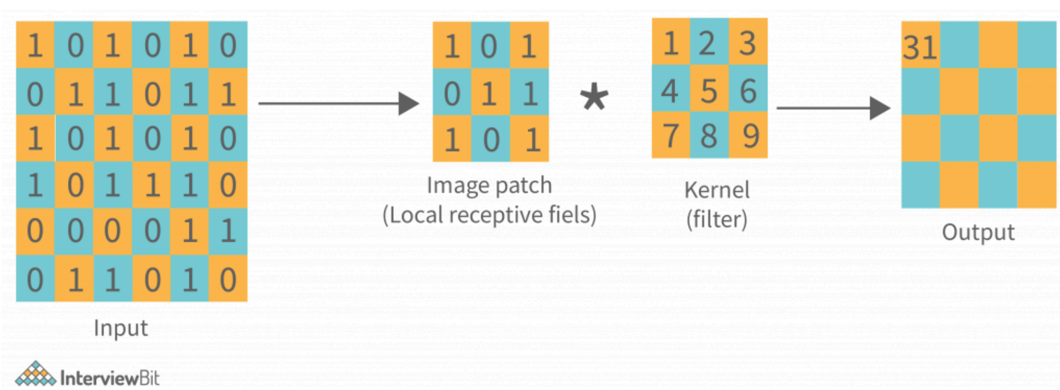
Unit-4

Creating CNN Structure:

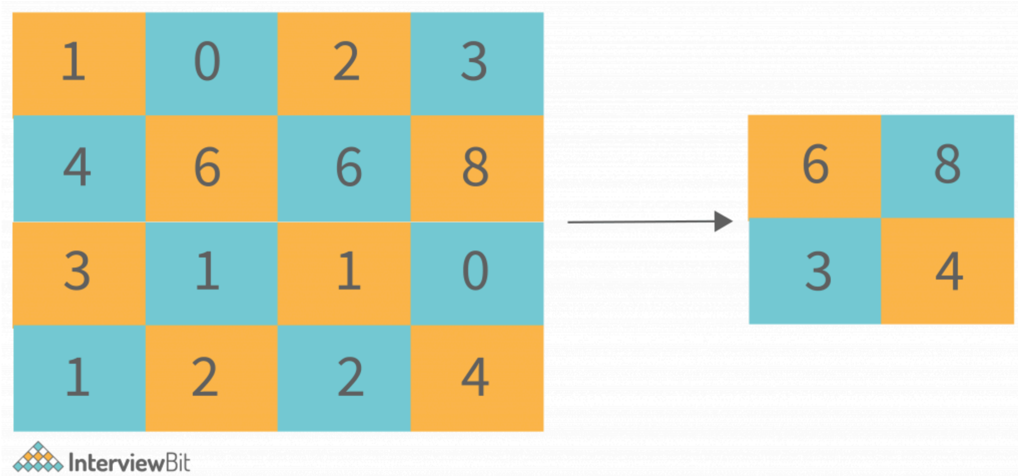
The ConvNet's job is to compress the images into a format that is easier to process while preserving elements that are important for obtaining a decent prediction. This is critical for designing an architecture that is capable of learning features while also being scalable to large datasets.

A convolutional neural network, Conv Nets in short has three layers which are its building blocks, let's have a look:

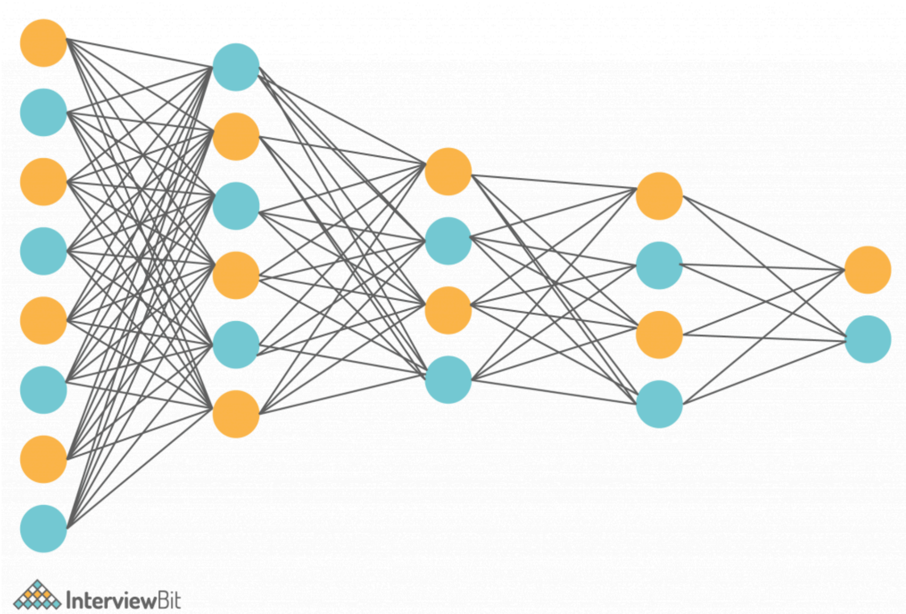
Convolutional Layer (CONV): They are the foundation of CNN, and they are in charge of executing convolution operations. The Kernel/Filter is the component in this layer that performs the convolution operation (matrix). Until the complete image is scanned, the kernel makes horizontal and vertical adjustments dependent on the stride rate. The kernel is less in size than a picture, but it has more depth. This means that if the image has three (RGB) channels, the kernel height and width will be modest spatially, but the depth will span all three.



Pooling Layer (POOL): This layer is in charge of reducing dimensionality. It aids in reducing the amount of computing power required to process the data. Pooling can be divided into two types: maximum pooling and average pooling. The maximum value from the area covered by the kernel on the image is returned by max pooling. The average of all the values in the part of the image covered by the kernel is returned by average pooling.



Fully Connected Layer (FC): The fully connected layer (FC) works with a flattened input, which means that each input is coupled to every neuron. After that, the flattened vector is sent via a few additional FC layers, where the mathematical functional operations are normally performed. The classification procedure gets started at this point. FC layers are frequently found near the end of CNN architectures if they are present.



