MIT HAN LAB

Lecture 14
Gradient Compression and Delayed Gradient Averaging

Song Han
songhan@mit.edu

Lecture Plan

Today we will:

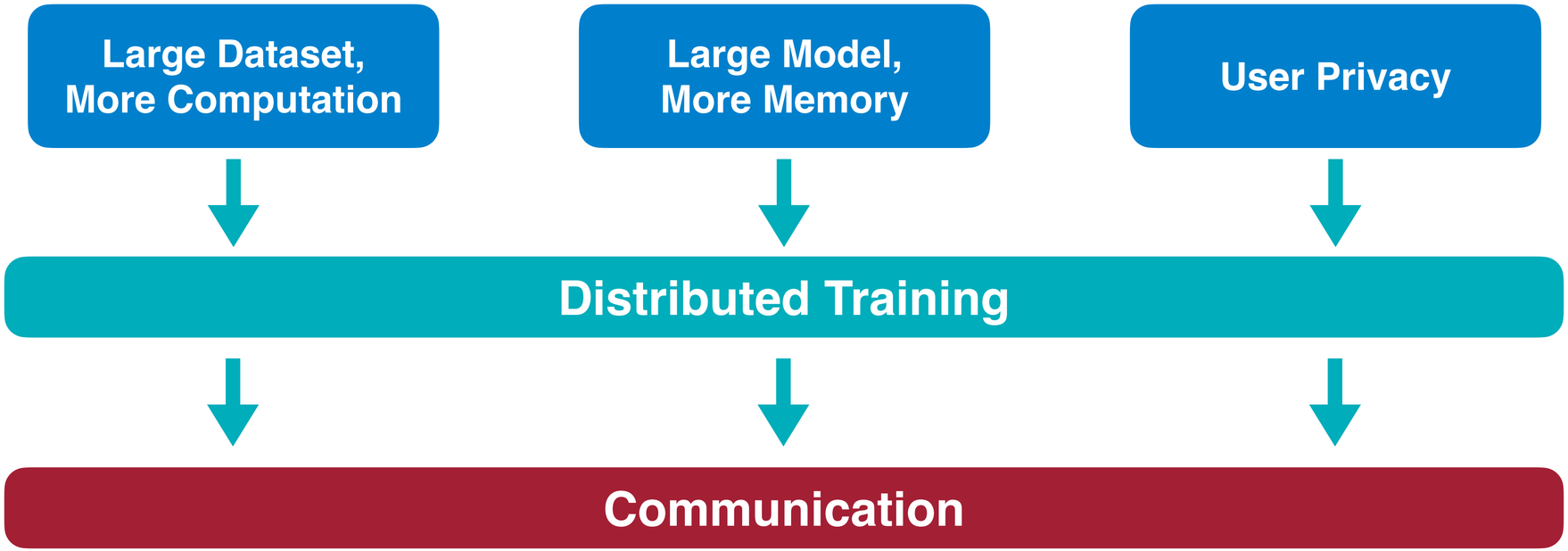
1. Understand the bandwidth and latency bottleneck of distributed training
2. Overcome the bandwidth bottleneck by gradient compression
3. Overcome the latency bottleneck by delayed gradient update

Section 1: Bottlenecks in Distributed Training

What makes scaling up training difficult?

Problems of Distributed Training

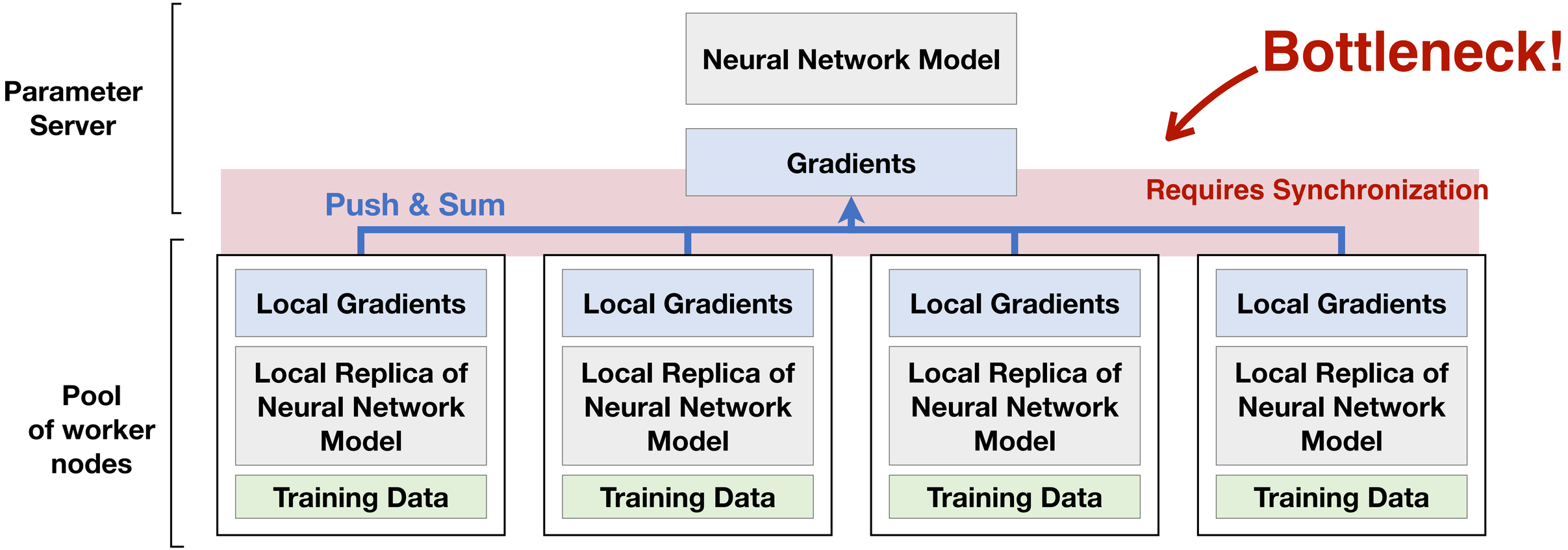
Communication in Distributed Training



Problems of Distributed Training

Communication

- **Requires Synchronization, High Communication Frequency**



Problems of Distributed Training

Communication

- Requires Synchronization, High Communication Frequency
- **Larger Model, Larger Transfer Data Size, Longer Transfer Time**

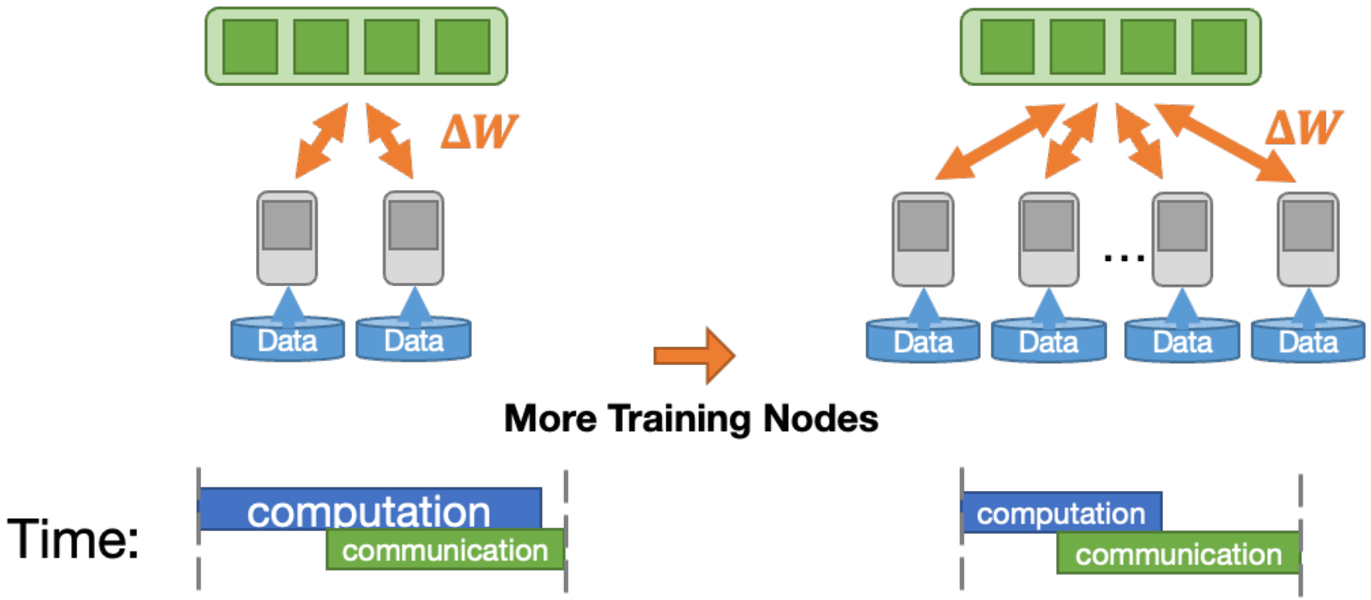
	# Param (M)	Iteration Time (2080Ti, bs=32, ms)	Bandwidth (Gb/s)
ResNet18	11.69	137	**2.7**
ResNet50	24.46	330	**2.3**
Swin-Tiny	28.10	590	**1.5**
Swin-Large	98.20	940	**3.3**

Normal Ethernet Bandwidth: **1Gb/s**

Problems of Distributed Training

Communication

- Requires Synchronization, High Communication Frequency
- Larger Model, Larger Transfer Data Size, Longer Transfer Time
- **More Training Nodes, More Communication Steps, Longer Latency**



Bandwidth-efficient deep learning [Han et al, 2018]

Distributed Training Scalability
under 20 Gbps Ethernet with GPUDirect™

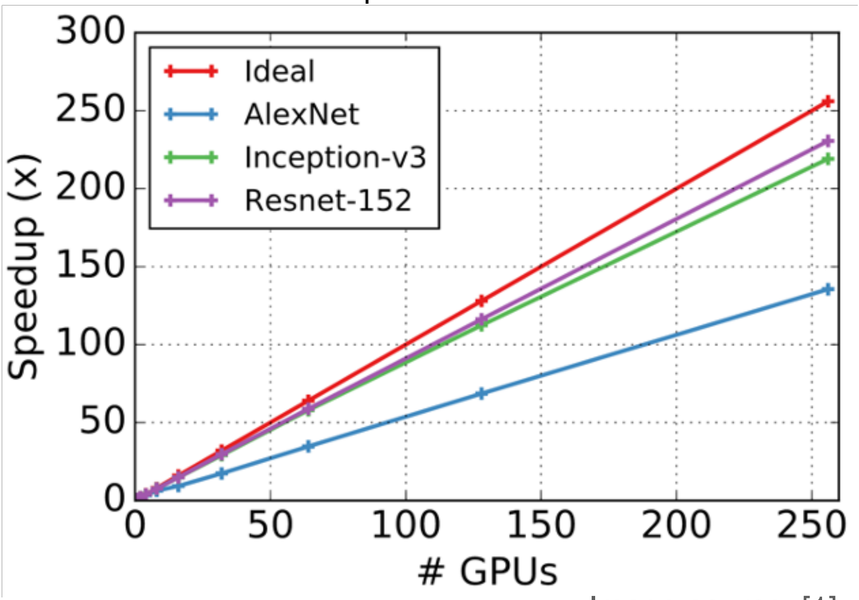
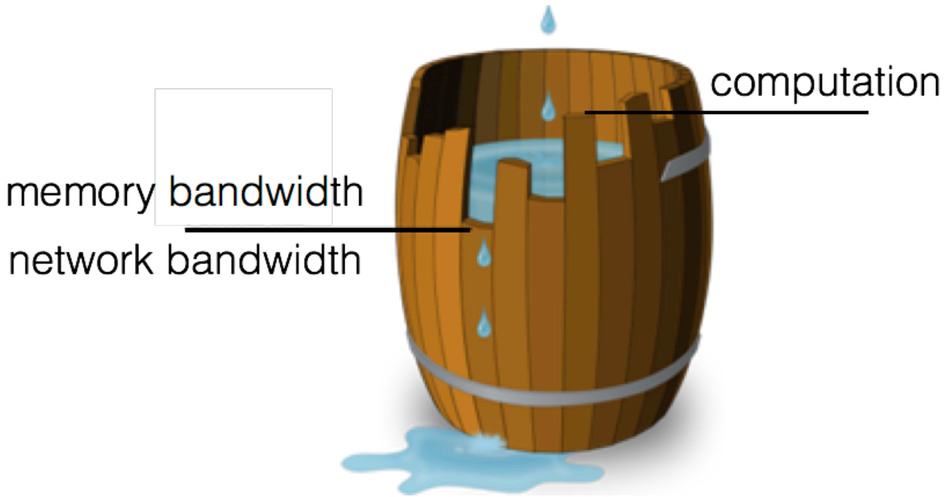


Image source: [1]

Problems of Distributed Training

Communication

- Requires Synchronization, High Communication Frequency
- Larger Model, Larger Transfer Data Size, Longer Transfer Time
- More Training Nodes, More Communication Steps, Longer Latency
- **Poor Network Bandwidth & Intermittent Connection for Cellular Network**



Distributed Training Scalability
under 20 Gbps Ethernet with GPUDirect™

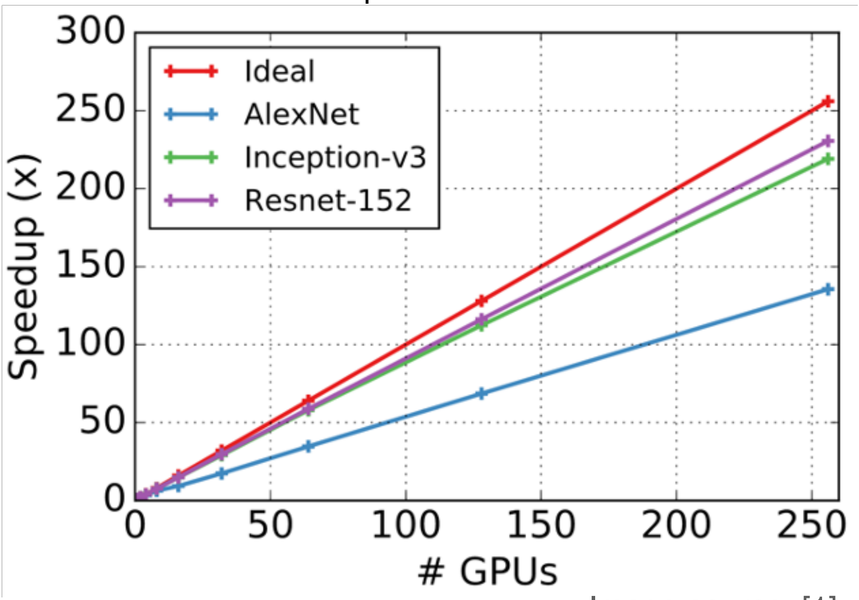


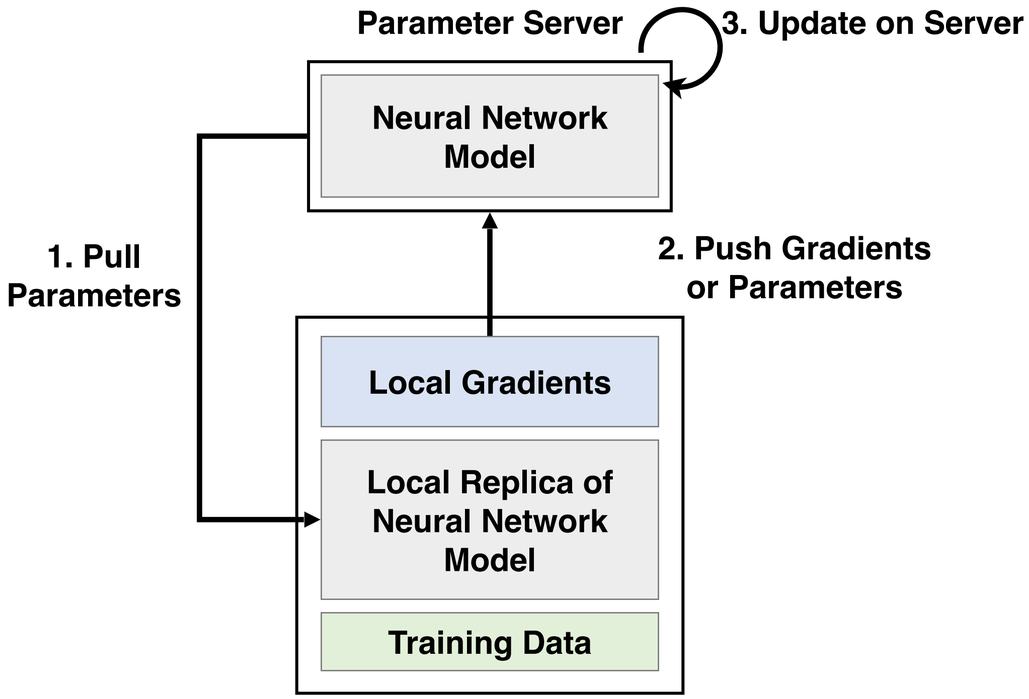
Image source: [1]

Section 2: Gradient Compression

Reduce the communication bandwidth in distributed training.

