On Convolution of Graph Signals *And* Deep Learning on Graph Domains

Candidate: Jean-Charles Vialatte
Advisors: Vincent Gripon, Mathias Herberts
Supervisor: Gilles Coppin

Jury: Pierre Borgnat*, Matthias Löwe*,
Paulo Goncalves, Juliette Mattioli
*: examiners

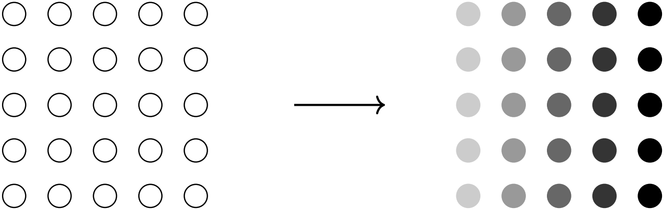
December 13th 2018

Outline

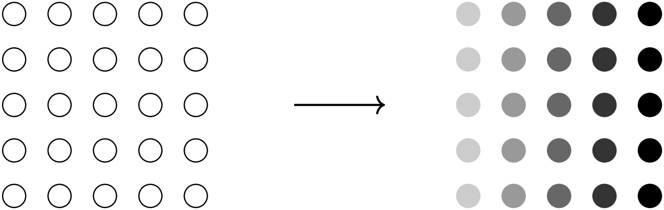
Outline

Definitions: Signals and Graphs

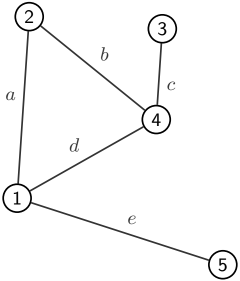
Signal:



Signal:



Graph:

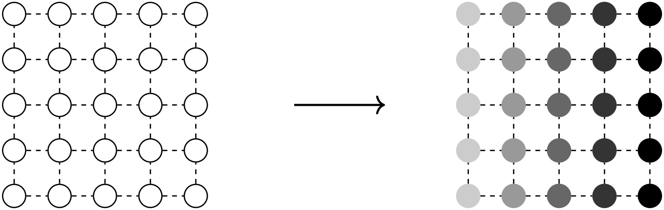


$$\begin{pmatrix} 0 & a & 0 & d & e \\ a & 0 & 0 & b & 0 \\ 0 & 0 & 0 & c & 0 \\ d & b & c & 0 & 0 \\ e & 0 & 0 & 0 & 0 \end{pmatrix}$$

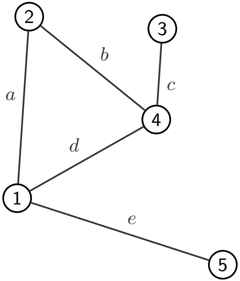
A: adjacency matrix

Definitions: Signals and Graphs

Signal:



Graph:

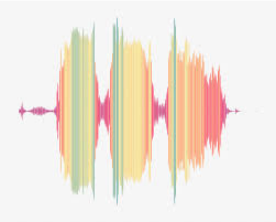


$$\begin{pmatrix} 0 & a & 0 & d & e \\ a & 0 & 0 & b & 0 \\ 0 & 0 & 0 & c & 0 \\ d & b & c & 0 & 0 \\ e & 0 & 0 & 0 & 0 \end{pmatrix}$$

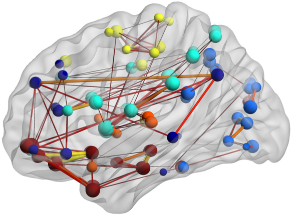
A: adjacency matrix

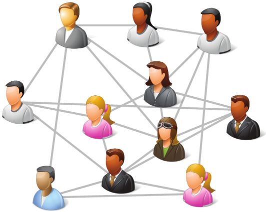
Examples of signals

Euclidean domains



Non-Euclidean domains

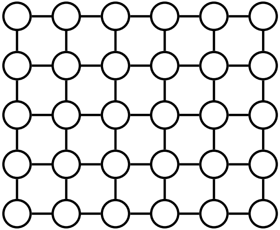




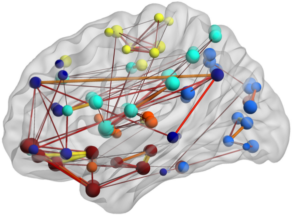
Examples of signals

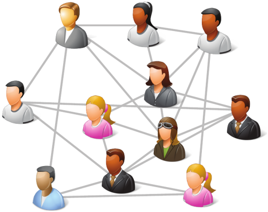
Most domains can be represented with a graph:

Euclidean domains

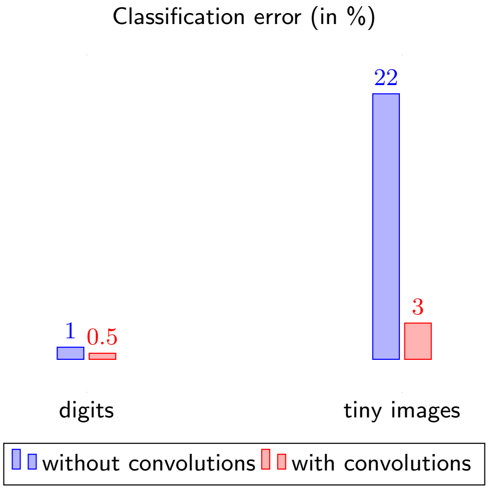


Non-Euclidean domains



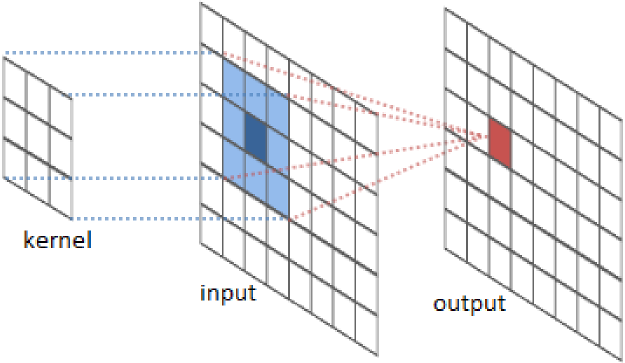


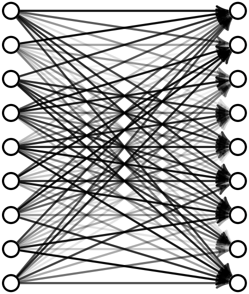
On Euclidean domains:



A key factor: convolution

Defined on Euclidean domains:

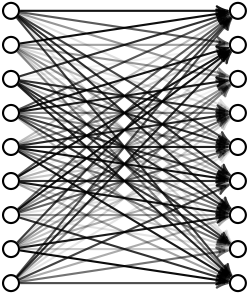


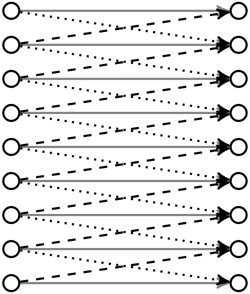


Dense layer

fully connected
no tied weights
ex: 81 different weights here

Connectivity pattern: MLP vs CNN





Dense layer

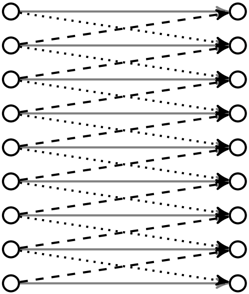
fully connected
no tied weights
ex: 81 different weights here

Convolutional layer

locally connected
with weight sharing
ex: 3 different weights here

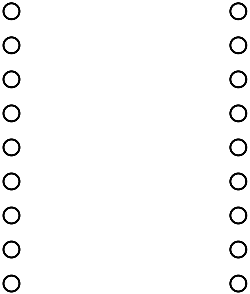
Problem: How to extend convolutions?

Euclidean structure



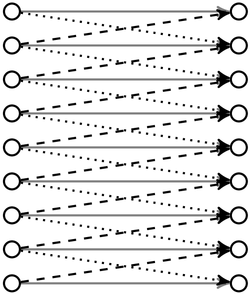
convolution

Non-Euclidean structure



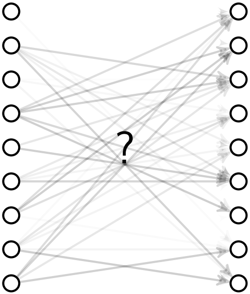
Problem: How to extend convolutions?

Euclidean structure



convolution

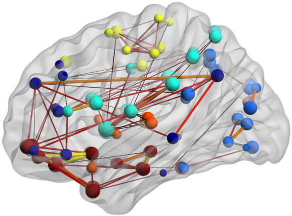
Non-Euclidean structure



?

Let X be a dataset. We classify its rows. Two different kind of graph structures:

Supervised classification of graph-structured data

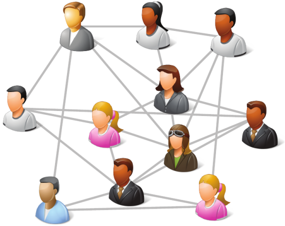


n

X

$(b \times n)$

Semi-supervised classification of nodes



n

X

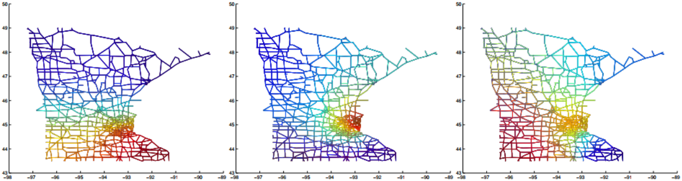
$(n \times p)$

Outline

Spectral approaches

Convolutions are defined in the graph spectral domain.

$$L = D - A = U \Lambda U^T \qquad \mathrm{GFT}(X) = UX$$



(a) λ_3 (b) λ_4 (c) λ_5

Figure: Example of signals of the Laplacian eigenbasis

Using the GFT, convolution amount to a pointwise multiplication in the spectral domain.

$$X \otimes \Theta = U^T (UX . U\Theta)$$

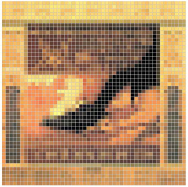
$$X \otimes \Theta = U^T(UX.U\Theta)$$

Pros

- Elegant and fast under some approximations
- Can be used off the shelf: no need to specify any weight sharing

Cons

- Introduce isotropic symmetries
- Do not match Euclidean convolutions on grid graphs



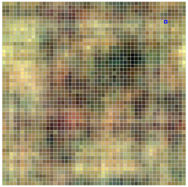
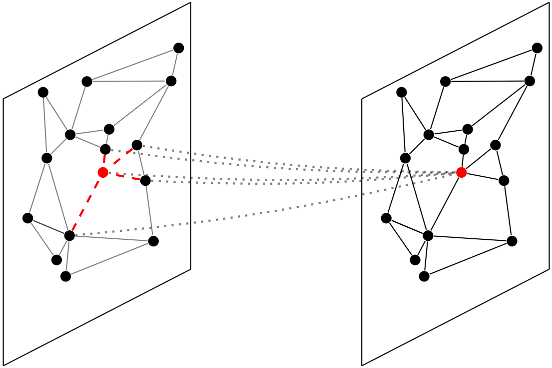


Figure: Example of translation defined as convolution with a dirac (courtesy of Pasdeloup, B.)

