

Xception: Deep Learning with Depthwise Separable Convolutions

Motivation

- Deep Neural Network의 성능을 향상시키기 위한 가장 간단한 방법

→ 모델의 **SIZE**를 증가 (increase the depth and width of the model)

- 두가지의 문제점이 발생

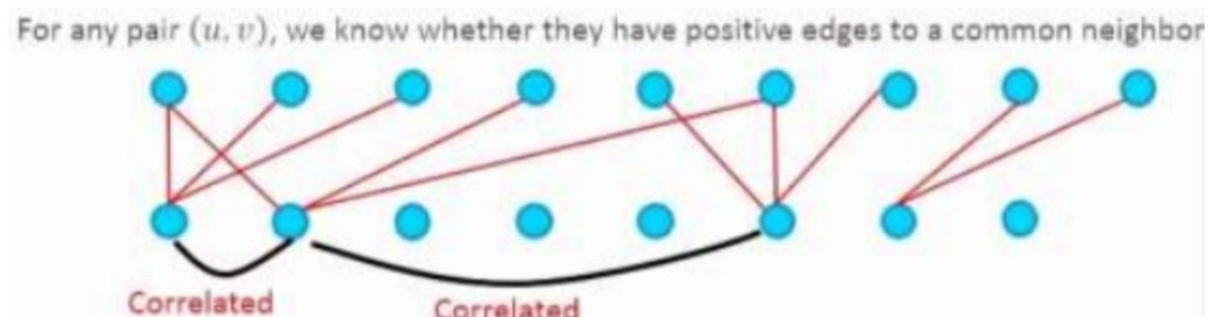
1. Overfitting이 되기 쉬움
2. 계산 자원(computation resource)이 증가함

- 이를 해결하기 위해

1. Fully connection architecture에서 Sparsely connected architecture로 만들자

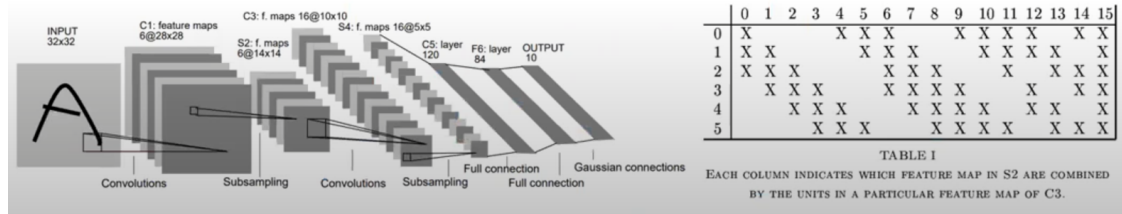
Sparsely connected architecture란?

- 전에 있는 input node들 사이의 correlation이 상당히 높다 하면 Fully connected처럼 다 연결된 것이 아니라 지난 output의 서로 연관관계가 높으면 둘은 연결시켜주되 나머지는 연결시키지 않는 것을 말함



- Cluster가 생기고, Sparse한 connection의 architecture가 됨
- But Sparsely connected architecture도 문제점이 발생함
 - Sparse matrix computation is very inefficient
 - dense matrix calculation is extremely efficient
 - (sparse matrix보다 dense matrix가 더 효과적으로 발전함)
 - 오늘날 컴퓨팅 인프라는 균일하지 않는 sparse 데이터 구조에 대해 수치 계산에 있어서 매우 비효율적

ex) Even ConvNet changed back from sparse connection to full connection for better optimize parallel computing



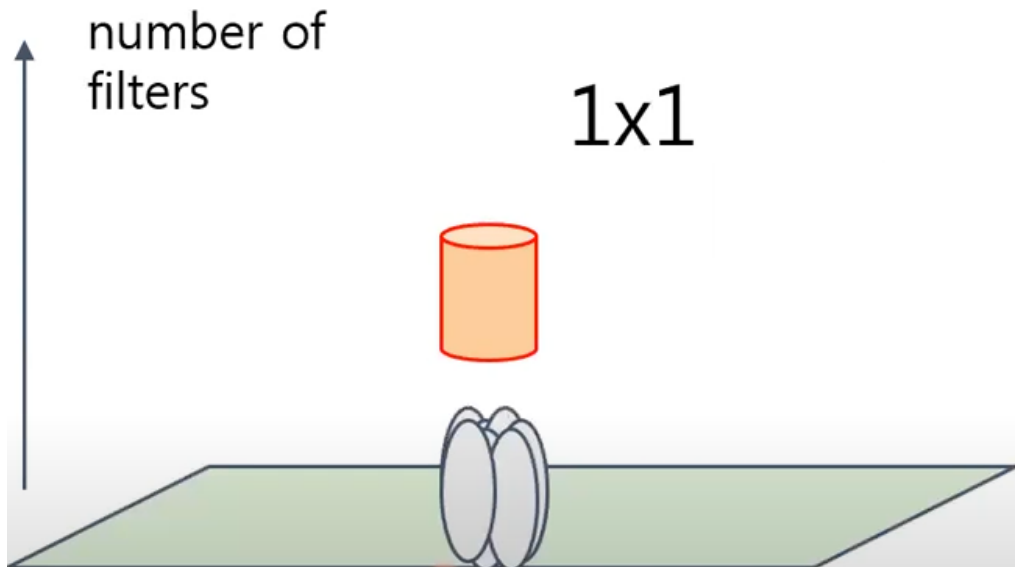
→ Fully connection architecture과 Sparsely connected architecture의 중간 단계는 없는가?

