Scaling-up SGD with mini-batches

ECE826

Today

- Synchronous Distributed Optimization
- Distributing SGD effort with minibatches
- Performance of Distributed SGD

Stochastic Gradient Descent

$$\min_{\mathbf{w}} \frac{1}{n} \sum_{i=1}^{n} \ell(\mathbf{w}; \mathbf{z}_i)$$

loss for data point i

- **Idea** ('50s, '60s [Robbins, Monro], [Widrow, Hoff]):
  Sample a data point + locally optimize.

SGD: An *Über*-algorithm

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \gamma \cdot \nabla \ell(\mathbf{w}_k; \mathbf{z}_{i_k})$$

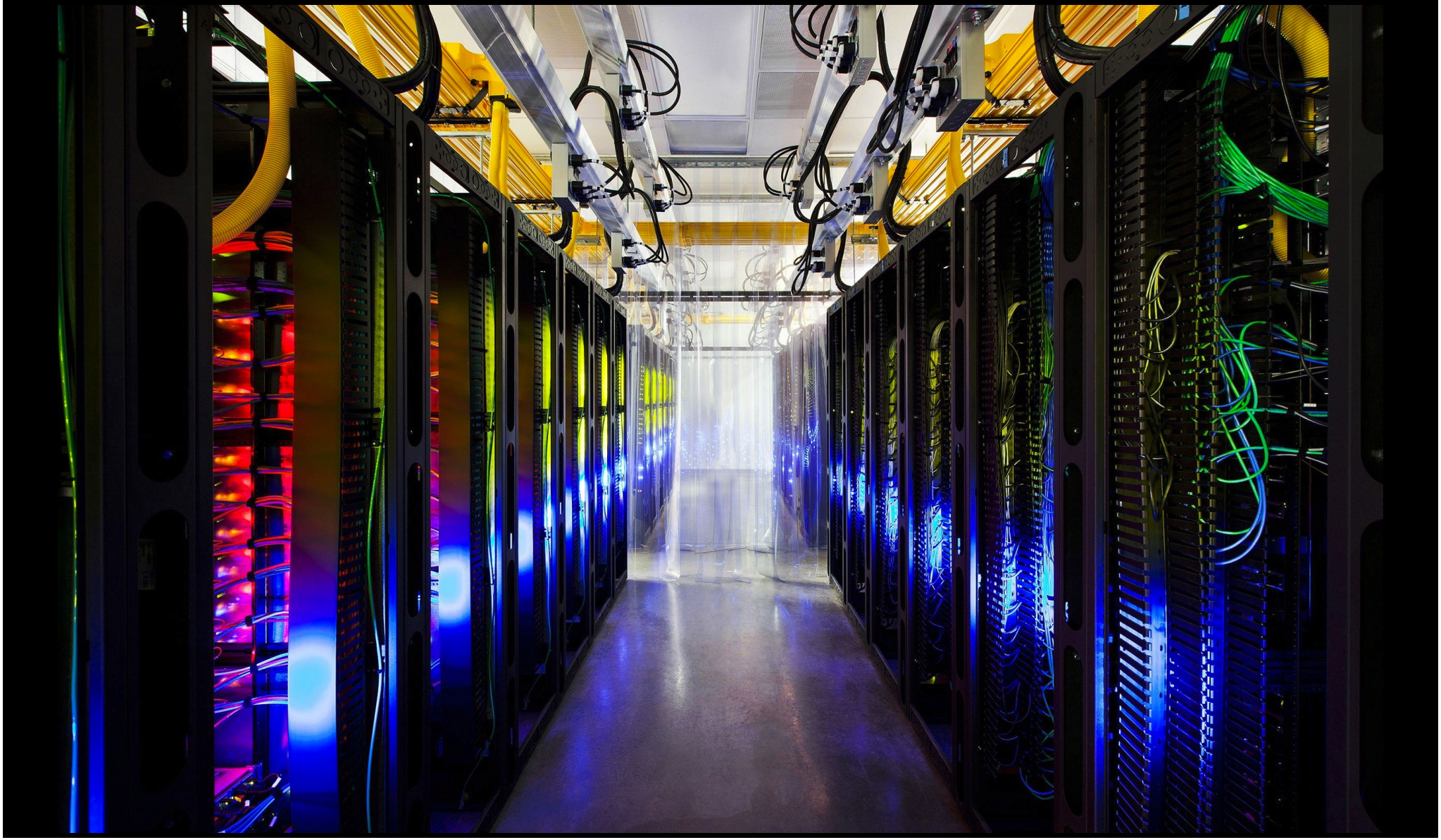
Stochastic Gradient Descent

SGD can take years on large data sets

Goal:

Speed up Machine Learning

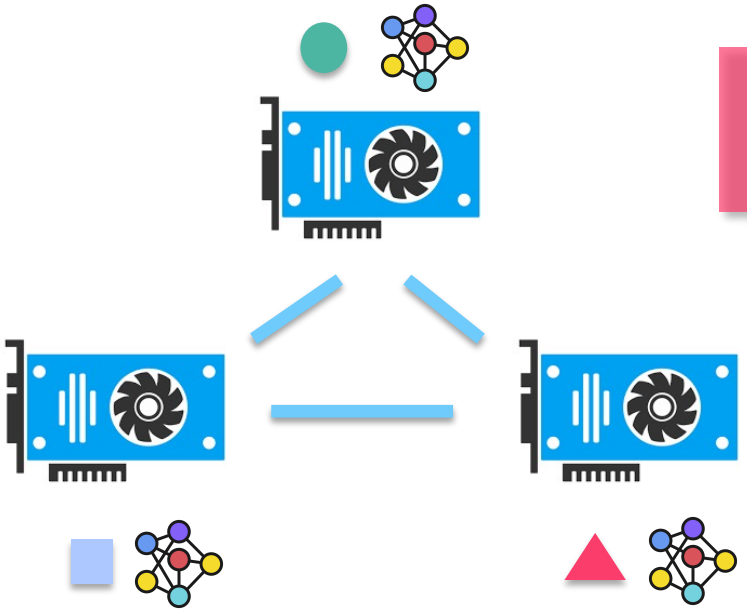
Idea:
Train at scale



Minibatch SGD

Algorithm of choice: minibatch SGD

All nodes compute gradients

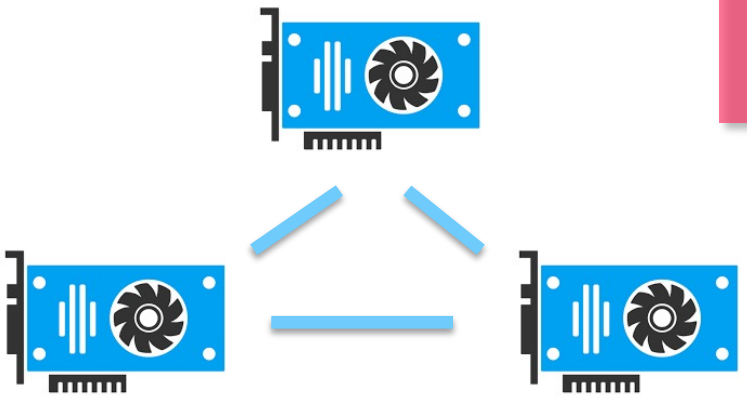


B = batch size,
gradients / iteration

Sergeev, et al. Horovod: fast and easy distributed deep learning in TensorFlow, https://arxiv.org/abs/1802.05799
https://github.com/baidu-research/baidu-allreduce

