

물체 인식 CNN 모델과 전이학습

이건명

충북대학교 소프트웨어학과

학습 내용

- 물체인식 CNN 모델들에 대해서 알아본다.
- 전이학습에 대해서 알아본다.

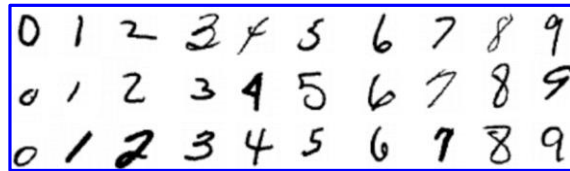
1. 물체 인식 CNN 모델

- LeNet
- AlexNet
- VGGNet
- GoogleNet
- ResNet
- ResNeXt
- DenseNet
- DPN (Dual Path Network)
- SENet
- MobileNet
- SuffleNet

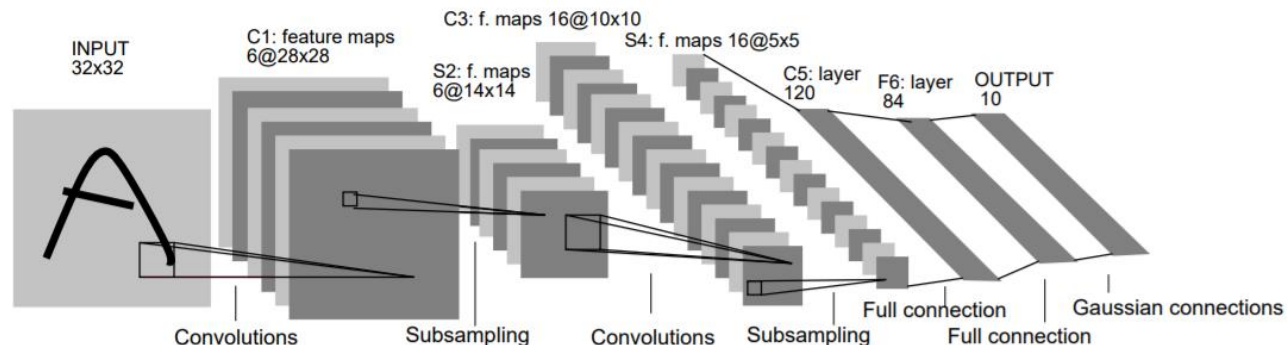
LeNet 모델

❖ LeNet 모델

- Yann LeCun 등의 제안(1998)
- **LeNet5 모델**
 - 5 계층 구조: Conv-Pool-Conv- Pool-Conv-FC-FC(SM)
- 입력 : 32x32 필기체 숫자 영상 (**MNIST** 데이터)



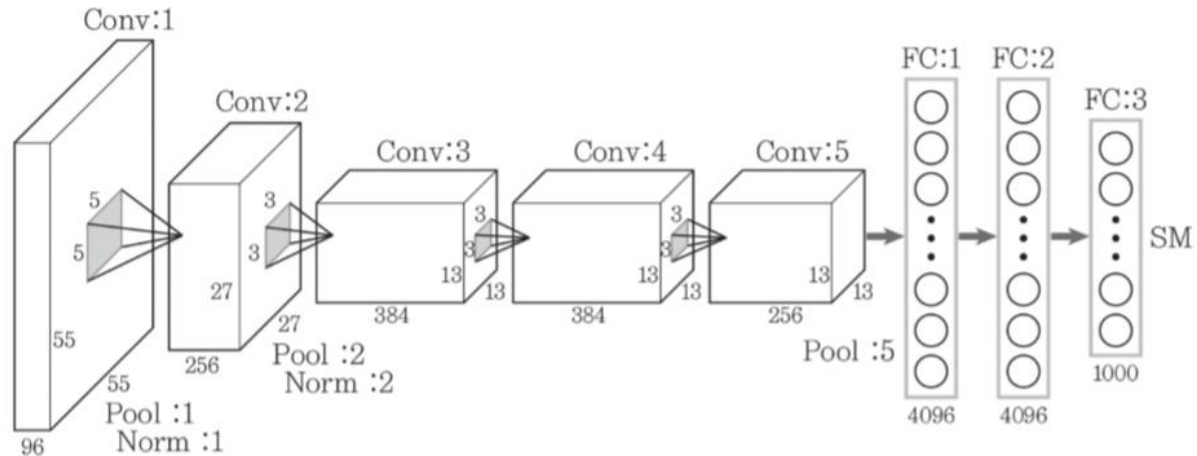
- 풀링 : 가중치x(2x2블록의 합) + 편차항
- 시그모이드 활성화 함수 사용
- 성능: 오차율 0.95%(정확도: 99.05%)



AlexNet 모델

❖ AlexNet – cont.

- 8 계층의 구조
 - Conv-Pool-Norm-Conv-Pool-Norm-Conv- Conv-Conv-Pool-FC-FC-FC(SM)

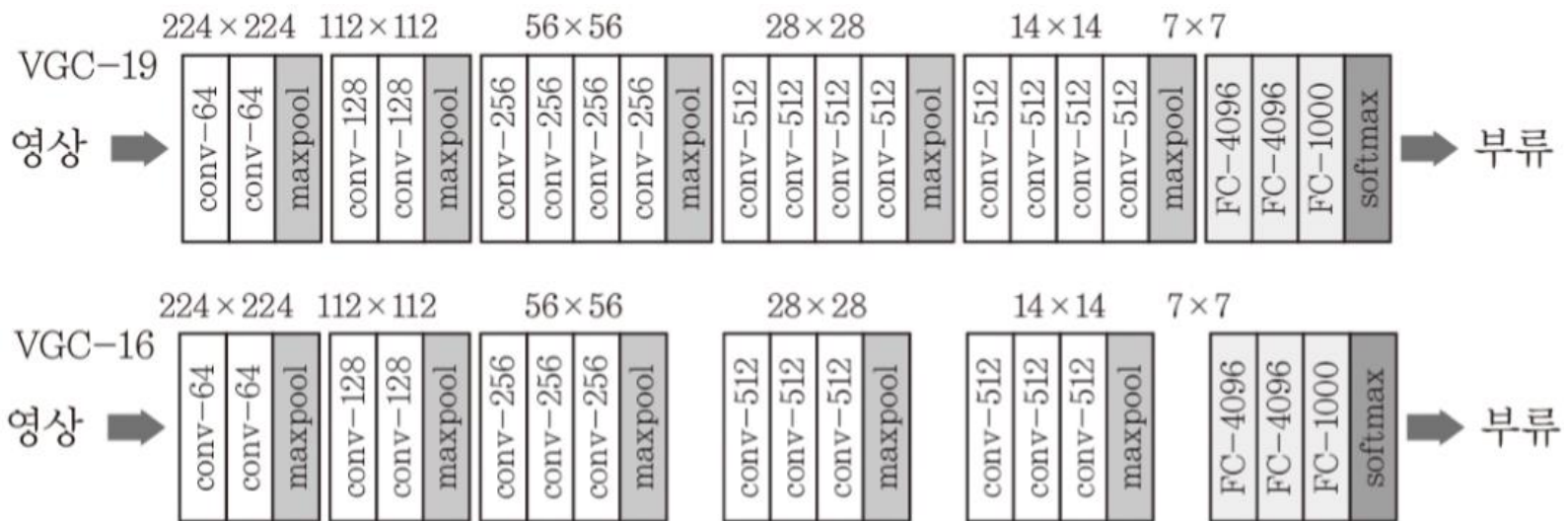


- ReLU 함수를 사용한 첫 모델
- FC 층에 드롭아웃(dropout) 기법 사용
- 최대값 풀링(max pooling) 사용

VGGNet 모델

❖ VGGNet

- 사이머니언와 지서만이 제안(2014년)
- VGG-16 모델(16개 층)
- VGG-19 모델(19개 층)
- 2014년 ILSVRC에서 2등 차지 (상위-5 오류율: 7.32%)
- 단순한 구조 (3x3 커널 사용)



VGGNet 모델

❖ VGGNet – cont.

