Graph Neural Networks
Efficient Tensor Operations In CUDA/GPU
Custom Deep Learning Framework In C++
Applications In Quantum Chemistry

Hy Truong Son, Chris Jones
Advisor: Prof. Risi Kondor

The University of Chicago

December 2017

Table of Contents

Our publication

Covariant Compositional Networks For Learning Graphs (ICLR 2018)

What are Graph Neural Networks?

Learning images, texts

In general, traditional neural networks (Convolutional Neural Networks, Recurrent Neural Networks, etc.) take the inputs as **fixed-size** vectors, matrices and tensors. The architecture of the neural network is always **fixed**.

Learning graphs, molecules

A graph neural network can be defined as a neural network that takes inputs from graphs with **various** sizes and structures. The architecture of the neural network is always **dynamic**.

THE UNIVERSITY OF CHICAGO

What are Tensor Operations?

Basic tensors:

- 1D Tensor: Vectors
- 2D Tensor: Matrices
- 3D Tensor: Cubes

What we are dealing with: 6D Tensors. We need the **tensor contraction** operation C to reduce from the **high-order** into a **low-order**:

$$C : \Re^{d\times d\times d\times d\times d\times c} \rightarrow \Re^{d\times d\times c}$$

What is GraphFlow framework?

We write our own Deep Learning framework name GraphFlow in C++ and CUDA to construct arbitrary neural networks that supports symbolic differentiation and dynamic computation graphs.

- Number of lines of C++ codes: $\sim 55,000$
- Total (including generated codes): $\sim 274,000$
- Done in this quarter for parallelization: $\sim 10,000$

THE UNIVERSITY OF CHICAGO

Why not TensorFlow or PyTorch?

TensorFlow's disadvantage

No (direct) support for dynamic computation graphs.

PyTorch's disadvantage

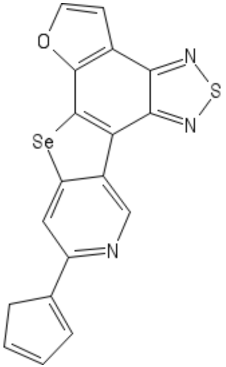
No support for tensor contractions.

GraphFlow's advantage

- Support dynamic computation graphs
- Support tensor contractions

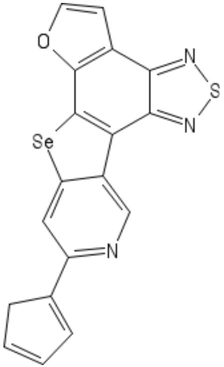
Molecular Chemical Representation

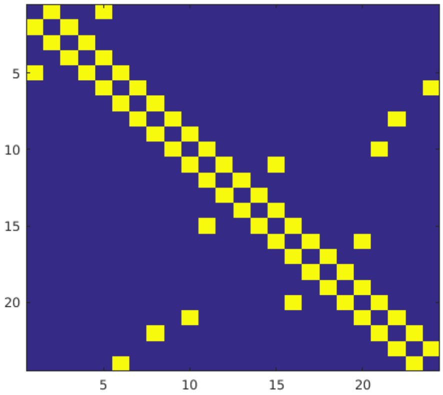
Harvard Clean Energy Project (HCEP) Dataset [2]



Compound: C18H9N3OSSe
SMILES: C1C=CC=C1c1cc2[Se]c3c4occc4c4nsnc4c3c2cn1
Power Conversion Efficiency (PCE, range 0 - 11): 5.16195

Molecular Graph Representation





C18H9N3OSSe

Adjacency matrix

Covariant (Graph) Neural Networks - Part 1

Input graph $G = (V, E)$. Receptive field of vertex $v \in V$ at level $l = 0$ of the network: $\Omega_0(v) = \{v\}$. Receptive field of the vertex at level $l > 0$:

$$\Omega_l(v) = \Omega_{l-1}(v) \cup \bigcup_{w \in B(v,1)} \Omega_{l-1}(w)$$

High-order representation of vertex v at level l:

$$f_l(v) \in \Re^{|\Omega_l(v)| \times |\Omega_l(v)| \times C}$$

where C is the number of channels. We will learn the vertex representation by our graph neural networks with back-propagation. The representation has to be **permutation - invariant**.

