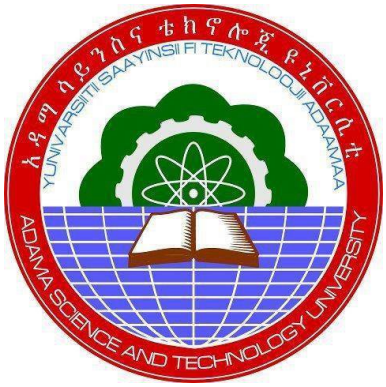
Solving Fractional Order Differential Equations Using Artificial Neural Network



Nugusu Kinfa Eshetu

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 Applied Mathematics
Office of Graduate Studies

Adama Science and Technology University

June, 2022

Adama, Ethiopia

Solving Fractional Order Differential Equations Using Artificial Neural Network.

Nugusu Kinfa Eshetu

ADVISOR: Tamirat Temesgen (PhD.)

A Thesis Submitted to the Department of Applied Mathematics
Presented in Partial Fulfillment of the Requirement for the degree of Master's Applied Mathematics
Office of Graduate Studies

School of Applied Natural Science

Adama Science and Technology University

June, 2022

Adama, Ethiopia

Declaration

I here by declare that this Masters Thesis entitled ***" Solving Fractional order differential equation using Artificial Neural Network"*** is my own work and has not been submitted to any university for similar purpose. All sources of materials that are used for this thesis have been duly acknowledged through citations.

Name of Student	Signature	Date

Recommendation of Advisors

We, the advisor of this thesis, here by certify that we have read the revised version of the thesis entitled "Solving Fractional Order Differential Equation Using Artificial Neural Networks" prepared under our guidance by Nugusu Kinfa Eshetu 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

Approval Page

We the advisor of the thesis entitled "Solving Fractional order differential equation using Artificial Neural Networks and developed by Nugusu Kinfa Eshetu, hereby certify that the recommendation and suggestion made by the board of the 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 Nugusu Kinfa Eshetu, have read and evaluated the thesis entitled "Solving Fractional order differential equation using artificial neural network" and examined the candidate during the open defense. This is, therefore to certify that the thesis is accepted for partial fulfillment of the requirement of the degree of Masters of science in Applied Mathematics.

Chairperson	Signature	Date
Internal Examiner	Signature	Date
External Examiner	Signature	Date

Finally approval and acceptance of the thesis are 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

First of all I would like to thanks for **My GOD** for giving me the strength from lower grade up to now with great accomplishment in what I would like to all the necessary things Second I would thanks **Tigist Kinfa** for heir financial, moral and material support. This contribute a great role for my complete work. Finally I would like to say thanks for my advisor **Dr**. **Tamirat Temesgen** who helped me by giving constrictive suggestion reference book and the work.

Contents

List of Tables

List of Figures

List of Acronyms and Symbols

ANN	Artificial Neural Network
FDE	Fractional Differential Equation
FODE	Fractional Order Differential Equation
ODE	Ordinary Differential Equation
SQP	Sequential Quadratic Programming
EMA	Exponential Moving Average
LeNN	Legendre Neural Network
FLANN	Functional Link Artificial Neural Network
IVP	Initial Value Problems
BVP	Boundary Value Problems
HeNN	Hermite Neural Network
ChNN	Chebyshev Neural Network

Symbols

α	Alpha (Order)
λ	Lambda (Constant numbers)
Γ	Gammas functions
η	Learning parameters
E	Error functions
k	Iteration step
ϵ	Constant numbers
β_1	Exponential decay rates for the moment estimates
β_2	Exponential decay rates for the moment estimates

Abstract

In this thesis, we find the solution of fractional order differential equations using Artificial Neural Network(ANN). In this study artificial neural network technique is developed to find solution of fractional order differential equations(FODE). The fractional differential equation has the advantage of being able to better represent a variety of real-world physical system application challenges. Here we have employed multi-layer feed forward neural architecture and error back propagation algorithm with unsupervised learning for minimizing the error function and modification of the parameters (weights and biases). Finally we compared the analytical solution and ANN solution for some illustrative examples.

Keywords: *Artificial Neural Network, Fractional Order Differential Equation, Back propagation algorithm, Unsupervised learning.*

CHAPTER 1

INTRODUCTION

1.1 Background of the study

History of Fractional Calculus

The history of fractional derivative started in 1695 by L'Hopetal, when he questioned Leibniz what would it mean $\frac{d^n f}{dx^n}$ in his letter, Leibniz answered that would be a paradox(Diethelm, 2010). This was the beginning of "fractional derivative" and influence on this new concept to a number of mathematicians like Laplace, Euler, Fourier, Lacroix, Riemann, Abel and Liouville. Lacroix was the first mathematician who released a paper mentioning fractional derivatives in it. He began with the polynomial $f(x) = x^m$, where m is a positive integer, and differentiated it n times where $m \geqslant n$ to get $\frac{D^n f}{Dx^n} = \frac{m!}{(m-n)!}x^{m-n}$ then he used Legendres symbol Γ to have $\frac{D^n f}{Dx^n} = \frac{\Gamma(m+1)}{\Gamma(m-n+1)}x^{m-n}$. Using this formula when $m = 1$ and $n = \frac{1}{2}$ he obtained $D^{\frac{1}{2}} f = \frac{2\sqrt{x}}{\sqrt{\pi}}$. Fractional calculus has become popular in the scientific community because it has numerous applications in various fields of science and engineering. Fractional calculus is the field of mathematical analysis which deals with the investigation and applications of integrals and derivatives of arbitrary order.

Fractional differential equations describe various phenomena such as fluid flow in porous material, anomalous diffusion transport, signal processing,control theory of dynamical systems, viscoelasticity etc. In most cases, analytical solutions of fractional order differential equations may not be obtained easily. Therefore, it is important to develop some reliable and efficient techniques to handle FDEs.

During the past decades, various numerical techniques have been developed to solve these equations. These methods include finite difference, Adomian decomposition, Wavelet method etc. Although these techniques provide good approximations to the solution, but most of these methods give series expansions in the neighborhood of initial

conditions are used. Besides that, rounding-off errors solemnly influence the solution precision in the numerical methods with complicatedness that also increase quickly with the number of sampling points(Laroche & Knittel, 2005).

In last few years,various machine Learning procedures in particular Artificial Neural Network (ANN) methods have been established as a powerful technique to solve a variety of real-world problems because of its excellent learning capacity. Artificial neural networks (ANNs) have been a topic of great interest in the machine learning community due to their ability to solve very difficult problems, particularly in the fields of image processing and object recognition, speech recognition, medical diagnosis, etc(Graupe, 2013).

Artificial neural network (ANN) is one of the popular areas of artificial intelligence (AI) and a computational model based on the organizational structure of the human brain(McCulloch & Pitts, 1943). The simplest definition of ANN is provided by the inventor of one of the first neurocomputers, Dr.Robert Hecht-Nielsen(Mall & Chakraverty, 2014). He defines a neural network as a computing system made up of a number of simple, highly interconnected processing elements, which process information by their dynamic state response to external inputs.

ANNs are processing devices (algorithms) that are loosely modeled after the neuronal structure of the mammalian cerebral cortex but on much smaller scales. Computer scientists have always been inspired by the human brain. In 1943, Warren S. McCulloch, a neuroscientist, and Walter Pitts, a logician, developed the first conceptual model of an ANN (McCulloch & Pitts, 1943). In their paper, they describe the concept of a neuron, a single cell living in a network of cells that receives inputs, processes those inputs, and generates an output. ANN is a computational model or an information processing paradigms inspired by biological nervous system as show on ANN approach has attracted much consideration to its advantages such as learning, adaptive, error computation etc. a multi-layer ANN model with unsupervised back propagation learning algorithm for solving fractional order differential equations. The ANN approximate solution of FDE is written as sum of two terms, first one satisfies initial/boundary conditions and the second part involves output of neural network with adjustable parameters. Feed forward neural network model and error back propagation principles (gradient descent procedure) are used for modification of the network parameters and to minimize the computed error function. Initial weights from input layer to hidden layer and hidden layer to output layer are considered as random. The approximate

solution of FDEs by ANN is found to be advantageous but it depends upon the ANN model that one considers such as Solution search proceeds without coordinate transformations, Simple implementation and easy computation and The back propagation algorithm is unsupervised.

1.2 Preliminaries

This section includes definitions of conformable fractional derivative and integral equation, artificial neural network and theorems.

Definition 1 *Let $a \geq 0$ and $\alpha \in (0,1)$. Given a function $f : [a,b] \longrightarrow \mathbb{R}$. Then conformable fractional derivative of a function is defined by (Kareem, 2017)*

$$D^{\alpha} f(t) = \lim_{\epsilon \longrightarrow 0} \frac{f(t + \epsilon t^{1-\alpha}) - f(t)}{\epsilon} \tag{1.1}$$

Theorem 1.2.1 *If a function f is α-differentiable at $t_0 > 0$, $\alpha \in (0,1)$ then f is continuous at t_0(Khalil et al., 2014) .*

Proof. Let f^{α} is differentiable at $x = t_0$, we know that

$$D^{\alpha} f(t) = \lim_{\epsilon \longrightarrow 0} \frac{f(t_0 + \epsilon t_0^{1-\alpha}) - f(t_0)}{\epsilon}$$

exists.

