COMPRESSED METHODS

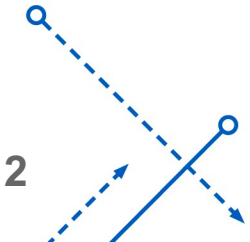
Error Feedback Fixes SignSGD and other Gradient Compression Schemes

Naman Tejaswi

University at Buffalo
School of Engineering and Applied Sciences

What Exactly gradient compression methods and why should we care about it ?

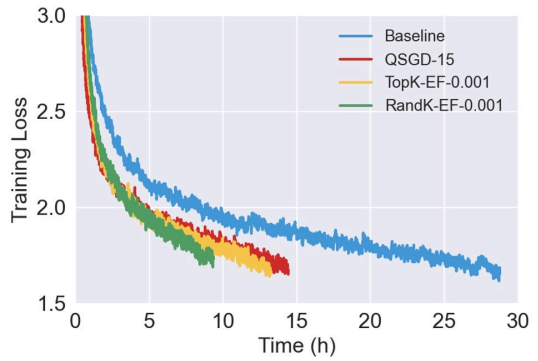
- Gradient Compression reduces bandwidth of the information which is to be backpropagated and it can make training more scalable and efficient without significant loss either in convergence rate or accuracy.
- The key word being significant of course there is no free lunch and scalable training with gradient compression does come at some trade off
- Bigger the model and more the number of fully connected components more is the speedup with gradient compression. Thus, practical models often take advantage of gradient compression
- Gradient Compression opens the door for distributed deep learning!



How does it work though?

- Gradient compression uses the approach of delaying the synchronization of weight updates which are small. Although small weight updates might not be sent for that batch, this information is not discarded.
- Once the weight updates for this location accumulate to become a larger value, they will be propagated. Since there is no information loss, but only delayed updates, it does not lead to a significant loss in accuracy or convergence rate.
- The graph shows how a popular gradient compression scheme GRACE performs in comparison to the baseline non compressed training.
- The greatest benefits from gradient compression are realized when using multi-node (single or multi-GPU) distributed training.
- Analogus to amdahls law if the latency of communication between the different nodes and sequential tasks are too much then we are better off using another method

Acceleration of BERT base pre-training (phase1) on 32 GPUs with GRACE
(4 nodes, each with 8 V100 GPUs, with 25Gbps RDMA connection)



EF: Error Feedback (also known as Residual Memory). 0.001 means 0.1% compression rate.
QSGD-15 means 15 states, or 4-bit quantization equivalently.

Signed Gradient Descent

- To minimize a continuous function which could be non covex, the classic stochastic gradient descent performs the iteration as
- $x := x_t - \gamma g_t$ where gamma is the learning rate and g is the stochastic gradient
- Now methods which perform gradient update only based on the sign of the gradient at each step have recently gained popularity for training deep learning models.
- $x := x_t - \gamma . (sign) g_t$
- This is commonly known as signSGD, signed stochastic gradient
- signSGD can be thought of as approximations of adaptive gradient methods such as ADAM in addition to being effective for compressed communication method

What lacks in Signed Gradient Descent

- For starters, they do not converge in general and can take forever to escape from a saddle point !
- Why even then bother to use signsgd at all ?
- Training large neural networks requires distributing learning across multiple workers, where the cost of communicating gradients can be a significant bottleneck. signSGD alleviates this problem by transmitting just the sign of each minibatch stochastic gradient. We prove that it can get the best of both worlds: compressed gradients and SGD-level convergence rate. The relative ℓ1/ℓ2 geometry of gradients, noise and curvature informs whether signSGD or SGD is theoretically better suited to a particular problem.
- In certain problems even finding a local optima is challenging and finding even a local optima would suffice for time being.

Why does signSGD fail and how do we fix it?s

- The reason why vanilla signSGD does not converge is it loses out on two critical inputs the magnitude and the direction of the gradient.
- signSGD fails to converge even simple non convex function however it has been shown that with optimization it can perform comparable to sgd.
- After using distributive training my majority vote algorithm and the classical optimization theory where local information in gradient tells you about global information about the direction to the minima.
- Sign based momentum technique signum has also helped
- And if the problem is nonconvex, you are in luck as the performance is better with signsgd

signSGD with error feedback

- We scale the signed vector by the norm of the gradient to ensure the magnitude of the gradient is not forgotten.
- We locally store the difference between the actual and compressed gradient
- Finally, we add it back into the next step so that the correct direction is not forgotten
- This is called EF-SIGNSGD

Algorithm 1 EF-SIGNSGD (SIGNSGD with Error-Feedb.)

