X AI

contents

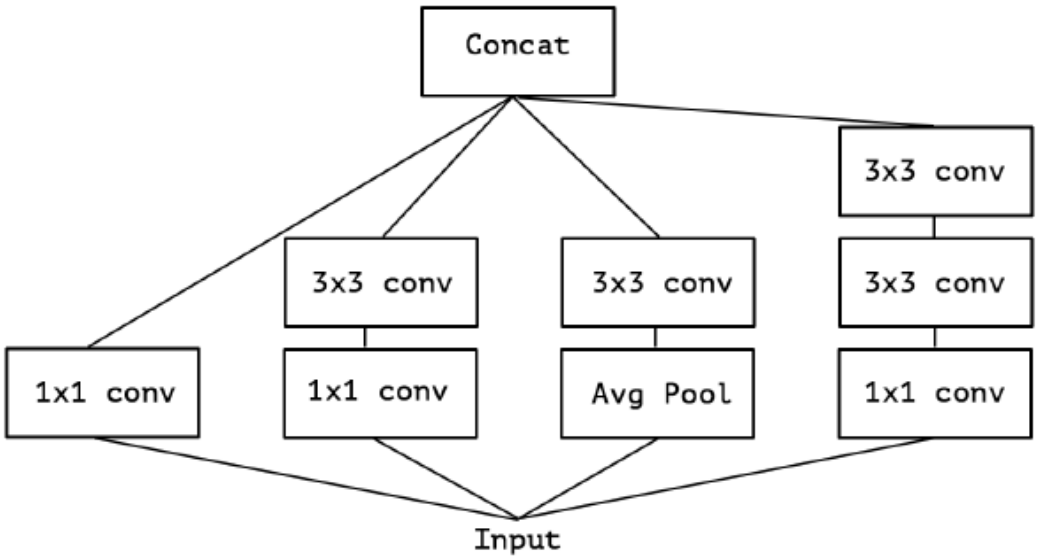
Xception : Deep Learning with Depthwise Separable Convolutions

Franc, ois Chollet Google, Inc.

Introduction

Convolutional Neural Network 설계의 역사는 피쳐 추출과 공간 서브샘플링을 위한 최대 풀링 연산을 위한 단순한 Convolution 스택인 LeNet-style 모델로 시작했다. 그 사이에 여러 모델을 거치며 2014년 Szegdy에 의해 GoogLeNet(Inception V1)으로 소개된 인셉션 아키텍처는 인셉션 V2, V3 그리고 ResNet19로 개선됐다. **인셉션은 JFT에서 사용중인 내부 데이터셋뿐만 아니라 ImageNet 데이터셋에서 가장 성능이 우수한 모델 제품 중 하나이다.**

Figure 1. A canonical Inception module (Inception V3).



(Inception V3 module은 다음 그림과 같습니다. Inception v1과 다른 점은 5x5→3x3+3x3으로 바뀐 것입니다. 5x5를 사용하는 것보다 3x3을 두 번 사용하는 것이 더 효과적이기 때문입니다.)

인셉션 스타일 모형의 기본 구성 요소는 인셉션 모듈이며, 그 중 몇 가지 다른 버전이 존재한다. 위의 그림 1에서는 Inception V3 아키텍쳐에서 볼 수 있는 Inception 모듈이 표준 형식입니다. 인셉션 모델은 이러한 모듈의 스택으로 이해할 수 있습니다. **인셉션 모듈은 개념적으로 컨볼루션과 유사하지만, 경험적으로 더 적은 매개 변수로 더 풍부한 표현을 배울 수 있는 것으로 보입니다.**

인셉션 모듈은 이전 단계의 활성화 지도에 다양한 필터 크기(Kernel_Size)로 합성곱 연산을 적용하는 방식입니다. 쉽게 표현하면, 강아지 사진에서 귀, 코 , 눈 등의 특징을 다른 방향으로 보는 것입니다. 다른 방향에서 보기 때문에, 같은 강아지 사진에서 다른 특성들을 추출할 수 있습니다. 인셉션은 적은 파라미터로 다양한 특징값을 추출하는데 의미가 있습니다

Inception hypothesis

인셉션의 기본 가설은 **“cross-channel correlations과 spatial correlations가 충분히 분리되어 있으므로 이들을 공동으로 매핑하지 않는 것이 바람직하다**.” 입니다.

그래서 1x1 conv (cross-channel correlations) 이후에 3x3 or 5x5 conv(spatial correlations) 연산이 수행되는 구조로 구성돼있습니다. 이를 cross-channel correlations와 spatial correlations를 독립적으로 수행한다고 합니다. 1x1 conv는 cross-channel correlation을 계산하고, 3x3은 spatial correlations를 수행하는 것입니다.

