Molecular activity prediction using graph convolutional deep neural network considering distance on a molecular graph

Masahito Ohue[1,a)] Ryota Ii[1] Keisuke Yanagisawa[1] Yutaka Akiyama[1,b)]

Abstract: Machine learning is often used in virtual screening to find compounds that are pharmacologically active on a target protein. The weave module is a type of graph convolutional deep neural network that uses not only features focusing on atoms alone (atom features) but also features focusing on atom pairs (pair features); thus, it can consider information of nonadjacent atoms. However, the correlation between the distance on the graph and the 3-D coordinate distance is uncertain. In this paper, we propose three improvements for modifying the weave module. First, the distances between ring atoms on the graph were modified to bring the distances on the graph closer to the coordinate distance. Second, different weight matrices were used depending on the distance on the graph in the convolution layers of the pair features. Finally, a weighted sum, by distance, was used when converting pair features to atom features. The experimental results show that the performance of the proposed method is slightly better than that of the weave module, and the improvement in the distance representation might be useful for compound activity prediction.

Keywords: graph convolutional neural network, ligand-based virtual screening, machine learning, deep learning

1. Introduction

In drug research and development, it takes at least ten years to produce a drug, and development costs are estimated to be several billion US dollars or more [1]. High-throughput screening methods for screening compounds that show activity against proteins targeted by drug discovery from large-scale compound libraries are popular [2]; however, screening vast numbers of compounds is expensive. In contrast, virtual screening is expected to be able to predict active compounds efficiently using a computer [3].

One of the frameworks of virtual screening is a ligand-based method that uses machine learning to predict activity using known activity information as a teacher label [4, 5]. In particular, in recent years, each atom of a compound is regarded as a node, and a bond is considered as an edge graph. Based on this, feature extraction can be performed using neural networks [6–8]. The graph convolutional neural network (GCN), which realizes the convolutional deep neural network by using a convolution operation on the graph structure, is used for such applications.

For graph feature extraction using GCN, neural graph fingerprints (NGF) [6], the GCN by Han *et al.* [7] and the weave module [8] are often used. These methods do not generate compound descriptors (feature vectors) based on a specific rule like ordinary fingerprints and have the advantage of being able to represent feature vectors by learning molecular structures flexibly. NGF and Han's GCN do not consider edge features in the molecular graph but focus on learning the relationship with the nearest neighbor node. On the other hand, the weave module of Kearnes *et al.* transforms feature vectors using pair features with distant atoms in addition to atom features focused only on atoms. Thus, the Weave module can consider features between distant atoms. However, the number of atoms forming a pair is different for each distance. Furthermore, the pair features of the Weave module cannot be considered in that respect.

In this paper, we propose a new improved GCN that can consider features between distant atoms by modifying the Weave module. In order to make effective use of the distance features on the molecular graph in the Weave module, we considered three improvements: correction of the distance on the molecular graph with respect to atoms in the ring structure, convolution method of pair features, and assembling of the pair features.

2. Weave Module [8]

The network architecture of the Weave module [8] is shown in **Fig. 1**. The Weave module consists of the seven transformations shown in ①–⑦ in Fig. 1. This study was targeted at improving the method of generating the initial feature and transformation operation ③ (transforming from

[1] Department of Computer Science, School of Computing, Tokyo Institute of Technology
[a)] ohue@c.titech.ac.jp
[b)] akiyama@c.titech.ac.jp

