3. GOOGLENET (2014) - Going Deeper with Convolutions

GoogLeNet

- network 내부의 Computing resources 개선
- Depth와 Width를 계산량을 유지하면서 늘리는 것이 목표
- Multi-scale processing과 Hebbian principle 구조를 기초로함
- GoogLeNet은 22개의 Layer로 구성

Introduce

- GoogleNet은 2012년 AlexNet보다 12배 적은 parameter를 사용하였음에도 훨씬 정확한 결과를 얻을 수 있었음
- Object Detection의 큰 이점은 Deep network와 아주 큰 model에서 오는 것이 아닌 R-CNN 같이 deep architecture와 고전적인 컴퓨터 비전의 시너지에서 오게 됨
- mobile 및 embedded 컴퓨팅이 지속적으로 발전하여 알고리즘의 효율성, 메모리 사용의 중요성이 커지고 있기에 해당 논문에서는 효율적인 계산, 메모리 사용량에 대한 내용이 포함
- 컴퓨터 비전 분야에서 효율적인 deep CNN architecture인 inception에 focus를 맞춤

Related Work

- LeNet-5을 시작으로 CNN은 일반적으로 하나 이상의 Fully-Connected Layer가 뒤에 오는 Stacked Convolutional Layer를 가지게됨
    - LeNet-5 ~ CNN 기본 구조
        - 컨볼루션 레이어 - 이미지에서 특징을 추출하는 역할, 필터를 사용하여 이미지를 스캔하면서 로컬패턴을 학습
        - 풀링 레이어 - 차원 축소를 위해 사용되며, 일반적으로 최대 풀링(max pooling)이나 평균 풀링(average pooling)을 사용
          이는 모델의 오버피팅을 줄이고 계산 효율성을 높이는 데 도움이 됨

        - 완전 연결 레이어 - 모델의 마지막 부분에 위치하며, 앞서 추출한 특징을 바탕으로 최종 분류 및 예측을 수행
    - GoogleNet의 혁신
        - Inception 구조를 도입하여 CNN의 깊이와 너비를 획기적으로 증가시킴
        - 해당 구조는 다양한 크기의 필터를 동시에 사용하여 다양한 스케일의 특징을 동시에
          학습할 수 있게 함
        - GoogleNet은 LeNet-5와 같은 초기 모델들보다 훨씬 깊은 네트워크 구조를 가지면서도, 파라미터의 수는 상대적으로 적게 유지하여 효율적인 학습이 가능하게 만들어짐
        - LeNet-5로 시작된 CNN의 발전은, 단순히 층을 쌓는 것에서 벗어나, 보다 복잡한 구조나 효율적인 학습 방법론으로 진화했으며 GoogleNet은 이러한 발전의 대표적인 예시 중 하나
- ImageNet classification같은 것은 대규모 데이터셋이므로 layer의 수와 layer의size를 늘리고 있으며, overfitting 문제를 해결하기 위해서 dropout을 사용
- maxpooling layer가 spatial information의 손상을 초래 한다는 우려에도 불구하고, 동일한 CNN 이 localization,object detection, human pose estimation 분야에서 성공적으로 채택
  Inception model이 여러번 반복되는 GooleNet model에 경우 22-layer의 deep model임
    - 공간 정보의 압축이 유용함
        - 특징강조 : Max Pooling은 입력 특징 맵에서 가장 두드러진 특징을 보존하면서 덜 중요한 정보를 제거함, 이는 중요한 특징을 강조하고 모델이 핵심적인 패턴을 더 잘 인식하도록도움
        - 오버피팅 감소 : MaxPooling은 모델의 파라미터 수를 감소시키고, 과적합의 위험을 줄여줌
        - 계산효율성 증가 : 공간 차원을 축소함으로 연산량과 메모리 사용량이 줄어들어 모델의 효율성이 향상됨
    - 계층적 특징 학습
        - CNN은 여러 계층을 통해 저수준 특징에서 고수준 특징으로 점차적으로 복잡한 패턴을 학습함
        - 초기 레이어는 간단한 경계와 모양을 학습하는 반면, 깊은 레이어는 객체의 고유한 특성을 식별

