Departamento de Eletrónica, Telecomunicações e Informática

Complements of Machine Learning

Lecture 4 : Deep NN - Residual Networks

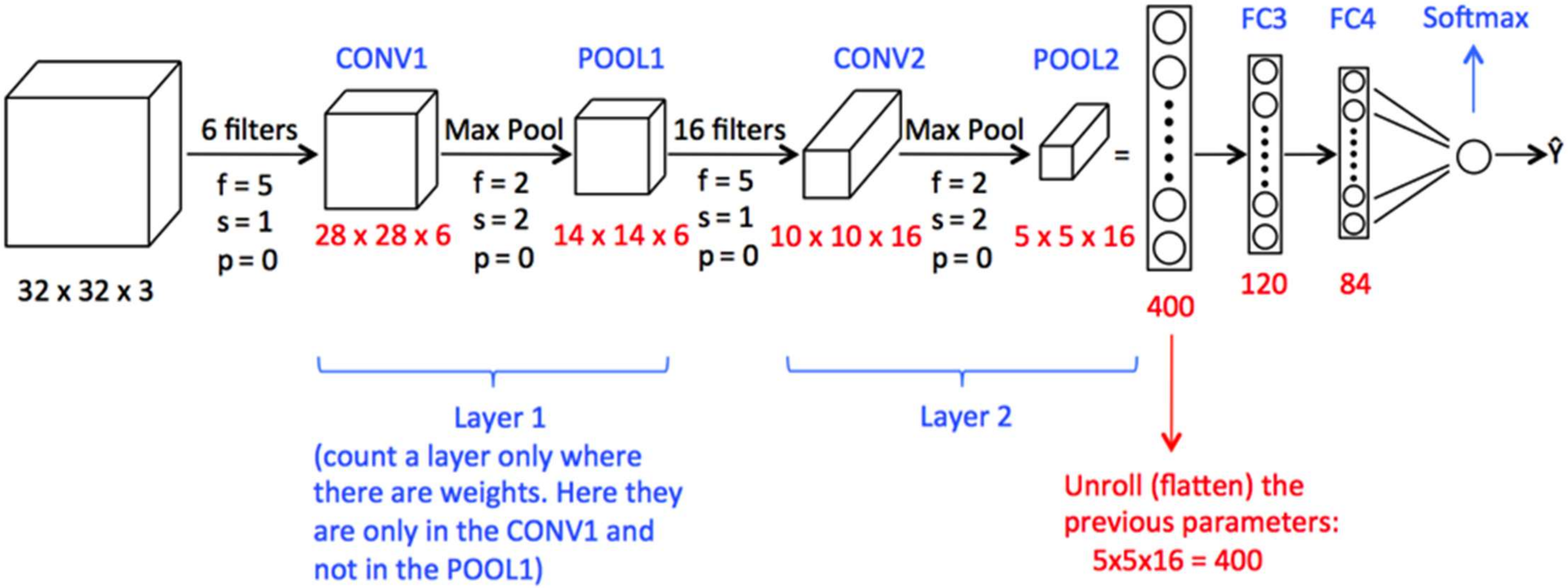
Petia Georgieva
(petia@ua.pt)

Outline

Deep NN case studies:

1. **Residual Networks (ResNet)**
2. **1x1 convolution**
3. **Inception Network**

CNN: LeNet5*

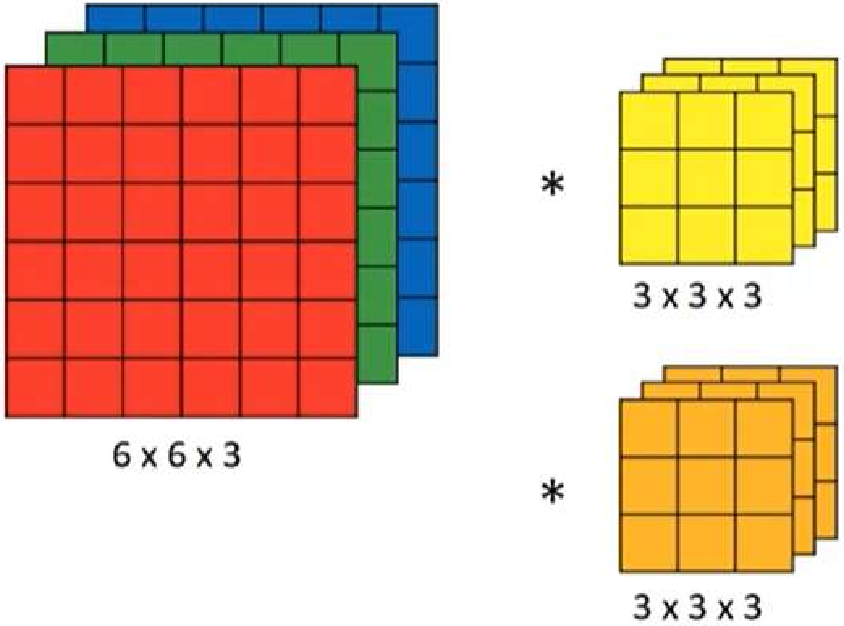


***LeCun et al., 1998, "Gradient-based learning applied to document recognition".** Original LeNet5 applied to handwritten digit recognition (grey scale images). Avg pooling, no padding, softmax classifier; ReLU and sigmoid/tanh neurons in the Fully Connected (FC) layers.

The activation function is present after the convolution, even if it is not drawn on the CNN diagram.

General trend: CNNs start with large image, then height and width gradually decrease as it goes deeper in the network, where as the number of channels increase.

Example of one Layer



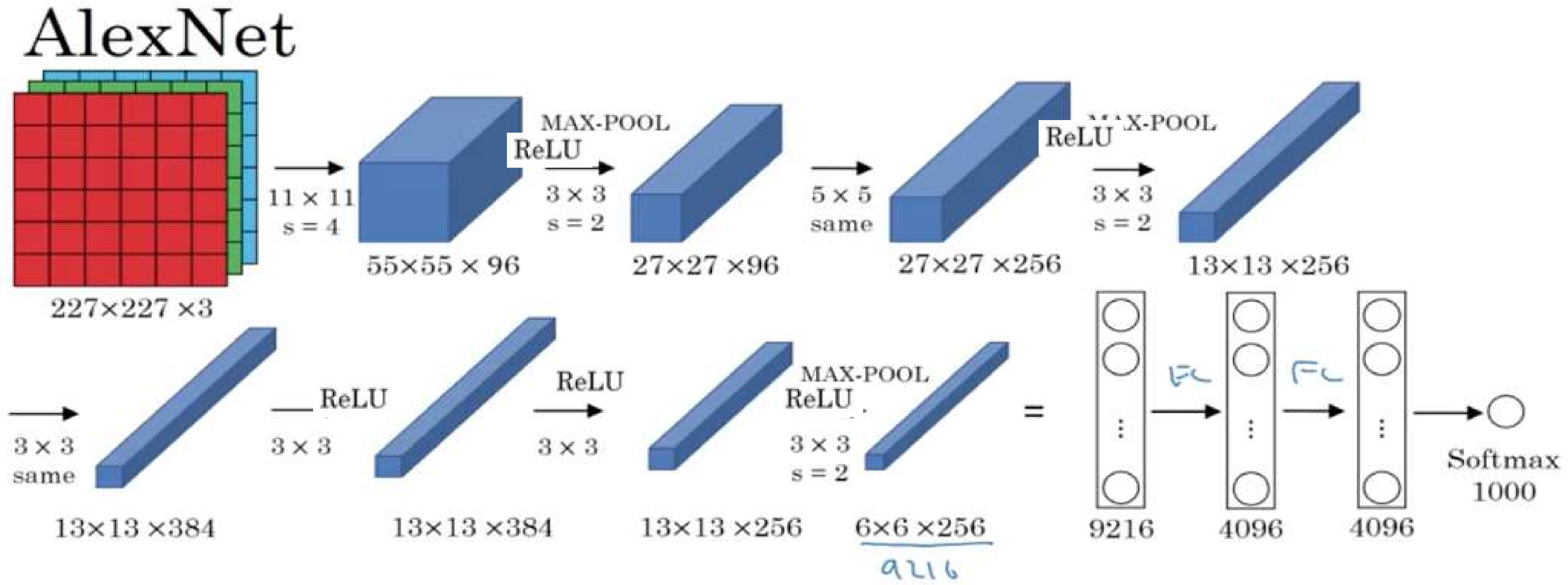
$$\lfloor \frac{n+2p-f}{s} + 1 \rfloor \text{ by } \lfloor \frac{n+2p-f}{s} + 1 \rfloor$$

