Training 4

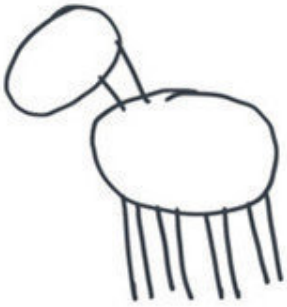
CS 217 Stanford

Ardavan Pedram

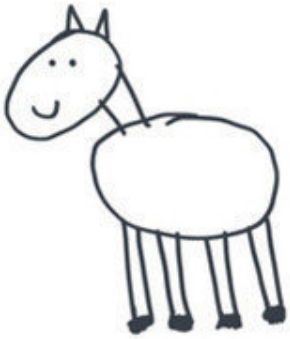
HOW TO:
DRAW A HORSE

BY VAN OKTOP

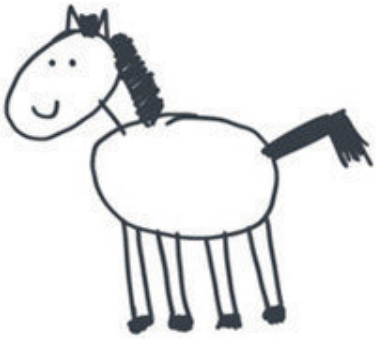
(1) DRAW 2 CIRCLES



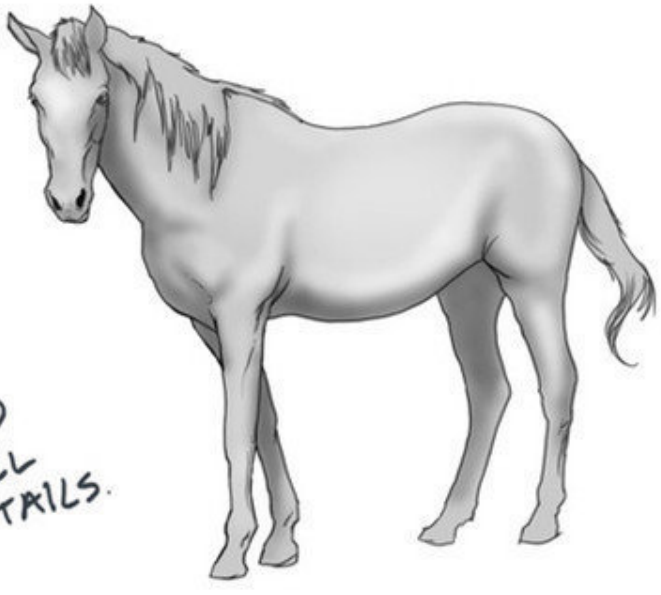
(2) DRAW THE LEGS



(3) DRAW THE FACE



(4) DRAW THE HAIR



(5) ADD SMALL DETAILS.

Training

1. Fundamentals of training
2. Optimizing learning
3. Efficient training
4. Scaling training

Training

1. Fundamentals of Training
   - Gradient Descent
   - Backpropagation
   - Stochastic Gradient Descent
   - Mini Batch Gradient Descent
2. Optimizing learning
   - Momentum
   - Learning Rate
   - Misc.
3. Efficient training
   - Low Precision
   - Volta 100
4. Scaling training
   - **Sparsity**
   - Scaling Computation power
   - Scaling Network Size
   - Large Batch Training
   - Asynchronous Training

Architectural Attributes

- Parallelism
- Precision, Quantization
- Sparsity

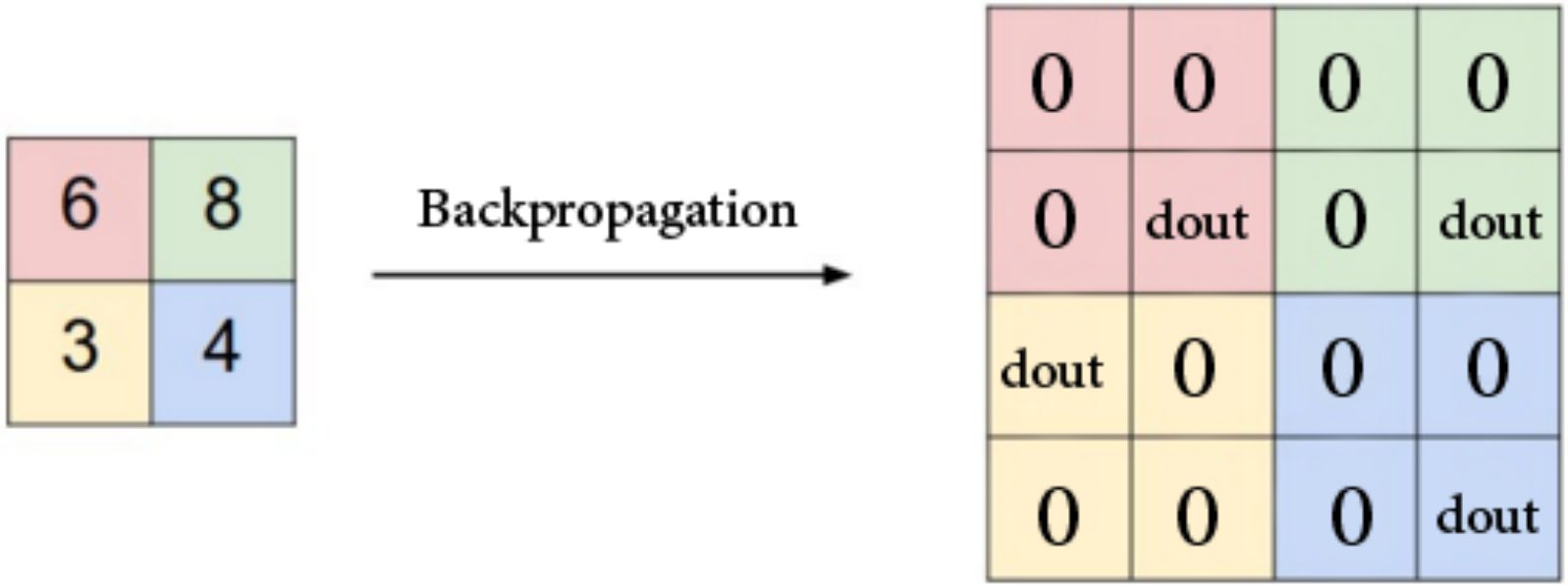
- **Parallelism**
  - ❖ Data parallelism
    - ○ Coarse grain
      - ❖ Mini-batch size
      - ❖ Amortizing the cost of communication latency
    - ○ Fine grain
      - ❖ SIMD
  - ❖ Model parallelism
    - ○ Granularity of network chunks

Sparsity

- Activation sparsity
  - Forward Pass ?
  - Backward Pass?

- Weight sparsity
  - Where is this useful?

Pooling Layer

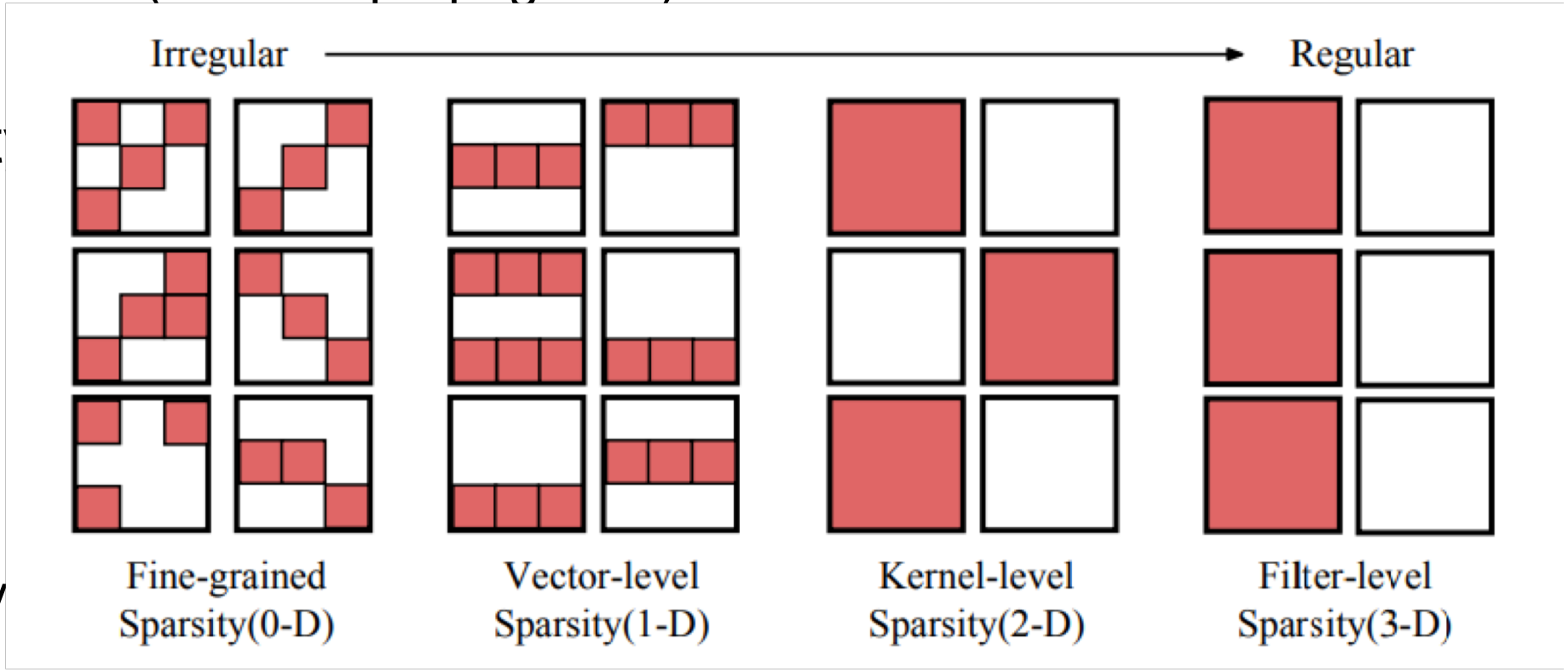


In the process we replace the maximum values before max pooling with 1 and set all the non maximum values to zero then use the **Chain rule** to multiply the gradient by them.

https://mc.ai/demystifying-convolutional-neural-networks/

Sparsity Has Many Flavors

- Activation Sparsity
  - RELU / MAXPOOL (on back propagation)
- Weight Sparsit
  - Fine grain
  - Per row
  - Per column
  - Per kernel
  - Per channel
  - Per filter
  - Block sparsity



Huizi Mao et. al https://arxiv.org/abs/1705.08922

RNNs & CNNS Benefit from Structured Sparsity

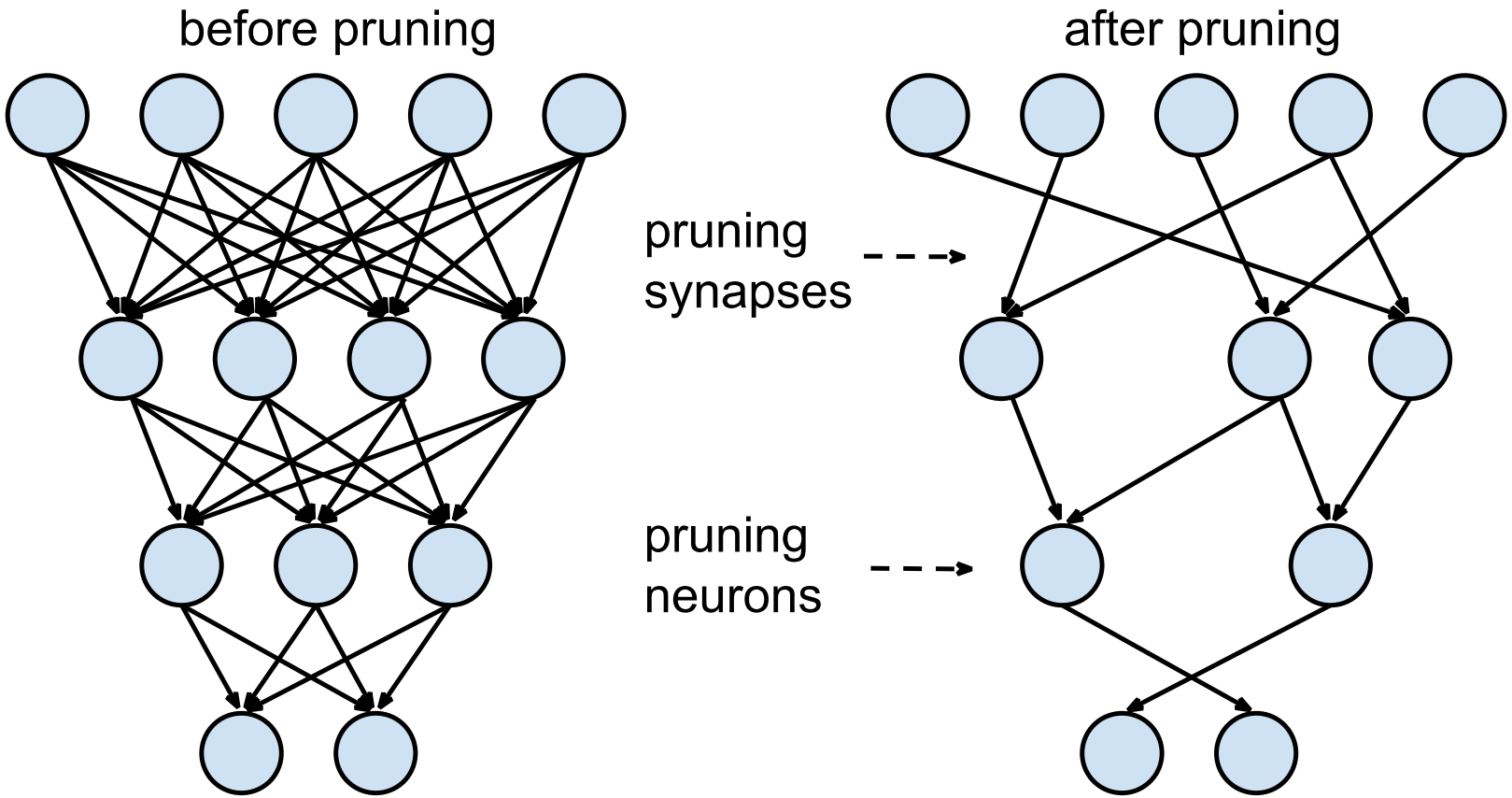
- RNNs [1]
  - 90% sparsity reduces relative accuracy by 10% to 20%
  - Solution: Make the sparse model larger
  - Large sparse model still have less parameters compared to the small dense baseline and achieves a slight increase in accuracy

- CNNs [2]
  - Pruning with large granularity will greatly hurt accuracy
  - Due to index savings, coarse-grain pruning can still achieve space savings even at a lower overall sparsity

[1]- Sharan Narang, Erich Elsen, Gregory Diamos, Shubho Sengupta, "Exploring Sparsity In Recurrent Neural Network", ICLR 2017. https://arxiv.org/abs/1704.05119

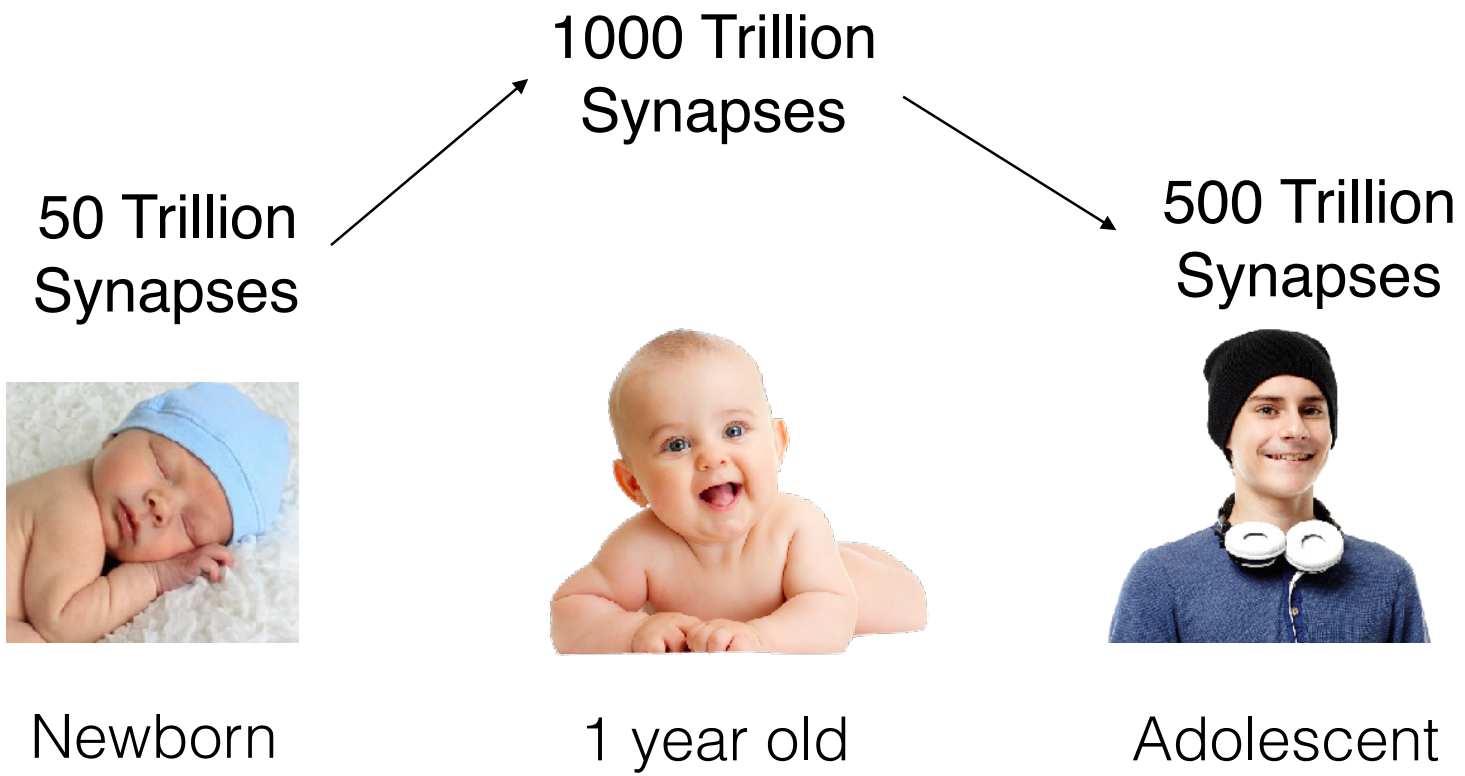
[2]- Huizi Mao, Song Han, Jeff Pool, Wenshuo Li, Xingyu Liu, Yu Wang, William J. Dally, "Exploring the Granularity of Sparsity in Convolutional Neural Networks" CVPR'17 TMCV workshop. https://arxiv.org/abs/1705.08922

