machine learning conference

signSGD: adaptive gradient meets gradient compression for distributed deep learning

Yu-Xiang Wang @yuxiangw
Applied Scientist | AWS Deep Engine-Science

Based on joint work with:
Jeremy Bernstein, Kamyar Azizzadenesheli, Anima Anandkumar, Rahul Huilgol



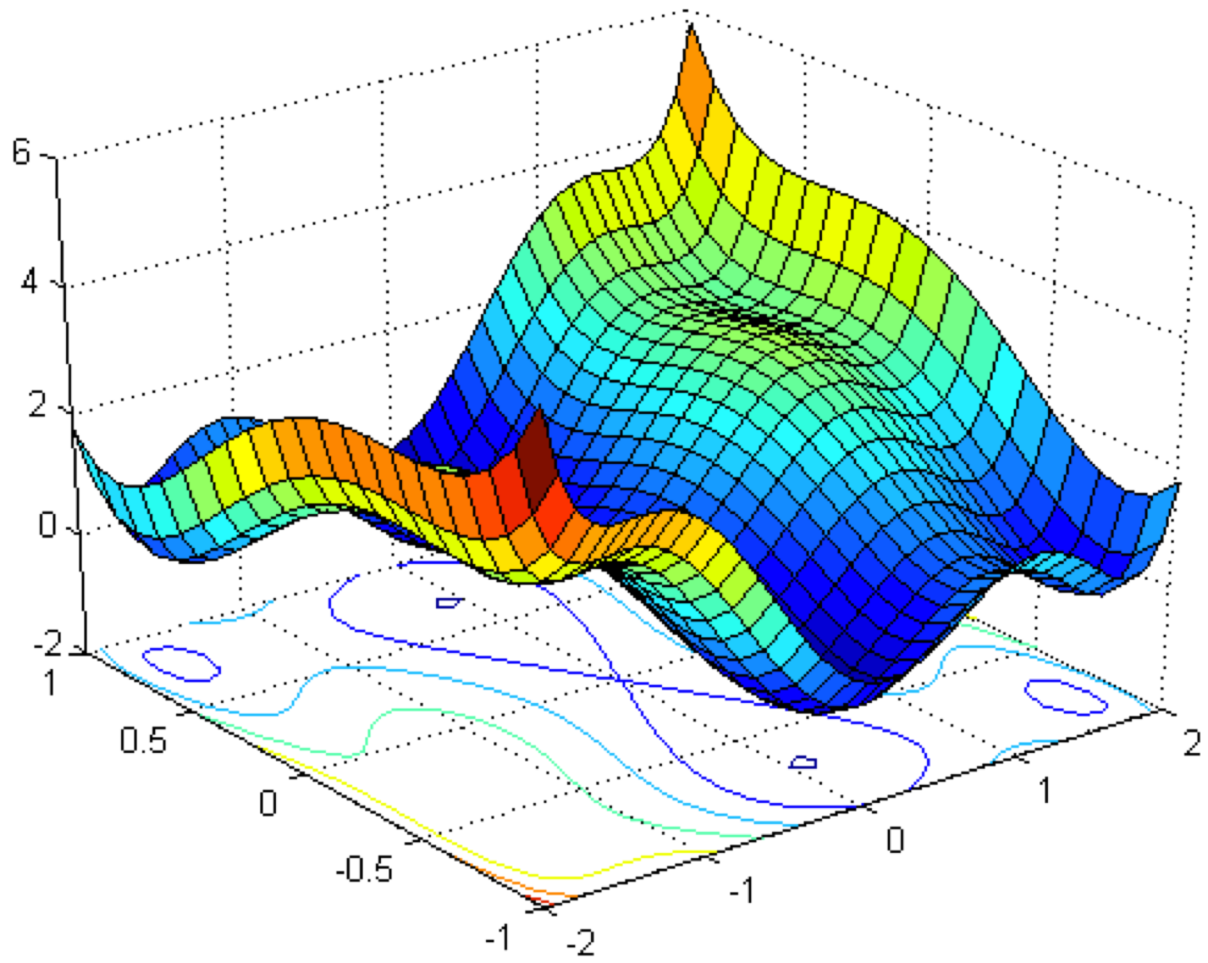

Outline:

1. From Adam to Signum in practical deep learning

2. The convergence rate of signSGD / Signum

3. 1-bit compression and communication-efficient distributed training with signSGD / Signum

Deep learning is a nonconvex optimization problem



Zoos of algorithms: for nonconvex optimization **theory**

- Nonconvex SVRG [Reddi et. al., 2015, Allen-Zhu & Hazan, 2015]
- Noisy GD / SGD [Ge et. al.2015, Jin et. al. 2017]
- Trust-region method [Sun et. al., 2015]
- Natasha 1/2 [Allen-Zhu, 2017]

Detailed computational theory on convergence rate to **stationary points** and to **local minima**!

Zoos of algorithms: for deep learning practice

- SGD [Robbins and Monro, 1951]
- Momentum [Polyak, 1964; Nesterov, 1983]
- Adagrad [Duchi et. al., 2011] / Adam [Kingma & Ba, 2014]
- Rprop [Riedmiller&Braun, 1993] / RMSprop [Tieleman&Hinton, 2012]

Not well understood theoretically (perhaps except SGD).

