CHEST DISEASE DETECTION FROM X-RAY

Project report submitted in partial fulfillment of the requirement for the degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

By

Prerna Kanwar (171006)

UNDER THE GUIDANCE OF

Lt.Pragya Gupta

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

May 2021

TABLE OF CONTENTS

DECLARATION

I hereby declare that the work reported in the B.Tech Project Report entitled **“Chest disease detection from x-ray”** submitted at **Jaypee University of Information Technology, Waknaghat, India** is an authentic record of our work carried out under the supervision of **Lt. Pragya Gupta.** I have not submitted this work elsewhere for any other degree or diploma.

Prerna Kanwar

171006

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Lt. Pragya Gupta

Date: 15/06/2021

Head of the Department/Project Coordinator

ACKNOWLEDGEMENT

Knowledge, energy and time are the resources in the completion of this project but the most requisite is the proper guidance. I would like to express our deepest appreciation to all those who have been helping us throughout the project and without whom this project would have been a very difficult task. I am highly indebted to our respected project guide, **Lt. Pragya Gupta** for her valuable suggestions, ideas and constant supervision and also for providing necessary information throughout the project.

I want to express gratitude towards **Dr. Rajiv Kumar**, Head of the Department, Department of ECE, Jaypee University of Information Technology, Waknaghat for providing us this opportunity to make our project.

I also want to thank all the panel members for their valuable suggestion in all viva-voce during evaluation from which I got to learn a lot.

No thanks can counter out indebtedness to our parents who have been with us throughout. I thank them from core of our hearts.

LIST OF ACRONYMS AND ABBREVIATIONS

1.	ML	Machine Learning
2.	PHC	Primary health care
3.	CXR	Chest radiography
4.	DR	Digital radiography
5.	MRI	Magnetic resonance imaging
6.	EXP	Exponential
7.	NIH	National Institutes of Health
8.	SVM	Support Vector Machine
9.	KNN	k-nearest neighbors
10.	RAM	Random Access Memory
11.	GB	Gega Byte
12.	CONV	Convolutional Layer
13.	POOL	Pooling Layer
14.	FCL	Fully Connected Layer
15.	b/w	b/w
16.	no.	no.
17.	2D	two dimensional
18.	3D	three dimensional

LIST OF SYMBOLS

1.	∑	Summation
2.	+	Addition
3.	=	Equals to
4.	/	Division
5.	x	Input of dimension (p__l, 1)
6.	W	Weight of dimensions (p__l, n__l)
7.	p__l	previous layer neurons
8.	n__l	current layer neurons
9.	b	Bias vector of dimensions (p__l, 1)
10.	g	Activation function (mostly ReLU)
	P	Padding
12.	S	Stride
13.	&	and

LIST OF FIGURES

ABSTRACT

In this project, the aim was to use Deep Learning for the purpose of disease detection in chest X-ray images and predict what will be the results and give in our contribution to the medical society. The idea is to use conventional and deep learning approaches to detect COVID-19 in chest X-ray and also distinguish COVID-19 pneumonia from bacterial pneumonia. infections. The whole aim is to employ this technology for assistance in the diagnosis of patients infected by COVID-19 which has globally affected the world so badly. Main objective is to amalgamate medical knowledge and technical expertise to create application for public good,to enable faster and hassle free diagnosis to help physician in subcentres and PHCs.

CHAPTER 1

INTRODUCTION

1.1 Background and Motivation

Bestowing knowledge to machines and making them progressively self-sufficient and autonomous has been an important need for the mankind. It is our motto to let the machines take on exhausting, repetitive or perilous work with the goal that we can submit our opportunity to progressively innovative assignments.

To accomplish this goal, besides equipment advancement, we need the product that can confer machine the insight to accomplish the work and act autonomously. One of the essential occupations with respect to this is vision, aside from different sorts of intelligences like learning and cognitive thinking.

Many research has been already done to diagnose many chest diseases using various methodologies of artificial intelligence.Like multi-layer,probabilistic and generalized-regression neural networks had been used for diagnosis of chest diseases. For preprocessing of the image in image segmentation,histogram equalization was applied, and for classification the feed-forward neural network is used. For classifying medical diseases these research works have been efficiently utilized although the performance were not as productive and efficient as those of deep neural networks when it comes to accuracy , minimum square error achieved and computation time.

The accuracy of image-classification is increased by applying deep learning based system. This was motivation behind applying these networks to medical images for classification of diseases , and for efficiently extracting features that are useful which help in distinguishing different classes of image's.

There are many more technological advancements being carried in this field and we aim to study and provide a cognitive solution to the problem.

1.1.1 Problem Statement

In India chest diseases are the most common diseases for which patients visit doctors and specialists. Chest & respiratory diseases are increasing and major concern these days for both the doctors and the patients .

COVID-19 continues to have a devastating impact on the lives of people all over the world. It is important to screen the infected patients in a timely and cost-effective manner in order to combat this disease. The lungs are the most affected organs of human body, despite the fact that covid-19 disease can affect multiple organs . As a result , the lungs are the human body’s focal point for the detection of covid-19 implications. On this basis, radiological testing is one of the most viable steps toward achieving this aim, with chest X-Ray being the most readily available and least expensive alternative.

In many countries, there is currently a medical staffing shortage as a result of the new Covid-19 crisis. Since there are few radiologists than patients, treatment may be inefficient. Researchers are working to develop reliable and intelligent diagnostic methods that can adapt rapidly to the high demand for diagnosis of COVID-19. AI has played a key role in the imaging detection of new coronary pneumonia, which has the potential to increase diagnosis speed, accuracy, and precision significantly.

To address this pressing need, the point of this research is to build a novel approach of analyzing chest x-ray images that will be able to detect and differentiate covid-19 , bacterial pneumonia , healthy person. Better and faster detection of disease pattern and symptoms will help decide suitable medical care management strategies and set up essential health care services.

1.2 Chest diseases

There are various chest diseases that need early diagnosis. Pneumonia is caused due to the bacterial infection in the alveoli.Tuberculosis is another form of pneumonia but progress slowly and is caused due to the bacterial infection.Pleural effusion involves the fluid collection in the normally tiny pleura space between the lung and the chest wall. Immediate treatment is required because if the spaces becomes large & the fluid accumulates then breathing problems occur.

An economical,easy medical imaging and diagnostic method is chest radiography.[1] This is the most frequently used diagnosis tool in medical practice and has an vital role in diagnosis of lung diseases.[1]Chest X-rays are used by trained radiologists to detect diseases like tuberculosis,pneumonia,interstitial lung disease and early stages of lung cancer.

Even in less developed areas,DR machines are affordable as chest xray has the advantage of their less cost and are easy to use . Hence these are extensively used for detection ,diagnosis of the lung diseases like tuberculosis, pneumonia ,and interstitial lung disease.Abundant amount of information of patients health is contained in these chest xray. However to correctly interpret the information is a important challenge for medical professionalists.

Chest X-rays, CT scans of the lungs, ultrasound scans of the chest, needle biopsy of the lung, and MRI scans of the chest can all be used to diagnose pneumonia . X-ray imaging is favoured over CT imaging because CT imaging takes significantly longer than X-ray imaging and many underdeveloped regions which lack adequate high-quality CT scanners. X-rays, on the other hand, are the most popular and widely available diagnostic imaging tool, and they play an important role in clinical care and epidemiological research.

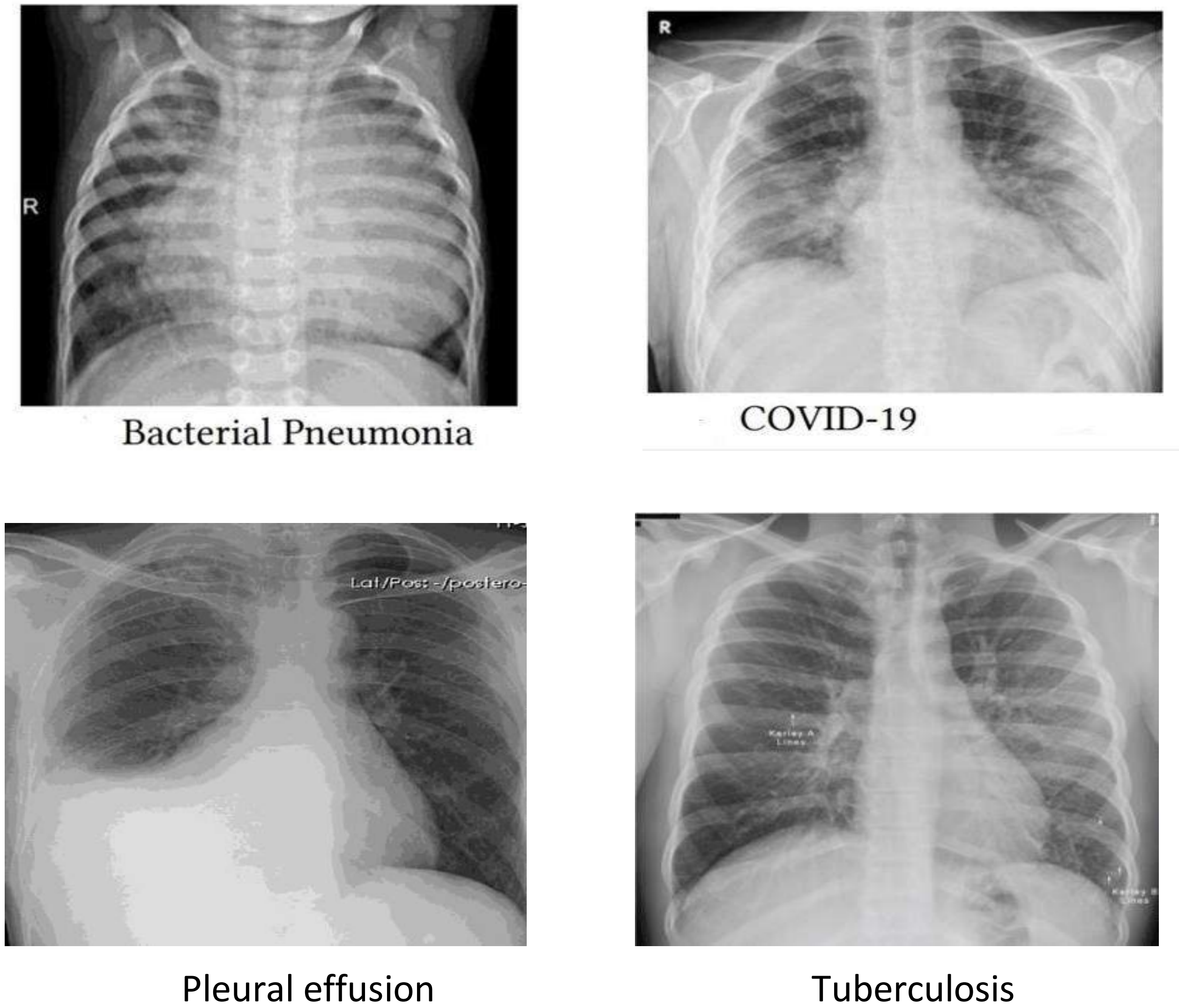


Figure 1.1: Various lung diseases

1.3 Medical Imaging

Medical imaging is regularly seen to assign the set of methods that noninvasively produce pictures of the inside part of the body. It is the procedures and processes used to make pictures of the human body for clinical purposes, for example, trying to reveal,analyze or inspect injury,or pathology.

