Exploitation of temporal structure in momentum-SGD for gradient compression

Tharindu Adikari, Stark Draper

University of Toronto

ISTC - August 2021

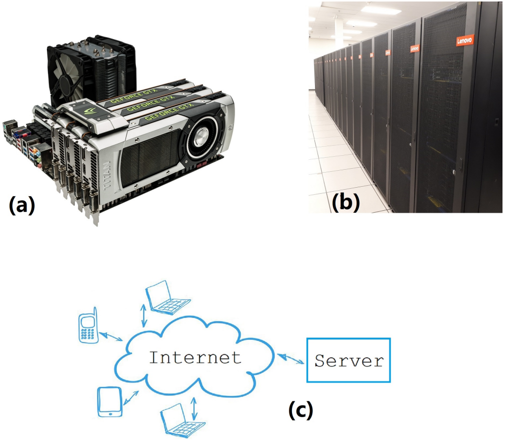
Motivation: Data exchange volumes can be massive in modern AI workloads

Problem background:

- Large datasets / parameterized models
- Decentralize data, synchronize computation
- Multi-GPU / data centres / edge devices
- Limited communication bandwidth

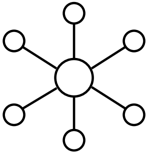
Motivating example:

- BERT benchmark model
- 340 million parameters
- Optimize with distributed SGD
- 1.3GB per gradient (32-bit floating-point)



Extended paper of our work: Tharindu B. Adikari and Stark C. Draper. "Compressing gradients by exploiting temporal correlation in momentum-SGD". In: *IEEE J. Select. Areas Inform. Theory* 2.3 (2021). DOI: 10.1109/JSAIT.2021.3103494

Setup: We start with a standard model of distributed optimization, "SGD"

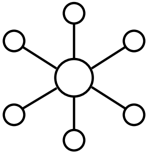


Master-worker

- Partition large dataset amongst n-workers
- Worker-i computes g_t^i from its data, a stochastic gradient
- Compute smoothed gradient $v_t^i = \beta v_{t-1}^i + (1-\beta) g_t^i$
- Send v_t^i to master and receive $\frac{1}{n}\sum_{i=1}^n v_t^i$ from master
- Workers update $w_{t+1} = w_t - \eta_t \frac{1}{n}\sum_{i=1}^n v_t^i$

- $\beta = 0$: SGD (stochastic gradient descent)
- $\beta \neq 0$: "momentum"-SGD

Setup: We start with a standard model of distributed optimization, "SGD"



Master-worker

- Partition large dataset amongst n-workers
- Worker-i computes g_t^i from its data, a stochastic gradient
- Compute smoothed gradient $v_t^i = \beta v_{t-1}^i + (1-\beta) g_t^i$
- Send v_t^i to master and receive $\frac{1}{n}\sum_{i=1}^n v_t^i$ from master
- Workers update $w_{t+1} = w_t - \eta_t \frac{1}{n}\sum_{i=1}^n v_t^i$

- $\beta = 0$: SGD (stochastic gradient descent)
- $\beta \neq 0$: "momentum"-SGD

Compress v_t^i with $Q : \mathbb{R}^d \to \mathbb{R}^d$

- x: input to encoder
- $Q(x)$: output of decoder
- $Q(x)$ takes smaller # bits than x

- Iterate $w_{t+1} = w_t - \eta_t \frac{1}{n}\sum_{i=1}^n Q(v_t^i)$

Examples for $Q(x)$:

- quantize components in x (e.g. "Scaled-sign")
- sparsify vector x (e.g. "Top-K")

Gradient compression: We aim to exploit "temporal" dependencies across iterations

Basic problem:

- Goal: compress $\ldots, v_{t-2}^i, v_{t-1}^i, v_t^i$ in each iteration
- Q trades off bit-rate for fidelity, "lossy compression"
- If entries in v_t^i are related can exploit *within*-vector structure to further reduce the bit-rate. This is a type of spatial correlation, e.g., Gradiveq[1] does this

