Highly Scalable Deep Learning Training System with Mixed Precision: Training ImageNet in Four Minutes

Tencent Inc., Hong Kong Baptist University

Presented by Xupeng Miao

2018/07/16

Outlines

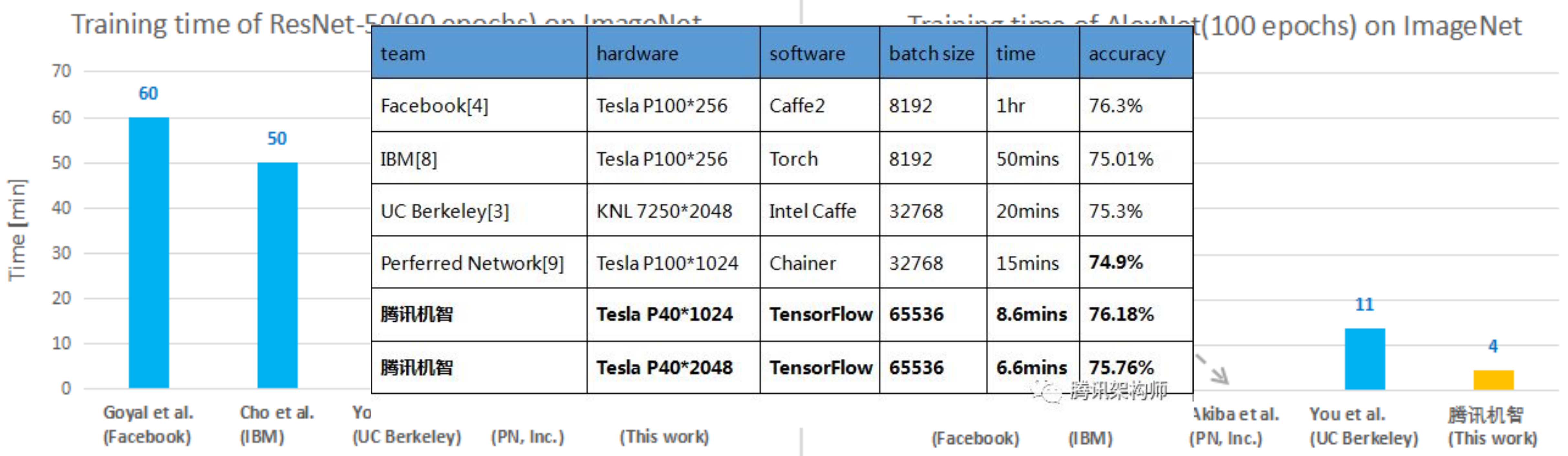
- Background
- Introduction
- System Overview
  - Mixed-Precision Training with LARS
  - Improvements on Model Architecture
  - Improvements on Communication Strategies
- Experiments

Background

- Large-scale deep neural networks with synchronized SGD

- Large mini-batch size
  - Improve the system scalability by reducing the communication-to-computation ratio
  - Hurt the generalization ability of the models

Background



team	hardware	software	batch size	time	accuracy
Facebook[4]	Tesla P100*256	Caffe2	8192	1hr	76.3%
IBM[8]	Tesla P100*256	Torch	8192	50mins	75.01%
UC Berkeley[3]	KNL 7250*2048	Intel Caffe	32768	20mins	75.3%
Perferred Network[9]	Tesla P100*1024	Chainer	32768	15mins	**74.9%**
腾讯机智	**Tesla P40*1024**	**TensorFlow**	**65536**	**8.6mins**	**76.18%**
腾讯机智	**Tesla P40*2048**	**TensorFlow**	**65536**	**6.6mins**	**75.76%**

GPU	SP performance	Memory	Bandwidth
P40	12 TFlops	24 GB	346 GB/s
P100	9.3 TFlops	16 GB	732 GB/s

Introduction

- Challenge
  - Large mini-batch size often leads to generalization gap
  - Large clusters is hard to achieve near-linear scalability.

