الجمهورية الديمقراطية الشعبية الجزائرية

Ministry of Higher Education and Scientific Research

Laghouat University

Faculty of Science

Computer Science Department

Thesis of graduation project

Presented to obtain the Master degree in Computer Science

Specialty: **Information and Decision system**

Theme:

Application of Deep Learning approach for COVID-19 cases forecasting

Presented by:

Atig Meriem

Jury members:

Mr. Maicha Med El Habib	MAA	University of Laghouat	President
Mr. Guellouma Younes	MCA	University of Laghouat	Examiner
Mr. Bouakkaz Mustafa	MCA	University of Laghouat	Supervisor

University year: 2021- 2022

Acknowledgments

I first want to thank -الله- for strength and patience

That he gave me to overcome all

The trials (happiness and misfortunes) experienced during this thesis.

This work occurred at the Computer Science Department.

I want to thank **Mr.Bouakkaz** for drafting this thesis,

Thank you for your professional support,

Thank you for all the helpful remarks and criticisms.

I also want to thank the members of the jury, who will review my work,

All the teachers of Computer Science department,

And all the promotion of 2d year Master 2021/2022.

M.A

At first, I would say-الحمد لله - who gave me the power and the patience to finish my work

I dedicate my work sincerely to my dear parents Med and Nacira;

My dear brothers Aissa, Omar and Abd-el-Kader;

My dear sisters Aisha, Ikram Fatima Zahraa;

And my best friend Nesrin;

M.A

ملخص

أدى كوفيد-19، المعروف أيضا باسم الفيروس كورونا، إلى شلل جزئي للعالم من جميع النواحي تقريبا، أكثر من 520 مليون شخص في جميع أنحاء العالم أصيبوا به، وتسبب في وفاة أكثر من 6 مليون شخص.

في العقود الماضية، أظهر مجال الذكاء الصناعي والتعلم العميق تطورا كبيرا واستخداما متعددا في الكثير من المجالات مثل الامن والسيارات ذاتية القيادة الروبوتات وخاصة الطب ... التأثير الكبير لهذا الوباء دفعنا إلى محاولة إيجاد واستكشاف طرق تساعد على الحد من انتشاره باستعمال هذه التكنولوجيا لهذا اقترحنا طريقة باستعمال الذاكرة طويلة المدى نوع من الشبكات العصبية المتكررة من اجل توقع عدد حالات في المستقبل القريب عن طريق الاستعانة بالمعلومات المتواجدة عندنا من حالات السابقة منذ بداية الوباء.

الكلمات المفتاحية: كوفيد-19، التنبؤ، التعلم العميق، الشبكة العصبية، الشبكة العصبية المتكررة، الذاكرة طويلة المدى، التعلم الآلي، الذكاء الاصطناعي.

Abstract

COVID-19, also known as the coronavirus, has paralyzed the whole world, infected more than 520 million people worldwide, and caused the death of more than 6 million people.

Over the past few decades, the field of artificial intelligence and deep learning has shown tremendous development and multiple uses in many fields such as security, self-driving cars, robotics, and especially healthcare... The enormous impact of this pandemic has prompted us to try to find and explore ways to use this technology to help limit its spread We proposed a method using long-short term memory (LSTM) a type of recurrent neural network (RNN) to predict the number of cases in the near future by using the information we have from previous cases since the beginning of the epidemic.

Keywords: covid-19, Prediction, Deep Learning, Neural Network(NN), Recurrent Neural Network (RNN), Long-Short Term Memory (LSTM), Machine Learning, Artificial Intelligence.

Table of Contents

Glossary

COVID-19: Coronavirus disease of 2019

SARS Cov-2: Severe Acute Respiratory Syndrome Coronavirus 2

SARS: Severe Acute Respiratory Syndrome

MERS: Middle East Respiratory Syndrome

WHO: World Health Organization

AI: Artificial Intelligence

ML: Machine Learning

ES: Exponential Smoothing

LASSO: Least Absolute Shrinkage and Selection Operator

LR: Linear Regression

SVM: Support Vector Machine

NB: Naïve Bayes

DL: Deep learning

NN: Neural Networks

ANN: Artificial Neural Networks

CNN: Convolutional Neural Network

DBN: Deep Belief Networks

RBM: Restreint Boltzmann machine

RNN: Recurrent Neural Networks

BRNN: Bi-directional RNN

LSTM: Long Short Term Memory

BLSTM: Bi-directional LSTM

GRU: Gated Recurrent Units

BPTT: Backpropagation Throw Time

JHU: Johns Hopkins University

CSSE: Center for Systems Science and Engineering

CRC: Coronavirus Resource Center

Lists of Figures

Lists of Tables

General Introduction

Coronaviruses are large RNA viruses known since the mid-1960s. They cause mild-to-moderate disease of the upper respiratory tract, similar to a cold [1]. Two known coronaviruses are severe acute respiratory syndrome coronavirus (SARS-CoV) and Middle East respiratory syndrome coronavirus (MERS-CoV). In December 2019, a new coronavirus infection emerged in Wuhan, Hubei Province, China [2]. On January 7, 2020, the new virus was identified as Covid-19. From December to March 2020, the world witnessed a huge spread of Covid-19 infections, when the World Health Organization (WHO) declared a pandemic. According to the World Health Organization [3], as of May 10, 2022, over 520 million confirmed Covid-19 cases and 6 million Covid-19 deaths have been reported globally. The virus, leading to various measures including country closures, curfews, and travel restrictions, has affected countries around the world. Although the most common symptoms of Covid-19 infection are usually mild, the infection can lead to serious and even fatal complications in some patients.

Machine learning and deep learning are branches of artificial intelligence (AI) that produce systems that can learn to predict future data from historical data. Regarding Covid-19, various studies have been carried out around the world to predict the number of new cases in the future. Therefore, forecasting new Covid-19 cases in the short, medium and long term provides governments with an effective means of preventing socio-economic crises, thus avoiding the uncontrollable situations like the lack of hospital capacity and the economic crisis with food shortages and cash flow that we experienced in the first year. In our project, we built a Long Short Term Memory (LSTM)-based predictive model, an extension of

Recurrent Neural Networks (RNN) on the daily updated Johns Hopkins dataset, to predict the number of new Covid-19 cases in the future. Our approach is based on applying deep learning techniques to time series data for forecasting. We evaluated our proposed model against many metrics and compared it with several other models; it showed accurate results.

Organization of The dissertation

First, we started with a general introduction where we showed the objective of our project, and then we introduced three chapters that are as follows:

1. COVID-19

   In this chapter, we will identify the disease caused by Covid-19 by technical definitions, and then we explain through the history of the disease its spread throughout the world and then focus, particularly, on the evolution of the pandemic in Algeria.

2. Machine Learning and Related work :

In this chapter, we will see a view on artificial intelligence and machine learning and some related work to our context to help us decide which machine learning method to use in our prediction mode

3. Deep Learning and Implementations and experimentations :

In this chapter, we will explain deep learning to know more about our used method and our implementation and experimentation phase

Finally, this paper ends with a general conclusion.

Chapter I: COVID-19

Introduction

In December 2019, an outbreak of pneumonia of unknown etiology was noticed in the city of Wuhan, China, which then spread around the world. In January 2020, this pneumonia-like illness was confirmed [4].Being a novel coronavirus known as SARS-CoV-2 [5]. This virus belongs to the Coronaviridae, a large family of enveloped single-stranded RNA viruses [6]. Coronaviruses are well known to cause various illnesses, from the common cold to large outbreaks, such as severe acute respiratory syndrome (SARS) [7] and the Middle East respiratory syndrome (MERS) [8]. In March 2020, the World Health Organization (WHO) classified Covid-19 as a pandemic that could threaten millions of people worldwide. Since then, the number of confirmed cases has increased because this new viral disease is highly contagious during the incubation period.

1. COVID-19

1.1. Definition of Covid-19

Covid-19 (Figure 1.1) refers to “Coronavirus Disease 2019”, the disease caused by a virus of the family Coronaviridae, SARS-CoV-2. This infectious disease is a zoonosis, the origin of which is still debated, which emerged in December 2019 in the city of Wuhan, in the province of Hubei in China. It quickly spread throughout China and then abroad, causing a worldwide epidemic. [9]Covid-19 is a respiratory disease that can be fatal in patients weakened by age or another chronic disease.

Figure 1.1: Covid-19 virus body.

1.2. The name COVID-19

During the early days of the COVID-19 pandemic, the disease and virus were sometimes called "coronavirus", "Wuhan coronavirus", or "Wuhan pneumonia". In January 2020, the World Health Organization (WHO) tentatively named it "2019-nCoV", short for "2019 Novel Coronavirus", or "2019 Novel Coronavirus Acute Respiratory Disease". This naming was based on the organization's 2015 guidelines for naming novel viruses and diseases, avoiding the use of geographic locations (such as Wuhan), in part to prevent social stigma. On 11 February 2020, the WHO named the disease COVID-19 (short for coronavirus disease 2019), and they named the causative virus SARS-CoV-2 (following SARS-CoV). WHO Director-General Tedros Adhanom Ghebreyesus explained that CO stands for coronavirus, VI for virus, and D for disease, while 19 stands for the year that the outbreak was first detected. [10]

1.3. Types of coronavirus

There are four main subgroups of coronaviruses, called alpha, beta, gamma, and delta. And seven different forms, four of which are common (less serious than the others) [11]:

• 229E (alpha-coronavirus)

• NL63 (alpha-coronavirus)

• OC43 (beta-coronavirus)

• HKU1 (beta-coronavirus)

• MERS-COV (the beta coronavirus that causes Middle East respiratory syndrome, or MERS discovered in 2012 in Saudi Arabia).

• SARS-COV (the beta-coronavirus that causes the severe acute respiratory syndrome, SARS, identified in China in 2002)

This coronavirus has many similarities with that of SARS (animal origin, genetically identical to 80%, responsible for lung infections) but also notable differences for scientists in terms of its contagion. It is contagious from the onset of symptoms or even sometimes in the absence of symptoms, whereas Sars was a few days after the first symptoms. There are also mild and asymptomatic forms of Covid-19, while Sars only caused severe forms. [12]

1.4. Transmission and spread of Covid-19

Covid-19 is transmitted by people who carry the virus. The disease spreads mainly from person to person during close contact through respiratory droplets expelled through the nose or mouth when a sick person coughs, sneezes or talks. These droplets are relatively heavy, do not travel great distances, and fall quickly to the ground. It is possible to contract covid-19 if you inhale these droplets. [13] Any situation in which people are close to each other for long periods increases the risk of transmission. The spaces Indoors, especially when poorly ventilated, pose greater risks than outdoor spaces. Activities in which the volume of particles expelled from the mouth is greater, such as singing or breathing heavily during physical exercise, also increase the risk of transmission. [14]

Transmission is facilitated in places and situations that meet the following 3 criteria:

* Crowded spaces.

* Close contact, for example when people are talking while standing very close to each other.

* Confined and enclosed spaces poorly ventilated.

The risk of spreading Covid-19 is higher in places where these three criteria coincide. It is therefore important to also apply the rules of respiratory hygiene (for example, by covering the mouth and nose with the bend of the elbow when coughing) and to keep people at a distance to prevent infection and slow down the transmission of Covid-19.

1.5. Symptoms of Covid-19

Manifestations of coronavirus (covid-19) appear less than 24 hours after infection. Most commonly, the virus causes mild to moderate respiratory illnesses (Figure 1.2) such as the common cold with symptoms such as [15]:

1.5.1. Most common symptoms:

Fever

Dry cough

Fatigue

1.5.2. Less common symptoms:

Aches and pains

Sore throat

Diarrhea

Conjunctivitis

Headache

Loss of taste or smell

Skin rash or discoloration of fingers or toes

1.5.3. Serious symptoms:

Breathing difficulties or shortness of breath

Pain or tightness in the chest

Loss of speech or difficulty moving

Figure 1.2: General symptoms of Covid-19.

1.6. Covid-19 Mortality

The mortality rate is relatively high for SARS and MERS-CoV, with nearly 10 to 15% and more than 36% of those affected dying. For covid, the most susceptible people are those over 65 [16], those with respiratory pathologies and immunocompromised, and other health problems (hypertension, heart or lung problems, diabetes, obesity, or cancer) at higher risk, to develop a severe form of the disease. However, anyone can contract Covid-19 and become seriously ill or die.

