

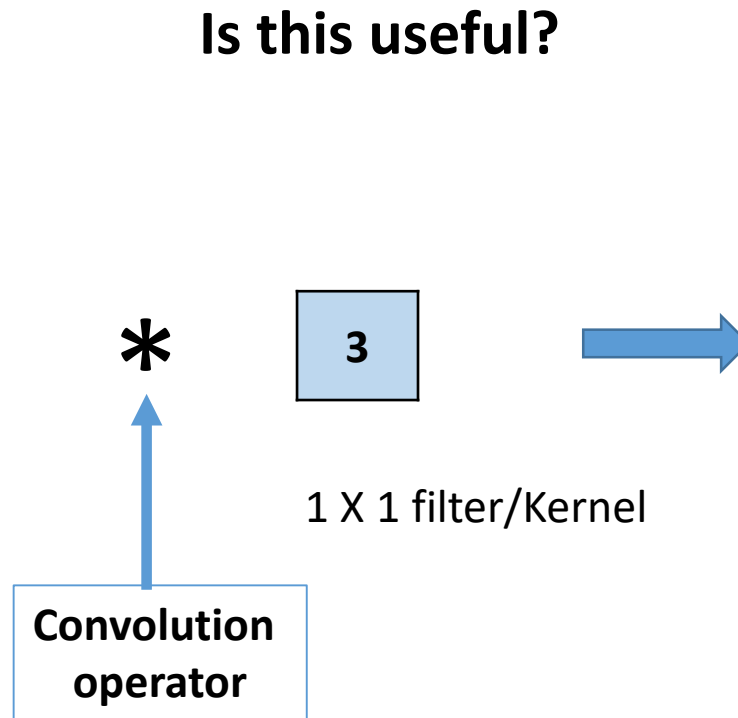
Deep Learning and Convolutional Neural Network (42028)

Inception, GoogleNet and ResNet

Foundation: 1 X 1 convolution

| | | | | | |
|-----|-----|-----|---|---|---|
| 100 | 100 | 100 | 0 | 0 | 0 |
| 100 | 100 | 100 | 0 | 0 | 0 |
| 100 | 100 | 100 | 0 | 0 | 0 |
| 100 | 100 | 100 | 0 | 0 | 0 |
| 100 | 100 | 100 | 0 | 0 | 0 |
| 100 | 100 | 100 | 0 | 0 | 0 |

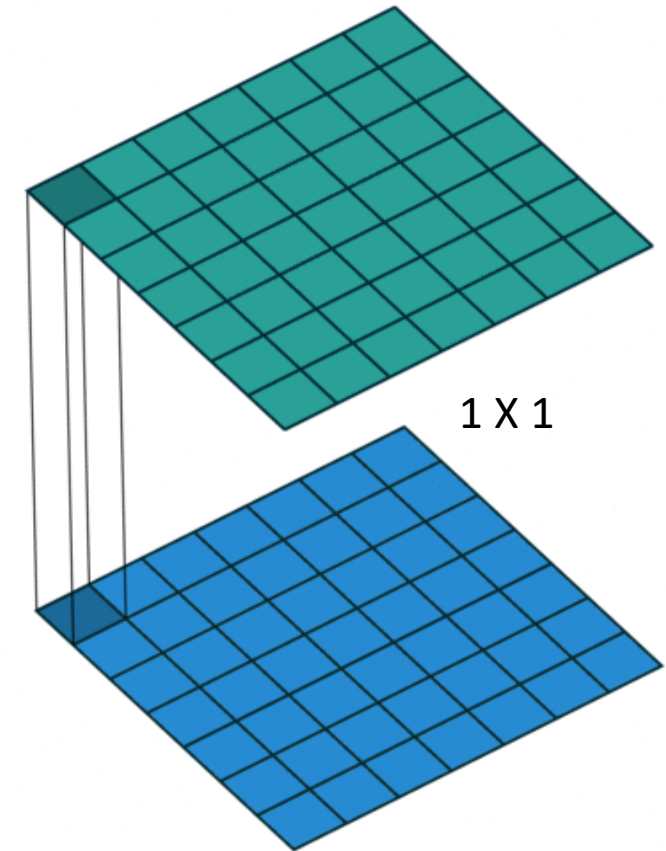
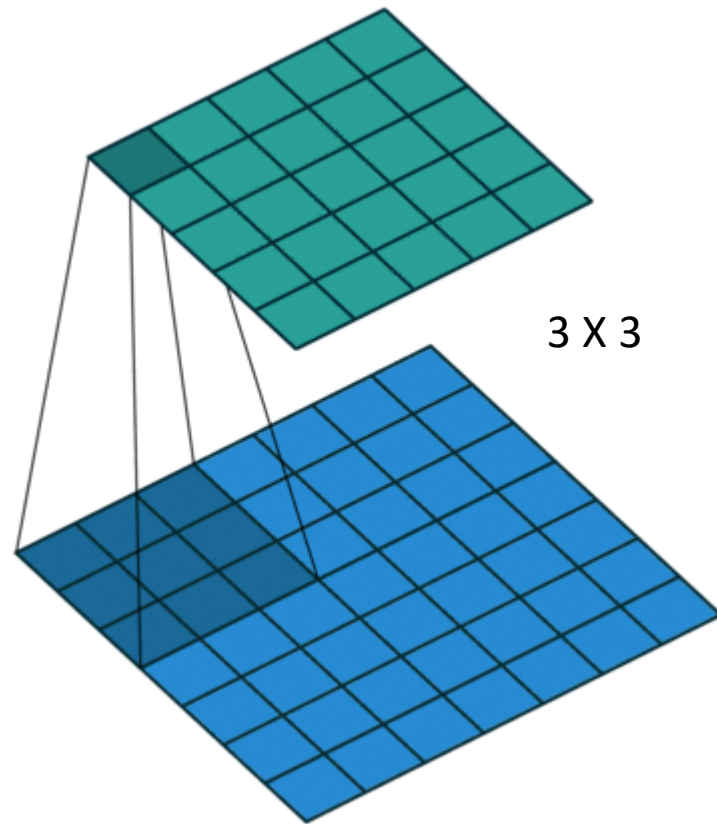
6 X 6 X 1 dimension image