Convolutions are defined as a sum over a neighborhood, usually a sum of dot products (cf references in thesis manuscript).

$$(X \otimes \Theta)(v_i) = \sum_{j \in \mathcal{N}_{v_i}} \theta_{ij} X(v_j)$$



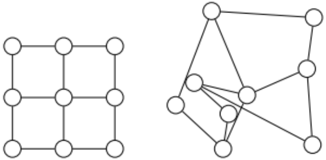
$$(X \otimes \Theta)(v_i) = \sum_{j \in \mathcal{N}_{v_i}} \theta_{ij} X(v_j)$$

Pros

- Match Euclidean convolutions on grid graphs
- Locally connected

Cons

- Weight sharing is not always explicit



A few important references

- Bruna et al., 2013: spectral filters with $\mathcal{O}(1)$ weights, K smoother matrix used to interpolate more weights.
$$g_\theta(X) = U^T(UX.K\theta)$$
- Defferard et al., 2016: filters based on Chebychev polynomials $(T_i)_i$.
$$g_\theta(L) = \sum_{i=0}^{k} \theta_i \, T_i(\widetilde{L})$$
- Kipf et al., 2016: application to semi-supervised settings.
$$Y = \widetilde{A} X \Theta$$
- Velickovic et al., 2017: introduction of attention coefficients $(A_k)_k$.
$$Y = \Big\Vert_{k=1}^{K} A_k X \Theta_k$$
- Du et al., 2017: convolution from the GSP field (Sandryhaila et al., 2013).
$$Y = \sum_{k=1}^{K} \widetilde{A}^k X \Theta_k$$

Outline

Recall: Euclidean convolution

Definition

Convolution on $\mathcal{S}(\mathbb{Z}^2)$
The (discrete) convolution $s_1 * s_2$ is a binary operation in $\mathcal{S}(\mathbb{Z}^2)$ defined as:

$$\forall (a,b) \in \mathbb{Z}^2, (s_1 * s_2)[a,b] = \sum_i \sum_j s_1[i,j]\, s_2[a-i, b-j]$$

A convolution operator f is a function parameterized by a signal $w \in \mathcal{S}(\mathbb{Z}^2)$ *s.t.* :

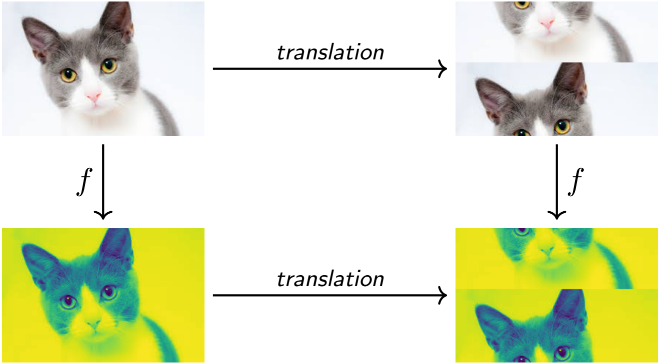
- $f = . * w$ (right operator)
- $f = w * .$ (left operator)

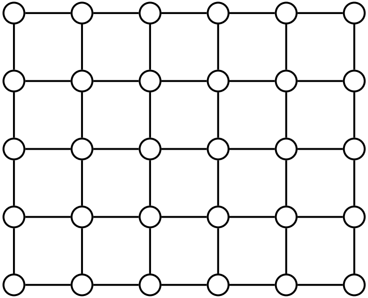
Some notable properties:

- Linearity
- Locality and weight sharing
- Commutativity (optional)
- Equivariance to translations (*i.e.* commutes with them)

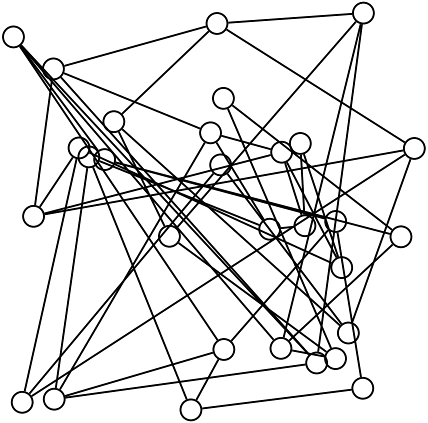
Theorem

Characterization of convolution operators on $\mathcal{S}(\mathbb{Z}^2)$
A linear transformation f is equivariant to translations $\Leftrightarrow$ it is a convolution operator.





Can use the Euclidean convolution



How to extend the convolution here ?

A few notions of representation theory

A group is a set (defined by some properties) which can act on other sets.