Imaging is a valuable resource for musculoskeletal condition and is a invaluable apparatus for physical specialists when utilized suitably. Imaging,for example, X-ray, MRI, CT scan,are perfect representations of practical diagnostic imaging that encourages exact determination, visualization, intervention,and assesment of wounds and dysfunctions that physical therapist address consistently. It is imperative to realize when imaging is appropriate to use, as pointless imaging will waste monetary assets and increment potential for untimely medical procedure.

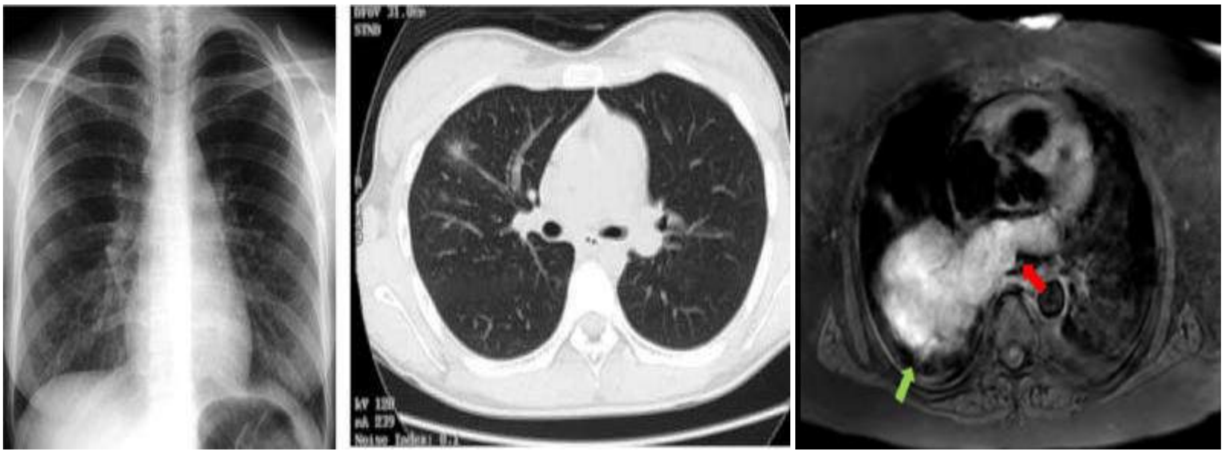


Figure 1.2: X-RAY vs CT SCAN vs MRI

1.3.1 X-ray

X-rays are the most utilized diagnostic imaging test and are generally accessible. Regardless of whether you require more modern body checks, it's feasible you'll get a x-ray first .X-ray creates 2D image.[1] They are used primarily to see bones and to detect cancers and pneumonia. They are commonly & widely available. X-ray uses radiation to produce images.

1.3.2 CT scan

A high-quality,detailed images of body are generated by CT scan. It is more powerful x-ray which takes a 360degree images of the spine,vertebrae,and internal organs. CT scan creates 3D images .[1] If compared to x-ray CT scans are more expensive and are not readily available at rural or small hospitals.

1.3.3 MRI

MRI is magnetic resonance imaging. It combines a strong magnet with radio waves. MRI creates 3D images and cross section images MRI scans are frequently used by doctors for diagnosis of joint or bone problems,also for assessing progress of any treatment,looking abnormalities in brain,and in evaluating pelvic pain .

1.4 Viral Pneumonia vs Bacterial Pneumonia

Pneumonia caused by a virus

In the early days of a viral pneumonia infection, congestion and cough with or without fever are common. [2]When a doctor listens to the lungs and hears no distinct breathing sounds on either side of the chest, a viral rather than bacterial origin is more likely. Viruses affect both sides of the lungs by causing a more uniform inflammatory response that results in a rise in cellular debris and mucus in previously open lung pockets. An xray of the lungs will reveal a more "diffuse" involvement.

Covid-19 pneuomonia

Ground-glass opacity is common, with the possibility of linear opacities (e.g., peripheral horizontal white lines) present in a typical case, resulting in slightly blurred lung markings. It should be noted that in extreme cases, due to the thick whiteness, the lung marks become invisible, a process known as consolidation. As a result, using chest -xray or combining it with laboratory and clinical evaluation, may be an effective method for detecting COVID-19 pneumonia early and accurately.

Pneumonia caused by bacteria.

Bacterial pneumonia is more common when a provider detects regular lung sounds on one side but none on the other.[2]Bacteria appear to invade one lobe or part of the lungs violently, creating a localised inflammatory response that takes over the cell’s that were previously filled with air. On an xray, one white condensed region or opacity will be visible, while the rest of the lung will be visualised as having normal air exchange.

CHAPTER 2
NEURAL NETWORKS

2.1 Introduction

With advancing technology there is a need for intelligent software to automate repetitive work, understand and translate speech or images, to be able to make medical diagnosis and support scientific research. The true challenge for AI is to solve problems that are easily performed by people but are often found hard to be described formally. Such problems are what humans can solve intuitively, like recognizing spoken words clearly,or objects in images. The presented solution is to allow computer to learn from experience (like humans do) and let it understand the hierarchy or order of concepts where every concept is defined by simpler concepts. This hierarchy empowers the computer to learn complex ideas by building them out of simpler ones.

Neural networks are one of the most useful programming paradigms. In conventional approach to programming, we break the problem into smaller parts and tell the computer what can be done, where only defined tasks that could be performed. In contrast to this, while building a neural network we expect the computer to learn from observing data and figuring out a solution to the problem by breaking into smaller parts itself.

2.2 What is a neural network?

Artificial neurons are information-processing units which are brain-inspired and replicate the way humans learning style. A neural network consists of neurons in input layers ,output layers and also in hidden layer which are functions capable of transforming the input into mathematical equations and process information that can be used by the output layer.

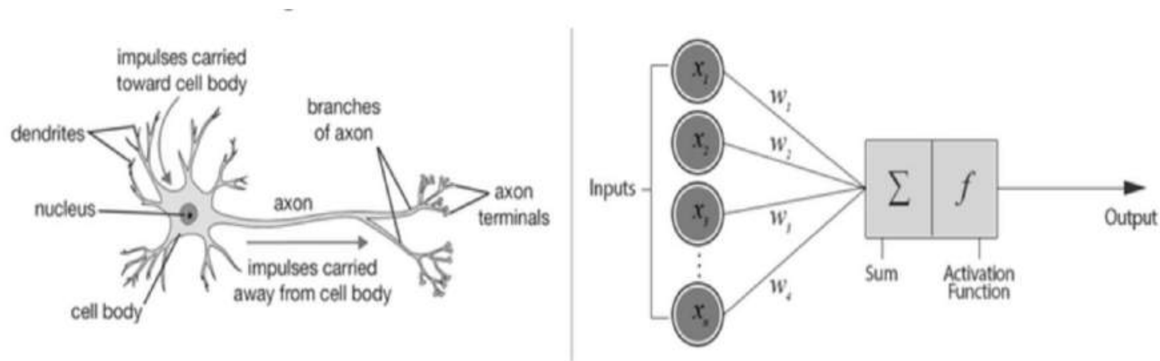


Figure 2.1: Biological Neuron vs Artificial Neural Network

Clustering and classification is where neural network helps us. It groups together unlabeled data as per similarities which are provided to it as examples for learning, and classifies data when it has a labeled dataset to train on. It is very useful to find patterns that are very complex for a traditional program to extract and it teaches the machine to recognize those pattern by matching them.

2.3Elements of Artificial Neural Network

Input (x(i))

These are external stimuli from the environment which are fed to the network. They might also come from output of another neural networks. They are real or discrete values from a set like {0,1}.

Weight (w(i))

The real-valued no. which helps in determining contribution of every input to the neurons weighted sum & eventually it's impact on the output. The main objective of a neural network training algorithms is to find a perfect set of weight values for every neuron for the given problem..

Threshold (u)

It is referred as a bias value. It is a real number that is added to the weighted sum before applying activation function.

Perceptrons

A perceptron take many binary input,x1,x2… and produces a single binary output. The perceptrons output is given by:

Output= {0 if $\sum_j w_j x_j \leq$ threshold

{1 if $\sum_j w_j x_j >$ threshold

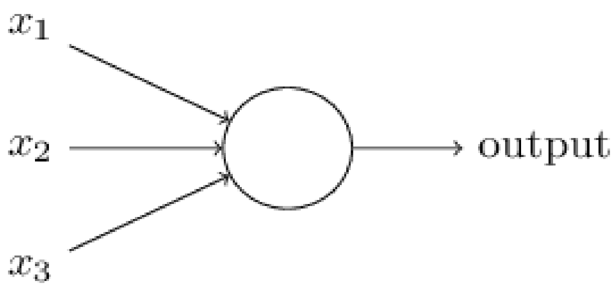


Figure 2.2: Perceptron with three inputs, x1,x2,x3

2.3.1 Classification in neural network

Every classification task is dependent on labeled datasets for classifying images. We need to pass correct labels i.e. human knowledge to the dataset so as to make neural net learn the correlation b/w data and labels. This process is called supervised learning.

Common examples of Supervised Learning are as follows:

- Face detection, identifying people in video/images.
- Identifying objects.
- Voice detection , transcribe speech to text.
- Classifying email/text as spam, recognize sentiment in text like for customer feedback.

2.3.2 Clustering in neural network

Clustering is the detection of the similarities. To detect similarities, labels are not required in deep learning. This domain is unsupervised learning .Majority data in the world is unlabeled data. Laws of machine learning states that the more data an algorithm is trained on,the more accurate it's results will be. Hence unsupervised learning can produce highly accurate models.

Some common unsupervised learning examples are as follows:

- Searching :Finding similar items by comparing documents, images and sounds.

- Anomaly detection: To detect unusual behavior or anomalies may turn out to be beneficial in many cases, which we want to identify and prevent like fraud.

2.3.3 Predictive Analytics or Regressions

Deeplearning is able to exhibit correlations b/w the image and name of a person. This astatic prediction. Deep learning is able to establish correlations b/w current, historical and future events if exposed to enough data. It is capable of running regression b/w past and future. Examples are like its very useful in case of hardware breakdown or health-breakdowns.

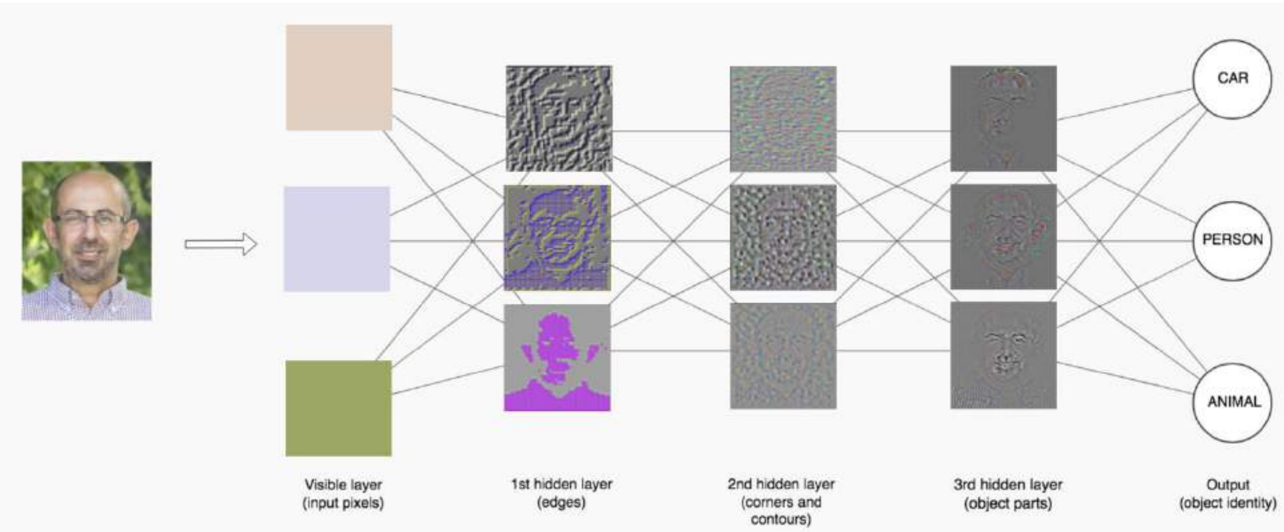


Figure 2.3: Neural network with multiple hidden layers to classify a human face.

