Adaptive Variational Neural Network for Singularly Peturbed Problems with Continuous and Discontinuous Source term.

A dissertation submitted in partial fulfillment

of the requirements for the degree

of

MASTER OF SCIENCE

in

APPLIED STATISTICS AND DATA ANALYTICS

by

Yashwant Popat Jadhav

CB.PS.P2ASD22012

DEPARTMENT OF MATHEMATICS

AMRITA SCHOOL OF PHYSICAL SCIENCES

AMRITA VISHWA VIDYAPEETHAM

COIMBATORE - 641 112 (INDIA)

JUNE - 2024

AMRITA SCHOOL OF PHYSICAL SCIENCES

AMRITA VISHWA VIDYAPEETHAM

COIMBATORE - 641 112

BONAFIDE CERTIFICATE

This is to certify that the dissertation entitled **"Adaptive Variational Singularly Peturbed Problems using Continuous and Discontinuous Source term."** submitted by **Yashwant Popat Jadhav(Roll Number: CB.PS.P2ASD22012)**, for the award of **Degree of Master of Science in Applied Statistics and Data Analytics** is a bonafide record of the work carried out by her under my guidance and supervision at Department of Mathematics, Amrita School of Physical Sciences, Coimbatore.

Signature of the Project Advisor

Signature of the Project Coordinator

Signature of the Chairperson

Signature of the Internal Examiner

Signature of the External Examiner

AMRITA SCHOOL OF PHYSICAL SCIENCES

AMRITA VISHWA VIDYAPEETHAM

COIMBATORE - 641 112

DEPARTMENT OF MATHEMATICS

DECLARATION

I, **Yashwant Popat Jadhav (Roll Number : CB.PS.P2ASD22012)**, hereby declare that this dissertation entitled **"Adaptive Variational Singularly Peturbed Problems With Continuous and Discontinuous source term."**, is the record of the original work done by me under the guidance of **Dr.Ramesh Babu**, Department of Mathematics, Amrita School of Physical Sciences, Coimbatore. To the best of knowledge this work has not formed the basis for the award of any degree/diploma/associateship/fellowship/or a similar award to any candidate in any University.

Place: Coimbatore **Signature of the Student**

Date:

COUNTERSIGNED

Dr. Ramesh Babu

Project Advisor

Department of Mathematics

Acknowledgement

I am deeply grateful to my project advisor, Dr. Ramesh Babu, Department of Mathematics, Amrita School of Physical Sciences, for his guidance, encouragement, criticism, and faith. He has been a role model for me, with his broad knowledge, inquisitive mind, uncompromising integrity, and enviable ability in carrying out my project works. His dedication and support has inspired and has left an indelible mark in my memory.

I am grateful to The Almighty for providing me this opportunity. I also express my gratitude to Her Holiness, Sri Mata Amritandamayi Devi, The Chancellor of Amrita Vishwa Vidyapeetham for the kind blessings and providing such a wonderful environment.

I take this opportunity to express my sincere thanks to Dr. S. Mahadevan, Principal, Amrita School of Physical Sciences for extending his support in giving me this opportunity.

I express my sincere thanks to Dr. K. Somasundaram, Chairperson, Department of Mathematics, Amrita School of Physical Sciences for his encouragement and providing necessary facilities in the Department.

I especially thank my Class Advisor Dr. Praveen I for the generous support and care throughout my course of study.

I acknowledge my deep sense of gratitude and dedicate this work to my family members and friends who have always been a source of inspiration to me throughout my study. I am so grateful for their patience, love and care they have shown during the period of my project work.

——— (Yashwant Popat Jadhav)

Contents

Abstract

This project introduces an innovation of **Deep Learning** and **Machine learning** technique to effectivly address Singularly Peturebed Problems, especially focusing on **Singularly peturbed convection diffusion problems** that exibhit interior layer along with boundary layers. Singularly Perturbed Problems (SPPs) are a type of problem in which the small parameter ε is multiplied with the highest order derivative term. These problems are common in many fields of applied mathematics and physics. To handle the intricate behaviour of these layer, the proposed approach incorporate the **Finite Element Method**, a well-established numerical technique. Singularly Perturbed Partial Differential Equations are challenging to solve with conventional numerical techniques with machine learning algorithm such as Finite Element Methods due to the presence of boundary and interior layers.

Here, we proposed a new Neural Network model with supervised learning by using the ML and DL techniques and trained by finite element solution. Hence it can be solved any type of singularly peturbed problem. We consider the convection diffusion problem with dirichlet boundary condition and continuous or discontinuous source term. In particular we choose the domain ω=[0,1] and layer fall on it. This model works in three phases. 1)There is an adaptivity in choosing the input values for the NN, σ is the transition point between the domain where half of the inputs values should be choosen from 0 to σ and rest of them from σ to 1. The adaptivity ensures that NN focuses on regions where the solution exhibits rapid changes, leading to improved accuracy and efficiency. 2)Training the neural network by the solution of finite element method (Supervised Learning). 3)It performs the Feed forward and calculate the corresponding output for the particular input and it is considered as a predicted output, after every feed forward it will calculate the loss between the FEM solution as true solution and the predicted solution obtained by feed forward. these predicted solution get updated with every epoch till it attains the minimum loss.

The results show that the NN-based approach achieves good accuracy rate compared to traditional FEM solutions particularly in capturing sharp transitions and interior layers within the solution domain. Additionally, the variational loss function leads to more efficient training process and reduces the number of training points required for convergence.

Keywords: Deep learning, Singularly perturbed problem, Machine learning, Neural Network, Supervised learning, FEM

Chapter 1

Introduction

Differential equations are an equations that describe how a function or system changes in relation to its variables and derivatives. Traditionally, these equations were solved using analytical methods which involved manipulating and integrating the equations to obtain the explicit solutions. However, as problems grew more intricate, numerical methods such as the Finite Difference Method (FDM) and Finite Element Method (FEM) were developed to approximate solutions by discretizing the equations and solving the corresponding system of algebric equations.

Singularly Perturbed Problems (SPPs) are a type of problem in which the parameter ε is multiplied with the highest order derivative term. These problems are common in many fields of applied mathematics, science and enginerring. Among various computational methods to solve SPPs, FEM provide greater adaptability in handling complex geometries compared to other methods. For detailed discussions on analytical and numerical solutions, one may refer to works by R.E.O'Malley [1], Doolan EP et.al [2], Miller JJH et.al [3].

While conventional numerical techniques have been successful in many cases they encounter difficulties when confronted with singularly peturbed differential equations. Singularly Peturbed Differential Equation (SPDE) are frequently encountered in various fields of applied science and engineering, including fluid dynamics, quantum mechanics, chemical reactions, electical networks, elasticity, plasticity, control theory and diffraction theory. The presence of these perturbation parameter leads to abrupt gradients or discontinuties, posing challenges for classical numerical methods to accurately capture the behaviour in the vicinity of these regions. These problems pose challenges when using traditional numerical techniques due to the presence of a peturbation parameter, which leads to the formation of boundary layers.

We apply a novel strategy that uses deep neural networks to solve ordinary differential equations (ODEs) with trainable parameters that are singularly perturbed. There are various benefits to approximating ODE solutions using neural networks: 1) It is possible to evaluate the solution quickly without reducing accuracy or the necessity for Model Order Reduction (MOR); 2) The resultant model is Non-Smooth and not differentiable, which makes it suitable for ODE problem solving.

There has been an increase in interest for solving differential equations using Machine Learning (ML) and deep learning (DL) approaches. The modern world has been completely shifted to the applications of Artificial Intelligence (AI), particularly in the areas of ML and DL. Algorithms that allow computers to learn from data and make predictions or judgments. DL, on the other hand, uses Artificial Neural Networks (ANN) with multiple layers to learn complex representations from data.

Researchers choose ML and DL for singularly perturbed problems because these methods excel in handling complex systems, adapting to non-linearities, and capturing intricate behaviors near perturbation regions. ML and DL algorithms learn patterns and relationships directly from data, eliminating the need for explicit equations or analytical manipulations. This flexibility and adaptability make them well-suited for solving singularly perturbed differential equation.

