# Application of Machine Learning in Predicting Gaseous Properties of Earth Atmosphere-Midway Progress Report

**Anna Binoy**
School of Physical Sciences
National Institute of Science Education and Research
Odisha
anna.binoy@niser.ac.in

**Sumegha M. T.**
School of Physical Sciences
National Institute of Science Education and Research
Odisha
sumegha.mt@niser.ac.in

## Abstract

The simulation of Earth's atmosphere using gas optics computation is a complex task, often requiring significant computational resources. Machine learning techniques have shown great potential for accelerating these simulations, reducing both the time and resources required. This paper presents a study of the use of machine learning methods to accelerate gas optics computation of Earth's atmosphere. The study compares the performance of traditional gas optics computation methods to a machine learning-based approach, using a range of atmospheric conditions and scenarios. The results demonstrate that machine learning techniques can significantly accelerate the simulation of Earth's atmosphere while maintaining high levels of accuracy. The proposed approach can be used to accelerate atmospheric simulations for a range of applications, including weather forecasting, climate modeling, and satellite imaging. This study provides a valuable contribution to the field of atmospheric science and demonstrates the potential of machine learning for accelerating simulations of complex physical phenomena.

## 1 Introduction

Atmospheric scientists use Radiative Transfer Models (RTMs) to understand how energy from the sun is absorbed, reflected, and scattered by the Earth's atmosphere. RTE+RRTMGP (Rapid Radiative Transfer Model for General Circulation Models, Plus) is a state-of-the-art radiative transfer model used to simulate this energy transfer. This model calculates the absorption and scattering of radiation by the atmosphere by solving a set of Radiative Transfer Equations (RTEs). It is a collaborative project between Columbia University and Atmospheric and Environmental Research , Inc. RTE+RRTMGP can accurately simulate a wide range of atmospheric conditions. Hence, they are useful in modelling the climate of the past, present and the future. The model is also able to accurately simulate the effects of aerosols, such as dust and pollution, on radiation. This has a significance in the study of global warming. This model consists of two parts: the first part, RRTM (Rapid Radiative Transfer Model for General Circulation Models) has been used in climate modelling for decades. The second part which is the General-purpose Graphics Processing Unit (GPGPU) accelerated Radiative Transfer for the Modern Era (RRTMGP) module, is a newer and more computationally efficient version of RRTM. By taking advantage of GPUs, RRTMGP offers higher performance and faster calculations than conventional CPU based models. Despite this, these calculations are computationally expensive, especially when performed over diverse atmospheric profiles. To speed up the calculations, RRTMGP uses precomputed lookup tables (LUTs) that contain the radiative transfer solutions for a range of atmospheric profiles. The LUTs are generated by running the RRTMGP model for a set of representative atmospheric profiles, and then storing the results in a data structure that can be quickly accessed during runtime. However, LUTs have limitatiins, especially in terms of speed and accuracy. LUTs require large amount of storage space to store the pre-calculated radiative transfer solutions for different atmospheric conditions. This can make the use of LUTs computationally expensive, especially when a wide range of atmospheric conditions and input parameters are required. Another limitation of LUTs is that they are only accurate within the range of atmospheric conditions for which they have been pre-calculated. This places the accuracy of LUTs for a wider range of atmospheric conditions under scrutiny.

The main goal of this experiment is to emulate the RRTMGP look up tables using neural networks and investigate the advantages of this utilisation. From the papers that have been mainly referred ([3],[4]), it has been inferred that employment of neural networks has made the calculations much faster and interpolates the data better for a wide range of atmospheric conditions. These results will also be verified as a part of this experiment.

At the midway point of the experiment, a feedforward neural network was developed. Its functionality is verified by training the network with 3600 atmospheric profiles.

## 2 Related works

We followed the first two papers given in Table2.
The first paper discusses the use of machine learning algorithms to improve our understanding of Earth's climate system. The authors highlight several areas where machine learning has been successfully applied, such as predicting extreme weather events and improving climate models. They also discuss the challenges associated with using machine learning in climate research, such as the need for large and diverse datasets and the potential for biases.

