

Cours 10 bis: CNN et GNN

Emmanuel Franck^{*},

19 septembre, 2024

Master CMSI, M2, Strasbourg

^{*}MACARON project-team, Université de Strasbourg, CNRS, Inria, IRMA, France

The Inria logo is written in a red, cursive script.The IRMA logo consists of the letters 'IRMA' in a blue, bold, sans-serif font. Below the letters is a horizontal blue line, and underneath that line, the text 'Institut de Recherche Mathématique Avancée' is written in a smaller, blue, sans-serif font.

Réseaux convolutifs

Réseaux sur graphes

- Traitement du signal sur graphe

- GNN spectraux

- GNN spatiaux

Réseaux convolutifs

Réseaux sur graphes

Traitement du signal sur graphe

GNN spectraux

GNN spatiaux

Réseaux convolutifs

Image et réseau de neurones

- Les réseaux de neurones ont connu un immense succès en traitement de l'image.
- Exemple d'utilisation:
 - ▶ **Classification:** déterminer la catégorie d'une image. On cherche une fonction $f : \mathcal{M}^{n,n}([0, 1]) \rightarrow [0, 1]$
 - ▶ **Segmentation:** déterminer les contours d'un objet dans une image. On cherche une fonction $f : \mathcal{M}^{n,n}([0, 1]) \rightarrow \mathcal{M}^{n,n}([0, 1])$

Difficulté

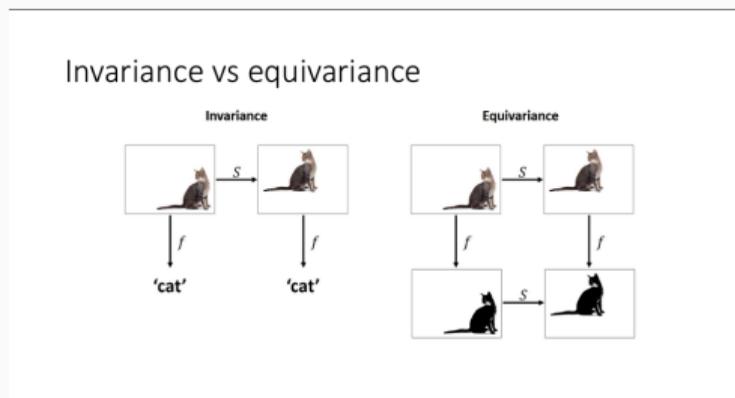
Il s'agit de problème en très grande dimension (ici n^2 le nombre de pixel).

Solution

En pratique il s'agit d'un problème structuré sur une grille 2D dont la dimension réelle est plus faible (les pixels d'une image sont corrélés). Afin de faciliter l'apprentissage on peut introduire des **a priori** sur la fonction qu'on cherche à construire **dans le modèle**. **Théorie moderne des CNN**.

A priori géométrique

Les problèmes de classification et segmentation sont **invariants/équivariants** par rapport a des transformations géométriques comme **la translation/rotation**.



CNN

Les CNN classiques imposent l'invariance par **translation**. Plus récemment l'invariance par rotation a aussi été imposée.

Réseaux CNN et a priori II

- Pour un signal sur une grille 1D, la translation d'un pixel est donné par la matrice

$$S = \begin{pmatrix} 0 & 1 & \cdots & \cdots & 0 \\ \vdots & 0 & 1 & \cdots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \vdots & \vdots & & \ddots & 1 \\ 1 & 0 & \cdots & \cdots & 0 \end{pmatrix}$$

et celle de k pixel par S^k .

Equivariance par rotation et convolution

Soit la couche $L(\mathbf{x}) = \sigma(A\mathbf{x} + \mathbf{b})$ avec une matrice circulante de la forme

$$A = \begin{pmatrix} c_0 & c_1 & c_2 & \cdots & c_1 \\ c_{n-1} & c_0 & c_1 & \cdots & c_{n-2} \\ c_{n-2} & c_{n-1} & c_0 & \cdots & c_{n-3} \\ \cdots & & & & \cdots \\ \cdots & & & & \cdots \\ c_1 & c_2 & c_3 & \cdots & c_0 \end{pmatrix}$$

avec $\mathbf{b} = 0$. Cette couche est équivariante par translation et donc satisfait $S^k L(\mathbf{x}) = L(S^k \mathbf{x})$.

Réseaux CNN et a priori III

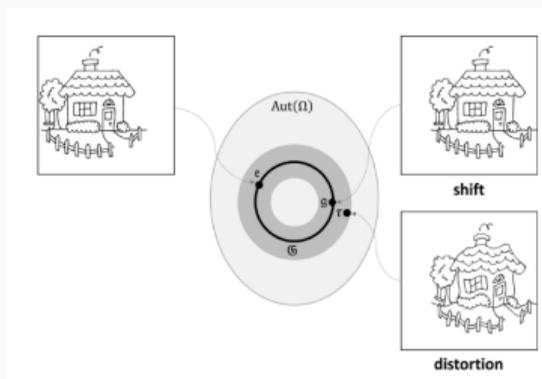
Equivariance par rotation et convolution

Les matrices circulantes correspondent à des **convolutions discret**.

