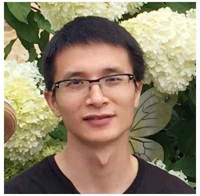
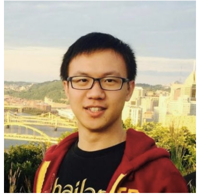
TernGrad: Ternary Gradients to Reduce Communication in Distributed Deep Learning

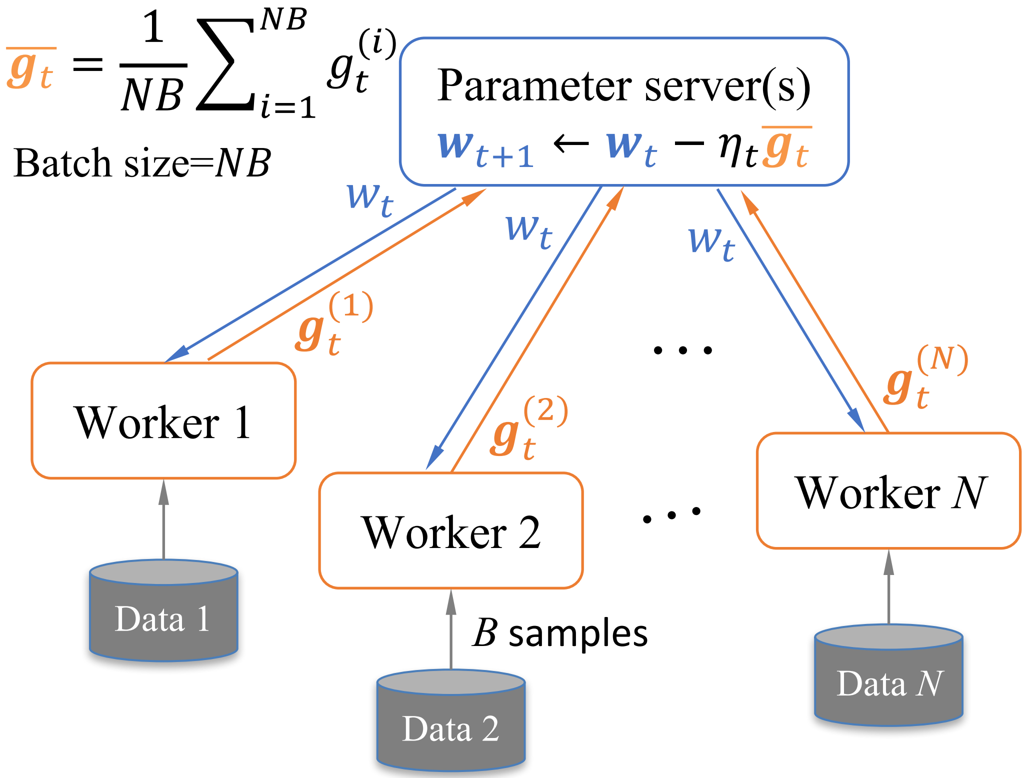
Wei Wen[1], Cong Xu[2], Feng Yan[3], Chunpeng Wu[1],
Yandan Wang[4], Yiran Chen[1], Hai (Helen) Li[1]

Duke University[1], Hewlett Packard Labs[2],
University of Nevada - Reno[3], University of Pittsburgh[4]




