Which hyperparameters to optimise?

Which Hyperparameters to Optimise? An Investigation of Evolutionary Hyperparameter Optimisation in Graph Neural Network for Molecular Property Prediction

Yingfang Yuan
School of Mathematical and Computer Sciences
Heriot-Watt University
Edinburgh, United Kingdom
yyy2@hw.ac.uk

Wenjun Wang
School of Mathematical and Computer Sciences
Heriot-Watt University
Edinburgh, United Kingdom
wenjun.wang@hw.ac.uk

Wei Pang*
School of Mathematical and Computer Sciences
Heriot-Watt University
Edinburgh, United Kingdom
w.pang@hw.ac.uk

ABSTRACT

Most GNNs for molecular property prediction are proposed based on the idea of learning the representations for the nodes by aggregating the information of their neighbour nodes in graph layers. Then, the representations can be passed to subsequent task-specific layers to deal with individual downstream tasks. Facing real-world molecular problems, the hyperparameter optimisation for those layers are vital. In this research, we focus on the impact of selecting two types of GNN hyperparameters, those belonging to graph layers and those of task-specific layers, on the performance of GNN for molecular property prediction. In our experiments, we employed a state-of-the-art evolutionary algorithm (i.e., CMA-ES) for HPO. The results reveal that optimising the two types of hyperparameters separately can improve GNNs' performance, but optimising both types of hyperparameters simultaneously will lead to predominant improvements.

CCS CONCEPTS

• **Computing methodologies** → *Neural networks*; **Search methodologies**; • **Applied computing**; • **General and reference** → **Experimentation**;

KEYWORDS

Graph Neural Networks, Molecular Property Prediction, Hyperparameter Optimisation, Evolutionary Computation

ACM Reference Format:
Yingfang Yuan, Wenjun Wang, and Wei Pang. 2021. Which Hyperparameters to Optimise? An Investigation of Evolutionary Hyperparameter Optimisation in Graph Neural Network for Molecular Property Prediction. In *Proceedings of 2021 Genetic and Evolutionary Computation Conference Companion (GECCO '21 Companion).* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

*Corresponding author

GECCO '21 Companion, July 10–14, 2021, Lille, France

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

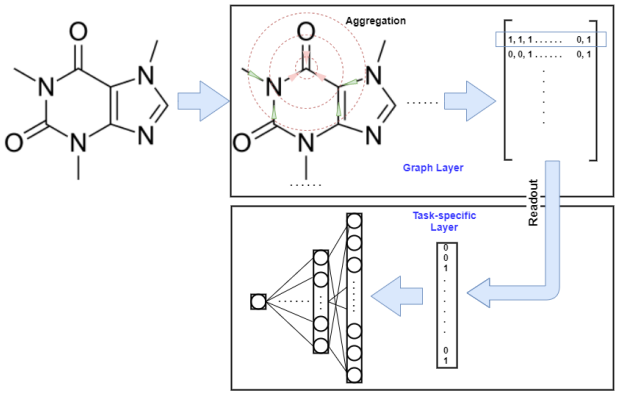

Figure 1: The neural architecture of GNN in molecular property prediction

1 INTRODUCTION

Graph neural networks (GNNs) have been applied to solve a wide range of problems. One advantage of GNNs is that they can be directly operated on graphs in an end-to-end manner for real-world problems, and task-specific representations can be learned automatically between latent layers. The study presented in [5] indicated that hyperparameter optimization (HPO) by evolutionary computation could improve the GNN's performance for predicting molecular properties. In this research, we employed CMA-ES [3] to investigate the differences of optimizing hyperparameters related to graph layers and those from task-specific layers. In this way, we expect to discover that optimizing which types of hyperparameters may bring more expected gains.

2 NEURAL ARCHITECTURE AND HYPERPARAMETERS

In general, facing the problem of molecular property prediction, the architectures of GNNs are classified into graph layer and task-specific layer (Fig. 1). The former denotes those layers that take responsibility for processing the structured data by aggregations, and generate vector representations for all the nodes in graph. Task-specific layers are exploited to deal with individual problems (e.g., classification, regression). In Fig. 1, task-specific layers are implemented by fully-connected layers, which take the molecular representation as input to output a value. The hyperparameters

Types	Hyperparameters	Ranges	Step sizes
Graph layer	n_g	$1-6$	1
	s_g	$32-512$	32
	s_d	$64-1024$	64
Fully-connected layer	n_f	$1-6$	1
	s_f	$64-1024$	64
	a	$sigmoid, relu, tanh$	1

Table 1: Hyperparameters summary

related to these two parts of GNN play different roles, but both affect the whole learning process.