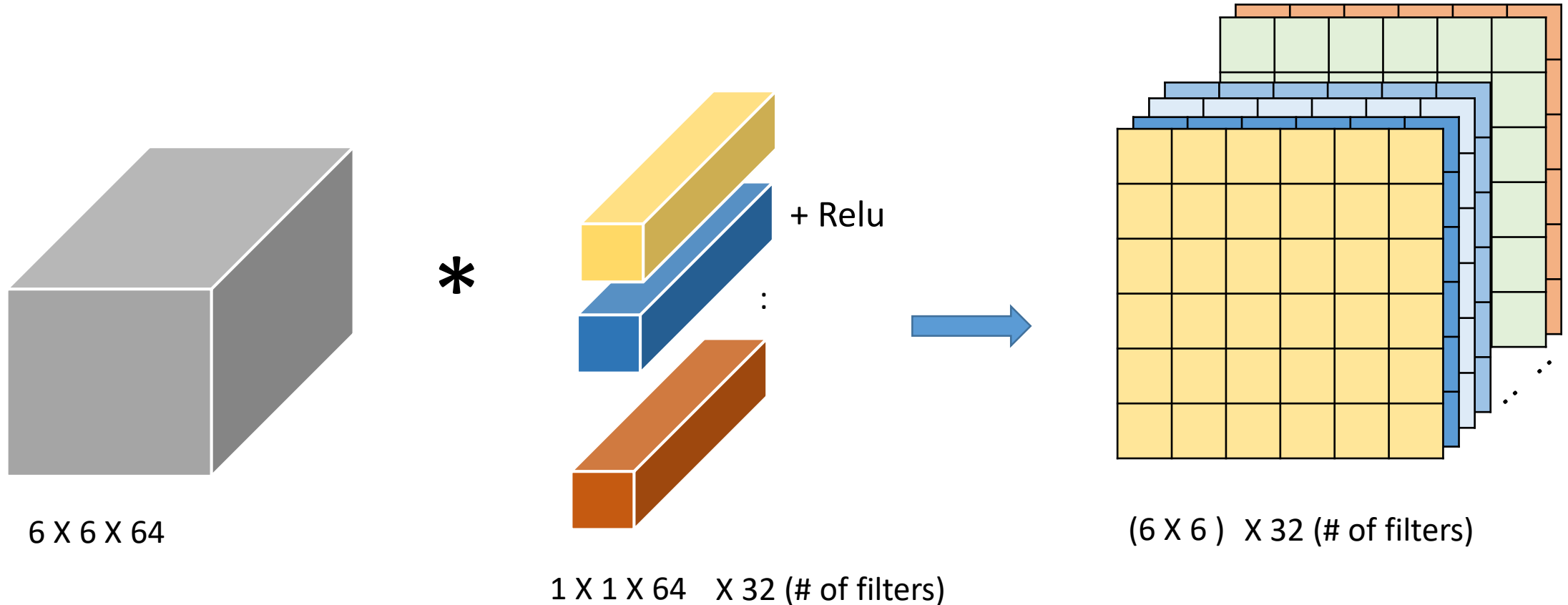
| | | | | | |
|-----|-----|-----|---|---|---|
| 300 | 300 | 300 | 0 | 0 | 0 |
| 300 | 300 | 300 | 0 | 0 | 0 |
| 300 | 300 | 300 | 0 | 0 | 0 |
| 300 | 300 | 300 | 0 | 0 | 0 |
| 300 | 300 | 300 | 0 | 0 | 0 |
| 300 | 300 | 300 | 0 | 0 | 0 |

6 X 6 X 1 dimension volume

Foundation: 1 X 1 convolution

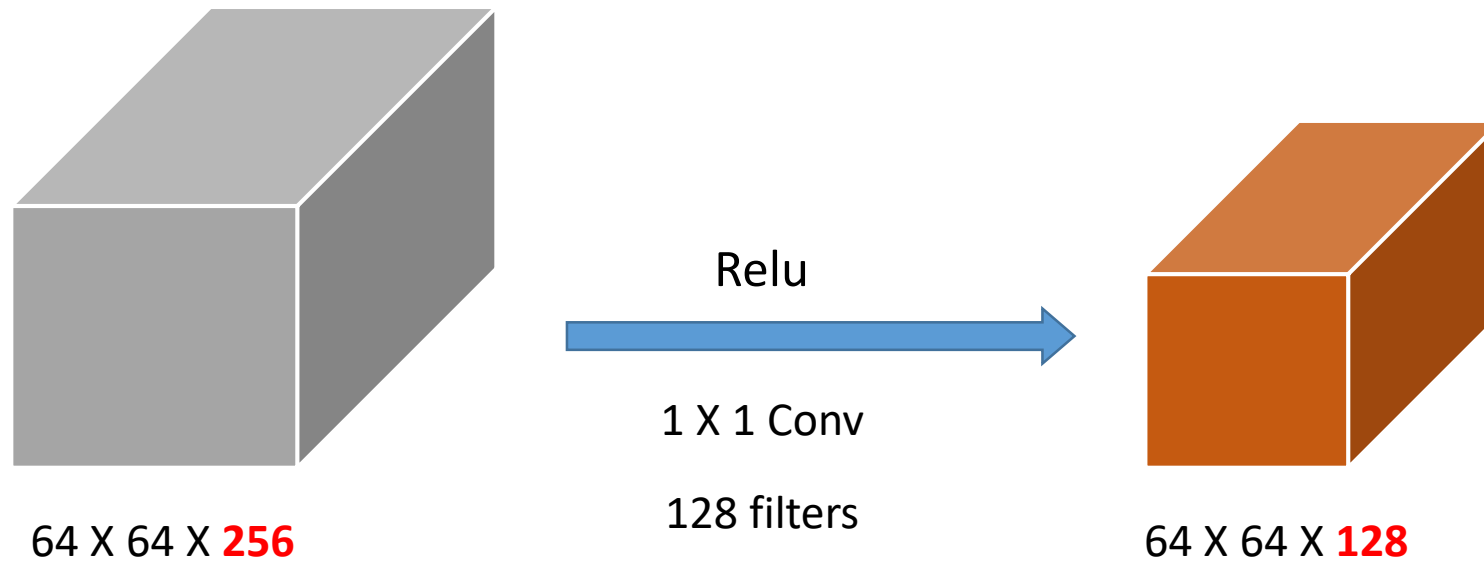


Foundation: 1 X 1 convolution



So, $(6 \times 6 \times 64) \rightarrow (6 \times 6 \times 32)$... reduced!

Foundation: 1 X 1 convolution



Inception - Motivation



- Large variation in object size
- How to choose the right filter size?

- Large filter preferred for large objects
- Small filters for small objects

Inception - Motivation

Designing CNN requires:

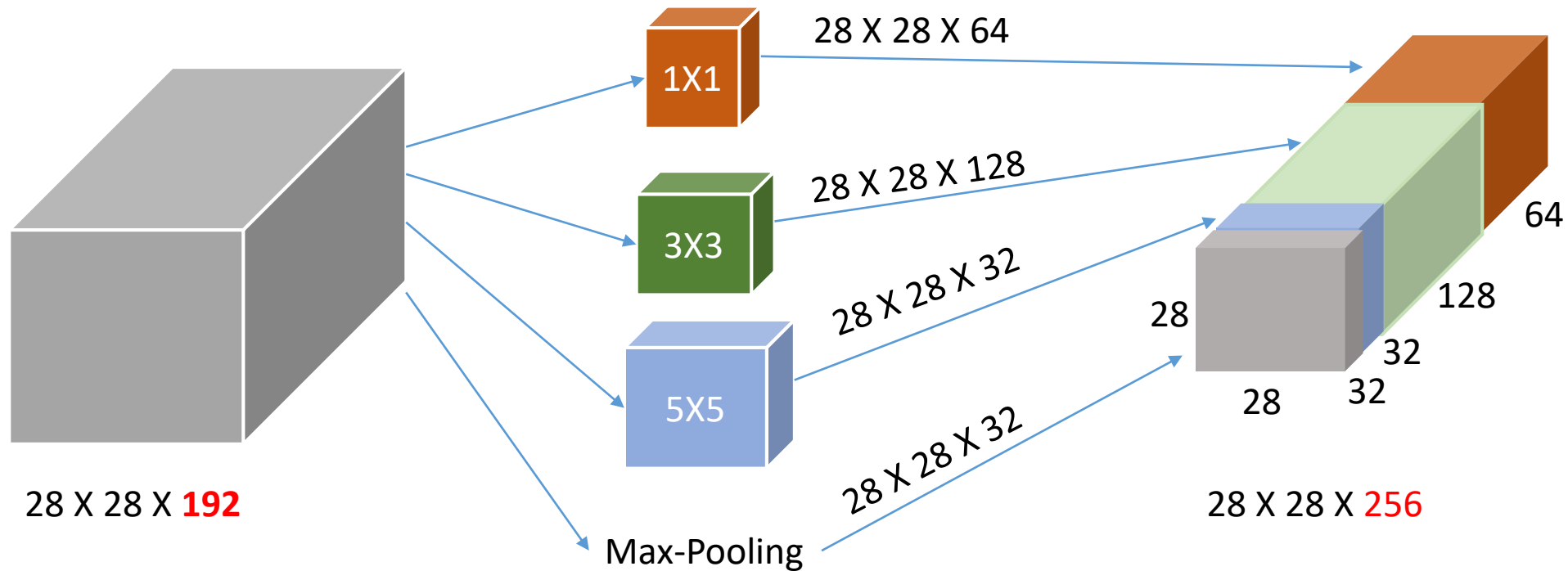
- Deciding filter size and number
- Number and type of layers etc.