Background - Distributed Deep Learning



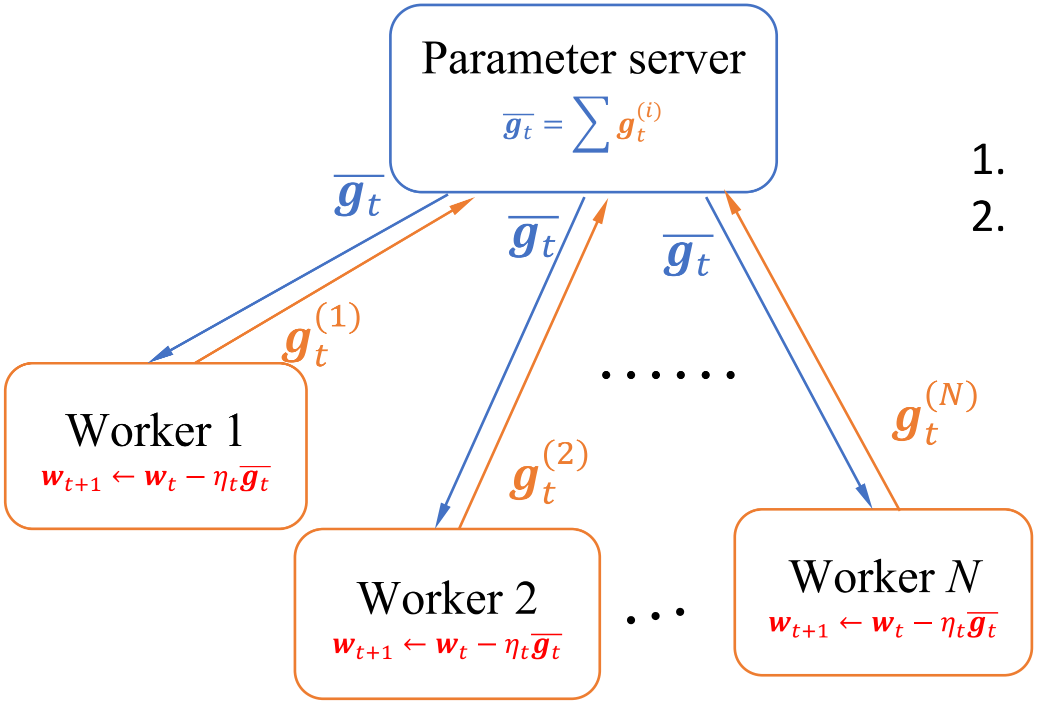
Synchronized Data Parallelism for Stochastic Gradient Descent (SGD):

1. Training data is split to N subsets
2. Each worker has a model replica (copy)
3. Each replica is trained on a data subset
4. Synchronization in parameter server(s)

Scalability:

1. Computing time decreases with N
2. Communication can be the bottleneck
3. This work: quantizing gradient to three (i.e., *ternary*) levels {-1, 0, 1} (<2bits)

An Alternative Setting



1. Only exchange gradients
2. Gradient quantization can reduce communication in both directions

Stochastic Gradients without Bias

Batch Gradient Descent

$$C(\boldsymbol{w}) \triangleq \frac{1}{n}\sum_{i=1}^{n} Q(\boldsymbol{z}_i, \boldsymbol{w})$$

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \frac{\eta_t}{n}\sum_{i=1}^{n} g_t^{(i)}$$

SGD

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta_t \cdot g_t^{(I)}$$

I is randomly drawn from [1,*n*]

$$\mathrm{E}\left\{g_t^{(I)}\right\} = \nabla C(\boldsymbol{w})$$

No bias

TernGrad

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta_t \cdot ternarize\left(g_t^{(I)}\right)$$

$$\mathrm{E}\left\{ternarize\left(g_t^{(I)}\right)\right\} = \nabla C(\boldsymbol{w})$$

No bias

TernGrad is Simple

$$\tilde{\boldsymbol{g}}_t = ternarize(\boldsymbol{g}_t) = s_t \cdot sign\left(\boldsymbol{g}_t\right) \circ \boldsymbol{b}_t$$

$$s_t \triangleq ||\boldsymbol{g}_t||_\infty \triangleq max\left(abs\left(\boldsymbol{g}_t\right)\right)$$

$$\begin{cases} P(b_{tk} = 1 \mid \boldsymbol{g}_t) = |g_{tk}|/s_t \\ P(b_{tk} = 0 \mid \boldsymbol{g}_t) = 1 - |g_{tk}|/s_t \end{cases}$$

Example:

$\boldsymbol{g}_t^{(i)}$: [0.30, -1.20, …, 0.9]

s_t: 1.20

Signs: [1, -1, …, 1]

$P(b_{tk} = 1|g_t)$: $[\frac{0.3}{1.2}, \frac{1.2}{1.2}, \ldots, \frac{0.9}{1.2}]$

$\boldsymbol{b}_t$: [0,1,…,1]

