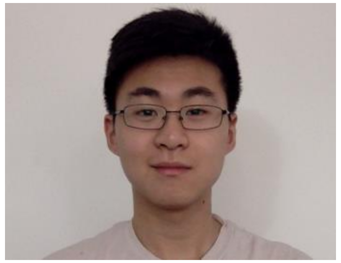
Quantized Stochastic Gradient Descent: Communication vs. Convergence

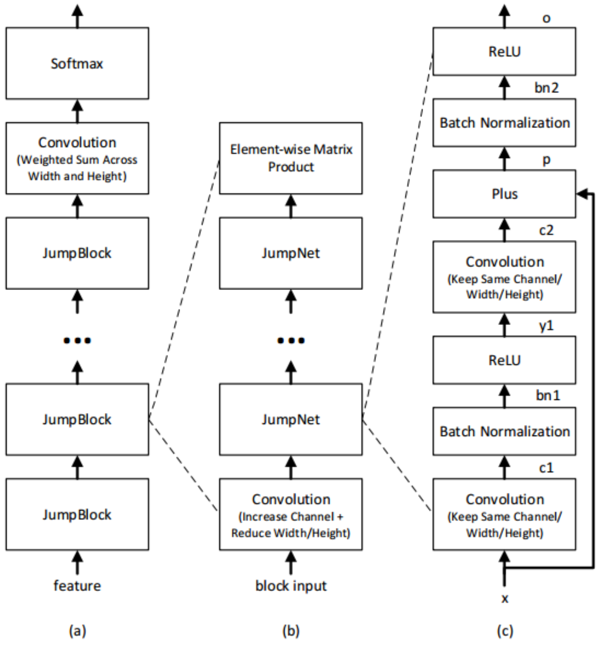
Dan Alistarh[1], Jerry Li[2,3], Ryota Tomioka[2], Milan Vojnovic[2]

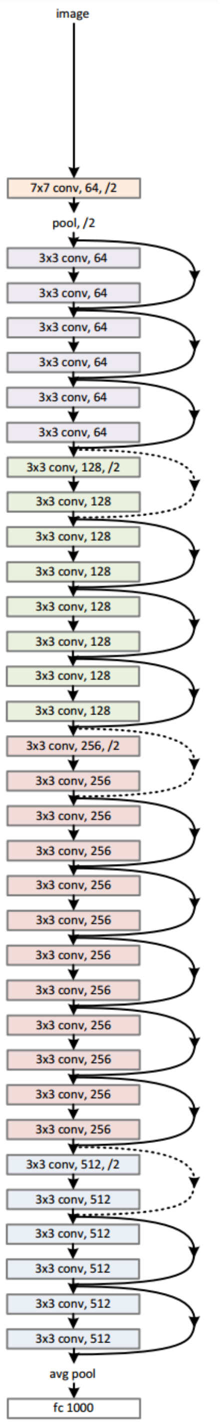
[1]ETH, [2]Microsoft Research, [3]MIT


Deep models: how to train them efficiently?

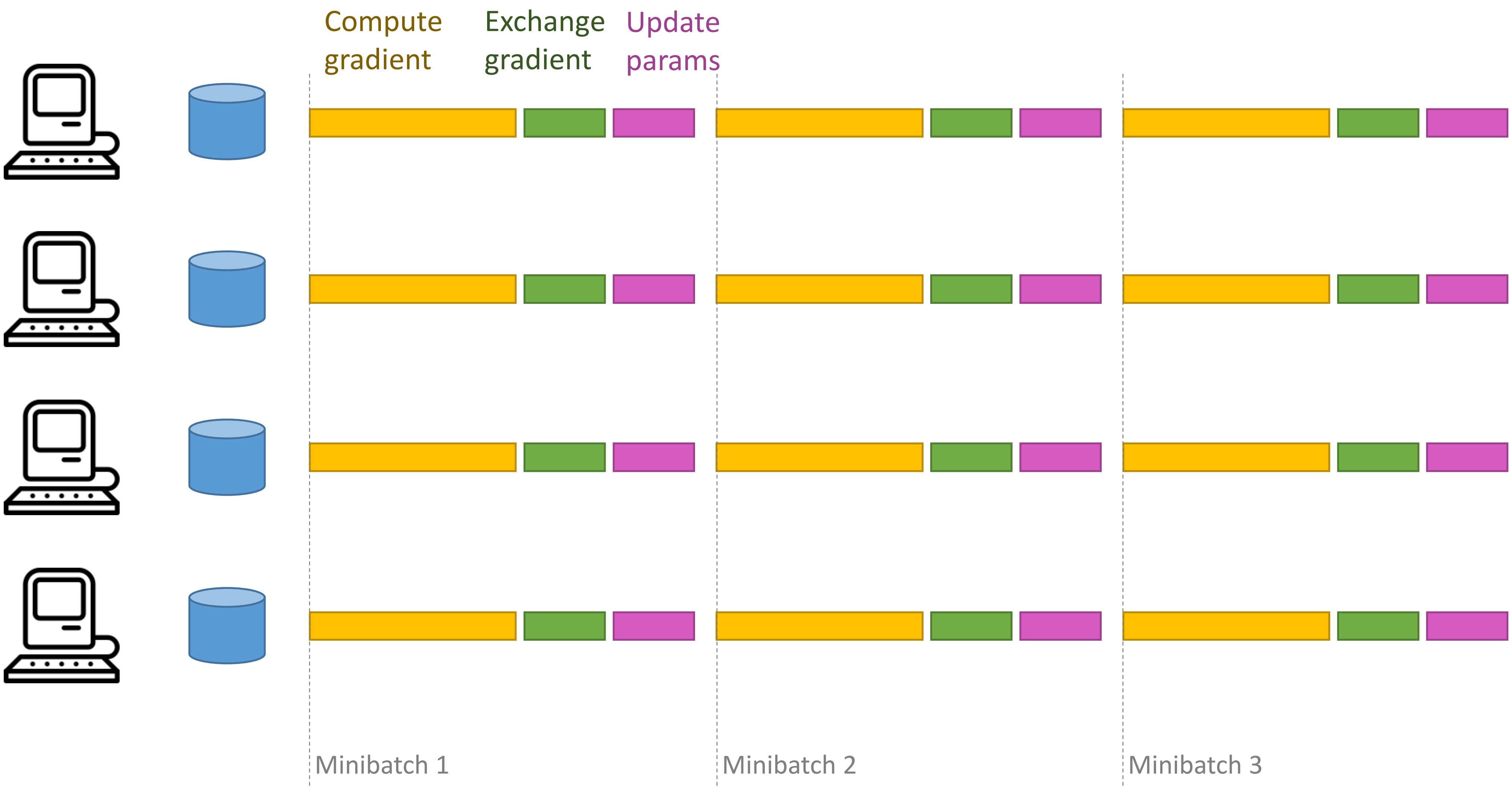
- Vision
  - ImageNet: 1.6 million images
  - ResNet-152 [He+15]: 152 layers, 60 million parameters
- Speech
  - NIST2000 Switchboard dataset: 2000 hours
  - LACEA [Yu+16]: 22 layers, 65 million parameters (w/o language model)