Introduction

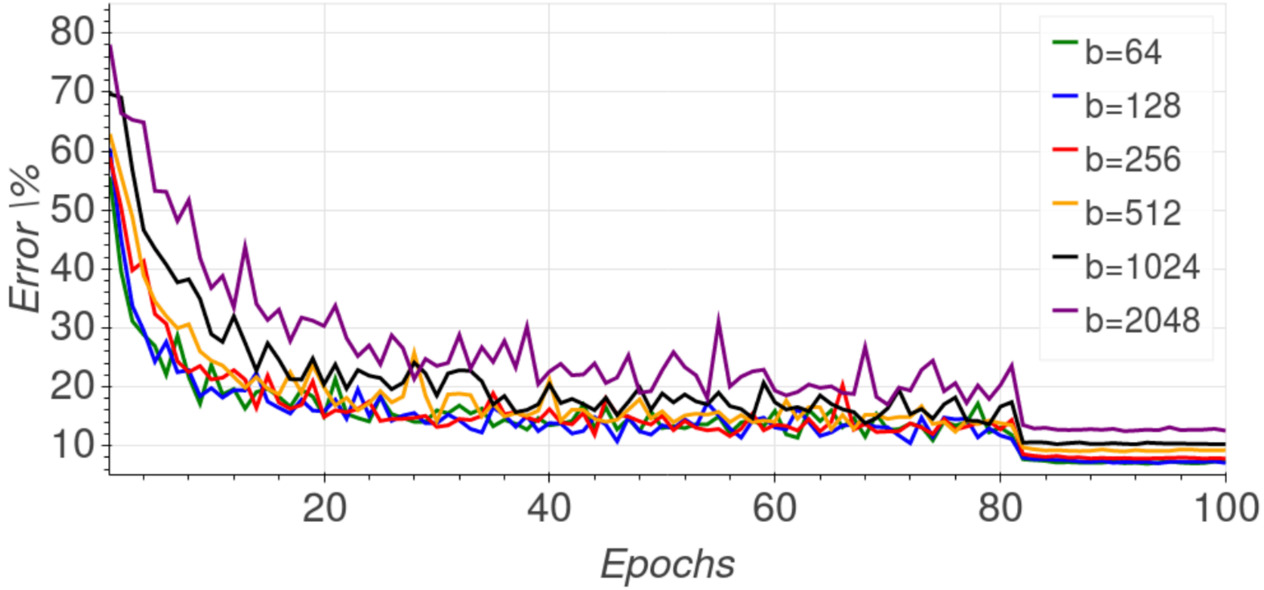
- Large mini-batch
  - Less update and Less communication
  - Less variance of gradients
  - Equivalent to decaying the learning rate to some degree

$$\mathrm{Var(g)} = \mathrm{Var}\left(\frac{1}{m}\sum_{i=1}^{m} g(x_i, y_i)\right) = \frac{1}{m^2} Var\big(g(x_1, y_1) + g(x_2, y_2) + \cdots + g(x_m, y_m)\big)$$

$$= \frac{1}{m^2} mVar\big(g(x_1, y_1)\big) = \frac{1}{m} Var\big(g(x_1, y_1)\big)$$

Introduction

- Large mini-batch
    - More epochs
    - Difficult to escape from saddle points/local minima



(b) Validation error

Introduction

- Facebook: ImageNet in 1 Hour

Linear Scaling Rule: *When the minibatch size is multiplied by k, multiply the learning rate by k.*

\+ **Warmup**

after k iterations $$w_{t+k} = w_t - \eta \frac{1}{n} \sum_{j<k} \sum_{x \in \mathcal{B}_j} \nabla l(x, w_{t+j})$$ minibatch size n

$\hat{\eta} = kn$ $$\hat{w}_{t+1} = w_t - \hat{\eta} \frac{1}{kn} \sum_{j<k} \sum_{x \in \mathcal{B}_j} \nabla l(x, w_t)$$ minibatch size kn

$$\nabla l(x, w_t) \approx \nabla l(x, w_{t+j})$$

Introduction

- Facebook: ImageNet in 1 Hour

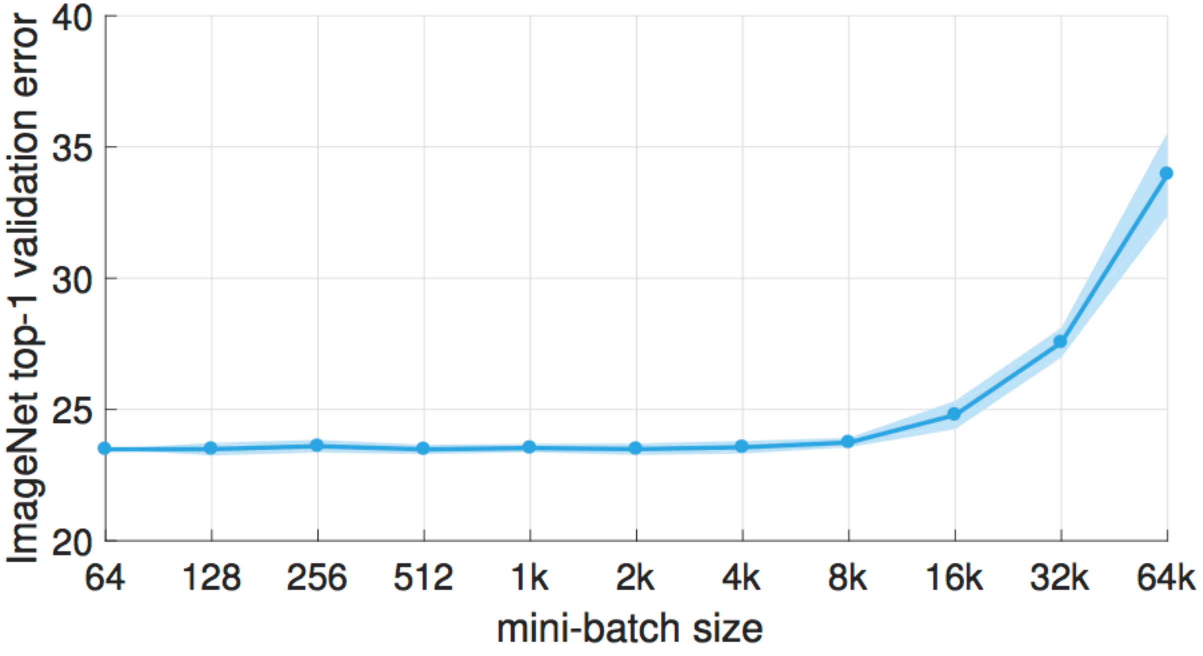


Figure 1. **ImageNet top-1 validation error *vs*. minibatch size.**

System Overview

- Mixed-Precision Training with LARS

$$\Delta w_t^l = \gamma \cdot \eta \cdot \frac{\|w^l\|}{\|\nabla L(w^l))\|} \cdot \nabla L(w_t^l)$$