2.4 Activation Function

These are mathematical equations which are used to find the output of neural-network. Every neuron in the network is attached to this activation function, and it tells it should be activated or not activated based on every neurons inputs.It helps normalizing o/p of every neuron b/w a range 1 & 0 or range -1 & 1.

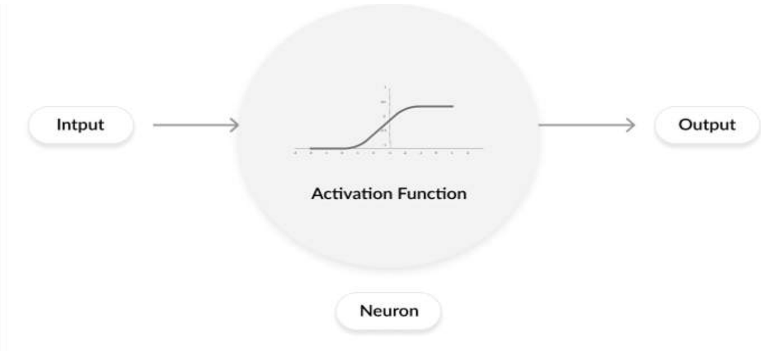


Figure 2.4: Activation function

2.4.1 Types of Activation Functions

1. Binary Step Function

The neuron activates when the value of input is above or below a certain value of the threshold and further send this to next layer. Multi-value outputs are not allowed with a step function like classifying input's into one of other categories is not supported.

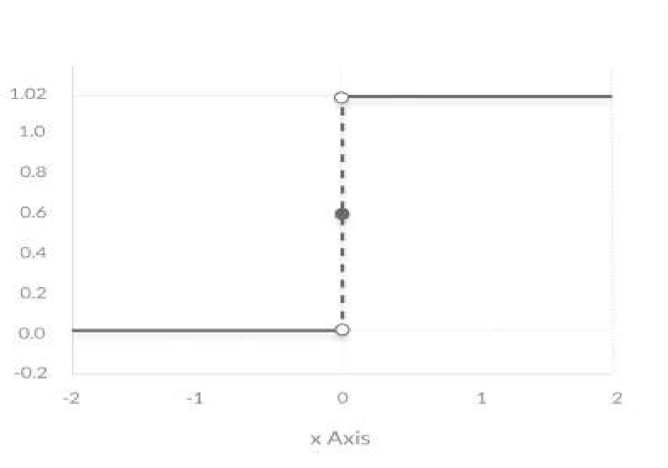


Figure 2.5: Binary Step Function

2. Linear Activation Function

This function is of A=CX form. The linear activation function takes input's, multiplied by weights for every neuron. An output signal which is proportional to input signal is created.[3]

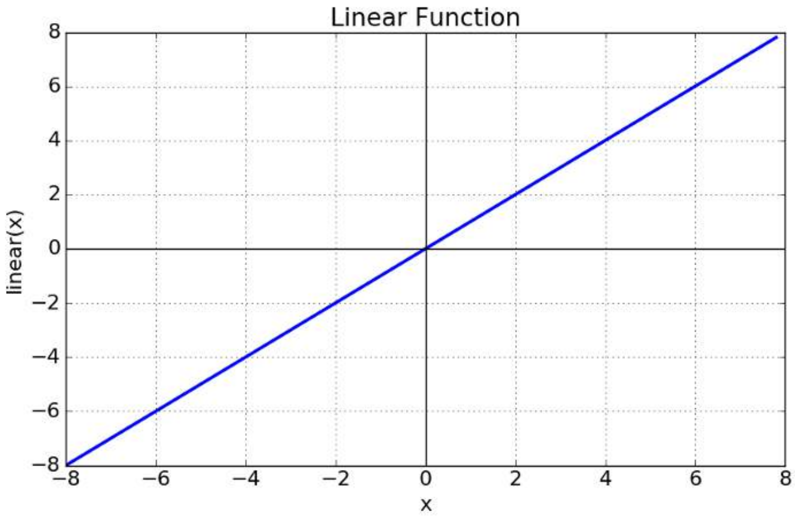


Figure 2.6: Linear function

3. Non-Linear Activation Functions

These are used now days for modern neural-network models.[3] These permit the model to make complex mappings b/w the input's and output's of the network that are important for the purpose of learning and demonstrating complex details like videos, images, sounds, and non linear data sets which have high dimensionality.

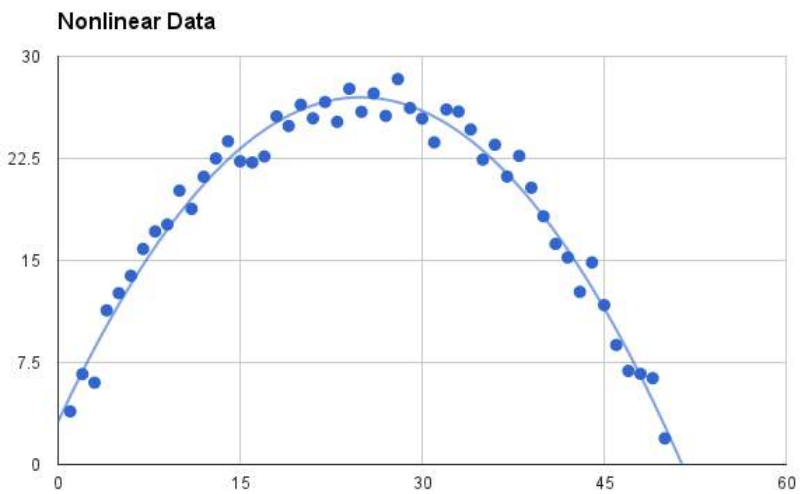


Figure 2.7: Nonlinear function

In a neural network, virtually every conceivable process can be interpreted as a functional computation, provided there is a non-linear activation function.

Non-Linear Activation functions :

1. Sigmoid / Logistic

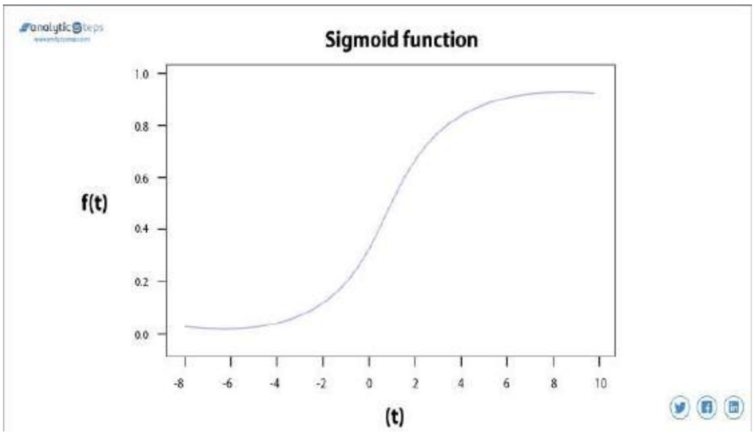


Figure 2.8: Sigmoid Function

Advantages:

- "Jumps" in output values are prevented as they have a smooth gradient
- The output of each neuron is normalized as output is bounded b/w 0 and 1.

Disadvantages:

- There is basically no change in predictions if the value of X is very high/low leading to vanishing gradient problem,which further results in slowing accurate predictions.
- There is no zero centered output.
- It is computationally expensive.

2. TanH / Hyperbolic Tangent

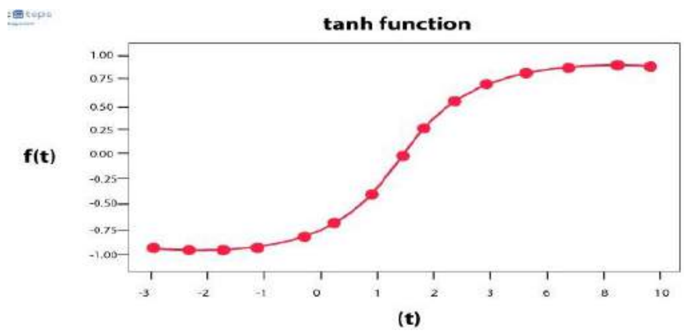


Figure 2.9: TanH Function

Advantages:

- It is zero centered and makes modeling of input easier.

Disadvantages:

- Similar as sigmoid function.

3. ReLU (Rectified Linear Unit)

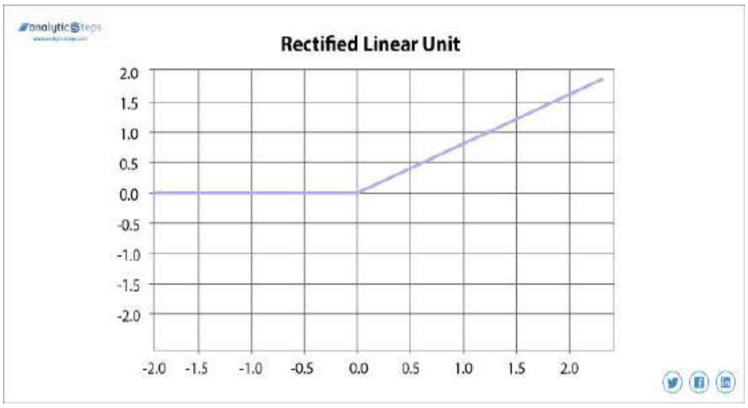


Figure 2.10: ReLU function

Advantages:

- Enables rapid integration of the network, thus it is computationally efficient.
- It has a function of derivatives and makes backpropagation possible. Though it looks similar to a linear function but it is a non linear.[3]

Disadvantages:

- The problem of dying ReLU: when input is approx 0 or negative, the network cannot learn and cannot do backpropagation the gradient of the function becomes zero.

4. Leaky ReLU

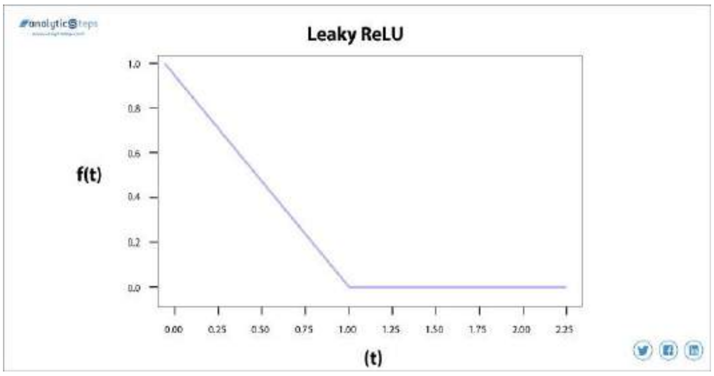


Figure 2.11: Leaky ReLU function

Advantages:

- It prevents the dying ReLU problem because in the negative field this has a slight positive slope, and even for negative values of the input it allows backpropagation.

