Deep Learning for Computer Vision

Pruning and Model Compression

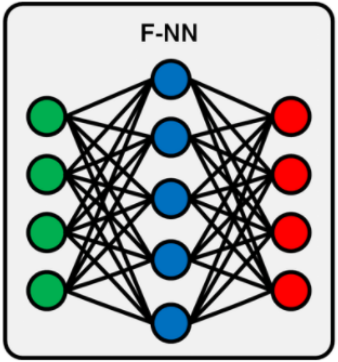
Vineeth N Balasubramanian

Department of Computer Science and Engineering
Indian Institute of Technology, Hyderabad

भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

Motivation

- Deep Neural Networks (DNNs) generally optimized for performance in terms of predictive accuracy
- As a result, DNNs are huge and have parameters in the order of millions
- The popular AlexNet has around 61M parameters! A trained AlexNet takes around 200MB of space

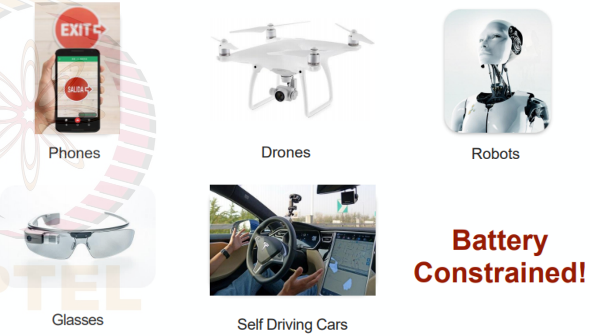


Credit: Xu et al, 2019

Motivation

- While it's acceptable for DNNs to utilize high-end GPUs for training, requiring such powerful processors for inference, is highly limiting

- Applications to various new and battery constrained technologies necessitate low-compute environments:
  - Mobile Phones
  - Unmanned Aerial Vehicles (UAVs)
  - IoT devices



Credit: Song Han, 2016

Motivation

- On mobile devices, crucial to reduce memory consumption for apps, as well as reduce energy consumption

Operation	Energy [pJ]	Relative Cost
32 bit int ADD	0.1	1
32 bit float ADD	0.9	9
32 bit Register File	1	10
32 bit int MULT	3.1	31
32 bit float MULT	3.7	37
32 bit SRAM Cache	5	50
32 bit DRAM Memory	**640**	**6400**

- DRAM accesses cost more energy, which drains battery
- If deep models were compact enough to fit on SRAM, that would reduce energy consumption drastically

Credit: Song Han, 2016

Categorization of Methods for Model Compression

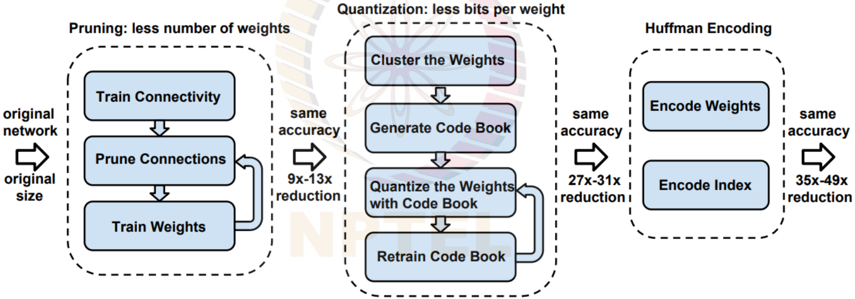
Category Name	Description
Parameter pruning and quantization	Reducing redundant parameters which are not sensitive to the performance
Low-rank factorization	Using matrix/tensor decomposition to estimate the informative parameters
Transferred/compact convolutional filters	Designing special structural convolutional filters to save parameters
Knowledge distillation	Training a compact neural network with distilled knowledge of a large model

We'll see a few sample methods: Pruning-based, Knowledge Distllation-based, and the "Lottery Ticket Hypothesis"

Credit: Cheng et al, A Survey of Model Compression and Acceleration for Deep Neural Networks, 2017

Deep Compression[1]

- One of the most popular, game-changing methods in this space
- A three-stage pipeline to reduce the storage requirement of neural nets



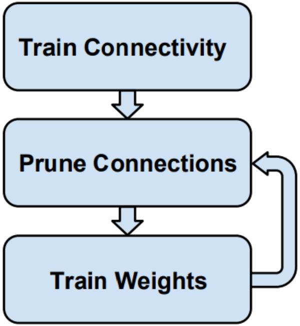
- Showed a 35× decrease in size of AlexNet from 240MB to 6.9MB!

