Machine Learning

Lect 11

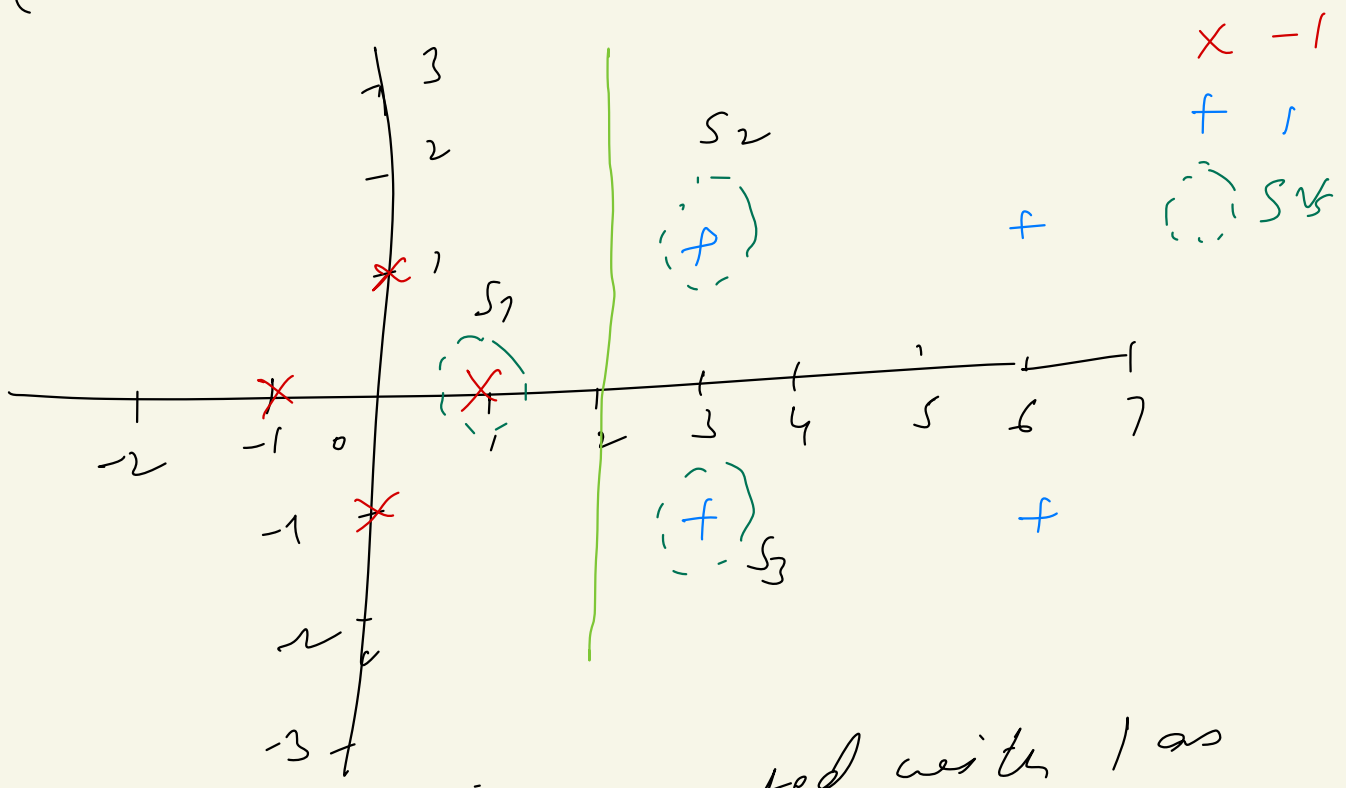
Dr. Swarna
Mukherjee

SVM Numerical

Suppose we are given the following data points

$\left\{ \begin{pmatrix} 3 \\ 1 \end{pmatrix}, \begin{pmatrix} 3 \\ -1 \end{pmatrix}, \begin{pmatrix} 6 \\ , \end{pmatrix}, \begin{pmatrix} 6 \\ -1 \end{pmatrix} \right\}$ true labeled $+1$

$\left\{ \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ -1 \end{pmatrix}, \begin{pmatrix} -1 \\ 0 \end{pmatrix} \right\}$ very labeled -1



Each vector is augmented with 1 as bias input

$S_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \tilde{S}_1 = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}$

$S_2 = \begin{pmatrix} 3 \\ 1 \end{pmatrix} \quad \tilde{S}_2 = \begin{pmatrix} 3 \\ 1 \\ 1 \end{pmatrix} \quad S_3 = \begin{pmatrix} 3 \\ -1 \end{pmatrix} \tilde{S}_3 \begin{pmatrix} 3 \\ -1 \\ 1 \end{pmatrix}$

$$\alpha_1 \tilde{s}_1 \tilde{s}_1 + \alpha_2 \tilde{s}_2 \tilde{s}_1 + \alpha_3 \tilde{s}_3 \tilde{s}_1 = -1$$

$$\alpha_1 \tilde{s}_2 \tilde{s}_1 + \alpha_2 \tilde{s}_2 \tilde{s}_2 + \alpha_3 \tilde{s}_2 \tilde{s}_3 = +1$$

$$\alpha_1 \tilde{s}_3 \tilde{s}_1 + \alpha_2 \tilde{s}_3 \tilde{s}_2 + \alpha_3 \tilde{s}_3 \tilde{s}_3 = +1$$

$$\alpha_1 \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} + \alpha_2 \begin{pmatrix} 3 \\ 1 \\ 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} +$$

$$\alpha_3 \begin{pmatrix} 3 \\ -1 \\ 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} = -1$$

$$\alpha_1 \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} \begin{pmatrix} 3 \\ 1 \\ 1 \end{pmatrix} + \alpha_2 \begin{pmatrix} 3 \\ 1 \\ 1 \end{pmatrix} \begin{pmatrix} 3 \\ 1 \\ 1 \end{pmatrix} +$$

$$\alpha_3 \begin{pmatrix} 3 \\ -1 \\ 1 \end{pmatrix} \begin{pmatrix} 3 \\ 1 \\ 1 \end{pmatrix} = +1$$

$$\alpha_3 \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} \begin{pmatrix} 3 \\ -1 \\ 1 \end{pmatrix} + \alpha_2 \begin{pmatrix} 3 \\ 1 \\ 1 \end{pmatrix} \begin{pmatrix} 3 \\ -1 \\ 1 \end{pmatrix}$$

$$+ \alpha_3 \begin{pmatrix} 3 \\ -1 \\ 1 \end{pmatrix} \begin{pmatrix} 3 \\ -1 \\ 1 \end{pmatrix} = +1$$

