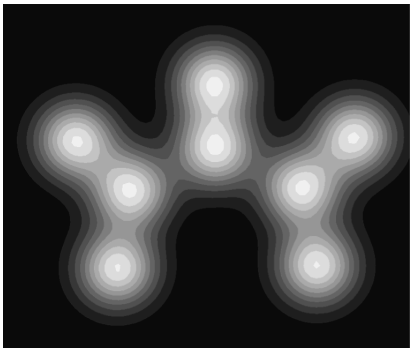
DU Big Data

Scientific project

Project report

Prediction of molecular properties from Neural networks



Weicheng HE

Tutor: Dario Rocca

5 February, 2021

1. Introduction

In the last few years, we have seen the transformative impact of deep learning in many applications, like speech recognition, natural language processing and computer vision. At the same time, researcher also began to take advances in machine learning and artificial intelligence and apply them to accelerate progress in natural science. For exemple, there was more and more success in applying machine learning and deep learning in the research of quantum chemistry and computational materials science.

Unlike other machine learning algorithms, including those used by past and current computational chemistry applications, deep learning distinguishes itself in the use of a hierarchical cascade of non-linear functions. This allows it to learn representations and extract necessary features from its input data to predict the desired property of interest.

The powerful representation learning ability of deep learning brings a lot of potential to computer vision. Before the birth of deep learning, researchers in computer vision invested substantial efforts in developing appropriate features, which are not always efficient. Nowadays, deep learning models have mostly replaced such expert-driven development by developing automatically their own set of internal features. In certain tasks of computer vision, deep learning has exceeded human-level accuracy. But deep learning is still at an early age of development so it can not replace human creativity or intelligence in the scientific research process.

Nevertheless, there is no doubt that deep learning has demonstrated that it outperforms human in task-specific applications that go beyond computer vision, for instance, Alpha Go and Google Neural Machine Translation. One of key evolution by deep learning is that what used to be in the domain of human experts, notably feature engineering, has

been to a large extent replaced by the representation learning ability of deep neural networks.

Predicting the properties of a molecule from its structure is a challenging task. Traditional studies based on density functional theory (DFT) in physics are proved to be time-consuming, especially for predicting large number of molecules. Thanks to its great representation learning ability, deep learning seems to be a promising tool for predicting molecular properties accurately with low cost.

In this study, we propose a method for molecular property prediction by image recognition based on neural networks. More precisely, we use data in the form of 2D image of molecules to predict the internal energies. The 2D images are created from geometric information of 4237 planar molecules by the gaussian representation method, which requires a minimal amount of chemical knowledge. Without heavy human-expert feature engineering in specific computational chemistry applications, these deep learning methods achieved a reasonable performance in energy prediction task. In addition, this model can be easily implemented and generalized to other molecular properties prediction tasks.

2. Methods

2.1 Dataset

In order to get massive molecular information, we use QM9 dataset, which is perhaps the most well-known benchmark dataset containing 134 thousand molecules at equilibrium geometry with up to 9 heavy atoms. It provides geometric, energetic, electronic and thermodynamic properties. All of the relaxed geometries and properties for all the 134 thousand molecules are calculated by DFT. In this study, particularly, we focus on planar molecules, which are easier to be represented in 2D images. In total, the QM9 dataset includes 4237 planar molecules, which contain C, H, O, N and F atoms.

2.2 Data preparation

Data preparation was performed using Python, using Numpy for mathematical computation, Matplotlib for data visualization and Pandas for table data manipulation. First of all, we created a table by extracting geometric information from QM9 dataset which are in format of xyz files.

	molecule_name	atom_index	atom	x	y	z
51818	dsgdb9nsd_133447	9	O	-1.088836	0.647870	0.000029
51819	dsgdb9nsd_133448	1	F	1.828669	1.458414	0.000267
51820	dsgdb9nsd_133448	2	C	0.767275	0.687483	0.000267
51821	dsgdb9nsd_133448	3	C	0.838157	-0.652018	0.000267
51822	dsgdb9nsd_133448	4	C	-0.417279	-1.343414	0.000038
51823	dsgdb9nsd_133448	5	F	-0.372129	-2.663751	-0.000164
51824	dsgdb9nsd_133448	6	N	-1.588909	-0.823600	-0.000379
51825	dsgdb9nsd_133448	7	O	-1.632641	0.565572	-0.000346
51826	dsgdb9nsd_133448	8	C	-0.524615	1.390907	0.000417
51827	dsgdb9nsd_133448	9	O	-0.671369	2.578580	-0.000697

