Systems problems in deep learning

The importance of machine learning systems to power deep learning algorithms
Deep learning enables emerging applications
Deep learning is powered by advances in systems
Deep Learning is fundamentally different from classical software engineering.

<table>
<thead>
<tr>
<th></th>
<th>SOFTWARE 1.0</th>
<th>“SOFTWARE 2.0”</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATA</td>
<td>NONE</td>
<td>TERABYTES</td>
</tr>
<tr>
<td>COMPUTE</td>
<td>SEVERAL CORES</td>
<td>PETAFLOPS</td>
</tr>
<tr>
<td>HARDWARE</td>
<td>CPU</td>
<td>GPU, TPU, ...</td>
</tr>
<tr>
<td>COMPILE TIME</td>
<td>SECONDS</td>
<td>WEEKS</td>
</tr>
<tr>
<td>ACCURACY</td>
<td>LOGICAL (SPEC)</td>
<td>PROBABILISTIC</td>
</tr>
<tr>
<td>DEBUGGING</td>
<td>PRINTF</td>
<td>TENSORBOARD</td>
</tr>
</tbody>
</table>
Deep learning systems support is emerging

MLSys

Workshop on ML for Systems at NeurIPS
As a DL engineer, you are not solely responsible for DL

PyTorch  Keras
As a DL engineer, you are not solely responsible for DL

- PyTorch
- Keras
- Infrastructure management
- Experiment tracking
- Hyperparameter optimization
- Distributed training
As a DL engineer, you are not solely responsible for DL
As a DL engineer, you are not solely responsible for DL.
The Deep Learning Tool We Wish We Had In Grad School

By Angela Jiang, Liam Li
November 05, 2020

https://determined.ai/blog/deep-learning-tools-grad-students/
Deep learning in grad school

- Distributed Training
- Experiment tracking for reproducibility
- Hyperparameter Optimization
- GPU Infrastructure management
Deep learning in grad school

GPU Infrastructure management
Deep learning in grad school

Experiment tracking for reproducibility
Deep learning in grad school

Do you still have questions about distributed TF?

Angela Jiang <tj.aijiang@gmail.com>

to Aaron

I can try to get them answered while I'm still here over the next two weeks. Let me know!

Thanks,
Angela

Aaron Harlap <aharlap@andrew.cmu.edu>

to me

Hey Angela

I do still have a question. I have been trying to figure out if it is possible to maintain the same for the quality of work size of 64, and get the same accuracy as training 5 epochs on two machines with mini batch size of 32 each. I have

Not a huge deal if you can't find someone who has a good answer for this, but would be nice to be able to do this.

Thanks

Aaron
Deep learning in grad school

Grid Search

Hyperparameter Optimization
Our talk today

- Distributed Training
- Experiment tracking for reproducibility
- Hyperparameter Optimization
- GPU infrastructure management
Our talk today

Distributed Training
Infrastructure challenges of distributed training
Infrastructure challenges of distributed training
Scaling Distributed Machine Learning with the Parameter Server

Ming Li¹, David G. Andersen², Jia Wu², Peter Bailis², Alexander J. Smola³, Anshul Gupta⁴, Vaishaal Shandilya⁵, James Lee⁶, Raghuram Sivakumar⁶, Ben Ying Se⁷

¹Cambridge University, ²Google, ³Yahoo! Research, ⁴Intel, ⁵Virginia Tech, ⁶University of California, Berkeley, ⁷Amazon

Abstract

We propose a parameter server framework for distributed machine learning problems. Each node is assigned an independent workload, while a single node maintains a global parameter server to manage data exchanges. The framework supports automatic scaling and communication models. It eliminates the need for a single leader for data management and contains fault tolerance.

To demonstrate the scalability of the proposed framework, we evaluate the performance of our implementation on CPU and GPU clusters and report on our experience tuning the system for optimal performance.

Table 1: Analysis of machine learning jobs for a three month period in data centers

- [Data processing rate] 12.3 GB/s
- [Job duration] 1230 seconds
- [Throughput] 12.3 TB/hour

2014
Parameter server

Parameter Server

Parameter Server. Li et al. 2014
Parameter server
Parameter server
Parameter server
Reduce network traffic with range-based communication

**Communication consists of small updates**

Batch communication into vectors

$$\text{vectorclock}(R) = (k_1, v_1), ..., (k_p, v_p)$$

- Range-based vector clocks
- Range-based push and pull
Relaxed consistency model

*Trade-off system efficiency with algorithm convergence rate*

Tasks can be executed asynchronously and in parallel.
Infrastructure challenges of distributed training

Scaling Distributed Machine Learning with the Parameter Server
*Cambridge, MIT; **IBM

Abstract
We propose a parameter server framework for distributed machine learning problems. Each node is assigned an unshared local machine, while the server node maintains globally shared parameters, and speakers serve as agents to receive and maintain the parameter server. We demonstrate that our approach can achieve excellent scalability and performance, even with up to 1,000 workers. The framework is inherently robust to failures, tolerates network partitions, and continues to function.

