딥러닝 4단계 : 합성곱 신경망 네트워크 (CNN)

2. 케이스 스터디

<왜 케이스 스터디를 하나요?>

- 하나의 컴퓨터 비전 작업에서 잘 작동한 구조가 다른 작업에도 유용하게 작동

1. Classic networks : LeNet-5, AlexNet, VGG

2. ResNet : 신경망의 깊이가 깊어질수록 ResNet 신경망은 152개의 층을 훈련시킨다

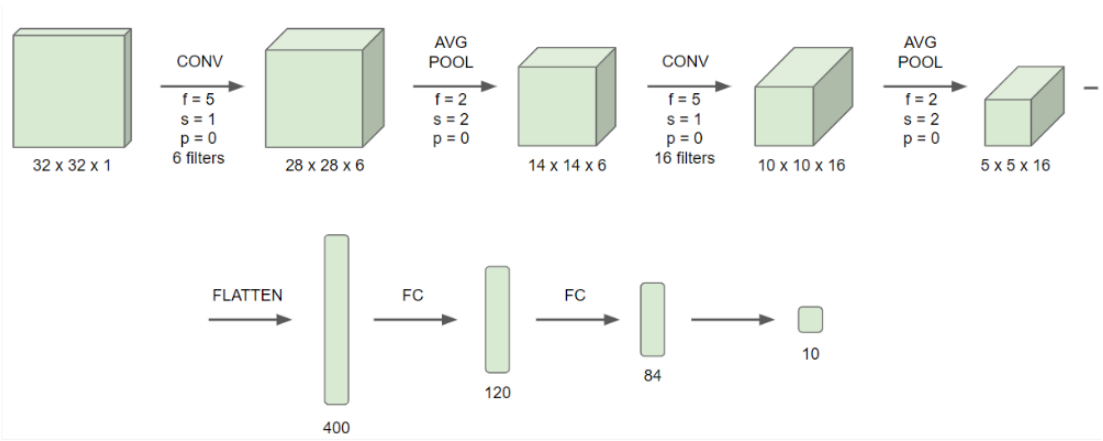
3. Inception

<고전적인 네트워크>

LeNet-5

목적 : 흑백의 손글씨를 인식

32x32x1



신경망이 깊어질수록 채널 수가 증가한다

CONV => POOL => CONV => POOL => FC FC => Output

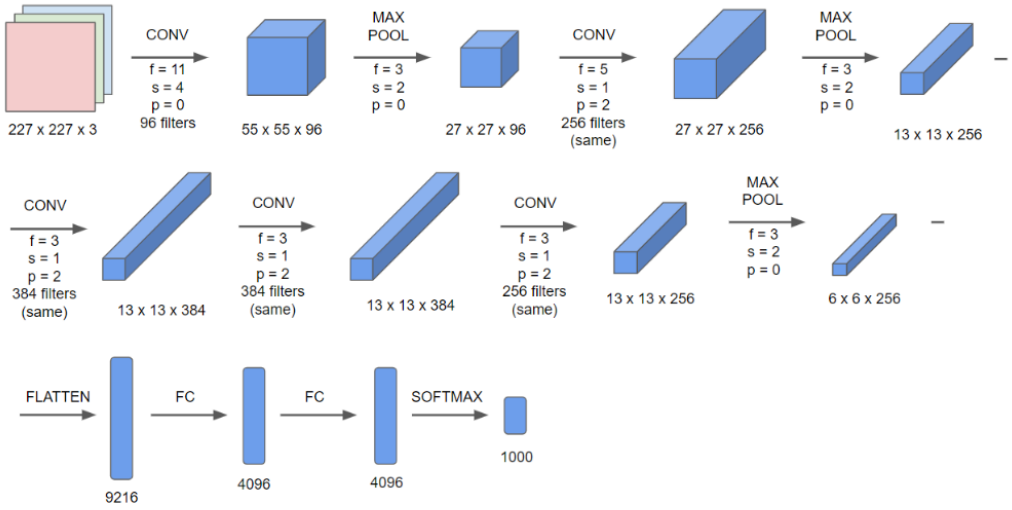
cf)

1. 당시 사람들은 시그모이드/tanh 를 사용했는데, nH x nW x nC의 신경망에서 f x f x nC 필터를 사용하면 변수처럼 계산을 줄이기 위해 각각의 필터가 서로 다른 채널에 적용되었다.