Algorithm of choice: minibatch SGD

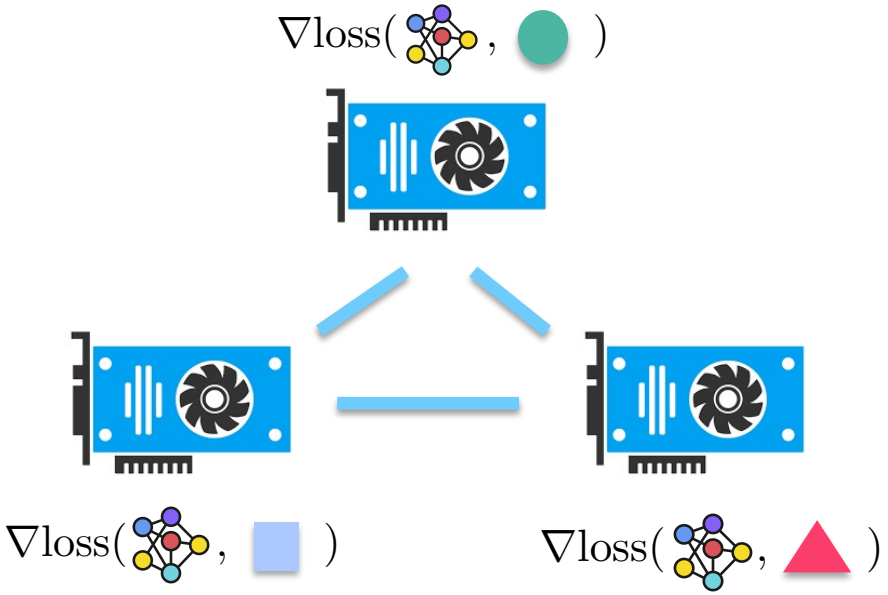
All nodes compute gradients



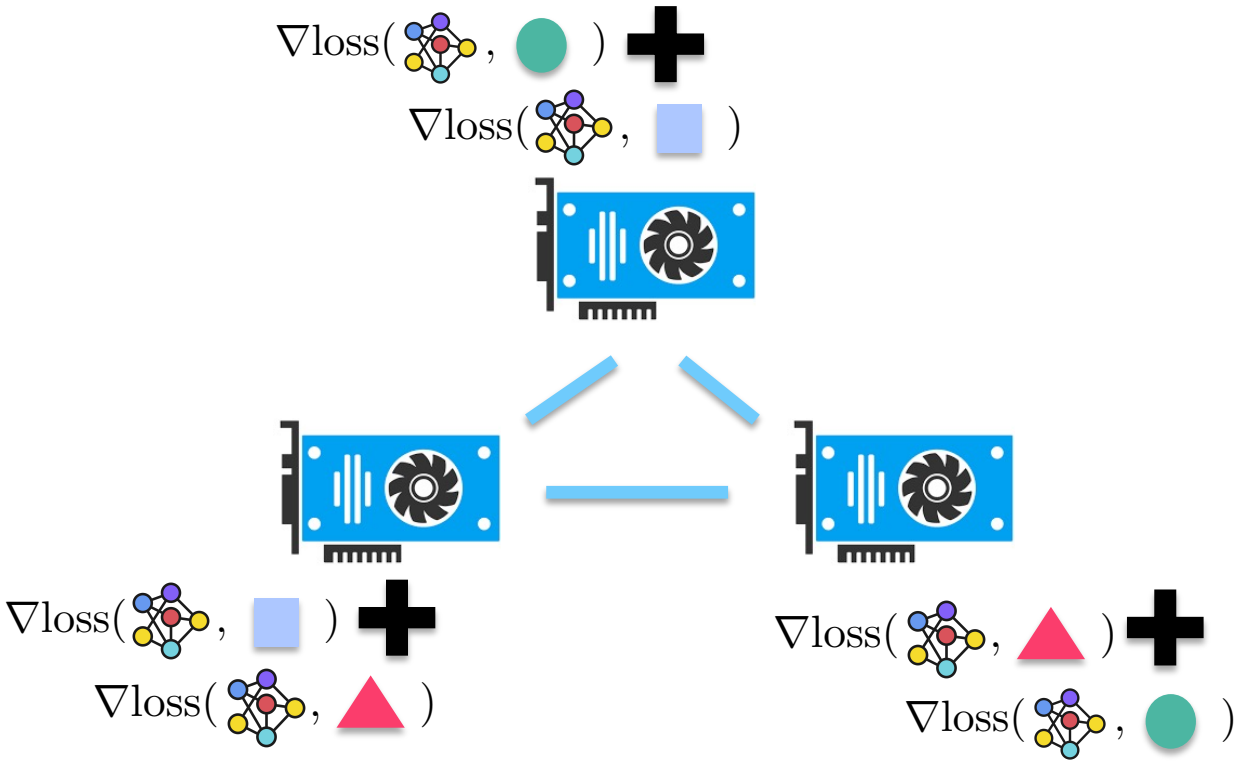
B = batch size,
gradients / iteration

Sergeev, et al. Horovod: fast and easy distributed deep learning in TensorFlow, https://arxiv.org/abs/1802.05799
https://github.com/baidu-research/baidu-allreduce

