Cell Complex Neural Networks (CXNs)

Mustafa Hajij[1], Kyle Istvan[2] ,Ghada Zamzmi[3]

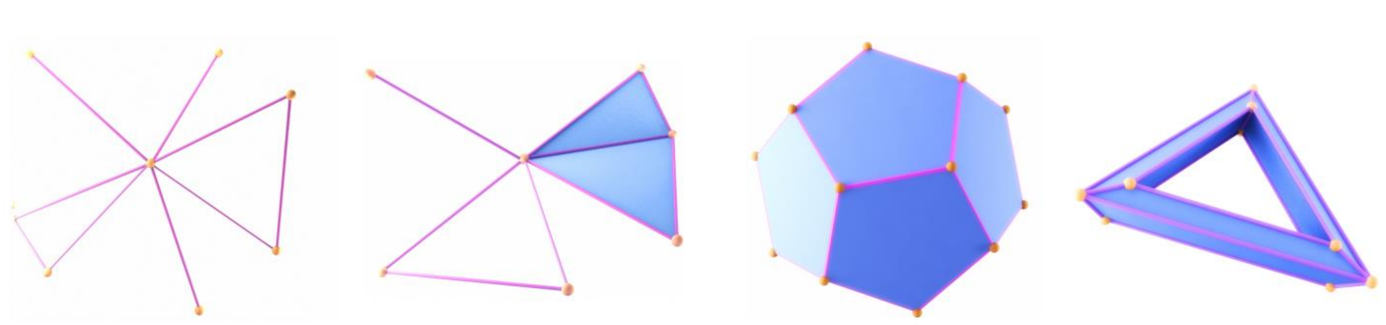
Santa Clara University[1] , KLA Corporation[2], and University of South Florida[3]

Motivation and Contributions

- We propose **Cell Complexes Neural Networks** (**CXNs**), a general, combinatorial, and unifying construction for performing neural network-type computations on cell complexes.
- Cell complexes are topological spaces constructed from simple blocks called cells. Why cell complexes form a better and more expressive generalization?
  - Hierarchical relational reasoning representation
  - Accommodation for most data forms that are significant in practice
  - Trainability of neural networks over geometric domains
  - Resource efficiency
- We introduce an inter-cellular message passing scheme on cell complexes
- We introduce a unified cell complex encoder-decoder framework

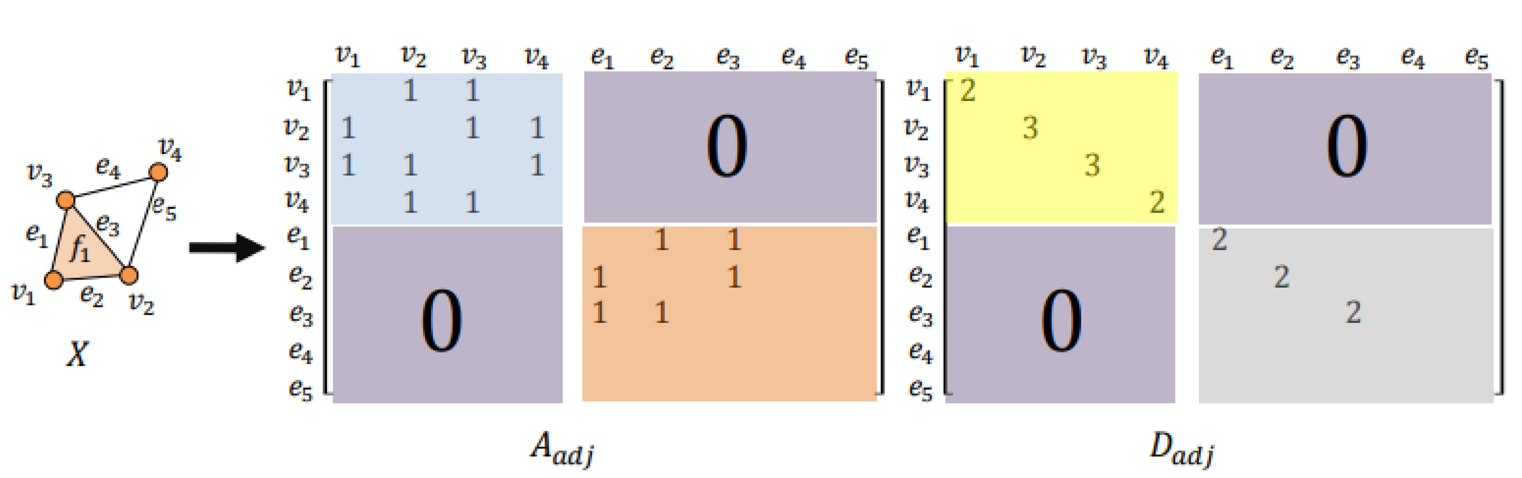
Cell Complex

- A cell complex is a topological space X obtained as a disjoint union of cells, each homeomorphic to the interior of a k-Euclidean ball for some k, attached together via attaching maps in a locally reasonable manner
- Cell complexes provide a combinatorial formalism that allows the inclusion of complicated relationships of restrictive structures
- They form a generalization of graphs, simplicial complexes and polygonal complexes as shown below



