Algebraic Neural Networks: Symmetry and Stability

Alejandro Parada-Mayorga and Alejandro Ribeiro
Electrical and Systems Engineering, University of Pennsylvania
aribeiro@seas.upenn.edu

September 4, 2020

- Structure, or symmetry, is the key to scalable machine learning $\Rightarrow$ Time. Images. Graphs. Groups.

- Convolutions have proven to be the key for leveraging symmetry

  $\Rightarrow$ Convolutional neural networks (NNs). Graph (convolutional) NNs. Group (convolutional) NNs.

- Algeraic signal models and neural networks are an abstraction for convolutional signal processing

  $\Rightarrow$ Of which Cconvolutional (C)NNs, Graph (G)NNs, and Group NNs are particular cases

  $\Rightarrow$ Of which some less well trodden architectures also arise. E.g., Graphon NNs.

- It is more than symmetry. It has has to be. $\Rightarrow$ Otherwise, convolutional linear filters would suffice

  $\Rightarrow$ CNNs are stable to diffeomorphisms. Convolutional filters are not. [Mallat'12]

  $\Rightarrow$ GNNs are stable to perturbations of the graph. Graph filters are not. [Gama-Bruna-Ribeiro'19]

- We will see here that both of these results are manifestations of a common underlying phenomenon

  $\Rightarrow$ Algebraic NNs are stable. Linear algebraic filters are not [Parada-Mayorga-Ribeiro'20]

Parada-Mayorga-Ribeiro, *Algebraic Neural Networks: Stability to Deformations*, TSP 2020, http://arxiv.org/abs/2009.01433

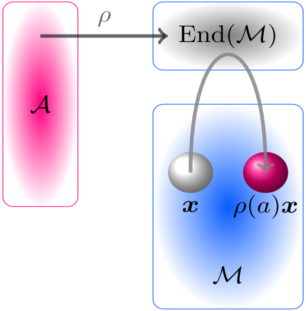
- Algebraic signal processing (ASP) $\Rightarrow$ Foundation of signal processing using representation theory

- An ASP Model is a triplet $(\mathcal{A}, \mathcal{M}, \rho)$
- $\mathcal{A}$ is an Algebra with Unity

  The rules of addition and multiplication
- $\mathcal{M}$ is a vector space

  The space where signals **x** live
- $\rho : \mathcal{A} \to \text{End}(\mathcal{M})$ is a homomorphism

  From $\mathcal{A}$ to the endomorphisms of $\mathcal{M}$



- Any $a \in \mathcal{A}$ is a filter which operates on signals according to the homomorphism $\Rightarrow \mathbf{y} = \rho(a)\mathbf{x}$

- This all sounds very complicated because it is abstract. But is very easy

- Operation rules are given by the algebra of polynomials. Filters are polynomials $\Rightarrow a = \sum_{k=0}^{K-1} h_k t^k$

- The vector space $\mathcal{M} = \ell^2$ is the space of square summable sequences $\Rightarrow \mathbf{x} = \{x_n\}_{n\in\mathbb{Z}}$

- The homomorphism leverages the shift operator S ($(S\mathbf{x})_n = x_{n-1}$) $\Rightarrow \rho(a) = \sum_{k=0}^{K-1} h_k S^k$

- Convolutions are what they should $\Rightarrow \mathbf{y} = \rho\left(\sum_{k=0}^{K-1} h_k t^k\right)\mathbf{x} = \left(\sum_{k=0}^{K-1} h_k S^k\right)\mathbf{x} = \sum_{k=0}^{K-1} h_k S^k \mathbf{x}.$

- But it is also very general $\Rightarrow$ Given a graph with N nodes and matrix representation $\mathbf{S}$
- Operation rules are given by the algebra of polynomials. Filters are polynomials $\Rightarrow a = \sum_{k=0}^{K-1} h_k t^k$
- The vector space $\mathcal{M} = \mathbb{C}^N$ is the space of N-dimensional vectors $\Rightarrow \mathbf{x} = [x_1; \ldots ; x_N)]$
- The homomorphism leverages the shift operator $\mathbf{S}$ ($(\mathbf{S}\mathbf{x})_n = \sum_{r=1}^{N} [\mathbf{S}]_{nr} x_r$) $\Rightarrow \rho(a) = \sum_{k=0}^{K-1} h_k \mathbf{S}^k$
- Convolutions are polynomials on $\mathbf{S}$ $\Rightarrow \mathbf{y} = \rho\left(\sum_{k=0}^{K-1} h_k t^k\right)\mathbf{x} = \left(\sum_{k=0}^{K-1} h_k \mathbf{S}^k\right)\mathbf{x} = \sum_{k=0}^{K-1} h_k \mathbf{S}^k \mathbf{x}.$

- Or, if $W(u, v) : [0,1]^2 \rightarrow \mathbb{R}_+$ is a graphon; a limit object of a graph sequence; we recover WSP

