딥러닝시스템

세종대학교 소프트웨어융합대학 지능기전공학부

고급 합성곱 신경망 구조

Advanced CNN Architectures

본 자료는 한빛미디어에서 나온 〈비전 시스템을 위한 딥러닝〉 책을
참고하여 제작 되었음을 알립니다.

5. 고급 합성곱 신경망 구조

- **학습 목표**
  - CNN 디자인 패턴 다루기
  - LeNet, AlexNet, VGGNet, 인셉션, ResNet 등의 신경망 구조 이해하기

- 학습 내용
  - 다섯가지 최신 합성곱 신경망 구조를 살펴본다.

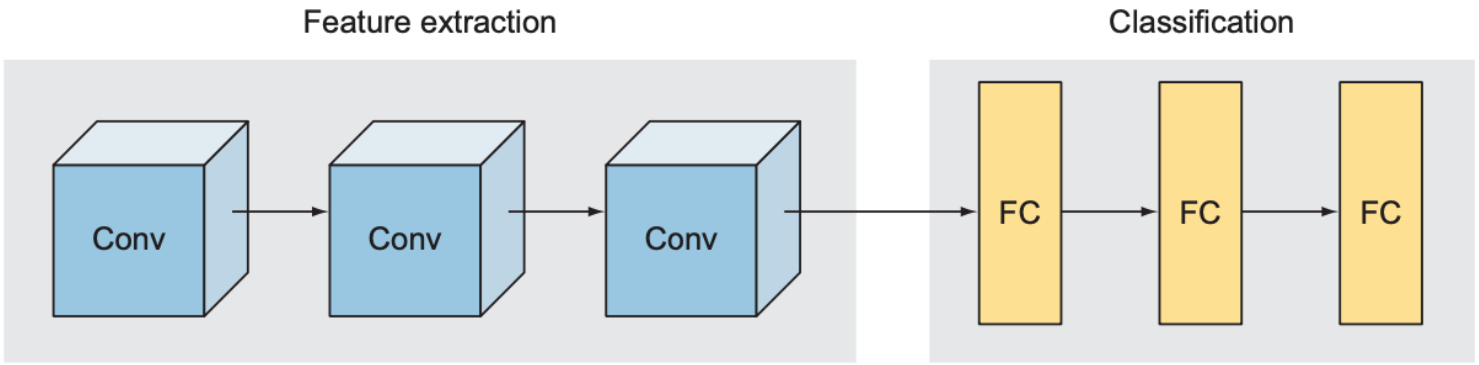
5. 고급 합성곱 신경망 구조

- **고급 합성곱 신경망 실습에 사용할 코드**
  - ⟨LeNet⟩
    https://www.kaggle.com/code/sukzoon1234/2024-1-dls-w5-lenet/notebook
  - ⟨AlexNet⟩
    https://www.kaggle.com/code/sukzoon1234/2024-1-dls-w5-alexnet/notebook?scriptVersionId=170429532
  - ⟨VGG⟩
    https://www.kaggle.com/code/sukzoon1234/2024-1-dls-w5-vgg/notebook?scriptVersionId=170465155
  - ⟨GoogLeNet⟩
    https://www.kaggle.com/code/sukzoon1234/2024-1-dls-w5-googlenet?scriptVersionId=170481069
  - ⟨ResNet⟩
    https://www.kaggle.com/code/sukzoon1234/2024-1-dls-w5-resnet?scriptVersionId=170486745

5. 고급 합성곱 신경망 구조

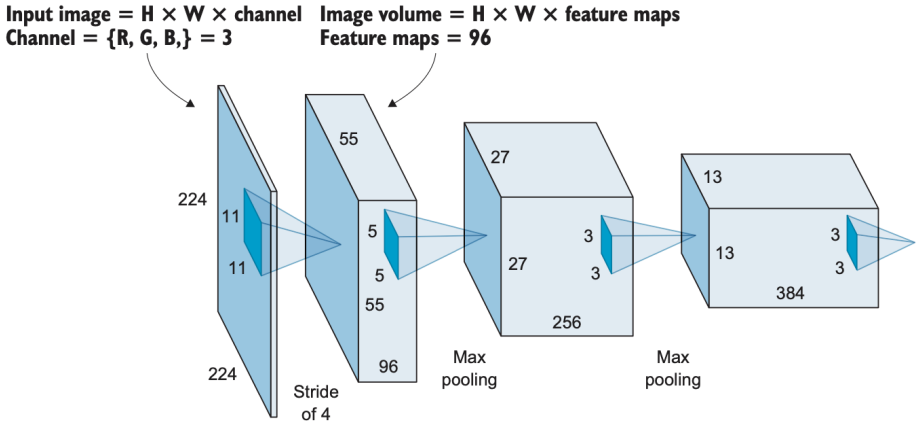
- **5.1 CNN의 디자인 패턴**
  - CNN을 이용한 딥러닝 모델 설계 시 사용하는 기존 패턴 구조

  - **첫번째 패턴 - 특징 추출과 분류**
    - 합성곱 신경망은 크게 특징 추출을 맡는 부분과 분류를 맡는 부분으로 나뉨
    - 특징 추출을 맡는 부분은 일련의 합성곱층, 분류를 맡는 부분은 전결합층으로 구성됨
    - LetNet, AlexNet, 인셉션, ResNet 거의 모든 합성곱 신경망이 이 구조를 따름



5. 고급 합성곱 신경망 구조

- 5.1 CNN의 디자인 패턴
  - **두번째 패턴 – 이미지 깊이는 증가, 크기는 감소**
    - 모든 층의 입력은 이미지
    - 각 층은 이전 층에서 생성된 새로운 이미지에 합성곱 연산을 적용함
    - 이미지는 높이, 폭, 깊이를 가진 3차원 대상이며, 깊이는 색상 채널color channel 이라고도 함
      - 입력층에서 깊이가 1이면 회색조 이미지, 3이면 컬러 이미지를 의미
      - 이후 계층에서 깊이는 색상 채널 대신 이전 층에서 추출된 특징을 나타내는 특징 맵이 됨
    - 합성곱층을 지날 때마다 이미지의 깊이가 증가하고 크기는 감소하는 경향은 모든 합성곱 신경망 공통으로 나타나는 특징임



5. 고급 합성곱 신경망 구조

- **5.1 CNN의 디자인 패턴**
  - **세번째 패턴 – 전결합층**
    - 대부분 모든 전결합층은 유닛 수가 같거나, 이어지는 유닛 수가 감소하는 패턴을 보임
    - 이어지는 층에서 유닛 수가 증가하는 경우는 거의 없음
    - 이어지는 모든 전결합층의 유닛을 같게 해서 신경망의 학습 능력이 저해되는 현상은 발생되지 않음

5. 고급 합성곱 신경망 구조

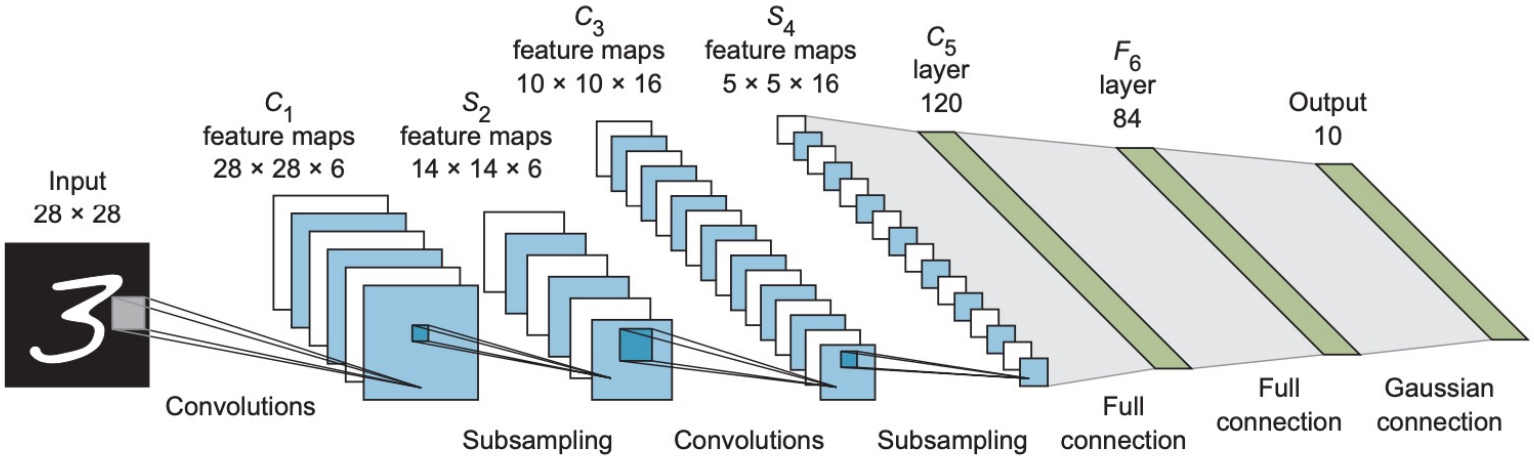
- 5.2 LeNet-5
  - 1998년 르쿤 연구진이 발표함
  - 직관적인 구조를 가졌음
  - LeNet-5는 가중치를 가진 5개의 층(3개의 합성곱층과 2개의 전결합층)으로 구성되었기 때문에 이러한 이름이 붙음

가중치를 가진 층

합성곱층과 전결합층을 통틀어 가중치를 가진 층이라고 부르며, 이 이름은 가중치가 없는 풀링층과 대비해 붙은 이름이다. 신경망의 깊이는 대개 가중치를 가진 층수를 기준으로 한다. 가중치를 가진 층수를 신경망의 깊이로 삼는 이유는 이 층수가 모델의 계산 복잡도와 직결되기 때문이다. 예를 들어 AlexNet은 5개의 합성곱층과 3개의 전결합층으로 구성되었으므로 층수가 8이다.

5. 고급 합성곱 신경망 구조

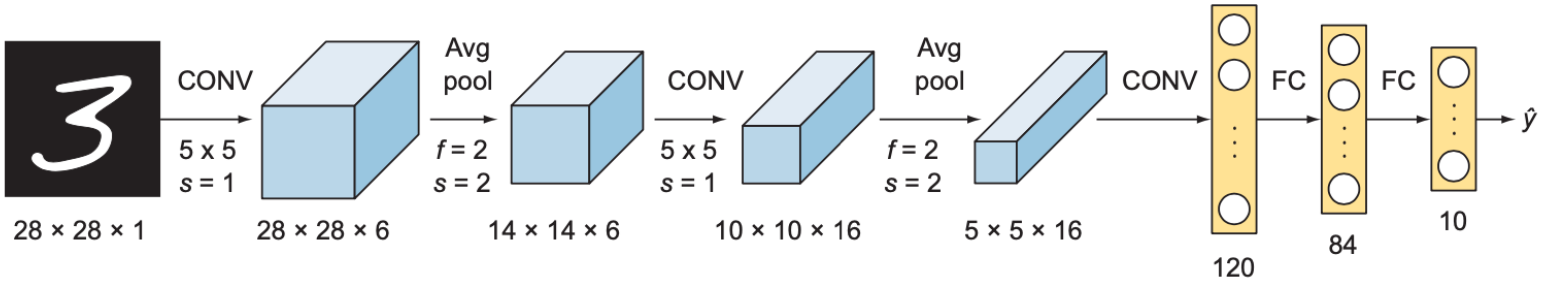
- **5.2.1 LeNet 구조**
  - 입력이미지=〉C1=〉TANH=〉S2=〉C3=〉TANH=〉S4=〉C5=〉TANH=〉FC6=〉SOFTMAX7
    - C는 합성곱층, S는 풀링층(서브샘플링층), FC는 전결합층