Let Γ be a group, $g \in \Gamma$, and V be a set. Example of group actions:

- $L_g : \Gamma \to \Gamma$ (auto-action)
- $g(.) : V \to V$ (action on V)

$$\begin{tikzcd}
 & h \mapsto gh \\
g \arrow[ur, "L_g"] \arrow[dr, "g(.)"'] & \\
 & v \mapsto g(v)
\end{tikzcd}$$

Definition

Group convolution
Let a group Γ, the group convolution between two signals s_1 and $s_2 \in \mathcal{S}(\Gamma)$ is defined as:

$$\forall h \in \Gamma, (s_1 *_{\mathrm{I}} s_2)[h] = \sum_{g \in \Gamma} s_1[g]\, s_2[g^{-1}h]$$

provided at least one of the signals has finite support if Γ is not finite.

Theorem

Characterization of group convolution operators
Let a group Γ, let $f \in \mathcal{L}(\mathcal{S}(\Gamma))$,

1. *f is a group convolution right operator $\Leftrightarrow$ f is equivariant to left multiplications,*
2. *f is a group convolution left operator $\Leftrightarrow$ f is equivariant to right multiplications,*
3. *f is a group convolution commutative operator $\Leftrightarrow$ f is equivariant to multiplications.*

Let a graph $G = \langle V, E \rangle$ *s.t.* $E \subset V^2$.
Starting point: bijective map φ between a group Γ and G (or a subgraph).

$$\begin{tikzcd}
\Gamma \arrow[r, "\varphi"] \arrow[d, "\text{lin. ext.}"'] & V \arrow[d, "\text{lin. ext.}"] \\
\mathcal{S}(\Gamma) \arrow[r, "\widetilde{\varphi}"'] & \mathcal{S}(V)
\end{tikzcd}$$

(Goal: convolution on the vertex set)
φ: bijective map

$$\begin{tikzcd}
\mathcal{S}(\Gamma) \arrow[d, "\widetilde{f}"'] & \mathcal{S}(V) \arrow[l, "\varphi^{-1}"'] \arrow[d, "f=\varphi\circ\widetilde{f}\circ\varphi^{-1}"] \\
\mathcal{S}(\Gamma) \arrow[r, "\varphi"'] & \mathcal{S}(V)
\end{tikzcd}$$

Equivariance theorem to operators of the form $\varphi \circ L_g \circ \varphi^{-1}$ holds.
But not necessarily to actions of Γ on V (*i.e.* of the form $g(.)$).

(Goal: equivariance theorem holds)
Condition: φ is a bijective equivariant map
Denote $g_v = \varphi^{-1}(v)$.

$$\begin{tikzcd} g_u \arrow[r, "L_{g_v}"] \arrow[d, "\varphi"'] & g_v g_u \arrow[d, "\varphi"] \\ u \arrow[r, dashed, "g_v(.)"'] & \varphi(g_v g_u) \end{tikzcd}$$

We need $g_v(.) = \varphi \circ L_{g_v} \circ \varphi^{-1}$
i.e. $\forall u \in V, g_v(u) = \varphi(g_v g_u)$.

φ: bijective equivariant map *i.e.* $g_v(u) = \varphi(g_v g_u)$.

Definition

φ-convolution
$\forall s_1, s_2 \in \mathcal{S}(V)$:

$$s_1 *_\varphi s_2 = \sum_{v \in V} s_1[v]\, g_v(s_2) \quad (1)$$

$$= \sum_{g \in \Gamma} s_1[\varphi(g)]\, g(s_2) \quad (2)$$

Theorem

Characterization of φ-convolution right operators

f is a φ-convolution right operator $\Leftrightarrow$ f is equivariant to Γ

Let Γ be an abelian group. No need to exhibit φ in this case:

Definition

Mixed domain convolution
$\forall r \in \mathcal{S}(\Gamma)$ and $\forall s \in \mathcal{S}(V)$:

$$r *_{\mathrm{M}} s = \sum_{g \in \Gamma} r[g]\, g(s) \in \mathcal{S}(V)$$

Equivariance theorem holds:

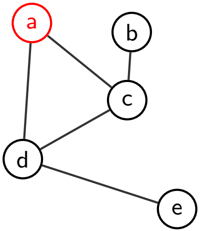
Corollary

f is a M-convolution left operator $\Leftrightarrow$ f is equivariant to Γ

(Converse sense still requires bijectivity between Γ and V).

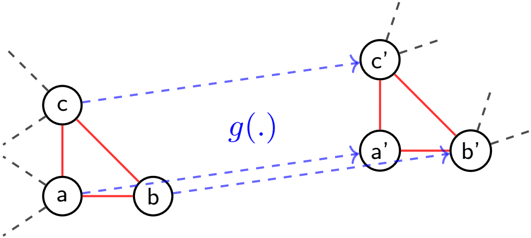
Inclusion and role of the edge set

- **Edge constrained (EC)**



$$g(a) \in \{c, d\}$$
$$g(a) \notin \{b, e\}$$

- **Locality Preserving (LP)**



(Goal: description of EC and LP convolutions)

Definition

Cayley graph and subgraph

Let a group Γ and one of its generating set $\mathcal{U}$. The *Cayley graph* generated by $\mathcal{U}$, is the digraph $\vec{G} = \langle V, E\rangle$ such that $V = \Gamma$ and E is such that, either:

