# Convolutional Networks on Graphs for Learning Molecular Fingerprints

* Abstract: - these networks allow end-to-end learning of prediction pipelines whose inputs are graphs of arbitrary size and shape. The researchers present an architecture that generalize standard molecular feature extraction methods based on circular fingerprints.
* Introduction: - currently, most machine learning pipelines can only handle inputs of a fixed size.
* The current state of the art is to use off-the-shelf fingerprint software to compute fixed-dimensional features vectors, and use those features as inputs to a fully-connected deep neural network or other standard machine learning method.
* In the research the researchers replace the bottom layer of this stack – the function that computes molecular fingerprint vectors – with a differentiable neural network whose input is a graph representing the original molecule.
* In this graph, vertices represent individual atoms and edges represent bonds.
* It has several advantages over fixed fingerprints: -
  + Predictive performance: - by using data adapting to the task at hand, machine-optimized fingerprints can provide substantially better predictive performance than fixed fingerprint.
  + Parsimony: - fixed fingerprints must be extremely large to encode all possible substructures without overlap.
  + Interpretability: - standard fingerprints encode each possible fragment completely distinctly, with no notion of similarity between fragments. Each feature of a neural graph fingerprint can be activated by similar but distinct molecular fragments, making the feature representation more meaningful.
* Circular fingerprints: - they are a refinement of the Morgan algorithm, designed to encode which substructures are present in a molecule in a way that is invariant to atom-relabeling.
* Circular fingerprints generate each layers’ features by applying a fixed hash function to the concatenated features of the neighborhood in the previous layer.
* The size of the substructures represented by each index depends on the depth of the network. Therefore, the number of layers is referred to as the ‘radius’ of the fingerprints.
* Creating a differentiable fingerprint: - the researchers designed a differentiable generalization of circular fingerprints.
* Hashing: - the hashing functions applied at each layer of circular fingerprints is to combine information about each atom and its neighboring substructures. It insures that any change in fragment, will lead to a different fingerprint index being activated.
* Indexing: - circular fingerprints use an indexing operation to combine all the nodes’ feature vectors into a single fingerprint of the whole molecule. Each node sets a single it of the fingerprint to one, at an index determined by the hash of its feature vector.
* Its pooling-like operation converts an arbitrary-sized graph into a fixed-sized vector.
* The sum of all these classification label vectors produces the final fingerprint. This operation is analogous to the pooling operation in standard convolutional neural networks.



* Canonicalization: - circular fingerprints are identical regardless of the ordering of atoms in each neighborhood. This invariance is achieved by sorting the neighboring atoms according to their feature, and bond features.
* An alternative to canonicalization is to apply a permutation-invariant function, such as summation. In the interests of simplicity and scalability, the researchers chose summation.
* Circular fingerprints can be interpreted as a special case of neural graph fingerprints having large random weights.
* In the above algorithms a fingerprint length *L*, and *F* features at each layer, the parameters of neural graph fingerprints consist of a separate output weight matrix of size *F* x *L* for each layer, as well as a set of hidden-to-hidden weight matrices of size *F* x *F* at each layer, one for each possible number of bonds an atom can have.
* Experiments: - the researchers ran two experiments to demonstrate the neural fingerprints with the large random weights behave similar to circular fingerprints.
* First, they examined whether distance between circular fingerprints were similar to distance between neural fingerprint-based distances.
* Distance was measured using a continuous generalization of the Tanimoto similarity measure, given by



