Critical Review 1: Molecular Graph Convolutions: Moving Beyond Fingerprints

Ben Kotopka, Ali Sharafat, Archa Jain, David Liu

November 14, 2016

1 Problem Statement

Deep learning allows for training on raw representations of data. However, for successful training, this data must still be encoded in a way that reflects relationships among the variables. Encodings for some problems are well-established, such as representing an image as a 2D array of pixel values or representing a DNA sequence with a one-hot vector encoding. However, the encoding problem becomes more difficult when the raw variables are related to one another not by adjacency in a 1D or 2D sequence, but in a graph.

This challenge is a central problem in cheminformatics, which applies machine learning techniques to learn relationships between the properties of molecules – for example, their physical properties, such as their boiling points [1] or their bioactivities, such as their ability to bind a particular protein – from the molecules' atomic structure.

Undirected graphs are an intuitive representation of molecules: each atom in the molecule is a node, and the bonds connecting atoms are the edges. However, most previous work in this area represented relationships between atoms with approaches like "fingerprint" encoding [2], which is fairly simple but does not preserve all the information encoded in a molecular graph.

2 Related Work

The work in the area of molecular representation can be roughly split into three categories of 2D and 3D descriptors as well as graph descriptors. 2D descriptors were defined by starting at an atom and iteratively expanding outwards over a few bonds [2]. The resulting neighborhood is then hashed into a "fingerprint". Such fingerprints are easy to compute but only encode 2D information in a small neighborhood. Other 2D descriptors were later developed based on looking at spanning trees starting at each atom [3] and encoding all pairwise interactions between atoms [4]. Improvements were later made to the above descriptors by considering the 3D structure of the molecules and the electrostatic fields created by the atoms and their respective bonds [5].

The Merck Molecular Activity Challenge spurred the application of DNNs to molecular descriptors. Multitask NNs showed promising results [8]. Duvenaud et al. [9] took a slightly different approach by assigning weights to each atom type-bond type pair and try to learn those weights through construction of a "molecular graph network". Bruna et al. [6] used CNNs on the spectral decomposition of the graphs, but their work was hampered by the fact that the graph structure is subject to change based on the environment. Masci et al. [7] attempted to remedy that defect

by defining non-Euclidean manifolds such that the graph was invariant to small perturbations in shape.

3 Method

The paper focuses largely on creating an alternate representation of molecules using neural nets, rather than the architecture of the classification layer. The inputs A and P are feature vectors for Atoms and Pairs respectively. Each atom's vector is an n-dimensional feature vector associated with each atom. The features are a mix of floating point, integer, and binary values (all encoded as floating point numbers in the network). Worth noting here is that they do minimal preprocessing of these feature vectors, using simple descriptors that are more or less "obvious", to demonstrate that learning can occur with as little preprocessing as possible.

3.1 Feature Weave Modules

This module outputs a modified Atom matrix A, which is calculated via a series of operations between the original A and P inputs (and outputs of intermediate operations). To explain these operations, we need to first understand the invariants the operations are subject to, so the final output is independent of the order of atoms/pairs in the inputs: *Property 1 (Order invariance).* The output of the model should be invariant to the order that the atom and bond information is encoded in the input. *Property 2 (Atom and pair permutation invariance).* The values of an atom layer and pair permute with the original input layer order. *Property 3 (Pair order invariance).* For all pair layers y, $P^y_{(a,b)} = P^y_{(b,a)}$.

The authors involve a series of operations subject to these invariants to allow the layers to operate on each other in every direction ($A \to A, P \to P, A \to P, P \to A$). These operations are then combined to create a new set of A' and P' outputs in a Weave module. These modules are then stacked in layers to create the final output A. Note that the final output from this module is just the A matrix, though the initial and intermediate P features can influence the final atom features through Weave module operations. The output A is then input to the Molecule level feature layer, via one final convolution.

3.2 Molecule level features

To ensure compliance with Property 1, the authors combine the vector of Atom level features to get a vector of Molecule level features. Instead of averaging each feature value from each atom to create a molecule level value (which would disregard the percentage of each individual contribution), they utilize so-called "Fuzzy histograms". They create one such histogram for each feature. Each histogram contains several bins, where the membership of each atom in each bin is an value with range [0,1]. The value of a bin in the histogram is just the sum of the normalized contributions for all the points.

3.3 Model training and evaluation

They built the model with Tensorflow, using AUC over ROC to report performance. They used 5-fold stratified cross validation, with a 60%: 20%: 20% training, validation and testing split. They optimized using the Adagrad optimizer with 10-20M steps.

4 Discussion

Using a graph convolution model with two weave modules, maximum atom pair distance of 2, Gaussian histogram molecule-level reduction, and two fully connected layers, the authors achieved a performance comparable to the baseline PMTNN on classification tasks [8] and published results of neural fingerprint (NFP) models [9] and influence relevance voter (IRV) methods [10]. This is an impressive accomplishment given the simplicity of the input representation.

