

Neural Message Passing for Quantum Chemistry, Gilmer+, '17

2017年6月9日

@shima_x

Agenda

- 概要
- モチベーション
- 手法
- 実験/結果
- コメント

概要

- Message Passing Neural Networks (MPNNs)の提案
 - グラフ構造の入力に対してmessage passing algorithmとaggregation functionを学習するモデル
 - 計算コストの高いDFT（密度汎関数理論）をNNでモデル化したもの(計算高速化)
 - 10^3 second \rightarrow 10^{-2} secondに計算時間を短縮
- 分子のグラフ構造データから、形状の同形性を考慮した普遍性を学習する
- 新規の構造との関連性/分子特性の把握
- QM9 datasetを使用
 - 約130k個の分子がそれぞれ13個の特性を持っており、それを予測するタスクで評価
- 分子特性を推定するタスクにおいてSoTA

貢献

- 精度の高いモデルを構築
 - chemical accuracyで11/13を的中
 - 分子の構造のみを入力とした場合でも5/13の特性を的中
- 大きなグラフ構造が入力されたとしても、高速にメモリ使用量を抑えた計算が可能なモデルを構築した

モチベーション

- 有機物の特性の推定を汎用的に低コストで行いたい

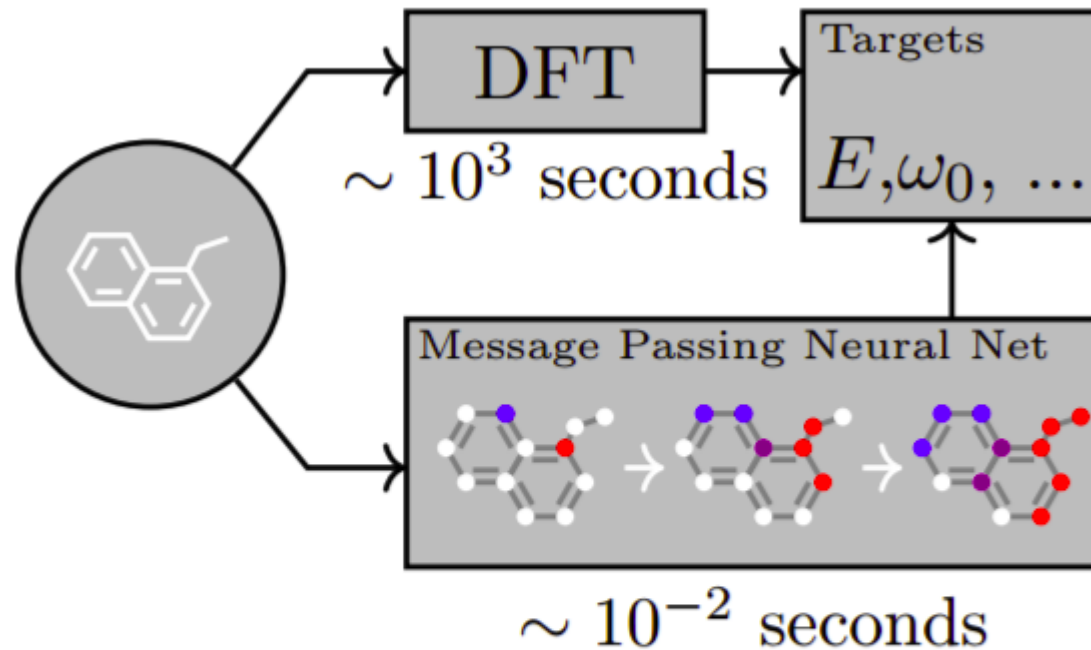


Figure 1. A Message Passing Neural Network predicts quantum properties of an organic molecule by modeling a computationally expensive DFT calculation.