Table 1: Effectiveness of using LARS on ResNet-50

Mini-Batch Size	Number of Epochs	LARS	Top-1 Accuracy
64K	90	NO	73.2%
64K	90	YES	76.2%

System Overview

- Improvements on Model Architecture

$$E(w) = E_0(w) + \frac{1}{2}\lambda \sum_i w_i^2$$

$$\hat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma^2 + \epsilon}}$$

$$y_i \leftarrow \gamma \hat{x}_i + \beta \equiv BN_{\gamma,\beta}(x_i)$$

Table 2: Effect of Regularization with b, β and γ for AlexNet

Batch	Epochs	Regularize b, β and γ	Top1
64K	95	Yes	55.8%
64K	95	No	57.1%

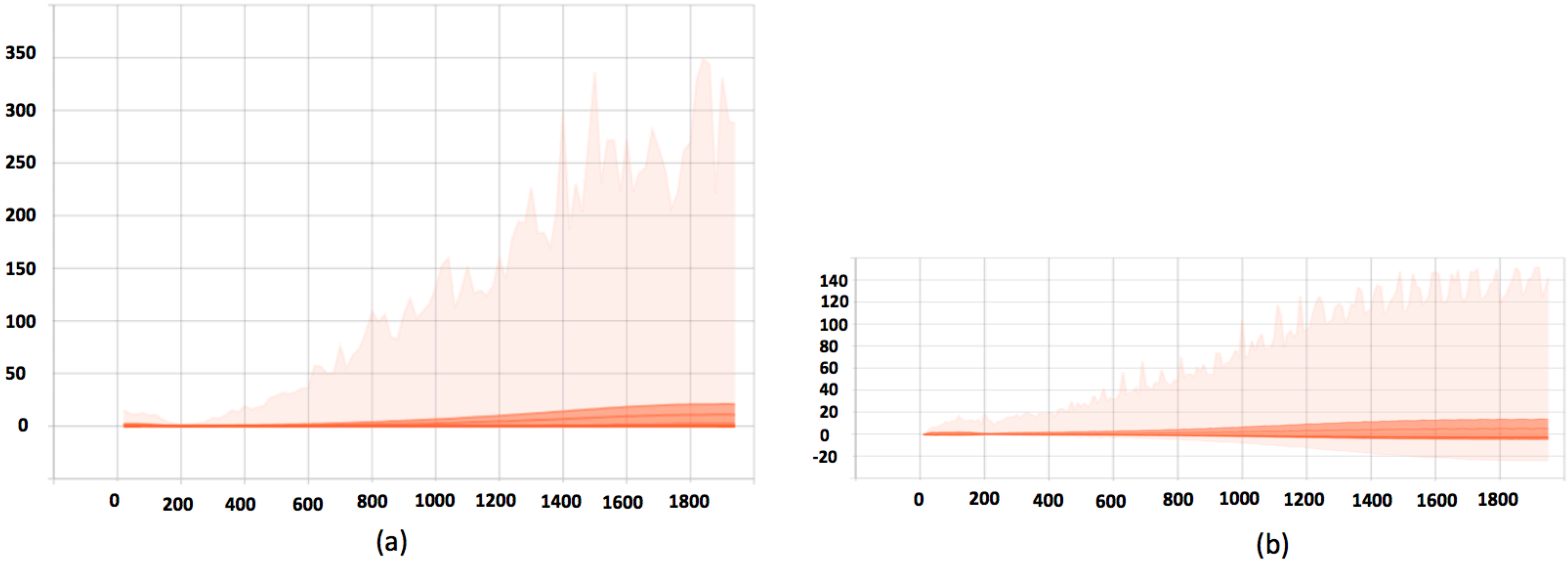
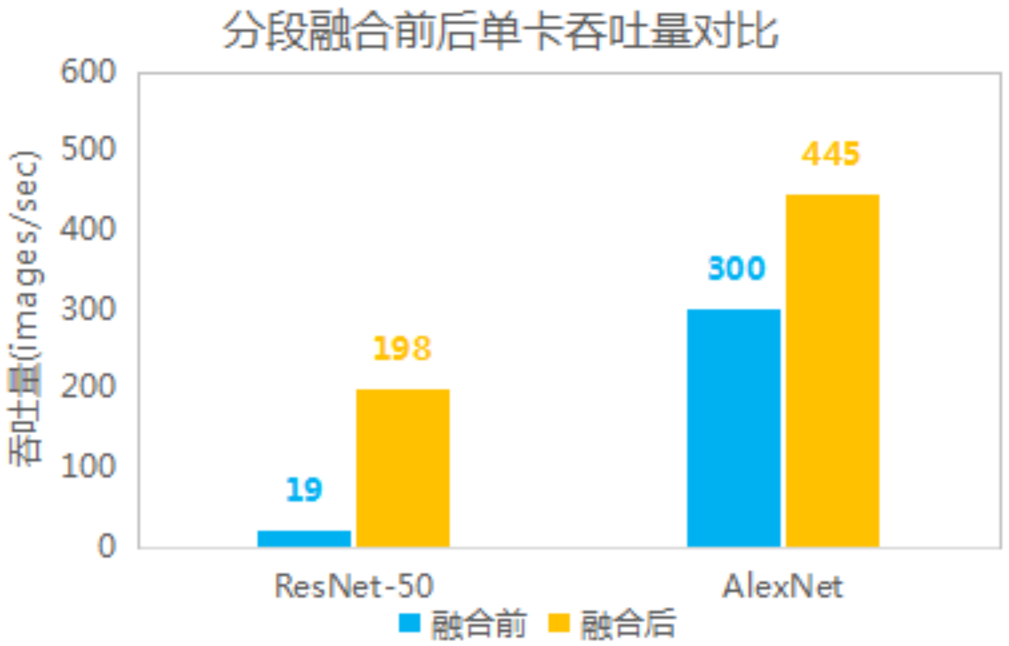