padding =0; stride =1:

Dimension of feature map (conv layer output) ?

How many trainable parameters has the layer ?

AlexNet*

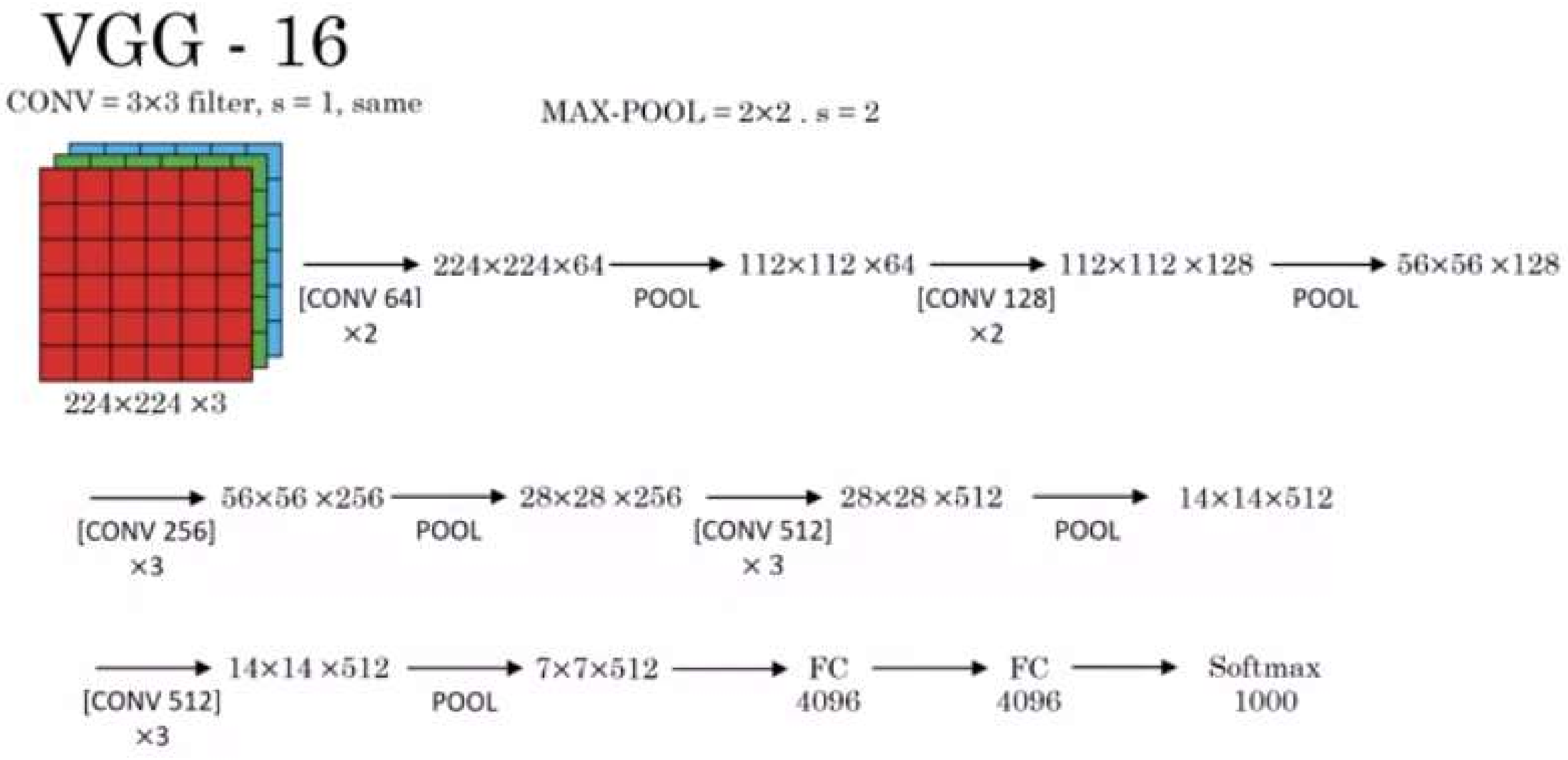


*** Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton, 2012, ImageNet classification with deep convolutional neural networks.**

5 conv layers, 3 FC layers with softmax output, 60 million parameters in total.
AlexNet applied to ImageNet LSVRC-2010 dataset to recognize 1000 different classes.

This paper convinced the Computer Vision (CV) community that DL really works and will have a huge impact not only in CV but also in speech/language processing.

CNN: VGG16



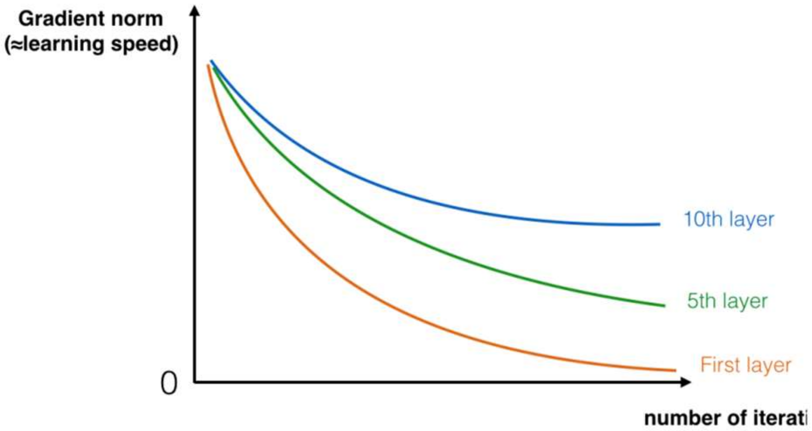
*** Karen Simonyan, Andrew Zisserman, 2015, Very Deep Convolutional Networks for Large-Scale Image Recognition**

