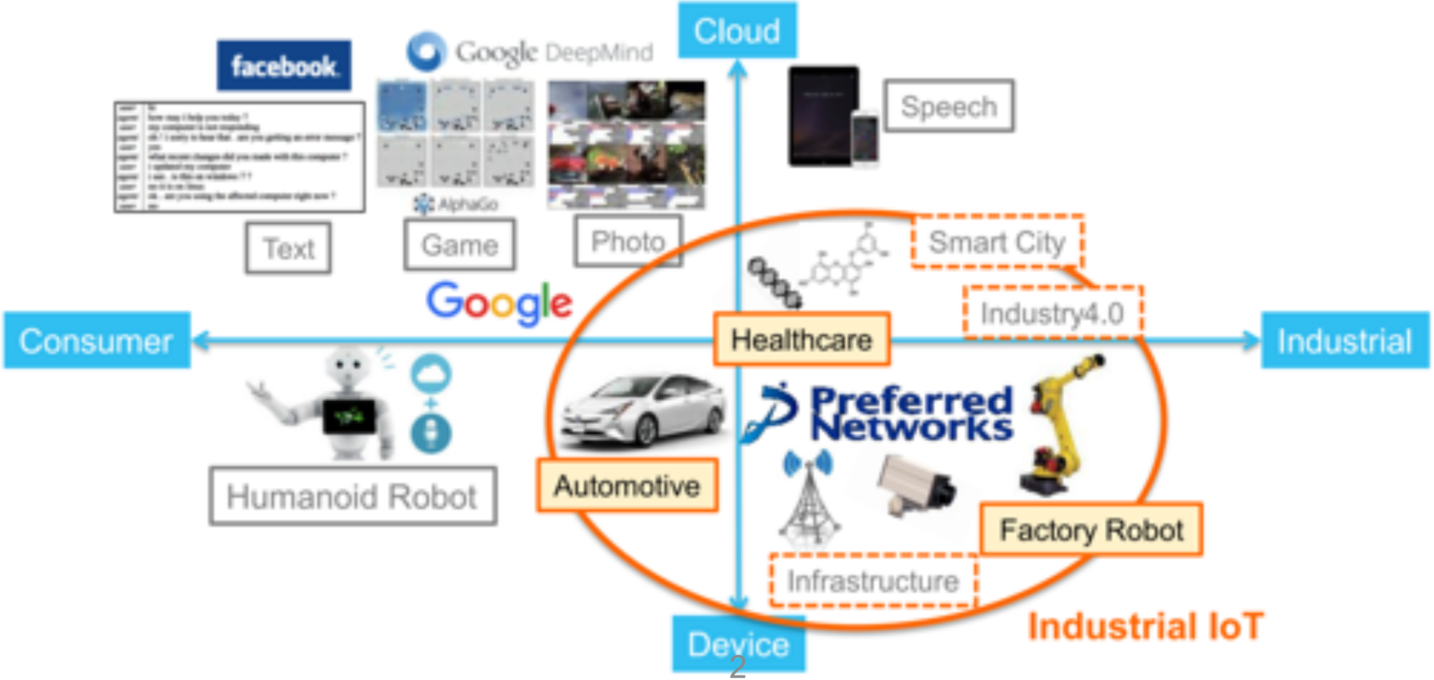
Large scale deep learning in PFN: from 15-min Imagenet to PFDet

Hirochika Asai, **Keisuke Fukuda**
Preferred Networks, Inc.

Who we are?

Preferred Networks, Inc. (PFN):

A Tokyo-based Deep Learning & IoT company



Preferred Networks

Our Strategic Partners

TOYOTA

FA & ROBOT & ROBOMACHINE FANUC

NTT

HITACHI Inspire the Next

Hakuhodo DY holdings

MIZUHO

MITSUI & CO.

CHUGAI

Roche A member of the Roche group

TEL TOKYO ELECTRON

and Collaborators

NVIDIA

CISCO

intel

IBM

Microsoft

2015: OPTIMIZATION OF BIN-PICKING FANUC ROBOTS

- Picking random object is a typical task that is “easy for human, hard for robots”.

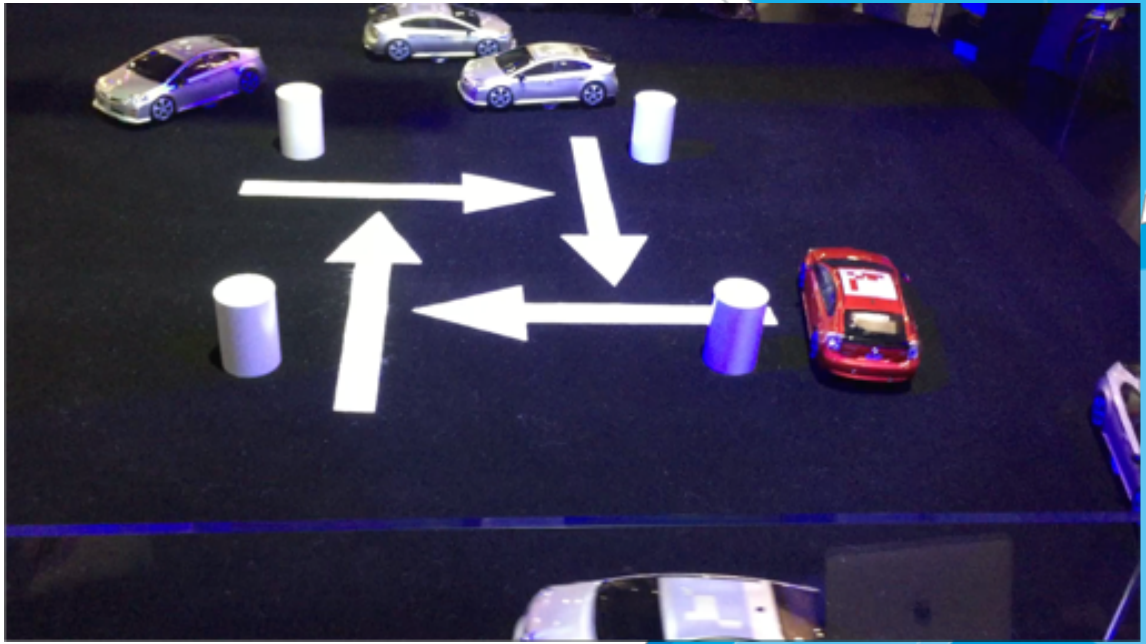
Preferred Networks

@CES 2016: CARS THAT DON'T CRASH

- Car positions are tracked from a ceiling camera and each car is controlled individually.

