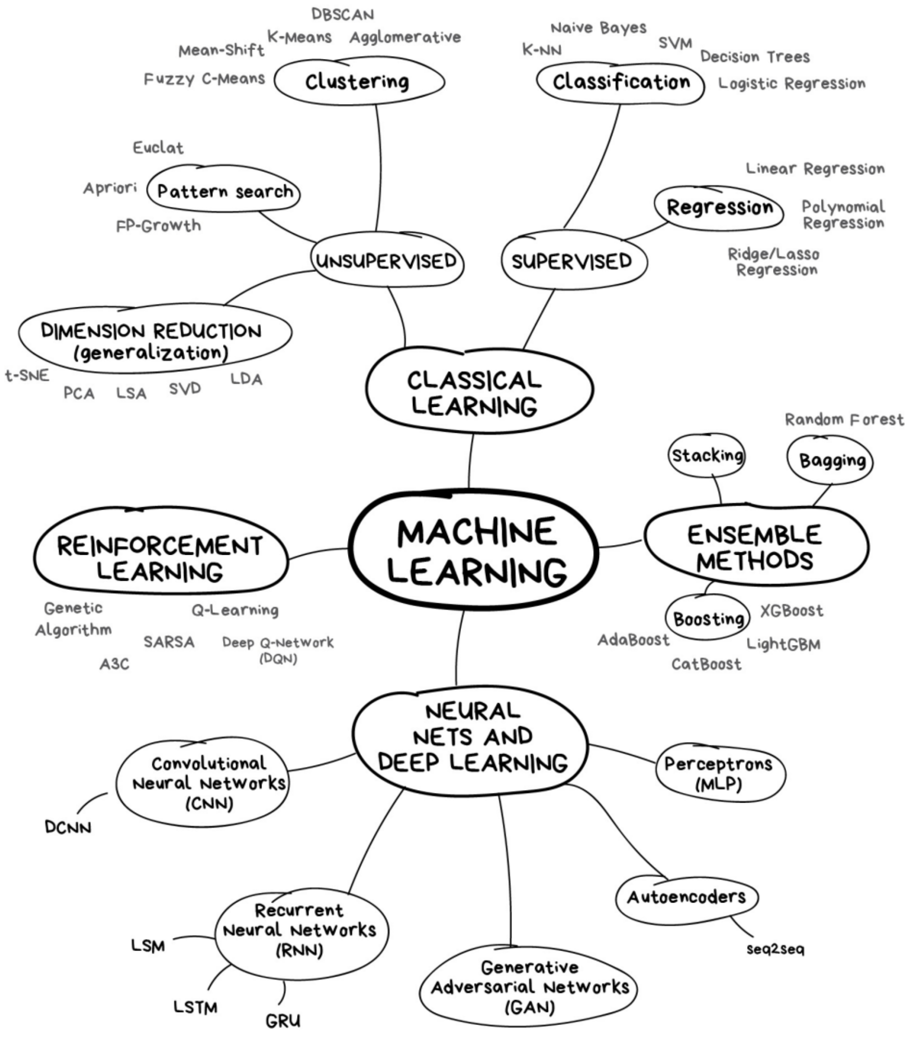
Deep Learning

Penyusun Modul: Chairul Aulia
Editor: Citra Chairunnisa

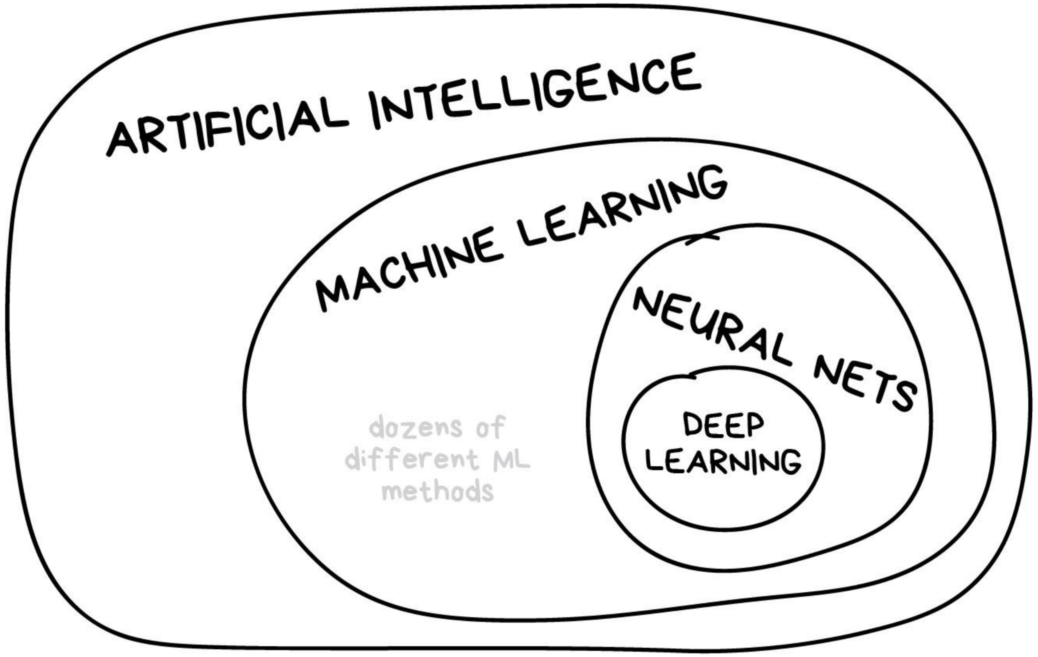
Let’s recap the machine learning world we have explored!

Wow, what a journey! There are still some methods not mentioned in the diagram and also some topics we did not cover! But we believe now with the basics and fundamental you have learned, you can explore to gain more knowledge on your own whenever you want to try new methods in the future (whether you are in a project involving machine learning or just doing a hobby).
Now we will discuss the next cluster you see at the bottom, i.e. Deep Learning



Deep Learning

- **Deep Learning** is a subset of **machine learning,**
- While machine learning uses simpler concepts, deep learning works with **artificial neural networks**, which are designed to imitate how humans think and learn.
- The concept of deep learning did exist back then, but we did not have hardware powerful enough to be able to make use of it. With the rise of computational power and big data generated everywhere it is now often to have neural networks of 10+ layers and 100+ layer tested.
- Now computers can go deeper by adding more and more layers hence the name “deep learning”

