Motifs locaux et super-graphe pour la classification de graphes symboliques avec des réseaux convolutionnels

Évariste Daller Luc Brun Sébastien Bougleux Olivier Lézoray

Normandie Univ, UNICAEN, ENSICAEN, CNRS, GREYC, Caen
{evariste.daller, bougleux, olivier.lezoray}@unicaen.fr, luc.brun@ensicaen.fr

Résumé

Les réseaux convolutionnels ont révolutionné le domaine de l'apprentissage machine. Ces réseaux s'appliquent naturellement aux images, vidéos et aux sons. En revanche, la structure fixe de leur couche d'entrée ne permet pas de les étendre facilement à des structures de topologie arbitraire tels que les graphes. On peut citer comme exemples d'applications la prédiction de propriétés de molécules chimiques ou la classification de maillages 3D. Dans le cadre de graphes symboliques, nous proposons une méthode permettant d'appliquer des réseaux basés sur une topologie fixe de la couche d'entrée à des graphes de topologie arbitraire. Nous proposons également d'enrichir l'information contenu dans chaque sommet pour améliorer la prédiction de ses propriétés ainsi qu'une nouvelle couche permettant d'interfacer des graphes de topologie arbitraire avec une couche entièrement connectée.

Mots Clef

CNN de graphes, classification de graphes, distance d'édition entre graphes.

Abstract

Convolutional neural networks (CNN) have deeply impacted the field of machine learning. These networks designed to process objects with a fixed topology readily apply to images, videos and sounds but can not be easily extented to structures with an arbitrary topology such as graphs. Examples of applications of machine learning to graphs include the prediction of the properties of molecular graphs or the classification of 3D meshes. Within the symbolic graphs framework, we propose a method to extend networks based on a fixed topology to input graphs with an arbitrary topology. We also propose an enriched feature vector attached to each vertex of a chemical graph in order to improve the prediction of its properties as well as a new bottleneck layer allowing to connect arbitrary topological graphs on a fully connected layer.

Keywords

Graph-CNNs, graph classification, graph edit distance.

1 Introduction

Les réseaux convolutionnels (CNN) [14] ont considérablement impacté le domaine de l'apprentissage automatique et des domaines connexes comme la vision par ordinateur. Initialement, ces réseaux permettent d'analyser des signaux définis spatialement par des grilles cartésiennes (typiquement sons et images), en effectuant principalement deux types d'opération : la convolution et la descente en résolution (composée de *coarsening* et *pooling*). Ils ne peuvent donc pas être directement utilisés pour analyser des signaux définis spatialement par des grilles quelconques, mais aussi pour analyser des ensembles de points, des maillages 3D ou des graphes. Plusieurs travaux récents [5, 10, 9, 4] proposent d'étendre les CNN pour analyser ces types de données. Nous nous focalisons ici sur les graphes.
Une première approche [5, 9] issue du traitement du signal sur graphe utilise le lien entre convolution et transformée de Fourier ainsi que les fortes similarités existant entre la transformée de Fourier et la décomposition spectrale d'un graphe. Du fait de l'utilisation du Laplacien, ces méthodes sont restreintes à la prédiction des propriétés de signaux sur un graphe mais ne peuvent prédire des propriétés de graphes de topologie arbitraire.
Une autre famille d'approches [13, 10, 1, 15, 20, 18, 17], également basée sur la notion de convolution sur graphe, n'utilisent pas l'analyse spectrale pour apprendre le poids des filtres. La problématique principale est alors la mise en correspondance des poids de convolution avec le voisinage (éventuellement étendu) de chaque sommet. En effet, d'une manière générale, les graphes ne permettent pas de distinguer naturellement un ensemble restreint de directions commune aux différents sommets (comme par exemple haut, bas, gauche, droite).
Dans cet article, nous présentons plusieurs contributions pouvant être appliquées à différents types de CNN sur graphes. Nous proposons d'étendre le champ d'application des réseaux basés sur l'analyse spectrale en utilisant un super-graphe d'une base d'apprentissage comme couche d'entrée du réseau. Cette méthode sera aussi appliquée à d'autres type de modèles. De plus, indépendamment de la méthode de convolution, nous proposons également une caractérisation vectorielle de chaque sommet d'un graphe symbolique, plus riche que son simple symbole. Enfin,

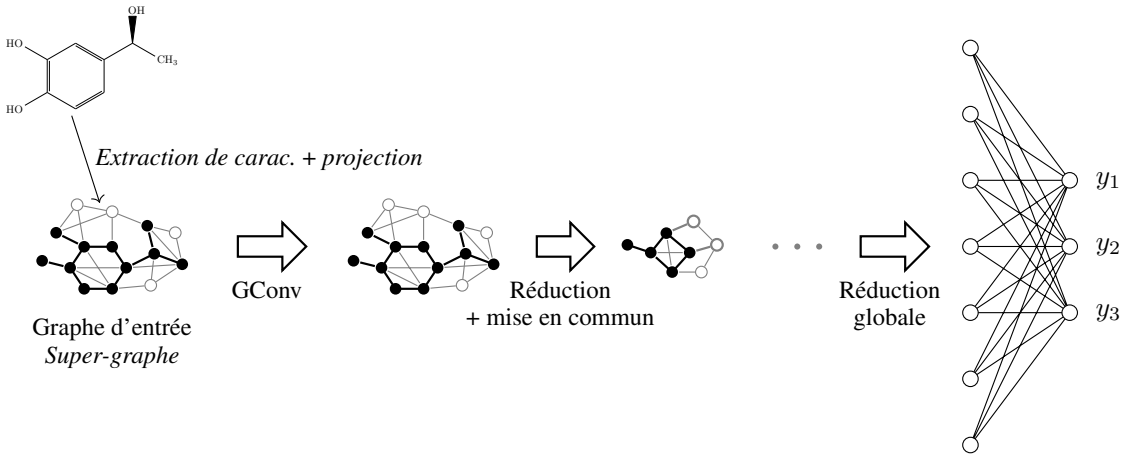


FIGURE 1 – Illustration de nos propositions appliqué un réseau convolutionnel sur graphe existant [9].

nous proposons une nouvelle couche de réduction en fin de réseau, dont les données d'entrée peuvent varier en taille et en topologie. Ces éléments peuvent être utilisés avec des réseaux existants.

