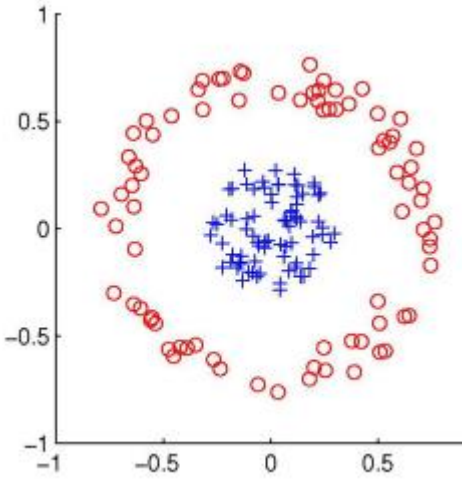
The different types of neural networks in deep learning, such as convolutional neural networks (CNN), recurrent neural networks (RNN), artificial neural networks (ANN), etc. are changing the way we interact with the world. These different types of neural networks are at the core of the deep learning revolution, powering applications like unmanned aerial vehicles, self-driving cars, speech recognition, etc
here are two key reasons why researchers and experts tend to prefer Deep Learning over Machine Learning:

- Decision Boundary
- Feature Engineering
  - ML cannot learn decision boundaries for nonlinear data like this one: Similarly, every Machine Learning algorithm is not capable of learning all the functions. This limits the problems these algorithms can solve that involve a complex relationship.

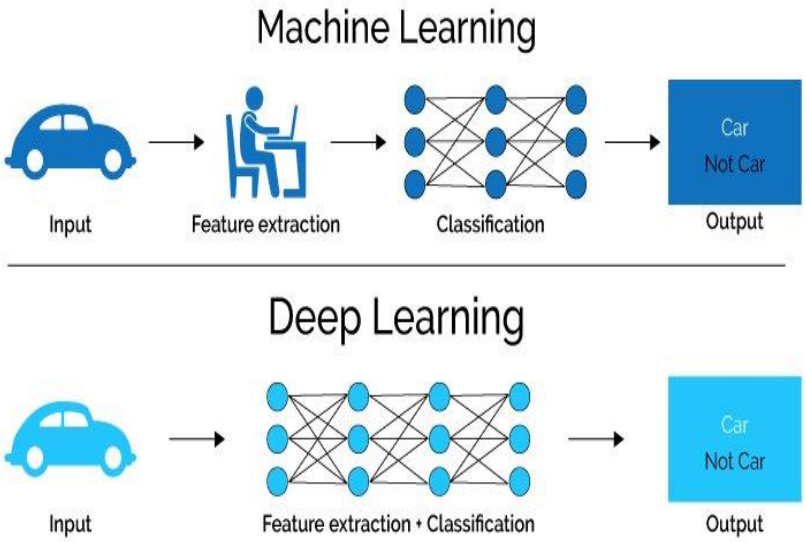


Machine Learning vs. Deep Learning: Feature Engineering
Feature engineering is a key step in the model building process. It is a two-step process:

1. **Feature extraction**
2. **Feature selection**

In feature extraction, we extract all the required features for our problem statement and in feature selection, we select the important features that improve the performance of our machine learning or deep learning model.

Consider an image classification problem. Extracting features manually from an image needs strong knowledge of the subject as well as the domain. It is an extremely time-consuming process. Using Deep Learning, we can automate the process of Feature Engineering!



Challenges with Artificial Neural Network (ANN)

- While solving an image classification problem using ANN, the first step is to convert a 2-dimensional image into a 1-dimensional vector prior to training the model. This has two drawbacks:
  - The number of trainable parameters increases drastically with an increase in the size of the image

  ANN loses the spatial features of an image. Spatial features refer to the arrangement of the pixels in an image.

- ANN cannot capture sequential information in the input data which is required for dealing with sequence data

CNN advantages: Very High accuracy in image recognition problems.

- Automatically detects the important features without any human supervision.
- Weight sharing.

Disadvantages:

- CNN do not encode the position and orientation of object.
- Lack of ability to be spatially invariant to the input data.
- Lots of training data is required.