2. COVID-19 in Algeria

2.1. History of the pandemic in Algeria

The first case, an Italian national, was notified on February 25, 2020, in a Hassi Messaoud living base in the wilaya of Ouargla. On March 2, 2020, an outbreak was detected in the wilaya of Blida following an alert launched by France after the confirmation of Covid-19 by two Algerian citizens residing in France who had stayed in Algeria. Since then, the epidemic has spread to the entire national territory. (Figure 1.3) [17]

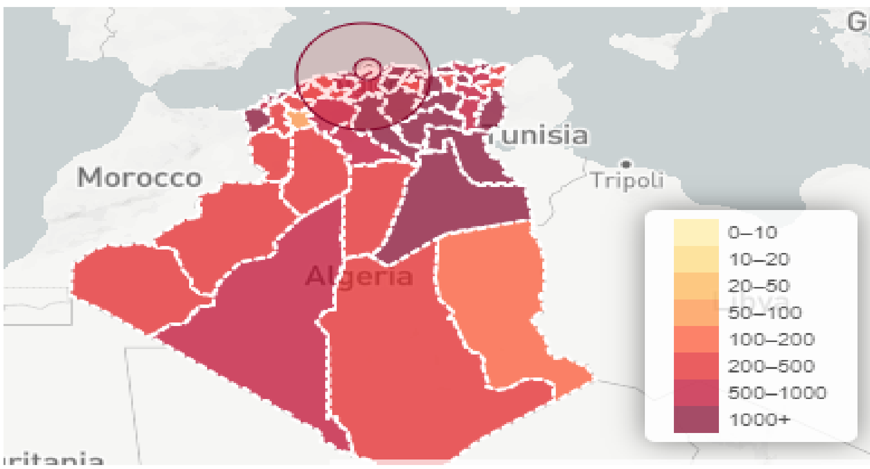


Figure 1.3: number of confirmed cases in the wilayas of Algeria.

2.2. Spread of the pandemic in Algeria

On August 8, 2020, all 48 wilayas notified confirmed cases of Covid-19 since the start of the epidemic in Algeria. To date, 265,745 cases have been identified, 178296 people have recovered and unfortunately, 6880 died. (Figure 1.4) [18]

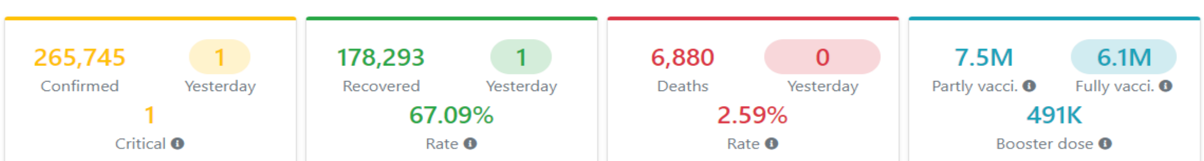


Figure 1.4: COVID-19 infection stats in Algeria

As we see in the graph (Figure 1.5), statistics of confirmed injuries, deaths, and recovery, and who is still being treated in Algeria since the beginning of the pandemic.

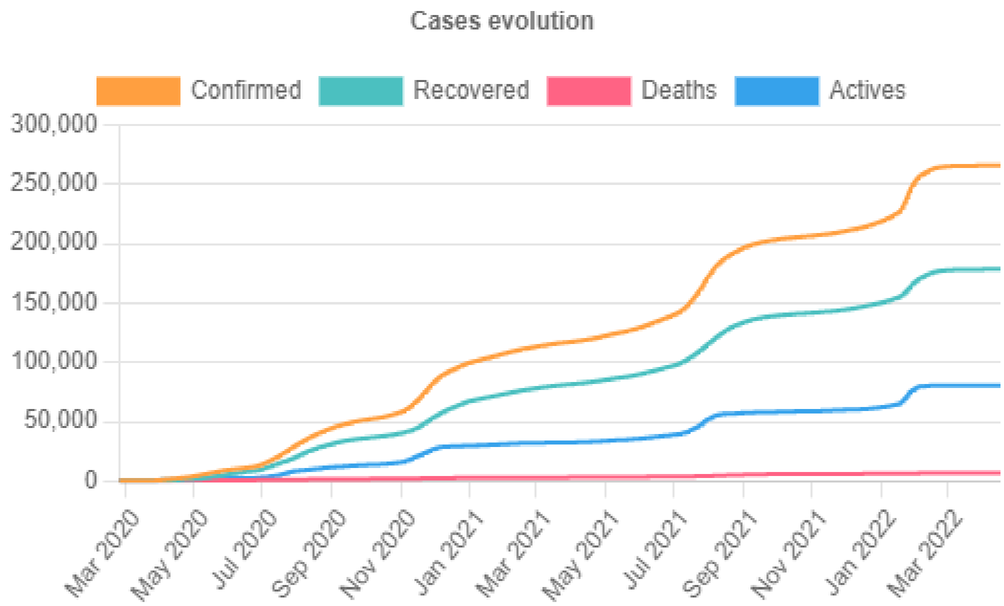


Figure 1.5: The evolution of the spread of the virus from the beginning (In Algeria)

3. Problem of COVID-19 prediction

In this study, our objective is to find a computer solution to the problem of COVID as well as to predict the contamination, in this context we have chosen to solve this problem using artificial intelligence techniques. Several techniques are offered in the literature. In the following chapter, we will introduce the most well-known ones

Conclusion

In this chapter, we have talked about this so-called Corona Virus pandemic (Covid-19) and explained where this disease comes from, its causes, and its consequences on human health, as well as its symptoms and effects. We have also seen the impact of the pandemic on our country Algeria. In the next chapter, we will present our prediction approach based on Machine Learning models.

Chapter II: Machine Learning and Related Work

Introduction

Artificial Intelligence (AI) is considered one of the most important fields of science and engineering that many researchers have been interested in. This field includes many branches the most important is Machine Learning (ML). With the rapid development of technology and the increase in data volume, the focus has increased on the field of ML which has become an important part of many applications such as image classification, natural language processing, video recommendation, text extraction, and many other applications.

In this chapter, we highlight the effectiveness of machine learning. We presented a brief reminder about AI, and the fundamentals of machine learning (ML), then we see some related work to our topic to guide us in our research.

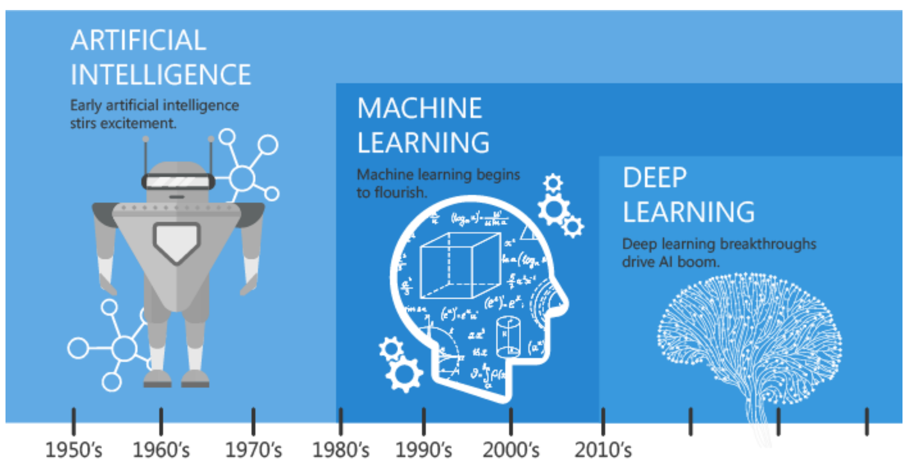


Figure 2.1: Artificial Intelligence

1. Artificial intelligence

AI is a collection of techniques that enable machines to perform tasks and solve problems usually reserved for humans. Tasks under AI are sometimes straightforward for humans, such as recognizing and locating objects in an image, planning the movements of a robot to grab an object, or driving a car. They sometimes require complex planning, for example, to play chess. The most complicated tasks require a lot of knowledge and common sense, such as, translating a text or to conduct a dialogue. In recent years, intelligence has usually been associated with learning skills. It is through learning that an intelligent system capable of performing a task can improve its performance with experience. Through learning, he will be able to learn to perform new tasks and acquire new skills. [19]

2. Machine Learning

2.1. Definition of machine learning

Machine learning ML is a branch of computer science that arose from the study of natural language processing, in artificial intelligence, pattern analysis and computational learning theory are used. A robot Predictive analytics on data is another term for learning processes. In 1959, Arthur Samuel defined ML as “a field of study that gives computers the ability to learn without being explicitly programmed”. Is concerned with designing algorithms that allow computers to have the property of learning without programming the rules for each problem. [20][21]

There are two important points in ML, which are training and expectation, that is, we train it to expect and then try it in order to expect the answers.

2.2. Element of machine learning

2.2.1. Data:

All learning methods are data-driven. Sets of data are used to train the system, These sets may be collected by humans, and maybe quite large so the data we feed our machine learning systems must be mathematical objects, such as vectors, matrices, or graphs

2.2.2. Models:

Models are often used in learning systems. A model provides a mathematical framework for learning. A model is human-derived and based on human observations and experiences

2.2.3. Training:

A system that maps an input to output needs the training to do this in a useful way. Just as people need to be trained to perform tasks, machine learning systems need to be trained. Training is accomplished by giving the system input and the corresponding output and modifying the structure (models or data) in the learning machine so that mapping is learned. In some ways, this is like curve fitting or regression. If we have enough training pairs, then the system should be able to produce correct outputs when new inputs are introduced. [22]

2.3 Types of learning

Depending on the type of available data, machine learning can be categorized into supervised learning, unsupervised learning, and reinforcement learning (Figure 2.2). Supervised and unsupervised are mostly used by a lot of machine learning engineers and data geeks, Reinforcement learning is really powerful but complex to apply.

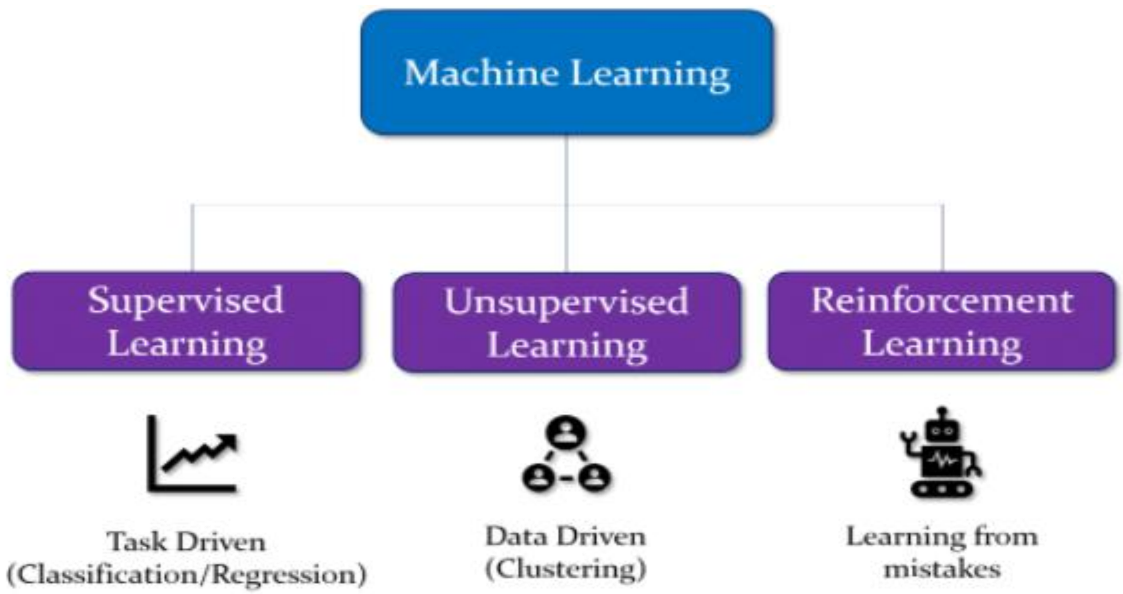


Figure 2.2: Types of learning

2.3.1. Supervised learning:

We give the computer some pairs of inputs and outputs (Training Data). We first train the model with lots of training data, then in the future when new inputs are presented you have an intelligent output. And there are two special types (algorithms) as we can see in (Figure 1.2) that are interesting to know:

2.3.1.1. Logistic regression:

Each learning is associated with a qualitative target value, which corresponds to a class. There can be two classes (binary classification) or more (multiclass classification).

2.3.1.2. Linear regression:

Each learning example is associated with a quantitative target value. The goal of the model is to estimate the correct output, given a feature vector. [23]

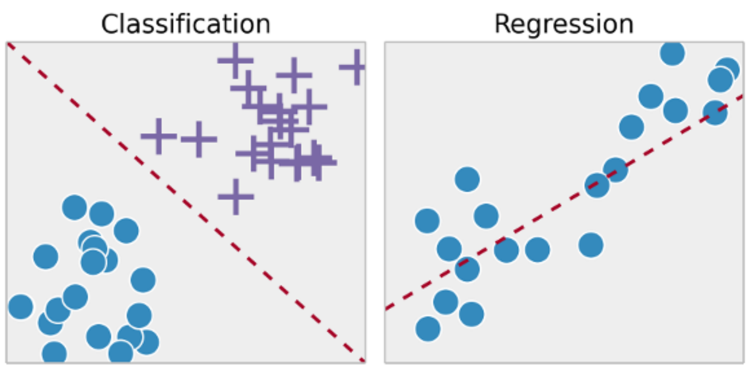


Figure 2.3: classification and regression

2.3.2. Unsupervised learning:

The training data does not include Targets here so we don't tell the system where to go. The training data is not structured (contains noisy data, unknown data, and …etc) and the system has to understand itself from the data we give, and automatically extract meaningful data. There are also different types of unsupervised learning like Clustering and anomaly detection.