Figure 4: Feature Map Distribution of Pool5(a) and Pool5-BN5(b) of AlexNet as shown in Figure 3. (the horizontal axis is the training steps, the vertical axis is the feature map distributions.)

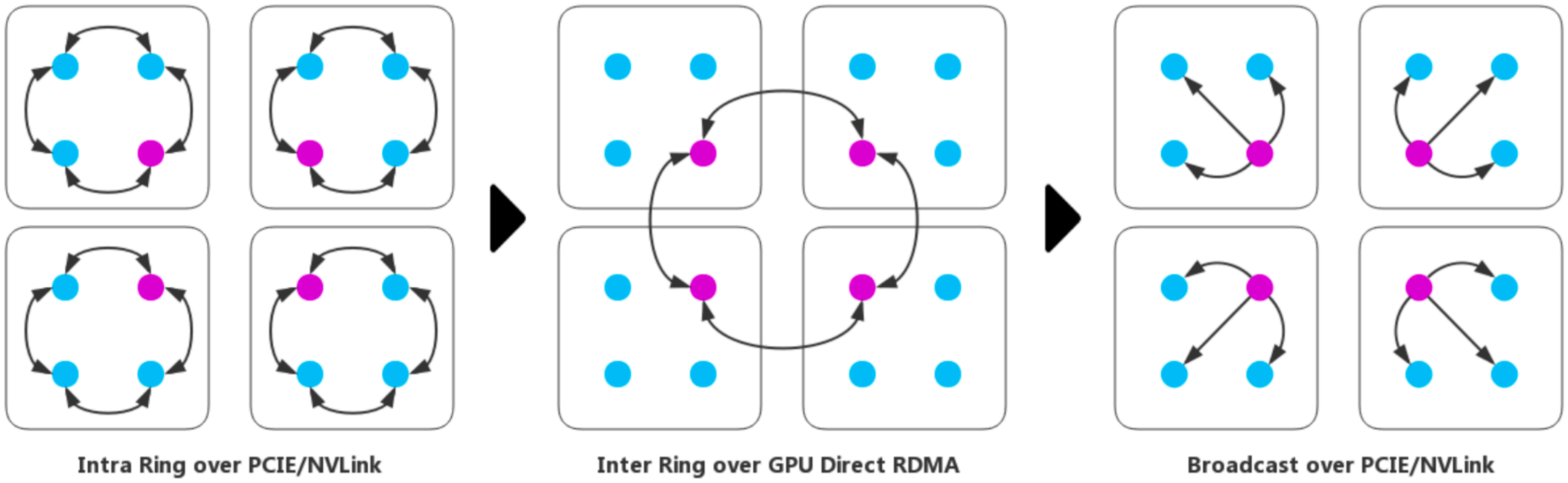
System Overview

- Improvements on Communication Strategies
  - Tensor Fusion



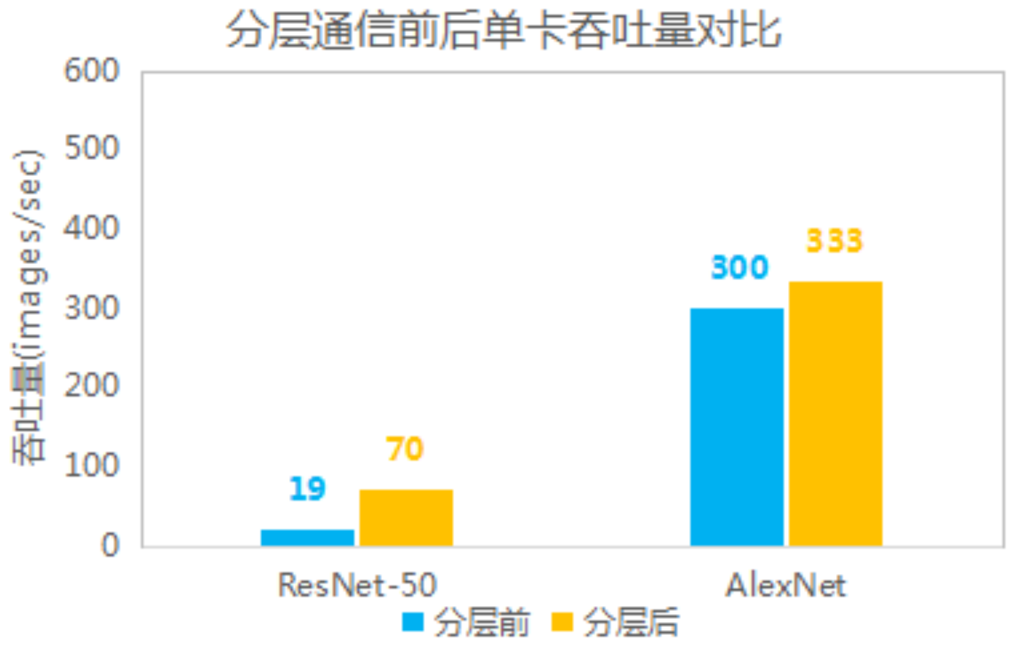
System Overview

- Improvements on Communication Strategies
  - Tensor Fusion
  - Hierarchical All-reduce



System Overview

- Improvements on Communication Strategies
  - Tensor Fusion
  - Hierarchical All-reduce



System Overview

- Improvements on Communication Strategies
  - Tensor Fusion
  - Hierarchical All-reduce
  - Hybrid All-reduce