- White cars are autonomous

- The red car is a manually-controlled "evil" car: trying to disrupt other cars



@ICRA 2017 VOICE RECOGNITION + OBJECT PICKING

"Interactively Picking Real-World Objects with Unconstrained Spoken Language Instructions"
arXiv:1710.06280

- ICRA is a top-tier conference on robotics
- Best Paper Award on Human-Robot Interaction

- Technologies:
  - Visual recognition
  - Natural language processing (NLP)

- The robot can understand ambiguous words:
  - "The Teddy bear"
  - "The brown fluffy stuff"

https://projects.preferred.jp/tidying-up-robot/

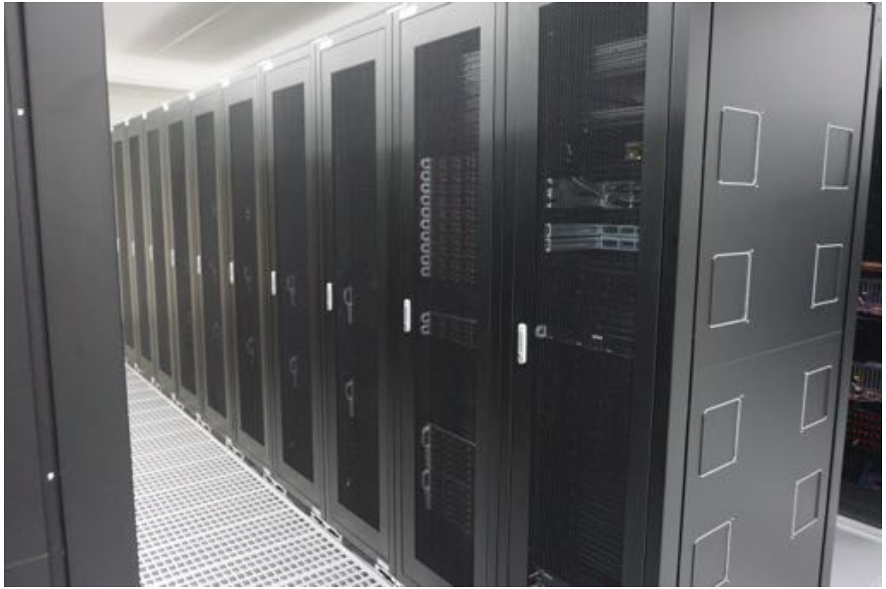
@CEATEC JAPAN 2018 Autonomous Tidying-up Robot System

Technologies behind the demos:

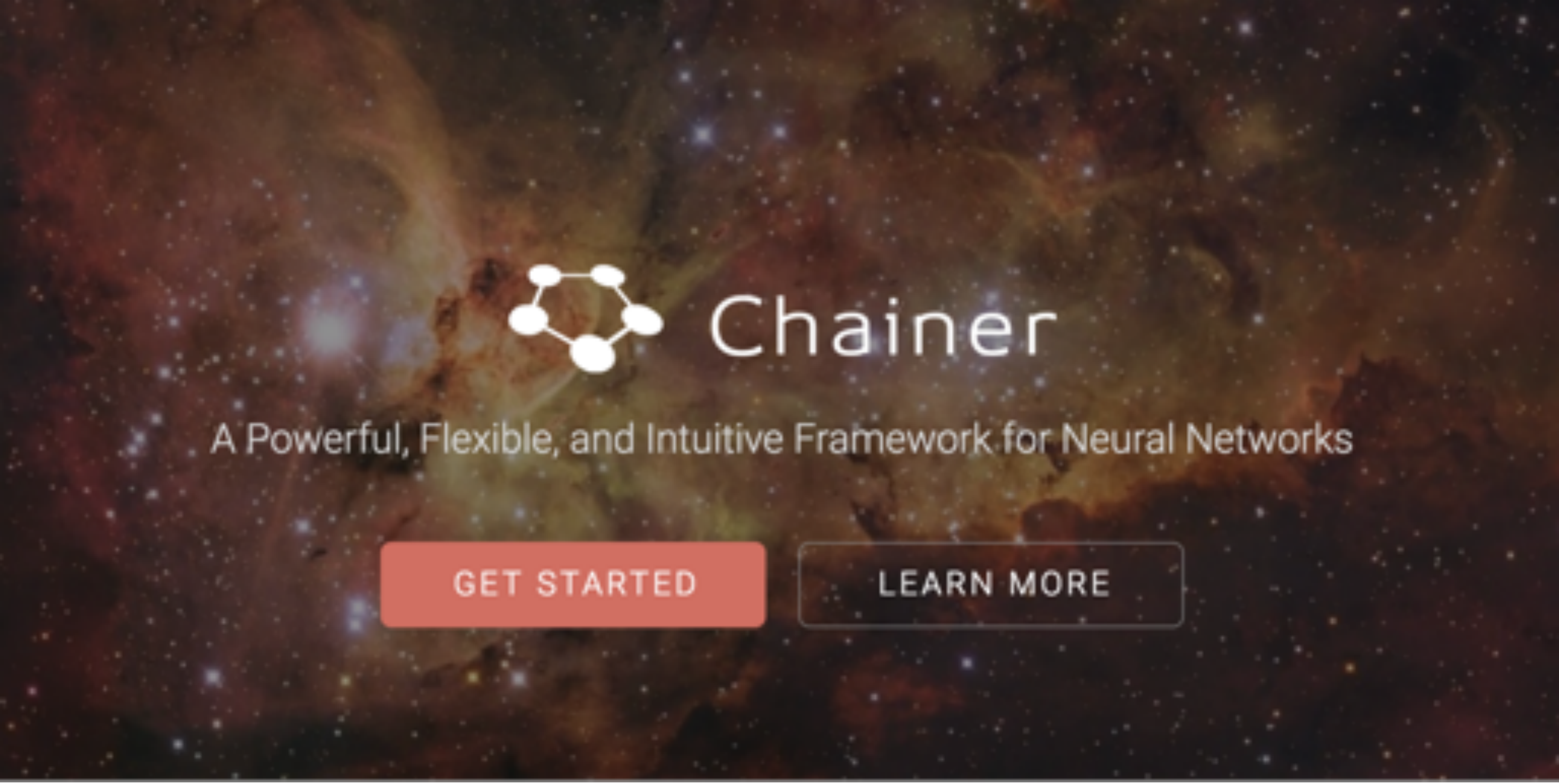
Distributed Deep Learning

MN-1: An in-house supercomputer

- **MN-1a** (Sep. '17~)
  - 1024 NVIDIA Tesla P100
  - InfiniBand FDR
  - Peak 9.3 Peta FLOPS (SP)
  - #227 in Top500 Nov. 2018
- **MN-1b** (July. '18~)
  - 512 NVIDIA Tesla V100 32GB
  - InfiniBand EDR
  - Peak 56 Peta (tensor) Flops



