Universal Invariant and Equivariant Graph Neural Networks

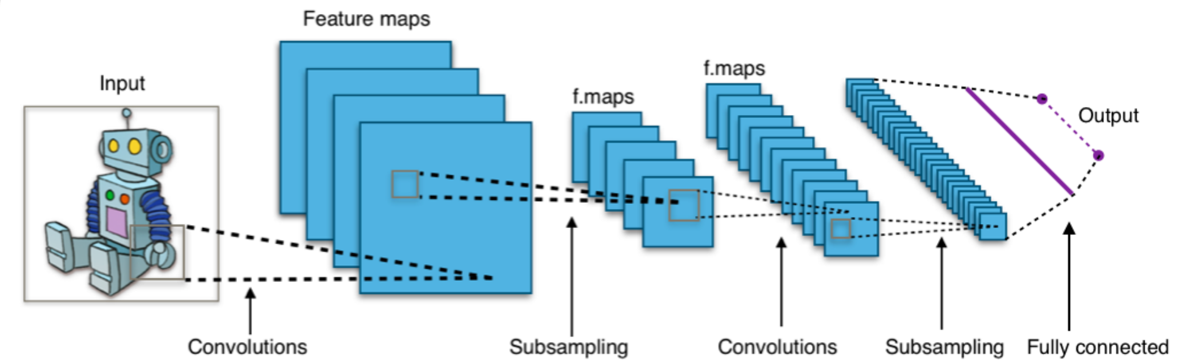
Nicolas Keriven[1], Gabriel Peyré[2]

[1]CNRS, Gipsa-lab

[2]CNRS, ENS Paris

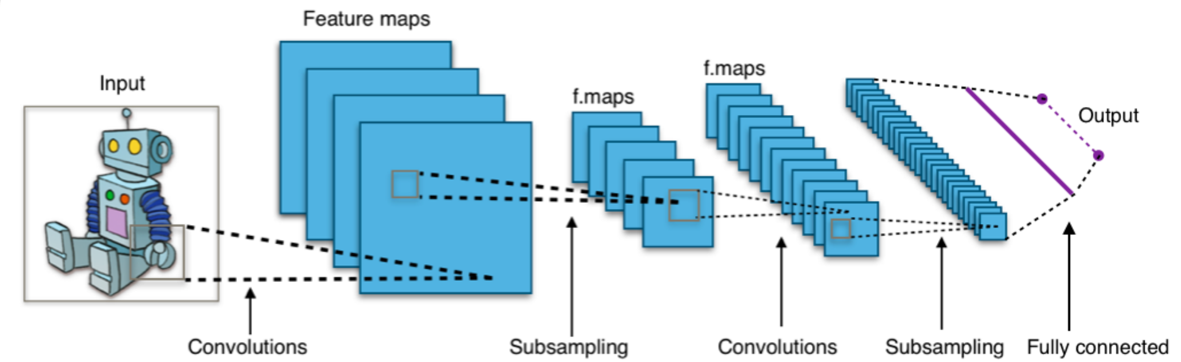
GdR Isis, 2019 Oct. 17th

Deep NN



« Alternate linearities and non-linearities »

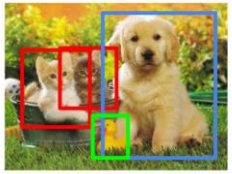
Deep NN



« Alternate linearities and non-linearities »

State-of-the-art in: **most everything ?** *(with sufficient data and domain knowledge…)*

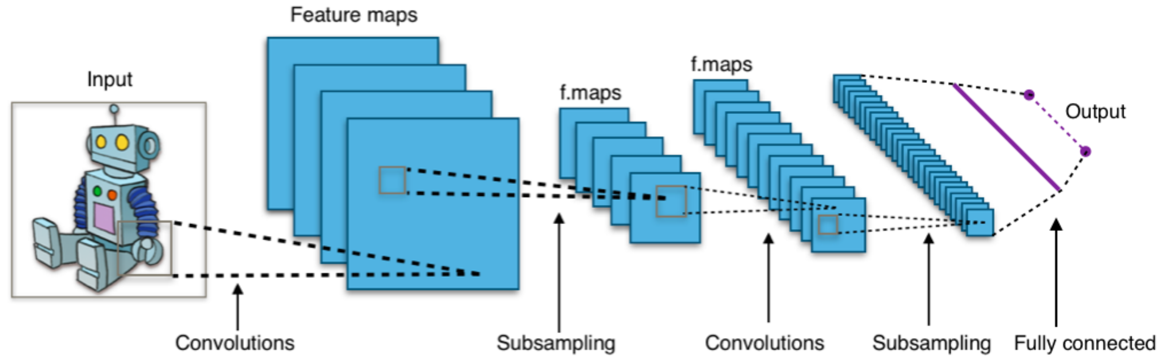
- Computer vision
- Speech recognition
- Natural Language Processing (NLP)
- Recommender systems
- Reinforcement learning (sequential learning)
- Etc etc etc.



CAT, DOG, DUCK

Many specialized *architectures for each application*

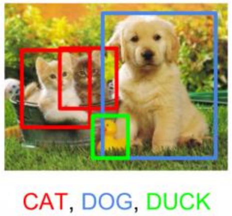
Deep NN



« Alternate linearities and non-linearities »

State-of-the-art in: **most everything ?** *(with sufficient data and domain knowledge…)*

- Computer vision
- Speech recognition
- Natural Language Processing (NLP)
- Recommender systems
- Reinforcement learning (sequential learning)
- Etc etc etc.



***Many specialized** architectures for each application*

Theory ?

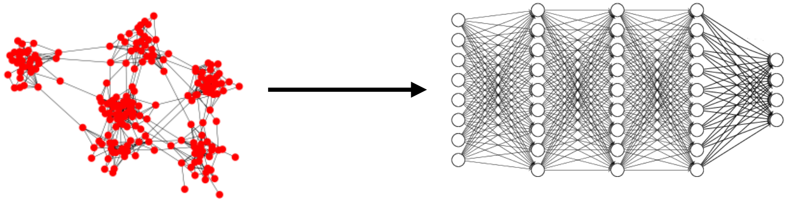
- Approximation theory
  - **Universal Approximation Theorem** *[Hornik 1989, Cybenko 1989, Pinkus 1999…]*
  - Approximation rate / smoothness space *[Cohen, Kutyniok, Gribonval…]*
- Generalisation / Sample complexity *[Barnett, Arora, Neyshabur…]*
- Optimisation / Regularization *[Du, Lee, Bach, Jordan, Montanari…]*

