments were done in a single machine using 3 virtual hosts. Although the performances are quite similar (due to the single machine environment), Vanilla TensorFlow and Parallax were the two best platforms for the training. I believe this is due to the PS architecture's optimization.

- Vanilla TensorFlow (1 PS, 2 workers)

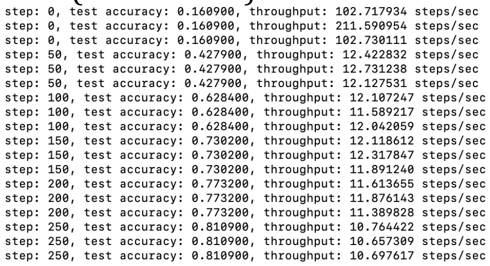


```
step: 0, test accuracy: 0.107000, throughput: 377.387201 steps/sec
step: 50, test accuracy: 0.472900, throughput: 14.787645 steps/sec
step: 100, test accuracy: 0.683800, throughput: 11.823526 steps/sec
step: 150, test accuracy: 0.774000, throughput: 11.585928 steps/sec
```



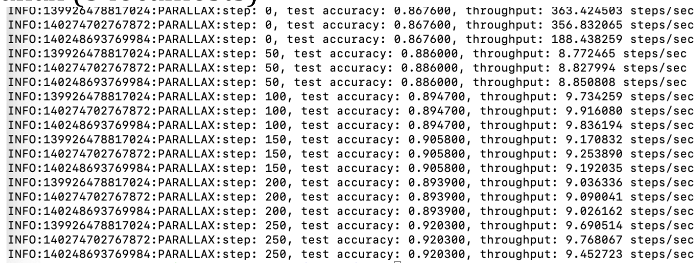
```
step: 0, test accuracy: 0.102800, throughput: 369.156459 steps/sec
step: 50, test accuracy: 0.494700, throughput: 12.249306 steps/sec
step: 100, test accuracy: 0.734400, throughput: 11.750163 steps/sec
step: 150, test accuracy: 0.779700, throughput: 11.625757 steps/sec
```

- Horovod (3 localhosts)



```
step: 0, test accuracy: 0.160900, throughput: 102.717934 steps/sec
step: 0, test accuracy: 0.160900, throughput: 211.590954 steps/sec
step: 0, test accuracy: 0.160900, throughput: 102.730111 steps/sec
step: 50, test accuracy: 0.427900, throughput: 12.422832 steps/sec
step: 50, test accuracy: 0.427900, throughput: 12.731238 steps/sec
step: 50, test accuracy: 0.427900, throughput: 12.127531 steps/sec
step: 100, test accuracy: 0.628400, throughput: 12.107247 steps/sec
step: 100, test accuracy: 0.628400, throughput: 11.589217 steps/sec
step: 100, test accuracy: 0.628400, throughput: 12.042059 steps/sec
step: 150, test accuracy: 0.730200, throughput: 12.118612 steps/sec
step: 150, test accuracy: 0.730200, throughput: 12.317847 steps/sec
step: 150, test accuracy: 0.730200, throughput: 11.891240 steps/sec
step: 200, test accuracy: 0.773200, throughput: 11.613655 steps/sec
step: 200, test accuracy: 0.773200, throughput: 11.876143 steps/sec
step: 200, test accuracy: 0.773200, throughput: 11.389828 steps/sec
step: 250, test accuracy: 0.810900, throughput: 10.764422 steps/sec
step: 250, test accuracy: 0.810900, throughput: 10.657309 steps/sec
step: 250, test accuracy: 0.810900, throughput: 10.697617 steps/sec
```

