IFT 615 – Intelligence Artificielle

Application – Traitement du langage naturel : *world embedding et étiquetage grammatical*

Professeur: Froduald Kabanza

Assistants: D'Jeff Nkashama et Léo Chartrand

UDS Université de Sherbrooke

Motivation

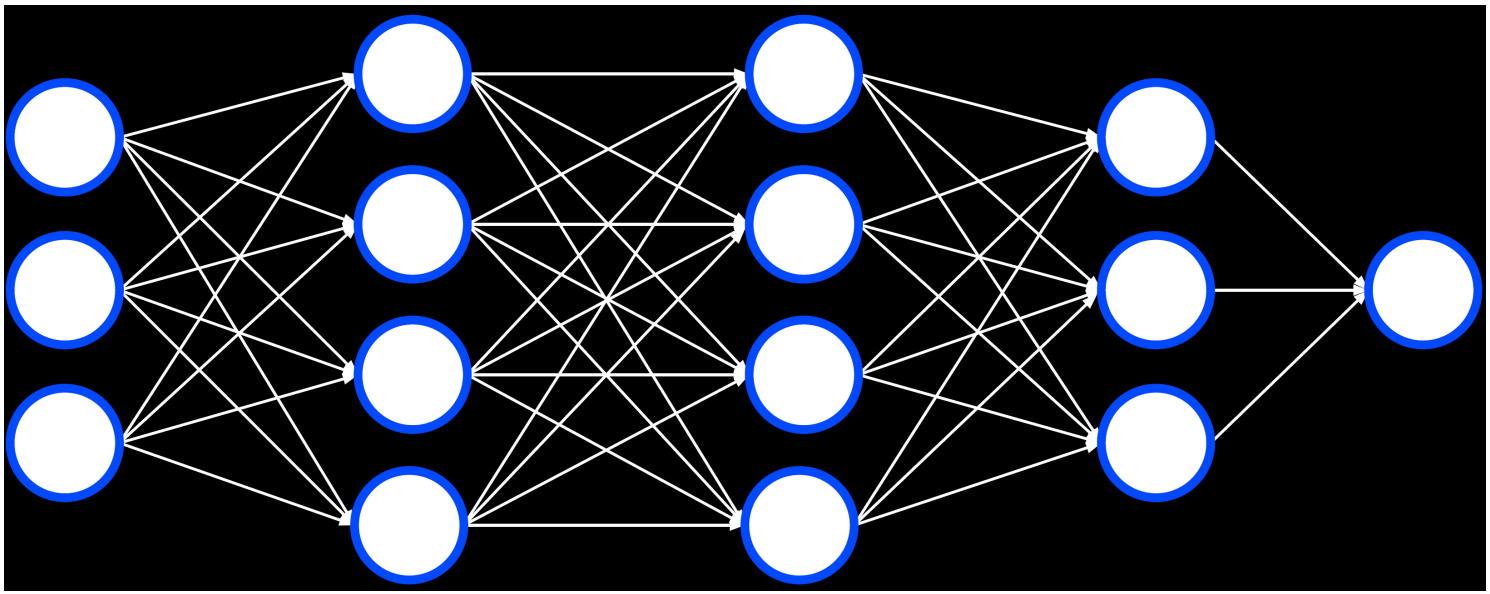
- Le langage est une capacité cognitive qui distingue les humains des animaux
- Le traitement du langage naturel comprend, entre autres, les grands modèles de langage comme ChatGPT et a plusieurs applications:
  - Traduction automatique
  - Interaction humain-machine
  - Résumé des documents
  - Génération de contenu
  - Cybersécurité – écoute électronique; détection de menaces

Sujets couverts

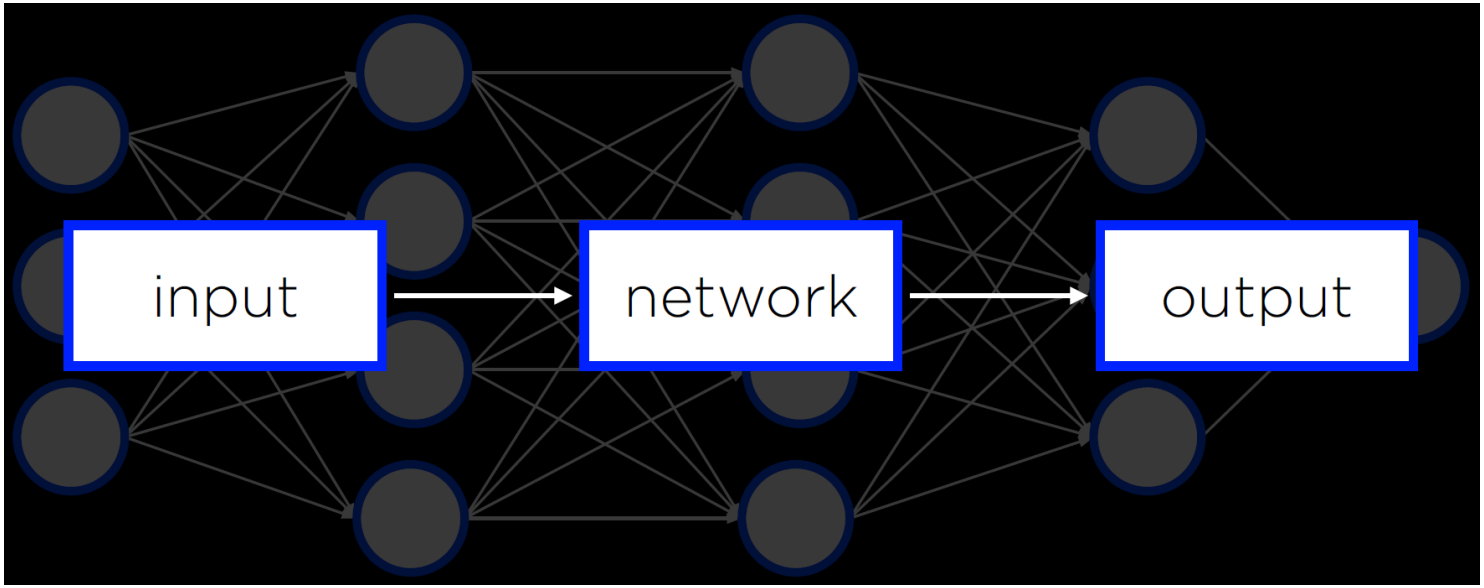
- Représentation des mots par des *Word embeddings*
- Application à l'étiquetage grammaticale
- Réseau de neurones récurrent (RNN)