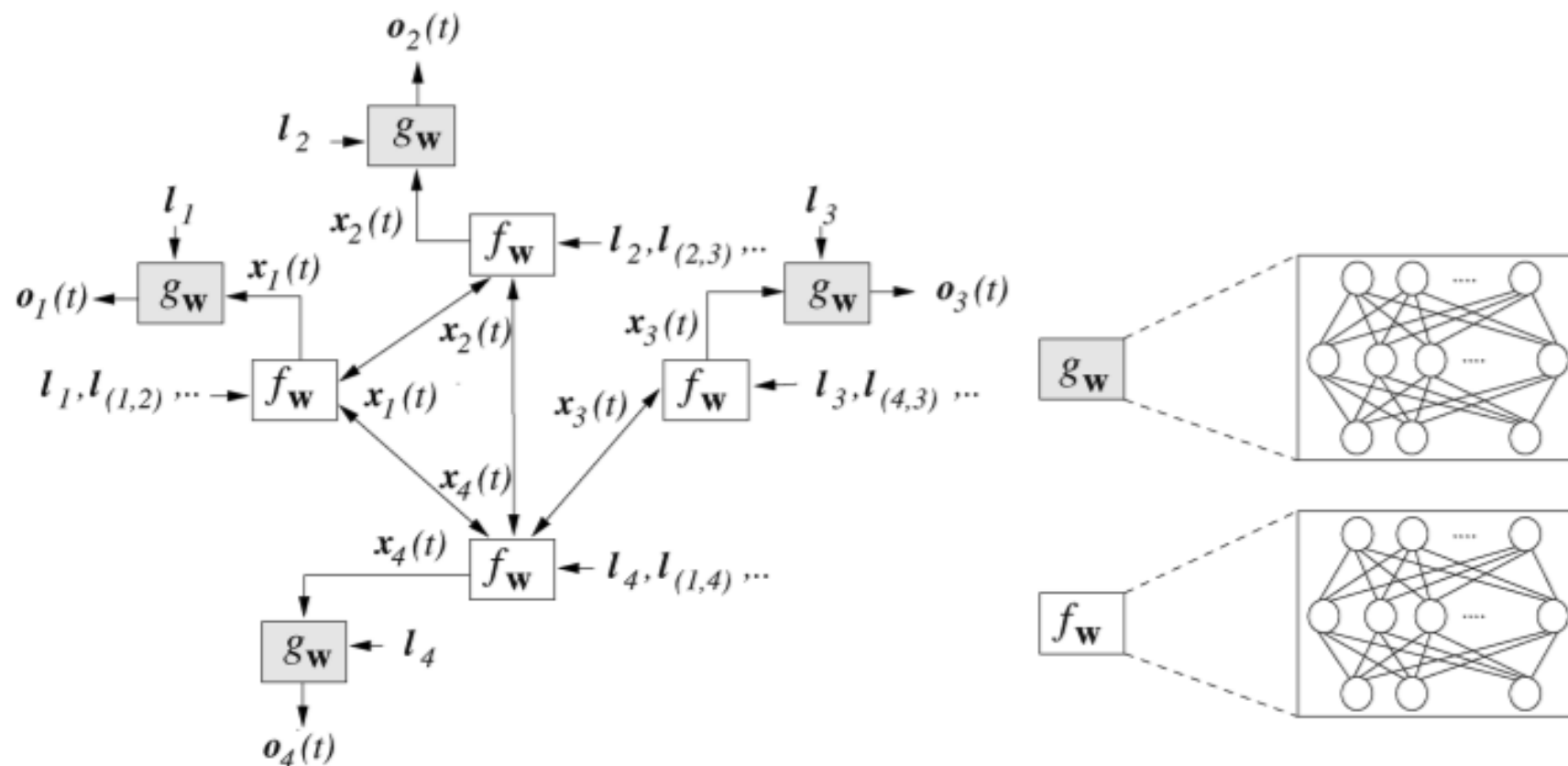
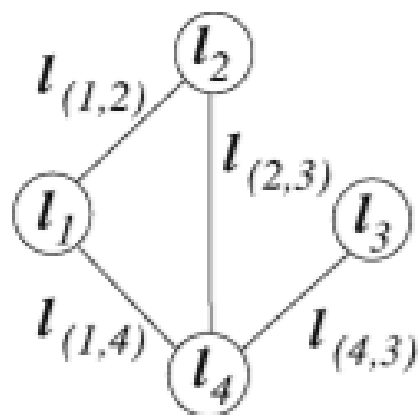
手法

