Artificial Neural Network For Solving System of Nonlinear Ordinary Differential Equations

Yodit Getachew

A Thesis Submitted to 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(ANN) For Solving System of Nonlinear Ordinary Differential Equations

Yodit Getachew

Advisor: Tamirat Temesgen (PhD)

A Thesis Submitted to 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 Solving System of Nonlinear Ordinary Differential Equations" 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 Solving System of Nonlinear Ordinary Differential Equations " prepared under our guidance by Yodit Getachew

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

We, the advisors of the thesis entitled " Artificial Neural Network For Solving System of Nonlinear Ordinary Differential Equations " and developed by Yodit Getachew, 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
Co-advisor	Signature	Date

Approval page

We, the undersigned, members of the Board of Examiners of the thesis by Yodit Getachew, have read and evaluated the thesis entitled "Artificial Neural Network For Solving System of Nonlinear Ordinary Differential Equations "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 Dr.Tamirat Temesgen for his help and guidance in writing this thesis.I would also like to extend my gratitude to Deme Dadi my fellow graduate student.

A spacial thanks also goes to my family for their continued support and parents for being an ear to listen and a motivator in time of despair.

Contents

List of Tables

List of Figures

Acronyms

ANN- Artifical Nueral Network

ODE- Ordinary Differential Equation

RK4- Runge-kutta order four

SEIR- Susceptible ,Exposed,Infectious,Recovered

Abstract

In this thesis we implemented Artificial neural network for solving system of nonlinear ordinary differential equations. We develop a vectorized algorithm for solving SEIR(S-Susceptible ,E-Exposed, I-Infectious and R-Recovered) model and apply python code.We develop more techniques to handle the challenges of experiments. For the learning of the neural network, we utilized the adaptive moment minimization method. Finally,we compare with Runge-Kutta order four method and We have shown that, the artificial neural network could gives better accuracy.

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

CHAPTER 1

Introduction and Background

Nonlinear ordinary differential equations describing various physical processes that are important for scientists to gain better understanding in mathematical physics. Some classical model equations are constructed and analyzed mathematically and numerically to simulate the essential mechanism of the nonlinear interaction. In most cases these equations are too complicated to be solved explicitly such that we need to approximate the solution by Artificial neural network(ANN).

The term "Artificial neural network" refers to a biologically inspired sub-field of artificial intelligence modeled after the brain. An Artificial neural network is usually a computational network based on biological neural networks that construct the structure of the human brain. Similar to a human brain has neurons interconnected to each other, artificial neural networks also have neurons that are linked to each other in various layers of the networks. These neurons are known as nodes.McFall (2010)

A system of differential equations is said to be nonlinear if it is not a linear system. Problems involving nonlinear differential equations are extremely diverse, and methods of solution or analysis are problem dependent.From examples of nonlinear differential equations:the SEIR model equations in epidemiology, the prey-predator in biology and others . But Our focus will be specifically on the equation SEIR model which describes the transmission dynamics diseases where the given population.Umar et al. (2020)

Epidemiology is broadly termed the science of public health involving quantitative investigation of disease outbreak with the aim of controlling epidemics. Epidemiology provides essential quantitative and analytical methods, principles of logical inquiry, and rules for evidence for disease management. It helps in measuring and projecting the health needs of community and populations and determining how to allocate and manage health care resources. It assesses intervention strategies and evaluates the impact of

required health services. Canzani and Lechner (2015) explored mathematical models in epidemiology, with emphasis on theories and methodologies. The study identified core building blocks of models and research patterns to model diseases dynamics in epidemic situations. Moreover, the mathematical modelling of epidemics dynamics was explored.

There are two possible approach to solve this problems one approach is to use systolic or wave front arrays another approach is to employ Artificial Neural Network (ANN) The motivation for studying various ANN models for nonlinear ODE problems is to twofold.One reason for investigating various ANN models is that basic and fundamental problems such as matrix inversion are encountered in engineering application where a very fast on-line solution is required .A second reason is that understanding properties feature of relatively simple ANN model can be aid to understanding and developing new architectures for more complex and general system of nonlinear ODE's problem.(Berg and Nyström, 2018)

1.1 Background of ANN

The study of the human brain is thousand of years old with the advent of modern electronics; it was only naturally to try to harness this thinking process. The first artificial neuron was the Threshold Logic Unit (TLU), or Linear Threshold Unit, first proposed by Warren McCulloch and Walter Pitts in 1943McCulloch and Pitts (1943). The model was specifically targeted as a computational model of the "nerve net" in the brain. As a transfer function, it employed a threshold, equivalent to using the Heaviside step function. Initially, only a simple model was considered, with binary inputs and outputs, some restrictions on the possible weights, and a more flexible threshold value. Since the beginning it was already noticed that any boolean function could be implemented by networks of such devices, what is easily seen from the fact that one can implement the AND and OR functions, and use them in the disjunctive or the conjunctive normal form. Researchers also soon realized that cyclic networks, with feedbacks through neurons, could define dynamical systems with memory, but most of the research concentrated (and still does) on strictly feed-forward networks because of the smaller difficulty they present .

