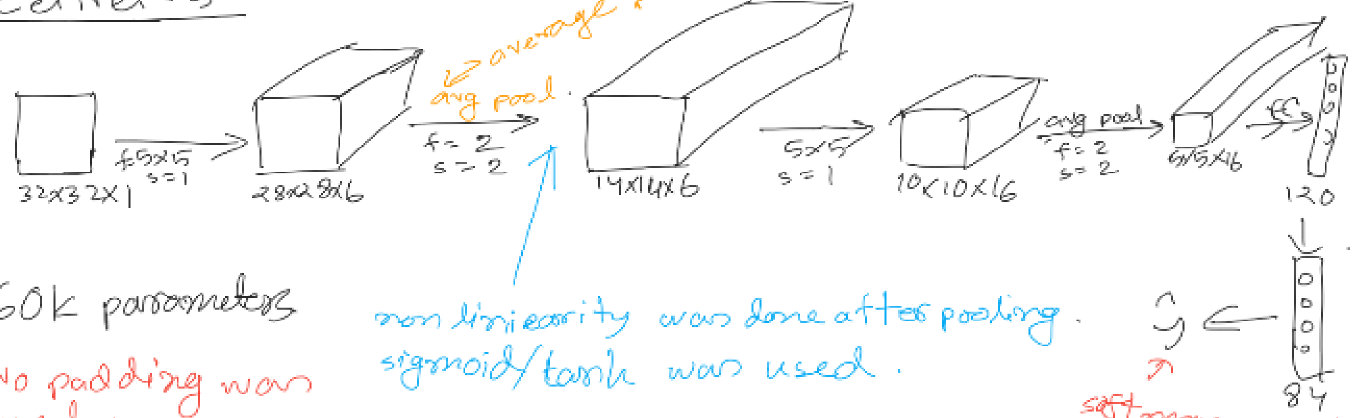
C4w2

Classic neural network architecture :-

- LeNet-5
- AlexNet
- VGG
- ResNet (155 layer).

LetNet-5



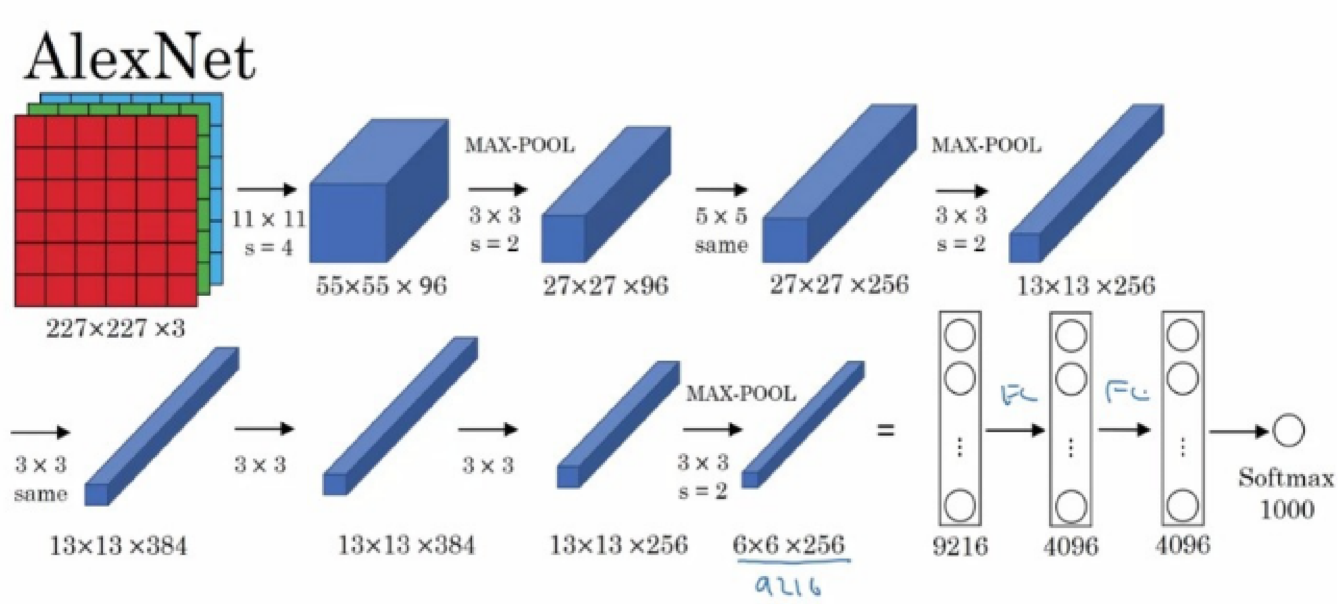
60K parameters

No padding was used.

non linearity was done after pooling.
sigmoid/tanh was used.

softmax wasn't used.

