[Graph 스터디] Neural Message Passing for Quantum Chemistry(MPNN)

작성자 : 채윤병

1. Introduction

논문의 목적 : 분자 그래프로부터 화학적 성질 예측을 위한 분자의 특성을 학습할 수 있고 그래프 동형(graph isomorphism)문제에 대해 invariant한 효과적인 머신러닝 모델을 제안하는 것

그래프 동형(graph isomorphism) - 노드의 순서의 차이로 인해 다르게 표현되지만 완전히 같은 구조를 가지고 있는 그래프

→ MPNN(Message Passing Neural Networks)

QM9 dataset - 화학 분자 benchmark dataset, 13만개의 분자의 DFT 연산으로 근사한 13가지 특성이 존재

DFT(Density functional theory)? - 기본 재료 특성과 같은 고차 매개변수를 요구하지 않고 양자 역학적 고려 사항을 기반으로 재료 거동을 예측하고 계산

QM9은 화학적 특성의 계산에 사용되는 원자들의 low energy conformation(낮은 에너지 상태의 형태)을 위한 공간적 정보가 들어있다. QM9은 원자 사이의 거리, 결합각과 같은 기하학적 정보가 있는 경우(a)와 원자(atom)와 결합(bond)에 대한 정보만 있고 원자들의 공간적인 위치에 따라 '정의되어 있는' 화학적 특성을 계산해야 하는 경우(b)를 고려한다.

(b)의 경우 모델이 내재적으로 low energy 3D conformation을 결정할 수 있게 학습이 이루어져야 한다!

Error

- DFT error - DFT approximation의 error
- Chemical accuracy - Chemistry community가 설립한 target error

Key Contribution

1. MPNN이 13개의 target value에 대해서 SOTA를 달성했고, 13개 중에서 11개가 chemical accuracy범위 내에 있었다.
2. 13개 중에서 5개를 chemical accuracy범위 내로 예측하는 다른 MPNN모델을 제안했다.
3. 연산 속도와 메모리의 증가 없이 MPNN을 학습시키는 general한 방법론을 제안했다.

2. Message Passing Neural Networks

Forward pass - 1. Message passing phase 2. Readout phase

Message passing phase - ① Message function, Mt ② Vertex update function, Ut

$$m_v^{t+1} = \sum_{w \in N(v)} M_t(h_v^t, h_w^t, e_{vw}) \quad (1)$$

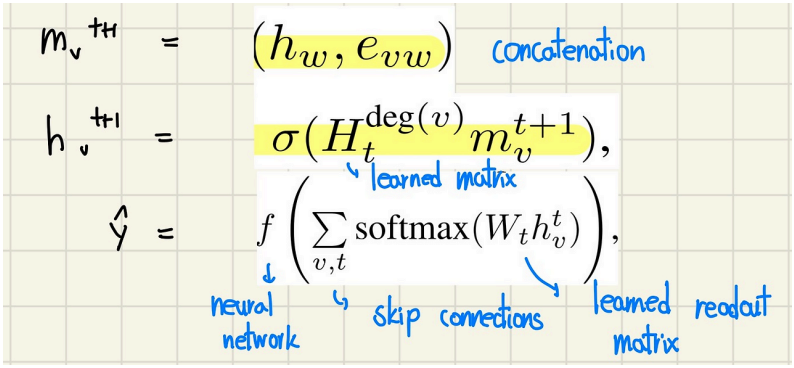
$$h_v^{t+1} = U_t(h_v^t, m_v^{t+1}) \quad (2)$$

$$\hat{y} = R(\{h_v^T \mid v \in G\}). \quad (3)$$

Mt, Ut, R은 모두 미분가능한 함수

R은 node state의 집합에 대해 적용되며 MPNN이 그래프 동형(graph isomorphism)문제에 대해 invariant하기 위해서 node state의 순서에 따라 invariant해야한다.