For fully-connected layers which usually have a much larger number of weights, ring-based all-reduce still outperforms our hierarchical all-reduce.

Experiments

Model	Input Size	Parameter Size	FLOPs	Baseline Top1
AlexNet	227x227	62M	727 M	58.8%
ResNet-50	224x224	25M	4 G	75.3%

Experiments

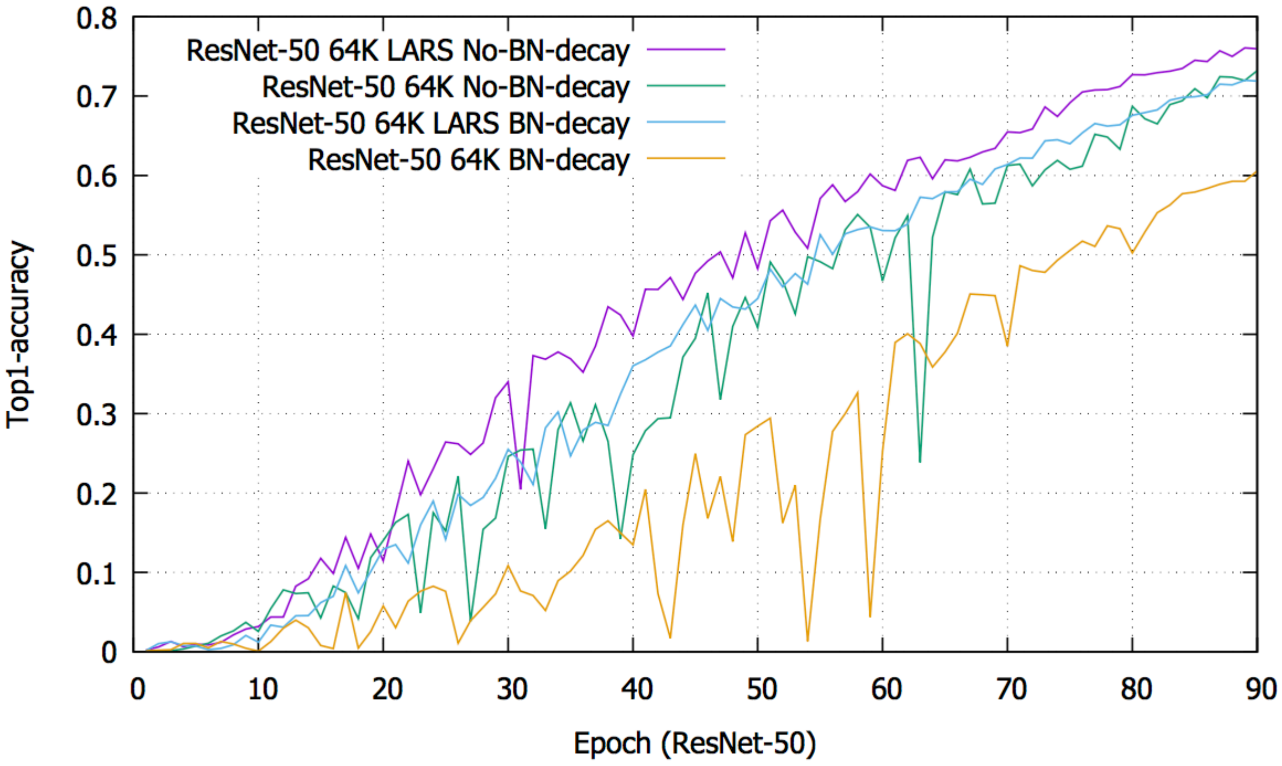


Figure 8: ImageNet Training with ResNet-50 Using 64K Mini-Batch Size

Table 4: Compare AlexNet training with different teams

Team	Batch	Hardware	Software	Top-1 Accuracy	Time
You et al. [27]	512	DGX-1 station	NVCaffe	58.8%	6h 10m
You et al. [27]	32K	CPU × 1024	Intel Caffe	58.6%	11min
This work	**64K**	Tesla P40 × 512	TensorFlow	**58.8%**	**5m**
This work	**64K**	Tesla P40 × 1024	TensorFlow	**58.7%**	**4m**

