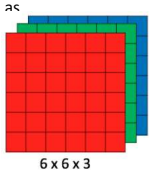
# Convolution Neural Network

## CNN

- Technically, deep learning CNN models to train and test, each input image will pass it through a series of convolution layers with
  - filters (Kernals),
  - Pooling,
  - fully connected layers (FC)
  - apply Softmax function to classify an object with probabilistic values between 0 and 1.
  - The below figure is a complete flow of CNN to process an input image and classifies the objects based on values.

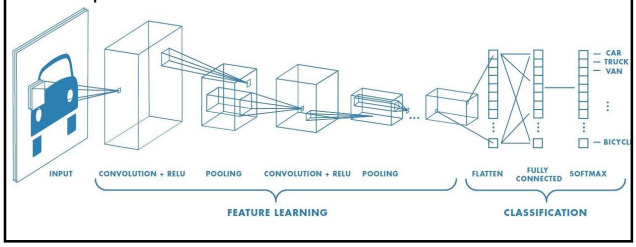
## CNN

- In neural networks, Convolutional neural network (ConvNets or CNNs) is one of the main which are mainly used in the areas such as
  - images recognition
  - images classifications
  - Objects detections
  - recognition faces etc.,
- CNN image classifications takes an
  - input image
  - process it
  - classify it under certain categories (Eg., Dog, Cat, Tiger, Lion).
- Computers sees an input image as array of pixels and it depends on the image resolution.
- Image resolution: **h** x **w** x **d**( h = Height, w = Width, d = Dimension ). Eg., An image of 6 x 6 x 3 array of matrix of RGB (3 refers to RGB values) and an image of 4 x 4 x 1 array of matrix of grayscale image.

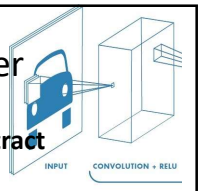



## CNN

- filters (Kernals),
- Pooling,
- fully connected layers (FC)
- apply Softmax function to classify an object with probabilistic values between 0 and 1.



## Convolution Layer




- Convolution is the first layer **to extract features from an input image**.
- Convolution preserves the relationship between pixels by learning image features using small squares of input data.
- It is a mathematical operation that takes two inputs such as image matrix and a filter or kernel.

## Convolution Layer

- Consider a
  - **5 x 5** whose image pixel values 0, 1
  - **filter matrix 3 x 3**

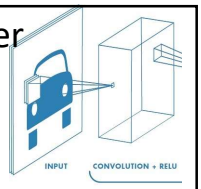

- Then the convolution of 5 x 5 image matrix multiplies with 3 x 3 filter matrix which is called "Feature Map" as output shown

| | | |
|---|---|---|
| 4 | 3 | 4 |
| 2 | 4 | 3 |
| 2 | 3 | 4 |

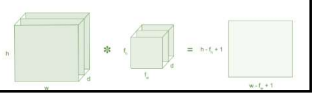
Convolved Feature

## Convolution Layer

- Convolution is the first layer **to extract features from an input image**.
- Convolution preserves the relationship between pixels by learning image features using small squares of input data.
- It is a mathematical operation that takes two inputs such as image matrix and a filter or kernel.

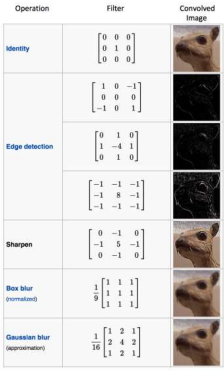



- An image matrix (volume) of dimension **(h x w x d)**
- A filter **($f_h$ x $f_w$ x d)**
- Outputs a volume dimension **(h - $f_h$ + 1) x (w - $f_w$ + 1) x 1**

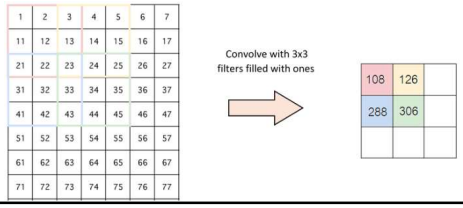


## Types of Filters

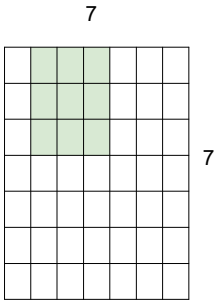
- Convolution of an image with different filters can perform operations such as
  - Edge detection
  - Blur
  - Sharpen
- example shows various convolution image after applying different types of filters (Kernels).



- Strides
- Stride is the number of pixels shifts over the input matrix. When the stride is 1 then we move the filters to 1 pixel at a time. When the stride is 2 then we move the filters to 2 pixels at a time and so on. The below figure shows convolution would work with a stride of 2.