AlexNet :-



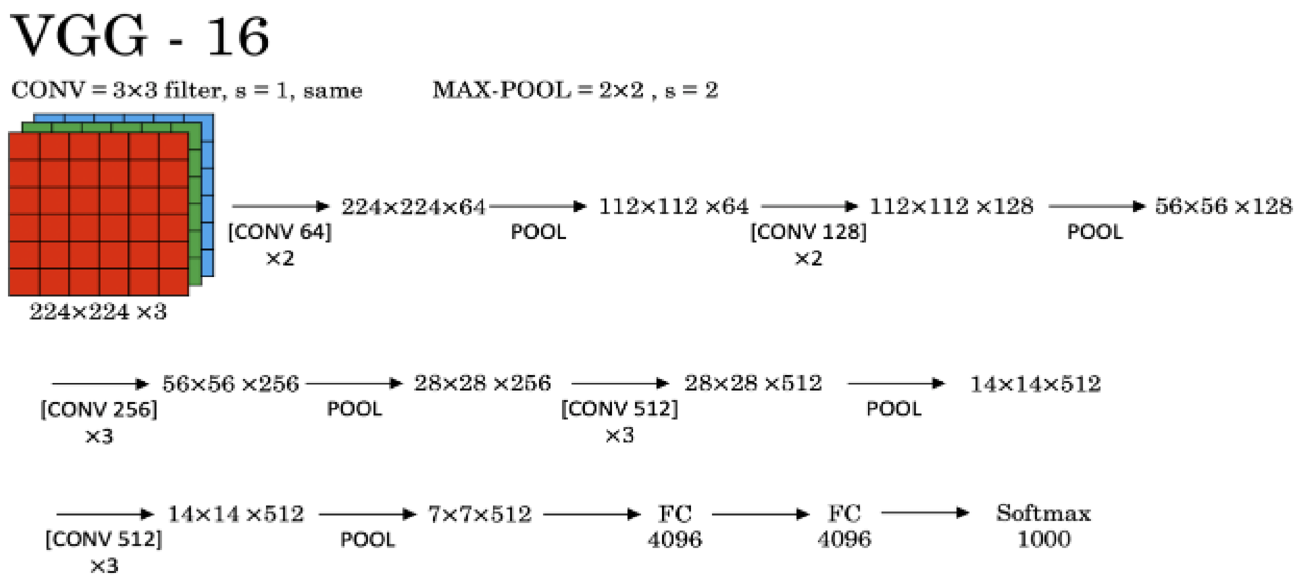
[Krizhevsky et al., 2012. ImageNet classification with deep convolutional neural networks]

Andrew Ng

- 60M parameter.

- ReLu
- Multiple GPU.
- Local response normalization (LRN) - it normalizes along the channel.

VGG-16



[Simonyan & Zisserman 2015. Very deep convolutional networks for large-scale image recognition]

Andrew Ng

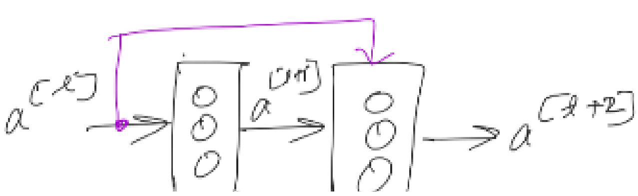
160M parameters.

it's simplicity makes it popular (like same filter size)

Residual Network (ResNet):-

Very, very deep neural networks are harder to train because of vanishing & exploding gradient.

