ARTIFICIAL NEURAL NETWORK FOR ESTIMATION OF PARAMETERS AND SOLVING DYNAMIC SYSTEM OF MAIZE FOLIAR DISEASE

Deme Dadi Jima

A Thesis Submitted to

the department of Applied Mathematics
School of Applied Natural Science

Presented in Partial Fulfillment of the Requirement for the Degree of Master's in Applied Mathematics

Office of Graduate Studies

Adama Science and Technology University.

June, 2022.

Adama, Ethiopia.

ARTIFICIAL NEURAL NETWORK FOR ESTIMATION OF PARAMETERS AND SOLVING DYNAMIC SYSTEM OF MAIZE FOLIAR DISEASE

Deme Dadi Jima

Advisor: Tamirat Temesgen (PhD)

A Thesis Submitted to

the department of Applied Mathematics
School of Applied Natural Science

Presented in Partial Fulfillment of the Requirement for the Degree of Master's in Applied Mathematics

Office of Graduate Studies

Adama Science and Technology University.

June, 2022.

Adama, Ethiopia.

Declaration

I hereby declare that this Master Thesis entitled "Artificial Neural Network for Estimation of Parameters and Solving Dynamic System of Maize Foliar Disease" is my original work. That is, it has not been submitted for the award of any academic degree, diploma or certificate in any other university. All sources of materials that are used for this thesis have been duly acknowledged through citation.

Name of student Signature Date

Recommendation of Advisors

We, the advisors of this thesis, hereby certify that we have read the revised version of the thesis entitled " Artificial Neural Network for Estimation of Parameters and Solving Dynamic System of Maize Foliar Disease " prepared under our guidance by Deme Dadi Jima submitted in partial fulfillment of the requirements for the degree of Master's of Science in Applied Mathematics. Therefore, we recommend that the submission of revised version of the thesis to the department following the applicable procedures.

Major Advisor	Signature	Date
Co-advisor	Signature	Date

Approval page

We, the advisors of the thesis entitled " Artificial Neural Network for Estimation ff Parameters and Solving Dynamic System of Maize Foliar Disease " and developed by Deme Dadi Jima, hereby certify that the recommendation and suggestion made by the board of examiners are appropriately incorporated into the final version of the thesis.

Major Advisor	Signature	Date

We, the undersigned, members of the Board of Examiners of the thesis by Deme Dadi Jima, have read and evaluated the thesis entitled "Artificial Neural Network for Estimation of Parameters and Solving Dynamic System of Maize Foliar Disease "and examined the candidate during the open defence. This is, therefore, to certify that the thesis is accepted for partial fulfilment of the requirement of the degree of Master of Science in Applied Mathematics.

Chairperson	Signature	Date
Internal Examiner	Signature	Date
External Examiner	Signature	Date

Finally approval and acceptance of the thesis is contingent upon submission of its final copy to the Office of Postgraduate Studies (OPGS) through the Department Graduate Council (DGC) and School Graduate Committee (SGC).

Department Head	Signature	Date
School Dean	Signature	Date
Office of Postgraduate Studies, Dean	Signature	Date

Acknowledgment

I would like to thank my advisor Dr.Tamirat Temesgen , for his helpful discussion, comments, giving direction, and providing the necessary materials in the preparation of this thesis.Secondly, I would like to extend my thanks to my families and all who encouraged me to complete my thesis and their heart full help while writing the thesis .

Lastly, I would like thanks brothers of St.Joseph Catholic School-Adama,for supporting financially and giving the necessary materials throughout the preparation of this thesis.

Contents

List of Tables

List of Figures

Acronyms

ANN- Artifical Nueral Network

DNN - Deep Nueral Network

ODE- Ordinary Differential Equation

PINN- physics-informed neural networks

AD- Automatic Differentiation

RK4- Runge-kutta order four

SER- Susceptible ,Exposed,Recovered

SEIR- Susceptible ,Exposed,Infectious,Recovered

Abstract

In this thesis, we applied Artificial neural network (ANN) for approximating the solution and estimating the parameters of first order systems of differential equations. First, the network trained to determine the learning parameters, the weights and biases. Once the learning parameters are known, it is possible to determine the ANN output and write the solution of the system in terms of the ANN output. We impose to this solution a minimization condition to obtain the unknown parameters of the system. A vectorised algorithm for this method was provide and test on non-linear systems of ODEs. Finally, we tested the accuracy of the ANN approach to approximate the solution and estimating the parameters by comparing it with theoretical values of the systems.

keywords: Artificial Neural Networks, System of Ordinary Differential Equation, Algorithm

CHAPTER 1

Introduction and Background

Differential equations are used as a modeling or expressing many problems in various fields of human knowledge, such as physics, chemistry, mechanics, economics, etc.Many real problems turned into differential equation then, by solving it the answer is described . Usually many of these problems do not have analytical solutions or their solution may have certain implications. Many researchers have tried to approximate the solutions of these equations and proposed a lot of algorithms such as Runge-Kutta , Adam-Beshforth , Adam-Molton , Predictor-Corrector methods and other methods.

Neural networks are machine learning technique that have received a great deal of attention in recent years due to their ability to solve very difficult problems, particularly in the fields of image processing and object recognition, speech recognition, medical diagnosis, etc(Grossi and Buscema, 2007; Uhrig, 1995). More recently, applications have been found in engineering, especially where large data sets are involved.

Artificial Neural Network models can approximate highly non-linear functions efficiently even for large data sets. Due to the nice approximation properties of ANN , it has been used for the solution of various mathematical problems. ANN models are similar to biological nervous systems, consisting of an interconnected network of nodes. The network consists of an input layer,an output layer and one or more hidden layer. The input and output layers represent the inputs,t, and the outputs,O, of a system respectively. ANN models aim to represent the input-output correlation by obtaining the weights and biases of the nodes in the network (Dua, 2011).

Mathematical modeling has significant role in understanding many real problems, including predator-prey population dynamics , chemical reaction , the spread diseases and many more. Suitable mathematical models can be used to analyses the real problems. Many mathematical models have been constructed and analysed to describe the dynamics like plant disease transmission under some assumption.

