IFT 615 – Intelligence Artificielle

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

Professeur: Froduald Kabanza

Assistants: D'Jeff Nkashama

UDS Université de Sherbrooke

Motivation

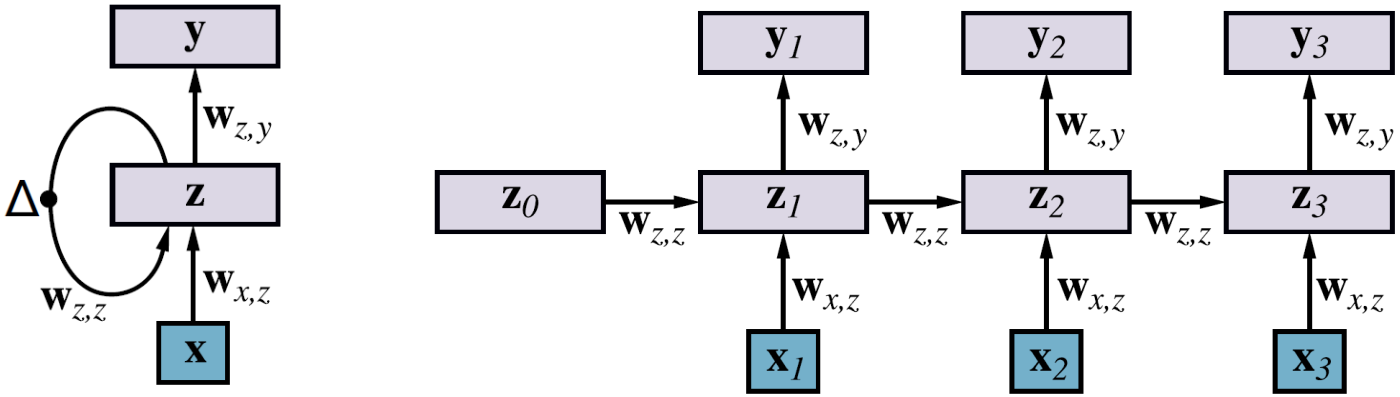
- Le langage est une des caractéristiques fondamentaux qui distingue les humains des animaux
- Plusieurs applications
  - Traduction automatique
  - Interaction humain-machine : Siri, robots
  - Service à la clientèle
  - Automatisation des processus (*robotic process automation*)
  - Classification de documents
  - Compréhension du texte
  - Cybersécurité – écoute électronique; détection de menaces
  - ChatGPT et LLM en général
  - Pus encore

