Model Compression

Hyewon Choi

CONTENTS

Model Compression

Model Compression

모델 경량화 : 비슷한 수준의 성능을 유지한 채 더 적은 파라미터 수와 연산량을 가지는 모델을 만드는 것.

1. 경량 알고리즘 연구
   1) 모델 구조 변경
   2) 합성곱 필터 변경
   3) 자동 모델 탐색
2. 알고리즘 경량화
   1) 모델 압축
   2) 지식 증류
   3) 하드웨어 가속화
   4) 모델 압축 자동 탐색

출처 : https://ettrends.etri.re.kr/ettrends/176/0905176005/34-2_40-50.pdf
ETRI, 경량 딥러닝 기술 동향(2019)

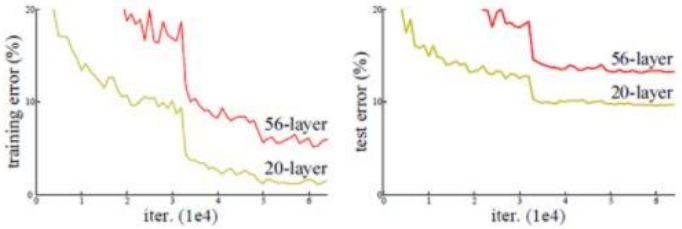
Model Compression

모델 경량화 : 비슷한 수준의 성능을 유지한 채 더 적은 파라미터 수와 연산량을 가지는 모델을 만드는 것.

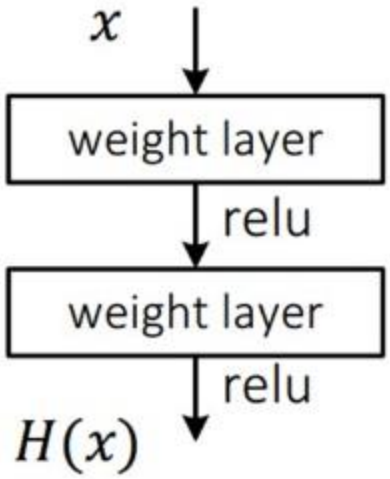
1. **경량 알고리즘 연구**
   1) 모델 구조 변경
   2) 합성곱 필터 변경
   3) 자동 모델 탐색
2. 알고리즘 경량화
   1) 모델 압축
   2) 지식 증류
   3) 하드웨어 가속화
   4) 모델 압축 자동 탐색

Model Compression

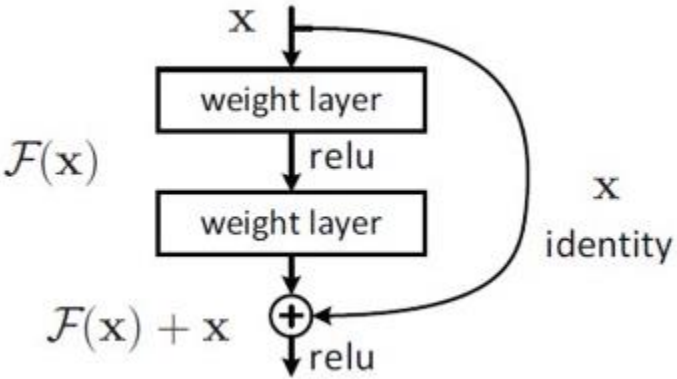
1) 모델 구조 변경
   - ResNet



기존 딥러닝의 문제점 :
층이 깊어질 수록 에러가 높아진다.