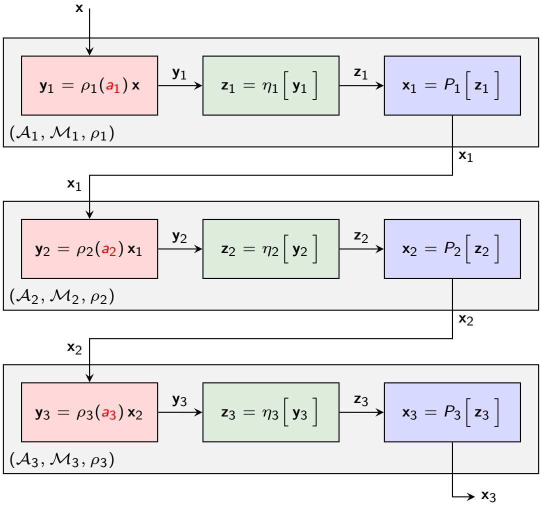
- Operation rules are given by the algebra of polynomials. Filters are polynomials $\Rightarrow a = \sum_{k=0}^{K-1} h_k t^k$

- The space $\mathcal{M} = L^2[0,1]$ contains square integrable functions on $[0,1]$ $\Rightarrow \mathbf{x} = \mathbf{x}(t) : [0,1] \rightarrow \mathbb{R}$

- The homomorphism relies on compositions of the operator $\Rightarrow (S\mathbf{x})(u) = \int_0^1 W(u,v)\mathbf{x}(v)dv$

- Convolutions are polynomials on compositions of S $\Rightarrow \mathbf{y} = \rho\left(\sum_{k=0}^{K-1} h_k t^k\right)\mathbf{x}$

- Algebraic NNs are minor variations of Algebraic filters $\Rightarrow$ Add pointwise nonlinearities and pooling



- Layers uses (possibly different) specific algebraic signal models $\Rightarrow (\mathcal{A}_\ell, \mathcal{M}_\ell, \rho_\ell)$
- Pointwise nonlinearities η operate on individual signal entries
- Pooling operators map the vector space $\mathcal{M}_\ell$ to the vector space $\mathcal{M}_{\ell+1}$
- Trainable parameters are the filters a_ℓ

  Numerically, we train directly on $\rho_\ell(a_\ell)$

- The algebraic models $(\mathcal{A}_\ell, \mathcal{M}_\ell, \rho_\ell)$ determine the equivariance properties of the Algebraic NN
  - ⇒ Equivariance to translations in CNNs
  - ⇒ Equivariance to permutations in GNNs and Graphon NNs
  - ⇒ Equivariance to actions of the group in Group NNs
- These are properties of the filters that the Algebraic NN inherits
  - ⇒ Algebraic NNs outperform Algebraic filters. Why? ⇒ Stability to Deformations

- To define model deformations we need the notion of generator of an algebra

Generators: The set $\mathcal{G} \subseteq \mathcal{A}$ generates $\mathcal{A}$ if all $a \in \mathcal{A}$ are polynomial functions of elements of $\mathcal{G}$

Shift Operators: The set $\mathcal{S}$ of homomorphism images $\mathbf{S} = \rho(g)$ is the set of shift operators

- Definitions of generators and shift operators allows writing filters as polynomials on shift operators

$$\rho(a) = p_{\mathcal{M}}\big(\rho(\mathcal{G})\big) = p_{\mathcal{M}}\big(\mathcal{S}\big) = p\big(\mathcal{S}\big)$$

- We define perturbations of Algebraic models as perturbations of shift operators $\Rightarrow \tilde{\mathbf{S}} = \mathbf{S} + \mathbf{T}(\mathbf{S})$

- The ASP model $(\mathcal{A}, \mathcal{M}, \rho)$ is consequently perturbed to the ASP model $(\mathcal{A}, \mathcal{M}, \tilde{\rho})$ such that

$$\tilde{\rho}(a) = p_{\mathcal{M}}\big(\tilde{\rho}(g)\big) = p_{\mathcal{M}}\big(\tilde{S}\big)$$

  That is, the polynomials that define filters are the same. But they use the perturbed shift operator

- Graphs $\Rightarrow$ Shift operator $\mathbf{S}$ represents a graph $\Rightarrow \tilde{\mathbf{S}}$ represents different graph

- Time $\Rightarrow \mathbf{S}$ represents translation equivariance $\Rightarrow \tilde{\mathbf{S}}$ represents quasi-translation equivariance

Perturbation Model

- We analyze a first order perturbation model of the form $\Rightarrow \mathbf{T}(\mathbf{S}) = \mathbf{T}_0 + \mathbf{T}_1\mathbf{S}$

