Atomisation Energy Prediction using Graph Neural Network

Project Report

Module Name: Machine Learning for Molecular Physics

Manish Lohani

Contents

List of Tables

List of Figures

1 Introduction

The quantum chemical calculations using DFT (density functional theory) [**1**] are very accurate but expensive with respect to both the computer resources and the computation time. In the last two decades, machine learning (ML) approaches are being heavily applied in scientific areas as well such as molecular simulation, chemistry and molecular property prediction. One of the key contributions in this direction, in 2007, Behler and Parrinello [**2**], used neural network (NN) for the DFT potential energy surfaces (PES) using the symmetry functions to define the atomic environment also knows as descriptors.

In 2012, Matthias Rupp et.al [**3**] predict the atomisation energies using the Coulomb matrix representation based on the nuclear charges and the atomic positions implementing with kernel ridge regression. The work of Mr. Behler and Parrinello was advanced by J. S. smith et.al, who used modified version Behler and Parrinello symmetry functions to build single-atomic environment vectors for molecular representation.

The task to define atomic descriptors resembles the task of exploring better exchange-correlation functional in DFT. The accuracy of ML models relies highly on the descriptors which are the inputs to the ML models. To tackle the problem of finding optimal descriptors, in recent years, graph neural networks (GNN) are being used to predict molecular properties, protein interaction, ligand-binding, drug-drug interaction and in other areas. These models takes advantage of the graph data structures to define the node and the edge feature representation with a message passing architecture in which neighbouring nodes or edges exchange information and influence each other's embedding to learn about their environment.

In this work, we use GNN so that we do not need to incorporate symmetry function for our investigation to predict atomisation energy at 0K for molecules with the QM9 dataset [**4**]) without providing the atomic positions. For this purpose, we have used graph convolutional network (GCN) [**5**] and relational graph convolutional networks (RGCN) [**6**]. Since the relational GCN allows us to incorporate edge type (bond type) as a edge feature in our model, it could be a good choice for molecules property prediction. The regular GCN only supports node features. For our calculations, we have used pytorch [**7**] and pytorch-geometric [**8**] python package.

In the below section, we present the architecture of our networks along with the results showing the comparison of GCN and RGCN models.

2 Dataset, Network Architecture and Results

2.1 Dataset

2.1.1 About QM9 dataset

QM9 dataset [**4**] consists of 134k smallest organic molecules containing up to 9 heavy atoms (C, O, N, or F; excluding H) along with their quantum properties which is widely used for molecular property prediction. In this project work, we have used the QM9 dataset from the the work "MoleculeNet: A Benchmark for Molecular Machine Learning [**9**] paper", consisting of same number of molecules as QM9 dataset but with 19 regression targets. We use it to predict atomization energy at 0 K. The atomization energy represents the energy gained or lost by arranging each atom in a crystalline solid, as compared with the gas state which is a measure of how atoms are bound together in a molecule.[**10**]

2.1.2 Data processing

We have used the built-in QM9 API of the pytorch-geometric package to preprocess and load the QM9 dataset. This API In the processed dataset, each node is defined with 11 features such as one-hot encoding of each atom type (H, C, N, O, F), atomic number, aromatic, hybridization type and number of hydrogen attached. The graph connectivity is represented in the coordinate (COO) format. We made an additional edge-type features for the implementation of RGCN.

We split the the dataset with around 80%, 10% and 10% for the train, validation and the test set respectively. To standardize the the experiments, we use the the same set of dataset indices for train, validation and test set for all models. The target values (atomisation energy U_0) are normalised for training set only with 0 mean and 1 as standard deviation. We did not normalise the target values for the complete set since it can leak information to validation and test sets.