평범한 CNN



RexNet

Model Compression

1) 모델 구조 변경
   - DensNet
     : ResNet 과 다르게 더하는 방식이 아닌 concatnate(연결)하는 방식

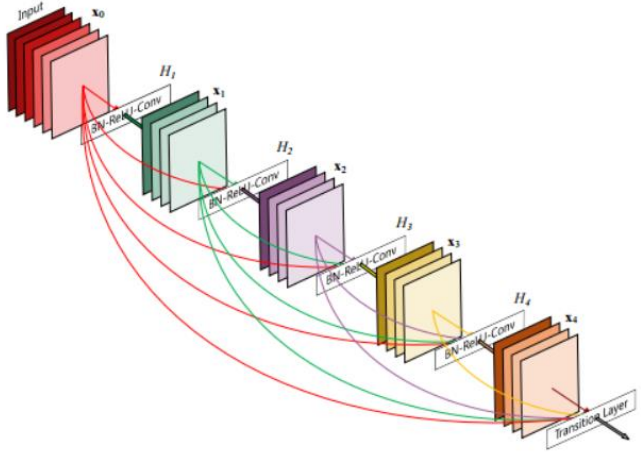
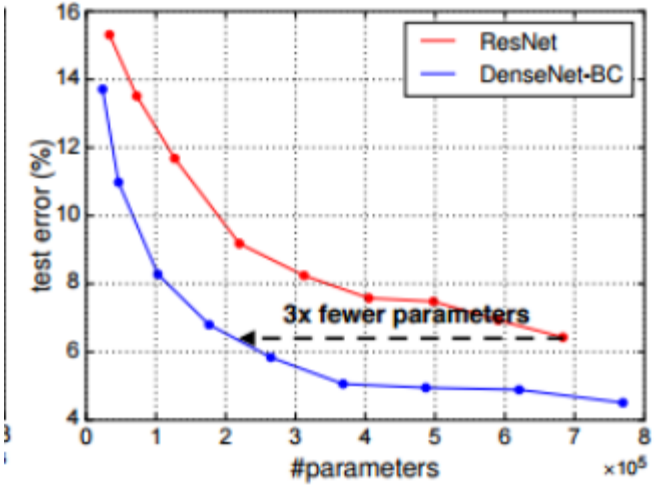


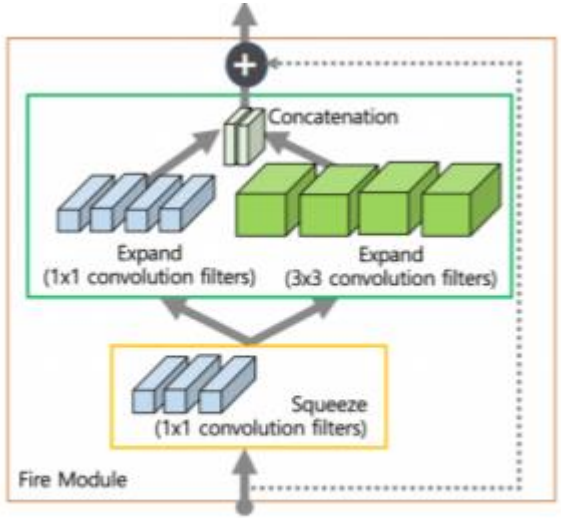
Figure 1: A 5-layer dense block with a growth rate of $k = 4$. Each layer takes all preceding feature-maps as input.



Model Compression

1) 모델 구조 변경
   - SqueezeNet

3x3 -> 1x1 필터로 대체

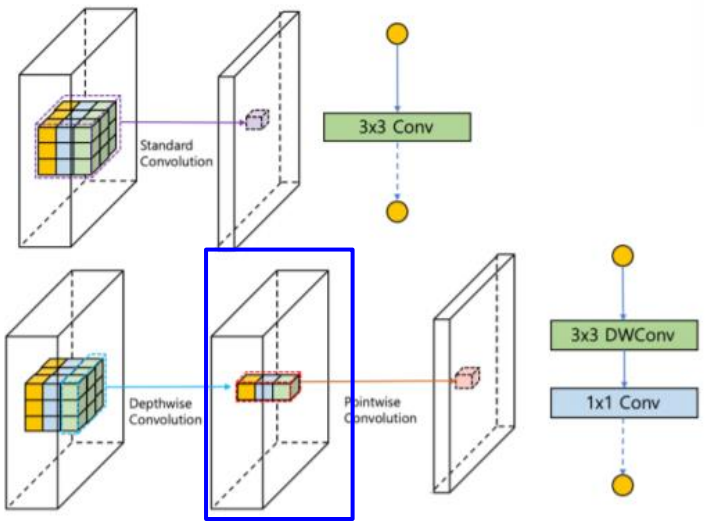


(그림 3) 스퀴즈넷(SqueezeNet)의 파이어 모듈(Fire Mod-ule)

