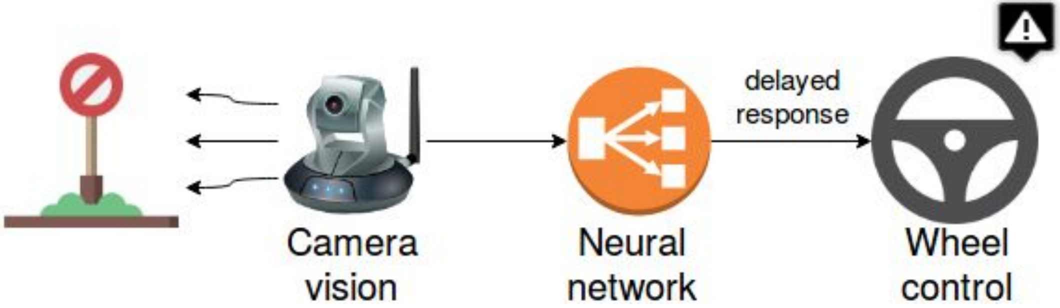
A Comparative Inference Evaluation of Deep Neural Network Compression Methods

Desiana Dien Nurchalifah, Deebul Nair, Paul G. Plöger
June 22, 2020

Introduction

Figure 1. Illustration of Image Classification in Autonomous Driving[1]



False action execution caused by:

- Incorrect classification
- Slow response time

[1] Image adapted from: A. Balakrishnan. (2019).

Why are we doing this?

- With increasing performance, the **number of parameters** is also increased up to 8 times.
- Increasing number of parameters leads to expansion of **memory usage**.

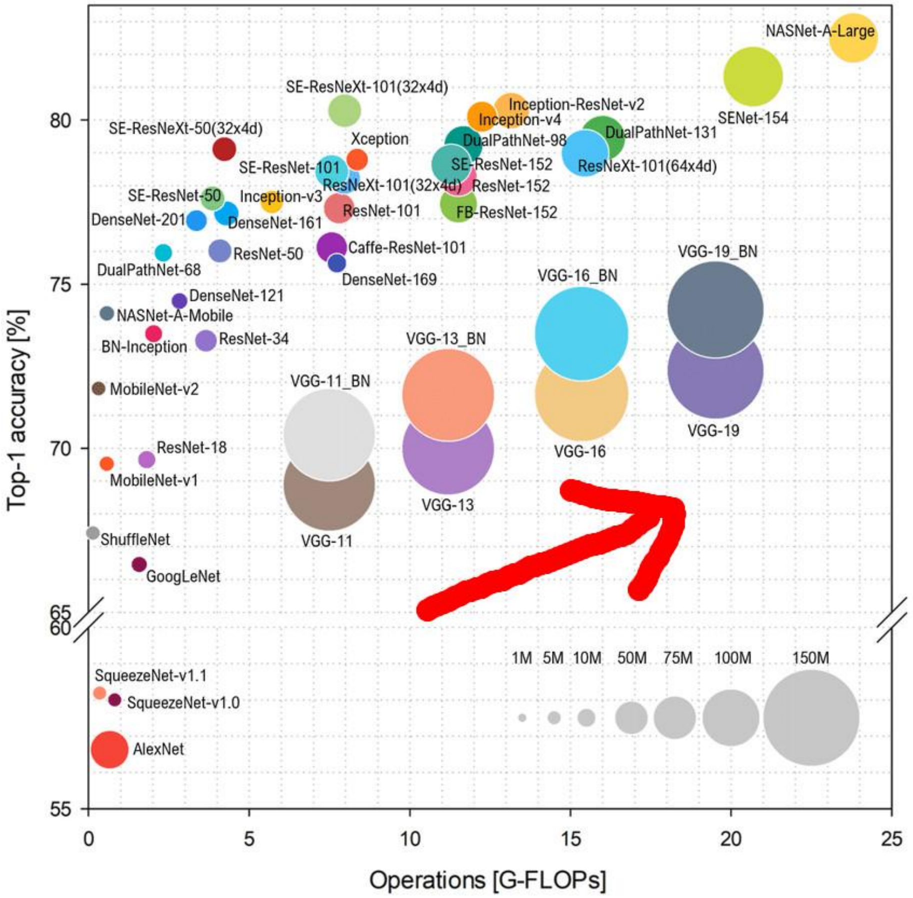


Figure 2. Number of Operations Versus Accuracy of DNN Architecture[2]

