MADS-16 Prototyping with Deep-Learning
Latent Representation Learning of Molecular Graph Structures using Graph Auto-Encoder

Max Sinner
Master of Data Science
University Of Luxembourg

Abstract

This project will give an introduction to Graph Auto-Encoders and their application to molecular graphs. The initial goal will be to use the provided QM9 data-set in Pytorch Geometric and translate the molecular graphs into a latent vector representation using a GAE architecture. After successful encoding, a MLP architecture will be used to learn the relation between the HOMO-LUMO gap and the molecular latent representation. A short discussion on improvements further research projects will be given.

Keywords: GAE, Molecular Generation, Latent Representation Regression Task

1 Introduction

The goal of this project is to make predictions of physical molecular properties, such as the HOMO-LUMO-gap from an encoded representation, which will be a useful stepping stone for molecular generation. This could be useful to find molecules or design molecules that have a specific property, such as chemical compounds used in the pharmaceutical industry. This novel approach to find suitable molecules for diverse applications (e.g pharmaceutical industry) allows to explore a vast array of options without a huge computational cost (as opposed to quantum mechanical simulations) and financial costs (real molecular design using chemical reactions). Due to these advantages, the exploration process is highly scaleable. As this method is not restricted to small molecules, this concept can be extended to crystalline structures, which will be explored in further works. However, we will limit ourselves to the encoding and the regression tasks, which consists of passing the graph molecular data to an Graph Auto-Encoder (GAE) and using the encoded latent representation to make a prediction on the HOMO-LUMO gap for every molecule. The HOMO-LUMO gap is the energy gap between the Highest Occupied Molecular Orbit (HOMO) and the Lowest Unoccupied Molecular Orbit (LUMO). If this can be implemented successfully, one might turn to

using a VGAE to sample new latent representations of molecules, and if the HOMO-LUMO gap has an interesting value, then one can decode those latent representation in order to analyze the molecular structure and properties.

2 Related Work

The primary source for GAE or VGAE can be traced back to the works of Kipf and Welling (2016) [14], as they provide the definitions and theoretical basis for using the concepts of Auto-Encoders in the domain of graph data. The readily provided GAE architecture from Kipf et al. will be used to encode the molecular graphs from the QM9 data-set into into an appropriate latent representation. More specific details can be found in the Pytorch Geometric manual about the GCNConv layers used in the Encoder part of the GAE.
The use of Graph Neural Networks in molecular generation and predicting material properties was first demonstrated by Xie et al. in 2018 [1]. The main advancement of this article is the computational speed advantage as compared to Density-Functional Theory (DFT) calculations for predicted values of material properties, as well as predictive advantages over Statistical Learning frameworks (multivariate local regression) on the same data-set (materialsproject-database).
Also in 2018, Liu et al [9] proposed GVAE to learn "to generate graphs that conform to a distribution observed in training data". This lead to successful generation of molecules that displayed optimal physical properties after reshaping the latent representations appropriately. However, one of the key shortcomings of the project was the lack of accurate metrics that can assess if the generated molecules could be useful in real-world applications.
This project tries to compete with a tutorial on Graph Neural Networks in the book [17], which treats the direct path of predicting the HOMO-LUMO gap from the molecular graphs.This however comes with the drawback that one cannot sample new structures using a simple Graph Neural Network as one does with a VGAE. The goal will be to compare the performance of the combined approach of GAE (autoencoding) and MLP(regression) with the GCNN direct regression task.
The implemetation of the QM9 data-set built in the PyTorch Geometric data library can be traced back to two papers; one from Gilmer et al. [15] and one from Wu et al. [16]. The former treats a similar task as the paper by Xie et al.[1], by providing a fast track to predicting molecular properties using neural networks instead of DFT computations (10^3 for DFT vs 10^{-2} for Message Passing Neural Nets (MPNN)). The breakthrough of the research lies in the combination of prior implemented models that learned message passing algorithms, which lead to state of the art results in many benchmarks in quantum chemistry. This general MPNN method, according to the authors, should be applied to larger data-sets with more precise ground truth descriptions/labels. The latter demonstrates that learnable representations are powerful tools for molecular machine learning and generally

deliver best performance by constructing a new benchmark for many new machine learning algorithms to check whether they perform well in their designated tasks.

