**45-DAYS INDUSTRIAL TRAINING REPORT**

AR-BOT

VIRTUAL PERONAL ASSISTANT

Submitted in partial fulfilment of requirement for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE ENGINEERING



**SUBMIITED BY :-**

**RITIK THAKUR (319021007)**

**ANUJ ATTRI (39021005)**

**Department Of Computer Science & Engineering Sri Sai University , Palampur-176061**

**ACKNOWLEDGEMENT:**

It is my pleasure to be indebted to various people, who directly or indirectly contributed in the development of this work and who influenced my thinking, behavior and acts during the course of study.

I am thankful to my project guide ***Ms. Amandeep Kaur*** for her support, cooperation, and motivation provided to me during the training for constant inspiration, presence and blessings.

Lastly, I would like to thank the almighty and my parents for their moral support and my friends with whom I shared my day-to -day experience and received lots of suggestions that my quality of work.

**DECLARATION**

We are Anuj Attri (319021005) and Ritik Thakur(319021007) , B.Tech (Semester-5th) of ***Sri Sai University, Palampur***, Himachal Pradesh ,hereby declare that the Training Report entitled **“AR-BoT -Virtual Personal Assistan**t ” is an original work and data provided in the study is authentic to the best of my knowledge. This report has not been submitted to any other Institute for the award of any other degree.

**TABLE OF CONTENTS**

**Sr. No. Name of chapter**

**Chapter 1 Company Profile**

* 1. Foundation

1.2 Service

**Chapter 2 Introduction of Project**

2.1 Introduction

**Chapter 3 Methodology**

3.1 Software Development -

Life Cycle

**Chapter 4 Software & Technology Used**

4.1 About Python

4.2 About VS Code

4.3 About AI

4.3(a) Introduction To Artificial

Intelligence(AI)

4.3(b) History of AI

4.3(c) Application of AI

4.4 About Machine Learning

(ML)

4.4(a) Introduction to ML

4.4(b) History of ML

4.5 Supervised Learning

4.6 Unsupervised Learning

4.7 Reinforcement Cycle

4.8 ML Life Cycle.

4.9 Application of ML

4.10 Introduction to TensorFlow

4.11 Introduction to Keras

**Chapter 5 Project Analysis**

5.1 Software Project

**Chapter 6 Summing UP**

6.1 Conclusion

6.2 Future Scope

6.3 Bibliograohy



**CHAPTER-1**

**COMPANY PROFILE**

**FOUNDATION:**

STS has been founded by group of senior IT Professional. Right from the inception of this start up, STS has prospered by Leaps and bounds in technology products and critical solutions. SachTech Solution established back in 29 December, 2011 at Mohali, India to serve the varying need of individuals as well as SMEs in today’s competitive market across the globe it was incorporated as SACHTECH SOLUTION PRIVATE LIMITED with CIN U72900CH2016PTC041177 on 11th Aug 2016 under the Companies Act, 2013 in India.

As of year 2018, SachTech has a strong team of more than 80 members in Mohali, India lead by passionate young entrepreneurs serving customers from across the globe in following countries: USA, Canada, UK, Brazil, Spain, Malaysia, UAE, Egypt, Australia, Finland and so on. We are continuously increasing our reach with potential customers and determined to expand our services to everyone in the globe. We embrace our responsibility to create a positive impact in the communities in which we work and live. We use proven knowledge to make recommendations and provide expert guidance to our customers. We listen, we care, and we serve. SachTech Solution innovates and constantly improves. We do what we say we’ll do. SachTech Solution believes in people and their dreams.

Within the span of six years, STS is the strong team of more than 80 members having its presence in India (Mohali) in 7000 sq ft. & in Canada. Our operations are spread in New York, UK, Australia, Saudi Arabia, UAE and Israel.

**Mission & Vision:**Our Mission is to be the world's leading IT Channel in products, services and solutions that empower and alchemize the way consumers and businesses assemble, manage, distribute and communicate information. Our vision is to become a world-class software development and technology provider and to provide clients with innovated technical and business solutions by utilizing industry standards and technology.

**Achievements:**STS believes in Quality and it is evident from various technology breaks through like from fastest development systems to Desktop Retail Applications integrated with highly innovative data center services. STS works along with the client to improve its business outcomes by exploring new business opportunities, deriving cost takeout, and increasing process efficiency without any major change. From innovative ideas to their implementation and thereafter, STS offers all business transformation outsourcing services to clients under one flagship in four different phases of consulting, developing, outsourcing and training.

**SERVICES WE OFFER**:

Consulting: STS has 360-degree approach including each business process through a panel of various domain experts, who work hard along with the client to identify the requirements to achieve client's goal while respecting its value. STS has devised ready to opt industry vertical consulting solutions for various processes like Business Case Analysis, Business process re-engineering and Management product, Development and Management, IT Strategy Formulation, Technology Support Development, Internal Marketing, Product Testing, Performance Management etc.

**Development:** STS has the honor of developing innovative technologies and the growth of cloud, mobile computing and social media have put additional burdens on staff looking to quickly provide modern solutions. We also offer bouquet of various enterprise solutions, Android applications, Desktop applications, Web & Device Applications.

**Outsourcing:** Besides various readymade STS business process outsourcing solutions for various processes like collocation services, Onsite Database Administration Services, Online Counter etc., we have specialization in various industrial critical, technical and general processes. Our man resources are trained for client processes and work as client’s integral part and are fully accessible by client directly.

**Industrial Training:** From corporate training to end user training and technical Trainings like System Administration, Enterprise Architecture, Enterprise Network etc.

STS has client based dedicated training programs to ensure client can take maximum advantage of our system, services and solutions. Apart from in-house trainers, we have ever-growing team of our training partners offering customized professional training modules to enterprising and up comings professionals.

**Chapter 2**

**Introduction of Project**

**“AR- BoT The Virtual Personal Assistant(VPA)”**  is an [artificial intelligence](https://en.m.wikipedia.org/wiki/Artificial_intelligence)–powered [virtual assistant](https://en.m.wikipedia.org/wiki/Virtual_assistant) developed by Anuj And Ritik  that is primarily available on [mobile](https://en.m.wikipedia.org/wiki/Mobile_device) and [smart home](https://en.m.wikipedia.org/wiki/Home_automation) devices.

**ALL ABOUT VIRTUAL ASSISTANT:**

Assistant initially debuted in May 2016 as part of Google's messaging app [Allo](https://en.m.wikipedia.org/wiki/Google_Allo" \o "Google Allo), and its voice-activated speaker [Google Home](https://en.m.wikipedia.org/wiki/Google_Home). After a period of exclusivity on the [Pixel and Pixel XL](https://en.m.wikipedia.org/wiki/Pixel_(smartphone)) smartphones, it began to be deployed on other [Android](https://en.m.wikipedia.org/wiki/Android_(operating_system)) devices in February 2017, including third-party smartphones and Android Wear (now [Wear OS](https://en.m.wikipedia.org/wiki/Wear_OS)), and was released as a standalone app on the [iOS](https://en.m.wikipedia.org/wiki/IOS) operating system in May 2017. Alongside the announcement of a [software development kit](https://en.m.wikipedia.org/wiki/Software_development_kit) in April 2017, the Assistant has been further extended to support a large variety of devices, including cars and third-party smart home appliances. The functionality of the Assistant can also be enhanced by third-party developers.

Users primarily interact with the Google Assistant through [natural voice](https://en.m.wikipedia.org/wiki/Natural_language_processing), though keyboard input is also supported. In the same nature and manner as Google Now, the Assistant is able to search the Internet, schedule events and alarms, adjust hardware settings on the user's device, and show information from the user's Google account. Google has also announced that the Assistant will be able to identify objects and gather visual information through the device's camera, and support purchasing products and sending money.

At [CES 2018](https://en.m.wikipedia.org/wiki/CES_2018), the first Assistant-powered smart displays ([smart speakers](https://en.m.wikipedia.org/wiki/Smart_speakers) with video screens) were announced, with the first one being released in July 2018 In 2020, Google Assistant is already available on more than 1 billion devices.Google Assistant is available in more than 90 countries and in over 30 languages, and is used by more than 500 million users monthly.