One important and pioneering artificial neural network that used the linear threshold function was the perceptron, developed by Frank Rosenblatt. This model already considered more flexible weight values in the neurons, and was used in machines with adaptive capabilities. The representation of the threshold values as a bias term was

introduced by Bernard Widrow in 1960 .

In the late 1980s, when research on neural networks regained strength, neurons with more continuous shapes started to be considered. The possibility of differentiating the activation function allows the direct use of the gradient descent and other optimization algorithms for the adjustment of the weights. Neural networks also started to be used as a general function approximation model. The best known training algorithm called back-propagation has been rediscovered several times but its first development goes back to the work of Paul Werbos.

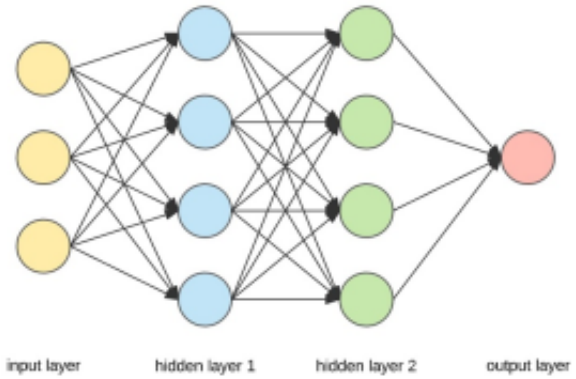

Figure 1.1: Model of nueral network

An artificial neural network in an inter connected group of nodes, inspired by a simplification of neurons in brain. Here each circular nodes represents an artificial neuron and an arrow represents a connection from the output of to Artificial Network input of another Huang (2007).

1.2 The Back propagation Algorithm

Back propagation (BP) is a widely used algorithm for training feedforward neural networks. Generalizations of back propagation exist for other artificial neural networks (ANNs), and for functions generally. These classes of algorithms are all referred to generically as "backpropagation".

Cilimkovic (2015) In fitting a neural network, backpropagation computes the gradient of the loss function with respect to the weights of the network for a single input output example, and does so efficiently, unlike a naive direct computation of the gradient with respect to each weight individually. This efficiency makes it feasible to use gradient methods for training multi layer networks, updating weights to minimize loss; gradient descent, or variants such as stochastic gradient descent, are commonly used.

The back propagation algorithm works by computing the gradient of the loss function with respect to each weight by the chain rule, computing the gradient one layer at a time, iterating backward from the last layer to avoid redundant calculations of intermediate terms in the chain rule; this is an example of dynamic programming. (Mall and Chakraverty, 2012)

1.3 ANN Training process

Once a network has been structured for a particular application, that network is ready to be trained. To start this process the initial weights are chosen randomly. Then the training or learning begins Gershenson (2003).

There are two approaches to training supervised and unsupervised. Supervised training involves a mechanism of providing the network with the desired output either by manually "grading" the network's performance or by providing the desired outputs with the inputs. Unsupervised training is where the network has to make sense of the inputs without outside help.

The vast bulk of networks utilize supervised training. Unsupervised training is used to perform some initial characterization on inputs. However, in the full blown sense of being truly self learning, it is still just a shining promise that is not fully understood, does not completely work, and thus is relegated to the lab.

1.3.1 Supervised Training

In supervised training, both the inputs and the outputs are provided. The network then processes the inputs and compares its resulting outputs against the desired outputs. Errors are then propagated back through the system, causing the system to adjust the weights which control the network. This process occurs over and over as the weights are continually tweaked. The set of data which enables the training is called the "training set." During the training of a network the same set of data is processed many times as the connection weights are ever refined.(Maind et al., 2014)

1.3.2 Unsupervised Training

The other type of training is called unsupervised training. In unsupervised training, the network is provided with inputs but not with desired outputs. The system itself must then decide what features it will use to group the input data. This is often referred to as self-organization or adaption.

1.4 Mathematical model of a neuron

A formal neuron, which is obtained by re-formulating a simplified function of biological neuron into a mathematical formalism, will be the basis of the mathematical model of neural network.

