Senet : Squeeze-and-Excitation Networks

Category	Computer Vision

요약

- spatial-wise, channel-wise 정보를 지역적인 receptive field 안에서 정보를 잘 추출하는게 Convolutional Layer
- Spatial encoding를 향상하는 것이 필요
- 채널 간 상호의존성을 명시적(explicit)하게 모델링하여 채널별 특징 반응을 적응적으로 재적용함.
- SE block들을 쌓아 올라감으로써 다양한 난이도의 데이터셋에서도 일반화 성능을 보임.
- SE block이 최소의 추가 비용으로 최고 성능을 기록하여, 2017년 ILsVRC 분류 대회에서 1등을 하였음(2016년 우승 기록보다 상대적으로 25%적으로 향상시키고, top-5 error를 2.251%로 줄임)

소개

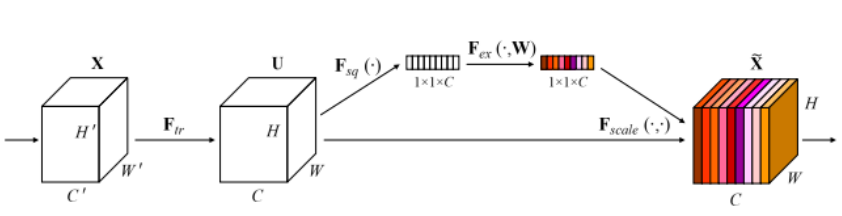

Fig. 1. A Squeeze-and-Excitation block.

- CNN은 각 합성곱 층에서 여러 개의 필터가 입력 채널을 따라 지역적인 공간 연결 패턴을 표현하며, 이를 통해 공간 및 채널 차원의 정보를 국소 수용 영역 내에서 결합함.

- Convolution 층을 비선형 활성화 함수 및 다운샘플링 연산자와 교차 배치함으로써, CNN은 계층적인 패턴을 포착하고 전체적인 이론적 영역을 수용 영역을 확보하는 이미지 표현을 생성할 수 있음
- 컴퓨터 비전에서는 주어진 task에 맞는 중요한 이미지의 속성만을 효과적으로 잘 포착하는 강력한 표현을 탐색해서 성능을 향상시키는 것.
- 최근 연구에 따르면 CNN이 생성하는 표현은 네트워크에 학습 매커니즘을 통합해 특징 간 공간적 상관관계를 포착함으로써 더욱 더 강해짐.
- Inception 계열 모델에서는 네트워크 모듈에 다중 스케일 처리를 통합해 성능을 향상시키는 방법이 있었음
- 이후에는 공간적 종속성을 보다 잘 모델링하고, 네트워크 구조에 공간적 어텐션(spatial attention)을 포함하려는 시도가 있었음.
- 이 논문은 채널 간 관계에 대해 연구함.
    - Squeeze-and-Excitation block을 도입하여 합성곱 특징의 채널 간 상호 의존성을 명시적으로 모델링함으로써 네트워크가 생성하는 표현의 품질을 향상하는 것이 목표.
    - feature recalibration(네트워크 피처 재보정)을 수행할 수 있도록 하는 매커니즘 제안
        - 이를 통해서 global한 정보를 활용하여 중요한 특징을 선택적으로 강조하고 덜 유용한 특징을 억제할 수 있게 학습 가능
- 합성곱 연산을 수행 후, feature recalibration(특징 재보정)을 수행하는 SE 블록 구성 가능
    - squeeze 연산 : 공간 차원에 걸쳐 특징 맵을 집계해 채널 설명자(channel descriptor)를 생성하는 과정
        - 채널별 특징 반응의 전역 분포(global distribution)을 임베딩 해 네트워크의 전역 수용 영역(global receptive field) 정보를 모든 층에서 활용할 수 있게 함
    - excitation 연산 : self-gating 메커니즘 형태를 갖고, 앞에서 생성된 임베딩을 입력으로 받아 채널별 변조 가중치(per-channel modulation weight) 생성함. 이런 가중치는 특징맵에 적용되어 SE block의 출력을 생성하여 이후 층으로 직접 전달함.
- SE 블록을 단순이 여러개 쌓아 올릴 수 있으,ㅁ
- 네트워크 아키텍처의 다양한 깊이에서 기존의 블록을 대체하는 방식으로도 사용 가능.