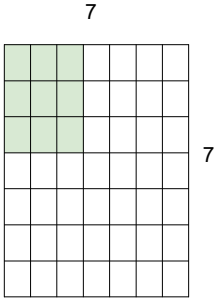



A closer look at spatial dimensions:



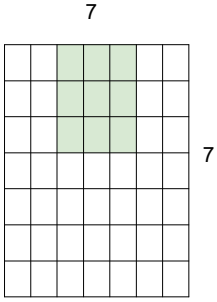

7x7 input (spatially) assume 3x3 filter

A closer look at spatial dimensions:



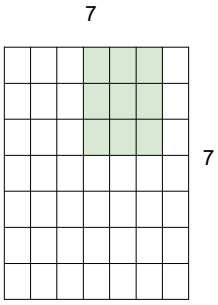

7x7 input (spatially) assume 3x3 filter

A closer look at spatial dimensions:



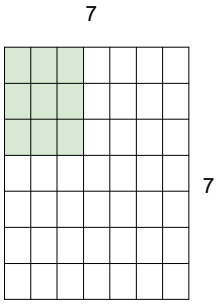

7x7 input (spatially) assume 3x3 filter

A closer look at spatial dimensions:



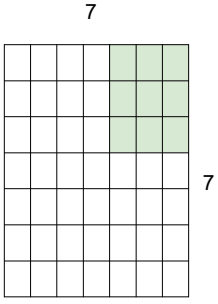

7x7 input (spatially)
assume 3x3 filter

A closer look at spatial dimensions:




7x7 input (spatially)
assume 3x3 filter
applied **with stride 2**

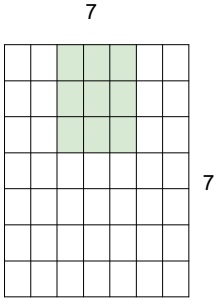
A closer look at spatial dimensions:




7x7 input (spatially)
assume 3x3 filter

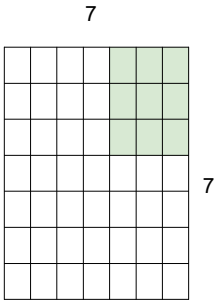
**=> 5x5 output**

A closer look at spatial dimensions:



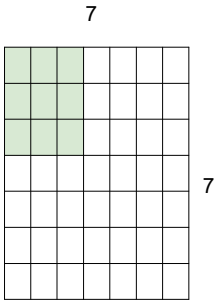

7x7 input (spatially)
assume 3x3 filter
applied **with stride 2**

A closer look at spatial dimensions:




7x7 input (spatially)
assume 3x3 filter
applied **with stride 2**
**=> 3x3 output!**

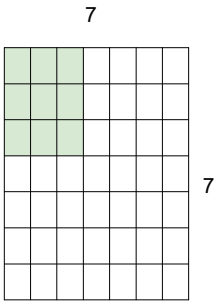
A closer look at spatial dimensions:



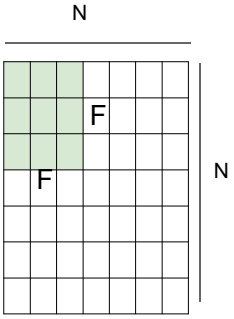

7x7 input (spatially)
assume 3x3 filter
applied **with stride 3?**

**doesn't fit!**
cannot apply 3x3 filter on
7x7 input with stride 3.

A closer look at spatial dimensions:




7x7 input (spatially)
assume 3x3 filter
applied **with stride 3?**



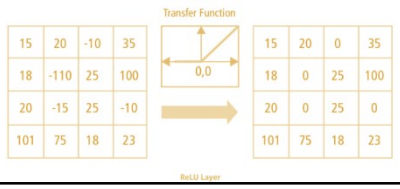

Output size:
**(N - F) / stride + 1**

e.g. N = 7, F = 3:
stride 1 => (7 - 3)/1 + 1 = 5
stride 2 => (7 - 3)/2 + 1 = 3
stride 3 => (7 - 3)/3 + 1 = 2.33 :\

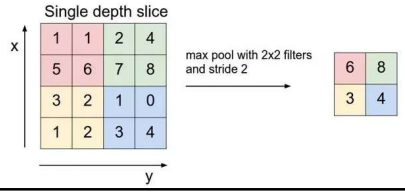
- **Padding**
- Sometimes filter does not fit perfectly fit the input image. We have two options:
  - Pad the picture with zeros (zero-padding) so that it fits
  - Drop the part of the image where the filter did not fit. This is called **valid padding** which keeps only valid part of the image.
- **Non Linearity (ReLU)**
- ReLU stands for Rectified Linear Unit for a non-linear operation. The output is ƒ(x) = max(0,x).
- Why ReLU is important : ReLU's purpose is to introduce non-linearity in our ConvNet. Since, the real world data would want **our ConvNet to learn would be non-negative linear values**.

