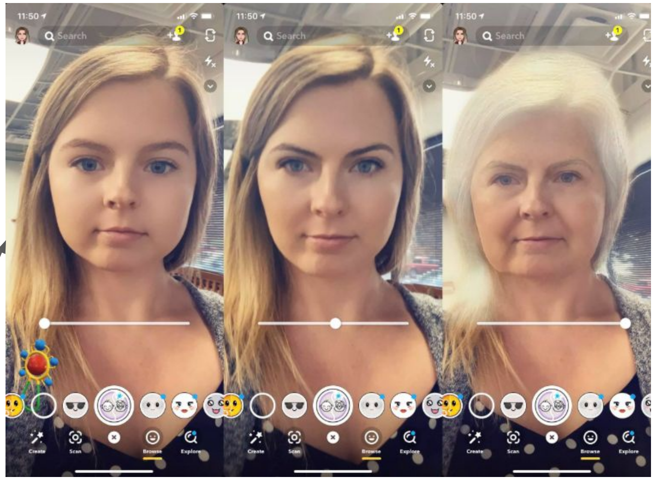
Overview of Model Compression and Acceleration

July 30, 2020

Seonyoung Kim

Cloud computing



Face detection

Self Driving Cars

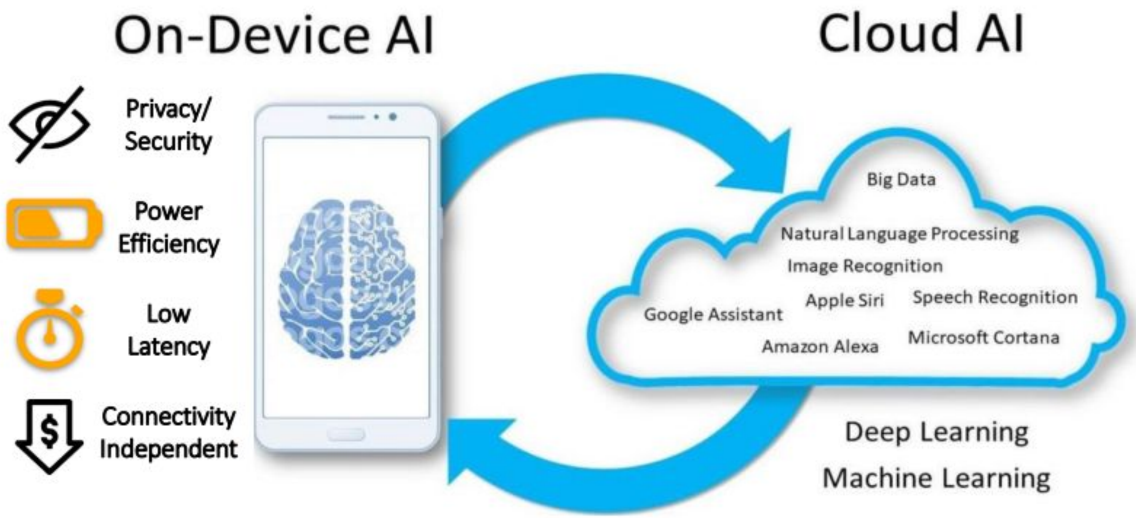
Latency

Privacy

Cost

On-device AI


온 디바이스 AI
명령
분석/수행


On-Device AI
Cloud AI
Privacy/
Security
Power
Efficiency
Low
Latency
Connectivity
Independent
Big Data
Natural Language Processing
Image Recognition
Google Assistant
Apple Siri
Speech Recognition
Amazon Alexa
Microsoft Cortana
Deep Learning
Machine Learning


온 디바이스 AI의 장점
개인정보 보호
빠른 응답속도
저전력/저비용

Model compression techniques

1. Neural Network Pruning
2. Quantization
3. Knowledge-Distillation(KD)
4. Low-Rank Approximation
5. Compact Networks Design

Techniques for compressing existing models (1–4)

Techniques for designing optimal models (5)

Experiment setting

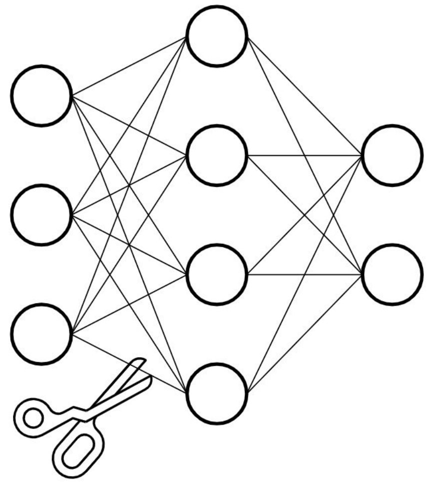
Model

- CNN models such as AlexNet, LeNet, VGG and ResNet
- LSTM, BERT, .. etc

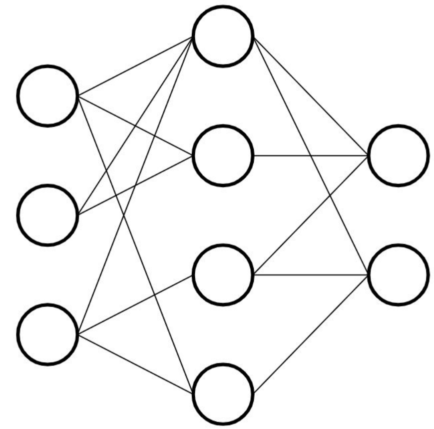
Dataset

- CIFAR10, CIFAR100, MNIST, tiny-ImageNet, ImageNet, Coco dataset
- Audioset, SST-2, MRPC, .. etc

Neural Network Pruning



Before pruning



After pruning

pruning 미국·영국 [prú:niŋ] 영국식
(나무 등의) 가지치기, 전지, 전정(剪定)

Neural Network Pruning

Learning both weights and connections for efficient neural network. (Han Song, NIPS 2015)

- Weight pruning

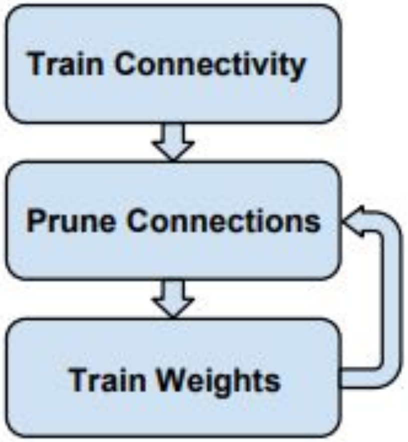


Figure 2: Three-Step Training Pipeline.