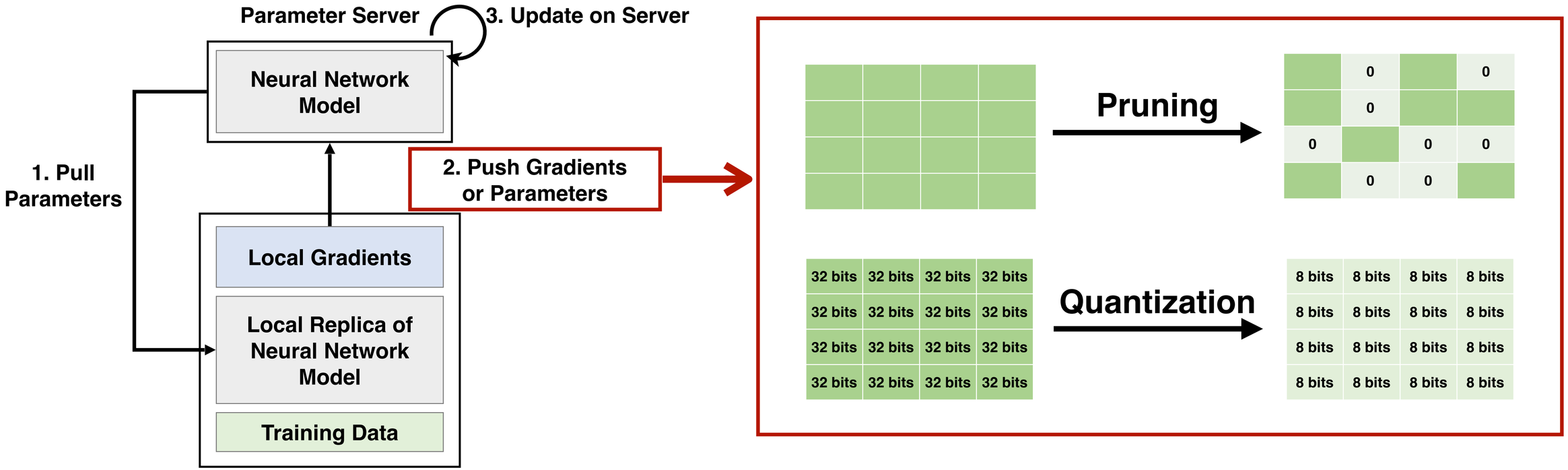
Reduce Transfer Data Size

Recall the workflow of Parameter-Server Based Distributed Training



Reduce Transfer Data Size

Recall the workflow of Parameter-Server Based Distributed Training



Gradient Compression

- Gradient Compression: Reduce the Gradient Size
  - Gradient Pruning
    - Sparse Communication
    - Deep Gradient Compression
    - PowerSGD
  - Gradient Quantization
    - 1-Bit SGD
    - TernGrad

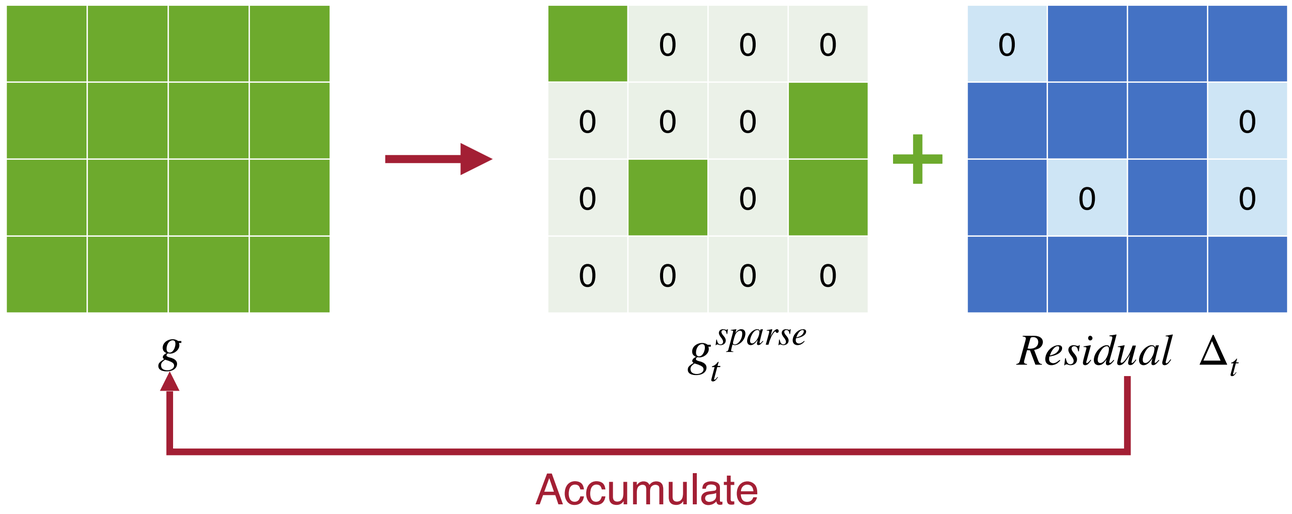
Gradient Compression

- Gradient Compression: Reduce the Gradient Size
  - Gradient Pruning
    - Sparse Communication
    - Deep Gradient Compression
    - PowerSGD
  - Gradient Quantization
    - 1-Bit SGD
    - TernGrad

Sparse Communication

With local gradient accumulation

- Send gradients **with top-k magnitude**
- Keep the un-pruned part as **error feedback (residual)**



Sparse communication for distributed gradient descent [Alham Fikri et al 2017]
Deep Gradient Compression: Reducing the Communication Bandwidth for Distributed Training [Lin et al 2017]

Sparse Communication

With local gradient accumulation

- Send gradients **with top-k magnitude**
- Keep the un-pruned part as **error feedback (residual)**
- Improve training speed

Drop Ratio	words/sec (NMT)	images/sec (MNIST)
0%	13100	2489
90%	14443	3174
99%	14740	3726
99.9%	14786	3921

Sparse communication for distributed gradient descent [Alham Fikri et al 2017]

Sparse Communication

With local gradient accumulation

- Send gradients **with top-k magnitude**
- Keep the un-pruned part as **error feedback (residual)**
- Improve training speed
- Work for simple neural networks, but fail on modern models like ResNet

Setting	Top 1 Accuracy of ResNet-110 on Cifar10		
GPUs=8 bs=128	Baseline	93.75%	0
	Sparse Communication	**92.75%**	**-1.00%**

Sparse communication for distributed gradient descent [Alham Fikri et al 2017]
Deep Gradient Compression: Reducing the Communication Bandwidth for Distributed Training [Lin et al 2017]

Limitations of Sparse Communication

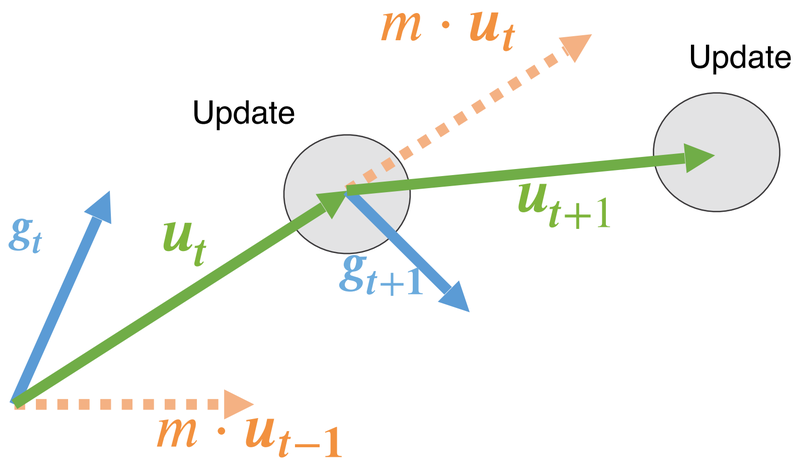
What leads to poor performance in gradient dropping?

Momentum!

Optimizers with Momentum

- Momentum is an extension to the gradient descent, useful to overcome local minima and oscillation of noisy gradients.

- Momentum is widely used to train modern deep neural networks (used in AlexNet, VGG, ResNet, Transformer, BERT …)



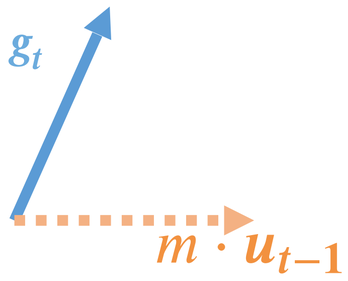
Vanilla Momentum SGD

$$u_t = m \cdot u_{t-1} + g_t$$
$$w_t = w_{t-1} - \eta \cdot u_t$$

On the momentum term in gradient descent learning algorithms. [Qian 1999]
Deep Gradient Compression: Reducing the Communication Bandwidth for Distributed Training [Lin et al 2017]

Optimizers with Momentum

Step 1 - Obtain the gradients g_t



Vanilla Momentum SGD

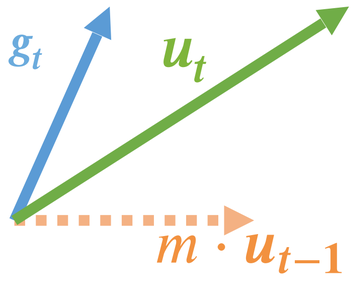
$$u_t = m \cdot u_{t-1} + g_t$$
$$w_t = w_{t-1} - \eta \cdot u_t$$

Calculate g_t same as the vanilla SGD.

Deep Gradient Compression: Reducing the Communication Bandwidth for Distributed Training [Lin et al 2017]

Optimizers with Momentum

Step 2 - Calculate the velocity u_t using momentum m



Vanilla Momentum SGD

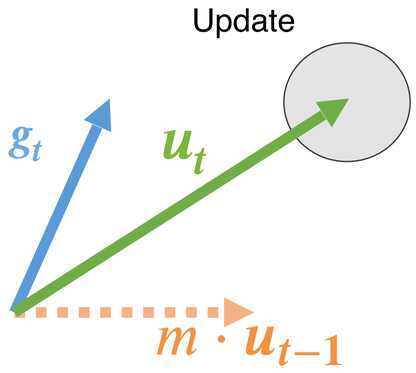
$$u_t = m \cdot u_{t-1} + g_t$$
$$w_t = w_{t-1} - \eta \cdot u_t$$

Momentum m is usually a value between 0 and 1.

Deep Gradient Compression: Reducing the Communication Bandwidth for Distributed Training [Lin et al 2017]

Optimizers with Momentum

Step 3 : Update model weights using velocity u_t



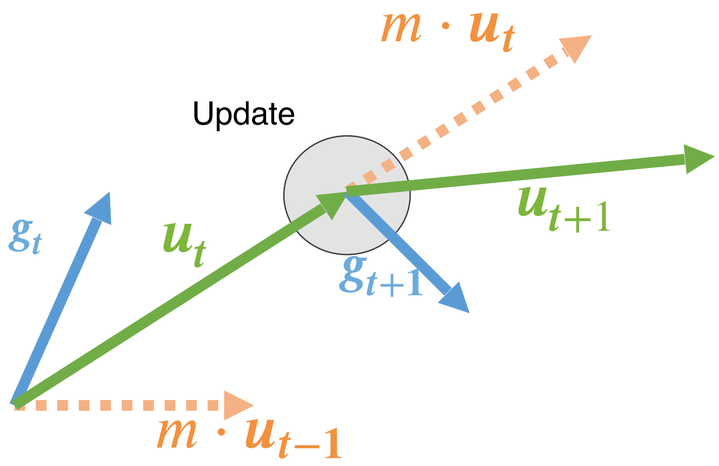
Vanilla Momentum SGD

$$u_t = m \cdot u_{t-1} + g_t$$
$$w_t = w_{t-1} - \eta \cdot u_t$$

Deep Gradient Compression: Reducing the Communication Bandwidth for Distributed Training [Lin et al 2017]

Optimizers with Momentum

Repeat, calculate the velocity for the next iteration



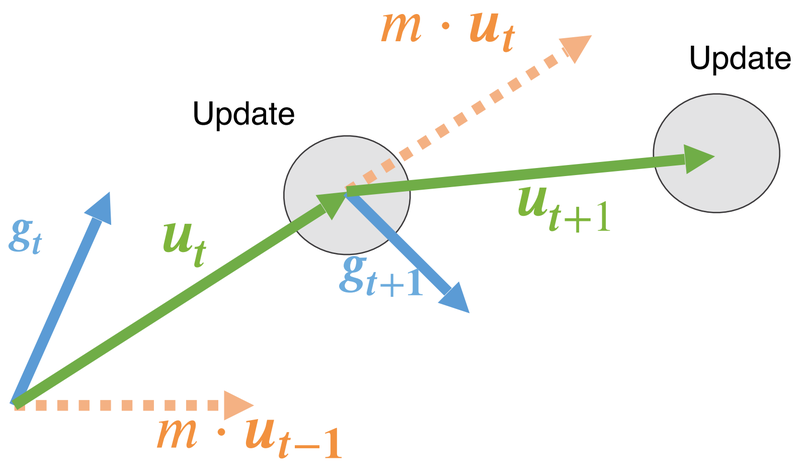
Vanilla Momentum SGD

$$u_t = m \cdot u_{t-1} + g_t$$
$$w_t = w_{t-1} - \eta \cdot u_t$$

Deep Gradient Compression: Reducing the Communication Bandwidth for Distributed Training [Lin et al 2017]

