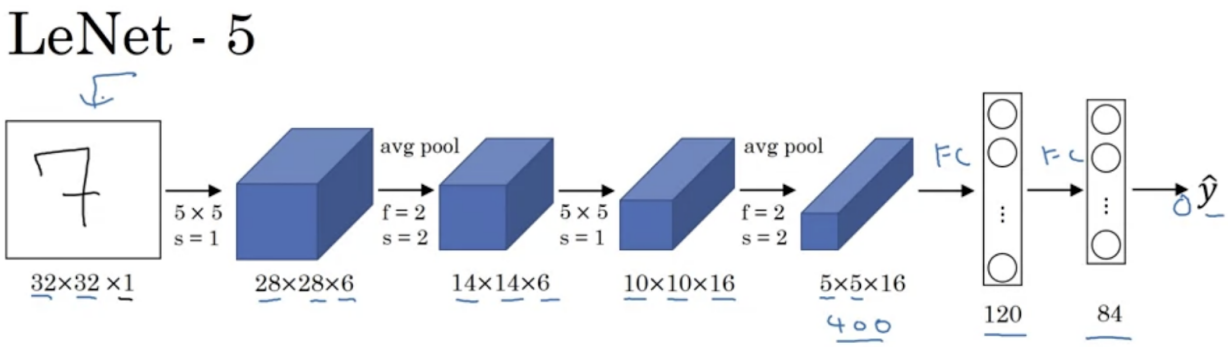
Convolutional Neural Networks DeepLearning AI

Case Studies

Alec Dewulf

Classic Neural Networks

The goal of LeNet - 5 was to recognize handwritten digits. It takes a grayscale image as input.

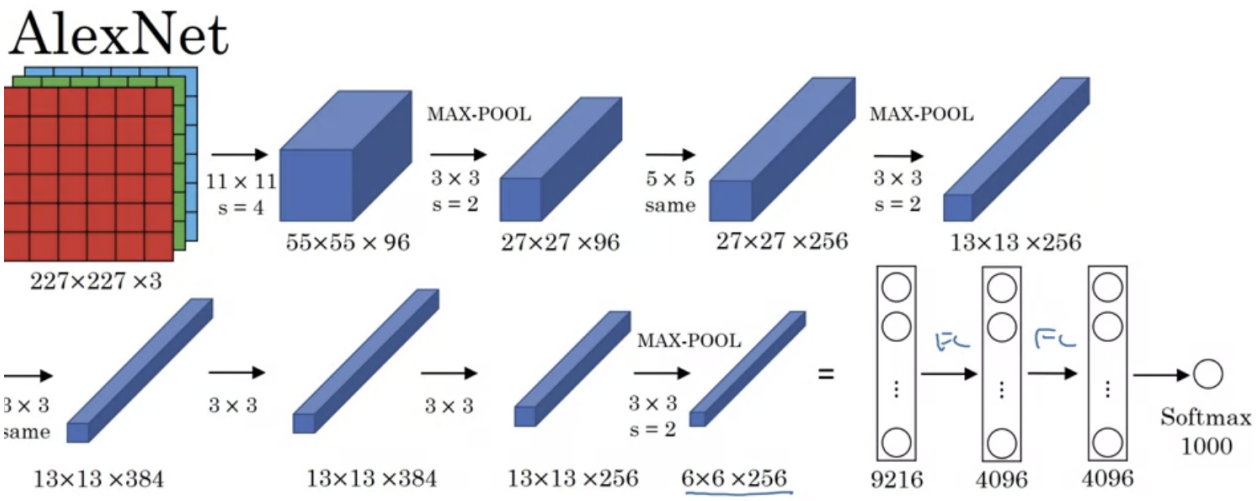


A modern version of this net would use a softmax activation function. There are about 60k parameters in this model.

This model is described in LeCun et al. 1998. Gradient-based learning applied to document recognition.

The original model had a nonlinearity after pooling which is not common today.

AlexNet takes 227x227x3 RGB images.