- 모든 층에서 **3x3** 필터 사용
- **3x3** 필터 2회 적용 \Rightarrow **5x5** 필터 적용 효과
- **3x3** 필터 3회 적용 \Rightarrow **7x7** 필터 적용 효과
27 가중치 49 가중치

ReLU 3회 적용 \Rightarrow 복잡한 결정경계 표현 가능

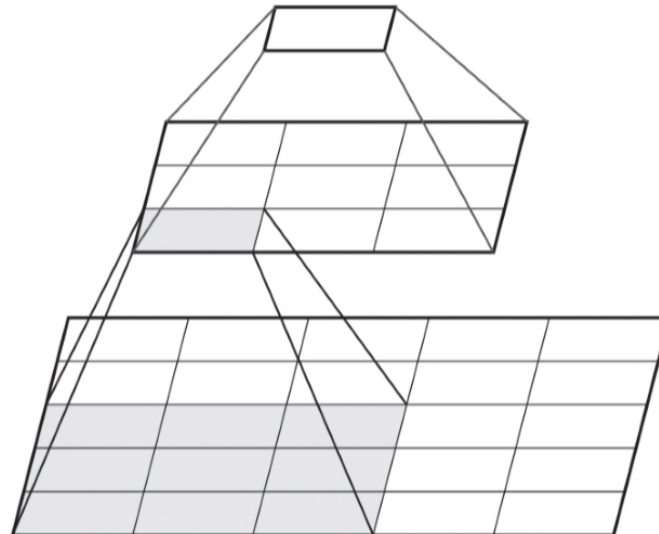
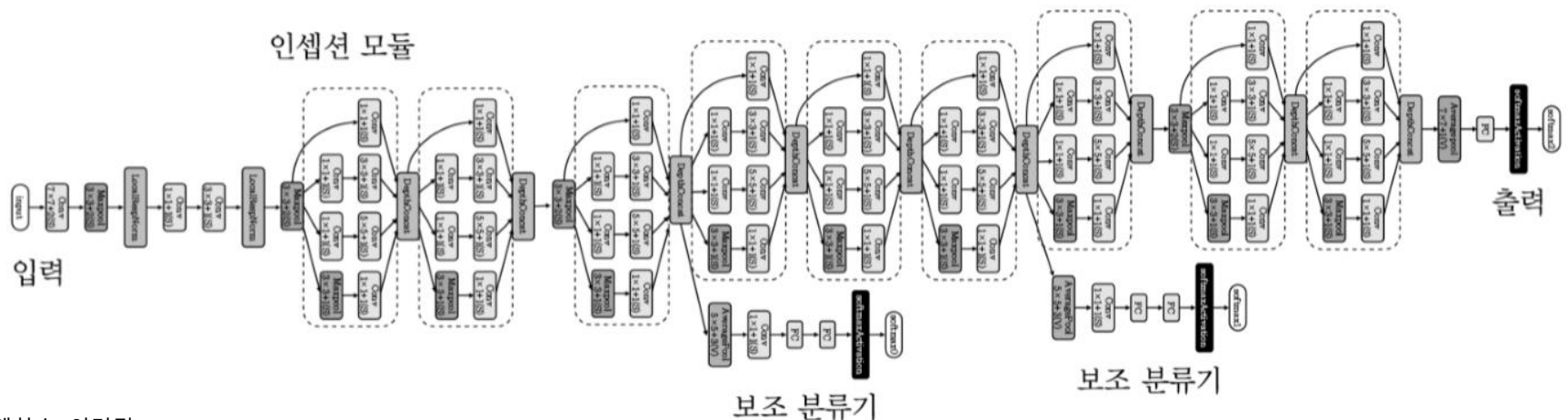


그림 5.23 2개 층의 3x3 컨볼루션에 의한 5x5 컨볼루션 구현

GoogleNet 모델

❖ GoogleNet

- 구글의 체게디 등이 개발
- 2014년 ILSVRC에서 우승(상위-5 오류율 : 6.67%)
- 22개 층의 구조
 - Conv-MPool-Conv-Incept-Incept-MPool-Incept-Incept-Incept-Incept-MPool-Incept-Incept-APool-FC-SM
 - MPool : 최대값 풀링
 - APool: 평균값 풀링
 - Incept : 인셉션(Inception)모듈



GoogleNet 모델

❖ GoogleNet - cont.

▪ 인셉션(Inception) 모듈

- 직전 층의 처리결과에 1×1 컨볼루션, 3×3 컨볼루션, 5×5 컨볼루션을 적용
- 여러 크기의 수용장에 있는 특징들을 동시에 추출

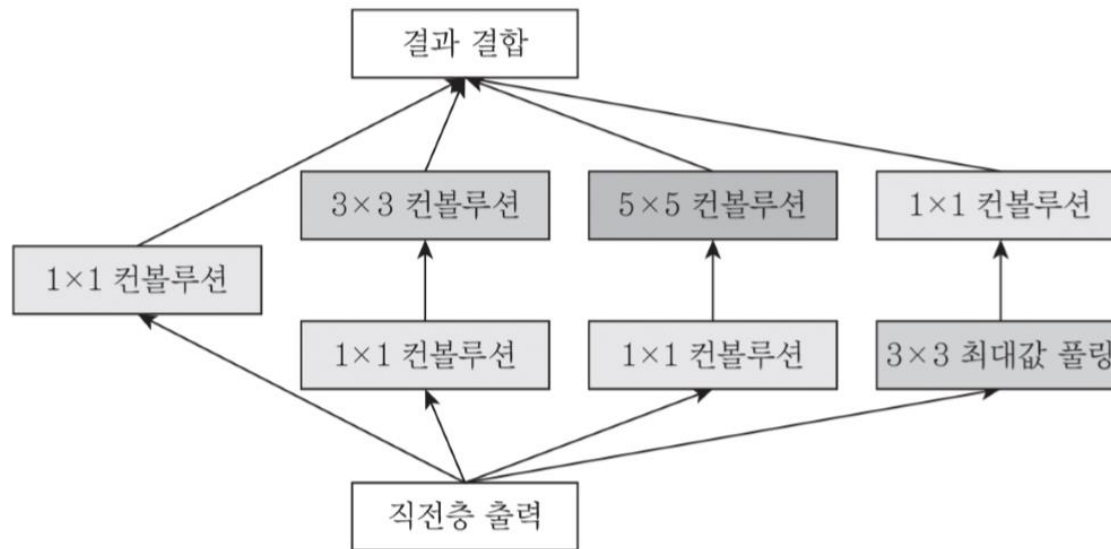


그림 5.25 GoogleNet에서 사용되는 인셉션(Inception) 모듈

GoogleNet 모델

❖ GoogleNet - cont.

▪ 1x1 컨볼루션

- 동일한 위치의 특징지도의 값을 필터의 가중치와 선형결합
- 1x1 컨볼루션 필터의 개수를 조정하여 출력되는 특징지도의 개수를 조정
 - $224 \times 224 \times 500 \Rightarrow (1 \times 1 \times 500) @ 120 \Rightarrow 224 \times 224 \times 120$

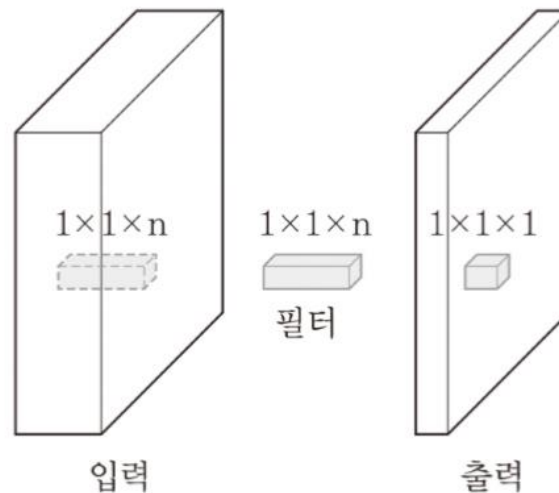
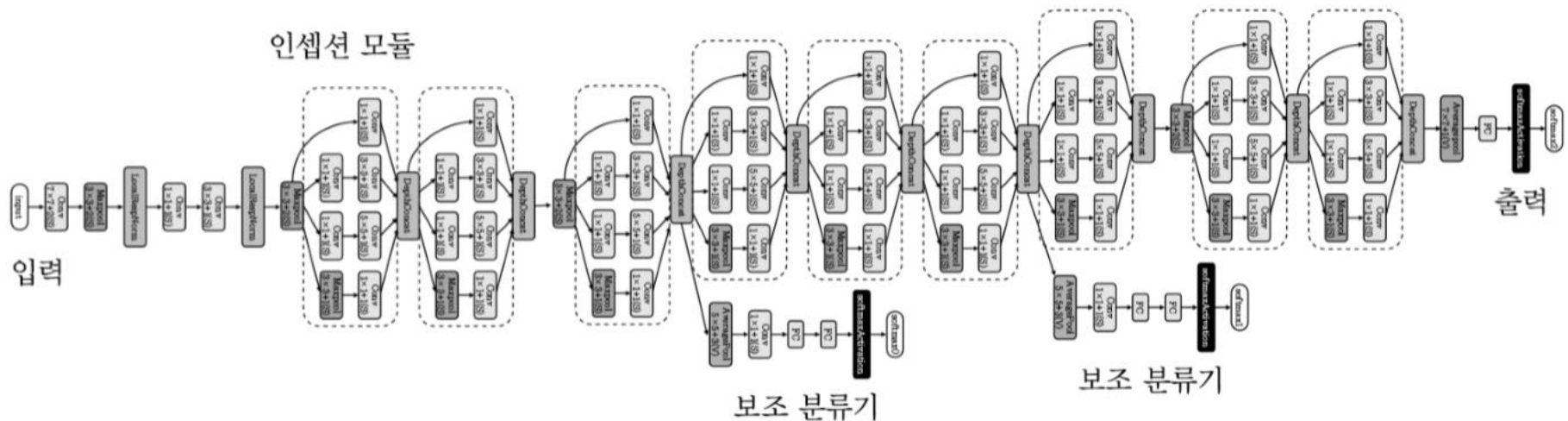


그림 5.26 1x1 컨볼루션의 동작

GoogleNet 모델

❖ GoogleNet - cont.