- $\forall a, b \in \Gamma, a \to b \Leftrightarrow \exists g \in \mathcal{U}, ga = b$ (*left Cayley graph*)
- $\forall a, b \in \Gamma, a \to b \Leftrightarrow \exists g \in \mathcal{U}, ag = b$ (*right Cayley graph*)
- both points above (*abelian Cayley graph*)

A *Cayley subgraph* is a subgraph that is isomorph to a Cayley graph.

$$\text{a} \xrightarrow{g} \text{b}$$

Figure: An edge of a Cayley graph

Theorem

Characterization by Cayley subgraphs

Let a graph $G = \langle V, E \rangle$, then:

1. *its left Cayley subgraphs characterize its EC φ-convolutions,*
2. *its right Cayley subgraphs characterize its LP φ-convolutions,*
3. *its abelian Cayley subgraphs characterize its EC and LP M-convolutions.*

Corollary

Properties of convolutions that are both EC and LP

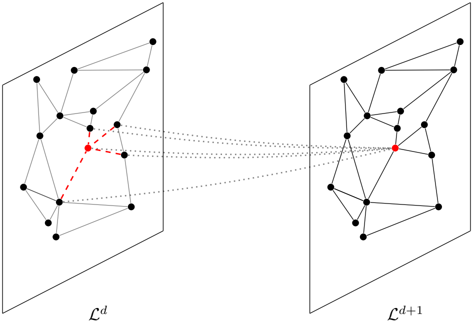
1. *If a φ-convolution of group Γ is EC and LP then Γ is abelian;*
2. *an M-convolution is EC if, and only if, it is also LP.*

Other results in the manuscript

- Description with smaller kernels
- The weight sharing is preserved
- More detailed results depending on laterality of operator and equivariance
- Analysis of limitations due to algebraic structure of the Cayley subgraphs
- Above theorems hold for groupoids of partial transformation under mild conditions
- They also hold for groupoids based on paths under restrictive conditions

Outline

Propagational representation of a layer

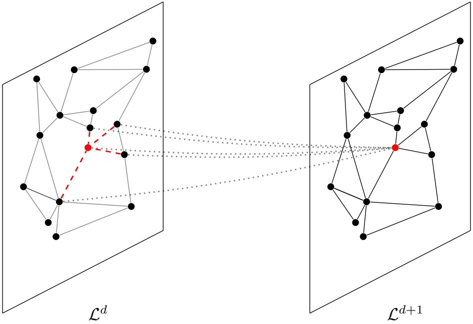


Definition

Edge-constrained layer

Connections (dotted lines) are constrained by edges (red lines) in a local receptive field.

Propagational representation of a layer



Definition

Edge-constrained layer
Connections (dotted lines) are constrained by edges (red lines) in a local receptive field.

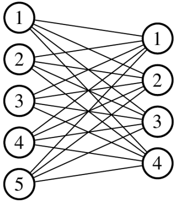
Theorem

Characterization by local receptive fields (LRF)
There is a graph for which a layer is EC $\Leftrightarrow$ its LRF are intertwined.

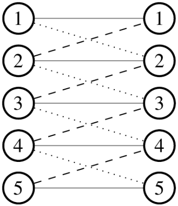
Connectivity matrix W

(Goal: generalized layer representation)

$$\mathbf{y} = h(W \cdot \mathbf{x} + b)$$



$$\begin{pmatrix} w_{11} & w_{12} & w_{13} & w_{14} \\ w_{21} & w_{22} & w_{23} & w_{24} \\ w_{31} & w_{32} & w_{33} & w_{34} \\ w_{41} & w_{42} & w_{43} & w_{44} \\ w_{51} & w_{52} & w_{53} & w_{14} \end{pmatrix}$$

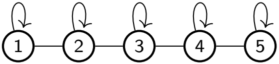


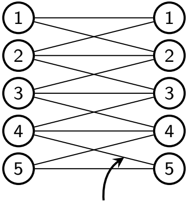
$$\begin{pmatrix} w_2 & w_3 & 0 & 0 & 0 \\ w_1 & w_2 & w_3 & 0 & 0 \\ 0 & w_1 & w_2 & w_3 & 0 \\ 0 & 0 & w_1 & w_2 & w_3 \\ 0 & 0 & 0 & w_1 & w_2 \end{pmatrix}$$

Scheme tensor S

(Goal: generalized layer representation)

$$W = \Theta \cdot S$$
$$\mathbf{y} = h(\Theta \cdot S \cdot \mathbf{x} + b)$$





$$\begin{pmatrix} \mathbf{s}_{11} & \mathbf{s}_{12} & 0 & 0 & 0 \\ \mathbf{s}_{21} & \mathbf{s}_{22} & \mathbf{s}_{23} & 0 & 0 \\ 0 & \mathbf{s}_{32} & \mathbf{s}_{33} & \mathbf{s}_{34} & 0 \\ 0 & 0 & \mathbf{s}_{43} & \mathbf{s}_{44} & \mathbf{s}_{45} \\ 0 & 0 & 0 & \mathbf{s}_{54} & \mathbf{s}_{55} \end{pmatrix}$$

$$W_{45} = \sum_{k=1}^{\omega} \Theta_k S_{45k}$$

$$\widehat{\Theta S X}[j,q,b] = \sum_{k=1}^{\omega} \sum_{p=1}^{P} \sum_{i=1}^{n} \Theta[k,p,q]\, S[k,i,j]\, X[i,p,b]$$

$$g(X) = \widehat{\Theta S X} \text{ where } \begin{cases} W_{pq}{}^{ij} = \Theta_{pq}{}^{k} S_{k}{}^{ij} \\ g(X)_{jq}{}^{b} = W_{jq}{}^{ip} X_{ip}{}^{b} \end{cases}$$

index	size	description
i	n	input neuron
j	m	output neuron
p	N	input channel
q	M	feature map
k	ω	kernel weight
b	B	batch instance

