**Report – Analysis between ANN and GNN model**

**Introduction**- A study was made on finding the hopping energy barrier for a system of fcc metal surfaces like Ag,Al,Ni,Pd,Pt,Cu for different configurations of nearest neighbors surrounding the site of hopping. A total of 28 sites is chosen from the lattice surface to study the energy barriers related to the system during the process.

**Results:** Initially the model was tested considering bias parameter ( ), one way and two way interactions only. The metal used for the fig 1, is Ag(100). Similarly this data was tested with all configurations (including 3 way interactions). Both models gave a good result, a low mean squared error value and r^2 value in the range 0.98-1.

Table 1: R^2 values obtained for different metals.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Two way ANN Model |  |  |  |
| Material | Training | Testing |  |  |
| Ag | 0.995 | 0.9948 |  |  |
| Al | 0.9928 | 0.9925 |  |  |
| Ni | 0.994 | 0.993 |  |  |
| Pd | 0.996 | 0.9959 |  |  |
| Pt | 0.992 | 0.991 |  |  |
| Cu | 0.991 | 0.9909 |  |  |
|  |  |  |  |  |

Table 1: R^2 values obtained for different metals.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Three way ANN Model |  |  |  |
| Material | Training | Testing |  |  |
| Ag | 0.994 | 0.9938 |  |  |
| Al | 0.993 | 0.992 |  |  |
| Ni | 0.993 | 0.9928 |  |  |
| Pd | 0.997 | 0.9969 |  |  |
| Pt | 0.994 | 0.993 |  |  |
| Cu | 0.992 | 0.991 |  |  |
|  |  |  |  |  |

**Training Data (For two way and three way)**

*Calculation of Activation energy:* From the slab of 100 atoms in a layer, 28 local sites were taken around the hopping atom. Symmetry conditions were imposed and negative energy value configurations were excluded. Along with the energy levels below 0.01 eV.

From the data, bias parameters and interaction energies is calculated. A total of 7680 configurations are taken out of the possible 2^28=268,435,456 possibilities. The fraction of these data used for training, testing and validation is 0.7, 0.1 and 0.2 (5376, 768, 1536). The number of one way, two way, three way interactions are,,. This is a total of 3682 parameters + bias parameter. This data is fed in the ANN model. In case of Ag, the input data has the relevant sites of 1 way, 2 way and 3 way interactions which is –

a. 8, 10, 14, 15

b. 4\*10, 8\*9, 9\*10, 9\*19, 9\*20, 10\*14, 10\*20, 13\*14

c. 8\*9\*14, 9\*10\*20, 9\*15\*19, 10\*14\*20

Instead of 28 configurations, 4+8+4 = 16 most relevant sites are fed.

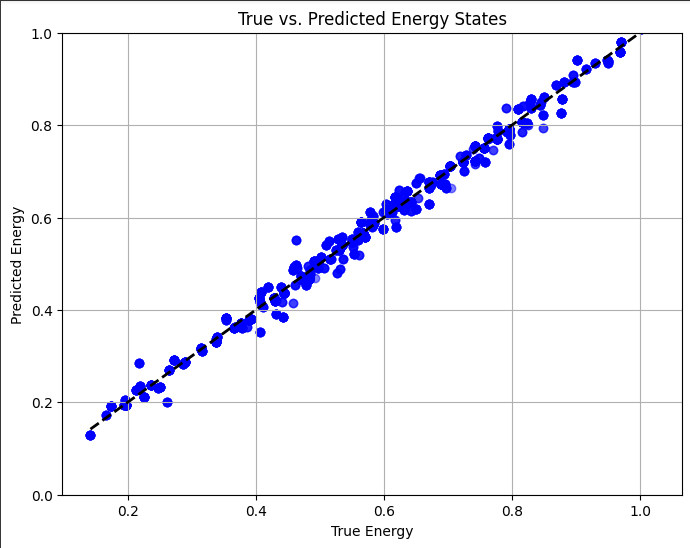
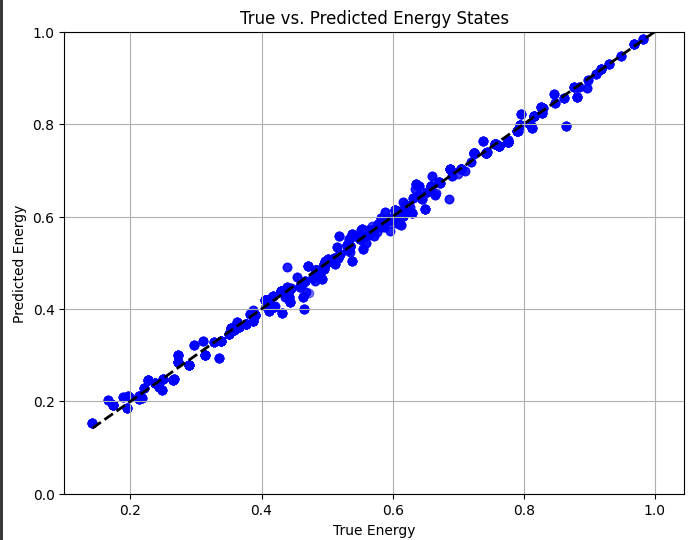
*Model:*

