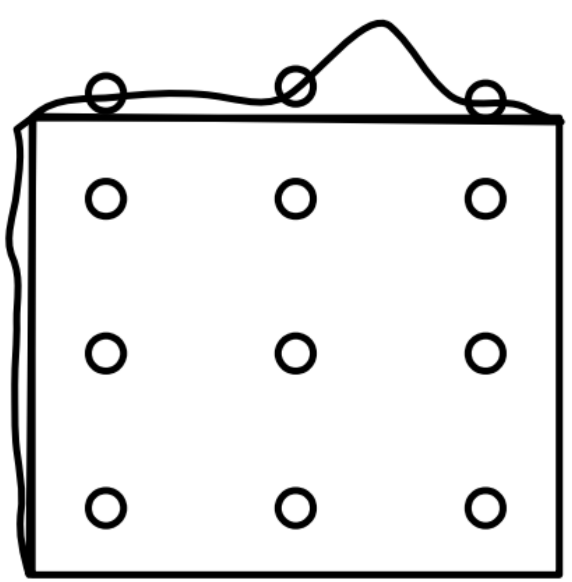
Practical Methodology

Arun Chauhan

Hyperparameter Search

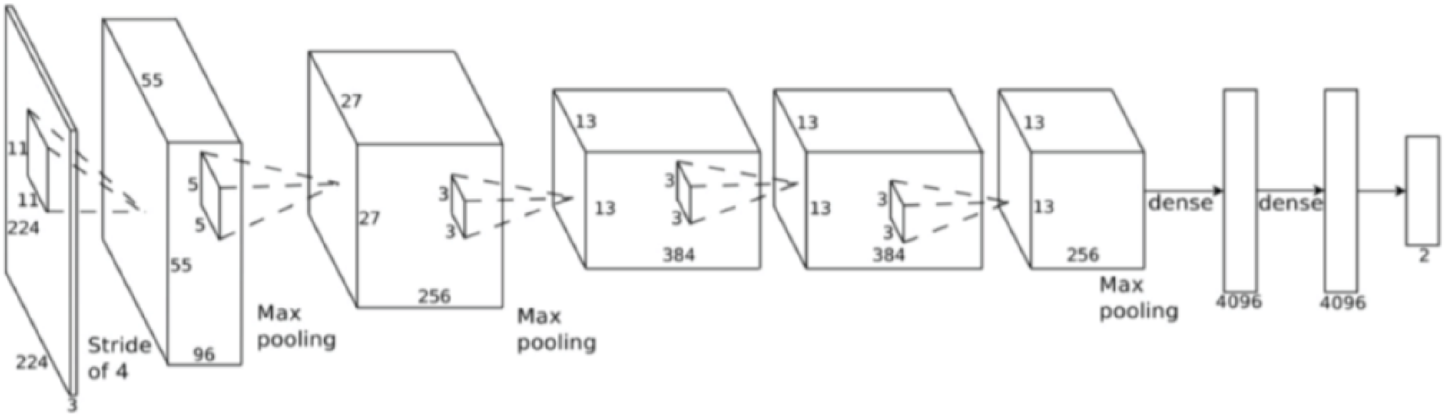


Grid Random

Transfer Learning/ Less Data

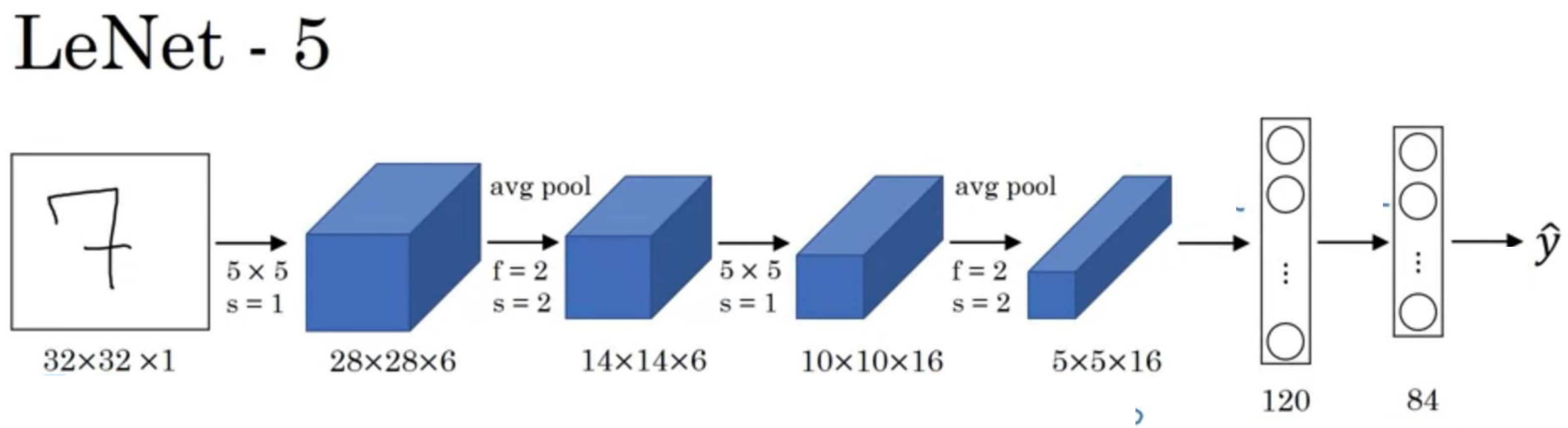
Practioners rarely train a CNN “from scratch”. Instead we could:

1. Take a pre-trained CNN model (e.g. AlexNet), and use its features network to compute **image features**, which we then use to classify our own images
2. Initialize our weights using the weights of a pre-trained CNN model (e.g. AlexNet)