Table: indices

tensor	shape
Θ	$\omega \times N \times M$
S	$\omega \times n \times m$
X	$n \times N \times B$
ΘS	$n \times m \times N \times M$
SX	$\omega \times m \times N \times B$
ΘX	$\omega \times n \times M \times B$
$\widehat{\Theta S X}$	$m \times M \times B$

Table: shapes

Properties

This formulation is:

- Linear
- Associative
- Commutative
- Generic (next slide)

It is explainable as a convolution of graph signals when:

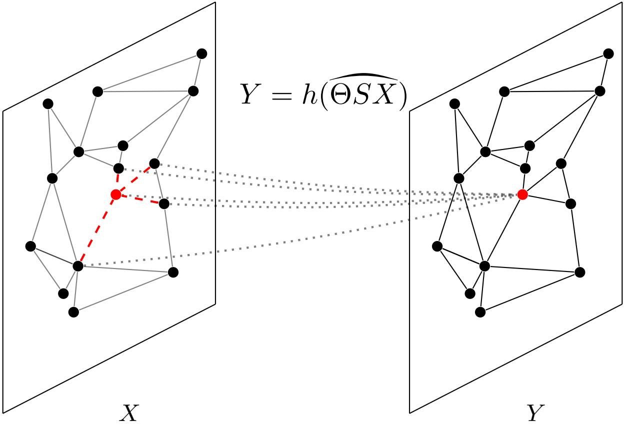
- in supervised application
- Either 0 or 1 weight per connection ($\mathbf{s_{ij}}$ are one-hot vectors)

In other cases it can be seen as a linear combination of convolutions.

Genericity of ternary representation

Given adequate specification of the weight sharing scheme S, we can obtain, *e.g.* :

- a dense layer
- a partially connected layer
- a convolutional layer
- a graph convolutional layer (GCN, Kipf et al.)
- a graph attention layer (GAT, Velickovic et al.)
- a topology-adaptive graph convolution layer (TAGCN, Du et al.)
- a mixture model convolutional layer (MOnet, Monti et al.)
- a generalized convolution under sparse priors
- any partial connectivity pattern, sparse or not



The propagation logic is in the scheme S. For example, it can be either:

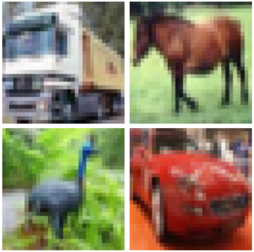
- given
- randomized
- learned
- inferred

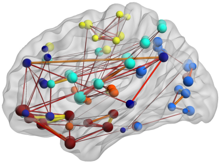
Outline

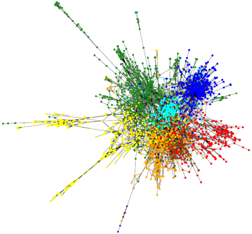
Experiments:

- Study of influence of symmetries using S
- Learning S when masked by an adjacency matrix and its powers
- Monte Carlo simulations with random realizations of S
- Learning S in semi-supervised applications
- Inferring S from translations

Datasets

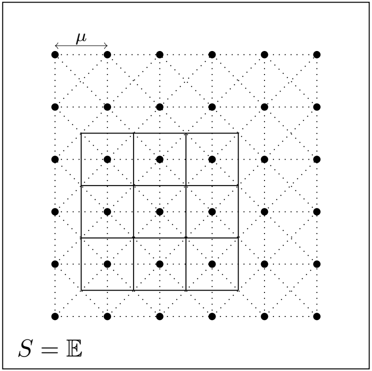


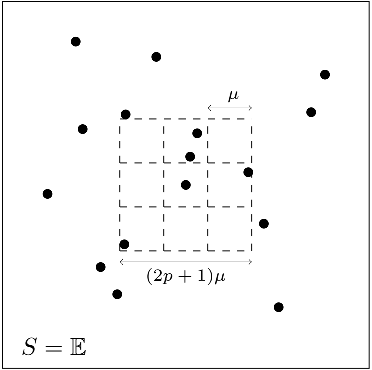




Influence of symmetries

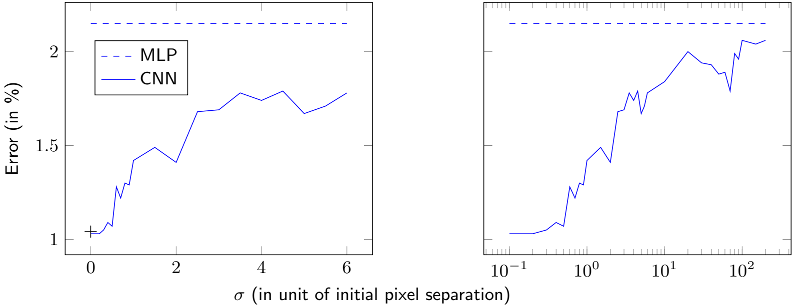
$$Y = h(\widehat{\Theta S X})$$





Influence of symmetries: results on MNIST

$$Y = h(\widehat{\Theta S X})$$



Learning both S and Θ on MNIST and scrambled MNIST

$$Y = h(\widehat{\Theta S X})$$

Ordering	Conv5x5	A^1	A^2	A^3	A^4	A^5	A^6
no prior	/	1.24%	1.02%	0.93%	0.90%	0.93%	1.00%
prior	0.87%	1.21%	0.91%	0.91%	0.87%	0.80%	0.74%