Table 5: Compare ResNet-50 training with different teams

Team	Batch	Hardware	Software	Top-1 Accuracy	Time
He et al. [13]	256	Tesla P100 × 8	Caffe	75.3%	29h
Goyal et al. [12]	8K	Tesla P100 × 256	Caffe2	76.3%	1h
Cho et al. [4]	8K	Tesla P100 × 256	Torch	75.0%	50min
Codreanu et al. [5]	32K	KNL × 1024	Intel Caffe	75.3%	42min
You et al. [27]	32K	KNL × 2048	Intel Caffe	75.4%	20min
Akiba et al. [2]	32K	Tesla P100 × 1024	Chainer	74.9%	15min
This work	**64K**	Tesla P40 × 1024	TensorFlow	**76.2%**	**8.7m**
This work	**64K**	Tesla P40 × 2048	TensorFlow	**75.8%**	**6.6m**

Experiments

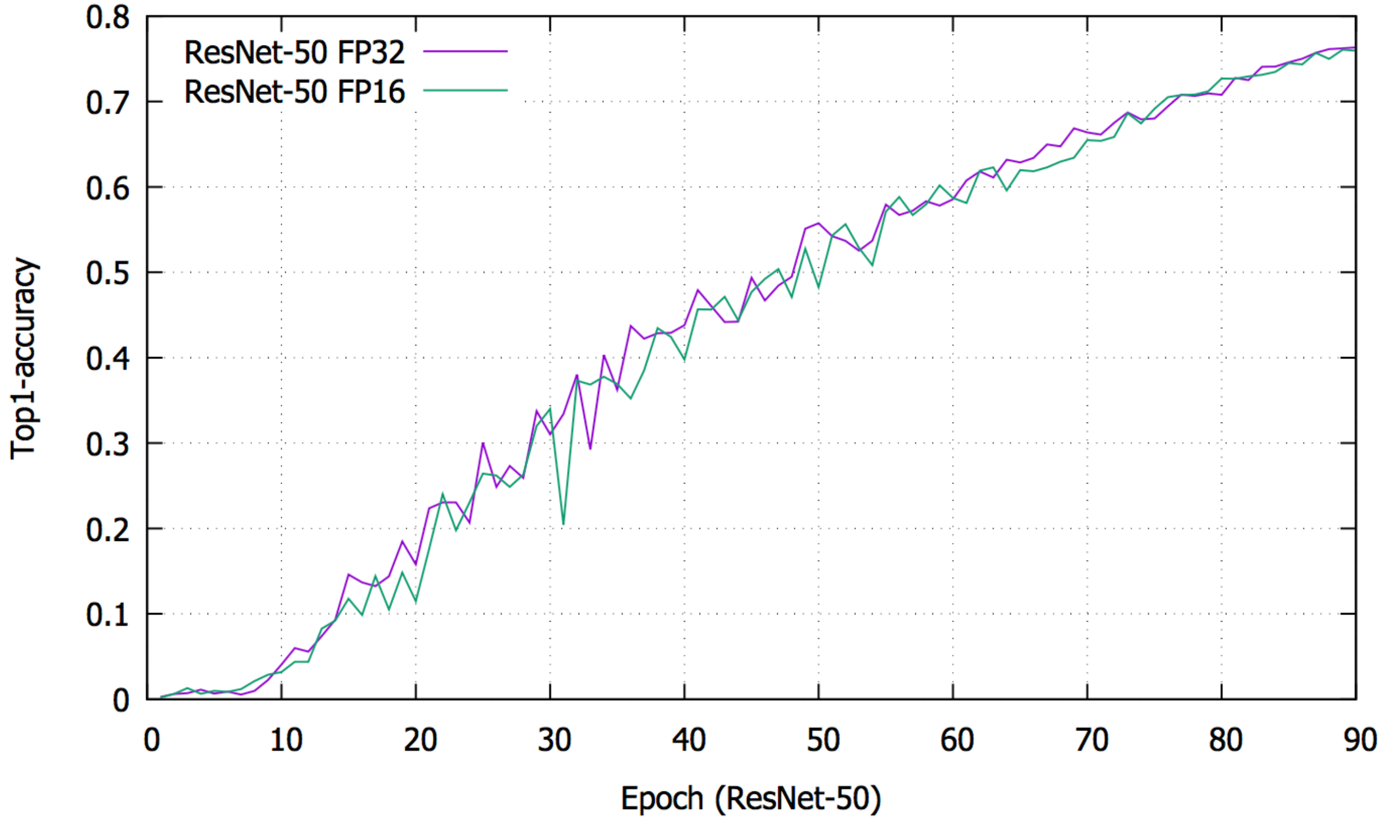


Figure 10: Compare the convergence of mixed-precision and single-precision training

Experiments

Table 6: Effect of LARS to ResNet-50 Training

Batch	LARS	Top-1 Accuracy
64K	×	60.6%
64K	✓	71.9%

Table 7: Effect of improvements to ResNet-50 Training

Batch	No Decay BN	Top1
64K	×	71.9%
64K	✓	76.2%

Experiments

Table 9: ResNet-50: Compare the speed of mixed-precision training and single-precision training

