Python Machine Learning Tutorial:

Machine learning ?

Ans. It is a technique which learn from the data. Imagine you have a friend who’s trying to learn how to identify different types of fruit. At first, you might show them a bunch of apples and say. These are apples. Then you show them some orange and say.These are oranges. Your frnd starts to notice patterns. Like apples are usually round and red in colour, while oranges are round and orange in color.

So Machine learning is similar, instead of a frnd, you have a computer program, and instead of fruit, you have data like pictures of apples and oranges. The program learns from this data, finding patterns and connections that help it make to predict or take decisions. Based on features it will recognised and will give apples or oranges.

Applications:

1. Self-driving Cars
2. Robotics
3. Language Processing
4. Vision Processing
5. Stock Markets Trends

Steps:

1. Import the data
2. Clean the data-> remove dupllicate data
3. Split the data into training : 80% / test sets : 20%
4. Create a model-> Allgorithm which give accuracy
5. Train the model-> make train to allgorithm
6. Make Predictions
7. Evaluate and Improve

Libraries and Tools:

1. Numpy : It is used to provide multiple dimension array and also used for numerical computing in Python. It provides support the large, multi-dimensional arrays and matrices.
2. Pandas : It is used to load clean explore and analyze data easily from spreadsheet or excel sheet.
3. MatPlotlib : It is 2 dimension graph which represent graphical representation with plotting.
4. Scikit-Learn : It is used to provides no of algorithm like decision tree and neural network.
5. Tensorflow : TensorFlow is an open-source machine learning framework developed by Google. It provides comprehensive support for building and training deep learning models, including neural networks, convolutional neural networks (CNNs), recurrent neural networks (RNNs), and more. TensorFlow offers flexibility and scalability and is commonly used in research and production environments.

Supervised Learning : it consist of training data, both I/p and o/p and classification with Naive Bayes algorithm or different type of algorithm.

I/p o/p

I/p o/p

I/p o/p ----------->Learning Algorithm ----------> New I/p

. . . Model

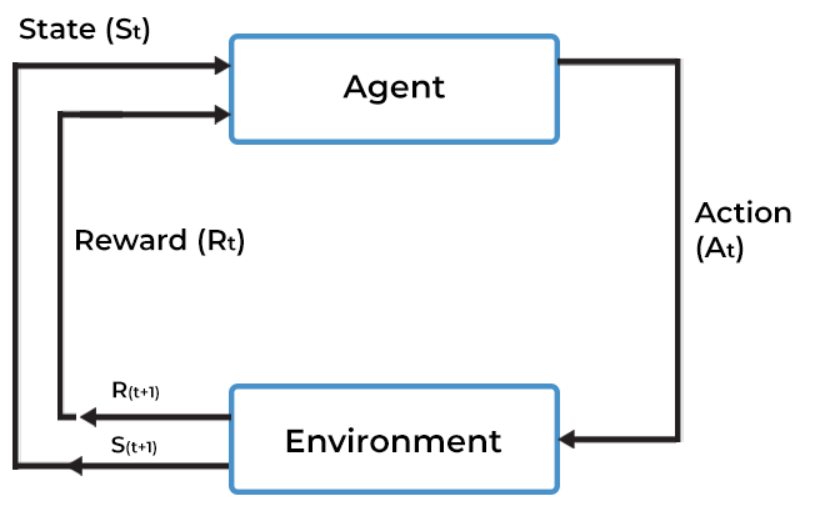
. . .