Parameter estimation arises in fitting models containing several unknown parameters to experimental data through adjustment of these parameters. Model formulation is not a unique process; many different formulations may be used to fit the data and optimize model parameters. Of particular concern are formulations that are sufficiently accurate to represent physical or chemical phenomena and also are amenable to reasonably efficient computer solutions. Here we consider a parameter estimation problem formulated by the Putri et al. (2019) in order that current parameter estimation methods can be compared with real data.

In this thesis, we are applying n Artificial Neural Network (ANN) approach to estimate the unknown parameters of a system using the vectorized algorithm developed by Dufera (2021) for solving the system of ODEs. For the calculation of gradients of the cost functions, we utilized the auto-differentiation technique supported in the code by the autograd package and for the optimization of the learning rule, we implemented the Adam method , which successfully addresses the local minimum problem present in the standard approaches. Finally, we were tested the effectiveness of deep network structures for estimating parameters and solving of systems of odes.

1.1 History of Artificial Neural Network

The ANN was introduced as computing machine in the 1940s , and Rosenblatt later introduced the first model of supervised learning based on a single layer NN with a single neuron. Rosenblatt (Yadav et al., 2015) also presented an algorithm to adjust the free parameters in a learning procedure,and proved the convergence of the algorithm, which is now known as the perceptron convergence theorem. Multi layer NNs were introduced naturally after the single layer case. However, due to the difficulty of properly training a multi layer NN , little progress has been made.

The recent development and progresses in the area is attributed to the exponential improvement in the computing capacity of machines both in storing data and processing speed . An artificial neural network is an information processing system that has certain performance characteristics in common with biological neural networks (Yadav et al., 2015). The network imitates the work of biological human brain. The structure of the architecture constitutes layers: input, hidden and output. Each layer have neurons or units. The name deep neural network (DNN) is used when the structure has more than one hidden layers.

In the mid-70s, the development of hardware and the back propagation algorithm renewed the interest in the NNs. One important technique is the automatic differentiation (AD). For closed-form expressions, the AD is a numerical evaluation of symbolic differentiation. However, the AD is useful not only for closed-form expressions, but also for any computing program constructed from elementary mathematical operations . The AD calculates the derivative of a function defined by a computer program by performing a non-standard interpretation of the program, and the interpretation replaces the domain of the variables and redefines the semantic of the operators. The AD algorithm calculates derivatives using the chain rule from the differential calculus . Thus, machine-learning researchers do not need to manually calculate derivatives and code the complicated formulae that are prone to errors(Baydin et al., 2018; Basheer and Hajmeer, 2000) .

The complexity of real neurons is highly abstracted when modelling artificial neurons. These basically consist of inputs (like synapses), which are multiplied by weights (strength of the respective signals), and then computed by a mathematical function which determines the activation of the neuron(Chakraverty and Mall, 2017).The output of a neuron is computed by applying a transformation function over the weighted sum of the input signals This transformation function is called the activation function.The choice of activation function has a large impact on the capability and performance of the neural network, and different activation functions may be used in different parts of the model.

A back propagation neural network uses a supervised learning method and feed-forward architecture(Jain et al., 1996). It is one of the most frequently utilized neural network techniques for classification and prediction. In back propagation algorithm, the outputs of hidden layers are propagated to the output layer where the output is calculated. This output is compared with the desired output for the given input. Based on this difference, the error is propagated back from the output layer to hidden layer, and from hidden layer to input layer. As the flow moves back, it changes the weights between the neurons. This cycle of going forward from input and output, and from output to input is called an epoch.

1.2 ANN Training Process

There are many different algorithms that can be used when training artificial neural networks, each with their own separate advantages and disadvantages. The learning process within artificial neural networks is a result of altering the network's weights, with some kind of learning algorithm.The three major learning paradigms are:

- **Supervised Learning**-The learning algorithm would fall under this category if the desired output for the network is also provided with the input while training the network.
- **Unsupervised Learning**-In this paradigm the neural network is only given a set of inputs and it is the neural network's responsibility to find some kind of pattern within the inputs provided without any external aid.
- **Reinforcement Learning**-Reinforcement learning is similar to supervised learning in that some feedback is given, however instead of providing a target output a reward is given based on how well the system performed. The aim of reinforcement learning is to maximize the reward the system receives through trial-and-error.

A neural network is first given a set of known input data and asked to obtain a known output. This is called training the network. The network undergoes many such epochs till the error (difference between actual output and desired output) is within a certain tolerance). Now the network is said to be trained. This process of training sets the weights between all the neurons in all the layers. The weights so obtained from a trained network are used in calculating the response of the network to an unknown data.

1.3 Statement of the problem

There are many methods to solve system of differential equations. The objective of this research is the proposal of neural network structures capable of estimating parameters and solving of nonlinear systems of equations. But in recent,there is no more work by the method of ANN for estimating parameters and solving differential equations.

There are many methods to solve system of differential equations. Mathematical models are developed to help us better understand the world around us. In order to accurately reproduce complex phenomena ,such models are often nonlinear and do not permit closed form of analytical solutions. Here, we used the ANN algorithm for estimating the parameters and solving the dynamics system of maize foliar disease modeled by Putri et al. (2019).

The model was constructed under the following assumptions:

The model consists of three compartments, namely number of susceptible maize population (S), number of infected maize population (I) and number of pathogens population (P).

a. Susceptible maize populations are planted with constant rate.

b. Natural death rate of maize population is constant.

c. Death rate of infected maize plants due to pathogenic infection is constant.

d. The infection rate of susceptible maize by pathogen is constant.

e. The increasing rate of pathogen population is proportional to the number of infected maize population.

f. Death rate of pathogen population is constant.

$$\begin{aligned}
\frac{dS}{dt} &= K - \frac{\beta SP}{S+I} - \mu S \\
\frac{dI}{dt} &= \frac{\beta SP}{S+I} - (\mu+\sigma)I \\
\frac{dp}{dt} &= \nu I - \epsilon PS
\end{aligned} \tag{1.1}$$

with initial condition

$S(0) > 0, I(0) > 0$ and $P(0) > 0$. All the parameters are non-negative.