2. 기존의 LeNet-5는 비선형성이 풀링 층 뒤에 적용했다.

AlexNet

227x227x3



- LeNet과 유사하지만 훨씬 큰 크기를 가진다
- LeNet은 6만개의 매개변수, AlexNet은 6천만개 정도의 매개변수를 가진다

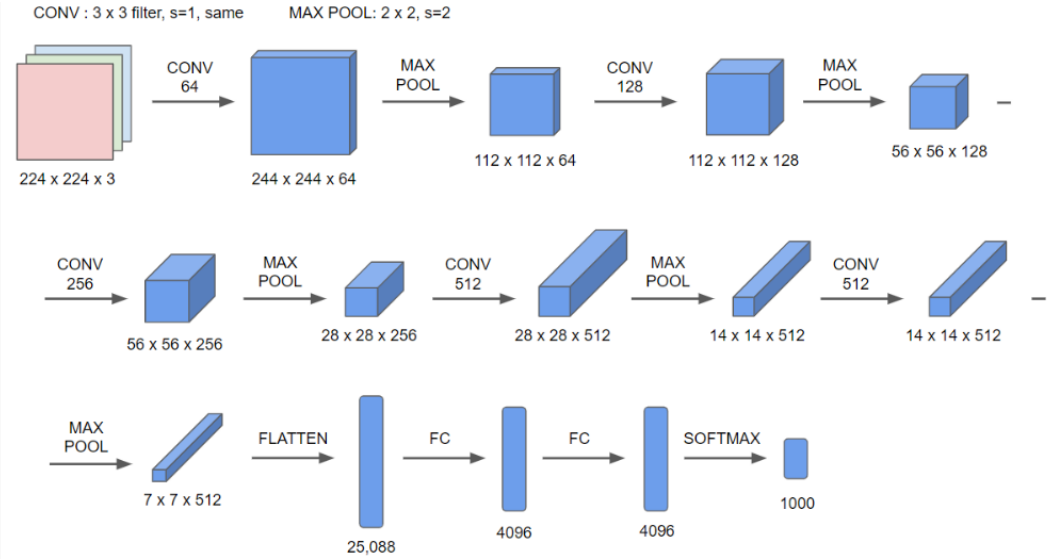
<LeNet과 구별되는 점>

- 더 많은 은닉 유닛과 더 많은 데이터로 훈련하기 때문에 훨씬 더 뛰어난 성능을 가진다
- ReLu 활성화 함수를 사용한다

- cf) 이러한 층들이 두개의 구별된 GPU에 나누어져서 두개의 GPU가 서로 소통하는 방식
- 기존 구조에서는 지역 응답 정규화라는 층이 있다
  - ex) 13x13x256 에서 LRN은 높이와 너비가 지정된 한 지점의 모든 채널을 보고 정규화한다
  - 각 위치에서 높은 활성값을 가진 뉴런을 너무 많이 원치 않기 때문 but 그리 유용하지 않음

VGG-16 네트워크

- 많은 하이퍼 파라미터를 가지는 대신 합성곱에서 s=1인 3x3 필터만을 사용해 동일 합성곱을 하고 최대 풀링층에서는 s=2인 2x2
- 간결한 구조를 가진다

