RISK CALCULATION OF COVID-19 FOR ASEAN COUNTRIES USING BACKPROPAGATION NEURAL NETWORK AND FUZZY INFERENCE SYSTEM

SABAI OO

M.C.Sc. **JUNE 2022**

RISK CALCULATION OF COVID-19 FOR ASEAN COUNTRIES USING BACKPROPAGATION NEURAL NETWORK AND FUZZY INFERENCE SYSTEM

BY

SABAI OO

B.C.Sc.(Hons:)

A Dissertation Submitted in Partial Fulfillment of the Requirements for the Degree of

Master of Computer Science

(M.C.Sc.)

University of Computer Studies, Yangon

JUNE 2022

ACKNOWLEDGEMENTS

To complete this thesis, many things are needed like my hard work and the supporting of many people who gave a lot of idea to me to complete this thesis.

Firstly, I am to express my appreciation and thanks to **Dr. Mie Mie Khin**, Rector of the University of Computer Studies, Yangon, for her kind permission to submit this dissertation.

I am thankful to my thesis supervisor, **Dr. Htar Htar Lwin**, Pro-Rector and Head of Faculty of Computer Systems and Technologies of the University of Computer Studies, Yangon, for her close supervision, proper guidance, valuable suggestions, guidance, advice and encouragement during the course of this work.

I would like to thank course coordinators, **Dr. Si Si Mar Win** and **Dr. Tin Zar Taw**, Professors of Faculty of Computer Science of the University of Computer Studies, Yangon, for their support, guidance, supervision, patience and encouragement during the period of study towards completion of this thesis.

I would like to express my gratitude to **Dr. Theingi**, Rector of the Technological University, Thanlyin and **Dr. Mon Mon Zaw,** Professor and Head of Department of Information Technology Engineering of the Technological University, Thanlyin, for helpful suggestions and administrative support during my academic study.

I also would like to express my gratitude and appreciation to **Daw Aye Aye Khine,** Associate Professor and Head of English Department of the University of Computer studies, Yangon, for her advice, editing and suggestion from the language point of view.

I also like to acknowledge my thanks to **all my teachers** who taught me throughout the master's degree course and **my friends** for their corporation and wish to express my gratitude to **my beloved parents** for their invaluable support and encouragement to fulfill my wish.

If not for the above mentioned people, my master's degree would never have been completed in such a successfully manner. I once again extend my sincere thanks to all of them.

STATEMENT OF ORIGINALITY

I hereby certify that the work embodied in this thesis is the result of original research and has been submitted for a higher degree to any other University or Institution.

………………………… ………………………

Date Sabai Oo

ABSTRACT

CORONAVIRUS DISEASE (COVID-19) is the infectious disease caused by the coronavirus that was first discovered in Wuhan City, China, and then spread throughout the world. Many researchers have proposed various methods to predict the spread of viruses. Predicting the number of COVID-19 patients is a crucial task in the effort to assist governments and healthcare departments respond rapidly to outbreaks. One type of prediction method is Artificial Neural Network (ANN), which is much more flexible and can handle more complicated and unassuming cases than the regression method. There are many ANN algorithms. Among them, the backpropagation algorithm is used in the proposed system. The backpropagation algorithm is a method for training multilayer feed-forward neural networks. It can be used to solve predictive problems with good results. Firstly, the proposed system implements a prediction model to estimate the number of COVID-19 sufferers in ASEAN Countries using a backpropagation neural network with Gradient Descent and a Backpropagation neural network with Stochastics Gradient Descent Optimizers. Among them, the method that produced the best performance is used to predict the future number of COVID-19 cases. And then these predicted results are used to decide the risk category of a country with Fuzzy Inference System. To evaluate the performance of the prediction methods for the number of COVID-19 sufferers, Root Mean Square Error (RMSE) is used and compared. According to the experimental results, the Backpropagation neural network with Stochastics Gradient Descent method has a better performance than the Backpropagation neural network with Gradient Descent method. The accuracy of the Fuzzy Inference method for the classification of the risk category of each country is calculated many times by using the preexisting actual trend data. As a result, the proposed system can be useful for risk categorization and long-term outbreak prediction in epidemics like COVID-19.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF EQUATIONS

CHAPTER 1
INTRODUCTION

COVID-19 is a big challenge and the whole world is now facing at the present time. In December 2019, Wuhan, China, reported the discovery of the first case. On December 31, 2019, the coronavirus disease 2019 was initially reported to the World Health Organization (WHO). On January 30, 2020, the COVID-19 outbreak was declared a global health emergency. Some common symptoms of COVID-19 include shortness of breath, coughing, fever, chills, fatigue, and loss of smell and taste, with less common ones including shivering, sore throat, headache, pink eye, a stuffy or runny nose, rash, gastrointestinal symptoms, muscle aches, and pains.

COVID-19 is a very serious disease because it mainly attacks the lungs in the human body and can cause death for the sufferers, especially such to congenital diseases or a weak immune system. Daily life, businesses, public health, food systems, educational systems, and employment have all been quickly affected by COVID-19.

Preventive measures to reduce the risk of infection include getting vaccinated, wearing face masks, ventilating indoors and avoiding crowded indoor spaces, washing hands with soap, maintaining social distancing, self-isolation, managing a healthy diet and lifestyle, and controlling international travel-related measures. With the emergence and spread of COVID-19, different modeling, estimating, and forecasting methodologies are being used to better understand and control the pandemic.

To control the spread of the virus and serve as a guide for health officials, it is crucial to predict the number of COVID-19 patients. Virus spread prediction can design better strategies to make more efficient decisions. Various prediction methods have been proposed to recognize and predict the COVID-19 outbreak. Artificial Neural Network (ANN) has been widely used by many researchers to analyze traditional classification and prediction problems.

ANN is one of the appropriate prediction methods and can learn from data and produce predictions or classifications as a result of their learning. There are many different types of ANN algorithms. The most used neural network algorithm is backpropagation, which is used for the system. Backpropagation is an important mathematical tool for improving the accuracy of predictions with good results. Its performance depends a lot on the optimization methods used during training. The most basic and commonly used optimization method is the Gradient Descent method.

In this system, the model is trained by using a backpropagation algorithm with GD and SGD optimizers. The backpropagation neural network with gradient descent method is compared to the backpropagation neural network with stochastic gradient descent method in this experiment. The method that produced the better performance is used in this proposed system to get the most accurate prediction model. Firstly, the backpropagation algorithm is used to predict the number of COVID-19 sufferers and then these results from the predicted model are used to decide the risk category of a country with Fuzzy Inference System.

1.1 Motivation of the Thesis

COVID-19 has long been considered a global pandemic. Different governments have responded to the COVID-19 pandemic in different ways, imposing varying restrictions on people's mobility within and between nations.

Because the COVID-19 epidemic has been a particularly negative impact on the global economy and has been disrupting people's daily lives, institutions, and markets, it is necessary to control the spread of the disease. These days, healthcare experts encounter a lot of difficulties in preserving the fine of healthcare.

Predictions of COVID-19 patient numbers are essential for controlling and preventing the spread of such diseases. Virus spread prediction can design better strategies to make more efficient decisions. Predictions, especially those involving epidemiological outbreaks, may or may not have medical applications, but they can be helpful planning, decision-making, and preventative tools.

1.2 Objectives of the Thesis

Modeling research has been carried out with a variety of objectives, which includes the prediction of transmission patterns, diagnosis and prognosis of infection, and predicting the number of COVID-19 sufferer in the future using the previous data. The objectives of the thesis are as follows:

(i) To predict the number of cases, the number of deaths, and the number of recoveries in the future using the previous data.

(ii) To predict the risk category of a country depends on the results of these cases.

(iii) To learn backpropagation neural networks by using GD and SGD optimizers.

(iv) To find a more powerful ANN Model by using the BP algorithm with GD and SGD optimizers for predicting COVID-19.

(v) To study the rates of cases and mortality rates due to the COVID-19 pandemic.

1.3 Contributions of the Thesis

The proposed system develops a predictive model-based neural network with backpropagation algorithm to predict the risk category of ASEAN countries. The proposed system is very useful for the Government and other healthcare organizations to enable a timely response to outbreaks and to ensure the availability of the required resources. The contributions of the thesis are as follows:

(i) An artificial neural network (ANN) model is developed to estimate the number of cases of COVID-19 in the ASEAN countries.

(ii) The system uses local trend data with a backpropagation neural network and a fuzzy inference system to predict the risk category of a country.

(iii) In the proposed system, the gradient descent method and stochastic gradient descent method are compared. Among them, the method that produced the best performance is used to predict the future number of COVID-19 cases.

(iv) A new way of predicting epidemic outbreaks and relating them to a country's risk is presented.

1.4 Organization of the Thesis

This thesis consists of five chapters, abstract, acknowledgment, and references. The introduction of the system is presented for the calculation of the risk of COVID-19 for ASEAN Countries. The motivation, contribution, and objectives of the research work are also described in this chapter.

The structure of the Artificial Neural Network and how it works step-by-step are presented in chapter two. Various learning techniques, Backpropagation neural network algorithm, learning datasets, and the step of Fuzzy Logic are briefly explained in this chapter.

The predicted model based on Backpropagation neural network is explained in detail in chapter three. First of all, an overview block diagram of the proposed system is described, and then a detailed explanation of data preprocessing using min-max normalization, Backpropagation neural network-based country-level risk calculation, and the step of fuzzy inference system to calculate the risk categories of each country is finally presented.

The design and implementation of the proposed system for calculating the risk of COVID-19 for ASEAN countries are presented in chapter four. First of all, the system flow diagram is described in this chapter. And then, the implementation of the predicted model for predicting the future number of COVID-19 and how the fuzzy inference system works to calculate the risk classes of each country are explained step by step. Finally, the experimental results are shown by Graphical User Interface, charts, graphs, and tables.

Chapter five presents the conclusion of the research work. Further extensions that propose some potential improvements are described in this chapter. This chapter also describes the limitations of the system.

CHAPTER 2
BACKGROUND THEORY

This chapter presents the features of Artificial Neural Networks, Learning Techniques, an Overview of Backpropagation Neural Networks, and Fuzzy Logic. Firstly, this chapter describes the difference between artificial neural networks and biological neural networks and the features and types of Artificial Neural Networks. Secondly, this chapter describes three types of Learning Techniques and an overview of the Backpropagation Neural Network that is important to implementing a prediction model. Finally, it presents Fuzzy Logic for Decision Making and Applications of Fuzzy Logic in Decision Making.

2.1 Biological Neurons and their Functions

The word neural comes from the (animal) human nervous system and refers to the nerve cells or neurons present in the brain [5]. The human brain nerve cells are divided into four sections:

- Dendrites: Other neurons send signals that are received.
- Cell body: It adds up all of the incoming signals to produce the input.
- Synapses: It is the interaction of one neuron with another. The signal amount is determined by the synaptic weight of connections.
- Axon: When the sum of neuron signals reaches a certain threshold, the neuron fires, sending the signal down the axon to other neurons. The strength of inhibitory connections naturally decreases or increases.

In general, a Neural Network (NN) is an interconnected network containing billions of neurons with numerous connections. The comparison of brains and traditional computers are shown in Table 2.1.

Table 2.1 Comparison of brains and traditional computers

Brain	Computers
Biological neurons or nerve cells	Silicon transistors
200 billion neurons, 32 trillion interconnections.	1 billion bytes RAM, trillion of bytes on disk.
Energy consumptions: 6-10 joules per operation per sec.	Energy consumption: 10-16 joules per operation per second.

Brain	Computers
Neuron size: 10-6 m.	Single transistor size: 10-9 m.
Learning capability	Programming capability

2.2 Artificial Neuron

A mathematical model of a biological neuron that receives one or more inputs and integrates them using an activation function to produce an output is known as an artificial neuron. It is an artificial replication of human brain neurons. It returns the value of the activation function after calculating the weighted sum of input signals (x). Figure 2.1 shows the working principle of an Aritificail Neural Network.

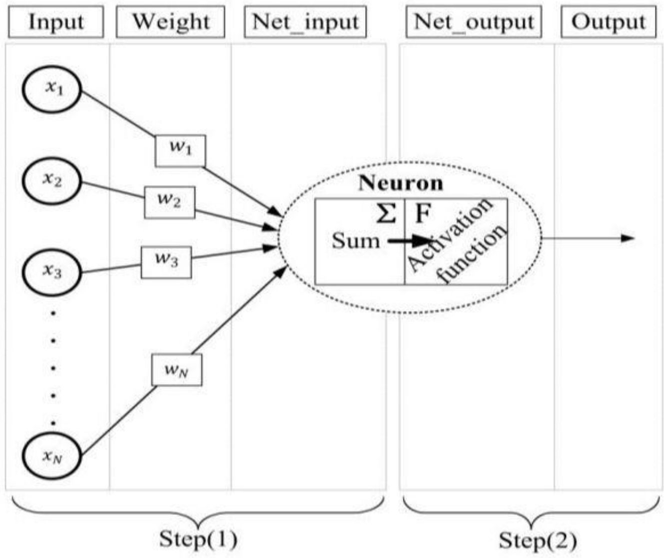

Figure 2.1 Working Principle of an Artificial Neuron

The differences between biological NN and ANN are listed in Table 2.2, although ANN is derived from the concept of biological NN.

Table 2.2 Difference between artificial neural network and biological neural network

Characteristics	Artificial Neural Network	Biological Neural Network
Storage	The old data may be deleted by newly added information.	A network of interconnected neurons that is extremely complex and dense and contains neurons with numbers between 1011 and 1015

Characteristics	Artificial Neural Network	Biological Neural Network
Size and Complexity	It is smaller and less complicated, and it cannot perform pattern recognition tasks.	It has a highly complex and dense network of interconnected neurons, each of which has between 1011 and 1015 connections.
Speed	Very fast in information processing using less than seconds	Little more slowly, milliseconds, while processing information
Control Mechanism	It has a control unit controlling all the activities	External computer disks are used instead of the main control unit.
Processing	Processing is serial	Processing is parallel
Fault tolerance	When the system has failed, it is impossible to recover corrupt information.	The system may adjust the stored information to new information while maintaining the previous information.

2.3 Activation Functions

An activation function in a neural network describes how a node or nodes in a layer convert the weighted sum of the input into an output. The activation function is sometimes known as a "transfer function." A "squashing function" is used to describe the activation function if its output range is restricted. Nonlinearity in the layer or network architecture refers to the fact that many activation functions are nonlinear.

Different activation functions may be used in various areas of the model, and they can have a major impact on the capabilities and performance of the neural network. The activation function is applied inside or after each network node's internal processing, even though networks are intended to employ the same activation function for all nodes in a layer.

A network has three different types of layers: output layers, which provide predictions, and hidden layers, which receive input from one layer and pass output to another. Input levels take raw domain information. The activation function is usually the same for all hidden layers. The output layer often employs a different activation function from the hidden layers, depending on the type of prediction required by the model.

Activation functions are usually differentiable, which means that for a given input value, the first-order derivative may be calculated. This is required because backpropagation of error algorithm, the process typically used to train neural networks, requires the derivative of prediction error to update the weights of the model.

In neural networks, there are many different types of activation functions, but only a few are employed in practice for hidden and output layers. The types of activation function in artificial neural network are shown in Tabel 2.3.

Table 2.3 The types of activation function in artificial neural network

Function Name	Formula	Range
Linear	$f(x) = x$	$-\infty, \infty$
Semi linear	$f(x) = \begin{cases} x, & x < 0 \\ 0, & x \geq 0 \end{cases}$	$0, \infty$
Logistic (Sigmoid)	$f(x) = \frac{1}{1+e^{-x}}$	$0, \infty$
Hyperbolic tangent	$f(x) = \frac{1}{1+e^{-2x}}-1$	$-1, 1$
Exponential	$f(x) = e^{-x}$	$0, \infty$
Sinusoidal (Sine)	$f(x) = \sin(x)$	$-1, 1$
Rational (Sigmoid)	$f(x) = \frac{x}{a+\lvert x\rvert}$	$-1, 1$
Step	$f(x) = \begin{cases} -1, & x \leq -1 \\ x, -1 < x < 1 \\ 1, & x \geq 1 \end{cases}$	$-1, 1$
Hard Limit (Threshold)	$f(x) = \begin{cases} 0, & x < 0 \\ 1, & x \geq 0 \end{cases}$	$0, 1$
Modular	$F(x) = \lvert x\rvert$	$0, \infty$
Signed (Signature)	$f(x) = \begin{cases} 1, & x < 0 \\ -1, & x \geq 0 \end{cases}$	$-1, 1$
Quadratic	$f(x) = e^2$	$0, \infty$
Relu	$f(x) = \max(0, x)$	$0, \infty$

Function Name	Formula	Range
Softmax	$f(x)_j = \frac{e^{x_j}}{\sum_{k=1}^{K} e^{x_k}}$	$0, 1$

2.4 Artificial Neural Networks

Deep learning techniques use neural network types called artificial neural networks (ANNs) and simulated neural networks (SNNs), which are a subset of machine learning. Their name and structure are inspired by the human brain. Artificial neural networks are algorithms that are based on models of human neurons (ANN). There are millions of neurons in the human brain. It uses electrical and chemical signals to send and process signals. Synapses are the specific structures that connect these neurons. Synapses allow neurons to pass another signal. Neural networks are formed from a large number of simulated neurons.

A method of processing information is an artificial neural network. Similar to how the human brain processes information, it functions. An ANN is a group of connected processing units that collaborate to process data. They also generate useful results from it. Not only may a neural network be used for categorization. It can also be used for regression of continuous target attributes.

These technologies are used in almost all business applications and commercial companies. Their main objective is to find solutions to complicated issues in a variety of fields, such as pattern identification, facial recognition, time series prediction, modeling, speech-to-text transcription, data analysis, check processing, healthcare, weather prediction, and signal processing.

2.4.1 The Architecture of Artificial Neural Networks

An artificial neural network is often set up in layers. Layers are composed of numerous interconnected "nodes" that each has an "activation function." The input layer, output layer, and hidden layer or layers are the three layers that make up a neural network. Each hidden layer node and the output layer nodes must be connected to the input layer nodes. The architecture of artificial neural network is shown in Figure 2.2.

