ARTIFICIAL INTELLIGENCE

Summer Internship Report Submitted in partial fulfillment of the requirement for under graduate degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE ENGINEERING

By

BODDAPATI VENKATA NAGA AISHWARYA

221810309012

Under the Guidance of

Dr.Arshad Ahmad Khan Mohammad

Assistant Professor

Department of Computer Science and Engineering

GITAM School of Technology

GITAM Deemed to be University

Hyderabad Campus – 502329

DECLARATION

I submit this Summer Internship entitled "**ARTIFICIAL INTELLIGENCE**" to GITAM School of Technology, GITAM Deemed to be University ,Hyderabad campus in partial fulfillment of the requirements for the award of the degree of "**Bachelor of Technology**" in "**Computer Science Engineering".**

I declare that it was carried out independently by me under the guidance of Dr. Arshad Ahmad Khan Mohammad , Assistant Professor-GITAM School of Technology , GITAM Deemed to be University, Hyderabad, India.

The results embodied in this report have not been submitted to any other University or institute for the award of any degree or diploma.

PLACE:-HYDERABAD BVN AISHWARYA

DATE:- 15-10-2021 221810309012

Department of Computer Science and Engineering
GITAM(Deemed to be University)
Rudraram Mandal,Sangareddy district,Patancheru, Hyderabad,Telangana 502329

CERTIFICATE

This is to certify that the Summer Internship Report entitled "**ARTIFICIAL INTELLIGENCE**" is being submitted by **BVN Aishwarya** (221810309012) in partial fulfillment of the requirement for the award of Bachelor of Technology in "**COMPUTER SCIENCE AND ENGINEERING**" at GITAM University, Hyderabad.

It is faithful record work carried out by her at the **Computer Science and Engineering Department**, GITAM School of Technology, GITAM Deemed to be University, Hyderabad Campus under my guidance and supervision.

Dr.Arshad Ahmad khan Mohammad	Prof.S. Phani kumar
Assistant Professor	Professor and HOD
Department of CSE	Department of CSE

MINI-PROJECT CERTIFICATE

VERZEO

Certificate Of Course Completion

Bvn Aishwarya

has successfully completed Artificial Intelligence course

from 01-07-2021 to 31-08-2021.

During this course, we found the student to be a keen and enthusiastic candidate.

Ajay Billav
Course Co-ordinator

T. Nikhil
Academic Head

Verified Certificate

Certificate ID: 3823289419

Date: 27-09-2021

Our Course Completion certificates are verified and are recognized by eminent industries and universities all over the world.

INTERNSHIP CERTIFICATE

VERZEO

Certificate of Internship

Bvn Aishwarya

has successfully completed an Internship Program Artificial Intelligence

from 01-07-2021 to 31-08-2021.

During the internship, the student was found to be dedicated, hardworking and intelligent.

T.Nikhil
Academic head

Verified Certificate

Certificate ID: 2919571096
Date: 13-10-2021

Our internship completion certificates are verified and are recognized by eminent industries and universities all over the world.

ACKNOWLEDGEMENT

This internship would not have been successful without the help of several people. I would like to thank the personalities who were part of project in numerous ways,those who gave us outstanding support from the birth of the seminar.

I am extremely thankful to our honorable Pro-Vice Chancellor ,Prof.Siva Prasad for providing necessary infrastructure and resources for the accomplishment of this seminar.

I am very much obliged to our beloved Prof.S.Phani Kumar, Head of the Department of Computer Science and Engineering for providing the opportunity to undertake this project and encouragement in completion of this seminar.

I hereby wish to express a deep sense of gratitude to Dr.Arshad Ahmad khan Mohammad, Assistant Professor -Department of Computer Science and Engineering for their esteemed guidance ,moral support and invaluable advice provided by them for the success of this internship.

I am also thankful to all the staff members of Computer Science and Engineering department who have cooperated in making this project a success. I would like to thank my parents and friends who extended their help, encouragement and moral support either or indirectly in this seminar.

BVN AISHWARYA

221810309012

ABSTRACT

Artificial Intelligence is a branch of computer science and a multidisciplinary field that deals with the simulation of intelligent behavior in computers. Two interesting and innovative fields Machine learning and Deep Learning are widely used to work with the real data.

In the recent years, deep learning has achieved great success in many fields, such as computer vision and natural language processing. Compared to traditional machine learning methods, deep learning has a strong learning ability and can make better use of datasets for feature extraction.

In this internship, we mainly introduce some advanced neural networks of deep learning and their applications. We build a simple neural network using classification model(Supervised Learning Technique) to make a heart disease predictive model to predict the “target” column of dataset.We used keras algorithm for developing and evaluating deep learning models.

In addition to above project, we also build a Convolution Neural Network which is used for image classification.Using CNN , we classified and predicted masked and non-masked images.

BVN AISHWARYA

221810309012

INDEX TABLE

1. PYTHON

1.1 Introduction to python
1.2 Features of python
1.3 Installation
1.4 Python Datatypes
1.5 Variables
1.6 Assigning values to variables
1.7 Data Structures
1.8 Arrays
1.9 Loops
1.10 Functions

2.Pandas and Visualization

2.1 Central Limit theorem
2.2 Exploratory Data Analysis

3.INTRODUCTION TO ARTIFICIAL INTELLIGENCE

3.1 What is Artificial Intelligence?
3.2 How Machine Learning and Deep Learning to related?
3.3 Machine Learning Techniques
3.4 Deep Learning is the next generation of Machine Learning

4. INTRODUCTION TO NEURAL NETWORK
4.1 Questions to be raised while learning neural network 4.2 Activation Function 4.3 Types of Activation Functions 4.4 Loss Function
5. CONVOLUTION NEURAL NETWORK
5.1 Introduction to CNN 5.2 Architecture of CNN 5.3 Activation Function used in CNN
6.ALGORITHMS AND LIBRARY
6.1 Role of keras 6.2 Role of opencv 6.3 Role of OS Module
7.MINI-PROJECT CODE
8.MAJOR-PROJECT CODE
9.CONCLUSION
10.REFERENCES

PYTHON

1.1 INTRODUCTION TO PYTHON:

Python is a high-level object-oriented programming language that was created by Guido van Rossum. It is also called general-purpose programming language as it is used in almost every domain we can think of as mentioned below:

- Web Development
- Software Development
- Game Development
- AI & ML
- Data Analytics

1.2 FEATURES OF PYTHON:

- Python is object oriented
- Its free(open source)
- Its powerful
- Its portable
- Its easy to learn and use
- Interpreted Language
- Interactive programming language
- Straight forward syntax

1.3 INSTALLATION:

There are many interpreters available freely to run Python scripts like IDLE (Integrated Development Environment) .

Steps to be followed and remembered:

Step 1: Select Version of Python to Install.

Step 2: Download Python Executable Installer.

Step 3: Run Executable Installer.

Step 4: Verify Python Was Installed On Windows.

Step 5: Verify Pip Was Installed.

Step 6: Add Python Path to Environment Variables

- We can download anaconda navigator and use python in jupyter notebook.

1.4 PYTHON DATATYPES

The data stored in memory can be of many types. For example, a student roll number is stored as a numeric value and his or her address is stored as alphanumeric characters. Python has various standard data types that are used to define the operations possible on them and the storage method for each of them.

- Int: Int, or integer, is a whole number, positive or negative, without decimals, of unlimited length.
  Eg: a=10

- Float : Float, or "floating point number" is a number, positive or negative, containing one or more decimals. Float can also be scientific numbers with an "e" to indicate the power of 10.
  Eg: x=35e3

- Boolean : Objects of Boolean type may have one of two values, True or False

- String : Strings in Python are identified as a contiguous set of characters represented in the quotation marks. Python allows for either pairs of single or double quotes.

  Eg: print(“Hello”)

1.5 VARIABLES

Variables are nothing but reserved memory locations to store values. This means that when you create a variable you reserve some space in memory. Based on the data type of a variable, the interpreter allocates memory and decides what can be stored in the reserved memory. Therefore, by assigning different data types to variables, you can store integers, decimals or characters in these variables.

Rules for Python variables:
• A variable name must start with a letter or the underscore character
• A variable name cannot start with a number
• A variable name can only contain alpha-numeric characters and underscores (A-z, 0-9, and _)
• Variable names are case-sensitive (age, Age and AGE are three different variables)

1.6 ASSIGNING VALUES TO VARIABLES:

Python variables do not need explicit declaration to reserve memory space.

- The declaration happens automatically when you assign a value to a variable.
- The equal sign (=) is used to assign values to variables.
- The operand to the left of the = operator is the name of the variable and the operand to the right of the = operator is the value stored in the variable.

1.7 DATA STRUCTURES:

Data structures are a way of organizing and storing data so that they can be accessed and worked with efficiently.

- Lists : A data structure that stores an ordered collection of items in Python is called a list. In other words, a list holds a sequence of items. You need to put all the items, separated by commas, in square brackets to let Python know that a list has been specified.

  Syntax of a list :

  X= [item1,item2,……itemn]

- Tuple : Similar to a list, the tuple is a built-in data structure in Python. However, it doesn't support the same level of extensive functionality. The most important difference between a list and a tuple is mutability. Unlike lists, tuples are immutable i.e. they can't be modified.

