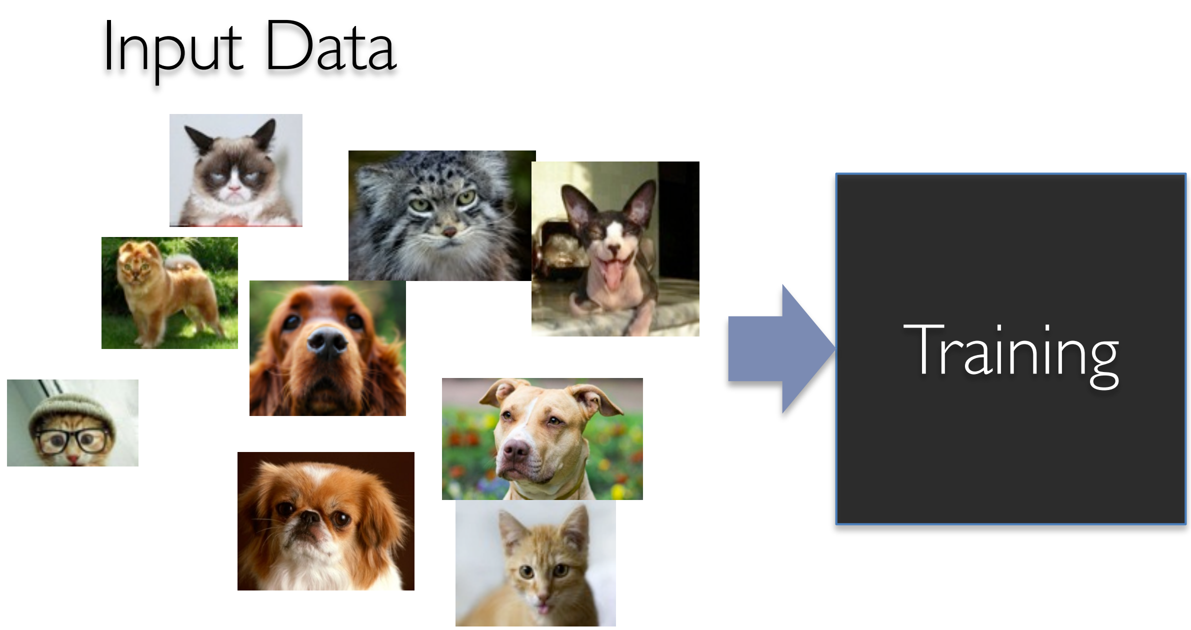
Inference on Deep Networks, Model Compression and Quantization