    - 네트워크 깊이에 따라 수행하는 역할이 다.
- 초반 레이어 : 클래스와 무관하게 유용한 특징을 강조하여 공유되는 저수준 표현 강화
- 후반 레이어: SE block이 점점 더 특화되어, 입력에 따라 클래스별로 반응함.
    - 특징 재조정(feature recalibration) 이점이 네트워크를 통해 누적됨.
- 보통은 CNN 모델을 설계할때 새롭고 많은 하이퍼파라미터나 레이어 구성을 선택해야 하는 경우가 맣은데, 이 SE 블록은, 단순하고, 기존의 최첨단 아키텍처에서도 특정 구성 요소들을 SE 블록으로 대체하는 방식으로만으로도 성능 개선 효과가 있음.
- ImageNet을 넘어섰으며, ILSVRC 2017 Classification 대회에서 1위
    - 최적의 모델 앙상블은 테스트 세트에서 2.251%의 top-5 오류율을 달성함.
    - 이는 전년도 우승 모델(top-5 오류율 2.991%)과 비교했을 때 약 25%의 상대적 개선을 의미한다.

관련 연구

- **깊은 아키텍처**
    - 깊이 쌓는 게 성능 향상에 유리(VGGNet, Inception)
    - 배치 정규화(BN, Batch Normalization) :
        - 레이어 입력을 조절하는 유닛을 삽입하여, gradient 전파를 개선함
    - ResNet은 identity 기반 skip connection을 통하여 더 깊은 네트워크를 효과적으로 학습
    - Highway Network는 gating 메커니즘을 사용해 단축 경로 (shortcut connection) 조정
    - 그룹화된 컨볼루션(grouped convolution)은 변환 집합의 크기(카디널리티, cardinality) 증가시키는 데 사용
    - 다중 분기 컨볼루션(multi-branch convolution)은 이러한 그룹화된 컨볼루션을 일반화해 연산자의 보다 유연한 조합을 가능하게 함
    - 최근엔, 자동화된 방법을 통해 학습된 조합(composition)이 경쟁력 있는 성능을 보이고 있음
    - 채널 간 상관관계(cross-channel correlation)은 일반적으로 새로운 특징 조합으로 매핑됨.

    - 공간적 구조(spttial structure)와 함께 독립적으로 또는 표준 컨볼루션 필터를 활용하여 1x1 convoltuion과 함께 공동으로 매핑됨.
        - 위 두개의 연구는 모델 및 연산 복잡도를 줄이는 데 초점을 맞춤
        - 채널 간 관계를 개별 인스턴스에 무관한(instance-agnostic) 함수의 조합과 로컬 수용 영역(local receptive field)으로 공식화할 수 있다는 가정을 반영
    - 이 논문 저자들은 전역 정보를 활용하여 채널 간의 동적이고 비선형적인 관계를 명시적으로 모델링할 수 있는 메커니즘을 제공하면 학습이 용이해지고 네트워크의 표현력을 크게 향상시킬 수 있다고 주장.
- 어텐션 및 게이팅 알고리즘
    - 어텐션은 입력 신호에서 정보량이 가장 많은 요소에 가용한 연산 자원을 집중하도록 조정하는 도구
        - 이미지에서의 객체 위치 추정(localization) 및 이해(understanding)에서 사용
        - 어텐션은 softmax 또는 sigmoid와 같은 gating function 및 순차적 기법과 함께 구현됨
        - 이미지 캡셔닝(image captioning)과 립 리딩(lip reading)에서는 , 어텐션은 일반적으로 한 개 이상의 고차원적 추상화 계층 위에서 사용되어 모달리티 간의 적응을 돕는다.
    - trunk-and-mask attention algorithm(Wang et al)은 시간축(hourglass) 모듈을 활용하여 심층 잔차 네트워크(deep residual network)의 중간 단계에 삽입됨
- 반면 SE block은 lightweight(가벼운) 게이팅 메커니즘이라, 효율적 방식으로 계산하여 채널 간 관계를 모델링하도록 설계.
- SE 블록은 네트워크 전체에서 기본 모듈의 표현력을 강화하는 역할을 수행한다.

