Invariant Graph Networks

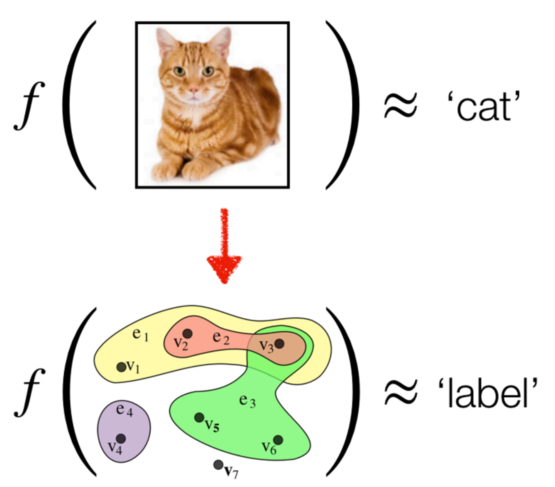
Haggai Maron

Weizmann Institute of Science

Motivation

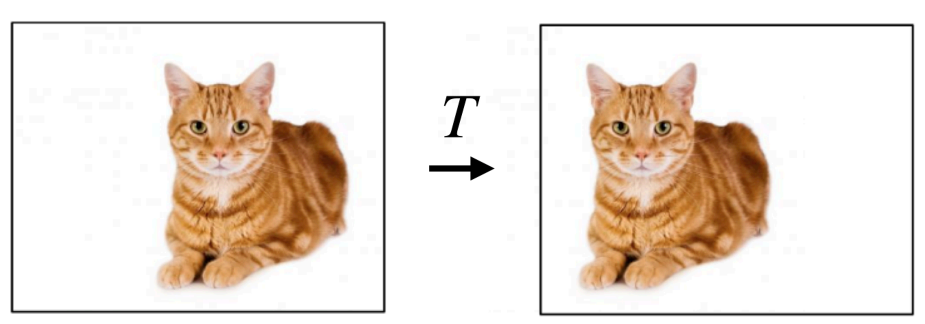
We present a family of neural network architectures suitable for learning irregular data in the form of graphs, or more generally, hypergraphs.

The main idea is to adapt the concept of image convolutions, as a means of dramatically reducing the number of parameters in a neural network, to graph and hypergraph data.

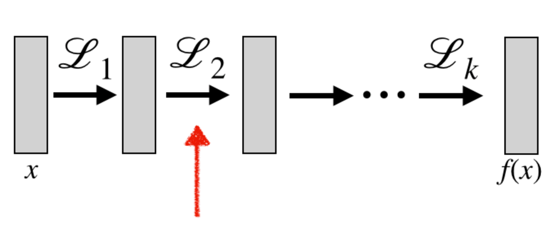


Algebraic view of convolutional neural networks

Translations as image symmetries. Translations of images are transformations T that do not change the image content. Most functions f one is interested to learn on images, like image classification, will be invariant to translations, namely will satisfy $f(x) = f(T \cdot x)$ for all translations T, where x represents the image, and T the application of the translation to the image.

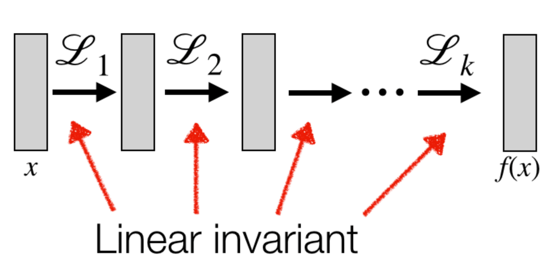


Multi layer Perceptrons. A Multi-Layer perceptron (MLP) is a general-purpose architecture that can approximate any continuous function (on compact sets).