Here, we consider a convection diffusion problem, such a problems some times unable to study the behaviour of the solution and its derivatives connected to layer phenomenon. Hence,D L and ML methods are proposed to find the solution and plot the graph of such problem, the most important thing here is points choosen are based on σ, where σ is the transition point based on formula $\sigma = 2\varepsilon ln(n)$, which adaptively choose the points between the domain [0,1].

The proposed model of the Neural Network is model-based supervised learning for the solution of ODEs. The model include the part similar to physics informed Neural Network techniques(PINN), it is get trained by the numerical solution given by the FEM of the singularly peturbed problem. The main reason behind taking approximate solution on a collection of sub domains is that it is easier to represent a complicated function as a collection of simple polynomials.

The application of ML and DL to singularly perturbed problems represents a promising advancement, offering more robust and versatile approaches for solving complex differential equations. These methods overcome the limitations of classical numerical techniques and open up new possibilities for progress in various scientific and engineering fields.

Chapter 2

Literature Survey

In this chapter, we shortly go through the work related to singularly peturbed problems which include numerical method as well as ML and DL algorithms.

Sangeeta Yadav et.al [18], discusses the development of the SPDE-ConvNet, a convolutional neural network designed to predict the optimal value of the stabilization parameter for the Streamline Upwind Petrov Galerkin (SUPG) technique in solving Singularly Perturbed Partial Differential Equations (SPPDEs). It discusses the difficulties in solving SPPDEs, disadvantages of the current deep learning-based solvers, and the proposed SPDE-ConvNet as a possible remedy. The study connects the suggested method with cutting-edge variational form-based neural network techniques and provides the mathematical understanding needed to comprehend it. It also describes the SPDE-ConvNet's network architecture and loss function and supplies an example to determine how accurate the system while solving a convection-diffusion problem.

Building on the theme of using neural networks to solve PDEs, the study by Reza Khodayi-mehr et.al [16], presents the VarNet algorithm. VarNet focuses on overcoming the drawbacks of traditional discretization-based numerical techniques, which often struggle with computational efficiency and flexibility. By employing deep neural networks, VarNet offers benefits such as rapid evaluation, smooth differentiability, and ease of parallelization. The innovative loss function based on the variational form of PDEs is a key feature, proving more successful in capturing accurate PDE solutions. The study compares VarNet with other approaches and demonstrates its effectiveness through applications to advection-diffusion problems, addressing training challenges and optimal sampling strategies, and highlighting its efficacy in MOR.

Further exploring neural network applications in solving SPPDEs, Tawfiq LNM et.al [13], investigate ANN for singular perturbation problems (SPP). These methods describe a trial approach that approximates the SPP solution by using an ANN with tunable parameters. Numerical experiments on second-order SPP examples demonstrate the methodology by comparing the obtained results with precise answers. The neural network is trained using a variety of training techniques, including quasi-Newton, levenberg-marquardt, and bayesian regulation. Neural networks have the capacity to solve complicated mathematical problems, as seen by the low error rates and strong agreement with precise solutions that prove the precision and effectiveness of the ANN technique in solving SPP. In the broader context of neural network methods for differential equations. The book "An Introduction to Neural Network Methods for Differential Equations", by Neha Yadav et.al [6], provides a comprehensive overview of the principles and advantages of neural networks over conventional numerical techniques. The book discusses various neural network designs, activation functions, and their historical evolution and applications in fields like signal processing and pattern identification. It also includes practical MATLAB code for solving differential equations using neural networks, reinforcing the robustness, high accuracy, and ongoing research in hybrid approaches and optimization algorithms, which are crucial for advancing the field.

M.W.M.G. Dissanayake et.al [4], discusses a method for solving PDEs using neural networks by transforming the problem into an unconstrained minimization task. This approach simplifies the computational process and provides accurate solutions swiftly. The method's effectiveness is illustrated through examples such as a linear Poisson equation and thermal conduction with non-linear heat generation, showing its suitability for both homogeneous and non-linear problems. These works collectively highlight the significant advancements and potential application of neural networks in solving differential equations. SPDE-ConvNet, VarNet, and various ANN approaches demonstrate unique solutions to specific challenges in this field. They address the limitations of traditional methods and offer robust, accurate, and efficient alternatives, contributing significantly to the advancement of neural network-based methods for differential equation solutions.

The literature analysis also examines numerical solutions for singularly perturbed situations, highlighting the need of parameter robust approaches. It includes time-delayed parabolic reaction-diffusion problems as well as others and looks at other numerical approaches for better accuracy, such as Richardson extrapolation and finite differences on Shishkin meshes. The overview covers asymptotic analysis for coupled systems of reaction-diffusion problems and layer-adapted mesh generation to guarantee uniform convergence. It mentions semi-linear reaction-diffusion issues and points out how little is known about finite difference approaches to them. Furthermore, domain decomposition techniques are highlighted for their potential in managing complicated domains and parallel computing, especially Schwarz domain decomposition and Schwarz waveform relaxation (SWR) approaches. The survey concludes by identifying the application of SWR methods to singularly perturbed time-dependent problems as an underexplored area with significant potential for new research [13, 4].

Chapter 3

Problem Statement

Imagine a river flowing strongly and smoothly. Liquid pollution pours into the water at a certain point. What shape does the pollution stain form on the surface of the river ? Two physical processes operate here: the pollution diffuses slowly through the water, but the dominant mechanism is the swift movement of the river, which rapidly convects the pollution downstream. Convection alone would carry the pollution along a one-dimensional curve on the surface ; diffusion gradually spreads that curve, resulting in a long thin curved wedge shape. When convection and diffusion are both present in a linear differential equation and convection dominates, we have a convection-diffusion problem. In this project, we consider the simplest mathematical model of a convection diffusion problem, that is, a two-point boundary value problem of the form.

$$-\varepsilon u''(x) - a(x)u'(x) + b(x)u(x) = f(x), \quad x \in (0,1) = \Omega \tag{1}$$

$$u(0) = A, \quad u(1) = B. \tag{2}$$

where ε is a small positive parameter, A and B are some constants and a, b and f are some given functions. In particular, f may be continuous or discontinuous function, here the term u'' corresponds to diffusion and its coefficient is small. The term u' represents convection, while u and f play the roles of a reaction and source term respectively. [Spriet and Vansteenkiste] [15] explain why diffusion and convection should be modelled by second order and first-order derivatives respectively.

3.1 The Components and Significance of the Boundary Layer Equation

Singularly Peturbed Parameter(ε): The Diffusion coefficient ε indicates how quickly the scalar quantity diffuses across the system. It affects how evenly and smoothly the quantity is distributed over the domain Ω, rapid diffusion is indicated by higher ε values, and slower diffusion is suggested by lower values.

Convection Term : Convection, describes how a fluid flow transports a scalar quantity. The function has vector values, it describes the quantity's direction of movement within the domain Ω, Knowing this term is essential to understanding how fluid movement affects the scalar quantity being studied.

Reaction Term :A reaction term, is used to indicate how quickly a scalar quantity interacts with other species in the system. It measures how the amount behaves and changes inside domain Ω. Studying the chemical or biological processes connected to the scalar amount requires an analysis of this word.

Forcing Function $f(x)$:The forcing function $'f'$ takes into account any external sources or sinks that impact the scalar quantity inside the domain Ω. It indicates the existence of external variables that affect the system's general dynamics, such as outside pressures, inputs, or boundary effects.

3.2 Domain and Boundary Conditions

The domain Ω is specified as the interval (0, 1), which denotes the study's geographic scope. It gives our inquiry context and includes the area where the scalar quantity is investigated. The scalar quantity values at the domain borders are indicated by the boundary conditions $u(0) = u_0 \quad and \quad u(1) = u_1$. These requirements are essential for figuring out how the scalar amount behaves and what its properties are inside the given range.