To assess the impact of optimizing different types of hyperparameters on the performance of GNNs, we employed the graph convolution model (GC) [2]. The hyperparameters from graph layers include the number of graph convolution layer n_g, the sizes of those graph convolution layers s_g, and the size of dense layer s_d which is defined in GC to generate molecular representations. These hyperparameters determine the range of the aggregation and affect the ability of learning representations. As for the hyperparameters in task-specific layers, in order to predict molecular properties $\mathbb{R}$, we employed a simple feedforward neural network consisting of a few fully-connected layers. So the task-specific hyperparameters include the number of fully-connected layers n_f (excluding the output layer), and the sizes of those layers s_f, and the activation function a. These hyperparameters will determine the ability to fit real problems to some degree. Furthermore, n_g and n_f will determine the number of s_g and s_f, so our search space is dynamic. All hyperparameters and their search ranges are summarised in Table 1. However, it is noted that CMA-ES does not support dynamic search space [1]. Therefore we turn to implement the pseudo-dynamic process, which keeps the CMA-ES to sample the maximum number of elements , thereafter n_g and n_f decide how many elements will be used to instantiate the model.

3 EXPERIMENTS

Lipophilicity [4] is a representative molecular benchmark dataset, and it corresponds to the task of predicting octanol/water distribution coefficient. The best hyperparameter values obtained from CMA-ES HPO are used to instantiate GCs, and these GCs are trained, validated, and tested for 30 times by the data sets with the ratio 80%/10%/10% of Lipophilicity dataset. The detailed results are shown in Table 2.

In Table 2, we observed that only optimizing fully-connected layers has relatively more serious over-fitting problem compared with the optimizing both types of hyperparameters, since it obtained less root mean square error (RMSE) value on the training set, and inversely had larger RMSE value on validation and test sets. In this case, it indicates optimizing fully-connected layers' hyperparameters would help to fit the problems. However, without optimizing graph layer hyperparameters, the molecular representations may not be better learnt, which leads to reduced performance of GNNs on test sets. Conducting HPO only on graph layers achieved lower performance than performing HPO on task-specific layers and the both. We believe the reason is that the default setting of GC only provides an output layer without hidden layers; it dramatically restricts the learning capability. Interestingly, after HPO, the hyperparameter a was assigned to *ReLU* in the experiment which is the same choice as described in [2], where the *ReLU* activation function was manually selected.

In summary, although graph layers and task-specific layers play different roles in GNNs, they need to be optimized together when solving practical problems. The reason is as follows: a better graph representation learned from graph layers needs to be supported by tailored task-specific layers to accomplish tasks. Similarly, task-specific layers also need appropriate graph representations to achieve good performance.

Lipophilicity		Graph layers	Fully-connected layers	Graph and fully-connected layers
Hyperparameters		$n_g = 6$, $s_g = [416, 256, 512, 320, 384, 128]$, $s_d = 768$	$n_f = 4$, $s_f = [1024, 896, 832, 64]$, $a = relu$	$n_g = 5$, $s_g = [480, 512, 256, 192, 224]$, $s_d = 960$, $n_f = 4$, $s_f = [704, 320, 128, 768]$, $a = relu$
Train	Mean RMSE	0.2148	**0.1369**	0.1701
	Mean Std	0.0206	0.0201	0.0361
Valid	Mean RMSE	0.6655	0.6656	**0.6239**
	Mean Std	0.0171	0.0144	0.0154
Test	Mean RMSE	0.7014	0.6786	**0.6472**
	Mean Std	0.0148	0.0165	0.0187

Table 2: HPO on the Lipophilicity dataset

4 CONCLUSIONS AND FUTURE WORK

In this paper, we elaborated the problem of HPO on GNNs for molecular property prediction, and investigated which types of hyperparameters should be selected to optimise when computational resources are limited. Based on our experiments, we conclude that both hyperparameters related to graph and task-specific layers should be optimised simultaneously, and leaving any one out will result in reduced performance.

Finally, we acknowledge that our experiments are based on one type of GNN model and one evolutionary strategy. However, we believe that our conclusion can be further generalised. We have selected the representative GNN model, used a state-of-the-art evolutionary HPO approach, and the benchmark dataset used for experiments is also representative in molecular property prediction problems [4]. Meanwhile, CMA-ES does not support the dynamic search space, which constrains its scalability. Other evolutionary HPO approaches can be applied in our future work to explore their effectiveness on optimising hyperparameters with dynamic search space for GNNs.

This research is supported by the Engineering and Physical Sciences Research Council (EPSRC) funded Project on New Industrial Systems: Manufacturing Immortality (EP/R020957/1). The authors are also grateful to the Manufacturing Immortality consortium.

REFERENCES

[1] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 2623–2631.

[2] David Duvenaud, Dougal Maclaurin, Jorge Aguilera-Iparraguirre, Rafael Gómez-Bombarelli, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P Adams. 2015. Convolutional networks on graphs for learning molecular fingerprints. *arXiv preprint arXiv:1509.09292* (2015).

[3] Nikolaus Hansen. 2016. The CMA evolution strategy: A tutorial. *arXiv preprint arXiv:1604.00772* (2016).

[4] Zhenqin Wu, Bharath Ramsundar, Evan N Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S Pappu, Karl Leswing, and Vijay Pande. 2018. MoleculeNet: a benchmark for molecular machine learning. *Chemical science* 9, 2 (2018), 513–530.

[5] Yingfang Yuan, Wenjun Wang, and Wei Pang. 2021. A Genetic Algorithm with Tree-structured Mutation for Hyperparameter Optimisation of Graph Neural Networks. *arXiv preprint arXiv:2102.11995* (2021).