Disadvantages:

- Results are inconsistent.

5. Softmax:

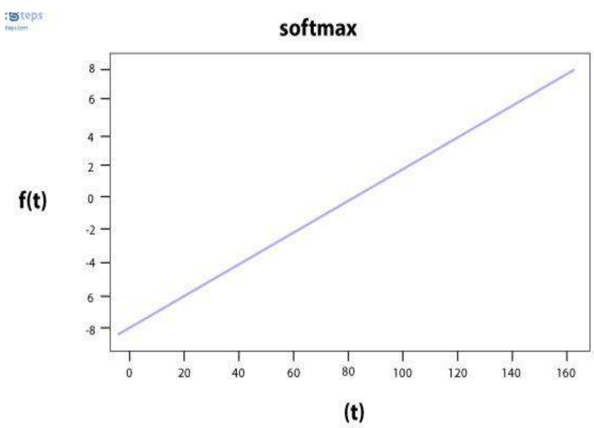


Figure 2.12: Softmax function

In neural networks Softmax function is used as an output function of the last layer like if network has N layers and is predicting more than 2 classes. The layer has the softmax activation function. This is important because the motive of the last layer is to change the score produced by neural network into values easily interpreted by us.

Advantages

- It is able to handle multiple classes unlike other activation function which haveonly one class.
- Standardizes the output's for every class b / w 0 and 1, then it is divided by their sum, that give probability that input value is in a particular class.
- In neural networks where there is need for classifying inputs into multiple categories there only for the output layer softmax is used.

It lets us express our inputs as a discrete distribution of probabilities.Mathematically, it is described as :

$$\mathbf{Softmax(x(i))=exp(x(i))\ /\ \sum(j)exp(x(j))}$$

For each input value in our input vector the Softmax value is the actual input exponent separated by a number of all input exponents. The negative inputs will be converted into nonnegative values due to exponential function. Every input is at interval (0,1).
Since the denominator is the same in each Softmax calculation, the values are equal to each other, thus making sure that their sum will be 1.

2.5 Architecture of Neural Network

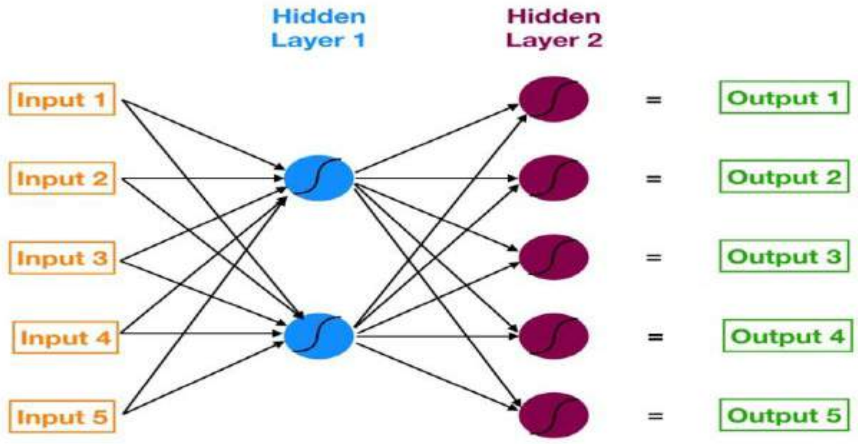


Figure 2.13: simple network comprised of 5 i/p 2 hidden layers & 5o/p

From the left hand side:[4]

1. Input layer of model.

2. First layer of neurons in hidden layer.

3. Second layer of neurons in hidden layer.

4. Model's output layer (prediction)

The arrows by which the dots are connected demonstrate how every neurons are interconnected.It also tells how data flows all way from the input layer to the output layer. The goal is to obtain predictions which closely match those target values.Simply,

Figure 2.14: Predicted probability

This is a single feature logistic regression (model is given only one variable i.e. X) expressed via a neural network.

Sigmoid(Br * X + Bo) = Predicted Probability

X : input. It is the only feature fed to the model to calculate a prediction.

B1 is the logistic regression 's predicted slope parameter — B1 shows to what extent the Log Odds increases with increase in X.
B0 is bias which is same as regression intercept term. The main distinction is, each neuron has its own bias concept in the neural networks (whereas in the regression, model have one intercept term).

The neuron also incorporates a function for sigmoid activation.[4] We use Sigmoid function to go from log-odds to probabilities.

1. When we apply sigmoid function to the quantity we obtain our predicted probability (B1*X + B0).

The complex neural networks are models with only more number of hidden layer, and this leads to many neurons b/w connections.[4] Such complex web of connections enables the neural network to "learn" the complex relationships of the data.

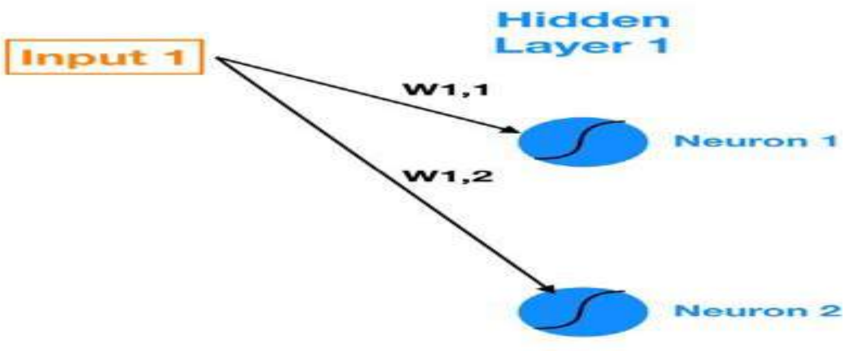


Figure 2.15: Connection b/w input1 and hidden layer1

To calculate the activation(outputs), we use the following formula:

$$Z = W1*X1 + W2*X2 + W3*X3 + W4*X4 + W5*X5 + b$$

(**W** denotes weight, **X** denotes input, b denotes bias).

In matrix form, it is: ***[Z] =[W] * [X] + [Bias]***

Here [W] is **n** x **m** matrix of weights. [X] is **m** x **1** matrix, [Bias] is **n** x **1** matrix and [Z] is **n x 1** matrix.[4] When we get [Z], we apply activation to every element of [Z].It gives us neuron outputs (activations), for the current layer.

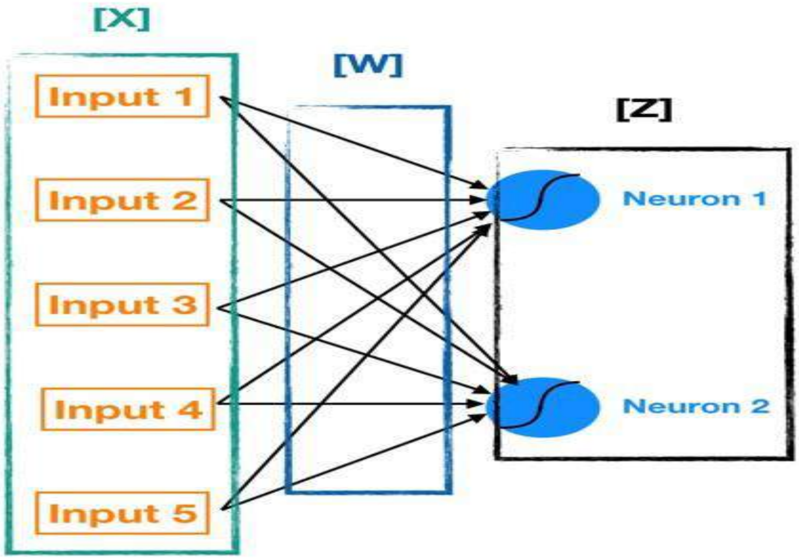


Figure 2.16: Intermediate output[Z]

2.6 Learning in Neural Network

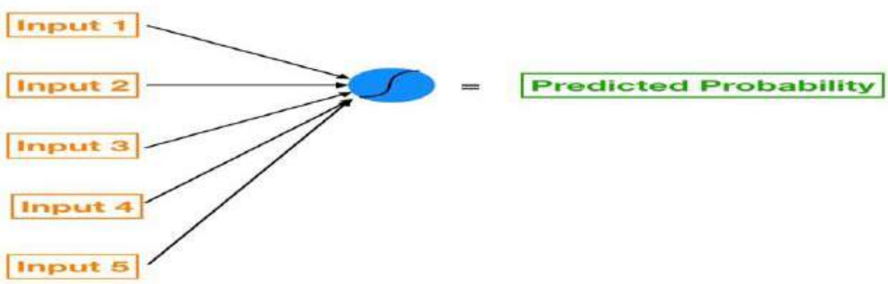


Figure 2.17: Learning in neural network

Inorder to move from inputs to outputs, calculate [Z] repeatedly for all the layers that follow. This is called forward propagation. Now evaluating the outputs and training the neural network is done.

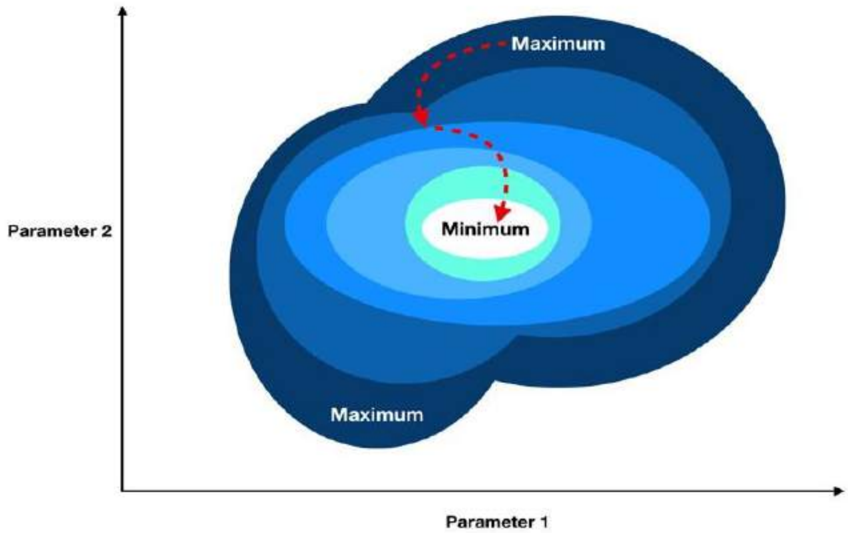


Figure 2.18: gradient descent

Backpropagation

The process of moving forward through the neural network is called forward propagation.[4] The reverse is Backpropagation. We transfer error backwards through our model, except in place of signal.