Graph NN

Adapting DNN to graph inputs…

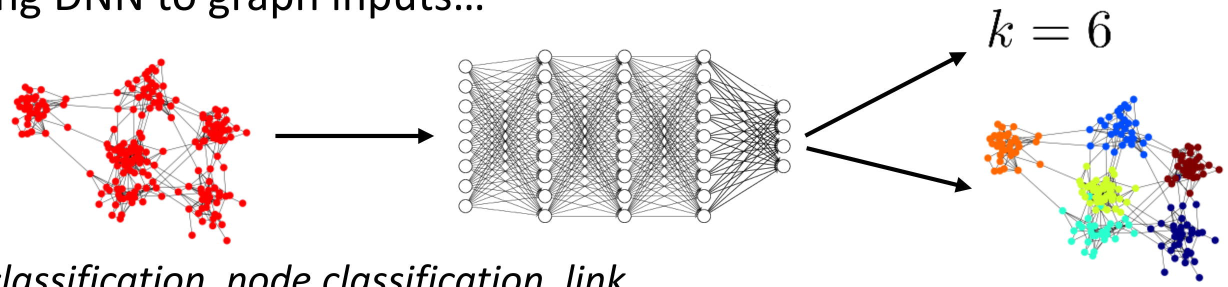
Graph NN

Adapting DNN to graph inputs…



Graph NN

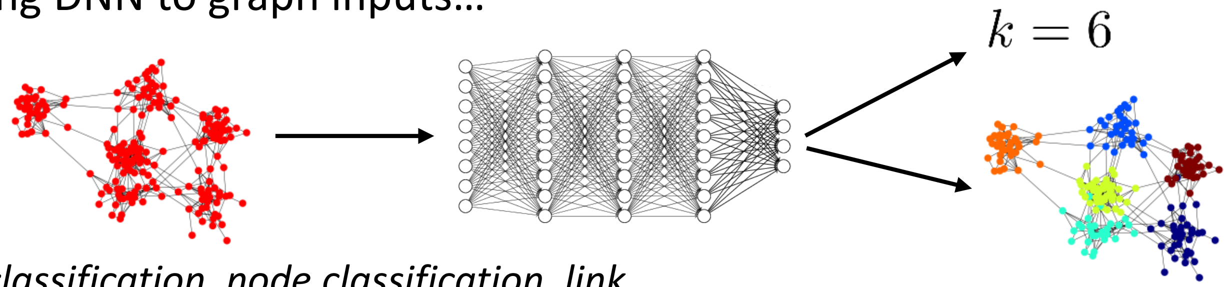
Adapting DNN to graph inputs…

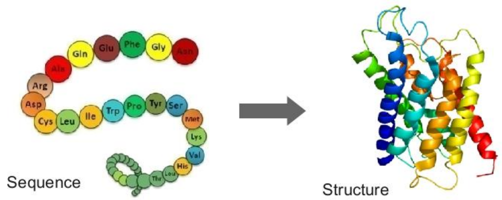


Graph classification, node classification, link prediction…

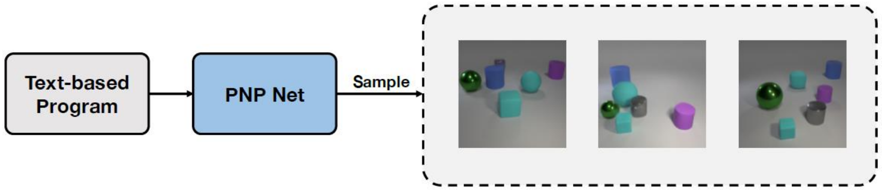
Graph NN

Adapting DNN to graph inputs…





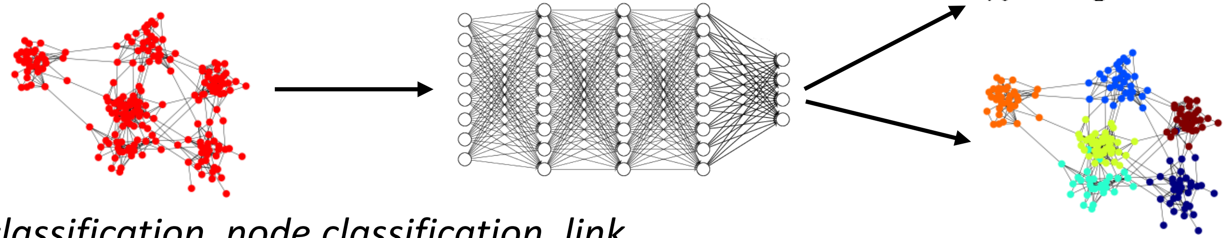
Graph classification, node classification, link prediction…

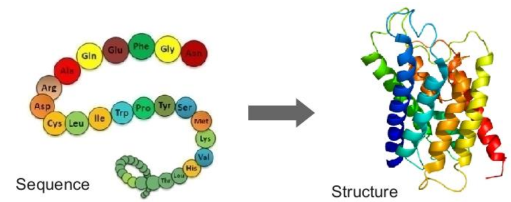


- **Computer vision**: scene generation, point cloud classification, action recognition…
- **NLP**: text classification (semantic graph)
- **Chemistry**: infer molecular properties, protein structure, *synthetize* new compound…

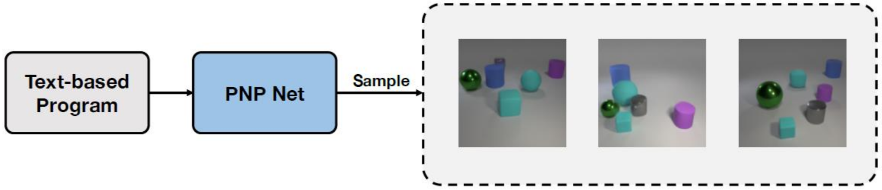
Graph NN

Adapting DNN to graph inputs…

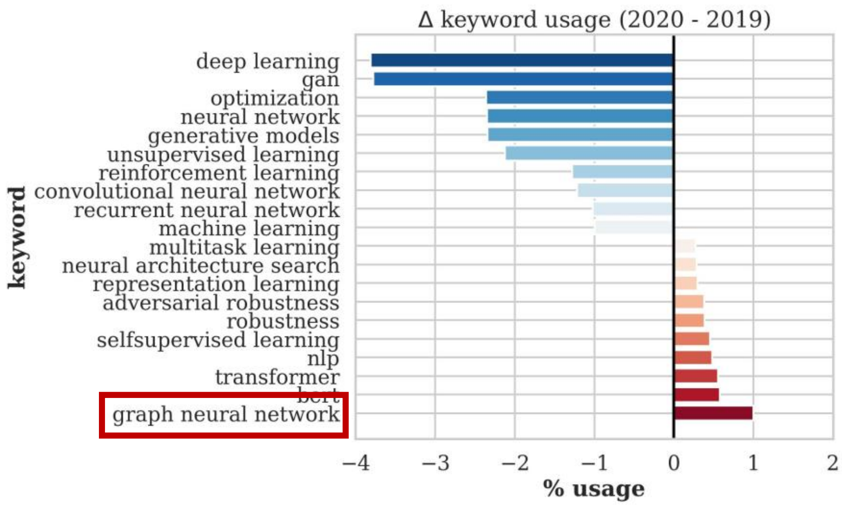




Graph classification, node classification, link prediction…



- **Computer vision**: scene generation, point cloud classification, action recognition…
- **NLP**: text classification (semantic graph)
- **Chemistry**: infer molecular properties, protein structure, *synthetize* new compound…