RNN

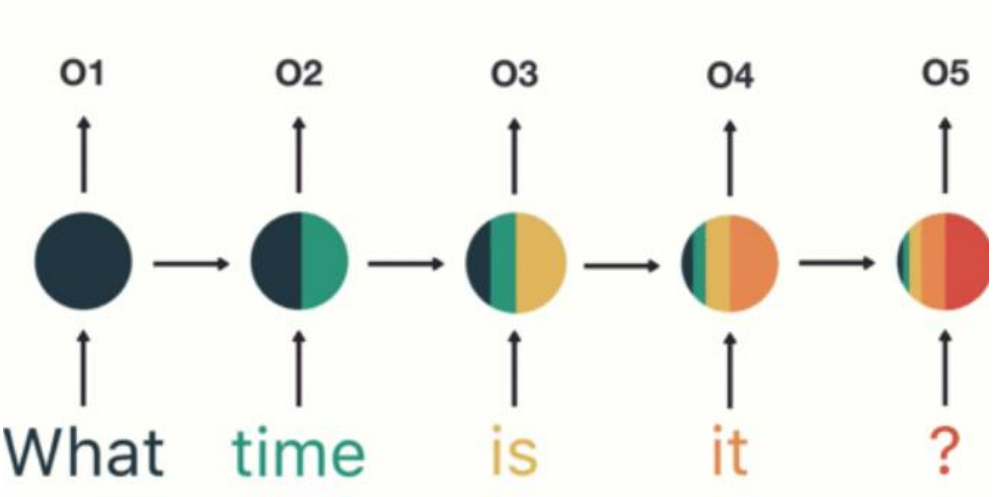
A looping constraint on the hidden layer of ANN turns to RNN. **As you can see here, RNN has a recurrent connection on the hidden state. This looping constraint ensures that sequential information is captured in the input data.**

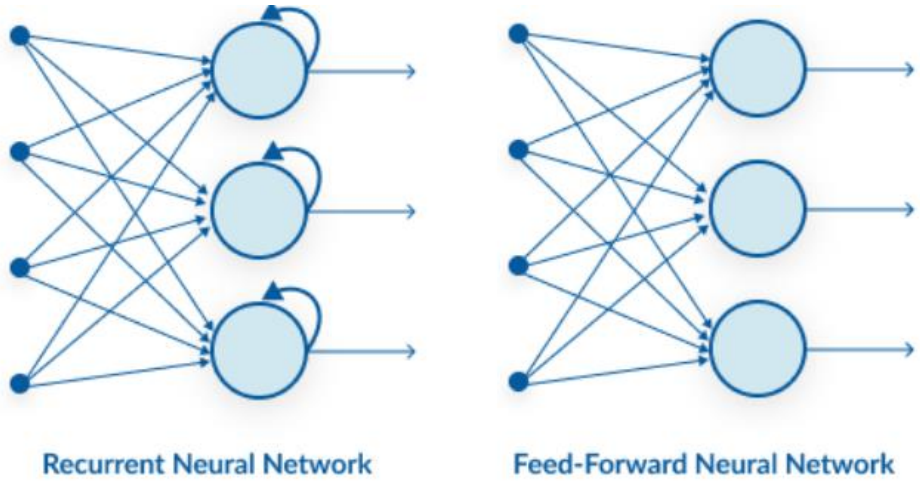
We can use recurrent neural networks to solve the problems related to:

- Time Series data
- Text data
- Audio data

Advantages of Recurrent Neural Network (RNN)

- RNN captures the sequential information present in the input data i.e. dependency between the words in the text while making predictions:





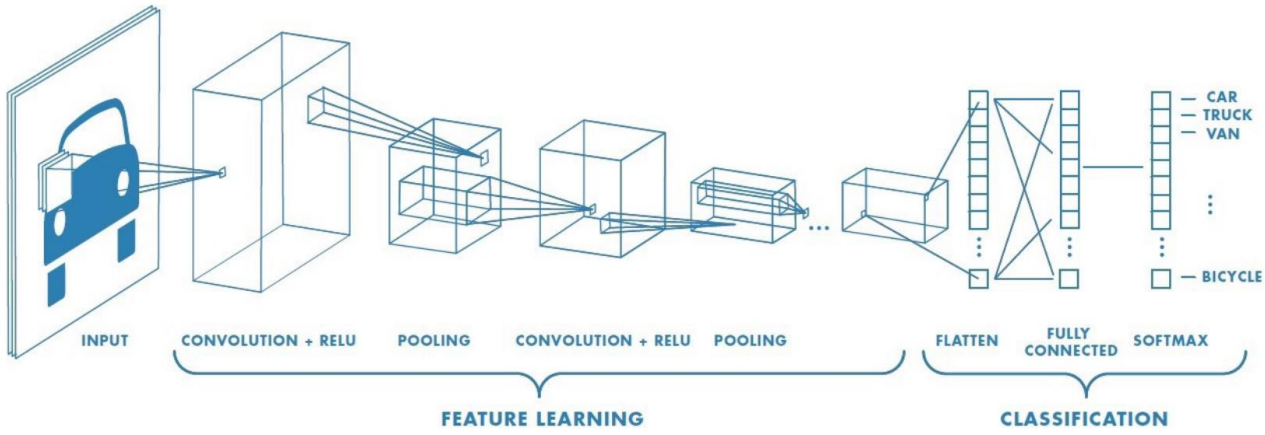
Advantages:

- An RNN remembers each and every information through time. It is useful in time series prediction only because of the feature to remember previous inputs as well. This is called Long Short Term Memory.
- Recurrent neural network are even used with convolutional layers to extend the effective pixel neighborhood.

Disadvantages:

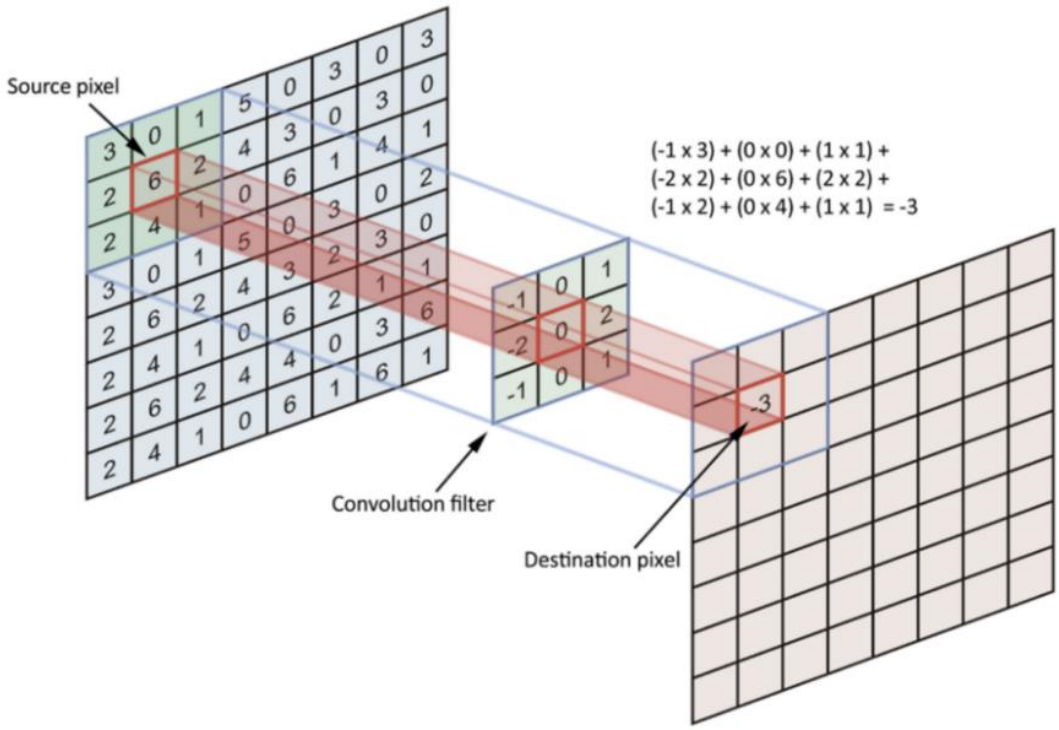
- Gradient vanishing and exploding problems.
- Training an RNN is a very difficult task.
- It cannot process very long sequences if using tanh or relu as an activation function.

