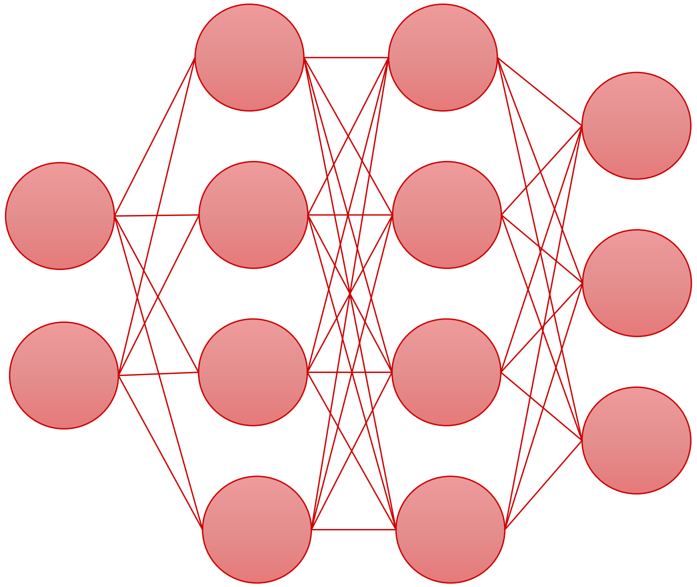
Uncited images created by the author using Adobe Illustrator

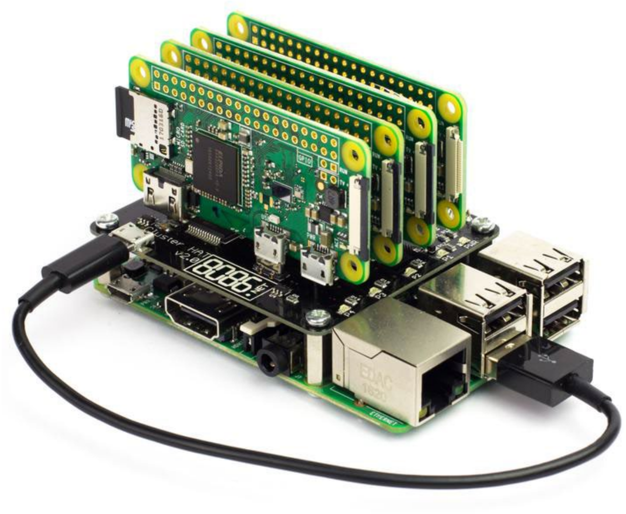
INF2009 – Edge Computing and Analytics [2024/25 T2]

Edge Analytics –
Deep Learning on the Edge

SingaporeTech.edu.sg

Deep Learning on the Edge

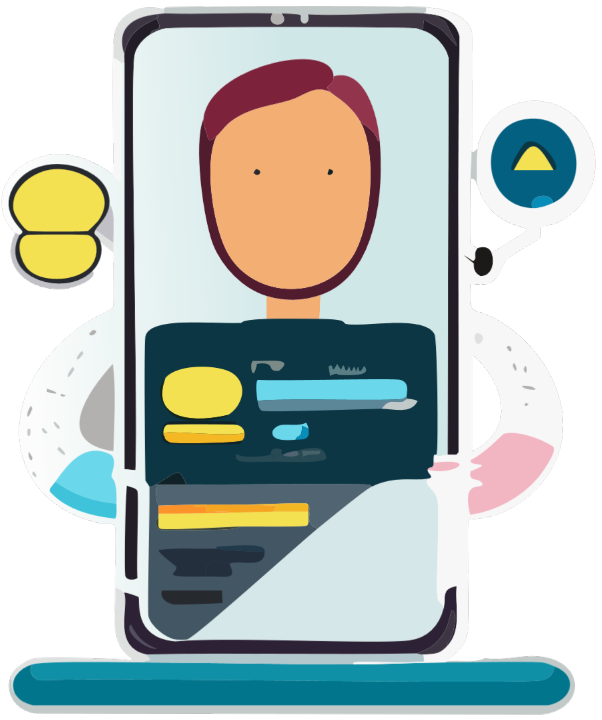




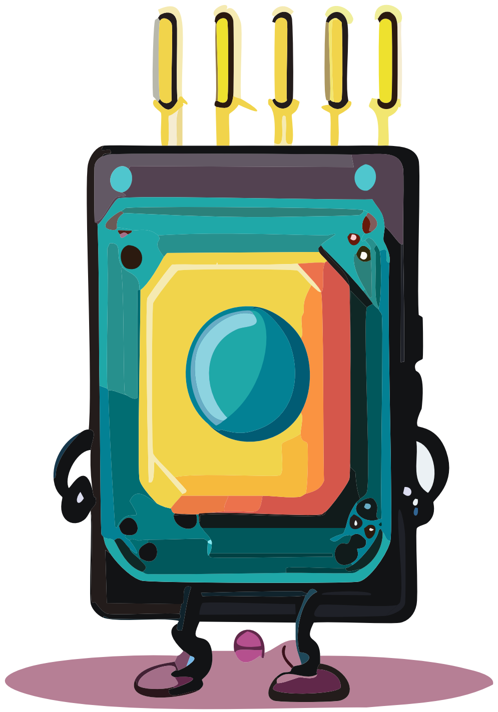
Reference:

- TinyML and Efficient Deep Learning Computing (MIT)
- Machine Learning Hardware and Systems (Cornell Tech)
- tinyML Talks: A Practical Guide to Neural Network Quantization

Contents

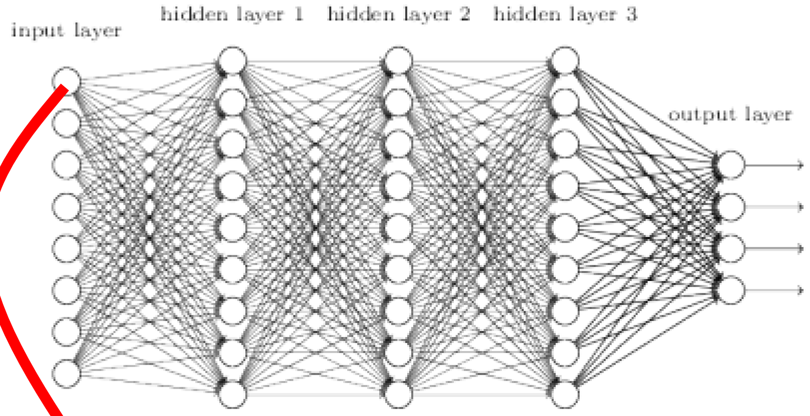


Deep Learning on the Edge

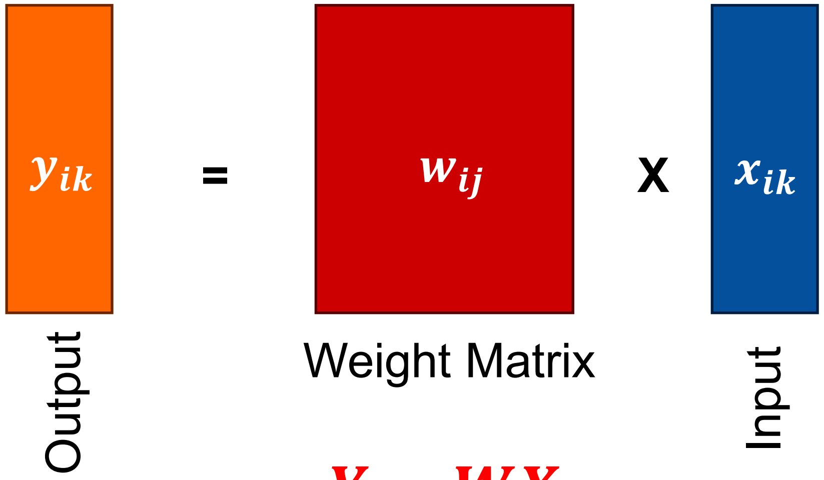


- *Model Size*
  - *Over-the-air update is important*
- *Speed*
  - *Model's inference speed is critical*
- *Energy Efficiency*
  - *Should consume lower battery*

What's Inside a DL and how to make it run on edge?



$$y_i = f\left(\sum_j w_{ij} x_i\right)$$



$$\boldsymbol{Y = WX}$$

*ignoring bias

- Could reduce number of weights (edges) → Pruning
  - Can be done within or across neurons (layers)
- Could reduce the bit size of the weights → Quantization
- Could share weights post quantization → Weight sharing

https://media.nips.cc/Conferences/2015/tutorialslides/Dally-NIPS-Tutorial-2015.pdf

Pruning (in Humans)