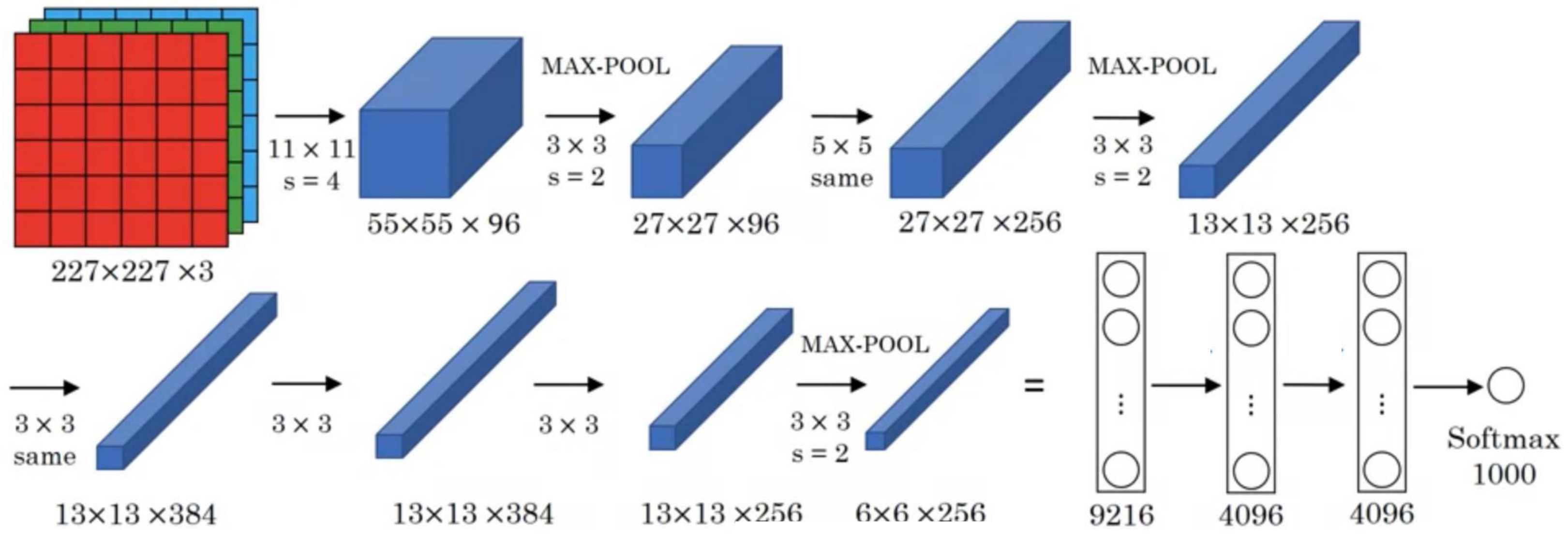
Variants of CNN

Classic Networks



[LeCun et al., 1998. Gradient-based learning applied to document recognition]