Optimizers with Momentum

Repeat, update weights



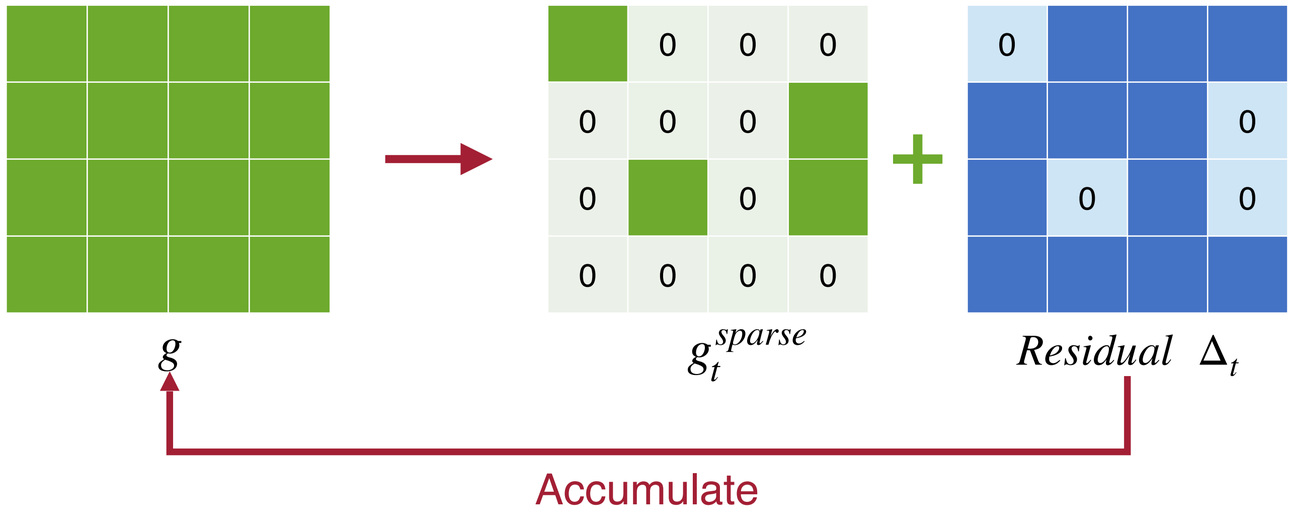
Vanilla Momentum SGD

$$u_t = m \cdot u_{t-1} + g_t$$
$$w_t = w_{t-1} - \eta \cdot u_t$$

Deep Gradient Compression: Reducing the Communication Bandwidth for Distributed Training [Lin et al 2017]

Momentum Needs Correction

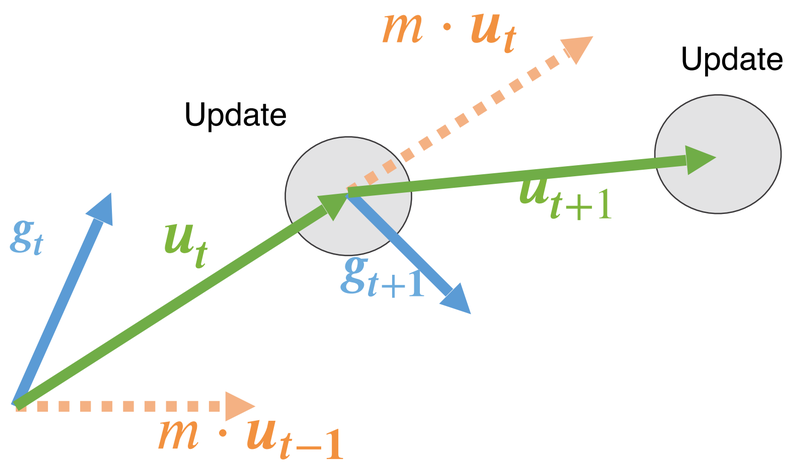
- In sparse communication, the pruned gradients are accumulated in local buffers.
- What will happen if we directly use the accumulated buffer for momentum calculation?



Deep Gradient Compression: Reducing the Communication Bandwidth for Distributed Training [Lin et al 2017]

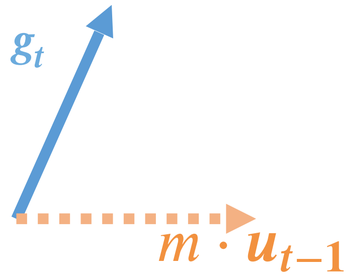
Deep Gradient Compression

Momentum Correction



Vanilla Momentum SGD

$$u_t = m \cdot u_{t-1} + g_t$$
$$w_t = w_{t-1} - \eta \cdot u_t$$



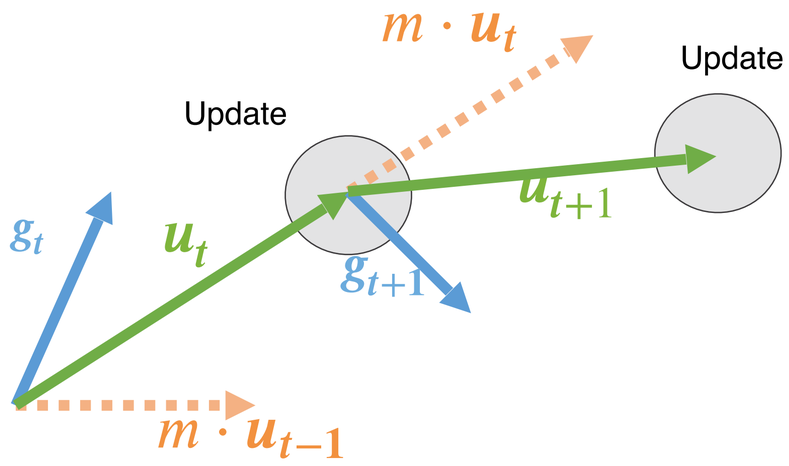
Momentum SGD with Local Accumulation

Starting from gradients and momentum

Deep Gradient Compression: Reducing the Communication Bandwidth for Distributed Training [Lin et al 2017]

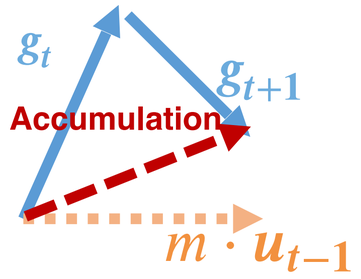
Deep Gradient Compression

Momentum Correction



Vanilla Momentum SGD

$$u_t = m \cdot u_{t-1} + g_t$$
$$w_t = w_{t-1} - \eta \cdot u_t$$



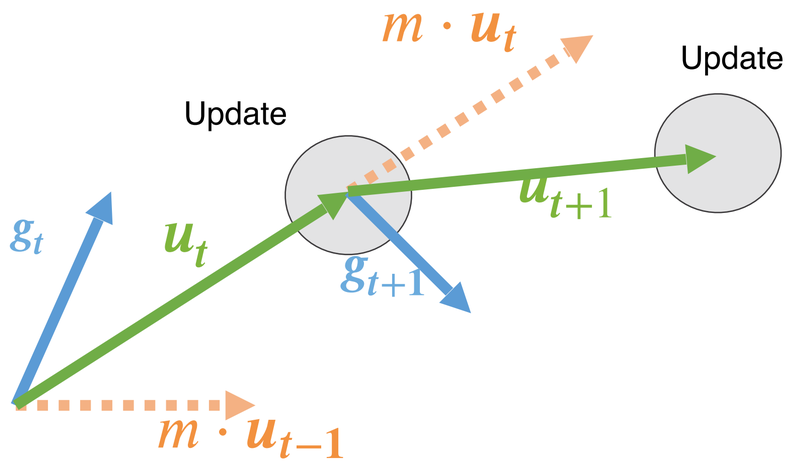
Momentum SGD with Local Accumulation

Accumulate the gradients in the local buffer

Deep Gradient Compression: Reducing the Communication Bandwidth for Distributed Training [Lin et al 2017]

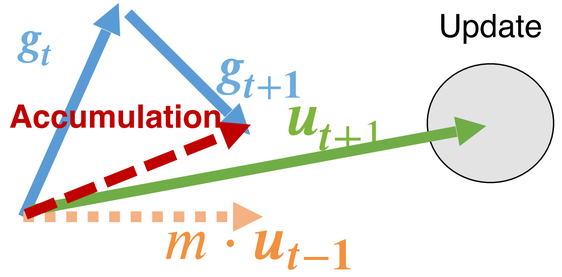
Deep Gradient Compression

Momentum Correction



Vanilla Momentum SGD

$$u_t = m \cdot u_{t-1} + g_t$$
$$w_t = w_{t-1} - \eta \cdot u_t$$



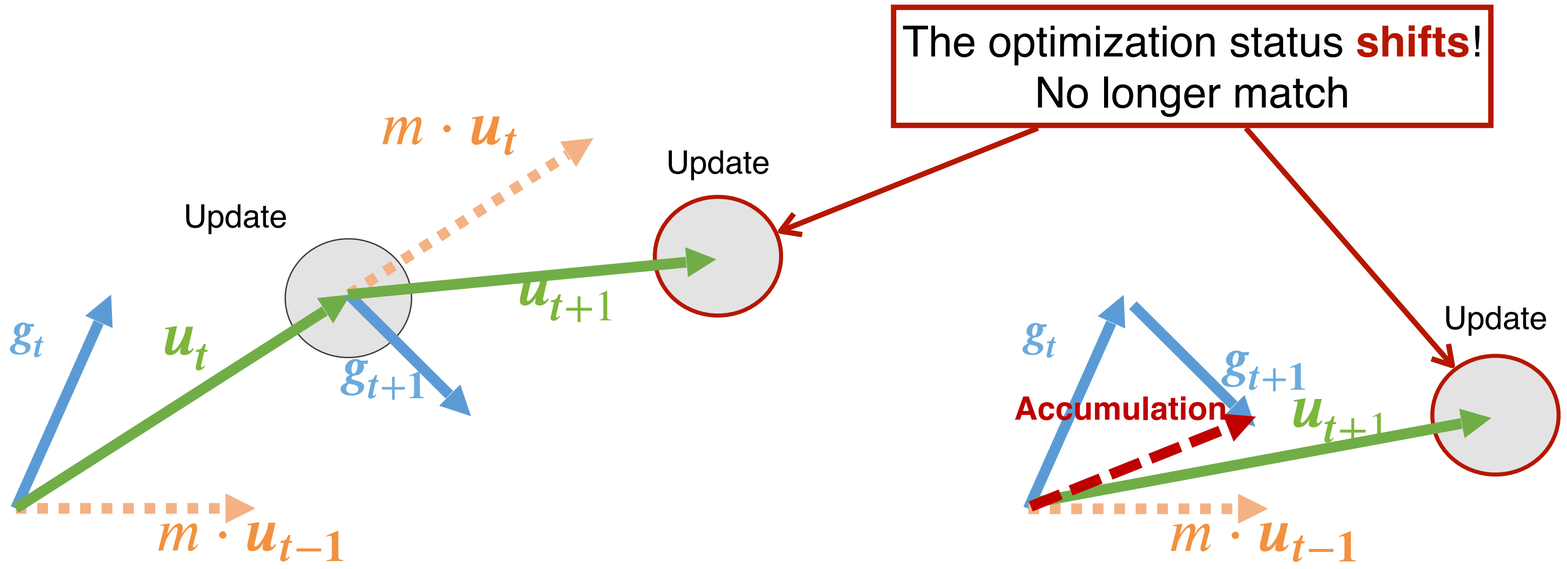
Momentum SGD with Local Accumulation

Compute the velocity using gradient accumulation and momentum

Deep Gradient Compression: Reducing the Communication Bandwidth for Distributed Training [Lin et al 2017]

Deep Gradient Compression

Momentum Correction



Vanilla Momentum SGD

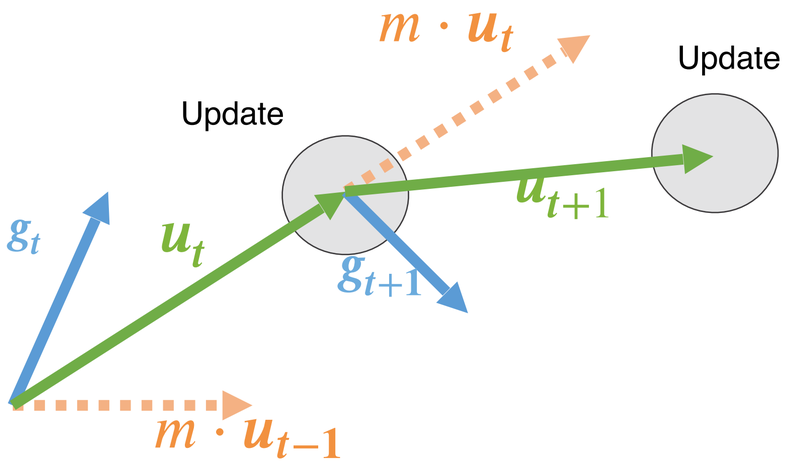
$$u_t = m \cdot u_{t-1} + g_t$$
$$w_t = w_{t-1} - \eta \cdot u_t$$

Momentum SGD with Local Accumulation

Deep Gradient Compression: Reducing the Communication Bandwidth for Distributed Training [Lin et al 2017]

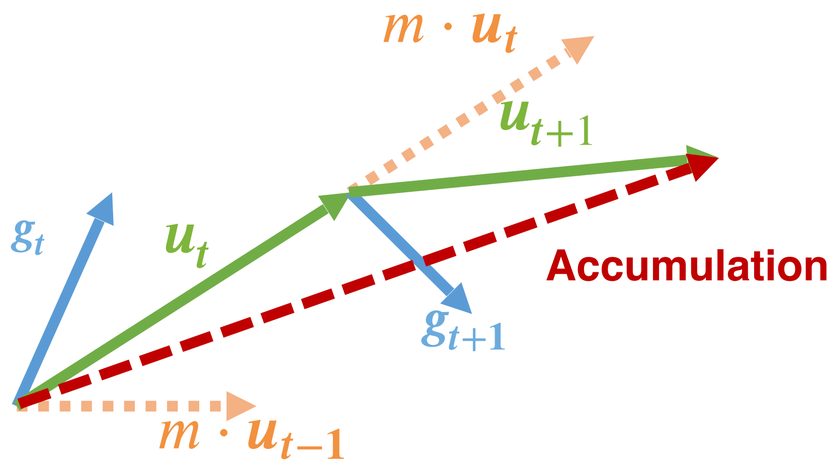
Deep Gradient Compression

Momentum Correction



Vanilla Momentum SGD

$$u_t = m \cdot u_{t-1} + g_t$$
$$w_t = w_{t-1} - \eta \cdot u_t$$



Momentum Correction

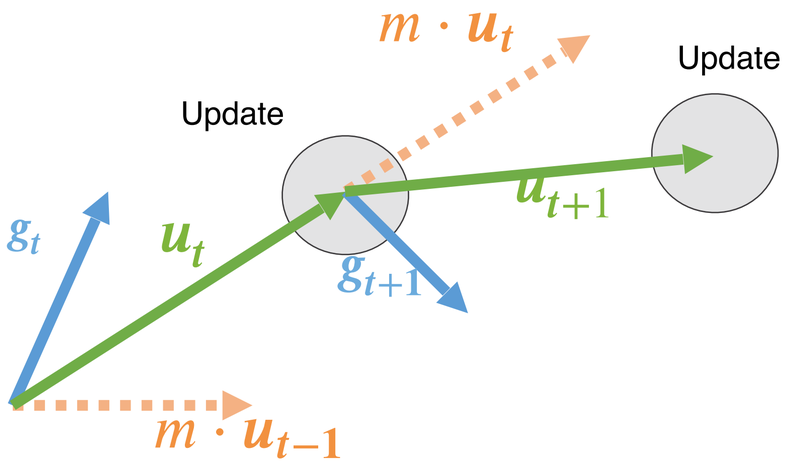
Accumulate the velocity, not gradients

accumulate the velocity through iterations

Deep Gradient Compression: Reducing the Communication Bandwidth for Distributed Training [Lin et al 2017]

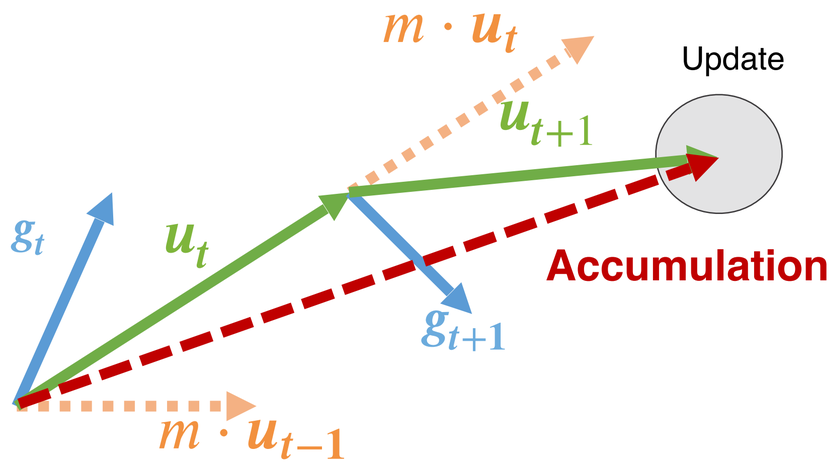
Deep Gradient Compression

Momentum Correction



Vanilla Momentum SGD

$$u_t = m \cdot u_{t-1} + g_t$$
$$w_t = w_{t-1} - \eta \cdot u_t$$



Momentum Correction

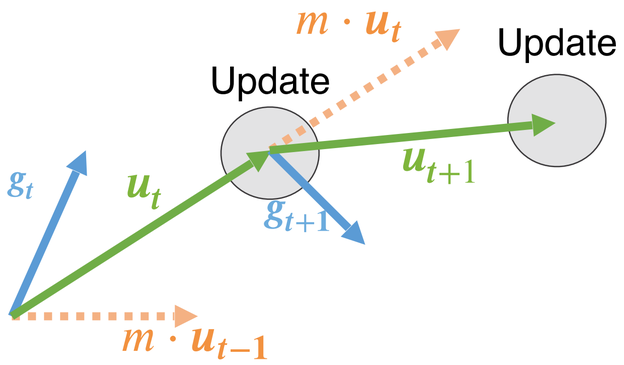
Accumulate the velocity, not gradients

With accumulated velocity, the optimization status now matches vanilla momentum SGD.