Pruning The Network



Source: Song Han

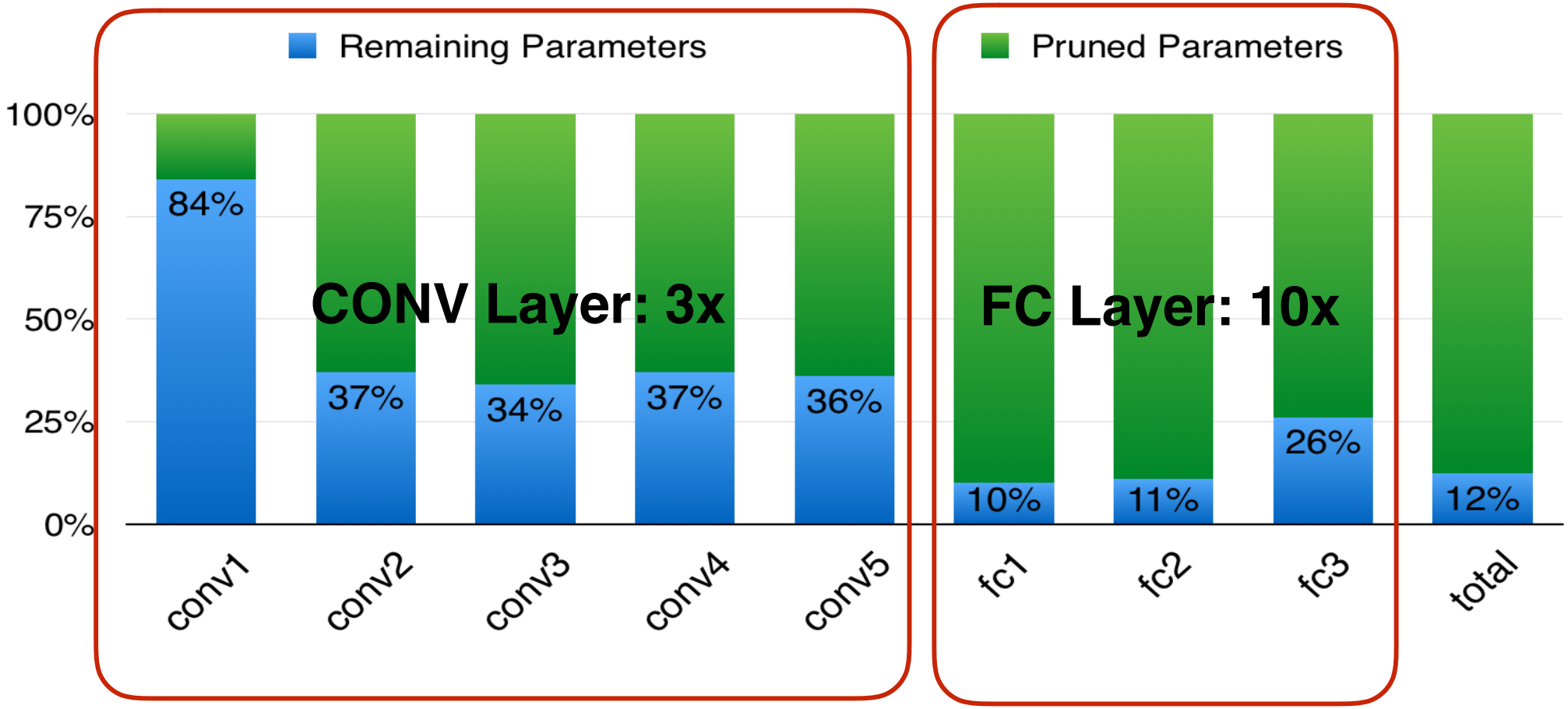
Human's Learn by Pruning



Christopher A Walsh. Peter Huttenlocher (1931-2013). Nature, 502(7470):172–172, 2013.

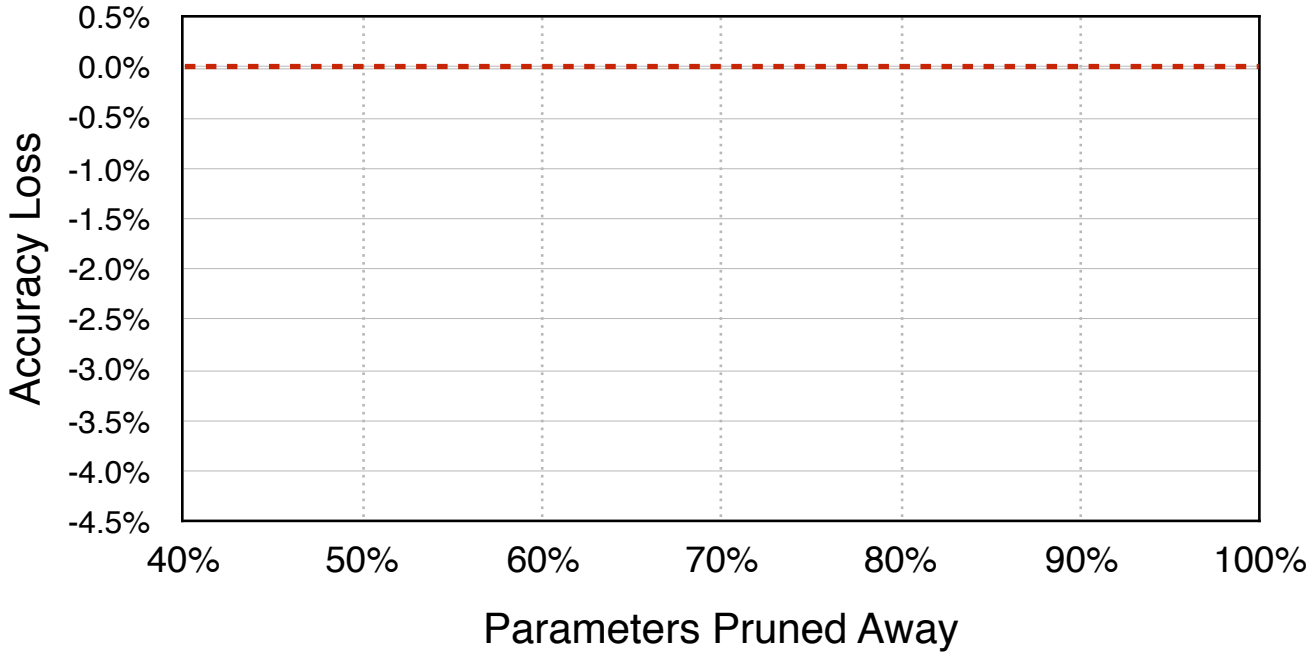
Source: Song Han

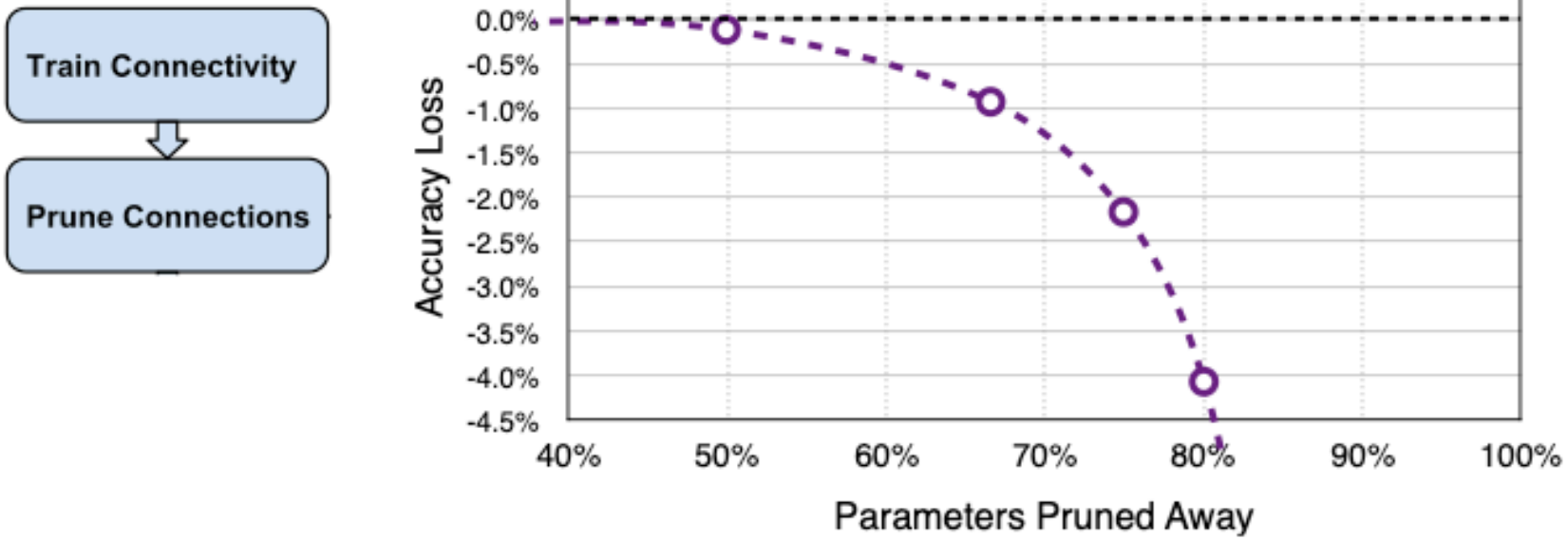
Alexnet Pruning



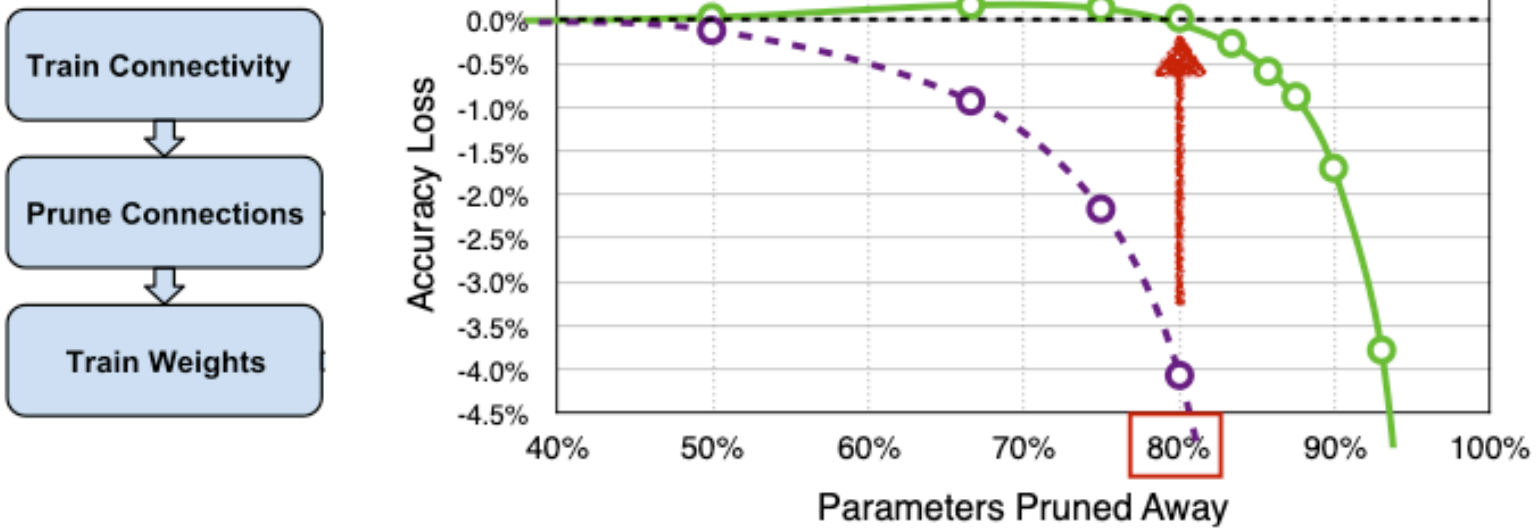
Source: Song Han

Train Connectivity



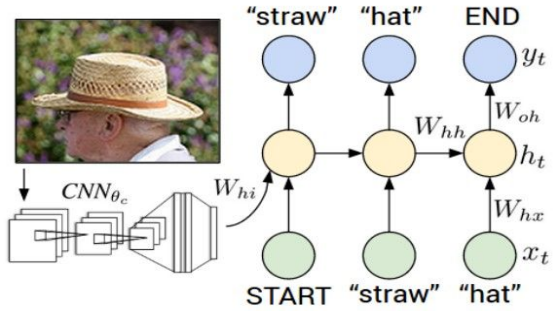


Source: Song Han

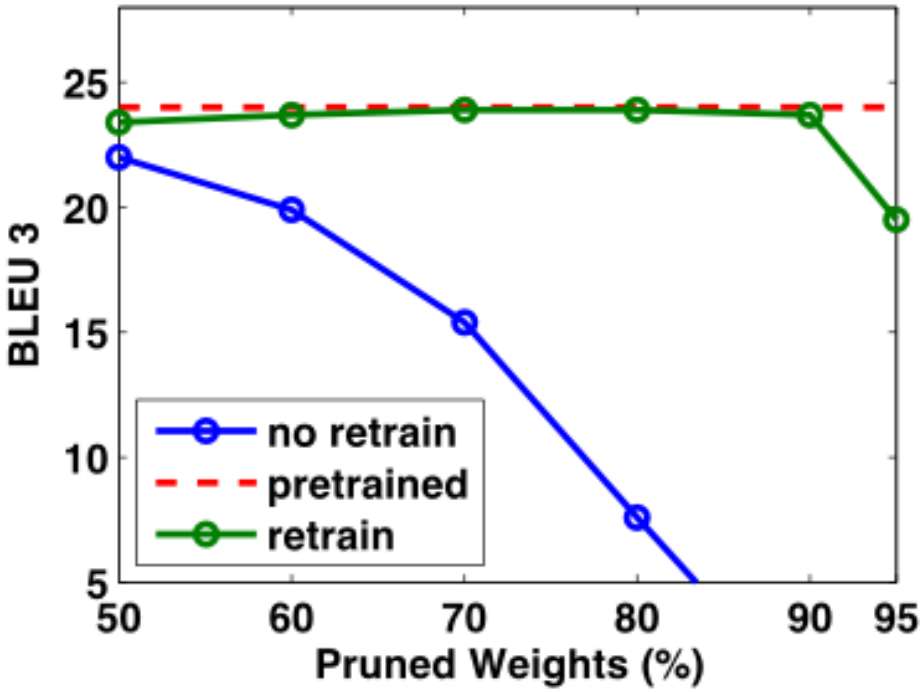


Source: Song Han

Pruning RNN and LSTM

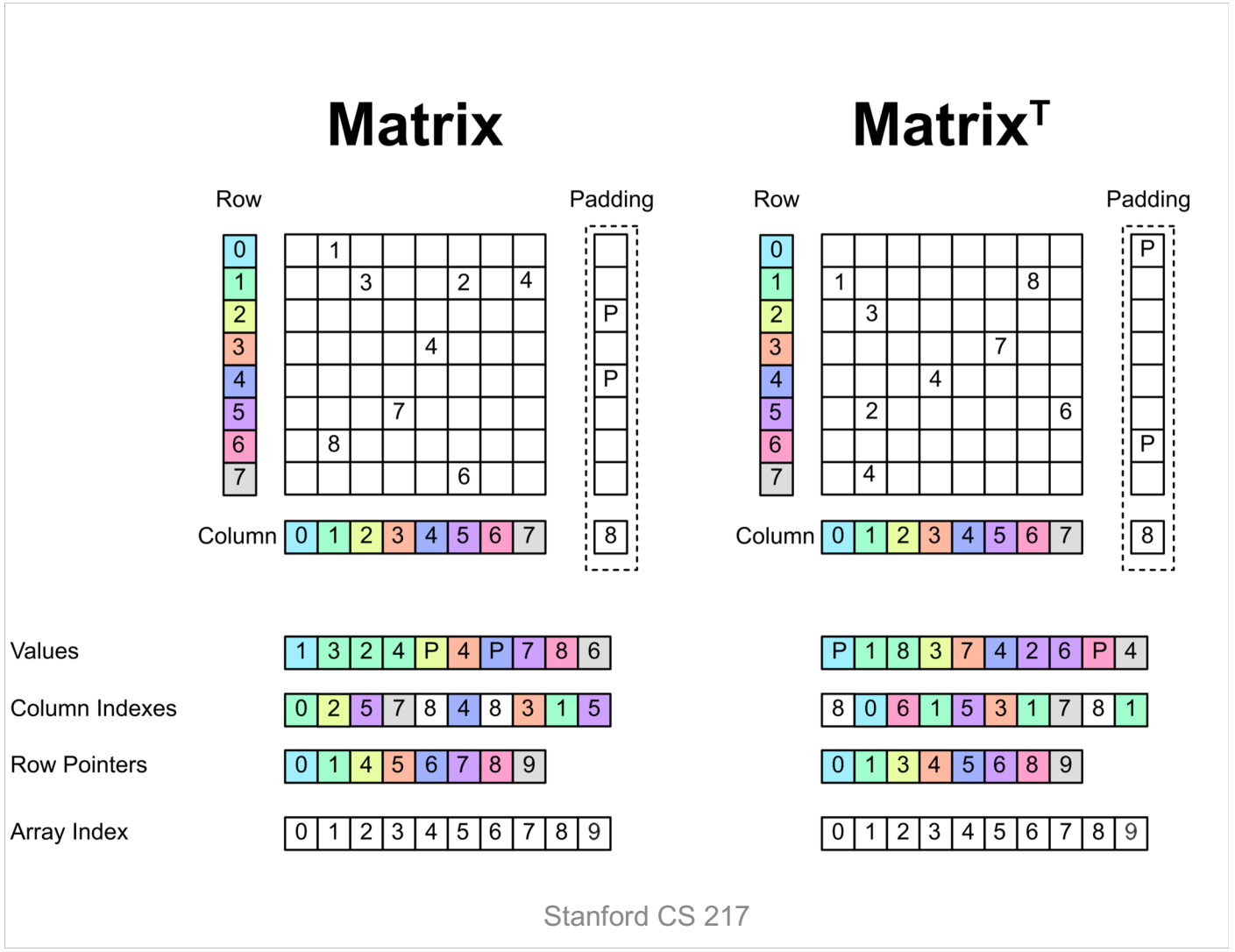


*Karpathy et al "Deep Visual-Semantic Alignments for Generating Image Descriptions"