Input: learning rate γ, initial iterate $\mathbf{x}_0 \in \mathbb{R}^d$, $\mathbf{e}_0 = \mathbf{0}$
for $t = 0, \ldots, T-1$ **do**
  $\mathbf{g}_t := \text{stochasticGradient}(\mathbf{x}_t)$
  $\mathbf{p}_t := \gamma \mathbf{g}_t + \mathbf{e}_t$ ▷ *error correction*
  $\Delta_t := (\|\mathbf{p}_t\|_1 / d)\,\text{sign}(\mathbf{p}_t)$ ▷ *compression*
  $\mathbf{x}_{t+1} := \mathbf{x}_t - \Delta_t$ ▷ *update iterate*
  $\mathbf{e}_{t+1} := \mathbf{p}_t - \Delta_t$ ▷ *update residual error*
end for

signSGD with error feedback

- et denotes the accumulated error from all quantization/compression steps.
- This residual error is added to the gradient step gt to obtain the corrected direction pt.
- When compressing pt, the signed vector is again scaled by kptk1 and hence does not lose information about the magnitude.

Algorithm 1 EF-SIGNSGD (SIGNSGD with Error-Feedb.)

Input: learning rate γ, initial iterate $\mathbf{x}_0 \in \mathbb{R}^d$, $\mathbf{e}_0 = \mathbf{0}$
for $t = 0, \ldots, T-1$ **do**
  $\mathbf{g}_t := \text{stochasticGradient}(\mathbf{x}_t)$
  $\mathbf{p}_t := \gamma \mathbf{g}_t + \mathbf{e}_t$ ▷ *error correction*
  $\Delta_t := (\|\mathbf{p}_t\|_1 / d)\, \text{sign}(\mathbf{p}_t)$ ▷ *compression*
  $\mathbf{x}_{t+1} := \mathbf{x}_t - \Delta_t$ ▷ *update iterate*
  $\mathbf{e}_{t+1} := \mathbf{p}_t - \Delta_t$ ▷ *update residual error*
end for

Performance increase of signSGD with error feedback

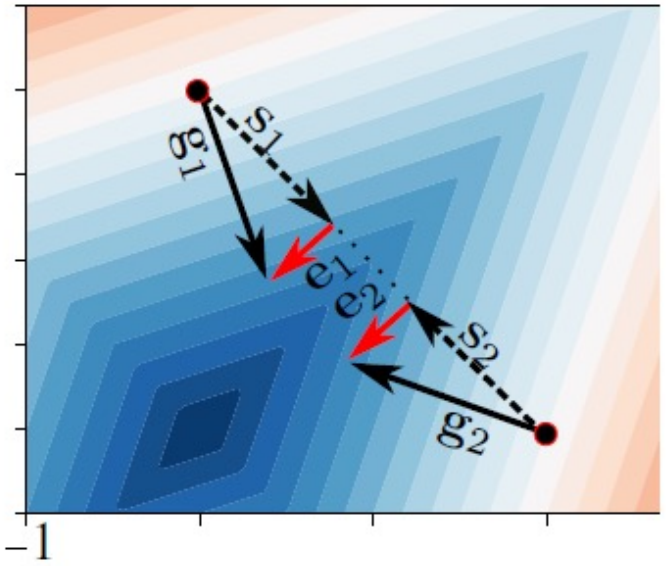
- By incorporating error-feedback, SIGNSGD always converge
- For non-convex smooth functions recovers the same rate as SGD, which means we get gradient compression for free*
- Unlike SIGNSGD, EF-SIGNSGD converges to the max-margin solution in over-parameterized least-squares EF-SIGNSGD generalizes much better than SIGNSGD.

Non convergence SignSGD on non smooth convex problems

- The sign operator loses track of the magnitude of the stochastic gradient. And the noise is bimodal.
- Consider the non smooth convex function with optima at 0,0

$$\min_{\mathbf{x}\in\mathbb{R}^2} \left[f(\mathbf{x}) := \epsilon|x_1 + x_2| + |x_1 - x_2| \right]$$

- Here we iterate from at x0 = (1; 1)> lie along the line x1 + x2 = 2.
- for any x such that x1 + x2 > 0, sign(g(x))=+-(1,-1)^T>, and hence x1 + x2 remains constant among the iteration.
- The gradients g (in solid black), signed gradient direction s = sign(g) (in dashed black), and the error e (in red) are plotted for ε= 0:5. SIGNSGD moves only along s = +-(1;-1) while the error e is ignored.
- The lack of direction towards the optima is something which error feedback would help.