I/p o/p o/p

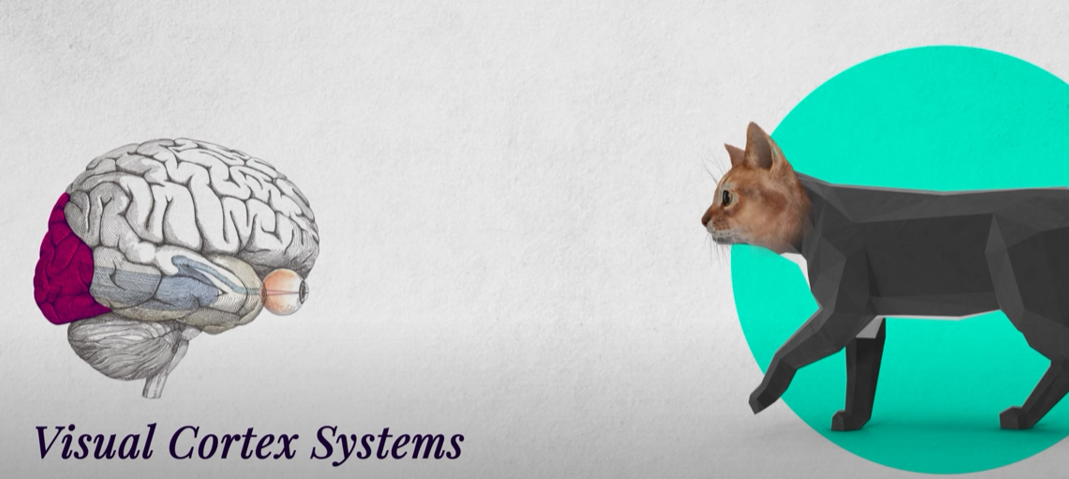
Training Data

Unsupervised Learning : Only I/p, clustering and k-mean.

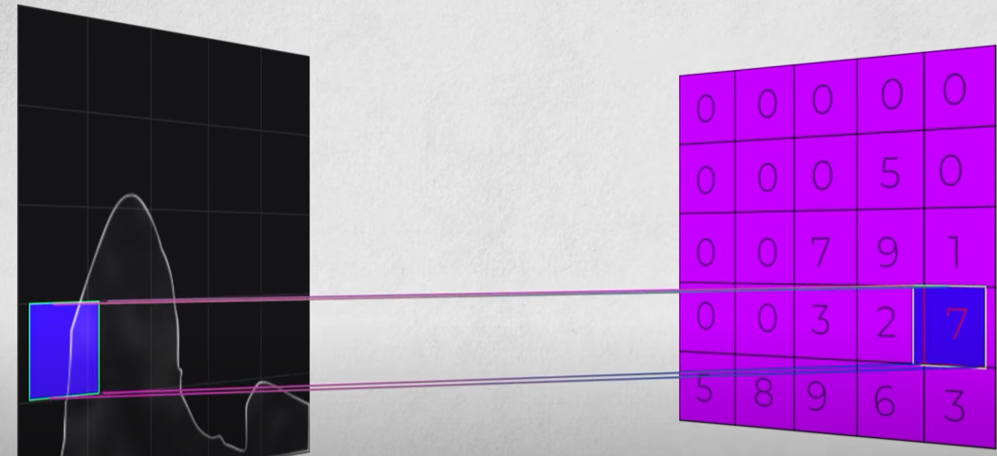
Reinforcement Learning : It depends on rewards and penalty .



CNN Algorithm : human being can easily recognised what actually a image is. Machine can’t