ANN 2 way/3 way model, the tensorflow 2.0 module in python was used. Model layers were designed in the keras.Sequential() function. An input layer of 32 neurons and activation of ‘Relu’ , along with 1 hidden layer with 64 neurons and same activation function to fit in the model , last layer is the output layer. Relu is chosen because it deals with the non- linearity of the data very well. The activation function is differentiable at all points except zero, for values greater than zero max of the function is considered.

Then the model is fitted with a batch size of 10 and optimizer. The optimizer used is SGD(Stochastic Gradient Descent). And the metrics used for defining the loss function is “Mean Squared Error”.

Below shown is the GD algorithm, but since for larger data the computation time involves finding residuals for all data points can be very high (number of data points \* features \* iterations). So SGD is used, instead of using all data points only one point is chosen with randomly probability and then the weights are calculated.

a) b)

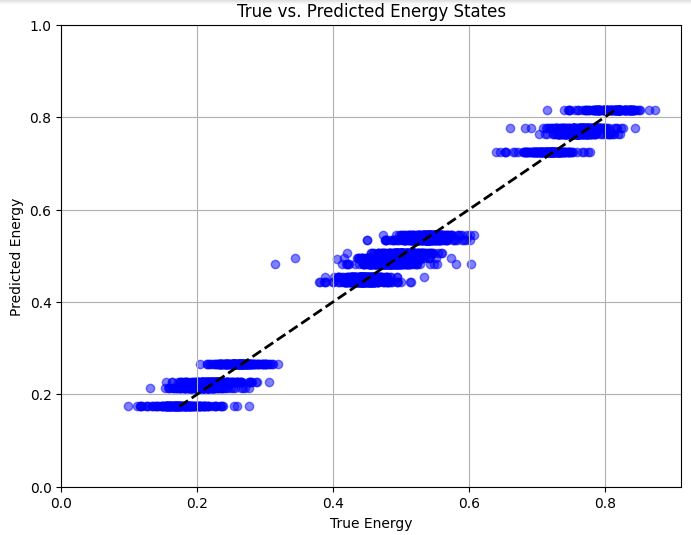
  *Fig(a) and Fig(b) shows the graph between predicted and actual energy barrier values in a two- way interaction ANN model and a three- way interaction ANN model respectively.*

It was found out that for only one way interaction, the results came out differently.

*Training data (One way):* The input data that is fed as raw configuration (0’s & 1’s) from 28 sites. And the corresponding energy barriers considering only one way interactions.

As for the model, the shape of input layers is changed to (28,1) array.