[2]Image reference: Bianco et al. (2018).

Challenges

- **Restriction** of embedded system resources:
  - Implementation without GPU
  - Limited memory space

- Multiply-accumulate operations (MACs) and floating point operations (FLOPs) are **not sufficient** to define faster inference

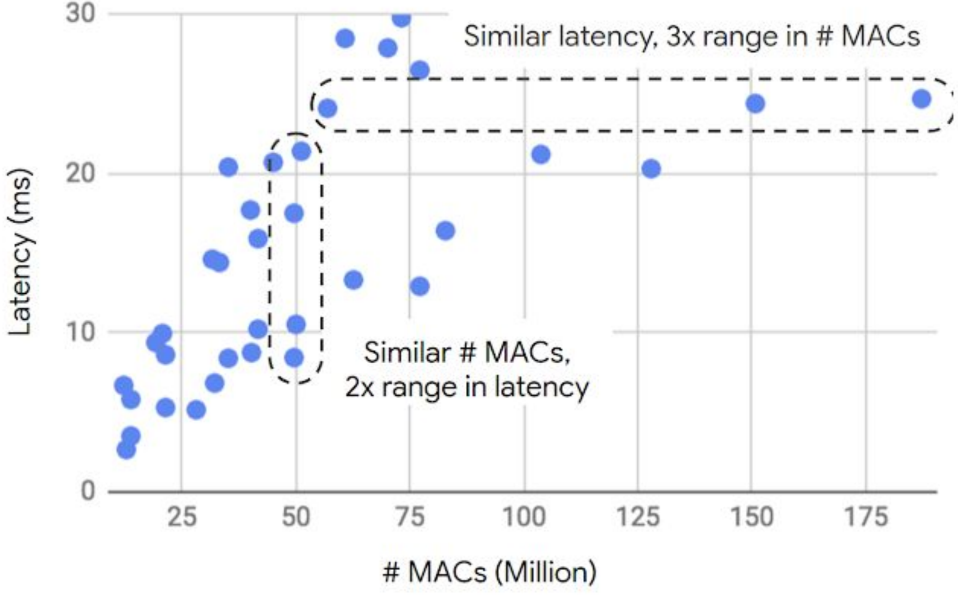
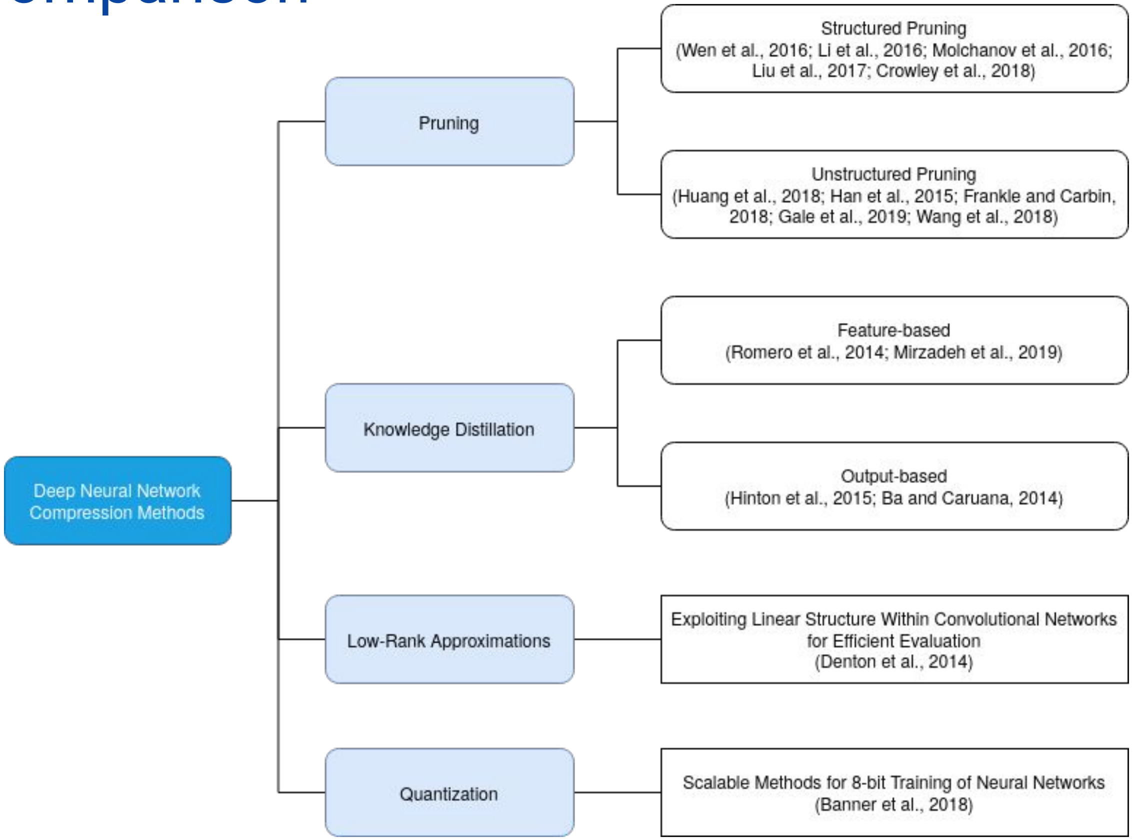


