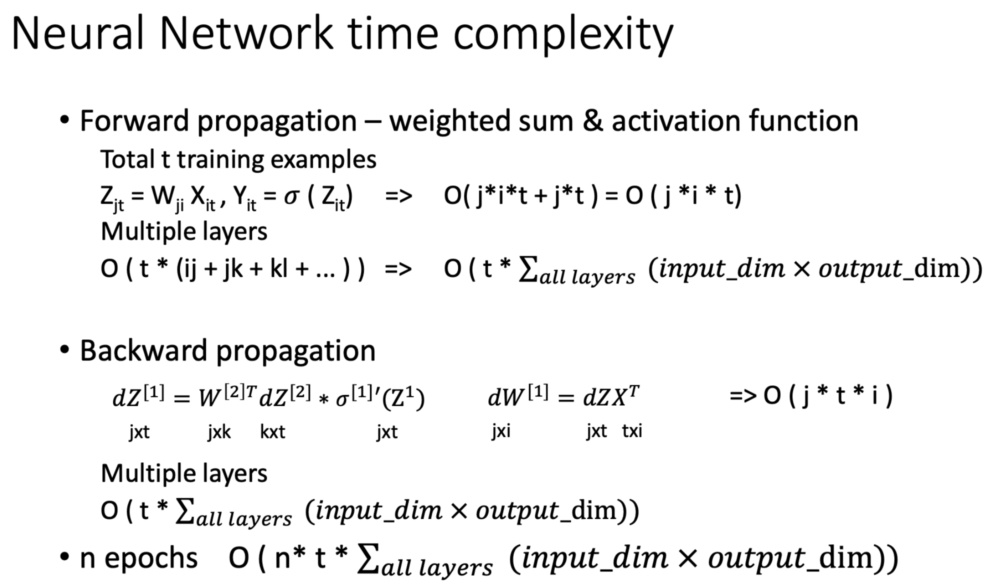
CSCIGA.3033 HW 3
Chenmeinian Guo

1.



Neural Network time complexity

- Forward propagation – weighted sum & activation function

  Total t training examples

  $Z_{jt} = W_{ji} X_{it}$, $Y_{it} = \sigma (Z_{it})$ => O(j*i*t + j*t) = O (j *i * t)

  Multiple layers

  O (t * (ij + jk + kl + …)) => O (t * $\sum_{all\ layers}$ $(input\_dim \times output\_\mathrm{dim})$)

- Backward propagation

  $\underbrace{dZ^{[1]}}_{jxt} = \underbrace{W^{[2]T}}_{jxk}\underbrace{dZ^{[2]}}_{kxt} * \underbrace{\sigma^{[1]\prime}(Z^1)}_{jxt}$ $\underbrace{dW^{[1]}}_{jxi} = \underbrace{dZ}_{jxt}\underbrace{X^T}_{txi}$ => O (j * t * i)

  Multiple layers

  O (t * $\sum_{all\ layers}$ $(input\_dim \times output\_\mathrm{dim})$)

- n epochs O (n* t * $\sum_{all\ layers}$ $(input\_dim \times output\_\mathrm{dim})$)

Time Complexity

- self.conv1=nn.Conv2d(1,32,3,1) #1 input channels, 32 output channels (or kernels), and a kernel size of 3x3

  FLOPs#1 (for conv1): 2×26×26×3×3×1×32

- self.conv2 = nn.Conv2d(32, 64, 3, 1) #32 input channels, 64 output channels (or kernels), and a kernel size of 3x3

  FLOPs#2 (for conv1): 2×24×24×3×3×32×64

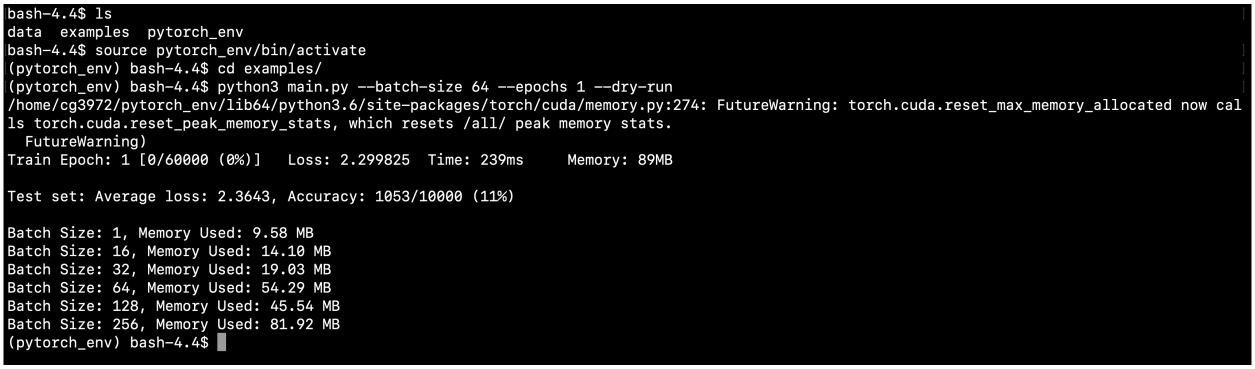
FLOPs for Conv2d-2 in total =
O(n * t * (FLOPs#1 + FLOPs#2)) where n is number of epochs and t is for time spent for each flop