**Chapter 3**

**Methedology**

*3.1 Software Development Life Cycle (SDLC)*

System development life cycle is a process of developing software on the basis of the requirement of the end user to develop efficient and good quality software. There are various software development approaches defined and designed which are used during development process of software, these approaches are also referred as “Software Development Process Models” (e.g. Waterfall model, incremental model, V-model, iterative model, etc.). Each process model follows a particular life cycle in order to ensure success in process of software development.

System development life cycle model describes the phases of the software cycle and the order in which those phases are executed. Each phase produces deliverables required by the next phase in the life cycle. Requirements are translated into design. Code is produced according to the design which is called development phase. After coding and development the testing verifies the deliverable of the implementation phase against requirements.



A software development process is the process of dividing software devlopment work into distinct phases to improve design, product management, and project management. It is also known as a software development life cycle. The methodology may include the pre-definition of specific deliverables and artifacts that are created and completed by a project team to develop or maintain an application.

Most modern development processes can be vaguely described as agile. Other methodologies include waterfall, prototyping, iterative and incremental development, spiral development, rapid application development, and extreme programming

.

Some people consider a life-cycle "model" a more general term for a category of methodologies and a software development "process" a more specific term to refer to a specific process chosen by a specific organization. For example, there are many specific software development processes that fit the spiral life-cycle model. The field is often considered a subset of the systems development life cycle.

**Chapter 4**

*Software and Technology Used*

**ABOUT PYTHON:**

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python’s simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

Python was conceived in the late 1980s, and its implementation began in December 1989 by Guido van Rossum at Centrum Wiskunde & Informatica (CWI) in the Netherlands as a successor to the ABC language (itself inspired by SETL) capable of exception handling and interfacing with the Amoeba operating system.

What can PYTHON do?

Python is a general-purpose programming language. Hence, you can use the programming language for developing both desktop and web applications. Also, you can use Python for developing complex scientific and numeric applications. Python is designed with features to facilitate data analysis and visualization.

In, February 1991, van Rossum published the code (labeled version 0.9.0) to alt.sources.Already present at this stage in development were classes with inheritance, exception handling, functions, and the core data-types of list, dict, str and so on. Also in this initial release was a module system borrowed from Modula-3; Van Rossum describes the module as “one of Python’s major programming units”.Python’s exception model also resembles Modula-3’s, with the addition of an else clause.In 1994 comp.lang.python, the primary discussion forum for Python, was formed, marking a milestone in the growth of Python’s user base.

Python reached version 1.0 in January 1994. The major new features included in this release were the functional programming tools lambda, map, filter and reduce. Van Rossum stated that “Python acquired lambda, reduce(), filter() and map(), courtesy of a Lisp hacker who missed them and submitted working patches”.

**ABOUT VS CODE**

**Visual Studio Code** is an [integrated development environment](https://en.m.wikipedia.org/wiki/Integrated_development_environment) made by [Microsoft](https://en.m.wikipedia.org/wiki/Microsoft) for [Windows](https://en.m.wikipedia.org/wiki/Windows), [Linux](https://en.m.wikipedia.org/wiki/Linux) and [macOS](https://en.m.wikipedia.org/wiki/MacOS) Features include support for [debugging](https://en.m.wikipedia.org/wiki/Debugging), [syntax highlighting](https://en.m.wikipedia.org/wiki/Syntax_highlighting), [intelligent code completion](https://en.m.wikipedia.org/wiki/Intelligent_code_completion), [snippets](https://en.m.wikipedia.org/wiki/Snippet_(programming)), [code refactoring](https://en.m.wikipedia.org/wiki/Code_refactoring), and embedded [Git](https://en.m.wikipedia.org/wiki/Git). Users can change the [theme](https://en.m.wikipedia.org/wiki/Theme_(computing)), [keyboard shortcuts](https://en.m.wikipedia.org/wiki/Keyboard_shortcut), preferences, and install [extensions](https://en.m.wikipedia.org/wiki/Plug-in_(computing)) that add additional functionality.

Microsoft has released most of Visual Studio Code's [source code](https://en.m.wikipedia.org/wiki/Source_code) on the microsoft/vscode repository of [GitHub](https://en.m.wikipedia.org/wiki/GitHub) using the "Code – OSS" name, under the permissive [MIT License](https://en.m.wikipedia.org/wiki/MIT_License) while the releases by Microsoft are proprietary [freeware](https://en.m.wikipedia.org/wiki/Freeware)

In the [Stack Overflow](https://en.m.wikipedia.org/wiki/Stack_Overflow) 2021 Developer Survey, Visual Studio Code was ranked the most popular developer environment tool, with 71.06% of 82,277 respondents reporting that they use it

Visual Studio Code was first announced on April 29, 2015, by Microsoft at the 2015 [Build](https://en.m.wikipedia.org/wiki/Build_(developer_conference)) conference. A Preview build was released shortly thereafter.

On November 18, 2015, Visual Studio Code was released under the [MIT License](https://en.m.wikipedia.org/wiki/MIT_License), having its source code available on [GitHub](https://en.m.wikipedia.org/wiki/GitHub). Extension support was also announced. On April 14, 2016, Visual Studio Code graduated from the [public preview](https://en.m.wikipedia.org/wiki/Beta_software) stage and was [released to the Web](https://en.m.wikipedia.org/wiki/Software_release_life_cycle#Web_release).

*4.5 INTRODUCTION TO AI:*

Artificial intelligence (AI) is defined as intelligence exhibited by an artificial entity. Such a system is generally assumed to be a computer.

Although AI has a strong science fiction connotation, it forms a vital branch of computer science, dealing with intelligent behavior, learning and adaptation in machines. Research in AI is concerned with producing machines to automate tasks requiring intelligent behavior. Examples include control, planning and scheduling, the ability to answer diagnostic and consumer questions, handwriting, speech, and facial recognition.



*4.6 HISTORY OF AI:*

The intellectual roots of AI, and the concept of intelligent machines, may be found in Greek mythology. Intelligent artifacts appear in literature since then, with real mechanical devices actually demonstrating behaviour with some degree of intelligence. After modern computers became available following World War-II, it has become possible to create programs that perform difficult intellectual tasks.

**1950 – 1960:**

The first working AI programs were written in 1951 to run on the Ferranti Mark I machine of the University of Manchester (UK): a draughts-playing program written by Christopher Strachey and a chess-playing program written by Dietrich Prinz.

**1960 – 1970 :**

During the 1960s and 1970s Marvin Minsky and Seymour Papert publish Perceptrons, demonstrating limits of simple neural nets and Alain Colmerauer developed the Prolog computer language. Ted Shortliffe demonstrated the power of rule-based systems for knowledge representation and inference in medical diagnosis and therapy in what is sometimes called the first expert system. Hans Moravec developed the first computer-controlled vehicle to autonomously negotiate cluttered obstacle courses.

**1980’s ONWARDS :**

In the 1980s, neural networks became widely used with the back propagation algorithm, first described by Paul John Werbos in 1974. The 1990s marked major achievements in many areas of AI and demonstrations of various applications. Most notably Deep Blue, a chess-playing computer, beat Garry Kasparov in a famous six-game match in 1997.

*4.7 APPLICATIONS OF AI :*

**Game Playing :**

You can buy machines that can play master level chess for a few hundred dollars. There is some AI in them, but they play well against people mainly through brute force computation—looking at hundreds of thousands of positions.

**Speech Recognition :**

In the 1990s, computer speech recognition reached a practical level for limited purposes. Thus United Airlines has replaced its keyboard tree for flight information by a system using speech recognition of flight numbers and city names. It is quite convenient. On the other hand, while it is possible to instruct some computers using speech, most users have gone back to the keyboard and the mouse as still more convenient.

**Understanding Natural Language :**

Just getting a sequence of words into a computer is not enough. Parsing sentences is not enough either. The computer has to be provided with an understanding of the domain the text is about, and this is presently possible only for very limited domains.

**Computer Vision :**

The world is composed of three-dimensional objects, but the inputs to the human eye and computer’s TV cameras are two dimensional. Some useful programs can work solely in two dimensions, but full computer vision requires partial three-dimensional information that is not just a set of two-dimensional views. At present there are only limited ways of representing three-dimensional information directly, and they are not as good as what humans evidently use.