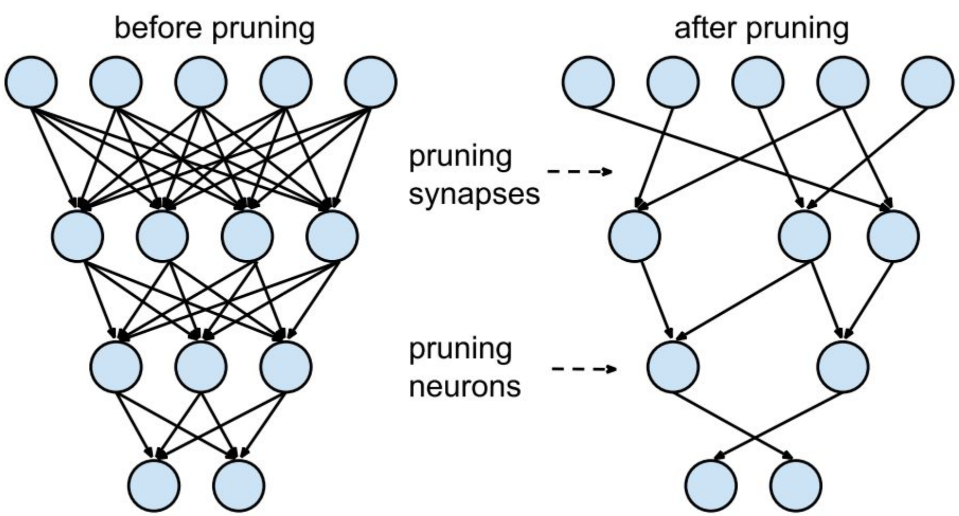
1. Train the network to learn which connections are important(i.e., pretrained model)
2. Prune unimportant connections(i.e., remove weight below threshold)
3. Retrain the network to fine tune the remaining weights
4. Iterate 2-3 steps

Neural Network Pruning

Learning both weights and connections for efficient neural network. (Han Song, NIPS 2015)

- Weight pruning

if | Weights | < Threshold → Pruning



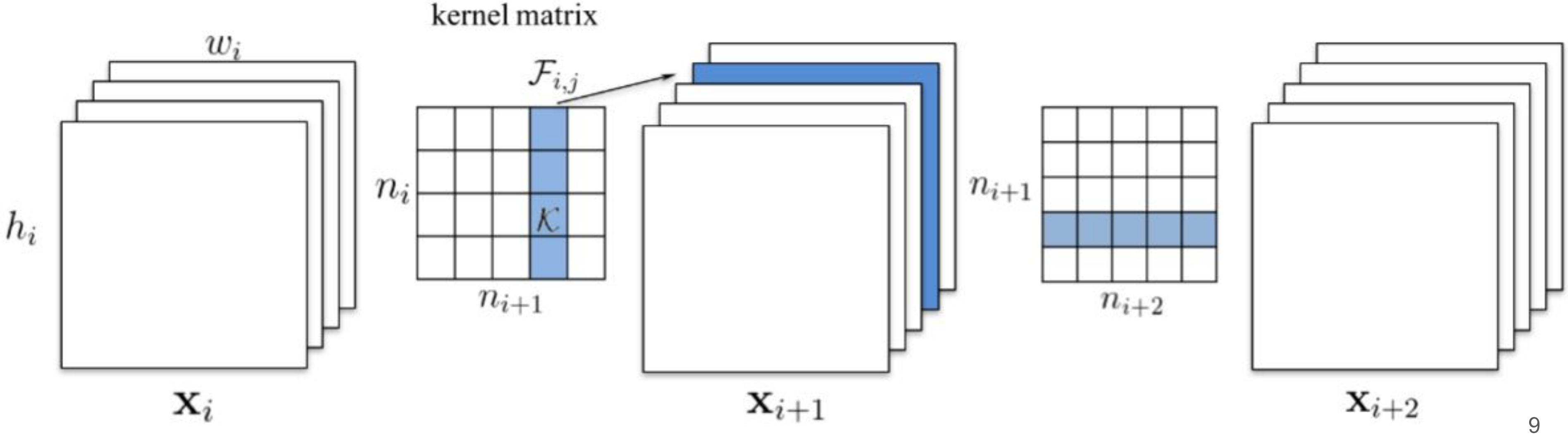
Weight Matrix

		6	4
	3		
8			
	5		9

Neural Network Pruning

Pruning Filters for Efficient ConvNets(Hao Li, ICLR 2017 Conference)

- Convolutional layer pruning
  1. Train Network(i.e., pretrained model)
  2. Ranks all filters(pruning criteria : L1 norm of each filter weight)
  3. Prune filters with low rank globally for all layers.
  4. Iterate 2-3 steps