A neural network consists of formal neurons which are connected in such a way that each neuron output further serves as the input of generally more neurons similarly as the axon terminals of a biological neuron are connected via synaptic bindings with dendrites of other neurons. The number of neurons and the way that they are interconnected determine the architecture (topology) of neural network. Regarding their purpose, the input, working (hidden layer, mediate) and output neurons may be distinguished in the network. The input and output neurons represent the receptors and effectors, respectively, and the connected working neurons create the corresponding channels between them to propagate the respective signals. These channels are called paths in the mathematical model. The signal propagation and information processing along a network path is realized by changing the states of neurons on this path. The states of all neurons in the network form the state of the neural network and the synaptic weights associated with all connections represent the configuration of the neural network. (Jose l.paz-paredes (2008))

The formal neuron has n, generally real, inputs $x_1, x_2, ., ., ., .x_n$ that model the signals coming from dendrites. The inputs are labeled with the corresponding, generally real, synaptic weights $w_1, w_2, ., ., ., ., w_n$ that measure their permeabilities. According to the neurophysiological motivation, some of these synaptic weights may be negative to express their inhibitory character. Then, the weighted sum of input values represents the excitation level of the neuron: The ANN architecture comprises of:

- input layer: Receives the input values
- hidden layer(s): A set of neurons between input and output layers. There can be single or multiple layers
- output layer: multiple outputs can also be present

The value of excitation level x, after reaching the threshold h, induces the output y (state) of the neuron which models the electric impulse generated by axon. The nonlinear grow of output value y = s(x) after the threshold excitation level h is achieved, is determined by the activation (transfer) functions.

The most common hard limiter types of activation function are :

- Sigmoid or logistic function,$\sigma(z) = \dfrac{1}{1+e^{-x}}$
- Hyperbolic Tangent function, $\tanh(x) = \dfrac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$
- Linear functions, $\sigma(x) = x$
- Positive linear/ ReLu function. ReLu stands for Rectified Linear Units, $\sigma(x) = max(0, x)$

A neural network based model for the solution of differential equations provides the following advantages over the standard numerical methods:(Lagaris et al., 1998)

- The neural network based solution of a differential equation is differentiable and is in closed analytic form that can be used in any subsequent calculation. On the other hand most other techniques offer a discrete solution or a solution of limited differentiability.
- The neural network based method to solve a differential equation provides solution with very good generalization properties.
- Computational complexity does not increase quickly in the neural net work method when the number of sampling points is increased while in the other standard numerical methods computational complexity increases rapidly as we increase the number of sampling points in the interval.
- The method is general and can be applied to the systems defined on ei ther orthogonal box boundaries or on irregular arbitrary shaped boundaries
- Model based on neural network offers an opportunity to tackle in real time difficult differential equation problems arising in many sciences and engineering applications.
- The method can be implemented on parallel architectures.

1.5 Statement Of The Problem

As nonlinear dynamical equations are difficult to solve, nonlinear systems are commonly approximated by linear equations (linearization). This works well up to some accuracy and some range for the input values, but some interesting phenomena such as solution, chaos, and singularities are hidden by linearization. It follows that some aspects of the dynamic behavior of a nonlinear system can appear to be counter intuitive, unpredictable or even chaotic. In general one restricts the study to systems such that the number of unknown functions equals the number of equations.

Here we consider the nonlinear system of differential equations.Hassan et al. (2015)

$$\begin{aligned} \frac{dS}{dt} &= \pi - \beta SI - \mu S, \\ \frac{dE}{dt} &= \beta SI - (\mu + \delta)E, \\ \frac{dI}{dt} &= \delta E - (\mu + \eta)I, \\ \frac{dR}{dt} &= \eta I - \mu R, \end{aligned} \tag{1.1}$$

With initial condition
$S(0) \geq 0$,$E(0) \geq 0$,$I(0) \geq 0$ and $R(0) \geq 0$.

S-Susceptible ,E-Exposed, I-Infectious and R-Recovered are governed by system of nonlinear ordinary differential equation with constant values of parameters,π, β, δ, η and μ . The purpose of this work is to investigate the solutions of the the above SEIR(Eq1.1) nonlinear model by using ANNs.

1.6 Objectives of the study

The mathematical theory of nonlinear partial differential equations describing various physical processes plays an important role in applied mathematics and mathematical physics . But only a few types of differential equations have been solved successfully in explicit representation formulas. Different qualitative methods are employed to solve differential equations. Different qualitative methods are employed to solve differential equations. ANN method is one of the most powerful tools of ways of solving differential equations, which allow us to obtain more accurate representation for solutions with respect to a large parameter.

1.6.1 General objective

The general objective of this study will be to solve a system of nonlinear ordinary differential equation by utilizing ANN .

1.6.2 Specific Objectives

The specific objective of this study are to:

1. determine the ANN Architecture and vectorized Algorithm of system for nonlinear ordinary differential Equations.
2. compute the solution of ANN with Runge-kutta methods
3. applies python code for ANN Architecture and Algorithm of system of nonlinear ordinary differential equation,specifically for SEIR model

1.7 Significance of the study

The main purposes of the study are:

- to contribute the knowledge of ANN to the community .
- for the growth of the research in the area

CHAPTER 2

Review of Related Literature

Lagaris et al. (1998) used ANN for solving ordinary and partial differential equations. For solving initial and boundary value problems, they used trail solution satisfying the given conditions. Then, network were trained to satisfy the differential equations. The results were compared with well established numerical method finite element. The authors obtained accurate and differentiable solution in a closed analytic form.