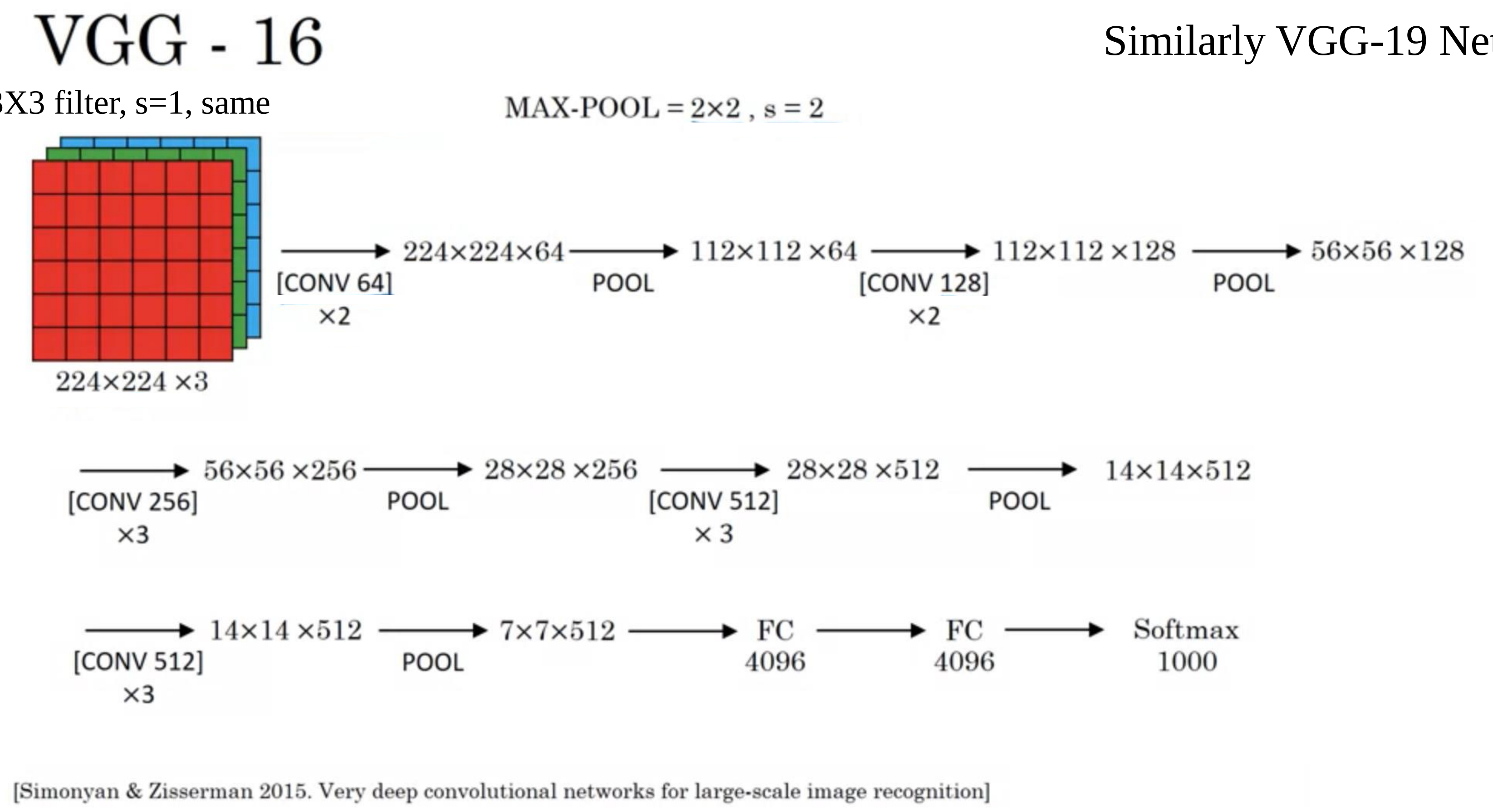
AlexNet



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

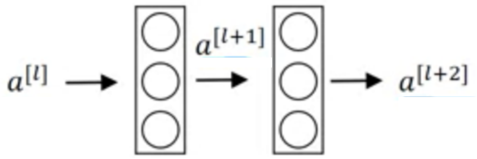
Similarly VGG-19 Network

CONV=3X3 filter, s=1, same



ResNets

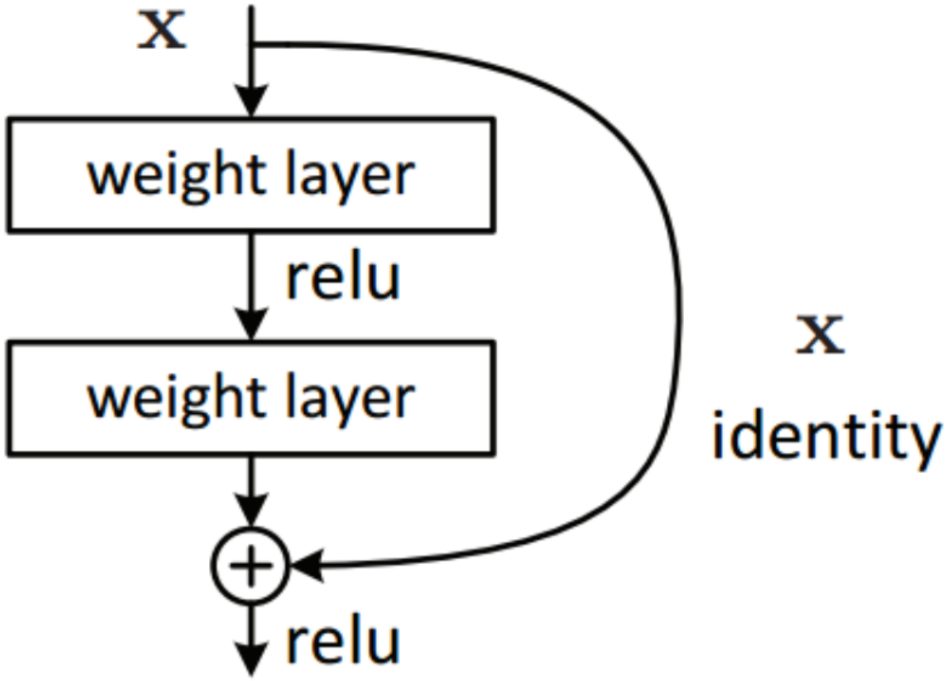
Residual block



$$z^{[l+1]} = W^{[l+1]} a^{[l]} + b^{[l+1]} \quad a^{[l+1]} = g(z^{[l+1]}) \quad z^{[l+2]} = W^{[l+2]} a^{[l+1]} + b^{[l+2]} \quad a^{[l+2]} = g(z^{[l+2]})$$

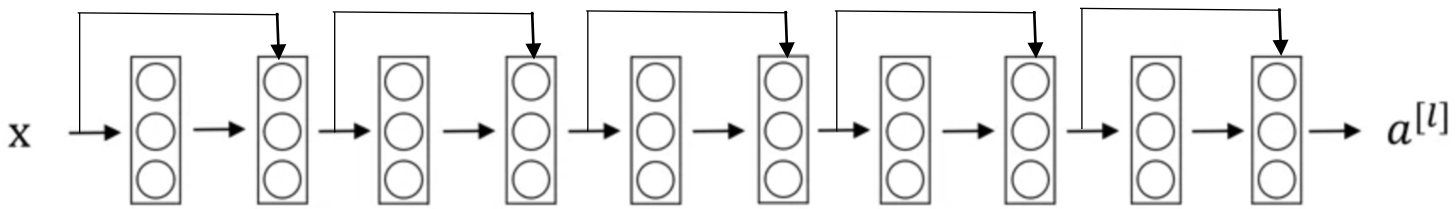
[He et al., 2015. Deep residual networks for image recognition]

