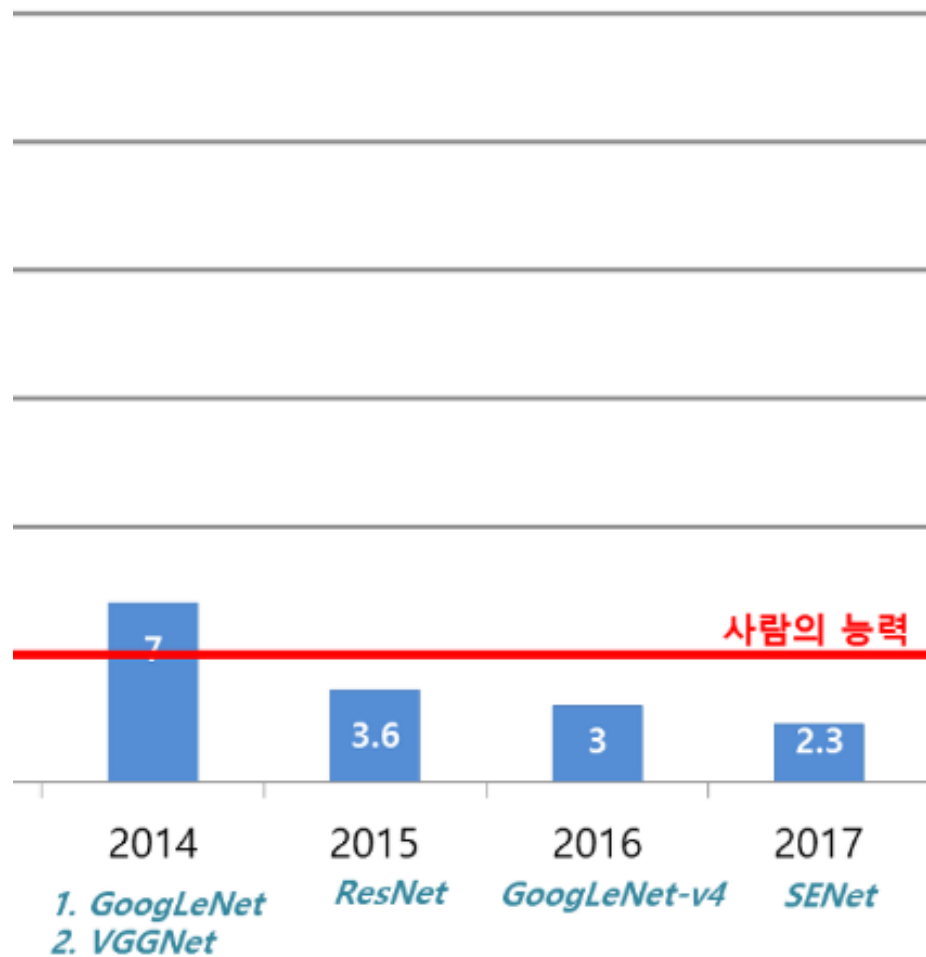


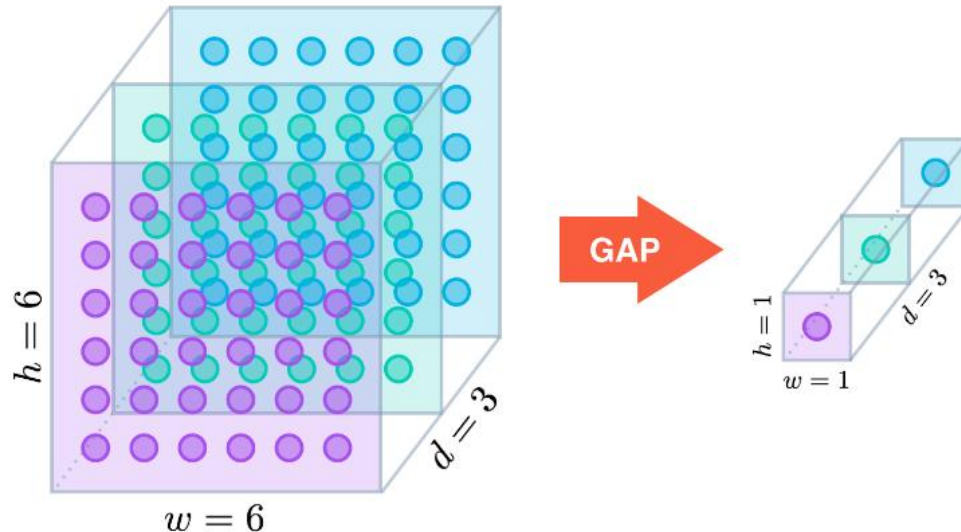
SENet

# 2017 ILSVRC 우승

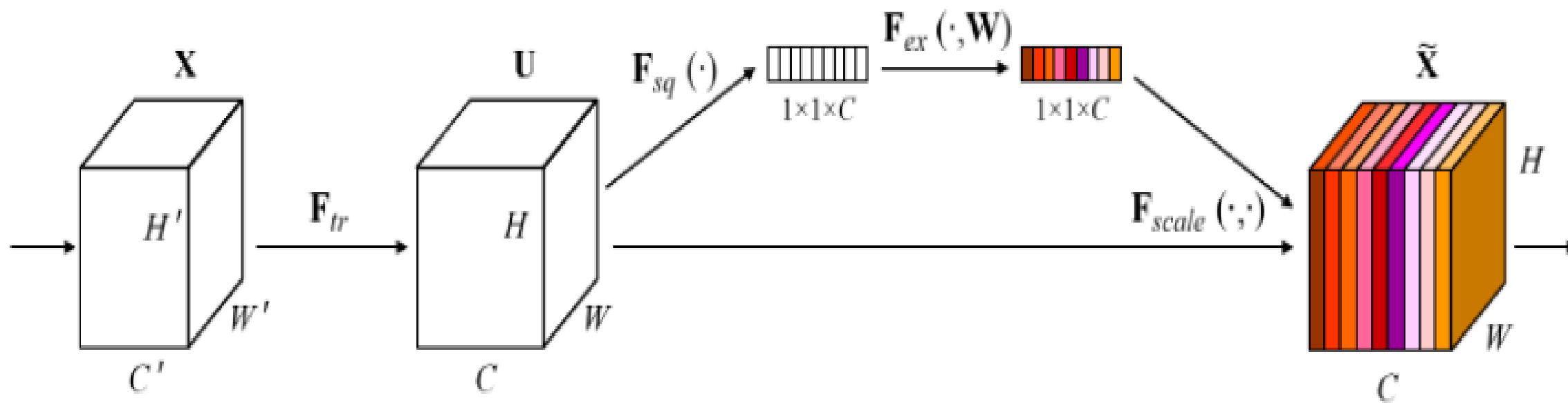


# SE block

- 기존 어떤 모델에도 적용 가능.
  - VGGNet, GoogLeNet, ResNet 등에 SE block을 첨가하여 성능을 향상시킬 수 있다.
  - **GAP(Global Average Pooling)을 통해  $h \times w$ 의 특징맵을  $1 \times 1$  으로 변환**
- > 성능은 꽤 많이 향상되는 반면에 hyperparameter는 많이 늘지 않기 때문에 연산량 증가가 크지 않다.



# SE block



# Squeeze -> Excitation

- Squeeze: GAP를 통해 각 2차원의 특성맵을 평균하여 하나의 값을 얻는다.

$$z_c = \mathbf{F}_{sq}(\mathbf{u}_c) = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W u_c(i, j).$$