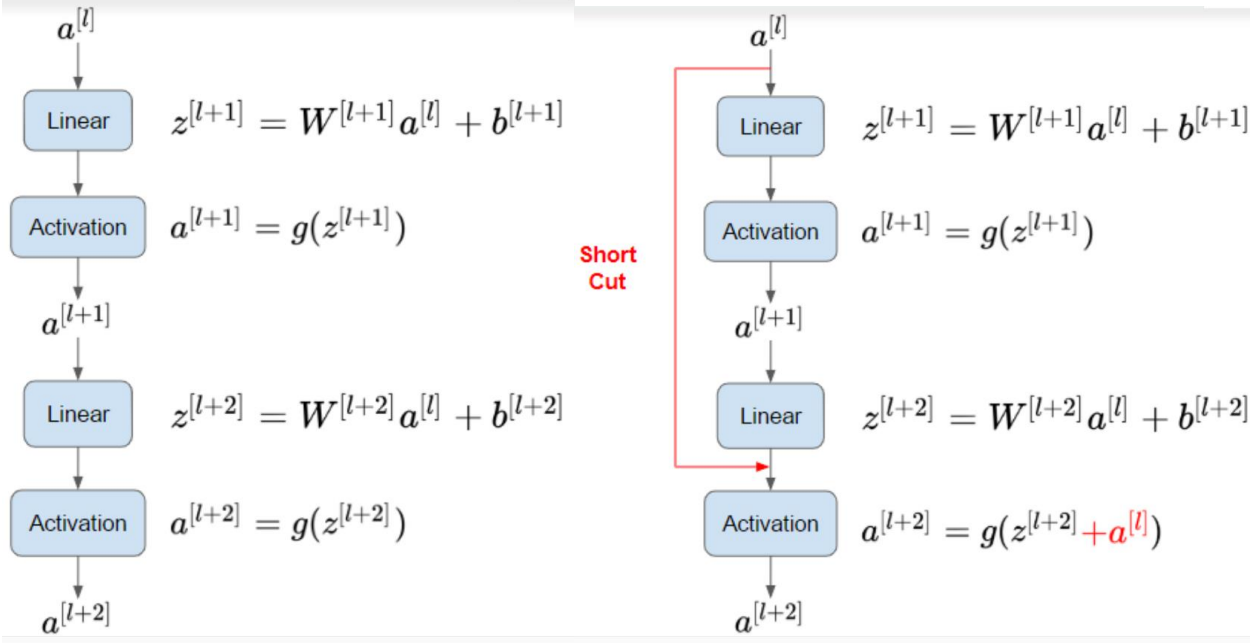


- 16개의 가중치를 가진 층이 있다
- 큰 네트워크
- 균일하다 : CON=> POOL (reduce width) => CONV (double) ...
- 훈련시킬 변수의 개수가 많아 네트워크의 크기가 커진다
- VGG-19은 더 큰 버전
- 깊이가 깊어질수록 보여지는 패턴이 풀링층에서는 높이와 너비가 매번 절반으로 줄어들고, 합성곱층에서는 채널의 수가 두배 가량 늘어난다 (체계적)

<Residual networks>

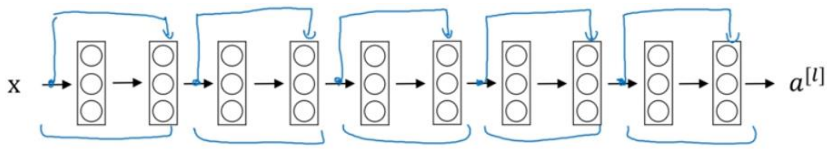
깊은 신경망의 훈련이 어려운 이유 : 경사 소실, 경사 폭발 증가 문제

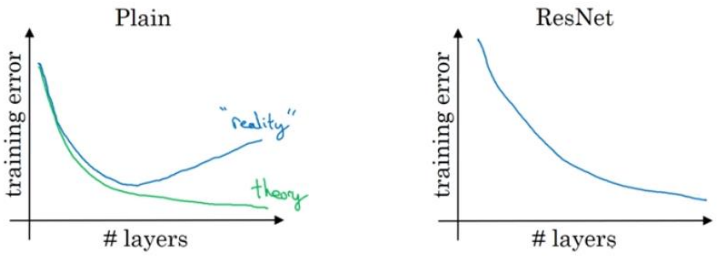
⇨ **스킵 연결** : 한 층의 활성값을 가지고 훨씬 깊은 층에 적용 => **Resnets**