To investigate the sensitivity of input features, the authors reduced the input features to a subset that match the typical 2D structural diagrams seen in chemistry textbooks: only atom type, bond type and graph distance are provided to the network. Surprisingly, the results achieved using the reduced input representation is very similar to the results achieved using the "full" representation (27 atom features and 12 pair features). This suggests that most of the features in the "full" representation are either mostly ignored during training or can be derived from a simpler representation.

The authors also presented an interesting demonstration of the atom feature evolution through graph convolution. Comparing the initial atom features of a single ibuprofen molecule to their source molecular graph, the aromatic carbons in the central ring are clearly visible. As the atom features are transformed by the weave modules, they become more heterogeneous and reflective of their unique chemical environments.

Hyper-parameter tuning experiments indicated that the number of weave modules is a critical optimization parameter, analogues to the number of hidden layers in traditional neural networks. Experimental data also showed that Gaussian histogram reduction method consistently outperforms the sum reduction method in terms of root-mean-square error. Comparing the performance of several models with different maximum pair distance, the model that uses only adjacent atom pairs achieves the best median AUC score for MUV and Tox21 datasets [11, 12]. This result suggests that graph convolution models implemented in this paper do not effectively make use of the initial graph distance features to preserve or emphasize distance-dependent information.

5 Conclusion and Future Work

The most remarkable accomplishment of this paper is demonstrating that graph convolution model using simple descriptions of molecular graph as input can achieve comparable performance to the state-of-the-art results. Graph convolutions present a new research direction in the field of computer-aided drug design and cheminformatics. One major advantage of the graph convolution architecture is the flexibility of the input representation; graph convolution models are free to use any of the available information for the task at hand, in a sense, every possible molecular "fingerprint" is available to the model.

It would be interesting to further investigate the correlation between features in the "full" feature representation; an idea would be using an auto-encoder to reduce the dimensionality of the "full" feature representation. It is also meaningful to augment the current "full" feature representation to include additional features that captures different aspects of structure of molecules. Furthermore, conducting additional experiments to understand the importance of individual features is also an important activity to pursue.

The choice of Gaussian membership functions and bin selection seems arbitrary; additional optimization through cross validation can potentially lead to better performance. The choice of

using a single final convolution layer also seems arbitrary. Additional experiments with more or less convolution layers can offer better insight into the dynamics of the final convolution process.

The implementation of the $P \rightarrow A$ operation uses a simple sum to combine pair features, such that large amount of information (potentially every pair of atoms in the molecule) is combined in a way that could prevent useful information from being available in later stages of the network. One way to propagate useful information down the network is to explicitly feed the information to later layer.

Lastly, the authors of the paper did not have the chance to fine tune some of the key parameters such as Weave module convolution depths; additional hyper-parameter search can improve the overall performance of the model.

References

[1] Micheli, Alessio. "Neural network for graphs: A contextual constructive approach." *IEEE Transactions on Neural Networks* 20.3 (2009): 498-511.

[2] Rogers, David, and Mathew Hahn. "Extended-connectivity fingerprints." *Journal of chemical information and modeling* 50.5 (2010): 742-754.

[3] Software, OpenEye Scientific. "Cheminformatics and Molecular Modeling Software." *Cheminformatics and Molecular Modeling Software*. N.p., n.d. Web. 14 Nov. 2016.

[4] Carhart, Raymond E., Dennis H. Smith, and R. Venkataraghavan. "Atom pairs as molecular features in structure-activity studies: definition and applications." *Journal of Chemical Information and Computer Sciences* 25.2 (1985): 64-73.

[5] Hawkins, Paul CD, A. Geoffrey Skillman, and Anthony Nicholls. "Comparison of shape-matching and docking as virtual screening tools." *Journal of medicinal chemistry* 50.1 (2007): 74-82.

[6] Bruna, Joan, et al. "Spectral networks and locally connected networks on graphs." *arXiv preprint arXiv:1312.6203* (2013).

[7] Masci, Jonathan, et al. "Geodesic convolutional neural networks on riemannian manifolds." *Proceedings of the IEEE International Conference on Computer Vision Workshops*. 2015.

[8] Ramsundar, Bharath, et al. "Massively multitask networks for drug discovery." *arXiv preprint arXiv:1502.02072* (2015).

[9] Duvenaud, David K., et al. "Convolutional networks on graphs for learning molecular fingerprints." *Advances in Neural Information Processing Systems*. 2015.

[10] Swamidass, S. Joshua, et al. "Influence relevance voting: an accurate and interpretable virtual high throughput screening method." *Journal of chemical information and modeling* 49.4 (2009): 756-766.

[11] Rohrer, Sebastian G., and Knut Baumann. "Maximum unbiased validation (MUV) data sets for virtual screening based on PubChem bioactivity data." *Journal of chemical information and modeling* 49.2 (2009): 169-184.

[12] Mayr, Andreas, et al. "DeepTox: toxicity prediction using deep learning." *Frontiers in Environmental Science* 3 (2016): 80.