Hammers and tricks

- Variance reduction
- Active noise adding
- Hessian-vector product
- Cubic regularization

v.s.

- Momentum
- Gradient clipping
- Adaptive gradient
- Batch normalization

Hammers and tricks

- Variance reduction
- Active noise adding
- Hessian-vector product
- Cubic regularization

v.s.

- Momentum
- Gradient clipping
- Adaptive gradient
- Batch normalization

Why don’t we try to analyze the tricks?

The Adam algorithm

Adam: A method for stochastic optimization

D **Kingma**, J Ba - arXiv preprint arXiv:1412.6980, 2014 - arxiv.org

Abstract: We introduce Adam, an algorithm for first-order gradient-based optimization of stochastic objective functions, based on adaptive estimates of lower-order moments. The method is straightforward to implement, is computationally efficient, has little memory

☆ 99 Cited by 5983 Related articles All 9 versions

[PDF] A stochastic approximation method

H **Robbins**, S **Monro** - The annals of mathematical statistics, 1951 - JSTOR

Let M (x) denote the expected value at level x of the response to a certain experiment. M (x) is assumed to be a monotone function of x but is unknown to the experimenter, and it is desired to find the solution x= θ of the equation M (x)= α, where α is a given constant. We

☆ 99 Cited by 5143 Related articles All 8 versions Web of Science: 1930 »

The Adam algorithm

- Adam update:

$$m_t = \beta m_{t-1} + (1-\beta) g_t$$ momentum

$$v_t = \gamma v_{t-1} + (1-\gamma) g_t^2$$ variance

$$x_t = x_{t-1} + \eta \frac{m_t}{\sqrt{v_t}}$$ variance adjusted momentum update

The Adam algorithm

- Adam update:

$$m_t = \beta m_{t-1} + (1-\beta) g_t$$ momentum

$$v_t = \gamma v_{t-1} + (1-\gamma) g_t^2$$ variance

$$x_t = x_{t-1} + \eta \frac{m_t}{\sqrt{v_t}}$$ variance adjusted momentum update

Key idea: dividing the gradient (coordinate wise) by its magnitude!

From Adam to SignSGD to Signum

- SignSGD update: (Related to Rprop [Riedmiller&Braun, 1993])

$$x_t = x_{t-1} + \eta \frac{g_t}{|g_t|}$$

From Adam to SignSGD to Signum

- SignSGD update: (Related to Rprop [Riedmiller&Braun, 1993])

$$x_t = x_{t-1} + \eta \frac{g_t}{|g_t|}$$

- Signum (SIGN momentUM):

$$m_t = \beta m_{t-1} + (1-\beta) g_t$$

$$x_t = x_{t-1} + \eta \frac{m_t}{|m_t|}$$

From Adam to SignSGD to Signum

- SignSGD update: (Related to Rprop [Riedmiller&Braun, 1993])

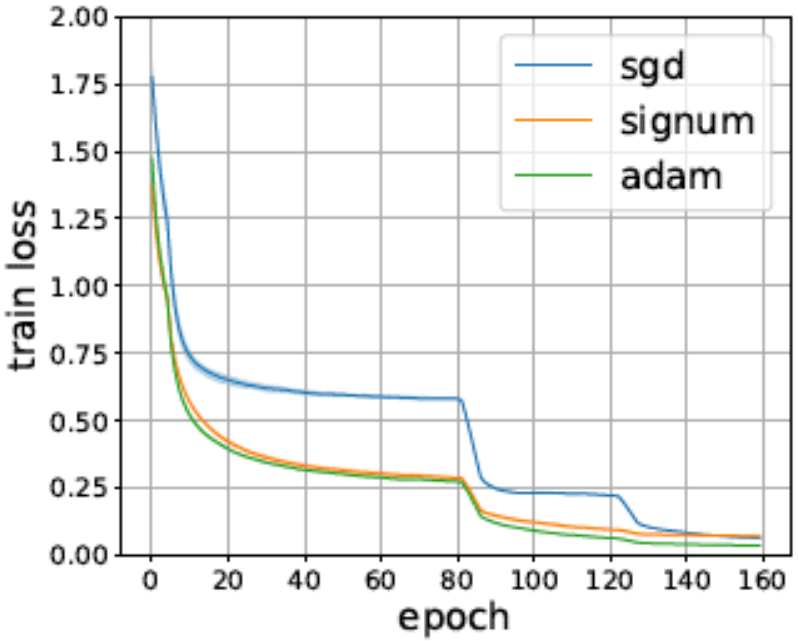
$$x_t = x_{t-1} + \eta \boxed{\frac{g_t}{|g_t|}} \longleftarrow \text{sign}(g_t)$$