Example of cell complexes

- Cell complexes are characterized via their adjacency matrices or coadjacency matrices



Examples of adjacency and degree matrices

Proposed Cell Complex Networks (CXN)

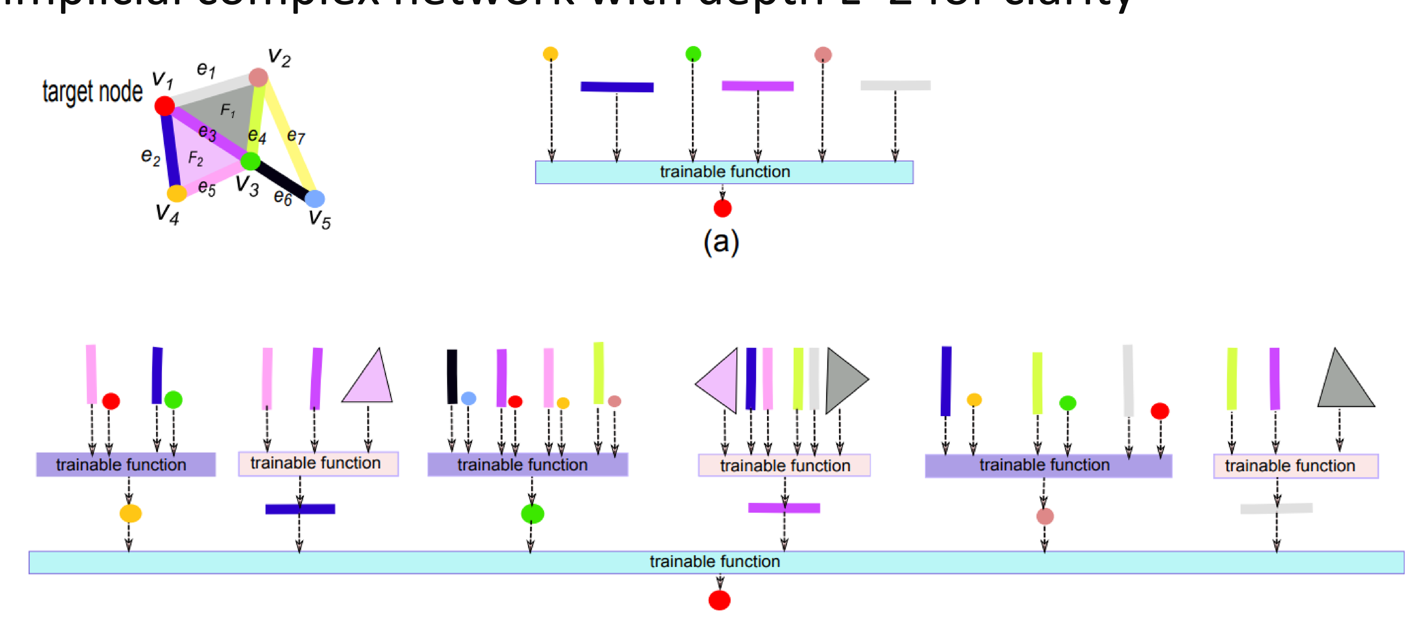
- The forward propagation computation of a cell complex neural net requires the following data as inputs:
  - A cell complex X of dimension n, possibly oriented
  - For each m-cell c m in X, we have an initial vector h (0) cm ∈ R l 0 m
- Given the desired depth L > 0 of the net one wants to define on the complex X, the forward propagation algorithm on X consists of L × n inter-cellular message passing scheme defined for 0 < k ≤ L:

$$h^{(k)}_{c^0} := \alpha^{(k)}_0\Bigg(h^{(k-1)}_{c^0}, E_{a^0\in\mathcal{N}_{adj}(c^0)}\Big(\phi^{(k)}_0(h^{(k-1)}_{c^0},h^{(k-1)}_{a^0}, F_{e^1\in\mathcal{CO}[a^0,c^0]}(h^{(k-1))}_{e^1}))\Big)\Bigg) \in \mathbb{R}^{l^k_0},$$