Malek and Beidokhti (2006) used ANN to solve coupled systems of partial differential equations, including accurate approximation of local Nusselt numbers in boundary layers and solving the Navier-Stokes equations for the entry length problem. With strengths including an explicit and continuous approximate solution, elimination of meshing concerns and simple implementation for nonlinear differential equations, this method is emerging as a viable alternative to traditional numerical techniques such as the finite element method.

Malek and Beidokhti (2006) applied optimization techniques and neural networks methods for the solution of high order ordinary differential equations. Here neural networks is considered as a part of large field called neural computing or soft computing. This means that we propose a new solution method for the approximated solution of high order ordinary differential equations using innovative mathematical tools and neural-like systems of computation. This hybrid method can result in improved numerical methods for solving initial/boundary value problems, without using preassigned discretisation points.

McFall (2010) applied artificial neural network method to solve coupled systems of partial differential equations, including accurate approximation of local Nusselt num-

bers in boundary layers and solving the Navier-Stokes equations for the entry length problem. With strengths including an explicit and continuous approximate solution, elimination of meshing concerns and simple implementation for nonlinear differential equations, this method is emerging as a viable alternative to traditional numerical techniques such as the finite element method.

Hassan et al. (2015) applied the numerical solution of the system of SEIR nonlinear ordinary differential equations, which are studied the effect of vaccine on the HIV (Human Immunology virus). They obtained the numerical solutions on stable manifolds by Runge-Kutta fourth order method.

Canzani and Lechner (2015) explored mathematical models in epidemiology, with emphasis on theories and methodologies. The study identified core building blocks of models and research patterns to model diseases dynamics in epidemic situations. Moreover, the mathematical modelling of epidemics dynamics was explored.

Chakraverty and Mall (2017) implemented ANN method to investigates the solution of Ordinary Differential Equations (ODEs) with initial conditions using Regression Based Algorithm (RBA) and compares the results with arbitrary- and regression-based initial weights for different numbers of nodes in hidden layer. Here, they have used feed forward neural network and error back propagation method for minimizing the error function and for the modification of the parameters (weights and biases).

Initial weights are taken as combination of random as well as by the proposed regression based model. they present the method for solving a variety of problems and the results are compared. Here, the number of nodes in hidden layer has been fixed according to the degree of polynomial in the regression fitting. For this, the input and output data are fitted first with various degree polynomials using regression analysis and the coefficients involved are taken as initial weights to start with the neural training. Fixing of the hidden nodes depends upon the degree of the polynomial. For the example problems, the analytical results have been compared with neural results with arbitrary and regression based weights with four, five, and six nodes in hidden layer and are found to be in good agreement.

Umar et al. (2020) implemented A stochastic numerical computing heuristic of SIR nonlinear model based on dengue fever optimized by a well-known global genetic algorithm (GA) and local refinements of sequential quadratic programming (SQP), i.e., ANN-GA-SQM. The optimization of an error based merit function is performed by

using the concepts of differential model along with the initial conditions to solve the SIR nonlinear model based dengue fever.

Dufera (2021) applied the deep neural network for solving system of ordinary differential equation(ODE).He has used the method algorithm in python and perform experimental simulations to look at the effects of different neural architecture on the performance of the model. Moreover, he observe the advantage of using the ANN over the traditional methods. Specifically, we consider the fourth order Runge-Kutta finite difference method.

Philemon et al. (2019)) applied standalone or hybrid ANNs in predicting epidemics, with suggested resolution for issues that affect the forecast process. The application of ANN in forecasting deadly infectious disease outbreaks was reviewed. Although most studies considered the multi-layer perceptron feed forward neural network (MLPFFNN), that is not to say that the epidemic forecast cannot be modeled according to other network architectures. In order to enhance the accuracy of ANN performance.

CHAPTER 3

Methodology and procedures

This section consists the method and materials used to undertake the study.when the equation structure is complicated, or the nonlinear term is strong, we need to develop more techniques to handle the challenges.

3.1 Feed-forward neural network

This class of networks consists of multiple layers of computational units, usually interconnected in a feed-forward way. Each neuron in one layer has directed connections to the neurons of the subsequent layer.The network is fully connected with input node fully connected to hidden layer and hidden layer completely connected to the output node of the network by means of connection weight (Berg and Nyström (2018)).
We are using w_{ij}, v_i and w_i to denote the connection weights from input node to hidden layer, connection weights from hidden layer to output node and bias weights for hidden nodes, respectively. The same convention will be used throughout the thesis. So, for an input x, the network will provide the output as:

$$N = \sum_{m=1}^{n} v_i * \sigma(x_i) w_i + b_i$$

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

$$N_j^l = \sum_{k=1}^{h} w_{jk}^{l-1} x_k^l + b_j^l \tag{3.1}$$

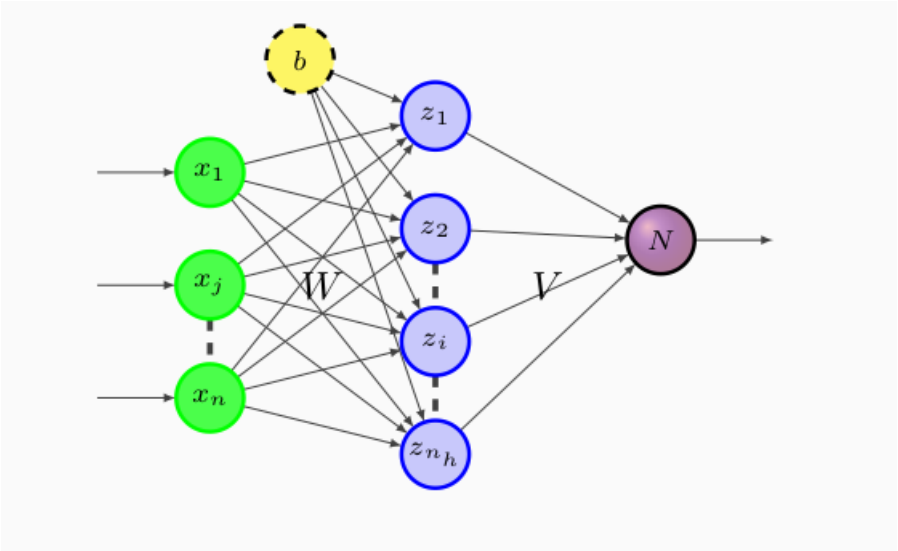

Figure 3.1: A forward neuron network

where h denotes the number of neurons in layer$l$. The matrix form

$$N = W^{l-1}X^l + B^l \tag{3.2}$$

where the matrix,W^{l-1} is weights w_{jk} and B^l is bias b^l_j

The values in (3.1) will pass to the next hidden layer by an appropriate choice of an activation functions, denoted by σ. In this study, we choose the same activation function for all neurons in a layer. Thus, the values for the next layer is expressed by,

$$x^l = \sigma(N^l) = \sigma(w^{l-1}x^l) + b^l \tag{3.3}$$

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

- First layer, $a^{1i} = \sigma(N^{1i})$, where $N^{1i} = W^{11} + b^1$
- Second layer, $a^{2i} = \sigma(N^{2i})$, where $N^{2i} = W^2a^1 + b^2$
- Third layer $a^{3i} = \sigma(N^{3i})$, where $N^{3i} = W^3a^2 + b^3$
  This process will continue until the output layer l.
  $a^{li} = \sigma(N^{li})$, where $N^{li} = W^la^{l-1} + b^l$

the above expressions can be written in matrix form as follows: on the Output layer l,

$$A^l = \sigma(N^l) \text{ and } N^l = W^la^{l-1} + b^l$$

3.2 ANN for solving system of nonlinear Ordinary Differential Equation

The general form of system of ODE'S is:

$$\frac{dy_r}{dx} = f_r(f(x, y_1, y_2, ..., y_n), \tag{3.4}$$

where $r = 1, 2, 3, ..., n$ with $x \in [a, b]$ and subject to $y_r(a) = A_r$ with $r = 1, 2, 3, ..., n$.

$$\begin{aligned} y_1'(x) &= b_1(x) + a_{1,1}y_1(x) + \cdots + a_{1,n}y_n(x) \\ &\vdots \\ y_n'(x) &= b_n(x) + a_{n,1}y_1(x) + \cdots + a_{n,n}y_n(x) \end{aligned} \tag{3.5}$$

The corresponding ANN trial solution has the following form:

$$y_{tr}(x) = A_r + (x - a)N(x, p_r) \tag{3.6}$$

where $r = \{1, 2, 3, \ldots, n\}$ with $x \in [a, b]$ and subject to $y_r(a) = A_r$ with $r = \{1, 2, 3, \ldots, n\}$. for all $r = 1, 2, 3, \ldots, n$ where $N_r(x, p_r)$ is the neural output of the feed for ward network with input and weights . For each $r, N_r r(x, p_r)$ is the output of the multi-layer ANN with input x and parameter p_r. The trial solution $y_{tr}(x)$ satisfies the initial condition. (Malek and Beidokhti, 2006)
From equation 3.6 we have,

$$\frac{dy_{tr(x,p_r)}}{dx} = N(x, p_r) + (x - a)\frac{dN_r}{dx}, \forall r = 1, 2, 3, ..., n \tag{3.7}$$

Then, the corresponding error function with adjustable network parameters will be written:

$$E(x, p) = \sum_{i=1}^{h}\sum_{r=1}^{l}\frac{1}{2}\left[\frac{dy_{tr(x,p_r)}}{dx} - f(x, dy_{tr}(x_i, p_r))\right]^2 \tag{3.8}$$

where $x_i \in [a, b]$
We train the network in such a way that the total cost function of equation converges to zero.We use unsupervised learning or training process as there are no targeted solutions.