50 Trillion Synapses

Newborn

1000 Trillion Synapses

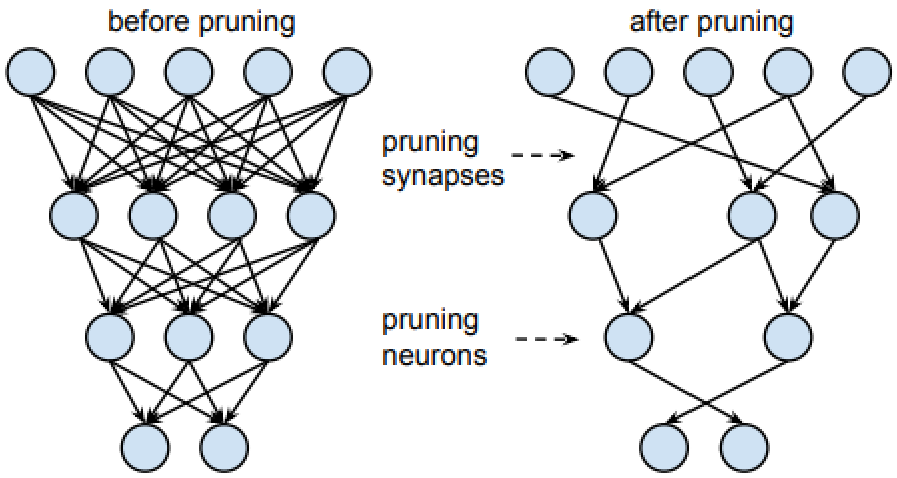
Child

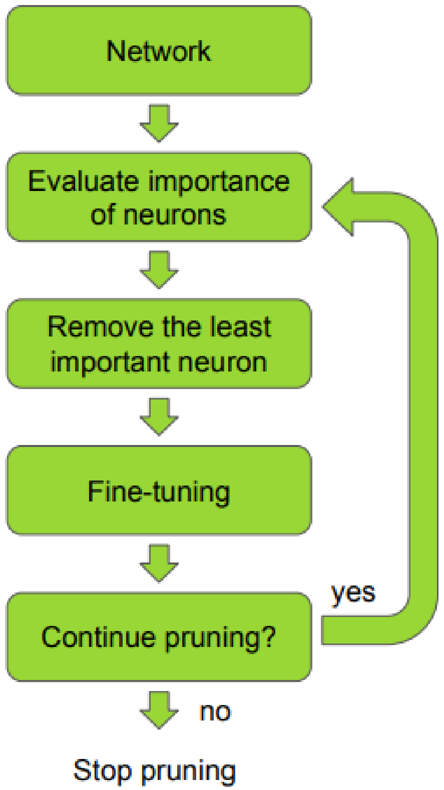
500 Trillion Synapses

Adulthood

Walsh, Christopher A. "Peter Huttenlocher (1931–2013)." *Nature* 502, no. 7470 (2013): 172-172.

Pruning – How?



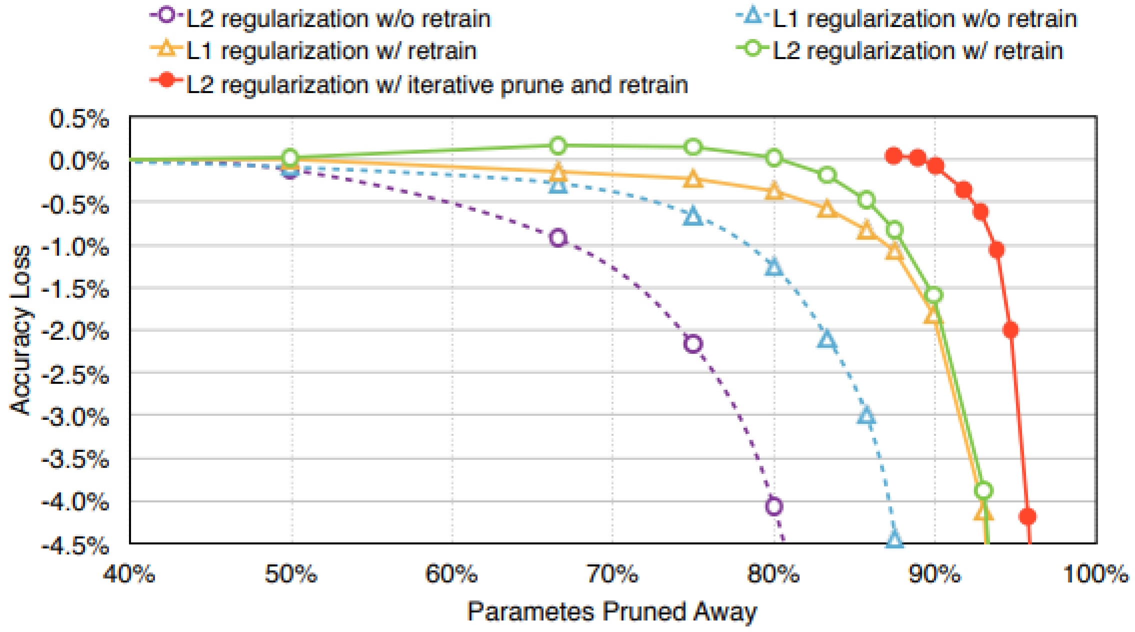


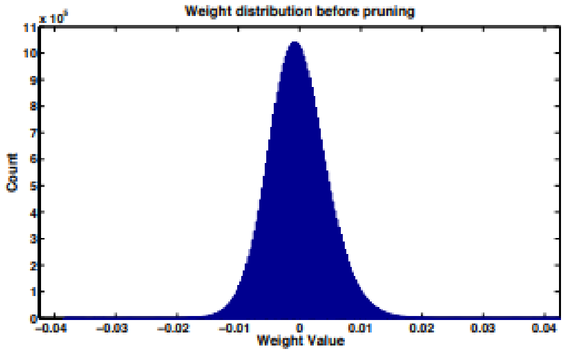
- Consider a set of training examples
  $D = \langle X = \{x_0, x_1, \ldots, x_N\} \mid Y = \{y_0, y_1, \ldots, y_N\} \rangle$
- The network's parameters
  $W = \{(w_1{}^1, b_1{}^1), (w_2{}^2, b_2{}^2), \ldots (w_L{}^{C_l}, b_L{}^{C_l})\}$ are optimized to minimize a cost value $C(D|W)$.
- The most common choice for a cost function $C(\cdot)$ is a negative log-likelihood function.

$$\min_{W'} |C(D|W') - C(D|W)| \quad s.t. \|W'\|_{l=0,1,2} \le B$$

Molchanov, P., S. Tyree, T. Karras, T. Aila, and J. Kautz. "Pruning convolutional neural networks for resource efficient inference." In *5th International Conference on Learning Representations, ICLR 2017.*
Han, Song, Jeff Pool, John Tran, and William Dally. "Learning both weights and connections for efficient neural network." *Advances in neural information processing systems* 28 (2015).

Pruning - Results





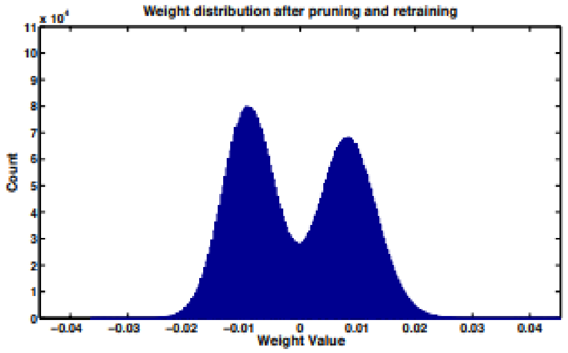


Figure 7: Generate sentence with pretrained model and 90% pruned model

Pretrained: a brown dog is running through a grassy **field**

Pruned: a brown dog is running through a grassy **area**

Figure 13: Generate sentence with pretrained model and 95% pruned model

Pretrained: a soccer player in red is running in the field

Pruned: a man in a red shirt **and black and white black shirt** is running through a field

Han, Song, Jeff Pool, John Tran, and William Dally. "Learning both weights and connections for efficient neural network." *Advances in neural information processing systems* 28 (2015).

Tang, Shijian, and Jiang Han. "A pruning based method to learn both weights and connections for LSTM." *Advances in Neural Information Processing Systems, NIPS, Montreal, QC, Canada* (2015): 7-12.