3 Materials and Methods

3.1 Material

The data used to predict molecular properties is the QM9 data-set of (small) organic molecules, which stems from the larger "Chemical Universe Database GDB-17". It contains molecules with the following elements: H-C-O-N-F, with up to 9 heavy atoms (non-hydrogen atoms, up to 29 atoms in total), and is provided in the internal library of PyTorch Geometric. In the data-set, there are 130,831 molecular graphs, being presented in the following format:

```
Data(x=[5, 11], edge_index=[2, 8],
edge_attr=[8, 4],
y=[1, 19],
pos=[5, 3],
idx=[1],
name='gdb_1',
z=[5])
```

One can see that the data has a feature vector **x**, a adjacency matrix **edge-index**, edge features in **edge-attr**, regression targets in **y**, along with atomic coordinates (**pos**) and the atomic numbers in **z**.

The features for the vertices stem from the Message Passing paper [15], while the four different bond types (single, double, triple, aromatic), which function as the edge attributes, stem from the MoleculeNet paper [16].

Table 1. Atom Features

Feature	Description
Atom type	H, C, N, O, F (one-hot)
Atomic number	Number of protons (integer)
Acceptor	Accepts electrons (binary)
Donor	Donates electrons (binary)
Aromatic	In an aromatic system (binary)
Hybridization	sp, sp2, sp3 (one-hot or null)
Number of Hydrogens	(integer)

Figure 1: From Gilmer *et al* "Neural Message Passing for Quantum Chemistry"

In our project, the target property will be the HOMO-LUMO-gap. This quantity can for instance describe stability and strength of a compound, or give information about a substances colour in solution.

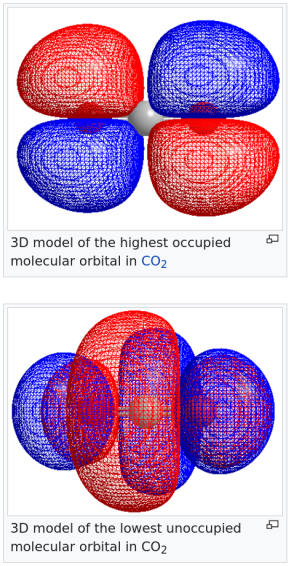

Figure 2: Example of HOMO-LUMO orbitals in CO_2

As already alluded to in the article of Gilmer et al. [15], there has been some work done on message passing neural networks in association to the data-set, which demonstrates the computational speed advantages of GNN over DFT calculations. As the data-set has been readily provided by the package, there has not been additional processing procedures applied to it.

3.2 Methods

3.2.1 *Network architecture 1: Get Latent Representation of Molecular Graphs*

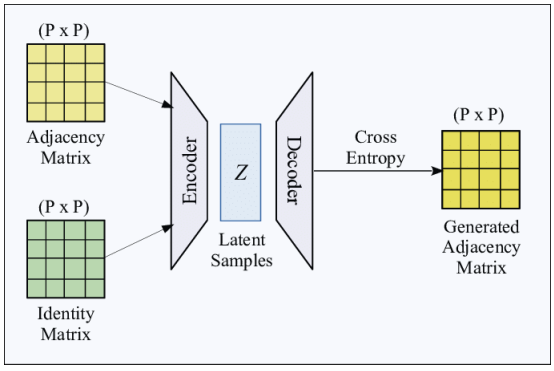

Figure 3: Illustration of a Graph Auto-encoder

As seen in Figure 3, a GAE will be used to encode the molecular graphs into a latent representation. In our case, we will use a simple two-layer Graph-Auto-Encoder architecture with two convolutional layers (which uses the default "add" aggregation method) in the encoder part to a latent representation Z, as demonstrated in the paper from Kipf et al. [14] . From the data, we provide the feature vector (containing 11 features for each atom in the molecule) and the adjacency matrix (also encodes the bonds between the the atoms). The information of the feature vector will be reduced from $(n_{atoms}, 11)$-dimensional to $(n_{atoms}, 1)$-dimensional vector form, with knowledge of the adjacency matrix. In order to assess the encoding procedure, we use the binary-cross-entropy as measure to compare "positive nodes" (nodes which are present in the original graphs) against "negative nodes", nodes which are originally not present in the molecular graphs. As the data-set itself is not very huge, the network is trained for 21 epochs. As batch size, we can choose the whole training data-set, as this gives the best convergence without fluctuation of the loss across the epochs. Here is the used architecture:

Graph Auto-Encoder Architecture:

```
GAE(
  (encoder): GCNEncoder(
    (conv1): GCNConv(11, 12)
    (conv2): GCNConv(12, 1)
  )
  (decoder): InnerProductDecoder()
)
```

More information on the GCNConv-layers and the InnerProductDecoder can be found in the PyTorch Geometric "GNN Cheatsheet" Manual [18]. In order to prepare the latent representation data to pass it to the next task, we collect the latent representations for the individual vertices (from 11-dimensional to 1 dimensional) are aggregated in arrays as in the original feature vectors of the molecular graphs. These are then converted to torch.Tensor-objects. As targets, the associated HL-gaps are collected in a torch.Tensor-object, and convert the combination of the latent-representation-vectors and energy-gaps to a torch.Dataset-object. This can easily be passed to a NN architecture build using the PyTorch-library.

3.2.2 *Network architecture 2: Regression task to get HOMO- LUMO-gap*

As opposed to using a Graph Convolutional Neural Network to predict the HL-gap directly (as provided in the book of Raschka [17]), the information for the graphs is encoded into a latent representation, which will be passed to a Multi-Layer-Perceptron (MLP) to predict the HL-gap. As input size, we will select the molecule with most atoms, which in this data-set will be 29 atoms. For molecules with less atoms, the entries of the latent representation are zero-padded to reach a length of 29, to have constant input shape for the MLP architecture. As activation function, ReLU will be used in the hidden layers, with a linear function between the second-to-last (64 nodes) and the last layer (1 node). In order to evaluate our training and testing, the MSE loss fucntion is selected.

Multi-Layer-Perceptron Architecture

```
MLP(
  (layers): Sequential(
    (0): Linear(in_features=29, out_features=64, bias=True)
    (1): ReLU()
    (2): Linear(in_features=64, out_features=64, bias=True)
    (3): ReLU()
    (4): Linear(in_features=64, out_features=64, bias=True)
    (5): ReLU()
    (6): Linear(in_features=64, out_features=1, bias=True)
  )
)
```

4 Results and Discussion

This section covers the findings of both research tasks of the project, and discusses the strong suits and shortcomings.

4.1 GAE architecture

As provided above, the GAE architecture was trained with a batch size of the full training set for 10 epochs, which brought very satisfying results. For the training set error,a value of $L^{Train}_{Cross-Loss} = 0.0000120$ was achieved, while the validation loss had a value of $L^{Val}_{Cross-Loss} = 0.0001212$. This implies that the decoder is able to reconstruct the adjacency matrices of the input data in reliable fashion

Both losses can be seen in Figures 4 and 5, where one observes a steady decline in training and validation losses. The overall loss is normalized by the number of graphs in the training and validation sets.