[1]Han et al, Deep Compression: Compressing Deep Neural Network with Pruning, Trained Quantization and Huffman Coding, ICLR 2016

Deep Compression: Pruning

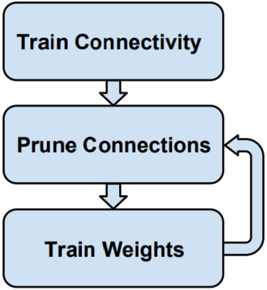
A three-step procedure:

1. **Train Connectivity:** Model weights are learned using standard neural network training
2. **Prune Connections:** Weights (connections) below a certain threshold are removed from network
3. **Train Weights:** Remaining sparse network is retrained



Deep Compression: Pruning[2]

Network	Top-1 Error	Top-5 Error	Parameters	Compression Rate
LeNet-300-100 Ref	1.64%	-	267K	
LeNet-300-100 Pruned	1.59%	-	**22K**	12×
LeNet-5 Ref	0.80%	-	431K	
LeNet-5 Pruned	0.77%	-	**36K**	12×
AlexNet Ref	42.78%	19.73%	61M	
AlexNet Pruned	42.77%	19.67%	**6.7M**	9×
VGG-16 Ref	31.50%	11.32%	138M	
VGG-16 Pruned	31.34%	10.88%	**10.3M**	13×

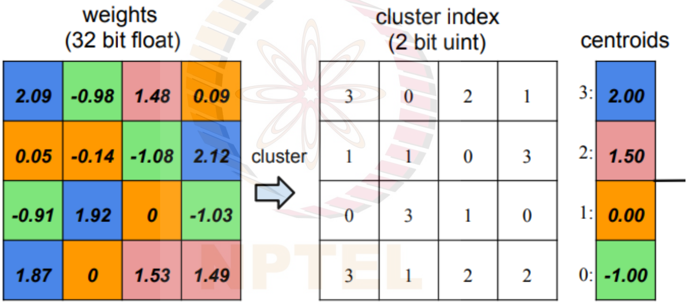


As seen in table, pruning shown to compress networks by 9-13×

[2]Han et al, Deep Compression: Compressing Deep Neural Network with Pruning, Trained Quantization and Huffman Coding, ICLR 2016

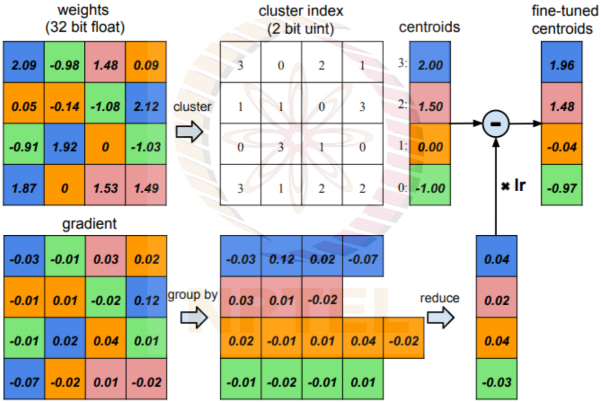
Deep Compression: Weight Sharing

- In each layer, weights are partitioned into k clusters using simple K-means clustering



- Weights (and gradients) with same color (cluster) are grouped together; all weights of same color are represented by corresponding centroid

Deep Model Compression: Weight Sharing



Gradients of same color are added, sum is used to update corresponding centroid

Deep Model Compression: Quantization and Huffman Coding[3]

- Instead of using 32-bit floating point values for weights, experiments showed no loss of accuracy when weights were quantized upto 8 bits

[3] Han et al, Deep Compression: Compressing Deep Neural Network with Pruning, Trained Quantization and Huffman Coding, ICLR 2016

Deep Model Compression: Quantization and Huffman Coding[3]

- Instead of using 32-bit floating point values for weights, experiments showed no loss of accuracy when weights were quantized upto 8 bits
- Pruned and quantized network encoded using **Huffman coding**; frequently observed values stored with less number of bits, and rare values stored with more bits

[3] Han et al, Deep Compression: Compressing Deep Neural Network with Pruning, Trained Quantization and Huffman Coding, ICLR 2016

Deep Model Compression: Quantization and Huffman Coding[3]

- Instead of using 32-bit floating point values for weights, experiments showed no loss of accuracy when weights were quantized upto 8 bits
- Pruned and quantized network encoded using **Huffman coding**; frequently observed values stored with less number of bits, and rare values stored with more bits
- Deep compression method compressed various networks from $35\times$ to $49\times$ less than original size with minimal loss of accuracy!

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**×
LeNet-5 Ref	0.80%	-	1720 KB	
LeNet-5 Compressed	0.74%	-	**44 KB**	**39**×
AlexNet Ref	42.78%	19.73%	240 MB	
AlexNet Compressed	42.78%	19.70%	**6.9 MB**	**35**×
VGG-16 Ref	31.50%	11.32%	552 MB	
VGG-16 Compressed	31.17%	10.91%	**11.3 MB**	**49**×

[3]Han et al, Deep Compression: Compressing Deep Neural Network with Pruning, Trained Quantization and Huffman Coding, ICLR 2016

Knowledge Distillation

Key Idea

Transfer "knowledge" from a cumbersome large model (**teacher**) to a small model (**student**), whose size is more optimized for deployment

Knowledge Distillation

Key Idea

Transfer "knowledge" from a cumbersome large model (**teacher**) to a small model (**student**), whose size is more optimized for deployment; but how? What is "knowledge" in a DNN model?