Table: Grid graphs on MNIST

$$Y = h(\widehat{\Theta S X})$$

Ordering	Conv5x5	A^1	A^2	A^3	A^4	A^5	A^6
no prior	/	1.24%	1.02%	0.93%	0.90%	0.93%	1.00%
prior	0.87%	1.21%	0.91%	0.91%	0.87%	0.80%	0.74%

Table: Grid graphs on MNIST

MLP	Conv5x5	Thresholded ($p = 3\%$)	k-NN ($k = 25$)
1.44%	1.39%	1.06%	0.96%

Table: Covariance graphs on Scrambled MNIST

$$Y = h(\widehat{\Theta S X})$$

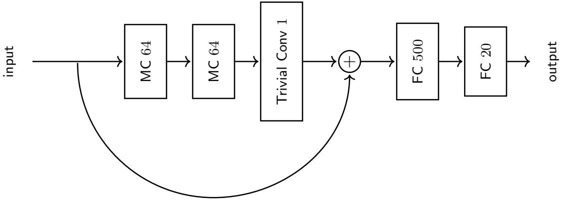


Figure: Diagram of the MCNet architecture used

MNB	FC2500	FC2500-500	ChebNet32	FC500	MCNet
68.51	64.64	65.76	68.26	71.46 (72.25)	70.74 (**72.62**)

Table: Accuracies (in %) on 20NEWS, given as *mean (max)*

$$Y = h(\widehat{\Theta S X})$$

Comparison of

- Graph Convolution Network (GCN),
- Graph Attention Network (GAT),
- Topology Adaptive GCN (TAGCN).

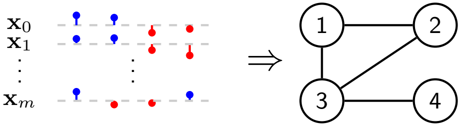
With our models:

- Addition of graph dropout to GCN (GCN*),
- Graph Contraction Network (GCT).

Dataset	MLP	GCN	GAT	TAGCN	GCN*	GCT
Cora	58.8 ± 0.9	81.8 ± 0.9	83.3 ± 0.6	82.9 ± 0.7	$\mathbf{83.4} \pm 0.7$	83.3 ± 0.7
Citeseer	56.7 ± 1.1	72.2 ± 0.6	72.1 ± 0.6	71.7 ± 0.7	72.5 ± 0.8	$\mathbf{72.7} \pm 0.5$
Pubmed	72.6 ± 0.9	79.0 ± 0.5	78.3 ± 0.7	78.9 ± 0.5	78.2 ± 0.7	$\mathbf{79.2} \pm 0.4$

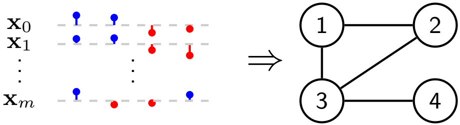
Table: Mean accuracy (in %) and standard deviation after 100 run

Step 0 : infer a graph

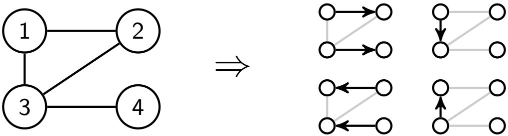


Another approach: finding translations in graphs to construct S

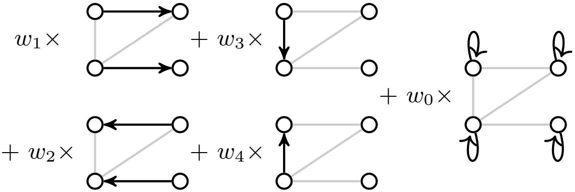
Step 0 : infer a graph



Step 1: infer translations

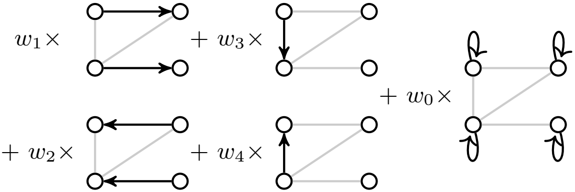


Step 2: design convolution weight-sharing

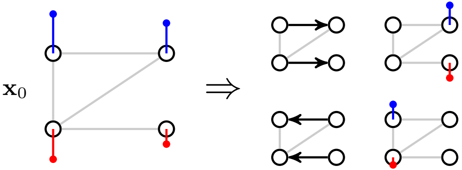


Another approach: finding translations in graphs to construct S

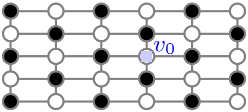
Step 2: design convolution weight-sharing

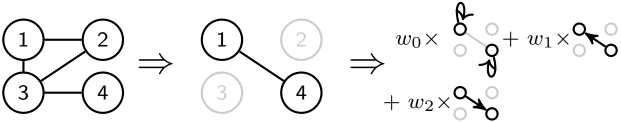


Step 3: design data-augmentation



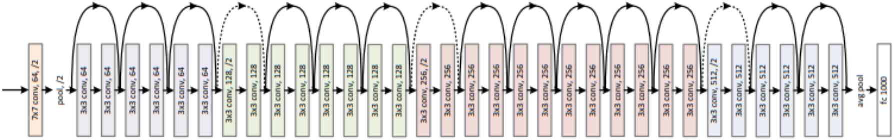
Step 4: design graph subsampling and convolution weight-sharing





Architecture

We used a variant of deep residual networks (ResNet).
We swap operations (data augmentation, convolutions, subsampling) with their counterparts.