Proof of convergence of error feedback signsgd

Assumption A (Compressor). *An operator* $\mathcal{C} : \mathbb{R}^d \to \mathbb{R}^d$ *is a* δ*-approximate compressor over* $\mathcal{Q}$ *for* $\delta \in (0, 1]$ *if*

$$\|\mathcal{C}(\mathbf{x}) - \mathbf{x}\|_2^2 \leq (1 - \delta)\|\mathbf{x}\|_2^2, \ \forall \mathbf{x} \in \mathcal{Q}\,.$$

- Note that δ= 1 implies that C(x) = x.
- The key is we now propose a lemma which shows that the residual errors maintained does not amount to much.

Lemma 3 (Error is bounded). *Assume that* $\mathbb{E}[\|\mathbf{g}_t\|^2] \leq \sigma^2$ *for all* $t \geq 0$. *Then at any iteration* t *of* EF-SGD, *the norm of the error* $\mathbf{e}_t$ *in Algorithm 2 is bounded:*

$$\mathbb{E}\|\mathbf{e}_t\|_2^2 \leq \frac{4(1-\delta)\gamma^2\sigma^2}{\delta^2}\,, \qquad \forall t \geq 0\,.$$

If $\delta = 1$, then $\|\mathbf{e}_t\| = 0$ and the error is zero as expected.

We employ standard assumptions of smoothness of the loss function and the variance of the stochastic gradient.

Assumptions for the lemma and deduced theorem

Assumption B (Smoothness). *A function* $f : \mathbb{R}^d \to \mathbb{R}$ *is* L*-smooth if for all* $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$ *the following holds:*

$$|f(\mathbf{y}) - (f(\mathbf{x}) + \langle \nabla f(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle)| \leq \frac{L}{2} \|\mathbf{y} - \mathbf{x}\|_2^2 .$$

Assumption C (Moment bound). *For any* $\mathbf{x}$*, our query for a stochastic gradient returns* $\mathbf{g}$ *such that*

$$\mathbb{E}[\mathbf{g}] = \nabla f(\mathbf{x}) \quad \textit{and} \quad \mathbb{E}\|\mathbf{g}\|_2^2 \leq \sigma^2 .$$

Given these assumptions, we can formally state our theorem followed by a sketch of the proof.

Theorem II (Non-convex convergence of EF-SGD). *Let* $\{\mathbf{x}_t\}_{t \geq 0}$ *denote the iterates of Algorithm 2 for any step-size* $\gamma > 0$*. Under Assumptions A, B, and C,*

$$\min_{t \in [T]} \mathbb{E}[\|\nabla f(\mathbf{x}_t)\|^2] \leq \frac{2 f_0}{\gamma (T+1)} + \frac{\gamma L \sigma^2}{2} + \frac{4 \gamma^2 L^2 \sigma^2 (1 - \delta)}{\delta^2} ,$$

with $f_0 := f(\mathbf{x}_0) - f^\star$.

Comparison of rate of convergence with SGD

Remark 4. *If we substitute* $\gamma := \frac{1}{\sqrt{T+1}}$ *in Theorem II, we get*

$$\min_{t\in[T]} \mathbb{E}[\|\nabla f(\mathbf{x}_t)\|^2] \leq \frac{4(f(\mathbf{x}_0)-f^\star)+L\sigma^2}{2\sqrt{T+1}} + \frac{4L^2\sigma^2(1-\delta)}{\delta^2(T+1)}$$

In the above rate, the compression factor δ *only appears in the higher order* $\mathcal{O}(1/T)$ *term. For comparison, SGD under the exact same assumptions achieves*

$$\min_{t\in[T]} \mathbb{E}[\|\nabla f(\mathbf{x}_t)\|^2] \leq \frac{2(f(\mathbf{x}_0)-f^\star)+L\sigma^2}{2\sqrt{T+1}}\,.$$

This means that after $T \geq \mathcal{O}(1/\delta^2)$ *iterations the second term becomes negligible and the rate of convergence catches up with full SGD—this is usually true after just the first few epochs. Thus we prove that compressing the gradient does not change the asymptotic rate of SGD.*[1]