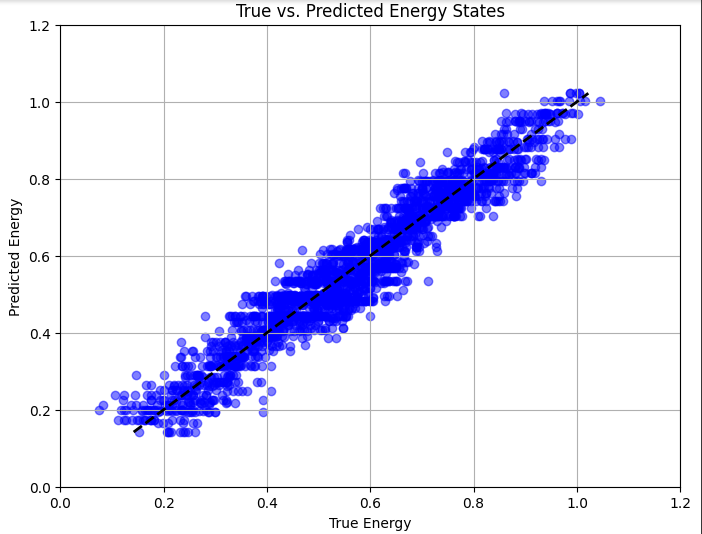
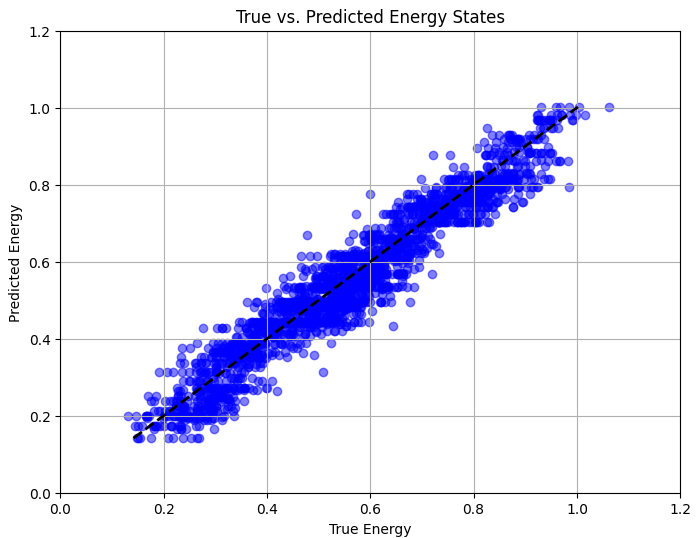
c)



Fig(c) *shows the graph between predicted and actual energy barrier values in a one- way interaction ANN model*

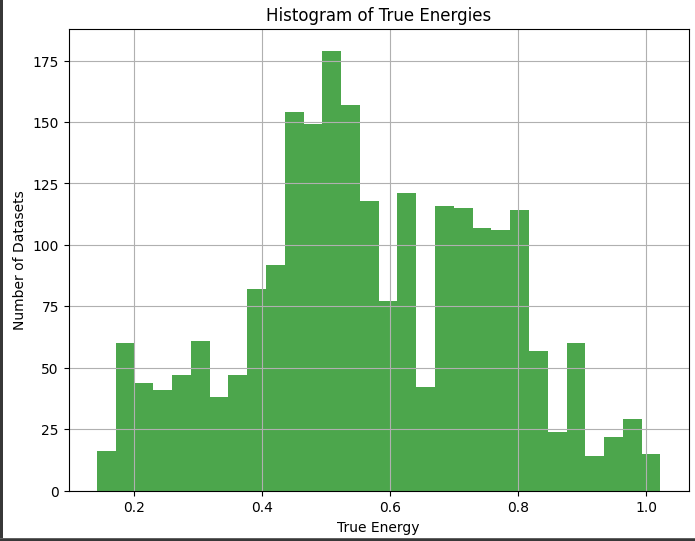
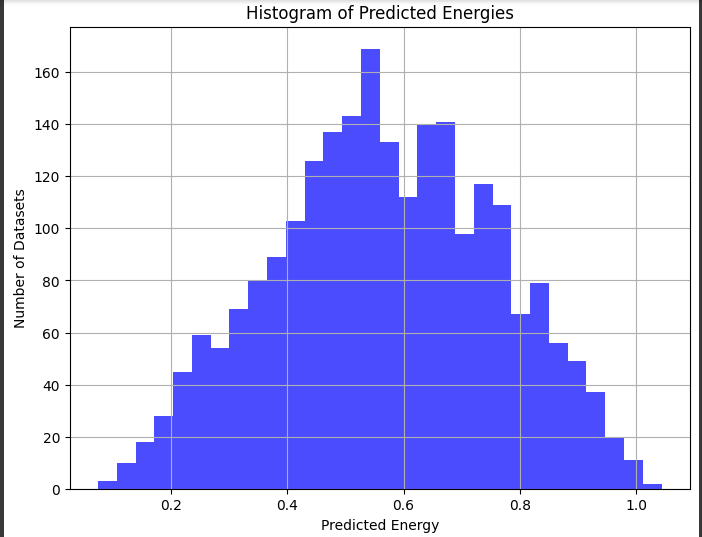
Another set of data was fed as input in the ANN, for input vector – (28,1) the energy levels used for 2 way and 3 way gave the following graphs.