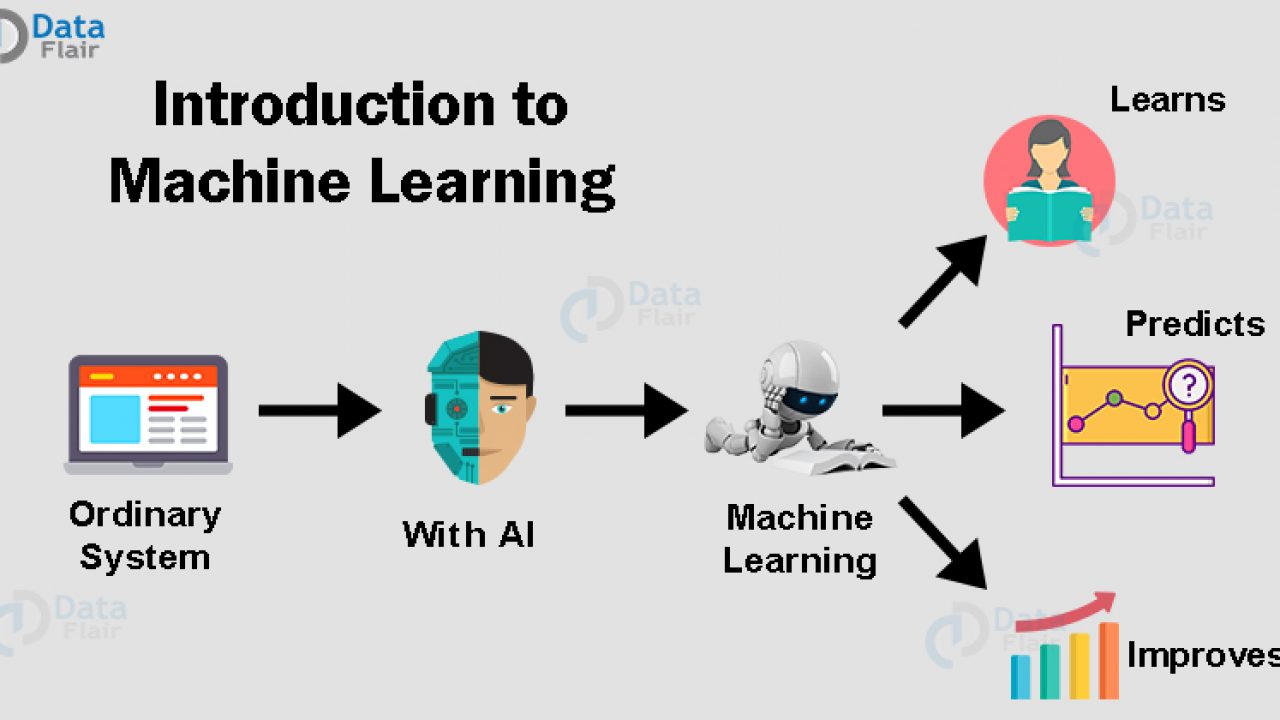
**Expert Systems :**

A ``knowledge engineer’’ interviews experts in a certain domain and tries to embody their knowledge in a computer program for carrying out some task. How well this works depends on whether the intellectual mechanisms required for the task are within the present state of AI. One of the first expert systems was MYCIN in 1974, which diagnosed bacterial infections of the blood and suggested treatments. It did better than medical students or practicing doctors, provided its limitations were observed.

**Heuristic Classification :**

One of the most feasible kinds of expert system given the present knowledge of AI is to put some information in one of a fixed set of categories using several sources of information. An example is advising whether to accept a proposed credit card purchase. Information is available about the owner of the credit card, his record of payment and also about the item he is buying and about the establishment from which he is buying it (e.g., about whether there have been previous credit card frauds at this establishment).

**4.8 Introduction to Machine learning** :

****

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it learn for themselves.

The process of learning begins with observations or data, such as examples, direct experience, or instruction, in order to look for patterns in data and make better decisions in the future based on the examples that we provide. The primary aim is to allow the computers learn automatically without human intervention or assistance and adjust actions accordingly.

But, using the classic algorithms of machine learning, text is considered as a sequence of keywords; instead, an approach based on semantic analysis mimics the human ability to understand the meaning of a text.

*4.8.1Some machine learning methods:*

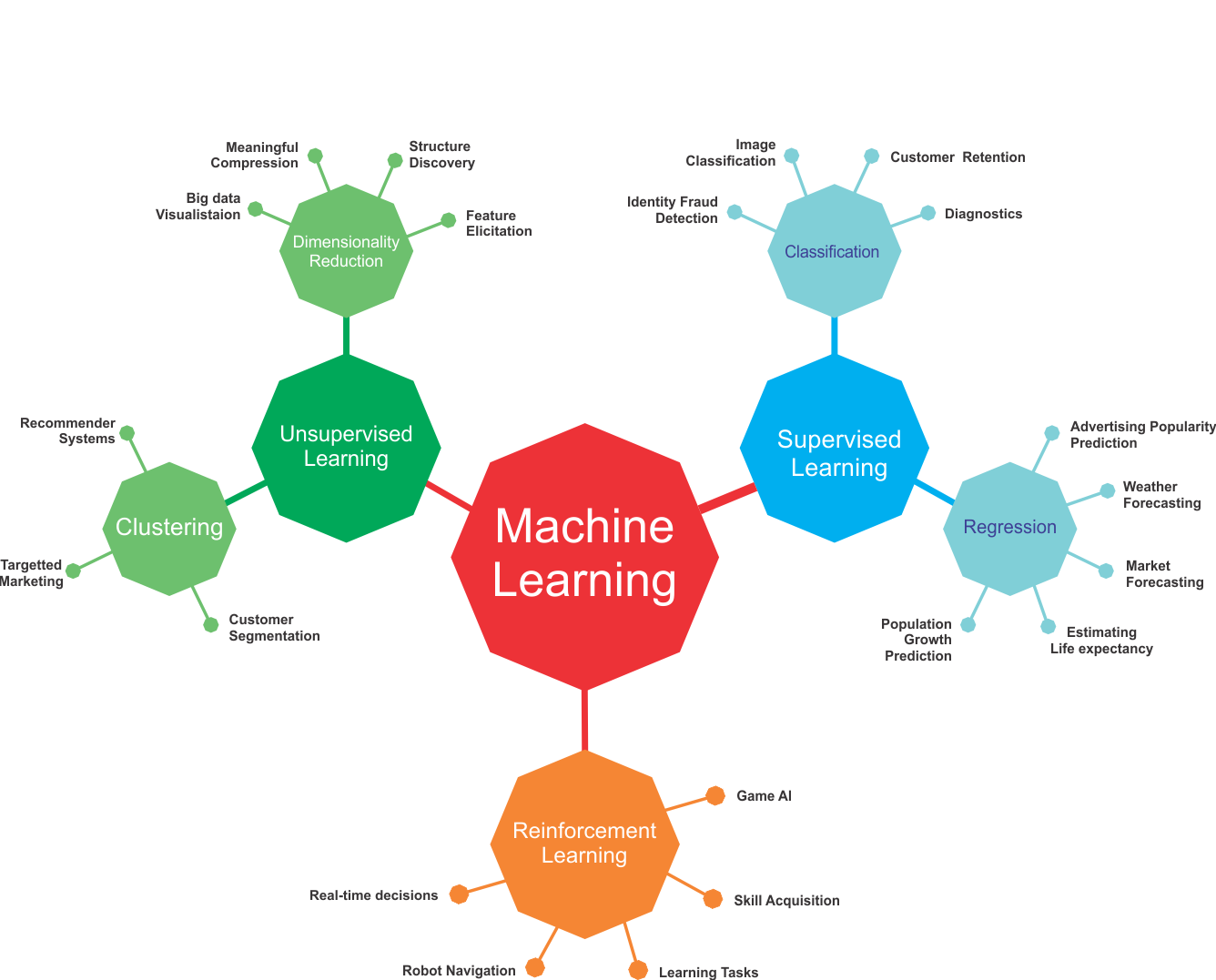
Machine learning algorithms are often categorized as supervised or unsupervised.

Supervised machine learning algorithms can apply what has been learned in the past to new data using labeled examples to predict future events. Starting from the analysis of a known training dataset, the learning algorithm produces an inferred function to make predictions about the output values. The system is able to provide targets for any new input after sufficient training. The learning algorithm can also compare its output with the correct, intended output and find errors in order to modify the model accordingly.

In contrast,**unsupervised machine learning** algorithms are used when the information used to train is neither classified nor labeled. Unsupervised learning studies how systems can infer a function to describe a hidden structure from unlabeled data. The system doesn’t figure out the right output, but it explores the data and can draw inferences from datasets to describe hidden structures from unlabeled data.