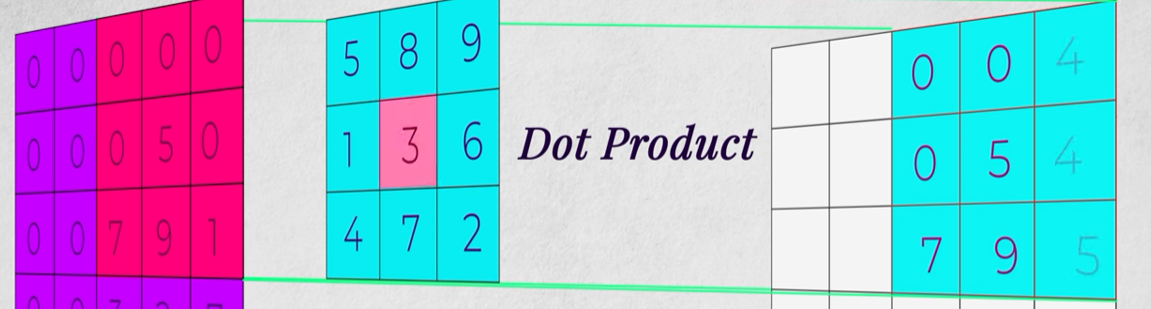


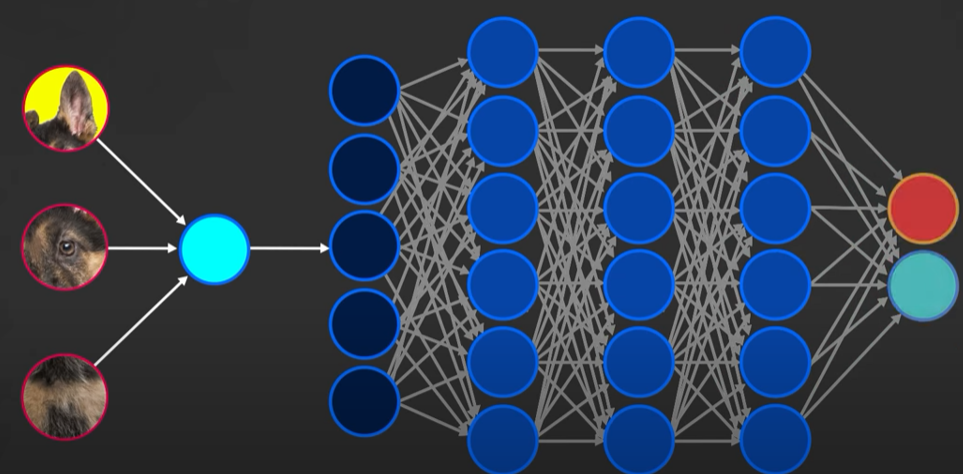
This is consist of convolutional and pooling, where convolutional means to scanning of an image in a different features like curve, color and blur and edge.

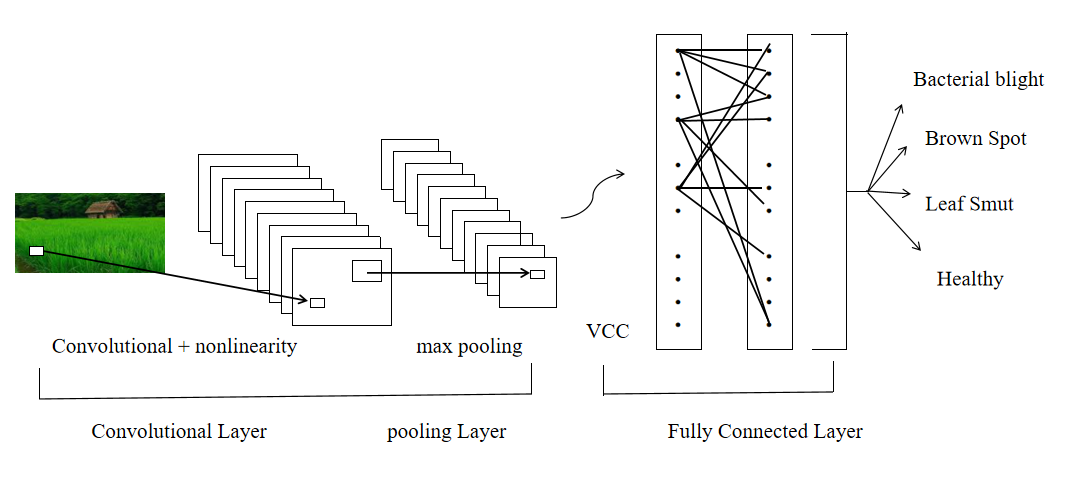


Random picking index value.

It will make coordinate with pixel value of image with random value.







Django :

Certainly! Django is a powerful web framework written in Python that simplifies the process of building web applications. Here's a breakdown in easy words:

Framework : Imagine a framework as a toolbox with pre-built tools that help you build something. Django is like a toolbox for building websites.

Python : It's a programming language, and Django is written in Python. So, to use Django, you write code in Python.

Web Applications : These are like websites but with more functionality. For example, a simple website might just display information, but a web application can allow users to interact, like signing up, logging in, and posting comments.

Simplifies Web Development: Django provides a set of ready-to-use components and tools for common web development tasks, like handling user authentication (sign up, log in), managing databases, and generating HTML.

pip install django

pip install django --upgrade

py -m pip install Django

django-admin startproject djangoapp

cd djangoapp

python manage.py runserver

python manage.py startapp home

python manage.py

MVT : Model View Templates

Model is data

View is physical view

Templates which contain dtl and html and css



It is contain of model and data and template to form a django

MVC :

Model--> Template

Template-----> View

View----> Controller

Node.js

Scenario: Simple Chat Application

Imagine a basic chat app where users can send messages to each other in real-time, and those messages instantly appear on everyone's screens. This is a common use case for Node.js due to its event-driven nature and ability to handle multiple connections efficiently.

Key Components:

Server-Side Node.js Application:

Written in JavaScript using Node.js.

Listens for incoming connections from web browsers (clients).