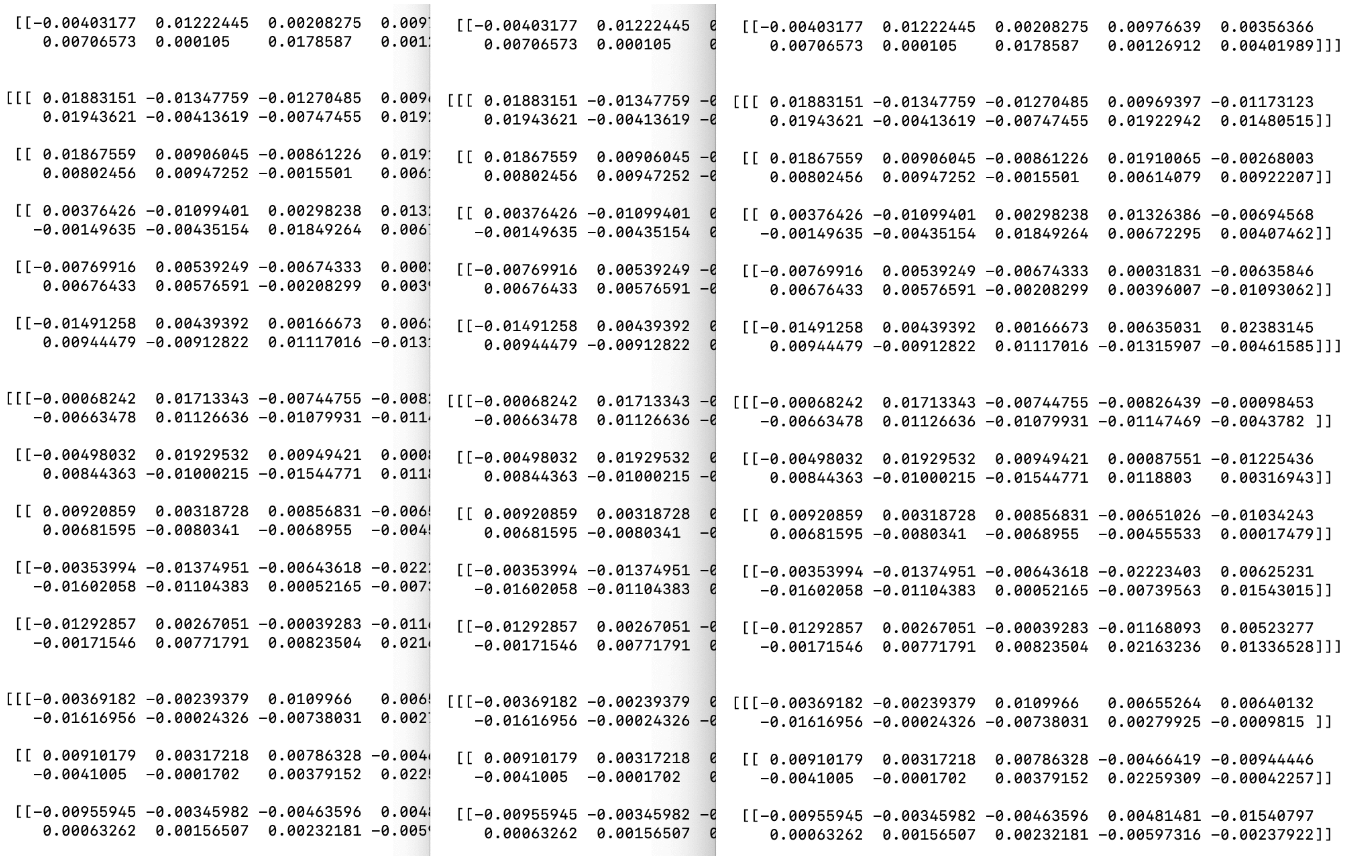
- Parallax (3 localhosts)



```
INFO:139926478817024:PARALLAX:step: 0, test accuracy: 0.867600, throughput: 363.424503 steps/sec
INFO:140274702767872:PARALLAX:step: 0, test accuracy: 0.867600, throughput: 356.832065 steps/sec
INFO:140248693769984:PARALLAX:step: 0, test accuracy: 0.867600, throughput: 188.438259 steps/sec
INFO:139926478817024:PARALLAX:step: 50, test accuracy: 0.886000, throughput: 8.772465 steps/sec
INFO:140274702767872:PARALLAX:step: 50, test accuracy: 0.886000, throughput: 8.827994 steps/sec
INFO:140248693769984:PARALLAX:step: 50, test accuracy: 0.886000, throughput: 8.850808 steps/sec
INFO:139926478817024:PARALLAX:step: 100, test accuracy: 0.894700, throughput: 9.734259 steps/sec
INFO:140274702767872:PARALLAX:step: 100, test accuracy: 0.894700, throughput: 9.916080 steps/sec
INFO:140248693769984:PARALLAX:step: 100, test accuracy: 0.894700, throughput: 9.836194 steps/sec
INFO:139926478817024:PARALLAX:step: 150, test accuracy: 0.905800, throughput: 9.170832 steps/sec
INFO:140274702767872:PARALLAX:step: 150, test accuracy: 0.905800, throughput: 9.253890 steps/sec
INFO:140248693769984:PARALLAX:step: 150, test accuracy: 0.905800, throughput: 9.192035 steps/sec
INFO:139926478817024:PARALLAX:step: 200, test accuracy: 0.893900, throughput: 9.036336 steps/sec
INFO:140274702767872:PARALLAX:step: 200, test accuracy: 0.893900, throughput: 9.090041 steps/sec
INFO:140248693769984:PARALLAX:step: 200, test accuracy: 0.893900, throughput: 9.026162 steps/sec
INFO:139926478817024:PARALLAX:step: 250, test accuracy: 0.920300, throughput: 9.690514 steps/sec
INFO:140274702767872:PARALLAX:step: 250, test accuracy: 0.920300, throughput: 9.768067 steps/sec
INFO:140248693769984:PARALLAX:step: 250, test accuracy: 0.920300, throughput: 9.452723 steps/sec
```