A priori de stabilité

Une petite locale de l'entrée doit induire une petite perturbation du résultat. Si on appelle x l'image et x_ϵ l'image modifié par une perturbation de taille ϵ on demande donc:

$$\|f(x_\epsilon) - f(x)\| \leq \epsilon \|x\|$$



- Pour obtenir la stabilité on **localise les filtres de convolutions spatialement**.

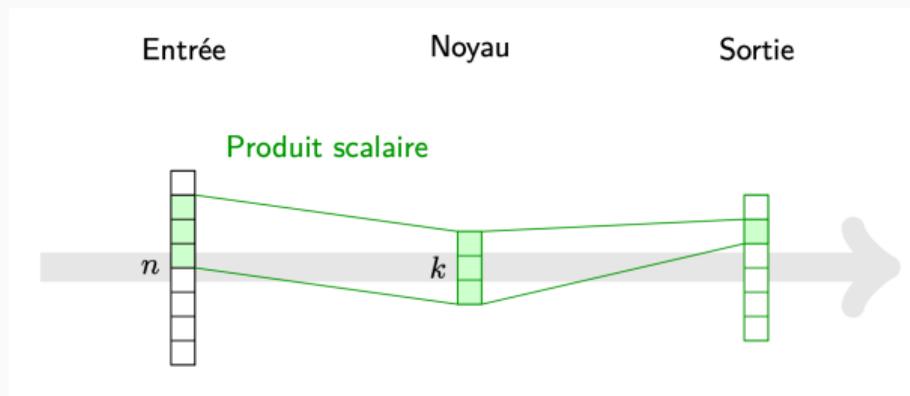
Réseaux CNN et a priori IV

Couche convolutive

Soit une entrée composée de plusieurs canaux (signaux) $x = (\mathbf{x}_1, \dots, \mathbf{x}_m)$ avec \mathbf{x}_i en dimension $d(1, 2 \text{ ou } 3)$ avec n données par dimension. Une **couche convolutive applique k filtres de convolution** (donc multiplie chaque signal par k matrices circulantes) et on obtient en sorti k signaux. Cela revient d'entrée de taille $m \times n^d$ et une sortie de taille $k \times n^d$:

$$g_i^{con} = \sigma \left(\sum_{j=1}^m K_{ij} f_j \right)$$

pour $0 \leq i \leq k$.



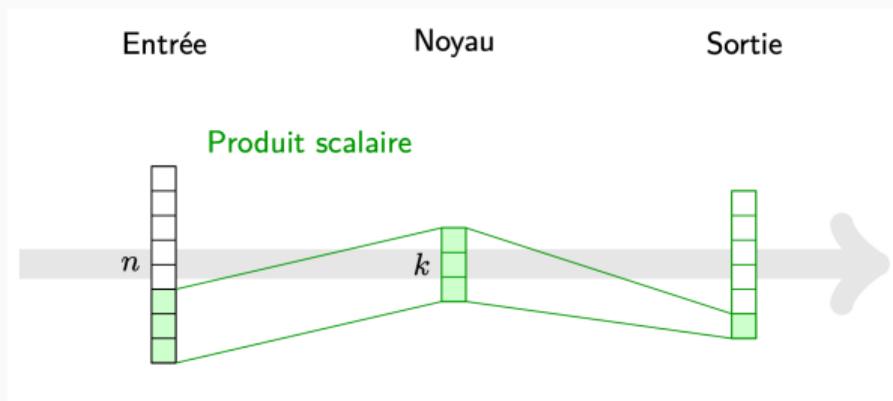
Réseaux CNN et a priori IV

Couche convolutive

Soit une entrée composée de plusieurs canaux (signaux) $x = (\mathbf{x}_1, \dots, \mathbf{x}_m)$ avec \mathbf{x}_i en dimension $d(1, 2 \text{ ou } 3)$ avec n données par dimension. Une **couche convolutive applique k filtres de convolution** (donc multiplie chaque signal par k matrices circulantes) et on obtient en sorti k signaux. Cela revient d'entrée de taille $m \times n^d$ et une sortie de taille $k \times n^d$:

$$g_i^{con} = \sigma \left(\sum_{j=1}^m K_{ij} f_j \right)$$

pour $0 \leq i \leq k$.