Table 1: Statistics of machine learning jobs for a three month period in two regions

2014  2016
How many servers are enough?
How many servers are enough?
How many servers are enough?
How many servers are enough?
Infrastructure challenges of distributed training

Scaling Distributed Machine Learning with the Parameter Server

Abstract

We propose a parameter server framework for distributed machine learning problems. The framework enables an arbitrary number of worker nodes, which can perform local training on distributed data, to share model parameters in real-time. This framework supports a wide range of distributed machine learning algorithms, including deep learning, and contains both regression and classification.

2014

2016

2018
Decentralized communication with allreduce
Infrastructure challenges of distributed training

Scaling Distributed Machine Learning with the Parameter Server
Mu Li¹, David G. Andersen, Jae Woo Park, Alexander J. Smola², Anirbam Bhaduri, Yaqin Zou², Jingdong Sun, Srikantan K. Bhaskara, Jinyang Huan
Carnegie Mellon University · Microsoft · Google

Abstract
We propose a parameter server framework for distributed machine learning systems. Each data is divided into a distributed set of index nodes, while the server node maintains globally shared parameters. The framework removes the master-slave communication bottleneck, enabling an order-of-magnitude increase in the number of workers without increasing the inter-node bandwidth. It also supports fault tolerance. To demonstrate the scalability of the proposed framework, we implement and evaluate the system in a cloud environment.

<table>
<thead>
<tr>
<th>Year</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2014</td>
<td>Google Launches TensorFlow</td>
</tr>
<tr>
<td>2016</td>
<td>TensorFlow 2.0 Released</td>
</tr>
<tr>
<td>2018</td>
<td>Horovod, a distributed deep learning library, released</td>
</tr>
</tbody>
</table>

Table 1: Analysis of machine learning jobs for a three month period in data centers

- TensorFlow now supports distributed training for large-scale machine learning tasks.
- Horovod can scale to thousands of workers and provides fault tolerance.
- TensorFlow 2.0 introduces new features like eager execution and automatic differentiation.

TF

HOROVOD
TensorFlow distributed training introduces program complexity

Start parameter server

Start each worker

Pass around discovery info (host IP address, network port)

Construct a training object with all the cluster information

    tf.Server()
    tf.ClusterSpec()
    tf.train.device_replica_setter()
Horovod uses allreduce and improves user experience

```
```
Infrastructure challenges of distributed training

Scaling Distributed Machine Learning with the Parameter Server

2014

2016

2018

Today
Outline

Hyperparameter Optimization
Hyperparameter tuning is critical for model performance

$k=2$  

$k=3$  

![Graphs showing loss over epochs for different learning rates with $k=2$ and $k=3$.](image-url)
Case study: Neural Architecture Search

NasNet. Zoph et al. 2018
Case study: Neural Architecture Search

800 GPUs for 28 days, or 22,400 GPU-hours
HP Search requires managing a large set of experiments

https://medium.com/@cjl2fv/an-intro-to-hyper-parameter-optimization-using-grid-search-and-random-search-d73b9834ca0a
Need to manage a large set of experiments efficiently
Run multiple experiments in parallel

[Image: Diagram of Grid Search with important parameter and multiple GPUs running experiments in parallel]

https://medium.com/@cjl2fv/an-intro-to-hyper-parameter-optimization-using-grid-search-and-random-search-d73b8634ca0a
Run multiple experiments in parallel

Grid Search

Learning Rate: 0.1
Decay: 0.005
Accuracy: 985%

Learning Rate: 0.01
Decay: 0.005
Accuracy: 99%

Learning Rate: 0.005
Decay: 0.005
Accuracy: 92%

https://medium.com/@cj2fv/an-intro-to-hyper-parameter-optimization-using-grid-search-and-random-search-d73b8634ca0a
Try configurations similar to those that have done well

