Understanding Molecular Data using Graph Neural Networks

Suraj Menon

Stanford
CS229

Motivation

- Problem: To create a probabilistic model that is able to generate novel compounds for drug development
- Solution & Goal: Model molecular datasets as graphs and use graph neural networks to develop a model that has an structural understanding of the data
- Motivation and Results::
- Past GCNs for understanding molecular structure have used solely Euclidean embeddings to represent graph nodes. We consider the possibility that a non-Euclidean embedding may be a better fit to molecular structures.
- We gauge the effectiveness of the model in 'understanding' the data by its accuracy on a node classification task on the molecular dataset.
- With a Euclidean embedding, our GCN is able to achieve a node classification accuracy of 57%. With a hyperbolic embedding the GCN achieves 61%.

Data

- We use the QM-9 database that contains 133,885 molecules. Data comes in SMILES format and is transformed into individual graphs
- Atom types are represented as node features and bond types are represented as different edge types between vertices in a graph.
- For node classification, we perform a random 80/10/10 train/val/test split on the input graph

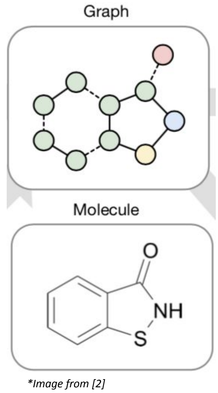


*Image from [2]

Features

- Each graph vertex can be one of 5 atom types and each graph edge can be one of 4 bond types
- The input to the model is a N length vector containing the atom types and a NxN matrix containing the bonds between atoms.
- We represent a large set of molecules with a large sparse matrix that dictate only connections between vertices.

Model: Euclidean GCN

$$\mathcal{G} = (\mathcal{V}, \mathcal{E})$$

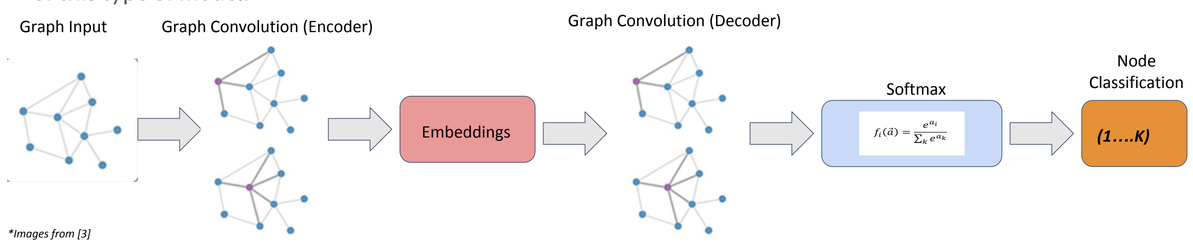
$$f : (\mathcal{V}, \mathcal{E}, (\mathbf{x}_i^{0,E})_{i \in \mathcal{V}}) \to Z \in \mathbb{R}^{|\mathcal{V}| \times d'}$$

$$\mathbf{h}_i^{\ell,E} = W^{\ell} \mathbf{x}_i^{\ell-1,E} + \mathbf{b}^{\ell}$$

$$\mathbf{x}_i^{\ell,E} = \sigma(\mathbf{h}_i^{\ell,E} + \sum_{j \in \mathcal{N}(i)} w_{ij} \mathbf{h}_j^{\ell,E})$$

*Image from [1]

- Graph Input: Set of Vertices and Edges.
- Function: Goal is to transform the set of vertices and edges to a set of embeddings
- Convolution Layer: Each node passes through a linear layer with a weight matrix and bias vector shared among all nodes to get its hidden value. The embedding for a node is then found by aggregating hidden values from adjacent nodes in the graph and passing it through an activation function.
- Output: The embeddings are passed through another convolution layer, then a softmax layer to output the node classification.
- Non-Euclidean Extension: Following the work in [1], we also tested HGCN, a hyperbolic extension of this type of model.



*Images from [3]

Results

- Train for node classification on 5000 molecules from the initial dataset for 100 epochs.
- Train set of 17152 nodes and test set of 2144 nodes

Node Classification Accuracy

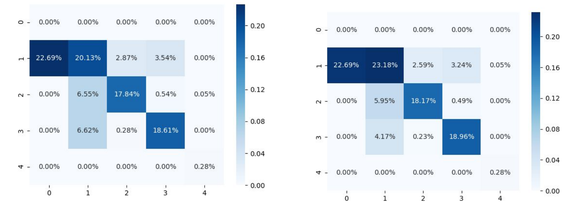
Models	Train Set Accuracy	Test Set Accuracy
Euclidean GCN	99%	57%
Hyperbolic GCN	99%	61%

Discussion

- The node classification task gives out these confusion matrix heatmaps. From this we see a primary confusion between class 0 and class 1.

Confusion Matrices for:

Left : Euclidean GCN
Right: Hyperbolic GCN



- Given our accuracies, we see that we are able to use GCN to understand the structure of molecular data.
- We also see from the slightly better performance that non-Euclidean embeddings may be better suited for representing molecular structures.
- We see that for both Euclidean and non-Euclidean embeddings we are able to overfit on our training set. This implies that our model is able to learn these structures and that our test accuracy may be helped by testing with more data.
- The level of success in this task gives hope of using non-Euclidean GCNs as the base for generative networks that could be used to generate novel compounds.

Future Work

- Next step to expand the GCN to a Relational GCN that makes use of the different bond types.
- Use the expanded RGCN as the basis for the discriminator in a GAN type model to attempt to develop novel compounds.

References & Acknowledgements

[1] Ines Chami, Rex Ying, Christopher Re, Jure Leskovec, Hyperbolic Graph Convolutional Neural Networks. http://web.stanford.edu/~chami/files/hgcn.pdf. 2019.

[2] Nicola De Cao, Thomas Kipf. MolGAN: An implicit generative model for small molecular graphs. https://arxiv.org/pdf/1805.11973.pdf. 2018.

[3] Thomas Kipf, Graph Convolutional Networks, https://tkipf.github.io/graph-convolutional-networks/. 2017

Understanding Molecular Data using Graph Neural Networks

Suraj Menon

Link to Video:

https://youtu.be/rbFAf-w3jTk