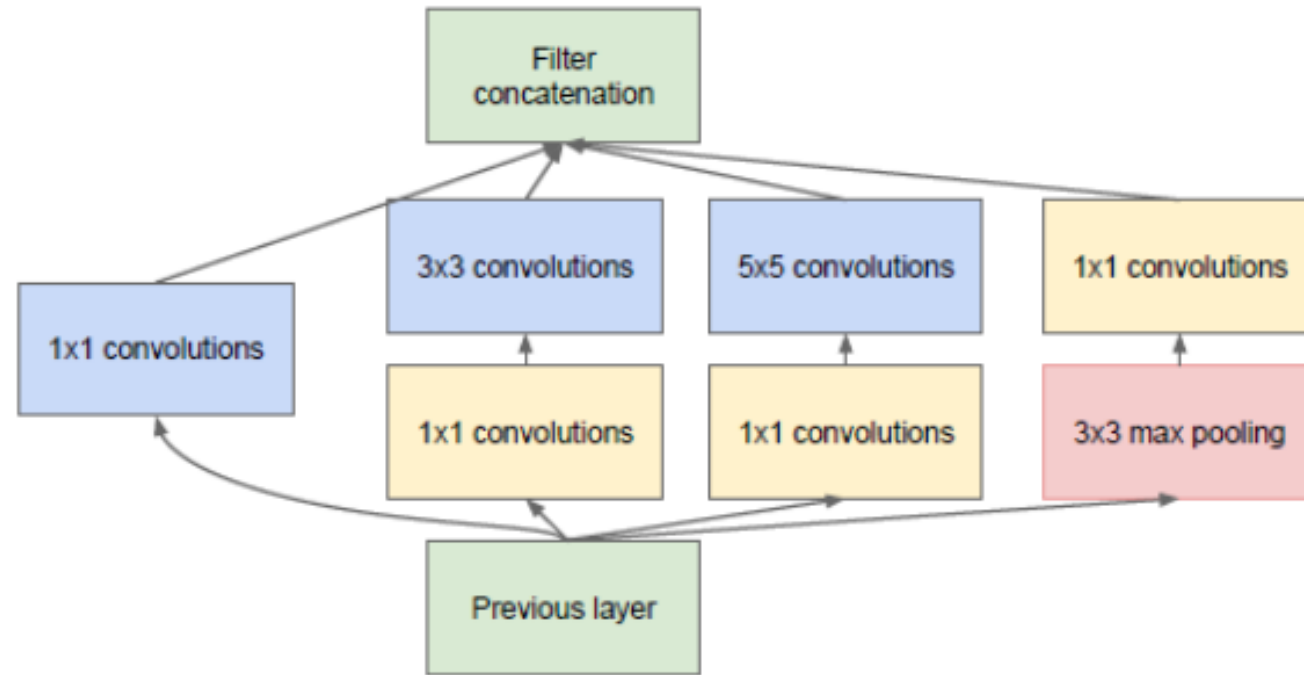


Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning

Background

- Inception은 피쳐맵 **1x1 convolution**을 추가하여 연산량을 줄였다을 여러 개로 나누어 다른 연산을 적용시키면 **더 다양한 정보**를 추출하는 방법이다. 그리고, 그리고 이 구조를 기반으로 발전 시킨 것이 Inception v2,v3,v4이다.



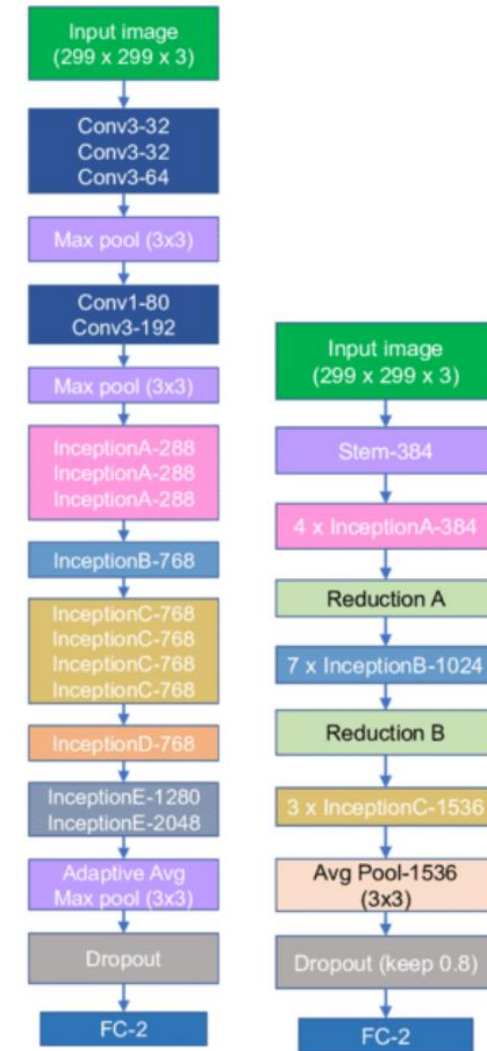
<Inception module>

Background

- Inception network에 **ResNet**구조를 적용시키지 않더라도 깊은 신경망에서 학습이 잘 되도록 연구하였다.
- Inception-v4는 Inception 신경망을 좀 더 **효과적으로 넓고 깊게** 만들기 위해 고안되었다. Inception-v3보다 **단순하고 획일화된 구조**와 더 많은 Inception module을 사용한다.
- ResNet은 깊은 신경망에서 학습이 잘 되도록 설계되었다. Inception network는 깊은 신경망 구조를 가지기 때문에 Inception network에 적용하였다. 그리고 이를 적용하지 않은 모델과 비교한다.