VGG-16 has 16 layers (with weights), ***138 million parameters !!!***

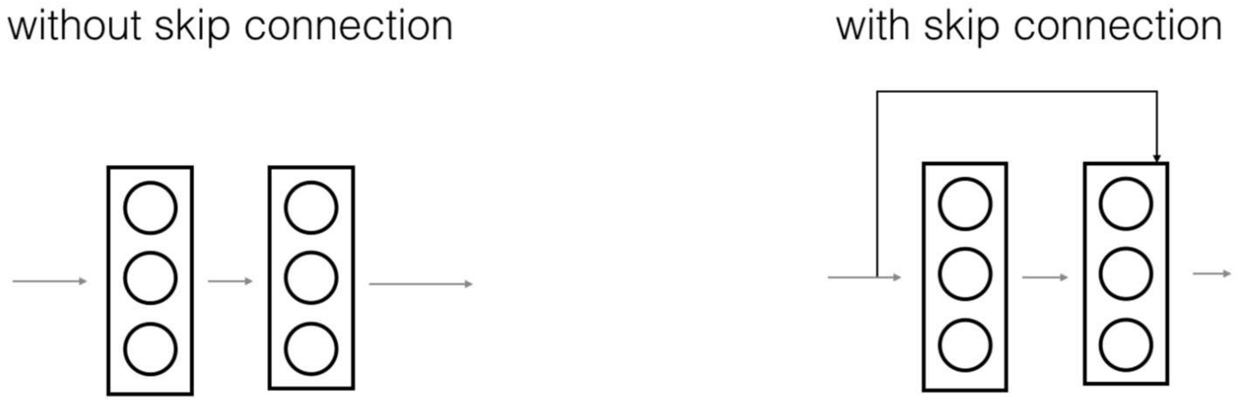
Unified architecture: All conv layers: (3x3) filters, s=1, same; All MaxPool =2x2, s=2.

At each convolution the height and width go down by a factor of 2, the channels go up by a factor of 2.

Residual Networks



Vanishing gradient



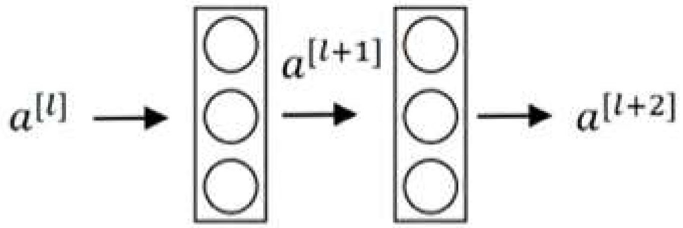
Residual block

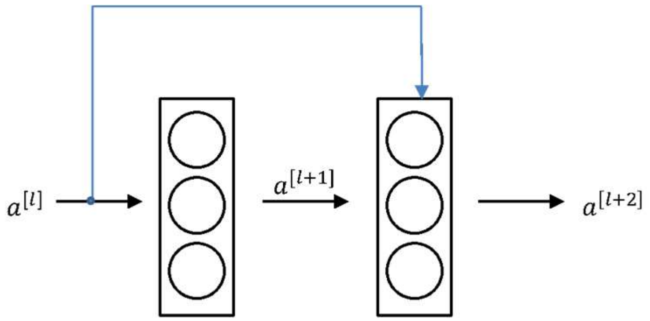
Very deep NNs are difficult to train because of vanishing or exploding gradient issues. During training, the magnitude of the gradient for the shallower (the first) layers decrease to 0 very rapidly as training proceeds, as shown on the figure.

Skip connection (short cut) takes the activation from one layer and feed it to another layer much deeper in the network.

Remark: Exploding gradient is easier to spot (NaN in computations – numerical overflow) => apply gradient clipping, i.e. limit the gradients up to some max value.

Residual Block



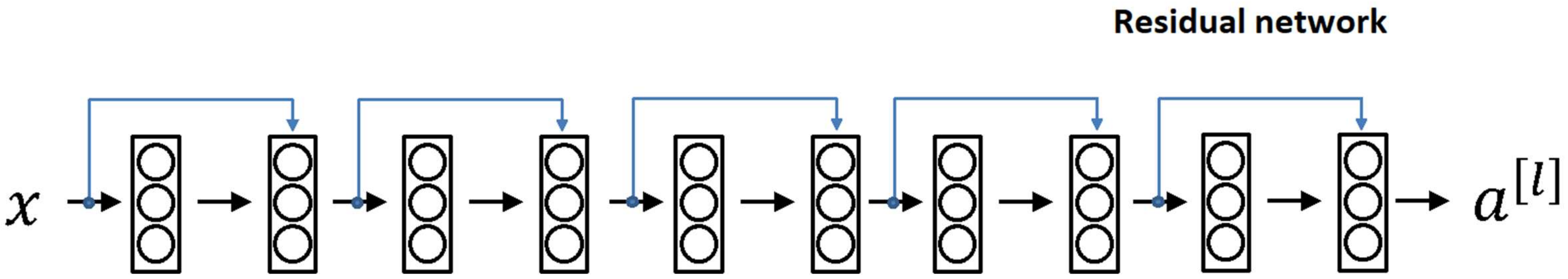


$a^{[l+2]} = g\left(W^{[l+2]}a^{[l+1]} + b^{[l+2]}\right) -$ without skip connection

$a^{[l+2]} = g\left(W^{[l+2]}a^{[l+1]} + b^{[l+2]} + a^{[l]}\right) -$ with identity skip connection

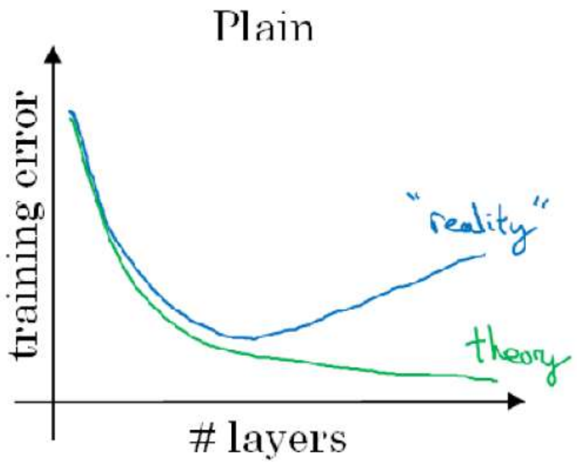
Due to regularization (L2, drop out) trainable parameters (*W*, *b*) in layer [*l*+2] may tend to small values (close to 0) => next layers will stop learning.
With skip connection it will transmit learned patterns of layer *l*.