모델의 핵심 연산

Squeeze

- 채널 간 의존성을 효과적으로 활용하는 문제를 해결하기 위해, 우리는 먼저 출력 특징 맵의 각 채널에 전달되는 신호를 고려함.
- 학습된 필터들은 로컬 수용 영역(local receptive field) 내에서 작동하기 때문에 transform된 output의 각각의 unit들은 영역 밖 context 정보를 활용할 수 없다는 단점이 발생함

    - 이는 해당 계층의 recpeptive field(수용 영역) 크기가 작아서임.
- 따라서 전역 공간 정보를 채널 기술자(channel descriptor)로 압축(squeeze)하는 방식을 제안
- **전역 평균 풀링(global average pooling)** 을 사용하여 채널별 통계를 생성한다.
- 결과를 공간 차원(spatial dimension) H x W로 축소하여 통계 벡터를 얻을 수 있음.
- 아래의 식은 z의 c번째 요소를 계산하는 수식임.

$$z_c = F_{sq}(u_c) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} u_c(i, j)$$

- transformation된 결과값은 local descriptor의 모음으로 설명할 수 있고, local descriptor의 통계는 전체 이미지를 표현함.
- 이러한 정보를 활용하는 것은 특징 공학(feature engineering)에서 널리 사용되는 기법이며, 이 논문의 저자는 장 간단한 방법인 **전역 평균 풀링(global average pooling)** 을 선택했지만, 이 과정에서 보다 정교한 통합(aggregation) 전략을 사용할 수도 있음.

Excitation : 적응형 재조정(Adaptive Recalibration)

- Squeeze 연산에서 집계된 정보를 활용하기 위해 채널 간 의존성을 완전히 포착하기.
    1. 유연성(Flexiability) : 채널 간 비선형적인 상호작용을 배울 수 있는 능력이 있어야 함.
    2. 비배타적(non-exclusive) 관계 학습 : 단일 채널만 활성화(one-hot activation)이 아니라, 여러 채널을 강조할 수 있어야 함.
- 이 모델에서는 게이팅 메커니즘으로 Sigmoid 활성화 함수를 사용하엿음.
- 모델 복잡성을 제한하고 일반화를 위하여, non-linear(비선형) 함수 주변에 병목 구조를 형성하는 두 개의 FC Layer를 사용하여 게이팅 매커니즘을 매개변수화 함.
    - 차원 축소하는 Layer가 있으며, 감소율(reduction ratio)로 차원을 줄임.
    - RELU 함수를 적용하고 차원 증가(dimensionality-increasing) layer를 적용함.
    - 최종 결과값은 transformation의 결과를 activation으로 재조정(rescaling)하는 것.
- activation은 descriptor에 적응된 채널 가중치로 작용힘
    - 이러한 면에서 SE 블록은 입력에 따라 동적으로 반응하는 매커니즘을 내재적으로 포함.

- 특징(feature)의 판별력(discriminability)을 강화하는데 도움을 줌

응용 사례

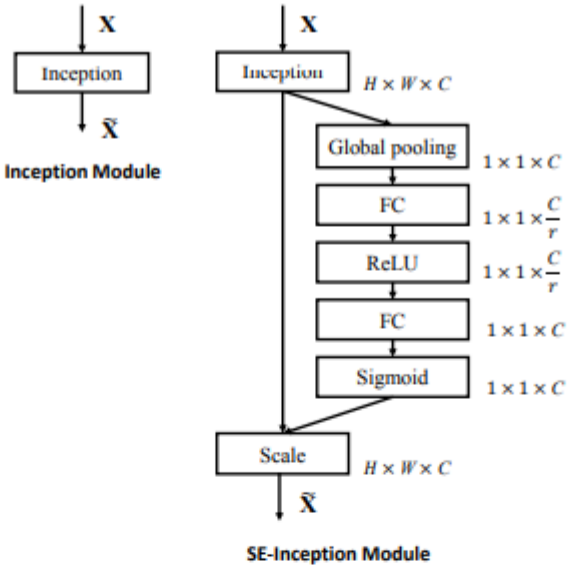

Figure 2: The schema of the original Inception module (left) and the SE-Inception module (right).