Convolutional Neural Network. enable machines to view the world as humans do, perceive it in a similar manner and even use the knowledge for a multitude of tasks such as Image & Video recognition, Image Analysis & Classification, Media Recreation, Recommendation Systems, Natural Language Processing, etc. The advancements in Computer Vision with Deep Learning has been constructed and perfected with time, primarily over one particular algorithm — a **Convolutional Neural Network**.

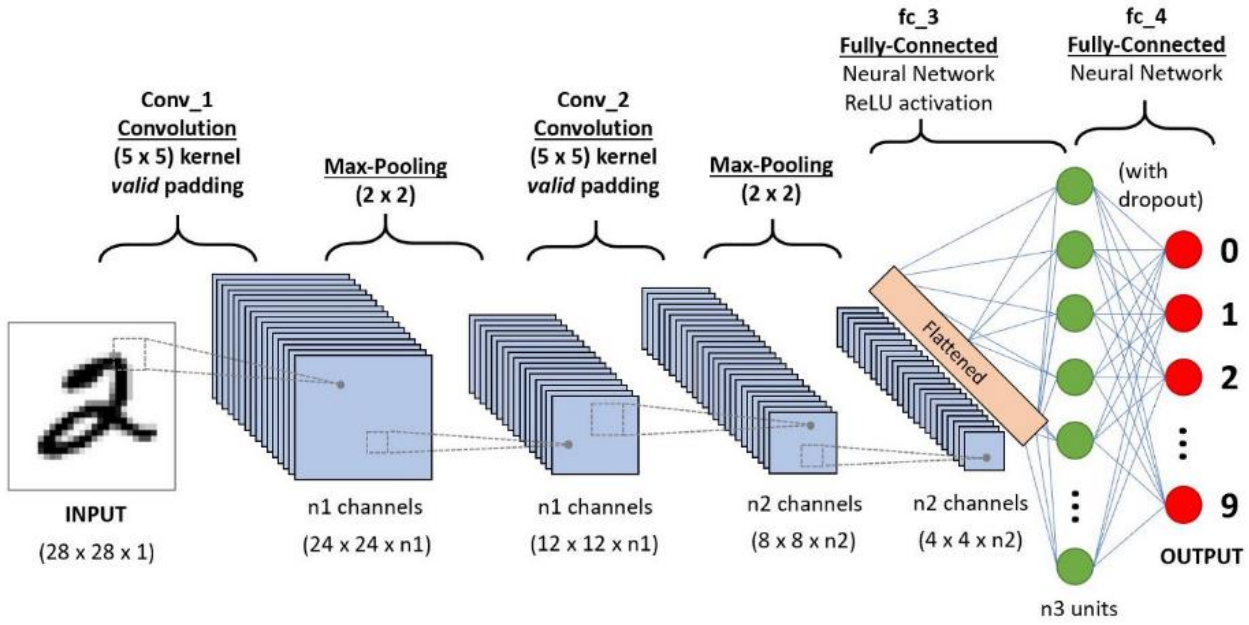


The convolution operation is typically denoted with an asterisk: $s(t) = (x * w)(t)$

In convolutional network terminology, the first argument (in this example, the function x) to the convolution is often referred to as the input , and the second argument (in this example, the function w) as the kernel . The output is sometimes referred to as the feature map



A CNN sequence to classify handwritten digits



A **Convolutional Neural Network (ConvNet/CNN)** is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, ConvNets have the ability to learn these filters/characteristics.