Figure 3. MAC vs Latency on MobileNet[3]

[3]https://ai.googleblog.com/2018/04/introducing-cvpr-2018-on-device-visual.html

Methods Comparison



Review of Methods: Pruning

Unstructured Pruning

- Pruning based on **rank of weights**
- Network has cluttered **sparse** representation

Training → Pruning → Fine Tuning

Training	Pruning	Fine Tuning
Train the data	Prune based on saliency criteria	Fine tune the model after sparsity has been induced

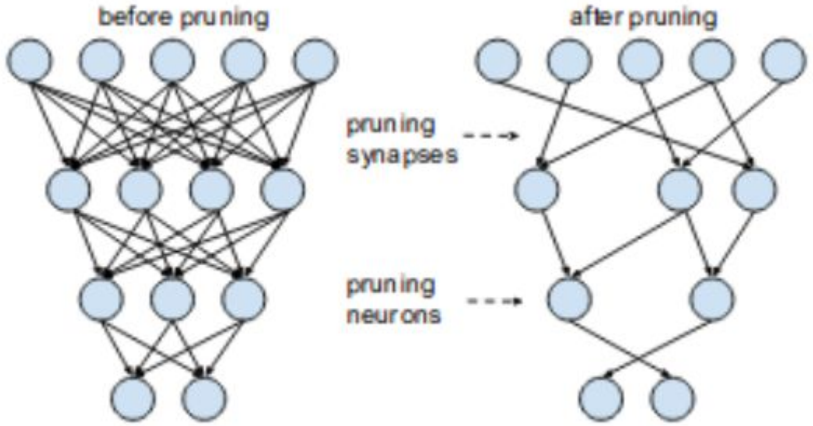


Figure 4. Unstructured Pruning Illustration[4]

[4]Han, S. et al. *Learning both Weights and Connections for Efficient Neural Networks*. NIPS. (2015)