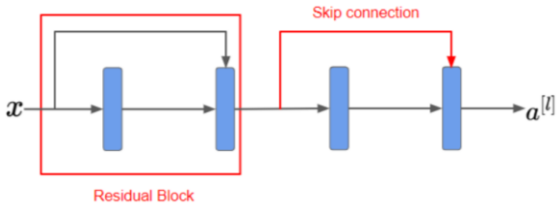


- 잔여블록으로 구성
- a[l] -> (선형 연산 후 relu 비선형성 적용) a[l+1] -> (선형 연산 후 relu 비선형성 적용) a[l+2]

  : main path
- a[l]을 복제해서 신경망의 먼 곳까지 단번에 가도록 하는 short cut (스킵 연결)
- a[l+2]=g(z[l+2]+a[l]) 에서 a[l]이 잔여 블록
- relu전에 더해지기 때문에 두번째 층으로 간다
- **a[l]은 선형 연산 뒤, relu 전에 삽입**
- 잔여 블록을 사용하면 훨씬 깊은 신경망을 훈련시킬 수 있다







- 평형망=>ResNet으로 바꾸려면 스킵 연결을 더해주면 된다
- 두 층 마다 잔여블록으로 만들어주기 위한 변화

⇨ 5개의 잔여 블록이 합쳐짐 => ResNet

- 표준 최적화 알고리즘을 사용하여 평형망을 훈련시키고 층의 개수를 늘리면 훈련 오류는 감소하다가 다시 증가한다

실제로는 평형망의 깊이가 매우 깊다면 최적화 알고리즘으로 훈련하는 것이 어려워지고 훈련 오류는 많아진다

★ ResNet에서는 층이 깊어져도 훈련 오류가 계속 감소한다!

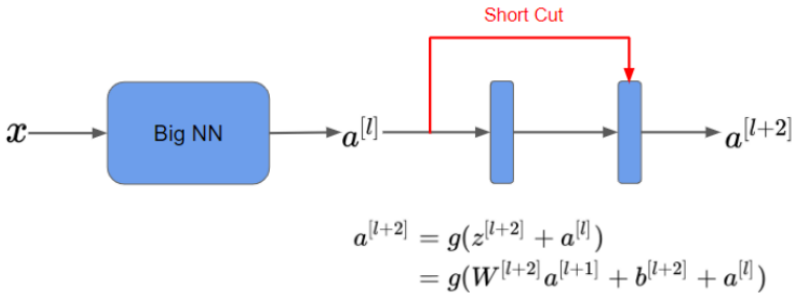
활성값 x나 중간활성값을 취하는 것으로 깊은 신경망을 사용할 수 있게 함

⇨ 경사 소실 문제 해결, 깊은 신경망 훈련 가능

<왜 ResNets이 잘 작동할까요?>

네트워크가 깊어질수록 훈련세트를 다루는 데에 지장을 줄 수 있어서 깊은 네트워크를 선호하지 않는다

⇨ But ResNet은 예외!



- **short cut** : 지름길을 추가해 잔차 블록으로 만들어준다
- relu 를 쓰기 때문에 X 제외한 모든 활성값은 0보다 크거나 같다
- **a[l+2]=g(z[l+2]+a[l])=g(w[l+2]a[l+1]+b[l+2]+a[l])** : a[l]은 이전에 추가한 스킵 연결
- w[l+2]=0 , b=0 이라면 g(a[l])=a[l] 이 된다 (Relu는 모든 활성값이 양수이기 때문)
- 항등 함수는 잔여 블록의 훈련을 용이하게 만들어준다