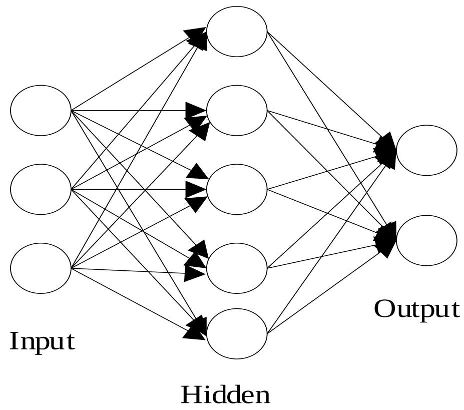

Figure 2.2 The Architecture of Artificial Neural Network

Input Layer: Artificial neurons in the input layers are those that take information from the outside world. The network's raw data can be fed into it through input unit activity. The "input layer," which communicates with one or more "hidden layers," is where the patterns are presented to the network. The nodes in the input layer are passive, meaning they have no impact on the data. They replicate a single value from their input across all of their outputs. Each value from the input layer is duplicated and sent to every concealed node.

Hidden Layer: Every computation on the input data is carried out by a network of neurons in the hidden layer. Any number of hidden layers are possible in neural networks. The most basic network consists of a single hidden layer. A set of weighted connections is used to carry out the actual processing in the hidden layer. The data entering a hidden node are multiplied by Weights, a collection of predetermined integers recorded in the software [25]. The final step is to add the weighted inputs to create a single number. Each hidden unit's activity is determined by the Hidden Layer. The links between the input and hidden units control the behaviors of input units and the weights.

Output Layer: An "output layer" is connected to the hidden layers after that. The output layer receives connections from the input layer or hidden layers. It provides an output value that matches the forecast of the response variable. In classification problems, there is typically only one output node. The active nodes of the output layer combine and change the data to create the output values. The selection of the weights must be accurate for the neural network to provide useful data manipulation. Conventional information processing is not the same as this. The behavior of the output units is influenced by the activity of the hidden units and the weights between the hidden and output units.

2.4.2 The Working of Artificial Neural Networks

Artificial neural networks can best be understood as weighted directed graphs with nodes created by artificial neurons and directed connections with weights denoting the relationship between neuron outputs and inputs. The input signals for the artificial neural network come from the outside world as patterns and images in the form of vectors. Then, these inputs are formally designated using the notations x(n) for every n number of inputs.

After that, each input is multiplied by the weights that have been allocated to it; these weights are the specifics that the artificial neural networks utilize to solve a certain problem [24]. The strength of the connections between the neurons in the artificial neural network is generally represented by these weights. All of the weighted inputs are combined within the computational unit.

If the weighted total is equal to zero, a bias must be added to make the output non-zero; otherwise, the system's response must be scaled up. Bias has a weight and an input that is always equal to 1. In this example, the range of the sum of the weighted inputs is 0 to positive infinity. To keep the response within the limits of the desired value, a specific threshold value is benchmarked. The sum of the weighted inputs is then passed on to the activation function.

The set of transfer functions used to produce the desired output is generally referred to as the activation function. Although there are many other types of activation functions, linear or non-linear sets of functions are the most popular. Among the most often used sets of activation functions are the Binary, Sigmoidal (linear), and Tan hyperbolic sigmoidal (non-linear) activation functions.

The input node converts the information into numerical form. Each node is assigned a number, which represents an activation value. The higher the number, the stronger the activation. The activation value is passed to the next node based on weights and the activation function. Each node computes and updates the weighted sum based on the transfer function (activation function). The activation function is then used to carry out the action. This neuron only has this function. The neuron must then select whether or not to transmit the signal. The ANN adjusts the weights, which determines the signal extension.

The activation travels throughout the network until it reaches the destination node. It makes sense that the output layer would share the data. The network compares the output and expected output using the cost function. Between actual and predicted values, there is a discrepancy known as the cost function. The result is more similar to the desired one when the cost function is smaller. Figure 2.3 shows the feedforward backpropagation neural network architecture.

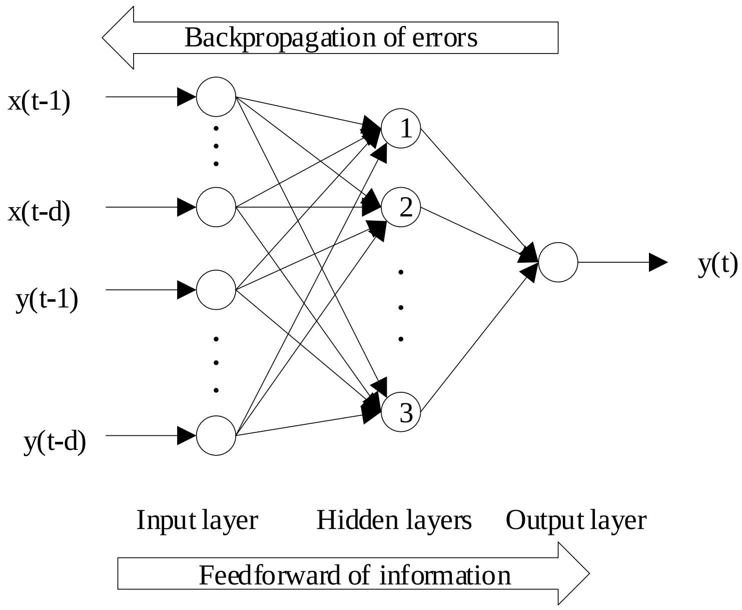

Figure 2.3 The Feedforward Backpropagation Neural Network Architecture

The cost function can be minimized using one of two methods:

1. **Back Propagation:** The heart of neural network training is backpropagation. It is the most important method of learning for neural networks. The data enters the input layer and proceeds to the output layer via the network. The output will then be equated with the expected output using the cost function. If the cost function's value is high, the information is returned, and the neural network learns to lower the cost function's value by changing the weights. The error rate is decreased and the model becomes certain when the weights are properly adjusted.
2. **Forward Propagation:** The data enters the input layer and proceeds to the output value via the network. The user compares the value to the expected outcomes. The next step is to calculate errors and transmit information backward. The user can now train the neural network and update the weights. The user can alter weights simultaneously with the structured algorithm. The

user will be able to determine which neural network weight is accountable for the error.

2.4.3 Types of Artificial Neural Network

There are two types of Artificail Neural Network : Feedforward Neural Network and Feedback Neural Network.

Feedforward Neural Network: Only one direction of information flow exists in feedforward ANNs. There are no feedback loops Any layer's output in such networks is independent of other layers. Straight-forward neural networks that link inputs with outputs are known as feed-forward neural networks. Their inputs and outputs are fixed. Pattern generation, pattern recognition, and classification are the most common applications. Figure 2.4 shows fully-connected Feedforward Neural Network.

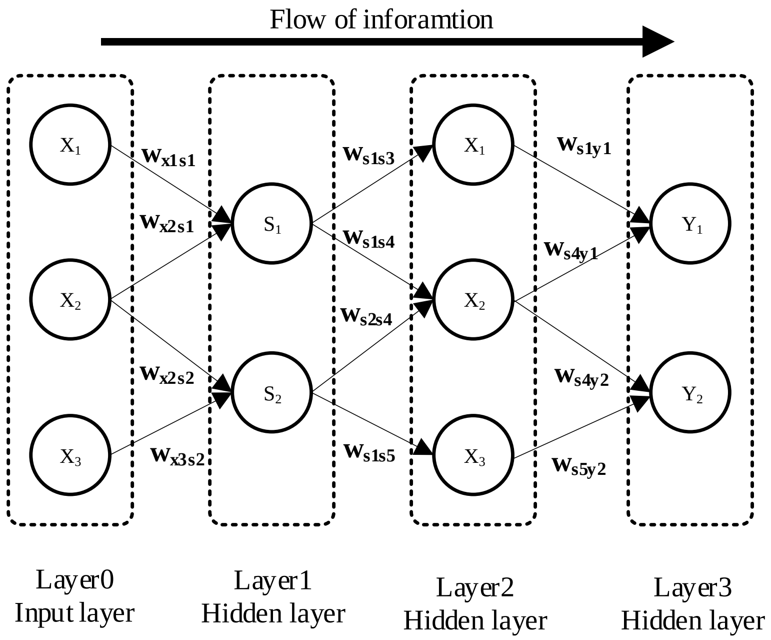

Figure 2.4 Fully-connected Feedforward Neural Network

Feedback Neural Network: Feedback loops are a component of the feedback ANNs. For example, recurrent neural networks are frequently employed in memory retrieval. These networks are best suited for data that is sequential or time-dependent. Recurrent neural networks (RNNs) are defined by feedback loops. They are used in a content addressable memory. The structure of Feedback Neural Network is shown in Figure 2.5 and the taxonomy of Artificial Neural Network architecture is shown in Figure 2.6.

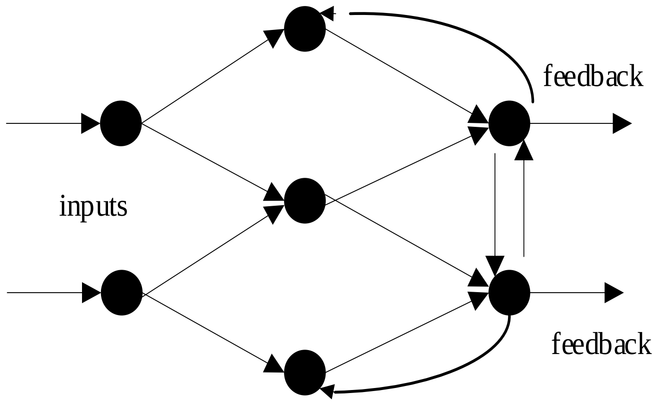

Figure 2.5 The Structure of a Feedback Neural Network

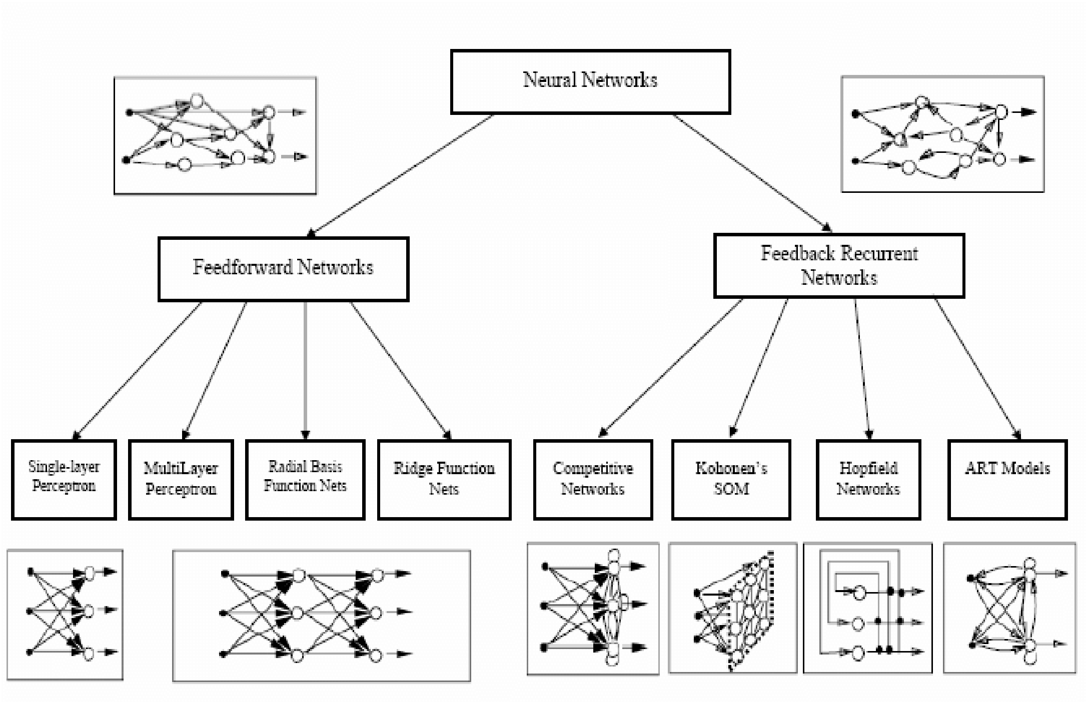

Figure 2.6 The Taxonomy of Artificial Neural Network Architecture

2.5 Learning Techniques

There are many different algorithms available for artificial neural network training, each with advantages and disadvantages. The learning processes of a neural network are divided into three groups. These include supervised, unsupervised, and reinforced learning [21]. The types of learning methods are shown in Figure 2.7.

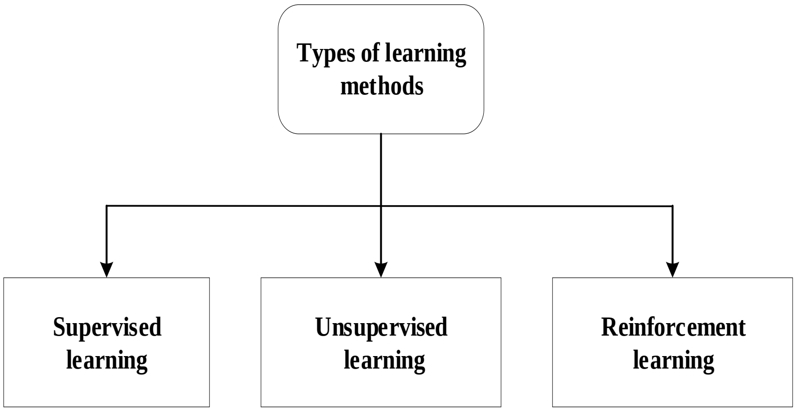

Figure 2.7 Types of Learning Methods

2.5.1 Supervised Learning

This kind of learning is carried out under the guidance of a teacher. Learning in this way is dependent. An artificial neural network's training results in an output vector once the input vector is presented to the network during supervised learning. This output vector is compared to the desired output vector. An error signal is produced if the actual output vector deviates from the anticipated output vector. Until the required output is achieved, the weights are adjusted based on the error signal. The process of supervised learning is described in Figure 2.8.

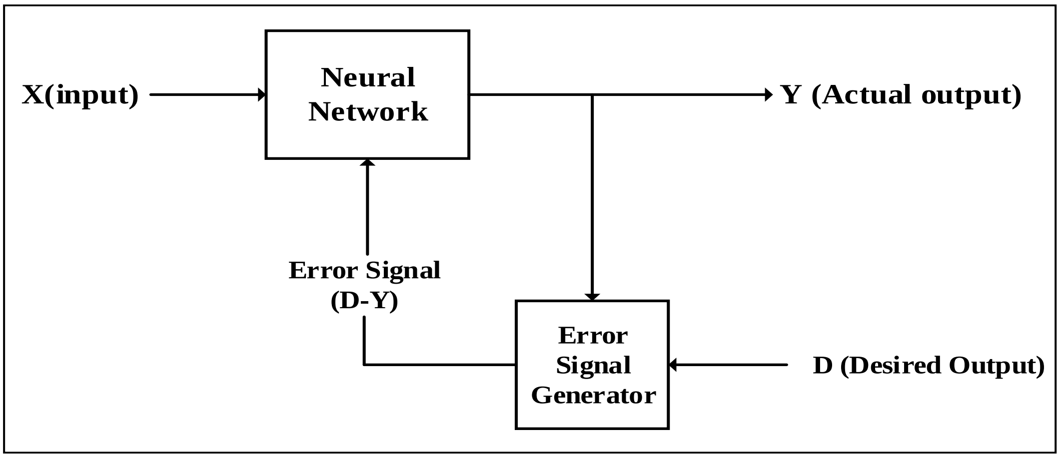

Figure 2.8 The Process of Supervised Learning

2.5.2 Unsupervised Learning

This kind of instruction is conducted without the guidance of a teacher. The learning process is self-contained. Unsupervised learning involves the formation of clusters during ANN training from input vectors of related types. The neural network responds with an output response that reflects the input pattern's class when a new input pattern is applied.

The environment does not provide any information about what the intended result should be or whether it is accurate. As a result, in this kind of learning, the network must ascertain the relationships between the input and output data as well as the patterns and features from the input data. Figure 2.9 shows the process of unsupervised learning.

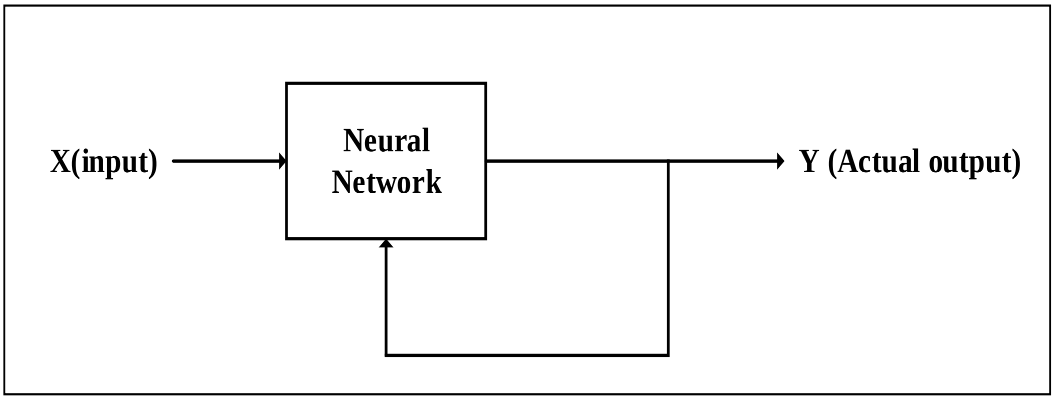

Figure 2.9 The Process of Unsupervised Learning

2.5.3 Reinforcement Learning

This kind of learning is used to reinforce or strengthen the network over certain important knowledge. Although there may be substantially less information in this learning process, it is similar to supervised learning.

During network training using reinforcement learning, the environment provides some feedback to the network. It resembles supervised learning as a result. As opposed to supervised learning, when an instructor would be present, the feedback is evaluative rather than informative. After receiving input, the network modifies the weights to produce future crucial information that is better. The process of reinforcement learning is shown in Figure 2.10.