Message Passing Neural Networks

forward phaseは2つのphaseが存在

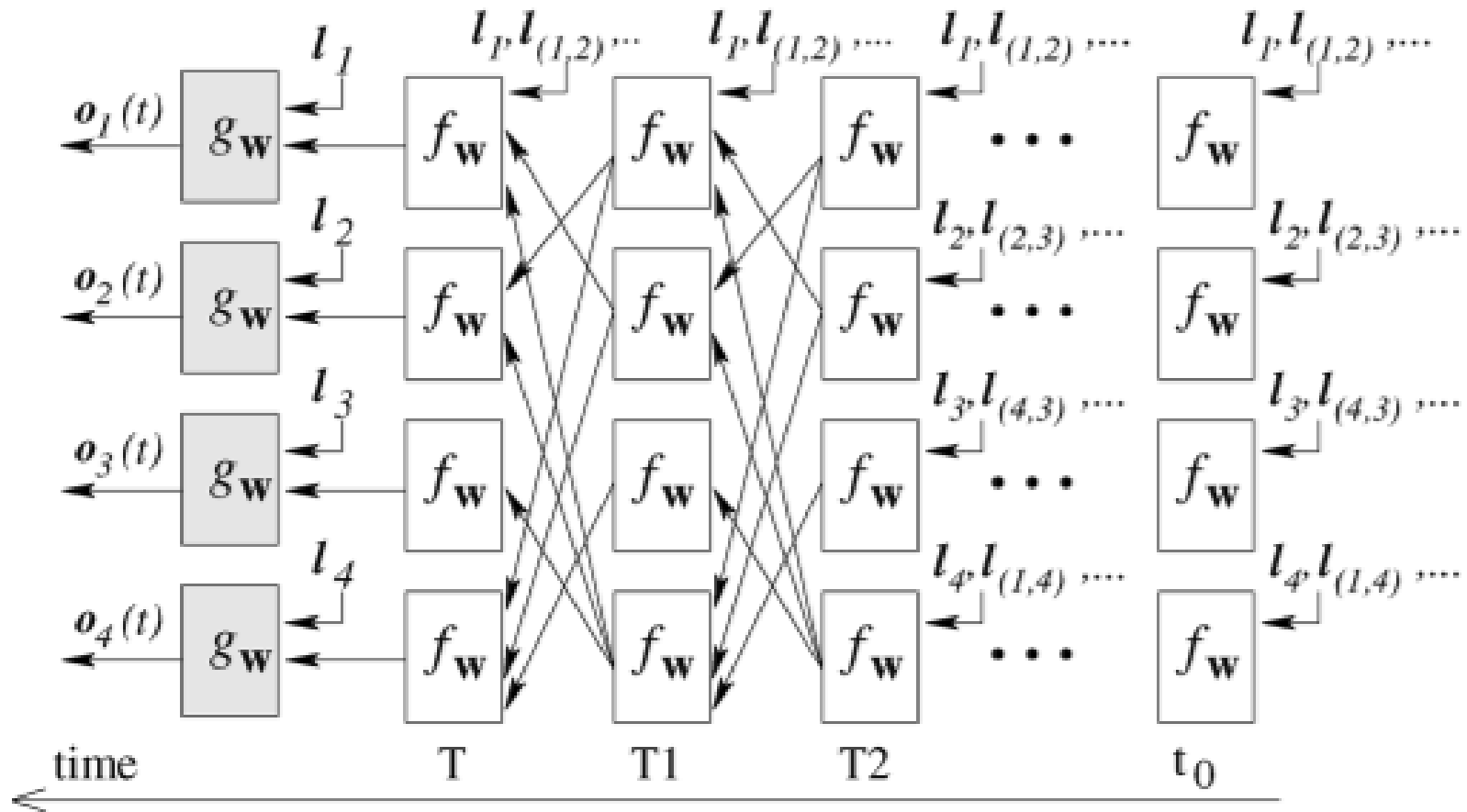
- message passing phase
 - T time steps
 - message functions: M_t
 - vertex update functions: U_t
 - 各ノードの隠れ状態: h_v^t
 - messages: m_v^t
- readout phase

ネットワークイメージ



引用: The graph neural network model, Scarselli+, '09

ネットワークイメージ



引用: The graph neural network model, Scarselli+, '09

Message Passing Neural Networks

Notation

- G : undirected graphs
- x_v : node features
- e_{vw} : edge features

Message Passing Neural Networks

message update

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

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

$N(v)$ denotes the neighbors of v in graph G

エッジの隠れ状態も $h_{e_{vw}}^t$ として同じように扱う事ができる (Kearnes et al. (2016) 参照)