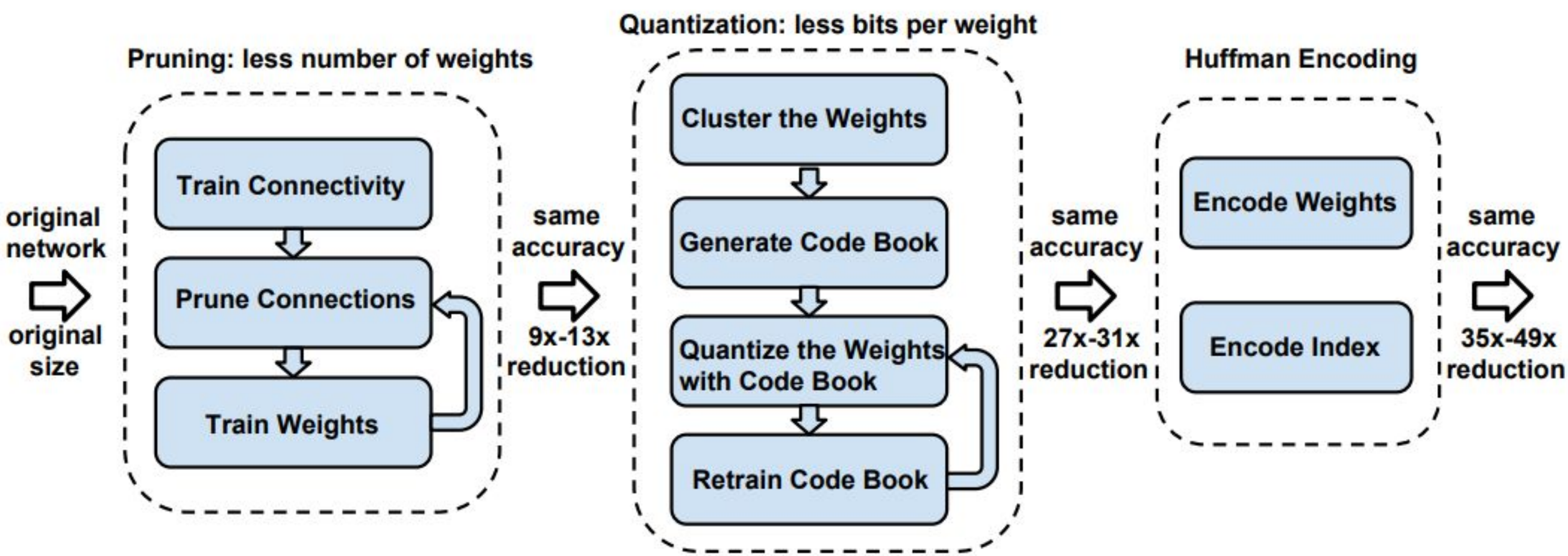


Quantization

- the process of converting a continuous range of values into a finite range of discrete values.
- Network Quantization compresses the original network by reducing the number of bits used to represent the weights
- e.g., 32 bits of Weight → 16 bits, 8 bits, 4 bits, or even 1 bit

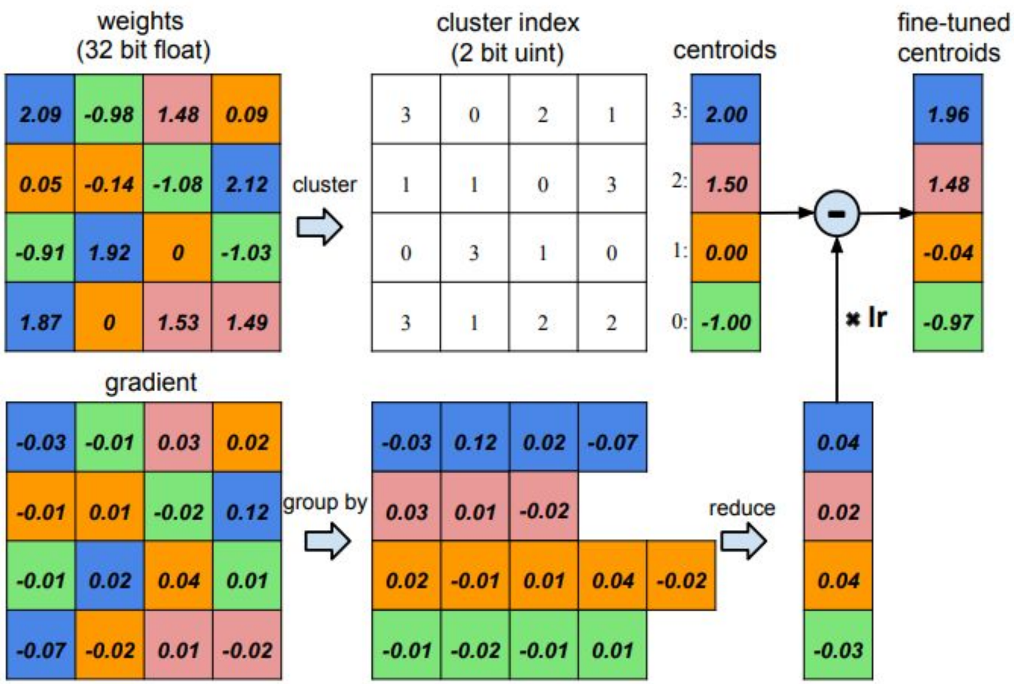
Quantization

Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding(Han Song, ICLR 2016 Best paper)



Quantization

Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding(Han Song, ICLR 2016 Best paper)