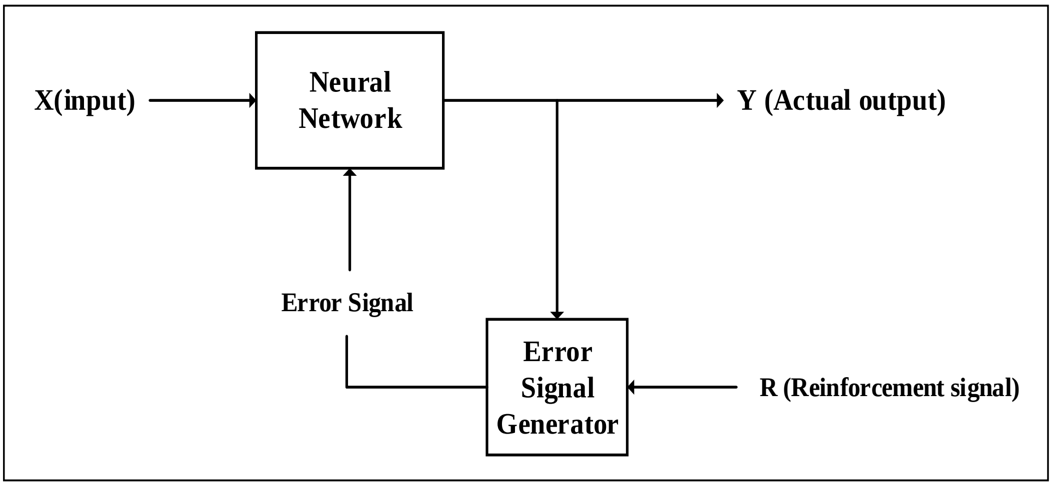

Figure 2.10 The Process of Reinforcement Learning

2.6 Training Algorithms and Neural Network to Data Application

The weights of the network are updated using a training technique that finds a decision function. The training algorithms come in many different forms. To appropriately map arbitrary inputs to outputs, the algorithms change the network weights and biases. Several of those are given in Table 2.4.

Table 2.4 The training algorithms and application of neural network

Paradigm	**Learning algorithm**	**Application**
Supervised learning (Classification, regression)	• Perceptron Learning Algorithm • Backpropagation • Adaline and Madaline • Boltzmann learning Algorithm • Linear discriminant analysis • Vector quantization • ART Map	• Classification of images • Approximation of functions • Prediction control • Classification of images • Data analysis, classification of images • Categorization within the class data compression

Paradigm	Learning algorithm	Application
Unsupervised learning (Clustering)	• Sammon mapping • Analysis of the principal components • Learning associative memory • Vector quantization • Kohonen Self Organizing Maps (SOM) • ART1, ART2	• Categorization within the class data compression • Data analysis, data compression • Associative memory • Categorization, data compression • Categorization, data analysis • Categorization

2.6.1 Back Propagation Neural Network Algorithm

The BP algorithm is at the heart of the most recent ANN learning research. This algorithm is entirely based on the learning technique of error correction. Because of the very general nature of the BP training method, it may be used to solve the problem in many areas.

It is a development of the gradient-based delta learning rule. Once an error is discovered, it is propagated backward from the output layer to the input layer via the hidden layer. In the case of a Multilayer Neural Network, it is used.

The following is a description of the BP learning process:

(1) Forward propagation of operating signal: From the input layer to the output layer, the input signal propagates through the concealed layer [12]. During the forward propagation of the operating signal, the weight value and offset value of the network are kept constant, and the condition of one layer of a neuron does not affect the status of the subsequent layer of the neuron. It is possible to switch to back propagation of the error signal if the desired output cannot be achieved in the output layer [6].

(2) Back propagation of error signal: The difference between the network's actual and expected outputs is known as the error signal, and in the back propagation of the error signal, the error signal propagates from the output end to the input layer. The error feedback that occurs during the back propagation of the error signal controls the

network's weight value. To get the actual output of the network closer to the expected one, the weight value and offset value are continuously changed [1].

2.6.2 Learning Data Sets

There are three learning data sets : training data set, validation set approach and making test set.

- Training Data Set

Several examples are used to fit the network's parameters or weights. On the training set, one approach includes one entire training cycle.

- Validation Set Approach

A set of examples for tuning the parameters [i.e., architecture] of the network. To choose the number of hidden units in a Neural Network, for example.

- Making Test Set

To assess the effectiveness [generalization] of a fully described network or to accurately forecast output from known input, just a set of instances is used.

2.7 Fuzzy Logic

The result produced by fuzzy logic systems (FLS) is satisfactory but unambiguous. Reasoning that is similar to human reasoning is called fuzzy logic (FL). The FL method uses human judgment to take into account any outcome between the digital values YES and NO. The traditional logic block in human language accepts exact input and produces a clear response as TRUE or FALSE, which is comparable to YES or NO.

Look at the diagram below. It demonstrates that values are represented in fuzzy systems by a number between 0 and 1. Absolute truth is given the value of 1.0, whereas absolute untruth is given the value of 0.0. A numerical representation of the value serves as the truth value in a fuzzy system. The basic concept of Boolean logic and fuzzy logic is shown in Figure 2.11.

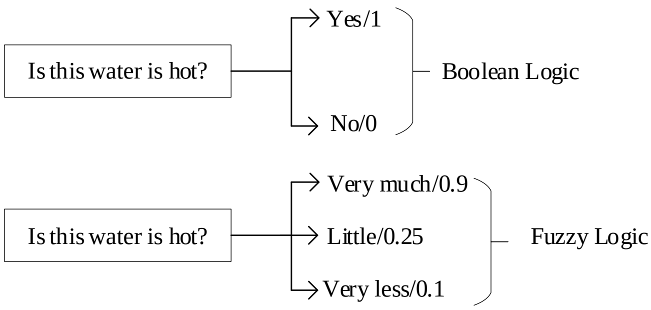

Figure 2.11 Boolean Logic and Fuzzy Logic

Instead of being a type of logic that is fuzzy, fuzzy logic is a type of logic that describes fuzziness. There are plenty of such instances like this that might help to understand the concept of fuzzy logic [2].

To produce a clear outcome, fuzzy logic operates on the levels of input possibilities. Depending on the size and capabilities of the system, it can be applied to everything from small microcontrollers to substantial networked workstation-based control systems.

2.7.1 Fuzzy Logic for Decision Making

Decision-making, which is widely defined to cover any decision or selection of options, is important in a variety of fields, including both "soft" social sciences and "hard" natural sciences and engineering disciplines. Fuzzy logic is based on a set of human language rules that the user supplies [4]. These rules are translated into mathematical equivalents by fuzzy systems. As a result, work for both the computer and the system designer is simplified, and approximations of how systems operate in reality are produced that are substantially more accurate.

Another benefit of fuzzy logic is that it is adaptable and simple to utilize. Fuzzy logic can represent nonlinear functions of any complexity and solve problems with imprecise or incomplete data. Traditional control techniques will not offer a better result than fuzzy control. To match any set of input-output data, a fuzzy system can be utilized. This is made easier by the inclusion of adaptive methods in the Fuzzy Logic Toolbox, such as fuzzy subtractive clustering and adaptive neuro-fuzzy inference systems (ANFIS).

Fuzzy inference systems and fuzzy logic models both consist of a set of conditional "if-then" rules [19].

These rules are easy to construct for a designer who is familiar with the system, and any number of rules needed to completely describe the system can be provided. The following are the steps in fuzzy decision-making:

1. The first stage is to identify variables and alternatives.
2. The technique of fuzzification transforms linguistic variables into actual variables.
3. The user determines which variables need to be kept in the knowledge base.
4. A mathematical expression of the membership function is the membership function.
5. The if-then condition rule will be given next. A single rule relates to each variable.
6. The next step is to convert the fuzzy value into an output variable.
7. The final phase of the fuzzy process is when the alternative is put into practice. The system will operate better and closer to the process's objective if the implementation is successful.

2.7.2 Application of Fuzzy Logic in Decision Making

To use fuzzy sets in decision-making, it was generally necessary to fuzzify the classical decision-making theories. To simulate risky decision-making, game theories and probabilistic decision theories have both been employed. Fuzzy decision theories address the ambiguity and non-specificity that define human choice, constraint, and goal formulation.

Following is a list of some of the more well-known applications:

- Because fuzzy logic simulates decision-making more quickly than humans, it is employed with neural networks. Data are combined in this process, and fuzzy sets are used to transform partial truths into more accurate facts.
- In the larger business setting, it is utilized for systems that support decision-making and for individual evaluation.
- It has been utilized to regulate traffic and speed in automotive systems.
- Several Artificial Intelligence applications, including Natural Language Processing, use fuzzy logic.
- It is employed in the aerospace sector to regulate satellite and spacecraft height.

- Modern control systems, including expert systems, frequently incorporate fuzzy logic.

2.8 Summary

The theoretical background for artificial neural networks and fuzzy logic is presented in detail in this chapter. Differences between artificial neural networks and Biological neural networks are explained.

Common features of ANN and their working are explained step by step. The role of backpropagation in the artificial neural network is described. In the proposed system, a backpropagation neural network was used to predict the future number of confirmed cases, recovered cases, and death cases of COVID-19 in ASEAN countries. To calculate the risk categories of each country, Fuzzy Logic and its applications are also presented in this chapter.

CHAPTER 3

RISK CALCULATION OF COVID-19 FOR ASEAN COUNTRIES

In this section, an overview of the proposed system and its method are described in detail. The implementation of a predictive model for the prediction of COVID-19 cases is explained using Backpropagation Neural Network. The loss function and cost function are the important parameters for optimizing the learning algorithms and the optimizers used for reducing the loss are also explained. Moreover, to calculate the risk of each country, the steps of the Fuzzy Inference System are presented in detail in this chapter.

3.1 Overview of the System

Since it first appeared in China at the end of the year 2019, COVID-19 has spread quickly to other countries and is now one of the major causes of mortality and health disasters globally.

The system consists of four parts: Data Set, Preprocessing, Trend Prediction, and Risk Prediction. Using Min-Max Normalization for Data Preprocessing is described. The proposed system was to design a predictive model based on an artificial neural network (ANN) model using a backpropagation algorithm to predict the future number of confirmed cases, recovered cases, and death cases caused by COVID-19. And then, the results from the prediction model were used to classify the risk of each country with a fuzzy inference system. Therefore, the backpropagation algorithm and fuzzy inference system play two significant roles in this system to calculate the risk of COVID-19 for ASEAN countries.

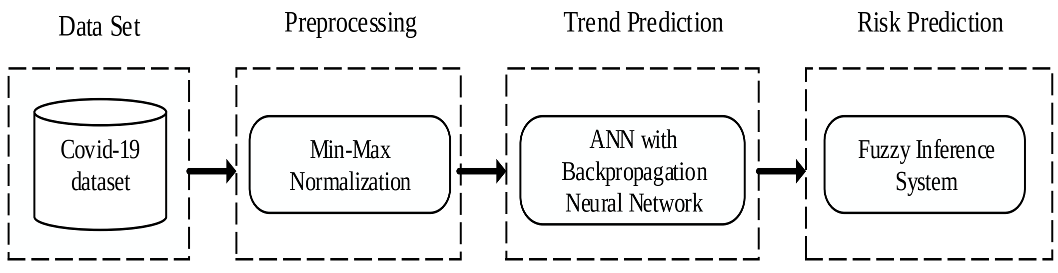

Figure 3.1 Overview Block Diagram of the Proposed System

The overview of the system is described in Figure 3.1. The Johns Hopkins University Center for Systems Science and Engineering offered a GitHub repository with information about the number of COVID-19 patients, that the system used. The COVID-19 dataset consists of date, country, confirmed cases, recovered cases, and total death. The data starts from 22-01-2020 to 31-12-2020. Firstly, the data was preprocessed using data normalization.

3.2 Data Preprocessing

The technique of preparing raw data to make it appropriate for building and training models is known as data preprocessing. Data preparation can be done using a variety of techniques. To remove noise and fix discrepancies in data, data cleaning can be utilized. The process of combing data from several sources into a cohesive data storage, such as a data warehouse, is known as data integration. Aggregation, the removal of redundant features, or clustering are all examples of ways that data reduction might lower the quantity of the data. Data transformations can be used to scale data to a narrower range, such as 0.0 to 1.0.

Normalization is a data preprocessing technique that allows changing the values of numeric columns in a dataset to a common scale. Although normalization is not required for all machine learning datasets, it is used whenever the attributes of the dataset have different ranges. Data is rescaled from its original range during the normalization process so that all values fall between 0 and 1.

Normalization can be useful and even necessary in some machine learning techniques when time-series data includes input values with varying scales. It may be required for algorithms that use distance calculations, such as k-Nearest Neighbors, as well as Linear Regression and Artificial Neural Networks, that weight input values. It can be useful for forecasting and prediction. It is applied when the attributes of the dataset have different ranges. It supports improving the performance and reliability of a machine learning model [11].

There are numerous methods for normalizing data.

- min-max normalization,
- z-score normalization,
- normalization by decimal scaling

In the proposed system, Min-Max Normalization was used for the data preprocessing step because this technique uses minimum and max values for scaling of model and is helpful when features are of different scales.

3.2.1 Min-Max Normalization

Data normalization methods include min-max normalization, z-score, decimal scaling, and normalization using standard deviation. One of the most popular methods for normalizing data into a more useful format is called Min-Max Normalization [10]. The minimum value of every feature is converted to a 0, its maximum value is converted to a 1, and all other values are converted to a decimal between 0 and 1.

Min-Max Normalization formula:

$$v' = \frac{v - min\ (A)}{max(A) - min\ (A)}(\text{new\_Max(A)} - \text{new\_Min(A)}) + \text{new\_Min(A)} \qquad 3.1$$

where A is the attribute data, min(A), max(A) are the min and max absolute value of A, $\boldsymbol{v'}$ is the new value of each entry in data, $\boldsymbol{v}$ is the old value of each entry in data, new_Max(A), new_Min(A) is the max and min value of the range. Min-Max Normalization can be performed as follows: For example:

Confirm Cases: 20, 17, 18, 14, 15, 25, 55

Min: The minimum value of the given attribute is 14.

Max: The maximum value of the given attributes is 55.

v : v is the respective value of the attribute is $v1 = 20$, $v2 = 17, v3 = 18,\ v4 = 14,\ v5 = 15\ , v6 = 25$ and $v7 = 55$

new_Max = 1

new_Min = 0

$$v' = \frac{v - min\ (A)}{max(A) - min\ (A)}(\text{new\_Max(A)} - \text{new\_Min(A)}) + \text{new\_Min(A)}$$

$$v' = \frac{20\ - 14}{55\ - 14}\ (1 - 0) + 0$$

$$v' = 0.146$$

After the number of confirmed cases of COVID-19 was normalized using Min-Max Normalization, the example dataset is shown in Table 3.1.

Table 3.1 The example dataset after being normalized

Date	Confirm Case	Normalized
1/22/2020	20	0.146
1/23/2020	17	0.073
1/24/2020	18	0.098
1/25/2020	14	0
1/26/2020	15	0.0244
1/27/2020	25	0.268
1/28/2020	55	1

3.3 Backpropagation Neural Network

One of the most fundamental concepts in a neural network, backpropagation is the foundation of neural net training. Backpropagation is used to determine the mathematical gradient of a loss function relative to the other weights in the neural network [14]. The calculations are then used to reduce the weight of network nodes with higher error rates compared to nodes with lower error rates. To improve the outputs, backpropagation uses a technique known as the chain rule. To modify the model's weights, the algorithm performs a backward pass through a network after each forward pass.

Backpropagation is mostly used by data scientists to show how modifying a weight function impacts loss functions and the general behavior of the neural network. Backpropagation is a method for optimizing how precisely or accurately a neural network processes specific input. Backpropagation uses gradient descent; the gradient of the output loss function is computed and distributed back through the layers of a deep neural network. The outcomes are modified weights for neurons. Although backpropagation can be used in both supervised and unsupervised learning, it is typically referred to as a supervised learning method because each input value must initially have a known, the desired result to calculate a loss function gradient.

In typical backpropagation-based neural network training, there are two main steps, the forward pass, and the backward pass. During the forward pass, the variables are forward propagated through the network, while in the backward pass, the error is backward propagated through the network.

3.3.1 Forward Propagation

Forward propagation is important in Neural Networks because it helps determine whether assigned weights are good to learn for the given problem statement. The structure of an Artificial Neuron with bias is shown in Figure 3.2. Technically, forward propagation consists of two major steps:

1. Sum the product: It means multiplying the weight vector by the given input vector. And, then it continues until the final layer, where the decision is made.
2. Pass the sum through the activation function: For each layer, the output layer is determined by the sum of the product of the weight and the input vector. And, then the output of 1 layer is fed into the input of the next layer, which is then multiplied by the weight vector in that layer [13]. And, this process keeps on till the output layer activation function.

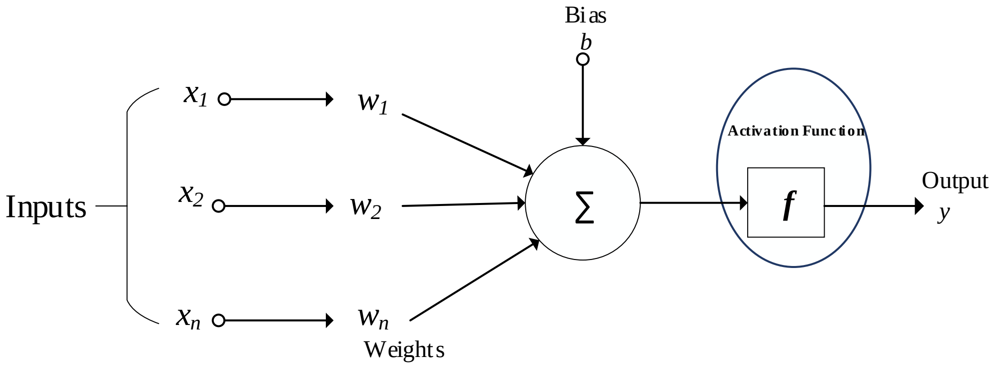

Figure 3.2 Structure of an Artificial Neuron with Bias

3.3.2 Backward Propagation

Backward propagation is an important approach for training the feed-forward network in machine learning. After passing through the forward network, the predicted output is compared to the desired output. The total loss can then be calculated, indicating whether the model is good to use or not. If this is not the case, the loss value is used to recalculate the weights for the forward pass. Back-propagation makes this weight recalculation process simple and efficient. The basic concept of backward propagation in neural network is shown in Figure 3.3.

