Inception

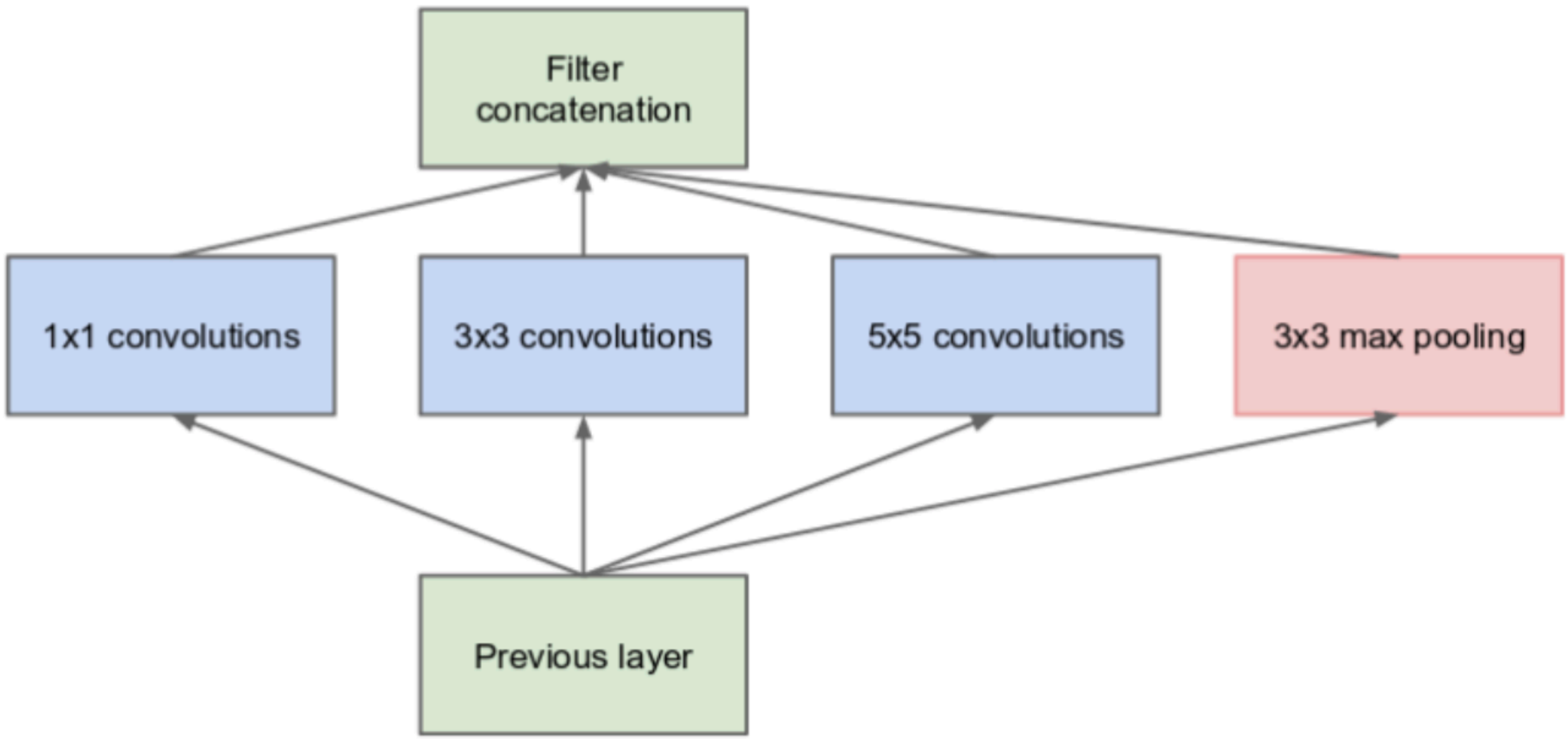
Cho Sung Man

Contents

- Inception v1
- Inception v2, v3
- Inception v4

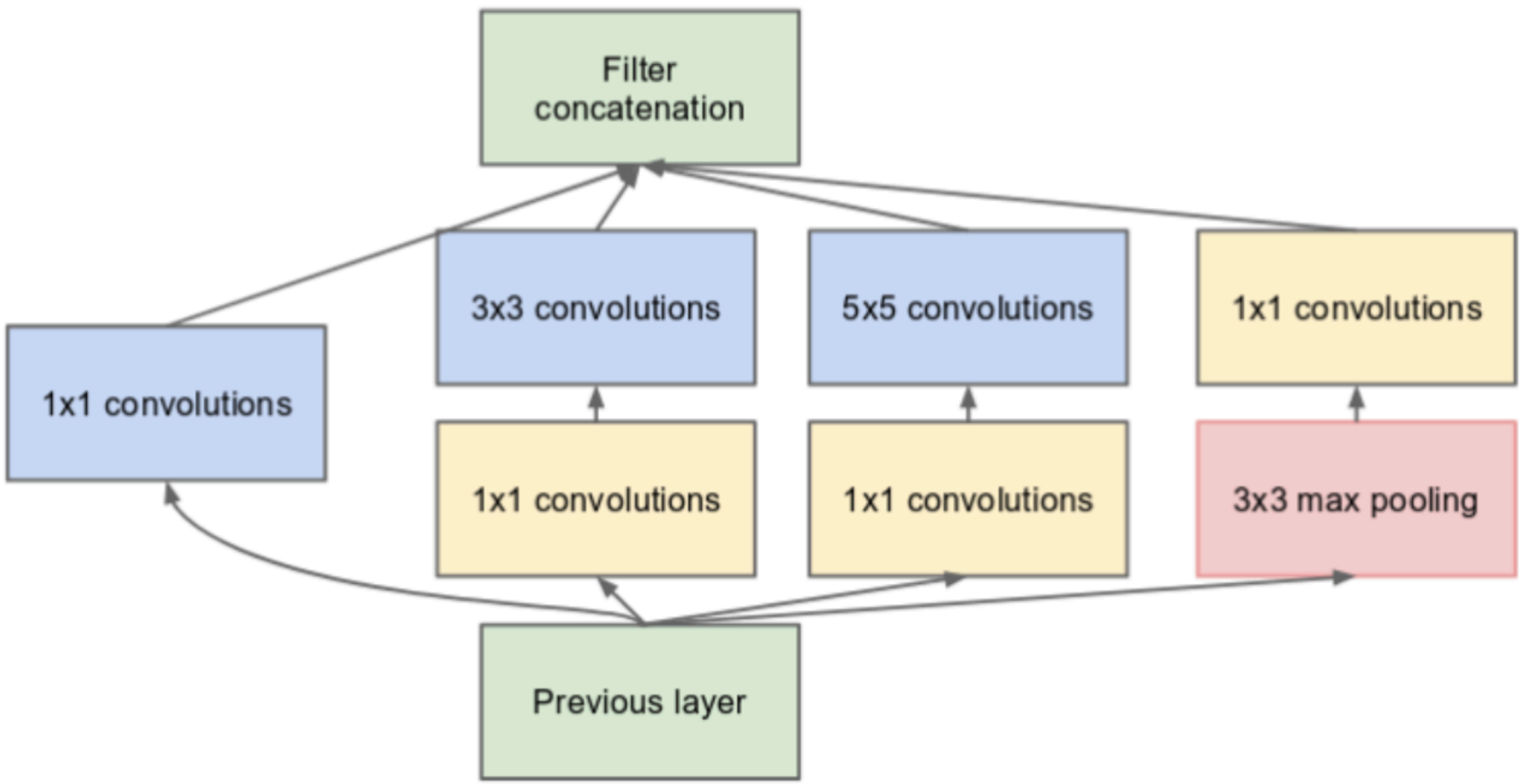
Inception v1

Naive Version



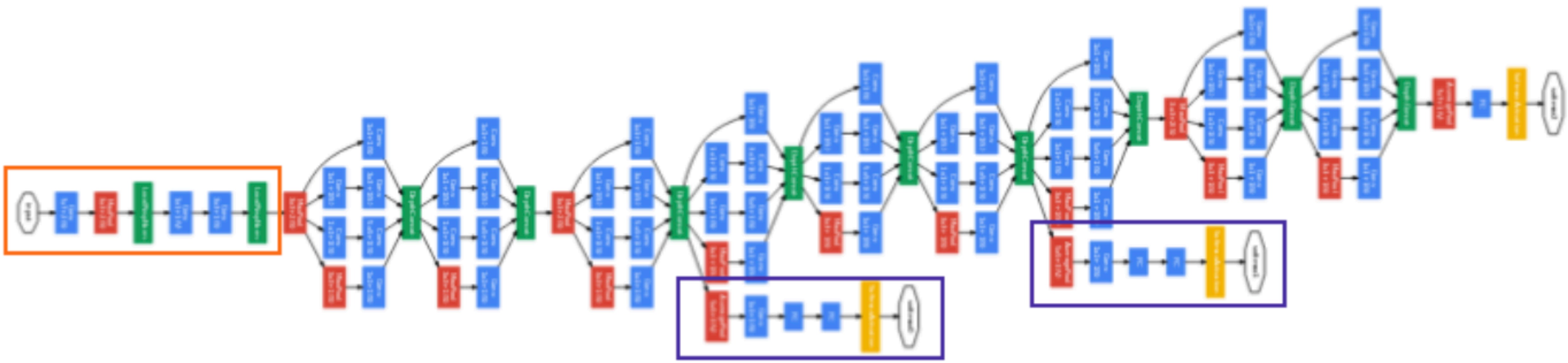
(a) Inception module, naïve version

Dimension Reduction



(b) Inception module with dimension reductions

Auxillary Classifiers



```
# The total loss used by the inception net during training.
total_loss = real_loss + 0.3 * aux_loss_1 + 0.3 * aux_loss_2
```