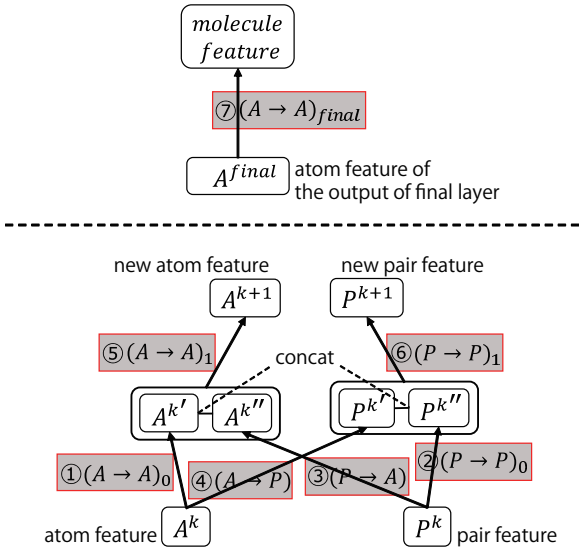

Fig. 1 Weave module [8]

the pair feature to the intermediate atom feature). These operations are described as follows, and their further details can be found in [8].

2.1 Initial features

Initial atom feature $\boldsymbol{A}^0 \in \mathbb{R}^{n_{max} \times d_a^0}$ and pair feature $\boldsymbol{P}^0 \in \mathbb{R}^{n_{max}^2 \times d_p^0}$, which are inputs to the network, use simple descriptors such as atom and bond types. n_{max} is the maximum number of atoms in the molecule. $\boldsymbol{A}^0$ is a matrix in which n_{max} number of d_a^0-dimensional feature vectors (row vectors) corresponding to one atom are vertically arranged. $\boldsymbol{P}^0$ is a matrix in which n_{max}^2 number of d_p^0-dimensional feature vectors (row vectors) corresponding to one atom pair are vertically arranged.

2.2 Transformation operation ③: transform intermediate atom feature from pair feature

In Weave module layer k, the following operation, as shown in **Fig. 2** is performed on all the atom pairs comprising atom i. The intermediate atom feature for atom i is calculated by adding them.

$$\boldsymbol{a}_i^{k''} = \sum_j f\left(\boldsymbol{W}_{PA}^k \boldsymbol{p}_{(i,j)}^k + \boldsymbol{b}_{PA}^k\right) \quad (1)$$

where $\boldsymbol{p}_{(i,j)}^k \in \mathbb{R}^{d_p^k}$ is an input pair feature vector of atom pair (i,j) in the k-th layer, $\boldsymbol{a}_i^{k''} \in \mathbb{R}^{d_{PA}}$ is an output atom feature vector of atom i, $\boldsymbol{W}_{PA}^k \in \mathbb{R}^{d_{PA} \times d_p^k}$ is a weight matrix, and $\boldsymbol{b}_{PA}^k \in \mathbb{R}^{d_{PA}}$ is a bias vector. $f(\cdot)$ is an activation function that applies ReLU to all elements of a vector. Atom feature $\boldsymbol{A}^{k''} \in \mathbb{R}^{n_{max} \times d_{PA}}$ is vertically arranged as $\boldsymbol{a}_i^{k''}$ for all atoms $i = 1, ..., n_{max}$.

2.3 Point of issue

The following issues are present in the Weave module.

(1) **Distance on the graph for atoms in a ring structure**
An uncertainty exists as to whether the distance on the graph and the real three-dimensional distance are correlated between atom pairs in the ring structure.

(2) **Convolution of pair features**
Uniform weights are used for all pair features regardless of the distance length on the graph.

(3) **Assembling of pair features**
All atoms in the pair are uniformly added to the convoluted pair feature, and the difference due to the distance between the pairs is not reflected.

