U.S. DEPARTMENT OF ENERGY

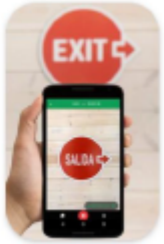
Pacific Northwest NATIONAL LABORATORY
Proudly Operated by Battelle Since 1965

Lossy Compression for Deep Neural Networks

Shuaiwei Song, Pacific Northwest National Laboratory
Jieyang Chen, University of California, Riverside

Deep Neural Networks

- Deep Neural Networks (DNNs) has been extensively developed and used
  - LeNet
  - AlexNet
  - GoogleNet
  - VGG
  - ResNet
- They have becoming more powerful and can handle more complicated tasks
  - Image classification
  - Object detection
  - Etc.



Phones

Robots

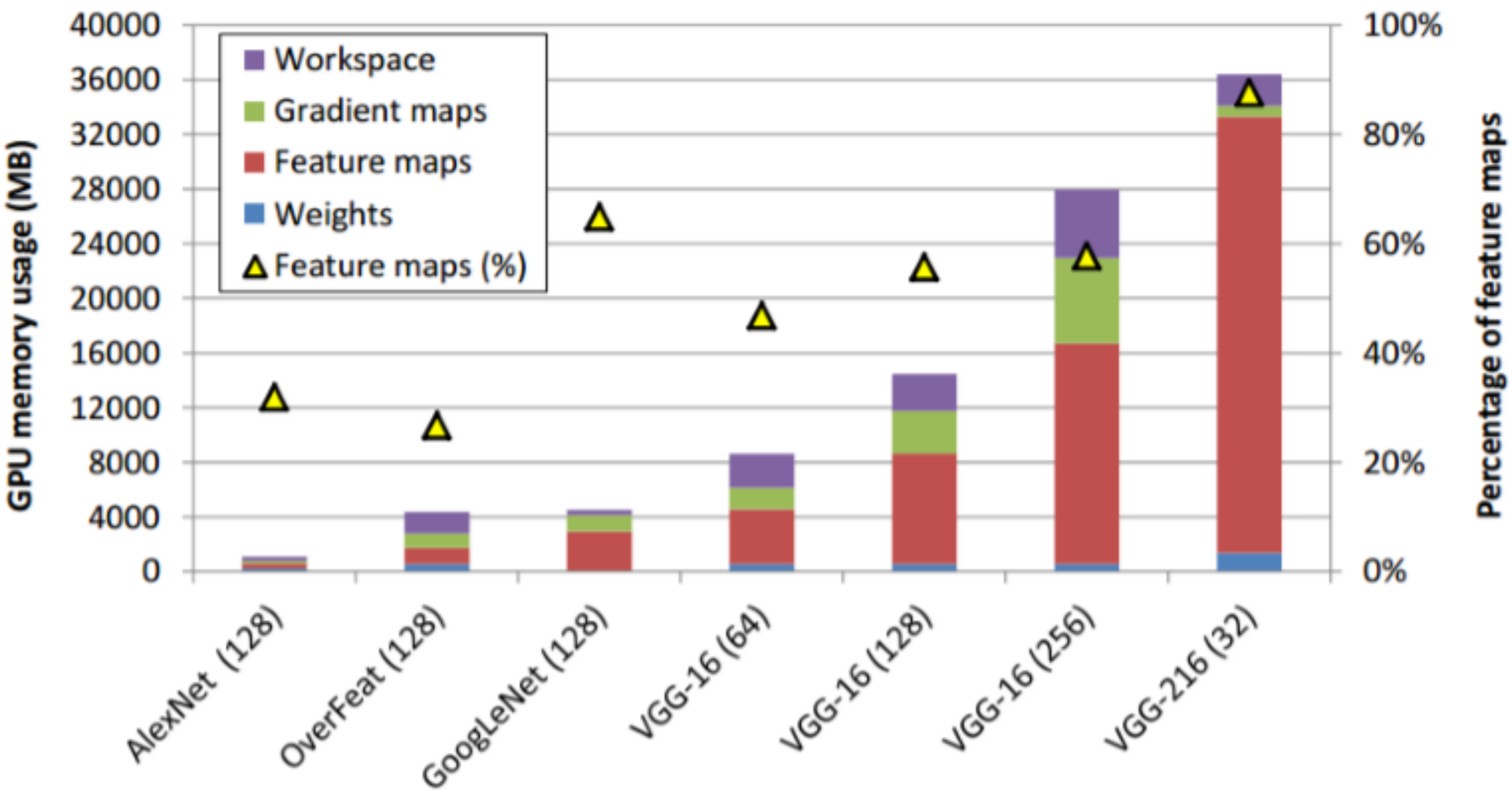
Glasses

Drones

Growing number of parameters

- More powerful DNNs are made possible by
  - More parameters
  - More complicated structures
- As more advanced DNNs are developed → DNNs becomes deeper
  - Lenet-5: 4 layers
  - Alexnet: 7 layers
  - VGG-16: 16 layers
  - ResNet: 18 – 152 layers

DNNs storage sizes are growing fast