Model Compression

2) 효율적인 합성곱 필터 기술

- MobileNet

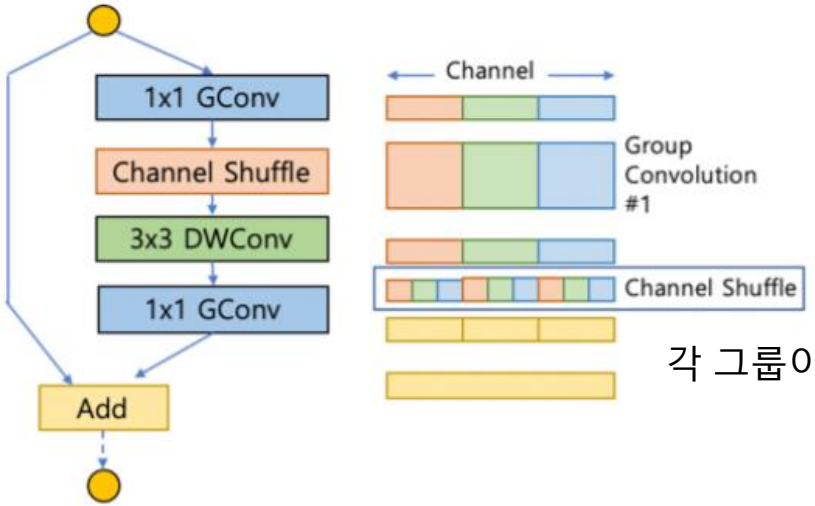


(그림 4) 모바일넷(MobileNet)의 합성곱 분해 구조

Model Compression

2) 효율적인 합성곱 필터 기술

- ShuffleNet



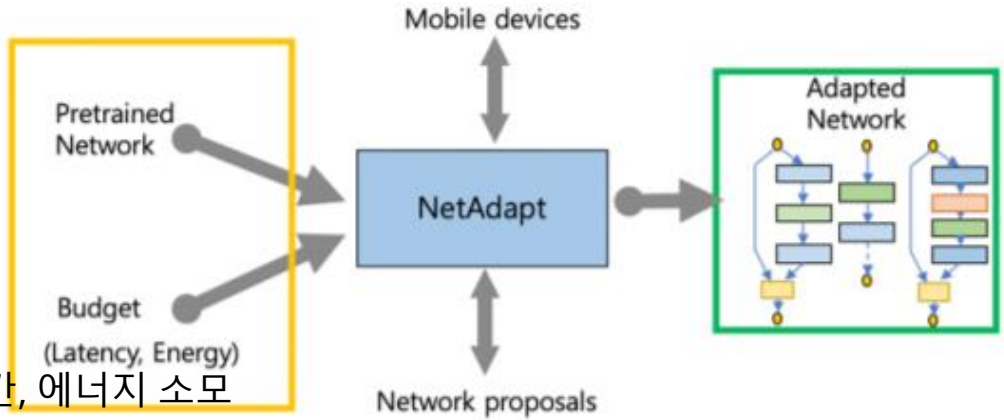
각 그룹이 잘 섞일 수 있도록 개선

(그림 5) 셔플넷(ShuffleNet)의 채널 셔플 구조

Model Compression

3) 경량모델 자동 탐색 기술

- NetAdapt



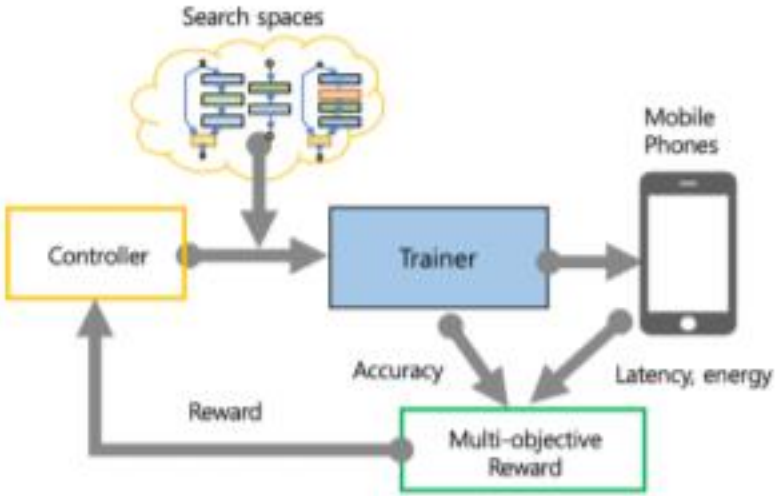
지연시간, 에너지 소모

(그림 6) 넷어탭트(NetAdapt)의 신경망 탐색 흐름

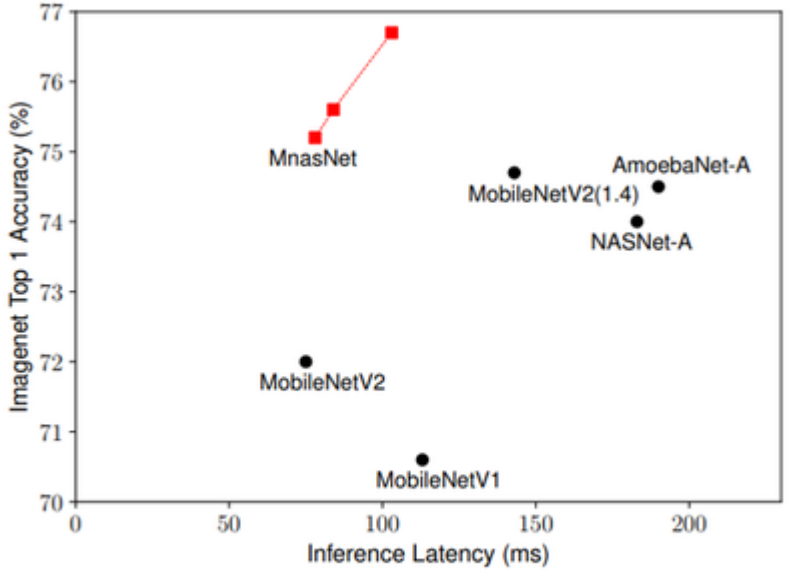
Model Compression