Table 1.1: Description of parameters of the proposed model

parameter	description
K	Planted rate of susceptible maize
β	Infection rate of susceptible maize by phathogen
μ	Natural death rate of maize population
σ	Death rate of infected maize plants due to pathogenic infection
ν	increasing rate of pathogen population
ϵ	Death rate of pathogen population

1.4 Objective of the problem

1.4.1 General objective

The general objective of this thesis is to propose the deep neural network for estimating parameters and solving systems of nonlinear differential equations

1.4.2 Specific objectives

The specific objectives of this thesis are to:

1. develop DNN algorithms for parameter estimation.
2. apply the algorithm for parameters estimation and solving the system of differential equations.
3. test the effectiveness of the method

1.5 Significance of the study

The main purposes of this thesis are to :

1. contribute the knowledge of ANN for solving and estimating parameters systems ODE for scientific community.
2. apply the proposed method to other non-linear system of DE equations to get an approximate parameters
3. study further in the area of ANN.

CHAPTER 2

Review of Related Literature

The ANN method has been established as a powerful technique to solve a variety of real world problems because of its excellent learning capacity. This method has been successfully applied in various fields such as function approximation, clustering, prediction, identification, pattern recognition, solving ordinary and partial differential equations, etc.Schalkoff (1997) Recently,more researchers are solving differential equations by using Artificial Neural work.

Lagaris et al. (1998) presented method for solving differential equations that relies upon the function approximation capabilities of the feed forward neural networks and provides accurate and differentiable solutions in a closed analytic form. The success of the method can be attributed to two factors.The first one is the employment of neural networks and the second is the form of the trial solution that satisfies by construction the boundary conditions.

According to these authors, the method is general and can be applied to both ODEs and PDEs by constructing the appropriate form of the trial solution. As indicated by their experiments the method exhibits excellent generalization performance. This is in contrast with the finite element method where the deviation at the testing points was significantly greater than the deviation at the training points.

In Aarts and Veer (2001) paper they demonstrated a neural network method to solve nonlinear differential equations and its boundary conditions. The idea of their method is to incorporate knowledge about the differential equation and its boundary conditions into neural networks and the training sets. Hereby they obtained specifically structured neural networks. To solve the nonlinear differential equation and its boundary conditions they have to train all obtained neural networks simultaneously.

The parameter estimation procedure for ODEs, multiple shooting, is reviewed and described in detail by Peifer and Timmer (2007) . In contrast to other attempts of estimating parameters in differential equations, the procedure does not suffer heavily from the attraction to local minima and the speed of convergence is considerably higher than global optimisation methods can achieve. Identifiability of the parameters can be regarded as central assumption for a successful operation of most of the numerical components. A regularisation procedure to weaken this assumption is included to the discussion of multiple shooting.The regularisation can further help to remove all unidentifiable parameters.

Gábor and Banga (2015) proposed a new parameter estimation strategy for nonlinear dynamical models of biological systems. The difficulties of parameter estimation problems in systems biology do not only depend on the number of parameters, but also on the structure (flexibility and non linearity) of the dynamic model, and the amount of information provided by the (usually scarce and noisy) available data. They tested this strategy with a set of case studies of increasing complexity. The results clearly indicate that an efficient global optimization approach should always be used, even for small models, to avoid convergence to local minima.

A method based on improved boundary chicken swarm optimization (IBCSO) algorithm is proposed by Chen et al. (2016) to solve the problem of parameter estimation for nonlinear systems.They have analyzed the influence of time series on the estimation accuracy;and achieved the following conclusions: shorter length of time series will benefit the estimation accuracy because that longer time series will make the objective function complicated. According to these authors, it is very important to select a suitable time series to reduce the estimation bias of aim nonlinear systems.

Grimm et al. (2020) considered a machine learning approach based on physics-informed neural networks(PINN) to estimate parameters of dynamical systems. They generalized it to the case of variable or time-dependent parameters and applied it to SIR and SEIR compartment models from mathematical epidemiology.They tested the new approach for several sets of numerical experiments with synthetic data that have been generated by numerically solving the systems of ODEs defining the different compartment models.

They conclude that the approach based on PINNs presented on their work, can identify parameters of SIR and SEIR models as long as the training data are in agreement with the model assumptions. For synthetic data, they consistently obtained more accurate parameter estimates for the SEIR model compared to the SIR model.

Dufera (2021) presented a vectorized algorithm for solving systems of ODE using DNN. He conducted different experiment using python code and simulated the result using graphs. According to the paper, more neuron size provides more accuracy, but more iteration for learning the parameters. He compared the ANN method with the well known fourth order Runge–Kutta method.He conclude that the ANN produced more accurate result for small number of the grid points. Moreover, for larger value of the domain, the ANN method provides better accuracy than RK4 method.

Shi and Xu (2021) discussed the possible forms of loss functions in an ANN method for solving differential equations, and investigated their efficiency through a number of numerical examples. The sensitivity on the accuracy of the ANN methods with respect to the size of training set, the number of nodes in hidden layer, as well as the penalty parameter was also studied. In order to understand the training behavior of the proposed neural networks, they selected several types of functions, and studied their approximation properties by the ANN methods.

They conclude that an ANN with exact boundary conditions is more accurate than an ANN with penalty and smooth activation function is preferable over non-smooth activation function in approximating smooth functions. Generally,they recommended to use a mixed of smooth and non-smooth activation functions .

The neural network training algorithm proposed scheme by Panghal and Kumar (2021) is applicable to different types of problems including ODEs, system of ODEs as well as higher-dimensional PDEs. They observed that the decrease in the error in the neural network solution with change in the number of points available and also the convergence rate is not uniform. Lack of uniform convergence rates can be attributed to the fact with same number of training points and same number of trainable parameters in the neural network .They observe that this neural network approach provides more accurate results with less training points as compared to more mesh points to achieve the desired accuracy.

CHAPTER 3

Research Methodology

This section consists of method and materials used to conduct the work.These are the study design and procedures of the study.

