Permutation Invariant Representations for Graph Convolutional Networks

Naveed Haghani[1], Maneesh Singh[2], Radu Balan[1]
[1]University of Maryland, [2]Verisk Analytics

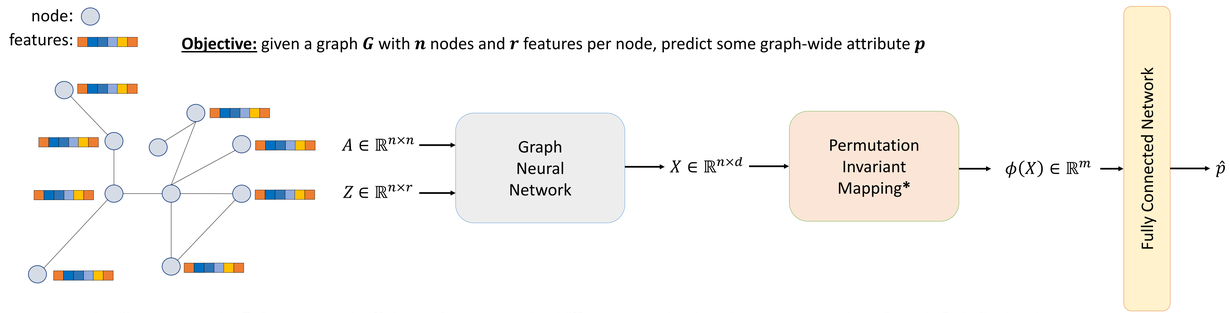
Introduction

We address the problem of **graph classification and graph-wide regression** using **Graph Neural Networks (GNNs)**.

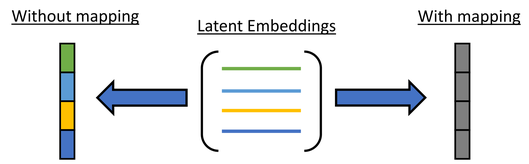
GNNs utilize a **graph's adjacency matrix** to learn a set of **latent node embeddings**. Permuting the order of the nodes in the adjacency leads to a different ordering of the rows in a latent embedding matrix. Therefore, a **network** that produces a single graph-wide estimate over these embeddings **may not be consistent over permutations of the node ordering**.

We introduce a **permutation invariant mapping** that takes a set of node embeddings produced by a GNN and produces an embedding for the entire graph that is **invariant to any permutation of the nodes**.

Rather than learn all possible permutations of every graph in a training set, we **employ a permutation invariant mapping** such that **any node ordering** of a particular graph provides an **identical result**.



A: adjacency matrix, Z: feature matrix, X: latent feature matrix, $\phi(X)$: permutation invariant representation of graph G, $\hat{p}$: final estimate



The embeddings are flattened in preparation to apply a deep network. Without the proposed mapping, the order of this flattened data would depend on our node ordering. With it we produce the same flattened data regardless of the ordering.

Problem

Given a set of node embeddings $X \in \mathbb{R}^{n\times d}$ (where each row corresponds to the latent feature set for a given node), develop a stable and injective mapping that produces an representation $\phi(X) \in \mathbb{R}^m$ that is invariant to row permutations of the input X.

Take the equivalence relation $\sim$ on $\mathbb{R}^{n\times d}$ such that given $V, V' \in \mathbb{R}^{n\times d}$, $V \sim V' if\ \exists \pi \in S_n\ s.t.\ V' = \pi V$

Find $\boldsymbol{\phi}$: $\mathbb{R}^{n\times d} \to \mathbb{R}^m$ such that we have:

a) **Permutation invariance; if $V \sim V'$, then $\phi(V) = \phi(V')$**

b) **Injectivity modulo permutations; if $\phi(V) = \phi(V')$, then $V \sim V'$**

c) **(Upper) Lipschitz: $\|\phi(V) - \phi(V')\|_2 \le L \min_{\pi\in S_n}\{\|V - \pi V'\|_F\}$**

d) **Lower Lipschitz: $L' \min_{\pi\in S_n}\{\|V - \pi V'\|_F\} \le \|\phi(V) - \phi(V')\|_2$**