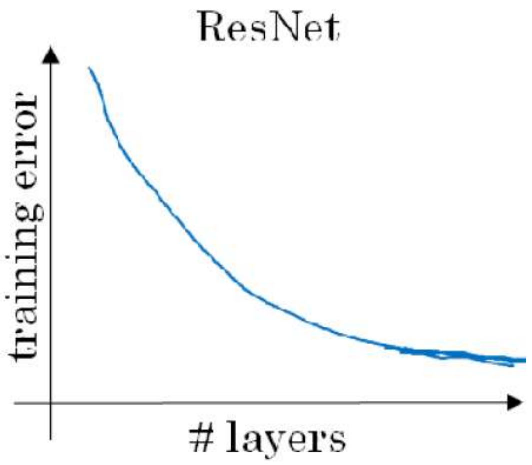
Residual Network vs Plain DNN



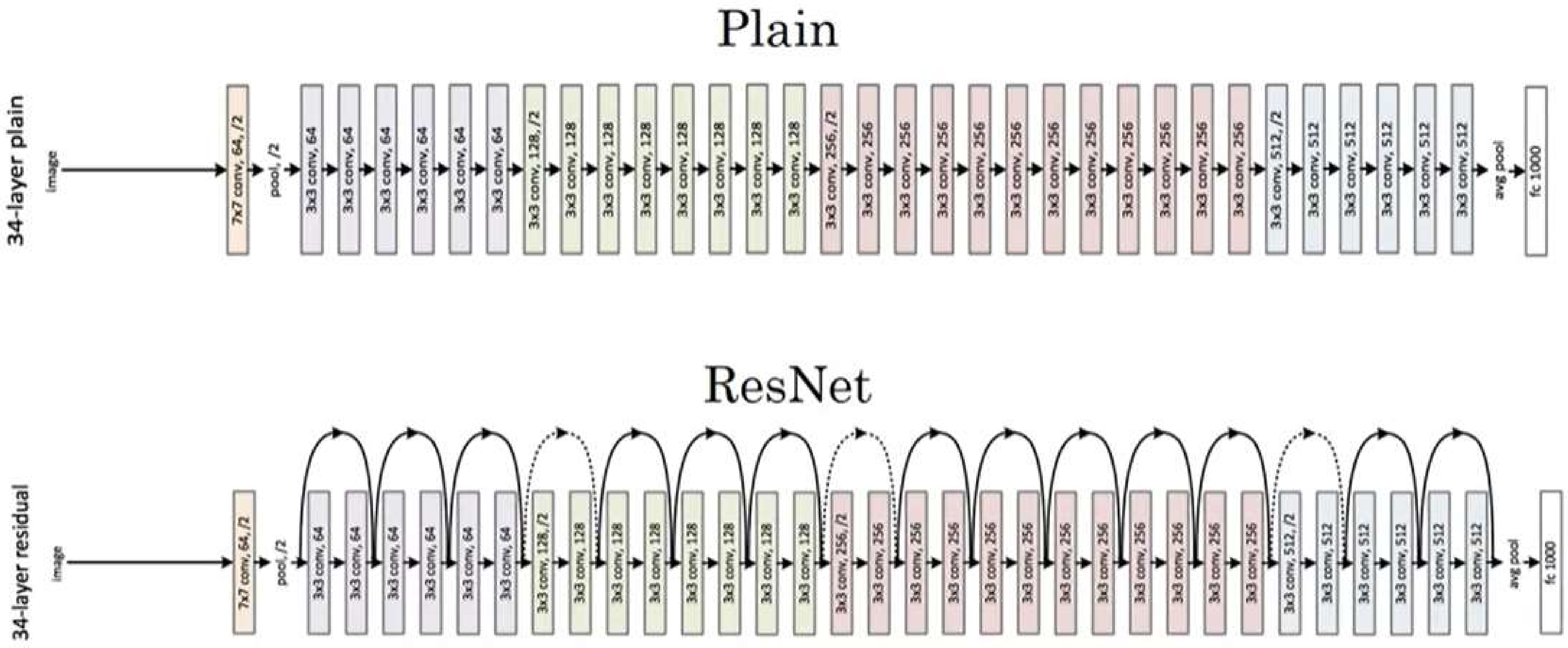
He, at all 2015, Deep residual networks for image recognition

Skip connection builds networks that enable to train very deep structures (over 100 layers).