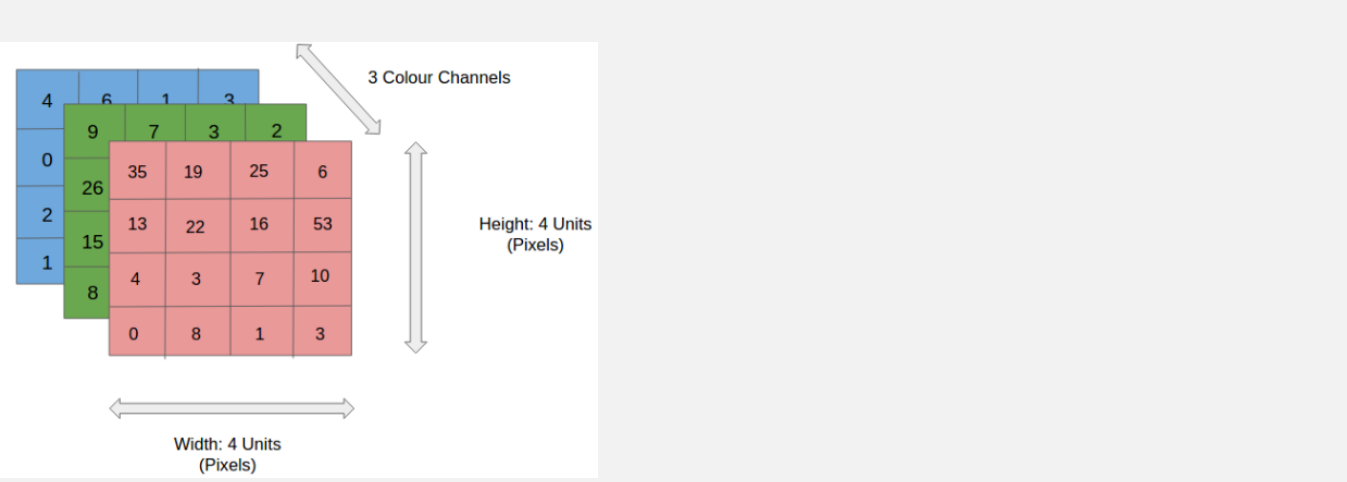
The CNN is a combination of two basic building blocks:

1. The Convolution Block — Consists of the Convolution Layer and the Pooling Layer. This layer forms the essential component of Feature-Extraction
2. The Fully Connected Block — Consists of a fully connected simple neural network architecture. This layer performs the task of Classification based on the input from the convolutional block

The CONVOLUTIONAL LAYER is related to feature extraction

The architecture of a ConvNet is analogous to that of the connectivity pattern of Neurons in the Human Brain and was inspired by the organization of the Visual Cortex. Individual neurons respond to stimuli only in a restricted region of the visual field known as the Receptive Field. A collection of such fields overlap to cover the entire visual area.
A ConvNet is able to **successfully capture the Spatial and Temporal dependencies** in an image through the application of relevant filters. The architecture performs a better fitting to the image dataset due to the reduction in the number of parameters involved and reusability of weights. In other words, the network can be trained to understand the sophistication of the image better.

Input Image



4x4x3 RGB Image

In the figure, we have an RGB image which has been separated by its three color planes — Red, Green, and Blue. There are a number of such color spaces in which images exist — Grayscale, RGB, HSV, CMYK, etc.

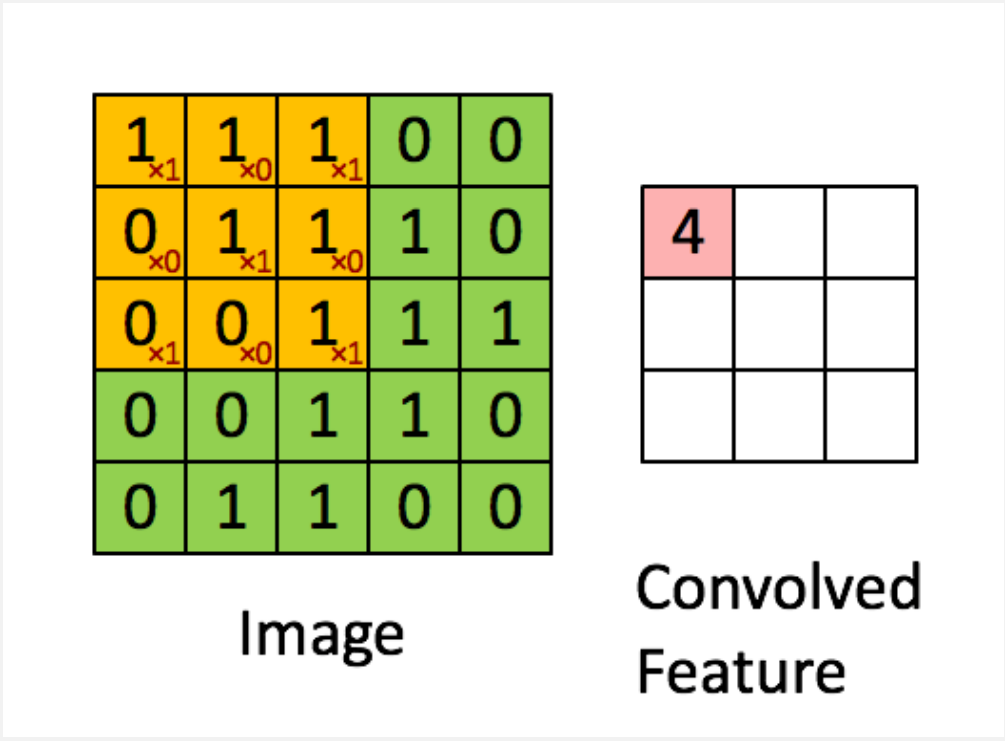
You can imagine how computationally intensive things would get once the images reach dimensions, say 8K (7680×4320). The role of the ConvNet is to reduce the images into a form which is easier to process, without losing features which are critical for getting a good prediction. This is important when we are to design an architecture which is not only good at learning features but also is scalable to massive datasets.