Review of Methods: Pruning

Structured Pruning

- Prune based on **grouped penalty** (layers, channels, filters)
- Effective representation on CPU

L1 Pruning based on the work (Liu et al., 2018), pruning filters algorithm is as follows:

1. Train deep network
2. For each convolution layer, calculate sum value
3. Sort sum values
4. Remove smallest m and corresponding feature maps
5. Create new layer by copying non-pruned filters

$$\triangle_c = \sum \| \mathit{weights} \|$$

Review of Methods: Pruning

Fisher Pruning (Theis et al., 2018)

- **Greedily remove parameters** one by one where delta loss is the smallest
- With:
    - N : number of examples
    - W: channel spatial width
    - H: channel spatial height
    - g: gradients of parameters w.r.t n^{th} data
    - A: activation of n^{th} data point

$$\Delta_c = \frac{1}{2N} \sum_{n}^{N} \left(-\sum_{i}^{W} \sum_{j}^{H} A_{nij} g_{nij} \right)^2$$

Review of Methods: Low-Rank Approximations

Approximation of weights matrix by decompose and reconstruct the weight matrix such that matrix consists of lower rank than its original value. With S as the rank approximated, reconstruction is done as follows:

$$y = Wx + b$$

$$\Downarrow$$

$$y = USV^T x + b$$

Review of Methods: Quantization

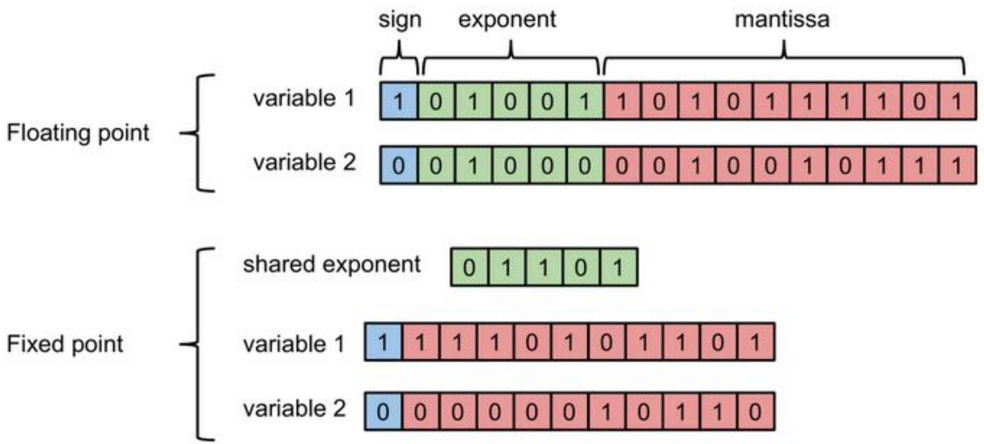
- Weights represented with **fixed value**
- With M bit precision and N vector to be quantized, orientation of weights vector should be preserved in the formula:

$$2^M \gg \sqrt{2ln(N)}$$

- In ResNet-50, 1024 examples yield to:

$$2^8 >> \sqrt{2\ln(3 \times 3 \times 2048 \times 1024)}$$

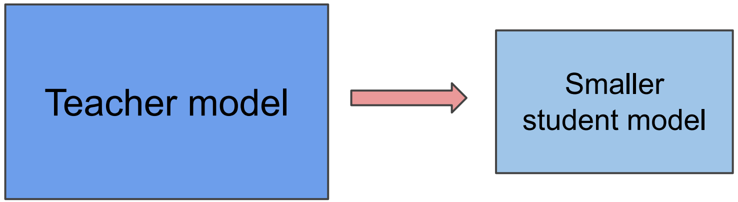
$$256 >> 5.788$$



[5]Image adapted from Courbariaux, et al. “Binarized neural networks: Training deep neural networks with weights and activations constrained to+1 or-1.” (2016).