3) 경량모델 자동 탐색 기술

- MNasNet



(그림 7) 엠나스넷(MNasNet)의 신경망 탐색 흐름



Accuracy와 Latency 성능비교 (MNasNet이 제일 좋다)

Model Compression

모델 경량화 : 비슷한 수준의 성능을 유지한 채 더 적은 파라미터 수와 연산량을 가지는 모델을 만드는 것.

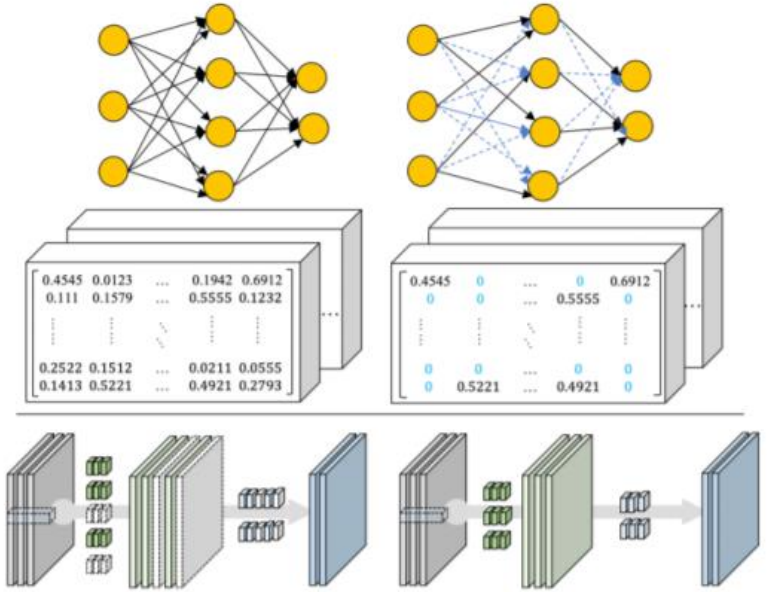
1. 경량 알고리즘 연구
   1) 모델 구조 변경
   2) 합성곱 필터 변경
   3) 자동 모델 탐색
2. **알고리즘 경량화**
   1) 모델 압축
   2) 지식 증류
   3) 하드웨어 가속화
   4) 모델 압축 자동 탐색

더 자세한 설명 :
https://computistics.tistory.com/22

Model Compression

1) 모델 압축 기술
   - 가중치 가지치기(prunning)