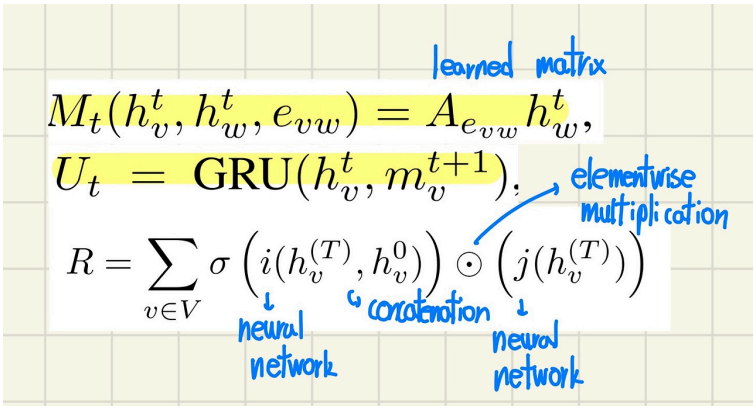
Convolutional networks for learning molecular fingerprints



Message passing이 연결된 node와 연결된 edge를 각각 더하는 방법으로 정의되기 때문에 edge state와 node state간의 correlation을 나타낼 수 없다.

Gated Graph Neural Networks (GG-NN)

각 타임 스텝에서 큰 그래프의 일부분만을 이용해 message를 passing 함으로써 연산량을 줄였다.



Interaction Networks

그래프의 각 node의 target이 있는 경우, 그래프 level 단위의 target이 있는 경우, 각 timp step에서 node level이 영향을 미치는 경우 고려

> some outside influence on the vertex v. The message function $M(h_v, h_w, e_{vw})$ is a neural network which takes the concatenation (h_v, h_w, e_{vw}). The vertex update function $U(h_v, x_v, m_v)$ is a neural network which takes as input the concatenation (h_v, x_v, m_v). Finally, in the case where there is a graph level output, $R = f(\sum_{v \in G} h_v^T)$ where f is a neural network which takes the sum of the final hidden states h_v^T. Note the original work only defined the model for $T = 1$.

Molecular Graph Convolutions

Message passing 단계에서 edge representation e_vw를 업데이트

$M(h_v^t, h_w^t, e_{vw}^t) = e_{vw}^t$. The vertex update function is $U_t(h_v^t, m_v^{t+1}) = \alpha(W_1(\alpha(W_0 h_v^t), m_v^{t+1}))$ where $(.,.)$ denotes concatenation, α is the ReLU activation and W_1, W_0 are learned weight matrices. The edge state update is defined by $e_{vw}^{t+1} = U_t'(e_{vw}^t, h_v^t, h_w^t) = \alpha(W_4(\alpha(W_2, e_{vw}^t), \alpha(W_3(h_v^t, h_w^t))))$ where the W_i are also learned weight matrices.

Deep Tensor Neural Networks

$$M_t = \tanh\left(W^{fc}((W^{cf} h_w^t + b_1) \odot (W^{df} e_{vw} + b_2))\right)$$

where W^{fc}, W^{cf}, W^{df} are matrices and b_1, b_2 are bias vectors. The update function used is $U_t(h_v^t, m_v^{t+1}) = h_v^t + m_v^{t+1}$. The readout function passes each node independently through a single hidden layer neural network and sums the outputs, in particular

$$R = \sum_v \text{NN}(h_v^T).$$

Laplacian Based Methods

Graph laplacian L을 도입해서 graph에 convolution의 개념을 도입(GCN)

The Kipf & Welling (2016) model results in a message function $M_t(h_v^t, h_w^t) = c_{vw} h_w^t$ where $c_{vw} = (\deg(v)\deg(w))^{-1/2} A_{vw}$. The vertex update function is $U_v^t(h_v^t, m_v^{t+1}) = \text{ReLU}(W^t m_v^{t+1})$. For the exact expres-

→ Moving forward.. 여러 MPNN의 예시에서 실용적인 중요도를 가지는 구체적인 활용에 대해 집중해야 한다. 그래야 모델링의 개선이나 활용의 detail을 정할 수 있다.

3. Related Work

그동안 과학자들은 양자 역학의 근사를 위해서 DFT(Density functional theory), GW approximation, Quantum Monte-Carlo 등 다양한 방법을 사용했다.

그 중 DFT는 연산량이 크기 때문에 Neural Network로 이를 근사하려는 노력이 있었다. but 성공적이지 못함.