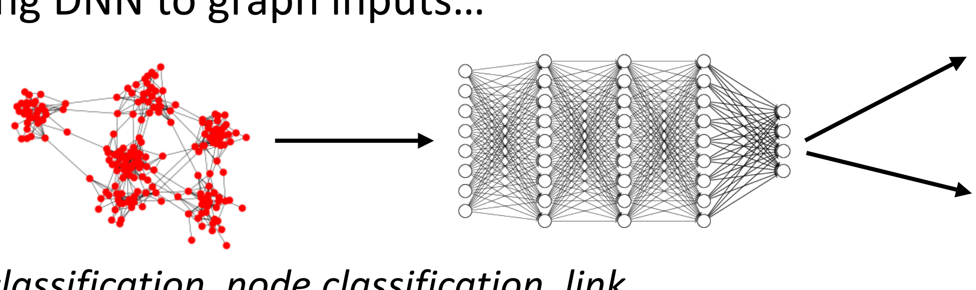


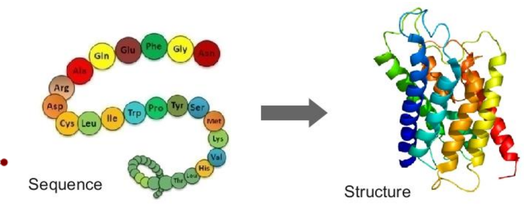
State-of-the-art: kinda (less clear)

Theory ? Most of it missing…

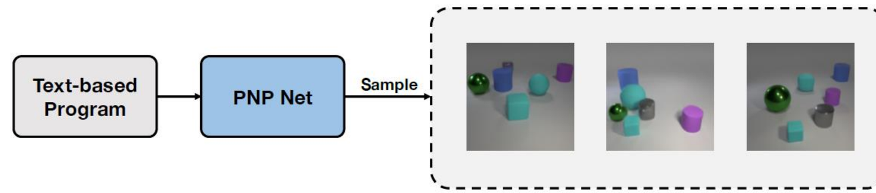
Graph NN

Adapting DNN to graph inputs…

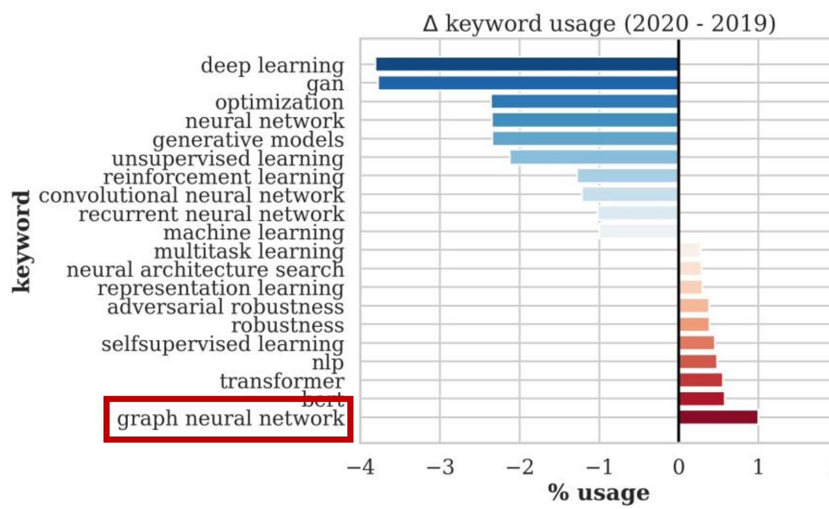




Graph classification, node classification, link prediction…



- **Computer vision**: scene generation, point cloud classification, action recognition…
- **NLP**: text classification (semantic graph)
- **Chemistry**: infer molecular properties, protein structure, *synthetize* new compound…



State-of-the-art: kinda (less clear)

Theory ? Most of it missing…

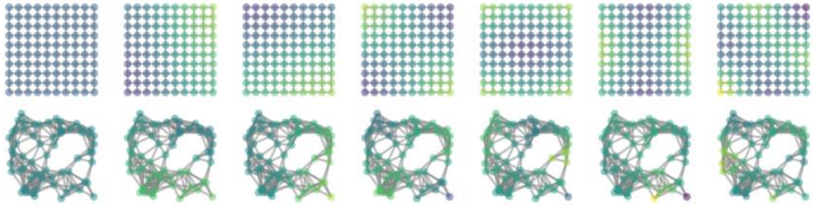
*In this talk: **Universal Approximation Theorem***

A zoology of Graph NN

- **Convolutional GNN** : takes inspiration from ***Convolutional*** *NN*

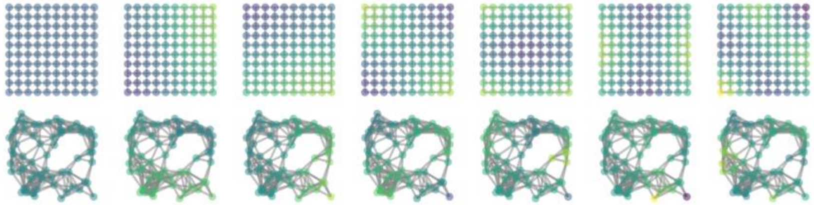
A zoology of Graph NN

- **Convolutional GNN** : takes inspiration from ***Convolutional*** *NN*
    - **Spectral** : convolution in Fourier domain (successive filtering using graph Laplacian)
    - **Spatial** : « convolution » in graph domain



A zoology of Graph NN

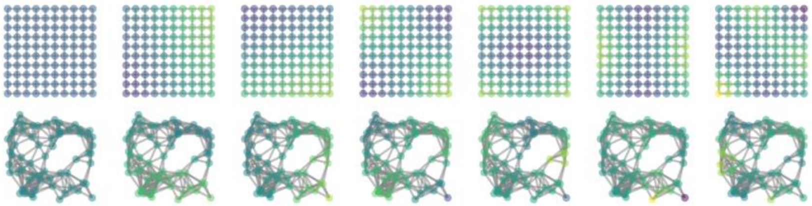
- **Convolutional GNN** : takes inspiration from ***Convolutional** NN*
  - **Spectral** : convolution in Fourier domain (successive filtering using graph Laplacian)
  - **Spatial** : « convolution » in graph domain