1. *ii)*

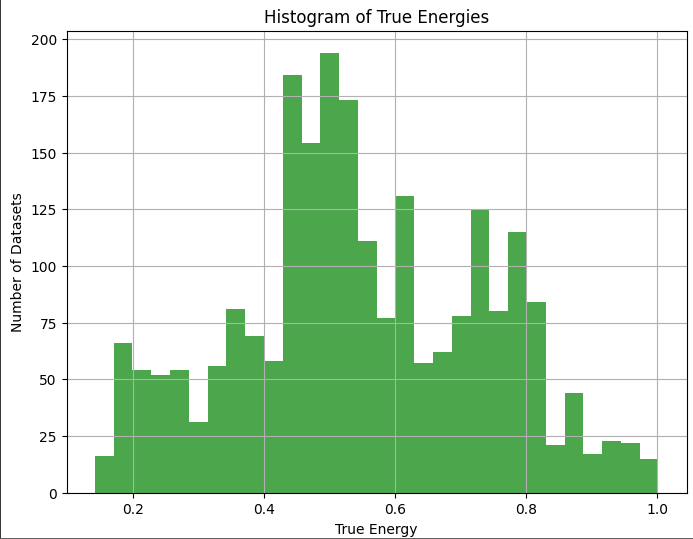
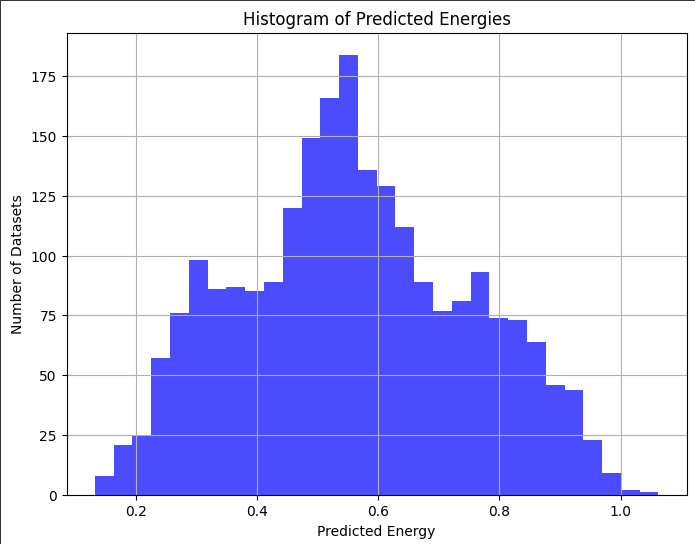
 

*Fig (i) and (ii) are plots of predicted energy vs true energy for two way energy and three way energy vector with (28,1) input configurations.*

1. *ii)*



*iii) iv)*



*Fig (i) and (ii) are histograms of predicted values vs the true energy respectively by using two way interaction energy data. Fig (iii) and (iv) is by using three way interaction energy data*

**Graphical Neural Network Model**

As for the simple GNN model, the training data consists of the information of the raw configurations.