Inception

- Optimal local construction을 찾고 이걸 공간적으로 반복하면 어떨까라는 것이 Inception module의 naive버전
- In images, correlations tend to be local



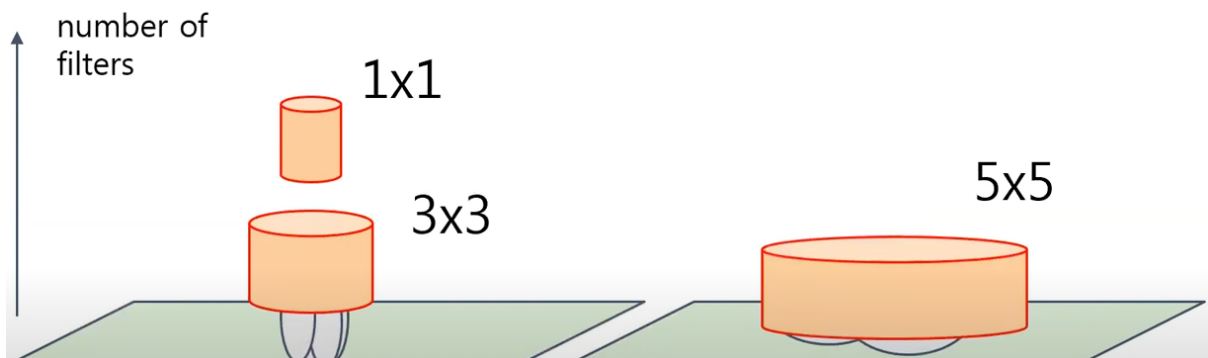
- Cover very local cluster by 1x1 convolutions



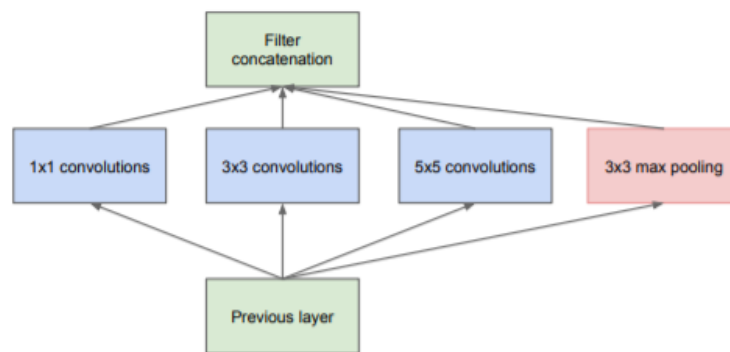
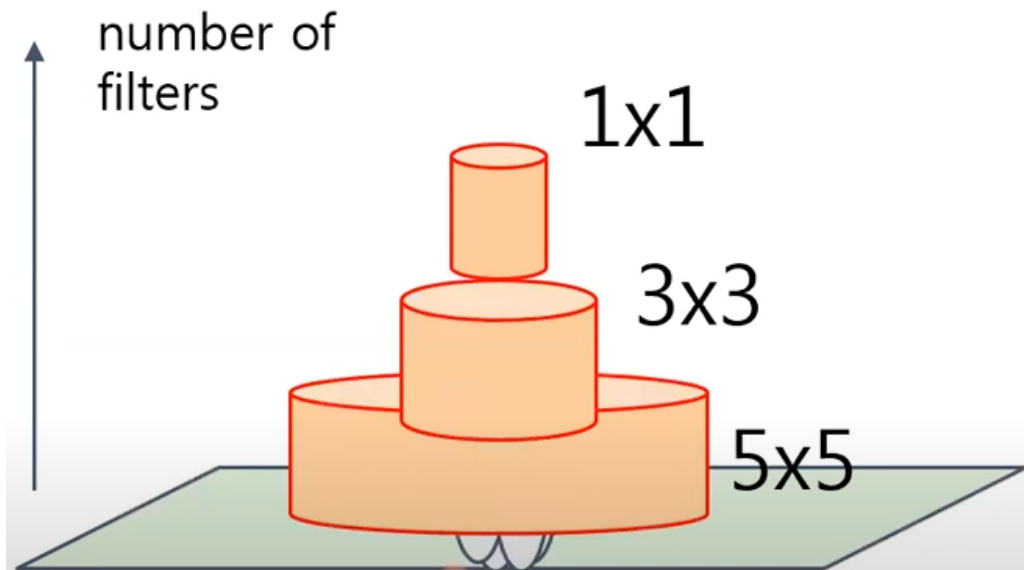
- Cover more spread out clusters by 3x3 convolutions