$\widetilde{\boldsymbol{g}_t^{(i)}}$: [0, -1, …, 1]*1.20

$$\mathbf{E}_{\boldsymbol{z},\boldsymbol{b}}\left\{\tilde{\boldsymbol{g}}_t\right\} = \mathbf{E}_{\boldsymbol{z},\boldsymbol{b}}\left\{s_t \cdot sign\left(\boldsymbol{g}_t\right) \circ \boldsymbol{b}_t\right\}$$

$$= \mathbf{E}_{\boldsymbol{z}}\left\{s_t \cdot sign\left(\boldsymbol{g}_t\right) \circ \mathbf{E}_{\boldsymbol{b}}\left\{\boldsymbol{b}_t | \boldsymbol{z}_t\right\}\right\} = \mathbf{E}_{\boldsymbol{z}}\left\{\boldsymbol{g}_t\right\} = \nabla_{\boldsymbol{w}} C(\boldsymbol{w}_t)$$

No bias

Convergence

Standard SGD almost truly converges under assumptions (Fisk 1965, Metivier 1981&1983, Bottou 1998)

Assumption 1:

$C(\boldsymbol{w})$ has a single minimum $\boldsymbol{w}^*$ and $\forall \epsilon > 0, \inf_{||\boldsymbol{w}-\boldsymbol{w}^*||^2>\epsilon} (\boldsymbol{w}-\boldsymbol{w}^*)^T \nabla_{\boldsymbol{w}} C(\boldsymbol{w}) > 0$

Assumption 2:

Learning rate γ_t decreases neither very fast nor very slow $\begin{cases} \sum_{t=0}^{+\infty} \gamma_t^2 < +\infty \\ \sum_{t=0}^{+\infty} \gamma_t = +\infty \end{cases}$

Assumption 3 (gradient bound):

$$\mathbf{E}\left\{||\boldsymbol{g}||^2\right\} \leq A + B\,||\boldsymbol{w}-\boldsymbol{w}^*||^2$$

Standard SGD *almost-truly* converges

Assumption 3 (gradient bound):

$$\mathbf{E}\left\{||\boldsymbol{g}||_\infty \cdot ||\boldsymbol{g}||_1\right\} \leq A + B\,||\boldsymbol{w}-\boldsymbol{w}^*||^2$$

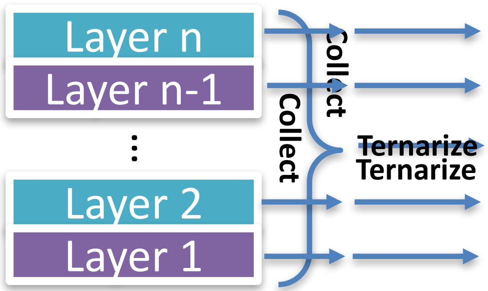
TernGrad almost-truly converges

$$\mathbf{E}\left\{||\boldsymbol{g}||^2\right\} \leq \mathbf{E}\left\{||\boldsymbol{g}||_\infty \cdot ||\boldsymbol{g}||_1\right\} \leq A + B\,||\boldsymbol{w}-\boldsymbol{w}^*||^2$$

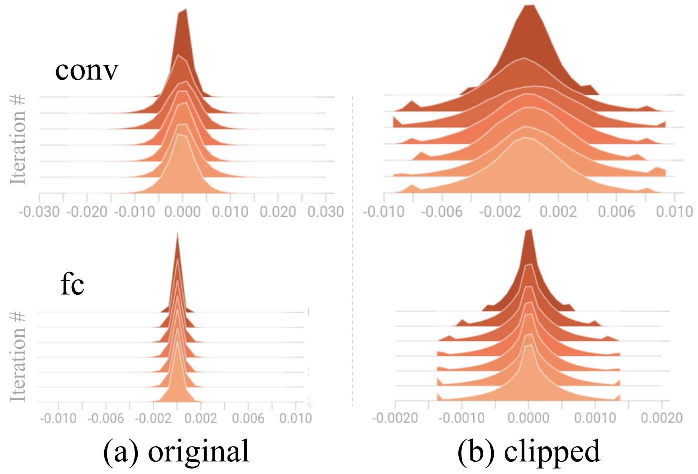
Stronger gradient bound in *TernGrad*

Closing Bound Gap

Two methods to push the gradient bound of *TernGrad* closer to the bound of standard SGD



