Deep Learning based Analytical Fourier Series Solution of a Laplace Equation with Arbitrary Boundary Condition

Asmita Basu, Shuvranil Sanyal, Debajyoti Kumar, Pabitra Mitra, and Somnath Roy
Centre for Computational and Data Sciences
Indian Institute of Technology Kharagpur, India
asmita.basu97,sanyalshuvranil,debajyotikumar,pabitra,mechsom@gmail.com

***Abstract*—An analytical solution to the Laplace equation can be derived using a Fourier series, which is determined by the boundary conditions of the domain. However, if one of the boundary conditions is altered, the solution must be recalculated due to the change in the Fourier coefficients of the solution. Within our framework, we have employed a deep learning model to forecast the Fourier coefficients for a non-specific second-order boundary condition on a single side of the computational domain. The anticipated Fourier coefficients are employed to determine the solution, taking into account the arbitrary first-order boundary condition. One benefit of this approach is that we avoid the need to solve the equation again for the specific second-order boundary condition.**

***Index Terms*—Fourier series, Laplace equation, Artificial Neural Network, boundary conditions**

I. Introduction

Fourier transform can be used to help approximate other functions. Neural networks can also be considered as a function approximation technique or universal function approximation technique.

The Fourier Transform is a fundamental mathematical tool that allows us to analyze signals in the frequency domain. To understand its underlying principles, we must first understand how a time-domain signal can be decomposed into a series of sinusoidal waves. For a continuous signal, this representation can be expressed as the function $f(t)$:

$$f(t) = \frac{a_0}{b} + \sum_{k=1}^{\infty} (a_k cos(2\pi kt) + b_k cos(2\pi kt)) \tag{1}$$

This equation illustrates that any signal can be described as an infinite sum of sinusoids, each with its own amplitude and frequency. The coefficients a_k and b_k play a crucial role in shaping the characteristics of the output signal. To determine these coefficients, we use the Fourier transform, which is a function of frequency. The Fourier transform integral is defined as:

$$X(w) = \int_{-\infty}^{+\infty} x(t) e^{-jwt} dt \tag{2}$$

The result of this transform can be interpreted as a set of coefficients that describe the frequency content of the original signal. This mathematical representation allows us to transition from the time domain to the frequency domain, providing valuable insights into the signal's spectral composition.

Machine learning can utilize some well-defined analytical solutions that have an analytical solution. For instance, the process of converting a categorical variable into a one-hot encoding is straightforward, consistent, and generally follows the same methodology regardless of the amount of integer values in the dataset. Regrettably, the majority of the problems that we are concerned with addressing in the field of machine learning lack analytical solutions.

The study conducted by [1] demonstrated a method for reconstructing the heat transfer coefficient in heat conduction modeling. The paper demonstrates the use of swarming algorithms, which are based on studies of biological phenomena such as bee swarms, ant colonies, bird flocks, and worm groups, to reconstruct the heat transfer coefficient to the continuous boundary condition. Swarm algorithms can do data analysis, enhance industrial processes, and facilitate data-driven decision-making. Numerical computations were performed using the authors' application software, which utilized classical types of swarm algorithms.

[2] created a technique for using machine learning to find heat conduction models that can simulate intricate processes using databases. The acquired models meet the criteria of the CD principle, making them hyperbolic balance laws that can be resolved by standard numerical techniques. One-dimensional heat conduction has been considered. Two fully linked neural networks depict the many freedoms present in this process. Numerous numerical tests validate some of the taught models' properties and benefits. The learned model can attain a very high accuracy using initial data that is different from any training data, according to numerical results. The trained models outperform the training data in terms of performance over an extended period.

[3] establishes the framework for a machine learning method based on Fourier analysis. By combining concepts like regularization, periodic extension, and the usage of hyperbolic crossings, it was able to address engineering applications that were of interest. He proposed a Fourier-based machine learning technique, which can be used as a substitute or addition to neural networks in engineering applications. To use partial sums of the Fourier series as approximations, the

fundamental idea is to extend f into a periodic function. The approach is applied to a few high-dimensional analytical function scenarios, which allows some neural network comparisons. The Machine learning approach diminishes the overhead costs associated with creating the training database and carrying out the learning process.