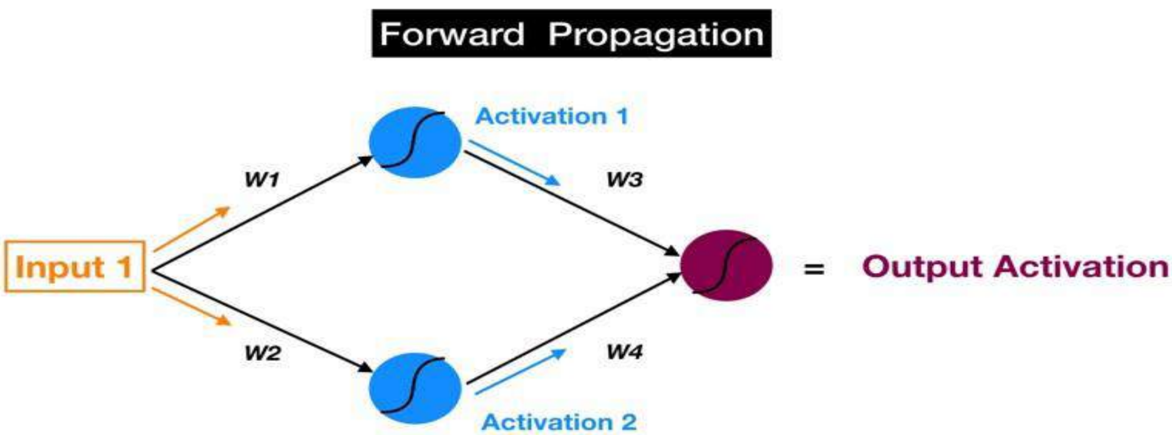


Figure 2.19: Forward propagation

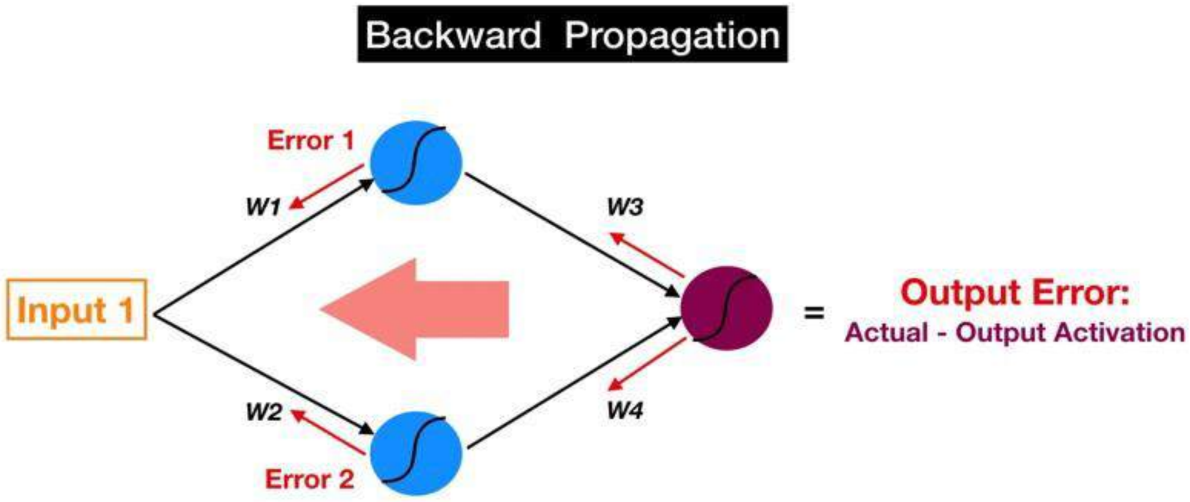


Figure 2.20: Backward propagation

Backpropagation can be summarized as:

- It is a method of moving the error backwards through layers, and to attribute the right amount of error to every neuron in network.
- Error due to a specific neuron is the fair estimate of how changing the neurons weight's and bias that affects the cost feature.

Factors affecting the network's optimization:

- **Optimization Algorithm Used:** The gradient descent is the optimisation algorithm most commonly used for neural networks.[4]Adam, RMSprop, Adagrad, Stochastic Gradient Descent are a few gradient descent variants that optimize the gradient update process and increase the performance of a model. There are several various methods for optimization apart from gradient descent, such as the Evolutionary Algorithms (EA) and the Particle Swarm .

- **Loss function Used:** Loss function plays an important part in the process of optimisation. A carefully chosen loss feature can help to enhance the training process.

- **Initialization of parameters:** The optimization process is greatly influenced by the paramenters' initial states. If initialization values are not correctly chosen, this may lead to divergence problems and may result in saddle points or local minimum.

- **Data size**: Sample size is a very significant component of neural network training. Large data sets can help in learning model parameter better and improve the process of optimization and generalization.

CHAPTER 3
CONVOLUTIONAL NEURAL NETWORKS

3.1 Introduction

The most popular deep learning architecture is Convolutional Neural Network (CNN) .It's a sub-category of neural networks which have proved to be very efficient in analyzing visual images.

For every image related problem CNN is the new go-to model. They are considered to be doing great in terms of accuracy . Recommender systems, natural language processing have successfully applied it . It automatically detects significant features without any human supervision that is the major advantage of CNN over its competition. It learns distinct features for every class by itself if given many pictures of dogs and cats.

They are computationally efficient as it utilizes special complexity, pooling operations and sharing of parameters is also done. Because of these features any device can run CNN models , which makes them accepted everwhere. CNN models are powerful and efficient for performing automatic feature extraction which can attain human perfection.

3.2Architecture

Almost all CNN models have a similar architecture like shown in the below figure :

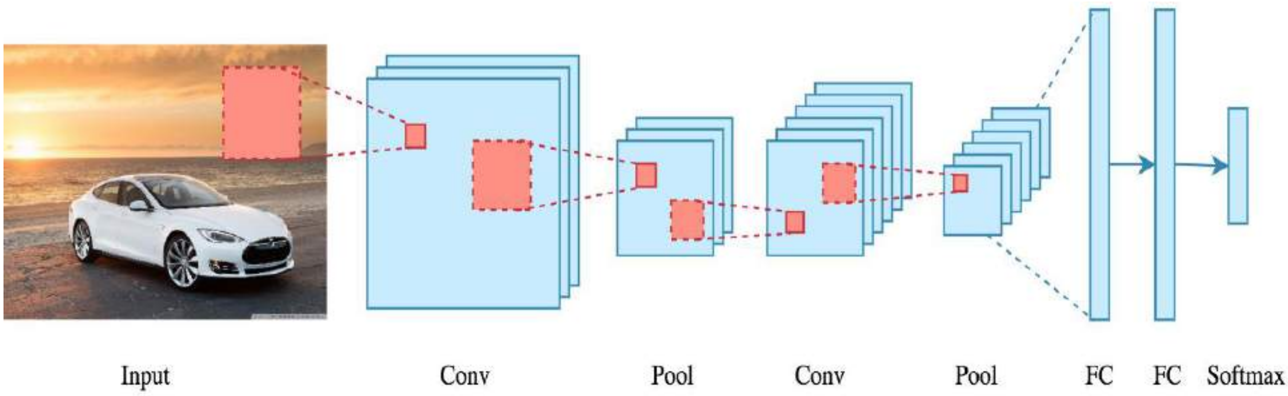


Figure 3.1: Architecture of CNN model

Firstly an input image is there to work with. A series convolution plus pooling operations is done. This is accompanied by many no. of fully connected layers. The output is softmax if we are performing multiclass classification .There are 3 types of layers a convolutional network is comprised of convolutional layer , pooling layer and fully connected layer.

3.2.1 Convolutional Layer

This layer is considered the fundamental block of a CNN. To combine 2 sets of information covolution is the main mathematical operation. Generally convolution is applied on input data with use of convolution filter leading to generation of feature-map.

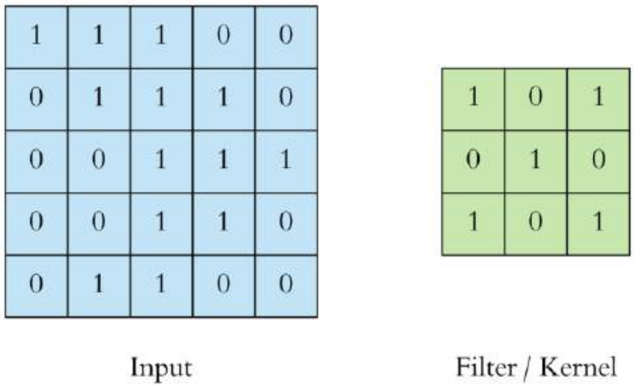


Figure 3.2: Convolution layer and filter

One is the convolution layer, like the input image. Second is the convolution filter/kernel. Due to the shape of the filter it is called a 3x3 convolution. The convolution process is performed by sliding this filter over the input. Then element-wise matrix multiplication is done at every point and result is summed. The feature map is the place where this sum goes. The receptive field is the place where the operation of convolution is done.

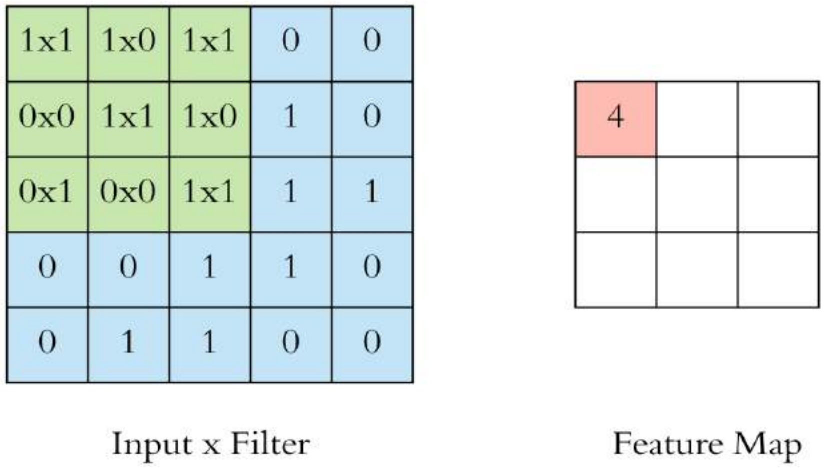


Figure 3.3: Mapping

In the figure the filter is at the top in left corner. The pink box below is the feature map which shows the output of convolution operation “4”. Same operation is done by sliding the filter to the right. We carry on in a similar way and in feature map convolution results are summed up .

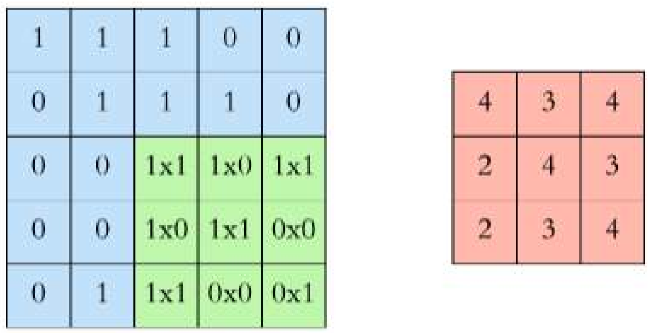


Figure 3.4: Feature map

The convolution operation shown here was in 2D which used a 3*3 filter. In actuality image is always represented as a 3D matrix having dimensions: height, width and depth. This filter has fixed height & width (3x3 or 5x5), & the entire depth of its input is covered by its design ,as a result it has to be 3D .

Before visualising the real operation of convolution on an input, multiple convolutions are done on it, with help of various filters which results in a different feature map. All of these feature maps are stacked with each other and this gives the final output of convolution layer .