3.1 Study design

The design of this thesis was to solve and estimate the parameters of system of nonlinear differential equations using deep neural network. we also used graphical representation for discussion and python code as a tool.

3.2 Procedures of the study

To attain the objective of the study, we followed the following procedures:

- ★ setting the of systems of nonlinear differential equations.
- ★ setting the general form of DNN
- ★ performing different experiment using python code.
- ★ implementing the algorithm and
- ★ compare the result with the value of data and numerical solution obtained using other method.

3.3 Description of ANN algorithm

3.3.1 Components of the Basic Artificial neural network

Input Layer

This is the first layer in the neural network. It takes input signals(values) and passes them on to the next layer. It doesn't apply any operations on the input signals(values) and has no weights and biased values associated.

Hidden Layer

This is the medium layer in the neural network. Hidden layers have neurons(nodes) that apply different transformations to the input data.

Output Layer

This layer is the last in the network and receives input from the last hidden layer. With this layer, we can get the desired number of values and in the desired range.

Weights(Parameters)

Weight is a parameter of ANN. It represents the strength of the connection between nodes. Initially, it begins randomly. As training continues, it adjusted toward the desired values and the correct output. A weight brings down the importance of the input value. Weights near zero means changing this input will not change the output. positive weights mean increasing this input will increase the output. Negative weights mean increasing this input will decrease the output. A weight decides how much influence the input will have on the output.

Bias(Offset)

It is an extra input neuron/node. It is used for shifting the activation function towards left or right, it can be referred to as a y-intercept in the line equation. A low bias means the network is making more assumptions about the form of the output, whereas a high bias means the network is making fewer assumptions about the output(Uhrig, 1995).

Activation Functions

An activation function is a function that acts upon the net (input) to get the output of the network. The activation function acts as a squashing function, such that the output of the neural network lies between certain values (usually 0 and 1, or -1 and 1). The Nonlinear Activation Functions are mainly divided on the basis of their range or curves (there are many type). The Sigmoid Function curve looks like a S-shape. It exists between (0 to 1). Tanh is also like logistic sigmoid but better. The range of the tanh function is from (-1 to 1). tanh is also sigmoidal (s - shaped). The advantage is that the negative inputs will be mapped strongly negative and the zero inputs will be mapped near zero in the tanh graph. The function is differentiable. Both tanh and logistic sigmoid activation functions are used in feed-forward networks.

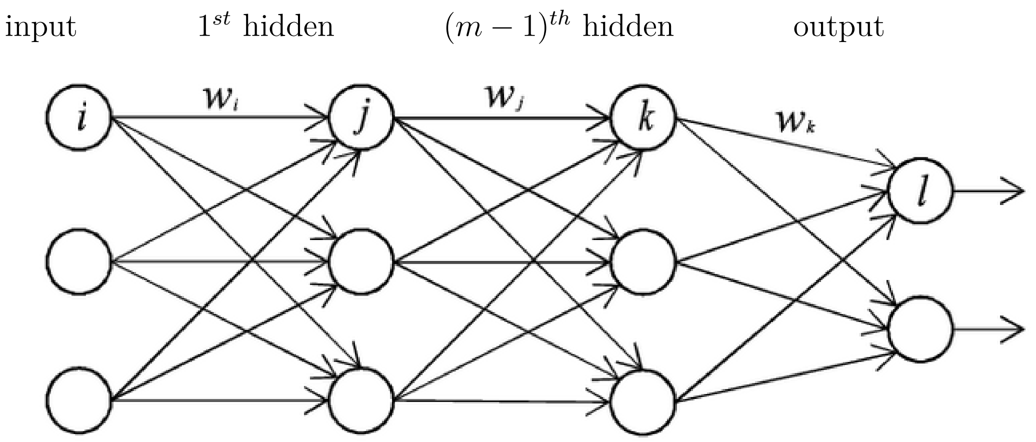
Summation Function

The work of the summation function is to bind the weights and inputs together and find their sum.
Neural networks can be considered as approximation schemes where the input data for a design of network consisting of only a set of unstructured discrete data points. Thus, an application of a neural network for solving differential equations can be regarded as a mesh-free numerical method. Recently, a lot of attention has been devoted to the study of ANN for solving differential equations. The approximate solution of differential equations by ANN is found to be advantageous, but it depends upon the ANN model that one considers.

3.3.2 Feed- neuralforward network

The network consists of an input layer,an output layer and one or more hidden layer. The input and output layers represent the inputs,t, and the outputs,O, of a system respectively. ANN models aim to represent the input-output correlation by obtaining the weights and biases of the nodes in the network . The network is fully connected input node to hidden layer,hidden layers with hidden layers and hidden layer completely connected to the output node of the network by means of connection weights Dua (2011).

The ANN model was constructed by using m-1 hidden layer and n nodes in the hidden layer. The ANN model obtained is given as follows:



. $O_j = \sigma(\sum_{i=1}^{n} w_i t_i) + b_i$ $\quad O_k = \sigma(\sum_{i=1}^{n} w_j t_j) + b_j$ $\quad O_l = (\sum_{i=1}^{n} w_k t_k) + b_k$

Figure 3.1: The scheme diagram of DNN

So, for an input t, the network will provide the output as:

$$O = \sum_{i=1}^{n} \sigma(t_i w_i) + b_i$$

Here, n denotes the number of neurons in the hidden layer and σ denotes the activation function of the hidden layer neurons.
Let us denote neurons on layer $m-1$ by k and neurons on layer m by j. Then, the following value goes into the j^{th} neuron of layer m,

$$O_j^m = \sum_{k=1}^{n} w_{jk}^m t_k^{m-1} + b_j^m \tag{3.1}$$

where n denotes the number of neurons in layer m.

The values in (3.1) will pass to the next hidden layer by an appropriate choice of an activation functions, denoted by σ_l The purpose of activation functions is to introduce non-linearity into the network and the choice has a substantial impact on network performance and training dynamics.This activation function must be Monotonic,differentiable and quickly converging with respect to the weights for optimization purposes.
Here, we choose the same activation function for all neurons in a layer. Thus, the values for the next layer is expressed by,

$$a^m = \sigma(O^m) = \sigma(w^l a^{m-1}) + b^m \tag{3.2}$$

For the t grid points, $i = 1, 2, ..., t$, we follow the following steps.

