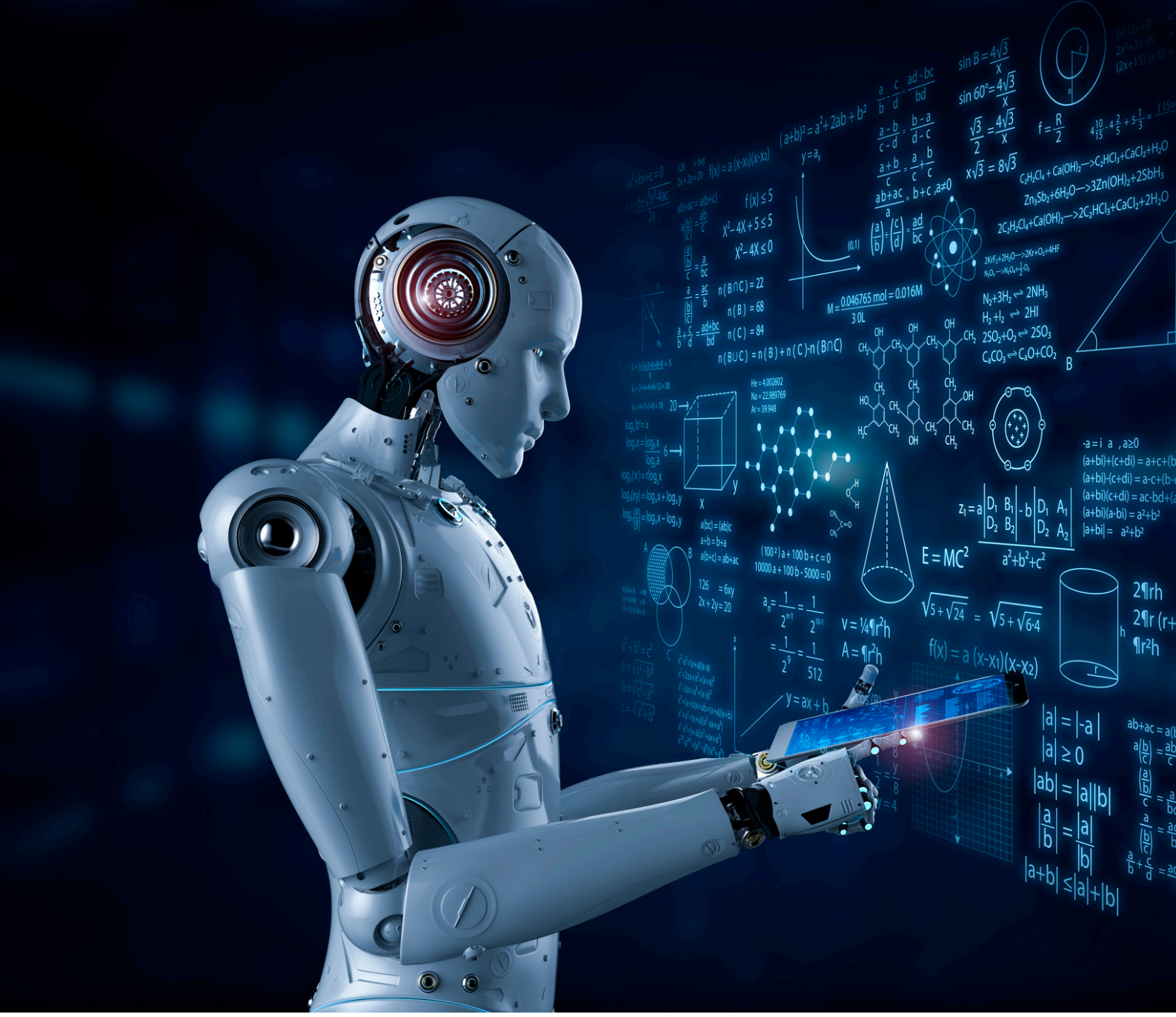


Machine Learning and Deep Learning in chatbot

Subject: Artificial Intelligence

Professor: Dr. Anand Nayyar
Team members: Nguyen Thi Hong Yen - 2320120895
Tran Thanh Danh - 2321121844
Class: CS 376 D

FOREWORD

In recent years, science and technology are gradually reaching a new level. Although new in the field of science and technology, Chatbots are being researched and developed at breakneck speed by research centers, universities, and academia, etc. be applied to this new technology. Chatbots are a rudimentary form of artificial intelligence software, a computer-generated program that conducts conversations through text, audio, and touches input methods that can answer questions. and handling situations is a tool that can communicate and interact with humans through pre-programmed artificial intelligence. There are many tools and libraries supporting Chatbots such as Dialogflow, Wit.ai, Watson Conversation Service, Microsoft 's LUIS, Google Natural Language API, Amazon Lex, etc.

A chatbot or conversational agent is a software that can communicate with a human by using natural language. One of the essential tasks in artificial intelligence and natural language processing is the modeling of conversation. Since the beginning of artificial intelligence, it's been the hardest challenge to create a good chatbot. Although chatbots can perform many tasks, the primary function they have to play is to understand the utterances of humans and to respond to them appropriately. In the past, simple statistic methods or handwritten templates and rules were used for the constructions of chatbot architectures. With the increasing learning capabilities, end-to-end neural networks have taken the place of these models in around 2015. Especially now, the encoder-decoder recurrent model is dominant in the modeling of conversations. This architecture is taken from the neural machine translation domain, and it performed very well there. Until now, plenty of features and variations are introduced that have remarkably enhanced the conversational capabilities of chatbots.

Nowadays, many companies hope can create Chatbot applies machine learning, deep learning is hoping to develop bots that can communicate naturally like humans use NLP (Natural language processing) or Deep learning technology to create bots that have the possibility on. But with all the hype around AI, telling the truth from fiction is sometimes difficult. In this series. I would like to present some of the Deep Learning techniques and machine learning methods used to create conversational agents (Chatbot). The role of machine learning and deep learning in chatbot and the limitation or problem of applying machine learning and deep learning to a particular chatbot.

TABLE OF CONTENTS

FOREWORD **2**

PLEDGED **5**

ACKNOWLEDGMENT **5**

A. OVERVIEW **6**

1. Introduction **6**

2. Objectives **7**

3. The content of research **7**

4. Boundary **7**

5. Layouts **7**

B. INTRODUCTION **7**

I. An overview of Artificial Intelligent **8**

1. What is Artificial Intelligence? 8

2. Brief history of Artificial Intelligence, Machine Learning and Deep Learning 8

2.1. Artificial intelligence 8

2.2. Machine learning 9

2.3. Deep learning 11

3. Machine learning in Artificial Intelligence 12

3.1. What is machine learning? 12

3.2. Machine learning methods 13

3.2.1. Supervised learning 13

3.2.2. Unsupervised learning 14

3.2.3. Semi-supervised learning 15

3.2.4. Reinforcement learning 15

4. Deep learning in Machine Learning 16

4.1. What is deep learning? 16

4.2. The principle definition of deep learning 17

4.2.1. Deep Convolutional neural networks (CNN) 17

4.2.2. Deep Recurrent neural networks (RNN) 17

4.2.3. Deep Artificial neural networks (ANN) 18

II. An overview of chatbot **18**

1. Chatbot definition 18

2. Brief History of chatbot 19

3. How are machine learning and deep learning used in chatbot? 21

4. The roles of machine learning and deep learning in chatbot 25

5. The limitation of chatbot: 26

6. Component of chatbot 28

7. The type of chatbot: 30

8. How it works? 31

9. Methods are used in chatbot 32

9.1. Artificial Neural Networks 32

9.2. Natural Language Understanding 32

9.2.1. Entities: 32

9.2.2. Intents: 33

9.2.3. Context: 33

9.3. Natural Language Processing 33

9.4. Dataset 34

10. Some platform support to develop chatbot 34

10.1. Dialog flow 34

10.1.1. Definition 34

10.1.2. History and Operation 34

10.2. Another platform 34

11. Example chatbot application 35

12. The development trends of chatbot (commerce chatbot) 35

C. TECHNOLOGY DESCRIPTION 36

I. Machine Learning in chatbot 36
1. Supervised learning 36
1.1. Linear Regression 36
1.2. Decision Trees: 37
1.3. Neural Networks: 38
1.4. Support Vector Machines (SVM): 39
1.5. Genetic algorithms 41
1.6. Logistic regression: 42
1.7. Naive Bayes 44
1.8. Random forests: 45
2. Unsupervised learning 46
2.1. K-means clustering 46
2.2. Association Rules: 47
3. Reinforcement learning 48
3.1. Q-Learning 48
3.2. Temporal Difference 50
3.3. Deep Adversarial Networks: 50
4. **Semi-supervised learning 53**

II. Deep learning in chatbot: 54
1. Deep Convolutional neural networks (CNN) 54
2. Deep Recurrent neural networks (RNN) 55
3. Deep Artificial neural networks (ANN) 57

III. Some techniques in chatbot 60
1. Artificial Neural Networks 60
2. Natural Language Understanding 62
2.1. Entities 62
2.2. Intents 62
2.3. Context 63
3. Natural Language Processing 63
4. Dataset 64

D. DESIGN 65

1. Recommender Systems 65

2. Pricing Optimization 66

3. Predicting Customer Behavior 67

4. Social Media: Tracking Brand & Customer Sentiment 67

E. CASE STUDY 68

I. Shopping problem and Solution to solve it 68
1. The issues of traditional shopping 68
1.1. Online payment 68
1.2. Time 68
2. Why online shopping is more important now? 68
3. The role of applying deep learning and machine learning on chatbot shopping 69
4. Chatbot shopping in Vietnam 70
5. The solution to solve the problem on chatbot shopping in Vietnam 72
5.1. Solutions to financial risk 72
5.2. Solutions to psychological risk 72
5.3. Solutions to product performance risk 73

F. CONCLUSION AND DEVELOPMENT 74

I. Conclusion 74

II. Development: 74

G. REFERENCES 75

PLEDGED

The project is performed by my group and research some documents. We do confirm that we don't copy from any document and existing project before.

Performed Students
Nguyen Thi Hong Yen
Tran Thanh Danh

ACKNOWLEDGMENT

We would like to express our sincere thanks to Dr. Anand Nayyar - Professor of Duy Tan University devotedly guided during the group project. He created many favorable conditions and gave valuable advice to help us complete the group project. Sincerely thank you so much to you have dedicated to teaching, imparting on us valuable knowledge, and facilitating help during the learning process during the course, and help us have a solid theoretical basis to study this project "Machine learning and deep learning in chatbot"

Thank you!

Performed Students
Nguyen Thi Hong Yen
Tran Thanh Danh

A. OVERVIEW

1. Introduction

In recent years, AI - Artificial Intelligence, Machine Learning, Deep Learning has emerged as evidence of the fourth industrial revolution. As the computing power of computers has risen to a new height and the vast amount of data collected by major technology firms, Machine Learning has come a long way and a field. just came out called Deep Learning. Deep Learning helped computers do what seemed impossible 10 years ago: sorting thousands of different objects in photos, creating photo captions, mimicking human speech, and writing., interact with people, or even write music or music. Artificial Intelligence is creeping into every area of our life that we may not realize. Self-driving cars from Google and Tesla, Facebook's Face ID self-tagging system, Apple's Siri virtual assistant, Amazon's product suggestion system, Netflix's movie suggestion system, Google's AlphaGo Go player DeepMind, etc. are just a few of the countless applications of AI / Machine Learning.

One of the applications of machine learning and deep learning is Chatbots is still being researched and developed at a fast pace by research centers, universities, institutes, etc. although it is still new in science and technology. A lot of applications in different fields like entertainment, health, commerce, automation, etc. Since chatbots. Strongly developed in 2016, it has quickly become one of the most interesting technology trends in that period and so far, in fact, the rise of chatbots has led to the prediction of Gartner (the world's leading information technology consulting and research company, providing insight into the technology needed for customers). to make the right decisions every day) that more than 85% of customer interactions will be managed without people by 2020.

Chatbots developed based on a combination of pre-existing scenarios and self-study during the interaction. We will interact with chatbots via messaging platforms. With the questions raised, Chatbots use natural language processing systems to analyze the data and then select the learning algorithms. Machine to give different types of feedback, they will predict and respond as accurately as possible. Chatbots use multiple systems to scan keywords inside the input, then the bot starts an action, pulls an answer with the most relevant keywords and responds with information from a database / API, or handed over to humans. If that situation has not happened (not included in the data), Chatbot will ignore it but will at the same time self-study to apply for future conversations.

One of the factors that make Chatbots' strength is its ability to learn by itself. The more you use and interact with users, the more "intelligent" Chatbot platform. Smart chatbots are capable of learning on their own based on the input data without having to be programmed specifically (that is called Machine Learning method). This makes it easy for developers to create chat programs and automate conversations with users. Currently, Chatbots are being widely applied in the fields of human life, including the help for the process of automating lighting processes (automatic opening and closing), irrigation., manage time, crops, ... for agriculture to minimize effort, cost, and time. And Dialogflow is one of the powerful tools for Chatbots, helping users to build up Automatic and optimal system.

2. Objectives

In this project, we focus on researching and deeply understanding in the machine learning and deep learning. The roles of machine learning and deep learning in common and also in particular chatbot. Some limitations and problem when applying machine learning and deep learning in chatbot. Give some sub-methods or descriptive technology in deep learning, machine learning and also in particular chatbot. We represent the major algorithm and specific technique in sub-methods. We introduce some platforms used in popular in chatbot such as Dialogflow. We give an example for each part by diagram, explaining and conclude what we actually understand.

3. The content of research

- Content1: Research and analyze the topic. Focusing on 3 field such as Artificial Intelligent, Deep learning, Machine Learning and Chatbot
- Content2: Analyze the difference between Artificial intelligent, Machine learning and Deep learning
- Content3: Research the role of deep learning and machine learning in Chatbot
- Content4: Research and understand about chatbot and the technique
- Content5: Research how to apply machine learning and deep learning in chatbot
- Content6: Research some methods of deep learning and machine learning in chatbot
- Content7: Research and write descriptive technologies algorithm methods of deep learning and machine learning in chatbot
- Content8: Case study: Research the application of machine learning and deep learning in chatbot in Retail Industry. The problem and the solution
- Content9: Write document

4. Boundary

Research about all key words in the topic.
Research and introduce about technology.
Research descriptive technology such as algorithm and example.
Design prototype.

5. Layouts

- Overview of the topic
- Introduction of technology
- Descriptive technology
- Design
- Case study
- Conclusion and development in the future

B. INTRODUCTION

I. An overview of Artificial Intelligent

1. What is Artificial Intelligence?

In modern-day, you can see Artificial Intelligence appears at almost all aspects of our life. Artificial Intelligence is the technology trend that leads technology companies are heading to, such as Google, Apple, Microsoft, Amazon, etc. These companies invest heavily in Artificial Intelligence to create products for human benefits.
Artificial Intelligence is the intelligence represented by any artificial system. The term is often used to refer to computers with an unspecified purpose and the science of theories and applications of artificial intelligence. It involves the behavior, learning, intelligent adaptability of machines.
Bellman (1978) defines: "*artificial intelligence is the automation of activities in accordance with human thinking, such as decision-making activities, problem solving, etc."*
Rich and Knight (1991) argued that: "*Artificial intelligence is the science of how computers can perform tasks that humans currently do better than computers".*
Each concept and definition have its own right, but for simplicity we can understand Artificial intelligence refers to the simulation of human intelligence in machines that are programmed to think like humans and mimic their actions. The term may also be applied to any machine that exhibits traits associated with a human mind such as learning and problem-solving. The ideal characteristic of artificial intelligence is its ability to rationalize and take actions that have the best chance of achieving a specific goal.
AI is which helps us to answer automated messages, learns how to drive a car, flies an airplane so we can sit there and watch without having to control, rearranges photos of different outings on separate albums. Even AI can help us manage our homes when we are away or shopping.

2. Brief history of Artificial Intelligence, Machine Learning and Deep Learning

2.1. Artificial intelligence

1642: First mechanical calculating machine-built mathematician and inventor Blaise Pascal.
1873: First design for a programmable machine, by Charles Babbage and Ada Lovelace.
1943: Foundations of neural networks established by Warren McCulloch and Walter Pitts, drawing parallels between the human brain and computing machines.
1950: Alan Turing introduces a test – the Turing test – as a way of testing a machine's intelligence.
1965: ELIZA, a natural language program, is created. ELIZA handles dialogue on any topic; similar in concept to today's chatbots.
1970s to 1980s: After the boom of the early research projects, no new software and AI programs were developed.
1997: Deep Blue, the IBM chess computer, won a tournament of six games against the then world chess champion Garry Kasparov – one year before, the computer had still lost against him. All over the world, the victory of Deep Blue caused anxiety and fear that machines would prevail over humans in the near future.
1980s: Edward Feigenbaum creates expert systems which emulate decisions of human experts.
1997: Computer program Deep Blue beats world chess champion Garry Kasparov.
2002: iRobot launches Roomba, an autonomous vacuum cleaner that avoids obstacles.
2009: Google builds the first self-driving car to handle urban conditions.
2011: IBM's Watson defeats champions of US game show Jeopardy.
2011 – 2014: Personal assistants like Siri, Google Now, Cortana, Alexa use speech recognition to answer questions and perform simple tasks.

2014: Ian Goodfellow comes up with Generative Adversarial Networks (GAN).
2016: AlphaGo beats professional Go player Lee Sedol 4-1.
2018: Most universities have courses in Artificial Intelligence.

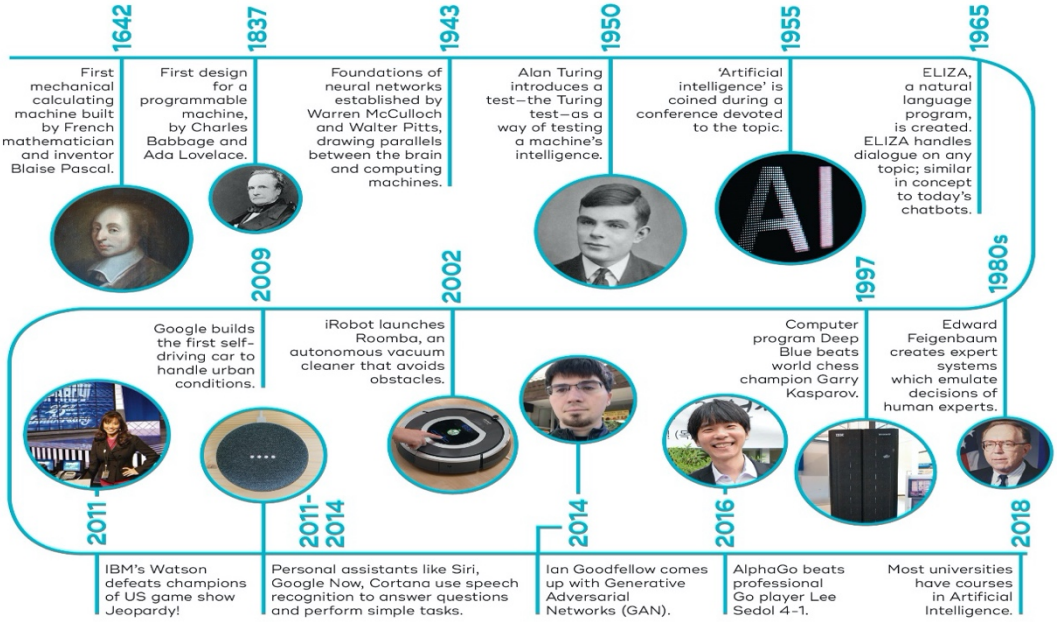


Figure 1: A brief history of Artificial Intelligence.

Example of AI:
- Tesla, Inc., an American electric vehicles and clean energy company use Artificial Intelligence to create an autopilot system for their self-driving Tesla cars.
- Facebook, Inc. is an American social media and technology company use Artificial Intelligence to identify and automatically highlights faces and suggest friends to tag. Facebook also uses AI to personalize your newsfeed and ensure you're seeing posts that interest you. And, of particular business interest to Facebook is showing ads that are relevant to your interests.

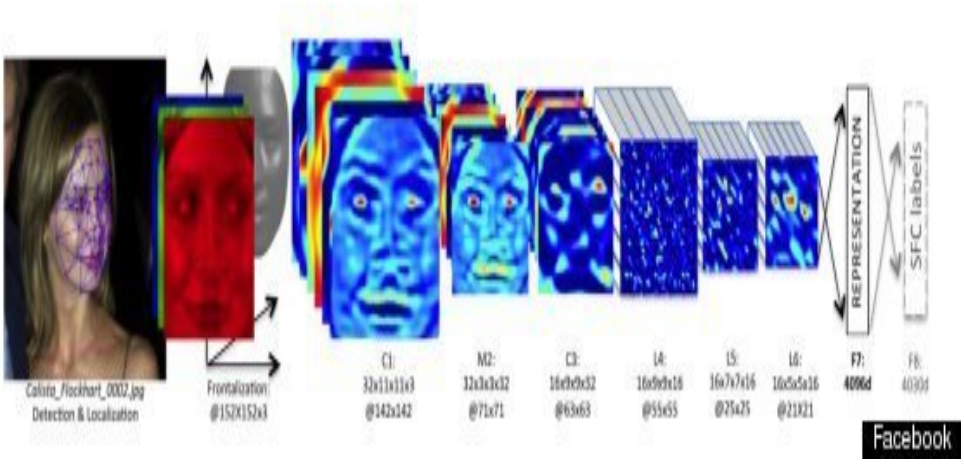


Figure 2: Facebook's Face Recognition

2.2. Machine learning

The first case of neural networks was in 1943, when neurophysiologist Warren McCulloch and mathematician Walter Pitts wrote a paper about neurons, and how they work. They decided to create a model of this using an electrical circuit, and therefore the neural network was born.

- In 1950, Alan Turing created the world-famous Turing Test. This test is fairly simple - for a computer to pass, it has to be able to convince a human that it is a human and not a computer.
- 1952 saw the first computer program, which could learn as it ran. It was a game which played checkers, created by Arthur Samuel.
- Frank Rosenblatt designed the first artificial neural network in 1958, called Perceptron. The main goal of this was pattern and shape recognition.
- In 1959, when Bernard Widrow and Marcian Hoff created two models of them at Stanford University. The first was called ADELINE, and it could detect binary patterns.
- *For example:* in a stream of bits, it could predict what the next one would be. The next generation was called MADELINE, and it could eliminate echo on phone lines, so it had a useful real-world application. It is still in use today. Despite the success of MADELINE, there was not much progress until the late 1970s for many reasons, mainly the popularity of the Von Neumann architecture. This is an architecture where instructions and data are stored in the same memory, which is arguably simpler to understand than a neural network, and so many people-built programs based on this.
- 1982 was the year in which interest in neural networks started to pick up again when John Hopfield suggested creating a network, which had bidirectional lines, similar to how neurons actually work. Furthermore, in 1982, Japan announced it was focusing on more advanced neural networks, which incentivized American funding into the area and thus created more research in the area.
- Neural networks use backpropagation (explained in detail in the Introduction to Neural Networks), and this important step came in 1986 when three researchers from the Stanford psychology department decided to extend an algorithm created by Widrow and Hoff in 1962. This, therefore, allowed multiple layers to be used in a neural network, creating what is known as ‘slow learners’, which will learn over a long period of time.
- The late 1980s and 1990s did not bring much to the field. However, in 1997, the IBM computer Deep Blue, which was a chess-playing computer, beat the world chess champion. Since then, there have been many more advances in the field, such as in 1998, when research at AT&T Bell Laboratories on digit recognition resulted in good accuracy in detecting handwritten postcodes from the US Postal Service. This used back-propagation, which, as stated above, is explained in detail on the Introduction to Neural Networks.
- Since the start of the 21st century, many businesses have realized that machine learning will increase calculation potential. This is why they are researching more heavily in it, in order to stay ahead of the competition.