Then the input data is converted in the form of adjacency matrix. Three kinds of values were used inside the matrix:

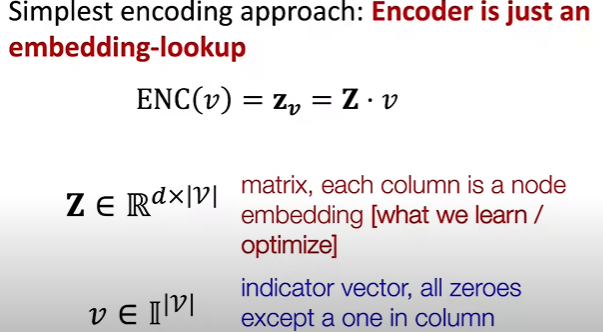
1. Products of (0,1) in respective sites
2. Assigning a physical value for the sites, to find the relative distance between the reacting sites.
3. Assigning 1/ (n=2, in this case), relative to the coordinates of reacting sites.

Values in the range of 0-1 inside the adjacency matrix gave better fitting results.

For the model, python’s pytorch library is used. Inside this library numerous modules have been used.

1. Torch.nn.functional – calls the activation functions like Relu
2. Torch\_geometric.data – imported modules –Data, DataLoader, GCNConv

All these functions are provide similar features as any other neural network just deals with the graph type data structures by making node embeddings from nodes. This is generally done using Enoder/Decoder algorithm (eg: Node2vector or Deepwalking).



*This figure shows the simple shallow encoding algorithm*

The adjacency matrix for 7680 datasets has a dimensionality of (28\*28\*7680) which can be computationally time consuming, so this data is converted into a sparse matrix. A sparse matrix is made by using the coordinates of sites which have ‘1’ and ignoring the sites which had a value of ‘0’. This decreases the dimensionality of the input matrix.

Hyper parameters like edge attributes, node attributes are considered by default=1. These weights have not been considered based on the data. After this sparse matrix is made and edge/node attributes are defined this data is made into a large list of “Graph\_data\_list”. This holds the structure of the graph data that is inputted.

For the functions used to design the 3 layers (input, hidden, output) in the model-

1. init\_\_super():

This function sets the basic layers that are convoluted (in our case a cost function is not used so simply performs additive operations in the layers ) .

First layer-(1,32) {argument is the size of array of neurons that is fed}

Second Layer-(32,16)

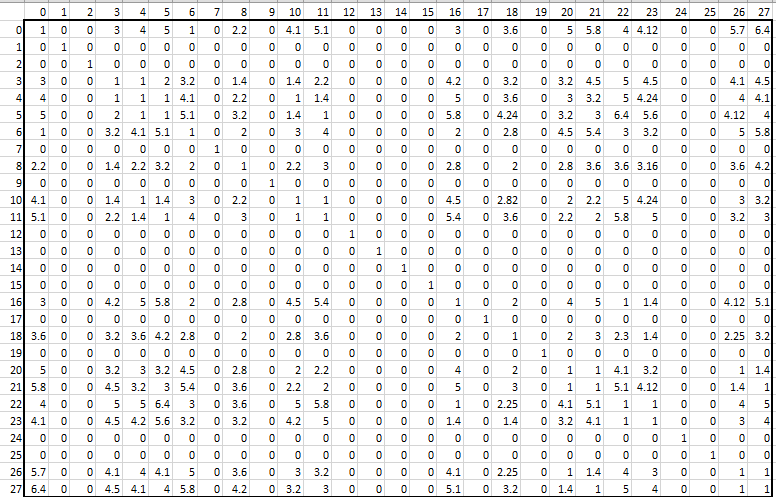
Third layer-(16,1) {Linear output layer}

1. forward():

The activation function and pooling of layers is added in this function. Also includes the usage of weights (node/edge). In our case “Relu”, “Pooling layer” is added.

After training and testing model with the set of values, these graphs were obtained for different sets of data that is used (one way, two way, three way interactions). The input for all these remained the 28 site raw configuration fed in form of an adjacency matrix.