Sujets couverts

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

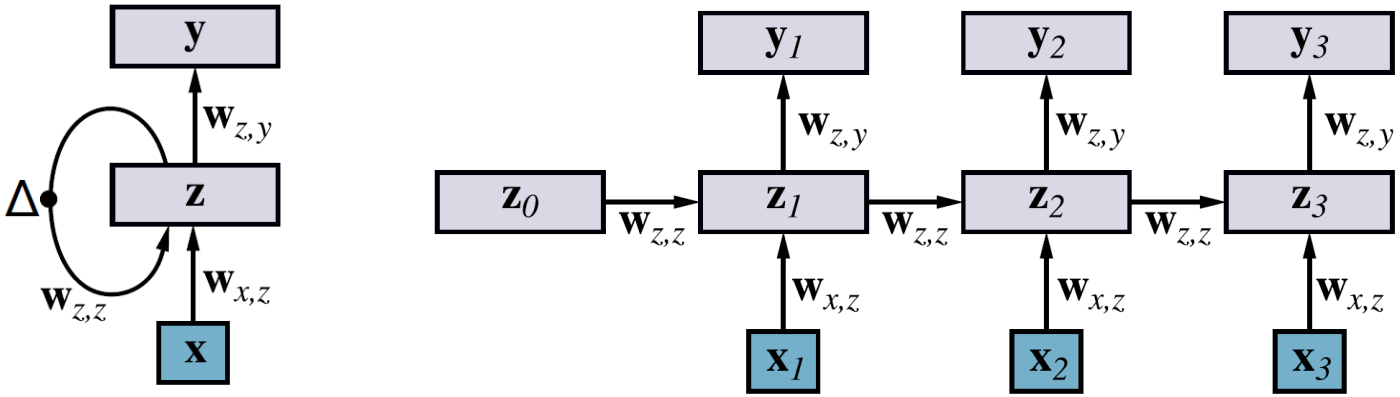
Réseau de neurone récurrent

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



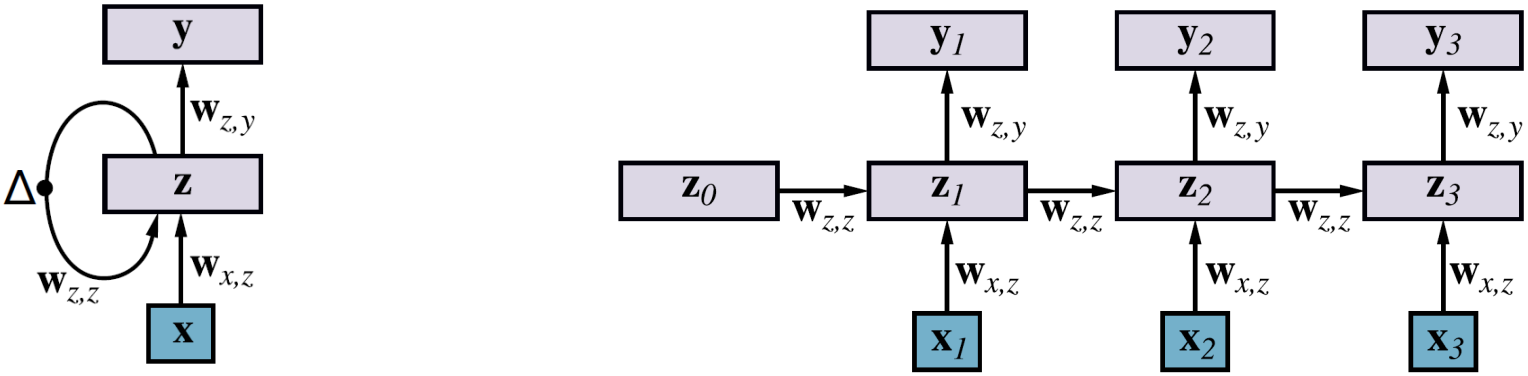
- z : couche (état) cache
- Δ est un délai
- **Hypothèse Markovienne**: l'état cache z_t suffit pour representer l'information de toutes les entrées précédentes
- Notons $z_t=f_w(z_{t-1},x_t)$, f_w étant la function paramétrique apprise par le reseau
  - On peut montrer que f_w est un **processus homogène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

RNN plus expressif que *Feedfoward*



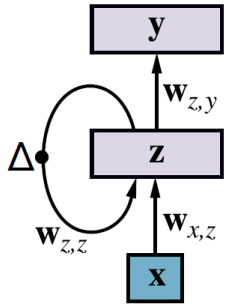
- $z_t=f_w(z_{t-1},x_t)$, f_w étant la function paramétrique apprise par le reseau
- On peut montrer que f_w est un **processus homogène en temps**.
- Ainsi un RNN ajoute un pouvoir d'expressivité au reseau *Feedfoward*