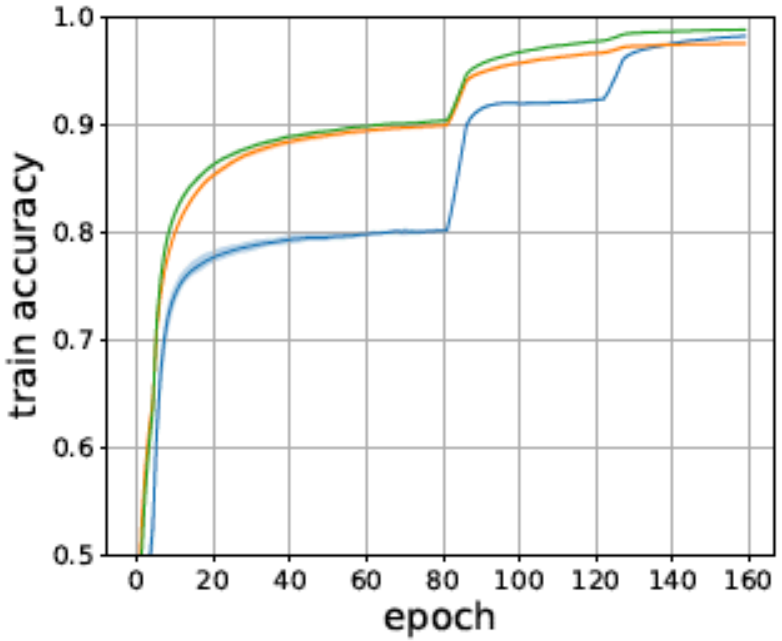
- Signum (SIGN momentUM):

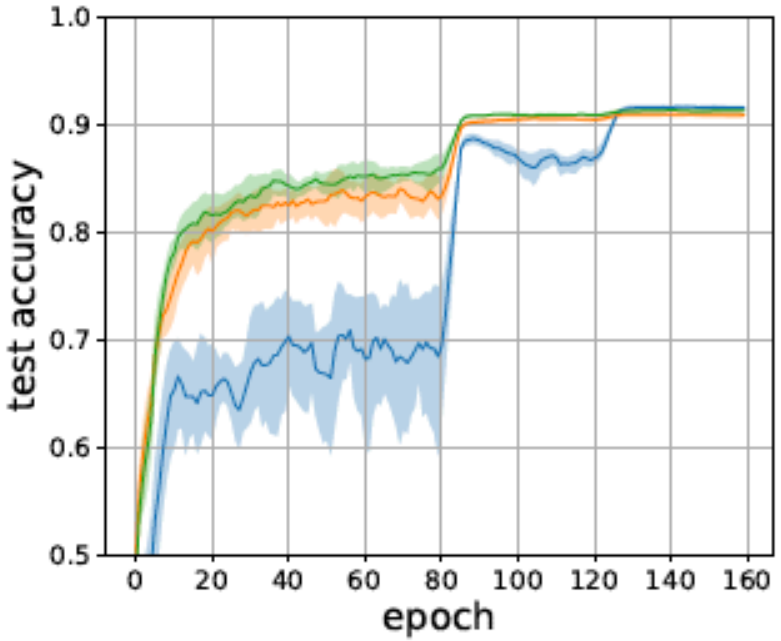
$$m_t = \beta m_{t-1} + (1-\beta) g_t$$

$$x_t = x_{t-1} + \eta \boxed{\frac{m_t}{|m_t|}} \longleftarrow \text{sign}(m_t)$$

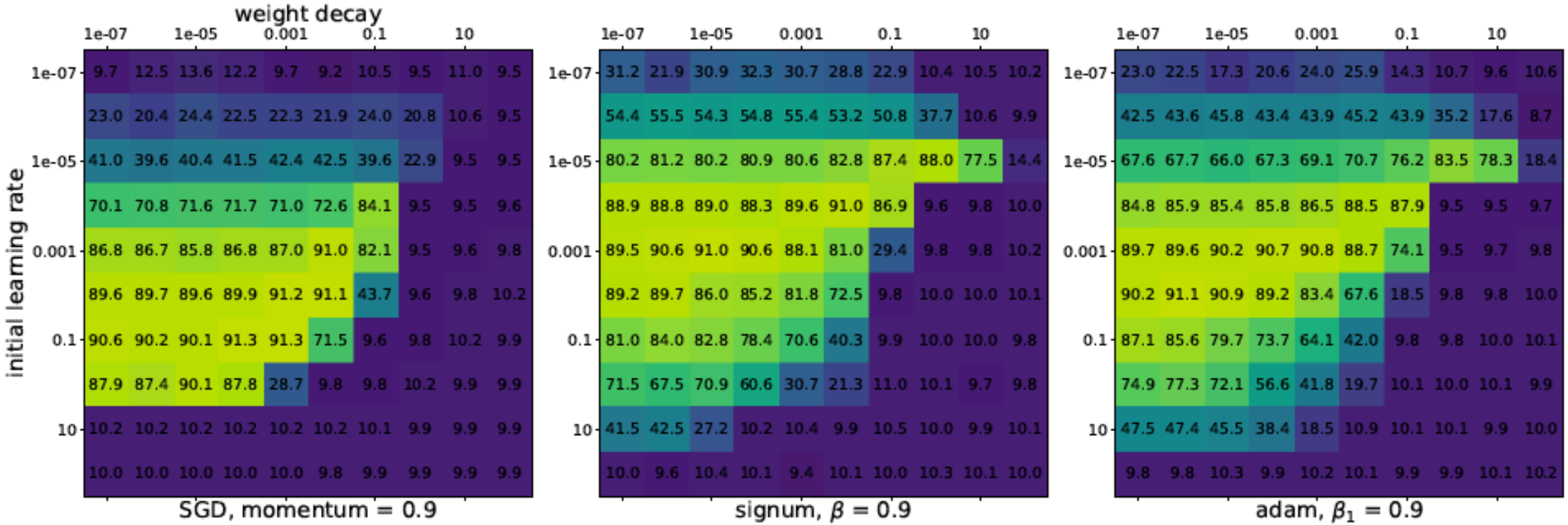
Experiment on CIFAR10 suggests Signum and Adam are very similar