- **Recursive GNN** : defined as equilibrium state (inspired by physics, message-passing algorithms…)
- **Dynamic GNN** : temporal graphs, use random walks…
- **Graph Auto Encoders** : graph embedding, graph generation…

A zoology of Graph NN

- **Convolutional GNN** : takes inspiration from ***Convolutional*** *NN*
    - **Spectral** : convolution in Fourier domain (successive filtering using graph Laplacian)
    - **Spatial** : « convolution » in graph domain



- **Recursive GNN** : defined as equilibrium state (inspired by physics, message-passing algorithms…)
- **Dynamic GNN** : temporal graphs, use random walks…
- **Graph Auto Encoders** : graph embedding, graph generation…

Review papers :

- *Bronstein et al (2017) : Geometric Deep Learning: Going beyond Euclidean data*
- *Hamilton et al (2017) : Representation learning on graphs: Methods and applications*
- *Wu et al (2019) : A Comprehensive Survey on Graph Neural Networks*

A simple architecture: Invariant and Equivariant layers

Input

Weight matrix $W \in \mathbb{R}^{n \times n}$

Multi-graph $W \in \mathbb{R}^{n^\ell}$

A simple architecture: Invariant and Equivariant layers

Input

Weight matrix $W \in \mathbb{R}^{n \times n}$

Multi-graph $W \in \mathbb{R}^{n^\ell}$

A minima, a GNN is **invariant** or **equivariant** by permutation of nodes

A simple architecture: Invariant and Equivariant layers

Input

Weight matrix $W \in \mathbb{R}^{n \times n}$

Multi-graph $W \in \mathbb{R}^{n^\ell}$

Permutation of nodes

For $\sigma \in \Sigma_n$, denote $\sigma \star W$ perm. of nodes

Ex: $P_\sigma W P_\sigma^\top$ for $\ell = 2$

A minima, a GNN is **invariant** or **equivariant** by permutation of nodes

A simple architecture: Invariant and Equivariant layers

Input

Weight matrix $W \in \mathbb{R}^{n \times n}$

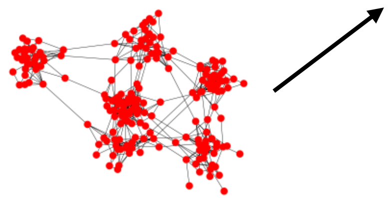
Multi-graph $W \in \mathbb{R}^{n^\ell}$

Permutation of nodes

For $\sigma \in \Sigma_n$, denote $\sigma \star W$ perm. of nodes

Ex: $P_\sigma W P_\sigma^\top$ for $\ell = 2$

A minima, a GNN is **invariant** or **equivariant** by permutation of nodes



$k = 6$

Invariant

$f(\sigma \star W) = f(W)$

A simple architecture: Invariant and Equivariant layers

Input

Weight matrix $W \in \mathbb{R}^{n \times n}$

Multi-graph $W \in \mathbb{R}^{n^\ell}$

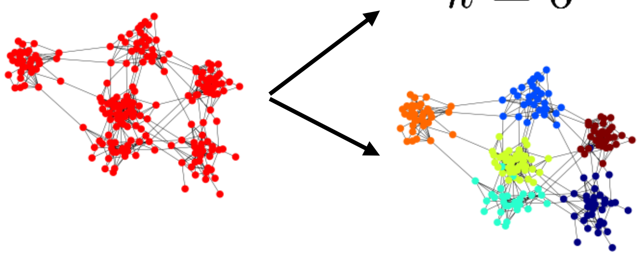
Permutation of nodes

For $\sigma \in \Sigma_n$, denote $\sigma \star W$ perm. of nodes

Ex: $P_\sigma W P_\sigma^\top$ for $\ell = 2$

A minima, a GNN is **invariant** or **equivariant** by permutation of nodes

$k = 6$



Invariant

$$f(\sigma \star W) = f(W)$$

Equivariant

$$f(\sigma \star W) = \sigma \star f(W)$$

A simple architecture: Invariant and Equivariant layers

Input

Weight matrix $W \in \mathbb{R}^{n \times n}$

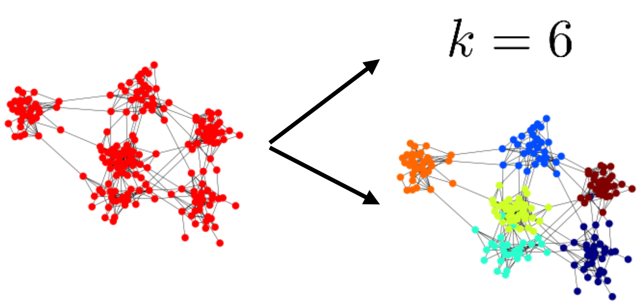
Multi-graph $W \in \mathbb{R}^{n^\ell}$

Permutation of nodes

For $\sigma \in \Sigma_n$, denote $\sigma \star W$ perm. of nodes

Ex: $P_\sigma W P_\sigma^\top$ for $\ell = 2$

A minima, a GNN is **invariant** or **equivariant** by permutation of nodes



Invariant

$$f(\sigma \star W) = f(W)$$

Equivariant

$$f(\sigma \star W) = \sigma \star f(W)$$

Idea : alternate **linear equivariant layers** with **non-linearities**, **invariant/equivariant last layer**

A simple architecture: Invariant and Equivariant layers

Input

Weight matrix $W \in \mathbb{R}^{n \times n}$

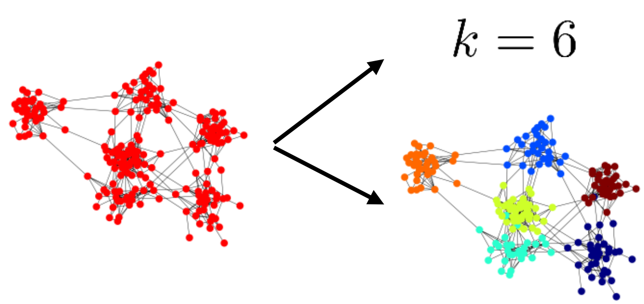
Multi-graph $W \in \mathbb{R}^{n^\ell}$

Permutation of nodes

For $\sigma \in \Sigma_n$, denote $\sigma \star W$ perm. of nodes

Ex: $P_\sigma W P_\sigma^\top$ for $\ell = 2$

A minima, a GNN is **invariant** or **equivariant** by permutation of nodes



Invariant

$$f(\sigma \star W) = f(W)$$

Equivariant

$$f(\sigma \star W) = \sigma \star f(W)$$

Idea : alternate **linear equivariant layers** with **non-linearities**, **invariant/equivariant last layer**