- First layer,

$$a^{1i} = \sigma(O^{1i}) \text{ , where } O^{1i} = w^1 a^1 + b^1$$

- Second layer,

$$a^{2i} = \sigma(O^{2i}) \text{ , where } O^{2i} = w^2 a^1 + b^2$$

- Third layer

$$a^{3i} = \sigma(O^{3i}) \text{ , where } O^{3i} = w^3 a^2 + b^3$$

- This process will continue until the output layer l,

$$a^{mi} = \sigma(O^{mi}) \text{ , where } O^{mi} = w^m a^{m-1} + b^m$$

3.3.3 Methods of Parameter Estimation

Here, we used by describing the ANN algorithm to estimate the parameters of the system of differential equations . This algorithm follows supervised learning that is different form the algorithm that described for solving ODE by Dufera (2021) .

Let us consider a system of ordinary differential equations involving unknown parameters in general form as follows:

$$\frac{dY_p(t)}{dt} = F_p(Y_p(t)), \quad t \in [t_0, T] \tag{3.3}$$

$$Y(t_0) = Y_0$$

with

$$Y_p(t) = \begin{pmatrix} y_p^1(t) \\ \vdots \\ y_p^n(t) \end{pmatrix} \quad and \quad F_p(Y_p) = \begin{pmatrix} f_p^1(Y_p) \\ \vdots \\ f_p^n(Y_p) \end{pmatrix}$$

where $y_p^i \in C_1(R)$ and $f_p^i \in C(R^n), i = 1, \ldots, n$ In particular, the function F_p depends on certain parameters $p \in R^k$, and therefore, the solution Y_p depends on p as well. Moreover, t_0 is the initial time and T the final time.
Let Y_j be given data at times $t_1, \ldots, t_M$.
. The solution of equation (3.3) will be denoted by $y^*(t)$.

Now,we compute the network output, denoted by $A_j^L(t_i, W, b)$. Then, the trial solution satisfying the initial conditions following the reference of Lagaris et al. (1998) is given by:

$$\hat{y}_j(t^{(i)}, W, b) = y_{0j} + (t_{(i)} - t_0)A_j^L(t, W, b), j = 1, ..., m. \tag{3.4}$$

The matrix form (3.4) is:

$$\hat{y}(T, W, b) = y_0 + (T - t_0)N(t, W, b) \tag{3.5}$$

Then, we train the network to find the numerical values w and b by minimizing the sum of the squares of the difference between $\hat{y^i}$ and y^i (cost function).
The cost function is

$$E(T, W, b) = \frac{1}{2}\sum_{i=1}^{m} ||\hat{y}(t^{(i)}, w, b) - y^{(i)}||^2 \tag{3.6}$$

converges to zero to obtain the optimal learning parameters W^* and b^*.

3.3.4 Solving system of ODE

We consider now the following system of first-order ODEs

$$\frac{dy_r}{dx} = f_r(x, y_1, y_2, ..., y_m), \tag{3.7}$$

where $r = 1, 2, 3, ..., m$ with $x \in [a, b]$ and subject to $y_r(a) = A_r$ with $r = 1, 2, 3, ..., m$. The initial value problem (3.7) can be written in the following ways;

$$\frac{dY}{dX} = F(x, y), Y(0) = Y_o$$

where $Y = |y_1, y_2, y_{3,..y_m}|^T$ is the unknown having dimension of $m \times 1$, and

$$F(x, y) = \begin{bmatrix} f_1(x, y_1, y_2, ..., y_m) \\ f_2(x, y_1, y_2, ..., y_m) \\ f_3(x, y_1, y_2, ..., y_m) \\ \vdots \\ f_r(x, y_1, y_2., y_m) \end{bmatrix}$$

The corresponding ANN trial solution has the following form:

$$y_{t_r}(x) = A_r + (x - a)N(x,p_r), \vee r = 1, 2, 3, ..., m \tag{3.8}$$

for all $r = 1, 2, 3, ...m$ where $N_r(x, p_r)$ is the neural output of the feed forward network with input and weights . For each $r, N_r(x, p_r)$ is the output of the multilayer ANN with input x and parameter p_r. The trial solution $y_{t_r}(x)$ satisfies the initial condition.

From Equation 3.8, we have

$$\frac{dy_{t_r}(x,p_r)}{dx} = N(x,p_r) + (x-a)\frac{dN_r}{dx}, \tag{3.9}$$

for all $r = 1, 2, 3, ..., m$.

Then, the corresponding error function with adjustable network parameters may be written as

$$E(x,p) = \sum_{i=1}^{h}\sum_{r=1}^{l}\frac{1}{2}[\frac{dy_{t_r}(x,p_r)}{dx} - f(x_i, y_{t_r}(x_i,p_r))]^2 \tag{3.10}$$

, where $x_i \in [a, b]$.
We train the network in such a way that the total cost function of equation (3.10) converges to zero.We use unsupervised learning for solving the system of differential equation.

3.3.5 Minimization of the Loss function

One of the widely used minimization algorithm in machine learning is the gradient decent method. We randomly initialize the parameters and update according to the following rule; for $j = 1, 2,$

$$p_j^{r+1} = p_j^r - \eta \bigtriangledown (Ep_j^r) \tag{3.11}$$

where η is the learning rate and r corresponds to iteration. One cycle through the complete training set forms one epoch.The above are repeated till cost function converges to zero. Note that, in addition to the simple gradient decent method, currently there are more advanced optimization tools and still is an active research topics.