Inception suggests:

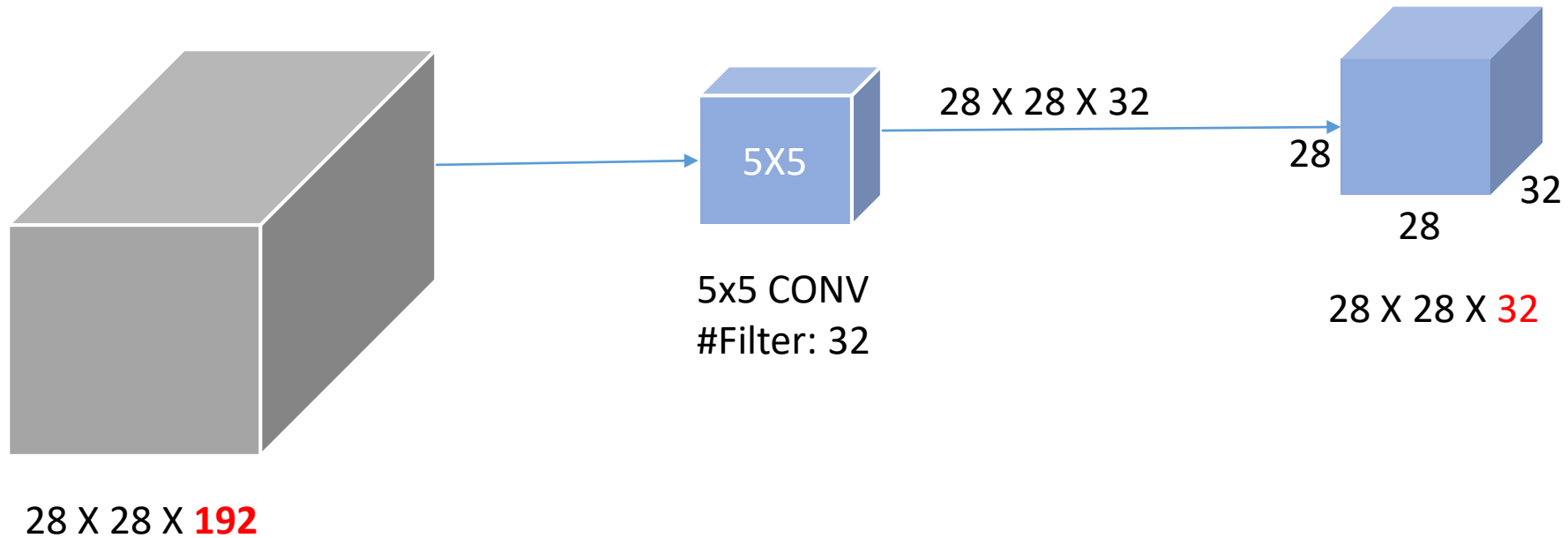
- Use filters with different size together!
- Use different types of layers (CONV, POOL etc.) together

Result → Complicated Architecture! & better performance

Inception - Motivation



Computation cost

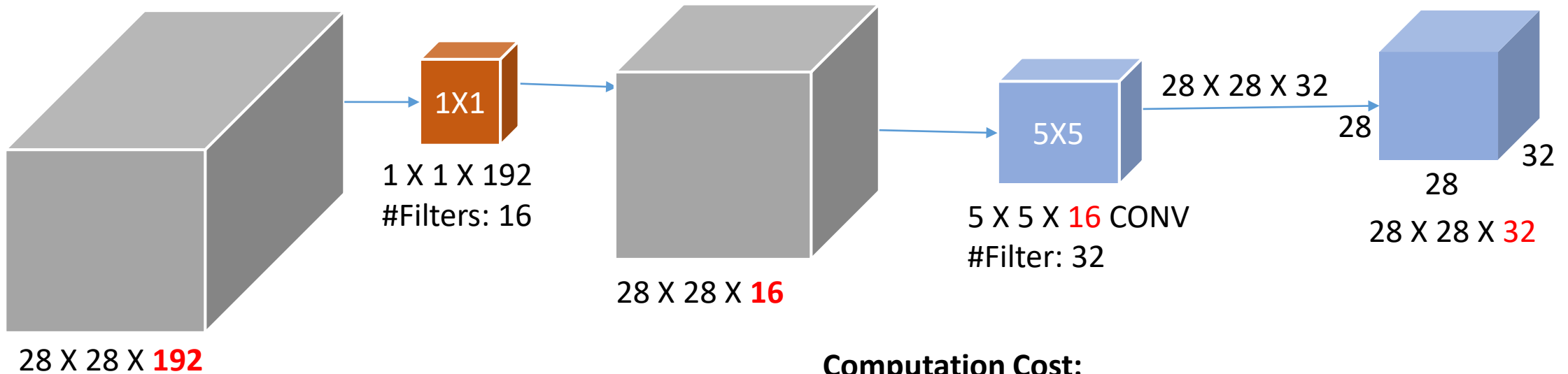


Computation Cost:

$28 \times 28 \times 32 \times 5 \times 5 \times 192 \approx 120\text{M}$ multiplications!

Quite expensive !