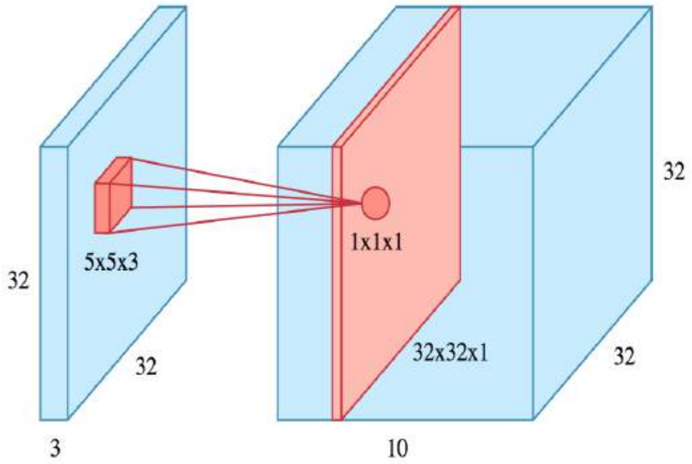


Figure 3.5: Convolution using a single-filter

If there is any 32*32*3 image and a filter of 5*5*3 is used then small volume of input will be covered by the filter and operation of convolution will be same as above. There is one difference during this time and that is in this sum of matrix multiply is done in 3-D and not in 2-D, still giving same result;that is a scalar (same) . The red part towards the right side shows the feature map size which is 32x32x1.

There would have been 10 feature maps of this size if we would have used 10 different filters . By stacking all these with dimension of depth this would have provided us with final output of convolution layer. Toward right side the blue box shows the volume(32x32x10). The sliding operation is done over the entire input in reality. [4]

Non-linearity

The neural network of any kind must be containing non-linearity to be powerful. On passing of the weighted sum of input's through the activation function, the ANN has achieved this. For CNN it is not so different The result which are observed from convolution operation is passed again through activation function(ReLU). Therefore in final feature maps the values present are not basically the aggregates , but these are activation function(ReLU)that been applied .Yet, any sort of convolution include a ReLU operation, without which the system will not achieve its actual potential.

Hyper-Parameters

If there is a convolution layer (ignoring pooling) .There are 2 important parameters :

- Filter size: we basically use 3x3 filters, but 5x5 or 7x7 are also used depending on the application. These filters are 3D and have a depth dimension as well, but at a given layer the depth of a filter is equal to the depth of its input, so we can omit that.

- Filter count: this is the most variable parameter, it’s a power of 2 anywhere b/w 32 and 1024. Using more filters results in a more powerful model, but there is a risk of overfitting due to increased parameter count. At the initial layers we start with a small no. of filters, and progressively increase the count as we go deeper into the network.

3.2.2 PoolingLayer

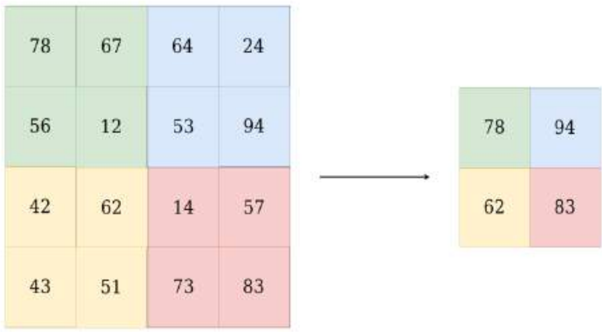


Figure 3.6: Pooling

Pooling is generally of two times namely maximum pooling and average pooling.The reduction in the no. of parameters of the input tensor is main purpose of this layer which further aids in reducing computation, overfittings and all these contribute for the efficiency. Pooling layer’s input is a tensor(array).

A kernel with size n*n(2x2 here)is slided over matrix and the maximum value at all location is taken for the case of maximum pooling like in fig4.6.These values are then put in output-matrix.

A kernel n*n size is slided over the matrix and average is taken of every values at all location which is then put in output-matrix.This is the case of average pooling.

3.2.3 Fully Connected Layer

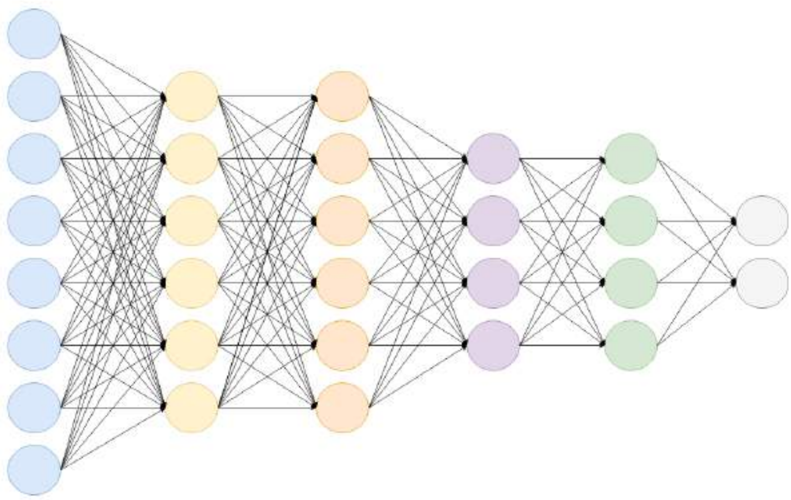


Figure 3.7: Network which is fully connected

This network is a type of feed-forward neural networks. The last few layers in a network form fully connected layers.

To feed output from final pooling/convoluyion layer to this fully connected one it is flattened first.This output from these final or pooling/convolutional layer is a 3-D matrix.For the purpose of flattening it all these values are unrolled to vector form.

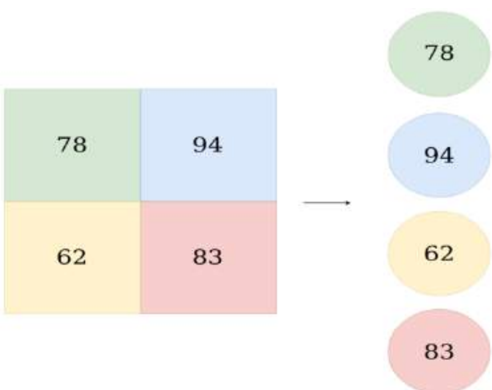


Figure 3.8: Flattenned vector

This obtained vector is further connected to some FC-layer that are similar to ANN. Then the given formula is used for the purpose of calculation for each and every layer :

$$\mathbf{G(Wx+b)}$$

here,“x” : input vector.The dimensions of it is [p__l, 1].

“W” : weighted matrix . The dimension of it is [p__l, n__l].

“p__l” : previous layer neuron’s.

“n__l” : current layer neuron’s.

“b” : bias vector. The dimensions of it is [p__l, 1]

“g” : activation function. (ReLU).

The same computation is done over and over again for each layer. As ReLU is used to get probabilities of input we generally use softmax activation function here instead of it. Hence probabilities of the object are achieved which belong to various types of classes.This is the working process of CNN which tells how image are classified as label.

For the purpose of calculation of the o/p tensor dimensions from i/p tensor is.:

$$W_2 = \frac{W_1 - F + 2P}{S} + 1$$

here,“W1” : width or height of the i/p tensor.

“F” : width or height of kernel.

“P” : padding.

“S” : stride.

“W2” : output width or height.

Padding: For a kernel to have uniform slide/movement over a matrix zero is padded around input matrix inorder to have desired dimension of output matrix..

Stride: Movement of kernel over a matrix is 1-pixel at a time. This means it have a stride=1.It can also be increased affecting dimensions of output and reducinf chances of overfitting. The no.of channels of output tensor = no. of kernels. For pooling layer,the channel’s in input tensor and output tensor remains same.[4]

CHAPTER 4
PROPOSED WORK

This chapter has in detail the work done while implementing this project and all the processes which surround the algorithm, the platform on which it was implemented, programming language used libraries and frameworks used.

4.1Software and Hardware Specifications

4.1.1Software Specifications

Python-3.7.2, is used to write all algorithms as *.py files. Python was chosen because it is a very powerful interpreter. Simultaneously the execution of partial code and debugging can be done in python. Also Not just that, the negligible syntax structure guarantees that the persuader doesn't get stalled by the syntactic subtleties of a specific programming language. Python likewise guarantees fast prototyping and quick execution within compiling softwares without the requirement for compiling program over again with very less changes. The external open source libraries utilized are:

1. **NumPy** : It is an array processing package for general purpose which provides high performance objects and tools for working with multidimensional array .

2. **Tensorflow** : An open source software library for ML.[5] It tends to be utilized across a variety of tasks but has a specific focus on training and inference of deep neural networks. It is a symbolic math library which is based on dataflow and differentiable programming.

3. **Keras** : It is a neural network library that is open source and can run on top of tensorflow ,R, microsoft cognitive toolkit and more frameworks. It is user-friendly and extensible.It is specially designed for enabling faster experimentation with deep neural networks.

Environment used:

1. **Google Colaboratory, or 'Colab'**, is a Google Research product.Writing and execution of arbitrary python code can be done by anyone through browser. It is especially used for machine learning, data analysis and educational purposes.

4.1.2 Hardware Specifications

Platform independent algorithms have been implemented. Any among Windows/Ubuntu/Mac platforms can be used for running them. I implemented it on Windows 10 with RAM 4GB of Acer laptop with processor Intel® Core™i5-8250U.

4.2 Network Architecture

VGG16 is a CNN model.It was given by K. Simonyan & A.Zisserman in their paper "Very Deep Convolutional Networks for Large-Scale Image Recognition". This model achieve 92.7% top-5 test accuracy in ImageNet,that is a huge dataset with over 14 million images that belong to 1000 classes.[6]