- Excitation(활성화): 두 개의 FC 층을 더해줘서 각 채널의 상대적 중요도를 알아낸다.

$$s = \mathbf{F}_{ex}(\mathbf{z}, \mathbf{W}) = \sigma(g(\mathbf{z}, \mathbf{W})) = \sigma(\mathbf{W}_2 \delta(\mathbf{W}_1 \mathbf{z})),$$

$\sigma$ : sigmoid  
 $\delta$ : ReLU

- FC 층을 bottle-neck 구조가 되도록 만든다. (hidden layer의 뉴런 개수를 input layer나 output layer보다 적게 만든다.)

1. Hyperparameter의 개수를 많이 늘리지 않고,

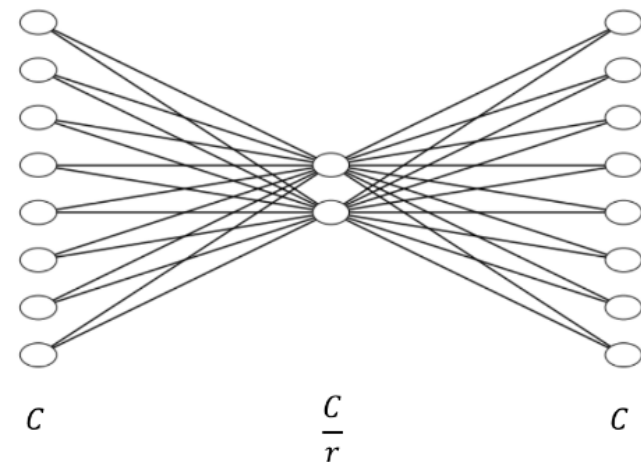
2. 일반화 성능

을 위해서.

- Hidden layer의 뉴런 개수는 Reduction ratio에 따라 결정.

Input:  $C$  -> Hidden:  $C/r$  -> Output:  $C$

$$\tilde{\mathbf{x}}_c = \mathbf{F}_{scale}(\mathbf{u}_c, s_c) = s_c \mathbf{u}_c$$