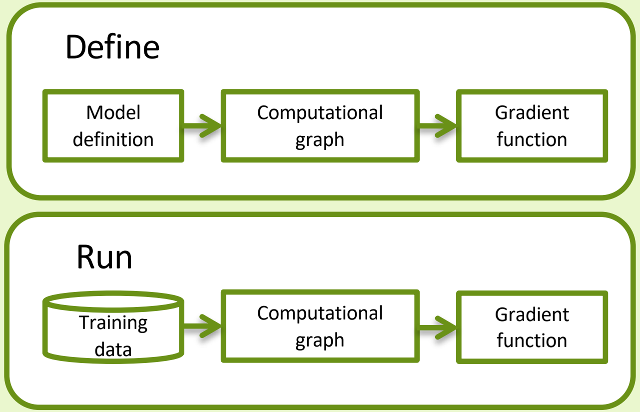
- Targeting Exa FL ops by 2020



https://chainer.org/

Chainer:
A Flexible Deep Learning Framework

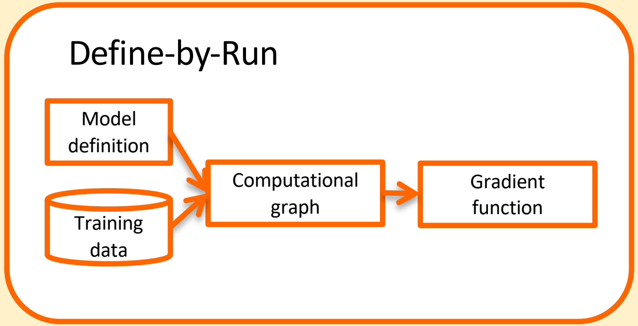
Define-and-Run



Caffe2, TensorFlow etc.

Define-by-Run

Chainer



PyTorch, TensorFlow(Eager Execution) etc.

ChainerMN: Distributed Training with Chainer

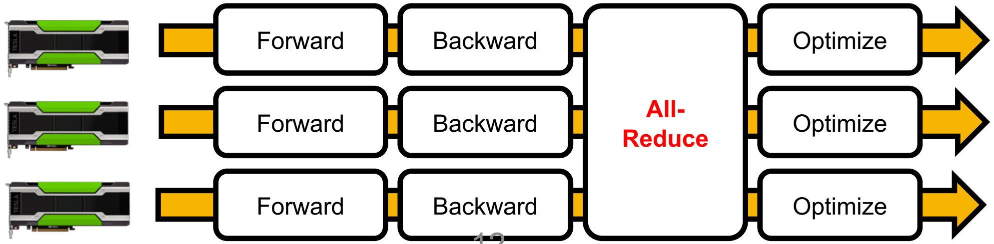
- Add-on package for Chainer
- Enables multi-node distributed deep learning using NVIDIA NCCL2

Features

- **Scalable**: Near-linear scaling with hundreds of GPUs
- **Flexible**: Even GANs, dynamic NNs, and RL are applicable