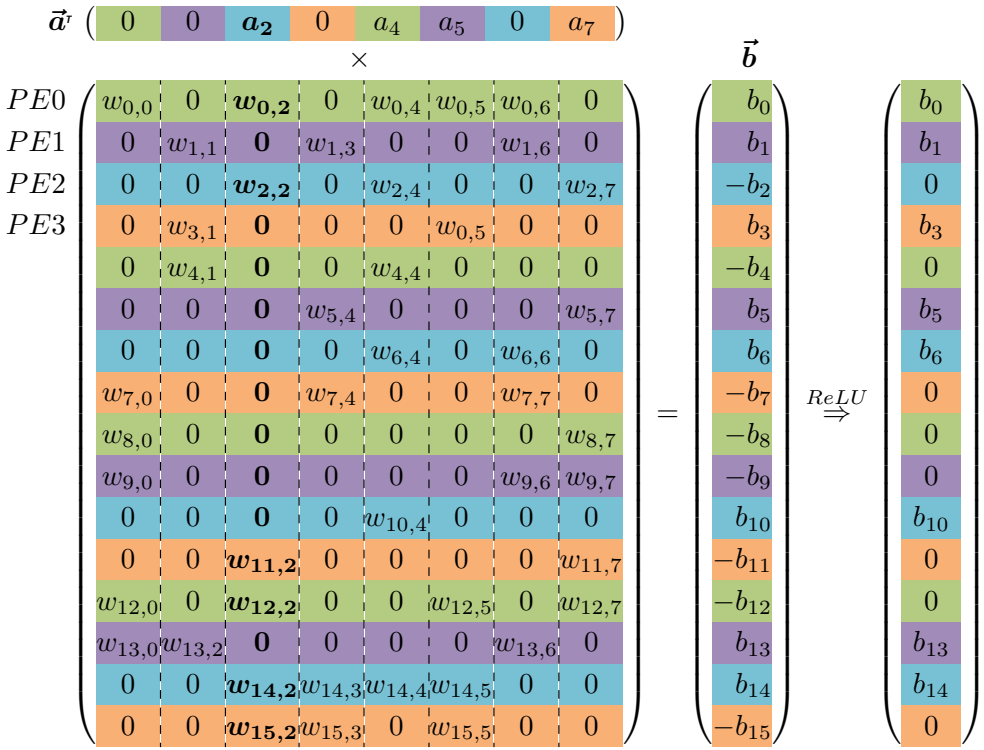
Quantization

Sparse Compressed Formats



Sparse Matrix Vector Multiplication for FC Layers

EIE [Han, 2016]

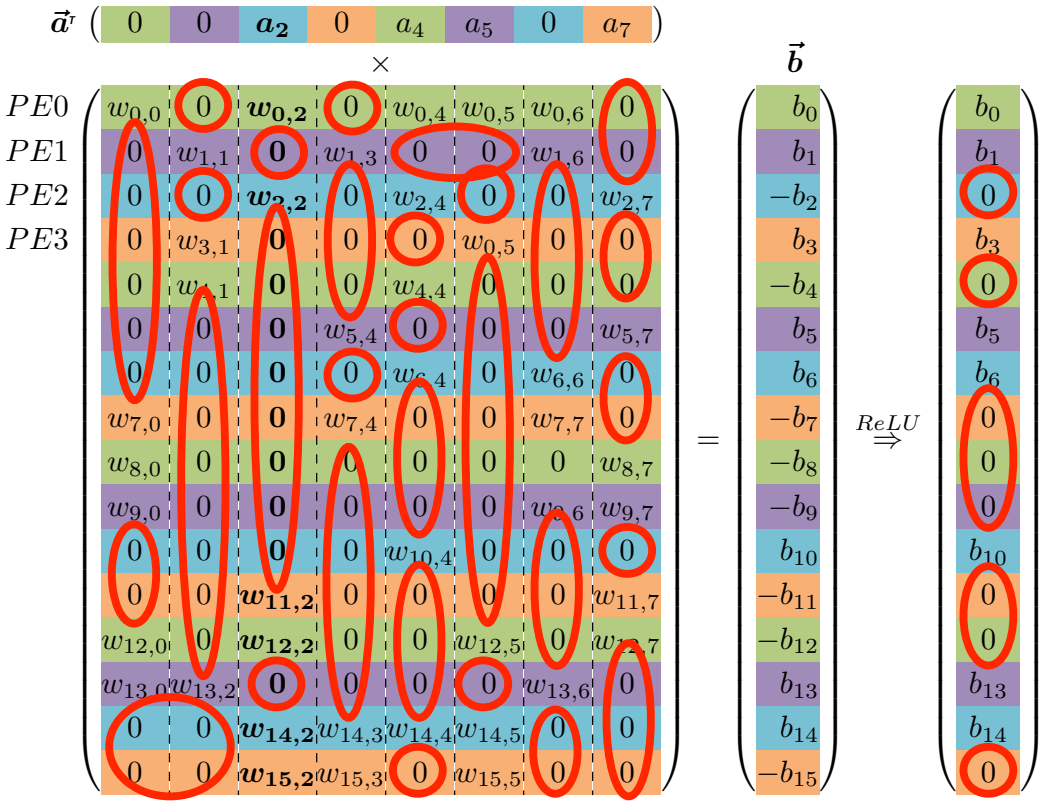


Virtual Weight	$W_{0,0}$	$W_{8,0}$	$W_{12,0}$	$W_{4,1}$	$W_{0,2}$	$W_{12,2}$	$W_{0,4}$	$W_{4,4}$	$W_{0,5}$	$W_{12,5}$	$W_{0,6}$	$W_{8,7}$	$W_{12,7}$
Relative Row Index	0	1	0	1	0	2	0	0	0	2	0	2	0
Column Pointer	0	3	4	6	6	8	10	11	13				

Save Memory by Deep Compression
10x-30x BW saving

Distributed SPMV

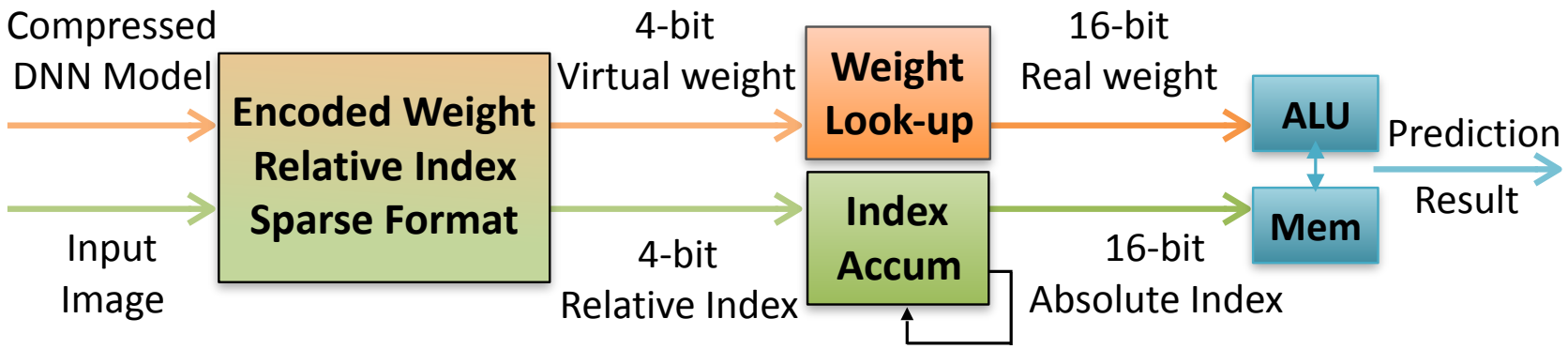
EIE [Han, 2016]

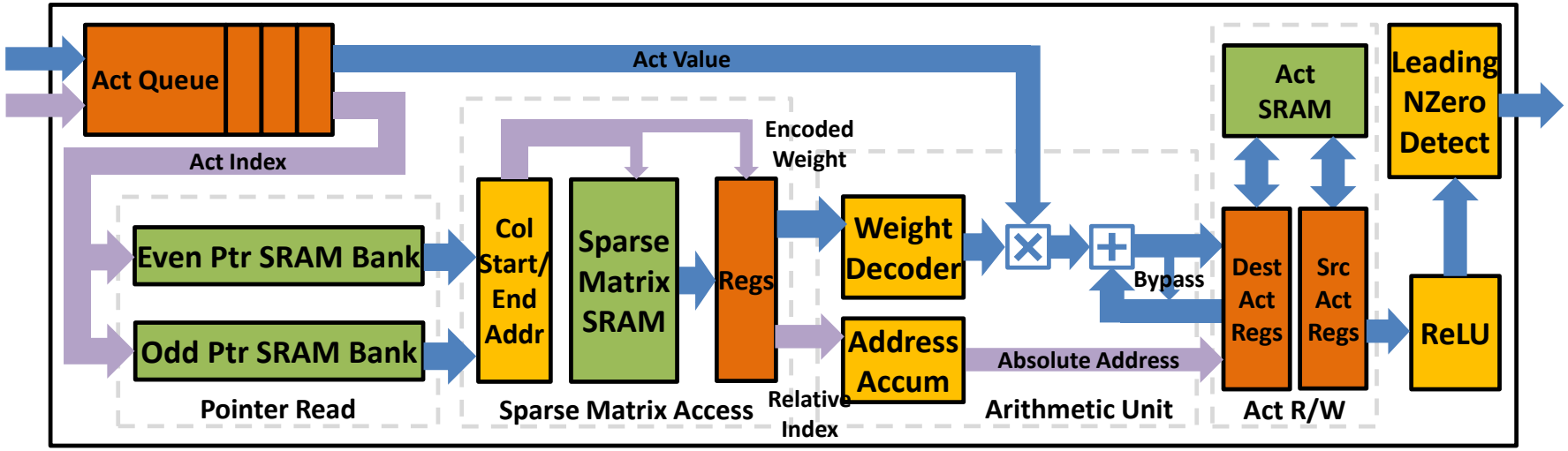


Virtual Weight	$W_{0,0}$	$W_{8,0}$	$W_{12,0}$	$W_{4,1}$	$W_{0,2}$	$W_{12,2}$	$W_{0,4}$	$W_{4,4}$	$W_{0,5}$	$W_{12,5}$	$W_{0,6}$	$W_{8,7}$	$W_{12,7}$
Relative Row Index	0	1	0	1	0	2	0	0	0	2	0	2	0
Column Pointer	0	3	4	6	6	8	10	11	13				

Save Memory by Deep Compression saving 10x-30x BW

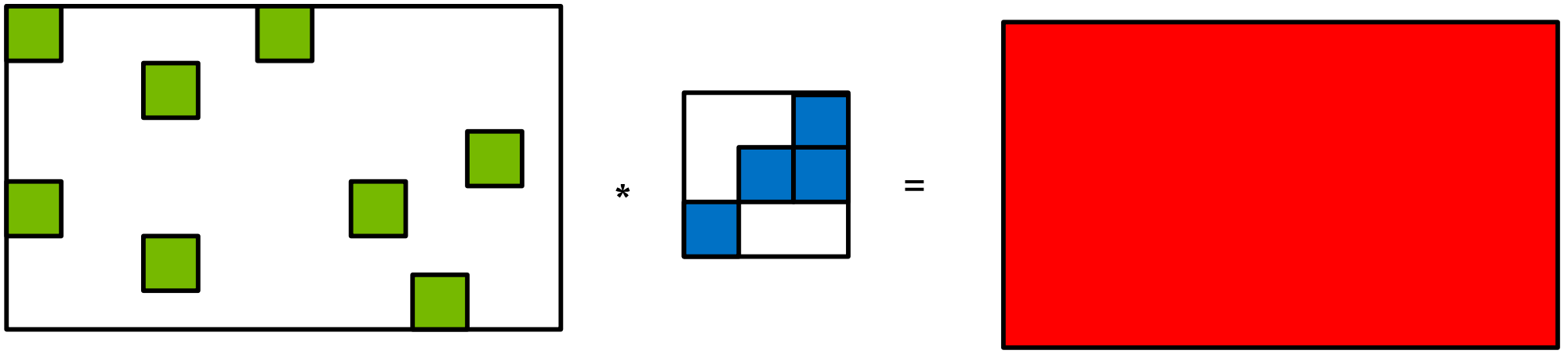
Track Several Pointers

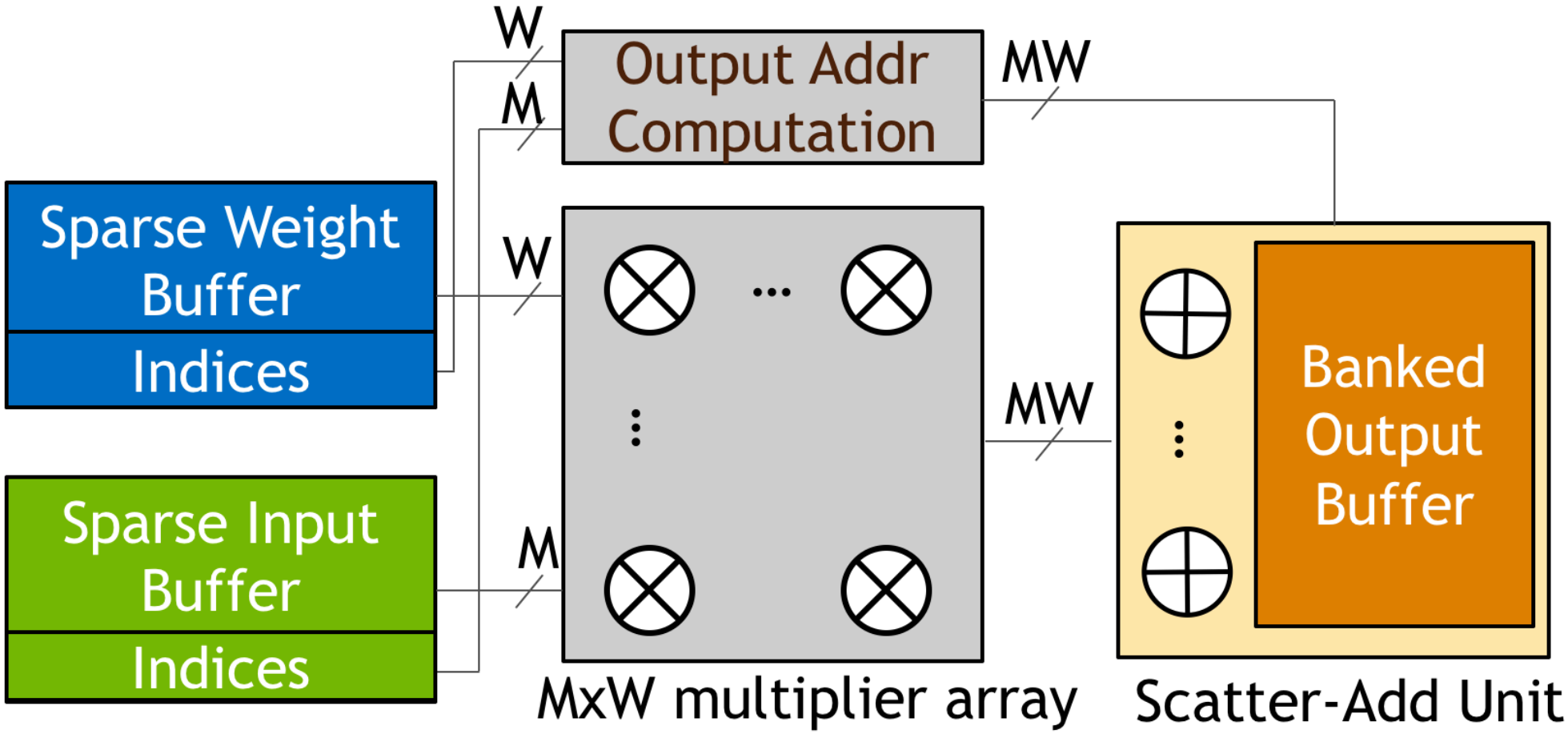




SCNN

- Only compute where both operands are nonzero




W
M
Output Addr Computation
MW
Sparse Weight Buffer
Indices
W
Sparse Input Buffer
Indices
M
MW
Banked Output Buffer
MxW multiplier array
Scatter-Add Unit

What about an off the shelf system?

- What is the problem with sparsity methods?
  - GPUs?
  - CPUs?

- Remember EIE works for FC Layers

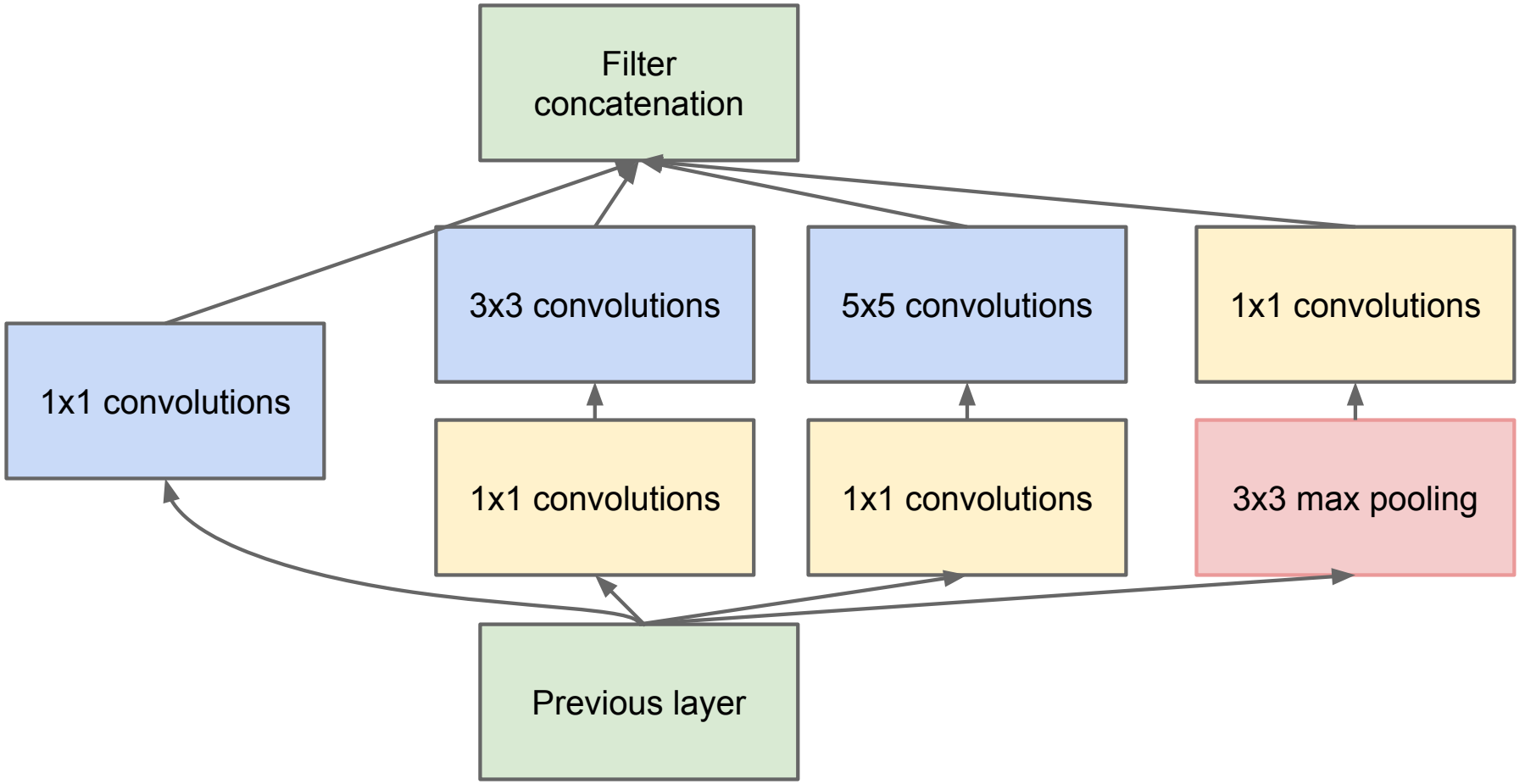
What is The Problem?