Along with the above layers, there are some additional terms that are part of a CNN architecture.

Activation Function: The last fully connected layer's activation function is frequently distinct from the others. Each activity necessitates the selection of an appropriate activation function. The soft max function, which normalizes output real values from the last fully connected layer to target class probabilities, where each value ranges between 0 and 1 and all values total to 1, is an activation function used in the multiclass classification problem.

Dropout Layers: The Dropout layer is a mask that nullifies some neurons' contributions to the following layer while leaving all others unchanged. A Dropout layer can be applied to the input vector, nullifying some of its properties; however, it can also be applied to a hidden layer, nullifying some hidden neurons. Dropout layers are critical in CNN training because they prevent the training data from overfitting. If they aren't there, the first batch of training data has a disproportionately large impact on learning. As a result, learning of traits that occur only in later samples or batches would be prevented:

Applications of CNN:

- Face detection: CNNs have been used to detect faces within images. The network takes an image as the input and produces a set of values that represent characteristics of faces or facial features at different parts of the image. CNN has shown improved accuracy over previous algorithms, identifying faces 97% of the time according to various research papers. CNN can also be used to help reduce the amount of distortion in a face. CNNs can detect features such as eyes, nose, and mouth with great accuracy reducing distortions from things like angles or shadows on faces.
- Facial emotion recognition: CNNs have been used to help distinguish between different facial expressions such as anger, sadness, or happiness. CNNs can also be adapted to perform well with various lighting conditions and angles of faces within images.
- Object detection: CNN has been applied to object recognition across images by classifying objects based on shapes and patterns found within an image. CNN models have been created that can detect a wide range of objects from everyday items such as food, celebrities, or animals to more unusual ones including dollar bills and guns. Object detection is performed using techniques such as semantic or instance segmentation. CNNs have been used to localize and identify objects within images as well as create different views of those objects such as for use in drones or self-driving cars.
- Self-driving or autonomous cars: CNN has been used within the context of automated vehicles to enable them to detect obstacles or interpret street signs. CNNs have been used in conjunction with reinforcement learning, a branch of machine learning that focuses on positive and negative feedback from the environment to improve how CNN models respond when encountering certain situations.
- Auto translation: CNN is being used within the context of deep learning for automated translation between language pairs such as English and French. CNNs have been used to translate between language pairs such as Chinese and English with a high degree of accuracy, removing the need for word-for-word translation or bilingual human assistants.
- Next word prediction in sentence: CNNs have been used to predict the next word of a sentence given some context. CNN models can process multiple sentences and learn

which words typically follow others, such as “I am from India” followed by “I speak Hindi.”

- Handwritten character recognition: CNNs can be used to recognize handwritten characters. CNNs take the image of a character as an input and break it down into smaller sections, identifying points that can connect or overlap with other points in order to determine the shape of the larger character. CNN models have been created that are able to identify different languages including Chinese, Arabic, and Russian even when they’re written differently.
- X-ray image analysis: CNNs have been used for medical imaging to identify tumors or other abnormalities in X-ray images. CNN models can take an image of a human body part, such as the knee, and determine where within that image there might be a tumor based on previous similar images processed by CNN networks. CNN models can also be used to determine abnormalities from X-ray images. CNNs have been used to determine which area of an X-ray image contains a tumor or other abnormalities such as fractured bones.

Convolution Neural Networks – Concepts:

Convolution Layer:

Convolution is a mathematical function that takes two parts and multiplies them to produce a third function representing how the shape of one process is affected by the other. In simple words, Convolution is the result of two images fed to the network as matrices that gives an output used to extract features from the image.

The filters are applied to the input images or the other feature maps in deep convolutional neural networks in the convolutional layers. It is here that most of the network parameters are specified. The essential and most critical factors or the parameters are the no of kernels and their size

ReLU Layer:

ReLU, also called the Rectified Linear unit, is an integral part of the convolutional neural networks’ process and not a separate component. It is an additional step to the convolution. The ReLU activation function outputs the input with a positive value if not zero. The ReLU activation function is used in many neural networks as it is easier to train the model with the ReLU activation function to achieve better performance:

Pooling Layer:

The Next immediate layer to the convolution layer is the pooling layer. The main goal of the pooling layer is to reduce the size of the convoluted feature map by reducing the connections between the layers.

Max Pooling: This type of pooling is the most significant element of the feature map.

Average Pooling: In this type of pooling, the average of the elements is calculated.

Sum Pooling: In this type of pooling, some of the elements in the predefined section is Calculated**.**

Flattening Layer:

We will have pooled feature map after the convolution, pooling and the padding layer. Then, in the flattening layer, the flattening of the pooled feature map into a column takes place. The main objective of flattening is to insert the flattened data into an artificial neural network.

Once the flattening is complete, we get a vector of input data used to pass the network for us to process further.

Full conversion Layer:

It is the fully connected layer where the artificial and the convolutional neural network intersect. As a result, the web gets complex and sophisticated in this layer.

In artificial neural networks, the layer sandwiched between the input and the output layer is the hidden layer, whereas it is called the fully-connected layer in the convolutional. Once the flattening is complete, we get a vector of input data used to pass the network for us to process further. The objective of the artificial Neural network is to take the vector input from the input layer and combine the features that make a convolutional neural network to classify the images.

