Graph Neural Networks

Jian Tang HEC Montreal Mila-Quebec AI Institute Email: jian.tang@hec.ca

Réseaux sociaux

Graphe des interactions entre protéines

Graphe des interactions entre protéines et médicaments

Molécules 泾 $\overline{\mathcal{A}}$ $\mathcal{L}^{\mathcal{A}}$ $\frac{1}{\sqrt{2}}$ ා Au Y. ≫ \sim a_{\sim} any চ ay \sim xx γ ⊲ g \sim π Ò

Quelques applications des graphes

- Recommandations d'amis sur les réseaux sociaux
- Prédiction de l'allégeance politique d'un utilisateur sur Facebook
- Prédiction de la diffusion d'informations sur les réseaux sociaux
- Prédiction du rôle des protéines dans un graphe des interactions entre protéines
- Prédiction des propriétés chimiques d'une molécule
- \bullet Etc.
- Ces applications nécessitent une bonne représentation du graphe!!

Apprentissage de graphes (semi) supervisé

- Au lieu de préserver la structure du graphe, des tâches supervisées sont données:
	- Classification des nœuds,
	- Présence d'une arête.
- Apprendre les représentations des nœuds pour une tâche spécifique

- Type du noeud
- Attribut du noeud

Rappel: Réseaux convolutifs (CNN) pour apprentissage de représentation

- Filtres convolutifs
	- Permet la reconnaissance d'attributs locaux.
	- Différents attributs peuvent être appris en fonction de leur emplacement sur l'image.

Champ récepteur local/ Local Receptive Field pour les graphes lamp recepteur local/
est Desembre Field norm les su

- Comment peut-on définir des local receptive fields pour des graphes?
	- Sous-graphes locaux
- Par contre, il n'y pas d'ordre entre les voisins:
	- Avec une image, les voisins peuvent être ordonnés.

Formalisme

- Soit le graphe $G = (V, E)$, où V est l'ensemble des noeuds et E est l'ensemble des arêtes.
- Deux types d'informations sont présentées:
	- Un vecteur d'attribut $x_i \in R^D$ pour chaque noeud v_i . L'ensemble des attributs pour V peut être représenté dans une matrice des attributs X de dimension $N\times D.$
	- La structure du graphe, généralement définie sous la forme d'une matrice adjacente A où \overline{A}_{ij} est le poids associé à l'arête (i, j).
- But: obtenir une représentation des noeuds, définie par H (de dimension $N \times \dot{F}$, où F est la dimension de chaque représentation).

Réseaux de neurones de graphe (formalisme)

- Réseaux de neurones de graphes (à plusieurs couches):
	- H^0 = X, la matrice des attributs des noeuds
	- De façon itérative, mettre à jour la représentation des noeuds
- \bullet La $k^{\mathsf{i\hat{e}me}}$ couche cachée du réseau de neurones est la $k^{\mathsf{i\hat{e}me}}$ représentation des noeuds, laquelle est symbolisée par H^k .
- $\boldsymbol{\cdot}$ Soit H^L la dernière représentation:
	- Peut être utilisée pour tâches spécifiques (classification de noeuds)

Apprentissage supervisé

- Apprentissage d'un classificateur, f, à l'aide de la représentation finale H^L .
- Fonction de perte est de la forme:

$$
O = \sum_{i \in \text{exemples libells}} loss(f(H_i^L), y_i)
$$
\nEntrée

\n

Couche cachée

Couche cachée

Comment mettre à jour la représentation des noeuds?

- Pour chaque couche d'un GNN et pour chaque noeud: Pour chaque couche d'un GNN et pour chaque no
	- AGREGER l'information associée aux voisins d'un noeud, **AGDEGED** l'information accoción aux voicine
	- COMBINER cette information à celle du noeud d'intérêt. rente de l'information accocide dant reigne a allitice de l'intérât

Réseaux convolutifs de graphes**

**Kipf et al. 2017. Semi-supervised Classification with Graph Convolutional Networks.

Réseaux convolutifs de graphes Graph Convolutional Networks Graph Convolutional Side

- Soit A, la matrice adjacente
- On pose:

$$
\hat{A} = A + I
$$

\n
$$
h_i^k = \sigma \left(\sum_{j \in \{N(i) \cup i\}} \frac{\hat{a}_{ij}}{\sqrt{d_i d_j}} W^k h_j^{k-1} \right)
$$

\n
$$
h_i^k = \sigma \left(\sum_{i \in N(i)} \frac{a_{ij}}{\sqrt{d_i d_j}} W^k h_j^{k-1} + \frac{1}{d_i} W^k h_i^{k-1} \right)
$$

Graphe de computation

• Deux couches de GCN • Deux coucries d

Figure from Ying et al. 2018

Images tirées de Ying et al. 2018

Peut-on changer les poids des arêtes?

