

IFT 615 – Intelligence Artificielle

Hiver 2022

Application – Traitement du langage naturelle :
***world embedding* et étiquetage syntaxique**

Professeur: Froduald Kabanza

Assistants: Maxime Alvarez, D'Jeff Nkashama & Jean-Charles Verdier

Motivation

- Le langage humain est une des caractéristiques importantes de l'intelligence humaine par rapport aux autres animaux.
- Plusieurs applications
 - ◆ Traduction automatique
 - ◆ Interaction humain-machine : Siri, robots
 - ◆ Service à la clientèle
 - ◆ Automatisation des processus (*robotic process automation*)
 - ◆ Classification de documents
 - ◆ Compréhension du texte
 - ◆ Cybersécurité – écoute électronique; détection de menaces
 - ◆ Et beaucoup d'autres

Apprentissage profond et traitement du langage naturel

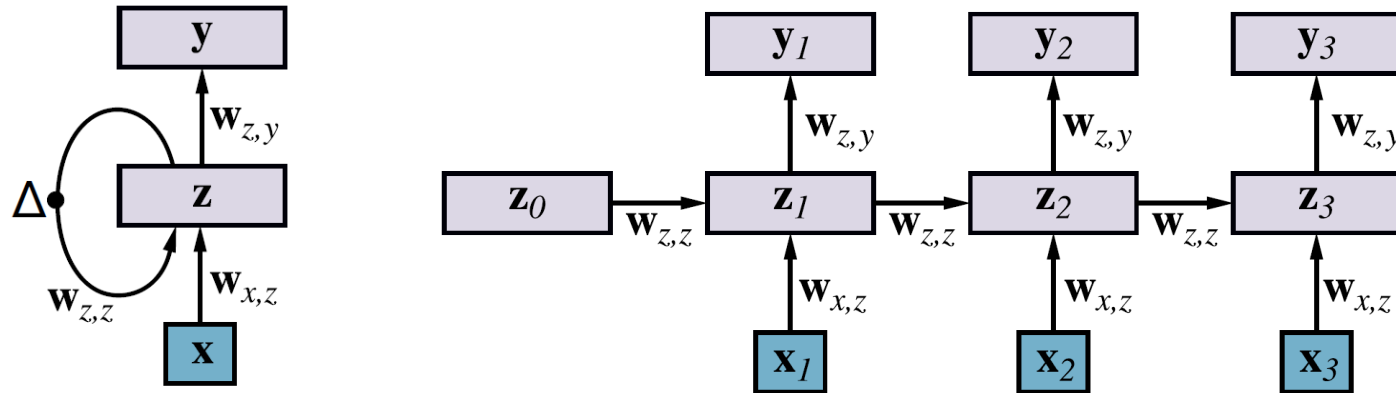
- Un pipeline traditionnel du traitement du langage naturel implique des spécifications et des éditions de connaissances expertes de façon manuelle en plus des algorithmes d'inférence automatiques (notamment probabiliste).
- Un pipeline de traitement du langage naturel par un apprentissage profond de bout-en-bout, représente le traitement comme une seule fonction (approximée par un réseau de neurone).
- Aujourd'hui biens des tâches du traitement du langage naturel sont effectuées par des réseaux de neurones profonds de bout-en-bout.

Sujets couverts

- Réseau de neurones récurrent (RNN)
- Représentation des mots par des *Word embeddings*
- Application à l'étiquetage grammaticale