Residual block:-



"short cut"/skip connection

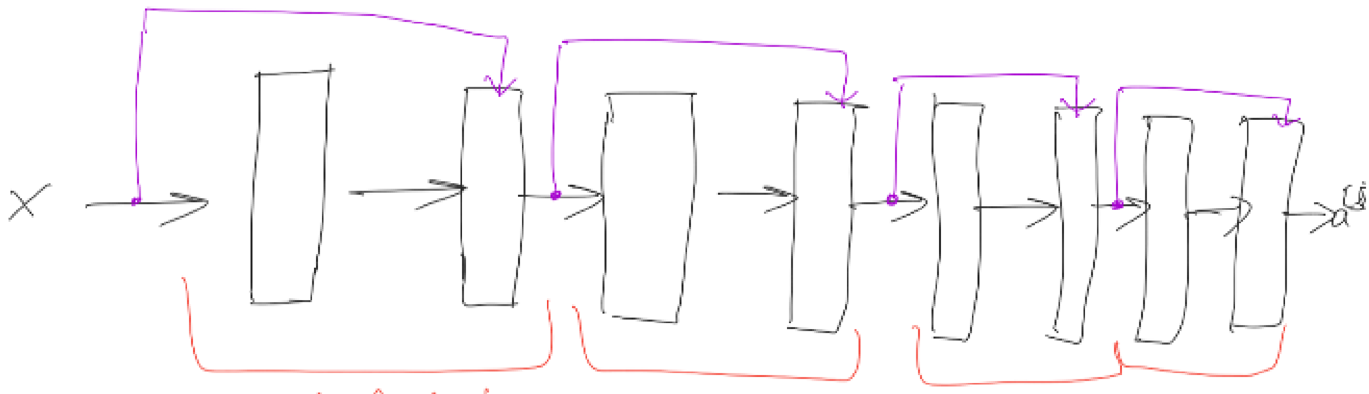
$a^{[l]} \to$ linear $\to$ ReLU $\xrightarrow{a^{[l+1]}}$ linear $\to \oplus \to$ ReLU $\to a^{[l+2]}$

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

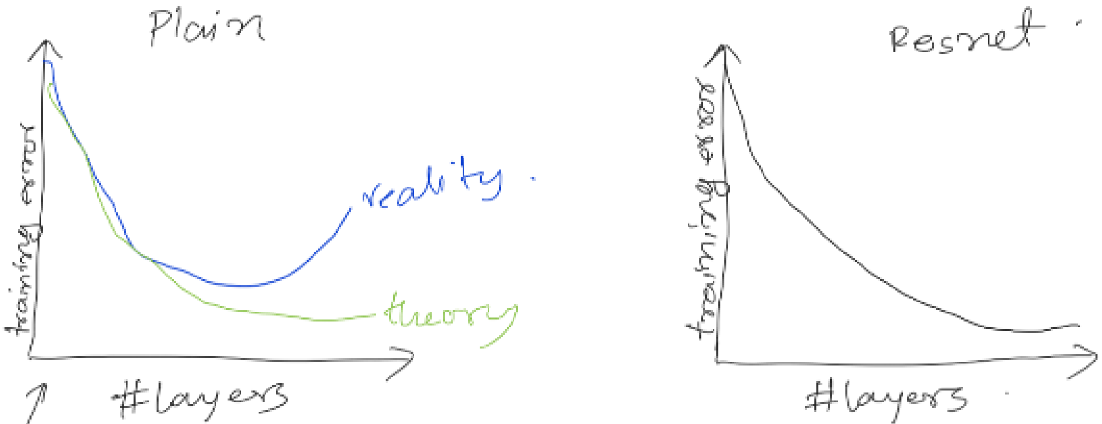
$z^{[l+2]} = W^{[l+2]} a^{[l+1]} + b^{[l+2]}$ $\qquad$ $\cancel{a^{[l+2]} = g(z^{[l+2]})}$

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

Residual network:-

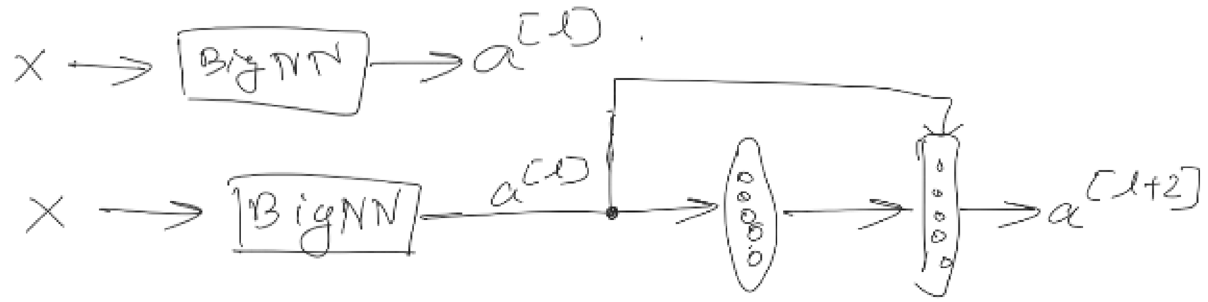


Residual block.



this might happen for vanishing/exploding gradients.