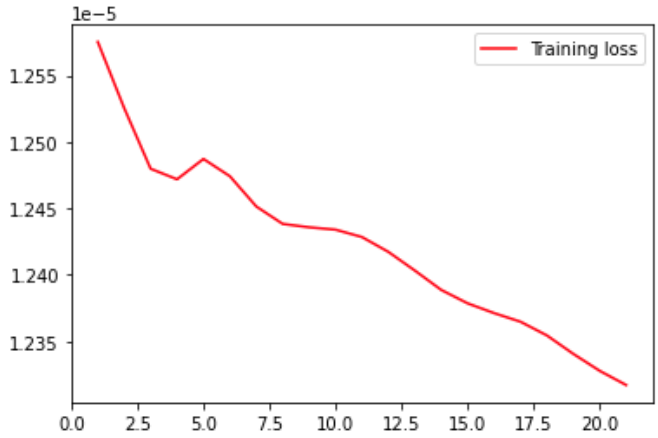

Figure 4: Train loss over epochs

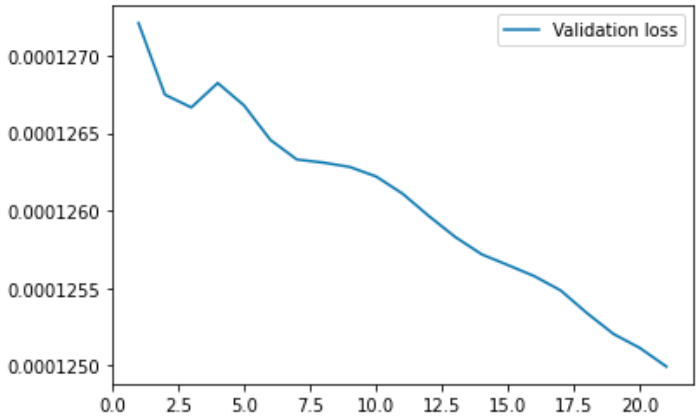

Figure 5: Validation loss over epochs

4.2 MLP architecture

Although the encoding seems to be working successfully, the current MLP model is not able to learn the association between the latent representation and its associated HOMO-LUMO gap. After training for 5000 epochs, the results converged successfully for an accumulated MSE error of $\sim 1,7/110k$ graphs $= 0.0000154$. Although this to be very low in terms

of numerical values for training loss, the graph comparing the real HL-gaps against the predicted HL-gaps reveals a systematic issue:

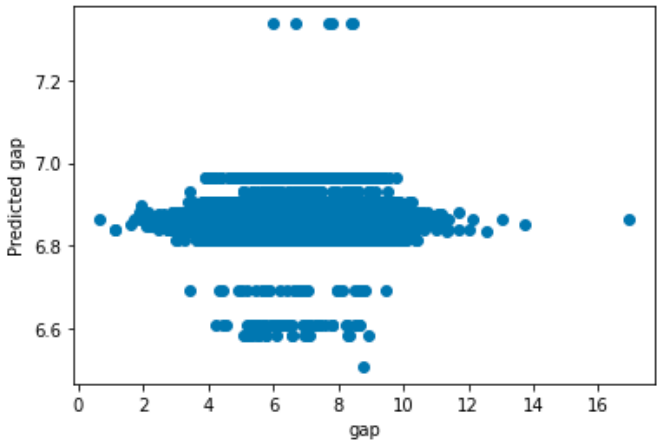

Figure 6: Real vs Predicted HOMO-LUMO gaps

This figure implies that the learner is unable to differentiate between some of the molecules, as a lot of the predictions are plotted on the same horizontal lines. There might be multiple reasons for this: first, some crucial information in the feature vectors might be missing. The original QM9 data-set contains information on the Mulliken Particle charges for each atom in the molecule, which might be a crucial identifier for a molecule. Secondly, maybe some information is lost when passing from features of dimension $(n_atoms, 11)$ to $(n_atoms, 1)$. Last but not least, maybe the selection of a MLP architecture of the present type is not sufficient to learn the relation between the latent molecular representation and the associated HL-gaps.

5 Conclusion

The project leaves us with the following takeaways: it is possible to encode molecular graph data into latent vector representations with high accuracy. Using a relatively simply GAE structure (presented in [15]), even a low number of training epochs leads to great performance for the validation set ($L^{Val}_{Cross-Loss} = 0.0001212$). On the contrary, the currently applied MLP architecture is not able to successfully predict HL-gaps from latent vector representations, which might be due to lack of crucial information in the molecular graphs, information loss in the encoders bottleneck for this particular regression task or selection of the wrong architecture for the task. In the future, some more physical properties will be added to the molecular graphs in the data-set to check whether this impacts the performance of the MLP architecture in the regression task. If carried out successfully, similar

applications in the domain of crystal graphs can be carried out.

Acknowledgements

I would like to thank Professor Luis LEIVA, Dr. Bereket YILMA and most importantly Professor Aldo ROMERO for inspiration, extensive discussions and guidance throughout the exciting and instructive project.