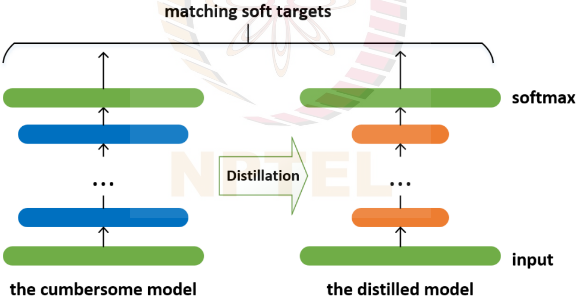
Knowledge Distillation[4]

- In the case of image classification, **knowledge** can be seen as the **mapping** between input (images) and output (softmax probabilities)

[4] Hinton et al, Distilling the Knowledge in a Neural Network, NeurIPS-W 2015

Knowledge Distillation[4]

- In the case of image classification, **knowledge** can be seen as the **mapping** between input (images) and output (softmax probabilities)
- Instead of training a student network with hard labels, they can be trained to **mimic the softmax** outputs of the teacher model, for each image



Credit: Yangyang, 2014

[4]Hinton et al, Distilling the Knowledge in a Neural Network, NeurIPS-W 2015

Knowledge Distillation: A Simple Example on MNIST

Models

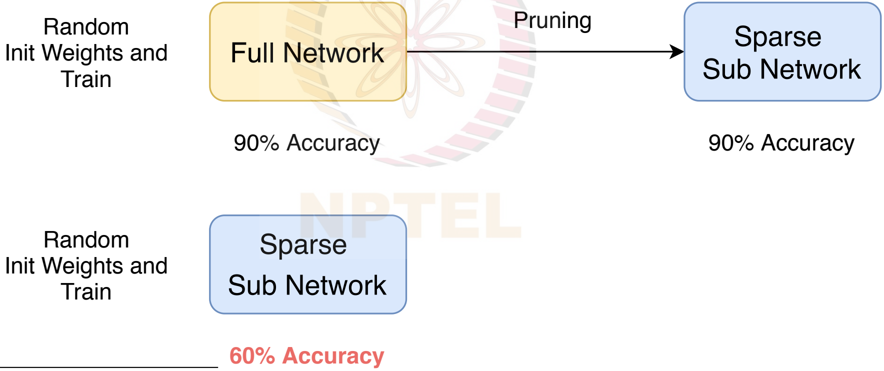
- Cumbersome model: 2 layers, 1200 ReLU nodes, dropout regularization
- Small model: 2 layers, 800 ReLU nodes, no regularization

Number of errors on MNIST

- Cumbersome Model: 67
- Small model with standard training: 146
- Small model with **distillation**: 74

Lottery Ticket Hypothesis: Motivation[5]

- **Observation:** A very sparse subnetwork obtained after pruning a fully trained network produces accuracy close to the full model



[5]Frankle and Carbin, The Lottery Ticket Hypothesis: Finding Sparse, Trainable NeuralNetworks, ICLR 2019

Lottery Ticket Hypothesis

The Hypothesis

A randomly-initialized, dense neural network contains a subnetwork that is initialized such that —- when trained in isolation —- it can match the test accuracy of the original network after training for at most the same number of iterations

Lottery Ticket Hypothesis

The Hypothesis

A randomly-initialized, dense neural network contains a subnetwork that is initialized such that —- when trained in isolation —- it can match the test accuracy of the original network after training for at most the same number of iterations

How to find the network?

Lottery Ticket Hypothesis

The Hypothesis

A randomly-initialized, dense neural network contains a subnetwork that is initialized such that —- when trained in isolation —- it can match the test accuracy of the original network after training for at most the same number of iterations

How to find the network? **One shot pruning:**

- Train a neural network with random initialization
- Prune p% of smallest weights
- Reset remaining weights to their previous initialization, to create the winning ticket

Lottery Ticket Hypothesis

The Hypothesis

A randomly-initialized, dense neural network contains a subnetwork that is initialized such that —- when trained in isolation —- it can match the test accuracy of the original network after training for at most the same number of iterations