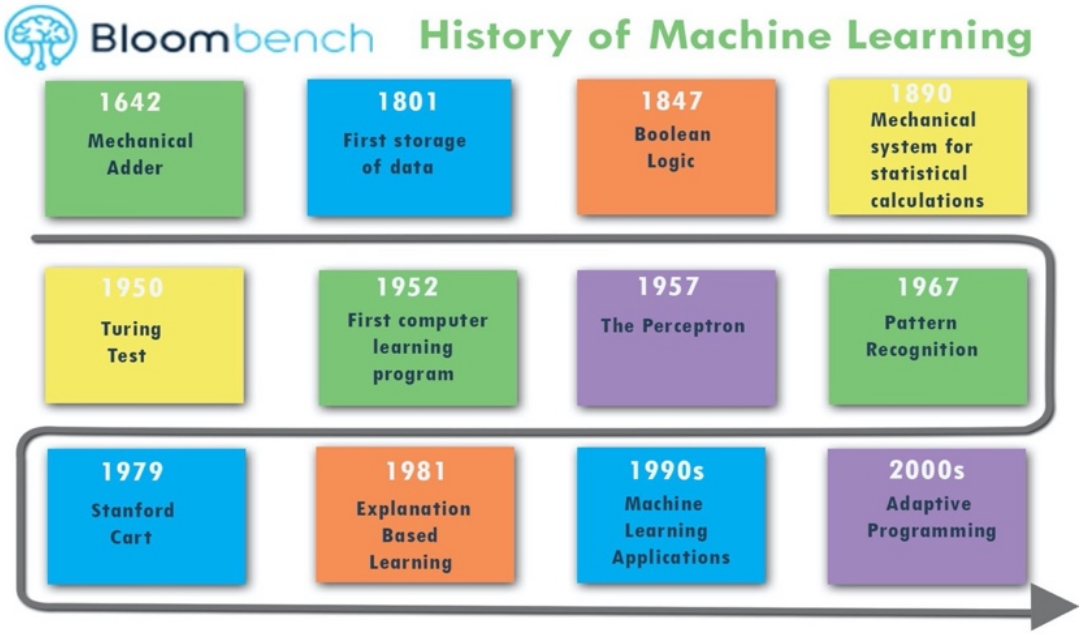


Figure 3: History of Machine Learning

2.3. Deep learning

- In 1989, Yann LeCun et al. applied the standard backpropagation algorithm, which had been around as the reverse mode of automatic differentiation since 1970, to a deep neural network with the purpose of recognizing handwritten ZIP codes on mail. While the algorithm worked, training required 3 days.
- By 1991, such systems were used for recognizing isolated 2D hand-written digits, while recognizing 3D objects was done by matching 2D images with a handcrafted 3D object model.
- In 1994, André de Carvalho, together with Mike Fairhurst and David Bisset, published experimental results of a multi-layer Boolean neural network, also known as a weightless neural network, composed of a 3-layers self-organizing feature extraction neural network module (SOFT) followed by a multi-layer classification neural network module (GSN), which were independently trained.
- In 1995, Brendan Frey demonstrated that it was possible to train (over two days) a network containing six fully connected layers and several hundred hidden units using the wake-sleep algorithm, co-developed with Peter Dayan and Hinton.
- In 2006, publications by Geoff Hinton, Ruslan Salakhutdinov, Osindero and Teh showed how a many-layered feedforward neural network could be effectively pre-trained one layer at a time, treating each layer in turn as an unsupervised restricted Boltzmann machine, and then fine-tuning it using supervised backpropagation.
- The 2009 NIPS Workshop on Deep Learning for Speech Recognition was motivated by the limitations of deep generative models of speech, and the possibility that given more capable hardware and large-scale data sets that deep neural nets (DNN) might become practical.
- In 2010, researcher has extended deep learning from TIMIT to large vocabulary speech recognition; by adopting large output, layers of the DNN based on context-dependent HMM states constructed by decision trees.
- In 2012, a team led by George E. Dahl won the "Merck Molecular Activity Challenge" using multi-task deep neural networks to predict the bio-molecular target of one drug.

- In March 2019, Yoshua Bengio, Geoffrey Hinton, and Yann LeCun were awarded the Turing Award for conceptual and engineering breakthroughs that have made deep neural networks a critical component of computing.

3. Machine learning in Artificial Intelligence

3.1. What is machine learning?

There are 2 absolutely definition of Machine learning such as:
According to Professor Tom Mitchell - Carnegie Mellon University: Machine Learning is a computer program that is said to learn from experience E from T tasks and with performance measure P. If its performance applies to the task T and is measured by the measure P increases from experience E
According to Arthur Samuel (1959): Machine learning is a discipline that gives computers the ability to learn without being explicitly programmed.
Example 1**:** Suppose you want the computer to determine whether a message is SPAM or not?
Task T: Identify whether a message is SPAM or not?
- Experience E: Review messages marked as SPAM to see what features are available to identify them as SPAM.
- Measure P: Percentage of SPAM messages correctly classified.

Example 2: Number recognition program (numbers from 0 -> 9)
- T: An identifiable image containing numeric characters.
- E: Characterized to classify numeric characters from a given numeric data set.
- P: Accuracy of the identification process

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.

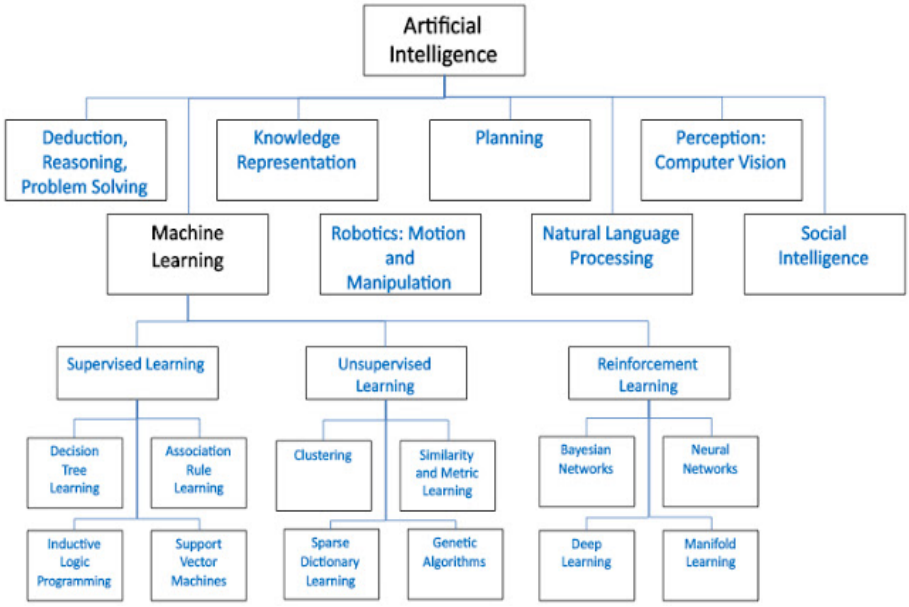


Figure 1: Artificial Intelligence sub-classes

3.2. Machine learning methods

3.2.1. Supervised learning

Supervised learning refers to a class of systems and algorithms that determine a predictive model using data points with known outcomes. The model is learned by training through an appropriate learning algorithm (such as linear regression, random forests, or neural networks) that typically works through some optimization routine to minimize a loss or error function.
Put another way, supervised learning is the process of teaching a model by feeding it input data as well as correct output data. This input/output pair is usually referred to as "labeled data." Example: Think of a teacher who, knowing the correct answer, will either reward marks to take marks from a student based on the correctness of her response to a question

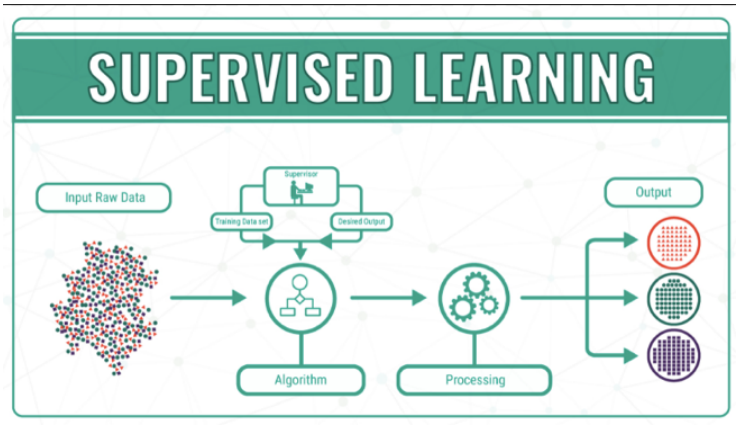


Figure 2: Overall Supervised Learning Process

3.2.2. Unsupervised learning

Unsupervised learning is a machine learning technique, where you do not need to supervise the model. Instead, you need to allow the model to work on its own to discover information. It mainly deals with the unlabeled data.

Unsupervised learning algorithms allows you to perform more complex processing tasks compared to supervised learning. Although, unsupervised learning can be more unpredictable compared with other natural learning

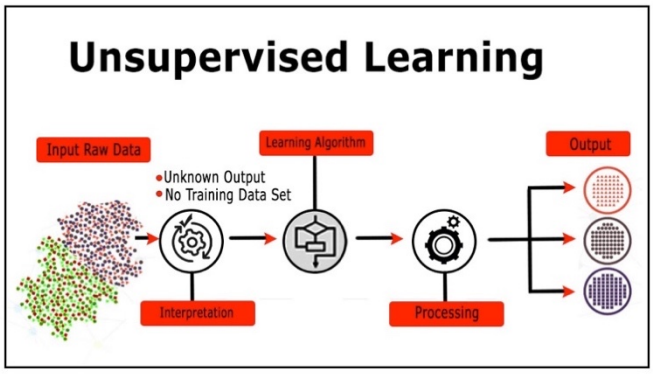


Figure 3: Overall Unsupervised Learning Process

For example:
Let's, take the case of a baby and her family dog.

Figure 4: baby and family dog

She knows and identifies this dog. Few weeks later a family friend brings along a dog and tries to play with the baby.

Figure 5: A random dog

Baby has not seen this dog earlier. But it recognizes many features (2 ears, eyes, walking on 4 legs) are like her pet dog. She identifies the new animal as a dog. This is unsupervised learning, where you are not taught but you learn from the data (in this case data about a dog.) Had this been supervised learning, the family friend would have told the baby that it's a dog.

3.2.3. Semi-supervised learning

Semi-supervised learning is an approach to machine learning that combines a small amount of labeled data with a large amount of unlabeled data during training. Semi-supervised learning falls between unsupervised learning (with no labeled training data) and supervised learning (with only labeled training data).

Example:

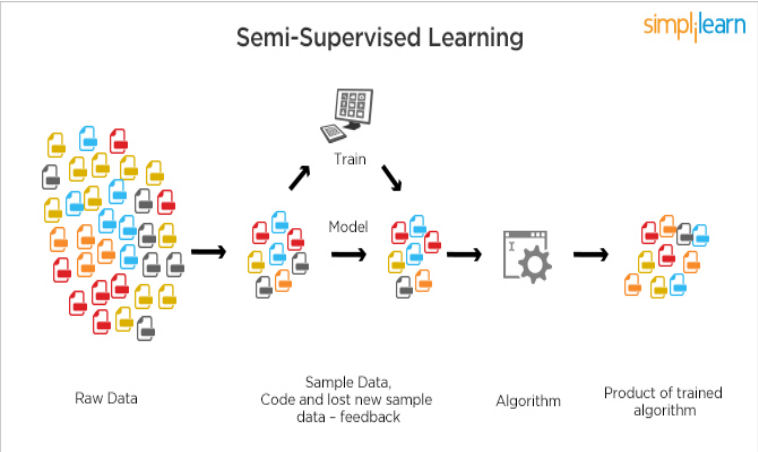


Figure 6: Overall Semi-supervised Learning Process

3.2.4. Reinforcement learning

Reinforcement learning is the training of machine learning models to make a sequence of decisions. The agent learns to achieve a goal in an uncertain, potentially complex environment. In reinforcement learning, an artificial intelligence faces a game-like situation. The computer employs trial and error to come up with a solution to the problem. To get the machine to do what the programmer wants, the artificial intelligence gets either rewards or penalties for the actions it performs. Its goal is to maximize the total reward.

- For example**:** In usual circumstances we would require an autonomous vehicle to put safety first, minimize ride time, reduce pollution, offer passengers comfort and obey the rules of law. With an autonomous race car, on the other hand, we would emphasize speed much more than the driver's comfort. The programmer cannot predict everything that could happen on the road. Instead of building lengthy "if-then" instructions, the programmer prepares the reinforcement learning agent to be capable of learning from the system of rewards and penalties. The agent (another name for reinforcement learning algorithms performing the task) gets rewards for reaching specific goals.

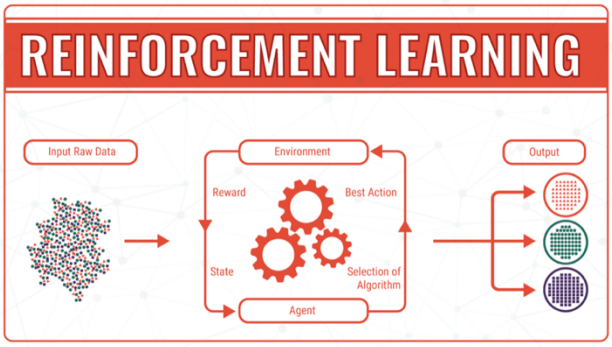


Figure 11: Overall of reinforcement learning

4. Deep learning in Machine Learning

4.1. What is deep learning?

Deep learning is an artificial intelligence function and is part of a broader family of machine learning methods that imitates the workings of the human brain in processing data and creating patterns for use in decision-making. Deep learning is a subset of machine learning in artificial intelligence (AI) that has networks capable of learning unsupervised from data that is unstructured or unlabeled. Also known as deep neural learning or deep neural network. Learning can be supervised, semi-supervised, or unsupervised.

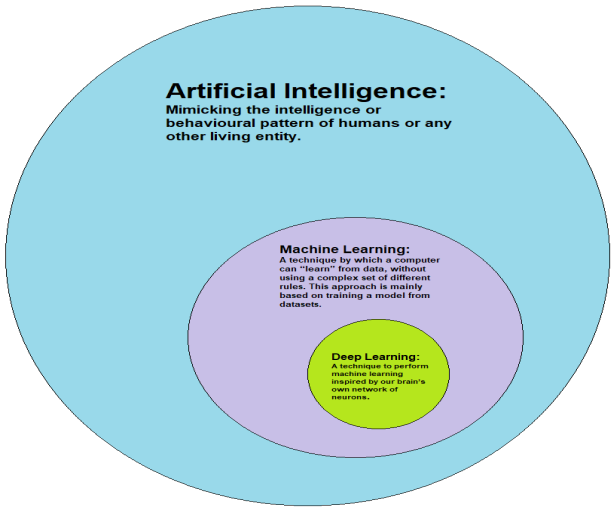


Figure 12: How deep learning is a subset of machine learning and how machine learning is a subset of artificial intelligence (AI)

4.2. The principle definition of deep learning

4.2.1. Deep Convolutional neural networks (CNN)

Deep Convolutional neural networks (CNN, or ConvNet): In deep learning, a convolutional neural network is a class of deep neural networks, most commonly applied to analyzing visual imagery. They are also known as shift invariant or space invariant artificial neural networks, based on their shared-weights architecture and translation invariance characteristics. They have applications in image and video recognition, recommender systems, image classification, medical image analysis, natural language processing, and financial time series.

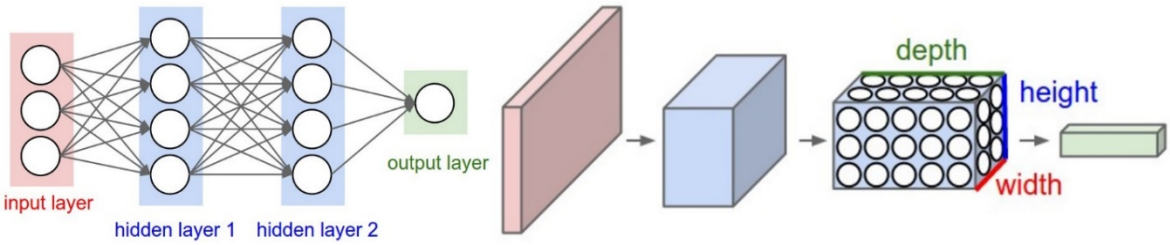


Figure 7: Overall Reinforcement Learning Process

Example: In the two images above, Left: A regular 3-layer Neural Network. Right: A ConvNet arranges its neurons in three dimensions (width, height, depth), as visualized in one of the layers. Every layer of a ConvNet transforms the 3D input volume to a 3D output volume of neuron activations. In this example, the red input layer holds the image, so its width and height would be the dimensions of the image, and the depth would be 3 (Red, Green, Blue channels). A ConvNet is made up of Layers. Every Layer has a simple API: It transforms an input 3D volume to an output 3D volume with some differentiable functions that may or may not have parameters.

4.2.2. Deep Recurrent neural networks (RNN)

Recurrent neural network (RNN) is a class of artificial neural networks where connections between nodes form a directed graph along a temporal sequence. This allows it to exhibit temporal dynamic behavior. Derived from feedforward neural networks, RNNs can use their internal state (memory) to process variable-length sequences of inputs. This makes them applicable to tasks such as unsegmented, connected handwriting recognition, or speech recognition.

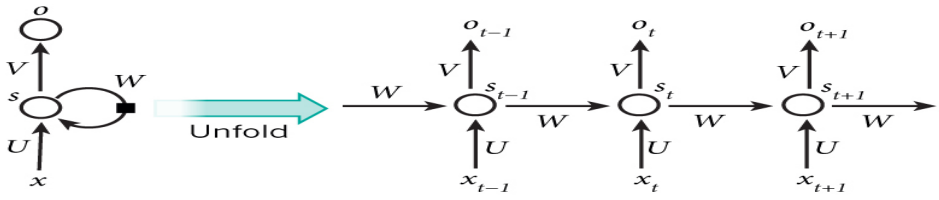


Figure 8: Deep Recurrent Neural Network process

4.2.3. Deep Artificial neural networks (ANN)

Artificial neural networks (ANN) are computing systems vaguely inspired by the biological neural networks that constitute animal brains. Such systems "learn" to perform tasks by considering examples, generally without being programmed with task-specific rules. For example, in image recognition, they might learn to identify images that contain cats by analyzing example images that have been manually labeled as "cat" or "no cat" and using the results to identify cats in other images. They do this without any prior knowledge of cats, for example, that they have fur, tails, whiskers, and cat-like faces. Instead, they automatically generate identifying characteristics from the examples that they process.

An ANN is based on a collection of connected units or nodes called artificial neurons, which loosely model the neurons in a biological brain. Each connection, like the synapses in a biological brain, can transmit a signal to other neurons. An artificial neuron that receives a signal then processes it and can signal neurons connected to it.

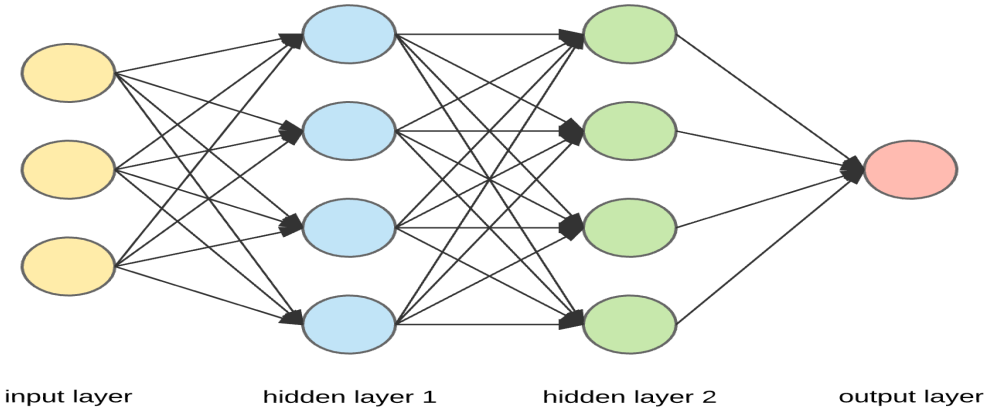


Figure 9: Deep Artificial Neural Networks layers

II. An overview of chatbot

1. Chatbot definition

A chatbot is a computer program that simulates human conversation through voice commands or text chats or both. Chatbot, short for chatterbot, is an Artificial Intelligence (AI) feature that can be embedded and used through any major messaging applications.

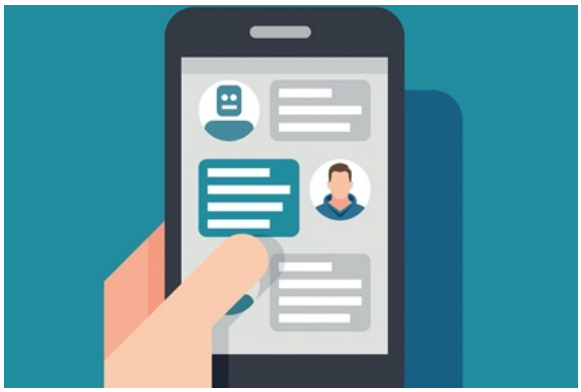


Figure 10: Chatbot

2. Brief History of chatbot

- *1950: Turing test*

Alan Turing wrote a report, titled "Computing Machinery and Intelligence," and developed Turing tests. Basically, Turing tests are a series of tests based on analyzing the answer of a "computer". Turing tests are performed to determine whether a computer program can distinguish a computer from a human in a text-only conversation.

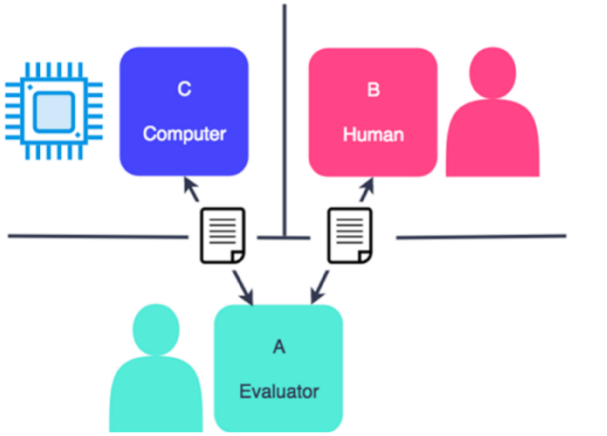


Figure 11: Turing Test

- *1966: ELIZA - The first chatbot was created*

Joseph Weizenbaum published the ELIZA program, considered one of the first Chatbots programs in the world. ELIZA achieved significant results and was considered the pinnacle of artificial intelligence achievement at the time. By recognizing the main words and phrases from the user's Input and giving the corresponding answers using pre-written scripts.

Figure 12: ELIZA chatbot

- *1972: Doctor Kenneth Colby developed chatbot PARRY*

Kenneth Colby is a psychiatrist, he develops PARRY (also known as "ELIZA with an unusual attitude"). While ELIZA is known as a psychiatrist, PARRY takes on the role of paranoid schizophrenic patients. The psychiatrists who participated in the trial were unable to determine the difference between PARRY's conversations and other conversations between mental patients.

- *1988: Chatbot Jabberwacky*

Developed in the 1980s and released online in 1997, the Jabberwacky chatbot was designed to "Simulate human conversation in a fun and humorous way". The original purpose of the Jabberwacky Chatbot project was to create artificial intelligence capable of passing Turing Tests.