  Syntax of tuple :

  X = (item1,item2,….item n)

- Dictionary : Another type of built-in data structure in Python is the dictionary. It stores data in the form of key-value pairs. The keys defined for a dictionary need to be unique. Though values in a dictionary can be mutable or immutable objects, only immutable objects are allowed for keys.

  Syntax of Dictionary:

  a={key1:"value1", key2:"value2",…..keyn:"value n"}

1.8 ARRAYS:

An array is a special variable, which can hold more than one value at a time.

To work with arrays in python , we use NumPy library.

Accessing array elements:

Eg: cars=[“Alto”,”BMW”,”Swift”]

cars[0]=”Alto”

cars[1]=”BMW”

cars[2]=”Swift”

The Length of an Array:

Use the len() method to return the length of an array.

Eg: x= len(cars)

X=3

1.9 LOOPS:

While Loop:

A while loop in python iterates till its condition becomes False. In other words, it executes the statements under itself while the condition it takes is True.

for loop:

It is used for iterating over a sequence (that is either a list, a tuple, a dictionary, a set, or a string).

This is less like the for keyword in other programming languages, and works more like an iterator method as found in other object-orientated programming languages. With the for loop we can execute a set of statements, once for each item in a list, tuple, set etc.

1.10 FUNCTIONS:

A function is a block of code which only runs when it is called.

Creating the function:

In Python a function is defined using the def keyword:

Inbuilt functions:

Functions that are built into Python are called inbuilt functions.

Ex: abs(),all().ascii(),bool() etc.

Userdefined functions:

Functions defined by the users themselves.

2.PANDAS AND VISUALIZATION:

Pandas library in python is mainly used for data analysis. It is not a data visualization library but, we can create basic plots using Pandas. Pandas is highly useful and practical if we want to create exploratory data analysis plots. We do not need to import other data visualization libraries in addition to Pandas for such tasks.

As Pandas is Python's popular data analysis library, it provides several different functions to visualizing our data with the help of the .plot() function.

There is one more advantage of using Pandas for visualization is we can serialize or create a pipeline of data analysis functions and plotting functions. It simplifies the task.

matplotlib.pyplot is a collection of functions that make matplotlib work like MATLAB.

2.1 CENTRAL LIMIT THEOREM:

The Central Limit Theorem states that the sampling distribution of the sample means approaches a normal distribution as the sample size gets larger.

The sample means will converge to a normal distribution regardless of the shape of the population. That is, the population can be positively or negatively skewed, normal or non-normal.

The Central Limit theorem is closely related to the Law of Large Numbers, which states that:

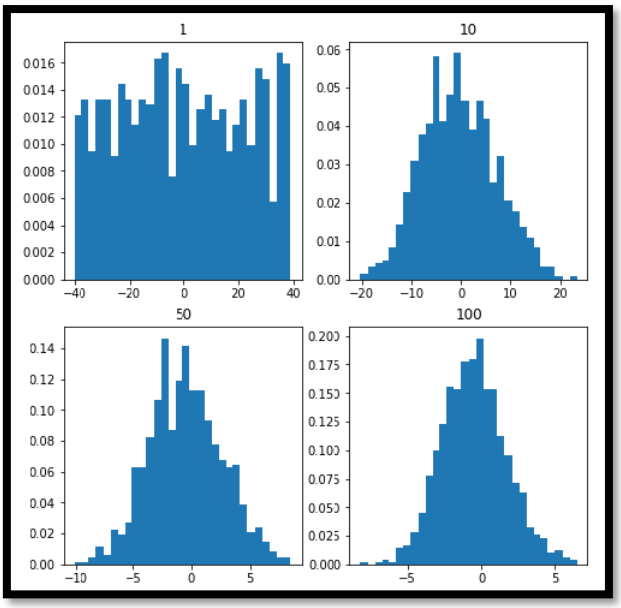
As a sample size grows, the sample mean gets closer to the population mean.

It turns out that the finding is critically important for making inferences in applied machine learning.

So, how are these two related?

CLT states that — as the sample size tends to infinity, the shape of the distribution resembles a bell shape (normal distribution).

The center of this distribution of the sample means becomes very close to the population mean — which is essentially the law of large numbers.



2.2EXPLORATARY DATA ANALYSIS:

Exploratory Data Analysis, or EDA, is an important step in any Data Analysis or Data Science project. EDA is the process of investigating the dataset to discover patterns, and anomalies (outliers), and form hypotheses based on our understanding of the dataset.

EDA involves generating summary statistics for numerical data in the dataset and creating various graphical representations to understand the data better.

To understand the steps involved in EDA, we will use Python as the programming language and Jupyter Notebooks because it's open-source, and not only it's an excellent IDE but also very good for visualization and presentation.

The main aim of EDA:

- Examine the data distribution
- Handling missing values of the dataset(a most common issue with every dataset)
- Handling the outliers
- Removing duplicate data
- Encoding the categorical variables
- Normalizing and Scaling

3. INTRODUCTION TO ARTIFICIAL INTELLIGENCE

3.1 WHAT IS ARTIFICIAL INTELLIGENCE?

Artificial Intelligence is an approach to make a computer, a robot, or a product to think how smart human think. AI is a study of how human brain think, learn, decide and work, when it tries to solve problems. And finally this study outputs intelligent software systems.

The aim of AI is to improve computer functions which are related to human knowledge, for example, reasoning, learning, and problem-solving.

The intelligence is intangible. It is composed of

- Reasoning
- Learning
- Problem Solving
- Perception

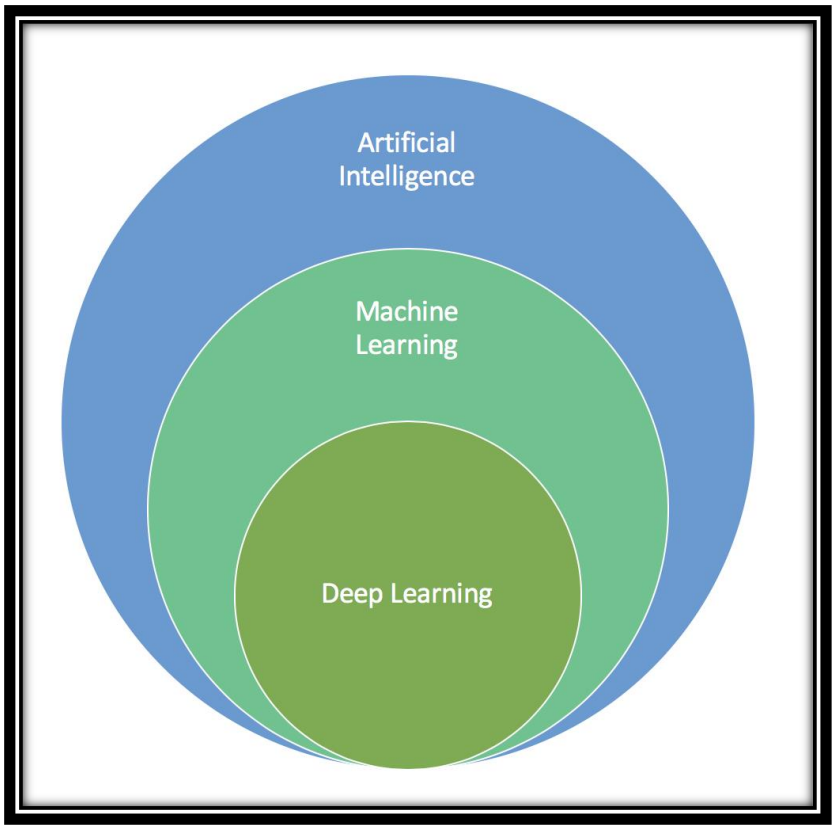
The objectives of AI research are reasoning, knowledge representation, planning, learning, natural language processing, realization, and ability to move and manipulate objects. There are long-term goals in the general intelligence sector.

Approaches include statistical methods, computational intelligence, and traditional coding AI.

During the AI research related to search and mathematical optimization, artificial neural networks and methods based on statistics, probability, and economics, we use many tools.

Computer science attracts AI in the field of science, mathematics, psychology, linguistics, philosophy and so on.

3.2 HOW MACHINE LEARNING AND DEEP LEARNING ARE REALTED TO ARTIFICIAL INTELLIGENCE?



AI Systems often incorporate artificial intelligence, machine learning, and deep learning to create a sophisticated intelligence machine that will perform given human functions well. Increasingly, all three units are individual pieces of the entire AI System's intelligence puzzle.

Machine Learning :-

It is an application of artificial intelligence that provides the AI System with the ability to automatically learn from the environment and applies that learning to make better decisions. There are a variety of algorithms that Machine Learning uses to iteratively learn, describe and improve data in order to predict better outcomes. These algorithms use statistical techniques to spot patterns and then perform actions on these patterns.

Deep Learning :-

It is the next generation of Machine Learning. It's a subset of Machine Learning. Deep Learning models can make their own predictions entirely independent of humans. Machine Learning models of the past still need human intervention in many cases to arrive at the optimal outcome. Deep Learning models use artificial neural networks. The design of this network is inspired by the biological neural network of the human brain.