Après un rappel des travaux connexes en Section 2, la Section 3 présente nos contributions. Les apports de celles-ci sont validés expérimentalement en Section 4.

2 Travaux connexes

Comme mentionné en Section 1, une première approche pour construire des réseaux de neurones sur graphes utilise l'analyse spectrale. Ainsi, Bruna *et al.* [5] définissent l'opération de convolution à partir du spectre du Laplacien du graphe servant de couche d'entrée au réseau. Toutefois cette approche nécessite une coûteuse décomposition en valeurs singulières du Laplacien lors de la création du réseau de convolution ainsi que de coûteuses multiplications matricielles lors de la phase de test. Ces limitations sont en partie résolues par Defferard *et al.* [9] qui proposent une implémentation rapide de la convolution (CGCNN), grâce à un codage du filtre à l'aide de polynômes de Chebyshev permettant une définition récursive et efficace de l'opération de filtrage. Cette convolution combinée à une opération de décimation (pooling) efficace permet de construire une pyramide de graphes réduits à partir d'une couche d'entrée de topologie fixe. Chaque graphe ainsi réduit correspond à une couche du réseau (Figure 1). Notons toutefois que ces deux méthodes reposent sur une structure du graphe fixée a priori. Ceci permet donc à ces réseaux de traiter différents signaux sur la couche d'entrée mais pas de prédire des propriétés de graphes dont la topologie peut varier.

Les réseaux spatiaux définissent les filtres directement dans l'espace des graphes et n'utilisent donc pas directement l'analyse spectrale. Kipf et Welling [13] ont proposé un modèle (GCN) pouvant être vu comme une approximation des filtres spectraux locaux de [9]. Si d_l dénote la dimension des données de la couche l, chaque étape de convolution est basée sur d_{l+1} opérations de filtrages associant un poids à chaque composante des vecteurs de caractéristiques des sommets. Chaque filtre correspond donc à une colonne de la matrice de poids $W^{(l)} \in \mathbb{R}^{d_l \times d_{l+1}}$ dans la règle de propagation entre une couche l et la suivante donnée en équation 1.

$$X^{(l+1)} = f(\hat{A}X^{(l)}W^{(l)}) \tag{1}$$

où f est une fonction d'activation, et $\hat{A}$ une normalisation de la matrice d'adjacence A (*i.e.* $\hat{A} = \tilde{D}\tilde{A}\tilde{D}^{-\frac{1}{2}}$ où $\tilde{A} = A + Id$, et $\tilde{D}$ est la matrice diagonale des degrés : $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$).

Dans cette formulation, les filtres ne dépendent pas du Laplacien, ce qui permet l'utilisation d'un même réseau sur des graphes différents. Le modèle proposé par Duvenaud *et al.* [10] pour l'extraction de fingerprints moléculaires est très proche de [13], mais considère une matrice de poids par degrés de sommets.

Ces deux dernières méthodes pondèrent les composantes des vecteurs attachés aux sommets et sont restreintes à des convolutions dans le voisinage immédiat de chaque sommet. Atwood et Towsley [1] (avec DCNN) reprennent quant à eux la notion de diffusion à k-pas introduite par Defferard *et al.* [9], en considérant les puissances d'une matrice de transition définie à partir de la matrice d'adjacence du graphe. Il s'agit donc d'une méthode non locale dans laquelle une pondération des caractéristiques est appliquée à une boule centrée sur le sommet considéré. Soit H le rayon de cette boule en nombre de pas, on fait correspondre à une couche l une matrice $X_h^{(l)}$ par nombre de pas $h \leq H$ par la règle de propagation suivante :

$$X_h^{(l+1)} = f(w_h \odot \hat{A}^h X^{(l)}) \tag{2}$$

où $w_h^{(l)} \in \mathbb{R}^{d_l}$ est un vecteur de poids appris.

D'autres heuristiques ont également été proposées pour définir la convolution. Niepert *et al.* [15] se basent sur un séquençage des sommets voisins suivant un ordre canonique.

Verma *et al.* [20] réalisent un appariement doux entre des matrices de poids et les sommets du voisinage et proposent d'apprendre cet appariement, relativement aux caractéristiques des sommets. Le réseau proposé par Simonovsky et Komodakis [18] apprend des poids associés à chaque label d'arrête. Finalement, Sankar *et al.* [17] déduisent les poids de convolution des rôles des sommets dans des sous-graphes locaux.

3 Contributions

Dans cette section, nous présentons des caractéristiques adaptées à des graphes symboliques, en particulier des graphes moléculaires, une couche d'entrée d'un réseau consistant en un super-graphe d'une base d'apprentissage, ainsi qu'une nouvelle couche de réduction globale admettant un graphe d'entrée de taille et topologie variables.

