

Predicting global accurate potential energy surfaces with complex topography using a combination of gaussian process regression and neural networks

*A B.Tech. Project Report Submitted
in Partial Fulfillment of the Requirements
for the Degree of*

Bachelor of Technology

by

Vijay Jangal
(190122054)

under the guidance of

Prof. Aditya N. Panda



**DEPARTMENT OF CHEMISTRY
INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI
GUWAHATI - 781039, ASSAM**

CERTIFICATE

*This is to certify that the work contained in this thesis entitled “**Predicting global accurate potential energy surfaces with complex topography using a combination of gaussian process regression and neural networks**” is a bonafide work of **Vijay Jangal (Roll No. 190122054)**, carried out in the Department of Chemistry, Indian Institute of Technology Guwahati under my supervision and that it has not been submitted elsewhere for a degree.*

Supervisor: **Prof. Aditya N. Panda**

Professor,

April, 2023

Department of Chemistry,

Guwahati

Indian Institute of Technology Guwahati, Assam.

Acknowledgements

*I would like to acknowledge the assistance of my supervisor **Prof. Aditya N. Panda** without whom this would not have been possible. I would like to express my heartfelt gratitude for his tremendous support and assistance in the completion of my project. I deeply appreciate his consistent supervision, patience, and insightful guidance.*

I would also like to thank Department of Chemistry, Indian Institute of Technology Guwahati, for providing me with this wonderful opportunity.

Abstract

*This project aims to explore the effectiveness of the combination of the Gaussian process regression (GPR) model and artificial neural network (ANN) model to represent potential energy surfaces (PES) with complex topography. At first, an initial dataset of 2500 points, generated using **MOLPRO**¹ software, was provided. The GPR model was then utilized for active data selection through various steps like interpolation¹¹ of PES. This was done by searching for configurations with the **highest predictive variance**¹¹, which were subsequently passed through an ab initio calculator and added to the dataset. The process continued until the highest predictive variance¹¹ converged, resulting in the final trainable dataset. The training was carried out several times using 100, 200, 400, and 500 data points from the training dataset. A sequential ANN model is constructed using **TensorFlow**² **2.6.0** and **Keras**³ **2.6.0** library based on results and training for each dataset was carried out for multiple values of a number of epochs ranging from 1000 – 10000. The code is written in **Python**¹⁴ **3.10.0**. Mean squared error¹⁵ (mse) is taken as the metric for comparing the accuracy of the model over various epochs and a number of training data. During training, 20% of the training data was selected at random in each epoch and used as test data to verify the model's accuracy. This trained ANN model was then used to generate the PES of the He_2H^+ system by itself and was compared with the calculated results. The model trained using 500 data points over 10000 epochs provided predicted results closest to the calculated values. As predicted PES has a 3D structure that has a quite complex topography, so we took some random datasets of only one atomic distance and potential energy to verify this model.*

Contents

1	Introduction	1
2	Programming Details	4
3	Results and Discussion	8
4	Conclusion	13
5	References	15

List of Figures

1.1	Gaussian process regression over a distributed data [Dr.Juan Camilo Orduz]	2
1.2	Structure of He_2H^+	3
1.3	ANN model architecture [Arden dertat]	3
2.1	Pseudo code for an active selection of data using GPR	5
2.2	Pseudo code for a sequential ANN model	6
3.1	Predicted PES when 100 data points were used	8
3.2	Predicted PES when 200 data points were used	9
3.3	Predicted PES when 400 data points were used	9
3.4	Predicted PES when 500 data points were used	10
3.5	original PES created using all data provided	10
3.6	Potential energy Vs atomic radius at real values and at predicted values . . .	11
3.7	Potential energy Vs atomic radius at real values and at predicted values for other test data	11
3.8	MSE vs Epoch of ANN model	12

Chapter 1

Introduction

Potential energy surfaces (PES) are a fundamental concept in the field of molecular simulations, and they play a crucial role in our understanding of the chemical reactions of systems and the behavior of molecules. In simple terms, a potential energy surface (PES) is a mathematical representation of the potential energy of a molecule as a function of its atomic positions. Calculation of potential energy surfaces of molecules with complex topography like He_2H^+ is resource-heavy and time-consuming. PES is also used for quantum dynamical studies. So for this study, He_2H^+ PES needs to be calculated.

Gaussian Process Regression (GPR) is a non-parametric Bayesian approach especially used for regression analysis. It is a powerful statistical method that can be used for supervised machine-learning tasks, such as classification and regression. It is often used for tasks such as regression modeling, data interpolation, and data imputation. As we can see in Fig 1.1. Gaussian process regression model which is actively selecting data over normally distributed datasets.