Challenges bring by larger DNNs

- Limited by the available RAM/storage space and network performance, it can be hard to:
  - Transferring DNNs between systems during training.
    - *E.g., coarse tuning on one system and fine tuning on another system.*
  - Publishing pre-trained DNNs on webs.
    - *E.g., ILSVRC winners want to share their novel DNNs.*
  - Deploying DNNs on systems for inference.
    - *E.g., An application that uses pre-trained DNN needs an update from vender.*
  - *Loading DNNs on GPUs with small memory*
    - *E.g., Sometimes our training/testing platform can be very heterogeneous.*

What can users do with large models?

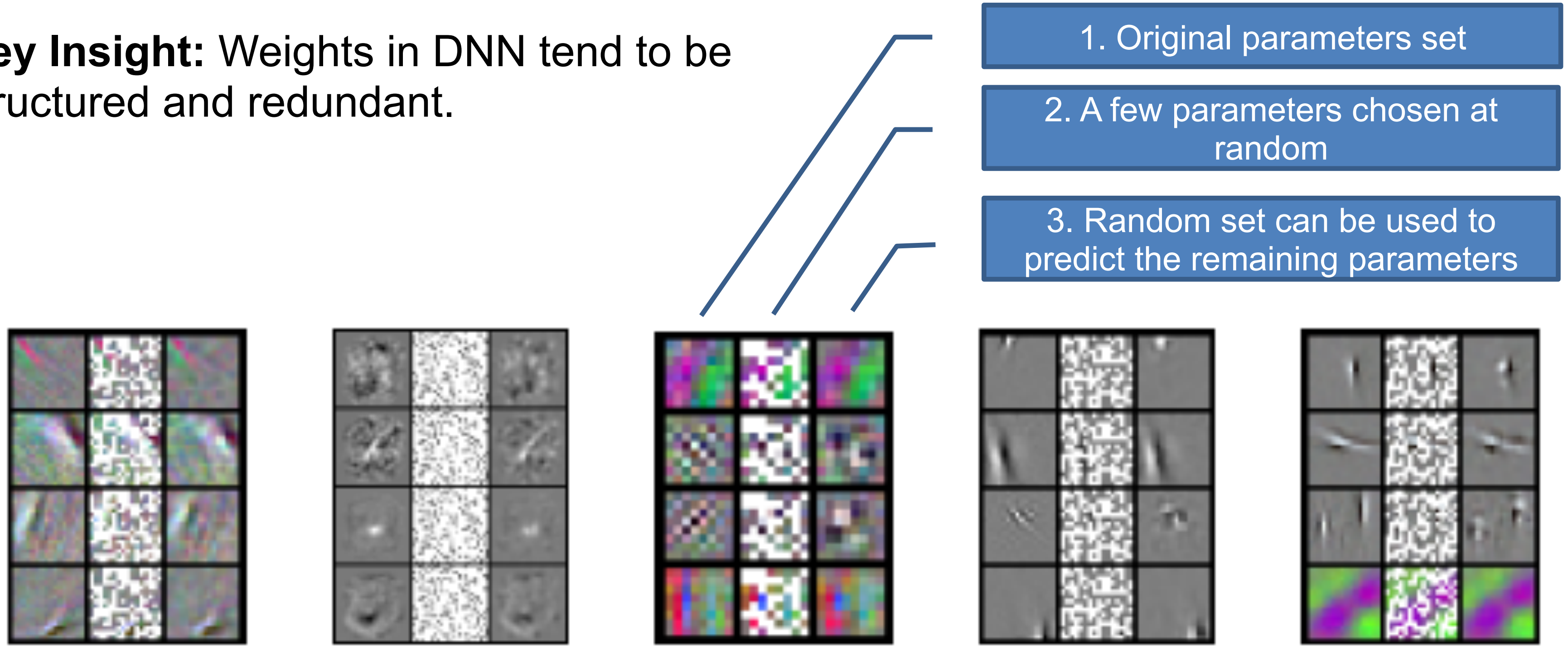
- Reduce batch size to make more space for the model. However, it may:
  - Decrease training speed
  - Impact accuracy
- Distribute on multiple GPUs or nodes
  - More computing resource requirement
  - Efficient design can be complicated