     :기존 신경망이 가지고 있는 가중치(Weights) 중 작은 가중치값을 모두 0으 로 하여 네트워크의 모델 크기를 줄이는 기술

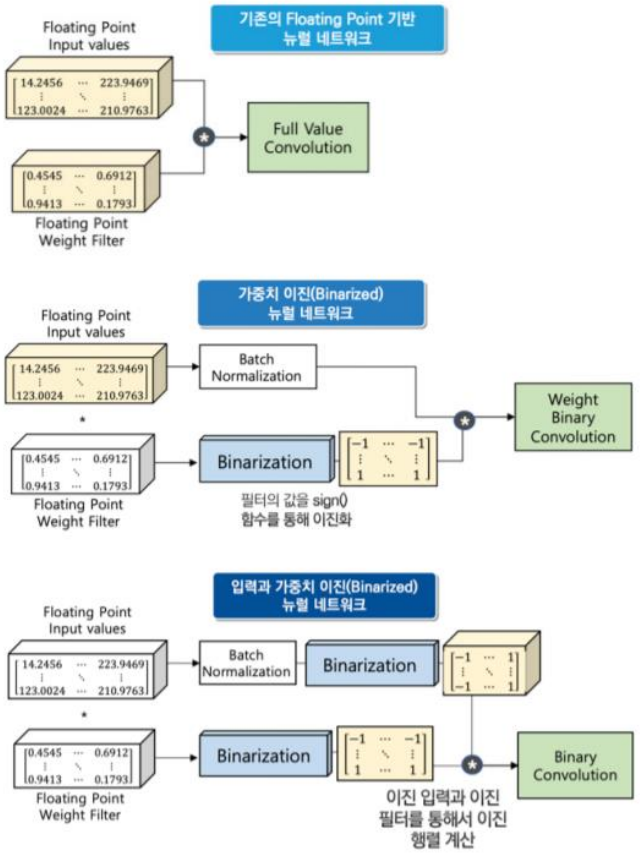


(그림 8) 가중치/채널 가지치기(Weight/Channel Pruning)의 예

Model Compression

1) 모델 압축 기술
   - 양자화 및 이진화

양자화 : 특정 비트수 만큼 줄여서 계산
이진화 : 입력 부호를 이진형태의 값으로 변환
(용량이 압축됨)



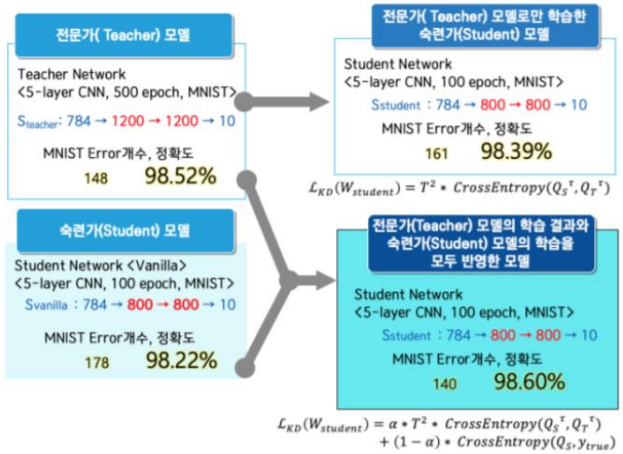
(그림 9) 이진화(Binarization)를 통한 합성곱의 예

Model Compression

1) 모델 압축 기술
   - 가중치 공유
     : 낮은 정밀도에 대한 높은 내성을 가진 신경망의 특징을 활용해 가중치를 근사하는 방법.
     : 근사값과 인덱스만 저장 -> 저장공간 절약

Model Compression

2) 지식 증류 기술(knowledge distillation)
: 앙상블(Ensemble) 기법을 통해 학습된 다수의 큰 네트워크들로부터 작은 하나의 네트워크에 지식을 전달할 수 있는 방법론 중의 하나
: Teacher모델, Student 모델 각각의 loss를 동시에 반영
: 현재 훈련된 네트워크보다 더 큰 네트워크로의 Knowledge Transfer연구도 진행 중.



(그림 10) 전문가(Teachers) 모델과 숙련가(Student) 모델의 학습 결과 예

Model Compression

3) 하드웨어 가속화 기술
   - TPU, VPU, GPU 등
   - 최근에는 경량 디바이스에서 사용 가능한 칩셋 혹은 usb스틱 형태

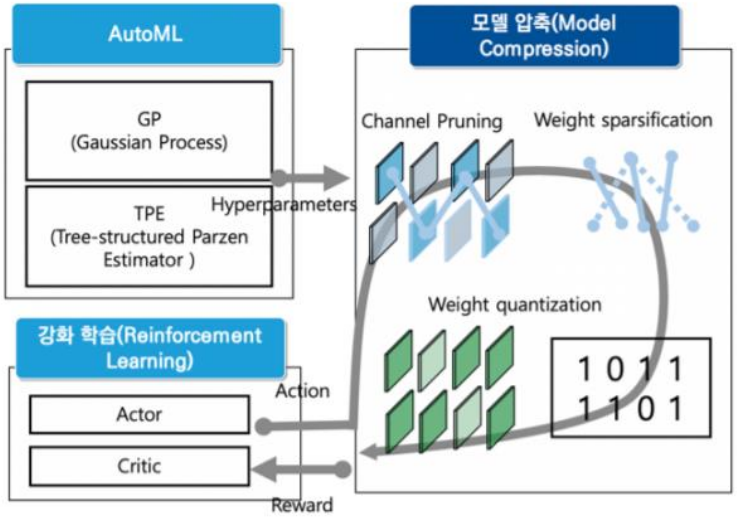
   ex)intel : 모비디우스를 통한 뉴럴 컴퓨트 스틱

   ex) NVIDIA : jetson TX2

   ex) google : edge TPU 등...