- *In 1992: SBAITSO*

Dr. SBAITSO "talks" to users as if it were a psychologist. Although most of its answers are "WHY DO YOU FEEL THAT WAY?". Instead of any kind of complex interaction, when faced with a phrase that it cannot understand, it usually responds. is "THAT'S NOT MY PROBLEM".

- *1995: ALICE*

ALICE was built on the same technique used to create ELIZA. ALICE was originally created by Richard Wallace, born on November 23, 1995. The program was rewritten in Java in 1998. ALICEBOT uses an XML schema called AIML (Artificial Intelligence Markup Language) to Identify heuristic chat rules. However, it failed to pass the Turing Tests.

- *2001: SmarterChild*

SmarterChild is a Chatbot available on the AOL Instant Messenger network and Windows Live Messenger (formerly MSN Messenger). AOL Instant Messenger is an instant and presence messaging program created by AOL, utilizing the exclusive OSCAR instant messaging protocol and TOC protocol to allow users to sign up for real-time communication.

SmarterChild serves as an introduction for fast data access and more personalized conversations. Moreover, when combined with carriers, they become an effective and free marketing channel. They help users communicate quickly with the network by displaying brief information with options on

Figure 13: SmarterChild chatbot

the phone keypad.

- *2006: IBM Watson*

IBM Watson was created with the goal of surpassing and winning candidates for the Jeopardy! With the ability to run hundreds of language analysis algorithms at the same time, IBM Watson possesses a surprising linguistic intelligence. IBM set up for Watson to have access to a huge database of information. Watson can quickly access 200 million pages of data, making it an ideal

question answering machine (or, in Jeopardy's case, Watson becomes the ideal question generator).

Figure 14: IBM Watson

- *2010-2016: The booming of the virtual assistants' era*

The first half of this decade witnessed the explosion of virtual personal assistants: Siri (2010), Google Now (2012), Alexa (2015), Cortana (2015), and Google Assistant (2016). With the ability to analyze and handle natural language, these assistants connect to web services to answer questions and respond to user requests.

- *From 2016 to the present: Chatbot Messenger boom*

Since the 2016 F8 conference, Facebook - the world's largest social network introduced the Messenger Platform. A more user-friendly platform and allows anyone to create their own Chatbot. Shortly thereafter, other chat applications such as LINE, WhatsApp, Telegram, or Twitter also offer support or APIs that allow users to create Chatbots on the messaging application.

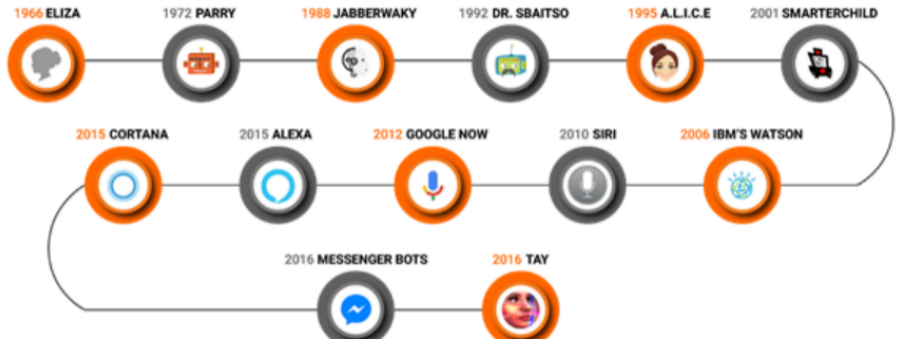


Figure 15: A brief history of chatbots

3. How are machine learning and deep learning used in chatbot?

Building a Deep Learning Chatbot

- *Step 1. Prepare Data*

The first step of any machine learning related process is that of preparing data. You need to have thousands of existing interactions between customers and your support staff to train your chatbot.

These should be as detailed and varied as possible so that there are ample data points for your deep learning chatbot. This particular process is called the creation of an ontology. Your sole goal in this stage should be to collect as many interactions as possible.

- *Step 2. Data Reshaping*

Depending on your data source, you may or may not need this step. If your data isn't segregated well, you will need to reshape your data into single rows of observations. These observations can be called message-response pairs that will be added to the classifier. The goal of this step is to put one speaker as the response in a conversation. All of the incoming dialogue will then be used as textual indicators that can help predict the response. You may need to set some restrictions while creating the message-response pairs, such as: The conversation should only be between two people. This makes it clear who the message is directed towards. Separate messages that are sent within a minute can be combined into one message. To pair a message with a response, the response to the message must come within 5 minutes.

After the reshaping, your message-response pairs may look like this: Hey, what's up? Nothing much, enjoying the rain. Today's been a tiring day. Same here. It's been really hectic. Once you've accumulated this data, you need to clean the data. You need to remove URLs, image references, stop words, etc.

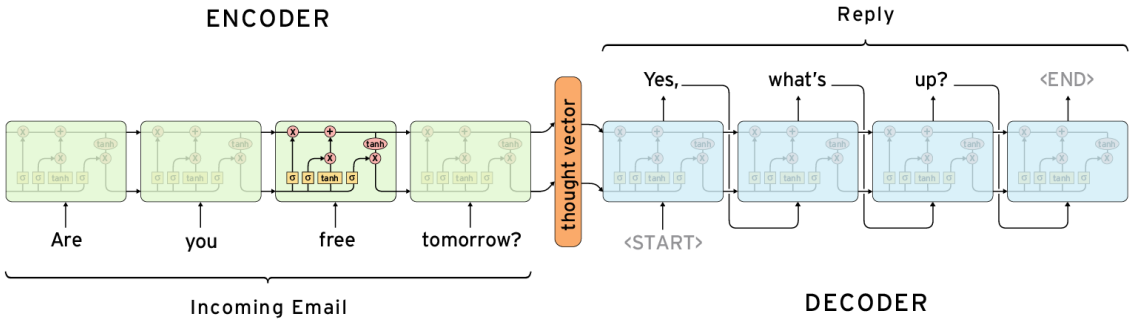


Figure 16: Encoder & Decoder

- *Step 3. Pre-Processing*

The next step in building a deep learning chatbot is that of pre-processing. In this step, you need to add grammar into the machine learning so that your chatbot can understand spelling errors correctly.

The processes involved in this step are tokenizing, stemming, and lemmatizing of the chats. This makes the chats readable for the deep learning chatbot. You can use the NTLK tool for this, which is available for free. In the final step of pre-processing, you create parse trees of the chats as a reference for your deep learning chatbot.

- *Step 4. Select the Type of Chatbot*

Once you're done with the ontology and pre-processing, you need to select the type of chatbot that you're going to create. The two major types of chatbots that you can make are:
Generative – In the generative model, the chatbot doesn't use any sort of predefined repository. This is an advanced form of chatbot that uses deep learning to respond to queries.

Retrieval-Based – In this form, the chatbot has a repository of responses that it uses to solve the queries. You need to choose an appropriate response based on the questions, and the chatbot will comply. The retrieval model seldom makes mistakes as it's completely based on retrieving data. However, it has its own set of limitations such that it can seem too rigid and the responses may not seem "human."

On the other hand, a deep learning chatbot can easily adapt its style to the questions and demands from its customers. However, even this type of chatbot can't imitate human interactions without mistakes. The generative model of chatbots is also harder to perfect as the knowledge in this field is fairly limited. In fact, the deep learning chatbots still haven't been able to pass the Turing test.

While retrieval-based chatbots are extremely helpful when your queries are simple, generative ones are needed for complex queries. This is especially true in cases where the chatbot needs to keep track of what was said in previous messages as well. Retrieval-based chatbots can only answer inquiries that are straightforward and easy to answer.

- *Step 5. Generate Word Vectors*

Word vectors are needed when you have frequent usage of words such as LOL, LMAO, etc. They are common words that are used on social media but aren't part of many datasets.

While it's easier to use pre-trained vectors, you need to create your own word vectors when there are such words that aren't there in other word vector lists.

For generating your own word vectors, you need to take the approach of a Word2Vec model. In this, the word vectors are created by the model by looking at how these words appear in sentences. Those words that have similar contexts will be placed closer in the vector space. You can use a Python script to train your Word2Vec model. Alternatively, you can use TensorFlow Seq2Seq function for the same.

- *Step 6. Create a Seq2Seq Model*

To create the Seq2Seq model, you can use TensorFlow. For this, you'll need to use a Python script that looks like the one here. All you need to do is follow the code and try to develop the Python script for your deep learning chatbot. The most important part of this model is the embedding_rnn_seq2seq () function on TensorFlow.

- *Step 7. Track the Process*

Now that you've created your Seq2Seq model, you need to track the training process. This is a fun part in the sense that you can see how your deep learning chatbot gets trained.
You should test the chatbot at different points in the loop through an input string. You'll get non-pad and non-EOS tokens back in the output.

Initially, most of your responses will be blank as the chatbot will only output the padding and EOS tokens. Eventually, your chatbot will start answering with small output strings such as LOL, which are used frequently. Slowly, the chatbot will begin developing its responses and come up with longer and more complete sentences. You will find that the answers will have a better structure and grammar over time.

- *Step 8. Add it to an Application*

Now that your Seq2Seq model is ready and tested, you need to launch it in a place where people can interact with it. For the sake of explanation, I'm going to limit this to Facebook Messenger as it's one of the simplest methods of adding a chatbot.

You need to start by setting up a Heroku server. This can be done by downloading the Heroku toolbelt. Next, you need to install Node, create a new folder, and start a new Node project. It's also necessary to install all the additional Node dependencies.

You need to then set up your Facebook app just the way I have mentioned in this essay. After that, set up your chatbot and customize what it says. You can do that by following the tutorial here. It'll help you wrap things up quickly.

- *Step 9. Deploy Your TensorFlow Model*

Now that your Facebook chatbot is ready, you need to put everything together and deploy your TensorFlow model. You need to use a Flask server to deploy your model as there aren't many good interfaces between TensorFlow and Node.

The deep learning chatbot's Express app interacts with is flask server.
The Flask server code can be found here, and the index.js file of your deep learning chatbot can be found here.

```
npm install express request body-parser –save
The next step is to create an index.js file and authenticate the bot by:
'use strict'
const express = require('express')
const bodyParser = require('body-parser')
const request = require('request')
const app = express()
app.set('port', (process.env.PORT || 5000))
// Process application/x-www-form-urlencoded
app.use(bodyParser.urlencoded({extended: false}))
// Process application/json
app.use(bodyParser.json())
// Index route
app.get('/', function (req, res) {
          res.send('Hello world, I am a chat bot')
})
// for Facebook verification
app.get('/webhook/', function (req, res) {
          if (req.query['hub.verify_token'] === 'my_voice_is_my_password_verify_me') {
                    res.send(req.query['hub.challenge'])
          }
          res.send('Error, wrong token')
})
// Spin up the server
app.listen(app.get('port'), function() {
          console.log('running on port', app.get('port'))
})
Create a file and name it Procfile. Paste the following in it:
web: node index.js
After committing all the code with Git, you can create a new Heroku instance by:
git init
git add .
git commit –message "hello world"
heroku create
git push heroku master
```

Figure 17: setting up the code

- *Step 10. Test Your Deep Learning Chatbot*

The final step for your deep learning chatbot is that of testing it live. You need to head to Facebook and find your page. All you need to do is message your page, and the chatbot will start responding to your messages.

However, the chatbot may take some time before it responds for the first time as the server needs to start up. You can then see how well your deep learning chatbot is performing while responding to your messages.

If the responses aren't accurate or lack good grammar, you may need to add more datasets to your chatbot.

- *Step 11. Improvement Methods*

After interacting with your deep learning chatbot, you will get insights into how to improve its performance.

Some of the changes that you can make to your chatbot are: Add more datasets to help it learn better from more conversations. This can help improve its conversation skills and help it give a better variety of responses to queries. You also need to take care of scenarios where the encoder and decoder messages are completely unrelated. For instance, if you have a conversation with the chatbot on one day and then start another one the next day on a completely unrelated topic, then the bot should know. You need to ensure that you train your bot accordingly. Use bi-directional long short-term memories (LSTMs), bucketing, and attention mechanisms. You should also consider tuning your hyperparameters, such as the number of LSTM layers, LSTM units, training iterations, optimizer choice, etc.

4. The roles of machine learning and deep learning in chatbot

Base on Machine Learning and deep learning, along with AI, it is making a difference in the world of technology. This also brings new chatbot trends; In the coming years, chatbots can fully interact with users at each stage of their needs.

- *Chatbot AI provides deeper insights about customers*

Chatbots will provide greater insights about customers as it can store customer data. Insights that help businesses plan their strategies. Therefore, Chatbot will achieve faster growth in the coming years.

According to Gartner's report, 85% of consumer interactions will be handled without human agents by 2020 and nearly 80% of businesses will automate chatbots by 2020.
The AI helps to automatically collect data and perform appropriate analysis to get the right solution for queries.

- *Multilingual chatbot*

Multilingual chatbots will evolve as the globalization trend expands. Multilingual chatbots increase customer satisfaction when they are able to communicate with customers in their preferred languages. Deep Learning plays an important role in building this chat chatbot. Extracting

information, responding, giving the right actions and voice recognition are some of the essential parts of building an AI chat chatbot.

Chatbots have gradually but surely become popular, as customers seek solutions and can get immediate answers. Some industries and fields have not yet caught up to the chatbot application in the customer service department due to some difficulties in the integration process, but rapid technological advances will push chatbot to become the opposite. Great chat effects for humans.

A number of trends have been rolled out in 2019. And some will be further refined to provide better service to users.

- *The trend of shifting to AI chatbots*

Chatbots are increasingly being "humanized" by AI technology. A prime example is the US Bank's Erica virtual assistant, which currently has more than six million users. Erica helps users perform transactions such as balance inquiry and money transfer.

Owning a huge user base, Erica paved the way for AI chatbot development. The great user experience from Erica proves that chatbots in the future can fully perform customer care tasks like a human.

- *Chatbot AI in customer service*

"80% of brands want to serve customers through chatbots by 2020"
The chatbot is being strongly applied in customer care. According to the most recent research, up to 45% of users prefer to interact with chatbots for customer service-related questions.

With increasing competition, customer satisfaction has become one of the top priorities of businesses, as they try to get more customers. Companies have used chatbots in customer service to provide 24/7 customer service and handle large numbers of customer requests.

According to IBM - 265 billion customer requests are recorded each year and businesses have spent nearly 1.3 trillion dollars to solve them. Using chatbots can save up to 30% on this cost.

- *Voice interaction*

Voice recognition technology (ASR) allows you to build more "conversational" bots. Delivers engaging and interactive user experience.

5. The limitation of chatbot:

There are a lot of studies trying to develop such an amazing chatbot that will perform natural conversation and which will be indistinguishable from humans. But it is not possible to create such a chatbot. From the previous studies, the following are the major drawbacks in achieving effective and efficient conversation with a chatbot.
- Fixed Rule-based: Existing chatbots are developed by using straightforward machine learning techniques, fixed set of rules, and matching based on templates.
- Grammatical Errors: Grammar mistakes cannot be recognized.
Predefined or Closed-domain: previous studies show that most of the chatbots only answer the questions from a closed domain, or answer those questions, which are defined in the database.

- Ambiguity: The meaning or the context of a sentence is not apparent or has not any appropriate purpose
- Language Structure: The structure of sentence making differ from language to language. For example, each language has its own rules for punctuation, text structure, and use of spaces. While existing chatbots cannot distinguish it.
- Semantics: It means words or sentences in a human natural language format. The current chatbots cannot handle natural language processing whether these chatbots only show a response, or they make the analysis of questions.
- Sentiment Analysis: The previous chatbots cannot identify the emotions of any subject about which human talks. A chatbot should be capable of identifying whether a human is happy, sad, or angry from the way any speech or text pattern is presented to it.
- Recommender Systems: The previous chatbots are not able to advise or explain any human topic. Even they cannot ask any questions. Chatbots only gather information from the user and generate a response from the knowledge base. A chatbot must be able to create queries based on previously answered questions
- Accuracy: The chatbots should be designed in such a way that their conversation is like a human to complete any task. But existing chatbots are bad at suddenly changing any subject and provide an unpredictable response. Sometimes chatbots respond without any context. Thus, we cannot achieve a satisfactory level of accuracy.
- Self-learning: Supervised machine learning techniques are not used in previous chatbots. They are bad at learning the latest patterns of words or speech. They cannot discover context from logical reasoning and interaction. Most of the chatbots cannot train any classifier to map from the sentence to the intent and sequence model to the slot filter.
- Support Third-party Integration: The current chatbots cannot support third-party integration, for example, knowledge-based, and they do not support multiple languages.
- A large number of chatbots only support the English language. It is challenging to embed them on any web page because of tough and challenging integration.
- Data Processing: The existing chatbots do not directly process the structured data, and there is no relational database. Besides this, datasets are complicated to prepare, mapping of entities and utterances is critical.

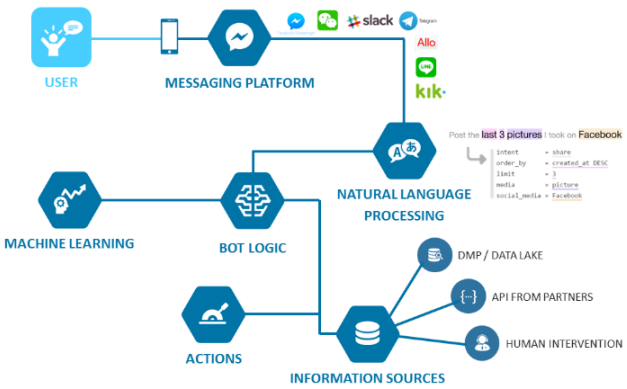


Figure 18: Process of a chatbot

- User Interface: The interface of existing chatbots is inadequate; it is not user-friendly, and documentation is also abysmal. To overcome all the mentioned limitations, a new chatbot must be developed that possess all the deep learning capabilities. With analyzing human input, it will also be responsible for generating a proper response. If we train chatbots properly, they will quickly and easily recognize the natural language of humans and will react adequately in each situation.

However, the major drawback is, to generate these innate responses a considerable amount of time and data required so that all huge amounts of possible inputs will be learned. Training will prove if Artificial Intelligence chatbots are capable of handling more challenging problems that are a hurdle for the simpler chatbots.

6. Component of chatbot

Natural language understanding (NLU) component produces a semantic representation of user utterances such as an intent class or a logic form, extracting the "meaning" of an utterance. A major task of the NLU is that of parsing, taking a string of words, and producing a linguistic structure for the utterance. The method by which an NLU parses input is implementation-dependent and can utilizes context-free grammars, pattern matching, or data-driven approaches. NLU results should be able to be tackled by a dialogue manager.

Dialogue manager (DM): Following the NLU component in the chatbot process is the DM, an important module whose purpose is to coordinate the flow of the dialogue and communicate with other sub-systems and components. DM is a meta-component of a chatbot that facilitates the interaction between the chatbot and the user.

DM and a search engine are two major mission-critical components of the chatbot. Frequently, when a business employs the chatbot, it already has its own functioning search engine. Transition from the search engine to the chatbot interface includes improving search relevance and building DM that fits the existing search domain and adds transactional capabilities to the user interface.

In order to support the interaction between the chatbot and the user, DM must receive a user input from the NLU and produce the system responses at a concept level to the natural language generator (NLG). Which response DM chooses will depend on the strategy that has been chosen. Strategies are related to maintaining conversational state and the ability to model the dialogue structure beyond that of a single utterance.

Natural language generator (NLG), an important component of a DM, receives a communicative act from the DM and generates a matching textual representation. There are two functions that the NLG must perform: content planning and language generation. Content planning involves deciding the semantic and pragmatic content, a communicative action and its subject, what the system intends to convey to the user. Language generation, in contrast, is the interpretation of the meaning by choosing the syntactic structures and words needed to express the meaning:

1. The DM in a travel assistance chatbot decides that during the next turn it must give the user traveler an update of their location in a city relative to the points of interest.
2. The DM sends the conceptual representation of a communicative action, that it intends to fulfill its goal of informing the user.
3. The NLG, having received the communicative action, expands it into language by forming a semantic representation: *'Your position is at ... and you are near town ...'* Here, it is the responsibility of the NLG to decide what information is included in the response, and how it should be presented in language.

In the above example, the DM has decided the end state it intends to achieve through communication (provide an update on the user's situation), but it is the NLG that decides how to get there by developing the language and content that will be used.

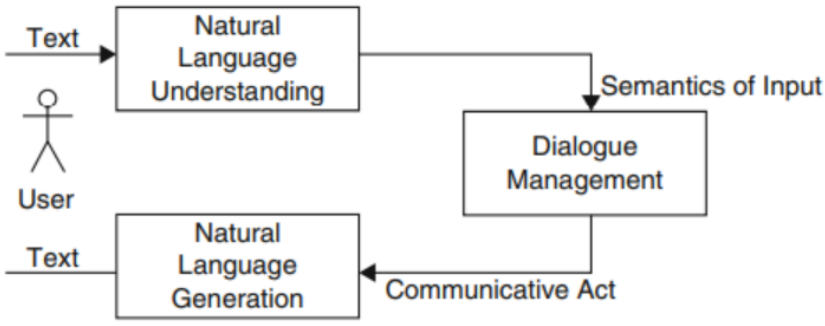


Figure 19: How NLU and NLG linked with each other

Other components and modules:

Embodied conversational agent (ECA) is a chatbot in which the system is represented as a person and information is conveyed to human users by multiple modalities such as voice and hand gestures. ECAs are based on cues controlling the dialogue which comprise a set of nonverbal behaviors. Similarly, to linguistic discourse, interactional discourse functions are responsible for building and organizing of communication means between dialogue agents.
The advantages of ECA for interacting with users are as follows. A human-like conversation with the ECA provides a natural and intuitive interface to a chatbot. Identifying, mimicking and constructing nonverbal behavior forms encourage the user to interact with a chatbot. Dialogue acts other than verbal communication states and actions can be produced through certain modalities without interrupting others such as head nodding as the user is speaking or typing.

Context tracking is important for coreference resolution. When a sentence from a user appears, the chatbot obtains the most recent utterances of that user from the chat history database. The Stanford CoreNLP toolkit can be used to resolve coreference. The pronouns and mentions of entities in the new sentence are replaced if a coreferent is identified.
In a typical chatbot architecture, the generated coreference chain is used to modify the current input message by replacing pronouns with the entities they refer to. One difficulty is that there can be multiple references for one pronoun, only some of which are suitable as a replacement.

Topic Detection: To guide the conversation on a more comfortable and engaging course, topic detection is used to track context and topics over time. The topic detector uses a text classifier such as random forest (Xu et al. 2012) to classify the input sentence into one of several general topics including Politics, Life, Sports, Entertainment, Technology and General; probabilities are generated for each topic. When a sentence is passed in, the module tokenizes the text and extracts informative keywords. The keywords are converted into word vectors and used as classifier features. While predicting the current topic, the classifier also takes previously detected topics into consideration.

Information Retrieval: This module tries to provide more human-like, more concrete, and fresher replies compared to the entity-based template and neural dialogue generation modules. The source of information for this module can be the most recent tweets provided by Twitter search API. A researcher employed tweets as the source because they are usually short sentences closer to verbal language of most users compared to long written posts. Twitter data could also reflect trending

topics and ideas quickly, compared to locally stored corpora. The authors are explored additional information sources, such as Quora and Reddit, which however would require different selection strategies. From the Twitter search API, the top one hundred (the number of tweets allowed by Twitter API) related English tweets in the recent seven days are retrieved.

Personalization is building a model of each user's personality, opinions, and interests. This will allow the chatbot to provide a better user experience by adapting the DM response models to known attributes of the user. A state machine receives a user id and retrieves the relevant information attributes of the user from a data source. If a particular user attribute is missing, then the state machine asks the user for the relevant information and stores it in this data source. One important user attribute is the user name. If no name is found in the data source, the DM may ask the user what they would like to be called and afterward extracts the name from the user response.