2.3.1.1. Clustering:

This is a type of problem where we group similar things. A bit similar to multi-class classification but here we do not provide the labels; the system understands the data itself and clusters the data. [24]

2.3.1.2. Association:

An association rule is an unsupervised learning method, which is used for finding the relationships between variables in the large database. It determines the set of items that occurs together in the dataset. Association rule makes marketing strategy more effective. Such as people who buy X item (suppose a bread) are also tend to purchase Y (Butter/Jam) item. A typical example of Association rule is Market Basket Analysis. [25]

2.3.3. Reinforcement learning:

Reinforcement learning is the type of machine learning where the agent “machine” learns through the trial-and-error method. The agent performs actions, the environment evaluates those actions and replies with a reward or a punishment. Based on that evaluation, the agent will determine if that action is considered the most accurate.

2.4 Machine Learning Algorithms

2.4.1. Logistic Regression (LR):

It is a supervised classification algorithm used to classify data into a set of discrete classes (identified classes) by returning probability values (between 0 and 1) using sigmoid/logit functions. Logistic regression uses a one-to-many approach to classify data when many classes need to be assigned. [26]

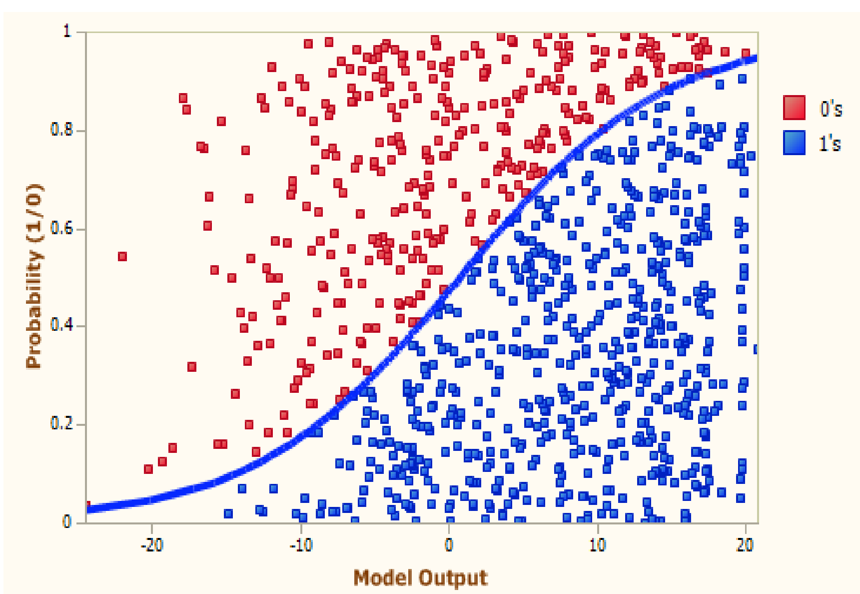


Figure 2.4: Logistic Regression example

2.4.2. Support Victor Machine (SVM):

Encouragement vector machines are one of the most famous and most discussed machine learning techniques. It is essentially a more refined approach. Constructing and performing classification tasks Hyperplanes are planes that are parallel to each other in a multidimensional space. Cases with different class names are separated. SVM (System Variable Modeling). [27]

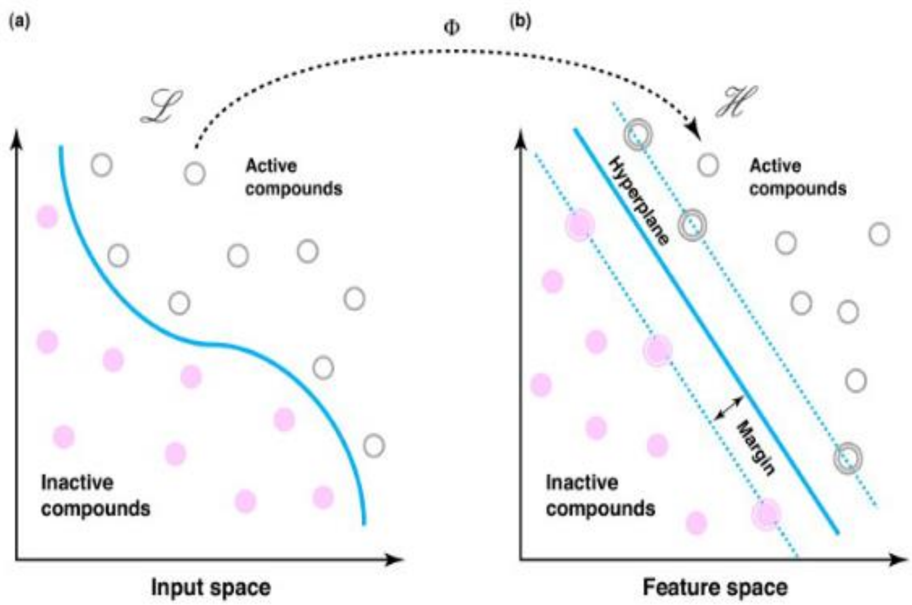


Figure 2.5: Support Victor Machine example

2.4.3. Naïve Bayes (Nb):

A naive Bayes classifier is not a single algorithm, but a family of machine learning algorithms which use probability theory to classify data with an assumption of independence between predictors It is easy to build and particularly useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods. [27]

2.4.4. Decision Tree:

The decision tree algorithm classifies objects by answering the "questions" placed in the nodes about their attributes. Depending on the answer, one of the branches is chosen, and another question is asked at the next intersection until the algorithm reaches the "leaf" of the tree, which represents the final answer. Decision Tree applications include a knowledge management platform for customer service, predictive pricing, and product planning. A typical example of a decision tree is determining the insurance premium that should be charged based on a person's circumstances. Decision trees can define complex standard maps such as location, claim type, environmental conditions, etc., and determine risk categories based on submitted claims and amounts spent. The system can then score new insurance claims and categorize them by risk level and potential financial loss. [28]

2.4.5. Random Forest:

Random Forest is a powerful machine learning algorithm. In an ensemble of machine learning algorithms called guided aggregation or bagging. Bootstrap is a Bootstrap is apowerful statistical method for estimating an ensemble from a sample of data. Like an average. You take many samples from the data, compute the mean, and then average all the means to get a better estimate of the true mean. Bagging uses the same method, but for estimating entire statistical models, most commonly decision trees. Take multiple samples of training data, and build a model for each data sample. When you need to predict new data, each model makes a prediction, and the predictions are averaged to better estimate the actual output value. Random forests are optimization of this approach, building decision trees for suboptimal splits by introducing randomness rather than choosing the best split point. As a result, the model created for each data sample is more diverse than the original, but still accurate in its unique and different ways. Combining their predictions gives a better estimate of the true potential output value. If you get a good result estimate of the true potential output value. If you get a good result. [28]

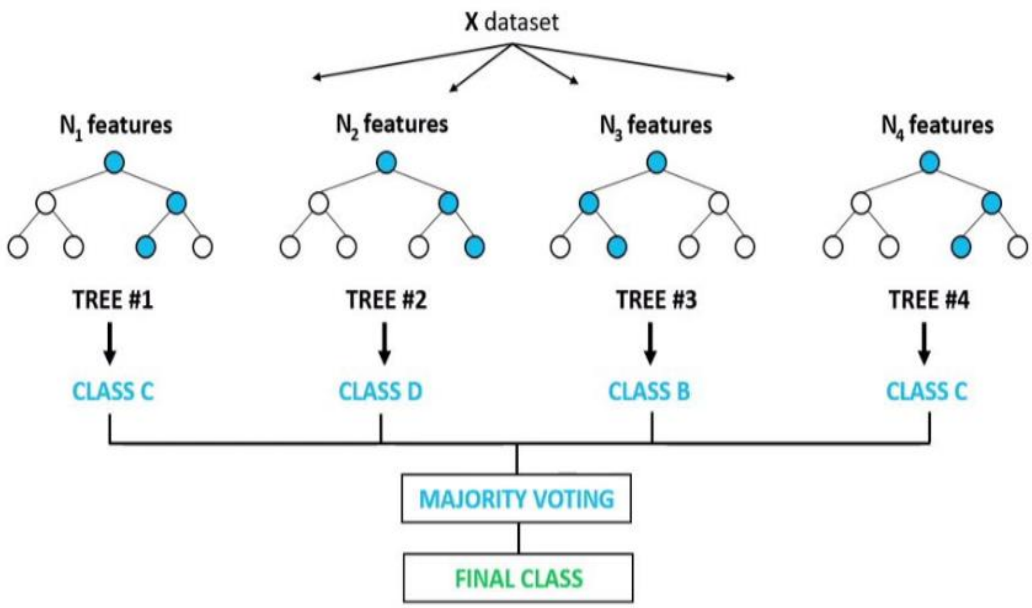


Figure 2.6: Random Forest

3. Related work:

Researchers are generating an unprecedented amount of data around COVID-19, with new and important studies coming out every day. Here's a look at some of COVID-19 research;

- The ANN-based models were utilized to estimate the confirmed cases of COVID-19 in China, Japan, Singapore, Iran, Italy, South Africa and United States of America. These models exploit historical records of confirmed cases, while their main difference is the number of days that they assume to have impact on the estimation process. The COVID-19 data were divided into a train part and a test part. The former was used to train the ANN models, while the latter was utilized to compare the purposes. The data analysis shows not only significant fluctuations in the daily confirmed cases but also different ranges of total confirmed cases observed in the time interval considered.[29]

- The objective of this study is to predict a time series of future cumulative totals of COVID-19 cases, using characteristics such as geographic location and number of countries/regions, analyzing and processing by the Python programming language. The prediction model deploys machine learning, linear regression, and support vector machine techniques on a database collected from around the world. The result illustrates an upward trend for a ten-day review but takes two distinct paths when examining the conformity of predicted cases to actual cases.[30]
- They demonstrates the capability of ML models to forecast the number of upcoming patients affected by COVID-19 which is presently considered a potential threat to mankind. In particular, four standard forecasting models, such as linear regression (LR), least absolute shrinkage and selection operator (LASSO), support vector machine (SVM), and exponential smoothing (ES) have been used in this study to forecast the threatening factors of COVID-19. Three types of predictions are made by each of the models[31]
- A deep learning model has been built by considering the features of weather and COVID-19 data (recovered, infected, and deceased) for predicting the number of cases expected in India. The model is built on Concurrent Neural Network (CNN), Recurrent Neural Network (RNN), bi-directional RNN (BRNN), Long Short-Term Memory (LSTM), and Bi-directional LSTM (BLSTM) based on the daily weather and COVID-19 data collected from the Indian subcontinent. The results revealed that the algorithm BRNN yields a better prediction model when compared with the other models.[32]
- They are predicting and forecasting the COVID-19 outbreak in India based on the machine learning approach, where they aim to determine the optimal regression model for an in-depth analysis of the novel coronavirus in India. They are implementing the two regression models namely linear and polynomial and evaluating the two using the R squared score and error values. The COVID-19 dataset for India is being used to serve the research of this paper. The model is predicting the number of confirmed, recovered, and death cases based on the data available from March 12 to October 31, 2020. For forecasting the future trend of these cases, they are utilizing the time series forecasting approach of tableau. Furthermore, the time series forecasting method is being employed to forecast the total count of confirmed cases in the future.[33]

	Author	Date	Dataset	Country	Method used
1	-Majid Niazkar -Hamid Reza Niazkar	26 April 2020	confirmed cases of the Republic of China, Japan, Singapore, Iran, Italy, South Africa and USA(WHO)	Iran	artificial neural networks(ANN)
2	-Boudouani Lilia -Gacem Rania	2020	Kaggel (Italy, India, USA, China...)	Algeria	Linear Regression
3	-Furqan Rustam -Aijaz Ahmad Reshi -Arif Mehmood -Saleem Ullah -Byung-Won -Waqar Aslam -Gyu Sang Choi	May 2020	GitHub repository provided by the Center for Systems Science and Engineering, Johns Hopkins University (Worldwide confirmed cases, deaths, and recoveries)	South Korea	-Linear Regression -LASSO Regression -Support Vector Machine -Exponential Smoothing
4	-A. Ronald Doni -T. Sasi Praba -S. Murugan	03 June 2021	Indian subcontinent	India	-Neural Network (CNN), -Recurrent Neural Network (RNN) - bi-directional RNN (BRNN), -Long Short-Term Memory (LSTM) - Bi-directional LSTM (BLSTM)
5	-Saud Shaikh -Jaini Gala -Aishita Jain -Sunny Advani -Sagar Jaidhara -Mani Roja Edinburgh	15 March 2021	Covid-19 outbreak in India all confirmed cases, deaths, and recoveries	India	Linear Regression

Table 1: Related works

In our context, we will use neural networks to build our prediction model.

Conclusion

In this chapter, we have seen the theoretical part of artificial intelligence and machine learning and some related work to our context that helped us decide which machine learning method to use in our prediction model

In the next chapter, we will explain deep learning to know more about our used method and our implementation and experimentation phase.