It analyzes data with a logical structure similar to how a human would draw conclusions. It helps to solve complex problems like computer vision.

3.3 MACHINE LEARNING TECHNIQUES:

The basics of machine learning comprise of learning from the environment, then applying that learning to make decisions. In order to do this effectively, there are categories of machine learning algorithms that make this possible.

Supervised Machine Learning :-

In supervised learning, the objective is to come up with a mapping function (f) that will best describe the input data (x) to conclude the output data (Y). We know x and we know Y. But, we have to find the mapping function (f) that will achieve a certain level of performance. Then, we can apply the mapping function (f) to new data to gain similar results. Training data is used to find the function f.

$$Y = f(X)$$

There are two types of Supervised Machine Learning problems: Classification and Regression depending on the type of output variable. If the output variable is categorical, then it is a classification problem. (Example: Color can be red, blue, purple, etc…)

If the output variable is a real value, then it is a regression problem. (Example: Height can be on a scale of 0ft to 10ft)

A List of Supervised Machine Learning algorithms include:

1. Linear Regression
2. Logistic Regression
3. Naives Bayes
4. Decision Trees
5. K-nearest neighbour algorithm

Unsupervised Machine Learning : -

Unlike Supervised Machine Learning, unsupervised machine learning does not assume a correct set of output “Y”. There are no outputs.

The objective here is to present the most interesting structure that best describes the input data.

There are two types of Unsupervised Machine Learning problems: Clustering and Association. Clustering problems are when you discover groupings inside the input data.

(Example: grouping voting behaviors by gender) Association is when you discover rules inside the input data. (Example: female voters tend to vote for female candidates)

A List of Unsupervised Machine Learning algorithms include:

1. Hierarchical Clustering
2. K-means Clustering
3. Expectation-maximization algorithm
4. Non-negative matrix factorization.
5. Local Outlier Factor

Reinforcement Learning : -

Unlike supervised ML and unsupervised ML, reinforcement learning is focused on finding the best path to take in a situation to maximize reward in a situation. The decision is made sequentially. Along each step the algorithm takes on the path to total reward, it will either have a positive or a negative reward. The total reward is the sum of all positive and negative rewards along the path. The goal is to find the best path that maximizes the reward. (A good example of this is an AI-enabled stock trading system.)

1. Q-Learning
2. State-Action-Reward-State-Action(SARSA)
3. Deep Q Network
4. Deep Deterministic Policy Gradient

3.4 DEEP LEARNING IS THE NEXT GENERATION OF MACHINE LEARNING

Deep Learning is the next generation of machine learning algorithms that use multiple layers to progressively extract higher level features from raw input. For instance, in image recognition applications, instead of just recognizing matrix pixels, deep learning algorithms will recognize edges at a certain level, nose at another level, and face at yet another level.

With the ability to understand data from the lower level all the way up the chain, a deep learning algorithm can improve its performance over time and arrive at decisions at any given moment in time.

The power of deep learning algorithm lies in its ability to take on both supervised learning tasks as well as unsupervised learning tasks. It also approximates many brain development theories of the human brain.

Deep learning algorithms are now used by computer vision systems, speech recognition systems, natural language processing systems, audio recognition systems, bioinformatics systems and medical image analysis systems.

4. INTRODUCTION TO NEURAL NETWORK

An Artificial Neuron Network (ANN), popularly known as Neural Network is a computational model based on the structure and functions of biological neural networks.

It is like an artificial human nervous system for receiving, processing, and transmitting information in terms of Computer Science.

Basically, there are 3 different layers in a neural network :-

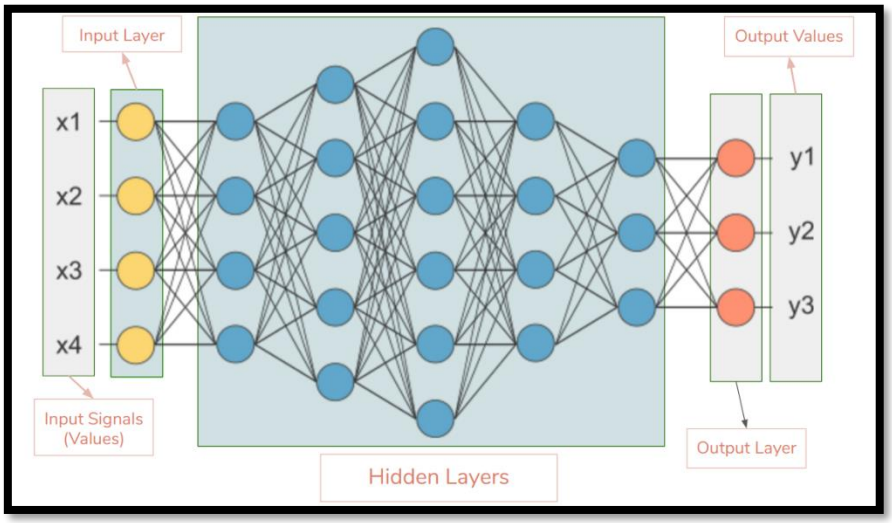
1. **Input Layer** :-All the inputs are fed in the model through this layer
2. **Hidden Layers** :- There can be more than one hidden layers which are used for processing the inputs received from the input layers.
3. **Output Layer** :- The data after processing is made available at the output layer.

   There are two algorithms in neural networks:

   - Forward Propagation Algorithm
   - Backward Propagation Algorithm

Forward Propagation : -

It is the way to move from the Input layer (left) to the Output layer (right) in the neural network.



Role of Input layer : -

- The Input layer communicates with the external environment that presents a pattern to the neural network.
- Its job is to deal with all the inputs only.
- This input gets transferred to the hidden layers which are explained below.
- The input layer should represent the condition for which we are training the neural network.
- Every input neuron should represent some independent variable that has an influence over the output of the neural network.

Role of Hidden Layers :-

The hidden layer is the collection of neurons which has activation function applied on it and it is an intermediate layer found between the input layer and the output layer. Its job is to process the inputs obtained by its previous layer. So it is the layer which is responsible extracting the required features from the input data. Many researches has been made in evaluating the number of neurons in the hidden layer but still none of them was successful in finding the accurate result. Also there can be multiple hidden layers in a Neural Network.

So you must be thinking that how many hidden layers have to be used for which kind of problem. Suppose that if we have a data which can be separated linearly, then there is no need to use hidden layer as the activation function can be implemented to input layer which can solve the problem.

But in case of problems which deals with complex decisions, we can use 3 to 5 hidden layers based on the degree of complexity of the problem or the degree of accuracy required. That certainly not means that if we keep on increasing the number of layers, the neural network will give high accuracy. A stage comes when the accuracy becomes constant or falls if we add an extra layer.

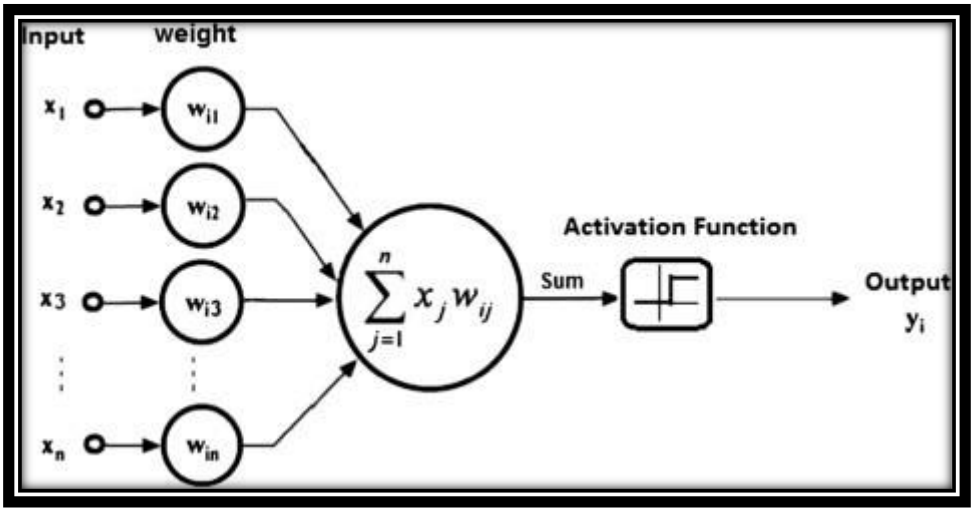
Also, we should also calculate the number of neurons in each network. If the number of neurons are less as compared to the complexity of the problem data then there will be very few neurons in the hidden layers to adequately detect the signals in a complicated data set.

If unnecessary more neurons are present in the network then Overfitting may occur. Several methods are used till now which do not provide the exact formula for calculating the number of hidden layer as well as number of neurons in each hidden layer.

Role of output layer :-

The output layer of the neural network collects and transmits the information accordingly in way it has been designed to give. The pattern presented by the output layer can be directly traced back to the input layer.

The number of neurons in output layer should be directly related to the type of work that the neural network was performing. To determine the number of neurons in the output layer, first consider the intended use of the neural network.