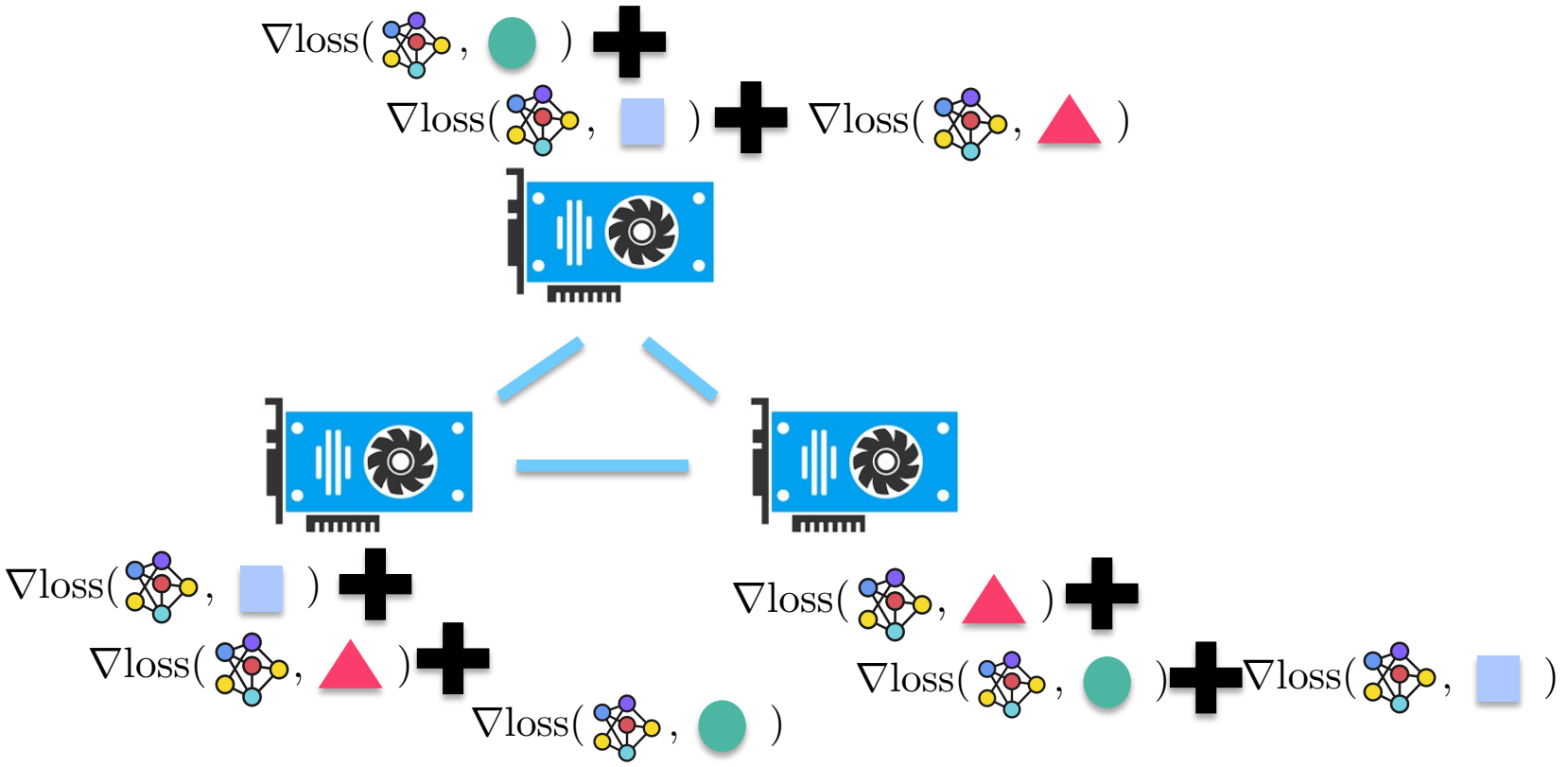
Algorithm of choice: minibatch SGD



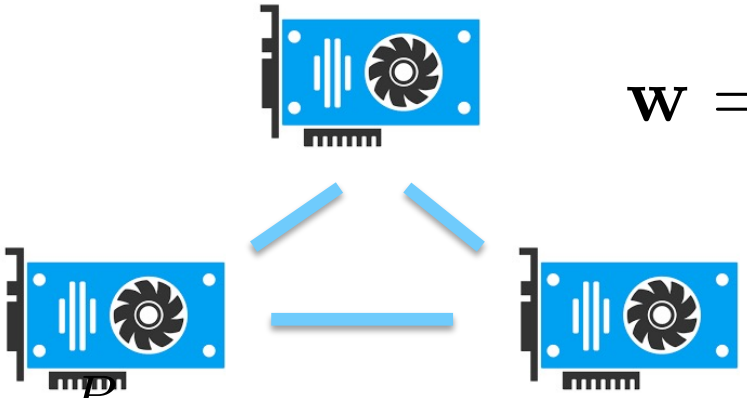
Algorithm of choice: minibatch SGD



Algorithm of choice: minibatch SGD



Algorithm of choice: minibatch SGD



$$\mathbf{w} = \mathbf{w} - \gamma \sum_{i=1}^{P} \mathbf{g}_i$$

$$\mathbf{w} = \mathbf{w} - \gamma \sum_{i=1}^{P} \mathbf{g}_i$$

$$\mathbf{w} = \mathbf{w} - \gamma \sum_{i=1}^{P} \mathbf{g}_i$$

Algorithm of choice: minibatch SGD

All nodes have the same model after each reduce round

Repeat until happy with model at hand

$$\mathbf{w} = \mathbf{w} - \gamma \sum_{i=1}^{P} \mathbf{g}_i$$

$$\mathbf{w} = \mathbf{w} - \gamma \sum_{i=1}^{P} \mathbf{g}_i$$

Potential issue?

- Compute multiple gradients in parallel

$$w_{k+1} = w_k - \gamma \nabla \ell_{s_k^1}(w_k; x_{s_k^1})$$

$$w_{k+1} = w_k - \gamma \nabla \ell_{s_k^2}(w_k; x_{s_k^2})$$

$$w_{k+1} = w_k - \gamma \nabla \ell_{s_k^3}(w_k; x_{s_k^3})$$

$$w_{k+1} = w_k - \gamma \nabla \ell_{s_k^4}(w_k; x_{s_k^4})$$

- Issue:

all 4 gradients computed on the same model

Q: Does it perform the same as SGD?

Evaluating the performance of mini-batch SGD

How to Analyze mini-batch?

- Measure of performance

$$\text{worst case speedup} = \frac{\text{bound on \#iter of SGD to } \epsilon}{\text{bound on \#iter of Parallel SGD to } \epsilon}$$

Main Question:
How does minibatch SGD compare against serial SGD?

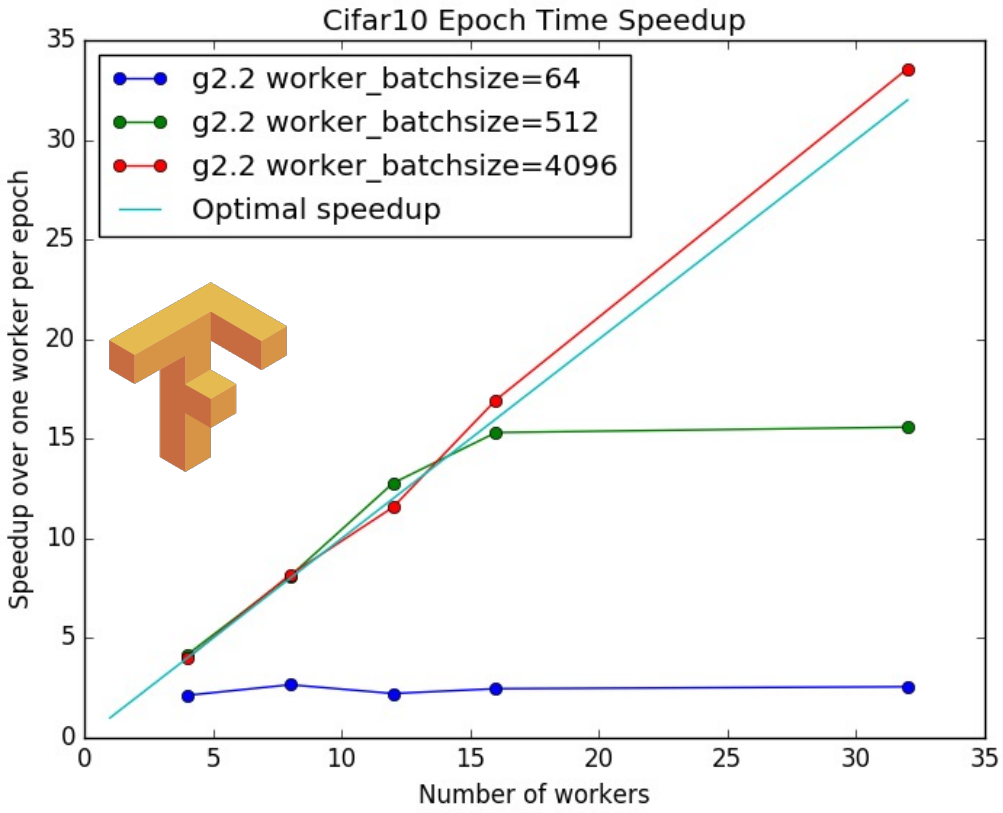
How to evaluate run-time

Two factors control run-time

Time to accuracy ε =
[time per data pass] X [#passes to accuracy ε]

Per iteration time

- TL;DR: **Becomes better with larger B**
- Why?