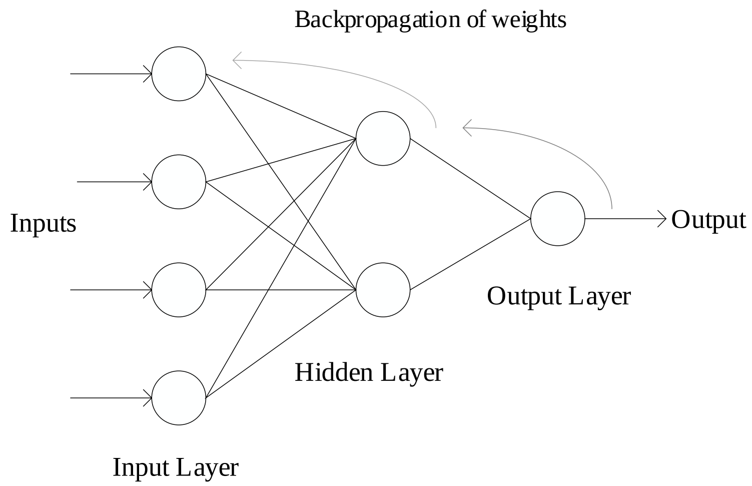

Figure 3.3 Backward Propagation in Neural Networks

Backpropagation is used to calculate the gradient of the loss function for the network weights.

- Compute the Errors: In the forward pass, there is a difference between expected and predicted output. Every single forward pass of this procedure will result in an Error/Loss. The model can be stored to test with test data if the loss/error is acceptable.
- Get the delta: To calculate the delta, multiply the error by the derivative.
- Sum the product: To get the sum of the product, multiply the delta by the input vector and so on.

Backpropagation takes place to calculate the delta after the forward pass and cost calculation (error/loss). It uses a partial derivative to calculate the delta and starts from the output to the last layer [6].

3.3.3 Loss Function

A machine learning algorithm is optimized using a loss function. The performance of the model in the training and validation data, which are used to calculate the loss, determines how the loss should be interpreted [5]. For each training or validation set, it represents the total number of errors made. After each optimization iteration, the loss value of the model shows how well or poorly it performs.

All that is required to train a model is to learn appropriate values for each weight and bias from labeled samples. A poor prediction will result in loss. A loss is a numerical measure of how wrong the model's forecast was in one particular instance [20]. If the prediction of the model is accurate, the loss is 0, but if it is not, the loss is higher. To identify a set of weights and biases that, on average, produce low loss across all cases, a model must be trained. The bigger the loss, the worse the model performs.

The performance of the model for these two sets is determined by the loss, which is calculated for both training and validation sets. A loss is not a percentage, unlike accuracy. It is the sum of the errors in the training or validation sets for each example. The left graph on the following two figures shows a high loss, and the right graph shows a low loss, representing the losses of two different models. The arrows indicate losses, whereas the blue lines indicate predictions. Figure 3.4 describes losses and accuracy of two different models [16].

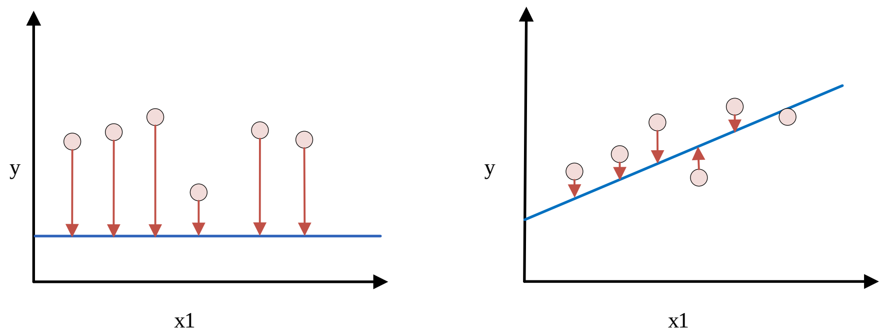

Figure 3.4 Losses and Accuracy of two Different Models

3.3.4 Cost Function

The effectiveness of a machine learning model on a particular dataset is determined by a cost function, which is a crucial parameter. It computes and displays the difference as a single real number between the predicted value and the expected value. The cost function, defined simply, measures how inaccurately the model predicts the link between the parameters X(input) and Y(output). A cost function often referred to as a loss function, can be constructed by iteratively running the model and contrasting estimated forecasts with known Y values. The primary objective of each machine learning model is to identify the parameters or weights that minimize the cost function [19].

The terms cost and loss functions have nearly identical meanings. However, the loss function only applies to a single training set, but the cost function considers a penalty for many training sets or the complete batch. It is also referred to as an error function. The loss function is a part of the cost function. An average of loss functions is used to calculate the cost function. The loss function is a value that is calculated in every instance. So, whereas the loss function is calculated numerous times for a single training cycle, the cost function is only calculated once.

The simple formula of cost function:

$$C = \frac{1}{N} \sum_{i=1}^{n} (y - y')^2 \quad 3.2$$

Where y is the predicted output and y' is the actual output.

3.3.5 Working of Backpropagation Algorithm

The backpropagation process is used to optimize the weights, enabling the neural network to learn how to properly map any inputs to outputs. Some data are required to work with to calculate initial weights, biases, and training input and output. The example of a multilayer feedforward neural network is described in Figure 3.5.

Inputs (i1) : 0.05 Output (o1) : 0.01

Input (i2) : 0.10 Output (o2) : 0.99

Step 1: The forward pass:

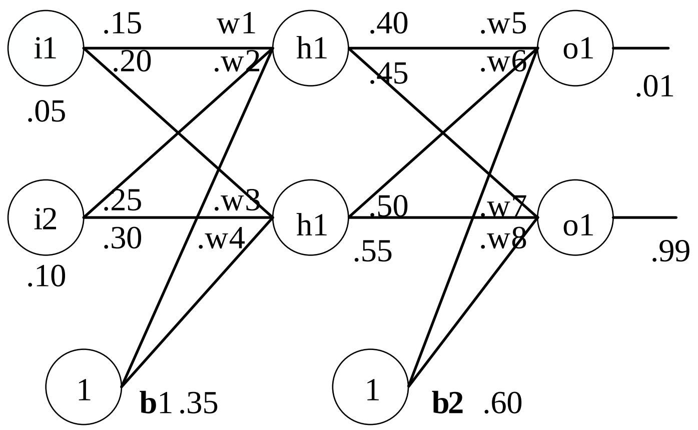

Figure 3.5 Example of a Multilayer Feedforward Neural Network

For the total net input h1:

The net input for h1 is computed as the product of the weight values and the associated input values, adding the bias value.

$$net\ h_1 = w_1 * i_1 + w_2 * i_2 + b_1 * 1 \quad 3.3$$

$$= 0.15 * 0.05 + 0.2 * 0.1 + 0.35 * 1 = 0.3775$$

For the output h1:

Applying a sigmoid function to h1's net input results in the calculation of h1's output. The range of 0 to 1 is the range for which the sigmoid function pumps the data. It is used in models that must predict probability.

$$out\ h_1 = \frac{1}{1 + e^{-net\ h1}} \qquad 3.4$$

$$= 0.5933$$

For the output h2:

$$out\ h_2 = 0.5969$$

For the output o1:

$$net\ o_1 = w_5 * out\ h_1 + w_6 * out\ h_1 + b_2 * 1 \qquad 3.5$$

$$= 0.4 * 0.5933 + 0.45 * 0.5969 + 0.6 * 1 = 1.1059$$

$$out\ o_1 = 0.7514$$

For the output o_2:

$$out\ o_2 = 0.7729$$

Calculating the total error:

$$E\ total = \sum \frac{1}{2}\ (target - output)^2 \qquad 3.6$$

The target output for o_1 = 0.01, the neural network output = 0.7514,

Error for o_1:

$$E\ o_1 = \frac{1}{2}\ (target\ o_1 - output\ o_1)^2 \qquad 3.7$$

$$= \frac{1}{2}\ (0.01 - 0.7514)^2 = 0.2748$$

Error for o_2:

$$E\ o_2 = 0.0236$$

Then, Total Error:

$$E\ total = E\ o_1 + E\ o_2 = 0.2748 + 0.0236 = 0.2984$$

Step 2: Backward Propagation: The process of backward propagation in neural network is shown in Figure 3.6.

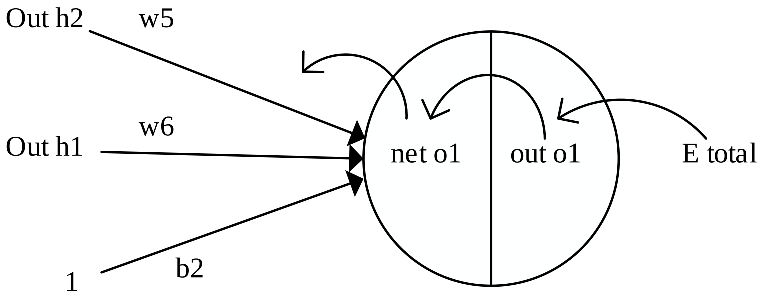

Figure 3.6 The Process of Backward Propagation in Neural Network

Using the rate of change of error with respect to the change in weight W5:

$$\frac{\delta E\ total}{(\delta w_5)} = \frac{\delta E\ total}{(\delta out\ o1)} * \frac{\delta out\ o1}{(\delta net\ o1)} * \frac{\delta net\ o1}{(\delta w_5)} \qquad 3.8$$

Using the outputs $o_1$and o_2 as inputs, calculate the change in total errors as follows:

$$\frac{\delta E\ total}{(\delta out\ o_1)} = -(target\ o_1 - out\ o_1) = -(0.01 - 0.7514) = 0.7414$$

The change in the output o1 relative to its total net can be calculated as follows:

$$out\ o_1 = \frac{1}{(1 + e^{-net\ o1})}$$

$$\frac{\delta out\ o_1}{(\delta net\ o_1)} = out\ o_1\ (\ 1 - out\ o_1\ ) =\ 0.7514\ (\ 1 - 0.7514\ ) = 0.1868$$

The total net input of o1 changes in relation to w5:

$$net\ o_1 = w_5 * out\ h_1 +\ w_6 * out\ h_2 +\ b_2 * 1 \qquad 3.9$$

$$\frac{\delta net\ o_1}{(\delta w_5)} = 1 * out\ h_1 *\ w_5^{(1-1)} + 0 + 0 = out\ h_1 = 0.5933$$

By adding up all of the values and determining the updated weight value:

$$\frac{\delta E\ total}{(\delta w_5)} = \frac{\delta E\ total}{(\delta out\ o1)} * \frac{\delta out\ o1}{(\delta net\ o1)} * \frac{\delta net\ o1}{(\delta w_5)} = 0.0822$$

For w_5 :

$$w_5^{+} = w_5 - n * \frac{\delta E\ total}{(\delta w_5)} = 0.4 - 0.5 * 0.0822 = 0.3589$$

The other weight values can be calculated in the same way. Then, using a gained propagation forward, calculate the output and the error once more. The process will stop here if the error is minimal; otherwise, it will propagate backward and change the weight values. Up until there is very little error, this procedure will be repeated.

3.4 Gradient Descent

The most often used optimization strategy in deep learning and machine learning is gradient descent. It can be combined with any method, it is used to train the model, and it is simple to understand and implement. Gradient quantifies how much a function's output changes when its inputs are slightly altered. A gradient can be used to define the slope of a function [11]. The formula for Gradient Descent:

$$b = a - \gamma \nabla f(a) \tag{3.10}$$

where, *b* = next value, *a* = current value, and '−' refers to the minimization part of the gradient descent. γ in the middle is the learning rate, and the gradient term $\nabla f(a)$ is simply the direction of the steepest descent. Figure 3.7 shows the reducing loss with GD.

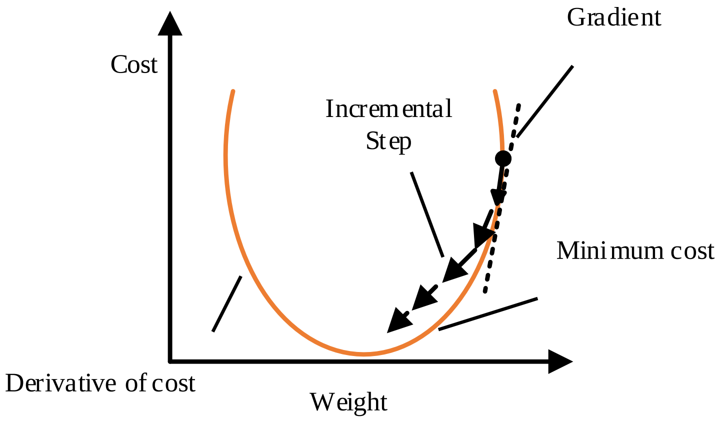

Figure 3.7 Reducing Loss with Gradient Descent

Gradient descent can be thought of as climbing down to the bottom of a valley rather than up a hill. This is because it minimizes a specified function using a minimization method. Batch gradient descent, stochastic gradient descent, and mini-batch gradient descent are the three (3) main types of gradient descent.

3.4.1 Batch Gradient Descent

Batch gradient descent is another name for gradient descent. This optimization approach has long been employed in data science and machine learning. The gradient is computed using the entire dataset or training set to find the optimal solution. Batch gradient descent is very slow because the gradient must be calculated on the complete dataset to perform just one update, which is a difficult task if the dataset is large.

- After the initialization of parameters, the cost function is calculated.
- It loads into memory every record that was read from the disk.
- Once the iteration of the sigma calculation is complete, go one step forward and continue the procedure.

3.4.2 Stochastic Gradient Descent

SGD is an optimization algorithm that saves both time and processing space to find the best optimal solution. The gradient of the cost function for updating each parameter is determined using just one training example in each iteration of stochastic gradient descent. Because only one training example is used for each cycle, it is also faster for larger datasets.

Each example in the dataset is chosen at random once every iteration via the stochastic gradient descent technique. With faster iteration and lower convergence rates, this random approximation of the data set avoids the computing cost of gradient descent. The technique simply takes one random stochastic gradient descent, iterates, and enhances it before moving on to the next random example. Although it simply takes and iterates one sample at a time, the result is produced with more noise.

3.4.3 Mini-batch Gradient Descent

Mini-batch gradient descent is the transitional phase between batch gradient descent and stochastic gradient descent. Mini-batch gradient descent simply divides the dataset into smaller batches, whereas batch gradient descent must go through the complete training set in each iteration and then pick one sample at a time in stochastic. As a result, it doesn't examine the full sample at once or examines each example separately. This gives the algorithm a balance, enabling it to determine both the robustness of stochastic and the processing efficiency of batch gradient descent.

It is a popular algorithm that produces faster and more accurate results. In this example, the dataset is split into small groups of 'n' training datasets. It is faster because it does not use the complete dataset. For every iteration, calculate the gradient of the cost function using a batch of 'n' training datasets. It results in more reliable convergence by lowering the variance of parameter changes. To speed up the computation of the gradient, it can also use a highly optimized matrix.

3.5 Basic Conceptions of Fuzzy Logic

To classify the risk categories of each country in ten Asian countries, the proposed system used fuzzy inference system. The fundamental approach of fuzzy inference entails taking the input variable, passing it through a mechanism made up of parallel if-then rules and fuzzy logical operations, and finally arriving at the output space. The If-Then rules are expressed in human language, and each word is regarded as a fuzzy set. All of these fuzzy sets need to be defined by membership functions before they can be utilized to build If-Then rules.

3.5.1 Fuzzy Set

The extension of classical set theory is fuzzy sets theory. The degree of membership in elements varies. A logic based on True and False is sometimes not enough to describe human thought. By using the entire range between 0 (false) and 1, fuzzy logic simulates human reasoning (true). Any set having a membership function that spans the range [0,1] and permits its members to have variable degrees of membership is said to be fuzzy.

From fuzzy set theory, fuzzy logic is derived. Numerous degrees of membership are allowed (between 0 and 1), and a membership function A(x) is connected to a fuzzy set A such that it maps every component of the universe of discourse X to the range [0,1]. The mapping is written as [0,1]: A (x): X. If X is a discourse universe and x is a particular element of X, then a fuzzy set A defined on X can be described as a collection of ordered pairs.

$$A = \{(x, \mu_{\tilde{A}}(x)), x \in X\} \qquad 3.11$$

3.5.2 Membership Functions

By giving each element a membership value, or degree of membership, a membership function (MF) is a curve that identifies the characteristic of a fuzzy set. It converts each input point into a membership value inside the [0, 1] unit interval. The degree to which an element resembles a fuzzy set is measured using a membership function [23]. Membership functions can either be created via machine learning techniques like artificial neural networks, genetic algorithms, etc., or they can be chosen arbitrarily by the user depending on the user's experience.

The five most common membership functions are, in general, the Sigmoidal membership function, the Generalized Bell membership function, the Gaussian membership function, and the Trapezoidal membership function [8]. Regardless of the structure of the fuzzy set, a single membership function can only define one [17]. Typically, several membership functions are used to describe a single input variable. In the proposed system, the rate of COVID-19 is represented by a three-level fuzzy system with fuzzy sets Low, Medium, and High.

3.5.3 Logical Operation

The membership values of standard binary logic are always 1 (true) or 0 (false). The most fundamental logical operations are AND, OR, and NOT. The operands A and B are membership values in the range [0, 1], in contrast to ordinary logical operations. A AND B is equal to min (A, B) because, in fuzzy logical operations, logical AND is expressed by function min (A, B). As logical OR is defined by the function max, A OR B becomes equivalent to max (A, B). Furthermore, the logical NOT convert operation NOT A into operation 1 – A. Fuzzy logical operation AND, OR and NOT are shown in Figure 3.8.

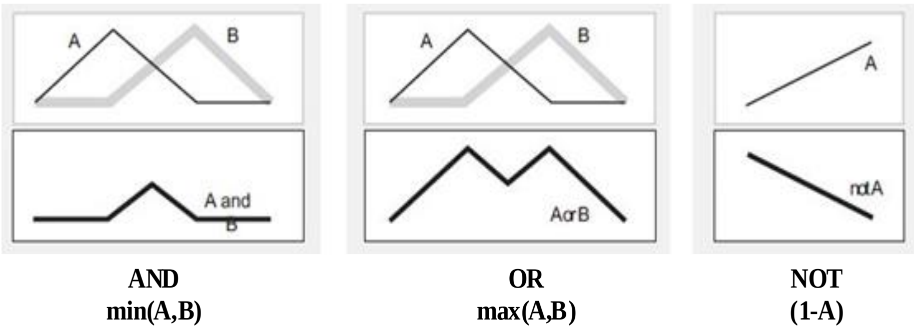

Figure 3.8 Fuzzy logical operation AND, OR and NOT

3.5.4 If-Then Rules

Parallel if-Then rules are used in the fuzzy inference process to determine how to project input variables onto the output space. The following is a description of the form of a single fuzzy if-then rule:

If x is A, Then y is B

The remainder of the Then-part is the consequent, and y is the output variable. If-Then conditional statements can be used in every situation because both A and B are linguistic values and they function the same way as human judgment. For example, an appropriate If-Then rule might be "IF death rate is high AND case rate is high AND recovery rate is low THEN risk is high".

