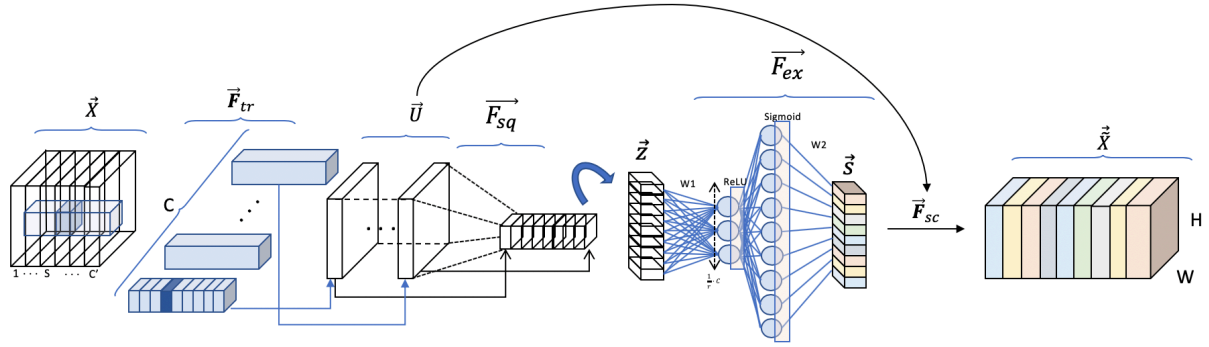
SE-Net – 2017

Pablo Ruiz – Harvard University – October 2018



What are the innovations? ***2***

New block rather than new network **2**

Motivation – What problems these blocks solve? **2**

What are the SE blocks then? **2**

Structure ***3***

1 – Transformation **3**

2 – Squeezing **5**

3 – Excitation **5**

3. Scale **6**

Application of SE Block ***7***

Results and Conclusions ***7***

Global Picture ***8***

What are the innovations?

New block rather than new network

Firstly, authors on this paper do not try to improve the state-of-art on the classic computer vision competitions by developing a novel CNN architecture are it usually is. However, and very interestingly, authors create a **block that can be used with the existing model to enhance their performance**.

This new block receives the name of SE block after the two main operations it does: Squeeze and Excitation

Motivation – What problems these blocks solve?

Authors claims that the output of a convolution result in entangled channel dependencies with the spatial correlation captured by the filters. Wow, this sound crazy! But don't worry, after a few visualizations this becomes very simple.

What do the authors want to do then with these blocks to solve that issue?

They goal is to increase the sensitivity of the network by explicitly modelling the channel interdependencies making used of gating networks, present in the SE blocks

---

What are the SE blocks then?

Shortly, SE blocks are lightweight gating mechanism in the channel-wise relationships.

Simplifying, networks are able to learn now how to understand the importance of each feature map in the stack of all the feature maps extracted after a convolution operation[1] and recalibrates that output to reflect that importance before passing the volume to the next layer.

All this will be detailed in the structure section, so don't worry if you don't see something right now!

But wait a minute, to learn, we need more parameters, right?

Exactly. A gating mechanism or gating network is no more than fully connected layers. This technique is heavily used in attention mechanism. I highly recommend this post to better understand attention mechanism and gating networks.

[1] It is applicable to other operation. However, we will see after a convolution operation since it is the most frequent and facilitates the visualization

Structure

Let's first take a look at the figure from the paper shown in Figure 1. We can call this figure as the simplified illustration, since we always want more details!

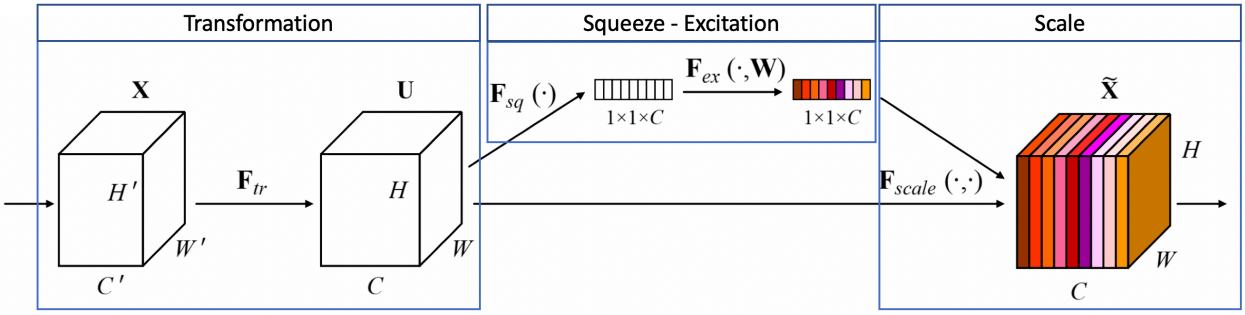


Figure 1. The SE Block from the paper.