1998년에는 ReLU가 발견되기 이전이었고, tanh와 시그모이드 함수가 주로 사용되는 것이 일반적이었음

5. 고급 합성곱 신경망 구조

- **5.2.2 LeNet-5 구현하기**
  - 각 합성층의 필터 수
    - C1: 6, C3:16, C5:120
  - 각 합성층의 커널 크기
    - Kernel_size의 값을 5x5라고 언급
  - 풀링층(서브샘플링층)
    - 수용 영역의 크기는 2x2, 최대 풀링 대신 평균 풀링 사용
  - 활성화 함수
    - Tanh 함수를 사용, 시그모이드 함수에 비해 가중치가 더 빨리 수렴한다고 생각함



5. 고급 합성곱 신경망 구조

- **5.2.3 하이퍼파라미터 설정하기**
  - LeNet-5는 미리 설정된 일정엔 맞춰 학습률을 감소시키는 학습률 감쇠를 사용
    - 처음 2 에포크에는 0.005, 그 다음 3 에포크는 0.0002, 다시 그 다음 4 에포크에는 0.00005, 그 이후로는 0.00001의 학습률이 적용됨.
    - 논문의 실험에서는 20 에포크까지 학습을 수행

5. 고급 합성곱 신경망 구조

- **5.2.4 MNIST 데이터셋에 대한 LeNet의 성능**
  - MNIST 데이터셋을 대상으로 LeNet 신경망을 학습하면 99% 이상의 정확도를 확보가능함
  - 은닉층의 활성화 함수를 ReLU로 교체하면 어떤 결과가 나오는지 확인

5. 고급 합성곱 신경망 구조

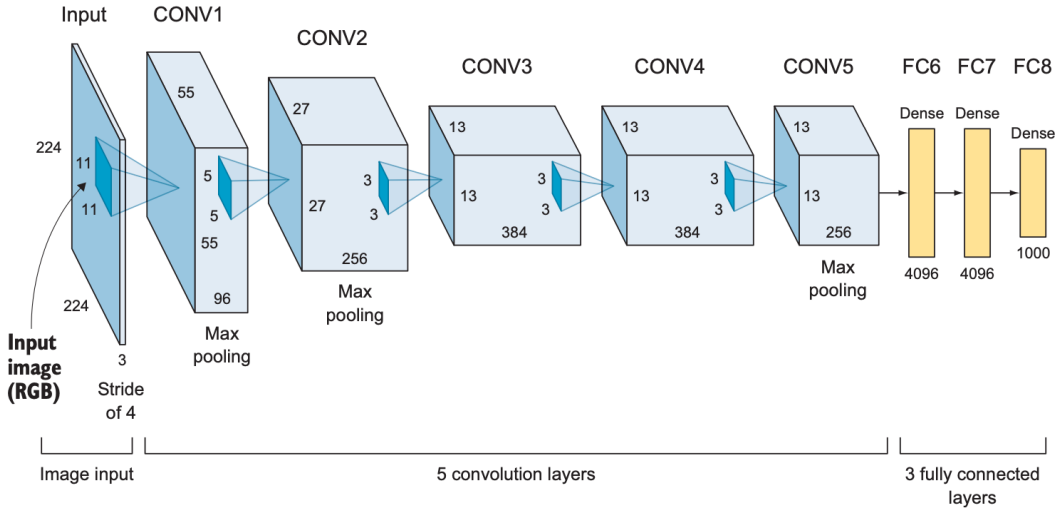
- **5.3 AlexNet**
  - LeNet이 MNIST에 대해 AlexNet 보다 높은 성능을 보임
  - AlexNet은 복잡도가 높은 이미지넷 문제 해결을 위해 제안된 신경망임
  - AlexNet은 2012년 ILSVRC 이미지 분류 콘테스트에서 우승을 차지함
  - 알렉스크리체프스키 연구진은 새로운 신경망 구조를 120만장의 이미지, 1000가지 클래스로 구성된 〈이미지넷 데이터셋〉으로 학습 시켰음
  - AlexNet 은 발표 당시 세계 최고의 성능을 자랑하며 컴퓨터비전분야에서 본격적인 딥러닝을 최초로 도입하여 합성곱 신경망의 응용이 확산되는 계기가 되었음

5. 고급 합성곱 신경망 구조

- **5.3 AlexNet**
  - AlexNet은 LeNet과 구조가 유사하지만 훨씬 층수가 많고, 규모가 큼(한 층당 필터 수가 많음)
  - 일력의 합성곱층과 풀링층의 조합이 이루어진 후 전결합층이 이어지다 소프트맥스 함수를 활성화 함수로 사용하는 출력층 기본 구조는 둘다 비슷
  - LeNet 신경망 구조는 6만여 개의 파라미터를 가졌음
  - AlexNet은 65만개 뉴런과 6천만개의 파라미터를 가졌음
  - AlexNet은 LeNet보다 훨씬 복잡한 특징을 학습할 수 있었고, 덕분에 2012년 ILSVRC 이미지 분류 콘테스트에서 두각을 나타내는 요인이 되었음

5. 고급 합성곱 신경망 구조

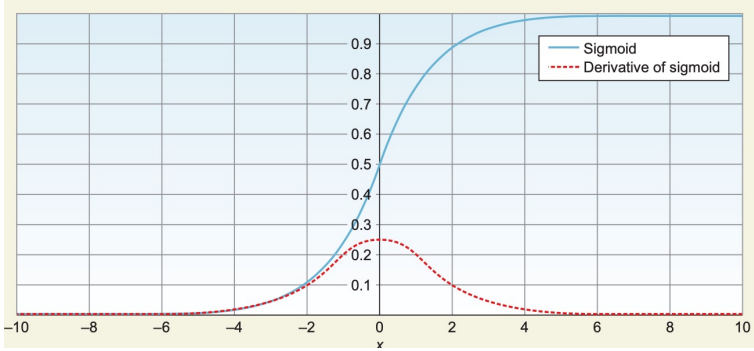
- 5.3.1 AlexNet 구조
  - 합성곱층의 필터 크기: 11x11, 5x5, 3x3
  - 최대 풀링 사용
  - 과적합 방지를 위한 드롭아웃 적용
  - 은닉층의 활성화 함수는 ReLU, 출력층의 활성화 함수는 소프트맥스 함수 사용
  - 입력이미지=〉Conv1=〉Pool2=〉Conv3=〉Pool4=〉Conv5=〉Conv6=〉Conv7=〉Pool8=〉FC9=〉FC10=〉SOFTMAX7



5. 고급 합성곱 신경망 구조

- **5.3.2 AlexNet 에서 발전된 부분**
  - AlexNet이 나오기 전에는 음성 인식 등 소수 분야에서만 딥러닝이 적용되고 있었음
  - AlexNet을 활용하여 이미지넷 대회에서의 우수한 성능을 보인 결과 컴퓨터 비전 분야의 많은 연구자들이 딥러닝을 새로운 돌파구로 생각하게 되었음

- **5.3.2.1 ReLU를 활성화 함수로 사용**
  - ReLU를 은닉층의 활성화 함수로 활용해서 학습 시간을 크게 단축 시켰음
  - 기울기 소실 문제vanishing gradient problem로 시그모이드/tanh 함수의 경우 빠른 학습에 어려움이 있었음



5. 고급 합성곱 신경망 구조

- **5.3.2.2 드롭아웃층**
  - 드롭아웃층은 신경망 모델의 과적합을 방지하기 위한 것
  - 드롭아웃을 통해 비활성화된 뉴런은 순방향 계산과 역전파 계산에서 모두 배제됨
  - 같은 입력을 가중치를 공유하는 매번 다른 구조의 신경망으로 학습하는 것과 같은 효과
  - 드롭아웃은 뉴런 간의 상호 적응을 방지하고 다양한 조합의 뉴런에 도움을 주는 유용한 특징을 학습하게 됨
  - AlexNet에는 두 전결합층에서 0.5의 드롭아웃 비율이 적용되었음

5. 고급 합성곱 신경망 구조

- **5.3.2.3 데이터 강화**
  - 레이블값을 변화시키지 않고 원 데이터만 변형하는 방식으로 데이터 양을 늘리는 기법도 과적합 방지에 효율적
  - 원 데이터를 변형하는 방법으로는 이미지 회전, 반전, 배율 조절 등이 있음

- **5.3.2.4 국소 응답 정규화**
  - AlexNet 에는 국소 응답 정규화 local response normalization 가 적용되어 있음
  - 국소 응답 정규화는 배치 정규화와 다른 기법임
  - 정규화는 가중치가 빨리 수렴되도록 하는 것이 목적으로 현재는 〈국소 응답 정규화〉 대신 〈배치 정규화〉가 많이 사용됨

5. 고급 합성곱 신경망 구조

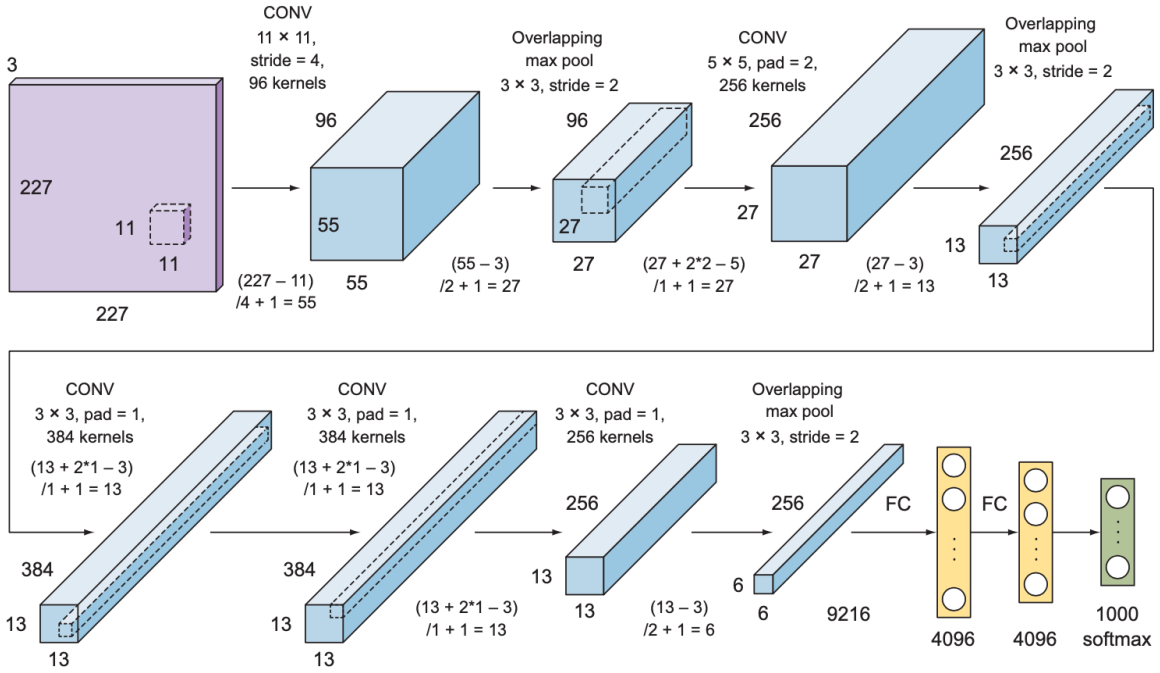
- **5.3.2.5 가중치 규제화**
  - AlexNet은 0.00005의 가중치 감쇠가 적용되었음
  - 가중치 규제화는 L2 규제화와 같은 개념