A can be thought of as a fuzzy set with a defined membership function, whereas B, depending on the fuzzy inference method, can either be a fuzzy set or a polynomial to input x. The goal of the If- portion of the antecedent is to determine the membership value of input variable x that corresponds to fuzzy set A. In contrast, the Then-part gives the output variable y a crisp value.

3.6 Fuzzy Inference System

Fuzzy inference is the act of mapping given input variables to an output space using a fuzzy logic-based deducing mechanism that incorporates membership functions, if-then rules, and fuzzy logical operations. Mamdani fuzzy inference, Sugeno fuzzy inference, and Tsukamoto fuzzy inference are the three main categories of fuzzy inference techniques. These three techniques can be separated into two different procedures. The three methods are the same in this first process, which involves fuzzifying the membership values of the input variables from their crisp values using the proper fuzzy sets. When all of the rule outputs are aggregated into a single precise output value in the second procedure, differences appear.

The consequence of the if-Then rule is defined as a fuzzy set in Mamdani inference. The output of each rule will result in a reshaped fuzzy set, and defuzzification is required after collecting all of these reshaped fuzzy sets [7]. However, in Sugeno inference, the output of each rule is a single integer since a polynomial is used to explain the If-Then rule's result to the input variables. The weighting technique is then used to achieve the final crisp result.

Sugeno inference does not require complicated defuzzification, but specifying the polynomial parameters is more time-consuming and challenging than constructing the output fuzzy sets for Mamdani inference. As a result, Mamdani inference is more common, and the proposed system only uses this approach. Although it is much less obvious, Tsukamoto's inference looks to be a combination of Mamdani and Sugeno approaches [3].

3.7 Mamdani-Type Fuzzy Inference Process

The Mamdani-type fuzzy inference procedure consists of five phases. The process of fuzzy inference system is shown in Figure 3.9.

Step 1: Fuzzy Input (Fuzzification)

Step 2: Apply Fuzzy Operator

Step 3: Apply Implication Method

Step 4: Apply Aggregation methods

Step 5: Defuzzify (Defuzzification)

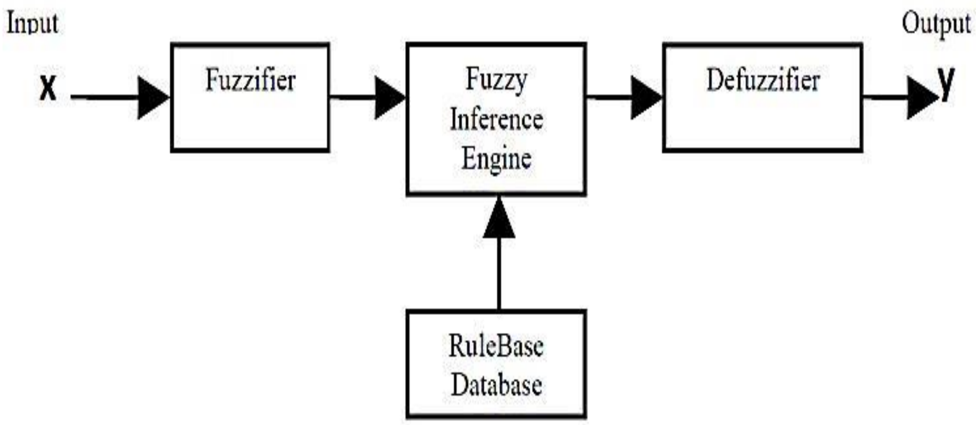

Figure 3.9 Fuzzy Inference System

3.7.1 Fuzzy Input (Fuzzification)

Using membership functions, the input variables' crisp numerical values are first converted into the corresponding membership values of the relevant fuzzy sets. Regardless of what the input variables indicate, the result of the fuzzification process is often a level of membership in the associated fuzzy linguistic sets in the range of 0 to 1 [10] Three input variables—the case rate, recovery rate, and death rate—must be fuzzified in the system using the membership functions of linguistic sets. For example, Figure 3.10 shows that if the input Recovery rate is High then the result of Fuzzification is 0.7. For fuzzifying, the example of input variable “Recovery rate” is described in Figure 3.10.

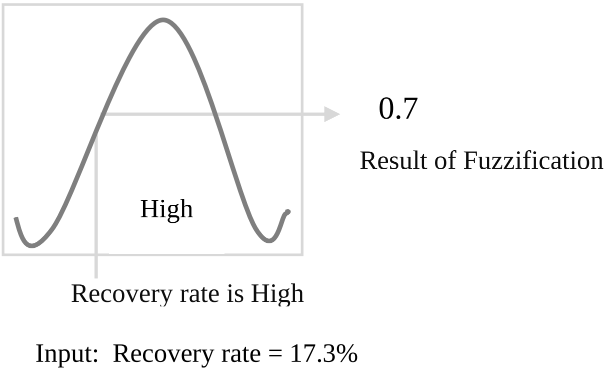

Figure 3.10 Fuzzifying Input Variable "Recovery rate"

3.7.2 Apply Fuzzy Operator

When the fuzzy inference system contains more than one input variable, the antecedent of the If-Then rule may always be determined by more than one fuzzy linguistic set because each input variable has a corresponding fuzzy set based on which to establish the degree of membership. The AND and OR operations are the two most popular fuzzy operators. These logical operations are expressed using the functions min and max.

The AND operation using function min is demonstrated in the example below. The fuzzy membership values produced by the antecedent in Rule's three different fuzzy sets were 0.1, 0.5, and 0.7, respectively. The minimum of these three values, 0.1, is chosen as the outcome of the antecedent in rule.

$$\mu\text{ (death rate == high) = 0.1 } \mu\text{(case rate == high) = 0.5 } \mu\text{(recovery rate == low) = 0.7}$$

$$\min(0.1, 0.5, 0.7) = 0.1$$

3.7.3 Apply Implication Method

The consequent part of the If-Then rule is another fuzzy linguistic set that is specified by a suitable membership function. The inference method in the Then-part reshapes the fuzzy set of the consequent part according to the result of the antecedent, or a single number, unlike the antecedent part of the if-Then rule, which yields a single numerical value. This strategy is known as the implication method. The fuzzy set of consequent parts is condensed using the AND operator. The fuzzy set of each rule output should be quantified by a single number from its corresponding antecedent. Figure 3.11 shows the example of applying implication method.

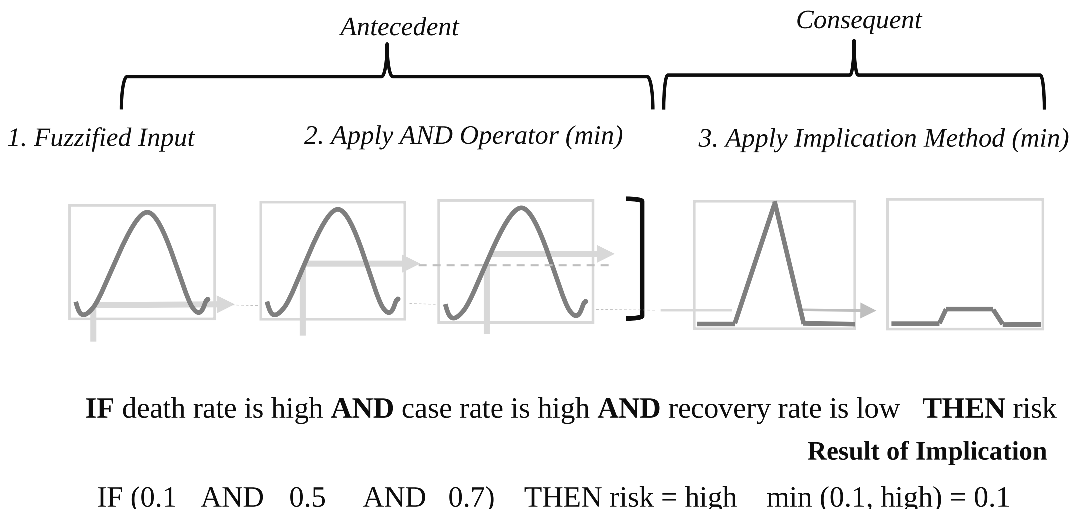

Figure 3.11 Applying Implication Method

3.7.4 Apply Aggregation Method

The aggregation strategy is used to assess if each If-Then rule generates a modified fuzzy set as an output by combining the fuzzy sets that reflect the outputs of rules into a single fuzzy set. All three functions—max, sum, and probabilistic OR—can be used for aggregating operations, but only function max is used in the system because it is more understandable and widely used.

In the example below, function max operates on five truncated fuzzy sets resulting from five rules in turn, and a combined new fuzzy set reflecting the result for the output variable "risk evaluation" is prepared for the final defuzzification process. The example of applying aggregation method is shown in Figure 3.12.

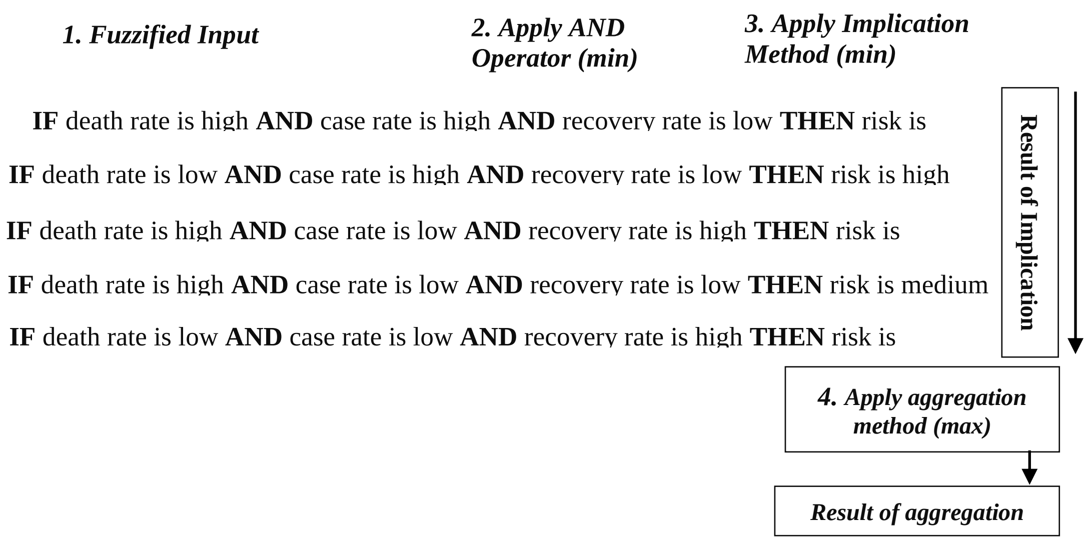

Figure 3.12 Applying Aggregation Method

3.7.5 Defuzzify (Defuzzification)

The final stage of the fuzzy inference process, known as defuzzification, turns the aggregated fuzzy set from the aggregation phase into a single output. The last technique extracts a precise quantity from the fuzzy set range to the output variable because the crisp values of the input variables are fuzzified into the degree of membership concerning fuzzy sets in the first procedure [18].

There are five built-in defuzzification methods supported: centroid, bisector, middle of maximum, largest of maximum, and smallest of maximum. Among the many defuzzification methods that have been proposed in the system, the single approach used in the system is the Centroid Method, which is the most popular and physically pleasing of all the defuzzification techniques. The equation of the centroid method is as followed:

$$Output = \frac{\sum_{i=1}^{N}(center_i * strength_i)}{\sum_{i=1}^{N} strength_i} \qquad 3.12$$

3.8 Summary

In this chapter, implementing the predictive model using the backpropagation algorithm and its steps are described. Moreover, the steps of the Fuzzy Inference System are also presented to calculate the country-level risk factors. The backpropagation neural network and fuzzy inference system play significant roles in the proposed system to calculate the risk country of COVID-19. In the proposed system, data set, preprocessing, trend prediction, and risk prediction steps are included. All the steps of the proposed system are explained in detail in this chapter.

CHAPTER 4
SYSTEM DESIGN AND IMPLEMENTATION

The detailed implementation of the proposed system for calculating the risk of each country in ten Asian countries is presented in this chapter. The design of the system and the comparison of the performance of the prediction methods are described. Moreover, it presents a graphical user interface of the system with step-by-step detailed explanation figures and experimental results of the system. This system in the Python environment is a tool used for predicting the number of COVID-19 sufferers and correlates with the risk of each country.

4.1 System Flow Diagram

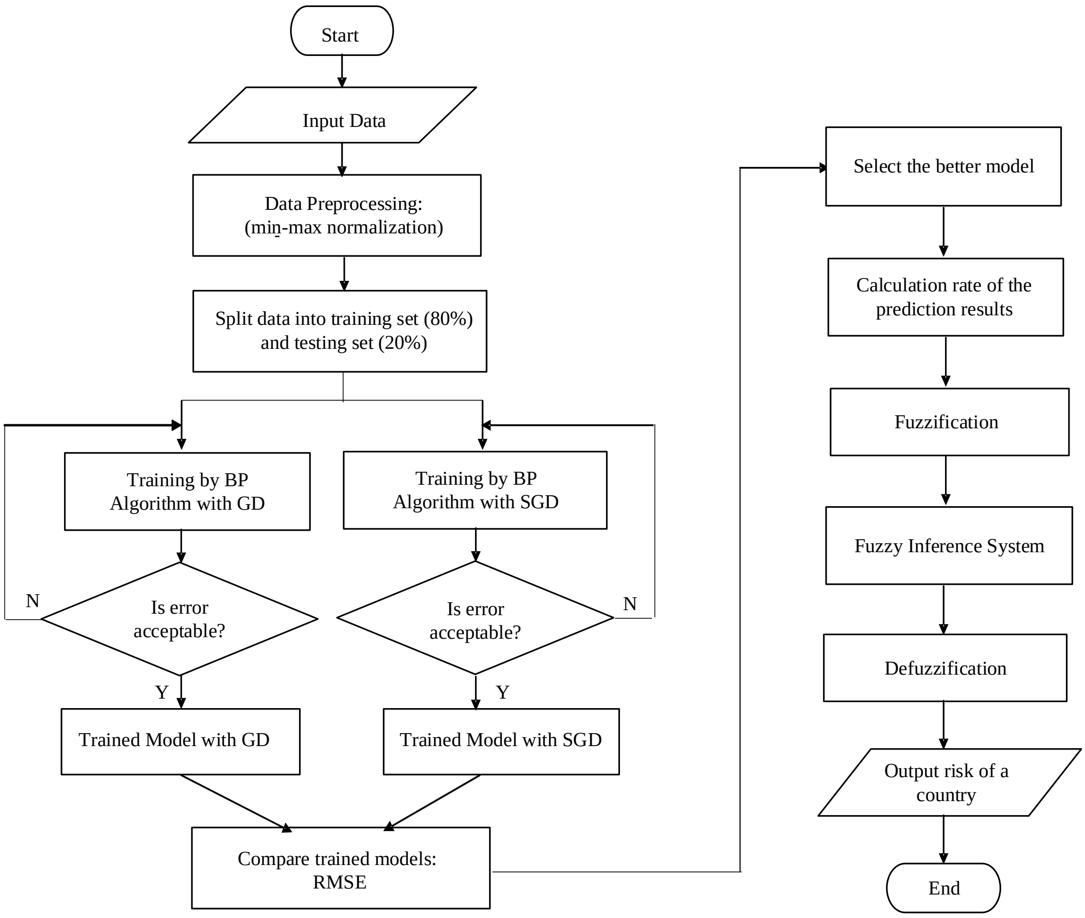

Figure 4.1 System Flow Diagram of Risk Calculation

The system flow diagram of risk calculation is shown in Figure 4.1. The main point of the proposed system is to learn the backpropagation neural network with the gradient descent method and the backpropagation neural network with the stochastic gradient descent method. Among them, the method that achieved the best performance was used to predict the future number of COVID-19 cases. In this system, there are five parts: COVID-19 dataset, data preprocessing, implementation of a predictive model, calculation of the rate of prediction results, and prediction of the risk categories of each country.

4.2 Implementation of the system

The proposed system is implemented to calculate the risk categories of each country in ASEAN Countries using a backpropagation neural network and fuzzy inference system. This system includes two main parts: the first part is the implementation of the prediction model using a backpropagation neural network to predict the future number of COVID-19 sufferers such as confirmed cases, recovered cases, and death cases. In the second part, these prediction results are used to calculate the risk categories of each country with a fuzzy inference system.

Raw data for the system consists of daily confirmed, recovered and dead COVID-19 data recorded by the Center for Systems Science and Engineering (CSSE) at Johns Hopkins University (JHU). These data were preprocessed by using min-max normalization and then split into two parts: the training dataset and the test dataset. After that, the predictive model was constructed using backpropagation neural network with GD and SGD methods. After building the model, it was used to forecast the future trend of the COVID-19 spread. The performance of the model was evaluated by the Root Mean Square Error (RMSE). The output results from the predictive model were used to decide the risk of each country with a fuzzy inference system. The performance of the proposed system was checked by using exising data from the trend data.

4.2.1 COVID-19 dataset for the Proposed System

In the first part, the COVID-19 dataset, which consists of the number of positive cases, the number of recoveries, and the number of deaths for ASEAN countries are used for the future predicting of COVID-19 spread. The Center for Systems Science and Engineering at Johns Hopkins University offered a GitHub repository from which the dataset was collected. The repository was largely made available by the university for the visual dashboard of the 2019 Novel Coronavirus [11].