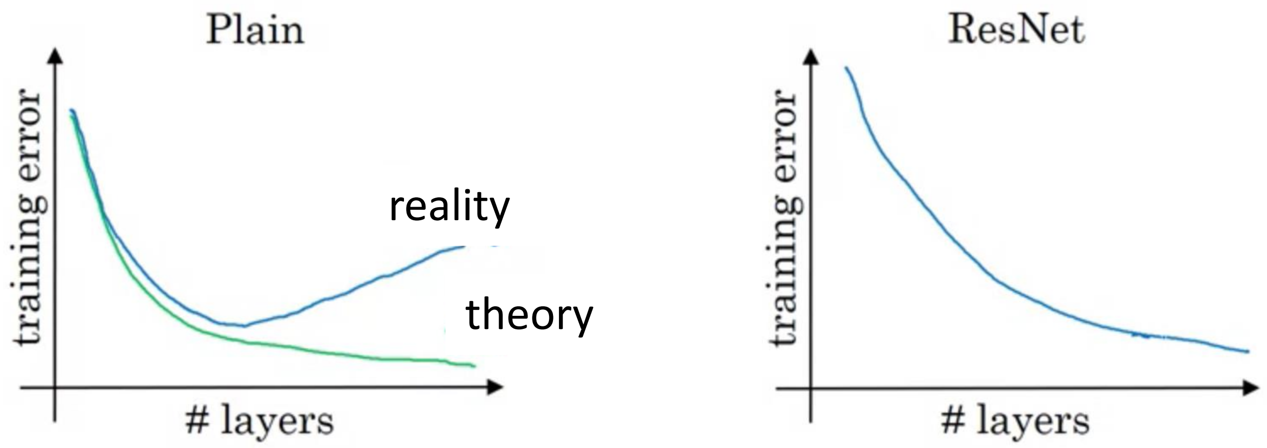
Single Residual Block



https://towardsdatascience.com/understanding-residual-networks-9add4b664b03

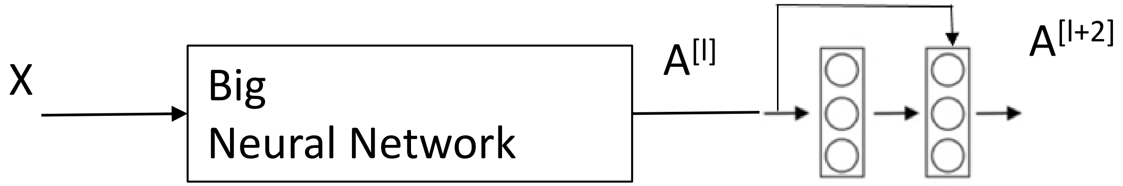
Residual Network





[He et al., 2015. Deep residual networks for image recognition]

Why do residual networks works?