No perfect choice for users!

Pacific Northwest NATIONAL LABORATORY

DNNs are over-parameterized [Denil et al. NIPS'13]

Key Insight: Weights in DNN tend to be structured and redundant.



CNN trained on STL-10

MLP trained on MNIST

CNN trained on CIFAR-10

R-ICA trained on Hyvarinen's natural image dataset

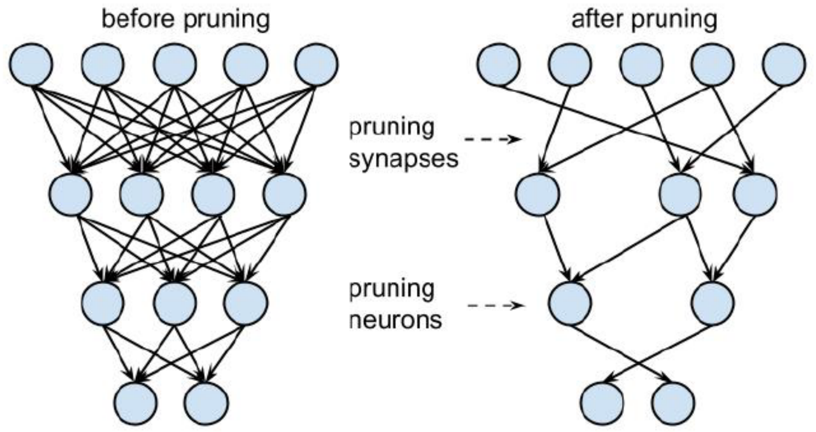
R-ICA trained on STL-10 trained

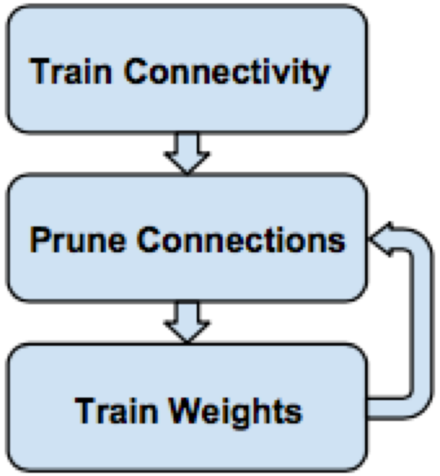
7

Current approaches for compressing model

- **Matrix decomposition:**
  - Denton et al. NIPS’14, Denil et al. NIPS’13
- **Network pruning:**
  - Han et al. ICLR’18, Han et al. ICLR’17, Han et al. ICLR’15
- **Weight quantization:**
  - Yunchao et al. ICLR’15, Han et al. ICLR’16, Courbariaux et al. NIPS’15, Gupta et al. ICML’15
- Etc.

Network pruning [Han et al. ICLR'15]