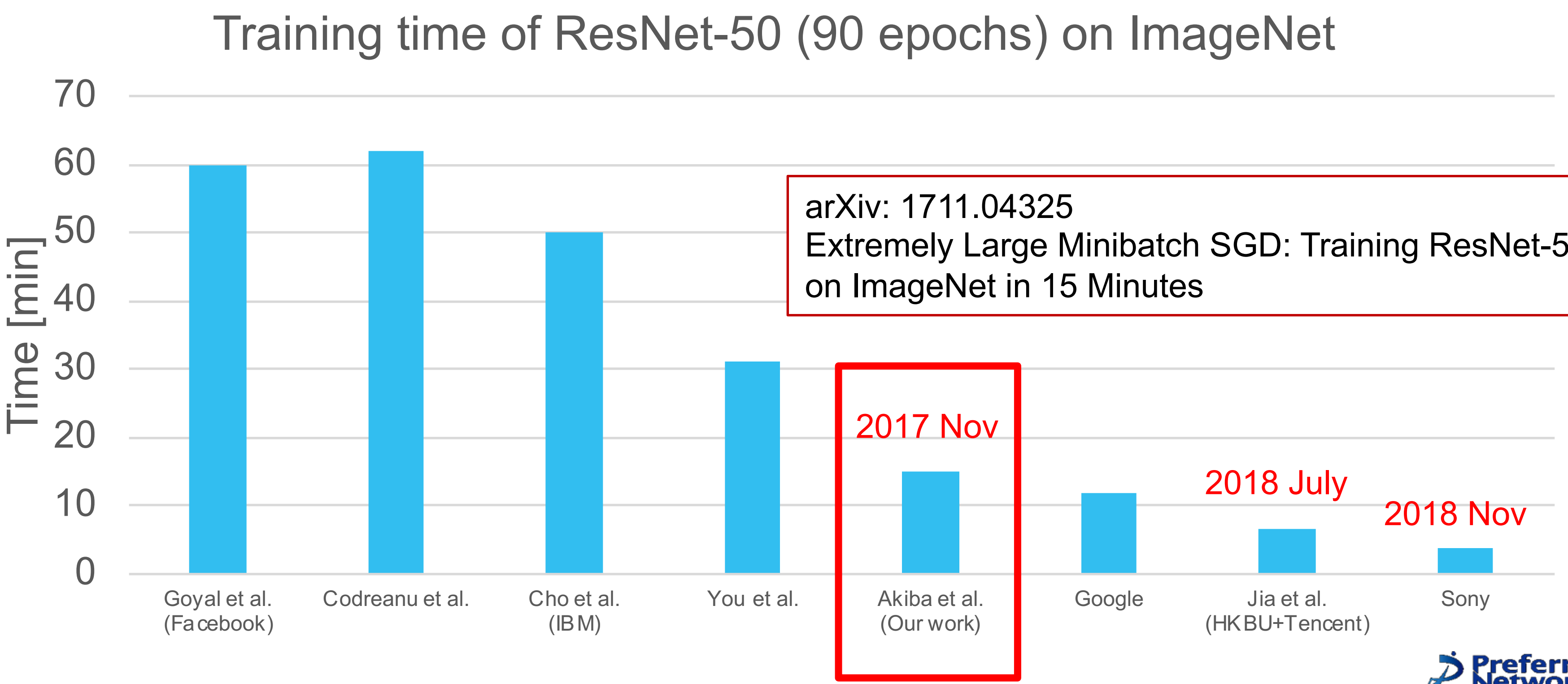
ChainerMN

Distributed Training with ChainerMN

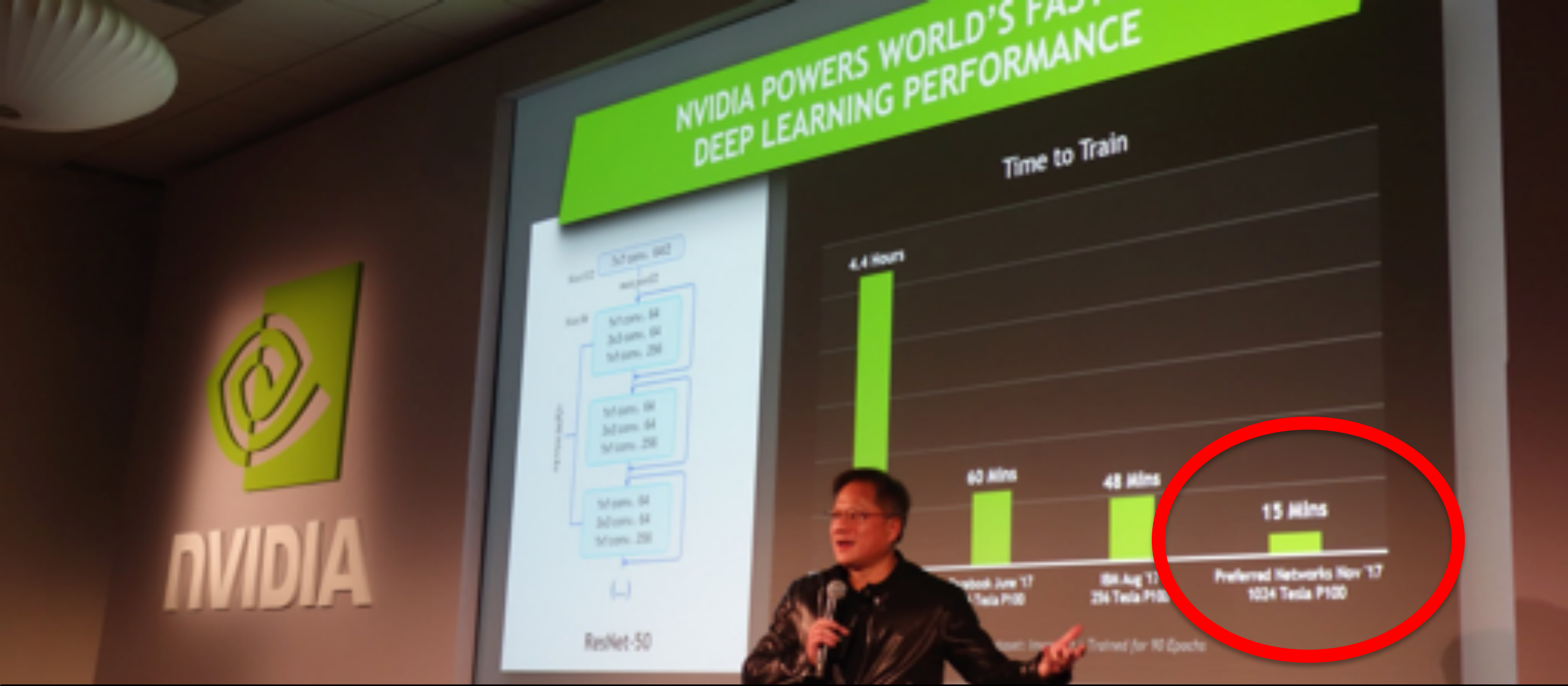


Preferred Networks

Achievement on **MN-1a**: ImageNet in 15 minutes



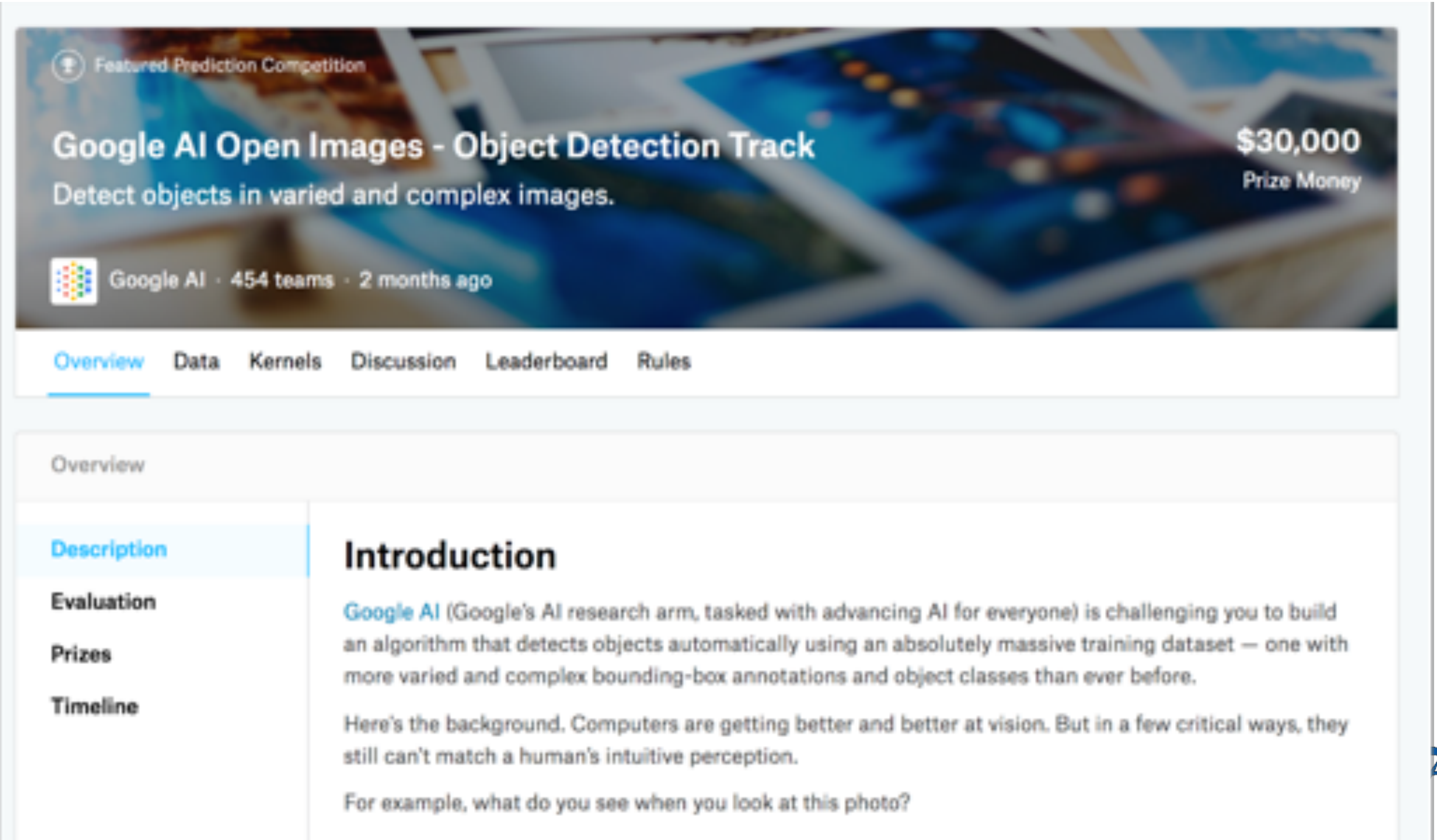
arXiv: 1711.04325
Extremely Large Minibatch SGD: Training ResNet-50 on ImageNet in 15 Minutes



Jen-Hsun Huang

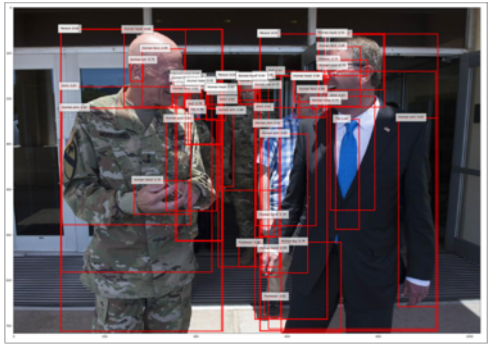
NVIDIA CEO, at SC'17