- 마지막 계층: 소프트맥스
 - 22개 층 모델이지만, AlexNet 모델에 비해 가중치 개수는 10% 증가
- 기울기 소멸 문제 완화 장치
 - 4번째, 7번째 계층에 보조 분류기 추가
 - 보조 분류기를 통해 그래디언트 정보 제공



ResNet 모델

❖ ResNet (Residual Net)

- 카이밍 허 등이 개발
- 2015년 ILSVRC에서 우승(상위-5 오류율: 3.75%)
- 152개 층의 모델
 - Conv-Mpool
 - [Conv-ReLU-Conv-ReLU-Conv-ReLU]x3
 - [Conv-ReLU-Conv-ReLU-Conv-ReLU]x8
 - [Conv-ReLU-Conv-ReLU-Conv-ReLU]x36
 - [Conv-ReLU-Conv-ReLU-Conv-ReLU]x3
 - APool-FC-SM

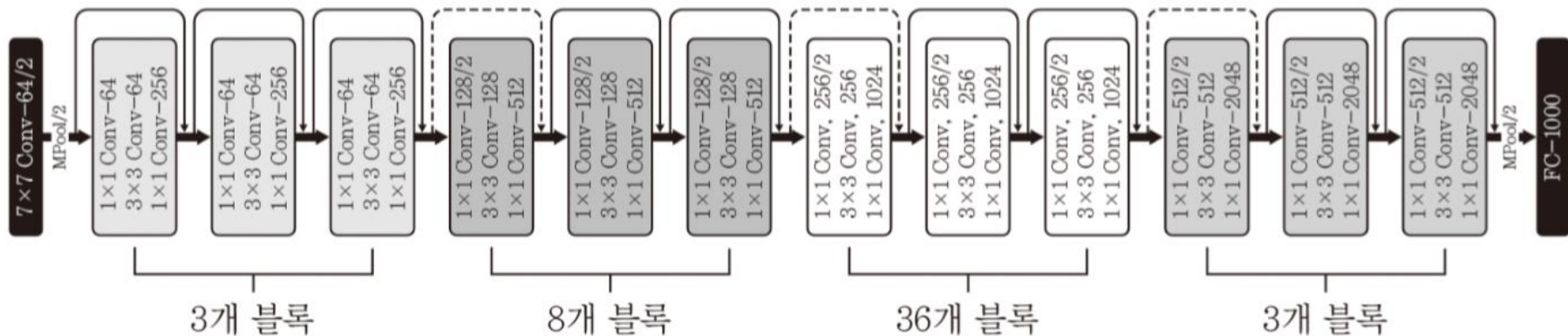
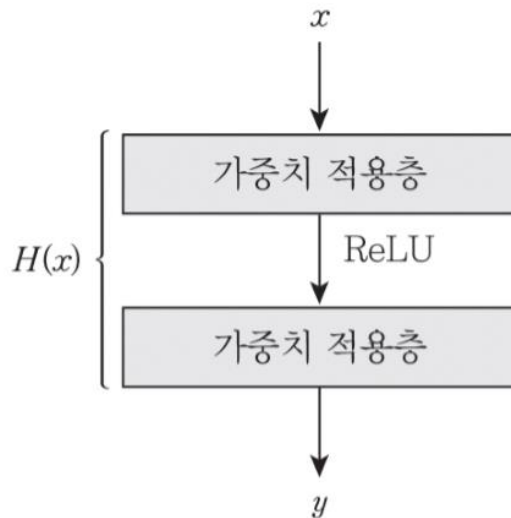


그림 5.27 ResNet의 구성
기계학습, 이진명

ResNet 모델

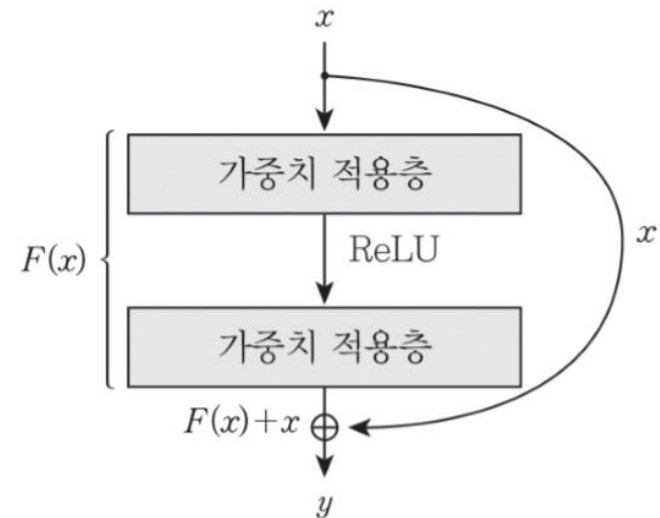
❖ ResNet – cont.

- 다수의 층 사용
 - 상위 계층에서 의미있는 특징 추출 가능
 - 다수 계층 사용시 기울기 소멸 문제 발생
- 잔차 모듈(residual module)



기존 신경망

$$y = H(x)$$



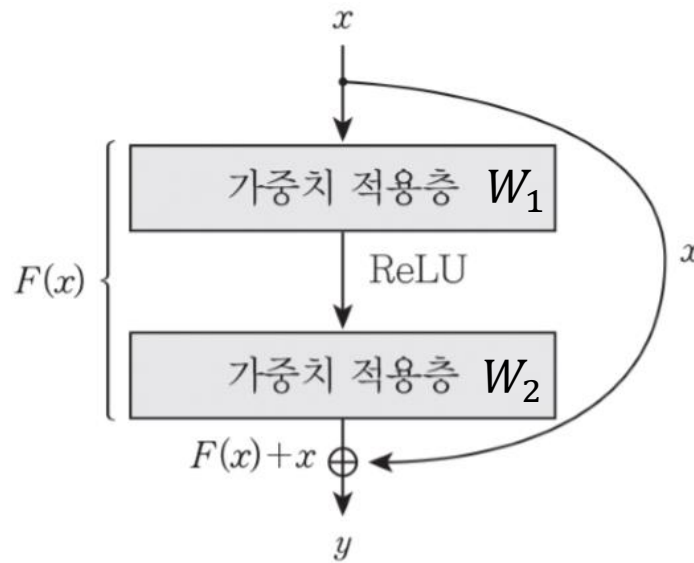
ResNet의 잔차 모듈

$$F(x) = y - x; \quad y = F(x) + x.$$

ResNet 모델

❖ ResNet – cont.

▪ 잔차 모듈



- 지름길 연결
- 항등 사상

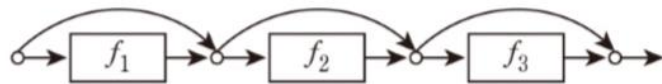
▪ 잔차 모듈 $F(x)$ 의 학습

$$y = F(x) + x = W_2 \rho(W_1 x) + x$$

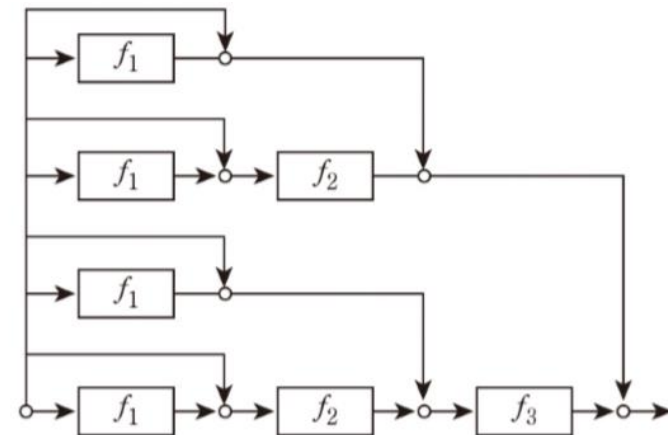
ResNet 모델

❖ 잔차 모듈의 특징

- 기대하는 출력과 유사한 입력이 들어오면 영벡터에 가까운 값을 학습
⇒ 입력의 작은 변화에 민감 ⇒ 잔차 학습
- 다양한 경로를 통해 복합적인 특징 추출
 - 필요한 출력이 얻어지면 컨볼루션 층을 건너뛸 수 있음
 - 다양한 조합의 특징 추출 가능