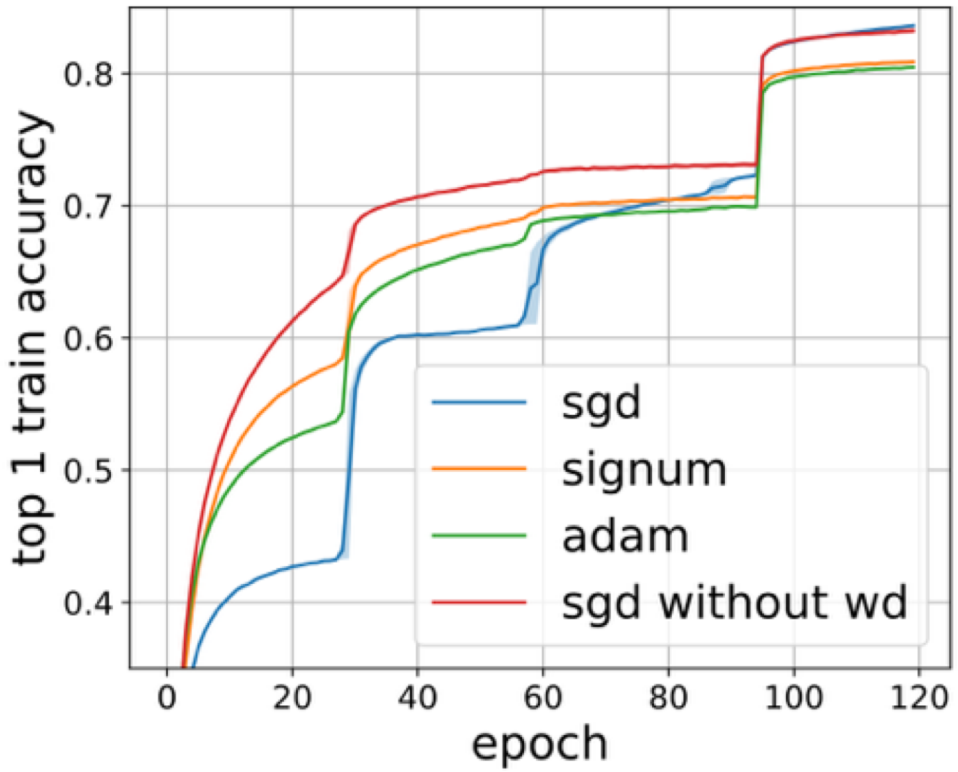


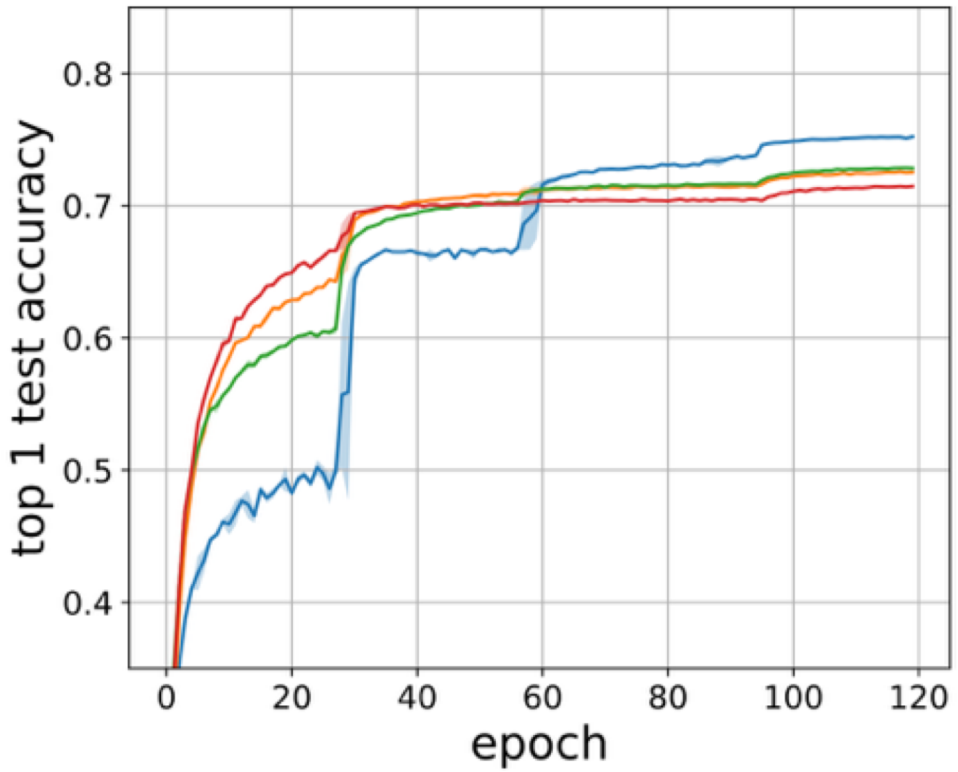


Experiment on CIFAR10 suggests Signum and Adam are very similar



Signum is competitive on ImageNet





Advantages of SignSGD/Signum over Adam

- Theoretical guarantee on convergence
  - For nonconvex problems, with or without momentum

- Build-in 1-bit gradient compression.
  - Communication-efficient.

Stationary point convergence of signSGD & Signum

- Stochastic optimization model with “noisy gradient” access.
- How close does a budget of N such access get us to a solution?