- cross-chanel correlation : 입력 채널들 간의 관계 학습

  1x1 convolution(=pointwise convolution)을 통해서 학습
- spatial correlation : filter와 특정 채널 사이의 관계 학습 (공간적인 특성 학습)

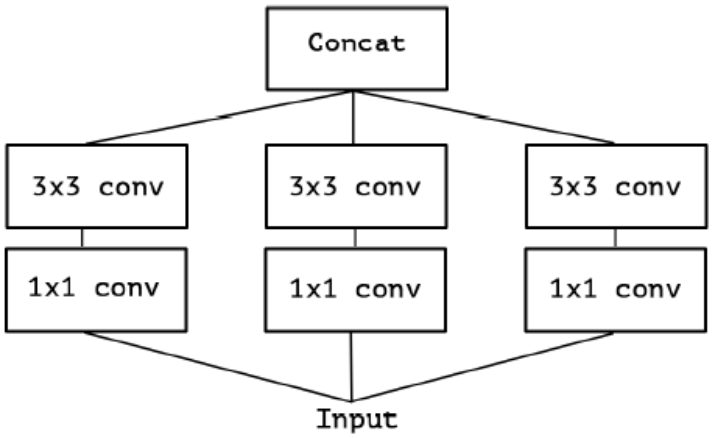
일반적인 convolution은 하나의 filter로 cross-channel correlation과 spatial correlation을 동시에 학습함. 하지만 I**nception은 이를 분산해서 서로가 매핑되지 않고 독립적으로 수행한다는 점이 Standard Convolution과 차별점입니다.**

이러한 인셉션을 좀 더 강하고 확실하게 해보자 하는 것이 바로 Xception입니다.

Xception

Xception은 완벽히 cross-channel correlations와 spatial correlations를 독립적으로 계산하고 mapping하기 위해 고안된 모델입니다. 그리고 Xception은 Depthwise Separable Convolution을 수정해서 Inception 모듈 대신에 사용합니다. 그리고 Extreme Inception이라고 부릅니다.

Figure 2. A simplified Inception module.



1) Inception module을 단순화 시킴

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

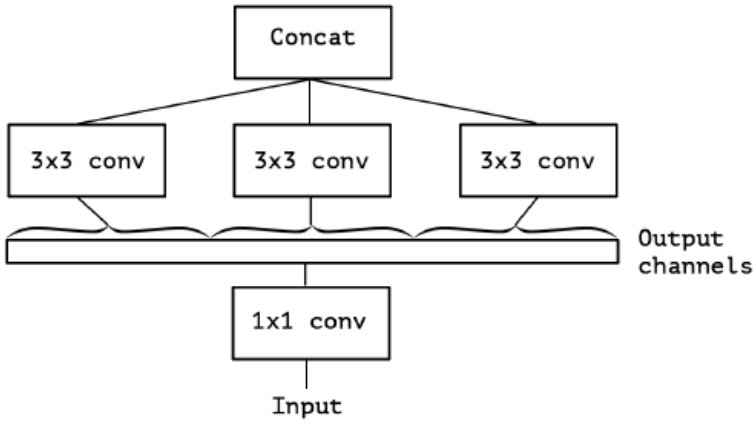


그림 2와 3은 서로 동일한 형태

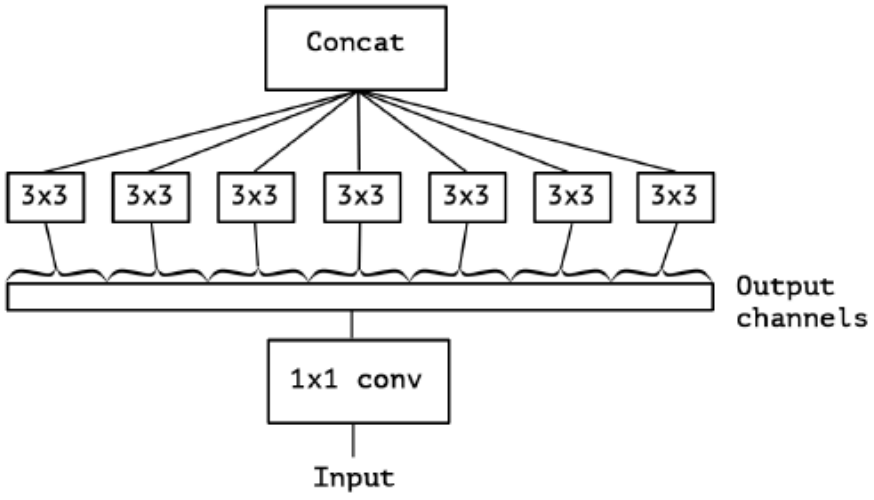
2) Inception module을 large 1x1 convolution으로 재구성하고 output channel이 겹치지 않는 부분에 대해서 spatial convolution(3x3)이 오는 형태로 재구성함. (먼저 채널간 상관 관계를 매핑하기 위해 1x1 컨볼루션을 사용한 다음 모든 출력 채널을 분리시켜서 output channel당 3x3 conv(공간 상관 관계)를 별도로 매핑합니다. 이렇게 함으로써 두 방향 (chanel wise, spatial)에 대한 mapping을 완전히 분리할 수 있습니다.