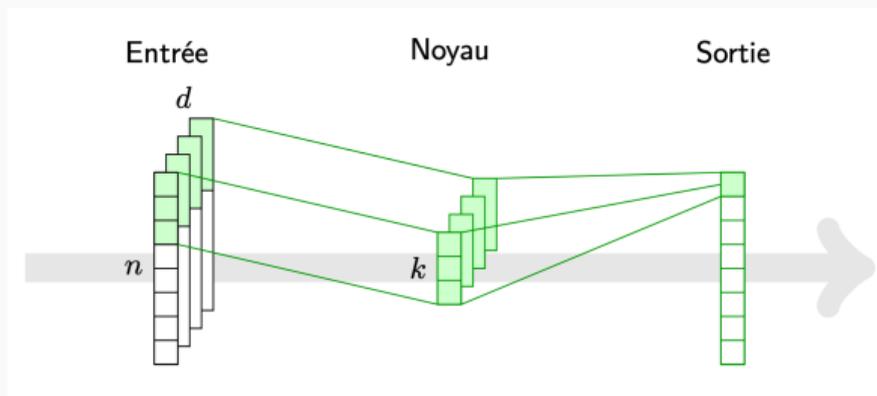
Réseaux CNN et a priori IV

Couche convolutive

Soit une entrée composée de plusieurs canaux (signaux) $x = (\mathbf{x}_1, \dots, \mathbf{x}_m)$ avec \mathbf{x}_i en dimension $d(1, 2 \text{ ou } 3)$ avec n données par dimension. Une **couche convolutive applique k filtres de convolution** (donc multiplie chaque signal par k matrices circulantes) et on obtient en sorti k signaux. Cela revient d'entrée de taille $m \times n^d$ et une sortie de taille $k \times n^d$:

$$g_i^{con} = \sigma \left(\sum_{j=1}^m K_{ij} f_j \right)$$

pour $0 \leq i \leq k$.



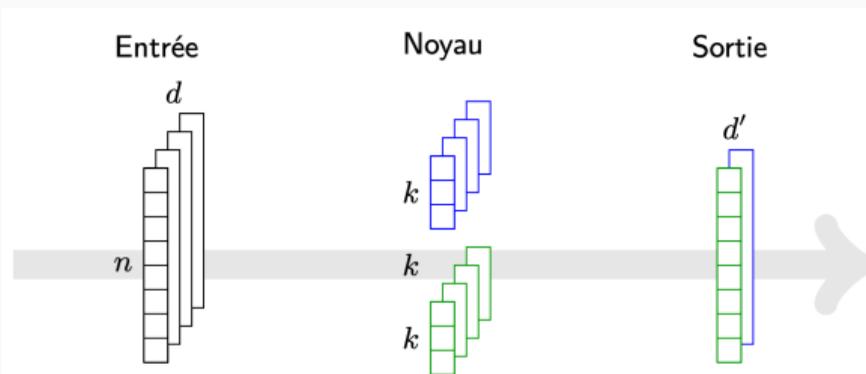
Réseaux CNN et a priori IV

Couche convolutive

Soit une entrée composée de plusieurs canaux (signaux) $x = (\mathbf{x}_1, \dots, \mathbf{x}_m)$ avec \mathbf{x}_i en dimension $d(1, 2 \text{ ou } 3)$ avec n données par dimension. Une **couche convolutive applique k filtres de convolution** (donc multiplie chaque signal par k matrices circulantes) et on obtient en sorti k signaux. Cela revient d'entrée de taille $m \times n^d$ et une sortie de taille $k \times n^d$:

$$g_i^{con} = \sigma \left(\sum_{j=1}^m K_{ij} f_j \right)$$

pour $0 \leq i \leq k$.



CNN et approches multi-échelles

A priori d'échelle

La séparation des échelles peut permettre de limiter l'effet de la grande dimension. Si les échelles locales ou globales dominent. On encode donc une **structure d'analyse multi-échelles dans les CNN**.



x



$$f^*(x) = P(y = \{ 'beach', 'mountain' \} | x)$$

- L'architecture des CNN et les filtres locaux permettent cette analyse multi-échelle.

CNN et approches multi-échelles

A priori d'échelle

La séparation des échelles peut permettre de limiter l'effet de la grande dimension. Si les échelles locales ou globales dominent. On encode donc une **structure d'analyse multi-échelles dans les CNN**.



$$f^*(x) = P(y = \{ 'beach', 'mountain' \} | x)$$

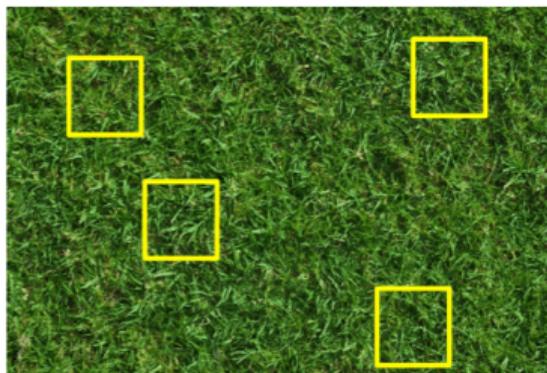
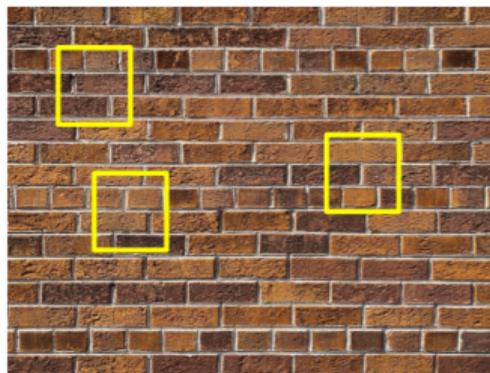
Coarse scales dominate

- L'architecture des CNN et les filtres locaux permettent cette analyse multi-échelle.

CNN et approches multi-échelles

A priori d'échelle

La séparation des échelles peut permettre de limiter l'effet de la grande dimension. Si les échelles locales ou globales dominent. On encode donc une **structure d'analyse multi-échelles dans les CNN**.