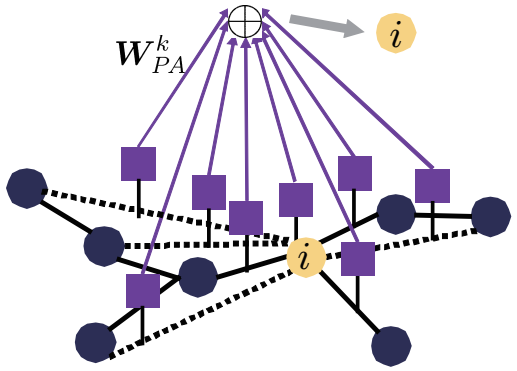

Fig. 2 Converting pair features to atom features

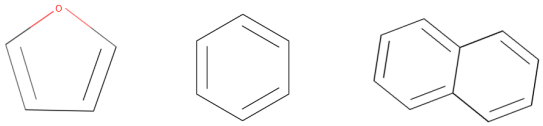

Fig. 3 Examples of ring structure

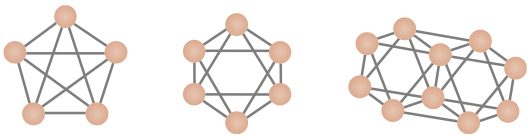

Fig. 4 Examples of redefined ring structure (Prop. A)

3. Proposed Method

Here, we introduce three improvements to solve the above-mentioned issues of the Weave module.

3.1 Correction of distances related to atoms in ring structures (Prop. A)

The pair feature of the Weave module defines the distance between atom pairs as the length of the shortest path on the graph. The ring structure is relatively rigid in terms of the actual molecular conformation compared to the chain structure. Moreover, the distance on the conformation is shorter than the distance on the graph, considering two atoms in the molecule. Therefore, with respect to the atom of interest, the atom pair at the orthoposition and metaposition is distance 1, and the atom pair at the para position is distance 2 (**Fig. 3** and **Fig. 4**).

3.2 Convolution of pair features with different weights (Prop. B)

We improved the weights for pair features to be determined by learning the use of neural networks. In the Weave module, pair features were convoluted using the same weight

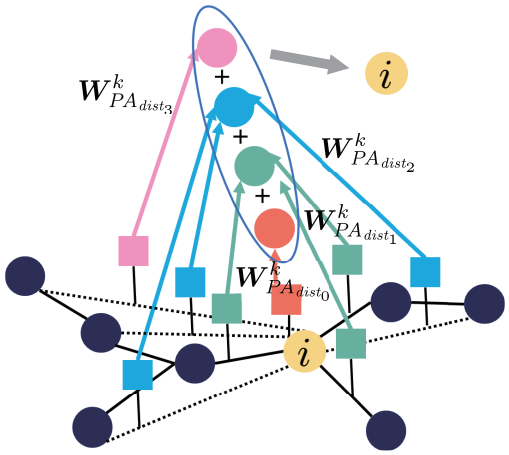

Fig. 5 Convolution using different weights (Prop. B)

matrix, regardless of the distance length. Therefore, we labeled distances $dist_0, dist_1, ..., dist_n, ..., dist_{max}, dist_\infty$ from the focus atom to distinguish each pair feature. Here, $dist_\infty$ represents all distances greater than the maximum atomic pair distance $dist_{max}$. We used different weight matrices $\boldsymbol{W}_{PA_{dist_0}}, \boldsymbol{W}_{PA_{dist_1}}, ..., \boldsymbol{W}_{PA_{dist_n}}, ..., \boldsymbol{W}_{PA_{dist_{max}}}, \boldsymbol{W}_{PA_{dist_\infty}}$ corresponding to these distances in the convolution of pair features.

In Prop. B, weight matrices according to the distance were used for atom pairs and convolution was performed. The intermediate atom feature of atom i was calculated by taking the sum of atom pairs of atom i. This operation is shown in **Fig. 5**.

3.3 Assembling pair features based on distance (Prop. C)

If the interatomic distance on the graph is large, the interatomic distance on conformation does not become constant. Atom pairs with large interatomic distances appear to be less important than those with small interatomic distances. Therefore, when finding the intermediate atom feature $\boldsymbol{a}_i^{k''}$ of atom i, the closer the distance d_{ij} is, the larger is the weighting performed by the three kinds of coefficients $g(d_{ij})$:

$$g(d) = 0 \ if \ d > dist_{max} \ else \ 1 \quad \text{(step)} \tag{2}$$

$$g(d) = -0.1d + 1 \quad \text{(linear)} \tag{3}$$

$$g(d) = 1/d^2 \quad \text{(quadratic)} \tag{4}$$

This modifies Eq. (1) as follows:

$$\boldsymbol{a}_i^{k''} = \sum_j g(d_{ij}) f\left(\boldsymbol{W}_{PA}^k \boldsymbol{p}_{(i,j)}^k + \boldsymbol{b}_{PA}^k\right) \tag{5}$$