Machine learning algorithms gained vast popularity over recent years. From image and speech detection to statistical modeling, natural language processing, fraud detection, and many more. One of them is Deep learning. **Deep learning** is a field of machine learning

that focuses on creating neural networks with multiple layers to learn and represent complex features from data. Deep learning models are designed to copy the structure and functionality of the human brain by using **interconnected layers**¹³ of artificial neurons to learn representations of data. The framework of a sequential ANN model is shown in Figure 1.3. There are many layers, each containing a certain number of nodes called neurons. The mode of operation of an ANN involves extracting the features from the training dataset fed into the input layer. Once the input layer nodes are provided with the data, definite weights are assigned to each neuron. During the training process, the weights on the connections between the neurons are adjusted using an optimization algorithm known as **backpropagation**. The main aim of this training process is to minimize the difference between the neural network's output and the accurate output, also known as the loss or error. This is usually done by adjusting the weights in the neural network to reduce the error on a set of training data and then testing the network on a separate set of test data to ensure that it is able to predict values to new data.

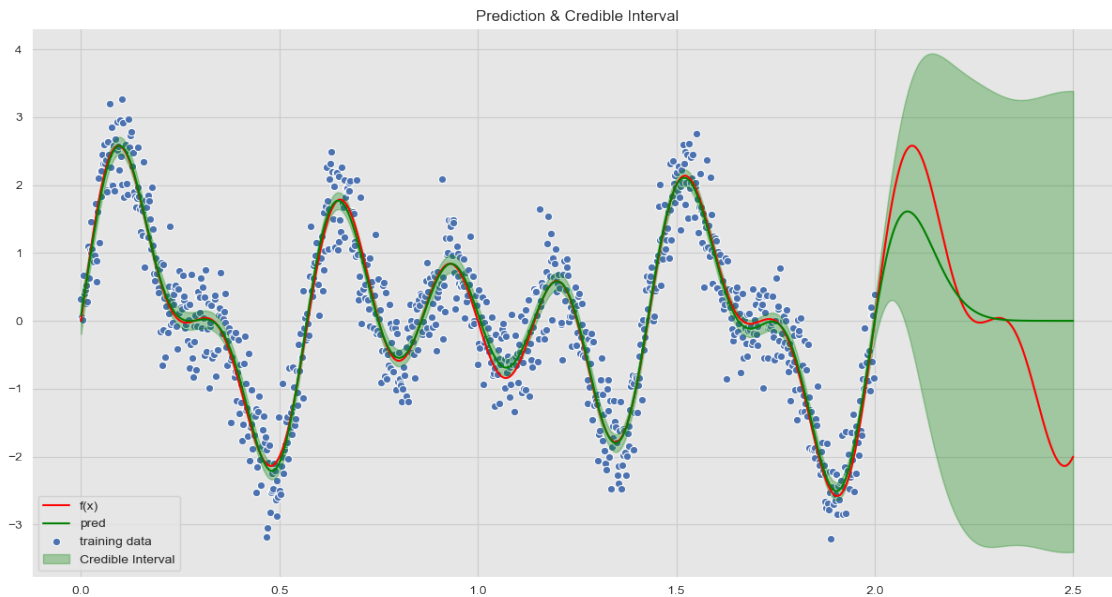


Fig. 1.1 Gaussian process regression over a distributed data [Dr.Juan Camilo Orduz]