3.3 Problem Description

A small positive parameter ε specifies the differential equation controlling the singularly perturbed convection-diffusion problem with dirichlet boundary conditions. As ε gets closer to 0, the solutions or their derivatives undergo sudden shifts that are referred to as layers. The convection-diffusion part of the problem introduces convective flow and diffusion effects, which complicates the investigation. The dirichlet boundary conditions define the values of the solution at the boundaries of the domain. This is an intricate topic requiring careful consideration and specific approaches meant to handle solitary perturbations and convection-diffusion processes with Dirichlet boundary conditions.

Understanding the behavior of the scalar variable u—including its diffusion, reaction, and movement through the fluid flow. Researchers can gain a better understanding of complex systems, including their dynamics, emerging patterns, and crucial components, by examining these equations. This information is necessary for creating effective systems and creating plans for managing or controlling them. By comparing the outcomes with actual data, researchers may validate their models and simulations by solving the reaction-convection-diffusion equation. while ensuring the models correctness and predictive power, this validation procedure directs further study and implementation.

Due to complexity of these equations, approximative solutions are frequently obtained by numerical techniques and simulations. These methods are continuously improved by researchers to solve the equations quickly and precisely, allowing for the investigation of several scenarios and system complexity. But, Neural Networks are an effective tool for addressing complicated issues; they are not the only method utilized to solve PDEs like the Reaction Convection-Diffusion equation. To estimate or discover solutions to PDEs, neural networks and other machine learning approaches can also be applied, especially in situations where standard numerical methods could be difficult or costly to compute.

3.4 Properties of Reaction-Convection-Diffusion Problems with Interior Layers

Reaction-convection-diffusion problems with interior layers have unique aspects that require careful study and numerical treatment, including:

The Interior Layers: In some instances, the solution or its derivatives experience sudden shifts inside the domain due to the existence of inner layers. Because of the interaction of the processes of convection, diffusion, and reaction, these layers are distinguished by sudden shifts in the solution profile.

Asymptotic Conduct: Certain singularly perturbed problems with inner layers exhibit sophisticated asymptotic behavior in nearby of the layer areas. Accurate modeling and numerical simulation need an understanding of and characterization of this behavior.

Conditions of Boundaries: The behavior of the solution inside the domain is largely determined by the boundary conditions, particularly in the surrounding area of the inner layers. For the purpose of precisely representing the effects of convection, diffusion, and reaction, boundary conditions must be specified.

Numerical Difficulties: The existence of sudden shifts and quick variations in the solution profile make numerically solving reaction-convection-diffusion problems with inner layers difficult. In order to capture the layer, specialized numerical techniques like layer-adapted grids and adaptive mesh refinement are frequently needed.

Convergence and Stability: It is important to guarantee the convergence and stability of numerical solutions for these issues. In order to obtain solution, thorough study and algorithm adjustment are necessary when dealing with the presence of interior layers, which might impact the convergence qualities of numerical schemes.

Dependency on Parameters: The selection of parameters, including diffusion coefficients, reaction rates, and convection velocities, can significantly impact reaction-convection-diffusion issues involving inner layers. Considerable differences in the solution behavior close to the inner layers might result from small adjustments to these parameters.

It is essential to know these characteristics in order to model, analyze, and solve reaction-convection-diffusion problems with inner layers numerically. To get accurate and trustworthy findings, special attention needs to be made to capture the complex dynamics close to the layers. In order to properly analyze and handle numerically, reaction-convection-diffusion issues involving internal layers must possess many unique characteristics and attributes.

Chapter 4

Methodology

A type of differential equations known as singularly perturbed problems occurs when a minor parameter ε has a considerable impact on the behavior of the solution, frequently leading to boundary or interior layers or abrupt changes in particular areas. These issues are sensitive, therefore traditional numerical approaches might not be able to handle them well. Because they can efficiently manage non-linearities and approximate complicated functions, neural networks (NNs) provide an alternative that is feasible. In this chapter, an overview of the process for applying neural networks to solve singularly perturbed situations is provided below:

4.1 Feedforward Neural Network

In feedforward, without any feedback loops, information moves from the input layer via the hidden layers to the output layer in a feedforward neural network. After processing the input it receives, each layer transfers the changed data to the layer above it. Throughout the training phase, the network learns by modifying the weights and biases connected to each connection in an effort to reduce the discrepancy between the goal values and the anticipated output. It is the most prevalent kind of neural network and is used in many different domains, including signal processing, image processing, control systems, pattern recognition, system modeling and stock market forecasting. The network is made up of linked neurons, each of which uses an activation function to produce an output after receiving weighted input signals.

Three primary layer types make up a feedforward neural networks architecture: input, hidden, and output layers. Below is a synopsis of every layer:

Input Layer: The neural networks input layer is where raw data or features are first introduced into the system. A feature or input variable is represented by each node in the input layer. The dimensionality of the input data determines how many nodes are in the input layer.

Hidden Layers: Between the input and output layers, where the real computing is done, are intermediary levels known as hidden layers. Every node in a hidden layer applies an activation function.It calculates the net value for the node using the weights and bias and it is calculated by using the formula $f(net) = w_i * x_i + b$, and the calculated $f(net)$ value is passes through the activation function and the resultant value is the nodes output value and then forwards the result to the subsequent layer. The number of neurons in each layer and the number of hidden layers is a design decision that might affect the network's capacity of identifying complicated patterns. (wi = weights, xi = input values, bi = bias values).

Output Layer: The output layer generates the neural network's ultimate output, or prediction. The kind of job (e.g., regression, multi-class classification, binary classification) for which the network is intended determines the number of nodes in the output layer. An activation function appropriate for the task—such as sigmoid for binary classification or softmax for multi-class classification—is usually used at the output layer.

Activation Function: Neural networks may discover complex patterns and correlations in the data by introducing non-linearities into the network through the use of activation functions. Neural networks capacity to learn and generalize from input would be severely limited in the absence of activation functions, since they would be reduced to a sequence of linear transformations.

An activation functions job is to decide, in response to input, whether or not a neuron needs to be activated. The neuron is triggered and delivers an output signal to the network's next layer if the input level exceeds a certain threshold. The neuron doesn't fire if the input is below the threshold.

Training: To understand the relationships between the input data and the intended output, FFNN is trained.

Loss Function: In order to reduce the gap between expected and actual outputs, neural networks train by modifying weights and biases. A crucial performance indicator for networks is loss function which measures this variation. The error between the networks predictions and the actual results is measured by the loss function.

Purpose of loss function: To evaluate and optimize the neural network's performance.

1. **Minimization**: Achieved through an algorithm known as back-propagation.
   Equation: The average squared energy, $\varepsilon_{av} = \frac{1}{N} \sum \varepsilon^2$ where,
   N is the number of patterns in the training set,
   ε is the instantaneous error energy for each output neuron.

2. **Back-propagation Algorithm:** One of the popular technique for training feed-forward neural networks is the back propogation. It entails back-propagating a gradient vector defined as the derivative of an error measure with respect to a parameter in order to update the synaptic weights of the network. As a supervised learning rule, this method needs a collection of desired outputs to be trained. By modifying the network's free parameters, such as synaptic weights and bias levels, it seeks to reduce the average error energy. There are two modes of operation for the back-propagation algorithm: batch mode, where updates happen after presenting all training instances, and sequential mode, where weight adjustments happen after each training example.

Weight Updates: Gradient descent or its variants are optimization algorithms that are used to modify the weights and biases of the network. By altering the weights and biases in the opposite direction as the computed gradients, the goal is to minimize the loss function.

Iteration: Until the network converges and reaches acceptable performance on the training set, the forward propagation, backpropagation, and weight update stages are repeatedly carried out iteratively on batches or subsets of the training data. By applying the learnt weights and biases to forward propagation across the network, the trained FFNN may then be used to forecast fresh, unknown data. Using FFNNs to estimate the solutions of singularly Perturbed differential equations has shown to be successful. FFNNs are able to accurately forecast the complicated dynamics of the equations by training the network using labeled data collected from SPDEs. This feature releases scientists and engineers from the need to only use conventional, computationally costly numerical approaches by enabling effective simulations and predictions, particularly for high-dimensional systems. As FFNNs are able to understand the link between input variables and associated outputs, they are essential in approximating solutions to SPDEs. This makes it possible to extend the behavior of SPDEs to unknown input circumstances, which makes it easier to simulate and analyze complicated systems that SPDEs model.