If we next assume that $x \neq t_0$ we can write the following
$f(t_0 + \epsilon t_0^{1-\alpha}) - f(t_0) = \frac{f(t_0+\epsilon t_0^{1-\alpha}) - f(t_0)}{\epsilon} \epsilon$
Then some basic properties of limits give us
$\lim_{\epsilon \longrightarrow 0} f(t_0 + \epsilon t_0^{1-\alpha}) - f(t_0) = \lim_{\epsilon \longrightarrow 0} \frac{f(t_0+\epsilon t_0^{1-\alpha}) - f(t_0)}{\epsilon} . \lim_{\epsilon \longrightarrow 0} \epsilon$

$\lim_{\epsilon \longrightarrow 0} f(t_0 + \epsilon t_0^{1-\alpha}) - f(t_0) = f'(t_0).0$

Let $h = \epsilon t^{1-\alpha}$.Then ,
$\lim_{h \longrightarrow 0} f(t_0 + h) = f(t_0)$. Hence, f is continuous at t_0

Theorem 1.2.2 *Let $\alpha \in (0,1]$ and a functions f, g is α-differentiable at a point t then(Khalil et al., 2014)*

1. If $b > a > 0$ and $f : [a,b] \longrightarrow \mathbb{R}$ is differentiable function then f α differentiable function at $t > a$, then

$$D^{\alpha} f(t) = t^{1-\alpha} \frac{df}{dt}$$

2. Let $\alpha \in (0,1)$ and f, g be $\alpha-$ differentiable at point $t > 0$. then

$$D^{\alpha}(cf(t) + dg(t)) = cD^{\alpha}f(t) + dD^{\alpha}g(t), \forall c, d \in \mathbb{R}$$

3. Let $\alpha \in (0,1)$ and f be $\alpha-$ differentiable power functions at point $t > 0$. then

$$D^{\alpha}(t^{p}) = pt^{p-\alpha}, \forall p \in \mathbb{R}$$

4. Let $\alpha \in (0,1)$ and f be constant functions $\alpha-$ differentiable at point $t > 0$. then

$$D^{\alpha}(\lambda) = 0$$

5. Let $\alpha \in (0,1)$ and f, g be $\alpha-$ differentiable at point $t > 0$. then

$$D^{\alpha}(fg)(t) = f(t)D^{\alpha}g(t) + g(t)D^{\alpha}f(t)$$

6. Let $\alpha \in (0,1)$ and f, g be $\alpha-$ differentiable at point $t > 0$. then

$$D^{\alpha}(\frac{f(t)}{g(t)}) = \frac{g(t)D^{\alpha}f(t) - f(t)D^{\alpha}g(t)}{(g(t))^{2}}$$

proof

By definition of conformable fractional derivative and taking $h = \epsilon t^{1-\alpha}$

1. The fractional derivative any α- differentiable functions

$$\begin{aligned} D^{\alpha}f(t) &= \lim_{\epsilon \longrightarrow 0} \frac{f(t + \epsilon t^{1-\alpha}) - f(t)}{\epsilon} \\ &= \lim_{\epsilon \longrightarrow 0} \frac{f(t+h) - f(t)}{ht^{\alpha-1}} \\ &= t^{1-\alpha} \lim_{h \longrightarrow 0} \frac{f(t+h) - f(t)}{h} \\ &= t^{1-\alpha} f'(t) \end{aligned}$$

2. A fractional derivative to add two functions

$$\begin{aligned} D^{\alpha}(cf+dg)(t) &= \lim_{\epsilon \longrightarrow 0} \frac{(cf+dg)(t+\epsilon t^{1-\alpha}) - (cf+dg)(t)}{\epsilon} \\ &= \lim_{\epsilon \longrightarrow 0} \frac{cf(t+\epsilon t^{1-\alpha} + dg(t+\epsilon t^{1-\alpha}) - cf(t) - dg(t))}{\epsilon} \\ &= \lim_{\epsilon \longrightarrow 0} \frac{cf(t+\epsilon t^{1-\alpha}) - f(t)}{\epsilon} + \lim_{\epsilon \longrightarrow 0} \frac{dg(t+\epsilon t^{1-\alpha}) - g(t)}{\epsilon} \\ &= cD^{\alpha}f(t) + dD^{\alpha}g(t) \end{aligned}$$

3. For any constant function the fractional derivative is zero

$$\begin{aligned} D^{\alpha}(\lambda) &= \lim_{\epsilon \longrightarrow 0} \frac{f(t+\epsilon t^{1-\alpha}) - f(t)}{\epsilon} \\ &= \lim_{\epsilon \longrightarrow 0} \frac{\lambda - \lambda}{\epsilon} \\ &= 0 \end{aligned}$$

4. The fractional derivative for power function using binomial property $(c+d)^n = \sum_{k=1}^{n} \binom{n}{k} c^{n-k} d^k$

   Thus,

$$\begin{aligned} (t+\epsilon t^{1-\alpha})^p &= \sum_{k=1}^{p} \binom{p}{k} t^{p-k} (\epsilon t^{1-\alpha})^k \\ &= (t+\epsilon t^{1-\alpha})^p \\ &= \binom{p}{0} t^p + \binom{p}{1} t^{p-1} (\epsilon t^{1-\alpha}) t^1 + \cdots + \binom{p}{p} t^0 (\epsilon t^{1-\alpha})^p \end{aligned}$$

   To proof that

   $D^{\alpha}(t^p) = pt^{p-\alpha}$

$$\begin{aligned} &\lim_{\epsilon \longrightarrow 0} \frac{t^p + \binom{p}{1} t^{p-1} (\epsilon t^{1-\alpha}) + \cdots + \binom{p}{p} \epsilon^{p(t^{1-\alpha})^p} - t^p}{\epsilon} \\ &= \lim_{\epsilon \longrightarrow 0} \epsilon \frac{pt^{p-1}t^{1-\alpha} + \cdots + \binom{p}{p} \epsilon^{p-1} (t^{1-\alpha})^p}{\epsilon} \\ &= pt^{p-1}t^{1-\alpha} \\ &= pt^{p-\alpha} \end{aligned}$$

5. A fractional derivative to multiply two functions

$$\begin{aligned}
D^{\alpha}(fg) &= \lim_{\epsilon \longrightarrow 0} \frac{f(t+\epsilon t^{1-\alpha})g(t+\epsilon t^{1-\alpha}) - f(t)g(t)}{\epsilon} \\
&= \lim_{\epsilon \longrightarrow 0} \frac{f(t+\epsilon t^{1-\alpha})g(t+\epsilon t^{1-\alpha}) + f(t)g(t+\epsilon t^{1-\alpha}) - f(t)g(t+\epsilon t^{1-\alpha}) - f(t)g(t)}{\epsilon} \\
&= g(t) \lim_{\epsilon \longrightarrow 0} \frac{f(t+\epsilon t^{1-\alpha})g(t+\epsilon t^{1-\alpha})}{\epsilon} + f(t) \lim_{\epsilon \longrightarrow 0} \frac{g(t+\epsilon t^{1-\alpha}) - g(t)}{\epsilon} \\
&= D^{\alpha}f(t) \lim_{\epsilon \longrightarrow 0} \frac{g(t+\epsilon t^{1-\alpha})}{\epsilon} + g(t)D^{\alpha} \lim_{\epsilon \longrightarrow 0} \frac{g(t+\epsilon t^{1-\alpha})}{\epsilon} \\
&= g(t)D^{\alpha}f(t)f(t)D^{\alpha}g(t) \\
&= g(t)D^{\alpha}f(t) + f(t)D^{\alpha}g(t)
\end{aligned}$$

6. The fractional derivative of fractional function

$$\begin{aligned}
D^{\alpha}(\frac{f}{g}) &= \lim_{\epsilon \longrightarrow 0} \frac{\frac{f(t+\epsilon t^{1-\alpha})}{g(t+\epsilon t^{1-\alpha})} - \frac{f(t)}{g(t)}}{\epsilon} \\
&= \lim_{\epsilon \longrightarrow 0} (\frac{f(t+\epsilon t^{1-\alpha})}{g(t+\epsilon t^{1-\alpha})} - \frac{f(t)}{g(t+\epsilon t^{1-\alpha})} + \frac{f(t)}{g(t+\epsilon t^{1-\alpha})} - \frac{f(t)}{g(t)}) \\
&= \lim_{\epsilon \longrightarrow 0} (\frac{f(t+\epsilon t^{1-\alpha}) - f(t)}{\epsilon g(t+\epsilon t^{1-\alpha})} + f(t). \lim_{\epsilon \longrightarrow 0} (\frac{1}{\epsilon g(t+\epsilon t^{1-\alpha})} - \frac{1}{\epsilon g(t)}) \\
&= D^{\alpha}f \lim_{\epsilon \longrightarrow 0} \frac{1}{g(t+\epsilon t^{1-\alpha})} + f(t) \lim_{\epsilon \longrightarrow 0} (-(\frac{g(t+\epsilon t^{1-\alpha} - g(t))}{\epsilon . g(t)g(t+\epsilon t^{1-\alpha})})) \\
&= D^{\alpha}f \frac{1}{g(t)} - f(t) \lim_{\epsilon \longrightarrow 0} (\frac{g(t+\epsilon t^{1-\alpha}) - g(t)}{\epsilon}). \lim_{\epsilon \longrightarrow 0} \frac{1}{g(t)g(t+\epsilon t^{1-\alpha})} \\
&= \frac{D^{\alpha}f(t)}{g(t)} - f(t)D^{\alpha}g(t).\frac{1}{g^2(t)} \\
&= \frac{g(t)D^{\alpha}f(t) - f(t)D^{\alpha}g(t)}{g^2(t)}
\end{aligned}$$

Theorem 1.2.3 *The conformable fractional derivative of a function is expressed as follows (Khalil et al., 2014)*

1. The fractional derivative of exponential function

$$D^{\alpha}(e^{ct}) = ct^{1-\alpha}e^{ct}, c \in \mathbb{R}$$

2. The fractional derivative of $\sin(t)$ function

$$D^{\alpha}(\sin(at)) = at^{1-\alpha}\cos(at), \quad a \in \mathbb{R}$$

3. The fractional derivative of $\cos(at)$ function

$$D^{\alpha}(\cos(at)) = -at^{1-\alpha}\sin(at), a \in \mathbb{R}$$

4. The fractional derivative of $\tan(at)$ function

$$D^{\alpha}(\tan(at)) = at^{1-\alpha}\sec^2(at), a \in \mathbb{R}$$

5. The fractional derivative of $\cot(at)$ function

$$D^{\alpha}(\cot(at)) = -at^{1-\alpha}\csc^2(at), a \in \mathbb{R}$$

6. The fractional derivative of $\sec(at)$ function

$$D^{\alpha}(\sec(at)) = at^{1-\alpha}\sec(at)\tan(at), a \in \mathbb{R}$$

7. The fractional derivative of $\csc(at)$ function

$$D^{\alpha}(\csc(at)) = -at^{1-\alpha}\csc(at)\cot(at), a \in \mathbb{R}$$

1. **Proof.**

   By using definition of conformable derivative and taking $h = \epsilon t^{1-\alpha}$

$$\begin{aligned} D^{\alpha}(e^{ct}) &= \lim_{\epsilon \longrightarrow 0} \frac{e^{c(t+\epsilon t^{1-\alpha})} - e^{ct}}{\epsilon} \\ &= e^{ct} \lim_{\epsilon \longrightarrow 0} \frac{e^{\epsilon t^{1-\alpha}} - 1}{\epsilon} \\ &= e^{ct} \lim_{\epsilon \longrightarrow 0} \frac{t^{1-\alpha}e^{ct^{1-\alpha}} - t^{1-\alpha}}{\epsilon t^{1-\alpha}} \end{aligned}$$