Réseau de neurone artificiel

Réseau de neuronne *feedfoward* (multi-perceptron)

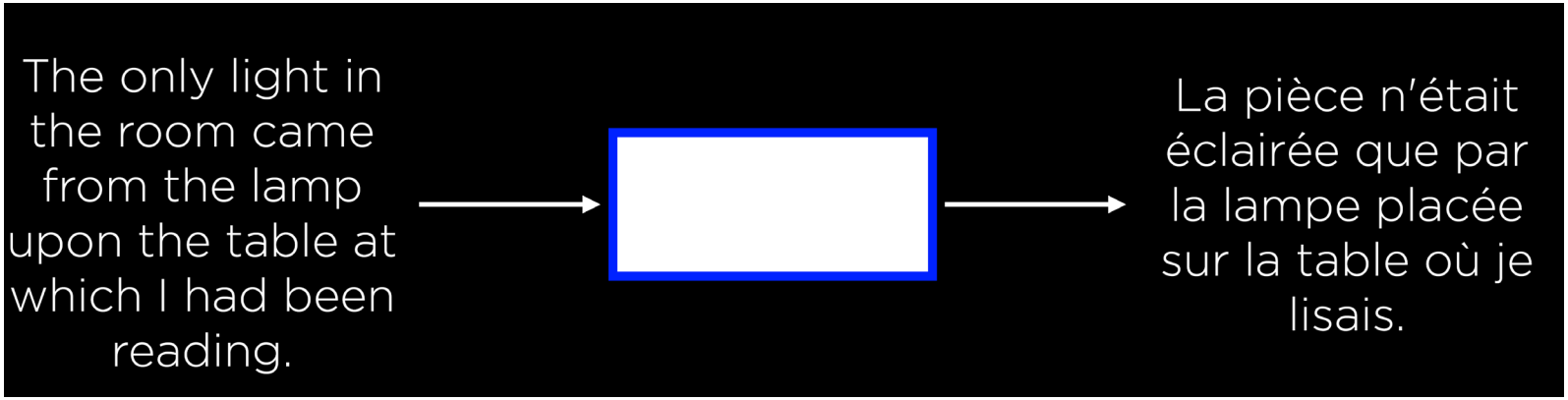


Réseau de neurone artificiel



Un de neurone prend des vecteurs d'entrées numériques

Réseau de neurone artificiel



Pour le traitement du langage naturel, il nous faut une représentation numérique des mots

Word Embedding

- Les réseaux de neurones prennent des vecteurs de nombres comme entrées

- Pour le traitement du langage naturelle, on voudrait une représentation des mots telle 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, de sorte que les mots apparentés ont des vecteurs proches.

Word Embedding

- Un *word embedding* est appris par un réseau de neurones sur un corpus.

  - Exemples: Word2Vec, GloVe, FASTTEXT

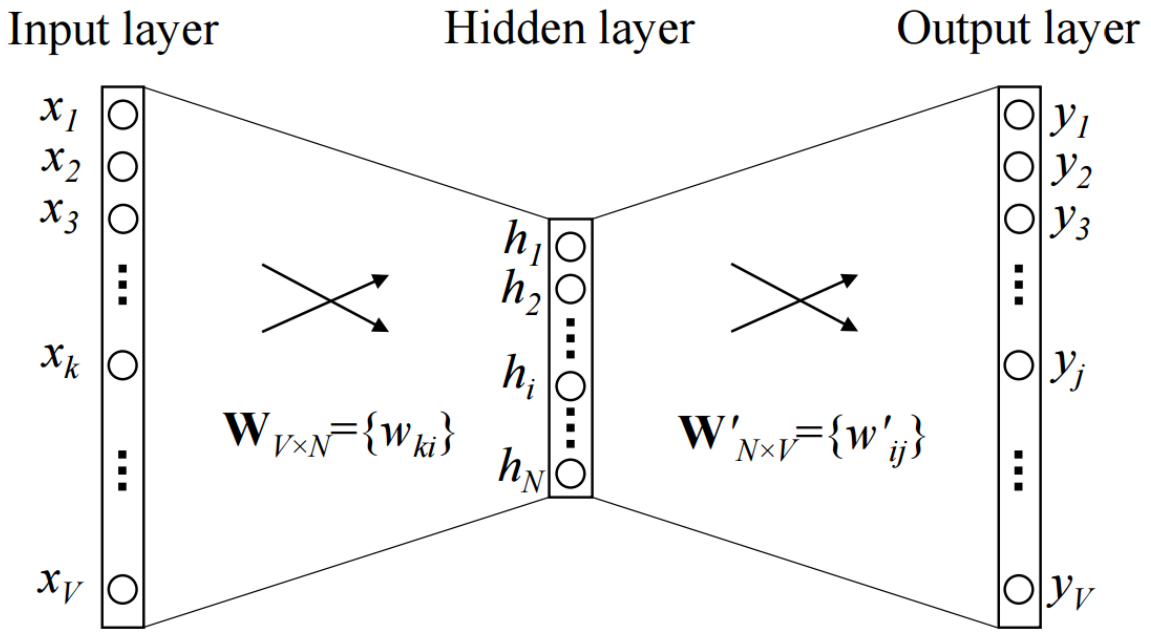
- Chaque word embedding est juste un vecteur de valeurs numériques sans apparente signification