Stationary point convergence of signSGD & Signum

- Stochastic optimization model with "noisy gradient" access.
- How close does a budget of N such access get us to a solution?

Theorem: For a fixed learning rate and minibatch size schedule

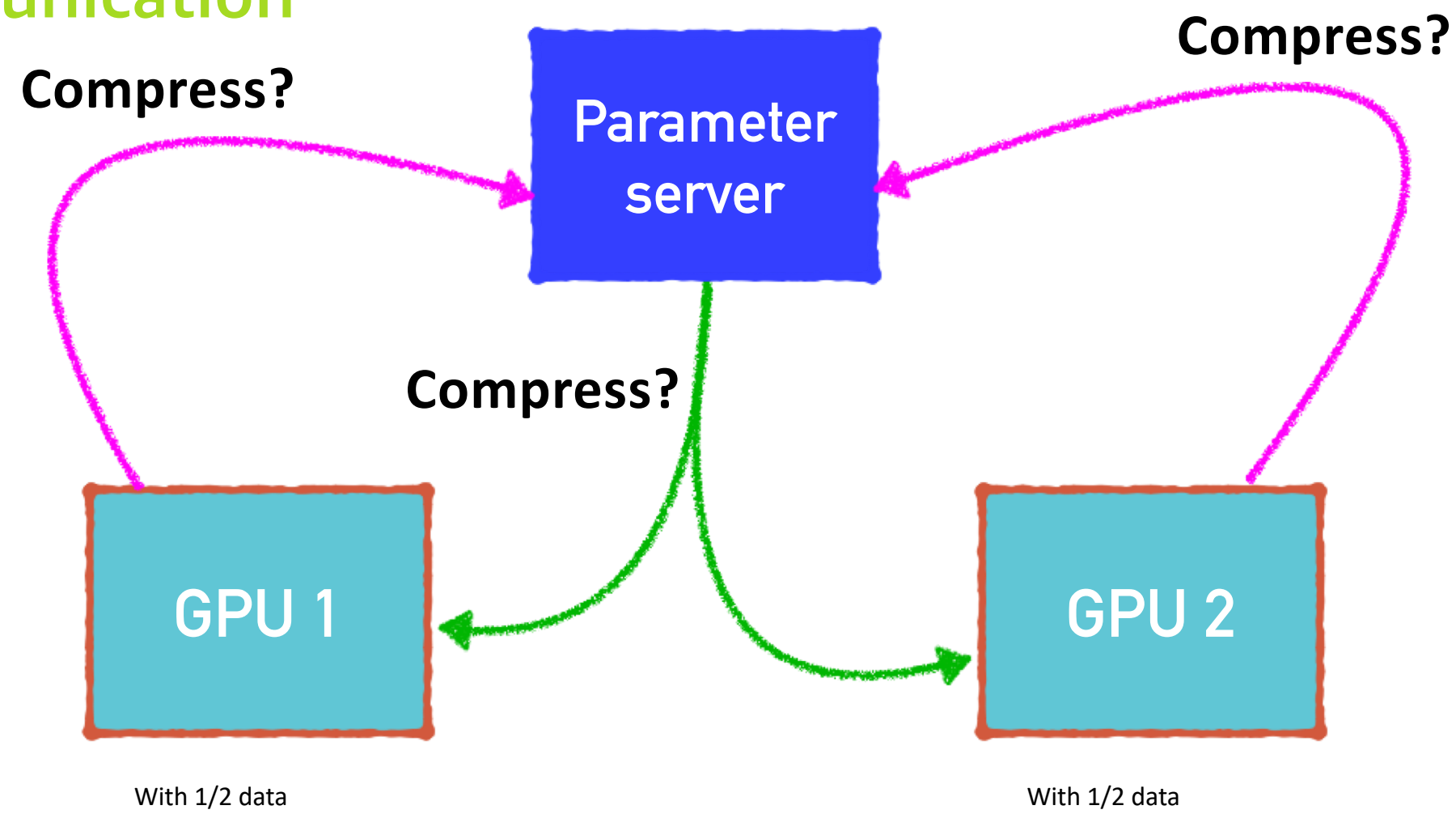
$$\min_{C \leq k \leq K-1} \mathbb{E}[\|g_k\|_1]^2 \leq \frac{1}{\sqrt{N}} \left[\sqrt{\|\vec{L}\|_1} \left(f_0 - f_* + \frac{1}{2} \right) + 2\|\vec{\sigma}\|_1 \right]^2$$

- Burn-in period: C=1 for SignSGD and O(1) for momentum.