Thm *(Maron et al. 2018)*:

Bell number

There is a basis of $b(k+\ell)$ possible **equivariant** linear operators $F : \mathbb{R}^{n^k} \to \mathbb{R}^{n^\ell}$

(idem for invariant)

A simple architecture: Invariant and Equivariant layers

Input

Weight matrix $W \in \mathbb{R}^{n \times n}$

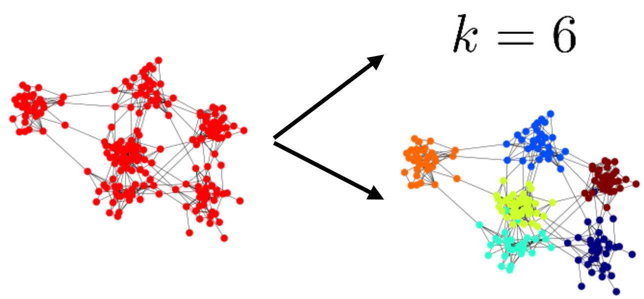
Multi-graph $W \in \mathbb{R}^{n^\ell}$

Permutation of nodes

For $\sigma \in \Sigma_n$, denote $\sigma \star W$ perm. of nodes

Ex: $P_\sigma W P_\sigma^\top$ for $\ell = 2$

A minima, a GNN is **invariant** or **equivariant** by permutation of nodes



Invariant

$$f(\sigma \star W) = f(W)$$

Equivariant

$$f(\sigma \star W) = \sigma \star f(W)$$

Idea : alternate **linear equivariant layers** with **non-linearities**, **invariant/equivariant last layer**

Thm *(Maron et al. 2018):* *Bell number*

There is a basis of $b(k+\ell)$ possible **equivariant** linear operators $F : \mathbb{R}^{n^k} \to \mathbb{R}^{n^\ell}$

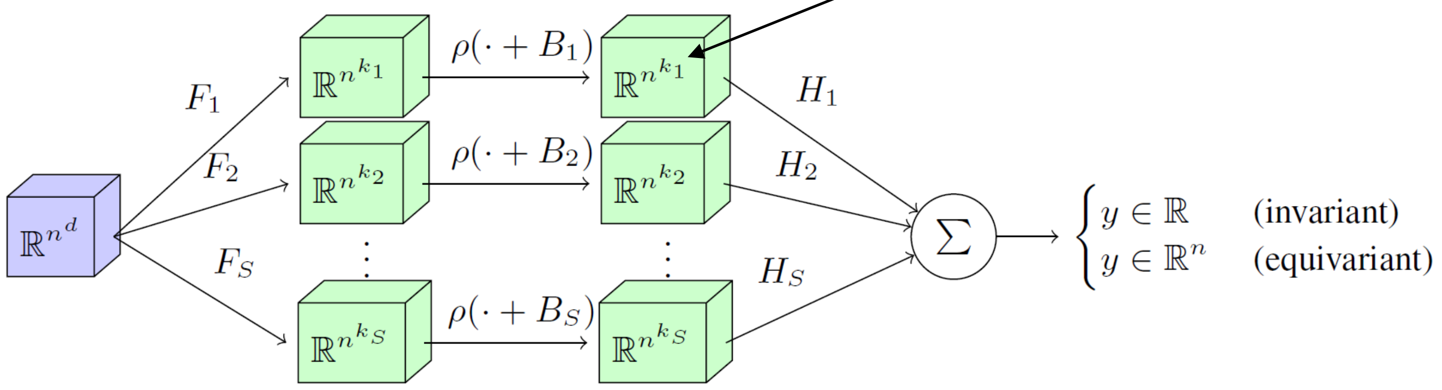
(idem for invariant)

Does not depend on n !! *Ex: there are only 15 equivariant linear operators* $\mathbb{R}^{n \times n} \to \mathbb{R}^{n \times n}$

Universality for Invariant NN

One-layer: $f(W) = \sum_s H_s \rho(F_s W + B_s)$

Can increase the order of the tensor !