Correctness

I wasn’t able to make the training platform fully deterministic. However the outputs of each experiment show that the accuracies and weights are converge to similar points.



```
[[-0.00403177  0.01222445  0.00208275  0.009
   0.00706573  0.000105    0.0178587   0.001


[[[ 0.01883151 -0.01347759 -0.01270485  0.009
   0.01943621 -0.00413619 -0.00747455  0.019

 [[ 0.01867559  0.00906045 -0.00861226  0.019
   0.00802456  0.00947252 -0.0015501   0.006

 [[ 0.00376426 -0.01099401  0.00298238  0.013
  -0.00149635 -0.00435154  0.01849264  0.006

 [[-0.00769916  0.00539249 -0.00674333  0.000
   0.00676433  0.00576591 -0.00208299  0.003

 [[-0.01491258  0.00439392  0.00166673  0.006
   0.00944479 -0.00912822  0.01117016 -0.013


[[[-0.00068242  0.01713343 -0.00744755 -0.008
  -0.00663478  0.01126636 -0.01079931 -0.011

 [[-0.00498032  0.01929532  0.00949421  0.000
   0.00844363 -0.01000215 -0.01544771  0.011

 [[ 0.00920859  0.00318728  0.00856831 -0.006
   0.00681595 -0.0080341  -0.0068955  -0.004

 [[-0.00353994 -0.01374951 -0.00643618 -0.022
  -0.01602058 -0.01104383  0.00052165 -0.007

 [[-0.01292857  0.00267051 -0.00039283 -0.011
  -0.00171546  0.00771791  0.00823504  0.021


[[[-0.00369182 -0.00239379  0.0109966   0.006
  -0.01616956 -0.00024326 -0.00738031  0.002

 [[ 0.00910179  0.00317218  0.00786328 -0.004
  -0.0041005  -0.0001702   0.00379152  0.022

 [[-0.00955945 -0.00345982 -0.00463596  0.004
   0.00063262  0.00156507  0.00232181 -0.005
```

```
[[-0.00403177  0.01222445  0
   0.00706573  0.000105    0


[[[ 0.01883151 -0.01347759 -0
   0.01943621 -0.00413619 -0

 [[ 0.01867559  0.00906045 -0
   0.00802456  0.00947252 -0

 [[ 0.00376426 -0.01099401  0
  -0.00149635 -0.00435154  0

 [[-0.00769916  0.00539249 -0
   0.00676433  0.00576591 -0

 [[-0.01491258  0.00439392  0
   0.00944479 -0.00912822  0


[[[-0.00068242  0.01713343 -0
  -0.00663478  0.01126636 -0

 [[-0.00498032  0.01929532  0
   0.00844363 -0.01000215 -0

 [[ 0.00920859  0.00318728  0
   0.00681595 -0.0080341  -0

 [[-0.00353994 -0.01374951 -0
  -0.01602058 -0.01104383  0

 [[-0.01292857  0.00267051 -0
  -0.00171546  0.00771791  0


[[[-0.00369182 -0.00239379  0
  -0.01616956 -0.00024326 -0

 [[ 0.00910179  0.00317218  0