Message Passing Neural Networks

readout phase

$$\hat{y} = R(\{h_v^T | v \in G\})$$

R: readout function

Message Passing Neural Networks

- 以下の手法をMPNNとして考える
 - Convolutional Networks for Learning Molecular Fingerprints
 - Gated Graph Neural Networks (GG-NN)
 - Interaction Networks
 - Molecular Graph Convolutions
 - Laplacian Based Methods

Message Passing Neural Networks

Convolutional Networks for Learning Molecular Fingerprints

- message function

$$M(h_v, h_w, e_{vw}) = (h_w, e_{vw})$$

(., .) denotes concatenation

- update function

$$U_t(h_v^t, m_v^{t+1}) = \sigma(H_t^{deg(v)} m_v^{t+1})$$

σ : シグモイド関数

$deg(v)$: ノードvの次数

H_t^N : 時間tの時の学習済みのマトリクス(Nはノードの時数)

Convolutional Networks for Learning Molecular

R はskip connectionを持っている

すべての隠れ層 h_v^t に対して

$$R = f\left(\sum_{v,t} \text{softmax}(W_t h_v^t)\right)$$

f : neural network

W_t : time step毎の学習済みのreadout matrices

- エッジとノードをそれぞれで足しこんでconcat
- したがって、エッジとノードを区別することが出来ないのが欠点

$$m_v^{t+1} = \left(\sum h_w^t, \sum e_{vw}\right)$$

Message Passing Neural Networks

Gated Graph Neural Networks (GG-NN)

- message function

$$M(h_v, h_w, e_{vw}) = A_{e_{vw}} h_w^t$$

$A_{e_{vw}}$ is a learned matrix, one for each edge label e (the model assumes discrete edge types)

- vertex update function

$$U_t = GRU(h_v^t, m_v^{t+1})$$

GRU(Gated Recurrent Unit)はweight tying的な役割で使用

Gated Graph Neural Networks (GG-NN)

- readout function

$$R = \sum_{v \in V} \sigma(i(h_v^{(T)}, h_v^0)) \odot j(h_v^{(T)})$$

i and j are neural networks, and \odot denotes elementwise multiplication

Message Passing Neural Networks

Interaction Networks

- message function: $M(h_v, x_v, m_v)$
 - 入力が (h_v, h_w, e_{vw}) の Neural Network
 - (\cdot, \cdot, \cdot) は concat
- vertex update function: $U(h_v, x_v, m_v)$
 - 入力が (h_v, x_v, m_v) の Neural Network

Interaction Networks

- readout function
 - 入力が最終的な隠れ状態である h_v^T を入力とした Neural Network

$$R = f\left(\sum_{v \in G} h_v^T\right)$$

originalのモデルではT=1でうまく学習/推定可能（らしい）

Message Passing Neural Networks

Molecular Graph Convolutions

一般的なMPNNsからすこし逸脱した手法。

- message function

$$M(h_v^t, h_w^t, e_{vw}^t) = e_{vw}^t$$

- vertex update function

$$U_t(h_v^t, m_v^{t+1}) = \text{ReLU}(W_1(\alpha(W_0 h_v^t), m_v^{t+1}))$$

- edge state update

$$e_{vw}^{t+1} = U'_t(e_{vw}^t, h_v^t, h_w^t) = \text{ReLU}(W_4(\text{ReLU}(W_2, e_{vw}^t), \text{ReLU}(W_3(h_v^t, h_w^t))))$$

where the W_i are also learned weight matrices

Message Passing Neural Networks

Laplacian Based Methods

- message function

$$M_t(h_v^t, h_w^t) = C_{vw}^t h_w^t$$

where the matrices C_{vw}^t are parameterized by the eigenvectors of the graph laplacian L

$$c_{vw} = (\deg(v)\deg(w))^{-1/2} A_{vw}$$

real valued adjacency matrix A

Laplacian Based Methods

- vertex update function

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

σ is a chosen non-linearity (such as ReLU)