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

- Mais les mots apparentés ont des représentations proches

[repas, déjeuner, souper, soupe]

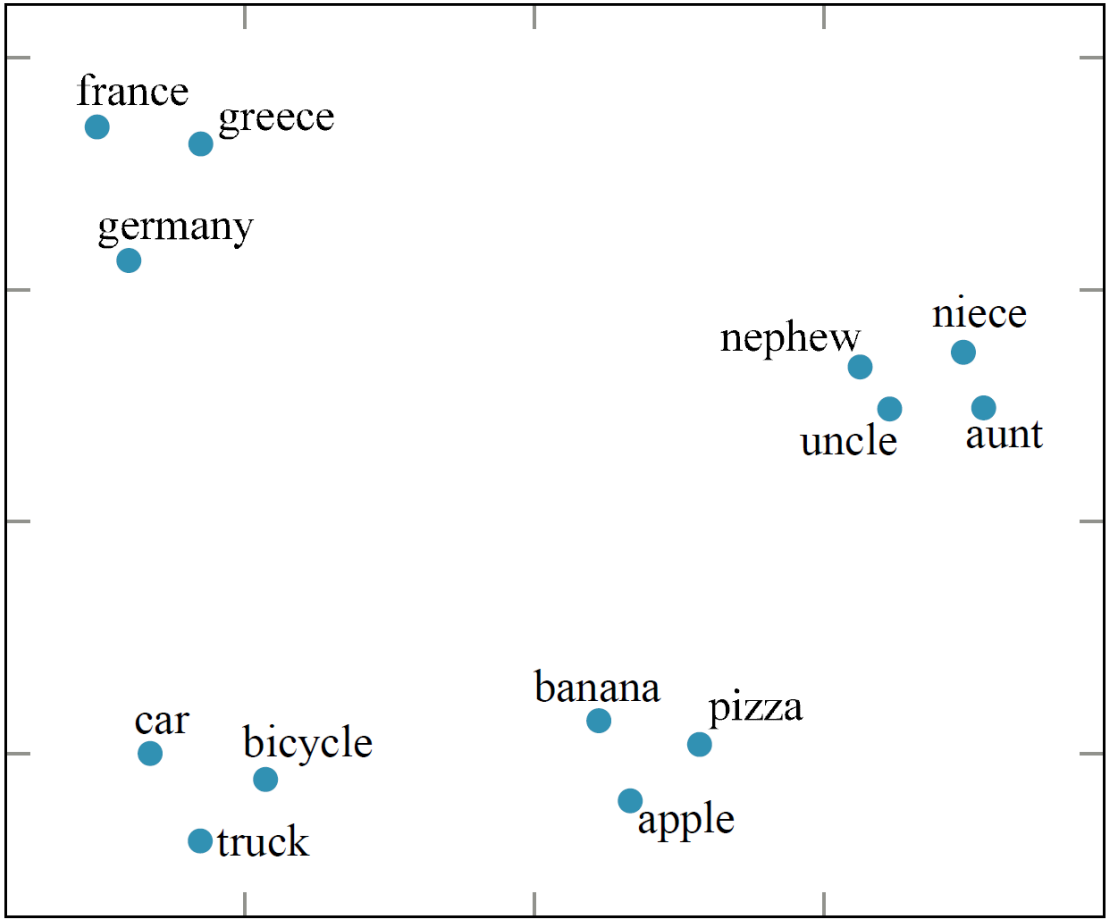
Réseau de neurone pour *word embedding*



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

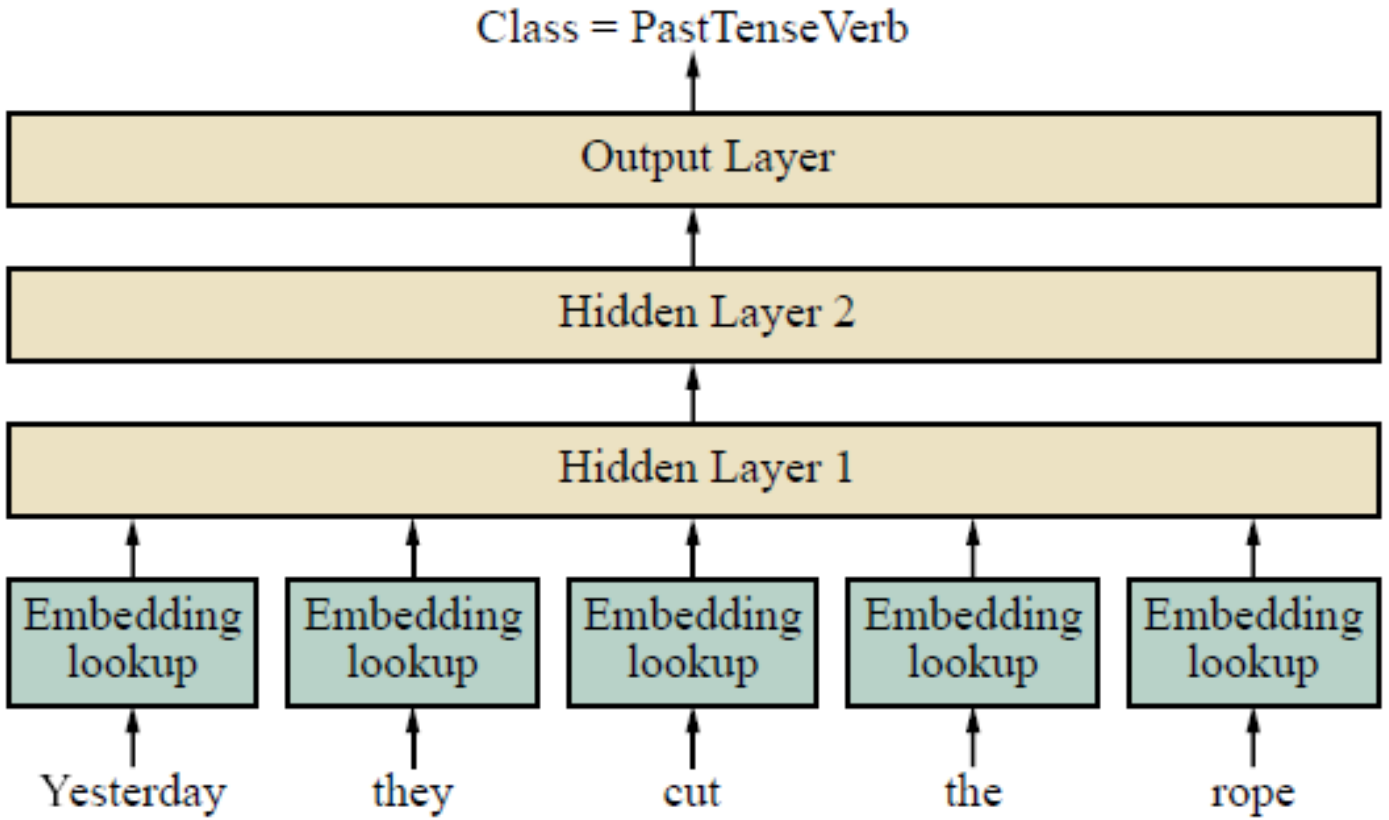
A	B	C	D = C + (B − A)	Relationship
Athens	Greece	Oslo	Norway	*Capital*
Astana	Kazakhstan	Harare	Zimbabwe	*Capital*
Angola	kwanza	Iran	rial	*Currency*
copper	Cu	gold	Au	*Atomic Symbol*
Microsoft	Windows	Google	Android	*Operating System*
New York	New York Times	Baltimore	Baltimore Sun	*Newspaper*