Deep Gradient Compression: Reducing the Communication Bandwidth for Distributed Training [Lin et al 2017]

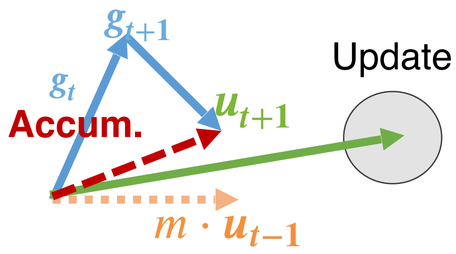
Deep Gradient Compression

Gradient Compression with momentum correction

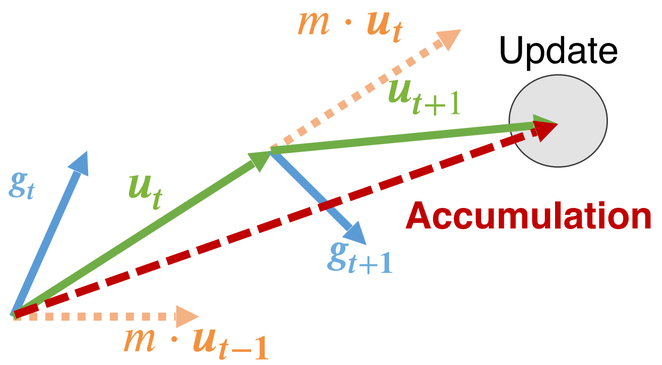


Vanilla Momentum SGD

$$u_t = m \cdot u_{t-1} + g_t$$
$$w_t = w_{t-1} - \eta \cdot u_t$$



Without Momentum Correction
(Wrong Direction)



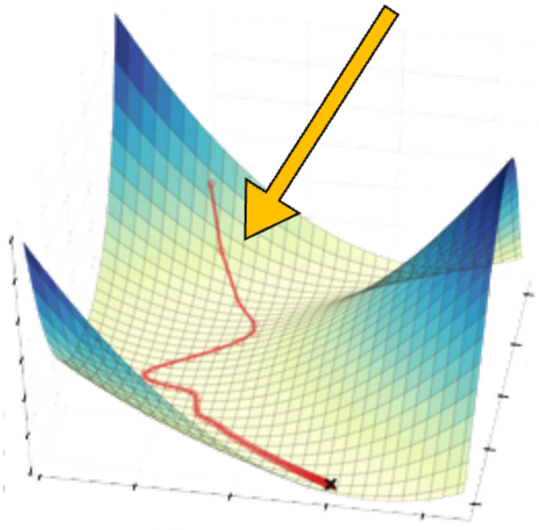
With Momentum Correction
(Correct Direction)

Deep Gradient Compression: Reducing the Communication Bandwidth for Distributed Training [Lin et al 2017]

Deep Gradient Compression

Method 3: Warm Up Training

- In the **early stages** of training, the network is **changing rapidly**
- Local gradient accumulation and stale gradient will **aggravate** the problem

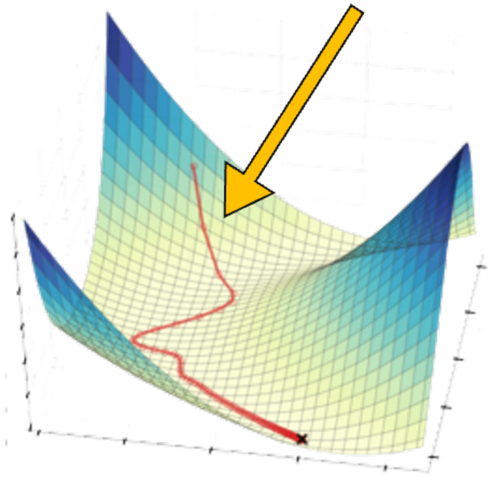


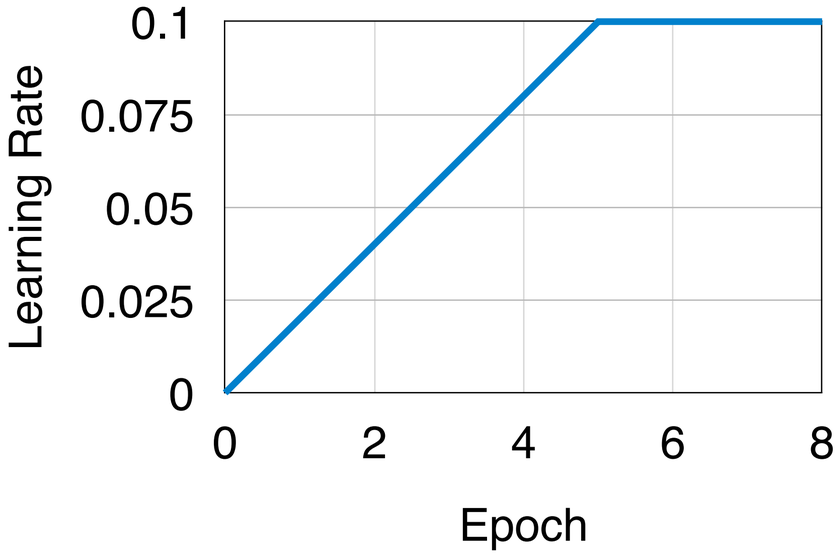
Deep Gradient Compression: Reducing the Communication Bandwidth for Distributed Training [Lin et al 2017]

Deep Gradient Compression

Warm Up Tricks

- **Warm up the learning rate**
- Warm up sparsity
  - **avoid** a sudden change in sparsity
  - **exponentially** increasing sparsity in first several epochs → help optimizer adapt to larger sparsity



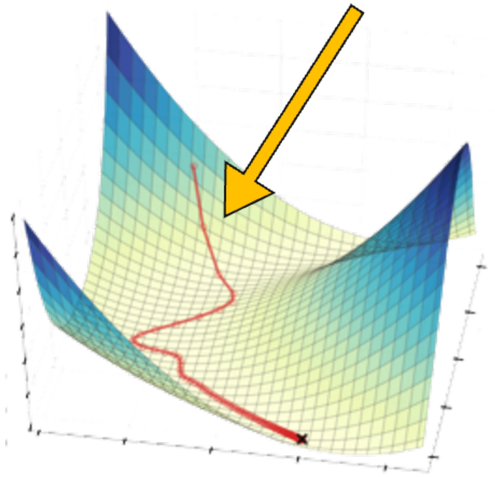


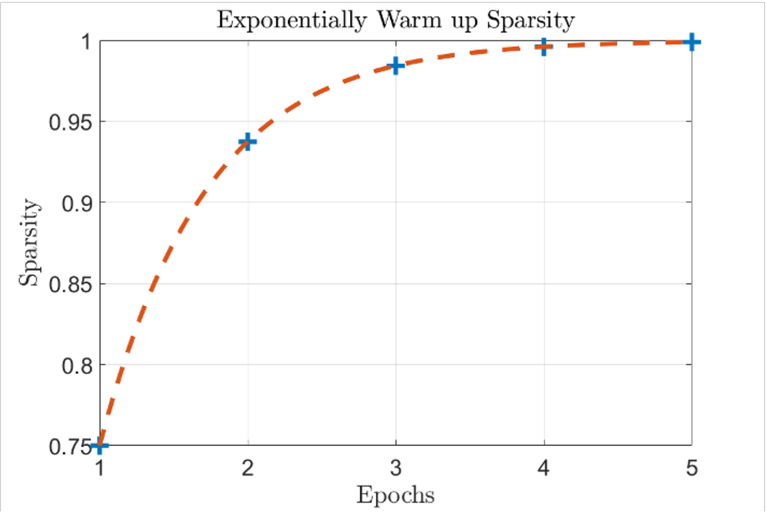
Deep Gradient Compression: Reducing the Communication Bandwidth for Distributed Training [Lin et al 2017]

Deep Gradient Compression

Warm Up Tricks

- Warm up learning rate
- **Warm up sparsity**
  - **avoid** a sudden change in sparsity
  - **exponentially** increasing sparsity in first several epochs → help the optimizer adapt to larger sparsity





Deep Gradient Compression: Reducing the Communication Bandwidth for Distributed Training [Lin et al 2017]

Deep Gradient Compression

DGC with 99.9% sparsity

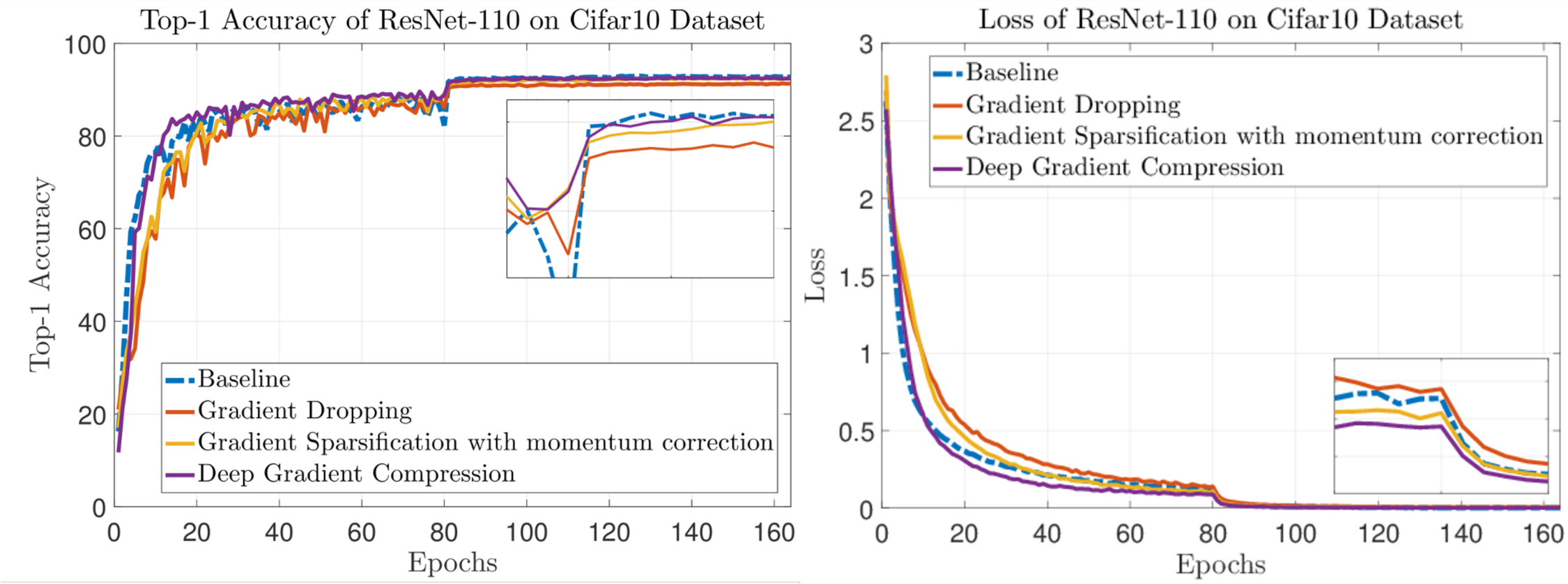
# GPUs in total	Top 1 Accuracy of ResNet-110 on Cifar10		
8	Baseline	92.92	0
	Naive Gradient Pruning	Not Converge	
	+ Local Gradient Accumulation	91.36	-1.56
	+ Local Gradient Accumulation + Momentum Correction	92.56	-0.36
	+ Local Gradient Accumulation + Warm Up Training	91.89	-1.03
	Deep Gradient Compression (Combine All Techniques)	93.28	+0.37

# GPUs in total	Word Error Rate (WER) of 5-Layer LSTM on AN4		
4	Baseline	10.76	0
	Naive Gradient Pruning	Not Converge	
	+ Local Gradient Accumulation	14.08	3.32
	+ Local Gradient Accumulation + Momentum Correction & Local Gradient Clipping	12.92	2.16
	+ Local Gradient Accumulation + Momentum Correction & Local Gradient Clipping	11.35	0.59
	Deep Gradient Compression (Combine All Techniques)	10.37	-0.39

Deep Gradient Compression: Reducing the Communication Bandwidth for Distributed Training [Lin et al 2017]

Deep Gradient Compression

Image Classification with 99.9% Gradient Sparsity



Deep Gradient Compression: Reducing the Communication Bandwidth for Distributed Training [Lin et al 2017]

Deep Gradient Compression

ImageNet Classification with 99.9% Gradient Sparsity

AlexNet trained on ImageNet Dataset

Training Method	Top-1	Top-5	Gradient Size	Compression Ratio
Baseline	58.17%	80.19%	232.56 MB	1 ×
DGC	58.20% (+0.03%)	80.20% (+0.01%)	0.39 MB	597 ×