- **5.3.2.6 다중 GPU 사용**
  - AlexNet 연구진은 3GB 비디오램이 장착된 GTX 580을 사용하여 학습을 진행함
  - GTX 580은 당시 최신 제품이었지만 1200만 개나 되는 이미지넷 데이터 셋을 학습하기에는 성능이 충분하지 못했음
  - 이에 신경망을 2개의 GPU에 나눠 담아 학습하는 복잡한 방식(각 층을 두 GPU 메모리에 분리하고 이들 GPU가 서로 통신하는 것)을 개발했음
  - 오늘날에는 분산 GPU 환경에서 딥러닝 모델을 학습하는 기법이 매우 발전하였음

5. 고급 합성곱 신경망 구조

- **5.3.3 AlexNet 구현하기**
  - AlexNet 신경망 구조는 가중치가 있는 8개의 층으로 구성됨
  - 앞의 다섯층은 합성곱층이며, 이어지는 세 층은 전결합층임
  - 마지막 전결합층의 출력은 뉴런이 1000개인 소프트맥스층으로 이어지며, 1000개의 클래스에 대한 예측 확률을 출력함

입력영상크기
: 227x227x3



5. 고급 합성곱 신경망 구조

- **5.3.3 AlexNet 구현하기**
  - Conv1: 가장 큰 커널 크기(11)와 스트라이드 값(4)이 적용되어 출력 이미지가 입력 이미지 크기의 ¼ 로 줄어듬 (227x227➔55x55). 출력 깊이(필터수)는 96이므로 채널을 포함한 출력의 실제 크기는 55x55x96 이 됨

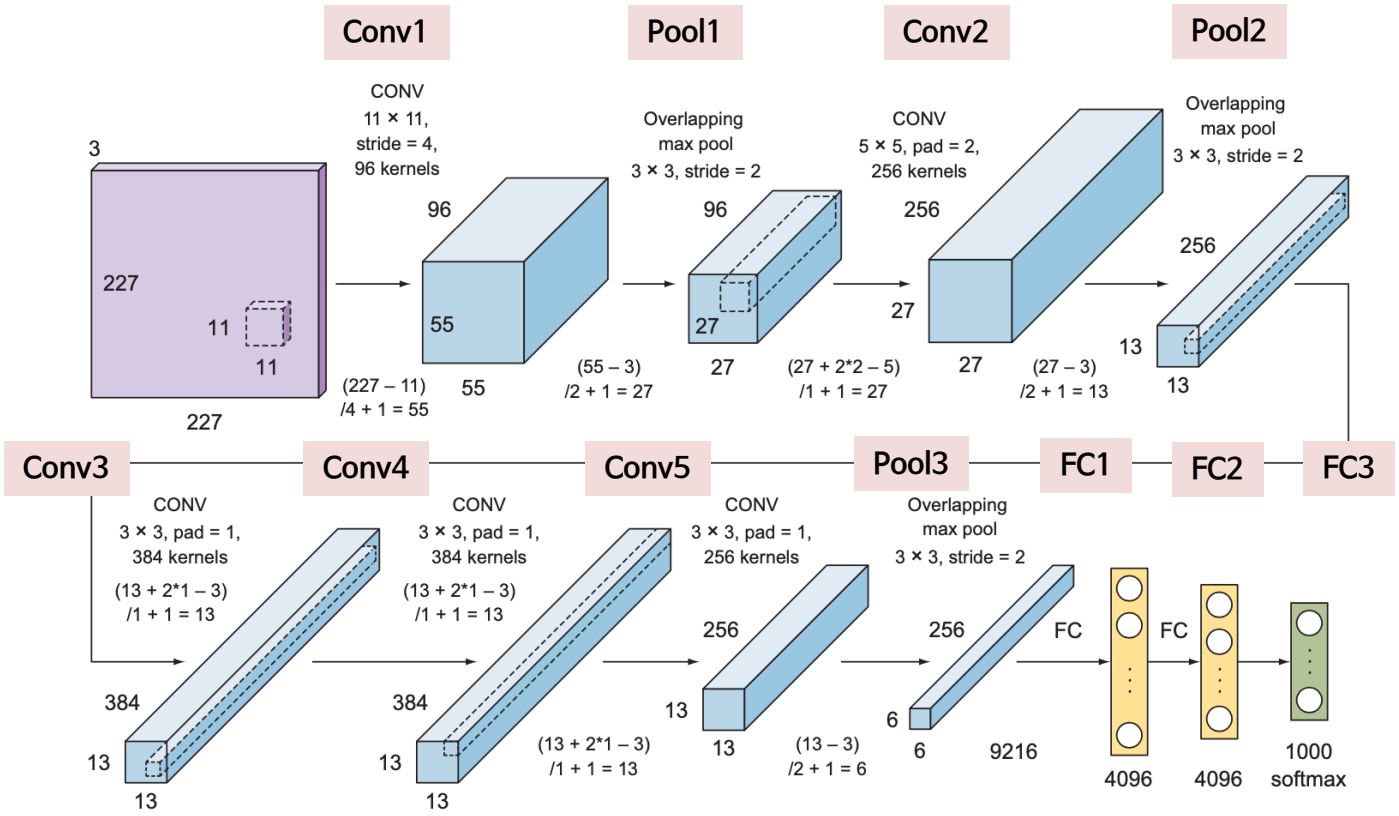
$$Conv\text{출력크기} = \frac{(\text{입력크기} - \text{커널크기})}{\text{스트라이드}} + 1 = \frac{(227 - 11)}{4} + 1 = 55$$

  - 필터 크기가 3x3인 Pool: 55x55의 입력이 27x27로 줄어듬. 풀링층은 이미지의 깊이에는 영향을 주지 않아 실제 크기는 27x27x96이 됨

$$Pool\text{출력크기} = \frac{(\text{입력크기} - \text{커널크기})}{\text{스트라이드}} + 1 = \frac{(55 - 3)}{2} + 1 = 27$$

5. 고급 합성곱 신경망 구조

- 5.3.3 AlexNet 구현하기



5. 고급 합성곱 신경망 구조

- **5.3.4 하이퍼파라미터 설정하기**
  - AlexNet은 90 에포크를 학습했으며, GTX 580 GPU를 동시에 사용해서 6일이 소요되었음
  - 초기에는 학습률을 0.01, 모멘텀은 0.9로 설정했으며, 검증 오차가 개선되지 않을 때마다 학습률을 이전의 1/10로 조정했음

5. 고급 합성곱 신경망 구조

- **5.3.5 AlexNet의 성능**
  - AlexNet은 2012 ILSVRC에서 크게 두각을 나타냈음
  - 15.3%의 Top-5오차율을 기록했음
  - 기존 분류기를 사용했던 2위 참가자의 성적과 26.2%의 차이를 보였음
  - 이 성과는 컴퓨터 비전 학계를 놀라게 했고, 이후 복잡한 시각 문제에 합성곱 신경망이 도입되고 이후 고급 합성곱 신경망 구조가 속속 개발되는 계기가 되었음

Top-1 오차율과 Top-5 오차율

Top-1 오차율과 Top-5 오차율은 알고리즘의 분류 성능을 나타내기 위한 개념임. Top-1 오차율은 분류기가 정답 클래스에 가장 높은 확률을 부여하지 않은 비율, Top-5 오차율은 정답이 예측 확률 상위 5개 안에 들어 있지 않은 비율을 가리킴.

5. 고급 합성곱 신경망 구조

- **5.4 VGGNet**
  - VGGNet은 2014년 옥스퍼드 대학교의 VGG 연구 그룹에서 제안한 신경망 구조
  - VGGNet의 구성 요소는 새로 고안된 요소 없이 LeNet이나 AlexNet과 동일하지만 신경망의 층수가 더 많음
  - VGG16 이라고 알려진 VGGNet은 가중치를 가진 층 16개로 구성 (합성곱층 13개, 전결합층 3개)
  - 모든 층의 하이퍼파라미터가 동일하게 설정되었기 때문에 신경망을 이해하기 쉽다는 점이 학계에 깊은 인상을 남김

5. 고급 합성곱 신경망 구조

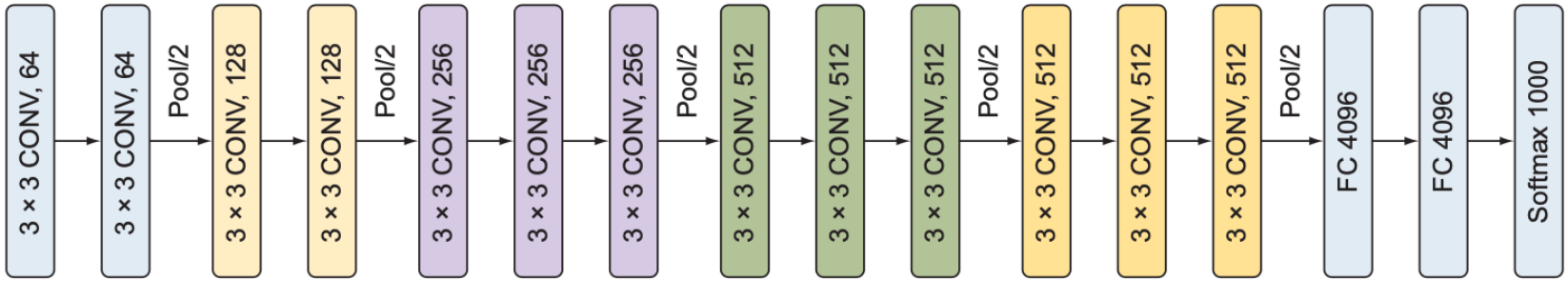
- **5.4.1 VGGNet 에서 발전된 부분**
  - VGGNet의 개선점은 동일하게 설정된 층(합성곱층과 전결합층)을 사용해서 신경망 구조를 단순화 시켰다는 점임
  - 전체적인 신경망 구조는 일련의 합성곱층 뒤에 역시 풀링층이 배치되는 구조로 모든 〈합성곱층은 3x3크기의 필터와 스트라이드1, 패딩1〉이 적용되었고, 모든 〈풀링층은 2x2크기의 풀링 영역과 스트라이드 2〉가 적용되었음
  - VGGNet에서 합성곱층의 필터 크기를 줄인(3x3) 이유는 AlexNet(11x11, 5x5)보다 더 세밀한 특징을 추출하기 위해서임
  - 수용영역의 크기가 같을 때 크기가 큰 하나의 커널보다 크기가 작은 커널을 여러개 쌓은 쪽이 더 성능이 좋음. 이는 커널을 여러개 쌓으며 비선형층을 늘리는 것이 신경망의 층수를 늘리는 것과 동일한 효과가 있기 때문임. 또한 파라미터 수를 억제하므로 더 낮은 비용으로 더 복잡한 특징을 학습할 수 있음.