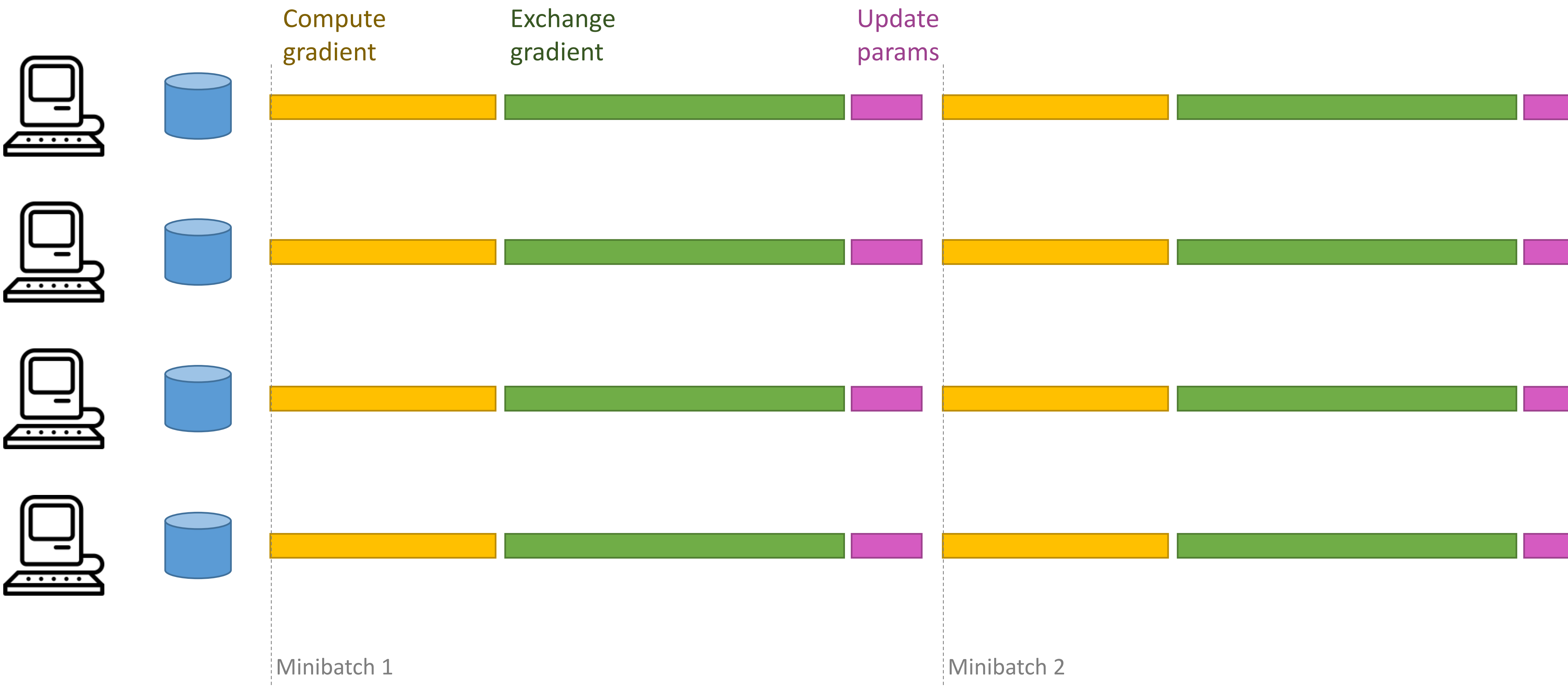


He et al. (2015) "Deep Residual Learning for Image Recognition"
Yu et al. (2016) "Deep convolutional neural networks with layer-wise context expansion and attention"