- Cover more spread out clusters by 5x5 convolutions

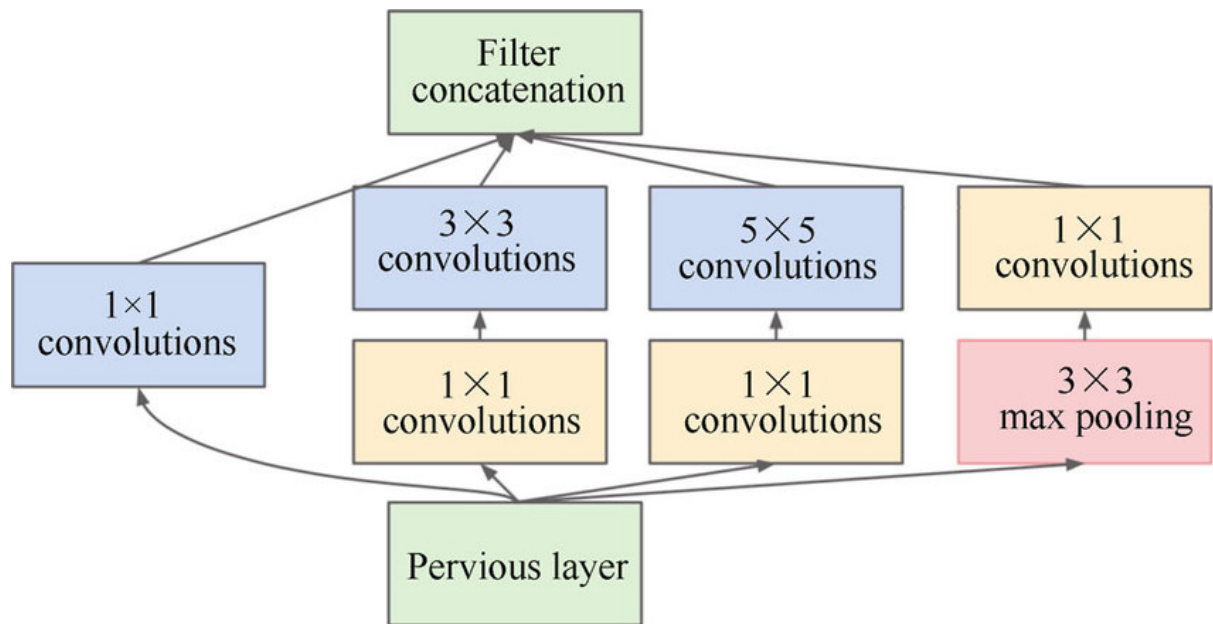


Inception Module naive version



- 이미지에서 local끼리는 correlation이 크니까 1x1 filter을 이용
- 멀리 떨어진 correlation 있는 것들의 cluster를 3x3, 5x5 filter 이용
- 3x3 max pooling은 이 당시에 성능이 좋아서 추가
- but 잘 작동되지 않음, 성능 저하
 - 풀링층과 conv층 출력을 합치는 것은 출력의 수 증가
 - 최적의 희소성 구조를 커버 가능 → but, 매우 비효율적
- 연산량을 줄이자고 했지만 오히려 증가

Inception Module



▼ 1x1 Convolution

- x,y의 값을 건드리지 않고 channel 수만 줄여줌
- Increase the representational power of nural Network
- dimension reduction의 역할

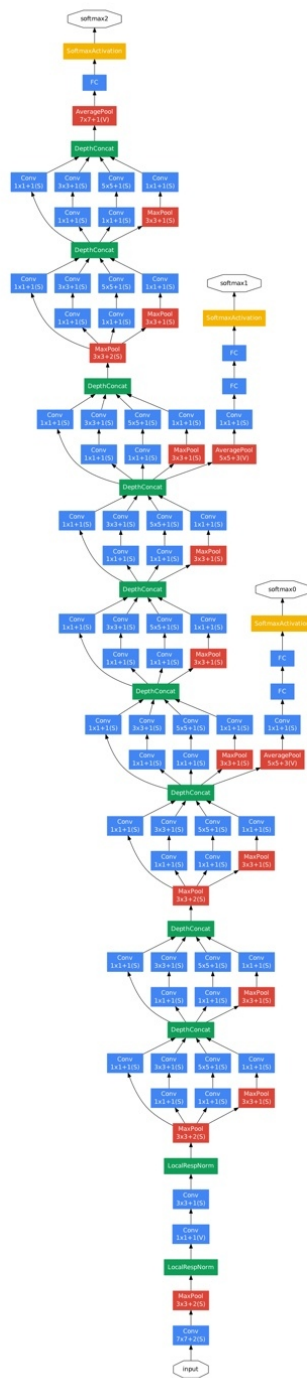
- 1x1 convolutions를 사용하여 naive version의 문제점을 해결

→ 연산해야 하는 노드들을 1x1 filter로 줄여주고 계산량이 높은 Convolution들을 계산

→ dimension reduction이 일어남

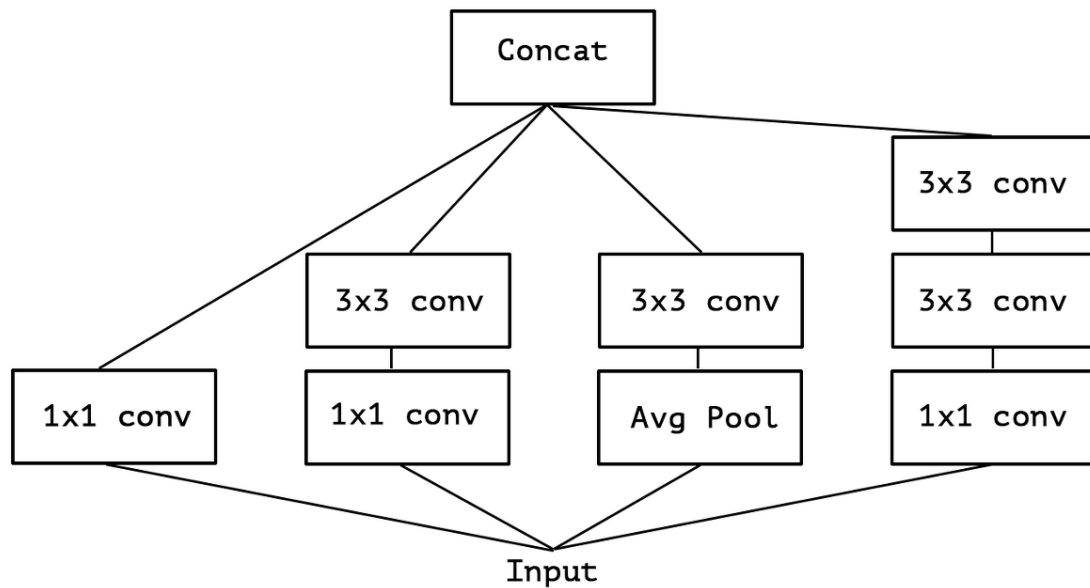
→ maxpooling은 maxpooling은 채널 수를 조절할 수 있는 방법이 없어서 나중에 조절

▼ Inception Module을 사용하여 만들어진 GoogleNet



Xception

Inception V3



- Inception V1 → Inception V3 ; $5 \times 5 \rightarrow 3 \times 3 + 3 \times 3$
- $5 \times 5 = 25 > 3 \times 3 + 3 \times 3 = 18 \Rightarrow$ Inception V3가 더 효율적

저자가 해석한 Inception

- 기존의 Convolution layer은 filter를 가지고 3D space(Height, Width, Channel)를 모두 학습하려고 함, 하나의 kernel(filter)로 cross-channel correlation과 spatial correlation을 동시에 mapping 함

→ 이러한 이유로 성능이 안 좋음

- Inception Module 아이디어는 cross-channel correlation과 spatial correlation을 독립적으로 살펴볼 수 있게 함으로써 이 프로세스를 좀더 쉽고 효율적으로 만듦
- 1×1 convolution을 통해 cross-channel correlation을 학습하고, 이후 3×3 , 5×5 convolution을 통해서 spatial correlation을 학습함.