(branch 1은 input에 대해 1x1 conv를 수행하고, output channels에 대해서 3x3 convolution을 수행하는데, brahch2, branch 3 역시 동일한 과정을 거치고, 마지막으로 각 결과를 concat)

3) 기존에는 3,4개 branch로 나눠서 각 branch에 대해서 spatial convolution 연산을 수행한 후 concat하였다면, Xception은 3,4개로 나누는 것이 아니라 각 output channel에 대해서 spatial convolution을 수행함.

그렇게 함으로써 cross-channel correlation과 spatial correlation을 완전하게 분리시켜서 학습할 수 있다고 생각함 (가설)

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



- Xception이 학습하는 과정

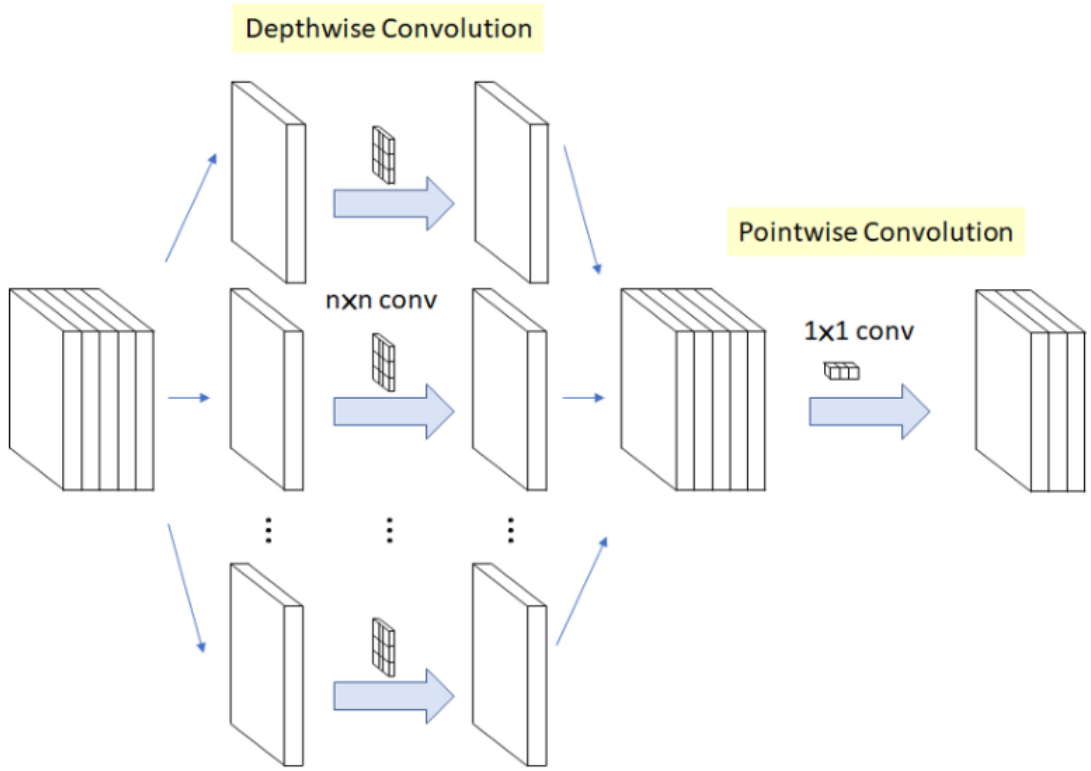
  1) 1x1 conv를 통해 cross-channel correlation을 학습

2) 각 output channel에 대해서 spatial correlation 학습

이러한 학습을 위해 Depthwise Separable convolution을 수정하여 사용함.

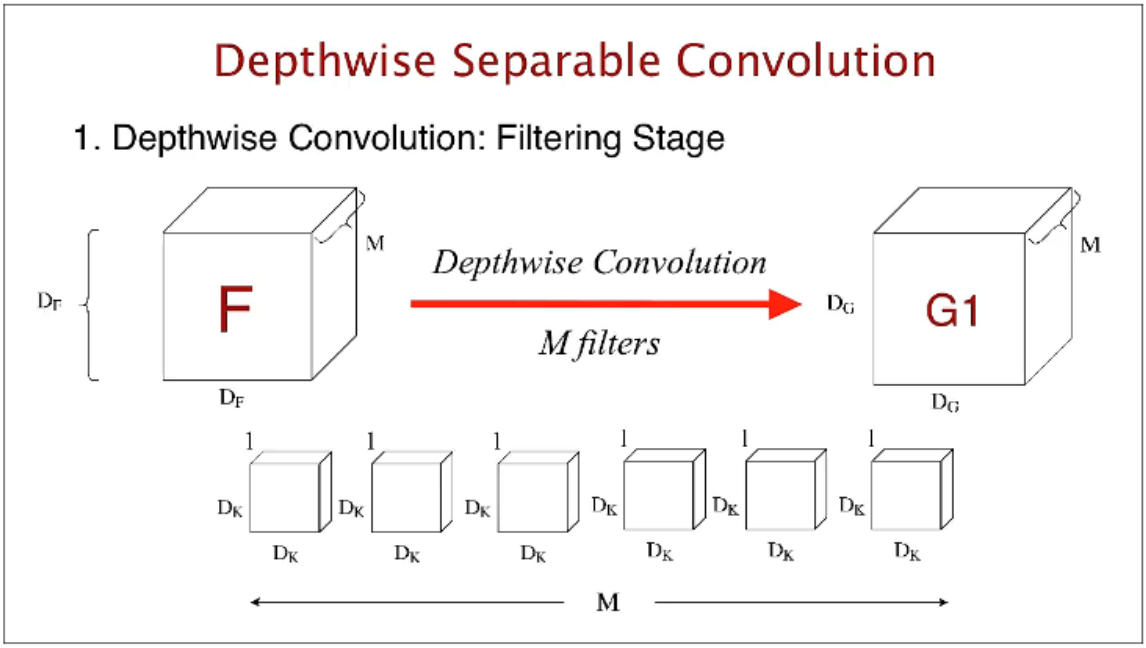
그럼 Depthwise Separanle Convolution을 살펴보도록 하겠다.