Pruning - Results

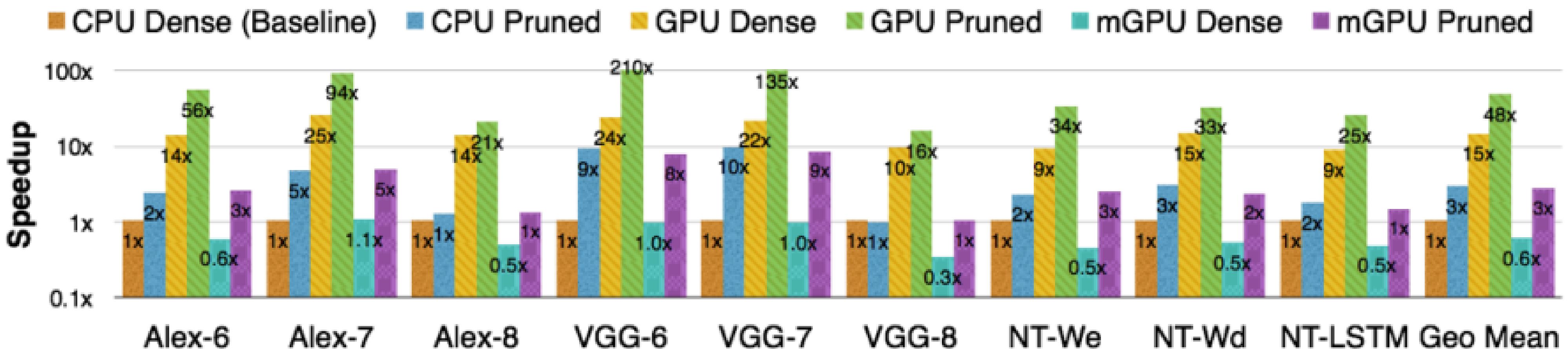
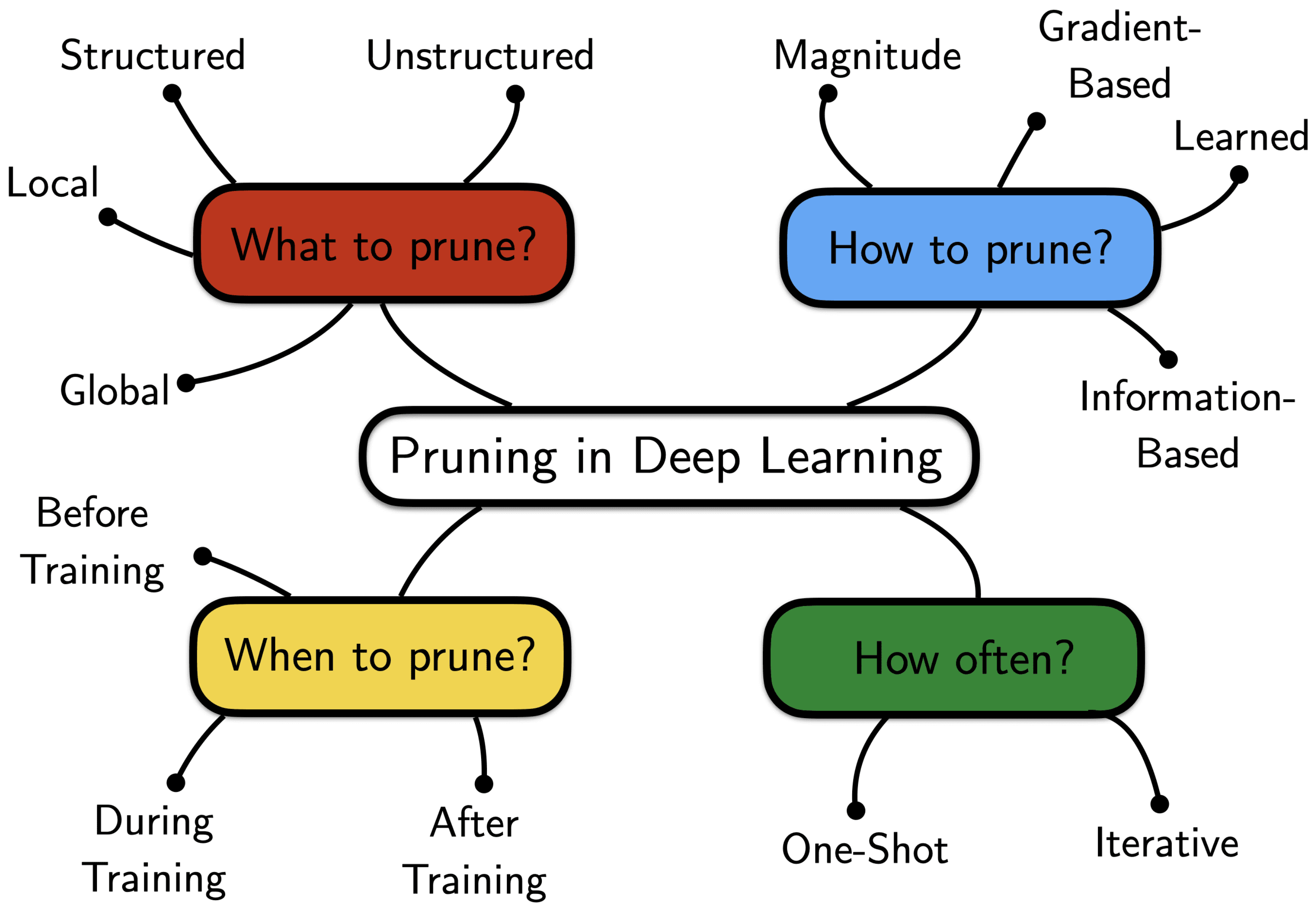


Figure 9: Compared with the original network, pruned network layer achieved 3× speedup on CPU, 3.5× on GPU and 4.2× on mobile GPU on average. Batch size = 1 targeting real time processing. Performance number normalized to CPU.

Intel Core i7 5930K: MKL CBLAS GEMV, MKL SPBLAS CSRMV
NVIDIA GeForce GTX Titan X: cuBLAS GEMV, cuSPARSE CSRMV
NVIDIA Tegra K1: cuBLAS GEMV, cuSPARSE CSRMV

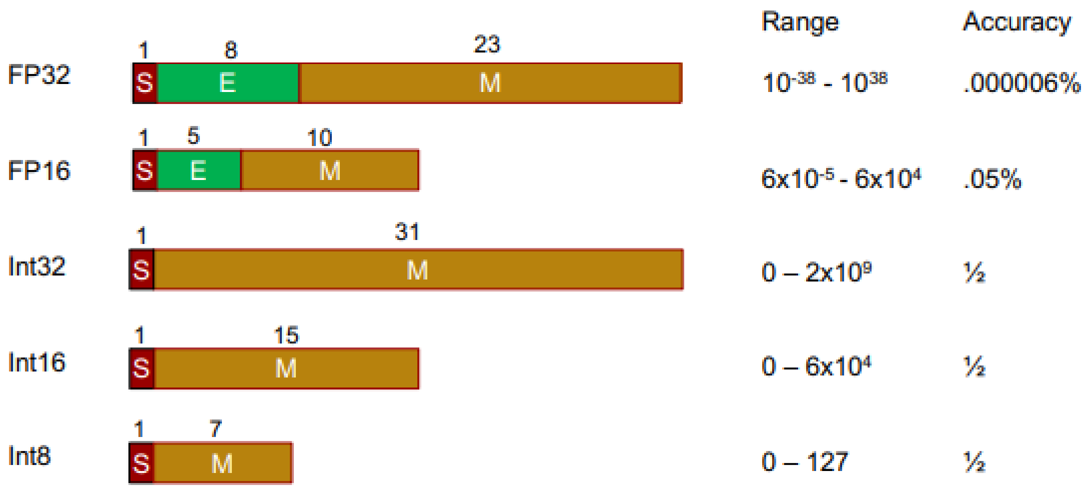
https://berkeley-deep-learning.github.io/cs294-dl-f16/slides/DL_HW_Berkeley_0916.pdf

What, When and How often to prune?