Table 4.1 COVID-19 confirmed cases time-series worldwide

Province/State	Country/ Region	Lat	Long	1/22/20	1/23/20	. . .	3/27/20
NaN Victoria NaN	Afghan Australia Algeria	33.00 - 37.81 28.30	68.00 145.96 1.95	0 0 0	0 0 0	. . . .	75 511 364

Table 4.2 COVID-19 recovered cases time-series worldwide

Province/ State	Country/ Region	Lat	Long	1/22/20	1/23/20	. . .	3/27/20
Colombia Victoria NaN	Canada Australia Algeria	49.28 - 37.81 28.03	-123.1 144.96 1.65	0 0 0	0 0 0	. . . .	4 70 65

Table 4.3 COVID-19 deaths cases time-series worldwide

Province/State	Country/ Region	Lat	Long	1/22/20	1/23/20	. . .	3/27/20
Northern Territory Diamond Princess NaN	Australia Canada Algeria	-12.46 0.000 28.30	130.84 0.000 1.65	0 0 0	0 0 0	. . . .	0 1 19

Data samples from the files are shown in Tables 4.1, 4.2, and 4.3 respectively. These data samples are time-series COVID-19 global datasets. Among them, the system only used the dataset from ASEAN countries: Brunei, Cambodia, Indonesia, Laos, Malaysia, Myanmar, Philippines, Singapore, Thailand, and Vietnam. Therefore, the number of confirmed cases, recovery cases, and death cases of each country in ASEAN countries were prepared as the following Table 4.4.

Table 4.4 Example dataset of COVID-19 for Laos

No.	Date	Confirmed Case	Recovered Case	Dead Case
1	1/22/2020	3	0	3
2	1/23/2020	6	0	6
3	1/24/2020	6	0	6
4	1/25/2020	8	0	8
5	1/26/2020	8	0	8
6	1/27/2020	8	0	8
7	1/28/2020	9	0	9
8	1/29/2020	10	0	10
9	1/30/2020	10	0	10
10	1/31/2020	10	0	10
11	2/1/2020	10	0	10
12	2/2/2020	11	0	11
13	2/3/2020	12	0	12
14	2/4/2020	14	0	14
15	2/5/2020	15	0	15
16	2/6/2020	16	0	16
17	2/7/2020	16	0	16
18	2/8/2020	18	0	18
19	2/9/2020	19	0	19
20	2/10/2020	3	0	3

4.2.2 Data Preprocessing

After manually preparing the dataset for each country from the global dataset, the second part is data preprocessing by using data normalization. Preparing raw data to make it usable for model construction and training is known as data preprocessing. Data preprocessing techniques include normalization. The term "normalization" refers to the rescaling of data from the original range to a new range between 0 and 1.

Normalization helps speed up the learning step of the backpropagation technique. There are many methods for data normalization such as min-max normalization, z-score normalization, and normalization by decimal scaling. etc [26]. Among them, min-max normalization was used in the proposed system.

The example COVID-19 dataset for Laos after being normalized is shown in Table 4.5.

Table 4.5 COVID-19 dataset for Laos after being normalized

No	Confirm Case	Normalized
1	109538	0.81190379
2	112093	0.83084164
3	114371	0.84772635
4	116924	0.86664937
5	118892	0.88123633
6	120600	0.89389616
7	121957	0.90395434
.	.	.
.	.	.
.	.	.
799	133726	0.99118704
800	134412	0.99627173
801	134915	1

4.2.3 Implementation of Backpropagation Neural Network with GD

After preprocessing steps, our dataset was split into two parts: 80% and 20%. The first part is used as a training dataset and the second one is used as a test dataset. And then, the backpropagation neural network was used to build the predictive model. There are several types of ANN algorithms depending on the prediction algorithm utilized, one of which is Backpropagation.

Backpropagation is a technique for solving prediction problems that work well, although its effectiveness depends on the optimization technique utilized during training. The gradient descent method is commonly used as an optimization method.

The disadvantage of this method is that it takes a long time to reach a point of convergence. However, the stochastic gradient descent method has a higher rate of convergence than the gradient descent method [27].

In the proposed system, Both the backpropagation neural network with gradient descent and the backpropagation neural network with stochastic gradient descent is used to implement the prediction model. Among them, the method with the best result was used to predict the future trend data of COVID-19 such as the number of confirmed cases, recovered cases, and death cases. The flowchart of the backpropagation algorithm is presented in Figure 4.2.

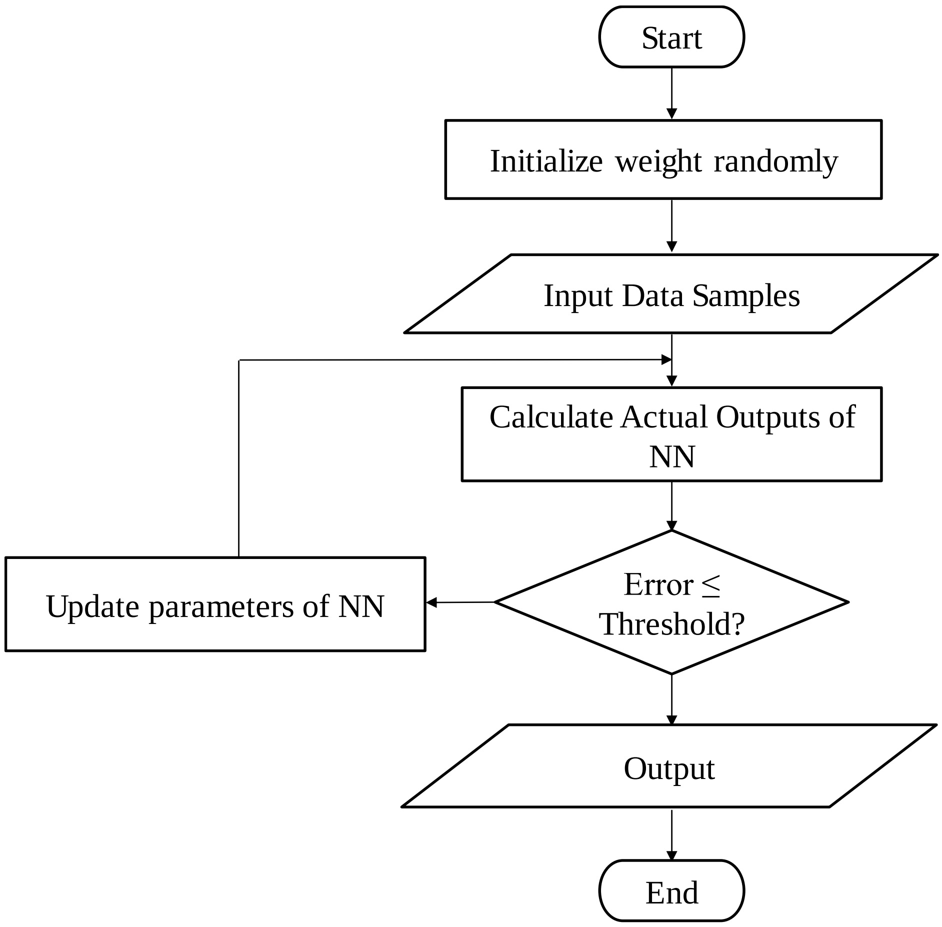

Figure 4.2 The Flowchart of the Backpropagation Algorithm

The backpropagation algorithm carries out learning on a multilayer feed-forward neural network. It iteratively learns a set of weights for determining the class label of tuples. A multilayer feed-forward neural network consists of an input layer, one or more hidden layers, and an output layer. Multilayer feed-forward neural network is shown in Figure 4.3.

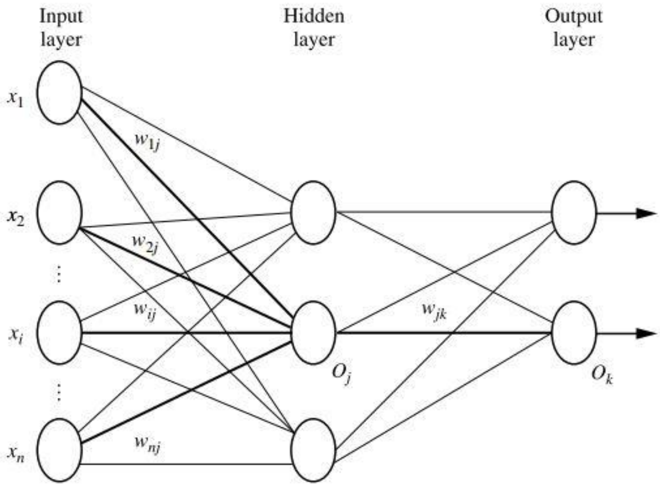

Figure 4.3 Multilayer Feed-forward Neural Network

Backpropagation learns by iteratively analyzing a data set of training tuples and comparing the network's prediction for each tuple with the actual known target value. For classification problems, the goal value could either be a continuous integer or the known class label from the training tuple (for numeric prediction). To lower the mean-squared error between the network's prediction and the actual target value, the weights are modified for each training tuple. These modifications are being made "backward." The weights will eventually converge, but there is no assurance of this, and learning will cease.

To implement the predictive model, the backpropagation with gradient descent method performs the following steps that describe the equations used to train a neural network.

Input: Network is a multilayer feed-forward network [9]. D is a data set made up of training tuples and the target values that go with them.

Output: A trained neural network.

Initialize the weights: The weights of the network are initialized to small random numbers (e.g., ranging from -1.0 to 1.0 or -0.5 to 0.5). There is a bias associated with each unit. Similarly, the biases are initialized to small random values.

Each training tuple, X, is processed by the following processes.

Propagate the inputs forward: The input layer of the network receives the training tuple initially.

As the inputs move through the input units, they remain unchanged. That is, the output, O_j, of an input unit, j, is equal to the input value, I_j. The individual units of the net input and output at the hidden and output layers are then calculated. The input units of the hidden or output layers are combined linearly to determine their net input. The unit's net input is calculated by multiplying each input by its appropriate weight and summing the results. The net input, I_j, to unit j in a hidden or output layer is given a unit, j.

$$I_j = \sum_i w_{ij} O_i + \theta_j, \tag{4.1}$$

where w_{ij} is the weight of the connection from unit i in the previous layer to unit j; O_i is the output of unit i from the previous layer; and θ_j is the bias of the unit. Each unit in the hidden and output layers takes its net input and then applies an activation function to it. The proposed system used the logistic, or sigmoid, function. Given the net input I_j to unit j, then O_j , the output of unit j, is computed as

$$O_j = \frac{1}{1+e^{-I_j}}, \tag{4.2}$$

Backpropagate the error: The error is propagated backward by changing the weights and biases to reflect the prediction error of the network. The error Err_j is calculated for a unit j in the output layer by:

$$Err_j = O_j(1 - O_j)(T_j - O_j), \tag{4.3}$$

where O_j is the actual output of unit j, and T_j is the known target value of the given training tuple. $O_j(1 - O_j)$ is the derivative of the logistic function. And then, to compute the error of a hidden layer unit j used the following equation.

$$Err_j = O_j(1 - O_j) \sum_k Err_k w_{jk}, \tag{4.4}$$

where w_{jk} is the weight of the connection from unit j to a unit k in the next higher layer, and Err_k is the error of unit k. To reflect the propagated errors, the weights and biases are then modified. Weights are updated by the following equations, where w_{ij} is the change in weight w_{ij}:

$$\Delta w_{ij} = (l) Err_j O_i. \tag{4.5}$$

$$w_{ij} = w_{ij} + \Delta w_{ij}. \tag{4.6}$$

The learning rate, or variable l, is a constant with a usual range of 0.0 to 1.0. Backpropagation learns by applying the gradient descent approach to find a set of weights that minimizes the mean-squared difference between the network's class prediction and the known target value of the tuples. Learning will go very slowly if the learning rate is too low. If the learning rate is too large, then oscillation between inadequate solutions may occur.

Biases are updated by the following equations, where θ_j is the change in bias θ_j :

$$\Delta\theta_j = (l)Err_j. \quad 4.7$$

$$\theta_j = \theta_j + \Delta\theta_j. \quad 4.8$$

The above-mentioned steps were performed until the terminating condition is satisfied. There are the three stages of the Backpropagation training process: feeding-forward step from pattern input training, related error backpropagation, and weight update. In the next step, the output was calculated by counting each input unit in the hidden layers. During the training process, the network's output was compared to the target, and the error was calculated. The optimization is then used to identify the variables that contribute to error spread. This parameter updates the weight between the input layer and the output layer.

The weights are incrementally adjusted after each epoch run over the training dataset using the backpropagation neural network using the Gradient Descent optimization algorithm. By taking in the opposite direction of the cost gradient, the size and direction of the weight update are calculated.

$$\Delta w_j = -\eta \frac{\partial J}{\partial w_j}, \quad 4.9$$

The following update rule is then used to update the weights after each epoch:

$$w := w + \Delta w, \quad 4.10$$

where Δw is a vector that contains the weight updates of each weight coefficient w, which are computed as follows:

$$\Delta w_j = -\eta \frac{\partial J}{\partial w_j}$$

$$= -\eta \sum \left(target^{(i)} - output^{(i)}\right)\left(-x_j{}^{(i)}\right)$$

$$= \eta \sum \left(target^{(i)} - output^{(i)}\right)x_j{}^{(i)}.$$

4.2.4 Implementation of Backpropagation Neural Network with SGD

A randomized version of the gradient descent is called stochastic gradient descent. Apply the gradient descent technique to one sample alone. In the stochastic gradient descent, a single sample is selected at random for each iteration rather than the entire data set.

Gradient descent optimization, also known as batch gradient descent, calculates the cost gradient using the entire training set. Because Gradient Descent only takes one step for each run over the training set, it can be rather costly when dealing with very large datasets. As a result, the method may update weights more slowly and may take longer to converge to the global cost minimum as the training set size increases. The stochastic gradient descent doesn't accumulate the weight updates as seen above for Gradient Descent:

- for any number of epochs:
    - for each weight j

$$w_j := w + \Delta w_j, \text{ where: } \Delta w_j = \eta \sum \left(target^{(i)} output^{(i)} \right) x_j{}^{(i)}$$

Instead, the weight is updated after each training sample:

- for an epoch or epochs, or until about Cost-minimum is reached:
    - for training sample i:
        - for each weight j

$$w_j := w + \Delta w_j, \text{ where: } \Delta w_j = \eta \left(target^{(i)} output^{(i)} \right) x_j{}^{(i)}$$

Because the network processes only one training sample, the stochastic gradient descent is easier to fit into memory. Moreover, it can converge faster for larger datasets because the parameters are updated more frequently. The proposed system implemented the predictive model using both the backpropagation neural network with gradient descent and the backpropagation neural network with stochastic gradient descent.

4.2.5 Fuzzy Inference System-Based Risk Categorization

The prediction of the number of cases, the number recovered and the number of deaths from the predictive model was used to predict the risk of the country with a fuzzy inference system. The risk categories of each country were defined into three classes (1) high risk (HR), (2) medium risk (MR), and (3) low risk (LR). Thus, first, the new case rate, recovery rate, and death rate were calculated as the following equations:

$$Confirm\ Rate = \frac{Number\ of\ cases}{Population} * 100 \qquad 4.11$$

$$Recovery\ Rate = \frac{Number\ of\ recoveries}{Number\ of\ cases} * 100 \qquad 4.12$$

$$Death\ Rate = \frac{Number\ of\ deaths}{Number\ of\ cases} * 100 \qquad 4.13$$

The three fuzzy membership functions are then defined to represent these parameters' risk measurement. For the fuzzy inference system, the variable, the fuzzy set, and the range are described in Table 4.6.

Table 4.6 The variable, fuzzy set, and the range for the fuzzy inference system

Variable	Fuzzy Set	Range
Case Rate	Low	[0 – 0.01]
	Medium	[0.01 – 0.1]
	High	>0.1
Recovery Rate	Low	[0 – 40]
	Medium	[40 – 70]
	High	>70
Death Rate	Low	[0 – 1]
	Medium	[1 – 2]
	High	>2

The risk measurement of the case rate, recovery rate, and the death rate is then defined using three membership functions, as shown in Figure 4.4.

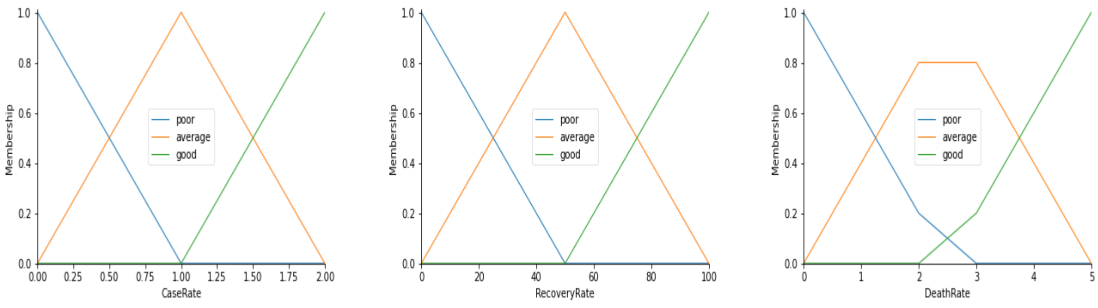

Figure 4.4 Fuzzy Membership Function for Case Rate, Recovery Rate and Death Rate

The imposing rules, which are described in Table 4.7, estimate the final class of risk.

Table 4.7 Fuzzy rule to estimate the risk factor of a country

Case Rate	Recovery Rate	Death Rate	Decision
High	High	High	HR
High	High	Medium	HR
High	High	Low	HR
High	Medium	High	HR
High	Medium	Medium	HR
High	Medium	Low	MR
High	Low	High	HR
High	Low	Medium	HR
High	Low	Low	HR
Medium	High	High	HR
Medium	High	Medium	HR
Medium	High	Low	LR
Medium	Medium	High	HR
Medium	Medium	Medium	HR
Medium	Medium	Low	LR
Medium	Low	High	HR
Medium	Low	Medium	HR
Medium	Low	Low	HR
Low	High	High	HR
Low	High	Medium	HR
Low	High	Low	LR
Low	Medium	High	HR
Low	Medium	Medium	HR
Low	Medium	Low	LR
Low	Low	High	HR
Low	Low	Medium	HR
Low	Low	Low	MR

4.3 Experimental Results

Firstly, the proposed system implements the predictive models using both Backpropagation neural network with gradient descent and Backpropagation neural network with stochastic gradient descent to predict the number of COVID-19 sufferers. Ten datasets were used in the experiment, ASEAN countries, from Jan. 22, 2020, to Dec. 31, 2020, provided by CSSE, Johns Hopkins University.

Figure 4.5 shows the calculation rate of prediction results based on the upcoming 30 days using backpropagation algorithm with a gradient descent optimizer by the user interface. It displays only the prediction result rate for Brunei among ASEAN countries. By selecting the other countries from the dropdown button of Country, their prediction results rate can also be calculated.