Chapter III: Deep Learning, Implementations, and experimentations

Introduction

Simple machine learning algorithms work very well on a wide variety of important problems. However, they have not succeeded in solving central problems in AI. However, there is a subfield of machine learning deep learning that is entirely concerned with algorithms inspired by the structure and function of artificial neural networks, which are inspired by the human brain. In this chapter, we will detail deep learning and some of its neural networks, especially the one that we will use in our prediction model. Long short-term memory (LSTM) which is a recurrent neural network (RNN), then we will start the implementation and experimental part.

1. Deep learning

1.1 Artificial Neural networks

Artificial Neural Networks (ANNs) mimic the structure of the brain and the way it analyzes and processes data; each artificial neuron connects to several other neurons, and together millions of neurons create a complex cognitive structure. NNs have a multilayer structure: neurons in one layer transmit data to several neurons in the next, and so on. Ultimately, the data reaches the output layer, where the network decides how to solve a problem, classify an object, etc. Due to the multi-layer nature of neural networks, a new field of study appeared known as “Deep learning” specialized in these multi-layer neural networks. [34]

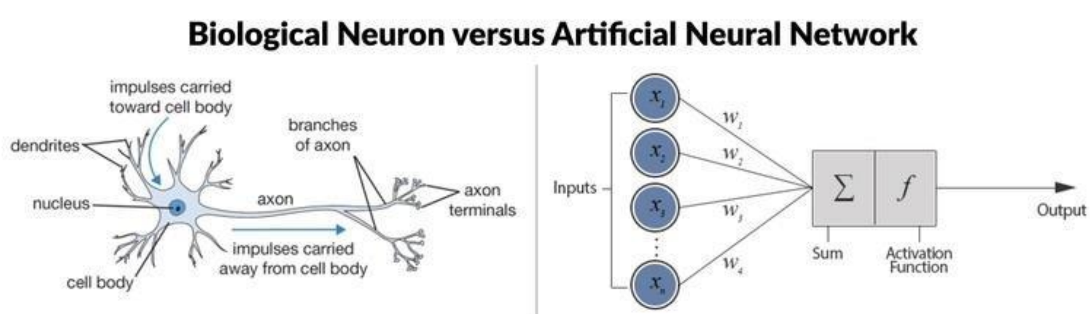


Figure 3.1: Neural networks

1.1.1 Activation Function

The activation function is an important feature of neural networks. It determines whether a neuron whether it should be activated, and whether the information received by the neuron is consistent with the given information or it should be ignored.

$Y = \text{Activation}\ (\text{Sum}\ ((\text{weight} * \text{input}) + \text{bias})))$ Where Y = result of the activation function

The activation function is a nonlinear transformation that is applied to the input , Then the converted output is sent as input to the next layer of neurons .

1.1.2 Types of activation functions

1.1.2.1 Binary Step Function

The binary step function is athreshold-basedclassifier, so if the input to the activation function is greater than a threshold, then the neuron is activated, else it is deactivated

$F(x) = 1, x>=0$

$F(x) = 0, x<0$

1.1.2.2 Linear Function

In the previous function (the binary step function), the gradient being zero, so it was impossible to refresh gradient during the back propagation. Alternatively, of a simple step function, we can try using a linear function F(x) =ax 1.4.3. Sigmoid Function It is one of the most largely used non-linear activation function. Sigmoid transforms the values between the range 0 and 1. Here is the mathematical expression for sigmoid: $F(x) = 1/ (1+e\text{^}{-x})$

1.1.2.3 Tanh Function

The tanh function is very close to the sigmoid function. It is actually just a proportioned version of the sigmoid function, so the mathematical expression would like this: tanh(x)=2sigmoid(2x)-1 so the final result : $\tanh(x) = 2\ (1 + e\ {-2x}) - 1$

1.1.2.4 ReLU Function

ReLU function defined by: f(x)=max(0,x) ReLU stands for Rectified Linear Unit. The fundamental benefit of using the ReLU function over other activation functions is that it does not activate all the neurons simultaneously. This means that the neurons will only be deactivated if the output of the linear tra

1.1.2.5 Leaky ReLU

Leaky ReLU function is nothing but an improved version of the ReLU function. As we saw that for the ReLU function, the gradient is 0 for x ,which would deactivate the neurons in that region. LeakyReLU is defined to address this problem. Instead of defining the Relu function as 0 for negative values of x, we define it as an extremely small linear component of x. Here is the mathematical expression:

$f(x)= 0.01x , x=0 , f(x)= x , x>=0$

1.2 Evolution of Deep learning

Warren McCulloch and Walter Pitts developed the earliest neural networks in the 1940s, shortly after AI research began. In 1943, the seminal work entitled "Logical Calculus of Inner Thoughts in Neural Activity" was published, proposing the first mathematical model of neural networks. The unit of this model is a simple formalized neuron, commonly referred to as a McCulloch-Pitts neuron. It is a mathematical function designed as a biological neuron model, a neural network. Since then, the concept of neural networks has developed, and with it [35]. In the 1960s, deep learning and multi-layer neural networks emerged, over time; several types of these networks appeared, with one each having its own advantages, characteristics, and shortcomings.

- Figure3.2: illustrates a timeline showing the evolution of deep models.

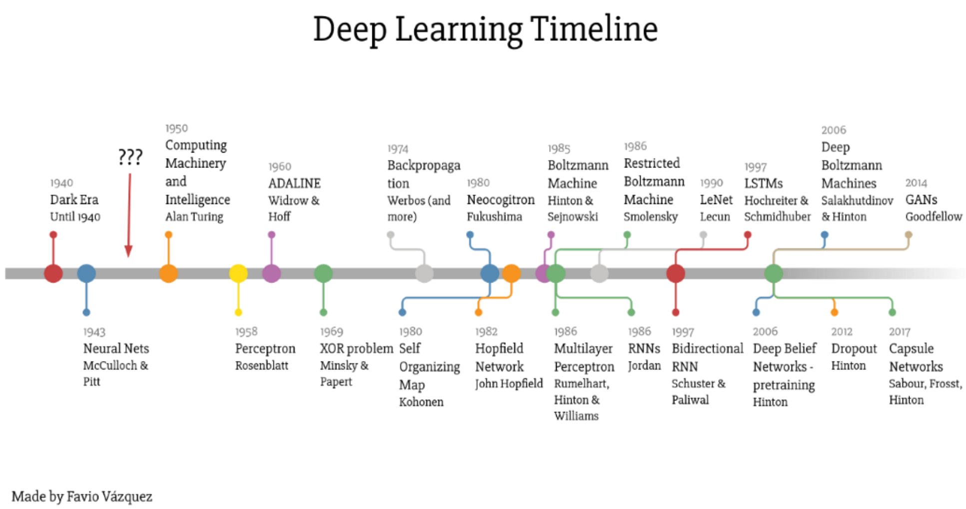


Figure 3.2: DL timeline

- Figure3.3: comparing the performance of traditional machine learning algorithms and deep learning algorithms. The performance of traditional machine learning algorithms becomes stable when it reaches the threshold of training data whereas deep learning upturns its performance with the increased amount of data.

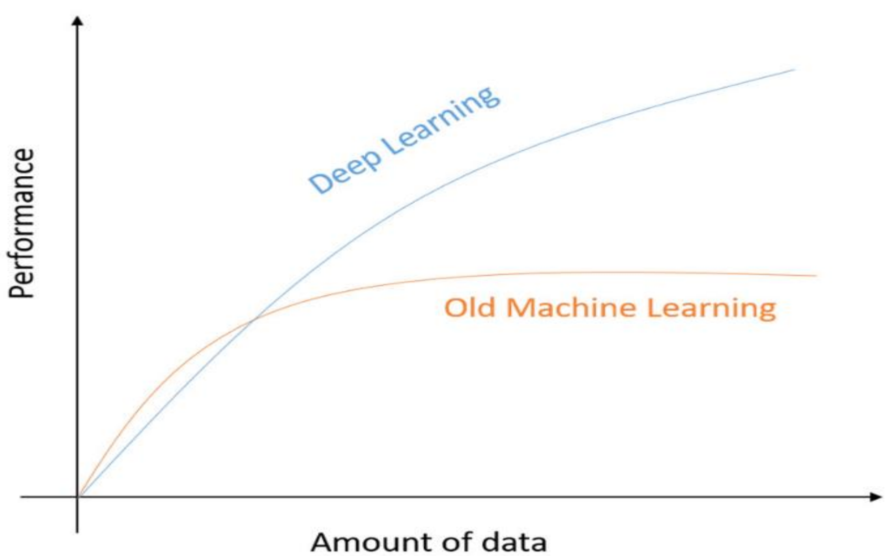


Figure3.3: deep learning performance

1.3 Definition of Deep learning

Deep Learning allows models composed of multiple layers of processing to learn representations of data with multiple levels of abstraction. [36] According to the work of Deng L et al [36], "Deep Learning" is a class of automatic learning techniques belonging to the field of "Machine Learning" in which multiple layers of iterative computational processing in hierarchical architectures are exploited for unsupervised learning algorithms for analysis and classification tasks. Deep learning essentially consists of calculating hierarchical characteristics of the parameters of artificial neural networks for vector representations of observation or input data. The family of deep learning methods is becoming increasingly rich, encompassing those of neural networks, hierarchical probabilistic models, as well as many supervised and unsupervised feature-learning algorithms.

1.4 Applications of Deep Learning

Deep learning technology is one of the most used techniques in many areas, including [37]

1.4.1. Virtual Assistants:

Virtual Assistants are cloud-based applications that understand natural language voice commands and complete tasks for the user. Amazon Alexa, Cortana, Siri, and Google Assistant are typical examples of virtual assistants. They need an internet-connected device to work with their full capabilities. Each time a command is fed to the assistant, they tend to provide a better user experience based on experiences using Deep Learning algorithms.

1.4.2. Chatbots:

Chatbots can solve customer problems in seconds. A chatbot is an AI application to chat online via text or text-to-speech. It is capable of communicating and performing actions similar to a human. Chatbots are used a lot in customer interaction, marketing on social network sites, and instant messaging the client. It delivers automated responses to user inputs. It uses machine learning and deep learning algorithms to generate different types of reactions.

1.4.3. Entertainment:

Companies such as Netflix, Amazon, YouTube, and Spotify give relevant movies, songs, and video recommendations to enhance their customer experience. This is all thanks to Deep Learning. Based on a person's browsing history, interest, and behavior, online streaming companies give suggestions to help them make product and service choices. Deep learning techniques are also used to add sound to silent movies and generate subtitles automatically.

1.4.4. News Aggregation and Fake News Detection:

Deep Learning allows you to customize news depending on the readers' persona. You can aggregate and filter out news information as per social, geographical, and economic parameters and the individual preferences of a reader. Neural Networks help develop classifiers that can detect fake and biased news and remove it from your feed. They also warn you of possible privacy breaches.

1.4.5. Composing Music:

A machine can learn the notes, structures, and patterns of music and start producing music independently. Deep Learning-based generative models such as WaveNet can be used to develop raw audio. Long Short Term Memory Network helps to generate music automatically. Music21 Python toolkit is used for computer-aided musicology. It allows us to train a system to develop music by teaching music theory fundamentals, generating music samples, and studying music.

1.4.6. Image Coloring:

Image colorization has seen significant advancements using Deep Learning. Image colorization is taking an input of a grayscale image and then producing an output of a colorized image. ChromaGAN is an example of a picture colorization model. A generative network is framed in an adversarial model that learns to colorize by incorporating a perceptual and semantic understanding of both class distributions and color.

1.4.7. Robotics:

Deep learning is heavily used to build robots that perform human-like tasks. Robots powered by deep learning use real-time updates to detect obstacles in their path and immediately plan their journey ahead of time. Can be used for cargo transportation in hospitals, factories, warehouses, inventory management, manufactured products, etc.

1.4.8. Image Captioning:

Image captions are methods for generating textual descriptions of images. It uses computer vision to understand the content of an image and a language model to convert the understanding of the image into words in the correct order.

1.4.9. Healthcare:

Deep learning has applications in healthcare. Deep learning enables computer-aided disease detection and computer-aided diagnosis. It is widely used in medical research, drug research, and diagnosing life-threatening diseases such as cancer and diabetic retinopathy through medical imaging processes.

*Like our approach, we can see that in the case of 'Covid-19', predictions using deep learning can aid healthcare by predicting future cases, which can help policymakers make appropriate decisions, to stop the spread of the COVID-19 pandemic.

1.5 Deep learning methods

Some of the powerful techniques that can be applied to deep learning algorithms to reduce the training time and to optimize the model:

1.5.1 Back propagation

While solving an optimization problem using a gradient-based method, backpropagation can be used to calculate the gradient of the function for each iteration [38].

1.5.2 Stochastic Gradient Descent

Using the convex function in gradient descent algorithms ensures finding an optimal minimum without being trapped in a local minimum. Depending upon the values of the function and learning rate or step size, it may arrive at the optimum value in different paths and manners [39]

1.5.3 Learning Rate Decay

Adjusting the learning rate increases the performance and reduces the training time of stochastic gradient descent algorithms. The widely used technique is to reduce the learning rate gradually, in which we can make large changes at the beginning and then reduce the learning rate gradually in the training process. This allows fine-tuning the weights in the later stages [40].