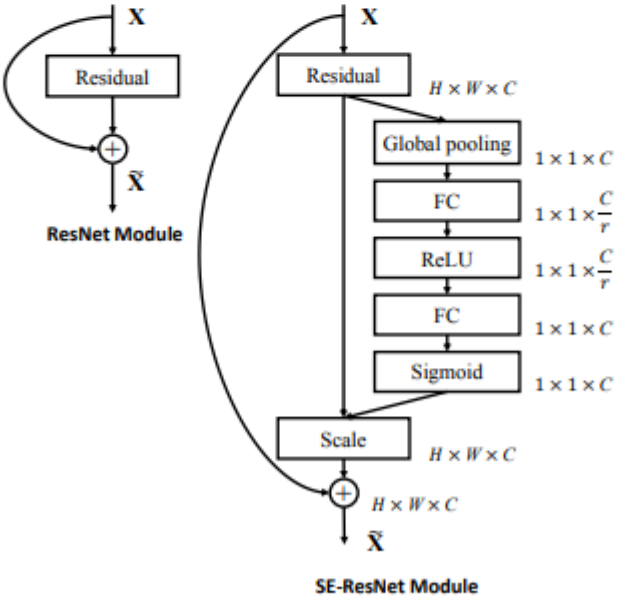

Figure 3: The schema of the original Residual module (left) and the SE-ResNet module (right).

- AlexNet 및 VGGNet에 간단하게 적용 가능
- 표준 합성곱 뿐만 아니라 다양한 transformation에 적용함.
- 이를 위해 SENet 개발
- 비잔차 네트워크인 경우
  - Inception인 경우,변환 함수를 Inception 모듈 전체로 설정
    - 각 Inception 모듈에 SE block 추가해서 SE-Inception Network 구성
- 잔차(Residual Network)인 경우
  - SE 블록 변환 함수는 잔차 모듈의 비-항등(non-identity) 분기로 설정
  - Squeeze와 Excitation 연산이 항등(identity) 분기 합산되기 전에 적용
- **ResNeXt** , **Inception-ResNet**, **MobileNet** , **ShuffleNet** 등 다양한 최신 네트워크 구조에도 SE 블록을 통합 가능