TABLE 1: ATOMIC STRUCTURE INFORMATION EXTRACTED FROM QM9

Then in order to describe the molecular internal structure, we represent each molecule in an image by gaussian representation given by:

$$V(x,y) = \sum_{i=1}^{N} Z_i exp(-\frac{[(x-x_i)^2 + (y-y_i)^2]}{2\gamma^2})$$

where x_i, y_i are the coordinates of atom i with atomic number Z_i. For the two dimensional images, the z coordinate is not considered. The γ parameter tunes the width of the Gaussian peaks, which impacts how the molecules are represented in the 2D images.

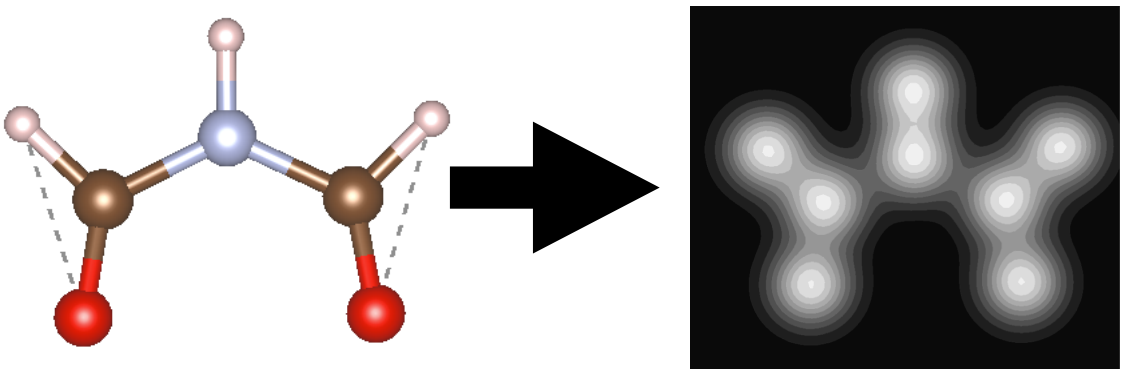


FIGURE 1: MOLECULE IMAGE BY GAUSSIAN REPRESENTATION

At the beginning, in order to better understand the effects of different parameters on the molecular representation, we tried to create images with different parameters as shown below:

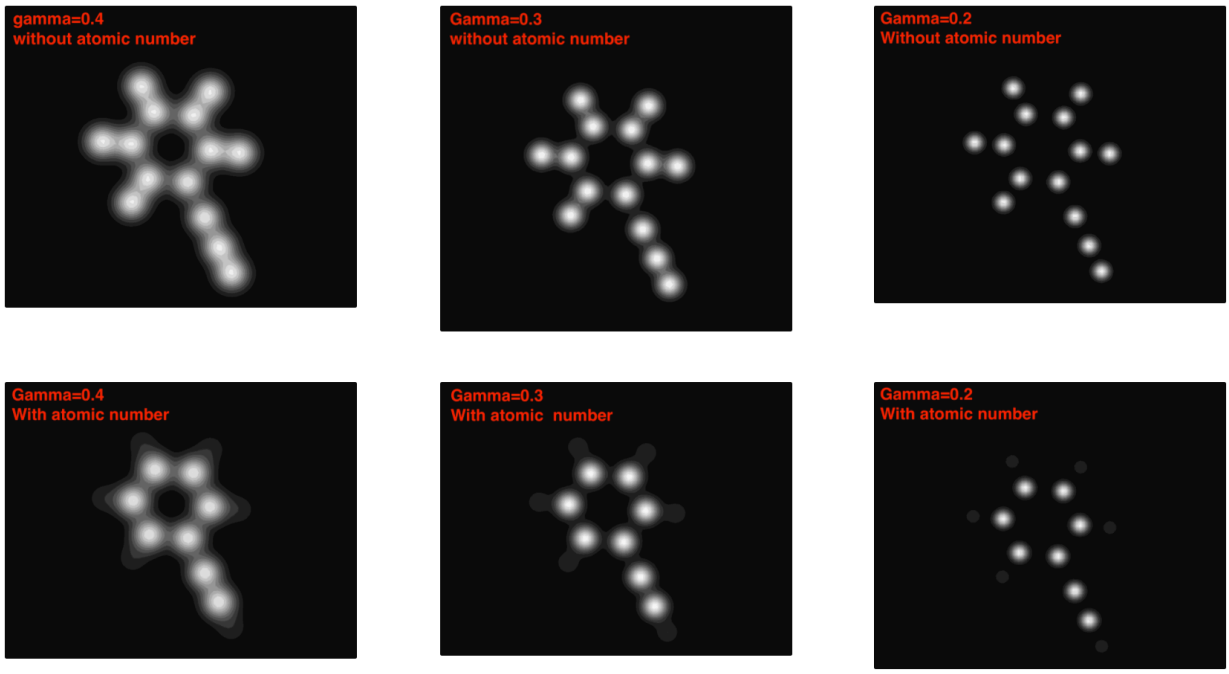


FIGURE 2: IMAGES CREATED WITH DIFFERENT PARAMETERS

From the images created, we can see that the images with $\gamma = 0.2\text{Å}$ can show the positions of atoms more clearly. So at the end, we chose $\gamma = 0.2\text{Å}$ as the width of the Gaussian peaks, which is consistent with the study of Kevin Ryczko et al. Concerning the atomic number, we didn't think it would have a big impact in this prediction task because the dataset QM9 only contains five elements: CHONF, whose atomic numbers are quite close. So in this case, we ignored the atomic number in the Gaussian representation. After creating the molecular images as features, we also extracted internal energy (in Hartree) from the QM9 dataset as labels to train the deep learning models.

2.3 Multilayer perceptron (MLP)

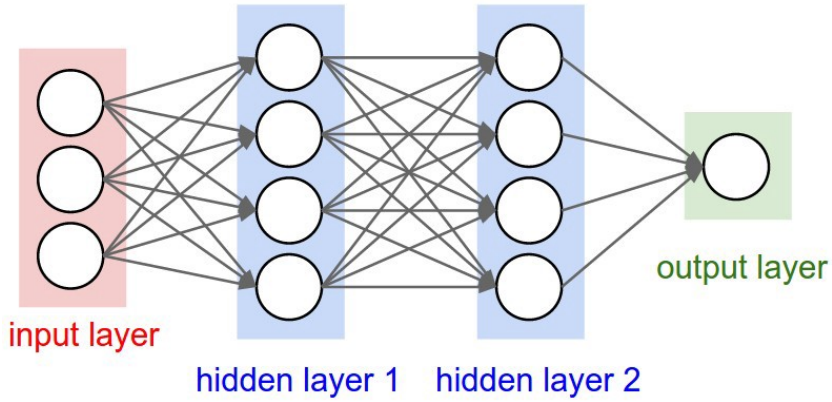


FIGURE 3: MLP ARCHITECTURE

With the data prepared, we decided to try a simple neural network model as a baseline: multilayer perceptron (MLP). MLPs are a class of feedforward artificial neural network, consisting of at least three layers of nodes: an input layer, a hidden layer and output layer. Except for the input nodes, each node is a neuron that uses a nonlinear activation function. MLP utilizes a supervised learning technique called back propagation for training. Its multiple layers and non-linear activation distinguish MLP from a linear perceptron. It can distinguish data that is not linearly separable.

MLP used to be applied in compter vision, now succeeded by Convolutional Neural Network (CNN). MLP is now deemed insufficient for modern advanced computer vision

tasks. It has the characteristic of fully connected layers, where each perception is connected with every other perceptron. As a result, the number of total parameters can grow to become very high. Accordingly, there is redundancy in high dimensions. In addition, it disregards spatial information. For example, in our case, we have molecular images as input, but we can't train the MLP model directly with images because it takes flattened vectors as inputs. So before the training, we need to convert images to one-dimensional vectors, which does not represent the spatial information.