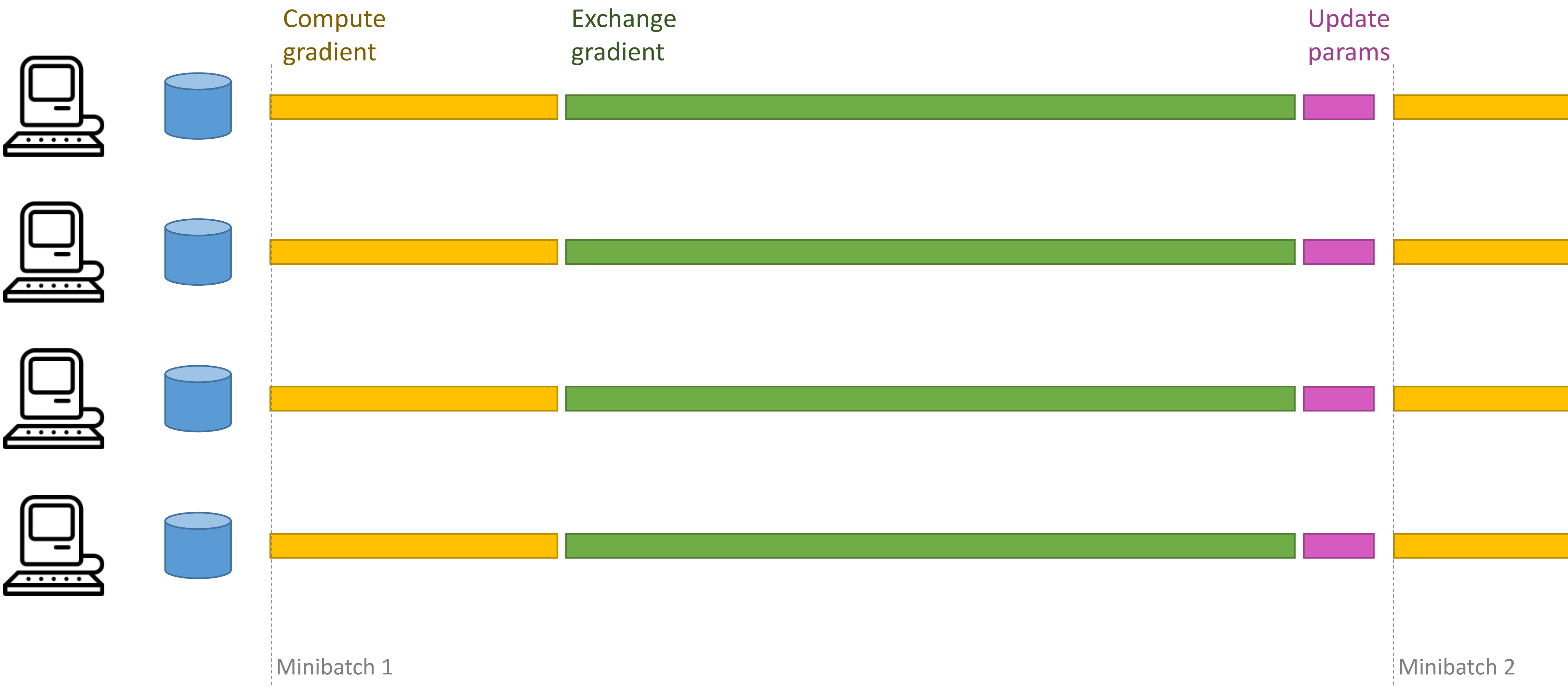
Data parallel SGD



Data parallel SGD (bigger model)



Data parallel SGD (biggggger model)



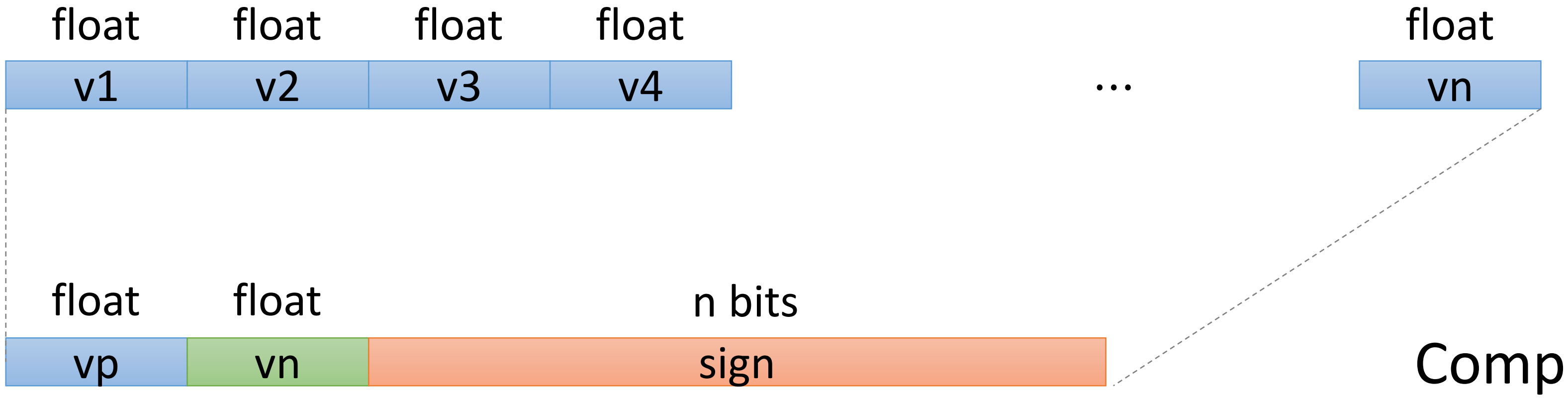
If we could *compress* the gradients…

Inspiration: 1-bit SGD [Seide et al 2014]

- Quantization function

$$Q_i[v] = \begin{cases} \bar{v}_p & \text{if } v_i \geq 0, \\ \bar{v}_n & \text{otherwise} \end{cases}$$

where $\boxed{\bar{v}_p = \text{mean}([v_i \text{ for } i : v_i \geq 0])}$, $\boxed{\bar{v}_n = \text{mean}([v_i \text{ for } i : v_i < 0])}$



Compression rate ≈ 32x

Unfortunately, no theoretical justification!

Seide et al (2014) "1-Bit Stochastic Gradient Descent and its Application to Data-Parallel Distributed Training of Speech DNNs"

Our contribution

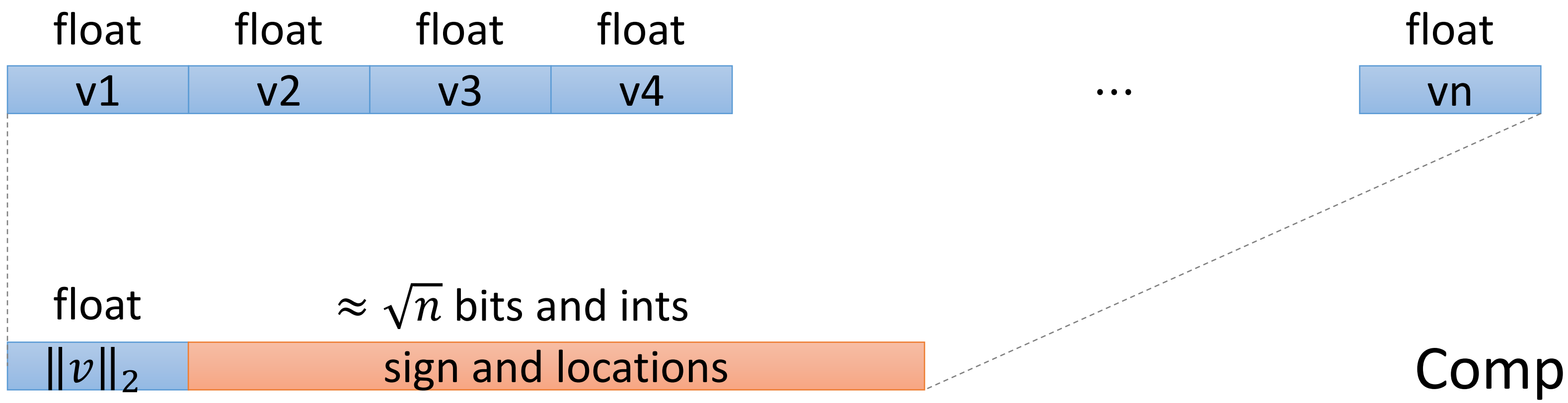
- We propose a (family of) new quantization function
  - Unbiased stochastic gradient
  - Allows super-constant $\tilde{O}(\sqrt{n})$ compression rate ☺
  - Convergence guarantee in $\tilde{O}(\sqrt{n})$ more steps ☹
  - Hyper-parameters control trade-off between convergence and compression
- We empirically show that the convergence does not slow-down too much