[?] demonstrated the utilization of neural networks to address the Laplace equation in a two-dimensional geometry. They analyzed a PDE problem that simulates the electric potential within the slit-well nanofluidic device. Their research indicates that even simple fully-connected neural networks can offer good accuracy with memory consumption ratios that are on par with the best finite element solutions. [?] introduced a novel Laplace transform artificial neural network (LTANN) learning technique and demonstrated the usage of LTANN to look for abnormalities in geological structures. From the data gathered from ground penetrating radar research, geological anomalies can be automatically recognized by LTANN networks. In this paper, the analytical equation with boundary conditions is obtained via Fourier series. If the boundary condition changes, the Fourier coefficient changes. If the Fourier coefficient changes, then the analytical solution changes. This becomes a hectic task, and machine learning has been used to address this. A second-order polynomial boundary condition on the right side of the computational domain has been considered. Randomly generated C_0, C_1, and C_2 values have been fed for predicting the values of Fourier coefficient A in an Artificial Neural Network model. Similarly, an Artificial Neural Network model has been employed to generate the values of A independently of each other, which have been used for finding the T function at the x and y coordinate, which has also been randomly generated in the range of 0 and 1. The Problem configuration and solution are presented in Section 2 which states the Analytical Solution. The Methodology is presented in Section 3, which gives brief explanations of the model and the model framework, including the proposed method. In Section 4, the results with diagrams have been shown. Finally, the conclusions are presented in Section 5.

II. Problem configuration and solution

We obtain a solution of Fourier coefficients $A_1, A_2, A_3 ... A_n$ as a function of C_0, C_1 and C_2. We generate the training dataset for a set of randomly uniformly generated C_0, C_1 and C_2. We test for a specific C_0, C_1, C_2 then obtain the actual $A_1, A_2, A_3 ... A_n$ and calculate T(x,y) function. Comparison and prediction of manual data has been done between predicted data of A_n data points. T(x,y) has been calculated based on newly predicted A_n. A neural network has been built to predict A_n.

The training dataset has been generated by solving the given solution:

$$A_n = \frac{2}{bsinh(\frac{n\pi a}{b})} \int_0^b (C_0 + C_1 y + C_1 y^2) sin(\frac{n\pi y}{b})) dy \quad (3)$$

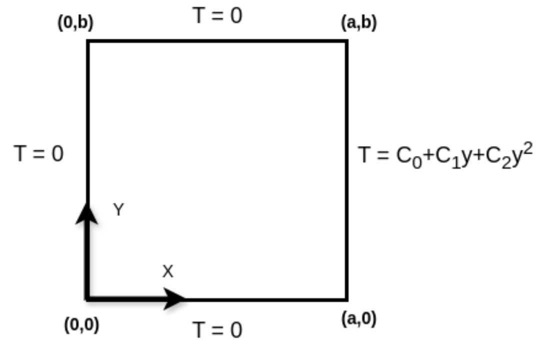

Fig. 1. Domain

Then the equation for T(x,y) at a point of x and y, which is given in a grid system has been calculated:

$$T(x,y) = \sum A_n sinh(\frac{n\pi x}{b}) sin(\frac{n\pi y}{b}) \quad (4)$$

The values of the manually generated function of T(x,y) have been plotted below.

III. Methodology

The proposed method is introduced in detail in Section 3.2. The data-driven technique is implemented in Python programming language. In this research, ‘Sklearn’ and ‘Keras’ packages by ‘Tensorflow’ backend are used for program development.