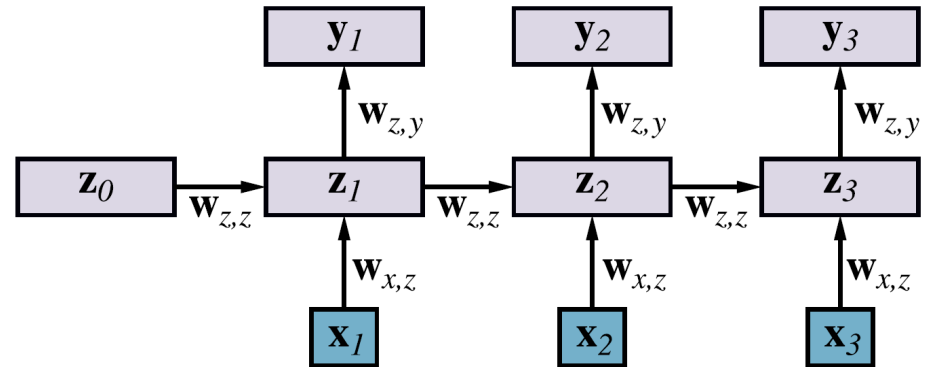
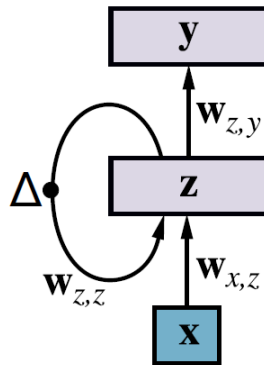
Réseau de neurone récurrent

- En anglais: *Recursive Neural Network* (RNN)



- z : couche (état) cache
- Δ est un délai
- **Hypothèse Markovienne**: l'état cache z_t suffit pour représenter l'information de toutes les entrées précédentes
- Notons $z_t = f_w(z_{t-1}, x_t)$, f_w étant la fonction paramétrique apprise par le réseau
- On peut montrer que f_w est un **processus homogène en temps**. C.-à-d., les lois de changement sous-jacents à f_w sont invariables.

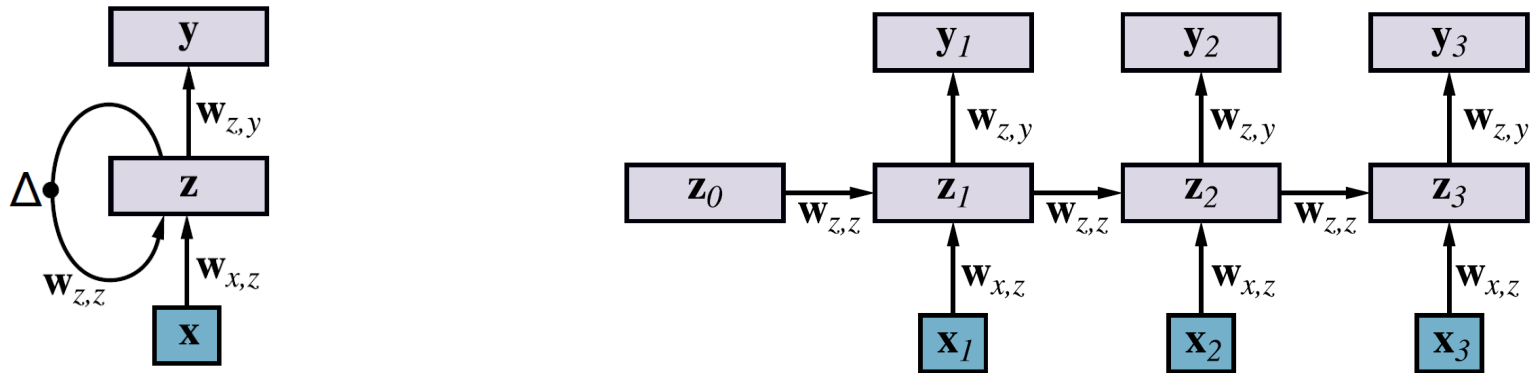
Entrainement d'un RNN



- $z_t = f_w(z_{t-1}, x_t) = g_z(W_{z,z}z_{t-1} + W_{x,z}x_t) \equiv g_z(in_{z,t})$
- $y_t = g_y(W_{z,y}z_t) \equiv g_y(in_{y,t})$
- Étant donné une sequence de vecteurs d'entrée x_1, \dots, x_T et une sequence de sortie y_1, \dots, y_T , on peut dérouler le RNN en un un reseau feedforward
- On peut alors calculer les gradients pour entrainer les poids.
- Par exemple (voir Section 21.6.1 du livre):

$$\frac{\partial z_t}{\partial W_{z,z}} = g'_z(in_{z,t}) (z_{t-1} + w_{z,z} \frac{\partial z_{t-1}}{\partial W_{z,z}})$$

