1. Supervised and Unsupervised learning

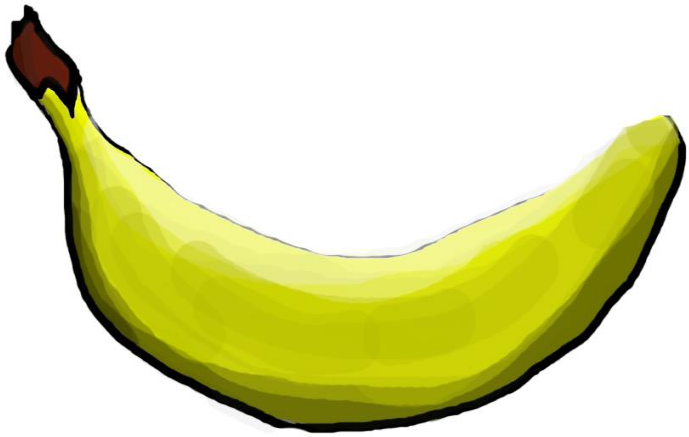
Supervised learning:

- Supervised learning, as the name indicates, has the presence of a supervisor as a teacher. Basically supervised learning is when we teach or train the machine using data that is well labelled. Which means some data is already tagged with the correct answer. After that, the machine is provided with a new set of examples(data) so that the supervised learning algorithm analyses the training data(set of training examples) and produces a correct outcome from labelled data.

For instance, suppose you are given a basket filled with different kinds of fruits. Now the first step is to train the machine with all the different fruits one by one like this:

- If the shape of the object is rounded and has a depression at the top, is red in color, then it will be labeled as –**Apple**.
- If the shape of the object is a long curving cylinder having Green-Yellow color, then it will be labeled as –**Banana**.

❖ Now suppose after training the data, you have given a new separate fruit, say Banana from the basket, and asked to identify it.



- Since the machine has already learned the things from previous data and this time has to use it wisely. It will first classify the fruit with its shape and color and would confirm the fruit name as BANANA and put it in the Banana category. Thus the machine learns the things from training data(basket containing fruits) and then applies the knowledge to test data(new fruit).

- **Supervised learning is classified into two categories of algorithms:**

    - **Classification**: A classification problem is when the output variable is a category, such as “Red” or “blue” , “disease” or “no disease”.
    - **Regression**: A regression problem is when the output variable is a real value, such as “dollars” or “weight”.

Supervised learning deals with or learns with “labeled” data. This implies that some data is already tagged with the correct answer.

Types:-

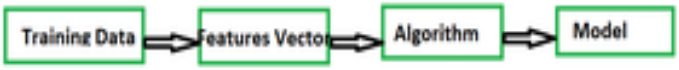
- Regression
- Logistic Regression
- Classification
- Naive Bayes Classifiers
- K-NN (k nearest neighbors)
- Decision Trees
- Support Vector Machine

Advantages:-

- Supervised learning allows collecting data and produces data output from previous experiences.
- Helps to optimize performance criteria with the help of experience.
- Supervised machine learning helps to solve various types of real-world computation problems.

Disadvantages:-

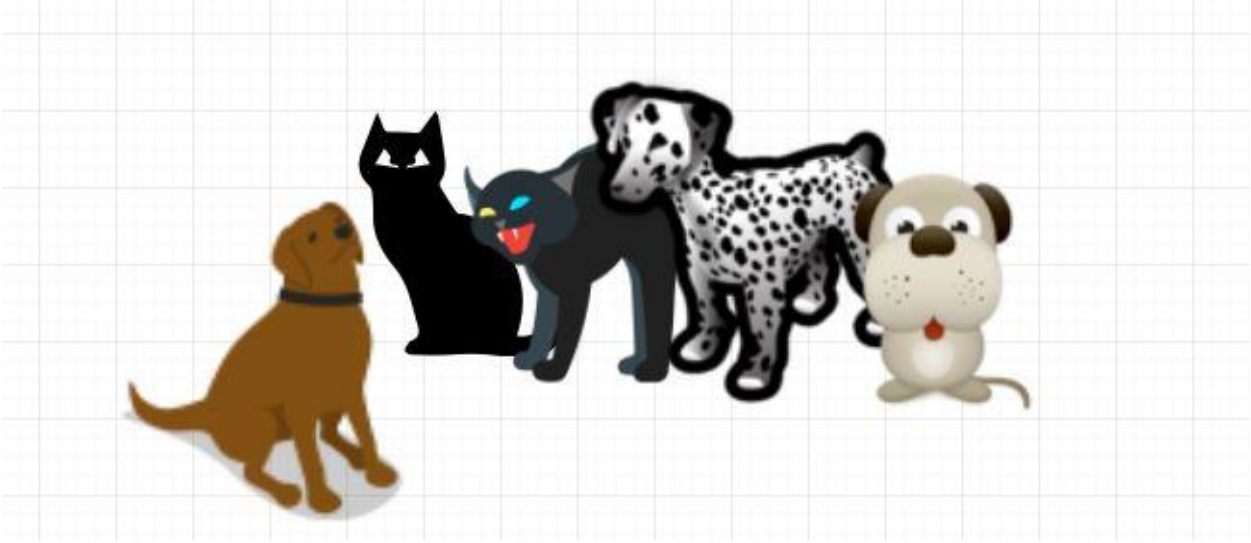
- Classifying big data can be challenging.
- Training for supervised learning needs a lot of computation time. So, it requires a lot of time.