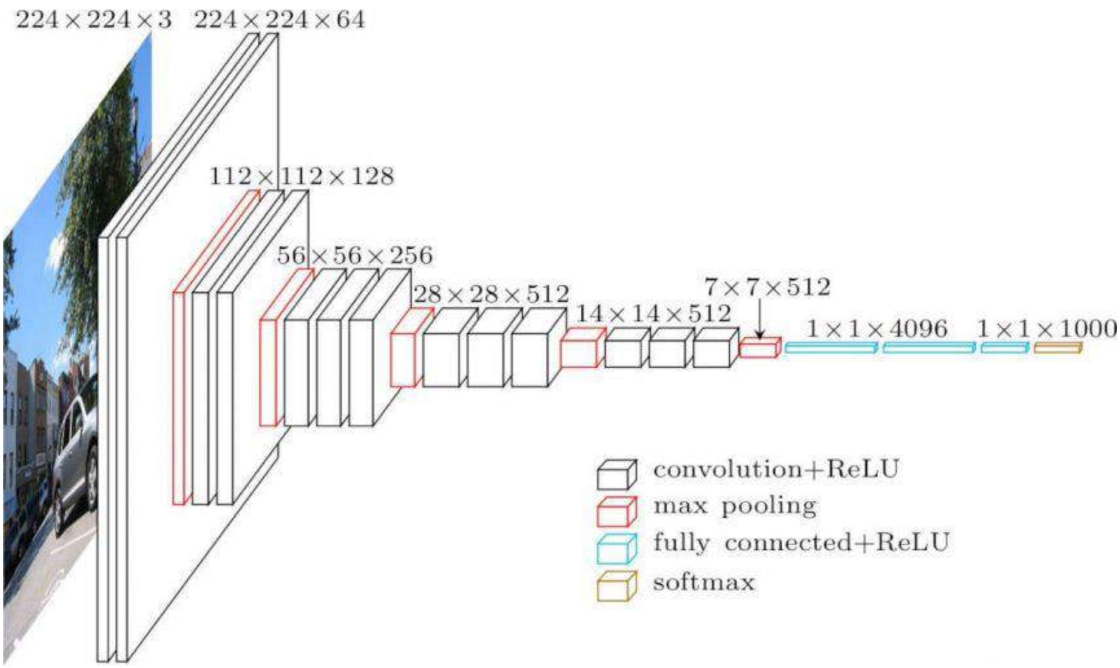


Fig 4.1: Architecture of VGG-16

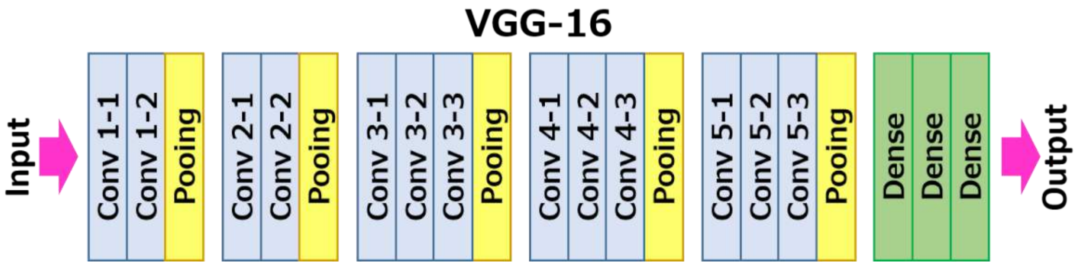


Fig 4.2: Layers of VGG-16

Input given to the convolution layer is of fixed size i.e. 224 x 224 in RGB format. This input is passed through convolution layers where the filter of size 3 x 3 is used, with the stride of 1 and same paddding.- to preserve the spatial resolution. It also makes use of 1 x 1 filter, which is the linear transformation of the input signal. %5max pooling layers are used after convolution layers. Max pooling is performed with 2 x 2 window with a stride of 2.

These are connected to fully connected layers (depth of which varies in different configurations.) . First & second fully connected layers have 4096 channels each. The third performs the ILSVRC classification and has 1000 channels.[6] Finally, softmax layer is connected. Activation function used in all hidden layers is ReLU. It is also noted that not all the networks are normalized as it doesn't have much effect on the accuracy but instead consumes more memory and computation time s also more.

4.3 Transfer Learning

A model that has been trained for one task is used as the starting point for solving another task in transfer learning. As a result, rather than going through the lengthy process of training with randomly initialised weights, pre-trained models are used as the starting point for certain particular tasks in transfer learning. As a result, it aids in the reduction of the significant computational resources needed to create neural network models to solve these problems.

The concept of transfer learning is based on the fact that in machine learning, we can use information learned from one problem A and apply it to another problem B that is connected to it. Furthermore, by using existing networks that have already been trained on large datasets, we can build accurate deep models on small datasets using transfer learning.Rather than training our network from scratch, we can use these pre-trained models, which have fairly standardised low-level features and can thus be used in any image classification task.

4.4 Dataset Preparation

For the final dataset 2 datasets were combined together to address the problem at hand . The first dataset used was the Kaggle Chest X-ray dataset which have 5863 X-ray images belonging to two categories normal & pneumonia(bacterial , viral).[7]

The other dataset is the COVID-19 Radiography Database also from Kaggle which includes 3616 chest xray images of COVID-19 positive cases, 10192 of normal, 6012 of lung opacity, 1345 of viral pneumonia.These datasets are made up of posterior-anterior chest photographs of patients.[8]

The final sampled dataset has 2200 total x-ray images with 3 categories :-

Category	***Training set***	***Test set***
COVID-19	700	34
Bacterial Pneumonia	700	32
Normal	700	34

4.5 Preprocessing of Data and Data augmentation

Every picture is to be preprocessed using the deep neural network that was being used. Resizing and normalisation were the 2 most critical stages. According to their architecture, different neural networks need images of various sizes. Images of size 224 x 224 are expected by ResNet18, DenseNet121, and Vgg16, while image size of 229 x 229 are required by InceptionV3 and Xception.[9] All of the images were also normalised according to the architectures they were produced in.[10]

Big data is needed for adequate neural network training. When data is scarce, parameters are weakened, and trained networks fail to generalise well. Data augmentation addresses this issue by making better use of existing data. It helps the model avoid overfitting the current training dataset by increasing the size of the dataset. [Code for data preprocessing is given in Appendix]

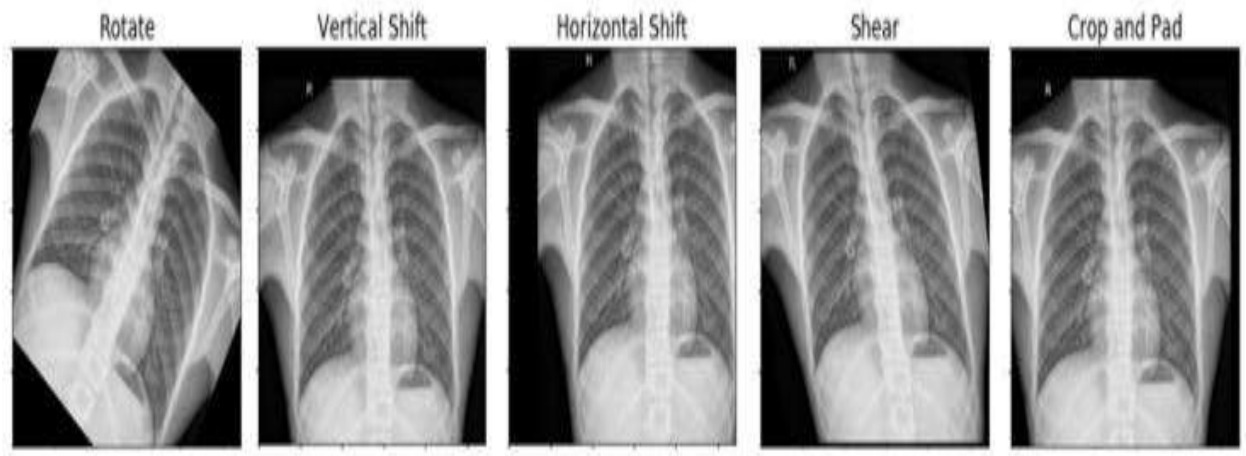


Fig 4.3 : Data augmentation technique

The raw chest X-ray images were been pre-processed and normalised. The dataset was then processed more effectively using data augmentation techniques. The networks' layers were all trainable, and these layers extracted the features from the images.

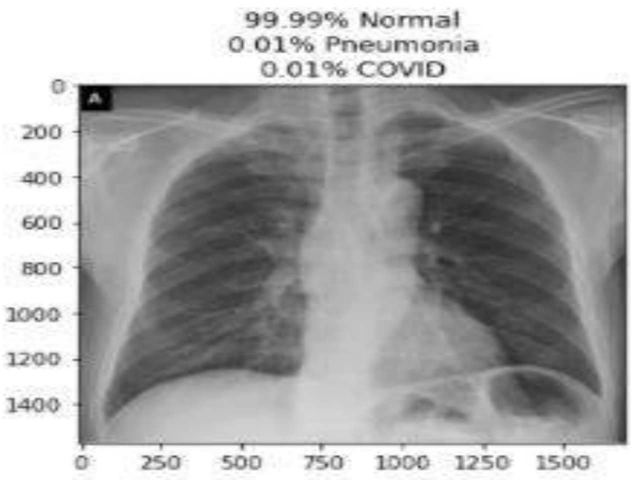
4.6 Training

A pre-trained VGG16 network is taken, and fitting it to a series of densely-connected layers of our own. The ImageDataGenerator class mode parameter is set to "categorical." It is 16 weight layers. The model was trained for multi classification of the Dataset. D configuration of the VGG model was used. Wherein we got 16 convolution layers, 5 max pooling layers an 3 fully connected layers which were finally connected to a softmax layer. The filters used are of size 3 x 3 and the padding used is same.

With a softmax activation function, the no. of neurons in final densely connected layer now corresponds to the no. of classes that were considered. This provide a probability as output for each classes, with the highest value serving as our final predicted result. A categorical crossentropy loss function is now used to compile the model. Inorder to accommodate increased situational complexity the no. of training epochs is increased. Similarly, the learning rate has been decreased to 1E-4.

4.7 Results Observed

Able to achieve an accuracy of 96% on the VGG-16 model .