4. Experiments

4.1 Dataset

We used the Biophysics datasets HIV, MUV, and PCBA from MoleculeNet [9]. Molecular data are provided in SMILES format and converted to 2-D molecular graphs using RDKit [10]. Hydrogen atoms were omitted, and compounds with the huge number of heavy atoms exceeding maximum number of atoms, n_{max}, were excluded from the dataset. The number of tasks in each dataset, number of active compounds, number of inactive compounds, number of compounds, and number of excluded compounds are shown in **Table 1**. Given that the same compound is registered with different labels between each task, the numbers of active and inactive compounds were counted in duplicate.

Table 1 Details of datasets

dataset	#tasks	#pos[*1]	#neg[*1]	#cmpds	#excluded
HIV	1	1,319	39,065	40,384	743
MUV	17	489	249,397	93,087	0
PCBA	128	471,273	33,509,569	437,035	894

[*1] Given that the same compound is registered with different labels between each task, the number counted in duplicate as described.

Table 2 Model hyperparameters

hyperparameter		value
maximum number of atoms in molecule n_{max}		60
maximum atomic pair distance $dist_{max}$		1–5
Weave module$k$		2
$d_{AA}, d_{PP}, d_{PA}, d_{AP}, d_A, d_P$		50
$d_{A_{final}}$		128
#fully connected layers		2000, 100
training	batch size	96
	optimizer	Adam
	learning rate	0.001
	epoch	100
train:valid:test	HIV	8 : 1 : 1
	PCBA, MUV	6 : 2 : 2
trial m	HIV	10
	PCBA, MUV	5

4.2 Training and evaluation

The GCN model was implemented using the deep learning library, Chainer Chemistry (version 0.4.0) [11]. The hyperparameters of GCN are listed in **Table 2**. These were the same as those used by Kearnes *et al.* [8]. We attempted to set maximum atom pair distance, $dist_{max}$, to 1–5.

In this study, the prediction performance of the model was evaluated using the ROC curve [12] and area under the curve (AUC), as shown in Eq. (6).

$$\mathrm{AUC} = 1 - \frac{1}{N_{\mathrm{Pos}}} \sum_{i=1}^{N_{\mathrm{Pos}}} \frac{N_{\mathrm{Neg}}^i}{N_{\mathrm{Neg}}}, \tag{6}$$

where N_{Pos} is the number of active compounds, N_{Neg} is the number of inactive compounds, N_{Neg}^i is the number of inactive compounds ranked higher than the i-th active compound, N is the number of compounds.

Each dataset was divided into the training data (train), validation data (valid), and test data (test) according to the ratio shown in Table 2. For each task in the dataset, we selected an epoch (learning checkpoint) that gives the best AUC for the validation data and applied it to the test data to calculate the averaged AUC value for each task. The AUC used in evaluation (AUC_eval) is calculated as follows.

$$n_{best,\mathcal{T}} = \underset{n}{\mathrm{argmax}} \underset{i}{\mathrm{mean}} \left(\mathrm{AUC}_{\mathcal{T},n,i}^{\mathrm{valid}}\right) \tag{7}$$

$$\mathrm{AUC\_eval} = \underset{\mathcal{T}}{\mathrm{median}} \underset{i}{\mathrm{mean}} \left(\mathrm{AUC}_{\mathcal{T},n_{best,\mathcal{T}},i}^{\mathrm{test}}\right), \tag{8}$$

where $\mathcal{T}$ represents each task, n is the epoch, and $i(= 1, ..., m)$ is the trial. $\mathrm{AUC}_{\mathcal{T},n,i}^{\mathrm{valid}}$ is an AUC value of task $\mathcal{T}$ of validation data using trained GCN with epoch n in trial i. $\mathrm{AUC}_{\mathcal{T},n_{best,\mathcal{T}},i}^{\mathrm{test}}$ is the AUC value of task $\mathcal{T}$ of test