1.5.4 Dropout

The overfitting problem in deep neural networks can be addressed using the drop out technique. This method is applied by randomly dropping units and their connections during training. Dropout offers an effective regularization method to reduce overfitting and improve generalization error. Dropout gives an improved performance on supervised learning tasks in computer vision, computational biology, document classification, speech recognition [41].

1.5.5 Max-Pooling

In max-pooling a filter is predefined, and this filter is then applied across the nonoverlapping sub-regions of the input taking the max of the values contained in the window as the output. Dimensionality, as well as the computational cost to learn several parameters, can be reduced using max-pooling [42].

1.5.6 Batch Normalization

Batch normalization reduces covariate shift, thereby accelerating deep neural network. It normalizes the inputs to a layer, for each mini-batch, when the weights are updated during the training. Normalization stabilizes learning and reduces the training epochs. The stability of a neural network can be increased by normalizing the output from the previous activation layer [43].

1.5.7 Skip-gram

Word embedding algorithms can be modeled using Skip-gram. In the skip-gram model, two vocabulary terms share a similar context; then those terms are identical. For example, the sentences “cats are mammals” and “dogs are mammals” are meaningful sentences, which shares the same meaning “are mammals.” Skip-gram can be implemented by considering a context window containing n terms and train the neural network by skipping one of this term and then use the model to predict skipped term [44].

1.5.3 Transfer learning

In transfer learning, a model trained on a particular task is exploited on another related task. The knowledge obtained while solving a particular problem can be transferred to another network, which is to be trained on a related problem. This allows for rapid progress and enhanced performance while solving the second problem [45].

1.6 Deep neural networks:

A deep neural network (DNN) is an artificial neural network (ANN) with multiple layers between the input and output layers. There are various types of deep neural networks, each of which has a specific use case. From one point of view, they can be categorized into supervised or unsupervised but they always consist of the same components: neurons, synapses, weights, biases, and functions. In this section, we will present a brief overview of the common structures found in many deep networks.

1.5.1 Convolutional neural networks (CNN):

Convolutional neural networks (CNN) are one of the most famous deep neural networks and the most widely used, especially in the field of computer vision, where its structure simulates the way the visual cortex works in the cat's brain [46]. It is designed to work with grid-structured inputs, which have strong spatial dependencies in local regions of the grid [47]. In another word, the input is multiple arrays; for example, a color image composed of three 2D arrays containing pixel intensities in the three color channels [48]. Despite the focus of CNNs on image data processing, we can also consider sequential data such as texts as a special case of grid structured data and use CNN to process it [47].

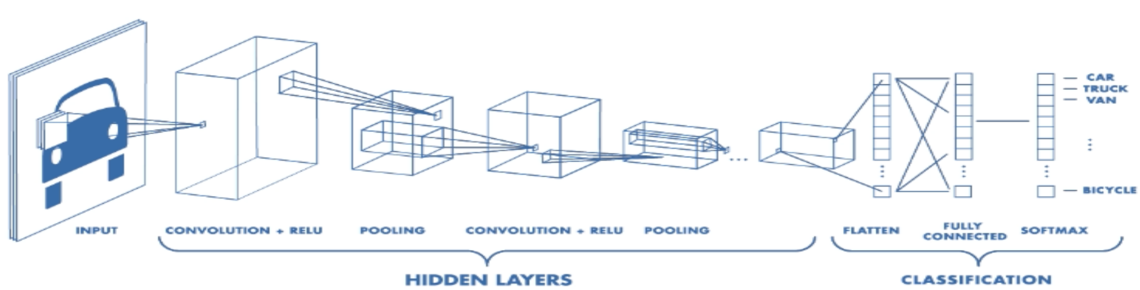


Figure 3.4: Convolutional Neural Network [49]

1.5.2 Deep belief network (DBN):

DBN is described as a combination of many constrained Boltzmann machines (RBM), with two feature recognition unit layers. RBM is also a special Markov random field. RBM is a generative random artificial neural network that can do this Learn a probability distribution from its input dataset. Can solve search problems in parameter space for deep architectures. DBN is one of the hottest topics in the field of neural networks. It has shown higher accuracy in recent years than some existing well-known image recognition, speech recognition, handwriting recognition, and other classification problems. Different from traditional flat learning One of the most important features of network DBN is that it makes it easier to identify networks in Deep Mode, which improves thinking skills and allows people to master the depth Difference between normal data and error data. [50]

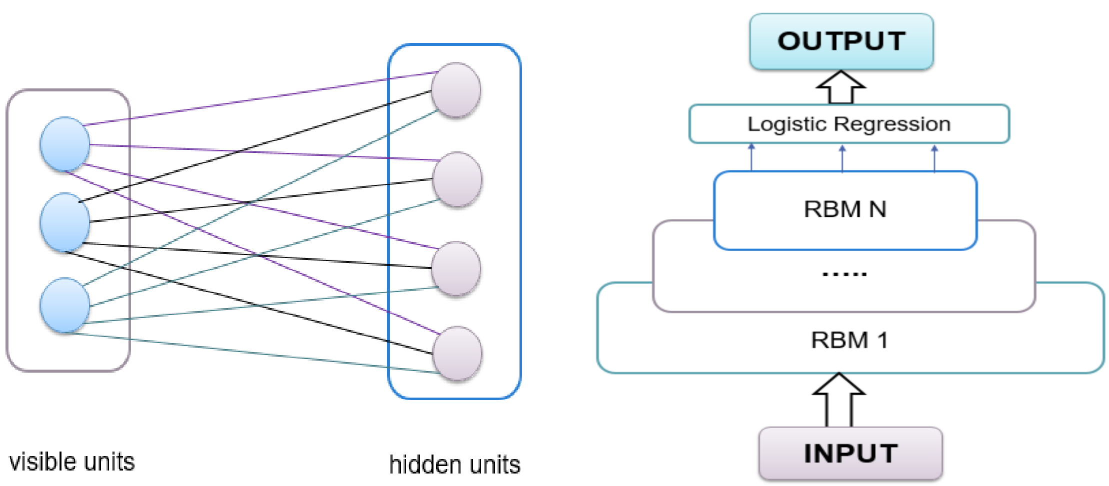


Figure 3.5: restricted Boltzmann machines and Belief Neural Network

1.5.3 Recurrent neural network (RNN):

Recurrent Neural Networks (RNN) are among the most famous deep neural networks that simulate repetitive brain structure. They have been proposed in the 80s and specialize in the sequential domain. RNN is characterized by its unique structure, which is different from the rest of the networks where it has feedback connections [51], which makes it one of the best ways to solve problems that need previous knowledge or experience such as prediction and production [52]; it is also used in cases where the input has a variable of unfixed length. RNNs process the entry sequence each time while keeping the state vector in its hidden units since the latter contains information about the history of all previous elements of the sequence. [48]

* Following this general view of deep learning, we will detail the method chosen in our research work, which is the recurrent neural network long short-term memory (LSTM).

1.7 Recurrent neural network (RNN)

RNN were created because there were a few issues in the feed-forward neural network:

- Cannot handle sequential data
- Considers only the current input
- Cannot memorize previous inputs

The solution to these issues is the RNN. An RNN can handle sequential data, accepting the current input data, and previously received inputs. RNNs can memorize previous inputs due to their internal memory. [53]

Figure 3.6: Recurrent Neural Networks work

As shown in Figure 3.7 the output of the hidden layer is its input and this operation is repeated for a specific times in different time steps. The weights are shared at each time step.

The hidden state at time t is given by a function of the input state at time t and the hidden state at time t-1 [47]: $h_t = f(h_{t-1}, x_t)$ (1)

Note that we have input hidden weight W_{xh} and hidden layer hidden weight W_{hh} and the resulted output hidden weight W_{yh} and by using tanh as a function the equation became:

$$h_t = \tanh(W_{hh}.h_t, W_{wh}.x_t) \qquad (2)$$

And the output is: $y_t = W_{yh}.h_t$ (3)

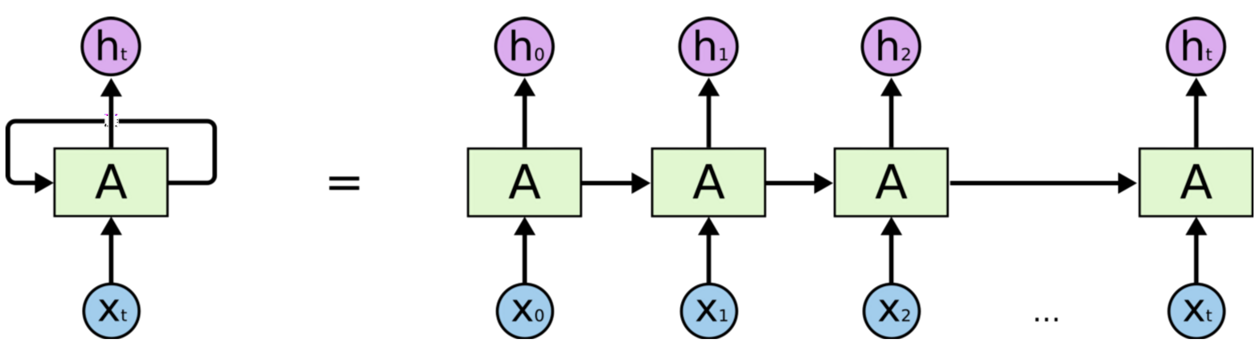


Figure 3.7: Recurrent Neural Network structure [54]

1.7.1 Types of Recurrent Neural Networks:

RNN can process inputs of different lengths and generate outputs of different lengths it has different representations:

- One to one: Used for fixed size input to fixed size output and it is appropriate for image classification.
- One to many: The output is a sequence and it is used for image captioning or music generation.
- Many to one: The input is a sequence and it is used for sentiment analysis or Classification
- Many to many: There is two types, the first has a sequence input and used in machine translation where the second has a synchronized sequence input and output and it is used for video classification.

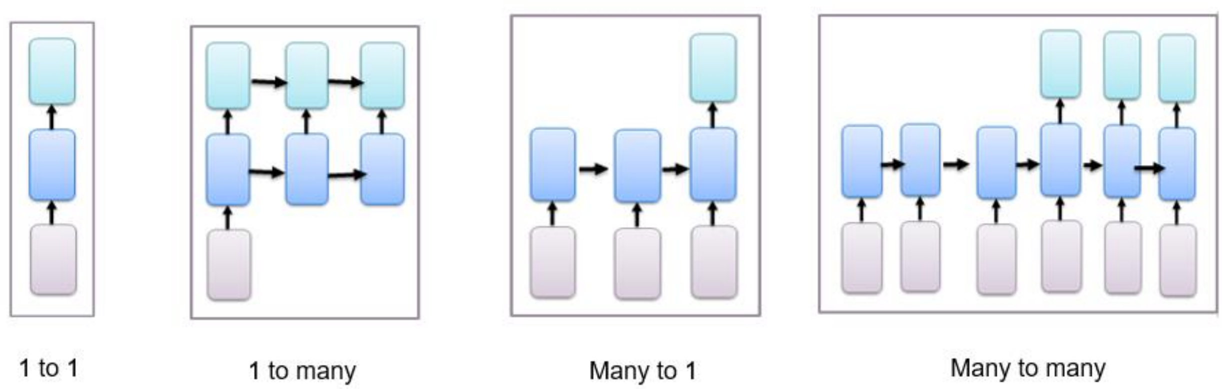


Figure 3.8: RNN representations [55]

1.7.2 Backpropagation Through Time:

Backpropagation Throw Time (BPTT) is an updated version of Back Propagation for RNNs which update weights throw time. BPTT unfold the RNN in time to create an equivalent feedforward every time a sequence is processed which make the derivatives calculable via standard BP [51]. BPTT performs gradient descent on a complete unfolded network. Suppose that the training starts at time t0 and ends at time t1, the total cost function is the sum over time of the standard error function $E_{sse/ce}$(t) at each time-step [52]:

$$E_{total}(t_0, t_1) = \sum_{t=t0}^{t1} E_{\frac{sse}{ce}}(t) \qquad (4)$$

And the gradient descent weight updates each time-step (5):

$$\Delta W_{ij} = -\eta \frac{\partial E_{total}(t_0, t_1)}{\partial W_{ij}} = -\eta \sum_{t=t0}^{t1} \frac{\partial E_{\frac{sse}{ce}}(t)}{\partial W_{ij}} \qquad (5)$$

The partial derivatives $\partial E_{sse/ce}/\partial W_{ij}$have contributions from each weight $W_{ij} \in \{W_{ih}, W_{hh}\}$ and depend on the inputs and hidden unit activation at previous time steps, after that the error back-propagate throw time and network. [52]

1.7.3 RNN Problems:

We can look at RNN as very deep feedforward network, where all layers share the same weights and as we mentioned earlier, they learn long-term dependencies and this produces a learning problem where it is difficult to store information for a long. [48]

While learning the RNN the gradients keep changing at each time step, they may grow or shrink which cause an exploding gradients problem or vanishing gradients problem.