Soft max:

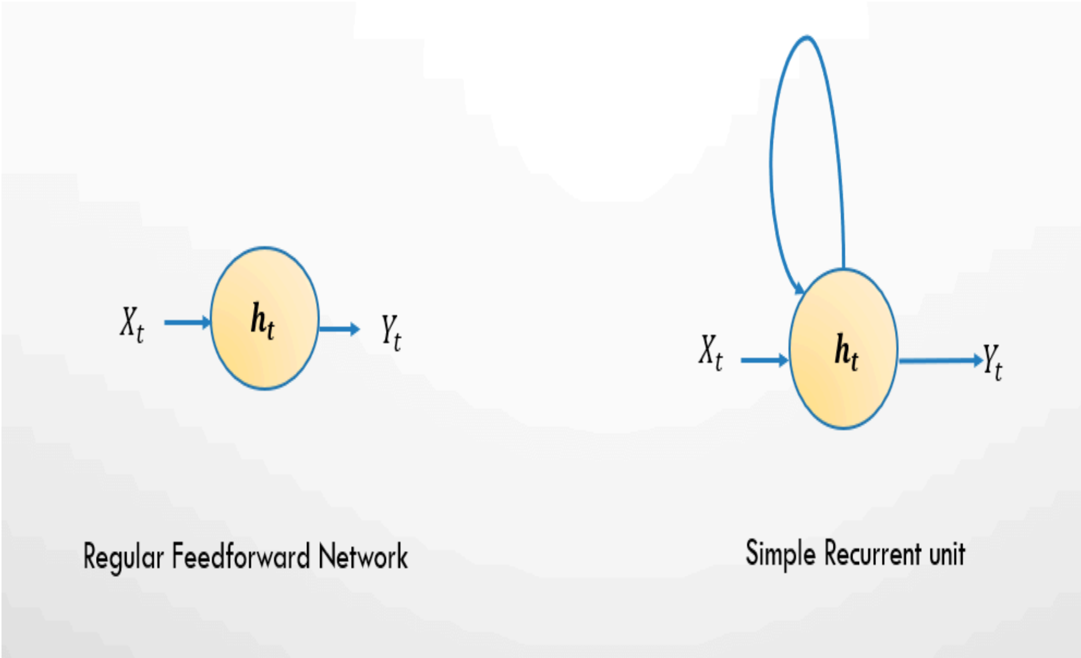
Classification results of the networks above the image raise how the predicted classes' prediction adds up to 1. The results are based on an activation function called the SoftMax.
Without the soft max function, the outputted classes will each have a real-value probability and summing them up will not result in any particular figure.

Cross-Entropy:

Cross-entropy is a loss function built upon the entropy theory and is generally calculated as the difference between two probability distributions. However, another process called KL Divergence is related to cross-entropy. The KL Divergence is calculated as the difference in relative entropy between the two probability distributions.

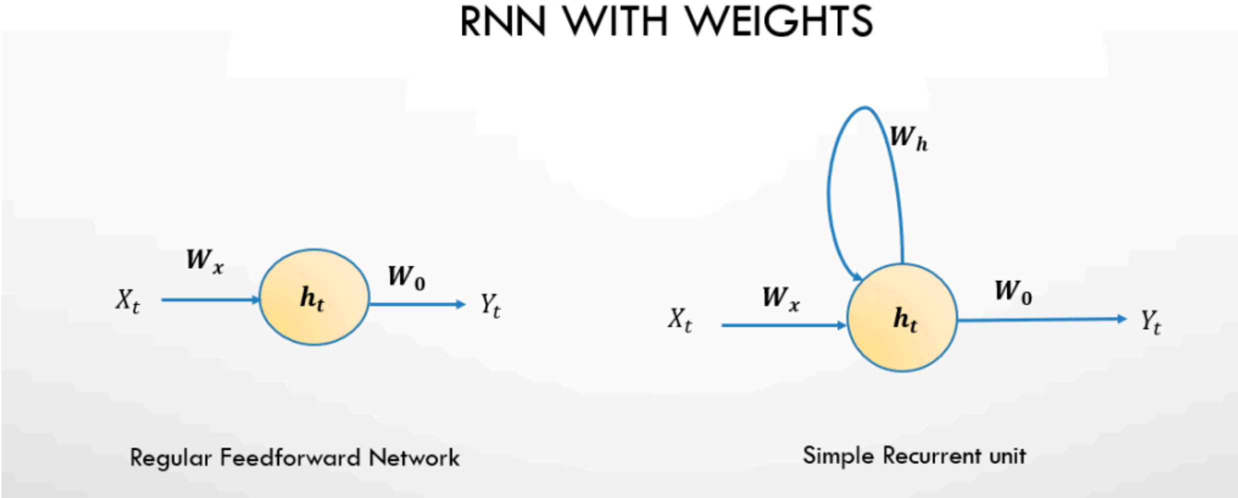
Recurrent Neural Network Architecture:

The right diagram in below figure in below figure represents a simple Recurrent unit.



Below diagram depicts the architecture with weights –

RNN WITH WEIGHTS



So from above figure we can write below equations-

$$h_t = f\left(W_h^{\,T} h_{t-1} + W_x^{\,T} X_t + b_h \right)$$

f= Sigmoid , tanh , ReLu

Note:- Function f could be any one of the usual hidden non-linearities that's usually sigmoid , tanh or ReLu. It's a hyper parameter just like other types of Neural networks .

This feedback loop makes recurrent neural networks seem kind of mysterious and quite hard to visualize the whole training process of RNNs. So let's unfold this Recurrent neural to understand its working .

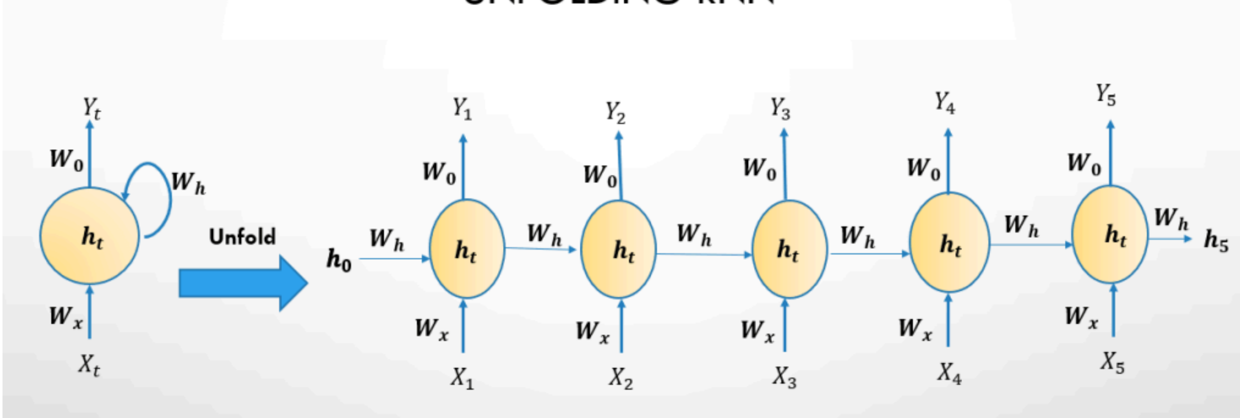
Unfolding a Recurrent Neural Network:-

Here we'd try to visualize the RNNs in terms of a feedforward network.