   Let $h = \epsilon t^{1-\alpha}$

   Then by using L'Hopital's rule, we get

$$\begin{aligned} &= t^{1-\alpha}e^{ct} \lim_{\epsilon \longrightarrow 0} \frac{e^{ch} - 1}{h} \\ &= ct^{1-\alpha}e^{ct} \lim_{\epsilon \longrightarrow 0} \frac{e^{ch}}{1} \\ &= ct^{1-\alpha}e^{ct} \end{aligned}$$

2. By using definition of conformable derivative and trigonometric property

$$\begin{aligned} D^{\alpha}(\sin(at)) &= \lim_{\epsilon \longrightarrow 0} \frac{\sin a(t+\epsilon t^{1-\alpha}) - \sin(at)}{\epsilon} \\ &= \lim_{\epsilon \longrightarrow 0} \frac{\sin(at)\cos(a\epsilon t^{1-\alpha}) + \cos(at)\sin(a\epsilon t^{1-\alpha}) - \sin(at)}{\epsilon} \\ &= \lim_{\epsilon \longrightarrow 0} \sin(at) \frac{[\cos(a\epsilon t^{1-\alpha}) - 1]}{\epsilon t^{1-\alpha}} + t^{1-\alpha}\cos(at) \lim_{\epsilon \longrightarrow 0} \frac{\sin(a\epsilon t^{1-\alpha})}{\epsilon t^{1-\alpha}} \end{aligned}$$

Let taking $h = \epsilon t^{1-\alpha}$ then we get

$$= t^{1-\alpha}\sin(at) \lim_{h \longrightarrow 0} \frac{[\cos(ah) - 1]}{h} + t^{1-\alpha}\cos(at) \lim_{h \longrightarrow 0} \frac{\sin(ah)}{h}$$

By using L'Hoputal Rule, we get

$$\begin{aligned} & t^{1-\alpha}\sin(at) \lim_{h \longrightarrow 0} \frac{-a\sin(ah)}{1} + t^{1-\alpha}\cos(at).a \\ &= at^{1-\alpha}\cos(at) \end{aligned}$$

3. By using definition of conformable derivative and trigonometric property

$$\begin{aligned} D^{\alpha}(\cos(at)) &= \lim_{\epsilon \longrightarrow 0} \frac{\cos a(t+\epsilon t^{1-\alpha}) - \cos(at)}{\epsilon} \\ &= \lim_{\epsilon \longrightarrow 0} \frac{\cos(at)\cos(a\epsilon t^{1-\alpha}) - \sin(at)\sin(a\epsilon t^{1-\alpha}) - \cos(at)}{\epsilon} \\ &= \lim_{\epsilon \longrightarrow 0} \cos(at)[\frac{\cos(a\epsilon t^{1-\alpha}) - 1}{\epsilon}] - \lim_{\epsilon \longrightarrow 0} t^{1-\alpha}\frac{\sin(at)\sin(a\epsilon t^{1-\alpha})}{\epsilon}t^{1-\alpha} \\ &= t^{1-\alpha}\lim_{\epsilon \longrightarrow 0} \cos(at)[\frac{\cos(a\epsilon t^{1-\alpha}) - 1}{\epsilon t^{1-\alpha}}] - \lim_{\epsilon \longrightarrow 0} t^{1-\alpha}\frac{\sin(at)\sin(a\epsilon t^{1-\alpha})}{\epsilon}t^{1-\alpha} \\ &= \lim_{h \longrightarrow 0} t^{1-\alpha}\cos(at)[\frac{\cos(ah) - 1}{h}] - \lim_{h \longrightarrow 0} t^{1-\alpha}\frac{\sin(a\epsilon t^{1-\alpha})}{h}.\sin(at) \\ &= \cos(at)t^{1-\alpha}\lim_{h \longrightarrow 0} \frac{\cos(ah) - 1}{h} - t^{1-\alpha}\sin(at)t^{1-\alpha}\lim_{h \longrightarrow 0} \frac{\sin(ah)}{h} \\ &= t^{1-\alpha}\cos(at)\lim_{h \longrightarrow 0} \frac{\cos(ah) - 1}{h} - t^{1-\alpha}\sin(at)\lim_{h \longleftrightarrow 0} \frac{\sin(ah)}{h} \\ &= -at^{1-\alpha}\cos(at).\lim_{h \longleftrightarrow 0} \frac{\sin(ah)}{h} - t^{1-\alpha}\sin(at).a \\ &= -at^{1-\alpha}\sin(at) \end{aligned}$$

4. By using definition of conformable fractional derivative, theorem(1.2.2) and trigono-

metric property

$$\begin{aligned} D^{\alpha}(\tan(at)) &= D^{\alpha}(\frac{\sin(at)}{\cos(at)}) \\ &= \frac{\cos(at)D^{\alpha}(\sin(at)) - \sin(at)D^{\alpha}\cos(at)}{\cos^2(at)} \\ &= \frac{\cos(at)(at^{1-\alpha}\cos(at)) - (\sin(at)(at^{1-\alpha}\sin(at)))}{\cos^2(at)} \\ &= \frac{at^{1-\alpha}\cos^2(at) + at^{1-\alpha}\sin^2(at)}{\cos^2(at)} \\ &= at^{1-\alpha}(1+\tan^2(at)) \\ &= at^{1-\alpha}\sec^2(at) \end{aligned}$$

5. By using definition of conformable fractional derivative ,theorem(1.2.2) and trigonometric property

$$\begin{aligned} D^{\alpha}(\cot)(at) &= D^{\alpha}(\frac{\cos(at)}{\sin(at)}) \\ &= \frac{\sin(at)D^{\alpha}(\cos(at)) - \cos(at)D^{\alpha}(\sin(at))}{\sin^2(at)} \\ &= \frac{\sin(at)(at^{1-\alpha}\sin(at)) - \cos(at)(at^{1-\alpha})\cos(at)}{\sin^2(at)} \\ &= -at^{1-\alpha}(\frac{\sin^2(at)+\cos^2(at)}{\sin^{at}}) \\ &= at^{1-\alpha}\frac{1}{\sin^2(at)} \\ &= -at^{1-\alpha}\csc^2(at) \end{aligned}$$

6. By using definition of conformable fractional derivative ,theorem(1.2.2) and trigonometric property

$$\begin{aligned} D^{\alpha}(\sec(at)) &= D^{\alpha}(\frac{1}{\cos(at)}) \\ &= \frac{(-1)D^{\alpha}(\cos(at))}{\cos^2(at)} \\ &= \frac{(-1)(at^{1-\alpha}\sin(at))}{\cos^2(at)} \\ &= at^{1-\alpha}\frac{\sin(at)}{\cos(at)}.\frac{1}{\cos(at)} \\ &= at^{1-\alpha}\tan(at)\sec(at) \end{aligned}$$

7. By using definition of conformable fractional derivative ,theorem(1.2.2) and trigono-

metric property

$$\begin{aligned} D^{\alpha}(csc(at)) &= D^{\alpha}(\frac{1}{\sin(at)}) \\ &= \frac{(-1)D^{\alpha}(\sin(at))}{\sin^2(at)} \\ &= \frac{(-1)(at^{1-\alpha}\cos(at))}{\sin^2(at)} \\ &= -at^{1-\alpha}\frac{\cos(at)}{\sin^2(at)} \\ &= -at^{1-\alpha}\frac{\cos(at)}{\sin(at)}.\frac{1}{\sin(at)} \\ &= -at^{1-\alpha}\cot(at).\csc(at) \end{aligned}$$

Definition 2 *Let $\alpha \in (n, n+1]$, $n \in N$ and f be an n differentiable at t, where $t > 0$. Then the conformable fractional derivative of f of order α is defined as(Kareem, 2017)*

$$D^{\alpha}f(t) = \lim_{\epsilon \longrightarrow 0} \frac{f^{([\alpha]-1))}(t + \epsilon t^{([\alpha]-\alpha)}) - f^{([\alpha]-1)}(t)}{\epsilon} \tag{1.2}$$

where $[\alpha]$ is the smallest integer greater than or equal to α.

Theorem 1.2.4 *(Kareem, 2017) Rolle's Theorem for conformable fractional differentiable function. Let $a > 0$ and $f : [a, b] \longrightarrow \mathbb{R}$ be a given function that satisfies*

1. f is continuous on $[a, b]$,
2. f is α-differentiable for some $\alpha \in (0, 1)$,
3. $f(a) = f(b)$.

Then, there exists $c \in (a, b)$, such that $f^{(\alpha)}(c) = 0$.

Proof:- Since f is continuous on $[a, b]$, and $f(a) = f(b)$, there is $c \in (a, b)$, which is a point of local extrema. With no loss of generality, assume c is a point of local minimum. So $f^{(\alpha)}(c) = \lim_{\epsilon \longrightarrow 0^+} \frac{f(c+\epsilon c^{1-\alpha}) - f(c)}{\epsilon} = \lim_{\epsilon \longrightarrow 0^-} \frac{f(c+\epsilon c^{1-\alpha}) - f(c)}{\epsilon}$, but the first limit is non - negative, and the second limit is non-positive. Hence, $f^{(\alpha)}(c) = 0$.

Theorem 1.2.5 *(Khalil et al., 2014) Mean Value Theorem for Conformable Fractional Differentiable Functions Let, $\alpha > 0$ and $f : [a, b] \longrightarrow \mathbb{R}$ be a given function that satisfies:-*

1. f is continuous on $[a,b]$.