Unsupervised learning:

- Unsupervised learning is the training of a machine using information that is neither classified nor labeled and allowing the algorithm to act on that information without guidance. Here the task of the machine is to group unsorted information according to similarities, patterns, and differences without any prior training of data.
- Unlike supervised learning, no teacher is provided that means no training will be given to the machine. Therefore the machine is restricted to find the hidden structure in unlabeled data by itself.

For instance, suppose it is given an image having both dogs and cats which it has never seen.



Thus the machine has no idea about the features of dogs and cats so we can't categorize it as 'dogs and cats '. But it can categorize them according to their similarities, patterns, and differences, i.e., we can easily categorize the above picture into two parts. The first may contain all pics having **dogs** in them and the second part may contain all pics having **cats** in them. Here you didn't learn anything before, which means no training data or examples.

Unsupervised learning is classified into two categories of algorithms:

- **Clustering**: A clustering problem is where you want to discover the inherent groupings in the data, such as grouping customers by purchasing behavior.
- **Association**: An association rule learning problem is where you want to discover rules that describe large portions of your data, such as people that buy X also tend to buy Y.

Clustering:

1. Exclusive (partitioning)
2. Agglomerative
3. Overlapping
4. Probabilistic

Clustering Types:-

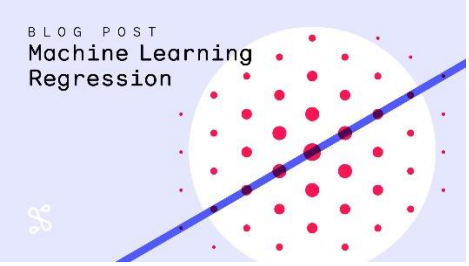
1. Hierarchical clustering
2. K-means clustering
3. Principal Component Analysis
4. Singular Value Decomposition
5. Independent Component Analysis

Supervised vs. Unsupervised Machine Learning:

Parameters	Supervised machine learning	Unsupervised machine learning
Input Data	Algorithms are trained using labeled data.	Algorithms are used against data that is not labeled
Computational Complexity	Simpler method	Computationally complex
Accuracy	Highly accurate	Less accurate
No. of classes	No. of classes is known	No. of classes is not known
Data Analysis	Uses offline analysis	Uses real-time analysis of data
Algorithms used	Linear and Logistics regression, Random forest, Support Vector Machine, Neural Network, etc.	K-Means clustering, Hierarchical clustering, Apriori algorithm, etc.

2.REGRESSION CLASSIFICATION AND CLUSTERING:

- Regression is **a technique for investigating the relationship between independent variables or features and a dependent variable or outcome**. It's used as a method for predictive modelling in machine learning, in which an algorithm is used to predict continuous outcomes



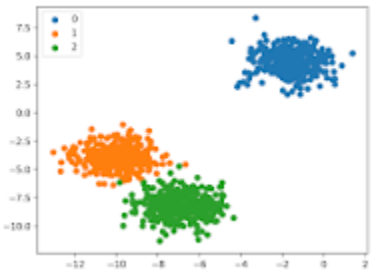
regression models:

- Regression analysis includes several variations, such as **linear, multiple linear, and nonlinear**. The most common models are simple linear and multiple linear. Nonlinear regression analysis is commonly used for more complicated data sets in which the dependent and independent variables show a nonlinear relationship.