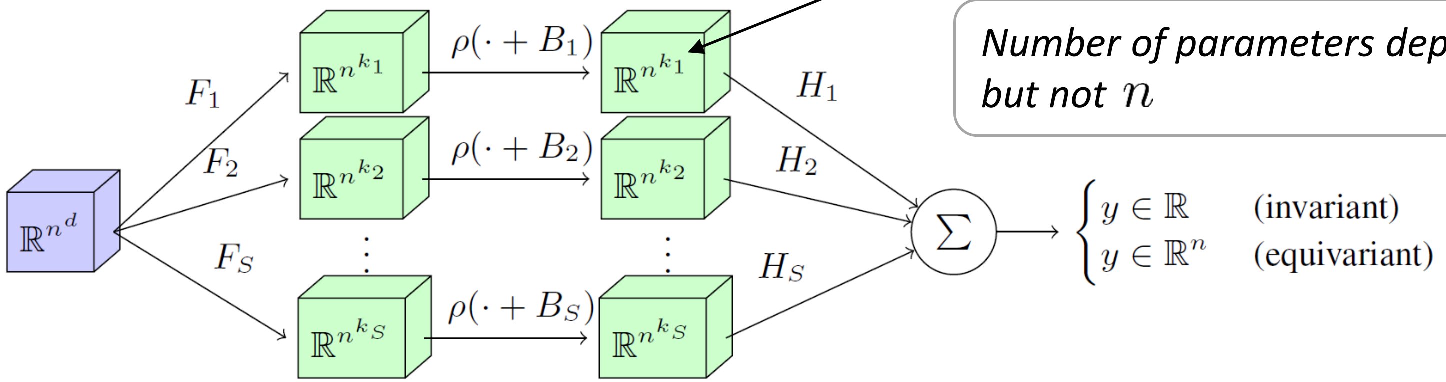


Universality for Invariant NN

One-layer: $f(W) = \sum_s H_s \rho(F_s W + B_s)$

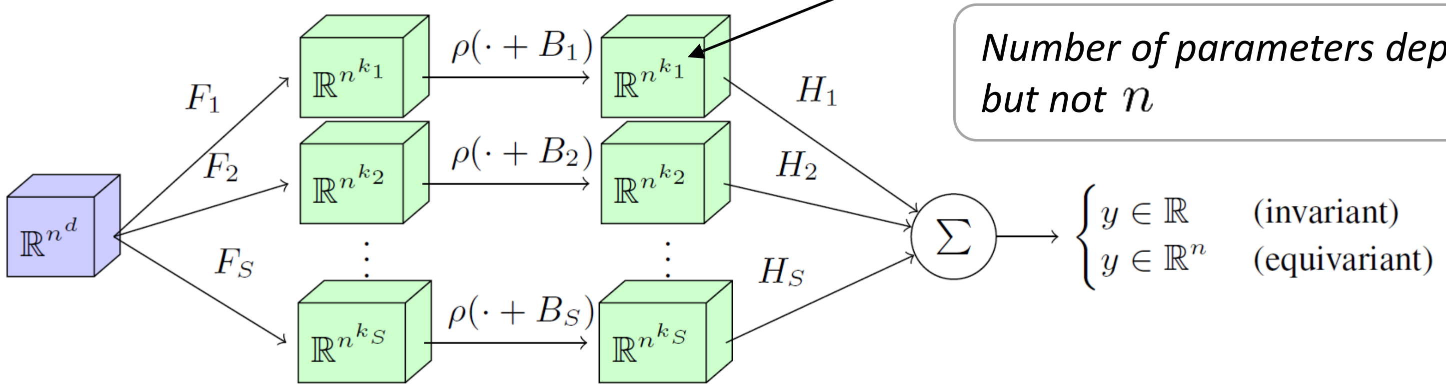
Can increase the order of the tensor !

Number of parameters depends on k_s *but not* n



Universality for Invariant NN

One-layer: $f(W) = \sum_s H_s \rho(F_s W + B_s)$



Thm *(Maron et al. 2019):*

When $S, k_s \to \infty$, can approximate **any continuous invariant function**.

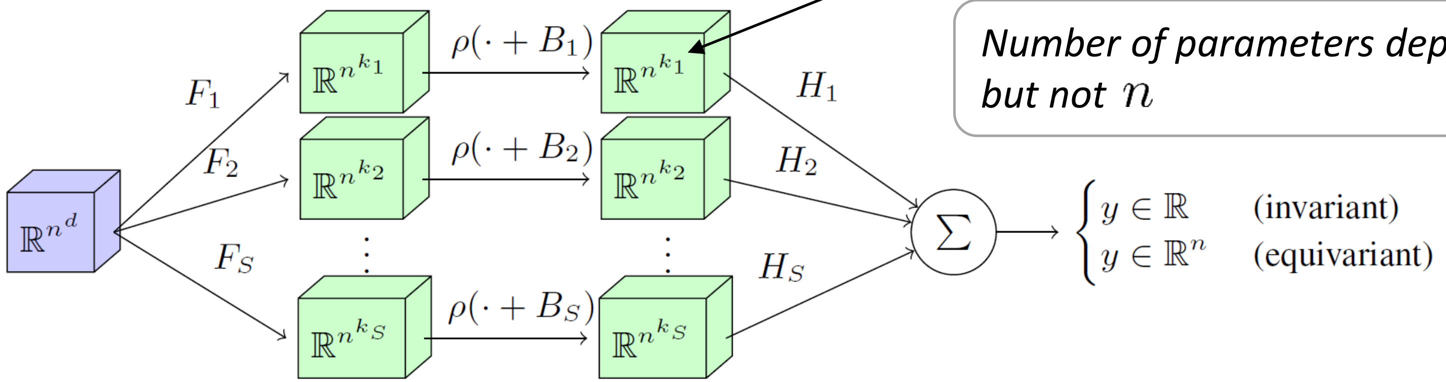
Proof: uses invariant polynomials, density of polynomials, and classical universality theorem.

Universality for Invariant NN

One-layer: $f(W) = \sum_s H_s \rho(F_s W + B_s)$

Can increase the order of the tensor !

Number of parameters depends on k_s *but not* n



Thm *(Maron et al. 2019):*

When $S, k_s \to \infty$, can approximate **any continuous invariant function**.

Proof: uses invariant polynomials, density of polynomials, and classical universality theorem.

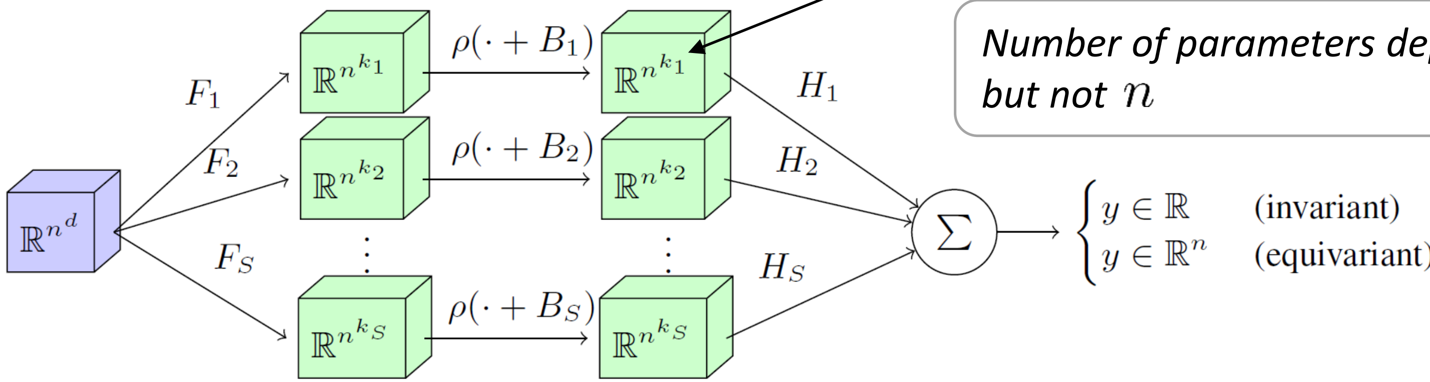
First contribution / warm-up :

Universality for Invariant NN

One-layer: $f(W) = \sum_s H_s \rho(F_s W + B_s)$

Can increase the order of the tensor !

Number of parameters depends on k_s *but not* n



Thm *(Maron et al. 2019):*

When $S, k_s \to \infty$, can approximate **any continuous invariant function**.