$$\alpha_1(1+0+1) + \alpha_2(3+0+1) + \alpha_3(3+0+1) = -1$$

$$\alpha_1(3+0+1) + \alpha_2(9+1+1) + \alpha_3(9-1+1) = 1$$

$$\alpha_1(3+0+1) + \alpha_2(9-1+1) + \alpha_3(9+1+1) = 1$$

$$2\alpha_1 + 4\alpha_2 + 4\alpha_3 = -1$$

$$9\alpha_1 + 11\alpha_2 + 9\alpha_3 = 1$$

$$4\alpha_1 + 9\alpha_2 + 11\alpha_3 = 1$$

$$\alpha_1 = -3.5 \quad \alpha_2 = 0.75 \quad \alpha_3 = 0.75$$

$$\tilde{w} = \sum_i \alpha_i \tilde{s}_i$$

$$= -3.5\begin{pmatrix}1\\0\\1\end{pmatrix} + 0.75\begin{pmatrix}3\\1\\1\end{pmatrix} + 0.75\begin{pmatrix}3\\-1\\1\end{pmatrix}$$

$$= \begin{bmatrix} 1 \\ 0 \\ -2 \end{bmatrix} \rightarrow \tilde{w}$$
← bias

$$y = wx + b$$

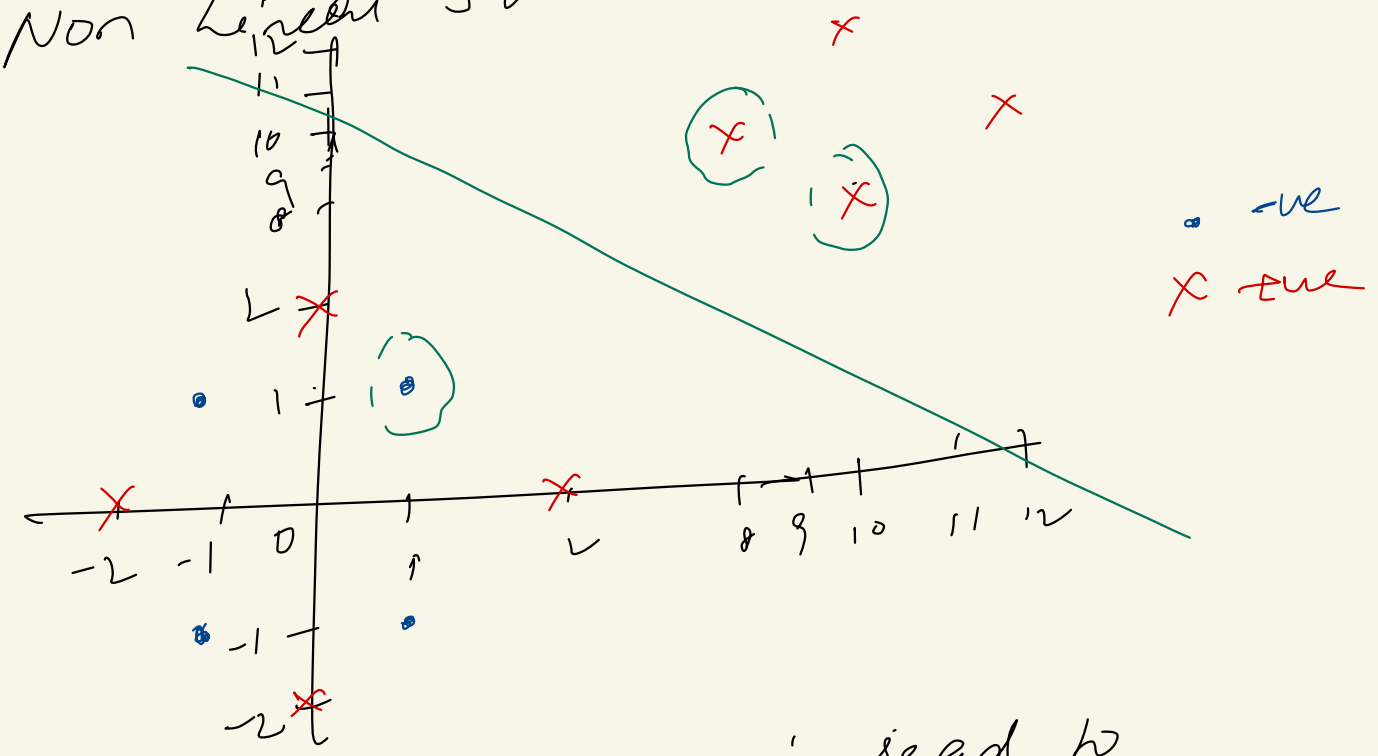
$$w = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad b = -2$$

Q/ -ve class points -1 $\begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ -1 \end{pmatrix}, \begin{pmatrix} 2 \\ 1 \end{pmatrix} \begin{pmatrix} 2 \\ -1 \end{pmatrix}$

+ve class points $+1$ $\begin{pmatrix} 4 \\ 0 \end{pmatrix}, \begin{pmatrix} 5 \\ 1 \end{pmatrix}, \begin{pmatrix} 5 \\ -1 \end{pmatrix}, \begin{pmatrix} 6 \\ 0 \end{pmatrix}$

find the generating function

Non Linear SVM



$\phi \Rightarrow$ Mapping function is reqd to transform the data to a new feature space where a seperating hyperplane can be found.

$$\phi\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{cases} 6 - x_1 + (x_1 - x_2)^2 \\ 6 - x_2 + (x_1 - x_2)^2 \end{cases} \quad \text{if } \sqrt{x_1^2 + x_2^2} > 2$$

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \quad \text{otherwise}$$

→ Blue class vectors are: $\begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} -1 \\ 1 \end{pmatrix}, \begin{pmatrix} -1 \\ -1 \end{pmatrix}, \begin{pmatrix} 1 \\ -1 \end{pmatrix}$ no change since