It shows the illustration provided, where I have separated on top of it the three main parts of it. It is important to see that the Squeeze and Excitation steps only happens in the middle block, whereas the first and the last block perform different operations.

Before showing the detailed pictured all at one, let's elaborate it part by part.

1 – Transformation

The transformation simply corresponds with the operation that the network where you are going to implement the SE block would perform in its natural scheme. For instance, if you are in a block within a ResNet, the $\boldsymbol{F}_{tr}$ term will correspond with the process of the entire residual block (convolution, batch normalization, ReLU…). Therefore,

---

The SE block is applied to the output volume of a transformation operation to enrich it by calibrating the features extracted.

So, if the ResNet at some point will have output the volume X, SE blocks enrich the information it contains and get passed to the next layer becoming to $\tilde{X}$.

To simplify the visualization, let's assume that the transformation is a simple convolution operation. We have depicted in previous posts about the convolution operation, however, let's try to be slightly more detailed to better understand what is coming.

The reason why I claim that this first step is important is because the motivation of the authors to build the SE block resides here.

---

*For a regular convolution layers are a set of filters $\vec{U}$ are learned to express local spatial connectivity patterns along input channels $\vec{X}^s$. So convolutional filters are combinations of **channel-wise information within local receptive fields**.*

If we take a look at the notation corresponding to Figure 2, we have:

$\vec{X}$	Input volume
$\vec{U}$	Output volume (of the regular operation)
H', W', C'	Height, width and channel dimensions of $\vec{X}$
$\vec{V}$	4D Convolutional Kernel with C filters
s	One specific channel at the input volume ($s \epsilon [1 - c']$
$\vec{v}_c^s$	Channel s of filter v_c (that convolves channel s of input volume)

---

*The **local receptive fields** they are talking about are precisely each channel 2 space $\vec{v}_c^s$ in each filter $\vec{v}_c$.*

And why the information is channel wise?

Well, note how the marked $\vec{v}_1^s$ for filter $\vec{v}_1$ will only convolve over the channel s of the input volume $\vec{X}^s$.

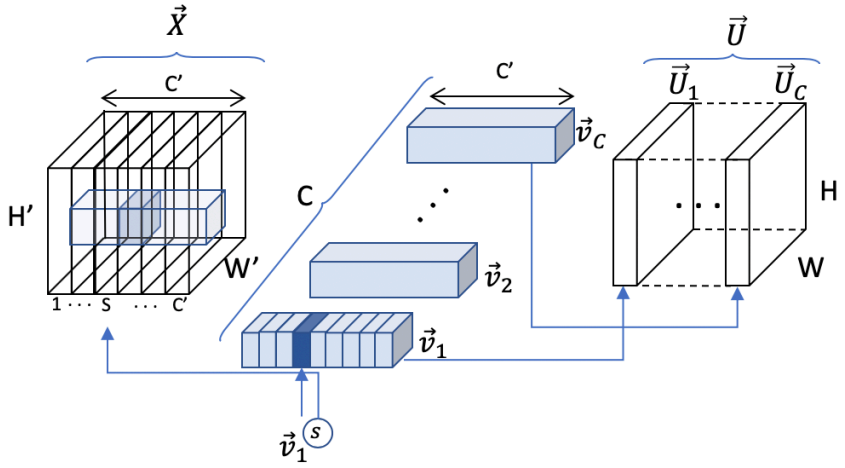


Figure 2. STEP 1: Transformation

We can represent the convolution operation as well channel-wisely to double check:

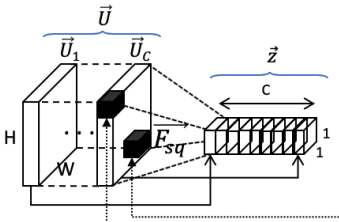
The result of convolving 1 single filter $\vec{v}_c$ in the volume $\vec{X}$ results on a single feature map, $\vec{U}_c$.

$$\vec{U}_c = \vec{v}_c \otimes \vec{X} = \sum_{s=1}^{c'} \vec{v}_c^s \otimes \vec{X}^s$$

In words, the convolution of one filter over the volume $\vec{X}$ is the sum of all the channel-wise convolutions of every channel of the input volume $\vec{X}^s$ with its correspondent channel in the filter $\vec{v}_c^s$.