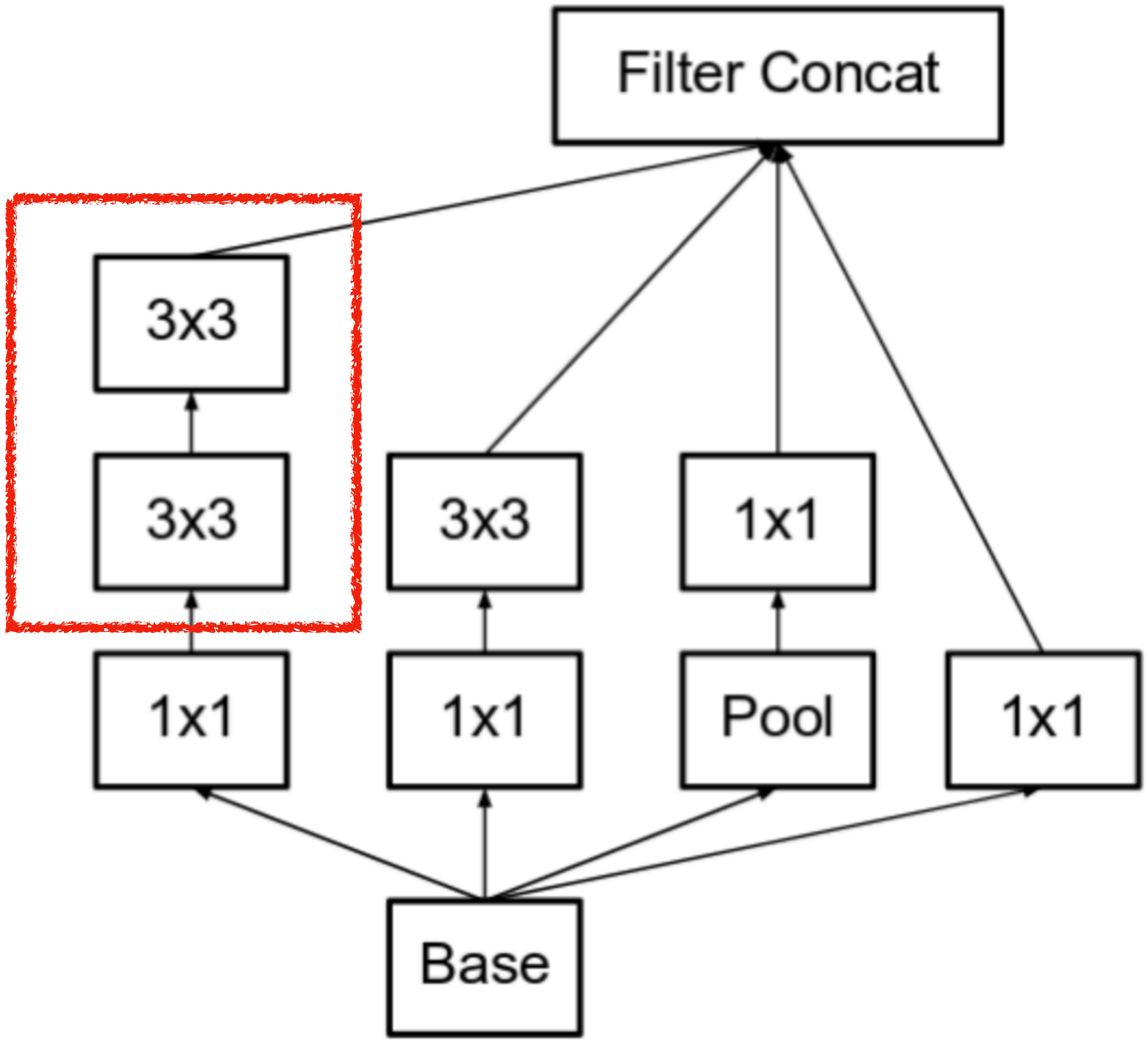
Inception v2

The Premise

- Reduce representational bottleneck. The intuition was that, neural networks perform better when convolutions didn't alter the dimensions of the input drastically. **Reducing the dimensions too much may cause loss of information**, known as a "**representational bottleneck**"

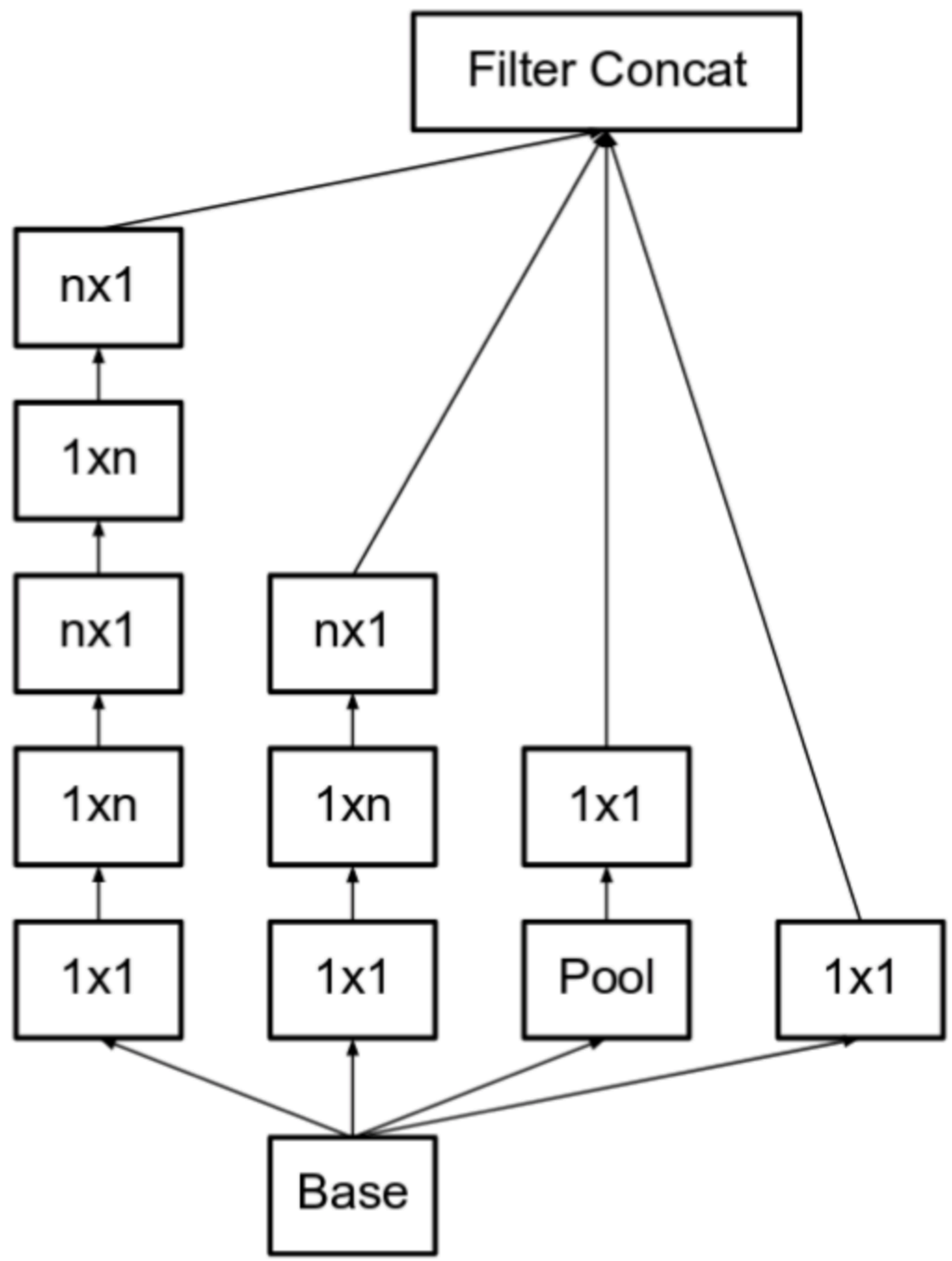
- Using **smart factorization methods**, convolutions can be made more efficient in terms of computational complexity.