Covariant (Graph) Neural Networks - Part 2

$$f_l(v) = \sigma\left(b_l + W_l \otimes \phi\left\{ \bigcup_{w \in B(v,1)} f_{l-1}(w) \right\}\right)$$

where:

- σ is the Leaky ReLU activation function
- $b_l \in \Re^{1\times 1\times C}$ is the learnable bias
- $W_l \in \Re^{C\times(K\cdot C)}$ is the learnable weight matrix
- C is the number of channels, K is number of contractions
- $\otimes$ is the broad-casting matrix-tensor multiplication
- $f_{l-1}(w) \in \Re^{|\Omega_{l-1}(w)|\times|\Omega_{l-1}(w)|\times C}$ is the vertex w representation at level $l-1$
- $\phi\{.\}$ is the combination of tensor product and tensor contraction operations of a set of high-order vertex representations

THE UNIVERSITY OF CHICAGO

Tensor Product and Tensor Contractions

$$\phi\left\{ \bigcup_{w \in B(v,1)} f_{l-1}(w) \right\}$$

can be expressed as:

- We stack the set of 3D tensors $\{f_{l-1}(w)\}$ into a 4D tensor $g_l(v)$
- We make the tensor product between $g_l(v)$ and the reduced adjacency matrix $A_{\Omega_l(v)}$ into a 6D tensor $h_l(v)$
- From the 6D tensor $h_l(v)$, we do the contraction to reduce to a 3D tensor $\phi_l(v)$

By combinatorics, there are exactly $K = 18$ unique ways of tensor contractions.

Virtual Indexing System

Huge tensor

We cannot store a huge tensor $\Re^{|\Omega_l(v)| \times |\Omega_l(v)| \times |\Omega_l(v)| \times |\Omega_l(v)| \times |\Omega_l(v)| \times C}$ in memory explicitly.

Solution: Inspired from Virtual Machine

We will not do the tensor product and tensor contraction directly, but via a **virtual indexing system** that allows us to access the value at position (a, b, c, d, e, f) efficiently.

THE UNIVERSITY OF CHICAGO

GPU Multi-threading

CPU Multi-threading

Performance Test: GPU Matrix Multiplication

Performance Test: GPU Tensor Contractions

Small-scale Molecular Test

Real-world Dataset

Conclusion and Future Research

We implemented the state-of-the-art generalized convolution operation for Graph Neural Networks in order to approximate Density Functional Theory. We obtained very promising results on the Harvard Clean Energy Project dataset.

We are developing our custom Deep Learning framework in CUDA/C++ named **GraphFlow** which supports symbolic differentitation and dynamic computation graph. We expect that this framework will enable us to design more flexible, efficient Graph Neural Networks at a large scale in the future.

Acknowledgements

We would like to acknowledge my advisor Professor Risi Kondor for his valuable instructions and especially for his great ideas in generalizing convolution operations in graphs. We also want to thank other members of Machine Learning group at the University of Chicago for their dedicated support.

Some of the neural network training in this paper was done using the Midway cluster at the UChicago Computing Research Center.

Reference

N. Shervashidze, P. Schweitzer, E. J. van Leeuwen, K. Mehlhorn, K. M. Borgwardt, “Weisfeiler-Lehman Graph Kernels”, *Journal of Machine Learning Research*, vol. 12, 2011.

J. Hachmann, R. O. Amaya, S. A. Evrenk, C. A. Bedolla, R. S. S. Carrera, A. G. Parker, L. Vogt, A. M. Brockway, and A. A. Guzik, “The Harvard Clean Energy Project: Large-Scale Computational Screening and Design of Organic Photovoltaics on the World Community Grid”, *The Journal of Physical Chemistry Letters*, pp. 2241–2251, 2011.

R. I. Kondor, and J. Lafferty, “Diffusion Kernels on Graphs and Other Discrete Structures”, *International Conference on Machine Learning (ICML)*, 2002.

THE UNIVERSITY OF CHICAGO

Reference

N. M. Kriege, P. L. Giscard, R. C. Wilson, “On Valid Optimal Assignment Kernels and Applications to Graph Classification”, *Neural Information Processing Systems (NIPS)*, 2016.

S. Kearnes, K. McCloskey, M. Berndl, V. Pande, P. Riley, “Molecular Graph Convolutions: Moving Beyond Fingerprints”, *Journal of Computer-Aided Molecular Design*, vol. 30, pp. 595–608, 2016.

D. Duvenaud, D. Maclaurin, J. A. Iparraguirre, R. G. Bombarelli, T. Hirzel, A. A. Guzik, R. P. Adams, “Convolutional Networks on Graphs for Learning Molecular Fingerprints”, *Neural Information Processing Systems (NIPS)*, 2015.

Reference

T. N. Kipf, M. Welling, “Semi-Supervised Classification with Graph Convolutional Networks”, *International Conference on Learning Representations (ICLR)*, 2017.

L. V. D. Maaten, G. Hinton, “Visualizing Data using t-SNE”, *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008.

THE UNIVERSITY OF CHICAGO

Thank you very much for your attention!