2 – Squeezing

The squeezing step is probably the most simply one. It basically performs a **max pooling at each channel** to create a 1x1 *squeezed* representation of the volume $\vec{U}$.



For 1 single channel, c:

$$\vec{z}_c = \overrightarrow{F_{sq}}(U_c) = \frac{1}{HxW}\sum_{i=1}^{H}\sum_{j=1}^{W} U_c(i,j)$$

$$\underline{For\ c = C:}$$

$$\vec{z}_C = \overrightarrow{F_{sq}}(U_C) = \frac{1}{HxW}\left[\left(U_c(0,0) + \cdots + U_c(0,W)\right) + \left(U_c(1,0) + \cdots + U_c(1,W)\right) + \cdots + \left(U_c(H,0) + \cdots + U_c(H,W)\right)\right]$$

Figure 3. Squeezing

3 – Excitation

This is the key part for the entire success of the SE block, so pay attention. This was indeed a word game, we are going to use an attention mechanism using a gating network :)

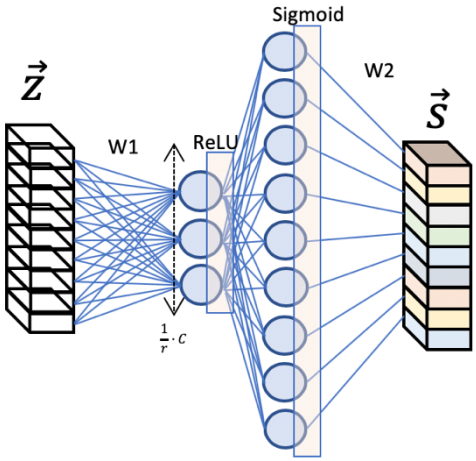


Figure 4. Excitation

The authors introduce a new parameter called the **reduction ratio** r, to introduce a first fully connected (FC) layer with a ReLU activation, before the gating network with the sigmoid activation.

The reason to do this is to introduce a bottleneck that allows us to reduce the dimensionality at the same time that introduce new non-linearities.

Furthermore, we can have better control on the model complexity and aid the generalization property of the network.

Having two FC layers will result on having 2 matrices of weights that will be learned by the network during the training in an end-to-end fashion (all of them are backpropagated together with the convolutional kernels).

The mathematical expression for this function results then:

$$\vec{s} = \vec{F}_{ex}(\vec{z}, \overrightarrow{W}) = \sigma\left(g(\vec{z}, \overrightarrow{W})\right) = \sigma\left(\overrightarrow{W}_2 \cdot ReLU(\overrightarrow{W}_1 \cdot \vec{z})\right)$$

We already have our squeezed feature maps excited!

See how the excitations is no more than a couple of neural networks that are trained to better calibrate those excitations during the training.

3. Scale

The last step, scaling, is indeed a re-scaling operation. We are going to give the squeezed vector its original shape, keeping the information obtained during the excitation step.