Why resNetwork:-



$$\underbrace{a^{[l+2]}}_{256} \rightarrow g(z^{[l+2]} + a^{[l]})$$

$$\rightarrow g(\cancel{w^{[l+2]} a^{[l+1]} + b^{[l+2]}} + \underbrace{W_s}_{256 \times 128} \underbrace{a^{[l]}}_{128})$$

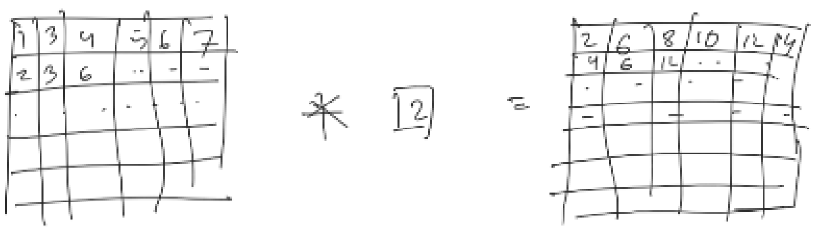
→ 0 if this term becomes 0 (vanishing gradient / regularization)

$$\rightarrow g(a^{[l]})$$

Identity function is easy to learn for residual block.

if dimension of $a^{[l+2]}$ & $a^{[l]}$ is different then we have to we have to multiply the term $a^{[l]}$ with w_s. W_s might be fixed value matrix. or it may 0 pads the $a^{[l]}$ to match the dimensions of $a^{[l+2]}$

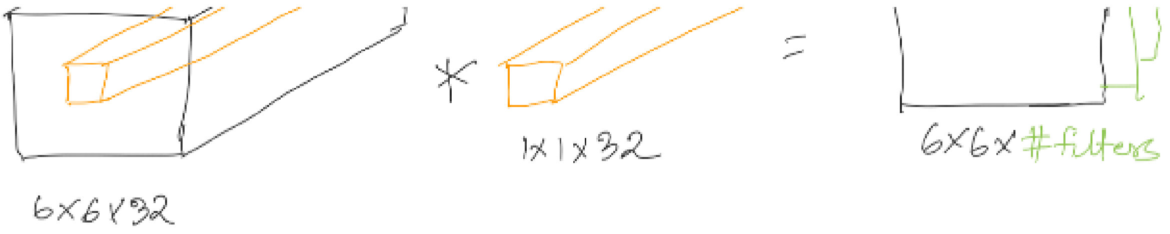
What does 1X1 filter do?



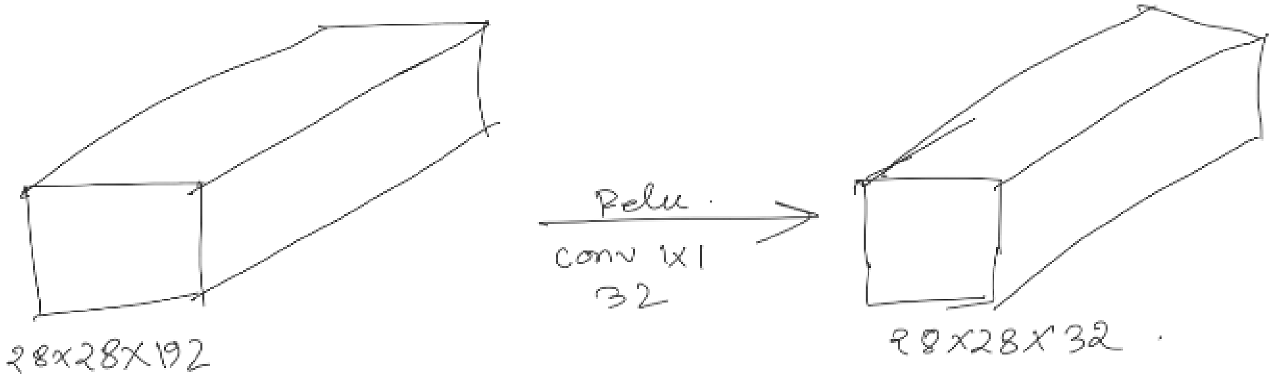
it seems 1x1 filter just multiply the orignal data. which is not usefull.