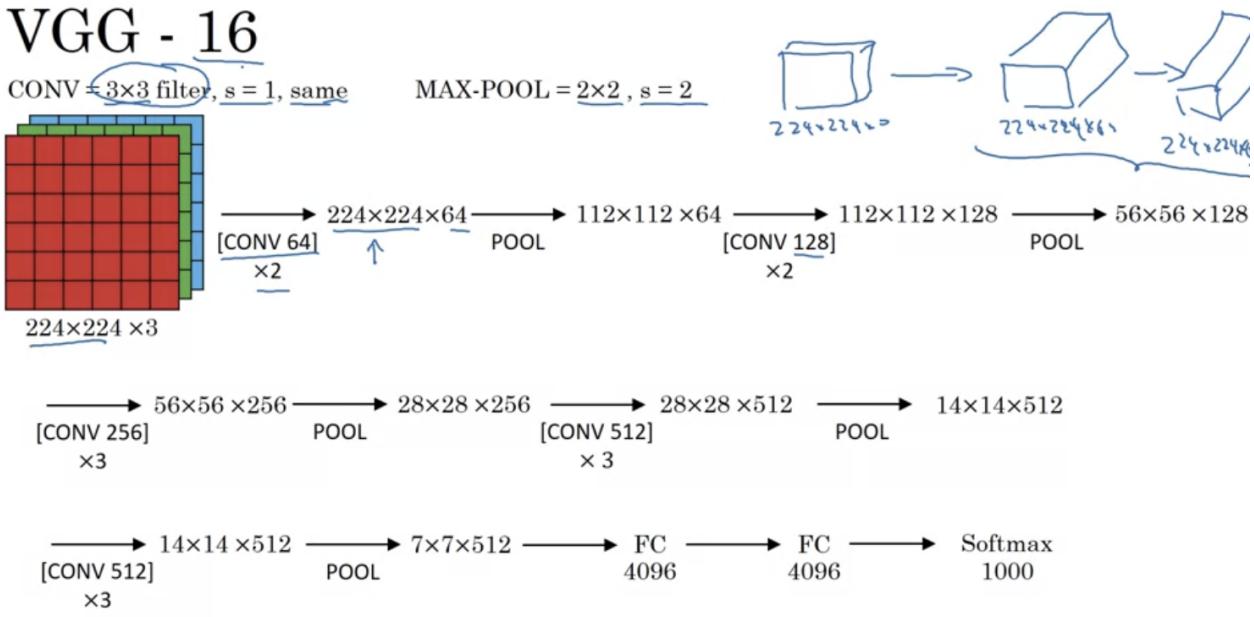


This was obviously a much bigger net than LeNet. It had about 60 million parameters and was trained on the ImageNet dataset. It also used ReLU which made it much better than LeNet.

Krizhevsky et al., 2012. ImageNet classification with deep convolutional neural networks.

Local response normalization layers were used in AlexNet, but these are rare. It normalizes vectors along channels. The motivation is maybe you don't want too many neurons with a high activation. This technique is probably not useful.

The VGG - 16 network was a much simpler network focused on conv layers with 3x3 filters, stride of 1, and same padding.

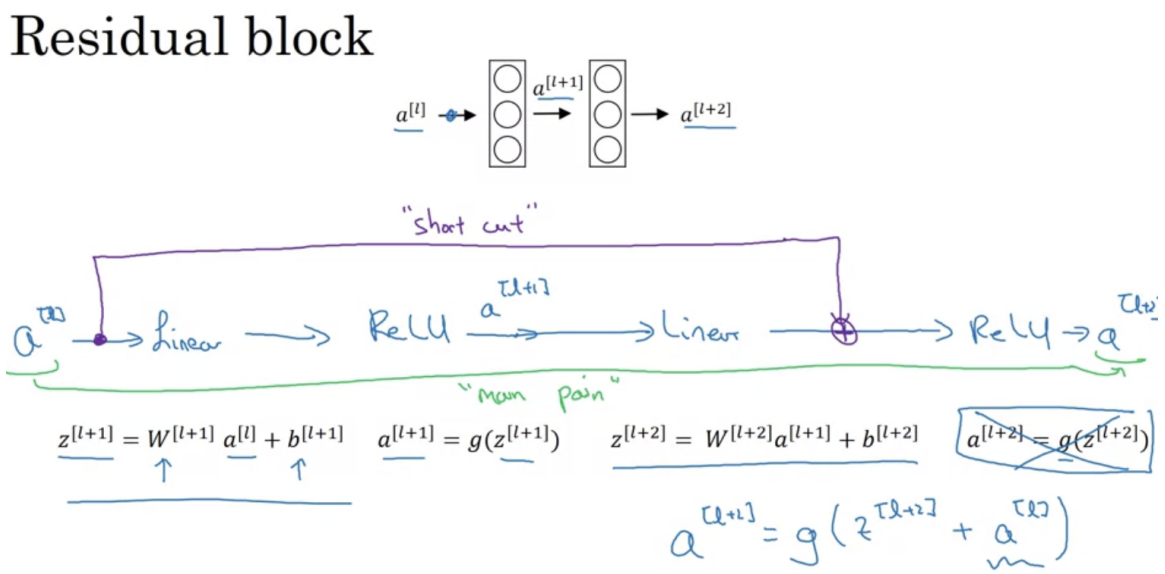


The 16 in the name refers to the 16 layers that have weights. There are about 138 million parameters in the network.

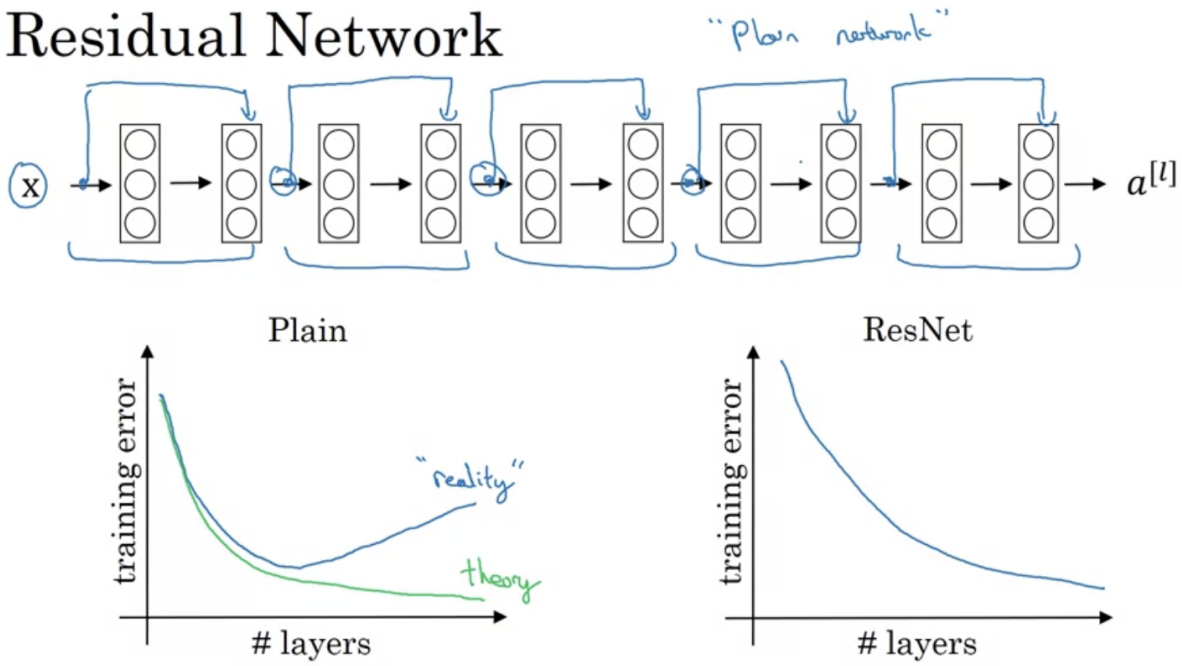
The number of filters in each Convlayer was increasing by a factor of two for the first half of the model. This is a basic principle underlying its structure.