• Pour les GCN, l'influence d'un noeud j sur le noeud i est déterminée en fonction du poids de leur arête (i,j) de même que de leur degré respectif:

$$
\frac{a_{ij}}{\sqrt{d_i d_j}}
$$

- Par contre,
	- Les arêtes peuvent contenir beaucoup de bruit
	- Peut ne pas être optimal pour certaines tâches

 Graph Attention Networks (Veličković et al. 2017)

Graph Attention Networks (GAT) We will start by describing a single *graph attentional layer*, as the sole layer utilized throughout all of the GAT architectures used in our experiments. The particular attentional setup utilized by us closely follows the work of Bahadanau et al. (2015) tures, at least one learnable linear transformation is required. To that end, as an initial step, and as an initial step, and as an initial step, as an initial step, as an initial step, as an initial step, a shared of the **Linear Transformation, parametrized by a** *W* **2 RF** 0 G 2 RF which is the normal mechanism $\sum_{i=1}^{n}$ on the nodes—a shared at $\sum_{i=1}^{n}$ of $\sum_{i=1}^{n}$ number of nodes, and *F* is the number of features in each node. The layer produces a new set of node **Graph Attention Networks (GA** In order to obtain sufficient expressive power to transform the input features into higher-level fea-

- Un mécanisme d'attention est utilisé afin d'apprendre les poids des arêtes **cartinality of potential in the structural information** $\frac{1}{\text{Published as}}$ Published as a conference paper at ICLR 2018 The input to our layer is a set of node features, ^h ⁼ *{*[~] *h*1*,*~ *h*2*,...,*~ *^h^N },*[~] *^hⁱ* ² ^R*^F* , where *^N* is the • Un mecanisme d'attention est utilise afin d'apprendre les $\frac{1 \text{ unbiased as a}}{1 \text{ times the area}}$ graph structure into the mechanism by performing *masked attention*—we only compute *eij* for nodes a Linear transformation, parameterized by a property of the matrix of the matrix of \overline{a} , \overline{b} **then perform interform in the cantrical metallic condensation in the non-particle of as a set of problem and** $\sum_{\text{Published as a}}$
- Requête: noeud actuel expressive power to transform the input features into higher-level features into higher*f* 2 *Nequete: Noeud* actuer *eij* = *a*(W~
- Mémoire: voisins (incluant le noeud actuel). linear transformation, parametrized by a *weight matrix*, ^W ² ^R*^F* ⁰ be exactly the first-order neighbors of *i* (including *i*). To make coefficients easily comparable across momon of volent θ moldant to hoodd dolder_/. that indicate the *importance* of node *j*'s features to node *i*. In its most general formulation, the model allows every node to attend on every other node, *dropping all structural information*. We inject the
- L'attention entre les noeuds i et j se calcule ainsi: of the methanism structure into the methanism by performing and *mattention* entry less nonly contained and *i* **L** allottion only 100 noded *i* of j oo daldalo all ion.

$$
e_{ij}=a(\mathbf{W}\vec{h}_i,\mathbf{W}\vec{h}_j)
$$

$$
\alpha_{ij} = \text{softmax}_j(e_{ij}) = \frac{\exp(e_{ij})}{\sum_{k \in \mathcal{N}_i} \exp(e_{ik})}.
$$

$$
\alpha_{ij} = \frac{\exp\left(\text{LeakyReLU}\left(\vec{\mathbf{a}}^T[\mathbf{W}\vec{h}_i\|\mathbf{W}\vec{h}_j]\right)\right)}{\sum_{k \in \mathcal{N}_i}\exp\left(\text{LeakyReLU}\left(\vec{\mathbf{a}}^T[\mathbf{W}\vec{h}_i\|\mathbf{W}\vec{h}_k]\right)\right)}
$$