Achievement on **MN-1b**: PFDet in OIC 2018



Achievement on **MN-1b**: PFDet in OIC 2018

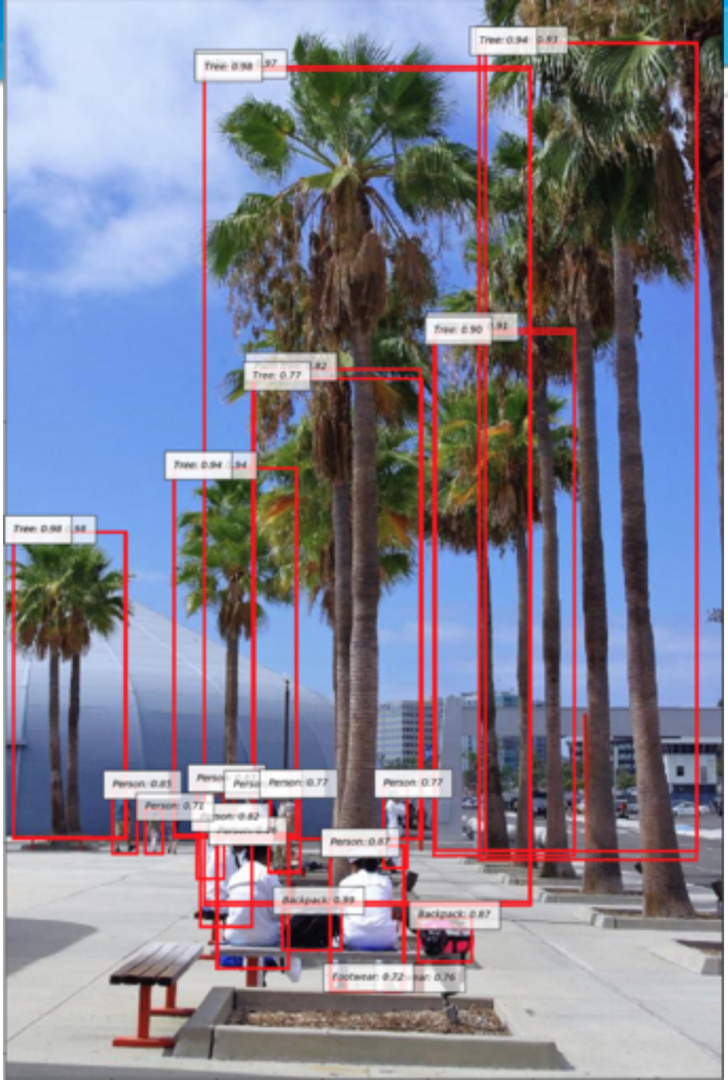
- Google AI Open Images - Object Detection Track
  - Competition using Largest-class image dataset
  - 12 million bounding boxes, 1.7 million images
  - 454 competitiors
  - Approx. 500GB (annotated subset)
- Object detection: much harder than object recognition task