- Today's computing infrastructures are very inefficient when it comes to numerical calculation on non-uniform sparse data structures
- The overhead of lookups and cache misses would dominate: switching to sparse matrices might not pay off
  - Even if the number of arithmetic operations is reduced by 100×

How About Large Grain Sparsity?

- A Network architecture that makes
  - Uses Filter-level sparsity,
  - Exploits our current hardware by utilizing computations on dense matrices
- Cluster Sparse Matrices into Dense Sub-blocks

GoogLeNet's Inception Module



The Dark Side of DNN Pruning

- Does it work everywhere?
  - it may reduce the confidence of DNN predictions.

The Dark Side of DNN Pruning

- Automatic Speech Recognition (ASR)

- Although top-1 accuracy may be maintained with DNN pruning, the likelihood of the class in the top-1 is significantly reduced when using the pruned models.

ASR: Automatic Speech Recognition Pipeline

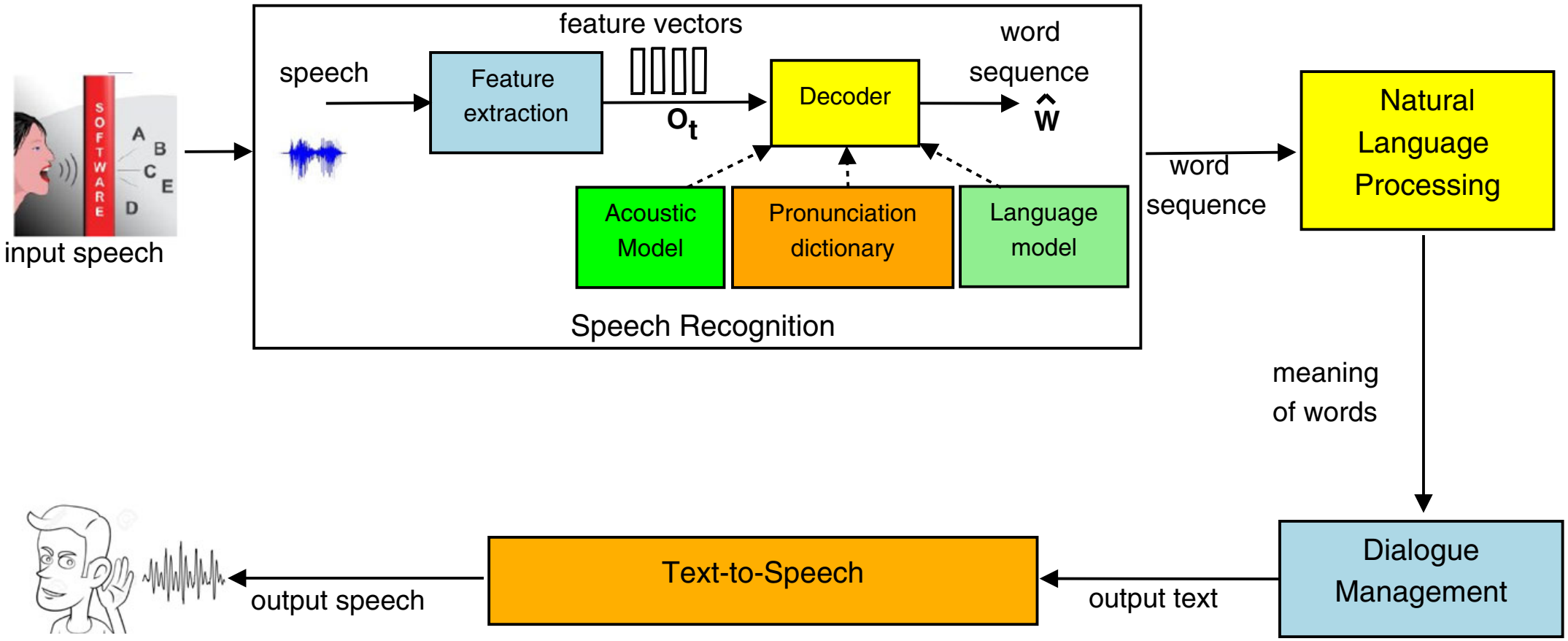
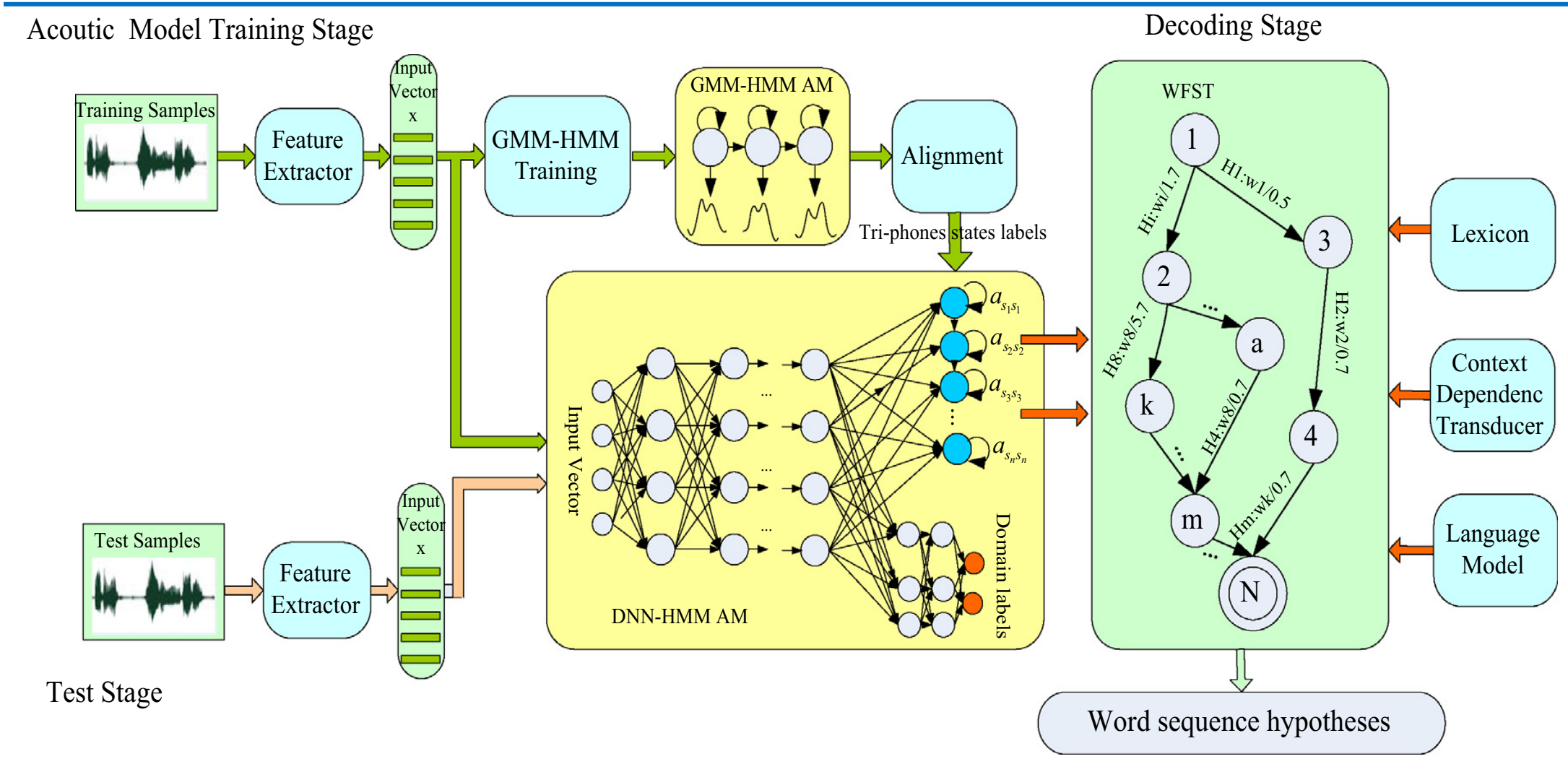


Fig. 1 Spoken dialog system and the components of the speech recognition module

ASR Cont.


Acoutic Model Training Stage
Decoding Stage
Training Samples
Feature Extractor
Input Vector x
GMM-HMM Training
GMM-HMM AM
Alignment
Tri-phones states labels
Input Vector
DNN-HMM AM
Domain labels
Test Samples
Feature Extractor
Input Vector x
Test Stage
WFST
Lexicon
Context Dependenc Transducer
Language Model
Word sequence hypotheses

ASR Cont.

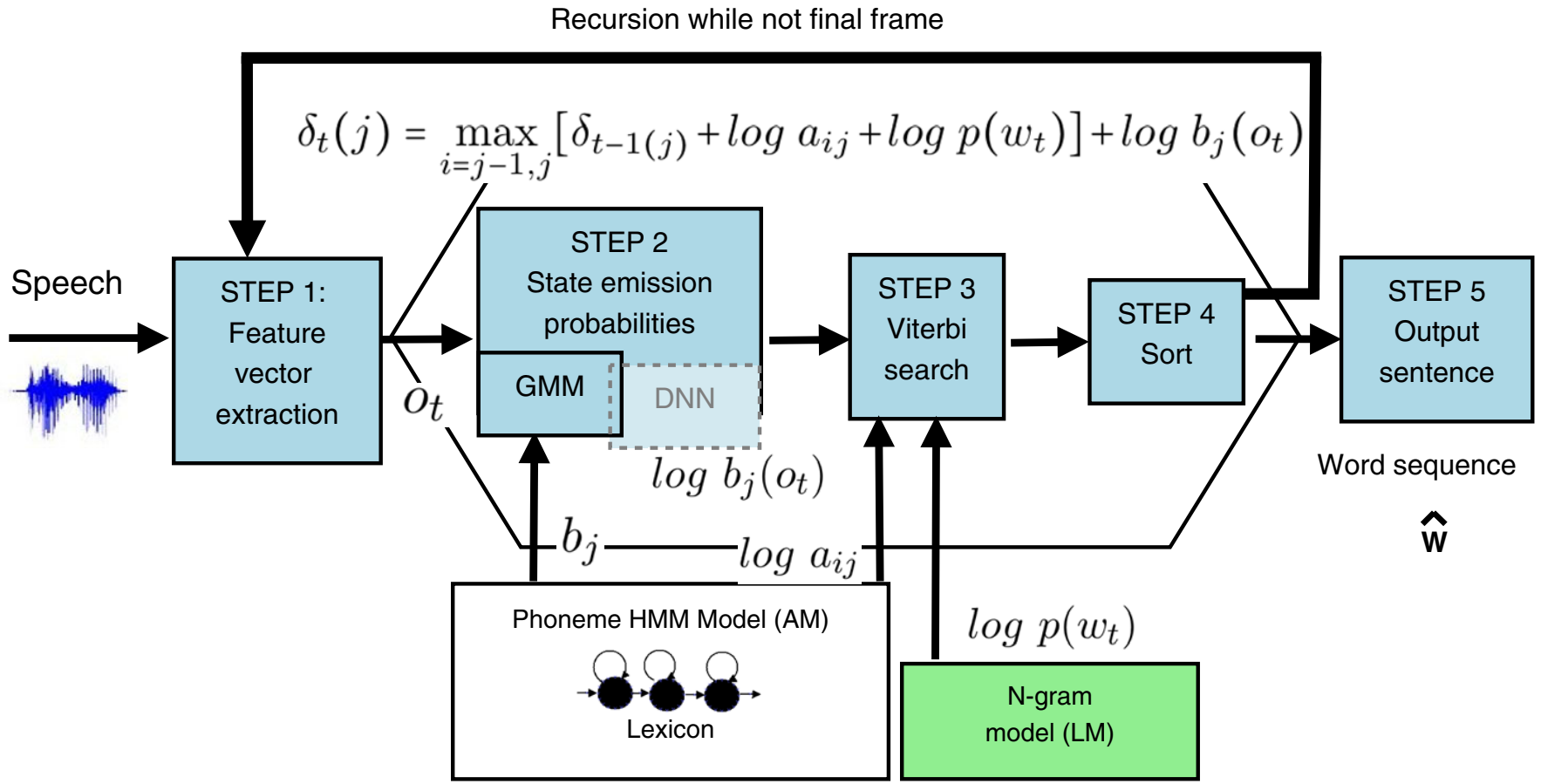


Fig. 6 Speech recognition flow with GMM-HMM

The Dark Side of DNN Pruning

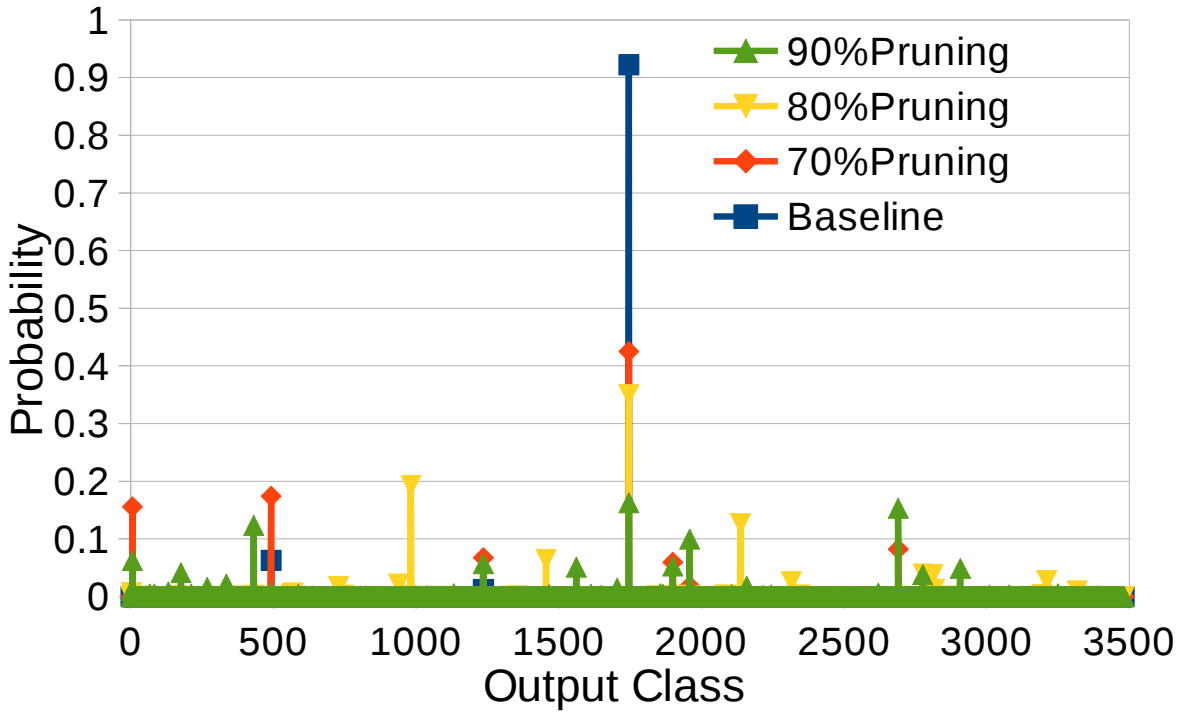


Fig. 1. Distribution of scores for a DNN for speech recognition and three pruned version at 70%, 80% and 90% of pruning. Although pruned models correctly identify top-1 class, the distribution of likelihoods is severely affected and confidence is largely reduced.

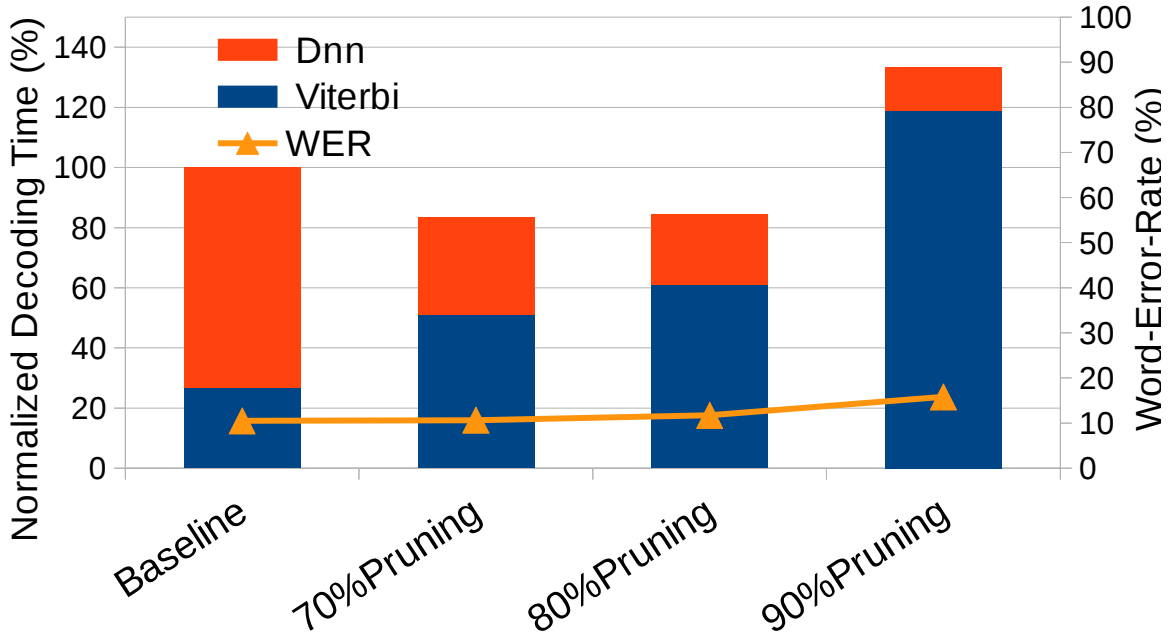


Fig. 2. Normalized execution time and Word Error Rate (WER) for a state-of-the-art ASR system, using a DNN with different degrees of pruning: 0% (Baseline), 70%, 80% and 90%. Pruning has a large impact on the execution time of the Viterbi beam search.