$\sqrt{x_1^2 + x_2^2} < 2$ # vectors

→ Red class vectors $\begin{pmatrix}2\\0\end{pmatrix}, \begin{pmatrix}0\\2\end{pmatrix}, \begin{pmatrix}-2\\0\end{pmatrix}, \begin{pmatrix}0\\-2\end{pmatrix}$

$$\phi\begin{pmatrix}x_1\\x_2\end{pmatrix} = \phi\begin{pmatrix}2\\0\end{pmatrix} = \begin{pmatrix}6-2+(2-0)^2\\6-0+(2-0)^2\end{pmatrix}$$

$$= \begin{pmatrix}8\\10\end{pmatrix}$$

$$\phi\begin{pmatrix}x_1\\x_2\end{pmatrix} = \phi\begin{pmatrix}0\\2\end{pmatrix} = \begin{pmatrix}10\\8\end{pmatrix}$$

$$\phi\begin{pmatrix}x_1\\x_2\end{pmatrix} = \phi\begin{pmatrix}-2\\0\end{pmatrix} = \begin{pmatrix}12\\10\end{pmatrix}$$

$$\phi\begin{pmatrix}x_1\\x_2\end{pmatrix} = \phi\begin{pmatrix}0\\-2\end{pmatrix} = \begin{pmatrix}10\\12\end{pmatrix}$$

$$S_1 = \begin{pmatrix}8\\10\end{pmatrix} \quad S_2\begin{pmatrix}10\\8\end{pmatrix} \quad S_3\begin{pmatrix}1\\1\end{pmatrix}$$

$$\tilde{S}_1 = \begin{pmatrix}8\\10\\1\end{pmatrix} \quad \tilde{S}_2 = \begin{pmatrix}10\\8\\1\end{pmatrix} \quad \tilde{S}_3 = \begin{pmatrix}1\\1\\1\end{pmatrix}$$

$$\alpha_1 \tilde{S}_1 \tilde{S}_1 + \alpha_2 \tilde{S}_1 \tilde{S}_2 + \alpha_3 \tilde{S}_1 \tilde{S}_3 = +1$$

$$\alpha_1 \tilde{S}_2 \tilde{S}_1 + \alpha_2 \tilde{S}_2 \tilde{S}_2 + \alpha_3 \tilde{S}_2 \tilde{S}_3 = +1$$

$$\alpha_1 \tilde{S}_3 \tilde{S}_1 + \alpha_2 \tilde{S}_3 \tilde{S}_2 + \alpha_3 \tilde{S}_3 \tilde{S}_3 = -1$$

$$165\alpha_1 + 161\alpha_2 + 19\alpha_3 = +1$$

$$164\alpha_1 + 165\alpha_2 + 19\alpha_3 = +1$$

$$19\alpha_1 + 19\alpha_2 + 3\alpha_3 = -1$$

$$\alpha_1 = \alpha_2 = 0.8898 \quad \alpha_3 = -1.4218$$

$$\tilde{w} = \sum_i \alpha_i \tilde{s}_i$$

$$\tilde{w} = \alpha_1 \begin{pmatrix} 8 \\ 10 \\ 1 \end{pmatrix} + \alpha_2 \begin{pmatrix} 10 \\ 8 \\ 1 \end{pmatrix} + \alpha_3 \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

$$= \begin{pmatrix} 0.1243 \\ 0.1243 \\ -1.2501 \end{pmatrix}$$

$$y = wx + b$$

$$w = \begin{pmatrix} 0.1243/0.1243 \\ 0.1243/0.1243 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

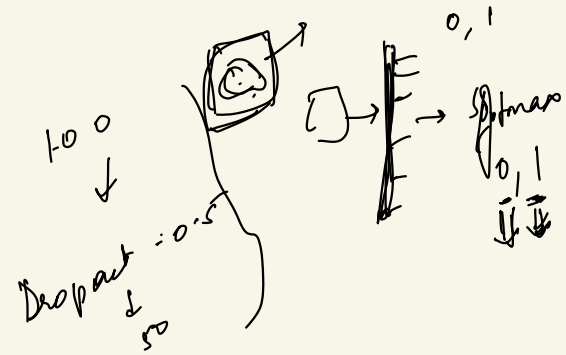
$$b = -1.2501/0.1243$$

$$= -10.057$$

Test point $x_1, x_2 = -1, 2$ belongs to Red point class true

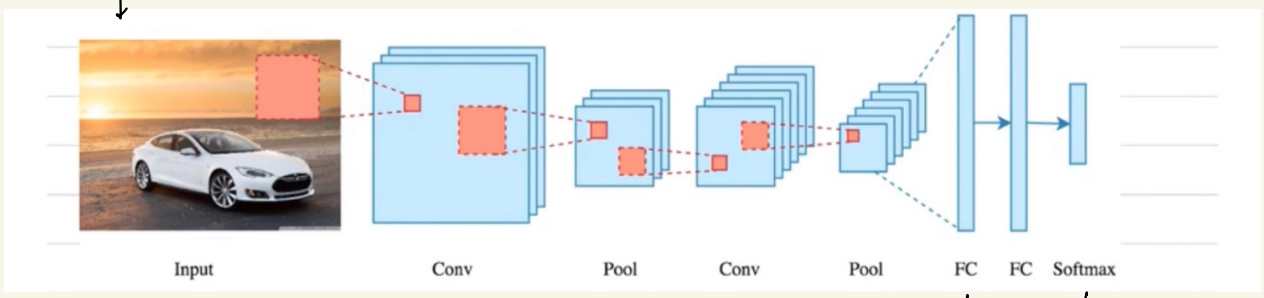
$$\phi\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 16 \\ 13 \end{pmatrix}$$

$$w\,\phi(x) = \begin{pmatrix} 1 \\ 1 \end{pmatrix}\begin{pmatrix} 16 \\ 13 \end{pmatrix} = 29 > 10$$



Introduction to Deep learning (CNNs)

Convolutional Neural networks



Correlation
Convolution
100°

1D input

$h(x)$ → f80°

CNNs over ANNs – i) No context learnt (Neighborhood of pixels) 3×3 4×4 2×2 array
ii) More parameters to learn in ANNs