Our target:

- Design a compression scheme to exploit correlations *across*-vectors structure (v_{t-1}^i and v_t^i), i.e., temporal correlations

Our idea:

- Analogy: image (JPEG) vs video (MPEG)
- Updates between iterations may be correlated, i.e., between g_{t-1}^i and g_t^i
- Especially true when using momentum, $\beta \neq 0$

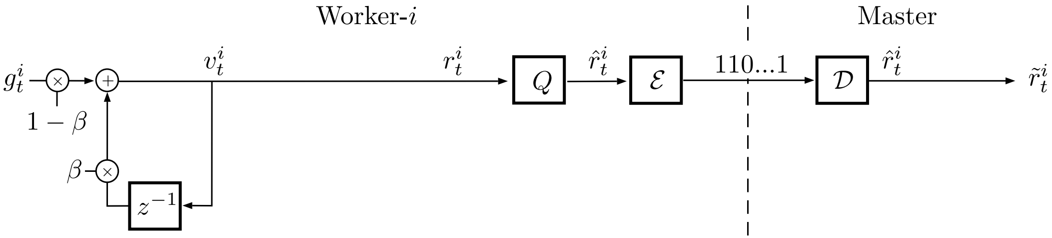
$$v_t^i = \beta v_{t-1}^i + (1-\beta) g_t^i$$

$$w_{t+1} = w_t - \eta_t \frac{1}{n} \sum_{i=1}^{n} Q(v_t^i)$$

- In practice β is in range of 0.9 to 0.99
- Momentum: components of v_t^i change slowly

[1] Mingchao Yu et al. "Gradiveq: Vector quantization for bandwidth-efficient gradient aggregation in distributed cnn training". In: *Advances in Neural Inf. Proc. Sys.* Montréal, 2018

Proposing our Q-diff (quantized-differential) algorithm



- $r_t^i = v_t^i = \beta v_{t-1}^i + (1-\beta) g_t^i$
- Quantize r_{t-1}^i and r_t^i with $\mathcal{Q}$ to produce $\hat{r}_{t-1}^i$ and $\hat{r}_t^i$
- r_{t-1}^i and $r_t^i \implies$ continuous alphabet, $\hat{r}_{t-1}^i$ and $\hat{r}_t^i \implies$ finite alphabet
- Implement a differential encoder in $\mathcal{E}$
- Encode $\hat{r}_t^i$ conditioned on knowledge of $\hat{r}_{t-1}^i$
- In our experiments employ a Lloyd-Max quantizer for $\mathcal{Q}$

Empirical evaluation of Q-diff: large savings vs. competing algorithms

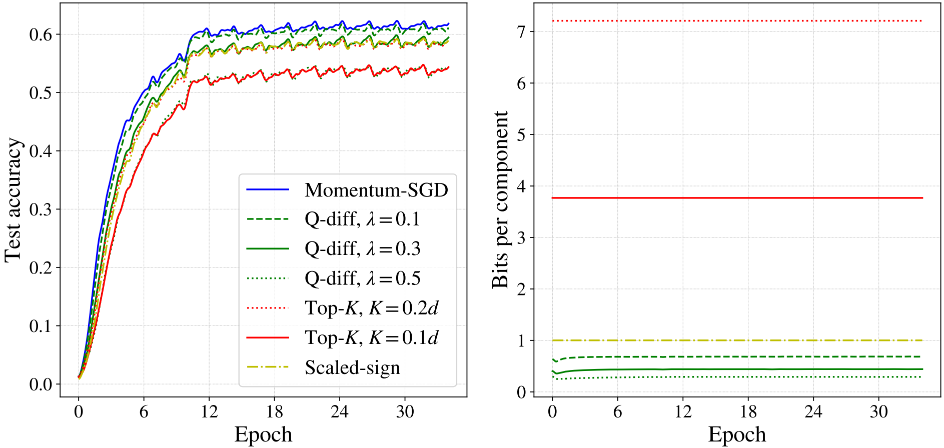


Figure: λ: hyper-parameter in Q-diff that controls compression ratio. $\beta = 0.99$ for momentum.

