Carl King
CS 6964
Progress report

As a reminder, I have been working on building an autoencoder for musical notes with a latent space representation as a simplicial complex. This is a continuation of a project I worked on last semester, wherein I built a wildly inefficient encoder network, where significant effort was spent trying to force outputs from the encoder that satisfied the strict boundary constraints of a simplicial complex, i.e. that given a complex K, every face of a simplex of K must also be in K. I have been to office hours several times to discuss approaches and think I may have finally figured out a method that will be efficient and correct for the chosen size of my complex (20 vertices), but will not scale well to complexes with an arbitrary number. The approach is as follows:

1. Using a series of convolutions, pooling and an MLP layer, transform the16-channel multiband audio signal into a vector of dimension $d = |V| + |E| + |T| + |Tt| = \binom{20}{1} + \binom{20}{2} + \binom{20}{3} + \binom{20}{4} = 6195.$
2. Use hard concrete distribution to transform these values into something resembling probabilities. This is similar to a sigmoid operation, but where there is probability mass on zero and one.
3. Split the output vector into separate components for vertices, edges, triangles, and tetrahedra.
4. Take the log of each of these vectors.
5. Multiply with pre-constructed incidence tensors for a complete simplicial complex of dimension 4 with 20 vertices. For example, to rectify the edge probabilities, take the incidence tensor that represents edge to vertex incidences and multiply it with the vertex probabilities. This gives a new vector of dimension |E|.
6. Take this new vector v and compute $e^{v/(d+1)}$ where d is the dimension of the simplex probabilities to rectify. This effectively computes the geometric mean of the face probabilities for a given simplex.
7. Take the minimum of the output probabilities of the network and the geometric mean of the face probabilities. This enforces that if any of the faces have probability zero, the higher dimensional simplex cannot exist.
8. Use these rectified probabilities to build up the simplicial complex.

The particular simplicial complex neural network I am using for the decoder requires as input a set of adjacency and incidence matrices for each relevant dimension. In office hours you suggested that I work with “soft” complexes and I mistakenly told you that was impossible. After digging around the code for this particular model, I realized that these matrices are only used in a series of matrix multiplications and as such there is nothing obviously impossible about treating the message passing probabilistically, where the messages are weighted based on the probabilities of a given simplex existing. For my use case, I would prefer to have the decoder learn to work with binary matrices, but this can be weakly enforced through the use of a loss term (something like binary entropy on the probabilities of the simplices). Since the purpose of this project is to create a new way to manipulate audio signals, perfect reconstruction is unnecessary. As long as the model performs moderately well at its task, manipulations of simplicial complexes should still create reasonable sounding outputs.

At this point, the encoder code is written, though I have yet to pass in an audio signal to ensure that there are no bugs that will cause my code to crash. On the other hand, the code I wrote to ensure that boundary constraints are satisfied seems to be working quickly and as intended. What remains is to improve my decoder network. I think first I will create embeddings for the vertices and MLPs that will project pairs, triples, and quads of the vertices together to get learned feature embeddings to pass into the Simplicial Complex Convolutional Network (SCCN) (see Yang et al). From that point I could use the architecture as is, but it seems fairly messy and I think I could improve it. This may be worth coming in to talk. The previous architecture output a fixed set of features for each complex since each had the same number of simplices. With the current approach, the number of simplices in the complex can vary dramatically. I could either use a decoder transformer to take in all the features and output as many vectors (or just directly predict the values of the original signal?) or take only the vertex features and use those to reconstruct. Additional background for simplicial complex neural networks and various message passing schemes also given below.

Yang, R., Sala, F. & Bogdan, P.. (2022). Efficient Representation Learning for Higher-Order Data With Simplicial Complexes. <i>Proceedings of the First Learning on Graphs Conference</i>, in <i>Proceedings of Machine Learning Research</i> 198:13:1-13:21 Available from https://proceedings.mlr.press/v198/yang22a.html.

Mathilde Papillon, Sophia Sanborn, Mustafa Hajij, & Nina Miolane. (2024). Architectures of Topological Deep Learning: A Survey of Message-Passing Topological Neural Networks.