Computational Efficiency



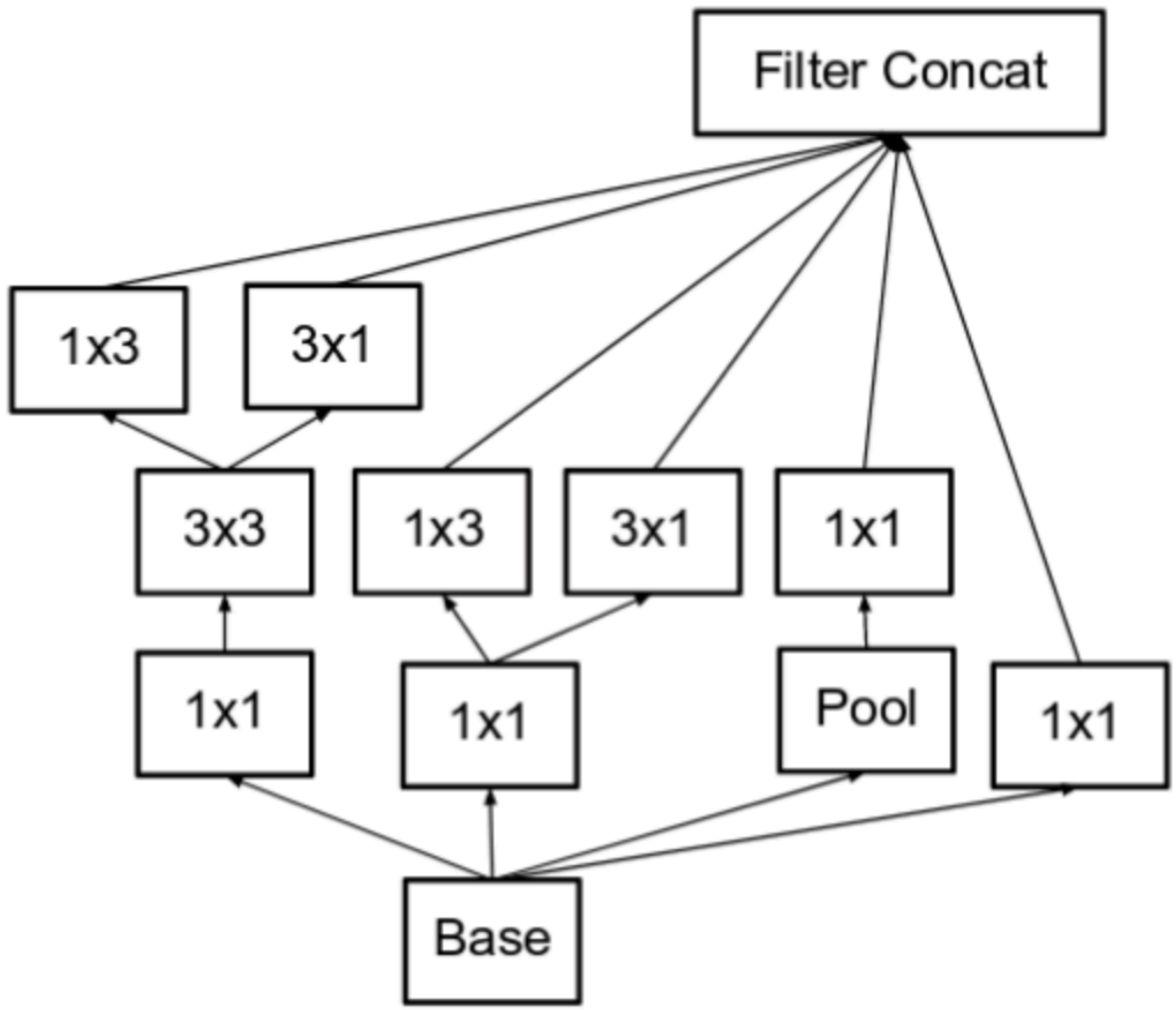
5x5 conv. is expensive 2.78 times more expensive than a 3x3 conv.

Computational Efficiency



33% cheaper !

Representational Bottleneck



\

Inception v2

type	patch size/stride or remarks	input size
conv	$3{\times}3/2$	$299{\times}299{\times}3$
conv	$3{\times}3/1$	$149{\times}149{\times}32$
conv padded	$3{\times}3/1$	$147{\times}147{\times}32$
pool	$3{\times}3/2$	$147{\times}147{\times}64$
conv	$3{\times}3/1$	$73{\times}73{\times}64$
conv	$3{\times}3/2$	$71{\times}71{\times}80$
conv	$3{\times}3/1$	$35{\times}35{\times}192$
$3{\times}$Inception	As in figure $\boxed{5}$	$35{\times}35{\times}288$
$5{\times}$Inception	As in figure $\boxed{6}$	$17{\times}17{\times}768$
$2{\times}$Inception	As in figure $\boxed{7}$	$8{\times}8{\times}1280$
pool	8×8	$8 \times 8 \times 2048$
linear	logits	$1 \times 1 \times 2048$
softmax	classifier	$1 \times 1 \times 1000$

\

Inception v3

The Premise

- The authors noted that the **auxiliary classifiers** didn't contribute much until near the end of the training process, when accuracies were nearing saturation. They argued that they function as **regularizes**, especially if they have BatchNorm or Dropout operations.

- Possibilities to improve on the Inception v2 without drastically changing the modules were to be investigated.

The Solution

- RMS optimizer.
- Factorized 7x7 convolution filter.
- BatchNorm in the Auxillary Classifiers.
- Label Smoothing (regularization)