Empirical evaluation of Q-diff: zoom in on "quantized Top-K" vs. our "Q-diff" alg

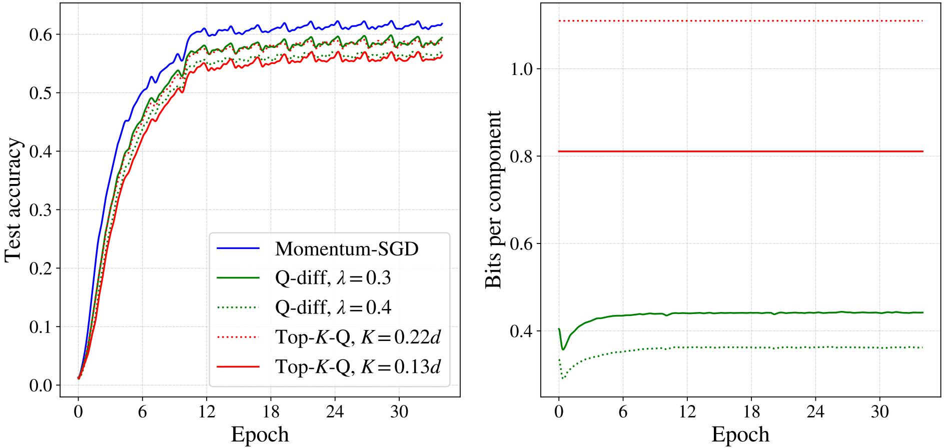
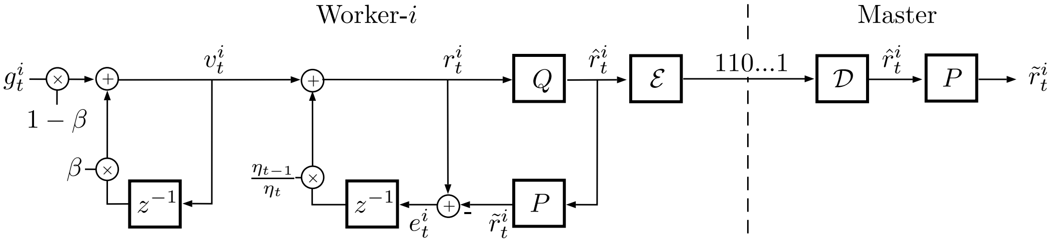


Figure: λ: hyper-parameter in Q-diff that controls compression ratio. $\beta = 0.99$ for momentum.

Incorporate 'error-feedback' to improve convergence of the algorithm



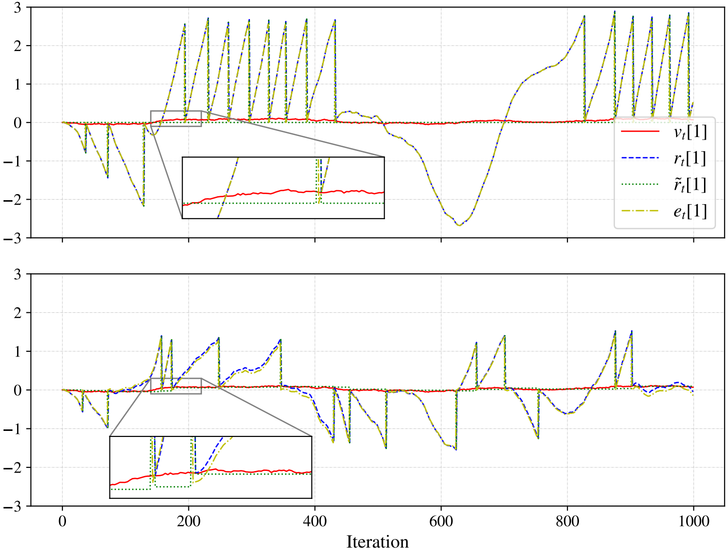
- r_{t-1}^i and r_t^i are affected due to error-feedback path
- Differential encoding is no longer suitable for $\hat{r}_{t-1}^i$ and $\hat{r}_t^i$
- Introduce the new algorithm Est-K that builds on top of Top-K
- Top-K: send only the largest K elements in the vector
- Role of P is to predict values in reconstruction vector r_t^i
- Useful prediction possible due to the temporal correlation that exists from one iteration to the next
- Smaller prediction error means easier to correct and reduces bit rate