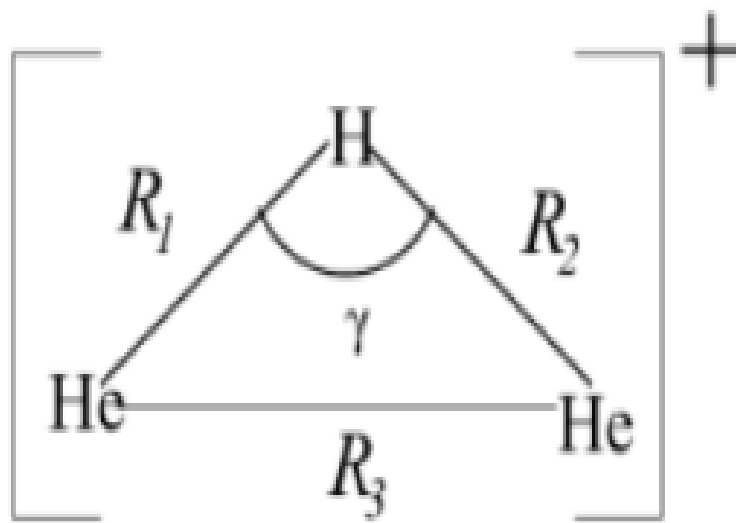


Fig. 1.2 Structure of He_2H^+

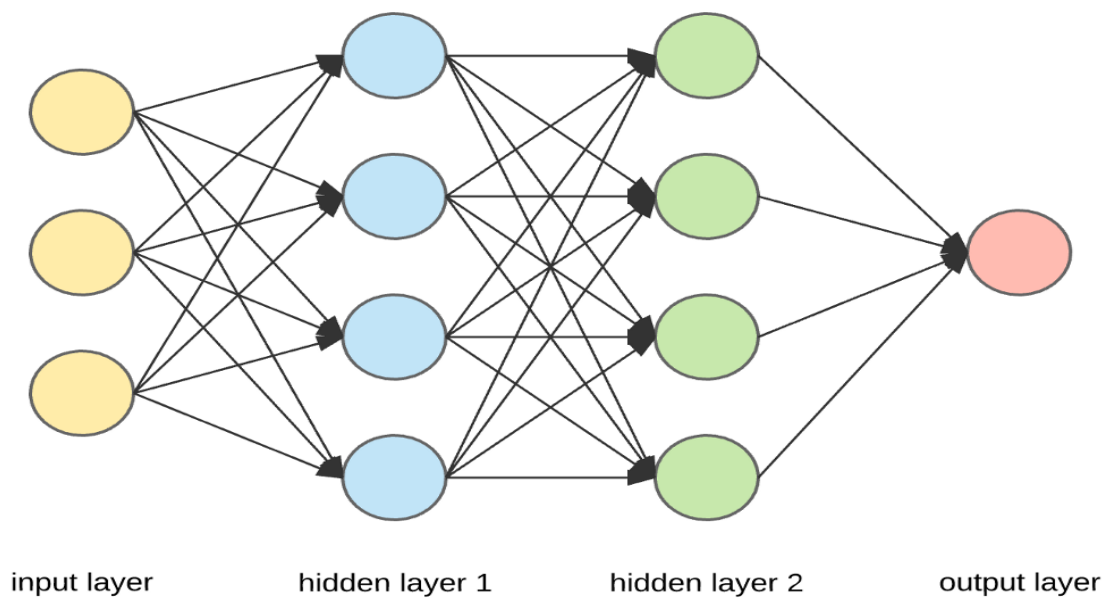


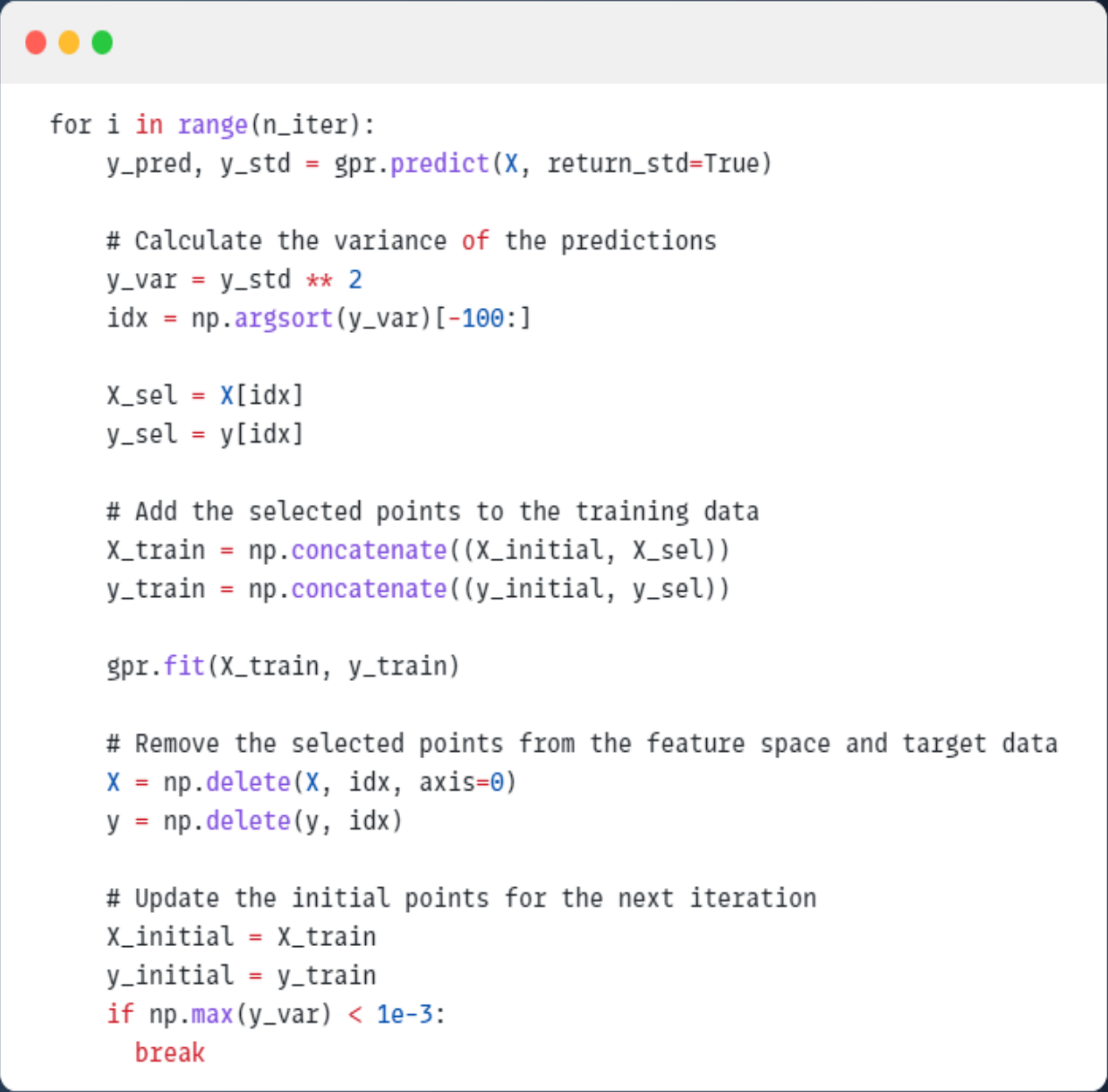
Fig. 1.3 ANN model architecture [Arden dertat]