Table 3 AUC of each dataset using Props. A and B. The value in bold-font is the best value for each model.

dataset	model	distance				
		1	2	3	4	5
HIV	Weave	0.796	0.798	0.795	0.793	**0.801**
	Prop. A	0.796	**0.803**	0.799	0.794	0.798
	Prop. B	0.794	0.797	0.797	0.799	**0.806**
	Prop. A&B	**0.806**	0.798	0.801	0.800	0.800
MUV	Weave	0.680	0.720	0.739	0.689	**0.743**
	Prop. A	0.706	**0.783**	0.735	0.741	0.754
	Prop. B	0.723	**0.738**	0.714	0.671	0.736
	Prop. A&B	0.757	**0.760**	0.704	0.737	0.693
PCBA	Weave	0.822	**0.824**	0.821	0.821	0.823
	Prop. A	0.821	**0.825**	0.823	0.823	0.824
	Prop. B	0.822	0.821	0.820	0.822	**0.823**
	Prop. A&B	0.819	0.821	**0.823**	0.822	0.821

Table 4 AUC of each dataset using Prop. C. The value in bold-font represents the best value for each model.

dataset	model	distance				
		1	2	3	4	5
HIV	Weave	0.796	0.798	0.795	0.793	**0.801**
	step	0.766	0.767	0.765	0.769	**0.772**
	linear	0.799	0.798	0.803	0.799	**0.807**
	quadratic	0.796	0.791	**0.803**	0.798	0.803
MUV	Weave	0.680	0.720	0.739	0.689	**0.743**
	step	0.629	**0.721**	0.692	0.677	0.690
	linear	0.731	**0.749**	0.687	0.713	0.729
	quadratic	**0.752**	0.742	0.713	0.722	0.702

data using a trained GCN with epoch $n_{best,\mathcal{T}}$ in trial i. The division of the dataset at each trial i is randomly performed each time.

5. Results and Discussion

5.1 Performance of Props. A and B

The results of comparing the AUC of each dataset is shown in **Table 3** for the models of the Weave module, Prop. A, Prop. B, and Prop. A&B. Prop. A provided higher prediction performance than the Weave module in the MUV dataset but remained as accurate as the Weave module in HIV and PCBA. The accuracy of the Prop. B alone is almost the same as that of the Weave module, while the combination of the Prop. A and B yields a slightly higher accuracy.

5.2 Results of Prop. C

The prediction results for the HIV and MUV datasets are listed in **Table 4** with respect to Prop. C, three functions, and the Weave module for assembling pair features. The models of the linear and quadratic functions of Prop. C have a higher AUC value. The improvement by the model of the step function was not significant.

5.3 Conclusion

Three types of improvements were made to the operation of converting a pair feature to an atom feature in the Weave module; (A) By changing distance d of the atom pair contained in the ring structure to $\lceil d/2 \rceil$, the distance on the graph was corrected to correlate with the distance on conformation. As a result of the evaluation experiment, the prediction accuracy is improved compared to the Weave module, and features between distant atoms were also successfully used. (B) We attempted to generalize the model by using different weights for each distance in the convolution process combined with the proposed correction of the distance on the graph in the ring structure in the compound. The prediction accuracy was higher when performing convolution with different weights for each distance compared to the Weave module. According to the analysis of the weight matrix dynamics, the proposed method was found to be useful, especially in the 0th layer of the Weave module. (C) We proposed a method of incorporating pair features that emphasize the atoms in the vicinity of the atom of interest by using coefficients according to the distance. We achieved some improvement in the prediction accuracy by assembling paired features by using linear and quadratic weights.

It is worthwhile to verify that this improvement is also effective for other tasks of compound supervised learning, e.g., drug-like compound filter [13], side-effect prediction [14], toxicity prediction [15], and stability prediction [16, 17].