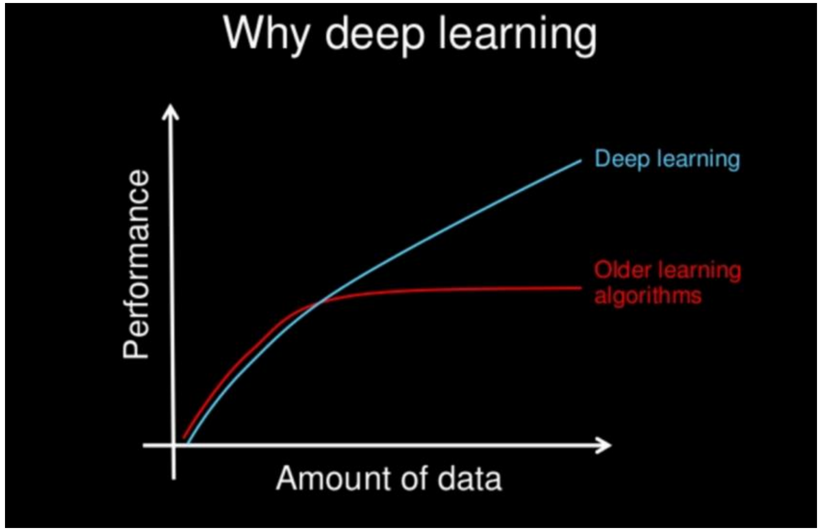


Importance of Deep Learning

- Machine learning works only with sets of structured and semi-structured data, while deep learning works with both structured and unstructured data
- Deep learning algorithms can perform complex operations efficiently, while machine learning algorithms cannot
- Machine learning algorithms use labeled sample data to extract patterns, while deep learning accepts large volumes of data as input and analyzes the input data to extract features out of an object
- The performance of machine learning algorithms decreases as the number of data increases; so to maintain the performance of the model, we need a deep learning

Differences between ML and DL

Data dependencies
The most important difference between deep learning and traditional machine learning is its performance as the scale of data increases. When the data is small, deep learning algorithms don't perform that well. This is because deep learning algorithms need a large amount of data to understand it perfectly. On the other hand, traditional machine learning algorithms with their handcrafted rules prevail in this scenario.



Differences between ML and DL

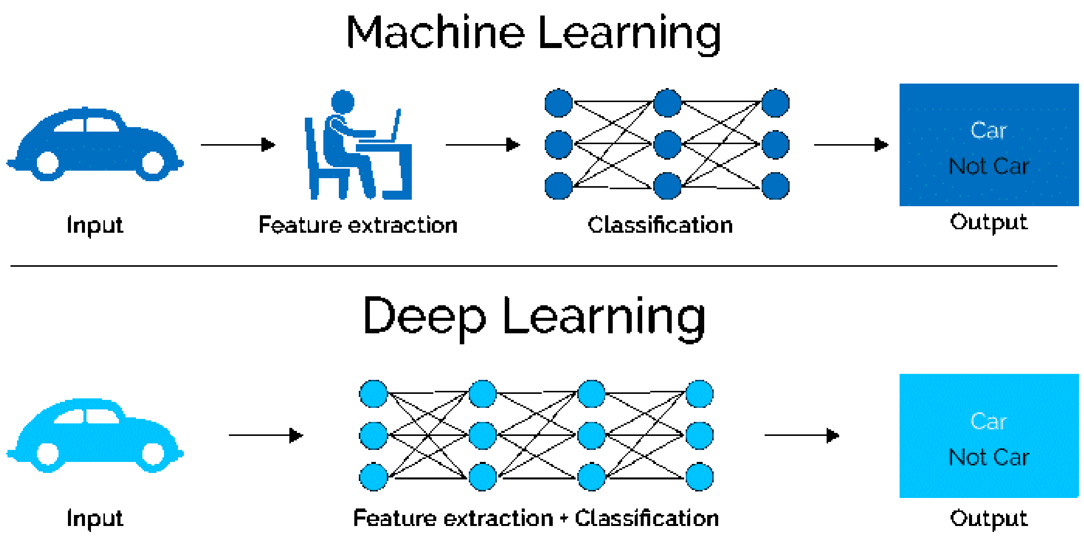
Hardware dependencies

Deep learning algorithms heavily depend on high-end machines, contrary to traditional machine learning algorithms, which can work on low-end machines. This is because the requirements of deep learning algorithm include GPUs which are an integral part of its working. Deep learning algorithms inherently do a large amount of matrix multiplication operations. These operations can be efficiently optimized using a GPU because GPU is built for this purpose.

Differences between ML and DL

Feature engineering

Feature engineering is a process of putting domain knowledge into the creation of feature extractors to reduce the complexity of the data and make patterns more visible to learning algorithms to work. This process is difficult and expensive in terms of time and expertise.
In Machine learning, most of the applied features need to be identified. Deep learning algorithms try to learn high-level features from data. This is a very distinctive part of Deep Learning and a major step ahead of traditional Machine Learning.



Differences between ML and DL

Execution time

Usually, a deep learning algorithm takes a long time to train. This is because there are so many parameters in a deep learning algorithm that training them takes longer than usual.

This is turn is completely reversed on testing time. At test time, deep learning algorithm takes much less time to run. Whereas, if you compare it with k-nearest neighbors (a type of machine learning algorithm), test time increases on increasing the size of data.

Differences between ML and DL

Interpretability
Suppose we use deep learning to give automated scoring to essays. The performance it gives in scoring is quite excellent and is near human performance. But there's is an issue. It does not reveal why it has given that score. Indeed mathematically you can find out which nodes of a deep neural network were activated, but we don't know what there neurons were supposed to model and what these layers of neurons were doing collectively. So we fail to interpret the results.