5. 고급 합성곱 신경망 구조

- **5.4.1 VGGNet 에서 발전된 부분**
  - 커널을 여러 개 쌓으며 비선형층을 늘리는 것이 신경망의 층수를 늘리는 것과 동일
    - 커널의 크기가 3x3인 합성곱층을 두 층 쌓은 구조(합성곱층 사이에 풀링층은 두지 않음)는 5x5 크기의 수용 영역을 가진 합성곱층과 효과가 동등함
    - 커널의 크기가 3x3인 합성곱층을 세 층 쌓은 구조(합성곱층 사이에 풀링층은 두지 않음)는 7x7 크기의 수용 영역을 가진 합성곱층과 효과가 동등함
    - 따라서 수용 영역 크기가 3x3인 커널을 여러개 쌓으면 비선형 함수ReLU를 여러개 포함시키는 것과 같아 결정 함수의 변별력을 향상시키는 효과가 있음
  - 파라미터 수를 억제하는 효과
    - C개 채널을 가진 7x7 크기의 커널을 가진 단일 합성곱층의 파라미터 수는 $7^2C^2=49C^2$
    - 3x3 크기의 커널을 3개 쌓은 구조의 파라미터수는 $3*3^2C^2=27C^2$ 개로 절반 가까이로 억제하는 효과가 있음

5. 고급 합성곱 신경망 구조

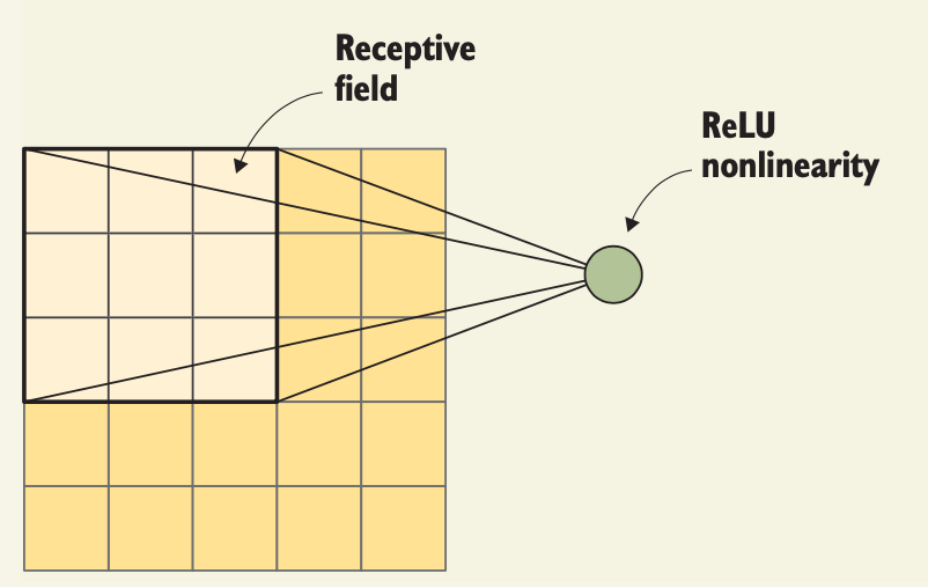
- **5.4.1 VGGNet 에서 발전된 부분**
  - VGGNet은 이렇게 합성곱 연산과 풀링 연산이 일어나는 구조를 통합하는 방법으로 전체 신경망의 구조를 단순하게 유지해서 신경망에 대한 이해 및 구현 편의성을 개선함
  - VGGNet은 3x3 크기의 커널을 여러 층 쌓은 합성곱층 사이에 간간이 2x2 크기의 풀링 연산을 하는 풀링층을 끼워 넣는 식으로 구성됨
  - 이 구조 뒤에 다시 일반적인 전결합층과 소프트맥스 함수가 배치되는 전형적인 분류기 구조가 배치됨



VGGNet-16 구조

5. 고급 합성곱 신경망 구조

- **5.4.1 VGGNet 에서 발전된 부분**
  - 수용 영역 (Receptive Field)
    - 수용영역은 출력의 한 점에 영향을 미치는 입력 이미지의 범위를 의미함



5. 고급 합성곱 신경망 구조

- **5.4.2 VGGNet 의 다양한 버전**
  - 구성 D와E가 가장 일반적으로 사용되며 가중치를 포함하는 층수를 붙여 이들을 각각 VGG16, VGG19라고 부름

ConvNet configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
Input (224 x 224 RGB image)					
conv3-64	conv3-64 LRN	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 conv3-256	conv3-256 conv3-256 conv3-256	conv3-256 conv3-256 conv3-256 conv3-256
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
FC-4096					
FC-4096					
FC-1000					

5. 고급 합성곱 신경망 구조

- **5.4.2 VGGNet 의 다양한 버전**
  - VGG16은 약 1억 3800만개의 파라미터가 있으며, VGG19의 파라미터 수는 약 1억 4400만 개임
  - VGG16이 VGG19보다 더 적은 파라미터로 거의 동등한 성능을 내기 때문에 일반적으로는 VGG16이 더 널리 쓰임

VGGNet 각 구성의 파라미터 수 (단위: 백만)

Network	A, A-LRN	B	C	D	E
No. of parameters	133	133	134	138	144

5. 고급 합성곱 신경망 구조

- **5.4.2.1 VGG16 구현하기**
  - VGGNet 중 VGG16과 VGG19가 층수가 많은 만큼 더 복잡한 모델을 학습할 수 있어 구현에 자주 사용됨
  - VGGNet 논문에서는 과적합을 방지하기 위해 다음 같은 규제화 기법을 적용했음
    - 가중치 감쇠율 $5x10^{-4}$ 을 적용한 L2 규제화
    - 편의를 위해 아래 구현 코드에는 포함하지 않았음
    - 비율이 0.5인 드롭아웃을 마지막 층을 제외한 두 전결합층에 적용했음

5. 고급 합성곱 신경망 구조

- **5.4.3 하이퍼파라미터 설정하기**
  - VGGNet의 학습 과정은 AlexNet의 학습 과정을 많이 참고했음
  - 미니배치 경사 하강법을 사용하되 모멘텀 0.9를 적용했음
  - 학습률은 초깃값 0.01로 시작해서 검증 데이터의 정확도가 정체될 때 마다 1/10 로 감소시킴

- **5.4.4 VGGNet의 성능**
  - VGG16의 이미지넷 데이터셋에 대한 Top-5오차는 8.1%로, 15.3%의 오차를 기록했던 AlexNet에 비해 개선된 성능을 보였음
  - VGG19의 성능은 이보다 더 높은 7.4%를 기록함
  - AlexNet보다 층수가 많고 파라미터 수가 더 많음에도 VGGNet 의 파라미터가 더욱 빨리 수렴했다는 것은 층수를 늘리고 합성곱 필터의 크기를 줄이면서 규제화와 같은 효과가 발생했기 때문임

5. 고급 합성곱 신경망 구조

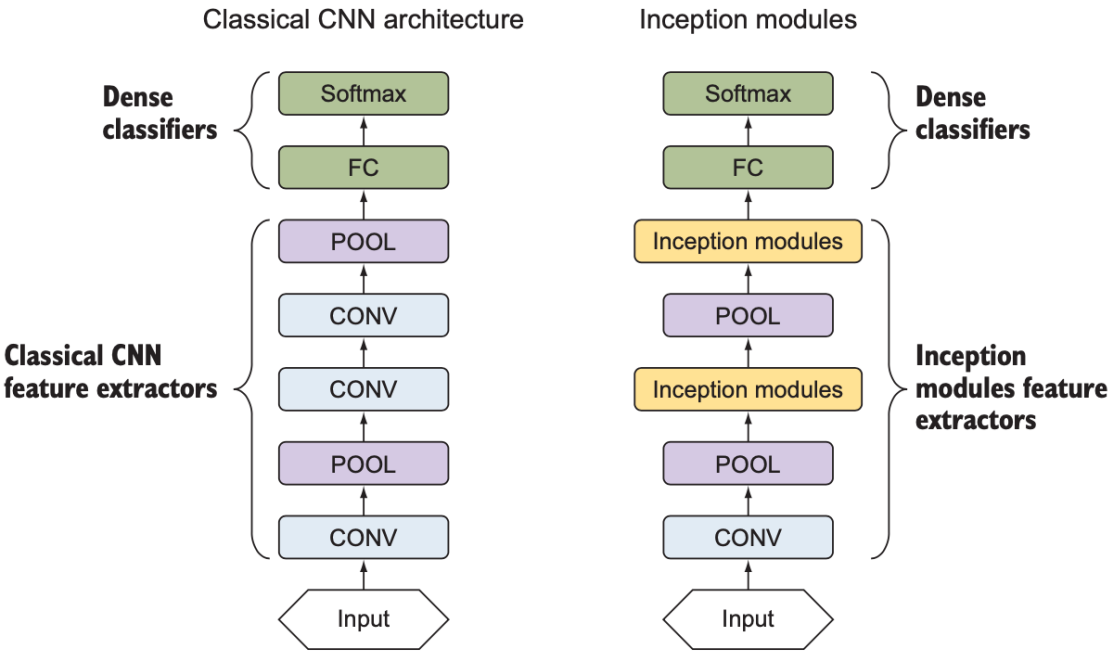
- **5.5 인셉션과 GoogLeNet**
  - 인셉션은 구글에서 연구 논문을 통해 2014년에 발표했음
  - 인셉션의 구조의 특징은 신경망 내부적으로 계산 자원의 효율을 높였음
  - 인셉션 신경망 구조를 구현한 GoogLeNet은 22개 층으로 구성(VGGNet보다 층수가 많음)되었으나 파라미터 수는 VGGNet의 1/12에 불가함
    - VGGNet: 1억 3800만개, GoogLeNet: 1300만개
  - GoogLeNet은 적은 파라미터 수로 향상된 성능을 보임
  - 인셉션 신경망 구조는 AlexNet이나 VGGNet에서 따온 고전적인 CNN 구조를 따르지만 인셉션 모듈 inception module 이라는 새로운 요소를 도입하였음

5. 고급 합성곱 신경망 구조

- **5.5.1 인셉션 구조에서 발전된 부분**
  - 기존 신경망 구조들은 각 층마다 〈합성곱층의 커널 크기〉와 〈풀링층의 배치〉 등의 파라미터를 결정하기 위해 다양한 시행착오가 필요했음
    - **합성곱층의 커널 크기**: 앞서 살펴본 신경망 구조는 커널 크기가 제각각임. 1x1, 3x3, 5x5, 큰 것은 11x11 (AlexNet)까지 있었음. 합성곱층을 설계하다 보면 데이터셋에 맞는 커널 크기를 선택하게 됨. 크기가 작은 커널은 이미지의 세세한 특징을 포착할 수 있고, 큰 커널은 세세한 특징을 놓치기 쉬움
    - **풀링층의 배치**:AlexNet 구조에서는 1개 또는 2개 합성곱층마다 풀링층을 배치해 특징을 다운샘플링했음. VGGNet은 2개, 3개, 4개 합성곱층마다 풀링층을 매치했는데 신경망의 얕은 쪽에 풀링층을 더 자주 배치했음 (즉, 2개 합성곱층마다 풀링층 배치)
  - 인셉션에서는 합성곱의 필터 크기나 풀링층의 배치를 직접 결정하는 대신 블록 전체에 똑같은 설정(인셉션모듈)을 적용
  - GoogLeNet은 기존의 방식대로 층을 쌓는 대신 〈커널 크기가 서로 다른 합성곱층으로 구성된 인셉션 모듈〉을 쌓는 방식으로 신경망을 구성함