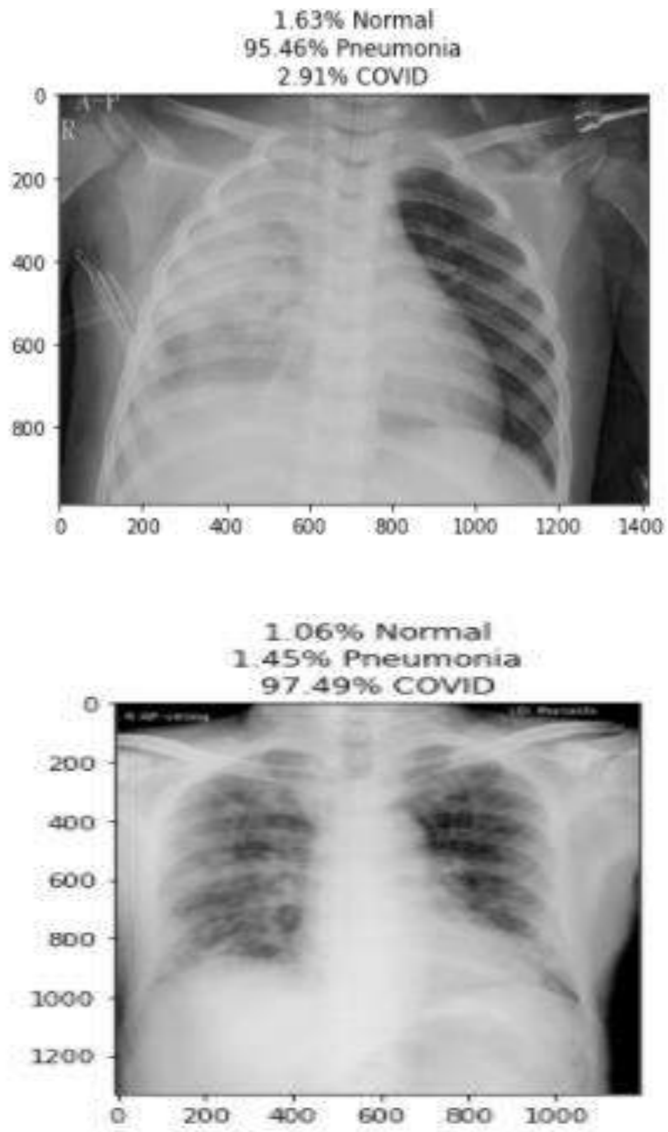


Figure 4.3: Output (Normal,Pneumonia,Covid19)

```
Epoch 80/100
16/16 [==============================] - 5s 315ms/step - loss: 0.2212 - acc: 0.9062 - val_loss: 1.1413 - val_acc: 0.7500
Epoch 81/100
16/16 [==============================] - 5s 315ms/step - loss: 0.2363 - acc: 0.9062 - val_loss: 0.3800 - val_acc: 0.8750
Epoch 82/100
16/16 [==============================] - 5s 320ms/step - loss: 0.2215 - acc: 0.9250 - val_loss: 0.6261 - val_acc: 0.8000
Epoch 83/100
16/16 [==============================] - 5s 324ms/step - loss: 0.2763 - acc: 0.8750 - val_loss: 0.1557 - val_acc: 0.9250
Epoch 84/100
16/16 [==============================] - 5s 315ms/step - loss: 0.1580 - acc: 0.9563 - val_loss: 0.6188 - val_acc: 0.8000
Epoch 85/100
16/16 [==============================] - 5s 317ms/step - loss: 0.2309 - acc: 0.9125 - val_loss: 0.5336 - val_acc: 0.8750
Epoch 86/100
16/16 [==============================] - 5s 316ms/step - loss: 0.1403 - acc: 0.9438 - val_loss: 0.3621 - val_acc: 0.8000
Epoch 87/100
16/16 [==============================] - 5s 318ms/step - loss: 0.2239 - acc: 0.9125 - val_loss: 0.3026 - val_acc: 0.8750
Epoch 88/100
16/16 [==============================] - 5s 319ms/step - loss: 0.2160 - acc: 0.9250 - val_loss: 0.1890 - val_acc: 0.9000
Epoch 89/100
16/16 [==============================] - 5s 319ms/step - loss: 0.2358 - acc: 0.9438 - val_loss: 0.5251 - val_acc: 0.8250
Epoch 90/100
16/16 [==============================] - 5s 318ms/step - loss: 0.1957 - acc: 0.9250 - val_loss: 0.4147 - val_acc: 0.8750
Epoch 91/100
16/16 [==============================] - 5s 322ms/step - loss: 0.2734 - acc: 0.9000 - val_loss: 0.1767 - val_acc: 0.7750
Epoch 92/100
16/16 [==============================] - 5s 330ms/step - loss: 0.2408 - acc: 0.9125 - val_loss: 0.2258 - val_acc: 0.8500
Epoch 93/100
16/16 [==============================] - 5s 317ms/step - loss: 0.2307 - acc: 0.9125 - val_loss: 0.2765 - val_acc: 0.8750
Epoch 94/100
16/16 [==============================] - 5s 320ms/step - loss: 0.2129 - acc: 0.9125 - val_loss: 0.5781 - val_acc: 0.6750
Epoch 95/100
16/16 [==============================] - 5s 325ms/step - loss: 0.1846 - acc: 0.9438 - val_loss: 0.1448 - val_acc: 0.8750
Epoch 96/100
16/16 [==============================] - 5s 320ms/step - loss: 0.1614 - acc: 0.9563 - val_loss: 0.5190 - val_acc: 0.9000
Epoch 97/100
16/16 [==============================] - 5s 320ms/step - loss: 0.1709 - acc: 0.9500 - val_loss: 0.2607 - val_acc: 0.7750
Epoch 98/100
16/16 [==============================] - 5s 320ms/step - loss: 0.1976 - acc: 0.9438 - val_loss: 0.2180 - val_acc: 0.9500
Epoch 99/100
16/16 [==============================] - 5s 316ms/step - loss: 0.1862 - acc: 0.9062 - val_loss: 0.1026 - val_acc: 0.8000
```

Figure 4.4: Accuracy~96%

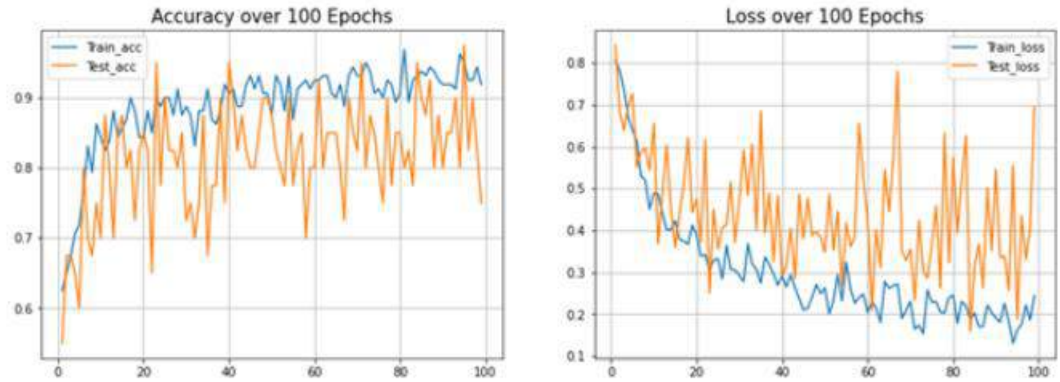


Figure 4.5: Accuracy and loss parameters

CONCLUSION

This project can assist in the analysis and creation of a models based on these important datasets. As the world is seeing new and new life threatening diseases everyday such projects and research can be an aid in future to use such technologies more often that can help the mankind.Covid-19 outbreak has taught us that how strong our healthcare system should be and how we should not take any symptoms lightly . Correct and early diagnosis are the need of the hour .Valuable datasets with age, patient data, gender, snapshot data, and Xray images are more and more required, which is very useful for such models. Doctors can diagnose a patient's wellbeing &their medical problems byreviewing X-ray records. From the output data of X-ray chest images, the intelligent machines can be of great help to physicians for diagnosis or analysis of chest diseases.

REFERENCES

[1] Envarad:“*Difference-between-x-ray-ct-scan-and-mri*”

[2] Hunton R.,“*Updated concepts in the diagnosis and management of community-acquired pneumonia.*”, 2019

[3] Dinesh Kumawat ,“*7 Types of Activation Functions in Neural Network*”,2019

[4] Srivignesh Rajan,“*An Introduction to Artificial Neural Networks*”,2019

[5] Matthew ZakAdam Krzyżak, “*Classification of Lung Diseases Using Deep Learning Models*”,2020

[6] Neurohive:“*VGG16 – Convolutional Network for Classification and Detection*”,2018

[7] Paul,Mooney, “*Chest X-ray dataset*”,Kaggle

[8] Tawsifur Rahman, “*COVID-19 Radiography Database*”,Kaggle

[9]Rahib H. Abiyev and Mohammad Khaleel Sallam Ma’aitah,"*Deep Convolutional Neural Networks for Chest Diseases Detection.*",2018

[10]Aydin Kayaa,, Ali Seydi Kecelia, Cagatay Catalb, Hamdi Yalin Yalica, Huseyin Temucina,Bedir Tekinerdogan, “*Analysis of transfer learning for deep neural network based plant classification models*” ,2019

APPENDIX

```
#Train datagen here is a preprocessor
train_datagen = ImageDataGenerator(rescale=1./255,
                                   rotation_range=50,
                                   featurewise_center = True,
                                   featurewise_std_normalization = True,
                                   width_shift_range=0.2,
                                   height_shift_range=0.2,
                                   shear_range=0.25,
                                   zoom_range=0.1,
                                   zca_whitening = True,
                                   channel_shift_range = 20,
                                   horizontal_flip = True ,
                                   vertical_flip = True ,
                                   validation_split = 0.2,
                                   fill_mode='constant')



train_batches = train_datagen.flow_from_directory(DATASET_PATH,
                                                  target_size=IMAGE_SIZE,
                                                  shuffle=True,
                                                  batch_size=BATCH_SIZE,
                                                  subset = "training",
                                                  seed=42,
                                                  class_mode="categorical"
                                                  )

valid_batches = train_datagen.flow_from_directory(DATASET_PATH,
                                                  target_size=IMAGE_SIZE,
                                                  shuffle=True,
                                                  batch_size=BATCH_SIZE,
                                                  subset = "validation",
                                                  seed=42,
                                                  class_mode="categorical"

                                                  )
```

```
from keras import models
from keras import layers
from keras.applications import VGG16
from keras import optimizers

conv_base = VGG16(weights='imagenet',
                  include_top=False,
                  input_shape=(224, 224, 3))


conv_base.trainable = False


model = models.Sequential()
model.add(conv_base)
model.add(layers.Flatten())
model.add(layers.Dense(256, activation='relu'))
model.add(layers.Dense(3, activation='softmax'))


model.compile(loss='categorical_crossentropy',

              optimizer=optimizers.Adam(lr=LEARNING_RATE),
              metrics=['acc'])
```

```
#FIT MODEL
print(len(train_batches))
print(len(valid_batches))

STEP_SIZE_TRAIN=train_batches.n//train_batches.batch_size
STEP_SIZE_VALID=valid_batches.n//valid_batches.batch_size

result=model.fit_generator(train_batches,
                          steps_per_epoch =STEP_SIZE_TRAIN,
                          validation_data = valid_batches,
                          validation_steps = STEP_SIZE_VALID,
                          epochs= NUM_EPOCHS,
                         )
```

```
import matplotlib.pyplot as plt


def plot_acc_loss(result, epochs):
    acc = result.history['acc']
    loss = result.history['loss']
    val_acc = result.history['val_acc']
    val_loss = result.history['val_loss']
    plt.figure(figsize=(15, 5))
    plt.subplot(121)
    plt.plot(range(1,epochs), acc[1:], label='Train_acc')
    plt.plot(range(1,epochs), val_acc[1:], label='Test_acc')
    plt.title('Accuracy over ' + str(epochs) + ' Epochs', size=15)
    plt.legend()
    plt.grid(True)
    plt.subplot(122)
    plt.plot(range(1,epochs), loss[1:], label='Train_loss')
    plt.plot(range(1,epochs), val_loss[1:], label='Test_loss')
    plt.title('Loss over ' + str(epochs) + ' Epochs', size=15)
    plt.legend()
    plt.grid(True)
    plt.show()

plot_acc_loss(result, 100)
```

```
import matplotlib.image as mpimg

pred = model.predict_generator(eval_generator,1000,verbose=1)
#print (list(enumerate(pred)))
try:
  for index, probability in enumerate(pred):
     image_path = test_dir + "/" +eval_generator.filenames[index]
     image = mpimg.imread(image_path)
    #BGR TO RGB conversion using CV2
     image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

     pixels = np.array(image)
     plt.imshow(pixels)

   #  print(eval_generator.filenames[index])
    # if probability > 0.5:
   #  plt.title("%.2f" % (probability[0]*100) + "% Normal")
    # else:
    #      plt.title("%.2f" % ((1-probability[0])*100) + "% COVID19 Pneumonia")

     s="%.2f" % (probability[0]*100) + "% Normal"
     y="%.2f" % (probability[1]*100) + "% Pneumonia"
     z="%.2f" % (probability[2]*100) + "% COVID"
     n="\n"
     t=s+n+y+n+z
     plt.title(t)
     plt.show()
except:
  print("done")
```

PLAGIARISM REPORT

Prerna_171006_btech_Report_19052021_2

ORIGINALITY REPORT

10%	7%	2%	5%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	towardsdatascience.com Internet Source	3%
2	Submitted to Jaypee University of Information Technology Student Paper	1%
3	Submitted to Sogang University Student Paper	1%
4	Submitted to Southern New Hampshire University - Continuing Education Student Paper	1%
5	Submitted to Nanyang Technological University Student Paper	1%
6	en.wikipedia.org Internet Source	<1%
7	Cuauhtemoc Mariscal-Garcia, Wendy Flores-Fuentes, Daniel Hernandez-Balbuena, Julio C. Rodriguez-Quinonez et al. "Classification of Vehicle Images through Deep Neural Networks for Camera View Position	<1%