A simple randomized quantization function

- Quantization function

$$Q_i[v] = \|v\|_2 \cdot \mathrm{sgn}(v_i) \cdot \xi_i(v)$$

where $\xi_i(v) = 1$ with probability $|v_i|/\|v\|_2$ and 0 otherwise.



Compression rate $\approx \sqrt{n}$

Note that $E[\sum_i \xi_i(v)] \leq \|v\|_1/\|v\|_2 \leq \sqrt{n}$

Properties of the proposed quantization function

- Quantization function

$$Q_i[v] = \|v\|_2 \cdot \mathrm{sgn}(v_i) \cdot \xi_i(v)$$

where $\xi_i(v) = 1$ with probability $|v_i|/\|v\|_2$ and 0 otherwise.

1. Sparsity:

$$E\left[\sum_i \xi_i(v)\right] \leq \|v\|_1/\|v\|_2 \leq \sqrt{n}$$

2. Unbiasedness:

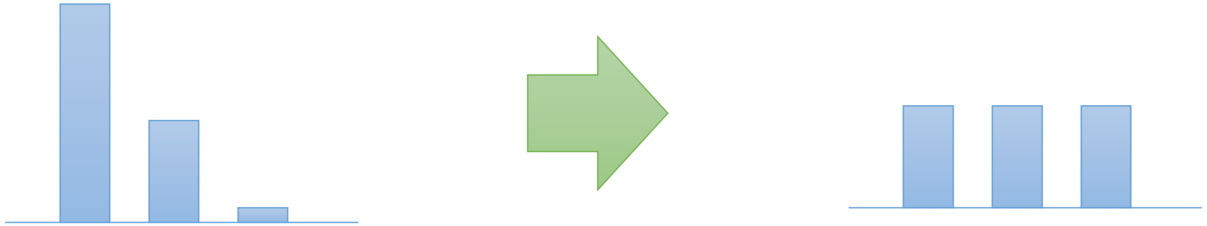
$$E\left[Q_i[v]\right] = v_i$$

3. Second moment bound:

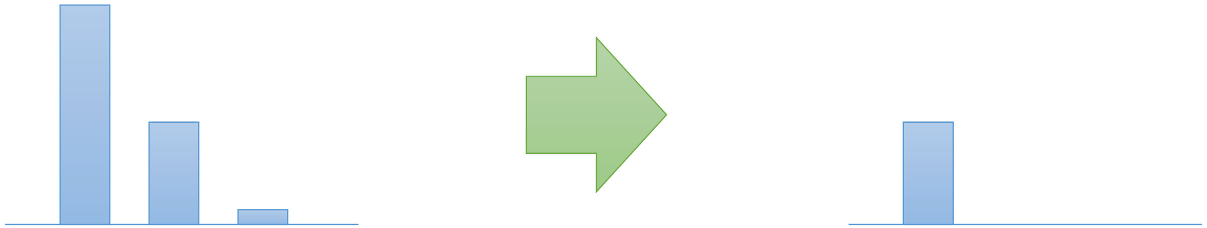
$$E[\|Q[v]\|^2] = \sqrt{n}\|v\|^2$$

Why this is a better quantization

- 1-bit SGD approximates large and tiny coordinates to the same mean value



- The proposed quantization function avoids this by randomizing



Theorem

- The quantized gradient can be communicated in

$$F + \sqrt{n}(\log n \ + \log 2e\ )$$

  bits in expectation

  - F is the number of bits to represent one float number
  - There are only $\sqrt{n}$ non-zero coordinates in expectation
  - For each non-zero entry we use $\mathrm{O}(\log(n))$ bits to encode the location and 1 bit to encode the sign
  - $\sqrt{n}$ times reduction in per iteration communication

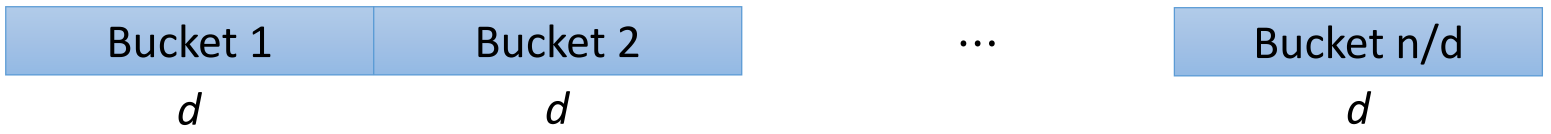
Bucketing

- Apply the quantization for every consecutive d coordinates   (n/d buckets in total)
  - Bucket size *d*=1 corresponds to no quantization
  - Reduced second moment bound => faster convergence