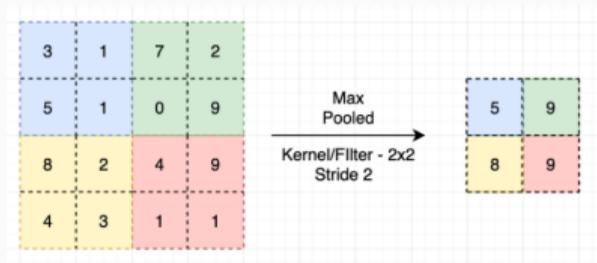


- L'architecture des CNN et les filtres locaux permettent cette analyse multi-échelle.

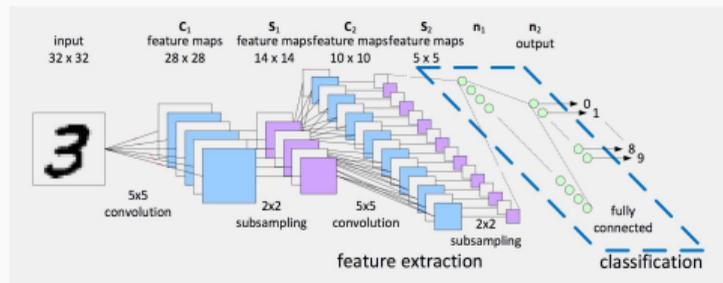
Architecture CNN

Couche de sous-échantillonnage

Soit un signal x en dimension d avec n données par dimension. L'opérateur de pooling est un opérateur de \mathbb{R}^{n^d} vers $\mathbb{R}^{\left(\frac{n}{k}\right)^d}$ ou $k \times d$ pixel sont transformés en 1 pixel. Si on prend la moyenne on parle pooling moyen local. Si on prend le maximum on parle pooling maximal local.



- Architecture globale:



Réseaux convolutifs

Réseaux sur graphes

Traitement du signal sur graphe

GNN spectraux

GNN spatiaux

Réseaux sur graphes

Réseaux convolutifs

Réseaux sur graphes

Traitement du signal sur graphe

GNN spectraux

GNN spatiaux

Des images vers les graphes

- On peut avoir des données structurées sur un objet autre qu'une grille.
- Par exemple en EDP, on a des données portés par un **maillage non structurés** associé à une géométrie complexe ou une variété.
- En apprentissage on considère des données portées par un **graphe**

Question

Comment généraliser les **réseaux convolutif** à des graphes/Maillages ?

- De quoi a besoin un réseaux convolutif ?
 - ▶ D'une operation de **convolution discrète** approchant $f(\mathbf{x}) = \int_{\Omega} k(\mathbf{x} - \mathbf{y})g(\mathbf{y})$,
 - ▶ De **filtres locaux** pour obtenir de la stabilité,
 - ▶ D'opération linéaire par rapport à la taille de l'entrée.
 - ▶ D'une operation de **"pooling"** afin de construit un réseaux multi-échelles.
 - ▶ De notion d'invariance/équivariance afin de réduire la dimension du problème.

Transformée de Fourier sur graphe

- Deux définitions de la **convolution**:

- ▶ Spatial:

$$f(\mathbf{x}) = \int_{\Omega} k(\mathbf{x} - \mathbf{y})g(\mathbf{y}),$$

- ▶ Spectral:

$$f(\mathbf{x}) = \mathcal{F}^{-1}(\mathcal{F}(k)\mathcal{F}(g))$$

avec \mathcal{F} la transformée de Fourier.

- Dans un premier temps c'est la deuxième qui a été utilisée pour généraliser les CNN aux graphes.
- Qu'est ce qui généralise la transformée de Fourier sur un **graphe/une variété** ?

Base de Fourier

La base de Fourier est composée des **vecteur propres du Laplacien**, généralisant les théorèmes spectraux des matrices définies positives.

Question

Peut t'on généraliser le **Laplacien sur une variété ou un graphe/maillage** ? Oui

Transformée de Fourier sur graphe II

- On peut définir un Laplacien sur une variété, **opérateur de Laplace Beltrami** et construire une base orthogonale de plus en plus oscillantes en le diagonalisant: exemple ici. Base du **traitement du signal géométrique**. **Et pour les graphes ?**

Hilbert sur les sommets d'un graphe

On note $\mathcal{H}(V, \chi)$ l'espace des fonctions définies sur les sommets. Il s'agit d'un espace de Hilbert $(\mathbb{R}^{|V|}, \langle \cdot, \cdot \rangle_V)$ muni du produit scalaire

$$\langle f, g \rangle_V = \frac{1}{n} \sum_{i=1}^n f_i g_i \chi_i$$

avec $\chi_i = \chi_{in}(\frac{1}{n} \sum_{j=1}^n w_{ji}) + \chi_{out}(\frac{1}{n} \sum_{j=1}^n w_{ij})$ (la somme des poids des arêtes entrantes et sortantes) et avec $\chi_{in}(x) > 0, \chi_{out}(x) > 0$ sur \mathbb{R}_+^* et $\chi_{in}(0) = 0, \chi_{out}(0) = 0$.