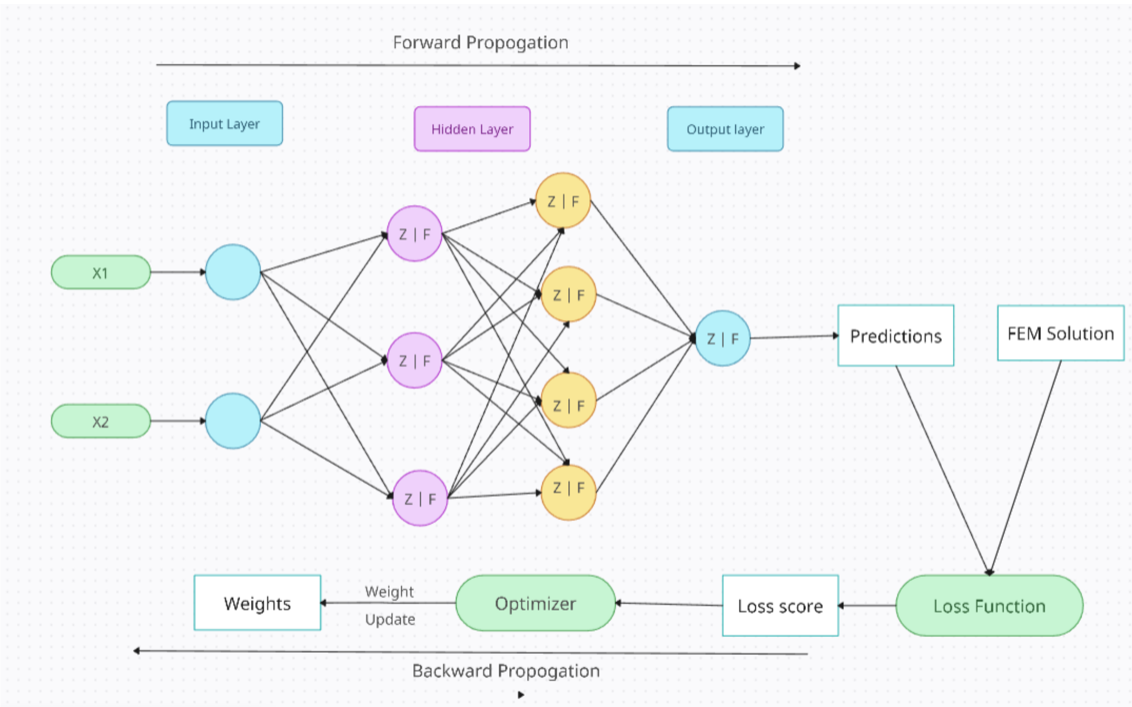


Figure 4.1: Adaptive VarNet Architecture

To perform the accuracy and approximating the solution the predicted solution from NN is iterated till it will not reach the defined minimum at every epoch the predicted solution get updated by calculating the loss and reduce the loss.The true solution is given by the FEM solution .

$$\text{Loss} = \frac{1}{2}\sum(\text{FEM\_solution} - \text{predicted\_solution})^2$$

Optimization of the weights is made by backward propagation of the error during training phase. The model reads the input and output values in the training data set and changes the value of the weighted links to reduce the difference between the predicted and target (FEM) solution. The error in prediction is minimized across many training cycles (iteration or epoch) until network reaches specified level of accuracy or defined minima. The variational loss directly leverages the physical constraints of the ODE, guiding the NN to learn solutions that conform to those physical laws.

4.2 Finite Element Method

Singularly perturbed differential equations (SPDEs) can be solved numerically with the Finite Element Method (FEM), which uses piecewise polynomial functions generated on a grid to approximate the solution. In FEM the domain is divided into smaller subdomains, or elements, and polynomial functions are used to approximate the solution inside each element. Within the literature, the typical Galerkin finite element method is implemented on a nontraditional Shishkin type mesh to solve singularly perturbed problems utilizing the FEM. Approximations of the solution inside each sub-region are obtained using the FEM, and the mesh is created to isolate the boundary layers from other sub-regions.

Building a finite element space with piecewise polynomial functions of a certain degree on each mesh sub-interval is the first step in the FEM process for solving SPDEs. After defining the weak formulation of the SPDEs, the FEM looks for an approximate solution that fulfills the weak formulation in the finite element space.

In order to approximate the solution inside the finite element space, interpolants are also constructed as part of the FEM process. These interpolants are utilized to obtain precise approximations of the solution within each mesh element, provided they meet specific requirements.

Basically, the way the FEM operates is that it discretizes the domain, uses polynomial functions to approximate the solution inside each element, and then combines these local approximations to provide a global approximation of the SPDEs solution. The technique is frequently applied in engineering and computer mathematics to solve a broad range of differential equations, including those that are singularly disturbed. The error calculation is as follows:

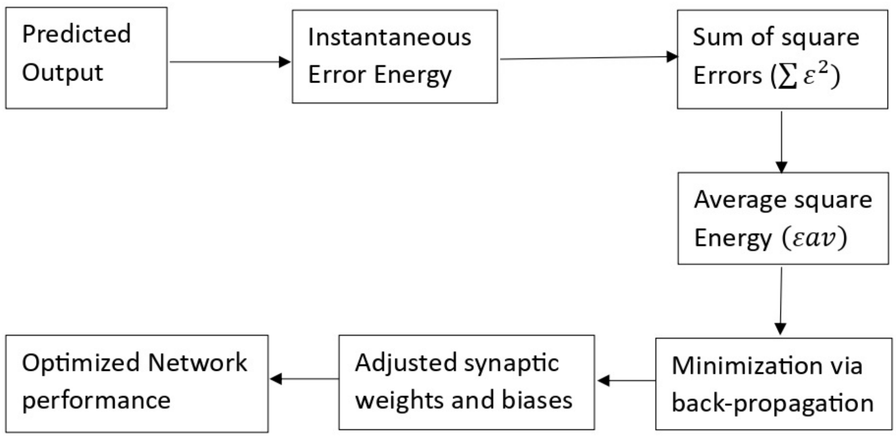


Figure 4.2: Error calculation

4.3 Adaptive Variational Neural Network (VarNet)

The Adaptive Variational Neural Network (VarNet) is a DL and ML technique developed to address Singularly Perturbed Ordinary Differential Equations (SPODEs), which are characterized by rapid solution changes that pose significant challenges to traditional numerical methods. At the core of VarNet is a variational loss function that is derived from the weak form of the governing equations, ensuring that the neural network's solutions are physically meaningful to the underlying principles of the problem. The network's adaptivity is achieved through a dynamic adjustment of input values within the computational domain, which allows the model to concentrate computational efforts on regions where the solution exhibits the most significant variations. This adaptivity, combined with the variational approach, enables VarNet to outperform conventional FEMs in accuracy and efficiency, particularly in capturing complex features such as internal layers and abrupt transitions within the solution domain. VarNet's architecture is based on a feed-forward neural network with multiple hidden layers, utilizing the hyperbolic tangent (tanh) activation function to introduce non-linearity, and it is trained using supervised learning with data from FEM solutions to ensure its predictions align closely with the true behavior of the SPODEs.

In the process of Adaptive VarNet to singularly peturbed problem, we have to make use of FE solution of the corresponding problem.Because of deriving the FEM solution,the derived SPP should be transformed into variational problem,Hence we have stated below the weak formulation of (1) as,
Find $u \in V$ such that

$$a(u, v) = l(u) \quad , \quad \forall \quad v \in V. \tag{3}$$

Where,

$$a(u, v) = \varepsilon < u', v' > - < bu', v > + < cu, v > \quad \text{and} \quad l(v) =< f, v >$$

Where, V is a solution space with piecewise continuous function and $<, >$ is standard inner-product defined on V.