- Pooling Layer
- Pooling layers section would reduce the number of parameters when the images are too large. **Spatial pooling** also called subsampling or downsampling which reduces the dimensionality of each map but retains important information. Spatial pooling can be of different types:
  - Max Pooling
  - Average Pooling
  - Sum Pooling
- Max pooling takes the largest element from the rectified feature map. Taking the largest element could also take the average pooling. Sum of all elements in the feature map call as sum pooling.

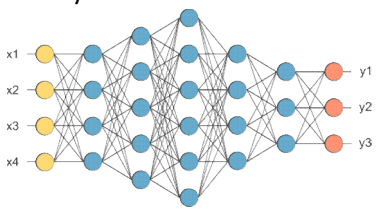
- **Non Linearity (ReLU)**
- ƒ(x) = max(0,x).
- Since, the real world data would want **our ConvNet to learn would be non-negative linear values**
- There are other non linear functions such as **tanh** or **sigmoid** that can also be used instead of ReLU.
- Most of the data scientists use ReLU since performance wise ReLU is better than the other two.




- Max pooling takes the largest element from the rectified feature map. Taking the largest element could also take the average pooling. Sum of all elements in the feature map call as sum pooling.



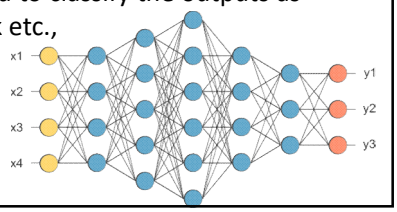
- Fully Connected Layer
- The layer we call as FC layer, we flattened our matrix into vector and feed it into a fully connected layer like a neural network.



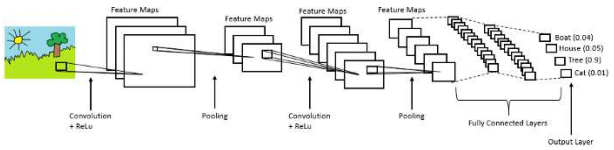

## Softmax

- The softmax function is a function that turns a vector of K real values into a vector of K real values that sum to 1.

- In diagram, the feature map matrix will be converted as vector (x1, x2, x3, …). With the fully connected layers, we combined these features together to create a model. Finally, we have an activation function such as softmax or sigmoid to classify the outputs as cat, dog, car, truck etc.,




## Complete architecture of CNN

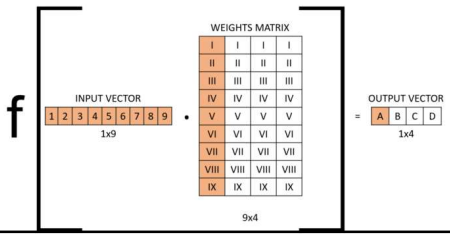


## Fully Connected Layer

- Neural networks are a set of dependent non-linear functions.
- Each individual function consists of a neuron (or a perceptron).
- **In fully connected layers**, *the neuron applies a linear transformation to the input vector through a weights matrix*.
- A non-linear transformation is then applied to the product through a non-linear activation function f.

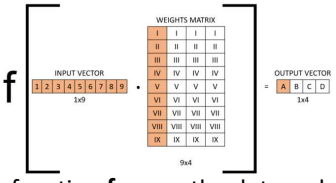
$$y_{jk}(x) = f\left(\sum_{i=1}^{n_H} w_{jk}x_i + w_{j0}\right)$$

- Example
- A layer in a FC Neural Network with an input size of 9 and an output size of 4, the operation can




- Here we are taking the dot product between the weights matrix W and the input vector x. The bias term (W0) can be added inside the non-linear function.

$$y_{jk}(x) = f\left(\sum_{i=1}^{n_H} w_{jk}x_i + w_{j0}\right)$$




- The activation function **f** wraps the dot product between the input of the layer and the weights matrix of that layer.
- Note that the columns in the weights matrix would all have different numbers and would be optimized as the model is trained.
- The input is a **1x9 vector**, the **weights** matrix is a **9x4 matrix**. By taking the dot product and applying the non-linear transformation with the activation function we get the **output vector (1x4)**.

## Fully Connected Layer