Prediction in Est-K reduces the dynamic range of the error versus Top-K

- **Top**-K (upper figure)
- **Est**-K (lower figure)

- Synthetic experiment with one worker
- Plot first coordinate in each vector $v_t, r_t, \hat{r}_t, e_t$
- $v_t[1]$ changes slowly
- Master applies $\tilde{r}_t[1]$

- Top-K applies zero in most iterations
- Est-K applies an estimated value
- Est-K incurs a lower magnitude in e_t



Empirical evaluation of Est-K (EF closed)

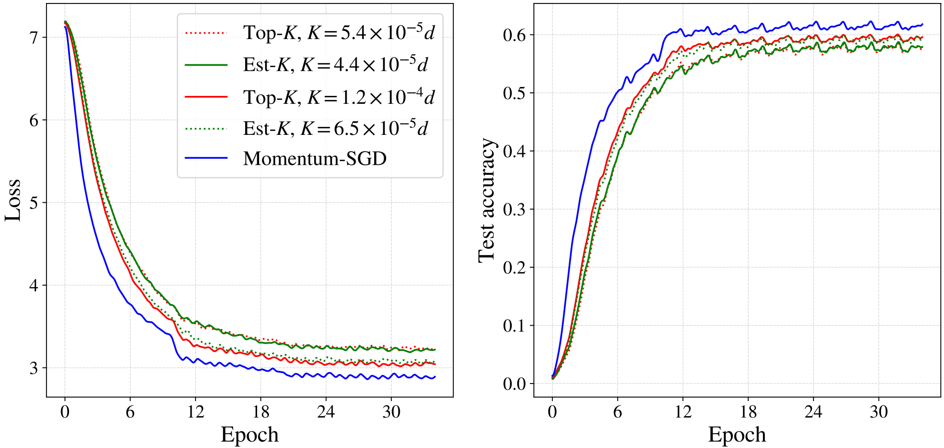


Figure: Comparing the performance of Est-K with Top-K. All algorithms employ momentum with $\beta = 0.99$ parameter. Est-K and Top-K employ error-feedback. From top to bottom in legend the algorithms incur 0.0026, 0.0021, 0.0056, 0.0031, and 32 bits per component.

Summary and next steps

- We exploit extant temporal correlation in update vectors in compression.
- Easy to design an algorithm when error-feedback is not used (Q-diff).
- When error-feedback is used, we design an algorithm based on Top-K quantizer (Est-K).
- Our two algorithms outperforms algorithms that do not exploit temporal correlation.

- Note that we do not use very advanced temporal compression in proposed algorithms.
- Q-diff only implements a first order differential encoder (differences between the current and last iteration), and Est-K implements only a constant estimator (time average of momentum between two updates).
- More advanced predictors should perform even better.

Thank you.

Extended paper of our work: Tharindu B. Adikari and Stark C. Draper. "Compressing gradients by exploiting temporal correlation in momentum-SGD". In: *IEEE J. Select. Areas Inform. Theory* 2.3 (2021). DOI: 10.1109/JSAIT.2021.3103494