(a)



(b)

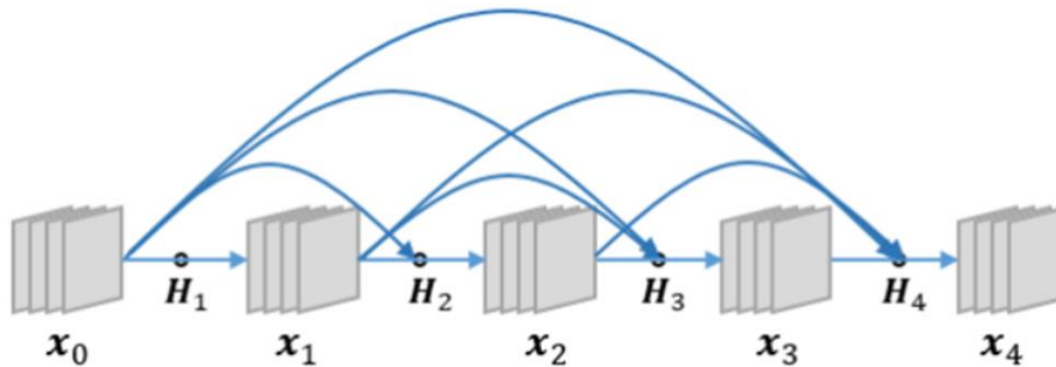
그림 5.29 ResNet의 잔차 모듈의 직렬 연결의 효과

(a) 잔차 모듈 구성 (b) 가능한 잔차 모듈 경유 경로

DenseNet 모델

❖ DenseNet

- 가오 후앙(Gao Huang) 등이 개발 (2016)
- 각 층은 모든 앞 단계에서 올 수 있는 지름길연결 구성



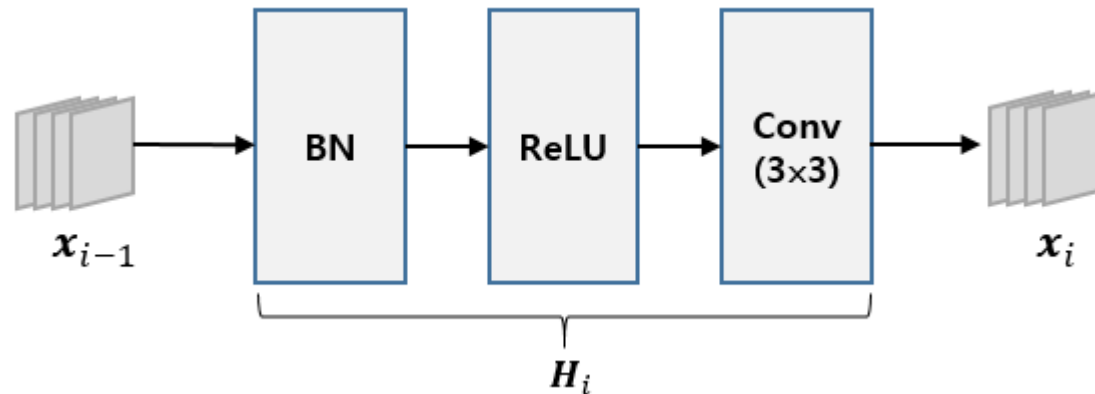
$$x_i = H_i([x_0, x_1, \dots, x_{i-1}])$$

배치 정규화, ReLU, 컨볼루션 연산

DenseNet 모델

❖ DenseNet – cont.

- 노드의 연산: H_i
 - 배치 정규화(BN)-ReLU-(3x3 컨볼루션)
 - 각 층은 입력 특징지도와 같은 차원의 특징지도 생성



- 병목층
 - 1x1 컨볼루션
 - 출력되는 특징지도의 채널 수 축소
- 병목층이 있는 층
 - BN-ReLU-(1x1 컨볼루션)-BN-ReLU-(3x3 컨볼루션)

DenseNet 모델

❖ DenseNet – cont.

- 특징지도의 크기를 줄이기 위해 풀링 연산 적용 필요
- 밀집 블록(dense block)과 전이층(transition layer)으로 구성
 - 전이층 : 1x1 컨볼루션과 평균값 풀링(APool)으로 구성



그림 5.35 밀집 블록으로 구성된 DenseNet

DPN 모델

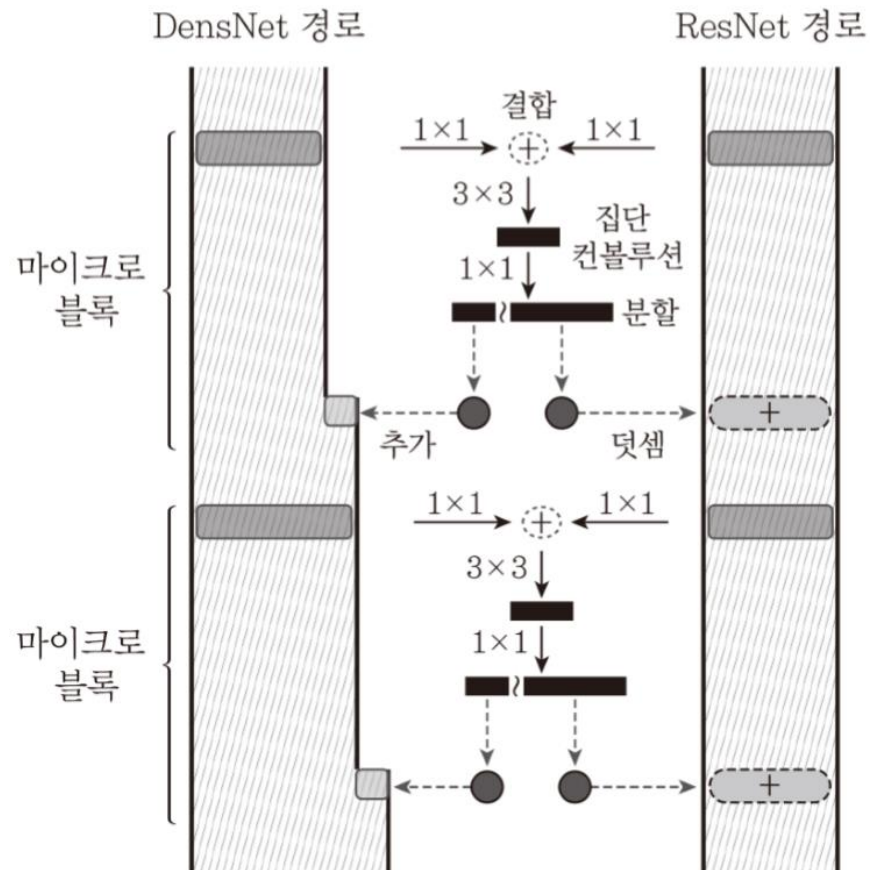
❖ DPN (Dual Path Network)

- ResNet과 DenseNet을 결합한 모델
- **ResNet**
 - 이전 단계의 동일한 특징 정보가 각 단계에 전달되어 이들 특징을 재사용하도록 하는 경향
 - 상대적으로 이전 단계의 특징들로부터 새로운 특징을 만드는 것에는 소극적
- **DenseNet**
 - 새로운 특징이 추출될 가능성이 높음
 - 이전에 추출된 특징이 다시 추출될 가능성도 높음

DPN 모델

❖ DPN – cont.

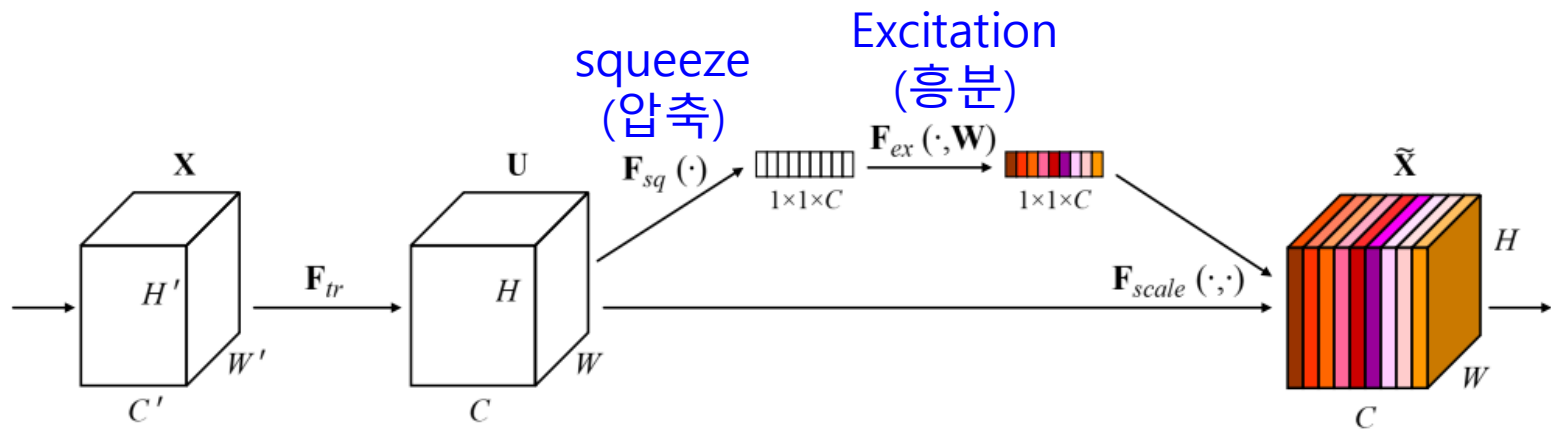
- 마이크로 블록에서 DenseNet과 ResNet의 특징 결합



SENet 모델

❖ SENet(Squeeze and excitation networks)