Regression applications:

- Regression analysis is used **to estimate the relationship between a dependent variable and one or more independent variables**. This technique is widely applied to predict the outputs, forecasting the data, analyzing the time series, and finding the causal effect dependencies between the variables

classification problem :

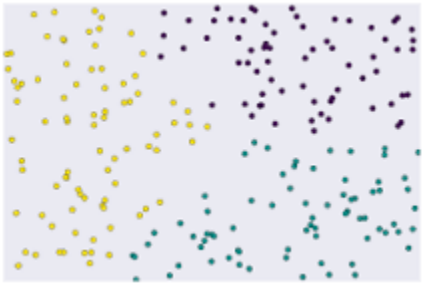


- In machine learning, classification refers to **a predictive modeling problem where a class label is predicted for a given example of input data**. Examples of classification problems include: Given an example, classify if it is spam or not. Given a handwritten character, classify it as one of the known characters.

CLASSIFICATION APPLICATION:

- Classification algorithms in machine learning use input training data to predict the likelihood that subsequent data will fall into one of the predetermined categories. One of the most common uses of classification is **filtering emails into "spam" or "non-spam."**

clustering :



- In machine learning too, we often group examples as a first step to understand a subject (data set) in a machine learning system. **Grouping unlabeled examples** is called clustering. As the examples are unlabeled, clustering relies on unsupervised machine learning.

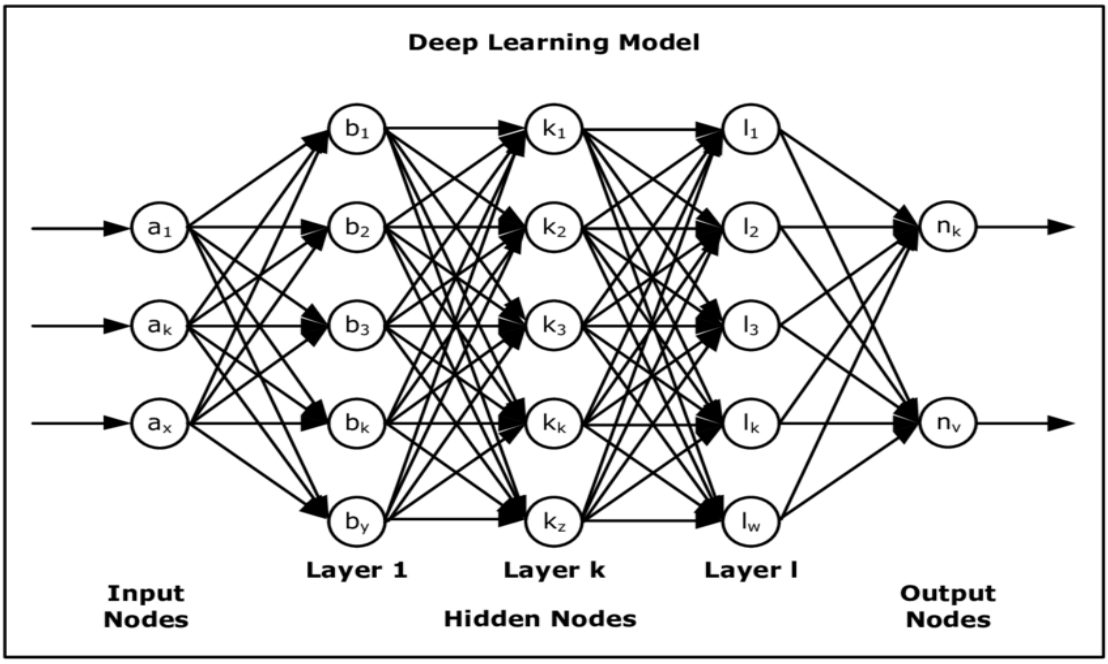
(a) Learning Phase

(b) Identification Phase

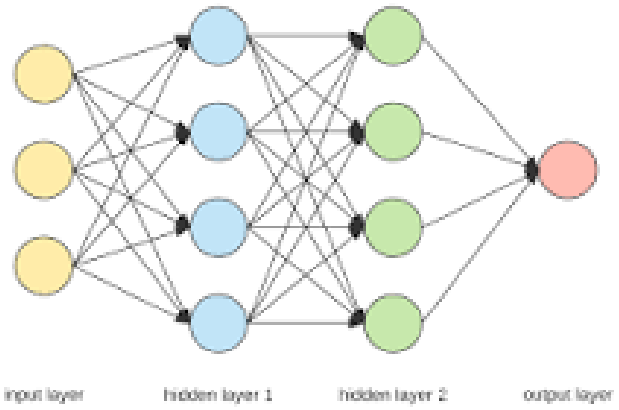
3.ARTIFICIAL NEURAL NETWORKS:(DEEP LEARNING)

- A neural network is **a method in artificial intelligence that teaches computers to process data in a way that is inspired by the human brain**. It is a type of machine learning process,

called deep learning, that uses interconnected nodes or neurons in a layered structure that resembles the human brain.