ReLU $\quad a>=0$

$A^{[l+2]} = g(Z^{[l+2]} + A^{[l]})$

$= g(W^{[l+2]} A^{[l+1]}+b^{[l+2]} + A^{[l]})$

If

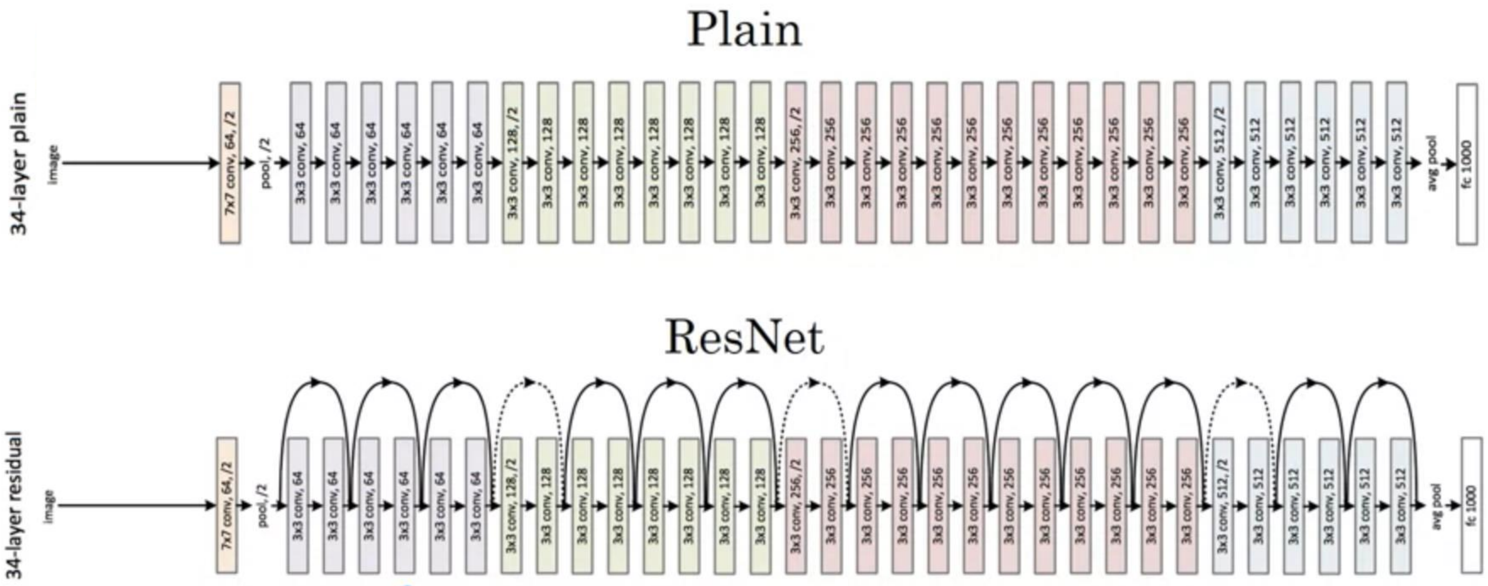
$W^{[l+2]}=0$,$b^{[l+2]}=0$

Then

$A^{[l+2]} = g(A^{[l]})$

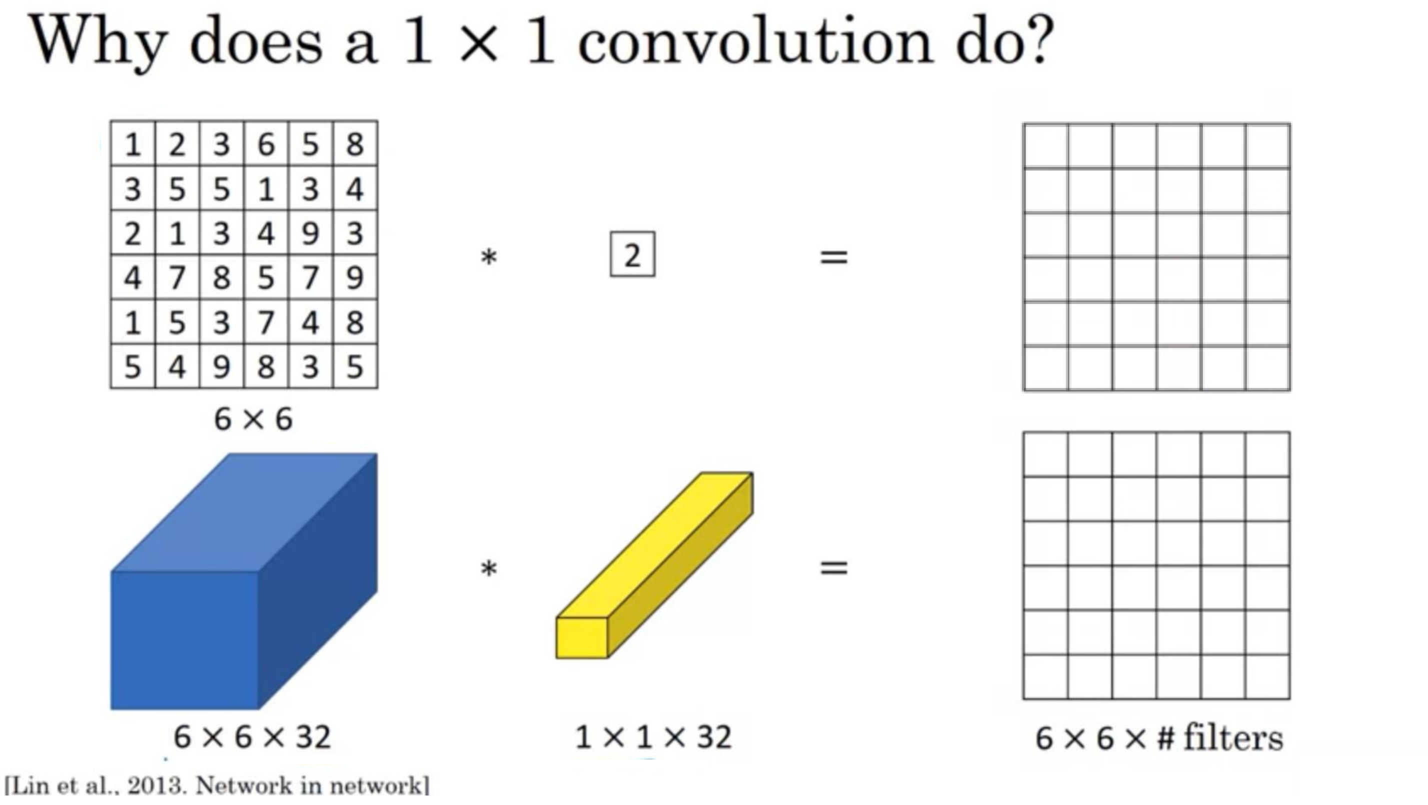
$=A^{[l]}$

ResNet

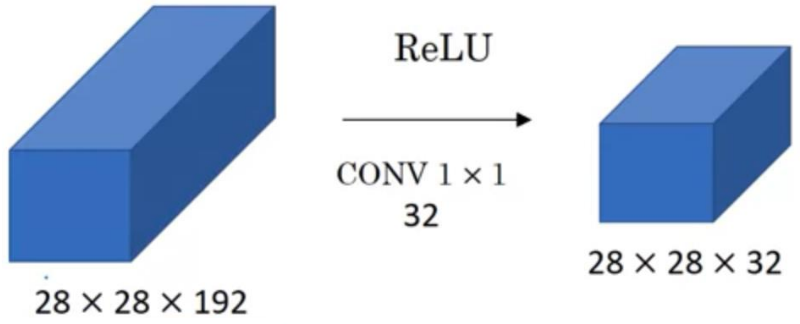


[He et al., 2015. Deep residual networks for image recognition]

Networks in Networks and 1x1 Convolutions

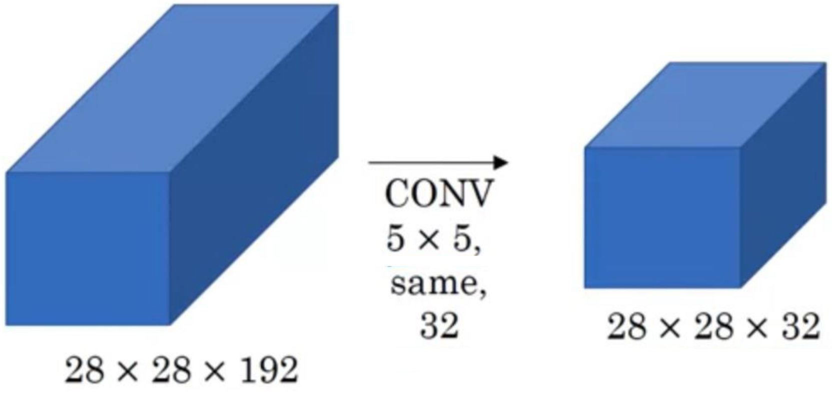


Using 1×1 convolutions



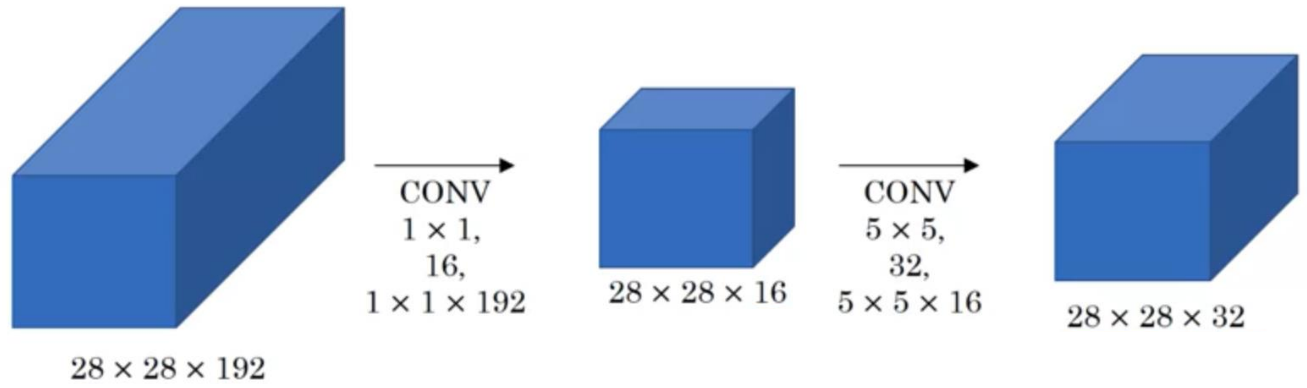
[Lin et al., 2013. Network in network]