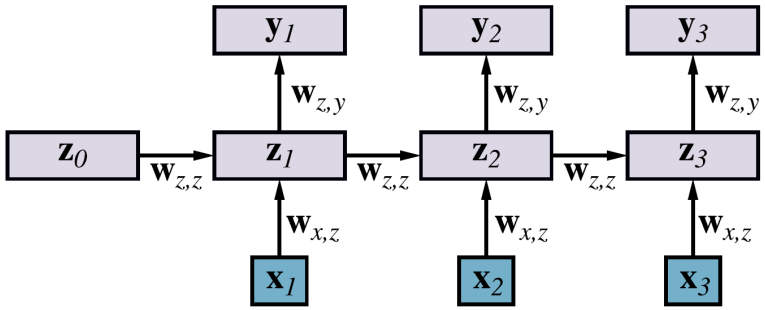
Entrainement d'un RNN



- $z_t=f_w(z_{t-1},x_t)=g_z(W_{z,z}\,Z_{t-1} + W_{x,z}\,x_t) \equiv g_z(in_{z,t})$
- $Y_t=g_z(W_{z,y}\,z_t) \equiv g_y(in_{y,t})$
- Étant donné une sequence de vecteurs d'entrée $x_1, \ldots, x_T$ et une sequence de sortie $y_1, \ldots, y_T$, on peut dérouler le RNN en un un reseau feedforward
- On peut alors calculer les gradients pour entrainer les poids.
- Cela conduit à l'algorithme de **propagation des gradients à travers le temps** (couvert par la Section 21.6.1 du livre; non couvert à l'examen):

Limites du RNN





- Pour cet exemple, le gradient pour la couche interne (voir Sec. 21.6.1):
$\frac{\partial z_t}{\partial W z,z} = g'_z(in_{z,t})(z_{t-1} + w_{z,z}\frac{\partial z_{t-1}}{\partial W z,z})$
- Ainsi, au temps T les gradients contiendront des termes proportionnels à $w_{z,z}\boldsymbol{\Pi}_{\boldsymbol{t=1}}^{\boldsymbol{T}} g'_z(in_{z,t})$
- Pour la *sigmoïde*, *tahn* et *ReLu*, on a *g' ≤ 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** qui deviant trop grand.

Word Embedding

- Les réseaux de neurones prennent des vecteurs comme entrées
- Pour le traitement du langage naturelle, on voudrait une représentation des mots tells que les mots apparentés 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-entrainés: Word2Vec, GloVe, FASTTEXT
- Chaque word embedding est juste un vecteur de valeurs numériques sans apparente signification