https://roberttlange.com/posts/2020/06/lottery-ticket-hypothesis/

Quantization - Significance

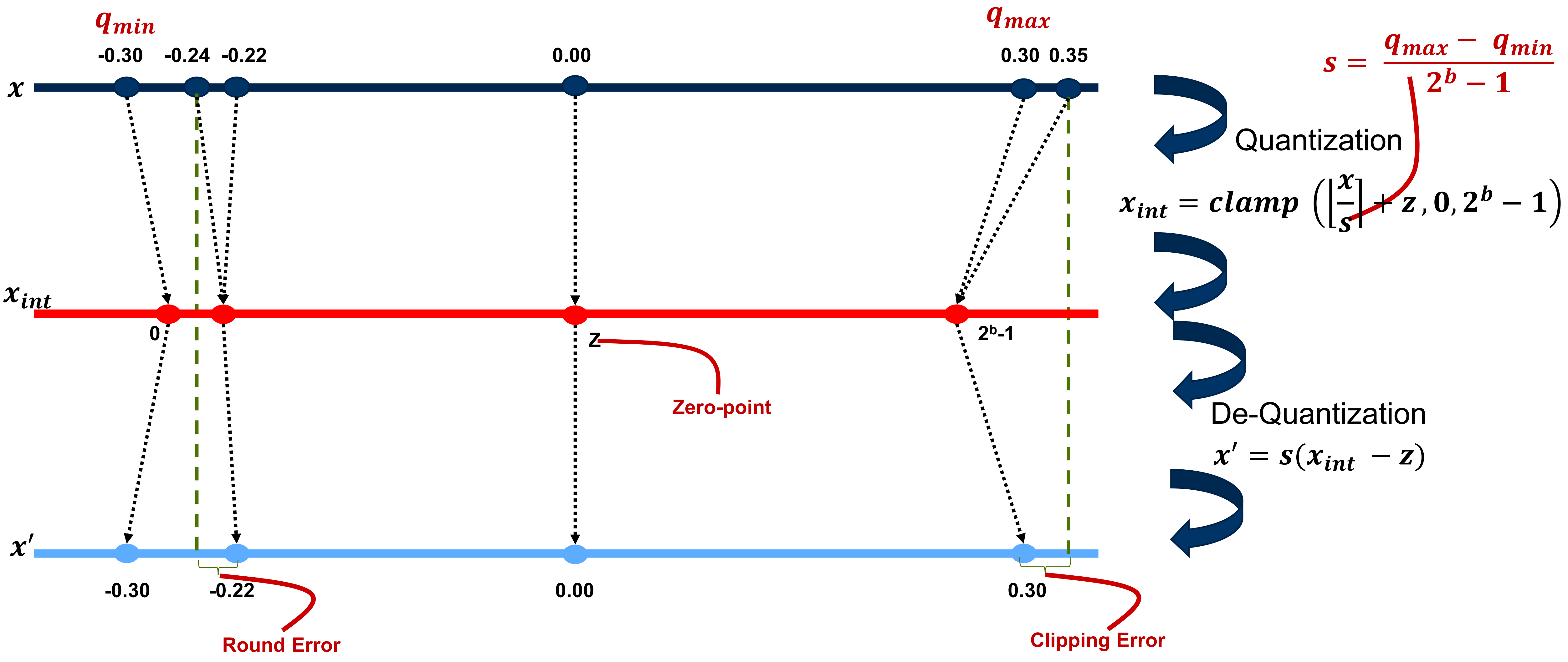


Operation	Precision	Energy (pJ)
Addition	INT8	0.03
	INT16	0.05
	INT32	0.1
	FP16	0.4
	FP32	0.9
Multiplication	INT8	0.2
	INT32	3.1
	FP16	1.1
	FP32	3.7

Significant savings in energy when moving from FP32 (typical NNs) to INT8

Horowitz, Mark. "1.1 computing's energy problem (and what we can do about it)." In *2014 IEEE international solid-state circuits conference digest of technical papers (ISSCC)*, pp. 10-14. IEEE, 2014.
https://berkeley-deep-learning.github.io/cs294-dl-f16/slides/DL_HW_Berkeley_0916.pdf

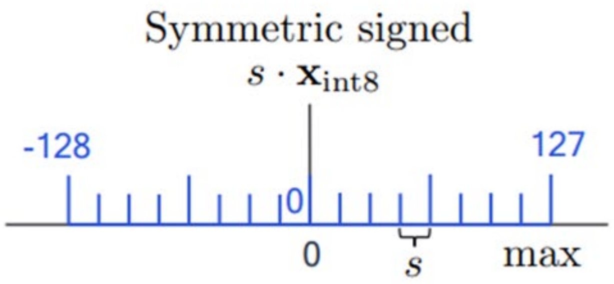
Quantization – How?

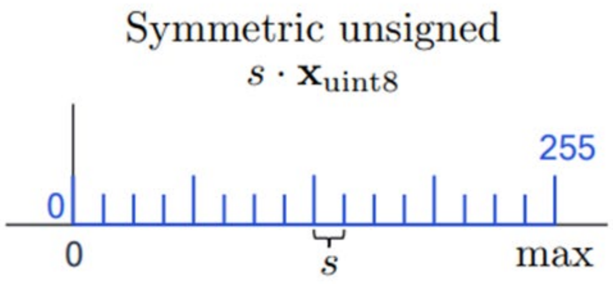


Nagel, Markus, Marios Fournarakis, Rana Ali Amjad, Yelysei Bondarenko, Mart van Baalen, and Tijmen Blankevoort. "A white paper on neural network quantization. arXiv 2021."
https://github.com/hkproj/quantization-notes

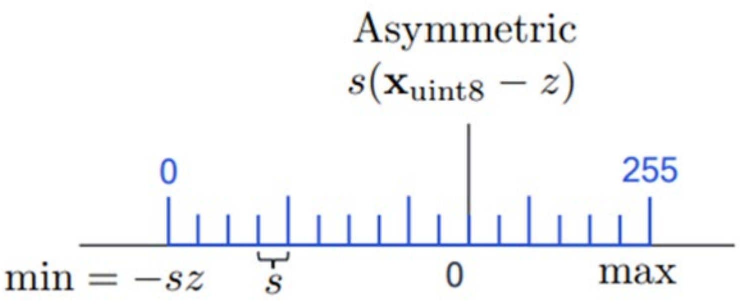
Types of Quantization

Symmetric





Asymmetric



$$Y' = (S_W S_X W_{int})\, X_{int}$$

$$Y' = S_W W_{int}\, S_X X_{int}$$

$$Y = W\, X$$

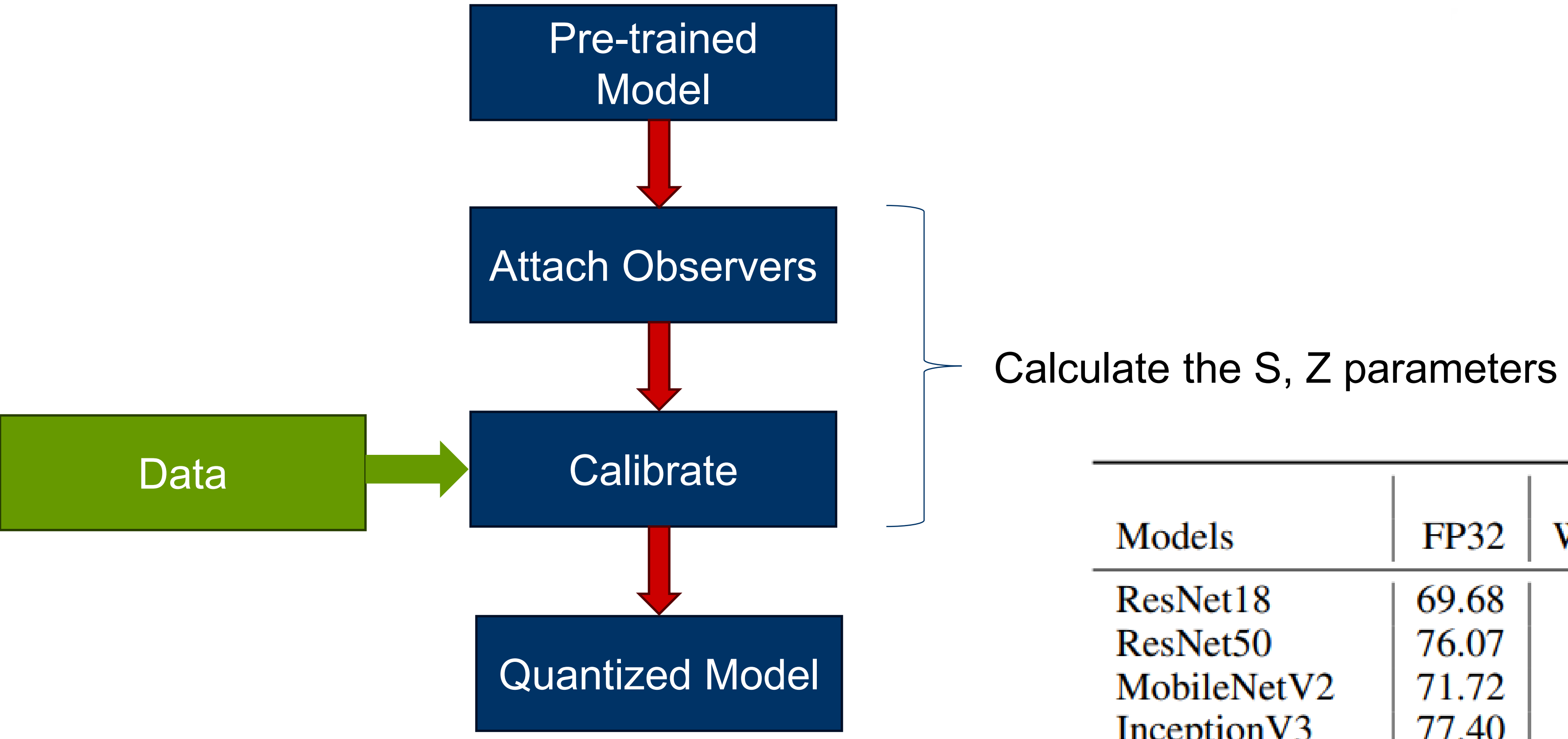
$$Y' = S_W (W_{int} - Z_W)\, S_X (X_{int} - Z_X)$$

$$Y' = (S_W\, S_X\, W_{int})\, X_{int} + S_W S_X Z_W Z_X - S_W S_X Z_X W_{int} - (S_W S_X\, Z_W) X_{int}$$

Additional computational complexity for asymmetric quantization

Nagel, Markus, Marios Fournarakis, Rana Ali Amjad, Yelysei Bondarenko, Mart van Baalen, and Tijmen Blankevoort. "A white paper on neural network quantization. arXiv 2021."