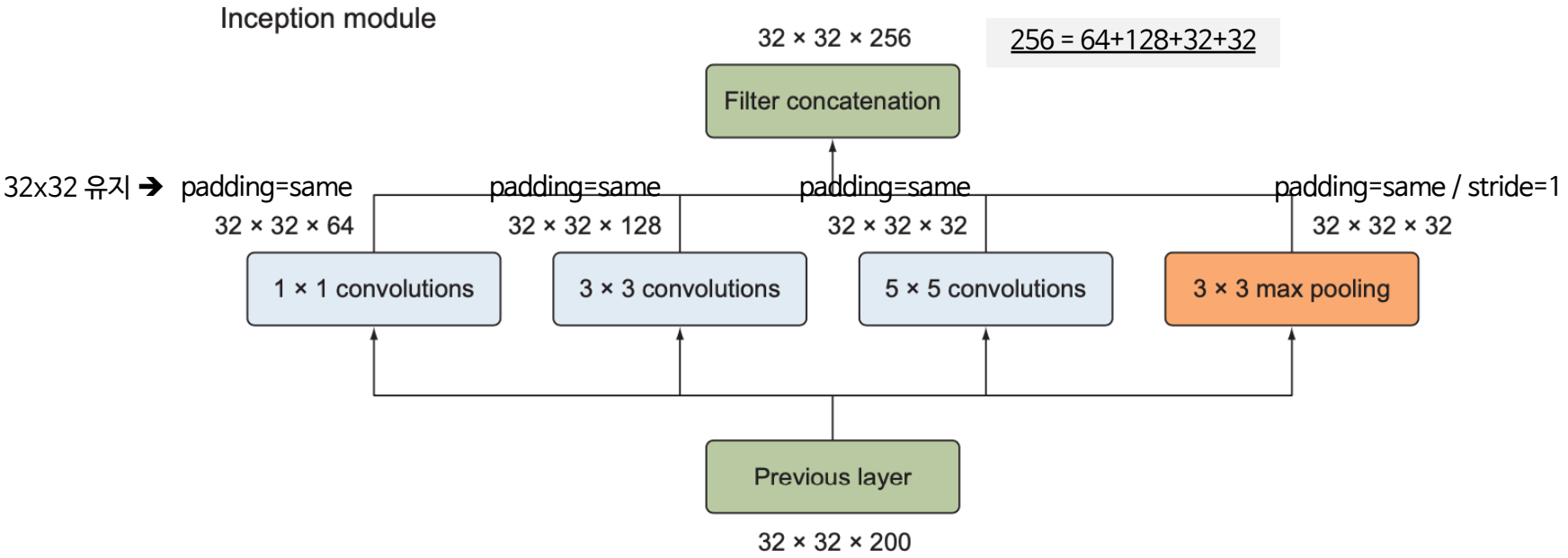
5. 고급 합성곱 신경망 구조

- **5.5.1 인셉션 구조에서 발전된 부분**
  - LeNet, AlexNet, VGGNet 등의 기존 방식에서는 합성곱층과 풀링층을 번갈아 쌓아 올리는 방식으로 특징 추출기를 구성함
  - 인셉션 구조에서는 인셉션 모듈과 풀링층을 쌓아 특징 추출기를 구성함



5. 고급 합성곱 신경망 구조

- **5.5.2 단순 인셉션 모듈**
  - 인셉션 모듈의 구성
    - 1x1 합성곱층, 3x3 합성곱층, 5x5 합성곱층, 3x3 최대 풀링층
    - 각 층의 출력은 연접 처리concatenation를 통해 하나의 출력으로 합쳐져 다음 단계의 입력이 됨



5. 고급 합성곱 신경망 구조

- **5.5.3 차원 축소가 적용된 인셉션 모듈**
  - 단순 인셉션 모듈은 5x5 합성곱층과 같은 크기가 큰 필터를 포함하기 때문에 계산 비용이 큼
    - 32x32x200 크기의 입력이 5x5크기의 합성곱 필터 32개를 갖춘 합성곱층에 입력
    - 필요한 곱셈의 수는 (32x32x200)x(5x5x32)=약 1억 6300만회
  - 이때 연산수를 줄이기 위해 차원 축소층을 도입이 필요

- **5.5.3.1 차원 축소층(1x1합성곱층)**
  - 1x1합성곱층을 도입하면 1억 6300만회의 곱셈을 약 1/10으로 줄일 수 있음
  - 3x3 또는 5x5 처럼 크기가 큰 필터를 포함하는 합성곱층 앞에 1x1합성곱층을 배치해서 입력의 깊이를 축소하고 필요한 연수 횟수를 줄임

5. 고급 합성곱 신경망 구조

- **5.5.3 차원 축소가 적용된 인셉션 모듈**
  - 단순 인셉션 모듈은 5x5 합성곱층과 같은 크기가 큰 필터를 포함하기 때문에 계산 비용이 큼
    - 32x32x200 크기의 입력이 5x5크기의 합성곱 필터 32개를 갖춘 합성곱층에 입력
    - 필요한 곱셈의 수는 (32x32x200)x(5x5x32)=약 1억 6300만회
  - 이때 연산수를 줄이기 위해 차원 축소층을 도입이 필요

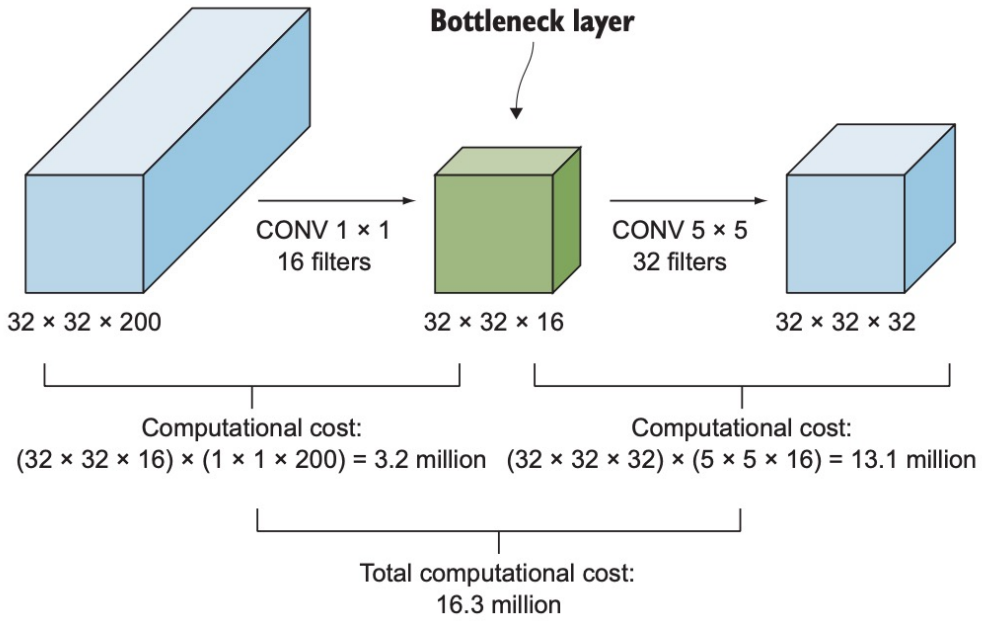
- **5.5.3.1 차원 축소층(1x1합성곱층)**
  - 1x1합성곱층을 도입하면 1억 6300만회의 곱셈을 약 1/10으로 줄일 수 있음
  - 3x3 또는 5x5 처럼 크기가 큰 필터를 포함하는 합성곱층 앞에 1x1합성곱층을 배치해서 입력의 깊이를 축소하고 필요한 연수 횟수를 줄임

5. 고급 합성곱 신경망 구조

- **5.5.3.1 차원 축소층(1x1합성곱층)**
  - 연산이 줄어드는 과정
    - 계산비용 = 1x1합성곱층의 연산 횟수 + 5x5합성곱층의 연산 횟수

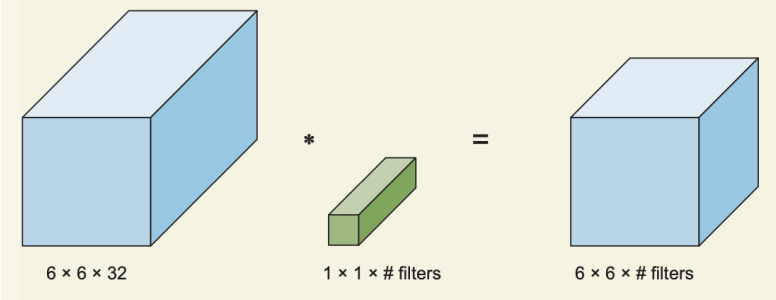
      = (32x32x200)x(1x1x16)+(32x32x16)x(5x5x32)

      = 3276800 + 13107200 = 약 1630만

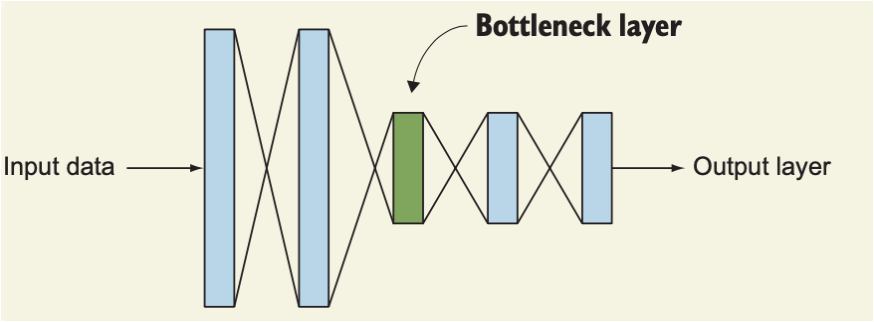


5. 고급 합성곱 신경망 구조

- **5.5.3.1 차원 축소층(1x1합성곱층)**
  - 1x1 합성곱층을 이용한 차원 축소는 이 합성곱층이 이미지 크기는 그대로 유지한 채 필터 수를 통해 출력의 채널 수(깊이)를 자유로이 변경할 수 있다는 점을 이용한 것

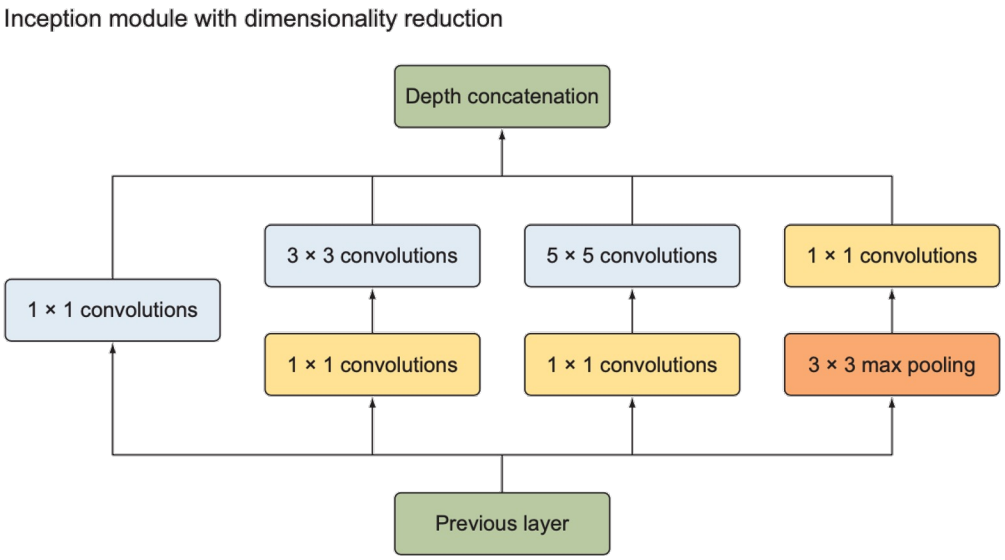


  - 1x1합성곱층은 병목층 bottleneck layer 라고 하는데, 신경망 구조 내에서 차원을 축소하는 이층의 역할이 마치 병에서 가장 좁은 병목을 연상케 하기 때문



5. 고급 합성곱 신경망 구조

- **5.5.3.2 차원 축소가 신경망 성능에 미치는 영향**
  - 과격한 차원 축소로 신경망의 성능에 나쁜 영향을 미치지 않을까 걱정될 수 있으나, 실험 결과 적정 수준을 유지하는 차원 축소는 신경망의 성능에 영향을 미치지 않음
  - 차원 축소가 적용된 인셉션 모듈
    - 3x3, 5x5 합성곱층 앞에 각각 1x1합성곱층을 배치
    - 풀링층은 입력의 차원을 축소하지 않고 〈3x3 최대 풀링층 뒤에 1x1 합성곱층을 하나 배치해〉 풀링층의 출력에 차원 축소를 적용