Chapter 2

Programming Details

The initial dataset contains 2500 points which are calculated using MOLPRO¹ software over a set of values for bond lengths R1 and R2 while setting the bond angle at a constant value as shown in Fig.1.2. Pre-processing of the data includes splitting positions and potential energy columns using **numpy**⁴, removal of blank values and organizing the data in columns namely X and y where X represents a 2D array containing both the R1 and R2 and y represents potential energy.

For active data selection, we have used the Gaussian process regressor(GPR) with **Matern**⁵ kernel having $\nu = 2.5$. First, we start with an initial dataset of a small size of 10 and then iterated a loop for finding the configurations with the highest predictive **variance**¹² by adding some of the remaining data with the initial dataset and running the loop until the predictive variance of data converges to 1e-3.



```

for i in range(n_iter):
    y_pred, y_std = gpr.predict(X, return_std=True)

    # Calculate the variance of the predictions
    y_var = y_std ** 2
    idx = np.argsort(y_var)[-100:]

    X_sel = X[idx]
    y_sel = y[idx]

    # Add the selected points to the training data
    X_train = np.concatenate((X_initial, X_sel))
    y_train = np.concatenate((y_initial, y_sel))

    gpr.fit(X_train, y_train)

    # Remove the selected points from the feature space and target data
    X = np.delete(X, idx, axis=0)
    y = np.delete(y, idx)

    # Update the initial points for the next iteration
    X_initial = X_train
    y_initial = y_train
    if np.max(y_var) < 1e-3:
        break


```

snappify.com

Fig. 2.1 Pseudo code for an active selection of data using GPR

This results in forming a trainable dataset that we can directly use in a sequential ANN model for predicting PES. Here we create a sequential ANN model by using keras³2.6.0 and tensorflow²2.6.0 libraries. First, we split the data into two parts one is trainable and the

other is test data which will be used in finding the accuracy and loss of the model known as `test_train_split`⁶.



```
# create the neural network model with three layers
model = keras.Sequential([
    keras.layers.Dense(64, activation='relu', input_shape=(2,)),
    keras.layers.Dense(32, activation='relu'),
    keras.layers.Dense(1, activation='linear')
])

# compile the model with mean squared error loss function and Adam optimizer
model.compile(optimizer='adam', loss='mean_squared_error', metrics=['mse'])

# train the model on the training data
history = model.fit(X_train, y_train, epochs=10000, batch_size=32, verbose=0)

# evaluate the model on the testing data
y_pred = model.predict(X_test)
```

snappify.com

Fig. 2.2 Pseudo code for a sequential ANN model

For the dataset, we got after active selection using GPR, the best configuration turned out to be a three-layered sequential ANN model containing 64, 32, and 1 neurons in the **1st**, **2nd**, and **3rd** layers respectively. The input shape is set as 2 based on the features X. **ReLU**⁷ activation function was used on layers 1st and 2nd while the **linear**⁸ activation function is used on the 3rd layer of the neural network model.