- SE 블록 사용
 - 채널 간의 상호 의존성(interdependencies between channels)을 모델링하여 채널별 특징 반응(channel-wise feature responses)을 적응적으로 재조정
- 2017년 ILSVRC 우승 모델



SENet 모델

❖ SENet(Squeeze and excitation networks)

▪ SE 블록

- 채널 간의 상호 의존성을 모델링하여 **채널별 특징 반응**을 적응적으로 재조정

▪ 압축(squeeze) 연산

- 채널별 통계량 계산 위해 **전역 평균 풀링(global average pooling)** 수행

$$z_c = \mathbf{F}_{sq}(\mathbf{u}_c) = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W u_c(i, j).$$

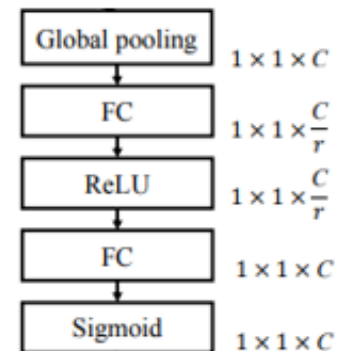
▪ 흥분(excitation) 연산

- 임베딩(embedding)을 입력으로 가져와서 채널별로 변경된 가중치들을 생성하는 self-gating mechanism

$$\mathbf{s} = \mathbf{F}_{ex}(\mathbf{z}, \mathbf{W}) = \sigma(g(\mathbf{z}, \mathbf{W})) = \sigma(\mathbf{W}_2 \delta(\mathbf{W}_1 \mathbf{z}))$$

$$\tilde{\mathbf{x}}_c = \mathbf{F}_{scale}(\mathbf{u}_c, s_c) = s_c \mathbf{u}_c$$

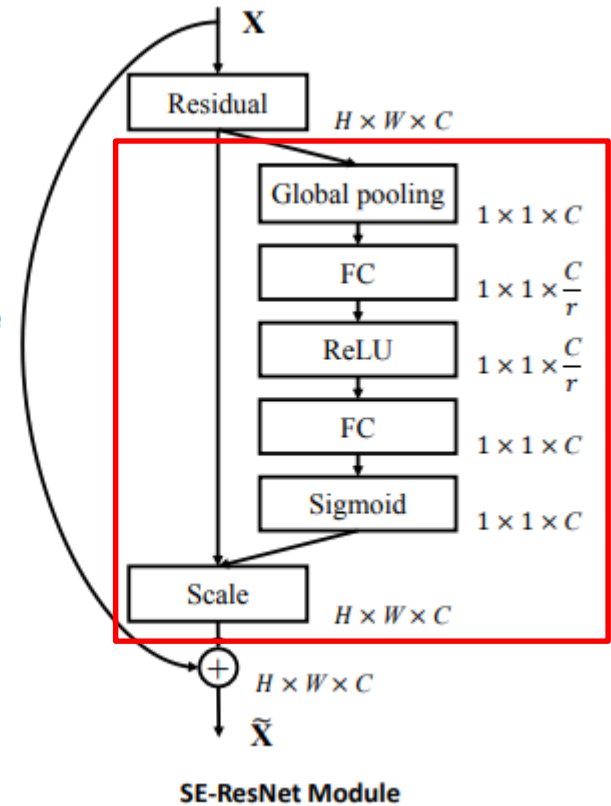
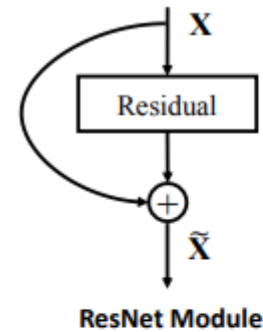
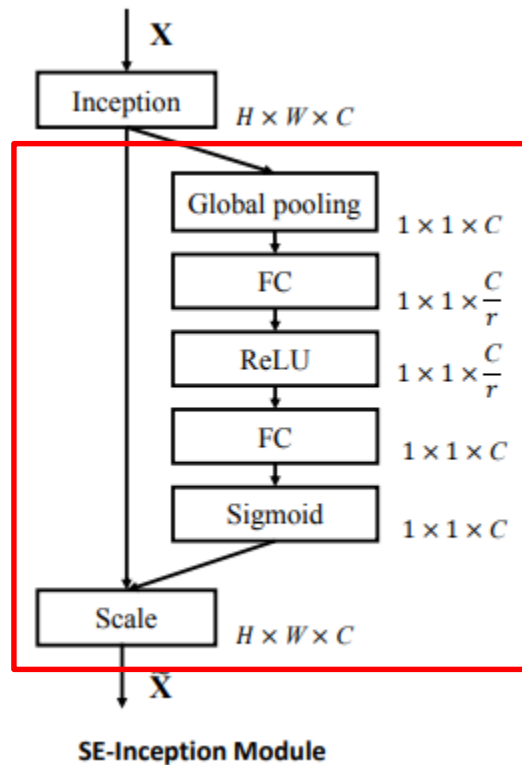
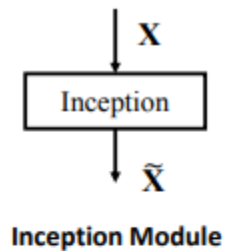
sigmoid ReLU