Hilbert sur les arêtes d'un graphe

On note $\mathcal{H}(E, \psi)$ l'espace des fonctions définies sur les arêtes. Il s'agit d'un espace de Hilbert $(\mathbb{R}^{|E|}, \langle \cdot, \cdot \rangle_E)$ muni du produit scalaire

$$\langle f, g \rangle_E = \frac{1}{2n^2} \sum_{i,j=1}^n f_{ij} g_{ij} \psi(w_{ij})$$

avec $\psi(x) > 0$ sur \mathbb{R}_+^* et $\psi(0) = 0$.

Transformée de Fourier sur graphe III

Dérivée

L'opérateur dérivée $d : \mathcal{H}(V, \chi) \rightarrow \mathcal{H}(E, \psi)$ peut être défini par

$$\forall e_{ij} \in E, \quad d(e_{ij}) = \gamma(w_{ij})(f(j) - f(i))$$

avec $f(i), f(j)$ la valeur de la fonction f aux sommets i et j . On peut aussi le définir avec:

$$\forall e_{ij} \in E, \quad d^n(e_{ij}) = \gamma(w_{ij}) \left(\frac{f(j)}{\sqrt{d(j)}} - \frac{f(i)}{\sqrt{d(i)}} \right)$$

avec $d(\cdot)$ le degré du noeud et $\gamma : \mathbb{R}_+^* \rightarrow \mathbb{R}_+^*$.

Dual

On peut définir l'opérateur dual avec la relation:

$$\langle df, g \rangle_V = \langle f, d^*g \rangle_E, \quad \forall f \in \mathcal{H}(V, \chi), g \in \mathcal{H}(E, \psi)$$

Le dual de l'opérateur dérivé d est donné par

$$\forall v_j \in V, \quad d^*(v_j) = \frac{1}{2n\chi(d_j)} \sum_{i=1}^1 \gamma(w_{ij})\psi(w_{ij})(g_{ij} - g_{ji})$$

avec $\psi(x) > 0$ sur \mathbb{R}_+^* et $\psi(0) = 0$.

Transformée de Fourier sur graphe IV

Laplacien sur graphe

L'opérateur **Laplacien sur un graphe** $\Delta : \mathcal{H}(V, \chi) \rightarrow \mathcal{H}(V, \psi)$ est défini par

$$\Delta(i) = d^*d$$

On peut obtenir une formule explicite:

$$\Delta(i) = \frac{1}{\chi(i)} \left[f(i) \frac{1}{n} \sum_{j=1}^n \gamma(w_{ji})^2 \psi(w_{ji}) - \frac{1}{n} \sum_{j=1}^n f(j) \gamma(w_{ji})^2 \psi(w_{ji}) \right]$$

- Soit D la matrice du graphe et W la matrice de poids

- Plusieurs Laplacien sur graphe:

- ▶ Le Laplacien dit **Marche aléatoire**:

obtenu avec $\frac{\gamma(w_{ji})^2 \psi(w_{ji})}{\chi(d_i)} = \frac{w_{ij}}{d_i}$

$$\Delta^{rw}f = (I_d - D^{-1}W)f$$

- ▶ Le Laplacien dit **non normalisé**:

obtenu avec $\frac{\gamma(w_{ji})^2 \psi(w_{ji})}{\chi(d_i)} = w_{ij}$

$$\Delta^u f = (D - W)f$$

- ▶ Le Laplacien dit **normalisé**:

$$\Delta^n f = D^{-\frac{1}{2}} (D - W) D^{-\frac{1}{2}}$$

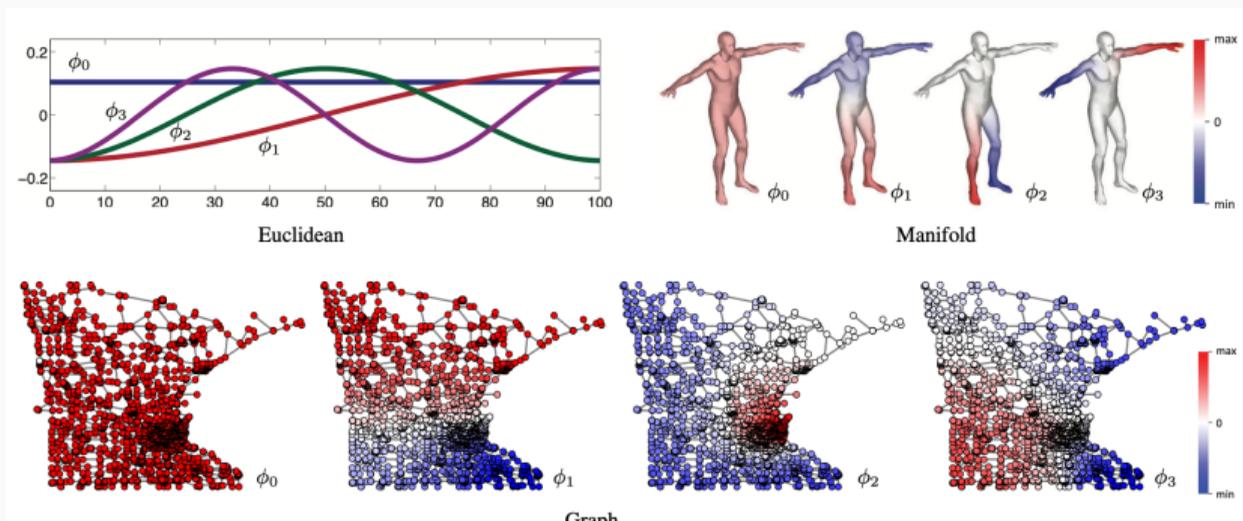
Transformée de Fourier sur graphe et GNN