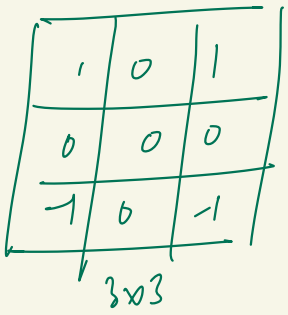
no flipping ↓ Flip kernel 48°

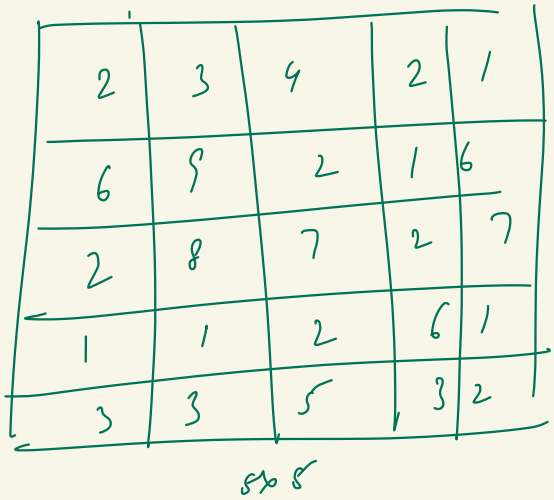
Correlation / Convolution

In case of CNNs → it is actually correlation only

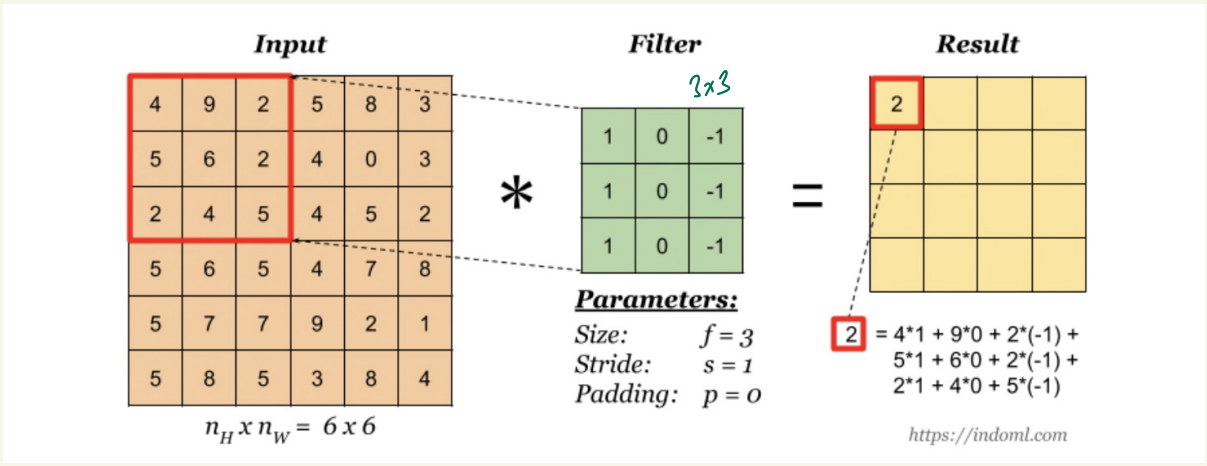
Conv (filter size, stride, padding) → pooling