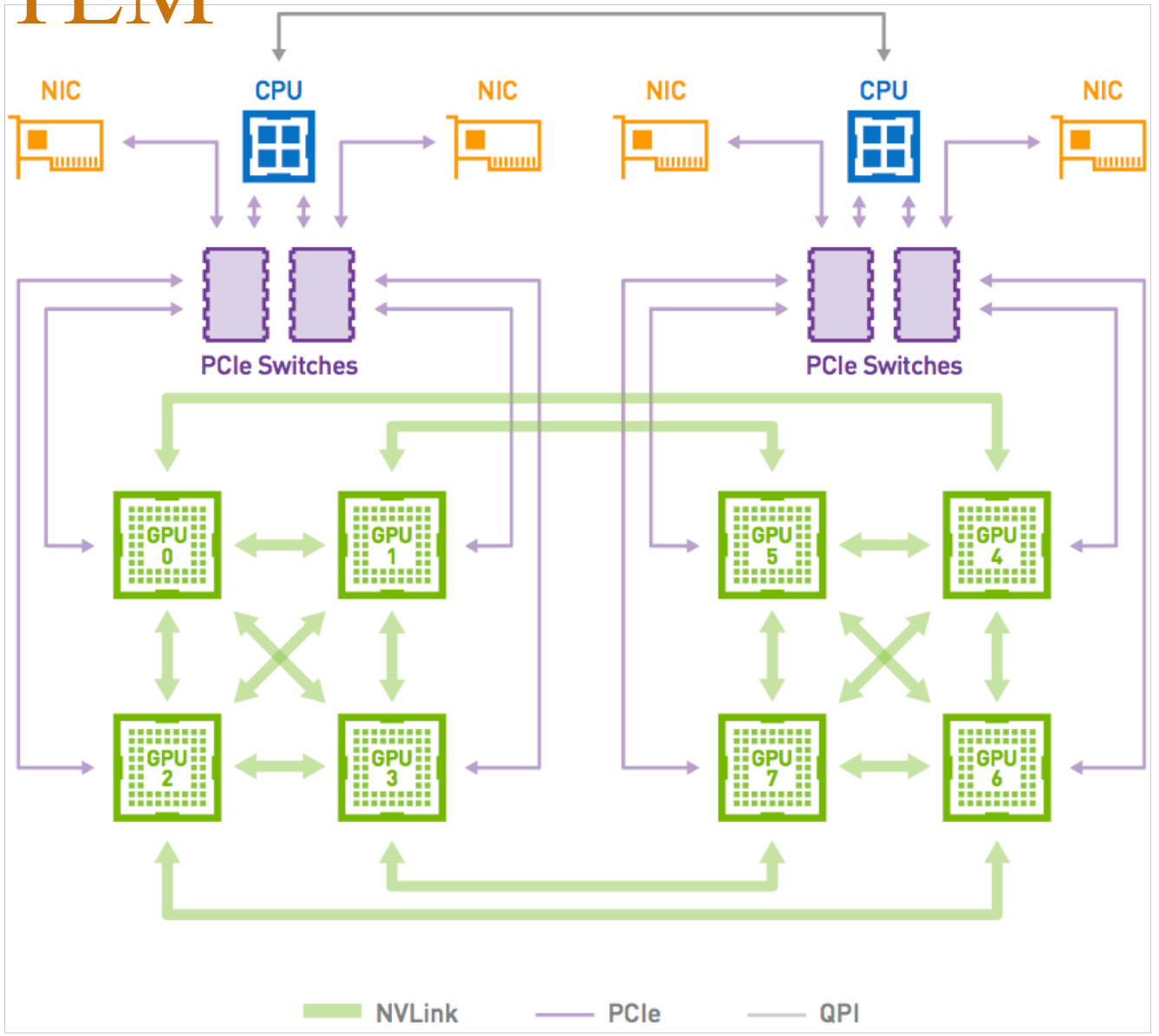
Sparsity

1. Activation
2. Weight
3. Network Topology

Scaling of Training

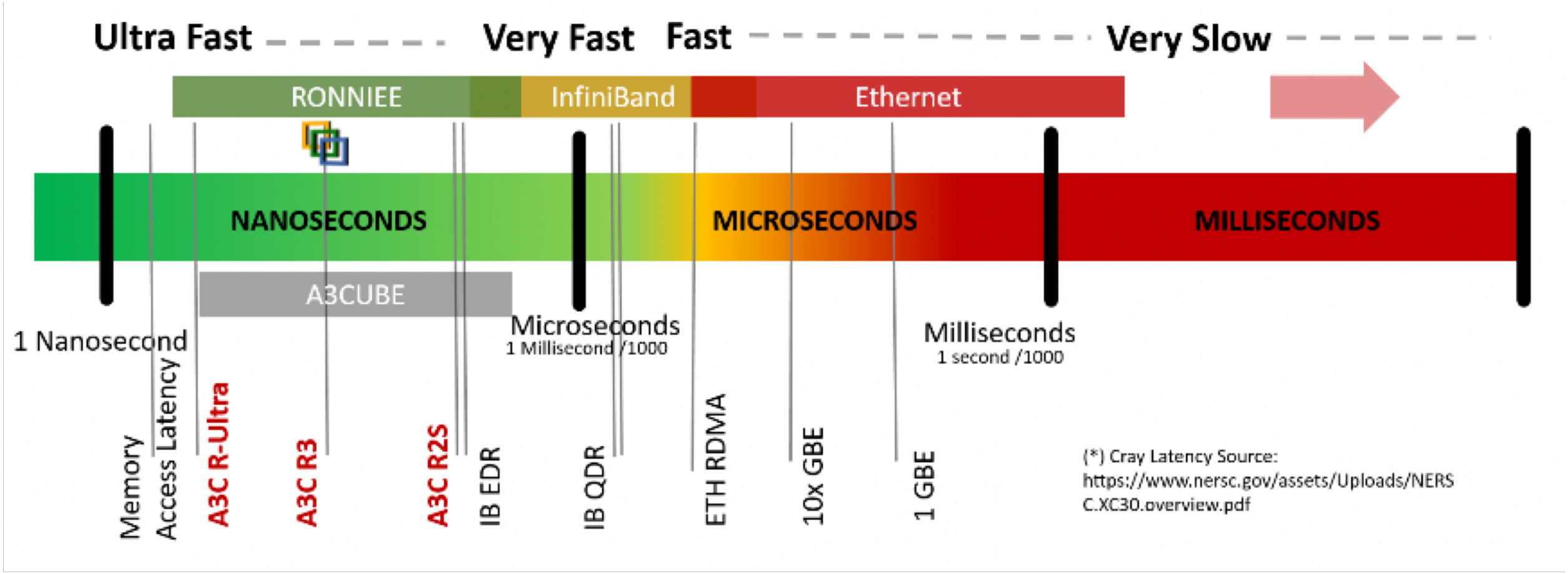
- Scaling the problem: Same system, bigger network
  - Memory bottleneck
  - Cost of computation vs. communication
- Scaling the system: Bigger system
  - Synchronization bottleneck
  - Data communication on the cloud
  - Cloud scale synchronized SGD
  - Asynchronous SGD

SCALING THE SYSTEM



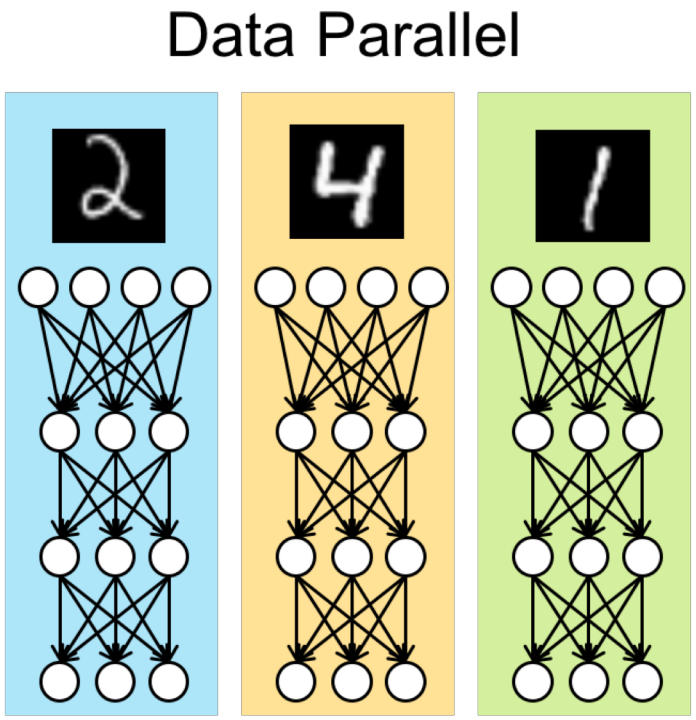
Mark Harris, “NVIDIA DGX-1: The Fastest Deep Learning System”, April 2017.

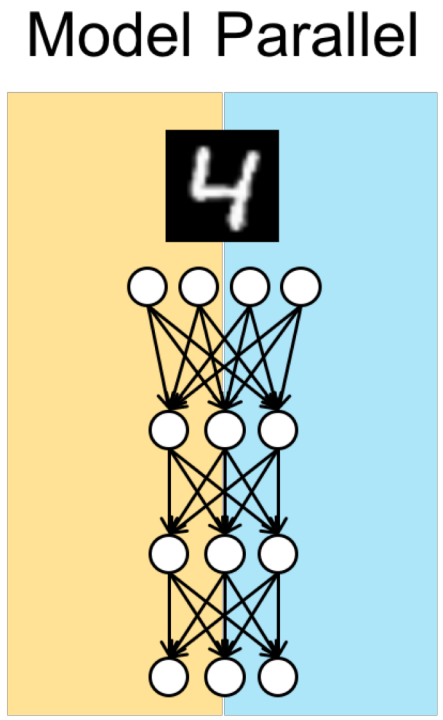
Latency IS A Bottleneck

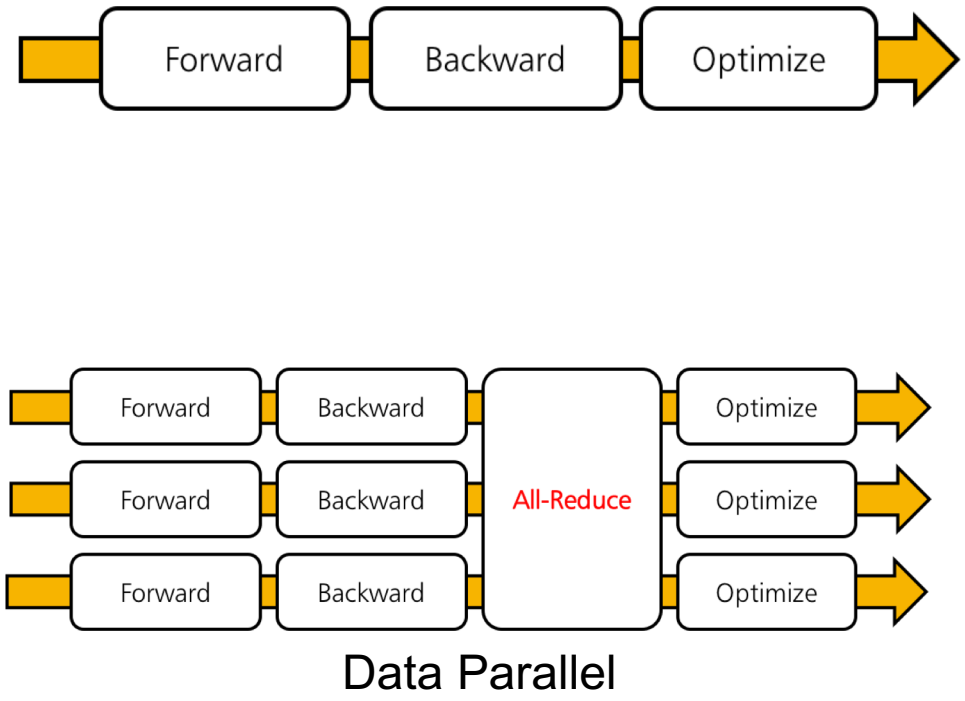


A3CUBE: “Latency Matters”, © A3CUBE INC
http://www.a3cube-inc.com/-latency-matters.html

Data vs. Model Parallelism



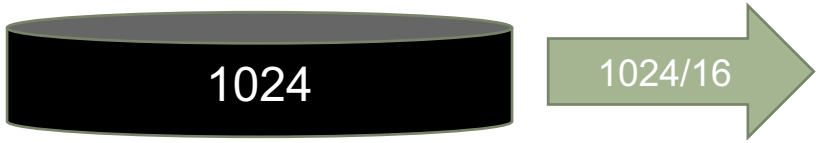


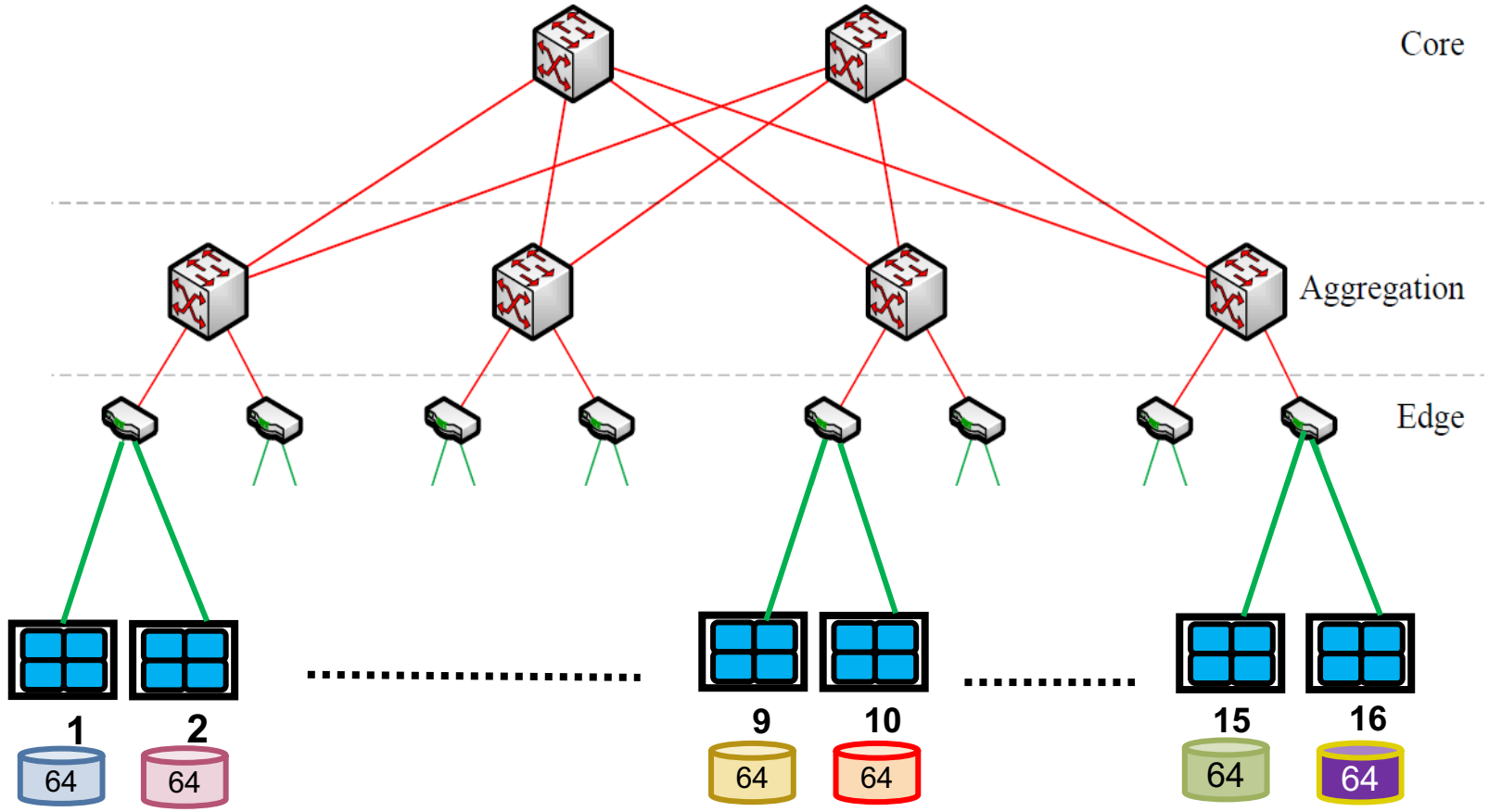


Distributed SGD Exploits data Parallelism

- Entire model on each processor
- Distribute the SGD batch evenly across each processor

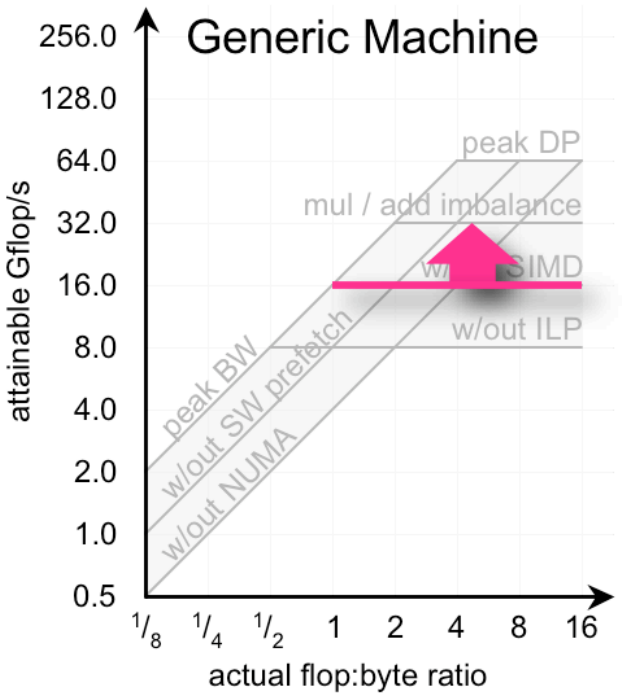
  (aka per-processor batch):
  - 1024 batch distributed over 16 PEs
  - Batch of 64/PE
- Communicate gradient updates all-to-all





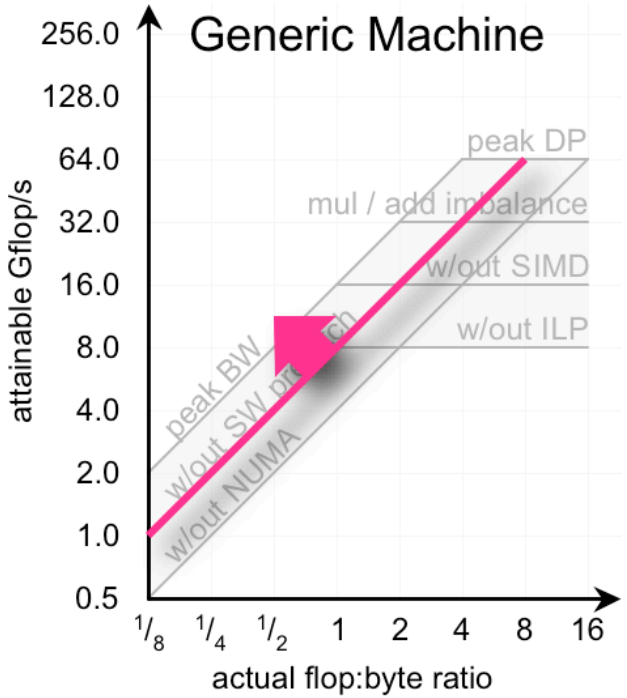
(Data center picture from)
Mohammad Al-Fares, Alexander Loukissas, Amin Vahdat, "A scalable, commodity data center network architecture" ACM SIGCOMM 2008 conference on Data communication.