$$E[\|Q_d[v]\|^2] = \sqrt{d}\|v\|^2$$

  - Communication cost

$$\frac{n}{d} \cdot \left(F + \sqrt{d}(\log d \ + \log 2e \ ) \right)$$



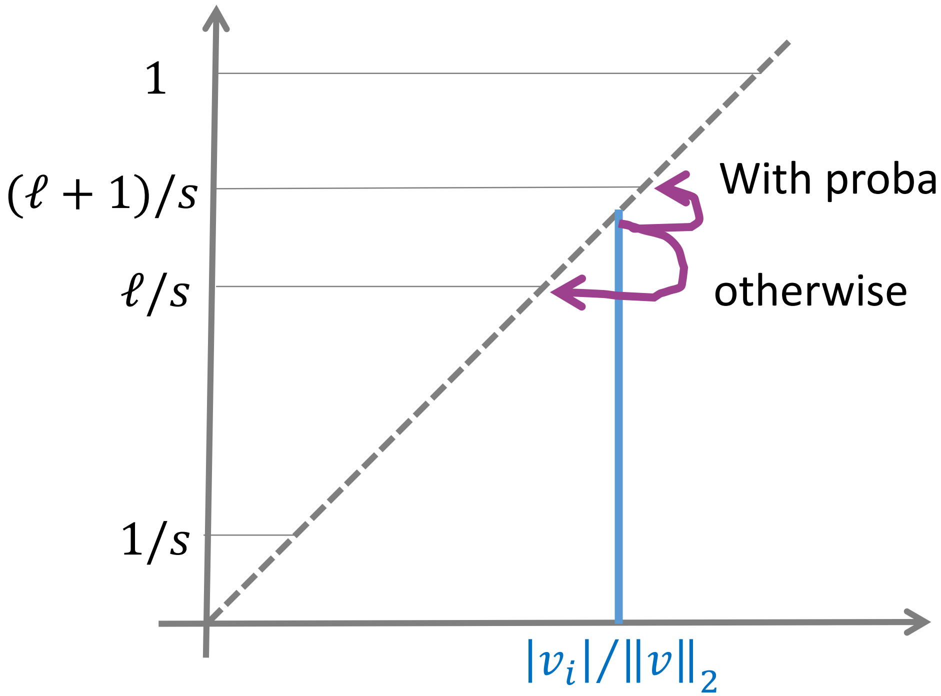
Generalized quantization sheme

- Quantization function

$$Q_i[v; s] = \|v\|_2 \cdot \mathrm{sgn}(v_i) \cdot \xi_i(v, s)$$

(s is a tuning parameter)

where



$\ell = \mathrm{floor}(s \cdot |v_i|/\|v\|_2)$

- Note: s=1 reduces to the simple quantization function.

Properties of the generalized quantization scheme

- Quantization function

$$Q_i[v;s] = \|v\|_2 \cdot \mathrm{sgn}(v_i) \cdot \xi_i(v,s)$$

- Properties

1. Sparsity

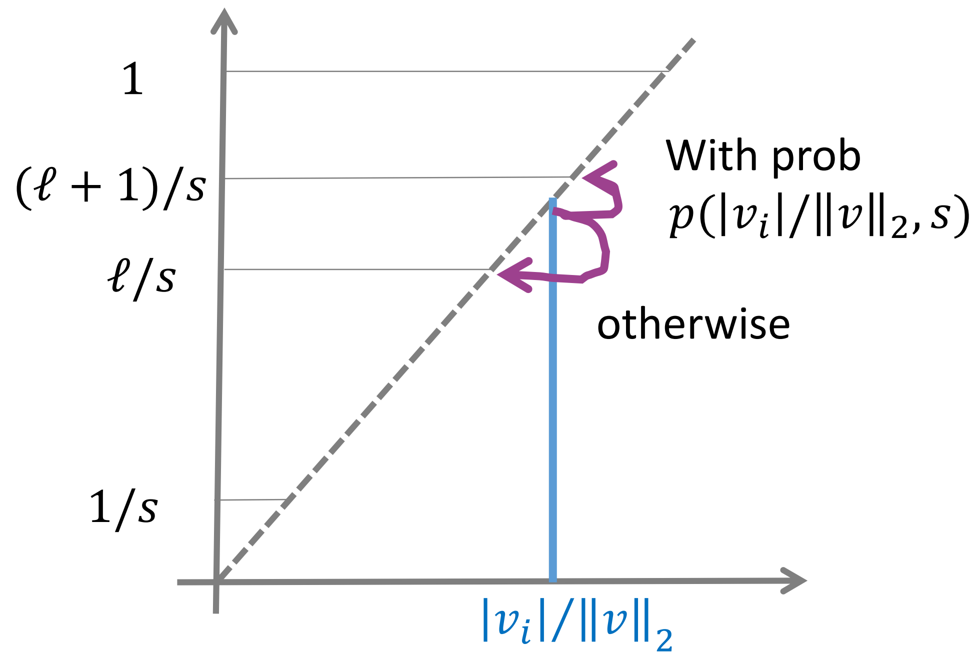
$$E[\|\xi(v,s)\|_0] \le s^2 + \sqrt{n}$$

2. Unbiasedness

$$E\big[Q_i[v;s]\big] = v_i$$

3. Second moment bound

$$E[\|Q[v;s]\|_2^2] \le \left(1 + \min\left(\frac{n}{s^2}, \frac{\sqrt{n}}{s}\right)\right) \cdot \|v\|_2^2$$



$\ell = \mathrm{floor}(s \cdot |v_i|/\|v\|_2)$

(Only $2\|v\|_2^2$ for $s = \sqrt{n}$)

Sublinear Theorem (for small *s*)

- In expectation, the quantized gradient can be communicated in

$$F + \left(3 + \frac{3}{2} \cdot (1 + o(1)) \log\left(\frac{2 \cdot (s^2 + n)}{s^2 + \sqrt{n}}\right)\right) \cdot (s^2 + \sqrt{n})$$

bits

- Communicate the difference of non-zero locations (at most $s^2 + \sqrt{n}$)
- Use Elias recursive coding
- Magnitude can be encoded by log (power per dimension)
- Recovers the simple case for s=1.

Linear Theorem (for large *s*)

- In expectation, the quantized gradient can be communicated in

$$F + \left(\frac{1+o(1)}{2}\left(\log\left(1+\frac{s^2+\min(n, s\sqrt{n})}{n}\right)+1\right)+2\right)\cdot n$$