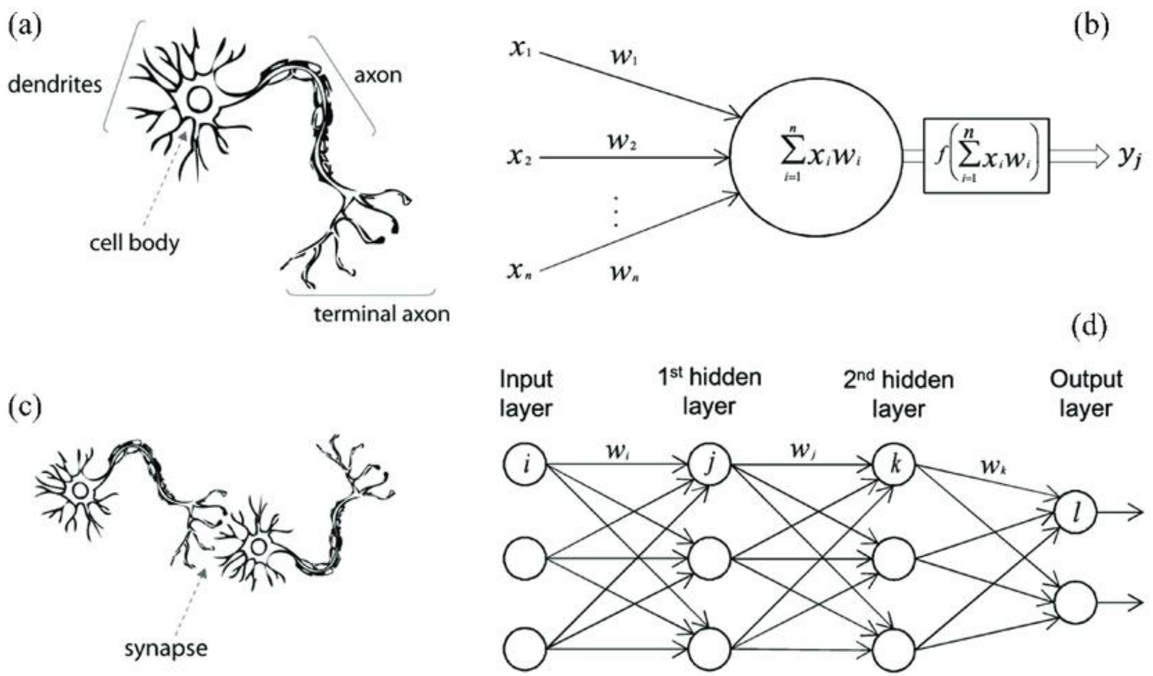
On the other hand, machine learning algorithms like decision trees give us crisp rules as to why it chose what it chose, so it is particularly easy to interpret the reasoning behind it. Therefore, algorithms like decision trees and linear/logistic regression are primarily used in industry for interpretability.

When To Use Deep Learning?

- If the data amount is large, Deep Learning outperforms conventional approaches. Traditional Machine Learning techniques, on the other hand, are preferred when dealing with tiny amounts of data.
- Deep Learning approaches require high-end infrastructure to train in a reasonable amount of time.
- When there is a shortage of domain awareness for feature introspection, Deep Learning approaches outperform others since you have to worry about tasks less.
- Deep Learning excels at complicated issues like image classification, natural language processing, and speech recognition.

Neural Network

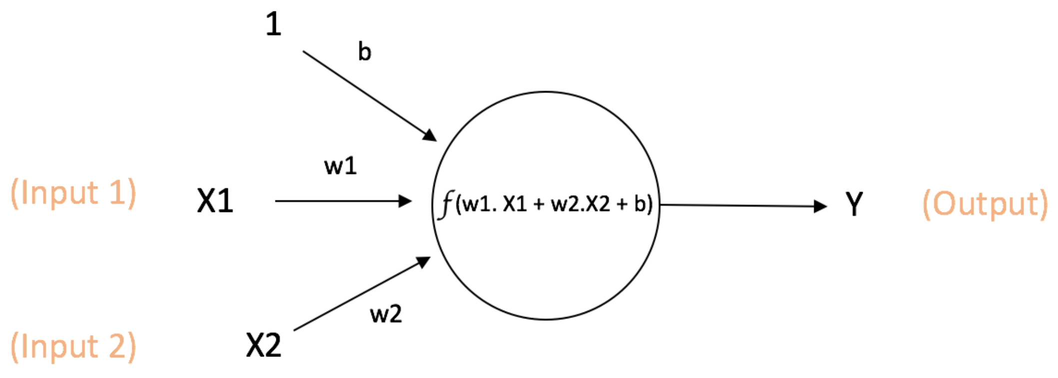
Neural network is a system **modeled on the human brain**, consisting of an input layer, multiple hidden layers, and an output layer. Each node is called **neuron**. Unlike the human neural system, neurons in neural networks are connected to each other through the **layers**. every neuron in a layer is connected to neurons in the next layer. Data is fed as input to the neurons. The information is transferred to the next layer using appropriate weights and biases



Neurons

Each neuron in a neural network performs the following operations:

- The product of each input and the weight of the channel it is passed over is found
- The sum of the weighted products is computed, which is called the weighted sum
- A bias value of the neuron is added to the weighted sum
- The final sum is then subjected to a particular function known as the activation function



Output of neuron $= Y = f(w1.X1 + w2.X2 + b)$

The above network takes numerical inputs **X1** and **X2** and has weights **w1** and **w2** associated with those inputs. Additionally, there is another input **1** with weight **b** (called the **Bias**) associated with it

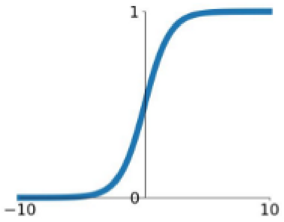
Activation Function

There are different activation functions that we can choose according to the value that we want to predict. Let's assume that we want to predict 2 classes indicated as 0 and 1. So, what we need is a function that can map input values into output values between 0 and 1. Several activation functions you may encounter in practice:

- **Sigmoid:** takes a real-valued input and squashes it to range between 0 and 1
- **tanh:** takes a real-valued input and squashes it to the range [-1, 1]
- **ReLU**: ReLU stands for Rectified Linear Unit. It takes a real-valued input and thresholds it at zero (replaces negative values with zero)