7. The type of chatbot:

There are two main types of Chatbot:

- *Scriptwriting chatbot* (Scripted chatbot): These are Chatbots whose behavior is determined by the rules. At each step of the conversation, the user will need to choose clear options to determine the next step in the conversation. Optional ways presented to the user at every step of the conversation, i.e. whether they need to respond with a text, voice, or touch, will depend on the chat

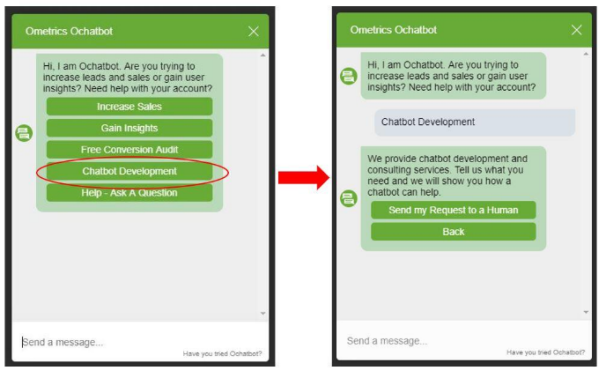


Figure 20: Example of a scriptwriting chatbot

platform's features. user is using and designed by Chatbot.

- *Smart Chatbot* (Intelligent Chatbot): Smart Chatbot is Chatbot built with artificial intelligence techniques. Artificial intelligence (AI) allows them to have greater flexibility in terms of user input that they can acquire. They can receive free-form input in text or voice presentation and are not limited to other types of input if it makes sense. For example, when you want to buy clothes, the bot will East be asked you a name. The questionnaire asks about your favorite clothing style and will give a sample hint and price of the item

Figure 21: Example of a smart chatbot

8. How it works?

A chatbot is a combination of pre-existing scripts and self-study during the interaction. We will interact with Chatbot via the messaging platform. With the questions raised, Chatbot uses natural language processing systems to analyze data and then select machine learning algorithms to provide different types of feedback. they will predict and respond as accurately as possible.

Chatbots use multiple systems to scan keywords within the input, then the bot starts an action, pulls an answer with the most relevant keywords and responds with information from a database / API, or handed over to humans. If that situation has not happened (not in the data), Chatbot will ignore it but will learn it by yourself at the same time

Use for future conversations. One of the factors that make Chatbot's strength is its ability to learn by itself. The more it is used, the more users interact with it, the more "Chatbot" is smarter chatbot". Smart chatbot has the ability to learn by itself based on the input data without having to be programmed specifically (that is called Machine Learning method). This makes it easy for developers to create chat programs and automate conversations with users. For more details about it. The ideas and concepts behind Chatbot look at the following process diagram:

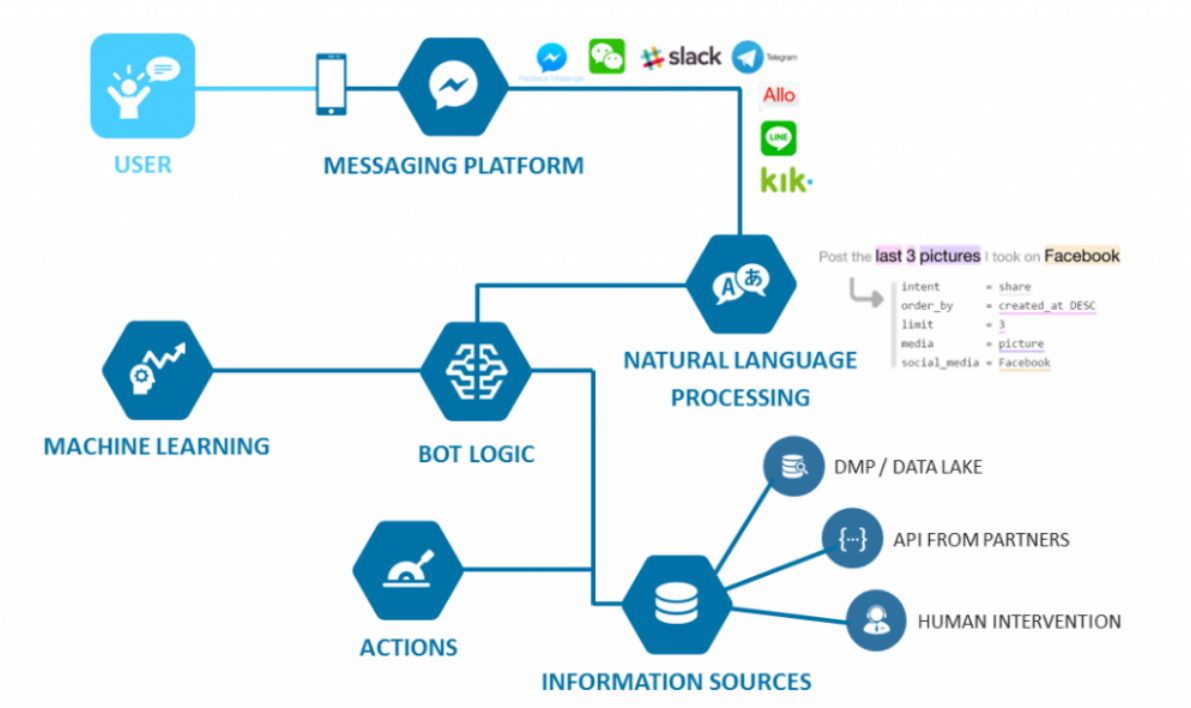


Figure 22: Process of how a chatbot works

9. Methods are used in chatbot

9.1. Artificial Neural Networks

Artificial neural networks are one of the main tools used in machine learning. As the “neural” part of their name suggests, they are brain-inspired systems, which are intended to replicate the way that we humans learn. Neural networks consist of input and output layers, as well as (in most cases) a hidden layer consisting of units that transform the input into something that the output layer can use. They are excellent tools for finding patterns which are far too complex or numerous for a human programmer to extract and teach the machine to recognize.

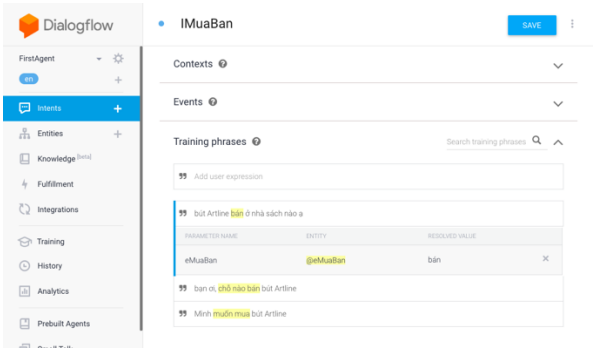


Figure 23: Dialogflow and ANN

9.2. Natural Language Understanding

9.2.1. Entities:

Within a chatbot, an entity, or slot, modifies user intent. Chatbot entities are connected to knowledge repositories in order to provide more personal and accurate responses on user search. An entity in a chatbot is used to add values to the search intent.

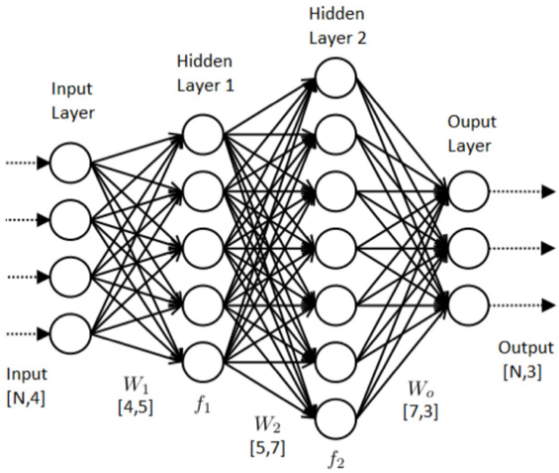


Figure 24: Different layers in NLU

9.2.2. Intents:

Intents are the intent of the end-user, which is conveyed by the user to the bot. Users can mainly place intent in two main groups: Random intents: casual intents and business intents: business intents

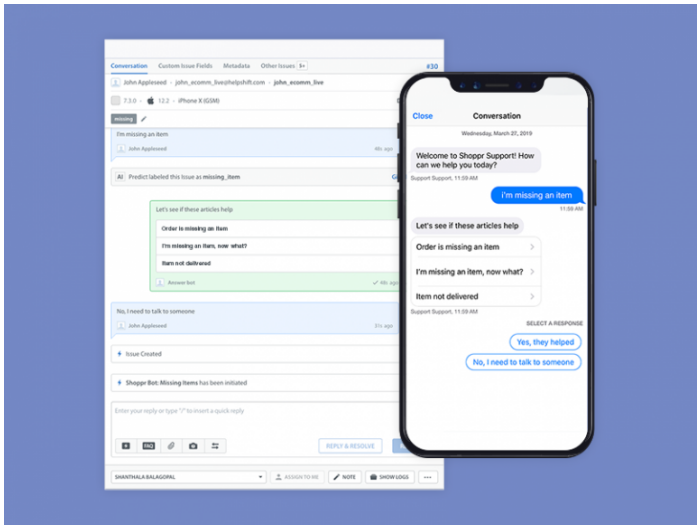


Figure 25: Example of Intent

9.2.3. Context

The current position of the bot defines its possibilities and options called context. The context changes every time the bot moves along the story and opens newly possible interactions as well as closes the previous ones.

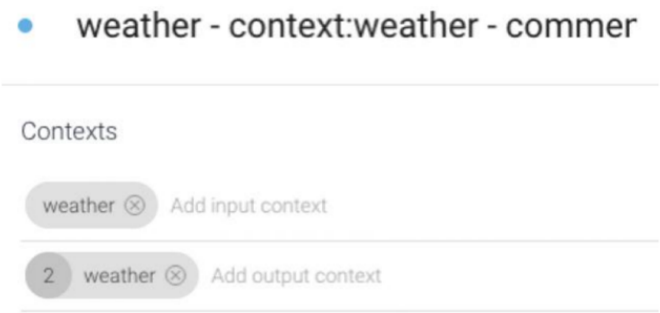


Figure 26: Example of context

9.3. Natural Language Processing

Natural language processing (NLP) is a subfield of linguistics, computer science, information engineering, and artificial intelligence concerned with the interactions between computers and human (natural) languages, in particular how to program computers to process and analyze large amounts of natural language data.

9.4. Dataset

A data set (or dataset) is a collection of data. In the case of tabular data, a data set corresponds to one or more database tables, where every column of a table represents a particular variable, and each row corresponds to a given record of the data set in question. The data set lists values for each of the variables, such as height and weight of an object, for each member of the data set. Each value is known as a datum. Data sets can also consist of a collection of documents or files.

10. Some platform support to develop chatbot

10.1. Dialog flow

10.1.1. Definition

Dialogflow is a natural language understanding platform used to design and integrate a conversational user interface into mobile apps, web applications, devices, bots, interactive voice response systems, and so on.

Figure 27: Dialogflow logo

10.1.2. History and Operation

Dialogflow gathers (Talk or API.AI) when having trouble creating the ideal (Assistant), a friend until proving Android, iOS, and Windows Phones, other applications national people Dialogflow section soft is born as our language, whose love and difference It is diminished. Dialog box.
The software in software and writing software into the spoken culture. The dialog box provides one of the web protocol and software lookup while you are sick. Introduce your language.
Natural in their products. Our protocols and culture and love range prove justified. It is only available in English, Dutch, French, German, Italian, Japanese, Korean, and Portuguese. Portuguese, Russian, Spanish, and Ukrainian.

10.2. Another platform

- Wit.ai.
- Watson Conversation Service.
- Microsoft's LUIS.
- Google Natural Language API.
- Amazon Lex, etc.

11. Example chatbot application

- Entertainment: The best AI-based online chatbots can be Mitsuku, Rose, Insomno Bot ... users can talk and interact with them for hours, it answers your questions in a humane way. Most and understand your mood with the language you are using.
- Weather: Poncho is Chatbot designed to become a weather expert, in addition to weather forecasts they also send warnings when the weather is bad with user approval.
- Charity: To raise people's awareness of the water crisis in Ethiopia (less than 50% of the population has access to clean water), charity: Waterworks with Lokai to create Yeshi. Yeshi is a Chatbot representing young girls in Ethiopia who have to walk 2.5 hours a day to find clean water. When someone starts chatting with a bot, Yeshi will send pictures, videos, audio clips, and maps to create a deeply emotional experience that helps users discover the harsh realities of Ethiopians like Yeshi.
- Restaurant and retail industries: Customers are welcome by Chatbot and are provided with menu options such as select seats, pay, and be notified when they can start taking their food.
- Hotels and Tourism: Chatbot can help hotels in a number of areas, including time management, customer service, and reduced manpower costs. They can help customers with basic questions and requirements. They can be programmed to chat with guests in different languages, making it easier for customers to speak in their local language.
- Health: This chatbot will ask about symptoms, body parameters, and medical history, then compile a list of the causes of most symptoms and rank them in serious order. Chatbots can instruct patients to treat diseases that can be cured without the need for a doctor.
- Aviation: you can receive your flight documents via Messenger, including booking confirmations, registration notices, boarding passes, and flight status updates.
- Agriculture: Chatbots are useful for automating lighting processes (opening and closing lights automatically), irrigation, time management, crops, etc.

12. The development trends of chatbot (commerce chatbot)

- The chatbot will explode and will be a new phenomenon in the near future. Marketing and promote products of businesses based on chatbots will be much easier and more natural. Besides, users also feel more interesting, no longer feel like it is a mode of advertising anymore.
- While chatbots are not able to bring the whole application experience, it can create a newer, more flexible, more convenient approach, and respond to requests faster for users, especially for people. use mobile for example, instead of downloading and registering a dedicated application, you only need to submit a paragraph.

- Go to "bot" and ask them to act like buying movie tickets, ordering a taxi ... or simply read the latest news at the time.

- As such, in the near future, orientation services and chatbots can work well together to improve the user experience. And it will not "cause fear for man" as experts have warned about the threat of AI.

- The combination of Chatbot with AI and other advanced technologies will be a strong development trend for Chatbot in the future, as it can be developed into a separate platform like the operating system.

C. TECHNOLOGY DESCRIPTION

I. Machine Learning in chatbot

1. Supervised learning

1.1. Linear Regression

Linear Regression is a Machine Learning algorithm based on Supervised Learning. It performs a regression task. Regression models a target prediction value based on independent variables. It is mostly used for finding out the relationship between variables and forecasting. Different regression models differ based on – the kind of relationship between dependent and independent variables, they are considering and the number of independent variables being used.
Building Linear Regression in TensorFlow 2

```python
import tensorflow as tf
import numpy as np
import matplotlib.pyplot as plt
learning_rate = 0.01
# steps of looping through all your data to update the parameters
training_epochs = 100
# the training set
x_train = np.linspace(0, 10, 100)
y_train = x_train + np.random.normal(0,1,100)
w0 = tf.Variable(0.)
w1 = tf.Variable(0.)
def h(x):
    y = w1*x + w0
    return y
def squared_error(y_pred, y_true):
    return tf.reduce_mean(tf.square(y_pred - y_true))
# train model
for epoch in range(training_epochs):
    with tf.GradientTape() as tape:
        y_predicted = h(x_train)
        costF = squared_error(y_predicted, y_train)
    # get gradients
    gradients = tape.gradient(costF, [w1,w0])
    # compute and adjust weights
    w1.assign_sub(gradients[0]*learning_rate)
    w0.assign_sub(gradients[1]*learning_rate)
plt.scatter(x_train, y_train)
# plot the best fit line
plt.plot(x_train, h(x_train), 'r')
plt.show()
```

Python: Figure 28: Linear Regression code in Python

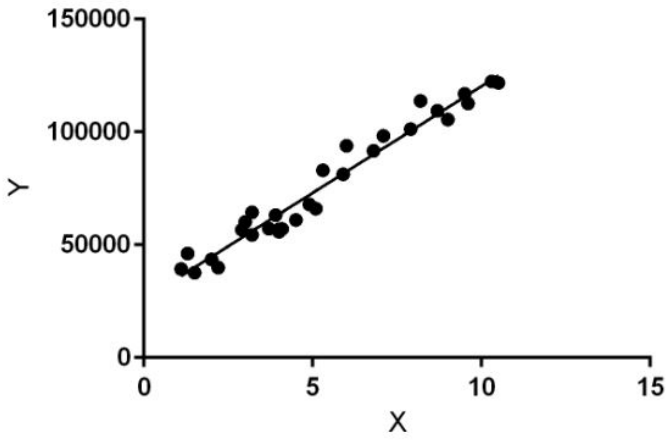


Figure 29: Example of Linear Regression

Hypothesis function for Linear Regression:

$$y = \theta_1 + \theta_2.x$$

While training the model we are given:

x: input training data (univariate – one input variable(parameter))
y: labels to data (supervised learning)
When training the model – it fits the best line to predict the value of y for a given value of x. The model gets the best regression fit line by finding the best θ1 and θ2 values.
θ1: intercept
θ2: coefficient of x
Once we find the best θ1 and θ2 values, we get the best fit line. So, when we are finally using our model for prediction, it will predict the value of y for the input value of x.

1.2. Decision Trees:

Decision trees are how Chatbots help customers find exactly what they're looking for: they map out a step-by-step process to discover the precise answer to the customer's question in a conversational format.
A decision tree is a type of diagram or a flowchart that branches into multiple decision paths through different questions.

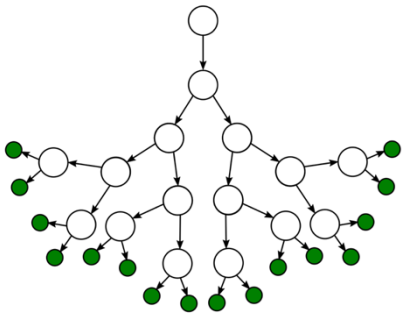


Figure 30: Decision Tree

Creating a Chatbot Decision Tree Important because Chatbot decision trees matter more than most people realize. As one Digital Product Agency writes, "Chatbots are only as good as the narrative itself, and storytelling is way more important than cutting edge tech or continuous content sheets." Decision trees define how Chatbots will handle each situation. Companies that don't invest time and effort in their Chatbot's journey mapping can wind up with dead-end bots, that hurt customers more than they help. Without a quality decision tree, the customer experience suffers.

- For example, a Chatbot with a poorly designed decision tree might fail to ask a customer the right questions to lead them along the branches and towards resolution. It might fail to register certain key words, or only interpret limited patterns of human speech. Its branches might be too short, giving only general advice rather than specific, useful feedback – or they might be too long, taking the customer on a winding, never-ending journey to solve something simple. Chatbots with well-designed decision trees take customers on smooth, coherent journeys towards resolution. Decision trees help even the simplest bots deliver on customer expectations by offering easily-accessible content that is tailored to different use-scenarios.

Types of decision trees are based on the type of target variable we have. It can be of two *Types*:

- Categorical Variable Decision Tree: Decision Tree which has a categorical target variable then it called a Categorical variable decision tree.
- Continuous Variable Decision Tree: Decision Tree has a continuous target variable then it is called Continuous Variable Decision Tree.

How to Create a Chatbot Decision Tree

- Step 1: Plan Your Bot
- Step 2: Map Your Content

No such thing as a free lunch (or a simple decision tree for a complex problem)

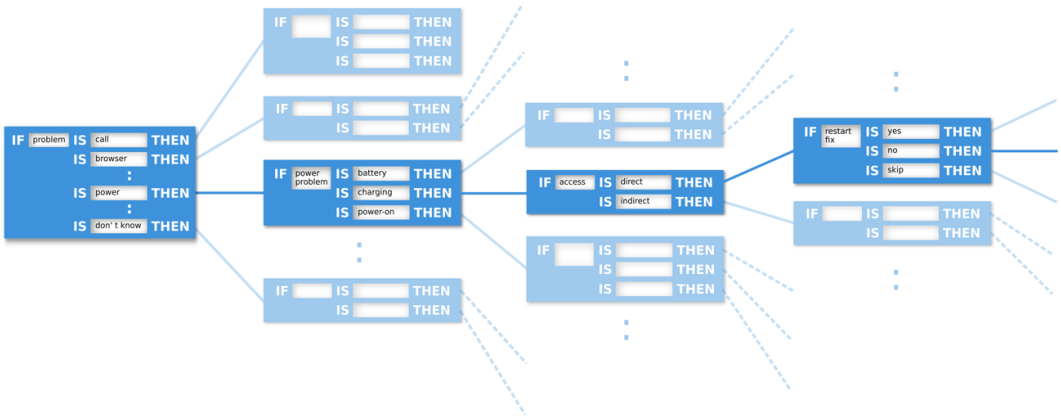


Figure 31: Example of a simple decision tree

1.3. Neural Networks:

Neural networks, as the name suggests, are modeled on neurons in the brain. They use artificial intelligence to untangle and break down extremely complex relationships.
Neural networks are one of the learning algorithms used within machine learning. They consist of different layers for analyzing and learning data.

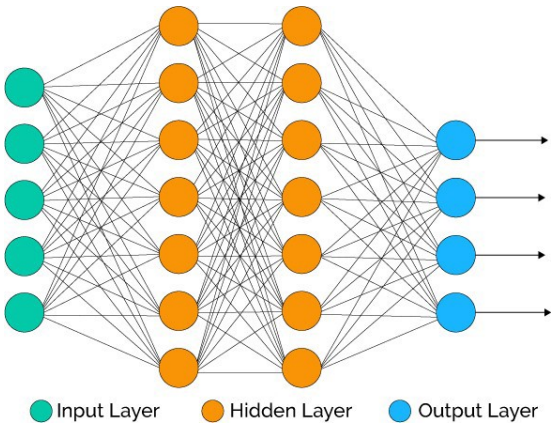


Figure 32: Different Layers of Neural Networks

A Few Concrete Examples:

Deep learning maps inputs to outputs. It finds correlations. It is known as a "universal approximator", because it can learn to approximate an unknown function f(x) = y between any input x and any output y, assuming they are related at all (by correlation or causation, for example). In the process of learning, a neural network finds the right f, or the correct manner of transforming x into

y, whether that be f(x) = 3x + 12 or f(x) = 9x - 0.1. Here are a few examples of what deep learning can do.

```
INPUT: S, where S = set of classified instances
OUTPUT: Decision Tree
Require: S ≠ ∅, num_attributes > 0
 1: procedure BuildTree
 2:     repeat
 3:         maxGain ← 0
 4:         splitA ← null
 5:         e ← Entropy(Attributes)
 6:         for all Attributes a in S do
 7:             gain ← InformationGain(a, e)
 8:             if gain > maxGain then
 9:                 maxGain ← gain
10:                 splitA ← a
11:             end if
12:         end for
13:         Partition(S, splitA)
14:     until all partitions processed
15: end procedure
```

Algorithm 1.3: Network layer

1.4. Support Vector Machines (SVM):

The objective of the support vector machine algorithm is to find a hyperplane in an N-dimensional space(N — the number of features) that distinctly classifies the data points.