3.1 Sélection de caractéristiques

La convolution ne peut pas directement être appliquée à des graphes symboliques, du fait de l'impossibilité de réaliser des sommes de symboles. Ces derniers sont donc généralement transformés en vecteurs unitaires de $\{0,1\}^{|\mathcal{L}|}$, où $\mathcal{L}$ est un ensemble de symboles, de la même manière que dans [1, 10, 18] pour encoder le type d'atome dans des graphes chimiques. Toutes les composantes d'un de ces vecteur sont nulles sauf celle correspondant au symbole.

Cet encodage a un inconvénient principal, la taille du noyau de convolution est généralement beaucoup plus petite que $|\mathcal{L}|$. En ajoutant le fait que ces vecteurs sont très creux, la convolution produit des moyennes qui n'ont pas beaucoup de sens pour la réduction de dimension. De plus, l'information attachée aux arêtes n'est généralement pas utilisée. Pour répondre à ces problèmes, nous proposons d'extraire des caractéristiques moins locales que les labels de sommets et moins creuses. À chaque sommet, nous attachons un vecteur représentant la distribution de motifs locaux dans le voisinage de ce sommet.

Considérons $G = (V, E, \sigma)$ un graphe attribué, avec V un ensemble de sommets, $E \subset V \times V$ un ensemble d'arêtes, et σ une fonction d'étiquetage des sommets.

Soit $\mathcal{N}_u$ le voisinage d'un sommet $u \in V$. Pour chaque sous-ensemble $S \subseteq \mathcal{N}_u$, le sous-graphe $M_u^S = (\{u\} \cup S, E \cap (\{u\} \cup S) \times (\{u\} \cup S), \sigma)$ est connecté (par u) et défini un motif local à u. L'énumération de ces sous-ensembles de $\mathcal{N}_u$ donnent tous les motifs locaux de u, qui peuvent être organisés un sein d'un vecteur codant le nombre d'occurrences de chacun de ces motifs. La figure 2 illustre le calcul d'un tel vecteur de caractéristiques. Notons que dans dans le cadre des graphes chimiques, le degré maximal est borné et généralement inférieur à 5.

Pendant la phase d'apprentissage, les motifs des sommets des graphes de la base déterminent un dictionnaire ainsi que la dimension des vecteurs de caractéristiques attachés à chaque sommet.

Lors de la phase de test, chaque sommet d'un graphe d'entrée est valué de la même manière avant d'être mis en entrée du réseau de neurones. Tout motif d'un graphe d'entrée non présent dans le dictionnaire appris lors de la phase d'apprentissage est alors ignoré. De manière à assurer encore la compacité de l'espace des caractéristiques, nous appliquons une ACP à l'ensemble de ces vecteurs et projetons chaque vecteur sur un sous-espace contenant 95% de l'information initiale.

Motif	C	C—O	C=O			
Fréquence	1	2	1	1	2	1

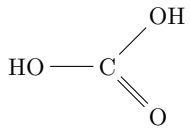
Graphe G : 

FIGURE 2 – Fréquences des motifs associés au sommet central (C) du graphe G

3.2 Super-graphe comme couche d'entrée

Comme mentionné en Section 1, les méthodes implémentant la convolution sur graphes à l'aide de l'analyse spectrale [5, 9] nécessitent un graphe d'entrée du réseaux de topologie fixe. Ces méthodes ne peuvent donc traiter que des fonctions définies sur un graphe de topologie fixée et ne peuvent prédire les propriétés de graphes topologiquement différents. Nous nous proposons ici de lever cette restriction en utilisant comme couche d'entrée du réseau, un graphe défini à partir de l'ensemble des graphes d'une base d'apprentissage.

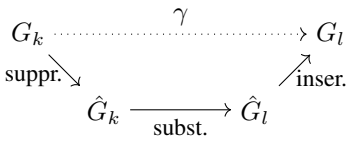
Un *super-graphe* de deux graphes G_1 et G_2 est un graphe S tel que G_1 et G_2 sont isomorphes à un sous-graphe de S. Plus généralement, un super-graphe d'un ensemble de graphes $\mathcal{G} = \{G_k = (V_k, E_k, \sigma_k, \phi_k)\}_{k=1}^{k=n}$ est un graphe $S = (V_S, E_S, \sigma_S, \phi_S)$ tel que tout graphe de $\mathcal{G}$ est isomorphe à un sous-graphe de S. Soient deux ensembles complémentaires $\mathcal{G}_1, \mathcal{G}_2 \subseteq \mathcal{G}$ avec $\mathcal{G}_1 \cup \mathcal{G}_2 = \mathcal{G}$, un super-graphe commun à un super-graphe de $\mathcal{G}_1$ et à un super-graphe de $\mathcal{G}_2$ est un super-graphe de $\mathcal{G}$. On alors peut définir un super-graphe de $\mathcal{G}$ en appliquant récursivement cette propriété. On obtient alors un arbre hiérarchique de super-graphes, dont la racine est un super-graphe de $\mathcal{G}$ et les feuilles sont les graphes de $\mathcal{G}$ (figure 3). Nous présentons une méthode permettant de construire hiérarchiquement un super-graphe formé d'un nombre minimum d'éléments.

Un super-graphe de $\mathcal{G}$ est un *plus petit super-graphe commun* (MCS) si il n'y a pas d'autre super-graphe $S' = (V_{S'}, E_{S'}, \sigma_{S'}, \phi_{S'})$ de $\mathcal{G}$ avec $(|V_{S'}| < |V_S|) \vee ((|V_{S'}| = |V_S|) \wedge (|E_{S'}| < |E_S|))$. La construction d'un tel super-graphe est un problème difficile et peut être lié à la notion suivante.