        - 계층적 접근 방식은 공간 정보의 일부 손실에도 불구하고, 객체를 효과적으로 인식하고 위치를 추정할 수 있게 해줌
    - 추가적인 기술과 결합
        - 멀티 스케일 특징 학습 : CNN 아키텍처들은 다양한 크기의 커널을 사용하여 멀티 스케일 특징을 학습하거나, 복수의 레이어에서 특징을 결합하여 공간 정보의 손실을 보완
        - Localization 과 Detection을 위한 추가 모듈 : 객체 검출과 자세 추정 같은 과제에서는 공간 정보의 정밀한 추정이 필요함, 이를 위해 CNN구조에서 추가적인 localization 또는 detection모듈을 도입하여 공간 정보의 소실 문제를 해결
    - Max Pooling 레이어가 공간 정보의 일부 손실을 초래함에도 불구하고 ,CNN은 여전히 다양한 컴퓨터 비전 태스크에서 성공적으로 적용됨, 이는 CNN이 계층적 특징 학습, 효율적인 파라미터 사용, 추가적인 기술과 모듈의 결합을 통해 이러한 단점을 극복하기 때문

Motivation and High Level Considerations

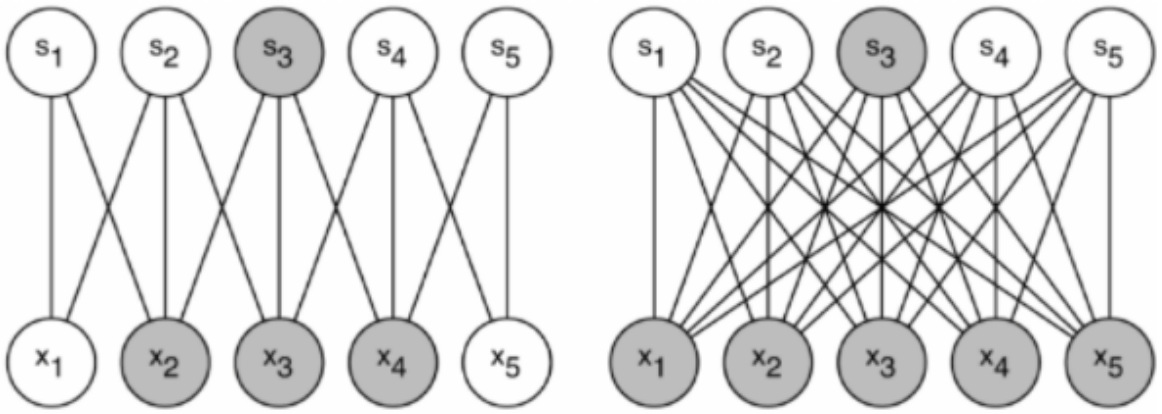
- deep neural networks는 성능을 향상시키는 것은 크기를 늘리는 것이라고 함
    - network의 depth와 width의 증가도 포함

(a) Siberian husky (b) Eskimo dog

Figure 1: Two distinct classes from the 1000 classes of the ILSVRC 2014 classification challenge.

- 아주 많은 양의 label된 train data를 사용할 수 있다는 점 덕분에 higher quality model을 쉽고 안전하게 훈련 가능하지만 두가지의 단점이 존재

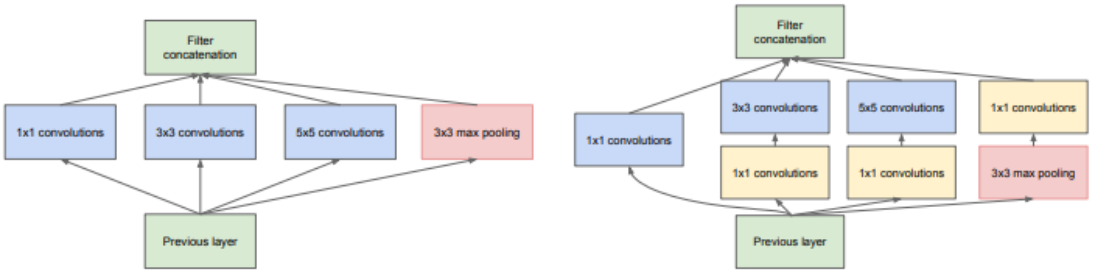
    - 크기가 클수록 일반적으로 parameter의 수가 많아서 특히 train set의 labeled된 예제 수가 제한 된 경우에는 networ가 overfitting되기 쉬움
    - high-quality의 training set을 만드는것은 상당히 까다롭고 비싸며, ImageNet과 같이 세분화된 영역에서는 이러한 것이 문제가될 수 있음
- 크기가 uniformly하게 증가된 network는 computational resource가 엄청나게 증가함
    - 예를들어, deep vision network에서 만약 두개의 convolution layer가 결합하면 filter 수가 균일하게 증가하여 계산량이 2배가 증가함
    - 만약 추가된 capacity의 가중치가 0에 가까운 경우 많은 계산량 낭비 할 수 있는 계산량을 유한하여, 주요 목적이 결과의 quality를 좋게 하기 위해선 크기보단 계산 resources를 효율적으로 분배하는것이 더 좋음