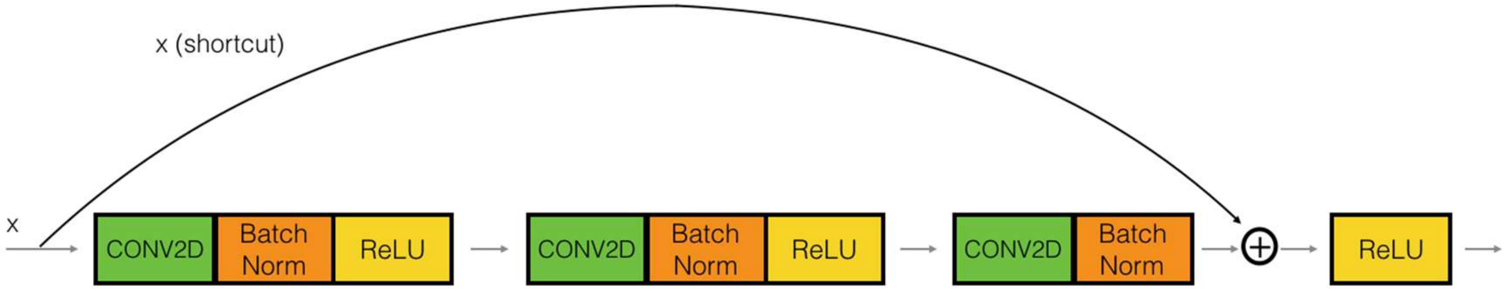


ResNet on Images

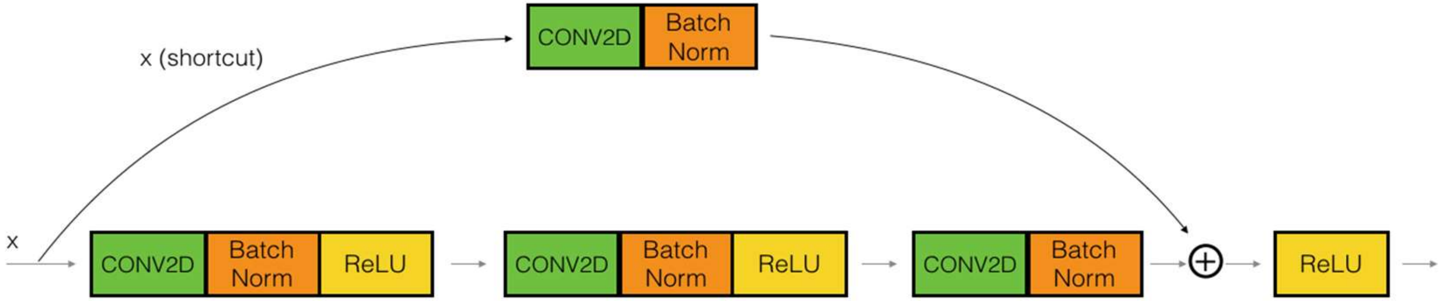


Lab: Residual blocks

Identity skip connection /shortcut

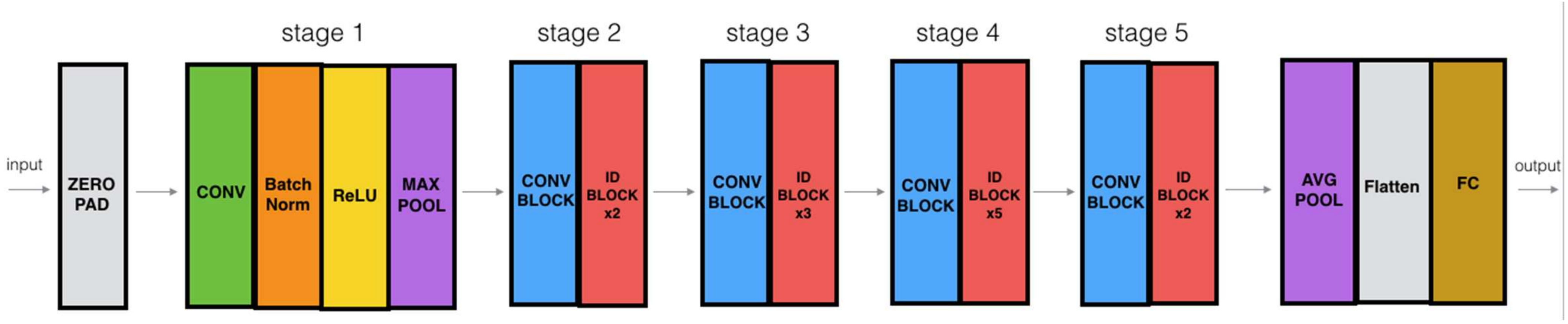


Convolution skip connection /shortcut



ResNets are built with such residual blocks.

Lab: ResNet 50 model



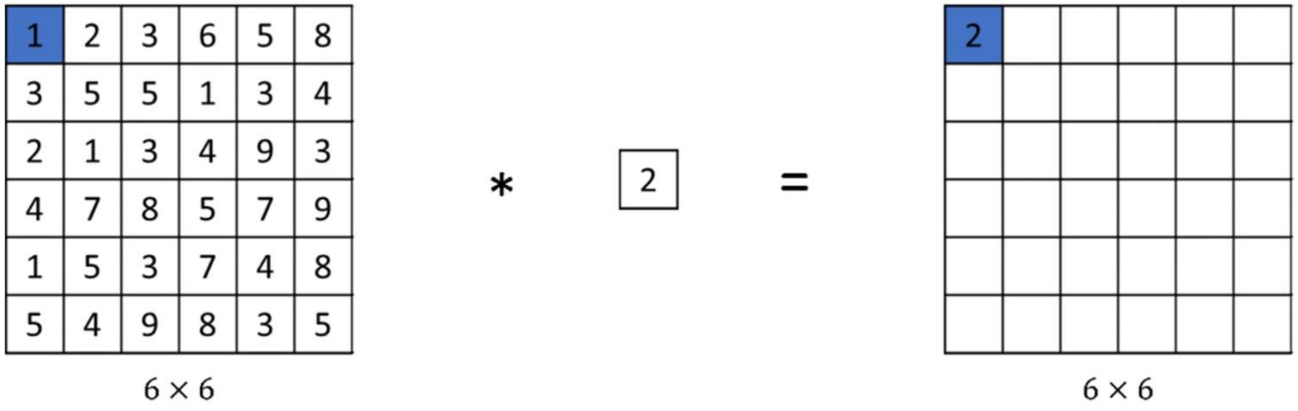
ResNet-50 has 5 stages with Convolution and Identity blocks.

Each Conv BLOCK has 3 convolution layers and each identity block (ID BLOCK) also has 3 convolution layers.

Stage1(1)+Stage2(3+2*3)+ Stage3(3+3*3)+ Stage4 (3+5*3)+ Stage5 (3+2*3)+1FC =50

ResNet-50 has over 23 million trainable parameters !!!