Results on CIFAR-10, scrambled CIFAR-10 and PINES fMRI

$$Y = h(\widehat{\Theta S X})$$

Support	MLP	CNN	Grid Graph		Covariance Graph
			ChebNet[c]	Proposed	Proposed
Full Data Augmentation	78.62%[a,b]	**93.80%**	85.13%	93.94%	92.57%
Data Augmentation w/o Flips	——	92.73%	84.41%	92.94%	91.29%
Graph Data Augmentation	——	92.10%[d]	——	92.81%	**91.07%**[a]
None	69.62%	87.78%	——	88.83%	85.88%[a]

[a] No priors about the structure
[b] Lin et al., 2015
[c] Defferard et al., 2016
[d] Data augmentation done with covariance graph

Table: CIFAR-10 and scrambled CIFAR-10

Results on CIFAR-10, scrambled CIFAR-10 and PINES fMRI

$$Y = h(\widehat{\Theta S X})$$

Support	MLP	CNN	Grid Graph		Covariance Graph
			ChebNet[c]	Proposed	Proposed
Full Data Augmentation	78.62%[a,b]	**93.80%**	85.13%	93.94%	92.57%
Data Augmentation w/o Flips	——	92.73%	84.41%	92.94%	91.29%
Graph Data Augmentation	——	92.10%[d]	——	92.81%	**91.07%**[a]
None	69.62%	87.78%	——	88.83%	85.88%[a]

[a] No priors about the structure
[b] Lin et al., 2015
[c] Defferard et al., 2016
[d] Data augmentation done with covariance graph

Table: CIFAR-10 and scrambled CIFAR-10

Support	None		Neighborhood Graph	
Method	MLP	CNN (1x1 kernels)	ChebNet[c]	Proposed
Accuracy	82.62%	84.30%	82.80%	**85.08%**

Table: PINES fMRI

Outline

Summary

We studied convolutions of graph signals and used them to build and understand extensions of CNN on graph domains.

Convolution of graph signals:

- Algebraic description of convolution of graph signals φ- and M-convolutions,
- Constructed as the class of linear operator that are equivariant to actions of a group.
- Strong characterization results for graphs with Cayley subgraphs.
- Extension with groupoids.

Deep learning on graphs:

- Novel representation based on weight sharing: the neural contraction
- Monte-Carlo Neural Networks (MCNN)
- Graph Contraction Networks (GCT)
- Graph dropout (GCN*)
- Translation-Convolutional Neural Network (TCNN)

Final words

Perspectives:

- In the literature of this domain: semi-supervised $>>$ supervised.
- Both tasks can be abstracted to a more general case.

$$Y = h(\widehat{\Theta S X})$$

- There can be more than one tensor rank which relations can be represented by a graph.

$$Y = h(g(X, A_1, A_2, ..., A_r))$$

- Extended range of applications for deep learning architecture.
- Thinking in AI might be about creating connections (captured by S) and not about updating weights.

Thank you for your attention !

Contributions

- **Generalizing the convolution operator to extend CNNs to irregular domains**, Jean-Charles Vialatte, Vincent Grippon, Grégoire Mercier, *arXiv preprint 2016*.
- **Neighborhood-preserving translations on graphs**, Nicolas Grelier, Bastien Pasdeloup, Jean-Charles Vialatte, Vincent Gripon, *IEEE GlobalSIP 2016*.
- **Learning local receptive fields and their weight sharing scheme on graphs**, Jean-Charles Vialatte, Vincent Gripon, Gilles Coppin, *IEE GlobalSIP 2017*.
- **A study of deep learning robustness against computation failures**, Jean-Charles Vialatte, François Leduc-Primeau, *ACTA 2017*.
- **Convolutional neural networks on irregular domains through approximate translations on inferred graphs**, Bastien Pasdeloup, Vincent Gripon, Jean-Charles Vialatte, Dominique Pastor, *arXiv preprint 2017*.
- **Translations on graphs with neighborhood preservation**, Bastien Pasdeloup, Vincent Gripon, Nicolas Grelier, Jean-Charles Vialatte, Dominique Pastor, *arXiv preprint 2017*.
- **Matching CNNs without Priors about data**, Carlos-Eduardo Rosar Kos Lassance, Jean-Charles Vialatte, Vincent Gripon, *IEEE DSW 2018*.
- **On convolution of graph signals and deep learning on graph domains**, Jean-Charles Vialatte, thesis, *unpublished*.
- **Convolution of graph signals**, Jean-Charles Vialatte, Vincent Gripon, Gilles Coppin, *unpublished*.
- **Graph contraction networks, Graph dropout, Monte-Carlo Networks**, Jean-Charles Vialatte, Vincent Gripon, Gilles Coppin, *unpublished*.