Summation function = X1Wi1+X2Wi2+…+XnWin

- Prediction of neural network is bad ,if the feature column shows a non-linear relationship with target column.

4.1 QUESTIONS TO BE RAISED WHILE LEARNING NEURAL NETWORK

QUESTIONS	ANSWERS
How many input layers are allowed?	1
How many hidden layers are allowed?	As many as
How many output layers are allowed?	1
How many neurons in input layer?	Depends on number of feature columns
How many neurons in hidden layer?	As many as
How many neurons in output layer?	Depends on number of target columns

4.2 ACTIVATION FUNCTION:

The purpose of an activation function is to add some kind of non-linear property to the function, which is a neural network. Without the activation functions, the neural network could perform only linear mappings from inputs ***x*** to the outputs ***y***.

Why is this so?

Without the activation functions, the only mathematical operation during the forward propagation would be dot-products between an input vector and a weight matrix.

In order to be able to compute really interesting stuff, neural networks must be able to approximate nonlinear relations from input features to output labels. Usually, the more complex the data is we are trying to learn something from, the more non-linear the mapping of features to the ground-truth-label is.

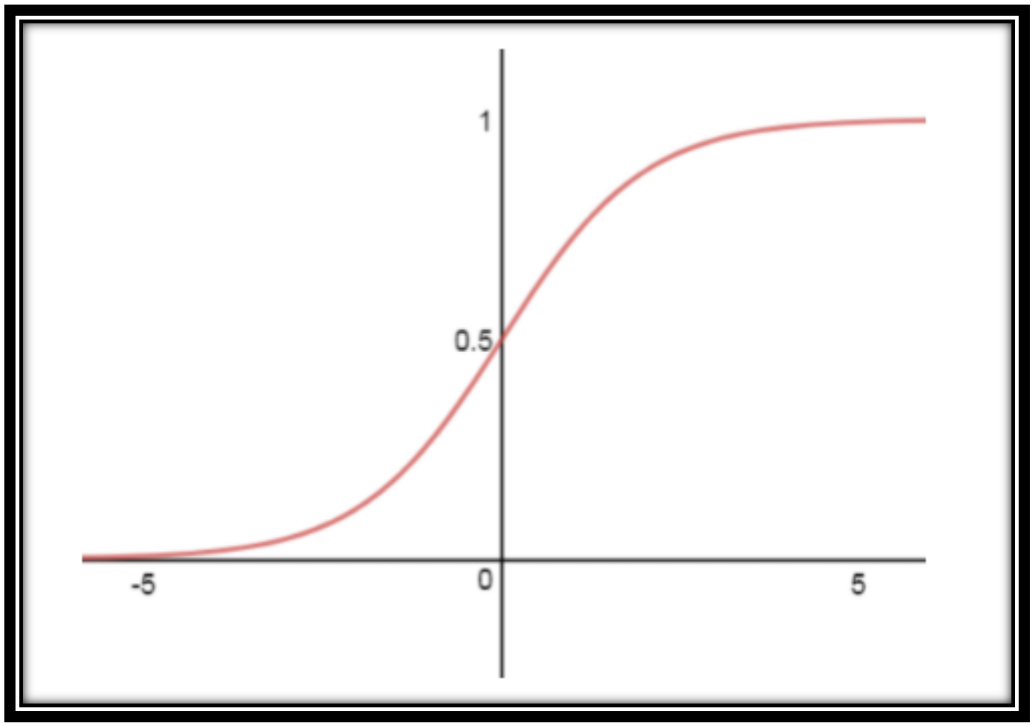
A neural network without any activation function would not be able to realize such complex mappings mathematically and would not be able to solve tasks we want the network to solve.

4.3 TYPES OF ACTIVATION FUNCTION :-

- Sigmoid Function
- ReLU Activation Function
- Softmax Function

➢ Sigmoid Function :

Some years ago the probably most common activation function you would have encountered is the sigmoid function. The sigmoid function maps the incoming inputs to a range between 0 and 1:



The sigmoid activation function is defined as follows:

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

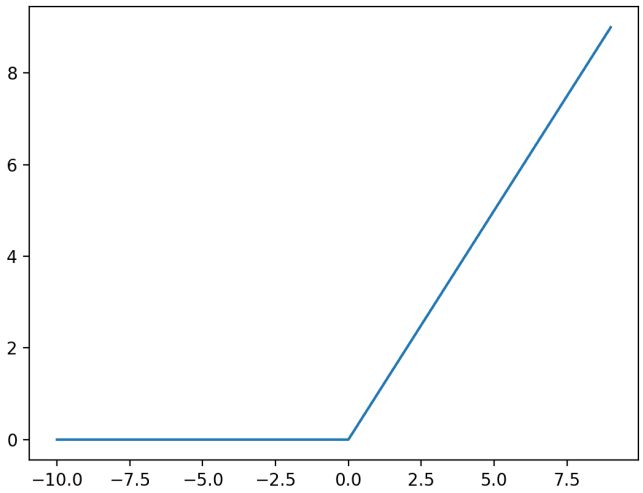
Drawback of sigmoid function :

- Sigmoids saturate and kill gradients. A very undesirable property of the sigmoid neuron is that when the neuron's activation saturates at either tail of 0 or 1, the gradient at these regions is almost zero.
- Sigmoid outputs are not zero-centered.

➢ Rectified linear Unit ->(ReLU)

The Rectified Linear Unit or just simply ReLUhas become very popular in the last few years. The activation is simply thresholded at zero: F(x)=max(0,x) or more precisely:

$$RELU(x) = \begin{cases} 0 \; if \; x < 0 \\ x \; if \; x >= 0 \end{cases}$$



➢ Softmax activation function:

Softmax is applied only in the last year and only when we want the neural network to predict probability scores during classification tasks.The softmax activation function forces the values of output neurons to take values between zero and one, so they can represent probability scores.

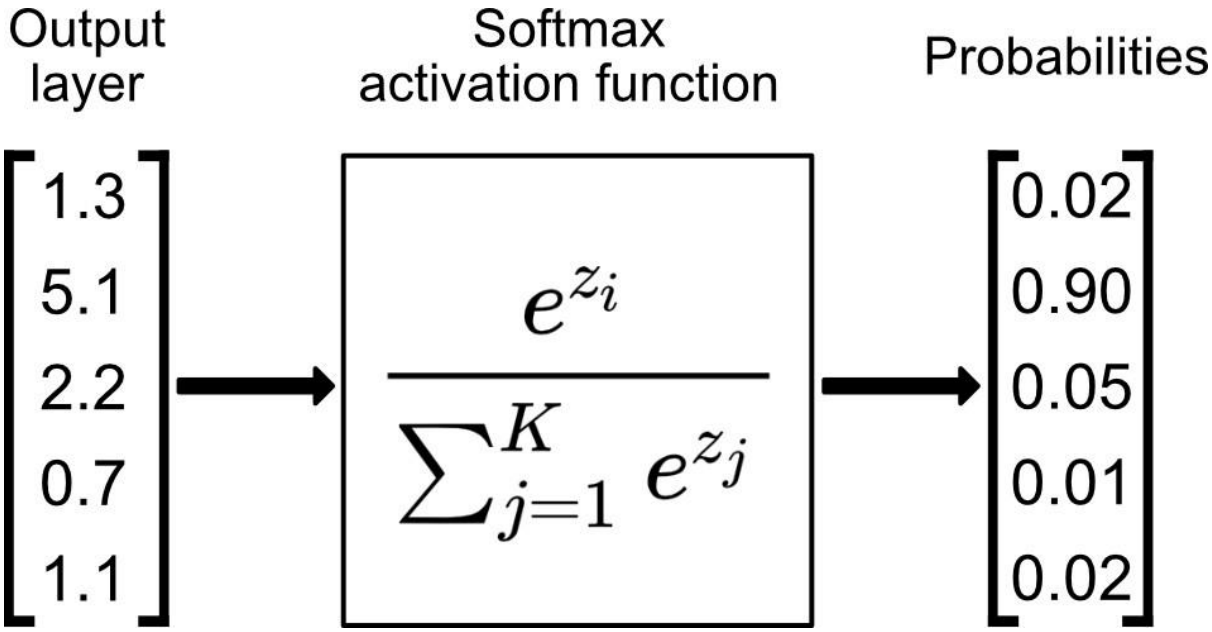
Another thing, that we must consider is, that when we perform the classification of input features into different classes, these classes are mutually exclusive.

This means that each feature vector ***x*** belongs to only one class. Meaning a feature vector that is an image of a dog can not represent a dog class with a probability of 50% and with a probability of 50% a cat class. This feature vector must represent the dog class with a probability of 100%

Besides, in the case of mutually exclusive classes,the probability scores across all output neurons must sum up to one**.** Only in this way the neural network represents a proper probability distribution.

A counterexample would be a neural network that classifies a dog's image into the class dog with a probability of 80% and with a probability of 60% into the class cat.

Fortunately, the softmax function does not only force the outputs into the range between zero and out, but the function also makes sure that the outputs across all possible classes sum up to one.



4.4 LOSS FUNCTION:

Loss function is one of the most important component of neural networks. It is a technique to combine multiple error values to reduce it to a single value.