CNNs are a specialized form of deep feedforward networks. Starting at the input layer, they are composed of multiple alternating convolutional and subsampling layers, finally followed by an output layer that is task dependent.

Convolution Layer — The Kernel

K =

```
1 0 1
0 1 0
1 0 1
```



Convoluting a 5x5x1 image with a 3x3x1 kernel to get a 3x3x1 convolved feature

Image Dimensions = 5 (Height) x 5 (Breadth) x 1 (Number of channels, eg. RGB)

In the above demonstration, the green section resembles our **5x5x1 input image, I**. The element involved in carrying out the convolution operation in the first part of a Convolutional Layer is called the **Kernel/Filter, K**, represented in the color yellow. We have selected **K as a 3x3x1 matrix.**

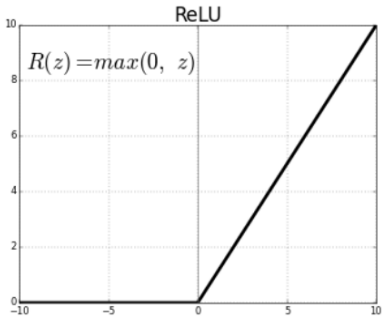
```
Kernel/Filter,
K =
1 0 1
0 1 0
1 0 1
```

The Kernel shifts 9 times because of **Stride Length = 1 (Non-Strided)**, every time performing a **matrix multiplication operation between K and the portion P of the image** over which the kernel is hovering.

Striding: In 'strided' convolution, instead of shifting the filter one-row or onecolumn at a time, we shift it, maybe, 2 or 3 rows or columns, each time. This is generally done to reduce the no of calculation and also reduce

the size of the output matrix. For large image, this doesn't results in loss of data, but reduces computation cost on a large scale

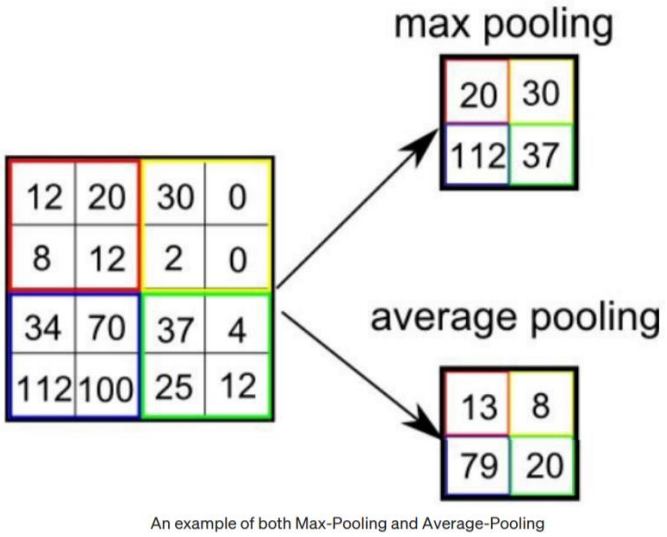
RELU Activation: RELU or Rectified Linear Unit is applied on all the cells of all the output-matrix. The function is defined as:



Convolutional networks were inspired by biological processes[9][10][11][12] in that the connectivity pattern between neurons resembles the organization of the animal visual cortex. Individual cortical neurons respond to stimuli only in a restricted region of the visual field known as the receptive field. The receptive fields of different neurons partially overlap such that they cover the entire visual field.
The name "convolutional neural network" indicates that the network employs a mathematical operation called convolution. Convolutional networks are a specialized type of neural networks that use convolution in place of general matrix multiplication in at least one of their layers.