2. f is α-differentiable for some $\alpha \in$ (0,1),

   Then, there exists $c \in (a,b)$, such that

$$f^{(\alpha)}(c) = \frac{f(b) - f(a)}{\frac{1}{\alpha}b^{\alpha} - \frac{1}{\alpha}a^{\alpha}}$$

Proof: The equation of the secant through $(a, f(a))and(b, f(b))$ is

$$y - f(a) = \frac{f(b) - f(a)}{\frac{1}{\alpha}b^{\alpha} - \frac{1}{\alpha}a^{\alpha}}(\frac{1}{\alpha}x^{\alpha} - \frac{1}{\alpha}a^{\alpha})$$

Which we can write as

$$y = \frac{f(b) - f(a)}{\frac{1}{\alpha}b^{\alpha} - \frac{1}{\alpha}a^{\alpha}}(\frac{1}{\alpha}x^{\alpha} - \frac{1}{\alpha}a^{\alpha}) + f(a)$$

Let

$$g(x) = f(x) - [\frac{f(b) - f(a)}{\frac{1}{\alpha}b^{\alpha} - \frac{1}{\alpha}a^{\alpha}}(\frac{1}{\alpha}x^{\alpha} - \frac{1}{\alpha}a^{\alpha}) + f(a)]$$

Note that $g(a) = g(b) = 0$,g is continuous on $[a,b]$ and differentiable on (a,b). So by Roll's theorem there are $c \in (a,b)$ such that $g^{(\alpha)}(c) = 0$.

But

$$g^{(\alpha)}(x) = f^{(\alpha)}(x) - [\frac{f(b) - f(a)}{\frac{1}{\alpha}b^{\alpha} - \frac{1}{\alpha}a^{\alpha}}]$$

So

$$g^{(\alpha)}(c) = f^{(\alpha)}(c) - [\frac{f(b) - f(a)}{\frac{1}{\alpha}b^{\alpha} - \frac{1}{\alpha}a^{\alpha}}] = 0$$

$$f^{(\alpha)}(c) = [\frac{f(b) - f(a)}{\frac{1}{\alpha}b^{\alpha} - \frac{1}{\alpha}a^{\alpha}}]$$

1.2.1 Conformable Fractional Integrals

Definition 3 *(Khalil et al., 2014) Let f be a continuous function then α-fractional integral of f is defined by:*

$$I^{\alpha}f(t) = I^{\alpha}(t^{\alpha-1}f(t)) = \int_a^t \frac{f(x)}{x^{1-\alpha}}dx \tag{1.3}$$

where $a > 0$,$\alpha \in (0, 1)$ and the integral is the usual Riemann improper integral.

1.2.2 Artificial Neural Network

Artificial neural networks are information processing structures providing the connection between input and output data by artificially simulating the physiological structure and functioning of human brain structures(Gallo, 2015).

Components of the Basic Artificial Neuron:

1. **Neuron(Node):** It is the basic unit of a neural network(Schalkoff, 1997).

2. **Input layer:** These layers are composed of neurons which are responsible for extracting patterns associated with the process or system being analyzed. These layers perform most of the internal processing from a network(Mehrotra et al., 1997).

3. **Hidden layer:** This is the medium layer in the neural network. Hidden layers have neurons (nodes) that apply different transformations to the input data(Mehrotra et al., 1997).

4. **Output layer :** This layer is also composed of neurons, and thus is responsible for producing and presenting the final network outputs, which result from the processing performed by the neurons in the previous layers(Mehrotra et al., 1997).

5. **Weights(Parameters):** Weights is a parameter of ANN. It represents the strength of the connection between nodes.Initially, it begins randomly. As training continuous, it adjust toward the desired values and the correct output. A weight bring down the importance of the input value. Weights near to 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(Mehrotra et al., 1997).

6. **Bias:** It is extra input neuron or node. It is used for shifting the activation function towards left or right, it can be referred to as ay intercept in th line equation. A low bias means the network is making more assumption about the form of the output, where as a high bias means the network is making fewer assumptions about the output(Mehrotra et al., 1997).

ANN Architecture

The structure of neural network architecture, which consists an input layer with single input node and a bias u_j, one hidden layer having five hidden nodes and output layer consists one output node. Initial weights w_j from input to hidden layer and v_j from hidden to output layer are considered as random.

Architecture of the three layer ANN with five hidden nodes, single input and output layer (with one node) is shown in Fig. 1.1

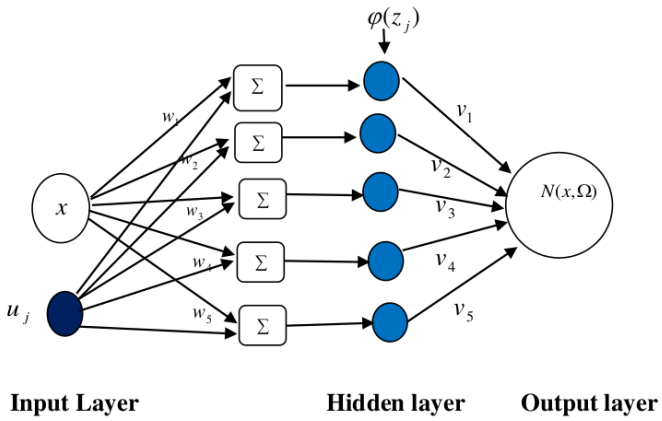


Figure 1.1: Artificial neural network architecture.

According to the structure of connections, different classes of neural network architecture can be identified as bellow (Mall & Chakraverty, 2016a).

Feed Forward Neural Network

In a feed-forward neural network, neurons are organized in the form of layers. Neurons in a layer receive input from the previous layer and feed their output to the next layer. Network connections to the same or previous layers are not allowed. Here, the data goes from the input node to the output node in a strictly feed-forward way. There is no feedback (back loops); that is, the output of any layer does not affect the same layer(Mall & Chakraverty, 2016a).

Here,$X = (x_1, x_2, \ldots, x_n)$ denotes the input vector, f is the activation function, and $O = (o_1, o_2, \ldots, o_m)$ is the output vector. Symbol $W = w_{ij}$ denotes the weight matrix or connection matrix, and $[WX]$ is the net input value, which is a scalar product of input vectors and weight vectors w_{ij}.

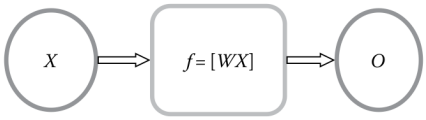


Figure 1.2: Block diagram of feed-forward ANN.

Feedback Neural Network

The output of one layer in a feedback neural network is routed back to the preceding layer. By introducing loops into the network, signals can go in both ways. This network is highly powerful and can become extremely intricate at times. All conceivable neuron connections are permitted. In optimization challenges, feedback neural networks are used to find the best arrangement of related components.

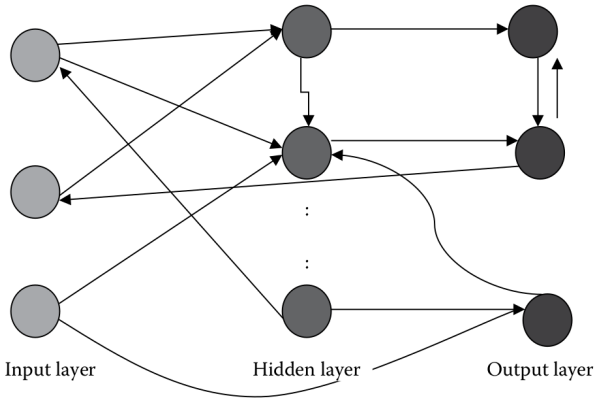


Figure 1.3: Diagram of feedback neural network.

Learning Methods

One of the most significant properties of ANN is its capacity to learn and generalize from a collection of training data. There are two sorts of learning situations in neural

networks: supervised and unsupervised learning.

Supervised Learning or Associate Learning:- The network is trained in supervised or Associate Learning by giving input and output patterns. External teachers or the system that houses the network can give these input to output pairings. To determine the mistake, the network's computed output is compared to the corrected expected output. The fault can then be used to alter network parameters, resulting in an increase in performance. (Mall & Chakraverty, 2016a).

Unsupervised or Self organization Learning:- In unsupervised training, the network is provided with inputs but not with desired out- puts. 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 adaptation. Unsupervised learning seems much harder than supervised learning, and this type of training generally fits into the decision problem framework because the goal is not to produce a classification but to make decisions that maximize rewards. An unsupervised learning task is to try to find the hidden structure in unlabeled data. In this type of learning, no desired or target is available to the network and only the input pattern is present, i.e. there is no teacher to learn the network. The system must learn by discovering and adapting to structured features in the input pattern. This is done by adapting to statistical regularities or clustering of patterns from the input training samples(Mall & Chakraverty, 2016a).

Activation or Transfer function

An activation or transfer function is a function that acts upon the net (input) to get the output of the network. It translates input signals to output signals. It acts as a squashing function such that the output of the neural network lies between certain values usually between 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). Sigmoid or Logistic Activation Function. The Sigmoid Function curves looks like a S-shape. it exist between 0 to 1. tanh is also like logistic sigmoid but better. The range of the tanh function is from -1 to 1. tanh also sigmoidal (S-shape). 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 nets.(Geremu, 2021)

In this study, we have used tangent hyperbolic activation functions only, which are continuously differentiable and tangent hyperbolic activation function resulted in the

most successful one for all the test parameters. tanh (hyperbolic tangent) function performs better recognition accuracy than those of the other functions. The out put of tangent hyperbolic activation functions lies between -1 to 1.
For example, the the tangent hyperbolic function is defines as is defines as

$$\varphi(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} = \frac{e^{2x} - 1}{e^{2x} + 1} \tag{1.4}$$

ANN Learning Rules

Learning is the most important characteristic of the ANN model. Every neural network possesses knowledge that is contained in the values of the connection weights. Most ANNs contain some form of "learning rule" that modifies the weights of the connections according to the input patterns. there are various kinds of learning rules used by neural networks such as (Rojas, 2013)

- Error back-propagation learning algorithm or delta learning rule
- Hebbian learning rule
- Perceptron learning rule
- Widrow–Hoff learning rule
- Winner-Take-All learning rule, etc.