A recurrent neural network can be thought of as multiple copies of a feedforward network network, each passing a message to a successor.

Now consider what happens if we unroll the loop: Imagine we've a sequence of length 5 , if we were to unfold the recurrent neural network in time such that it has no recurrent connections at all then we get this feedforward neural network with 5 hidden layers like shown in below figure.

UNFOLDING RNN



It is as if h_0 is the input and each X_t is just some additional control signal at each step. We can see that hidden to hidden weight W_h is just repeated at every layer. So it's like a deep network with the same shared weights between each layer. Similarly W_x is shared between each of the five Xs going into hidden layers.

Where ,

h_t – *Current state i.e. state at time step t*

h_{t-1} – previous state i.e. state at time step t – 1

X_t – *Input at time step 6*

W_h – *Weight at recurrent neuron*

W_x – *Weight at input neuron*

$$Y_t \quad - \quad Output\ at\ time\ step\ t$$

How Recurrent Neural Network Works?

In Recurrent Neural networks, the information cycles through a loop to the middle hidden layer. The input layer 'x' takes in the input to the neural network and processes it and passes it onto the middle layer.

The middle layer 'h' can consist of multiple hidden layers, each with its own activation functions and weights and biases. If you have a neural network where the various parameters of different hidden layers are not affected by the previous layer, ie: the neural network does not have memory, then you can use a recurrent neural network.

The Recurrent Neural Network will standardize the different activation functions and weights and biases so that each hidden layer has the same parameters. Then, instead of creating multiple hidden layers, it will create one and loop over it as many times as required.

Types of Recurrent Neural Networks

1.One-to-One: The simplest type of RNN is One-to-One, which allows a single input and a single output. It has fixed input and output sizes and acts as a traditional neural network. The One-to-One application can be found in Image Classification.

2. One-to-Many: One-to-Many is a type of RNN that gives multiple outputs when given a single input. It takes a fixed input size and gives a sequence of data outputs. Its applications can be found in *Music Generation* and *Image Captioning*.

3. Many-to-One: Many-to-One is used when a single output is required from multiple input units or a sequence of them. It takes a sequence of inputs to display a fixed output. Sentiment Analysis is a common example of this type of Recurrent Neural Network.

4. Many-to-Many: Many-to-Many is used to generate a sequence of output data from a sequence of input units.

This type of RNN is further divided into the following two subcategories:

1. Equal Unit Size: In this case, the number of both the input and output units is the same. A common application can be found in Name-Entity Recognition.

2. Unequal Unit Size: In this case, inputs and outputs have different numbers of units. Its application can be found in Machine Translation.

Applications of Recurrent Neural Networks

1. **Language Modelling and Generating Text**

Taking a sequence of words as input, we try to predict the possibility of the next word. This can be considered to be one of the most useful approaches for translation since the most likely sentence would be the one that is correct.In this method, the probability of the output of a particular time-step is used to sample the words in the next iteration.

2. **Machine Translation**

RNNs in one form or the other can be used for translating text from one language to other . Almost all of the Translation systems being used today use some advanced version of a RNN. The input can be the source language and the output will be in the target language which the user wants.

3. **Speech Recognition**

RNNs can be used for predicting phonetic segments considering sound waves from a medium as an input source .The set of inputs consists of phoneme or acoustic signals from an audio which are processed in a proper manner and taken as inputs. The RNN network will compute the phonemes and then produce a phonetic segment along with the likelihood of output.

4. **Generating Image Descriptions**

A combination of CNNs and RNNs are used to provide a description of what exactly is happening inside an image. CNN does the segmentation part and RNN then uses the segmented data to recreate the description.

5. **Text Summarization**

This application can provide major help in summarizing content from literatures and customising them for delivery within applications which cannot support large volumes of text. For example, if a publisher wants to display the summary of one of his books on its back page to help the readers get an idea of the content present within, Text Summarization would be helpful.

RNN Intuition

A class of neural networks for dealing with sequential data problems can be achieved by recurrent neural network. It works exceptionally well with natural language processing problems. Within NLP, the recurrent varied tasks like language translation, classification of texts and text generation automatically.

Tackling Vanishing Gradient Problem:

There are many ways to deal with the vanishing gradient problem. Ex: Alternate Weight initialization, layer-wise training etc. However, the most recent advancement in this issue fix is

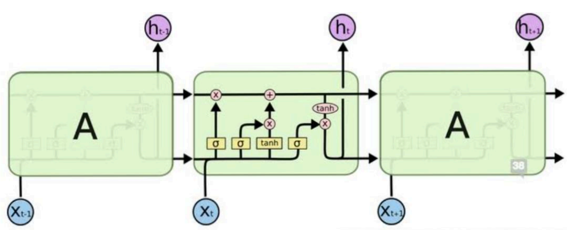
using the ReLU activation function instead of the Tangent activation function. The ReLU activation function enables the network to collect a sparse matrix representation.
For example, 50% of the hidden node's output is zeros after initializing the weights. Moreover this can increase with regularization.

Long-Short-Term Memory:

LSTMs, also called the Long Short Term Memory networks are particular Recurrent neural networks that can resolve long-term dependencies. For example: Let us use a sentence: The man who rode the bike had blonde hair. In this example, blonde is the hair color and not the bike is a long term dependency.
LSTMs, are specifically used to avoid long term dependencies. Remember information for a more significant period. The recurrent neural networks take a form of a chain of repeating modules of the neural network. The repeating module has a straightforward structure in typical recurrent neural networks, known as the tanh layer.