- cross-channel correlation: 입력 채널들 간의 관계 학습
 - 1×1 convolution(=pointwise convolution)을 통해서 학습 가능
- spatial correlation: filter와 특정 채널 사이의 관계 학습 (공간적인 특성 학습)

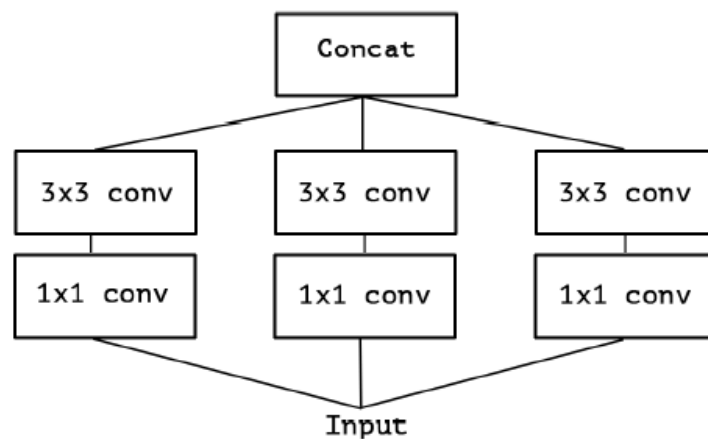
→ 저자는 Single convolution kernel(filter) 하나가 하려는 것을 Spatial correlation(3×3 , 5×5)을 분석해주면서 cross-channel correlation(1×1)으로 두 가지 역할을 잘 분산해주기 때문에 Xception 저자는 Inception이 잘 된 것이라고 생각함

Inception hypothesis

- 저자의 가설
 - "cross-channel correlation과 spatial correlation의 mapping은 완전히 분리될 수 있다. "
- Xception은 완벽히 cross-channel correlations와 spatial correlations를 독립적으로 계산하고 mapping하기 위해 고안된 모델
- Inception Architecture의 기초가 되는 가설의 더 강력한 버전이기 때문에 "**Extreme Inception**"을 의미하는 **Xception**이라고 부름

1) 먼저 Inception Module을 단순화시킴

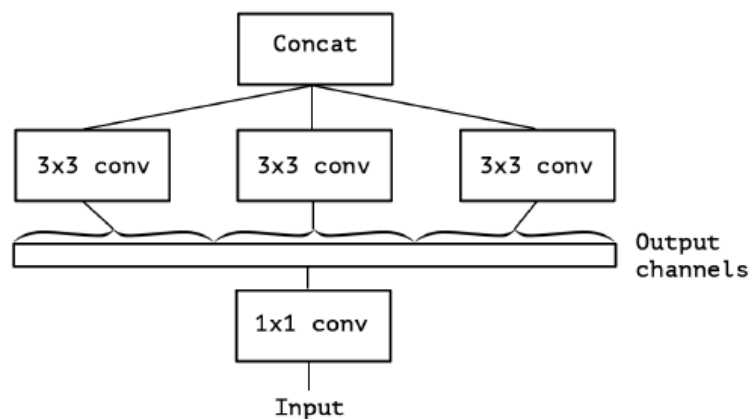
Figure 2. A simplified Inception module.



- 하나의 크기의 Convolution만 사용하고 averaging pooling을 포함하지 않는 단순화된 Inception module을 만들

2) 해당 Inception module을 large 1x1 convolution으로 재구성하고 output channel이 겹치지 않는 부분에 대해서 spatial convolution(3x3)이 오는 형태로 재구성함

Figure 3. A strictly equivalent reformulation of the simplified Inception module.



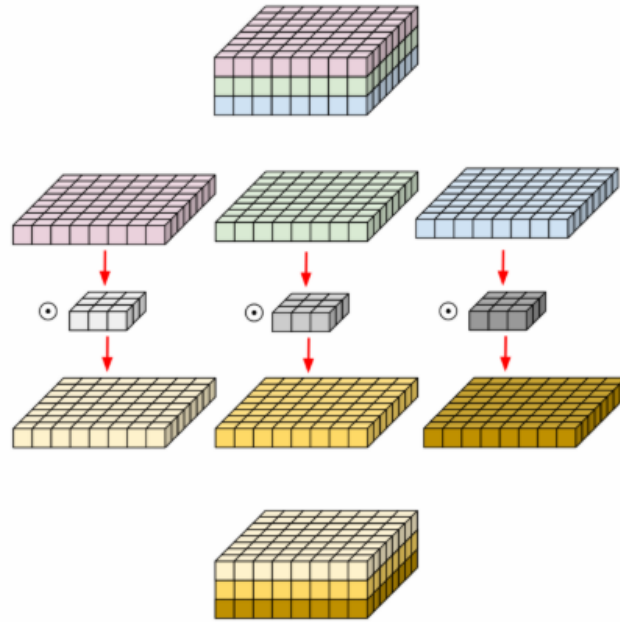
- Figure2와 Figure3은 서로 동일한 형태

(branch 1은 input에 대해 1x1 conv를 수행하고, output channels에 대해서 3x3 convolution을 수행하는데, branch 2, branch 3 역시 동일한 과정을 거치고, 마지막으로 각 결과를 concat)

- 1x1 conv는 cross-channel correlation을 계산하고, 3x3은 spatial correlations를 수행

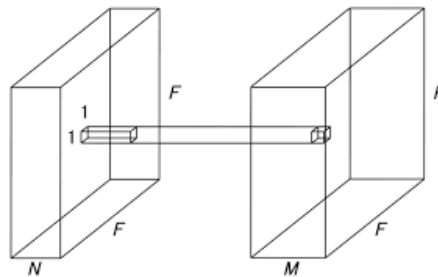
Depthwise Separable Convolution

Depthwise Convolution



- 위 처럼 $H \times W \times C$ 의 conv output을 C 단위로 분리하여 각각 conv filter를 적용하여 output을 만들고 그 결과를 다시 합치면 conv filter가 **훨씬 적은 파라미터**를 가지고서 동일한 크기의 아웃풋을 낼 수 있음