TYPES OF LOSS FUNCTION:

- **Mean Squared Error**

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(\hat{Y}_i - Y_i)^2.$$

Mean square error *is measured as the average of squared* difference between predictions and actual observations. The mean squared error function is widely used as it is simple, continuous and differentiable. MSE has nice mathematical properties which makes it easier to calculate gradients.MSE is not Robust to outlier.

The lesser the value of MSE, the better are the predictions.

➢ Categorical cross entropy:

Categorical crossentropy is a loss function that is used in multi-class classification tasks. These are tasks where an example can only belong to one out of many possible categories, and the model must decide which one. Formally, it is designed to quantify the difference between two probability distributions.

$$\text{Loss} = -\sum_{i=1}^{\text{output size}} y_i \cdot \log \hat{y}_i$$

It will calculate the average difference between the actual and predicted probability distributions for all classes in the problem. The score is minimized and a perfect cross-entropy value is 0.

The target need to be one-hot encoded this makes them directly appropriate to use with the categorical cross-entropy loss function.

BACKWARD PROPAGATION:

Backward Propagation is the preferable method of adjusting or correcting the weights to reach the minimized loss function.

Adam's formula:

New weight=current weight-[learning weight * slope]

5.CONVOLUTION NEURAL NETWORK

5.1 INTRODUCTION TO CNN

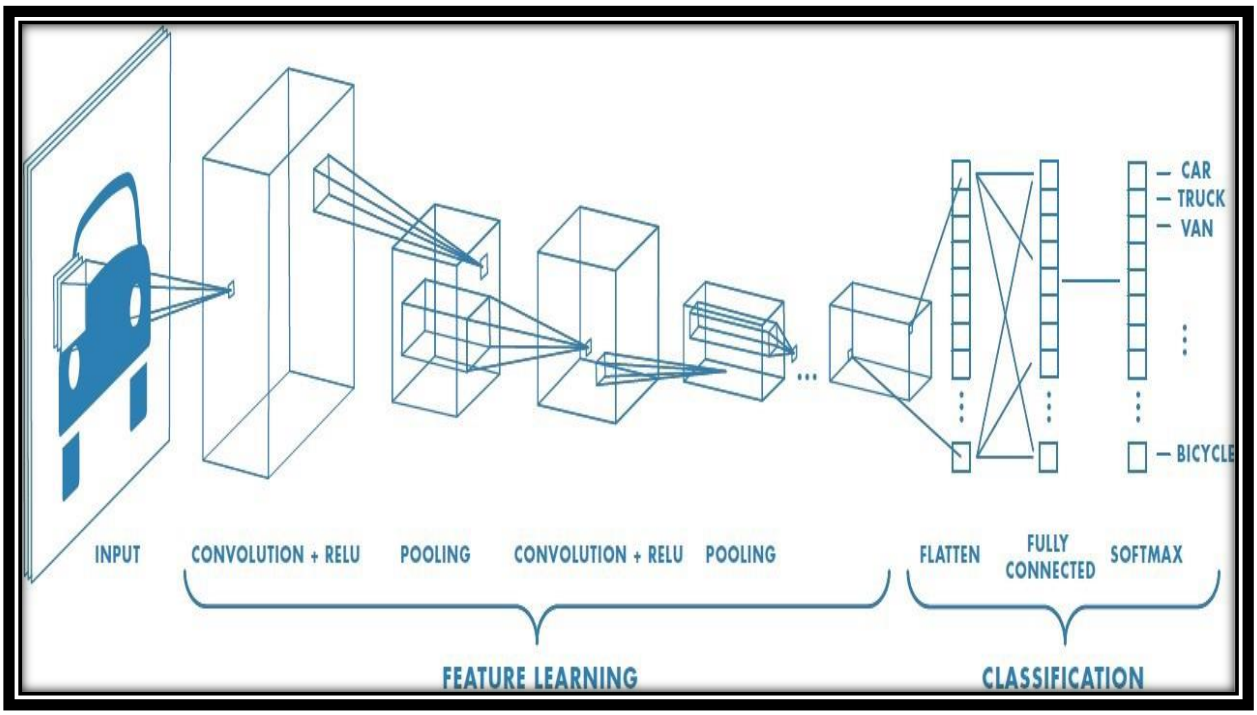
In the past few decades, Deep Learning has proved to be a very powerful tool because of its ability to handle large amounts of data.

The interest to use hidden layers has surpassed traditional techniques, especially in pattern recognition. One of the most popular deep neural networks is Convolutional Neural Networks.

The agenda for this field is to enable machines to view the world as humans do, perceive it in a similar manner and even use the knowledge for a multitude of tasks such as Image & Video recognition, Image Analysis & Classification, Media Recreation, Recommendation Systems, Natural Language Processing, etc.

The advancements in Computer Vision with Deep Learning has been constructed and perfected with time, primarily over one particular algorithm — a **Convolutional Neural Network**.

5.2 ARCHITECTURE OF CNN



There are three main layers of CNN:

- Convolution Layer
- Pooling Layer
- Fully Connected Layer
- Drop out
- Activation Function

1.Convolution Layer :

This layer is the first layer that is used to extract the various features from the input images. In this layer, the mathematical operation of convolution is performed between the input image and a filter of a particular size MxM.

By sliding the filter over the input image, the dot product is taken between the filter and the parts of the input image with respect to the size of the filter (MxM).

The output is termed as the Feature map which gives us information about the image such as the corners and edges. Later, this feature map is fed to other layers to learn several other features of the input image.

2.Pooling Layer:

In most cases, a Convolutional Layer is followed by a Pooling Layer. The primary aim of this layer is to decrease the size of the convolved feature map to reduce the computational costs.

This is performed by decreasing the connections between layers and independently operates on each feature map. Depending upon method used, there are several types of pooling operations.

In Max Pooling, the largest element is taken from feature map. Average Pooling calculates the average of the elements in a predefined sized Image section.

The total sum of the elements in the predefined section is computed in Sum Pooling. The Pooling Layer usually serves as a bridge between the Convolutional Layer and the FC Layer

3.Fully Connected Layer:

The Fully Connected (FC) layer consists of the weights and biases along with the neurons and is used to connect the neurons between two different layers.

These layers are usually placed before the output layer and form the last few layers of a CNN Architecture.

In this, the input image from the previous layers are flattened and fed to the FC layer. The flattened vector then undergoes few more FC layers where the mathematical functions operations usually take place.

In this stage, the classification process begins to take place.

4.Dropout :

Usually, when all the features are connected to the FC layer, it can cause overfitting in the training dataset. Overfitting occurs when a particular model works so well on the training data causing a negative impact in the model's performance when used on a new data.

To overcome this problem, a dropout layer is utilised wherein a few neurons are dropped from the neural network during training process resulting in reduced size of the model.

On passing a dropout of 0.3, 30% of the nodes are dropped out randomly from the neural network.

5.3 ACTIVATION FUNCTIONS USED IN CNN:

Finally, one of the most important parameters of the CNN model is the activation function. They are used to learn and approximate any kind of continuous and complex relationship between variables of the network.

In simple words, it decides which information of the model should fire in the forward direction and which ones should not at the end of the network.

It adds non-linearity to the network. There are several commonly used activation functions such as the ReLU, Softmax, tanH and the Sigmoid functions.

Each of these functions have a specific usage. For a binary classification CNN model, sigmoid and softmax functions are preferred an for a multi-class classification, generally softmax is used.

6.1 ROLE OF KERAS:

Keras is a high-level neural networks API, capable of running on top of Tensorflow, Theano, and CNTK.

It enables fast experimentation through a high level, user-friendly, modular and extensible API. Keras can also be run on both CPU and GPU.

6.2 ROLE OF OPENCV:

Opencv is an open source library which is very useful for **computer vision applications** such as video analysis, CCTV footage analysis and image analysis

6.3 ROLE OF OS MODULE:

The OS module in python provides functions for creating and removing a directory(Folder)

7.MINI-PROJECT DESCRIPTION :

Make a heart disease predictive model to predict on the “target” column of the dataset.