*Ordering Approach

1. Introduce redundancy into the embeddings by concatenating additional columns out of linear combinations of the rows of Y:

$$\hat{X} = X \cdot [I\ M]$$
$$I: identity\ and\ M: linear\ transformation\ matrix$$

2. Order each column in descending order:

Let $\lambda: \mathbb{R}^n \to \mathbb{R}^n$, $\lambda(x) = \left(x_{\pi(k)}\right)_{k=1}^n$, $where\ x_{\pi(1)} \ge \cdots \ge x_{\pi(n)}$

$$\phi(X)^{(i)} = \lambda\left(\hat{X}^i\right)$$
$$where\ Z^{(i)}\ is\ the\ i^{th}\ column\ of\ Z,\ \hat{Y}^{(i)}\ is\ the\ i^{th}\ column\ of\ \hat{Y}$$

$$\phi(X) = \left(\ \overset{\text{order descending}}{\boxed{\ }}\ X\ \middle|\ X \cdot \mathrm{M}\ \right)$$

Theorem:

- **If $D = 1 + (d-1)n!$ And $F \in \mathbb{R}^{d\times D}$ so that each submatrix of d columns is full rank, the map ϕ: $\mathbb{R}^{n\times d}/\sim \longrightarrow \mathbb{R}^{n\times D}$, $X \mapsto \phi(X) = \downarrow(XF)$ is injective and bi-Lipschitz. In this case $m = (1 + (d-1)n!)n$.**
- **Let ϕ: $\mathbb{R}^{n\times d}/\sim \longrightarrow \mathbb{R}^{n\times(d+1)}$ be given as before with $F = [I_d\ 1]$ a single column of ones concatenated to the identity matrix, then ϕ is Lipschitz everywhere with Lipschitz constant given by the largest singular value of F, and it is injective almost everywhere.**

*Kernel Approach

1. Take a set of m kernel vectors,

$$a_i \in \mathbb{R}^d, for\ i = 1, \dots, m$$

2. Take the following kernel scheme:

$$\nu: \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$$
$$\nu(a_i, t) = \exp\left\{-\frac{\|a_i - t\|^2}{\sigma_i^2}\right\}$$
$$\sigma_i^2 = E_t[\|a_i - t\|^2]$$

3. Let $x^{(i)} \in \mathbb{R}^d$ represent the i^{th} row of X (transposed to a column vector), then we produce $\phi(X) \in \mathbb{R}^m$

$$where\ \phi(X)_i = \sum_{k=1}^n \nu(a_i, x^{(k)}),\ for\ i \in \{1, \dots, D\}$$

$$\phi(X) = \begin{pmatrix} * \\ * \\ * \\ \vdots \\ * \end{pmatrix} \qquad \phi(X)_i = \sum_{k=1}^n e^{\frac{-\|a_i - x^{(k)}\|^2}{\sigma_i^2}}$$

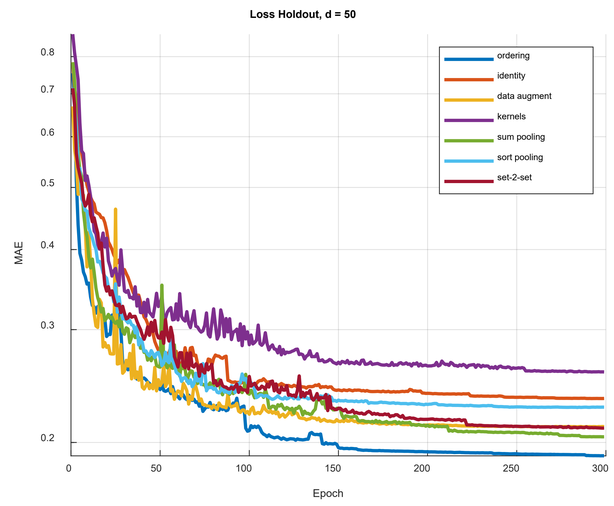
Theorem:

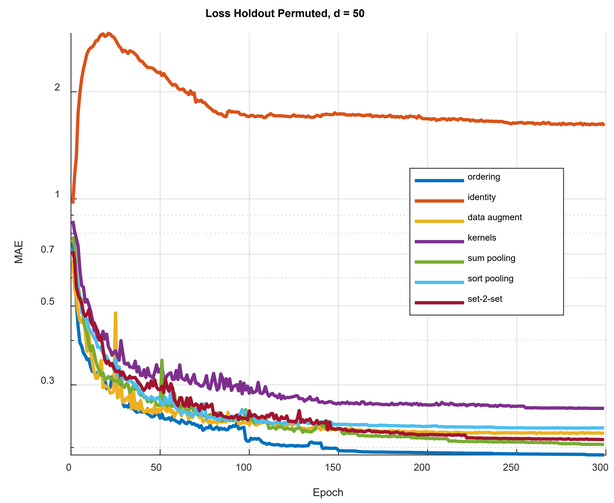
- **Let $\mathbf{K} \subset \mathbb{R}^{n\times d}$ be a compact subset and let ϕ: $K \longrightarrow \mathbb{R}^m$ be the kernel scheme defined above mapping $\mathbf{X} \in K$ to $\phi(X) \in \mathbb{R}^m$, then ϕ is Lipschitz and lifts to a Lipschitz map from $\widehat{K} := \{\widehat{X} \in \mathbb{R}^{n\times d}/\sim, X \in K\}$ to $\mathbb{R}^m$.**

Experiments & Data

Experiments were run on the **qm9** molecular dataset. The data consists of **134 thousand** chemical compounds along with 13 computational derived quantum chemical properties for each compound. We employ our network to perform regression over these values, specifically we look here at the **electron energy gap $\Delta\varepsilon$** (eV).

Results





References

1. Bruna, J.; Zaremba, W.; Szlam, A.; and LeCun, Y.; 2013. Spectral Networks and Deep Locally Connected Networks on Graphs.
2. Defferrard, M., Bresson, X., and Vandergheynst, P. 2016. Convolutional neural networks on graphs with fast localized spectral filtering.
3. Duvenaud, D. K.; Maclaurin, D.; Iparraguirre J.; Bombarell, R.; Hirzel, T.; Aspuru-Guzik, A.; and Adams, R.; 2015. Convolutional networks on graphs for learning molecular fingerprints.
4. Gilmer, J.; Schoenholz, S.; Riley, P. F.; Vinyals, O.; and Dahl, G. E.; 2017. 2017. Neural message passing for quantum chemistry.
5. Hammond, D. K.; Vandergheynst, P.; and Gribonval, R.; 2011. Wavelets on graphs via spectral graph theory.
6. Henaff, M.; Bruna, J.; and LeCun, Y.; 2015. Deep convolutional networks on graph-structured data.
7. Kipf, T.; Welling, M.; 2016. Semi-Supervised Classification with Graph Convolutional Networks.
8. Li, Y.; Tarlow, D.; Brockschmidt, M.; and Zemel, R.; 2015. Gated Graph Sequence Neural Networks.
9. Maron, H.; Ben-Hamu, H.; Shamir, N.; and Lipman, Y.; 2019. Invariant and Equivariant Graph Networks.
10. Qi, C. R.; Su, H.; Mo, K.; and Guibas, L. J.; 2017. Pointnet: Deep learning on point sets for 3d classification and segmentation.
11. Ramakrishnan, R.; Dral, P.; Dral, P.; Rupp, M.; Von Lilienfeld, O. A.; 2014. Quantum chemistry structures and properties of 134 kilo molecules.
12. Vinyals, O.; Bengio, S.; and Kudlur, M. 2015. Order Matters: Sequence to sequence for sets.
13. Zaheer, M.; Kottur, S.; Ravanbakhsh, S.; Poczos, B.; Salakhutdinov, R. R.; and Smola, A. J.; 2017. Deep Sets.
14. Zhang, M.; Cui, Z.; Neumann, M.; and Chen, Y.; 2018. An end-to-end deep learning architecture for graph classification.