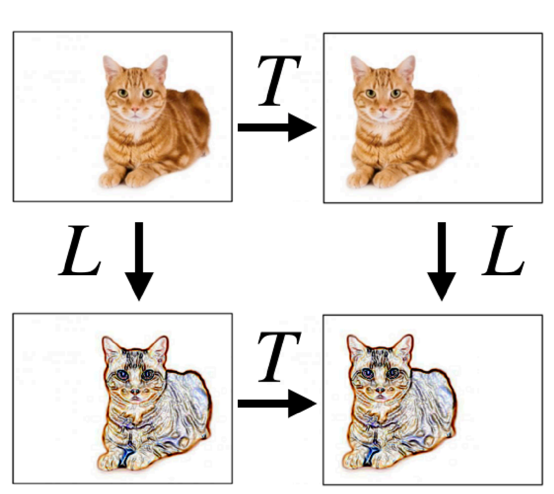


$$\mathscr{L}(x) = \sigma(L(x)) = \sigma(Ax + b)$$

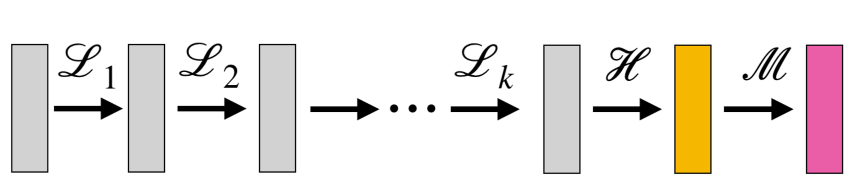
Invariant Model. Motivated by the fact that we are looking to approximate invariant functions, it is reasonable to restrict our models to invariant functions as well. This can be done by replacing the general linear transformations $x \mapsto L(x) = Ax + b$ with linear transformations that are themselves invariant to translations. Sadly, this condition implies that A and b are constant, so these networks lack expressiveness.



Equivariant-Invariant Model. A much more useful idea is to think about equivariant linear operators, namely linear operators that commute with the translations, mathematically satisfying $L(T \cdot x) = T \cdot L(x)$ for all x, T. This condition implies that A is a convolution operator (in fact, equivariance is a defining property of convolutions) and b is a constant vector.

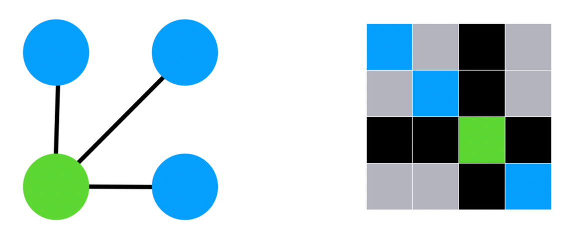


As for images, we will consider neural networks defined by composing several equivariant layers $\mathcal{L}_i$, followed by a single invariant layer, $\mathcal{H}$, and an MLP, $\mathcal{M}$, namely $f(X) = \mathcal{M} \circ \mathcal{H} \circ \mathcal{L}_k \circ \cdots \circ \mathcal{L}_1(X)$.

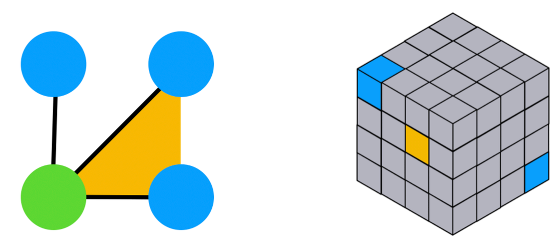


Representing graphs as tensors

A graph can be defined as a set of n elements (nodes) for which we have some information x_i attached to its i-th element, and some information x_{ij} attached to pairs of elements (edges). We will encode this data using a tensor $\mathbf{X} \in \mathbb{R}^{n \times n}$, where the diagonal elements $\mathbf{X}_{i,i} = x_i$ encode the node data and the off-diagonal elements, $\mathbf{X}_{i,j} = x_{ij}$, $i \neq j$, the edge data.

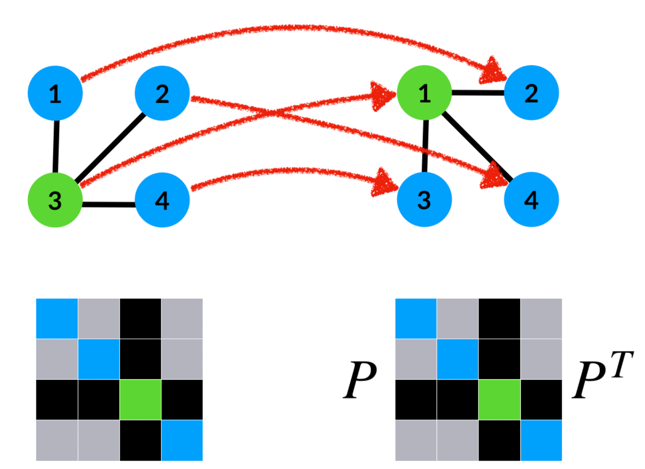


We represent hypergraph data using $\mathbf{X} \in \mathbb{R}^{n^k}$, and each entry $\mathbf{X}_{i_1,i_2,\ldots,i_k}$ represents the information of the corresponding k-tuple of elements.