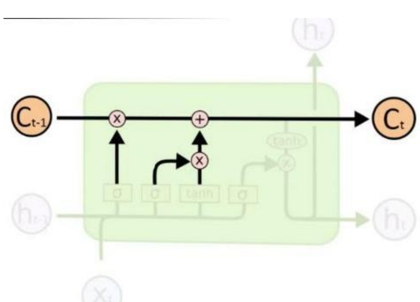
Building a RNN:

The LSTMs have a chain-like structure, and the repeating module has quite a different design. The module has four neural network layers instead of one and interacts uniquely.



Evaluating the RNN:

The cell state in LSTMs plays a vital role, represented by a straight line in the figure below. The cell state runs around the entire length of the chain with only some minor linear interactions.



The information flow through this is very smooth.
The LSTM, along with the help of gates (a structure), adds or removes information to the cell State. Then, gates let the information flow through.

CNN vs RNN:

S. No	CNN	RNN
1.	CNN stands for Convolutional Neural Network**.**	RNN stands for Recurrent Neural Network.
2.	CNN is considered to be more potent than RNN**.**	RNN includes less feature compatibility when compared to CNN.
3.	CNN is ideal for images and video processing**.**	RNN is ideal for text and speech Analysis
4.	It is suitable for spatial data like images.	RNN is used for temporal data, also called sequential data
5.	The network takes fixed-size inputs and generates fixed size outputs.	RNN can handle arbitrary input/ output lengths
6.	CNN is a type of feed-forward artificial neural network with variations of multilayer perceptron's designed to use minimal amounts of preprocessing.	RNN, unlike feed-forward neural networks- can use their internal memory to process arbitrary sequences of inputs
7.	CNN's use of connectivity patterns between the neurons. CNN is affected by the organization of the animal visual cortex, whose individual neurons are arranged in such a way that they can respond to overlapping regions in the visual field..	Recurrent neural networks use time-series information- what a user spoke last would impact what he will speak next

Data collection strategy for ML:

Collecting data is the first step after data identification in any Deep Learning-based solution The quality of the data collected is very important as the Machine Learning predictions are accurate only when the Quality of the data is good. Typically find many issues with the data's collected.

• Inaccuracies in the collected dataset: The collected data may not be relevant to the problem. Also, there are chances that incorrect entries are present in the dataset

Missing values in the dataset: The data collected may be value missing for all fields or columns present in the dataset.

• The Data might be Imbalanced: The data collected may be highly imbalanced, meaning that one set of categorical samples might be proportionately higher than the other.

How much data is needed?

• Complexity of the problem

It is a difficult question to answer. The quantity of data required directly depends on the Type of algorithm chosen for implementation and the problem's difficulty level, Knowing How much data is enough is an intractable problem and can only be answered by a Detailed investigation.

- Domain Expertise

While solving machine learning problems, typically mag an input function to an output Function. However, the mapping function works well only when providing quality data to The model. Hence, domain expertise is necessary to extract the important complexity in the Problem reasonably.

- Statistical Heuristic methods

Some heuristic statistical methods may help determine the model's data sample size.
Consider the following:
1.importance of the number of Classes
2. Importance of the number of features.
3. Importance of the number of model parameters

• Collect More Data

While working on traditional Machine Learning predictive modelling, the training data size will Have proportionally smaller benefits. Therefore, it is critical to analyse the problem and the Selected model to find out where that is happening.
Do not stop modelling, continue
The issue of deciding the amount of data needed must not stop from proceeding further and Starting the modelling process. So go ahead and initiate the model-building process.
Experience with the model will help to analyse the situation better, to augment data, go Ahead and augment the data or collect more data.

Is the data good enough?:
Well, this question can sometimes be easier to answer. Bad data does have any values and Produces bad results. Therefore, poor quality data in a machine learning system will make Bas predictions. However, one can turn insufficient quality data into good quality data.
Bad quality data means :
•The Data is inconsistent
The Data is inaccessible
The Data is unavailable
Rule Based mechanisms are carried out to take care of data quality issues as shown below:
Usually, Machine Learning solutions are employed to care for data quality concerns,
Poor Schema

• Redundant Data
Delayed arrival of Data
Data with anomalies
Dimensionality Reduction: Dimensionality Reduction reduces the number of input variables
In a given dataset by finding each column’s importance.
Clustering: Clustering usually organizes the data into a group of clusters or segments based
On identified patterns or similarities.
Anomaly Detection: Anomaly detection identifies the data points that do not fit into the Concerned group or away from the points.

Applications of CNN:

1. **Decoding Facial Recognition**

   Facial recognition is broken down by a convolutional neural network into the following major components -

   - Identifying every face in the picture
   - Focusing on each face despite external factors, such as light, angle, pose, etc.
   - Identifying unique features
   - Comparing all the collected data with already existing data in the database to match a face with a name.

   A similar process is followed for scene labeling as well.

2. **Analysing Documents**

   Convolutional neural networks can also be used for document analysis. This is not just useful for handwriting analysis, but also has a major stake in recognizers. For a machine to be able to scan an individual's writing, and then compare that to the wide database it has, it must execute almost a million commands a minute. It is said with the use of CNNs and newer models and algorithms, the error rate has been brought down to a minimum of 0.4% at a character level, though it's complete testing is yet to be widely seen.

3. **Collecting Historic and Environmental Elements**

CNNs are also used for more complex purposes such as natural history collections. These collections act as key players in documenting major parts of history such as biodiversity, evolution, habitat loss, biological invasion, and climate change.

4. **Understanding Climate**

   CNNs can be used to play a major role in the fight against climate change, especially in understanding the reasons why we see such drastic changes and how we could experiment in curbing the effect. It is said that the data in such natural history collections can also provide greater social and scientific insights, but this would require skilled human resources such as researchers who can physically visit these types of repositories. There is a need for more manpower to carry out deeper experiments in this field.