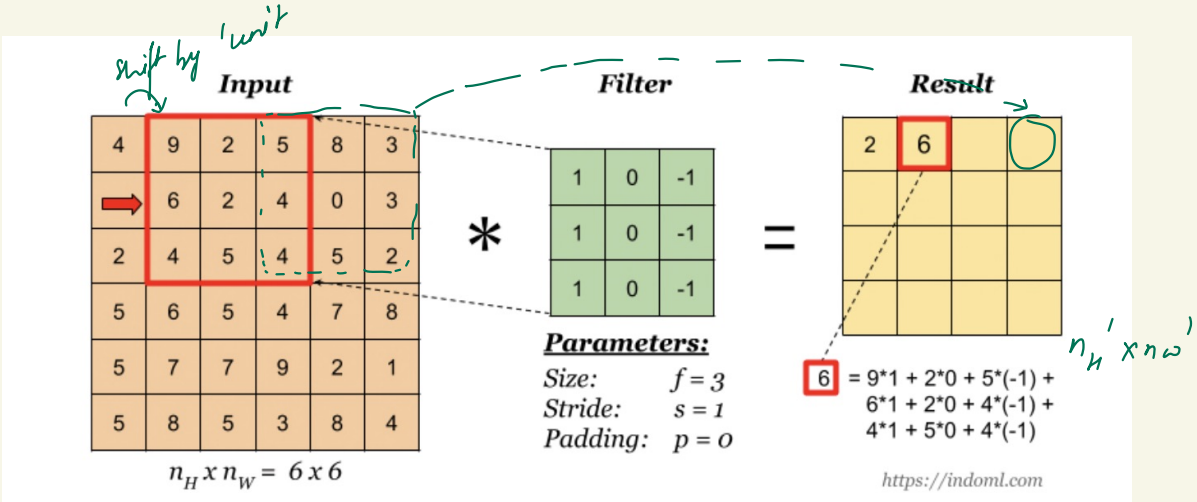
① Filter size





② Convolution





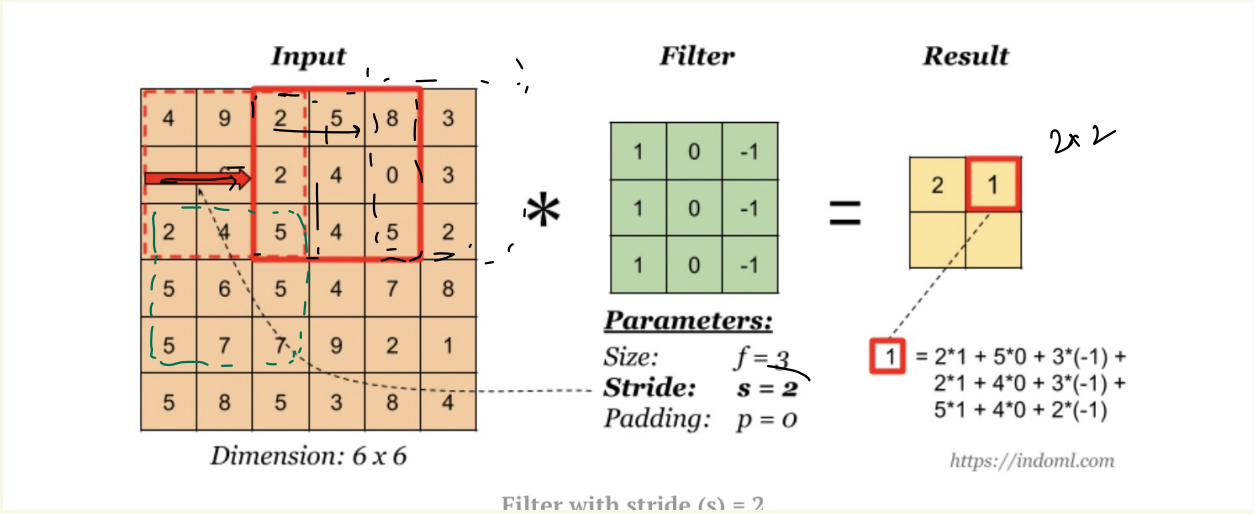
$$n_H' = n_H - f + 1 = 6 - 3 + 1 = 4$$

Stride

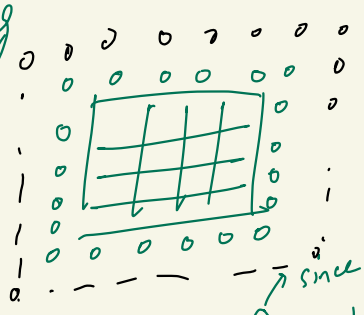
Let $s=1$, $p=0$

$$n_H' = n_H - f + 1$$

$$n_H' = \left\lfloor \frac{n_H + 2p - f}{s} + 1 \right\rfloor$$



Padding



$4\times4 \rightarrow 6\times6 \rightarrow 8\times8$

$p=1$ $\quad p=2$

since $p=1$ increase image size by 2 units

$$n_H' = n - f + 2p + 1$$

Convolution reduces size

Is padding 'valid' || 'same' $\rightarrow p = \frac{f-1}{2}$

'valid' → reduces size of image

'same' → keeping original size of image same

$= \frac{3-1}{2} = 1$

$6\times6 \rightarrow 6\times6$

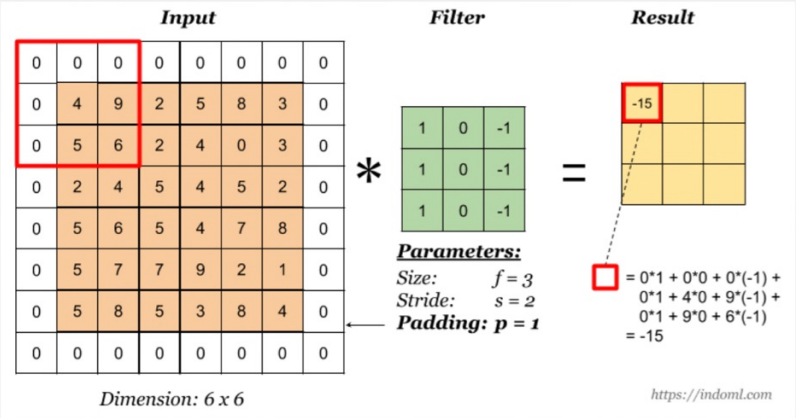
$$n_H' = \left\lfloor \frac{n_H + 2p - f}{s} + 1 \right\rfloor = n_H$$

$$\Rightarrow p = \frac{f-1}{2}$$

Padding

Padding has the following benefits:

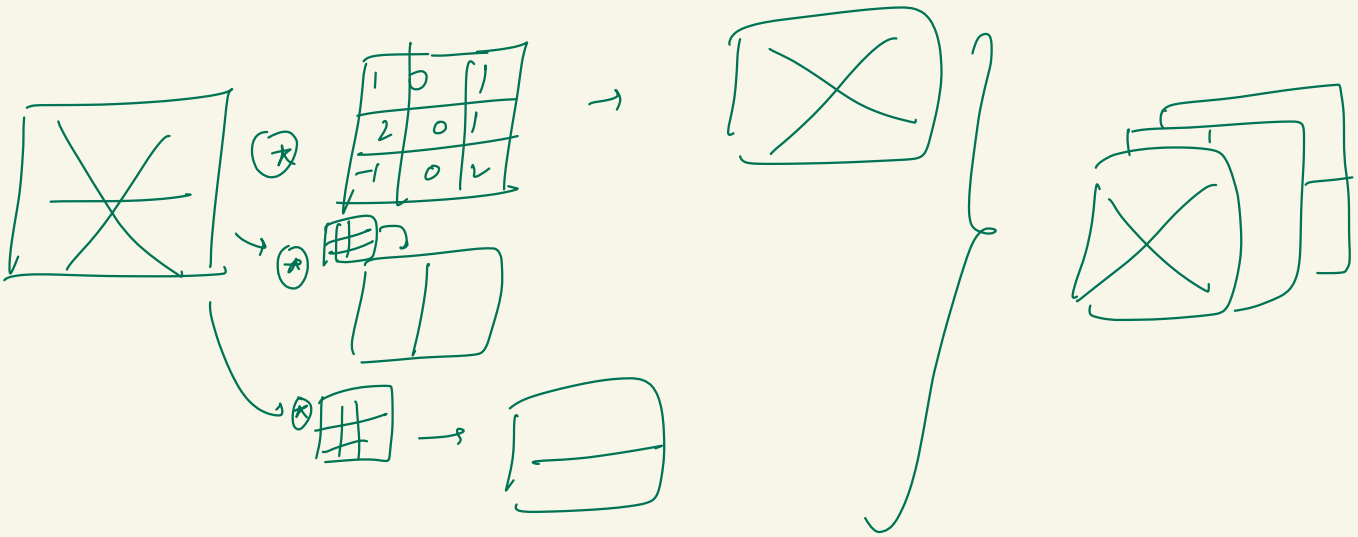
1. It allows us to use a CONV layer without necessarily shrinking the height and width of the volumes. This is important for building deeper networks, since otherwise the height/width would shrink as we go to deeper layers.
2. It helps us keep more of the information at the border of an image. Without padding, very few values at the next layer would be affected by pixels as the edges of an image.