Why not 1000x?
- Index overhead
- Biases are not pruned

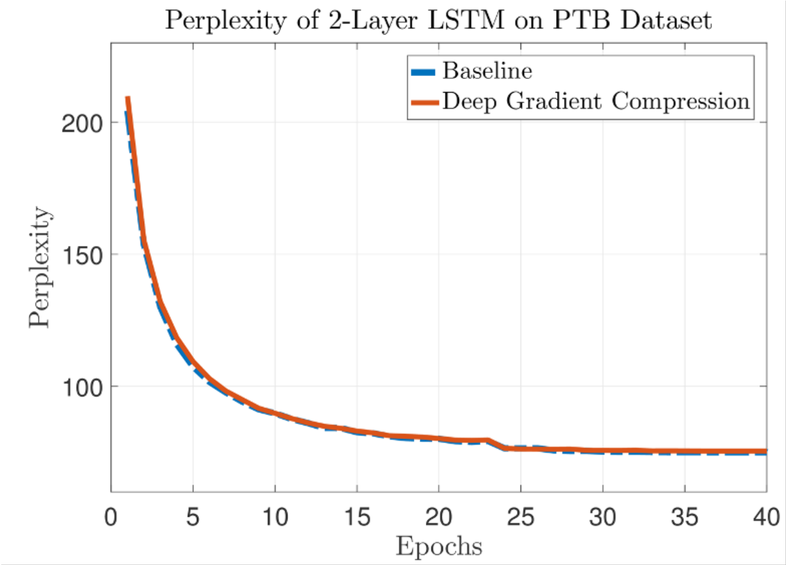
VGG trained on ImageNet Dataset

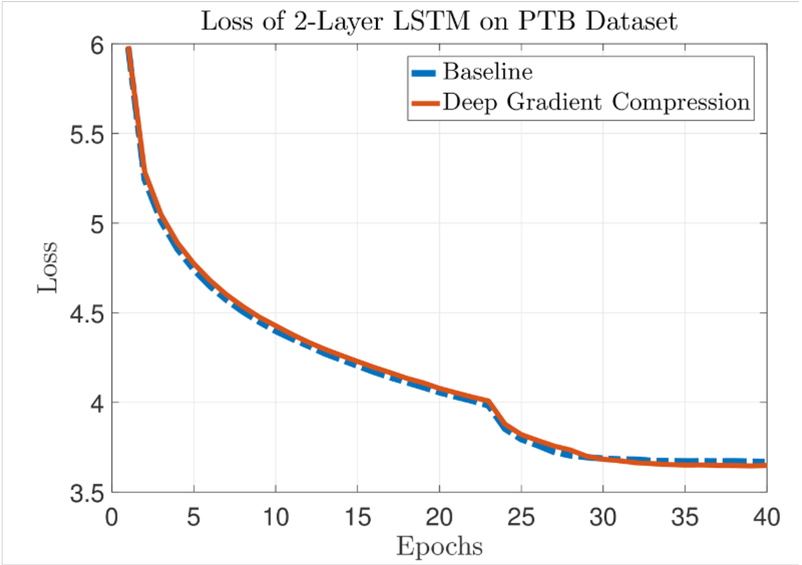
Training Method	Top-1	Top-5	Gradient Size	Compression Ratio
Baseline	75.96%	92.91%	97.49 MB	1 ×
DGC	76.15% (+0.19%)	92.97% (+0.06%)	0.35 MB	277 ×

Deep Gradient Compression: Reducing the Communication Bandwidth for Distributed Training [Lin et al 2017]

Deep Gradient Compression

Language Modeling with 99.9% Gradient Sparsity



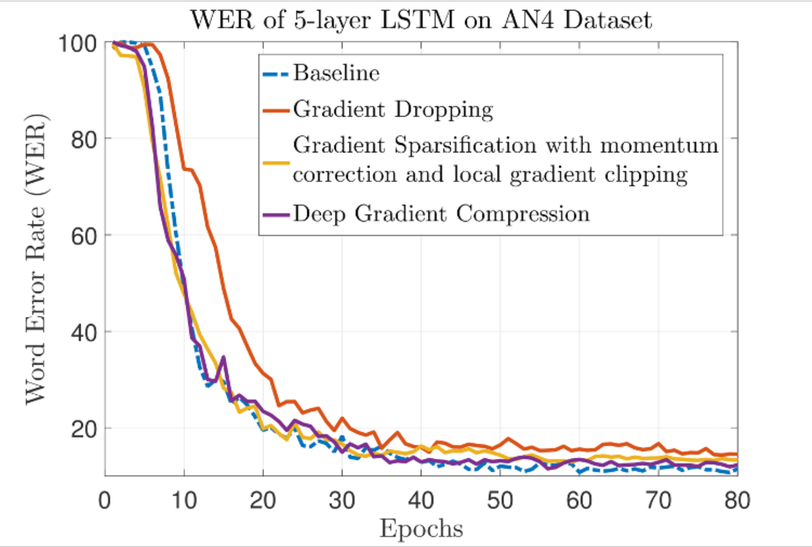


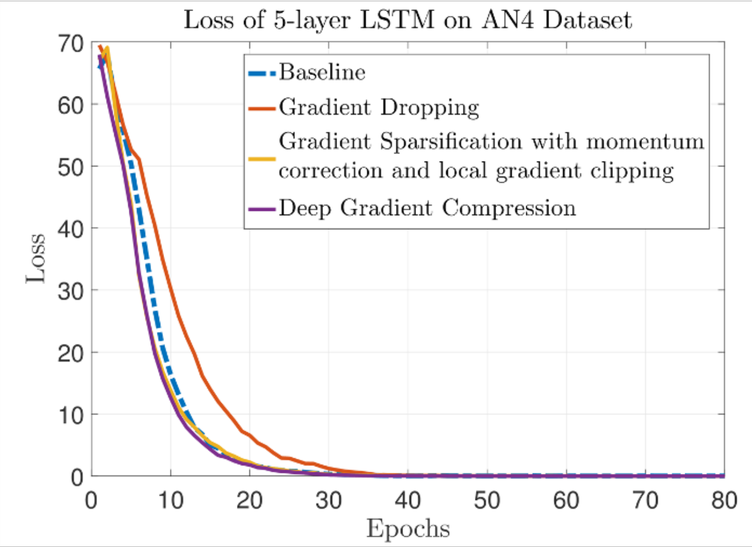
Training Method	Perplexity	Gradient Size	Compression Ratio
Baseline	72.30	194.68 MB	1 ×
Deep Gradient Compression	72.24 (-0.06)	0.42 MB	462 ×

Deep Gradient Compression: Reducing the Communication Bandwidth for Distributed Training [Lin et al 2017]

Deep Gradient Compression

Language Modeling with 99.9% Gradient Sparsity



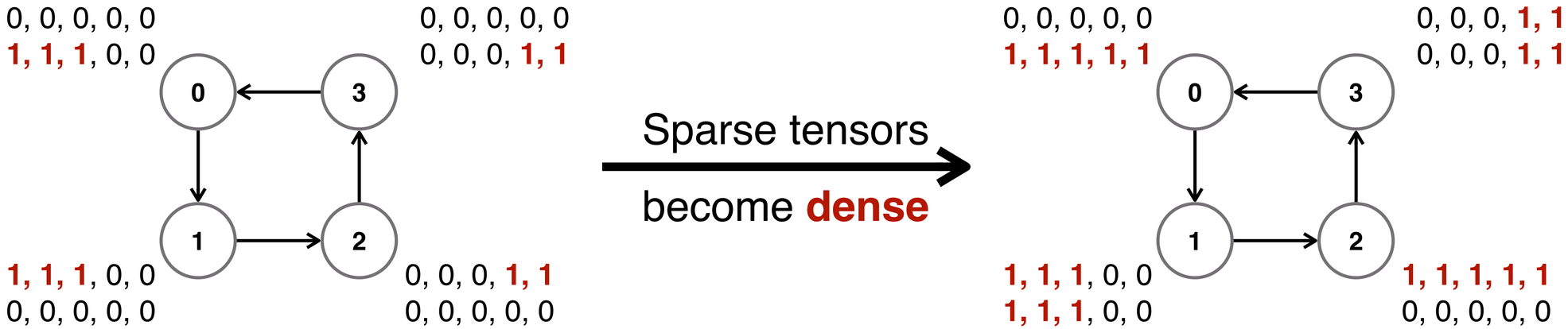


Training Method	Word Error Rate (WER) Test-clean	Test-other	Gradient Size	Compression Ratio
Baseline	9.45%	27.07%	488.08 MB	1 ×
Deep Gradient Compression	9.06% (-0.39)	27.04% (-0.03%)	0.74 MB	608 ×

Deep Gradient Compression: Reducing the Communication Bandwidth for Distributed Training [Lin et al 2017]

Deep Gradient Compression

Problem: sparse gradients gets denser during all-reduce



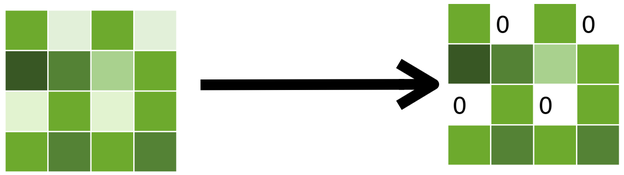
Deep Gradient Compression: Reducing the Communication Bandwidth for Distributed Training [Lin et al 2017]

PowerSGD: Low-Rank Gradient Compression

Motivation: Address the irregular sparse pattern in gradient compression, prevent gradients from getting denser

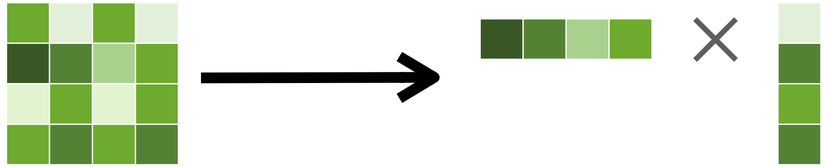
Method: Instead using fine-grained pruning, adapt low-rank factorization instead.

Deep Gradient Compression [Lin 2018]



The sparse pattern can be **different** on different servers.

PowerSGD [Vogels 2019]

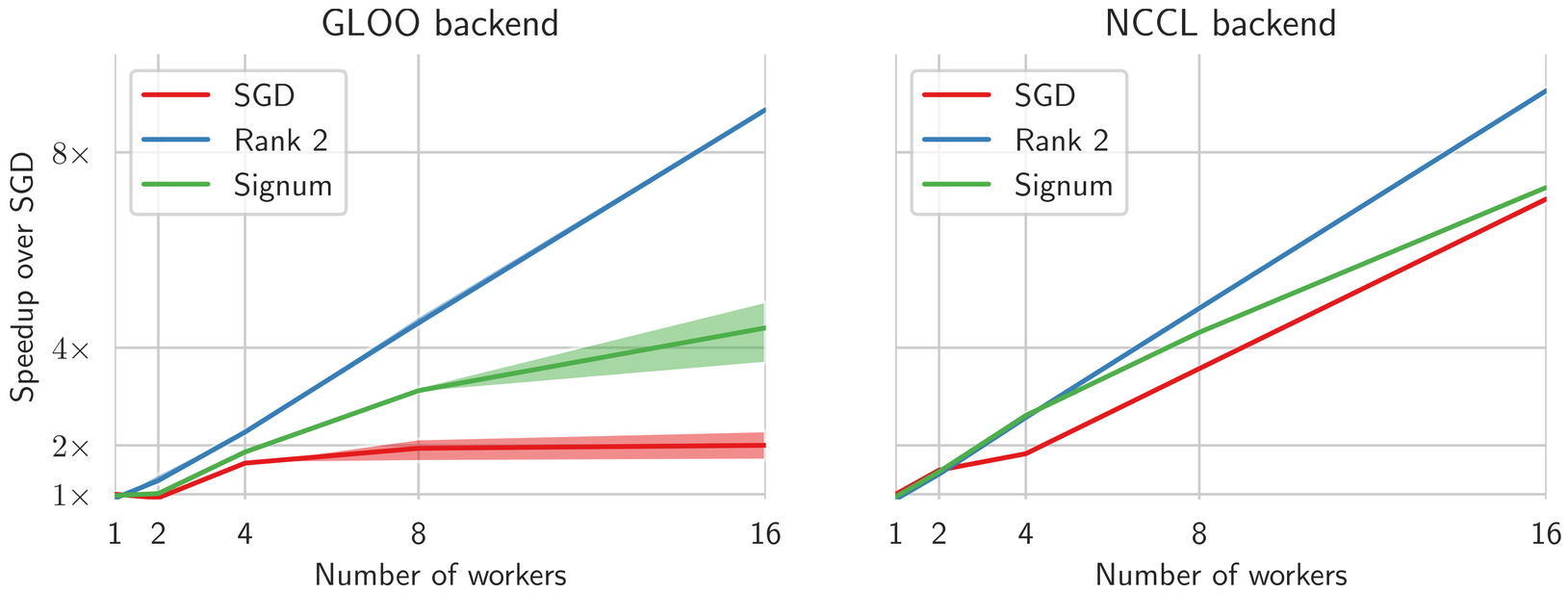


The low rank matrix dimension are **same** across servers.

PowerSGD: Practical Low-Rank Gradient Compression for Distributed Optimization [Vogels et al 2019]

PowerSGD: Low-Rank Gradient Compression

Speed Evaluation



There is an linear speedup when #num workers increases when using PowerSGD

PowerSGD: Practical Low-Rank Gradient Compression for Distributed Optimization [Vogels et al 2019]

PowerSGD: Low-Rank Gradient Compression

Accuracy Evaluation

Algorithm	Test accuracy	Data/epoch	Time per batch	
SGD	94.3%	1023 MB	312 ms	+0%
Atomo	92.6%	113 MB	948 ms	+204%
Signum	93.6%	32 MB	301 ms	−3%
Rank 2	94.4%	8 MB	239 ms	−23%

Dataset: CIFAR10

Algorithm	Test perplexity	Data/epoch	Time per batch	
SGD	91	7730 MB	300 ms	+0%
Signum	142	242 MB	424 ms	+41%
Rank 4	91	64 MB	134 ms	−55%

Dataset: WikiText-2

PowerSGD reduces the transferred gradient while keeping the accuracy at the same level.

PowerSGD: Practical Low-Rank Gradient Compression for Distributed Optimization [Vogels et al 2019]

Comparison of Gradient Pruning Methods

- Gradient Sparsification
  - Local gradient accumulation
  - Low sparsity ratio

- Deep Gradient Compression
  - Momentum correction, gradient clipping and warmup training.
  - Higher sparsity ratio (**99.9%**) while keeping the accuracy
  - Gradients get denser doing all-reduce

- PowerSGD
  - Choose low-rank instead of fine-grained pruning
  - Integrated into PyTorch.