Exploding gradients problem: A problem happens when training the RNN with the backpropagation. The shared weights are the same, so the gradient is multiplied with the same quantity resulting an exploding in the gradient when the weight w is bigger than one. [47]

Vanishing gradients problem: Is a problem that happens when the shared weight w is less than one which makes the gradient keep shrinking throw time. [47]

Solutions: There is different solutions [47][56]:

• We can change the used activation function because its derivation is included in the product while calculating the gradient, an activation function such as ReLU may be more helpful then tanh and segmoid.

• For the exploding gradient problem we can use the Gradient Clipping which prevent gradients from exploding by rescaling them. It means capping the maximum value for the gradient.

• We can use also a technique called Input Reversal which means reversing the input words order when the gaps between the input and the output are very long.

• Another solution proposed where the structure of RNN have been improved with an architecture called short-term long-term memory, denoted by LSTM.

1.7.4 Some of RNN Architectures:

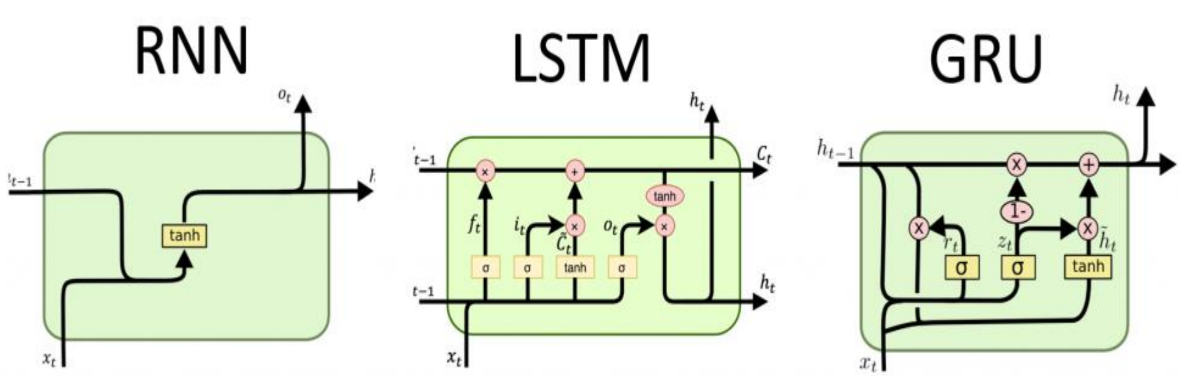


Figure 3.9: RNN & LSTM & GRU

1.7.4.1 LTSMs:

LSTMs are a special kind of RNN — capable of learning long-term dependencies by remembering information for long periods is the default behavior. All RNNs are in the form of a chain of repeating modules of a neural network. In standard RNNs, this repeating module will have a very simple structure, such as a single tanh layer. [53]

1.7.4.2 Bidirectional RNN:

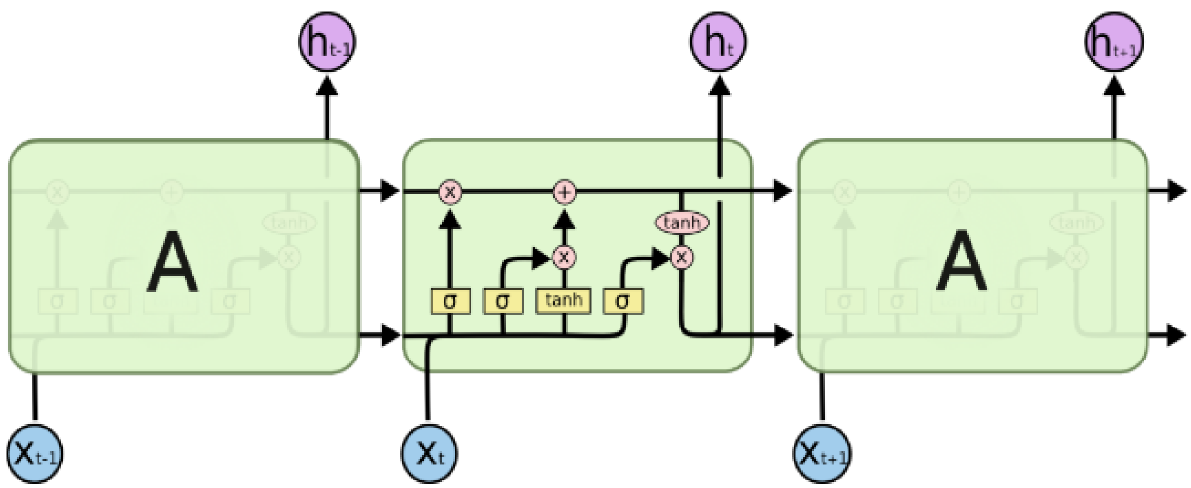
As mentioned earlier, RNN have knowledge of previous entries, but this knowledge is up to a certain point, and so does not have information about future cases. In the bidirectional recurrent network, we have separate hidden states h (t) and g (t) for the forward and backward directions as shown in Figure 1.16. The forward states interact in the forwards direction, while the backwards states interact in the backwards direction. Both h (t) and g (t) receive input from the same vector x (t) and they interact with the same output vector o (t). [48]

1.7.4.3 Gated Recurrent Units:

Gated Recurrent Units (GRU) is an updated version of LSTM, where the forget and input gates are replaced with an update gate zt , and a reset gate rt is introduced to modify the ht−1, and the internal memory Ct is eliminated. [48]

1.8 Long short-term memory (LSTM)

Long Short Term Memory (LSTM) is an update of RNN introduced by Hochreiter and Jürgen Schmidhuber and Bengio [57], in order to solve the exploding/vanishing gradient problems, where the basic idea was to modify RNN to allow a better flow to error derivations. LSTM was designed to transmit important information many time steps into the future in order to learn and remember long-term dependencies. The key component of LSTM is the memory cell, which holds the information that it has learned over time until it's needed. The activations correspond to short-term memory while the weights correspond to long-term memory [56] [58]. LSTM maintains the state of the cell as well as carrying weights to ensure that the signal and information are not lost during sequence processing. In addition, it can learn how to make a multi-step prediction from one-step and it is useful for predicting a time series.



The repeating module in an LSTM contains four interacting layers.

Figure 3.10: LSTM

1.8.1 LSTMs work:

The long term memory is called the cell state, this cell can be modified by the forget gate, the previous cell state is multiplied with the forget gate, and then adds the new information comes from the input gates, the forget gate decides which information to forget by multiplying a specific position in the matrix with 0, it uses a sigmoid function if the output is 1 the information is kept in the cell state, and the modification of a cell state based on two functions, sigmoid for update and tanh to create a vector, with the cell state updated, the output is corresponding to the updated cell state, choosing a fraction of cell to go through a tanh function to be the output, the change in the state cell effects the state passed from a time unit to the next.

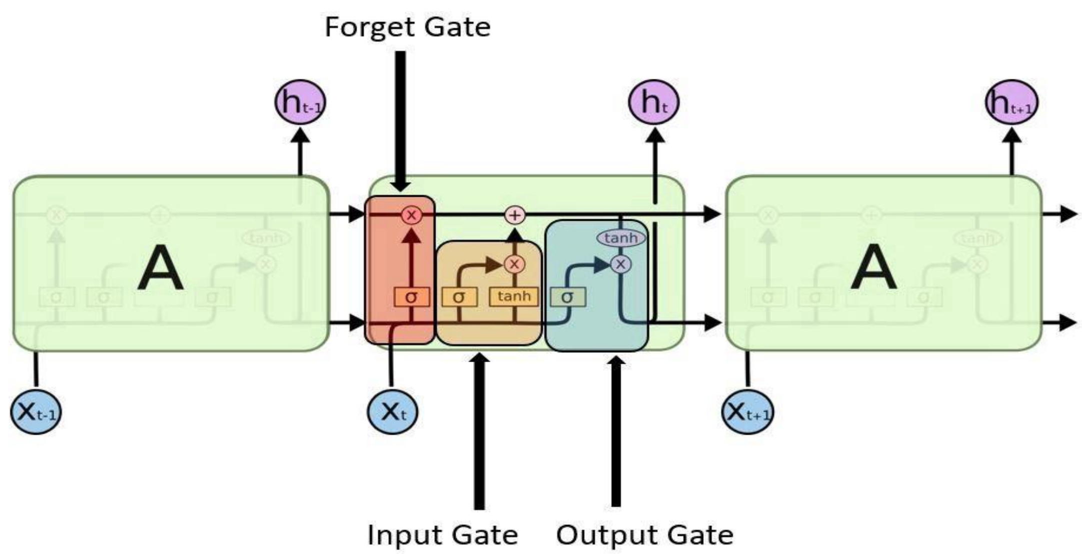
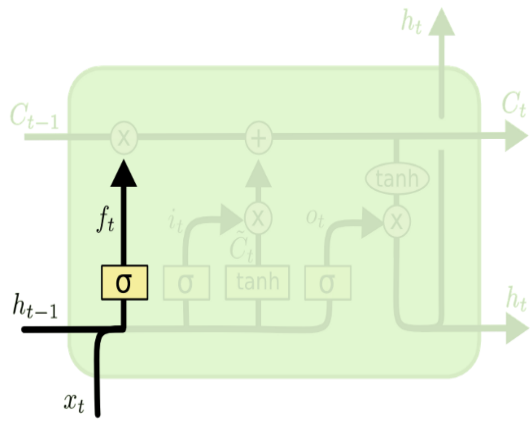


Figure 3.11: LSTM gates

We can resume LSTM work in three steps:

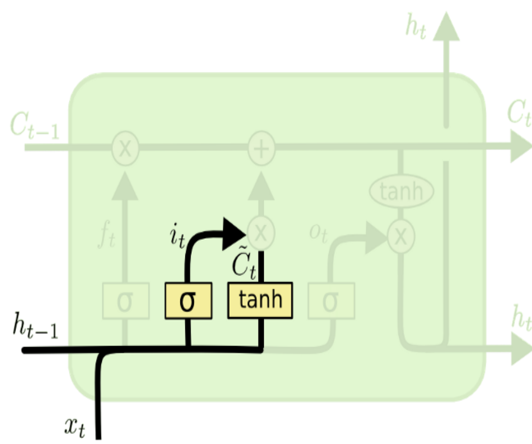
1. The first step in the LSTM is to decide which information should be omitted from the cell in that particular time step. The sigmoid function determines this. It looks at the previous state (ht-1) along with the current input xt and computes the function.



$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \ + \ b_f\right)$$

Figure 3.12: the forget gate

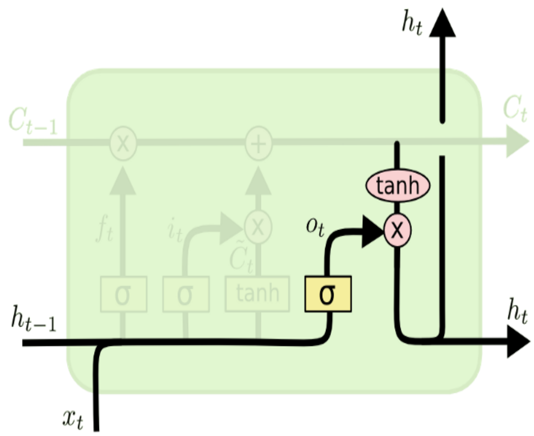
2. The second layer has two parts. One is the sigmoid function and the other is the tanh function. In the sigmoid function, it decides which values (0 or 1) to pass. The tanh function weights the passed values and decides their importance (-1 to 1).