Depthwise Separable Convolution



Depthwise Convolution은 입력 채널 각각에 독립적으로 3x3 conv를 수행합니다. 입력 채널이 5개이면 5개의 3x3 conv가 연산을 수행하여, 각각 입력값과 동일한 크기 피쳐맵을 생성합니다. 그리고 각 피쳐맵을 연결하여 5개 채널의 피쳐맵을 생성합니다. **Pointwise Convolution은 모든 채널에 1x1 conv를 수행하여, 채널 수를 조절하는 역할을 합니다.** 이렇게 연산을 수행하면 연산량이 감소합니다.

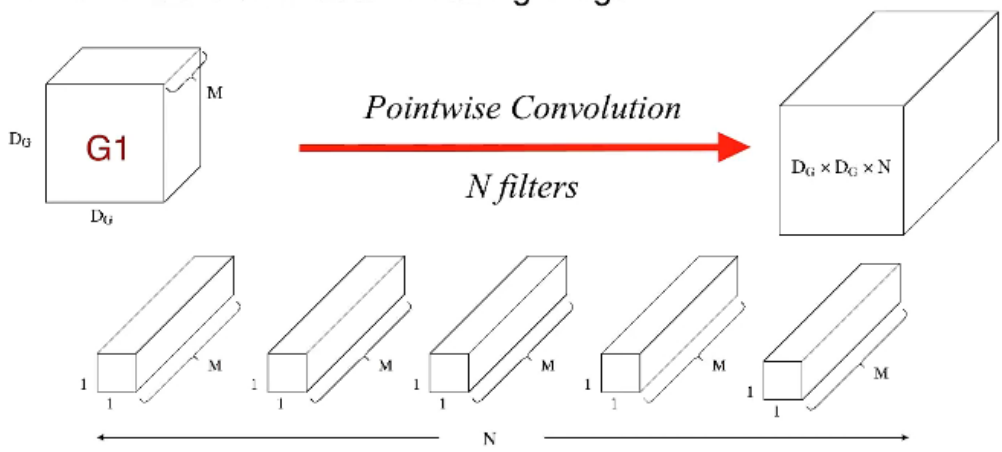
연산량은?



M개의 channel이 있으면 첫번째 channel인 (D_F, D_F, M_1)에 대해서 $(D_K, D_K, 1)$ filter가 존재. 최종적으로는 총 M개의 filter가 존재하게 됨. output volume = (D_G, D_G, M)

Depthwise Separable Convolution

2. Pointwise Convolution: Filtering Stage



depthwise convolution의 output인 (D_G, D_G, M)을 input으로 받아서 N개의 (1,1,M) filter를 사용하여 convolution 연산. output volume = (D_G, D_G, N)

Comparison Standard Vs. Depthwise

$$\frac{No.Mults\ in\ Depthwise\ Separable\ Conv}{No.Mults\ in\ Standard\ Conv} = \frac{M \times D_G^2\ (D_K^2 + N)}{N \times D_G \times D_G \times D_K \times D_K \times M}$$

$$\frac{No.\ Mults\ in\ Depthwise\ Separable\ Conv}{No.\ Mults\ in\ Standard\ Conv} = \frac{D_K^2 + N}{(D_K^2 \times N)} = \frac{1}{N} + \frac{1}{D_K^2}$$

$$N = 1{,}024 \qquad D_K = 3$$

$$\frac{No.\ Mults\ in\ Depthwise\ Separable\ Conv}{No.\ Mults\ in\ Standard\ Conv} = \frac{1}{1024} + \frac{1}{3^2} = 0.112$$

일반적인 convolution 연산량과 비교를 해보면, output channel의 수가 1024이고 filter의 size가 약 3x3일 때, 약 1/9만큼 연산량이 줄어들었음.