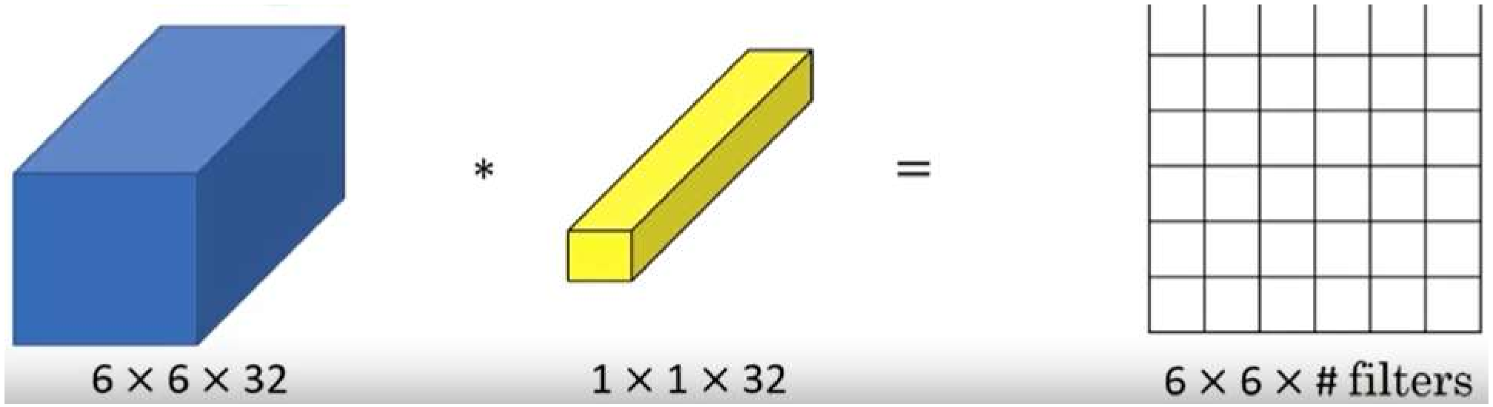
1 by 1 Convolution



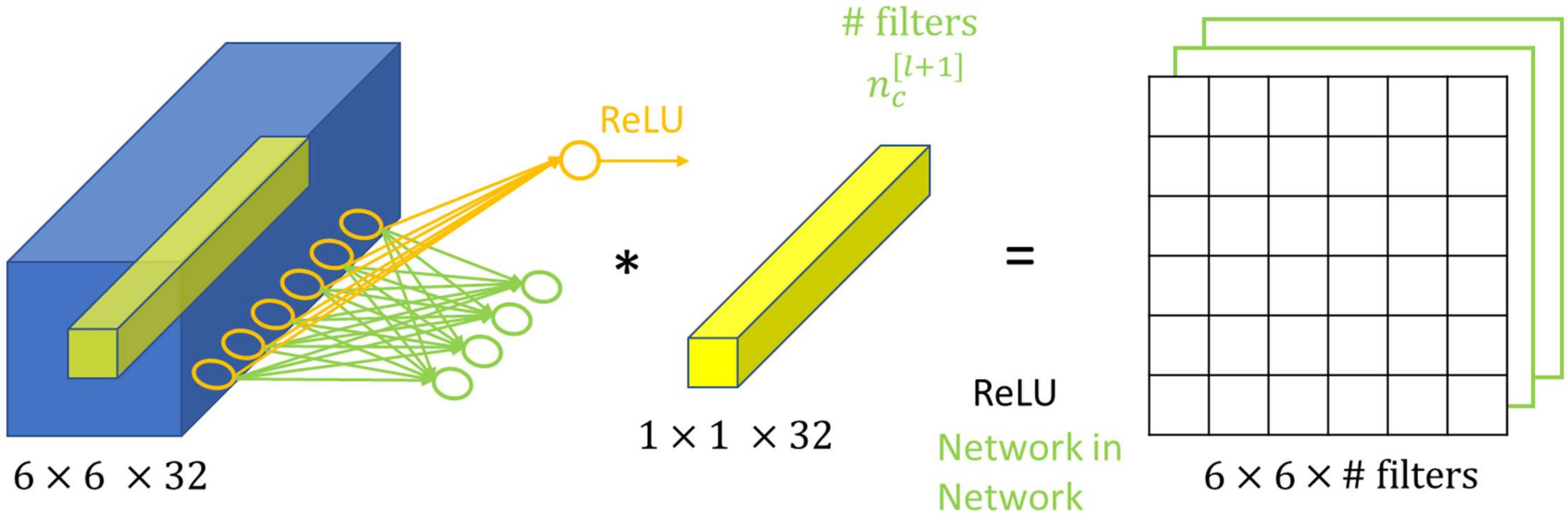
1	2	3	6	5	8
3	5	5	1	3	4
2	1	3	4	9	3
4	7	8	5	7	9
1	5	3	7	4	8
5	4	9	8	3	5

If we have 2D image and convolve with 1x1 filter it does not seem very usefull, just a miltiplication by a number.

But if we have a 3D volume (e.g. 6 x 6 x 32 channels), convolution with 1x1x32 is useful.



1 by 1 Convolution (Network in Network)



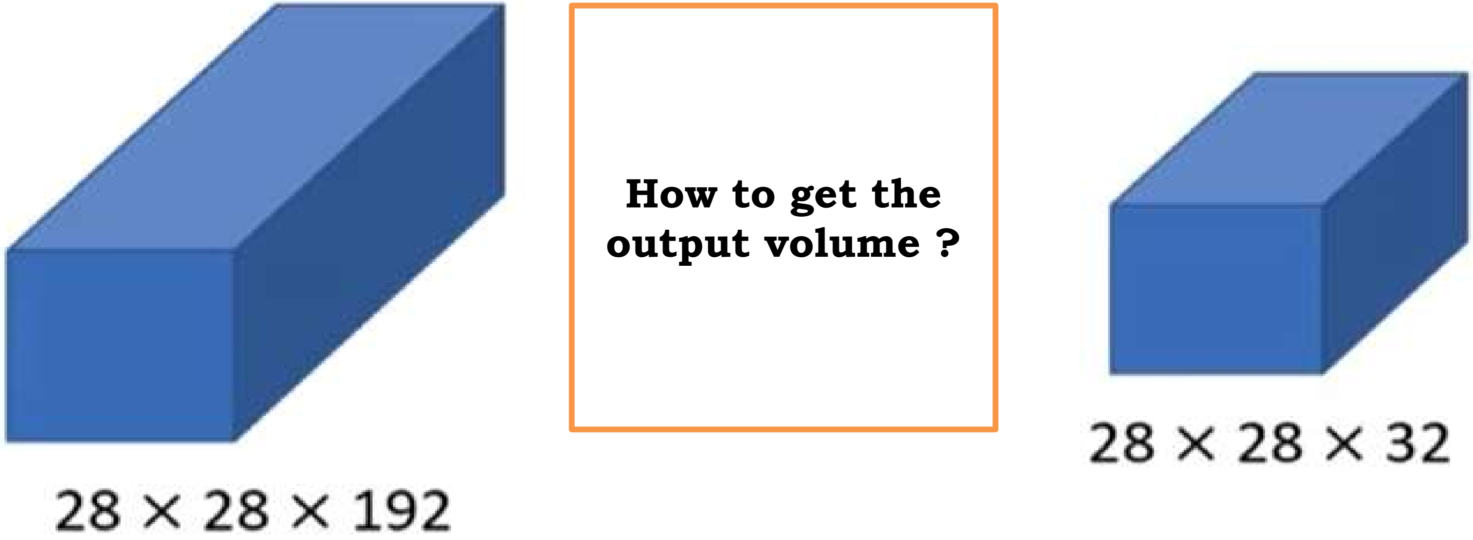
1 by 1 convolution is useful for:

a) Learn complex non-linear relationships over a slice of the input volume.

b) Way to control (reduce, increase) number of output channels.

Paper: Lin at al., 2013, Network in Network

1x1 Convolution –
example of output channel reduction



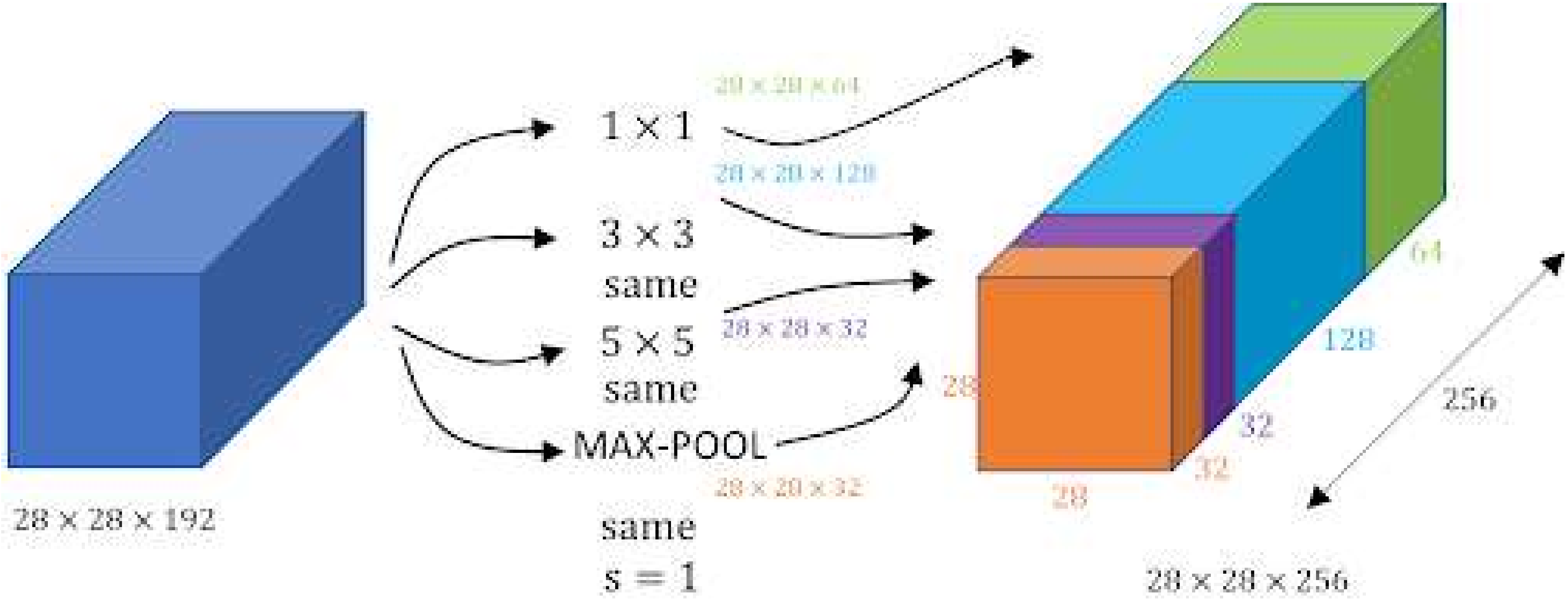
How to get the output volume ?