The problem of computational cost



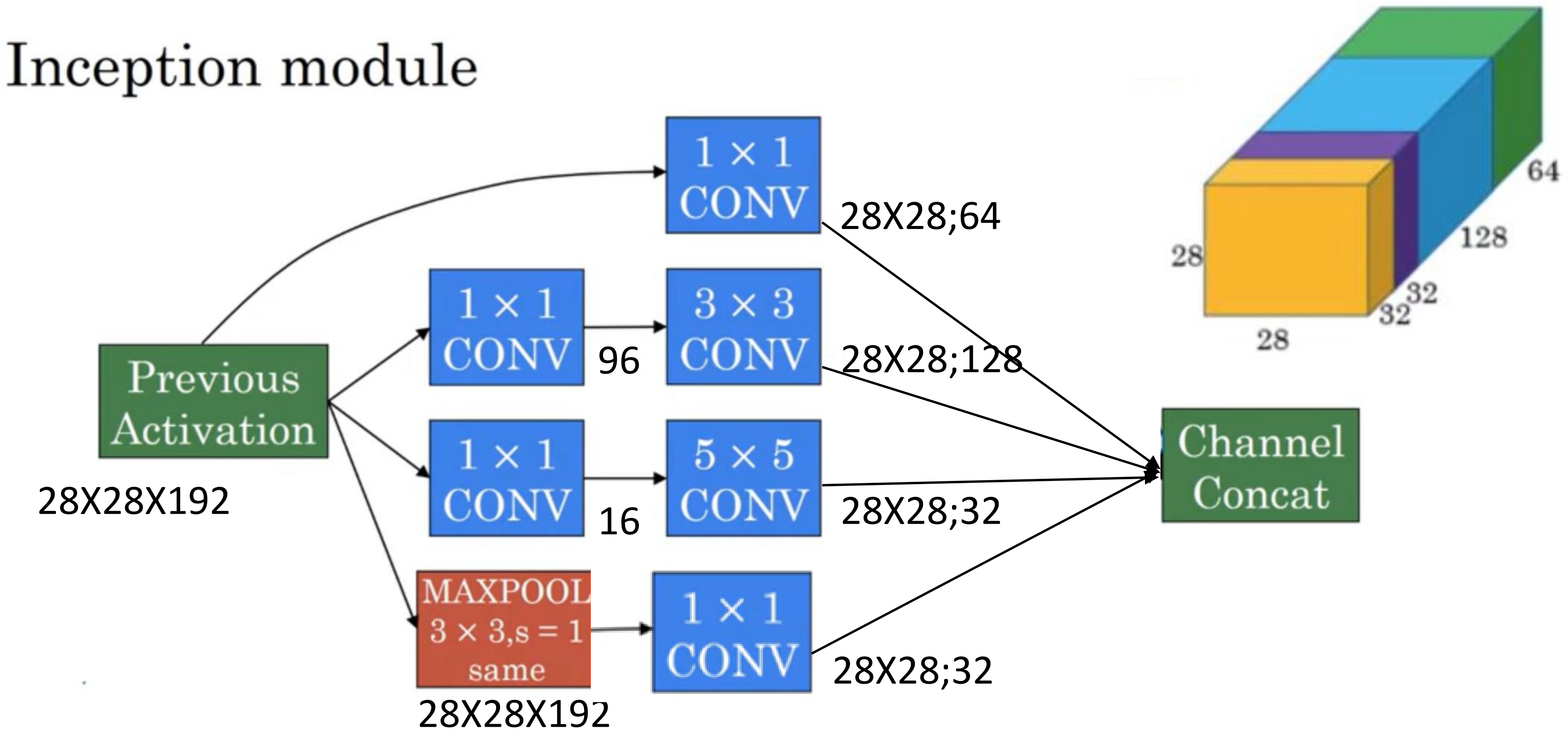
28X28X32 X 5X5X192 ≈ 120 M

Using 1×1 convolution

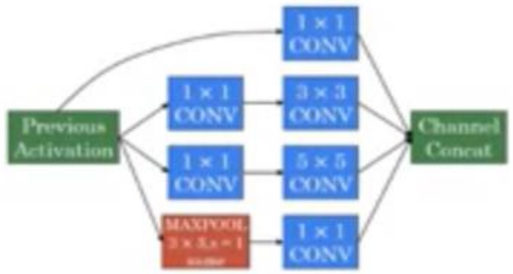


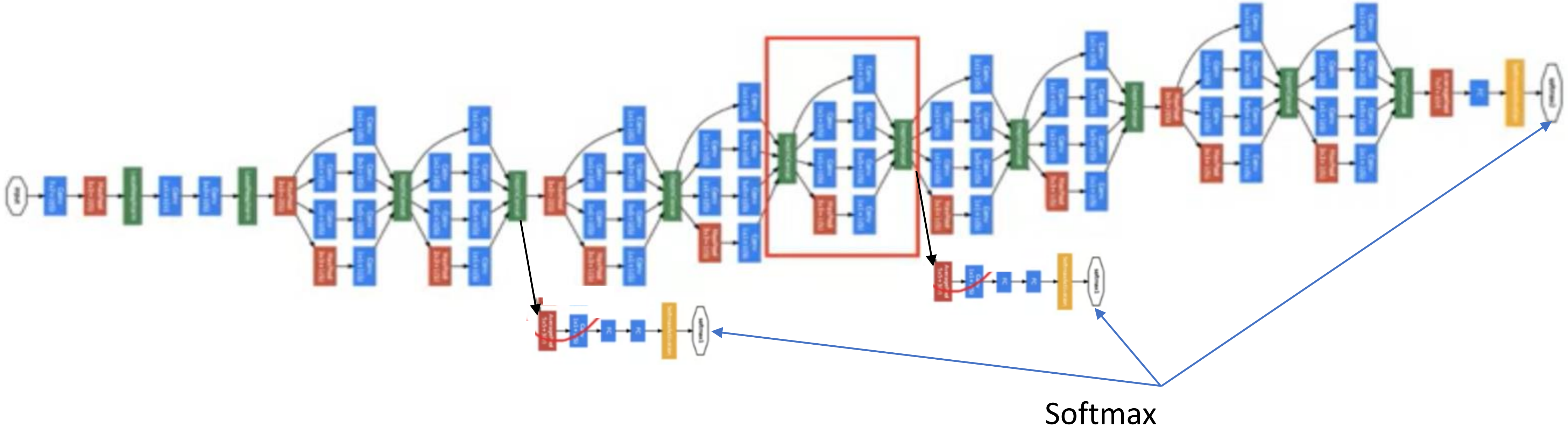
28X28X16	X	192	≈	2.4M
28X28X32	X	5X5X16	≈	10M
Total			≈	12.4 M

Inception Network



Inception network

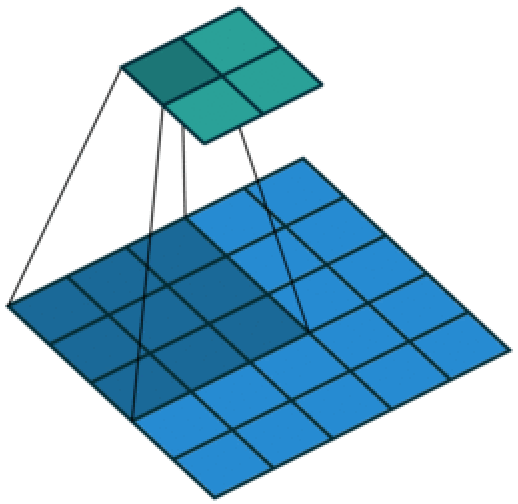


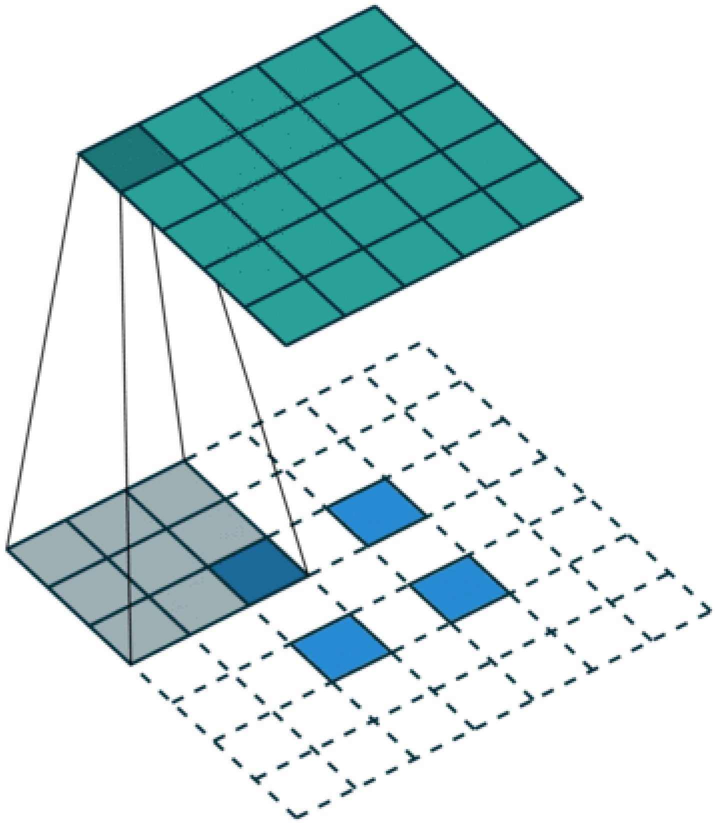


[Szegedy et al., 2014, Going Deeper with Convolutions]