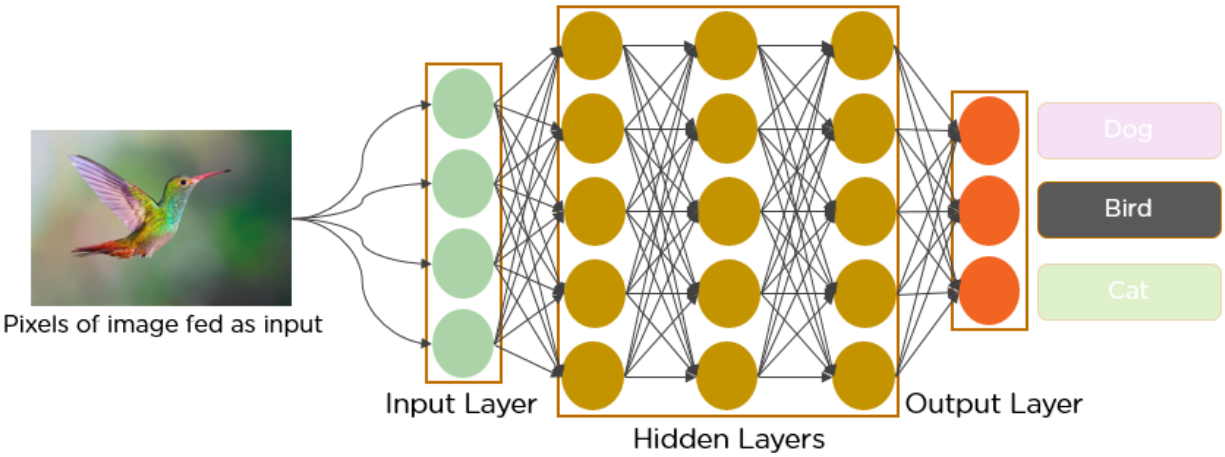
ANN :



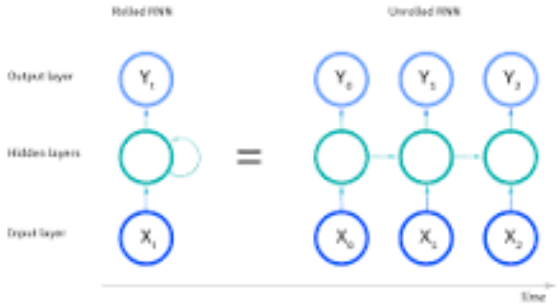
- Artificial Neural Networks (ANN) are **multi-layer fully-connected neural nets** that look like the figure below. They consist of an input layer, multiple hidden layers, and an output layer. Every node in one layer is connected to every other node in the next layer.

CNN :

- A CNN is **a kind of network architecture for deep learning algorithms and is specifically used for image recognition and tasks that involve the processing of pixel data**. There are other types of neural networks in deep learning, but for identifying and recognizing objects, CNNs are the network architecture of choice.



RNN :

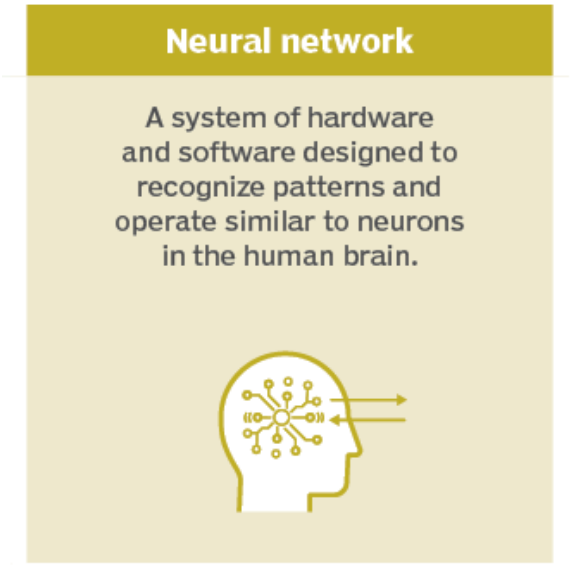


- A recurrent neural network (RNN) is **a type of artificial neural network which uses sequential data or time series data**.

4.convolutional neural networks:

- Convolutional neural network, a subset of machine learning, is **a type of artificial neural network**.