An Adaptive VarNet algorithm has been given as follows,

Algorithm 1: Adaptive VarNet Algorithm

- **Require:** Inputs selected adaptively within the domain [0,1] where σ, which is defined as σ=$2\epsilon \log n$, splits the input data into two halves.
- **Require:** FE method for neural network training In order to train and assess the neural network in the context of resolving the given problem, the FEM approach shows how important it is to have exact and accurate reference data.
- **Require:** MLP-NN layer widths and the quantity of training points the Neural Network's architecture and design to maximize performance while effectively using computing resources
- It will continuously update the loss and iterate the NN until it fails to attain the necessary minimal loss.
- By calculating the loss and comparing it to the FE solution as the actual solution, we want to demonstrate that the NN has received sufficient training and is able to provide results that are superior to those found using the FE approach.

Chapter 5

Application of Adaptive VarNet to Singularly peturbed problems

5.1 Singularly Peturbed Problems

Differential equations with one or more tiny parameters (ε) are singularly disturbed problems. Resulting in solutions like boundary layers or interior layers that alter quickly. These are difficult challenges because it is frequently not possible to adequately represent these quick changes using typical numerical approaches. For instance, let us take a differential equation with a single small parameter:

$$\begin{aligned} -\varepsilon u''(x) - a(x)u'(x) + b(x)u(x) &= f(x), \quad 0 < x < 1, \\ u(0) = 0, \quad u(1) &= 0. \end{aligned} \tag{4}$$

In certain situations, the solution of (4) can show a boundary layer, changing quickly over a limited area. Solutions are found using methods such as matched asymptotic expansions, which include solving for the inner solution (valid within the boundary layer) and the outer solution (valid away from the boundary layer) independently and matching the results.

In more intricate situations with just two little factors, as follow:

$$-\varepsilon_1 u''(x) - \varepsilon_2 u'(x) + b(x)u(x) = f(x), \quad 0 < x < 1,$$
$$u(0) = 0,\ u(1) = 0.$$

where ε_1 and ε_2 are small positive parameters, the problem involves multiple scales and possibly overlapping bounndary layers. The solutions may show interactions between many levels or numerous boundary layers. Because of the fast changes, solving these issues sometimes necessitates the use of specialized numerical techniques like adaptive mesh refinement. Alternatively, the complex nature of singularly perturbed issues may be handled with flexibility and efficiency by using Physics-Informed Neural Networks (PINNs), which directly include the differential equations into the training process.

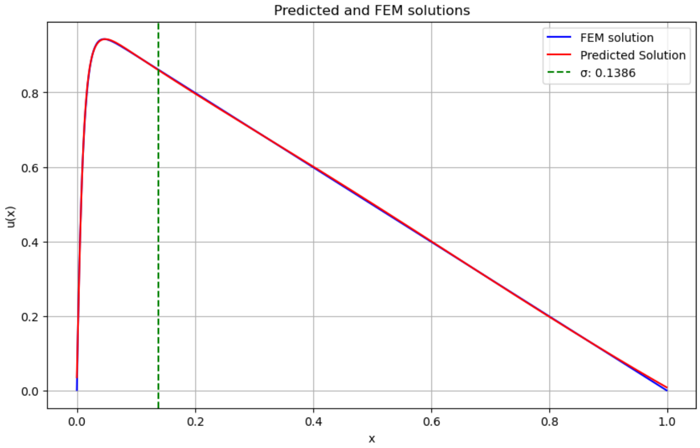
5.1.1 Convection-Diffusion (Smooth data)

$$-\varepsilon u''(x) - a(x)u'(x) + b(x)u(x) = f(x), \quad x \in (0,1),$$
$$u(0) = 0,\ u(1) = 0.$$

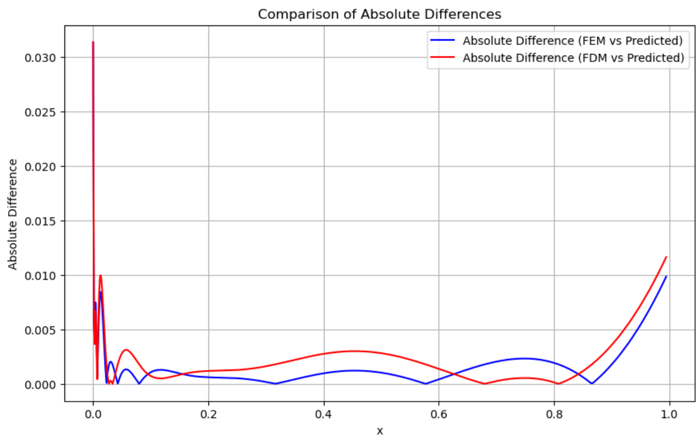
Let us consider, $\varepsilon = 0.001$, $a(x) = 1$, $b(x) = 0$, $f(x) = 1$, min loss $= 1 \times 10^{-5}$, Activation Function = tanh.

$$-0.001\, u''(x) - u'(x) = 1, \quad 0 < x < 1 \tag{5.1.1}$$

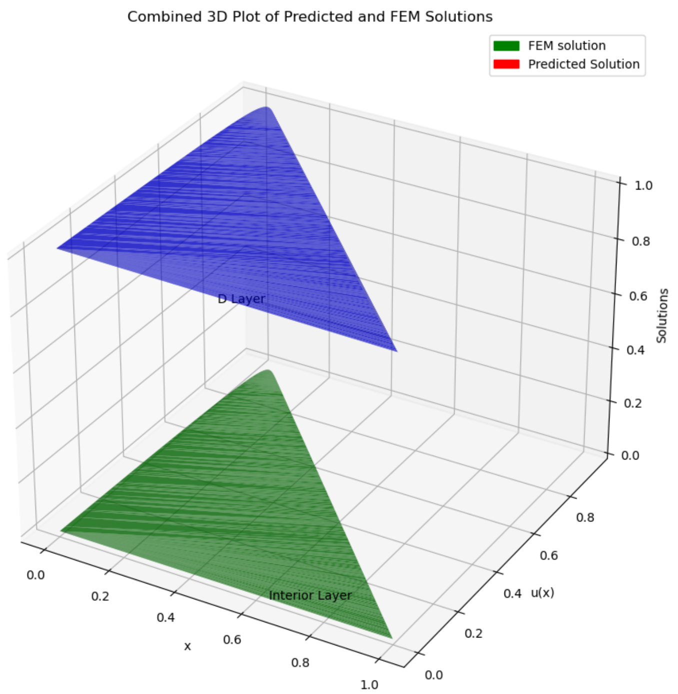
$\varepsilon u''$: Represents the second derivative of u multiplied by the constant ε.
This term introduces a damping or diffusion effect, where the magnitude of the second derivative affects the overall behavior of the function.
$b(x)$: Represents the first derivative of u . This term introduces a linear contribution to the equation, which can represent a restoring force, a growth rate, or any other physical or mathematical influence depending on the specific context.
$f(x)$: represents a forcing function or source term.



(a) Solution profile obtained from Adaptive VarNet and FEM.



(b) Error graph obtained from predicted Vs FEM and predicted Vs FDM.



(c) Comparison of predicted and FEM solution via 3D plot

Figure 5.1: Various diagrams of Convection-Diffusion problem with smooth coefficient (5.1.1).