Pointwise Convolution



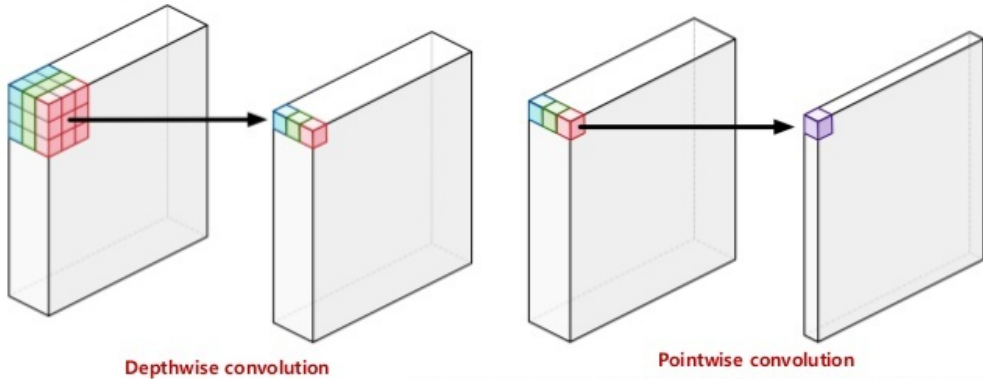
- 흔히 1x1 Conv라고 불리는 필터
- 주로 기존의 matrix의 결과를 논리적으로 다시 shuffle해서 뽑아내는 것을 목적으로함
- 총 channel수를 줄이거나 늘리는 목적으로도 많이 사용함

Depthwise Separable Convolution

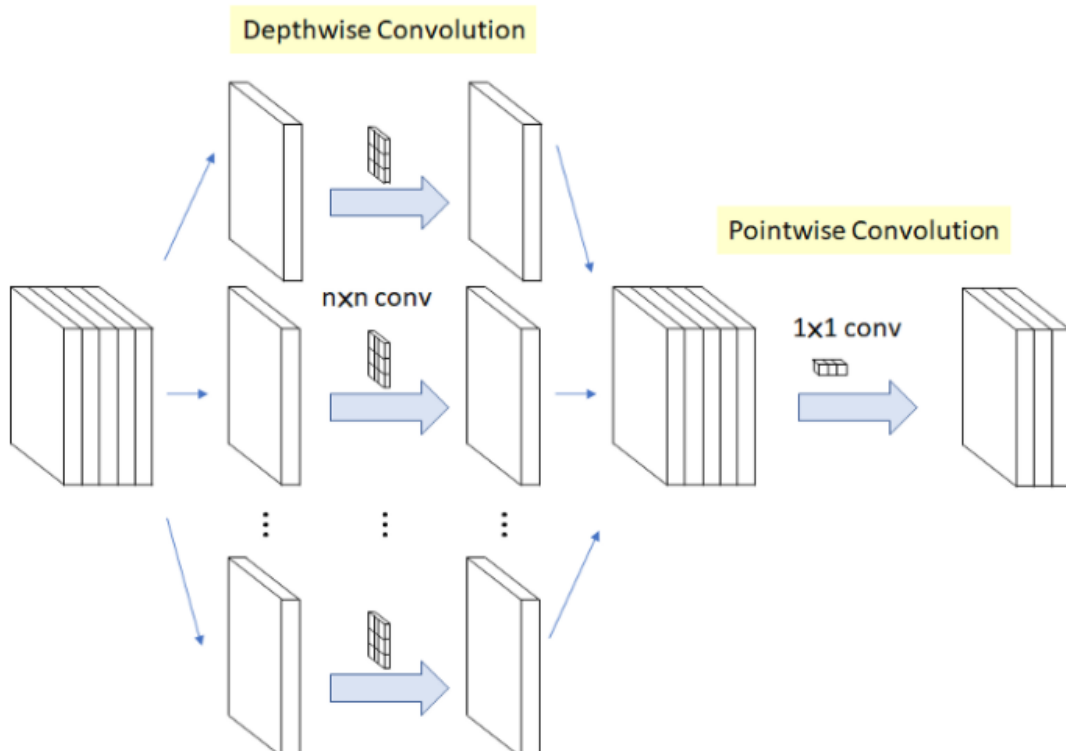
- Inception 모듈과 작동방식이 비슷한 Depthwise Separable Convolution

Depthwise Separable Convolution

- Depthwise Convolution + Pointwise Convolution(1x1 convolution)



- Depthwise convolution을 먼저 수행한 후 Pointwise convolution을 수행
- 3x3의 필터를 통해 conv 연산도 진행하고, 서로 다른 channel들의 정보도 공유하면서 동시에 **파라미터 수도 줄일 수 있음**