2.2 Network Architecture

2.2.1 Initial Model for Investigation

The output of the series of k (a variable) number of GNN convolution layers is fed into a global mean pool layer. The pooling layer returns batch-wise graph-level-outputs by averaging the node features across the node dimension, so that the a single graph level prediction can be done. The output is fed into a linear layer with Relu activation function and a dropout layer. We have used L1 loss (MAE) function for training our network since we found that the L2 loss exhibited

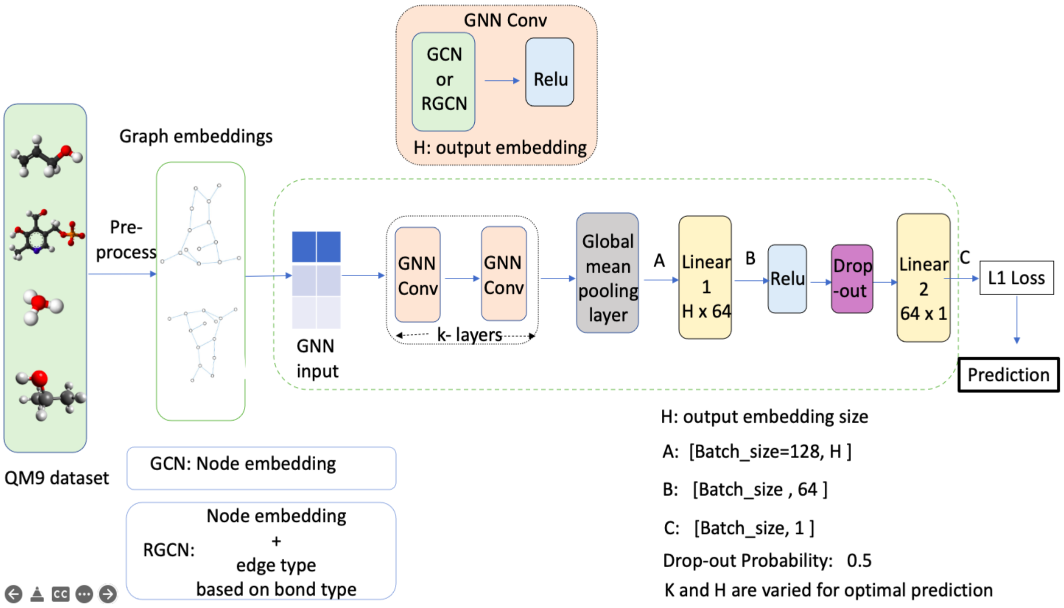

Figure 2.1: Network architecture: Processed QM9 dataset generates graph embeddings which fed into ML model. GCN and RGCN are implemented individually for comparison. L1 loss is used for final prediction. k (the number of GNN conv module) and H (output node and edge embedding size) are varied for better prediction.

very slow convergence. The batch size for data loaders is taken as 128 in all experiments. The learning rate is taken as 0.001 and we used Adam optimiser.

2.2.2 Results

To check the inclusion of dropout layer or not, we first investigated the validation error results after 100 epochs. We found out that drop out layer generates lower validaton error after 100 epochs. However, the difference was higher with higher number of conv embeddings (H).

We then investigated for size of output embeddings (H) for GNN conv layer. Table 2.1 shows that 64-dim network gives better results than 16-dim after 100 epochs for two GNN conv layers model. Also, RGCN has higher convergence than GCN.

Number of GNN conv layers (k)	**Size of output embedding (H)**	**Validation Error (kcal/mol)**	
		GCN	**RGCN**
2	dim-16	49.233	46.558
2	dim-64	44.851	42.914

Table 2.1: Comparison of validation error for different ouput embedding for k=2 layers

We further investigated with increasing number of GNN conv layers to check the validation error. We found that higher k is giving better results for the same architecture and set of parameters. So we ran the 3-layer Conv network for 300 epochs for the both GCN and RGCN with H = 64 and 128. We found that both 3-layer GCN and RGCN layer models have higher convergence rate and gives better results. The reason could be in the 2-layer network, each node will learn the embedding of the nodes/edges which are only 1 hop away thus learning about only the angle between 3 nodes with message passing framework. However with 3-layer network, each node can learn about the embedding of nodes/edges which are 2 hops away, thus learning about the dihedral angle. Table 2.2 lists the comparison of results for different model architecture. We found 3-layer dim-128 GCN gives better results than other. Figure 2.2 shows the result of 3-layer dim-128 GCN result.