Un *plus grand sous-graphe commun* (mcs) à deux graphes G_k et G_l est un graphe $G_{k,l}$ isomorphe à un sous-graphe

$\hat{G}_k$ de G_k et à un sous-graphe $\hat{G}_l$ de G_l, et tel qu'il n'existe pas de sous-graphe G' commun à G_k et G_l avec $(|V_{G'}| > |V_{G_{k,l}}|) \vee ((|V_{G'}| = |V_{G_{k,l}}|) \wedge (|E_{G'}| > |E_{G_{k,l}}|))$. Pour un mcs donné $G_{k,l}$, le graphe S obtenu de $G_{k,l}$ en ajoutant les éléments de G_k absents de $\hat{G}_k$ et les éléments de G_l absents de $\hat{G}_l$ est alors un MCS de G_k et G_l. Cette propriété montre qu'un MCS peut être construit à partir d'un mcs. Ces notions sont toutes deux liées à la notion d'appariement de graphe correcteur d'erreur et de distance d'édition entre graphe [6].

Distance d'édition entre graphes. La distance d'édition entre graphes (GED) capture la quantité minimale de distorsion à appliquer à un graphe attribué G_k pour le transformer en un graphe attribué G_l en éditant itérativement la structure et les attributs de G_k, jusqu'à ce qu'on obtienne G_l. La séquence d'*opérations d'éditions* γ, appelée chemin d'édition, transforme G_k en G_l. Son coût est mesuré par $L_c(\gamma) = \sum_{o\in\gamma} c(o)$, où $c(o)$ est le coût de l'opération o. Parmi tous les chemins d'édition de G_k à G_l, ensemble noté $\Gamma(G_k, G_l)$, la GED entre G_k et G_l est définie comme le coût minimal des chemins de $\Gamma(G_k, G_l)$: $d(G_k, G_l) = \min_{\gamma\in\Gamma(G_k,G_l)} L_c(\gamma)$.
La distance d'édition $d(G_k, G_l)$ est unique, mais peut être atteinte par plusieurs chemins d'éditions différents, *i.e.* $\text{card}(\text{argmin}_{\gamma\in\Gamma(G_k,G_l)} L_c(\gamma)) \geq 1$.
On peut montrer [3] que sous certaines contraintes, un chemin d'édition peut être réorganisé en une succession de suppressions, suivie d'un ensemble de substitutions puis d'un ensemble d'insertions. De plus, chaque élément n'est affecté par une opération qu'une seule fois au maximum. On peut donc définir deux graphes $\hat{G}_k = (\hat{V}_k, E_k \cap \{\hat{V}_k \times \hat{V}_k\}, \sigma_k, \phi_k)$ et $\hat{G}_l = (\hat{V}_l, E_l \cap \{\hat{V}_l \times \hat{V}_l\}, \sigma_l, \phi_l)$ avec $\hat{V}_k \subseteq V_k$ et $\hat{V}_l \subset V_l$ représentant les principales étapes du chemin d'édition γ :



Par construction, $\hat{G}_k$ et $\hat{G}_l$ sont structurellement isomorphes et on appelle appariement de graphes correcteur d'erreur (*ECGM*) entre G_k et G_l une fonction bijective $f : \hat{V}_k \to \hat{V}_l$ faisant correspondre les sommets des deux graphes (on peut déduire un appariement des arêtes à partir des sommets). Pour des fonctions de coûts spécifiques [6], si un ECGM f correspond à un chemin d'édition optimal entre G_k et G_l, alors $\hat{G}_k$ et $\hat{G}_l$ sont des mcs de G_k et G_l. De plus, ajouter à un mcs de G_k et G_l les éléments manquants des deux graphes donne un MCS de G_k et G_l. Nous utilisons cette propriété pour construire un super-graphe global d'un ensemble de graphe.

Construction du super-graphe. La construction hiérarchique proposée d'un super-graphe commun à un ensemble de graphes $\mathcal{G} = \{G_k\}_{k=1}^{k=n}$ est illustrée en figure 3. Chaque

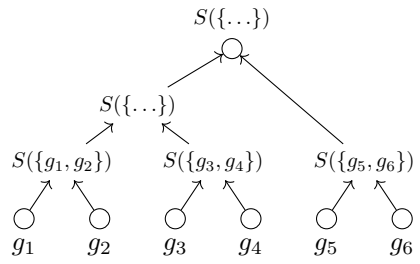


FIGURE 3 – Construction d'un super-graphe d'une base de graphes. Les graphes sont appariés en fonction de leurs distances d'édition, puis agrégés en super-graphes. Ces derniers sont alors appariés de la même façon, et ainsi de suite.

niveau k de la pyramide contient N_k graphes, fusionnés par paires pour produire $\lfloor N_k/2 \rfloor$ super-graphes. De manière à restreindre la taille du super-graphe final, une heuristique naturelle consiste à fusionner des graphes proches en terme de distance d'édition. Ceci peut se formaliser comme la recherche d'un couplage maximum $M^\star$ du graphe complet sur $\mathcal{G}$, minimisant :

$$M^\star = \min_M \sum_{(g_i,g_j)\in M} d(g_i, g_j) \tag{3}$$

où $d(\cdot,\cdot)$ désigne la distance d'édition. Nous utilisons pour cela l'algorithme d'Edmonds [11] à chaque niveau. Un avantage de ce type de construction est qu'il est hautement parallélisable. Cependant, le calcul de la distance d'édition étant NP difficile, les algorithmes résolvant le problème exact ne peuvent être raisonnablement utilisés ici. Nous utilisons donc une approximation bipartite [16] pour calculer $d(\cdot,\cdot)$ et résoudre 3, ainsi qu'un algorithme plus précis mais aussi plus coûteux temporellement [7] pour calculer les ECGM à l'origine des super-graphes.