Layer-wise ternarizing



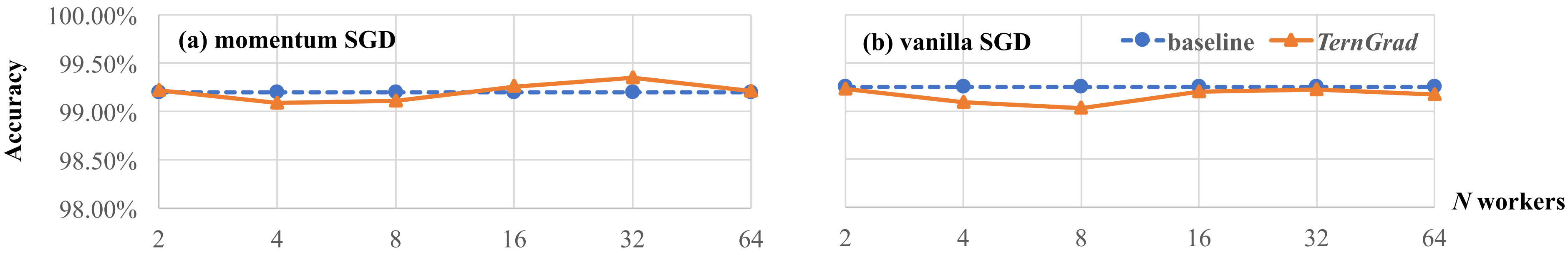
Gradient clipping

(More in poster session)

Integration with Manifold Optimizers

(All experiments: All hyper-parameters are tuned for standard SGD and fixed in *TernGrad*)

LeNet (total mini-batch size 64): close accuracy & randomness in *TernGrad* results in small variance



CIFAR-10, mini-batch size 64 per worker

Adam: D. P. Kingma, 2014

SGD	base LR	total mini-batch size	iterations	gradients	workers	accuracy
Adam	0.0002	128	300K	floating *TernGrad*	2 2	86.56% 85.64% (-0.92%)
Adam	0.0002	2048	18.75K	floating *TernGrad*	16 16	83.19% 82.80% (-0.39%)

Scaling to Large-scale Deep Learning

TernGrad: Randomness & regularization → (1) decrease randomness in dropout or (2) use smaller weight decay

No new hyper-parameters added

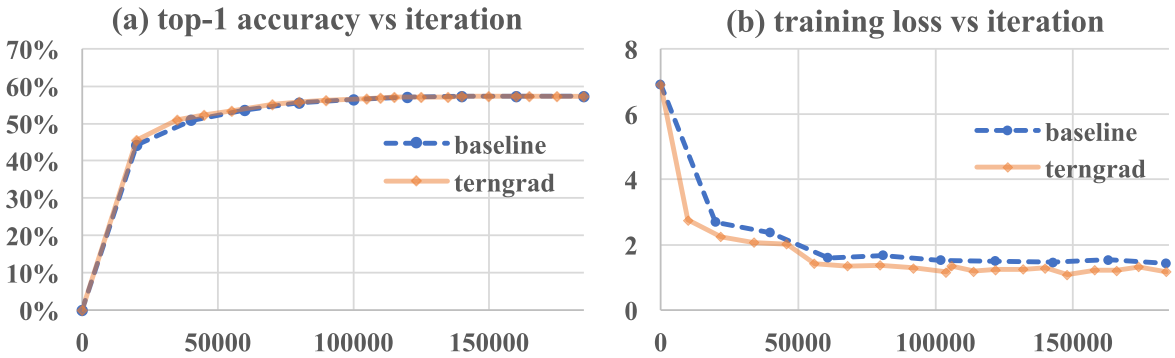
base LR	mini-batch size	workers	iterations	gradients	weight decay	DR†	top-1	top-5
0.01	256	2	370K	floating	0.0005	0.5	57.33%	80.56%
				TernGrad	0.0005	0.2	57.61%	80.47%
				TernGrad-noclip ‡	0.0005	0.2	54.63%	78.16%
0.02	512	4	185K	floating	0.0005	0.5	57.32%	80.73%
				TernGrad	0.0005	0.2	57.28%	80.23%
0.04	1024	8	92.5K	floating	0.0005	0.5	**56.62%**	80.28%
				TernGrad	0.0005	0.2	**57.54%**	80.25%

† DR: dropout ratio, the ratio of dropped neurons. ‡ *TernGrad* without gradient clipping.