Model Compression

4) 모델 압축을 적용한 경량 모델 자동 탐색 기술



(그림 11) 강화학습을 통해 모델 압축/가속화 기법들을 자동 탐색하는 예

MobileNet

MobileNet

3 x 3 Depthwise Convolution과 1 x 1 Convolution 사이에 Batchnormalization과 ReLU가 추가

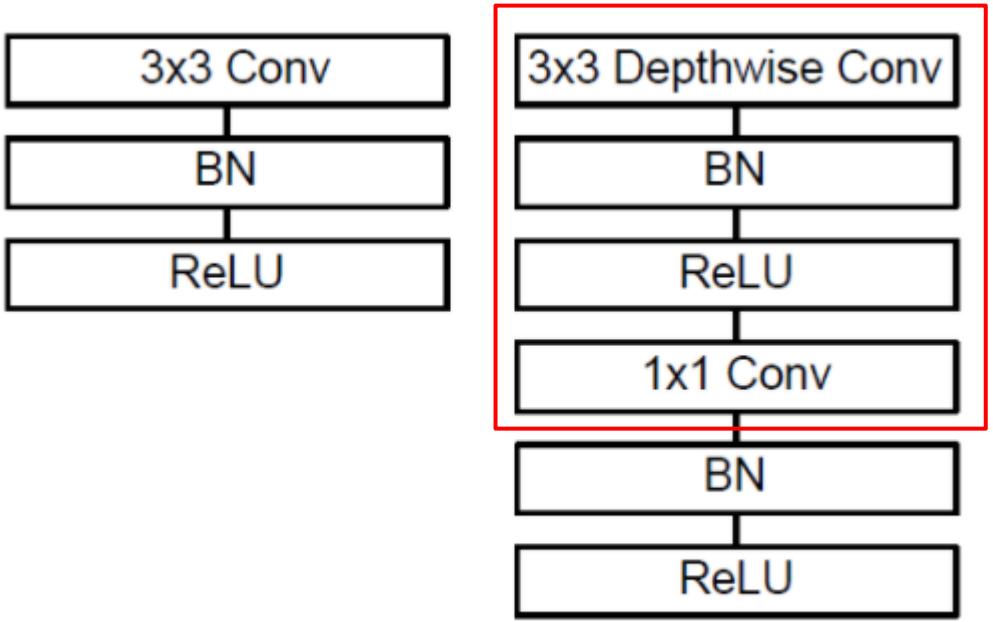


그림 2 Standard convolutional block & Depthwise Separable convolutional block

MobileNet

Depthwise separable convolution?

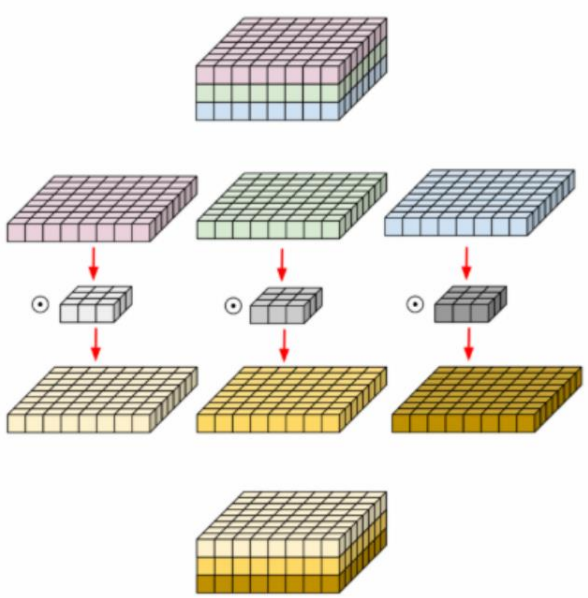
= Depth-wise convolution + Point-wise Convolution

MobileNet

Depthwise convolution?