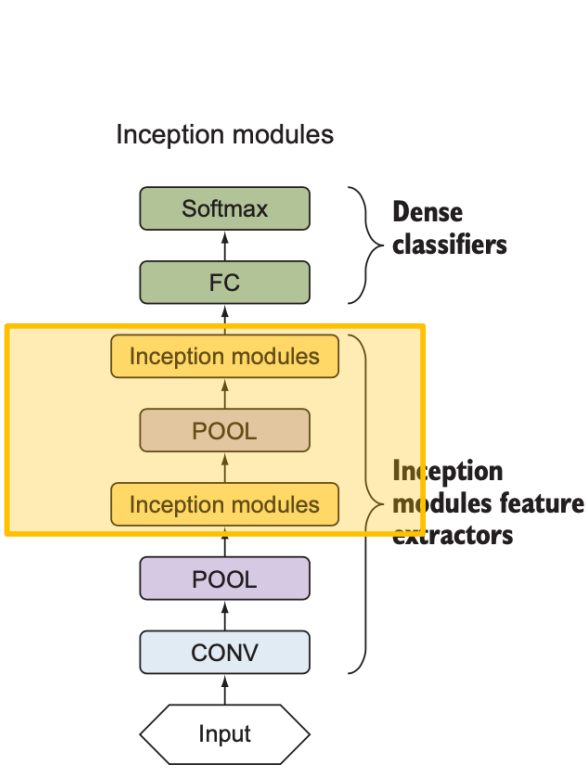


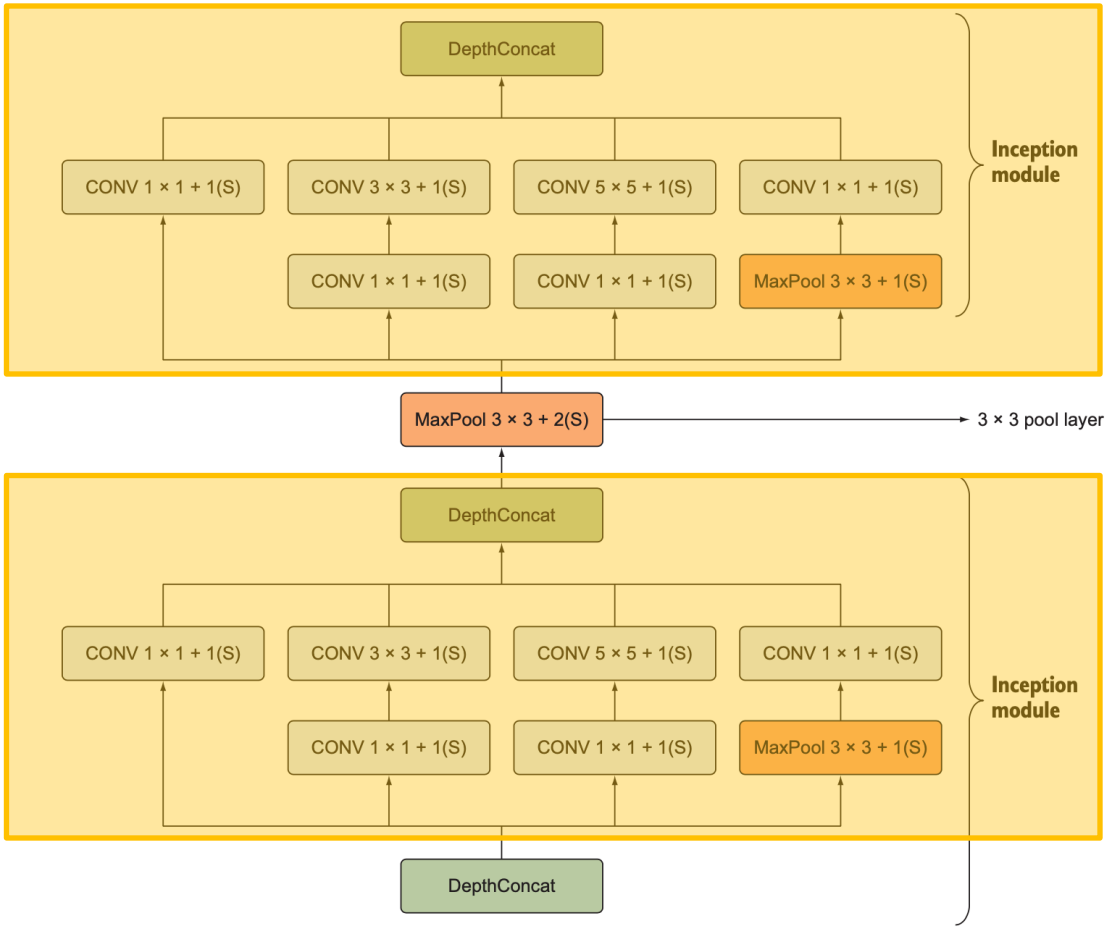
5. 고급 합성곱 신경망 구조

- **5.5.3.3 인셉션 모듈 정리**
  - 합성곱층의 필터 크기나 풀링층의 위치를 따로 정하고 싶지 않다면 인셉션 모듈을 이용해서 원하는 모든 크기의 필터를 사용한 결과를 모두 연접해서 출력하면 되며, 이러한 방식은 **인셉션 모듈의 단순 표현** naïve representation 에 해당됨
  - 단순 표현 시 발생하는 계산 비용 문제를 해결하기 위해 3x3, 5x5 합성곱층 앞과 최대풀링층 뒤에 1x1합성곱층인 축소층을 배치할 수 있으며, 이러한 방식은 **차원이 축소된 인셉션 모듈**에 해당됨

5. 고급 합성곱 신경망 구조

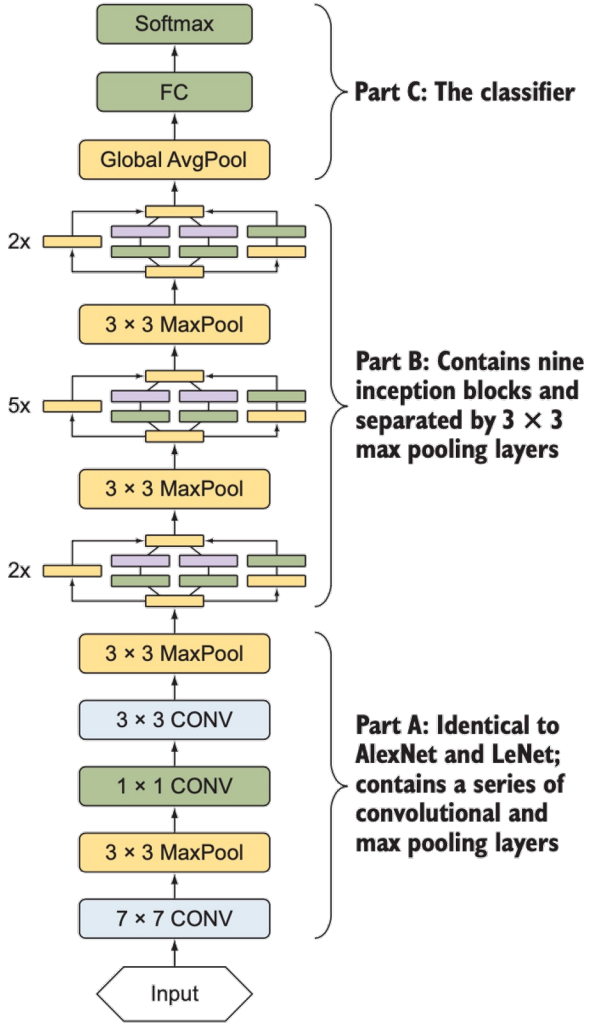
- 5.5.4 인셉션 구조
  - 차원 축소가 적용된 인셉션 모듈





5. 고급 합성곱 신경망 구조

- 5.5.4 GoogLeNet 구조



전결합층과 소프트맥스층으로 구성된 신경망의 분류기 구조

9개의 인셉션 모듈 ➔
인셉션 모듈 2개+풀링층+인셉션 모듈 5개+풀링층+인셉션 모듈 2개

합성곱층과 최대 풀링층이 번갈아 배치된 AlexNet이나 LeNet 같은 구조

5. 고급 합성곱 신경망 구조

- 5.5.5.2 GoogLeNet의 신경망 구조

	type	patch size/ stride	output size	depth	#1 × 1	#3 × 3 reduce	#3 × 3	#5 × 5 reduce	#5 × 5	pool proj	params	ops
Part A	convolution	7 × 7/2	112 × 112 × 64	1							2.7K	34M
	max pool	3 × 3/2	56 × 56 × 64	0								
	convolution	3 × 3/1	56 × 56 × 192	2		2	2	2	2	2	112K	360M
	max pool	3 × 3/2	28 × 28 × 192	0								
Part B	inception (3a)		28 × 28 × 256	2	64	96	128	16	32	32	159K	128M
	inception (3b)		28 × 28 × 480	2	128	128	192	32	96	64	380K	304M
	max pool	3 × 3/2	14 × 14 × 480	0								
	inception (4a		14 × 14 × 512	2	192	96	208	16	48	64	364K	73M
	inception (4b)		14 × 14 × 512	2	160	112	224	24	64	64	437K	88M
	inception (4c)		14 × 14 × 512	2	128	128	256	24	64	64	463K	100M
	inception (4d)		14 × 14 × 528	2	112	144	288	32	64	64	580K	119M
	inception (4e)		14 × 14 × 832	2	256	160	320	32	128	128	840K	170M
	max pool	3 × 3/2	7 × 7 × 832	0								
	inception (5a)		7 × 7 × 832	2	256	160	320	32	128	128	1072K	54M
	inception (5b)		7 × 7 × 1024	2	384	192	384	48	128	128	1388K	71M
Part C	avg pool	7 × 7/1	1 × 1 × 1024	0								
	dropout (40%)		1 × 1 × 1024	0								
	linear		1 × 1 × 1000	1							1000K	1M
	softmax		1 × 1 × 1000	0								

5. 고급 합성곱 신경망 구조

- **5.5.6 하이퍼파라미터 설정하기**
  - 학습률 스케줄러: 8 에포크 마다 학습률 learning rate 4%씩 감소
  - 최적화 알고리즘: 경사하강법
  - 모멘텀: 0.9

5. 고급 합성곱 신경망 구조

- **5.5.7 CIFAR 데이터셋을 대상으로 한 인셉션의 성능**
  - GoogLeNet은 2014년 ILSVRC에서 우승하였음
  - GoogLeNet의 top-5 오차율은 6.67%로 AlexNet이나 VGGNet 등 기존 CNN 구조보다 훨씬 뛰어난 것은 물론이고 사람에 필적할만한 성능을 보였음

5. 고급 합성곱 신경망 구조

- **5.6 ResNet (Residual Neurual Network)**
  - 잔차 신경망ResNet은 2015년에 마이크로소프트 리서치 팀에서 제안한 신경망 구조
  - ResNet에는 잔차 모듈residual module 과 스킵 연결skip connection 이라는 새로운 구조가 사용되었으며, 은닉층에도 강한 배치 정규화가 적용 되었음
  - ResNet은 2015년에 ILSVRC에서 top-5 오차율 3.57%을 기록하며 가장 좋은 성능을 기록함

