



**University
of Dayton**

**CPS 534: Distributed Computing with
Big Data– Fall 2023**

Title: Pytorch

Date: 22/Nov/2023

To: Professor. Zhongmei Yao

Students:

Student ID: **101744371** || Name: Kevin Richard

1) Optimizer parameters:

I have set learning rate to 0.001 as it gave me over 50% accuracy on the 1st epoch, while a learning rate of 0.1 only gave me an accuracy of around 15% in the first epoch.

```
optimizer = optim.SGD(model.parameters(), lr=0.001,
                      momentum=0.9, weight_decay=0.0001)
```

2) Assignment Part 1 (pyTorch):

i) CPU:

Run with 'cpu' using colab T4 GPUs

```
[4] torch.set_num_threads(4)
    device = "cpu"
    # device = 'cuda' if torch.cuda.is_available() else 'cpu'
```

The compute time is really slow with one epoch taking approximately 1120 secs. 1 epochs gave me a decent accuracy of 56%

```
Files already downloaded and verified
Files already downloaded and verified
epoch : 1 batch_no: 20 MeanLoss_last_20_batches: 2.071 current_batch_time: 5.11 secs
epoch : 1 batch_no: 40 MeanLoss_last_20_batches: 1.714 current_batch_time: 5.03 secs
epoch : 1 batch_no: 60 MeanLoss_last_20_batches: 1.536 current_batch_time: 5.16 secs
epoch : 1 batch_no: 80 MeanLoss_last_20_batches: 1.458 current_batch_time: 4.98 secs
epoch : 1 batch_no: 100 MeanLoss_last_20_batches: 1.432 current_batch_time: 5.77 secs
epoch : 1 batch_no: 120 MeanLoss_last_20_batches: 1.354 current_batch_time: 6.40 secs
epoch : 1 batch_no: 140 MeanLoss_last_20_batches: 1.331 current_batch_time: 6.64 secs
epoch : 1 batch_no: 160 MeanLoss_last_20_batches: 1.279 current_batch_time: 5.27 secs
epoch : 1 batch_no: 180 MeanLoss_last_20_batches: 1.266 current_batch_time: 5.07 secs
Time taken for epoch : 1 = 1117.47 secs
Epoch 1 | Training checkpoint saved at checkpoint.pt
Test set: Average loss: 1.2219, Accuracy: 5584/10000 (56%)
```

ii) GPU:

Run with 'cuda' or GPU using colab T4 GPUs

```
[ ] torch.set_num_threads(4)
    # device = "cpu"
    device = 'cuda' if torch.cuda.is_available() else 'cpu'
```

The compute time is really fast with one epoch taking approximately 24 secs. I have run it for 30 epochs giving me a decent accuracy of 84%.

```
epoch : 30 batch_no: 20 MeanLoss_last_20_batches: 0.293 current_batch_time: 0.04 secs
epoch : 30 batch_no: 40 MeanLoss_last_20_batches: 0.289 current_batch_time: 0.05 secs
epoch : 30 batch_no: 60 MeanLoss_last_20_batches: 0.288 current_batch_time: 0.05 secs
epoch : 30 batch_no: 80 MeanLoss_last_20_batches: 0.309 current_batch_time: 0.05 secs
epoch : 30 batch_no: 100 MeanLoss_last_20_batches: 0.317 current_batch_time: 0.05 secs
epoch : 30 batch_no: 120 MeanLoss_last_20_batches: 0.293 current_batch_time: 0.05 secs
epoch : 30 batch_no: 140 MeanLoss_last_20_batches: 0.313 current_batch_time: 0.05 secs
epoch : 30 batch_no: 160 MeanLoss_last_20_batches: 0.339 current_batch_time: 0.06 secs
epoch : 30 batch_no: 180 MeanLoss_last_20_batches: 0.318 current_batch_time: 0.05 secs
Time taken for epoch : 30 = 23.75 secs
Epoch 30 | Training checkpoint saved at checkpoint.pt
Test set: Average loss: 0.5201, Accuracy: 8360/10000 (84%)
```