$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \ + \ b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \ + \ b_C)$$

Figure 3.13: the input gate

3. The third step is to decide what the output is. First, we run a sigmoid layer that decides which parts of the cell state go into the output. Then we set the cell states by tanh to have values between -1 and 1 and multiply them by the output of the sigmoid gate.



$$o_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right)$$

$$h_t = o_t * \tanh\left(C_t\right)$$

Figure 3.14: the output gate

1.8.2 LSTM Use Case:

A practical implementation to predict the prices of stocks using the “Google stock price” data. Based on the stock price data between 2012 and 2016, we will predict the stock prices of 2017. [53]

1. Add the LSTM layer

```
regressor.add(LSTM(units = 50, return_sequences = True, input_shape = (X_train.shape[1], 1)))
regressor.add(Dropout(0.2))
```

Figure 3.16: LSTM layer

2. Visualize the results of predicted and real stock price

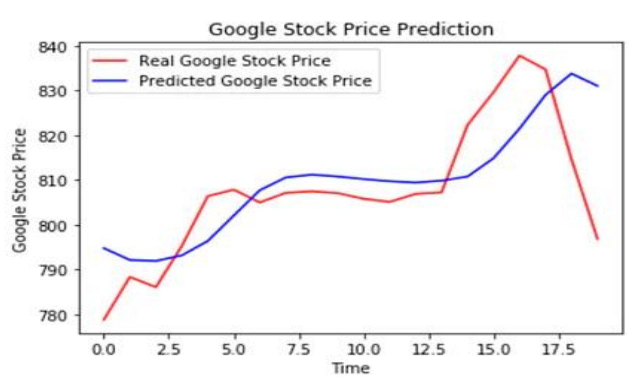


Figure 3.18: visualizing the prediction

2. Implementations and experimentations

2.1. Development environment and tools

2.1.1 Google Colab:

Colaboratory, or Colab for short, is a product of Google Research. It allows anyone to write and run any Python code from a browser, and is especially useful for machine learning, data analysis, and education. Technically, Colab is a hosted Jupyter notebook service that requires no setup and provides free access to computing resources, including GPUs. Colab is used extensively in the machine learning community with applications including:

- Getting started with TensorFlow
- Developing and training neural networks
- Experimenting with TPUs
- Disseminating AI research

2.1.2 Python:

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

2.1.2.1 Python libraries:

The libraries used in our work are:

- TensorFlow: Released on November 15 2015 by Google, TensorFlow is an open-source library written in Python for numerical computation. It has a great success in the Machine Learning community and in less than one year, it also had a lot of support and development by Google itself, moreover, by many community projects, developed in any area of Deep Learning. The peculiarity of TensorFlow is its work flux, made by data flow graphs. Where nodes represent mathematical operations, edges represent the multidimensional data arrays communicated between them; the latter can be considered, as in electronics, a Tensor, from here its name. In May 2016, Google revealed that it has used TensorFlow in the AlphaGo project, with special hardware dedicated to boosting the library's performance. To reach this goal rapidly, Google dedicated special attention to the user experience of TensorFlow, which arrives with great basic support and a well-grown GitHub community, the key to this quick improvement.
- Pandas: is a library for working with and manipulating tabular style data. In many ways, you can think of it as a replacement for a spreadsheet, only it is much more powerful.
- Keras: is an open-source library written in Python, running on top of the machine learning and deep neural networks platform TensorFlow. It was developed with a focus on enabling fast experimentation. Being able to go from idea to result as fast as possible is key to doing good research.
- Numpy: is a python library, intended to handle multidimensional matrices or arrays and mathematical functions.
- Matplotlib: is a python library, it is used for plotting and visualizing data as a graph.

2.2 Dataset used

COVID-19 Data is from repository of the Center for Systems Science and Engineering (CSSE) at Johns Hopkins University. This is a repository for the data for 2019 Novel Coronavirus from Visual Dashboard operated by the Johns Hopkins University Center for Systems Science and Engineering (JHU CSSE). Along with the ESRI Living Atlas Team and the Johns Hopkins University Applied Physics Lab (JHU APL).The Johns Hopkins Coronavirus Resource Center (CRC) for the continuously updated source of COVID-19 data and expert guidance. They collect and analyze the best data available on cases, deaths, tests, hospitalizations, and vaccines to help the public, policymakers, and healthcare professionals worldwide respond to the pandemic

Update frequency: Once a day around 23:59 (UTC). [60]

2.3 Experiments

- Import data from the CSV file and transfer it to a dataframe.

```
{url = "https://raw.githubusercontent.com/CSSEGISandData/COVID-
19/master/csse_covid_19_data/csse_covid_19_time_series/time_series_covid19
_confirmed_global.csv"
df_confirmed = pd.read_csv(url)}
```

- Change the dataset shape into 2 columns of confirmed cases and the date, make the date as an index, and create time series

```
{time_series_data = pd.DataFrame()
for i in range(len(cases)-1): df =  pd.DataFrame
(df_list[i][df_list[i].columns[4:]].sum(),columns=[cases[i]])time_series_d
```

```
ata = pd.concat([time_series_data,df],axis = 1)time_series_data.index = pd
.to_datetime (time_series_data.index,format='%m/%d/%y')}
```

- Splitting the dataset into 20% for testing and 80% for training

```
{split = round(0.8*len(dataset))
dataset = dataset.reshape(-1,1)}
```

- Scale the training data which is the first 80% using the MinMaxScaler library

```
{scaler = MinMaxScaler(feature_range=(0,1)
scaler.fit(dataset[:split])}
 dataset_n = scaler.transform(dataset).flatten()
dataset_n.shape
```

- Creating the datasets testing and training

```
{def create_dataset(df,previous,split_ratio):
    X, Y = [], []
    for i in range(len(df)-previous):
        a = df[i:(i+previous)]
        X.append(a)
            .
            .
return X,X_train,X_test,Y,Y_train,Y_test
T = 5 #number of past days used to predict the value for the current day
X,X_train,X_test,Y,Y_train,Y_test = create_dataset(dataset_n,T,0.8)}
```

	Shape	**train**	**test**
data	850	680	170

Table 3.1 : Data splitting

- Define LSTM Model

```
{model = Sequential()
model.add(LSTM(150, activation='relu', return_sequences=True, input_shape=
(seq_size, n_features)))
model.add(LSTM(64, activation='relu'))
model.add(Dense(64))
model.add(Dense(1))
model.compile(optimizer='adam', loss='mean_squared_error')}
```

- The summary of the model

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm (LSTM)                 (None, 7, 150)            91200


 lstm_1 (LSTM)               (None, 64)                55040


 dense (Dense)               (None, 64)                4160


 dense_1 (Dense)             (None, 1)                 65


=================================================================
Total params: 150,465
Trainable params: 150,465
Non-trainable params: 0
_________________________________________________________________
```

- The predicting model