Error Back-propagation Learning Algorithm or Delta Learning Rule:
Rumelhart et al. introduced the error back-propagation learning algorithm. (McClelland, Rumelhart, & Hinton, 1986). It is also known as the delta learning rule and is one of the most commonly used learning rules. This learning algorithm is valid for continuous activation function and is used in the supervised/ unsupervised training method.

We may determine the flow of error direction in the network by taking the partial derivative of error of the network with respect to each of its weights. The error will decrease until it approaches a local minimum if we take the negative derivative and then add it to the weights. If the derivative is negative, we must add a negative number to the weight or the reverse. Then, going from the output layer weights to the hidden layer weights, and then from the hidden layer weights to the input layer weights, these partial derivatives are applied to each of the weights.

The training of the network involves feeding samples as input vectors, calculating the error of the output layer, and then adjusting the weights of the network to minimize the error. The average of all the squared errors E for the outputs is computed to make the derivative simpler. After the error is computed, the weights can be updated one by one. The weights W_j of the bias b are initialized as randomly and then updated a follow (Parisi et al., 2003).

$$w_j^{k+1} = w_j^k + \triangle\, w_j^k + (-\eta \frac{\partial E(x,p)^k}{\partial w_j^k}) \tag{1.5}$$

where
η is the learning parameter (which is lies between 0 and 1)
k is the iteration step used to update the weights as usual in ANN
E is the error function

$$b^{k+1} = b^k + \triangle\, b^k = b^k + (-\eta \frac{\partial E(x,p)^k}{\partial b^k}) \tag{1.6}$$

where b is bias.

1.3 Statement of the problem

We consider fractional order differential equation,

$$D^\alpha y(x) = f(x, y(x)), \tag{1.7}$$

subject to

$$y(x_0) = y_0$$

where $n - 1 < \alpha \leq n$ $n \in N$.

Let $y_t(x,p)$ denotes the trail solution of ANN model with p is a vector containing corresponding weights and x is the input data. The above FODE is transformed into the ANN problem as (Mall & Chakraverty, 2018).

$$D^\alpha y_t(x,p) = f(x, y_t(x,p)). \tag{1.8}$$

The trail solution $y_t(x,p)$ of feed forward neural network with network parameters p

may be written in the form

$$y_t(x,p) = A(x) + F(x, N(x,p)). \tag{1.9}$$

The first term $A(x)$ is does not contain adjustable parameters and satisfies only initial condition whereas the second term $F(x, N(x,p))$ contain the single output $N(x,p)$ of feed forward neural network with input x and vector containing the corresponding weights p .

Here we consider a three layer ANN model with one input node x, one hidden layer consisting of m number of nodes and one output node $N(x,p)$. The output $N(x,p)$ is expressed as

$$N(x,p) = \sum_{j=1}^{m} v_j \phi(z), \tag{1.10}$$

Where $z_j = w_j x + u_j$ and w_j is weight from input to j^{th} hidden unit, v_j denotes the weight from j^{th} hidden unit to output unit and lastly u_j j^{th} is the bias for j^{th} hidden node. In this investigation we have considered the tangent hyperbolic function $\phi(x) = \frac{e^x - e^{-x}}{e^x + e^x}$ as an activation function. General form of corresponding error function for the fractional order differential equations may be formulated as

$$E(x,p) = \sum_{j=1}^{h} D^{\alpha} y_t(x_i,p)) - f(x_i, y_t(x_i,p))^2. \tag{1.11}$$

This study intended to the answer the following basic questions;

1. How to develop an algorithm for the artificial neural network method?
2. How to compare analytical and artificial neural network result?
3. How to solve fractional order differential equation using ANN method?

1.4 Objectives

1.4.1 General objective

The general objective of this study is to solve fractional order differential equations using artificial neural network.

1.4.2 Specific objective

The specific objectives of this study is to:

1. develop an algorithm for the artificial neural network method,
2. compare the analytical and artificial neural network solution,
3. formulate the artificial neural network method for solution of fractional order differential equation,

1.5 Significance of the study

The result of this study will help to:

- understand how to solve fractional differential equation in context of Artificial neural network.
- apply proposed method to solve fractional differential equation to get approximate solution.

CHAPTER 2

LITERATURE REVIEW

Some literature's relevant in this chapter are reviewed to solve for fractional differential equations including ordinary differential and partial differential. the first research group who gave a new definition of fractional derivative and fractional integral. The form of the definition shows that it looks like a derivative of calculus that they have studied before. They obtained many theorems in order to solve the fractional differential equations.

During the past decades, various numerical techniques have been developed to solve these equations. The variational iteration method and the Adomian decomposition method are implemented to give approximate solutions for linear and nonlinear systems of differential equations of fractional order. The two methods in applied mathematics can be used as alternative methods for obtaining analytic and approximate solutions for different types of differential equations. In these schemes, the solution takes the form of a convergent series with easily computable components(Momani & Odibat, 2007). They developed a framework to obtain approximate numerical solutions to ordinary differential equations (ODEs) involving fractional order derivatives using Legendre wavelet approximations. The properties of Legendre wavelets are first presented. These properties are then utilized to reduce the fractional ordinary differential equations (FODEs) to the solution of algebraic equations(Jafari et al., 2011).

Fractional differential equations have been investigated by variational iteration method. However, the previous works avoid the term of fractional derivative and handle them as a restricted variation. They proposed here in a fractional variational iteration method with modified Riemann Liouville derivative which is more efficient to solve the fractional differential equations(Wu & Lee, 2010).

The Analytical approximate solution of a fractional diffusion equation is deduced with the help of powerful Variational Iteration method. By using an initial value, the explicit solutions of the equation for different cases have been derived, which accelerate the rapid convergence of the series solution(Das, 2009).

In last few years, various machine intelligence procedures in particular Artificial Neural Network (ANN) methods have been established as a powerful technique to solve a variety of real-world problems because of its excellent learning capacity. ANN approach has attracted much consideration to its advantages such as learning, adaptive, error computation, fault-tolerance etc. Currently, a lot of attention has been devoted to the study of ANN for solving ordinary and partial differential equations(Zurada, 1992).

A single layer Legendre Neural Network (LeNN) model has been developed to solve initial and boundary value problems. In the proposed approach a Legendre polynomial based Functional Link Artificial Neural Network (FLANN) is developed. Nonlinear singular initial value problem (IVP), boundary value problem (BVP) and system of coupled ordinary differential equations are solved by the proposed approach to show the reliability of the method(Mall & Chakraverty, 2016a).

Hermite polynomial-based functional link artificial neural network (FLANN) is proposed here to solve the Van der Pol–Duffing oscillator equation. A single-layer Hermite neural network (HeNN) model is used, where a hidden layer is replaced by expansion block of input pattern using Hermite orthogonal polynomials. A feed forward neural network model with the unsupervised error back propagation principle is used for modifying the network parameters and minimizing the computed error function(Mall & Chakraverty, 2016b). A new algorithm has been proposed to solve singular initial value problems of Emden–Fowler type equations. Approximate solutions of these types of equations have been obtained by applying Chebyshev Neural Network (ChNN) model for the first time(Mall & Chakraverty, 2015).

Artificial neural networks approach made this non- parametric model a powerful tool in solving various complicated mathematical problems. The current research attempts to produce an approximate polynomial solution for special type of fractional order Volterra integrodifferential equations. The present technique combines the neural networks approach with the power series method to introduce an efficient iterative technique(Jafarian et al., 2017). A new method for solving the fractional differential equations of initial value problems by using neural networks which are constructed from cosine basis functions with adjustable parameters. By training the neural networks repeatedly the numerical solutions for the fractional differential equations were obtained. Moreover, the technique is still applicable for the coupled differential equations of fractional order(Qu & Liu, 2015).

Artificial neural networks approach made this non- parametric model a powerful tool in solving various complicated mathematical problems. The current research attempts to produce an approximate polynomial solution for special type of fractional order Volterra integrodifferential equations(Jafarian et al., 2017). A new computational intelligence technique is presented for solution of non-linear quadratic Riccati differential equations of fractional order based on artificial neural networks (ANNs) and sequential quadratic programming (SQP). The power of feed forward ANNs in an unsupervised manner is exploited for mathematical modeling of the equation; training of weights is carried out with an efficient constrained optimization technique based on the SQP algorithm. The proposed scheme is evaluated on two initial value problems of the Riccati fractional order equation with integer and non-integer derivatives. Comparison of results with the exact solution, and with reference numerical methods demonstrates the correctness of the proposed methodology. Performance of the proposed scheme is also validated using results of statistical analysis based on a sufficiently large number of independent runs(Raja et al., 2015).