Model

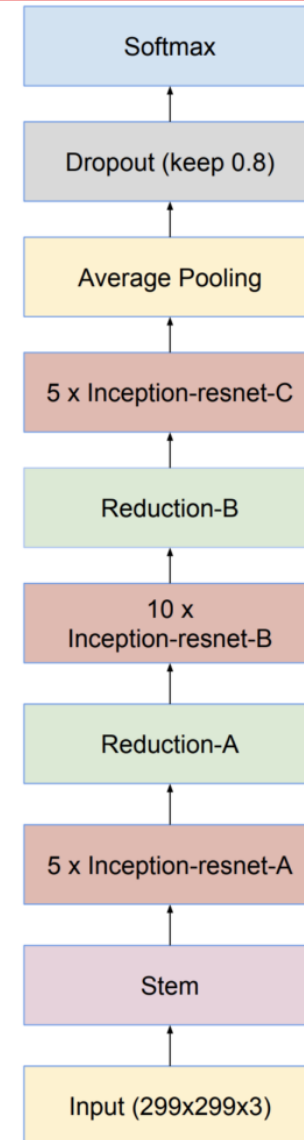
- Inception v4는 Inception v3보다 단순하게 만들기 위해 반복 사용되는 블록이 Inception v3보다 많다. 그리고 블록을 수정하여 더 넓고, 깊게 만들었다.



<왼쪽은 Inception v3,
오른쪽은 Inception v4>

Model

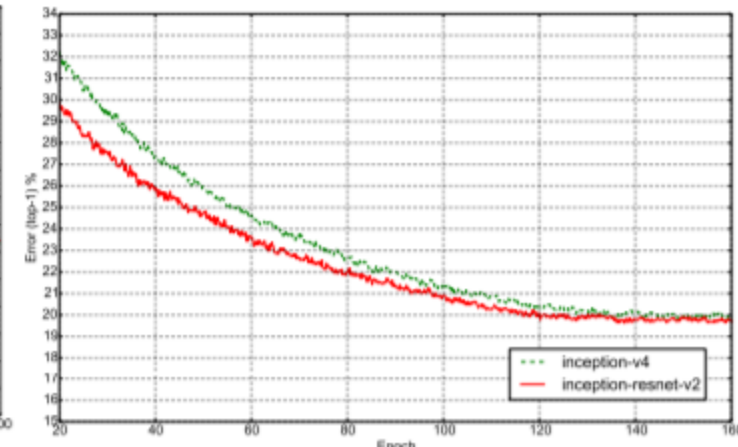
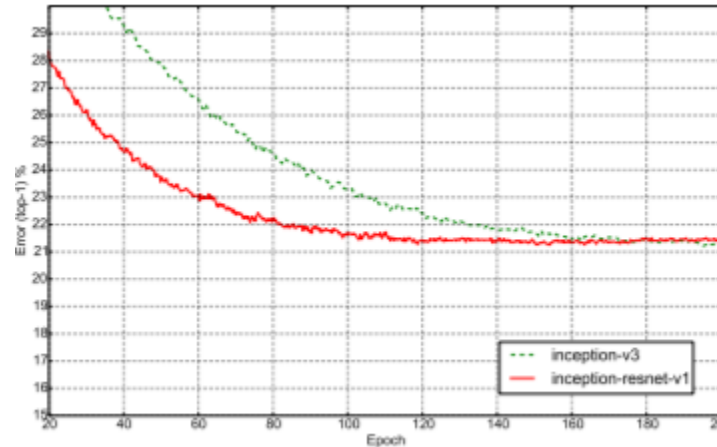
- Inception-resnet-v1, v2는 같은 구조를 가지지만 **필터의 개수가 다르다**. 그리고 Stem 블록이 Inception-resnet-v1, v2는 각각 Inception v3, v4의 Stem 블록과 유사하다.
- Inception-resnet 블록은 residual function 구조의 **bottleneck**을 **Inception**으로 변경한 것이다.



<Inception-resnet-v1,v2의 구조>

Result

- Inception v3와 Inception-resnet-v1의 연산량이 비슷하고, Inception v4와 Inception-resnet-v2의 연산량이 비슷하다.
- Resnet 적용 여부에 따른 학습 속도 차이를 비교하였을 때, **Inception v3**에서는 확실한 차이가 있지만 **Inception v4**에서는 비슷한 학습 속도를 보였다.
- 또한 검증 성능을 비교하였을 때 각각 비슷한 성능을 보였다.



<Inception과 ResNet+Inception 학습 속도 비교>

Network	Top-1 Error	Top-5 Error
BN-Inception [6]	25.2%	7.8%
Inception-v3 [15]	21.2%	5.6%
Inception-ResNet-v1	21.3%	5.5%
Inception-v4	20.0%	5.0%
Inception-ResNet-v2	19.9%	4.9%

<Inception을 개선한 방법과 ResNet을 적용한 Inception들을 비교>