Dimitris Papailiopoulos
University of Wisconsin-Madison

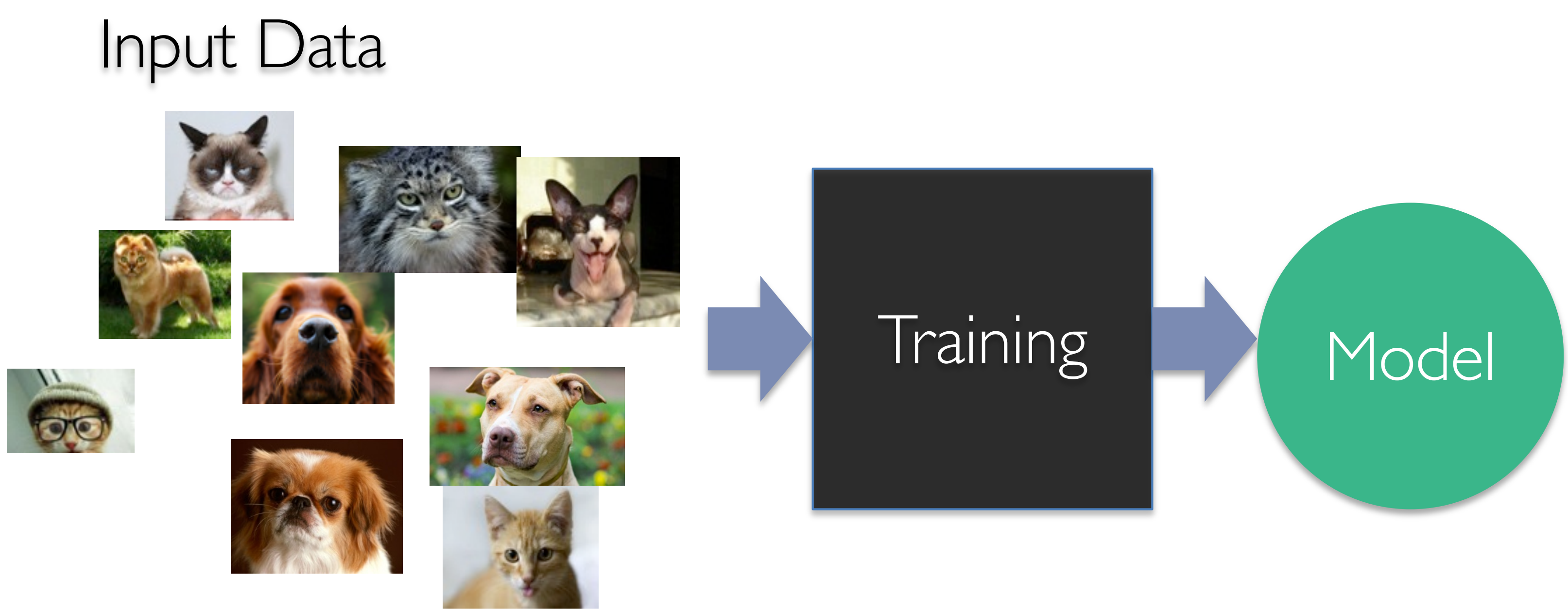
Standard ML Pipeline

Input Data

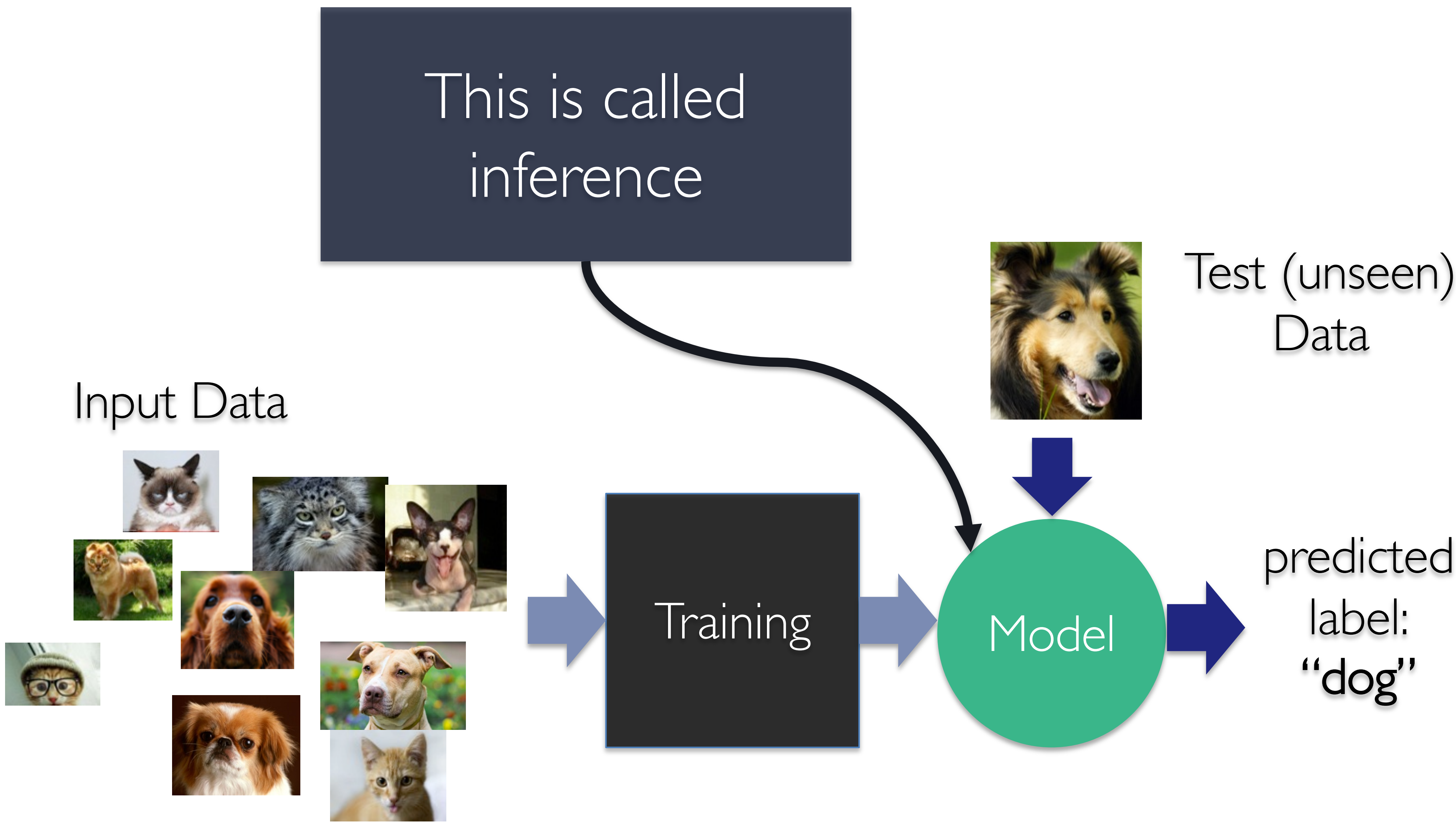
Standard ML Pipeline



Standard ML Pipeline



Standard ML Pipeline



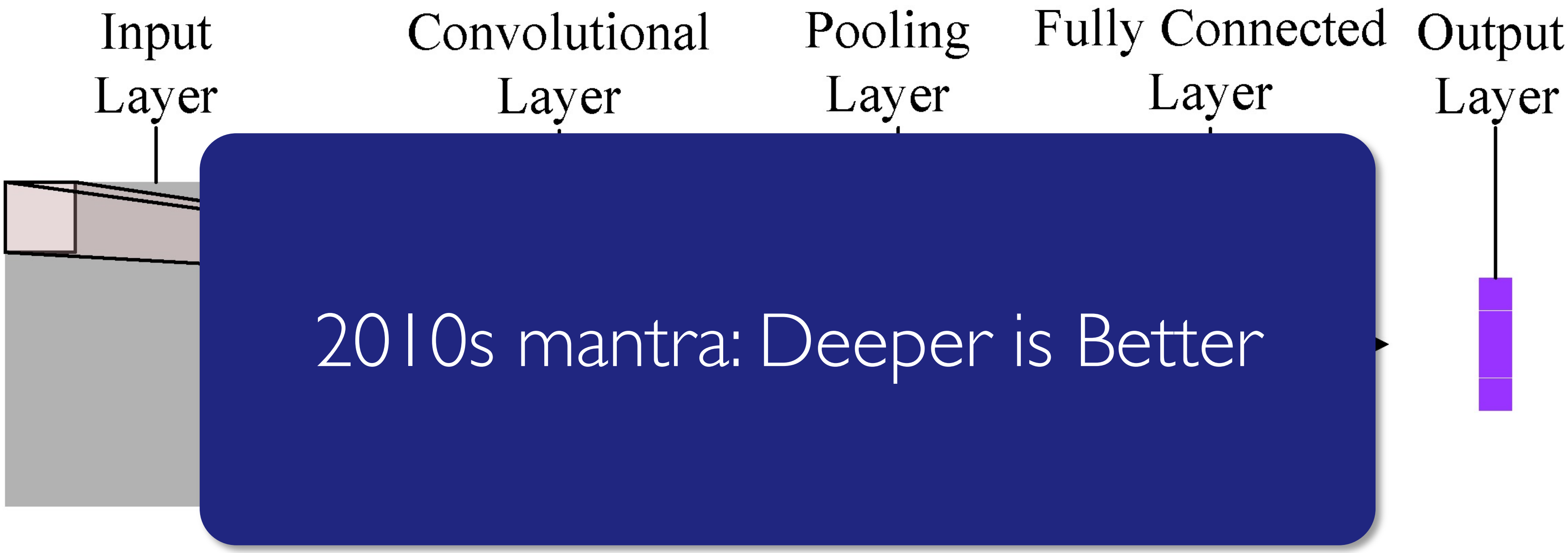
Today

- Cost of Inference

- Compression

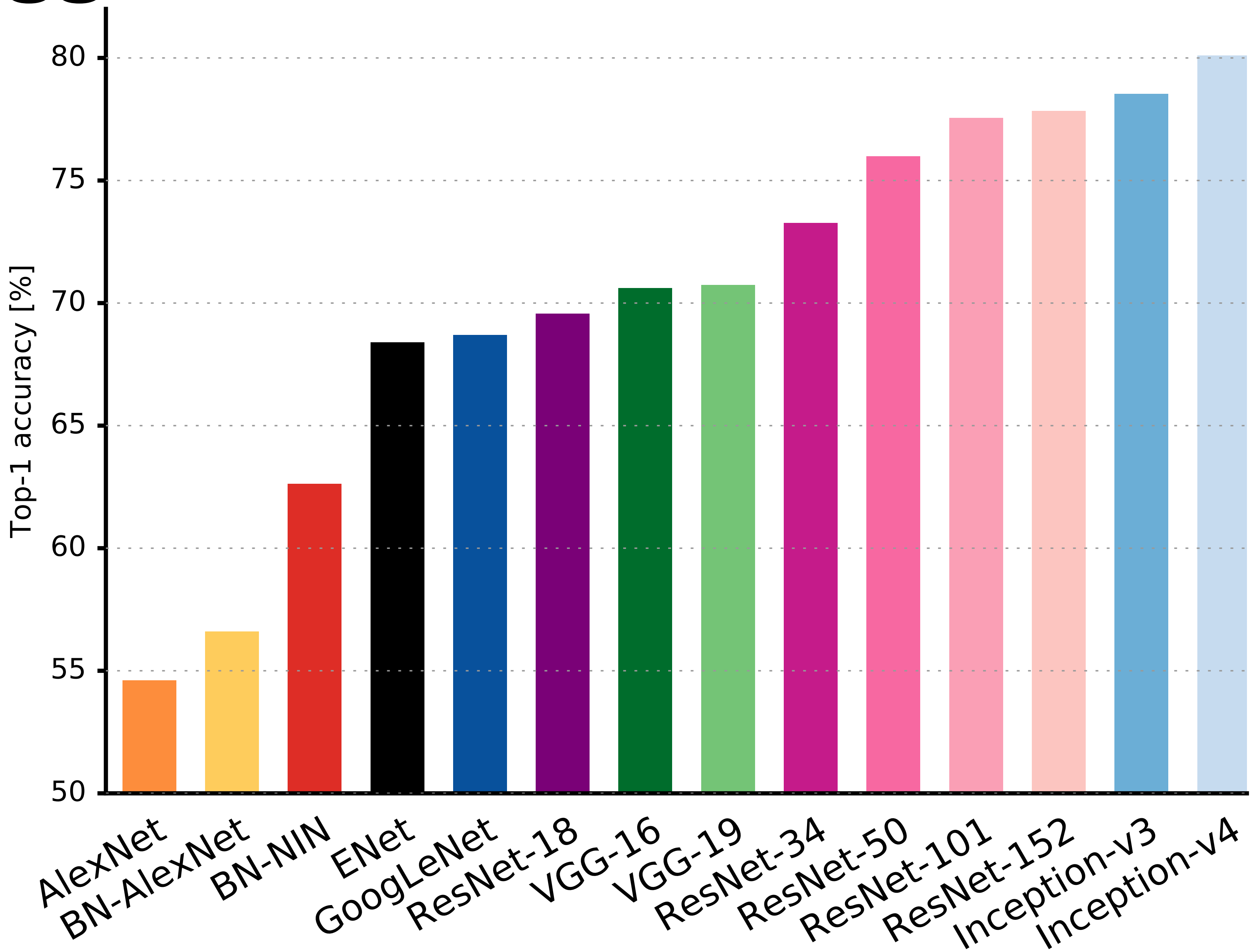
- Low-precision and Quantization

Cost of Inference