다시.. Xception

다시 Xception으로 돌아와서 Xception은 Depthwise Separable Convolution을 수정해서 inception 모듈 대신에 사용합니다. 그리고 **Extreme Inception**이라고 부릅니다. **아래 구조를 활용하면 Inception 모듈보다 효과적으로 cross-channels correlations와 spatial correlations를 독립적으로 계산할 수 있습니다.**

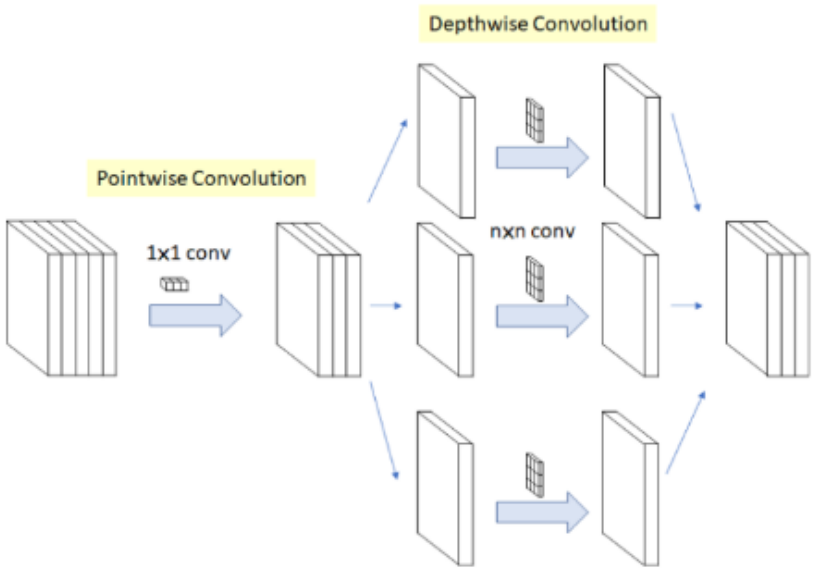
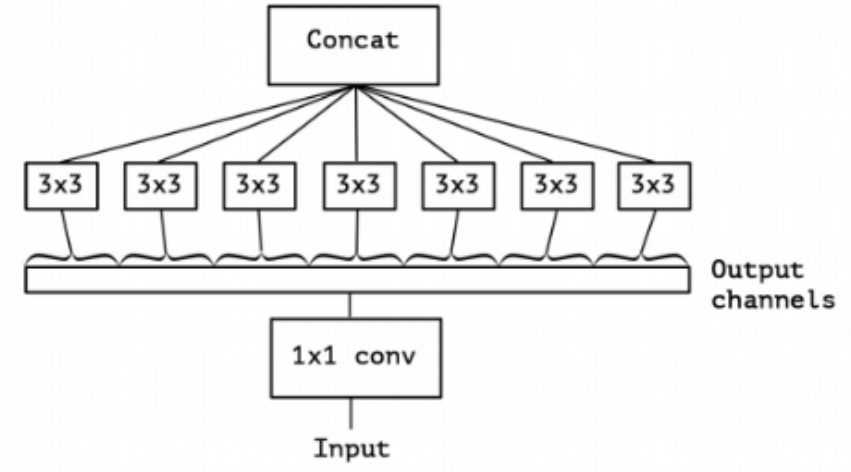


Figure 4. An "extreme" version of our Inception module, with one spatial convolution per output channel of the 1x1 convolution.



다시 정리하자면 Xception은 1x1 conv를 통해 cross-channel correlation 학습을 하고 각 output channel에 대해서 spatial convolution을 수행함

그러나, depthwise sepaarable convolution(depth → point) 순서대로 convolution을 수행해도 상관이 없다고함.(stacked setting에서 사용되기 때문에)

추후 Xception 코드를보면 tensorflow에서 제공하는 depthwise separable convolution을 그대로 사용하는 것을 볼 수 있음

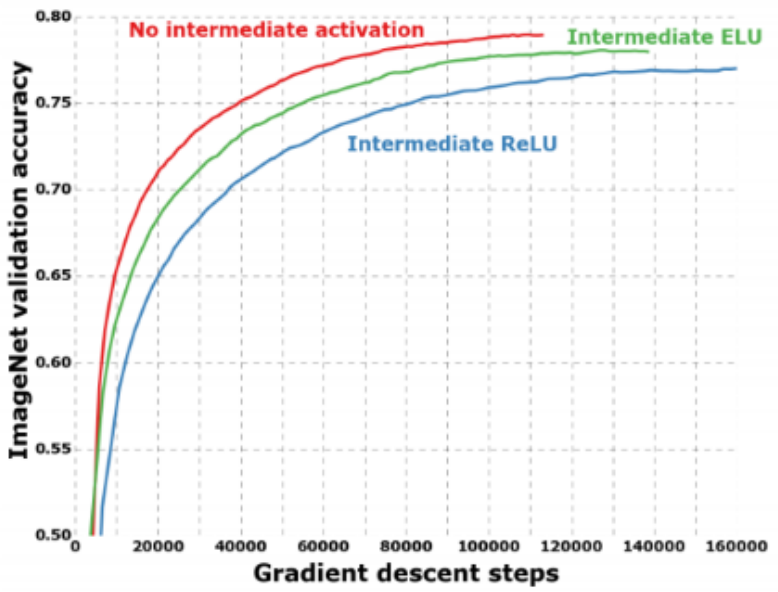
modified depthwise separable convoltuion with Xception

(1) 연산의 순서가 다릅니다.