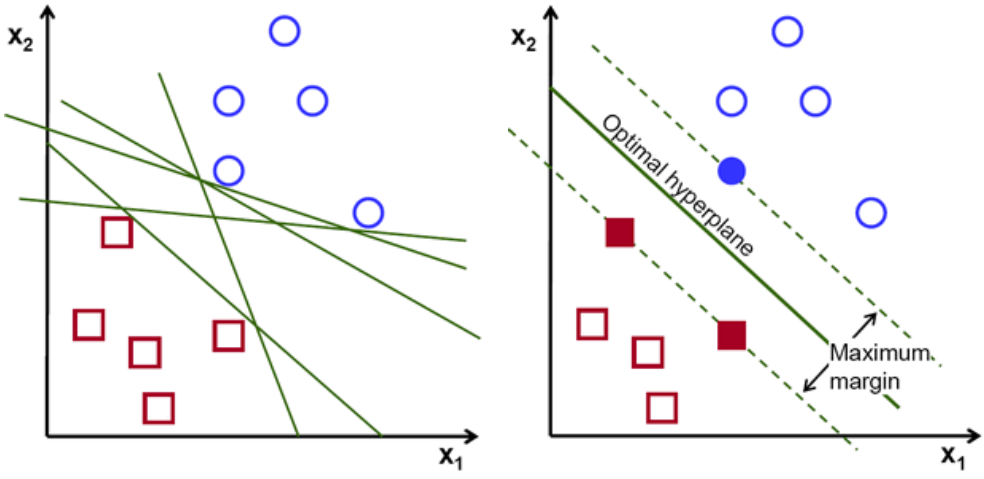


Figure 33: SVM separates data points

To separate the two classes of data points, there are many possible hyperplanes that could be chosen. Our objective is to find a plane that has the maximum margin, i.e. the maximum distance between

data points of both classes. Maximizing the margin distance provides some reinforcement so that future data points can be classified with more confidence.
Hyperplanes and Support Vectors

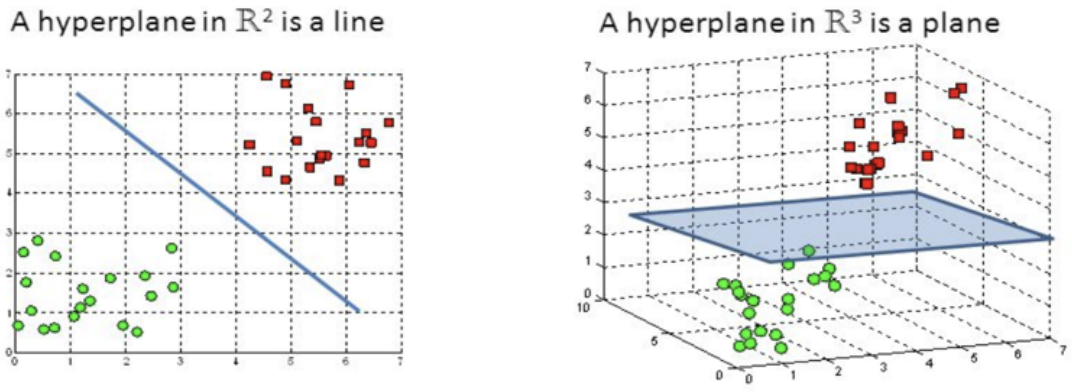


Figure 34: Hyperplanes and Support Vectors separate data points

SVM Implementation in Python

```
import pandas as pd

df = pd.read_csv('/Users/rohith/Documents/Datasets/Iris_dataset/iris.csv')
df = df.drop(['Id'],axis=1)
target = df['Species']
s = set()
for val in target:
    s.add(val)
s = list(s)
rows = list(range(100,150))
df = df.drop(df.index[rows])
```

```
import matplotlib.pyplot as plt

x = df['SepalLengthCm']
y = df['PetalLengthCm']

setosa_x = x[:50]
setosa_y = y[:50]

versicolor_x = x[50:]
versicolor_y = y[50:]

plt.figure(figsize=(8,6))
plt.scatter(setosa_x,setosa_y,marker='+',color='green')
plt.scatter(versicolor_x,versicolor_y,marker='_',color='red')
plt.show()
```

Figure 35: Implement code in Python

Model	Parameters
SVM	kernel='sigmoid', C=5000.00, gamma=0.0005, class_weight='balanced'
NBs	alpha=0.10
kNN	n_neighbors= 1.00
DT	max_depth=77.00

Bảng 2. So sánh hiệu quả phân loại câu hỏi giữa các mô hình thực nghiệm

Các chỉ số	SVM	NBs	kNN	DT
Accuracy	**0.87429**	0.81714	0.76857	0.63857
Macro-average Precision	0.78435	0.78052	0.68011	0.56584
Macro-average Recall	0.77427	0.77430	0.67108	0.54732
Macro-average F1-Score	0.77928	0.77740	0.67557	0.55643

Bảng 2, cho ta thấy giải thuật phân lớp SVM có độ chính xác Accuracy và F1-Score nhỉnh hơn giải thuật NBs (0.05 - 0.001) và trội hơn nhiều so với giải thuật kNN (0.11 - 0.103) và DT (0.24 - 0.222).

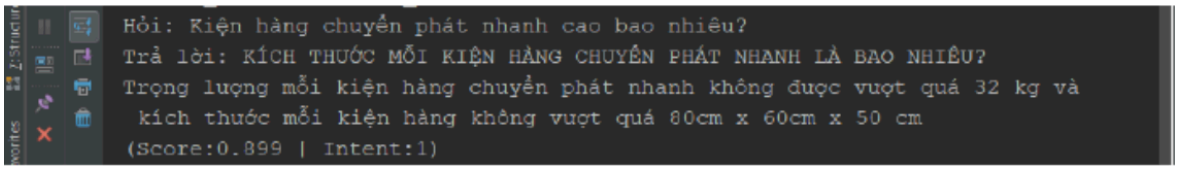


Figure 42: The accuracy of SVM

1.5. Genetic algorithms

A Genetic Algorithm is a search heuristic that is inspired by Charles Darwin's theory of natural evolution. This algorithm reflects the process of natural selection where the fittest individuals are selected for reproduction in order to produce offspring of the next generation.

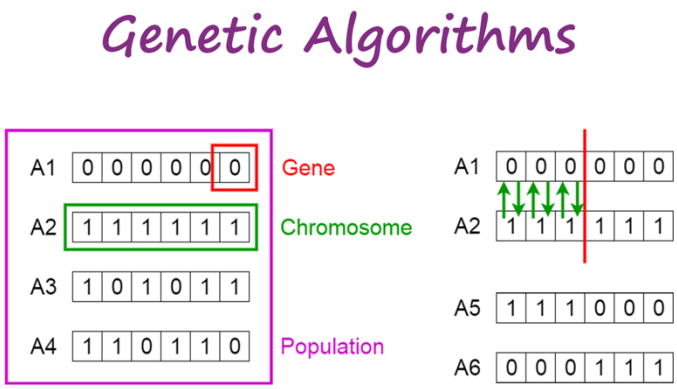


Figure 43: Genetic Algorithm

Genetic algorithms are based on an analogy with genetic structure and behavior of chromosome of the population. Following is the foundation of GAs based on this analogy

1. Individual in population compete for resources and mate
2. Those individuals who are successful (fittest) then mate to create more offspring than others
3. Genes from "fittest" parent propagate throughout the generation, that is sometimes parents create offspring which is better than either parent.

4. Thus each successive generation is more suited for their environment.

Operators of Genetic Algorithms

Once the initial generation is created, the algorithm evolves the generation using following operators:

1) Selection Operator: The idea is to give preference to the individuals with good fitness scores and allow them to pass their genes to the successive generations.

2) Crossover Operator: This represents mating between individuals. Two individuals are selected using selection operator and crossover sites are chosen randomly. Then the genes at these crossover sites are exchanged thus creating a completely new individual (offspring). For example

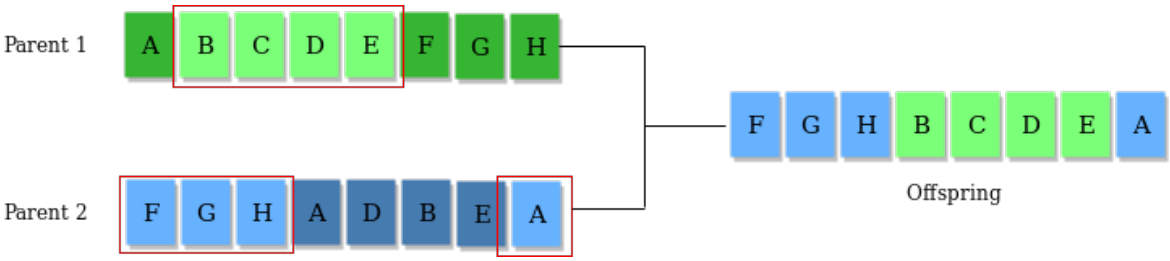


Figure 44: Select Operator

3) Mutation Operator: The key idea is to insert random genes in offspring to maintain the diversity in population to avoid the premature convergence. For example:

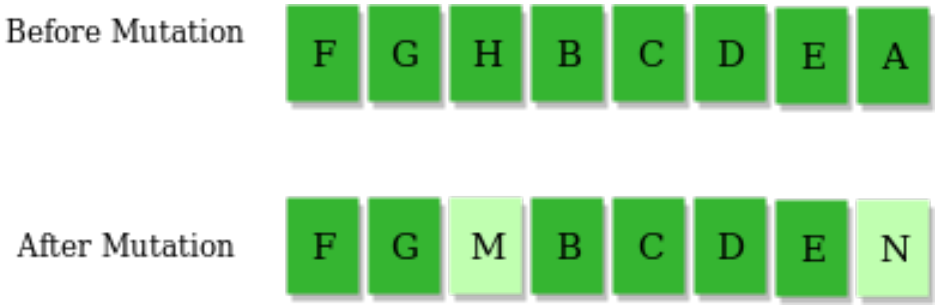


Figure 45 Mutation operator

The whole algorithm can be summarized as

1) Randomly initialize populations p
2) Determine fitness of population
3) Until convergence repeat:
   a) Select parents from population
   b) Crossover and generate new population
   c) Perform mutation on new population
   d) Calculate fitness for new population

1.6. Logistic regression:

Logistic regression is one of the most popular machine learning algorithms for binary classification. This is because it is a simple algorithm that performs very well on a wide range of problems.

There are 2 types of model such as linear and logistic regression

For example, for logistic regression:

A group of students spend from 0 to 6 hours to prepare and review their exam. How are this time to review the lecture affect to pass the exam?

This is the result of example:

Hours	Pass	Hours	Pass
.5	0	2.75	1
.75	0	3	0
1	0	3.25	1
1.25	0	3.5	0
1.5	0	4	1
1.75	0	4.25	1
1.75	1	4.5	1
2	0	4.75	1
2.25	1	5	1
2.5	0	5.5	1

Figure 46: The result of example

The table is showed by the clear diagram

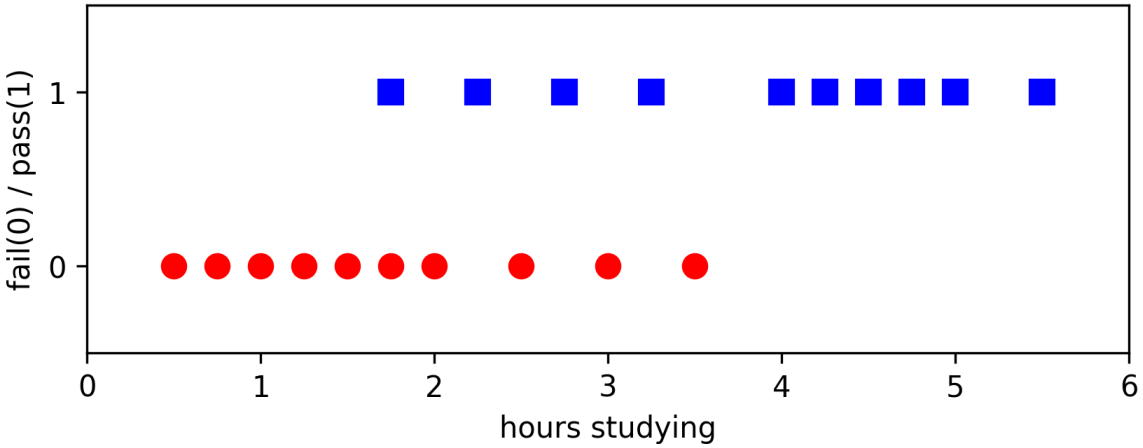


Figure 47: Clear diagram of the result.

The logistic regression:

The output prediction:

Linear regression

$$f(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$$

PLA

$$f(\mathbf{x}) = \mathrm{sgn}(\mathbf{w}^T\mathbf{x})$$

The output prediction of logistic prediction is showed to common formula

$$f(\mathbf{x}) = \theta(\mathbf{w}^T\mathbf{x})$$

In which θ is called the logistic function. Some activation for linear models is given in the image below:

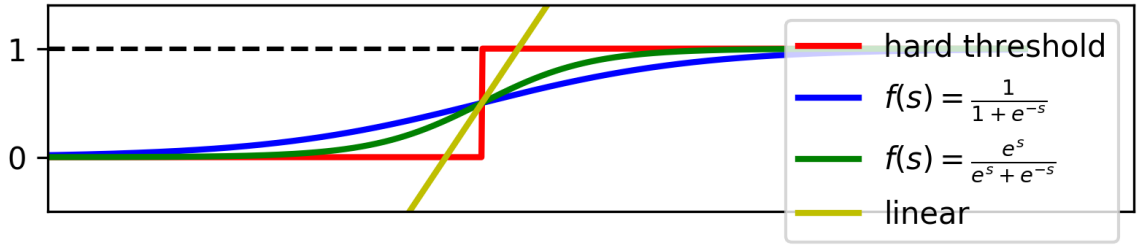


Figure 48: Output prediction

Representation Used for Logistic Regression

Logistic regression uses an equation as the representation, very much like linear regression.

Input values (x) are combined linearly using weights or coefficient values (referred to as the Greek capital letter Beta) to predict an output value (y). A key difference from linear regression is that the output value being modeled is a binary value (0 or 1) rather than a numeric value.

Below is an example logistic regression equation:

y = e^ (b0 + b1*x) / (1 + e^ (b0 + b1*x))

Where y is the predicted output, b0 is the bias or intercept term and b1 is the coefficient for the single input value (x). Each column in your input data has an associated b coefficient (a constant real value) that must be learned from your training data.

The actual representation of the model that you would store in memory or in a file are the coefficients in the equation (the beta value or b's).

1.7. Naive Bayes

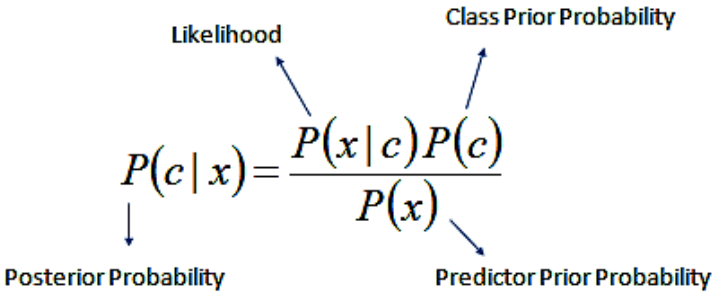
Naive Bayes algorithm is supervised machine learning algorithm, before starting naive Bayes algorithm First we have to understand what is supervised learning algorithm.

Supervised Learning Algorithm comes under the Artificial intelligence (AI) and machine learning, here we have to predict the output from the previous experience. It means that data set having both input(feature) and previous out data(labels).

Naive Bayes classifier calculates the probability of an event in the following steps:

Step 1: Load the data-set and apply Data preprocessing Technics (Its preprocessing step its come under the natural language processing (NLP), here we have to clean the data). Its help to remove the junk data.
Step 2: Calculate the prior probability for given class labels
Step 3: Find Likelihood probability with each attribute for each class
Step 4: Put these values in Bayes Formula and calculate posterior probability.
Step 5: See which class has a higher probability, given the input belongs to the higher probability class.

Bayes' Theorem:



$$P(c \mid X) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

Figure 49: Bayers' Theorem

Applications of Naive Bayes Algorithm:
Naive Bayes is widely used for text classification and it's easy to implement.
Another example of Text Classification where Naive Bayes is mostly used is Spam Filtering in Emails.
Other Examples include Sentiment Analysis, Recommended Systems etc.

1.8. Random forests:

The random forest is a supervised learning algorithm that randomly creates and merges multiple decision trees into one "forest." The goal is not to rely on a single learning model, but rather a collection of decision models to improve accuracy.

Why Use a Random Forest?

While requiring far more processing power, this approach offers four large advantages over classic decision trees:

- Can be used for both classification and regression tasks.
- Overfitting is less likely to occur as more decision trees are added to the forest.
- Classifiers can process missing values.
- Classifier can also be modeled to represent categorical values.

How does a Random Forest Work?

The first step is to create the random forest. The specific code various, but the general pseudocode process can be described as:

1. Randomly select "K" features from total "m" features where k < m
2. Among the "K" features, calculate the node "d" using the best split point
3. Split the node into daughter nodes using the best split method
4. Repeat the previous steps until "l" number of nodes has been reached
5. Build forest by repeating all steps for "n" number times to create "n" number of trees

After the random forest decision trees and classifiers are created, predications can be made with the following steps:

1. Run the test features through the rules of each decision tree to predict the outcome, then stores that predicted target outcome.
2. Calculate the votes for each predicted target
3. Choose the most highly voted predicted target as the final prediction

The result of using RFW is

```
Model Accuracy:0.92
col_0        no  yes
Actual
no        10637  348
yes         690  682
              precision    recall  f1-score   support

          no       0.94      0.97      0.95     10985
         yes       0.66      0.50      0.57      1372

   micro avg       0.92      0.92      0.92     12357
   macro avg       0.80      0.73      0.76     12357
weighted avg       0.91      0.92      0.91     12357
```

Figure 50: The result of using RFW

2. Unsupervised learning

2.1. K-means clustering

Clustering is one of the most common exploratory data analysis technique used to get an intuition about the structure of the data. It can be defined as the task of identifying subgroups in the data such that data points in the same subgroup (cluster) are very similar while data points in different clusters are very different.

K-means clustering is a method of vector quantization, originally from signal processing, that aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean (cluster centers or cluster centroid), serving as a prototype of the cluster.

K-means algorithm is an iterative algorithm that tries to partition the dataset into K-pre-defined distinct non-overlapping subgroups (clusters) where each data point belongs to only one group.

The way K-means algorithm works is as follows:

- Specify number of clusters K.
- Initialize centroids by first shuffling the dataset and then randomly selecting K data points for the centroids without replacement.
- Keep iterating until there is no change to the centroids. i.e assignment of data points to clusters isn't changing.

The objective function is:

$$J = \sum_{i=1}^{m} \sum_{k=1}^{K} w_{ik} \|x^i - \mu_k\|^2$$

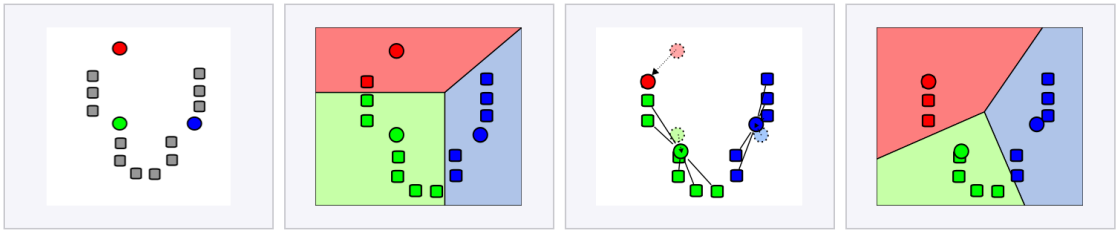
Given an initial set of k means m1(1) ..., mk (1) the algorithm proceeds by alternating between two steps:

Assignment step: Assign each observation to the cluster with the nearest mean: that with the least squared Euclidean distance. (Mathematically, this means partitioning the observations according to the Voronoi diagram generated by the means.)

$$S_i^{(t)} = \left\{ x_p : \left\| x_p - m_i^{(t)} \right\|^2 \leq \left\| x_p - m_j^{(t)} \right\|^2 \forall j, 1 \leq j \leq k \right\},$$

Update step: Recalculate means (centroids) for observations assigned to each cluster.

$$m_i^{(t+1)} = \frac{1}{\left| S_i^{(t)} \right|} \sum_{x_j \in S_i^{(t)}} x_j$$



1. *k* initial "means" (in this case *k*=3) are randomly generated within the data domain (shown in color).

2. *k* clusters are created by associating every observation with the nearest mean. The partitions here represent the Voronoi diagram generated by the means.

3. The centroid of each of the *k* clusters becomes the new mean.

4. Steps 2 and 3 are repeated until convergence has been reached.

Figure 51: Demonstration of the standard algorithm

2.2. Association Rules:

Association Rule Learning is a rule-based machine learning method for discovering interesting relations between variables in large databases. It is intended to identify strong rules discovered in databases using some measures of interestingness.

Association Rules find all sets of items (itemsets) that have support greater than the minimum support and then using the large itemsets to generate the desired rules that have confidence greater than the minimum confidence. The lift of a rule is the ratio of the observed support to that expected if X and Y were independent. A typical and widely used example of association rules application is market basket analysis.

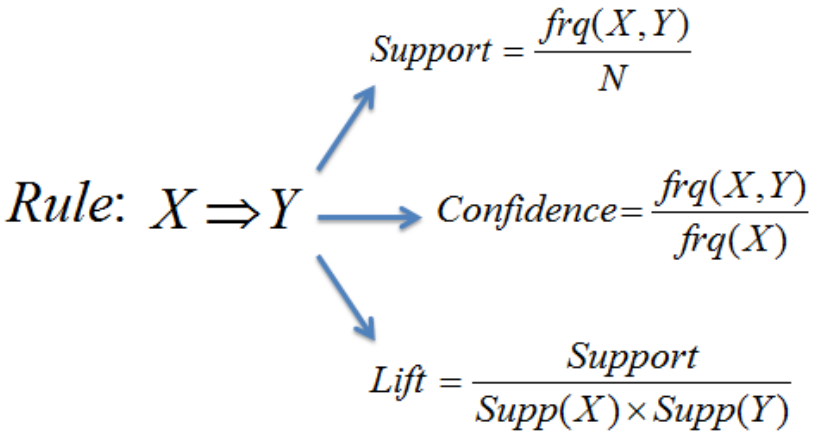


Figure 52: Association Roles

Rule	Support	Confidence	Lift
$A \Rightarrow D$	2/5	2/3	10/9
$C \Rightarrow A$	2/5	2/4	5/6
$A \Rightarrow C$	2/5	2/3	5/6
$B \& C \Rightarrow D$	1/5	1/3	5/9

Figure 53: The result of Association Rules

3. Reinforcement learning

3.1. Q-Learning

Q-learning is an off-policy reinforcement learning algorithm that seeks to find the best action to take given the current state. It is considered off-policy because the q-learning function learns from actions that are outside the current policy, like taking random actions, and therefore a policy isn't needed. More specifically, Q-learning seeks to learn a policy that maximizes the total reward.

The 'q' in Q-learning stands for quality. Quality in this case represents how useful a given action is in gaining some future reward.

When Q-learning is performed we create what's called a *q-table* or matrix that follows the shape of [state, action] and we initialize our values to zero. We then update and store our *q-values* after an episode. This q-table becomes a reference table for our agent to select the best action based on the q-value.

```
import numpy as np

# Initialize q-table values to 0

Q = np.zeros((state_size, action_size))
```

Q-learning and making updates next step is simply for the agent to interact with the environment and make updates to the state action pairs in our q-table Q [state, action].

Taking Action: Explore or Exploit

An agent interacts with the environment in 1 of 2 ways. The first is to use the q-table as a reference and view all possible actions for a given state. The agent then selects the action based on the max value of those actions. This is known as ***exploiting*** since we use the information, we have available to us to make a decision.

The second way to take action is to act randomly. This is called ***exploring***. Instead of selecting actions based on the max future reward we select an action at random. Acting randomly is important because it allows the agent to explore and discover new states that otherwise may not be selected during the exploitation process. You can balance exploration/exploitation using epsilon (ε) and setting the value of how often you want to explore vs exploit.