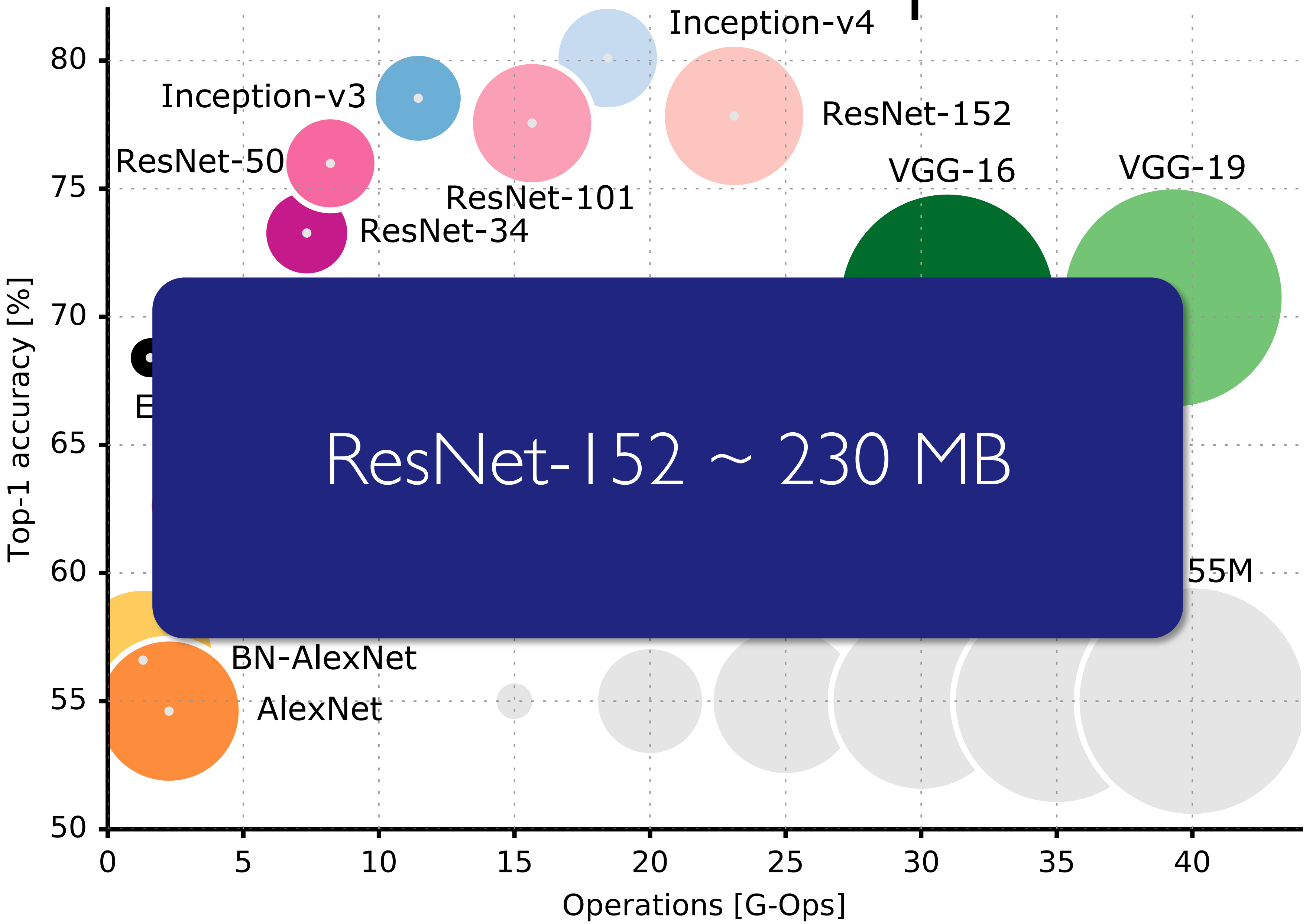
- *Memory:* storing the model is O(#parameters)
- *Computation:* For each input, you do a "forward pass"

Bigger models = Better Models?



[Canziani, Culurciello, Paszke, 2016]

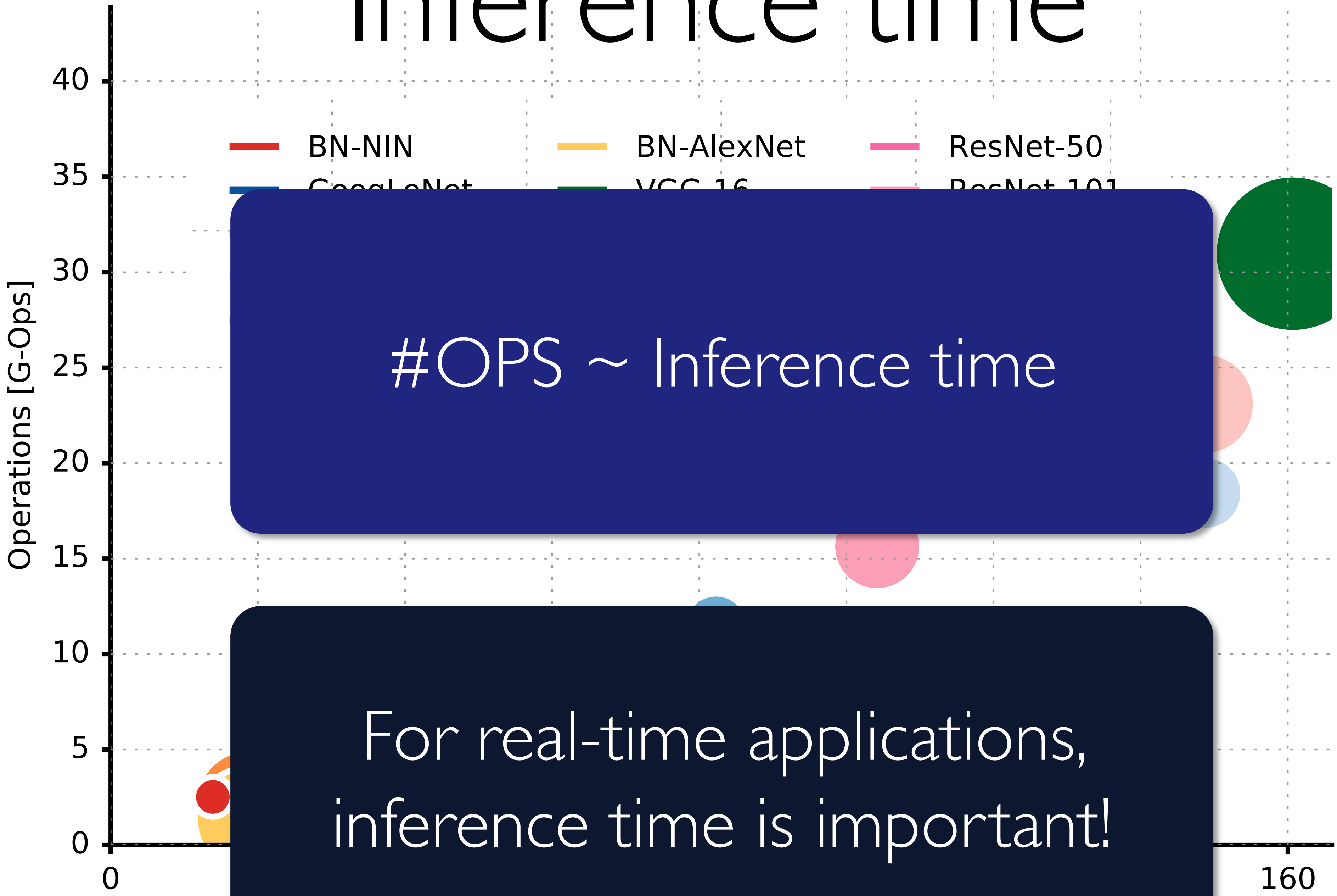
Size and Ops



ResNet-152 ~ 230 MB

[Canziani, Culurciello, Paszke, 2016]

Inference time



#OPS ~ Inference time

For real-time applications, inference time is important!