3.3 Projections comme données d'entrée

Le super-graphe calculé dans la section précédente peut être utilisé comme couche d'entrée d'un réseau convolutionnel sur graphes. Ceci permet notamment, d'envisager l'utilisation de méthodes spectrales pour la convolution [5, 9]. Toutefois il nous faut au préalable transformer chaque graphe présenté au réseau en un signal sur le super-graphe. Cette opération est basée sur la notion de projection, une notion voisine de la distance d'édition entre graphe.

Définition 1 (Projection)**.** *Soit f un ECGM entre un graphe attribué $G = (V_G, E_G, \sigma, \phi)$ et un graphe $S = (V_S, E_S)$, et soit $(\hat{V}_S, \hat{E}_S)$ le sous-graphe de S défini par f. Une projection de G sur S est un graphe $P_S^f(G) = (V_S, E_S, \sigma_P, \phi_P)$ où :*

$$\begin{cases} \forall u \in \hat{V}_S, & \sigma_P(u) = (\sigma \circ f^{-1})(u), \\ \forall u \in V_S - \hat{V}_S, & \sigma_P(u) = 0_{\mathcal{L}}. \end{cases}$$

De même pour les arêtes :

$$\begin{cases} \forall \{u,v\} \in \hat{E}_S, & \phi_P(\{u,v\}) = (\phi \circ f^{-1})(\{u,v\}), \\ \forall \{u,v\} \in E_S - \hat{E}_S, & \phi_P(\{u,v\}) = 0_{\mathcal{L}} \end{cases}$$

Soit $\{g_1, \ldots, g_n\}$ une base d'apprentissage et S le super-graphe associé. La projection $P_S^f(g_i)$ d'un graphe g_i induit un signal sur le super-graphe S (Définition 1) associé à un label ou une valeur à prédire. Pour chaque sommet de S appartenant à la projection de g_i, ce signal est égal à l'attribut du sommet correspondant dans g_i ; il est nul partout ailleurs. De plus, si la distance d'édition $d(g_i, S)$ peut être calculée par plusieurs chemins d'édition et donc plusieurs ECGM $f_1, \ldots, f_m$ (l'argmin n'est pas unique, Section 3.2), le graphe g_i sera associé à plusieurs projections $P_S^{f_1}(g_i), \ldots, P_S^{f_m}(g_i)$. Ce dernier point permet de réaliser naturellement une augmentation de données en apprenant plusieurs représentations équivalentes d'un même graphe associées à une même prédiction. Précisons que des graphes symétriques peuvent induire plusieurs projections identiques. Cependant, l'utilisation du vecteur de caractéristiques proposé en Section 3.1 réduit le nombre de graphes concernés, car elle intègre une information plus riche (caractérisant le voisinage) que les symboles des sommets seuls.

Les données peuvent être encore augmentées en considérant des ECGM non minimaux. A cette fin, en utilisant [7], un ensemble d'ECGM entre g_i et le super-graphe S est trié par ordre croissant de coûts associés aux chemins d'édition. Nous considérons les m premiers ECGM de plus faible coût et ajoutons les μm suivants, où μ est un paramètre de l'algorithme. Le nombre d'ECGM disponibles (fournis par [7]) peut être contrôlé et a été fixé à 16 dans nos expériences. Un graphe de la base de test peut également avoir plusieurs projections. Nous avons décidé, dans ce cas d'affecter à chaque graphe sa classe majoritaire parmi ses projections.

3.4 Couche de réduction de taille variable

Le perceptron multi-couche habituellement utilisé dans la dernière partie des réseaux multi-couches requiert une couche précédente de taille et de topologie fixe. Cependant, sans utiliser de super-graphe, cette dernière condition n'est généralement pas satisfaite. De fait, la taille et la topologie des couches intermédiaires dépendent du graphe d'entrée qui peut varier. La plupart des réseaux de neurones sur graphes résolvent ce problème en utilisant une étape de mise en commun globale (*global pooling*).

En général, cette étape consiste a moyenner les vecteurs de caractéristiques composante par composante, sur l'ensemble des sommets du graphe donné en entrée. Cette stratégie est nommée *Global Average Pooling* (GAP). Si les vecteurs de caractéristiques des sommets de la couche précédente ont une dimension D, cette étape produit un vecteur $\left(\frac{1}{|V|}\sum_{v\in V} h_c(v)\right)_{c\in\{1,\ldots,D\}} \in \mathbb{R}^D$ pour tout le graphe, décrivant la valeur moyenne des sommets pour chaque composante, avec V l'ensemble des sommets de la couche précédente et $h_c(v)$ la valeur de la $c^{\text{ième}}$ composante.

Nous proposons d'améliorer cette étape en prenant en compte la distribution des activations des caractéristiques à travers le graphe plutôt que leur moyenne. Cependant, un simple histogramme ne peut être utilisé ici car il nécessite plusieurs opérations non dérivables. Nous proposons d'approximer cet histogramme par une version continue basée sur le moyennage d'activations gaussiennes.