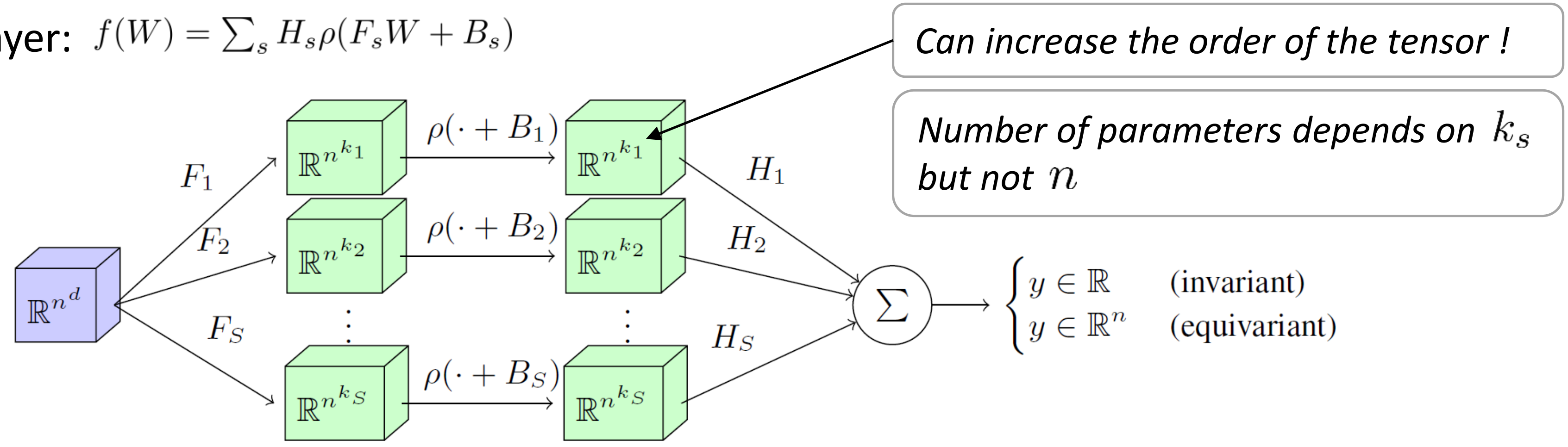
Proof: uses invariant polynomials, density of polynomials, and classical universality theorem.

First contribution / warm-up :

- Alternative proof based on Stone-Weierstrass (SW) theorem.

Universality for Invariant NN

One-layer: $f(W) = \sum_s H_s \rho(F_s W + B_s)$



Thm *(Maron et al. 2019):*

When $S, k_s \to \infty$, can approximate **any continuous invariant function**.

Proof: uses invariant polynomials, density of polynomials, and classical universality theorem.

First contribution / warm-up :

- Alternative proof based on Stone-Weierstrass (SW) theorem.

- With a ***single set of parameters***, can approximate a function defined on **graphs of varying size** $n \leq n_{\max}$ *(continuous for the edit distance)*

Our proof : SW theorem for graphs

Apply Stone-Weierstrass theorem in the space of graphs quotiented by permutations.

Thm *(Stone-Weierstrass)*

On a compact space, an **algebra** of continuous real-valued functions that **separates points** is dense in the space of continuous functions.

Our proof : SW theorem for graphs

Apply Stone-Weierstrass theorem in the space of graphs quotiented by permutations.

Thm *(Stone-Weierstrass)*

On a compact space, an **algebra** of continuous real-valued functions that **separates points** is dense in the space of continuous functions.

*Must authorize **product** of NNs…*

Our proof : SW theorem for graphs

Apply Stone-Weierstrass theorem in the space of graphs quotiented by permutations.

Thm *(Stone-Weierstrass)*

On a compact space, an **algebra** of continuous real-valued functions that **separates points** is dense in the space of continuous functions.

Must authorize ***product*** *of NNs…*

Use the « cos trick » *[Hornik 1989]*

- Do the proof with $\rho = \cos$
- A **product** of cos is also a **sum**
- Approximate cos with any ρ

Our proof : SW theorem for graphs

Apply Stone-Weierstrass theorem in the space of graphs quotiented by permutations.

Thm *(Stone-Weierstrass)*

On a compact space, an **algebra** of continuous real-valued functions that **separates points** is dense in the space of continuous functions.

*Must authorize **product** of NNs…*

Use the « cos trick » *[Hornik 1989]*

- Do the proof with $\rho = \cos$
- A **product** of cos is also a **sum**
- Approximate cos with any ρ

Easy

Our proof : SW theorem for graphs

Apply Stone-Weierstrass theorem in the space of graphs quotiented by permutations.

Thm *(Stone-Weierstrass)*

On a compact space, an **algebra** of continuous real-valued functions that **separates points** is dense in the space of continuous functions.

*Must authorize **product** of NNs…*

Use the « cos trick » *[Hornik 1989]*

- Do the proof with $\rho = \cos$
- A **product** of cos is also a **sum**
- Approximate cos with any ρ

Easy

« For any two distinct points, there is a function that distinguishes them »

Our proof : SW theorem for graphs

Apply Stone-Weierstrass theorem in the space of graphs quotiented by permutations.

Thm *(Stone-Weierstrass)*

On a compact space, an **algebra** of continuous real-valued functions that **separates points** is dense in the space of continuous functions.

Must authorize ***product*** *of NNs…*

Use the « cos trick » *[Hornik 1989]*

- Do the proof with $\rho = \cos$
- A **product** of cos is also a **sum**
- Approximate cos with any ρ

Easy

« For any two distinct points, there is a function that distinguishes them »

« For any two non-isomorphic graphs, there is a NN that distinguishes them »

Our proof : SW theorem for graphs

Apply Stone-Weierstrass theorem in the space of graphs quotiented by permutations.

Thm *(Stone-Weierstrass)*

On a compact space, an **algebra** of continuous real-valued functions that **separates points** is dense in the space of continuous functions.

Must authorize ***product*** *of NNs…*

Use the « cos trick » *[Hornik 1989]*

- Do the proof with $\rho = \cos$
- A **product** of cos is also a **sum**
- Approximate cos with any ρ

Easy

« For any two distinct points, there is a function that distinguishes them »

« For any two non-isomorphic graphs, there is a NN that distinguishes them »

« Any two graphs that yield the same result for all NNs are isomorphic »

Our proof : SW theorem for graphs

Apply Stone-Weierstrass theorem in the space of graphs quotiented by permutations.

Thm *(Stone-Weierstrass)*

On a compact space, an **algebra** of continuous real-valued functions that **separates points** is dense in the space of continuous functions.

*Must authorize **product** of NNs…*

Use the « cos trick » *[Hornik 1989]*

- Do the proof with $\rho = \cos$
- A **product** of cos is also a **sum**
- Approximate cos with any ρ

Easy

« For any two distinct points, there is a function that distinguishes them »

« For any two non-isomorphic graphs, there is a NN that distinguishes them »

« Any two graphs that yield the same result for all NNs are isomorphic »

Hard ! *(core of the proof)*

Universality for Equivariant NN

Main contribution:

Thm *(Keriven, Peyré 2019):*

One layer **equivariant** GNNs $\mathbb{R}^{n^d} \to \mathbb{R}^n$ are dense in the space of **continuous equivariant functions**.