[Canziani, Culurciello, Paszke, 2016]

Tradeoffs

- A Good model has to:
    - Have high accuracy
    - Be easily trainable
    - Be fast during inference
    - Be compact

Model Compression and Quantization

Deep Compression

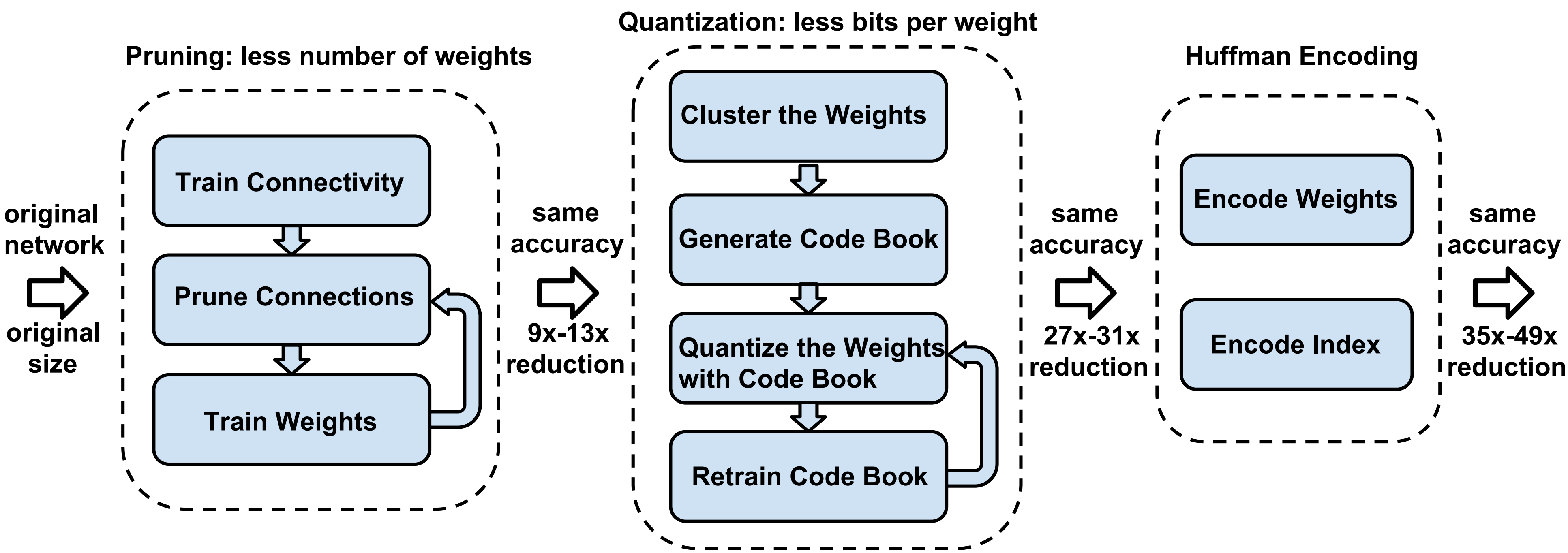
Motivation: Large models are difficult to deploy in resource limited setups

Three step procedure:

- Prune weight, while training
- Quantize weights using k-means
- Compress quantized weights

[Han, Mao, Dally, ICLR 2016]

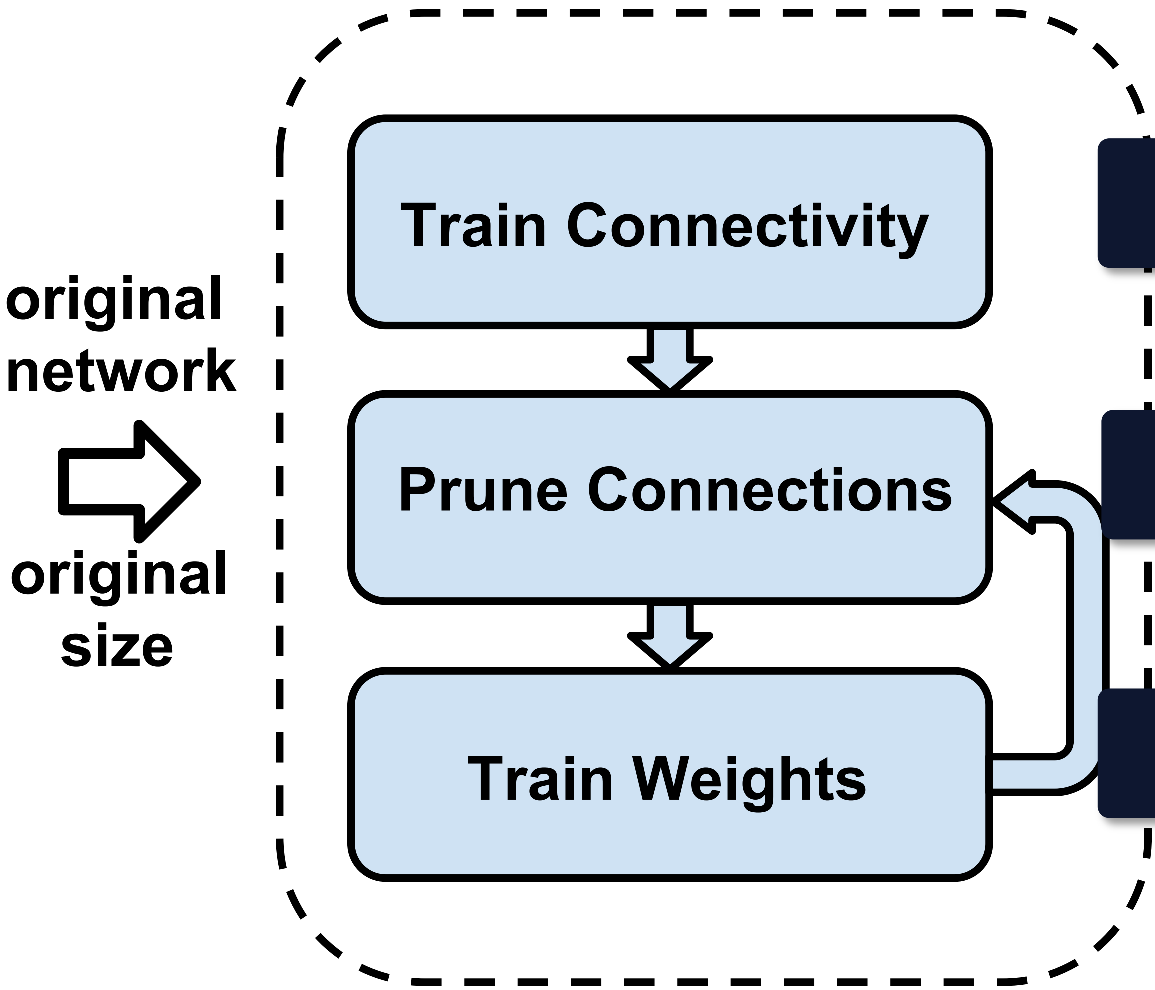
Deep Compression



[Han, Mao, Dally, ICLR 2016]