Batch/GPU	Data Type	Images/Sec
64	FP32	172
64	mixed	218

Experiments

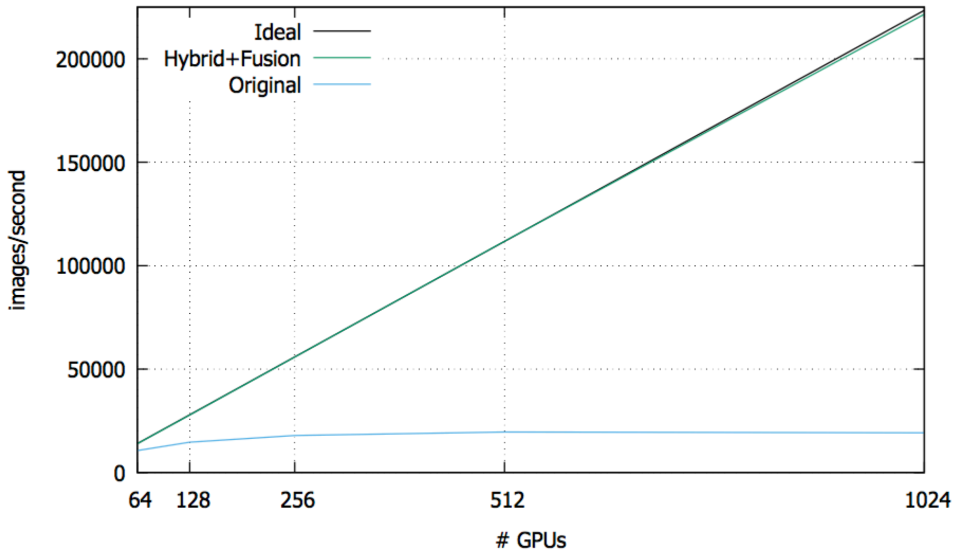


Figure 11: ResNet-50 training throughput with batch 64/GPU

99.2%

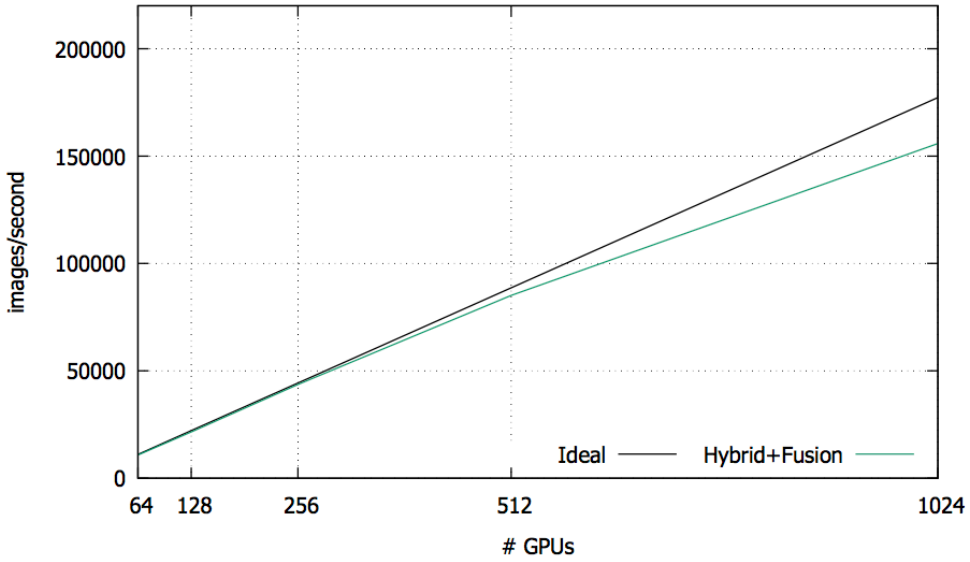


Figure 12: ResNet-50 training throughput with batch 32/GPU

87.9%