References

[1] Tian Xie and Jeffrey C. Grossman, Crystal Graph Convolutional Neural Networks for an Accurate and Interpretable Prediction of Material Properties, Phys. Rev. Lett. 120, 145301 (2018)

[2] Jonggul Lee et al 'Descriptors of atoms and structure information for predicting properties of crystalline materials', Mater. Res. Express 8 026302 (2021)

[3] Hoja, J., Medrano Sandonas, L., Ernst, B.G. et al.'QM7-X, a comprehensive data-set of quantum-mechanical properties spanning the chemical space of small organic molecules' . Sci Data 8, 43 (2021)

[4] A. Jain*, S.P. Ong*, G. Hautier, W. Chen, W.D. Richards, S. Dacek, S. Cholia, D. Gunter, D. Skinner, G. Ceder, K.A. Persson (*=equal contributions) 'The Materials Project: A materials genome approach to accelerating materials innovation' APL Materials, 2013, 1(1), 011002.

[5] Bandgap Engineering in the Configurational Space of Solid Solutions via Machine Learning: (Mg,Zn)O Case Study, Scott D. Midgley, Said Hamad, Keith T. Butler, and Ricardo Grau-Crespo The Journal of Physical Chemistry Letters 2021 12 (21), 5163-5168

[6] Xiang Li, Shaowu Ning, Zhanli Liu, Ziming Yan, Chengcheng Luo, Zhuo Zhuang, 'Designing phononic crystal with anticipated band gap through a deep learning based data-driven method', Computer Methods in Applied Mechanics and Engineering,Volume 361,2020,112737

[7] F., Gori, M., Tsoi, A., Hagenbuchner, M. Monfardini, G. 'The graph neural network model'2009,, IEEE Transactions on Neural Networks, vol. 20, no. 1, pp. 61-80.

[8] A band-gap database for semiconducting inorganic materials calculated with hybrid functional Kim, Sangtae; Lee,, Miso; Hong, Changho; Yoon, Youngchae; An, Hyungmin; Lee, Dongheon; et al. (2020).: data-set. https://doi.org/10.6084/m9.figshare.12839240.v5

[9] Liu, Qi, Miltiadis Allamanis, Marc Brockschmidt, and Alexander Gaunt. "Constrained graph variational autoencoders for molecule design." Advances in neural information processing systems 31 (2018).

[10] Rigoni, Davide, Nicolò Navarin, and Alessandro Sperduti. "Conditional constrained graph variational autoencoders for molecule design." In 2020 IEEE Symposium Series on

Computational Intelligence (SSCI), pp. 729-736. IEEE, 2020.
[11] Ward, Logan, Alexander Dunn, Alireza Faghaninia, Nils ER Zimmermann, Saurabh Bajaj, Qi Wang, Joseph Montoya et al. "Matminer: An open source toolkit for materials data mining." Computational Materials Science 152 (2018): 60-69.
[12] Fey, Matthias, and Jan Eric Lenssen. "Fast graph representation learning with PyTorch Geometric." arXiv preprint arXiv:1903.02428 (2019).
[13] Weng, M., Wang, Z., Qian, G. et al. Identify crystal structures by a new paradigm based on graph theory for building materials big data. Sci. China Chem. 62, 982–986 (2019).
[14] Thomas, N.K. and Welling, M. Variational graph auto-encoders. In Neural Information Processing Systems Workshop on Bayesian Deep Learning. (2016)
[15] J Gilmer, SS Schoenholz, PF Riley, O Vinyals, GE Dahl Neural Message Passing for Quantum Chemistry. International conference on machine learning, (2017)
[16] Wu Z, Ramsundar B, Feinberg EN, Gomes J, Geniesse C, Pappu AS, Leswing K, Pande V. MoleculeNet: a benchmark for molecular machine learning. Chem Sci. (2017)
[17] Raschka S., Liu Y., Mirjalili V.Machine Learning With PyTorch and Scikit-Learn, 3rd Edition Packt Publishing Ltd. (2019)
[18] Fey, M., Lenssen, J. E. (2019). Fast Graph Representation Learning with PyTorch Geometric [Computer software]. https://github.com/pyg-team/pytorch$_geometric$