Optimizer is set as Adam which is used for updating the weights of neurons in each epoch

and mean squared error was taken as the metric for model evaluation. The generated model has trainable data of 500 parameters to be trained on. The feature, X , was fed to the model as a 2D numpy⁴ array. The predicted output is a 1D array of equal size to the length of the input array. At the start, we split the data into testing and training data to check the predictions of the ANN model on those testing data for validation. We split the data as 20% will be used as test data and the remaining 80% will be trainable data. **Matplotlib**⁹ **3.4.1** was used for plotting the predicted data. The **Trisurf**¹⁰ library was used for visualizing the model-generated PES.

Chapter 3

Results and Discussion

The model was trained several times using 100, 200, 300, and 500 data points after active selection by GPR on epoch numbers starting from 3 times of size dataset to 10000 epochs. On running the model multiple times here are some observations:

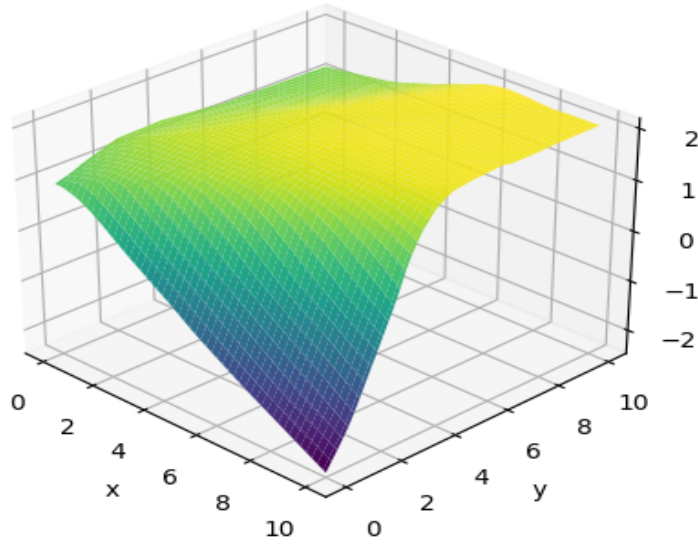


Fig. 3.1 Predicted PES when 100 data points were used

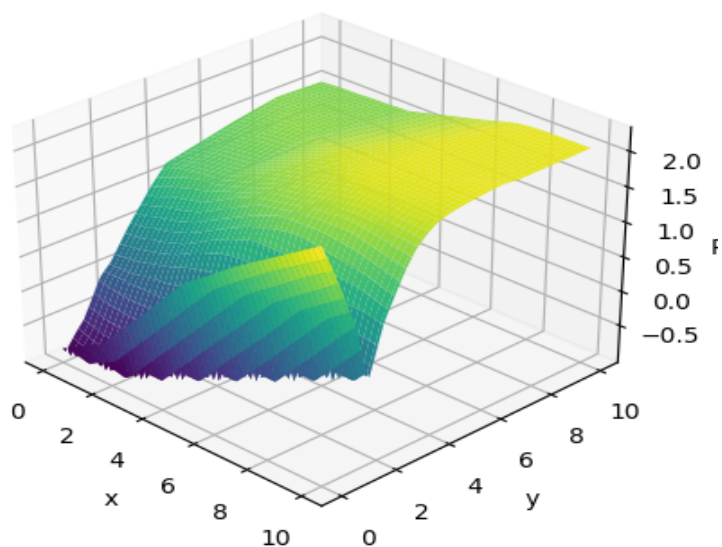


Fig. 3.2 Predicted PES when 200 data points were used

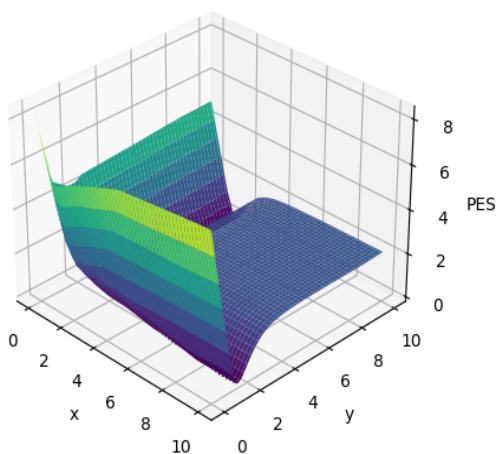


Fig. 3.3 Predicted PES when 400 data points were used

This PES has complex topography to verify our model. I took some random datasets of size 50 of the initial dataset provided in 2 text files. The original PES is created by using Matplotlib₉ on all data provided initially.