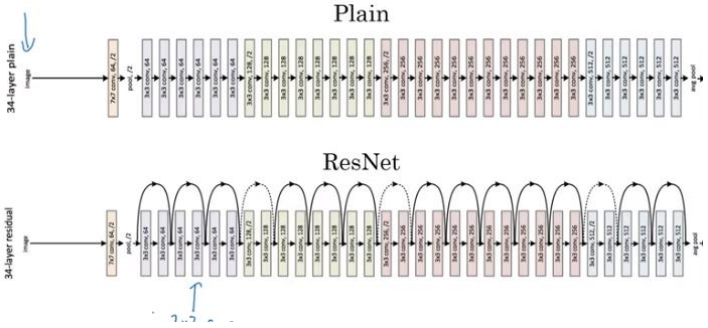
  a[l+2]=a[l] => 스킵 연결 때문에 같다

⇨ 잔차 블록을 신경망에 추가해도 성능에 지장이 없는 이유!

- 은닉 유닛들이 학습을 할 수 있다면 항등 함수를 학습하는 것보다 성능이 향상된다
- 스킵 연결이 없다면 항등함수를 학습하는 것이더라도 깊은 네트워크에서 변수를 선택하기 어려워진다
- 추가된 층이 항등 함수를 학습하기 용이하기 때문에 ResNet이 성능이 좋다!
-

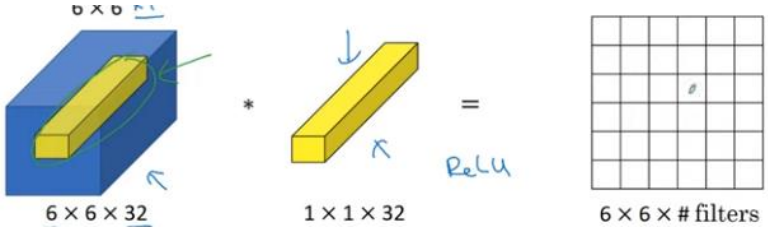
★ z[l+2]랑 a[l]은 같은 치원을 가져야 한다

- 입력(128 차원)과 출력(256 차원)의 차원이 다른 경우에는 W_s라는 행렬(256x128) 을 추가하여
- **W_s : 학습된 변수를 가진 행렬 or 제로 패딩으로 고정값을 가진 행렬**



⇨ 동일 합성곱이라 차원이 유지된다

<Network 속의 Network>



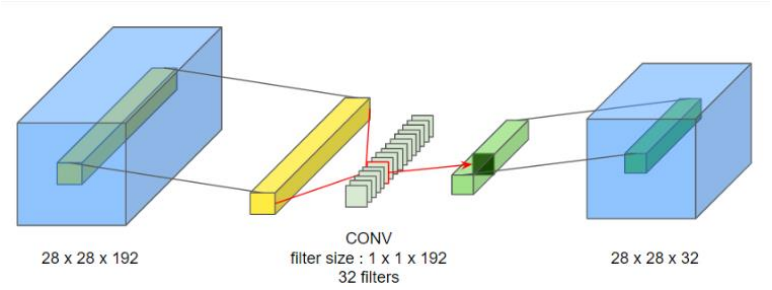
- 36개의 위치를 각각 살펴보고 32개의 숫자를 필터의 32개의 숫자와 곱하고 ReLu 비선형성을 적용

  32개의 숫자를 입력받고 32개의 숫자를 각각 같은 높이와 너비에 해당하는 채널에 각각 곱해주고, relu 비선형성을 적용한 값을 출력

  ★ 네트워크 속의 네트워크

⇨ 완전연결 신경망을 36개의 위치에 각각 적용해서 32개의 숫자를 입력값으로 받고 필터의 수만큼 출력한다

n_c[l+1]