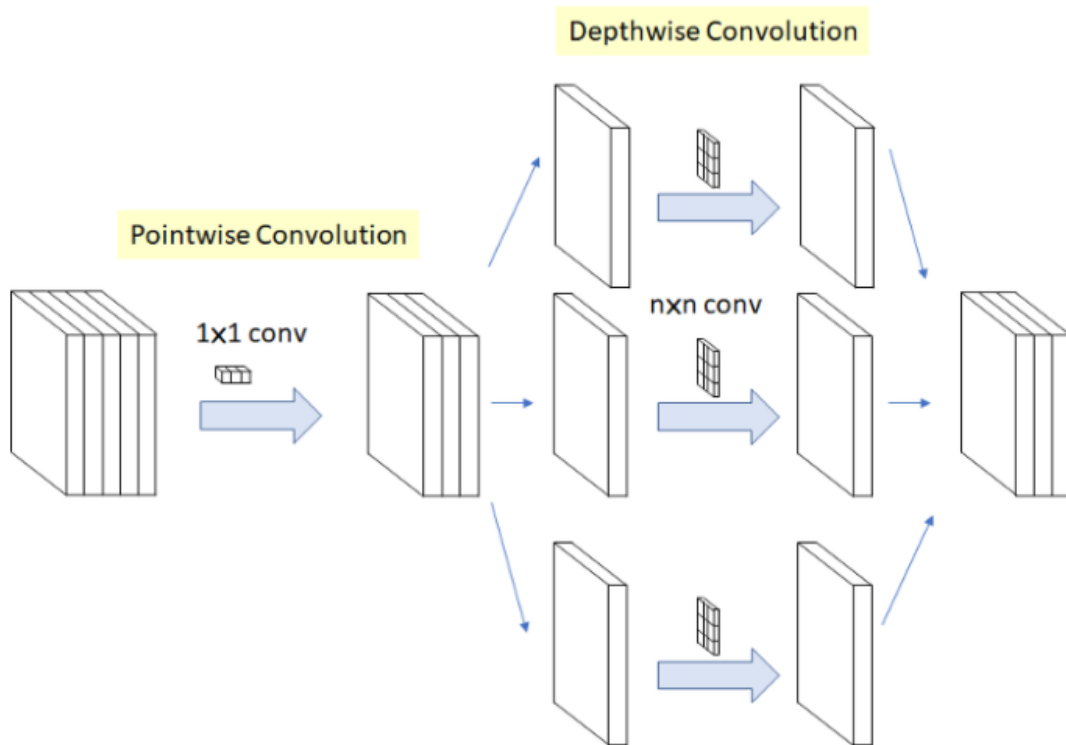


- **Depthwise Convolution**은 입력 채널 각각에 독립적으로 3x3 conv를 수행함
- 입력 채널이 5개이면 5개의 3x3 conv가 연산을 수행하여, 각각 입력값과 동일한 크기 피쳐맵을 생성

- 각 피쳐맵을 연결하여 5개 채널의 피쳐맵을 생성
- **Pointwise Convolution**은 모든 채널에 **1x1 conv**를 수행하여, 채널 수를 조절하는 역할

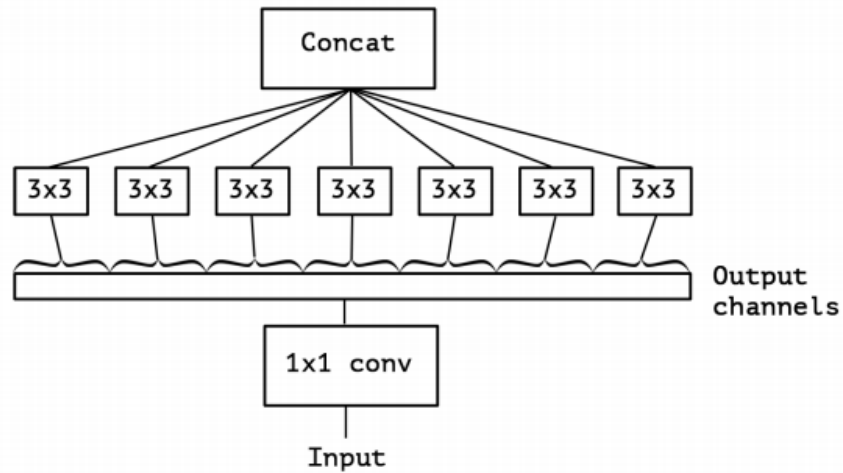
Modified Depthwise Separable Convolution(Extreme Inception)

- Xception은 Depthwise Separable Convolution을 수정해서 inception 모듈 대신에 사용



- Inception 모듈보다 효과적으로 cross-channels correlations와 spatial correlations를 독립적으로 계산할 수 있음

Figure 4. An “extreme” version of our Inception module, with one spatial convolution per output channel of the 1x1 convolution.



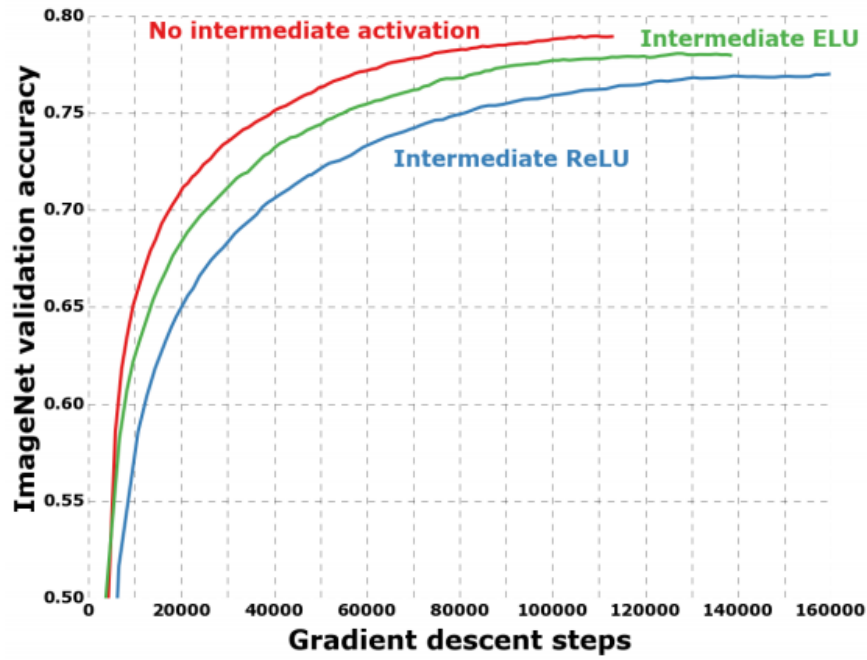
- 입력값에 1x1 conv를 수행하여 채널 수를 조절 → 채널 수는 n개의 segment로 나뉘지며, 이 n은 하이퍼파라미터임
- 두 방향(channel wise, spatial)에 대한 mapping을 완전히 분리할 수 있음

ex)