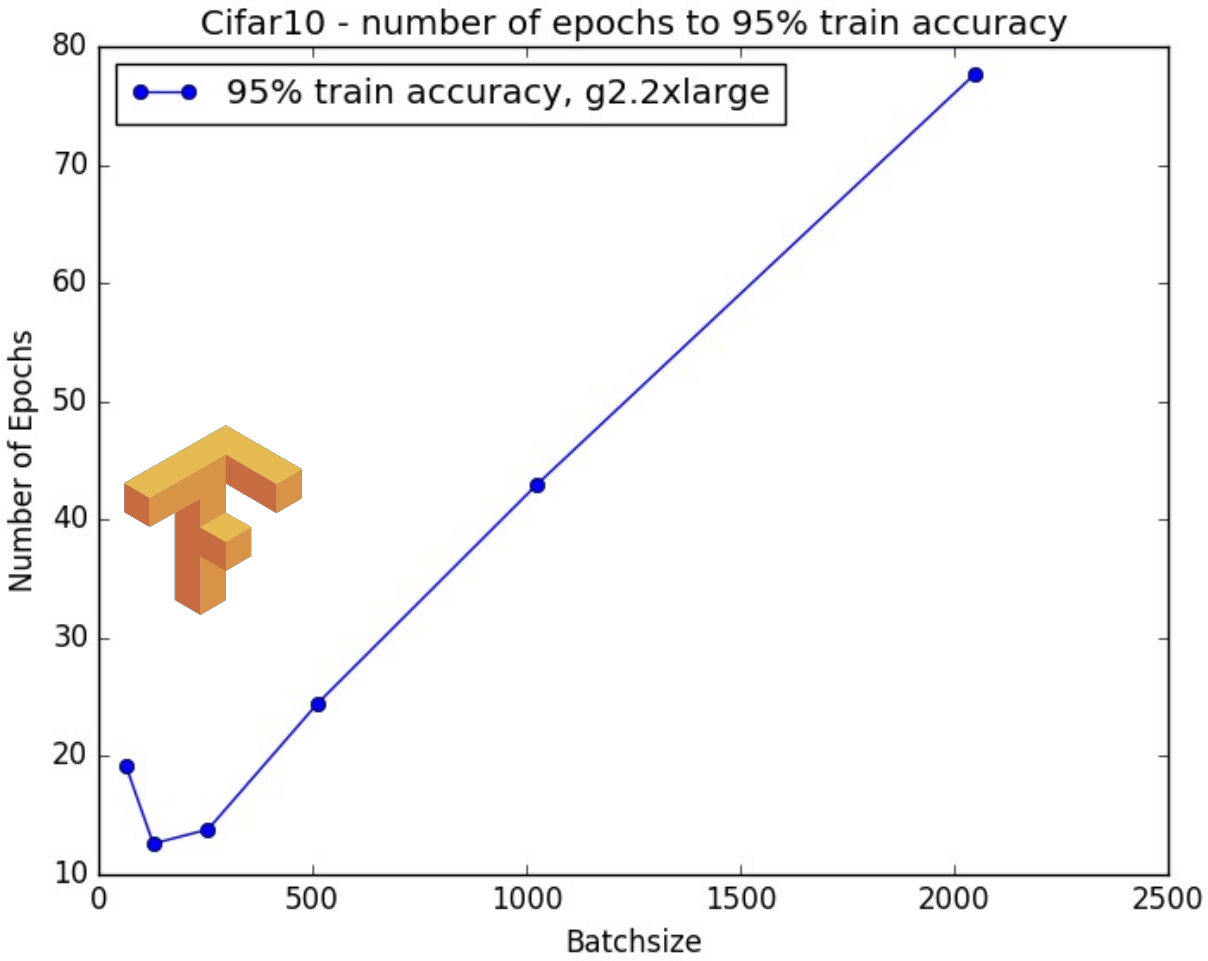
Time per pass:
time for dataset_size/batch_size distributed iterations

Bigger Batch *
⇒ Better GPU utilization

Bigger Batch
⇒ Less Communication
(smaller time per epoch)

Number of passes to ε accuracy

- TL;DR: Becomes worse with larger B



Large Batch
⇒ worse train error
(more #passes to accuracy **ε**)

WHY?

Widely observed issue

Accurate, Large Minibatch SGD:
Training ImageNet in 1 Hour

Priya Goyal Piotr Dollár Ross Girshick Pieter Noordhuis
Lukasz Wesolowski Aapo Kyrola Andrew Tulloch Yangqing Jia Kaiming He

Facebook

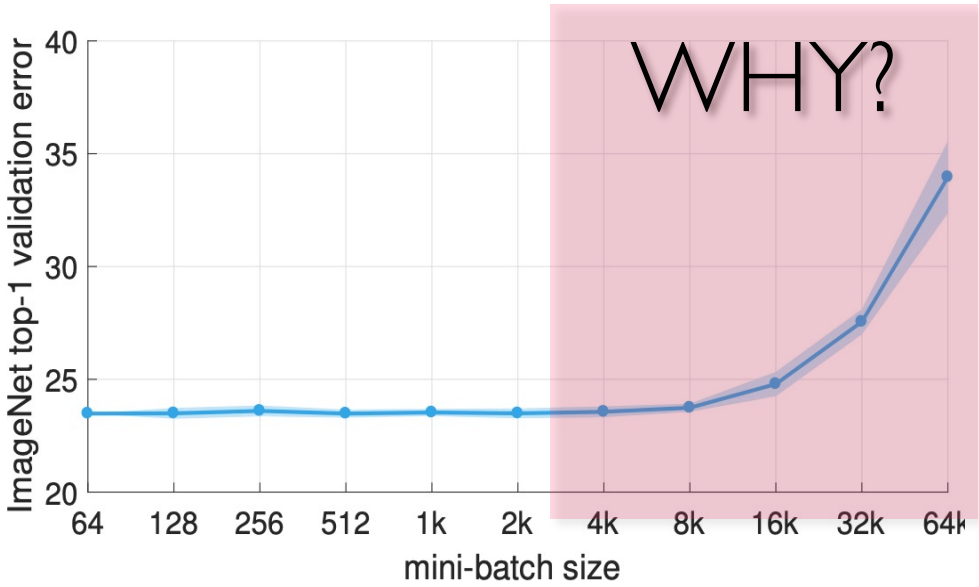
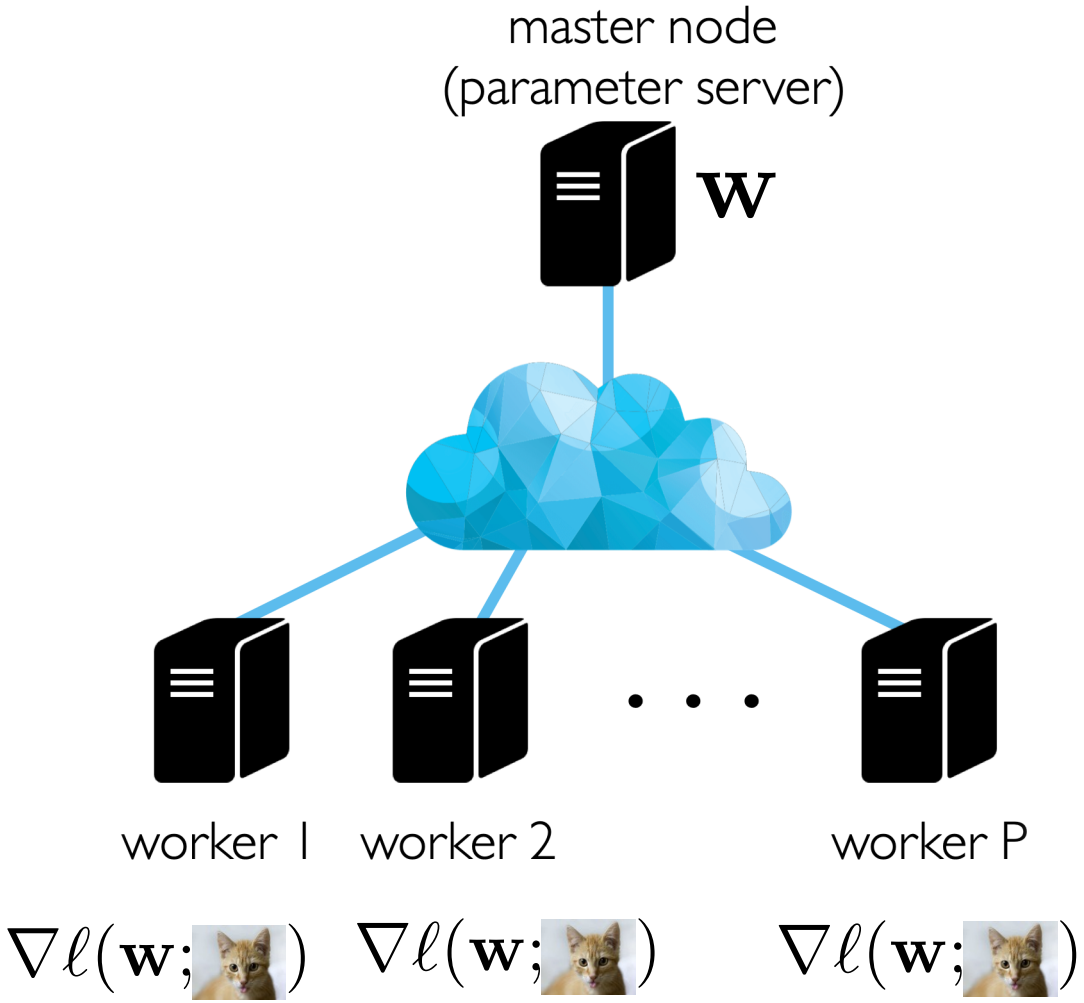