 \overline{a} \overline{b} \overline{c}

~ *h*0 @^X *ⁱ* =

 $\sqrt{ }$

↵*ij*W~ *hj* 1

A *.* (4)

 $\limsup_{n\to\infty} \frac{1}{n^2}$ and $\limsup_{n\to\infty} \frac{1}{n^2}$ en aleman (with the preaty) by note 1 on hy heremoticon; princem any alleges and The Attites mathias in when the AM and the Lease our incomplete and the correct An interinsical \mathbf{B} dependem \mathbf{u} the information the information of the neighbors with a contribution of the neighbors \vec{h}' . $\mathbf{W}h_j$ **a** $\frac{h_5}{\sqrt{h_5}}$ *h^j*) employed by our model, parametrized by Λ + + a p + i **head of the attention method in** \mathbf{p} **(With Alice in Membry of Sy our model Aarametrized** by a weight vector $\vec{a} \in \mathbb{R}^{2F}$, applying a LeakyReLU activation. Right: An illustration of multiby a weight vector $\vec{a} \in \mathbb{R}^{2F}$, applying a LeakyReLU activation. **Right:** An illustration of multi-
head attention (with $K = 3$ heads) by node 1 on its neighborhood. Different arrow styles and alculated a nonlinearity of the column η \mathbf{F} igure 1: \mathbf{F} The attention mechanism $a(\mathbf{W})$ $\mathop{\mathrm{codim}}\nolimits\mathop{\mathrm{codim}}\nolimits\mathop{\mathrm{codim}}\nolimits\mathop{\mathrm{codim}}\nolimits\mathop{\mathrm{codim}}\nolimits\mathop{\mathrm{codim}}\nolimits\mathop{\mathrm{codim}}\nolimits\mathop{\mathrm{codim}}\nolimits\mathop{\mathrm{codim}}\nolimits\mathop{\mathrm{codim}}\nolimits\mathop{\mathrm{codim}}\nolimits\mathop{\mathrm{codim}}\nolimits\mathop{\mathrm{codim}}\nolimits\mathop{\mathrm{codim}}\nolimits\mathop{\mathrm{codim}}\nolimits\mathop{\mathrm{codim}}\nolimits\mathop{\mathrm{codim}}\nol$ concatenated or averaged to obtain \vec{h}'_1 .

 \sim .

 \vec{h}_i W \vec{h}_j

 $\mathbf{W}\vec{h}_i$

hⁱ W~

applying a nonlinearity,
$$
\sigma
$$
):
\n
$$
\vec{h}'_i = \sigma \left(\sum_{j \in \mathcal{N}_i} \alpha_{ij} \mathbf{W} \vec{h}_j \right)
$$
\n
$$
\vec{h}'_i = \sigma \left(\sum_{j \in \mathcal{N}_i} \alpha_{ij} \mathbf{W} \vec{h}_j \right).
$$
\n(4)

• Notez que chaque nœud peut se connecter à lui-même : softmax*j* \vec{a} \vec{b} *h*1 ploy *multi-head attention* to be beneficial, similarly to Vaswani et al. (2017). Specifically, *K* inde- \mathbb{Z} e following output, feature representation: \bar{h} h_1' vg ple a put child we flet attention, we have found extending our mechanism to ent. ITE . pendent attention mechanisms execute the transformation of Equation 4, and then their features are $\overbrace{\vec{h}'_1}^{\text{vg}}$ concatenated, resulting in the following output feature representation: $\vec{\alpha}_{11}$ To stabilize the learning process of self-attention, we have found extending our mechanism to em-

$$
\sqrt{\vec{h}_i'} = \sqrt{\vec{h}_i'} \frac{K}{k} \underbrace{\left(\vec{h}_3 \mathbf{G} + \frac{\mathbf{W}^k}{\mathbf{W}^k \mathbf{G}} \mathbf{W}^k \mathbf{H}_j\right)}_{\mathbf{W}^k \mathbf{W}^k} \underbrace{\mathbf{W}^k \mathbf{H}_j}_{\mathbf{W}^k \mathbf{W}^k} \underbrace{\vec{h}_1}_{\mathbf{W}^k} \tag{5}
$$

