

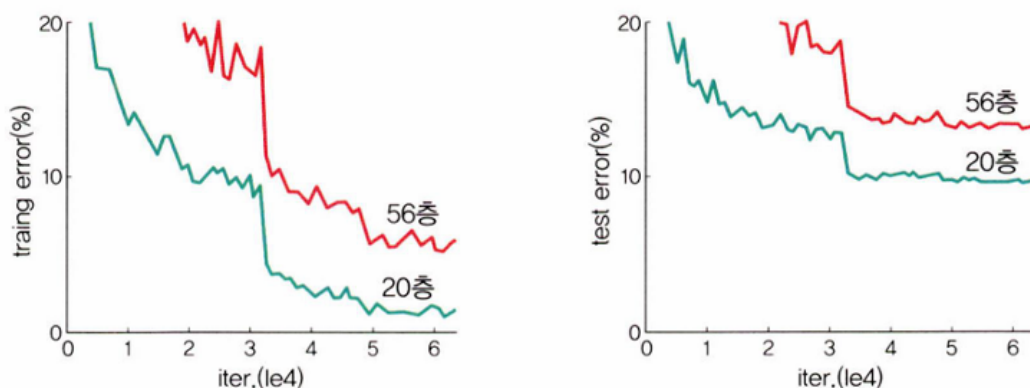


# Week 9\_예습과제\_김정은

## 6.1.5 ResNet

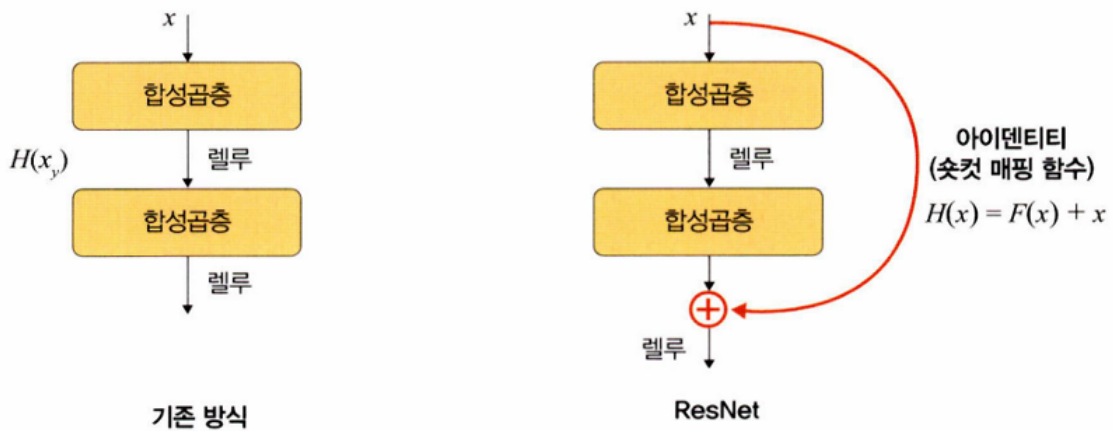
- 마이크로소프트에서 개발한 알고리즘으로, "Deep Residual Learning for Image Recognition" 이라는 논문에서 발표됨.
- 깊어진 신경망을 효과적으로 학습하기 위한 방법, 레지듀얼(Residual) 개념을 고안함
- 일반적으로 신경망 깊이가 깊어질수록 딥러닝 성능은 좋아질 것 같지만, 그렇지 않음.
  - 신경망은 깊이가 깊어질수록 성능이 좋아지다가 일정한 단계에 다다르면 성능이 나빠짐.

▼ 그림 6-24 네트워크 56층이 20층보다 더 나쁜 성능을 보임



- **레지듀얼 블록** : 기울기가 잘 전파될 수 있도록 일종의 숏컷(shortcut, skip connection)을 만들어준다.
- **필요성** : GoogLeNet은 층이 총 22개로 구성된 것에 비해 ResNet은 층이 152개로 구성 → 기울기 소멸 문제의 위험성 → 숏컷을 두어 기울기 소멸 문제 방지