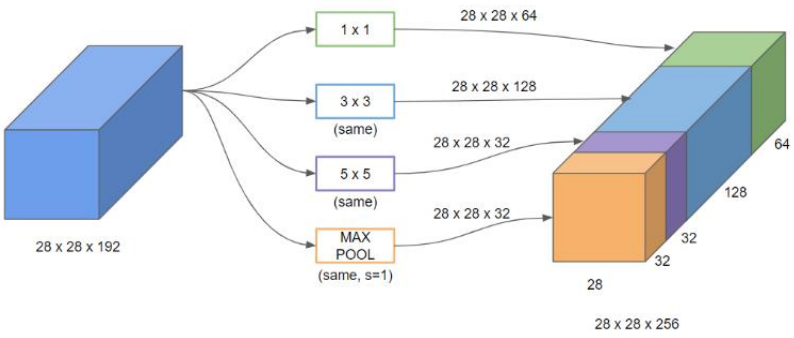


★ 높이와 너비를 줄이려면 풀링 층을 사용하면 된다 but **채널의 수**가 너무 많아서 줄이고 싶다면?

⇨ **32개의 1x1필터를 사용!**

- 각 필터가 1x1x192 (입력 채널의 수와 같아야한다
- 출력은 28x28x32

<inception 네트워크의 아이디어>



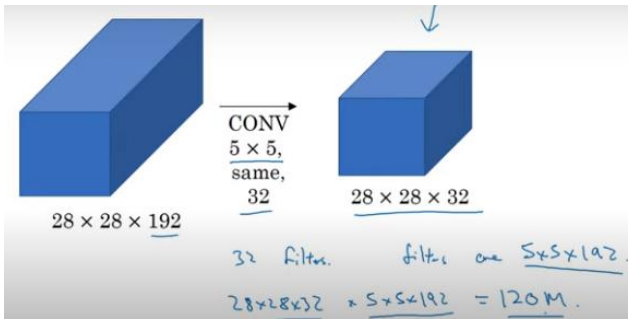
인셉션 네트워크 : 필터의 크기를 정하지 않고 합성곱 또는 풀링층을 모두 사용하는 것

- 동일 합성곱을 사용해서 출력이 크기가 동일하게 28x28이 유지되도록 한다
- 최대 풀링에서 크기를 맞추려면 패딩을 사용해야한다
- 높이와 너비를 28x28로 하고 다른 출력들의 크기와 맞추려면 패딩과 스트라이드=1을 사용해야한다
- 출력은 이 숫자를 전부 더한 값이다
- 입력: 28x28x192 => 출력 : 28x28x256

★ 필터의 크기나 풀링을 결정하는 대신 출력을 다 엮어낸 뒤 네트워크가 스스로 변수나 필터 크기의 조합을 학습하도록 하는 것!

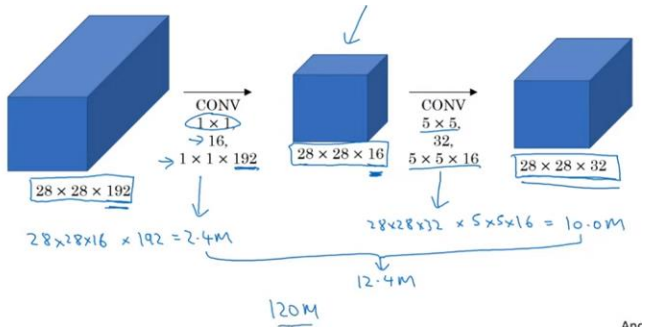
but 계산 비용이 문제

계산 : 각각의 출력값을 계산하기 위한 곱셈 + 출력값의 개수를 곱한 수 =(=1억2천만)



