Strong Scaling Strategy for Deep Neural Network Seismic Segmentation Models

Eduardo Rodrigues

Rice Oil & Gas HPC - Houston - March 4th, 2019
Deep Neural Networks have become the main tool for visual recognition
Deep Neural Networks have become the main tool for visual recognition.

They also have been used by seismologists to help interpret seismic data.
Seismic Segmentation Models based on DNNs

A symbiotic partnership

- Deep Neural Networks have become the main tool for visual recognition
- They also have been used by seismologists to help interpret seismic data
- Relevant training examples may be sparse
Seismic Segmentation Models based on DNNs

A symbiotic partnership

- Deep Neural Networks have become the main tool for visual recognition
- They also have been used by seismologists to help interpret seismic data
- Relevant training examples may be sparse
- Training these models may take very long
Deep Neural Networks have become the main tool for visual recognition.

They also have been used by seismologists to help interpret seismic data.

Relevant training examples may be sparse.

Training these models may take very long.

Parallel execution speed up training.
Seismic Segmentation Models based on DNNs

Challenges

- Current deep leaning models (Alexnet, VGG, Inception) do not fit well the task
  - They are too big
  - Little data (compared to traditional vision recognition tasks)
  - Data pre-processing forces model’s input to be smaller
Current deep learning models (Alexnet, VGG, Inception) do not fit well the task

- They are too big
- Little data (compared to traditional vision recognition tasks)
- Data pre-processing forces model’s input to be smaller

Parallel execution strategies proposed in the literature are not appropriate
What is the recommendation:

- Accurate, Large Minibatch SGD: Training ImageNet in 1 Hour
  - Priya Goyal
  - Lukasz Wesolowski
  - Piotr Dollár
  - Aapo Kyrola
  - Ross Girshick
  - Andrew Tulloch
  - Pieter Noordhuis
  - Yangqing Jia
  - Kaiming He
  - Facebook

- Scaling SGD Batch Size to 32K for ImageNet Training
  - Yang You
  - Igor Gitman
  - Boris Ginsburg
  - UC Berkeley
  - CMU
  - NVIDIA

- DON'T DECAY THE LEARNING RATE,
  INCREASE THE BATCH SIZE

  - Samuel L. Smith
  - Pieter-Jan Kindermans
  - Chris Ying
  - Quoc V. Le
  - Google Brain

{sllsmith, pikinder, chrisying, qvl}@google.com
Basics: deep learning sequential execution

Training basics
- loop over mini-batches and epochs
  - forward propagation
  - compute loss
  - backward propagation (gradients)
  - update parameters

\[
L = \frac{1}{N_{bs}} \sum_{i} L_i, \quad \frac{\partial L_i}{\partial W_n}
\]
Parallel execution
single node - multi-GPU system

Many ways to divide the deep neural network
The most common strategy is to divide mini-batches across GPUs

- The model is replicated across GPUs
- Data is divided among them
- Two possible approaches:
  - non-overlapping division
  - shuffled division
- Each GPU computes forward, cost and mini-batch gradients
- Gradients are then averaged and stored in a shared space (visible to all GPUs)
Parallelization strategies

multi-node

One can use a similar strategy with multi-node

It requires communication across nodes

Two strategies:
- Asynchronous
- Synchronous
- Can be implemented with high efficiency protocols
- No need to exchange variables
- Faster in terms of time to quality
Traditional technique

minibatch ➔ Accelerator

Loss landscape
Traditional technique

- Minibatch 0 ➔ Accelerator 0
- Minibatch 1 ➔ Accelerator 1
- Minibatch 2 ➔ Accelerator 2

Loss landscape
Traditional technique pitfalls

Key assumptions are:

- the full batch is very large
- the effective minibatch is still a small fraction of the full batch

A hidden assumption is that small full batches don’t need to run in parallel
Not only Imagenet can benefit from parallel execution

LUng Nodule Analysis
2016
weak scaling, strong scaling
weak scaling, strong scaling
The dataset used in our experiments is one expert interpretation of a public 3D seismic survey called Netherlands Offshore F3 Block which is freely available.
Time to run 200 epochs

- Strong Scaling
- Weak Scaling

<table>
<thead>
<tr>
<th># of GPUs</th>
<th>Execution Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
</tr>
<tr>
<td>500</td>
<td></td>
</tr>
<tr>
<td>1000</td>
<td></td>
</tr>
<tr>
<td>1500</td>
<td></td>
</tr>
<tr>
<td>2000</td>
<td></td>
</tr>
<tr>
<td>2500</td>
<td></td>
</tr>
</tbody>
</table>

Eduardo Rodrigues – Strong Scaling Strategy for Deep Neural Network Seismic Segmentation Models

© 2019 IBM Corporation
our experiments (1)

Time to run 200 epochs

Intersection over union

- Strong 2 GPUs
- Strong 4 GPUs
- Strong 8 GPUs
- Weak 2 GPUs
- Weak 4 GPUs
- Weak 8 GPUs

Execution time (s)

- Strong
- Weak

IOU

Epochs
Time to reach 60% IOU

execution time (s)

# of GPUs

Strong

Weak
Time to reach 60% IOU

Intersection over union

- Strong
- Weak

- # of GPUs
- execution time (s)

- IOU
- Epochs

- strong 2 GPUs
- strong 4 GPUs
- strong 8 GPUs
- weak 2 GPUs
- weak 4 GPUs
- weak 8 GPUs
Deep Neural Networks have a large potential to help seismologists with their work.

In early-stage projects these professionals may not have enough time to supply a large number of examples to train DNNs.

In this scenario, our proposed strategy to run those models in parallel can run faster and, consequently, improve productivity.

This strategy goes against the recommendation found in the literature, mainly because these recommendations have focused on benchmarks that may not be of particular interest for traditional industries such as those in the energy sector.
@ IEEE/ACM CCGrid - Cyprus

http://hpml2019.github.io