Pooling layers

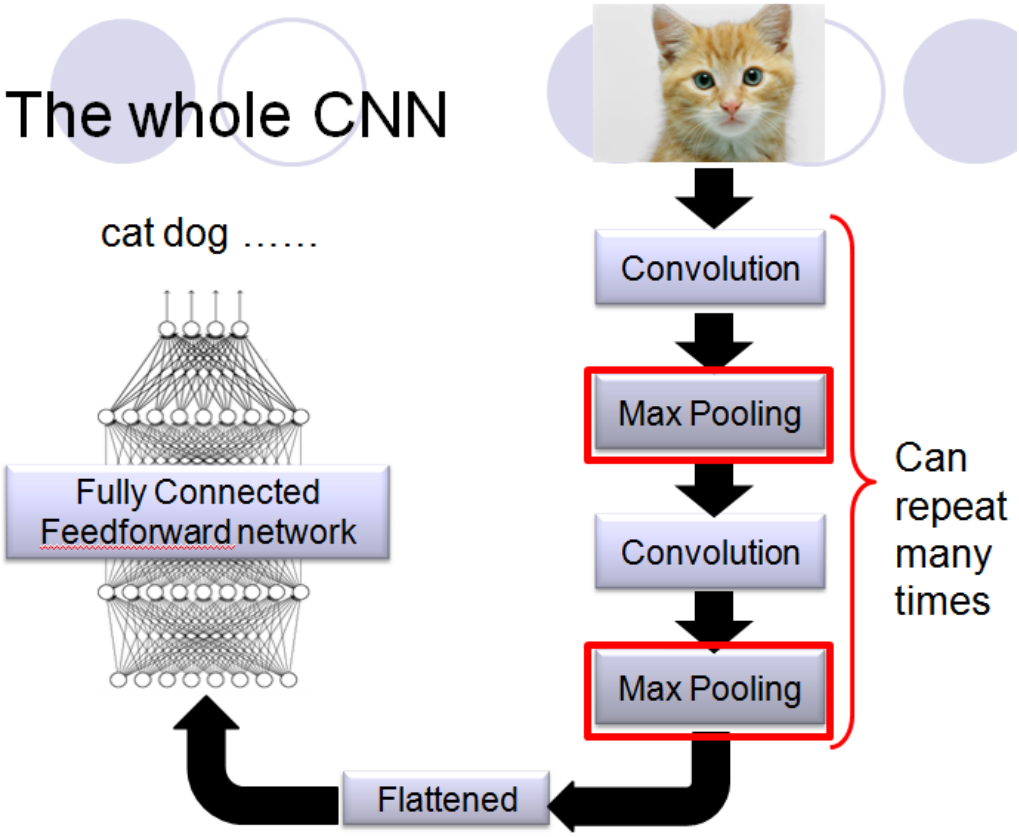
Convolutional networks may include local and/or global pooling layers along with traditional convolutional layers. Pooling layers reduce the dimensions of data by combining the outputs of neuron clusters at one layer into a single neuron in the next layer. Local pooling combines small clusters, tiling sizes such as 2 x 2 are commonly used. Global pooling acts on all the neurons of the feature map.There are two common types of pooling in popular use: max and average. *Max pooling* uses the maximum value of each local cluster of neurons in the feature map,while *average pooling* takes the average value.