 \vec{h}_5

Figured **Refle**: is the corresponding input linear transform^{anetrized} by a weight vector and corresponding input initial transformation.
https://weight.org/weight/2011.com/id=0.5.7.7.7.1.6.externation the mode returned output, h will consist of K F^\prime features , In this setting, the different returned output, \mathbf{n} , will consist of ΛF realisties (rather than F) colors denote independent attention computations. The aggregated features from each head are consecuted or expr node. concatenated or averaged to obtain \vec{h}'_1 . represents concatenation, α_{ij}^k are not individual attention coefficients computed by the k-th mechanism $(a$ attention mechanism (a^k) , Figure 1 W^{ork} is the corresponding input linear transformation's weight matrix. where \parallel represents concatenation, α_{ij}^k are normalized attention coefficients computed by the *k*-th ϵ ^oncatemente σ , and σ and σ are σ and σ . Note that, in this setting, the final returned output, h^{\vee} will consist of KF' features (rather than F') for each node.

Attentions multiples (Multi-head attention) Attentions multip To stabilize the learning process of self-attention, we have found extending our mechanism to employ *multi-head attention* to be beneficial, similarly to Vaswani et al. (2017). Specifically, *K* indepersonation mechanisms formation in the transformation of \sim

- · De façon analogue à l'attention multiple dans les modèles de transformers, l'attention multiple peut être mise à profit pour *l*'apprentissage de graphe. *h* analogue a rattention multiple dans les
rmers, l'attention multiple peut être mise à concatenated, resulting in the following output feature representation: ralogue a Fallen
- · Cette représentation peut concatener, ou simplement faire une moyenne, des différents mécanismes d'attention. · Cette représentation peut concatener, ou simplen concatenated, resulting in the following output feature representation: attention mechanism (*a^k*), and W*^k* is the corresponding input linear transformation's weight matrix. moyonno, aco amoronto modamonto a allonton.

$$
\vec{h}_i^\prime=\mathop{\parallel}\limits^K_{k=1}\sigma\left(\sum_{j\in{\cal N}_i}\alpha_{ij}^k\mathbf{W}^k\vec{h}_j\right)_{\left(\begin{matrix}\vec{h}_2\\ \vec{h}_3\end{matrix}\right)\text{concally}}\left(\begin{matrix}\vec{h}_2\\ \vec{h}_3\end{matrix}\right)_{\left(\begin{matrix}\vec{h}_1\\ \vec{h}_2\end{matrix}\right)\text{concally}}\left(\begin{matrix}\vec{h}_1\\ \vec{h}_1\end{matrix}\right)_{\left(\begin{matrix}\vec{h}_2\\ \vec{h}_3\end{matrix}\right)\text{concally}}\left(\begin{matrix}\vec{h}_1\\ \vec{h}_2\end{matrix}\right)_{\left(\begin{matrix}\vec{h}_1\\ \vec{h}_2\end{matrix}\right)\text{concally}}\left(\begin{matrix}\vec{h}_1\\ \vec{h}_2\end{matrix}\right)_{\left(\begin{matrix}\vec{h}_2\\ \vec{h}_3\end{matrix}\right)\text{concally}}\left(\begin{matrix}\vec{h}_1\\ \vec{h}_2\end{matrix}\right)_{\left(\begin{matrix}\vec{h}_1\\ \vec{h}_2\end{matrix}\right)}
$$

Quelques problèmes (en pratique)

- Certains noeuds possèdent beaucoup de voisins
- Randomly sample a fixed number of neighbors in each iteration of SGD (Hamilton et al. 2017).

On peut échantillonner de façon aléatoire un nombre fixe de voisins pour chaque itération.

Image tirée de Wang et al. (201

Réseaux de neurones avec propagation de message

Gilmer et al. (2017). Neural Message Passing for Quantum Chemistry.

Réseaux de neurones avec propagation du message (MPNN)

- Tout graphe de réseaux de neurones peut être formalisé à l'aide du concept de propagation de message neural (neural message passing)
	- Le message (sous forme de vecteurs) est propagé de façon itérative à travers les noeuds du graphe
- Deux fonctions
	- Fonction de message
	- Fonction de la mise à jour du noeud

Gilmer et al. (2017). Neural Message Passing for Quantum Chemistry.