AlexNet

N. S. Keskar, et al., ICLR 2017

Scaling to Large-scale Deep Learning



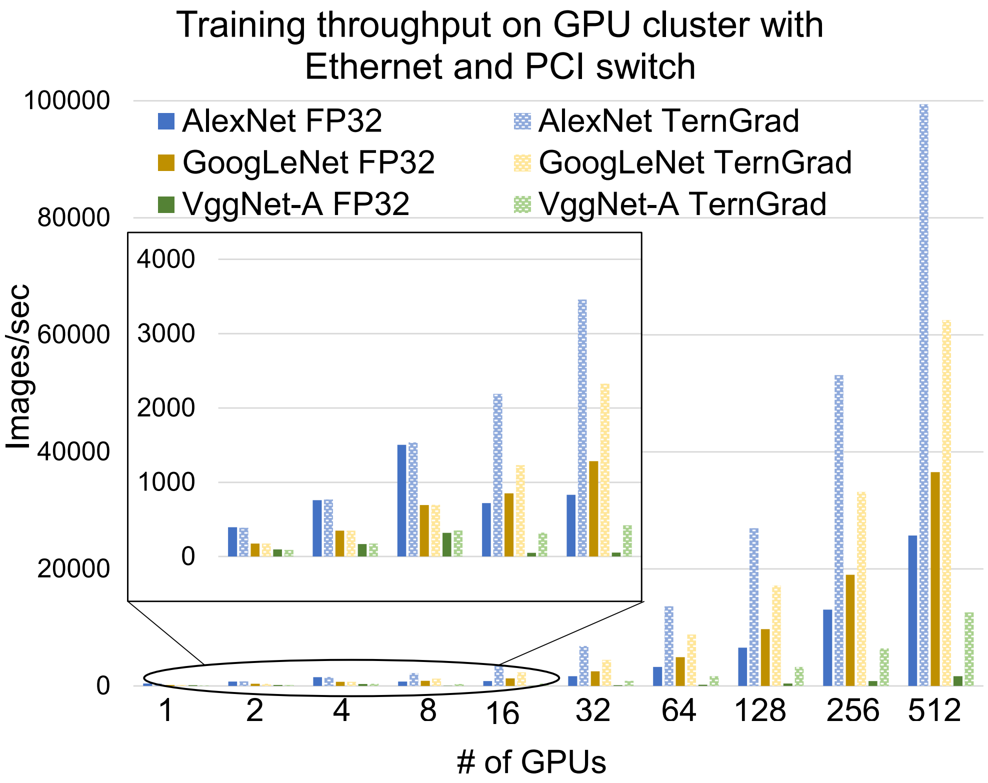
AlexNet trained on 4 workers with mini-batch size 512

base LR	mini-batch size	workers	iterations	gradients	weight decay	DR	top-5
0.04	128	2	600K	floating	4e-5	0.2	88.30%
				TernGrad	1e-5	0.08	86.77%
0.08	256	4	300K	floating	4e-5	0.2	87.82%
				TernGrad	1e-5	0.08	85.96%
0.10	512	8	300K	floating	4e-5	0.2	89.00%
				TernGrad	2e-5	0.08	86.47%

GoogLeNet
<2% on avg.

Tune hyper-parameters for *TernGrad* may reduce accuracy gap

Performance Model



TernGrad gives higher speedup when

1. using more workers
2. using smaller communication bandwidth (Ethernet vs InfiniBand)
3. training DNNs with more fully-connected layers (VggNet vs GoogLeNet)

Conclusion

- Communication reduction by ternary gradients - *TernGrad*
- *TernGrad* can train from scratch and coverages
  - within the same epochs
  - using the same learning rate policy
- Easy to be implemented
  - https://github.com/wenwei202/terngrad
- Related work @ NIPS 2017
  - Dan Alistarh, *et al*. (Spotlight)
  - Xiangru Lian, *et al.* (Oral)

Thanks!

Poster # **127** @ Pacific Ballroom

Wed Dec 6th 06:30 -- 10:30 PM