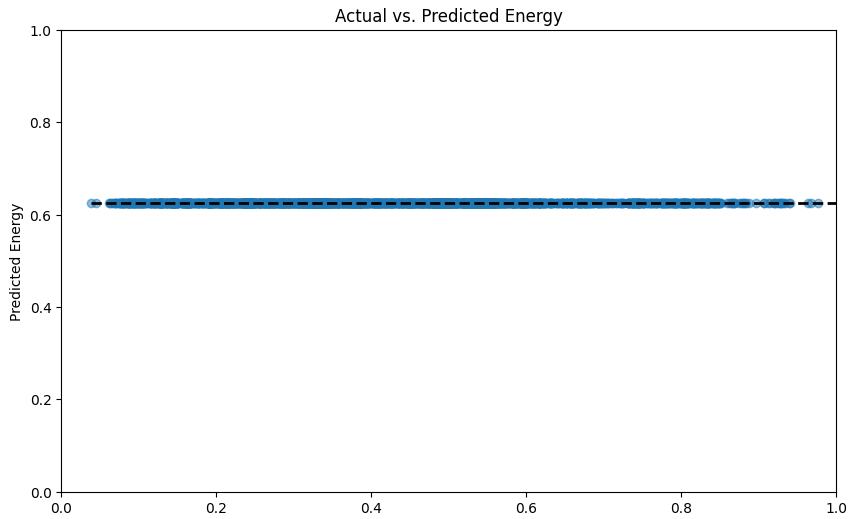
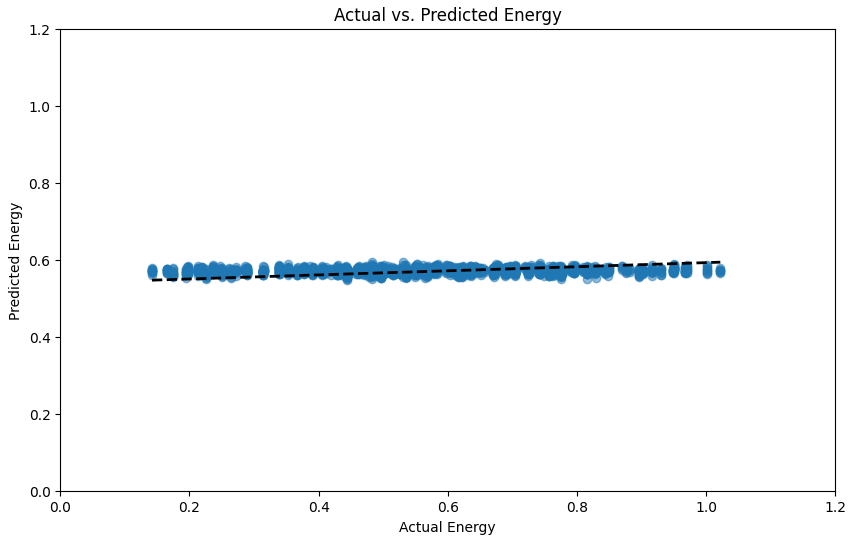
d)