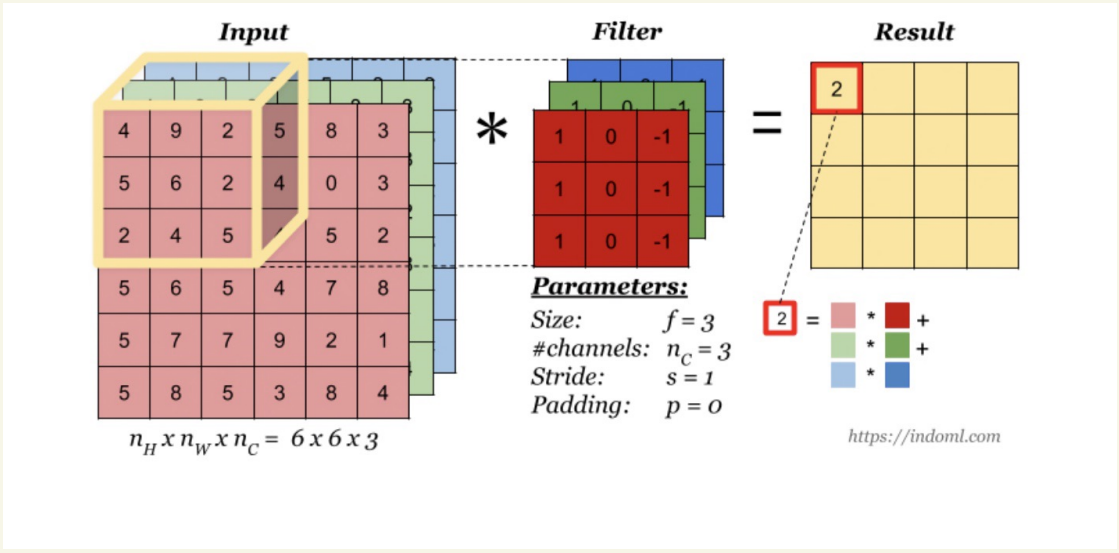


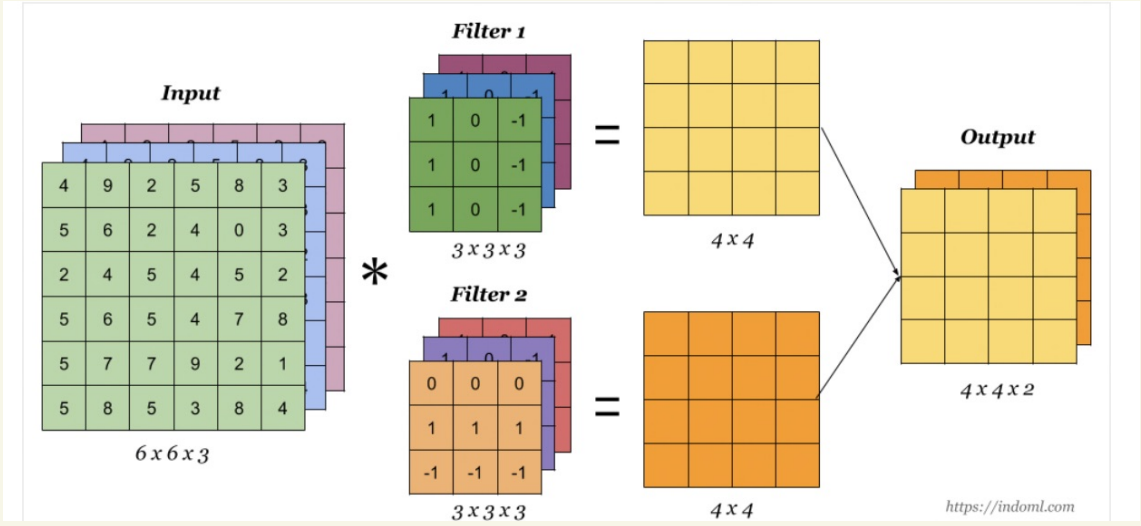
Notice the the dimension of the result has changed due to padding. See the following section on how to calculate output dimension.

Some padding terminologies:

- "**valid**" padding: no padding
- "**same**" padding: padding so that the output dimension is the same as the input

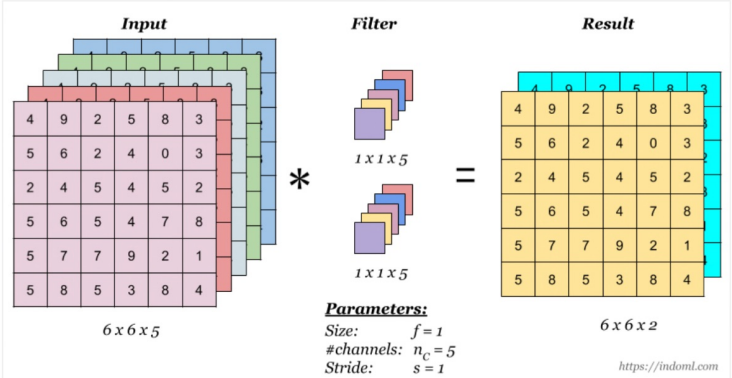


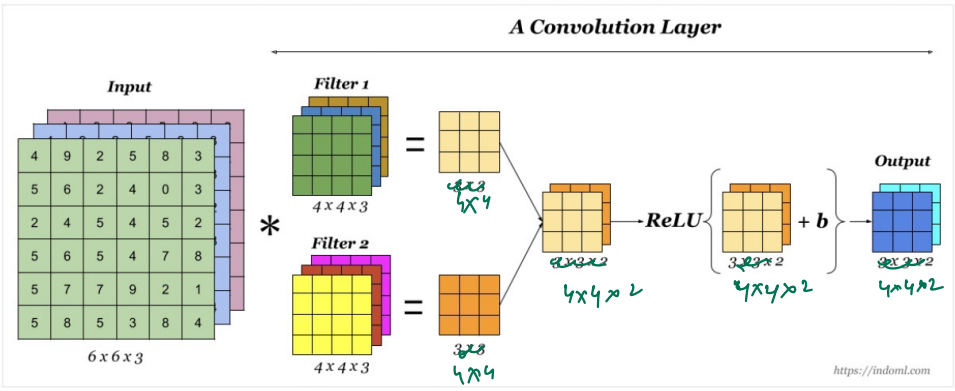




1 x 1 Convolution

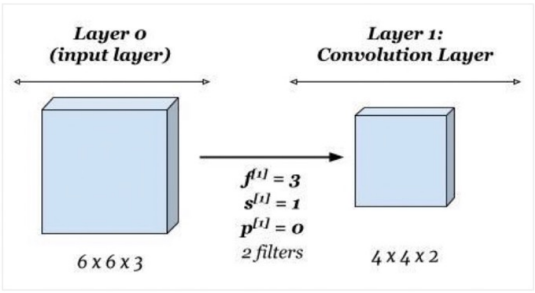
This is convolution with 1 x 1 filter. The effect is to flatten or "merge" channels together, which can save computations later in the network:





Shorthand Representation

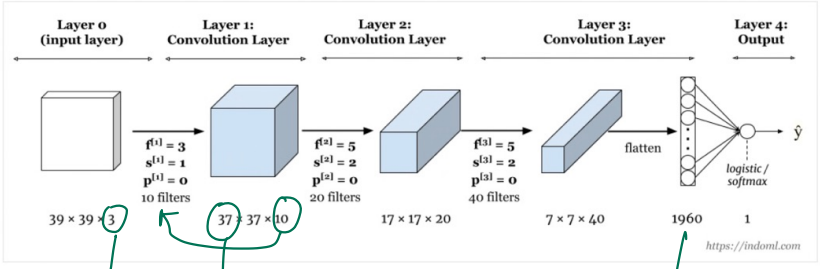
This simpler representation will be used from now on to represent one convolutional layer:



~

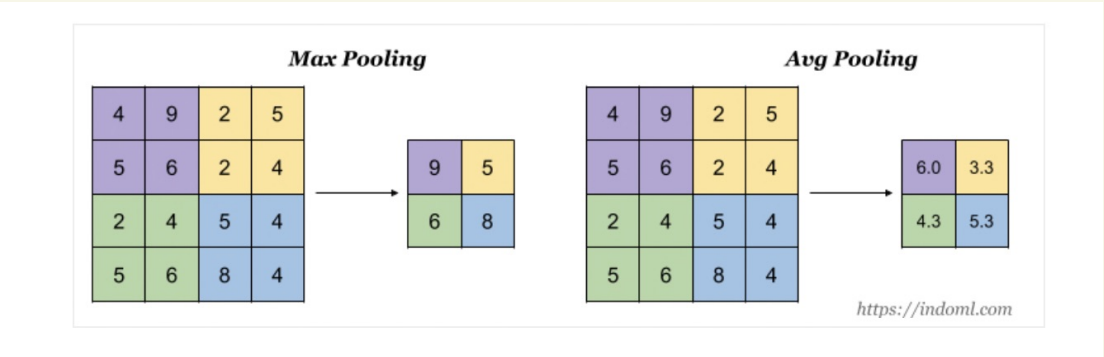
Sample Complete Network

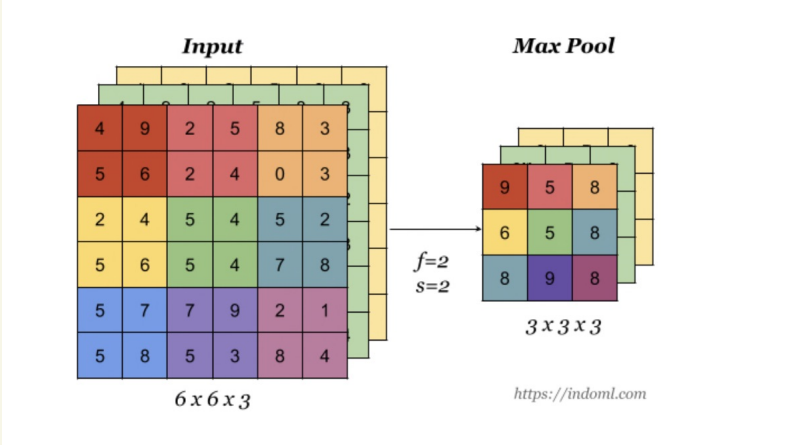
This is a sample network with three convolution layers. At the end of the network, the output of the convolution layer is flattened and is connected to a logistic regression or a softmax output layer.



Pooling → Avg
→ Max (more used)

Hy peparameters: size (f), stride(s), type (max/avg)

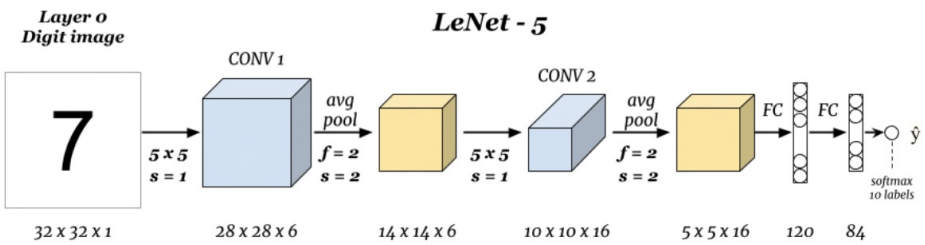




Well Known Architectures

Classic Network: LeNet – 5

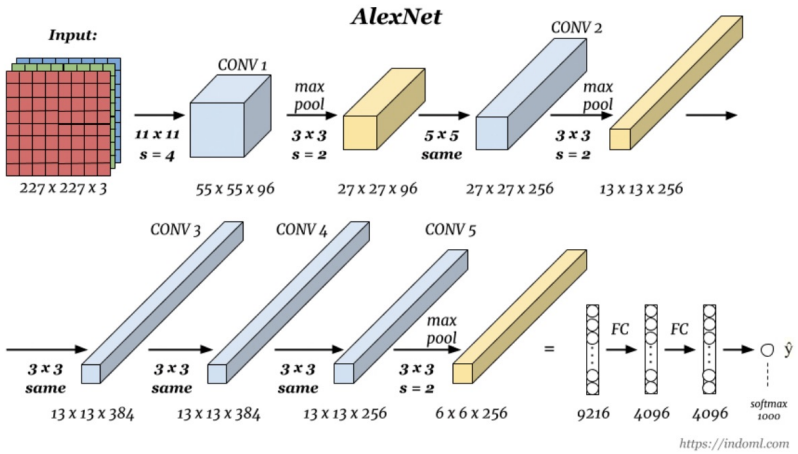
One example of classic networks is LeNet-5, from *Gradient-Based Learning Applied to Document Recognition* paper by Y. Lecun, L. Bottou, Y. Bengio and P. Haffner (1998):



- Number of parameters: ~ 60 thousands.

Classic Network: AlexNet

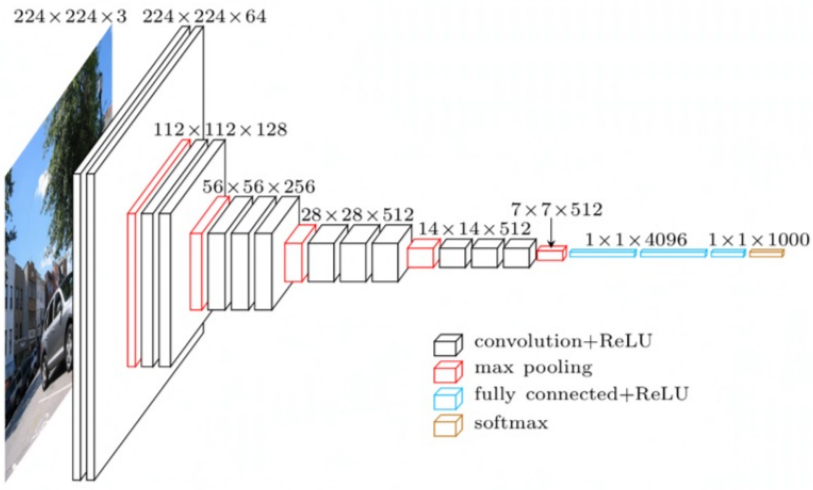
AlexNet is another classic CNN architecture from *ImageNet Classification with Deep Convolutional Neural Networks* paper by Alex Krizhevsky, Geoffrey Hinton, and Ilya Sutskever (2012).



- Number of parameters: ~ 60 millions

Classic Network: VGG-16

VGG-16 from *Very Deep Convolutional Networks for Large-Scale Image Recognition* paper by Karen Simonyan and Andrew Zisserman (2014). The number 16 refers to the fact that the network has 16 trainable layers (i.e. layers that have weights).