모델 및 시간 복잡도

Output size	ResNet-50	SE-ResNet-50	SE-ResNeXt-50 (32 × 4d)
112 × 112	conv, 7 × 7, 64, stride 2		
56 × 56	max pool, 3 × 3, stride 2		
	$\begin{bmatrix} \text{conv}, 1\times1, 64 \\ \text{conv}, 3\times3, 64 \\ \text{conv}, 1\times1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} \text{conv}, 1\times1, 64 \\ \text{conv}, 3\times3, 64 \\ \text{conv}, 1\times1, 256 \\ fc, [16, 256] \end{bmatrix} \times 3$	$\begin{bmatrix} \text{conv}, 1\times1, 128 \\ \text{conv}, 3\times3, 128 \quad C=32 \\ \text{conv}, 1\times1, 256 \\ fc, [16, 256] \end{bmatrix} \times 3$
28 × 28	$\begin{bmatrix} \text{conv}, 1\times1, 128 \\ \text{conv}, 3\times3, 128 \\ \text{conv}, 1\times1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} \text{conv}, 1\times1, 128 \\ \text{conv}, 3\times3, 128 \\ \text{conv}, 1\times1, 512 \\ fc, [32, 512] \end{bmatrix} \times 4$	$\begin{bmatrix} \text{conv}, 1\times1, 256 \\ \text{conv}, 3\times3, 256 \quad C=32 \\ \text{conv}, 1\times1, 512 \\ fc, [32, 512] \end{bmatrix} \times 4$
14 × 14	$\begin{bmatrix} \text{conv}, 1\times1, 256 \\ \text{conv}, 3\times3, 256 \\ \text{conv}, 1\times1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} \text{conv}, 1\times1, 256 \\ \text{conv}, 3\times3, 256 \\ \text{conv}, 1\times1, 1024 \\ fc, [64, 1024] \end{bmatrix} \times 6$	$\begin{bmatrix} \text{conv}, 1\times1, 512 \\ \text{conv}, 3\times3, 512 \quad C=32 \\ \text{conv}, 1\times1, 1024 \\ fc, [64, 1024] \end{bmatrix} \times 6$
7×7	$\begin{bmatrix} \text{conv}, 1\times1, 512 \\ \text{conv}, 3\times3, 512 \\ \text{conv}, 1\times1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} \text{conv}, 1\times1, 512 \\ \text{conv}, 3\times3, 512 \\ \text{conv}, 1\times1, 2048 \\ fc, [128, 2048] \end{bmatrix} \times 3$	$\begin{bmatrix} \text{conv}, 1\times1, 1024 \\ \text{conv}, 3\times3, 1024 \quad C=32 \\ \text{conv}, 1\times1, 2048 \\ fc, [128, 2048] \end{bmatrix} \times 3$
1 × 1	global average pool, 1000-d fc, softmax		

Table 1: (**Left**) ResNet-50. (**Middle**) SE-ResNet-50. (**Right**) SE-ResNeXt-50 with a 32×4d template. The shapes and operations with specific parameter settings of a residual building block are listed inside the brackets and the number of stacked blocks in a stage is presented outside. The inner brackets following by *fc* indicates the output dimension of the two fully connected layers in an SE module.

	original		re-implementation			SENet		
	top-1 err.	top-5 err.	top-1err.	top-5 err.	GFLOPs	top-1 err.	top-5 err.	GFLOPs
ResNet-50 [10]	24.7	7.8	24.80	7.48	3.86	$23.29_{(1.51)}$	$6.62_{(0.86)}$	3.87
ResNet-101 [10]	23.6	7.1	23.17	6.52	7.58	$22.38_{(0.79)}$	$6.07_{(0.45)}$	7.60
ResNet-152 [10]	23.0	6.7	22.42	6.34	11.30	$21.57_{(0.85)}$	$5.73_{(0.61)}$	11.32
ResNeXt-50 [47]	22.2	-	22.11	5.90	4.24	$21.10_{(1.01)}$	$5.49_{(0.41)}$	4.25
ResNeXt-101 [47]	21.2	5.6	21.18	5.57	7.99	$20.70_{(0.48)}$	$5.01_{(0.56)}$	8.00
VGG-16 [39]	-	-	27.02	8.81	15.47	$25.22_{(1.80)}$	$7.70_{(1.11)}$	15.48
BN-Inception [16]	25.2	7.82	25.38	7.89	2.03	$24.23_{(1.15)}$	$7.14_{(0.75)}$	2.04
Inception-ResNet-v2 [42]	$19.9^{\dagger}$	$4.9^{\dagger}$	20.37	5.21	11.75	$19.80_{(0.57)}$	$4.79_{(0.42)}$	11.76

Table 2: Single-crop error rates (%) on the ImageNet validation set and complexity comparisons. The *original* column refers to the results reported in the original papers. To enable a fair comparison, we re-train the baseline models and report the scores in the *re-implementation* column. The *SENet* column refers to the corresponding architectures in which SE blocks have been added. The numbers in brackets denote the performance improvement over the re-implemented baselines. † indicates that the model has been evaluated on the non-blacklisted subset of the validation set (this is discussed in more detail in [42]), which may slightly improve results. VGG-16 and SE-VGG-16 are trained with batch normalization.