下の構造がKipf & Welling (2016)では使われている

$$U_v^t(h_v^t, m_v^{t+1}) = \text{ReLU}(W^t m_v^{t+1})$$

Message Passing Neural Networks

- 上で説明したアプローチの欠点
 - 計算コストが高い
- 解決策の一つ(Marino et al, 2016参照)
 - 各時間ステップでグラフのサブセットのみにメッセージを渡す
 - これにより、GG-NNアーキテクチャをより大きなグラフに適合

QM9 Dataset

データセット内に含まれている原子

- Hydrogen (H)
- Carbon (C)
- Oxygen (O)
- Nitrogen(N)
- Flourine (F)
- up to 9 heavy (non Hydrogen) atoms

化合物種類

- 134k druglike organic molecules that span a wide range of chemistry

有機物特性

1. the atomization energy at $0K$ U_0 (eV)
2. atomization energy at room temperature U (eV)
3. enthalpy of atomization at room temperature H (eV)
4. free energy of atomization G (eV)
5. highest fundamental vibrational frequency ω_1 (cm^{-1})
6. zero point vibrational energy (ZPVE)(eV)
7. highest occupied molecular orbital (HOMO) ϵ_{HOMO} (eV)
8. the energy of the lowest unoccupied molecular orbital (LUMO) ϵ_{LUMO} (eV)
9. the electron energy gap ($\Delta\epsilon$ (eV))
10. The electron energy gap is simply the difference $\epsilon_{HOMO} - \epsilon_{LUMO}$
11. the electronic spatial extent $\langle R^2 \rangle$ ($Bohr^2$)
12. the norm of the dipole moment μ (Debye)
13. the norm of static polarizability α ($Bohr^3$)

MPNNをイジリオす

これ以降のnotation

- d : 各ノードの隠れ層の次元
- n : グラフ内のノード数

実験概要

- GG-NNをベースラインとして考える
- 様々なmessage function, output functionを試した
- 適切な入力表現を探した
- 適切なハイパーパラメタを探した
- 有向グラフとして計算した
 - message channel数が2倍になる $=2d$ になる

MPNNモデルへの入力

- グラフのノードに対する特徴ベクトルのセット x_v
- 結合の種類、原子間の距離を含んだ隣接行列 A
- 初期隠れ状態 h_v^0 は原子入力特徴ベクトル x_v で設定
 - 次元 d まで埋める
- 各時間ステップ t で重みを共有
- update function では GRU を使用

Message Functionsの実験

3種類つかってイジってみた

Matrix Multiplication

- $M(h_v, h_w, e_{vw}) = A_{e_{vw}} h_w$

Edge Network

- $M(h_v, h_w, e_{vw}) = A(e_{vw})h_w$
 - $A(e_{vw})$ はニューラルネットワーク
 - $d \times d$ の形に変形する

Message Functionsの実験

Pair Message

- matrix multiplication ruleの特徴
 - ノード w からノード v へのメッセージが隠れ状態 h_w およびエッジ e_{vw} の関数のみ
 - 隠れ状態 h_v^t に依存しない
- 理論的には、メッセージチャネルをより効率的に使用できる
 - ノードメッセージが送信元ノードと送信先ノードの両方 n の情報をよりどころとする場合は
- したがってメッセージ関数の変形を使用することも試みた(Battaglia et al., 2016)
- エッジ e に沿った w から v へのメッセージは、 $m_{wv} = f(h_w^t, h_v^t, e_{vw})$
- f はニューラルネットワーク
- 有向グラフの場合、message functionsは M_{in} と M_{out} の2つを使用

Virtual Graph Elements

- エッジの無い部分に仮想のエッジを作る
- グラフ内のすべてのノードに仮想的に繋がっている潜"master"ノードを考えたモデルも使用した
- パフォーマンスに大きな影響はない $O(|E|d^2 + nd_{master}^2)$

Readout Functions

- 2つ実験した
 - GG-NNで使用されていた関数（論文中式4参照）
 - set2setモデル
 - ORDER MATTERS: SEQUENCE TO SEQUENCE FOR SETS, Vinyals+, '16
 - このモデルを最初に適用