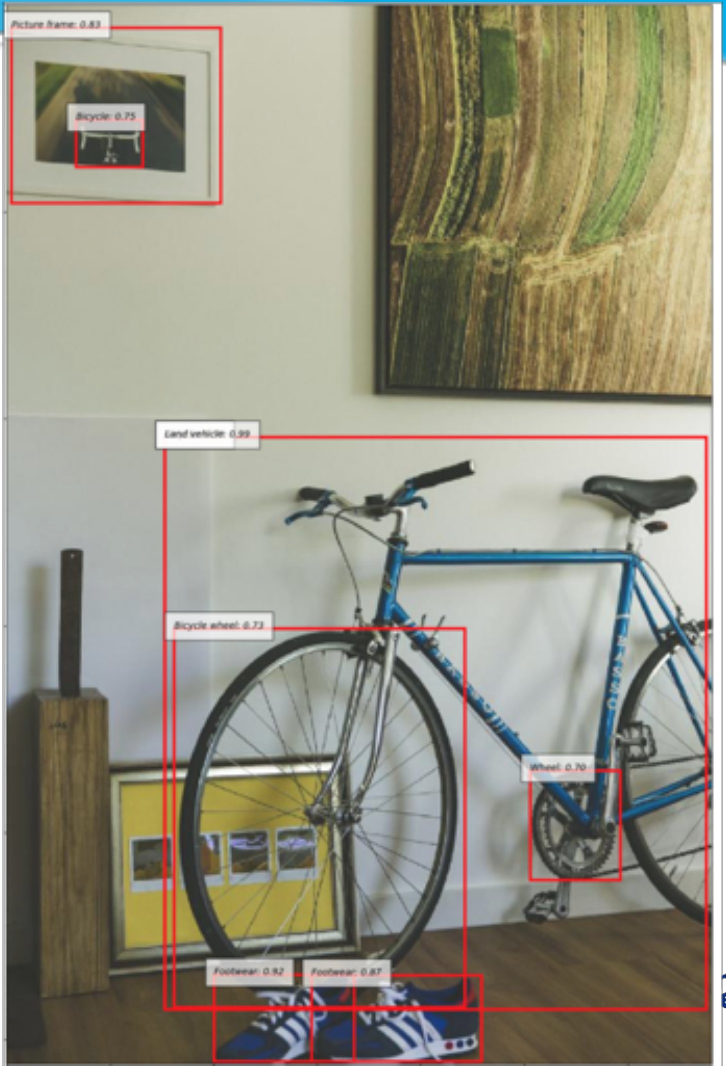


Object Detection

Detecting objects in an image by predicting...

- bounding boxes that contain them
- category of the objects


Tree: 0.94
Tree: 0.98
Tree: 0.90
Tree: 0.77
Tree: 0.94
Tree: 0.98
Person: 0.85
Person: 0.71
Person: 0.77
Person: 0.77
Person: 0.87
Backpack: 0.99
Backpack: 0.87
Footwear: 0.72


Picture frame: 0.83
Bicycle: 0.75
Land vehicle: 0.99
Bicycle wheel: 0.73
Wheel: 0.70
Footwear: 0.92
Footwear: 0.87

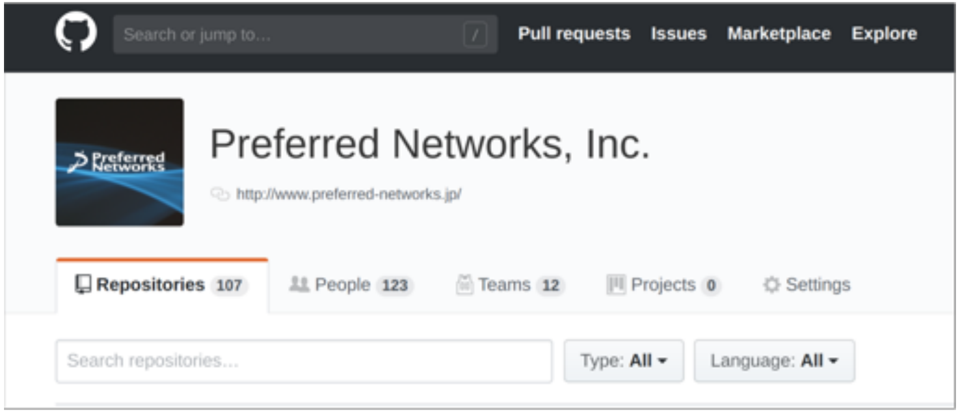
Achievement on **MN-1b**: PFDet in OIC 2018

- We won the 2nd position (0.023% diff to the 1st)

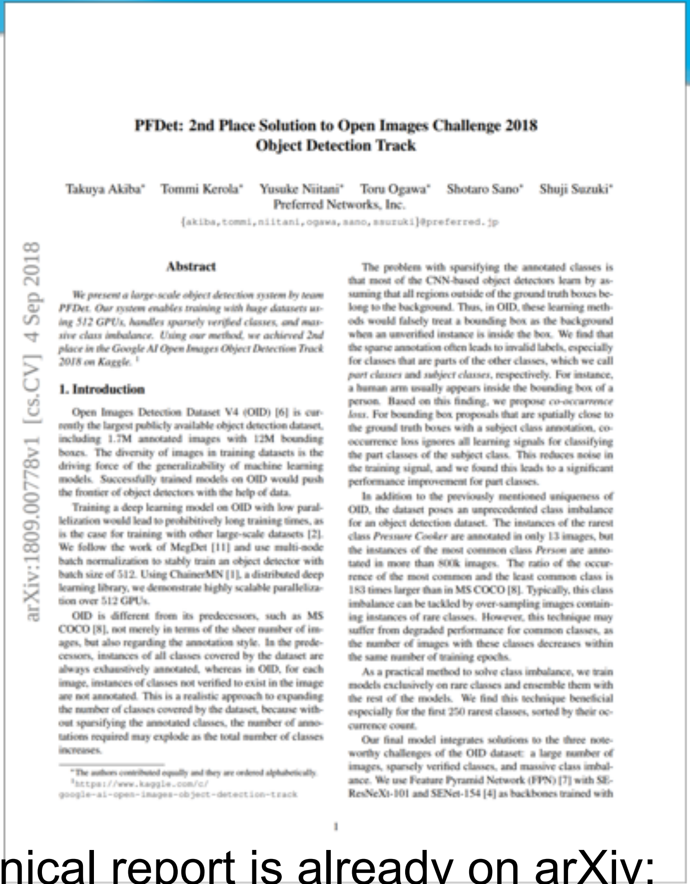
#	△pub	Team Name	Kernel	Team Members	Score	Entries	Last
1	▲2	kivajok			0.58657	102	2mo
2	▼1	PFDet		+3	0.58634	49	2mo
3	▼1	Avengers			0.58616	64	2mo
4	—	XJTU			0.58348	22	2mo
5	—	ikciting		+5	0.56801	39	2mo
6	—	Sogou_MM			0.53909	105	2mo
7	—	QLearning			0.53709	20	2mo