Reduce Computation cost using 1X1 CONV



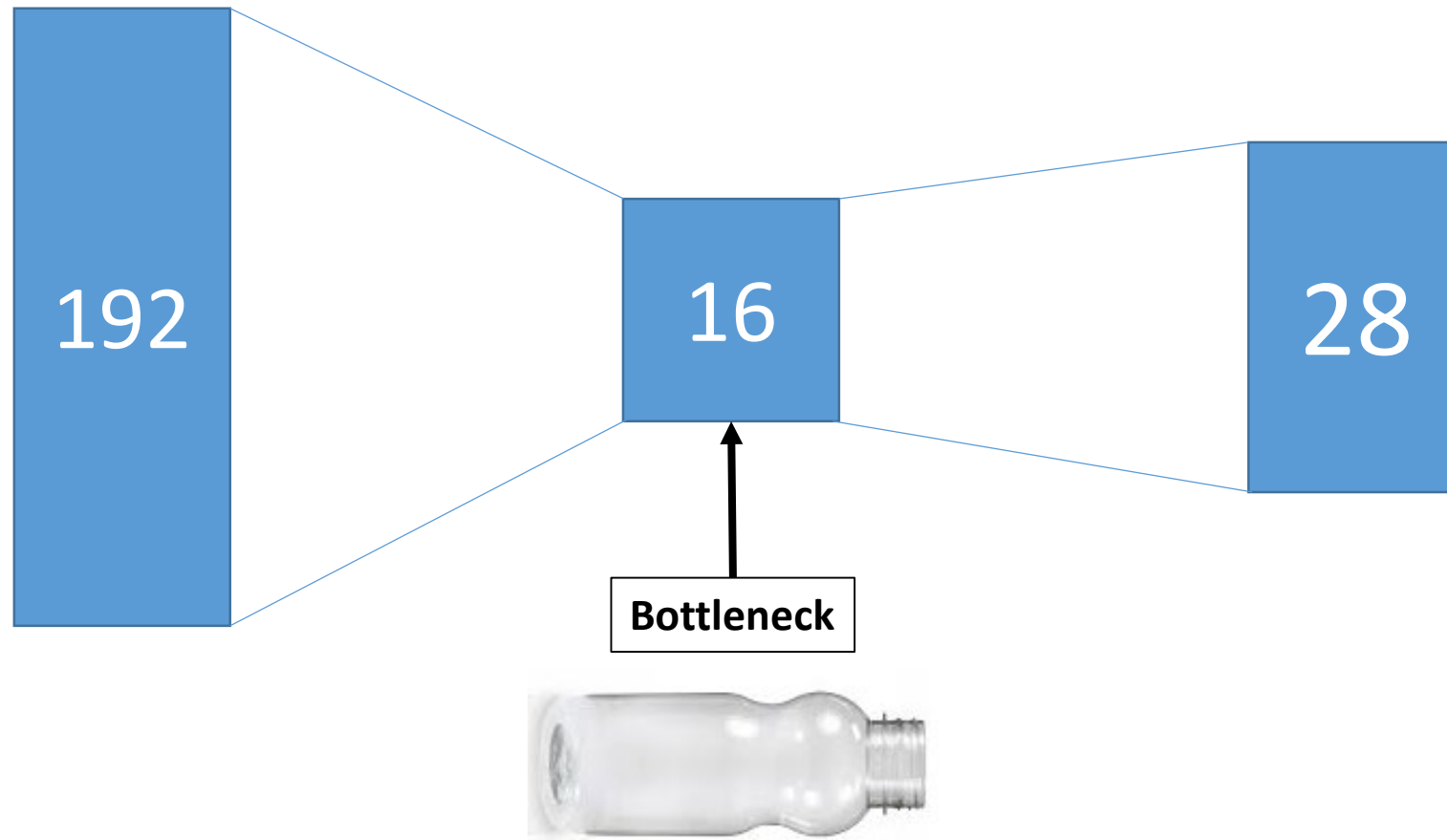
Computation Cost:

1X1: $28 \times 28 \times 16 \times 192 \approx 2.4M$ multiplications!

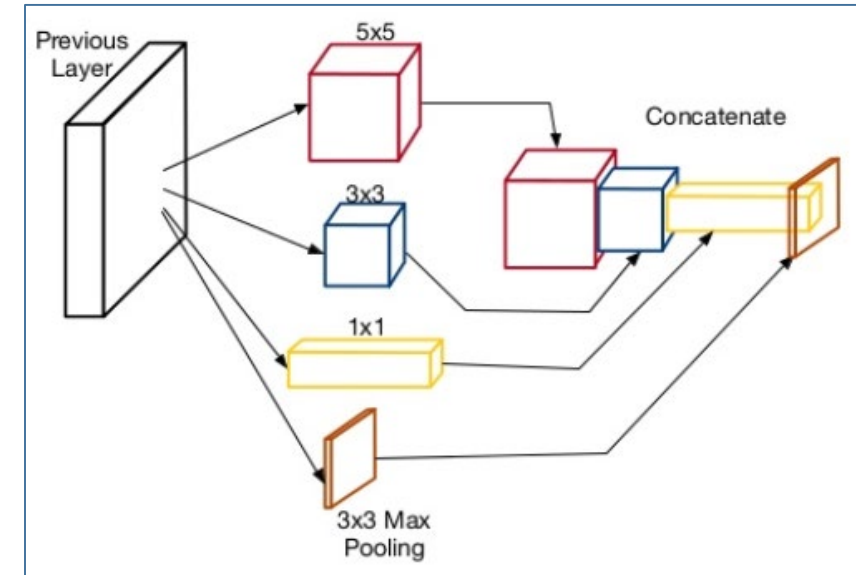
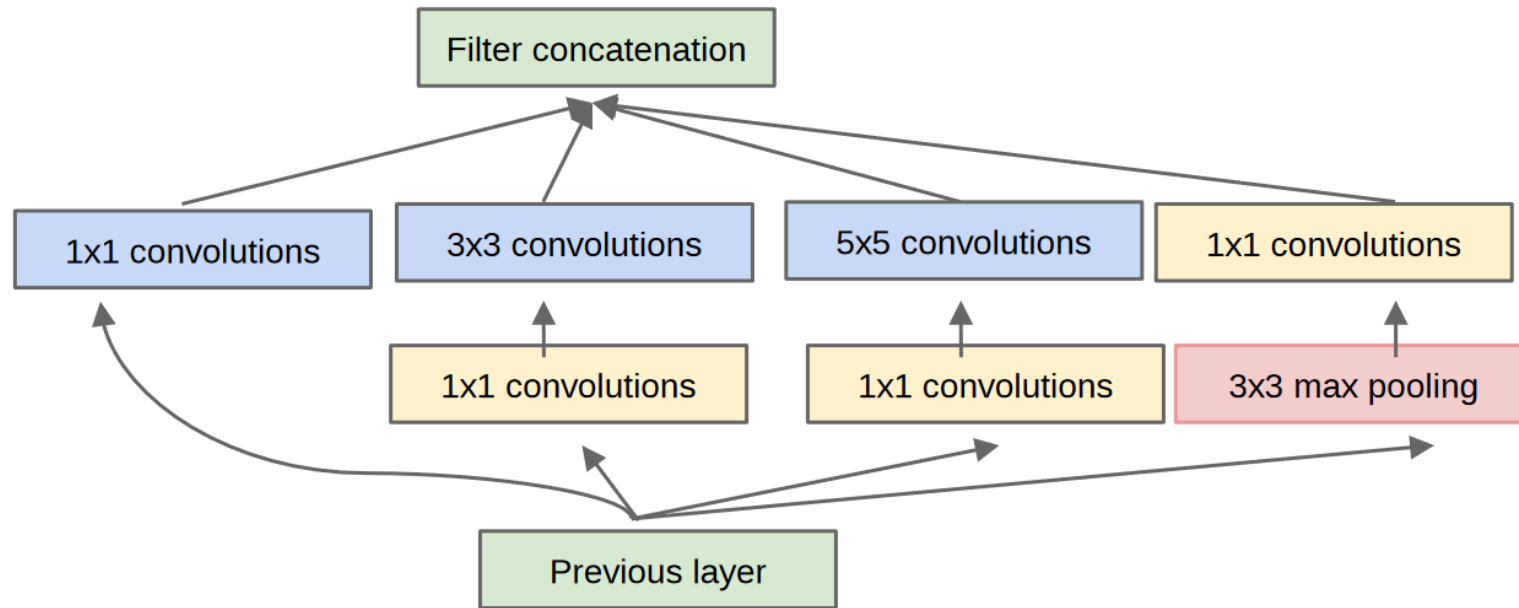
5X5: $28 \times 28 \times 32 \times 5 \times 5 \times 16 \approx 10M$ multiplications!

Total : **12.4M multiplications!** ← Reduced by 10 times!