Figure 1. **ImageNet top-1 validation error *vs*. minibatch size.** Error range of plus/minus *two* standard deviations is shown. We present a simple and general technique for scaling distributed synchronous SGD to minibatches of up to 8k images *while maintaining the top-1 error of small minibatch training*. For all minibatch sizes we set the learning rate as a *linear* function of the minibatch size and apply a simple warmup phase for the first few epochs of training. All other hyper-parameters are kept fixed. Using this simple approach, accuracy of our models is invariant to minibatch size (up to an 8k minibatch size). Our techniques enable a linear reduction in training time with ~90% efficiency as we scale to large minibatch sizes, allowing us to train an accurate 8k minibatch ResNet-50 model in 1 hour on 256 GPUs.

High-level idea: "Similarity Hurts"



High-level idea: "Similarity Hurts"

master node

Having workers near identical gradients is useless!
=> no speedup

Really large batches
=> "similar" gradient updates

worker 1 worker 2 worker P

$\nabla \ell(\mathbf{w};$  $)$ $\nabla \ell(\mathbf{w};$  $)$ $\nabla \ell(\mathbf{w};$  $)$

1-sample SGD
vs
B-sample SGD

Convergence of mini-batch SGD

Main question:

How does mini-batch SGD compare to 1-sample SGD?

Idea: Compare T iterations of 1-sample SGD with T/B iterations of minibatch SGD

Hope: under some assumptions updating the global model every B gradients is not a big issue.

A single iteration of 1-sample SGD

Progress in a single iteration:

$$\mathbb{E}\|w_1 - w^*\|^2 = \mathbb{E}\|w_0 - \gamma \nabla f_{s_1}(w_0) - w^*\|^2$$
$$= \mathbb{E}\|w_0 - w^*\|^2 - 2\gamma \mathbb{E}\langle \nabla f_{s_1}(w_0), w_0 - w^* \rangle + \gamma^2 \mathbb{E}\|\nabla f_{s_1}(w_0)\|^2$$

A single iteration of 1-sample SGD

Progress in a single iteration:

$$\mathbb{E}\|w_1 - w^*\|^2 = \mathbb{E}\|w_0 - \gamma \nabla f_{s_1}(w_0) - w^*\|^2$$
$$= \mathbb{E}\|w_0 - w^*\|^2 - 2\gamma \mathbb{E}\langle \nabla f_{s_1}(w_0), w_0 - w^* \rangle + \gamma^2 \mathbb{E}\|\nabla f_{s_1}(w_0)\|^2$$
$$= \mathbb{E}\|w_0 - w^*\|^2$$
$$- 2\gamma \mathbb{E}\langle \nabla f(w_0), w_0 - w^* \rangle + \gamma^2 \mathbb{E}\|\nabla f_{s_1}(w_0)\|^2$$

When mini-batch "works", you'd expect B times more progress!

Single iteration of B-sample SGD

Progress in a single mini-batch iteration:

$$\|w_B - w^*\| = \|w_0 - w^* - \gamma \sum_{i=1}^{B} \nabla f_{s_i}(w_0)\|^2$$

$$= \|w_0 - w^*\|^2$$

$$- 2\gamma \left\langle \sum_{i=1}^{B} \nabla f_{s_i}(w_0), w_0 - w^* \right\rangle + \gamma^2 \left\| \sum_{i=1}^{B} \nabla f_{s_i}(w_0) \right\|^2$$

How does it compare with 1-sample SGD?

Single iteration of B-sample SGD

"Progress" is equal to:

$$-2\gamma\mathbb{E}\left\langle \sum_{i=1}^{B} \nabla f_{s_i}(w_0), w_0 - w^* \right\rangle + \gamma^2 \mathbb{E}\left\| \sum_{i=1}^{B} \nabla f_{s_i}(w_0) \right\|^2$$

$$= -2B\gamma\mathbb{E}\left\langle \nabla f(w_0), w_0 - w^* \right\rangle + \gamma^2 \mathbb{E}\left\| \sum_{i=1}^{B} \nabla f_{s_i}(w_0) \right\|^2$$

The "variance" term is equal to:

$$\mathbb{E}\left\| \sum_{i=1}^{B} \nabla f_{s_i}(w_0) \right\|^2 = \mathbb{E}\left(\sum_{i=1}^{B} \|\nabla f_{s_i}(w_0)\|^2 + \sum_{i=1}^{B} \sum_{j=1, j\neq i}^{B} \langle \nabla f_{s_i}(w_0), \nabla f_{s_j}(w_0) \rangle \right)$$

Single iteration of B-sample SGD

"Progress" is equal to:

$$-2\gamma\mathbb{E}\left\langle \sum_{i=1}^{B}\nabla f_{s_i}(w_0), w_0 - w^* \right\rangle + \gamma^2\mathbb{E}\left\|\sum_{i=1}^{B}\nabla f_{s_i}(w_0)\right\|^2$$

$$= -2B\gamma\mathbb{E}\left\langle \nabla f(w_0), w_0 - w^* \right\rangle + \gamma^2\mathbb{E}\left\|\sum_{i=1}^{B}\nabla f_{s_i}(w_0)\right\|^2$$

The "variance" term is equal to:

$$\mathbb{E}\left\|\sum_{i=1}^{B}\nabla f_{s_i}(w_0)\right\|^2 = \mathbb{E}\left(\sum_{i=1}^{B}\|\nabla f_{s_i}(w_0)\|^2 + \sum_{i=1}^{B}\sum_{j=1,j\neq i}^{B}\langle \nabla f_{s_i}(w_0), \nabla f_{s_j}(w_0)\rangle\right)$$

$$= B\cdot\mathbb{E}\|\nabla f_{s_1}(w_0)\|^2 + \sum_{i=1}^{B}\sum_{j=1,j\neq i}^{B}\mathbb{E}\langle \nabla f_{s_i}(w_0), \nabla f_{s_j}(w_0)\rangle$$

Single iteration of B-sample SGD

"Progress" is equal to:

$$-2\gamma\mathbb{E}\left\langle \sum_{i=1}^{B} \nabla f_{s_i}(w_0), w_0 - w^* \right\rangle + \gamma^2 \mathbb{E}\left\| \sum_{i=1}^{B} \nabla f_{s_i}(w_0) \right\|^2$$

$$= -2B\gamma\mathbb{E}\left\langle \nabla f(w_0), w_0 - w^* \right\rangle + \gamma^2 \mathbb{E}\left\| \sum_{i=1}^{B} \nabla f_{s_i}(w_0) \right\|^2$$