Experiments

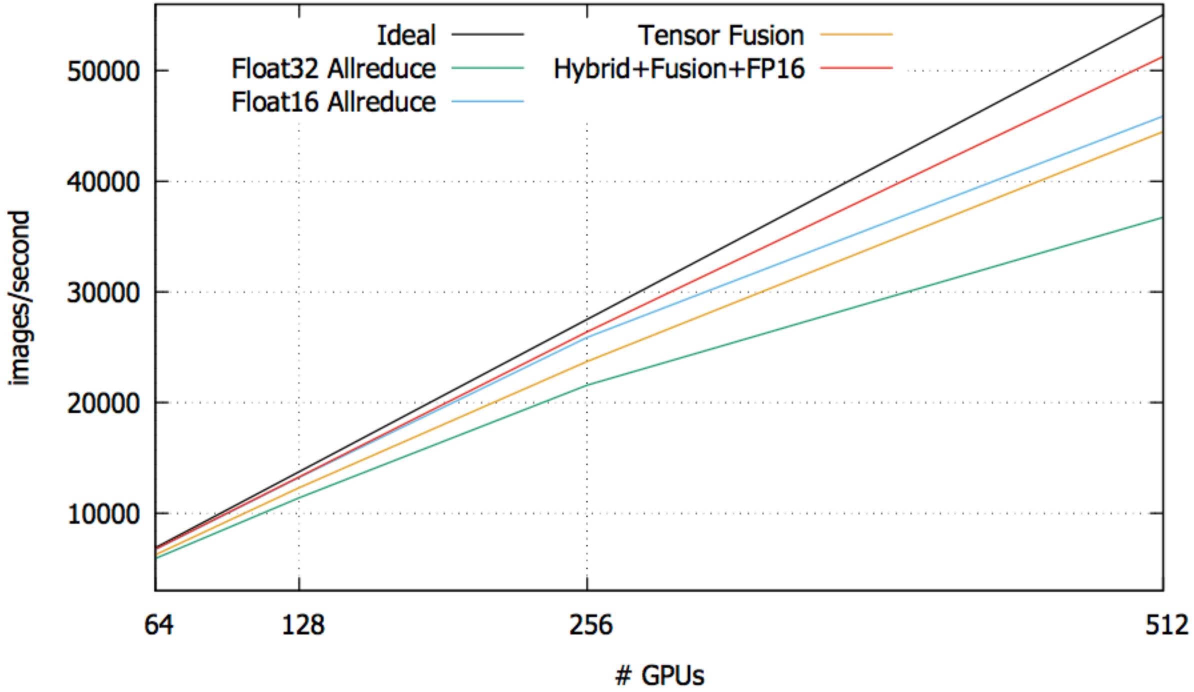
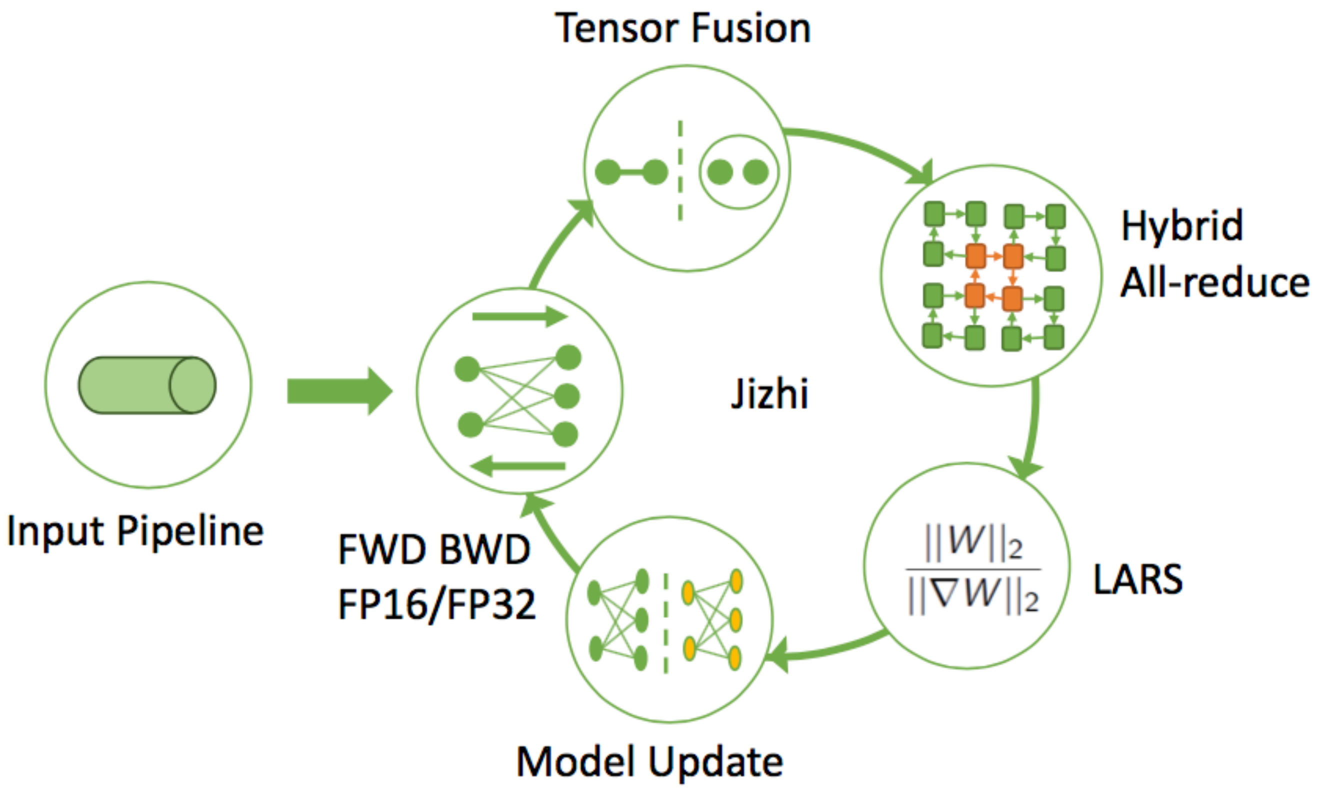


Figure 13: AlexNet training throughput with batch 128/GPU

91.4%

Conclusion


Tensor Fusion
Hybrid
All-reduce
Jizhi
Input Pipeline
FWD BWD
FP16/FP32
LARS
$\frac{||W||_2}{||\nabla W||_2}$
Model Update

Q&A