♥ 그림 6-25 ResNet 구조



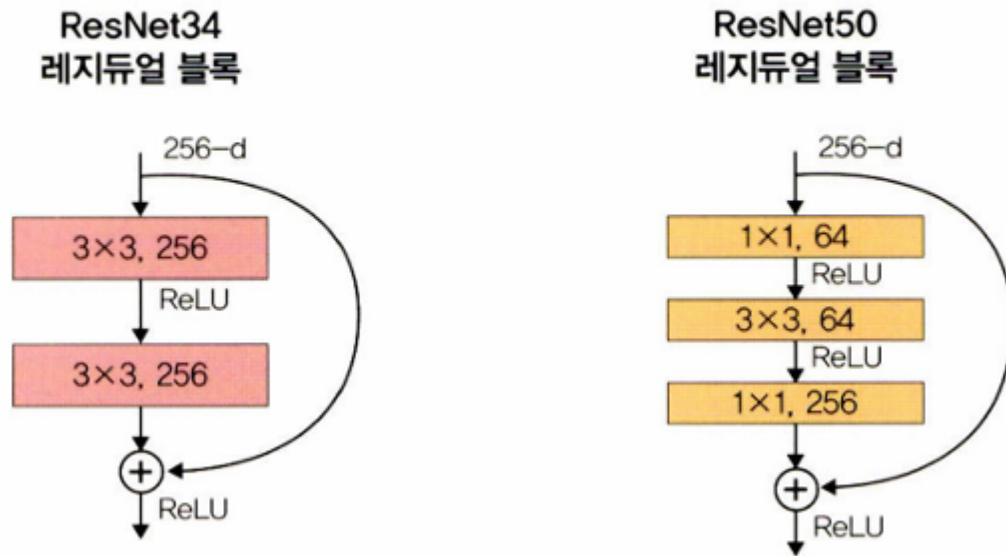
- 블록(block) : 계층의 묶음. 합성곱층을 하나의 묶음으로 묶은 것.
- 아래 그림에서 색깔로 블록을 구분, 이렇게 묶인 계층들을 하나의 레지듀얼 블록이라고 한다.
- 그리고 이러한 레지듀얼 블록을 여러 개 쌓은 것 → ResNet

♥ 그림 6-26 ResNet 모델 전체 네트워크



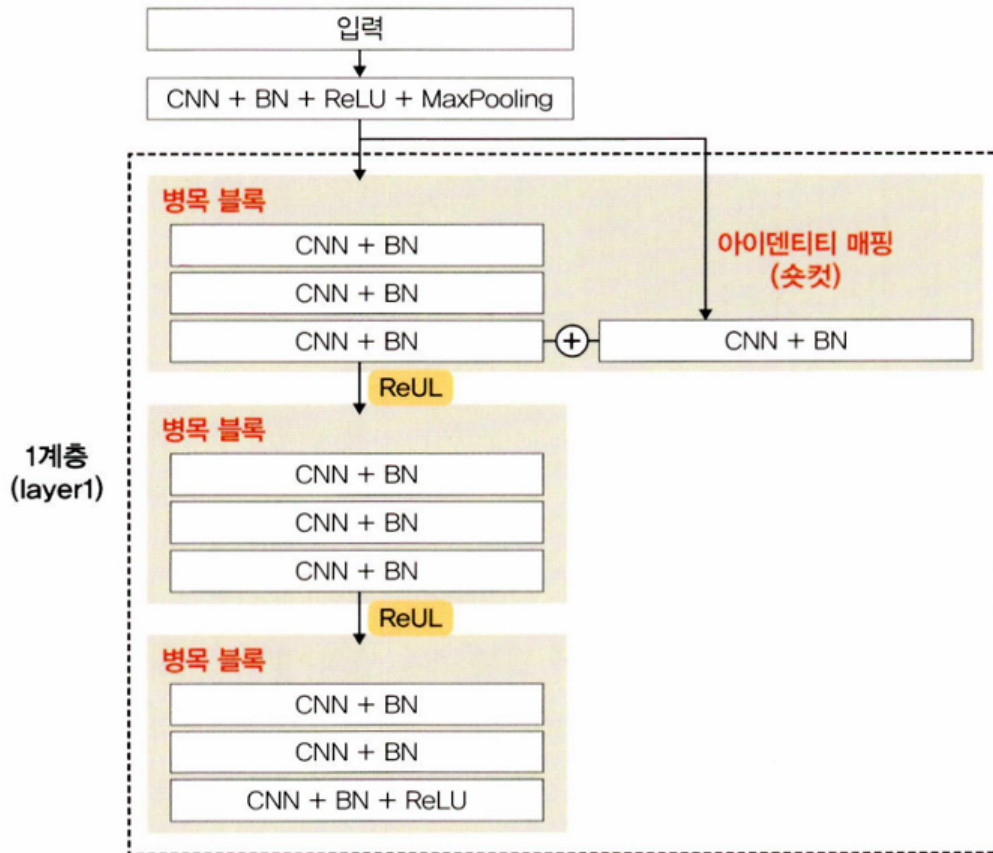
- **파라미터 수의 문제** : 계층이 깊어질수록 파라미터는 무제한으로 증가. 이러한 문제를 해결하기 위해 병목 블록을 두었다.