- 해결 방법
    - fully connected된 것에서 sparsely connected architecture로 바꾸는 것
        - 완전 연결 아키텍처의 한계
            - 완전 연결 레이어에서는 모든 입력 뉴런이 다음 레이어의 모든 뉴런과 연결됨
              이는 레이어간에 매우 높은 수의 파라미터를 유발함
                - 오버피팅 : 너무 많은 파라미터는 모델이 학습 데이터에 과적합 될 가능성을 증가시킴
                - 계산비용 : 모든 뉴런 간의 연결을 계산하려면 많은 계산 자원이 필요함 이는 학습과 추론 시간을 늘림
                - 메모리 요구사항 : 대규모 데이터셋에 대한 모델 학습은 많은 양의 메모리를 필요로 함

        - 희소 연결 아키텍처의 이점
            - 이 아키텍처는 연결의 수가 크게 감소하여, 모델이 필요로 하는 파라미터의 수와 계산 비용을 줄일 수 있음
                - 효율적인 파라미터 사용 : 더 적은 수의 연결을 사용하여 학습해야 할 파라미터의 수가 감소, 오버피팅의 위험을 줄이고 모델이 보다 일반화된 특징을 학습할 수 있도록 도움
                - 계산 효율성 : 더 적은 수의 연결은 더 적은 계산량을 의미하므로, 학습과 추론 과정을 더 빠르게 만듬
                - 구조적 유사성 : 자연의 신경망 구조, 특히 시각처리 시스템은 희소 연결 구조를 사용하여 실제 세계 데이터의 특성을 효과적으로 반영함
- 생물학적 시스템을 보고 따라한 것 이외에도, 획기적인 연구로 인해서 단단한 이론적 토대가 될 수 있음
- 어떤 연구에서는 주요 dataset의 probability distribution이 크고 spared deep neural network에 표현될 수 있는 경우, last later의 activation correclation statics를 분석하고 상관관계가 높은 출력을 가진 neuron을 clustering 하여 최적의 network toplogy를 얻을 수 있다고 함
- Spares matrix 연산을 다루는 문제는 Sparse matrix를 클러스터링 하여 Dense한 Sunbmatrix를 만드는 것을 제한함
- GoogleNet의 저자들은 Inception 구조는 Sparse 구조를 test하기 위해 시작했는데, hyperparameter를 조정하고 실험한 결과, 좋은 성과가 나옴

Architecture



(a) Inception module, naïve version (b) Inception module with dimension reductions

Figure 2: Inception module

- Inception architecture의 주요 아이디어는 convolution vision network의 optimal local sparse structure를 쉽게 구할 수 있는 Dense가 높은 components로 근사하고 커버가 가능한 방법을 찾는것을 base로 함
    - 최적의 지역적 희소 구조(Optimal Local Sparse Structure)
        - 컨볼루션 신경망에서 각 뉴런은 입력 이미지의 작은 영역에만 연결되어 있음
        - 실제 세계의 시각 시스템이 지역적인 정보를 기반으로 객첼르 인식하는 방식을 모방한 것
        - 신경망 전체를 통틀어 볼 때, 지역적 연결성은 전체 구조에서 볼 때 희소성(Sparse)을 의미함
    - 밀집된 컴포넌트로의 근사
        - 실제로 최적의 희소 구조를 직접 설계하는 것은 매우 어려움
        - Inception 아키텍처의 핵심 아이디어는 이러한 최적의 희소 구조를 밀집된 컴포넌트(dense components)로 근하는 것
        - 여러 크기의 컨볼루션 필터를 사용하여 다양한 지역적 특성을 동시에 캡처하고 이를 통합하는 방식으로 구현
        - 효율적인 계산과 더불어 다양한 크기의 특징을 포착하여 네트워크가 더 유연하게 중요한 정보를 학습할 수 있도록 함
- Translation invariance를 가정한 뒤에 network의 convolution building block을 구축한다는 것을 의미
    - Translation invariance는 객체가 이미지 내에서 위치를 달리해도 인식될 수 있어야 한다는 원칙
    - 컨볼루션 레이어가 자연스럽게 가지고 있는 특성으로, 같은 객체나 패턴이 이미지의 다른 위치에 나타나더라도동일하게 인식될 수 있음
    - Inception 아키텍처에서는 이 원칙을 확장하여 다양한 크기와 비율의 객체가 이미지 내에서 어디에 위치하든지 간에 효과적으로 인식될 수 있도록 설계됨
    - Inception 아키텍처의 구축 블록은 이러한 원칙을 바탕으로 여러 크기의 컨볼루션 필터를
      병렬로 배치하고, 결과를 통합하여 네트워크의 입력으로 사용함
        - 네트워크는 더 많은 컨텍스트 정보와 공간적 해상도를 유지하면서 다양한 스케일의 특징을 효과적으로 학습할 수 있음
    - 복잡한 이미지 내에서 다양한 객체와 패턴을 정확하고 효율적으로 인식할 수 있는 강력한 신경망 구조를 만들어냄