Transformée de Fourier sur graphe

Soit L la matrice graphe Laplacien. Soit Φ les vecteurs propres de L alors la transformée de Fourier sur le graphe $G = [V, E]$ est donnée par

$$\mathcal{F}(f) = \Phi^t f$$

avec $f \in \mathcal{H}(V, \chi)$.



- Traitement du signal sur graphe: PyGSP

Réseaux convolutifs

Réseaux sur graphes

Traitement du signal sur graphe

GNN spectraux

GNN spatiaux

Convolution spectral et première Couche

On obtient donc que

$$f \star g = \Phi((\Phi^t f)(\Phi^t g))$$

et cela permet de définir une couche de convolution spectral sur graphe:

$$\mathbf{g}_l = \sigma \left(\sum_{l'=1}^q \Phi_k \Gamma_{l,l'} \Phi_k^T \mathbf{f}_{l'} \right),$$

avec $\mathbf{F} = (\mathbf{f}_1, \dots, \mathbf{f}_p)$ la matrice contenant p signaux de taille n et $\mathbf{G} = (\mathbf{g}_1, \dots, \mathbf{g}_q)$ la matrice contenant q signal de taille n . Le **filtre est paramétrisé directement en Fourier a travers** la matrice $\Gamma_{l,l'}$. On garde que les k 1er vecteurs propres.

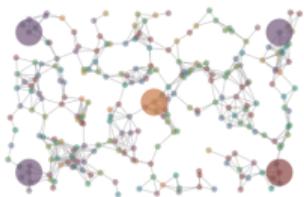
- Cette approche est couteuse: car il faut calculer Φ ($O(n^2)$) et elle ne produit pas de filtre locaux en espace.
- Pour éviter cela on propose le choix de filtre suivant: $\Gamma_{l,l'} = p_{\alpha_{l,l'}}(\Lambda)$ avec $p_{\alpha_{l,l'}}$ un polynôme et Λ les valeurs propres de L .
- **Résultat classique des matrices:** $\Phi P(\Lambda) \Phi^T = P(L)$.
- Cela limite **les couts de calcul et localise les filtre (Le Laplacien est local)**. Les paramètres deviennent ceux du polynôme.

GNN Spectraux II

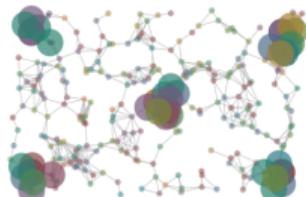
- Donc notre convolution peut se ramener a un **polynôme du graphe Laplacien**.

Opérateur de décalage sur graphe

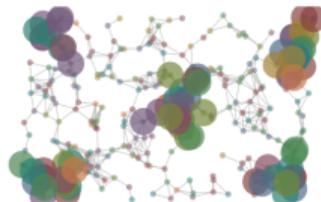
Un opérateur de décalage S est une représentation matricielle du graphe. Cela peut être la matrice d'adjacence A , ou une matrice du graphe Laplacien. L'appliquer **plusieurs fois permet de diffuser le signal sur le graphe**.



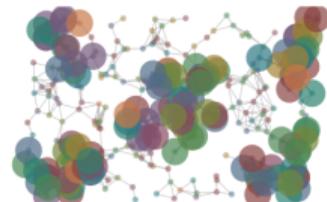
$S^0x = x$



S^1x



S^2x



S^3x

Filtre sur graphe

Soit $f \in H(V, \chi)$ un signal sur le graphe. On appelle filtre sur graphe une transformation du type:

$$L(f) = \sum_{i=1}^K \theta_k S^k f$$

GNN spectral

Soit $\mathbf{f} \in H(V, \chi)^p$ un vecteur de signaux p sur le graphe. Une couche est donnée par

$$\mathbf{g} = \sum_{i=1}^K S^i \mathbf{f} \Theta$$

avec $\mathbf{f} \in \mathbb{R}^{n,p}$, $\Theta \in \mathbb{R}^{p,q}$, $S \in \mathbb{R}^{n,n}$ et $\mathbf{g} \in \mathbb{R}^{n,q}$. Un GNN spectral est donné par la récurrence:

$$\mathbf{f}_{l+1} = \sigma \left(\sum_{i=1}^K S^i \mathbf{f}_l \Theta_l \right)$$

- Les premières architectures faisant des choix de polynômes particuliers. On va en introduire 2.

GNN Spectraux IV

- On va calculer des polynômes de matrices: pour éviter trop de calcul on propose des **polynômes de Chebyshev** construit de façon récurrente.
- Soit L la matrice du Laplacien.