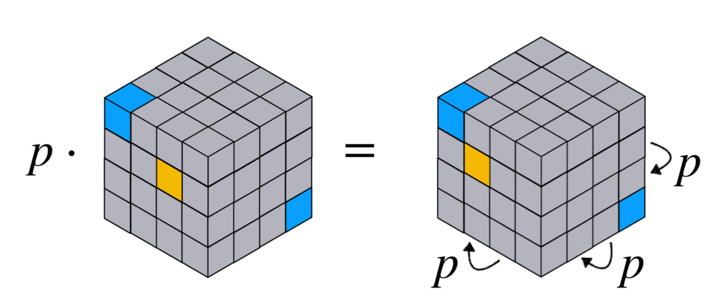


Symmetries of graphs

Transformations that do "not change" the input data will be called symmetries. Two graphs $\mathbf{X}, \mathbf{Y}$ will be considered as the same (a.k.a. isomorphic) if there exists a permutation so that $\mathbf{Y} = p \cdot \mathbf{X}$, where $p \cdot \mathbf{X}$ is a rearrangement of the rows and columns of $\mathbf{X}$ according to p, that is, $(p \cdot \mathbf{X})_{i,j} = \mathbf{X}_{p^{-1}(i),p^{-1}(j)}$.



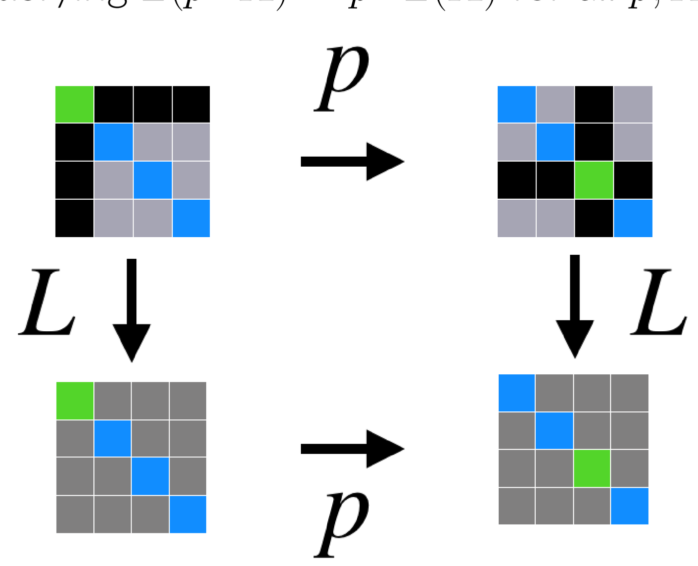
These symmetries generalize to hypergraphs where the permutation p applied to all dimensions of $\mathbf{X} \in \mathbb{R}^{n^k}$, namely $(p \cdot \mathbf{X})_{i_1,\ldots,i_k} = \mathbf{X}_{p^{-1}(i_1),\ldots,p^{-1}(i_k)}$.



Invariant graph networks

Similarly to the image case, trying to consider linear transformations of graph data $\mathbf{X} \in \mathbb{R}^{n \times n}$ that are invariant, i.e., $L(p \cdot \mathbf{X}) = L(\mathbf{X})$, leads to a poor space of operators. Exactly as in the image case, remedy comes from considering (the larger space of) equivariant operators, namely linear operators satisfying $L(p \cdot \mathbf{X}) = p \cdot L(\mathbf{X})$ for all $p, \mathbf{X}$.

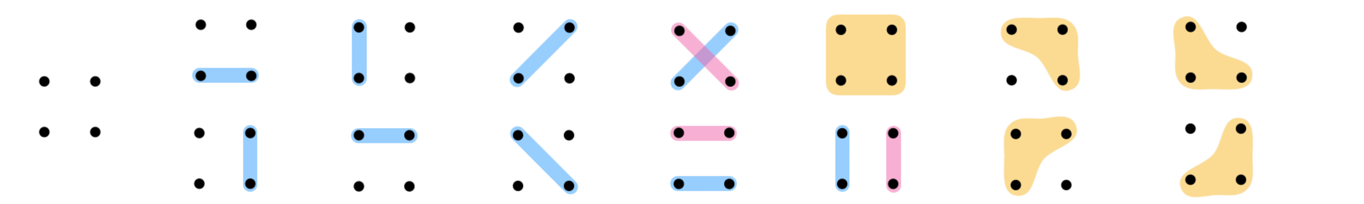
Note that in an IGN the hidden variables can be arbitrary tensors, $\mathbf{Y} \in \mathbb{R}^{n^k}$, even when the input tensors are of a lower order than k. Indeed, equivariant linear operators can map between different order tensors $\mathbb{R}^{n^k} \to \mathbb{R}^{n^l}$.



Linear equivariant operators and the fixed point equations

Fixed point equations. In [2] we are looking to characterize affine transformations $\mathbb{R}^{n^k} \to \mathbb{R}^{n^l}$ equivariant ($l \geq 1$) or invariant ($l = 0$) to the permutation action $\mathbf{X} \mapsto p \cdot \mathbf{X}$, as defined above: a linear transformation $L : \mathbb{R}^{n^k} \to \mathbb{R}^{n^l}$ can be encoded as a tensor $\mathbf{L} \in \mathbb{R}^{n^{k+l}}$. The equations $L(p \cdot \mathbf{X}) = p \cdot L(\mathbf{X})$ can be expressed compactly as $p \cdot \mathbf{L} = \mathbf{L}$.

