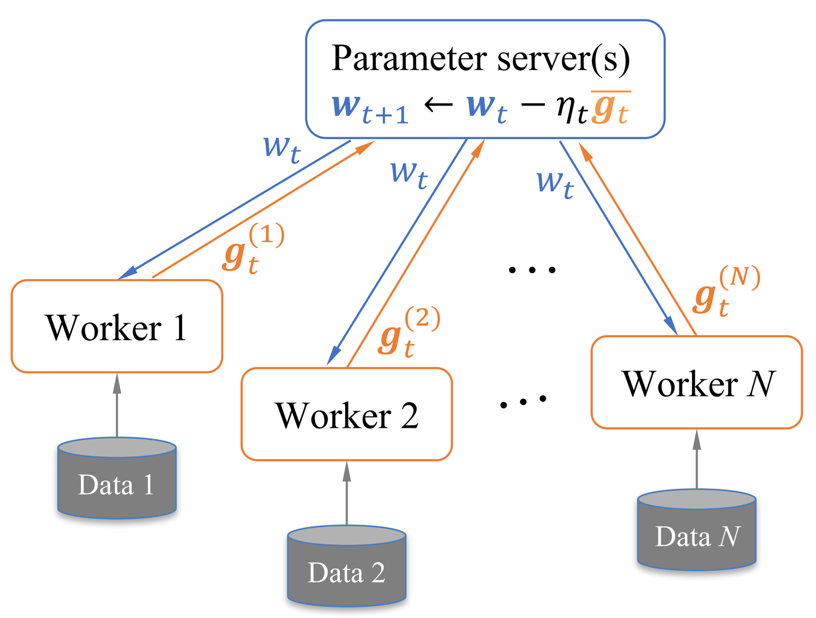
TernGrad: Ternary Gradients to Reduce Communication in Distributed Deep Learning

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]
[1]{wei.wen, chunpeng.wu, yiran.chen, hai.li}@duke.edu, [2]cong.xu@hpe.com, [3]fyan@unr.edu, [4]yaw46@pitt.edu

Background & Motivation

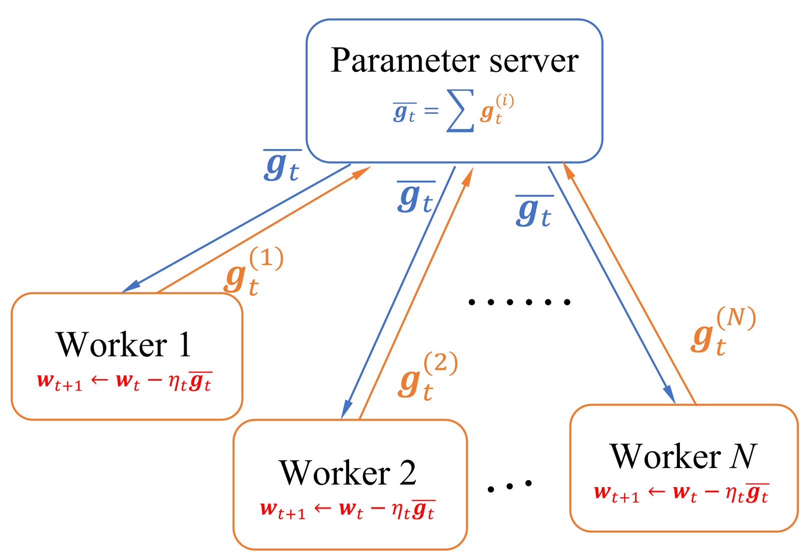
Goal

Speedup distributed training (of deep neural networks) by overcoming communication bottleneck.