2.4 Convolutional Neural Network(CNN)

Since this prediction model is based on computer vision, the convolutional neural network model is inevitable. The CNN model is current the favorite of computer vision algorithms, winner of multiple ImageNet competitions. The CNN model was designed to map image data to an output variable, which fits exactly our needs.

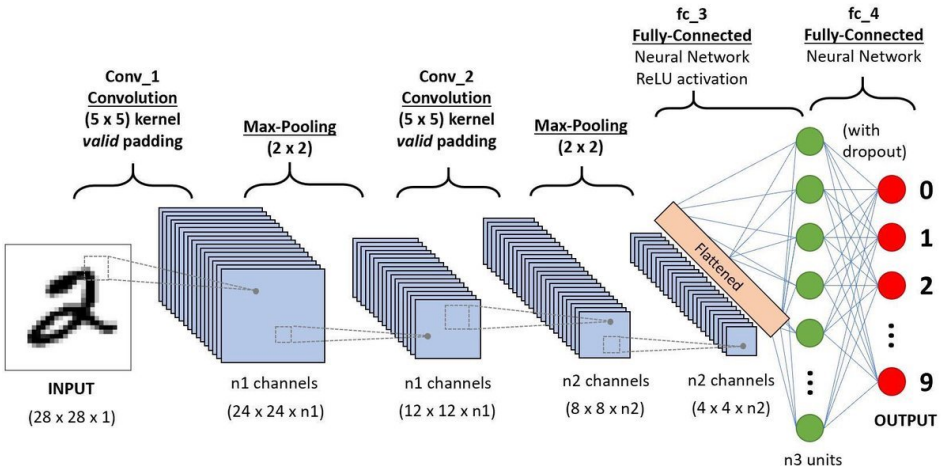


FIGURE 4: CNN ARCHITECTURE

In a CNN, each filter is panned around the entire image according to certain size and stride, which allows the filter to find and match patterns no matter where the pattern is located in a given image. In this way, the model can account for local connectivity. Different from MLP, layers of CNN are sparsely connected rather than fully connected.

Moreover, the panning of filters in CNN essentially allows parameters sharing, weight sharing so that the filter looks for a specific pattern in variant structures in the data.

In this study, we investigated two types of CNN models. The first one is a simple CNN model with in total 6 layes: 3 convolution layers, 3 max pooling layers. The second one is very deep convolutional network: ResNet with 50 layers. After the success of AlexNet, ResNet (Residual Network) may be the most groundbreaking work in the computer vision in the last few years. Since VGG and GoogleNet, the state-of-the-art CNN architecture is going deeper and deeper because it seems to be true that "the deeper the better" in CNN. However, increasing network depth does not simply stacking layers together. Because of the vanishing gradient problem, which means the gradient becomes infinitively small during the back-propagation, the deep networks are hard to train. As a result, its performance gets saturated or even starts degrading rapidly.