Autoencoders

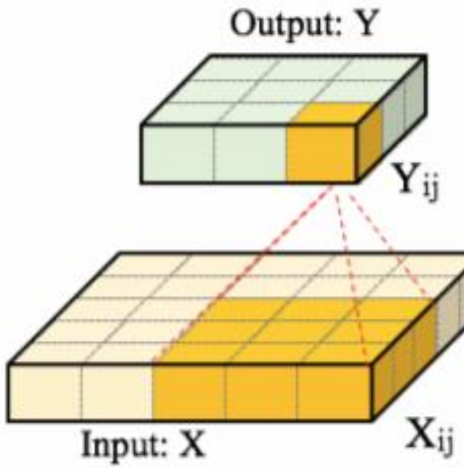
Convolution Vs Deconvolution



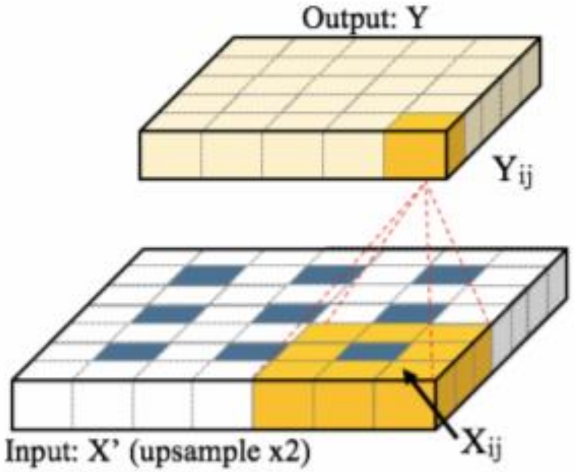


https://towardsdatascience.com/types-of-convolutions-in-deep-learning-717013397f4d

Convolution Vs Deconvolution



(**a**) Convolutional layer: the input size is $W_1 = H_1 = 5$; the receptive field $F = 3$; the convolution is performed with stride $S = 1$ and no padding ($P = 0$). The output Yis of size $W_2 = H_2 = 3$.