3.3 The vectorized algorithm

In this section we express the algorithm of ANN method for solving system of ODE such as:

1. Input data

We take m discrete points from the domain $[a, b]$ and form a vector $x = [x_1, x_2, ..., x_m]$ of size $1 \times m$

2. Define the neural network structure.
   Here we determine the num ber of:

   - layers l,
   - input layer having one units,
   - hidden layers and its units
   - output layer having n units which is equal to the number of unknown in the system.

3. Initialize the parameters
   we need to initialize a corresponding sets of parameters.Initialize weight matrix of dimension $1 \times k$

4. Compute the cost function and its gradient
   We calculate gradients with respect of x and with respect to the learning parameters.

5. Update the weights and biases

6. Repeating the above process for multiple times. .

CHAPTER 4

Implementation and conclusions

4.1 Implementation

In this section we implemented the ANN algorithm for solving a non- linear systems of differential equations. we perform simulation for selecting appropriate number of layers and neurons in the layer.Then, we compare with the analytical solution and with a numerical solution obtained using the traditional methods. For this purpose, we consider the following SEIR Model found in Hassan et al. (2015)

$$\begin{aligned} \frac{dS}{dt} &= \pi - \beta SI - \mu S, \\ \frac{dE}{dt} &= \beta SI - (\mu + \delta)E, \\ \frac{dI}{dt} &= \delta E - (\mu + \eta)I, \\ \frac{dR}{dt} &= \eta I - \mu R, \end{aligned} \tag{4.1}$$

With initial condition

$S(0) = 1$,$E(0) = 0$,$I(0) = 1$ and $R(0) = 0$
parameters $\pi = 0, \beta = 0.5, \mu = 0.25, \delta = 0.75$ and $\eta = 0.75$.

For input x the network provided output of the model is:

$$\begin{aligned} S &= \sum_{i=1}^{n} v_{1i} * \sigma(x_i) w_i + b_i \\ E &= \sum_{i=1}^{n} v_{2i} * \sigma(x_i) w_i + b_i \\ I &= \sum_{i=1}^{n} v_{3i} * \sigma(x_i) w_i + b_i \\ R &= \sum_{i=1}^{n} v_{4i} * \sigma(x_i) w_i + b_i \end{aligned}$$

4.1.1 Experiment on the network

Now we use the ANN method for solving the system of differential equations. In line with the simulations, we selected a single hidden layer with different neurons (tuning in plus minus may not have significant effect). For this experiment,$m = 21$, uniform grid points were sampled from the given interval. The solutions using ANN and RK4 of the experiment shown in tabular and graphics form.

To implement the method for the equation 4.1 we managed an experiment on number of neurons in a layer, observed at the effect of number of neurons on the error function, took various sizes of neurons in the hidden layer,$h = 5, 11, 30$, plotted the cost function versus the number of iterations for the comparison of convergence and displayed the cost at the end of iterations corresponding to each neuron size and the time it take for the calculation.