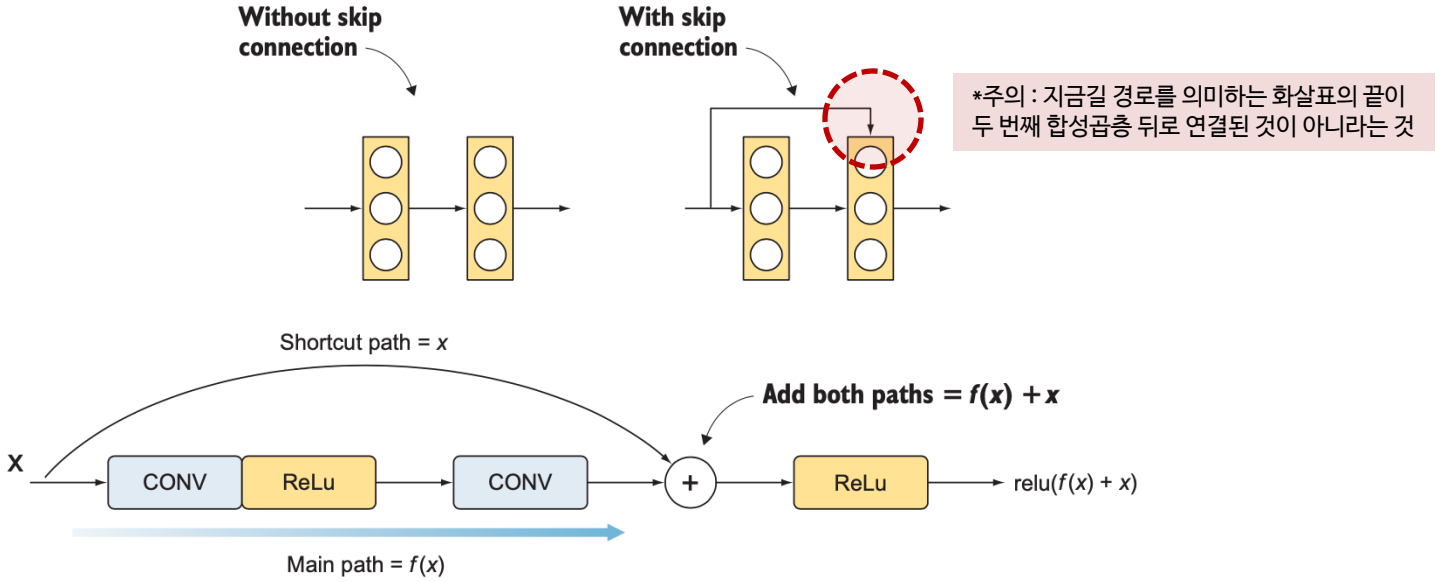
  - 배치 정규화가 강하게 적용된 덕분에 파라미터가 있는 층이 50층, 101층, 152층이나 되는 깊은 신경망(ResNet50, ResNet101, ResNet152)의 복잡도를 훨씬 층수가 적은 VGGNet(19층)보다 낮출 수 있음
    - 신경망이 지나치게 깊어지면 과적합이 발생하기 쉬움
    - ResNet은 배치 정규화를 통해 과적합 문제를 해결하였음

5. 고급 합성곱 신경망 구조

- **5.6.1 ResNet에서 발전된 부분**
  - LeNet➔AlexNet➔VGGNet➔GoogeLeNet 신경망 구조의 발전을 통해 신경망의 층수가 많을 수록 이미지에서 더 좋은 특징을 추출할 수 있다는 것을 발견함
    - 일반적으로 신경망의 층수가 많을 수록 신경망 모델의 표현력이 좋아지며, 이는 단순한 모서리(앞쪽층)에서 부터 복잡한 패턴(뒤쪽층)에 이르기까지 다양한 추상화 수준의 특징을 학습할 수 있기 때문임
  - VGG19(19층)와 GoogLeNet(22층) 구조보다 신경망 층수를 늘리기 위해서는 과적합 문제와 기울기 소실 문제를 해결해야 함
    - 과적합 문제는 드롭아웃, L2 규제화, 배치 정규화를 통해 해결 가능
    - 기울기 소실 문제는 ResNet 구조가 제안한 스킵 연결을 통해 해결 가능

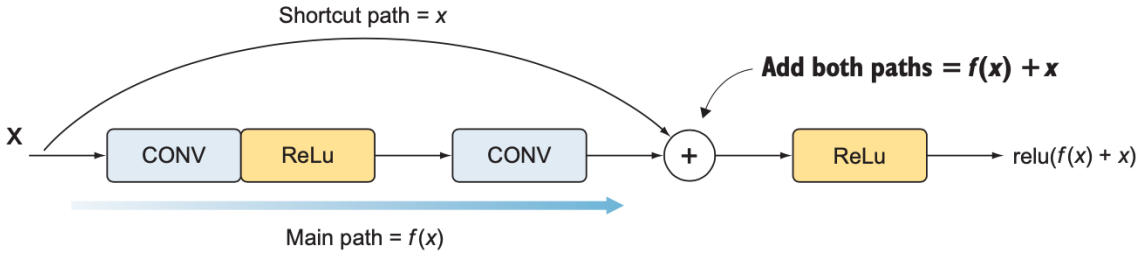
5. 고급 합성곱 신경망 구조

- **5.6.1 ResNet에서 발전된 부분**
  - ResNet 연구진은 기울기 소실문제 해결을 위해 스킵 연결을 제안함
  - 스킵 연결이란 뒤쪽 층의 기울기를 앞쪽 층에 직접 전달하는 별도의 경로를 추가하는 것으로 〈앞쪽 층의 정보가 뒤쪽 층에 전달〉하는 것을 의미함
  - 스킵 연결은 〈항등 함수를 학습〉할 수 있어 층이 쌓여도 앞쪽 층보다 성능이 하락하지 않는 역할을 수행함



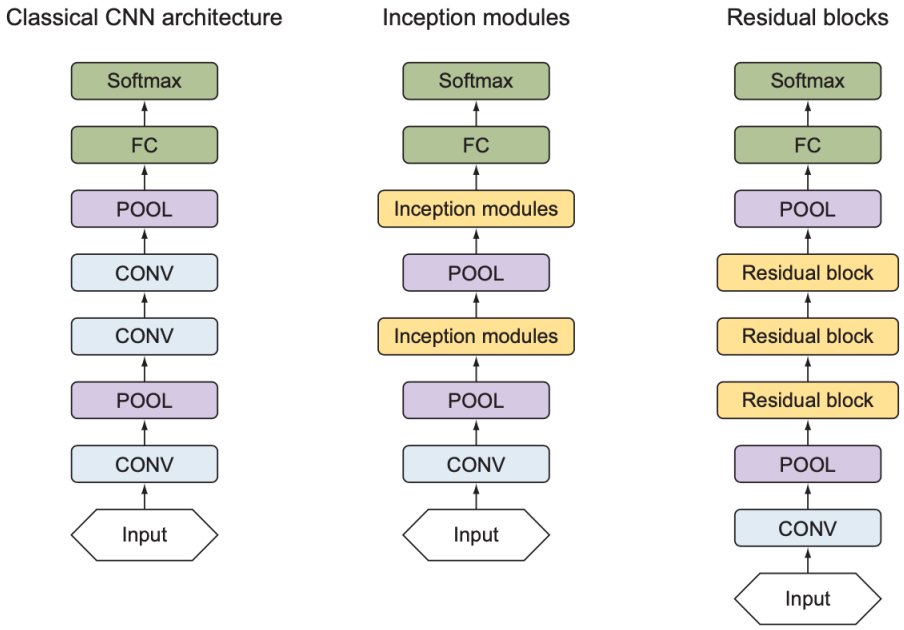
5. 고급 합성곱 신경망 구조

- **5.6.1 ResNet에서 발전된 부분**
  - 합성곱층에 스킵 연결을 추가한 구조를 잔차 블록 residual block 이라고 함



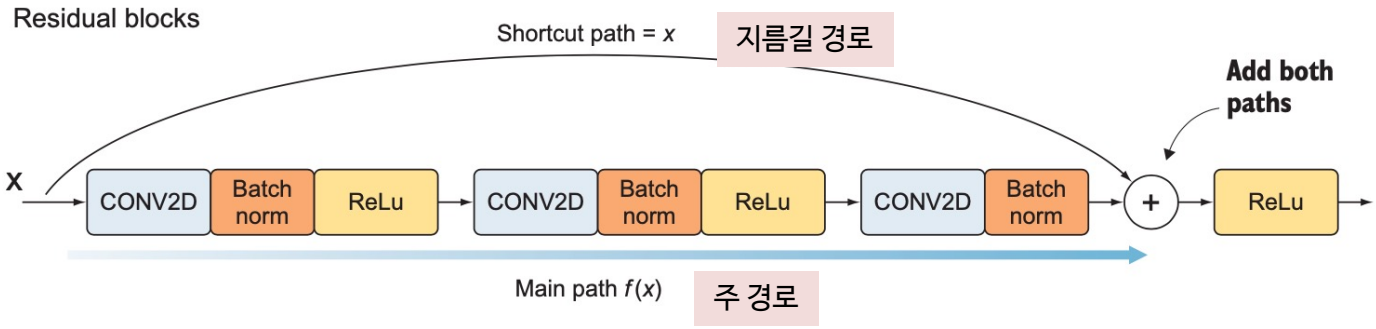
5. 고급 합성곱 신경망 구조

- **5.6.1 ResNet에서 발전된 부분**
  - 인셉션과 마찬가지로 ResNet 구조도 여러 개의 잔차 블록이 늘어선 형태로 구성됨
    - 특징 추출기: ResNet 의 특징 추출기 부분은 합성곱층과 풀링층으로 시작해서 그 뒤로 잔차 블록이 1개 이상 이어지는 구조. ResNet은 원하는 만큼 잔차 블록을 추가해서 신경망의 층수를 늘릴 수 있음
    - 분류기: 분류기 부분는 다른 신경망과 마찬가지로 전결합층과 소프트맥스 함수로 구성됨



5. 고급 합성곱 신경망 구조

- **5.6.2 잔차 블록**
  - 잔차 블록은 다음 2개의 경로로 나뉨
    - **지름길 경로:** 입력을 주 경로의 ReLU 입력 전으로 전달함
    - **주 경로:** 일련의 합성곱 연산 및 활성화 함수. 주 경로는 ReLU 활성화 함수를 사용하는 3개의 합성곱층으로 구성됨. 과적합 방지와 학습 속도 향상을 위해 각 합성곱층마다 배치 정규화를 적용함. 결과적으로 주 경로의 구조는 [CONV=〉BN=〉ReLU]x3 와 같음