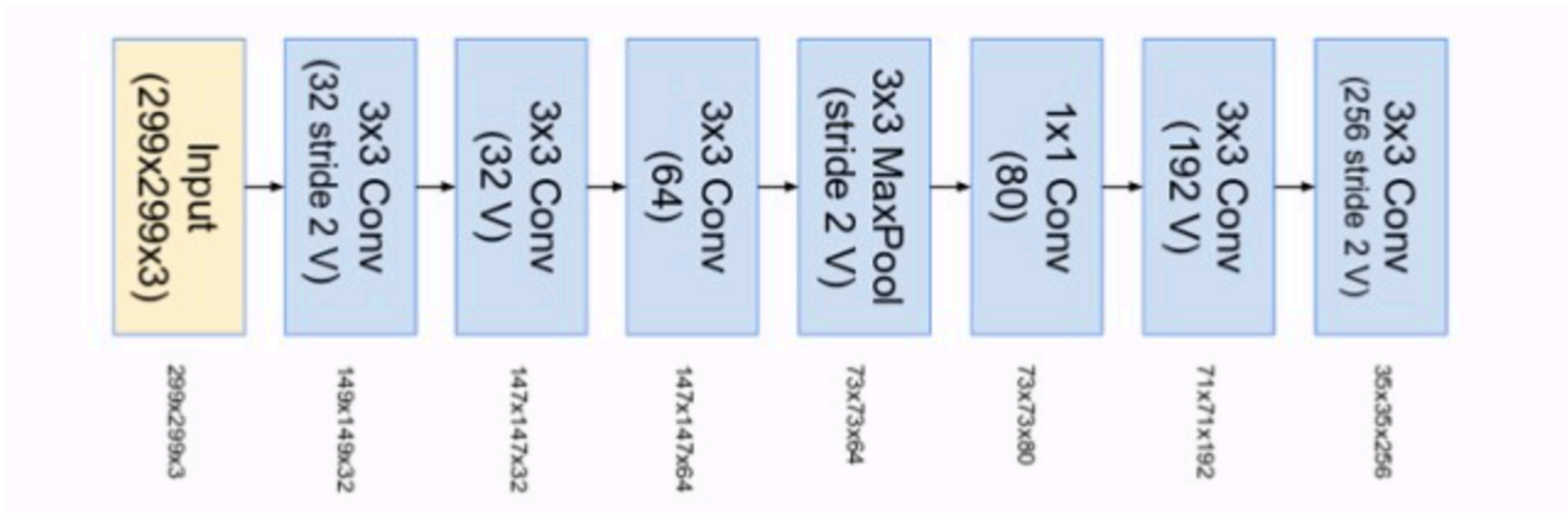
Inception v4

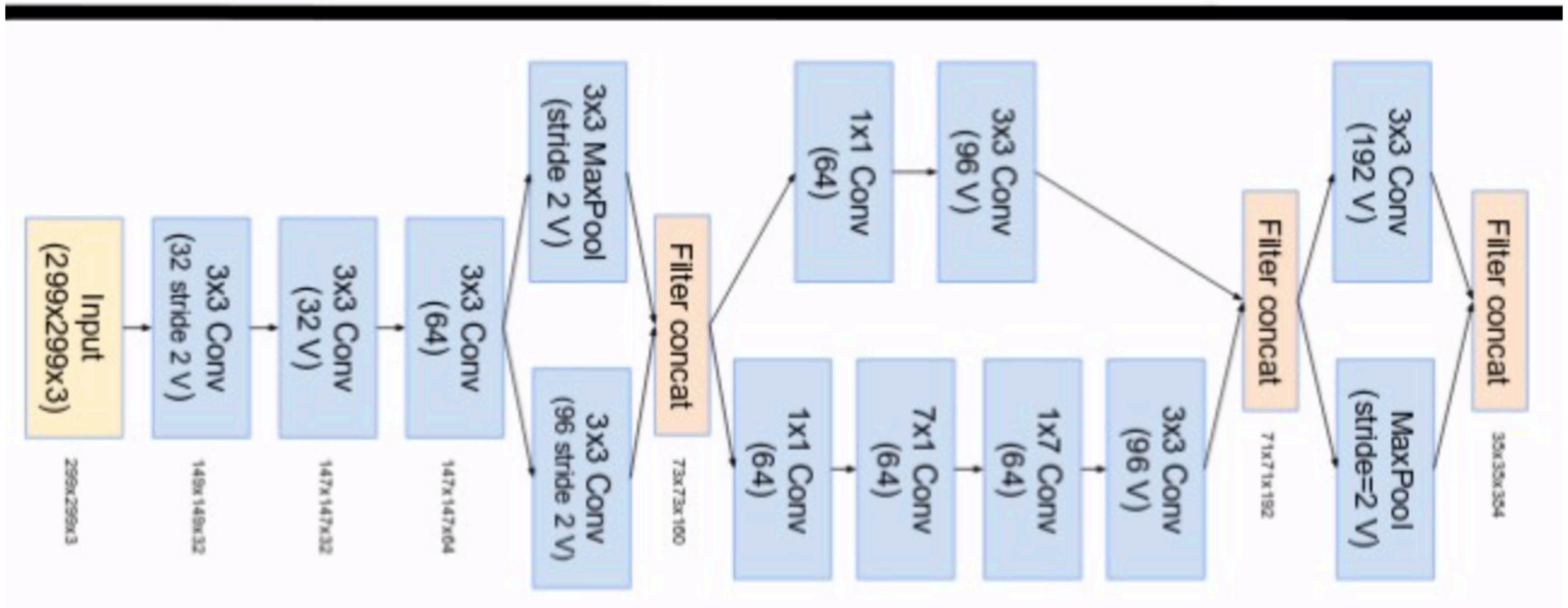
The Premise

- Make the modules more **uniform**. The authors also noticed that some of the modules were **more complicated than necessary**. This can enable us to boost performance by adding more of these uniform modules.