3) Assignment Part 2 (DDP):

i) DDP setup:

```
[4] def ddp_setup(rank, world_size):
    """
    Args:
        rank: Unique identifier of each process
        world_size: Total number of processes
    """
    print("in ddp_setup")
    os.environ["MASTER_ADDR"] = "localhost"
    os.environ["MASTER_PORT"] = "12355"
    #init_process_group(backend="nccl", rank=rank, world_size=world_size)
    # rank = torch.device("cuda:{}".format(current_gpu_index))

    torch.distributed.init_process_group(
        backend="nccl",
        init_method="env://",
        world_size=world_size,
        rank=rank,
    )
    torch.cuda.set_device(rank)
```

ii) Handling different GPUs:

I have a new class called 'Trainer()', which initializes with different values depending on the process thread or the GPU it is running. This class also trains my model.

```
class Trainer:
    def __init__(self, model, gpu_id, train_loader, optimizer):
        self.gpu_id = gpu_id
        self.model = model.to(gpu_id)
        self.model = DDP(model, device_ids=[gpu_id])
        self.train_loader = train_loader
        self.optimizer = optimizer
```

iii) Main() Function:

```
def main(rank, world_size):
    print("in")
    ddp_setup(rank, world_size)
```

```
model = VGG11()
optimizer = optim.SGD(model.parameters(), lr=0.1, momentum=0.9, weight_decay=0.0001)
total_epochs = 20
trainer = Trainer(model, gpu_id, train_loader, optimizer)
trainer.train(total_epochs)

destroy_process_group()
```

Using "torch.multiprocessing.start_processes" as "torch.multiprocessing.spawn" is not supported in colab or any jupyter notebook IDEs.

```
if __name__ == "__main__":
    world_size = torch.cuda.device_count()
    # mp.spawn(main, args=(world_size,), nprocs=world_size, join=True, start_method='fork')
    mp.start_processes(main, args=(world_size,), nprocs=world_size, join=True, start_method='fork')
```

iv) Training:

I store the model output of only the first GPU as the DDP allegedly converges my model data across all GPU processes if multiple GPUs present.

```

if self.gpu_id == 0 :
    ckp = self.model.module.state_dict()
    PATH = "checkpoint.pt"
    torch.save(ckp, PATH)
    print(f"Training checkpoint saved at {PATH}")
    running_loss = 0.0

```

Ran it for 20 epochs with T4 Colab GPU. (Rank -> GPU_Id)

```

rank : {0} Time taken for epoch : 19 = 32.00 secs

rank : 0 epoch : 20 batch_no: 20 MeanLoss_last_20_batches: 0.452 current_batch_time: 0.05 secs
Training checkpoint saved at checkpoint.pt
rank : 0 epoch : 20 batch_no: 40 MeanLoss_last_20_batches: 0.437 current_batch_time: 0.05 secs
Training checkpoint saved at checkpoint.pt
rank : 0 epoch : 20 batch_no: 60 MeanLoss_last_20_batches: 0.437 current_batch_time: 0.05 secs
Training checkpoint saved at checkpoint.pt
rank : 0 epoch : 20 batch_no: 80 MeanLoss_last_20_batches: 0.425 current_batch_time: 0.05 secs
Training checkpoint saved at checkpoint.pt
rank : 0 epoch : 20 batch_no: 100 MeanLoss_last_20_batches: 0.425 current_batch_time: 0.06 secs
Training checkpoint saved at checkpoint.pt
rank : 0 epoch : 20 batch_no: 120 MeanLoss_last_20_batches: 0.401 current_batch_time: 0.05 secs
Training checkpoint saved at checkpoint.pt
rank : 0 epoch : 20 batch_no: 140 MeanLoss_last_20_batches: 0.428 current_batch_time: 0.05 secs
Training checkpoint saved at checkpoint.pt
rank : 0 epoch : 20 batch_no: 160 MeanLoss_last_20_batches: 0.395 current_batch_time: 0.05 secs
Training checkpoint saved at checkpoint.pt
rank : 0 epoch : 20 batch_no: 180 MeanLoss_last_20_batches: 0.414 current_batch_time: 0.06 secs
Training checkpoint saved at checkpoint.pt
rank : {0} Time taken for epoch : 20 = 31.46 secs