The calculation rate of prediction results based on the upcoming 30 days using backpropagation algorithm with a stochastic gradient descent optimizer is shown by the user interface in Figure 4.6.

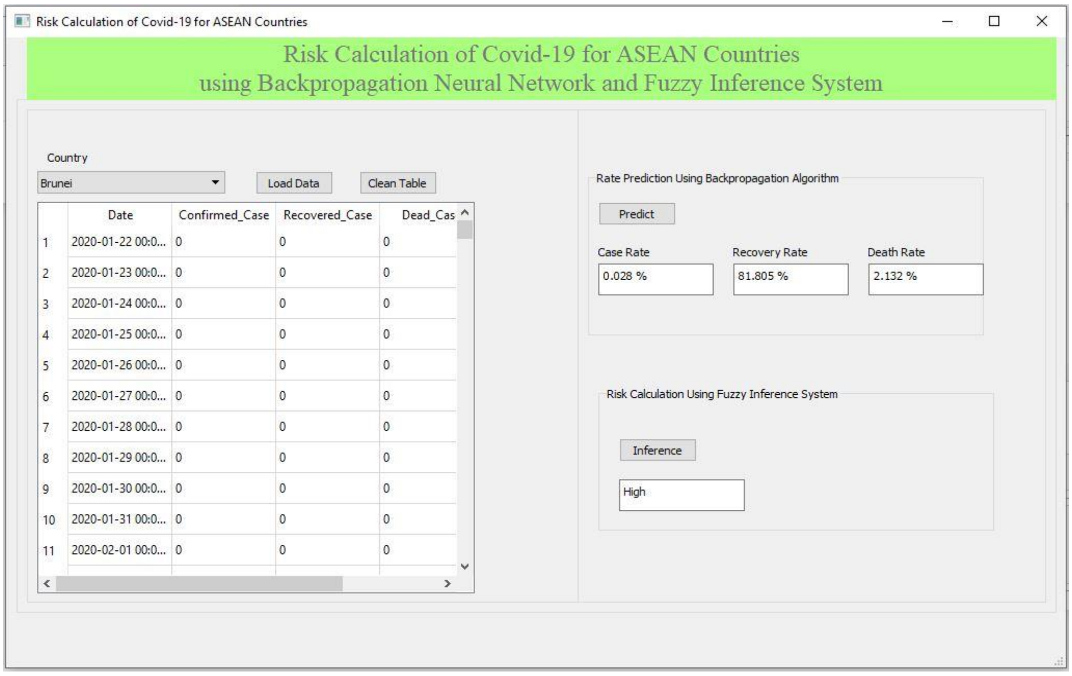

Figure 4.5 The Prediction Results Based on the Upcoming 30 Days Using Backpropagation Algorithm with Gradient Descent Optimizer

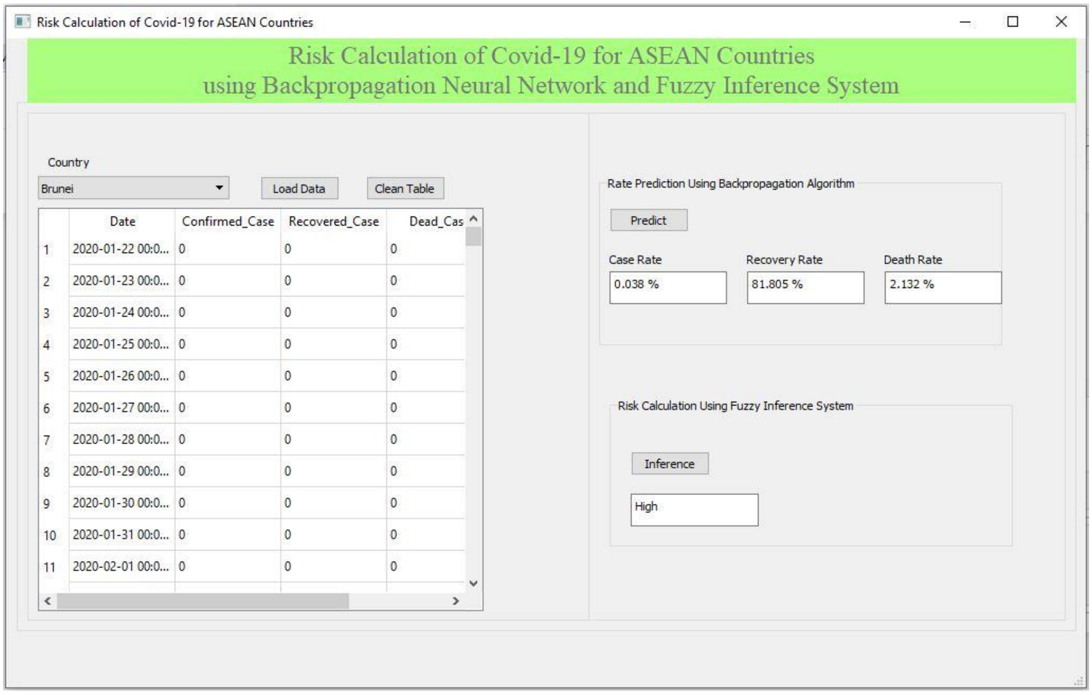

Figure 4.6 The Prediction Results Based on the Upcoming 30 Days Using Backpropagation Algorithm with the Stochastic Gradient Descent Optimizer

The performance comparison between the gradient descent and stochastic gradient descent for optimizing Backpropagation ANN with several learning rates is shown in Table 4.8.

Table 4.8 Evaluation of the performance of the prediction method by using RMSE for several learning rates

RMSE		
Learning Rate	Gradient Descent Method	Stochastic Gradient Descent Method
0.0005	0.379	0.134
0.0003	0.492	0.364
0.0002	0.558	0.468
0.001	0.178	0.137
0.01	0.631	0.184

Root-Mean-Square Error was used to evaluate the performance of the prediction methods. In Table 4.8, the root-mean-square error (RMSE) resulting from the Backpropagation neural network using the stochastic gradient descent decreases for several learning rates. Moreover, the gradient descent and stochastic gradient descent were compared for ASEAN countries with a learning rate of 0.0005. The comparison of the performance of the prediction method for ASEAN countries with a learning rate of 0.0005 is shown in Figure 4.7.

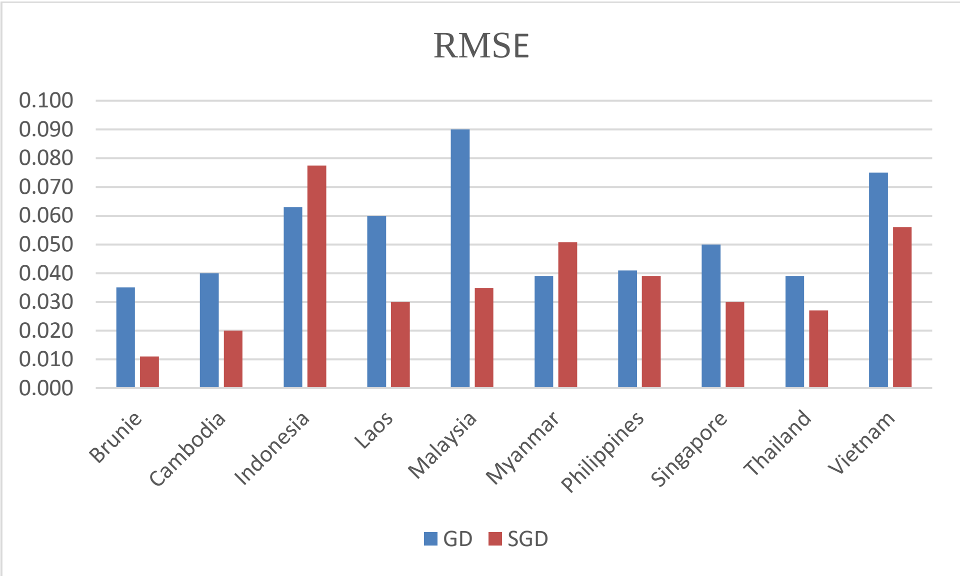

Figure 4.7 Comparison of the Performance of the Prediction Method for ASEAN Countries with a Learning Rate of 0.0005

From Table 4.8 and Figure 4.7, it can be concluded that the stochastic gradient descent method has much better accuracy than the gradient descent method. So, the model that was implemented using the Backpropagation neural network with stochastic gradient descent was used to predict the number of COVID-19 sufferers in ASEAN countries. Figure 4.8 shows the initial best fit line and final best fit line on the training dataset.

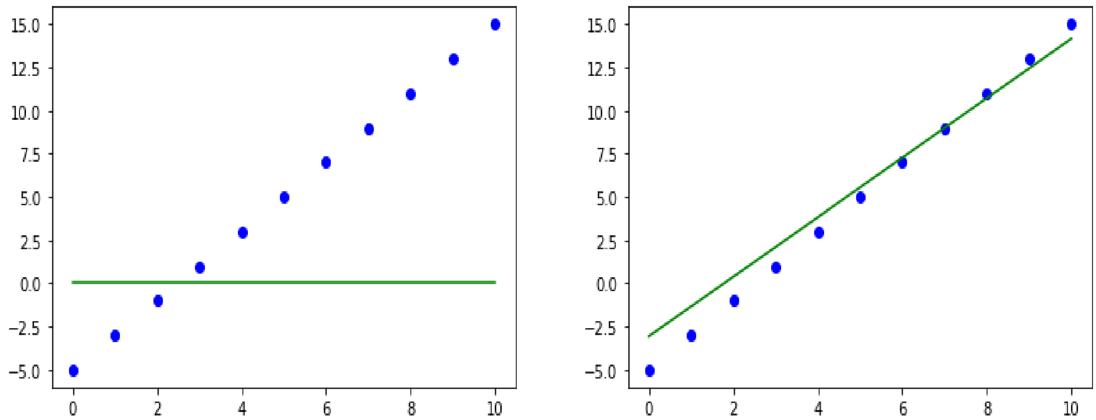

Figure 4.8 On the Training Dataset, (a) Initial Best Fit Line and (b) Final Best Fit Line

In Figure 4.9, the curve illustrating the loss function convergence rate for each iteration, the convergence rate on the training dataset is displayed.

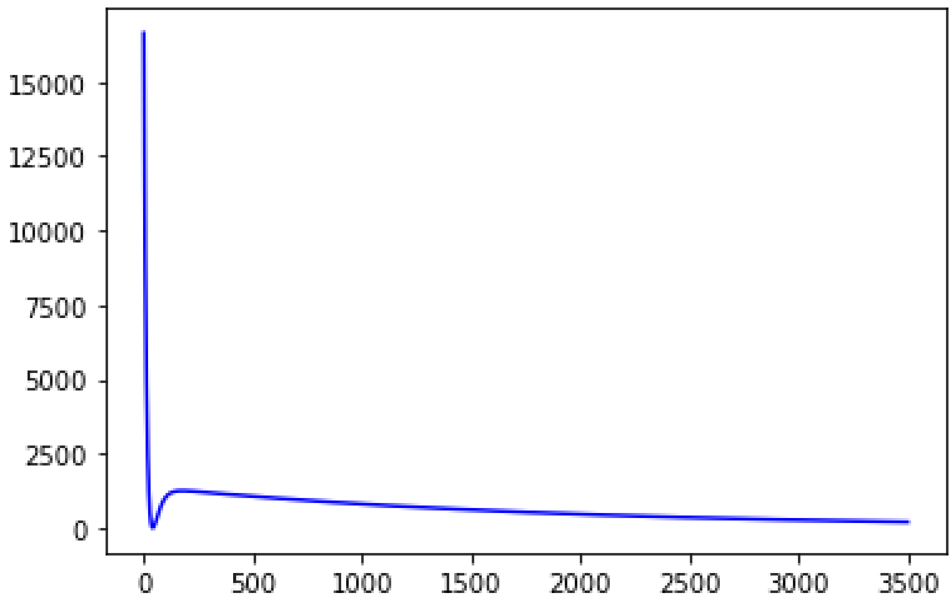

Figure 4.9 Convergence Rate of Loss Function on the Training Dataset

In Figure 4.10, the number of actual cases in Thailand from 22 January 2020 to 31 December 2020 is shown for all three categories i.e. confirmed cases, recoveries, and deaths.

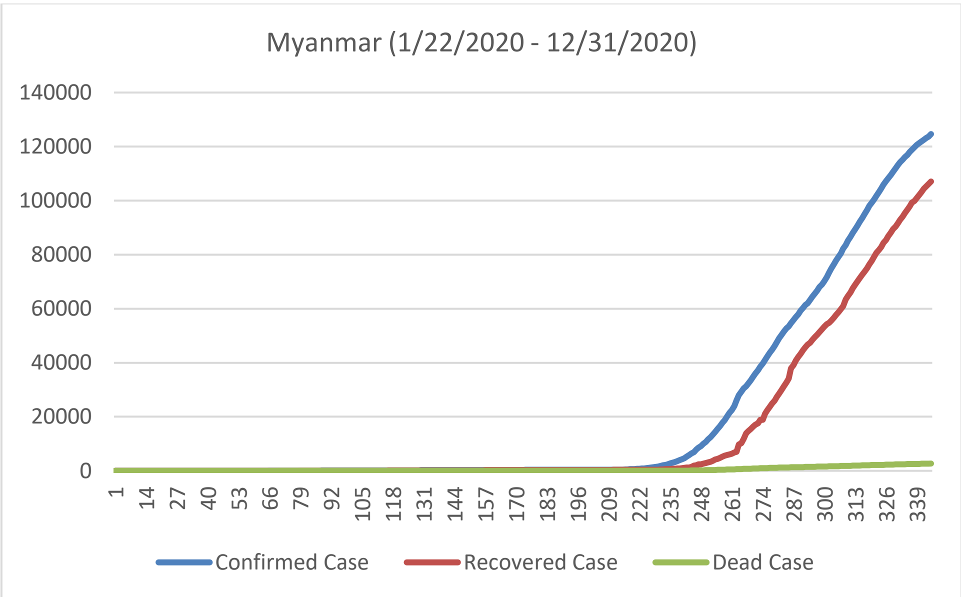

Figure 4.10 The Number of Confirmed Cases, Recovery and Deaths in Myanmar (1/22/2020 – 12/31/2020)

The prediction trends of Myanmar by backpropagation neural network with the Stochastic gradient descent method for the upcoming 30 days are shown in Figures 4.11, 4.12 and 4.13. With training data, the accuracy of the model can be seen, and the difference between ANN-predicted and actual results can be seen.

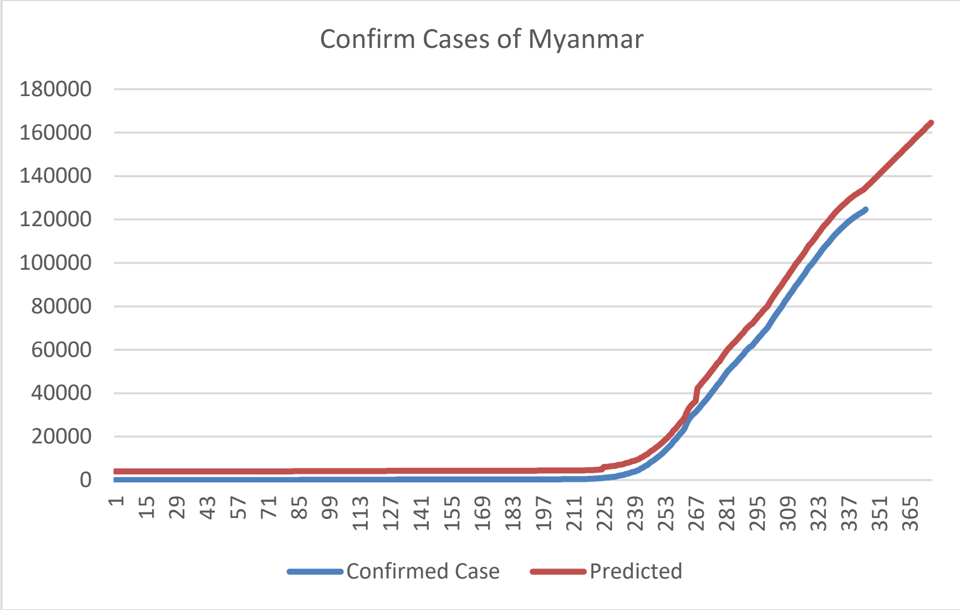

Figure 4.11 Confirm Cases Prediction of Myanmar by Backpropagation with SGD for the Upcoming 30 Days

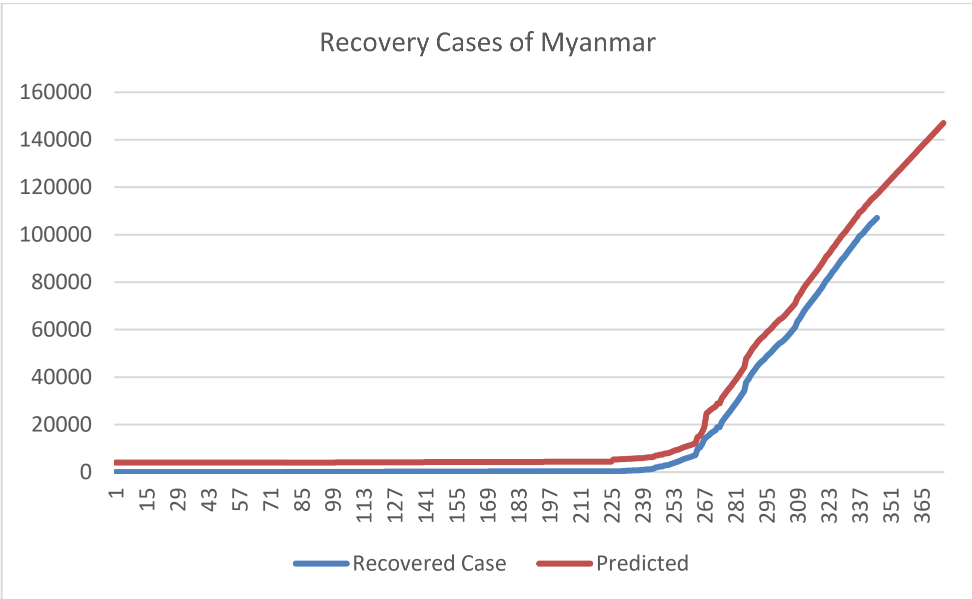

Figure 4.12 Recovery Cases Prediction of Myanmar by Backpropagation with SGD for the Upcoming 30 Days