Deep Compression: Step 1

Pruning: less number of weights



SGD

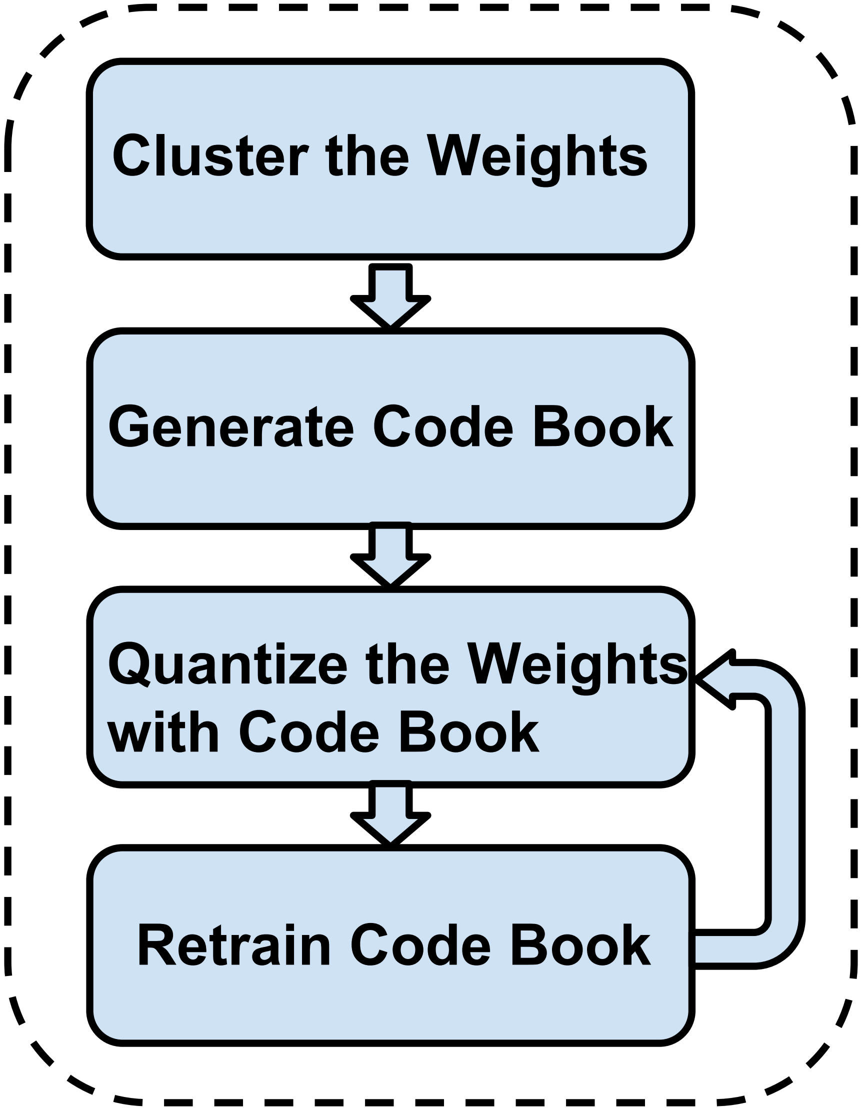
Prune |weights| < t

SGD

[Han, Mao, Dally, ICLR 2016]

Deep Compression: Step 2

Quantization: less bits per weight

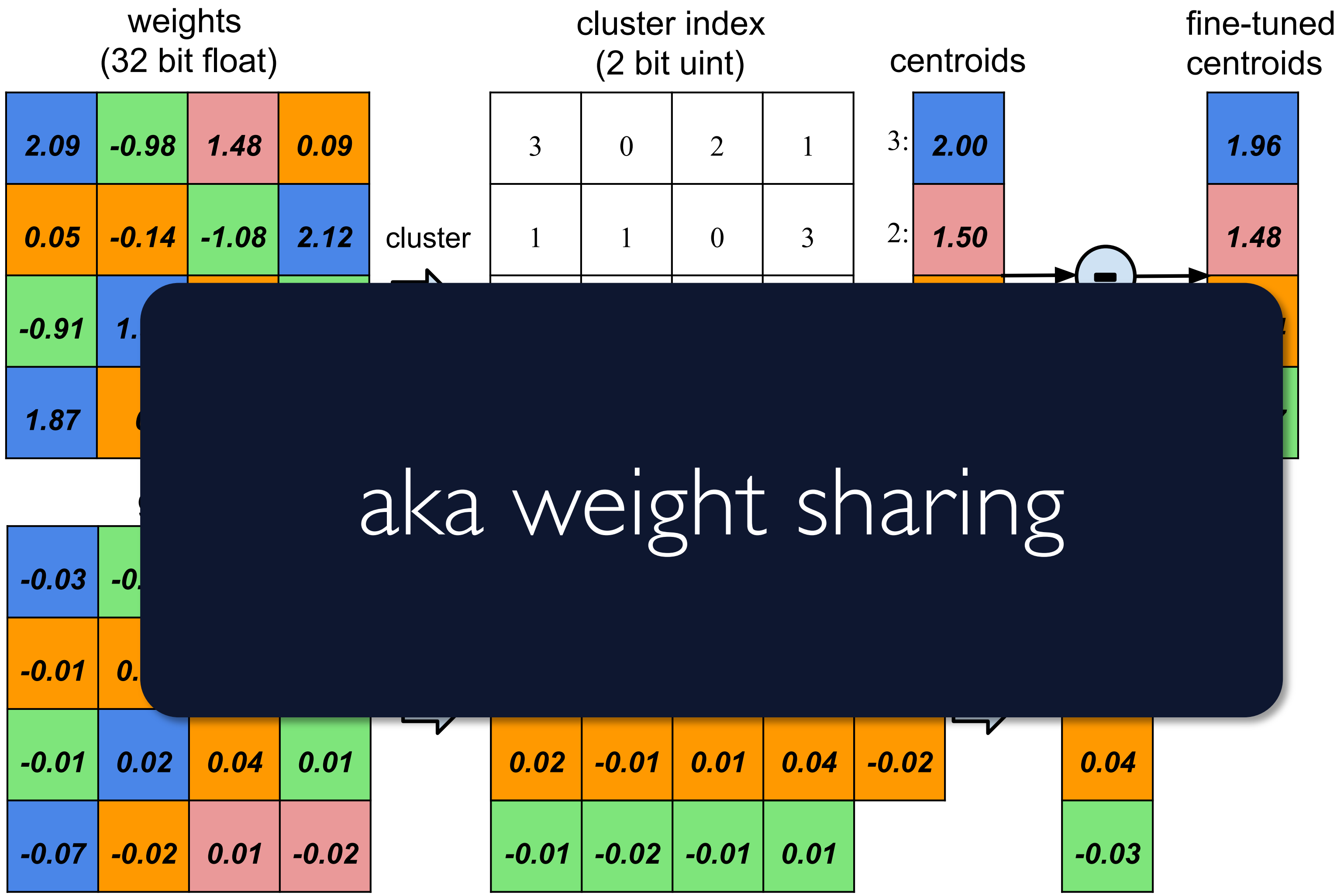


K-means used for clustering

$$\min_{S_1,\ldots,S_k} \sum_{i=1}^{k} \sum_{x \in S_i} \|x - \mu_i\|^2$$

[Han, Mao, Dally, ICLR 2016]

Deep Compression: Step 2

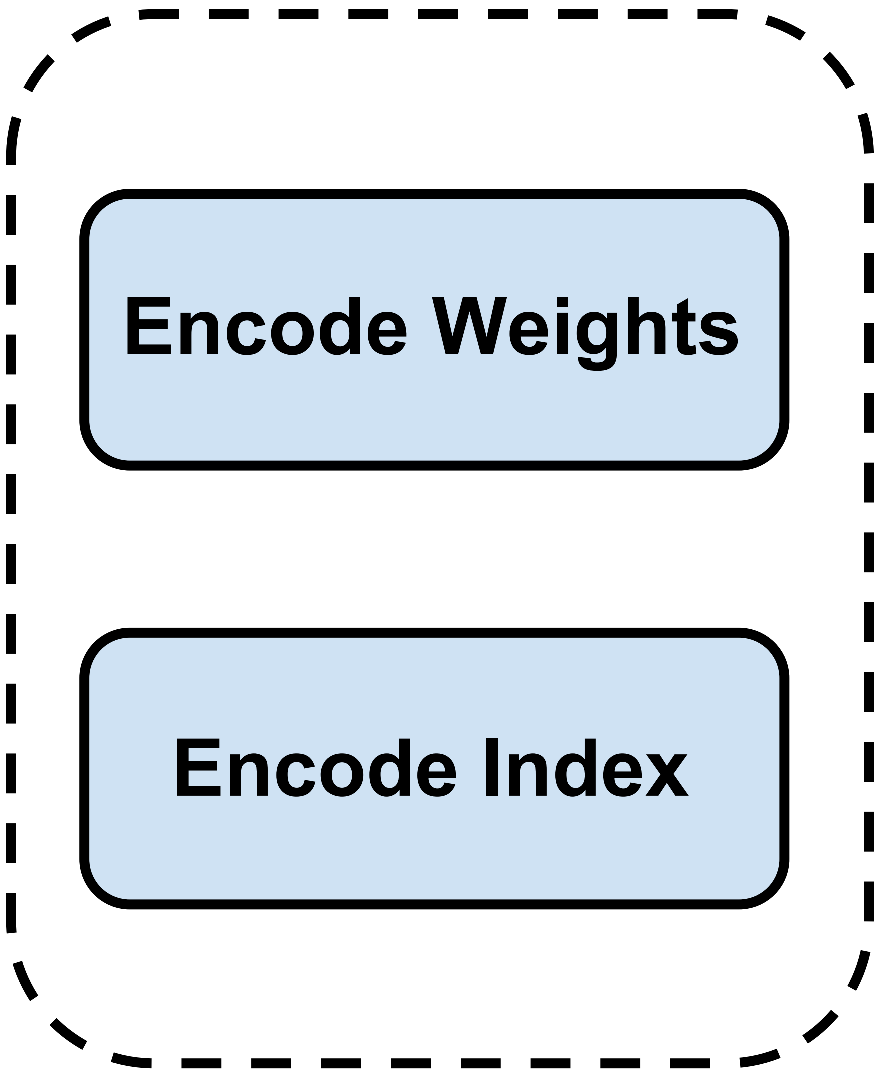


aka weight sharing

[Han, Mao, Dally, ICLR 2016]

