Self-Supervised Learning for Molecular Property Prediction

Laurent Dillard, Shinya Yuki
Elix, Inc., Tokyo, Japan

Motivation

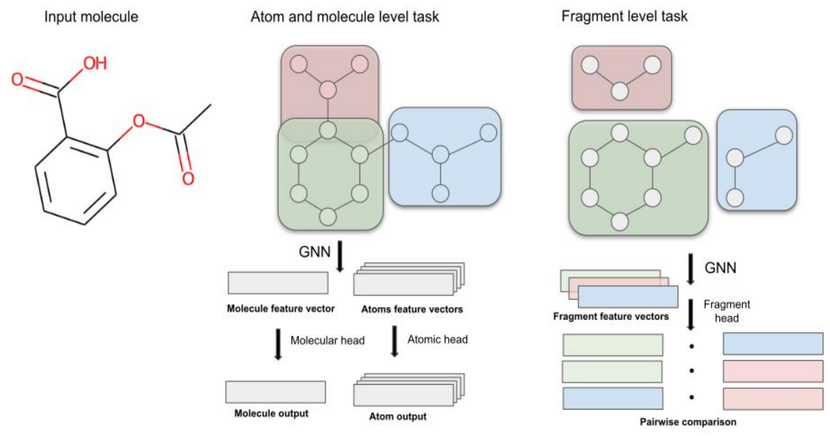
In drug discovery pipelines, computational methods have become of critical importance to allow virtual screening of large datasets of molecules and select promising candidates for further experimental validation.
Graph Neural Networks (GNNs) based predictive models have been successfully applied to the task of predicting molecular properties [1], however their potential is still limited by their reliance on large quantities of annotated data to reach desirable performance.
Several large databases of chemical compounds, like ZINC [2], counting millions to billions of samples have become available. Those provide massive amounts of unlabeled data and open up the way to unsupervised learning methods such as self-supervised learning.

We introduce a self-supervised framework tailored specifically for GNNs and molecular property prediction and evaluate its performance on a variety of benchmark datasets and GNN architectures. We also analyze the impact of the choice of input features on the benefits provided by self-supervision.

Self-supervised framework

To apply GNNs, molecules are first converted into graphs where nodes represent atoms while edges represent chemical bonds between atoms.

To learn feature representations relevant to molecular property prediction downstream tasks, we chose to design the pretext tasks of our self-supervised framework around 3 different scales of molecules which are all relevant to predicting molecular properties:



- **Molecule level**: Although many self-supervised methods for GNNs only focus on node level pre-training, molecular properties are often related to global molecular level characteristics therefore it is important to consider pretext task associated with graph level representations. The molecule level pretext task is a multi-label classification task where the model is taught to recognize which fragments, from a predefined list of 2000 molecular fragments, are present in the molecule.
- **Fragment level**: Some properties, such as toxicity, are especially related to the presence of certain functional groups and therefore best understood on the molecular fragment level. The fragment level pretext task is based on decomposing the molecules into fragments of random sizes and removing the edges between distinct fragments, then training the model to recognize which fragments originate from the same molecules. This is cast as a binary classification problems where the logits are obtained by dot product comparison of pairs of feature representations.
- **Atom level**: The core of GNNs is to extract node level representations from which graph level representations can then be obtained, it is therefore critical to also pretrain the GNN at the node level to ensure useful graph level representations. The atom level pretext task is defined as a classification problem to recognize which fragment each atom belongs to. Using the same 2000 fragments as in the molecule level task, each atom is labeled as the largest fragment it belongs to and the model is trained on this classification task.

The final loss to optimize is a weighted combination of the loss for each task:

$$\mathcal{L}_{final} = \lambda_{atom} * \mathcal{L}_{atom} + \lambda_{fragment} * \mathcal{L}_{fragment} + \lambda_{molecule} * \mathcal{L}_{molecule} \quad (1)$$