The "variance" term is equal to:

$$\mathbb{E}\left\| \sum_{i=1}^{B} \nabla f_{s_i}(w_0) \right\|^2 = B \cdot \mathbb{E}\|\nabla f_{s_1}(w_0)\|^2 + \sum_{i=1}^{B} \sum_{j=1, j\neq i}^{B} \mathbb{E}\langle \nabla f_{s_i}(w_0), \nabla f_{s_j}(w_0) \rangle$$

$$= B \cdot \mathbb{E}\|\nabla f_{s_1}(w_0)\|^2 + B(B-1)\mathbb{E}\|\nabla f(w_0)\|^2$$

Mini-batch Progress

Progress in a single mini-batch SGD iteration:

$$-B \cdot \Bigg(2\mathbb{E}\gamma \left\langle \nabla f(w_0), w_0 - w^* \right\rangle - \gamma^2 \mathbb{E}\|\nabla f_{s_1}(w_0)\|^2 - (B-1)\gamma^2 \mathbb{E}\|\nabla f(w_0)\|^2 \Bigg)$$

Let's compare with SGD!

Single iteration of B-sample SGD

1-sample SGD Progress:

$$-2\gamma\mathbb{E}\left\langle \nabla f(w_0), w_0 - w^* \right\rangle + \gamma^2 \mathbb{E}\left\| \nabla f_{s_1}(w_0) \right\|^2$$

B-sample SGD Progress:

$$-B \cdot \Bigg(2\mathbb{E}\gamma \left\langle \nabla f(w_0), w_0 - w^* \right\rangle - \gamma^2 \mathbb{E}\| \nabla f_{s_1}(w_0) \|^2$$

$$- (B-1)\gamma^2 \mathbb{E}\| \nabla f(w_0) \|^2 \Bigg)$$

Extra term directly controlled by batchsize

Single iteration of B-sample SGD

B-sample SGD Progress:

$$-B \cdot \left(2\mathbb{E}\gamma \left\langle \nabla f(w_0), w_0 - w^* \right\rangle - \gamma^2 \mathbb{E}\|\nabla f_{s_1}(w_0)\|^2 \right.$$

$$\left. - (B-1)\gamma^2 \mathbb{E}\|\nabla f(w_0)\|^2 \right)$$

Simple idea: Make B so that the extra term is smaller than $\mathbb{E}\|\nabla f_{s_1}(w_0)\|^2$

Set $B = \delta \dfrac{\mathbb{E}\|\nabla f_{s_1}(w_0)\|^2}{\mathbb{E}\|\nabla f(w_0)\|^2}$

Single iteration of B-sample SGD

If $B = \delta \dfrac{\mathbb{E}\|\nabla f_{s_1}(w_0)\|^2}{\mathbb{E}\|\nabla f(w_0)\|^2}$

We get

$$-B \cdot \left(2\mathbb{E}\gamma \left\langle \nabla f(w_0), w_0 - w^* \right\rangle - (1+\delta)\gamma^2 \mathbb{E}\|\nabla f_{s_1}(w_0)\|^2 \right)$$

1- sample SGD gets:

$$-2\gamma \mathbb{E} \left\langle \nabla f(w_0), w_0 - w^* \right\rangle + \gamma^2 \mathbb{E} \left\|\nabla f_{s_1}(w_0)\right\|^2$$

B times more progress! (approximately)

Gradient Diversity

Sum of grad norms:

$$M^2(w) = \frac{1}{n} \sum_{i=1}^{n} \|\nabla f_i(w)\|^2$$

Gradient Diversity:

$$\Delta(w) = \frac{\sum_{i=1}^{n} \|\nabla f_i(w)\|^2}{\|\sum_{i=1}^{n} \nabla f_i(w)\|^2}$$

$$= \frac{\sum_{i=1}^{n} \|\nabla f_i(w)\|^2}{\sum_{i=1}^{n} \|\nabla f_i(w)\|^2 + \sum_{i \neq j} \langle \nabla f_i(w), \nabla f_j(w) \rangle}$$

Main Result

Main Theorem (assumption-less!)

Let $w_{k\cdot B}$ be a fixed model, and let $w_{(k+1)\cdot B}$ denote the model after a mini-batch iteration with batch-size $B = \delta n \cdot \Delta(w) + 1$. Then, we have

$$\mathbb{E}\{\, \|w_{(k+1)\cdot B} - w^*\|^2 \;\mid\; w_{kB}\} \leq \mathbb{E}\|w_{k\cdot B} - w^*\|^2 - B\Big(2\gamma\langle \nabla f(w_{k\cdot B}), w_{k\cdot B} - w^*\rangle - (1+\delta)\gamma^2 M^2(w_{k\cdot B})\Big)$$

Remarks:

- This is a local lemma
- True for both global, local min, and critical points
- It's universal! (no assumptions, always true)

Gradient Diversity

Gradient Diversity:

$$\Delta(w) = \frac{\sum_{i=1}^{n} \|\nabla f_i(w)\|^2}{\|\sum_{i=1}^{n} \nabla f_i(w)\|^2}$$

$$= \frac{\sum_{i=1}^{n} \|\nabla f_i(w)\|^2}{\sum_{i=1}^{n} \|\nabla f_i(w)\|^2 + \sum_{i \neq j} \langle \nabla f_i(w), \nabla f_j(w) \rangle}$$

Measures similarity between gradients

- Big Diversity: Larger batches => better speedups
- Small Diversity: Smaller Batches => worse speedup

Examples:

1. All gradients are orthogonal, Diversity = 1
2. All gradients identical, Diversity = 1/n

Main Result

Corollary: If $B = \delta n \cdot \Delta(w) + 1$

Function class	serial SGD step-size $\gamma(\epsilon)$	mini-batch SGD step-size $\gamma(\epsilon)/(1+\delta)$
λ-strongly convex	$\frac{M^2 \log(2D_0/\epsilon)}{2\lambda^2\epsilon}$	$(1+\delta)\frac{M^2 \log(2D_0/\epsilon)}{2\lambda^2\epsilon}$
convex	$\frac{M^2 D_0}{\epsilon^2}$	$(1+\delta)\frac{M^2 D_0}{\epsilon^2}$
β-smooth	$\frac{2M^2\beta(F(\mathbf{w}_0)-F^*)}{\epsilon^2}$	$(1+\delta)\frac{2M^2\beta(F(\mathbf{w}_0)-F^*)}{\epsilon^2}$
β-smooth μ-PL	$\frac{M^2\beta \log(2(F(\mathbf{w}_0)-F^*)/\epsilon)}{4\mu^2\epsilon}$	$(1+\delta)\frac{M^2\beta \log(2(F(\mathbf{w}_0)-F^*)/\epsilon)}{4\mu^2\epsilon}$

Set batch-size ~= grad diversity and you're good!

Can do better with odd LR schedules

Accurate, Large Minibatch SGD: Training ImageNet in 1 Hour

Priya Goyal Piotr Dollár Ross Girshick Pieter Noordhuis
Lukasz Wesolowski Aapo Kyrola Andrew Tulloch Yangqing Jia Kaiming He

Facebook

Gradual warmup. We present an alternative warmup that *gradually* ramps up the learning rate from a small to a large value. This ramp avoids a sudden increase of the learning rate, allowing healthy convergence at the start of training. In practice, with a large minibatch of size kn, we start from a learning rate of η and increment it by a constant amount at each iteration such that it reaches $\hat{\eta} = k\eta$ after 5 epochs (results are robust to the exact duration of warmup). After the warmup, we go back to the original learning rate schedule.