Type of model	**Test set MAE (kcal/mol)**	**Validation set MAE (kcal/mol**
3-layer dim-64 GCN	43.214	40.977
3-layer dim-64 RGCN	39.063	36.803
3-layer dim-128 GCN	18.757	18.050
3-layer dim-128 RGCN	38.625	36.250

Table 2.2: Comparison of different models after 300 epochs

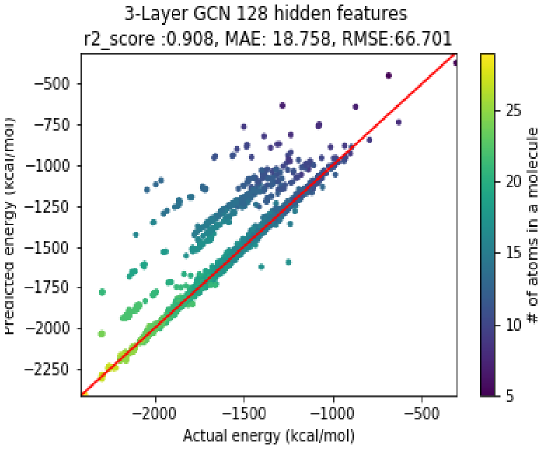

Figure 2.2: 3 Layer 128-dim GCN output embedding with trial network result

We found that for all above four cases with 3-layer network, after 250 epochs the convergence of validation error goes very slow. The same architecture will show better performance if it is run for higher number of epochs since the validation error is still decreasing at 300 epochs also. We further changed the architecture of our network to get better results.

2.2.3 Architecture 2

Figure 2.3 shows the architecture of our second investigating model and table 2.3 lists the validation and test set error for both GCN and the RGCN layer models. In the model, we use Adam as an optimiser with learning rate 0.002. Figure 2.4 shows the results for the test set.We trained the the model for 300 epochs since we found that the convergence around 250-300 epochs is found to be very slow.

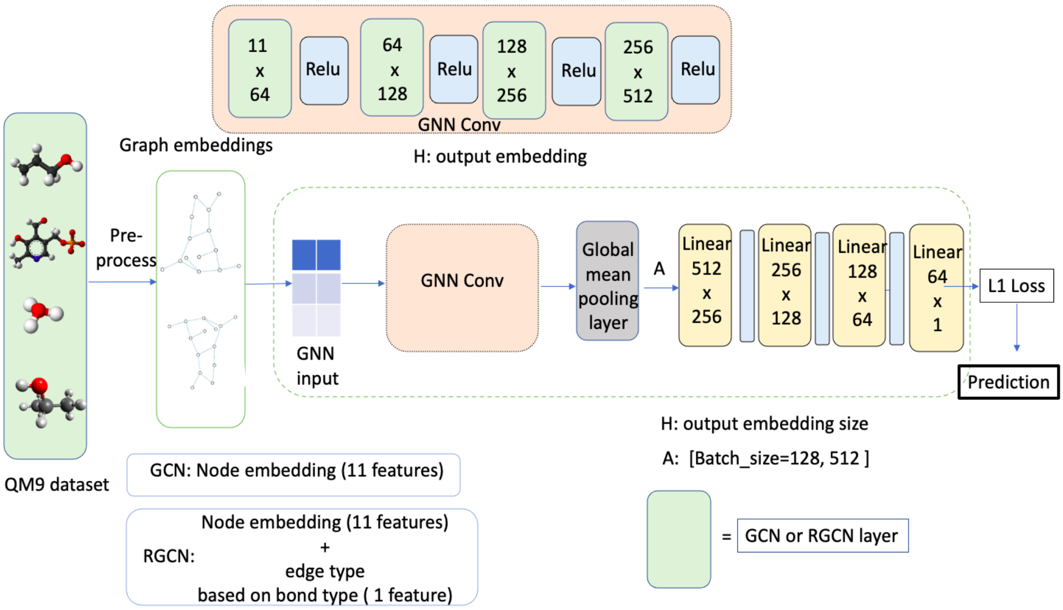

Figure 2.3: Network architecture: Processed QM9 dataset generates graph embeddings which fed into ML model. GCN and RGCN are implemented individually for comparison.