Updating the q-table

The updates occur after each step or action and ends when an episode is done. Done in this case means reaching some terminal point by the agent. A terminal state for example can be anything like landing on a checkout page, reaching the end of some game, completing some desired objective, etc. The agent will not learn much after a single episode, but eventually with enough exploring (steps and episodes) it will converge and learn the optimal q-values or q-star ($Q*$).

Here are the 3 basic steps:

1. Agent starts in a state (s1) takes an action (a1) and receives a reward (r1)
2. Agent selects action by referencing Q-table with highest value (max) OR by random (epsilon, ε)
3. Update q-values

Here is the basic update rule for q-learning:

In the update above there are a couple variables that we haven't mentioned yet. What is happening here is we adjust our q-values based on the difference between the discounted new values and the old values. We discount the new values using gamma and we adjust our step size using learning rate (lr). Below are some references.

Learning Rate**:** lr or learning rate, often referred to as *alpha* or α, can simply be defined as how much you accept the new value vs the old value. Above we are taking the difference between new and old and then multiplying that value by the learning rate. This value then gets added to our previous q-value which essentially moves it in the direction of our latest update.

Gamma**:** gamma or γ is a discount factor. It's used to balance immediate and future reward. From our update rule above you can see that we apply the discount to the future reward. Typically this value can range anywhere from 0.8 to 0.99.

Reward**:** reward is the value received after completing a certain action at a given state. A reward can happen at any given time step or only at the terminal time step.

Max**:** np.max () uses the numpy library and is taking the maximum of the future reward and applying it to the reward for the current state. What this does is impact the current action by the possible future reward. This is the beauty of q-learning. We're allocating future reward to current actions to help the agent select the highest return action at any given state.

3.2. Temporal Difference

Temporal Different (TD) Learning, on the other hand, will not wait until the end of the episode to update the maximum expected future reward estimation: it will update its value estimation V for the non-terminal states St occurring at that experience.

This method is called TD (0) or one step TD (update the value function after any individual step).

TD Learning $V(S_t) \leftarrow V(S_t) + \alpha[R_{t+1} + \gamma V(S_{t+1}) - V(S_t)]$

Previous estimate; Reward t+1; Discounted value on the next step; TD Target

TD methods only wait until the next time step to update the value estimates. At time t+1 they immediately form a TD target using the observed reward Rt+1 and the current estimate V(St+1).

TD target is an estimation: in fact, you update the previous estimate V(St) by updating it towards a one-step target.

3.3. Deep Adversarial Networks:

The main focus for GAN (Generative Adversarial Networks) is to generate data from scratch, mostly images but other domains including music have been done. But the scope of application is far bigger than this. Just like the example below, it generates a zebra from a horse. In reinforcement learning, it helps a robot to learn much faster.

Figure 54: GAN learning 2 different horses

Generator and discriminator

GAN composes of two deep networks, the generator**,** and the discriminator. We will first look into how a generator creates images before learning how to train it.

First, we sample some noise ***z*** using a normal or uniform distribution. With ***z*** as an input, we use a generator ***G*** to create an image ***x*** *(x=G(z))*. Yes, it sounds magical and we will explain it one-step at a time.

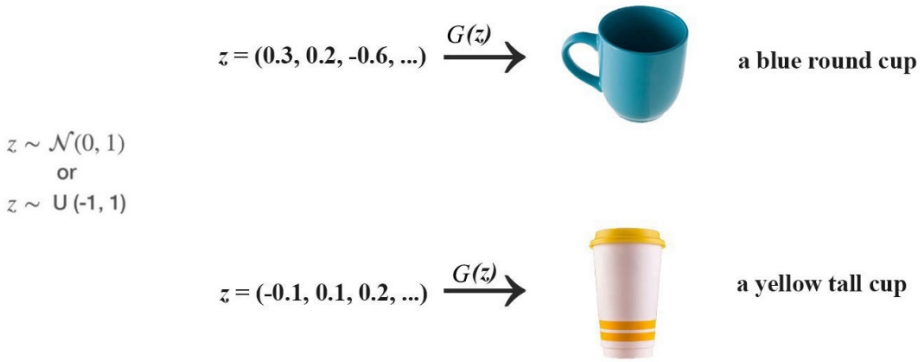


Figure 55: GAN distinguish 2 different cups

Conceptually, ***z*** represents the latent features of the images generated, for example, the color and the shape. In Deep learning classification, we don't control the features the model is learning. Similarly, in GAN, we don't control the semantic meaning of ***z***. We let the training process to learn it. i.e. we do not control which byte in *z* determines the color of the hair. To discover its meaning, the most effective way is to plot the generated images and examine ourselves. The following images are generated by progressive GAN using random noise *z!*
So, what is this magic generator *G*? The following is the DCGAN which is one of the most popular designs for the generator network. It performs multiple transposed convolutions to up sample *z* to generate the image *x*. We can view it as the deep learning classifier in the reverse direction.

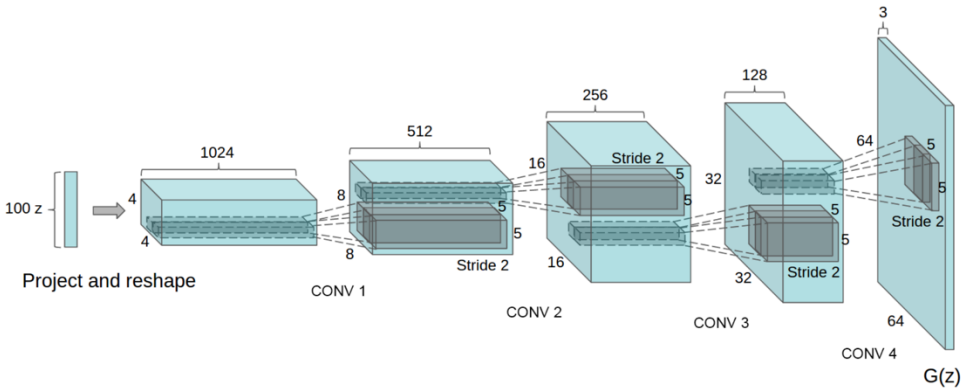


Figure 56: The process of GAN

But a generator alone will just create random noise. Conceptually, the discriminator in GAN provides guidance to the generator on what images to create. Let's consider a GAN's application, CycleGAN, that uses a generator to convert real scenery into a Monet style painting.

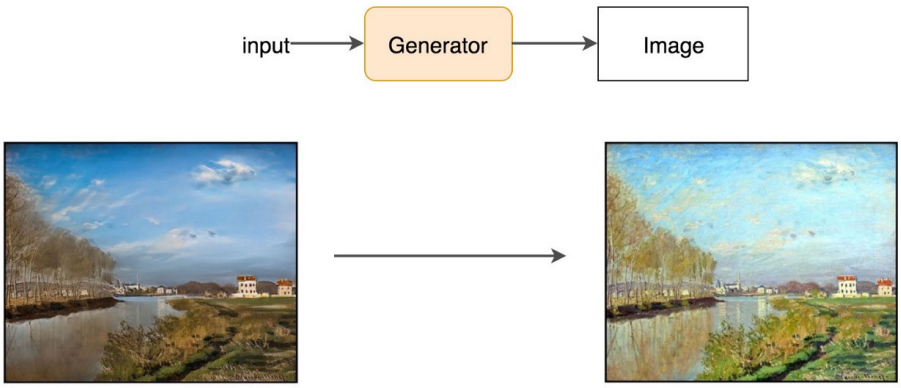


Figure 57: GAN generate a new image

By training with real images and generated images, GAN builds a discriminator to learn what features make images real. Then the same discriminator will provide feedback to the generator to create paintings that look like the real Monet paintings.

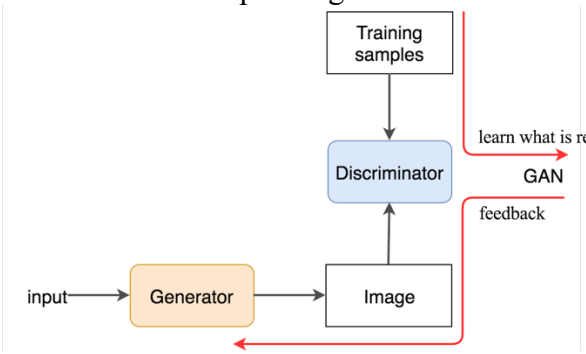
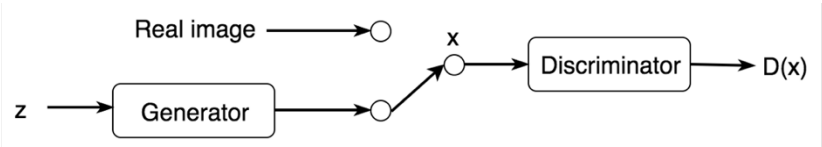


Figure 58: GAN training process

So how is it done technically? The discriminator looks at real images (training samples) and generated images separately. It distinguishes whether the input image to the discriminator is real or generated. The output *D(X)* is the probability that the input *x* is real, i.e. *P(class of input = real image)*.



We train the discriminator just like a deep network classifier. If the input is real, we want *D(x)=1*. If it is generated, it should be zero. Through this process, the discriminator identifies features that contribute to real images.

On the other hand, we want the generator to create images with *D(x) = 1* (matching the real image). So, we can train the generator by backpropagation this target value all the way back to the generator, i.e. we train the generator to create images that towards what the discriminator thinks it is real.

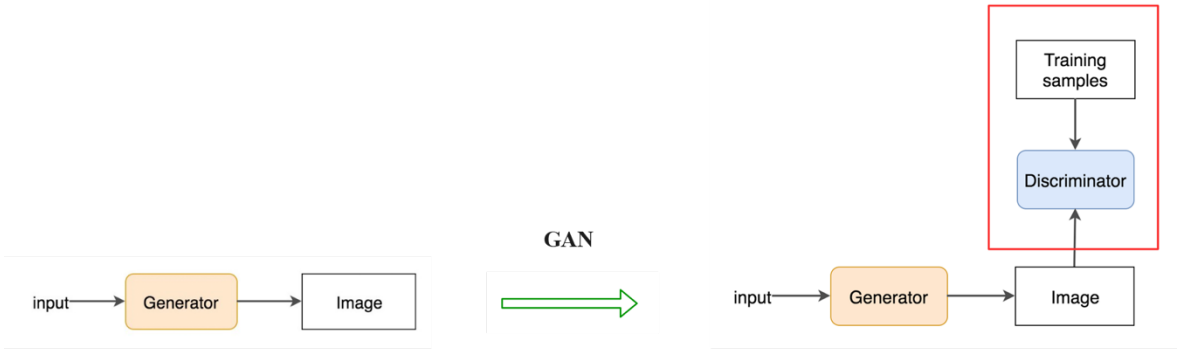


Figure 59: GAN process

Algorithm 1 Minibatch stochastic gradient descent training of generative adversarial nets. The number of steps to apply to the discriminator, k, is a hyperparameter. We used $k = 1$, the least expensive option, in our experiments.

for number of training iterations **do**

for k steps **do**

- Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
- Sample minibatch of m examples $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ from data generating distribution $p_{\text{data}}(\boldsymbol{x})$.
- Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right) \right].$$

end for

- Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
- Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right).$$

end for

The gradient-based updates can use any standard gradient-based learning rule. We used momentum in our experiments.

$$-\nabla_{\theta_g} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right) \rightarrow 0 \quad \textit{change to} \quad \nabla_{\theta_g} \log\left(D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right)$$

Algorithm 1: GANs

4. Semi-supervised learning

Semi-supervised Learning (SSL) is one of the artificial intelligences (AI) methods that have become popular in the last few months. Companies such as Google have been advancing the tools and frameworks relevant for building semi-supervised learning applications. Google Expander is a great example of a tool that reflects the advancements in semi-supervised learning applications.

Semi-Supervised Learning in the Real World

Semi-supervised learning models are becoming widely applicable in scenarios across a large variety of industries. Let's explore a few of the most well-known examples:

- Speech Analysis: Speech analysis is a classic example of the value of semi-supervised learning models. Labeling audio files typically is a very intensive tasks that requires a lot of human resources. Applying SSL techniques can really help to improve traditional speech analytic models.
- Protein Sequence Classification: Inferring the function of proteins typically requires active human intervention.
- Web Content Classification: Organizing the knowledge available in billions of web pages will advance different segments of AI. Unfortunately, that task typically requires human intervention to classify the content.

There are plenty of other scenarios for SSL models. However, not all AI scenarios can directly be tackled using SSL. There are a few essential characteristics that should be present on a problem to be effectively solvable using SSL.

1. Sizable Unlabeled Dataset: In SSL scenarios, the seize of the unlabeled dataset should be substantially bigger than the labeled data. Otherwise, the problem can be simply addressed using supervised algorithms.
2. Input-Output Proximity Symmetry: SSL operates by inferring classification for unlabeled data based on proximity with labeled data points. Inverting that reasoning, SSL scenarios entail that if two data points are part of the same cluster (determined by a K-means algorithm or similar) their outputs are likely to be in close proximity as well. Complementarily, if two data points are separated by a low-density area, their output should not be close.
3. Relatively Simple Labeling & Low-Dimension Nature of the Problem: In SSL scenarios, it is important that the inference of the labeled data doesn't become a problem more complicated than the original problem. This is known in AI circles as the "Vapnik Principle" which essentially states that in order to solve a problem we should not pick an intermediate problem of a higher order of complexity. Also, problems that use datasets with many dimensions or attributes are likely to become really challenging for SSL algorithms as the labeling task will become very complex.

II. Deep learning in chatbot:

1. Deep Convolutional neural networks (CNN)

Convolutional Neural Networks are very similar to ordinary Neural Networks from the previous chapter: they are made up of neurons that have learnable weights and biases. Each neuron receives some inputs, performs a dot product and optionally follows it with a non-linearity. The whole network still expresses a single differentiable score function: from the raw image pixels on one end to class scores at the other. And they still have a loss function (e.g. SVM/Softmax) on the last (fully-connected) layer and all the tips/tricks we developed for learning regular Neural Networks still apply.

Architecture Overview

Recall: Regular Neural Nets. As we saw in the previous chapter, Neural Networks receive an input (a single vector), and transform it through a series of hidden layers. Each hidden layer is made up of a set of neurons, where each neuron is fully connected to all neurons in the previous layer, and where neurons in a single layer function completely independently and do not share any connections. The last fully-connected layer is called the "output layer" and in classification settings it represents the class scores.

Regular Neural Nets don't scale well to full images. In CIFAR-10, images are only of size 32x32x3 (32 wide, 32 high, 3 color channels), so a single fully-connected neuron in a first hidden layer of a regular Neural Network would have 32*32*3 = 3072 weights. This amount still seems manageable, but clearly this fully-connected structure does not scale to larger images. For example, an image of more respectable size, e.g. 200x200x3, would lead to neurons that have 200*200*3 = 120,000 weights. Moreover, we would almost certainly want to have several such neurons, so the parameters would add up quickly! Clearly, this full connectivity is wasteful and the huge number of parameters would quickly lead to overfitting.

3D volumes of neurons. Convolutional Neural Networks take advantage of the fact that the input consists of images and they constrain the architecture in a more sensible way. In particular, unlike a regular Neural Network, the layers of a ConvNet have neurons arranged in 3 dimensions: width, height, depth. (Note that the word depth here refers to the third dimension of an activation volume, not to the depth of a full Neural Network, which can refer to the total number of layers in a network.) For example, the input images in CIFAR-10 are an input volume of activations, and the volume has dimensions 32x32x3 (width, height, depth respectively). As we will soon see, the neurons in a layer will only be connected to a small region of the layer before it, instead of all of the neurons in a fully-connected manner. Moreover, the final output layer would for CIFAR-10 have dimensions 1x1x10, because by the end of the ConvNet architecture we will reduce the full image into a single vector of class scores, arranged along the depth dimension.

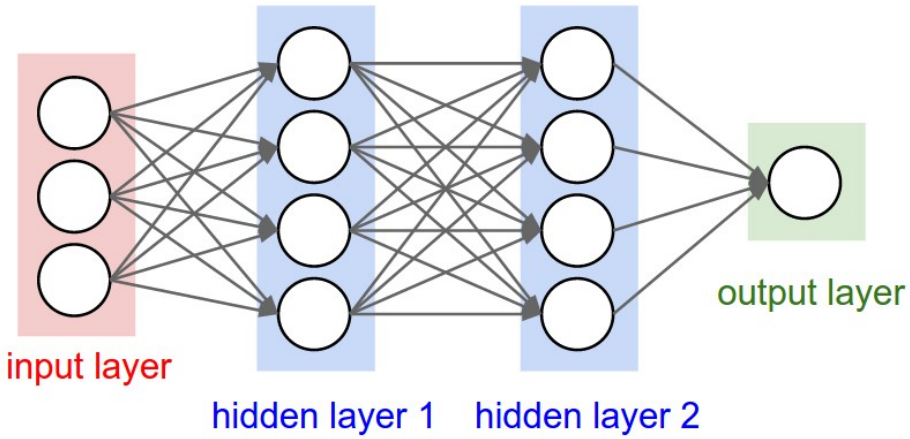


Figure 60: Different layers of CNN

2. Deep Recurrent neural networks (RNN)

The idea behind RNNs is to make use of sequential information. In a traditional neural network, we assume that all inputs (and outputs) are independent of each other. But for many tasks that's a very bad idea. If you want to predict the next word in a sentence you better know which words came before it. RNNs are called recurrent because they perform the same task for every element of a sequence, with the output being depended on the previous computations and you already know that they have a "memory" which captures information about what has been calculated so far.

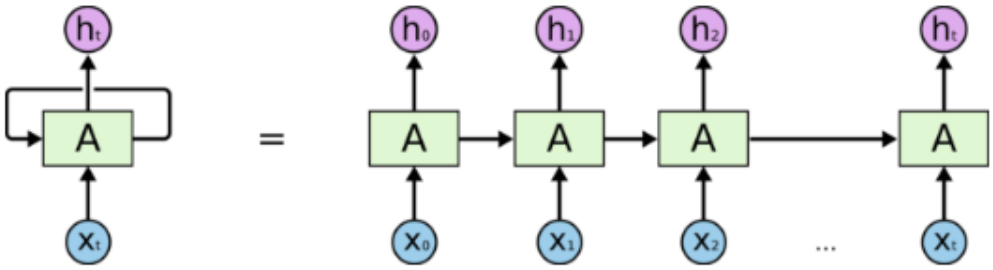


Figure 61: An unrolled recurrent network.

"Whenever there is a sequence of data and that temporal dynamics that connects the data is more important than the spatial content of each individual frame."

– Lex Fridman (MIT)

Different types of RNN's

The core reason that recurrent nets are more exciting is that they allow us to operate over sequences of vectors: Sequences in the input, the output, or in the most general case both. A few examples may make this more concrete:

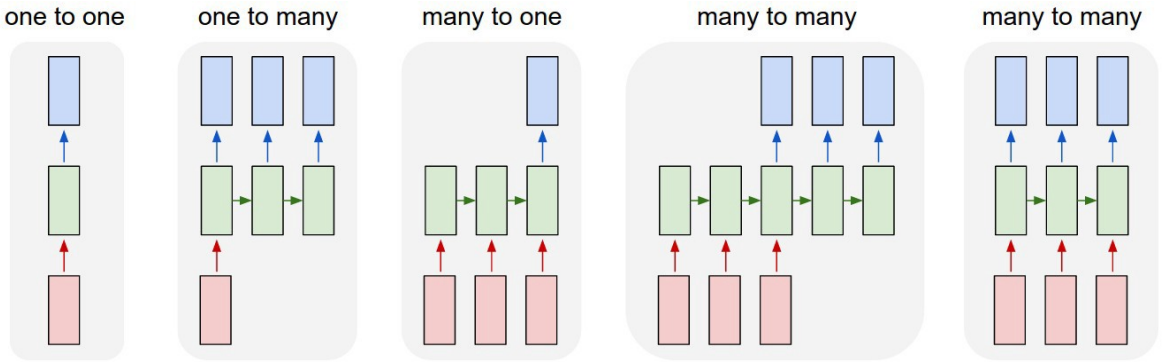


Figure 62: Different types of RNN.

Different types of Recurrent Neural Networks. (2) Sequence output (e.g. image captioning takes an image and outputs a sentence of words). (3) Sequence input (e.g. sentiment analysis where a given sentence is classified as expressing positive or negative sentiment). (4) Sequence input and sequence output (e.g. Machine Translation: an RNN reads a sentence in English and then outputs a sentence in French). (5) Synced sequence input and output (e.g. video classification where we wish to label each frame of the video). Notice that in every case are no pre-specified constraints on the lengths sequences because the recurrent transformation (green) is fixed and can be applied as many times as we like.

Each rectangle in above image represent Vectors and Arrows represent functions. Input vectors are Red in color, output vectors are blue and green holds RNN's state.

- One-to-one:

This also called as Plain/Vaniall Neural networks. It deals with Fixed size of input to Fixed size of Output where they are independent of previous information/output.

Ex: Image classification.

- One-to-Many:

It deals with fixed size of information as input that gives sequence of data as output.

Ex: Image Captioning takes image as input and outputs a sentence of words.

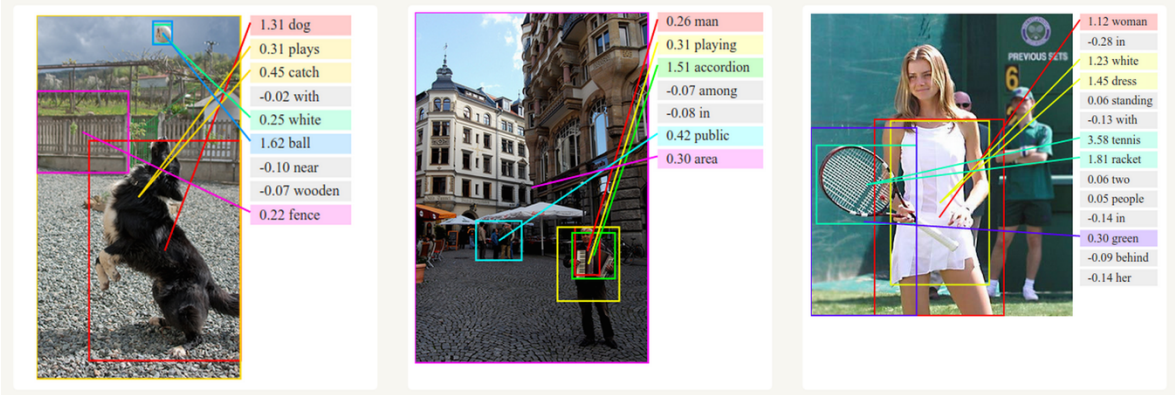


Figure 63: RNN one-to-many

- Many-to-One:

It takes Sequence of information as input and outputs a fixed size of output.

Ex: sentiment analysis where a given sentence is classified as expressing positive or negative sentiment.

- Many-to-Many:

It takes a Sequence of information as input and process it recurrently outputs a Sequence of data.

Ex: Machine Translation, where an RNN reads a sentence in English and then outputs a sentence in French.

- Bidirectional Many-to-Many:

Synced sequence input and output. Notice that in every case are no pre-specified constraints on the lengths sequences because the recurrent transformation (green) is fixed and can be applied as many times as we like.

- Example: video classification where we wish to label each frame of the video.

Advantages of Recurrent Neural Network

The main advantage of RNN over ANN is that RNN can model sequence of data (i.e. time series) so that each sample can be assumed to be dependent on previous ones. Recurrent neural network is even used with convolutional layers to extend the effective pixel neighborhood.