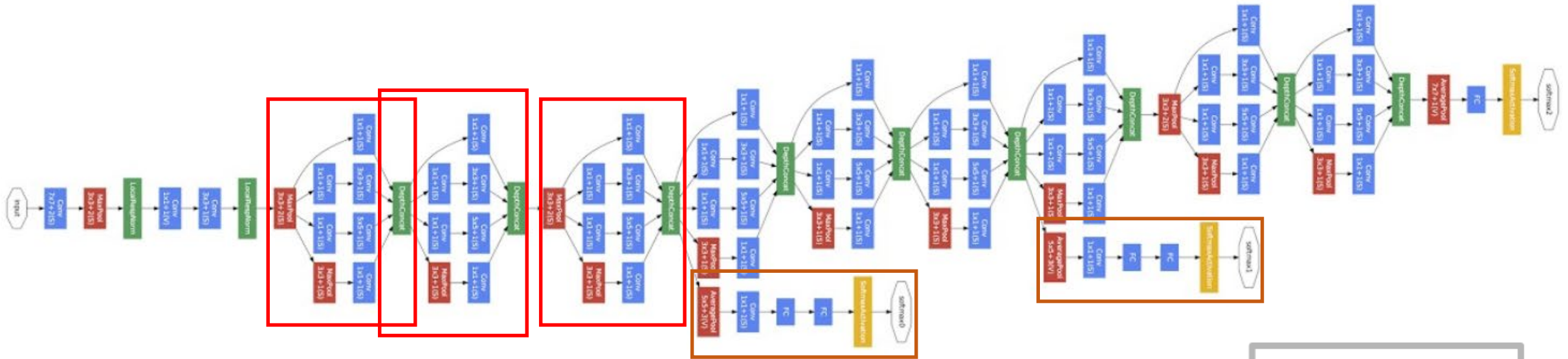
Bottleneck Layer



Inception Module V1



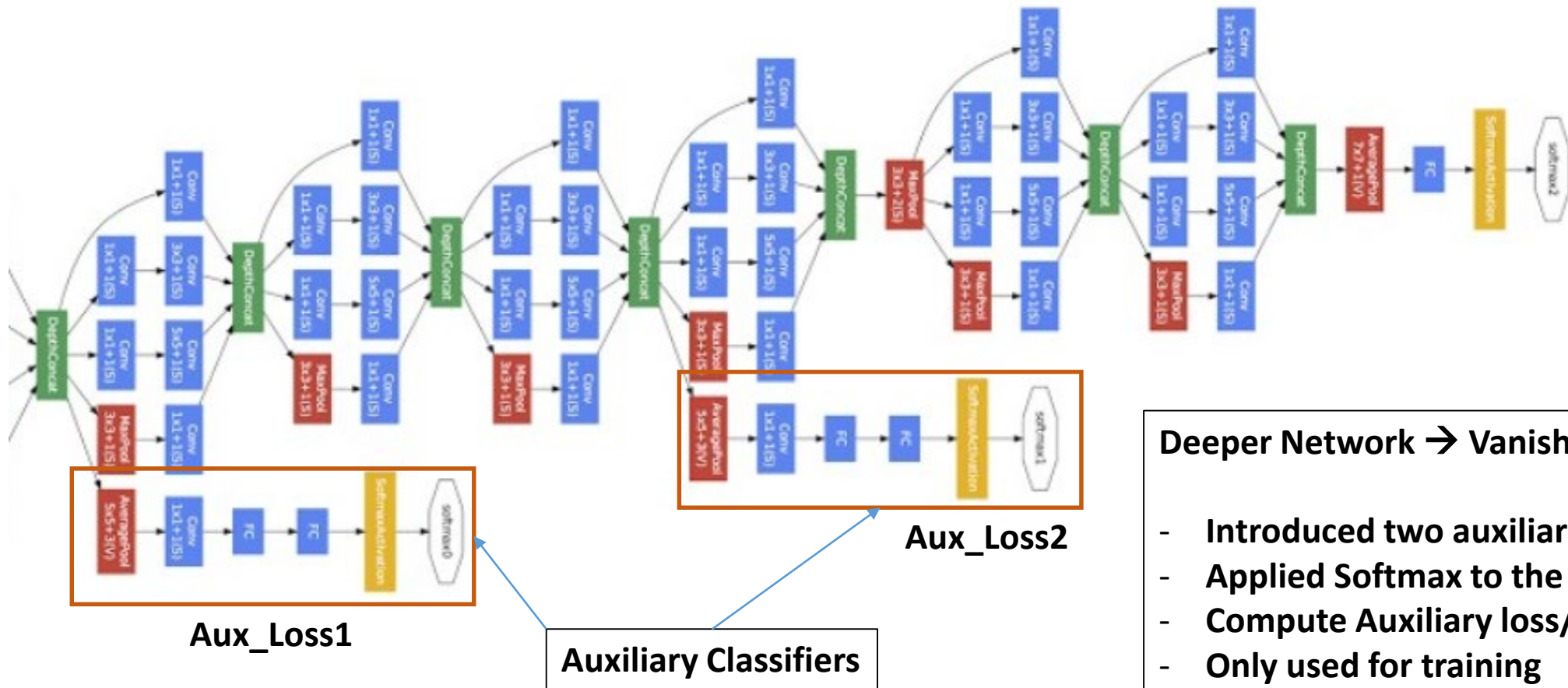
Inception Network



GoogLeNet(2014): 9 Inception modules stacked together

Convolution
Pooling
Softmax
Other

Inception Network



Deeper Network → Vanishing Gradient

- Introduced two auxiliary classifier
- Applied Softmax to the output
- Compute Auxiliary loss/cost
- Only used for training

$$\text{Total Loss/cost} = \text{Real_Loss} + 0.3 \times \text{Aux_Loss1} + 0.3 \times \text{Aux_Loss2}$$

Inception V3 Modules

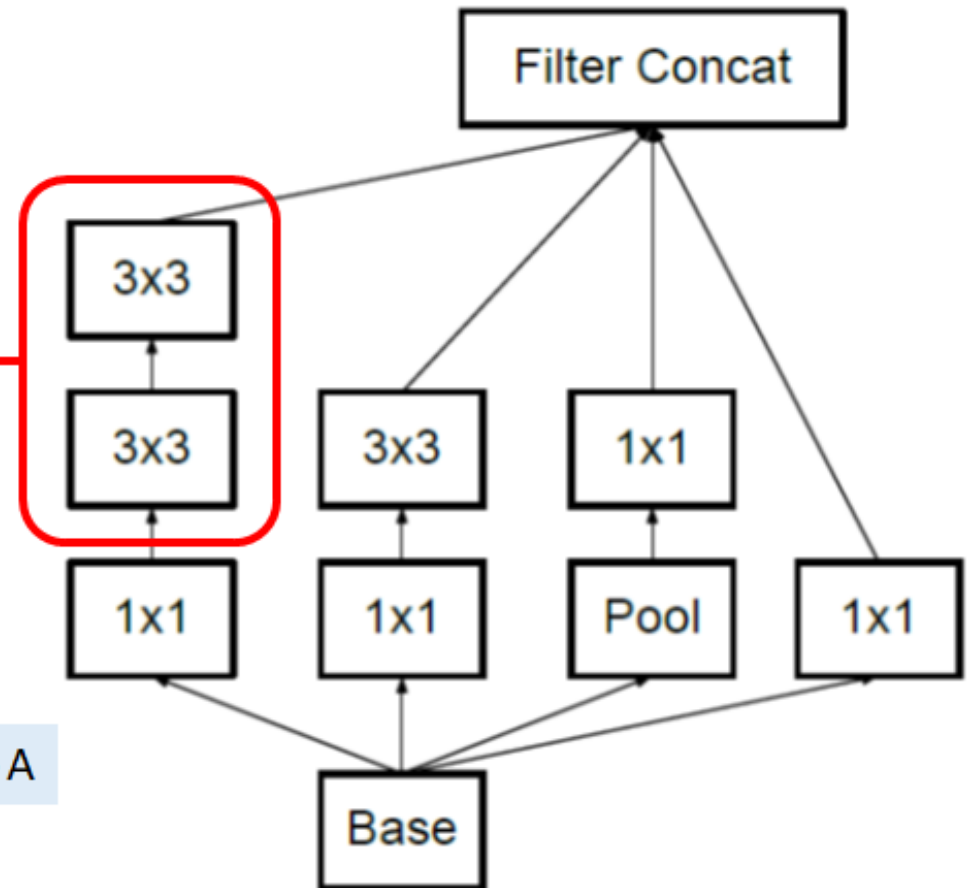
Authors suggested 3 different modules
-Factorizing Convolutions:
Reducing the number of parameters

