model_optimization

December 15, 2022

1

NVIDIA Jetson Raspberry Pi

1.1

```
paddle-bfloat       0.1.7
paddlepaddle-gpu    2.3.2.post112
paddleslim          2.2.1
paddlex             2.1.0
pandas              1.3.5
pexpect             4.8.0
pickleshare         0.7.5
Pillow              9.2.0
numpy               1.21.6
opencv-python       4.6.0.66
openpyxl            3.0.10
scikit-learn        0.23.2
scipy               1.7.3
```

1.2 cuda

```
$ nvcc --version $ nvidia-smi
```

cuda

```
ubuntu20.04
Driver Version: 525.60.11
Cuda compilation tools, release 11.2, V11.2.142
Build cuda_11.2.r11.2/compiler.29558016_0
cuDNN Version: 8.2
```

paddlepaddle-gpu ubuntu20.04

1.3

1. paddlepaddle-gpu:

paddlepaddle-gpu 2.4.1 bug cuda .3.2

conda `conda install paddlepaddle-gpu==2.3.2 cudatoolkit=10.2 --channel https://mirrors.tuna.tsi`

```
pip: python -m pip install paddlepaddle-gpu==2.3.2 -i https://pypi.tuna.tsinghua.edu.cn/simple
```

```
docker:
1.    PaddlePaddle
nvidia-docker pull registry.baidubce.com/paddlepaddle/paddle:2.3.2-gpu-cuda10.2-cudnn7.6-trt7.0
2.     Docker
nvidia-docker run --name paddle -it -v $PWD:/paddle registry.baidubce.com/paddlepaddle/paddle:2
```

pytorch paddlepaddle-gpu cuda cuda.cudnn /usr/local/ cuda docker paddle

2. paddlex:

```
pip install paddlex==2.1.0 -i https://mirror.baidu.com/pypi/simple
```

https://github.com/PaddlePaddle/PaddleX/blob/develop/docs/quick_start_API.md

2

PaddleSlim

2.0.1

1.
2.
3. 2 1

OCR

2.1

725 58

```
sftp://10.10.2.208:50022/data/data1/zhangyl/meter_det
```

YOLOv3_ResNet34

PPyolov2_ResNet50vd_dcn

YOLOv3_MobileNetV3

ppyolov2_r50vd_dcn

`sftp://10.10.2.208:50022/data/data1/zhangyl/output1/ppyolov2_r50vd_dcn/best_model`

```
num_epochs=270,
train_batch_size=4,
learning_rate=0.000125/2,
warmup_steps=500,
warmup_start_lr=0.0,
lr_decay_epochs=[213, 240],
lr_decay_gamma=0.1,
save_interval_epochs=10,
log_interval_steps=25,
pretrain_weights='COCO'
```

	(MiB)	GPU (MiB)		(s)	(MB)
PPyolov2_ResNet50vd_dcn	3750.9	2774	0.991	2.53	364.6
YOLOv3_ResNet34	4552.4	2534	0.986	2.88	306.3
YOLOv3_MobileNetV3	3988.4	2194	0.978	1.84	185.7

:

2.2

PaddleSlim 2.1.0

`params_analysis.py`, ppyolov2_r50vd_dcn

`sftp://10.10.2.208:50022/data/data1/zhangyl/yibiaopan/params_analysis.py`

API

- step 1:

API :

```
python   model = pdx.load_model('output/yolov3_darknet53/best_model')
model.analyze_sensitivity(       dataset=eval_dataset,       batch_size=1,
save_dir='output/yolov3_darknet53/prune')
```

`output/yolov3_darknet53/prune` `model.sensi.data`

output/yolov3_darknet53/prune/model.sensi.data

- step 2: FLOPs

python `model.prune(pruned_flops=.2, save_dir='./')`

FLOPs 0.2

`sensi.data`

- step 3:

`sensi.data`

“‘python model.train(num_epochs=270, train_dataset=train_dataset, train_batch_size=8, eval_dataset=eval_dataset, learning_rate=0.001 / 8, warmup_steps=1000, warmup_start_lr=0.0, save_interval_epochs=5, lr_decay_epochs=[216, 243], save_dir=’output/yolov3_darknet53/prune’)

“‘

`output/yolov3_darknet53/prune`

pretrain_weights None pretrain_weights

2.2.1

:

	(MiB)	GPU (MiB)		(s)	(MB)
PPyolov2__ResNet50vd__dcn	2408.8	945	0.989	0.38	221.4
YOLOv3__ResNet34	2330.8	938	0.983	0.64	306.3
YOLOv3__MobileNetV3	1767.6	909	0.975	0.101	101.9

1000Mb GPU 1500 b 85 40

NVIDIA Jetson,Raspberry Pi

3

/

3.1

##

```
python quantize.py.py
```

`quantize.py`,

`sftp://10.10.2.208:50022/data/data1/zhangyl/yibiaopan/quantize.py`

`quantize.py` API

step 1:

```
model = pdx.load_model('output/mobilenet_v3/best_model')
```

step 2:

```
model.quant_aware_train(
    num_epochs=100,
    train_dataset=train_dataset,
    train_batch_size=4,
    eval_dataset=eval_dataset,
    learning_rate=0.0001 / 4,
    save_dir='output/mobilenet_v3/quant',
    use_vdl=True)
```

`output/mobilenet_v3/quant`

pretrain_weights None pretrain_weights

3.1.1

:

	(MiB)	GPU (MiB)		(s)	(MB)
PPyolov2_ResNet50vd_dcn	4152.8	950	0.981	0.72	224.2
YOLOv3_MobileNetV3	2586.8	936	0.965	0.34	97.1

GPU

4

Python memory_profiler Pytorch-Memory-Utils

4.0.1 memory_profiler

> pip install memory_profiler#Load its magic function

%load_ext memory_profiler

from memory_profiler import profile

1. %memit,

```
%memit x = 10+5
#Output
peak memory: 54.01 MiB, increment: 0.27 MiB
```

(peak memory) /

2.

```
def addition():
    a = [1] * (10 ** 1)
    b = [2] * (3 * 10 ** 2)
    sum = a+b
    return sum

%memit addition()
#Output
peak memory: 36.36 MiB, increment: 0.01 MiB
```

3. : @profile , “‘python from memory_profiler import profile

@profile def addition(): a = [1] * (10 ** 1) b = [2] * (3 * 10 ** 2) sum = a+b return sum %memit addition() “‘

```
#Output
Line #    Mem usage    Increment   Line Contents
================================================
     2     36.4 MiB     36.4 MiB   @profile
     3                             def addition():
     4     36.4 MiB      0.0 MiB       a = [1] * (10 ** 1)
     5   3851.1 MiB   3814.7 MiB       b = [2] * (3 * 10 ** 2)
     6   7665.9 MiB   3814.8 MiB       sum = a+b
     7   7665.9 MiB      0.0 MiB       return sum
peak memory: 7665.88 MiB, increment: 7629.52 MiB
```

4.0.2 Pytorch-Memory-Utils

Pytorch-Memory-Utils GPU

```
import torch
import inspect

from torchvision import models
from gpu_mem_track import MemTracker  #
```

```
device = torch.device('cuda:0')

frame = inspect.currentframe()
gpu_tracker = MemTracker(frame)      #

gpu_tracker.track()                  #
cnn = models.vgg19(pretrained=True).to(device)  #
gpu_tracker.track()
```