CODE:

Step 1 :Read the csv file and convert into the dataframe.

```
[ ] df=pd.read_csv("heart.csv")
```

```
[ ] df
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
298	57	0	0	140	241	0	1	123	1	0.2	1	0	3	0
299	45	1	3	110	264	0	1	132	0	1.2	1	0	3	0
300	68	1	0	144	193	1	1	141	0	3.4	1	2	3	0
301	57	1	0	130	131	0	1	115	1	1.2	1	1	3	0
302	57	0	1	130	236	0	0	174	0	0.0	1	1	2	0

303 rows × 14 columns

```
df.shape
```

```
(303, 14)
```

Step 2: We have to clean the data.This can be done in two ways

- isnull method
- sum method

```
[ ] df.isnull()
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	False	False	False	False	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
298	False	False	False	False	False	False	False	False	False	False	False	False	False	False
299	False	False	False	False	False	False	False	False	False	False	False	False	False	False
300	False	False	False	False	False	False	False	False	False	False	False	False	False	False
301	False	False	False	False	False	False	False	False	False	False	False	False	False	False
302	False	False	False	False	False	False	False	False	False	False	False	False	False	False

303 rows × 14 columns

```
df.isnull().sum(axis="rows")
```

```
age         0
sex         0
cp          0
trestbps    0
chol        0
fbs         0
restecg     0
thalach     0
exang       0
oldpeak     0
slope       0
ca          0
thal        0
target      0
dtype: int64
```

Here the data is already clean

```
df.shape
```

```
(303, 14)
```

BUILD THE MODEL:

1. SPECIFY THE ARCHITECTURE

(a) Type of the Model -> Sequential Model

(b) Type of layers involved in the model -> Dense layers

(c) Number of layers involved in the model -> 3

(d) Number of neurons in each layer ->

1. Input layer = 13 neurons
2. Hidden layer = 60 neurons
3. Output layer = 2 neurons

(e) Activation Function -> relu(Rectified Linear activation function) and softmax

```
from keras.layers import Dense
```

```
from keras.models import Sequential
```

```
x.shape
```

```
(303, 13)
```

```
model=Sequential()
model.add(Dense(60,activation="relu",input_shape=(13,)))
model.add(Dense(2,activation="softmax"))
```

2. COMPILE THE MODEL

(a) Loss Function -> Categorical crossentropy

(b) Optimize the weights -> adam formula

```
model.compile(optimizer="adam",loss="categorical_crossentropy",metrics=["accuracy"])
```

3.FIT THE MODEL

(a) Pass the data(Features and Target)

(b) Iterations of forward and Backward Propagation.

```
model.fit(x,y,epochs=30)
```

```
Epoch 2/30
10/10 [==============================] - 0s 2ms/step - loss: 0.3792 - accuracy: 0.8416
Epoch 3/30
10/10 [==============================] - 0s 2ms/step - loss: 0.4559 - accuracy: 0.7822
Epoch 4/30
10/10 [==============================] - 0s 2ms/step - loss: 0.4622 - accuracy: 0.8020
Epoch 5/30
10/10 [==============================] - 0s 2ms/step - loss: 0.3933 - accuracy: 0.8218
Epoch 6/30
10/10 [==============================] - 0s 2ms/step - loss: 0.4347 - accuracy: 0.7921
Epoch 7/30
10/10 [==============================] - 0s 2ms/step - loss: 0.3907 - accuracy: 0.8251
Epoch 8/30
10/10 [==============================] - 0s 2ms/step - loss: 0.3941 - accuracy: 0.8152
Epoch 9/30
10/10 [==============================] - 0s 2ms/step - loss: 0.4335 - accuracy: 0.7921
Epoch 10/30
10/10 [==============================] - 0s 2ms/step - loss: 0.3880 - accuracy: 0.8416
Epoch 11/30
10/10 [==============================] - 0s 2ms/step - loss: 0.4797 - accuracy: 0.7723
Epoch 12/30
```

```
10/10 [==============================] - 0s 2ms/step - loss: 0.4154 - accuracy: 0.8119
Epoch 16/30
10/10 [==============================] - 0s 2ms/step - loss: 0.3935 - accuracy: 0.8119
Epoch 17/30
10/10 [==============================] - 0s 2ms/step - loss: 0.4554 - accuracy: 0.8086
Epoch 18/30
10/10 [==============================] - 0s 2ms/step - loss: 0.3926 - accuracy: 0.8218
Epoch 19/30
10/10 [==============================] - 0s 2ms/step - loss: 0.4844 - accuracy: 0.7888
Epoch 20/30
10/10 [==============================] - 0s 2ms/step - loss: 0.4301 - accuracy: 0.8119
Epoch 21/30
10/10 [==============================] - 0s 2ms/step - loss: 0.4121 - accuracy: 0.7756
Epoch 22/30
10/10 [==============================] - 0s 2ms/step - loss: 0.3597 - accuracy: 0.8647
Epoch 23/30
10/10 [==============================] - 0s 2ms/step - loss: 0.4148 - accuracy: 0.8317
Epoch 24/30
10/10 [==============================] - 0s 2ms/step - loss: 0.3680 - accuracy: 0.8416
Epoch 25/30
10/10 [==============================] - 0s 2ms/step - loss: 0.3619 - accuracy: 0.8548
Epoch 26/30
10/10 [==============================] - 0s 2ms/step - loss: 0.3960 - accuracy: 0.8053
Epoch 27/30
10/10 [==============================] - 0s 2ms/step - loss: 0.3854 - accuracy: 0.8251
Epoch 28/30
10/10 [==============================] - 0s 2ms/step - loss: 0.3689 - accuracy: 0.8251
Epoch 29/30
10/10 [==============================] - 0s 2ms/step - loss: 0.3905 - accuracy: 0.8185
Epoch 30/30
```

4. PREDICT ON SAMPLE DATA

```
x
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2
...	...	...	...	...	...	...	...	...	...	...	...	...	...
298	57	0	0	140	241	0	1	123	1	0.2	1	0	3
299	45	1	3	110	264	0	1	132	0	1.2	1	0	3
300	68	1	0	144	193	1	1	141	0	3.4	1	2	3
301	57	1	0	130	131	0	1	115	1	1.2	1	1	3
302	57	0	1	130	236	0	0	174	0	0.0	1	1	2

303 rows × 13 columns

```
[ ] data=pd.DataFrame({"age":[75],"sex":[1],"cp":[2],"trestbps":[162],"chol":[230],"fbs":[1],"restecg":[1],"thalach":[180],"exang":[1],"oldpeak":[3.2],"slope":[2],"
```

```
[ ] data
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal
0	75	1	2	162	230	1	1	180	1	3.2	2	3	2

```
[ ] model.predict(data)
```

```
array([[0.9179939 , 0.08200606]], dtype=float32)
```

```
[ ] data=pd.DataFrame({"age":[30],"sex":[0],"cp":[4],"trestbps":[118],"chol":[200],"fbs":[0],"restecg":[1],"thalach":[130],"exang":[0],"oldpeak":[0.3],"slope":[1],"
```

```
data
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal
0	30	0	4	118	200	0	1	130	0	0.3	1	1	3

```
[ ] model.predict(data)
```

```
array([[0.02856844, 0.97143155]], dtype=float32)
```

Conclusion:

Here according to the data given ,the model can predict whether the person can have chances of heart disease or not.In first case,there are attributes whose value is abnormally high which results in heart attack.The model predicted that there is a probability of 91% heart attack.Similarly in the second case,the attributes values are normal .The model predicted that there is a probability of 97% of not having heart attacks.

8.MAJOR PROJECT DESCRIPTION:

Build a neural network to classify “mask” or “without_mask “ images.

Using opencv,predict on each image in a webcam video(whether it belongs to “mask” or “without-mask” image).Round a rectangle on the detected face,and also put a text above that rectangle ,depicting whether it is a masked or a non-masked image.

CODE:

```
[ ] import cv2
```

```
[ ] import os
```

```
import numpy as np
```

```
[ ] import tensorflow as tf
```

```
[ ]
    path=r'C:/Users/pranay/Desktop/Masks/'
    Folder_name=os.listdir(path)
    labels=[i for i in range(len(Folder_name))]
    label_dict=dict(zip(Folder_name,labels))
    print(Folder_name)
    print(labels)
    print(label_dict)
```

```
['without_mask', 'with_mask']
[0, 1]
{'without_mask': 0, 'with_mask': 1}
```

```
h=32
w=32
features=[]
target=[]
for i in Folder_name:
    folder_path=os.path.join(path,i)
    img_names=os.listdir(folder_path)

    for j in img_names:
        img_path=os.path.join(folder_path,j)
        img=cv2.imread(img_path)
        try:
            gray=cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
            resized=cv2.resize(gray,(h,w))
            features.append(resized)
            target.append(label_dict[i])

        except Exception as e:
            print("Exception occured",e)
```

```
[ ] features=np.array(features)/255.0
    features=np.reshape(features,(features.shape[0],h,w,1))
    target=np.array(target)
```

```
[ ] features.shape
```

```
(1376, 32, 32, 1)
```

```
[ ] target.shape
```

```
(1376,)
```

Split the data into training and testing

```
[ ] from sklearn.model_selection import train_test_split
```

```
[ ] train_features,test_features,train_target,test_target=train_test_split(features,target,test_size=0.2)
```

```
[ ] train_features.shape
```

```
(1100, 32, 32, 1)
```

Image Augmentation Techniques

1. Rotation
2. Width Shifting
3. Height Shifting
4. Shearing
5. Zoom

```
[ ] from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

```
[ ] dataGen=ImageDataGenerator(rotation_range=10,width_shift_range=0.1,height_shift_range=0.1,shear_range=0.1,zoom_range=0.2)
```

```
[ ] dataGen.fit(train_features)
```

```
[ ] id=dataGen.flow(train_features,train_target,batch_size=20)
```

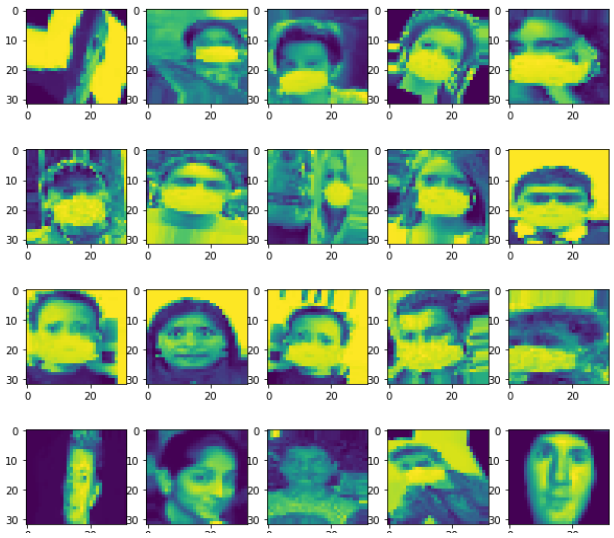
```
x_batch,y_batch=next(id)
```

```
[ ] x_batch.shape

    (20, 32, 32, 1)
```

```
[ ] import matplotlib.pyplot as plt
```

```
[ ] plt.figure(figsize=(10,10))
    for i in range(20):
        plt.subplot(4,5,i+1)
        plt.imshow(x_batch[i].reshape(32,32))
    plt.show()
```

[ ]



Since the target column has finite set of possibilities ,we have to build classification model.