따라서 최근에 양자 역학의 근사를 Neural Network로 직접하려는 연구가 있었다.

1. 분자 거동 예측(molecular dynamics simulation)을 위한 분자 에너지와 구성의 근사를 NN으로 시도
2. Kernel Ridge Regression(KRR)을 이용하여 넓은 범위의 분자의 atomization(**Atomization** refers to breaking bonds in some substance to obtain its constituent atoms in gas phase.) 에너지를 측정

이 두 가지 모두 대칭 함수와 같은 hand-engineered feature를 사용했는데 이는 generalize 문제(원자의 종류가 많아질 경우)와 graph isomorphism에 invariant 하지 않을 수 있다는 문제점이 있다.

4. QM9 dataset

H, C, O, N, F 등 9개의 원자로 이루어져 있으며 134k의 약물과 유사한 분자에 대한 정보를 담고 있는 dataset.

각 분자에 대해 합리적인 low energy structure를 찾고 atom position이 가능한지(availability) 판별하기 위해 DFT가 사용된다. 또한 여러 기본적인 화학적 특성이 계산되어 담겨있음.

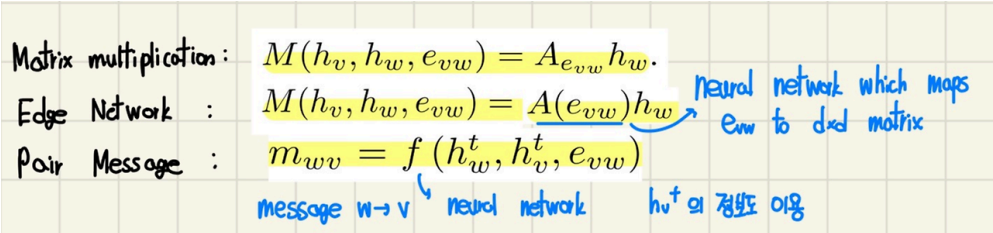
Bond 관련 특성 4가지 + 분자의 진동에 대한 특성 2가지 + 전자 상태에 대한 특성 3가지 + 전자의 공간적 분포에 대한 특성 3가지

5. MPNN Variants

GG-NN을 baseline으로 잡고 다른 message function, output function 등을 시도해 보았다.

MPNN의 input은 그래프 노드의 feature vector x_v, 인접 행렬 A(두 원자의 공간적인 거리와 결합 종류의 정보까지 담고 있는 weighted matrix)

5.1 Message Functions



5.2 Virtual Graph Elements

연결되어 있지 않은 node 쌍 간의 "virtual edge type"을 추가하고 모든 node와 특별한 edge type(virtual)으로 연결되어 있는 "master node"를 추가. Master node는 그래프의 global한 정보를 담고 있다.

5.3 Readout Functions

첫 번째로 원래 GG-NN에서 사용한 readout function 사용, 다른 방법으로는 set2set model(Input으로 projection한 tuple을 사용하여 단순히 더하는 방법보다 표현력이 좋은 방법) 사용

5.4 Multiple Towers

d차원의 node embedding을 k개의 d/k차원의 node embedding으로 만들어 k개의 copies에 대해 각각 propagation step을 진행 → 연산의 효율 증가

6. Input representation

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)

Number of Hydrogens와 같은 feature로 atom feature에 포함시키는 것과 달리 수소 분자를 외부 노드로 만들어서도 실험을 진행했다(이 경우 그래프의 노드개수는 최대 29개 → QM9 dataset의 분자는 small molecule).

3가지의 edge representation

1. Chemical Graph - Discrete bond types(single, double, triple, aromatic)
2. Distance bins - 거리에 대한 히스토그램을 그려 10개의 hand chosen bin 설정(결합의 경우 bond type에 대한 정보 + 결합되지 않았을 때의 거리정보)
3. Raw distance feature - (Atom pair의 거리정보 + Bond type에 대한 원핫인코딩)

7. Training

실험의 세부 설정들 - Uniform random hyperparameter(50 trials), Adam optimizer, learning rate 1e-5 ~ 5e-4, Target value normalize(mean 0 ,variance 1), Random 하게 13만개 molecule중에서 1만개 validation, 1만개 test 선택, MSE minimize하고 MAE로 evaluate

8. Results

QM9 dataset에 대해 가장 적절한 input representation 뿐만 아니라 best MPNN을 찾기 위해서 여러가지 실험을 진행했다. 결합 타입과 공간적인 정보를 담고 있는 edge feature를 사용하고 수소 분자를 하나의 노드(explicit node)로 설정했을 때가 가장 성능이 좋았던 실험 설정이었다 + (message function - edge network, output function - set2set).