1 layer of 5×5 filter, #parameters = $5 \times 5 = 25$
2 layers of 3×3 filters, #parameters = $3 \times 3 + 3 \times 3 = 18$

Number of parameters is reduced by 28%

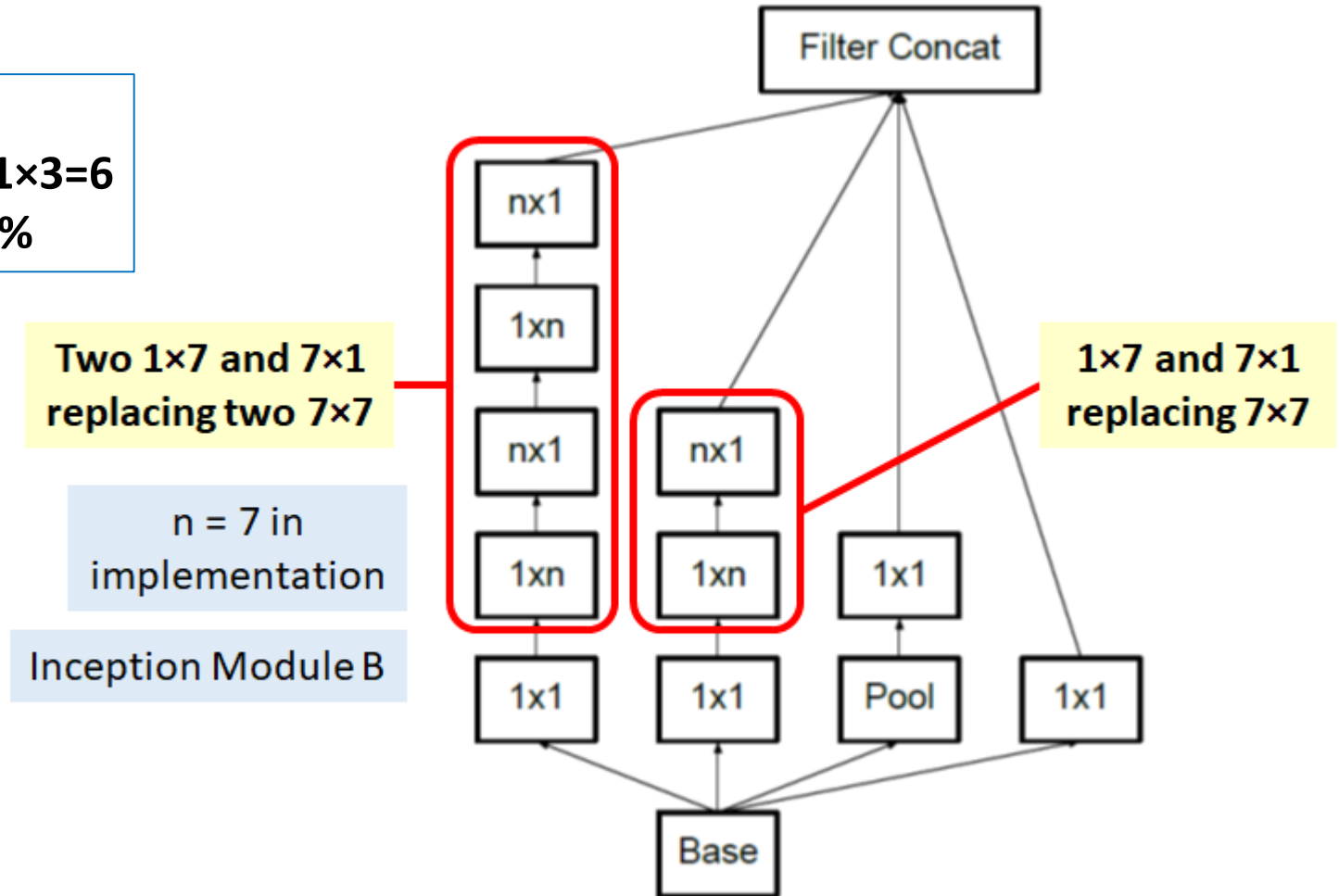
5×5 in
GoogLeNet
(Inception-v1)