Berlusconi	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 ?»

Étiquetage grammatical

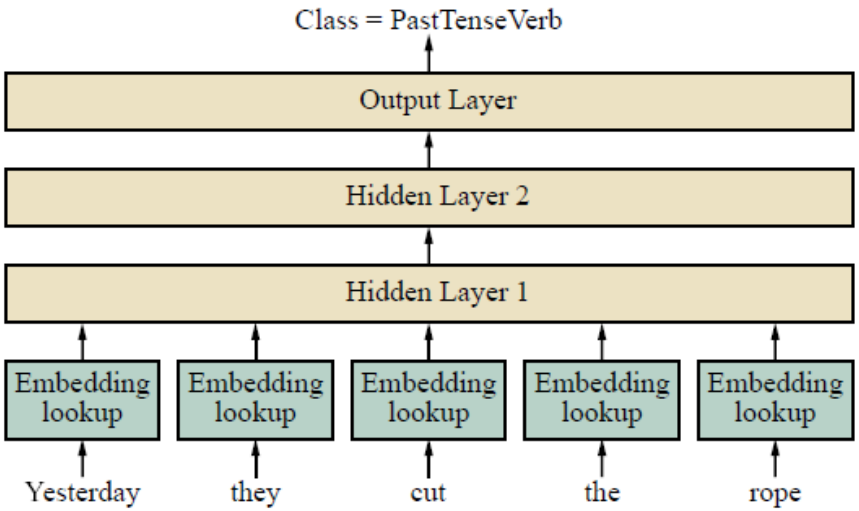
- L'étiquetage grammatical (*part-of-speech* ou *POS tagging* en anglais) consiste à identifier les catégories grammaticales d'un texte: nom, verbe, adjectif, etc.
- C'est une étape importante dans l'analyse syntaxique
- Ce n'est pas un problème facile parce que des mots peuvent être catégorisés différemment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

Génération du texte

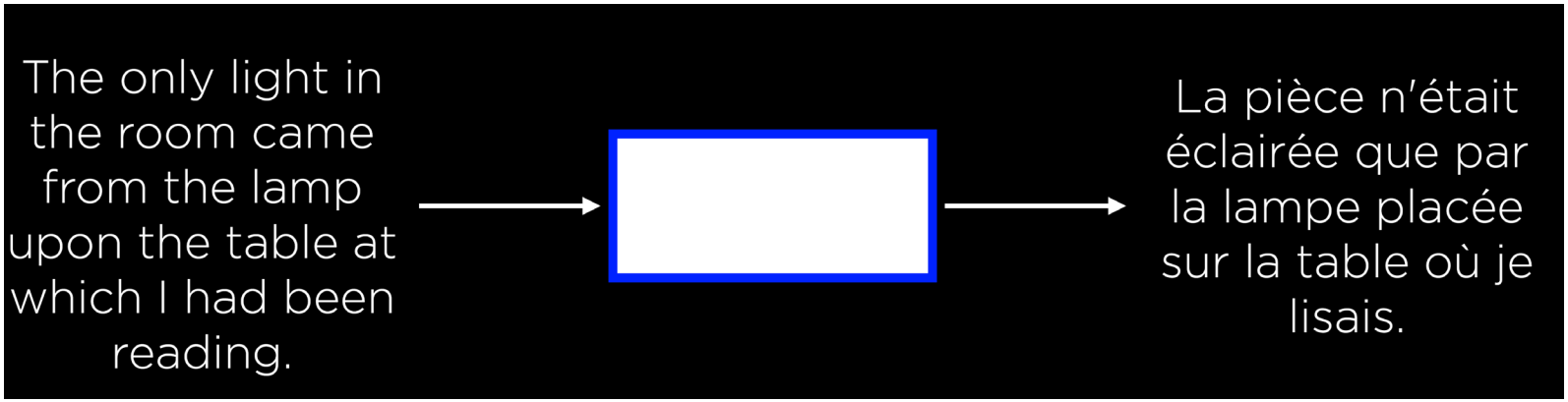


Une fois entrainé, le réseau est un **modèle de langage**. Il peut générer du texte.