(**b**) Transposed convolutional layer: input size $W_1 = H_1 = 3$; transposed convolution with stride $S = 2$; padding with $P = 1$; and a receptive field of $F = 3$. The output Yis of size $W_2 = H_2 = 5$.

What are autoencoders?

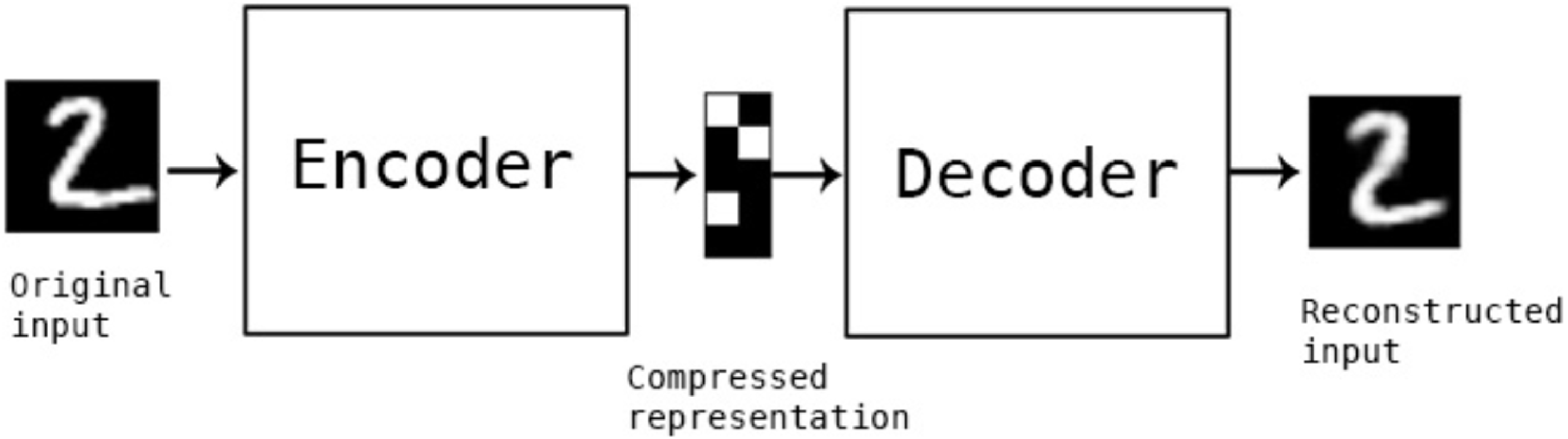


Image Autoencoder

An image autoencoder is a neural network used to find a **low-dimensional representation** of some images. This is an **unsupervised learning** task (no labels).

An image autoencoder has two components:

1. An **encoder** neural network that takes the image as input, and produces a low-dimensional embedding.
2. A **decoder** neural network that takes the low-dimensional embedding as input, and reconstructs the image.

Idea: A good, low-dimensional representation should allow us to reconstruct everything about the image.

The components of an autoencoder

Encoder:

- Input = image
- Output = low-dimensional embedding

Decoder:

- Input = low-dimensional embedding
- Output = image

Why autoencoders?

- Dimension reduction:
  - find a low dimensional representation of the image
- Image Generation:
  - generate new images not in the training set

Autoencoders are considered a **generative model**.

How to train autoencoders?

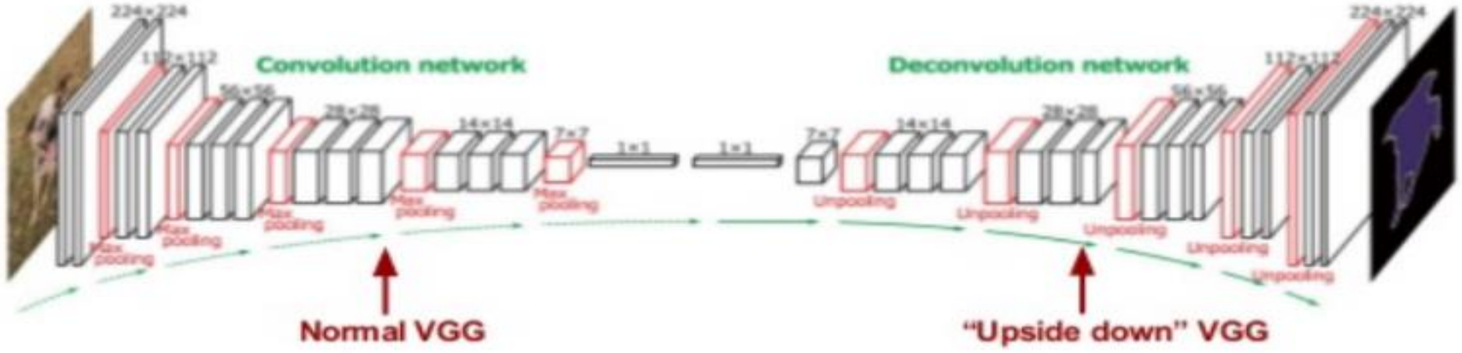
- Loss function:
  - How close were the reconstructed image from the original?
  - **Mean Square Error Loss**: look at the mean square error across all the pixels.

Architectures with Transpose Convolution

More than one upsampling layer

DeconvNet:

VGG-16 (conv+Relu+MaxPool) + mirrored VGG (Unpooling+'deconv'+Relu)



Noh et al, “[Learning Deconvolution Network for Semantic Segmentation]”, ICCV 2015

- END with Intro to Variational Autoencoder.