[Option 1] Post Training Quantization

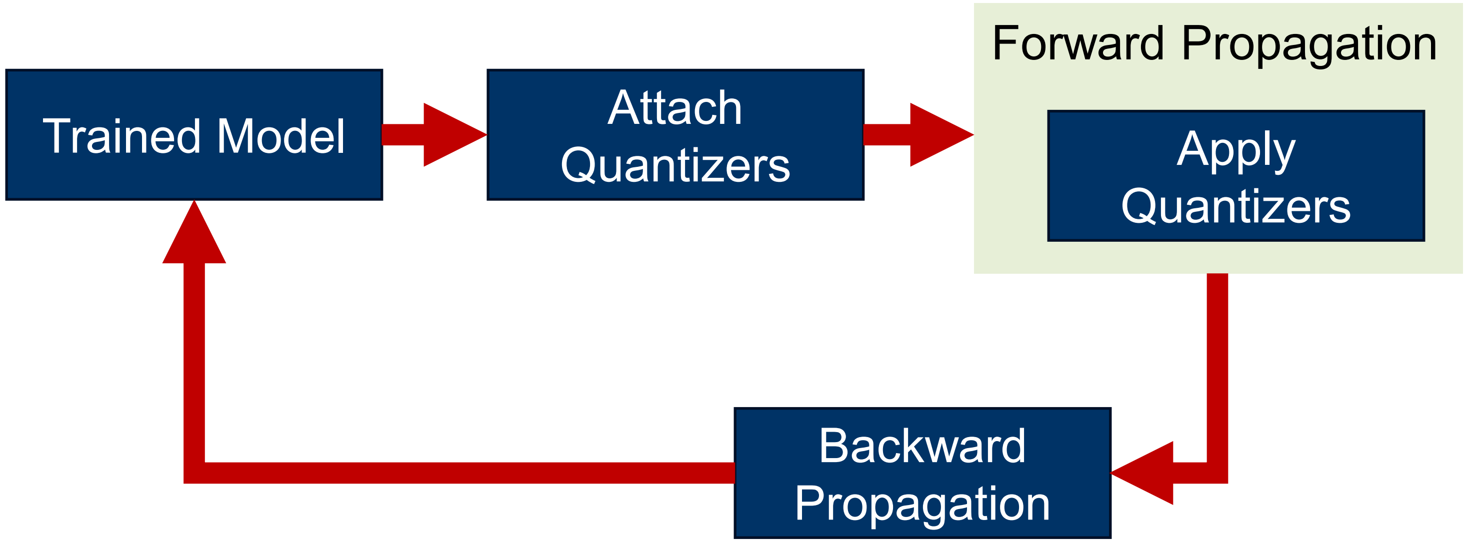


Models	FP32	Per-tensor W8A8	Per-tensor W4A8	Per-channel W8A8	Per-channel W4A8
ResNet18	69.68	69.60	68.62	69.56	68.91
ResNet50	76.07	75.87	75.15	75.88	75.43
MobileNetV2	71.72	70.99	69.21	71.16	69.79
InceptionV3	77.40	77.68	76.48	77.71	76.82
EfficientNet lite	75.42	75.25	71.24	75.39	74.01
DeeplabV3	72.94	72.44	70.80	72.27	71.67
EfficientDet-D1	40.08	38.29	0.31	38.67	35.08
BERT-base†	83.06	82.43	81.76	82.77	82.02

W→ Weights, A→ Activations

Nagel, Markus, Marios Fournarakis, Rana Ali Amjad, Yelysei Bondarenko, Mart van Baalen, and Tijmen Blankevoort. "A white paper on neural network quantization. arXiv 2021."

[Option 2] Quantization Aware Training

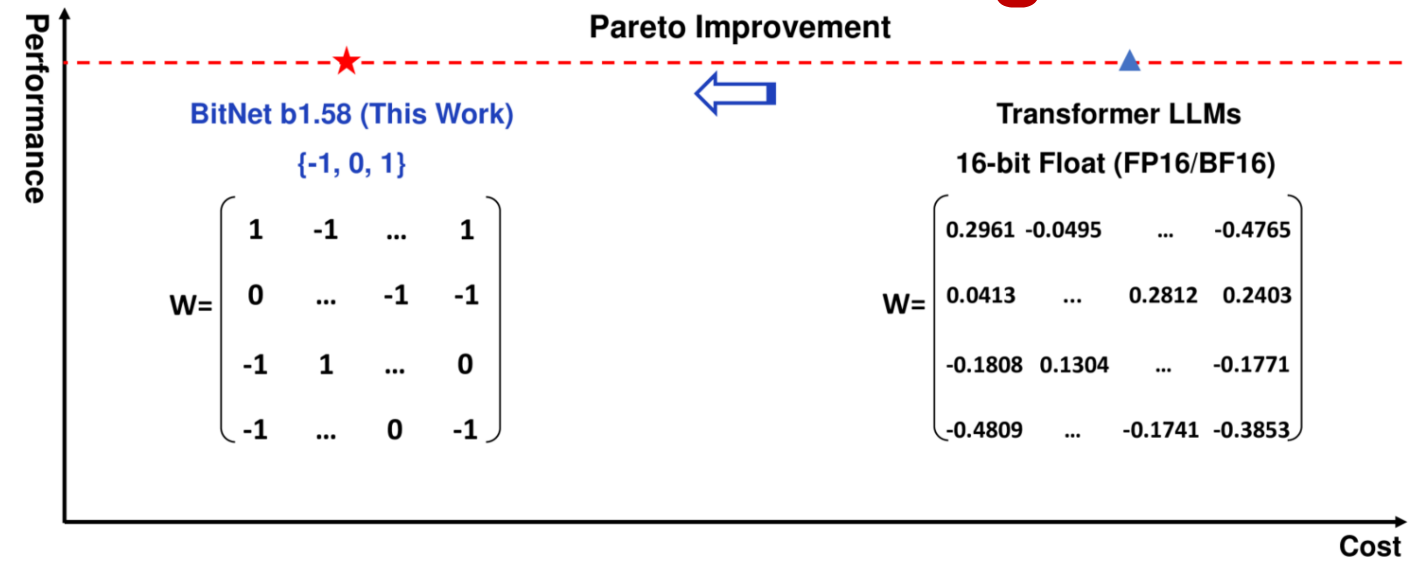


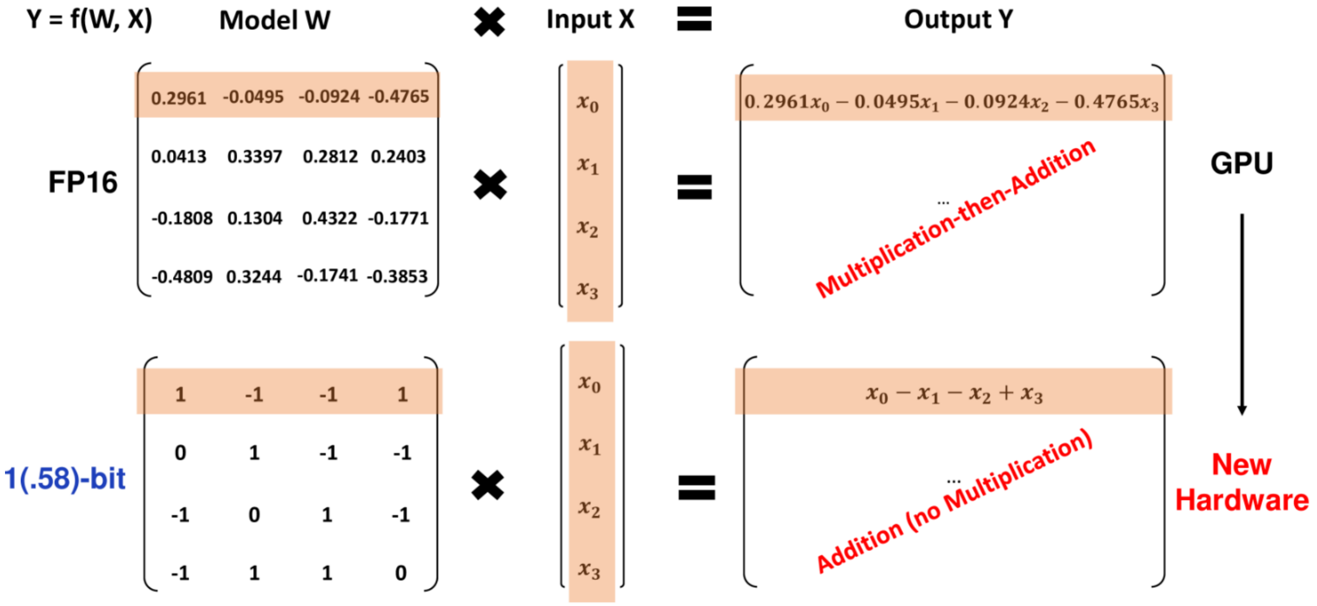
Models	FP32	Per-tensor W8A8	Per-tensor W4A8	Per-tensor W4A4	Per-channel W8A8	Per-channel W4A8	Per-channel W4A4
ResNet18	69.68	70.38	69.76	68.32	70.43	70.01	68.83
ResNet50	76.07	76.21	75.89	75.10	76.58	76.52	75.53
InceptionV3	77.40	78.33	77.84	77.49	78.45	78.12	77.74
MobileNetV2	71.72	71.76	70.17	66.43	71.82	70.48	66.89
EfficientNet lite	75.42	75.17	71.55	70.22	74.75	73.92	71.55
DeeplabV3	72.94	73.99	70.90	66.78	72.87	73.01	68.90
EfficientDet-D1	40.08	38.94	35.34	24.70	38.97	36.75	28.68
BERT-base	83.06	83.26	82.64	78.83	82.44	82.39	77.63