1. Using k-Means, cluster weights
2. Weights are represented by their centroid
3. Fine-tune with gradient

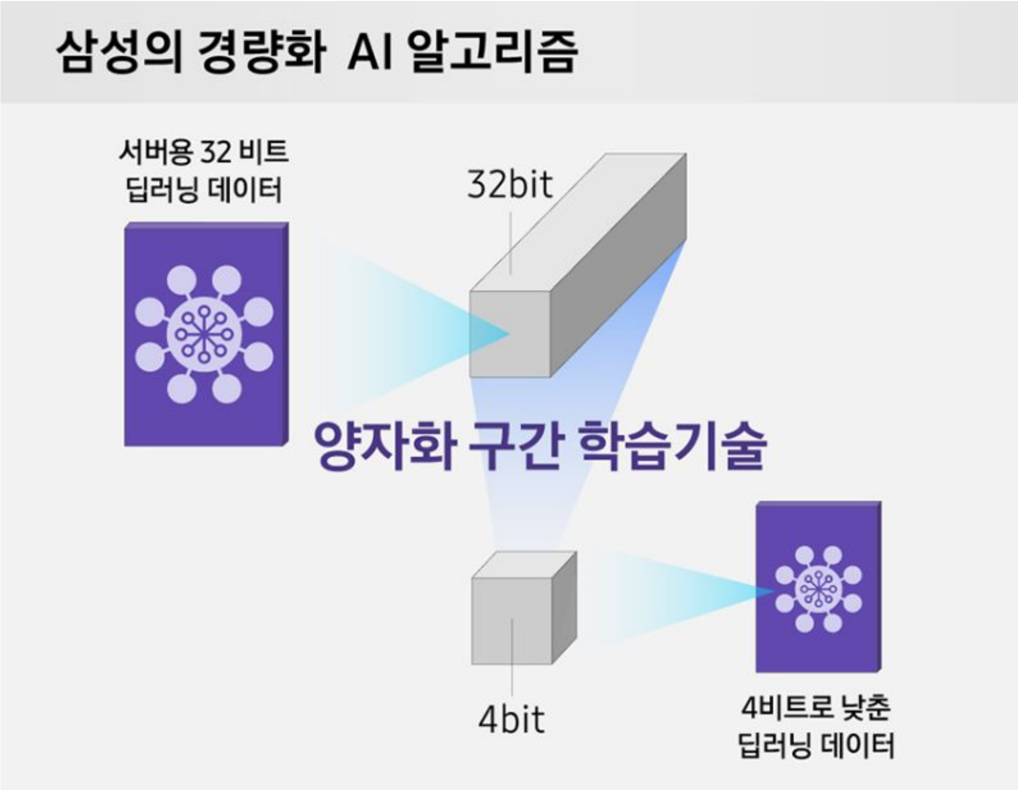
Before Quan.: 16x32 = 512 bits

After Quan.: 4x32 + 2x16 = 160 bits

Compression Rate = 3.2

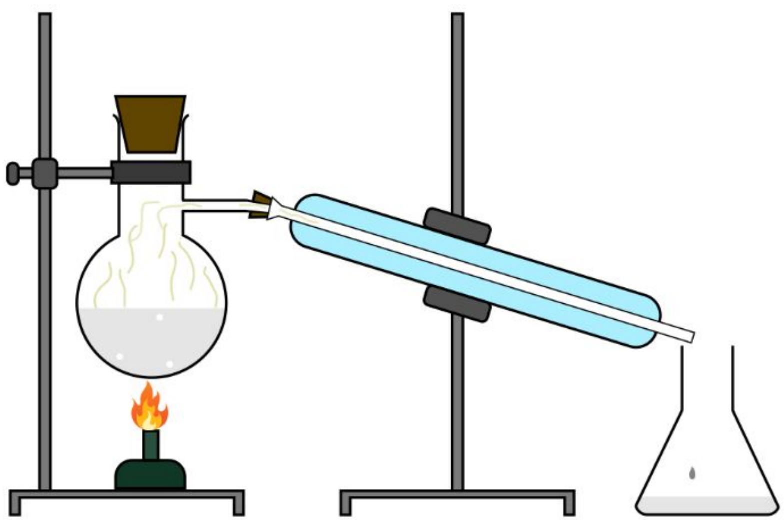
Quantization

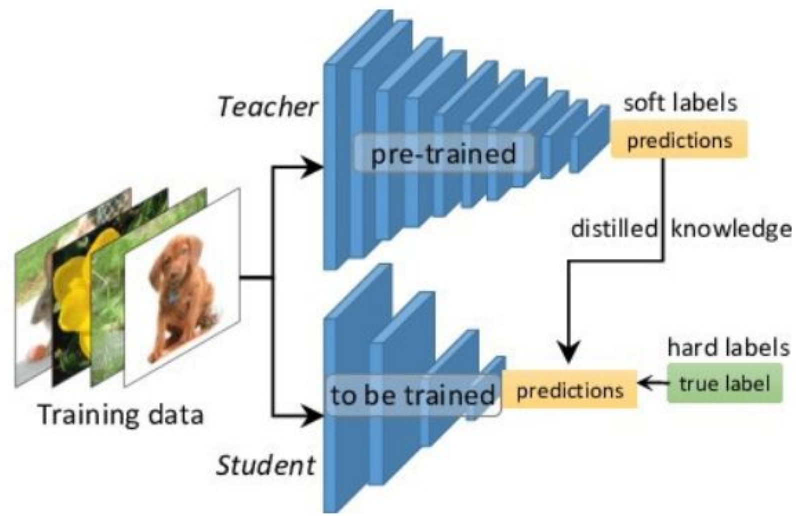
Learning to Quantize Deep Networks by Optimizing Quantization Intervals with Task Loss(Samsung, Sung Ju Hwang, CVPR 2019)



Knowledge-Distillation(KD)

Distilling the Knowledge in a Neural Network(Hinton Geoffrey, NIPS 2014 Workshop)





distillation 미국식 [dìstəléiʃən] 다른 뜻(1건)
[U] 증류(법), [UC] 증류물, 정수

Large Network → Teacher Network

Small Network → Student Network

Knowledge-Distillation(KD)

Distilling the Knowledge in a Neural Network(Hinton Geoffrey, NIPS 2014 Workshop)

- Softmax Output = Knowledge = Soft Label

dog

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

Comparison with the 'hard label' and the 'soft label'

$$p_i = \frac{\exp\left(\frac{z_i}{T}\right)}{\sum_j \exp\left(\frac{z_j}{T}\right)}$$