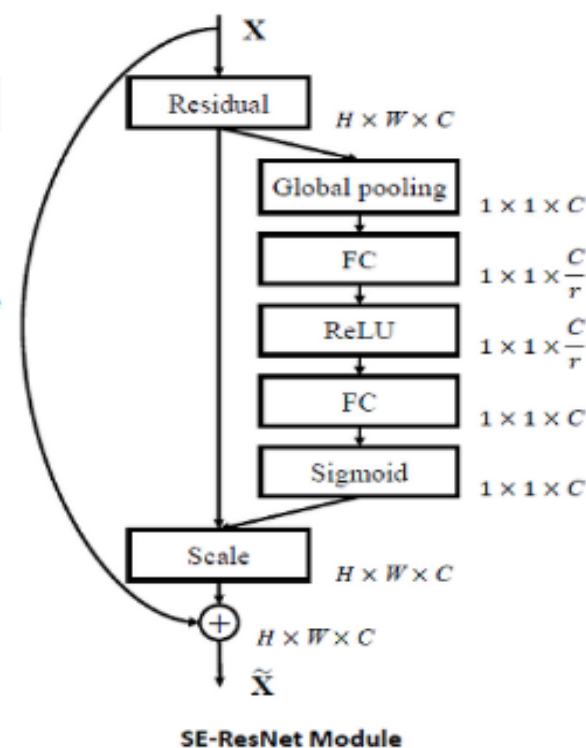
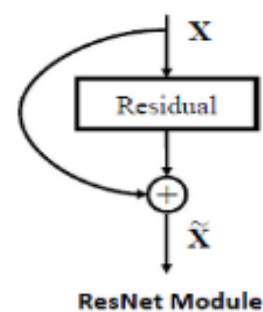
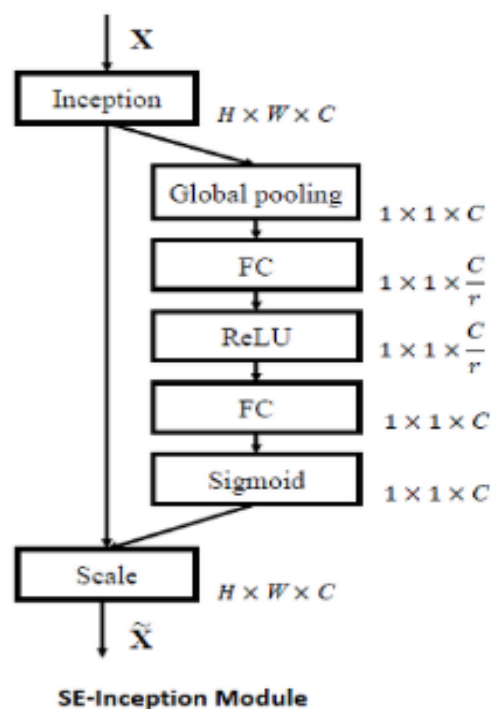
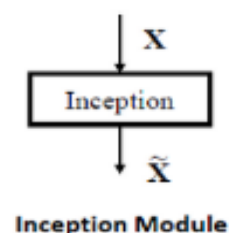
특성맵은  $X \rightarrow U(\text{conv}) \rightarrow X'$ (SE block)의 과정을 거쳐 변환된다.

# 정리

- SE Block의 목적:  
: Convolution을 통해 생성된 특성 -> 채널당 중요도를 고려해서 **재보정** (Recalibration)하는 것.
- 이런 SE block을 convolution 연산 뒤에 붙여줌으로써 성능 향상을 도모

## 기존 모델과의 결합

- VGGNet은 Convolution 및 Activation 뒤에,
- GoogLeNet은 Inception 모듈 뒤에
- ResNet은 Residual 모듈 뒤에 붙여 사용할 수 있다.



	original		re-implementation			SENet		
	top-1 err.	top-5 err.	top-1 err.	top-5 err.	GFLOPs	top-1 err.	top-5 err.	GFLOPs
ResNet-50 [13]	24.7	7.8	24.80	7.48	3.86	23.29 <sub>(1.51)</sub>	6.62 <sub>(0.86)</sub>	3.87
ResNet-101 [13]	23.6	7.1	23.17	6.52	7.58	22.38 <sub>(0.79)</sub>	6.07 <sub>(0.45)</sub>	7.60
ResNet-152 [13]	23.0	6.7	22.42	6.34	11.30	21.57 <sub>(0.85)</sub>	5.73 <sub>(0.61)</sub>	11.32
ResNeXt-50 [19]	22.2	-	22.11	5.90	4.24	21.10 <sub>(1.01)</sub>	5.49 <sub>(0.41)</sub>	4.25
ResNeXt-101 [19]	21.2	5.6	21.18	5.57	7.99	20.70 <sub>(0.48)</sub>	5.01 <sub>(0.56)</sub>	8.00
VGG-16 [11]	-	-	27.02	8.81	15.47	25.22 <sub>(1.80)</sub>	7.70 <sub>(1.11)</sub>	15.48
BN-Inception [6]	25.2	7.82	25.38	7.89	2.03	24.23 <sub>(1.15)</sub>	7.14 <sub>(0.75)</sub>	2.04
Inception-ResNet-v2 [21]	19.9 <sup>†</sup>	4.9 <sup>†</sup>	20.37	5.21	11.75	19.80 <sub>(0.57)</sub>	4.79 <sub>(0.42)</sub>	11.76



# 참고자료

- [1] <https://bskyvision.com/640>
- [2] <https://jayhey.github.io/deep%20learning/2018/07/18/SENet/>
- [3] <https://wwiiii.tistory.com/entry/SqueezeandExcitation-Networks>