Étant donné une caractéristique c, la taille d'un *bin* k de ce pseudo-histogramme est donné par :

$$b_{ck}(h) = \frac{1}{|V|}\sum_{v\in V} e^{\frac{-(h_c(v)-\mu_{ck})^2}{\sigma_{ck}^2}} \tag{4}$$

avec $h(v)$ un vecteur de dimension D pour chaque sommet v. Dans cet article, le paramètre μ_{ck} et σ_{ck} sont fixés et non appris par le réseau. La taille de notre couche de réduction est $D \times K$ où K est le nombre (fixé) de gaussiennes pour chaque composante.

Notons que cette couche n'a aucun paramètre appris, et que par conséquent, une procédure de rétro-propagation du gradient va ajuster les poids $\alpha_c(i)$ de la couche précédente h pour chaque sommet $i \in V$ en fonction de la dérivée partielle de la fonction de coût L : $\frac{\partial L}{\partial \alpha_c(i)} = \frac{\partial L}{\partial b_{ck}(h)}\frac{\partial b_{ck}(h)}{\partial h_c(i)}\frac{\partial h_c(i)}{\partial \alpha_c(i)}$. Le deuxième facteur, c'est-à-dire la dérivée de la couche de réduction relativement à son entrée est :

$$\frac{\partial b_{ck}(h)}{\partial h_c(i)} = \frac{1}{|V|}\cdot\frac{-2(h_c(i)-\mu_{ck})}{\sigma_{ck}^2}\cdot e^{\frac{-(h_c(i)-\mu_{ck})^2}{\sigma_{ck}^2}} \tag{5}$$

et est bornée par $-\frac{\sqrt{2}}{|V|\sigma_{ck}}e^{-1/2}$ et $\frac{\sqrt{2}}{|V|\sigma_{ck}}e^{-1/2}$.

4 Expériences

Afin de valider nos contributions, nous utilisons trois réseaux de base : DCNN [1], GCN [13] et CGCNN [9] auxquels nous ajoutons nos propositions définies en Section 3. Toutes les expériences décrites dans cette section sont liées à une tache de classification.

4.1 Jeux de données

Nous appliquons nos méthodes à un ensemble de jeux de données chimiques standard : NCI1, MUTAG, ENZYMES et PTC, ainsi qu'aux deux jeux de données : MAO et PAH [1]. Les caractéristiques de ces bases sont résumées dans la Table 1. NCI1 [21] contient 4110 composés chimiques classés suivant leur capacité à inhiber la croissance de certaines cellules cancéreuses. Les 188 graphes de MUTAG [8] sont des composés nitro aromatiques et hétéroaromatiques dont il faut prédire la mutagénicité. La base ENZYMES [2] contient 600 protéines représentées par des

1. Ces deux jeux de données sont disponibles à : `https://iapr-tc15.greyc.fr/links.html`

TABLE 1 – Caractéristiques des jeux de données utilisés. V et E désignent resp. les ensembles de sommets et d'arêtes des graphes des bases, et V_S et E_S sont les ensembles de sommets et d'arêtes des super-graphes. La ligne Répartition donne la répartition des exemples dans les bases sous la forme #Positifs / #Négatifs.

	NCI1	MUTAG	ENZYMES	PTC	MAO	PAH
Nombre de graphes	4110	188	600	344	68	94
$\|V\|$ moyenne	29,87	17,93	32,63	14,29	18,38	20,70
$\|E\|$ moyenne	32,30	19,79	62,14	14,69	19,63	24,42
Nombre de labels	37	7	3	19	3	1
Nombre de motifs	424	84	240	269	22	4
$\|V_S\|$ moyenne	192,8	42,6	177,1	102,6	25,8	26,8
$\|E_S\|$ moyenne	4665	146	1404	377	44,8	79
Nombre de classes	2	2	6	2	2	2
Répartition	2057 / 2053	125 / 63	–	152 / 192	30 / 38	59 / 35

TABLE 2 – Principales caractéristiques des réseaux testés.

		DCNN	GCN	CGCNN
Poids par :	– composante	✓	✓	✓
	– distance (pas)	✓	–	✓
Convolution : voisinage		Étendu	Direct	Étendu
# Couches de convolution		1	1	2

graphes appartenant à 6 classes d'enzymes (100 par classe) et PTC [19] contient 344 composés dont il faut prédire la propriété cancérigène pour les rats et les souris. La base MAO (Mono Amine Oxydase) contient des graphes moléculaires composés d'hétéro-atomes pouvant potentiellement inhiber la monoamine oxydase (médicament antidépresseur). Enfin, PAH contient des molécules cycliques non labellisées (cycles de carbones) dont il faut prédire le caractère cancérigène.

La base ENZYMES est équilibrée entre les exemples de chaque classe. La répartition des autres jeux de données est indiquée dans la table 1 (Répartition).

4.2 Réseaux considérés

On considère trois réseaux de référence, dont l'un est basé sur une approche spectrale pour la convolution, CGCNN [9], et les deux autres sur des approches spatiales : GCN [13] et DCNN [1] (voir Section 2).

Tous ces réseaux ont été testés en valuant les sommets par le vecteur de fréquences de motifs décrit en Section 3.1. Afin d'évaluer le bénéfice apporté par ces caractéristiques, nous avons également testé GCN et DCNN avec un vecteur canonique représentant le label de l'atome (Section 3.1). CGCNN requiert un graphe d'entrée fixe et ne peut donc traiter les problèmes de classification décrits en Section 4.1 sans utiliser le super-graphe comme couche d'entrée. Inversement, les deux réseaux GCN et DCNN ont été testés avec et sans utiliser le super graphe. Afin de simplifier le protocole, les expériences utilisant le super-graphe sont systématiquement exécutées en employant le vecteur de motif.