Acknowledgments This work was partially supported by JSPS KAKENHI (Grant No. 17H01814, 17J06897, 18K18149), JST CREST (Grant No. JPMJCR1303), MEXT Program for Building Regional Innovation Ecosystems, JST Research Complex Program, and AMED BINDS (Grant No. JP18am0101112).

References

[1] A. Mullard, "New drug costs US $2.6 billion to develop," *Nat. Rev. Drug. Discov.*, vol. 13(12), pp. 877, 2014.
[2] R. Macarron, M. N. Banks, D. Bojanic, *et al.* "Impact of high-throughput screening in biomedical research," *Nat. Rev. Drug. Discov.*, vol. 10, pp. 188–195, 2011.
[3] S. P. Leelananda, S. Lindert, "Computational methods in drug discovery," *Beilstein J. Org. Chem.*, vol. 12, pp. 2694–2718, 2016.
[4] L. J. Colwell, "Statistical and machine learning approaches to predicting protein–ligand interactions," *Curr. Opin. Struct. Biol.*, vol. 49, pp. 123–128, 2018.
[5] B. J. Neves, R. C. Braga, C. C. Melo-Filho, *et al.* "QSAR-based virtual screening: advances and applications in drug discovery," *Front. Pharmacol.*, vol. 9, no. 1275, 2018.
[6] D. Duvenaud, D. Maclaurin, J. Aguilera-Iparraguirre, *et al.* "Convolutional networks on graphs for learning molecular fingerprints," In *Proc. NIPS*, pp. 2215–2223, 2015.
[7] H. Altae-Tran, B. Ramsundar, A. S. Pappu, *et al.* "Low data drug discovery with one-shot learning," *ACS Cent. Sci.*, vol. 3, pp. 283–293, 2017.
[8] S. Kearnes, K. McCloskey, M. Berndl, *et al.* "Molecular graph convolutions: moving beyond fingerprints," *J. Comput.-Aided Mol. Des.*, vol. 30(8), pp. 595–608, 2016.
[9] Z. Wu, B. Ramsundar, E. N. Feinberg, *et al.* "MoleculeNet: a benchmark for molecular machine learning," *Chem. Sci.*, vol. 9, pp. 513–530, 2018.
[10] G. Landrum, RDKit: Open-source cheminformatics software. http://www.rdkit.org.
[11] Chainer Chemistry: a library for deep learning in biology and chemistry. http://github.com/pfnet-research/chainer-chemistry (accessed March 28, 2019)
[12] A. N. Jain, A. Nicholls, "Recommendations for evaluation of computational methods," *J. Comput.-Aided Mol. Des.*, vol. 22(3–4), pp. 133–139, 2008
[13] M. Mochizuki, S. D. Suzuki, K. Yanagisawa, *et al.* "QEX: target-specific druglikeness filter enhances ligand-based virtual screening," *Mol. Divers*, vol. 23(1), pp. 11–18, 2019.
[14] M. Zitnik, M. Agrawal, J. Leskovec, "Modeling polypharmacy side effects with graph convolutional networks," *Bioinformatics*, vol. 34(13), pp. i457–i466, 2018.
[15] M. Fernandez, F. Ban, G. Woo, *et al.* "Toxic Colors: the use of deep learning for predicting toxicity of compounds merely from their graphic images," *J. Chem. Inf. Model.*, vol. 58(8), pp. 1533–1543, 2018.
[16] T. Tajimi, N. Wakui, K. Yanagisawa, *et al.* "Computational prediction of plasma protein binding of cyclic peptides from small molecule experimental data using sparse modeling techniques," *BMC Bioinform.*, vol. 19(19), no. 527, 2018.
[17] R. Watanabe, T. Esaki, H. Kawashima, *et al.* "Predicting fraction unbound in human plasma from chemical structure: improved accuracy in the low value ranges," *Mol. Pharm.*, vol. 15(11), pp. 5302–5311, 2018.