ResNets

ResNets are built out of residual blocks. They allow information to skip layers in the network by taking a shortcut deeper into the network.



The inventors of ResNet found that using these blocks allows you to train much deeper networks.



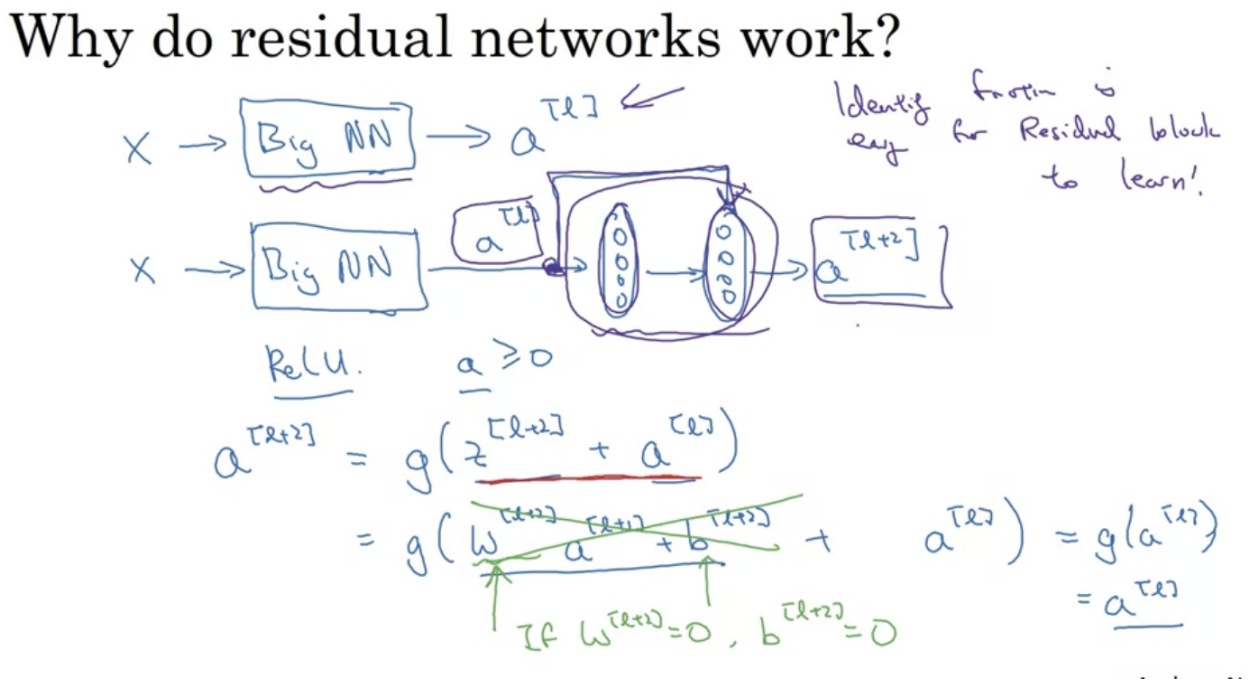
ResNets helps with the vanishing and exploding gradient problems.

Why ResNets Work

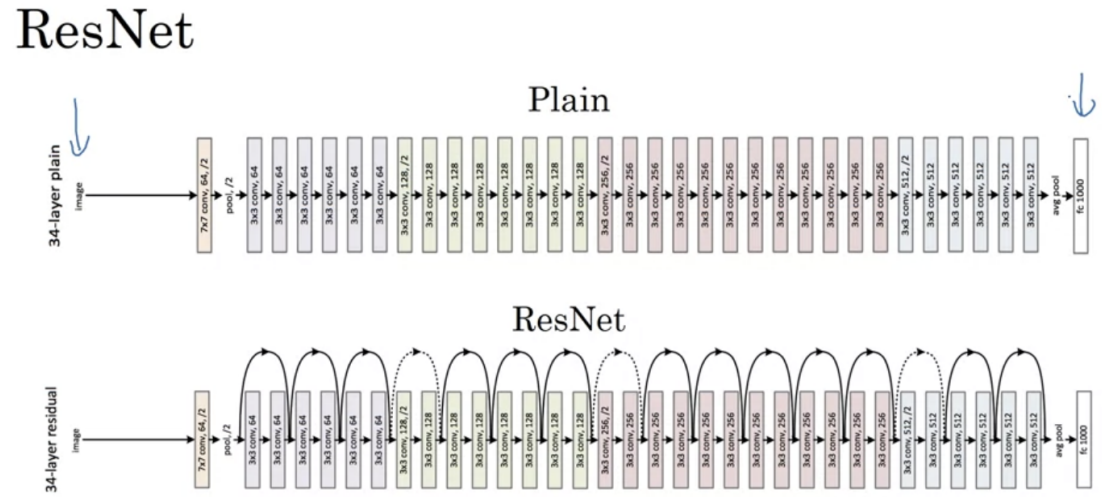
The identify function is easy for a residual block to learn. If there is a weight of zero, the layer becomes g(a^[l]). Skipping this layer is therefore equivalent.