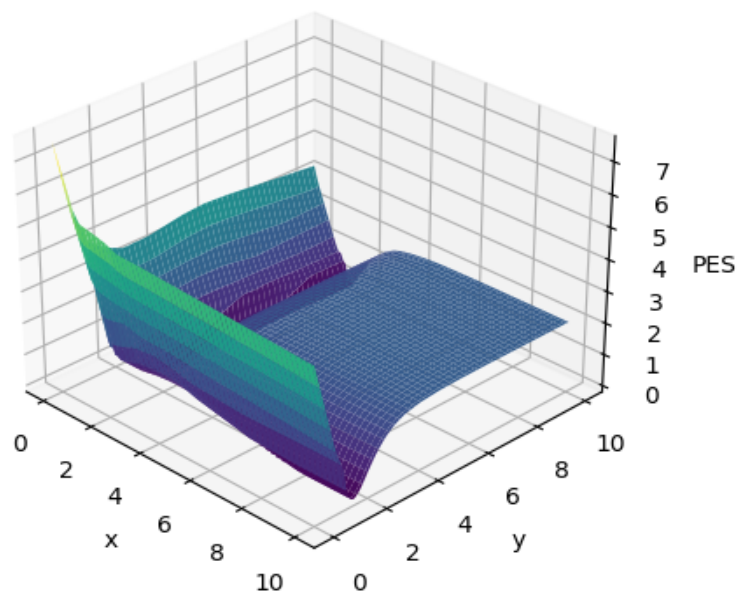


Fig. 3.4 Predicted PES when 500 data points were used

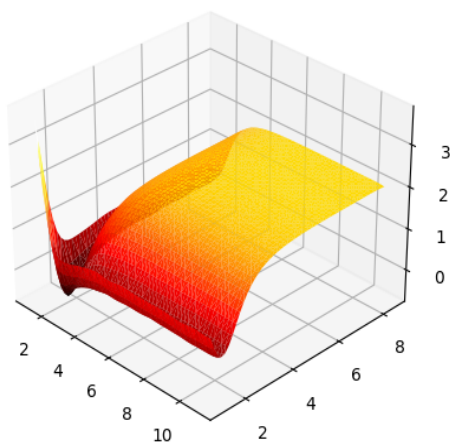


Fig. 3.5 original PES created using all data provided

In this case, instead of going the way we predicted potential energies at unknown R-values before, we will take only one atomic distance and potential energy at that point it will be a 2D curve. At first, we will plot 50 points directly, and then by using the ANN model here

are some observations:

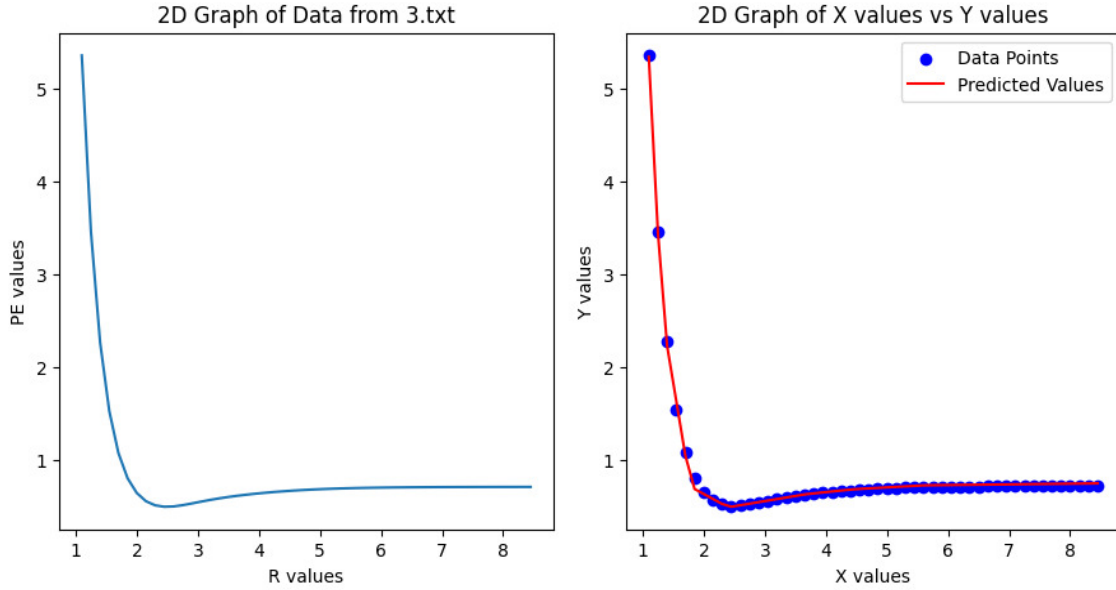


Fig. 3.6 Potential energy Vs atomic radius at real values and at predicted values