* We can see that there is a correlation of *r* = 0.823 between the distances.
* Second, they examined the predictive performance of neural fingerprints with large random weights vs. that of circular fingerprints.
* The performance of both methods follows similar curves. In contrast, the performance of neural fingerprints with small random weights follows a different curve, and I substantially better.
* It suggests that even with random weights, the relatively smooth activation of neural fingerprints helps generalization performance.
* Examining learned features: - each feature of a circular fingerprint vector can each only be activated by a single fragment of a single radius, except for accidental collisions.
* Neural graph fingerprint features can be activated by variations of the same structure, making them more interpretable, and allowing shorter feature vectors.
* Solubility features: - the fingerprint network was trained as inputs to a linear model predicting solubility, as measured in John S. Delaney paper Estimating aqueous solubility directly from molecular structure.
* Toxicity features: - they also trained the same model architecture to predict toxicity, as measured in two different datasets in Tox21 Challenge. National center for advancing translational sciences.
* Predictive Performance: - the researchers ran several experiments to compare the predictive performance of neural graph fingerprints to that of the standard state-of-the-art setup: circular fingerprints fed into a fully-connected neural network.
* Experimental setup: - the pipeline tasks as input the SMILES string encoding of each molecule, which is then converted into a graph using RDKit.
* In the convolutional networks, the initial atom and bond features were chosen to be similar to those used by ECFP: initial atom features concatenated a one-hot encoding of the atoms’ element, its degree, the number of attached hydrogen atoms, and the implicit valence, and an aromaticity indicator.
* Training and Architecture: - training used batch normalization. The researchers also used tanh and ReLu activation functions for both network layers.
* They also experimented with dropconnect, a variant of dropout in which weights are randomly set to zero instead of hidden units, they found that it led to worse validation error in general.
* Hyperparameter Optimization: - for optimizing hyperparameter they used random search. The following hyperparameters were optimized: log learning rate, log of the initial weight scale, the log *L*2 penalty, fingerprint length, fingerprint depth, and the size of the hidden layer in the fully-connected network.
* Datasets: - the researchers compare the performance of standard circular fingerprints against neural graph fingerprints on a variety of domains:
* Solubility: the aqueous solubility of 1144 molecules as measured by John S. Delaney paper Estimating aqueous solubility directly from molecular structure.
  + Drug efficacy: the half-maximal effective concentration (EC50) *in vitro* of 10,000 molecules against a sulfide-resistant strain of *P. falciparum,* the parasite that causes malaria, as measured in the paper “Thousands of chemical starting points for antimalarial lead identification.”
  + Organic photovoltaic efficiency: the Harvard Clean Energy Project uses expensive DFT simulations to estimate the photovoltaic efficiency of organic molecules. They used a subset of 20,000 molecules from this dataset.
* Predictive accuracy: - the researchers check the performance of circular fingerprints and neural graph fingerprints in two conditions: in the first condition, predictions were made by a linear layer using the fingerprints as input.
* In the second condition, predictions were made by a one-hidden-layer neural network using the fingerprints as input.
* Software: - because the researchers required relatively complex control flow and indexing in order to implement variants of algorithm 2, they used a more flexible automatic differentiation package for python called Autograd.
* Limitations: - Computational cost – computing the neural fingerprints of depth *R*, fingerprint length *L* of a molecule with *N* atoms using a molecular convolutional net having *F* features at each layer costs . In practice, normally training neural networks on top of circular fingerprints usually took several minutes, while training both the fingerprints and the network on top took on the order of an hour on the larger datasets.
* Limited computation at each layer: - it may be fruitful to apply multiple layers of nonlinearities between each message-passing step, or to make information preservation easier by adapting the Long Short-Term Memory architecture to pass information upwards.
* Limited information propagation across the graph: - the local message-passing architecture developed in this paper scales well in the size of the graph, but its ability to propagate information across the graph is limited by the depth of the network.
* The worst case, it can take a depth network of distinguish between graphs of size *N*.
* To avoid this a hierarchical clustering of graph substructures are proposed. A tree-structured network could examine the structure of the entire graph using only log(*N*) layers, but would require learning to parse molecules.
* Inability to distinguish stereoisomers: - Special bookkeeping is required to distinguish between stereoisomers, including enantiomers and *cis/trans* isomers.
* Related Work: - Neural nets for quantitative structure-activity relationship (QSAR) – the modern standard for predicting properties of novel molecular is to compose circular fingerprints with fully-connected neural networks or other regression methods.
* Neural graph fingerprints: - in a neural network which have graph-valued inputs remove all cycles and build the graph into a tree structure, choosing one atom to be the root.
* A recursive neural network is then run from the leaves to the root to produce a fixed-size representation. The final descriptor is a sum of the representations computed by all distinct graphs. There are as many distinct graphs as there are atoms in the network.
* The computational cost of this method thus grows as , where *F* is the size of the feature vector and *N* is the number of atoms, making it less suitable for large molecules.
* Convolutional neural networks: - standard convolutional architectures use a fixed computational graph, making them difficult to apply to objects of varying size or structure, such as molecules.
* Recently, Sergey Ioffe and Christian Szegedy in “Batch normalization: Accelerating deep network training by reducing internal covariate shift” and others have developed a convolutional neural network architecture for modeling sentences of varying length.
* Neural networks on fixed graphs: - Joan Bruna, Wojciench Zaremba, Arthur Szlam, and Yann LeCum in their paper “Spectral networks and locally connected networks on graphs” introduce convolutional networks on graphs in the regime where the graph structure is fixed, and each training example differs only in having different features at the vertices of the same graph.
* The researchers’ network in the contrast addresses the situation where each training input is a different graph.
* Neural networks on input-dependent graphs: - it has an interesting training procedure. The forward pass consists of running a message-passing scheme to equilibrium, a fact which allows the reverse-mode gradient to be computed without storing the entire forward computation.
* The researchers’ model differs from this or this type of model in following ways – their model replaces their complex training algorithms with simple gradient-based optimization, generalizes existing circular fingerprint computations, and applies these networks in the context of modern QSAR pipelines which use neural networks on top of the fingerprints to increases model capacity.
* Unrolled inference algorithms: - sometimes iterative inference procedures resemble the feedforward computation of a recurrent neural network. One natural extension of these ideas is to parameterize each inference step, and train a neural network to approximately match the output of exact inference using only a small number of iterations.
* Conclusion: - the researchers generalized existing hand-crafted molecular features to allow their optimization for diverse tasks.
* They make each operation in the feature pipeline differentiable; they can use standard neural-network training methods to scalably optimize the parameters of these neural molecular fingerprints end-to-end.
* They also demonstrate and predictive performance of these new fingerprints.