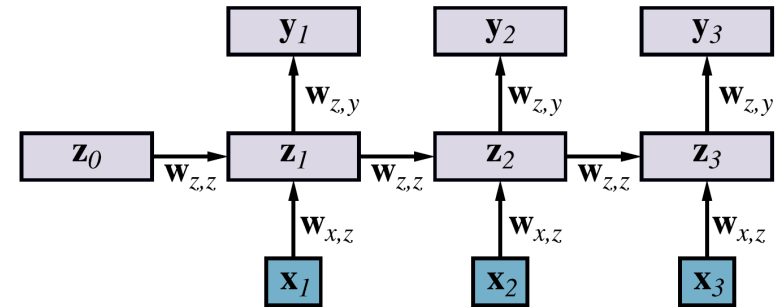
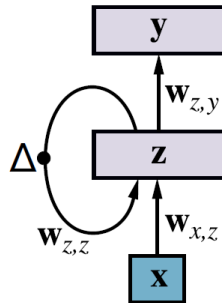
Entrainement d'un RNN



- Pour cet exemple, le gradient pour la couche interne (voir Sec. 21.6.1):

$$\frac{\partial z_t}{\partial w_{z,z}} = g'_z(\text{in}_{z,t}) (z_{t-1} + w_{z,z} \frac{\partial z_{t-1}}{\partial w_{z,z}})$$
- On peut aussi calculer les gradients pour les poids de la couche de sortie $w_{z,y}$ et la couche d'entrée $w_{x,z}$
- Cela conduit à l'algorithme de **propagation des gradients à travers le temps**

Limites du RNN



- Pour cet exemple, le gradient pour la couche interne (voir Sec. 21.6.1):

$$\frac{\partial z_t}{\partial W_{z,z}} = g'_z(\text{in}_{z,t}) (z_{t-1} + w_{z,z} \frac{\partial z_{t-1}}{\partial W_{z,z}})$$
- On peut constater aussi que au temps T les gradients contiendront des termes proportionnels à $w_{z,z} \prod_{t=1}^T g'_z(\text{in}_{z,t})$
- Pour *sigmoïde*, *tahn* et *ReLu*, on a $g' \leq 1$. Le RNN va subir le problème de l'**evanescence du gradient** (*vanishing gradient*)
- D'autre part, si $w_{z,z} > 1$, on pourrait avoir un **problème d'explosion de gradients**.

Word Embedding

- Les réseaux de neurones prennent des vecteurs comme entrées
- Pour le traitement du langage naturelle, on voudrait une représentation des mots tels que les mots reliés ont une représentation proche l'une de l'autre
 - ◆ Apparentés syntaxiquement (ex. « idéal » et « pertinent » sont des adjectifs)
 - ◆ Apparentés sémantiquement (ex. « chat » et « lion » sont des félins)
 - ◆ Réfèrent au même sujet (ex. « soleil » et « pluie » réfèrent au climat)
 - ◆ Reliés sentimentalement (ex. « sublime » et « mauvais » indiquent des sentiments opposés)
- Un « *word embedding* » est un vecteur représentant un mot. La représentation est telle que les mots apparentés ont des vecteurs proches.

Word Embedding

- Un « *word embedding* » est un vecteur représentant un mot. La représentation est telle que les mots apparentés ont des vecteurs proches.
- Un *word embedding* est appris par un réseau de neurones sur un corpus.
- Il existe des *word embeddings* pré-entraînés: Word2Vec, GloVe, FASTTEXT
- Chaque *word embedding* est juste un vecteur de valeurs numériques sans apparente signification

« aaddrvark » = [-0.7, +0.2, -3.2, ...]