Uses a library like Socket.IO to establish real-time communication channels (sockets) with clients.

Broadcasts messages received from one user to all connected users in real-time.

Client-Side Web Page (HTML, CSS, JavaScript):

Connects to the Node.js server using Socket.IO.

Provides an interface for users to type and submit messages.

Listens for incoming messages (broadcasts) from the server and displays them on the screen.

npm init : we used to import package

Package name : Prateek

Version: enter

Description: Its an amazing package

Entry point: (index.js)

Test command:

Git repository:

Keywords:

Author: Prateek

License: (ISC)

Is this ok? (yes) yes

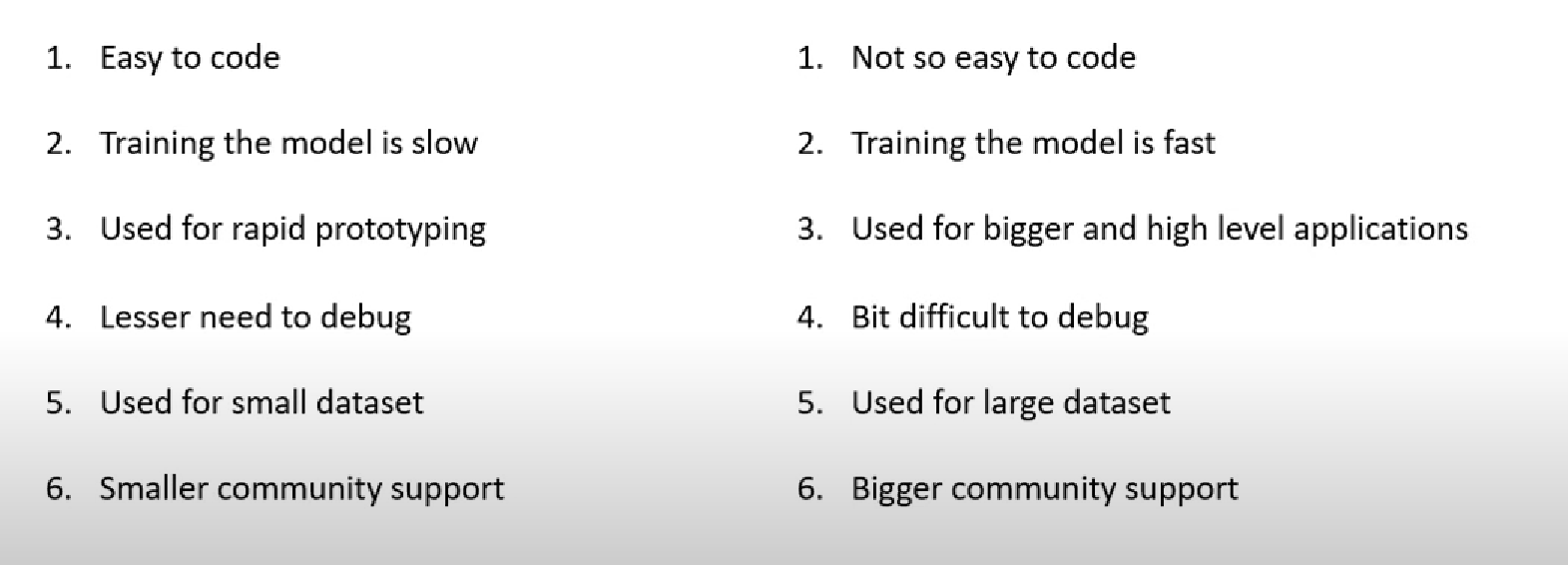
Note: here note modules is consist of all dependencies from internet.

If I want to install any package as global then we have to use command this : npm i -q nodemon

We use npm i for getting express module

^

Keras and Tensorflow :



Back propagation :

Imagine you're trying to teach a robot to catch a ball. At first, it's terrible at it. But every time it tries, you tell it how far off it was from catching the ball. This is like giving it feedback on its performance.

Now, the robot wants to get better at catching the ball. It adjusts its movements based on the feedback you gave it. If it moved too slow, it speeds up next time. If it moved too fast, it slows down. This adjusting process is like updating its strategy based on the feedback.

Back propagation works similarly in neural networks. It's the process of adjusting the weights of connections between neurons based on the difference between the predicted output and the actual output. This adjustment helps the network get better at making predictions over time, just like our robot gets better at catching the ball with practice and feedback.