The second paper examines the potential of machine learning to improve our understanding of the atmosphere's dynamics and improve weather forecasting. The authors discuss the use of machine learning algorithms to analyze satellite data and identify atmospheric features, such as cloud patterns and wind fields. They also highlight the potential for machine learning to improve the accuracy of weather forecasts, particularly in regions with sparse observational data.

## 3 Baseline Algorithms and Experiment

We used Feedforward Neural Network(FNN) as the baseline algorithm.The code is written in python using the tensorflow package.

### 3.1 Github Link

The code is uploaded here.The data used is uploaded in the dpath folder given in the above link.

### 3.2 Dataset

The step-by-step procedure to generate training, testing and validation dataset and the details are as follows:

1. Download CMIP6 (Coupled Model Intercomparison Project 6) Forcing Datasets (input4MIPs) run by the UColorado from [https://esgf-node.llnl.gov/search/input4mips/](https://esgf-node.llnl.gov/search/input4mips/). It consists of 18 experiments for 100 samples. This dataset offers atmospheric conditions for radiative transfer calculations, such as temperature, pressure, and gas concentration profiles,

| Topic | Year | ML models used |
|---|---|---|
| Predicting atmospheric optical properties for radiative transfer computations using neural networks | 2021 | Feedforward Neural Network (FNN) |
| Accelerating Radiation Computations for Dynamical Models With Targeted Machine Learning and Code Optimization | 2020 | Feedforward Neural Network (FNN) |
| Exploring Pathways to More Accurate Machine Learning Emulation of Atmospheric Radiative Transfer | 2022 | Feedforward Neural Network (FNN) and Recurrent Neural Network (RNN) |
| RadNet 1.0: exploring deep learning architectures for longwave radiative transfer | 2021 | Feedforward Neural Network (FNN) and Convoluted Neural Network (CNN) |

Table 1: Related works and the machine learning models used .

as well as boundary conditions such as total solar irradiance, solar zenith angle, and surface temperature. These inputs are to be provided to the radiation code of each model, with spectrally-integrated fluxes reported at each level.

2. The Github repository contains a script (rfmip_createinput.py) that generates diverse atmospheric profiles for a range of experiments. This code can be used to randomly perturb the pressure, temperature, ozone concentration and water vapour concentration for each profile. In generating pressure level array, setting a minimum pressure value of 1.005183574463, and then looping until the minimum pressure difference between any two adjacent layers is greater than or equal to 1 Pa. Temperature is perturbed by adding a random value between -10 and 10 degrees Celsius to each grid point. The layer temperature and level temperature is perturbed by adding a random value between -5 and 5 degrees Celsius to each layer/level. The concentration of water and ozone in the profile is perturbed by adding a random value between -75% and 75% of its current value to each grid point. This guarantees that the atmospheric profiles created have some degree of accuracy.
3. To produce data sets for training, testing, and validation purposes, the atmospheric profiles that were generated are utilized as input to the RTE+RRTMGP binary file. Prior to this step, the C++ file for RTE+RRTMGP must be compiled.

The binary file for RTE+RRTMGP could not be executed due to an inability to link C++ libraries, despite having been installed on the machine. The generated 3600 atmospheric profiles are uploaded in the Github repository.

### 3.3 Parameters,Hyperparameters

Here we used layer temperature to predict the water vapour concentration.So the parameter is temperature.The hyperparameters are no. of epochs and learning rate.Also we varied the number of hidden layers and nodes in each layer in the FNN .The best hyperparameters that give least MSE is given in Table3.3

### 3.4 Graphs

The epoch vs MSE for training and validating data for various FNN models with different number of layers and nodes are given in Figure1,2,3,4 and 5.

| Model | Epoch | Learning Parameter | Test Loss |
|---|---|---|---|
| 1L_32 | 100 | 0.001 | 36.56846 |
| 1L_64 | 100 | 0.001 | 36.20854 |
| 2L_32_32 | 150 | 0.001 | 1.409719 |
| 2L_64_64 | 50 | 0.001 | 7.306636 |
| 3L_32_64_128 | 100 | 0.001 | 0.236868 |