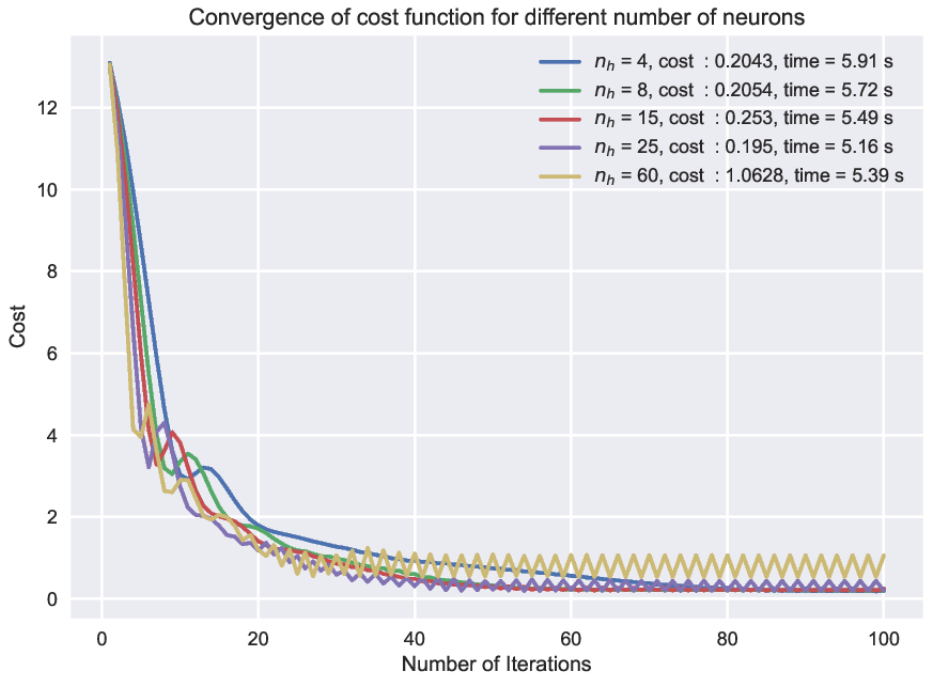

Figure 4.1: convergence of cos function

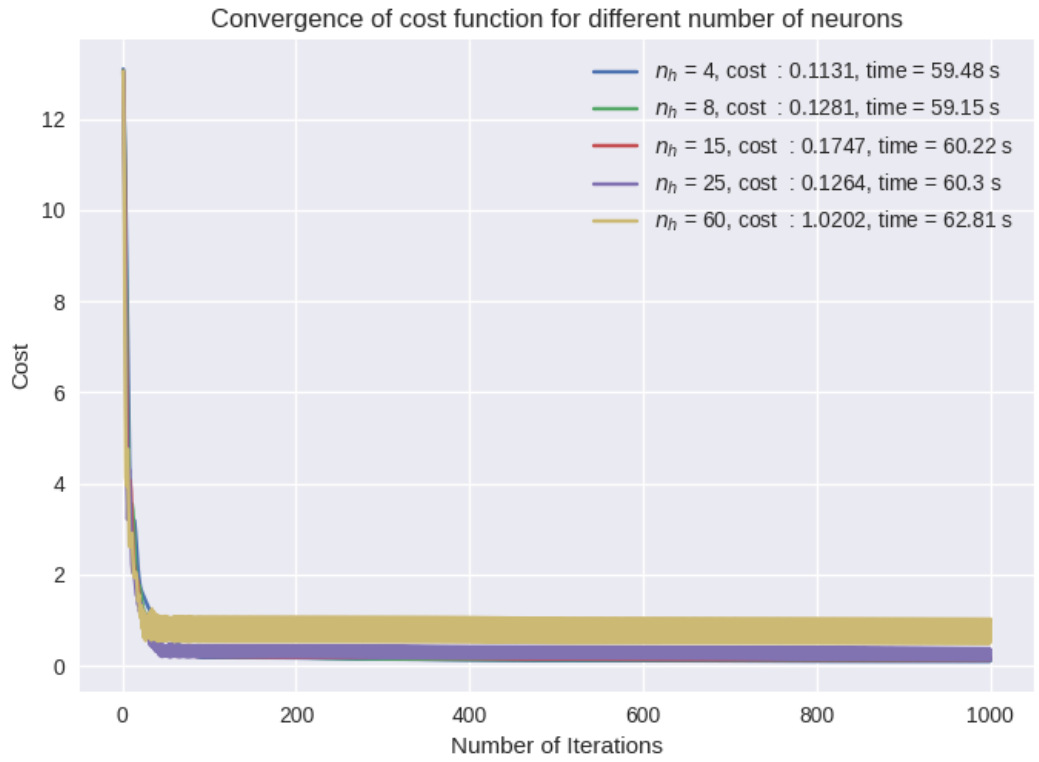

Figure 4.2: convergence of cos function

4.2 Discussion

From the simulation shown Fig.4.2, we observe that, one can get the required accuracy even for a single neuron in the hidden layer. However, it needs large number of iterations for smaller number of neurons leading to problem of computational time. Increasing the number of neurons has advantage on the performance of the model.However, an arbitrary increase is unnecessary. In this case $h = 200$ has similar accuracy with $h = 1000$ with less computational time.