Remark 5. *The use of error-feedback was motivated by our counter-examples for biased compression schemes. However our rates show that even if using an unbiased compression (e.g. QSGD [Alistarh et al., 2017]), using error-feedback gives significantly better rates. Suppose we are given an unbiased compressor* $cU(\cdot)$ *such that* $\mathbb{E}[\mathcal{U}(\mathbf{x})] = \mathbf{x}$ *and* $\mathbb{E}\left[\|\mathcal{U}(\mathbf{x})\|_2^2\right] \leq k\|\mathbf{x}\|^2$*. Then without feedback, using standard analysis (e.g. [Alistarh et al., 2017]) the algorithm converges* k *times slower:*

$$\min_{t\in[T]} \mathbb{E}[\|\nabla f(\mathbf{x}_t)\|^2] \leq \frac{(f(\mathbf{x}_0)-f^\star)+Lk\sigma^2}{2\sqrt{T+1}}\,.$$

Instead, if we use $\mathcal{C}(\mathbf{x}) = \frac{1}{k}\mathcal{U}(\mathbf{x})$ *with error-feedback, we would achieve*

$$\min_{t\in[T]} \mathbb{E}[\|\nabla f(\mathbf{x}_t)\|^2] \leq \frac{4(f(\mathbf{x}_0)-f^\star)+L\sigma^2}{2\sqrt{T+1}} + \frac{2L^2\sigma^2k^2}{T+1}\,,$$

thereby pushing the dependence on k *into the higher order* $\mathcal{O}(1/T)$ *term.*

Our counter-examples showed that biased compressors may not converge for non-smooth functions. Below we prove that adding error-feedback ensures convergence under standard assumptions even for non-smooth functions.

Comparison of sgd, signsgd and signsgd with ef

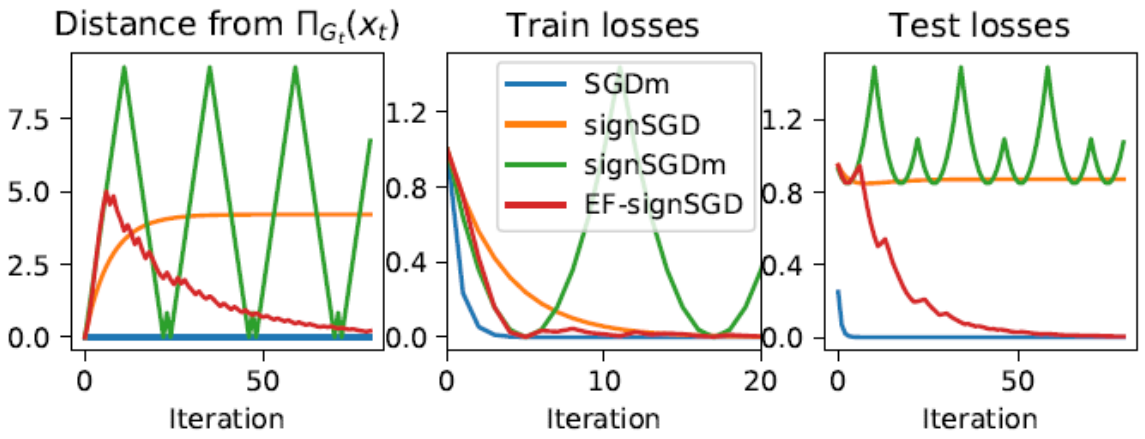


Figure 3: Left shows the distance of the iterate from the linear span of the gradients $\|\mathbf{x}_t - \Pi_{\mathbf{G}_t}(\mathbf{x}_t)\|$. The middle and the right plots show the train and test loss. SIGNSGD and SIGNSGDM have a high distance to the span, and do not generalize (test loss is higher than 0.8). Distance of EF-SIGNSGD to the linear span (and the test loss) goes to 0.