Disadvantages of Recurrent Neural Network

- Gradient vanishing and exploding problems.
- Training an RNN is a very difficult task.
- It cannot process very long sequences if using tanh or relu as an activation function

3. Deep Artificial neural networks (ANN)

The Artificial Neural Network (ANN) consists of 3 main components: The input layer and the output layer consist of only 1 layer; the hidden layer can have 1 or more layers depending on the specific problem. ANN works in a way that describes how the nervous system works with neurons connected

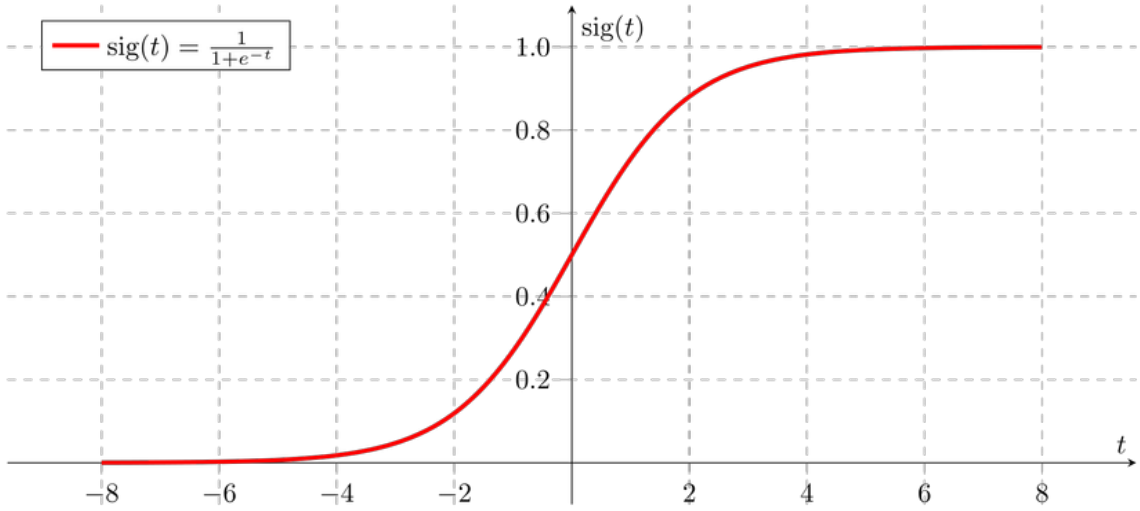
In ANN, except for the input layer, all nodes belonging to other layers are fully-connected with nodes belonging to the previous layer. Each node in the hidden layer receives the input matrix from the previous layer and combines the weight to get the result
The overview of ANN
Logistic Regression has the function is sigmoid function

$$g(z) = \frac{1}{1 + e^{-z}}$$

Hypothesys function:

$$h_\theta(x) = g(\theta^T X) = \frac{1}{1 + e^{-(\theta^T X)}}$$

The diagram is:



Cost function:

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m} Cost(h_\theta(x^{(i)}), y^{(i)})$$

With

$$Cost(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) \text{ nếu } y = 1 \\ -\log(1 - h_\theta(x)) \text{ nếu } y = 0 \end{cases}$$

cost function:

$$J(\theta) = -\frac{1}{m}[\sum_{i=1}^{m} y^{(i)} \log\left(h_\theta(x^{(i)})\right) + (1 - y^{(i)}) \log\left(1 - h_\theta(x^{(i)})\right)]$$

Combine to Regularization

$$J(\theta) = -\frac{1}{m}\left[\sum_{i=1}^{m} y^{(i)} \log\left(h_\theta(x^{(i)})\right) + (1 - y^{(i)}) \log\left(1 - h_\theta(x^{(i)})\right)\right] + \frac{\lambda}{2m}\sum_{j=1}^{n} \theta_j^2$$

So, with ANN for every node belonging to another layer, the input layer is a Logistic Regression we have

$$J(\Theta) = -\frac{1}{m}\left[\sum_{i=1}^{m}\sum_{k=1}^{K} y_k^{(i)} \log\left(h_\theta(x^{(i)})_k\right) + \left(1 - y_k^{(i)}\right) \log\left(1 - h_\theta(x^{(i)})_k\right)\right] + \frac{\lambda}{2m}\sum_{l=1}^{L}\sum_{i=1}^{s_l}\sum_{j=1}^{s_l+1}(\Theta_j^{(l)})^2$$

Our job now is to find \ Theta Θ such that J (\ Theta) J (Θ) min.
To find the minimum of J (\ Theta) J (Θ) we apply the Gradient Descent algorithm.

Repeat {

$$\Theta_j := \Theta_j - \alpha \frac{\partial}{\partial \Theta_j} J(\Theta);$$

}

A is the learning rate.
In order to do this, it is necessary to calculate \ frac {∂} {∂ \ Theta_ {j}} J (\ Theta)
∂Θ
j ∂ J (Θ), to calculate this derivative is relatively difficult and we need to implement an algorithm called backpropagation to calculate.

Forward Propagation
We have the neuron network:

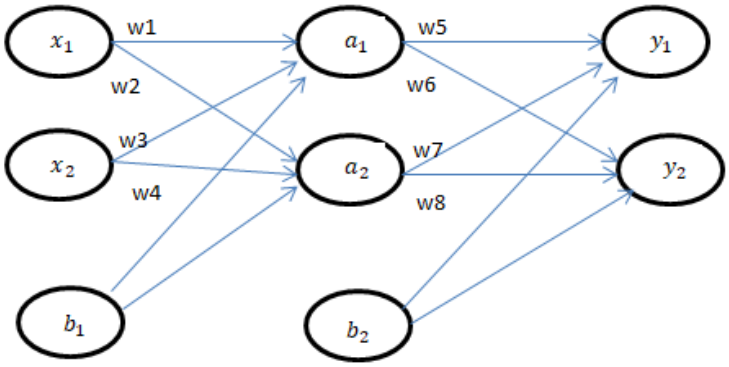


Figure 64: Example neural network.

Note:
X1, x2 is features of input
Y1, y2 is output
B1, b2 is bias
W1, w2 is number
With the name of forward propagation, we carry out calculate a1, a2, y1, y2 from left to right

$$z_1 = x_1 w_1 + x_2 w_3 + b_1$$

$$a_1 = sigmoid(z_1) = \frac{1}{1+e^{x_1 w_1 + x_2 w_3 + b_1}}$$

The same:

$$z_2 = x_1 w_2 + x_2 w_4 + b_2$$
$$a_2 = \frac{1}{1+e^{x_1 w_2 + x_2 w_4 + b_2}}$$
$$z_3 = a_1 w_5 + a_2 w_7 + b_2$$
$$y_1 = \frac{1}{1+e^{a_1 w_5 + a_2 w_7 + b_2}}$$
$$z_4 = a_1 w_6 + a_2 w_8 + b_2$$
$$y_2 = \frac{1}{1+e^{a_1 w_6 + a_2 w_8 + b_2}}$$

Forward propagation is a process of calculating the value at each node to serve the calculation in Backpropagation.

Back Propagation

$$\frac{\partial J(w)}{\partial y_1} = [-(T_1 * \log(y_1) + T_2 * \log(y_2))]' = -T_1 * \frac{1}{y_1 * \ln(10)}$$

$$\frac{\partial y_1}{\partial z_3} = \left(\frac{1}{1+e^{-(z_3)}}\right)' = -\frac{\left(e^{-(z_3)}\right)'}{(1+e^{-(z_3)})^2} = -\frac{e^{-(z_3)}\left(-(z_3)\right)'}{(1+e^{-(z_3)})^2} = \frac{e^{-(z_3)}}{(1+e^{-(z_3)})^2}$$

$$\frac{\partial z_3}{\partial w_5} = (a_1 w_5 + a_2 w_7 + b_2)' = a_1$$

III. Some techniques in chatbot

1. *Artificial Neural Networks*

Artificial neural networks (ANN) are a buzzword in machine learning right now - both for the technical expert and the everyday user. We see these algorithms at work every day to classify security images, recognize your face when you go to unlock your phone, and interpret your voice commands. In today's technology-driven world, we use neural networks every day, in all kinds of places.

However, while the application may be new, the technology itself has been around awhile. ANNs have been around since 1943, when they were first introduced in Warren McCulloch and Walter Pitts' paper "A Logical Calculus of the Ideas Immanent in Nervous Activity", describing how networks of artificial neurons could be used to solve various logic problems. However, they were not widely used until 1969 when a technique called backpropagation was created. This technique allows the artificial neurons to adjust themselves when the solution they come up with is not the solution that was expected. It increased accuracy in networks by allowing researchers to create self-adjusting networks that modify their own connections between neurons.

Today, ANNs are growing in popularity due to the increased amount of data and computing power available. In the past, these were two things that prevented researchers from using them in

academic settings. Now that we have more data than ever, and faster computer-processing, we can properly apply ANNs in more practical settings.

ANN is system modeled after the human brain, mimicking the ways we learn and make decisions. These networks consist of input and output layers, as well as hidden layers, similar to the neural networks in our brains.

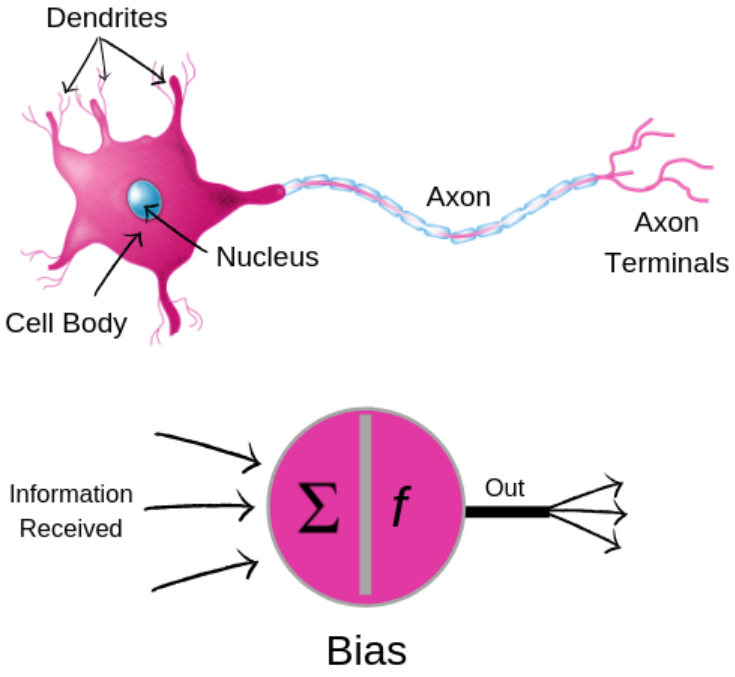


Figure 65: Human neurons

Each of these layers is simply a group of neurons. The input layer represents the data that we feed the ANN. The output layer is where the results of the algorithm are displayed for us to examine. In cases with greater information and more complexity, there are hidden layers between the input and output layers performing additional computations.
ANNs with more than one hidden layer are classified as deep-learning and their depth is determined by the exact number of hidden layers.

ANN in chatbot

One of the more recent uses of ANNs is in voice recognition technologies in chatbot. For this application, the ANN must be trained to accurately understand what people with different voices and accents are saying. This data is trained by looking at the mathematics behind sound waves and what constitutes a certain word. Once this data is properly trained, you can input data into the model and allow the hidden layers to compute. These layers learn to prioritize the wave patterns that mimic language. Once these hidden layers output accurate information, the model is ready to be used to recognize different voice commands that humans give it.

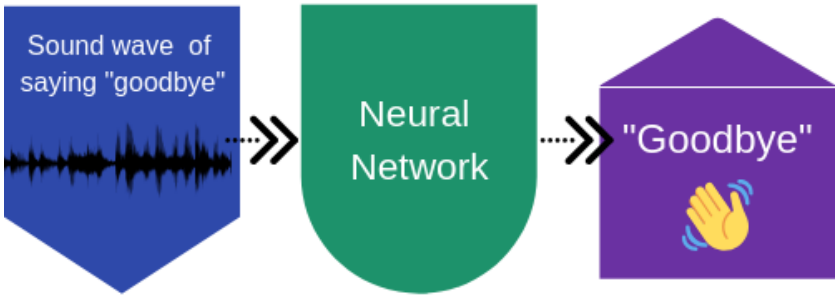


Figure 66: Chatbot using voice recognition

2. *Natural Language Understanding*

2.1. Entities

Entities are powerful tools used to extract parameter values from natural language. Anything that users want to know from their content will have a corresponding ENTITY.

For example: In the saying: “This Friday, John will fly from Hanoi to Da Nang instead of tomorrow. For example, above, "Tomorrow" or "This Friday" will have a corresponding entity called "Time". Hanoi, Da Nang, ... the corresponding entity will be "Location".
There are three types of Entities: the system (defined by Dialogflow), the developer (defined by the developer), and the user (built for each individual end-user in every request) entity. Each of these can be classified as mapping (with reference value), enum (without reference value), or mixed (containing other entities with specific names and return object type values).

2.2. Intents

Definition: Intents are the intent of the end-user, which is conveyed by the user to the bot. Users can mainly place intent in two main groups: Random intents: casual intents, and business intents: business intents.

Classify:

- Casual Intents:

These are the intent to start or end a conversation. For example, greetings such as hi, hello, hallo, ciao, or bye are commands that begin or end a conversation. These intentions direct the user's bot to respond with a suggestive answer like, "Hello, how can I help you?" or "Goodbye, thanks for talking to me." Random intentions also include Affirmative and Negative intent for statements like: "Ok", "yes", "No, not this." By general intentions, the bot's Casual Intents can handle all interactions with the user instead of having to put the story into a conversation with the bot.

- Business Intents:

These are the intentions that are directly connected with the original intent of the bots when being built, meaning the bot needs to support and handle these intentions of the end-users when they visit the website. It focuses on dealing with more intensive issues and takes time to handle as well as make a specific request. For example, when a user begins with the sentence: "When will Avengers 4 be released?", The bot will determine the intention of the user at this time is to know the information of an object, specifically here is a movie. This is a business intent (Business Intents), the bot will now have to find out the

year of the film's release and will also label the question in an understandable name, like "GetReleaseYearByTile" to then process the information and continue interacting with the user.

2.3. Context

Context means context or scope. For example, in Chatbot, users have just asked "Weather in Hanoi" and have received an answer, they put one more question "So what about Da Nang?", The case if there is no context Chatbot cannot understand what users are asking about anything related to Da Nang, and in case of context, Chatbot will immediately know the user is asking about time more of Da Nang. In an Intent of Dialogflow, there is input context and output context as shown below:

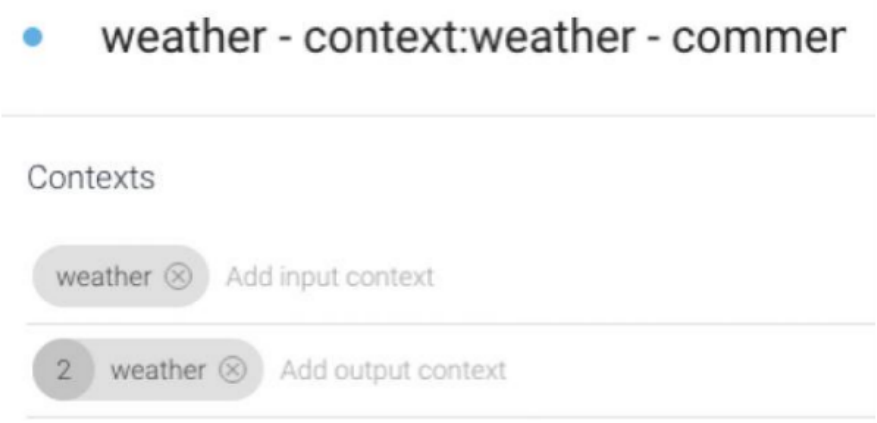


Figure 67: Example of context.

The input context is responsible for helping the Dialogflow determine that this intent is only available when there are corresponding contexts. By default, a Dialogflow context will expire 5 requests or 10 minutes after it was created. As shown in the figure, the number 2 in context weather means that this context will expire after 2 requests (or after 10 minutes).

3. *Natural Language Processing*

According to Wikipedia, “Natural Language Processing, also known as NLP, is an area of computer science and artificial intelligence concerned with the interactions between computers and human (natural) languages, in particular how to program computers to fruitfully process large amounts of natural language data.”

In lamens terms, Natural Language Processing (NLP) is concerned with how technology can meaningfully interpret and act on human language inputs. NLP allows technology such as Amazon’s Alexa to understand what you’re saying and how to react to it. Without NLP, AI that requires language inputs is relatively useless.

Why do Chatbots need NLP?

Like the previous example with Amazon’s Alexa, chatbots would be able to provide little to no value without Natural Language Processing (NLP). Natural Language Processing is what allows chatbots to understand your messages and respond appropriately. When you send a message with “Hello”, it is the NLP that lets the chatbot know that you’ve posted a standard

greeting, which in turn allows the chatbot to leverage its AI capabilities to come up with a fitting response. In this case, the chatbot will likely respond with a return greeting.

Without Natural Language Processing, a chatbot can't meaningfully differentiate between the responses "Hello" and "Goodbye". To a chatbot without NLP, "Hello" and "Goodbye" will both be nothing more than text-based user inputs. Natural Language Processing (NLP) helps provide context and meaning to text-based user inputs so that AI can come up with the best response.

NLP in Chatbots

NLP is a tool for computers to analyze, comprehend, and derive meaning from natural language in an intelligent and useful way. This goes way beyond the most recently developed chatbots and smart virtual assistants. In fact, natural language processing algorithms are everywhere from search, online translation, spam filters and spell checking.
So, by using NLP, developers can organize and structure the mass of unstructured data to perform tasks such as intelligent:

- Automatic summarization (intelligently shortening long pieces of text)
- Automatic suggestions (used to speed up writing of emails, messages, and other texts)
- Translation (translating phrases and ideas instead of word for word)
- Named entity recognition (used to locate and classify named entities in unstructured natural languages into pre-defined categories such as the organizations; person names; locations; codes; quantities; price; time; percentages)
- Relationship extraction (extraction of semantic relationships among the identified entities in natural language text/speech such as "is located in", "is married to", "is employed by", "lives in", etc.)
- Sentiment analysis (helps identify, for instance, positive, negative and neutral opinion form text or speech widely used to gain insights from social media comments, forums or survey responses)
- Speech recognition (enables computers to recognize and transform spoken language into text – dictation – and, if programmed, act upon that recognition – e.g. in case of assistants like Google Assistant Cortana or Apple's Siri)
- Topic segmentation (automatically divides written texts, speech or recordings into shorter, topically coherent segments and is used in improving information retrieval or speech recognition)

4. Dataset

Question-Answer Datasets for Chatbot Training
- Question-Answer Dataset: This corpus includes Wikipedia articles, manually-generated factoid questions from them, and manually-generated answers to these questions, for use in academic research.
Link: http://www.cs.cmu.edu/~ark/QA-data/
- The WikiQA Corpus: A publicly available set of question and sentence pairs, collected and annotated for research on open-domain question answering. In order to reflect the true information, need of general users, they used Bing query logs as the question source. Each question is linked to a Wikipedia page that potentially has the answer.

Link:https://www.microsoft.com/en-us/download/details.aspx?id=52419&from=http%3A%2F%2Fresearch.microsoft.com%2Fapps%2Fmobile%2Fdownload.aspx%3Fp%3D4495da01-db8c-4041-a7f6-7984a4f6a905
Customer Support Datasets for Chatbot Training
- Ubuntu Dialogue Corpus: Consists of almost one million two-person conversations extracted from the Ubuntu chat logs, used to receive technical support for various Ubuntu-related problems. The full dataset contains 930,000 dialogues and over 100,000,000 words
Link: https://www.kaggle.com/rtatman/ubuntu-dialogue-corpus
- Relational Strategies in Customer Service Dataset: A collection of travel-related customer service data from four sources. The conversation logs of three commercial customer service IVAs and the Airline forums on TripAdvisor.com during August 2016.
Link: https://s3-us-west-2.amazonaws.com/nextit-public/cl2017data.html
- Customer Support on Twitter: This dataset on Kaggle includes over 3 million tweets and replies from the biggest brands on Twitter.
Link: https://www.kaggle.com/thoughtvector/customer-support-on-twitter
Dialogue Datasets for Chatbot Training
- Semantic Web Interest Group IRC Chat Logs: This automatically generated IRC chat log is available in RDF, back to 2004, on a daily basis, including time stamps and nicknames.
Link: http://chatlogs.planetrdf.com/swig/
- Cornell Movie-Dialogs Corpus: This corpus contains a large metadata-rich collection of fictional conversations extracted from raw movie scripts: 220,579 conversational exchanges between 10,292 pairs of movie characters involving 9,035 characters from 617 movies.
Link: http://www.cs.cornell.edu/~cristian/Cornell_Movie-Dialogs_Corpus.html
- ConvAI2 Dataset: The dataset contains more than 2000 dialogues for a PersonaChat competition, where human evaluators recruited via the crowdsourcing platform Yandex.Toloka chatted with bots submitted by teams.
Link: http://convai.io/data/

D. DESIGN

Application machine learning and deep learning on chatbot shopping at Viet Nam

1. Recommender Systems

One of the most popular examples of machine learning in retails and e-commerce is the application of recommender system to increase sales by offering the relevant items for purchase that users could be highly interested in.

By leveraging the data collected across systems, many retailers and e-commerce (e.g. Amazon) companies have successfully implemented recommender systems providing highly personalized offers and a custom-tailored online shopping journey to users via their websites.

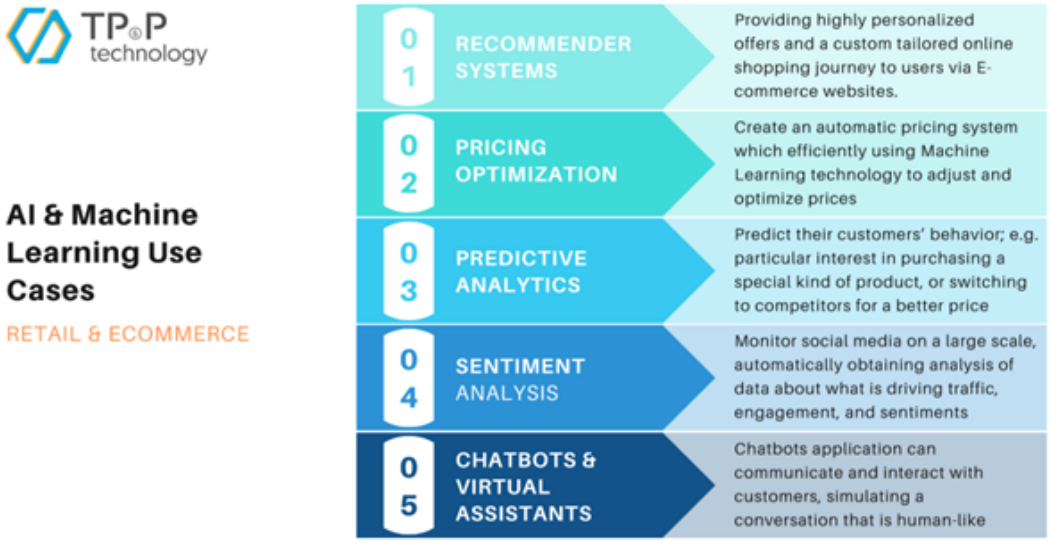


Figure 68: AI & Machine Learning use cases.

Not only recommender system makes it easier for customers to search for the content they're interested in, it also provides suggestions to users on the offers they would have never searched for in the first place. Moreover, companies are able to enhance their marketing activities by sending out personalized emails offering recipients with special or relevant items that suit their purchase profiles.

Once customers begin to feel like they are understood and paid particular attention to, they will likely to purchase more products or consume the services more.