  - 지름길 경로의 값은 잔차 블록 마지막 합성곱층의 활성화 함수 바로 앞에서 주 경로의 값과 합하여 이 값을 ReLU 함수에 통과시킴

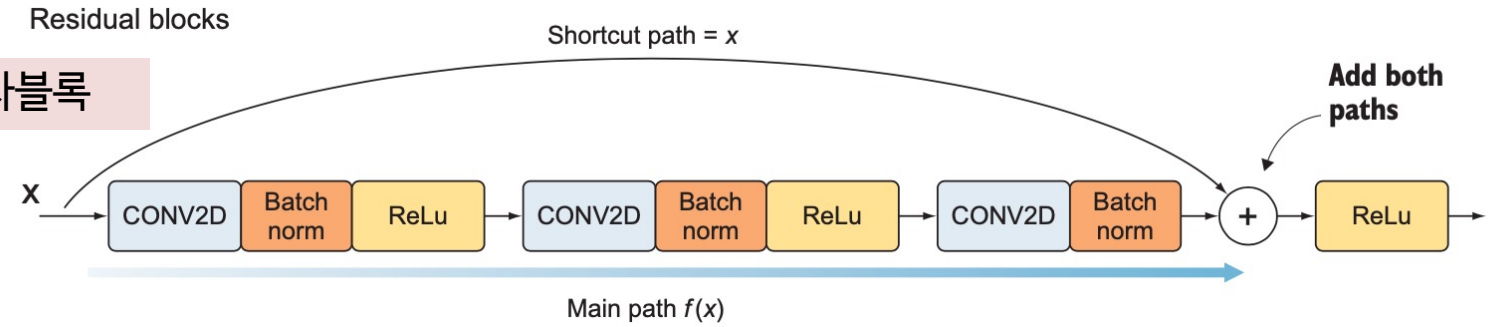
5. 고급 합성곱 신경망 구조

- **5.6.2 잔차 블록**
  - 잔차 블록에는 풀링층이 없음
  - 인셉션 구조와 비슷하게 1x1 합성곱층을 사용하여 다운샘플링을 적용
    - 잔차 블록 처음에 1x1 합성곱층을 배치하고, 출력에는 3x3 합성곱층과 1x1 합성곱층을 하나씩 배치해서 두 번에 걸쳐 차원을 축소함. 이 구조는 여러 층에 걸쳐 입출력의 차원을 제어할 수 있다는 것이 장점
    - 차원 축소가 적용된 블록을 **병목 잔차 블록**bottleneck residual block 이라고 함
  - 주 경로와 지름길 경로로 전달되는 정보의 차원이 동일해야 연산(행렬덧셈)이 가능
    - 주 경로에 차원 축소층을 적용하여 정보의 차원을 축소 했다면, 지름길 경로에도 **병목층(1x1합성곱층+배치정규화)** 을 적용하여 정보의 차원을 축소해야 함
    - 이러한 구조를 **축소 지름길 경로**reduce shortcut 라고 함

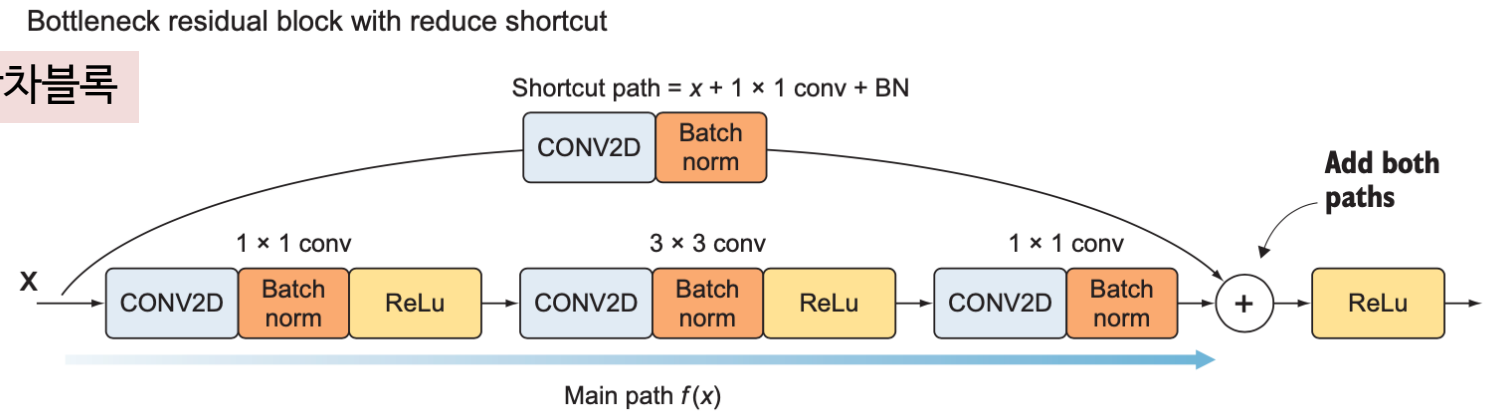
5. 고급 합성곱 신경망 구조

- 5.6.2 잔차 블록

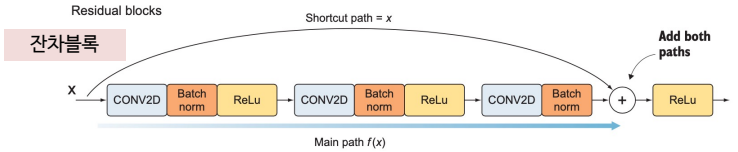
잔차블록

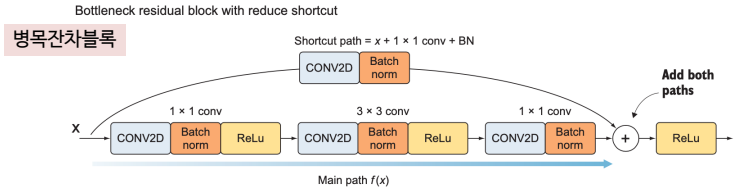


병목잔차블록



5. 고급 합성곱 신경망 구조





- **5.6.2 잔차 블록**
  - 잔차 블록에는 지름길 경로와 주경로가 있음
  - 주 경로는 3개의 합성곱층에 각각 배치 정규화가 적용됨
    - 1x1합성곱층
    - 3x3합성곱층
    - 1x1합성곱층
  - 지름길 경로를 구현하는 1x1합성곱층 방법은 두가지
    - 일반 지름길 경로: 주 경로와 입력을 더하는 일반적인 지름길 경로
    - 축소 지름길 경로: 주 경로와 입력을 더하는 지점 이전에 합성곱층이 하나 배치된 지름길 경로

5. 고급 합성곱 신경망 구조

- **5.6.3 ResNet 구현하기**
  - 잔차 블록을 모아 ResNet 신경망 구조를 구현
  - 논문에 소개된 ResNet 신경망의 종류와 구조
    - ResNet50 : ResNet 의 여러 버전 중에서도 50개의 파라미터층을 갖춘 구조

Layer name	Output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112x112	7x7, 64, stride 2				
conv2_x	56x56	3x3, maxpool, stride 2				
		[3x3, 64 3x3, 64] x2	[3x3, 64 3x3, 64] x3	[1x1, 64 3x3, 64 1x1, 256] x3	[1x1, 64 3x3, 64 1x1, 256] x3	[1x1, 64 3x3, 64 1x1, 256] x3
conv3_x	28x28	[3x3, 128 3x3, 128] x2	[3x3, 128 3x3, 128] x4	[1x1, 128 3x3, 128 1x1, 512] x4	[1x1, 128 3x3, 128 1x1, 512] x4	[1x1, 128 3x3, 128 1x1, 512] x8
conv4_x	14x14	[3x3, 256 3x3, 256] x2	[3x3, 256 3x3, 256] x6	[1x1, 256 3x3, 256 1x1, 1024] x6	[1x1, 256 3x3, 256 1x1, 1024] x23	[1x1, 256 3x3, 256 1x1, 1024] x36
conv5_x	7x7	[3x3, 512 3x3, 512] x2	[3x3, 512 3x3, 512] x3	[1x1, 512 3x3, 512 1x1, 2048] x3	[1x1, 512 3x3, 512 1x1, 2048] x3	[1x1, 512 3x3, 512 1x1, 2048] x3
	1x1	Average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

5. 고급 합성곱 신경망 구조

- **5.6.3 ResNet 구현하기**
  - ResNet50의 구성
    - 1단계: 7x7 합성곱층
    - 2단계: 잔차 블록 3개 = 합성곱층 9개
      - (각각 1x1합성곱층+3x3 합성곱층 +1x1합성곱층)
    - 3단계: 잔차 블록 4개 = 합성곱층 12개
    - 4단계: 잔차 블록 6개 = 합성곱층 18개
    - 5단계: 잔차 블록 3개 = 합성곱층 9개
    - 소프트맥스 활성화 함수를 사용하는 전결합층

Layer name	Output size	50-layer
conv1	112x112	7x7, 64, stride 2
conv2_x	56x56	3x3, maxpool, stride 2 [1x1, 64 3x3, 64 1x1, 256] x3
conv3_x	28x28	[1x1, 128 3x3, 128 1x1, 512] x4
conv4_x	14x14	[1x1, 256 3x3, 256 1x1, 1024] x6
conv5_x	7x7	[1x1, 512 3x3, 512 1x1, 2048] x3
	1x1	Average pool, 1000-d fc, softmax
FLOPs		3.8×10^9

블록구조

잔차블록수

5. 고급 합성곱 신경망 구조

- **5.6.4 하이퍼파라미터 설정하기**
  - 미니배치 경사하강법
  - 모멘텀 0.9 / 학습률 초기값 0.1, 검증오차 감소 없을 시 학습률 1/10로 감소
  - 가중치 감소율 0.0001
  - L2규제화 함께 사용 (단, 아래 구현코드에서는 생략)

5. 고급 합성곱 신경망 구조

- **5.6.5 ImageNet 데이터셋을 대상으로 한 ResNet의 성능**
  - 2015년 ILSVRC에서
    - ResNet-512은 top-5 오차율 4.49% 달성
    - 앙상블 모델 기준 top-5 오차율 3.57% 달성 (대회에서 우승!)
      - [참고] GoogLeNet은 top-5 오차율 6.67%

5. 고급 합성곱 신경망 구조

- **5.7 마치며**
  - 기본적인 합성곱 신경망 구조는 서로 다른 설정을 가진 합성곱층과 풀링층을 번갈아가며 배치하는 방식으로 구성된다.
  - LeNet은 5개의 파라미터층으로 구성된다. 그중 3개는 합성곱층, 2개는 전결합층이며, 첫번째와 두 번째 합성곱층 뒤에는 풀링층이 배치되어 있다.
  - AlexNet은 LeNet보다 많은 8개의 파라미터층을 갖는다. 5개는 합성곱층, 3개는 전결합층이다.
  - VGGNet은 전체 신경망에서 동일한 하이퍼파라미터 설정을 사용하는 방식으로 합성곱층 및 풀링층의 하이퍼파라미터 설정 문제의 해결을 시도했다.
  - 인셉션은 VGGNet과는 다른 방식으로 하이퍼파라미터 설정 문제를 해결하려 했다. 특정 필터 또는 풀링 크기를 지정하는 대신 다양한 필터 크기와 풀링 크기를 한꺼번에 사용했다.
  - ResNet은 인셉션과 비슷한 접근 방식으로 잔차 블록을 도입했다. 전체 신경망은 이 잔차 블록으로 구성된다.신경망의 층수가 일정 이상 늘어나면 기울기 소실 문제로 학습이 잘되지 않는데, 신경망의 서로 떨어진 층을 이어주는 스킵 연결 도입을 통해 기울기를 전달했다. 이런 방법으로 수백 층에 이르는 거대한 신경망의 학습을 가능케 하는 돌파구를 열었다.

실습

본 자료는 한빛미디어에서 나온 〈비전 시스템을 위한 딥러닝〉 책을 참고하여 제작 되었음을 알립니다.

5. 고급 합성곱 신경망 구조

- **고급 합성곱 신경망 실습에 사용할 코드**

  - ⟨LeNet⟩
    https://www.kaggle.com/code/sukzoon1234/2024-1-dls-w5-lenet/notebook

➔ 학습까지 가능한 형태의 코드

  - ⟨AlexNet⟩
    https://www.kaggle.com/code/sukzoon1234/2024-1-dls-w5-alexnet/notebook?scriptVersionId=170429532
  - ⟨VGG⟩
    https://www.kaggle.com/code/sukzoon1234/2024-1-dls-w5-vgg/notebook?scriptVersionId=170465155
  - ⟨GoogLeNet⟩
    https://www.kaggle.com/code/sukzoon1234/2024-1-dls-w5-googlenet?scriptVersionId=170481069
  - ⟨ResNet⟩
    https://www.kaggle.com/code/sukzoon1234/2024-1-dls-w5-resnet?scriptVersionId=170486745

➔ **신경망 설계만 되어 있는 코드를 학습까지 직접 해보기! (1epoch)**