13개의 target value각각에 대해 따로 학습을 시키는 것이 최대 40%의 성능 향상이 있었다.(상당히 번거롭지 않을까..?, Weakness)

Table 2. Comparison of Previous Approaches (left) with MPNN baselines (middle) and our methods (right)

Target	BAML	BOB	CM	ECFP4	HDAD	GC	GG-NN	DTNN	enn-s2s	enn-s2s-ens5
mu	4.34	4.23	4.49	4.82	3.34	0.70	1.22	-	**0.30**	0.20
alpha	3.01	2.98	4.33	34.54	1.75	2.27	1.55	-	**0.92**	0.68
HOMO	2.20	2.20	3.09	2.89	1.54	1.18	1.17	-	**0.99**	0.74
LUMO	2.76	2.74	4.26	3.10	1.96	1.10	1.08	-	**0.87**	0.65
gap	3.28	3.41	5.32	3.86	2.49	1.78	1.70	-	**1.60**	1.23
R2	3.25	0.80	2.83	90.68	1.35	4.73	3.99	-	**0.15**	0.14
ZPVE	3.31	3.40	4.80	241.58	1.91	9.75	2.52	-	**1.27**	1.10
U0	1.21	1.43	2.98	85.01	0.58	3.02	0.83	-	**0.45**	0.33
U	1.22	1.44	2.99	85.59	0.59	3.16	0.86	-	**0.45**	0.34
H	1.22	1.44	2.99	86.21	0.59	3.19	0.81	-	**0.39**	0.30
G	1.20	1.42	2.97	78.36	0.59	2.95	0.78	.84[2]	**0.44**	0.34
Cv	1.64	1.83	2.36	30.29	0.88	1.45	1.19	-	**0.80**	0.62
Omega	0.27	0.35	1.32	1.47	0.34	0.32	0.53	-	**0.19**	0.15
Average	2.17	2.08	3.37	53.97	1.35	2.59	1.36	-	**0.68**	0.52

왼쪽 5개의 baseline - hand engineered feature 사용, GC와 GG_NN - hand engineered feature를 사용한 baseline

13개의 target value 모두 기존 모델들보다 좋은 결과를 보였으며 13개 중 11개의 target value가 chemical accuracy를 달성했다.

Training without spatial information

Spaital information을 넣지 않고 실험을 진행했을 때는 MPNN이 long range interaction을 고려하도록 하는 것이 성능향상에 큰 도움이 되었다.

Towers

Table 4. Towers vs Vanilla GG-NN (no explicit hydrogen)

Model	Average Error Ratio
GG-NN + joint training	1.92
towers8 + joint training	**1.75**
GG-NN + individual training	1.53
towers8 + individual training	**1.37**

d차원의 node embedding을 k개의 d/k차원의 node embedding으로 만들어 k개의 copies에 대해 각각 propagation step을 진행하는 multiple tower 방식이 정규화의 역할을 했다.

9. Conclusion

적절한 message, update, output function을 가진 MPNN 모델이 분자의 특성을 예측하기 위한 유용한 inductive bias를 제공한다. Master node나 set2set output과 같이 node간의 long range interaction을 고려하는 것이 중요하다.

미래에 MPNN이 해결해야할 문제 → Generalization - 더 큰 그래프(molecule)에 대해서도 generalization을 개선하는 것

공간적 정보를 사용할 때 큰 분자에서 generalization이 어려운 이유

1. Pairwise distance의 분포가 atom 개수에 많이 영향을 받는다.
2. Incoming message의 개수가 node 개수에 영향을 받는 fully connected graph를 사용해야한다. → Attention mechanism을 적용할 수 있다.

☆ Attention을 적용한 MPNN 읽어보기