Experimental Results

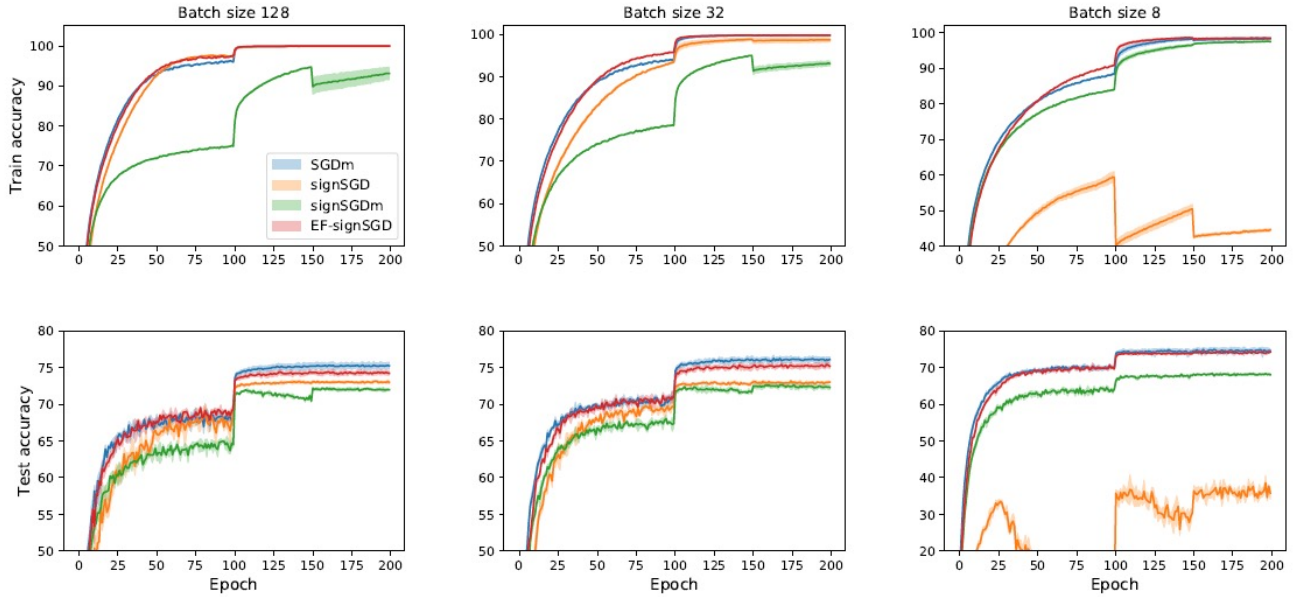


Figure 4: Experimental results showing the train and test accuracy percentages on CIFAR-100 using Resnet18 for different batch-sizes. The solid curves represent the mean value and shaded region spans one standard deviation obtained over three replications. Note that the scale of the y-axis varies across the plots. EF-SIGNSGD consistently and significantly outperforms the other sign-based methods, closely matching the performance of SGDM.

Experimental Results

Figure 4: Experimental results showing the train and test accuracy percentages on CIFAR-100 using Resnet18 for different batch-sizes. The solid curves represent the mean value and shaded region spans one standard deviation obtained over three replications. Note that the scale of the y-axis varies across the plots. EF-SIGNSGD consistently and significantly outperforms the other sign-based methods, closely matching the performance of SGDM.

Batch-size	SGDM	SIGNSGD	SIGNSGDM	EF-SIGNSGD
128	75.35	-2.21	-3.15	**-0.92**
32	76.22	-3.04	-3.57	**-0.79**
8	74.91	-36.35	-6.6	**-0.64**

Conclusion

- EF-SIGNSGD is faster than SGDM on train
- EF-SIGNSGD almost matches SGDM on test
- SIGNSGD performs poorly for small batch-sizes.
- The performance of SIGNSGD is always worse than EF-SIGNSGD indicating that scaling is insufficient, and that error-feedback is crucial for performance.
- Further all metrics (train and test, loss and accuracy) increasingly become worse as the batch- 11 size decreases indicating that SIGNSGD is indeed a brittle algorithm. In fact for batch-size 8, the algorithm becomes extremely unstable.
- If we are using signsgd we are always better off using signsgd with error feedback however sgd at the expense of time can be better than signsgd with error feedback i.e once again no free lunch.

References

- **Error Feedback Fixes SignSGD and other Gradient Compression Schemes Sai Praneeth Karimireddy, Quentin Rebjock, Sebastian U. Stich, Martin Jaggi**
- **https://mxnet.apache.org/versions/1.9.1/api/faq/gradient_compression.htm**
- **Léon Bottou. Large-scale machine learning with stochastic gradient descent.**
- **Bernstein, Jeremy, et al. signSGD: Compressed optimisation for non-convex problems. ICML 2018**