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?

Plus un modèle de langage est bon, plus il génère des textes vraisemblables (GPT et BERT sont des modèles du langages plus puissant sur l'architecture *Transformer* non couvert dans ce cours)

Réseau de neurone artificiel

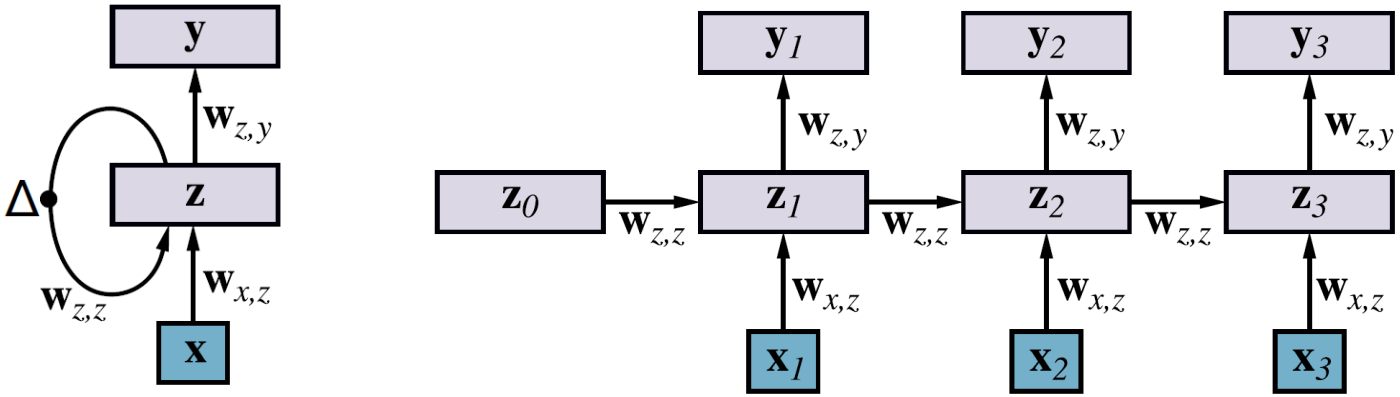


Un réseau feedfoward est limité pour le langage naturel:

- Longueur arbitraire des textes
- De longs textes signifieraient un long vecteur d'entrée, un réseau de neurones complexe