« abbacus » = [+0.5, +0.9, -1.3, ...]

...

« zyzyzyva » = [-0.1, +0.8, -0.4, ...]

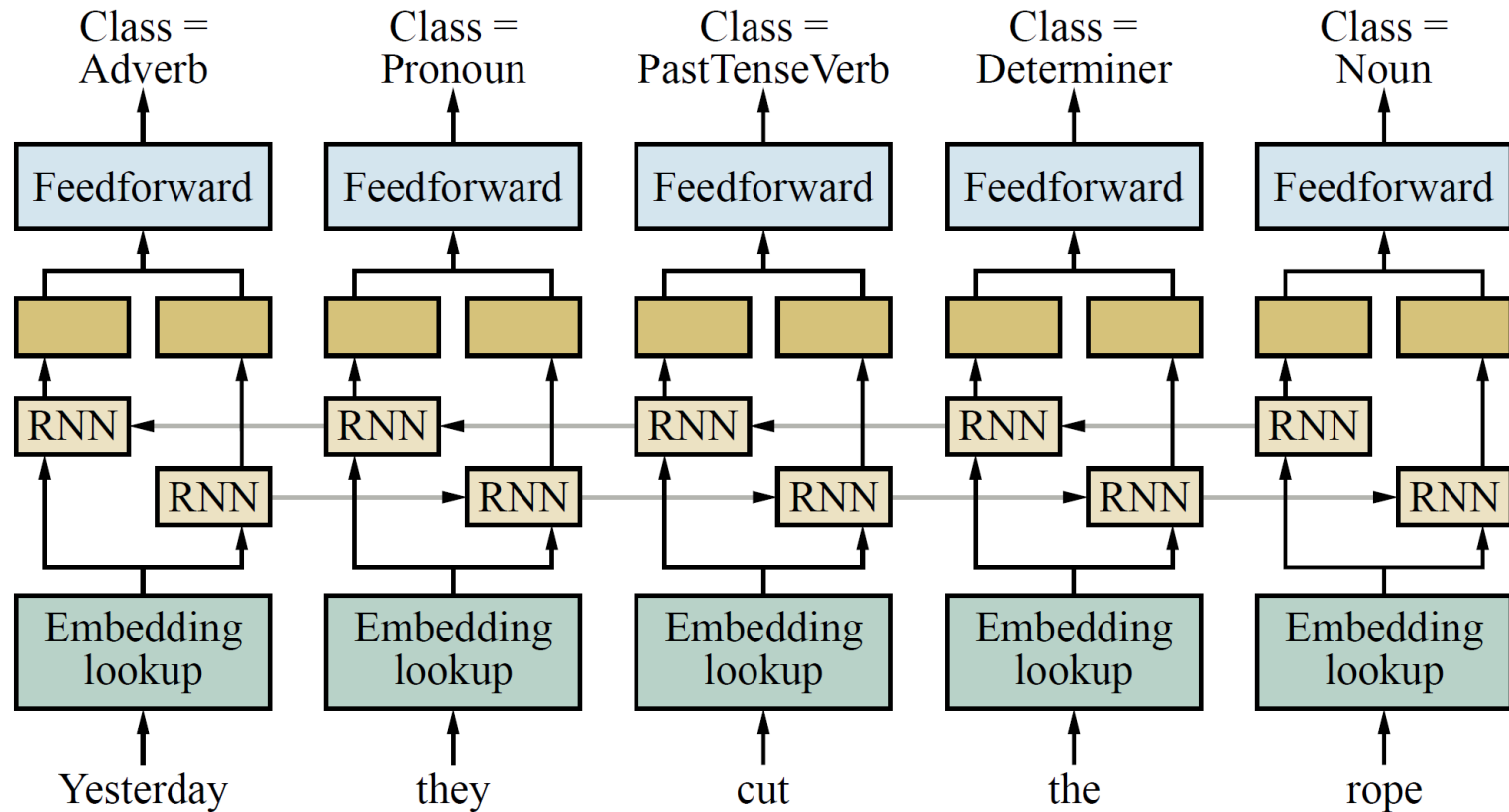
- Mais les mots apparentés ont des représentations proches