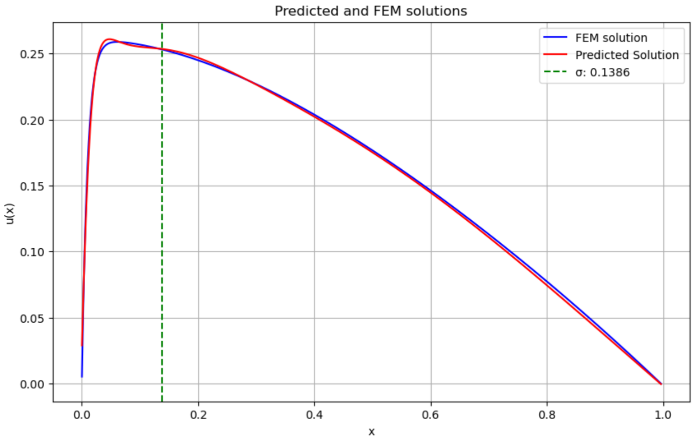
5.1.2 Reaction-Convection-Diffusion (Smooth data)

A model that taken into account convectional effects in addition to diffusion and chemical reactions is known as the reaction-convection-diffusion problem. The movement of materials as a result of fluid flow, such as in a river or the atmosphere, is referred to as convection. This problem describes the transportation of substances not only via diffusion but also by bulk movement within the fluid in which they are immersed and their mutual interactions. It is particularly helpful for researching situations like environmental contamination, heat transmission, and chemical reactions in flowing systems, where fluid movement has a substantial impact on the distribution and reactivity of chemicals. This phenomenon modelled as below:

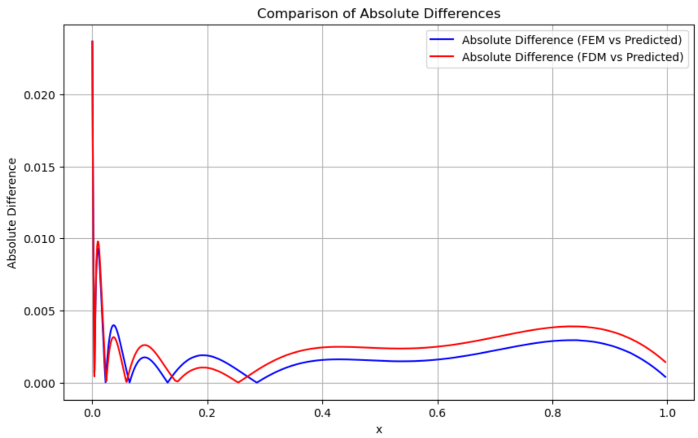
$$-\varepsilon u''(x) - a(x)u'(x) + b(x)u(x) = f(x), \quad x \in (0,1)$$
$$u(0) = 0, u(1) = 0.$$

Let us consider , $\varepsilon = 0.001$, $a(x) = 1 + x$, $b(x) = x^2$, $f(x) = sin(x)$,
min loss $= 1 \times 10^{-5}$, Activation Function = tanh.

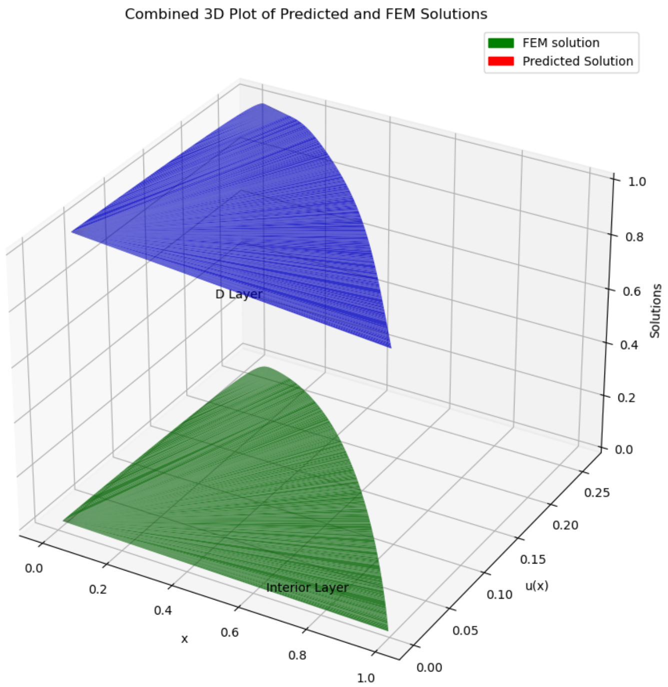
$$-0.001\, u''(x) - (1 + x)u'(x) + x^2 u(x) = sin(x), \quad 0 < x < 1 \tag{5.1.2}$$



(a) Solution profile obtained from Adaptive VarNet and FEM.



(b) Error graph obtained from predicted vs FEM and predicted vs FDM.



(c) Comparison of predicted and FEM solution via 3D plot.

Figure 5.2: Various diagrams of Reaction-Convection-diffusion problem with smooth coefficient (5.1.2).

5.1.3 Convection-Diffusion (Non-smooth data)

$$\begin{aligned} -\varepsilon u''(x) - a(x)u'(x) + b(x)u(x) &= f(x) \quad (x \in [0,1]) \\ u(0) = 0, u(1) = 0. \end{aligned}$$

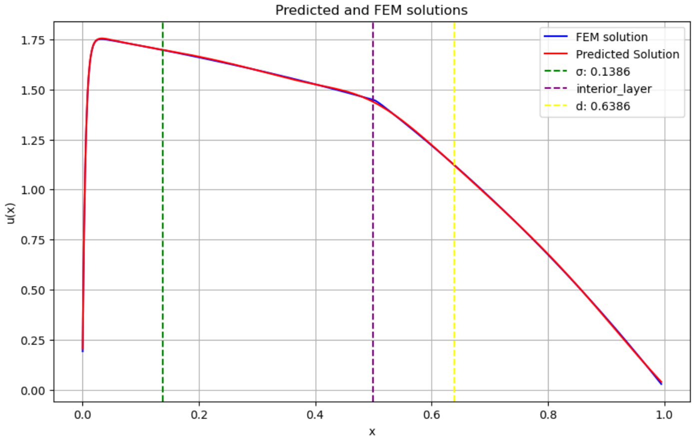
Let us consider $\varepsilon = 0.001$, $a(x) = 2$, $b(x) = 0$,

$$f(x) = \begin{cases} e^x, & 0 < x < 0.5, \\ 2 + e^x, & x = 0.5, \\ e^{x+1}, & 0.5 < x \leq 1. \end{cases}$$

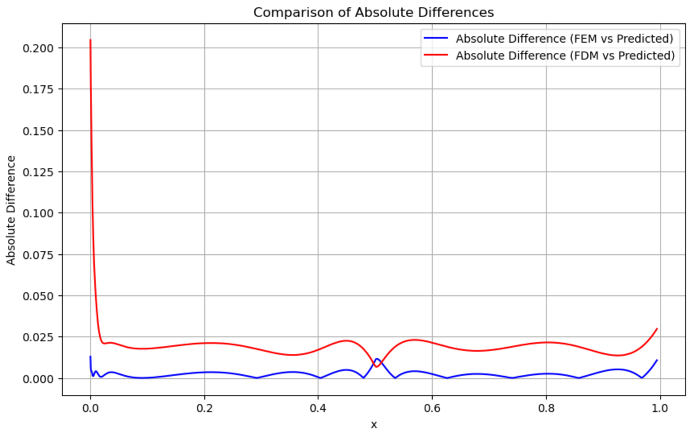
min loss $= 1 \times 10^{-5}$, Activation Function $=$ tanh

$$-0.001u''(x) - 2u'(x) = \begin{cases} e^x, & 0 < x < 0.5, \\ 2 + e^x, & x = 0.5, \\ e^{x+1}, & 0.5 < x \leq 1. \end{cases} \tag{5.1.3}$$

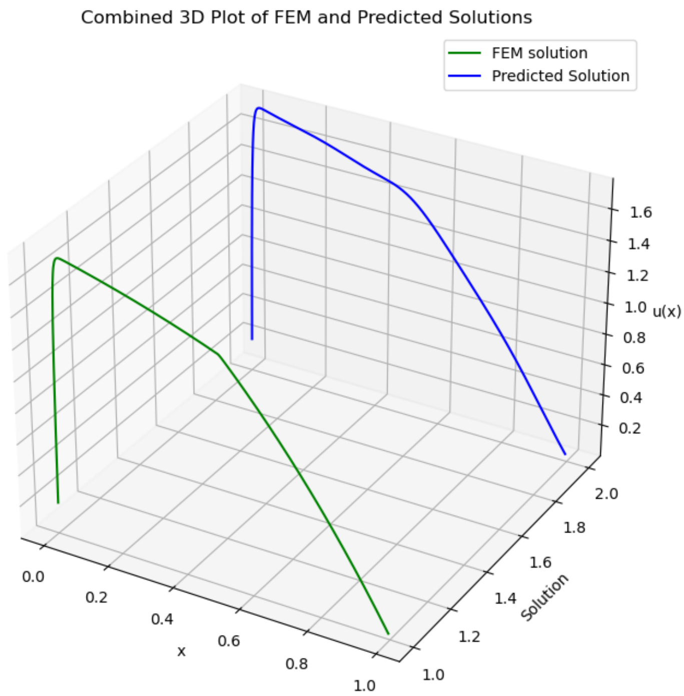
Here, the source term f(x) isdiscontinuous function and a piecewise continuous function that changes its form at a specific point within the domain, in this case, at x = 0.5. This discontinuity in the source term represents a sudden change in the physical process being modeled by the equation. For example, it could represent a change in the properties of a material at a certain point or the introduction of an external influence that alters the system's behavior.