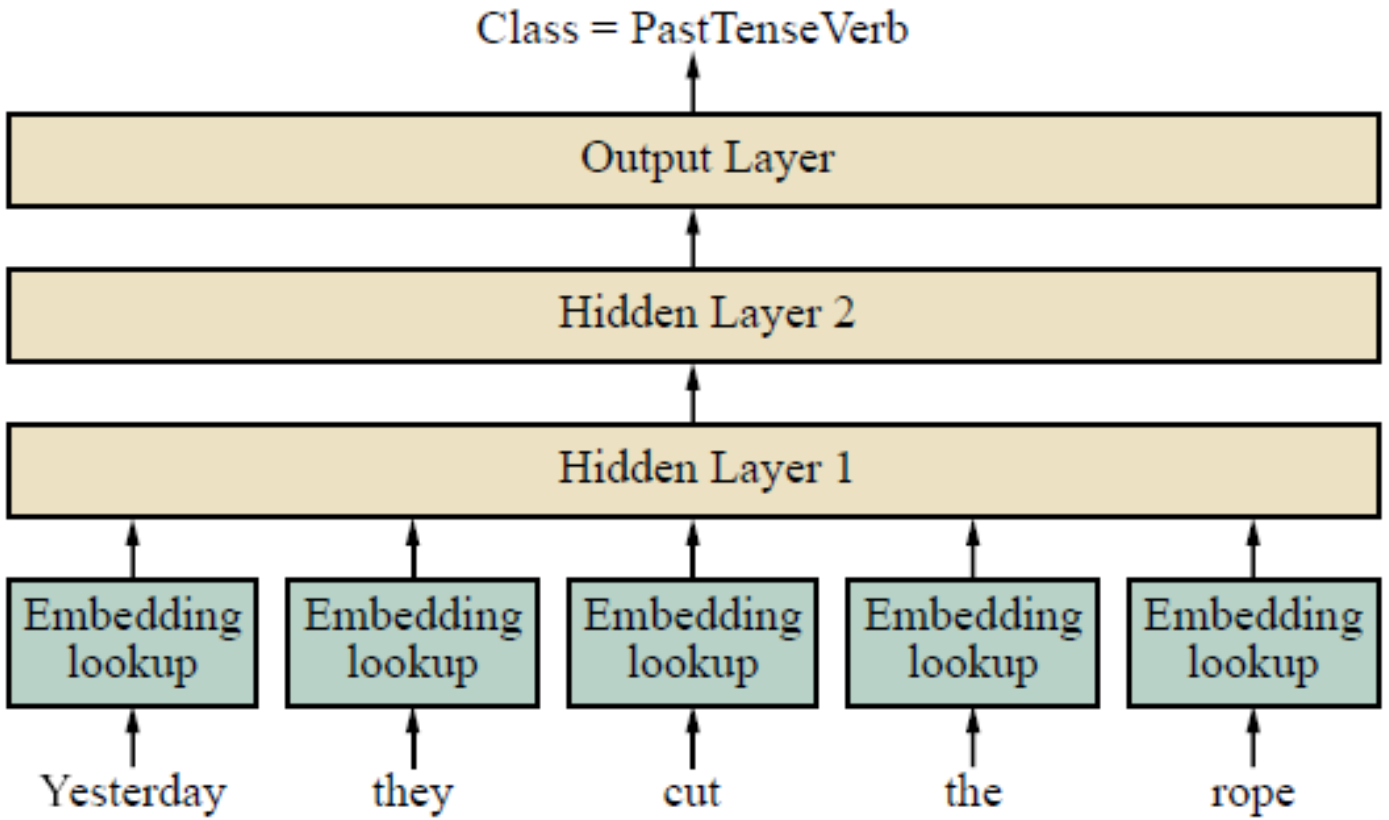
  « aadrdvark » = [-0.7, +0.2, -3.2, …]
  « abbacus » = [+0.5, +0.9, -1.3, …]
  …
  « zyzzyva » = [-0.1, +0.8, -0.4, …]