Results

Dataset	BACE (1522)	BBBP (2053)	ClinTox (1491)	SIDER (1427)	Toxcast (8615)	Tox21 (8014)	Average	Average gain
GCN	0.831(0.027)	0.898(0.033)	0.909(0.038)	0.605(0.027)	0.651(0.014)	0.769(0.013)	0.777(0.025)	
GCN (SSL)	0.858(0.023)	0.904(0.035)	0.927(0.026)	0.615(0.017)	0.653(0.012)	0.761(0.024)	0.786(0.023)	+0.009
GIN	0.844(0.028)	0.885(0.038)	0.902(0.045)	0.602(0.023)	0.625(0.013)	0.773(0.012)	0.772(0.027)	
GIN (SSL)	0.854(0.025)	0.891(0.032)	0.904(0.033)	0.614(0.017)	0.630(0.013)	0.773(0.018)	0.778(0.023)	+0.006
DMPNN	0.800(0.034)	0.894(0.038)	0.908(0.029)	0.620(0.021)	0.637(0.012)	0.762(0.018)	0.770(0.025)	
DMPNN (SSL)	0.855(0.027)	0.898(0.034)	0.910(0.040)	0.605(0.025)	0.618(0.013)	0.757(0.020)	0.774(0.026)	+0.004
GCN †	0.717(0.048)	0.864(0.038)	0.630(0.121)	0.572(0.023)	0.624(0.009)	0.715(0.048)	0.687(0.042)	
GCN (SSL) †	0.856(0.019)	0.896(0.037)	0.687(0.051)	0.592(0.019)	0.649(0.004)	0.755(0.017)	0.739(0.025)	+ 0.052
GIN †	0.816(0.038)	0.893(0.032)	0.560(0.070)	0.576(0.023)	0.598(0.019)	0.740(0.022)	0.697(0.033)	
GIN (SSL) †	0.849(0.032)	0.904(0.026)	0.605(0.148)	0.594(0.029)	0.623(0.011)	0.750(0.015)	0.721(0.043)	+0.014
DMPNN †	0.676(0.037)	0.833(0.051)	0.509(0.108)	0.564(0.024)	0.603(0.018)	0.710(0.039)	0.649(0.046)	
DMPNN (SSL) †	0.831(0.032)	0.877(0.041)	0.595(0.109)	0.594(0.014)	0.598(0.033)	0.733(0.016)	0.704(0.041)	+0.055
GROVER [18]	**0.894(0.028)**	**0.940(0.019)**	**0.944(0.021)**	**0.658(0.023)**	**0.737(0.010)**	**0.831(0.025)**	**0.834(0.021)**	
GROVER-GIN [18]	0.862(0.020)	0.925(0.036)		0.648(0.015)				
Hu et al.	0.851(0.027)	0.915(0.040)	0.762(0.058)	0.614(0.006)	0.714(0.019)	0.811(0.015)	0.778(0.028)	

Table 1: Evaluation of our self-supervised framework denoted by (SSL). For each architecture the baseline results correspond to training from randomly initialized weights. Two SOTA methods have been included for comparison. †denotes experiments where the reduced set of input features were used. Shaded results indicate best between baseline and SSL while **bold** results indicates best performance overall. For GROVER, GROVER-GIN and Hu et al., results were taken from [18].

For each experiment, the dataset was splitted into 80%/10%/10% train, validation and test sets using scaffold splitting. We measured the ROC-AUC on the test set and report both the mean value and standard deviation across 10 runs. We also report the results of two other state-of-the-art methods: GROVER [3] and Hu et al. [4].

Improvement largely varies depending on the dataset and model however when using a reduced set of input features the improvement obtained by using self-supervision significantly increases and becomes more consistent.

Conclusion

Our results indicate that self-supervision can successfully improve the performance of GNNs for molecular property prediction, especially in low data regime. However, our framework was not able to improve the performance consistently across datasets and architectures. Another important finding highlighted is the importance of the choice of input features for self-supervision. When using a very limited set of input features, the gain in performance obtained by applying self-supervision increased significantly and was consistent across all datasets and GNN architectures tested.

References

[1] Kevin Yang, et Al.: Analyzing learned molecular representations for property prediction. Journal of Chemical Information and Modeling, 59(8):3370–3388, Aug 2019. ISSN 1549-9596. doi: 10.1021/acs.jcim.9b00237

[2] Shoichet Brian K Irwin John J. Zinc–a free database of commercially available compounds for virtual screening. Journal of chemical information and modeling, 2005. doi: 10.1021/ci049714

[3] Yu Rong, Yatao Bian, Tingyang Xu, Weiyang Xie, Ying Wei, Wenbing Huang, and Junzhou Huang. Self-supervised graph transformer on large-scale molecular data. 2020

[4] Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay S. Pande, and Jure Leskovec. Pre-training graph neural networks. CoRR, abs/1905.12265, 2019