Roofline Model For Distributed Training



(a) maximizing in-core performance

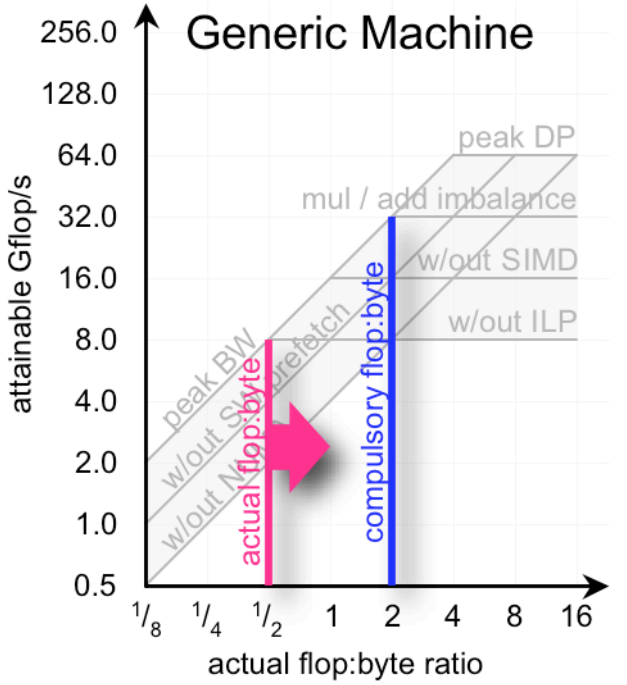
Use accelerators



(b) maximizing bandwidth

Interconnect network BW

DRAM BW



(c) minimizing traffic

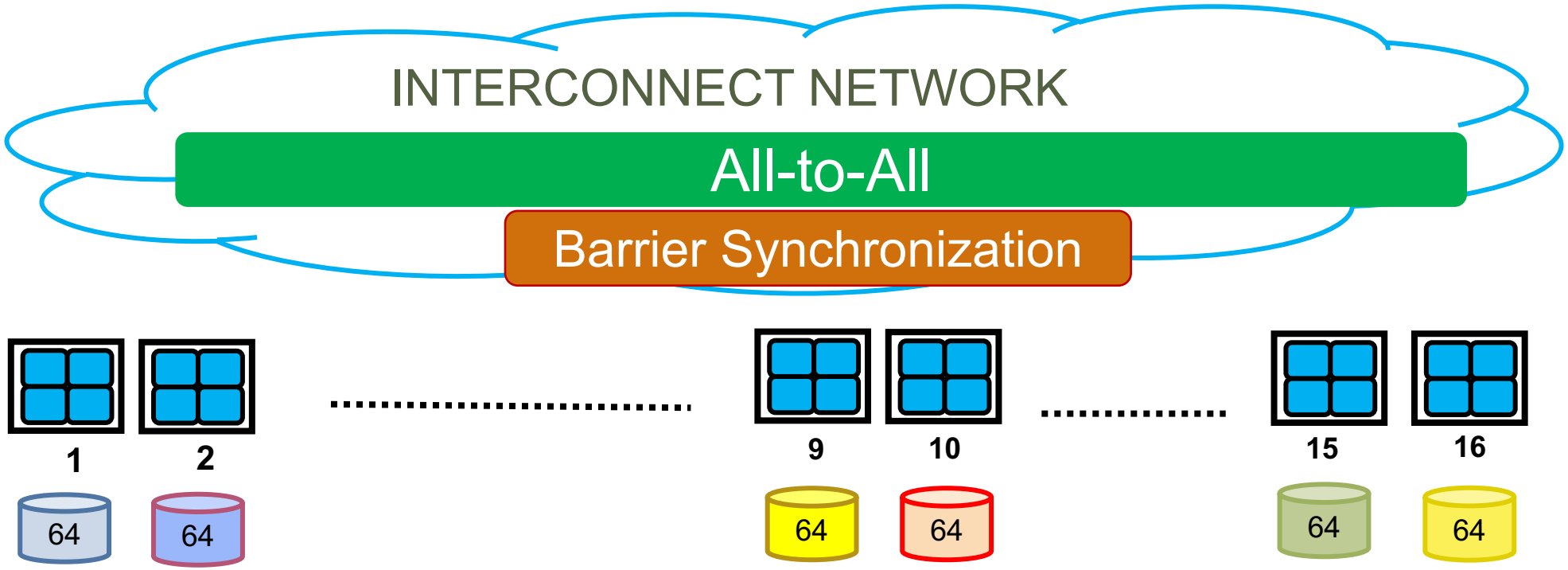
Node locality

Exploit sparsity

Less comm / synch

Samuel Williams, Andrew Waterman, and David Patterson, "Roofline: An Insightful Visual Performance Model for Floating-Point Programs and Multicore Architectures". Communications of the ACM April 2008.

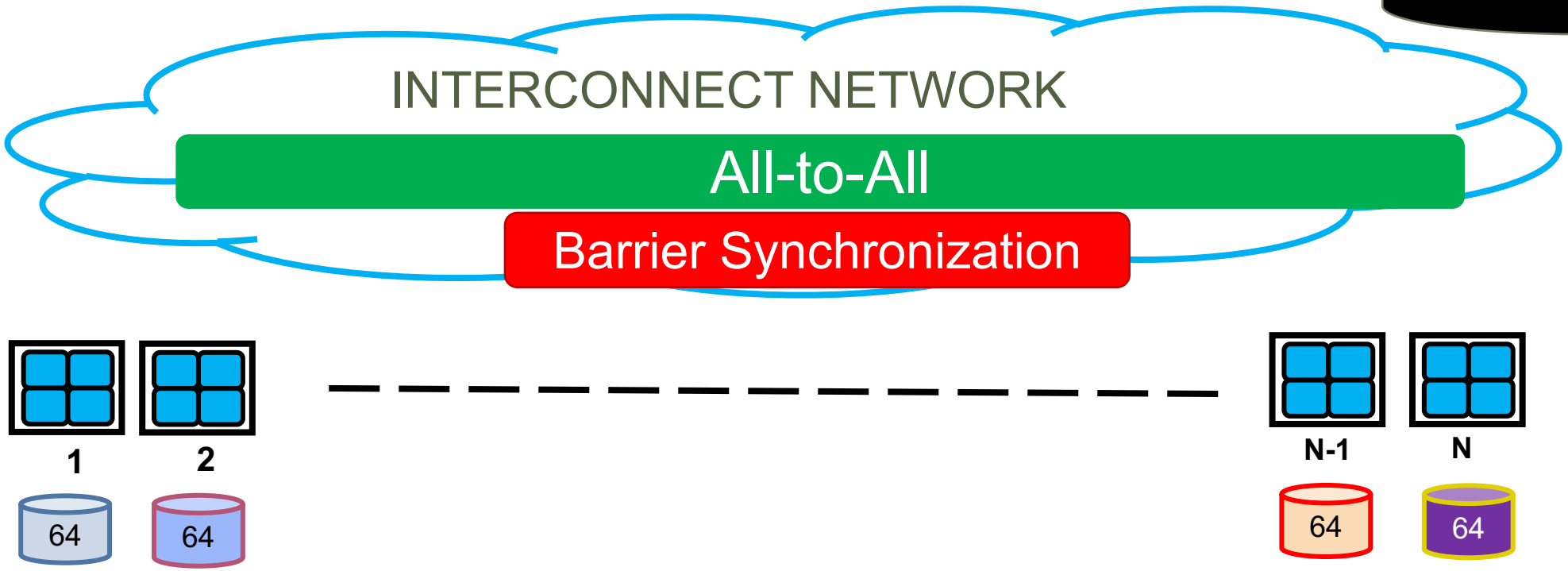
Bulk Synchronous SGD



- Synchronization bottleneck
- Various approach to ameliorate this but the problem is inherent

Rogers, R. O et. al.

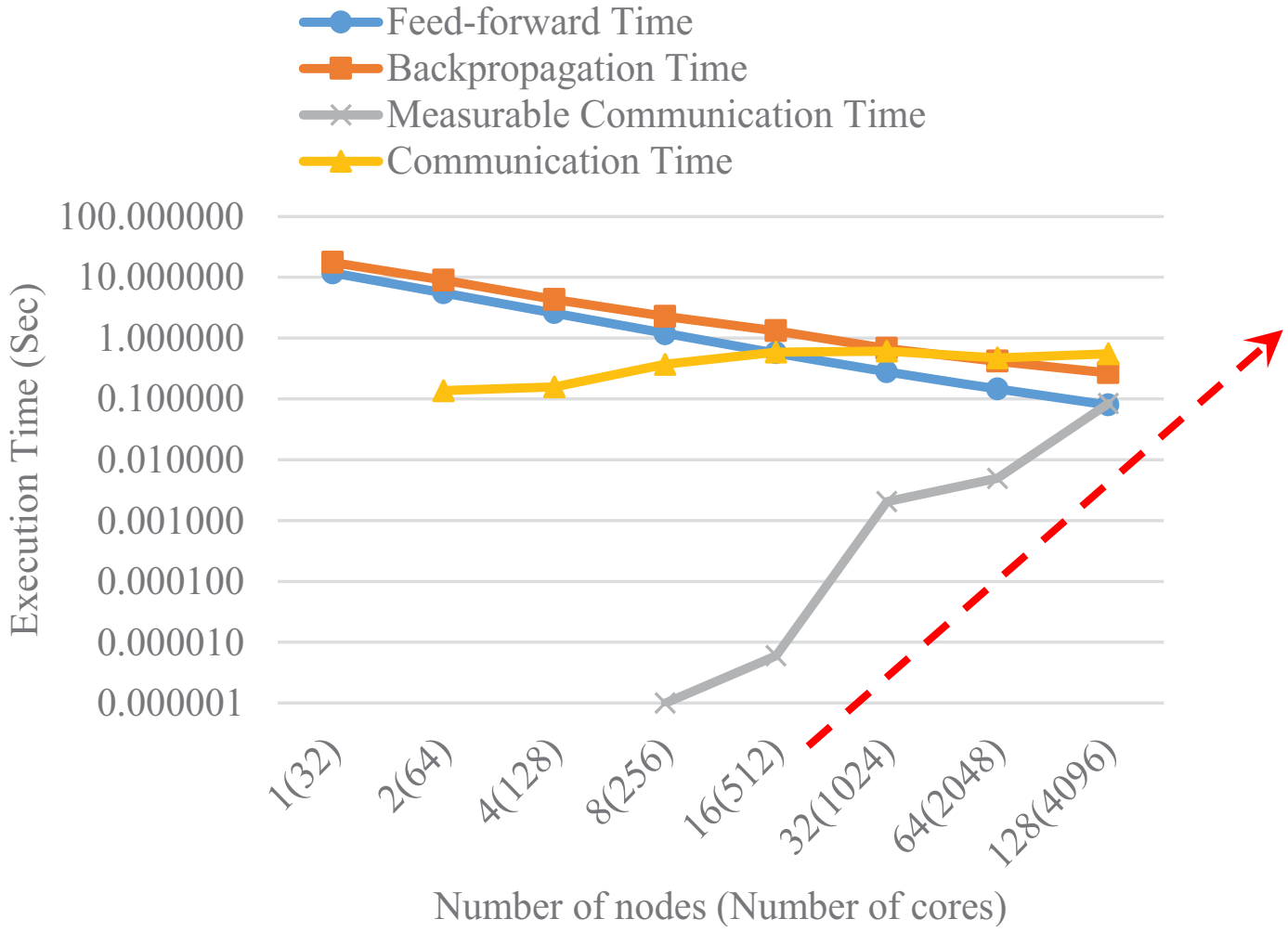
We MUSt Increase The Batch size



- If we want to keep scaling synchronous SGD then we have to keep increasing the batch size
- N=256 -> Batch Size=**16K**

Overlap Communication and Compute To Hide Network Latency

- Breakdown for VGG
- Minibatch 256

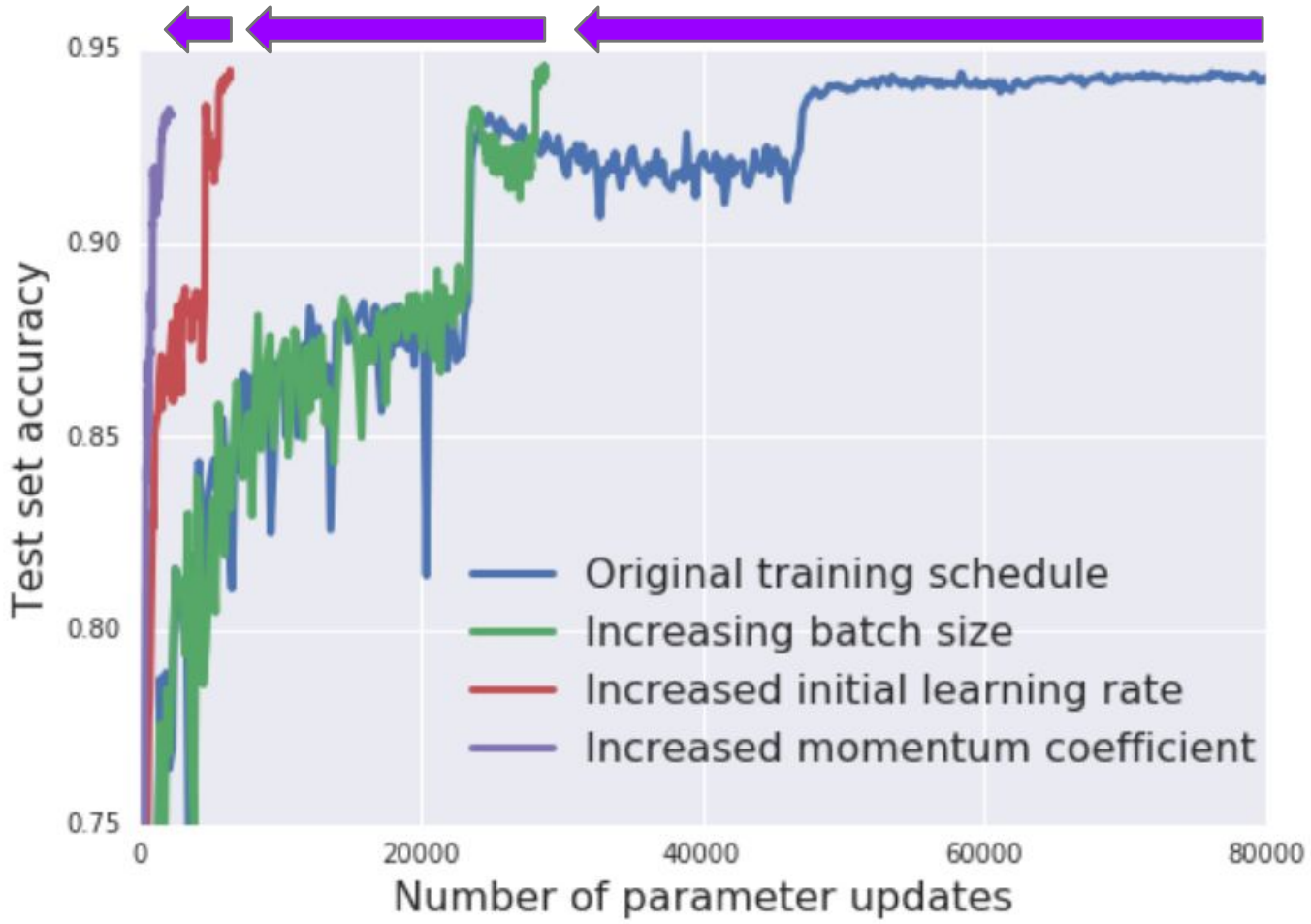


Sunwoo Lee, Dipendra Jha, Ankit Agrawal, Alok Choudhary, and Wei-keng Liao, “Parallel Deep Convolutional Neural Network Training by Exploiting the Overlapping of Computation and Communication “, IEEE 24th International Conference on High Performance Computing, 2017.

Das, Dipankar, Sasikanth Avancha, Dheevatsa Mudigere, Karthikeyan Vaidynathan, Srinivas Sridharan, Dhiraj Kalamkar, Bharat Kaul, and Pradeep Dubey. "Distributed deep learning using synchronous stochastic gradient descent." *arXiv preprint arXiv:1602.06709* (2016).