But it is useful in multidimensional image

multidimension:-



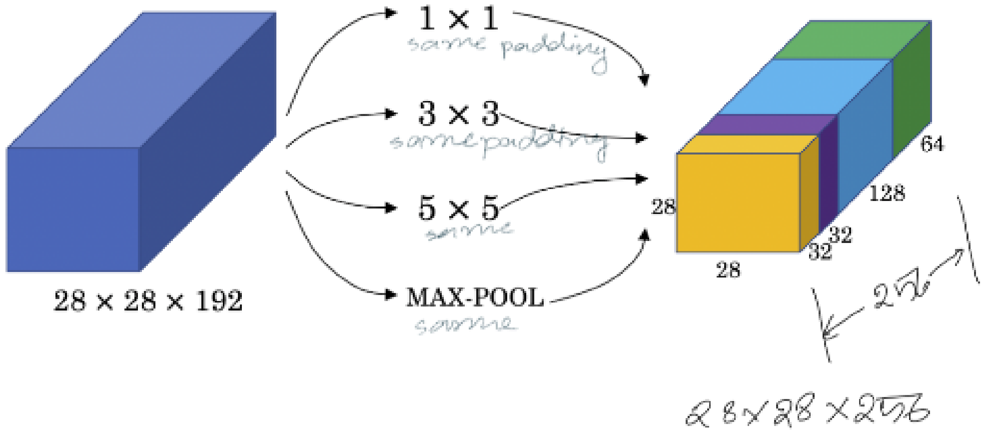
it is also called networ in network architecture.



if we have a large channel and want to shrink the channel then we can use 1x1 filter to shrink the channel pooling layer is used to shrink the height & width.

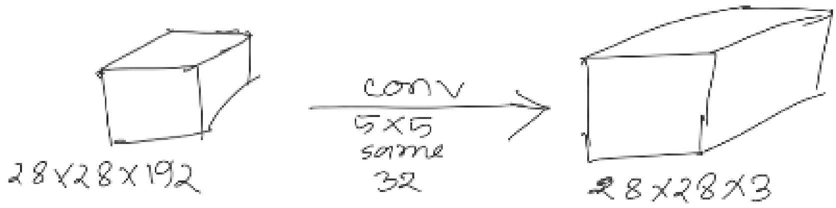
Inception network:- (GoogleNet)

Instead of choosing the filter size/max pooling/avg pooling. inception network does all of them at the same time. And it let the parameter learn whatever it wants to learn



Computational cost is higer for inception :-

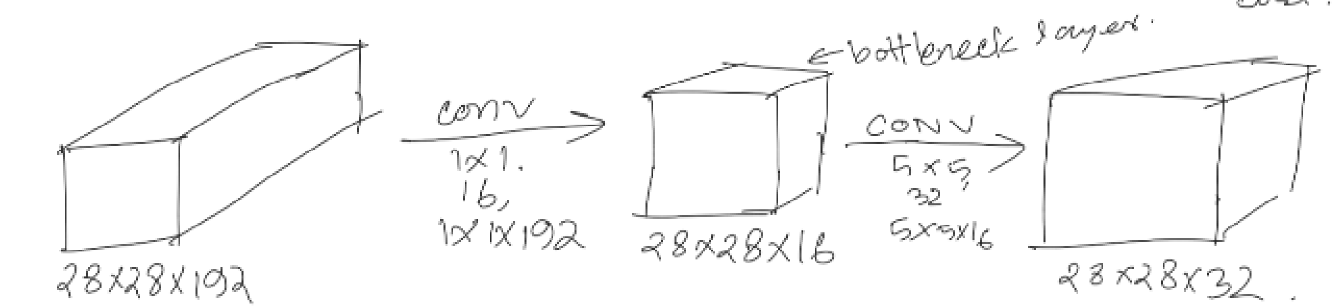
Let's focus on the computational cost of 5x5 filter



each of the 32 filters are 5x5x192

so total computation, $28 \times 28 \times 32 \times 5 \times 5 \times 192 = 120M$.

We can use 1x1 convolution to reduce the computation cost :-



Cost :-

$28 \times 28 \times 16 \times 1 \times 1 \times 192 = 2.4\ M$. $\quad 28 \times 28 \times 32 \times 5 \times 5 \times 16 = 10M$.

$2.4M + 10M = \boxed{12.4\ M}$