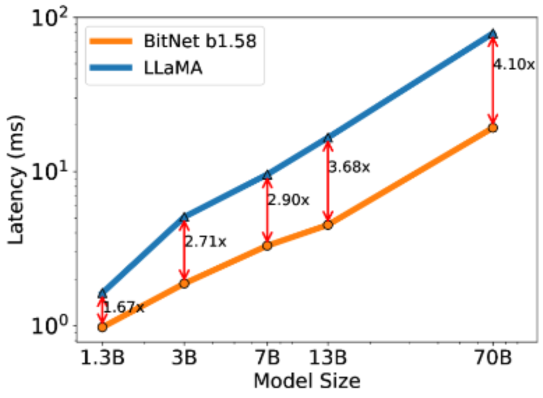
W→ Weights, A→ Activations

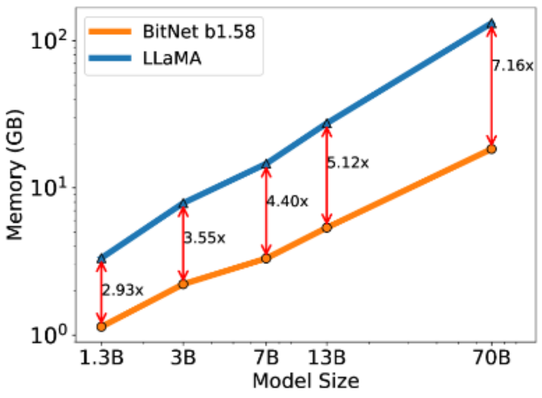
Nagel, Markus, Marios Fournarakis, Rana Ali Amjad, Yelysei Bondarenko, Mart van Baalen, and Tijmen Blankevoort. "A white paper on neural network quantization. arXiv 2021."

Quantization gone extreme!



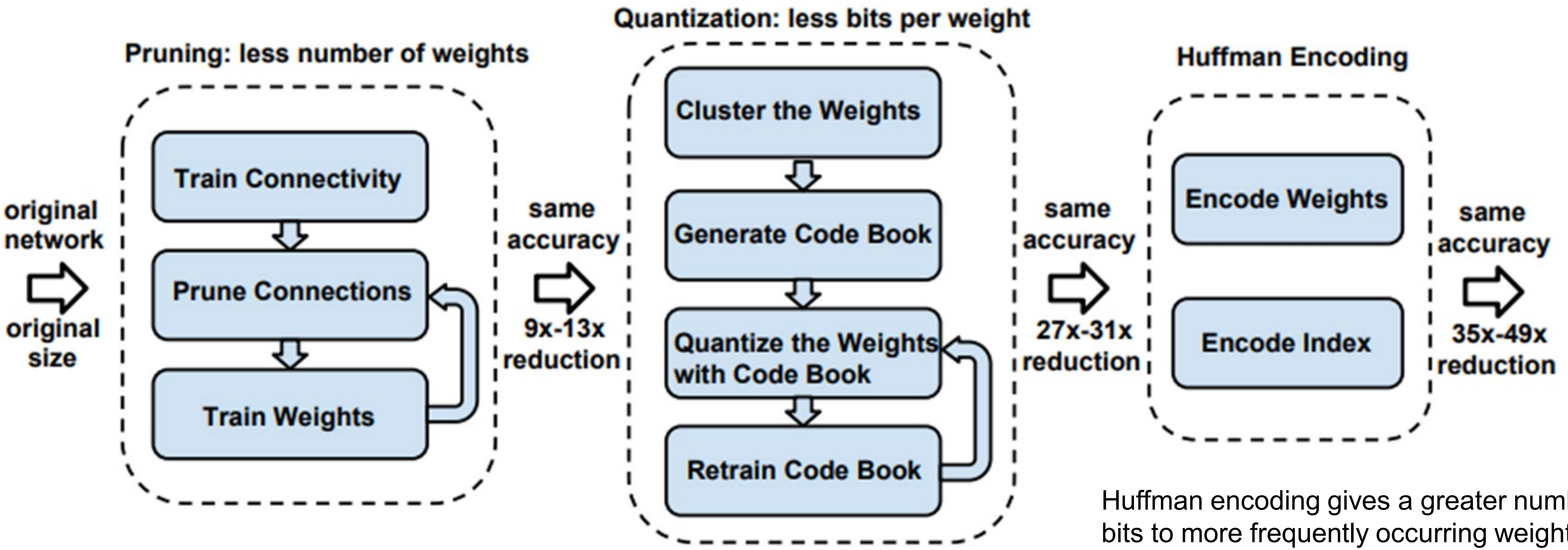






Ma, Shuming, Hongyu Wang, Lingxiao Ma, Lei Wang, Wenhui Wang, Shaohan Huang, Li Dong, Ruiping Wang, Jilong Xue, and Furu Wei. "The Era of 1-bit LLMs: All Large Language Models are in 1.58 Bits." *arXiv preprint arXiv:2402.17764* (2024).