ChebNet

La couche est donnée par

$$\mathbf{f}_{l+1} = \sigma \left(\sum_{k=0}^{K-1} T_k(L) \mathbf{f}_l \Theta_l \right)$$

avec $T_0 = I_d$, $T_1 = \hat{L}$ et $T_k = 2\hat{L}T_{k-1} - T_{k-2}$ et

$$\hat{L} = \frac{2L}{\lambda_{\max}} - I_d$$

- Les polynômes de Chebyshev prennent en entrée des valeurs dans $[-1, 1]$ d'où la normalisation \hat{L} .

GCN vanilla

On prend un ChebNet avec $K = 2$. Cela revient à

$$\mathbf{f}_{l+1} = \sigma \left(\sum_{k=0}^{K-1} (I_d + D^{-\frac{1}{2}} W D^{-\frac{1}{2}}) \mathbf{f}_l \Theta_l \right)$$

Matrice de Permutation

Une matrice carré P si $P \in \{0, 1\}^{n \times n}$ et si elle satisfait $P1 = 1$ et $P^T 1 = 1$.

Le produit $P^T x$ ré-ordonne les entrées du vecteur x .

Le produit $P^T S P$ effectue un ré-ordonne des lignes et colonnes de S .

- Soit un graphe défini par son **opérateur de décalage** S et un signal $x \in H(V, \chi)$ vivant sur les noeuds du graphe.
- Soit le couple (S, x) alors $(P^T S P, P^T x)$ est un re-ordonnancement du graphe et du signal. **Le signal reste le même.**

Filtres et permutations

- On applique le même filtre à signal sur graphe (S, x) et au signal permuté $(\hat{S}, \hat{x}) = (P^T S P, P^T x)$

- Filtres:

$$H(S)x = \sum_{k=0}^{K-1} h_k S^k x \quad H(\hat{S})\hat{x} = \sum_{k=0}^{K-1} h_k \hat{S}^k \hat{x}$$

- **Filtre $H(S)x$** : Coefficients h_k . Signal d'entrée x . Opérateur de shift S
- **Filtre $H(\hat{S})\hat{x}$** : Coefficients h_k . signal d'entrée permuté \hat{x} . Opérateur de décalage permuté \hat{S}

Théorème d'équivariance

On se donne une permutation P , On se donne un opérateur de $\hat{S} = P^T \mathbf{S} P$ et un signal $\hat{x} = P^T x$ auquel on a appliqué cette permutation alors:

$$H(\hat{S})\hat{x} = P^T H(S)x$$

Les **les filtres de graphes sont équivariant par permutation**. En effet permuter le signal et l'opérateur de décalage puis appliquer le filtre est donne le même résultat qu'appliquer la permutation au résultat du filtre.

- Preuve:

- ▶ On introduit le matrice de décalage et le signal permuté $\hat{S} = P^T S P$ and $\hat{x} = P^T x$

$$H(\hat{S})\hat{x} = \sum_{k=0}^{K-1} h_k \hat{S}^k \hat{x} = \sum_{k=0}^{K-1} h_k (P^T S P)^k P^T x$$

- ▶ Puisque ($P^T P = I_d$) alors on a ($P^T S P$)^k = $P^T S^k P$
- ▶ On injecte cela dans l'expression du filtre ce qui donne

$$H(\hat{S})\hat{x} = \sum_{k=0}^{K-1} h_k P^T S^k P P^T x = \sum_{k=0}^{K-1} h_k P^T S^k I_d x = P^T \sum_{k=0}^{K-1} h_k S^k x = P^T H(S)x$$

Théorème d'équivariance

On se donne une permutation P , On se donne un opérateur de $\hat{S} = P^T S P$ et un signal $\hat{x} = P^T x$
Soit un réseau GNN à filtres:

$$x_l = \sigma \left[\sum_{k=0}^{K-1} h_{lk} S^k x_{l-1} \right] = \sigma[H_l(S)x_{l-1}] \quad \Phi(x; S, \mathcal{H}) = x_L$$

on a

$$\Phi(\hat{x}; \hat{S}, \mathcal{H}) = P^T \Phi(x; S, \mathcal{H})$$

Les **GNN a filtres** sont donc invariants par permutations.

- Preuve:

- ▶ On suppose que l'entrée de la couche l satisfait $\hat{x}_{l-1} = P^T x_{l-1}$. On a

$$\hat{x}_l = \sigma \left[H_l(\hat{S}) \hat{x}_{l-1} \right] = \sigma \left[P^T H_l(S) x_{l-1} \right] = P^T \sigma \left[H_l(S) x_{l-1} \right] = P^T x_l$$

car le filtre est **équivariant** et σ est identique sur chaque composante.

- ▶ Par hypothèse on a $\hat{x} = P^T x$ à la première couche.

Transférabilité

Un filtre appris est valide pour tous les graphes avec la même structure. En gros si ils discrétisent la même variété/géométrie. Utile notamment pour les maillages.

Réseaux convolutifs

Réseaux sur graphes

Traitement du signal sur graphe

GNN spectraux

GNN spatiaux

- Les réseaux basés sur une **approche spectrale** de la convolution ne supporte pas le changement de topologie.
- Pour résoudre ce problème il a été proposé des approches **ou la convolution discrète est construit dans le domaine spatial**.
- On se donne une sous-variété \mathcal{M} de dimension D (qui sera ensuite remplacer par un graphe).