Étiquetage grammatical

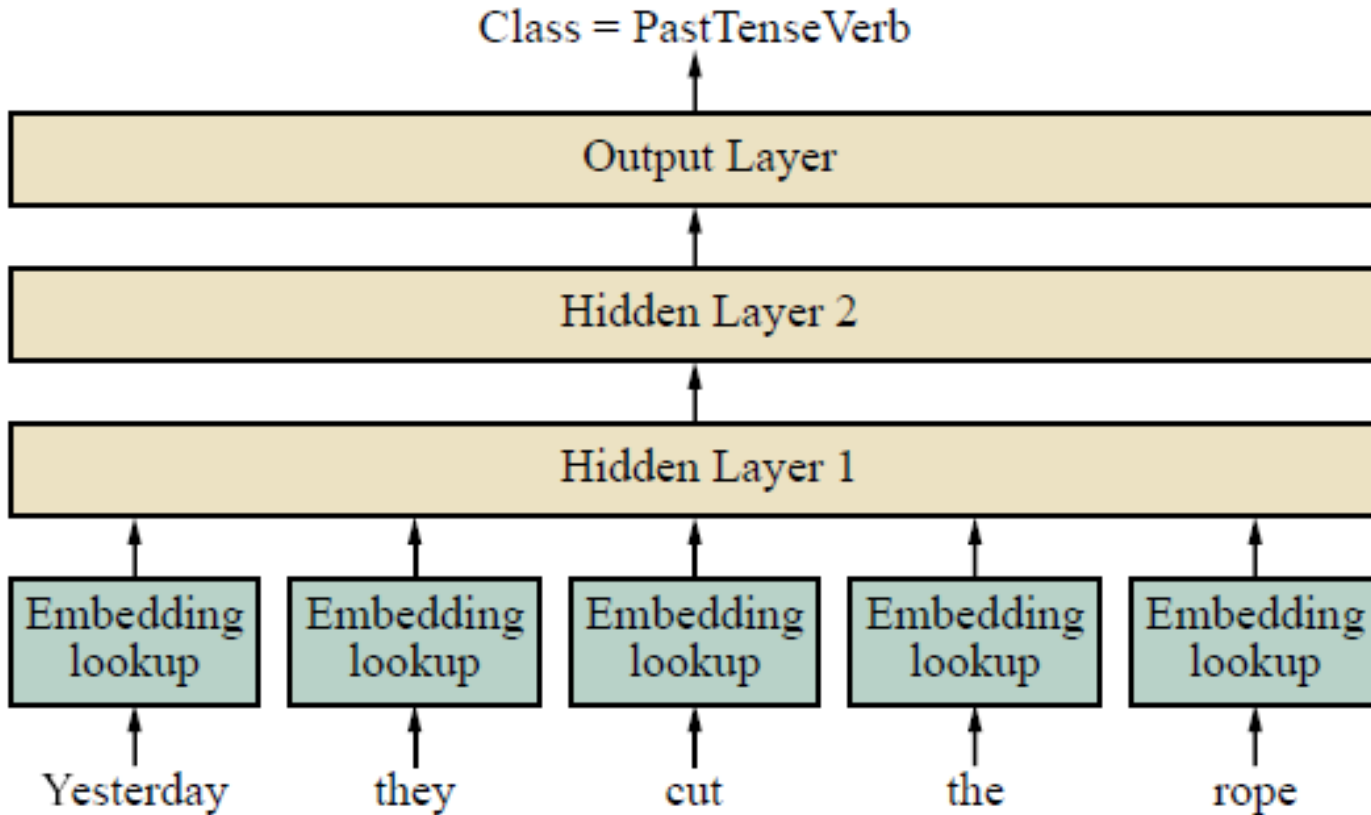
- L'étiquetage grammatical (*part-of-speech* ou *POS tagging* en anglais) consiste à identifier les catégories grammaticales d'un texte: nom, verbe, adjectif, etc.
- C'est une étape importante dans l'analyse syntaxique
- Ce n'est pas un problème facile parce que des mots peuvent être catégorisés différemment selon le contexte.
 - ◆ Exemple en français: courant
- L'identification implique une certaine prédiction du mot qui devrait le plus probablement suivre étant donné ceux observés jusqu'à date