▼ 그림 6-27 기본 블록과 병목 블록



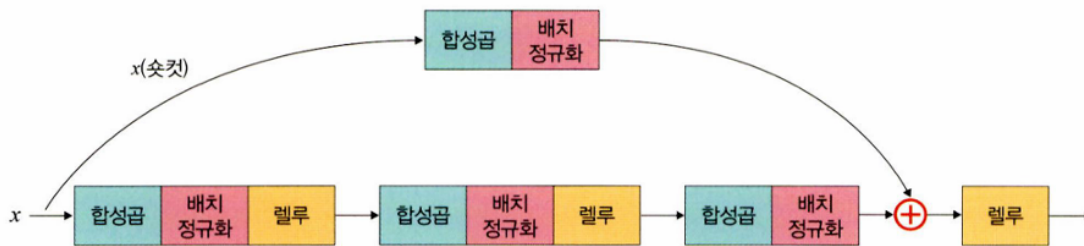
- 병목 블록을 두면 깊이가 깊어져도 파라미터의 수가 적어지는 현상을 관찰할 수 있다. 파라미터 수의 문제 해결 가능.
- ResNet34와는 다르게 ResNet50에서는  $3 \times 3$  합성곱층 앞뒤로  $1 \times 1$  합성곱층이 붙어 있는데,  $1 \times 1$  합성곱층의 채널 수를 조절하면서 차원을 줄였다 늘리는 것이 가능 → 파라미터 줄이기 가능

▼ 그림 6-28 아이덴티티 매핑(숏컷)



- **아이덴티티 매핑** : 위 그림의 + 부분. 입력  $x$ 가 어떤 함수를 통과하더라도 다시  $x$ 라는 형태로 출력되도록 만든다.
- **다운샘플(downsample)** : 특성 맵(feature map) 크기를 줄이기 위한 것으로 풀링과 같은 역할을 한다.
- 형태가 다른데 이들 간의 형태를 맞추지 않으면 아이덴티티 매핑을 할 수 없게 된다. 따라서 아이덴티티에 대해 다운샘플이 필요하다.
- **아이덴티티 블록** : 입력과 출력의 차원이 같은 것
- **프로젝션 숏컷, 합성곱 블록** : 입력 및 출력 차원이 동일하지 않고 입력의 차원을 출력에 맞추어 변경해야 하는 것

♥ 그림 6-30 합성곱 블록



- ResNet은 기본적으로 VGG19 구조를 뼈대로 하며 거기에 합성곱층들을 추가해 깊게 만든 후 숏컷들을 추가한 것

## 파이토치 구현 시 주의 사항

- **네임드 튜플(Named tuple)** : 파이썬의 자료형 중 하나. 말 그대로 튜플의 성질을 갖고 있는 자료형이지만 인덱스 뿐만 아니라 키값으로 데이터에 접근 가능
- **다운샘플**은 입력 데이터의 크기와 네트워크를 통과한 후 출력 데이터의 크기가 다를 경우 사용됨. 사용할 때는 합성곱 층에 스트라이드를 적용
- **아이덴티티 매핑** → 특정 층에 존재하는 출력 결과를 다음 합성곱층을 통과한 출력 결과에 더해준다. 스킵 연결이라고도 부름.
- **병목 블록** → ResNet50, ResNet101, ResNet152에서 사용되며 1x1 합성곱층, 3x3 합성곱층, 1x1 합성곱층으로 구성됨.
  - **기본 블록을 병목 블록으로 변경하는 이유** : 계층을 더 깊게 쌓으면서 계산에 대한 비용을 줄일 수 있음. 계층이 많아진다는 것은 활성화 함수가 기존보다 더 많이 포함된다는 것을 의미하며 이것은 더 많은 비선형성을 처리할 수 있음을 의미. 다양한 입력 데이터에 대한 처리가 가능.
- 각 레지듀얼 분기에 있는 마지막 BN을 0으로 초기화 → 다음 레지듀얼 분기를 0에서 시작할 수 있도록 한다.
- `tensor.topk` → `torch.argmax`와 같은 효과. 주어진 텐서에서 가장 큰 값의 인덱스를 얻기 위해 사용. 네트워크의 출력에서 가장 확률이 높은 값의 인덱스를 반환.