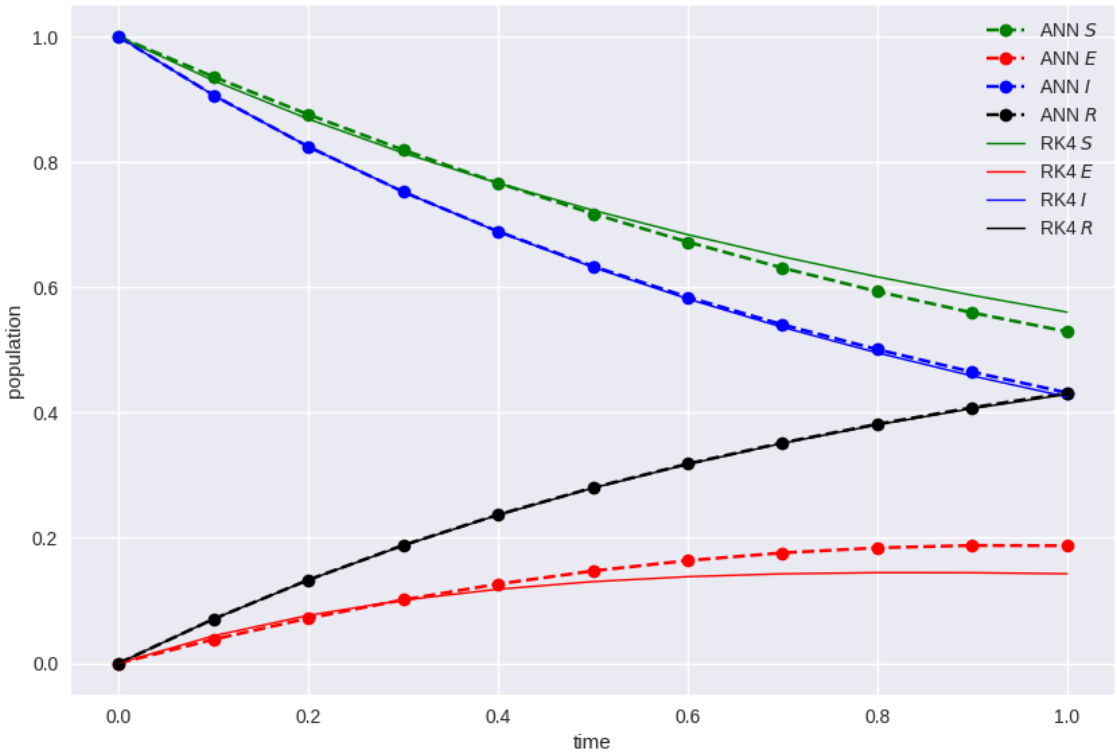

Figure 4.3: Solution of ANN and Runge-Kutta order

Table 4.1: Solutions of ANN and numerical solutions of RK4

time	ANN S	RK S	ANN E	RK E	ANN I	RK I	ANN R	RK R
0.0	1.00	1.00	0.00	0.00	1.00	1.00	0.00	0.00
0.1	0.94	0.93	0.04	0.04	0.91	0.91	0.07	0.07
0.2	0.88	0.87	0.07	0.08	0.82	0.82	0.13	0.13
0.3	0.82	0.81	0.10	0.10	0.75	0.75	0.19	0.19
0.4	0.77	0.77	0.13	0.12	0.69	0.69	0.23	0.24
0.5	0.71	0.72	0.15	0.13	0.63	0.63	0.28	0.28
0.6	0.67	0.68	0.16	0.13	0.58	0.58	0.31	0.31
0.7	0.63	0.64	0.18	0.14	0.54	0.54	0.35	0.35
0.8	0.59	0.62	0.18	0.14	0.50	0.49	0.38	0.38
0.9	0.56	0.59	0.19	0.14	0.464	0.45	0.40	0.40
1.0	0.52	0.56	0.18	0.14	0.43	0.42	0.43	0.42

CHAPTER 5

Conclusions and Recommendations

In this study ,we analyzed a vectorized algorithm for solving system of nonlinear ODEs by using ANN . The experiment shown that main advantage of using ANN method over RK4 provides better performance for small grid points . but,the revers is true in RK4. In addition to this we observe that the result of ANN is more accurate than RK4. Depending on the advantage of ANN over RK4 ANN method can be additional alternative method to solve nonlinear Ordinary Differential Equations.

For a future work, further analytical investigation is required to strength the foundation of ANN for solving system of ODE and PDE. These include looking at stability and convergence and ANN related to solving system of ODEs. In the same way looking at using other architecture

References

Berg, J. and Nyström, K. (2018). A unified deep artificial neural network approach to partial differential equations in complex geometries. *Neurocomputing*, 317:28–41.

Canzani, E. and Lechner, U. (2015). Insights from modeling epidemics of infectious diseases-a literature review. *ISCRAM*.

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

Cilimkovic, M. (2015). Neural networks and back propagation algorithm. *Institute of Technology Blanchardstown, Blanchardstown Road North Dublin*, 15(1).

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

Gershenson, C. (2003). Artificial neural networks for beginners. *arXiv preprint cs/0308031*.

Hassan, A., Ibrahim, S. H., and Ibrahim, M. G. (2015). Numerical solution of a system seir nonlinear odes by runge-kutta fourth order method. *International Journal of Computer Applications*, 124(3).

Huang, K.-Y. (2007). Application of artificial neural network for detecting phalaenopsis seedling diseases using color and texture features. *Computers and Electronics in agriculture*, 57(1):3–11.

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.

Maind, S. B., Wankar, P., et al. (2014). Research paper on basic of artificial neural network. *International Journal on Recent and Innovation Trends in Computing and Communication*, 2(1):96–100.

Malek, A. and Beidokhti, R. S. (2006). Numerical solution for high order differential equations using a hybrid neural network—optimization method. *Applied Mathematics and Computation*, 183(1):260–271.

Mall, S. and Chakraverty, S. (2012). Regression based neural network model for the solution of initial value problem. In *National conference on Computational and Applied Mathematics in Science and Engineering (CAMSE-2012)*.

McCulloch, W. S. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133.

McFall, K. S. (2010). Solving coupled systems of differential equations using the length factor artificial neural network method. *American Society of Mechanical Engineers Early Career Technical Journal*, 9:27–34.

Philemon, M. D., Ismail, Z., and Dare, J. (2019). A review of epidemic forecasting using artificial neural networks. *International Journal of Epidemiologic Research*, 6(3):132–143.

Umar, M., Sabir, Z., Raja, M. A. Z., and Sánchez, Y. G. (2020). A stochastic numerical computing heuristic of sir nonlinear model based on dengue fever. *Results in Physics*, 19:103585.