Stem

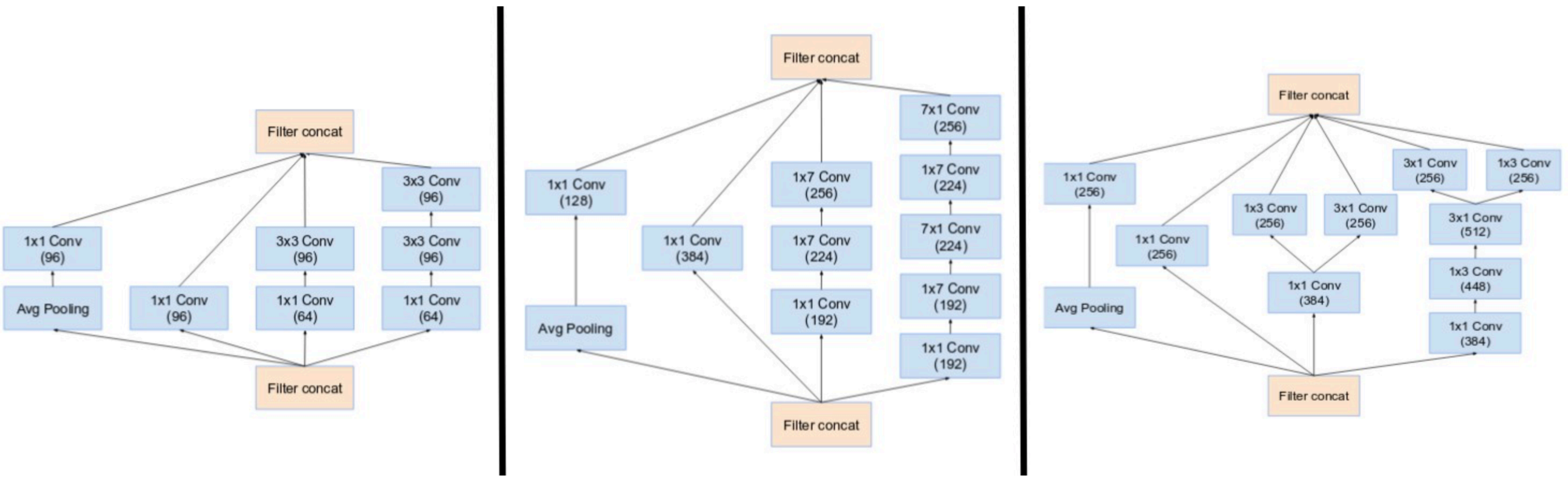
Inception-ResNet-V1



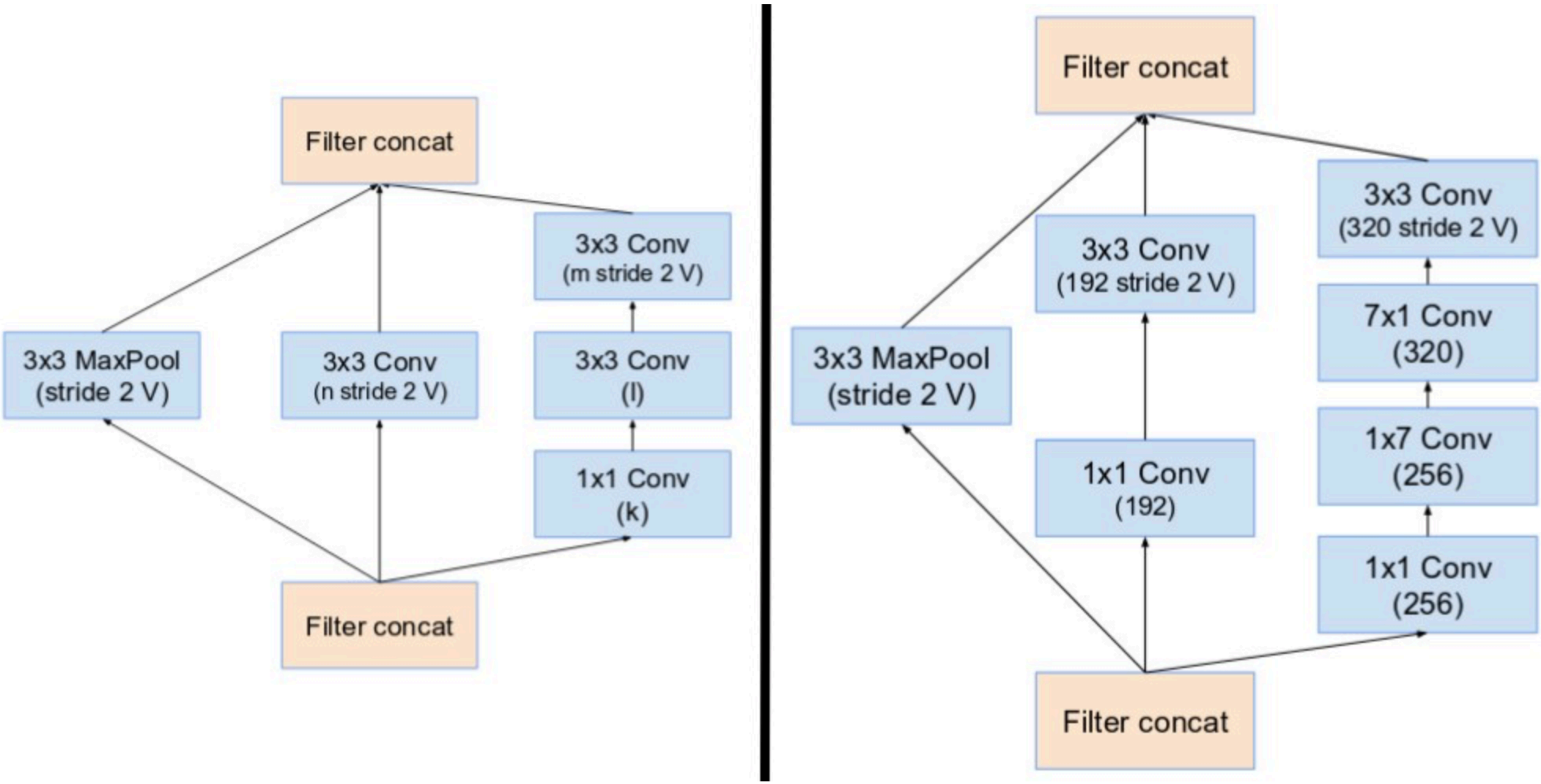


Inception-ResNet-V2, Inception V4

Inception Module



Reduction Block



Inception ResNet v1,2

The Premise

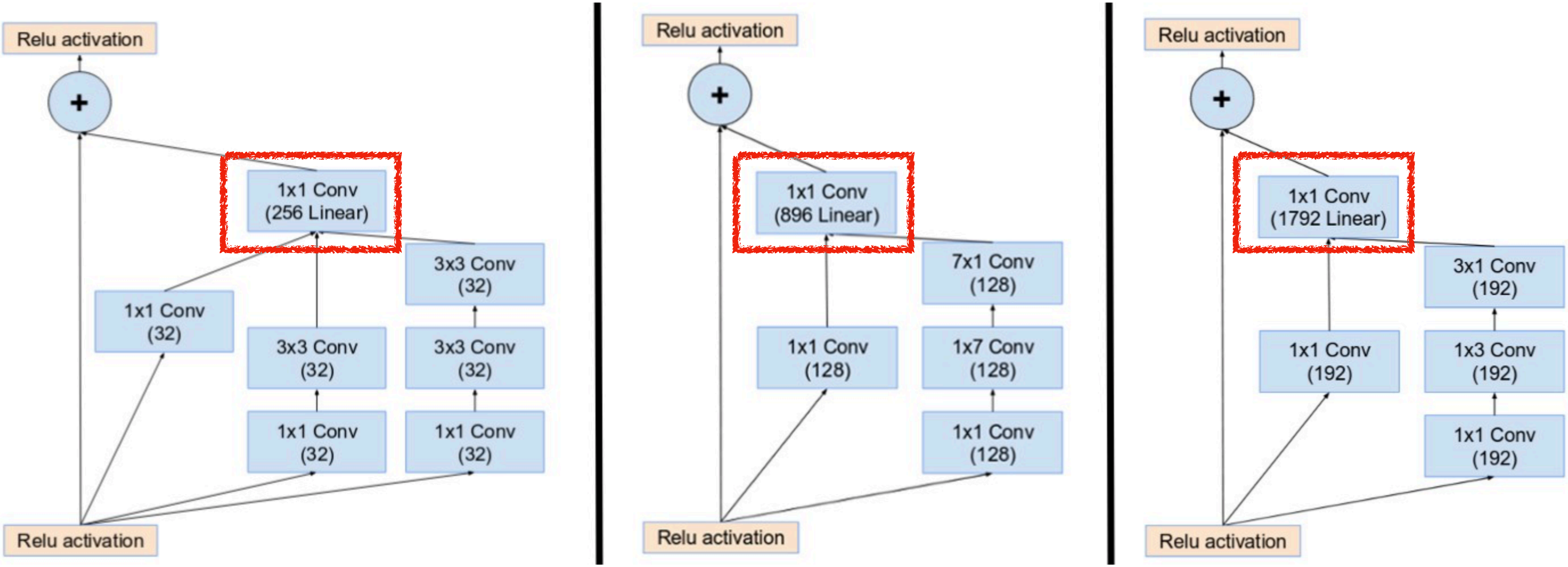
- Inception-ResNet v1 has a computational cost that is similar to that of Inception v3.