Don't Decay the Learning Rate, Increase the Batch Size

- The key difficulty
  - Numerical optimization
- Decrease # of parameter updates
- Batch Size vs. learning rate
- CIFAR-10 & Imagenet
- Not general enough



Samuel L. Smith, Pieter-Jan Kindermans, Chris Ying, Quoc V. Le," Don't Decay the Learning Rate, Increase the Batch Size," ICLR 2018. https://arxiv.org/abs/1711.00489

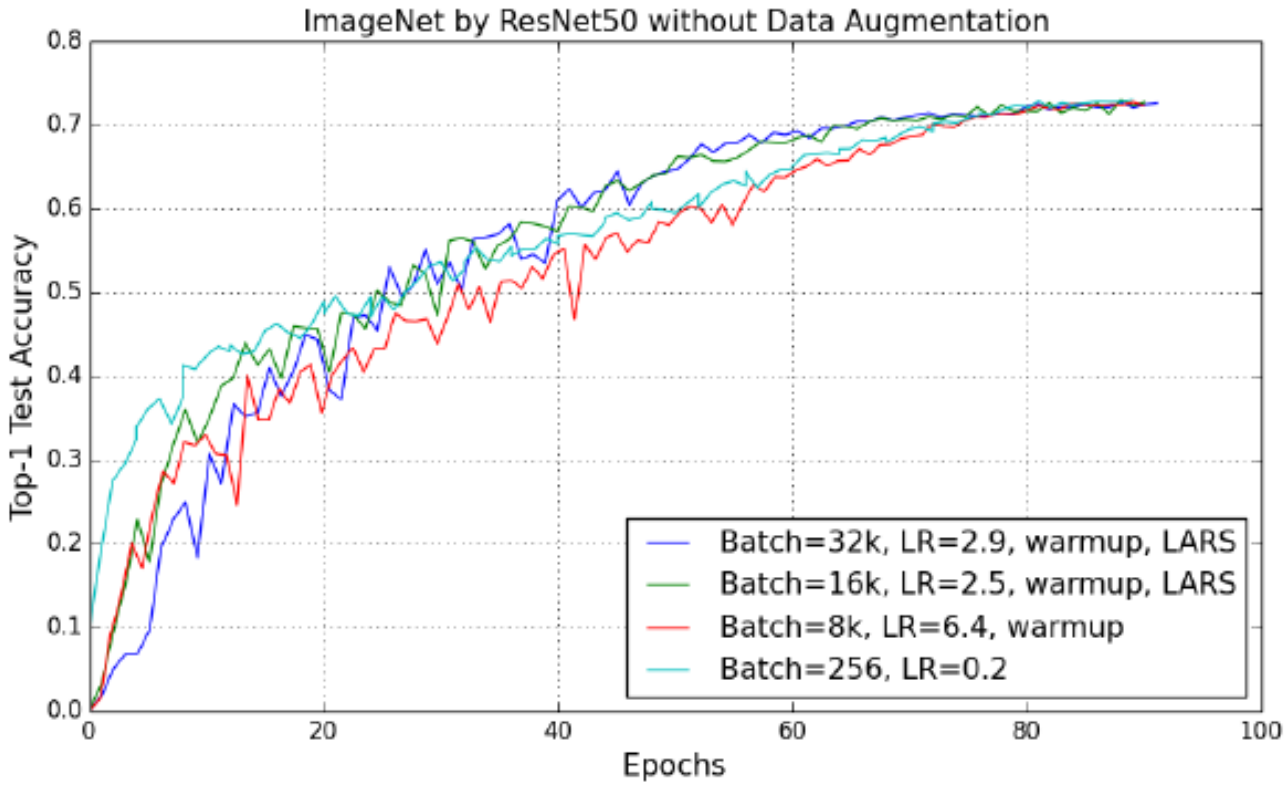
Layer-Wise Adaptive Learning Rate

RESNET-50 WITH LARS: B→ 32K

- LARS:
  - Adapts the learning rate for each layer
  - Scaling to

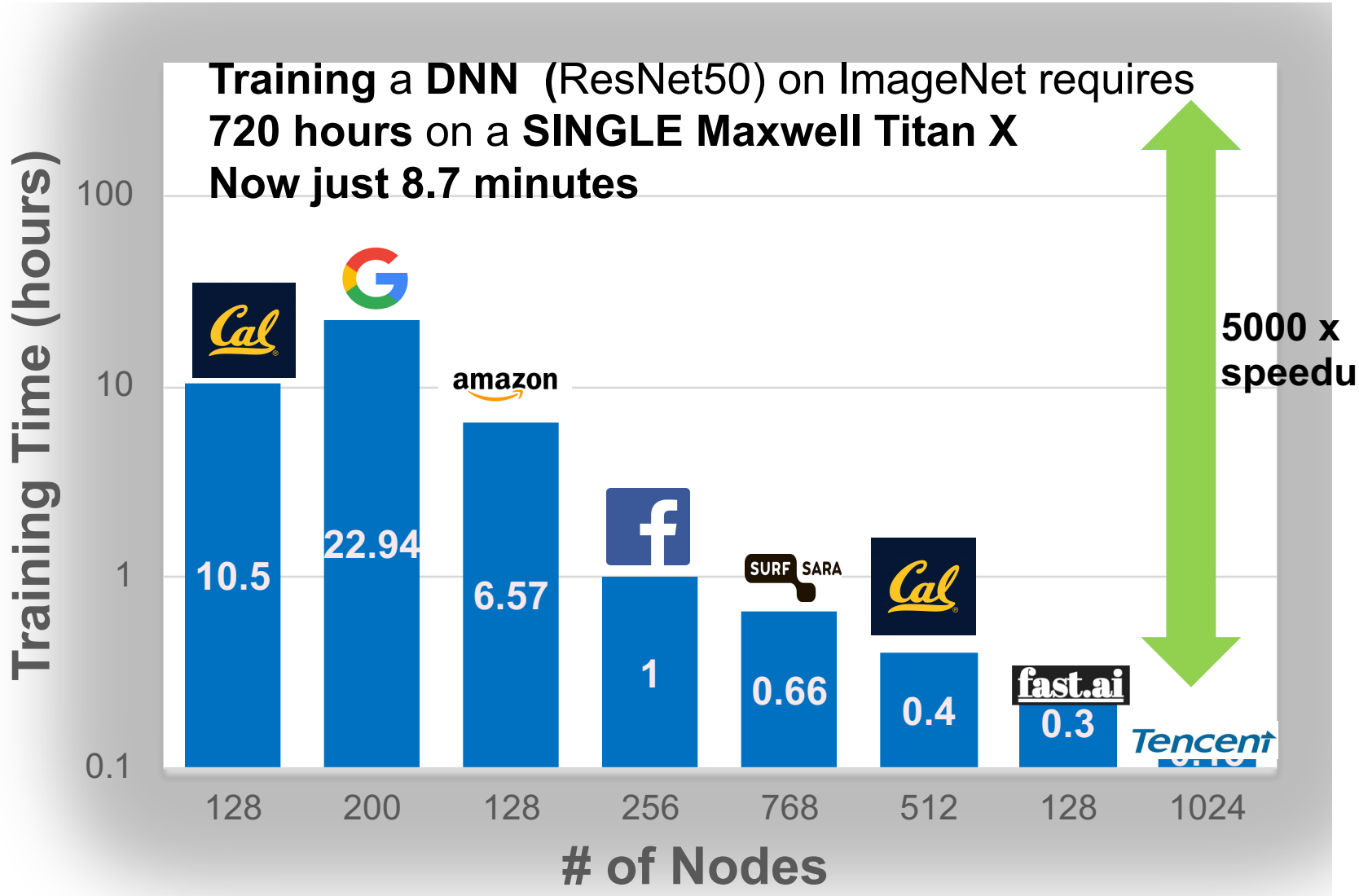
    B=8K for Alexnet

    B=32K for Resnet-50.
- Above 32K without accuracy loss is still open problem

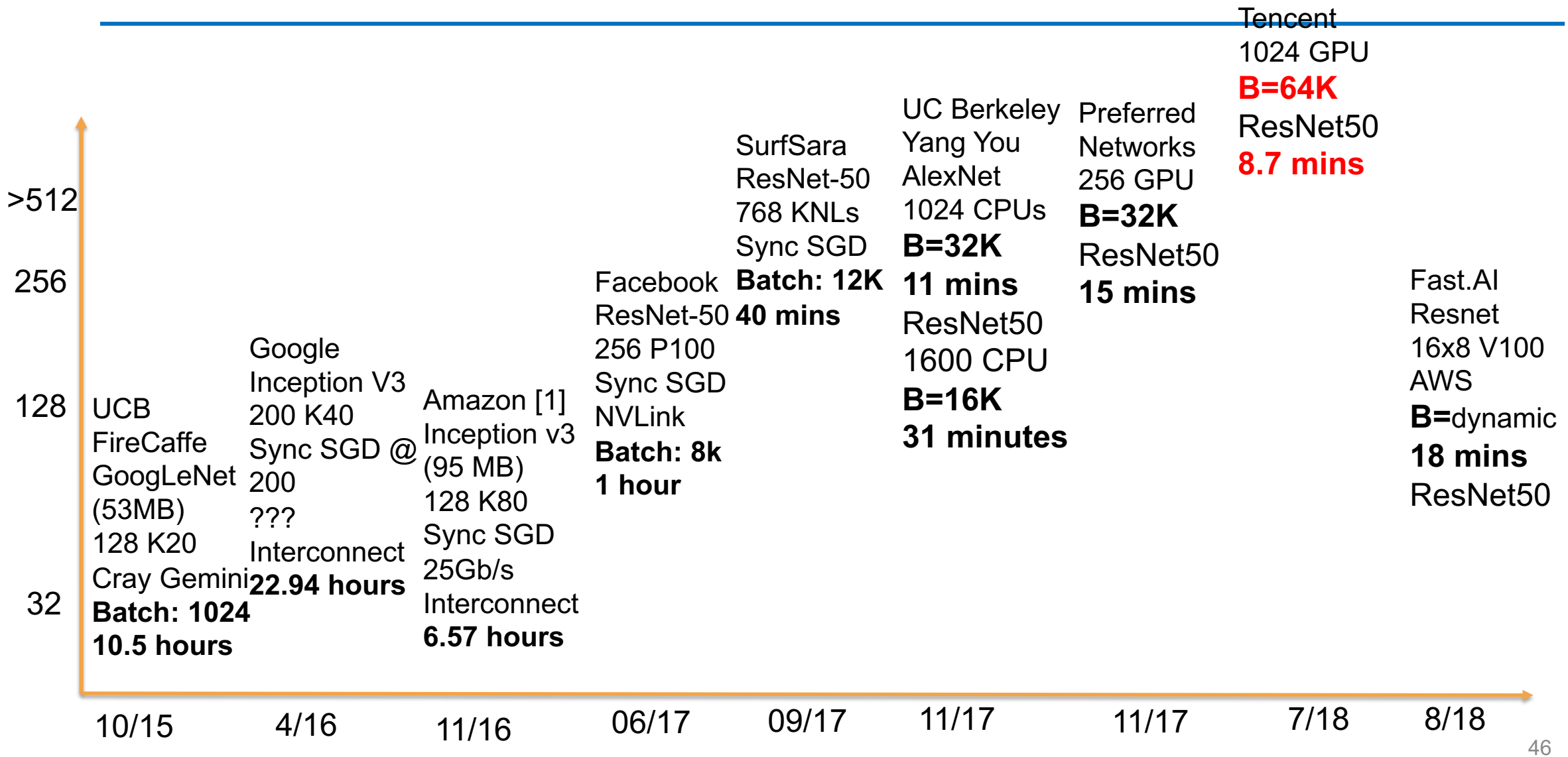


- You, Yang, Igor Gitman, and Boris Ginsburg. "Scaling SGD Batch Size to 32K for ImageNet Training." *arXiv preprint arXiv:1708.03888* (2017).
- You, Yang, Zhao Zhang, C. Hsieh, James Demmel, and Kurt Keutzer. "ImageNet training in minutes." ICPP 2018. https://arxiv.org/abs/1709.05011

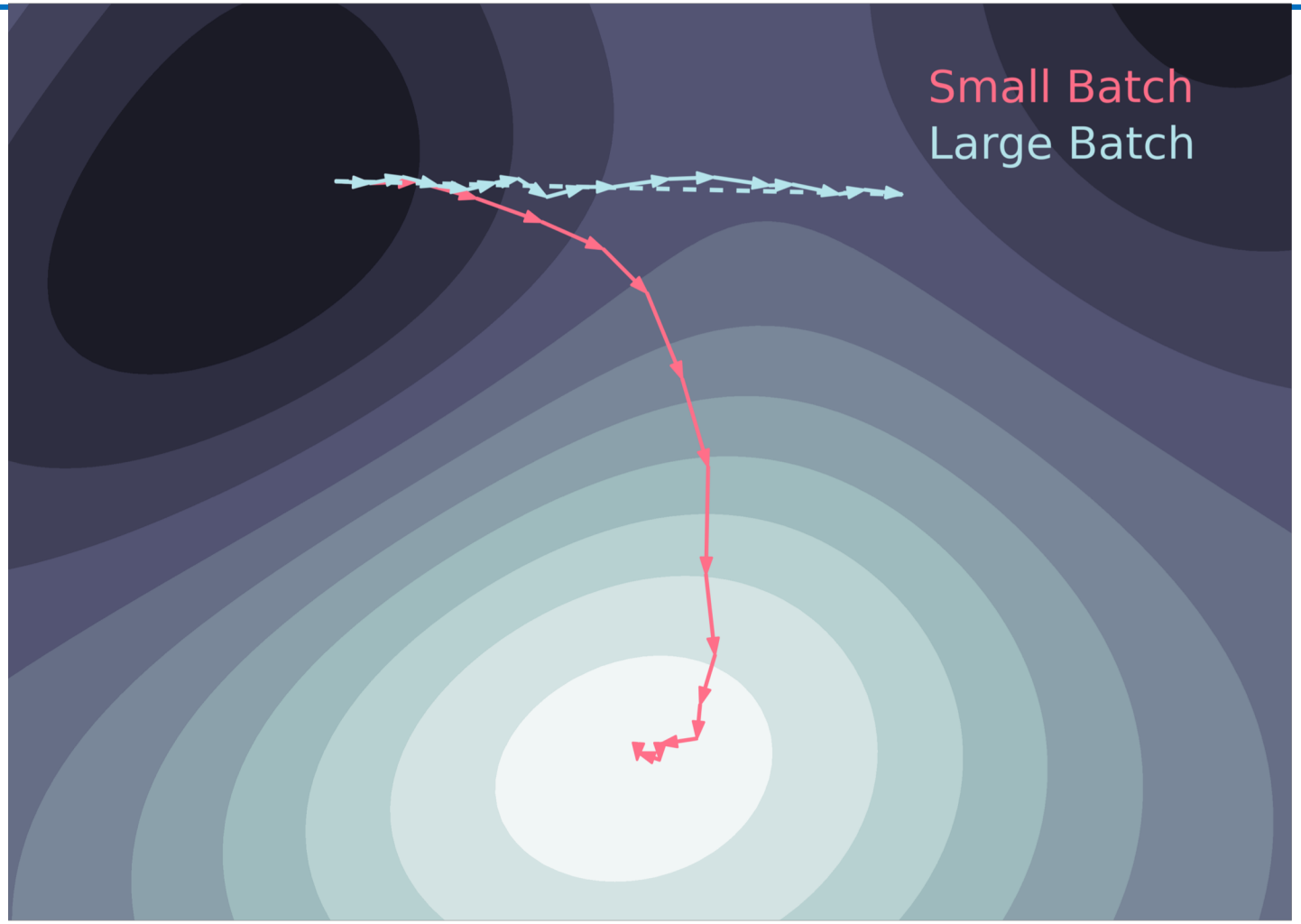
Recent Progress



Pushing Synchronous SGD Farther

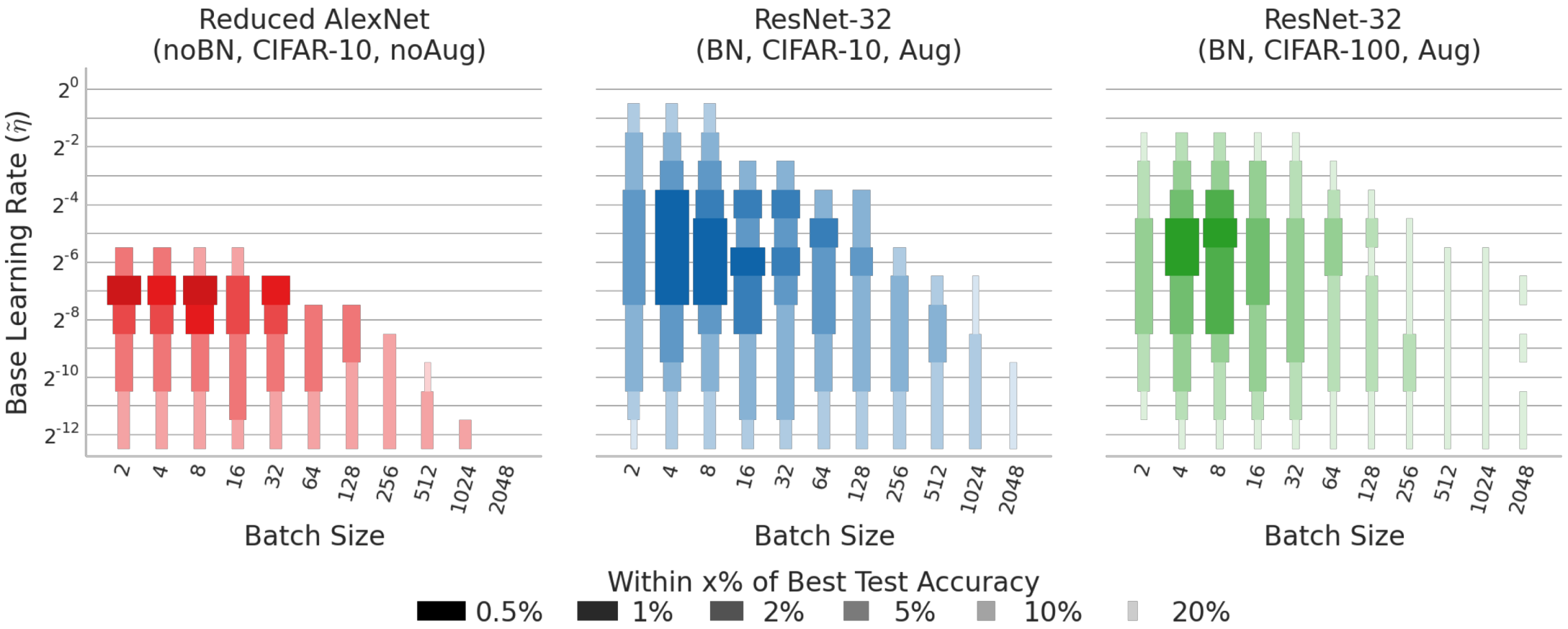


Large Batches Miss THE Minimum



Dominic Masters, Carlo Luschi, “Revisiting Small Batch Training for Deep Neural Networks” arXiv:1804.07612 2017.
https://www.graphcore.ai/posts/revisiting-small-batch-training-for-deep-neural-networks

LARGE BATCHs Miss Test Accuracy



Dominic Masters, Carlo Luschi, “Revisiting Small Batch Training for Deep Neural Networks” arXiv:1804.07612 2017.
https://arxiv.org/abs/1804.07612

Scaling Training With LARGE BATCHEs

Cons

- Constrained approach: Need to employ large batches to capture efficiency
- May not achieve target accuracy
- Only demonstrated on CNNs
- More prone to adversarial attacks[1]

Pros

- Robust: Relatively less hyperparameter tuning
- Sequential consistency
- Good infrastructural support from HPC frameworks (MPI)
- Fault tolerance is practically handled by snapshots and rollback otherwise

[1] Zhewei Yao, Amir Gholami, Qi Lei , Kurt Keutzer, Michael Mahoney, "Hessian-based Analysis of Large Batch Training and Robustness to Adversaries", arXiv:1802.08241v. https://arxiv.org/pdf/1802.08241.pdf

Asynchronous SGD

- Hogwild![1]
  - Asynchronous on shared memory
- Distributed asynchronous SGD
  - Googles training (Dist belief)[2]
  - Accuracy has degraded at large scaling >32 [3,4]
- Deep gradient compression[5]