Review of Methods: Model Distillation

- Main idea: transfer knowledge of model function into smaller model[6]
- Distillation: extraction of the most **important aspect** or the **imperative meaning** in teacher network by student network.



	cow	dog	cat	car
original hard targets	0	1	0	0
output of geometric ensemble	10^{-6}	.9	.1	10^{-9}
softened output of ensemble	.05	.3	.2	.005

[6] Hinton, et al. Distilling the knowledge in a neural network. ArXiv, abs/1503.02531, 2015.

Review of Methods: Knowledge Distillation

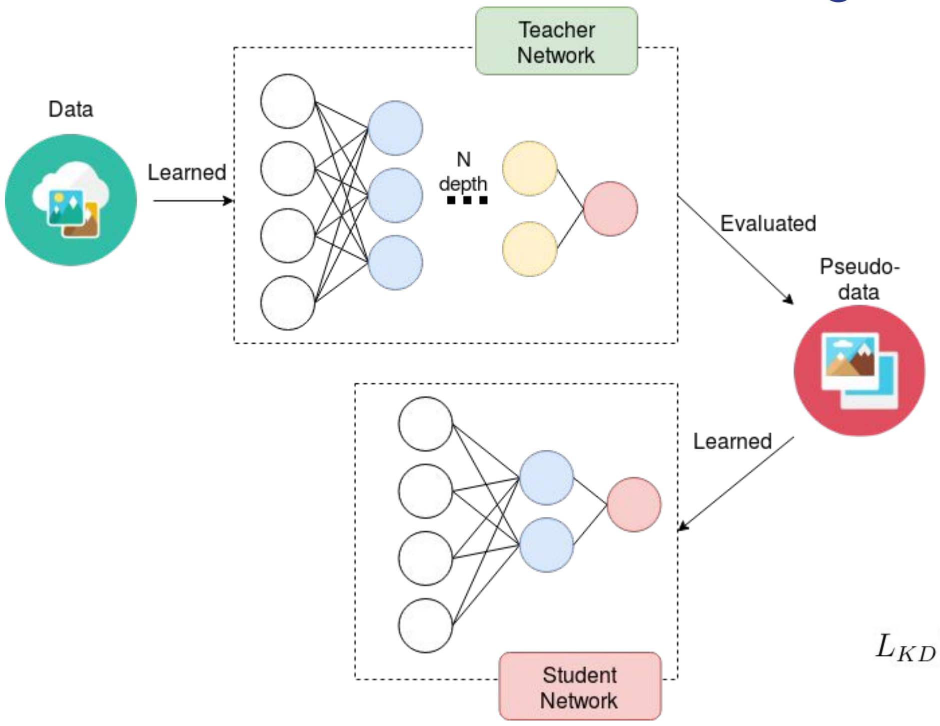


Figure 6. Knowledge Distillation Illustration

Output-based

- With z as logits, obtain knowledge by using softmax temperature:

$$S(i) = \frac{e^{z_i/T}}{\sum_j e^{z_j/T}}$$

- With H as cross-entropy loss and λ as turn-able parameters, distillation loss calculated by:

$$L_{KD}(W_s) = H(y, S(z_s; T = 1)) + \lambda H(S(z_t; T = \tau), S(z_s, T = \tau))$$