5. **Understanding Gray Areas**

   Introduction of the gray area into CNNs is posed to provide a much more realistic picture of the real world. Currently, CNNs largely function exactly like a machine, seeing a true and false value for every question. However, as humans, we understand that the real world plays out in a thousand shades of gray. Allowing the machine to understand and process fuzzier logic will help it understand the gray area we humans live in and strive to work against. This will help CNNs get a more holistic view of what human sees.

6. **Advertising**

   CNNs have already brought a world of difference to advertising with the introduction of programmatic buying and data-driven personalized advertising.

Convolution Neural Network – Concepts

Convolution layer: A convolutional layer is the main building block of a CNN. It contains a set of filters (or kernels), parameters of which are to be learned throughout the training. The size of the filters is usually smaller than the actual image. Each filter convolves with the image and creates an activation map.

ReLU Layer: In this layer we remove every negative value from the filtered image and replace it with zero. This function only activates when the node input is above a certain quantity. So, when the input is below zero the output is zero. However, when the input rises above a certain threshold it has *linear relationship* with the dependent variable. *This means* that it is able *to accelerate the speed of* a training data set in a deep neural network that is faster than other activation functions – this is done to avoid summing up with zero.

Pooling Layer: The main purpose of pooling layer is to progressively reduce the spatial size of the input image, so that number of computations in the network are reduced. Pooling performs downs sampling by reducing the size and sends only the important data to next layers in CNN. There are 3 types of pooling they are,

- **Max Pooling**: This type of pooling is the most significant element of the feature map.
- **Average Pooling:** In this type of pooling, the average of element is calculated.
- **Sum Pooling:** In this type of pooling, some of the elements in the pre defined section is calculated.

Flattening Layer: Flattening is converting the data into a 1-dimensional array for inputting it to the next layer. We flatten the output of the convolutional layers to create a single long feature vector. And it is connected to the final classification model, which is called a fully-connected layer.

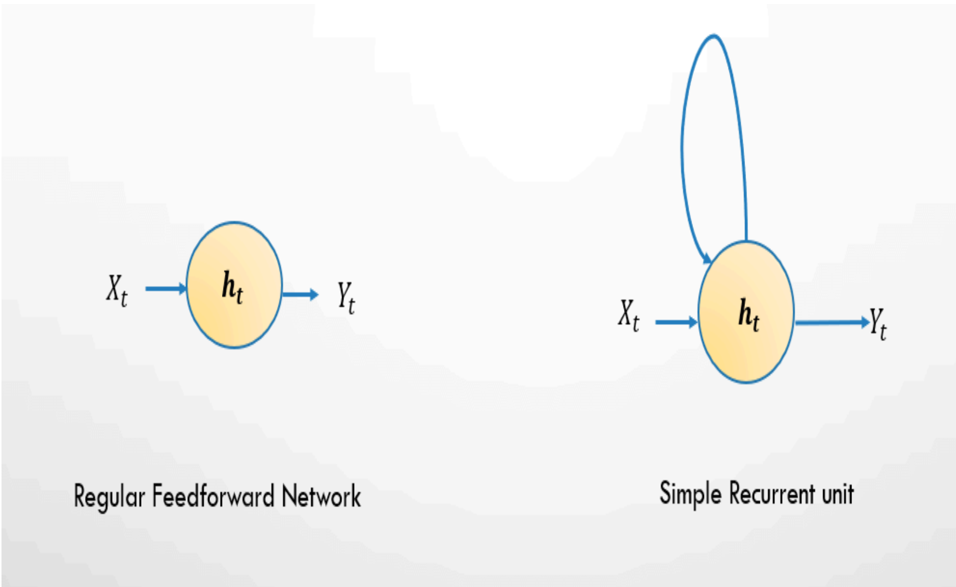
Full conversion layer: It is the fully connected layer where the artificial and the convolution neural network intersect. As a result, the web gets complex and sophisticated in this layer. In artificial neural network, the layer sandwiched between input and the output layers is the hidden layer, where it is fully connected layer in the convolution. Once the flattening layer is complete, we get a vector of input data used to pass the network for us to process further.

Soft max: The soft max function is used as the activation function in the output layer of neural network models that predict a multinomial probability distribution. That is, softmax is used as the activation function for multi-class classification problems where class membership is required on more than two class labels.

Cross Entropy: Cross-entropy is a measure of the difference between two probability distributions for a given random variable or set of events. You might recall that information quantifies the number of bits required to encode and transmit an event.

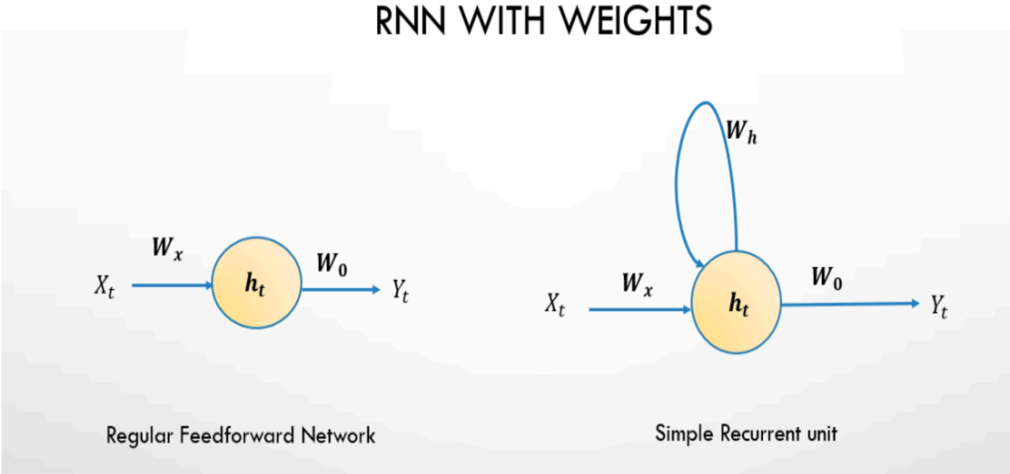
Recurrent Neural Network Architecture:

The right diagram in below figure in below figure represents a simple Recurrent unit.