Space Complexity

- self.conv1 = 1×3×3×32 for weights and 32 for biases
- self.conv2 = 32×3×3×64 for weights and 64 for biases

# this is a vague approximation

2.

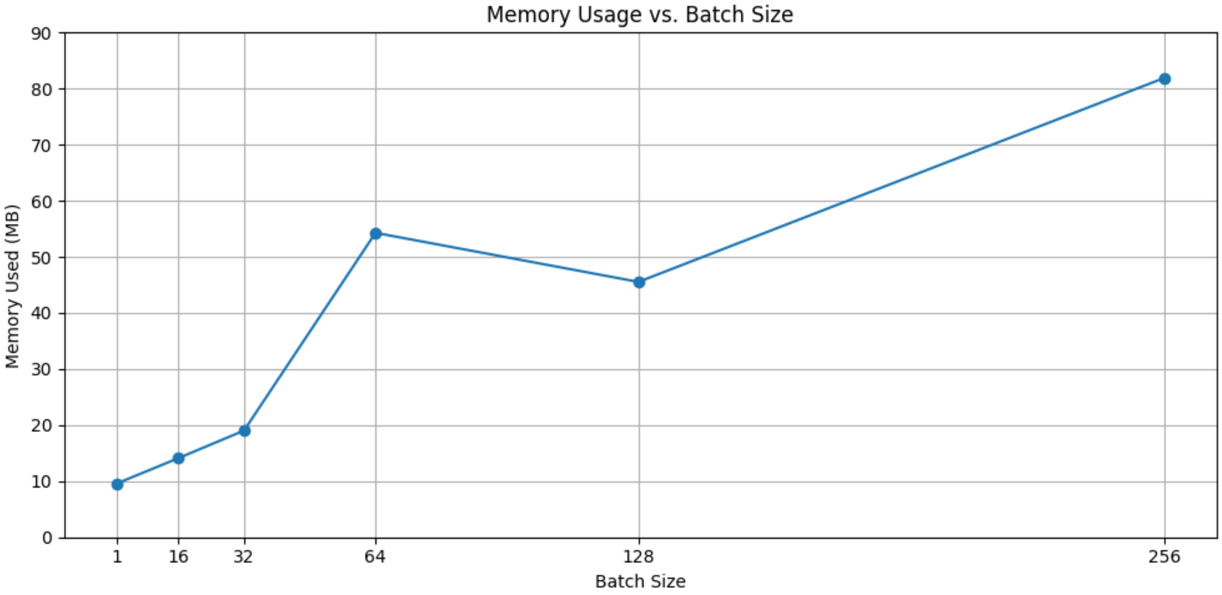


```
bash-4.4$ ls
data  examples  pytorch_env
bash-4.4$ source pytorch_env/bin/activate
(pytorch_env) bash-4.4$ cd examples/
(pytorch_env) bash-4.4$ python3 main.py --batch-size 64 --epochs 1 --dry-run
/home/cg3972/pytorch_env/lib64/python3.6/site-packages/torch/cuda/memory.py:274: FutureWarning: torch.cuda.reset_max_memory_allocated now cal
ls torch.cuda.reset_peak_memory_stats, which resets /all/ peak memory stats.
  FutureWarning)
Train Epoch: 1 [0/60000 (0%)]   Loss: 2.299825  Time: 239ms     Memory: 89MB

Test set: Average loss: 2.3643, Accuracy: 1053/10000 (11%)

Batch Size: 1, Memory Used: 9.58 MB
Batch Size: 16, Memory Used: 14.10 MB
Batch Size: 32, Memory Used: 19.03 MB
Batch Size: 64, Memory Used: 54.29 MB
Batch Size: 128, Memory Used: 45.54 MB
Batch Size: 256, Memory Used: 81.92 MB
(pytorch_env) bash-4.4$ 
```

3.



4. The measured result is more comprehensive and more reliable. The estimated result ignores the specific software implementations, potential GPU accelerations from techniques such as memory caching, hardware efficiency, etc. Based on the output shown, the memory usage generally increases with the batch size, which is expected.