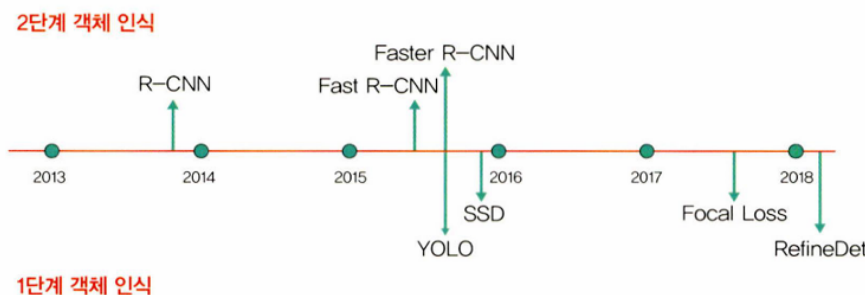
## 6.2 객체 인식을 위한 신경망

- 객체 인식이란 이미지나 영상 내에 있는 객체를 식별하는 컴퓨터 비전 기술
- 객체 인식이란 이미지나 영상 내에 있는 여러 객체에 대해 각 객체가 무엇인지 분류하는 문제와 그 객체 위치가 어디인지 박스로 나타내는 위치 검출 문제를 다루는 분야

객체 인식 = 여러 가지 객체에 대한 분류 + 객체의 위치 정보를 파악하는 위치 검출

딥러닝을 이용한 객체 인식 알고리즘은 크게 1단계 객체 인식(1-stage detector)과 2단계 객체 인식(2-stage detector)으로 나눌 수 있습니다.

▼ 그림 6-35 1단계 객체 인식 vs 2단계 객체 인식 흐름도



- 1단계 객체 인식 : 분류와 위치 검출을 동시에 행하는 방법. 비교적 빠름, 하지만 정확도가 낮음(ex YOLO 계열, SSD 계열)
- 2단계 객체 인식 : 두 문제를 순차적으로 행하는 방법. 비교적 느림, 하지만 정확도가 높음 (ex R-CNN 계열)

### R-CNN

- 이미지 분류를 수행하는 CNN과 이미지에서 객체가 있을 만한 영역을 제안해주는 후보 영역 알고리즘을 결합한 알고리즘.
- 3 단계의 복잡한 학습 과정, 긴 학습시간과 대용량의 저장 공간, 객체 검출 속도 문제로 인해 크게 발전하지는 못함.

### 공간 피라미드 풀링

- 기존 CNN 구조들은 모두 완전연결층을 위해 입력 이미지를 고정해야 했다. 그렇기 때문에 신경망을 통과시키려면 이미지를 고정된 크기로 자르거나 비율을 조정해야 함. → 그러나 물체의 일부분이 잘리거나 본래의 생김새와 달라지는 문제점.
  - 공간 피라미드 풀링을 도입.
    - 입력 이미지의 크기에 상관없이 합성곱층을 통과시키고 완전 연결층에 전달되기 전에 특성 맵들을 동일한 크기로 조절해주는 풀링층을 적용하는 기법
    - 입력 이미지의 크기를 조절하지 않고 합성곱층을 통과시키기 때문에 원본 이미지의 특징이 훼손되지 않는 특성 맵을 얻을 수 있음. 또한, 이미지 분류나 객체 인식 같은 여러 작업에 적용할 수 있음.