Adam, Adaptive Moment, is an adaptive learning method which combines AdaGrad, Adaptive Gradient and Root Mean Square Propagation methods(Dufera, 2021; Kingma and Ba, 2014).
In our case the updating rule have the form;

$$M_w^k = \beta_1 M_w^{k-1} + (1-\beta_1)\nabla E_1(w^k)$$
$$M_b^k = \beta_1 M_w^{k-1} + (1-\beta_1)\nabla E_1(b^k)$$

$$M_w^k = \frac{M_w^k}{1-\beta_1^k}, V_w^k = \frac{V_w^k}{1-\beta_2^k}$$

$$M_b^k = \frac{M_b^k}{1-\beta_1^k}, M_b^k = \frac{V_b^k}{1-\beta_2^k}$$

$$\begin{aligned} W^{k+1} &= W^k - \frac{\eta}{\sqrt{V_w^k} + \epsilon} M_w^k \\ b^{k+1} &= b^k - \frac{\eta}{\sqrt{V_b^k} + \epsilon} M_b^k \end{aligned} \tag{3.12}$$

where $\beta_1, \beta_2 \in [0, 1)$, are decay rates for the moment estimates, η is the learning rate, and V_w, V_b and M_w, M_b are the first and the second moment vectors respectively that are initialized to be zero. The values $\eta = 0.001$.

3.3.6 Algorithm of ANN

Here, we describe algorithm for artificial neural network method for estimating parameters of system of differential equations (1.1).

Step 1:- Define input data
Take discrete points $t^{(i)} \in [a, b]$ where $i = 1, 2, \ldots, m$ and form an input vector $T = [t^1, t^2, t^3, \ldots, t^m]$

Step 2:- Define the neural network structure:
Determine the numbers of layers M, input layer with one unit, hidden layers $M - 2$ having h^m units,for $1 \leq m \leq M - 1$ and the output layers having n units which is equal to the number of unknown parameters in the system..

Step 3:- Initialize the network parameters(Choose the weights and biases)
-Randomly generate the weight vector W , the size is 1 $\times$ n and the entries are small.
-Set the bias to be zero. That is, b = 0.

Step 4:- Forward propagation:It makes prediction depending up on weights present in its layers.

- For input layer start by assigning, $A^0 = X$.
- For the hidden layers, $1 \leq m \leq M - 1$
  $Z^m = WX + b$
  $O^m(T, W, b) = \sigma(Z^m)$
  where,σ is the activation function corresponding to the hidden layer.

- For the output layer,
  $Z^M = WX + b$
  $O^M(T, W, b) = \sigma(Z^M)$

Step 5:- we assign the trial solution

Step 6:- Compute the cost function and its gradient ,using equation (3.10):
Calculate gradients with respect to t and with respect to the learning parameters. Here we implement the automatic differentiation(Baydin et al., 2018).

Step 7:- Update the weights and biases
We use the method of gradient decent which is widely used as best optimization method.We randomly initialize the parameters and update according to equation (3.11).

3.4 Implementation of the Method

In this section we use the DNN algorithm for estimating the parameters and solving the dynamics system of maize foliar disease modeled by Putri et al. (2019).

$$\begin{aligned}\frac{dS}{dt} &= K - \frac{\beta SP}{S+I} - \mu S \\ \frac{dI}{dt} &= \frac{\beta SP}{S+I} - (\mu+\sigma)I \\ \frac{dp}{dt} &= \nu I - \epsilon PS\end{aligned} \tag{3.13}$$

with initial condition
$S(0) > 0, I(0) > 0$ and $P(0) > 0$.
It consists of three unknown functions, five parameters with three initial conditions. Initial conditions and parameters value taken from Putri et al. (2019) are:
$(S(0), I(0), P(0)) = (6, 3, 2)$ and
$(\mu, k, \beta, \sigma, \nu, \epsilon)$= ($0.0103, 10, 0.1, 0.206, 0.0143, 0.1236$) .

To estimate the parameters , we considered a neural architecture with one hidden layer of 80 neurons.Furthermore, sigmoid activation function are applied to each unit and trained with 80000 epochs.Finally,we used the true values of the parameters for testing the validity of the proposed model.

3.4.1 Discussion

The results of the unknown parameters applying the algorithm are shown in Table 3.1.This table shows the accuracy of the method as compared with true values which are taken from literature. The obtained parameters by the method has no more deviation from the true value. That is, the absolute error is bounded by 0.00223 or the average error of the model is approximately 0.00057. Increasing number of neurons and epochs on this experiment shows the same result,but it took more running time.The absolute error of the proposed method is affected by the number of neuron and iterations.That is,it increase in some iteration and decrease in some iterations.

The solutions using ANN and the corresponding numerical solutions are indicated in figure 3.2 .To solve the dynamics system of maize foliar disease of the modeled equation using ANN , we considered 13 discrete points and compared with fourth order Runge-kutta method.The solution and absolute ANN error are shown in table 3.2 and table 3.3 ,respectively.The absolute error of the solutions is less than 0.00048.

Table 3.1: Comparison of value of data and estimated parameters for the modeled system of equation.

parameter	value of data	Value of ANN	Error
K	10	9.99959	0.00041
β	0.1	0.10025	0.00025
μ	0.0103	0.01022	0.00008
σ	0.206	0.20563	0.00037
ν	0.0143	0.012070	0.00223
ϵ	0.1236	0.12237	0.00001