Phase de propriet de la propie de la partie de la pa <u>ni</u> we reinted in terms of inessage renctions $w_t e^{x}$ is recurrent only incorporating and et al. (2014). 1, and incorporation is w $3e$ v and y awea dated based en messages m_{w}^{t+1} according Dand vertex undate functions Alexandri Mostageneassing forsQuantumeChemistrytying, so the same up time steps and is defined in terms of message functions M_t Recurrent Unit introduced in Cho *w*²N(wcmdinig to w) each time step twhere lighed j are, no v_v^{t+1} actor(*i*v) v_v^{t+1} $m_v = \sum_{l} M_t(n_v, n_w, e_{vw})$ (1) Wh \mathbb{G}_{ν}^{++1} = in the sum \mathbb{G}_{ν}^{++1} of \mathbb{G}_{ν}^{++1} and \mathbb{G}_{ν}^{++1} for \mathbb{G}_{ν}^{++1 G. The readout phase agentunction feature vector apply g **sernex equatie function** R according to **a** Networks. Battaglia at al. (2016).
lenotes the neighbors of v in graph e readout phase agniputed *du* feature vector for thes work eranh using some readout function *R* according fet at each
It **permessegratures fonsava**eved conponinting fons U_t , and readoutefugadion *Rua*cial Ragardifferentiable **Evel** targe permutations of the node states and must be invariant to be node in order to be R operates on the set of node states and must be invariant to α or $d\overline{\sigma}$ for α is η in π or \mathcal{U} is a substant in $d\overline{\sigma}$ or \mathcal{U} is π or η is a substant in \mathcal{U} is a substant in π *r* step *t*. Fir
moduced i *r*_d_e ⇣ *i*(*y*, *t d*, *(* \mathbf{R} is \mathbf{R} in the graph are chemistry time step t . Finally, $\sum_{I} \left(\frac{1}{\mathcal{N}(\mathcal{A})} \mathcal{U}(T) \right)_{I=0}^{I} \left(\sum_{i} \left(\frac{1}{I} \right) \right)$ \mathcal{N} interaction inetworks. Battagua m_v^{t+1} where in the sum W^t ψ denotes the neighbors of hence and W^{loc} and heuran heropsidered both the scale level the track of the consideration of the consideration of the case of the case of the case of the case of the α etron ivetworks, **Battagira et al.**
level target. It also considered t (*hv, xv, mv*) where *x^v* is an external vector representing some out out the vertex *visite district* tion *M*(*hv, hw, evw*) is a neural network which takes the conceated to the vertex where the vertex update function of the vertex update function of the vertex update function $\frac{1}{2}$ *U*(*hv, xv, mv*) is a neural network which takes as input the concatenation (*hv, xv, mv*). Finally, in the case where v G . The read out phase egenture ion feature vectal entirely at each node in the graph, and G . α is the step and attentions U_t During the message passeure the distribution of (T_t) and α and (T_t) d vertex update Randen states by at bach hode in the example area were time step the Finally update function is σ base, hidden states h_v at each mode in the graph are w) each time step *t*. This ally $\begin{array}{c} \epsilon^{u+1} \end{array}$ according to h_v^t whele graph using semiex endert function R according to the Retworks, Battaglia et al. (2016).
Ein the sum $N(v)$ denotes the neighbors of v in graph where in the sum, *N*(*v*) denotes the neighbors of *v* in graph The Yeadout pearls ing some readout aunction *Feaccord* ole graph usineadoutefunctions Reparticula **Reapport differe** $\hat{y} = R(f)$ and $\hat{y} = R(f)$ and $\hat{y} = R(f)$ and $\hat{y} = R(f)$. $R = \sum_{n=1}^{\infty}$ \dot{v} Th σ µ
∤ $\hat{u}^{(n)}(x^T), h^0(y)$ l
/ $\begin{array}{c} \mathbf{\langle}\mathbf{\cdot}\mathbf{)} \ \mathbf{k} \end{array}$ $\sqrt{2}$ $j(h_x^{(T)})$ $\left(\eta_{k}^{(T)}, h_{k}^{(0)}\right) \odot \left(\eta_{k}^{(T)}\right)$ G. The readout phase of <u>Fill a</u> fill a feature we for the work considered both the ease where refere whole graph using some readout function *R* according get at each node in the graph, and where there is level target. It also considered the case where the case of the node level effects applied at each time step, if case the update function takes as imput the conce $\frac{1}{\text{m}}$ ing phase, moden states that Message Passing for Oraph are chemistry time step t. Finally,