- Resnet-50과 SE-block을 적용한 Resnet-50과 비교
  - 정확도 : Se-ResNEt50이 더 정확함.
  - 감소비율(reduction ratio)을 16으로 설정.(정확도와 복잡도 간의 균형이 잘 이루어짐)
  - 초당 부동소숫점 연산량(GFLops)이 se-block을 적용한게 약간 증가한 건 있음.
    - 허나 기존 GPU 라이브러리에서 작은 inner-product(내적) 연산과 전역 풀링에서 최적화가 덜 된 문제고, 작은 차이만 있어 수용할 만한 수준임.
  - 추가한 파라미터는 가장 많은 채널 자원을 처리하는 네트워크의 마지막 스테이지에서 발생하지만, 비용이 높은 마지막 스테이지를 제거하면 성능 저하를 최소화(Top-1 정확도에서 0.1% 감소)하면서, 추가 파라미터 증가를 4% 수준으로 낮출수 있음
    - 이는 모델 크기가 중요한 경우엔 유용한 옵션임.

실험

	original		re-implementation				SENet			
	top-1 err.	top-5 err.	top-1 err.	top-5 err.	MFLOPs	Million Parameters	top-1 err.	top-5 err.	MFLOPs	Million Parameters
MobileNet [13]	29.4	-	29.1	10.1	569	4.2	$25.3_{(3.8)}$	$7.9_{(2.2)}$	572	4.7
ShuffleNet [52]	34.1	-	33.9	13.6	140	1.8	$31.7_{(2.2)}$	$11.7_{(1.9)}$	142	2.4

Table 3: Single-crop error rates (%) on the ImageNet validation set and complexity comparisons. Here, MobileNet refers to "1.0 MobileNet-224" in [13] and ShuffleNet refers to "ShuffleNet $1 \times (g = 3)$" in [52].

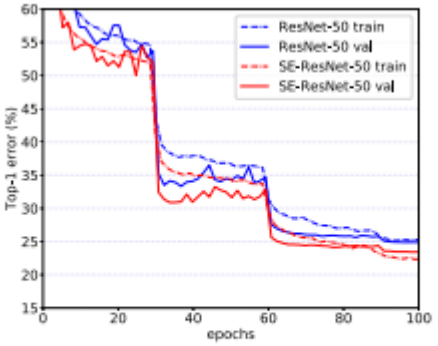

Figure 4: Training curves of ResNet-50 and SE-ResNet-50 on ImageNet.

	224×224		320×320 / 299×299	
	top-1 err.	top-5 err.	top-1 err.	top-5 err.
ResNet-152 [10]	23.0	6.7	21.3	5.5
ResNet-200 [11]	21.7	5.8	20.1	4.8
Inception-v3 [44]	-	-	21.2	5.6
Inception-v4 [42]	-	-	20.0	5.0
Inception-ResNet-v2 [42]	-	-	19.9	4.9
ResNeXt-101 (64×4d) [47]	20.4	5.3	19.1	4.4
DenseNet-264 [14]	22.15	6.12	-	-
Attention-92 [46]	-	-	19.5	4.8
Very Deep PolyNet [51] †	-	-	18.71	4.25
PyramidNet-200 [8]	20.1	5.4	19.2	4.7
DPN-131 [5]	19.93	5.12	18.55	4.16
SENet-154	**18.68**	**4.47**	**17.28**	**3.79**
NASNet-A (6@4032) [55] †	-	-	17.3‡	3.8‡
SENet-154 (post-challenge)	-	-	**16.88‡**	**3.58‡**

Table 4: Single-crop error rates of state-of-the-art CNNs on ImageNet validation set. The size of test crop is 224×224 and 320×320 / 299×299 as in [11]. † denotes the model with a larger crop 331×331. ‡ denotes the post-challenge result. SENet-154 (post-challenge) is trained with a larger input size 320×320 compared to the original one with the input size 224×224.