Scheduler

Important parameter

98%

https://medium.com/@cjzj/intro-hyper-parameter-optimization-using-grid-search-and-random-search-d73b834ca0a
Stop experiments early if they are not performing well

https://medium.com/@cjl2fv/an-intro-to-hyper-parameter-optimization-using-grid-search-and-random-search-d73b8b634ca0a
Stop experiments early if they are not performing well

Important parameter

Scheduler

https://medium.com/@cj/fv/an-intro-to-hyper-parameter-optimization-using-grid-search-and-random-search-d73b6634ca0a
Stop experiments early if they are not performing well

https://medium.com/@cjl2fv/an-intro-to-hyper-parameter-optimization-using-grid-search-and-random-search-d73b683ca0a
ASHA algorithm is 3x faster than Vizier training an LSTM on PTB

https://blog.ml.cmu.edu/2018/12/12/massively-parallel-hyperparameter-optimization/

ASHA. Li et al. 2020
Running HP search on the cloud is commonplace
Running HP search on the cloud is expensive
Running HP search on the cloud is expensive

p3.2xlarge = $3 per hour

100 8hr experiments = $2k
On demand instances can cost 10x more than preemptible instances

Preemptible Instances

Google Cloud Platform = 70%-90% Savings

aws
spot instances

https://medium.com/@cjlihv/an-intro-to-hyper-parameter-optimization-using-grid-search-and-random-search-d73b9634ca0a
On demand instances can cost 10x more than preemptible instances
On demand instances can cost 10x more than preemptible instances

https://medium.com/@cj2fh/an-intro-to-hyper-parameter-optimization-using-grid-search-and-random-search-d73b8634ca0a
The result is orders of magnitude more efficient HP search

\[
\begin{align*}
4x \times 3x \times 10x &= 100x \\
\text{Distributed training} \times \text{ASHA} \times \text{Preemptible instances} &= \text{savings}
\end{align*}
\]
Containerized training for reproducibility
Structured artifacts and logs for experiment tracking

```
@ubuntu:/$ ls -lh
total 92K
drwxr-xr-x 2 root root 4.0K Jul  9 15:53 bin
drwxr-xr-x 3 root root 4.0K Jul  9 16:01 boot
drwxr-xr-x 2 root root 4.0K Jul  9 15:43 cdrom
drwxr-xr-x 15 root root 4.3K Jul  9 16:01 dev
drwxr-xr-x 128 root root 12K Jul  9 16:04 etc
drwxr-xr-x 3 root root 4.0K Jul  9 15:45 home
```
Outline

GPU infrastructure management
<table>
<thead>
<tr>
<th>Resource sharing in traditional vs. deep learning clusters</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Traditional schedulers</strong></td>
</tr>
<tr>
<td>One-time placement</td>
</tr>
<tr>
<td>Stuck w/ decision for entire job</td>
</tr>
<tr>
<td>Systems level profiling</td>
</tr>
<tr>
<td>Profiles CPU/GPU usage</td>
</tr>
<tr>
<td>Traditional schedulers</td>
</tr>
<tr>
<td>------------------------</td>
</tr>
<tr>
<td>One-time placement</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Systems level profiling</td>
</tr>
<tr>
<td>Profiles CPU/GPU usage</td>
</tr>
</tbody>
</table>
## Resource sharing in traditional vs. deep learning clusters

<table>
<thead>
<tr>
<th>Traditional schedulers</th>
<th>Deep learning schedulers</th>
</tr>
</thead>
<tbody>
<tr>
<td>One-time placement</td>
<td>Continuous/introspective</td>
</tr>
<tr>
<td>Stuck w/ decision for entire job</td>
<td>Recovers from mistakes</td>
</tr>
<tr>
<td>Systems level profiling</td>
<td></td>
</tr>
<tr>
<td>Profiles CPU/GPU usage</td>
<td></td>
</tr>
<tr>
<td>App level profiling</td>
<td></td>
</tr>
<tr>
<td>Measures batch-level work</td>
<td></td>
</tr>
<tr>
<td>Distributed train</td>
<td></td>
</tr>
<tr>
<td>HP search</td>
<td></td>
</tr>
</tbody>
</table>
Determined AI
As a DL engineer, only be responsible for DL

- PyTorch
- Keras
As a DL engineer, only be responsible for DL

- Infrastructure management
- Hyperparameter optimization
- Experiment tracking
- Distributed training
Try out Determined -- it's open source!
We’d love to hear from you!

@jiangela
www.angelahjiang.com

@determinedai
www.determined.ai
determined-community.slack.com
Thank you!

Learn more at Determined AI