Homology Fed Neural Network

Dilan Karaguler

1

https://github.com/DilanKaraguler/cmse802_project.git

Abstract. *This project explores the integration of topological invariants such as betti numbers into graph learning models using topological regularization. In traditional machine learning, graphs are processed through Graph Neural Networks (GNNs), but these methods may not adequately capture higher-order structures like loops, voids, or other topological features that can be essential for graph-based problems. The goal of this project is to incorporate homology as a regularization term in the training of NNs. This regularization will act to preserve certain topological features of the graph during learning, helping the model to respect the underlying topology of the data. This is especially relevant in applications where the graph's topology is a key component of the information being represented, such as in social networks, biological networks, or citation graphs.*

1. Background and Motivation

Mathematical Background: Homology and Path Complexes

In algebraic topology, homology provides a set of topological invariants that can capture important features of a space, such as connectedness, holes, and voids. These features are crucial when analyzing complex structures like molecules, where different molecular properties are encoded in their topological relationships.

In data science, homology-based techniques, such as persistent homology, have been shown to effectively capture the essential features of data that may not be immediately obvious from standard methods like clustering. This is particularly important in the context of molecular data, where subtle interactions and structural properties need to be analyzed in order to predict molecular properties accurately.

Why Use Homology in Molecular Predictions?

Molecules can be represented as graphs or networks, where atoms are nodes and bonds are edges. The topological properties of these molecular graphs contain crucial information about the molecular behavior that can influence various chemical properties. Traditional methods, such as molecular descriptors or physicochemical properties, rely on basic structural features, but may overlook complex interactions that arise from the higher-order topology of the molecule.

Homology of path complexes can be viewed as a more sophisticated way to encode these interactions. By capturing information about the molecule's connectivity, cycles, and higher-dimensional structures, homology provides a set of invariants that are useful for understanding and predicting molecular behavior. These invariants can be used to build more accurate machine learning models, which is where our approach, involving homology-fed neural networks, comes into play.

By feeding the homological features of molecular graphs into a neural network, we aim to exploit the power of these topological invariants to predict molecular properties that

traditional methods may not capture efficiently. This approach allows us to consider complex relationships between atoms and bonds, enhancing the model's ability to predict properties such as dipole moment, atomization energy, and other chemical features.

The Motivation Behind the Project

The motivation for this project stems from the need to develop more accurate and robust models for molecular property prediction. Traditional methods often struggle to account for complex, non-linear relationships between atomic configurations and molecular properties. By incorporating homology-based features, we aim to build a model that leverages the deeper structure of molecules, potentially improving the accuracy and predictive power of machine learning models.

Furthermore, this project seeks to explore the intersection of topology and chemistry. The homological features of molecules offer a novel way to bridge mathematical techniques with chemical prediction tasks, which could lead to new insights in molecular modeling, drug design, and material science.

2. Methodology

2.1. Data Preprocessing

The QM9 dataset was preprocessed before feeding it into the neural network. This preprocessing included:

- Data Sampling (10%): To reduce computational load, we sampled 10% of the dataset due to the high computational demands of calculating path complexes
- Extraction of Location and Atom Type Information: Relevant features such as atomic positions and atom types were extracted for each molecule.
- Computation of Path Complexes: Since no existing library was available, I implemented the code to compute path complexes from the molecular data.
- Computation of Boundary Maps: Similarly, I wrote custom code to compute boundary maps, as no suitable library was available.
- Computation of Betti Numbers: I also developed a custom implementation for calculating Betti numbers, which are used to capture the topological features of the path complexes.

2.2. Model Selection

The model chosen for this task was a fully connected feedforward neural network, designed to learn the complex relationships between molecular descriptors and their corresponding properties. The decision to use this architecture was driven by the simplicity and interpretability of fully connected networks for regression tasks.

Model Architecture: The neural network architecture is as follows:

- Input Layer: Betti numbers (list of length 6 for each molecule)
- Hidden Layer 1: A layer with 64 units and ReLU activation to learn complex patterns from the input.
- Hidden Layer 2: A second layer with 128 units and ReLU activation, allowing the model to capture deeper relationships.
- Hidden Layer 3: A third layer with 64 units and ReLU activation, refining the learned features.
- Output Layer: The final layer has 17 units with a linear activation, predicting 17 continuous properties for each molecule.

The input layer accepts a 6-dimensional feature vector of betti numbers. The two hidden layers use the ReLU activation function to capture non-linear patterns in the data. The output layer consists of 17 neurons (one for each molecular property), and a linear activation is used as we are dealing with regression.