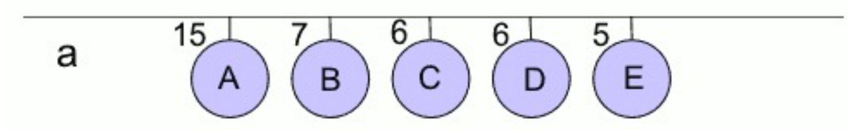
Deep Compression: Step 3

Huffman Encoding

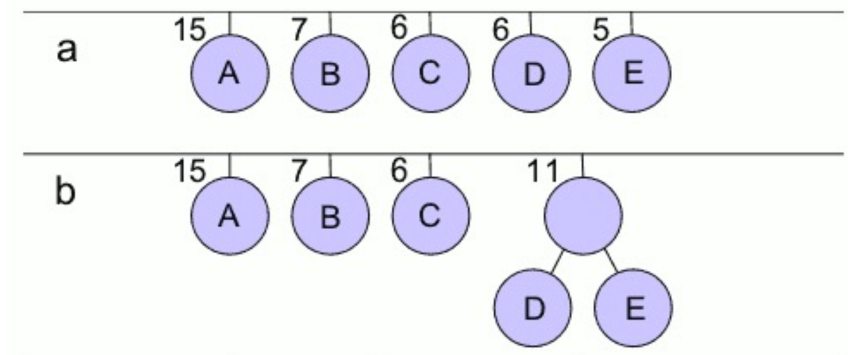


[Han, Mao, Dally, ICLR 2016]