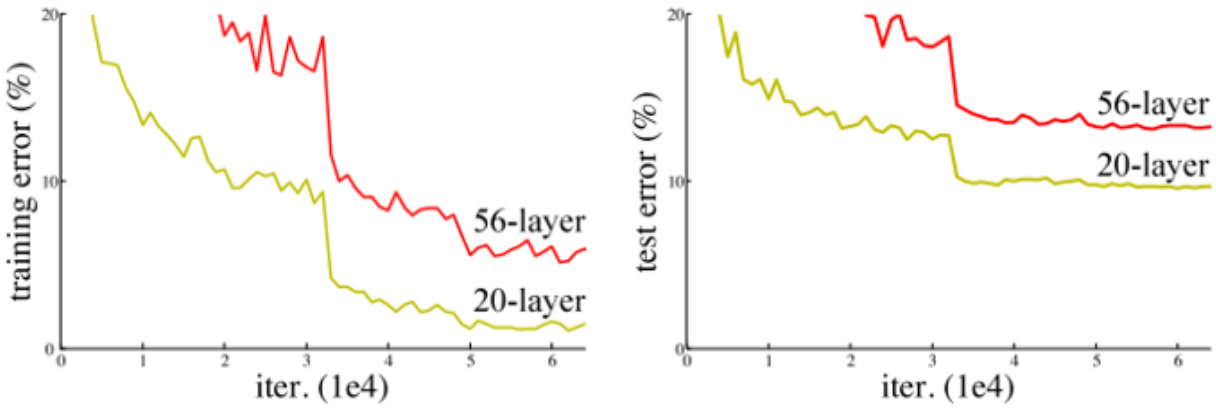


FIGURE 5: VANISHING GRADIENT PROBLEM

ResNet was introduced in 2015 to deal with the vanishing gradient issue. With ResNet, the gradients can flow directly through the skip connections backwards from later layers to initial filters. So the deep ResNet can outperform the other shallow networks without vanishing gradient problem. As ResNet gains more and more popularity in the research community, its architecture is getting studied heavily. Serval variants of different depth

appeared, including ResNet 34, ResNet 50, ResNet 128, etc. However, with the increasing depth, the number of parameters to train is considerable. For example, the ResNet 101 has up to 42 million parameters, which need a lot of computing power. Finally, we chose ResNet 50 which has a good balance between performance and computing power. Since training ResNet is heavy task and very time-consuming, we decided to use GPU to accelerate the training. The Nvidia Tesla P100 GPU provided freely on kaggle.com was used to train our model.

2.5 Graph Convolutional Network (GCN)

Molecules, composed of atoms and bonds, would be typically represented as a graph, in which atoms and bonds can be considered as nodes and edges, respectively. In the recent publications on molecular properties prediction, we can see a trend on the use of graph convolutional networks (GCN), which can learn directly the molecular graph instead of pixels of images. So in this context, we think it would be interesting to try this model on this prediction task even though it's not based on image recognition.

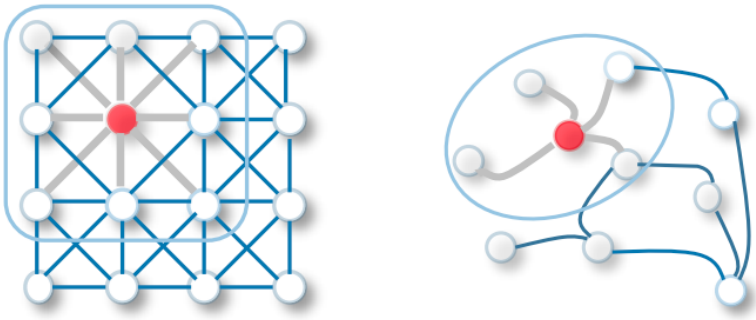


FIGURE 6: FILTERS IN CNN AND GCN

Similar to CNN, GCN does the same 'convolution' operation: multiplying the input neurons with a set of weights that are commonly known as filters or kernels. The model learns the features by inspecting neighboring nodes in a graph structure.

In order to use the GCN model, we need more information on bonds and atoms in addition to atoms coordinates. A molecule can be represented as a graph with atoms as nodes and bonds as edges. For each node, there is a feature vector corresponding to the atom features such as atom type, number of hydrogens attached, etc. In this study, we prepared the molecular graphs by using the DeepChem framework, a framework with aim to democratize deep learning for chemistry and physics. The QM9 dataset is integrated in the DeepChem and it also allows to convert SMILES file (Simplified molecular-input line-entry system), a chemical notation to molecule graph. Moreover, we can also use the GCN algorithm integrated in DeepChem directly to train our model, which is very convient.

2.6 Data augmentation

During the training of CNN models, we performed additional data augmentation to the image using OpenCV package in Python, so as to bolster the limited number of data that we have for this prediction task. Such data augmentation techniques are a common and recommended practice in the computer vison literature. In this case, we augmented the dataset by adding 6/12 regular rotations to each molecular as shown below.