Review of Methods: Knowledge Distillation

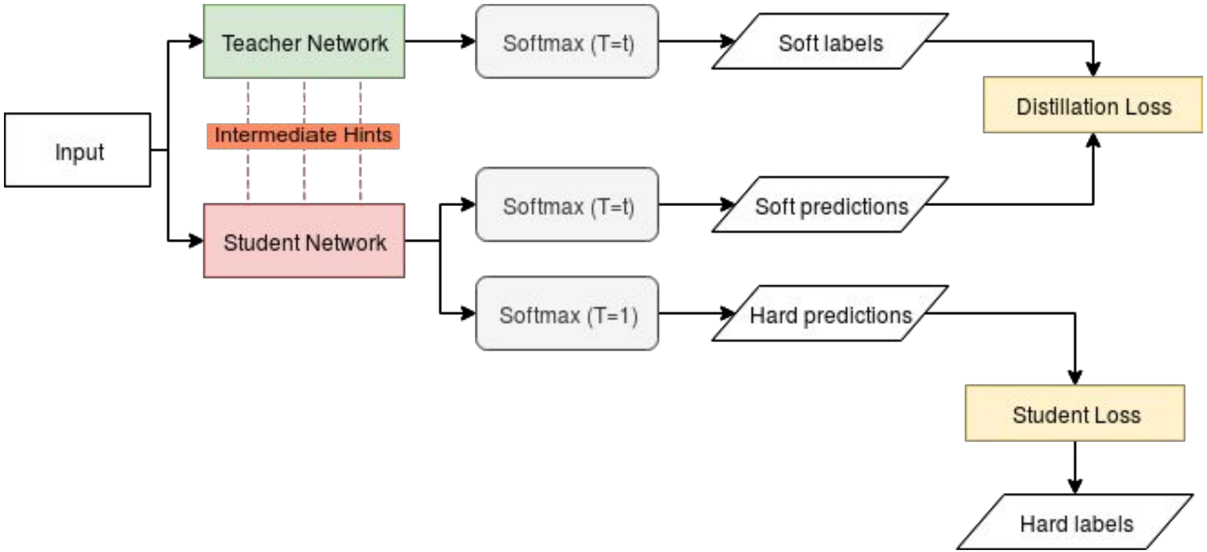


Figure 6. Knowledge Distillation Summary[7]

Feature-based

- Contain **intermediate layer hints**
- With r as regressor and u_h as teacher function and v_g as student function, the loss is minimized as follows:

$$L_{HT}(W_{Guided}, W_r) = \frac{1}{2} \parallel u_h(x; W_{Hint}) - r(v_g(x; W_{Guided}); W_r) \parallel^2$$

[7] Neta Zmora, Guy Jacob, Lev Zlotnik, Bar Elharar, and Gal Novik. Neural network distiller, June 2018. URL https://doi.org/10.5281/zenodo.1297430.

Experiments: Model Compression

Settings:

- 100 iterations each method
- MLMark benchmark[8]

Table 1. Model Compression Accuracy and Parameters Results

Method	Number of Parameters	Reduction Ratio	Top-1 Accuracy
ResNet-56			
Baseline	869530	1	93.13
Weight Pruning	613349	0.705	93.17
L1-Norm	**773336**	0.889	93.24
Fisher	855770	0.984	86.64
PreResNet-110			
Baseline	1146842	1	94.99
Weight Pruning	808408	0.704	94.97
L1-Norm	1088618	0.949	93.51