**Semi-supervised machine learning** algorithms fall somewhere in between supervised and unsupervised learning, since they use both labeled and unlabeled data for training – typically a small amount of labeled data and a large amount of unlabeled data. The systems that use this method are able to considerably improve learning accuracy. Usually, semi-supervised learning is chosen when the acquired labeled data requires skilled and relevant resources in order to train it / learn from it. Otherwise, acquiring unlabeled data generally doesn’t require additional resources.

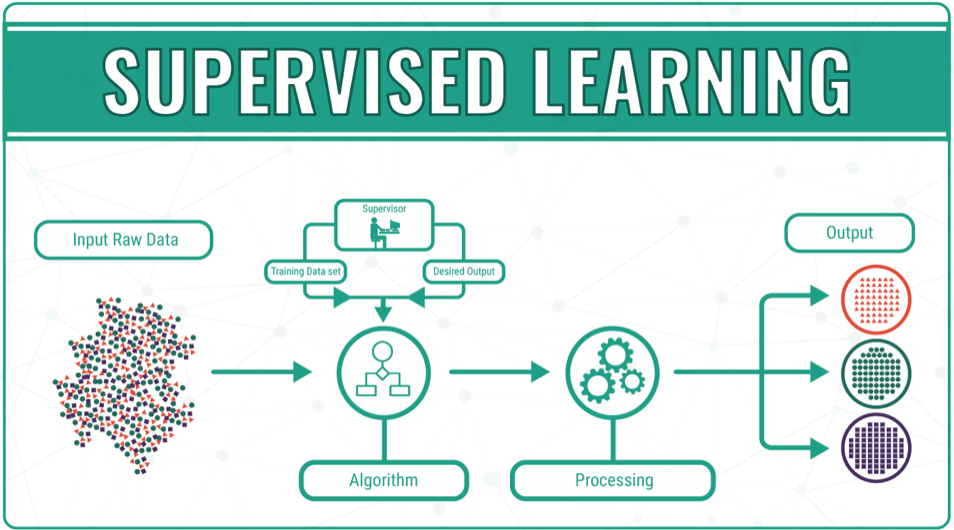
**Reinforcement machine learning** algorithms is a learning method that interacts with its environment by producing actions and discovers errors or rewards. Trial and error search and delayed reward are the most relevant characteristics of reinforcement learning. This method allows machines and software agents to automatically determine the ideal behavior within a specific context in order to maximize its performance. Simple reward feedback is required for the agent to learn which action is best; this is known as the reinforcement signal.

****

*4.9 Supervised machine learning:*

Supervised machine learning algorithms can apply what has been learned in the past to new data using labeled examples to predict future events. Starting from the analysis of a known training dataset, the learning algorithm produces an inferred function to make predictions about the output values.

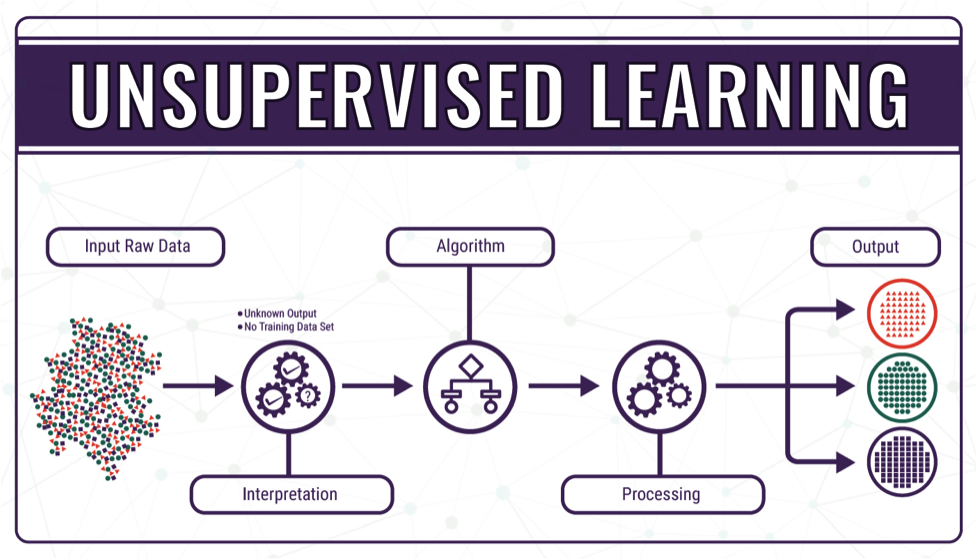
The system is able to provide targets for any new input after sufficient training. The learning algorithm can also compare its output with the correct, intended output and find errors in order to modify the model accordingly.



*4.10 Unsupervised machine learning*:

Unsupervised machine learning algorithms are used when the information used to train is neither classified nor labeled. Unsupervised learning studies how systems can infer a function to describe a hidden structure from unlabeled data.

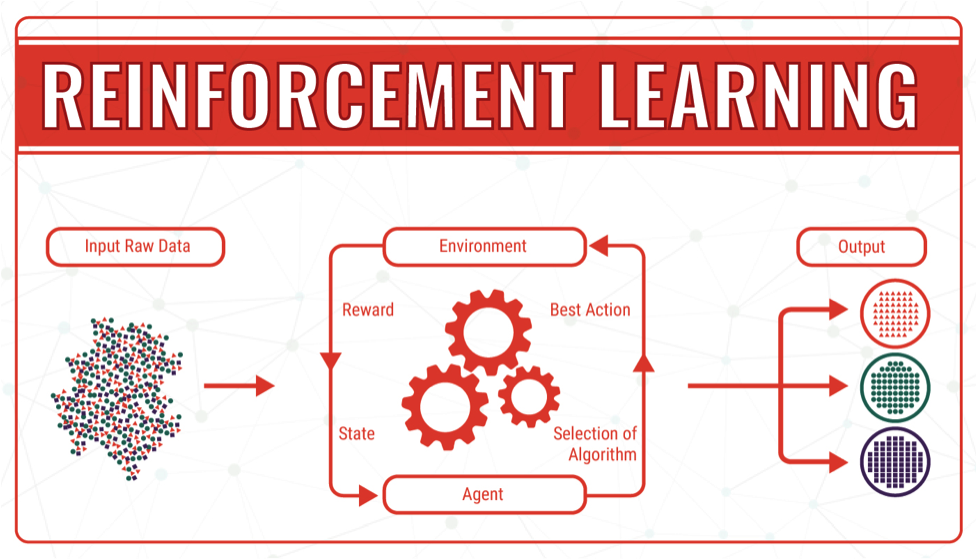
The system doesn’t figure out the right output, but it explores the data and can draw inferences from datasets to describe hidden structures from unlabeled data.

****

.

*4.11Reinforcement machine learning:*

Reinforcement machine learning algorithms is a learning method that interacts with its environment by producing actions and discovers errors or rewards. Trial and error search and delayed reward are the most relevant characteristics of reinforcement learning.

This method allows machines and software agents to automatically determine the ideal behavior within a specific context in order to maximize its performance. Simple reward feedback is required for the agent to learn which action is best; this is known as the reinforcement signal.  
****

*4.12Machine learning Life cycle*

Machine learning has given the computer systems the abilities to automatically learn without being explicitly programmed. But how does a machine learning system work? So, it can be described using the life cycle of machine learning.

Machine learning life cycle is a cyclic process to build an efficient machine learning project. The main purpose of the life cycle is to find a solution to the problem or project.

Machine learning life cycle involves seven major steps, which are given below:

* Gathering Data
* Data preparation
* Data Wrangling
* Analyse Data
* Train the model
* Test the model
* Deployment



The most important thing in the complete process is to understand the problem and to know the purpose of the problem. Therefore, before starting the life cycle, we need to understand the problem because the good result depends on the better understanding of the problem.

In the complete life cycle process, to solve a problem, we create a machine learning system called “model”, and this model is created by providing “training”. But to train a model, we need data, hence, life cycle starts by collecting data.

1. **Gathering Data:**

Data Gathering is the first step of the machine learning life cycle. The goal of this step is to identify and obtain all data-related problems.

In this step, we need to identify the different data sources, as data can be collected from various sources such as files, database, internet, or mobile devices. It is one of the most important steps of the life cycle. The quantity and quality of the collected data will determine the efficiency of the output. The more will be the data, the more accurate will be the prediction.