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 subset
4. Synchronization in parameter server(s)

Scalability (large N):
1. Computing time decreases with N
2. Communication time becomes the bottleneck
3. This work: ternary gradients to reduce Communication



An alternative setting Synchronized Data Parallelism :
1. Only exchange gradients
2. Quantization can reduce communication in both directions
3. An identical model across workers (use the same initialization seed)

TernGrad and Convergence

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

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

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

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

$$\mathbf{E}_{\boldsymbol{z},\boldsymbol{b}}\{\tilde{\boldsymbol{g}}_t\} = \mathbf{E}_{\boldsymbol{z},\boldsymbol{b}}\{s_t \cdot sign(\boldsymbol{g}_t) \circ \boldsymbol{b}_t\}$$
$$= \mathbf{E}_{\boldsymbol{z}}\{s_t \cdot sign(\boldsymbol{g}_t) \circ \mathbf{E}_{\boldsymbol{b}}\{\boldsymbol{b}_t | \boldsymbol{z}_t\}\} = \mathbf{E}_{\boldsymbol{z}}\{\boldsymbol{g}_t\} = \nabla_{\boldsymbol{w}} C(\boldsymbol{w}_t)$$ No bias

Example:

$g_t^{(i)}$: [0.30, -1.20, ..., 0.9]

s_t: 1.20

Signs: [1, -1, ..., 1]

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

$\widetilde{g_t^{(i)}}$: [0, -1, ..., 1]*1.20

Convergence of standard SGD and *TernGrad* (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) $\mathbf{E}\{||\boldsymbol{g}||^2\} \leq A + B\,||\boldsymbol{w} - \boldsymbol{w}^*||^2$ SGD almost truly converges | $\mathbf{E}\{||\boldsymbol{g}||_\infty \cdot ||\boldsymbol{g}||_1\} \leq A + B\,||\boldsymbol{w} - \boldsymbol{w}^*||^2$ *TernGrad* almost truly converges

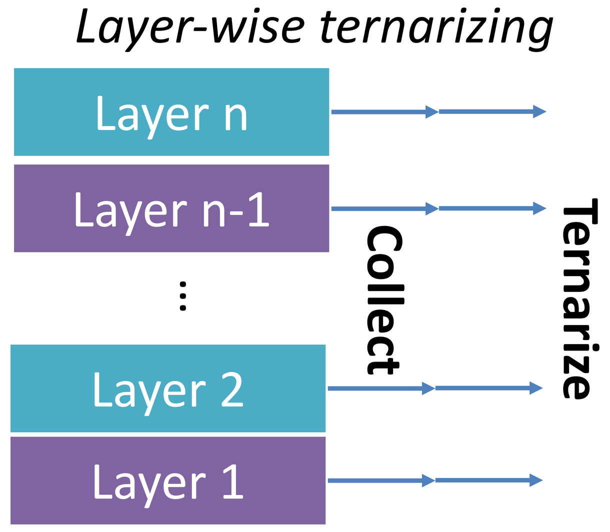
$\mathbf{E}\{||\boldsymbol{g}||^2\} \leq \mathbf{E}\{||\boldsymbol{g}||_\infty \cdot ||\boldsymbol{g}||_1\} \leq A + B\,||\boldsymbol{w} - \boldsymbol{w}^*||^2$ Stronger gradient bound in *TernGrad*

Gradient Bound and Variance

Two views to improve the convergence of *TernGrad*:
1. Variance
   - ✓ Relatively smaller $||g_t||_\infty$ can result in smaller variance
2. Gradient Bound
   - ✓ Relatively smaller $||g_t||_\infty$ pushs gradient bound assumption of *TernGrad* closer to the gradient bound in standard SGD

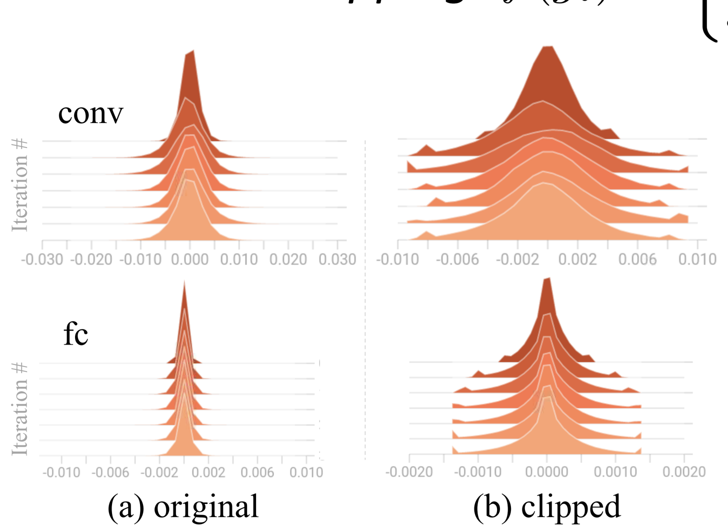
Two methods:

Layer-wise ternarizing



Extreme case: ternarizing gradient one by one is just the floating SGD

Gradient clipping $f(g_i) = \begin{cases} g_i & |g_i| \leq c\sigma \\ sign(g_i) \cdot c\sigma & |g_i| > c\sigma \end{cases}$



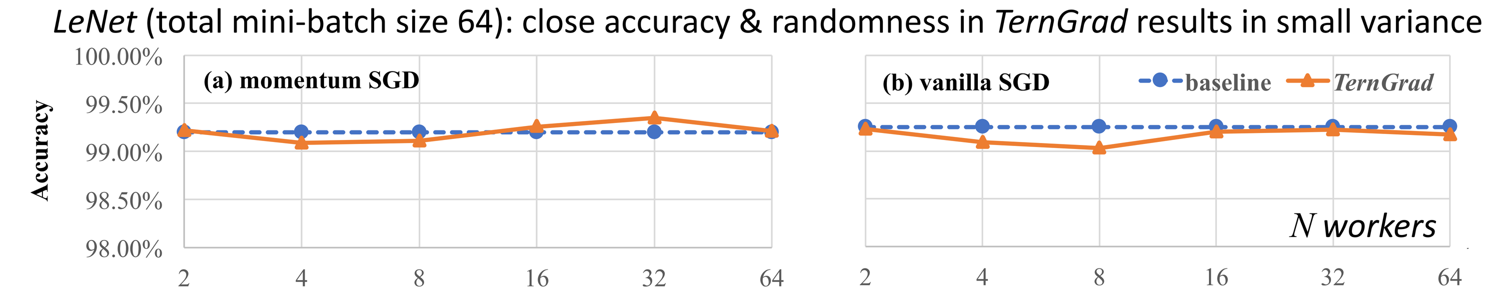
Near Gaussian distribution

c=2.5 works well for all tested datasets, DNNs and optimizers.

1. Change length 1%-1.5%
2. Change direction 2° -3°
3. Small bias with variance reduced

Experiments

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

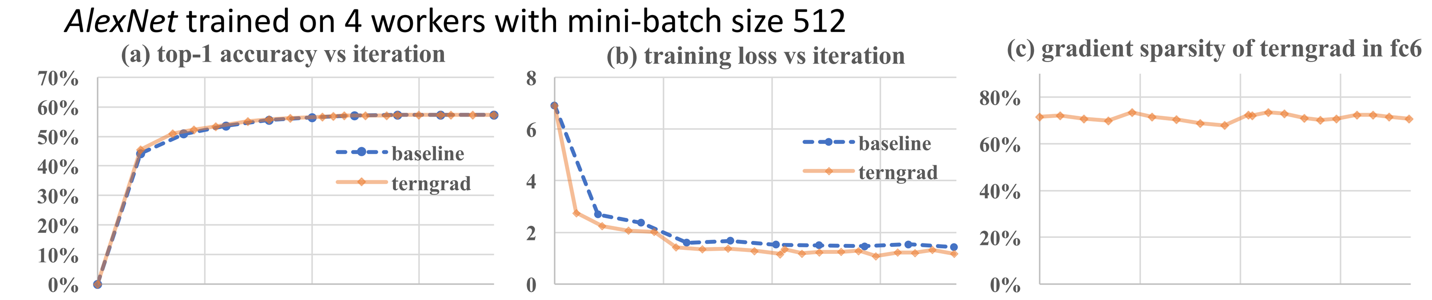


CIFAR-10, mini-batch size 64 per worker (all hyper-parameters are the same)

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

D. P. Kingma, 2014

AlexNet trained on 4 workers with mini-batch size 512



GoogLeNet (<2% accuracy loss on average)

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%

Tune hyper-parameters specifically for *TernGrad* may reduce accuracy loss

TernGrad: Randomness & regularization (in large-scale dataset like ImageNet) → (1) decrease randomness in dropout or (2) use smaller weight decay

No new hyper-parameters added

AlexNet

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.

A performance model to evaluate the speedup