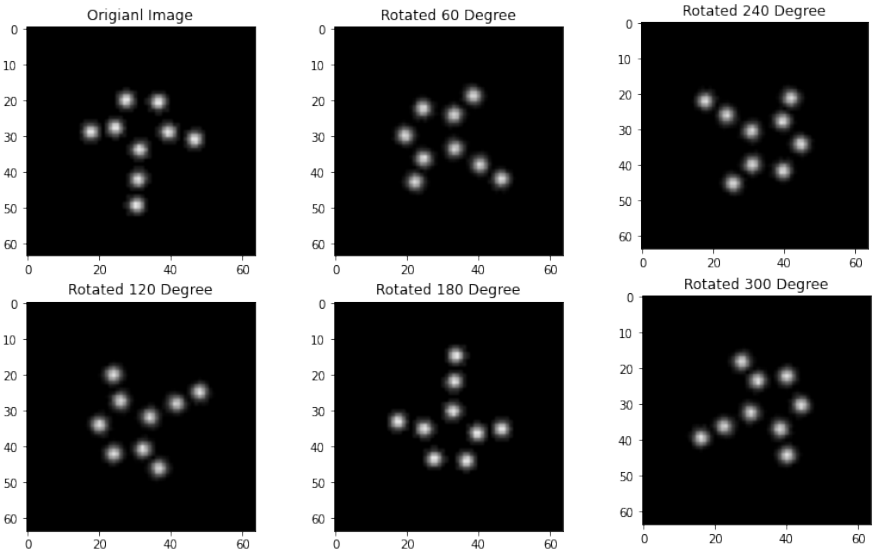


FIGURE 7: DATA AUGMENTATION ON INPUT IMAGES

3. Results and discussion

3.1 MLP

The Multilayer perceptron model is a baseline model in this study. Considering its shallow architecture, we assumed that it wouldn't work well with computer vision task.

The results confirmed this assumption on poor performance of MLP. The RMSE (root mean squared error) in training set and test set are both bigger than 40 Hartree, which is substantial. From the prediction plot, we can see the prediction results are far from the true values.

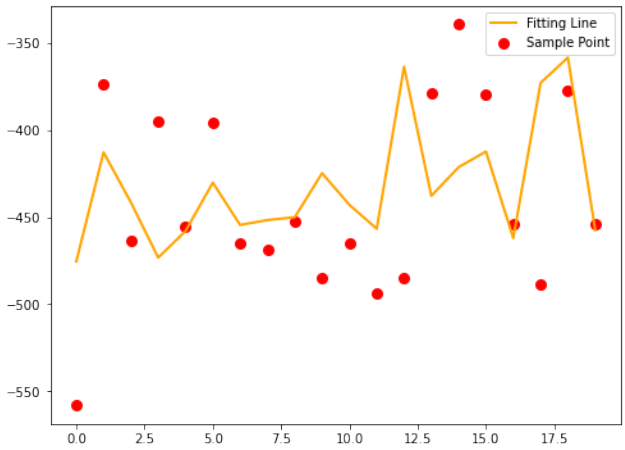


FIGURE 8: PREDICTION PLOT OF MLP

3.2 CNN

Compared to MLP, CNN seems to be more capable for learning features in images. Here, we compared two CNN models with the same number of epochs on the original dataset. The ResNet has 50 layers of nodes, which is much deeper than normal CNN with 6 layers

.

Model	Normal CNN	ResNet50 V2
Number of epochs	50 epochs	50 epochs
Computation time (on GPU)	50 sec	4 min 10 sec
RMSE (training set)	24.8 Hartree	5.5 Hartree
RMSE (test set)	26.3 Hartree	8.0 Hartree

TABLE 2: COMPARISON BETWEEN NORMAL CNN AND RESNET

According to the results, even the normal CNN with only 6 layers of nodes can get a 50% lower training error compared to a MLP model. In addition, we can see that ResNet achieved even better performance with a RMSE of 5.5 Hartree din training set, much lower than normal CNN model, but ResNet is also more time-consuming.