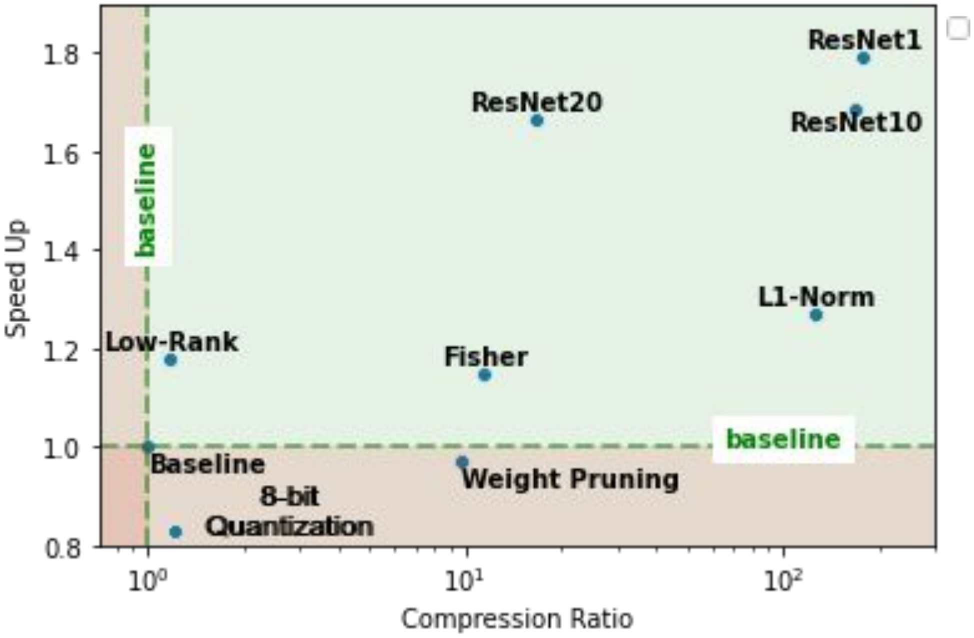


Figure 7. Speedup vs Compression Comparison

[8]https://www.eembc.org/mlmark/

Experiments: Model Compression

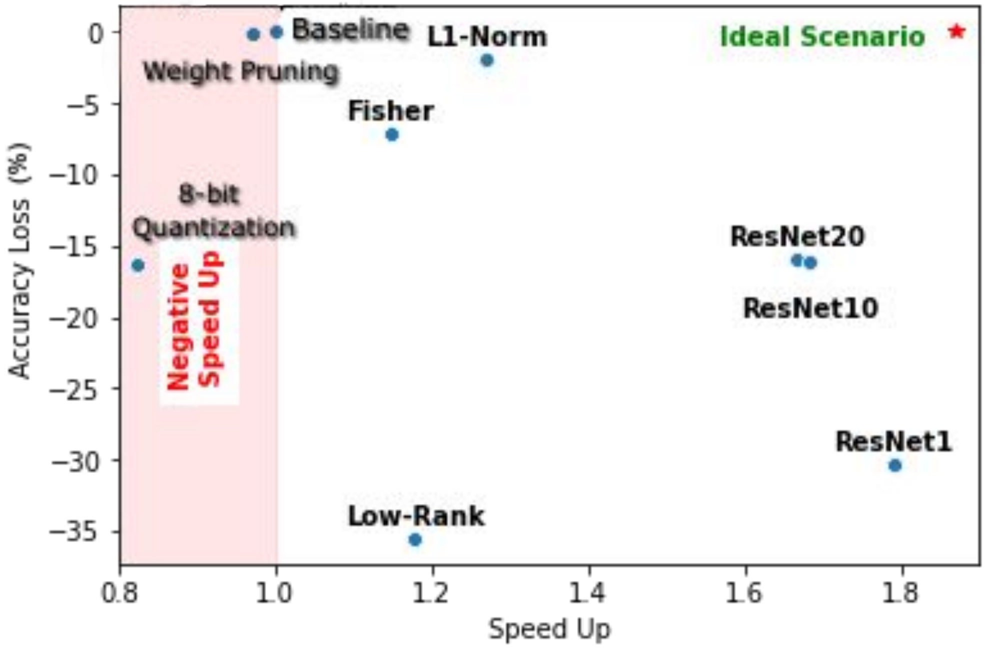


Figure 8. Accuracy vs Speedup Comparison

Evaluation: Model Distillation

Table 2. Model Distillation Accuracy and Parameters Results

Student	Number of Parameters	Reduction Ratio	Top-1 Accuracy KD	Top-1 Accuracy FitNets
ResNet-56 Teacher				
ResNet8	78042	0.089	60.61	**63.63**
ResNet18	175258	0.201	72.42	**77.89**
ResNet26	272474	0.313	74.74	**78.0**
PreResNet-110 Teacher				
ResNet26	272474	0.024	62.10	**76.01**