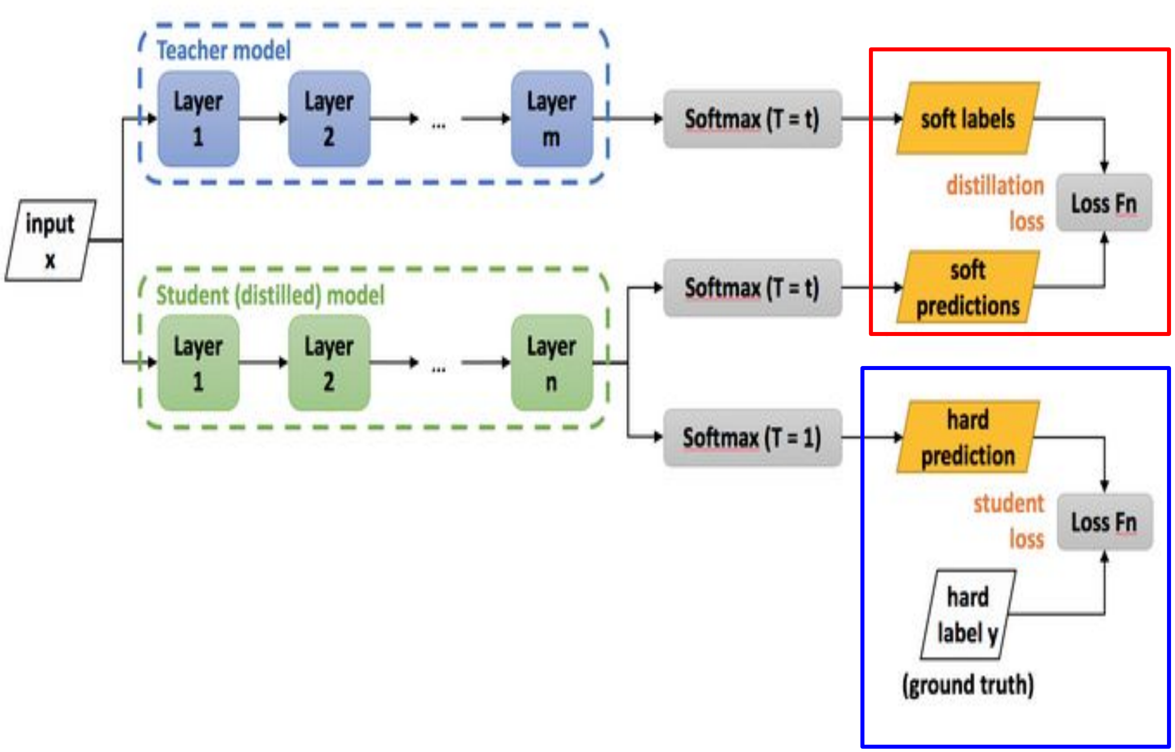
Softmax with temperature T

e.g., hard label → this image is dog

soft label → This image is the closest to the dog, but it seems to resemble a cat a little. And very little, but it also has the characteristics of a cow and a car.

Knowledge-Distillation(KD)

Distilling the Knowledge in a Neural Network(Hinton Geoffrey, NIPS 2014 Workshop)



$$Total\ Loss = \boxed{(1-\alpha)L_{CE}(\sigma(Z_s),\ \hat{y})} + \boxed{2\alpha T^2 L_{CE}(\sigma(\frac{Z_s}{T}),\sigma(\frac{Z_s}{T}))}$$

$L_{CE}()$: Cross entropy loss

$\sigma()$: Softmax

Z_s: Output logits of Student network

Z_t: Output logits of Teacher network

$\hat{y}$: Ground truth(one-hot)

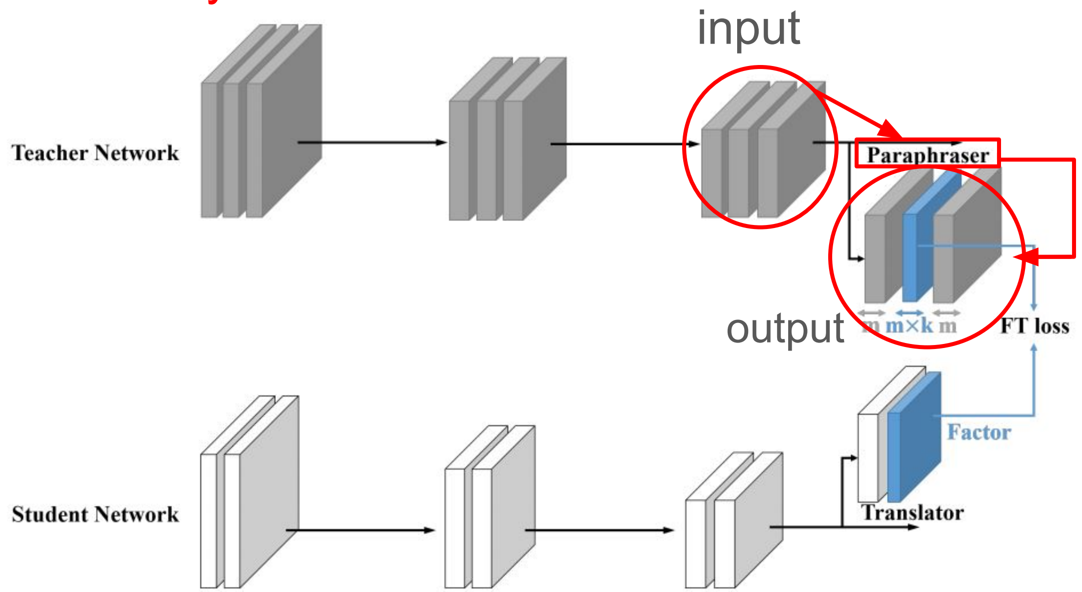
α: Balancing parameter

T: Temperature hyperparameter

Knowledge-Distillation(KD)

Paraphrasing Complex Network: Network Compression via Factor Transfer(Jangho Kim, NIPS 2018)

- Existing methods transfer knowledge of teacher network to the student network directly

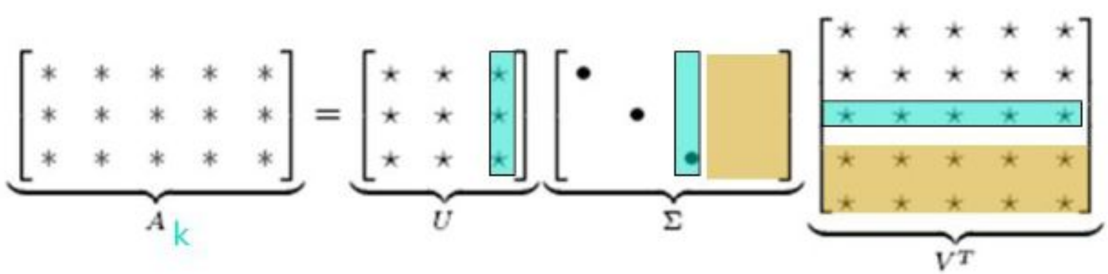


- Paraphraser(Similar to autoencoder) extracts the information from feature maps of the last group and the output of paraphraser's middle layer, as 'teacher factors'.