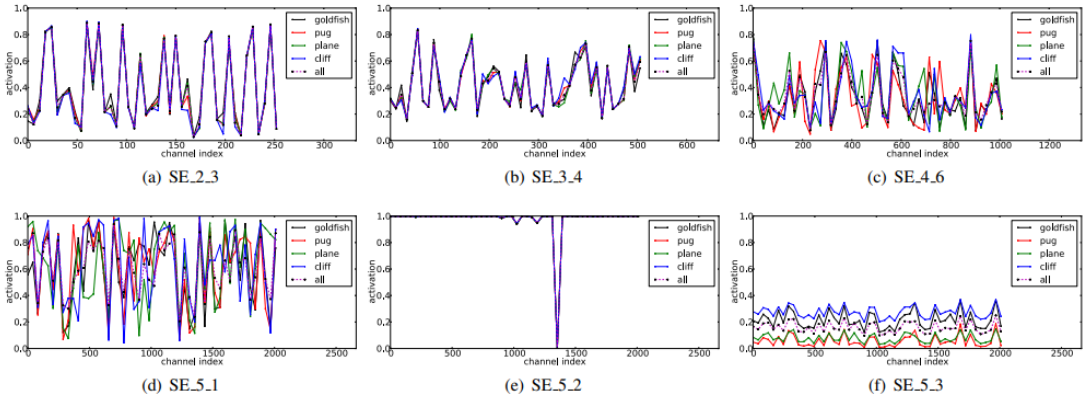

Figure 5: Activations induced by *Excitation* in the different modules of SE-ResNet-50 on ImageNet. The module is named as "*SE_stageID_blockID*".

Excitation의 역할에 대한 고찰

- ImageNet 데이터셋에서 의미론적이고 외형적으로도 차이가 있는 네 개의 클래스(금붕어, 퍼그(개),비행기, 절벽) 샘플링
- 검증 세트에서 각각 클래스별로 50개 샘플을 검증 세트에서 추출 후, 네트워크의 각 스테이지의 마지막 SE block(다운샘플링 직전) 내 50개의 균등 샘플링된 채널에 대한 평균 activation들을 계산하고, 분포를 그렸음.
- 1000개 모든 클래스에 대한 평균 활성값의 분포를 보고 관찰 결과 다음과 같은 결론을 얻음.
    - 낮은 계층에서는 서로 다른 클래스 간 활성화 분포가 동일
        - 네트워크 초기 단계에서는 다양한 클래스가 공통적으로 사용하는 feature channel 존재
    - 네트워크가 깊어질수록 각각의 클래스가 특정한 feature를 더 선호하는 경향성을 지님
        - 계층이 깊어지면, 깊은 계층에서는 클래스별로 구별된(discriminative) 특징의 중요도가 달라짐.
        - 이전 연구와도 일치하는데, 낮은 계층의 feature는 분류 관점에서 클래스에 무관하게 범용적인 면을 지니고, 높은 계층의 특징은 더 구체적인 면이 있음.
        - 따라서 se 블록에서의 재조정(recalibration)은 특징 추출과 특화(specialization)을 효과적으로 도움
    - 네트워크 마지막 단계에서는 대부분의 활성값이 1에 가깝고, 일부는 0에 가까움

    - 모든 활성값이 1이 되면 이 블록은 표준 잔차(Residual) block과 동일한 동작을 함
    - 클래스별로 약간의 스케일 조정만이 이루어질 수 있고, 따라서 이전 블록들보다 네트워크를 재조정(calibration)하는 데 있어 덜 중요함.
    - 따라서, 마지막 스테이지의 SE-block을 제거하여도 성능 저하가 미미한 대신, 전체 파라미터 수는 상당히 줄일 수 있음.

결론

- 네트워크가 동적 채널별로 feature recalibration을 수행할 수 있도록 하여 표현력을 향상시킴
- SE block이 유도하는 특징 중요도는 네트워크 압축(compression)을 위한 pruning(가지치기)작업에도 활용 가능.

참고

https://github.com/hujie-frank/SENet (깃허브 레포)

https://openaccess.thecvf.com/content_cvpr_2018/papers/Hu_Squeeze-and-Excitation_Networks_CVPR_2018_paper.pdf (논문 링크)