CHAPTER 3

RESEARCH METHODOLOGY

3.1 Fractional Order Differential Equations

In this thesis we find the solution of Fractional Order Differential Equations (FODE) of the form

$$D^{\alpha}y + h(x)y = g(x) \tag{3.1}$$

$$D^{\alpha}y + h(x)y = 0 \tag{3.2}$$

Is called the homogeneous, If $g(x) \neq 0$ is called the non-homogeneous.
Subject to the initial condition

$$y(x_0) = y_0$$

where $\alpha \in (n-1, n]$, $\forall\ n \in N$

Theorem 3.1.1 *(Mahmood et al., 2021) The homogeneous solution of the conformable differential equation (3.2) is indicated by*

$$y_h(x) = ce^{-I^{\alpha}h(x)} = ce^{r\frac{1}{\alpha}x^{\alpha}} \tag{3.3}$$

where y_h is homogeneous solution and h is any continuous function in the domain of I^{α}.

proof: We have just verified that equation (3.2) is fulfilled by obtaining the function

$$y(x) = ce^{-I^{\alpha}h(x)} \tag{3.4}$$

We get by substituting into the above equation and applying theorem (1.2.2)

$$\begin{aligned} D^{\alpha}y + h(x)y &= cx^{1-\alpha}\frac{d}{dx}(e^{-I^{\alpha}h(x)}) + ch(x)e^{-I^{\alpha}h(x)} \\ &= -cx^{1-\alpha}\frac{d}{dx}(I^{\alpha}h(x))e^{-I^{\alpha}h(x)} + ch(x)e^{-I^{\alpha}h(x)} \\ &= -cx^{1-\alpha}\frac{h(x)}{x^{1-\alpha}}e^{-I^{\alpha}h(x)} + ch(x)e^{-I^{\alpha}h(x)} \\ &= 0 \end{aligned}$$

Theorem 3.1.2 *(Mahmood et al., 2021) The particular solution of the conformable differential equation* (3.1) *is expressed by*

$$y_p(x) = \lambda(x)e^{-I^{\alpha}h(x)} \tag{3.5}$$

Where h is any continuing function in the field of I^{α} and the function $\lambda : R \longrightarrow R$ is obtained through the following condition,

$$\lambda(x) = I^{\alpha}(g(x))e^{I^{\alpha}g(x)} \tag{3.6}$$

Proof: By obtaining the function, we have confirmed that equation (3.5) is satisfied

$$y_p(x) = \lambda(x)e^{-I^{\alpha}h(x)}$$

Substituting in the latter equation and applying the definition (3) We've got

$$\begin{aligned} D^{\alpha}y + h(x)y &= D^{\alpha}(I^{\alpha}k(x)e^{-I^{\alpha}h(x)})e^{-I^{\alpha}h(x)} + h(x)(I^{\alpha}(k(x))e^{-I^{\alpha}h(x)})e^{-I^{\alpha}h(x)} \\ &= k(x)e^{0} - I^{\alpha}(k(x)e^{-I^{\alpha}h(x)})h(x)e^{-I^{\alpha}h(x)} + h(x)(I^{\alpha}(k(x))e^{-I^{\alpha}h(x)})e^{-I^{\alpha}h(x)} \\ &= k(x) \end{aligned}$$

Remark: Therefore the Analytic solution of conformable fractional order differential

equation is written as the form of

$$y(x) = y_p(x) + y_h(x) \tag{3.7}$$

Bernoulli fractional differential equations

Bernoulli fractional differential equation have been solved the general solution of first order linear fractional differential equations, based on conformable fractional derivative. The linear fractional differential equation is the most common form of fractional differential equation, where the higher order conformable fractional derivative is a linear function of the lower order conformable fractional derivative consequently, the first order in general the conformable fractional derivative of fractional differential equation is defined as(Kareem, 2017)

$$D^{\alpha}y + h(x)y = g(x) \tag{3.8}$$

Where $h(x)$ and $g(x)$ are α time differentiable functions, and Equation (3.8) will be written as using theorem(1.2.2)

$$x^{1-\alpha}y^{'} + h(x)y = g(x) \tag{3.9}$$

$$y^{'} + \frac{h(x)}{x^{1-\alpha}}y = \frac{g(x)}{x^{1-\alpha}}$$

Equation (3.9) is a general solution to a first order linear ordinary differential equation

$$y = e^{-\int \frac{h(x)}{x^{1-\alpha}}dx}[\int \frac{g(x)}{x^{1-\alpha}}e^{\int \frac{h(x)}{x^{1-\alpha}}dx}dx + c] \tag{3.10}$$

Wherever c is a fixed value and arbitrary. Using (1.9) and replacement in equation (3.10),we can get

$$y = e^{-I^{\alpha}h(x)}[I^{\alpha}(g(x)e^{I^{\alpha}h(x)}) + c] \tag{3.11}$$

As a consequence, equation (3.11) has a general solution of equation (3.9).

Definition 4 *Bernoulli fractional differential equations can be expressed in the following way(Kareem, 2017).*

$$D^{\alpha}y + h(x)y = g(x)y^{n} \tag{3.12}$$

When $h(x)$ and $g(x)$ are α time differentiable function, and y is a function that is unknown. Equation (3.12) is obtained by applying theorem(3.11)

$$x^{1-\alpha}y^{'} + h(x)y = g(x)y^{n} \tag{3.13}$$

$$y^{'} + \frac{h(x)}{x^{1-\alpha}}y = \frac{g(x)}{x^{1-\alpha}}y^{n}$$

We know it will be linear for $n = 0$ or $n = 1$ and by changing the dependent variable, it can be reduced to a linear ordinary equation for every other value of n

$$y^{'}y^{-n} + \frac{h(x)}{x^{1-\alpha}}y^{1-n} = \frac{g(x)}{x^{1-\alpha}}$$

Let $z = y^{1-n}$ then $z^{'} = (1-n)y^{-n}y^{'}$

$$z^{'} + (1+n)\frac{h(x)}{x^{1-\alpha}}z = (1-n)\frac{g(x)}{x^{1-\alpha}}$$

According to the results the general solution is as follows

$$y = [e^{-I^{\alpha}((1-n)h(x))}(I^{\alpha}(1-n))g(x)e^{I^{\alpha}(1-n)h(x)} + c]^{\frac{1}{1-n}} \tag{3.14}$$

3.2 ANN Algorithm for FODE

Following the same procedure as (Dufera, 2021);

Step 1 Take discrete points $x_i \in [a, b,]$ where $i = 1, 2, 3, \ldots m$ and form an input vector $X = [x_1, x_2, x_3, \ldots x_m]$
Step 2 Initialize parameters(weights and bias):
–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 3 Forward propagation :
–$Z = WX + b$

–$N(X, W, b) = \tanh(Z)$

Step 4 Compute the cost and its gradient, using (4.17).
Step 5 Update the parameter using the methods of gradient descent or its family . In this thesis we use the Adam, adaptive moment.
–Gradient descent
–Adam, adaptive moment :(Jagannath et al., 2018)

•initialize $w_0, b_0, m_0 \longleftarrow 0 u_0 \longleftarrow 0, x \longleftarrow 0$

• while W_x, b_x not converged

$$x \longleftarrow x + 1 \tag{3.15}$$

$$g_t w \longleftarrow \bigtriangledown w f_x(W_x - 1) \tag{3.16}$$

$$g_t b \longleftarrow \bigtriangledown b f_x(b_x - 1) \tag{3.17}$$

$$m_{xw} \longleftarrow \beta_1 m_{xw-1} + (1 - \beta_1) g_{xw} \tag{3.18}$$

$$m_{xb} \longleftarrow \beta_1 m_{xb-1} + (1 - \beta_1) g_{xb} \tag{3.19}$$

$$u_{xw} \longleftarrow \beta_2 u_{xw-1} + (1 - \beta_2)(g_{xw})^2 \tag{3.20}$$

$$u_{xw} \longleftarrow \beta_2 u_{xb-1} + (1 - \beta_2)(g_{xb})^2 \tag{3.21}$$

bias correction

$$m_{xw} \longleftarrow \frac{m_{xw}}{1 - \beta_1^x}, V_{xw} \longleftarrow \frac{u_{xw}}{1 - \beta_2^x} \tag{3.22}$$

$$m_{xb} \longleftarrow \frac{m_{xb}}{1-\beta_1^x}, V_{xb} \longleftarrow \frac{u_{xb}}{1-\beta_2^x} \tag{3.23}$$

Update

$$W_x \longleftarrow W_{x-1} - \alpha \frac{m_{xw}}{\sqrt{u_{xw}} + \epsilon} \tag{3.24}$$

$$b_x \longleftarrow b_{x-1} - \alpha \frac{m_{xb}}{\sqrt{u_{xb}} + \epsilon} \tag{3.25}$$

Where $f(W, b)f$ is the loss function to be optimized given the parameters (weights) W and bias b.
$g - x$:is the gradient at step x.
$m - x$:m is the exponential moving average (EMA) of $(g - x)$.
$u - x$:u is the exponential moving average (EMA) of $(g - x)^2$
β_1, β_2:These are the hyperparameteres used in the moving averages of $g-x$ and $(g-x)^2$, most commonly set to 0.9 and 0.999 respectively.
α:The learning rate.
ϵ:Avery small number, used to avoid the denominator= 0 scenarios.
So what's happening here is we have a loss function $f(W, b)$ that is to be minimized by finding the optimal values of W and b such that the loss function achieve its minima. In order to do that, we use gradient descent where we basically compute the gradients of the function and keep subtracting them from the weights and bias until we reach the minimum (Ismael & Şengür, 2021).

$$w_j^{k+1} = w_j^k + \triangle\ w_j^k + (-\eta \frac{\partial E(x,p)^k}{\partial w_j^k})$$
$$b^{k+1} = b^k + \triangle\ b^k = b^k + (-\eta \frac{\partial E(x,p)^k}{\partial b^k})$$

CHAPTER 4

RESULTS AND DISCUSSIONS

This chapter describe the general formulation of fractional order differential equations using ANN and Applications of Fractional Order Differential Equations including with result and discussion.

4.1 Formulation of the Problem

A general form of fractional order differential equation with initial value problem maybe written as(Mall & Chakraverty, 2018)

$$D^{\alpha}y(x) = f(x, y(x)), x \in D \subset \mathbb{R} \tag{4.1}$$

$$y(x_0) = y_0$$

where $\alpha \in (n-1, n]$, $\forall\ n \in N$

where $D^{\alpha}y(x)$ is conformable fractional derivative of $y(x)$ of order α,$y(x)$ denotes the solution of conformable fractional order differential equation and D is the discretized domain over a finite set of points.

Transformation

Let $y_t(x, p)$ is the trail solution of feed forward neural network with neural network parameters p and the input data x then the above equation (4.1)fractional differential equation transform into the following problem :

$$D^{\alpha}y_t(x, p) = f(x, y_t(x, p)) \tag{4.2}$$

In general the ANN trail solution $y_t(x, p)$ for equation (4.2) with parameters (weights) p may be expressed as

$$y_t(x,p) = A(x) + F(x, N(x,p)) \tag{4.3}$$

Where the first term $A(x)$ be does not contain adjustable parameters and satisfies only initial conditions, whereas the second term $F(x, y_t(x,p))$ contain the single output $N(x,p)$ of feed forward neural network with input x and adjustable parameter p.