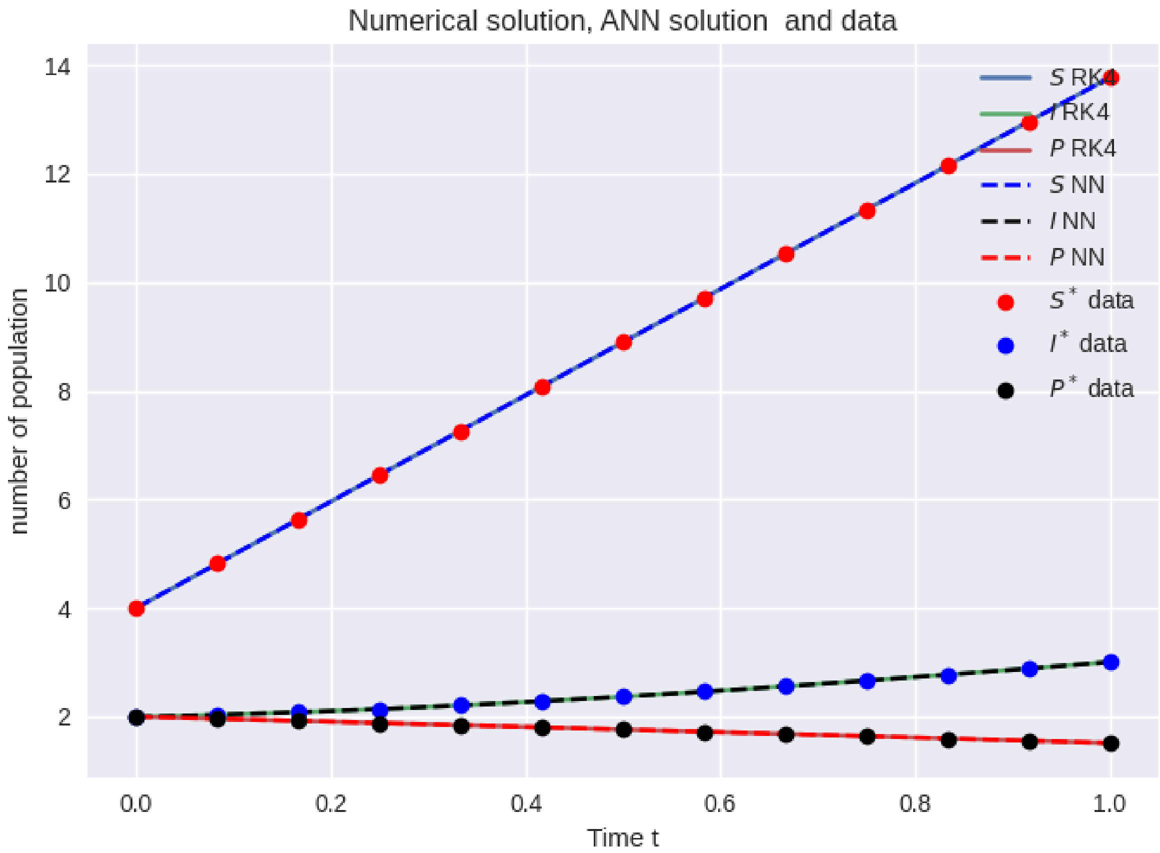


Figure 3.2: Numerical solution,ANNand data

Table 3.2: Discrete points of ANN and numerical solutions of the equation.

time	S ANN	S RK4	I ANN	I RK4	P ANN	P RK4
0.000000	2.000000	2.000000	4.000000	4.000000	3.000000	3.000000
0.083333	2.822061	2.821757	3.987181	3.987122	2.883807	2.884278
0.166667	3.642090	3.641866	3.991499	3.991340	2.772407	2.772802
0.250000	4.460560	4.460514	4.011216	4.011068	2.665171	2.665281
0.333333	5.277805	5.277896	4.044706	4.044678	2.561590	2.561423
0.416667	6.094064	6.094204	4.090438	4.090568	2.461257	2.460938
0.500000	6.909514	6.909620	4.146966	4.147202	2.363852	2.363548
0.583333	7.724286	7.724303	4.212920	4.213146	2.269132	2.268987
0.666667	8.538478	8.538393	4.287002	4.287089	2.176913	2.177010
0.750000	9.352162	9.352002	4.367980	4.367853	2.087066	2.087389
0.833333	10.165395	10.165220	4.454686	4.454396	1.999506	1.999917
0.916667	10.978219	10.978112	4.546017	4.545807	1.914184	1.914409
1.000000	11.790664	11.790721	4.640932	4.641297	1.831078	1.830694