Iterative pruning to minimize connections

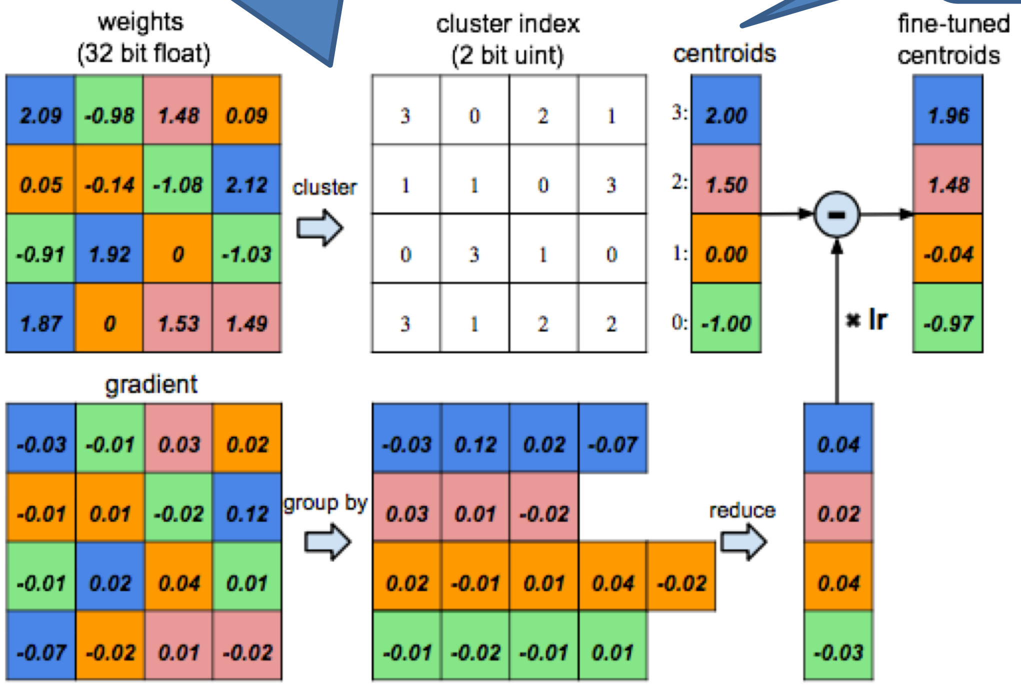
Store sparse weights in CSR/CSC format

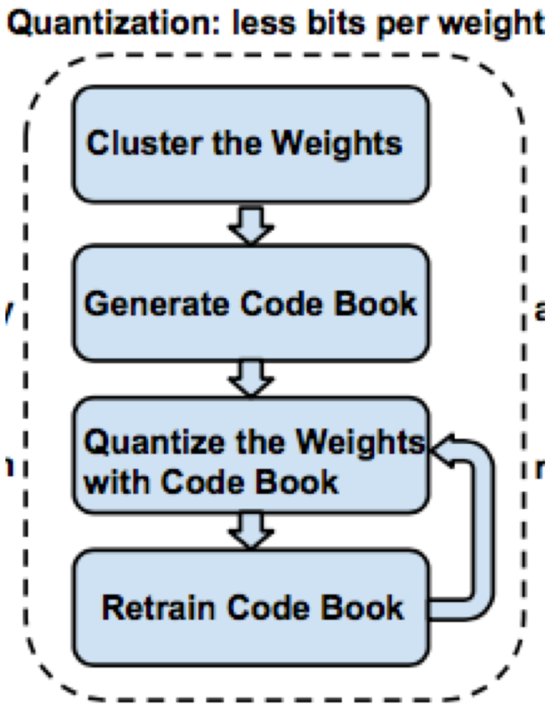
1. Run normal network training
   - Find out which connections are important
2. Prune the small-weight connections
   - L1/L2 regularization
3. Retrain the network on remaining sparse connections
   - Dropout to prevent overfitting

Weight Quantization - Weight Sharing [Han et al. ICLR

Using K-means

Using centroids to approx. weights





n = num. of weights (16)
b = num. of bit for original data (32)
k = num. of clusters (4)