Inception Module A

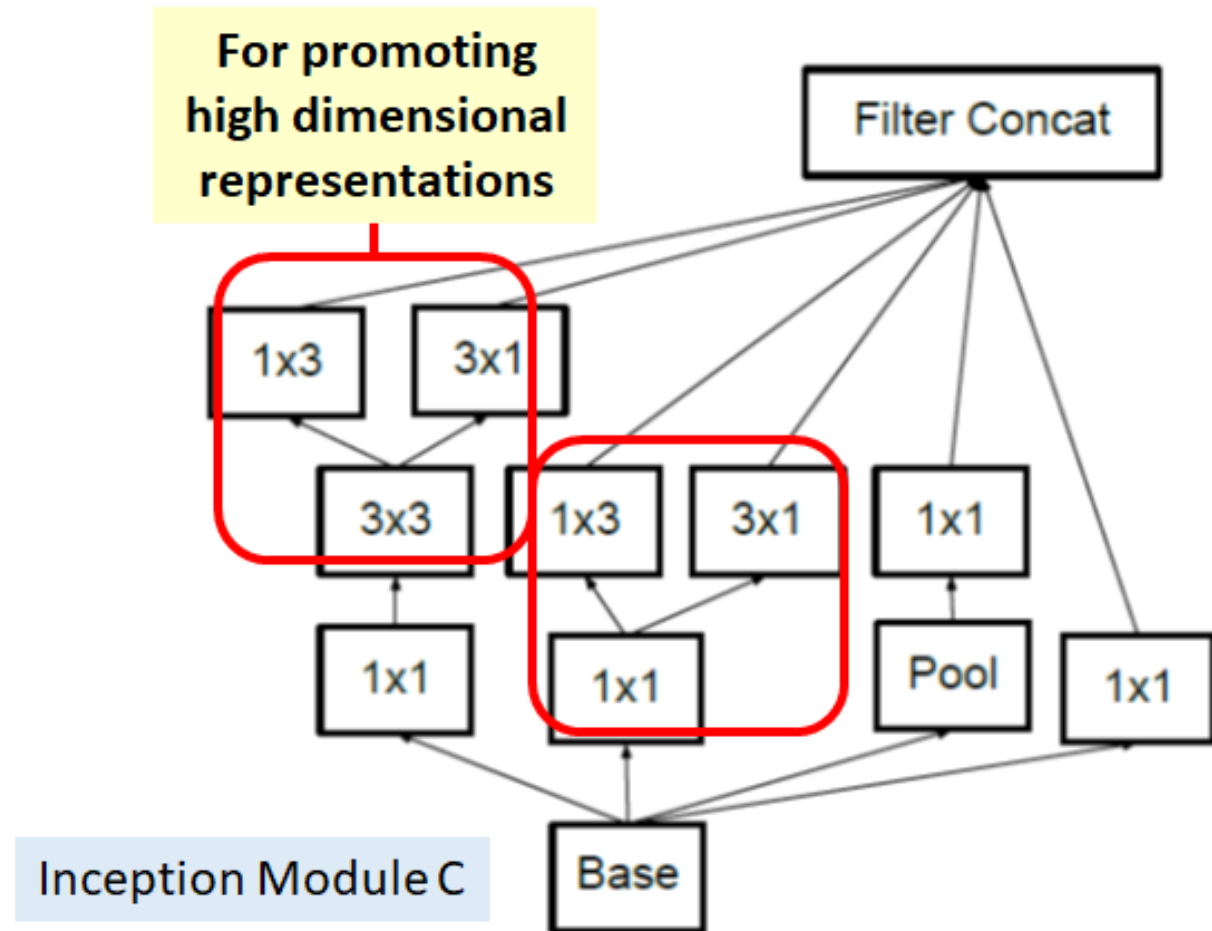


Inception V3 Modules

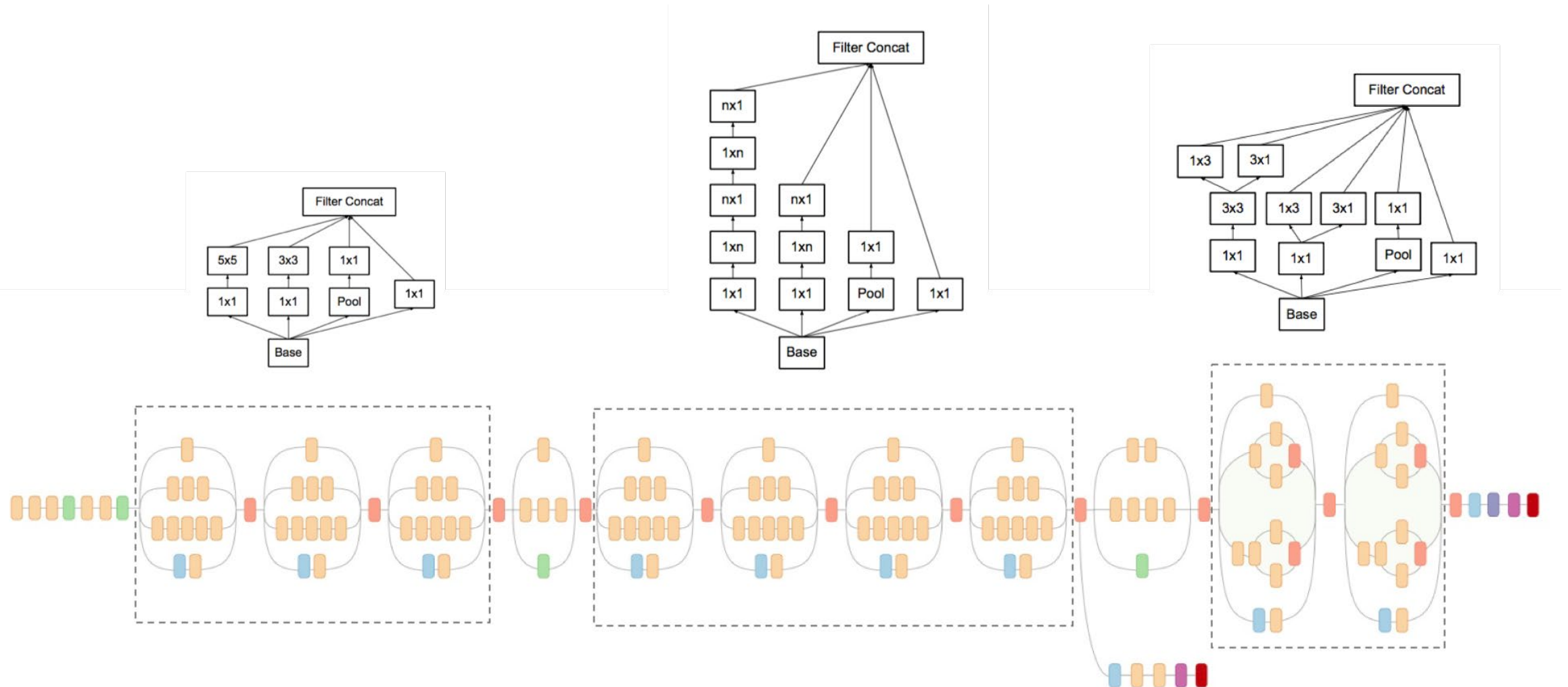
3×3 filter, #parameters = 3×3=9
3×1 and 1×3 filters, #parameters = 3×1+1×3=6
Number of parameters is reduced by 33%



Inception V3 Modules



Inception V3 Architecture



ResNet

- Deep Residual networks (ResNet) → Skip connections
- Enabled the development of the much deeper networks (100s of layers!)
- ResNet is composed of Residual Blocks were introduced!
- Degradation problem: Adding more layers eventually have negative effect on the final performance.

ResNet

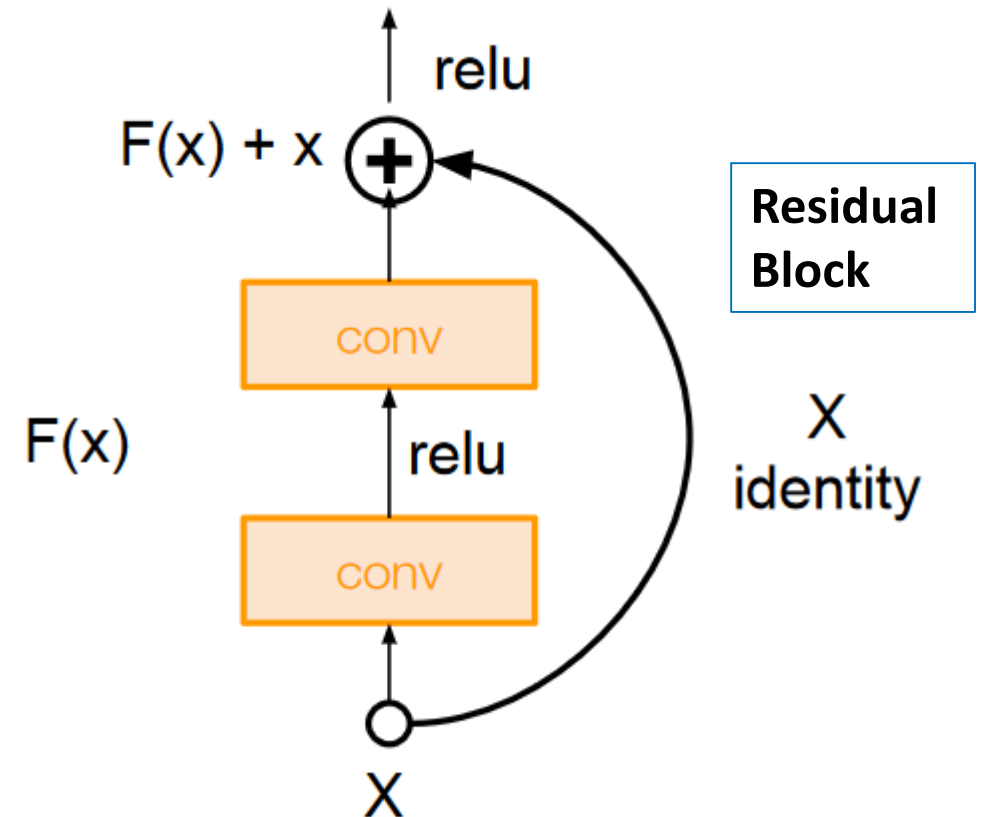
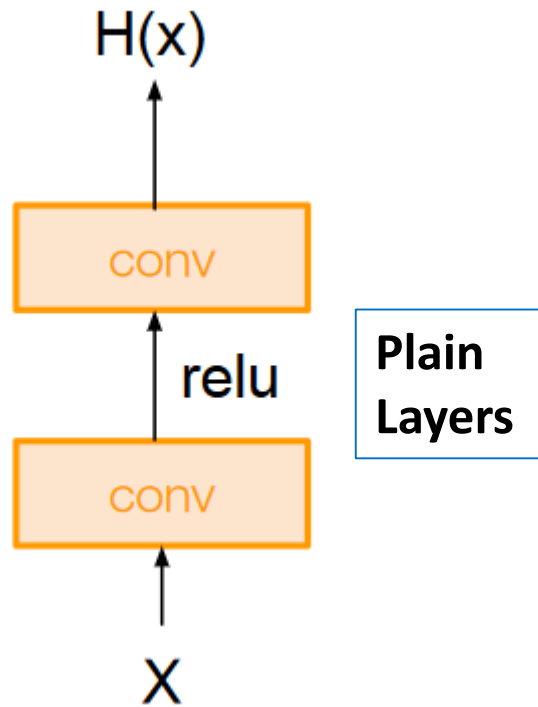
What wrong with this curves? Overfitting?