Specially, the ANN trail solution for equation (4.2) is expressed as

$$y_t(x,p) = y_0 + (x - x_0)N(x,p) \tag{4.4}$$

where
N(x,p) is feed forward neural network with input x and adjustable parameter p Thus, the trail solution $y_t(x,p)$ satisfies the initial conditions.

As such, $N(x,p)$ the neural network output with input x and parameters (weights) p may be computed as

$$N(x,p) = \tanh(z_j) = \frac{e^{z_j} - e^{-z_j}}{e^{z_j} + e^{-z_j}}, \tag{4.5}$$

where,

$$z_j = w_j x + u_j. \tag{4.6}$$

Here, x is input data w_j is the weight vectors of ANN , u_j is a bias and $j = 1, 2, 3, \ldots, m$. The nonlinear tangent hyperbolic function is considered as the activation function.

Gradient Computation of the ANN

For minimizing the error function $E(x,p)$, we differentiate $E(x,p)$ with respect to the network parameters. Thus the gradient of the network output with respect to its input is computed as follows. The fractional derivative of $N(x,p)$ according to conformable fractional derivative with respect to input x is

$$D^{\alpha}(N(x,p)) = v_j \phi^{'}(z_j) w_j x^{1-\alpha}. \tag{4.7}$$

Let G be the derivative of the output with respect to its input then $D^{\alpha}(N(x,p)) = G$.

The gradient of the output with respect to the network parameters of the ANN may be formulated as The derivative of the network output with respect to other parameters according to conformable fractional derivative

$$\frac{\partial G}{\partial w_j} = v_j x^{1-\alpha}(\phi'(z_j) + \phi''(z_j) w_j x) \tag{4.8}$$

$$\frac{\partial G}{\partial v_j} = w_j x^{1-\alpha}\phi'(z_j) \tag{4.9}$$

$$\frac{\partial G}{\partial u_j} = w_j v_j x^{1-\alpha}\phi'(z_j) \tag{4.10}$$

Here $N(x,p) = \sum_{j=1}^{m} v_j \phi(z_j)$ $z_j = w_j x + u_j$ From (4.3) we have by differentiating according to conformable fractional derivative

$$D^{\alpha} y_t(x,p) = (x-x_0)^{1-\alpha}(N(x,p)) + (x-x_0) D^{\alpha}(N(x,p)) \tag{4.11}$$

simplifying this equation

$$D^{\alpha} y_t(x,p) = (x-x_0)^{1-\alpha}(N(x,p)) + (x-x_0) w_j v_j \phi'(z_j) x^{1-\alpha}. \tag{4.12}$$

The error function to be minimized for the fractional order differential equation is found to be

$$E(x,p) = \sum_{i=1}^{h} [D^{\alpha} y_t(x_i,p) - F(x_i, y_t(x_i,p))]^2. \tag{4.13}$$

Updating weights and bias

To update the weights we required to find the derivatives of the error function with respect to w_j and is written as

$$\frac{\partial E(x,p)}{\partial w_j} = \frac{\partial}{\partial w_j} \sum_{i=1}^{h} [D^{\alpha} y_t(x_i,p) - F(x_i, y_t(x_i,p))]^2 \tag{4.14}$$

Know,the weights are updated by taking negative gradient at each iteration as

$$w_j^{k+1} = w_j^k + \triangle\, w_j^k + (-\eta \frac{\partial E(x,p)^k}{\partial w_j^k}) \tag{4.15}$$

And the bias are updated by taking negative gradient at each iteration as follows

$$b^{k+1} = b^k + \triangle\, b^k = b^k + (-\eta \frac{\partial E(x,p)^k}{\partial b^k}) \tag{4.16}$$

4.2 Applications for Solving Fractional Order Differential Equations

Now this section we will solve fractional order differential equations according to conformable definition:

Example 1 Here we consider fractional order differential equation given by;

$$D^\alpha y(x) = x, x \in [0,1]$$

with initial conditions $y(0) = 0$ and $0 < \alpha \leq 1$ find exact solution and ANN trail solutions.

Solution: According to Bernoulli methods find exact solutions of the fractional order differential equations applying equation 3.10 we get

$$y(x) = \frac{x^{\alpha+1}}{\alpha+1}$$

In this case,the ANN trail solution is represented as

$$\begin{aligned} y_t(x,p) &= y_0 + (x - x_0)N(x,p) \\ &= 0 + (x-0)N(x,p) \\ &= xN(x,p) \end{aligned}$$

In this example, the network has been trained with 10000 epochs in the domain from $x = 0$ to $x = 1$ for computing the results. The network has been trained here for 10 equidistant points in $[0,1]$ and 7 hidden nodes.

Table 4.1: Exact and ANN results for $\alpha = 0.5$

x	exact solution	ANN solution
0	0	0
0.1000	0.0238	0.0235
0.2000	0.0673	0.0663
0.3000	0.1236	0.1285
0.4000	0.1903	0.1900
0.5000	0.2660	0.2676
0.6000	0.3496	0.3287
0.7000	0.4406	0.4389
0.8000	0.5383	0.5481
0.9000	0.6423	0.6595
1.0000	0.7523	0.7595

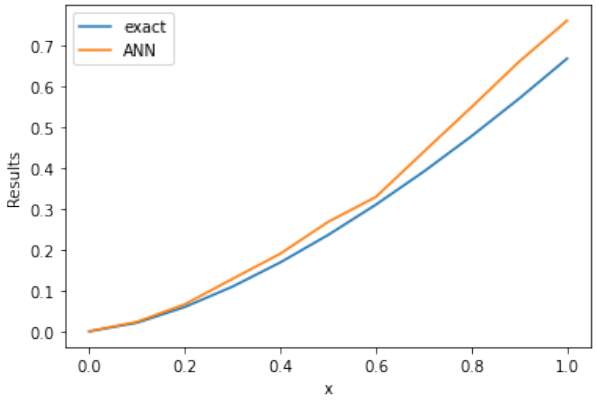


Figure 4.1: Plot of exact and ANN results for $\alpha = 0.5$.

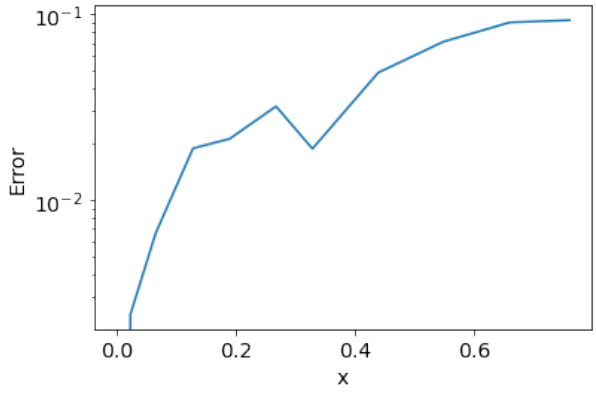


Figure 4.2: Plot of error between exact and ANN results for $\alpha = 0.5$.

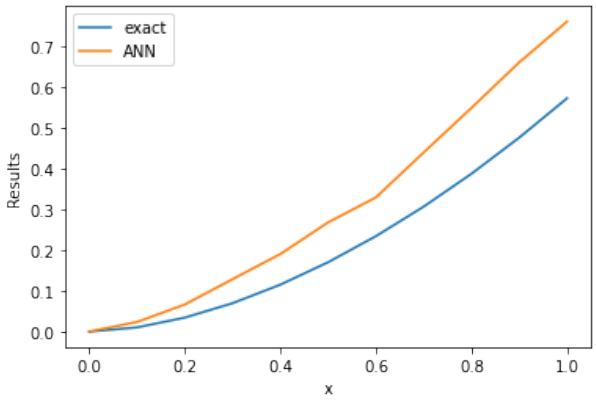


Figure 4.3: Plot of exact and ANN results for $\alpha = 0.75$.

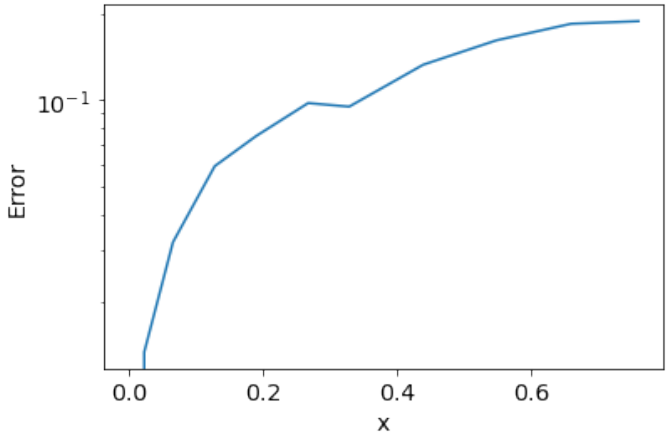


Figure 4.4: Plot of error between exact and ANN results for $\alpha = 0.75$.

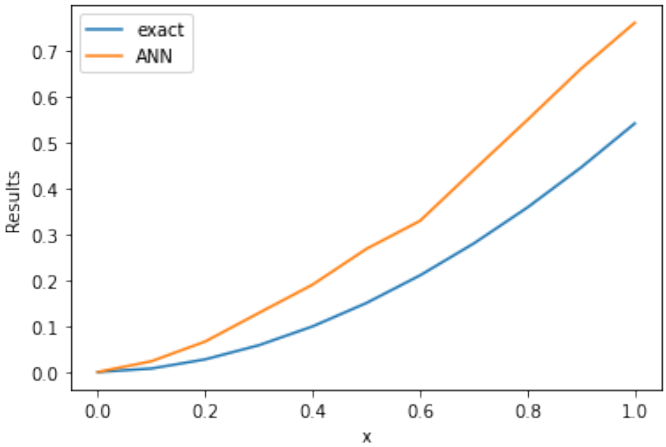


Figure 4.5: Plot of exact and ANN results for $\alpha = 0.85$.