Therefore if we have the option to skip a connection, it can't hurt performance.

It's easy for residual layers to learn the identity function.

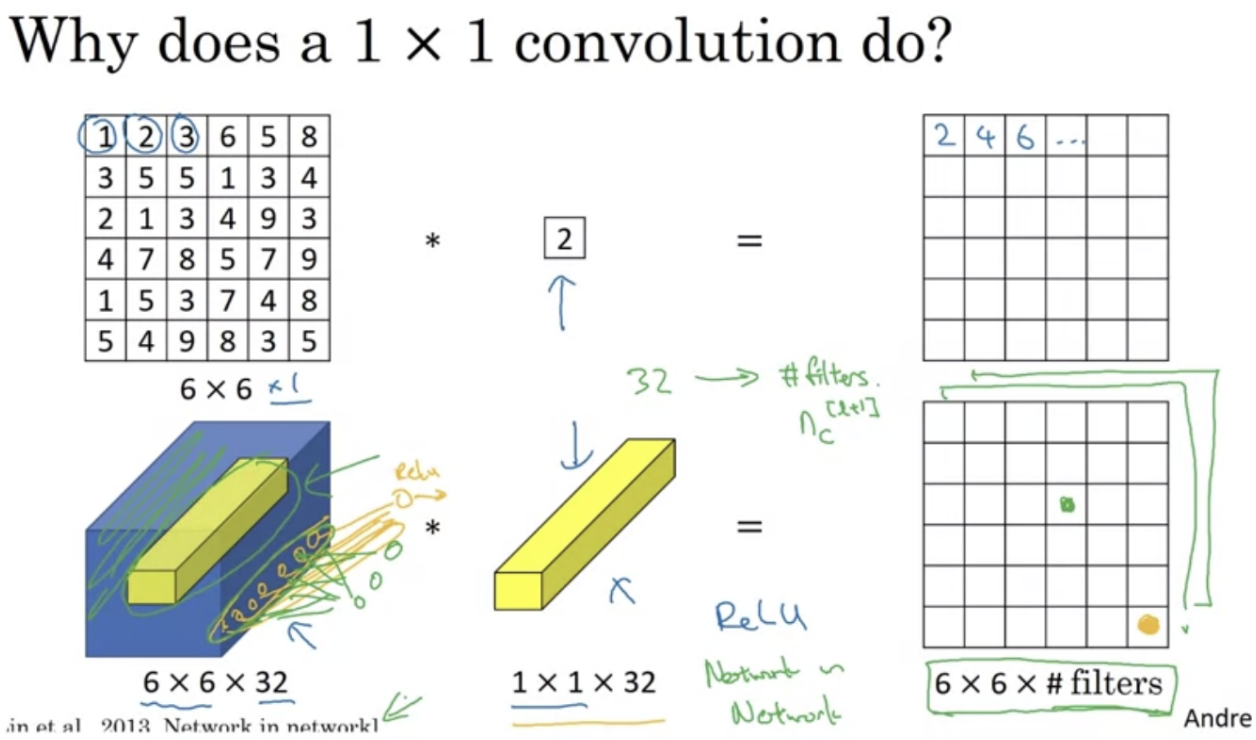


A lot of same convolutions are used in ResNets to keep the dimensions the same. This allows for short-circuit connections.

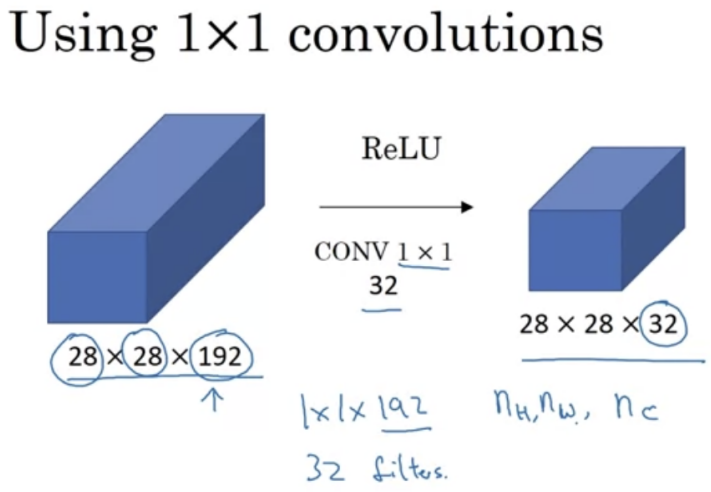


Networks in Networks and 1x1 Convolutions

A convolution by a 1x1 filter is a scalar multiplication. Doing this with a 1x1xn_H allows you to change your dimension. This allows you to carry out non-trivial computations.