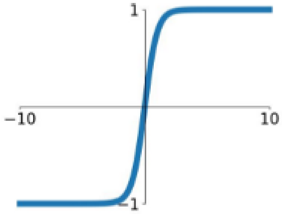
Sigmoid

$\sigma(x) = \frac{1}{1+e^{-x}}$



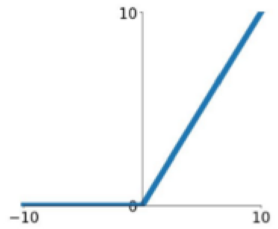
tanh

$\tanh(x)$



ReLU

$\max(0, x)$



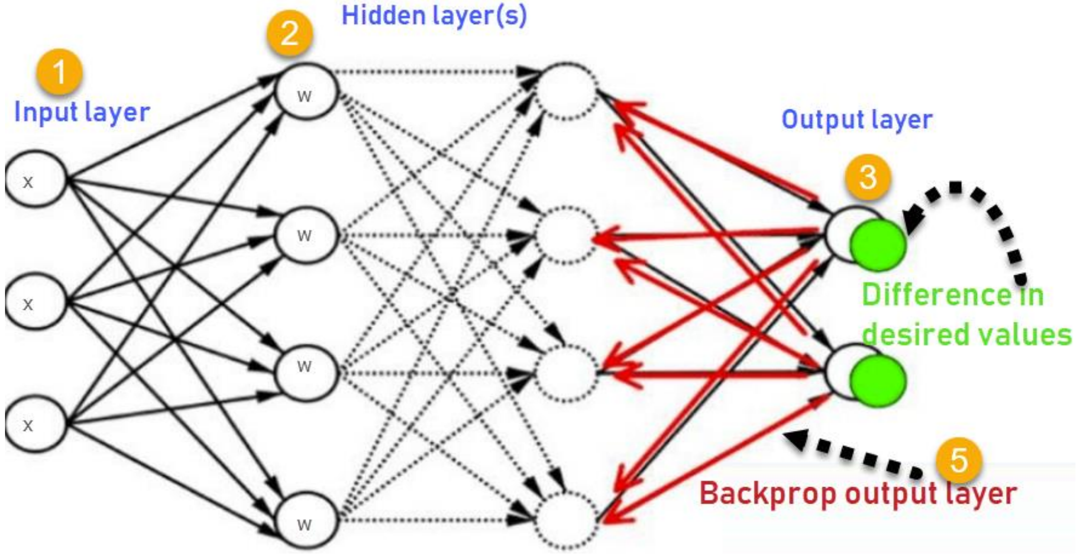
Different activation functions perform differently on different data distribution. So sometimes you have to check different activation functions and find out which works better for a particular problem

Back-Propagation

Backpropagation is the essence of neural network training. It is the method of fine-tuning the weights of a neural network based on the error rate obtained in the previous epoch (i.e., iteration). Proper tuning of the weights allows you to reduce error rates and make the model reliable by increasing its generalization.

Backpropagation in neural network is a short form for “backward propagation of errors.” It is a standard method of training artificial neural networks. This method helps calculate the gradient of a loss function with respect to all the weights in the network.

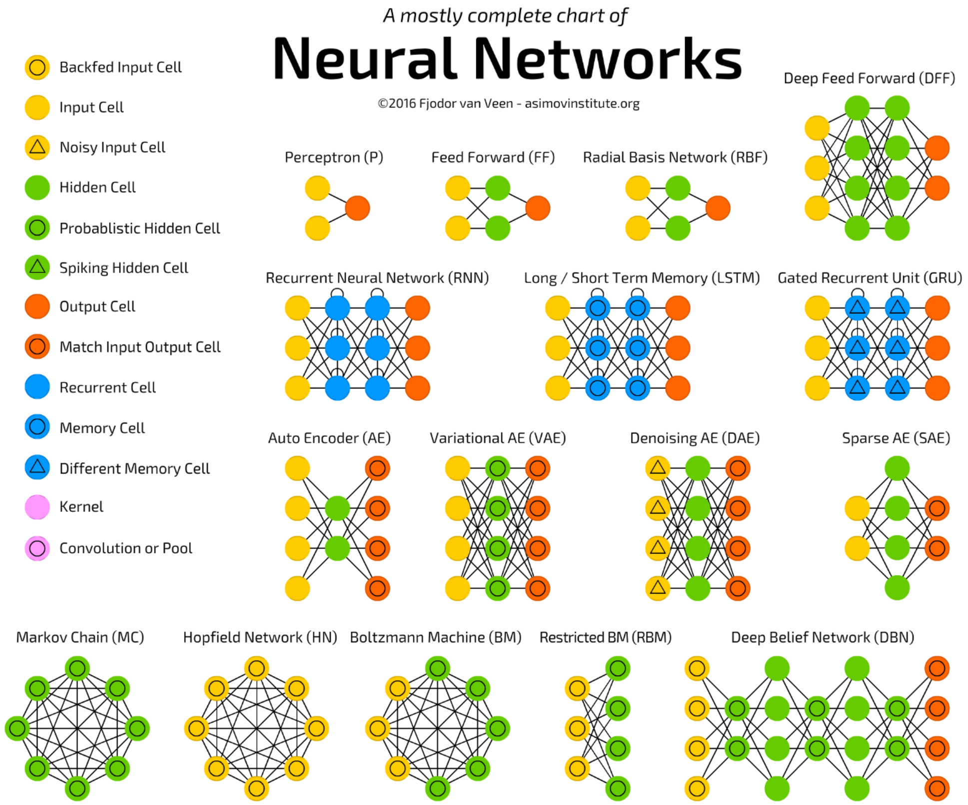
Back-Propagation Steps



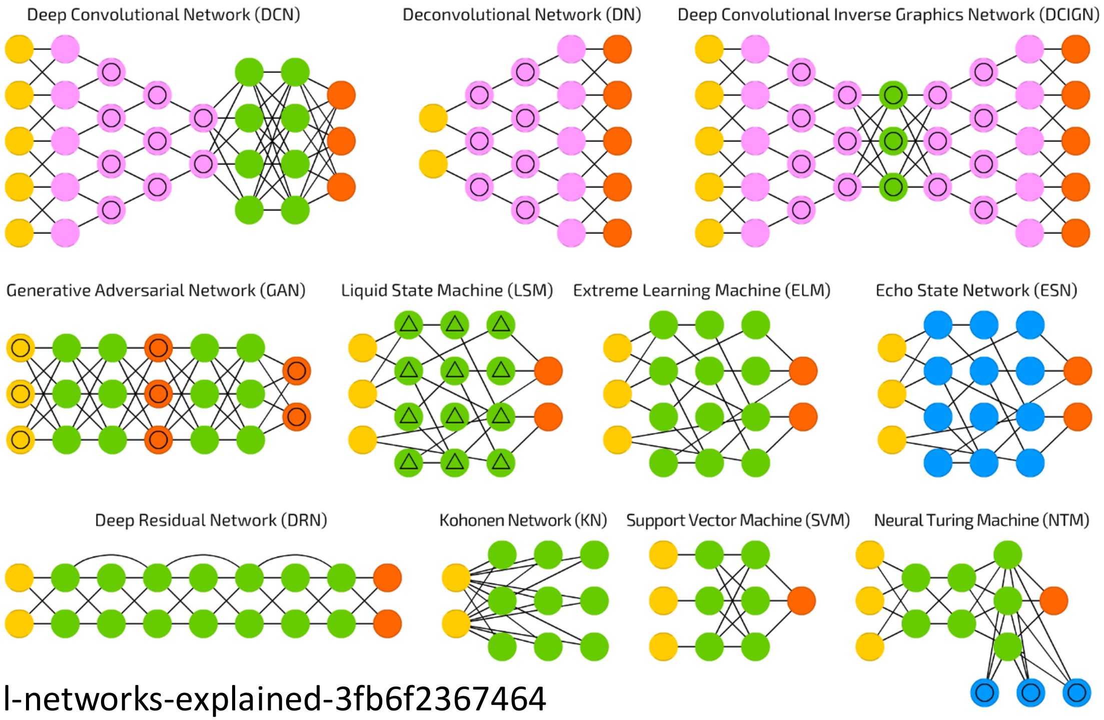
1. Inputs X, arrive through the preconnected path
2. Input is modeled using real weights W. The weights are usually randomly selected.
3. Calculate the output for every neuron from the input layer, to the hidden layers, to the output layer.
4. Calculate the error in the outputs