  bits

  - Don't communicate the locations
  - For $s = \sqrt{n}$, we have the bound

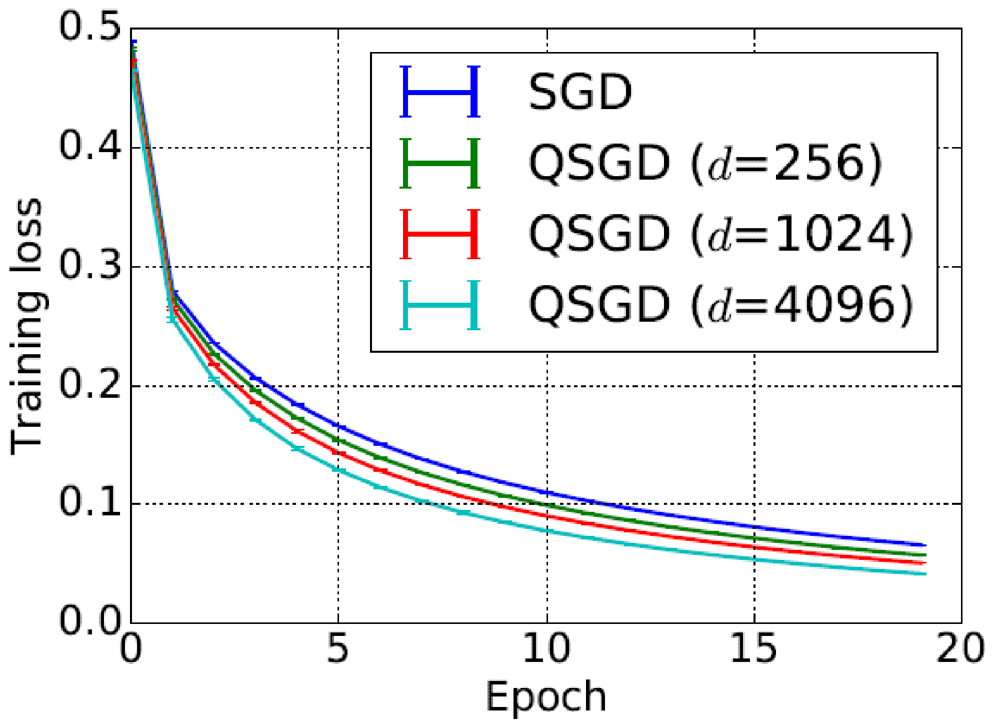
$$F + 2.8n$$

  - Linear in dimension *n* but much smaller constant 2.8 compared to uncompressed float (32 bits) and second moment only 2 times worse.

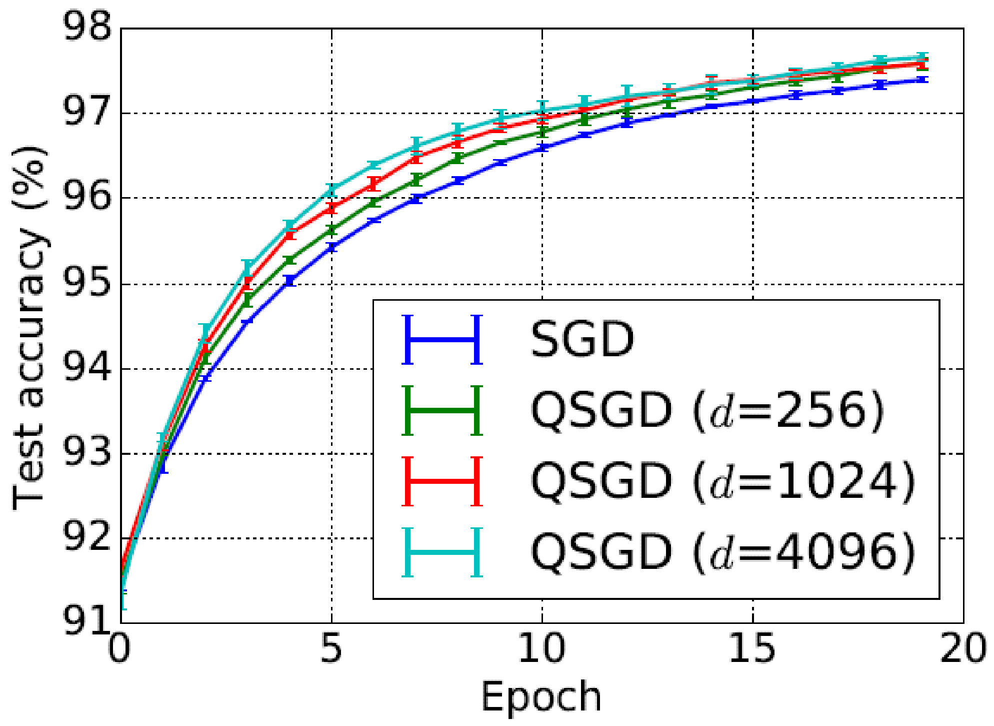
Experiments

MNIST (digit recognition task)

- Two-layer Network (non-convex!):
  - Input 784 -> hidden 4096 -> output 10
- Used the simple quantization scheme with bucketing (bucket size: d)