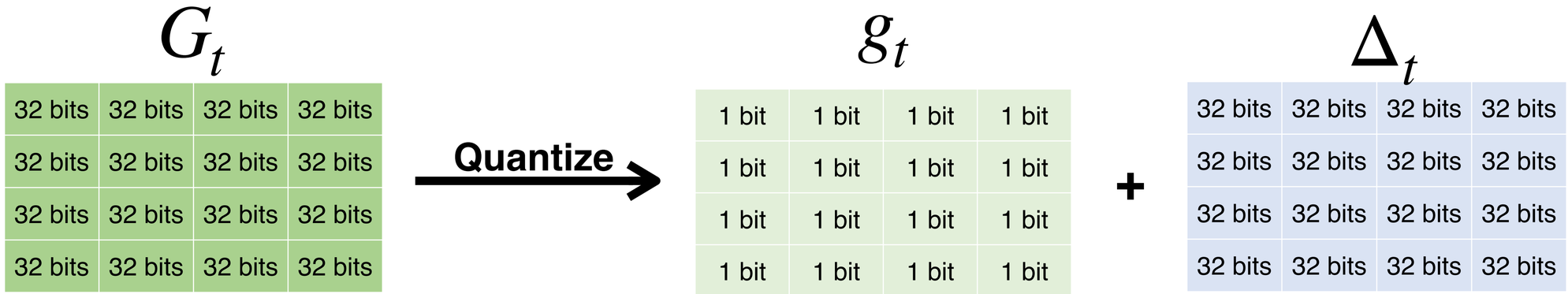
Gradient Compression

- Gradient Compression: Reduce the Gradient Size
  - Gradient Pruning
    - Sparse Communication
    - Deep Gradient Compression
    - PowerSGD
  - Gradient Quantization
    - 1-Bit SGD
    - TernGrad

Gradient Quantization

1 bit SGD

- With column-wise scaling factor
- Quantization error accumulation

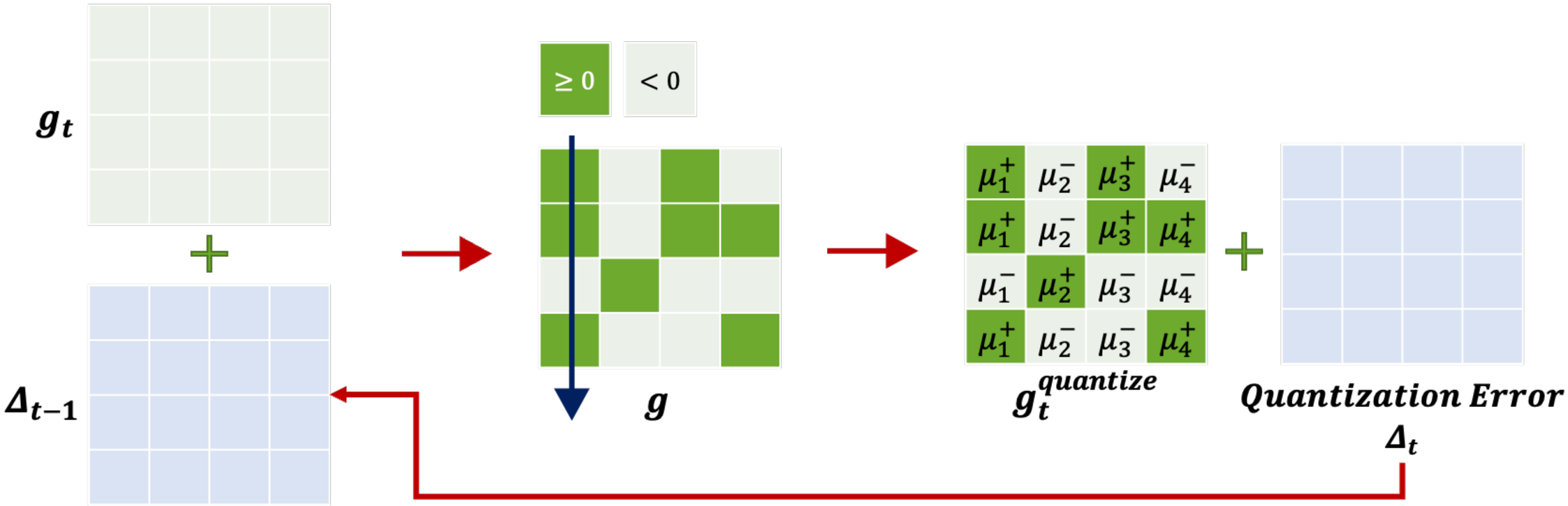


1-bit stochastic gradient descent and its application to data-parallel distributed training of speech DNNs [Frank 2014]

Gradient Quantization

1 bit SGD

- With column-wise scaling factor
- Quantization error accumulation

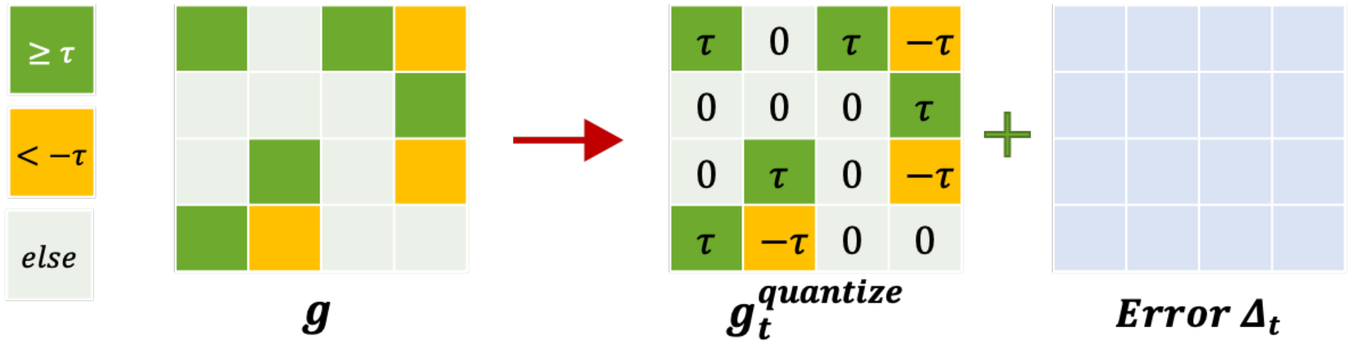


1-bit stochastic gradient descent and its application to data-parallel distributed training of speech DNNs [Frank 2014]

Gradient Quantization

Threshold Quantization

- τ – Threshold: τ is both threshold and reconstruction value, chosen in advance
- Quantization error accumulation
- Need to empirically choose τ

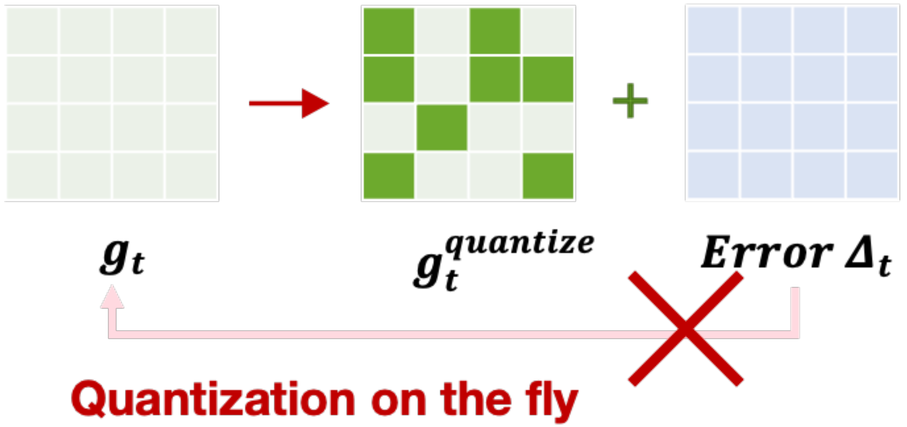


Scalable distributed DNN training using commodity GPU cloud computing [Nikko 2015]

Gradient Quantization

TernGrad

- Quantize $\frac{g_i}{\max(\mathbf{g})}$ to 0, 1, -1 with probability $\frac{|g_i|}{\max(\mathbf{g})}$, s.t. $\mathbb{E}\Big(\text{Quantize}\big(g_i\big)\Big) = g_i$
- No quantization error accumulation, i.e., on the fly

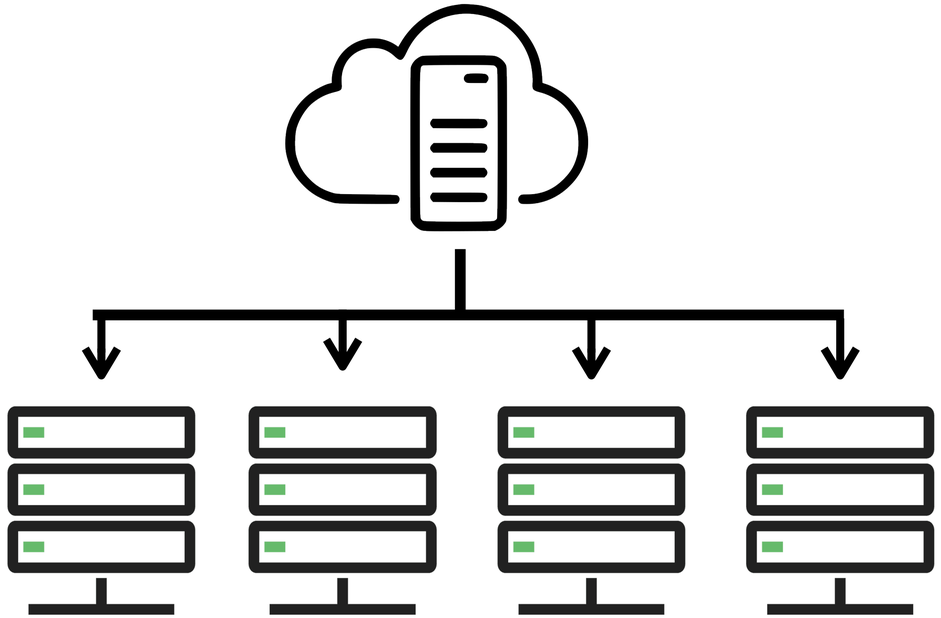


QSGD: Randomized quantization for communication-optimal stochastic gradient descent [Dan et al 2016]

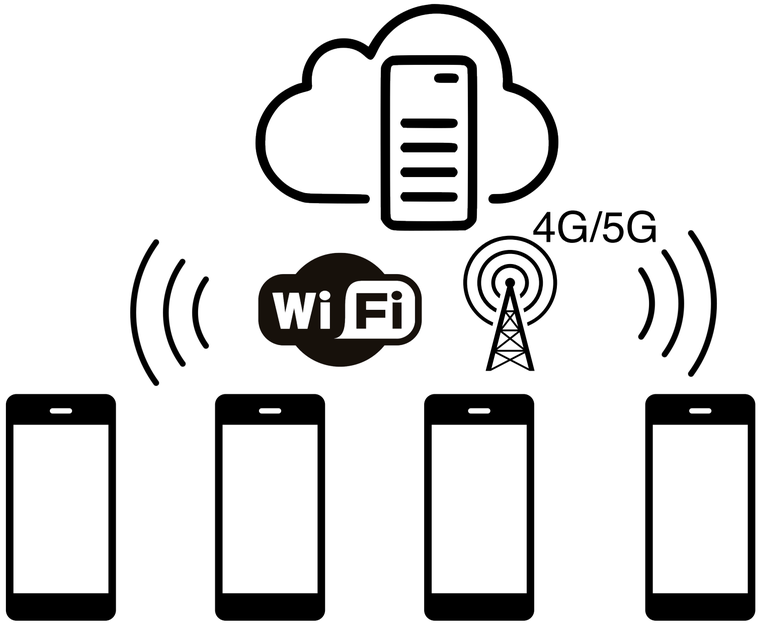
Section 3: Delayed Gradient Update

Tolerate networking latency

Latency Bottleneck



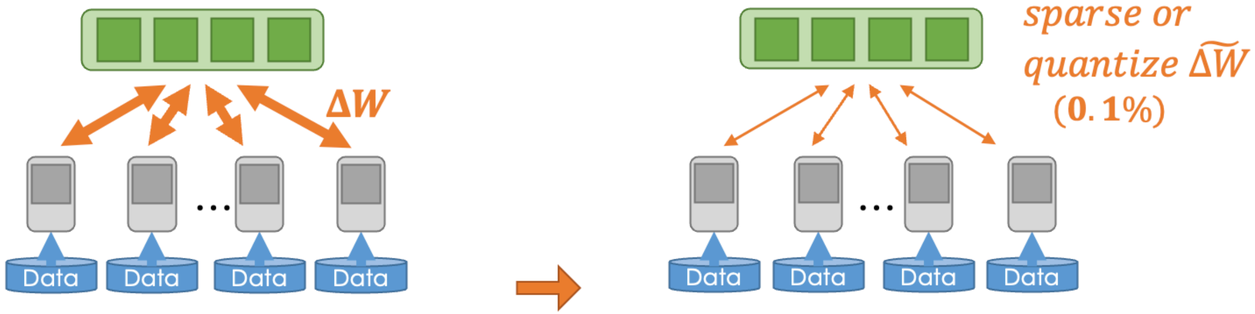
Connected through wired ethernet or infinity band
Bandwidth as high as **100Gb/s**, Latency as low as **1us**



Connected through WiFi or Cellular network
Bandwidth up to **1Gbp/s**, Latency **~200ms**.

Delayed Gradient Averaging: Tolerate the Communication Latency in Federated Learning. [Zhu 2021]

Latency Bottleneck



Bandwidth can be always improved by

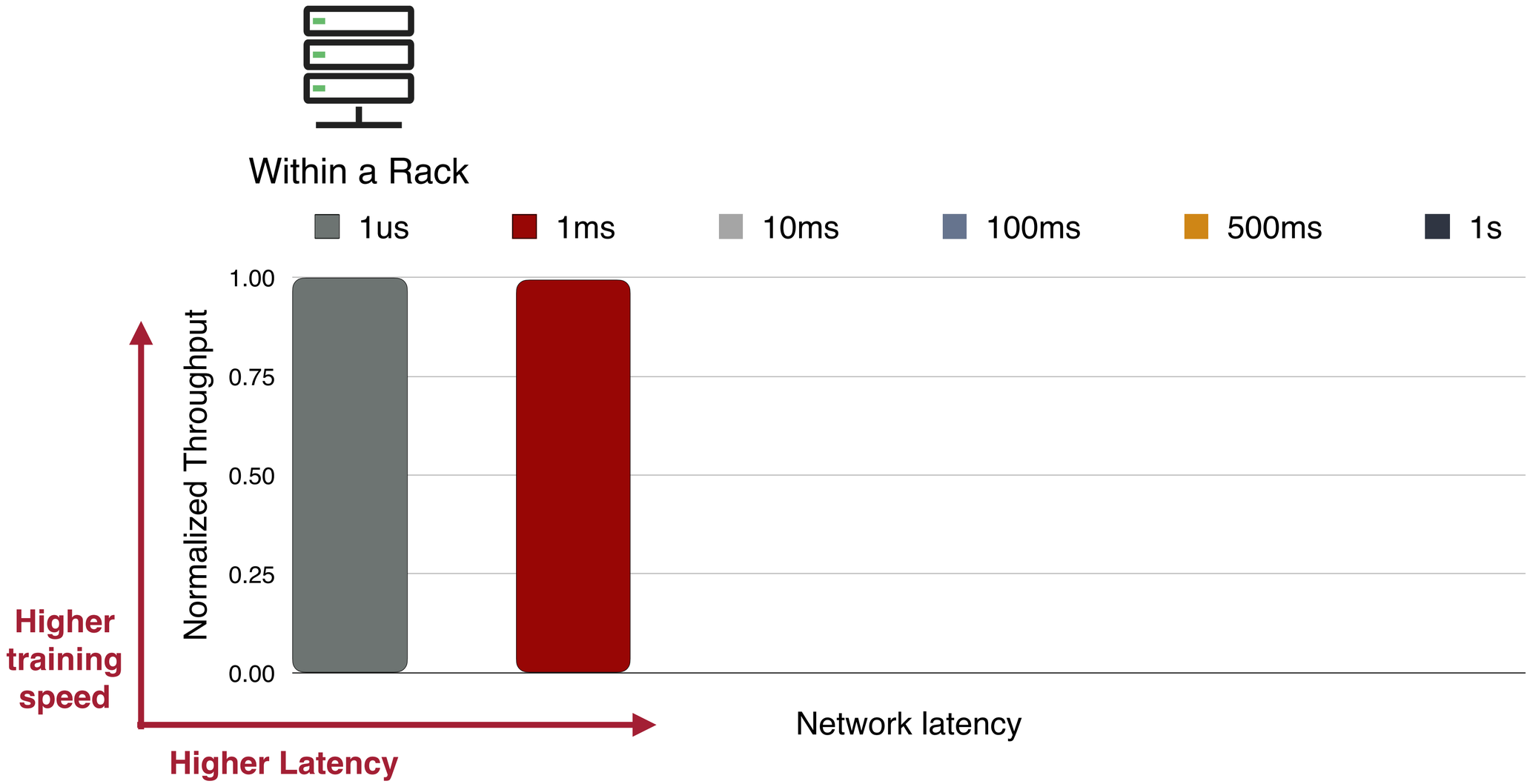
- Hardware upgrade
- Gradient compression and quantization

Latency **is hard to improve** because

- Physical limits: traveling from Shanghai to Boston at the speed of light still takes 162ms.