   $Error_B$= Actual Output – Desired Output
5. Travel back from the output layer to the hidden layer to adjust the weights such that the error is decreased.
6. Keep repeating the process until the desired output is achievedt

Different Architectures of Neural Networks



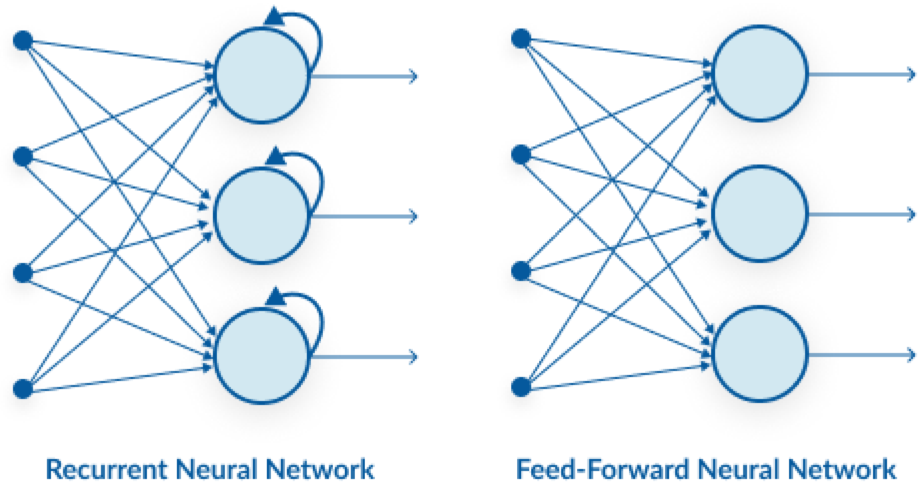
Of course you don't have to remember all these charts, just note that there are many forms of networks used in different applications.



https://towardsdatascience.com/the-mostly-complete-chart-of-neural-networks-explained-3fb6f2367464

Recurrent Neural Network (RNN)

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:

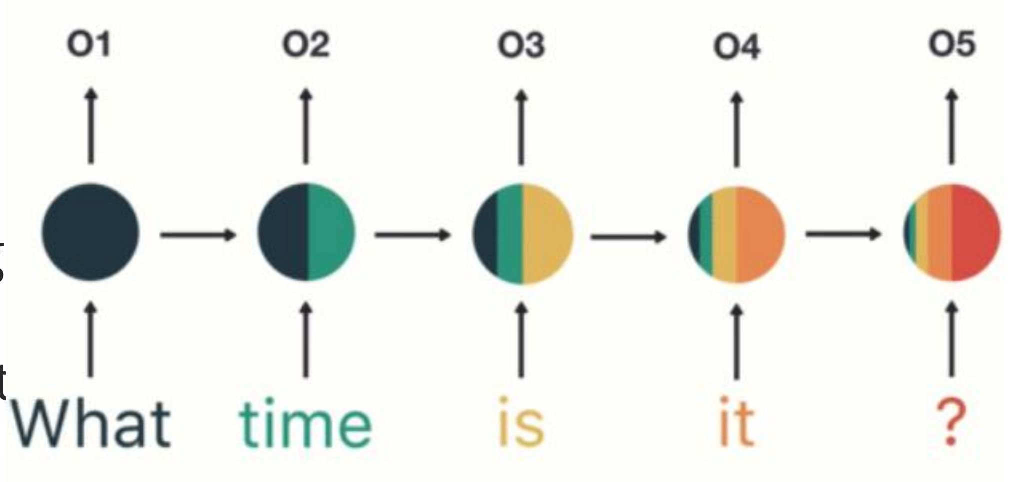
- Time Series data
- Text data
- Audio data

“Whenever there is a sequence of data and that temporal dynamics that connects the data is more important than the spatial content of each individual frame.” – Lex Fridman (MIT)

Recurrent Neural Network (RNN)

Advantages:

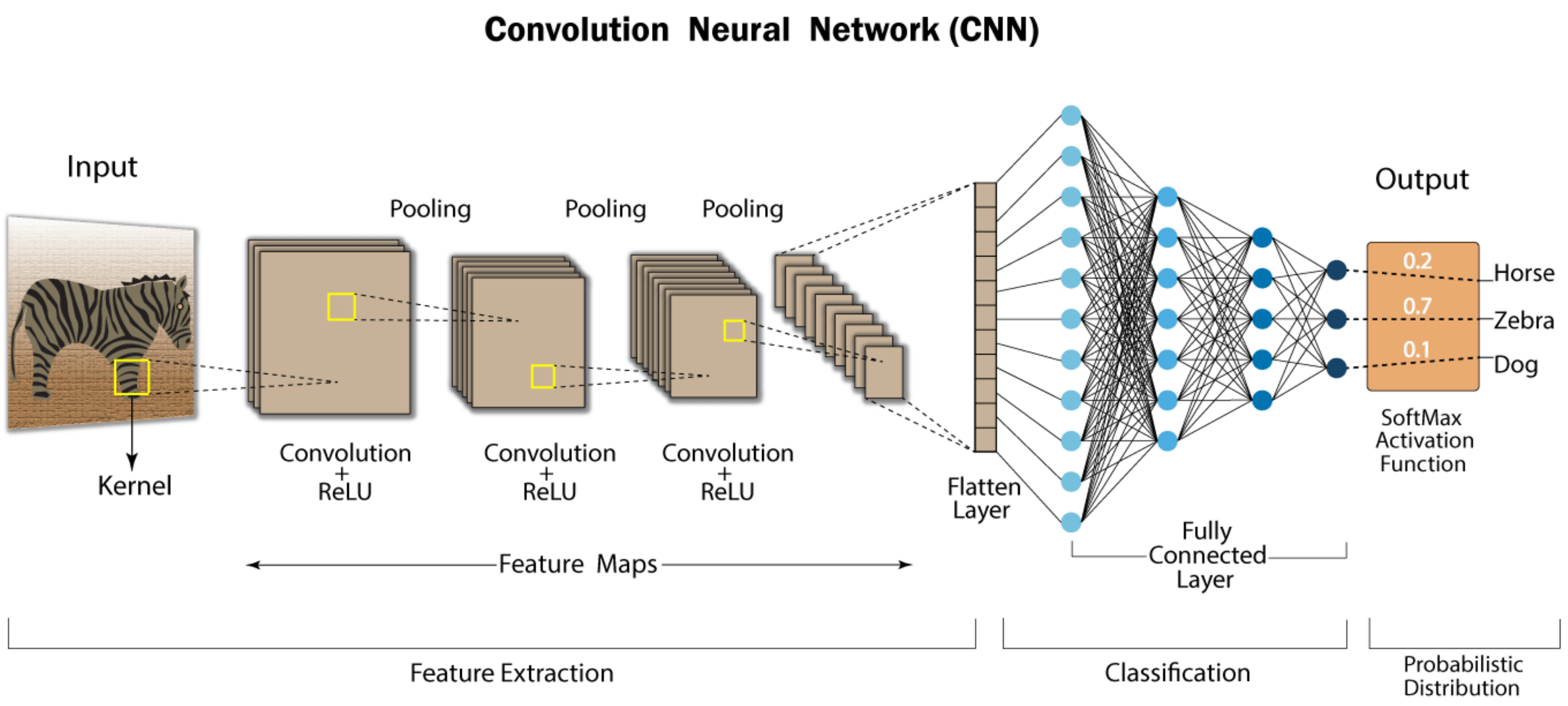
- RNN captures the sequential information present in the input data i.e. dependency between the words in the text while making predictions
- RNNs share the parameters across different time steps. This is popularly known as Parameter Sharing. This results in fewer parameters to train and decreases the computational cost