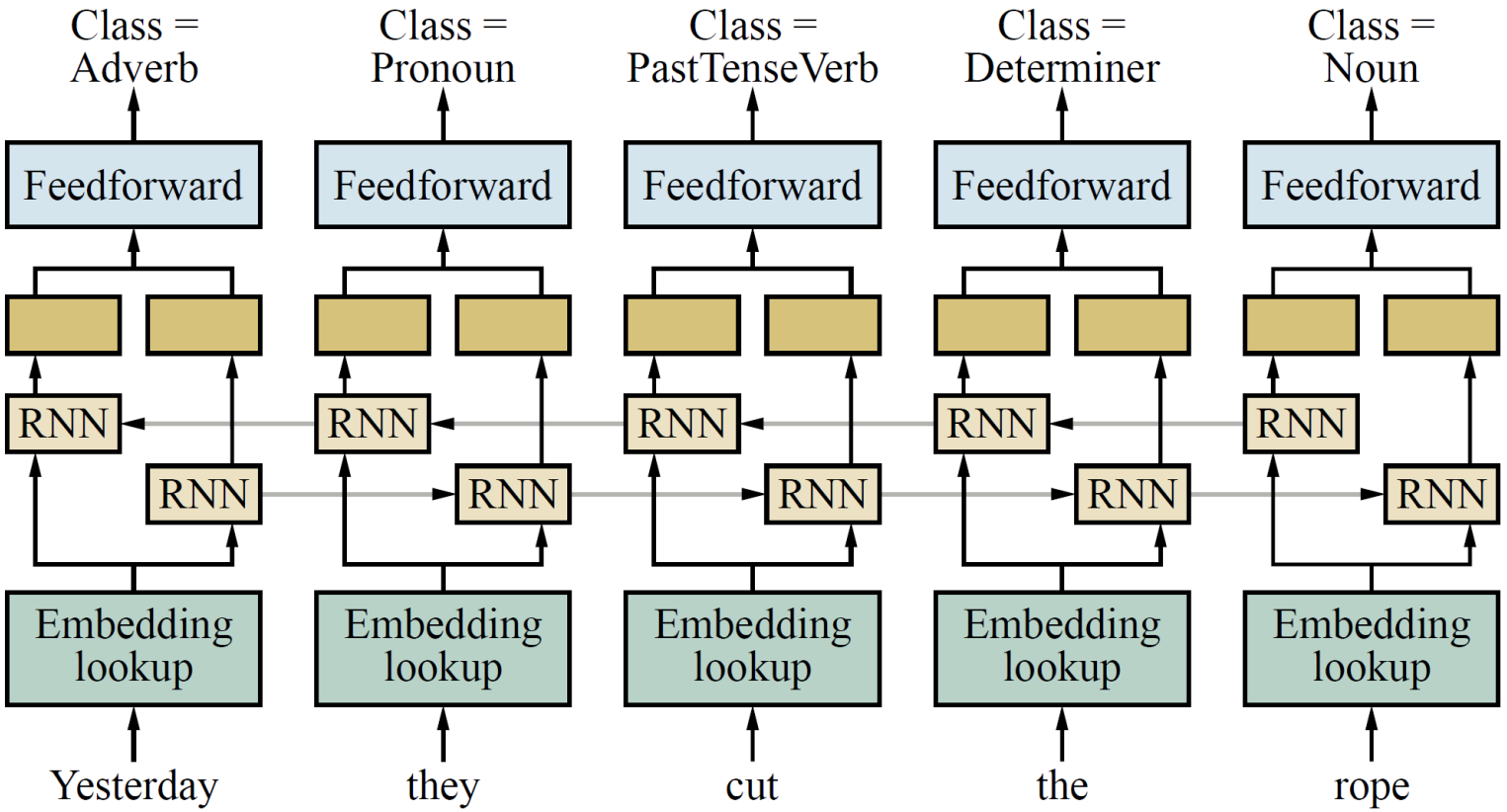
Réseau de neurone récurrent

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

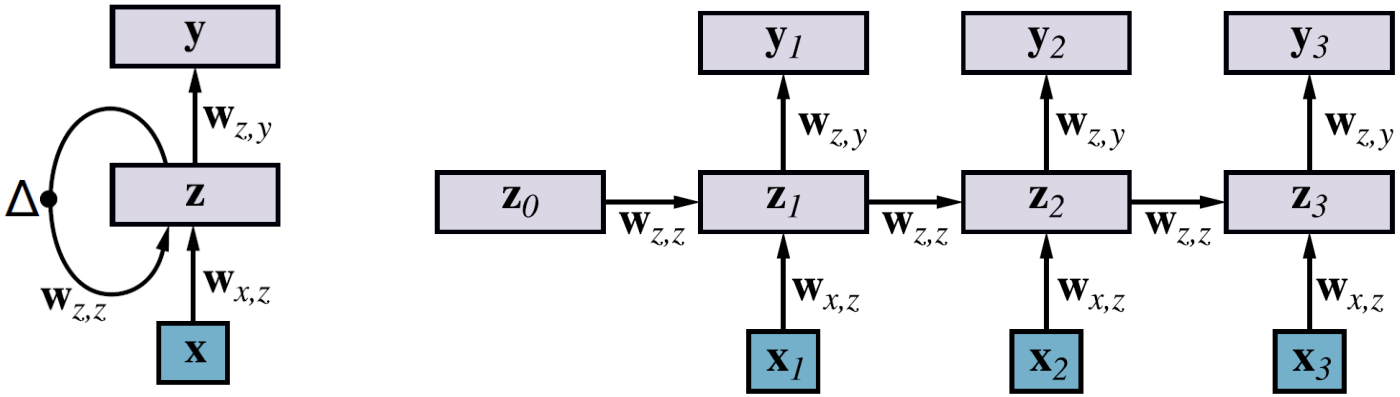


- z : couche cachée
- Δ est un délai
- Étant donné une sequence de vecteurs d'entrée $x_1, \ldots, x_T$ et une sequence de sortie $y_1, \ldots, y_T$, on peut dérouler le RNN en un un reseau feedforward
- $z_t = g_z(W_{z,z} Z_{t-1} + W_{x,z} x_t) \equiv g_z(in_{z,t})$
- $Y_t = g_z(W_{z,y} z_t) \equiv g_y(in_{y,t})$