Low-Rank Approximation

- a weight matrix A with m x n dimension is replaced by smaller dimension matrices
- Singular value decomposition(SVD) is a common and popular factorization scheme for reducing the number of parameters

$$A_k = U\,\mathrm{diag}(\sigma_1,\ldots,\sigma_k,\underbrace{0,\ldots,0}_{\substack{\text{set smallest r-k}\\ \text{singular values to zero}}})V^T$$



Low-Rank Approximation

Compression of Acoustic Event Detection Models with Low-rank Matrix Factorization and Quantization Training(Shi Bowen, NIPS 2018 CDNNRIA Workshop)

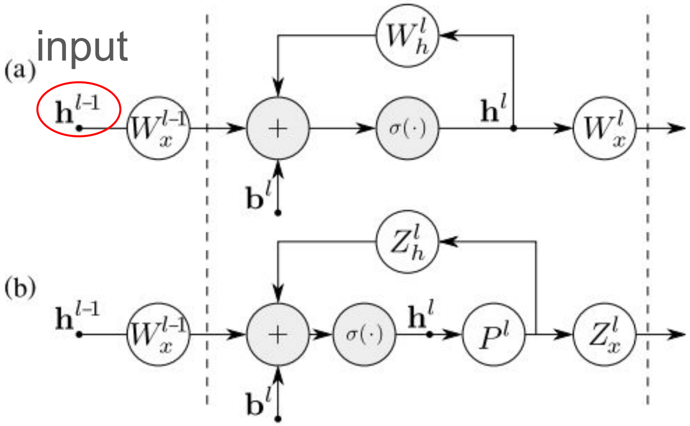


Fig. 1. The initial model (Figure (a)) is compressed by jointly factorizing recurrent (W_h^l) and inter-layer (W_x^l) matrices, using a shared recurrent projection matrix (P^l) [3] (Figure (b)).

Applied to RNN model

$$W_h^l = U_h^l \Sigma_h^l {V_h^l}^T \approx \left(\widetilde{U_h^l}\widetilde{\Sigma_h^l}\right)\widetilde{V_h^l}^T = Z_h^l P^l$$

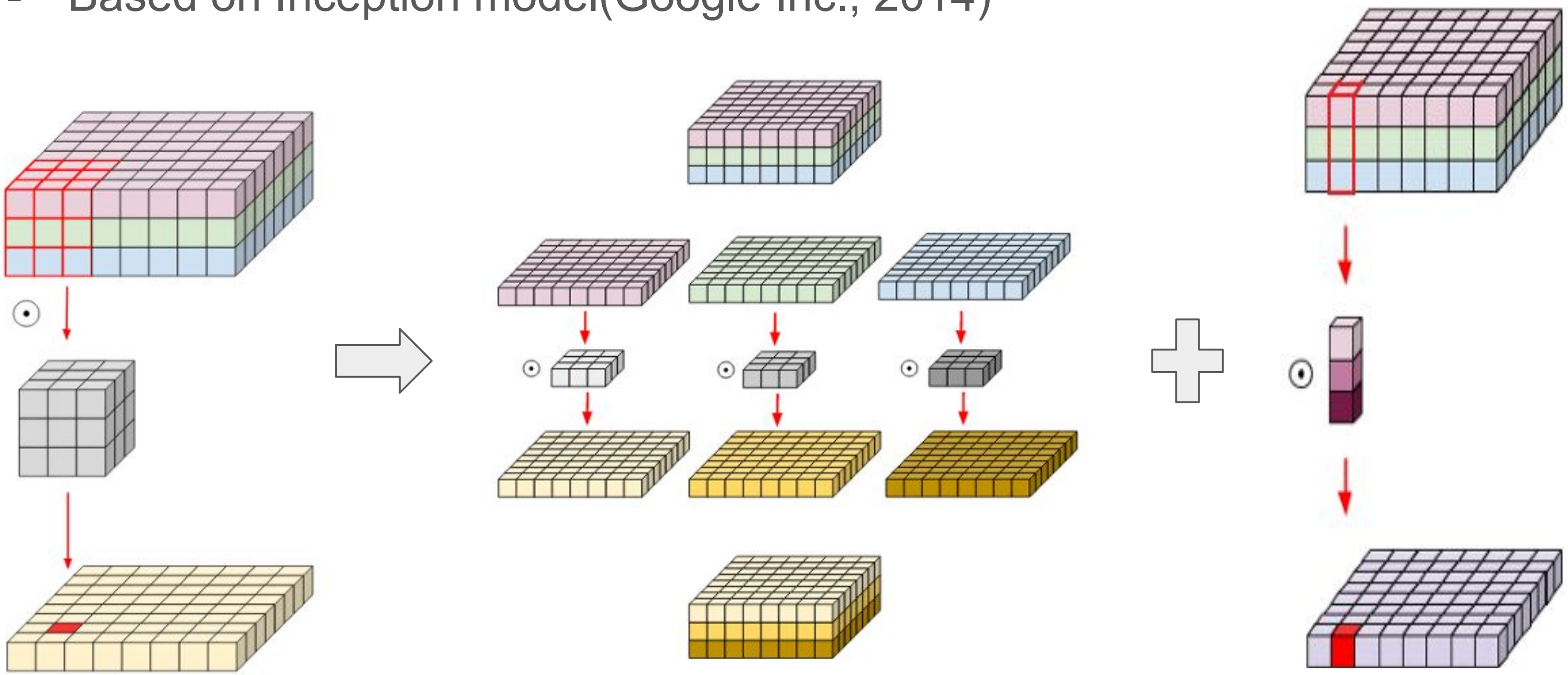
$$\begin{aligned} \mathbf{h}_t^l &= \sigma(W_x^{l-1}\mathbf{h}_t^{l-1} + Z_h^l P^l \mathbf{h}_{t-1}^l + \mathbf{b}^l) \\ \mathbf{h}_t^{l+1} &= \sigma(Z_x^l P^l \mathbf{h}_t^l + W_h^{l+1}\mathbf{h}_{t-1}^{l+1} + \mathbf{b}^{l+1}) \end{aligned}$$

$$Z_x^l = \arg\min_Y \|YP^l - W_x^l\|_{\mathcal{F}}^2$$

Efficient network architectures

MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications(Google Inc., 2017)