Convolutional Neural Network (CNN)

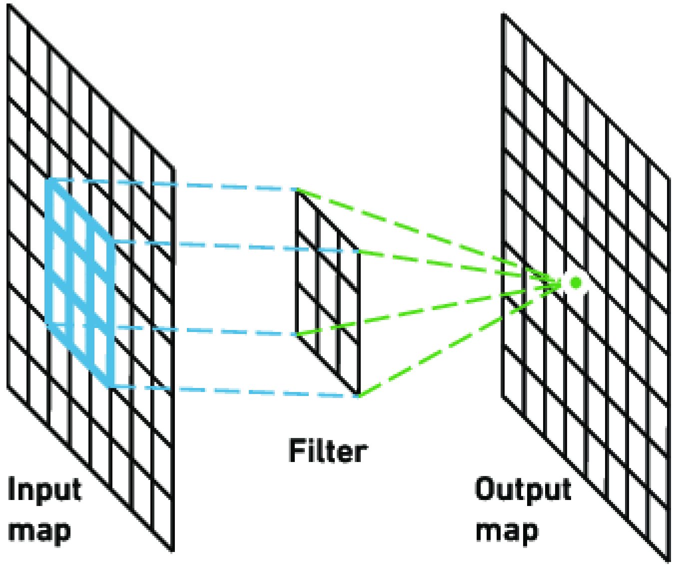
- Convolutional Neural Networks also known as CNNs or ConvNets, are a type of neural network whose connectivity structure is inspired by the organization of the animal visual cortex. Small clusters of cells in the visual cortex are sensitive to certain areas of the visual field. Individual neuronal cells in the brain respond or fire only when certain orientations of edges are present. Some neurons activate when shown vertical edges, while others fire when shown horizontal or diagonal edges. A convolutional neural network is a type of artificial neural network mostly used to evaluate visual information.
- The building blocks of CNNs are filters a.k.a. kernels. Kernels are used to extract the relevant features from the input using the convolution operation.

Architecture of CNN

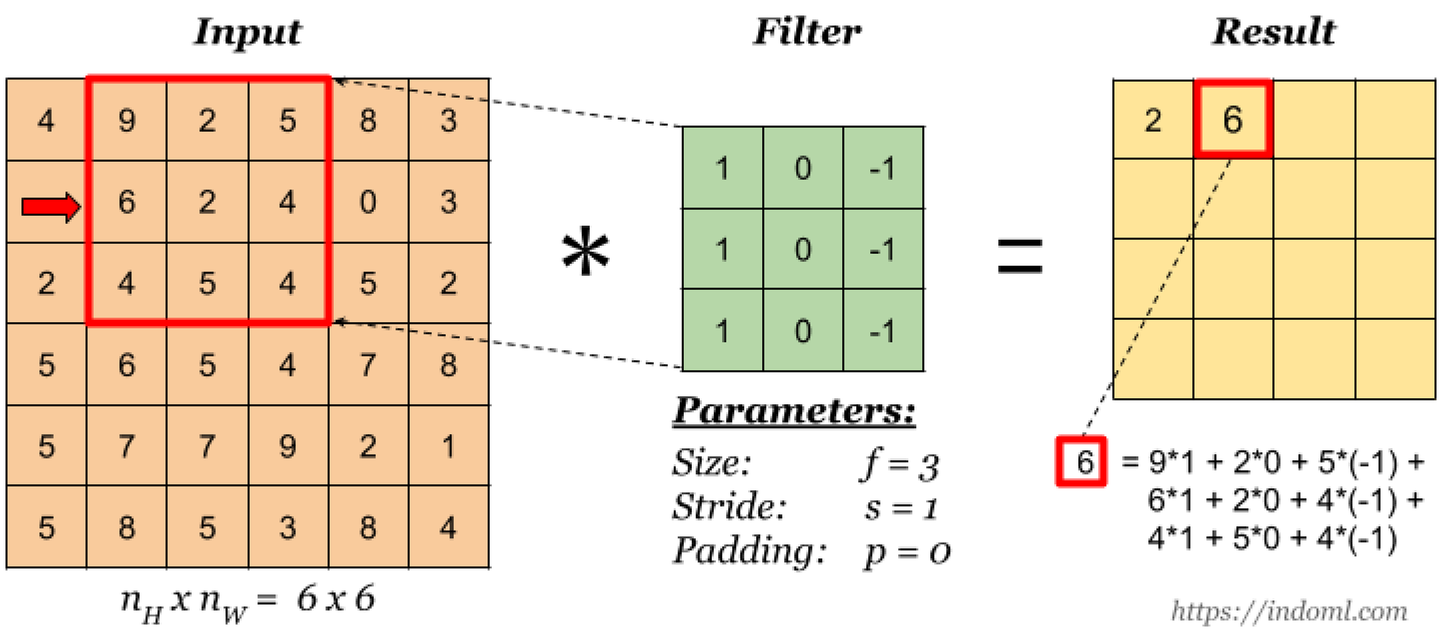


Convolutional Layer

This layer often contains input vectors (image), filters (feature detector) and output vectors (feature map). The image is abstracted to a feature map, also known as an activation map, after passing through a convolutional layer.