Étiquetage grammatical par un RNN



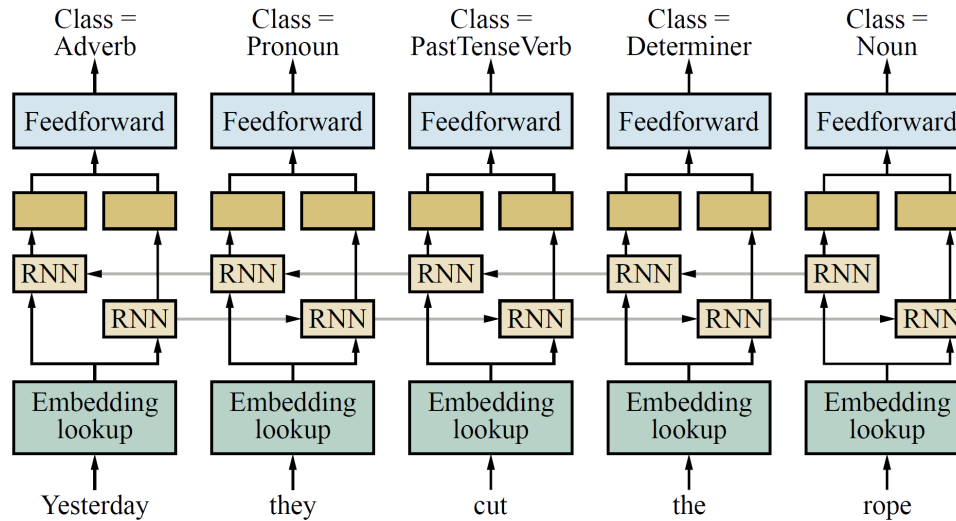
Un RNN peut aussi apprendre l'étiquetage grammatical

Étiquetage grammatical par un réseau *feedforward*



Le modèle prend en entrée une fenêtre de 5 mots et prédit l'étiquette du mot au milieu

Génération du texte



Une fois entraîné, le modèle peut générer du texte.

*Mary, and will, my lord, to weep in such a one were prettiest
Yet now I was adopted heir
Of the world's lamentable day
To watch the next way with his father with his face?*

C'est une façon d'évaluer la qualité du modèle. Plus le modèle est bon, plus il génère des textes vraisemblables (GPT et BERT sont des modèles du langage très connus basés sur l'architecture *Transformer* non couvert dans ce cours)

Conclusion

- Ce cours ne donne qu'une introduction au RNN et à deux de ses applications au traitement du langage naturel
- Les RNNs ont beaucoup d'autres applications sur les données séquentielles: traitement du langage, maintenance prédictive, analyse de séries temporelles et plus.
- Cours plus avancés:
 - ◆ **IFT 607 – Traitement automatique des langues naturelles** (cours de maîtrise)
 - ◆ **IFT 725 – Réseaux neuronaux** (cours de maîtrise)

Concepts et algorithmes

```
graph TD; A[Vision par ordinateur] --> B[Traitement du Langage naturel];
```