ImageNet Training in Minutes

Yang You
UC Berkeley
youyang@cs.berkeley.edu

Zhao Zhang
TACC
zzhang@tacc.utexas.edu

Cho-Jui Hsieh
UC Davis
chohsieh@ucdavis.edu

James Demmel
UC Berkeley
demmel@cs.berkeley.edu

Kurt Keutzer
UC Berkeley
keutzer@cs.berkeley.edu

3.4 Scaling up Batch Size

To improve the accuracy for large batch training, a new rule of learning rate (LR) schedule was developed. As discussed in §2.1, we use $w = w - \eta \nabla w$ to update the weights. Each layer has its own weight w and gradient ∇w. Standard SGD algorithm uses the same LR (η) for all the layers. However, from our experiments, we observe that different layers may need different LRs. The reason is that the ratio between $||w||_2$ and $||\nabla w||_2$ varies significantly for different layers. From example, we observe that $||w||_2/||\nabla w||_2$ is only 20 for conv1.1 layer (Table 6). However, $||w||_2/||\nabla w||_2$ is 3,690 for fc6.1 layer. To speedup the convergence for fc6.1 layer, the users need to use a large LR. However, this large LR may lead to divergence on the conv1.1 layer. We believe this is an important reason of the optimization difficulty in large batch training.

Can do better with odd LR schedules

Accurate, Large Minibatch SGD: Training ImageNet in 1 Hour

Priya Goyal Piotr Dollár Ross Girshick Pieter Noordhuis
Lukasz Wesolowski Aapo Kyrola Andrew Tulloch Yangqing Jia Kaiming He

Facebook

Gradual warmup. We present an alternative warmup that *gradually* ramps up the learning rate from a small to a large value. This ramp avoids a sudden increase of the learning rate, allowing healthy convergence at the start of training. In practice, with a large minibatch of size kn, we start from a learning rate of η and increment it by a constant amount at each iteration such that it reaches $\hat{\eta} = k\eta$ after 5 epochs (results are robust to the exact duration of warmup). After the warmup, we go back to the original learning rate schedule.

ImageNet Training in Minutes

Yang You
UC Berkeley
youyang@cs.berkeley.edu

Zhao Zhang
TACC
zzhang@tacc.utexas.edu

Cho-Jui Hsieh
UC Davis
chohsieh@ucdavis.edu

James Demmel
UC Berkeley
demmel@cs.berkeley.edu

Kurt Keutzer
UC Berkeley
keutzer@cs.berkeley.edu

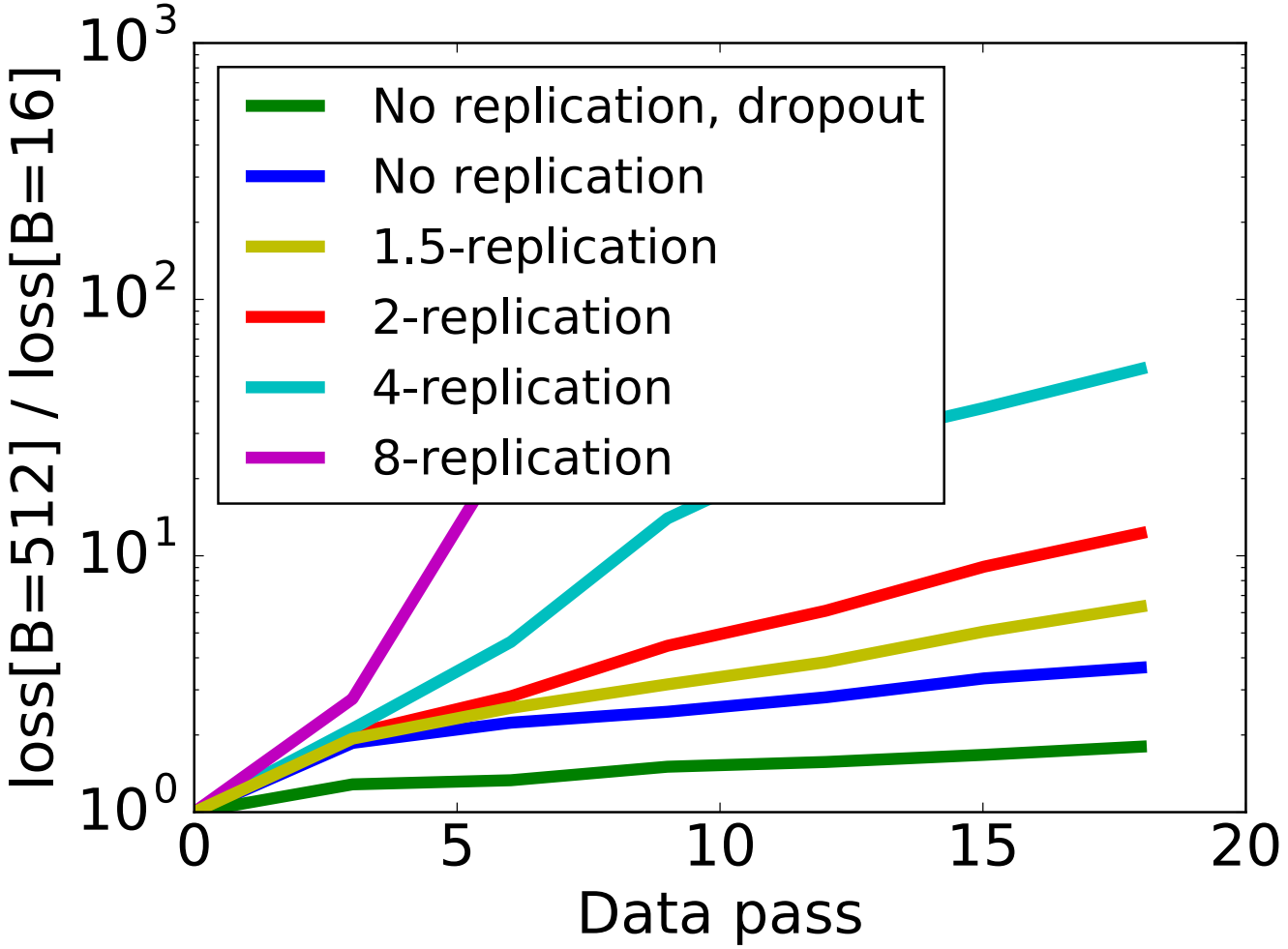
3.4 Scaling up Batch Size

To improve the accuracy for large batch training, a new rule of learning rate (LR) schedule was developed. As discussed in §2.1, we use $w = w - \eta\nabla w$ to update the weights. Each layer has its own weight w and gradient ∇w. Standard SGD algorithm uses the same LR (η) for all the layers. However, from our experiments, we observe that different layers may need different LRs. The reason is that the ratio between $||w||_2$ and $||\nabla w||_2$ varies significantly for different layers. From example, we observe that $||w||_2/||\nabla w||_2$ is only 20 for conv1.1 layer (Table 6). However, $||w||_2/||\nabla w||_2$ is 3,690 for fc6.1 layer. To speedup the convergence for fc6.1 layer, the users need to use a large LR. However, this large LR may lead to divergence on the conv1.1 layer. We believe this is an important reason of the optimization difficulty in large batch training.