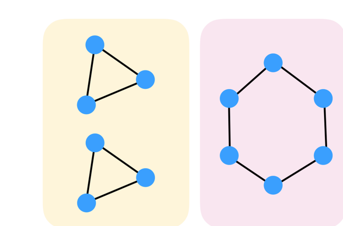
Solving the fixed point equations. Any solution $\mathbf{L}$ is constant along orbits of the permutation group. For example, one can see that $\mathbf{L}_{1,1,2,3}$ is equal to all entries of the form $\mathbf{L}_{i,i,j,s}$, where i, j, s are all different. In general $\mathbf{L}$ is constant along indices i, j, s, t that have the same equality pattern, that is, indices that preserve the equality and inequality relations between pairs of indices. The number of different orbits is the number of different equality patterns on four indices which equals the number of partitions of a set of size four, also known as the *Bell Number*; in this case, $\mathrm{bell}(4) = 15$.



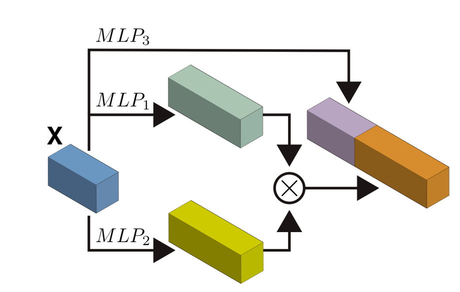
How expressive are IGNs?

Function approximation point of view. In [2] we proved that a 2-IGNs, i.e., an IGN with a maximal tensor degree of 2, can approximate any message passing neural network to an arbitrary precision (on compact sets). Message passing neural networks are very popular models for learning graph data which currently provide state of the art results on various graph learning benchmarks. In [3] we further generalize this result and show that these networks are universal when using sufficiently large tensor order k.

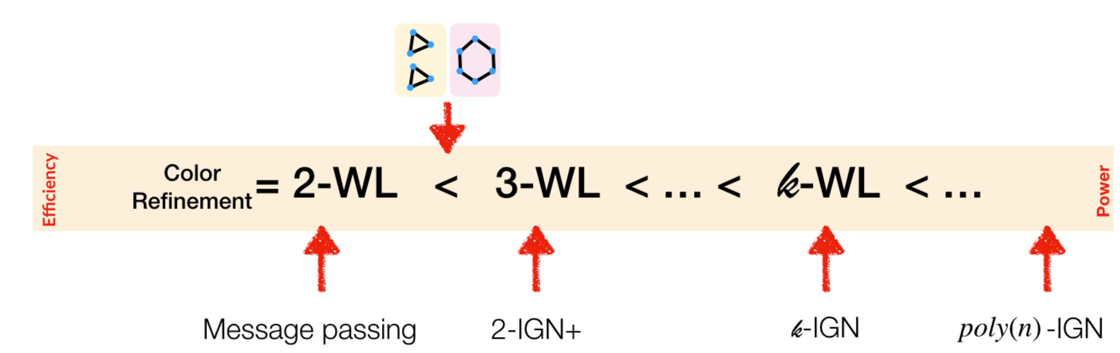
Graph discrimination point of view. k-IGNs are tightly related to a hierarchy of graph isomorphism tests called the Weisfeiler-Leman (WL) hierarchy. The WL hierarchy defines an algorithm, called k-WL, for every $k \in \mathbb{N}$. It can be shown that for any $l > k$, l-WL is strictly more powerful than k-WL. In [1] we prove that k-IGNs can discriminate graphs at least as good as the k-WL algorithm, for every k.



A simple and expressive variant of IGNs. The main drawback of k-IGNs is the fact that one needs to store and process k-order tensors. In [1] we suggest a variant of 2-IGNs that is at least as expressive as the 3-WL test. Hence, strictly more expressive than message passing models.



Summary of the IGN expressiveness. The following figure illustrates the expressiveness results for IGNs. It provides an overview of the main tradeoffs between efficiency and approximation power.



References

[1] Haggai Maron, Heli Ben-Hamu, Hadar Serviansky, and Yaron Lipman. Provably powerful graph networks. *arXiv preprint arXiv:1905.11136*, 2019.

[2] Haggai Maron, Heli Ben-Hamu, Nadav Shamir, and Yaron Lipman. Invariant and equivariant graph networks. In *International Conference on Learning Representations*, 2019.

[3] Haggai Maron, Ethan Fetaya, Nimrod Segol, and Yaron Lipman. On the universality of invariant networks. In *International conference on machine learning*, 2019.