$$\text{Compress rate} = \frac{nb}{n\log_2(k) + kb} = 3.2$$

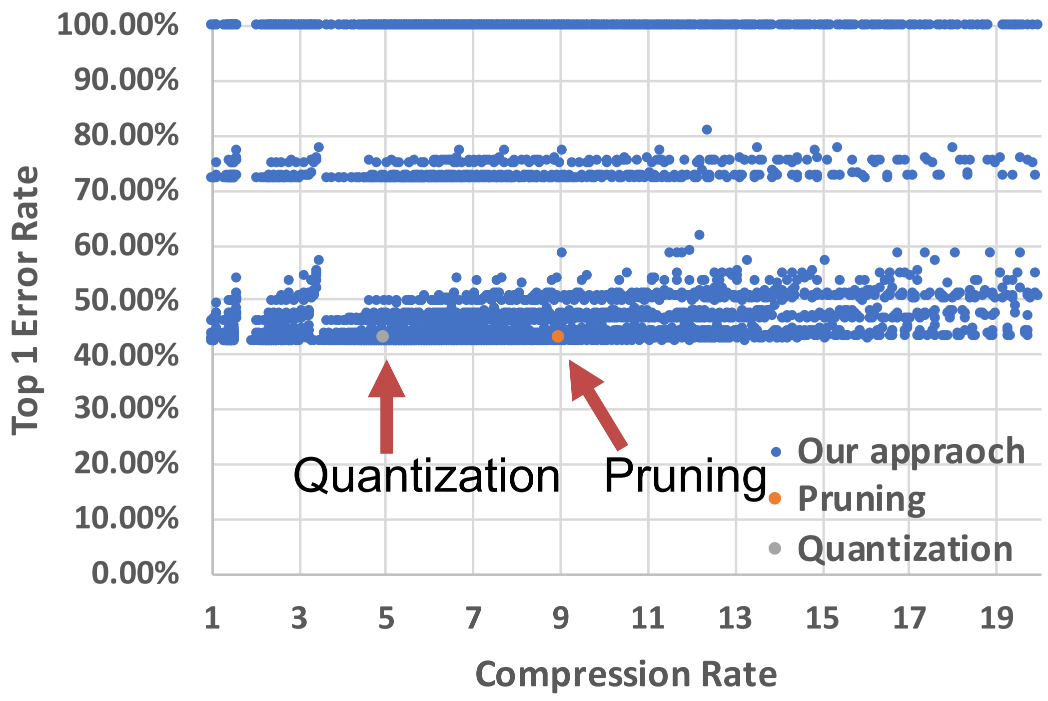
Error-bounded Compression for DNN

- Error-bounded compression can be viewed as ***generalized*** approach compared with pruning and quantization.
  - **Pruning** is a compression technique with error bound ***fixed*** at 100%.
  - **Quantization** is a compression technique with ***limited*** but ***uncontrollable*** error bound.
- SZ-2.0*: error-bound controlled lossy compressor
  - Based on multi-dimensional fitting
  - Error-bounded quantization techniques
  - Linear regression

* SZ 2.0: https://github.com/disheng222/SZ

Error-bounded Compression for DNN

- Error-bounded compression is ***adjustable*** and ***controllable***.
  - Compression strength is adjustable through *compression parameters*.
  - E.g.: Applying different compression configurations (~6500) on AlexNet *(tested on ILSVRC'12)*
- *Error-bounded compression shows promising performance.*
  - *Better compression rate.*
  - *Comparable accuracy loss.*



Self-adaptive Compression for DNN

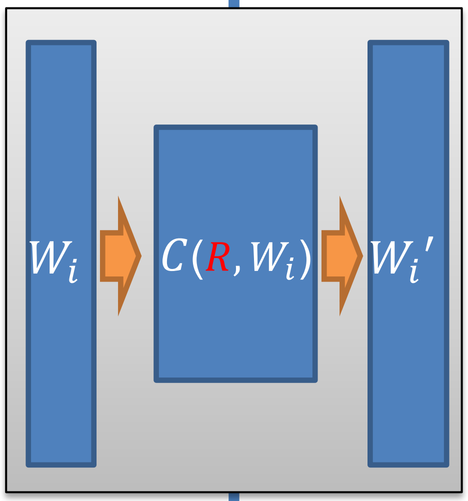
- As DNNs becoming deeper (e.g., ResNet-152) much more compression configurations need to be searched to find the best one.
- Compression configurations can grow exponentially as DNN grow:
  - Lenet-5 (~6.5K)
  - Alexnet (~4.7M)
  - VGG-16 (~$1.8*10^{15}$)
  - ResNet-152 (~$1.1*10^{145}$)
- Impractical to apply brute force search or search by hand.

Self-adaptive Compression for DNN

- We propose self-adaptive compression for DNN
  - Compression parameters (R) : ***Hyper-parameter*** → ***Learnable parameter***
  - Learn compression parameters as if they are network parameters.