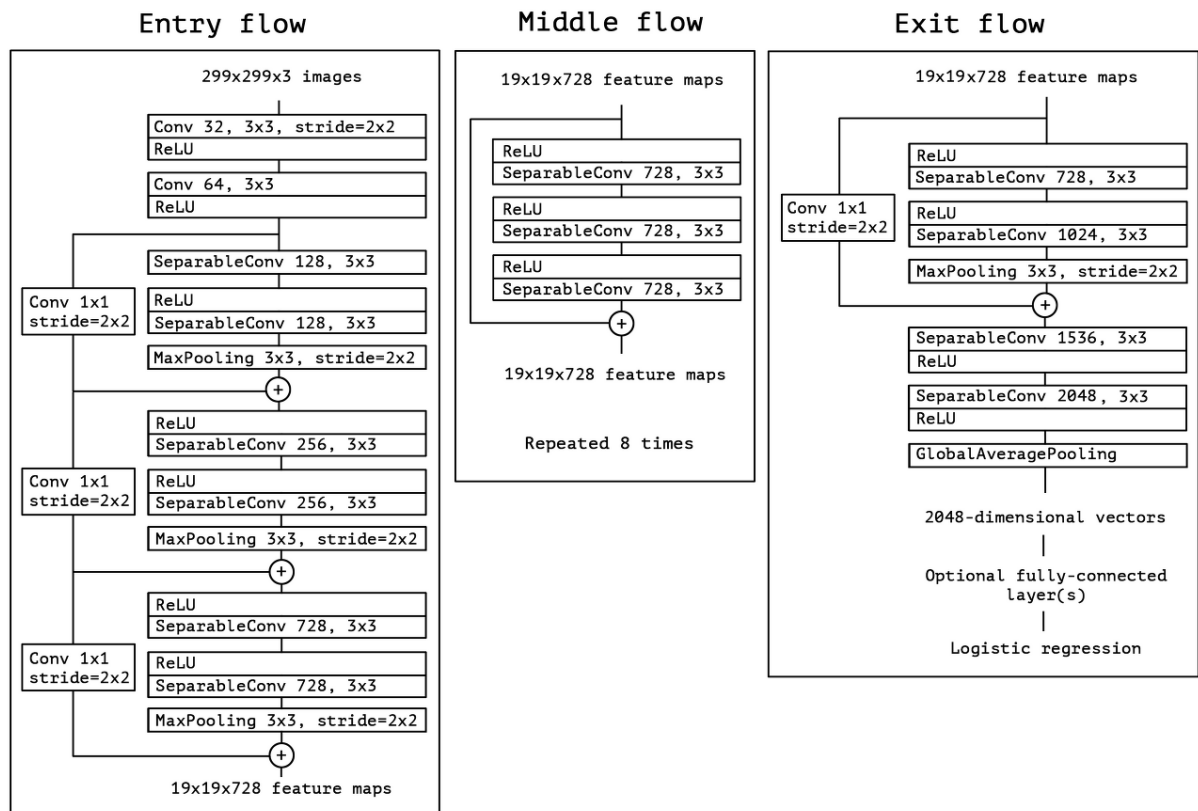
1. 100개의 채널 수가 3~4개의 segment로 나뉘짐
2. 나뉜 segment 별로 depthwise convolution(3x3 conv)를 수행
3. 각 출력값은 concatenate 됨

Xception vs Depthwise separable convolution

1. 연산의 순서가 다름
 - 기존 depthwise separable convolution은 depthwise convolution(3x3 conv)를 먼저 수행하고 pointwise convolution(1x1 conv)를 수행
 - 수정된 버전은 pointwise convolution(1x1 conv)를 수행하고, depthwise convolution(3x3 conv)를 수행
2. 비선형 함수의 존재 유무
 - Xception Module은 1x1 → ReLU → 3x3이고, Depthwise: 1x1이랑 spatial convolution 사이에 ReLU 같은 활성화 함수가 들어가지 않음 (전체구조 참고)



전체 Xception 구조



- Xception은 14개 모듈로 이루어져있고, 총 36개의 convolutional layer가 존재

- 그리고 residual connection을 사용
- 입력값은 Entry flow 거치고 middle flow를 8번 거쳐서 exit flow를 통과

Result

	Top-1 accuracy	Top-5 accuracy
VGG-16	0.715	0.901
ResNet-152	0.770	0.933
Inception V3	0.782	0.941
Xception	0.790	0.945

	FastEval14k MAP@100
Inception V3 - no FC layers	6.36
Xception - no FC layers	6.70
Inception V3 with FC layers	6.50
Xception with FC layers	6.78

	Parameter count	Steps/second
Inception V3	23,626,728	31
Xception	22,855,952	28

- Inception V3와 비교 실험 상 성능이 marginally 하게 더 높게 나옴을 알 수 있음
- 파라미터 수를 동일하게 맞춰줬기 때문에 순전히 모델의 차이라고 말함
- ImageNet에 대해 Inception과 Xception의 성능이 marginal 하게 차이가 나지만 Inception은 ImageNet에 Overfitting 된 느낌이 있는거에 비해 Xception은 다른 데이터셋에서도 좋은 성능을 보임
- Inception은 노드간의 연결을 줄이는데 힘썼다면, Xception은 채널간의 관계를 찾는 것과 이미지의 지역정보를 찾는 것을 완전히 분리하고자 하는데 목적을 둬

참고자료

- 유튜브

PR-034: Inception and Xception

Introduction to Inception and Xception Slide:

<https://www.slideshare.net/thinkingfactory/pr12-inception-and-xception-jaejun-yoo>Papers:Going Deeper with Convo...