기존 depthwise separable convolution은 depthwise convolution(3x3 conv)를 먼저 수행하고 pointwise convoltion(1x1 conv)를 수행합니다. 수정된 버전은 pointwise convoltuion(1x1 conv)를 수행하고, depthwise convolution(3x3 conv)를 수행합니다.

(2) 비선형 함수의 존재 유무입니다.

Inception 모듈은 1x1 conv 이후에 ReLU 비선형 함수를 수행합니다. 하지만 Xception에서 사용하는 모듈은 비선형 함수를 사용하지 않습니다. 아래는 실험 결과입니다.



비선형함수를 사용하지 않을 때가 성능이 더 좋았습니다.

기존의 inception은 1x1 conv이후와 sptial conv이후 모두 non-linearity로 ReLU를 수행하는데, Xception의 경우 activiation을 사용하지 않는 것이 더 성능이 좋았음. 저자는 이에 대해서 spatial convolution이 적용되는 intermediate feature space의 depth가 non-

lineartity의 실용성에 매우 중요하게 작용된다고 말함 즉, inception module에 있는 deep feature space(여러개의 channel)의 경우에는 non-linearity가 도움이 되지만, shallow feature space(1-channel)의 경우에는 오히려 정보 손실로 인해 해로울 수가 있다고 함.

전체 Xception 구조

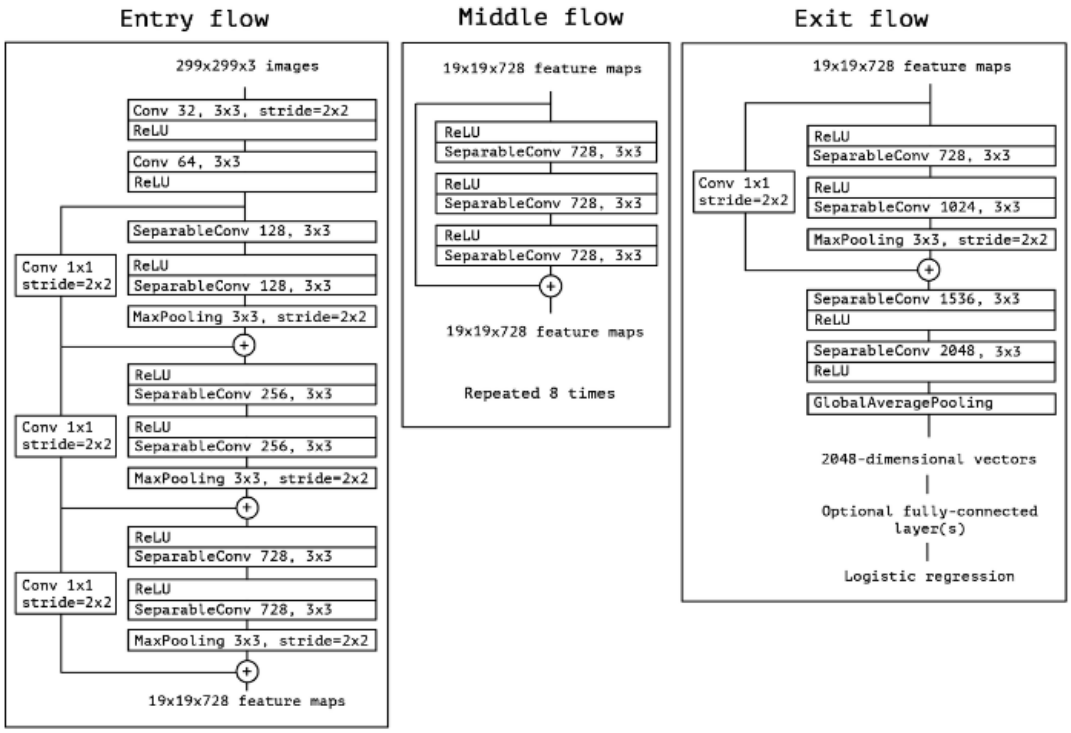


그림 4 Xception Architecture

Xception 구조는 residual connection이 있는 depthwise separble convolution의 linear stack으로 볼 수 있음. 36개의 컨볼루션 레이어는 14개의 모듈로 구성되어 있으며, 첫 번째 모듈과 마지막 모듈을 제외하고 모두 주변에 linear residual connections가 있습니다. 따라서 정의 및 수정이 용이합니다.

모든 Convolution 레이어 및 Separable Convolution 레이어 뒤에 배치 정규화가 계속된다. 모든 SeparableConvolution 레이어는 1의 깊이 승수를 사용합니다(깊이 확장 x)

Experimental evaluation

Table 3. Size and training speed comparison.

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

비교 모델 : Xception VS Inceptino V3

Xception과 Inception V3의 규모 유사성 때문에 Xception을 Inception V3 아키텍처와 비교하기로 선택했다. 즉, Xception과 Inception V3의 매개변수 수는 거의 같으며 따라서 성능 격차는 네트워크 용량의 차이에 기인할 수 없다.