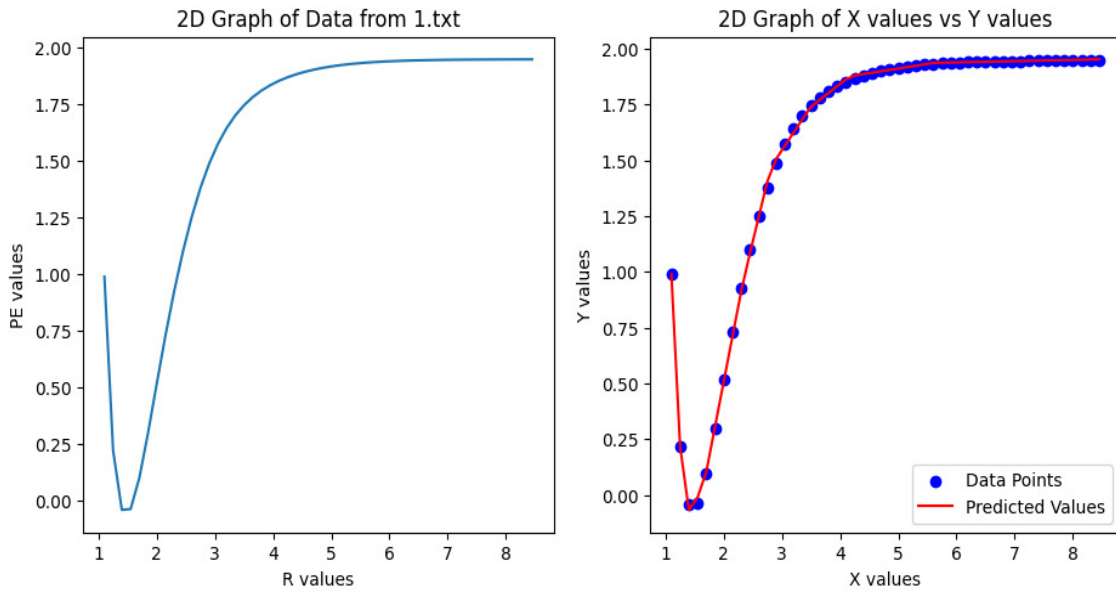


Fig. 3.7 Potential energy Vs atomic radius at real values and at predicted values for other test data

As we see in Fig.3.6 and Fig.3.7 predicted values by the ANN model are almost equal to accurate values. This results that the ANN model is working fine with very high accuracy

and quite low MSE¹⁵.