SENet 모델

❖ CNN 모델에 SE 블록 삽입 형태

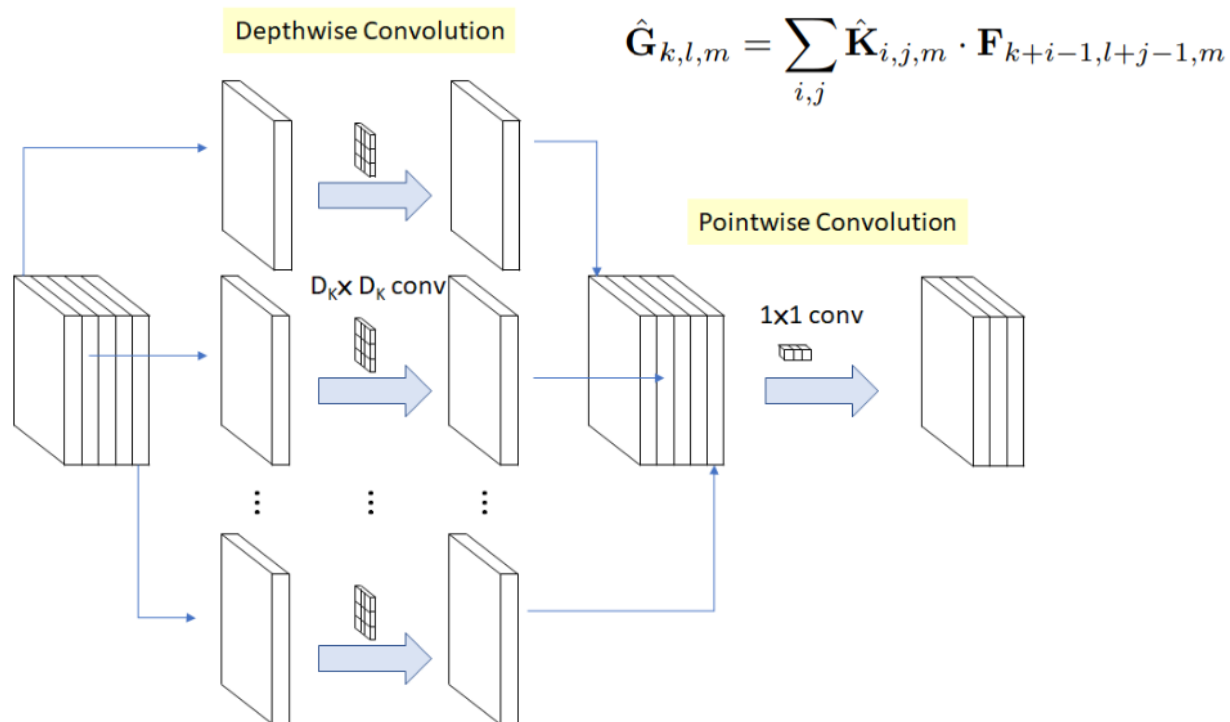
- 기존 모델에 추가하여 사용



MobileNet 모델

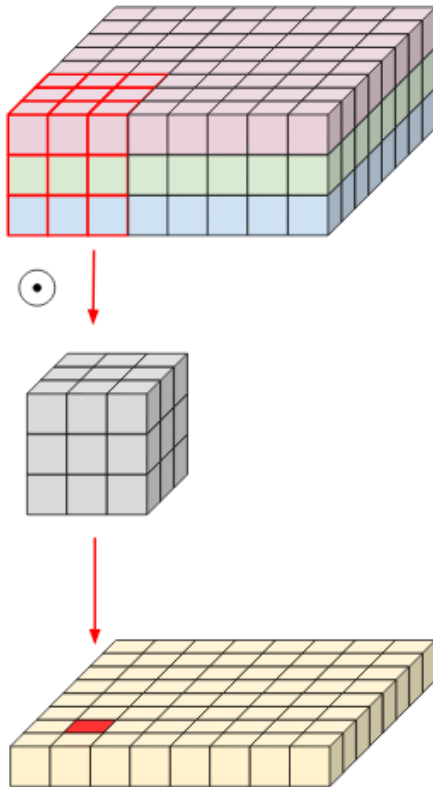
❖ MobileNet 모델

- 컴퓨팅 자원 제약이 있는 환경을 위한 CNN 모델
- **깊이별 분할 컨볼루션**(Depthwise separable convolution) 사용
 - **깊이별 컨볼루션**(depthwise convolution)
+ **위치별 컨볼루션**(pointwise convolution)

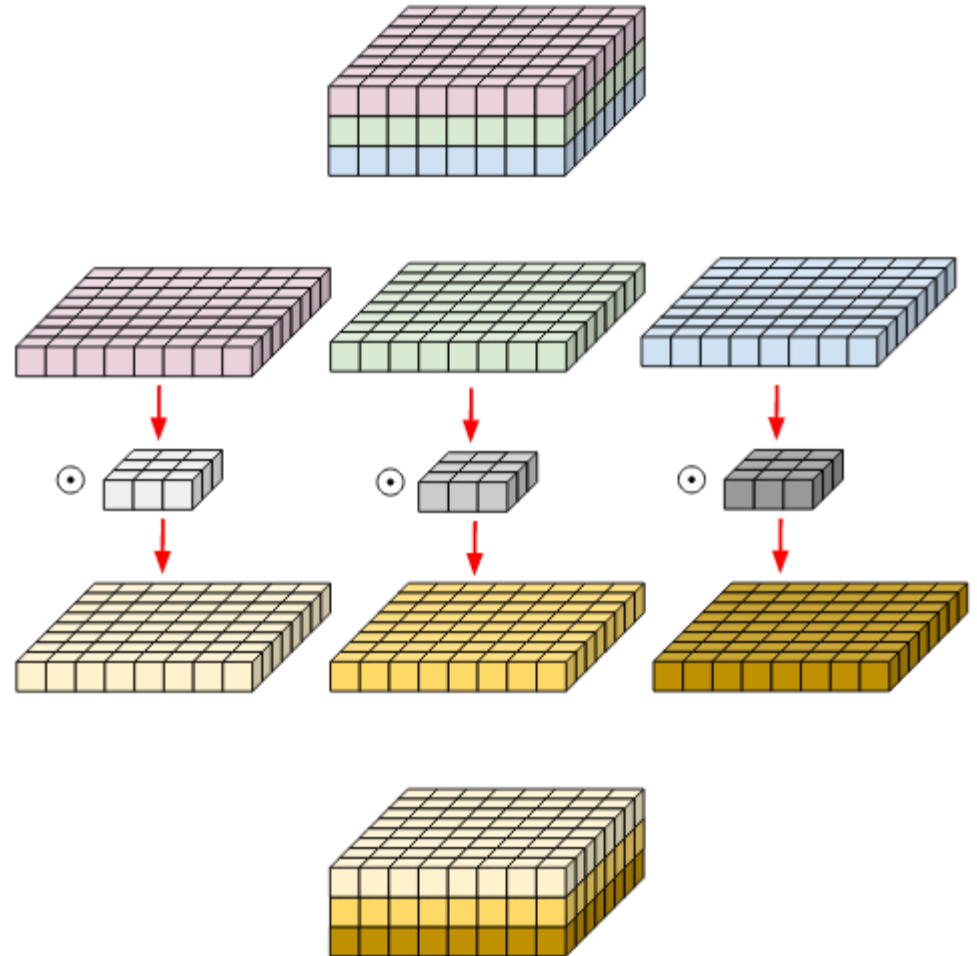


컨볼루션의 형태

❖ 일반 컨볼루션 vs 깊이별 컨볼루션



일반 컨볼루션

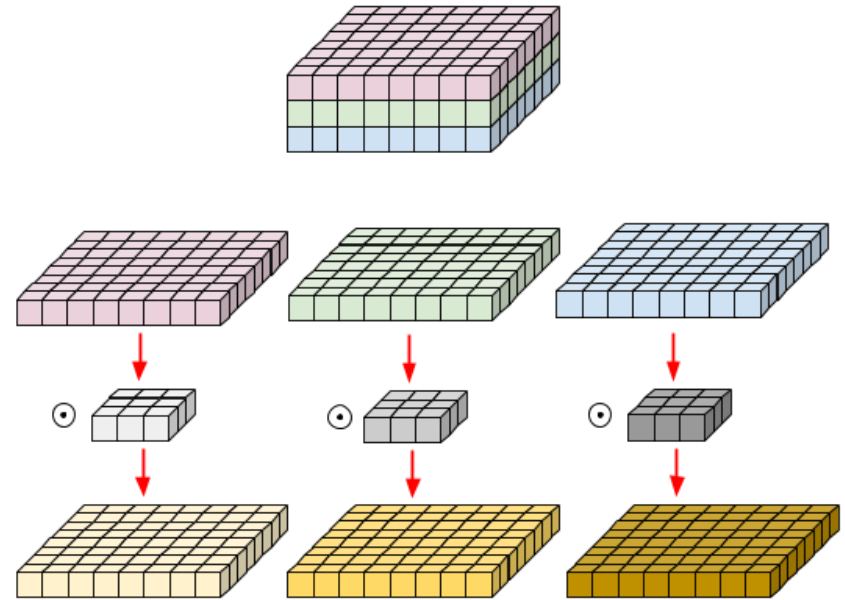


깊이별 컨볼루션

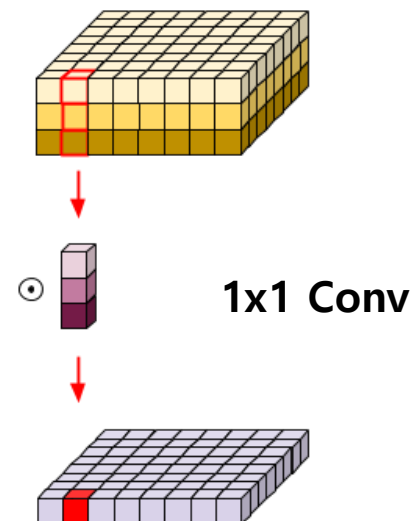
컨볼루션의 형태

❖ 깊이별 분할 컨볼루션

깊이별 컨볼루션



위치별 컨볼루션



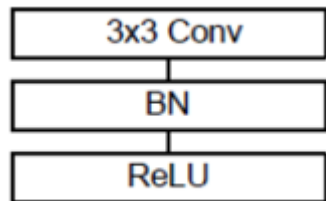
MobileNet 모델

❖ MobileNet 모델의 계산 비용

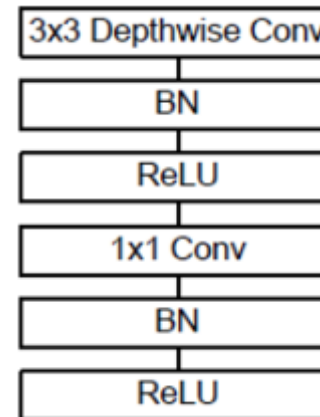
- 입력 : $D_F \times D_F \times M$ 특징지도 F
- 출력 : $D_F \times D_F \times N$ 특징지도 G
 - D_F : 특징지도의 폭과 높이
 - M : 입력 깊이 (no. of input channels)
 - N : 출력 깊이 (no. of output channels)
- 표준 컨볼루션 층(Standard convolutional layer)
 - 컨볼루션 커널의 차원 K
 - $D_K \times D_K \times M \times N$
 - 계산비 용
 - $D_K \cdot D_K \cdot M \cdot N \cdot D_F \cdot D_F$
- 깊이별 분할 컨볼루션 (Depthwise separable convolution)
 - 계산 비용
$$D_K \cdot D_K \cdot M \cdot D_F \cdot D_F + M \cdot N \cdot D_F \cdot D_F$$