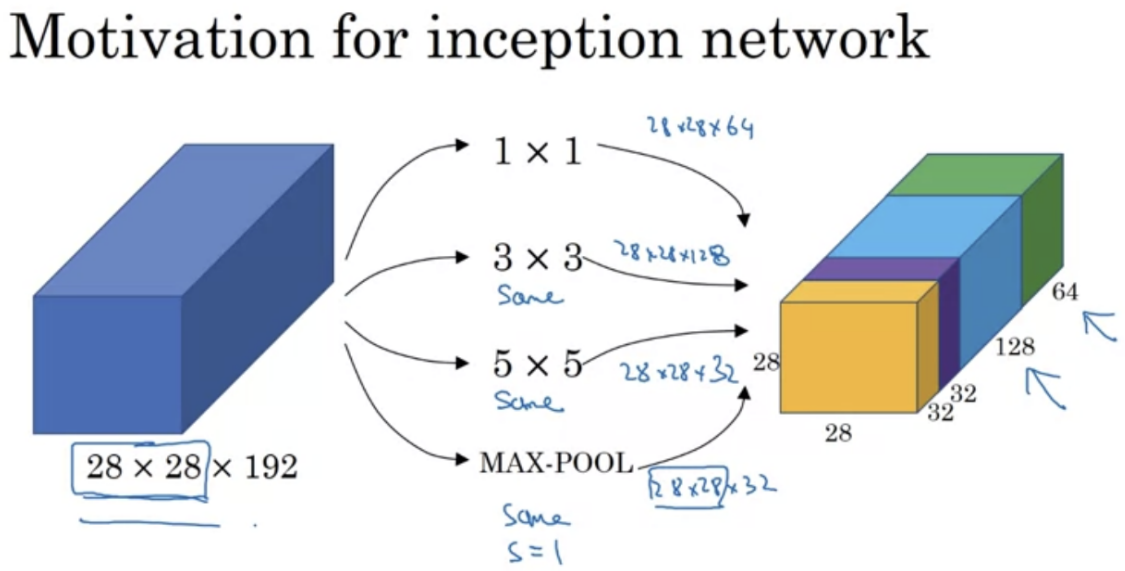


1x1 convolutions are sometimes called network in networks.

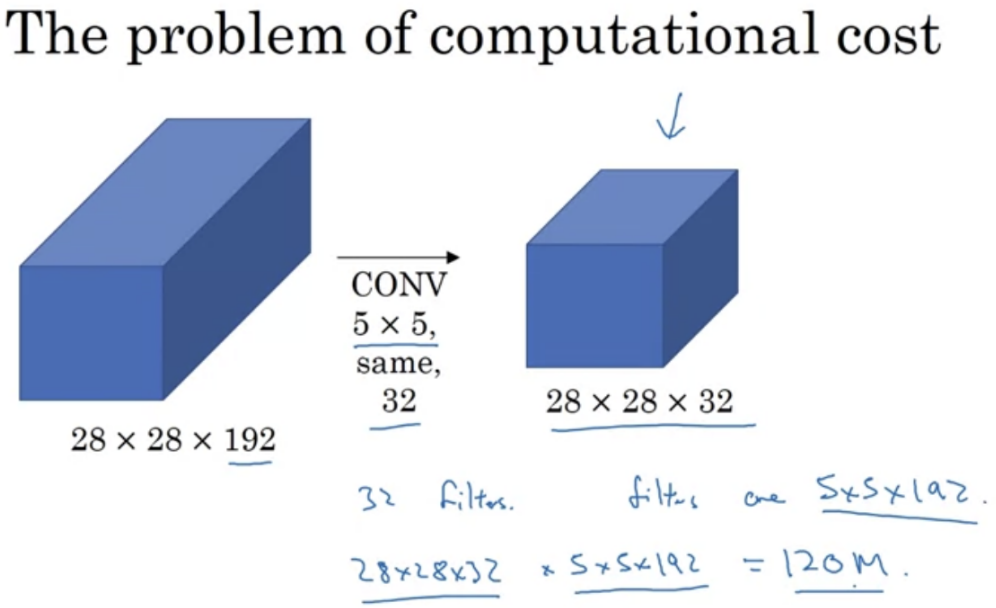


Here we use 1x1 convolutions to shrink n_C.