- The operators $\mathbf{T}_0$ and $\mathbf{T}_1$ are compact normal with norms satisfying $\|\mathbf{T}_0\| \leq 1$ and $\|\mathbf{T}_1\| \leq 1$

- $\mathbf{T}_r$ and $\mathbf{S}$ do not commute. Write $\mathbf{S}\mathbf{T}_r = \mathbf{T}_r\mathbf{S} + \mathbf{S}\mathbf{P}_r$. and define commutation factor $\delta = \max_r \dfrac{\|\mathbf{P}_r\|}{\|\mathbf{T}_r\|}$

Stable Operators: We say operator $p(\mathbf{S})$ is stable if there exist constants C_0, $C_1 > 0$ such that

$$\left\| p(\mathbf{S})\mathbf{x} - p(\tilde{\mathbf{S}})\mathbf{x} \right\| \leq \left[C_0 \sup_{\mathbf{S}\in\mathcal{S}} \left\|\mathbf{T}(\mathbf{S})\right\| + C_1 \sup_{\mathbf{S}\in\mathcal{S}} \left\| D_{\mathbf{T}}(\mathbf{S}) \right\| + \mathcal{O}\left(\left\|\mathbf{T}(\mathbf{S})\right\|^2 \right) \right] \left\|\mathbf{x}\right\|$$

for all $\mathbf{x} \in \mathcal{M}$ and $D_{\mathbf{T}}(\mathbf{S})$ denoting the Fréchet derivative of $\mathbf{T}$.

- $\left\| p(\mathbf{S})\mathbf{x} - p(\tilde{\mathbf{S}})\mathbf{x} \right\|$ is bounded by the size of the deformation. Measured by value and rate of change
- Staility is not a given $\Rightarrow$ Counter examples in GNN and processing of time signals.

- Filters are polynomials on shift operators $\Rightarrow$ Isomorphic to polynomials with complex variables

Lipschitz Filter: Polynomial $p : \mathbb{C} \rightarrow \mathbb{C}$ is Lipschitz if $\|p(\lambda) - p(\mu)\| \leq L_0 \|\lambda - \mu\|$ for some L_0

Integral Lipschitz: Polynomial $p : \mathbb{C} \rightarrow \mathbb{C}$ is Integral Lipschitz if $\left\| \lambda \dfrac{dp(\lambda)}{d\lambda} \right\| \leq L_1$ for some L_1

- Restricted attention to algebras with a single generator. Generalizations are cumbersome but ready

Stability of Algebraic Filters

Theorem

A filter that is Lipschitz and Integral Lipschitz is stable

$$\left\| p(\mathbf{S})\mathbf{x} - p(\tilde{\mathbf{S}})\mathbf{x} \right\| \leq \left[(1+\delta) \left(L_0 \sup_{\mathbf{S}} \|\mathbf{T}(\mathbf{S})\| + L_1 \sup_{\mathbf{S}} \|D_{\mathbf{T}}(\mathbf{S})\| \right) + \mathcal{O}(\|\mathbf{T}(\mathbf{S})\|^2) \right] \|\mathbf{x}\|$$

- Good news $\Rightarrow$ Algebraic filters can be made stable to perturbations
- Alas, We either have stability or discriminability. Integral Lipschitz Filter $\Rightarrow \left\| \lambda \dfrac{dp(\lambda)}{d\lambda} \right\| \leq L_1$
- Commutativity factor affects stability constant but does not generate instability

Stability of Algebraic Filters

Theorem

Let $\Phi_\ell(\mathbf{S}, \mathbf{x})$ and $\Phi_\ell(\tilde{\mathbf{S}}, \mathbf{x})$ be the operators associated with layer ℓ of an Algebraic NN. If the layer filters are Lipschitz and Integral Lipschitz,

$$\left\|\Phi_\ell(\mathbf{S}, \mathbf{x}) - \Phi_\ell(\tilde{\mathbf{S}}, \mathbf{x})\right\| \leq \left[(1+\delta)\left(L_0 \sup_{\mathbf{S}} \|\mathbf{T}(\mathbf{S})\| + L_1 \sup_{\mathbf{S}} \|D_{\mathbf{T}}(\mathbf{S})\|\right) + \mathcal{O}\left(\|\mathbf{T}(\mathbf{S})\|^2\right)\right] \|\mathbf{x}\|$$

- Good news $\Rightarrow$ Algebraic NNs can be made stable to perturbations. It's the same bound
- Individual layers loose discriminability. Integral Lipschitz Filter $\Rightarrow \left\|\lambda \dfrac{dp(\lambda)}{d\lambda}\right\| \leq L_1$
- Nonlinearity mixes frequency components $\Rightarrow$ Recover discriminability in subsequent layers

- Algebraic NNs unify stability analysis of neural networks where convolutions are used.

- Algebraic linear filterscan be either stable **OR** discriminative. But not both

- Algebraic neural networks can be both, stable **AND** discriminative