This step includes the below tasks:

1. Identify various data sources
2. Collect data
3. Integrate the data obtained from different sources

By performing the above task, we get a coherent set of data, also called as a dataset. It will be used in further steps.

1. **Data preparation**

After collecting the data, we need to prepare it for further steps. Data preparation is a step where we put our data into a suitable place and prepare it to use in our machine learning training.

In this step, first, we put all data together, and then randomize the ordering of data.

This step can be further divided into two processes:

**Data-exploration:**

It is used to understand the nature of data that we have to work with. We need to understand the characteristics, format, and quality of data.

A better understanding of data leads to an effective outcome. In this, we find Correlations, general trends, and outliers.

Data pre-processing:

Now the next step is preprocessing of data for its analysis.

1. **Data Wrangling**

Data wrangling is the process of cleaning and converting raw data into a useable format. It is the process of cleaning the data, selecting the variable to use, and transforming the data in a proper format to make it more suitable for analysis in the next step. It is one of the most important steps of the complete process. Cleaning of data is required to address the quality issues.

It is not necessary that data we have collected is always of our use as some of the data may not be useful. In real-world applications, collected data may have various issues, including:

* Missing Values
* Duplicate data
* Invalid data
* Noise

So, we use various filtering techniques to clean the data.

It is mandatory to detect and remove the above issues because it can negatively affect the quality of the outcome.

1. **Data Analysis**

Now the cleaned and prepared data is passed on to the analysis step. This step involves:

* Selection of analytical techniques
* Building models
* Review the result

The aim of this step is to build a machine learning model to analyze the data using various analytical techniques and review the outcome.

It starts with the determination of the type of the problems, where we select the machine learning techniques such as Classification, Regression, Cluster analysis,

 Association, etc. then build the model using prepared data, and evaluate the model.

Hence, in this step, we take the data and use machine learning algorithms to build the model.

1. **Train Model**

Now the next step is to train the model, in this step we train our model to improve its performance for better outcome of the problem.

We use datasets to train the model using various machine learning algorithms. Training a model is required so that it can understand the various patterns, rules, and, features.

1. **Test Model**

Once our machine learning model has been trained on a given dataset, then we test the model. In this step,

We check for the accuracy of our model by providing a test dataset to it.

Testing the model determines the percentage accuracy of the model as per the requirement of project or problem.

1. **Deployment**

The last step of machine learning life cycle is deployment, where we deploy the model in the real-world system.

If the above-prepared model is producing an accurate result as per our requirement with acceptable speed, then we deploy the model in the real system. But before deploying the project,

We will check whether it is improving its performance using available data or not.

The deployment phase is similar to making the final report for a project.

**4.13Applications of Machine learning**

Machine learning is a buzzword for today’s technology, and it is growing very rapidly day by day. We are using machine learning in our daily life even without knowing it such as Google Maps, Google assistant, Alexa, etc. Below are some most trending real-world applications of Machine Learning:

1. *Image Recognition:*

Image recognition is one of the most common applications of machine learning. It is used to identify objects, persons, places, digital images, etc. The popular use case of image recognition and face detection is, Automatic friend tagging suggestion:

Facebook provides us a feature of auto friend tagging suggestion. Whenever we upload a photo with our Facebook friends, then we automatically get a tagging suggestion with name, and the technology behind this is machine learning’s face detection and recognition algorithm.

It is based on the Facebook project named “Deep Face,” which is responsible for face recognition and person identification in the picture.

1. *Speech Recognition*

While using Google, we get an option of “Search by voice,” it comes under speech recognition, and it’s a popular application of machine learning.

Speech recognition is a process of converting voice instructions into text, and it is also known as “Speech to text”, or “Computer speech recognition.” At present, machine learning algorithms are widely used by various applications of speech recognition. Google assistant, Siri, Cortana, and Alexa are using speech recognition technology to follow the voice instructions.

1. *Traffic prediction:*

If we want to visit a new place, we take help of Google Maps, which shows us the correct path with the shortest route and predicts the traffic conditions.

It predicts the traffic conditions such as whether traffic is cleared, slow-moving, or heavily congested with the help of two ways:

Real Time location of the vehicle form Google Map app and sensors

Average time has taken on past days at the same time.

Everyone who is using Google Map is helping this app to make it better. It takes information from the user and sends back to its database to improve the performance.

1. *Product recommendations:*

Machine learning is widely used by various e-commerce and entertainment companies such as Amazon, Netflix, etc., for product recommendation to the user.

Whenever we search for some product on Amazon, then we started getting an advertisement for the same product while internet surfing on the same browser and this is because of machine learning.

Google understands the user interest using various machine learning algorithms and suggests the product as per customer interest.

As similar, when we use Netflix, we find some recommendations for entertainment series, movies, etc., and this is also done with the help of machine learning.

1. Self-driving cars:

One of the most exciting applications of machine learning is self-driving cars. Machine learning plays a significant role in self-driving cars. Tesla, the most popular car manufacturing company is working on self-driving car. It is using unsupervised learning method to train the car models to detect people and objects while driving.

1. *Email Spam and Malware Filtering:*

Whenever we receive a new email, it is filtered automatically as important, normal, and spam. We always receive an important mail in our inbox with the important symbol and spam emails in our spam box, and the technology behind this is Machine learning. Below are some spam filters used by Gmail:

* Content Filter
* Header filter
* General blacklists filter
* Rules-based filters
* Permission filters

Some machine learning algorithms such as Multi-Layer Perceptron, Decision tree, and Naïve Bayes classifier are used for email spam filtering and malware detection.

1. Virtual Personal Assistant:

We have various virtual personal assistants such as Google assistant, Alexa, Cortana, Siri. As the name suggests, they help us in finding the information using our voice instruction. These assistants can help us in various ways just by our voice instructions such as Play music, call someone, Open an email, Scheduling an appointment, etc.

These virtual assistants use machine learning algorithms as an important part.

These assistant record our voice instructions, send it over the server on a cloud, and decode it using ML algorithms and act accordingly.

1. *Online Fraud Detection:*

Machine learning is making our online transaction safe and secure by detecting fraud transaction. Whenever we perform some online transaction, there may be various ways that a fraudulent transaction can take place such as fake accounts, fake ids, and steal money in the middle of a transaction. So to detect this,

 Feed Forward Neural network helps us by checking whether it is a genuine transaction or a fraud transaction.

For each genuine transaction, the output is converted into some hash values, and these values become the input for the next round. For each genuine transaction, there is a specific pattern which gets change for the fraud transaction hence, it detects it and makes our online transactions more secure.

1. *Stock Market trading:*

Machine learning is widely used in stock market trading. In the stock market, there is always a risk of up and downs in shares, so for this machine learning’s long short term memory neural network is used for the prediction of stock market trends.

*10.Medical Diagnosis:*

In medical science, machine learning is used for diseases diagnoses. With this, medical technology is growing very fast and able to build 3D models that can predict the exact position of lesions in the brain.It helps in finding brain tumors and other brain-related diseases easily.

1. *Automatic Language Translation:*

Nowadays, if we visit a new place and we are not aware of the language then it is not a problem at all, as for this also machine learning helps us by converting the text into our known languages. Google’s GNMT (Google Neural Machine Translation) provide this feature, which is a Neural Machine Learning that translates the text into our familiar language, and it called as automatic translation.

That means that for an image, for example, the input might be a matrix of pixels. The first layer might encode the edges and compose the pixels. The next layer might compose an arrangement of edges.

The next layer might encode a nose and eyes. The next layer might recognize that the image contains a face, and so on.

What happens inside the neuron?

The input node takes in information in a numerical form. The information is presented as an activation value where each node is given a number. The higher the number, the greater the activation.

Based on the connection strength (weights) and transfer function, the activation value passes to the next node. Each of the nodes sums the activation values that it receives (it calculates the weighted sum) and modifies that sum based on its transfer function. Next, it applies an activation function. An activation function is a function that’s applied to this particular neuron. From that, the neuron understands if it needs to pass along a signal or not.

Each of the synapses gets assigned weights, which are crucial to Artificial Neural Networks (ANNs). Weights are how ANNs learn. By adjusting the weights, the ANN decides to what extent signals get passed along. When you’re training your network, you’re deciding how the weights are adjusted

The activation runs through the network until it reaches the output nodes. The output nodes then give us the information in a way that we can understand. Your network will use a cost function to compare the output and the actual expected output.