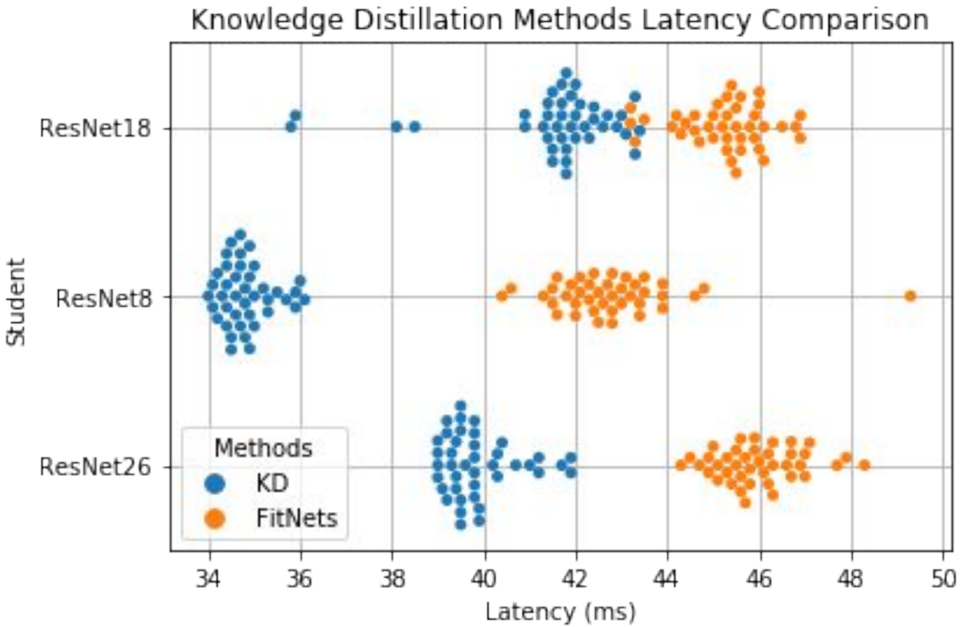


Figure 9. Model Distillation Latency Results

Conclusion

- L1 Regularization provides the fastest inference among model compression methods
- Unstructured sparsity in neural network without accelerators is proved to behave slower than the original model itself (reverified the work of Liu, et al. 2018)
- Distillation method based on feature of the teacher leads to better accuracy-latency trade off than distillation based on the output
- Exact metrics such as latency needed to measure performance

Future Work

- Software: Neural Architecture Search (NAS)

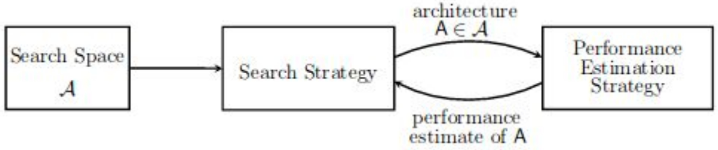


Figure 10. NAS Illustration[9]

- Hardware: Utilization of **accelerators**
  a. Specialty in addressing weight sparsity: Eyeriss, cerebras
  b. Reduced precision: Nvidia Pascal, Intel NNP-L

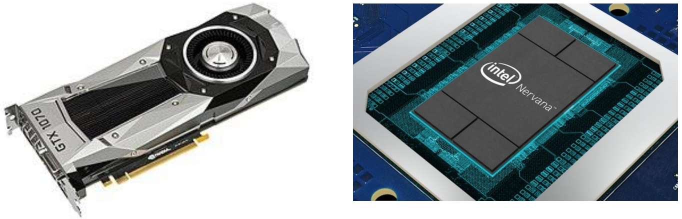


Figure 11. Nvidia Pascal, Intel NNP-L[10]

[9]Thomas Elsken, Jan Hendrik Metzen, and Frank Hutter. Neural architecture search: A survey. 2018.
[10]Thttps://en.wikipedia.org/wiki/Pascal_(microarchitecture);, https://www.intel.ai/nervana-nnp/.