⇨ 1x1 합성곱을 이용하여 계산 비용을 줄일 수 있다

⇨



192개의 채널을 16개로 줄이고 5x5 합성곱을 하면 최종출력

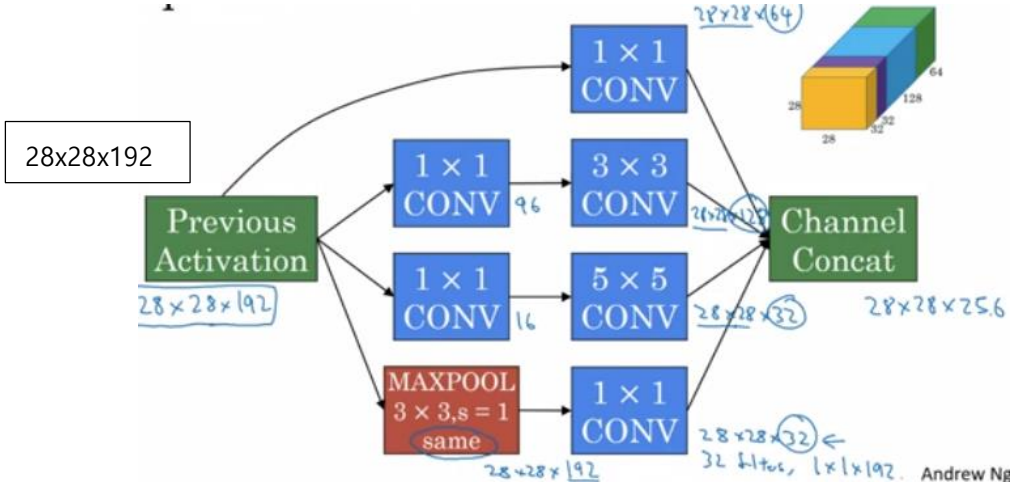
★ 병목 층 (bottleneck layer)

1x1x192 필터 필요

⇨ 비용(곱셉의 수) = 28x28x16x 1x1x192 + 28x28x32x5x5x16

<Inception 네트워크>

인셉션 모듈 : 활성값이나 이전 층의 출력을 입력값으로 받는다

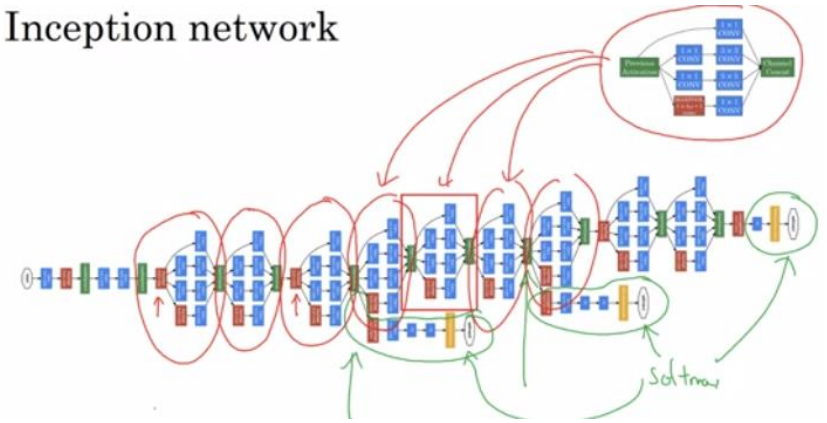


- 풀링에 동일한 패딩을 적용하여 높이와 너비를 같도록 한다

- 인셉션 네트워크는 이런 모듈을 하나로 모아놓은 것!
- 하나의 블록이 인셉션 모듈
- 차원을 바꾸기 위한 최대 풀링층 추가
- 네트워크의 마지막 몇 개의 층은 완전연결층, 뒤에는 예측을 위한 소프트맥스 층

(은닉층을 가지고 예측을 하는 일을 한다)

- 초록색 부분이 softmax : 결과 예측
- 은닉층이나 중간층에서 계산된 특성들도 이미지 결과를 예측하는데 나쁘지 않다
- 인셉션 네트워크에 정규화 효과를 주고 네트워크의 과대적합을 방지한다



- 이 인셉션 네트워크 : 구글넷
- 딥러닝 사회는 협력적
- 인셉션 네트워크라는 이름

- ResNet의 스킵 연결