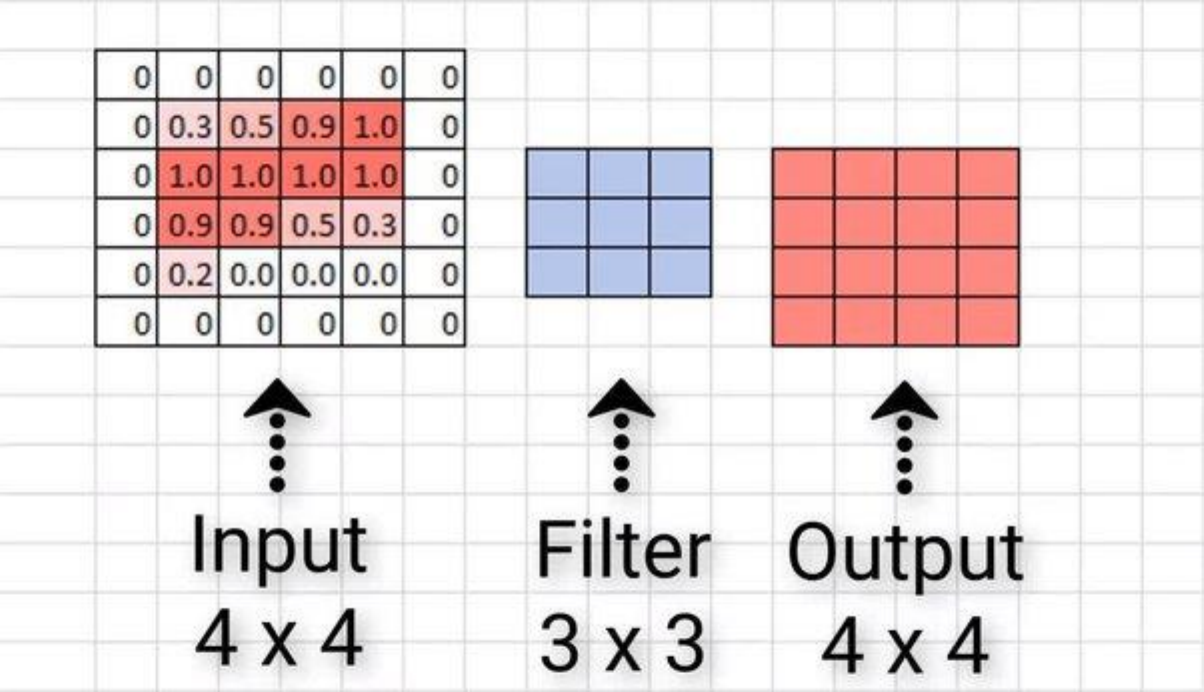


Convolution Operation



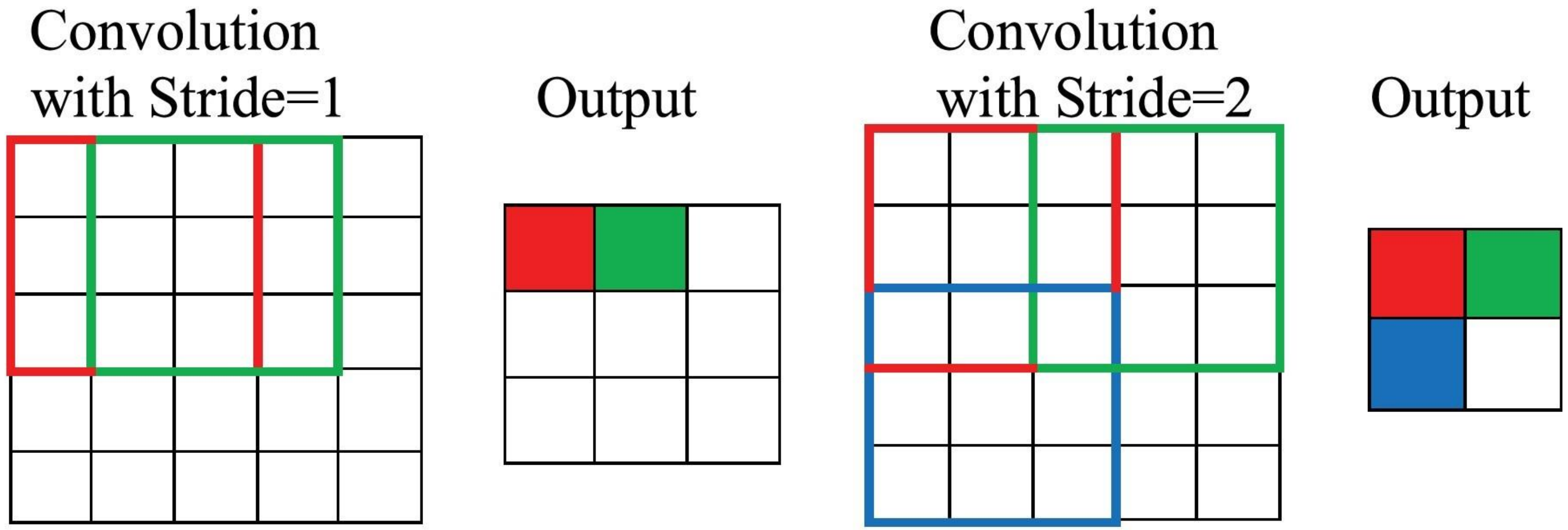
Padding

Padded convolution is used when preserving the dimension of an input matrix that is important to us and it helps us keep more of the information at the border of an image. We have seen that convolution reduces the size of the feature map. To retain the dimension of feature map as that of an input map, we pad or append the rows and column with zeros