2.3. Training and Optimization

The model was trained using the Adam optimizer, which adapts the learning rate during training. The loss function used was mean squared error (MSE), which is standard for regression problems. The model's performance was evaluated using mean absolute error (MAE), providing a more interpretable measure of the prediction error.

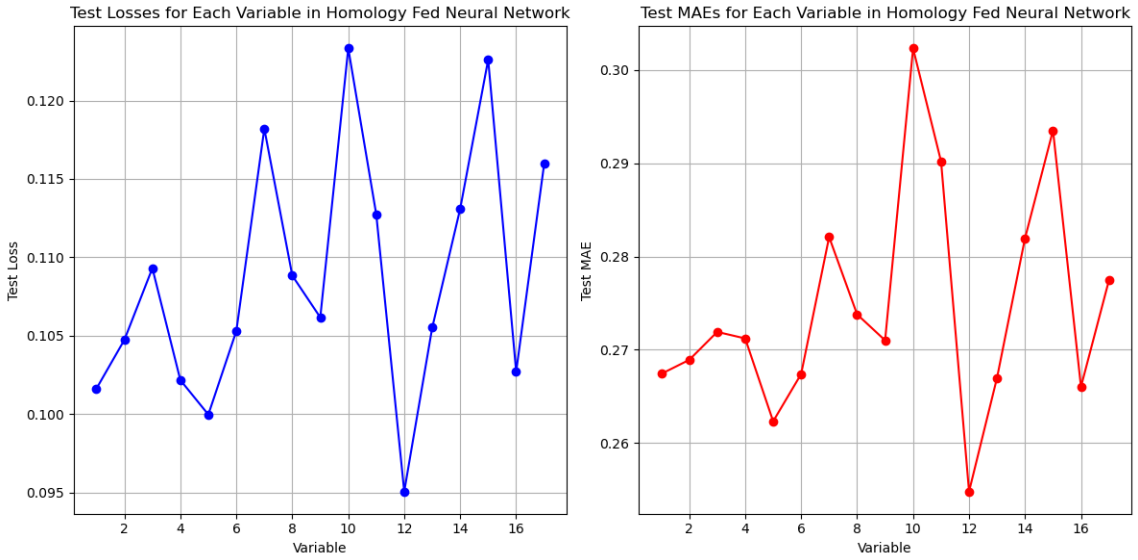
The training data was split into training and validation sets, and the model was trained for several epochs to optimize the weights and biases.

Training Settings:

- Optimizer: Adam
- Loss Function: Mean Squared Error (MSE)
- Metrics: Mean Absolute Error (MAE)
- Epochs: 50

2.4. Model Performance Evaluation

The model's performance was monitored during training by plotting the loss and MAE over each epoch.



We can say that betti summary information worked really well for some classes which are Atomization energy at 0K, Electronic spatial extent etc. This

Here is the list of properties:

Dipole Moment, Isotropic Polarizability, Highest Occupied Molecular Orbital Energy (HOMO), Lowest Unoccupied Molecular Orbital Energy (LUMO), Gap Between HOMO and LUMO, Electronic Spatial Extent, Zero Point Vibrational Energy, Internal Energy at 0K, Internal Energy at 298.15K, Enthalpy at 298.15K, Free Energy at 298.15K, Heat Capacity at 298.15K, Atomization Energy at 0K, Atomization Energy at 298.15K, Atomization Enthalpy at 298.15K, Atomization Free Energy at 298.15K, Rotational Constant, Rotational Constant (2nd value), Rotational Constant (3rd value)

3. Synthesis and Discussion

What did you learn from your results?

From my results, I learned that homology information can indeed provide valuable insights that can potentially reduce the complexity of neural networks. Specifically, persistence and Mayer homology can be used as inputs, with each offering different types of information through their Betti numbers, persistence diagrams, and Mayer homology invariants.

What obstacles did you run into?

The main obstacle I encountered was the unexpectedly high computational complexity, particularly with the persistence computations. This complexity slowed down the process and made it more challenging than anticipated.

What would you do differently next time?

Next time, I plan to focus on improving the computational efficiency of my code, especially for the persistence part. Additionally, I aim to further explore and extend the use of these three invariants (Betti numbers, persistence, and Mayer homology) to gain a deeper understanding of their individual contributions to reducing neural network complexity.

What is the answer to your question(s) and why?

The answer to my question is that homology invariants, particularly Betti numbers, persistence, and Mayer homology, do provide useful information that can help reduce the complexity of neural networks. The challenge lies in optimizing the computational aspects of these methods for practical use, which I hope to address in future work.