Étiquetage grammatical par un RNN



Limitation des RNN

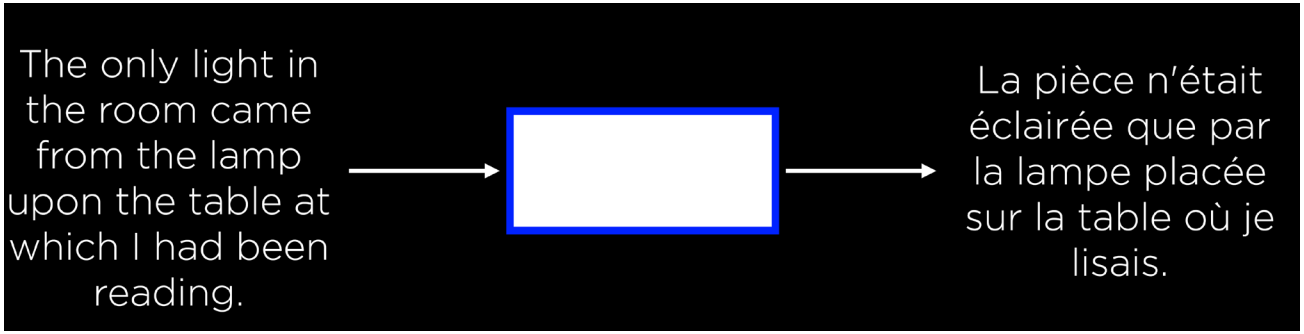


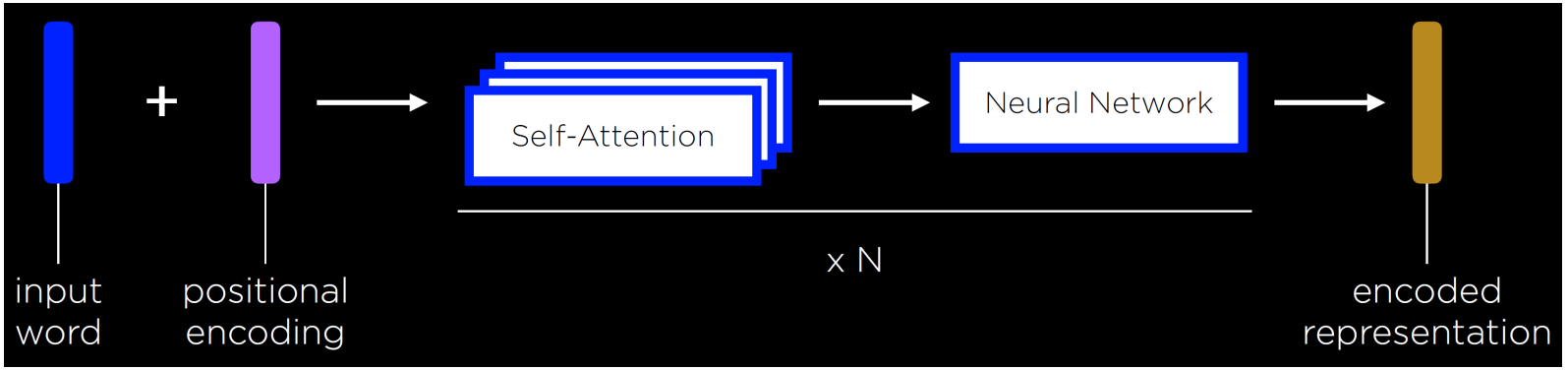
- Gradient évanescent
- Séquentiel – pas paralléisable, donc inefficaces à entrainer

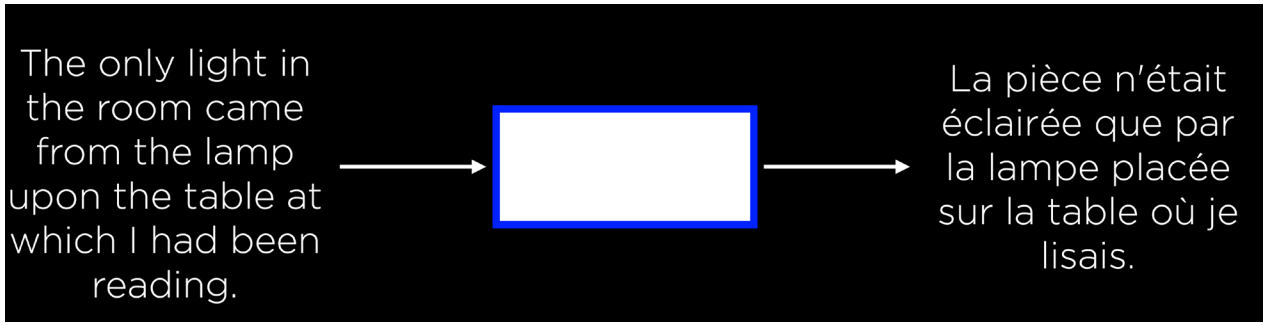
Transformer

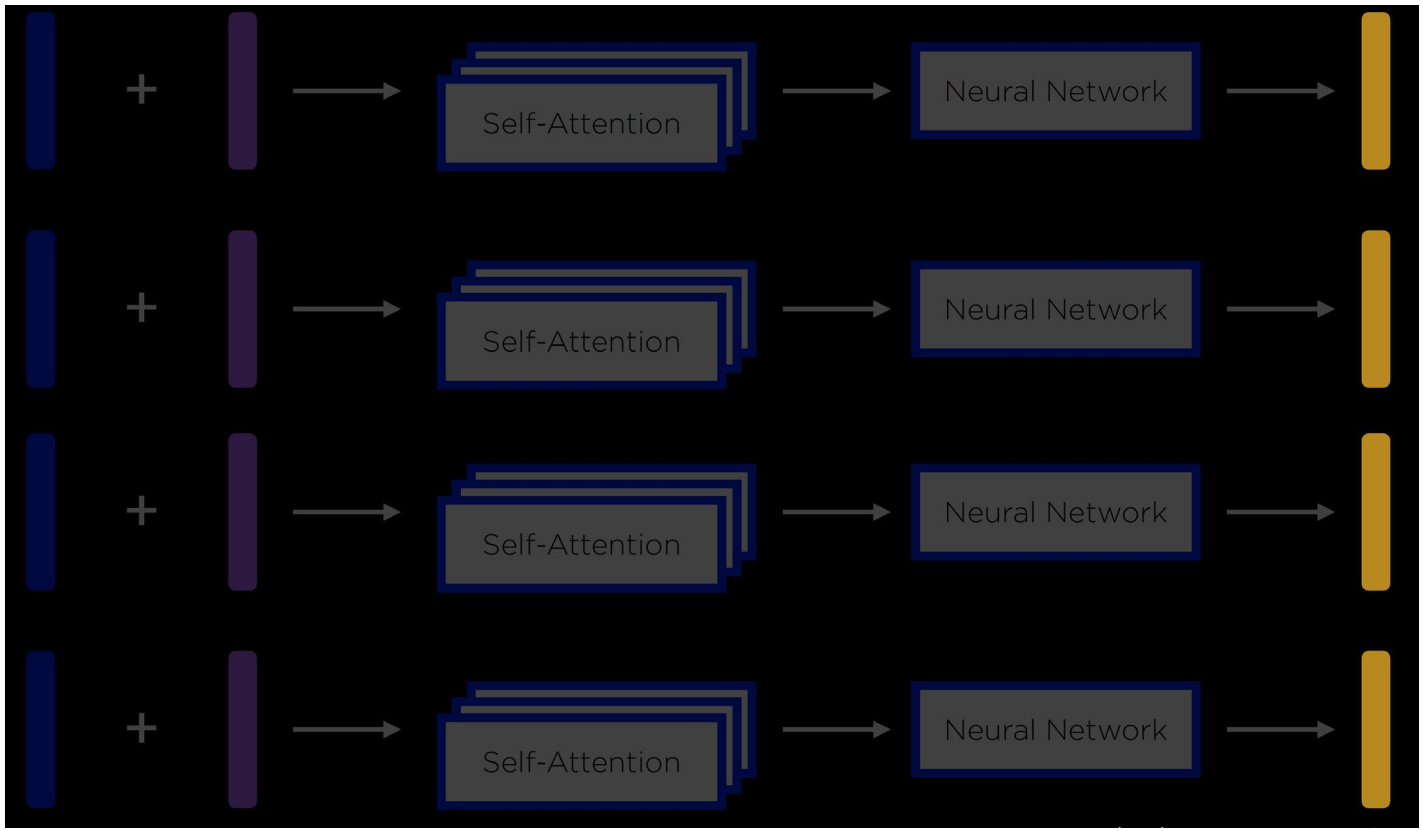
Une architecture de réseau de neurones pour le traitement des données séquentiel basé sur des concepts d'**attention** et de **codage de position**.