A. Machine learning algorithm

An Artificial Neural Network (ANN) is an excellent and adaptable model that can effectively capture complex correlations within data by iteratively adjusting weights during the training process. The model’s design and activation functions enable it to accurately capture and analyse non-linear interactions, making it extremely suitable for various applications including image recognition and natural language processing. An artificial neural network (ANN) model replicates the information processing capabilities of biological neural networks seen in the human brain. The system is composed of interconnected clusters of artificial neurons (nodes) arranged in layers. Every neural connection possesses a corresponding weight, which is modified as part of the learning process.

An Artificial Neural Network consists of neurons, also known as nodes. Neurons serve as the primary processing units inside the network since they receive input, undergo a transformation, and generate an output. Layers refer to structured collections of neurons. It consists of an input layer which is responsible for receiving the input data and hidden layers which refer to the intermediate layers of a neural network that are responsible for processing inputs received from the input layer. Finally, the output layer is responsible for generating the final output.

A Neuron Activation in a neural network receives input signals, assigns a weight to each signal, adds them together,

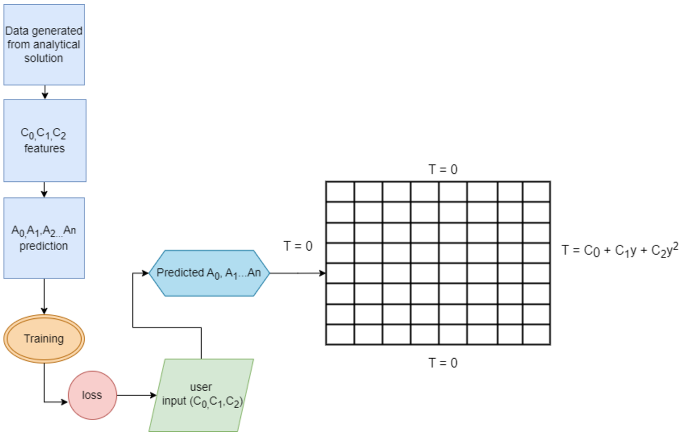

Fig. 2. Flowchart

and then sends the sum through an activation function to generate an output.

$$z_j = \sum_i w_{ij} x_i + b_j \tag{5}$$

where x_i are the inputs to the neuron,w_{ij} are the weights associated with each input, b_j is the bias term and z_j is the weighted sum.

Activation Function is the weighted sum z_i is processed through an activation function f in order to incorporate non-linearity:

$$a_j = f(z_i) \tag{6}$$

Common activation functions include Sigmoid, Hyperbolic Tangent (tanh) and Rectified Linear Unit (ReLU).

For an output neuron k,

$$y_k = f\left(\sum_j w_{jk} a_j + b_k\right) \tag{7}$$

A split of 25 percent testing, 15 percent validation and 60 percent training from the dataset have been taken. C_0, C_1 and C_2 have been used as inputs and A_n as outputs. Two layers have been implemented. The layers consist of 10 and 5 neurons, respectively. “Adam” optimizer has been used and loss functions of MAE and MSE have been calculated after training to check the validity of the model performance.

The training has been done on 60 percent data set from the manually generated data and on 50 epochs.

IV. Result

The predicted values for a set of $A_1, A_2, A_3 ... A_n$ have been generated for 100 data points, where A_n is the Fourier coefficient and n varies from 5, 10, 30 and 50. Subsequently, T(x,y) domain has also been generated along with A_n. Loss functions such as Mean Absolute Error (MAE) and Mean Square Error (MSE) have been calculated to measure the model’s performance. The performance of our model is stagnating when the values of A_n are increased by a huge margin.