- 56 layer model is not better than the 20 layers!
- What happens when we keep add more layers to a plain CNN to make it deeper?

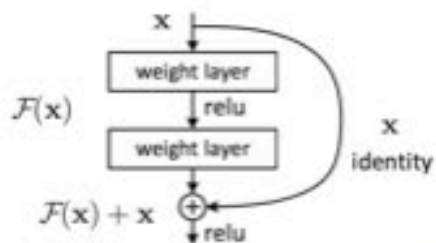
ResNet

In principle deeper model should perform better than shallow CNNs

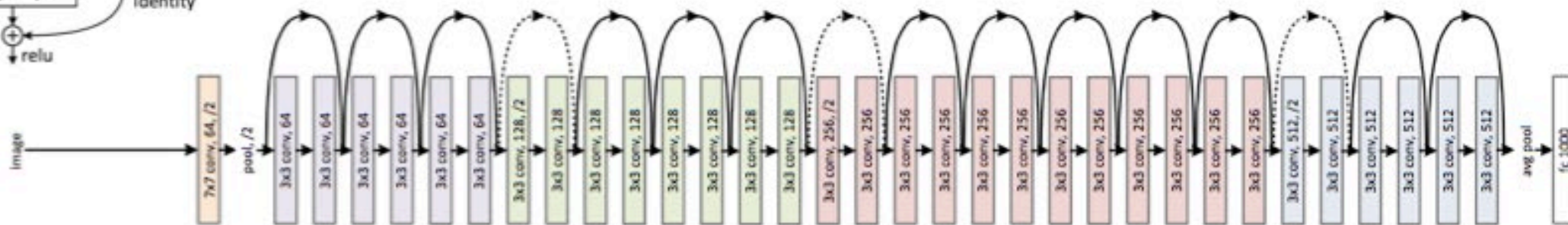


ResNet

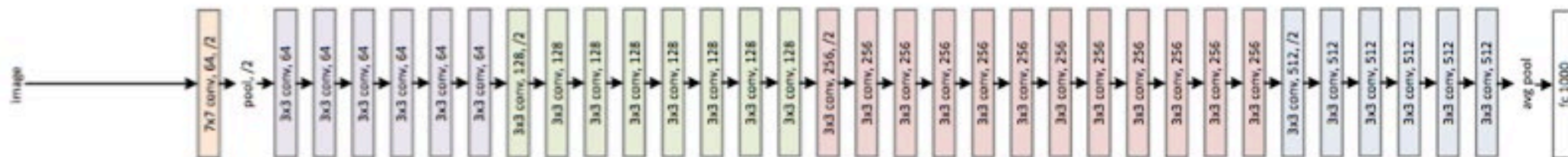
residual connection



Residual Network

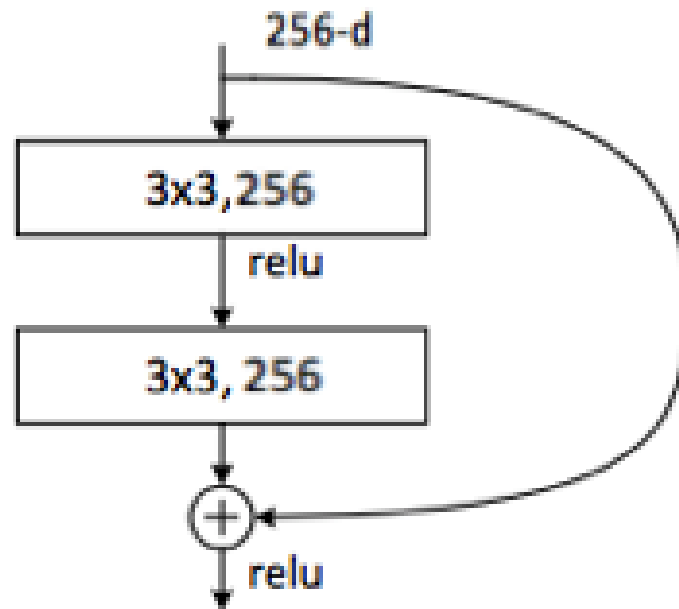


Plain Network

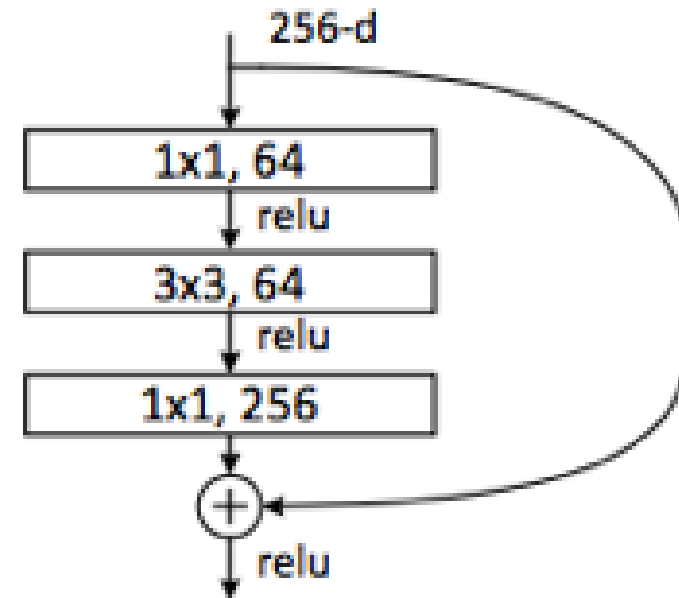


ResNet

ResNet 34
residual block



ResNet 50
residual block



Summary