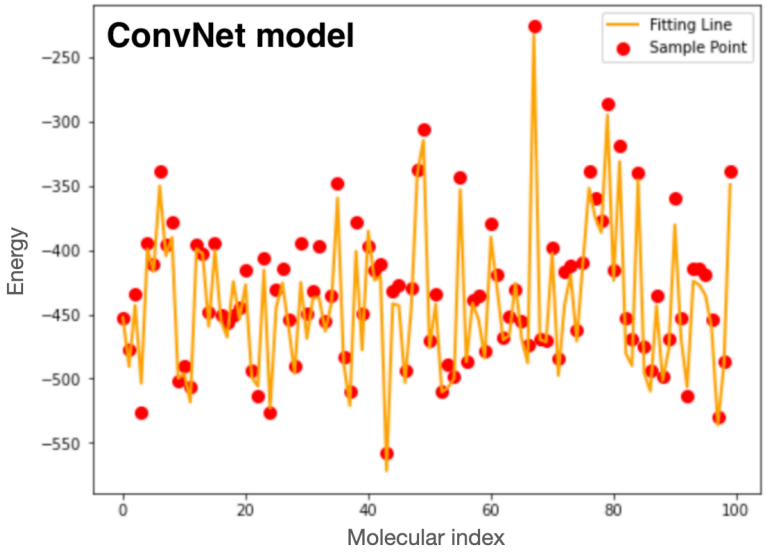


FIGURE 9: PREDICTION PLOT OF RESNET

In the prediction plot, we observed that ResNet can predict most of values in a good direction. However, the RMSE of 5.5 is still very large compared to the state-of-the-art study. We assumed that it's due to the limited dataset size. So we decided to augmented datasets by adding up to 12 rotations to each molecule.

Dataset	Original dataset	Augmented dataset
Input	4237 images	50844 images
Number of epochs	100	100
Computation time (on GPU)	8 min 30 sec	1 h 31 min
RMSE (training set)	5.1 Hartree	1.1 Hartree
RMSE (test set)	7.9 Hartree	5.2 Hartree

TABLE 3: COMPARISON BETWEEN ORIGINAL DATASET AND AUGMENTED DATASET

With the augmented dataset, the ResNet achieved a RMSE of 1.1, which is 80% lower than the training with original dataset. However, the heavy task increased considerably the computation time, 10 times longer than before. Moreover, in the learning curve, we

can detect the overfitting of training. We assumed that we could deal with this problem by reducing the depth of model or adding regularization in the further study.

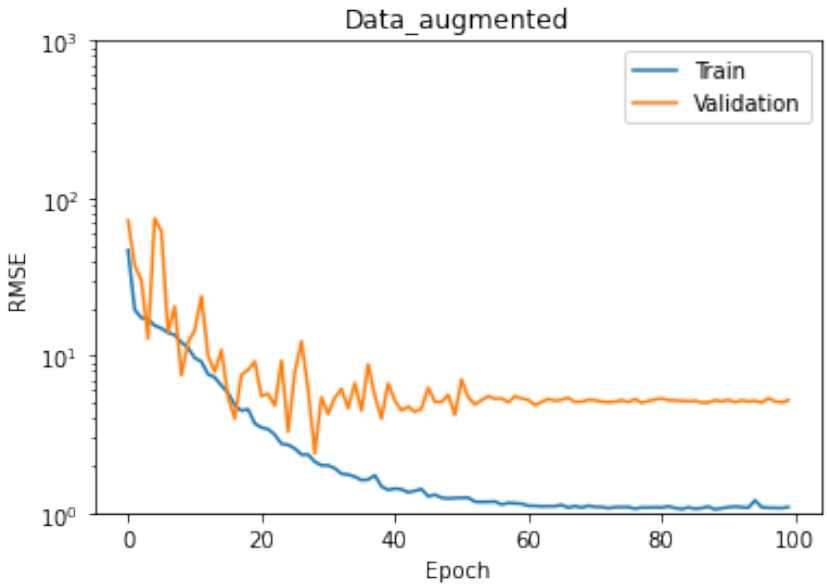


FIGURE 10: LEARNING CURVE OF TRAINING ON AUGMENTED DATASET

3.3 GCN

By CNN models, we get a better prediction results compared to MLP, but the RMSE is still far from the ideal level. So we tried the state-of -the-art model GCN to see if we can improve the results. Since in GCN, we train models with graphs instead of images, we cannot use the augmented dataset.