Opérateur de voisinage

On définit un opérateur de voisinage au point $\mathbf{x} \in \mathcal{M}$ comme

$$D(\mathbf{x})f = \int_{\mathcal{M}} w_{\theta}(u(\mathbf{x}, \mathbf{y}))f(\mathbf{y})d\mathbf{y}$$

avec w_{θ} une fonction de point et $u(\mathbf{x}, \mathbf{y})$ une fonction qui renvoie des pseudo coordonnées locales. Sur un graphe cela devient:

$$D(\mathbf{x})f = \sum_{\mathbf{y} \in \mathcal{N}(\mathbf{x})} w_{\theta}(u(\mathbf{x}, \mathbf{y}))f(\mathbf{y})$$

GNN spatiaux II

Convolution spatiale

On va définir la convolution spatiale sur graphe comme étant:

$$f \star g = \sum_{i=1}^q g_i D_i(\mathbf{x}) f$$

avec $D(\mathbf{x})f = \sum_{\mathbf{y} \in \mathcal{N}(\mathbf{x})} w_{j,\theta}(u(\mathbf{x}, \mathbf{y})) f(\mathbf{y})$

- Dans une couche convolutive **on apprend les g_j** . Exemple d'architecture:

- ▶ **CNN Géodésique:** $\mathbf{u}(\mathbf{x}, \mathbf{y}) = (\rho, \theta)$ la distance et l'angle entre les deux vecteurs.

$$w(u(\mathbf{x}, \mathbf{y})) = e^{-\frac{1}{2} \begin{pmatrix} \mathbf{u}-\mathbf{u}_j \\ 0 \end{pmatrix} \begin{pmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_\theta^2 \end{pmatrix} \begin{pmatrix} \mathbf{u}-\mathbf{u}_j \\ 0 \end{pmatrix}}$$

avec \mathbf{u}_j , σ_r^2 et σ_θ^2 fixe.

- ▶ **CNN anisotrope:** $\mathbf{u}(\mathbf{x}, \mathbf{y}) = (\rho, \theta)$ la distance et l'angle entre les deux vecteurs.

$$w(u(\mathbf{x}, \mathbf{y})) = e^{-\frac{1}{2} \begin{pmatrix} \mathbf{u}R(\theta_j) \\ 0 \end{pmatrix} \begin{pmatrix} \alpha & 0 \\ 0 & 1 \end{pmatrix} R^{-1}(\theta_j) \mathbf{u}}$$

avec α et $R(\theta_j)$ fixe.

- ▶ **Monet:** $\mathbf{u}(\mathbf{x}, \mathbf{y}) = (\rho, \theta)$ la distance et l'angle entre les deux vecteurs (ou autre chose).

$$w(u(\mathbf{x}, \mathbf{y})) = e^{-\frac{1}{2} \left((\mathbf{u}-\mu_j) \Sigma^{-1} (\mathbf{u}-\mu_j) \right)}$$

avec \mathbf{u}_j , Σ_j **entraînable**.

Message Passing I

- Depuis un cadre très général a été proposé pour les réseaux sur graphe: **le message passing**.

Message Passing

Soit (\mathbf{x}_i) les informations aux sommets du graphe et informations aux arrêtes \mathbf{e}_{ji} . Le message passing est une transformation de la forme suivante

$$\mathbf{x}'_i = \phi\left(\mathbf{x}_i, \square_{j \in \mathcal{N}(i)} \psi(\mathbf{x}_i, \mathbf{x}_j, \mathbf{e}_{ji})\right),$$

avec

- ψ : Une fonction combinant l'information au noeud, a son voisin $\mathbf{x}_i, \mathbf{x}_j$ et les informations aux arrêtes \mathbf{e}_{ji} si il y en a.
- $\square_{j \in \mathcal{N}(i)}$: une **fonction d'agrégation invariante par permutation**, (somme, moyenne, soft max etc.);
- ϕ : une transformation finale qui utilise l'information au noeud et le résultat de l'agrégation). Elle contient souvent l'activation.

Message Passing II

- Il a été proposé cette classification.

Familles de GCN

$$\mathbf{h}_u = \phi(\mathbf{x}_i, \square_{j \in \mathcal{N}_i} c_{ij} \psi(\mathbf{x}_j)), \quad \text{Convolution}$$

$$\mathbf{h}_u = \phi(\mathbf{x}_i, \square_{j \in \mathcal{N}_i} a(\mathbf{x}_i, \mathbf{x}_j) \psi(\mathbf{x}_j)), \quad \text{Attention}$$

$$\mathbf{h}_u = \phi(\mathbf{x}_i, \square_{j \in \mathcal{N}_i} \psi(\mathbf{x}_i, \mathbf{x}_j)), \quad \text{Message-Passing}$$

avec ϕ et ψ des réseaux et \square une opération redinvariante par permutation.

- Ces trois familles sont de plus en plus générales.
- Les couches a filtre sur graphe rentre dans la 1er catégorie. Les réseaux spatiaux introduit précédemment et un réseau assez utilisé GAT rentre dans la deuxième catégorie.

