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.

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
```

Execution (Same as default commands)

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 TensorFlow. However, to do the performance evaluation, I switched to RNN model which was not the suited dataset for this project.

```
[[[[ 0.01764052  0.00400157  0.00978738  0.02240893  0.01867558
    -0.00977278  0.00950088 -0.00151357 -0.00103219  0.00410599]]

  [[ 0.00144044  0.01454274  0.00761038  0.00121675  0.00443863
     0.00333674  0.01494079 -0.00205158  0.00313068 -0.00854096]]

  [[-0.0255299   0.00653619  0.00864436 -0.00742165  0.02269755
    -0.01454366  0.00045759 -0.00187184  0.01532779  0.01469359]]

  [[ 0.00154947  0.00378163 -0.00887786 -0.01980796 -0.00347912
     0.00156349  0.01230291  0.0120238  -0.00387327 -0.00302303]]

  [[-0.01048553 -0.01420018 -0.0170627   0.01950775 -0.00509652
    -0.00438074 -0.01252795  0.0077749  -0.01613898 -0.0021274 ]]]


 [[[-0.00895467  0.00386903 -0.00510805 -0.01180632 -0.00028182
     0.00428332  0.00066517  0.00302472 -0.00634322 -0.00362741]]

  [[-0.0067246  -0.00359553 -0.00813146 -0.01726283  0.00177426
    -0.00401781 -0.01630198  0.00462782 -0.00907298  0.00051945]]

  [[ 0.00729091  0.00128983  0.01139401 -0.01234826  0.00402342
    -0.0068481  -0.00870797 -0.0057885  -0.00311553  0.00056165]]

  [[-0.0116515   0.00900827  0.00465662 -0.01536244  0.01488252
     0.01895889  0.0117878  -0.00179925 -0.01070753  0.01054452]]

  [[-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]]
```

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/