The model performance is evaluated by the cost function. It’s expressed as the difference between the actual value and the predicted value. There are many different cost functions you can use, you’re looking at what the error you have in your network is. You’re working to minimize loss function. (In essence, the lower the loss function, the closer it is to your desired output).

The information goes back, and the neural network begins to learn with the goal of minimizing the cost function by tweaking the weights. This process is called backpropagation.

In forward propagation, information is entered into the input layer and propagates forward through the network to get our output values. We compare the values to our expected results.

Next, we calculate the errors and propagate the info backward. This allows us to train the network and update the weights. (Backpropagation allows us to adjust all the weights simultaneously.)

During this process, because of the way the algorithm is structured, you’re able to adjust all of the weights simultaneously. This allows you to see which part of the error each of your weights in the neural network is responsible for.

4.15How does an artificial neural network learn?

There are two different approaches to get a program to do what you want. First, there’s the specifically guided and hard-programmed approach. You tell the program exactly what you want it to do.

Then there are neural networks. In neural networks,

You tell your network the inputs and what you want for the outputs, and then you let it learn on its own.

By allowing the network to learn on its own, you can avoid the necessity of entering in all of the rules. You can create the architecture and then let it go and learn. Once it’s trained up, you can give it a new image and it will be able to distinguish output.

Feedforward and feedback networks

A feedforward network is a network that contains inputs, outputs, and hidden layers. The signals can only travel in one direction (forward). Input data passes into a layer where calculations are performed. Each processing element computes based upon the weighted sum of its inputs. The new values become the new input values that feed the next layer (feed-forward). This continues through all the layers and determines the output. Feedforward networks are often used in, for example, data mining.

A feedback network (for example, a recurrent neural network) has feedback paths. This means that they can have signals traveling in both directions using loops. All possible connections between neurons are allowed. Since loops are present in this type of network, it becomes a non-linear dynamic system which changes continuously until it reaches a state of equilibrium. Feedback networks are often used in optimization problems where the network looks for the best arrangement of interconnected factors.

The Backpropagation Algorithm

Learning as gradient descent

Multilayered networks are capable of computing a wider range of Boolean functions than networks with a single layer of computing units. However the computational effort needed for finding the correct combination of weights increases substantially when more parameters and more complicated topologies are considered. In this chapter we discuss a popular learning method capable of handling such large learning problems — the backpropagation algorithm. This numerical method was used by different research communities in different contexts, was discovered and rediscovered, until in 1985 it found its way into connectionist AI mainly through the work of the PDP group [382]. It has been one of the most studied and used algorithms for neural networks learning ever since. In this chapter we present a proof of the backpropagation algorithm based on a graphical approach in which the algorithm reduces to a graph labeling problem. This method is not only more general than the usual analytical derivations, which handle only the case of special network topologies,

But also much easier to follow. It also shows how the algorithm can be efficiently implemented in computing systems in which only local information can be transported through the network.

Differentiable activation functions

The backpropagation algorithm looks for the minimum of the error function in weight space using the method of gradient descent. The combination of weights which minimizes the error function is considered to be a solution of the learning problem. Since this method requires computation of the gradient of the error function at each iteration step, we must guarantee the continuity and differentiability of the error function. Obviously we have to use a kind of activation function other than the step function used in perceptrons,

The Backpropagation Algorithm because the composite function produced by interconnected perceptrons is discontinuous, and therefore the error function too. One of the more popular activation functions for backpropagation networks is the sigmoid, a real function sc : IR → (0, 1) defined by the expression sc(x) = 1 1 + e−cx .

The constant c can be selected arbitrarily and its reciprocal 1/c is called the temperature parameter in stochastic neural networks. The shape of the sigmoid changes according to the value of c, as can be seen in Figure 7.1. The graph shows the shape of the sigmoid for c = 1, c = 2 and c = 3. Higher values of c bring the shape of the sigmoid closer to that of the step function and in the limit c → ∞ the sigmoid converges to a step function at the origin

. In order to simplify all expressions derived in this chapter we set c = 1, but after going through this material the reader should be able to generalize all the expressions for a variable c. In the following we call the sigmoid s1(x) just s(x)

The derivative of the sigmoid with respect to x, needed later on in this chapter, is d dx s(x) = e −x (1 + e−x) 2 = s(x)(1 – s(x)). We have already shown that, in the case of perceptrons, a symmetrical activation function has some advantages for learning. An alternative to the sigmoid is the symmetrical sigmoid S(x) defined as S(x) = 2s(x) – 1 = 1 – e −x 1 + e−x . This is nothing but the hyperbolic tangent for the argument x/2 whose shape is shown in Figure 7.2 (upper right). The figure shows four types of continuous “squashing” functions. The ramp function (lower right) can also be used in learning algorithms taking care to avoid the two points where the derivative is undefined.

Many other kinds of activation functions have been proposed and the backpropagation algorithm is applicable to all of them. A differentiable activation function makes the function computed by a neural network differentiable (assuming that the integration function at each node is just the sum of the inputs), since the network itself computes only function compositions. The error function also becomes differentiable. Figure 7.3 shows the smoothing produced by a sigmoid in a step of the error function. Since we want to follow the gradient direction to find the minimum of this function, it is important that no regions exist in which the error function is completely flat. As the sigmoid always has a positive derivative, the slope of the error function provides a greater or lesser descent direction which can be followed

.

We can think of our search algorithm as a physical process in which a small sphere is allowed to roll on the surface of the error function until it reaches the bottom.

Regions in input space The sigmoid’s output range contains all numbers strictly between 0 and 1. Both extreme values can only be reached asymptotically. The computing units considered in this chapter evaluate the sigmoid using the net amount of excitation as its argument. Given weights w1, . . ., wn and a bias −θ, a sigmoidal unit computes for the input x1, . . .,

Xn

The output 1 1 + exp (Pn i=1 wixi – θ) .

The Backpropagation Algorithm of weights so that the network function ϕ approximates a given function f as closely as possible. However, we are not given

The function f explicitly but only implicitly through some examples. Consider a feed-forward network with n input and m output units. It can consist of any number of hidden units and can exhibit any desired feed-forward connection pattern. We are also given a training set {(x1, t1), . . .,(xp, tp)}

Consisting of p ordered pairs of n- and m-dimensional vectors, which are called the input and output patterns. Let the primitive functions at each node of the network be continuous and differentiable.

The weights of the edges are real numbers selected at random. When the input pattern xi from the training set is presented to this network, it produces an output oi different in general from the target ti .

What we want is to make oi and ti identical for I = 1, . . ., p, by using a learning algorithm. More precisely, we want to minimize the error function of the network, defined as E = 1 2 Xp i=1 koi – tik 2 .

After minimizing this function for the training set, new unknown input patterns are presented to the network and we expect it to interpolate. The network must recognize whether a new input vector is similar to learned patterns and produce a similar output.

The backpropagation algorithm is used to find a local minimum of the error function. The network is initialized with randomly chosen weights.

The gradient of the error function is computed and used to correct the initial weights. Our task is to compute this gradient recursively

The first step of the minimization process consists of extending the network, so that it computes the error function automatically.

How this is done. Every one of the j output units of the network is connected to a node which evaluates the function 1 2 (oij – tij ) 2 , where oij and tij denote the j-th component of the output vector oi and of the target ti

. The outputs of the additional m nodes are collected at a node which adds them up and gives the sum Ei as its output.

The same network extension has to be built for each pattern ti . A computing unit collects all quadratic errors and outputs their sum E1 + · · · + Ep.

The output of this extended network is the error function E. We now have a network capable of calculating the total error for a given training set.

The weights in the network are the only parameters that can be modified to make the quadratic error E as low as possible. Because E is calculated by the extended network exclusively through composition of the node functions,

It is a continuous and differentiable function of the ` weights w1, w2, . . . , w` in the network.

We can thus minimize E by using an iterative process of gradient descent, for which we need to calculate the gradient

∇E = ( ∂E ∂w1 , ∂E ∂w2 , . . ., ∂E ∂w` )

Each weight is updated using the increment

T ∆wi = −γ ∂E ∂wi for I = 1, . . ., `

Where γ represents a learning constant, i.e., a proportionality parameter which defines the step length of each iteration in the negative gradient direction.

With this extension of the original network the whole learning problem now reduces to the question of calculating the gradient of a network function with respect to its weights.