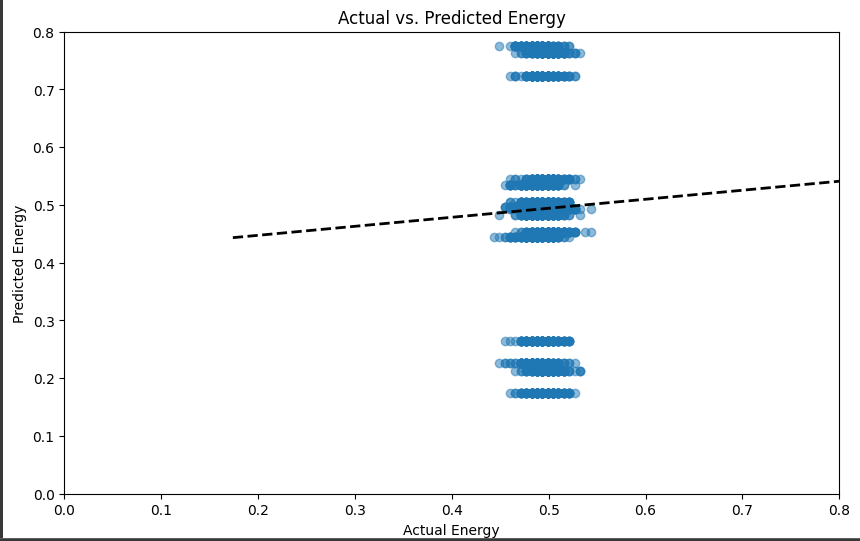
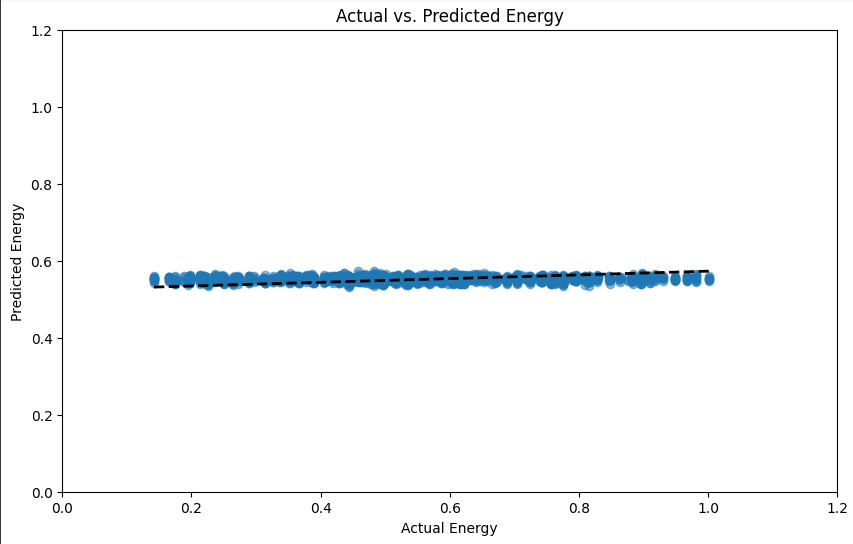
*Fig(d) shows an adjacency matrix with relative radius as its values*

When these matrices are fitted in the model, the following results are obtained for different values of energy barriers used (one way, two way, three way interactions).

e) f)

g) h)

*Fig(e), Fig(f), Fig(g) and Fig(h) shows the graph between predicted and actual energy barrier values in a GNN model considering actual, two way, one way and three way energy barrier.*

A common conclusion that can be drawn from all these graphs are that the results obtained is closer to the mean values of the energy barriers fed in the model. To be noted that one way interaction results are very similar for both the models (ANN & GNN).

Further investigation was made to find the weights and biases in each layer of the model. Following values were obtained:

In the ANN model (for only one way interaction)-

*Case-1(small/no deviation)*

For the energy barrier fed to 0.496 and predicted value coming out as 0.49009775. The following biases were used.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| layer\_0\_weights | layer\_0\_biases | layer\_1\_weights | layer\_1\_biases | layer\_2\_weights | layer\_2\_biases |
| 0.050580535 | -0.025398988 | 0.17408411 | -0.038681805 | -0.28371271 | 0.32837686 |

*Case- 2(negative deviation)*

For the energy barrier used was 0.776 eV and resulted in values like 0.239. The following biases were used.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| layer\_0\_weights | layer\_0\_biases | layer\_1\_weights | layer\_1\_biases | layer\_2\_weights | layer\_2\_biases |
| 0.25051776 | -0.020228993 | 0.0770788 | 0.011935617 | 0.011663596 | 0 |

*Case -3 (positive deviation)*

For the energy barrier used was 0.174 eV and resulted in values like 0.779. The following biases were used.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| layer\_0\_weights | layer\_0\_biases | layer\_1\_weights | layer\_1\_biases | layer\_2\_weights | layer\_2\_biases |
| 0 | 0 | 0.056374677 | 0 | 0 | 0 |

So three broad classes of results are found, that is showing a big positive deviation from the mean and the other that is showing a big negative deviation from the mean value.

For the three layers added, the bias and weight add up following this equation, where are the input values and is the activation function.

**Conclusion**: The GNN model is working similar to the ANN when only one way interactions are provided, but in case of two or three way interactions it is calculating mean values and approximating the predictions.