Table 2: The Feedforward neural network and various hyperparameters

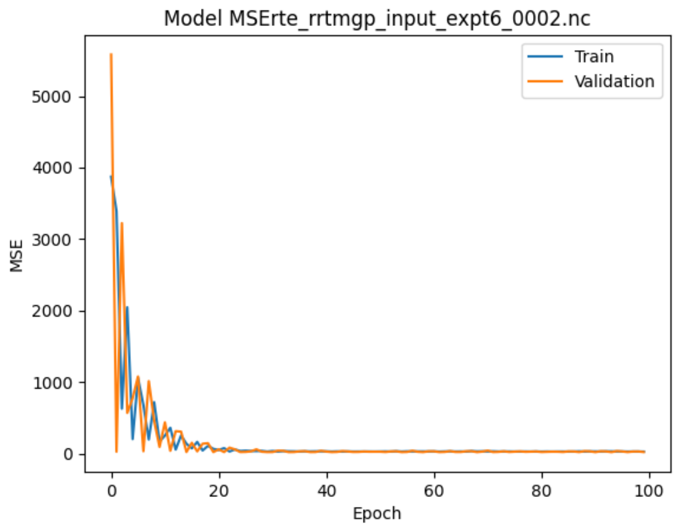

Figure 1: Epoch vs. MSE error for FNN with 1 hidden layer having 32 nodes

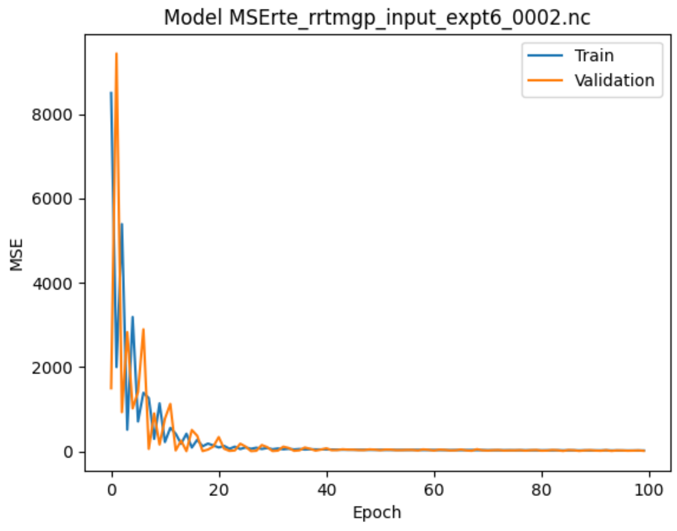

Figure 2: Epoch vs. MSE error for FNN with 1 hidden layer having 64 nodes

## 4 Future Plans

For the atmospheric datasets of this project, a technical issue has arisen after generating 3600 profiles. The plan is to resolve the problem by compiling the original Fortran code ([2]), rather than the C++