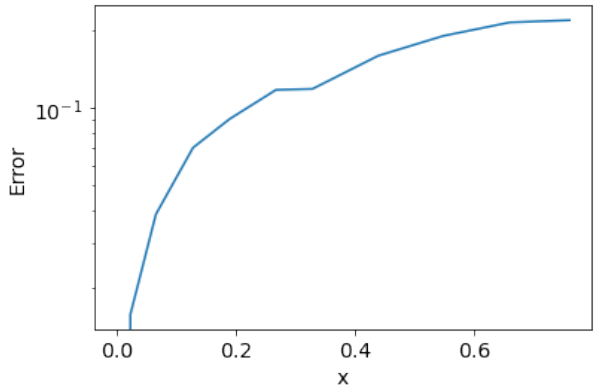


Figure 4.6: Plot of error between exact and ANN results for $\alpha = 0.85$.

Example 2 Here we consider fractional order differential equation given by;

$$D^{\alpha}y + y = x^2 + 2x^{\frac{3}{2}}, x \in [0, 1]$$

with initial conditions $y(0) = 0$ and $0 < \alpha \leq 1$ find exact solution and ANN trail solutions.

Solution: homogeneous solution is Applying equation 3.4 then

$$y_h(x) = ce^{-I^{(\frac{1}{2})}(1)} = ce^{-\sqrt{x}}$$

Particular solution is

Let

$$y_p(x) = Ax^2 + Bx + C$$

then

$$\begin{aligned} y_p^{(\frac{1}{2})} &= D^{\frac{1}{2}}(Ax^2) + D^{\frac{1}{2}}(bx) + D^{\frac{1}{2}}(C) \\ &= 2Ax^{\frac{3}{2}} + Bx^{\frac{1}{2}} \end{aligned}$$

So,

$$y_p = x^2$$

Thus

$$y(x) = y_h + y_p = ce^{-\sqrt{x}} + x^2, y(0) = 0$$

then,

$$y(x) = x^2.$$

In this case,the ANN trail solution is represented as

$$\begin{aligned} y_t(x,p) &= y_0 + (x - x_0)N(x,p) \\ &= 0 + (x - 0)N(x,p) \\ &= xN(x,p) \end{aligned}$$

In this example, the network has been trained with 10000 epochs in the domain from $x = 0$ to $x = 1$ for computing the results. The network has been trained here for 10 equidistant points in $[0, 1]$ and 7 hidden nodes.

Table 4.2: Exact and ANN results for $\alpha = 0.5$

x	exact solution	ANN solution
0	0	0
0.1000	0.0100	0.0130
0.2000	0.0400	0.0405
0.3000	0.0900	0.0903
0.4000	0.1600	0.1608
0.5000	0.2500	0.2534
0.6000	0.3600	0.3513
0.7000	0.4900	0.4889
0.8000	0.6400	0.6469
0.9000	0.8100	0.8270
1.0000	1.0000	0.9926

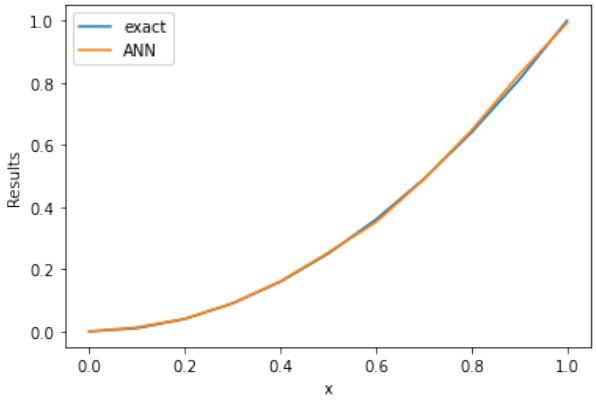


Figure 4.7: Plot of exact and ANN results.

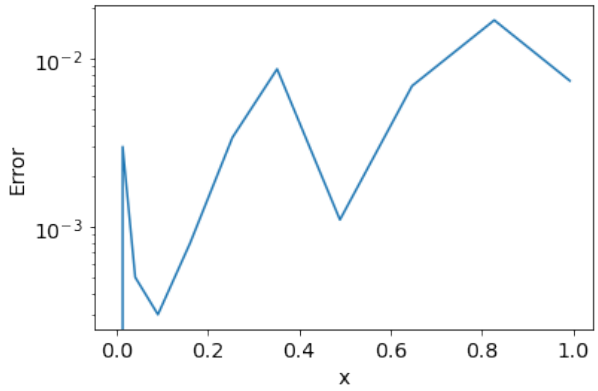


Figure 4.8: Plot of error between exact and ANN results

4.3 Discussions

In this thesis we used artificial neural network methods to solve fractional order differential equations. First we find the analytic solution of fractional order differential equation using undetermined coefficient method and and Bernoulli method. Second find the approximate solution of artificial neural network method. A feed-forward neural networks model and an unsupervised version of error back propagation algorithm is used to minimize the error function and to update the network parameters without optimization techniques. The weights from the input layer to the hidden layer and from the hidden layer to the output layer initialized randomly and tangent hyperbolic activation functions considered as from the input layer to the output layer. Then we compare the solution obtained by using artificial neural network method with exact solution graphically.

CONCLUSIONS AND RECOMMENDATIONS

Conclusions

In this study, we saw the definitions of artificial neural network and fractional order differential equations, theorems of conformable fractional derivative,components of artificial neural network and the algorithm of artificial neural neural network. A feed-forward neural networks model and an unsupervised version of error back propagation algorithm is used to minimize the error function and to update the network parameters without optimization techniques. The weights from the input to the output layer initialized randomly and tangent hyperbolic activation functions(tanh) considered from the input layer to the output layer.

Recommendations

Here, this thesis studied on artificial neural networks. Artificial neural networks is very interesting methods and easy to find approximate solution of fractional order differential equation. This proposed method efficient to solve fractional order differential equation. Excellent agreement between the approximated ANN solution and exact solution. Future researchers should try compare with other numerical methods.

References

Das, S. (2009). Analytical solution of a fractional diffusion equation by variational iteration method. *Computers & Mathematics with Applications*, *57*(3), 483–487.

Diethelm, K. (2010). *The analysis of fractional differential equations: An application-oriented exposition using differential operators of caputo type*. Springer Science & Business Media.

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

Gallo, C. (2015). Artificial neural networks tutorial. In *Encyclopedia of information science and technology, third edition* (pp. 6369–6378). IGI Global.

Geremu, A. (2021). *Solving second order non-linear ordinary differential equation using artificial neural networks.* (Unpublished doctoral dissertation). ASTU.

Graupe, D. (2013). *Principles of artificial neural networks* (Vol. 7). World Scientific.

Ismael, A. M., & Şengür, A. (2021). Deep learning approaches for covid-19 detection based on chest x-ray images. *Expert Systems with Applications*, *164*, 114054.

Jafari, H., Yousefi, S., Firoozjaee, M., Momani, S., & Khalique, C. M. (2011). Application of legendre wavelets for solving fractional differential equations. *Computers & Mathematics with Applications*, *62*(3), 1038–1045.

Jafarian, A., Rostami, F., Golmankhaneh, A. K., & Baleanu, D. (2017). Using anns approach for solving fractional order volterra integro-differential equations.

Jagannath, J., Polosky, N., O'Connor, D., Theagarajan, L. N., Sheaffer, B., Foulke, S., & Varshney, P. K. (2018). Artificial neural network based automatic modulation classification over a software defined radio testbed. In *2018 ieee international conference on communications (icc)* (pp. 1–6).

Kareem, A. M. (2017). Conformable fractional derivatives and it is applications for solving fractional differential equations. *IOSR J. Math*, *13*, 81–87.

Khalil, R., Al Horani, M., Yousef, A., & Sababheh, M. (2014). A new definition of fractional derivative. *Journal of computational and applied mathematics*, *264*, 65–70.

Laroche, E., & Knittel, D. (2005). An improved linear fractional model for robustness analysis of a winding system. *Control Engineering Practice*, *13*(5), 659–666.

Mahmood, S. S., Hamad, K., Hamad, S., & Omar, D. (2021). Bernoulli and riccati fractional differential equations can be solved analytically by using conformable derivatives. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, *12*(13), 7158–7165.

Mall, S., & Chakraverty, S. (2014). Chebyshev neural network based model for solving lane–emden type equations. *Applied Mathematics and Computation*, *247*, 100–114.

Mall, S., & Chakraverty, S. (2015). Numerical solution of nonlinear singular initial value problems of emden–fowler type using chebyshev neural network method. *Neurocomputing*, *149*, 975–982.

Mall, S., & Chakraverty, S. (2016a). Application of legendre neural network for solving ordinary differential equations. *Applied Soft Computing*, *43*, 347–356.

Mall, S., & Chakraverty, S. (2016b). Hermite functional link neural network for solving the van der pol–duffing oscillator equation. *Neural computation*, *28*(8), 1574–1598.

Mall, S., & Chakraverty, S. (2018). Artificial neural network approach for solving fractional order initial value problems. *arXiv preprint arXiv:1810.04992*.

McClelland, J. L., Rumelhart, D. E., & Hinton, G. E. (1986). The appeal of parallel distributed processing. *MIT Press, Cambridge MA*, 3–44.

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

Mehrotra, K., Mohan, C. K., & Ranka, S. (1997). *Elements of artificial neural networks*. MIT press.

Momani, S., & Odibat, Z. (2007). Numerical approach to differential equations of fractional order. *Journal of Computational and Applied Mathematics*, *207*(1), 96–110.

Parisi, D. R., Mariani, M. C., & Laborde, M. A. (2003). Solving differential equations with unsupervised neural networks. *Chemical Engineering and Processing: Process Intensification*, *42*(8-9), 715–721.

Qu, H., & Liu, X. (2015). A numerical method for solving fractional differential equations by using neural network. *Advances in Mathematical Physics*, *2015*.

Raja, M. A. Z., Manzar, M. A., & Samar, R. (2015). An efficient computational intelligence approach for solving fractional order riccati equations using ann and sqp. *Applied Mathematical Modelling*, *39*(10-11), 3075–3093.

Rojas, R. (2013). *Neural networks: a systematic introduction*. Springer Science &

Business Media.

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

Wu, G.-c., & Lee, E. (2010). Fractional variational iteration method and its application. *Physics Letters A*, *374*(25), 2506–2509.

Zurada, J. (1992). Introduction to artificial neural systems, west pub. *Co., St. Paul, MN*.