(a) Solution profile obtained from Adaptive VarNet and FEM.



(b) Error graph obtained from predicted vs FEm and predicted vs FDM.



(c) Comparison of predicted and FEM solution via 3D plot.

Figure 5.3: Various diagrams of Convection-diffusion problem (5.1.3).

5.1.4 Reaction-Convection-Diffusion (Non-smooth data)

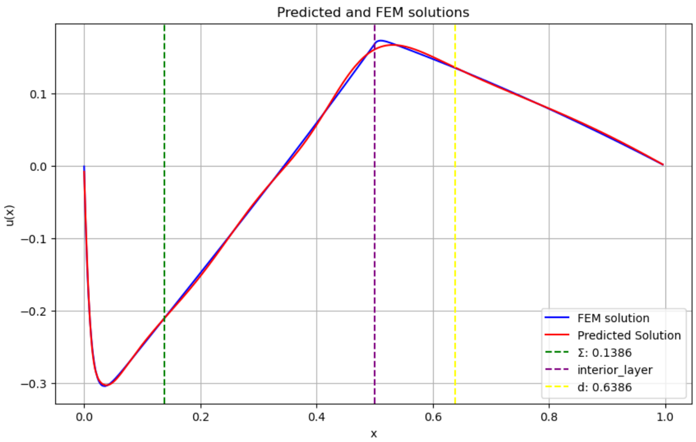
$$
\begin{aligned}
&-\varepsilon u''(x) - a(x)u'(x) + b(x)u(x) = f(x), \quad x \in (0,1) \\
&u(0) = 0, u(1) = 0.
\end{aligned}
$$

Let us consider $\varepsilon = 0.001$, $a(x) = 1 + x$, $b(x) = x^2$,

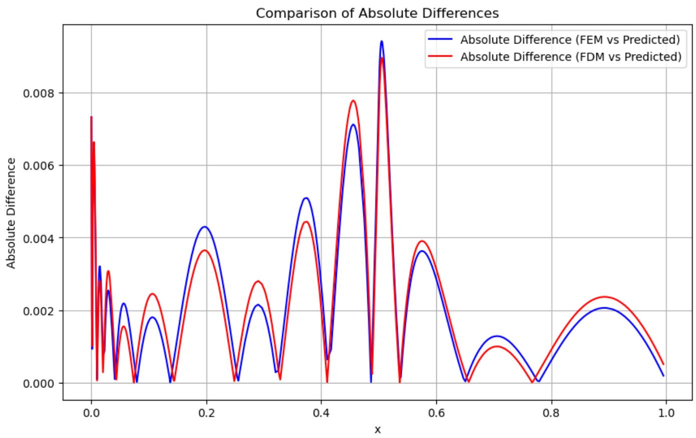
$$
f(x) = \begin{cases} -e^x, & 0 \le x \le 0.5, \\ \sin(x), & 0.5 < x \le 1. \end{cases}
$$

min loss $= 1 \times 10^{-5}$, Activation Function $=$ tanh

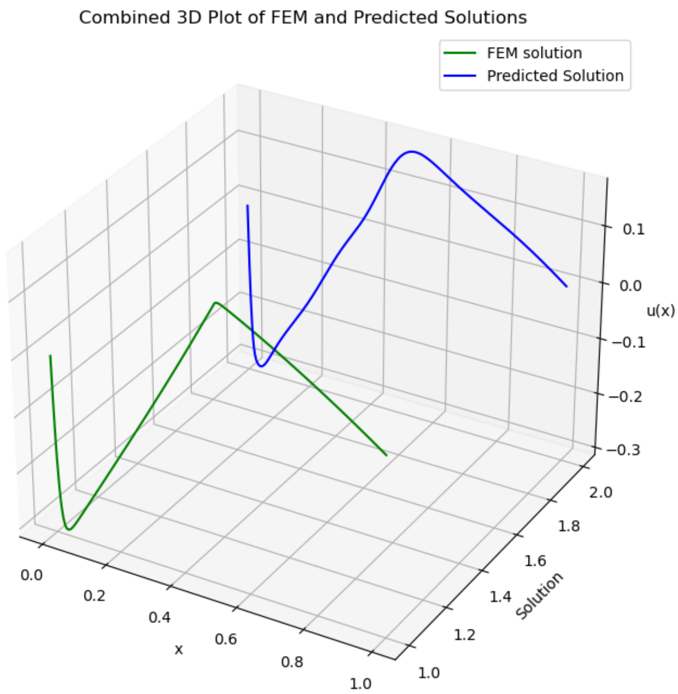
$$
-0.001\, u''(x) - (1+x)u'(x) + x^2 u(x) = \begin{cases} -e^x, & 0 \le x \le 0.5, \\ \sin(x), & 0.5 < x \le 1. \end{cases} \tag{5.1.4}
$$



(a) Solution profile obtained from Adaptive VarNet and FEM.



(b) Error graph obtained from predicted vs FEm and predicted vs FDM.



(c) Comparison of predicted and FEM solution via 3D plot.

Figure 5.4: various diagrams of Reaction-Convecction-diffusion problem (5.1.4).

Chapter 6

Results

While applying Adaptive VarNet to convection diffusion problem. We arrived some interesting outputs/ results.

In Subsection 5.1.1, Figure 5.1 (a), we can observe the solution obtained by the model for the reaction-diffusion problem which approximate to the exact solution obtained by the Finite element method. Clearly, we can understand that the supervised model build using the physics informed technique worked properly. The solution satisfies the boundary condition and gives the valuable solution that of exact solution.It tooks 376 epochs for tanh activation function and 1112 for sigmoid activation function to reach the minimum loss of 1×10^{-5},which shows the tanh activation functions gives the good results than sigmoid activation function.

The Figure 5.1 (b), clearly indicates the advantage of FEM vs predicted and shows least error compare to FDM vs predicted.

In Figure 5.1 (c), the 3D plot provides a visual representation of the solution to the convection diffusion problem in three dimensions. The solution varies across the spatial domain and this plot helps in analyzing the accuracy of the model by comparing the predicted solution with the exact solution. It allows for a clear visualization of how well the model satisfies the boundary conditions and the overall behavior of the solution across different points.

In Subsection 5.1.2, There are different approches to solve the reaction-convection-diffusion equations. Finite element and finite difference method are the numerical methods used to solve these equations when the equation fails to hold the analytical solution.

From Figure 5.2 (a), the solution is then plotted to visualize its variational over the spatial domain. The NN solution is updated or evaluated by comparing the its solution as predicted solution with the true solution which is obtained by the FE numerical method. To hit the minimum loss of 1×10^{-5} it tooks 213 and 1229 epochs for tanh and sigmoid activation function respectively.

In Figure 5.2 (b), clearly indicates the advantage of FEM vs predicted and shows least error compared to FDM vs predicted.

From Figure 5.2 (c), this plot helps in analyzing the accuracy of the model by comparing the predicted solution with the exact solution. They allow for the visualization of complex data structures and relationships between variables that are not easily represented in two dimensions. 3D solution give the full 3D view of the solution which shows the better approximation obtained by the model compared to the FEM solution.