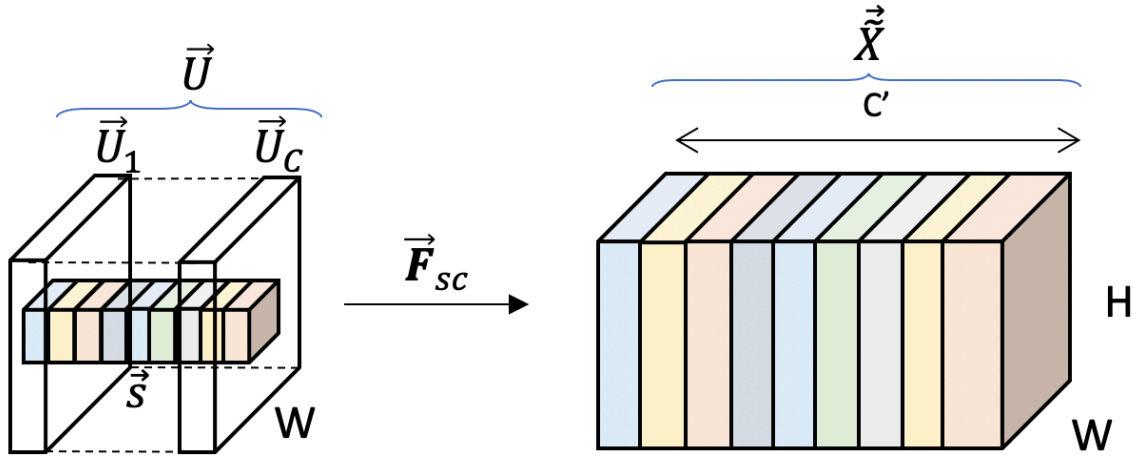


Figure 5. Rescaling

Mathematically, the scaling is achieved by simple scalar product of each channel on the input volume with the corresponding channel on the activated 1x1 squeezed vector.

For 1 channel:

$$\overrightarrow{\tilde{x}_C} = \vec{F}_{ex}(\vec{U}_c, s_c) = \ s_c \cdot \vec{U}_c$$

Discussion from the authors:

The activations act as channel weights adapted to the input-specific descriptor $\vec{z}$.

Application of SE Block

As was introduced as one of the innovations, SE block is not a new neural network architecture. It attempts to improve already existing models by giving them more sensitivity to channel-wise relationships.

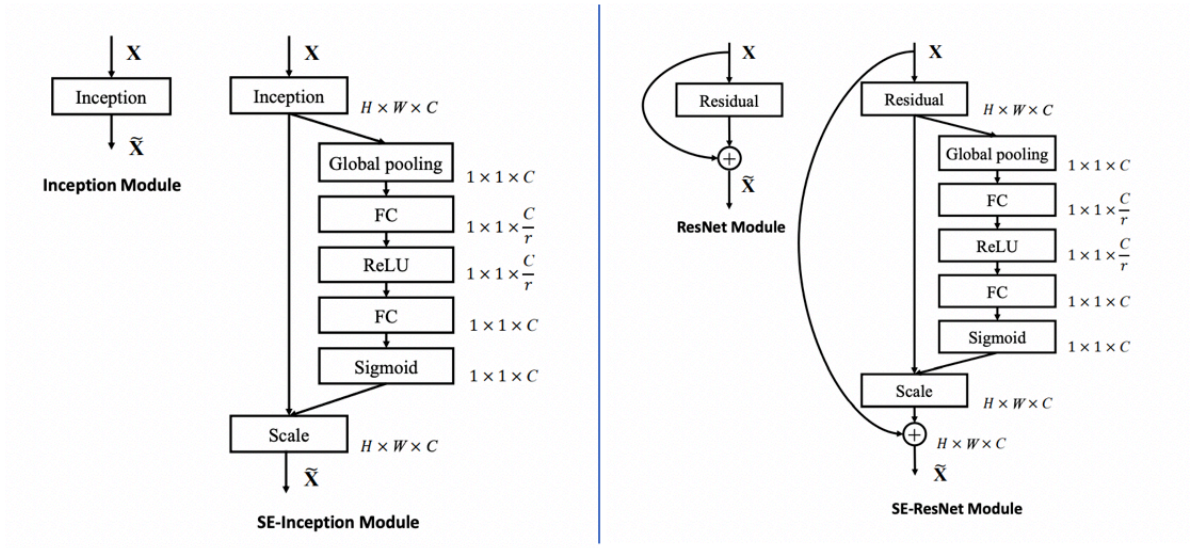


Figure 6. SE block applied to Inception (left) and ResNets (right) modules

Figure 6 shows how the same SE block is applied right after the output volume of a convolutional layer, to enrich its representation before moving it to the next layer. In this case, Inception and ResNets modules are shown.

Results and Conclusions

The results confirm that introducing SE blocks in the state-of-art networks improve their performance in different computer vision applications: ImageNet classification, scene classification (Places365-Challenge dataset) and object detection (COCO dataset). The results are too broad to be included in this summary and I recommend going over them in the original paper.

The empirical study value of the **reduction factor** comes with a interesting observation. It turns out that 16 is a good value and beyond that, performance does not improve monotically with the capacity of the model.

It is likely that SE block can overfit the channel interdependencies

---

Lastly, the authors perform the average activation for fifty uniformly sampled channel at each stage (or layer) where the SE block was introduced for 5 significantly different classes. It has been observed that the distribution across different classes is nearly **identical in the lower layers**, whereas the value of each channel becomes more **class-specific at greater depth**.

Global Picture