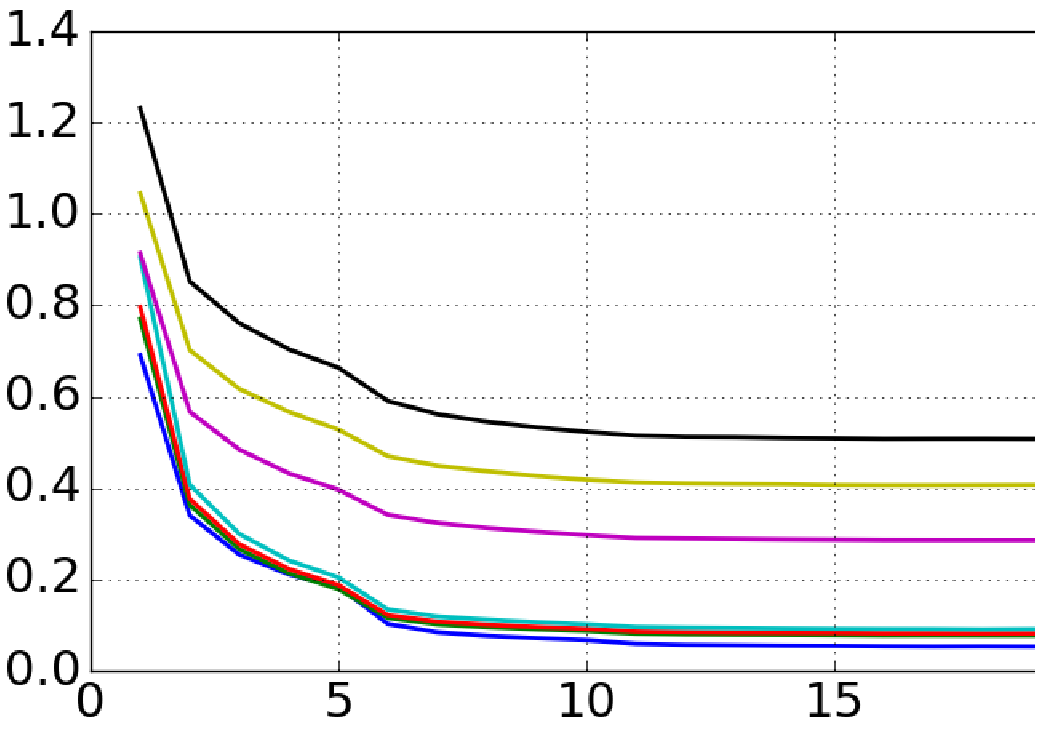
(a) MNIST training loss



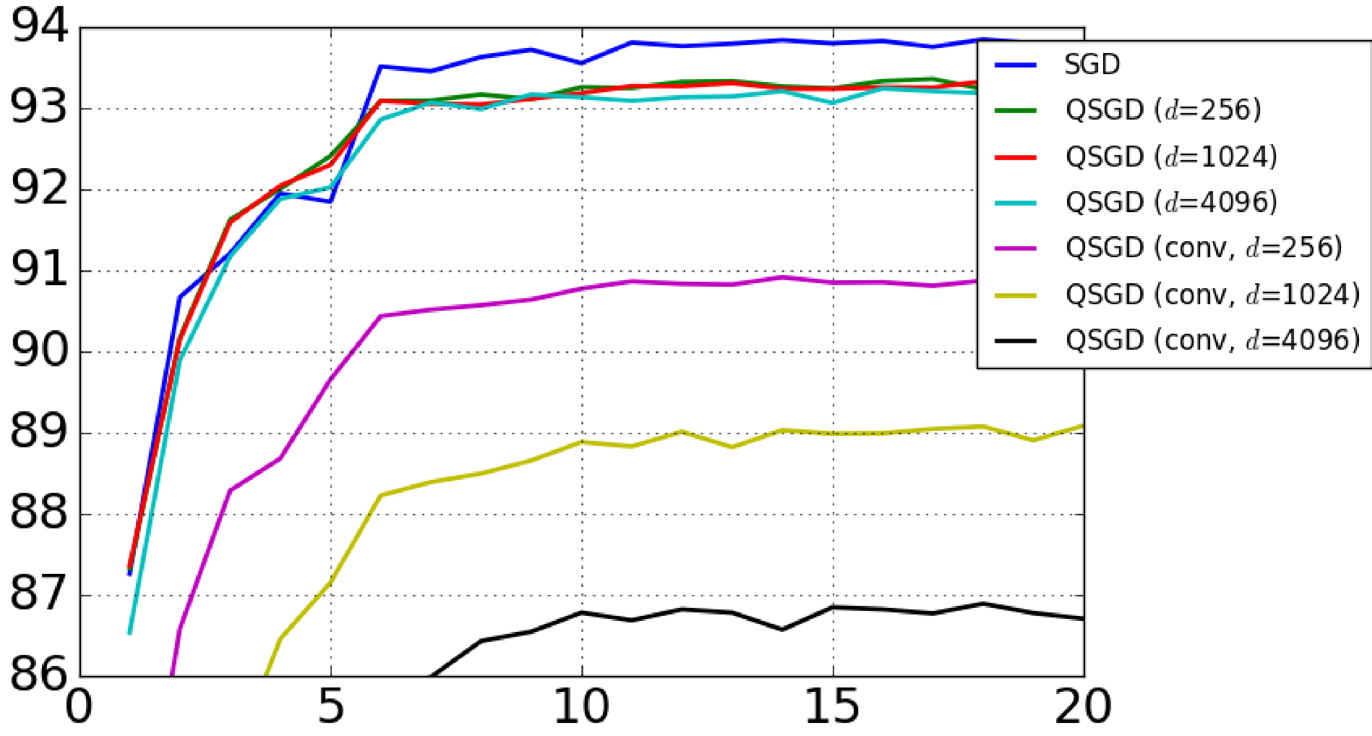
(b) MNIST test accuracy

CIFAR-10 (object recognition task)

- Convolutional network (a small VGG network), 12 layers
  - Input → Conv → BN → Conv → BN → … → Hidden 4096 → Hidden 4096 → Hidden 4096 → Output 10