Stride

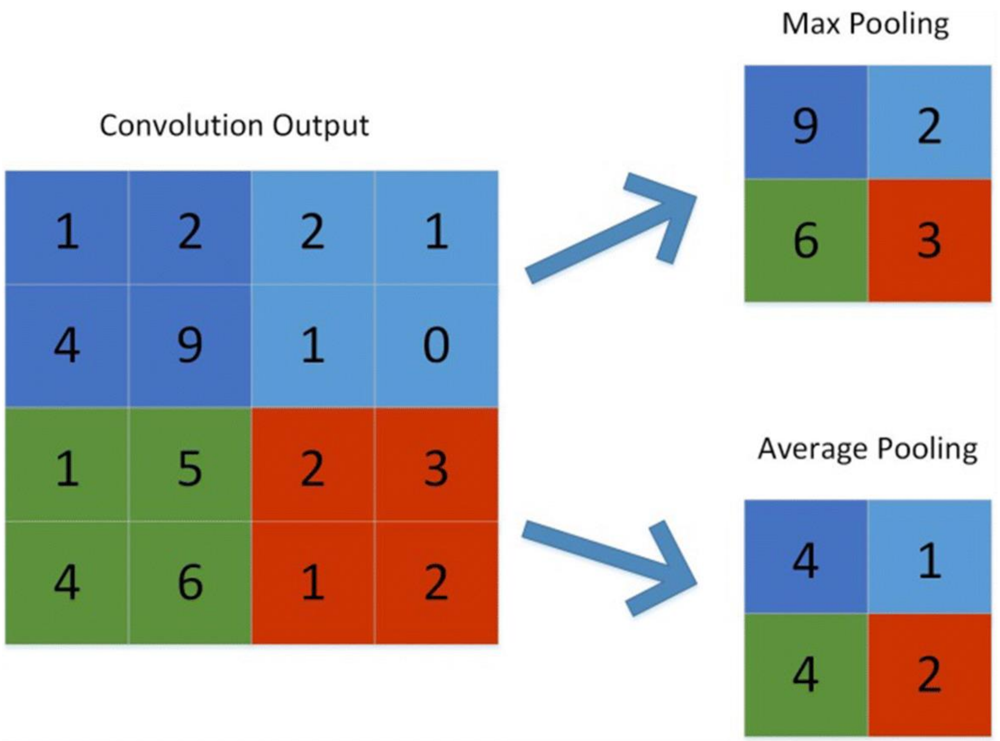
Stride determines how the filter convolves over the input matrix, i.e. how many pixels shift. When stride is set to 1, the filter moves across one pixel at a time, and when the stride is set to 2, the filter moves across two pixels at a time. The smaller the stride value, the smaller the output, and vice versa.



Pooling

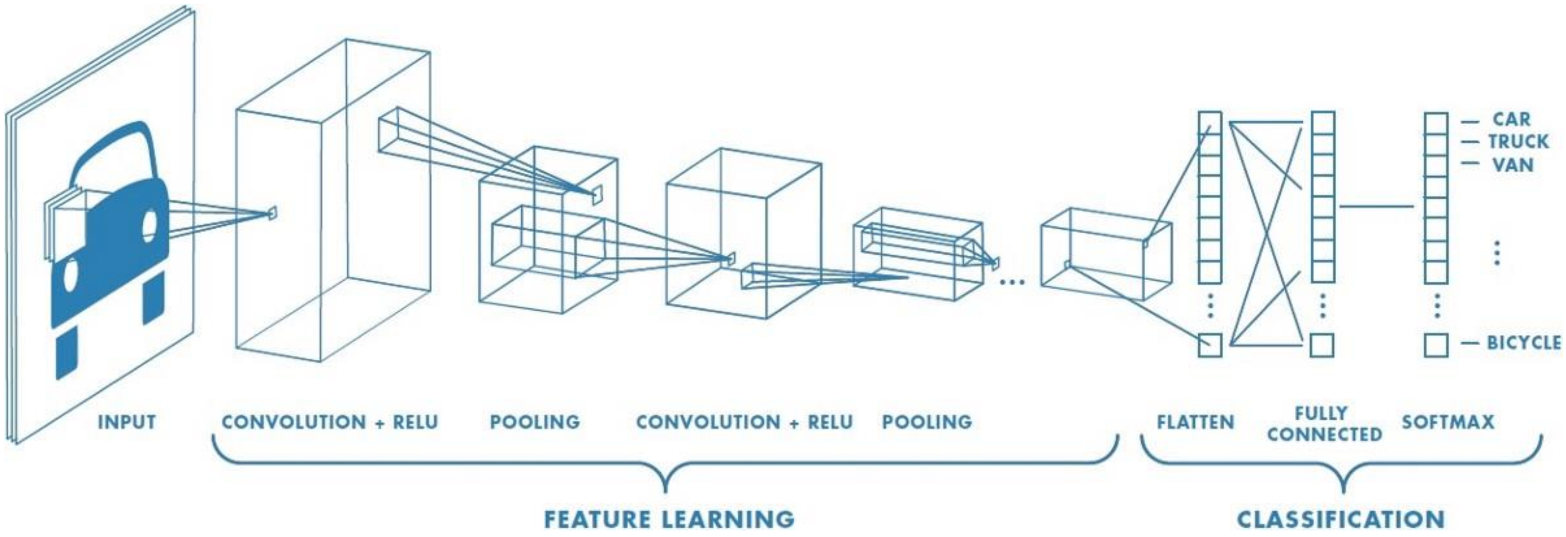
- Max-pooling: It chooses the most significant element from the feature map. The feature map's significant features are stored in the resulting max-pooled layer. It is the most popular method since it produces the best outcomes.
- Average pooling: It entails calculating the average for each region of the feature map.

Pooling gradually reduces the spatial dimension of the representation to reduce the number of parameters and computations in the network, as well as to prevent overfitting. If there is no pooling, the output has the same resolution as the input.



Fully connected

- After the convolution + pooling layers we add a couple of fully connected layers to wrap up the CNN architecture.
- Remember that the output of both convolution and pooling layers are 3D volumes, but a fully connected layer expects a 1D vector of numbers. So we flatten the output of the final pooling layer to a vector and that becomes the input to the fully connected layer. Flattening is simply arranging the 3D volume of numbers into a 1D vector, nothing fancy happens here.



Summation of Different Networks

	Artificial Neural Network (ANN)	Convolutional Neural Network (CNN)	Recurrent Neural Network (RNN)
Type of Data	Tabular Data, Text Data	Image Data	Sequence data
Parameter Sharing	No	Yes	Yes
Fixed Length input	Yes	Yes	No
Recurrent Connections	No	No	Yes
Vanishing and Exploding Gradient	Yes	Yes	Yes
Spatial Relationship	No	Yes	No
Performance	ANN is considered to be less powerful than CNN, RNN.	CNN is considered to be more powerful than ANN, RNN.	RNN includes less feature compatibility when compared to CNN.
Application	Facial recognition and Computer vision.	Facial recognition, text digitization and Natural language processing.	Text-to-speech conversions.
Main advantages	Having fault tolerance, Ability to work with incomplete knowledge.	High accuracy in image recognition problems, Weight sharing.	Remembers each and every information, Time series prediction.
Disadvantages	Hardware dependence, Unexplained behavior of the network.	Large training data needed, don't encode the position and orientation of object.	Gradient vanishing, exploding gradient.