Below diagram depicts the architecture with weights –

RNN WITH WEIGHTS



So from above figure we can write below equations-

$$h_t = f\left(W_h^T h_{t-1} + W_x^T X_t + b_h \right)$$

f= Sigmoid , tanh , ReLu

Note:- Function f could be any one of the usual hidden non-linearities that's usually sigmoid , tanh or ReLu. It's a hyper parameter just like other types of Neural networks .

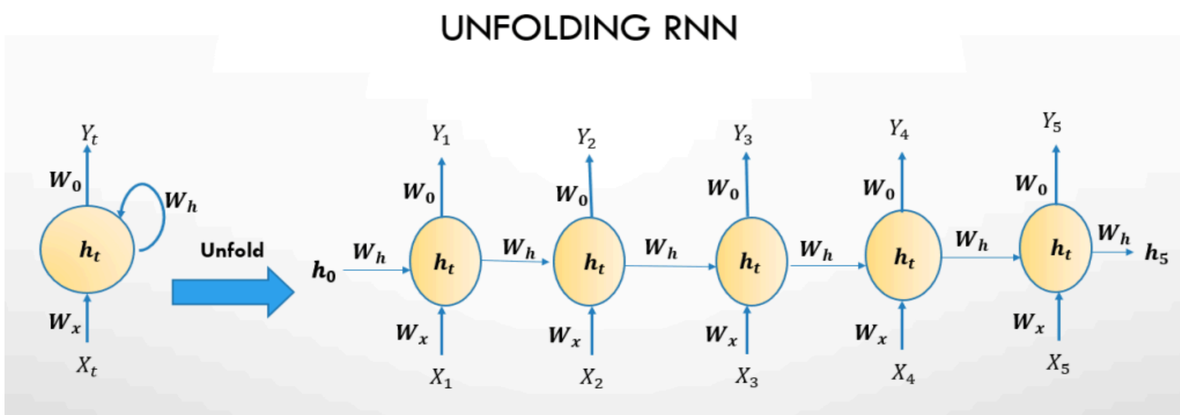
This feedback loop makes recurrent neural networks seem kind of mysterious and quite hard to visualize the whole training process of RNNs. So let's unfold this Recurrent neural to understand its working .

Unfolding a Recurrent Neural Network:-

Here we'd try to visualize the RNNs in terms of a feedforward network.

A recurrent neural network can be thought of as multiple copies of a feedforward network network, each passing a message to a successor.

Now consider what happens if we unroll the loop: Imagine we've a sequence of length 5 , if we were to unfold the recurrent neural network in time such that it has no recurrent connections at all then we get this feedforward neural network with 5 hidden layers like shown in below figure.



It is as if h_0 is the input and each X_t is just some additional control signal at each step. We can see that hidden to hidden weight W_h is just repeated at every layer. So it's like a deep network with the same shared weights between each layer. Similarly W_x is shared between each of the five Xs going into hidden layers.

Where ,

h_t – Current state i.e. state at time step t
h_{t-1} – previous state i.e. state at time step t – 1
X_t – Input at time step 6
W_h – Weight at recurrent neuron
W_x – Weight at input neuron
Y_t – Output at time step t

How Recurrent Neural Network Works?

In Recurrent Neural networks, the information cycles through a loop to the middle hidden layer. The input layer 'x' takes in the input to the neural network and processes it and passes it onto the middle layer.

The middle layer 'h' can consist of multiple hidden layers, each with its own activation functions and weights and biases. If you have a neural network where the various parameters of different hidden layers are not affected by the previous layer, ie: the neural network does not have memory, then you can use a recurrent neural network.

The Recurrent Neural Network will standardize the different activation functions and weights and biases so that each hidden layer has the same parameters. Then, instead of creating multiple hidden layers, it will create one and loop over it as many times as required.

Types of Recurrent Neural Networks

1.One-to-One: The simplest type of RNN is One-to-One, which allows a single input and a single output. It has fixed input and output sizes and acts as a traditional neural network. The One-to-One application can be found in Image Classification.

2. One-to-Many: One-to-Many is a type of RNN that gives multiple outputs when given a single input. It takes a fixed input size and gives a sequence of data outputs. Its applications can be found in *Music Generation* and *Image Captioning*.

3. Many-to-One: Many-to-One is used when a single output is required from multiple input units or a sequence of them. It takes a sequence of inputs to display a fixed output. Sentiment Analysis is a common example of this type of Recurrent Neural Network.

4. Many-to-Many: Many-to-Many is used to generate a sequence of output data from a sequence of input units.

This type of RNN is further divided into the following two subcategories:

1. Equal Unit Size: In this case, the number of both the input and output units is the same. A common application can be found in Name-Entity Recognition.

2. Unequal Unit Size: In this case, inputs and outputs have different numbers of units. Its application can be found in Machine Translation.

Applications of Recurrent Neural Networks

3. **Language Modelling and Generating Text**

Taking a sequence of words as input, we try to predict the possibility of the next word. This can be considered to be one of the most useful approaches for translation since the most likely sentence would be the one that is correct.In this method, the probability of the output of a particular time-step is used to sample the words in the next iteration.

4. **Machine Translation**

RNNs in one form or the other can be used for translating text from one language to other . Almost all of the Translation systems being used today use some advanced version of a RNN. The input can be the source language and the output will be in the target language which the user wants.

4. **Speech Recognition**

RNNs can be used for predicting phonetic segments considering sound waves from a medium as an input source .The set of inputs consists of phoneme or acoustic signals from an audio which are processed in a proper manner and taken as inputs. The RNN network will compute the phonemes and then produce a phonetic segment along with the likelihood of output.

6. **Generating Image Descriptions**

A combination of CNNs and RNNs are used to provide a description of what exactly is happening inside an image. CNN does the segmentation part and RNN then uses the segmented data to recreate the description.

7. **Text Summarization**

This application can provide major help in summarizing content from literatures and customising them for delivery within applications which cannot support large volumes of text. For example, if a publisher wants to display the summary of one of his books on its back page to help the readers get an idea of the content present within, Text Summarization would be helpful.