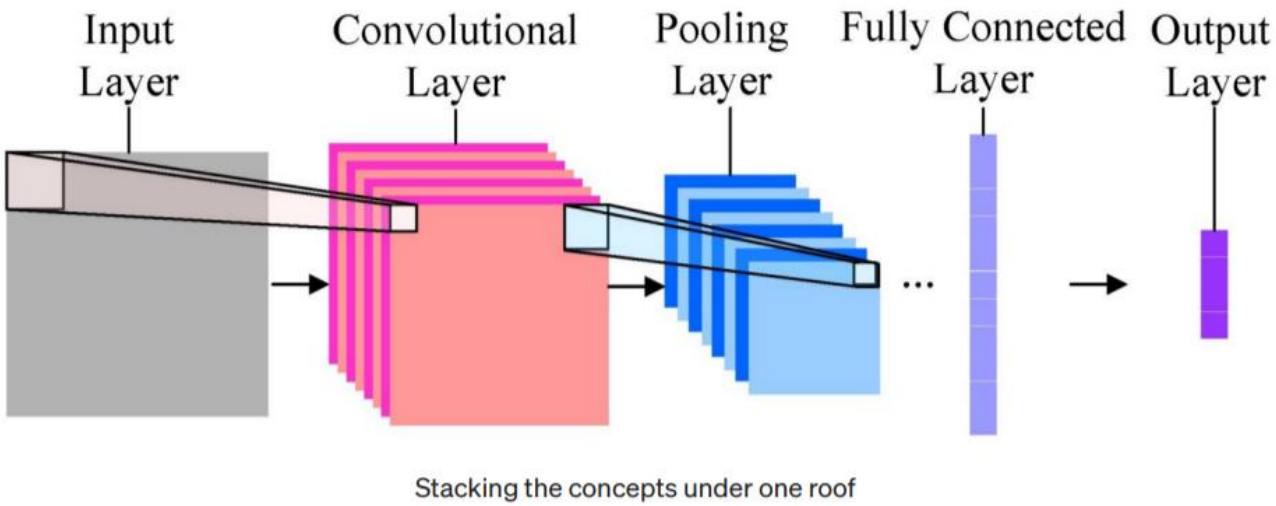


An example of both Max-Pooling and Average-Pooling

Fully connected layers[edit]

Fully connected layers connect every neuron in one layer to every neuron in another layer. It is the same as a traditional multi-layer perceptron neural network (MLP). The flattened matrix goes through a fully connected layer to classify the images.





Stacking the concepts under one roof

So, what happens from the beginning to the end of the CNN.

1. We give input an RGB image. It is generally a 2-D matrix defined for the 3 color channels. Let each channel be of size n X n. Thus, the input is a n X n X 3 dimension matrix.
2. We have a 3-D matrix, consisting of (say) 'k' no. of filters of a size (say f X f).
3. We perform PADDING on the image of say 'p' rows & columns. Thus, the input-matrix becomes (n+2*p) X (n+2*p) X 3 dimensions.
4. Next, we perform the strided convolution operation of the filter-matrices on the input image-matrices, as described before, using a stride of say 's'. Thus, the output matrix becomes
5. We perform POOLING over the output-matrix of each layer. The dimension of the output-matrices depends of the size of the pooling-filter and the stride length we have defined.
6. We perform the same operation from step 3–5, nearly three time.
7. On receiving the output-matrices of some dimension say a X b X l, we flatten the output into a 1-D array, i.e., we arrange all the values from the matrices sequentially in an array which forms the input matrix for the Fully — Connected Neural Network.
8. This neural network performs the desired calculation and gives the result.

Difference Between CNN and RNN

Convolutional Neural Networks	**Recurrent Neural Networks**
In deep learning, a convolutional neural network (CNN, or ConvNet) is a class of deep neural networks, most commonly applied to analyzing visual imagery.	A recurrent neural network (RNN) is a class of artificial neural networks where connections between nodes form a directed graph along a temporal sequence.
It is suitable for spatial data like images.	RNN is used for temporal data, also called sequential data.
CNN is a type of feed-forward artificial neural network with variations of	RNN, unlike feed-forward neural networks- can use their internal

multilayer perceptron’s designed to use minimal amounts of preprocessing.	memory to process arbitrary sequences of inputs.
CNN is considered to be more powerful than RNN.	RNN includes less feature compatibility when compared to CNN.
This CNN takes inputs of fixed sizes and generates fixed size outputs.	RNN can handle arbitrary input/output lengths.
CNN’s are ideal for images and video processing.	RNNs are ideal for text and speech analysis.
Applications include Image Recognition, Image Classification, Medical Image Analysis, Face Detection and Computer Vision.	Applications include Text Translation, Natural Language Processing, Language Translation, Sentiment Analysis and Speech Analysis.