Congratulations to the winners again!
Object Detection track
1st place: kivajok
Bai du 百度
2d place: PFDet
Preferred Networks
3d place: Avengers
Bai du 百度
Visual Relationship Detection track
1st place: Seiji
2d place: tito (individual)
3d place: Kyle (no disclosure)
4th place: toshif (individual)

Open Sourcing PFDet:



We may make the implementation public



arXiv:1809.00778v1 [cs.CV] 4 Sep 2018

PFDet: 2nd Place Solution to Open Images Challenge 2018 Object Detection Track

Takuya Akiba* Tommi Kerola* Yusuke Niitani* Toru Ogawa* Shotaro Sano* Shuji Suzuki*
Preferred Networks, Inc.
{akiba,tommi,niitani,ogawa,sano,ssuzuki}@preferred.jp

Abstract

We present a large-scale object detection system by team PFDet. Our system enables training with huge datasets using 512 GPUs, handles sparsely verified classes, and massive class imbalance. Using our method, we achieved 2nd place in the Google AI Open Images Object Detection Track 2018 on Kaggle. [1]

1. Introduction

Open Images Detection Dataset V4 (OID) [6] is currently the largest publicly available object detection dataset, including 1.7M annotated images with 12M bounding boxes. The diversity of images in training datasets is the driving force of the generalizability of machine learning models. Successfully trained models on OID would push the frontier of object detectors with the help of data.

Training a deep learning model on OID with low parallelization would lead to prohibitively long training times, as is the case for training with other large-scale datasets [2]. We follow the work of MegDet [11] and use multi-node batch normalization to stably train an object detector with batch size of 512. Using ChainerMN [1], a distributed deep learning library, we demonstrate highly scalable parallelization over 512 GPUs.

OID is different from its predecessors, such as MS COCO [8], not merely in terms of the sheer number of images, but also regarding the annotation style. In the predecessors, instances of all classes covered by the dataset are always exhaustively annotated, whereas in OID, for each image, instances of classes not verified to exist in the image are not annotated. This is a realistic approach to expanding the number of classes covered by the dataset, because without sparsifying the annotated classes, the number of annotations required may explode as the total number of classes increases.

The problem with sparsifying the annotated classes is that most of the CNN-based object detectors learn by assuming that all regions outside of the ground truth boxes belong to the background. Thus, in OID, these learning methods would falsely treat a bounding box as the background when an unverified instance is inside the box. We find that the sparse annotation often leads to invalid labels, especially for classes that are parts of the other classes, which we call *part classes* and *subject classes*, respectively. For instance, a human arm usually appears inside the bounding box of a person. Based on this finding, we propose *co-occurrence loss*. For bounding box proposals that are spatially close to the ground truth boxes with a subject class annotation, co-occurrence loss ignores all learning signals for classifying the part classes of the subject class. This reduces noise in the training signal, and we found this leads to a significant performance improvement for part classes.

In addition to the previously mentioned uniqueness of OID, the dataset poses an unprecedented class imbalance for an object detection dataset. The instances of the rarest class *Pressure Cooker* are annotated in only 13 images, but the instances of the most common class *Person* are annotated in more than 800k images. The ratio of the occurrence of the most common and the least common class is 183 times larger than in MS COCO [8]. Typically, this class imbalance can be tackled by over-sampling images containing instances of rare classes. However, this technique may suffer from degraded performance for common classes, as the number of images with these classes decreases within the same number of training epochs.

As a practical method to solve class imbalance, we train models exclusively on rare classes and ensemble them with the rest of the models. We find this technique beneficial especially for the first 250 rarest classes, sorted by their occurrence count.

Our final model integrates solutions to the three noteworthy challenges of the OID dataset: a large number of images, sparsely verified classes, and massive class imbalance. We use Feature Pyramid Network (FPN) [7] with SE-ResNeXt-101 and SENet-154 [4] as backbones trained with

*The authors contributed equally and they are ordered alphabetically.
[1] https://www.kaggle.com/c/google-ai-open-images-object-detection-track

1

Technical report is already on arXiv:
arXiv:1809.00778

Computation resources used in PFDet

- Single training process of 16 epochs takes **33 hours** with **512 x V100** GPUs of MN-1b
- Repeated model development & parameter tuning

June 3rd:
First commit

June 29th:
FPN added

Aug. 31st:
Finish

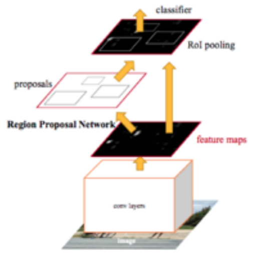
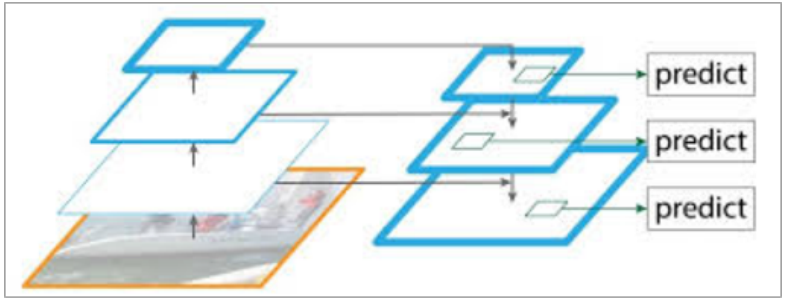


Figure 2: Faster R-CNN is a single, unified network for object detection. The RPN module serves as the 'attention' of this unified network.