Inception Module

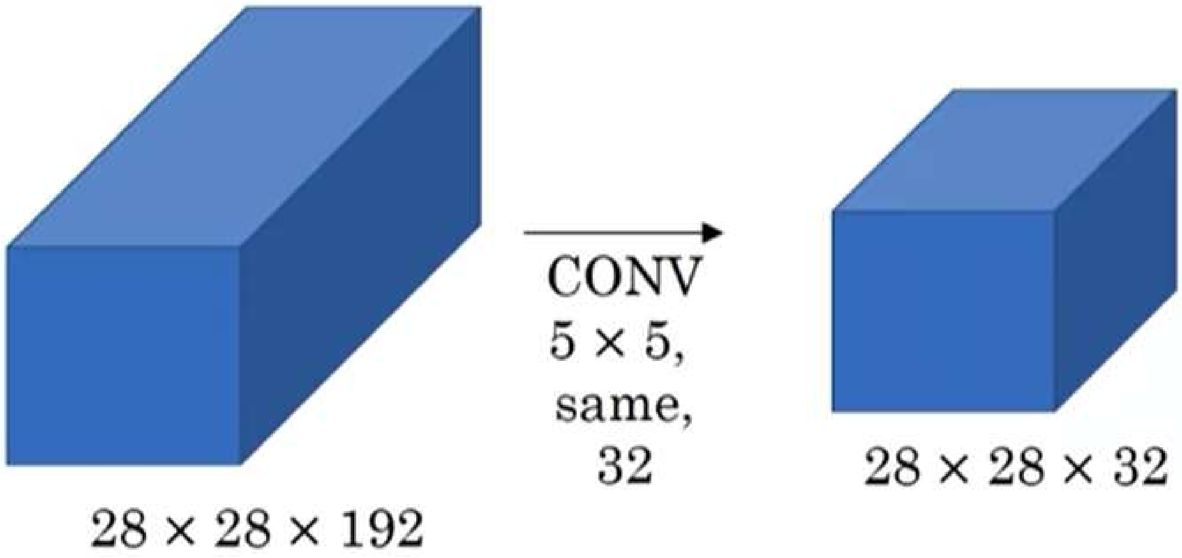
When design a Conv Layer we may want to choose different filters (e.g. 1x1; 3x3; 5x5, or pooling payer).
In Inception network we may apply all of them and concatenate the outputs.

But such big/complex network will increase the computational cost.

Paper: Szegedy at al., 2014, Going deeper with the convolutions



Computational Cost - example

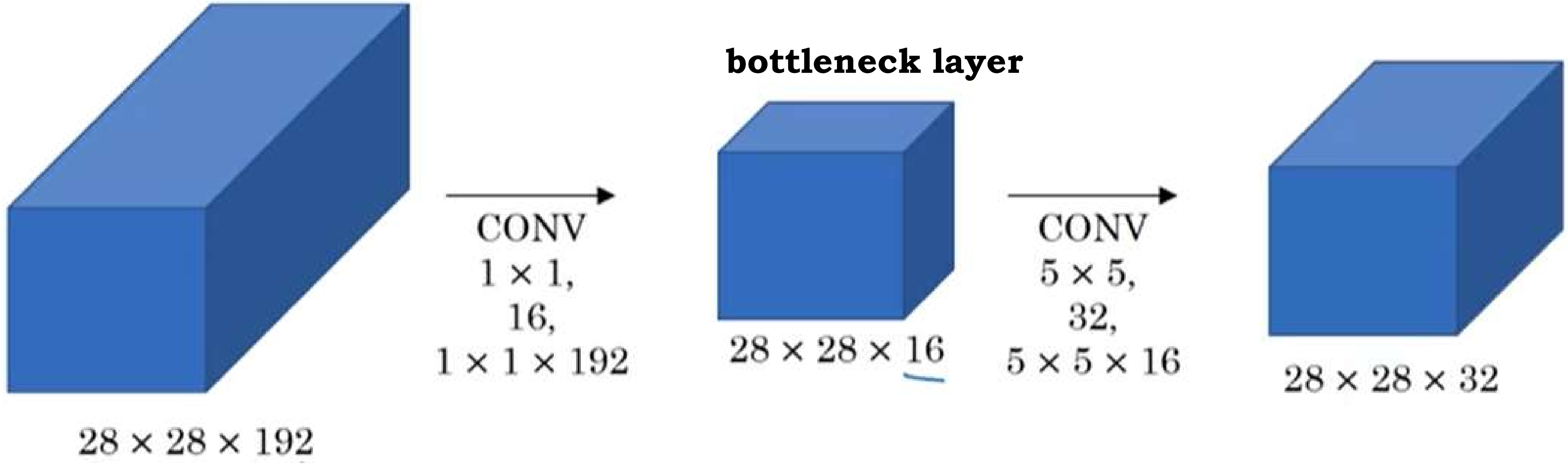


32 filters, each of them of size ?

padding =“same” (no height and width shrink)

Computational Cost (how many computations) ?

Comp. cost using 1x1 convolution -example



Input and output dimensions are still the same.

Shrink the big input volume into a much smaller intermediate volume (28x28x16) , known as **bottleneck layer** (the smallest part of the network).

Computational Cost (using 1x1 convolutions):