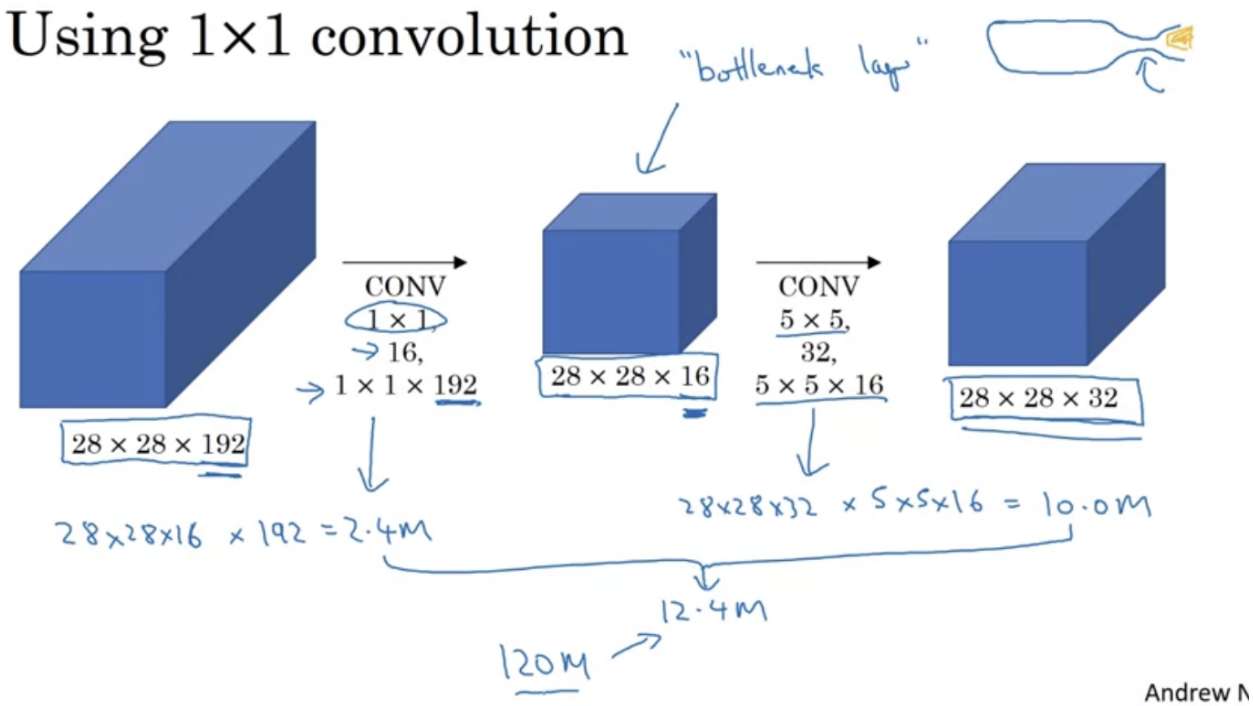
Inception Network Motivation



This is an inception module which is the heart of the inception network. They perform a variety of different filter sizes and pooling types instead of making you choose one. They then concatenate the outputs. The problem with this is computational cost.

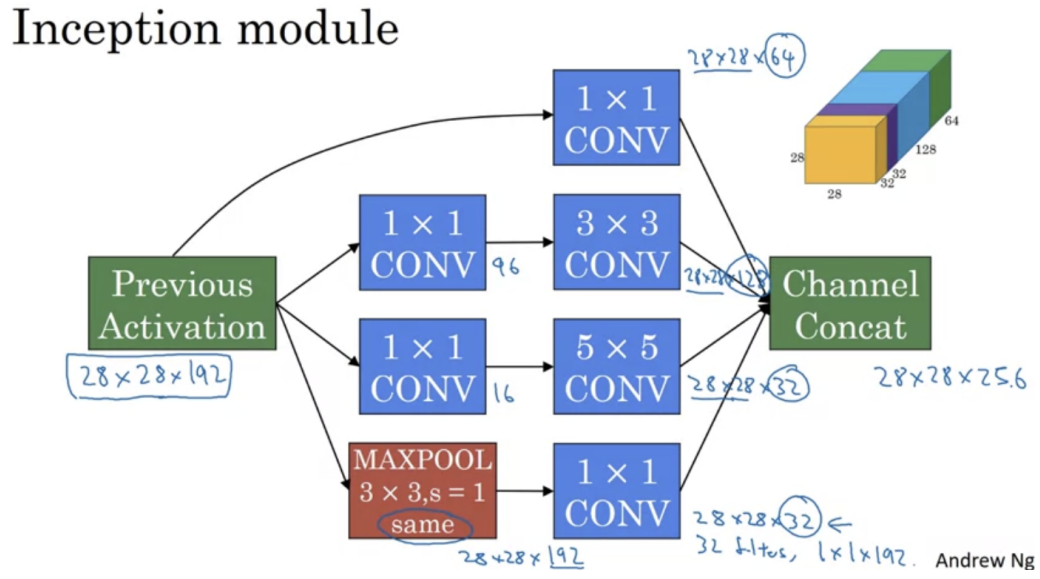


We can solve this by using a 1x1 convolution to reduce the channels.



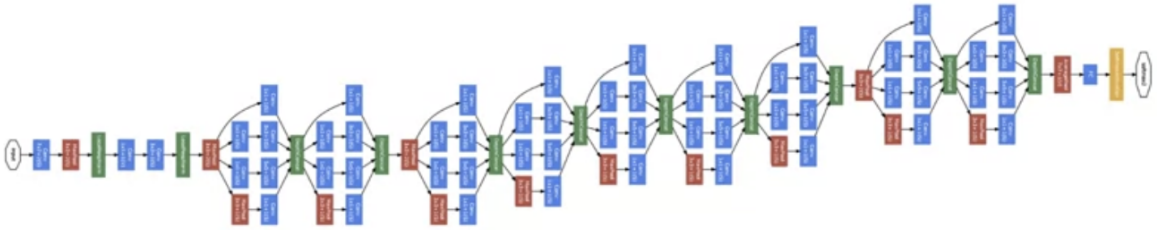
The middle layer is sometimes called a bottleneck layer. It doesn't seem to hurt performance.

Inception Network

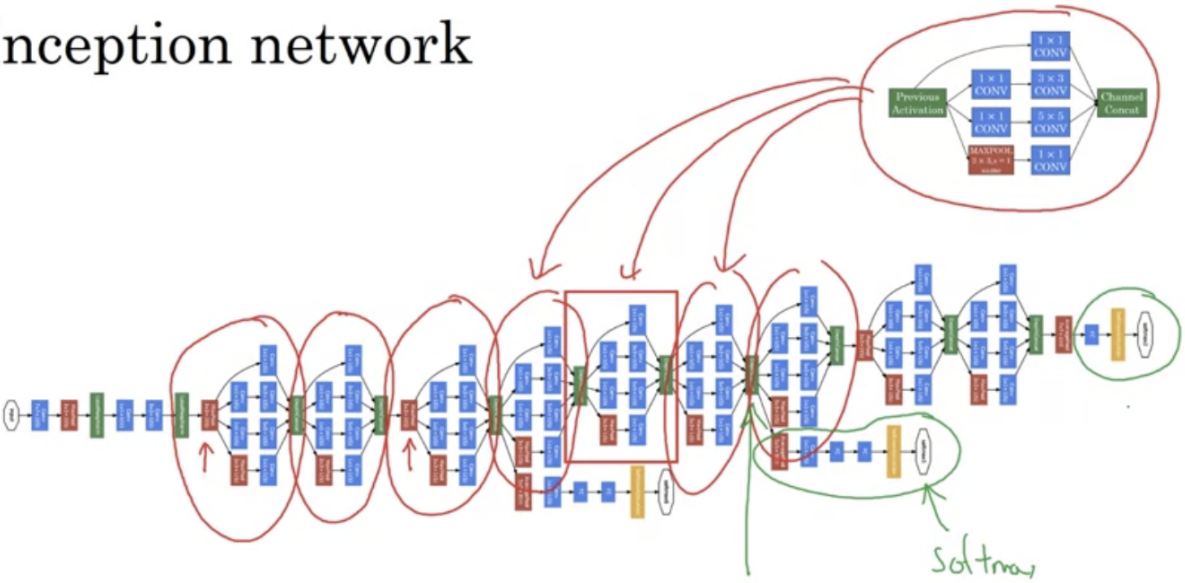


Inception networks put a lot of these modules together.