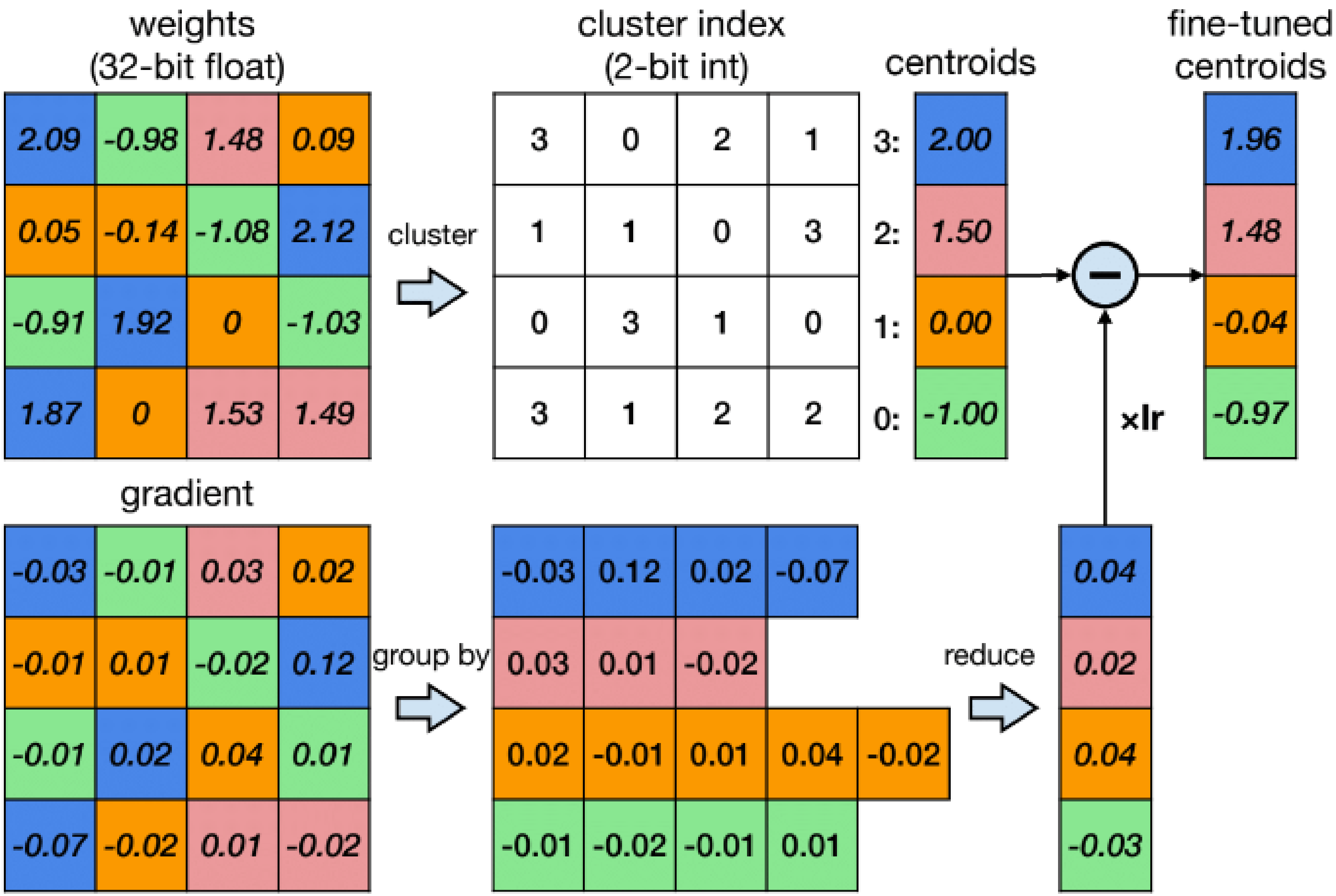
Weight Sharing



Huffman encoding gives a greater number of bits to more frequently occurring weights

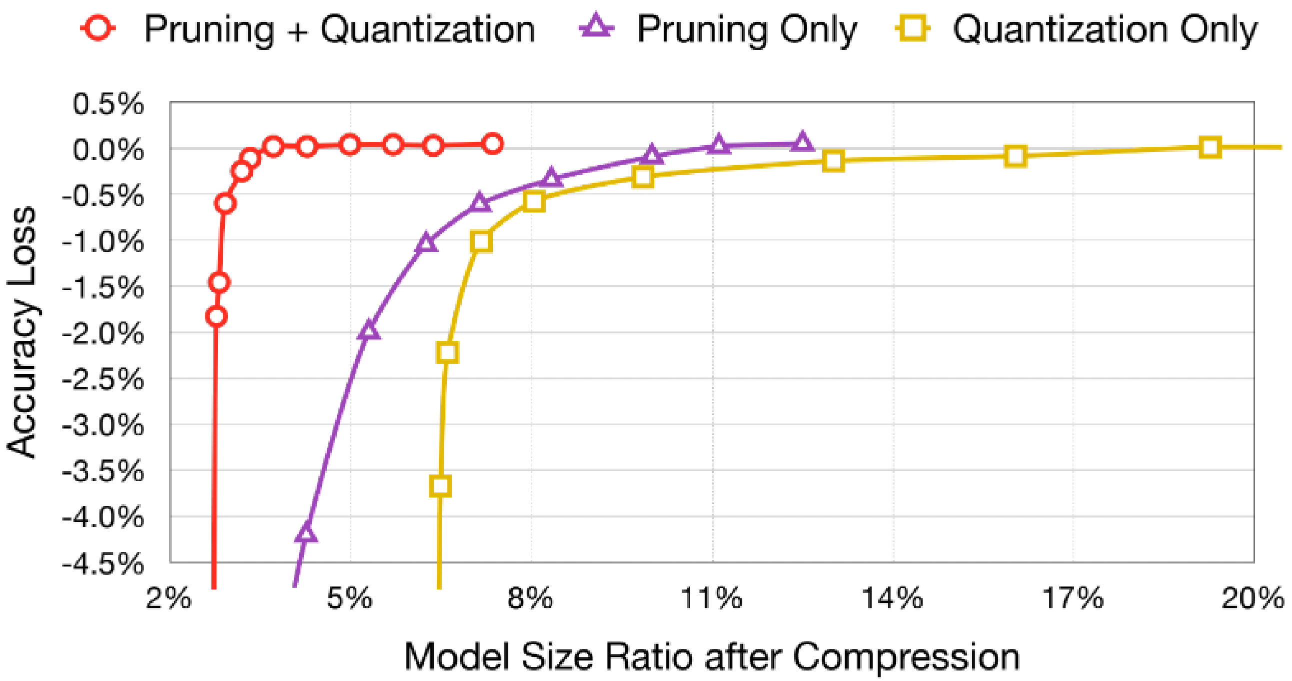
Han, Song, Huizi Mao, and William J. Dally. "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding." *arXiv preprint arXiv:1510.00149* (2015).

Weight Sharing – How?



Han, Song, Huizi Mao, and William J. Dally. "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding." *arXiv preprint arXiv:1510.00149* (2015).