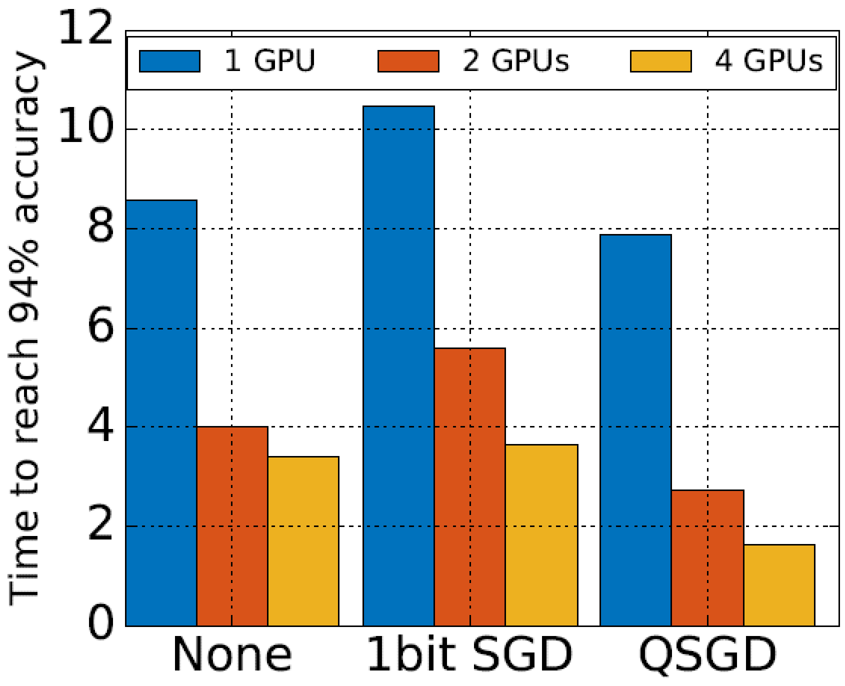


Training loss

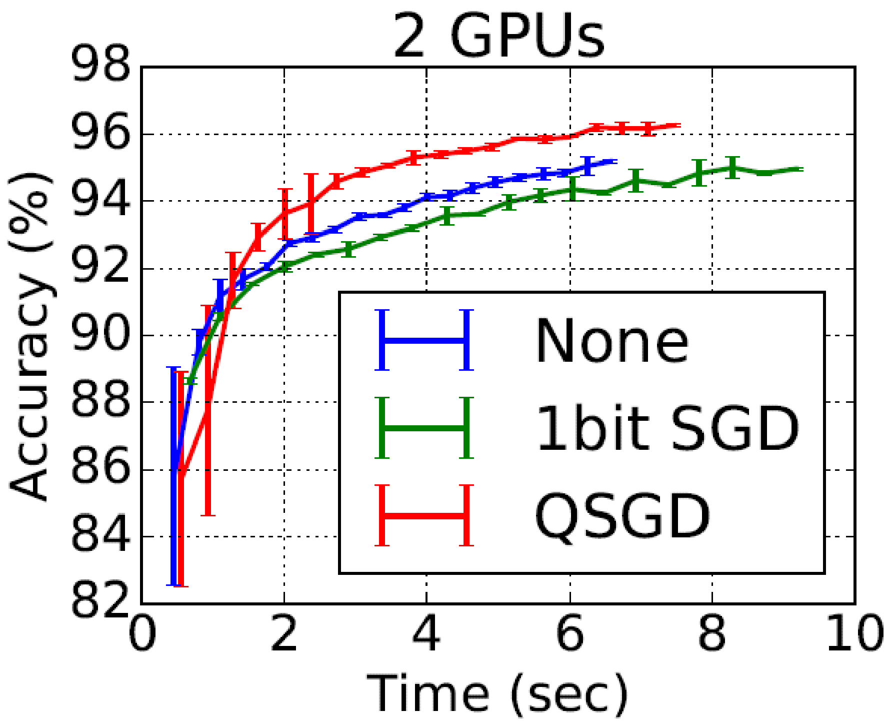


Test accuracy

Parallelization (preliminary)



(a) Data parallel training time



(b) Data parallel accuracy

Conclusion

- Simple, easy-to-implement quantization scheme
  - Sublinear ($\sqrt{n}$) number of bits per iteration
  - Performance guarantee
  - Bucketing to control compression / convergence trade-off
- General quantization scheme
  - Requires roughly 3 bits per coordinate and convergence guarantee only 2 times worse compare to SGD

Generalized quantization sheme

- Quantization function

$$Q_i[v; s] = \|v\|_2 \cdot \mathrm{sgn}(v_i) \cdot \xi_i(v, s)$$

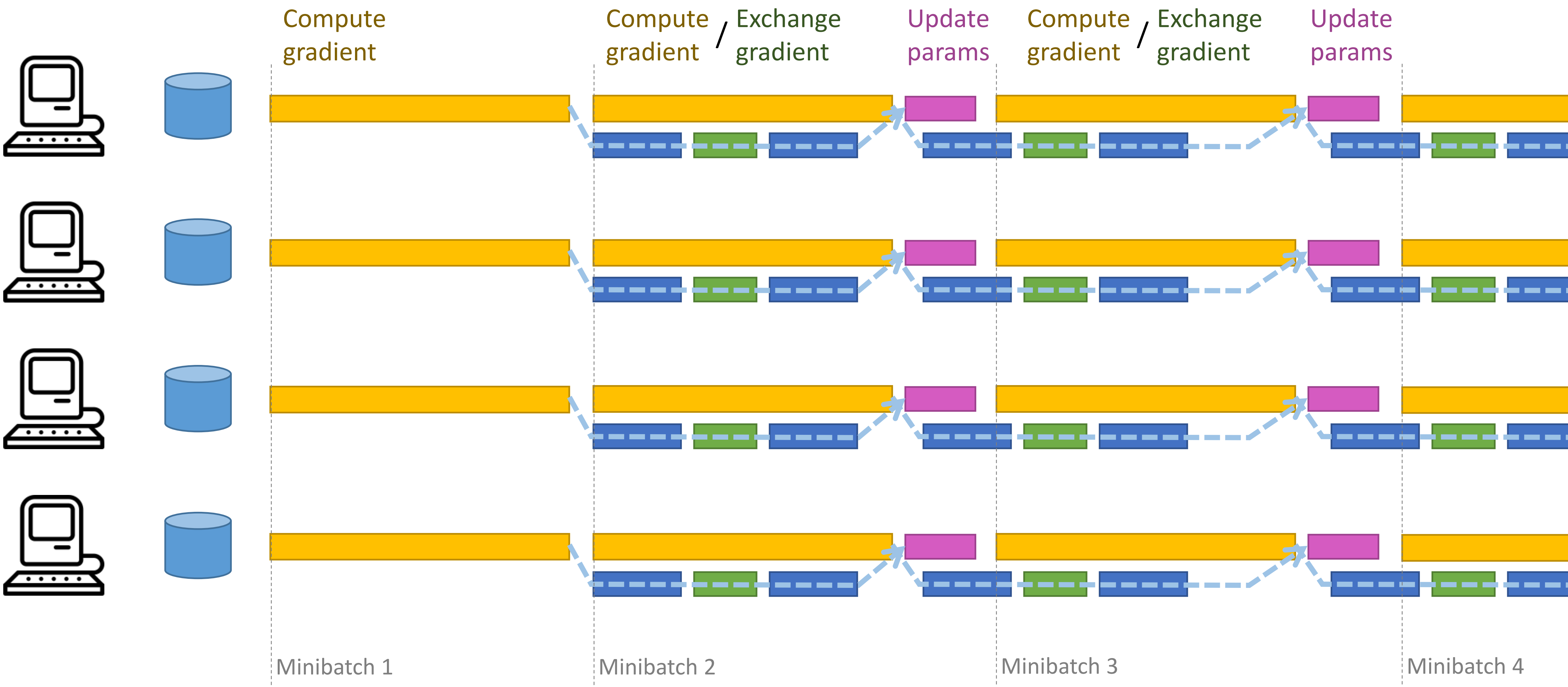
where

$$\xi_i(v, s) = \begin{cases} \ell/s & \text{with probability } \ell + 1 - s \cdot |v_i|/\|v\|_2, \\ (\ell + 1)/s & \text{otherwise,} \end{cases}$$

with $\ell = \mathrm{floor}(s \cdot |v_i|/\|v\|_2)$ and s is a hyper-parameter.

- Note: s=1 reduces to the simple quantization function.

Double buffering



Thanks

Streamline icons