## Fast R-CNN

- R-CNN은 바운딩 박스마다 CNN을 돌리고 분류를 위한 긴 학습 시간이 문제였다.
- 속도 문제를 개선하려고 RoI 풀링을 도입.
- 선택적 탐색에서 찾은 바운딩 박스 정보가 CNN을 통과하면서 유지되도록 하고 최종 CNN 특성 맵은 풀링을 적용하여 완전 연결층을 통과하도록 크기 조정 → 바운딩 박스마다 CNN을 돌리는 시간 단축
  - RoI 풀링 : 크기가 다른 특성 맵의 영역마다 스트라이드를 다르게 최대 풀링을 적용하여 결괏값 크기를 동일하게 맞추는 방법

## Faster R-CNN

- 더욱 빠른 객체 인식을 수행하기 위한 네트워크
- 기존 Fast R-CNN 속도의 걸림돌이었던 후보 영역 생성을 CNN 내부 네트워크에서 진행할 수 있도록 했다. 외부의 느린 선택적 탐색(CPU) 대신 내부의 빠른 RPN(GPU) 사용
- RPN은 마지막 합성곱층 다음에 위치, 그 뒤에 RoI 풀링과 분류기, 바운딩 박스 회귀가 위치

# 6.3 이미지 분할을 위한 신경망

- 이미지 분할은 신경망을 훈련시켜 이미지를 픽셀 단위로 분할하는 것 → 이미지에 포함된 객체 추출

## 완전 합성곱 네트워크

- 고정된 크기의 입력만 받아들이며, 완전연결층을 거친 후에는 위치 정보가 사라진다는 한계를 극복. → 완전 연결층을 1X1 합성곱으로 대체.
- 이미지 분류에서 우수한 성능을 보인 CNN 기반 모델을 변형시켜 이미지 분할에 적합하도록 만든 네트워크
- 단점 : 여러 단계의 합성곱층과 풀링층을 거치면서 해상도가 낮아진다. 낮아진 해상도를 복원하기 위해 업 샘플링 방식을 사용하기 때문에 이미지의 세부 정보들을 잃어버리는 문제 발생.

## 합성곱 & 역합성곱 네트워크

- 완전 합성곱 네트워크의 단점을 해결하기 위해 역합성곱 네트워크를 도입한 것
- CNN의 최종 출력 결과를 원래의 입력 이미지와 같은 크기로 만들고 싶을 때 사용
- 시멘틱 분할 등에 활용 가능, 업 샘플링이라고도 함.
- CNN에서 합성곱층은 합성곱을 사용하여 특성 맵 크기를 줄인다. 하지만 역합성곱층은 이와 반대로 특성 맵 크기를 증가시키는 방식으로 동작.

## U-Net

- 바이오 메디컬 이미지 분할을 위한 합성곱 신경망
- 속도가 빠르다. 검증이 끝난 패치는 건너뛰기 때문에.
- 트레이드 오프에 빠지지 않는다.
- FCN 기반으로 구축되었으며 수축 경로와 확장 경로로 구성되어 있다. 수축 경로는 컨텍스트를 포착하며 확장 경로는 특성 맵을 업 샘플링하고 수축 경로에서 포착한 특성 맵의 컨텍스트와 결합하여 정확한 지역화를 수행.
- 크기가 다양한 이미지의 객체를 분할하기 위해 크기가 다양한 특성 맵을 병합할 수 있도록 다운 샘플링과 업 샘플링을 순서대로 반복하는 구조.



## PSPNet

- 완전연결층의 한계를 극복하기 위해 피라미드 풀링 모듈을 추가

## DeepLabv3/DeepLabv3+

- 완전연결층의 단점을 보완하기 위해 Atrous 합성곱을 사용하는 네트워크
- 인코더와 디코더 구조를 가지며, 일반적인 인코더-디코더 구조에서는 불가능했던 인코더에서 추출된 특성 맵의 해상도를 Atrous 합성곱으로 제어할 수 있도록 했다.
- 이미지 분할에서 높은 성능을 내려면 수용 영역의 크기가 중요한데, 수용 영역을 확대하여 특성을 갖는 범위를 넓게 한다. Atrous 합성곱을 활용하면 파라미터의 수를 늘리지 않으면서도 수용 영역을 크게 키울 수 있기 때문에 이미지 분할 분야에서 많이 사용.