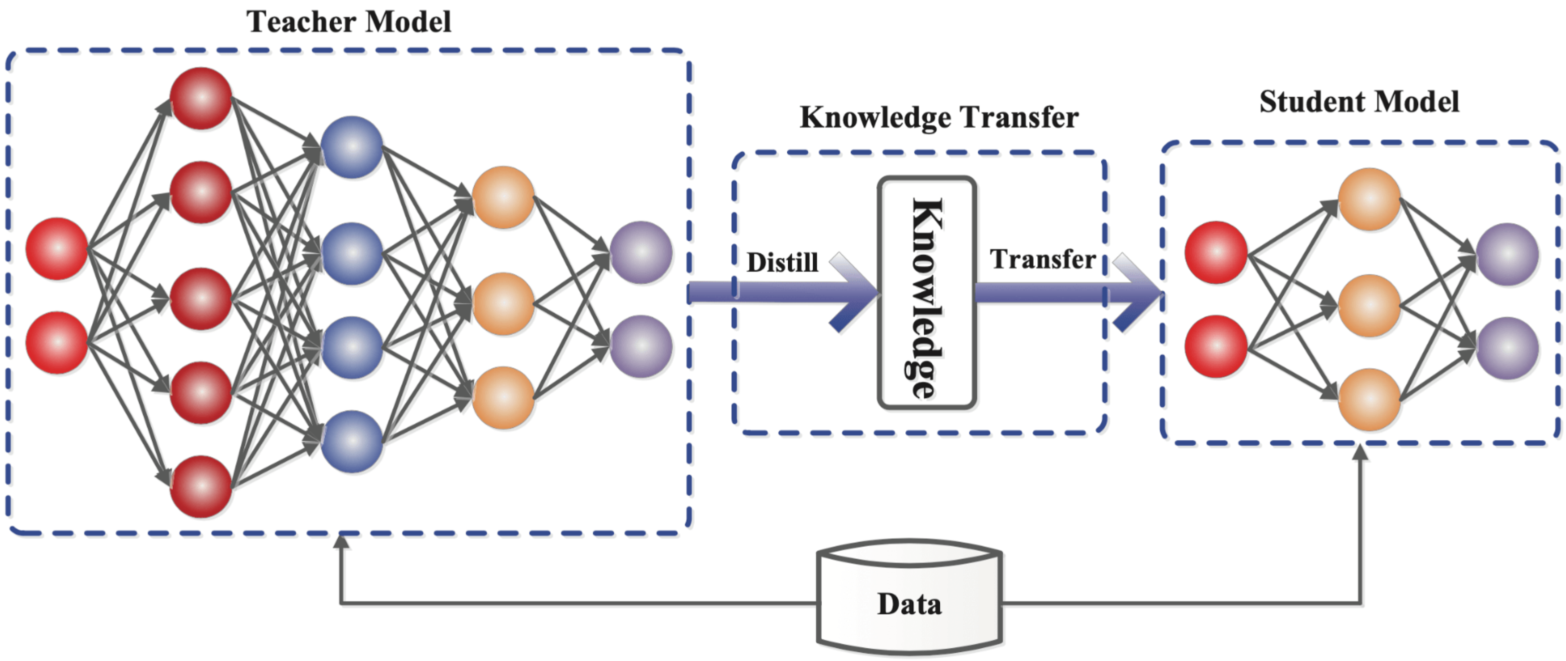
Weight Sharing - Results

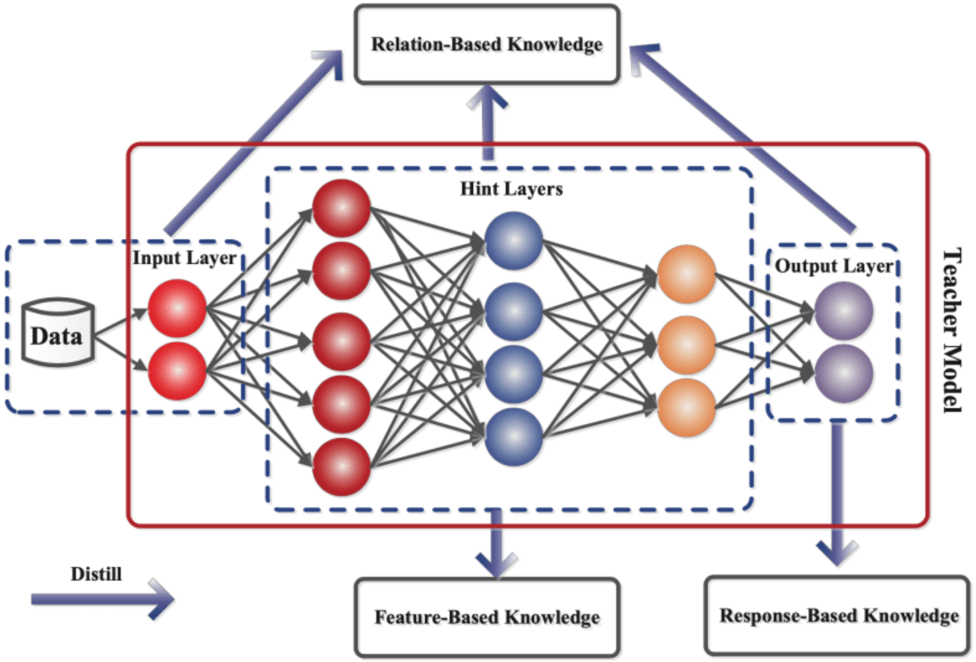


Network	Original Size	Compressed Size	Compression Ratio	Original Accuracy	Compressed Accuracy
LeNet-300	1070KB	27KB	**40x**	98.36%	98.42%
LeNet-5	1720KB	44KB	**39x**	99.20%	99.26%
AlexNet	240MB	6.9MB	**35x**	80.27%	80.30%
VGGNet	550MB	11.3MB	**49x**	88.68%	89.09%
GoogleNet	28MB	2.8MB	**10x**	88.90%	88.92%
ResNet-18	44.6MB	4.0MB	**11x**	89.24%	89.28%

Han, Song, Huizi Mao, and William J. Dally. "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding." *arXiv preprint arXiv:1510.00149* (2015).

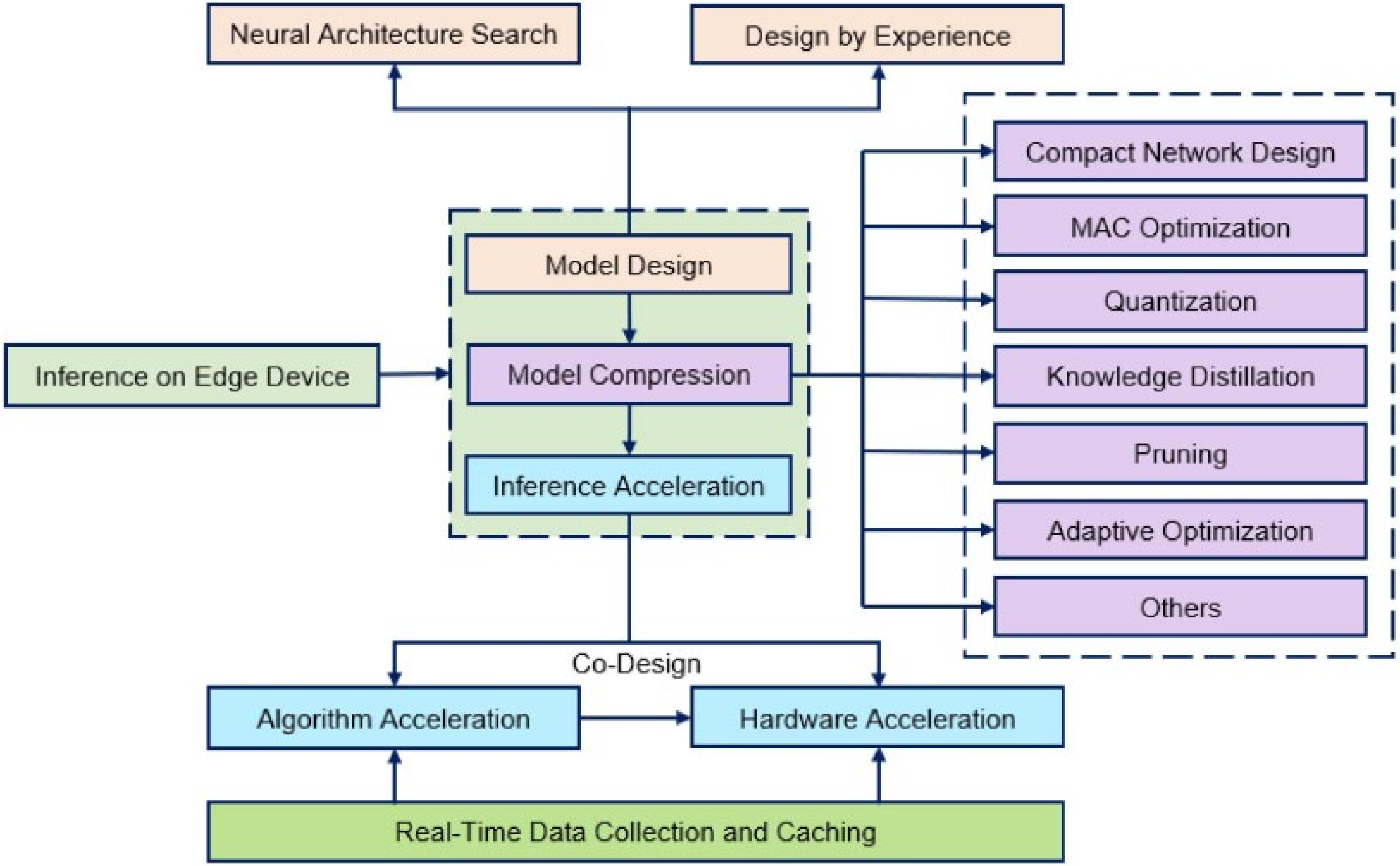
[Other] Options – Knowledge Distillation





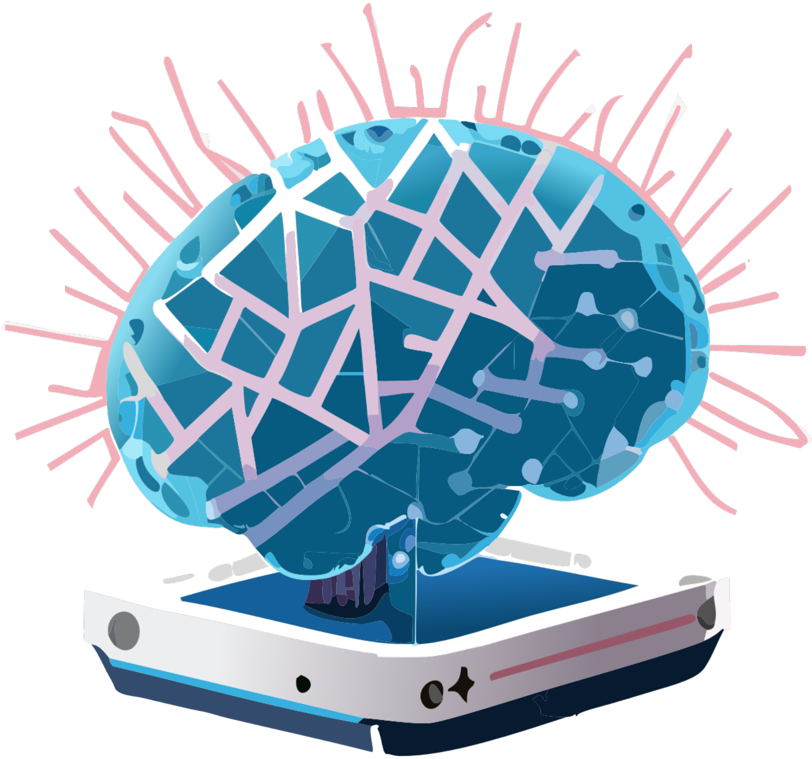
Gou, Jianping, Baosheng Yu, Stephen J. Maybank, and Dacheng Tao. "Knowledge distillation: A survey." *International Journal of Computer Vision* 129, no. 6 (2021): 1789-1819.

Summary – DL on Edge



Shuvo, Md Maruf Hossain, Syed Kamrul Islam, Jianlin Cheng, and Bashir I. Morshed. "Efficient acceleration of deep learning inference on resource-constrained edge devices: A review." Proceedings of the IEEE 111, no. 1 (2022): 42-91.

Summary



- *Adapting deep neural networks on edge devices is important*

- *Pruning, Quantization and their combinations is a popular approach*