Inception network



Inception network



The side branches try to make predictions based on the outputs of the hidden layers. This helps prevent the network from overfitting. These values can be checked against the final output to prevent overfitting.

This network is called GoogLeNet in homage to Yann LeCun.

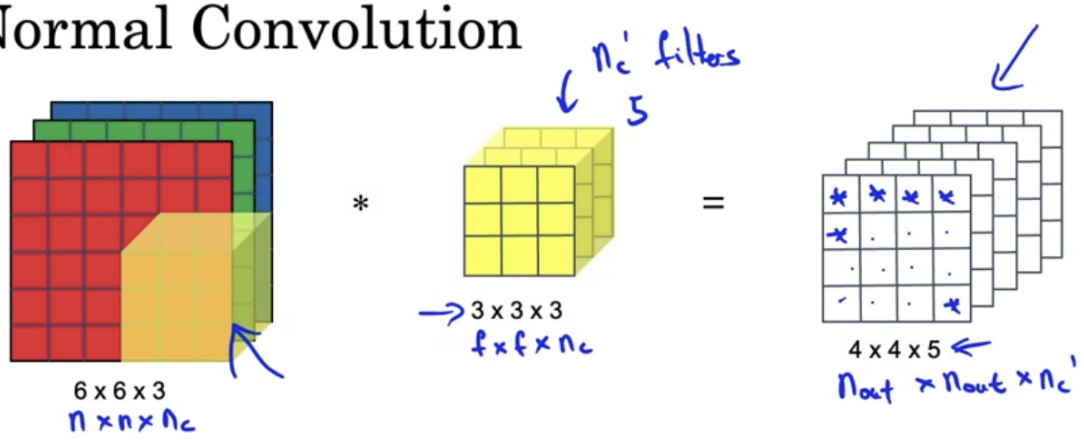
MobileNet

MobileNets are a architecture that performs well in low compute environments.

Motivation for MobileNets

- Low computational cost at deployment
- Useful for mobile and embedded vision applications
- Key idea: Normal vs. depthwise-separable convolutions

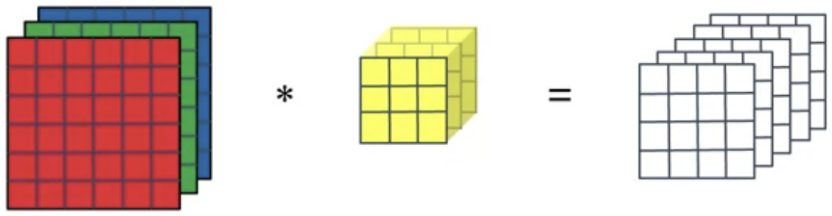
Normal Convolution



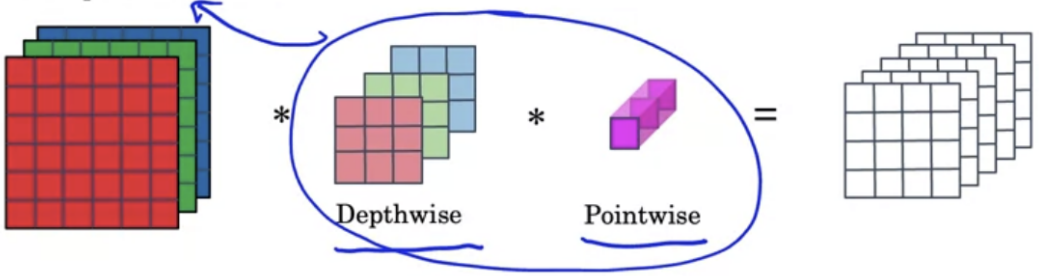
Computational cost	=	#filter params	x	# filter positions	x	# of filters
2160	=	3x3x3	x	4x4	x	5