-H*W*C의 conv output을 C단위로 분리하여 각각 conv filter을 적용하여 output을 만들고 그 결과를 다시 합치는 방식
-훨씬 **적은 파라미터**를 가지고 동일한 크기의 아웃풋을 낼 수 있음.

- 필터에 대한 연산 결과가 다른 필터로부터 독립적일 필요가 있을 경우에 특히 장점



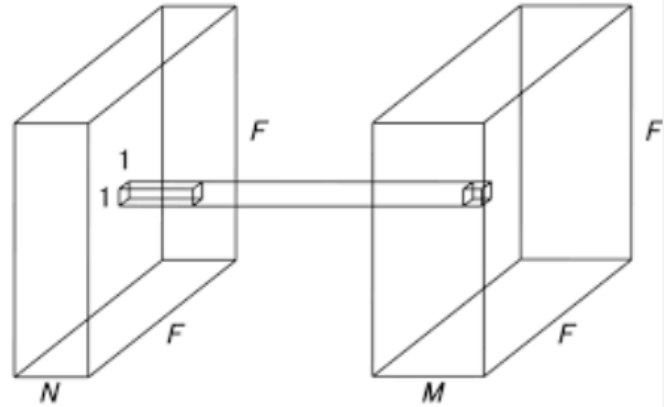
Depth-wise Convolution

MobileNet

Point-wise Convolution?
- 흔히 1x1 Conv라고 불리는 필터

- 주로 기존의 matrix의 결과를 논리적으로 다시 shuffle해서 뽑아내는 것을 목적

- 총 channel수를 줄이거나 늘리는 목적



Point-wise Convolution

MobileNet

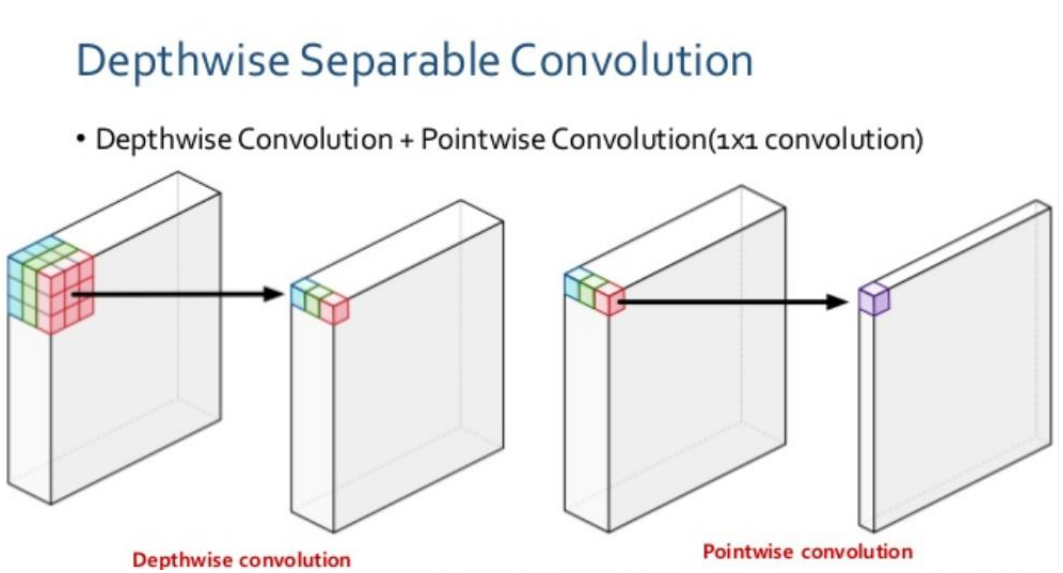
Depthwise separable convolution?

= Depthwise convolution + Point-wise Convolution

-3x3 필터로 conv 연산도 진행,

-서로 른 channel들의 정보 공유,

-파라미터 수 줄일 수 있음.



MobileNet

모바일 넷의 전체 구조

Type / Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32$ dw	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64$ dw	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
$5\times$ Conv dw / s1	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
$5\times$ Conv / s1	$1 \times 1 \times 512 \times 512$	$14 \times 14 \times 512$
Conv dw / s2	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
Conv dw / s2	$3 \times 3 \times 1024$ dw	$7 \times 7 \times 1024$
Conv / s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
Avg Pool / s1	Pool 7×7	$7 \times 7 \times 1024$
FC / s1	1024×1000	$1 \times 1 \times 1024$
Softmax / s1	Classifier	$1 \times 1 \times 1000$

그림 3 MobileNets Body Architecture

MobileNet

1. Width Multiplier

: input channel M과 output channel(filter개수와 동일) N에 a(알파)를 곱해서 채널을 줄이는 역할을 합니다.