Type of GNN layer	**Test set MAE (kcal/mol)**	**Validation set MAE (kcal/mol)**
GCN	8.689	8.351
RGCN	70.813	10.165

Table 2.3: Comparison of MAE's for architecure 2 after 300 epochs

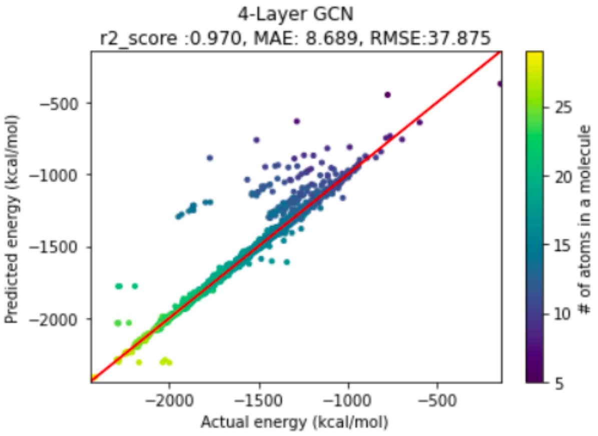

Figure 2.4: Diagram for actual vs predicted energy for the test set with GCN model

Our model with GCN layer shows better result with around 8 kcal/mol than initial model but still it is large with respect to the chemical accuracy (1 kcal/ mol). We would need more sophisticated models will achieve chemical accuracy, such as Gated GCN. This would be necessary since Gated structure would lead to flow of information between far apart nodes (atoms) in the graph.

Bibliography

[1] Hohenberg, P. and Kohn, W. Inhomogeneous electron gas. Phys. Rev., 136:B864–B871, Nov 1964. doi: 10.1103/ PhysRev.136.B864. URL http://link.aps.org/doi/10.1103/PhysRev.136.B864.

[2] Behler, Jörg and Parrinello, Michele. Generalized neural-network representation of high-dimensional potential-energy surfaces. Phys. Rev. Lett., 98: 146401, Apr 2007. doi: 10.1103/PhysRevLett.98. 146401. URL *http://link.aps.org/doi/10. 1103/PhysRevLett.98.146401.*

[3] Rupp, Matthias, Tkatchenko, Alexandre haand Muller, Klaus-Robert, and von Lilienfeld, O. Anatole. Fast and accurate modeling of molecular atomization energies with machine learning. Physical review letters, 108 (5):058301, Jan 2012. URL *http://dx.doi.org/10.1103/PhysRevLett.108.058301.*

[4] Ramakrishnan, Raghunathan, Dral, Pavlo O, Rupp, Matthias, and Von Lilienfeld, O Anatole. Quantum chemistry structures and properties of 134 kilo molecules. Scientific data, 1, 2014.

[5] Thomas N. Kipf, Max Welling. Semi-Supervised Classification with Graph Convolutional Networks. https://doi.org/10.48550/arXiv.1609.02907,2016

[6] Michael Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, Max Welling. Modeling Relational Data with Graph Convolutional Networks. https://doi.org/10.48550/arXiv.1703.06103.2017

[7] https://pytorch.org

[8] https://pytorch-geometric.readthedocs.io/en/latest/index.html

[9] Zhenqin Wu, Bharath Ramsundar, Evan N. Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S. Pappu, Karl Leswing, Vijay Pande. MoleculeNet: A Benchmark for Molecular Machine Learning. https://doi.org/10.48550/arXiv.1703.00564

[10] Masahiko Morinaga. A Quantum Approach to Alloy Design: Material today,Elsevier, 185-220, 2019