1.Increase the number of neurons in output layer(One-Hot Encoding Technique)
2.Convert the values in neurons of output layer to probability values(Using activation function-softmax)

```
from tensorflow.keras.utils import to_categorical
train_target=to_categorical(train_target)
```

```
train_target.shape
```

```
(1100, 2)
```

BUILD THE MODEL

STEP 1: SPECIFY THE ARCHITECTURE

```
from tensorflow.keras.layers import Dense,Flatten,Conv2D,MaxPooling2D,Dropout
```

```
from tensorflow.keras.models import Sequential
```

```
model=Sequential()
model.add(Conv2D(60,(3,3),activation="relu",input_shape=(32,32,1)))
model.add(Conv2D(60,(3,3),activation="relu"))
model.add(MaxPooling2D((2,2)))
model.add(Conv2D(60,(3,3),activation="relu"))
model.add(Conv2D(60,(3,3),activation="relu"))
model.add(MaxPooling2D((2,2)))
model.add(Flatten())
model.add(Dropout(0.5))
model.add(Dense(500,activation="relu"))
model.add(Dense(2,activation="softmax"))
```

STEP 2: COMPILE THE MODEL

```
from tensorflow.keras.optimizers import Adam
```

+ Code + Text

```
model.compile(Adam(lr=0.001),loss='categorical_crossentropy',metrics=["accuracy"])
```

STEP 3: TRAIN THE MODEL ON THE TRAINING DATASET

+ Code + Text

```
model.fit_generator(dataGen.flow(train_features,train_target,batch_size=20),epochs=20)
```

```
WARNING:tensorflow:From <ipython-input-28-fed536c9e338>:1: Model.fit_generator (from tensorflow.python.keras.engine.training) is deprecated and will be removed i
Instructions for updating:
Please use Model.fit, which supports generators.
Epoch 1/20
55/55 [==============================] - 4s 67ms/step - loss: 0.6528 - accuracy: 0.5964
Epoch 2/20
55/55 [==============================] - 4s 67ms/step - loss: 0.4202 - accuracy: 0.8155
Epoch 3/20
55/55 [==============================] - 4s 68ms/step - loss: 0.2606 - accuracy: 0.8964
Epoch 4/20
55/55 [==============================] - 4s 68ms/step - loss: 0.2198 - accuracy: 0.9127
Epoch 5/20
55/55 [==============================] - 4s 68ms/step - loss: 0.1804 - accuracy: 0.9336
Epoch 6/20
55/55 [==============================] - 4s 68ms/step - loss: 0.1698 - accuracy: 0.9436
Epoch 7/20
55/55 [==============================] - 4s 68ms/step - loss: 0.1637 - accuracy: 0.9436
Epoch 8/20
55/55 [==============================] - 4s 69ms/step - loss: 0.1329 - accuracy: 0.9509
Epoch 9/20
55/55 [==============================] - 4s 70ms/step - loss: 0.1113 - accuracy: 0.9636
Epoch 10/20
55/55 [==============================] - 4s 71ms/step - loss: 0.1144 - accuracy: 0.9582
```

```
Epoch 11/20
55/55 [==============================] - 4s 72ms/step - loss: 0.1064 - accuracy: 0.9682
Epoch 12/20
55/55 [==============================] - 4s 71ms/step - loss: 0.1635 - accuracy: 0.9400
Epoch 13/20
55/55 [==============================] - 4s 71ms/step - loss: 0.1156 - accuracy: 0.9664
Epoch 14/20
55/55 [==============================] - 4s 71ms/step - loss: 0.1002 - accuracy: 0.9655
Epoch 15/20
55/55 [==============================] - 4s 71ms/step - loss: 0.0995 - accuracy: 0.9709
Epoch 16/20
55/55 [==============================] - 4s 71ms/step - loss: 0.0814 - accuracy: 0.9764
Epoch 17/20
55/55 [==============================] - 4s 71ms/step - loss: 0.1060 - accuracy: 0.9636
Epoch 18/20
55/55 [==============================] - 4s 71ms/step - loss: 0.1307 - accuracy: 0.9509
Epoch 19/20
55/55 [==============================] - 4s 71ms/step - loss: 0.0819 - accuracy: 0.9755
Epoch 20/20
55/55 [==============================] - 4s 71ms/step - loss: 0.1304 - accuracy: 0.9518
<tensorflow.python.keras.callbacks.History at 0x24879da9400>
```

```
[ ] from tensorflow.keras.models import model_from_json
```

```
file=open('C:/Users/pranay/Desktop/MasksTest.json',"w")
model_json=model.to_json()
file.write(model_json)
file.close()
```

```
[ ] model.save_weights('C:/Users/pranay/Desktop/MasksTestWeight.h5')
```

```
[ ] file=open('C:/Users/pranay/Desktop/MasksTest.json','r')
    loaded_model_json=file.read()
    loaded_model=model_from_json(loaded_model_json)
```

```
[ ] loaded_model.load_weights('C:/Users/pranay/Desktop/MasksTestWeight.h5')
```

STEP 4:TEST THE MODEL BY USING IT FOR PREDICTIONS

```
[ ] face_clsfr=cv2.CascadeClassifier('C:/Users/pranay/Desktop/haarcascade_frontalface_default.xml')
```

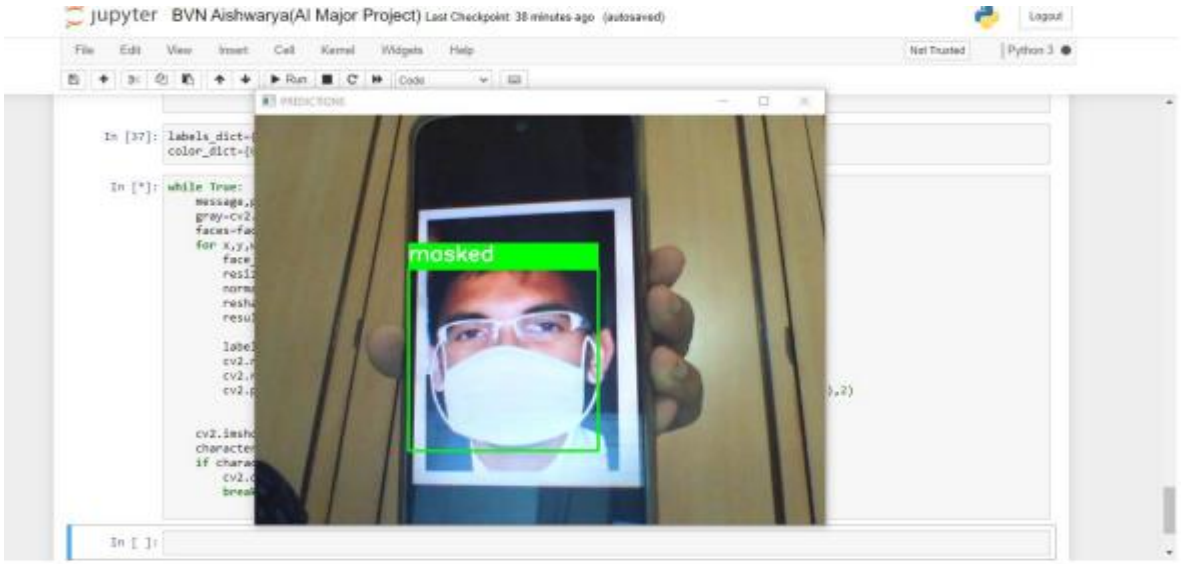
```
[ ] cap=cv2.VideoCapture(0)
```

```
[ ] labels_dict={0:'non-masked',1:'masked'}
    color_dict={0:(0,0,255),1:(0,255,0)}
```