- To reiterate from the Neural Networks Learn Hub article, neural networks are a subset of machine learning, and they are at the heart of deep learning algorithms. They are comprised of node layers, containing an input layer, one or more hidden layers, and an output layer. Each node connects to another and has an associated weight and threshold. If the output of any individual node is above the specified threshold value, that node is activated, sending data to the next layer of the network. Otherwise, no data is passed along to the next layer of the network.



How do convolutional neural networks work:

- Convolutional neural networks are distinguished from other neural networks by their superior performance with image, speech, or audio signal inputs. They have three main types of layers, which are:

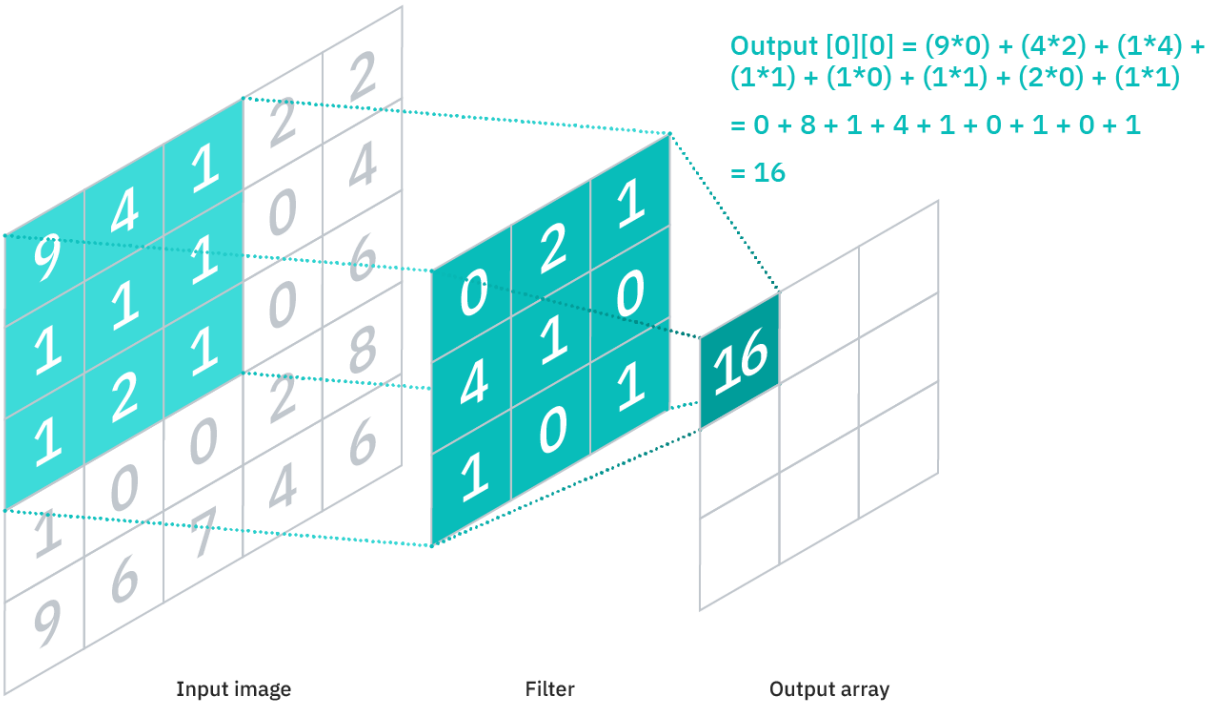
  - Convolutional layer
  - Pooling layer
  - Fully-connected (FC) layer

- The convolutional layer is the first layer of a convolutional network. While convolutional layers can be followed by additional convolutional layers or pooling layers, the fully-connected layer is the final layer. With each layer, the CNN increases in its complexity, identifying greater portions of the image. Earlier layers focus on simple features, such as colors and edges. As the image data progresses through the layers of the CNN, it starts to recognize larger elements or shapes of the object until it finally identifies the intended object.

Convolutional Layer:

The feature detector is a two-dimensional (2-D) array of weights, which represents part of the image. While they can vary in size, the filter size is typically a 3x3 matrix; this also determines the size of the receptive field. The filter is then applied to an area of the image, and a dot product

is calculated between the input pixels and the filter. This dot product is then fed into an output array. Afterwards, the filter shifts by a stride, repeating the process until the kernel has swept across the entire image. The final output from the series of dot products from the input and the filter is known as a feature map, activation map, or a convolved feature.



❖ As you can see in the image above, each output value in the feature map does not have to connect to each pixel value in the input image. It only needs to connect to the receptive field, where the filter is being applied. Since the output array does not need to map directly to each input value, convolutional (and pooling) layers are commonly referred to as "partially connected" layers. However, this characteristic can also be described as local connectivity.