- Mais les mots apperentés ont des représentations proches

Étiquetage grammatical

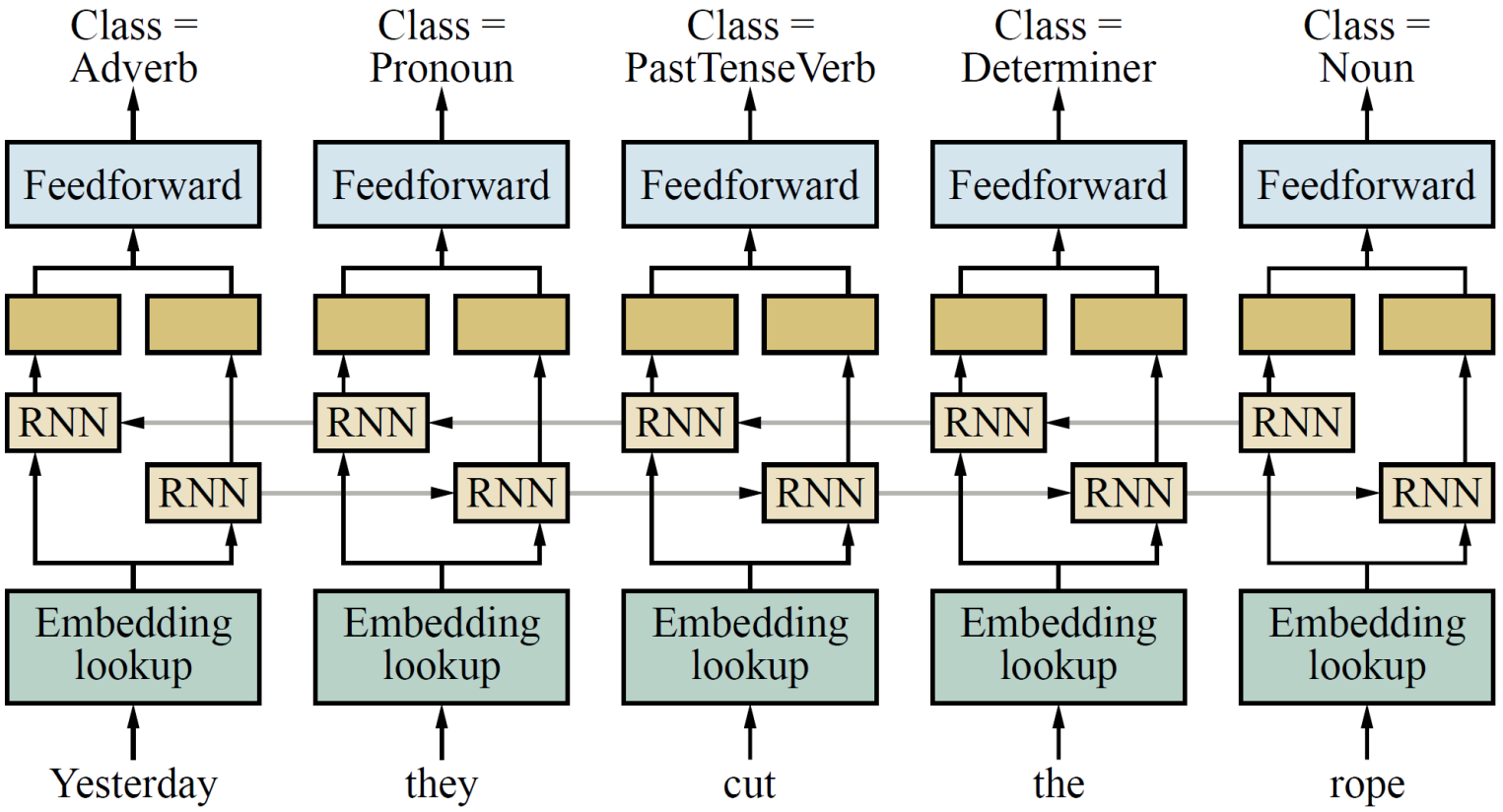
- L'étiquetage grammatical (*part-of-speech* ou *POS tagging* en anglais) consiste à identifier les catégories grammaticales d'un texte: nom, verbe, adjectif, etc.
- C'est une étape importante dans l'analyse syntaxique
- Ce n'est pas un problème facile parce que des mots peuvent être catégorisés différemment 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 réseau *feedfoward*



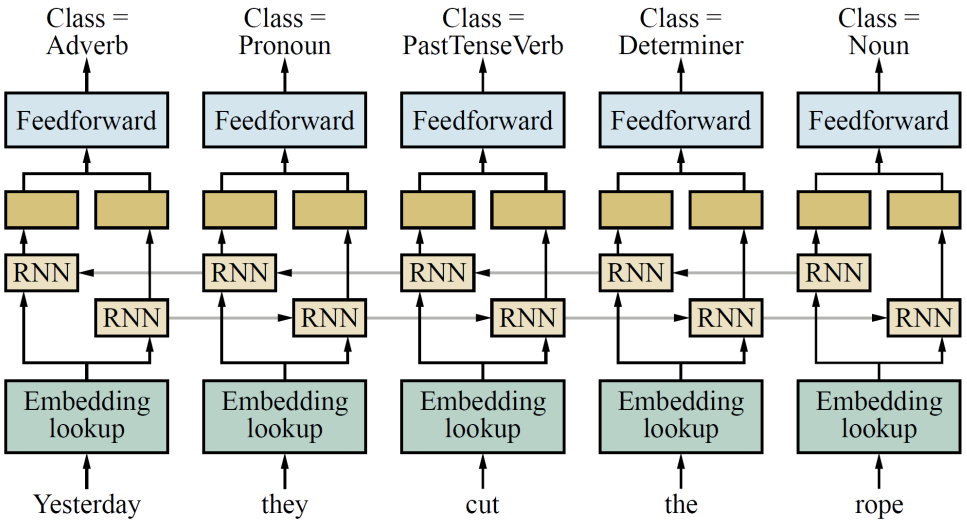
Le modèle prend en entrée une fenêtre de 5 mots et prédit l'etiquette du mot au milieu