Les grands modèles de langages (LLM) comme ChatGPT utilise des *transformers*


The only light in the room came from the lamp upon the table at which I had been reading.
La pièce n'était éclairée que par la lampe placée sur la table où je lisais.


+
Self-Attention
Neural Network
x N
input word
positional encoding
encoded representation


The only light in the room came from the lamp upon the table at which I had been reading.
La pièce n'était éclairée que par la lampe placée sur la table où je lisais.


Self-Attention
Neural Network
Self-Attention
Neural Network
Self-Attention
Neural Network
Self-Attention
Neural Network

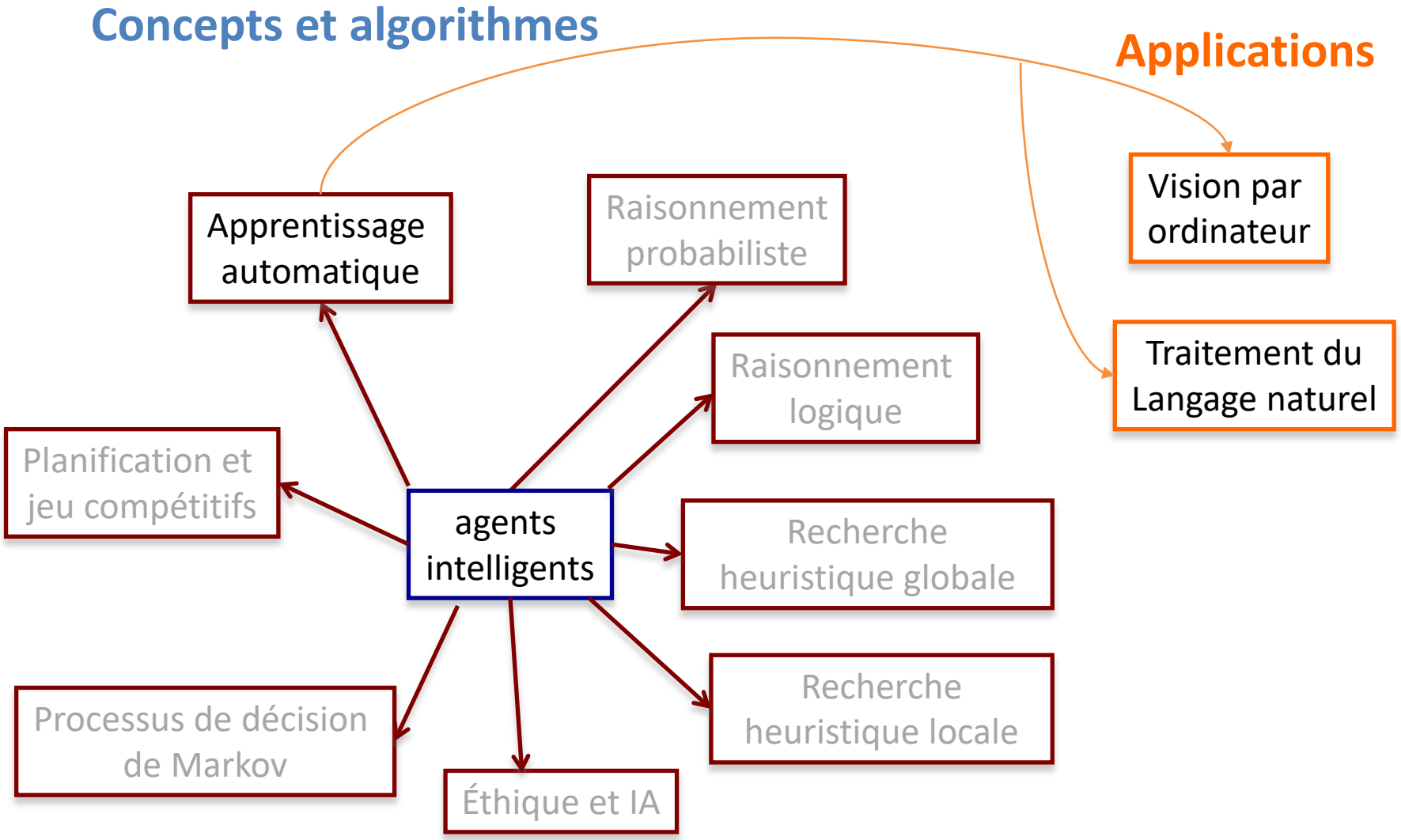
Conclusion

- 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 naturel.

- Par contre le RNN a des limitations (traitement séquentiel, gradient évanescent). Ils ont pendant longtemps étaient l'architecture de prédilection pour les données séquentielles, mais sont aujourd'hui remplacés par le *transformer* pour beaucoup d'applications.

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

Sujets couverts



Vous devriez être capable de...

- Expliquer la différence entre un RNN et un réseau feedforward
- Expliquer ce qu'un *word embedding* et comment le créer
- Expliquer l'application d'étiquetage grammatical.