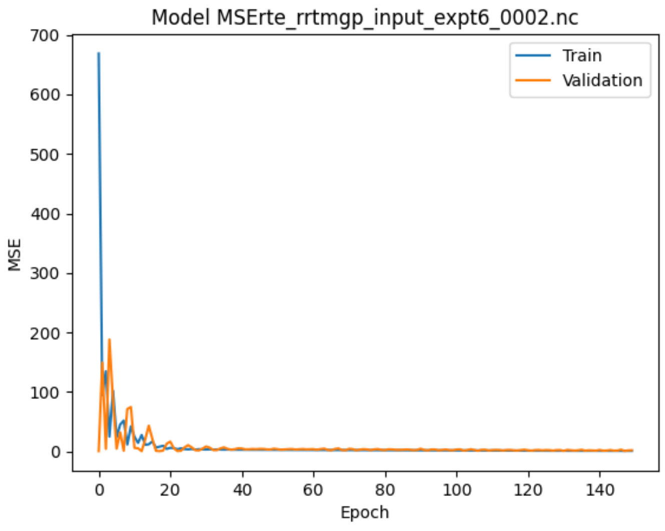

Figure 3: Epoch vs. MSE error for FNN with 2 hidden layer having 32 nodes each

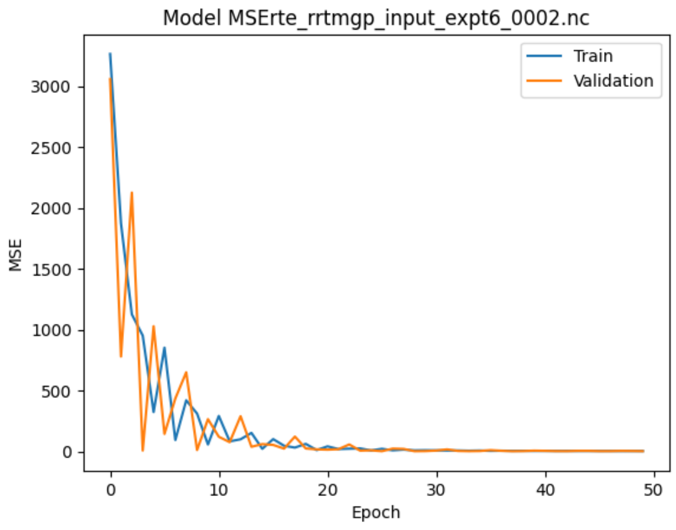

Figure 4: Epoch vs. MSE error for FNN with 2 hidden layer having 64 nodes each

code for RTE+RRTMGP. Another option is to utilize PyRRTM, a Python implementation of the Rapid Radiative Transfer Model (RRTM) available in climt (Climate Modelling and Diagnostics Toolkit) ([1]). PyRRTM interfaces with the Fortran-based RRTM code, allowing for running RRTM in a Python environment.
Once the datasets have been labeled, the plan is to train, validate, and test them on feedforward neural networks. Additionally, more profiles will be generated using other datasets available, such as CAM5, CKDIMP and will be used for training, validating, and testing on a Recurrent Neural Network. The performance of this model will be compared to that of the previous model.

## References

[1] Joy Merwin Monteiro and Rodrigo Caballero. “The climate modelling toolkit”. In: *Proceedings of the 15th Python in Science Conference*. 2016, pp. 69–74.

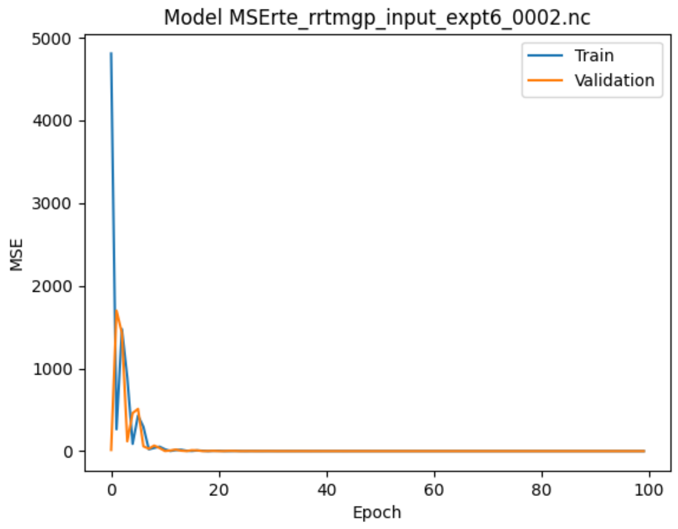

Figure 5: Epoch vs. MSE error for FNN with 3 hidden layer with 32 nodes in the first layer,64 in the second and 128 in the third

[2] Robert Pincus, Eli J Mlawer, and Jennifer S Delamere. “Balancing accuracy, efficiency, and flexibility in radiation calculations for dynamical models”. In: *Journal of Advances in Modeling Earth Systems* 11.10 (2019), pp. 3074–3089.

[3] Peter Ukkonen et al. “Accelerating radiation computations for dynamical models with targeted machine learning and code optimization”. In: *Journal of Advances in Modeling Earth Systems* 12.12 (2020), e2020MS002226.

[4] Menno A Veerman et al. “Predicting atmospheric optical properties for radiative transfer computations using neural networks”. In: *Philosophical Transactions of the Royal Society A* 379.2194 (2021), p. 20200095.