  -0.0041005  -0.0001702   0

 [[-0.00955945 -0.00345982 -0
   0.00063262  0.00156507  0
```

```
[[-0.00403177  0.01222445  0.00208275  0.00976639  0.00356366
   0.00706573  0.000105    0.0178587   0.00126912  0.00401989]]]


[[[ 0.01883151 -0.01347759 -0.01270485  0.00969397 -0.01173123
   0.01943621 -0.00413619 -0.00747455  0.01922942  0.01480515]]

 [[ 0.01867559  0.00906045 -0.00861226  0.01910065 -0.00268003
   0.00802456  0.00947252 -0.0015501   0.00614079  0.00922207]]

 [[ 0.00376426 -0.01099401  0.00298238  0.01326386 -0.00694568
  -0.00149635 -0.00435154  0.01849264  0.00672295  0.00407462]]

 [[-0.00769916  0.00539249 -0.00674333  0.00031831 -0.00635846
   0.00676433  0.00576591 -0.00208299  0.00396007 -0.01093062]]

 [[-0.01491258  0.00439392  0.00166673  0.00635031  0.02383145
   0.00944479 -0.00912822  0.01117016 -0.01315907 -0.00461585]]]


[[[-0.00068242  0.01713343 -0.00744755 -0.00826439 -0.00098453
  -0.00663478  0.01126636 -0.01079931 -0.01147469 -0.0043782 ]]

 [[-0.00498032  0.01929532  0.00949421  0.00087551 -0.01225436
   0.00844363 -0.01000215 -0.01544771  0.0118803   0.00316943]]

 [[ 0.00920859  0.00318728  0.00856831 -0.00651026 -0.01034243
   0.00681595 -0.0080341  -0.0068955  -0.00455533  0.00017479]]

 [[-0.00353994 -0.01374951 -0.00643618 -0.02223403  0.00625231
  -0.01602058 -0.01104383  0.00052165 -0.00739563  0.01543015]]

 [[-0.01292857  0.00267051 -0.00039283 -0.01168093  0.00523277
  -0.00171546  0.00771791  0.00823504  0.02163236  0.01336528]]]


[[[-0.00369182 -0.00239379  0.0109966   0.00655264  0.00640132
  -0.01616956 -0.00024326 -0.00738031  0.00279925 -0.0009815 ]]

 [[ 0.00910179  0.00317218  0.00786328 -0.00466419 -0.00944446
  -0.0041005  -0.0001702   0.00379152  0.02259309 -0.00042257]]

 [[-0.00955945 -0.00345982 -0.00463596  0.00481481 -0.01540797
   0.00063262  0.00156507  0.00232181 -0.00597316 -0.00237922]]
```

References

GAN model by Aymeric Damien

- https://github.com/aymericdamien/TensorFlow-Examples/blob/master/examples/3_NeuralNetworks/gan.py

RNN / LSTM

- https://github.com/aymericdamien/TensorFlow-Examples/blob/master/examples/3_NeuralNetworks/gan.py
- https://ratsgo.github.io/natural%20language%20processing/2017/03/09/rnnlstm/