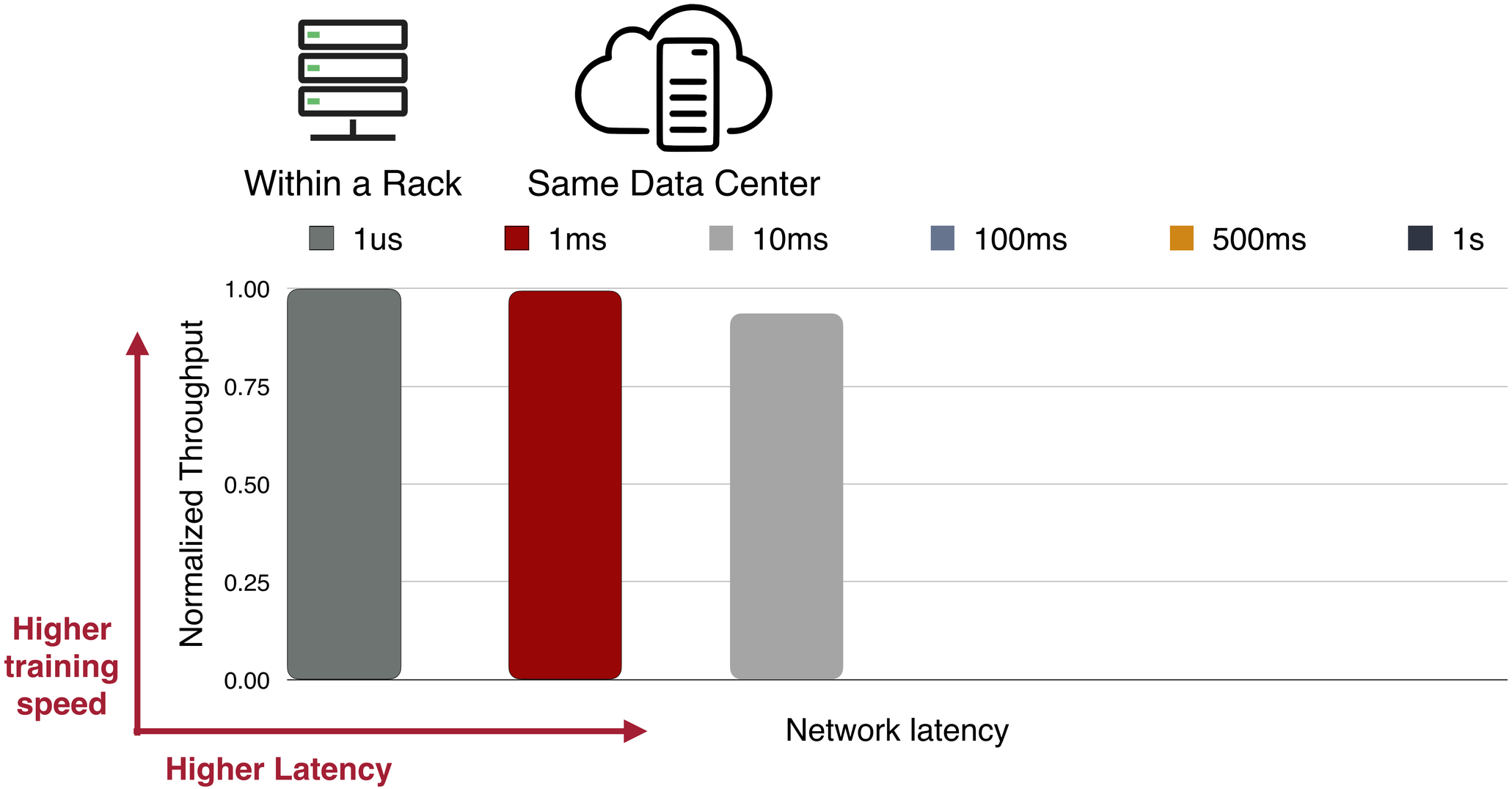
Delayed Gradient Averaging: Tolerate the Communication Latency in Federated Learning. [Zhu 2021]

High Network Latency Slows Distributed Training



Delayed Gradient Averaging: Tolerate the Communication Latency in Federated Learning. [Zhu 2021]

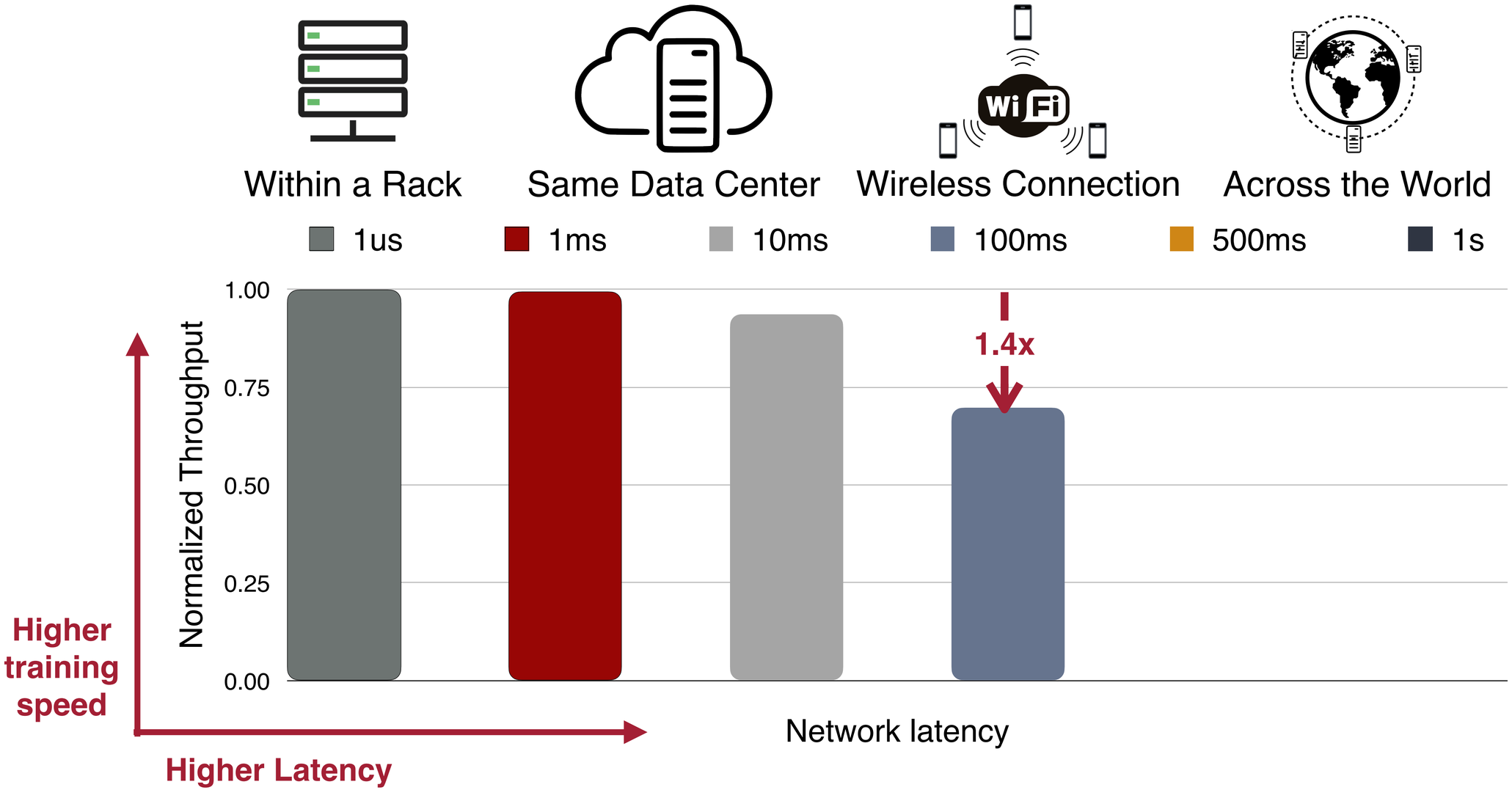
High Network Latency Slows Federated Learning



In cluster network latency does not affect training

Delayed Gradient Averaging: Tolerate the Communication Latency in Federated Learning. [Zhu 2021]

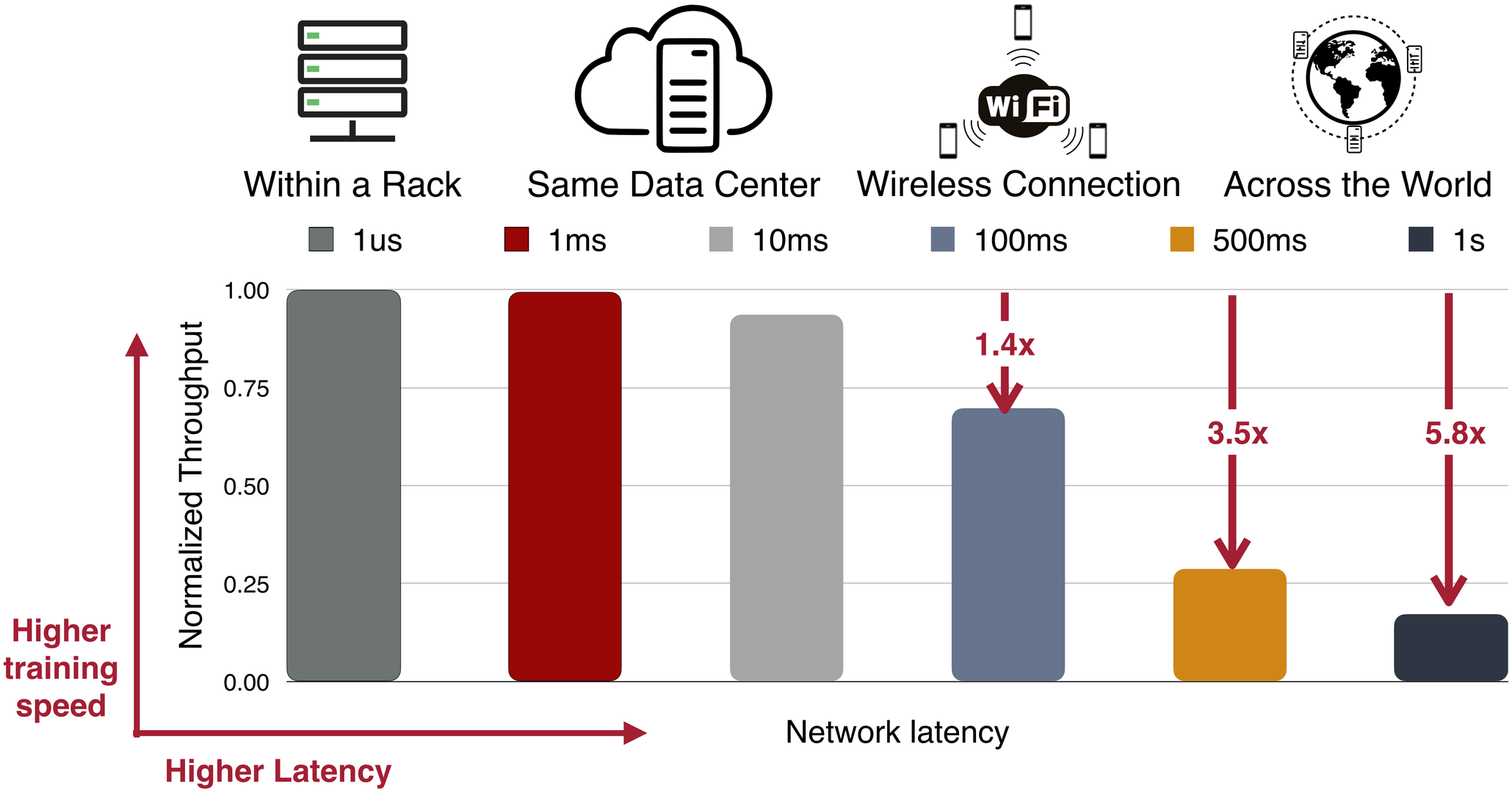
High Network Latency Slows Federated Learning



In home wireless connection slows the training by certain margin.

Delayed Gradient Averaging: Tolerate the Communication Latency in Federated Learning. [Zhu 2021]

High Network Latency Slows Federated Learning



Long-distance connection slows the training by a large margin.

Delayed Gradient Averaging: Tolerate the Communication Latency in Federated Learning. [Zhu 2021]

Conventional Algorithms Suffer from High Latency

Vanilla Distributed Synchronous SGD

Computation Communication

i: iteration, j: work index, x: training data, w: model weights

Conventional Algorithms Suffer from High Latency

Vanilla Distributed Synchronous SGD

Computation Communication

i: iteration, j: work index, x: training data, w: model weights

Conventional Algorithms Suffer from High Latency

Vanilla Distributed Synchronous SGD

Local updates and communication are performed sequentially.
Worker **has to wait the transmission finish** before next step.

Computation Communication

i: iteration, j: work index, x: training data, w: model weights

Delayed Gradient Averaging: Tolerate the Communication Latency in Federated Learning. [Zhu 2021]

Conventional Algorithms Suffer from High Latency

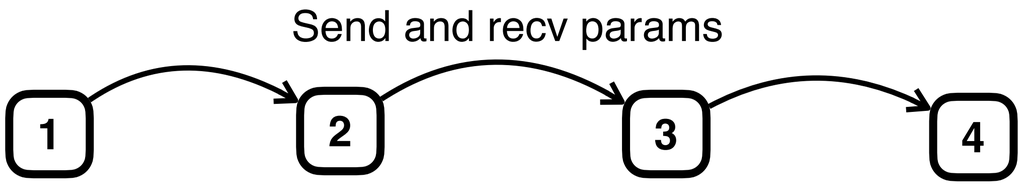
Vanilla Distributed Synchronous SGD

How to improve training throughput under high latency?

Can we allow stale gradients? So we can overlap communication with computation
analogy: use late days to submit your homework (gradient)

Delayed Gradient Averaging: Tolerate the Communication Latency in Federated Learning. [Zhu 2021]

Delayed Gradient Averaging

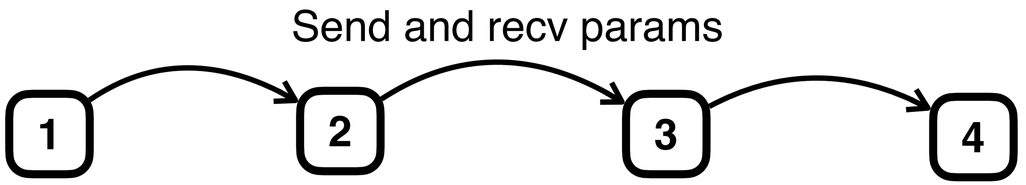


(Distributed SGD)

Without delay: all the local machines are blocked to wait for the synchronization to finish

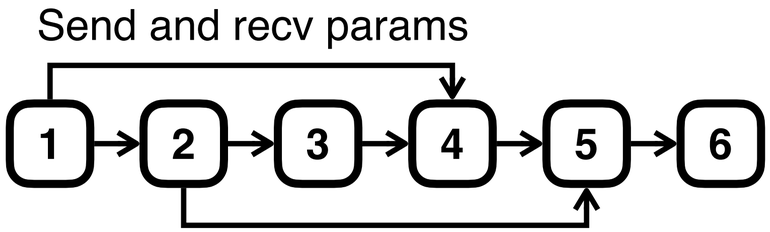
Delayed Gradient Averaging: Tolerate the Communication Latency in Federated Learning. [Zhu 2021]

Delayed Gradient Averaging



(Distributed SGD)

Without delay: all the local machines are blocked to wait for the synchronization to finish



(Delayed Gradient Averaging)

With delay: Worker keep performing local updates while the parameters are in transmission.