- Inception-ResNet v2 has a computational cost that is similar to that of Inception v4.

- They have **different stems**, as illustrated in the Inception v4 section.

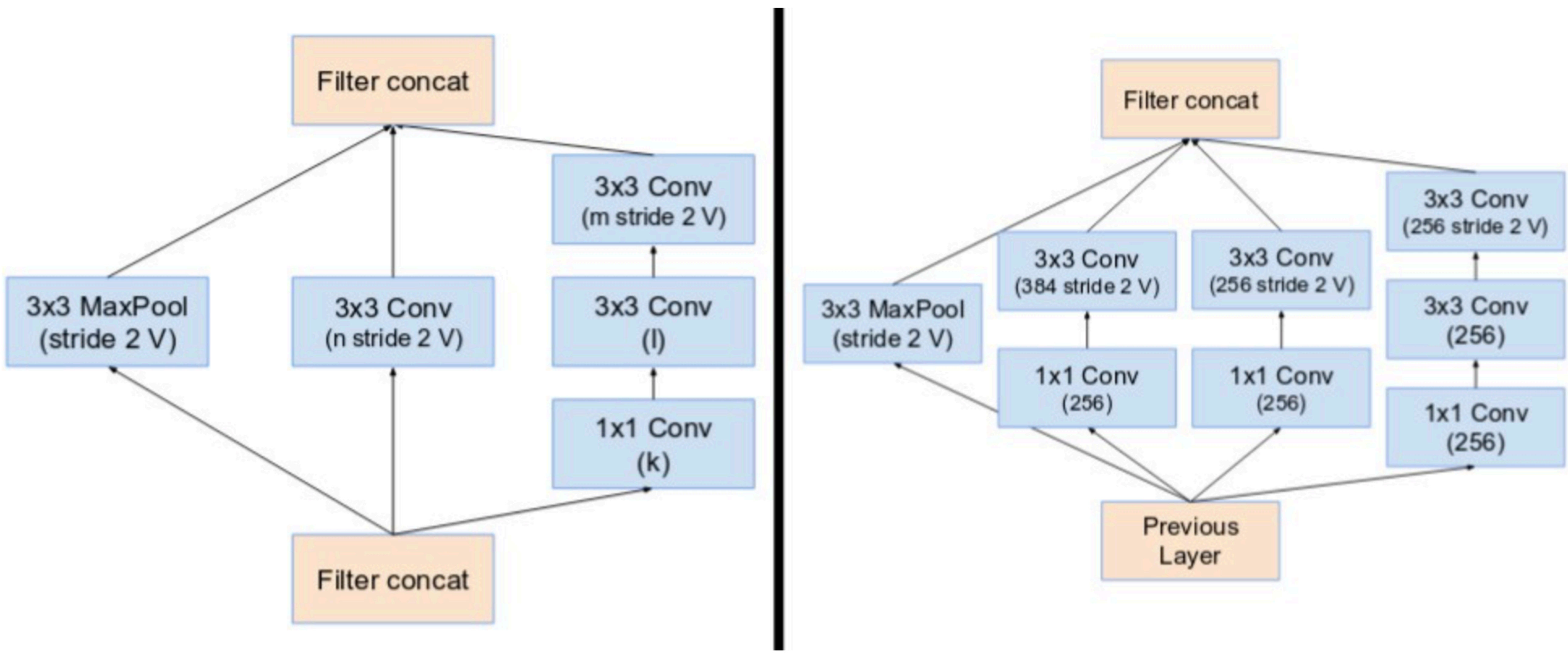
- Both sub-versions have the **same structure** for the **modules A, B, C** and the **reduction blocks**. Only **difference** is the **hyper-parameter** settings.