[1]- Feng Niu, Benjamin Recht, Christopher R´e and Stephen J. Wright, “Hogwild!: A Lock-Free Approach to Parallelizing Stochastic Gradient Descent”, NIPS 2011. https://people.eecs.berkeley.edu/~brecht/papers/hogwildTR.pdf

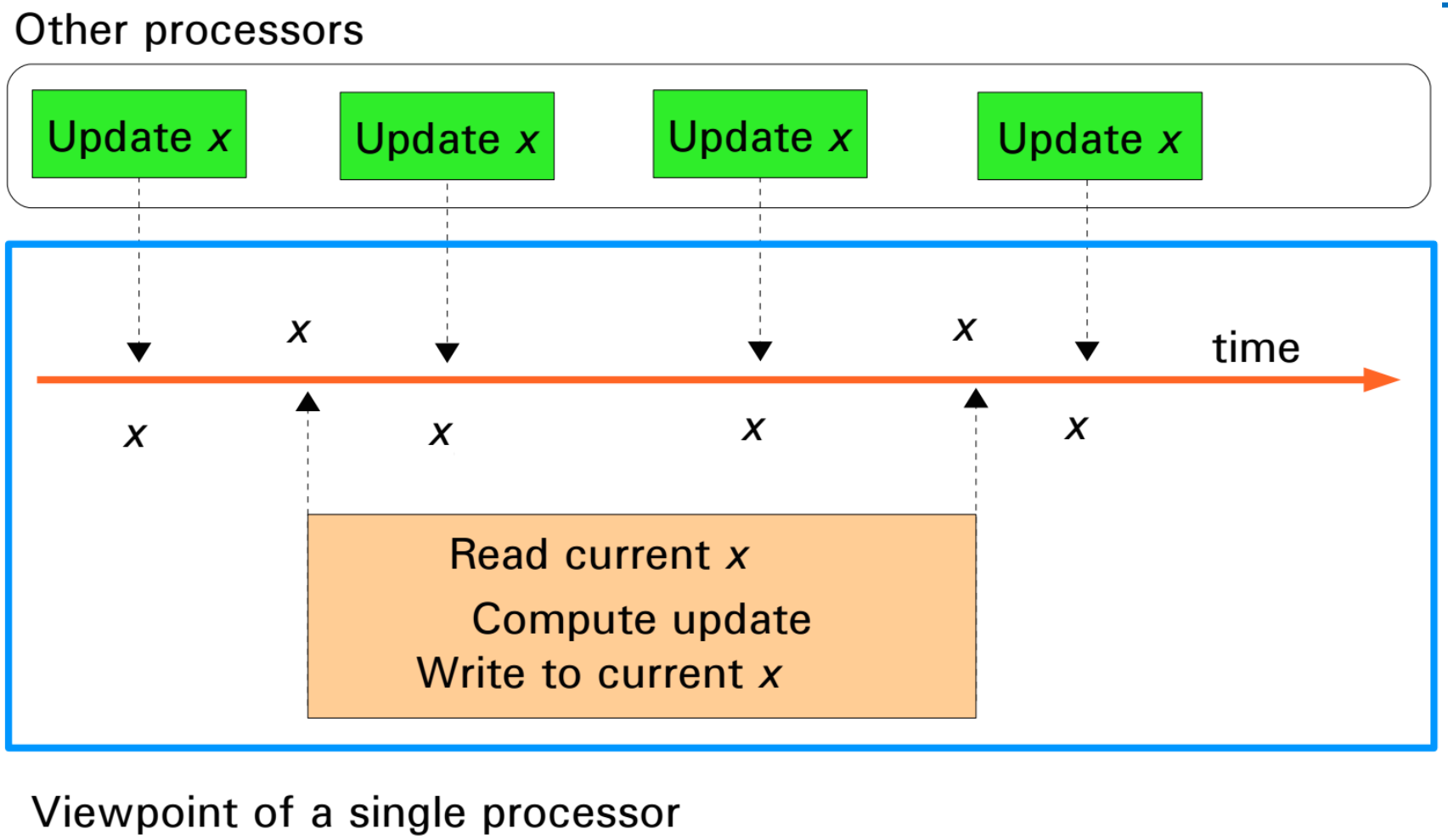
[2] - Jeffrey Dean, Greg S. Corrado, Rajat Monga, Kai Chen, Matthieu Devin, Quoc V. Le, Mark Z. Mao, Marc’Aurelio Ranzato, Andrew Senior, Paul Tucker, Ke Yang, Andrew Y. Ng , “Large Scale Distributed Deep Networks”. https://ai.google/research/pubs/pub40565

[3]- Zhang, Sixin, Anna E. Choromanska, and Yann LeCun. "Deep learning with elastic averaging SGD." In Advances in Neural Information Processing Systems, pp. 685-693. 2015.

[4]- Jin, Peter H., Qiaochu Yuan, Forrest Iandola, and Kurt Keutzer. "How to scale distributed deep learning?." arXiv preprint arXiv:1611.04581 (2016). NIPS MLSys 2017.

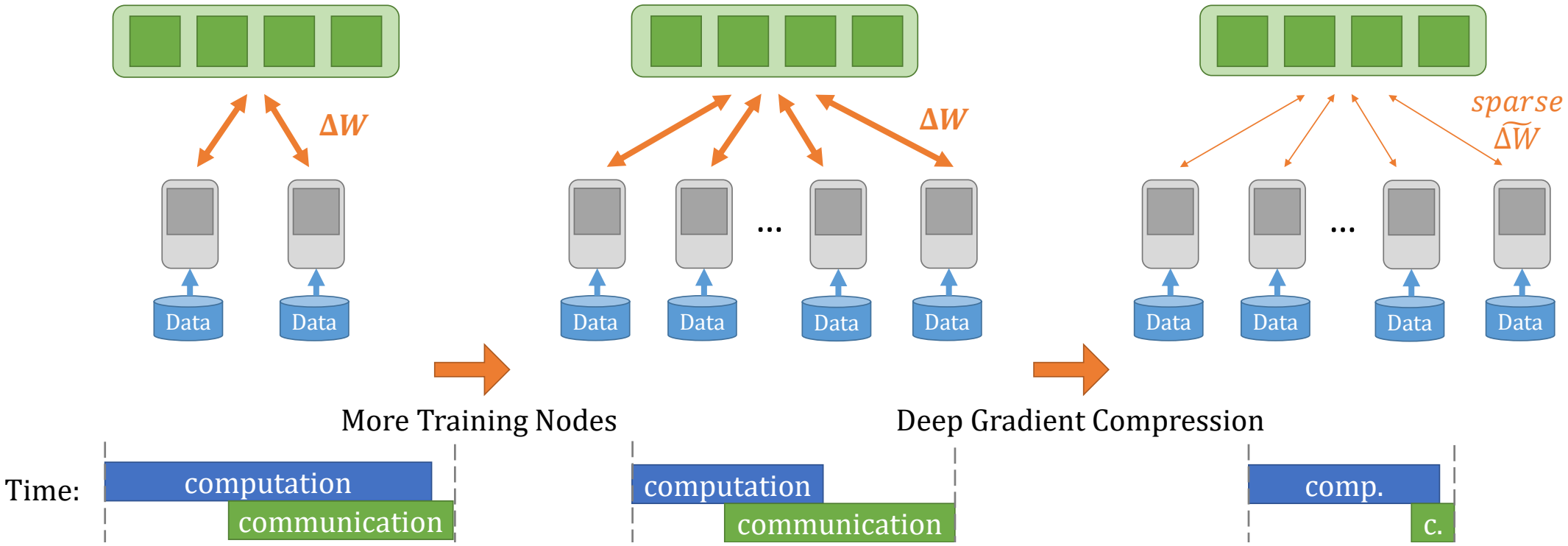
[5]- Yujun Lin, Song Han, Huizi Mao, Yu Wang, William J. Dally, “Deep Gradient Compression: Reducing the Communication Bandwidth for Distributed Training” ICLR 2018. https://arxiv.org/abs/1712.01887

HOGWILD!



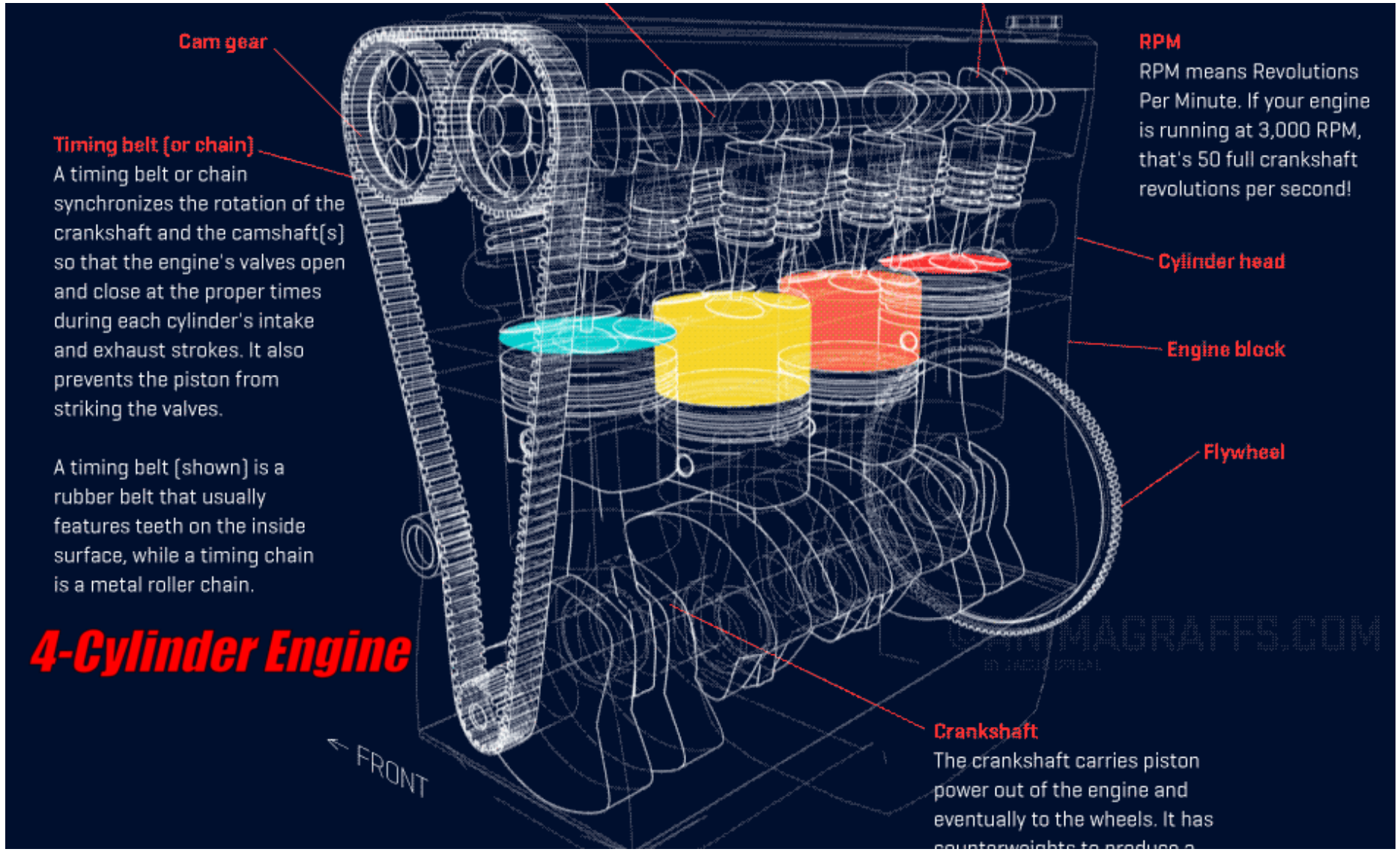
Feng Niu, Benjamin Recht, Christopher R´e and Stephen J. Wright, “Hogwild!: A Lock-Free Approach to Parallelizing Stochastic Gradient Descent”, NIPS 2011. https://people.eecs.berkeley.edu/~brecht/papers/hogwildTR.pdf

Deep Gradient Compression



Yujun Lin, Song Han, Huizi Mao, Yu Wang, William J. Dally, "Deep Gradient Compression: Reducing the Communication Bandwidth for Distributed Training" ICLR 2018. https://arxiv.org/abs/1712.01887

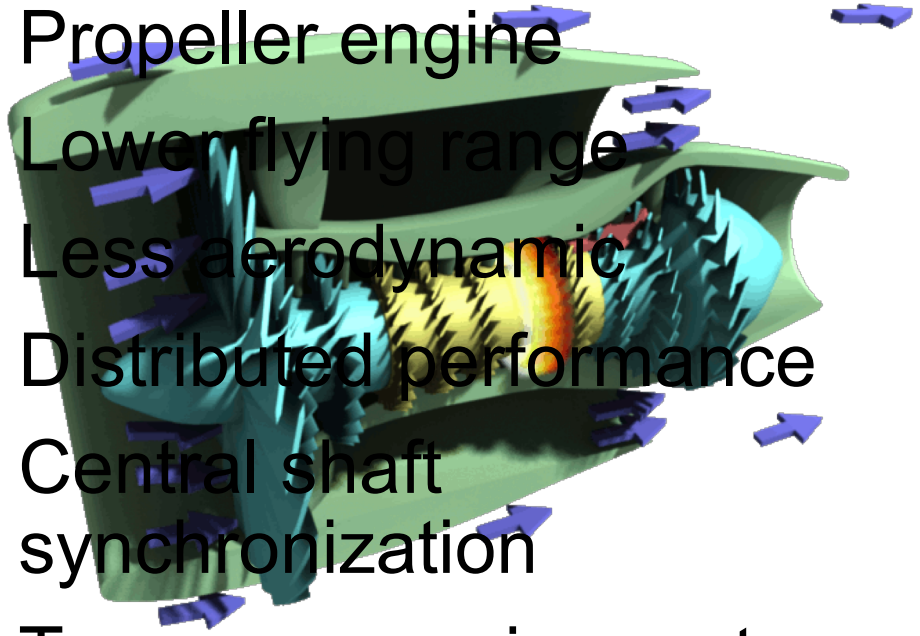
Why Did We Put Engines On The Course Website?



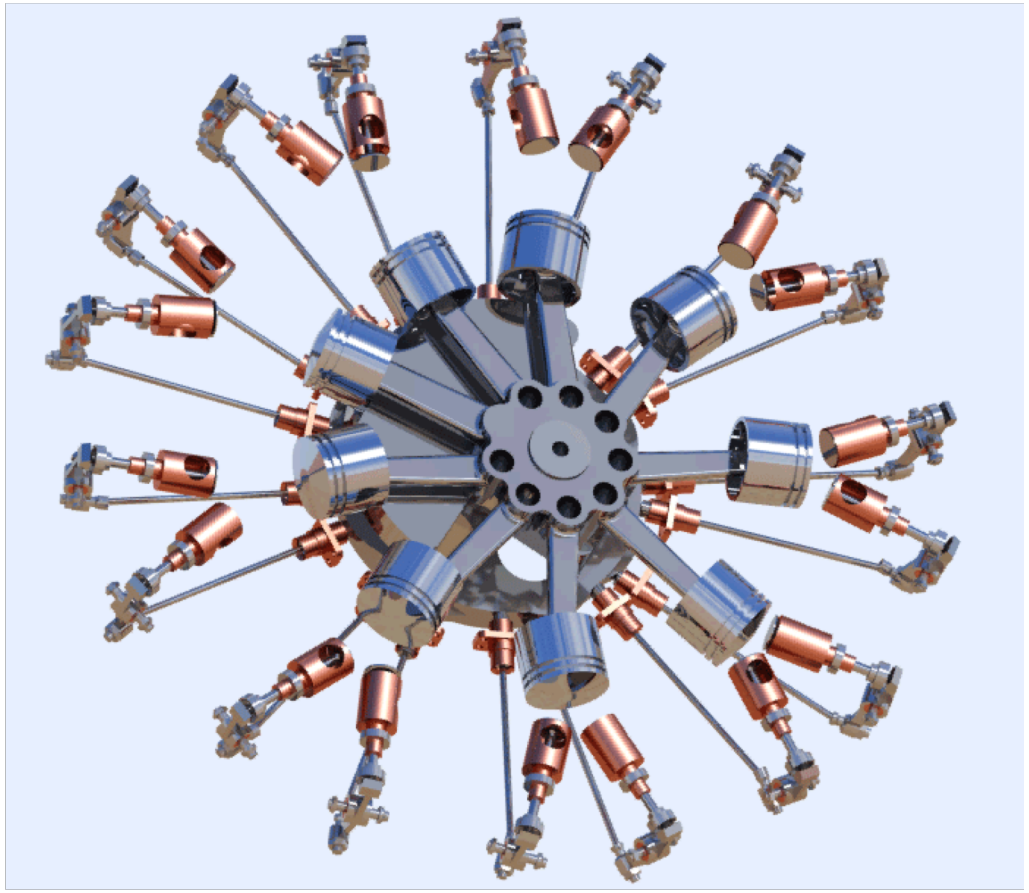
“If we all worked on the assumption that what is accepted as true is really true, there would be little hope of advance.” Orville Wright

Radial Engine

Turbo Fan Jet Engine



- Propeller engine
- Lower flying range
- Less aerodynamic
- Distributed performance
- Central shaft synchronization
- Too many moving parts

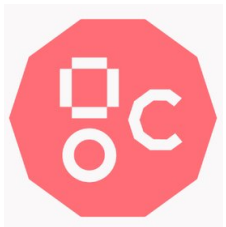
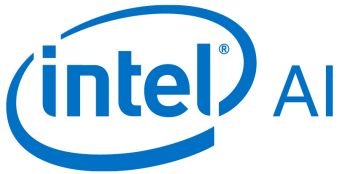


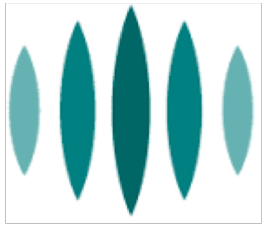
1. https://imgur.com/gallery/79Qo0/comment/12698133
2. By RichardWheeler from wikipedia

FUTURE OF Accelerated TRAINING

- Increase physical scale to support model parallelism
- Architectures to improve communication and memory bandwidth
- Dedicated silicon area to neural network compute
- Exploit sparsity

Keep an eye out for companies that will contribute to cloud training

Conclusion: Today

- Scaling training is a matter of processor performance & communication latency
- Current accelerators:
  - GEMM centric to overcome memory wall
  - DNN frequently does not fit
  - Require large batch size to achieve high utilization/performance
- Scaling synchronous SGD requires large batch methods
  - Accuracy may require extensive hyperparameter tuning
  - Very large batch sizes only demonstrated on CNNs
- Minimize the impact of communication latency bottleneck
  - Use asynchronous approaches, but those have all negatively impacted accuracy
  - Hide communication latency by pipelining gradients
- Benchmarks emerging with some difficulty