```

v) Testing:

The model saved during training is loaded and the output is directly predicted by running my last two Colab blocks.

```

device = 'cuda' if torch.cuda.is_available() else 'cpu'
training_criterion = torch.nn.CrossEntropyLoss()

test_model = VGG11()
test_model.to(device)

checkpoint = torch.load('checkpoint.pt')
test_model.load_state_dict(checkpoint)

testing_model(test_model, test_loader, training_criterion, device)

```

vi) Accuracy:

```

Files already downloaded and verified
Test set: Average loss: 0.5737, Accuracy: 8084/10000 (81%)

```

4) Assignment optional part 3 (torchserve)

I only tried a little bit of torchserve and was unable to make an api run. So, I have just attached the python file that might work if all dependencies are installed properly.

5) Issues with running Multiprocessing:

I have included my Multiprocessing .ipynb file. It needs to be converted to python file and run in local machine with cuda and nccl installed.

The Paradox:

Google colab : nccl Supported , multiprocessing.spawn not supported

Windows : nccl not supported, multiprocessing.spawn supported

Alternative:

torch.multiprocessing.start_processes

i) Cuda installation:

conda install pytorch torchvision torchaudio pytorch-cuda=11.8 -c pytorch -c nvidia

ii) Google Colab:

I got the below error:

```
if __name__ == "__main__":
    world_size = torch.cuda.device_count()
    mp.spawn(main, args=(world_size,), nprocs=world_size, join=True)
```

```
152         else:
--> 153             raise ProcessExitedException(
154                 "process %d terminated with exit code %d" % (error_index,
exitcode),
155                 error_index=error_index,
ProcessExitedException: process 0 terminated with exit code 1
```

when I looked it up, I found that the command might not be compatible with notebooks like jupyter or colab.

iii) In Windows VS code:

I tried to install run it as a .py file in my local machine after installing pytorch and cuda. Now, I don't get the above error anymore and my function passes successfully, but i get error that 'nccl' is not installed which is for using the nvidia graphic card and things.

```
-- Process 0 terminated with the following error:
Traceback (most recent call last):
  File "D:\Users\kevr0\AppData\Local\Programs\Python\Python310\lib\site-packages\torch\multiprocessing\spawn.py", line 109, in main
    fn(i, *args)
  File "d:\CPS_534\cps_534_a4_3_main.py", line 109, in main
    ddp_setup(rank, world_size)
  File "d:\CPS_534\cps_534_a4_3_main.py", line 40, in ddp_setup
    init_process_group(backend="nccl", rank=rank, world_size=world_size)
  File "D:\Users\kevr0\AppData\Local\Programs\Python\Python310\lib\site-packages\torch\distributed\c10d_logger.py", line 109, in func_return = func(*args, **kwargs)
  File "D:\Users\kevr0\AppData\Local\Programs\Python\Python310\lib\site-packages\torch\distributed\distributed_c10d.py", line 109, in _new_process_group_helper
    raise RuntimeError("Distributed package doesn't have NCCL built in")
RuntimeError: Distributed package doesn't have NCCL built in
```

Devices on GPU. Unfortunately NVidia NCCL is not supported on Windows, but it is supported for other platforms. Nov 25, 2022

6) Attached files:

- i) CPS_534_A4_part1.ipynb (Part 1)
- ii) CPS_534_A4_multiprocessing.ipynb (Part 2)
- iii) vgg_handler.py (Part 3)
- iv) checkpoint.pt (pretrained model)

7) References:

<https://medium.com/deelvin-machine-learning/model-parallelism-vs-data-parallelism-in-unet-speedup-1341bc74ff9e>

https://pytorch.org/tutorials/beginner/blitz/cifar10_tutorial.html

<https://pytorch.org/tutorials/beginner/introyt/trainingyt.html>

<https://www.youtube.com/watch?v=Cvdhwx-OBBo>

<https://www.youtube.com/watch?v=jdE4hPf9juk>