데이터 셋 : ImageNet dataset, JFT dataset(+FastEval14k)

사용된 JFT dataset은 대규모 이미지 분류 google 내부 데이터셋으로, 17000개 클래스 집합의 레이블이 달린 3억 5천만개 이상의 고해상도 이미지로 구성됩니다. JFT에 대해 교육된 모델의 성능을 평가하기 위해 보조 데이터 세트인 FastEval14k를 사용합니다.

성능평가 : Mean Average Precision(MAP@100)

소셜 미디어 이미지에서 클래스가 얼마나 공통(중요한)지를 추정하는 점수로 각클래스의 기여도를 평가합니다. 이 평가 절차는 구글에서 생산모델에 소셜미디어에서 자주 발생하는 라벨의 성능을 포착하기 위한 것이다.

Optimization configuration : 동일한 optimization configuration 사용

인셉션-v3와 동일한 optimization configuration을 사용했고, Xception에 최적화된 hyperparameter를 조정하려는 시도를 하지 않고 inception-v3에 맞춰진 최적값이었는데, 비슷한 수의 파라미터를 가지면서 정확도를 향상시킴. 저자는 모델이 parameter를 효율적으로 사용했다고 말함

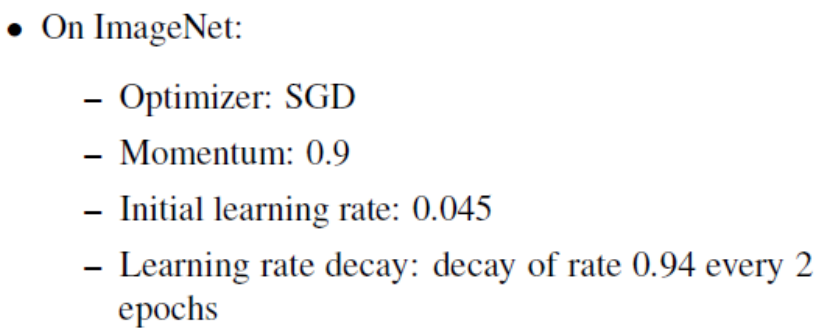
#또한 모든 모델 추론시 polyak averaging을 사용

Regularizaiton configuration : Weight decay, Dropout, Auxiliary loss tower

ImageNet dataset

	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**

Model	Size	Top-1 Accuracy	Top-5 Accuracy	Parameters	Depth
Xception	88 MB	0.790	0.945	22,910,480	126
VGG16	528 MB	0.713	0.901	138,357,544	23
VGG19	549 MB	0.713	0.900	143,667,240	26
ResNet50	98 MB	0.749	0.921	25,636,712	-
ResNet101	171 MB	0.764	0.928	44,707,176	-
ResNet152	232 MB	0.766	0.931	60,419,944	-
ResNet50V2	98 MB	0.760	0.930	25,613,800	-
ResNet101V2	171 MB	0.772	0.938	44,675,560	-
ResNet152V2	232 MB	0.780	0.942	60,380,648	-
InceptionV3	92 MB	0.779	0.937	23,851,784	159
InceptionResNetV2	215 MB	0.803	0.953	55,873,736	572
MobileNet	16 MB	0.704	0.895	4,253,864	88
MobileNetV2	14 MB	0.713	0.901	3,538,984	88
DenseNet121	33 MB	0.750	0.923	8,062,504	121
DenseNet169	57 MB	0.762	0.932	14,307,880	169
DenseNet201	80 MB	0.773	0.936	20,242,984	201
NASNetMobile	23 MB	0.744	0.919	5,326,716	-
NASNetLarge	343 MB	0.825	0.960	88,949,818	-



ImageNet에서 Xception은 Inception V3보다 약간 더 나은 결과를 볼 수 있습니다.

JFT dataset

JFT에 대해 학습된 모델의 성능 평가는 보조 dataset인 FastEval14k 사용

Table 2. Classification performance comparison on JFT (single crop, single model).

	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**

- On JFT:
  - Optimizer: RMSprop [22]
  - Momentum: 0.9
  - Initial learning rate: 0.001

JFT에서 Xception은 FastEval14k MAP@100 매트릭에서 4.3% 향상되었습니다. 또한 REsnet-50, 101,152 에 대해서도 능가하는 것을 볼 수 있습니다.