Vision par ordinateur

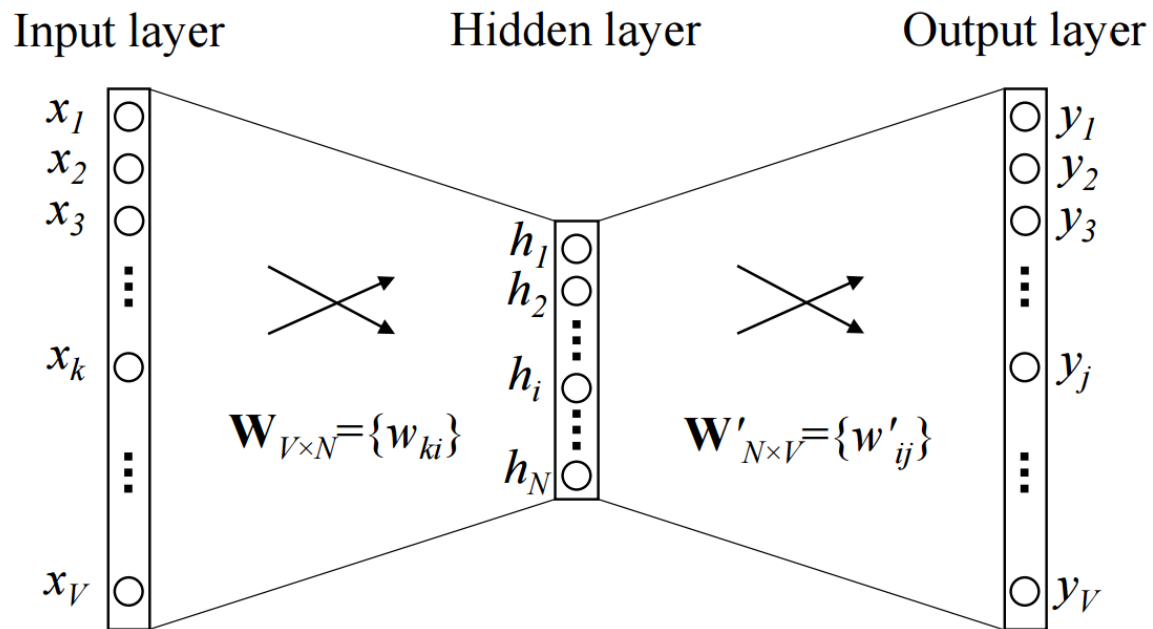
Traitement du Langage naturel



Vous devriez être capable de...

- Décrire l'architecture RNN
- Expliquer la source des problèmes de gradients évanescents et d'explosion de gradients dans un RNN
- Expliquer ce qu'un *word embedding* et comment le créer
- Expliquer le problème d'étiquetage grammatical.
- Expliquer en quoi le RNN est utile à l'étiquetage grammatical