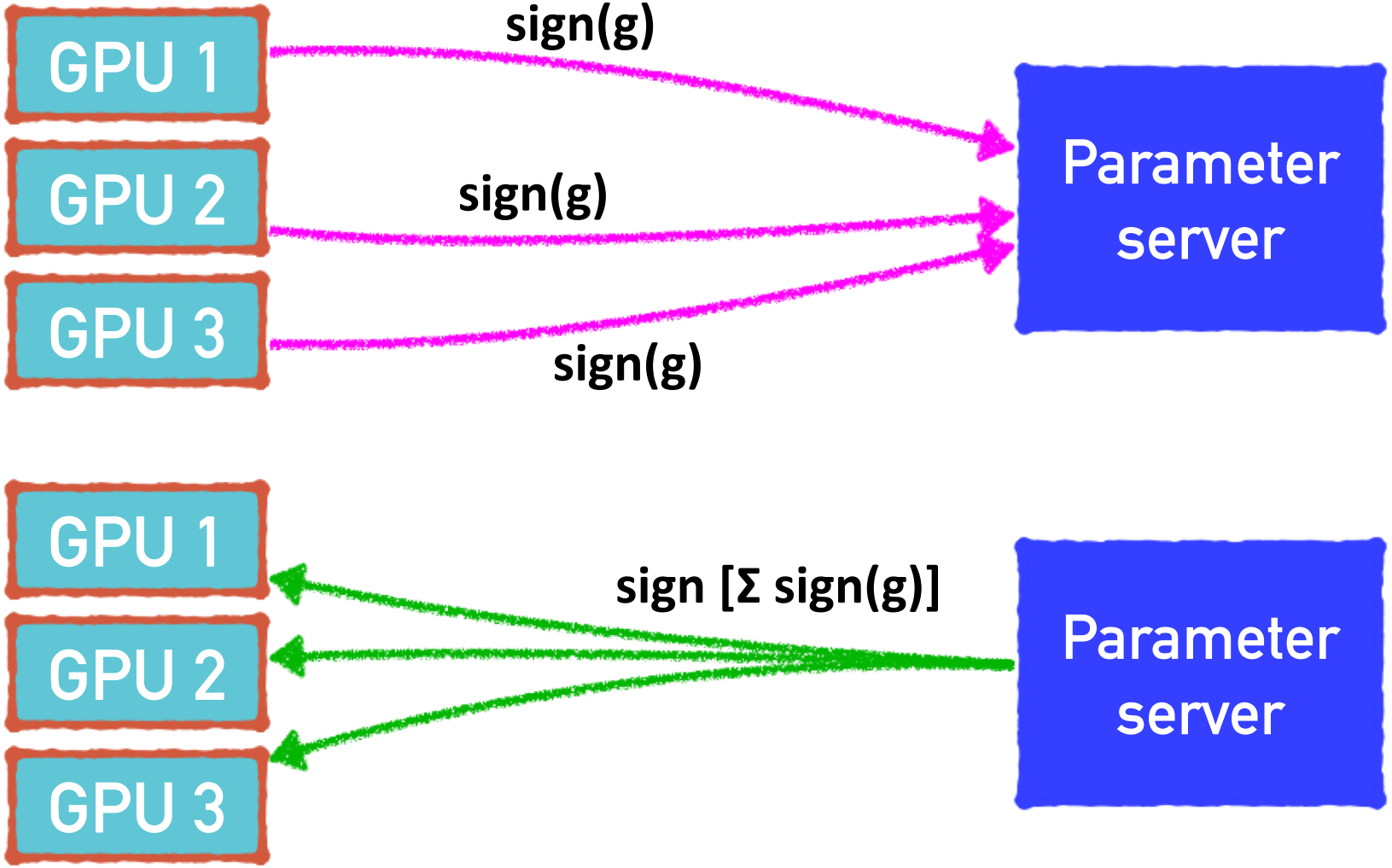
Key messages from our theory

- SignSGD is not approximating SGD.
  - Unlike Seide et. al. (2014), or Alistarh et. al. (2017).
- SignSGD can be faster than SGD in some cases (and vice versa)
- Radically low-precision training (1 bit) does **NOT** come with a price!