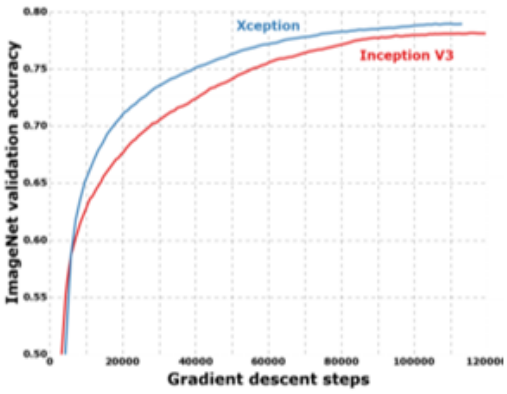


Figure 6. Training profile on ImageNet

그림 6.ImageNet 교육 프로필

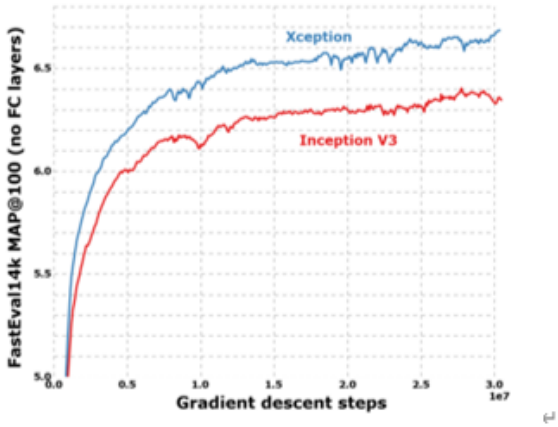


Figure 7. Training profile on JFT, without fully-connected layers

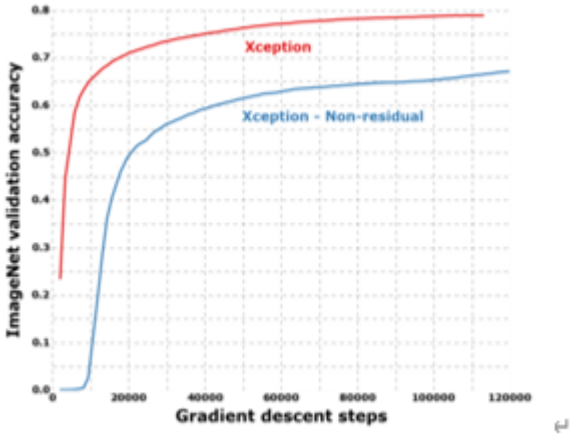
그림 7 JFT에서의 트레이닝 프로파일(완전 접속 레이어 없음)

Xception이라는 이 아키텍처는 ImageNet 데이터셋(Inception V3용으로 설계됨)에서 Inception V3를 약간 능가하며, 3억 5천만 개의 이미지와 17,000개의 클래스로 구성된 대규모 이미지 분류 데이터셋에서 Inception V3을 크게 능가합니다. ImageNet에서

Xception에 대한 더 나은 하이퍼파라미터를 검색하면 추가 개선 효과를 얻을 수 있을 것이다.

Xception 아키텍처는 Inception V3과 동일한 수의 매개변수를 가지므로 성능 향상은 용량 증가가 아니라 모델 매개변수를 보다 효율적으로 사용함으로써 나타난다는 것을 볼 수 있다.

Training profile with and without residual connections (대충 패스해도 괜찮은 부분)



residual connections는 속도와 최종 분류 성과 면에서 정합화를 돕는 데 있어 필수적이다. 이 결과는 특정 아키텍처에 대한 residual connections의 중요성을 보여줄 뿐이며 depthwise separable convolutions 스택 모델을 구착하기 위해 필요한 것은 아니다.

비선형성의 부재로 인한 효과

4.7. Effect of an intermediate activation after pointwise convolutions

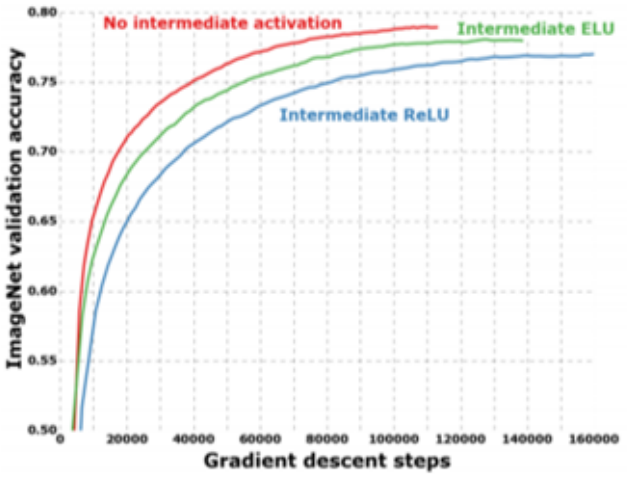


Figure 10. Training profile with different activations between the depthwise and pointwise operations of the separable convolution layers.

비선형성의 부재로 인해 수렴속도가 빨라지고 최종 성능이 향상된다는 것을 알 수 있습니다. 공간 변환이 적용되는 중간 피쳐 공간의 깊이는 비선형성의 유용성에 중요한 것을 알 수 있다. 앞서 말했듯이 채널이 1개개인 경우이기 때문에. (정보손실)

결론

인셉션 모듈을 neural computer vision 아키텍처에서 Depthwise separable convolutions로 대체할 것을 제한하게 됐습니다. 우리는 이 아이디어를 바탕으로 Inception V3와 유사한 매개변수 수를 가진 Xception이라는 새로운 아키텍처를 제시했습니다. Inception V3에 비해, Xception은 ImageNet 데이터셋에서 분류 성능이 약간 향상되고 JFT 데이터셋에서 큰 폭으로 향상됩니다. Depthwise separable convolutions는 인셉션 모듈과 유사한 특성을 제공하면서도 일반 컨버전스 레이어처럼 사용하기 쉽기 때문에 향후 컨버전스 네트워크 아키텍처 설계의 초석이 될 것으로 예상합니다.