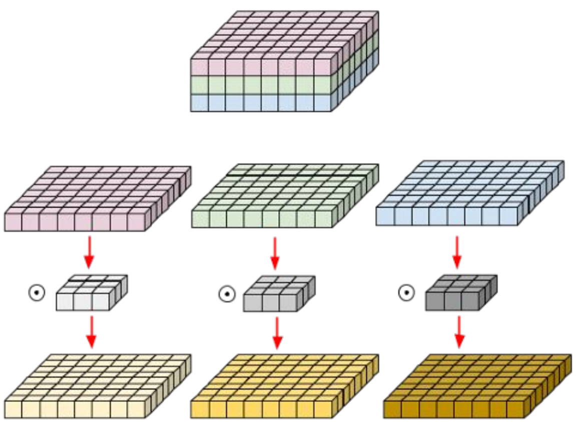
- Based on Inception model(Google Inc., 2014)



Standard Convolution Depthwise Convolution Pointwise Convolution

Efficient network architectures

MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications(Google Inc., 2017)

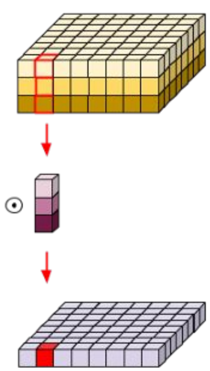


Depth-wise separable convolution = Depthwise + Pointwise

Depthwise Convolution

Extract spatial features of each channel

Number of input channels = Number of output channels



Pointwise Convolution

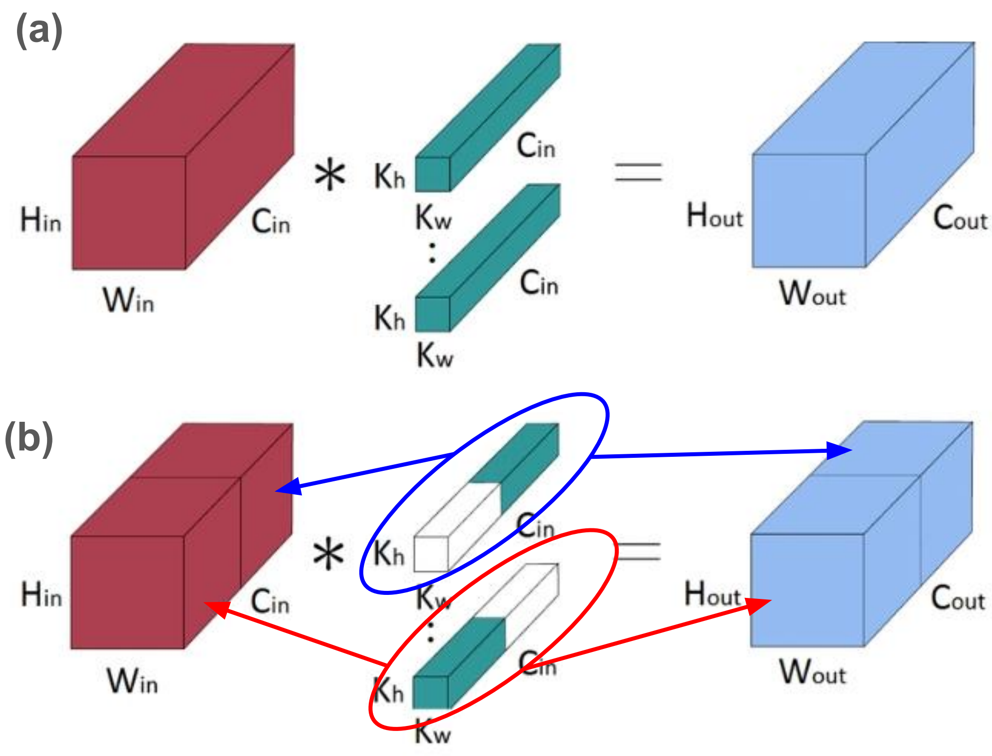
No spatial features

Dimensional reduction

=> When the depthwise filter is k x k, it is about k^2 times less in the computational amount

Efficient network architectures

ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices (Face++, CVPR 2018)