Once we have a method to compute this gradient, we can adjust the network weights iteratively.

Once we have a method to compute this gradient, we can adjust the network weights iteratively.

In this way we expect to find a minimum of the error function, where ∇E = 0.

What is a weighted sum?

Inputs to a neuron can either be features from a training set or outputs from the neurons of a previous layer. Each connection between two neurons has a unique synapse with a unique weight attached. If you want to get from one neuron to the next, you have to travel along the synapse and pay the “toll” (weight). The neuron then applies an activation function to the sum of the weighted inputs from each incoming synapse

. It passes the result on to all the neurons in the next layer. When we talk about updating weights in a network, we’re talking about adjusting the weights on these synapses.

A neuron’s input is the sum of weighted outputs from all the neurons in the previous layer. Each input is multiplied by the weight associated with the synapse connecting the input to the current neuron. If there are 3 inputs or neurons in the previous layer, each neuron in the current layer will have 3 distinct weights: one for each synapse.

In a nutshell, the activation function of a node defines the output of that node.

The activation function (or transfer function) translates the input signals to output signals. It maps the output values on a range like 0 to 1 or -1 to 1.

It’s an abstraction that represents the rate of action potential firing in the cell. It’s a number that represents the likelihood that the cell will fire

. At it’s simplest, the function is binary: yes (the neuron fires) or no (the neuron doesn’t fire).

The output can be either 0 or 1 (on/off or yes/no), or it can be anywhere in a range.

If you were using a function that maps a range between 0 and 1 to determine the likelihood that an image is a cat, for example, an output of 0.9 would show a 90% probability that your image is, in fact, a cat.

4.17 What is an activation function?

In a nutshell, the activation function of a node defines the output of that node.

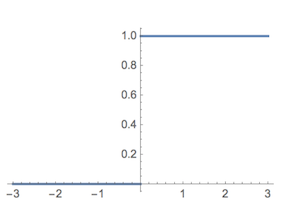
The activation function (or transfer function) translates the input signals to output signals. It maps the output values on a range like 0 to 1 or -1 to 1. It’s an abstraction that represents the rate of action potential firing in the cell. It’s a number that represents the likelihood that the cell will fire. At it’s simplest, the function is binary: yes (the neuron fires) or no (the neuron doesn’t fire)

. The output can be either 0 or 1 (on/off or yes/no), or it can be anywhere in a range

There are many activation functions, but these are the four very common ones:

**Threshold function**

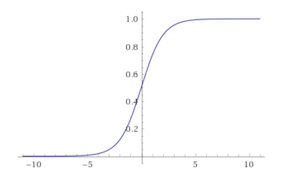
This is a step function. If the summed value of the input reaches a certain threshold the function passes on 0. If it’s equal to or more than zero, then it would pass on 1. It’s a very rigid, straightforward, yes or no function.



Example threshold function

**Sigmoid function**

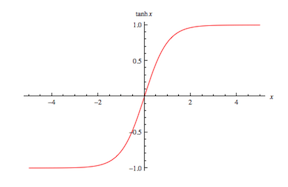
This function is used in logistic regression. Unlike the threshold function, it’s a smooth, gradual progression from 0 to 1. It’s useful in the output layer and is used heavily for linear regression



Example Sigmoid function

Hyperbolic Tangent Function

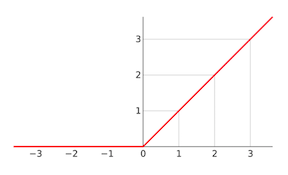
This function is very similar to the sigmoid function. But unlike the sigmoid function which goes from 0 to 1, the value goes below zero, from -1 to 1. Even though this isn’t a lot like what happens in a brain, this function gives better results when it comes to training neural networks. Neural networks sometimes get “stuck” during training with the sigmoid function. This happens when there’s a lot of strongly negative input that keeps the output near zero, which messes with the learning process.



Example of Hyperbolic tangent Function

**Rectifier function**

This might be the most popular activation function in the universe of neural networks. It’s the most efficient and biologically plausible. Even though it has a kink, it’s smooth and gradual after the kink at 0. This means, for example, that your output would be either “no” or a percentage of “yes.” This function doesn’t require normalization or other complicated calculations.



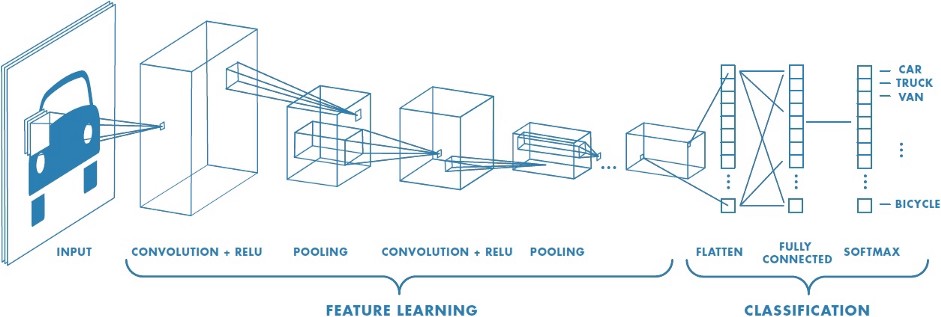
Example of rectifier function

4.18 What is CNN

In neural networks, Convolutional neural network (ConvNets or CNNs) is one of the main categories to do images recognition, images classifications. Objects detections, recognition faces etc., are some of the areas where CNNs are widely used.

CNN image classifications takes an input image, process it and classify it under certain categories (Eg., Dog, Cat, Tiger, Lion). Computers sees an input image as array of pixels and it depends on the image resolution. Based on the image resolution, it will see h x w x d( h = Height, w = Width, d = Dimension ). Eg., An image of 6 x 6 x 3 array of matrix of RGB (3 refers to RGB values) and an image of 4 x 4 x 1 array of matrix of grayscale image

*Technically, deep learning CNN models to train and test, each input image will pass it through a series of convolution layers with filters (Kernals), Pooling, fully connected layers (FC) and apply Softmax function to classify an object with probabilistic values between 0 and 1. The below figure is a complete flow of CNN to process an input image and classifies the objects based on values.*



**Convolution Layer**

Convolution is the first layer to extract features from an input image. Convolution preserves the relationship between pixels by learning image features using small squares of input data. It is a mathematical operation that takes two inputs such as image matrix and a filter or kernel.

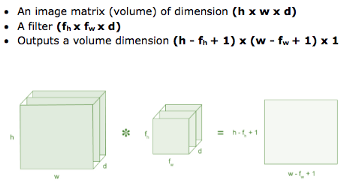


Figure : Image matrix multiplies kernel or filter matrix

Consider a 5 x 5 whose image pixel values are 0, 1 and filter matrix 3 x 3 as shown in below

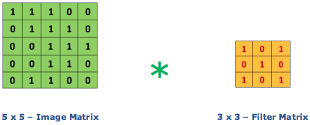


Figure 4: Image matrix multiplies kernel or filter matrix

Then the convolution of 5 x 5 image matrix multiplies with 3 x 3 filter matrix which is called “Feature Map” as output shown in below

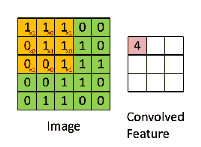
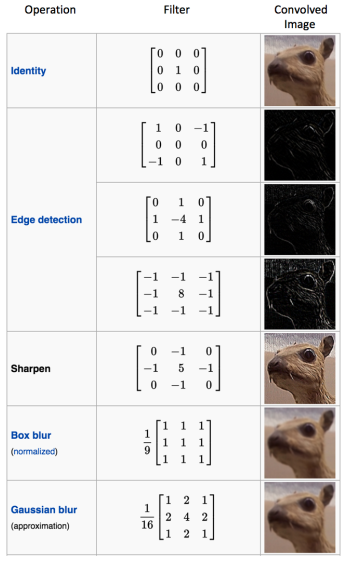


Figure 5: 3 x 3 Output matrix

Convolution of an image with different filters can perform operations such as edge detection, blur and sharpen by applying filters. The below example shows various convolution image after applying different types of filters (Kernels)



Strides

Stride is the number of pixels shifts over the input matrix. When the stride is 1 then we move the filters to 1 pixel at a time. When the stride is 2 then we move the filters to 2 pixels at a time and so on. The below figure shows convolution would work with a stride of 2

Padding

Sometimes filter does not fit perfectly fit the input image. We have two options:

Pad the picture with zeros (zero-padding) so that it fits

Drop the part of the image where the filter did not fit. This is called valid padding which keeps only valid part of the image

Non Linearity (ReLU)

ReLU stands for Rectified Linear Unit for a non-linear operation. The output is ƒ(x) = max(0,x).