- Inception module은 서로 stacked되는데, output의 correlation staristics는 달라질 수 밖에 없음higher abstraction의 features가 higher layer의 captured됨에 따라서 spatial concentration이 감소할 것으로 예상이 되어 3x3 및 5x5 convolution의 비율이 높아져야 함
  - 높은 수준의 추상화와 공간 집중도의 감소
    - 신경망의 초기 층은 주로 간단한 패턴(예: 가장자리, 각도 등)을 학습하는 반면, 네트워크가 깊어질수록 더 복잡한 객체와 개념(예: 텍스처, 객체의 일부 등)을 인식하게 됨
    - 이 과정에서, 더 높은 레이어는 더 넓은 영역의 정보를 통합하여, 더 높은 수준의 추상화된 특징을 학습함
      - 이로인해 공간 집중도(spatial concentration)가 감소하게 되는데, 이는 더 넓은 영역에서 정보를 수집하고 합치기 때문에 발생함
  - 3×3 및 5x5 컨볼루션 필터의 중요성 증가
    - Inception 아키텍처는 더 높은 레이어에서 3x3 및 5x5 같은 더 큰 컨볼루션 필터의 비율을 증가시킴
    - 이러한 컨볼루션 필터는 더 넓은 영역에서 정보를 수집할 수 있으며, 더 높은 레이어에서 요구되는 더 넓은 컨텍스트를 포착하는데 적합함
    - 큰 필터를 사용함으로써, 모델은 더 넓은 영역의 패턴과 상관 관계를 학습할 수 있으며
      이는 복잡한 이미지 내에서 더 높은 수준의 특징을 효과적으로 인식하는데 도움이 됨
    - 네트워크가 깊어질수록 변화하는 출력의 상관 관계 특성과 공간적 집중도를 고려하여, 네트워크가 다양한 스케일의 특징을 효과적으로 학습하고 통합할 수 있도록 함
    - 결과적으로 Inception 모듈은 깊이가 증가함에 따라 다양한 크기의 컨볼루션 필터를 조합하여 사용함으로써, 신경망의 학습능력과 정확도를 향상시킬 수 있음
- train 중에 메모리 효율성으로 인해, 낮은 layer들은 기존의 convolution 방식을 유지하고, 높은 layer들은 inception module을 사용함
- 실질적으로 유용한 측면이 있는데 다양한 시각정보를 다양한 규모로 처리 한 뒤에 통합하여야 한다는 직관과 일치하며 계산량이 상당히 조절되어 어려움 없이 width와 depth를 늘릴 수 있음

GoogleNet

type	patch size/ stride	output size	depth	#1×1	#3×3 reduce	#3×3	#5×5 reduce	#5×5	pool proj	params	ops
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		64	192				112K	360M
max pool	3×3/2	28×28×192	0								
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
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								

Table 1: GoogLeNet incarnation of the Inception architecture

- inception modules 내부를 포함한 모든 Convolution Layer는 ReLU를 사용함
- Network의 receptive field는 224 x 224로 RGB Channel의 평균으로 subtraction 함
- 3 × 3 reduce와 5 x 5 reduce는 3 x 3 및 5 x 5 convolution 이전에 사용 된 reduction layer의 filter수를 나타냄
- Maxpooling 후 projection layer에서 1 x 1 filter 수를 볼 수 있음