MobileNet 모델

- ❖ BN(batch normalization)과 ReLU를 포함한 깊이별 분할 컨볼루션 (Depthwise separate convolution)



일반 컨볼루션

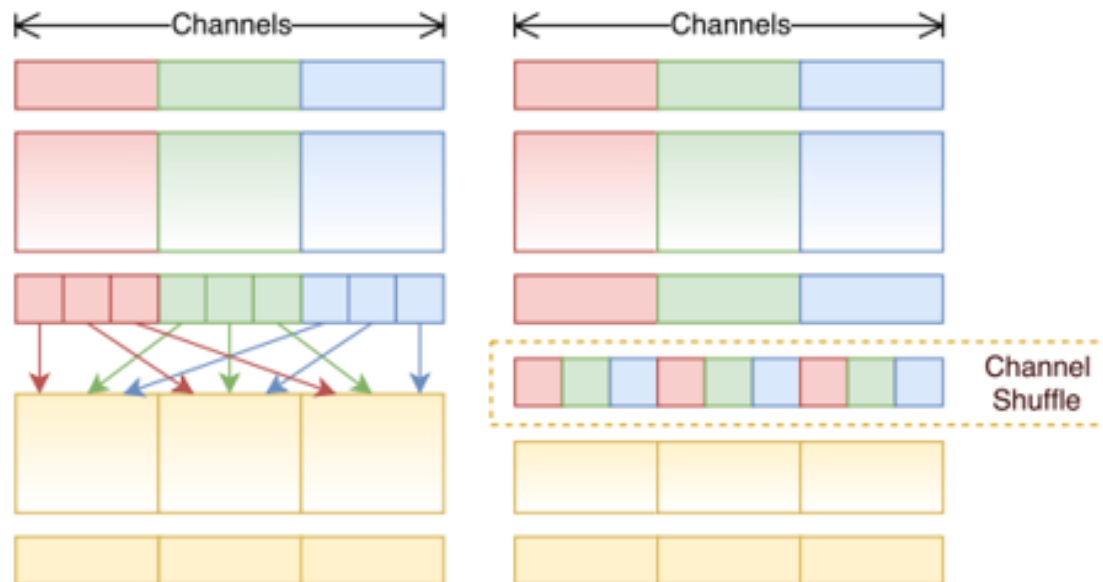


깊이별 분할 컨볼루션

ShuffleNet 모델

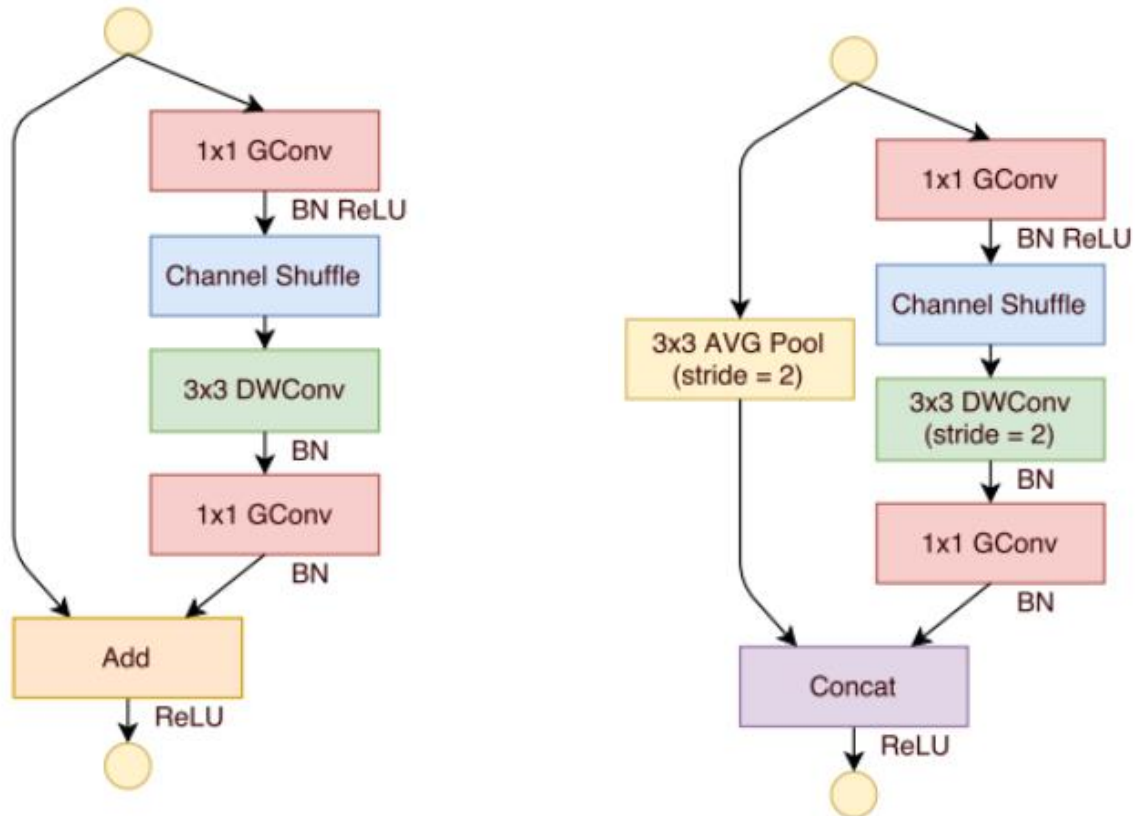
❖ ShuffleNet 모델

- 컴퓨팅 자원 제약이 있는 **모바일 단말기**를 위한 모델
- 정확도 유지 및 계산 비용 축소
- 위치별 집단 컨볼루션(**pointwise group convolution**)과 채널 섞기 (**channel shuffle**) 연산 사용



ShuffleNet 모델

❖ ShuffleNet 모듈



모델별 성능 비교

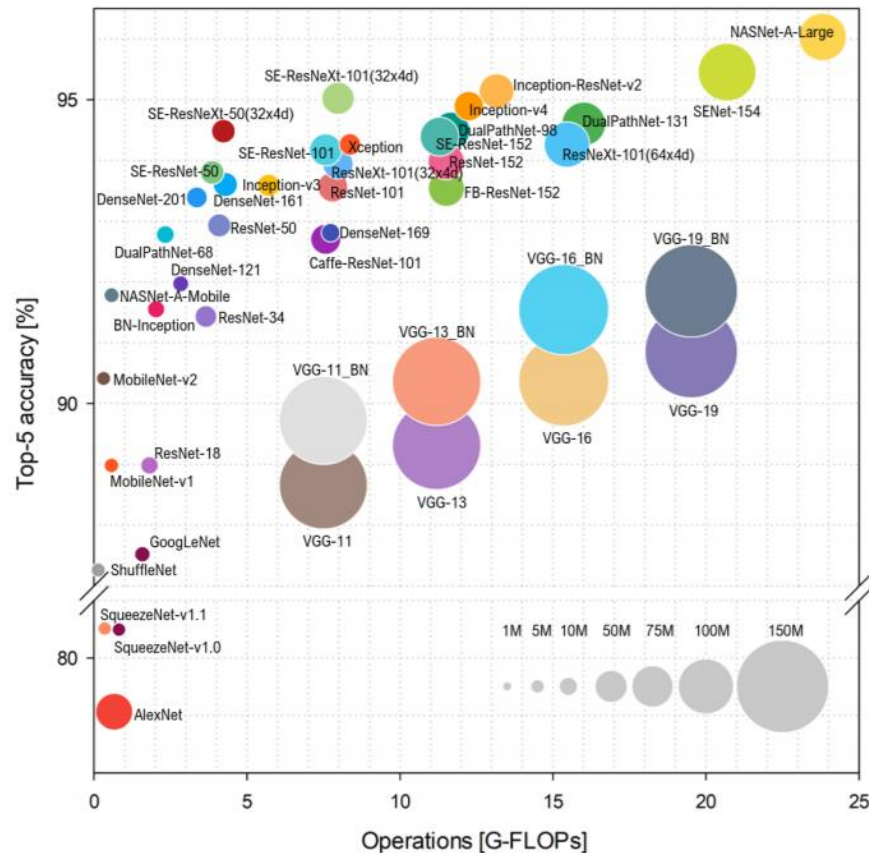
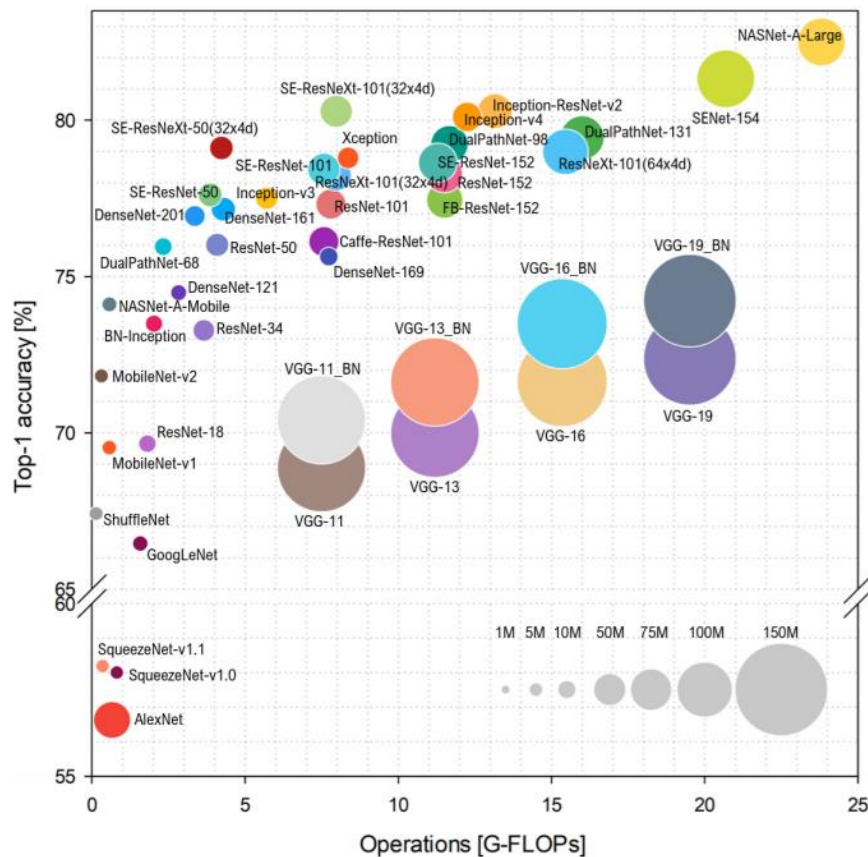