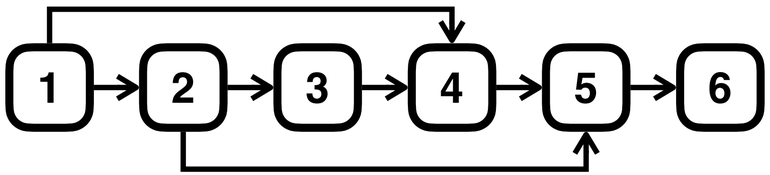
Delayed Gradient Averaging: Tolerate the Communication Latency in Federated Learning. [Zhu 2021]

Delayed Gradient Averaging

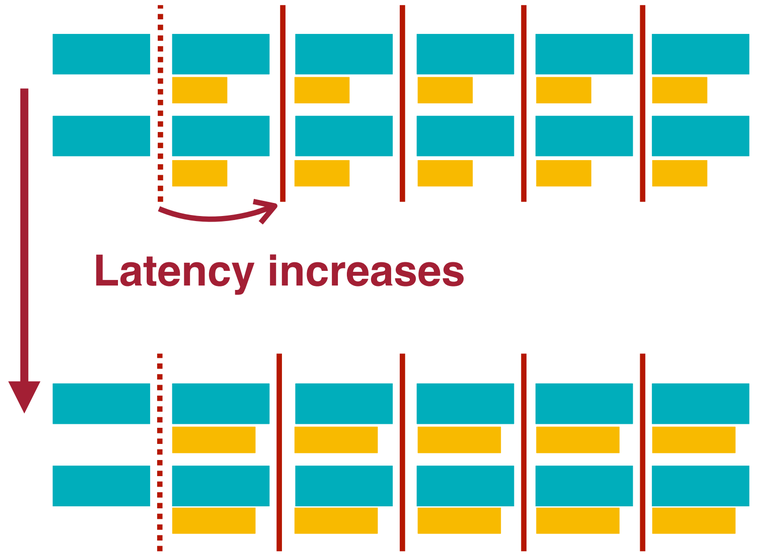
```
G = {}
for iter in 1...max_iters:
 g = grad(net, data)
 send(m, id=iter)
 G[iter] = g
 m = 0
 avg_g = recv(id=iter - D)
 W = W - lr * (g - G[iter-D] + avg_g)
```

Delay *D* steps

Send and recv params



Communication is covered by computation.



Total training time is not affected.

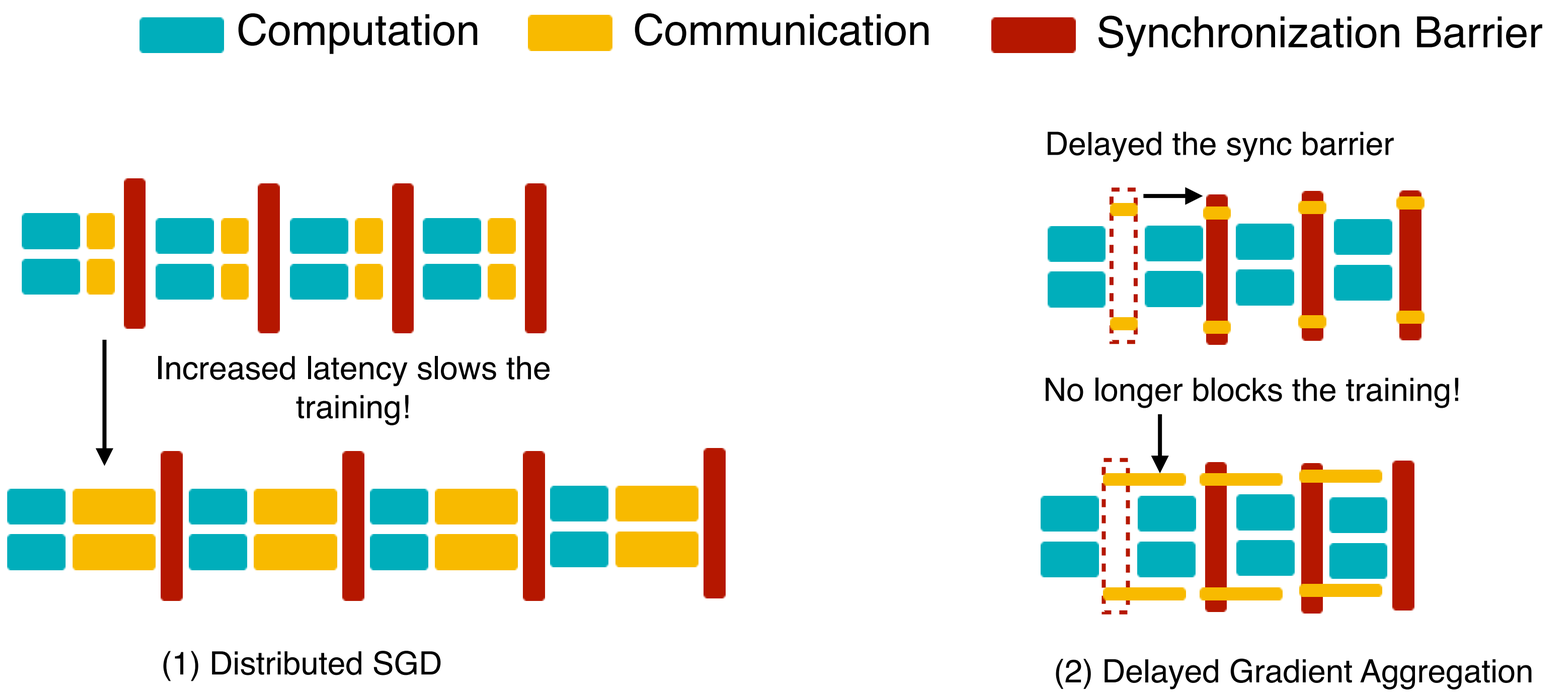
Key idea: delay the receiving timing of i^{th} gradient to a later iteration $(i + D)^{th}$

Computation
Communication
Synchronization Barrier

Delayed Gradient Averaging: Tolerate the Communication Latency in Federated Learning. [Zhu 2021]

Delayed Gradient Averaging

Compared SGD, FedAvg and DGA



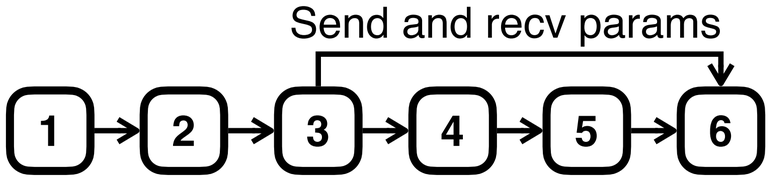
As long the transmission finishes in D rounds of local computation, then it will not block the training.

Delayed Gradient Averaging: Tolerate the Communication Latency in Federated Learning. [Zhu 2021]

Delayed Gradient Averaging

Deal with Staleness

- At 6^{th} iteration, the averaged gradients from 3^{rd} iteration arrives (example on the right)



- How to apply this to optimizers?

- Directly apply

$$w_{(i,j)} = w_{(i,j)} - \eta \overline{\nabla w_{(i-D)}}$$

will hurt the model performance.

ResNet18, CIFAR10

	w/o correction	w/ gradient correction
D=5	88.7	89.2
D=10	86.9	89.3
D=15	85.5	89.0
D=20	84.2	88.7

- Apply with correction terms

$$w_{(i,j)} = w_{(i,j)} - \eta(\nabla w_{(i,j)} - \nabla w_{(i-D,j)} + \overline{\nabla w_{(i-D)}})$$

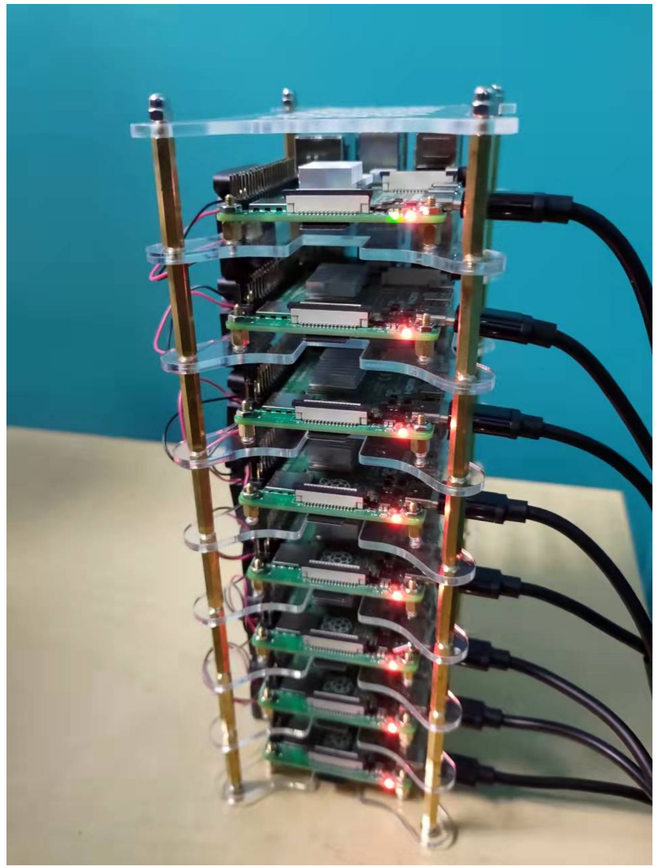
Delayed Gradient Averaging: Tolerate the Communication Latency in Federated Learning. [Zhu 2021]

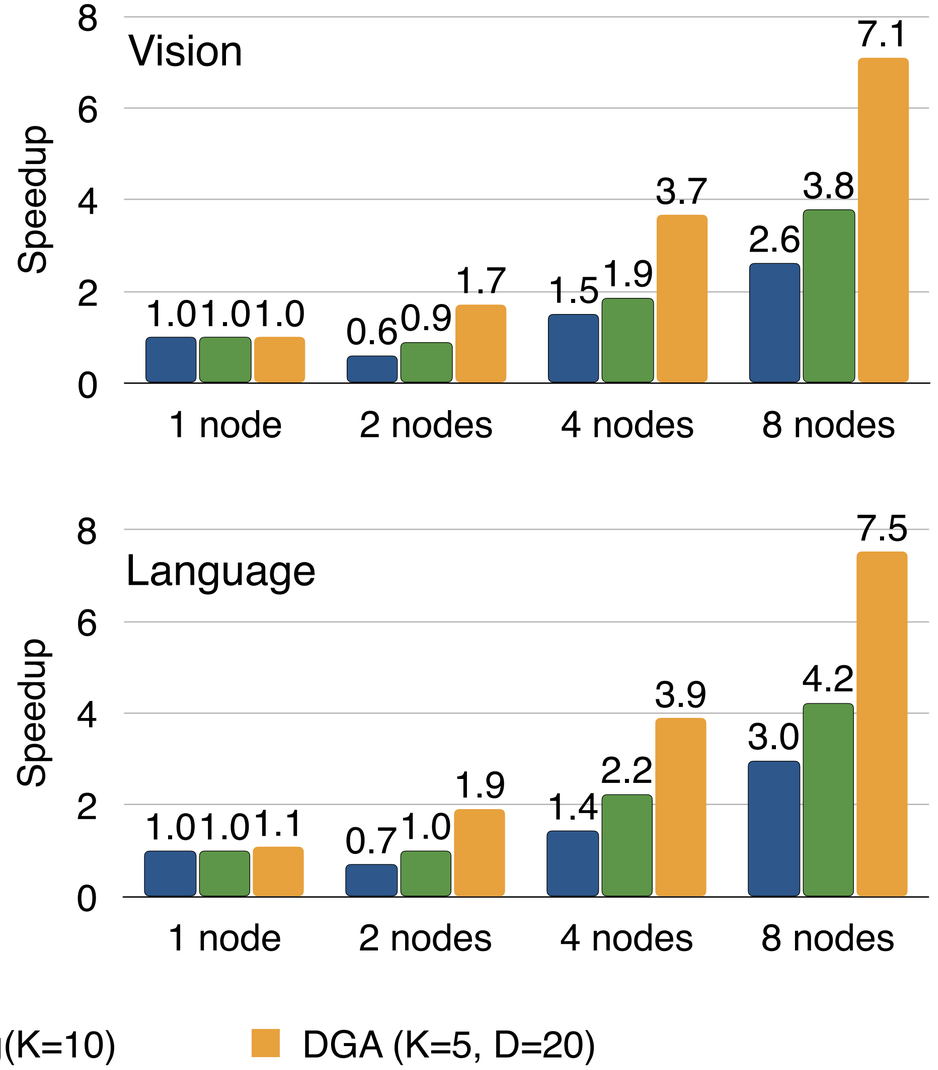
DGA Accuracy Evaluation

Datasets	Partition	FedAvg (K=5)		FedAvg (K=10)		FedAvg (K=20)		DGA (K=5, D=20)	
		Acc	Speedup	Acc	Speedup	Acc	Speedup	Acc	Speedup
CIFAR	i.i.d	88.7	1×	88.5	1.51×	88.1	2.05×	88.6	3.16×
	non-i.i.d	48.2		47.2		43.9		48.0	
ImageNet	i.i.d	76.6	1×	76.5	1.43×	76.2	1.81×	76.4	2.55×
	non-i.i.d	55.4		52.5		48.6		54.9	
Shakespeare	i.i.d	47.6	1×	47.3	1.66×	47.4	2.51×	47.1	4.07×
	non-i.i.d	36.9		34.3		30.1		36.3	

Delayed Gradient Averaging: Tolerate the Communication Latency in Federated Learning. [Zhu 2021]

Real-world Benchmark





Delayed Gradient Averaging: Tolerate the Communication Latency in Federated Learning. [Zhu 2021]

Summary of Today's Lecture

- The bottleneck of scaling a distributed training is the communication.
- Overcome the networking bandwidth bottleneck:
  - Deep gradient compression (sparse communication, momentum correction, gradient clipping and warm-up training; quantized gradient (1-bit, ternary, etc)
- Overcome the network latency bottleneck:
  - Delayed Gradient Averaging allows stale gradient, pipelines the communication and computation, tolerates the latency in distributed training.

References

- Sparse communication for distributed gradient descent [Alham Fikri et al 2017]
- Deep Gradient Compression: Reducing the Communication Bandwidth for Distributed Training [Lin et al 2017]
- PowerSGD: Practical Low-Rank Gradient Compression for Distributed Optimization [Vogels et al 2019]
- 1-bit stochastic gradient descent and its application to data-parallel distributed training of speech DNNs [Frank 2014]
- Scalable distributed DNN training using commodity GPU cloud computing [Nikko 2015]
- QSGD: Randomized quantization for communication-optimal stochastic gradient descent [Dan et al 2016]
- Delayed Gradient Averaging: Tolerate the Communication Latency in Federated Learning. [Zhu 2021]