- Note that the weights in the feature detector remain fixed as it moves across the image, which is also known as parameter sharing. Some parameters, like the weight values, adjust during training through the process of backpropagation and gradient descent. However, there are three hyperparameters which affect the volume size of the output that need to be set before the training of the neural network begins. These include:

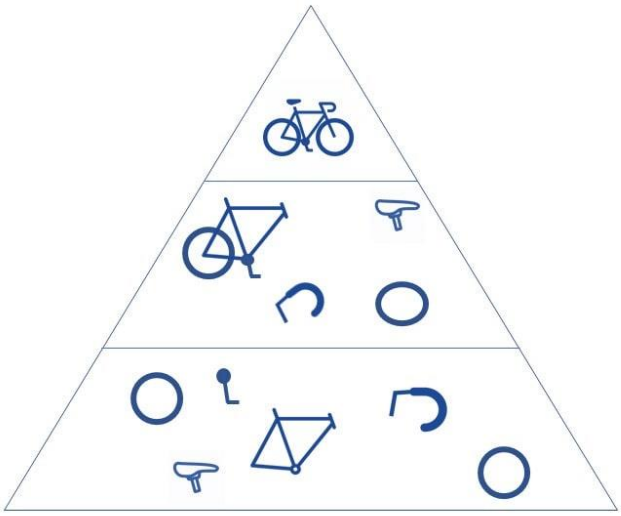
1. The **number of filters** affects the depth of the output. For example, three distinct filters would yield three different feature maps, creating a depth of three.

2. **Stride** is the distance, or number of pixels, that the kernel moves over the input matrix. While stride values of two or greater is rare, a larger stride yields a smaller output.

3. **Zero-padding** is usually used when the filters do not fit the input image. This sets all elements that fall outside of the input matrix to zero, producing a larger or equally sized output. There are three types of padding:

- **Valid padding:** This is also known as no padding. In this case, the last convolution is dropped if dimensions do not align.
- **Same padding:** This padding ensures that the output layer has the same size as the input layer
- **Full padding:** This type of padding increases the size of the output by adding zeros to the border of the input.

After each convolution operation, a CNN applies a Rectified Linear Unit (ReLU) transformation to the feature map, introducing nonlinearity to the model.



Ultimately, the convolutional layer converts the image into numerical values, allowing the neural network to interpret and extract relevant patterns.

Pooling Layer:

❖ Pooling layers, also known as downsampling, conducts dimensionality reduction, reducing the number of parameters in the input. Similar to the convolutional layer, the pooling operation sweeps a filter across the entire input, but the difference is that this filter does not have any weights. Instead, the kernel applies an aggregation function to the values within the receptive field, populating the output array. There are two main types of pooling:

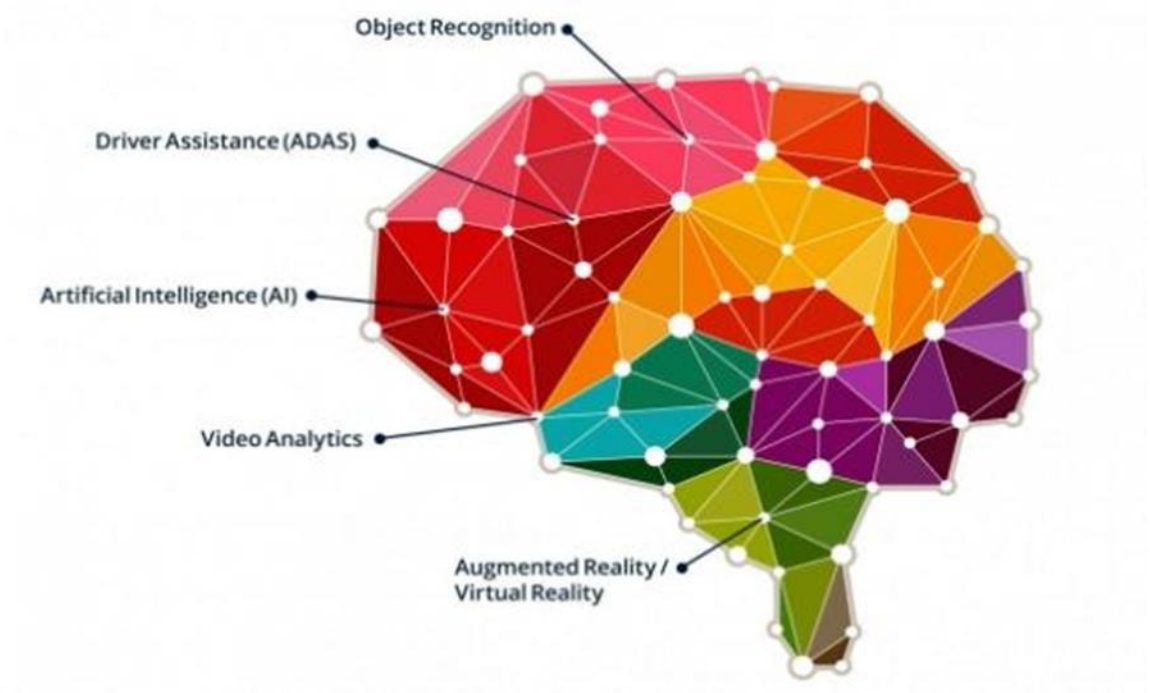
- **Max pooling:** As the filter moves across the input, it selects the pixel with the maximum value to send to the output array. As an aside, this approach tends to be used more often compared to average pooling.
- **Average pooling:** As the filter moves across the input, it calculates the average value within the receptive field to send to the output array.

