**Assignment-2**

GMLFA (AI60007) - Autumn,2024 - IIT Kharagpur

Release Date: [23/08/2024]  
Submission Date: [13/09/2024]

Total Marks: 21

# **Instructions:**

* All graded questions are compulsory to solve, non-graded questions are optional.
* Each group has to ***submit only one file*** named ‘group\_number\_assignment1.ipynb’.
* ***Negative marking*** will be there as per our ***plagiarism policy*** given in the course webpage.
* You can use any language for coding questions, but ***‘python’*** is preferred.
* Frameworks like Pytorch, Tensorflow are encouraged to construct deeper neural network architectures.
* You will be provided with one supporting code notebook (.ipynb) file with pseudocode if required.
* Any required help will be provided to you in the code notebook regarding data or any specific library.

# **Dataset:**

For all the questions asked in this assignment you have to use the **QM9 dataset.**

## **QM9 Dataset:**

The QM9 dataset is a widely used benchmark dataset in the field of graph neural networks (GNNs) and molecular property prediction. It contains about 134,000 small organic molecules with up to 9 heavy atoms (C, O, N, F). Each molecule is represented as a graph, where atoms are nodes and bonds are edges.

Key features of QM9:

- Number of graphs: ~134,000

- Node features: Atom properties (e.g., atomic number, charge)

- Edge features: Bond properties (e.g., bond type)

- Graph labels: Various molecular properties (e.g., energy, dipole moment)

Available target properties:

0: mu, 1: alpha, 2: homo, 3: lumo, 4: gap, 5: r2, 6: zpve, 7: U0, 8: U, 9: H, 10: G, 11:Cv, 12: omega1

Link: https://pytorch-geometric.readthedocs.io/en/latest/generated/torch\_geometric.datasets.QM9.html

The dataset is used for regression tasks, predicting molecular properties from graph structures.

**Use Case:**

* **We are going to use 1000 graphs for training, 100 graphs for validation and 100 graphs for test.**
* **We will use ‘0: mu’, first property, as a label for the regression task.**
* You will get the Data-Loaded in the code notebook.
* This is the Regression task so you have to take one label for every graph.

## **Part (A): [3 marks]**

Use the library implementation of following shallow embedding methods to generate the node embeddings and then compute the graph features by averaging all the node features.

* DeepWalk (embedding\_dimensions= 64, walk\_length=10, num\_walks=50)
* Node2Vec (embedding\_dimensions= 64, walk\_length=10, num\_walks=50, p=1, q=0.5)
* Implement a custom Struc2Vec to generate the embeddings of dimensions= 64, walk\_length=10 and num\_walks=50.

**Now, implement a custom Deep Neural Network for the regression task. [Every graph has one embedding and corresponding label to be predicted]**

Report the following:

* RMSE Metric for each of the methods in the test set.

## **Part (B): [6 marks]**

Graph Convolutional Network (GCN) with Node Features:

* GCN Layer you have to implement:

where is the adjacency matrix with added self-loops, is the degree matrix, is the node feature matrix at layer lll, and is the weight matrix.

* **Task:**
  + Implement a Graph Convolutional Network (GCN) using the original node features.
  + You can try out various aggregators like ‘sum’, ‘mean’ etc to get graph features at the end.
  + Show the effect of GCN layers into the learning [use upto 4 GCN layers].
  + Perform Regression on the test set and report RMSE.

## **Part (C): [6 marks]**

Message Passing Neural Network (MPNN) with node and edge Features:

* MPNN you have to implement, here is the full formation of the MPNN layer.
  + Message function: , where is a differentiable message function (e.g., an MLP), are the features of node i,j and represent the features of the edge connecting nodes i and j.
  + Message aggregation:
  + Node Update function: where is a parametrized neural network.
* Task:
  + Implement the given MPNN using original Node and Edge features.
  + You can try out various aggregators like ‘sum’, ‘mean’ etc to get graph features at the end.
  + Show the effect of MPNN layers into the learning [use upto 4 MPNN layers].
  + Perform Regression on the test set and report RMSE.

## **Part (D): [6 marks]**

Attention Mechanism in GNN (EGATConv):

* You have to implement an attention based GNN as given by the following equations.
  + Attention Mechanism: , where and are the learnable weight metrics , is a learnable attention function (e.g., an MLP) and || is the concatenation operator.
  + Normalised attention coefficient:
  + Node Update:
* Task:
  + Implement an attention-based GNN, incorporating the concepts of the Edge-Weighted Graph Attention Network (EGATConv)
  + You can try out various aggregators like ‘sum’, ‘mean’ etc to get graph features at the end.
  + Show the effect of EGATConv layers into the learning [use upto 4 layers].
  + Perform Regression on the test set and report RMSE.