Model	ResNet V2 (CNN)	Graph ConvNet
Input	50844 images	4237 graphs
Number of epochs	100	1000
Computation time (on GPU)	1 h 31 min	32 min 51 sec
RMSE (training set)	1.1 Hartree	1.3 Hartree
RMSE (test set)	5.2 Hartree	7.45 Hartree

TABLE 4: COMPARISON BETWEEN RESNET AND GCN

As a result, the dataset size for GCN is limited to 4237 graphs. According to the results, in terms of RMSE, the GCN got almost the same level as ResNet on augmented dataset.

Besides, training on GCN was 2/3 faster than ResNet. Therefore, we can consider the GCN is more efficient on this prediction task.

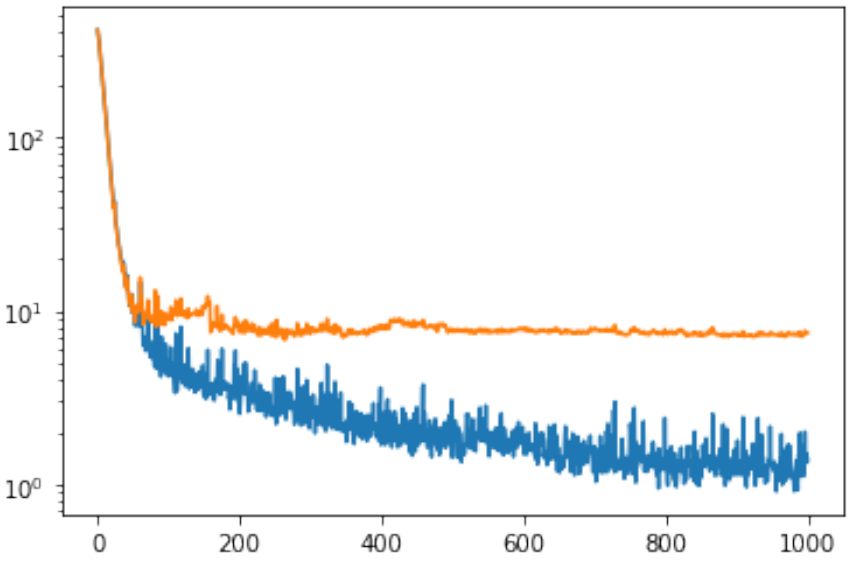


FIGURE 11: LEARNING CURVE OF GCN

However, even after 1000 epochs of training, the performance of GCN on this dataset is still limited compared to the benchmark. Besides, in learning curve, the overfitting can also be detected. We assumed that the limited dataset size may account for the mediocre performance and overfitting problem.

4. conclusion

In this study, we have introduced the emerging techniques in molecular property prediction, including CNN and GCN. We proved that the new technique outperforms the traditional methods like MLP significantly. In order to improve the prediction performance and solve the overfitting problem, we also did data augmentation by adding serval rotations to each molecule. By increasing 12 times the dataset size, we have reduced the RMSE by 80% on ResNet. Even though GCN didn't outperform the ResNet as we expected, it achieved the same level of performance as ResNet with only 1/3 of training time. Due to limited dataset size and other unknown issues, our prediction performance is still far from the benchmark. In the further study, we could improve CNN model by implementing a rotation-invariant algorithm to better learn features of molecules in different orientations. As for GCN, recent study shows that by merging with the attention mechanism, another influential idea of deep learning, GCN can learn the interaction effect between substructures, which could be very useful to enhance the performance of GCN for molecular property prediction.

References

[1]Goh, Garrett B., et al. "Chemception: a deep neural network with minimal chemistry knowledge matches the performance of expert-developed QSAR/QSPR models." *arXiv preprint arXiv:1706.06689* (2017).

[2]Lu C, Liu Q, Wang C, et al. Molecular property prediction: A multilevel quantum interactions modeling perspective[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2019, 33(01): 1052-1060.

[3]Ryczko K, Mills K, Luchak I, et al. Convolutional neural networks for atomistic systems[J]. Computational Materials Science, 2018, 149: 134-142.

[4] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.

Appendix

1. MLP, CNN models training notebook: https://www.kaggle.com/hugo1995/projet-recherche
2. GCN training notebook: https://colab.research.google.com/drive/1g1ett721e8i38yl-RYsVrv8QpbuNJNzE8?usp=sharing