time steppdared states drohimessages michaele Passing for Orantum Chemistry time step t. Finally, $m_v^{t+1} = \sum_{l}^{n_v} \sum_{l}^{n_v} \frac{W_l(n_v^n, m_v^{t-1})}{M_l(n_v^n, h_w^n, e_{vw})}$ (1) where in the sum, $\tilde{N}(v)$ denotes the neighbors of *v* in graph *y***<u>adiont</u> //www.eduple.com/differentiable.html**
 $y = R(\gamma h_v \mid w \in G)$. Γ he message functions M_{σ} vantex update functions U_{σ} $R = \sum_{k=0}^{R} a_k$ ⇣ $\left(\begin{matrix} h(x), h(y) \\ h(x), h(y) \end{matrix} \right)$ \mathfrak{p} \odot $\frac{1}{2}$ which is the condition of $\frac{1}{2}$ and $\frac{1}{2}$ and $\frac{1}{2}$ are neural networks, *j*(*h*(*T*) where *i* and *j* are neural networks, and \odot n the sum, $N(v)$ denotes the neighbors of v in graph
e readout phase of $\overline{R}(\{h^T\}_{u\in G})$ for the work considerate inditie imaginally here level target, the also where edithe case niche levelae footsidappligdage oach stract case the update function takes as imputed the (*hv, xv, mv*) where *x^v* is an external vector representing some outside influence on the vertex *v*. The message funcupdated based on messages *m^t*+1 *^v* according to h_v^t = $V_t(h_v^t, m_v^{t+1})$ (2) where in the sum, *NG (0)* denotes the neighbors of *v* here *raft* ${\rm w}$ h $\rm \rm dd$ gr $({\rm zp})$ n using **somex eadate function** R according to The message functions *Mt*, vertex update functions *Ut*, and readoutefunadion *Run*ction Reagandifferentiable functions. R operates on the set of node states and must be invariant to p eodo $\overline{\sigma}$ formit the note that $\overline{\psi}$ when the state single state in $\overline{\psi}$ each time step *t*. Finally, \sum_{i} \sum_{k} \sum_{i} \sum_{j} \sum_{k} \sum_{k wise multiplication. Interaction Networks, Battaglia \mathbf{a} in the sufficial of \mathcal{W} \mathcal{W} \mathcal{W} algebra \mathcal{W} and \mathcal{W} are integrated by the second \mathcal{W} $G.$ The readolf phase agency feature vector multiplication Networks, Battaglia et al. (2) level target. It also considered the node brever lette at hat til de de la ver (*hv, xv, mv*) where *x^v* is an external vector representing syghe ffortsdappligd ut on the war tion *M*(*hv, hw, evw*) is a neural network which takes the concatemations (*h***s**, *h*pexternal Mo v evel target. It also considered t
Graph using **semex cadate function** R according to **Example Research and Called** l vertexsupdate runden states burnig the message passage passage passage to the example of the same update function is phase, indicient states h_v^m at each hole in the graph are w each time step twiting livand j are neural networks. $\mathcal{W}^{u\cdots u}_{w}$ according to v^{-1} $m_{v}t+1 - h_{v}^{t+1} = V_{f}(h_{v}^{t}, \eta_{v}^{t})$ where in the sum of \mathbf{u} and \mathbf{u} in the sum of \mathbf{u} in graph \mathbf{u} in \mathbf{v} is a sum of \mathbf{u} in graph \mathbf{u} ole graph usi**readoute negotions** *from* **the and learned different** R operates on the set of node states and my each time step *time* step *ti rese vand j* ⇣ *i***(***p***)**
*v***eaneural ne** չե $\frac{1}{\sqrt{2}}$ ⇣ *j*(*h*(*T*) *^v*) **a** whole graph using some readout function *Re*according get at each nodes in the graph, and where there is
Teadout permessage funcs for sature vector fourthuncions U_t , and a Javab affectional adopt anothy time evel target. It also considered the case where is lode . Ie ver etteenst applied "at each" time step," If
invariant to rase the lipdate rumetion takes as miput time cones Lessage Passing for Graph are Chemistry lime step ind vertex undate functions *U_{le}s During the message pass-*current Child introduced in Che stude (bodate th r α *din*g ext α^{t+1}_v) the sum, *N(v)* denotes deminited avteature vector for thes aph using adoutefugadion *Reacce in Reaccorditive* tontiable¹⁸ *x*¹ and the set of mode states and must be miva α = θ = θ *v***22** $\sum_{i=1}^n$ ષ
' $\left(\begin{matrix}0\\x\end{matrix}\right)\right)\oplus\left(\begin{matrix}j\end{matrix}\right)$ /
, ($\frac{1}{2}$ ⇣ tat at eaglend deun date fuan honnt where is a tarevel target, the graph, and where there is lode level togsådered the case where t hase the inpolate Aunotion takes as upput it care canceles the update of the update the update conce