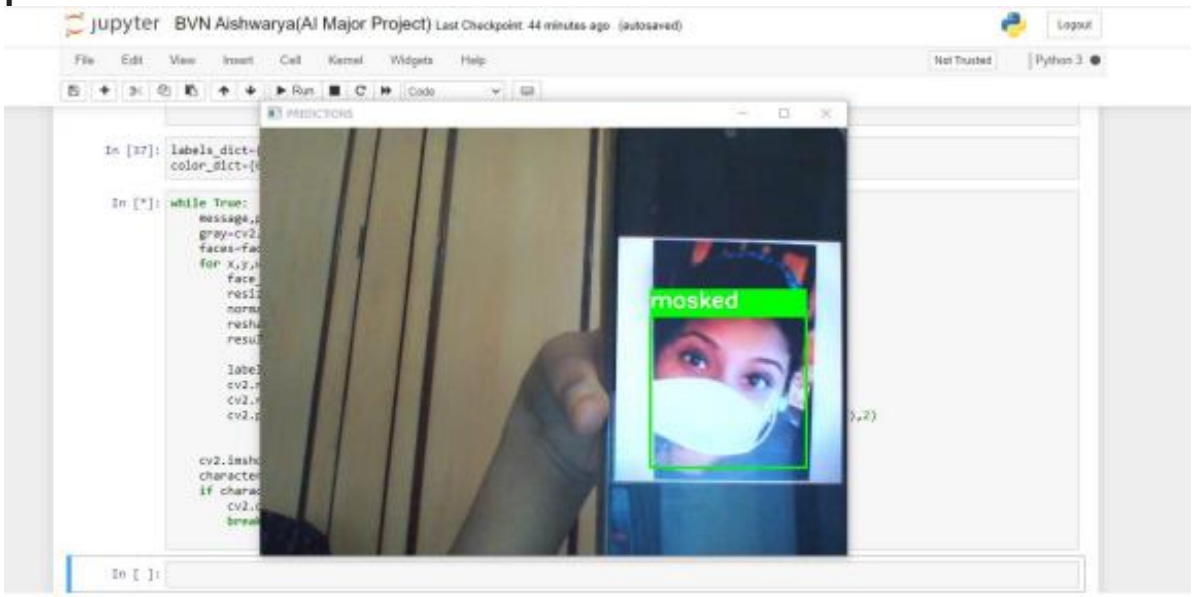
```
[ ] while True:
        message,pixels=cap.read()
        gray=cv2.cvtColor(pixels,cv2.COLOR_BGR2GRAY)
        faces=face_clsfr.detectMultiScale(gray,1.3,5)
        for x,y,w,h in faces:
            face_img=gray[y:y+w,x:x+w]
            resized=cv2.resize(face_img,(32,32))
            normalized=resized/255.0
            reshaped=np.reshape(normalized,(1,32,32,1))
            result=loaded_model.predict(reshaped)

            label=np.argmax(result,axis=1)[0]
            cv2.rectangle(pixels,(x,y),(x+w,y+h),color_dict[label],2)
            cv2.rectangle(pixels,(x,y-30),(x+w,y),color_dict[label],-1)
            cv2.putText(pixels,labels_dict[label],(x,y-10),cv2.FONT_HERSHEY_SIMPLEX,0.8,(255,255,255),2)


        cv2.imshow('PREDICTIONS',pixels)
        characterAscii=cv2.waitKey(1)
        if characterAscii==ord("p"):
            cv2.destroyAllWindows()
            break
```

OUTPUT:

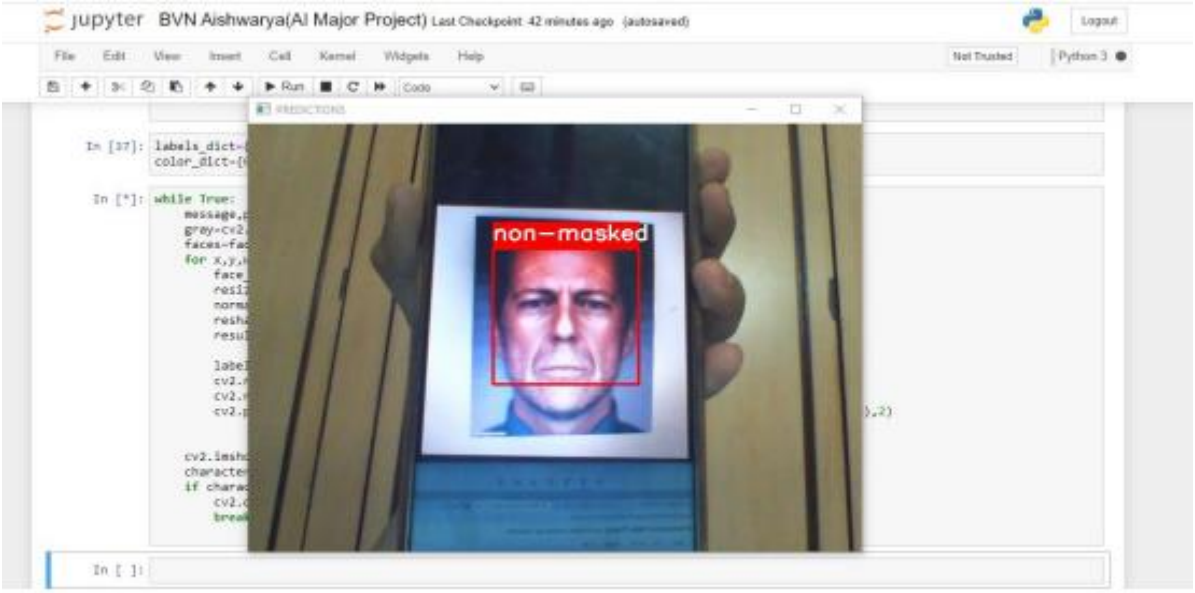
CASE 1: MODEL PREDICTS MASKED IMAGES



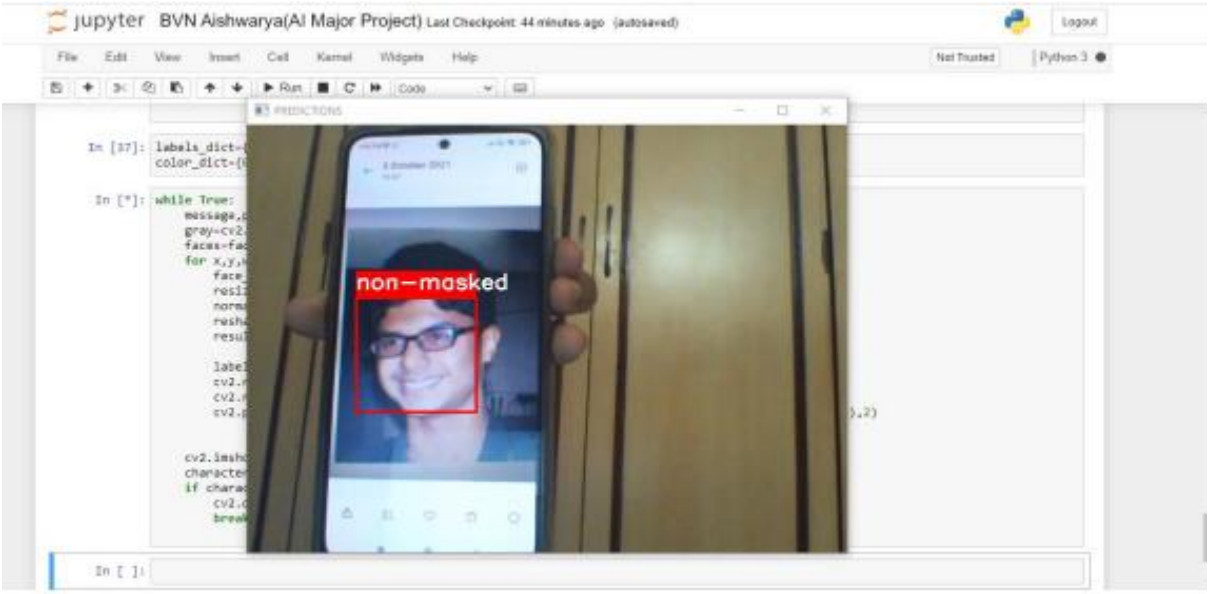
CNN helps the model to predict the images in terms of edges not positions.



CASE 2: MODEL PREDICTS NON-MASKED IMAGES



CNN helps the model to predict the images in terms of edges not positions.



9.CONCLUSION

Artificial Intelligence is a broad branch of computer science that is focused on a machine's capability to produce rational behavior from external inputs.The goal of AI is to create systems that can perform tasks that would otherwise require human intelligence.

Neural networks reflect the behavior of the human brain ,allowing computer programs and solve common problems in the fields of AI ,machine learning and deep learning.

Convolution Neural Network is a popular deep learning technique for current visual recognition tasks.Like all deep learning techniques,CNN is very dependent on the size and quality of the training data.CNN takes the image's raw pixel data,trains the model ,then extracts the features automatically for better classification.

With the help of the AI fields , we can able to make heart disease predictive model in the mini project and CNN helps us to predict masked or without masked images.

10.REFERENCES:

- https://www.kaggle.com/ronitf/heart-disease-uci
- https://drive.google.com/drive/folders/1tvLtA8UvfvQO6dEZBfOJi0aoIVOIzHAT?usp=sharing