Fully-Connected Layer:

- The name of the full-connected layer aptly describes itself. As mentioned earlier, the pixel values of the input image are not directly connected to the output layer in partially connected layers. However, in the fully-connected layer, each node in the output layer connects directly to a node in the previous layer.

Convolutional Neural Network Design :

- The construction of a convolutional neural network is a multi-layered feed-forward neural network, made by assembling many unseen layers on top of each other in a particular order.
- It is the sequential design that give permission to CNN to learn hierarchical attributes.
- In CNN, some of them followed by grouping layers and hidden layers are typically convolutional layers followed by activation layers.



5.FLASK:

Overview

- What is a Framework
- FrontEnd vs BackEnd
- What is Flask Framework
- Installation of Flask
- Creating our first Flask app
- Routing
- Static Route vs Dynamic Route
- HTML Injections
- HTML Escaping
- Hypertext Transfer Protocol
- GET and POST Methods

What is a Framework:

- The framework is the basis upon which software programs are built. It serves as a foundation for software developers, allowing them to create a variety of applications for certain platforms. It is a set of functions and predefined classes used to connect with the system software and handle inputs and outputs.
- It simplifies the life of a developer while giving them the ability to use certain extensions and makes the online applications scalable and maintainable.

Frontend Development vs Backend Development:

1. The front end of a website is the area with which the user immediately interacts. It contains everything that users see and interact with: text colours and styles, images and videos, graphs and tables, the navigation menu, buttons, and colours. HTML, CSS, and JavaScript are used in developing the front end. Some of the frameworks for frontend are:
    - AngularJS
    - Sencha Ext JS
    - React
2. The backend of a website refers to the server-side of the website. It saves and organizes data and ensures that everything on the client-side of the website functions properly. It is the section of the website that you are unable to interact with or access. It is the part of the software that has no direct communication with the users. Back End development is done using Java, Python, C#, and JavaScript. Some of the framework for the backend are :

    - Flask
    - Tornado
    - PyramidArmin Ronacher
    - Django

Flask Framework:

- Flask is used for developing web applications using python, implemented on Werkzeug and Jinja2. Advantages of using Flask framework are:

    - There is a built-in development server and a fast debugger provided.
    - Lightweight

- Secure cookies are supported.
- Templating using Jinja2.
- Request dispatching using REST.
- Support for unit testing is built-in.

Installation of Flask:

Python Version

- Install the latest version of Python or at least use a version >= Python 3.7

Creating Virtual Environment

- Virtual environments are separate collections of Python libraries, one for each project. Installed packages for one project do not affect other projects or the operating system's packages. Python has the venv package, which allows you to build virtual environments.

For Windows

```
> mkdir myproject
> cd myproject
> py -3 -m venv venv
```

For Mac

```
$ mkdir myproject
$ cd myproject
$ python3 -m venv venv
```

Make the Environment Active

Before you begin working on your project, turn on the environment:

For Windows

```
> venvScriptsactivate
```

For Mac

$. venv/bin/activate

The name of the current active environment will be shown in your shell prompt.