Étiquetage grammatical par un RNN



Un RNN peut aussi apprendre l'étiquetage grammatical

Génération du texte



Une fois entrainé, 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 langages très connus basés sur l'architecture *Transformer* non couvert dans ce cours)

Conclusion

- Cette leçon a introduit le concept de RNN. Ce qu'il faut retenir est que grâce à l'introduction de la récurrence, on est capable de traiter des données séquentielles, par exemple le langage naturelle.

- Les RNNs ont beaucoup d'autres applications sur les données séquentielles: traitement du langage, maintenance prédictive, analyse de séries temporelles et plus.

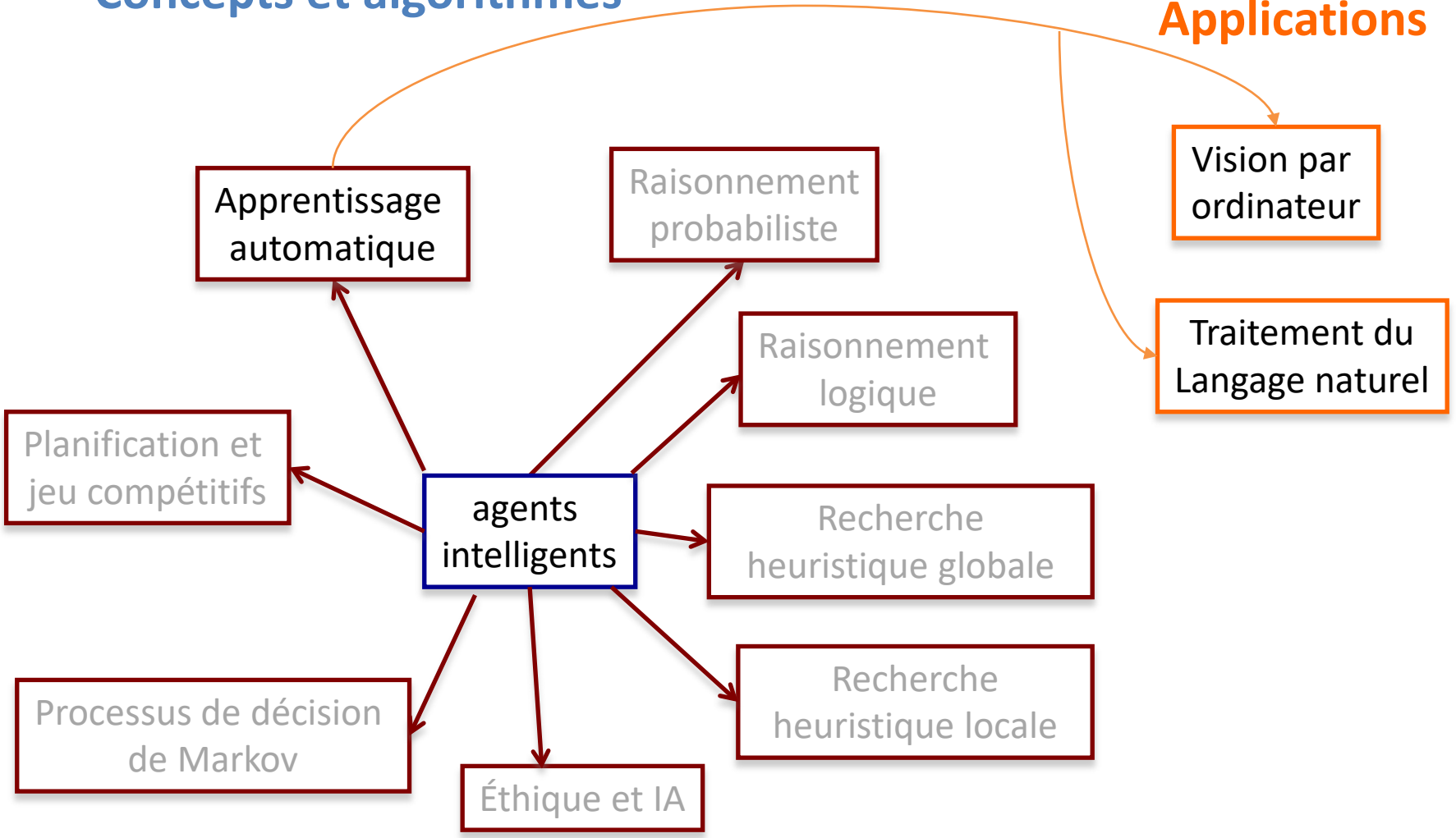
- Cette leçon a illustré l'étiquetage syntaxique et la génération de texte.