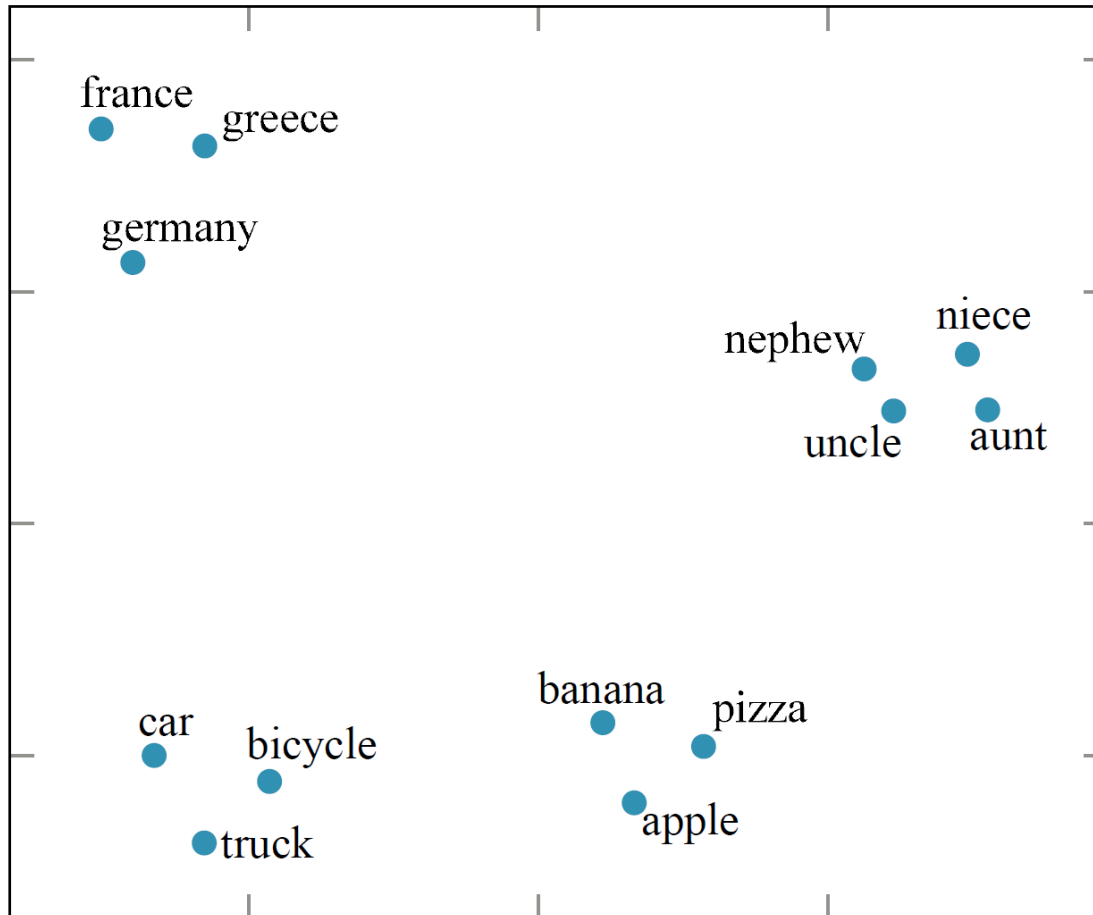
Réseau de neurone pour *word embedding*



Architecture Common Bag of Words (CBOW) de Word2Vec

Source : [\(Karani, Towards Data Science, 2018\)](#)

Vecteurs de *word embeddings* calculés par GloVe



GloVe a 6 milliards de mots

Vecteurs de 100 dimensions

On voit que les mots apparentés apparaissent les uns proches des autres

Un *word embedding* peut représenter des relations peu triviales

A	B	C	$D = C + (B - A)$	Relationship
Athens	Greece	Oslo	Norway	<i>Capital</i>
Astana	Kazakhstan	Harare	Zimbabwe	<i>Capital</i>
Angola	kwanza	Iran	rial	<i>Currency</i>
copper	Cu	gold	Au	<i>Atomic Symbol</i>
Microsoft	Windows	Google	Android	<i>Operating System</i>
New York	New York Times	Baltimore	Baltimore Sun	<i>Newspaper</i>
Berlusconi	Silvio	Obama	Barack	<i>First name</i>
Switzerland	Swiss	Cambodia	Cambodian	<i>Nationality</i>
Einstein	scientist	Picasso	painter	<i>Occupation</i>
brother	sister	grandson	granddaughter	<i>Family Relation</i>
Chicago	Illinois	Stockton	California	<i>State</i>
possibly	impossibly	ethical	unethical	<i>Negative</i>
mouse	mice	dollar	dollars	<i>Plural</i>
easy	easiest	lucky	luckiest	<i>Superlative</i>
walking	walked	swimming	swam	<i>Past tense</i>

Les *word embeddings* de chacun de ces mots permettent de répondre à la question «Quel est le mot similaire à C comme B est similaire à A ?»