Depthwise Separable Convolution

Normal Convolution



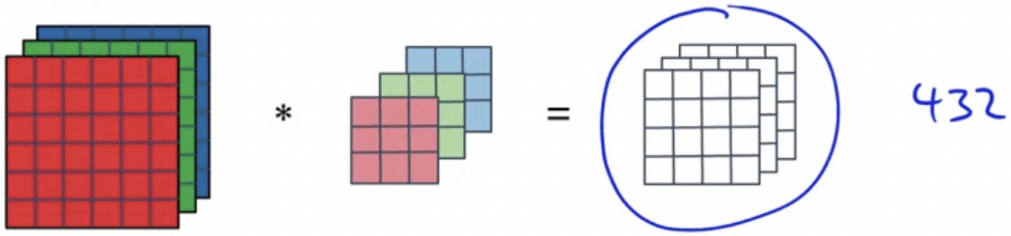
Depthwise Separable Convolution



Andrew

Depthwise Separable Convolution

Depthwise Convolution



Pointwise Convolution

Cost Summary

Cost of normal convolution 2160

Cost of depthwise separable convolution

depthwise + pointwise

432 + 240 = 672

$$\frac{672}{2160} = 0.31$$

$$= \frac{1}{n_c} + \frac{1}{f^2}$$

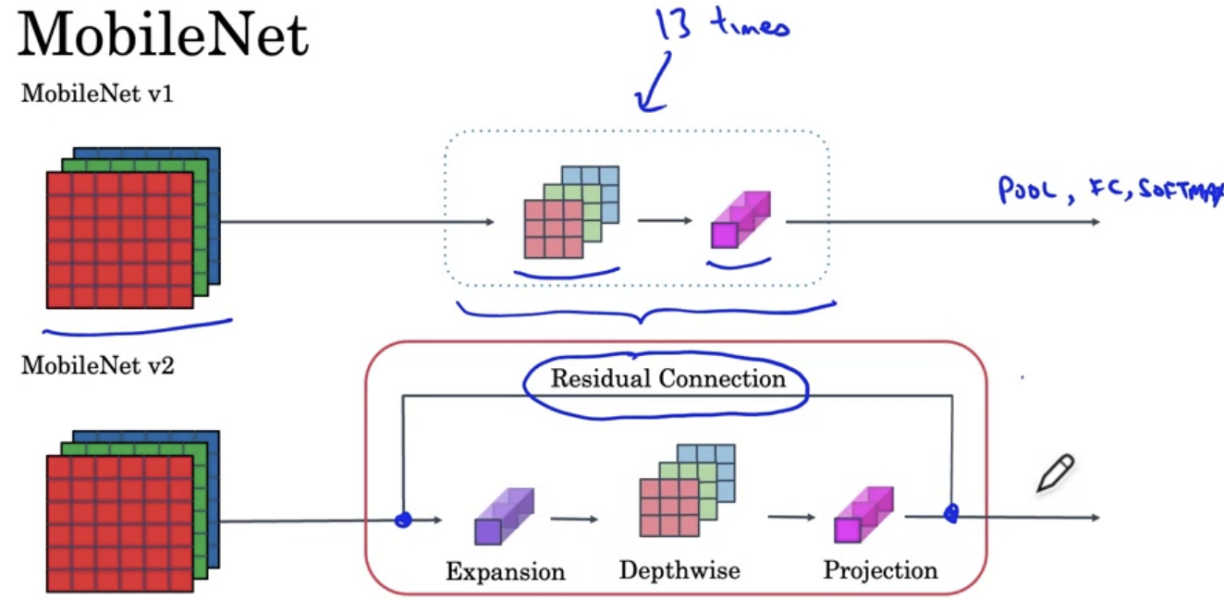
$$\frac{1}{5} + \frac{1}{9}$$

$$= \frac{1}{512} + \frac{1}{3^2}$$

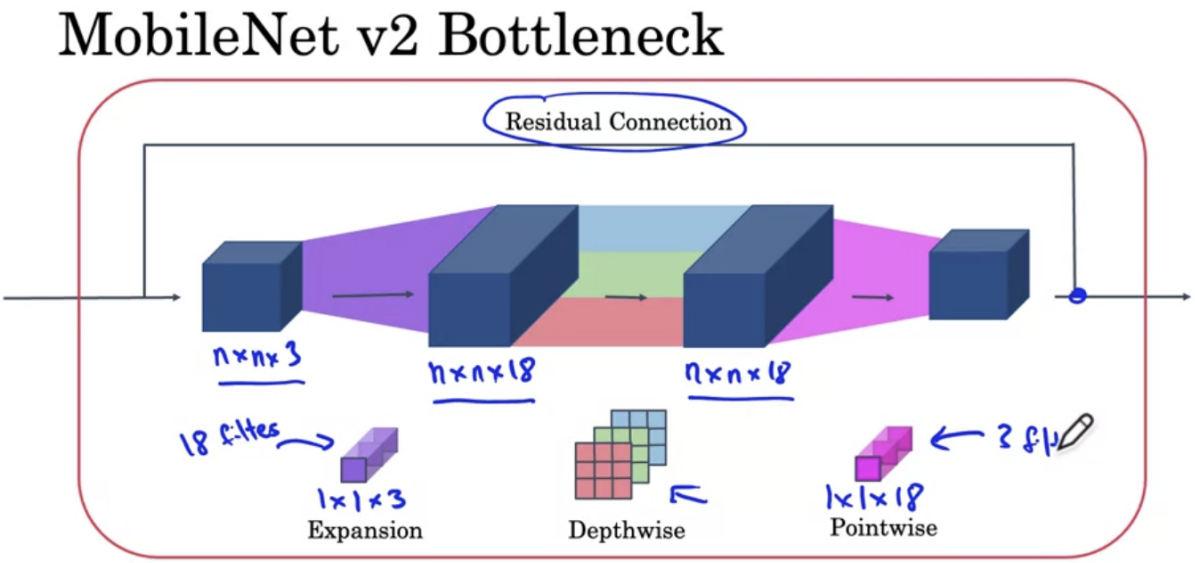
~10 times cheaper

MobileNet Architecture

In the MobileNet architecture the depthwise separable convolution is used instead of the normal convolution because it saves on compute.



The v2 also adds an expansion layer. The v2 model repeated the residual connection 17 times.



[Sandler et al. 2019, MobileNetV2: Inverted Residuals and Linear Bottlenecks] Andrew Ng

The bottleneck block accomplishes two things:

- Increases the size of the representation in the block allowing the network to learn a richer function
- Keeps the amount of memory (the size of the activations passed from layer to layer) small

EfficientNet
EfficientNet allows you to automatically scale up or down networks for a particular device.

You can vary the parameters

- r: resolution
- d: depth
- w: width

The network explores the best ways to scale these parameters to get the best performance within your computational budget.

To choose r, d, and w look at open source models to figure out how to scale the parameters.