Distributed training involves computation and communication

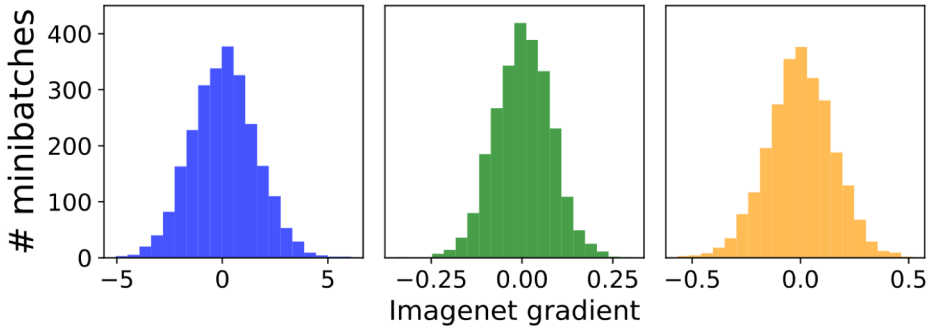


Distributed training by SignSGD with Majority Voting



Does Majority Voting work?

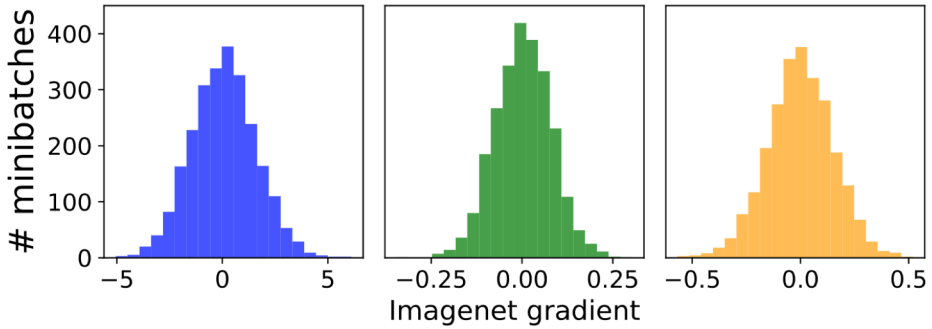
If gradients are unimodal
and symmetric…



…which is reasonable by the
central limit theorem…

Does Majority Voting work?

If gradients are unimodal and symmetric…



…which is reasonable by the central limit theorem…

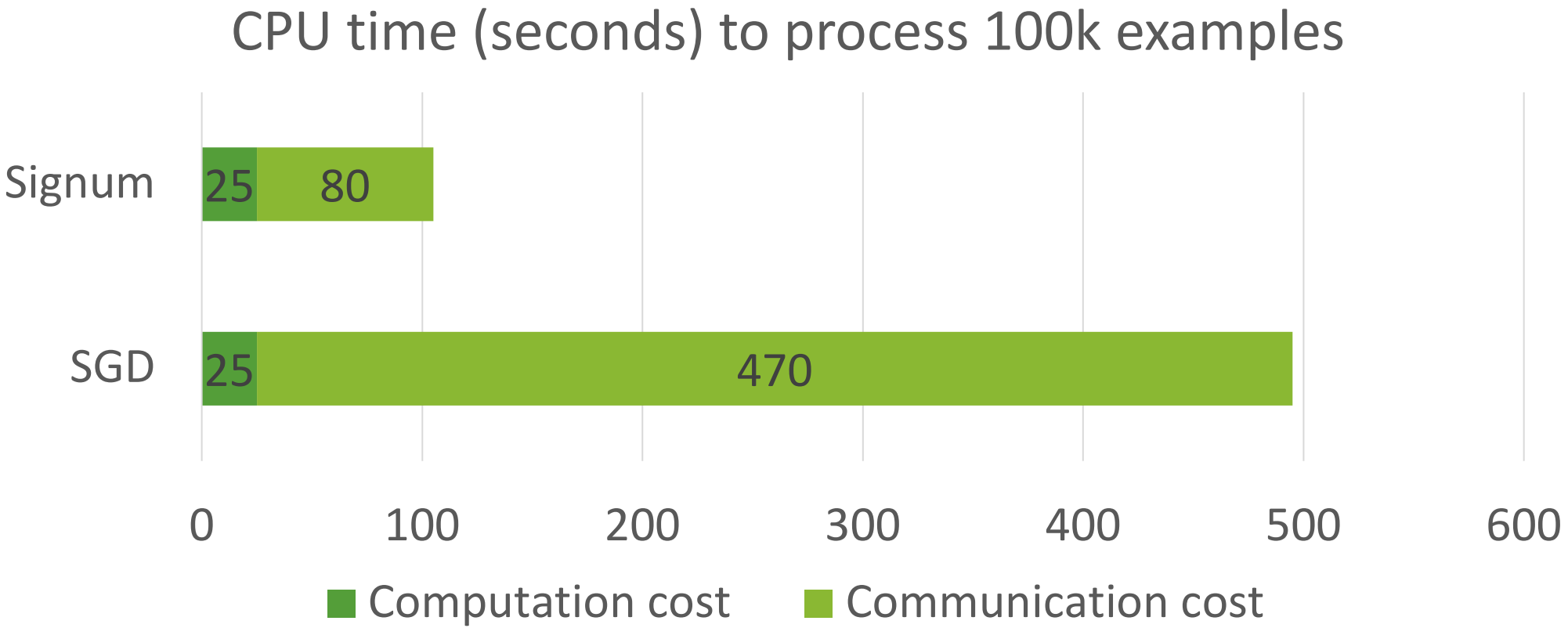
…then majority vote with M workers converges at rate:

$$\mathbb{E}\left[\min_{0\leq k\leq K-1}\|g_k\|_1\right]^2$$

$$\leq \frac{1}{\sqrt{N}}\left[\sqrt{\|\vec{L}\|_1}\left(f_0 - f_* + \frac{1}{2}\right) + \frac{2}{\sqrt{M}}\|\vec{\sigma}\|_1\right]^2$$