FIGURE 1: Ball chart reporting the Top-1 and Top-5 accuracy vs. computational complexity. Top-1 and Top-5 accuracy using only the center crop versus floating-point operations (FLOPs) required for a single forward pass are reported. The size of each ball corresponds to the model complexity. (a) Top-1; (b) Top-5.

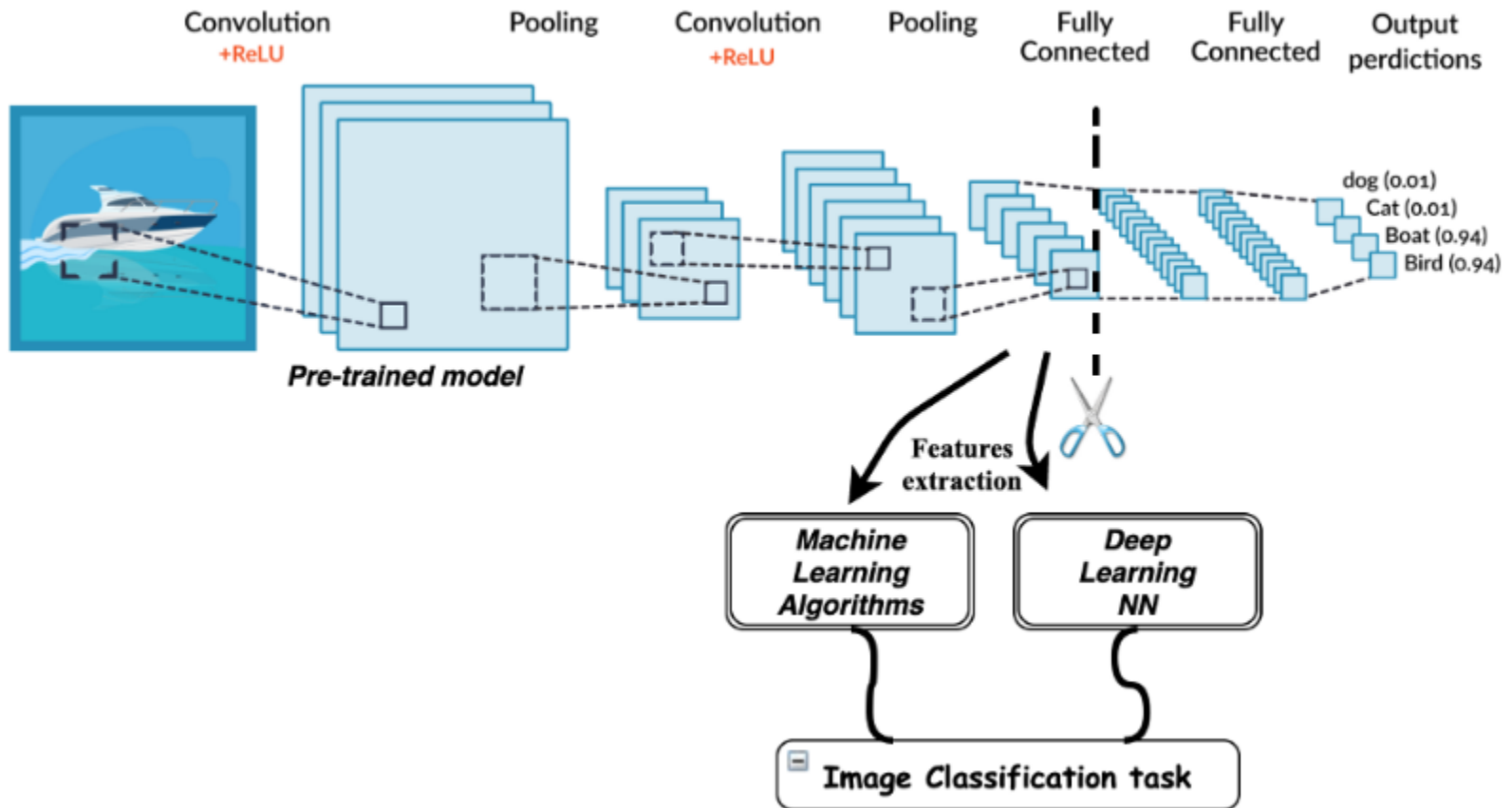
2. 전이 학습

❖ 전이 학습(transfer learning)

- 큰 규모의 딥러닝 신경망을 학습시킬 때는, **많은 학습 데이터와 상당한 학습 시간**이 필요
- 대규모 영상 데이터베이스인 **ImageNet 데이터**를 학습한 여러 컨볼루션 신경망 모델 공개
- 공개된 모델을 가져다가 누구나 자신의 문제가 적용해 볼 수도 있고, 모델의 일부 활용 가능
- **학습된 컨볼루션 신경망의 컨볼루션 층들을 가져오고 뒤 단계에서 분류하는 다층 퍼셉트론 모델을 붙여서 학습**

전이 학습

❖ 전이 학습의 형태



전이 학습

❖ PyTorch 모델

```
import torchvision.models as models  
  
resnet18 = models.resnet18(pretrained=True)  
alexnet = models.alexnet(pretrained=True)  
squeezenet = models.squeezenet1_0(pretrained=True)  
vgg16 = models.vgg16(pretrained=True)  
densenet = models.densenet161(pretrained=True)  
inception = models.inception_v3(pretrained=True)  
googlenet = models.googlenet(pretrained=True)  
shufflenet = models.shufflenet_v2_x1_0(pretrained=True)  
mobilenet = models.mobilenet_v2(pretrained=True)  
resnext50_32x4d = models.resnext50_32x4d(pretrained=True)  
wide_resnet50_2 = models.wide_resnet50_2(pretrained=True)  
mnasnet = models.mnasnet1_0(pretrained=True)
```

Quiz

1. VGGNet 모델은 3x3 컨볼루션을 사용하여 많은 층을 갖더라도 네트워크 구조가 단순하다. (O,X)
2. 인셉션 모델을 사용하면 여러 크기의 수용장에 대한 특징을 동시에 추출할 수 있다. (O,X)
3. ResNet 모델이 많은 층을 갖게 된 것에는 지름길 연결을 사용한 덕분이다. (O,X)
4. DPN에서 마이크로 블록은 채널 교환을 통해 채널 간의 상관관계를 반영하게 한다. (O,X)
5. SENet 모델은 다른 기존 모델에 SENet 모듈을 적용함으로써 성능을 개선하게 한다. (O,X)
6. SuffleNet 모델은 깊이별 분할 컨볼루션을 사용하여 계산 비용을 줄인다. (O,X)
7. 전이 학습을 하게 되면 학습된 기존 모델을 사용하여 정확도를 높이는 데는 도움이 되지만, 일반 학습에서도 더 많은 학습 데이터를 사용해야 한다. (O,X)