RNN Intuition

A class of neural networks for dealing with sequential data problems can be achieved by recurrent neural network. It works exceptionally well with natural language processing problems. Within NLP, the recurrent varied tasks like language translation, classification of texts and text generation automatically.

Tackling Vanishing Gradient Problem:

There are many ways to deal with the vanishing gradient problem. Ex: Alternate Weight initialization, layer-wise training etc. However, the most recent advancement in this issue fix is using the ReLU activation function instead of the Tangent activation function. The ReLU activation function enables the network to collect a sparse matrix representation.

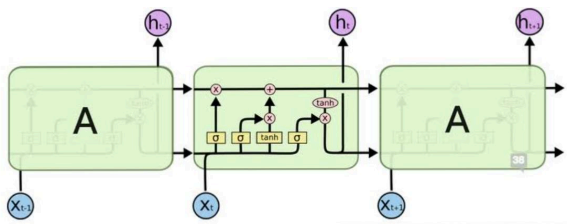
For example, 50% of the hidden node's output is zeros after initializing the weights. Moreover this can increase with regularization.

Long-Short-Term Memory:

LSTMs, also called the Long Short Term Memory networks are particular Recurrent neural networks that can resolve long-term dependencies. For example: Let us use a sentence: The man who rode the bike had blonde hair. In this example, blonde is the hair color and not the bike is a long term dependency.
LSTMs, are specifically used to avoid long term dependencies. Remember information for a more significant period. The recurrent neural networks take a form of a chain of repeating modules of the neural network. The repeating module has a straightforward structure in typical recurrent neural networks, known as the tanh layer.

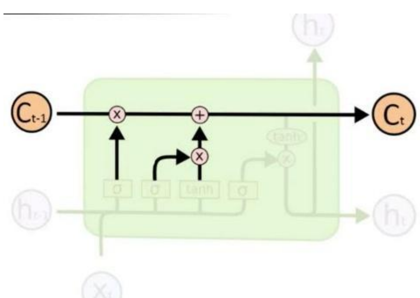
Building a RNN:

The LSTMs have a chain-like structure, and the repeating module has quite a different design. The module has four neural network layers instead of one and interacts uniquely.



Evaluating the RNN:

The cell state in LSTMs plays a vital role, represented by a straight line in the figure below. The cell state runs around the entire length of the chain with only some minor linear interactions.



The information flow through this is very smooth.
The LSTM, along with the help of gates (a structure), adds or removes information to the cell State. Then, gates let the information flow through.

CNN vs RNN:

S. No	CNN	RNN
1.	CNN stands for Convolutional Neural Network**.**	RNN stands for Recurrent Neural Network.

2.	CNN is considered to be more potent than RNN**.**	RNN includes less feature compatibility when compared to CNN.
3.	CNN is ideal for images and video processing**.**	RNN is ideal for text and speech Analysis
4.	It is suitable for spatial data like images.	RNN is used for temporal data, also called sequential data
5.	The network takes fixed-size inputs and generates fixed size outputs.	RNN can handle arbitrary input/ output lengths
6.	CNN is a type of feed-forward artificial neural network with variations of multilayer perceptron's designed to use minimal amounts of preprocessing.	RNN, unlike feed-forward neural networks- can use their internal memory to process arbitrary sequences of inputs
7.	CNN's use of connectivity patterns between the neurons. CNN is affected by the organization of the animal visual cortex, whose individual neurons are arranged in such a way that they can respond to overlapping regions in the visual field..	Recurrent neural networks use time-series information- what a user spoke last would impact what he will speak next

Data collection strategy for ML:

Collecting data is the first step after data identification in any Deep Learning-based solution The quality of the data collected is very important as the Machine Learning predictions are accurate only when the Quality of the data is good. Typically find many issues with the data's collected.

• Inaccuracies in the collected dataset: The collected data may not be relevant to the problem. Also, there are chances that incorrect entries are present in the dataset

Missing values in the dataset: The data collected may be value missing for all fields or columns present in the dataset.

• The Data might be Imbalanced: The data collected may be highly imbalanced, meaning that one set of categorical samples might be proportionately higher than the other.

How much data is needed?

• Complexity of the problem

It is a difficult question to answer. The quantity of data required directly depends on the Type of algorithm chosen for implementation and the problem's difficulty level, Knowing How much data is enough is an intractable problem and can only be answered by a Detailed investigation.

- Domain Expertise

While solving machine learning problems, typically mag an input function to an output Function. However, the mapping function works well only when providing quality data to The model. Hence, domain expertise is necessary to extract the important complexity in the Problem reasonably.

- Statistical Heuristic methods

Some heuristic statistical methods may help determine the model's data sample size.
Consider the following:
1.importance of the number of Classes
2. Importance of the number of features.
3. Importance of the number of model parameters

- Collect More Data

While working on traditional Machine Learning predictive modelling, the training data size will Have proportionally smaller benefits. Therefore, it is critical to analyse the problem and the Selected model to find out where that is happening.
Do not stop modelling, continue
The issue of deciding the amount of data needed must not stop from proceeding further and Starting the modelling process. So go ahead and initiate the model-building process.
Experience with the model will help to analyse the situation better, to augment data, go Ahead and augment the data or collect more data.

Is the data good enough?:
Well, this question can sometimes be easier to answer. Bad data does have any values and Produces bad results. Therefore, poor quality data in a machine learning system will make Bas predictions. However, one can turn insufficient quality data into good quality data.
Bad quality data means :
•The Data is inconsistent
The Data is inaccessible
The Data is unavailable
Rule Based mechanisms are carried out to take care of data quality issues as shown below:
Usually, Machine Learning solutions are employed to care for data quality concerns,
Poor Schema
• Redundant Data
Delayed arrival of Data
Data with anomalies
Dimensionality Reduction: Dimensionality Reduction reduces the number of input variables In a given dataset by finding each column's importance.
Clustering: Clustering usually organizes the data into a group of clusters or segments based

On identified patterns or similarities.

Anomaly Detection: Anomaly detection identifies the data points that do not fit into the Concerned group or away from the points.