Grouped Convolution

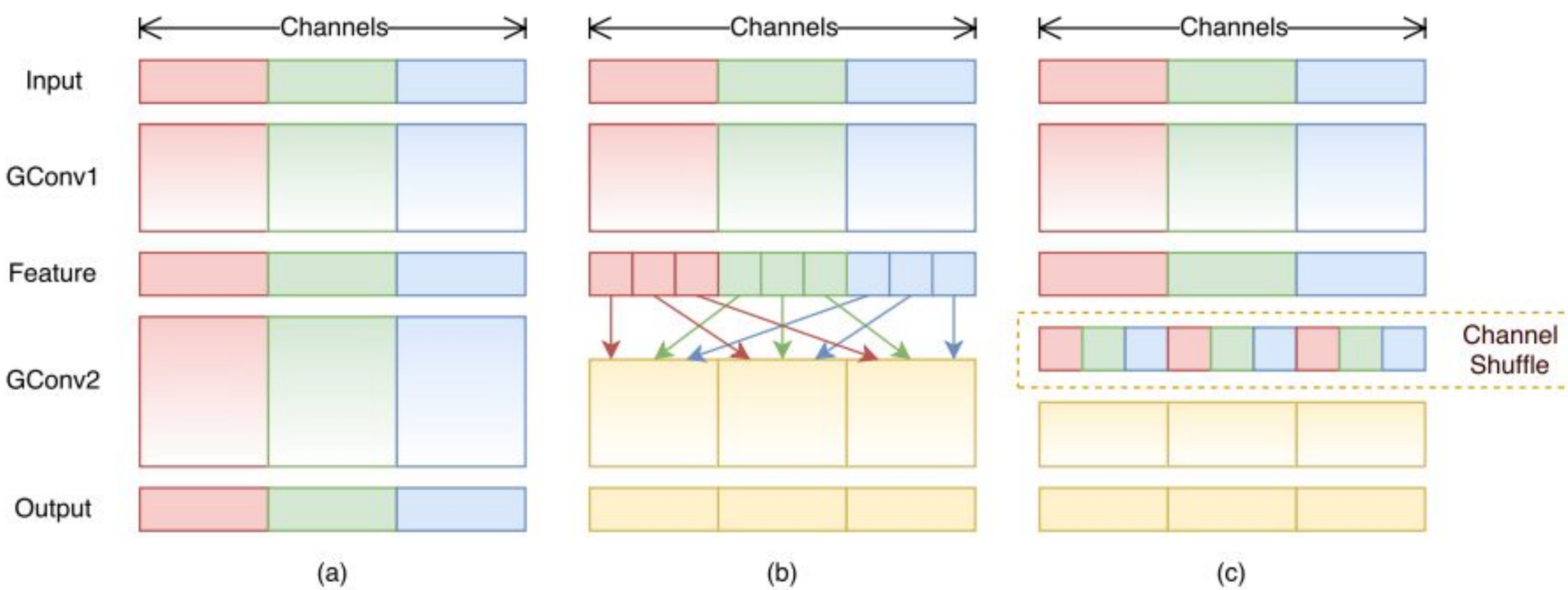
- The filters are separated into different groups
- The model learns highly correlated information for each group.
- The operation parameter becomes sparse

→ However, there is no cross talk between groups(i.e., No information exchange)

(a) Standard Convolution; **(b) Grouped Convolution**

Efficient network architectures

ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices (Face++, CVPR 2018)



Channel Shuffle

- Shuffle the channels in each group so that all groups can exchange information

Model compression techniques

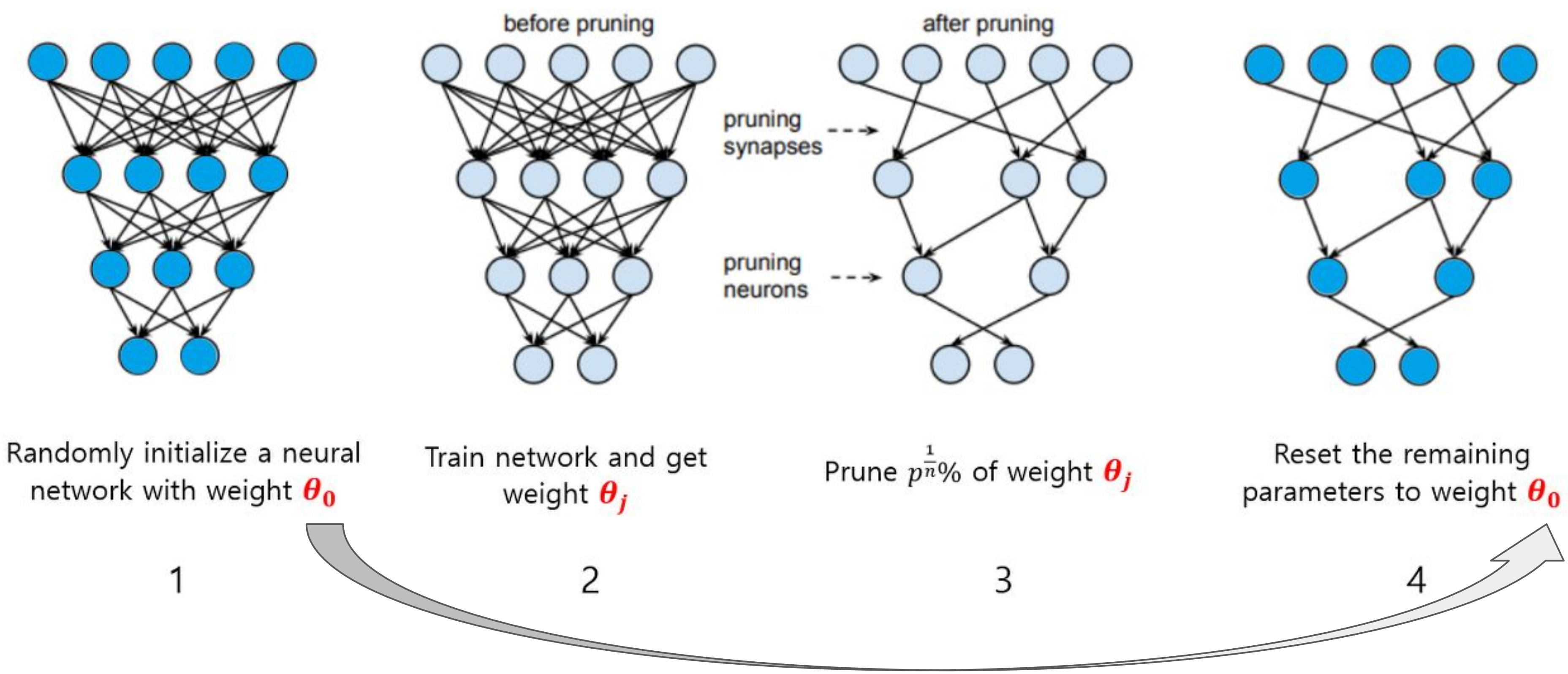
1. Neural Network Pruning
   ➔ Remove connectivity between weight, filter, .. etc.
2. Quantization
   ➔ Reduce the number of bits used to represent the weights, data, .. etc.
3. Knowledge-Distillation(KD)
   ➔ Transfer knowledge of a large model to a small model
4. Low-Rank Approximation
   ➔ Reduce the dimension of the weight matrix
5. Compact Networks Design
   ➔ Construct small model with efficiency

Thank you

Neural Network Pruning

The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks(Frankle J, ICLR 2019 Best paper)

- When retraining, the weight is randomly initialized or used the weight before pruning → the accuracy is greatly reduced.



Reset the parameters to the initial value of weight before pruning