```
Y_pred = model.predict(X_test)
Y_pred = scaler.inverse_transform(Y_pred)
Y_test = scaler.inverse_transform(Y_test.reshape(-1,1))
Y_train = scaler.inverse_transform(Y_train.reshape(-1,1))
```

- After doing the prediction using model. predict we invert these predictions back to prescaled values using scaler.inverse_transform to it compare with actual numbers of Covid cases test data .

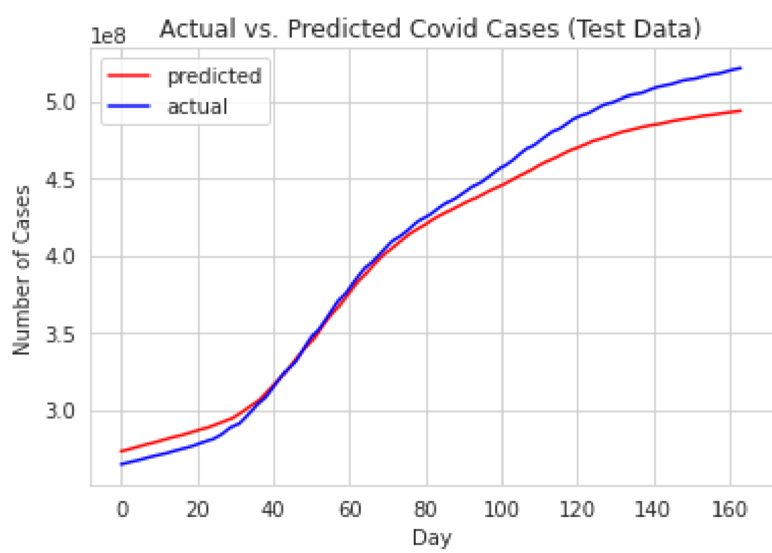


Figure 3.17: Actual vs Prediction covid cases

- to know the efficacy of the model we calculate its accuracy where:

$$Accuracy = \frac{Min(actual\ number\ of\ cases, predicted\ number\ of\ cases)}{Max(actual\ number\ of\ cases, predicted\ number\ of\ cases)}$$

Finding that the model has 92% accuracy

2.4 Comparison

The following table shows a comparison between the results of COVID prediction got from our model and other models that used the same datasets as us: [61]

	Method used	Run Time	Dataset	Accuracy
1	Polynomial Regression	10 second	John Hopkins Dataset	57%
2	Support Vector Machine(SVM)	10 second	John Hopkins Dataset	86%
3	Bayesian Ridge	15 second	John Hopkins Dataset	74%
4	Long Short-Term Memory (LSTM) (Our model)	37 second	John Hopkins Dataset	92%

Table 3.1: Comparison between some prediction models and our model

* As is shown in the table, we notice that our model, which is a deep learning model, is much more effective than the other models' machine learning methods, no matter how advanced and tailored they are.

2.5 Result and Discussion:

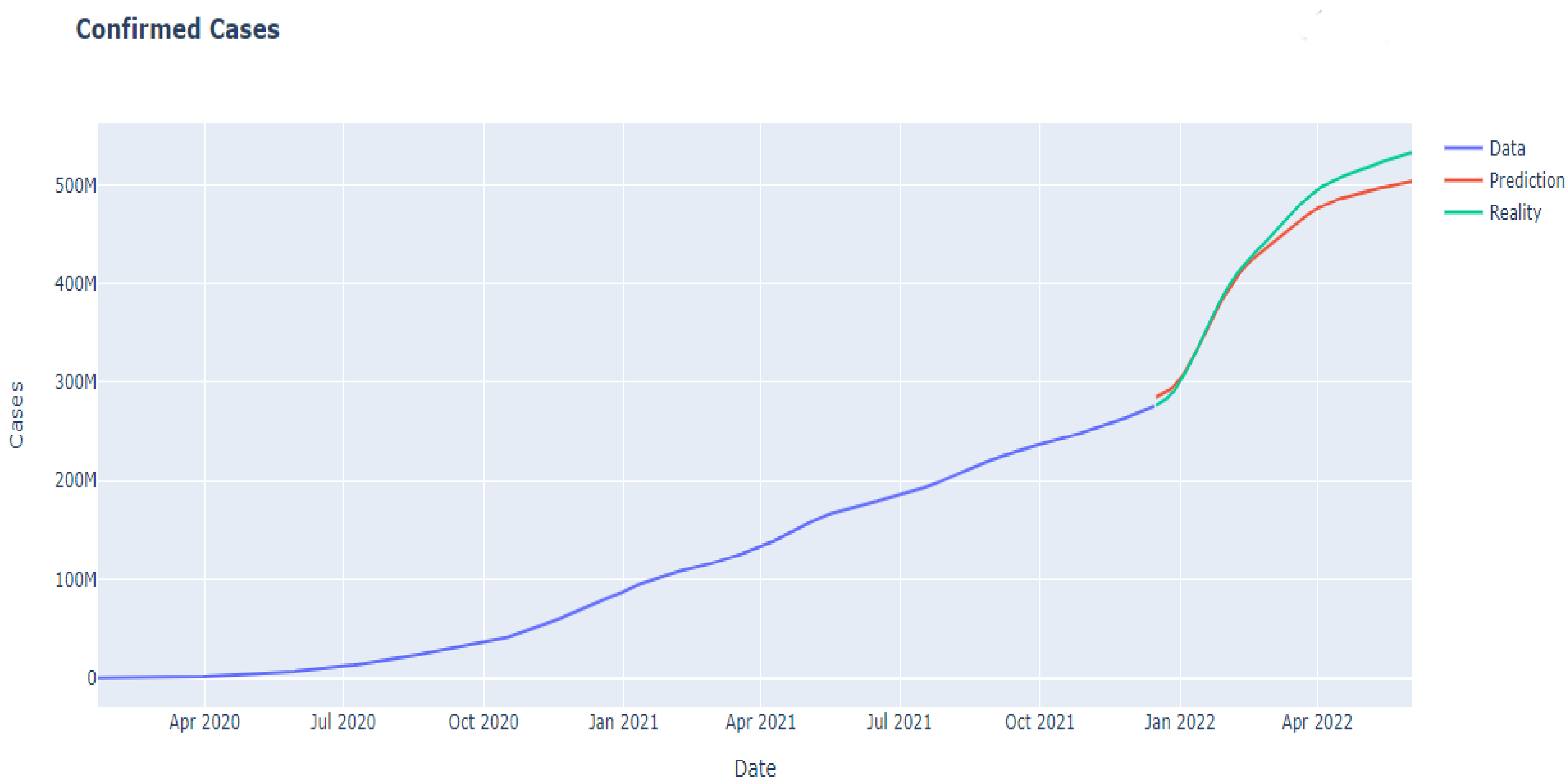


Figure 3.18: Forecasting try

Attempt to predict the number of cases in the past to compare it to the actual number of cases, finding that the model predicted 493.7086M on 09-05-2022 while the real number was 521.1289M finding that we achieved an accuracy of 95%

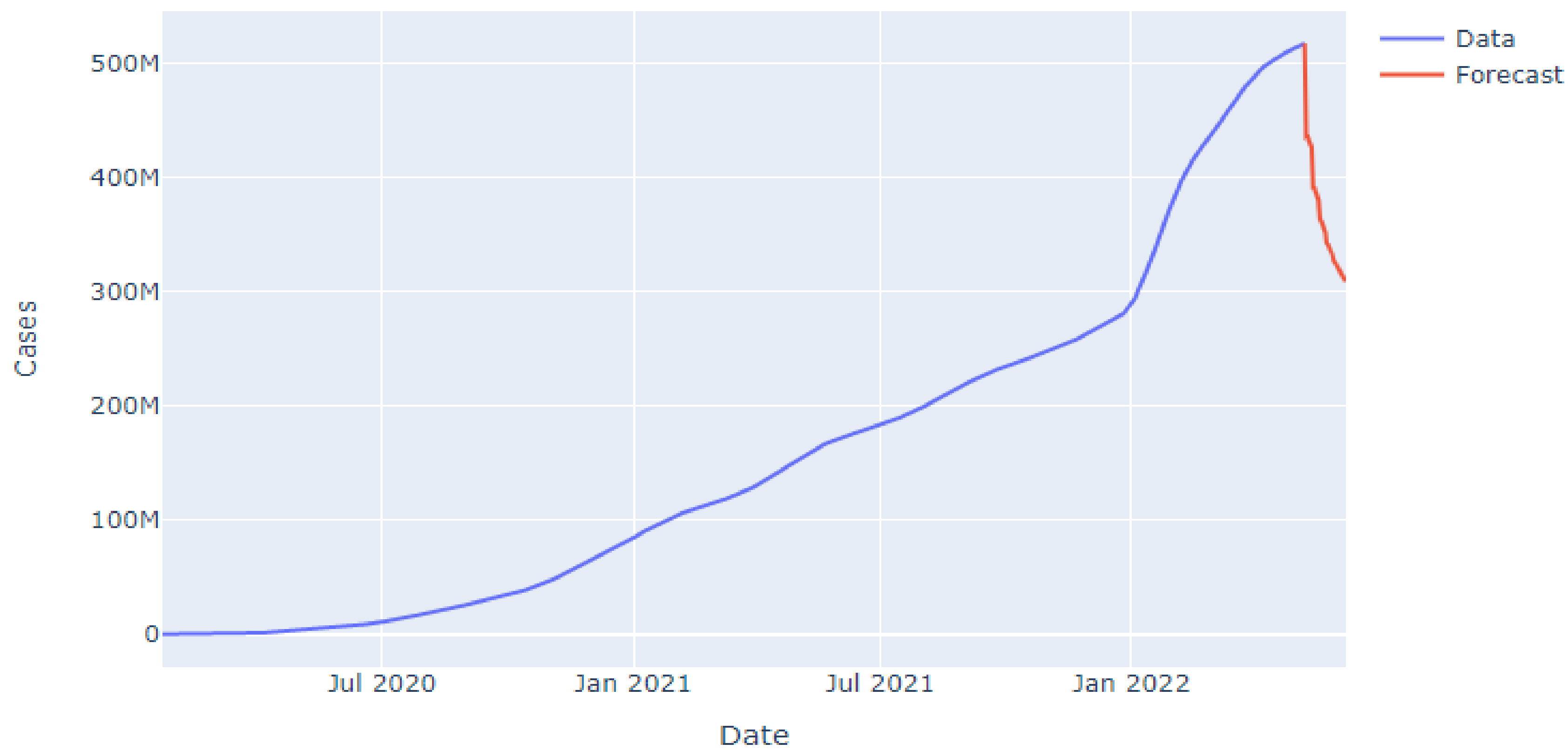


Figure 3.19: Forecasting the future cases

We forecast Covid-19 cases for the next 30 days from the day **09-05-2022** to **08-06-2022** the result is more than 11 million new cases worldwide

Conclusion

In this chapter, we have two sections:

In the first section, we had an overview of the field of deep learning and some of its neural networks, as well as the neural networks we will use in our predictive models. Long Short Term Memory (LSTM) is a type of Recurrent Neural Network (RNN).

In the second part, we consider the implementation and experimental phase, where we demonstrate our model using Long Short Term Memory (LSTM) and the Johns Hopkins University dataset, where we obtain predictions with 92% accuracy, as we predict the same for new cases in the next 30 days.

General Conclusion

The pandemic of Covid-19 has upset the world order and resulted in major crises on all fronts, especially on the socio-economic axis. Like all those affected by this pandemic, we consider contributing by providing a reliable way to predict the number of new cases in the next few days. It can be assumed that the results presented will help deal with the virus in future cases. In this case, we introduced a deep learning-based model to predict future Covid-19 cases. We chose a well-known forecasting method in the field of time series forecasting with LSTM models. Using the LSTM model and the daily updated Johns Hopkins dataset, we obtained predictions with 92% accuracy. We compared this result with three other models using the same dataset as ours. We found our model to be more accurate than either of them. After receiving the latest results of our Covid-19 case forecast for the next 30 days, we have detected more than 11 million new cases globally.

This work is only a beginning and it remains open to several extensions, as perspectives:

- Continue the prediction process as confirmed case results progress.
- Work with other prediction models for time series in order to have a better prediction with a reduced error rate.
- Make predictions based on other factors such as weather, region, or vaccination rate.

Bibliography

[1] Bol Med Hosp Infant Mex. 2021;78(1):41-58. doi: 10.24875/BMHIM.20000249.

[2] 14. Wang C, Horby PW, Hayden FG, Gao GF. A novel coronavirus outbreak of global health concern. Lancet. 2020;395:470–473.

[3] 15. World Health Organization . WHO; Geneva: 2020. Novel coronavirus (2019-nCoV) situation report 22.https://apps.who.int/iris/handle/10665/330991 Available at: [accessed 13 April 2020] .

[4] https://www.lemonde.fr/planete/article/2020/01/09/une-pneumonie-d-origine-inconnue-enchine_6025276_3244.html

[5] World Health Organization . WHO; Geneva: 2020. Coronavirus disease 2019 (COVID-19) situation

[6] Peiris JSM. Coronaviruses. In: Greenwood D, Barer M, Slack R, Irving W, editors. Medical microbiology. 18th ed. Churchill Livingstone; Edinburgh: 2012. pp. 587–593.

[7] World Health Organization. SARS statistics. Summary of probable SARS cases with onset of illness from 1 November 2002 to 31 July 2003. Geneva: WHO. Revised on 26 Sep 2003. Available at:https://www.who.int/csr/sars/country/table2004_04_21/en/ [accessed 13 April 2020].

[8] World Health Organization. Statistics on Middle East respiratory syndrome. Geneva: WHO. Published on Nov 2019. Available at:https://www.who.int/emergencies/mers-cov/en/ [accessed 13 April 2020].

[9] https://www.futura-sciences.com/sante/definitions/coronavirus-covid-19-18585.

[10] https://en.wikipedia.org/wiki/COVID-19_naming.

[11] Jie Cui, Fang Li et Zheng-Li Shi, « Origin and evolution of pathogenic coronaviruses », Nature Reviews Microbiology, vol. 17, no 3, mars 2019, p. 181–192 (ISSN 1740-1534, DOI 10.1038/s41579-018-0118-9.

[12] Y. Chen, « The Novel Severe Acute Respiratory Syndrome Coronavirus 2 (SARS-CoV-2) Directly Decimates Human Spleens and Lymph Nodes », University of Oxford - Immunology Network,202

[13] https://www.cdc.gov/coronavirus/2019-ncov/science/science-briefs/sars-cov-2-transmission.html .

[14] https://www.who.int/fr/health-topics/coronavirus/coronavirus#tab=tab_1.

[15] https://www.who.int/fr/emergencies/diseases/novel-coronavirus-2019/question-and-answ ers-hub/q-a-detail/coronavirus-disease-covid-19#:~:text=symptoms

[16] Coronavirus et Covid-19 · Inserm, La science pour la santé

[17] Algérie : Rapport de situation sur l'épidémie du COVID-19 #451 (du 22 juin 2021) - Algeria | ReliefWeb

[18] https://corona-dz.live/

[19] Yann Lecun. Les enjeux de la recherche en intelligence artificielle. Interstices, February

2016. URL https://hal.inria.fr/hal-01350469.

[20] Ali, M. A., Svetinovic, D., Aung, Z., & Lukman, S. (2017). Malware Detection In Android Mobile PlatformUsing Machine Learning Algorithms. 2017 International Conference On Infocom Technologies And UnmannedSystems (Trends And Future Directions) (Ictus)Khalifa University Of Science And Technology Abu Dhabi, UnitedArab Emirates.

[21] Tom M. Mitchell. Machine Learning. Mcgraw-Hill Series In Computer Science. Mcgraw-Hill, 1997

[22] Stephanie Thomas, Michael Paluszek, Matlab Machine Learning

[23] https://aldro61.github.io/microbiome-summer-school-2017/sections/basics/

visited(25/4/2018)

[24] https://medium.com/deep-math-machine-learning-ai/different-types-of-machinelearning-and-their-types-34760b9128a2 visited(25/42018)

[25] https://www.javatpoint.com/unsupervised-machine-learning

[26] Recommended Citation Zhao, K. (2018) ,''Handwritten Digit Recognition And Classification Using Maching Learning''. M.Sc. In Computing (Data Analytics), Technological University Dublin.

[27] Vansh Jatana,November 2018,SRM Institute of Science and Technology,Dive Into

[28] https://builtin.com/data-science/tour-top-10-algorithms-machine-learning-newbies

[29] https://ghrp.biomedcentral.com/articles/10.1186/s41256-020-00175-y

[30]http://di.univblida.dz:8080/jspui/bitstream/123456789/10306/1/COVID19Regression%20%281%29.pdf

[31] https://ieeexplore.ieee.org/document/9099302

[32] https://link.springer.com/article/10.1007/s13198-021-01272-y

[33] https://ieeexplore.ieee.org/document/9377137

[34] Oleksii Tsymbal,5 Essential Machine Learning Techniques For Business Applications,

https://mobidev.biz/blog/5-essential-machine-learning-techniques

[35] McCulloch, Warren; Walter Pitts (1943). "A Logical Calculus of Ideas Immanent in Nervous Activity". Bulletin of Mathematical Biophysics. 5 (4): 115–133. doi:10.1007/BF02478259.

[36] Deng, L., & Yu, D. (2014). Deep learning: methods and applications. Foundations and

[37] https://www.simplilearn.com/tutorials/deep-learning-tutorial/deep-learning-applications

[38] Abhishek Panigrahi, Yueru Chen, and C-C Jay Kuo. Analysis on gradient propagation in batch normalized residual networks. arXiv preprint arXiv:1812.00342,2018.

[39] Jonathan Lorraine and David Duvenaud. Stochastic hyperparameter optimization through hypernetworks. arXiv preprint arXiv:1802.09419, 2018

[40] Nathan Hubens. Deep inside: Autoencoders - towards data science, Apr 2018.

[41] Alessandro Achille and Stefano Soatto. Information dropout: Learning optimal representations through noisy computation. IEEE transactions on pattern analysis and machine intelligence, 40(12):2897–2905, 2018. doi: 10.1109/TPAMI.2017.2784440.

[42] Toshihiro Takahashi. Statistical max pooling with deep learning, July 3 2018. US Patent 10,013,644.

[43] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167, 2015

[44] Chaochun Liu, Yaliang Li, Hongliang Fei, and Ping Li. Deep skip-gram networks for text classification. In Proceedings of the 2019 SIAM International Conference on Data Mining pages 145–153. SIAM, 2019.

[45] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. IEEE Transactions on knowledge and data engineering, 22(10):1345–1359, 2009. doi: 10.1109/TKDE.2009.191.

[46] Samira Pouyanfar, Saad Sadiq, Yilin Yan, Haiman Tian, Yudong Tao, Maria Presa Reyes, Mei-Ling Shyu, Shu-Ching Chen, and SS Iyengar. A survey on deep learning: Algorithms, techniques, and applications. ACM Computing Surveys (CSUR), 51(5):1–36, 2018.

[47] Charu C. Aggarwal. Neural Networks and Deep Learning - A Textbook. Springer, 2018a. ISBN 978-3-319-94462-3. doi: 10.1007/978-3-319-94463-0. URL https://doi.org/10.1007/978-3-319-94463-0.

[48] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. nature, 521(7553): 436–444, 2015.

[49] Akshay Kulkarni and Adarsha Shivananda. Natural language processing recipes. Springer, 2019.

[50] https://www.sciencedirect.com/topics/engineering/deep-belief-network

[51] Ke-Lin Du and Madisetti NS Swamy. Neural networks and statistical learning. Springer Science & Business Media, 2013.

[52] John A Bullinaria. Recurrent neural networks. Neural Computation: Lecture, 12, 2013.

[53] https://www.simplilearn.com/tutorials/deep-learning-tutorial/rnn

[54] Weijiang Feng, Naiyang Guan, Yuan Li, Xiang Zhang, and Zhigang Luo. Audiovisual speech recognition with multimodal recurrent neural networks. In 2017 International Joint Conference on Neural Networks (IJCNN), pages 681–688. IEEE, 2017

[55] Vishnu Subramanian. Deep Learning with PyTorch: A practical approach to building neural network models using PyTorch. Packt Publishing Ltd, 2018.

[56] Roger Grosse. Lecture 15: Exploding and vanishing gradients. University of Toronto Computer Science, 2017.

[57] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. Neural Comput.,9(8):1735–1780, November 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735.URL https://doi.org/10.1162/neco.1997.9.8.1735.

[58] Nikhil Buduma and Nicholas Locascio. Fundamentals of deep learning: Designing next-generation machine intelligence algorithms.

[59] The Verge. Google revelead AlphaGo secret Hardware with ASIC chip-tensor Processor Unit for Machine-Learning. May 27, 2018. http://www.theverge.com/circuitbreaker /2016/5/19/11716818/google-alphago-hardwareasicchip-tensorprocessor-unit-machine-learning.

[60] https://github.com/CSSEGISandData/COVID-19

[61] https://www.kaggle.com/code/sohelranaccselab/coronavirus-visualization-prediction-using-ml