Autonomous Tyding-up robot

Integration of a wide range of DL:

- Object Detection
  (based on PFDet)
- Audio recognition
- NLP
- Picking planning

Technical topics

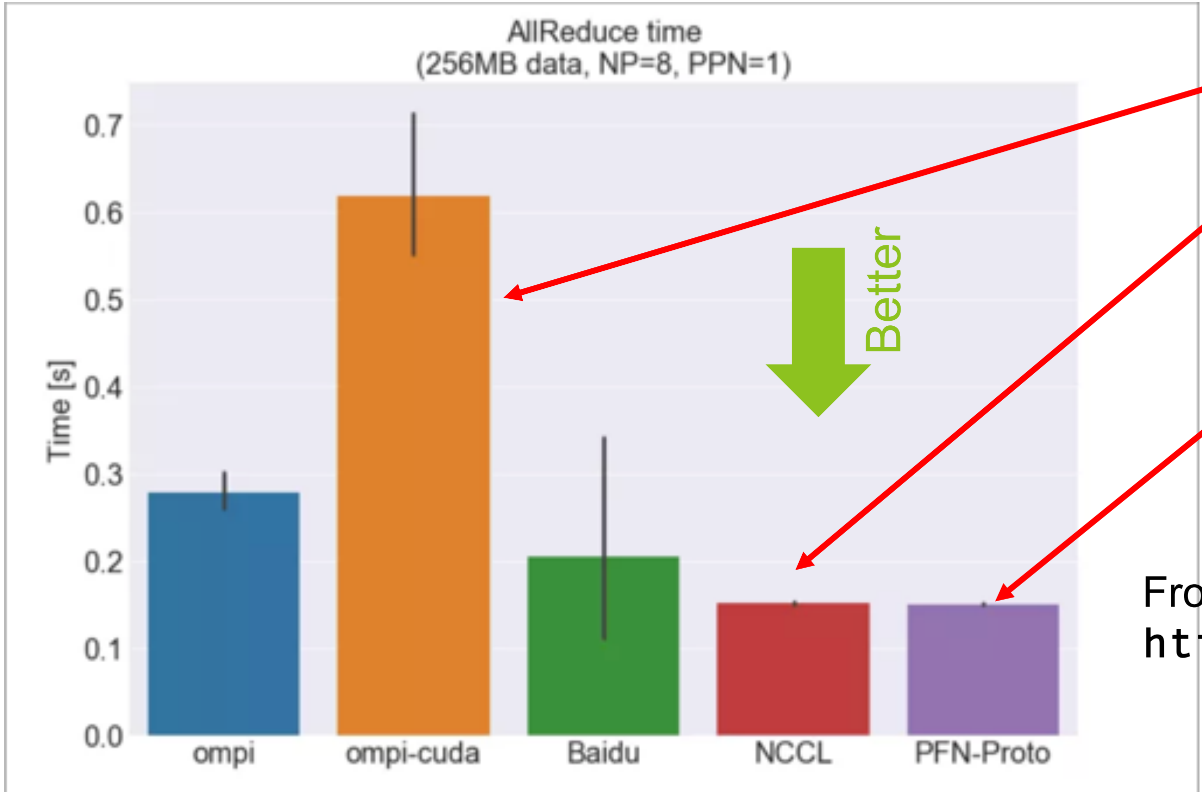
1. Communication & Fault tolerance
2. Storage
3. Resource Management

Communication

Allreduce: that always matters

- Critical component for distributed deep learning
- NVIDIA NCCL is the de-facto standard communication library
- Typical usage in ChainerMN:
  - MPI for process coordination
  - NCCL for actual communication

Communication



Open MPI 2.1.3
(configured as CUDA-aware)

NCCL is the fastest production-ready library

Our demonstration-purposed prototype library

From our research blog:
`https://bit.ly/2Pt2Gcg`

Communication

NCCL is now open-sourced 🎉

- Very interesting internal architecture
- NCCL still has some bug in large scale execution… we know what to do to an OSS software? ☺

Scaling Deep Learning Training with NCCL

By Sylvain Jeaugey | September 26, 2018 Tags: Accelerated Computing, DGX-1, DGX-2, InfiniBand, NCCL, NVLink, open source

NVIDIA Collective Communications Library (NCCL) provides optimized implementation of inter-GPU commun
Developers using deep learning frameworks can rely on NCCL's highly optimized, MPI compatible and topolo
available GPUs within and across multiple nodes.

NCCL is optimized for high bandwidth and low latency over PCIe and NVLink high speed interconnect for intr
for inter-node communication. NCCL—allows CUDA applications and DL frameworks in particular—to efficie
complex communication algorithms and adapt them to every platform.

The latest NCCL 2.3 release makes NCCL fully open-source and available on GitHub. The pre-built and teste
available on Developer Zone. This should provide you with the flexibility you need and enable us to have open

Preferred Networks

Communication: open questions (1)

- NCCL is optimized for bandwidth-bound (i.e. large buffer) communication

- What about short-messages?
  - Inter-process Batch Normalization
  - Model parallelism

- Fault tolerance?
  - NCCL now supports communication timeout...
  - But the MPI standard does not support FT (yet?)

Communication: open questions (2)

- Do we still use MPI…?
  - Legacy
  - The software stack is huge: installation and maintenance is hard for non-HPC users
  - No fault tolerance
  - No separation of process coordination and communication
  - Hard to use with Kubernetes

Resource Management

- We deploy Kubernetes on our computing cluster.

- Why Kubernetes?
  - Advanced features from the cloud computing community
    - Fault tolerance, dynamic job size management, preemption, etc.
  - We also deploy server-based services (such as JupyterHub, CI, etc.) on the same cluster

- Still many challenges
  - Batch job scheduling,

Resource Management: Open Questions

- Preemption is a strong tool for efficient resource management
  - How DL framework and scheduler cooperate ?
  - Job preemption/restarting & flexible resource (re-)allocation

- Resource isolation
  - Bandwidth isolation for Infiniband?
  - NUMA- / topology- aware pod placement?

NOTE: We operate an in-house cluster, so all jobs are our own.

Conclusion & Takeaways:

- PFN runs a cluster of 1,500 NVIDIA GPUs
- We develop real-world solutions as well as tackle challenging competitions.
- We deploy advanced & challenging software to manage the computing resources

- And… we are hiring!

Preferred Networks

Questions?

If you have in-depth technical questions, please send it to

Keisuke Fukuda <kfukuda@preferred.jp>

Thank you!