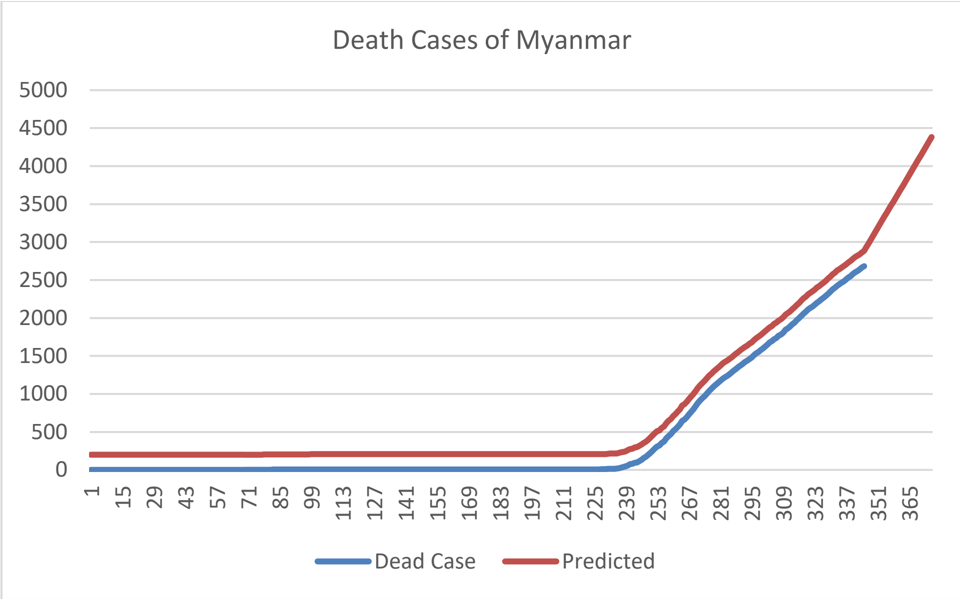

Figure 4.13 Death Cases Prediction of Myanmar by Backpropagation with SGD for the Upcoming 30 Days

And then, the prediction results from the predictive model implemented by using backpropagation algorithm with stochastic gradient descent are used to classify the risk of each country with a fuzzy inference system. The final fuzzy-rule-based classification depends on case rate, recovery rate, and death rate. The categories of risks were defined into three classes such as high risk (HR), medium risk (MR), and low risk (LR). The proposed system has used 30 days ahead to calculate such risk classes. For ASEAN countries, the results of the proposed system and the results calculated using the actual data are shown in Table 4.9. It shows the predicted risk categories for the upcoming 30 days using COVID-19 datasets for 2020.

Table 4.9 Comparison of risk factors for ASEAN countries based on upcoming 30 days using COVID-19 datasets for 2020

Country	Actual Risk	Predicted Risk
Brunei	High Risk	High Risk
Cambodia	High Risk	High Risk
Indonesia	High Risk	High Risk
Laos	High Risk	High Risk
Malaysia	Low Risk	Low Risk
Myanmar	High Risk	High Risk
Philippines	High Risk	High Risk

Country	Actual Risk	Predicted Risk
Singapore	Low Risk	Low Risk
Thailand	Low Risk	Low Risk
Vietnam	High Risk	High Risk

Table 4.10 shows the predicted risk categories for the upcoming 30 days using COVID-19 datasets of 2021.

Table 4.10 Comparison of risk factors for ASEAN countries based on upcoming 30 days using COVID-19 datasets for 2021

Country	Actual Risk	Predicted Risk
Brunei	Low Risk	Medium Risk
Cambodia	High Risk	High Risk
Indonesia	High Risk	High Risk
Laos	Low Risk	Low Risk
Malaysia	High Risk	High Risk
Myanmar	High Risk	High Risk
Philippines	High Risk	High Risk
Singapore	Low Risk	Low Risk
Thailand	Low Risk	Low Risk
Vietnam	High Risk	High Risk

The performance of the proposed system is calculated many times by using the preexisting actual trend data. Moreover, the system has experimented with each year's COVID-19 datasets to predict the next year’s risk categories. Tabel4.11 shows Comparison of risk factors for ASEAN countries using COVID-19 datasets of 2020 for several days.

Table 4.11 Comparison of risk factors for ASEAN countries using COVID-19 datasets of 2020 for several days

COVID-19 datasets of 2020 (SGD Learning rate 0.0005)						
Country	30 days		40 days		365 days	
	Actual	Predicted	Actual	Predicted	Actual	Predicted
Brunei	HR	HR	HR	HR	LR	HR
Cambodia	HR	HR	HR	HR	HR	HR
Indonesia	HR	HR	HR	HR	HR	HR
Laos	LR	LR	LR	LR	LR	HR
Malaysia	LR	LR	LR	LR	HR	LR
Myanmar	HR	HR	HR	HR	HR	HR
Philippines	HR	HR	HR	HR	HR	HR
Singapore	LR	LR	MR	LR	LR	HR
Thailand	LR	LR	LR	LR	LR	LR
Vietnam	HR	HR	HR	HR	HR	HR

The comparison of risk factors for ASEAN countries using COVID-19 datasets of 2021 for several days is shown in Table 4.12.

Table 4.12 Comparison of risk factors for ASEAN countries using COVID-19 datasets of 2021 for several days

COVID-19 datasets of 2021 (SGD Learning rate 0.0005)				
Country	30 days		40 days	
	Actual	Predicted	Actual	Predicted
Brunei	LR	LR	LR	LR
Cambodia	HR	HR	HR	HR
Indonesia	HR	HR	HR	HR
Laos	LR	LR	LR	LR
Malaysia	HR	HR	HR	HR
Myanmar	HR	HR	HR	HR
Philippines	HR	HR	HR	HR
Singapore	LR	LR	LR	LR
Thailand	LR	LR	LR	LR
Vietnam	HR	HR	HR	HR

By using COVID-19 datasets for 2020 and 2021, the proposed system has been experimented with fifty times for ASEAN countries, five times per country. Among Fifty times, the system only produced five incorrect results. So, it is observed that the proposed system produces a relatively 90% high accuracy of prediction risk compared with the actual trend risk calculation.

4.4 Comparative Analysis

Many researchers have analyzed the various ways to overcome the COVID-19 problem. M. Pourhomayoun (2021) [16] used several machine learning algorithms, such as Support Vector Machine (SVM), Artificial Neural Networks, Random Forest, Decision Tree, Logistic Regression, and K-Nearest Neighbor (KNN), to implement a prediction model to estimate the mortality rate in COVID-19 patients. Among them, it presented that the neural network algorithm with an accuracy of 90 % has much better accuracy than other algorithms.

To analyze and predict the number of daily confirmed COVID-19 cases, two statistical models—ARIMA and GARCH—and a deep learning model—LSTM DNN—were composed in Meejoung Kim (2021) [15]. The performance of the deep learning model was compared to two statistical models by using mean-square-error. LSTM DNN predicts best for all datasets, according to experimental results, while the predictions of two statistical models are dataset-dependent. Moreover, this research only used data from a limited period, which was insufficient for learning the condition.

Syaiful Anam (2021) [22] considered the Backpropagation neural network with the Fletcher–Reeves method for predicting the number of COVID-19 sufferers in Malang. To optimize Backpropagation ANN using several different architectures and learning rates, the performance comparison of the Fletcher–Reeves and gradient descent methods was demonstrated. As a result, the Backpropagation neural network with the Fletcher–Reeves method achieved better results than the Backpropagation neural network with the gradient descent method.

The proposed system focused to calculate the risk categories of each country in ASEAN Countries using a backpropagation neural network and fuzzy inference system. The prediction models are implemented using both the backpropagation neural network with gradient descent and the backpropagation neural network with stochastic gradient descent. Their performances were evaluated by using RMSE for several learning rates. Moreover, the gradient descent and stochastic gradient descent were compared for ASEAN countries with a learning rate of 0.0005. Among them, the stochastic gradient descent method has much better performance than the gradient descent method. Therefore, the model that was implemented using the Backpropagation neural network with stochastic gradient descent. In addition to this, these prediction results are used to calculate the risk categories of each country in ASEAN countries with a fuzzy inference system. The proposed system has been experimented with many times by using the preexisting trend data. As a result, the proposed system has achieved 90 % accuracy in prediction risk.

4.5 Summary

In this chapter, the implementation of the proposed system is presented in detail the step-by-step. The proposed system is implemented as the risk calculation of COVID-19 for the ASEAN Countries system by using Backpropagation Neural Network and Fuzzy Inference System. The main point of the proposed system is to compare the performance of two prediction methods: Gradient Descent and Stochastic Gradient Descent using the Backpropagation Algorithm. The architecture of the proposed system consists of data preprocessing, implementing the predictive model, calculating the rate for the input variables of the fuzzy inference system, and calculating the risk factors of each country in ASEAN countries. To show the performance of the predicted model, all the country-specific datasets were evaluated by using root-mean-square error (RMSE). To evaluate the performance of the proposed system for Risk Calculation, the experiments have been done many times using the existing actual data. According to the comparison results, the proposed system has achieved high accuracy in many times.

CHAPTER 5
CONCLUSION AND FURTHER EXTENSIONS

The proposed system uses daily case data, an artificial neural network with backpropagation, and a combination of fuzzy inference systems to solve the problem of calculating the long-term risk of a country.

Firstly, the data is preprocessed by using data normalization. The dataset is split into training and test sets with a ratio of 80 to 20. A prediction model is implemented on the training dataset using a Backpropagation neural network with Gradient Descent and Stochastic Gradient Descent Optimizers. And then, the models are tested by using root-mean-square error (RMSE) on validation and 10% test dataset. Experimental results show that the Backpropagation neural network with Stochastics gives better results compared to the Backpropagation neural network with the gradient descent. Therefore, the Backpropagation neural network is optimized by the Stochastics method, which is used to predict the number of COVID-19 sufferers.

The number of cases, the number of recoveries, and the number of deaths that were obtained from the predictive model has been used to calculate the risk of each country. The system introduces a new way to predict an epidemic outbreak and associates it with the risk of a country. In addition, the system presents that an Artificial Neural Network (ANN) can be used to solve regression problems and to predict future health risks.

5.1 Advantages and Limitations of the System

The proposed system calculates the risk of COVID-19 in each country of the ASEAN. The exactness and completeness of the proposed system are proved by comparing it with the preexisting actual trend data.

As a result, the developed system is useful for the Government and other healthcare organizations in being prepared, understanding the real significance of the risk, and taking the best prevention measures.

Moreover, the forecasts of the system could be of considerable assistance to the authorities in helping them decide when to act and how to act to limit the COVID-19 situation as well as for future public safety and pandemic control if the same pattern continues.

However, the proposed system concentrated only on three factors: the number of confirmed cases, the number of recovered cases, and the number of death cases to calculate the risk of COVID-19 for ASEAN countries.

5.2 Further Extensions

The proposed system is calculated by using only three parameters such as confirm cases, recovery cases, and death cases. In future work, the dataset can be extended into a hybrid dataset by comprising heterogeneous parameters, e.g., mobility measures, policy responses, vaccinations, economic and social aspects, etc., to better describe the spread and development of COVID-19. Therefore, the system would help in making decisions for managing parameters and in developing more precise rules and decisions.

AUTHOR'S PUBLICATIONS

[1] Sabai Oo, Htar Htar Lwin, "Risk Calculation of COVID-19 for ASEAN Countries Using Backpropagation Neural Network and Fuzzy Inference System", Parallel and Soft Computing Journal (PSC), UCSY, Yangon, Myanmar, June 2022.

[2] Zayar Aung, Daw Toe Toe, Sabai Oo, "Burmese Text Images Classification of Based on Neural Network", 2020 2nd International Conference on Control Systems, Mathematical Modeling, Automation and Energy Efficiency (SUMMA), IEEE, Lipetsk, Russia, 17 December 2020, pp. 337-340.

[3] Htar Htar Lwin, Sabai Oo, Kyaw Zaw Ye, Kyaw Kyaw Lin, Wai Phyo Aung, Phyo Paing Ko, "Feedback Analysis in Outcome Base Education Using Machine Learning", 2020 17th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), IEEE, Phuket Thailand, 04 August 2020, pp. 767-770.

REFERENCES

[1] "Advances in Computer Science and Information Engineering", Springer Science and Business Media LLC, 2012

[2] "Fuzzy Logic Augmentation of Neural and Optimization Algorithms: Theoretical Aspects and Real Applications", Springer Science and Business Media LLC, 2018

[3] "Intelligent and Fuzzy Techniques in Big Data Analytics and Decision Making", Springer Science and Business Media LLC, 2020

[4] Abdolraouf Samadi-Maybodi, Mohsen Nikou. "Modeling of removal of an organophosphorus pesticide from aqueous solution by amagnetic metal–organic framework composite", Chinese Journal of Chemical Engineering, 2021

[5] Anish Gupta, Ayodeji Olalekan Salau, Prateek Chaturvedi, Segun Ayokunle Akinola, Nnamdi Ikechi Nwulu. "Artificial Neural Networks: Its Techniques and Applications to Forecasting", 2019 International Conference on Automation, Computational and Technology Management (ICACTM), 2019

[6] Annisa Aditsania, Adiwijaya, Aldo Lionel Saonard. "Handling imbalanced data in churn prediction using ADASYN and backpropagation algorithm", 2017 3rd International Conference on Science in Information Technology (ICSITech), 2017

[7] CÖzgen Karacan "A fuzzy logic approach for estimating recovery factors of miscible CO2- EOR projects in the United States", Journal of Petroleum Science and Engineering, 2020

[8] Chun-Tian Cheng, Jian-Yi Lin, Ying-Guang Sun, Kwokwing Chau. "Chapter 145 Long-Term Prediction of Discharges in Manwan Hydropower Using Adaptive-Network-Based Fuzzy Inference Systems Models", Springer Science and Business Media LLC, 2005

[9] Chunxiang Cao, Min Xu, Patcharin Kamsing, Sornkitja Boonprong, Peera Yomwan, Apitach Saokarn. "Environmental Remote Sensing in Flooding Areas", Springer Science and Business Media LLC, 2021

[10] Elia Oey, Benjamin Suwito Rahardjo. "Does culture influence our ways in handling COVID 19?", International Journal of Sociology and Social Policy, 2021

[11] Furqan Rustam, Aijaz Ahmad Reshi, Arif Mehmood, Saleem Ullah, Byungwon On, Waqar Aslam, Gyu Sang Choi. "COVID-19 Future Forecasting Using Supervised Machine Learning Models", IEEE Access, 2020

[12] Guoqing Wu, Yang Wang, Qian Gong, Li Li, Xiaoyan Wu. "An Intelligent Temperature Control Algorithm of Molecular Beam Epitaxy System Based on the Back-Propagation Neural Network", IEEE Access, 2022

[13] Hisham El-Amir, Mahmoud Hamdy. "Deep Learning Pipeline", Springer Science and Business Media LLC, 2020

[14] Khadijeh Alibabaei, Pedro D. Gaspar, Tânia M. Lima. "Crop Yield Estimation Using Deep Learning Based on Climate Big Data and Irrigation Scheduling", Energies, 2021

[15] Meejoung Kim, "Prediction of COVID-19 Confirmed Cases after Vaccination: Based on Statistical and Deep Learning Models", SciMedicine Journal, Published 01 June 2021, Vol.3, No.2

[16] Mohammad Pourhomayoun, and Mahdi Shakibi, "Predicting mortality risk in patients with COVID-19 using machine learning to help medical decision-making", Elsevier Public Health Emergency Collection, Published by Elsevier Inc., 16 January 2021

[17] Maja Kalinic, Jukka M. Krisp. "Fuzzy inference approach in traffic congestion detection", Annals of GIS, 2019

[18] Manasi Dhokare, Sharvari Teje, Shivani Jambukar, Vaishali Wangikar. "Evaluation of Academic Performance of Students Using Fuzzy Logic", 2021 International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA), 2021

[19] Mohamad Sofian Abu Talip, Aisha Hassan Abdalla, Ahmed Asif, Abdurazzag Ali Aburas. "Fuzzy Logic Based Algorithm for Disk Scheduling Policy", 2009 International Conference of Soft Computing and Pattern Recognition, 2009

[20] Nishchal J, neel bhandari. "Computational Complexity of Gradient Descent Algorithm", Institute of Electrical and Electronics Engineers (IEEE), 2021

[21] Annisa Aditsania, Adiwijaya, Aldo Lionel Saonard. "Handling imbalanced data in churn prediction using ADASYN and backpropagation algorithm",

2017 3rd International Conference on Science in Information Technology (ICSITech), 2017

[22] Anam, Maulana, N. Hidayat, I. Yanti, Z. Fitriah, and D. M. Mahanani, “Predicting the Number of COVID-19 Sufferers in Malang City Using the Backpropagation Neural Network with the Fletcher–Reeves Method”, Applied Computational Intelligence and Soft Computing, Published 29 April 202

[23] Shweta Sanghi. "Determining the membership values to optimize retrieval in a fuzzy relational database", Proceedings of the 44th annual southeast regional conference on - ACM-SE 44 ACM-SE 44, 2006

[24] Siddhi Khanse, Payal Bhandari, Rumjhum Singru, Neha Runwal, Atharva Dharane. "Comparative Study of Genetic Algorithm and Artificial Neural Network for Multi-class Classification based on Type-2 Diabetes Treatment Recommendation model", 2020 Sixth International Conference on Parallel, Distributed and Grid Computing (PDGC), 2020

[25] Steven W Smith. "Neural Networks (and more!)", Elsevier BV, 2003

[26] Subrata Saha, Sanguthevar Rajasekaran, Jinbo Bi, Sudipta Pathak. "Efficient techniques for genotype-phenotype correlational analysis", BMC Medical Informatics and Decision Making, 2013

[27] Syaiful Anam, Mochamad Hakim Akbar Assidiq Maulana, Noor Hidayat, Indah Yanti, Zuraidah Fitriah, Dwi Mifta Mahanani. "Predicting the Number of COVID-19 Sufferers in Malang City Using the Backpropagation Neural Network with the Fletcher–Reeves Method", Applied Computational Intelligence and Soft Computing, 2021