More importantly, by understanding what exactly your customers want or currently looking for and providing it to them right away, it will be less likely that they will leave your platform looking elsewhere. This means higher conversion rates while reducing the chance of losing to competitors.

By integrating recommendation system into their e-commerce websites, retailers are able to provide added value to their customers, and at the same time enhance their sales process, staying ahead of the competition.

2. Pricing Optimization

Pricing is one of the critical factors ensuring business success and profitability. Retailers and e-commerce companies can leverage the immense power of machine learning to build an effective automation pricing solution.

As you may already know, a machine learning algorithm can learn patterns from data, instead of being explicitly programmed.

In case of pricing optimization, a developed algorithm takes into account a number of pricing variables to determine the best prices for items sold by retailers, as well as understanding how customers react to different pricing of products and services.

Retails companies can consider various factors such as: demand, supply, competition, and other external factors which affect their businesses in order to create an automatic pricing system which efficiently using Machine Learning Technology to adjust and optimize prices.

3. Predicting Customer Behavior

Imagine if businesses can predict their customers' behavior – such as particular interest in purchasing a special kind of product, or switching to competitors for a better price – having access to this information could open up so many sales opportunities for your companies.

Based on data collected regarding customers' previous behaviors, a system developed based on machine learning can analyze to predict how customers will behave in the future. Such system enables businesses to carry personalized marketing activities that are more effective than the traditional approaches.

For example, being able to predict which customers are more likely to convert to paid subscription after the trial period ends, or to know which ones are more likely to purchase a company's product in the next holiday season etc. allow businesses to send out very personalized offers, or provide special customers support focus on those users who are more likely to make purchase, result in a better conversion rate.

Performing sales and marketing activities that are based on predicted customers' needs also helps increase loyalty and retention rate.

4. Social Media: Tracking Brand & Customer Sentiment

As mentioned in our previous blog post, Social media has changed the way people shop with almost all major retailers brand have an active online presence these days.

More than just for social networking, customers can actually browse products and services, send enquiries and/or placed orders directly through social media.

Furthermore, many also use social media as an official contact channels providing customers care to their customers. Thus, tracking customers' sentiment and monitoring brand through social media is highly important to retailers. Thanks to AI and machine learning technologies, companies can now monitor their social media on a large scale, automatically obtaining analysis of business data about what is driving traffic, engagement, and customers' sentiments.

E. CASE STUDY

I. Shopping problem and Solution to solve it

1. The issues of traditional shopping

1.1. Online payment

Another challenge is to find a payment gateway that is smooth. Sometimes when the customers are directed to the payment page, their money is deducted and suddenly, the page shuts off without any notice to the consumer. And that's when the customer is in a fix. Then chasing the company for a refund is a different challenge altogether. With bulk sales happening at the same time, sometimes it's a WooCommerce specialist to ensure that the payment page is working fine. Also, sometimes the website asks too many secret questions or too much info before the customer can make the payment. This too can increase the perceived inconvenience during the purchase and leads to an abandoned cart.

Possible Solution**:** A quick fix is to email payment confirmation to the customer. If a customer gets an email confirming the order, they are not worried about the outcome. They know they are paying for an order that has been placed successfully. Also, keep the payment process simple and easy to execute without including too many stages.

1.2. Time

To cope with time/convenience risk, eBay Toolbar is a wise choice, with which shoppers can log into My eBay and your alerts quickly, seek merchandise efficiently in eBay.com, eBay Motors, Half.com and more, as well as manage items properly. For example, when searching items online, consumers can access both the Web and the eBay, customize the appearance of result list according to different aspects of individual references, such as location, buying format, distance. Then during the management period after the selection of goods, desktop alerts of the eBay Toolbar provides buyers with timely notifications to prevent neglecting any items that may be overdue. Traders can also browse their buying and selling status right through the desktop instead of logging in, which saves a lot of time.

Considering the shipping stage, eBay adopts a discriminating service strategy which provides various delivery types in order to meet different needs towards time demand. For instance, in United States, there are options like Next Day Air, 2nd Day Air, 3 Day Select among others, which takes corresponding time to deliver goods.

2. Why online shopping is more important now?

These are main reasons consumers give for online shopping.

You can see that convenience and price comparison are the main drivers. This chart also illustrates the importance of free shipping offers. The 40-page report also considers variation in perception by country and in different generation groups from Baby Boomers to Gen X and Millennials.

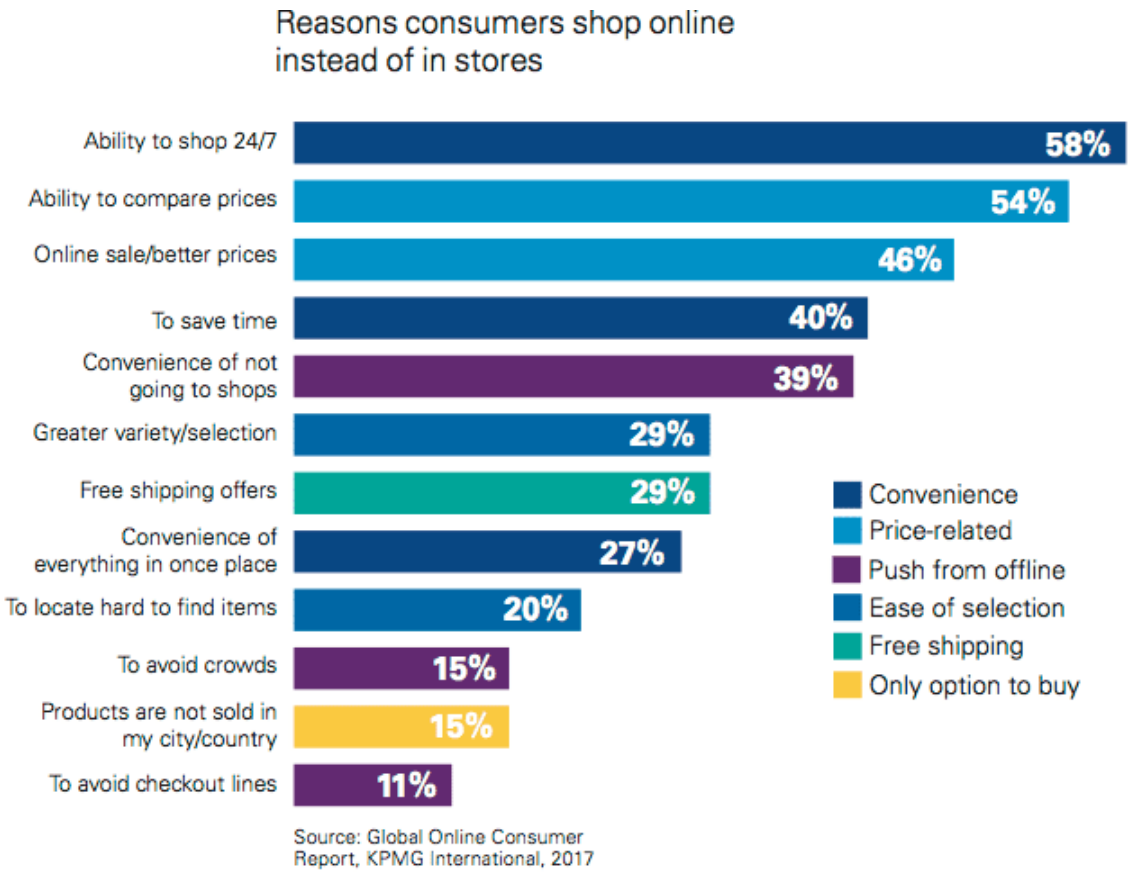


Figure 69: Reasons consumer's shop online instead of in stores

3. The role of applying deep learning and machine learning on chatbot shopping

Chatbot is another popular example of applying AI and Machine learning technologies into retail businesses. A typical chatbots application can communicate and interact with customers, simulating a conversation that is human-like providing answers to frequently asked questions (FAQ) by customers. For big e-commerce companies with large catalogue of many items, sometime it would be difficult for customers to search for the specific item they need. More particularly, many customers want and need to search for items based on item attributes (e.g. color, size, etc.) without knowing the exact search term. An advanced chatbot application can help customers with their request fairly easily, similar to a human sales associate does.

Additionally, chatbot can provide added value, enhancing customers shopping experience by suggesting additional items for purchase, and handle a significant part of your company's online customer service. Above are some of the common examples of applying machine learning technology in retail, business organizations in other industries such as manufacturing, healthcare, and transportations, etc. are also using machine learning today to better serve customers, improve their business process and innovations.

- Healthcare**:** One example for the use of AI in healthcare is to help detect abnormalities in X-rays and MRIs or application of a HealthCare bot which is an AI application patient can interact with via a HealthCare website or via telephone to receive help with their requests.
- Transportation: Autonomous Vehicles such as self-driven cars and trucks have been of high interest in the last several years. The leading examples include big players like Uber and Tesla who have successfully built self-driving cars and trucks solutions to save time and increase productivity.

- Finance**:** AI and Machine Learning technologies can be used in finance to detect and prevent frauds automatically. By giving the algorithm large datasets about real and fraudulent activities, machine learning models can train themselves to make better guesses on which transactions are most likely to be fraudulent and such.

Data Required for Using Machine Learning in Your Businesses:

In the various examples described above, machine learning models are trained with business data in different formats (e.g. text, images, numerical, etc.) regarding customers, products, competitors, etc. gathering from various sources. Without data, machine learning models wouldn't be able to train themselves.

In case your businesses are newly established (e.g. start-ups) or do not have the data required yet, you may need to consider ways to collect such data by crawling the web or using services from machine learning consulting companies.

Generally speaking, the bigger the volume of data gathered, the better the results (although in some cases, small data-sets still provide good and exploitable results).

However, the more important thing is to understand your business well and put yourself in the perspective of customers so that you can provide precisely what they want anytime, anywhere in a very personalized way.

4. Chatbot shopping in Vietnam

5 Popular Chatbots in Vietnam. In other words, Chatbot is a robot that automatically responds to customer messages on Facebook, Website, ... And with outstanding advantages, far beyond people in the field of care. Customer care, the chatbot has been and is a useful solution for many businesses.

CHATFUEL
Chatfuel is a free chatbot creation tool that helps you build a chatbot. More than 200,000 chatbots have been created using Chatfuel (as of June 2018) and they serve more than 145 million users globally.
Here are the outstanding advantages of Chatfuel that will surely make you "moved":
- All free
- Do not use code, easily set up bots with blocks
- No need to submit a review with Facebook.
- Easy connection to Facebook fanpage Unlimited number of bots can create
- Easily connect to other social networks and services like Instagram, Twitter, Youtube, ... with plugins
- Upgrade and easy maintenance
- Integrating all kinds of APIs into one chatbot, helping developers easily upgrade and improve your chatbot
- There is a sample library (template) for your reference
- Support more than 50 languages worldwide

MESSNOW

Messnow is the startup of choice to accelerate global startup from Facebook and is sponsored by Facebook, Microsoft, and Amazon. Trusted by many Vietnamese and international brands.
What advantages does Messnow have? Why do many businesses in the field of events, hotels, restaurants in Vietnam use Messnow? Here is the answer for you:

The interface is entirely in Vietnamese

There are detailed instructions menu step by step to install and deploy the chatbot
Integration with Zalo messaging application and Zalo shop, can create chatbots on Zalo
Multilingual capability with more than 50 languages worldwide
Diverse application store, which helps you integrate more functions
Create and deploy Facebook chatbot quickly, easily
The ability to handle natural language accurately
Warehouse of professional templates with hundreds of different presets set up
Upgrade to the Pro version for only 199,000 VND / month
Statistics show that more than 10,000 chatbots have been created from Messnow with over 5 million users. With a full interface in Vietnamese, this is definitely a tool to help you easily create chatbots as well as edit and upgrade chatbots.

MANYCHAT

This is a tool that directly competes with Chatfuel in terms of features as well as chatbot deployment technology. Here are the great features that ManyChat offers when you create chatbots with this tool:
Completely free and no code is needed
Easy to install on Facebook Fan page without creating an app or waiting for approval
Create unlimited chatbots
Create a chatbot script and set up quick options
In addition, ManyChat also integrates many real-time technologies (real-time), extremely good support for online shop owners, or Livestream sales. Compared to Chatfuel, ManyChat offers some of the following free tools:
Activate the free marketing plan series
Subscribe users to join chat automatically
Zapier integration can be combined with CRM and database synchronization
There are many other features of ManyChat that we cannot cover in this article. You can read the comparison article between the many tools ManyChat and Chatfuel here.

CHATTY PEOPLE

With ChattyPeople's chatbot creation tool, you can control how it works by building a decision tree with the questions and answers you create. The highlight of ChattyPeople is its simplicity.
However, compared to other chatbot creation tools, ChattyPeople has its own strengths.
Support multimedia chat, picture, sound, etc.
Integrating with many payment systems such as PayPal, Stripe, etc.
Good support for and syncing from multiple sources such as on the Website, Facebook, Shopify, etc.
Integrate e-commerce platforms such as OpenCart, Magento and WooCommerce.
Let's push notifications when you have promotions.

With ChattyPeople, your users can place orders without leaving Facebook Messenger. You can even specify exactly which product or service you want to show to users based on the questions they ask.

HARAFUNNEL

Sponsored by Haravan, helps you put potential customers in contact channels into the care flow and interact directly 24/7 on Facebook Messenger.
As of June 2018, more than 35,000 Facebook Messenger Bots were created using Harafunnel. Trusted by more than 15,000 businesses and brands.
The most outstanding advantages of Harafunnel are:

- Help you create a sales funnel and take care of potential customers
- All free
- Chatbot initialization interface in Vietnamese
- Enthusiastic support team, support 7 days a week
- Manage Comments and Inbox effectively
- There are various tools and support features
- Take advantage of a chatbot as a tool to collect multi-channel customer data, 24/7 automated customer care, and can set up automated marketing processes.

5. The solution to solve the problem on chatbot shopping in Vietnam

5.1. Solutions to financial risk

Considering consumers' worry of misuse of credit card information, eBay Inc. acquired PayPal, a global leader in online payment solutions, in October 2002, which uses the most exceptional exclusive fraud prevention systems, enabling any individual or business with an email address to securely, easily and quickly send and receive payments online. When using PayPal, the credit card information is kept in its server, thus preventing sellers from knowing that information. On the other hand, the payment process completes immediately as the money transfers directly to the seller's account. Then, customer can track the progress of the payment by My eBay or PayPal account. Additionally, PayPal also provide Buyer Protection in case of non-receipt of item or dramatically difference from seller's description. This program covers eligible transactions for up to the full purchase price plus original shipping charges. In order to use PayPal, consumers are required to bind the eBay and PayPal account together and confirm the address with it.

5.2. Solutions to psychological risk

As the bad emotion of this risk arises from the disclosure of personal information, keeping this information private plays an essential role in the process to solve the problem. Stated by eBay, they treasure the data as valuable assets, so store them on their servers located in America by using variety of methods (password, encryption, physical security, etc.) to prevent unlawful entrance and exposure, unauthorized disclosure to third party for marketing purpose as well. They also make use of Cookies, most of which are session cookies that can disappear by itself immediately after logging out from eBay or exiting from the web pages, to lessen the times of inputting passwords, thus enforcing trust and safety. Another guarantee of eBay is that they forbid spreading spam or spyware through their communication tools by automatic scan and manual filter.

5.3. Solutions to product performance risk

Josh Boyd suggested that individual identities combined with feedback ratings and icons comprise a trust-building function at eBay which can reduce product performance risk perceived by consumers. Community members choose their own usernames to maintain individual identities used to distinguish each other. Feedback ratings are the numbers appear next to one's username, calculated by the sum of positive comments minus that of negative ones describing the behavior of the user during past trades. A higher number means a good reputation, resulting out more traders to involve in the trade with that person. By clicking the rating, there will arise more details about the comments, including negative, positive, as well as neutral ones. To maintain a minimum performance level from all sellers, eBay sets out specific amount of detailed seller ratings (DSRs) from the aspects of communication, delivery time, shipping fees and items. Sellers who do not meet the requirement will be forbidden to register new accounts or use the old ones to bypass any restrictions as a punishment. Icons also appear after usernames, reflecting the status of the user and reminding potential bidders or sellers of the recent change in status or new comers. For example, a pair of sunglasses indicates the recent change of status while stars of various colors represent certain levels of experience.

To sum up, the username distinguishes individuals, the feedback rating reflects the amount of positive experience attained by different individuals, and the icon generates the status according to the feedback rating. The higher status, the more traders will be attracted. The desire to gain or maintain privileged status motivates people to follow eBay's rules (Josh Boyd, 2002).
Solution to time/convenience risk

To cope with time/convenience risk, eBay Toolbar is a wise choice, with which shoppers can log into My eBay and your alerts quickly, seek merchandise efficiently in eBay.com, eBay Motors, Half.com and more, as well as manage items properly. For example, when searching items online, consumers can access both the Web and the eBay, customize the appearance of result list according to different aspects of individual references, such as location, buying format, distance. Then during the management period after the selection of goods, desktop alerts of the eBay Toolbar provides buyers with timely notifications to prevent neglecting any items that may be overdue. Traders can also browse their buying and selling status right through the desktop instead of logging in, which saves a lot of time.

Considering the shipping stage, eBay adopts a discriminating service strategy which provides various delivery types in order to meet different needs towards time demand. For instance, in United States, there are options like Next Day Air, 2nd Day Air, 3 Day Select among others, which takes corresponding time to deliver goods.

Comparison of solutions

Although financial risk, psychological risk, product performance risk and time/convenience risk reflect concerns on money, psychology, product, time respectively, which are different dimensions of consumer utility, they go hand in hand, which means that once one aspect of solution take place, it will prompt other solutions to carry out.

F. CONCLUSION AND DEVELOPMENT

I. Conclusion

After studying and perform this project about “Deep learning and Machine learning in chatbot”, we done project and gain the knowledge such as

- Understanding deeply Artificial Intelligence
- Understanding Machine Learning and deep learning
- Know types of methods deep learning and machine learning.
- Study the algorithm in each method
- Know some framework that support to develop chatbot
- Understand some methods and techniques in chatbot

Besides that, my group also gain a lot of useful knowledge and the power of its. After studying this project, we gain a lot of useful things and achieve our favorite topic. It’s still existed some obstacles, but I will base on this project and learn more in the future to have real project

II. Development:

- Training chatbots to respond more flexibly and smarter through expanding the construction and improving the database of the system to become more optimal and diversified.
- Integrating the automatic control program into the smartphone to make it more convenient for users as well as control information and control the system in many locations at the same time.
- Developing a more user-friendly system, in addition to designing an easier-to-use interface, is also making chatbots become like a "psychologist" interacting with users through data. collect from a variety of sources, bots' Machine Learning, and simulation of meetings
- Voice is predicted by psychologists and then saved in the system, etc.
- In addition, we can integrate weather forecasting, automatic shopping online and pay by card, or even check basic health information to remind and alert status for users through monitoring the heart rate, calories, weight, number of steps of the day, the long sitting of the user, etc. Chatbots training flexibility and smarter response through the expansion of database construction and improvement of the system become more optimal and more diverse.
- Integrating the automatic control program into the smartphone to make it more convenient for users as well as control information and control the system in many locations at the same time.
- Developing a more user-friendly system, in addition to designing an easier-to-use interface, is also making chatbots become like a "psychologist" interacting with users through data. collect from a variety of sources, bots' Machine Learning, and simulation of meetings
- Voice is predicted by psychologists and then saved in the system, etc.
- In addition, we can integrate weather forecasting, automatic shopping online and pay by card, or even check basic health information to remind and alert people. Status for users through monitoring heart rate, calories, weight, steps of the day, long sitting of the user, etc.

G. REFERENCES

Daley, S. (2018, December 19). *19 EXAMPLES OF ARTIFICIAL INTELLIGENCE SHAKING UP BUSINESS AS USUAL*. Retrieved from builtin: https://builtin.com/artificial-intelligence/examples-ai-in-industry https://cs.stanford.edu/people/eroberts/courses/soco/projects/neural-networks/History/history1.html https://en.wikipedia.org/wiki/Deep_learning https://www.androidauthority.com/what-is-ma

Wikipedia. (2020, June 23). *Deep learning*. Retrieved from wikipedia: https://en.wikipedia.org/wiki/Deep_learning

Sims, G. (2015, July 7). *What is machine learning?* . Retrieved from androidauthority: https://www.androidauthority.com/what-is-machine-learning-621659/

Expertsystem. (2020, May 6). *What is Machine Learning? A definition* . Retrieved from expertsystem: https://expertsystem.com/machine-learning-definition/

Wikipedia. (2020, June 24). *Recurrent neural network* . Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Recurrent_neural_network

Martin Adam, Michael Wessel, & Alexander Benlian . (2020). AI-based chatbots in customer service and their effects on user compliance. *Electronic Markets*, 204–215.

Chatcompose. (n.d.). *What is a chatbot?* . Retrieved from chatcompose: https://www.chatcompose.com/what-are-chatbots.html

June. (1995). *The application of semantic classification trees to natural language understanding*. Retrieved from researchgate: https://www.researchgate.net/publication/3192424_The_application_of_semantic_classification_trees_to_natural_language_understanding

Gonfalonieri, A. (2018, November 21). *How Amazon Alexa works? Your guide to Natural Language Processing (AI)*. Retrieved from Medium: https://towardsdatascience.com/how-amazon-alexa-works-your-guide-to-natural-language-processing-ai-7506004709d3

Geeksforgeeks. (n.d.). *Genetic Algorithms* . Retrieved from geeksforgeeks: https://www.geeksforgeeks.org/genetic-algorithms/

Dabbura, I. (2018, September 18). *K-means Clustering: Algorithm, Applications, Evaluation Methods, and Drawbacks*. Retrieved from medium: https://towardsdatascience.com/k-means-clustering-algorithm-applications-evaluation-methods-and-drawbacks-aa03e644b48a

Violante, A. (2019, March 19). *Simple Reinforcement Learning: Q-learning*. Retrieved from towardsdatascience: https://towardsdatascience.com/simple-reinforcement-learning-q-learning-fcddc4b6fe56

Hui, J. (2018, June 20). *GAN — What is Generative Adversarial Networks GAN?* Retrieved from medium: https://medium.com/@jonathan_hui/gan-whats-generative-adversarial-networks-and-its-application-f39ed278ef09

gudikandula, p. (2019, March 28). *Recurrent Neural Networks and LSTM explained*. Retrieved from medium: https://medium.com/@purnasaigudikandula/recurrent-neural-networks-and-lstm-explained-7f51c7f6bbb9

Schott, M. (2019 , April 26). *Artificial Neural Networks for Machine Learning*. Retrieved from medium: https://medium.com/capital-one-tech/artificial-neural-networks-for-machine-learning-79c67d0681e9

Tpptechnology. (n.d.). *AI & Machine Learning Use Cases in Retail and Other Industries | TP&P Technology Vietnam*. Retrieved from tpptechnology:

https://www.tpptechnology.com/ai-machine-learning-use-cases-in-retail-and-other-industries

Icecubedigital. (n.d.). *Problems Customers Face While Shopping Online* . Retrieved from icecubedigital: https://www.icecubedigital.com/blog/problems-customers-face-while-shopping-online/

Online Shopping Problems and Solutions . (n.d.). Retrieved from ukessays: https://www.ukessays.com/essays/economics/internet-shopping-problems-and-solutions.php

Muggleton, S. (2014, Juanary). *Alan Turing and the development of Artificial Intelligence* . Retrieved from Researchgate: https://www.researchgate.net/publication/262290526_Alan_Turing_and_the_development_of_Artificial_Intelligence

Stanford. (n.d.). *Neural Network*. Retrieved from Stanford: https://cs.stanford.edu/people/eroberts/courses/soco/projects/neural-networks/History/history1.html