- Cours plus avancés:
  - **IFT 607 – Traitement automatique des langues naturelles** (cours de maîtrise)
  - **IFT 725 – Réseaux neuronaux** (cours de maîtrise)

Sujets couverts

Concepts et algorithmes

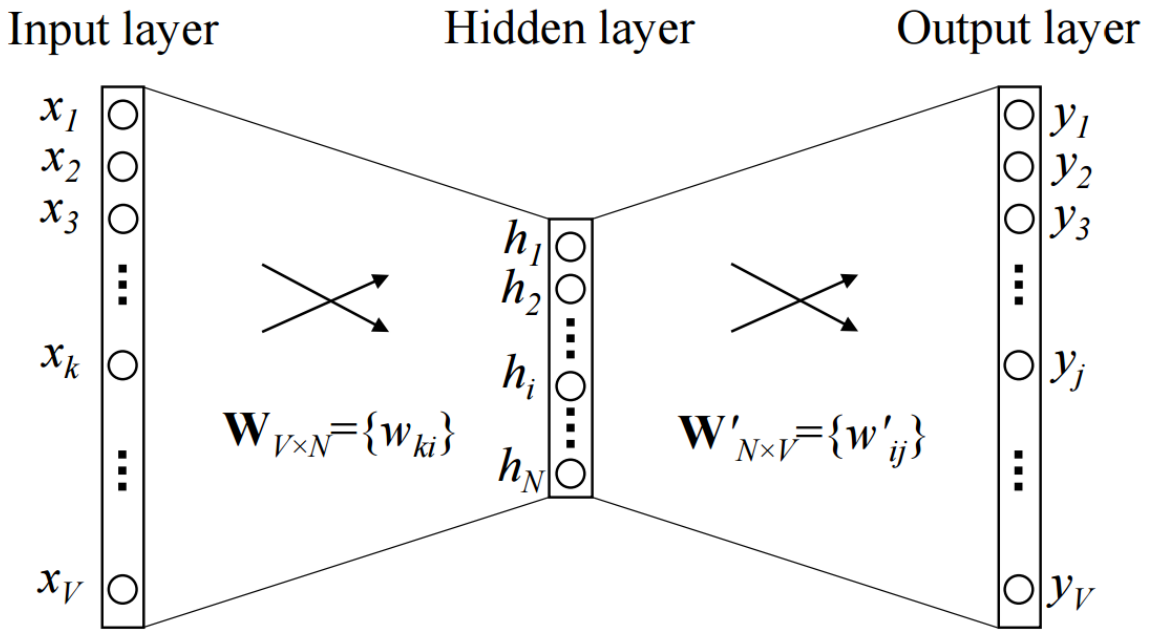
Applications



Vous devriez être capable de...