Froward propagation: Evaluate accuracy loss and size of network after applying compression.

① Pushes R → weaker compression
② Pushes R → stronger compression

… 

… = $F_{loss}(W) + \alpha * compressed\ size\ (CS)$

① $\frac{\partial R}{\partial F_{loss}}$ ② $\frac{\partial R}{\partial CS}$

Add compression layer
(compress first, then decompress)

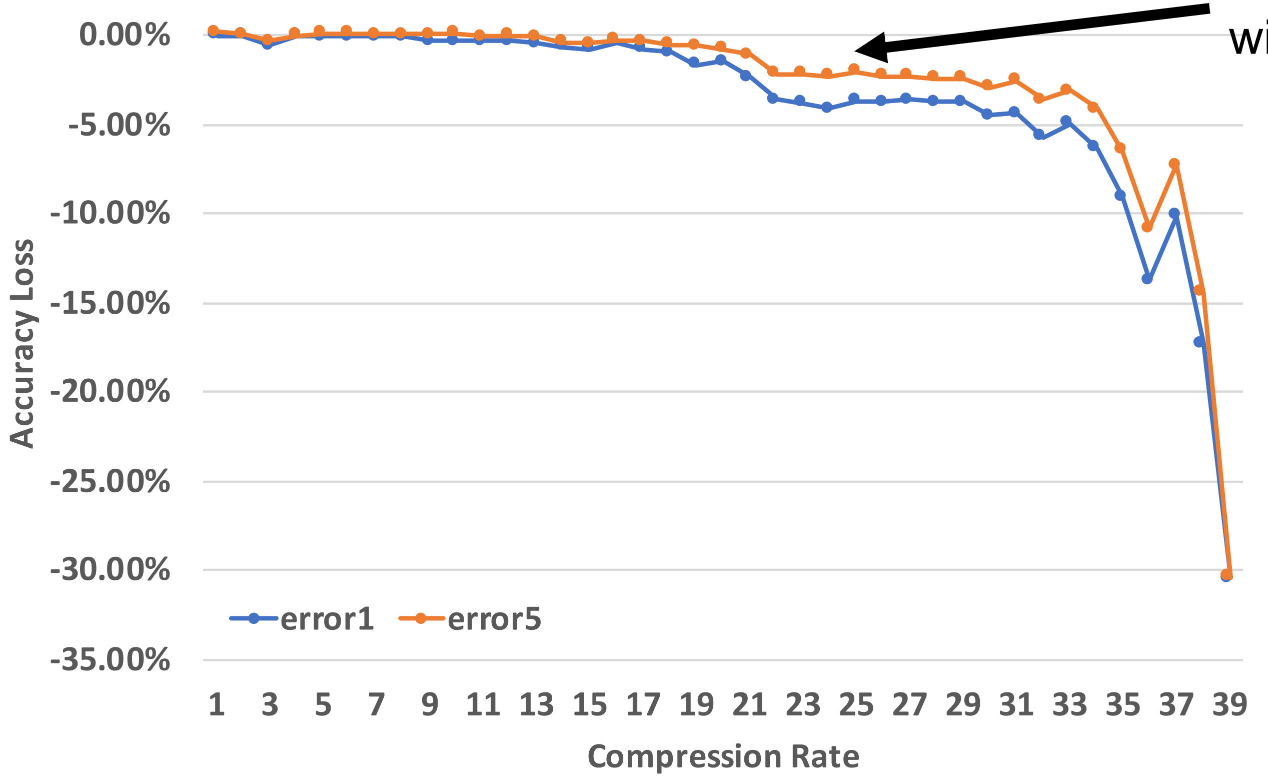
Back propagation: Adjust compression parameters based on accuracy loss and size of network

Customized SGD:
For each layer: Derive ① based on F_{loss} Derive ② based on CS Update R = R – lr * (① + ②)

Self-adaptive Compression for DNN

Pacific Northwest
NATIONAL LABORATORY

- We propose self-adaptive compression for DNN
  - Compression parameters (R) : ***Hyper-parameter*** → ***Learnable parameter***
  - Learn compression parameters as if they are network parameters.



Adaptive compression result will be one of those points

Trade-off between accuracy loss and compression rate

Thanks!

- Questions?