Distributed training of VGG-19 on ImageNet

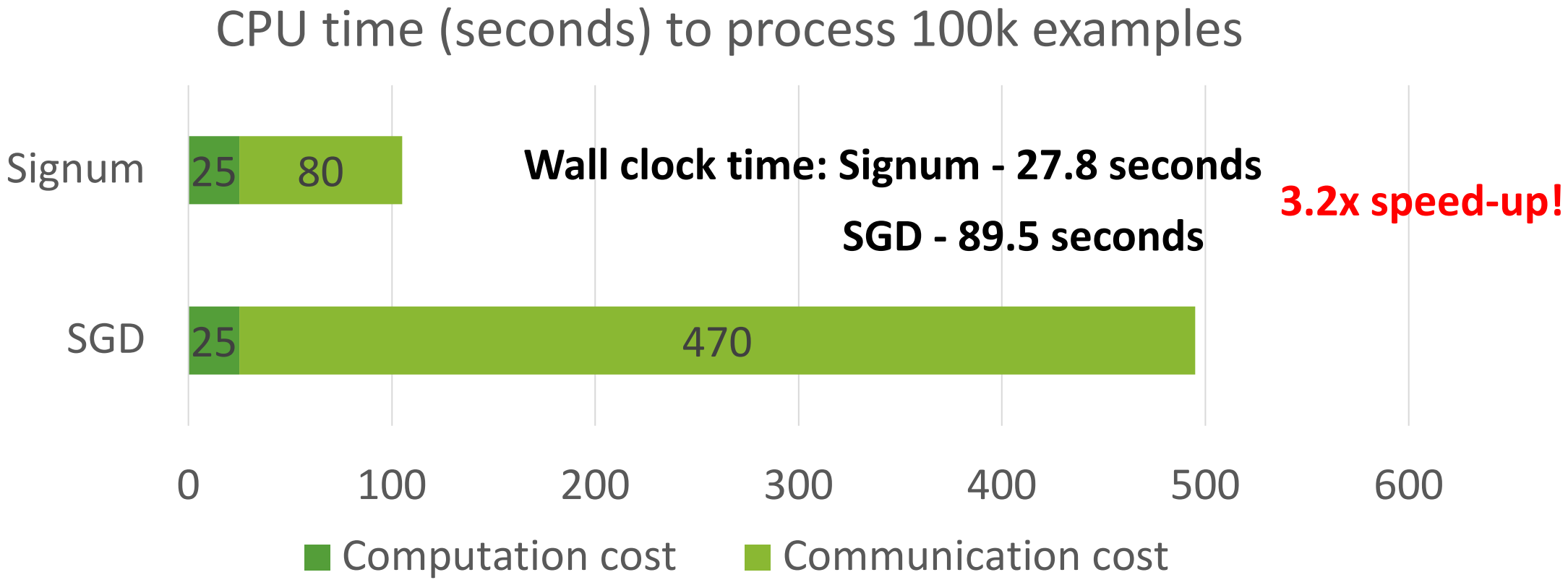
Setup: 10 machine cluster of p3.8x machines in the same placement group. 10k data each.



Rahul Huilgol

Distributed training of VGG-19 on ImageNet

Setup: 10 machine cluster of p3.8x machines in the same placement group. 10k data each.



Wall clock time: Signum - 27.8 seconds

SGD - 89.5 seconds

3.2x speed-up!

Rahul Huilgol

Conclusion: **Just follow the signs!**

- SignSGD and Signum optimizers ≈ Adam
- They come with theoretical guarantees with a rate of convergence
- Majority voting: 1-bit compression in both ways for distributed training!

Signum implementation available in mxnet:

1. Uncompressed version

```
trainer = gluon.Trainer( net.collect_params(),
                                                    'sgd',
                                                   {'learning_rate': opt.lr,
                                                    'momentum': opt.momentum})

trainer = gluon.Trainer( net.collect_params(),
                                                    'signum',
                                                   {'learning_rate': opt.lr/10,
                                                    'momentum': opt.momentum})
```

2. Compressed version——————Majority vote

Coming soon!!

Thanks for your attention!

Yu-Xiang Wang @yuxiangw

Applied Scientist | AWS Deep Engine-Science

Based on joint work with:

Jeremy Bernstein, Kamyar Azizzadenenesheli, Anima Anandkumar, Rahul Huilgol