Install Flask

Run the following command in the active environment to install Flask:

$ pip install Flask

Creating our first Flask app:

Let's make our first flask app now. Flask application may look like this:

Python Code:

If everything works fine this will start a built-in server, which is used for testing but probably not suitable for production usage, your console will show the following output:

```
* Serving Flask app 'main' (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: on
* Restarting with stat
* Debugger is active!
* Debugger PIN: 145-731-869
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

This means that our app is running on http://127.0.0.1:5000/. Here 127.0.0.1 is the default IP address.

Now let's understand the code:

1. We started by importing the Flask class.
2. We then make an instance of this class. The '__name__' argument is passed which is the name of the application's module or package. Flask needs this to know where to look for resources like templates and static files.

3. The route() decorator is then used to inform Flask which URL should activate our method.
4. This method returns the message that should be shown in the user's browser.

Debug mode:

❖ The flask run command is capable of doing more than simply starting the development server. When we enable debug mode, the server will immediately reload if the code changes, and an interactive debugger will appear in the browser if an error occurs during a request.

Routing in Flask:

❖ Creating fixed URLs is so damn simple. Users are more likely to enjoy and return to a website if it has a meaningful URL that they can remember and use to get directly to the page. To assist users, modern online apps employ meaningful URLs. To tie a function to a URL, use the route() decorator.

```
@app.route('/')
def index():
   return 'This is Home Page'


@app.route('/hello')
def hello():
   return 'This is Hello, World Page'
```

Now let's understand the code:

1. app.route('/') will route to the home page URL, trailing slash '/'is generally used as a convention for the home page.
2. app.route('/hello') will route to the hello page URL.

In this way, we can render as many distinct web page URLs we want.

Static Routes vs Dynamic Routes in Flask:

Static Route:

- As we can understand from the name that static routes are fixed routes i.e. for each route functionalities we have to explicitly define the same function for each URL route.

```
@app.route('/Elon')
def greet():
    return 'Welcome Elon. What would you like to order.'


@app.route('/John')
def hello():
    return 'Welcome John. What would you like to order.'
```

This code will greet different users with the same message.

Although implementing the same function for multiple routes isn't a big problem until you get to a certain number of routes. It quickly becomes a frantic process when we are dealing with hundreds of routes.

Dynamic Route:

- Instead of defining the same function for each route, we can add variable sections to a URL by annotating sections with . The variable name is then sent to the defined function as the parameter.

```
@app.route('/')
def greet(name):
    return f'welcome {name}. What would you like to order.'
```

in the route captures a value from the URL and passes it to the function. So even if million of users route this web page we can handle all the routes by defining merely one function.

We can also use a converter to define the type of the parameter, such as.

```
@app.route('/post/')
def show_post(post_id):
    return f'Post ID is: {post_id}'


@app.route('/path/')
def show_subpath(subpath):
    return f'The given subpath is: {subpath}'
```

Defining datatype along with variable name enforces the user to follow the convention while passing the route name to the server.

Converter types:

int	only accepts positive integers
float	only accepts positive floating-point values
string	accepts text (by default) without a slash
path	like string but also accepts slashes
uuid	accepts UUID strings

HTML Injections:

- HTML injection is a way of modifying a web page that an application presents to its users by using non-validated input. If a program does not validate the user data, an attacker can send a malicious text in HTML format which can modify the content of the site, seen by other users. A well-designed query can result in the insertion of attacker-controlled HTML elements into a web page, changing the way application content is displayed on the web.

HTML Escaping:

To guard against injection attacks, all user-provided values rendered in the output must be enclosed when returning the HTML (the default response type in Flask).

The escape() function can be used manually. Most examples remove it for brevity, but you should always be mindful of how you're accessing untrusted data.

```
from flask import escape


@app.route("/")
def hello(name):
    return f"Hello, {escape(name)}!"
```

Hypertext Transfer Protocol:

- ❖ When we are searching for any web pages, to access their server and client employs several HTTP methods to communicate. As you work with Flask, you should become familiar with the HTTP methods.
- ❖ The Hypertext Transfer Protocol (HTTP) is a protocol that allows clients and servers to share information. HTTP is a request-response protocol used to communicate between a client and a server. A client (browser) sends an HTTP request to the server, and the server answers.

HTTP methods

- GET
- POST
- PUT
- HEAD
- DELETE
- PATCH
- OPTIONS

Currently, we will look at GET and POST, these are the most commonly used methods.

GET Method:

GET is used to fetch information from a certain website.

POST Method:

- POST is used to send data to a server to update or create a resource.

```
from flask import request
@app.route('/login', methods=['GET', 'POST'])
def login():
    if request.method == 'POST':
        return do_the_login()
    else:
        return show_the_login_form()
```

- ❖ A route, by default, only responds to GET queries. The route() decorator's methods parameter may be used to handle discussed HTTP methods.
- ❖ After reading this article, I recommend that you should practice running each code in your system so that you can get used to it.