Comment procéder pour apprenc représentation du graphe complè What if we want to learn the representations of in direction duries • Learn the representations of molecular graphs and vertex update functions *Ut*. During the message pass**ing proceder pour apprence in the graph area in the graduate in the graph area in the graduate** α updated based on messages *m^t*+1 *^v* according to

- Apprentissage de représentation pour un graphe *n*^{*x*} *mt<i>x***** *pour* un graphe
	- **Afin de prédire les propriétés d'une molécule**
- Ajouter une fonction de lecture R, laquelle considère la dernière phase computer computer a feature representations de representations de la compute de la computer de la compute de la compute de la compute de la compute where in the sum, *N*(*v*) denotes the neighbors of *v* in graph

 $\mathsf{R}\colon \mathsf{fonction}$ $\hat{y} = R(\lbrace h_v^T \mid v \in G \rbrace)$ **R: fonction**

- \cdot \hat{y} est la représentation complète du graphe st la representation complete du graphe réseantation complète du graphe *R* operation complete ad graphic
- R peut être une fonction très simple (somme, moyenr

Br

N

N NH

N

N

N

NH

N

NH2

N

N

 O H NH \sim \sim \sim \sim

NH

N N

(*hv, xv, mv*) where *x^v* is an external vector representing

some outside influence on the vertex *v*. The message func-

tion *M*(*hv, hw, evw*) is a neural network which takes the

concatenation (*hv, hw, evw*). The vertex update function

 $\mathbf{L}^{\mathbf{A}}$

the concatenation (*hv, xv, mv*). Finally, in the case where

where *i* and *j* are neural networks, and denotes element-

 $^\mathrm{Cl}$ NH \longrightarrow

Applications Système de recommandations**

- Prédire les items les plus pertinents pour un utilisateur
	- Graphe usager-item ou encore item-item

**Qu et al. An End-to-End Neighborhood-based Interaction Model for Knowledge-enhanced Recommendation.

Applications Compréhension du langage naturel (NLP)

- Étiquetage de rôles sémantiques (*Semantic Role Labeling*)
	- Encode les phrases à l'aide de GCN

Figure 1: An example sentence annotated with semantic (top) and syntactic dependencies (bottom).

Applications Découverte de médicaments

- Réorientation de médicaments
	- Graphe protéines médicaments maladie
- Prédiction des propriétés d'une molécule

Images tirées de Zeng et al. 2019

Applications Optimisation combinatoire

• Problème du commis voyageur

Joshi et al. An Efficient Graph Convolutional Network Technique for the Travelling Salesman Problem.

Applications **Transports**

- Prédiction du trafic:
	- La carte routière comme un graphe

Yu et al. Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting. $\overline{1}$

Applications Réseaux sociaux

- Prédiction d'influences
	- Prédit le statut d'un usager en fonction de ses voisins ou «ami.e.s»

Qiu et al. DeepInf: Social Influence Prediction with Deep Learn

Quelques implémentations

- PyTorch Geometric: [https://pytorchgeometric.readthedocs.io/en/](https://pytorch-geometric.readthedocs.io/en/latest/) [latest/](https://pytorch-geometric.readthedocs.io/en/latest/)
- Deep Graph Learning:<https://www.dgl.ai/>

Exemple: GCN (Pytorch Geometric)

gcr

• [https://github.com/rusty1s/pytorch_geometric/blob/master/examples/](https://github.com/rusty1s/pytorch_geometric/blob/master/examples/gcn.py)

Merci!