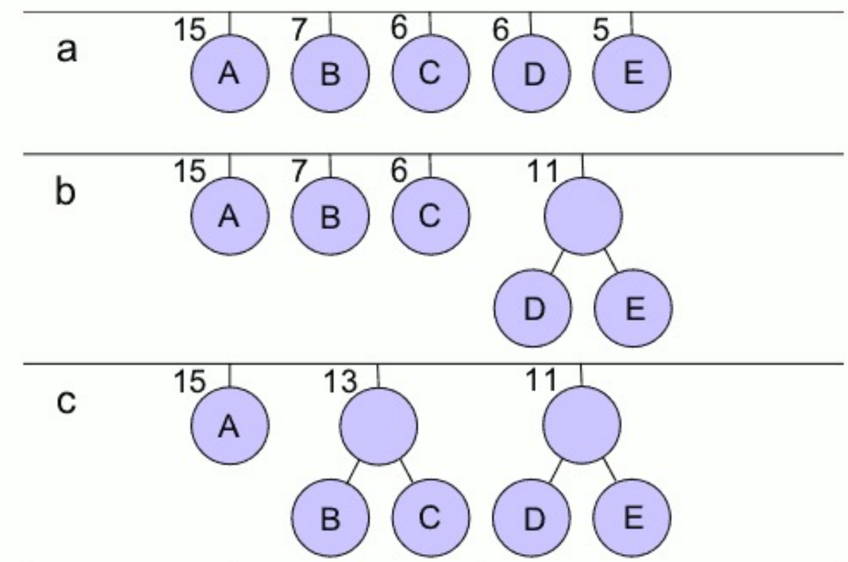
Huffman Encoding



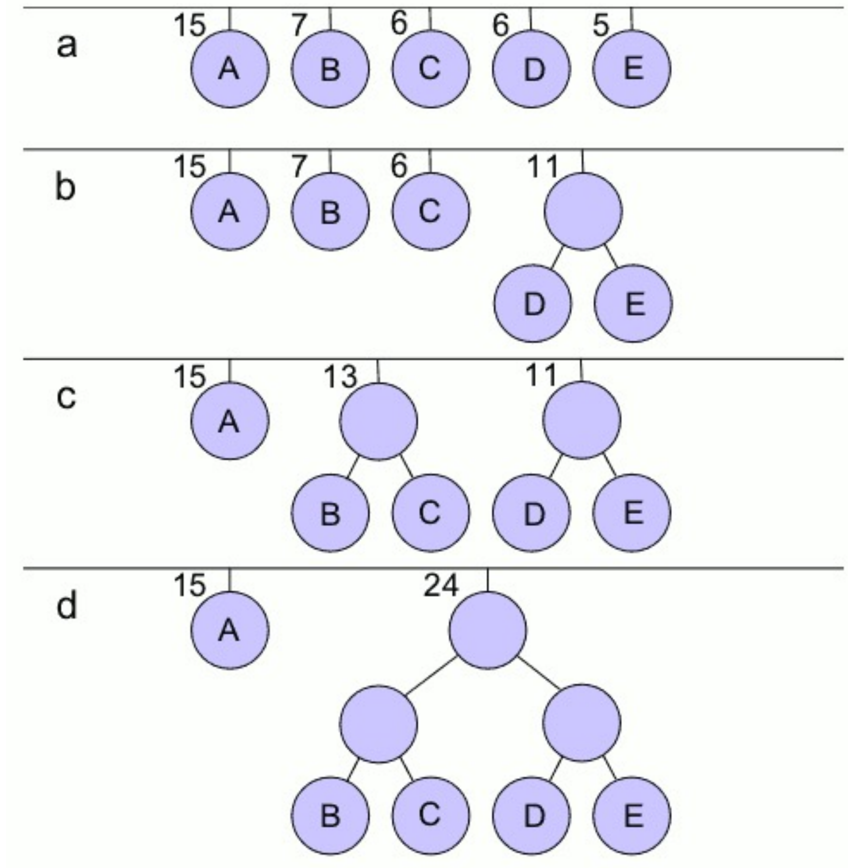
Huffman Encoding



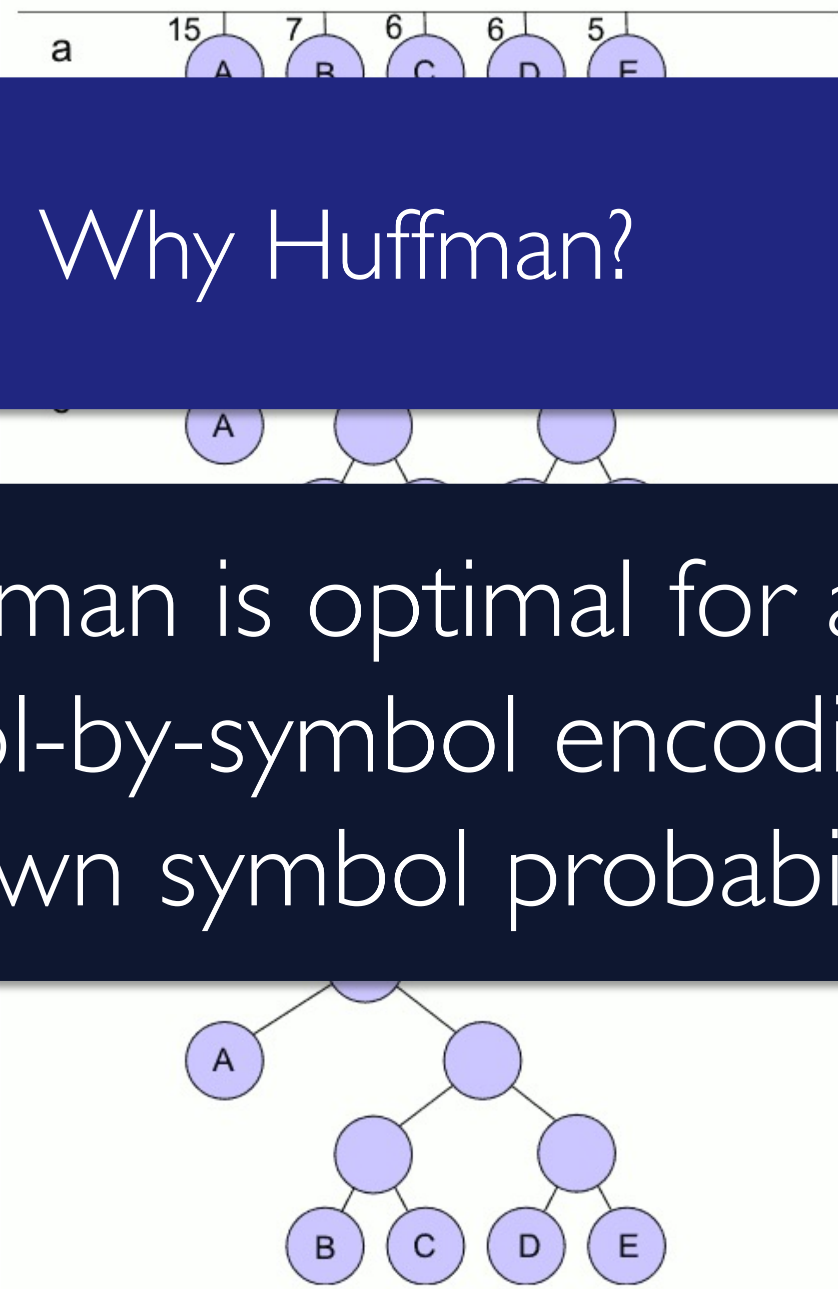
Huffman Encoding



Huffman Encoding



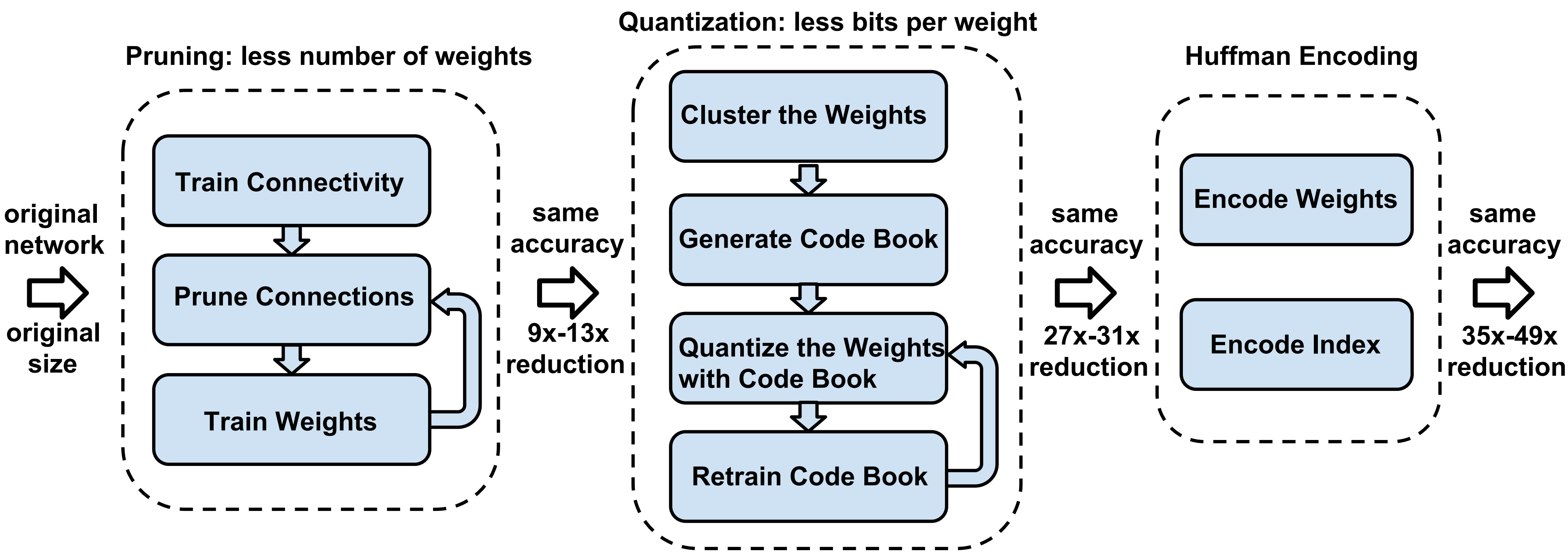
Huffman Encoding



Why Huffman?

Huffman is optimal for a symbol-by-symbol encoding and known symbol probabilities

Deep Compression



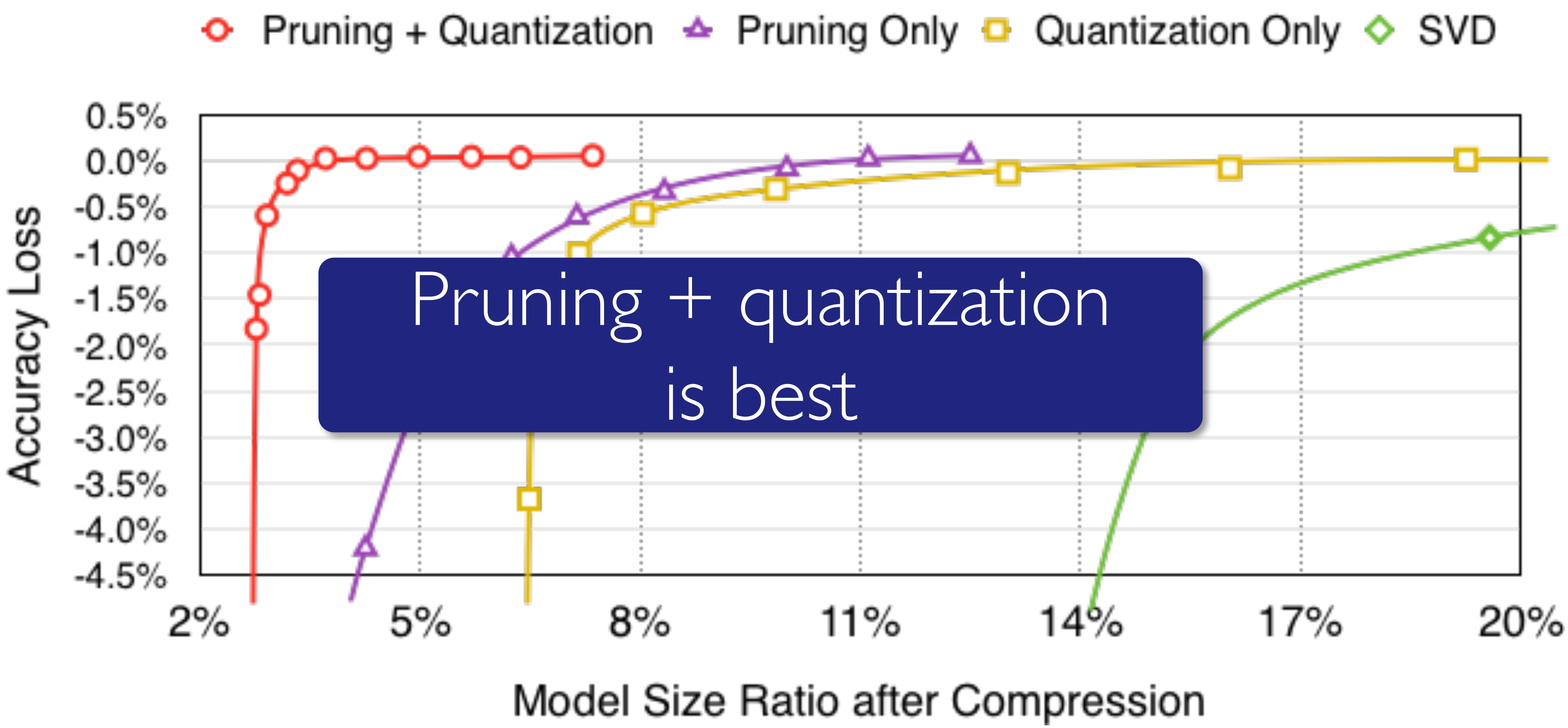
[Han, Mao, Dally, ICLR 2016]

Deep Compression: Experiments

Network	Top-1 Error	Top-5 Error	Parameters	Compress Rate
LeNet-300-100 Ref	1.64%	-	1070 KB	
LeNet-300-100 Compressed	1.58%	-	**27 KB**	**40**$\times$
LeNet-5 Ref	0.80%	-	1720 KB	
LeNet-5 Compressed	0.74%	-	**44 KB**	**39**$\times$
AlexNet Ref	42.78%	19.73%	240 MB	
AlexNet Compressed	42.78%	19.70%	**6.9 MB**	**35**$\times$
VGG-16 Ref	31.50%	11.32%	552 MB	
VGG-16 Compressed	31.17%	10.91%	**11.3 MB**	**49**$\times$