In Subsection 5.1.3, the discontinuous source term is presented in the right hand side. The jump at x = 0.5 indicates a discontinuity, where the value of the source term changes abruptly from e^x to e^{x+1}, with $2 + e^x$ possibly representing a different behavior at the point of discontinuity itself. Solving such SPODEs with discontinuous source terms is challenging because the solution must not only capture the internal layers due to the small parameter ε but also accurately represent the behavior around the discontinuity at x = 0.5. Traditional numerical methods often struggle with these issues, leading to inaccuracies or requiring extensive computational resources to achieve the desired precision.

In Figure 5.3 (a), compares the solutions obtained by the FEM and the NN. It visually demonstrates the NN's ability to approximate the FEM solution, which is considered the benchmark for accuracy. The neural network's performance was evaluated using different activation functions to reach a minimum error of 1×10^{-5}. The results indicated that the hyperbolic tangent (tanh) activation function required 1774 epochs to achieve the desired level of accuracy,while the sigmoid activation function required 4389 epochs. This comparison suggests that the tanh activation function is more efficient in this context as it reaches the minimum error threshold in fewer iterations.

From Figure 5.3 (b), clearly indicates that the VarNet's solution has a smaller error compared to the FDM when both are contrasted against the predicted solution, particularly in regions with sharp gradients or discontinuities.

Figure 5.3 (c), represent both solutions offers a comprehensive view of the NN performance across the entire domain. It illustrates how the NN handles the discontinuous source term and captures the solution's behavior, including the transition through the discontinuity at x = 0.5.

In Subsection 5.1.4, Figure 5.4 (a), the solution has a weak interior layer which is formed inside the domain. Two approaches are worked on this equation: A traditional numerical method and a machine learning-based approach using a NN. It tooks 1457 and 3569 epoch are taken to reach the minimum loss of 1×10^{-5}.

In 5.4 (b) shows that the solution obtain by FEM gives the less error compared to the solution obtained by FDM .

The table given below shows the epoches taken by different activation functions to reach the 1×10^{-5} minimum loss.

Example	Activation Function	Number of iterations / epochs
5.1.1	tanh	376
	sigmoid	1112
5.1.2	tanh	213
	sigmoid	1229
5.1.3	tanh	1457
	sigmoid	3569
5.1.4	tanh	1774
	sigmoid	4389

Table 6.1: For various Activation function,number of iteration to attain the convergence on $\|u^{FEM} - u^{NN}\| < \delta = 1 \times 10^{-5}$.

Chapter 7

Conclusion and Future works

7.1 Conclusion

In this project, we explore the novel technique of the deep learning and machine learning methods to solve the singularly peturbed problems.

Many techniques of numerical method are used for solving the problems that involve complex discritization to handle the solution near the boundaries or interior layers. However we have to investigate an alternate method using the deep learning and machine learning techniques called adpative variational neural network technique, which approximate the solution with required accuracy. The method demonstrates remarkable accuracy and efficiency in capturing complex phenomena such as sharp transitions and interior layers within the solution domain.

This approach offers benefits in terms of computational efficiency, adaptability to complex geometries, and insights into the underlying physics. Further research and development in this field can enhance our understanding and application of machine learning methods for singularly perturbed problems across diverse scientific and engineering domains.

7.2 Future Works

Solving the sigularly peturbed problems through the Machine learning has been emerged topic of interest in our scientific society. Our future work will rely on applying ML algorithm to the complex problem.

1) Advancement of model : Mostly, we will focus the advancement of the model, the current supervised model will be get upgraded to the unsupervised model, which will be more faster and provide the appropriate and approximate solution better than numerical methods like FEM and FDM. The model will be upgraded in such a way that it will work for the ODEs as well as PDEs.

2) Parameters : Working with more than one small singularly peturbed parameters problems which are multiplyed with first order derivative and highest second order derivatives. In more intricate situations with just two little factors,as follows:

$$
\begin{aligned}
&-\varepsilon_1 u''(x) - \varepsilon_2 u'(x) + b(x)u(x) = f(x), \quad 0 < x < 1, \\
&u(0) = 0, \; u(1) = 0,
\end{aligned}
$$

where ε_1 and ε_2 are small positive parameters, the problem involves multiple scales and possibly overlapping layers. The solutions may show interactions between many levels or numerous boundary layers. Because of the fast changes, solving these issues sometimes necessitates the use of specialized numerical techniques like adaptive transition points. Alternatively, the complex nature of singularly perturbed issues may be handled with flexibility and efficiency by using physics-informed neural networks (PINNs), which directly include the differential equations into the training process.

3) Activation functions : Inventing the new activation function which will properly and perfectly suits the solution of the singularly peturbed problems with less error between the true solution and the predicted solution.

Bibliography

[1] O'malley Robert E : Singular perturbation methods for ordinary differential equations, New York: Springer, Vol. 89, (1991).

[2] E.P. Doolsn, John James Henry Miller: Uniform Numerical Methods for Problems with Initial and Boundary Layers, SIAM Review; Philadelphia ,Society for Industrial and Applied Mathematics, vol.25, (1983).

[3] Miller John JH, Eugene O'riordan, and Grigorii I. Shishkin : Fitted numerical methods for singular perturbation problems: error estimates in the maximum norm for linear problems in one and two dimensions, World scientific, (1996).

[4] Dissanayake M. W. M. G, and Nhan Phan-Thien : "Neural-network-based approximations for solving partial differential equations." communications in Numerical Methods in Engineering, 10(3), 195-201, (1994).

[5] Mehrotra Kishan, Chilukuri K. Mohan, and Sanjay Ranka : Elements of artificial neural networks, MIT press, (1997).

[6] Yadav Neha, Anupam Yadav, and Manoj Kumar : An introduction to neural network methods for differential equations, Vol. 1, Berlin: Springer, (2015).

[7] Li Jichun : "Convergence analysis of finite element methods for singularly perturbed problems," Computers and Mathematics with Applications, 40(6-7), 735-745, (2000).

[8] Seshu P : Textbook of finite element analysis. PHI Learning Pvt. Ltd., (2003).

[9] Sazli Murat H : "A brief review of feed-forward neural networks", Communications Faculty of Sciences University of Ankara Series A2-A3 Physical Sciences and Engineering, 50(1), (2006).

[10] Galushkin Alexander I : Neural networks theory, Springer Science and Business Media, (2007).

[11] Amiraliyev G. M., I. G. Amiraliyeva, and Mustafa Kudu : "A numerical treatment for singularly perturbed differential equations with integral boundary condition", Applied mathematics and computation, 185(1), 574-582, (2007).

[12] Sharma Kapil K, Pratima Rai, and Kailash C. Patidar : "A review on singularly perturbed differential equations with turning points and interior layers", Applied Mathematics and computation, 219(22), 10575-10609, (2013).

[13] Tawfiq L. N. M, and K. M. M. Al-Abrahemee : "Design Neural Network to Solve Singular Pertubation Problems", J. of Applied and Computational Mathematics, 3, 1-5, (2014).

[14] Anila S, T Lalithasree, and A Ramesh Babu : "Energy norm error estimate for singularly perturbed fourth-order differential equation with two parameters", International Journal of Computer Mathematics, 100(3), 681-701, (2023).

[15] Roos Hans-Görg : Robust numerical methods for singularly perturbed differential equations, Verlag Berlin Heidelberg-Springer, (2008).

[16] Khodayi-Mehr Reza, and Michael Zavlanos : "VarNet: Variational neural networks for the solution of partial differential equations", Learning for Dynamics and Control, PMLR, 298-307, (2020).

[17] Whiteley Jonathan : "Finite element methods." A Practical Guide 1 (2014).

[18] Yadav Sangeeta, and Sashikumaar Ganesan : "SPDE-ConvNet: Solve singularly peturbed partial differential equation with deep learning", World Congress in Computational Mechanics and ECCOMAS Congress, Scipedia SL, (2022).

[19] Huang Jianguo, Haoqin Wang, and Tao Zhou : ”An augmented Lagrangian deep learning method for variational problems with essential boundary conditions”, arXiv preprint arXiv:2106.14348 (2021).

[20] Yadav Sangeeta : Data Driven Stabilization Schemes for Singularly Perturbed Differential Equations, Diss. Indian Institute of Science Bangalore–560 012, India, (2023).