Dans nos expériences, les réseaux spatiaux reprennent l'architecture proposée par [1] composée d'une seule couche de convolution. Ce choix permet de quantifier l'apport de nos contributions par rapport aux réseaux originaux. Pour CGCNN, nous utilisons deux couches de convolutions pour bénéficier de la descente en résolution. Pour DCNN, nous fixons un rayon $H = 4$, et pour GCN et CGCNN nous utilisons 32 filtres. L'optimisation se fait avec l'optimiseur Adam [12], et un pas d'apprentissage de évoluant entre 10^{-4} et $5 \cdot 10^{-3}$ suivant les modèles et les jeux de données. Nous utilisons au maximum 500 itérations (une itération correspondant au passage de tous les graphes de la base dans le réseau), avec arrêt prématuré (*early stopping*). Les expérimentations sont réalisées en validation croisée avec 10 échantillons. Dans le cas de l'utilisation d'un super-graphe, ceci requiert de calculer les super-graphes de chaque échantillon d'apprentissage. Dans cette configuration, les données sont augmentées de 20% avec la méthode décrite en section 3.3 (on considère 20% de projections en plus des projections de coût minimal de l'ensemble). Le nombre d'ECGM considéré pour chaque graphe est fixé à 16.

Les résultats obtenus en terme de précision sont donnés à la Table 3. Les principales caractéristiques des réseaux utilisés sont reportées en Table 2.

4.3 Discussion

Comme on peut le voir en Table 3, les vecteurs de caractéristiques que nous proposons (section 3.1, colonne carac. dans le tableau) améliorent la précision dans la plupart des cas. Pour certains jeux de données, le gain est même supérieur à 10%. Remplacer la moyenne globale par la couche de réduction proposée dans l'étape de réduction globale (colonne gpool) permet d'améliorer encore les résultats pour les deux réseaux spatiaux, et sur tous les jeux de données, excepté MAO (le réseau spectral ne présente pas cette étape, car utilise une réduction pyramidale basée sur un super-graphe). Cette base de graphes étant la plus petite (68 molécules), ces moins bonnes performances peuvent être expliquées par le faible rapport entre le nombre de paramètres dans le réseau et la quantité de données d'apprentis-

TABLE 3 – Précision moyenne en validation croisée (10 échantillons) avec les modèles initiaux, les motifs locaux comme carractéristiques (-motifs), et le supergraphe comme couche d'entrée (-SG). Les deux méthodes de mise en commun globale (*global pooling*) sont par *Global Average Pooling* (GAP) et histogramme (hist).

GConv	carac.	s-g	gpool	NCI1	MUTAG	ENZYMES	PTC	MAO	PAH
	–	–	GAP	62.61	66.98	18.10	56.60	51.73	57.18
	✓	–	GAP	67.81	81.74	31.25	59.04	79.06	54.70
DCNN	✓	–	hist	71.47	82.22	38.55	**60.43**	76.40	66.90
	✓	✓	GAP	67.00	81.91	35.40	55.18	**82.37**	63.52
	✓	✓	hist	**73.95**	**83.57**	**40.83**	56.04	68.64	**71.35**
	–	–	GAP	55.44	70.79	16.60	52.17	52.17	63.12
	✓	–	GAP	66.39	82.22	32.36	58.43	82.13	57.80
GCN	✓	–	hist	**74.76**	**82.86**	37.90	**62.78**	75.47	**72.80**
	✓	✓	GAP	65.33	79.66	37.04	59.53	**82.37**	63.53
	✓	✓	hist	73.02	80.44	**46.23**	61.60	68.9	71.50
CGCNN	✓	✓	–	68.36	75.87	33.27	60.78	58.72	63.73

TABLE 4 – Temps de calcul des super-graphes de chaque jeu de données pour un échantillon de validation croisée. 16 threads ont été utilisés sur un processeur AMD Opteron 6282SE (16 cœurs) cadencé à 2,6 GHz.

	MAO	PAH	MUTAG	PTC	ENZYMES	NCI1
unité	s	s	s	min	min	h
t_d	0,8	1,4	5,2	0,28	3,13	7,95
t_{d+s}	13,6	20,2	31,0	1,13	28,43	9,48
t_{d+s+p}	42,8	78,2	182,6	9,80	312	37,0
N_{base}	68	94	188	344	600	4110
N_{train}	31	42	84	155	270	1848

sage disponibles. En effet, la couche de réduction proposée induit plus de paramètres qu'une réduction par moyenne dans la couche entièrement connectée suivante. De plus, en contraste avec PAH (base de taille comparable), MAO présente des vecteurs de caractéristique de dimension plus importante, augmentant ainsi le nombre de paramètres dans le réseau. Ces résultats montrent que l'étape de mise en commun globale est un composant important de ce type de réseaux, et ne doit pas être négligée.

L'utilisation d'un super-graphe comme couche d'entrée (colonne s-g dans la Table 3) nous permet d'étendre le champs d'action des réseaux reposant sur une définition spectrale de la convolution sur graphe à des graphes de topologie différente, ce qui constitue un résultat intéressant en soi. La précision obtenue sur ces tâches de classification est comparable aux autres méthodes (meilleure que les modèles initiaux avec GAP) mais nous gardons à l'esprit qu'il s'agit d'un premier résultat pour ce type de réseaux.

Les tailles des super-graphes, données en Table 1, restent raisonnables par rapport au nombre de graphes. Néanmoins, il faut noter que cette stratégie élargit la taille de chaque donnée à celle du super-graphe, ce qui peut induire des temps d'apprentissage plus longs.