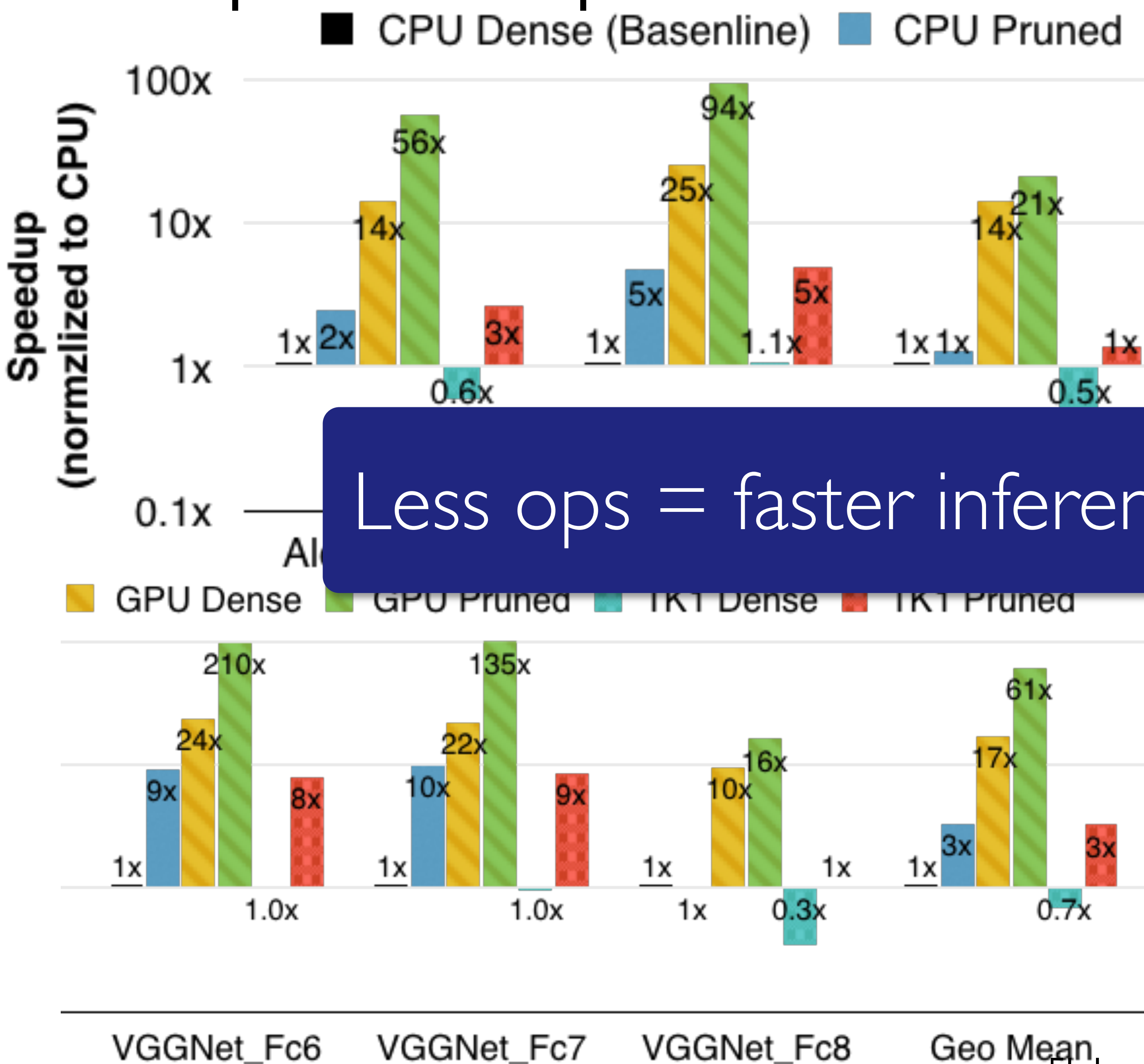
[Han, Mao, Dally, ICLR 2016]

Deep Compression: Experiments



[Han, Mao, Dally, ICLR 2016]

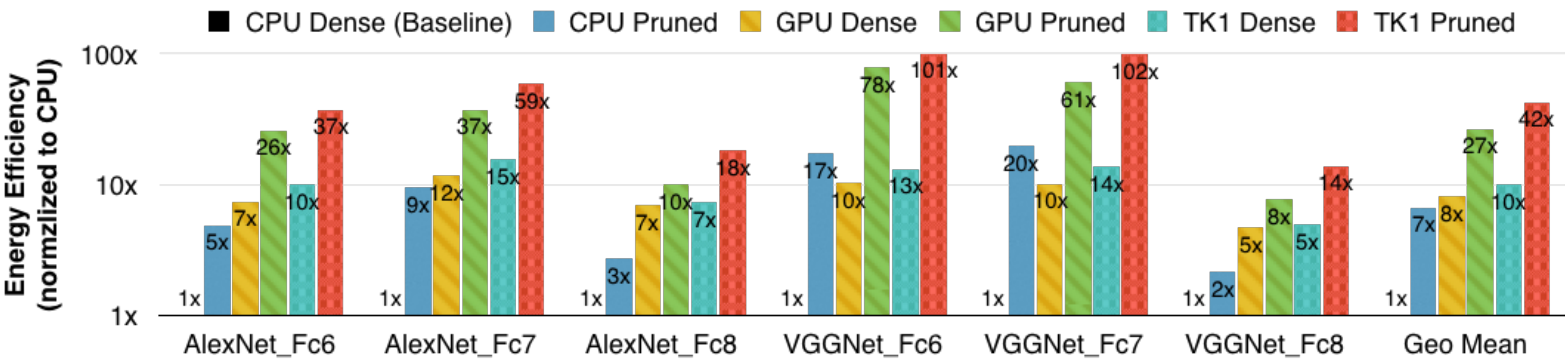
Deep Compression: Experiments



Less ops = faster inference

[Han, Mao, Dally, ICLR 2016]

Deep Compression: Experiments



Less ops = less energy

[Han, Mao, Dally, ICLR 2016]

Deep Compression: Experiments

Table 6: Accuracy of AlexNet with different aggressiveness of weight sharing and quantization. 8/5 bit quantization has no loss of accuracy; 8/4 bit quantization, which is more hardware friendly, has negligible loss of accuracy of 0.01%; To be really aggressive, 4/2 bit quantization resulted in 1.99% and 2.60% loss of accuracy.

#CONV bits / #FC bits	Top-1 Error	Top-5 Error	Top-1 Error Increase	Top-5 Error Increase
32bits / 32bits	42.78%	19.73%	-	-
8 bits / 5 bits	42.78%	19.70%	0.00%	-0.03%
8 bits / 4 bits	42.79%	19.73%	0.01%	0.00%
4 bits / 2 bits	44.77%	22.33%	1.99%	2.60%

Quantized models are accurate

[Han, Mao, Dally, ICLR 2016]

Remarks

- Several interesting papers on model quantization and compression, especially for edge devices/low-power HW
  - Low-rank factorization
  - Training quantization levels
  - SqueezeNets/MobileNets/Ternary Nets/ShuffleNet

- Which one is best?

- Theory for pruned/quantized nets?
  - how many weights can I throw away before I incure an error ε?
  - Use of expanders?
  - Sparse approximation theory?
  - Matrix Sketching?

The end

Reading List

- Song Han, Huizi Mao, William J. Dally, Deep Compression: Compressing Deep Neural Network with Pruning, Trained Quantization and Huffman Coding. ICLR 2016

- Blalock, D., Gonzalez Ortiz, J.J., Frankle, J. and Guttag, J., 2020. What is the state of neural network pruning?. Proceedings of machine learning and systems, 2, pp.129-146.

- Tan, M. and Le, Q., 2019, May. Efficientnet: Rethinking model scaling for convolutional neural networks. In International conference on machine learning (pp. 6105-6114). PMLR.

- Iandola, F.N., Han, S., Moskewicz, M.W., Ashraf, K., Dally, W.J. and Keutzer, K., 2016. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size. arXiv preprint arXiv:1602.07360.

- Liu, Z., Sun, M., Zhou, T., Huang, G. and Darrell, T., 2018. Rethinking the value of network pruning. arXiv preprint arXiv:1810.05270.