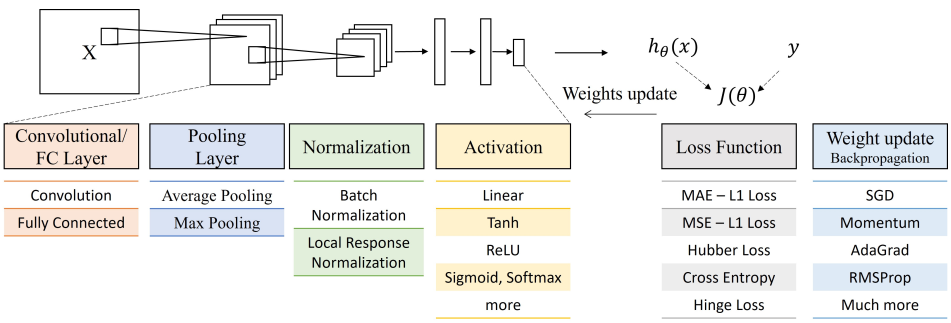
Convolutional Neural Network - Part 1

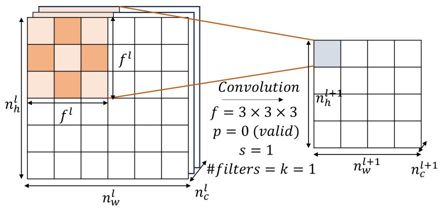
A Convolutional Neural Network (CNN) is a deep learning architecture primarily used for computer vision tasks. It consists of convolutional layer, pooling layer, fully connected layer and normalization layer.



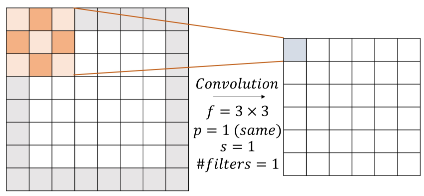
Convolutional Layer:

Convolutional filter with same channels as input, slides through the input feature map. It computes dot product between filter weights and input then bias term is added. It is used to extract the features of the image.

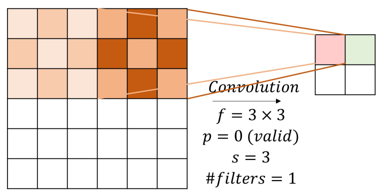
Notations:	**Padding(p):**	**Stride(s):**
Input size $= n = n_h^l \times n_w^l \times n_c^l$	Valid: No padding, output shrinks, drops last convolution if no match	The number of pixels by which the filter window slides.
Filter size $= f = f^l \times f^l \times n_c^l$	$n_h^{l+1} \times n_w^{l+1} = \lfloor \frac{n-f+1}{s} \rfloor \times \lfloor \frac{n-f+1}{s} \rfloor$	denoted by s
# Filters $= k = n_c^{l+1}$	Same: padding, output size same, also called half padding	used for convolutional and pooling layer.
Output size $= n_h^{l+1} \times n_w^{l+1} \times n_c^{l+1}$	$n_h^{l+1} \times n_w^{l+1} = \lfloor \frac{n+2p-f+1}{s} \rfloor \times \lfloor \frac{n+2p-f+1}{s} \rfloor$	



(a) Valid padding example



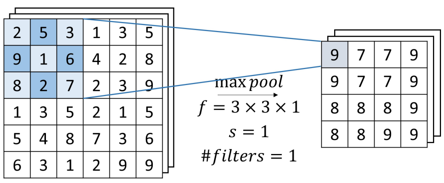
(b) Same padding example



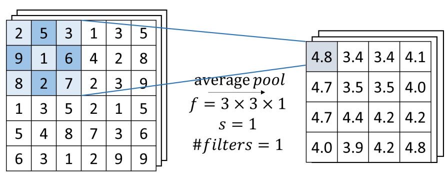
(c) Stride example

Pooling Layer:

It reduces the spatial dimensions of the input using max or average operation. This filter is applied in each channel, so same # channels between input and output. No weight parameters. No padding hyperparameter.



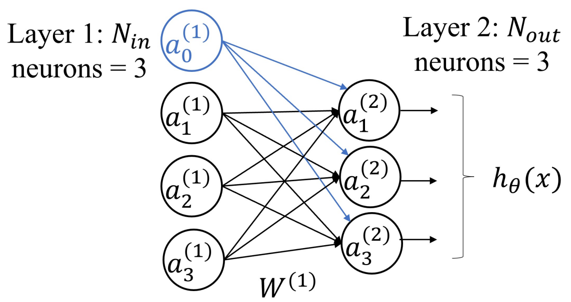
(a) Max pooling example



(b) Average pooling example

Fully Connected Layer:

Input is flattened neuron layer. It is multiplied with weight matrix and bias vector is added. Each output neuron has one bias parameter.



$$W^{(1)} = (N_{out}) \times (N_{in} + 1) = 3 \times 4$$

$$a_1^{(2)} = g(\theta_{10}^{(1)} a_0^{(1)} + \theta_{11}^{(1)} a_1^{(1)} + \theta_{12}^{(1)} a_2^{(1)} + \theta_{13}^{(1)} a_3^{(1)})$$

$$a_2^{(2)} = g(\theta_{20}^{(1)} a_0^{(1)} + \theta_{21}^{(1)} a_1^{(1)} + \theta_{22}^{(1)} a_2^{(1)} + \theta_{23}^{(1)} a_3^{(1)})$$

$$a_3^{(2)} = g(\theta_{30}^{(1)} a_0^{(1)} + \theta_{31}^{(1)} a_1^{(1)} + \theta_{32}^{(1)} a_2^{(1)} + \theta_{33}^{(1)} a_3^{(1)})$$