Residual Connection

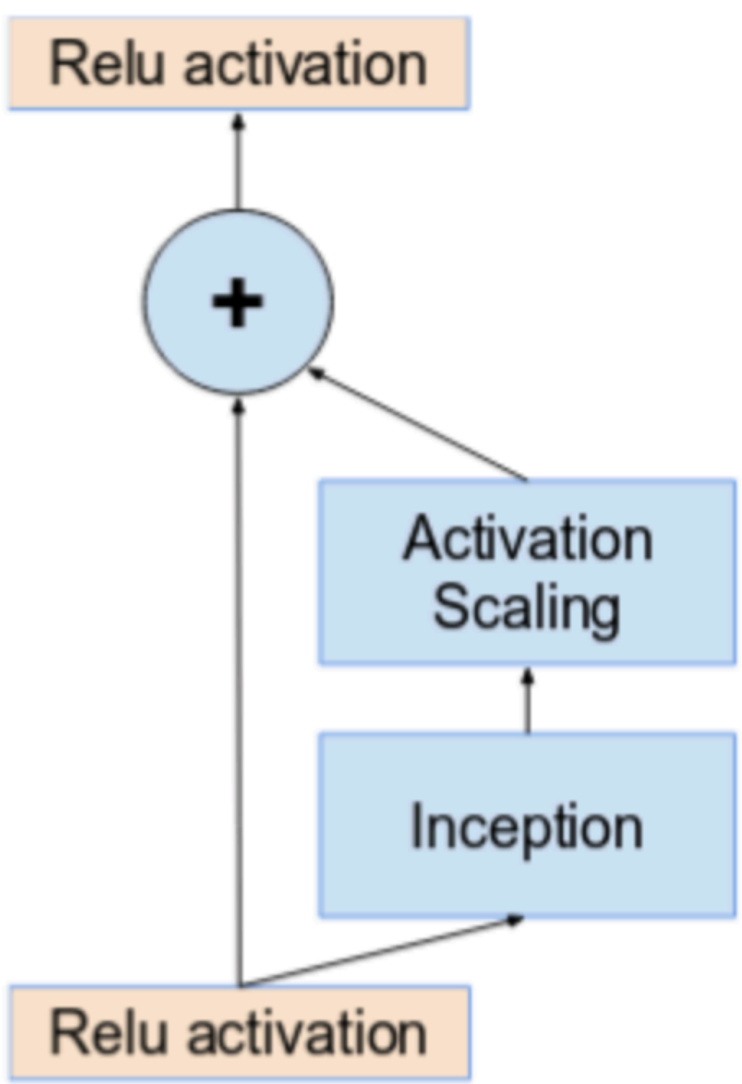


Dimension Matching

Reduction Block



Activation Scaling

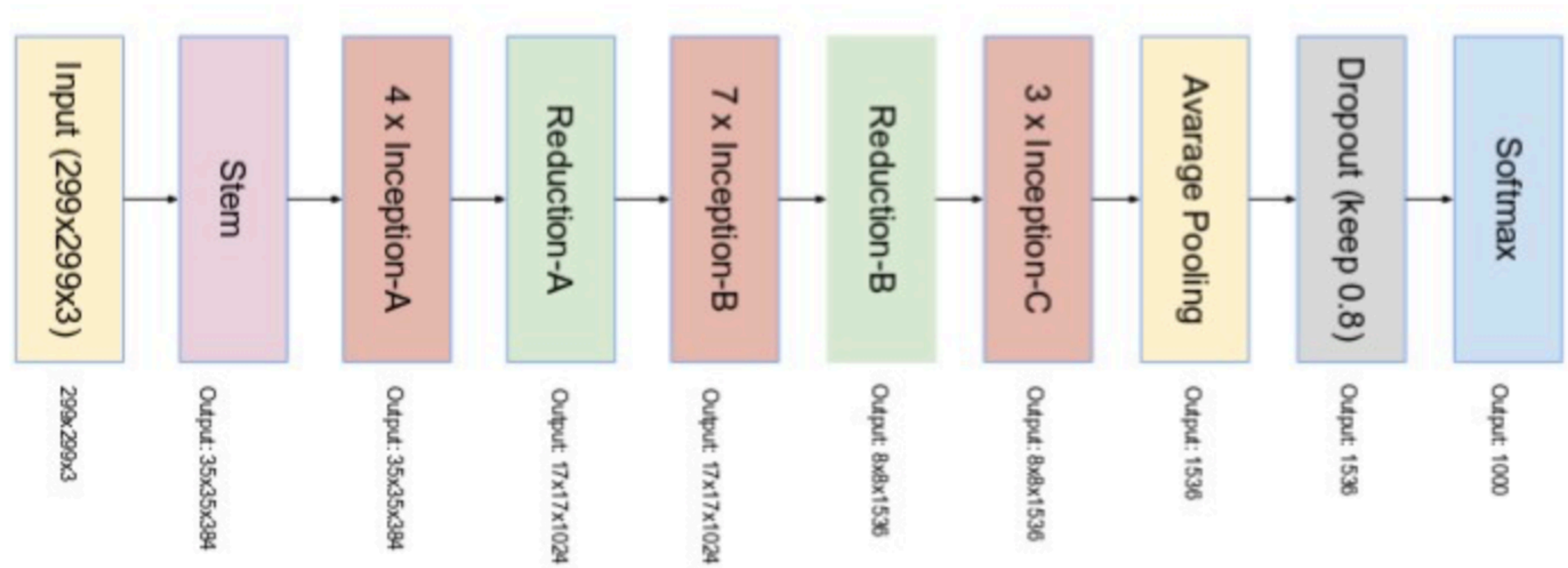


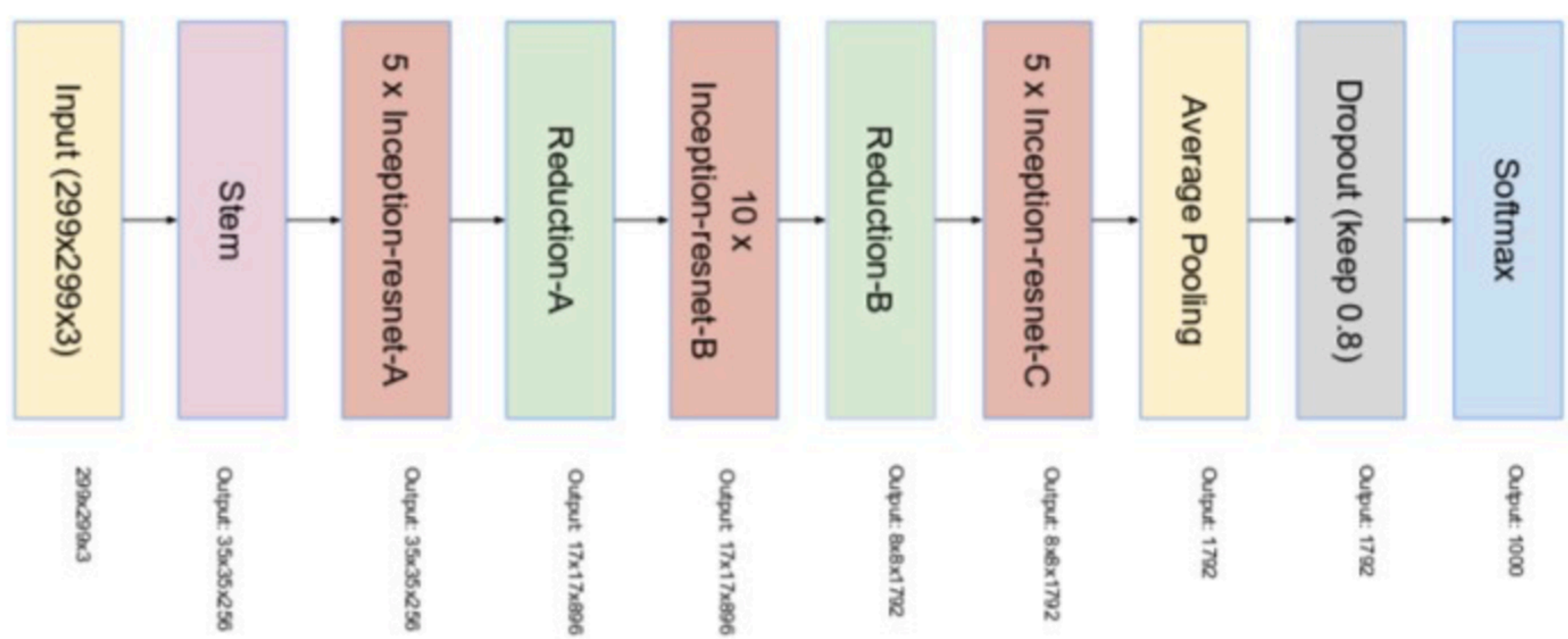
Networks with residual units deeper in the architecture **caused the network to "die" if the number of filters exceeded 1000.**

scaled the residual activations by a value around 0.1 to 0.3.

Architecture

Inception v4





Inception-ResNet

Thank You.