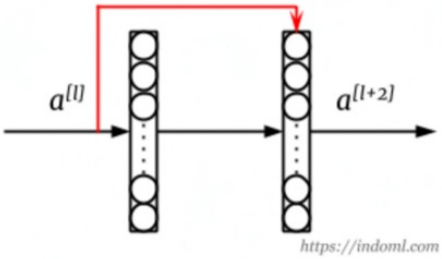


(image from **blog.heuritech.com**)

- Number of parameters: ~ 138 millions.
- The strength is in the simplicity: the dimension is halved and the depth is increased on every step (or stack of layers)

ResNet

The problem with deeper neural networks are they are harder to train and once the number of layers reach certain number, the training error starts to raise again. Deep networks are also harder to train due to exploding and vanishing gradients problem. ResNet (Residual Network), proposed by He at all in *Deep Residual Learning for Image Recognition paper* (2015), solves these problems by implementing skip connection where output from one layer is fed to layer deeper in the network:



In the image above, the skip connection is depicted by the red line. The activation $a^{[l+2]}$ is then calculated as:

$$z^{[l+2]} = W^{[l+2]}\, a^{[l+1]} + b^{[l+2]}$$

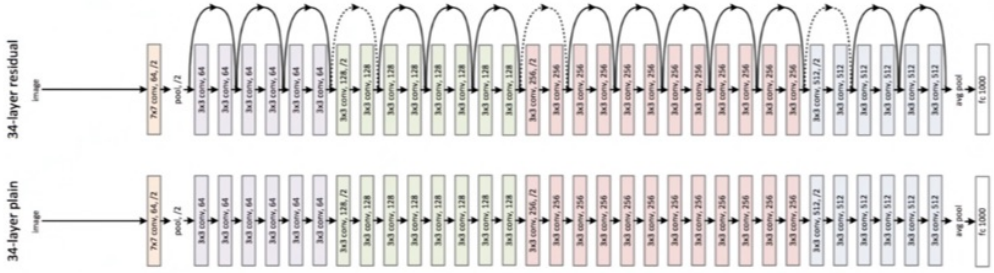
$$a^{[l+2]} = g^{[l+2]}(z^{[l+2]} + a^{[l]})$$

The advantages of ResNets are:

- performance doesn't degrade with very deep network
- cheaper to compute
- ability to train very very deep network

ResNet works because:

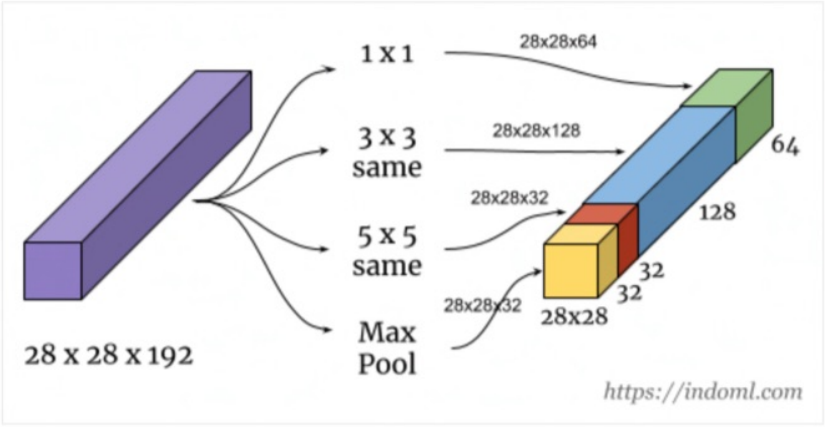
- identify function is easy for residual block to learn
- using a skip-connection helps the gradient to back-propagate and thus helps you to train deeper networks



Comparison of 34 layers ResNet with plain network (image from **euler.stat.yale.edu**)

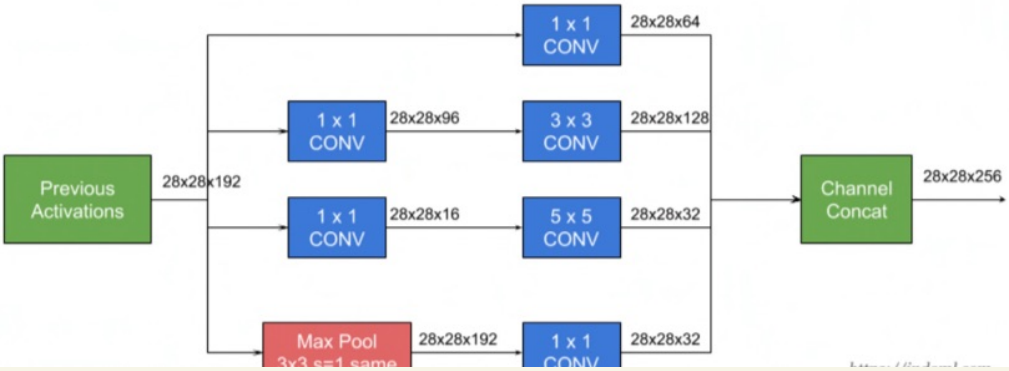
Inception

The motivation of the inception network is, rather than requiring us to pick the filter size manually, let the network decide what is best to put in a layer. We give it choices and hopefully it will pick up what is best to use in that layer:

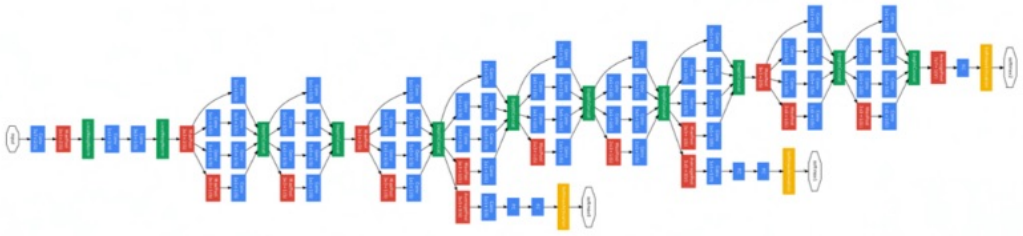


The problem with the above network is computation cost (e.g. for the 5 x 5 filter only, the computation cost is (28 x 28 x 32) x (5 x 5 x 192) = ~ 120 millions).

Using 1 x 1 convolution will reduce the computation to about 1/10 of that. With this idea, an inception module will look like this:



Below is an inception network called **GoogLeNet**, described in *Going Deeper with Convolutions paper* by Szegedy et all (2014), which has 9 inception modules:



GoogLeNet architecture (image **source**)

Data Augmentation

- Mirroring
- Random cropping
- Less common (perhaps due to their complexity):
  - Rotation
  - Shearing
  - Local warping
- Color shifting