Mults once = D_K^2

Mults 1 Channel = $D_G^2 \times D_K^2$

DC Mults = $M \times D_G^2 \times D_K^2$

Mults once = M

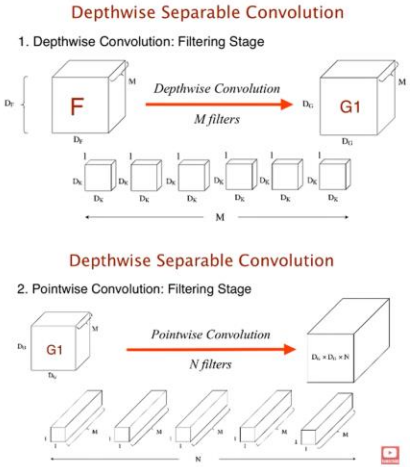
Mults 1 Kernel = $D_G \times D_G \times M$

PC Mults = $N \times D_G \times D_G \times M$

Total = DC Mults + PC Mults

$M \times D_G^2 \times D_K^2 + N \times D_G^2 \times M$

$M \times D_G^2 \, (D_K^2 + N)$



The computational cost of a depthwise separable convolution with width multiplier α is:

$$D_K \cdot D_K \cdot \alpha M \cdot D_F \cdot D_F + \alpha M \cdot \alpha N \cdot D_F \cdot D_F \quad (6)$$

- $D_K * D_K$: kernel size (filter size)
- M : input channel
- $D_F * D_F$: input
- N : output channel (filter 개수)

MobileNet

2. Resolution Multiplier
: Input의 가로 세로 크기를 줄여서 연산량을 줄이는 역할을 합니다.

$$D_K \cdot D_K \cdot \alpha M \cdot \rho D_F \cdot \rho D_F + \alpha M \cdot \alpha N \cdot \rho D_F \cdot \rho D_F \quad (7)$$

- $D_K * D_K$: kernel size (filter size)
- M : input channel
- $D_F * D_F$: input
- N : output channel (filter 개수)

MobileNet

결과 비교

Table 4. Depthwise Separable vs Full Convolution MobileNet

Model	ImageNet Accuracy	Million Mult-Adds	Million Parameters
Conv MobileNet	71.7%	4866	29.3
MobileNet	70.6%	569	4.2

MobileNet

젯슨 보드에 탑재..

Specification

- GPU: 128-core Maxwell
- CPU:Quad-core ARM A57 @ 1.43 GHz
- Memory: 4 GB 64-bit LPDDR4 25.6 GB/s
- Storage: microSD (not included)
- Video Encoder: 4K @ 30 | 4x 1080p @ 30 | 9x 720p @ 30 (H.264/H.265)
- Video Decoder: 4K @ 60 | 2x 4K @ 30 | 8x 1080p @ 30 | 18x 720p @ 30|(H.264/H.265)
- Camera: 1x MIPI CSI-2 DPHY lanes
- Connectivity: Gigabit Ethernet, M.2 Key E
- Display: HDMI 2.0 and eDP 1.4
- USB: 4x USB 3.0, USB 2.0 Micro-B
- Others: GPIO, I2C, I2S, SPI, UART
- Mechanical: 100 mm x 80 mm x 29 mm

MobileNet

Object Detection

Network	CLI argument	NetworkType enum	Object classes
SSD-Mobilenet-v1	`ssd-mobilenet-v1`	`SSD_MOBILENET_V1`	91 (COCO classes)
SSD-Mobilenet-v2	`ssd-mobilenet-v2`	`SSD_MOBILENET_V2`	91 (COCO classes)
SSD-Inception-v2	`ssd-inception-v2`	`SSD_INCEPTION_V2`	91 (COCO classes)
DetectNet-COCO-Dog	`coco-dog`	`COCO_DOG`	dogs
DetectNet-COCO-Bottle	`coco-bottle`	`COCO_BOTTLE`	bottles
DetectNet-COCO-Chair	`coco-chair`	`COCO_CHAIR`	chairs
DetectNet-COCO-Airplane	`coco-airplane`	`COCO_AIRPLANE`	airplanes
ped-100	`pednet`	`PEDNET`	pedestrians
multiped-500	`multiped`	`PEDNET_MULTI`	pedestrians, luggage
facenet-120	`facenet`	`FACENET`	faces

MobileNet

감사합니다