$$h^{(k)}_{c^{n-1}} := \alpha^{(k)}_{n-1}\Bigg(h^{(k-1)}_{c^{n-1}}, E_{a^{n-1}\in\mathcal{N}_{adj}(c^{n-1})}\Big(\phi^{(k)}_{n-1}(h^{(k-1)}_{c^{n-1}},h^{(k-1)}_{a^{n-1}}, F_{e^n\in\mathcal{CO}[a^{n-1},c^{n-1}]}(h^{(0)}_{e^n}))\Big)\Bigg) \in \mathbb{R}^{l^k_{n-1}}$$

where $h^{(k)}_{e^m}, h^{(k)}_{a^m}, h^{(k)}_{c^m} \in \mathbb{R}^{l^k_m}$, E,F are permutation invariant differentiable functions , and $\alpha^{(k)}_j, \phi^{(k)}_j$ are trainable differentiable functions where, $0 \le j \le n-1$ and $0 < k \le L$.

- The following figure demonstrates the above construction/formulation on a simplicial complex network with depth L=2 for clarity



Info flow goes from lower cells to higher incident cells

- Other message passing schemes that can be defined on cell complexes are presented in the paper

Convolutional Cell Complex Nets (CCXN)

- The input for a CCXN is specified by cell embeddings $H(0) \in R^{N\times d}$ that define the initial cells features on every cell in $X^{<n}$. Here, d is the embedding dimension of the cells. The convolutional message passing scheme on X is defined by

$$H^{(k)} := \mathrm{ReLU}(\hat{A}_{adj} H^{(k-1)} W^{(k-1)})$$

where $\hat{A}_{adj} = I_{\hat{N}} + D^{-1/2}_{adj} A_{adj} D^{-1/2}_{adj}$, $H^{(k)} \in \mathbb{R}^{\hat{N}\times d}$ are the node embeddings computed after k steps of applying 3, and $W^{(k)} \in \mathbb{R}^{d\times d}$ is a trainable weight matrix at the layer k.

Cell Complex Autoencoders (CXNA)

- A cell complex autoencoder (CXNA) consists of three components:
  - An encoder-decoder system, this is the trainable components of the autoencoder
  - A similarity measure on the cell complex, which is auser-defined similarity function that represents a notion of similarity between the cells in the complex
  - A loss function, which is a user-defined function utilized to optimize the encoder-decoder system according to the similarity measure.
- Mathematically, let X be a cell complex of dimension n. Then, an encoder on Xi s a function of the form:

$$enc : X^{<n} \rightarrow \mathbb{R}^d.$$

This encoder associates to every k-cell c^k (0≤k<n) a feature vector $z_c \in R^d$ that encodes the structure of this cell and its relationship to other cells in X. A decoder is a function of the form:

$$dec : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}^+$$

The functions **enc** and **dec** are typically trainable functions that are optimized using a user-defined similarity measure and a user-defined loss function. A similarity function is a function of the form:

$$sim_X : X^{<n} \times X^{<n} \rightarrow \mathbb{R}^+$$

that describes the similarity between cell of the complex. Examples are adjacency matrices. The **enc-dec** functions are optimized such that:

$$dec(enc(a^k), enc(c^l)) = dec(\mathbf{z}_{a^k}, \mathbf{z}_{c^l}) \approx sim_X(a^k, c^l).$$

- To this end, let **l:R×R→R** be a user-defined loss function and define:

$$\mathcal{L}_k = \sum_{\text{all possible } \mathcal{CO}[a^k,c^k]\subset X^{k+1}} l(dec(enc(\mathbf{z}_{a^k}), enc(\mathbf{z}_{c^k})), sim(a^k, c^k)),$$

and finally define: $\mathcal{L} := \sum_{k=0}^{n-1} \mathcal{L}_k$.

Examples: Cell2vec and more

- Several examples of autoencoders can be defined using the above formalism; i.e., if we choose the softmax function to be the decoder and define the loss function to be the cross entropy, then we obtain cell2vec, an encoding scheme of cell complexes inside Euclidian spaces that generalize node2vec. The following table shows more examples:

Table 1: Various definitions of cell complex autoencoders.

Method	Decoder	similarity	Loss
Laplacian eigenmaps [5]	$\|\|\mathbf{z}_a - \mathbf{z}_c\|\|^2_2$	general	$dec(\mathbf{z}_a,\mathbf{z}_c).sim(a,c)$
Inner product methods [1]	$\mathbf{z}_a^T\mathbf{z}_c$	$A_{adj}(a,c)$	$\|\|dec(\mathbf{z}_a,\mathbf{z}_c) - sim(a,c)\|\|^2_2$
Random walk methods [31, 17]	$\frac{e^{\mathbf{z}_a^T\mathbf{z}_c}}{\sum_{b\in X^k} e^{\mathbf{z}_a^T\mathbf{z}_b}}$	$p_X(a\|c)$	$-log(dec(\mathbf{z}_a,\mathbf{z}_c))$