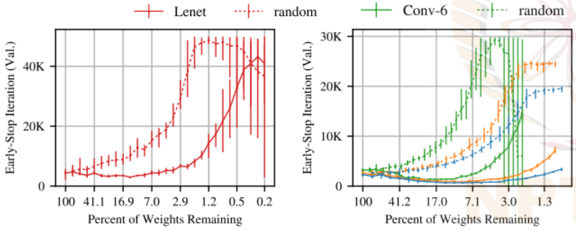
How to find the network? **One shot pruning:**

- Train a neural network with random initialization
- Prune p% of smallest weights
- Reset remaining weights to their previous initialization, to create the winning ticket

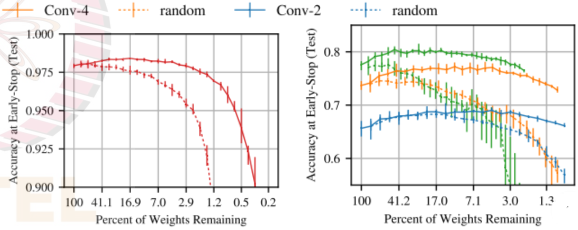
Alternatively, repeatedly pruning the network over n rounds (iterative pruning) has shown much better results, although more computationally expensive

Lottery Ticket Hypothesis: Results

Percent of weights remaining vs early stop iterations (MNIST and CIFAR-10 datasets)



Percent of weights remaining vs accuracy (MNIST and CIFAR-10 datasets)



Dotted lines show randomly sampled sparse networks while solid lines represent winning tickets (which attain more accuracy than randomly sampled sparse nets)

Lottery Ticket Hypothesis: Limitations and Further Work

Limitations

- While iterative pruning produces better results, it requires training the network 15 times per round of pruning (5 trials, training each winning ticket 3 times and taking the average)
- Harder to study large datasets like ImageNet

Lottery Ticket Hypothesis: Limitations and Further Work

Limitations

- While iterative pruning produces better results, it requires training the network 15 times per round of pruning (5 trials, training each winning ticket 3 times and taking the average)
- Harder to study large datasets like ImageNet

Further Studies

- Can we find winning tickets early on in training? (You et al, 2020)
- Do winning tickets generalize across datasets and optimizers? (Morcos et al, 2019)
- Can this hypothesis hold in other domains like text processing/NLP? (Yu et al, 2019)

Extensions and Other Methods

Pruning and Quantization

- **XNOR-Net**: Using binary weights and approximating convolutions with XNOR operations
- **Thi-Net**: Compressing CNNs with filter pruning

Distillation

- **Noisy Teachers**: Perturbing teacher logits to regularize the student
- **Relational Knowledge Distillation**: Adapting metric learning for distillation

Architectures

- **MobileNets**: Depth-wise separable convolutions
- **ShuffleNet**: Group Convolutions and Channel Shuffle
- **SqueezeNet**: Replacing 3x3 with 1x1 convolutions
- **SqueezeDet**: Fully convolutional network for fast object detection
- **SEP-NET**: Transforming k $\times$ k convolutions into binary patterns for reducing model size

Recall: Categorization of Methods for Model Compression

Category Name	Description
Parameter pruning and quantization	Reducing redundant parameters which are not sensitive to the performance
Low-rank factorization	Using matrix/tensor decomposition to estimate the informative parameters
Transferred/compact convolutional filters	Designing special structural convolutional filters to save parameters
Knowledge distillation	Training a compact neural network with distilled knowledge of a large model

Many more methods!

Credit: Cheng et al, A Survey of Model Compression and Acceleration for Deep Neural Networks, 2017

Homework

Readings

- Robert T. Lange, Lottery Ticket Hypothesis: A Survey, 2020
- Cheng *et al.*, A Survey of Model Compression and Acceleration for Deep Neural Networks, 2017.

References

- Song Han et al. "Learning both Weights and Connections for Efficient Neural Networks". In: *CoRR* abs/1506.02626 (2015). arXiv: 1506.02626.
- Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. "Distilling the Knowledge in a Neural Network". In: *NIPS Deep Learning and Representation Learning Workshop*. 2015.
- Song Han, Huizi Mao, and W. Dally. "Deep Compression: Compressing Deep Neural Network with Pruning, Trained Quantization and Huffman Coding". In: *CoRR* abs/1510.00149 (2016).
- Jonathan Frankle and Michael Carbin. "The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks". In: *International Conference on Learning Representations*. 2019.
- Ari Morcos et al. "One ticket to win them all: generalizing lottery ticket initializations across datasets and optimizers". In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach et al. Vol. 32. Curran Associates, Inc., 2019, pp. 4932–4942.
- T. Xu and I. Darwazeh. "Design and Prototyping of Neural Network Compression for Non-Orthogonal IoT Signals". In: *2019 IEEE Wireless Communications and Networking Conference (WCNC)*. 2019, pp. 1–6.
- Haoran You et al. "Drawing Early-Bird Tickets: Toward More Efficient Training of Deep Networks". In: *International Conference on Learning Representations*. 2020.