Les temps de calcul des super-graphes sont aussi présentés en Table 4, de manière cumulative par étape (tous les étages de la pyramide confondus). Le nombre de graphes dans la base d'apprentissage est donné par la ligne N_{train}. Les valeurs t_d correspondent aux temps de calcul des matrices de distances d'édition entre graphes, t_{d+s} ajoute le temps de création des super-graphes intermédiaires, et t_{d+s+p} prend en compte les projections des N_{base} graphes de la base de donnée sur le super-graphe. Cette dernière étape est la plus longue car elle concerne beaucoup plus de graphes et qu'un des graphes (le super-graphe) présente une taille plus importante.

5 Conclusion

Nous proposons dans cet article de nouveaux ensembles de caractéristiques permettant d'améliorer les performances de classification des réseaux de convolution sur graphe avec des données symboliques comme des graphes chimiques. Nous proposons aussi l'utilisation d'un super-graphe comme couche d'entrée de ces réseaux de manière à étendre le champ d'action des réseaux sur graphes basés sur une définition spectrale de la convolution à la prédiction de propriétés de graphes de différentes topologies. Ce super-graphe peut être combiné avec n'importe quel réseau de neurones sur graphes et améliore aussi les performances de réseaux spatiaux sur certains jeux de données. Enfin, nous proposons une alternative au goulot d'étranglement par moyenne globale généralement utilisée dans la dernière partie de ce type de réseaux permettant de caractériser la distribution des caractéristiques sur le graphe.

Références

[1] James Atwood and Don Towsley. Diffusion-convolutional neural networks. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, pages 2001–2009, USA, 2016. Curran Associates Inc.

[2] Karsten M. Borgwardt, Cheng Soon Ong, Stefan Schönauer, S. V. N. Vishwanathan, Alex J. Smola, and Hans-Peter Kriegel. Protein function prediction

via graph kernels. *Bioinformatics*, 21(suppl 1) :i47–i56, 2005.

[3] Sébastien Bougleux, Luc Brun, Vincenzo Carletti, Pasquale Foggia, Benoit Gaüzère, and Mario Vento. Graph edit distance as a quadratic assignment problem. *Pattern Recognition Letters*, 87 :38 – 46, 2017. Advances in Graph-based Pattern Recognition.

[4] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst. Geometric deep learning : Going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4) :18–42, July 2017.

[5] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann Lecun. Spectral networks and deep locally connected networks on graphs, 2014.

[6] Horst Bunke, Xiaoyi Jiang, and Abraham Kandel. On the minimum common supergraph of two graphs. *Computing*, 65(1) :13–25, Jul 2000.

[7] Évariste Daller, Sébastien Bougleux, Benoit Gaüzère, and Luc Brun. Approximate Graph Edit Distance by Several Local Searches in Parallel. In *7th International Conference on Pattern Recognition Applications and Methods*, Funchal, Madeira, Portugal, January 2018.

[8] Asim Debnath, Rosa L. Lopez de Compadre, Gouranga Debnath, Alan Shusterman, and Corwin Hansch. Structure-activity relationship of mutagenic aromatic and heteroaromatic nitro compounds. correlation with molecular orbital energies and hydrophobicity. 34 :786–797, 02 1991.

[9] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 3844–3852. Curran Associates, Inc., 2016.

[10] David Duvenaud, Dougal Maclaurin, Jorge Aguilera-Iparraguirre, Rafael Gómez-Bombarelli, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P. Adams. Convolutional networks on graphs for learning molecular fingerprints. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'15, pages 2224–2232, Cambridge, MA, USA, 2015. MIT Press.

[11] Jack Edmonds. Optimum branchings. *Journal of Research of the National Bureau of Standards*, 71B :233–240, Nov 1966.

[12] Diederik P. Kingma and Jimmy Ba. Adam : A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[13] Thomas Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *Proceedings of ICLR'17*, 2017.

[14] Yann LeCun and Yoshua Bengio. The handbook of brain theory and neural networks. chapter Convolutional Networks for Images, Speech, and Time Series, pages 255–258. MIT Press, Cambridge, MA, USA, 1998.

[15] Mathias Niepert, Mohamed Ahmed, and Konstantin Kutzkov. Learning convolutional neural networks for graphs. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, pages 2014–2023. JMLR.org, 2016.

[16] Kaspar Riesen and Horst Bunke. Approximate graph edit distance computation by means of bipartite graph matching. *Image and Vision Computing*, 27(7) :950 – 959, 2009. 7th IAPR-TC15 Workshop on Graph-based Representations (GbR 2007).

[17] Aravind Sankar, Xinyang Zhang, and Kevin Chen-Chuan Chang. Motif-based convolutional neural network on graphs, 2017. ArXiv :1711.05697.

[18] Martin Simonovsky and Nikos Komodakis. Dynamic Edge-Conditioned Filters in Convolutional Neural Networks on Graphs. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017)*, Honolulu, United States, July 2017.

[19] Hannu Toivonen, A Srinivasan, R King, S Kramer, and Christoph Helma. Statistical evaluation of the predictive toxicology challenge. 02 2018.

[20] Nitika Verma, Edmond Boyer, and Jakob Verbeek. Dynamic Filters in Graph Convolutional Networks. working paper or preprint, June 2017.

[21] Nikil Wale, Ian A. Watson, and George Karypis. Comparison of descriptor spaces for chemical compound retrieval and classification. *Knowledge and Information Systems*, 14(3) :347–375, Mar 2008.