Multiple Towers

- MPNNの問題の一つはスケーラビリティ
 - 非常に計算量が多い $O(n^2 d^2)$
- 対応策
 - h_v^t を使用するのではなく、k次元毎に分割した $h_v^{t,k}$ を使用
 - k個並列にupdate functionが計算できるので、計算効率を上げられる

入力表現

入力の表現も3考えてみた

今回のデータではノードの上限数は29だった

Chemical Graph

- 距離無し
- エッジ情報は結合なし、一重結合、二重結合、三重結合

入力表現

Distance bins

- matrix multiply message functionは離散的なエッジ型
- 原子間の距離を10のbinに分割
 - $[2,6]$ は1を1binとする
 - $[0,2],[6,\infty]$ はまとめて1bin
 - 分割幅は手動で決定した
- 隣接行列
 - 結合された原子の結合タイプ
 - 結合されていない原子の距離
 - 原子間距離は結合タイプによってほぼ決まる
 - 距離の10bin+結合タイプ4=14次元ベクトルの特徴量

入力表現

Raw distance feature

- 隣接行列Aの次元（1ノードが持つベクトルの次元）は5次元
- 内訳は、原子間距離で1次元、結合タイプを表す4次元の1ホットベクトル

Training

ハイパーパラメタ

- 50パターンのハイパーパラメタの探索（一様サンプリング）
- T は $3 \leq T \leq 8$ (経験的に $T \geq 3$ だという感じに学習)
- set2setの計算回数 M : $1 \leq M \leq 12$

学習条件

- 最適化手法: adam
- batch size: 20, N of step: 2million(360epochs)
- initial learning rate: $[1e^{-5}, 5e^{-4}]$ で一様サンプリング
- decay factor F : 線形、学習係数を最終的に初期値の10%から90%にする

Training

データセット

- QM-9 datasetを使用
- 全130,462分子
- validation set: 10,000分子をランダムに選択
 - early stopping, model選択
- test set: 10,000分子をランダムに選択
 - 結果のスコア算出
- training set: 残り
- すべての特性値を0から1に正規化
- loss: 特性値との2乗誤差

Training

原子の特徴

Table 1. Atom Features

Feature	Description
Atom type	H, C, N, O, F (one-hot)
Atomic number	Number of protons (integer)
Partial Charge	Calculated charge (float)
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)

実験結果

アルゴリズムごとの比較

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

Target	BAML	BOB	CM	ECFP4	HDAD	GC	GG-NN	enn-s2s	enn-s2s-ens5
mu	4.34	4.23	4.49	4.82	3.34	0.70	1.22	0.26	0.21
alpha	3.01	2.98	4.33	34.54	1.75	2.27	1.55	1.05	0.75
HOMO	2.20	2.20	3.09	2.89	1.54	1.18	1.17	0.95	0.75
LUMO	2.76	2.74	4.26	3.10	1.96	1.10	1.08	0.88	0.73
gap	3.28	3.41	5.32	3.86	2.49	1.78	1.70	1.56	1.29
R2	3.25	0.80	2.83	90.68	1.35	4.73	3.99	0.22	0.14
ZPVE	3.31	3.40	4.80	241.58	1.91	9.75	2.52	1.44	1.16
U0	1.21	1.43	2.98	85.01	0.58	3.02	0.83	0.43	0.31
U	1.22	1.44	2.99	85.59	0.59	3.16	0.86	0.45	0.31
H	1.22	1.44	2.99	86.21	0.59	3.19	0.81	0.55	0.36
G	1.20	1.42	2.97	78.36	0.59	2.95	0.78	0.42	0.31
Cv	1.64	1.83	2.36	30.29	0.88	1.45	1.19	0.81	0.60
Omega	0.27	0.35	1.32	1.47	0.34	0.32	0.53	0.21	0.22
Average	2.17	2.08	3.37	53.97	1.35	2.59	1.36	0.71	0.55

※ 結果はすべてmean absolute error(MAE)

実験結果

データ構造を変えて試してみた

Table 3. Models Trained Without Spatial Information

Model	Average Error Ratio
GG-NN	3.47
GG-NN + Virtual Edge	2.90
GG-NN + Master Node	2.62
GG-NN + set2set	2.57

実験結果

- joint training: 特性値を一括で推定
- individual training: 特性値1個につき1つのモデルを作成
- tower8: $k=8$ のtower構造にした場合

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