Why ReLU is important : ReLU’s purpose is to introduce non-linearity in our ConvNet. Since, the real world data would want our ConvNet to learn would be non-negative linear values.

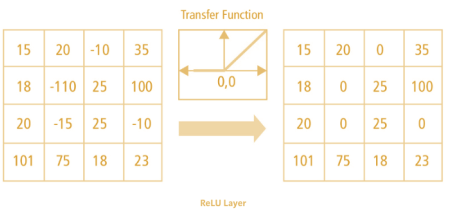


Fig Relu function

There are other non linear functions such as tanh or sigmoid that can also be used instead of ReLU. Most of the data scientists use ReLU since performance wise ReLU is better than the other two.

Pooling Layer

Pooling layers section would reduce the number of parameters when the images are too large. Spatial pooling also called subsampling or downsampling

Which reduces the dimensionality of each map but retains important information.

* Spatial pooling can be of different types:
* Max pooling
* Average pooling
* Sum pooling

Max pooling takes the largest element from the rectified feature map. Taking the largest element could also take the average pooling. Sum of all elements in the feature map call as sum pooling.

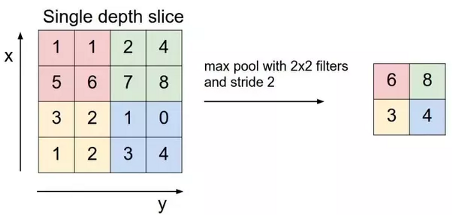


Figure :max pooling

Fully connected layer:

The layer we call as FC layer, we flattened our matrix into vector and feed it into a fully connected layer like a neural network.

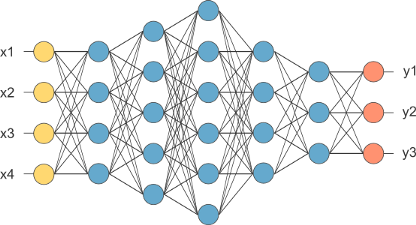


Figure: fully connected layer

In the above diagram, the feature map matrix will be converted as vector (x1, x2, x3, …). With the fully connected layers, we combined these features together to create a model. Finally, we have an activation function such as softmax or sigmoid to classify the outputs as cat, dog, car, truck etc.,

4.20 Introduction to TENSORFLOW:

TensorFlow is an open source library for numerical computation and large-scale machine learning. TensorFlow bundles together a slew of machine learning and deep learning (aka neural networking) models and algorithms and makes them useful by way of a common metaphor. It uses Python to provide a convenient front-end API for building applications with the framework, while executing those applications in high-performance C++.

TensorFlow can train and run deep neural networks for handwritten digit classification, image recognition, word embeddings, recurrent neural networks, sequence-to-sequence models for machine translation, natural language processing, and PDE (partial differential equation) based simulations. Best of all, TensorFlow supports production prediction at scale, with the same models used for training.

4.20.1How TensorFlow works :

TensorFlow allows developers to create dataflow graphs—structures that describe how data moves through a graph, or a series of processing nodes. Each node in the graph represents a mathematical operation, and each connection or edge between nodes is a multidimensional data array, or tensor.

TensorFlow provides all of this for the programmer by way of the Python language. Python is easy to learn and work with, and provides convenient ways to express how high-level abstractions can be coupled together. Nodes and tensors in TensorFlow are Python objects, and TensorFlow applications are themselves Python applications.

The actual math operations, however, are not performed in Python. The libraries of transformations that are available through TensorFlow are written as high-performance C++ binaries. Python just directs traffic between the pieces, and provides high-level programming abstractions to hook them together.

TensorFlow applications can be run on most any target that’s convenient: a local machine, a cluster in the cloud, iOS and Android devices, CPUs or GPUs. If you use Google’s own cloud, you can run TensorFlow on Google’s custom TensorFlow Processing Unit (TPU) silicon for further acceleration. The resulting models created by TensorFlow, though, can be deployed on most any device where they will be used to serve predictions.

TensorFlow 2.0, released in October 2019, revamped the framework in many ways based on user feedback, to make it easier to work with (e.g., by using the relatively simple Keras API for model training) and more performant. Distributed training is easier to run thanks to a new API, and support for TensorFlow Lite makes it possible to deploy models on a greater variety of platforms. However, code written for earlier versions of TensorFlow must be rewritten—sometimes only slightly, sometimes significantly—to take maximum advantage of new TensorFlow 2.0 features.

**4.21 Introduction to Keras:**

Keras is one of the leading high-level neural networks APIs. It is written in Python and supports multiple back-end neural network computation engines.

Keras principles

Keras was created to be user friendly, modular, easy to extend, and to work with Python. The API was “designed for human beings, not machines,” and “follows best practices for reducing cognitive load.”

Neural layers, cost functions, optimizers, initialization schemes, activation functions, and regularization schemes are all standalone modules that you can combine to create new models. New modules are simple to add, as new classes and functions. Models are defined in Python code, not separate model configuration files.

3.15.1Why Keras?

The biggest reasons to use Keras stem from its guiding principles, primarily the one about being user friendly. Beyond ease of learning and ease of model building, Keras offers the advantages of broad adoption, support for a wide range of production deployment options, integration with at least five back-end engines (TensorFlow, CNTK, Theano, MXNet, and PlaidML), and strong support for multiple GPUs and distributed training. Plus, Keras is backed by Google, Microsoft, Amazon, Apple, Nvidia, Uber, and others.

**Chapter 5**

***Project Analysis***

We plan to Integrate Jarvis with mobile using python and AI , to provide a

synchronized experience between the two connected devices.

Further, in the long run, Jarvis is planned to feature auto deployment

supporting elastic beanstalk, backup files, and all operations which a

general Server Administrator does. The functionality would be seamless

enough to replace the Server Administrator with Jarvis.

**Chapter 6**

***Summing UP***

**CONCLUSION**

Through this voice assistant, we have automated various services using a single

line command. It eases most of the tasks of the user like searching the web,

retrieving weather forecast details, vocabulary help and medical related queries.

We aim to make this project a complete server assistant and make it smart

enough to act as a replacement for a general server administration. The future

plans include integrating Jarvis with mobile using React Native to provide a

synchronised experience between the two connected devices. Further, in the

long run, Jarvis is planned to feature auto deployment supporting elastic

beanstalk, backup files, and all operations which a general Server Administrator

does. The functionality would be seamless enough to replace the Server

Administrator with Jarvis

**PROBLEM STATEMENT**

We are all well aware about Cortana, Siri, Google Assistant and many other virtual

assistants which are designed to aid the tasks of users in Windows, Android and

iOS platforms.

PURPOSE

This Software aims at developing a personal assistant for python-based systems.

The main purpose of the software is to perform the tasks of the user at certain

commands, provided in either of the ways, speech or text. It will ease most of

the work of the user as a complete task can be done on a single command. AR

draws its inspiration from Virtual assistants like Cortana for Windows and Siri

for iOS. Users can interact with the assistant either through voice commands or

keyboard input.

**PRODUCT GOALS AND OBJECTIVES**

Currently, the project aims to provide the Linux Users with a Virtual Assistant

that would not only aid in their daily routine tasks like searching the web,

extracting weather data, vocabulary help and many others but also help in

automation of various activities.

In the long run, we aim to develop a complete server assistant, by automating

the entire server management process - deployment, backups, auto-scaling,

logging, monitoring and make it smart enough to act as a replacement for a General server administrative .

SCOPE

Presently, Jarvis is being developed as an automation tool and virtual assistant.

Among the Various roles played by Jarvis are:

1. Search Engine with voice interactions

2. Medical diagnosis with Medicine aid.

3. Reminder and To-Do application.

4. Vocabulary App to show meanings and correct spelling errors.

5. Weather Forecasting Application.

There shall be proper Documentation available on its Official Github repository

for making further development easy and we aim to release our virtual assistant as

an Open Source Software where modifications and contributions by the

community are warmly welcomed.

“**RESPONSES FROM AR-VPA”**

**THANK-YOU**

****