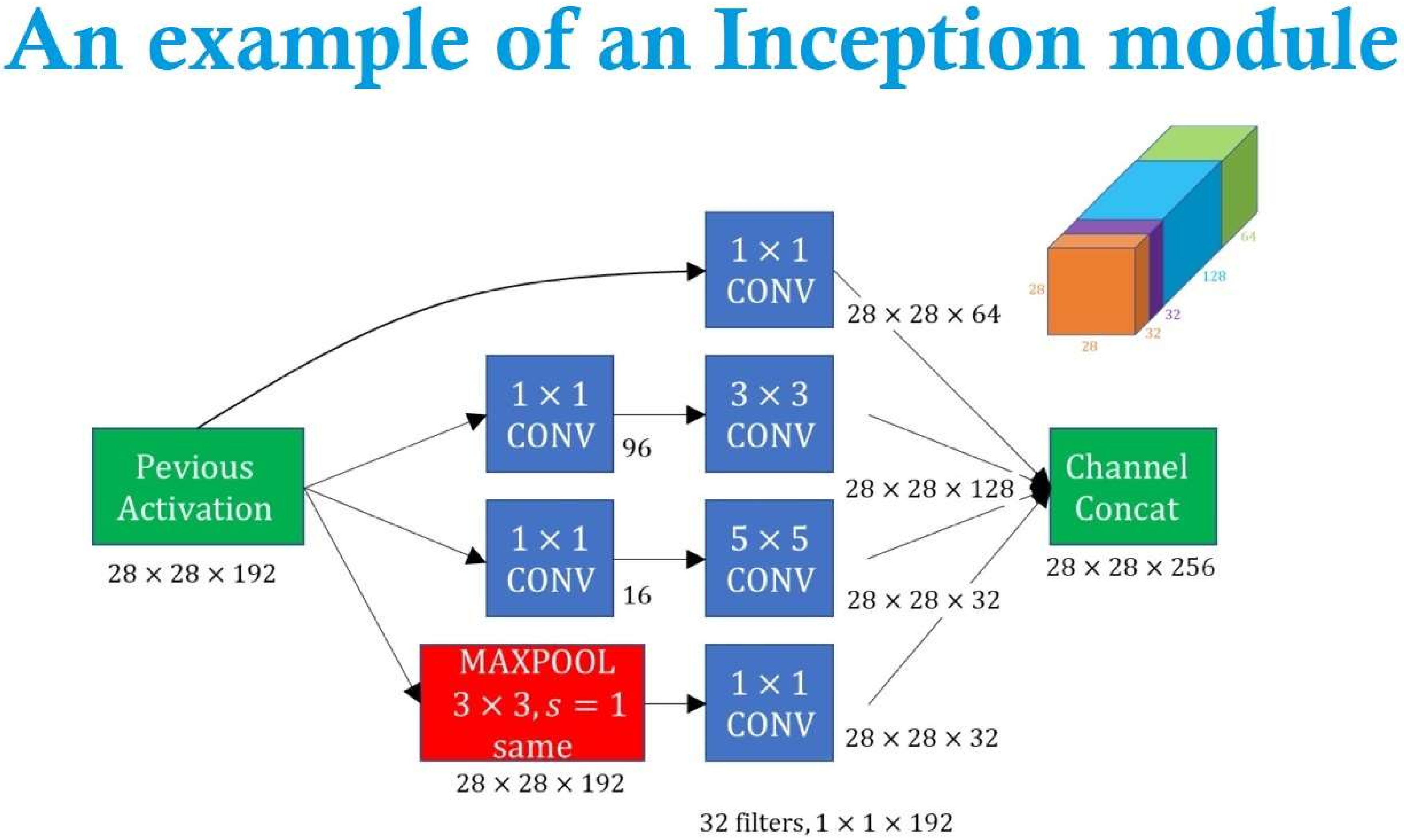
Stage1 (Input-Bottleneck connection): ? computations

+

Stage2 (Bottleneck –Output connection): ? computations

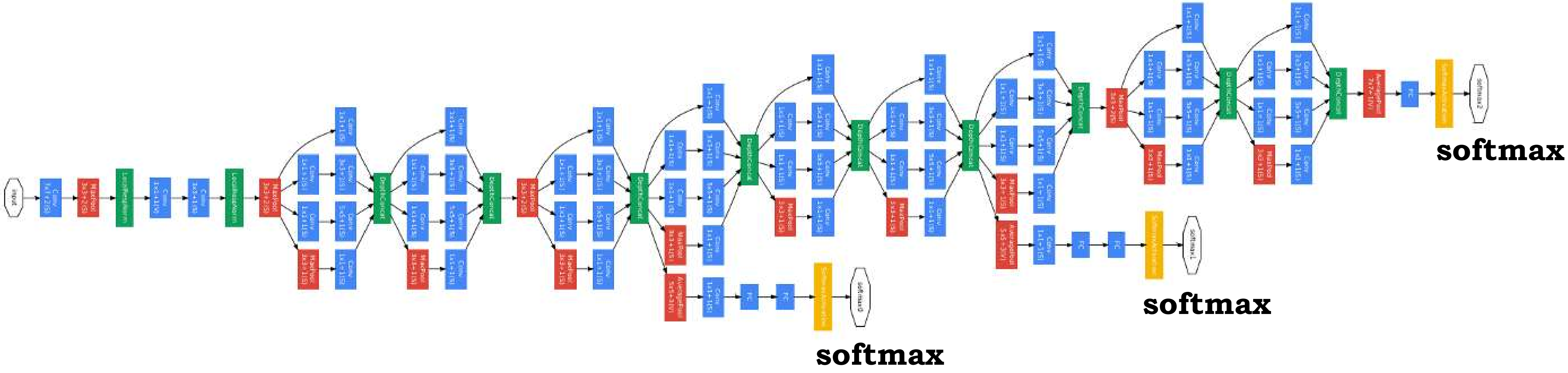
INCEPTION MODULE

Inception Network puts all these modules together, using 1x1 convolutions to save computations.



An example of an Inception module

INCEPTION NETWORK (GoogLeNet)



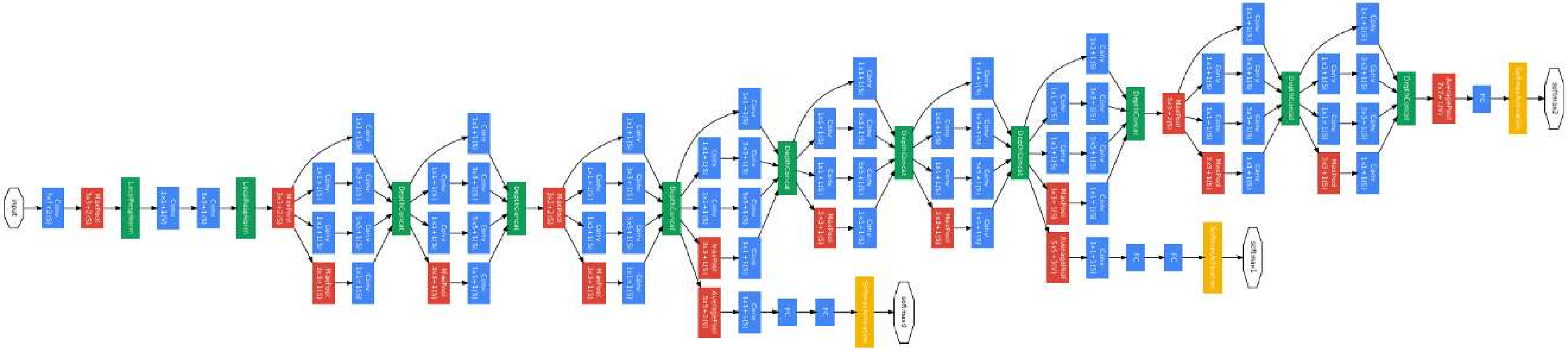
Ref. Szegedy et al 2014, “Going deeper with Convolutions”. Around 25 mln. of parameters.

Many inception blocks repeated at different positions of the network.

The last layers are fully connected followed by softmax layer trying to predict what is the output label.

Side branches take inf from previous (early) layers and try to make predictions.
The earlier features are not that bad to make predictions.
This has a regularization effect, combat the over fitting.

INCEPTION NETWORK – several architectures



INCEPTION NETWORK = GoogLeNet =Inception_v1
Other versions are available, Inception v2, v3, v4, etc., they all are based on the idea of inception module.