- Expliquer la différence entre un RNN et un perceptron multi-couche
- Expliquer la source des problèmes des gradients évanescents 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

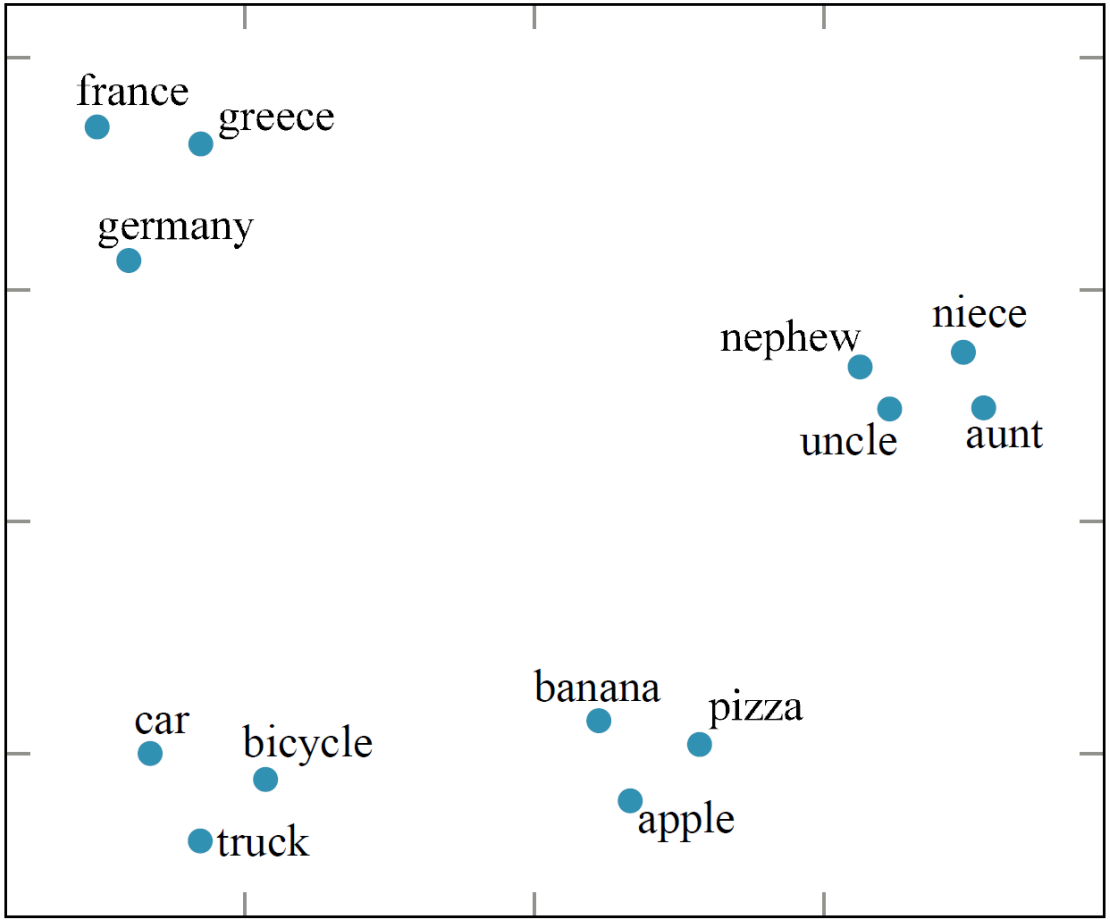
Réseau de neurone pour *word embedding*



Architecture Common Bag of Worlds (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 un proche des autres

Un *word embedding* peut représenter des relations peu triviales

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

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 ?»