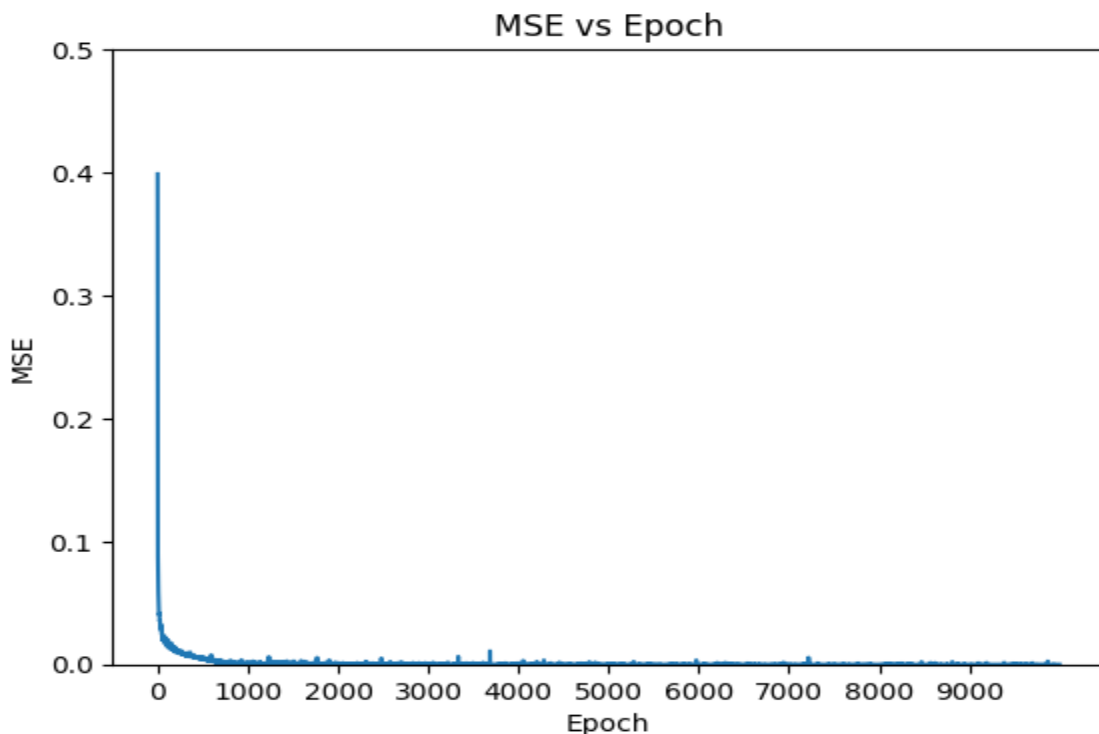


Fig. 3.8 MSE vs Epoch of ANN model

The MSE vs Epoch results of the ANN model can be observed in Fig. 3.8 indicating MSE gradually decreases as the number of epochs increases. As shown in figure 3.8 MSE is suddenly dropping at a fast rate from epoch 0 to 1000 and then gradually converging to 0.00084 over 10000 epochs. Hence this shows that this method is more reliable compared to both individual methods of predicting PES using GPR(which is highly uncertain at the random point) and directly using the ANN model on the whole dataset which takes very much time. This method counters the limitation of the model by using the GPR model for selecting active data incorporating this with the ANN model which will take less time than a simple ANN model. So, incorporating GPR along with NN fitting makes the method reliable for generating potential energy surfaces.

Chapter 4

Conclusion

In this project, we have analyzed the use of Gaussian process regression along with artificial neural networks to predict the globally accurate potential energy surfaces of He_2H^+ systems that are used in quantum dynamical studies. A training dataset was generated using active data selection by the Gaussian process regression model and took the point which has the highest predictive variance and turned the configuration that has high variance as trainable data for the ANN model for predicting PES. Then after, a sequential ANN model was created and trained upon four different datasets. 100, 200, 400, and 500 data points were taken in the four datasets. The results were obtained after using different combinations of neuron numbers and different numbers of layers of the neural network of dataset and epochs were observed. The metric is set as mean squared error and the ANN model trained on 500 data points over 10000 epochs performed the best among all the datasets having a training MSE of 0.00084. The model is also tested for random datasets taken from initial data. The ANN model is able to generate globally accurate PES from a small number of data points. Also, more data will give better results. The above results show that incorporating GPR with the ANN model is a reliable tool for generating accurate PES for a system having complex topology while reducing the computational cost of ab initio points and time consumption by a significant amount. This counters the limitations of both the model i.e. constructed using

GPR or generated using ANN as GPR is highly uncertain hence making it an unreliable tool for accurate predictions Also ANN on big data takes so much time which is not that fast so incorporating the power of both the model GPR and ANN we get a reliable option which makes it highly reliable and very less time-consuming compared to ANN model.

Chapter 5

References

1. Werner, H.-J.; Knowles, P. J.; Knizia, G.; Manby, F. R.; Schütz, M. Molpro: a general-purpose quantum chemistry program package. Wiley Interdiscip. Rev.: Comput. Mol. Sci. 2012, 2, 242–253..
2. https://www.tensorflow.org/guide/keras/sequential_model
3. <https://github.com/fchollet/keras>
4. <https://numpy.org/doc/stable/user/index.html>
5. https://scikit-learn.org/stable/modules/generated/sklearn.gaussian_process.kernels.Matern.html
6. https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html
7. <https://builtin.com/machine-learning/relu-activation-function>
8. <https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6>
9. <https://matplotlib.org/stable/tutorials/index>
10. https://matplotlib.org/stable/gallery/mplot3d/trisurf3d_2.html

11. Interpolation and Extrapolation of Global Potential Energy Surfaces for Polyatomic Systems by Gaussian Processes with Composite Kernels J. Dai and R. V. Krems Journal of Chemical Theory and Computation 2020 16 (3), 1386-1395 DOI: 10.1021/acs.jctc.9b00700
12. <https://www.javatpoint.com/bias-and-variance-in-machine-learning>
13. <https://keras.io/api/layers/>
14. <https://wiki.python.org/moin/BeginnersGuide>
15. https://scikit-learn.org/stable/modules/generated/sklearn.metrics.mean_squared_error.html

Predicting global accurate potential energy surfaces with complex topography using combination of both gaussian process regression and neural networks

ORIGINALITY REPORT

17%

SIMILARITY INDEX

1%

INTERNET SOURCES

1%

PUBLICATIONS

15%

STUDENT PAPERS

PRIMARY SOURCES

1

Submitted to Indian Institute of Technology
Guwahati

Student Paper

14%

2

Submitted to University of Auckland

Student Paper

1%

3

Udit Jindal, Sheifali Gupta. "Deep Learning-
Based Knowledge Extraction From Diseased
and Healthy Edible Plant Leaves",
International Journal of Information System
Modeling and Design, 2021

Publication

1%

4

eprints.utar.edu.my

Internet Source

1%

5

repositories.lib.utexas.edu

Internet Source

1%

Exclude bibliography Off