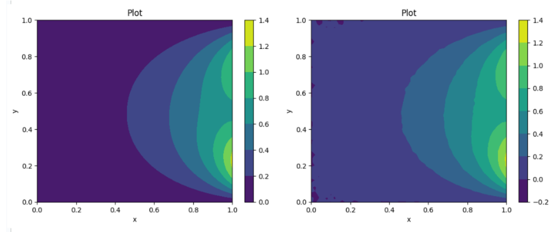

Fig. 3. Graph of Analytical and Predicted Fourier coefficients from A_1 to A_5

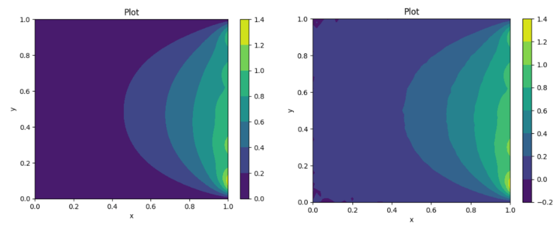

Fig. 4. Graph of Analytical and Predicted Fourier coefficients from A_1 to A_{10}

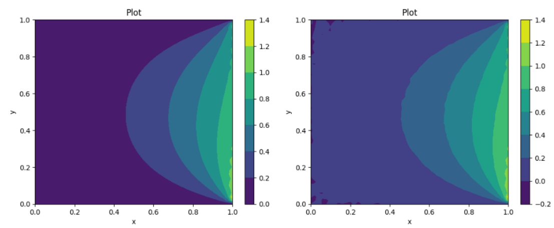

Fig. 5. Graph of Analytical and Predicted Fourier coefficients from A_1 to A_{30}

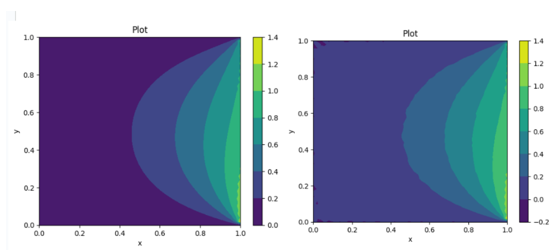

Fig. 6. Graph of Analytical and Predicted Fourier coefficients from A_1 to A_5

$$MAE = \sum_{i=1}^{D} |x_i - y_i| \tag{8}$$

$$MSE = \sum_{i=1}^{D} (x_i - y_i)^2 \tag{9}$$

TABLE I
PERFORMANCE PARAMETERS

Number of A_n	Accuracy	MAE	MSE
5	0.9050	0.0018	6.3710e-06
10	1	0.0017	5.3795e-06
30	1	0.0012	2.9722e-06
50	1	0.0012	2.8649e-06

V. CONCLUSION

In this paper, we predicted the boundary conditions of the domain by using an analytical solution to the Laplace equation derived using a Fourier series. The predictions provide a method to find the boundary values without solving equations for different boundaries with unique domains. The solution of A_n is calculated with randomly generated for a set of inputs C_0, C_1 and C_2, for the boundary conditions C_0 + C_1y + C_2y^2 and predicted independently for $A_1, A_2, A_3...A_n$. Our model gave good performance as assessed by MAE and MSE, but it stagnated for huge values of A_n. Solving for an unknown domain of different geometry, we can analyse the given boundary conditions of a new domain and analyse the solution without manually solving for the data. Predicting data without solving for equations saves time. Our study can be further extended by using polynomial equations of higher degree as boundary conditions.

ACKNOWLEDGMENT

I would like to thank the NSM Nodal Centre project for financial support.

REFERENCES

[1] E. Gawronska, M. Zych, R. Dyja, G. Domek, Using artificial intelligence algorithms to reconstruct the heat transfer coefficient during heat conduction modeling, Scientific Reports 13 (1) (2023) 15343.

[2] J. Zhao, W. Zhao, Z. Ma, W.-A. Yong, B. Dong, Finding models of heat conduction via machine learning, International Journal of Heat and Mass Transfer 185 (2022) 122396.

[3] M. Peigney, A fourier-based machine learning technique with application in engineering, International Journal for Numerical Methods in Engineer- ing 122 (3) (2021) 866–897.

[4] M. Magill, F. Z. Qureshi, H. W. de Haan, Compact neural network solu- tions to laplace's equation in a nanofluidic device (2018).

[5] P. Szymczyk, M. Szymczyk, Supervised learning laplace transform artifi- cial neural networks and using it for automatic classification of geological structure, Neurocomputing 154 (2015) 70–76.