▶ https://www.youtube.com/watch?v=V0dLhyg5_Dw

PR12와 함께 이해하는

Inception & Xception
(GoogleNet)

Jaejun Yoo
Ph.D. Candidate @KAIST
PR12
10th Sep, 2017

[PR12] Inception and Xception - Jaejun Yoo

1. Inception & Xception PR12와 함께 이해하는 Jaejun Yoo Ph.D. Candidate @KAIST PR12
10th Sep, 2017 (GoogLeNet) 2. Today's contents GoogLeNet : Inception models * Going
Deeper with Convolution * Rethinking the Inception Architecture for Computer Vision *

 <https://www.slideshare.net/thinkingfactory/pr12-inception-and-xception-jaejun-yoo>

PR12와 함께 이해하는


Inception & Xception
(GoogLeNet)

Jaejun Yoo
Ph.D. Candidate @KAIST
PR12
10th Sep, 2017

• Inception

[딥러닝 모델 경량화] Inception

안녕하세요! 저번 포스팅에서 딥러닝 모델 경량화 동향을 살펴보았을 때 합성곱 필터의 변경해서 만든 모델 중 하나인 MobileNet을 봤었죠? 이에 대해서 더 자세히 공부하려고 MobileNet 논문을 보는데 Inception, Xception 모델을 먼저 공부하고서 봐야 할 것 같더라고요! 이번 글에서는


 <https://sotudy.tistory.com/13>



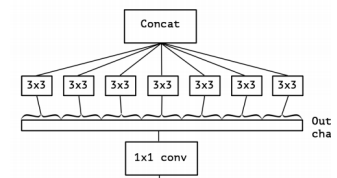
• Xception

[논문 읽기] Xception(2017) 리뷰, Deep Learning with Depthwise Separable Convolutions

이번에 읽어볼 논문은 Xception: Deep Learning with Depthwise Separable Convolutions 입니다. Xception은 Inception 모듈에 대한 고찰로 탄생한 모델입니다. Xception은 완벽히 cross-channel correlations와 spatial correlations를 독립적으로 계산하기 위해 고안된 모델입니다. 이를 위해 새로운


 <https://deep-learning-study.tistory.com/529>

spatial convolution per output channel of the 1x1 convol



[딥러닝 모델 경량화] Xception


안녕하세요! 오늘은 Inception으로부터 발전한 Xception에 대해서 알아보도록 하겠습니다! Inception에 대한 자세한 설명은 지난 글에서 확인하실 수 있습니다. 2020/08/02 - [Deep Learning/papers] - [딥러닝 모델 경량화] Inception 앞에서 말했듯이 Xception은 Inception을

 <https://sotudy.tistory.com/14>



[Classification] Deep Learning with Depthwise Separable Convolutions : Xception 논문 리뷰

투빅스 14기 장혜림 Xception : eXtreme Inception 기존 Inception 1x1 convolution을 통해 cross-channel correlation을 학습하고, 이후 3x3, 5x5 convolution을 통해서 spatial correlation을 학습한다. cross-channel correlation: 입력 채널들 간의 관계 학습 1x1 convolution(=pointwise convolution)을 통해서 학습 가능


 <https://vlog.io/@mink7878/classification-Xception-Deep-Learning-with-Depthwise-Separable-Convolutions>

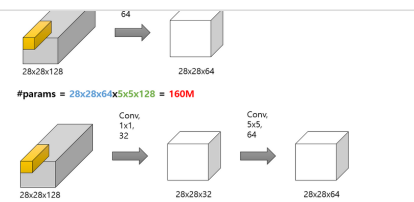


• BASE

1x1 convolution이란,

GoogLeNet 즉, 구글에서 발표한 Inception 계통의 Network에서는 1x1 Convolution을 통해 유의미하게 연산량을 줄였습니다. 그리고 이후 Xception, Squeeze, Mobile 등 다양한 모델에서도 연산량 감소를 위해 이 방법을 적극적으로 채택하고 있습니다. 뿐만 아니라 Semantic

 <https://hwiyong.tistory.com/45>



CNN, Convolutional Neural Network 요약

Fully Connected Layer 만으로 구성된 인공 신경망의 입력 데이터는 1차원(배열) 형태로 한정됩니다. 한 장의 컬러 사진은 3차원 데이터입니다. 배치 모드에 사용되는 여러장의 사진은 4차원 데이터입니다. 사진 데이터로 전연결(FC, Fully Connected) 신경망을 학습시켜야 할 경우에, 3차원

<http://taewan.kim/post/cnn/>



Convolution vs. Cross-Correlation

This post will overview the difference between convolution and cross-correlation. This post is the only resource online that contains a step-by-step worked example of both convolution and cross-correlation together (as far as I know - and trust me, I did a lot of

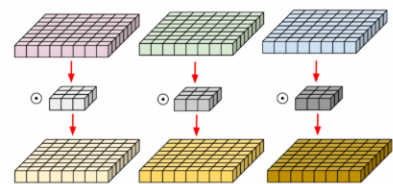
<https://glassboxmedicine.com/2019/07/26/convolution-vs-cross-correlation/>



Depthwise Separable Convolution 설명 및 pytorch 구현

우선 Depth-wise Seperable Convolution에 대한 설명을 하기에 앞서 Depth-wise Convolution에 대한 설명을 먼저 할까 한다. 기본적인 개념은 쉽다. 위 처럼 $H*W*C$ 의 conv output을 C단위로 분리하여 각각 conv filter를 적용하여 output을 만들고 그 결과를 다시 합치면 conv filter가 훨씬

<https://wingnim.tistory.com/104>



• Code

<https://github.com/Hyunjulie/KR-Reading-Computer-Vision-Papers>

pytorch-Xception/Xception_pytorch.ipynb at master · hoyao12/pytorch-Xception

Simple Code Implementation of "Xception" architecture using PyTorch. - pytorch-Xception/Xception_pytorch.ipynb at master · hoyao12/pytorch-Xception

https://github.com/hoyao12/pytorch-Xception/blob/master/Xception_pytorch.ipynb

hoyao12/**pytorch-Xception**

Simple Code Implementation of "Xception" architecture using PyTorch.

1 Contributor 0 Issues 13 Stars 2 Forks