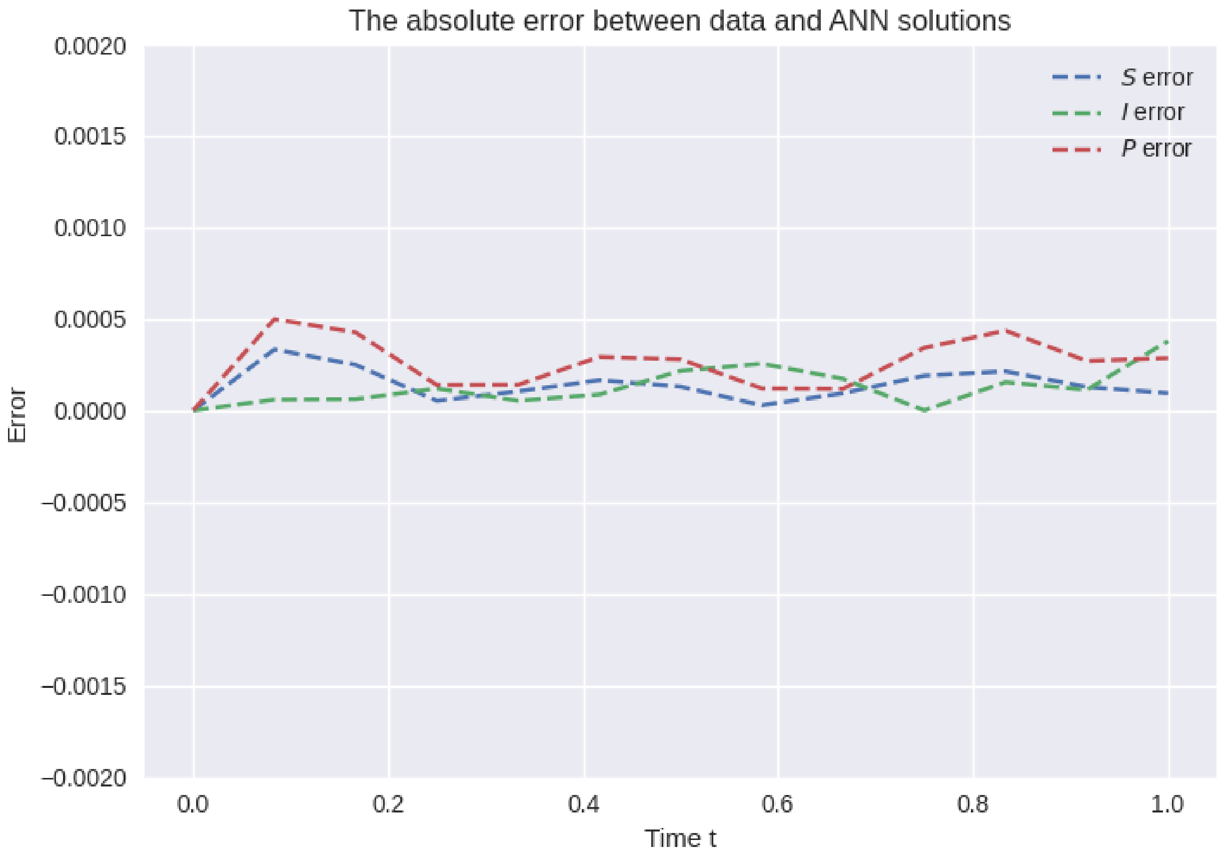


Figure 3.3: The absolute error between data Ann solution

Table 3.3: ANN absolute error

time	error S	error I	error P
0.000000	0.000000	0.000000	0.000000
0.083333	0.000304	0.000059	0.000471
0.166667	0.000224	0.000159	0.000395
0.250000	0.000046	0.000148	0.000110
0.333333	0.000091	0.000028	0.000167
0.416667	0.000140	0.000130	0.000319
0.500000	0.000106	0.000236	0.000304
0.583333	0.000017	0.000226	0.000145
0.666667	0.000085	0.000087	0.000097
0.750000	0.000160	0.000127	0.000323
0.833333	0.000175	0.000290	0.000411
0.916667	0.000107	0.000210	0.000225
1.000000	0.000057	0.000365	0.000384

CHAPTER 4

Conclusions and Recommendation

In this thesis, we developed a vectorized artificial neural network algorithm for estimating the unknown parameters and solving the dynamics system of maize foliar disease.

Finally, the result of the model shows that, for deep neural network, reasonable accuracy of parameter estimation could be achieved by wisely determining the number of epochs in order to minimize the running time. The absolute error obtained when compared with true value of the parameters is bounded approximately 0.00379.

We also compared the solutions of ANN method with fourth order Runge–Kutta method. The result showed that, the absolute error of ANN solution is bounded by 0.000384. This proposed methods efficient for estimate the parameters of dynamic system and solving system of equation. But, the error function increase in some iterations and decreased in some iteration however those iterations not known.

For the future study ,further investigation is required to strength the effectiveness of the model by comparing the proposed method with the traditional methods in terms of accuracy, computational complexity, provability and robustness.In addition to these, the effective number of iterations,neurons and hidden layers needs more investigation.

References

Aarts, L. P. and Veer, P. V. d. (2001). Solving nonlinear differential equations by a neural network method. In *International Conference on Computational Science*, pages 181–189. Springer.

Basheer, I. A. and Hajmeer, M. (2000). Artificial neural networks: fundamentals, computing, design, and application. *Journal of microbiological methods*, 43(1):3–31.

Baydin, A. G., Pearlmutter, B. A., Radul, A. A., and Siskind, J. M. (2018). Automatic differentiation in machine learning: a survey. *Journal of Marchine Learning Research*, 18:1–43.

Chakraverty, S. and Mall, S. (2017). *Artificial neural networks for engineers and scientists: solving ordinary differential equations*. CRC Press.

Chen, S., Yang, R., Yang, R., Yang, L., Yang, X., Xu, C., Xu, B., Zhang, H., Lu, Y., and Liu, W. (2016). A parameter estimation method for nonlinear systems based on improved boundary chicken swarm optimization. *Discrete Dynamics in Nature and Society*, 2016.

Dua, V. (2011). An artificial neural network approximation based decomposition approach for parameter estimation of system of ordinary differential equations. *Computers & chemical engineering*, 35(3):545–553.

Dufera, T. T. (2021). Deep neural network for system of ordinary differential equations: vectorized algorithm and simulation. *Machine Learning with Applications*, 5:100058.

Gábor, A. and Banga, J. R. (2015). Robust and efficient parameter estimation in dynamic models of biological systems. *BMC systems biology*, 9(1):1–25.

Grimm, V., Heinlein, A., Klawonn, A., Lanser, M., and Weber, J. (2020). Estimating the time-dependent contact rate of sir and seir models in mathematical epidemiology using physics-informed neural networks.

Grossi, E. and Buscema, M. (2007). Introduction to artificial neural networks. *European journal of gastroenterology & hepatology*, 19(12):1046–1054.

Jain, A. K., Mao, J., and Mohiuddin, K. M. (1996). Artificial neural networks: A tutorial. *Computer*, 29(3):31–44.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Lagaris, I. E., Likas, A., and Fotiadis, D. I. (1998). Artificial neural networks for solving ordinary and partial differential equations. *IEEE transactions on neural networks*, 9(5):987–1000.

Panghal, S. and Kumar, M. (2021). Optimization free neural network approach for solving ordinary and partial differential equations. *Engineering with Computers*, 37(4):2989–3002.

Peifer, M. and Timmer, J. (2007). Parameter estimation in ordinary differential equations for biochemical processes using the method of multiple shooting. *IET Systems Biology*, 1(2):78–88.

Putri, K. M. et al. (2019). A maize foliar disease mathematical model with standard incidence rate. In *IOP Conference Series: Materials Science and Engineering*, volume 546, page 052085. IOP Publishing.

Schalkoff, R. J. (1997). *Artificial neural networks*. McGraw-Hill Higher Education.

Shi, E. and Xu, C. (2021). A comparative investigation of neural networks in solving differential equations. *Journal of Algorithms & Computational Technology*, 15:1748302621998605.

Uhrig, R. E. (1995). Introduction to artificial neural networks. In *Proceedings of IECON'95-21st Annual Conference on IEEE Industrial Electronics*, volume 1, pages 33–37. IEEE.

Yadav, N., Yadav, A., Kumar, M., et al. (2015). *An introduction to neural network methods for differential equations*, volume 1. Springer.