Gradient Diversity in Practice

Gradient Diversity in Experiments

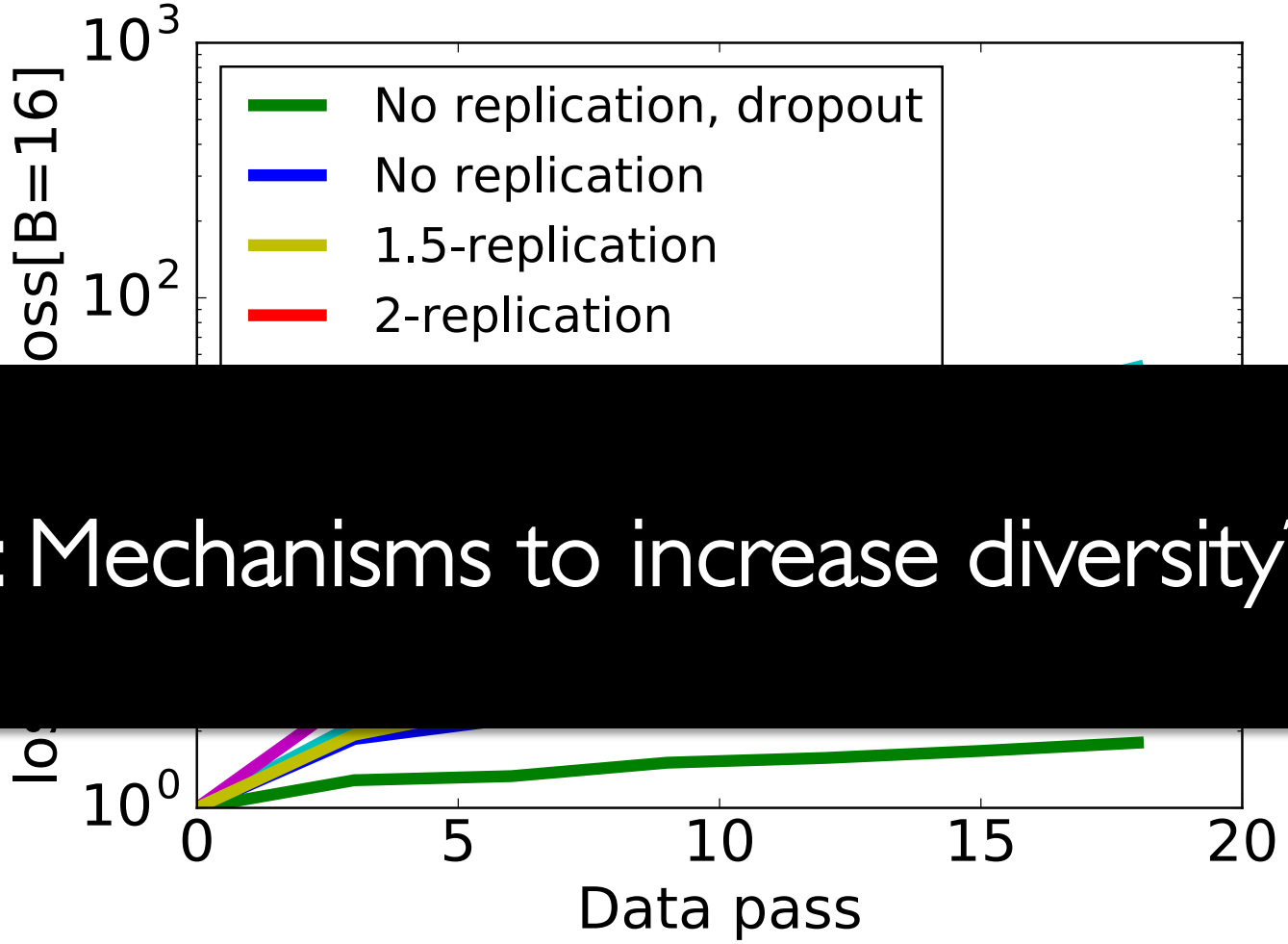
- CIFAR-10 (cuda conv-net)



Smaller diversity => slower convergence

Gradient Diversity in Experiments

- CIFAR-10 (cuda conv-net)



Q: Mechanisms to increase diversity?

Smaller diversity => slower convergence

Many more Questions….

- Generalization?
- What happens with delayed nodes?
- Comm. is expensive, how often do we average?

Generalization?

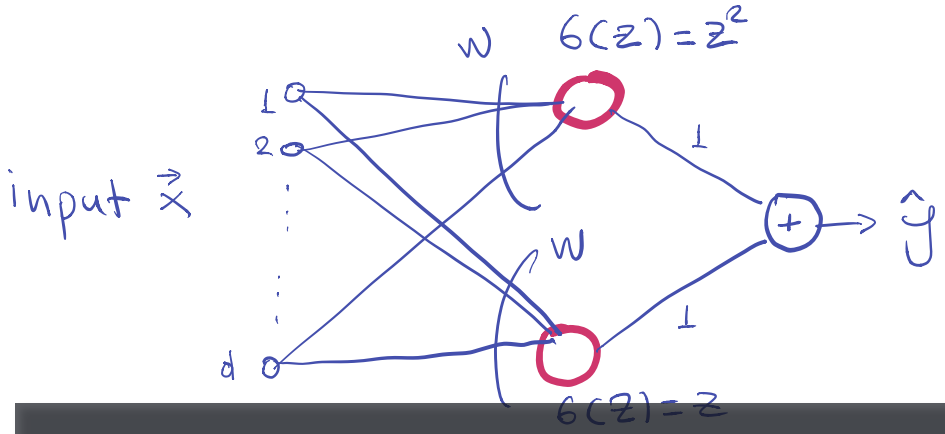
Mini vs large batch phenomena not well understood

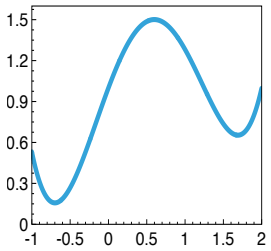
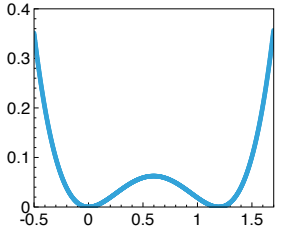
Batch	Top-1 Acc	Top-5 Acc
256	58.42%	81.51%
512	59.19%	81.84%
1024	59.00%	81.94%
2048	58.88%	81.73%
4096	57.97%	81.00%
8192	55.90%	79.40%

Alexnet on Imagenet

Generalization?

Theorem (informal):
There exist neural nets where SGD can be stable but GD is not



Loss = n  + 

Mini vs large batch phenomena not well understood

Next Time

- Asynchronous Optimization
- Stragglers
- Hogwild

Reading list

Goyal, P., Dollár, P., Girshick, R., Noordhuis, P., Wesolowski, L., Kyrola, A., Tulloch, A., Jia, Y. and He, K., 2017. Accurate, large minibatch sgd: Training imagenet in 1 hour. arXiv preprint arXiv:1706.02677.

Vancouver

https://arxiv.org/pdf/1706.02677.pdf

Hoffer, E., Hubara, I. and Soudry, D., 2017. Train longer, generalize better: closing the generalization gap in large batch training of neural networks. Advances in neural information processing systems, 30.

https://arxiv.org/pdf/1705.08741.pdf

You, Y., Zhang, Z., Hsieh, C.J., Demmel, J. and Keutzer, K., 2018, August. Imagenet training in minutes. In Proceedings of the 47th International Conference on Parallel Processing (pp. 1-10).

https://www2.eecs.berkeley.edu/Pubs/TechRpts/2020/EECS-2020-18.pdf

Keskar, N.S., Mudigere, D., Nocedal, J., Smelyanskiy, M. and Tang, P.T.P., 2016. On large-batch training for deep learning: Generalization gap and sharp minima. ICLR 2017

https://arxiv.org/pdf/1609.04836.pdf,

Yin, D., Pananjady, A., Lam, M., Papailiopoulos, D., Ramchandran, K. and Bartlett, P., 2018, March. Gradient diversity: a key ingredient for scalable distributed learning. In International Conference on Artificial Intelligence and Statistics (pp. 1998-2007). PMLR.

Vancouver

http://proceedings.mlr.press/v84/yin18a/yin18a.pdf