(defined on graphs of varying size $n \leq n_{\max}$ *)*

Universality for Equivariant NN

Main contribution:

Thm *(Keriven, Peyré 2019):*

One layer **equivariant** GNNs $\mathbb{R}^{n^d} \to \mathbb{R}^n$ are dense in the space of **continuous equivariant functions.**

(defined on graphs of varying size $n \leq n_{\max}$ *)*

Proof : « equivariant » polynomial ? ❌ Stone-Weierstrass theorem ? ❌ 🤔

Universality for Equivariant NN

Main contribution:

Thm *(Keriven, Peyré 2019):*

One layer **equivariant** GNNs $\mathbb{R}^{n^d} \to \mathbb{R}^n$ are dense in the space of **continuous equivariant functions.**

(defined on graphs of varying size $n \leq n_{\max}$ *)*

Proof : « equivariant » polynomial ? ✗ Stone-Weierstrass theorem ? ✗

Thm (SW for equivariant functions)
(Keriven, Peyré 2019)

An **algebra of equivariant functions** (for coordinate-wise product) that **separates points** and **separates coordinates** is dense in the space of equivariant continuous functions.

Universality for Equivariant NN

Main contribution:

Thm *(Keriven, Peyré 2019):*

One layer **equivariant** GNNs $\mathbb{R}^{n^d} \to \mathbb{R}^n$ are dense in the space of **continuous equivariant functions.**

(defined on graphs of varying size $n \leq n_{\max}$ *)*

Proof : « equivariant » polynomial ? ✗ Stone-Weierstrass theorem ? ✗

Thm (SW for equivariant functions)
(Keriven, Peyré 2019)

An **algebra of equivariant functions** (for coordinate-wise product) that **separates points** and **separates coordinates** is dense in the space of equivariant continuous functions.

« Any two coordinates not related by an automorphism can be separated by a function »

Universality for Equivariant NN

Main contribution:

Thm *(Keriven, Peyré 2019):*

One layer **equivariant** GNNs $\mathbb{R}^{n^d} \to \mathbb{R}^n$ are dense in the space of **continuous equivariant functions.**

(defined on graphs of varying size $n \leq n_{\max}$ *)*

Proof : « equivariant » polynomial ? ✗ Stone-Weierstrass theorem ? ✗

Thm (SW for equivariant functions)
(Keriven, Peyré 2019)

An **algebra of equivariant functions** (for coordinate-wise product) that **separates points** and **separates coordinates** is dense in the space of equivariant continuous functions.

- **Proof** : non-trivial modif. from *Brosowski et al. « An elementary proof of Stone-Weierstrass theorem » (1981)*

« Any two coordinates not related by an automorphism can be separated by a function »

Universality for Equivariant NN

Main contribution:

Thm *(Keriven, Peyré 2019):*

One layer **equivariant** GNNs $\mathbb{R}^{n^d} \to \mathbb{R}^n$ are dense in the space of **continuous equivariant functions.**

(defined on graphs of varying size $n \leq n_{\max}$)

Proof : « equivariant » polynomial ? ✗ Stone-Weierstrass theorem ? ✗

Thm (SW for equivariant functions)
(Keriven, Peyré 2019)

An **algebra of equivariant functions** (for coordinate-wise product) that **separates points** and **separates coordinates** is dense in the space of equivariant continuous functions.

- **Proof** : non-trivial modif. from *Brosowski et al. « An elementary proof of Stone-Weierstrass theorem » (1981)*
- Not valid for **output** $\mathbb{R}^{n^k}$
- Not valid for subgroups of Σ_n

« Any two coordinates not related by an automorphism can be separated by a function »

Conclusion

Summary

- Defined a class of Universal Invariant and Equivariant GNNs
- Proved an Equivariant Stone-Weierstrass Theorem (of independent interest)

Conclusion

Summary

- Defined a class of Universal Invariant and Equivariant GNNs
- Proved an Equivariant Stone-Weierstrass Theorem (of independent interest)

Outlooks

- More general SW theorem to handle graph output and subgroups of permutations

Conclusion

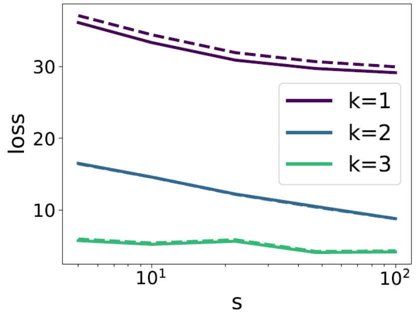
Summary

- Defined a class of Universal Invariant and Equivariant GNNs
- Proved an Equivariant Stone-Weierstrass Theorem (of independent interest)

Outlooks

- More general SW theorem to handle graph output and subgroups of permutations

- Deal with computational complexity due to high-order tensor / characterize the power of low-order GNNs *(see Maron et al 2019)*



Conclusion

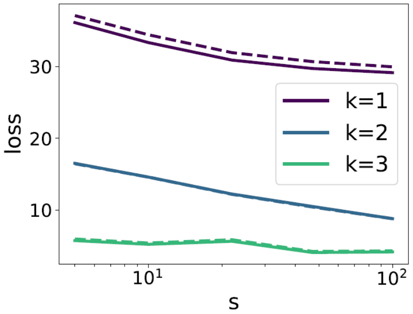
Summary

- Defined a class of Universal Invariant and Equivariant GNNs
- Proved an Equivariant Stone-Weierstrass Theorem (of independent interest)

Outlooks

- More general SW theorem to handle graph output and subgroups of permutations

- Deal with computational complexity due to high-order tensor / characterize the power of low-order GNNs *(see Maron et al 2019)*



- Study Universality / stability wrt weaker metrics (Gromov-Wasserstein…)

Conclusion

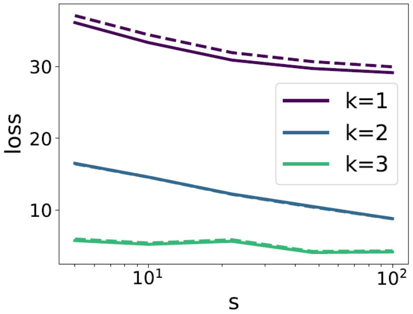
Summary

- Defined a class of Universal Invariant and Equivariant GNNs
- Proved an Equivariant Stone-Weierstrass Theorem (of independent interest)

Outlooks

- More general SW theorem to handle graph output and subgroups of permutations

- Deal with computational complexity due to high-order tensor / characterize the power of low-order GNNs *(see Maron et al 2019)*



- Study Universality / stability wrt weaker metrics (Gromov-Wasserstein…)

Take-home msg: GNNs are still in their infancy, both theoretically and in practice. Scalability and stability remain challenging. Many opportunities !

Thank you !

Keriven, Peyré. **Universal Invariant and Equivariant Graph Neural Networks**
NeurIPS 2019, arxiv:1905.04943

More at *nkeriven.github.io*